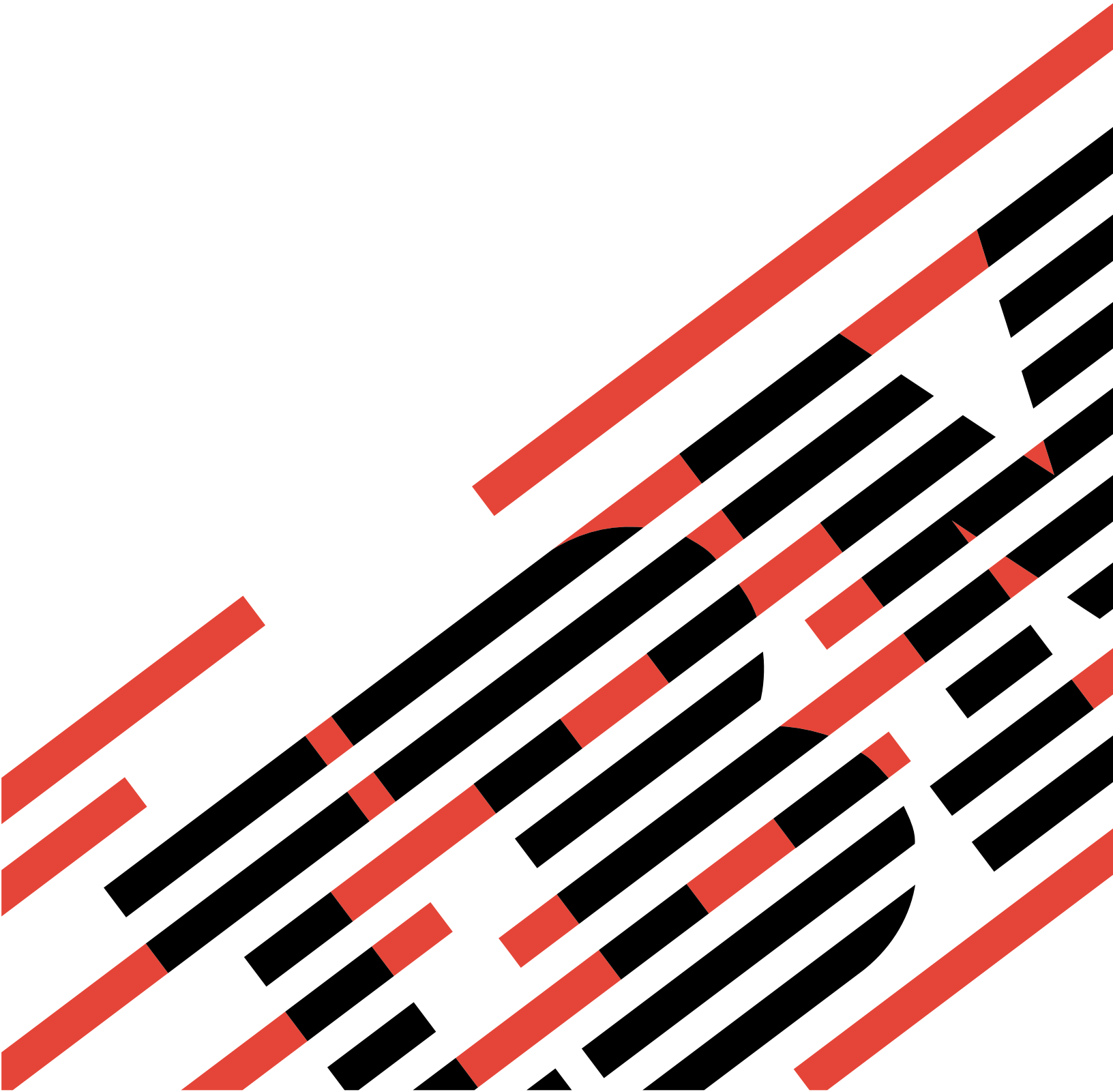
IBM

eserver

iSeries

iSeries용 DB2 Universal Database - 데이터베이스 성능 및 조회 최적화



IBM

@server

iSeries

iSeries용 DB2 Universal Database - 데이터베이스 성능 및 조회 최적화

목차

iSeries용 DB2 UDB 데이터베이스 성능 및 조회 최적화에 대한 정보

여기서는 프로그래머와 관리자를 대상으로 다음 내용에 대해 설명합니다.

- 데이터베이스 어플리케이션에서 최상의 성능을 얻기 위해 iSeries용 DB2 UDB에서 사용 가능한 툴 및 기능을 사용하는 방법
- iSeries용 DB2 UDB 통합 데이터베이스의 기능을 최대한 활용하는 조회를 실행하는 방법

iSeries용 DB2 UDB 지침 및 어플리케이션 프로그래밍 환경에서 이를 구현하는 예에 대한 자세한 정보는 iSeries Information Center의 데이터베이스 및 파일 시스템 범주에 있는 다음 정보를 참조하십시오.

- SQL 참조서
- SQL 프로그래밍 개념
- 호스트 언어를 사용한 SQL 프로그래밍
- SQL 호출 레벨 인터페이스(ODBC)
- 데이터베이스 프로그래밍
- Query/400 사용
- ODBC
- SQLJ

Java 데이터베이스 연결(JDBC) 정보는 iSeries™ Information Center에 있는 프로그램 범주의 Java™용 IBM® Developer Kit에 있습니다.

고급 데이터베이스 기능에 대한 추가 정보는 *DATABASE 2/400 Advanced Database Functions*, GG24-4249를 참조하십시오.

데이터베이스 성능 및 조회 최적화 서적을 읽어야 할 대상

이 서적은 기본 데이터베이스 어플리케이션을 이해하고 조회 조정 방법을 이해하려는 프로그래머와 데이터베이스 관리자들을 대상으로 합니다. 여기서는 조회 성능 문제를 디버깅하기 위해 사용 가능한 툴을 사용하는 방법을 보여줍니다.

이 서적의 사용자는 다음을 포함하여 언어 및 인터페이스에 익숙해야 합니다.

- iSeries용 COBOL
- iSeries용 ILE COBOL
- iSeries PL/I
- iSeries용 ILE C
- ILE C++
- iSeries용 VisualAge® C++
- REXX

- RPG III(iSeries용 RPG의 일부)
- iSeries의 ILE RPG
- Query/400
- OPNQRYF 명령
- 호출 레벨 인터페이스(CLI)
- ODBC
- JDBC

SQL문 예에 관한 가정

이 서적에 있는 조회의 예는 SQL 프로그래밍 개념의 부록 A, ″iSeries용 DB2 UDB 샘플 표″에 있는 샘플 표를 기초로 합니다. SQL 예에 대해 다음과 같이 가정합니다.

- 예는 대화식 SQL 환경에서 표시되거나 ILE C 또는 COBOL로 작성됩니다. EXEC SQL 및 END-EXEC는 COBOL 프로그램에서 SQL문을 분리하는 데 사용됩니다. COBOL 프로그램에서 SQL문을 사용하는 방법에 대한 설명은 호스트 언어를 사용한 SQL 프로그래밍의 ″COBOL 어플리케이션에서 SQL문 코드화″를 참조하십시오. ILE C 프로그램에서 SQL문을 사용하는 방법에 대한 설명은 호스트 언어를 사용한 SQL 프로그래밍의 ″C 어플리케이션에서 SQL문 코드화″를 참조하십시오.
- 각각의 SQL 예는 분리 행에 각각의 명령문 절을 사용하여 여러 행에 표시됩니다.
- SQL 키워드는 강조표시됩니다.
- SQL 프로그래밍 개념의 부록 A, ″iSeries용 DB2 UDB 샘플 표″에 제공된 표 이름은 콜렉션 CORPDATA를 사용합니다. 부록 A, ″iSeries용 DB2 UDB 샘플 표″에 없는 표 이름은 사용자가 콜렉션을 작성하여 사용하여야 합니다.
- 연산된 열은 소괄호() 및 대괄호[] 안에 넣습니다.
- SQL 명명 규칙이 사용됩니다.
- COBOL에서 디폴트 옵션이 아닐지라도 APOST 및 APOSTSQL 사전컴파일러 옵션이 사용됩니다. SQL문 및 호스트 언어 명령문 내의 문자 스트링 상수는 작은따옴표(')로 구분됩니다.
- 별도로 언급되지 않는 경우 *HEX의 정렬 순서가 사용됩니다.
- 일반적으로 SQL문의 전체 구문이 하나의 예에서 표시되지 않습니다. 이 안내서에 설명된 모든 명령문에 대한 전체 설명 및 구문은 SQL 참조서를 참조하십시오.

예가 이러한 가정과 다른 경우 다음과 같습니다.

이 안내서는 어플리케이션 프로그래머를 대상으로 하므로, 대부분의 예가 어플리케이션 프로그램에서 작성된 것처럼 보입니다. 그러나, 약간 변경하여 대화식 SQL을 사용하여 대화식으로 실행할 수 있는 예가 많이 있습니다. 대화식 SQL문을 사용하는 경우, SQL문의 구문은 프로그램에 삽입될 때 명령문 형식이 약간 달라집니다.

구문 다이어그램 해석 방법

이 서적에서, 구문은 다음과 같이 정의된 구조로 설명합니다.

- 행을 따라 왼쪽에서 오른쪽으로, 맨 위에서 맨 아래로 구문 다이어그램을 읽습니다.

►►— 기호는 명령문의 시작을 나타냅니다.

—► 기호는 명령문 구문이 다음 행에 계속됨을 나타냅니다.

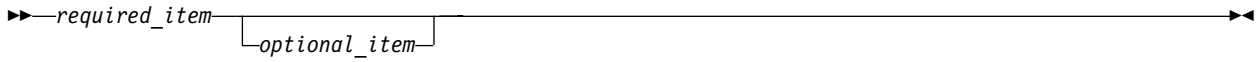
►— 기호는 명령문이 이전 행에서 계속됨을 나타냅니다.

—►◄ 기호는 명령문의 끝을 나타냅니다.

전체 명령문이 아닌 다른 구문 단위의 다이어그램은 ►— 기호로 시작하고 —► 기호로 끝납니다.

- 필수 항목은 수평 행(기본 경로)에 표시됩니다.

- 선택 항목은 기본 경로 아래에 표시됩니다.



선택 항목이 기본 경로 위에 표시되면, 해당 항목은 명령문 실행에 영향을 주지 않으며 읽기 전용으로만 사용됩니다.

- 두 개 이상의 항목을 선택할 수 있는 경우, 스택에 수직으로 표시됩니다.

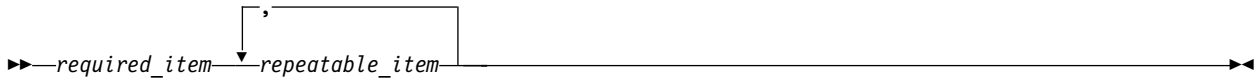
  항목 중 하나를 *반드시* 선택해야 하는 경우 스택 항목 중 하나가 기본 경로에 표시됩니다.

  항목 중 하나를 선택하는 것이 선택사항인 경우, 전체 스택이 기본 경로 아래에 표시됩니다.

  항목 중 하나가 디폴트인 경우, 기본 경로 위에 표시되며 나머지 선택사항은 아래에 표시됩니다.

- 기본 경로 위의 왼쪽으로 리턴하는 화살표는 반복될 수 있는 항목을 표시합니다.

  반복 화살에 쉼표가 들어 있는 경우, 반복된 항목을 쉼표로 분리해야 합니다.



  스택 위의 반복 화살표는 스택에 있는 항목을 반복할 수 있음을 표시합니다.

- 키워드는 대문자로 표시됩니다(예: FROM). 키워드 철자는 표시된 대로 정확히 써야 합니다. 모든 변수는 소문자로 표시됩니다(예: *column-name*). 변수는 사용자가 제공한 이름 또는 값을 나타냅니다.
- 구두점, 소괄호, 산술 연산자 또는 기타 기호가 표시된 경우, 해당 기호를 구문의 일부로 입력하여야 합니다.

V5R2의 새로운 사항

| 이 서적에서 다루는 새로운 주요 피처는 다음과 같습니다.

| • 도달순 결합

| • 조회 엔진 재설계

| • 통계 관리자

| 코드 면책사항 정보

| 이 문서에는 프로그래밍 예제가 들어 있습니다.

| IBM은 귀하에게 유사한 기능을 귀하의 특정 요구에 맞게 조정하여 생성할 수 있도록 모든 프로그래밍 코드 | 예제를 사용할 수 있는 비독점적인 저작권 사용권을 부여합니다.

| 모든 샘플 예제는 IBM에 의해 예시 목적으로만 제공됩니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 | 것은 아닙니다. 따라서 IBM은 이들 프로그램의 신뢰성, 실용성 또는 기능에 대해 보증할 수 없습니다.

| 여기에 포함된 모든 프로그램은 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 어떠한 종류 | 의 보증 없이 ″현상태대로″ 제공됩니다.

제 1 장 데이터베이스 성능 및 조회 최적화: 개요

데이터베이스 성능 조정의 목표는 조회의 응답 시간을 최소화하고, 네트워크 통신량, 디스크 I/O 및 CPU 시간을 최소화시켜 서버의 자원을 가장 잘 사용할 수 있도록 하는 것입니다. 이 목표는 자료의 논리 및 물리 구조에 대한 이해, 서버에서 사용되는 응용프로그램에 대한 이해 및 데이터베이스의 여러가지 상충적인 사용이 데이터베이스 성능에 영향을 줄 수 있는 방법에 대한 이해를 통해서만 성취할 수 있습니다.

성능상의 문제점을 피할 수 있는 가장 좋은 방법은 성능 문제가 진행 중인 개발 활동의 일부로 되어 있는지 확인하는 것입니다. 데이터베이스 개발 주기의 시초부터 신중한 설계를 통해 가장 좋은 성능 향상을 실현할 수 있습니다. 성능을 가장 효과적으로 최적화하려면, 가장 다양한 상황에 대해 가장 큰 성능 증가를 산출하는 영역을 식별하고 이 영역에 대한 분석에 초점을 맞춰야 합니다.

| V5R2에서는 iSeries용 DB2 UDB의 조회 엔진이 재설계되어 많은 SQL 읽기 전용 조회의 성능이 개선되었
| 습니다. 재설계로 인해 성능이 향상된 조회 유형, 새로운 개선사항을 이용하여 Optimizer를 지원하는 방법 등
| 개선된 성능에 대해 자세한 정보를 제공하는 레드북이 2002년 말에 출판될 예정입니다.

액세스 경로 및 조회 Optimizer에 대한 이해

iSeries는 여러 하드웨어 자원을 관리하고, 비용 기본 최적화 공식을 사용하여 SQL문 실행을 위해 가장 효율적인 액세스 계획을 결정하므로, 시스템이 가장 효율적인 액세스 방식을 판별하는 방법과 어떤 요소들이 시스템에 의한 선택을 결정하는지를 알아야 합니다. 이 내용은 자료 액세스 방법에서 다루고 있습니다. 또한, iSeries 조회 Otimizer를 명백히 이해함으로써, 조회 Optimizer의 비용 추정 및 의사결정 규칙을 레버리지하는 조회를 설계하는 데 도움을 받을 수 있습니다.

조회 향상

이 개념에 대해 잘 알고 있다면, 다음에 있는 자료를 검토하여 조회를 점차 향상시킬 수 있습니다.

주제	설명
조회 최적화 툴을 사용하여 조회 성능 최적화	조회에 관한 통계를 수집하고 조회의 처리를 제어하여 자료 검색 시간을 향상시키기 위해 조회 최적화 툴을 사용하는 방법에 대해 설명합니다. 이 툴이 제공하는 결과를 사용하여, 서버에서 선택된 자료 액세스 방식을 변경하거나 올바른 색인을 작성 및 이 색인을 효과적으로 사용할 수 있습니다.
대형 표에 빨리 액세스하기 위해 색인 사용	표 액세스에 대한 색인 기본 검색 방법을 설명하고, 숫자 변환, 연산식, 문자 스트링 채우기 및 유사 패턴 사용과 같은 것을 피함으로써 효과적으로 색인을 작성하는 방법을 설명합니다.
어플리케이션 설계를 통해 데이터베이스 성능 증가	사용자 어플리케이션의 올바른 설계가 어떻게 성능을 증가시킬 수 있는 지에 대해 설명합니다. 어플리케이션 설계 고려사항은 매개변수 전달 기법, 실제 자료 사용, 열기 조작 수 감소 및 커서 위치 보유를 포함합니다.
프로그래밍 기법을 사용하여 데이터베이스 성능 증가	올바른 프로그래밍 기법이 성능을 향상시킬 수 있는 방법에 대해 설명합니다. OPTIMIZE 절 사용, FETCH n ROWS 사용, INSERT n ROWS 사용, 데이터베이스 관리자 블로킹 제어, SELECT문으로 선택된 열 수의 최적화, 중복 유효성 제거, 대화식으로 표시된 자료 페이징과 같은 기법을 설명합니다.
일반 iSeries 성능	일부 일반 서버 고려사항 및 이 고려사항이 조회의 성능에 영향을 미치는 방법에 대해 설명합니다.

조회 작성

다음 인터페이스를 사용하여 조회를 작성할 수 있습니다.

- SQL
- 열린 조회 파일(OPNQRYF) 명령
- 열린 데이터베이스 연결(ODBC)
- iSeries용 조회
- Java 데이터베이스 연결(JDBC)
- 호출 레벨 인터페이스(CLI)

조회 Optimizer는 이 인터페이스들을 사용하여 작성한 모든 조회를 최적화합니다.

제 2 장 iSeries용 DB2 UDB에 대한 자료 액세스: 자료 액세스 경로 및 방식

이 절에서는 iSeries용 DB2 Universal Database 및 사용권 내부 코드가 조회 및 액세스 자료를 처리하기 위해 사용하는 자료 액세스 방식을 소개합니다. 자료 액세스 방식은 키순이 아닌 액세스, 키순 액세스 및 임시 결과 파일 액세스 방식으로 그룹화됩니다.

iSeries는 근본적으로 조회에 지정된 자료를 검색하기 위해 두 가지 방식을 사용합니다. 즉, 색인을 통하거나(키순 액세스 방식) 또는 표를 직접 통한(키순이 아닌 액세스 방식) 방식입니다. 이 액세스 방식은 여러 방법으로 결합되어 자료를 검색할 수 있습니다. 자료 액세스 방식은 표 스캔, 색인 또는 이들의 조합이거나 코드화 벡터 색인이 될 수 있습니다.

표 스캔

표 스캔 또는 도달순은 조회에 지정된 자료를 찾기 위해 자료가 표에 저장될 때의 행의 순서를 사용합니다. 표 스캔을 사용하여 표를 처리하는 것은 기존 시스템에서 순차적 또는 직접 파일을 처리하는 것과 유사합니다.

색인

색인 또는 키순 액세스 경로는 키 열의 목차에 따라 배열되는 표에 대한 액세스를 제공합니다. 키순은 행이 검색되는 순서입니다. 액세스 경로는 행이 표에 추가 또는 삭제될 때마다 또는 색인 열의 목차가 변경될 때마다 자동으로 유지보수됩니다. 가장 좋은 색인 예는 CREATE INDEX문을 사용하여 작성된 색인이나 CRTLF 명령을 사용하여 작성된 키순 논리 파일입니다.

색인에 적합한 열은 다음과 같습니다.

- 행 선택 술부에 자주 참조되는 열
- 그룹화 또는 순서화에 자주 참조되는 열
- 표를 결합하기 위해 사용되는 열(38 페이지의 『결합 최적화』)

코드화된 벡터 색인

코드화된 벡터 색인은 코드를 고유한 키 값에 할당한 후 이러한 값을 배열에 표시하여 데이터베이스 표에 액세스를 제공합니다. 배열의 요소는 표시되어야 하는 고유한 값의 수에 따라 1, 2 또는 4바이트의 길이가 될 수 있습니다. 크기가 작고 단순하여 코드화 벡터 색인은 보다 용이하게 병렬로 처리될 수 있는 고속 스캔에 제공됩니다.

CREATE ENCODED VECTOR INDEX문을 사용하여 코드화 벡터 색인을 작성합니다. 코드화 벡터 색인의 사용 및 유지보수에 대한 정보는 코드화 벡터 색인이란?을 참조하십시오. accelerating your queries with encoded vector indexes 에 대한 자세한 정보는 iSeries용 DB2 Universal Database 웹 페이지를 참조하십시오.

자료 액세스: 자료 액세스 방식

사용권 내부 코드 및 iSeries용 DB2 Universal Database는 액세스 방식에 대한 작업을 공유합니다. 사용권 내부 코드는 선택, 결합 기능, 해시 및 색인 작성을 포함하는 하위 레벨 처리를 실행합니다.

조회 최적화 처리는 각각의 조회에 대해 가장 효율적인 액세스 방식을 선택하여 이 정보를 액세스 계획에 보유합니다. 액세스 유형은 행 수, 예상되는 페이지 결함 수[1] 및 기타 범주에 따라 다릅니다.

이 책에서 나중에 설명하는 툴과 추가 정보를 사용하여 조회 Optimizer가 조회를 처리하는 방법에 영향을 줄 수 있습니다.

Optimizer는 자료를 검색하기 위해 다음과 같은 방식을 사용합니다. 이러한 방식에 대한 요약은 5 페이지의 『자료 액세스 방식: 요약』을 참조하십시오.

- 표 스캔 방식(자료 공간은 표에 자료가 들어 있는 내부 오브젝트)
- 병렬 표 사전페치 방식
- 색인 스캔 키 선택 방식
- 색인 스캔 키 위치지정 방식
- 병렬 표 또는 색인 사전로드
- Index-from-index 방식
- 색인 전용 액세스 방식
- 해시 방식
- 비트맵 처리 방식
- 정렬 액세스 방식

DB2 UDB 대칭형 멀티프로세싱 방식을 사용한 자료 액세스

DB2 UDB 대칭형 멀티프로세싱 피처는 병렬 처리가 들어 있는 자료 검색 추가 방식을 제공합니다.

대칭 다중처리(SMP)는 메모리와 디스크 자원을 공유하는 다중 프로세서(CPU 및 I/O 프로세서)가 하나의 최종 결과를 얻기 위해 동시에 작업하는 단일 서버에 갖추어진 병렬 형태입니다. 이 병렬 처리는 데이터베이스 관리자가 단일 조회에 대해 작업하는 두 개 이상(또는 모든)의 서버 프로세서를 동시에 가질 수 있음을 의미합니다. 서버에서 둘 이상의 프로세서에 프로세서 로드를 분배함으로써 다중 프로세서 서버의 SMP 피처로 CPU 바인드 조회의 성능이 향상될 수 있습니다.

일단 DB2 UDB 대칭형 멀티프로세싱 피처가 서버에 설치되면, 다음과 같은 방식을 Optimizer에 사용할 수 있습니다.

- 병렬 표 스캔 방식
- 병렬 색인 스캔 키 선택 방식

1. 프로그램이 주 기억장치에 없는 4K바이트의 페이지를 참조할 때 발생하는 인터럽트

- 병렬 색인 스캔 키 위치지정 방식
- 병렬 색인 전용 액세스 방식
- 병렬 해싱 방식
- 병렬 비트맵 처리 방식

추가 고려 사항:

다음 주제는 액세스 방식에 대한 추가 백그라운드 정보를 제공합니다.

- 7 페이지의『순서화 조회 결과』
- 7 페이지의『조회에 대해 병렬 처리』
- 8 페이지의『자동으로 자료 확산』

자료 액세스 방식: 요약

다음 표는 이 서적에서 설명하는 자료 관리 방식을 요약한 것입니다.

표 1. 자료 액세스 방식 요약

액세스 방식	선택 처리	양호한 경우	양호하지 않은 경우	선택 시기	장점
8 페이지의『표 스캔 액세스 방식』	모든 행 읽기. 자료 공간의 자료에 선택 범주 적용.	대략 > 20% 행 선택	대략 < 20% 행 선택	순서화, 그룹화 또는 결합을 하지 않으며, 대략 > 20% 행 선택	사전페치를 통해 페이지 I/O를 최소화함
11 페이지의『병렬 표 사전페치 액세스 방식』	자료가 병렬 스트림의 보조 기억장치에서 검색되었음. 모든 행 읽기. 자료 공간의 자료에 선택 범주 적용.	대략 > 20% 행 선택 1. 적당한 활동 메모리를 사용할 수 있음. 2. 그렇지 않으면 조회가 I/O 바인드됨 3. 자료가 복수 디스크 장치에 유포됨.	대략 < 20% 행 선택 조회가 CPU 바인드됨	순서화, 그룹화 또는 결합을 하지 않으며, 대략 > 20% 행이 선택되고, 서버 작업이 I/O 병렬의 장점을 갖도록 구성되었음.	병렬 표 사전페치를 통해 페이지 I/O 대기 시간을 최소화함.
12 페이지의『병렬 표 스캔 방식』	병렬 타스크에서 자료를 읽고 선택.	대략 > 10% 행 선택, 대형 표. 1. 적당한 활동 메모리를 사용할 수 있음. 2. 자료가 복수 디스크 장치에 유포됨. 3. DB2 UDB 대칭형 멀티프로세싱이 설치됨. 4. 다중 프로세서 서버	대략 < 10% 행 선택. 조회는 단일 프로세서 서버에서 I/O 바인드됨.	1. DB2 UDB 대칭형 멀티프로세싱이 설치됨. 2. 다중 프로세서 서버에서 CPU 바인드되거나 실행 중.	다중 프로세서에서 특히 유효한 성능임

표 1. 자료 액세스 방식 요약 (계속)

액세스 방식	선택 처리	양호한 경우	양호하지 않은 경우	선택 시기	장점
14 페이지의 『색인 스캔 키 선택 액세스 방식』	색인에 선택 범주 적용	순서화, 그룹화 및 결합	다수의 행이 선택됨.	색인이 요구되며 색인 스캔 키 위치지정 방식을 사용할 수 없음	자료 공간은 색인 스캔 키 선택 범주에 대응하는 행에 대해서만 액세스됨
15 페이지의 『병렬 색인 스캔 키 선택 액세스 방식(DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용 가능)』	병렬 타스크의 색인에 선택 범주 적용	색인의 크기가 자료 공간보다 훨씬 작음. DB2 UDB 대칭형 멀티프로세싱이 설치되어야 함	다수의 행이 선택됨.	결과의 순서화가 필요하지 않은 경우	병렬 타스크가 I/O를 수행하므로 I/O는 겹침. 다중 프로세서 서버를 충분히 활용할 수 있음.
16 페이지의 『색인 스캔 키 위치지정 액세스 방식』	색인 항목 범위에 선택 범주 적용. 일반적으로 사용되는 옵션.	대략 < 20% 행 선택	대략 > 20% 행 선택	선택 열이 가장 왼쪽 키와 일치하고, 대략 < 20% 행 선택	색인 및 자료 공간이 선택 기준에 일치하는 행에 대해서만 액세스됨.
21 페이지의 『병렬 색인 스캔 키 위치지정 액세스 방식 (DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용 가능)』	병렬 타스크로 색인 항목 범위에 선택 범주 적용	대략 < 20% 행 선택 DB2 UDB 대칭형 멀티프로세싱이 설치되어야 함	다수의 행이 선택됨.	1. 결과의 순서화가 필요하지 않은 경우 2. 선택 열이 가장 왼쪽 키와 일치하고, 대략 < 20% 행 선택	1. 색인 및 자료 공간이 선택 기준에 일치하는 행에 대해서만 액세스됨. 2. 병렬 타스크가 실행하므로 I/O가 겹침. 3. 다중 프로세서 서버를 충분히 활용할 수 있음
25 페이지의 『Index-from-index 액세스 방식』	영구 색인에서 키 행 위치지정. 선택된 색인 항목에 대해 임시 색인 작성.	순서화, 그룹화 및 결합.	대략 > 20% 행 선택	기존 색인이 순서화를 만족시키지 못하지만, 선택을 만족시키고 대략 < 20% 행 선택.	색인 및 자료 공간이 선택 기준에 일치하는 행에 대해서만 액세스됨.
31 페이지의 『정렬 액세스 방식』	표 스캔 처리 또는 색인 스캔 키 위치지정을 사용한 자료 읽기 순서화	대략 > 20% 행 선택 또는 대형 행 결과 세트 선택.	대략 < 20% 행 선택 또는 소형 행 결과 세트 선택.	지정된 순서화에는 순서화를 만족시키는 색인이 없거나 대형 결과 세트가 예상됨.	표 스캔과 이 표에서 색인 스캔 키 위치지정을 참조.
23 페이지의 『색인 전용 액세스 방식』	다른 색인 액세스 방식과 조합하여 수행.	조회에서 사용된 모든 열은 키 열로서 존재함.	대략 < 20% 행 선택 또는 소형 행 결과 세트 선택.	조회에서 사용된 모든 열은 키 열로서 존재함.	자료 공간에 대해 I/O가 감소.
24 페이지의 『병렬 표 또는 색인 기준 사전로드 액세스 방식』	색인 또는 표 자료가 임의 액세스를 피하기 위해 병렬로 로드됨.	그렇지 않으면 과도한 임의 활동이 오브젝트에 반하여 발생하고 활동 메모리가 전체 오브젝트를 보유하는 데 사용됨.	활동 메모리가 이미 과도하게 할당됨.	그렇지 않으면 과도한 임의 활동이 조회 처리로 발생하며 활동 메모리가 전체 오브젝트를 보유하는 데 사용됨.	임의의 I/O를 피하여 I/O 바인드 조회를 향상시킴.

표 1. 자료 액세스 방식 요약 (계속)

액세스 방식	선택 처리	양호한 경우	양호하지 않은 경우	선택 시기	장점
26 페이지의 『해시 액세스 방식』(병렬 또는 비 병렬)	공통의 상호 연관된 자료(공통된 값을 가짐)가 포함된 행	장기 실행 그룹화 및 결합 조회	단기 실행 조회	지정된 결합 또는 그룹화	색인 방식과 비교할 때 임의 I/O를 감소시킴. DB2 UDB 대칭형 멀티프로세싱이 설치되면, SMP 병렬을 사용할 수 있음.
27 페이지의 『비트맵 처리 방식』	키 위치/색인 스캔 키 선택이 비트맵 작성에 사용됨. 비트맵은 표의 행을 누르지 않도록 사용됨.	선택이 색인에 적용될 수 있으며, 대략 > 5% 또는 대략 < 25% 행이 선택되거나 OR 연산자가 하나의 색인만 사용하지 못하도록 하는 선택과 관련됨.	대략 > 25% 행 선택.	색인이 선택 범주에 대응함.	자료 공간에 대한 페이지 I/O를 줄임. 표 당 복수 색인이 허용됨.

순서화 조회 결과

결과의 특정 순서화를 보증하려면 ORDER BY절(또는 OPNQRYF KEYFLD 매개변수)을 지정해야 합니다. 병렬 액세스 방식을 사용할 수 없었을 때 데이터베이스 관리자는 순차적인 방식으로 표 행(및 키순)을 처리했습니다. 이렇게 하면, 순서화가 원래 조회 요구에 포함되지 않았더라도 결과의 순서를 어느정도 예측할 수 있게 됩니다(일반적으로 행은 자료 공간에 저장되었던 순서로 검색되었습니다). 병렬 방식은 표 행의 블록 및 키 값을 동시에 처리하므로 검색된 결과의 순서화가 더욱 임의적이고 예측할 수 없게 됩니다.

ORDER BY절은 결과의 특정 순서를 보증할 수 있는 유일한 방법입니다. 그러나, 순서화 요구는 결과 정렬로 인해 CPU 이용율 및 응답 시간이 모두 증가될 수 있으므로 꼭 필요한 경우에만 지정해야 합니다.

조회에 대해 병렬 처리

조회를 제출하거나 어플리케이션을 코드화할 때 조회에 대해 병렬 처리를 사용할 수 있어야 합니다. Optimizer가 자동으로 병렬을 선택된 액세스 방식으로 사용하지는 않습니다.

조회 Optimizer가 사용하는 병렬의 정도를 제어하기 위해 시스템 값 QQRYDEGREE, 조회 옵션 파일 또는 조회 속성 변경(CHGQRYA) 명령에 대한 DEGREE 매개변수를 사용할 수 있습니다. 병렬 처리 제어 방법에 대한 정보는 108 페이지의 『조회에 대한 병렬 처리 제어』를 참조하십시오.

데이터베이스 시스템 타스크 세트는 서버 시작시 데이터베이스 관리자에 의해 작성됩니다. 데이터베이스 관리자는 타스크를 사용하여 다른 디스크 장치로부터 자료를 처리하고 검색합니다. 이러한 타스크는 다중 프로세서에서 동시에 실행될 수 있으므로, 조회 경과 시간이 감소될 수 있습니다. 타스크가 병렬 조회의 많은 I/O 및 CPU 처리를 수행하더라도 사용된 I/O 및 CPU 자원 계정은 어플리케이션 작업에 전송됩니다. 이런 유형의 어플리케이션에 대해 요약된 I/O 및 CPU 자원은 활동 작업에 대한 작업(WRKACTJOB) 명령으로 계속 정확하게 표시됩니다.

자동으로 자료 확산

iSeries용 DB2 Universal Database는 자료가 할당된 ASP(보조 기억장치 풀)에서 사용할 수 있는 디스크 장치에 자료를 자동으로 확산시킵니다. 이렇게 하면 사용자의 간섭없이 자료를 확산시킵니다. 자료가 확산되면 데이터베이스 관리자는 다른 디스크 장치에서 행의 블록을 병렬로 용이하게 처리할 수 있습니다.

iSeries용 DB2 Universal Database가 ASP 내에서 디스크 장치에 자료를 확산할지라도, 때로 자료 확장의 할당(자료의 연속 세트)이 고르지 않게 확산될 수도 있습니다. 이런 현상은 장치에서 공간 할당 균형이 맞지 않거나 새로운 장치가 ASP에 추가될 때 발생합니다. 자료 공간 할당은 표 저장, 삭제 및 복원으로 다시 확산될 수 있습니다.

표 스캔 액세스 방식

표에 있는 모든 행을 읽습니다. 선택 기준이 각각의 행에 대해 적용되며 범주에 대응하는 행만 호출 어플리케이션으로 리턴됩니다. 표의 행이 보증되지 않은 순서로 처리됩니다. 특정 순서로 처리하려면, ORDER BY절(또는 OPNQRYF KEYFLD 매개변수)을 지정해야 합니다.

표 스캔은 다음과 같은 이유로 효율적일 수 있습니다.

- 제공된 페이지의 모든 행이 처리되므로 페이지 I/O 조작 수를 최소화하며 일단 페이지가 주 기억장치에 있으면 페이지는 다시 검색되지 않습니다.
- 데이터베이스 관리자는 자료 공간으로부터 페이지 순서를 쉽게 예측하여 검색할 수 있습니다. 이러한 이유로, 데이터베이스 관리자는 페이지의 비동기 I/O를 보조 기억장치로부터 주 기억장치로 스케줄할 수 있습니다. 이것은 일반적으로 **사전페치**라고 합니다. 이것은 주 기억장치에서 페이지를 사용할 수 있도록 데이터베이스 관리자가 자료에 액세스할 경우에 실행됩니다.

표 스캔 액세스 방식에 가장 효율적인 경우

이 선택 방식은 높은 비율이 행이 선택될 경우에 유용합니다. 높은 비율은 일반적으로 20% 이상입니다.

표 스캔 액세스 방식이 가장 비효율적인 경우

표 스캔 처리는 표에서 적은 비율의 행이 선택될 경우 비효율적입니다. 표의 모든 행이 점검되므로, 이것은 I/O 및 처리 장치 자원을 불필요하게 사용하게 됩니다.

표 스캔 액세스에 대한 고려사항

삭제된 행이 들어 있는 표에서 행을 선택할 때 표 스캔 처리는 영향을 받을 수 있습니다. 이것은 삭제 조작이 행이 삭제되었다고 표시만 하기 때문입니다. 표 스캔 처리의 경우, 데이터베이스 관리자는 삭제된 행을 선택한 적이 없을지라도 모든 삭제된 행을 읽습니다. 실제 파일 멤버 재구성(RGZPFM) CL 명령을 사용하여 삭제된 행을 제거하여야 합니다. 실제 파일에 REUSEDLT(*YES)를 지정하여, 삭제된 행 공간을 재사용할 수도 있습니다. 모든 SQL 표는 REUSEDLT(*YES)를 사용하여 작성됩니다.

PRTSQLINF 명령 메세지

자료 공간 선택 방식을 사용하는 SQL 프로그램에서 조회를 설명하기 위해 PRTSQLINEF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4010  Table scan access for table 1.
```

표 스캔 액세스 방식에 대한 선택 알고리즘

사용권 내부 코드는 표 스캔이 처리될 때, 선택에 대한 두 개의 알고리즘, 즉, 파생 열 선택 및 자료 공간만 선택 중 하나를 사용할 수 있습니다. 자료 공간만 선택에는 자료 공간 루핑 및 자료 공간만 필터링의 두 가지 형식이 있습니다. 자료 공간만 필터링은 파생된 조작에 앞서 레코드를 제거하기 위한 또다른 단계지만, 자료 공간 루핑은 대량의 레코드 세트를 효과적으로 처리합니다.

모든 액세스 방식은 자료 공간 필터링을 사용하지만, 자료 공간 루핑은 표 스캔이 높은 비율의 레코드를 처리할 때만 사용됩니다.

다음 의사 코드는 *파생 열 선택 알고리즘*을 설명합니다.

```
DO UNTIL END OF TABLE

1. Address the next (or first) row

2. Map all column values to an internal buffer, performing all derived
   operations.

3. Evaluate the selection criteria to a TRUE or FALSE value using
   the column values as they were copied to internal buffer.

4. IF the selection is TRUE
   THEN
     Copy the values from the internal buffer into the
      user's answer buffer.
   ELSE
     No operation
   END
```

*표 스캔 선택 알고리즘*은 다음과 같습니다.

```
DO UNTIL END OF TABLE

1. Calculate a search limit. This limit is usually the number of
   rows which are already in active memory, or have already
   had an I/O request done to be loaded into memory.

2. DO UNTIL (search limit reached
   or row selection criteria is TRUE)
   a. Address the next (or first) row

   b. Evaluate any selection criteria which does not
      require a derived value directly for the dataspace
      row.

   END
```

```
3. IF the selection is true
   THEN
   a. Map all column values to an internal buffer, performing all
      derived operations.

   b. Copy the values from the internal buffer into the
      user's answer buffer.
   ELSE
     No operation
   END
```

표 스캔 선택 알고리즘은 다음 두 가지 이유로 파생 열보다 나은 성능을 제공합니다.

- 자료 이동 및 계산은 선택된 행에 대해서만 실행됩니다.
- 표 스캔 선택 알고리즘의 2단계에서 루프가 실행 가능한 코드 버스트로 생성됩니다. 낮은 비율의 행이 실제로 선택되면, iSeries용 DB2 Universal Database는 행이 발견될 때까지 이 작은 프로그램을 실행합니다.

조회 코드화 지침

표 스캔 액세스 방식의 표 스캔 선택 알고리즘을 사용하는 조회에 대해 어떤 조치도 필요하지 않습니다. 모든 조회 인터페이스가 이러한 향상을 활용할 수 있습니다. 그러나, 다음의 지침은 선택 술부가 자료 공간 선택으로 구현될 수 있는지 여부를 판별합니다.

- Optimizer는 항상 모든 선택 항목에 대한 피연산자가 호환가능한 지 확인합니다. 따라서, 사용자는 조회를 처리하기 전에 피연산자가 호환가능한 지 확인하여 조회를 향상시킬 수 있습니다.
- 술부의 피연산자는 어떤 종류의 파생 값, 함수, 서브스트링, 연결 또는 숫자 표현식도 될 수 없습니다.
- 선택 술부의 양쪽 피연산자 둘 다 숫자 열인 경우, 두 열은 모두 같은 유형, 스케일 및 정밀도를 가져야 하며, 그렇지 않은 경우 하나의 연산자가 파생 값에 맵핑되어야 합니다. 예를 들어, DECIMAL(3,1)은 다른 DECIMAL(3,1) 열에 대해서만 비교되어야 합니다.
- 선택 술부의 한 피연산자가 숫자 열이고 다른 피연산자는 상수 또는 호스트 변수이면, 그 유형은 동일해야 하며, 상수 또는 호스트 변수의 정밀도와 스케일은 열의 정밀도와 스케일과 같거나 작아야 합니다.
- 열의 팩 십진 또는 숫자 유형과 관련된 선택 술부는 표가 SQL CREATE TABLE문에 의해 작성된 경우에만 실행될 수 있습니다.
- 다양한 길이의 문자 열은 선택 술부에서 참조될 수 없습니다.
- 선택 술부의 한 피연산자가 문자 열이고 다른 피연산자는 상수 또는 호스트 변수일 경우 호스트 변수의 길이는 열의 길이보다 클 수 없습니다.
- 문자 열 자료의 비교는 CCSID 또는 키보드 시프트 변환을 요구하지 말아야 합니다.

어떤 경우에는 79-80%까지 표 스캔 선택에 대한 CPU 및 응답 시간이 크게 감소할 수 있으므로 파생 열 선택을 피하는 것이 중요할 수 있습니다. 자료 공간만 선택에서 유용한 조회는 실제로 표의 60% 미만으로 선택되는 조회입니다. 선택된 행의 비율이 낮을수록, 성능이 뛰어납니다.

병렬 표 사전페치 액세스 방식

또한 iSeries용 DB2 Universal Database는 *병렬 표 사전페치* 처리를 사용하여 장기 실행, I/O 바인드 표 스캔 조회에 필요한 처리 시간을 단축할 수 있습니다.

이 방식은 표 스캔 방식과 동일한 특성을 갖지만 I/O 처리가 병렬로 실행됩니다. 이는 자료를 사전페치하기 위해 표에 대해 복수 입력 스트림을 시작함으로써 이루어집니다.

병렬 표 사전페치 액세스 방식에 가장 효율적인 경우

다음 경우에 이 방식이 가장 효율적입니다.

- 자료가 복수 디스크 장치에 유포됨.
- Optimizer에 의해 조회가 I/O 바인드될 것으로 판별됨.
- 모든 입력 스트림으로부터 수집된 자료를 보유할 수 있는 주 기억장치의 여유 공간이 있음.

iSeries용 DB2 Universal Database 자료 확산

앞에서 언급한 대로, iSeries용 DB2 Universal Database는 데이터베이스 관리자가 표 자료를 병렬로 사전페치할 수 있도록 하고, 사용자 간섭 없이 디스크 장치에 자료를 확산합니다. 데이터베이스 관리자는 타스크를 사용하여 다른 디스크 장치로부터 자료를 검색합니다. 보통 전체 확장(자료의 연속 세트)을 위해 요구합니다. 전체 확장을 하면 디스크 장치가 자료에 순조롭게 순차 액세스를 사용할 수 있으므로 성능을 향상시킵니다. 이러한 최적화로 인해, 병렬 사전페치는 SETOBJACC CL 명령보다 더 신속하게 활동 메모리로 자료를 사전로드할 수 있습니다.

iSeries용 DB2 Universal Database가 ASP 내에서 디스크 장치에 자료를 확산할지라도, 때로 자료 공간 확장의 할당이 균등하게 확산되지 않을 수 있습니다. 이런 현상은 장치에서 공간 할당 균형이 맞지 않거나 새로운 장치가 ASP에 추가될 때 발생합니다. 자료 공간 할당은 표 저장, 삭제 및 복원으로 재확산될 수 있습니다.

조회 Optimizer가 이 방식을 사용하는 조회를 선택하는 방법

조회 Optimizer는 이 유형을 구현할 수 있는 후보자 조회를 선택합니다. Optimizer는 조회 처리에 요구된 CPU 시간을 추정하고 입력 처리에 필요한 추정 시간을 비교하여 후보를 선택합니다. 추정한 입력 처리 시간이 CPU 시간을 초과할 때, 조회 Optimizer는 조회가 병렬 I/O으로 이행될 수 있음을 표시합니다.

DB2 UDB 대칭형 멀티프로세싱이 설치된 경우, 조회 Optimizer는 대개 DB2 UDB 대칭형 멀티프로세싱 병렬 방식을 선호합니다.

요구사항 처리

병렬 표 사전페치를 사용하려면 입력 및 출력 병렬 처리가 시스템 값 QQRYDEGREE, 조회 옵션 파일 또는 조회 속성 변경(CHGQRYA) 명령에 대한 DEGREE 매개변수에 의해 작동 가능하게 되어야 합니다. 병렬 처리 제어 방법에 대한 정보는 108 페이지의 『조회에 대한 병렬 처리 제어』를 참조하십시오. 병렬 표 사전페치를 사용하여 처리되고 있는 조회가 주 기억장치 및 디스크 I/O 자원을 적극 사용하므로, 병렬 표 사전페치를 사용하는 조회 수는 제한되고 제어되어야 합니다. 병렬 사전페치는 여러 개의 디스크 암(arm)을 사용하지만,

제공된 조회에 대해서는 여러 CPU를 거의 사용하지 않습니다. 병렬 사전페치 I/O는 I/O 자원을 집중적으로 사용합니다. 과다 할당된 I/O 서브 시스템이 있는 서버에 병렬 사전페치 조회를 허용하면 과다 할당 문제가 더 심각해질 수 있습니다.

활동 메모리를 더욱 효율적으로 사용하게 하므로, *CALC 페이징 옵션을 사용하여 공유 기억장치 풀에서 작업을 실행해야 합니다. iSeries용 DB2 Universal Database는 자동화 시스템 튜너를 사용하여 이 처리에 사용할 수 있는 메모리 양을 판별합니다. 실행시, 사용권 내부 코드는 메모리 통계가 메모리 자원을 과다 할당하지 않음을 나타내는 경우에만 병렬 표 사전페치 사용을 허용합니다. 페이징 옵션에 대한 자세한 정보는 작업 관리 주제의 자동 시스템 조정 섹션을 참조하십시오.

병렬 표 사전페치는 복수 입력 스트링을 사용하여 검색되고 있는 자료를 캐시하기에 충분한 사용 가능한 메모리를 요구합니다. 대형 표의 경우, 일반 확장 크기는 1MB입니다. 이는 두 개의 입력 스트림을 동시에 사용하려면 2MB 메모리가 사용 가능해야 함을 나타냅니다. 풀에 있는 사용 가능한 메모리 양을 증가시키면 더 많은 입력 스트림을 사용할 수 있습니다. 많은 메모리를 사용할 수 있는 경우, 표에 대한 전체 자료 공간은 조회가 열려 있을 때 활동 메모리로 로드될 수 있습니다.

PRTSQLINF 명령 메세지

병렬 표 사전페치 액세스 방식을 사용하는 SQL 프로그램에서 조회를 설명하기 위해 PRTSQLINEF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4023  Parallel dataspace prefetch used.
```

병렬 표 스캔 방식

iSeries용 DB2 Universal Database는 DB2 UDB 대칭형 멀티프로세싱 피처가 설치되어 있을 때 이 병렬 액세스 방식을 사용하여 장기 실행 표 스캔 조회에 필요한 처리 시간을 단축시킬 수 있습니다. 병렬 표 스캔 방식은 병렬 표 사전페치 액세스 방식처럼 I/O 처리 시간을 감소시킵니다. 또한, 프로세서가 두 개 이상인 서버에서 실행되는 경우, 이 방식은 표 스캔 처리를 동시에 복수 프로세서에서 실행할 수 있는 타스크로 분할하여 조회 경과 시간을 감소시킬 수 있습니다. 모든 선택 및 열 처리는 타스크에서 실행됩니다. 어플리케이션의 작업은 타스크에 대한 작업 요구를 스케줄하고 결과를 어플리케이션에 리턴된 결과 버퍼로 병합합니다.

병렬 표 스캔 액세스 방식이 가장 효율적인 경우

다음 경우에 이 방식이 가장 효율적입니다.

- 자료가 복수 디스크 장치에 유포됨
- 서버에 사용할 수 있는 복수 프로세서가 있음
- 자료 버퍼 및 결과 버퍼를 보유할 수 있는 주 기억장치의 여유 공간이 있음
- OLAP 또는 일괄처리 환경의 대형 표에 사용되는 경우

조회 Optimizer가 이 방식을 사용하는 조회를 선택하는 방법

앞에서 언급한 대로, iSeries용 DB2 Universal Database는 데이터베이스 관리자가 표 자료를 병렬로 사전페치할 수 있도록 하고, 사용자 간섭 없이 디스크 장치에 자료를 확산합니다. 이 방법은 각 타스크가 저장된 자

료의 각 해당 부분에 집중할 수 있도록 합니다. 이 방법은 자료에 대한 액세스를 확보하고 조회의 해당 부분을 수행하는 데 다른 타스크와의 경합이 생기지 않습니다.

조회 Optimizer는 이 유형을 구현할 수 있는 후보 조회를 선택합니다. Optimizer는 조회 처리에 요구된 CPU 시간을 추정하고 입력 처리에 필요한 추정 시간을 비교하여 후보를 선택합니다. Optimizer는 사용되어야 할 연산되는 타스크 수를 기준으로 표 스캔에 대한 추정 경과 시간을 감소시킵니다. 서버의 프로세서 수, 작업 풀에서 사용할 수 있는 메모리 양 및 DEGREE 조회 속성의 현재 값을 기준으로 타스크 수를 연산합니다. 병렬 표 스캔이 가장 신속한 액세스 방식인 경우 선택됩니다.

요구사항 처리

병렬 표 스캔에서는 시스템 값 QQRYDEGREE, 조회 옵션 파일 또는 조회 속성 변경(CHGQRYA) 명령에 대한 DEGREE 매개변수를 사용하여 SMP 병렬 처리가 가능해야 합니다. 병렬 처리 제어 방법에 대한 정보는 108 페이지의 『조회에 대한 병렬 처리 제어』를 참조하십시오.

병렬 표 스캔은 다음이 필요한 조회에 대해 사용될 수 없습니다.

- *ALL 확약 제어 레벨의 스펙
- 내포된 루프 결합 구현. 38 페이지의 『내포된 루프 결합 구현』을 참조하십시오.
- 역방향 화면 이동. 예를 들어, 병렬 표 스캔은 어플리케이션이 마지막 행으로 위치이동 및 이전 행 검색을 시도하므로 ALWCPYDTA(*YES) 또는 ALWCPYDTA(*NO)를 지정하는 조회 파일 열기(OPNQRYF) 명령에 의해 정의된 조회에 대해 사용할 수 없습니다. 화면 이동할 수 없게 정의된 SQL 정의된 조회는 이 방식을 사용할 수 있습니다. 조회를 정의하는 데 어떤 인터페이스를 사용했건 정렬 또는 해시 조작과 같은 임시 결과를 작성하는 동안 병렬 표 스캔을 사용할 수 있습니다. OPNQRYF는 ALWCPYDTA 매개변수에 대해 *OPTIMIZE 매개변수 값을 지정하여 화면 이동할 수 없이 정의될 수 있으며 대부분의 병렬 액세스 방식을 사용할 수 있습니다.
- 커서 위치 복원. 예를 들면, 커서 위치가 SQL ROLLBACK HOLD문 또는 ROLLBACK CL 명령의 결과로 복원되어야 하는 조회. *NONE을 제외한 확약 제어 레벨을 사용하는 SQL 어플리케이션은 사전컴파일러 매개변수 ALWBLK에 대한 값으로 *ALLREAD를 지정하여 이 방식을 사용할 수 있습니다.
- 갱신 또는 삭제 기능

활동 메모리를 더욱 효율적으로 사용할 수 있으므로 *CALC 페이징 옵션이 있는 공유 기억장치 풀에서 작업을 실행해야 합니다. 페이징 옵션에 대한 자세한 정보는 iSeries Information Center에 있는 작업 관리 주제의 자동 시스템 조정 섹션을 참조하십시오.

병렬 표 스캔은 활동 메모리가 검색 중인 자료를 버퍼하고 각 타스크에 대한 결과 버퍼를 분리할 것을 요구합니다. 각각의 타스크에 필요한 일반적 총량은 약 2MB입니다. 예를 들어, 네 개의 병렬 표 스캔 타스크를 동시에 사용하려면 약 8MB의 메모리가 필요합니다. 풀에 있는 사용 가능한 메모리 양을 증가시키면 더 많은 입력 스트림을 사용할 수 있습니다. 다양한 길이의 대형 문자 열이 있는 표에 액세스하는 조회 또는 표의 실제 행 길이 보다 더 큰 결과 값을 생성하는 조회는 각각의 타스크에 대해더 많은 메모리를 요구할 수 있습니다.

여러 행 잠금이 상충되거나 자료 맵핑 오류가 발생하면 병렬 표 스캔의 성능이 심각하게 제한을 받을 수 있습니다.

색인 스캔 키 선택 액세스 방식

이 액세스 방식은 색인을 요구합니다. 전체 색인을 읽고 색인의 키 열을 참조하는 선택 범주가 색인에 대해 적용됩니다. 이 방식의 장점은 자료 공간이 색인에 대해 적용된 선택 범주를 만족하는 행을 검색하는 데만 액세스되는 것입니다. 색인 스캔 키 선택 방식을 통해 실행되지 않은 추가 선택은 자료 공간 레벨에서 실행됩니다.

다음과 같은 이유로 탐색 조건이 다수의 행에 적용된다면 색인 스캔 키 선택 액세스 방식은 비용이 많이 들 수 있습니다.

- 전체 색인이 처리됨
- 색인으로부터 선택된 모든 키에 대해 자료 공간에 대한 임의 I/O가 발생함

조회 Optimizer가 이 방식을 사용하는 조회를 선택하는 방법

일반적으로, Optimizer는 탐색 조건이 다수의 행에 적용될 때 표 스캔 처리를 사용하기 위해 선택됩니다. Optimizer는 20% 미만의 키가 선택되거나 조작에 색인을 사용해야 하는 경우 색인 스캔 키 선택 방식을 선택합니다. 색인 사용을 강제할 수 있는 조작은 다음과 같습니다.

- 순서화
- 그룹화
- 결합

이러한 경우, Optimizer는 기존 색인을 사용하기보다 임시 색인 작성을 선택합니다. Optimizer가 임시 색인을 작성할 때, 1차 다이얼에 대해 64K 페이지 크기를 사용하며 2차 다이얼에 대해 8K 페이지 크기를 사용합니다. SQL CREATE INDEX문을 사용하여 작성된 색인은 64K 페이지 크기를 사용합니다. CRTLF 명령을 사용하여 작성된 색인 또는 V4R5MO 이전에 작성된 SQL 색인의 경우 색인 크기는 일반적으로 16K입니다.

또한 Optimizer는 임시 색인을 작성하는 동안 가능한 많은 선택을 처리합니다. Optimizer에 의해 작성된 거의 모든 임시 색인이 선택/생략 또는 희소 색인입니다. 마지막으로 Optimizer는 색인을 작성할 때 복수 병렬 타스크를 사용할 수 있습니다. 페이지 크기 차이가 상대적으로 페이지 수가 작은 스와핑으로부터 해당하는 성능 향상 및 병렬 타스크를 사용하여 색인을 작성하는 기능은 색인 작성 총 비용을 극복할 수 있기에 충분합니다. 자료 공간 선택은 임시 색인을 작성하는 데 사용됩니다.

색인 스캔 키 선택 액세스 방식이 사용되는 경우 조회가 (색인이 요구된) 조회 수행 순서를 지정했기 때문에 다음과 같은 매개변수를 사용하여 순서화를 조회 정렬과 함께 실행하여 조회 성능이 향상될 수 있습니다.

- SQL의 경우 사전컴파일러 매개변수 조합은 다음과 같습니다.
  - ALWCPYDTA(*OPTIMIZE), ALWBLK(*ALLREAD) 및 COMMIT(*CHG 또는 *CS)
  - ALWCPYDTA(*OPTIMIZE) 및 COMMIT(*NONE)
- OPNQRYF의 경우 매개변수는 다음과 같습니다.

- ALWCPYDTA(*OPTIMIZE) 및 COMMIT(*NO)
- ALWCPYDTA(*OPTIMIZE) 및 COMMIT(*YES) 및 확약 제어 레벨은 *NONE, *CHG 또는 *CS의 확약 레벨로 시작됩니다.

조회가 선택/생략 색인을 지정하여 Optimizer가 임시 색인을 작성하도록 결정할 때, 선택/생략 색인의 모든 선택이 조회로부터 적용 가능한 선택 다음의 임시 색인으로 삽입됩니다.

병렬 색인 스캔 키 선택 액세스 방식(DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용 가능)

병렬 색인 스캔 키 선택 액세스 방식의 경우, 가능한 키 값은 논리적으로 파티션됩니다. 각 파티션은 색인 스캔 키 선택 액세스 방식에서와 마찬가지로 별도 타스크로 처리됩니다. 동시에 처리된 파티션 수는 조회 Optimizer에 의해 판별됩니다. 키가 순서대로 처리되지 않으므로, 이 방식은 색인이 순서화에 사용 중이면 Optimizer가 사용할 수 없습니다. 색인에서 대부분의 기존 키가 들어 있는 키 파티션은 다른 파티션의 처리가 완료될 때 세밀히 분할됩니다.

병렬 색인 스캔 키 선택 액세스 방식이 가장 효율적인 경우

다음 예는 Optimizer가 색인 스캔 키 선택 방식을 선택할 수 있는 조회를 보여줍니다.

```
    CREATE INDEX X1 ON EMPLOYEE(LASTNAME,WORKDEPT)

     DECLARE BROWSE2 CURSOR FOR
  SELECT * FROM EMPLOYEE
      WHERE WORKDEPT = 'E01'
           OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''E01''')
```

Optimizer가 4 정도를 사용하여 병렬로 이 조회를 실행하기를 선택한다면, 다음은 동시에 처리되는 논리 키 파티션이 될 수 있습니다.

```
LASTNAME values            LASTNAME values
leading character          leading character
partition start            partition end
   'A'                          'F'
   'G'                          'L'
   'M'                          'S'
   'T'                          'Z'
```

첫 번째 및 두 번째 파티션에 상대적으로 키가 적게 있는 경우, 해당 키 값 처리는 세 번째 및 네 번째 파티션보다 더 빨리 완료됩니다. 처음 두 파티션이 완료된 후, 마지막 두 파티션에 남아있는 키 값은 더 세밀히 분할됩니다. 다음은 첫 번째 및 두 번째 파티션이 완료되고 분할이 발생한 후 처리되는 네 개의 파티션을 보여줍니다.

```
LASTNAME values            LASTNAME values
leading character          leading character
partition start            partition end
```

```
'O'                     'P'
'Q'                     'S'
'V'                     'W'
'X'                     'Z'
```

요구사항 처리

병렬 색인 스캔 키 선택은 다음이 필요한 조회에 대해 사용될 수 없습니다.

- *ALL 확약 제어 레벨의 스펙
- 내포된 루프 결합 구현. 38 페이지의 『내포된 루프 결합 구현』을 참조하십시오.
- 역방향 화면 이동. 예를 들어, 병렬 색인 스캔 키 선택은 어플리케이션이 마지막 행으로 위치 이동 및 이전 행의 검색을 시도하므로 ALWCPYDTA(*YES) 또는 ALWCPYDTA(*NO)를 지정하는 조회 파일 열기(OPNQRYF) 명령에 의해 정의된 조회에 대해 사용할 수 없습니다. OPNQRYF는 ALWCPYDTA 매개변수에 대해 *OPTIMIZE 매개변수 값을 지정하여 화면 이동할 수 없이 정의될 수 있으며 대부분의 병렬 액세스 방식을 사용할 수 있습니다. 화면 이동할 수 없게 정의된 SQL 정의된 조회는 이 방식을 사용할 수 있습니다. 조회를 정의하는 데 어떤 인터페이스를 사용했건 정렬 또는 해시 조작과 같은 임시 결과를 작성하는 동안 병렬 색인 스캔 키 선택을 사용할 수 있습니다.
- 커서 위치 복원(예를 들면, 커서 위치가 SQL ROLLBACK HOLD문 또는 ROLLBACK CL 명령의 결과로 복원되어야 하는 조회). *NONE을 제외한 확약 제어 레벨을 사용하는 SQL 어플리케이션은 사전컴파일러 매개변수 ALWBLK에 대한 값으로 *ALLREAD를 지정하여 이 방식을 사용할 수 있습니다.
- 갱신 또는 삭제 기능

활동 메모리를 더욱 효율적으로 사용할 수 있으므로 *CALC 페이징 옵션이 있는 공유 풀에서 작업을 실행해야 합니다. 페이징 옵션에 대한 자세한 정보는 iSeries Information Center에 있는 작업 관리 주제의 자동 시스템 조정 섹션을 참조하십시오.

병렬 색인 스캔 키 선택에서는 시스템 값 QQRYDEGREE, 조회 옵션 파일 또는 조회 속성 변경(CHGQRYA) 명령에 대한 DEGREE 매개변수를 사용하여 SMP 병렬 처리가 가능해야 합니다. 병렬 처리 제어 방법에 대한 정보는 108 페이지의 『조회에 대한 병렬 처리 제어』를 참조하십시오.

색인 스캔 키 위치지정 액세스 방식

이 액세스 방식은 색인 스캔 키 선택 액세스 방식과 아주 유사합니다. 두 방식 모두 키순 색인이 필요합니다. 색인 스캔 키 선택 액세스 방식에서, 처리는 색인의 시작에서 시작되어 끝까지 계속되며 모든 키가 페이지 내에 있습니다. 색인 스캔 키 위치지정 액세스 방식에서, 선택은 일부 또는 모든 선택 범주에 대응하는 키의 범위에서 직접 색인에 대한 것입니다. 이 범위에 해당되는 키만 페이지되고, 나머지 색인 선택은 색인 스캔 키 선택 방식에 의해 수행됩니다. 색인 스캔 키 위치지정 또는 색인 스캔 키 선택을 통해 실행되지 않는 모든 선택은 자료 공간 레벨에서 실행됩니다. 색인 스캔 키 위치지정은 색인에 있는 키의 서브세트만 검색하므로, 색인 스캔 키 위치지정 방식의 성능이 색인 스캔 키 선택 방식의 성능보다 우수합니다.

색인 스캔 키 위치지정 액세스 방식이 가장 효율적인 경우

색인 스캔 키 위치지정 방식은 낮은 비율의 행이 선택되는 경우(약 20% 미만) 가장 효율적입니다. 대략 행이 20% 이상 선택되는 경우 Optimizer는 일반적으로 다음과 같이 선택합니다.

- 표 스캔 처리 사용(색인이 필요하지 않는 경우)
- 색인 스캔 키 선택 사용(색인이 필요한 경우)
- 조회 정렬 루틴 사용(조건이 적용되는 경우)

조회 Optimizer가 이 방식을 사용하는 조회를 선택하는 방법

색인을 요구하지 않는 조회의 경우(순서화, 그룹화 또는 결합 연산이 없음), Optimizer는 색인 스캔 키 위치지정에 사용할 기존 색인을 찾습니다. 기존 색인을 찾을 수 없는 경우, Optimizer는 색인을 작성하는 것보다 표 스캔 처리를 사용하는 것이 더 빠르므로 자료에 키순 액세스 사용을 중지한 후 색인 스캔 키 위치지정을 실행합니다.

다음 예는 Optimizer가 색인 스캔 위치지정 방식을 선택할 수 있는 조회를 보여줍니다.

```
CREATE INDEX X1 ON EMPLOYEE(WORKDEPT)

  DECLARE BROWSE2 CURSOR FOR
SELECT * FROM EMPLOYEE
   WHERE WORKDEPT = 'E01'
        OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
  QRYSLT('WORKDEPT *EQ ''E01''')
```

이 예에서, 데이터베이스 지원은 *X1*을 사용하여 'E01'에 해당하는 *WORKDEPT* 값을 갖는 첫 번째 색인 항목으로 위치지정합니다. 'E01'에 해당하는 각 키의 경우, 자료 공간[2] 행을 선택합니다. 색인 스캔 키 선택이 E01 이상의 키 값으로 이동할 때 조회가 종료됩니다.

이 예의 경우 처리된 모든 색인 항목과 검색된 행이 선택 범주에 부합됨에 유의하십시오. 색인 키 위치지정을 통해 실행되지 않는 추가 선택이 추가되는 경우, (예: 복수 열에 대해 색인의 첫 번째 키 열과 대응하지 않는 선택 열과 같은) Optimizer는 가능한 한 많이 추가 선택을 실행할 수 있도록 색인 스캔 키 선택을 사용합니다. 모든 나머지 선택은 자료 공간 레벨에서 실행됩니다.

SQL 프로그램에서 이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4008  Index X1 used for table 1.
 SQL4011  Key row positioning used on table 1.
```

색인 스캔 키 위치지정 액세스 방식에는 추가 처리 기능이 있습니다. 그러한 기능 중 하나는 여러 값에 대해 범위 선택을 실행하는 것입니다. 예를 들면 다음과 같습니다.

---

2. 임의의 액세스는 키가 자료 공간에서와 동일한 행 순서로 있지 않으므로 발생

```
CREATE INDEX X1 EMPLOYEE(WORKDEPT)

     DECLARE BROWSE2 CURSOR FOR
  SELECT * FROM EMPLOYEE
      WHERE WORKDEPT BETWEEN 'E01' AND 'E11'
           OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ %RANGE(''E01'' ''E11'')')
```

위 예에서, 데이터베이스 지원은 값 'E01'과 같은 첫 번째 색인 항목으로 위치지정하며 행은 'E11'에 대한 마지막 색인 항목이 처리될 때까지 처리됩니다.

PRTSQLINF 명령 메세지

SQL 프로그램에서 이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
   SQL4008  Index X1 used for table 1.
    SQL4011  Key row positioning used on table 1.
```

복수 범위 색인 스캔 키 위치지정

복수 범위 색인 스캔 키 위치지정이라는 이 액세스 방식의 확장을 사용할 수 있습니다. 이 방식을 사용하여 복수 열에 걸친 색인의 첫 번째 키 열에 대해 복수 범위 값의 행을 선택할 수 있습니다.

```
    CREATE INDEX X1 ON EMPLOYEE(WORKDEPT)

     DECLARE BROWSE2 CURSOR FOR
   SELECT * FROM EMPLOYEE
      WHERE WORKDEPT BETWEEN 'E01' AND 'E11'
        OR WORKDEPT BETWEEN 'A00' AND 'B01'
           OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ %RANGE(''E01'' ''E11'')
   *OR WORKDEPT *EQ %RANGE(''A00'' ''B01'')')
```

위 예에서, 위치지정 및 처리 기법은 각 범위의 값에 대해 한 번씩, 두 번 사용됩니다.

SQL 프로그램에서 이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
   SQL4008  Index X1 used for table 1.
    SQL4011  Key row positioning used on table 1.
```

이제까지 색인 스캔 키 위치지정의 모든 예에서 색인 키 중 하나 즉, 가장 왼쪽 키만 사용되었습니다. 색인 스캔 키 위치지정은 두 개 이상의 키도 처리합니다.

```
CREATE INDEX X2
   ON EMPLOYEE(WORKDEPT,LASTNAME,FIRSTNME)

 DECLARE BROWSE2 CURSOR FOR
SELECT * FROM EMPLOYEE
    WHERE  WORKDEPT = 'D11'
     AND  FIRSTNME = 'DAVID'
        OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''D11''
   *AND FIRSTNME *EQ ''DAVID''')
```

두 개의 선택 키(WORKDEPT 및 FIRSTNME)가 연속하지 않으므로 이 예에 대한 복수 키 위치지정 지원은 없습니다. 따라서, 선택의 WORKDEPT = 'D11' 부분만 색인에 대해 적용할 수 있습니다(단일 키 색인 스캔 키 위치지정). 이것이 허용될 수 있지만, 행 처리가 'D11'의 첫 번째 키로 시작한 후, 색인 스캔 키 선택을 사용하여 WORKDEPT 키 값이 'D11'인 모든 9개 항목에 대해 FIRSTNME = 'DAVID'를 처리함을 의미합니다.

위의 조회 예는 다음 색인 X3을 작성하여, 복수 키 사용을 실행하여 색인 스킨 키 위치지정을 수행합니다.

```
CREATE INDEX X3
   ON EMPLOYEE(WORKDEPT, FIRSTNME, LASTNAME)
```

복수 키 색인 스캔 키 위치지정 지원은 색인 스캔 키 위치지정으로 선택의 양쪽 부분에 모두 적용할 수 있습니다. 이렇게 하면 성능이 크게 향상됩니다. 시작 값은 두 개의 선택 값을 'D11DAVID'로 연결하여 작성되며 선택은 가장 왼쪽의 두 키가 해당 값을 갖는 색인 항목으로 위치지정됩니다.

SQL 프로그램에서 이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4008  Index X3 used for table 1.
SQL4011  Key row positioning used on table 1.
```

다음 예에서는 더욱 유익한 복수 색인 스캔 키 위치지정의 사용을 보여줍니다.

```
CREATE INDEX X3 ON EMPLOYEE(WORKDEPT,FIRSTNME)

 DECLARE BROWSE2 CURSOR FOR
SELECT * FROM EMPLOYEE
    WHERE  WORKDEPT = 'D11'
    AND  FIRSTNME IN  ('DAVID','BRUCE','WILLIAM')
        OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''D11''
   *AND FIRSTNME *EQ %VALUES(''DAVID'' ''BRUCE''
   ''WILLIAM'')')
```

조회 Optimizer는 WHERE 절을 분석하고 절을 상응하는 양식으로 리빌드합니다.

```
    DECLARE BROWSE2 CURSOR FOR
   SELECT * FROM EMPLOYEE
      WHERE   (WORKDEPT = 'D11' AND FIRSTNME = 'DAVID')
           OR (WORKDEPT = 'D11' AND FIRSTNME = 'BRUCE')
           OR (WORKDEPT = 'D11' AND FIRSTNME = 'WILLIAM')
           OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('(WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ
   ''DAVID'')
  *OR (WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ ''BRUCE'')
   *OR (WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ ''WILLIAM'')')
```

리빌드된 조회 양식에서, WORKDEPT 및 FIRSTNME의 연결 값에 대해 실제로 있습니다.

```
      Index X3 Start value       Index X3 Stop value
   'D11DAVID'                 'D11DAVID'
   'D11BRUCE'                 'D11BRUCE'
   'D11WILLIAM'               'D11WILLIAM'
```

색인 스캔 키 위치지정은 선택된 키의 수를 단지 세 개로 줄이면서 각 범위에 대해 실행됩니다. 모든 선택은 색인 스캔 키 위치지정을 통해 이루어질 수 있습니다.

다음 예에서 이 범위 분석은 더욱 복잡해집니다.

```
        DECLARE BROWSE2 CURSOR FOR
   SELECT * FROM EMPLOYEE
      WHERE (WORKDEPT = 'D11'
                  AND FIRSTNME IN ('DAVID','BRUCE','WILLIAM'))
            OR (WORKDEPT = 'E11'
                    AND FIRSTNME IN ('PHILIP','MAUDE'))
            OR (FIRSTNME BETWEEN 'CHRISTINE' AND 'DELORES'
                    AND WORKDEPT IN ('A00','C01'))
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('(WORKDEPT *EQ ''D11''
   *AND FIRSTNME *EQ %VALUES(''DAVID'' ''BRUCE'' ''WILLIAM''))
   *OR (WORKDEPT *EQ ''E11''
   *AND FIRSTNME *EQ %VALUES(''PHILIP'' ''MAUDE''))
   *OR (FIRSTNME *EQ %RANGE(''CHRISTINE'' ''DELORES'')
   *AND WORKDEPT *EQ %VALUES(''A00'' ''C01''))')
```

조회 Optimizer는 WHERE 절을 분석하고 절을 상응하는 양식으로 리빌드합니다.

```
    DECLARE BROWSE2 CURSOR FOR
   SELECT * FROM EMPLOYEE
      WHERE   (WORKDEPT = 'D11' AND FIRSTNME = 'DAVID')
           OR (WORKDEPT = 'D11' AND FIRSTNME = 'BRUCE')
           OR (WORKDEPT = 'D11' AND FIRSTNME = 'WILLIAM')
           OR (WORKDEPT = 'E11' AND FIRSTNME = 'PHILIP')
           OR (WORKDEPT = 'E11' AND FIRSTNME = 'MAUDE')
           OR (WORKDEPT = 'A00' AND FIRSTNME BETWEEN
```

```
                                        'CHRISTINE' AND 'DELORES')
        OR (WORKDEPT = 'C01' AND FIRSTNME BETWEEN
                                        'CHRISTINE' AND 'DELORES')
    OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('(WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ
   ''DAVID'')
  *OR (WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ ''BRUCE'')
   *OR (WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ
   ''WILLIAM'')
  *OR (WORKDEPT *EQ ''E11'' *AND FIRSTNME *EQ ''PHILIP'')
  *OR (WORKDEPT *EQ ''E11'' *AND FIRSTNME *EQ ''MAUDE'')
  *OR (WORKDEPT *EQ ''A00'' *AND
   FIRSTNME *EQ %RANGE(''CHRISTINE'' ''DELORES''))
  *OR (WORKDEPT *EQ ''C01'' *AND
  FIRSTNME *EQ %RANGE(''CHRISTINE'' ''DELORES''))')
```

조회에서, WORKDEPT 및 FIRSTNME의 연결 값에 대해 실제 7개 분리 범위의 키 값이 있습니다.

```
      Index X3 Start value       Index X3 Stop value
   'D11DAVID'                 'D11DAVID'
   'D11BRUCE'                 'D11BRUCE'
   'D11WILLIAM'               'D11WILLIAM'
   'E11MAUDE'                 'E11MAUDE'
   'E11PHILIP'                'E11PHILIP'
   'A00CHRISTINE'             'A00DELORES'
   'C01CHRISTINE'             'C01DELORES'
```

색인 스캔 키 위치지정이 각 범위에 대해 실행됩니다. 범위 내에 키 값의 해당 행이 있는 경우에만 리턴됩니다. 모든 선택은 색인 스캔 키 위치지정을 통해 이루어질 수 있습니다. 이것은 이 조회의 성능을 상당히 향상시킵니다.

병렬 색인 스캔 키 위치지정 액세스 방식(DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용 가능)

병렬 색인 스캔 키 위치지정 액세스 방식을 사용할 때 기존 키 범위는 분리 데이터베이스 타스크에서 동시에 분리 타스크에 의해 처리됩니다. 동시 타스크 수는 Optimizer에 의해 제어됩니다. 조회는 사용 중인 최대 병렬 정도까지 조회의 키 범위 처리를 시작합니다. 해당 범위의 처리가 완료되면 리스트에 있는 다음 범위의 처리가 시작됩니다. 범위에 대한 처리가 완료되고 리스트에 처리할 범위가 더 이상 없을 때 병렬 색인 스캔 키 선택 방식에서 처럼 처리할 키가 남아 있는 범위가 분할됩니다. 데이터베이스 관리자는 분리 키 범위를 각각 처리하면서 사용 중인 모든 타스크를 활동 상태로 유지하려 합니다. 단일 값, 값 범위를 사용하면 복수 범위 색인 스캔 키 위치지정을 사용하건 범위는 동시에 더 세밀히 파티션 및 처리될 수 있습니다. 키가 순서대로 처리되지 않으므로, 이 방식은 색인이 순서화에 대해 사용 중이면 Optimizer에 의해 사용될 수 없습니다.

조회 Optimizer가 이 방식을 사용하는 방법

SQL문이 병렬 정도 4를 사용하여 실행 중인 경우 다음 예를 고려하십시오.

```
 DECLARE BROWSE2 CURSOR FOR
SELECT * FROM EMPLOYEE
   WHERE   (WORKDEPT = 'D11' AND FIRSTNME = 'DAVID')
        OR (WORKDEPT = 'D11' AND FIRSTNME = 'BRUCE')
        OR (WORKDEPT = 'D11' AND FIRSTNME = 'WILLIAM')
        OR (WORKDEPT = 'E11' AND FIRSTNME = 'PHILIP')
        OR (WORKDEPT = 'E11' AND FIRSTNME = 'MAUDE')
        OR (WORKDEPT = 'A00' AND FIRSTNME BETWEEN
                                       'CHRISTINE' AND 'DELORES')
        OR (WORKDEPT = 'C01' AND FIRSTNME BETWEEN
                                       'CHRISTINE' AND 'DELORES')
        OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
  QRYSLT('(WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ ''DAVID'')
  *OR (WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ ''BRUCE'')
  *OR (WORKDEPT *EQ ''D11'' *AND FIRSTNME *EQ ''WILLIAM'')
  *OR (WORKDEPT *EQ ''E11'' *AND FIRSTNME *EQ ''PHILIP'')
  *OR (WORKDEPT *EQ ''E11'' *AND FIRSTNME *EQ ''MAUDE'')
  *OR (WORKDEPT *EQ ''A00'' *AND
  FIRSTNME*EQ %RANGE(''CHRISTINE'' ''DELORES''))
  *OR (WORKDEPT *EQ ''C01'' *AND
  FIRSTNME *EQ %RANGE(''CHRISTINE'' ''DELORES''))')
```

데이터베이스 관리자가 시작하는 키 범위는 다음과 같습니다.

```
      Index X3 Start value       Index X3 Stop value
Range 1   'D11DAVID'                 'D11DAVID'
Range 2   'D11BRUCE'                 'D11BRUCE'
Range 3   'D11WILLIAM'               'D11WILLIAM'
Range 4   'E11MAUDE'                 'E11MAUDE'
Range 5   'E11PHILIP'                'E11PHILIP'
Range 6   'A00CHRISTINE'             'A00DELORES'
Range 7   'C01CHRISTINE'             'C01DELORES'
```

범위 1 - 4가 분리 타스크에서 동시에 처리됩니다. 네개 중 하나가 완료되면 바로 범위 5가 시작됩니다. 또 다른 범위가 완료되면 범위 6이 시작됩니다. 진행 중인 네개 범위 중 하나가 완료되고 리스트에 시작할 새 범위가 더 이상 없으면 다른 키 범위 중 하나에 남아있는 나머지 작업이 분할되어 각각의 절반이 별도로 처리됩니다.

요구사항 처리

병렬 색인 스캔 키 위치지정은 다음이 필요한 조회에 대해 사용될 수 없습니다.

- *ALL 확약 제어 레벨의 스펙
- 내포된 루프 결합 구현. 38 페이지의 『내포된 루프 결합 구현』을 참조하십시오.
- 역방향 화면 이동. 예를 들어, 병렬 색인 스캔 키 위치지정은 어플리케이션이 마지막 행으로 위치 이동 및 이전 행을 검색 시도하므로 ALWCPYDTA(*YES) 또는 ALWCPYDTA(*NO)를 지정하는 조회 파일 열기(OPNQRYF) 명령에 의해 정의된 조회에 대해 사용할 수 없습니다. 화면 이동할 수 없게 정의된 SQL 정의된 조회는 이 방식을 사용할 수 있습니다. 조회를 정의하는 데 어떤 인터페이스를 사용했건 정렬 또는 해시 조작과 같은 임시 결과를 작성하는 동안 병렬 색인 스캔 키 위치지정을 사용할 수 있습니다.

OPNQRYF는 ALWCPYDTA 매개변수에 대해 *OPTIMIZE 매개변수 값을 지정하여 화면 이동할 수 없이 정의될 수 있으며 대부분의 병렬 액세스 방식을 사용할 수 있습니다.

- 커서 위치 복원. 예를 들면, 커서 위치가 SQL ROLLBACK HOLD문 또는 ROLLBACK CL 명령의 결과로 복원되어야 하는 조회. *NONE을 제외한 확약 제어 레벨을 사용하는 SQL 어플리케이션은 사전컴파일러 매개변수 ALWBLK에 대한 값으로 *ALLREAD를 지정하여 이 방식을 사용할 수 있습니다.
- 갱신 또는 삭제 기능

활동 메모리를 더욱 효율적으로 사용할 수 있으므로 *CALC 페이징 옵션이 있는 공유 풀에서 작업을 실행해야 합니다. 페이징 옵션에 대한 자세한 정보는 iSeries Information Center에 있는 작업 관리 주제의 자동 시스템 조정 섹션을 참조하십시오.

병렬 색인 스캔 키 선택에서는 시스템 값 QQRYDEGREE, 조회 옵션 파일 PARALLEL_DEGREE 옵션 또는 조회 속성 변경(CHGQRYA) 명령에 대한 DEGREE 매개변수를 사용하여 SMP 병렬 처리가 가능해야 합니다. 병렬 처리 제어 방법에 대한 정보는 108 페이지의 『조회에 대한 병렬 처리 제어』를 참조하십시오.

색인 전용 액세스 방식

색인 전용 액세스 방식은 방식에 대한 병렬 옵션을 포함하여 색인 스캔 키 선택 또는 색인 스캔 키 위치지정 액세스 방식과 함께 사용될 수 있습니다(병렬 옵션은 DB2 UDB 대칭형 멀티프로세싱 피처가 설치되어 있을 때만 사용할 수 있습니다). 선택에 대한 처리는 이러한 방식에 대해서 이미 기술한 내용과 동일합니다.

그러나 모든 자료는 자료 공간에 임의 I/O를 실행하지 않고 색인에서 추출됩니다. 색인 항목은 조회에 지정되었을 수 있는 결과 맵핑 또는 파생에 대한 입력으로 사용됩니다.

색인 전용 방식이 가장 효율적인 경우

Optimizer는 다음과 같은 경우에 이 방식을 선택합니다.

- 조회 내에서 참조된 모든 열이 영구 색인 내에 있거나 Optimizer가 작성하기로 결정한 임시 색인의 키 열 내에 있습니다.
- 자료 값은 색인에서 추출될 수 있어야 하며 읽을 수 있는 형식으로 사용자에게 리턴되어야 합니다. 즉, 조회 열과 일치하는 모든 키 열은 다음을 갖지 않습니다.
  - 지정된 절대값
  - 지정된 정렬 순서 또는 대체 배열 순서
  - 지정된 존 또는 숫자 강제
- 조회는 왼쪽 외부 결합 또는 예외 결합을 사용하지 않습니다.
- 비SQL 사용자의 경우, 가변 길이 또는 널(null) 허용 열은 키 피드백을 요구할 수 없습니다.

다음 예는 Optimizer가 색인 전용 방식을 실행하기 위해 선택할 수 있는 조회를 보여줍니다 .

```
CREATE INDEX X2
  ON EMPLOYEE(WORKDEPT,LASTNAME,FIRSTNME)
```

```
DECLARE BROWSE2 CURSOR FOR
 SELECT FIRSTNME FROM EMPLOYEE
  WHERE  WORKDEPT = 'D11'
      OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''D11''')
```

이 예에서, 데이터베이스 관리자는 X2를 사용하여 WORKDEPT='D11'에 대한 색인 항목으로 위치지정한 후 해당 항목에서 열 FIRSTNME에 대한 값을 추출합니다.

색인 키 열은 실행될 색인 전용 액세스에 대해 색인의 가장 왼쪽 키에 인접하지 않아도 된다는 점에 유의하십시오. 색인의 키 열은 색인 전용 조회에 대한 자료를 제공하는 데 사용될 있습니다. 색인은 선택을 완료한 후 데이터베이스 관리자가 조회 처리를 완료할 수 있도록 단지 자료에 대한 소스로 사용됩니다.

주: 색인 전용 액세스는 특정 표에 대해 이행되어 일부 또는 모든 결합 조회 표에 대해 색인 전용 액세스를 실행할 수 있습니다.

PRTSQLINF 명령 메세지

SQL 프로그램에서 이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4008  Index X2 used for table 1.
 SQL4011  Key row positioning used on table 1.
SQL4022  Index only access used on table 1.
```

병렬 표 또는 색인 기준 사전로드 액세스 방식

색인 스캔 키 선택을 사용하여 구현되는 일부 조회는 자료 공간을 액세스하기 위해 많은 임의의 I/O를 요구할 수 있습니다. 이 때문에, 자료 공간에 있는 자료가 높은 비율로 참조됩니다. iSeries용 DB2 Universal Database는 조회 처리가 시작될 때 색인 또는 표 기준 사전로드를 초기화하여 이 임의의 I/O를 피합니다. 전체 표 또는 색인은 병렬 표 사전페치에 대해 실행된 것처럼 병렬로 활동 메모리에 로드됩니다. 이것은 전체 오브젝트를 로드하기에 충분한 메모리가 풀에 있어야 합니다.

표 또는 색인이 메모리로 로드된 후, 자료에 대한 임의의 액세스는 더 이상의 I/O 없이 수행됩니다. iSeries용 DB2 Universal Database 비용 기준 조회 Optimizer는 I/O 병렬 처리를 사용할 수 있는 경우 표 또는 색인 사전로드로부터 유리한 조회 및 오브젝트를 인식합니다. 병렬 처리 제어 방법에 대한 정보는 108 페이지의 『조회에 대한 병렬 처리 제어』를 참조하십시오. DB2 UDB 대칭형 멀티프로세싱이 설치된 경우, 조회 Optimizer는 대개 DB2 UDB 대칭형 멀티프로세싱 병렬 방식을 선호합니다.

병렬 사전로드 방식은 다른 자료 액세스 방식과 함께 사용될 수 있습니다. 사전로드는 조회가 열려 있고 사전로드가 완료되기 전에 제어가 어플리케이션으로 리턴될 때 시작됩니다. 어플리케이션은 사전로드를 인식하지 못하고 다른 데이터베이스 액세스 방식을 사용하여 행 페치를 계속합니다.

Index-from-index 액세스 방식

데이터베이스 관리자는 자료 공간의 모든 행을 읽지 않고 기존 색인으로부터 임시 색인을 작성할 수 있습니다. 일반적으로, 이 선택 방식은 가장 효율적인 방식 중 하나입니다. 작성되는 임시 색인에는 선택 술부와 부합되는 행에 대한 항목만 들어 있습니다. 이것은 선택/생략 논리 파일에 의해 작성된 색인(흔히 희소 색인이라고 함)과 유사합니다.

index-from-index 액세스 방식이 가장 효율적인 경우

Optimizer는 다음과 같은 경우에 이 방식을 선택합니다.

- 조회는 그룹화, 순서화 또는 결합 처리를 사용하므로 색인을 요구합니다.
- 선택 열을 가장 왼쪽 키로 갖는 영구 색인이 있으며, 가장 왼쪽 키는 매우 선택적입니다(즉, 색인 스캔 키 위치지정이 사용될 수 있습니다).
- 선택 열이 순서화, 그룹화 또는 결합 열과 동일하지 않습니다.

조회 Optimizer가 이 방식을 사용하는 방법

index-from-index 액세스 방식을 사용하려면 데이터베이스 관리자는 다음을 실행합니다.

1. 조회 선택 범주가 있는 영구 색인에 색인 스캔 키 위치지정을 사용합니다.
2. 선택된 항목을 사용하여 새로운 임시 색인에 색인 항목을 빌드합니다.
3. 임시 색인의 키 열은 그룹화, 순서화 또는 결합 열과 일치합니다.

결과는 선택 기준에 일치하는 행에 대한 필수 키순으로 된 항목을 포함하는 색인입니다(그룹화, 순서화, 결합).

일반 index-from-index 액세스 방식 예는 다음과 같습니다.

```
    CREATE INDEX X1 ON EMPLOYEE(WORKDEPT)

     DECLARE BROWSE2 CURSOR FOR
   SELECT * FROM EMPLOYEE
       WHERE  WORKDEPT = 'D11'
    ORDER BY LASTNAME
          OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''D11''')
   KEYFLD((LASTNAME))
```

이 예의 경우, 임시 선택/생략 색인은 1차 키 열 LASTNAME를 사용하여 작성됩니다. *WORKDEPT = 'D11'* 이 있는 행에 대한 색인 항목만 들어 있으며, 대략 20% 미만의 항목이 선택된다고 가정됩니다.

PRTSQLINF 명령 메세지

SQL 프로그램에서 이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4012  Index created from index X1 for table 1.
 SQL4011  Key row positioning used on table 1.
```

index-from-index 방식의 대안

index-from-index 액세스 방식을 사용하지 않고 조회 정렬 루틴을 사용 할 수 있습니다. 자세한 정보는 31 페이지의『정렬 액세스 방식』을 참조하십시오.

이런 결정은 검색될 행 수를 기준으로 합니다.

해시 액세스 방식

해시 액세스 방식은 그룹화 또는 상관 방식으로 자료를 처리해야 하는 해당 조회(그룹화 및 결합)에 대해 대체 방식을 제공합니다. 색인은 자료를 정렬 및 그룹화하는 데 사용되며 조회 그룹화 및 결합 조작을 수행하는 데 효율적인 경우도 있습니다. 그러나 Optimizer가 해당 조회에 대해 임시 색인을 작성해야 하는 경우, 요구된 조회가 실행될 수 있기 전에 색인을 작성할 때 초과 프로세서 시간 및 자원을 추가로 사용합니다.

해시 액세스 방식이 가장 효율적인 경우

해시 액세스 방식은 색인을 보충하거나 이의 대안으로 사용될 수 있습니다. 각각의 선택된 행에 대해, 행에 지정된 그룹화 또는 결합 값이 해시 기능을 통해 실행됩니다. 계산된 해시 값은 해시 표의 특정 파티션을 탐색하는 데 사용됩니다. 해시 표는 임시 작업 표와 비슷하나 지정된 조회을 기준으로 논리적으로 파티션된 다른 구조를 갖습니다. 행의 소스 값이 표에 없는 경우, 데이터베이스 표에서 이 소스 값을 처음 발견할 때 표시를 합니다. 새로운 해시 표 항목은 이 첫 번째 값으로 초기화되며 추가 처리는 요구된 조작(예:그룹화 또는 결합)을 기준으로 실행됩니다. 행의 소스 값이 표에서 발견되면, 이 값에 대한 해시 표 항목이 검색되며 추가 조회 처리가 (그룹화 또는 결합과 같은) 요구된 연산을 기준으로 실행됩니다. 해시 방식만이 동일한 값을 상관(또는 그룹화)할 수 있으며 해시 표 행은 오름차순 또는 내림차순으로 확실히 정렬되지 않습니다.

이 방식이 사용될 수 있는 경우

해시 방식은 데이터베이스 관리자에 의해 작성된 해시 표가 선택된 행의 임시 사본이므로 임시 결과가 필요한 경우를 제외하고 ALWCPYDTA(*OPTIMIZE) 옵션이 지정된 경우에만 사용될 수 있습니다.

이 방식의 작업 방법

해시 알고리즘을 사용하여 데이터베이스 관리자는 소스 자료가 임의적이고 분배된 경우 균형잡힌 해시 표를 작성할 수 있습니다. 해시 표는 요구된 조회 조작 및 처리중인 소스 값 수를 기준으로 파티션됩니다. 해시 알고리즘은 새로운 해시 표 항목이 해시 표 파티션을 통해 고르게 분배되도록 합니다. 해시 표의 다른 파티션에 있는 스캔이 동일한 수의 항목을 처리하도록 하려면 이러한 균형된 분배가 필요합니다. 하나의 해시 표 파티션에 다수의 해시 표 항목이 들어 있는 경우, 해당 파티션에 대한 스캔은 해시 표에 있는 다수의 항목을 점검하여야 합니다. 이것은 그다지 효율적이 아닙니다.

해시 방식은 일반적으로 표의 행을 순차적으로 처리하므로, 데이터베이스 관리자는 조회 처리에 필요한 데이터베이스 표로부터 메모리 페이지 순서를 쉽게 예측할 수 있습니다. 이것은 표 스캔 액세스 방식의 장점과 비슷

합니다. 예측할 수 있으면 데이터베이스 관리자는 주 기억장치로 표 페이지의 비동기 I/O를 스케줄할 수 있습니다(사전페치라고도 함). 사전페치를 하면 해시 방식에 대해 효율적인 I/O 조작이 가능하므로 조회 성능이 향상됩니다.

이와 달리, 키순 액세스 방식으로 처리하는 조회는 점검된 모든 키 값에 대해 데이터베이스 표에 대한 임의 I/O를 합니다. I/O 조작은 색인에 있는 자료의 키순이 데이터베이스 표에 있는 행의 실제 순서와 일치하지 않으므로 임의적입니다. 임의 I/O는 I/O 및 프로세서 장치 자원을 불필요하게 사용하게 되므로 조회 성능을 감소시킬 수 있습니다.

또한 색인은 키순으로 표 행을 처리하기 위해 해시 방식에 의해 사용될 수 있습니다. 색인은 해시 방식이 처리해야 할 표 행 수를 상당히 줄일 수 있습니다. 이것은 색인과 연관된 임의 I/O 비용을 줄일 수 있습니다.

조회가 열리기 전에 해시 표가 작성되어 채워집니다. 일단 해시 표가 지정된 데이터베이스 행으로 완전히 채워지면, 데이터베이스 관리자는 해시 표를 사용하여 조회 결과 리턴을 시작합니다. 요구된 조회 조작에 따라 결과 해시 표 행에서 추가 처리가 필요할 수 있습니다.

표 행의 블록이 자동 유포되므로, 해시 액세스 방식 또한 여러 그룹의 행이 동시에 해시될 수 있도록 병렬로 실행됩니다. 이것은 데이터베이스 표에 있는 모든 행을 해시하는 데 걸리는 시간을 단축합니다.

DB2 UDB 대칭형 멀티프로세싱 피처가 설치되어 있으면, 해싱 방식이 병렬로 수행될 수 있습니다.

비트맵 처리 방식

이름이 나타내듯이, 이 방식은 자료 공간에 액세스하는 동안 사용되는 비트맵을 생성합니다. 비트맵 처리 방법은 다음 목적으로 사용됩니다.

- 색인을 키 위치와 결합하여 사용하거나 색인 스캔 키 선택 방식을 사용할 때, 자료 공간에서 발생하는 임의 I/O를 최소화.
- 건너뛰 순차 방식을 사용할 때 자료 공간에 있는 많은 자료 섹션을 건너뛰기하여 I/O를 보다 효율적으로 스케줄.
- 특정 자료 공간을 액세스하는 데 여러 색인 사용 허용.

비트맵 처리 방식 작업 방법

이 방법에서, Optimizer는 하나 이상의 색인을 선택하여 자료 공간으로부터 행 선택을 돕는 데 사용됩니다. 임시 비트맵은 각각의 색인에 대해 하나를 할당(및 초기화)됩니다. 각각의 비트맵에는 기초가 되는 자료 공간의 각 행에 대해 하나의 비트가 들어 있습니다. 각 색인에 대해, 비트맵 초기화시 선택 기준을 적용하기 위해 이 색인 스캔 키 행 위치지정 및 색인 스캔 키 행 선택 방식이 사용됩니다.

선택된 각 색인 항목에 대해, 이 행과 연관된 비트는 1로 설정됩니다(켜짐). 자료 공간은 액세스되지 않습니다. 색인 처리가 완료되면, 비트맵에는 기초가 되는 자료 공간으로부터 선택될 행에 대한 정보가 들어 있습니다. 이 프로세스는 각각의 색인에 대해 반복됩니다. 둘 이상의 색인이 사용되는 경우, 임시 비트맵이 하나의 결과 비트맵을 얻기 위해 논리적으로 AND 및 OR로 변환합니다. 일단 결과 비트맵이 작성되면, 조회에 의해 선택

되지 않는 한 자료 공간으로부터 행에서 맵핑을 방지하는 데 사용됩니다. 자료 공간으로부터의 행 선택을 스케줄하거나, 액세스되고 있는 기초 자료 공간보다 먼저 다른 필터링 레벨을 제공하는 데 사용됩니다.

비트맵 생성에 사용된 색인은 선택된 행에 액세스하는 데 실제로 사용되지 않습니다. 그러므로, 이 색인을 **3차 색인**이라고 합니다. 반대로, 마지막 행에 액세스하는 데 사용된 색인은 **1차 색인**이라고 합니다. 1차 색인은 순서화, 그룹화, 결합 및 비트맵이 사용되지 않은 경우의 선택에 대해 사용됩니다.

이 방식이 사용되는 경우

비트맵은 Optimizer가 1차 액세스 방식을 통해 조회 처리를 시작하기 전에 항상 사전처리됩니다. 비트맵 처리 방식은 1차 액세스 방식 표 스캔, 색인 스캔 키 행 위치지정 또는 색인 스캔 키 행 선택과 함께 사용됩니다. 비트맵 처리는 병렬 표 사전페치 및 병렬 표/색인 사전로드처럼, 자료 공간에서 행을 실제로 선택하지 않고 단지 1차 방식을 지원합니다.

비트맵이 표 스캔 방식과 함께 사용되는 경우, 비트맵은 건너뜀 순차 처리를 시작합니다. 표 스캔(및 병렬 표 스캔)은 비트맵을 사용하여, 선택된 행이 없는 페이지(즉, 비트맵의 비트가 1로 설정되지 않음)를 ″건너 뛰기″ 합니다. 여기에는 몇가지 장점이 있습니다.

- 선택되지 않은 행을 처리하는 데 CPU 처리가 사용되지 않습니다.
- I/O가 최소화되며 메모리가 전체 자료 공간의 목차로 꽉 채워지지 않습니다.

예: 비트맵 처리 방식이 표 스캔 방식과 함께 사용되는 경우

다음 예는 조회 Optimizer가 표 스캔과 함께 비트맵 처리 방식을 선택하는 조회를 보여줍니다.

```
CREATE INDEX IX1 ON EMPLOYEE (WORKDEPT)
CREATE INDEX IX2 ON EMPLOYEE (SALARY)

DECLARE C1 CURSOR FOR
  SELECT * FROM EMPLOYEE
  WHERE WORKDEPT = 'E01' OR SALARY>50000
  OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''E01'' *OR SALARY > 50000')
```

이 예에서, 색인 IX1과 IX2 모두가 사용됩니다. 데이터베이스 관리자는 우선 색인 IX1에 대해 적용하는 선택 WORKDEPT = ‘E01’을 적용(색인 키 위치지정 사용) 결과로부터 비트맵을 생성합니다. 그런 다음 데이터베이스 관리자는 (또한 색인 스캔 키 위치지정을 사용하여) 색인 IX2에 대해 선택 SALARY>50000을 적용(색인 스캔 키 위치지정 재사용)한 결과로부터 비트맵을 생성합니다.

그런 다음, 데이터베이스 관리자는 각 비트맵을 모두 논리적으로 OR하여 한개의 결과 비트맵으로 결합합니다. 마지막으로, 표 스캔이 시작합니다. 표 스캔은 자료 공간 행에 걸쳐 비트맵을 사용하여 선택된 행만 검색하고 비트맵을 사용하여 나머지는 건너뜁니다. 이 방법은 많은 자료 부분을 건너 뛰기함으로써 성능을 향상시킵니다.

이 예는 또한 비트맵 처리로 제공된 추가 기능을 보여줍니다(AND 선택을 위한 색인은 이미 사용할 수 있었으나 현재 비트맵 처리는 두 개 이상의 색인을 허용). 비트맵 처리를 사용하면, OR이 주요 기본 부울 연산자가 되는 선택에 대해 복수 색인을 사용할 수 있습니다.

이 조회를 설명하기 위해 사용될 때 PRTSQLINF CL 명령에 의해 작성되는 메세지는 다음과 같습니다.

```
SQL4010  Table scan for table 1.
SQL4032  Index IX1 used for bitmap processing of table 1.
SQL4032  Index IX2 used for bitmap processing of table 1.
CPI4329  Arrival sequence access was used for file EMPLOYEE.
CPI4388  2 access path(s) used for bitmap processing of file EMPLOYEE.
```

예: 비트맵 처리가 색인 스캔 키 위치지정 액세스 방식과 함께 사용되는 경우

비트맵이 색인 스캔 키 행 위치지정 또는 색인 스캔 키 행 선택 방식과 함께 사용되는 경우, 비트맵(3차 색인으로부터 생성된)이 1차 색인 액세스를 지원하는 데 사용되고 있음을 의미합니다. 다음 예는 비트맵 처리가 1차 색인에 대해 색인 스캔 키 위치지정과 함께 사용되는 조회를 보여줍니다.

```
CREATE INDEX PIX ON EMPLOYEE (LASTNAME)
CREATE INDEX TIX1 ON EMPLOYEE (WORKDEPT)
CREATE INDEX TIX2 ON EMPLOYEE (SALARY)

DECLARE C1 CURSOR FOR
  SELECT * FROM EMPLOYEE
  WHERE WORKDEPT = 'E01' OR SALARY>50000
  ORDER BY LASTNAME
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE))
   QRYSLT('WORKDEPT *EQ ''E01'' *OR SALARY > 50000')
        KEYFLD(LASTNAME)
```

이 예에서, 색인 TIX1과 TIX2가 비트맵 처리에 사용됩니다. 데이터베이스 관리자는 우선 색인 TIX1에 대해 선택 WORKDEPT = ‘E01’을 적용(색인 스캔키 위치지정 사용)한 결과로부터 비트맵을 생성합니다. 색인 TIX2에 대해 선택 SALARY>50000을 적용(색인 키 스캔 키 위치지정 재사용)한 결과로부터 비트맵을 생성합니다.

그런 다음, 데이터베이스 관리자는 OR 논리를 사용하여 두 개의 비트맵을 하나의 비트맵으로 결합합니다. 색인 스캔 키 선택 방식이 (1차) 색인 PIX를 사용하여 시작됩니다. 색인 PIX의 각 항목에 대해, 비트맵을 검사합니다.항목이 비트맵에 의해 선택되면, 자료 공간 행이 검색되고 처리됩니다.

이 조회를 설명하기 위해 PRTSQLINF CL 명령에 의해 작성된 메세지는 다음과 같습니다.

```
SQL4008  Index PIX used for table 1.
SQL4032  Index TIX1 used for bitmap processing of table 1.
CPI4328  Access path of file PIX was used by query.
CPI4338 2 access path(s) used for bitmap processing of file EMPLOYEE.
```

예: 비트맵 처리가 결합 조회와 함께 사용되는 경우

결합 조회에 대해 비트맵 처리가 또한 사용될 수 있습니다. 비트맵 처리는 표 당 기준으로 하고 있으므로, 결합의 각 표는 비트맵 처리를 독립적으로 사용할 수 있거나 사용할 수 없습니다.

다음 예는 비트맵 처리가 결합 조회의 첫 번째 표가 아닌 두 번째 표에 대해 사용되는 조회를 보여줍니다.

```
CREATE INDEX EPIX ON EMPLOYEE(EMPNO)
CREATE INDEX TIX1 ON EMPLOYEE(WORKDEPT)
CREATE INDEX TIX2 ON EMPLOYEE(SALARY)
DECLARE C1 CURSOR FOR
SELECT * FROM PROJECT, EMPLOYEE
WHERE RESEMP=EMPNO AND
(WORKDEPT='E01' OR SALARY>50000)
```

OPNQRYF 명령 사용:

```
OPNQRYF FILE((PROJECT) (EMPLOYEE)) FORMAT(RESULTFILE)
        JFLD((1/RESPEMP 2/EMPNO))
        QRYSLT('2/WORKDEPT=''E01'' *OR 2/SALARY>50000')
```

이 예에서, Optimizer는 결합 순서는 표 EMPLOYEE에 대해 표 PROJECT임을 판별합니다. 표 스캔은 표 PROJECT에서 사용됩니다. 표 EMPLOYEE의 경우, 색인 EPIX는 결합(1차 색인)을 처리하는 데 사용됩니다. 색인 TIX1 및 TIX2는 비트맵 처리에 사용됩니다.

데이터베이스 관리자는 표 PROJECT의 첫 번째 행으로 위치지정합니다. 그런 다음, 색인 EPIX를 사용하여 결합을 실행합니다. 그 다음, 색인 TIX1에 대해 선택 WORKDEPT = ‘E01’을 적용(색인 스캔 키 위치지정 사용)한 결과로부터 비트맵을 생성합니다. 색인 TIX2에 대해 선택 SALARY>50000을 적용(색인 키 스캔 키 위치지정 재사용)한 결과로부터 비트맵을 생성합니다.

그 다음, 데이터베이스 관리자는 OR 논리를 사용하여 두 개의 비트맵을 하나의 비트맵으로 결합합니다. 마지막으로 EPIX가 현재 위치지정된 항목이 비트맵에 대해 검사됩니다. 항목은 비트맵에 의해 선택 또는 거부됩니다. 항목이 선택되면, 행은 기초가 되는 자료 공간으로부터 검색됩니다. 그 다음, 색인 EPIX가 다음 결합 행에 대해 탐색됩니다. 항목을 찾으면 비트맵에 대해 비교되어 선택 또는 거부됩니다. 비트맵은 한 번만(맨 처음 필요한 경우) 생성되고 그 이후에는 재사용될 뿐임에 유의하십시오.

작업 기록부에 입력되는 조회는 Optimizer 디버그 메세지는 다음과 같습니다.

```
CPI4327 File PROJECT processed in join position 1.
CPI4326 File EMPLOYEE processed in join position 2.
CPI4338 2 access path(s) used for bitmap processing of file EMPLOYEE.
```

비트맵 처리 및 복합 키 색인

비트맵 처리는 복합 키 색인(한 색인의 복수 키 열)을 갖는 데 연관된 일부 문제점을 다소 해결합니다.

예를 들어, 다음과 같은 SQL 조회가 제공됩니다.

```
DECLARE C1 CURSOR FOR
   SELECT * FROM EMPLOYEE
   WHERE WORKDEPT='D11' AND
   FIRSTNAME IN ('DAVID', 'BRUCE', 'WILLIAM')
```

또는 OPNQRYF 명령을 사용하는 다음의 동일한 조회가 제공됩니다.

```
OPNQRYF FILE((EMPLOYEE))
  QRYSLT('WORKDEPT=''D11'' *AND
  FIRSTNME = %VALUES(''DAVID'' ''BRUCE'' ''WILLIAM'')')
```

키(WORKDEPT, FIRSTNAME)를 갖는 색인이 이 조회에 부합하기 위해 사용되는 최상의 색인입니다. 그러나 두 개의 색인 즉, WORKDEPT 키를 갖는 색인과 FIRSTNME 키를 갖는 색인이 AND 결합된 결과 비트맵과 결과 검색에 사용된 표 스캔과 함께 비트맵 처리에 사용될 수 있습니다.

비트맵 처리 방식을 사용하여, 각각 하나의 키 열만을 갖는 여러 색인을 작성하여 Optimizer가 많은 조회에 일반 목적의 색인으로 사용할 수 있습니다. 표에 대해 실행된 모든 조회에 대한 최고의 복합 키를 판별과 관련된 문제를 피할 수 있습니다. 비트맵 처리는 복수 키 열 색인 사용과 비교할 때 사용이 보다 수월하지만 성능면에서 다소 떨어집니다.

주: 복합 키 색인을 사용하면 항상 최고의 성능을 달성한다는 점을 명심하십시오.

비트맵 처리에 대한 고려사항

비트맵 처리에 대한 일부 추가 사항은 다음과 같습니다.

- DB2 UDB 대칭형 멀티프로세싱 피처가 설치되어 있는 한, 비트맵 처리를 사용할 때마다 병렬 처리를 사용할 수 있습니다. 이 경우, 비트맵이 2차 색인에 대한 병렬 색인 스캔 키 위치지정 또는 병렬 색인 스캔 키 선택의 실행 결과로부터 빌드됩니다.
- 비트맵은 조회에 대한 최초 I/O 요구에서 사전처리됩니다. 그러므로 페치된 처음 행은 후속 행 보다 검색하는 데 시간이 더 오래 걸립니다.
- 비트맵은 본래 정적 선택을 포함하고 있는데, 이 선택은 ALWCPYDTA에 의해 제어됩니다. 비트맵은 일단 생성되면, 새로운 행 또는 수정된 행을 선택하지 않습니다. 그러므로, 비트맵 처리는 ALWCPYDTA (*OPTIMIZE)가 지정된 경우에만 사용됩니다.

  예를 들어, (QRYSLT('QUANTITY >5')를 지정하는 OPNQRYF문이 비트맵 처리를 사용하여 열려 첫 번째 행이 읽힌다고 가정하십시오. 표에 대한 갱신이 레코드가 선택되게 하지 않지만, 갱신 및 레코드가 선택되지 않게 한 갱신 전에 비트맵이 생성된 이 예를 반전시키면, Optimizer는 검색된 기초 자료 공간에 적용된 중복 선택 때문에 이 레코드를 확보하지 않도록 합니다. 별도의 데이터베이스 조작을 통해, QUANTITY가 4인 모든 행이 갱신되어 QUANTITY가 10이 됩니다. (OPNQRYF로부터 새로운 행 페치가 ID를 여는 동안) 비트맵이 이미 OPNQRYE 열린 ID에서 첫 번째 행 페치를 하는 동안 작성되었으므로 갱신된 열은 OPNQRYF 열린 ID를 통해 후속 페치에 대해 검색되지 않습니다.

  이에 대한 예외는 조회에 그룹화 또는 하나 이상의 총계 함수(예: SUM, COUNT, MIN, MAX)가 들어 있는 경우로, 정적 자료가 이미 작성되고 있는 경우입니다.
- 조회 Otimizer는 삽입, 갱신 또는 삭제가 허용되는 조회에 대해 비트맵 처리를 사용하지 않습니다. OPNQRYF의 경우, OPTION 매개변수를 *INP로 설정하고 SEQONLY 매개변수를 *YES로 설정해야 합니다. SEQONLY(*NO)로 대체해서는 안됩니다.

정렬 액세스 방식

정렬 액세스 방식은 순서화된 방식(ORDER BY)으로 자료를 처리해야 하는 해당 조회에 대해 대체 방식을 제공합니다. 색인은 자료를 정렬하는 데 사용될 수 있으며 순서화를 이행하는 데 어떤 경우에는 효율적입니다. 그러나, Optimizer가 해당 조회에 대해 임시 색인을 작성해야 하는 경우, 요구된 조회를 실행할 수 있기 전에 색인을 작성할 때 초과 프로세서 시간 및 자원을 추가로 사용합니다.

이 방식이 사용될 수 있는 경우

Optimizer는 다음과 같은 경우에 이 방식을 선택합니다.

- SQL에 대해 다음 사전컴파일 옵션 중 하나를 지정하는 경우(148 페이지의 『데이터베이스 성능에 영향을 미치는 ALWCPYDTA 매개변수』 참조)
  - ALWCPYDTA(*OPTIMIZE), ALWBLK(*ALLREAD) 및 COMMIT(*CHG 또는 *CS)
  - ALWCPYDTA(*OPTIMIZE) 및 COMMIT(*NONE)
- OPNQRYF에 대해 다음 옵션 중 하나를 지정하는 경우
  - ALWCPYDTA(*OPTIMIZE) 및 COMMIT(*NO)
  - ALWCPYDTA(*OPTIMIZE) 및 COMMIT(*YES) 및 확약 제어가 *NONE, *CHG 또는 *CS의 확약 레벨로 시작합니다.
- 임시 결과가 순서화 기능 이전에 요구되는 경우
- 키별로 순서화된 수가 120을 초과하거나 정렬 키의 결합 길이가 2000바이트를 초과하는 경우

이 방식의 작업 방법

정렬 알고리즘은 행을 정렬 공간으로 읽어 들인 다음 지정된 순서화 키를 기준으로 행을 정렬합니다. 그 다음에는 순서화된 정렬 공간에서 사용자에게로 행이 리턴됩니다.

제 3 장 iSeries용 DB2 UDB 조회 Optimizer: 개요

조회 Optimizer의 개요는 더욱 효율적으로 수행하고 서버 자원을 보다 효과적으로 사용하는 조회를 설계하기 위한 지침을 제공합니다. 이 개요는 조회 Optimizer를 통해 최적화되는 조회를 포함하여 SQL, OPNQRYF, APIs(QQQQRY), ODBC 및 Query/400 조회와 같은 인터페이스에 대해서도 설명합니다. 이 지침을 적용하든 안하든 조회 결과는 정확합니다.

주: 이 개요에서 다루는 정보는 복잡한 내용입니다. 개념을 보다 잘 이해하려면 이 장의 내용을 읽으면서 iSeries 서버에 대해 실험해 보는 것이 도움이 될 수 있습니다.

iSeries용 DB2 Universal Database가 조회를 처리하는 방법을 이해하고 있으면, 이 장에서 설명하는 지침이 성능에 미치는 영향을 보다 쉽게 이해할 수 있습니다. iSeries용 DB2 Universal Database가 조회를 처리하는 데는 다음 두 가지의 중요한 구성요소가 있습니다.

- 서버가 자료에 액세스하는 방법. 4 페이지의 『자료 액세스: 자료 액세스 방식』을 참조하십시오.

  이 방법은 디스크에서 자료를 검색하는 데 사용되는 알고리즘입니다. 액세스 방식에는 색인 사용 및 행 선택 기법이 포함됩니다. 또한, DB2 UDB 대칭형 멀티프로세싱 오퍼레이팅 시스템 피처와 함께 병렬 액세스 방식을 사용할 수도 있습니다.
- 조회 Optimizer. 『조회 Optimizer가 조회의 효율을 향상시키는 방법』을 참조하십시오.

  조회 Optimizer는 조회를 구현하는 데 사용할 수 있는 유효한 기법들을 식별하고 그 중 가장 효과적인 기법을 선택합니다.

조회 Optimizer가 조회의 효율을 향상시키는 방법

Optimizer는 다음과 같은 이유로 iSeries용 DB2 Universal Database의 중요한 구성요소입니다.

- 데이터베이스의 성능에 영향을 미치는 주요 사항을 결정합니다.
- 조회를 구현하는 데 사용할 수 있는 기법들을 식별합니다.
- 가장 효율적인 기법을 선택합니다.

SELECT와 같은 자료 조작 명령문은 사용자가 원하는 자료가 무엇인지만 지정하며, 그 자료를 검색하는 방법은 지정하지 않습니다. 이러한 자료에 대한 경로는 Optimizer에 의해 선택되며 액세스 계획에 저장됩니다. 이 절에서는 다음과 같은 타스크를 수행하기 위해 조회 Optimizer가 채택하는 기법들을 다루고 있습니다.

- 34 페이지의 『조회 비용 추정』
- 37 페이지의 『액세스 계획 유효성 검사』
- 38 페이지의 『결합 최적화』
- 56 페이지의 『그룹화 최적화』

Optimizer의 의사 결정 규칙

Optimizer는 일반적인 지침들을 사용하여 자료에 액세스하기 위한 최상의 방법을 선택합니다. Optimizer는 다음과 같은 역할을 합니다.

- 선택절의 각 술부에 대한 디폴트 필터 요소를 판별합니다.
- 내부에 저장된 정보에서 표의 속성을 추출합니다.
- 선택 술부가 색인의 가장 왼쪽 키와 일치할 때 술부의 진짜 필터 요소를 판별하기 위해 키 범위 추정을 실행합니다.
- 색인이 필요하지 않은 경우에는 표 스캔 처리에 드는 비용을 판별합니다.
- 색인이 필요한 경우, 표에 대한 색인을 작성하는 데 드는 비용을 판별합니다. 이 색인은 표 스캔을 실행하거나 색인의 색인을 작성함으로써 작성됩니다.
- 선택 조건이 적용되고 색인이 필요한 경우, 정렬 루틴이나 해싱 방법을 사용하는 데 드는 비용을 판별합니다.
- Optimizer는 사용할 수 있는 각 색인에 대해(대개는 가장 최근에 작성된 색인부터 시작하여 가장 오래전에 작성된 색인의 순서로) 다음 작업을 실행합니다. 이 작업은 시간 제한이 초과되기 전까지 이루어집니다.
  - 내부에 저장된 통계 정보에서 색인의 속성을 추출합니다.
  - 색인이 선택 기준을 만족시키는지 여부를 판별합니다.
  - 비용을 판별하는 데 도움이 되는 예상된 페이지 결함과 술부 필터 요소를 사용하여 색인 사용의 비용을 판별합니다.
  - 이 색인을 사용하는 데 드는 비용을 이전 비용(현재 최선의 비용)과 비교합니다.
  - 그 중 더 저렴한 쪽을 선택합니다.
  - 시간 제한이 초과되거나 색인이 더 이상 없을 때까지 최상 색인을 계속 검색합니다.

  *시간 제한*은 Optimizer가 구현 방법을 선택하는 데 소요되는 시간을 제어합니다. 현재까지 소요된 시간과 현재 발견된 최선의 구현 비용을 기준으로 사용합니다. 좋은 생각은 Optimizer가 조회를 실제로 실행하는 데 사용하는 시간보다 조회를 최적화하는 데 더 많은 시간을 사용하지 못하도록 하는 것입니다. 동적 SQL 조회는 Optimizer의 시간 제한사항의 영향을 받습니다. 그러나 정적 SQL 조회의 최적화 시간에는 한계가 없습니다. OPNQRYF의 경우, OPTALLAP(*YES)를 지정하면 최적화 시간에 한계가 없게 됩니다.

작은 표의 경우에는 조회 Optimizer가 조회를 최적화하는 데 시간을 거의 소모하지 않습니다. 그러나 큰 표의 경우에는 조회 Optimizer가 더 많은 색인들을 고려하게 됩니다. 일반적으로 Optimizer는 최적화 시간이 초과되기 전까지 각 결합 표에 대해 다섯 내지 여섯 개의 색인을 고려합니다. 따라서 표가 클수록 조회 분석에 소요되는 시간이 길어지는 것이 보통입니다.

조회 비용 추정

실행시 Optimizer는 데이터베이스의 현재 상태를 기준으로 *구현 비용*을 연산하는 방법으로 조회에 대한 최적의 액세스 방식을 선택합니다. Optimizer는 다음 사항 각각에 대한 액세스 비용을 모델화합니다.

- 표에서 행을 직접 읽기(표 스캔 처리)
- 색인을 통한 행 읽기(색인 스캔 키 선택 또는 색인 스캔 키 위치 지정 사용)
- 자료 공간에서 직접 색인 작성
- 기존 색인으로부터 색인(색인의 색인) 작성

- 조회 정렬 루틴 또는 해싱 방식 사용(조건이 만족되는 경우)
- 비트맵 처리 사용

특정 방식에 소요되는 비용은 다음 비용을 합계하여 계산됩니다.

- 시동 비용
- 지정된 최적화 모드와 연관된 비용
  - **SQL 사용시 최적화 모드와 연관된 비용:** SQL의 경우, 사전컴파일 옵션 ALWCPYDTA와 OPTIMIZE FOR n ROWS 절은 조회 Optimizer에게 최적화 목표를 달성할 것을 지시합니다.
    - Optimizer는 다음 두 가지 목표 중 하나를 가지고 SQL 조회를 최적화할 수 있습니다.
      1. 표에서 행의 첫 번째 버퍼를 검색하는 데 필요한 시간을 최소화합니다. 이 목표는 최적화가 색인을 작성하는 방법이 아닌 다른 방법으로 이루어지게 합니다.

         즉, 자료 스캔이나 기존 색인 중 하나가 선호됩니다. 이 모드는 다음의 두 가지 방법으로 지정할 수 있습니다.

         a. OPTIMIZE FOR n ROWS에서는 조회에서 검색하기를 바라는 행의 수를 사용자가 지정할 수 있습니다.

            Optimizer는 이 값을 사용하여 리턴되어 최적화될 행의 비율을 판별합니다. 값이 작은 경우에는 Optimizer가 첫 번째 n개의 행을 검색하는 데 필요한 시간을 최소화하게 됩니다.

         b. ALWCPYDTA(*NONE) 또는 (*YES)를 사전컴파일러 매개변수로 지정하면 Optimizer가 결과 행의 처음 3%를 검색하는 데 필요한 시간을 최소화할 수 있습니다.

            이 옵션은 OPTIMIZE FOR n ROWS를 지정하지 않은 경우에만 유효합니다.

      2. 선택한 모든 행이 어플리케이션으로 리턴된다고 가정하여 전체 조회를 처리하는 시간을 최소화합니다. 이 옵션은 Optimizer가 특정 액세스 방식에 치우치지 않도록 합니다. 이 모드는 다음의 두 가지 방법으로 지정할 수 있습니다.

         a. OPTIMIZE FOR n ROWS에서는 조회에서 검색하기를 바라는 행의 수를 사용자가 지정할 수 있습니다.

            Optimizer는 이 값을 사용하여 리턴되어 최적화될 행의 비율을 판별합니다. 예상되는 결과 행 수보다 크거나 같은 값을 지정하면 Optimizer가 전체 조회를 실행하는 데 필요한 시간을 최소화하게 됩니다.

         b. 사전컴파일러 매개변수로 지정된 ALWCPYDTA(*OPTIMIZE).

            이 옵션은 OPTIMIZE FOR n ROWS를 지정하지 않은 경우에만 유효합니다.

  - **OPNQRYF 사용시 최적화 모드와 연관된 비용:**
    - 지정된 최적화 매개변수(*FIRSTIO, *ALLIO 또는 *MINWAIT)와 연관된 비용.
      - *FIRSTIO -- 표에서 행의 첫 번째 버퍼를 검색하는 데 필요한 시간을 최소화합니다. 또한, 최적화가 색인을 작성하는 방법이 아닌 다른 방법으로 이루어지게 합니다. 즉, 자료 스캔이나 기존 색인 중 하나가 선호됩니다. *FIRSTIO를 선택하면 조회에서 검색하기를 바라는 행 수를 사용자가 지정할 수도 있습니다. Optimizer는 이 값을 사용하여 리턴되어 최적화될 행의 비율을 판별합니다. 값

이 작으면 첫 번째 n개의 행을 검색하는 데 필요한 시간을 최소화하게 되며 이는 *FIRSTIO와 유사합니다. 값이 크면 모든 n개의 행을 검색하는 데 필요한 시간을 최소화하게 되며 이는 *ALLIO와 유사합니다.

- *ALLIO -- 모든 조회 행을 표에서 읽었다고 가정하여 전체 조회를 처리하는 시간을 최소화합니다. 이 옵션은 Optimizer가 특정 액세스 방식에 치우치지 않도록 합니다.

  주: 사용자가 ALWCPYDTA(*OPTIMIZE)를 지정하고 조회 Optimizer가 정렬 루틴을 사용하기로 결정한 경우, 조회는 *ALLIO 최적화 매개변수에 따라 분석됩니다.

- *MINWAIT - 표에서 행을 읽을 때의 지연 시간을 최소화합니다. 즉, 열기 시간을 줄여 I/O 시간을 최소화합니다. 이 옵션은 최적화가 임시 색인을 작성하거나 정렬을 실행하는 쪽으로 이루어지게 합니다. 따라서 색인이 작성되거나 기존 색인이 사용됩니다.

- 색인 작성 비용
- 행 읽기에서 예상한 페이지 결함 수에 대한 비용과 예상한 수의 행 처리 비용.

  페이지 결함 및 처리되는 행의 수는 Optimizer가 데이터베이스 오브젝트로부터 얻은 통계 정보를 통해 예측할 수 있습니다. 이 정보에는 다음이 포함됩니다.

  – 표 *크기*
  – 행 *크기*
  – 색인 *크기*
  – 키 *크기*

  색인만 액세스하는 방법을 실행할 수 있는 경우에는 페이지 결함에 큰 영향을 미쳐 자료 공간에 대한 임의 입출력이 제거될 수도 있습니다.

  처리할 예상 행 수의 가중치 측정은 행 선택 술부의 관계 연산자인 *디폴트 필터 요소*의 검색 가능성을 기준으로 합니다.

  – 10% - 같다
  – 33% - 보다 작다, 보다 크다, 보다 작거나 같다, 보다 크거나 같다
  – 90% - 같지 않다
  – 25% - BETWEEN 범위(OPNQRYF %RANGE)
  – 10% - 각 IN 리스트 값(OPNQRYF %VALUES)

  키 범위 추정은 Optimizer가 하나 이상의 선택 술부에서 선택한 예상 행 수의 보다 정확한 추정치를 얻기 위해 사용하는 방법입니다. Optimizer의 추정 방법은 기존 색인의 가장 왼쪽 키에 대해 선택 술부를 적용하는 것입니다. 그러면 키 범위 기준 추정치를 통해 디폴트 필터 요소를 보다 세밀하게 정의할 수 있습니다. 가장 왼쪽의 키가 행 선택 술부에서 사용된 열과 일치하는 색인이 있는 경우, 이 색인은 선택 범주와 일치하는 항목의 수를 추정하는 데 사용될 수 있습니다. 항목 수에 대한 추정은 페이지 수 및 기계 색인의 키 밀도를 기준으로 하며, 항목을 실제로 액세스하지 않고 수행됩니다. 선택 술부에서 사용된 열에 대해 색인을 작성하면 최적화에 큰 도움을 줄 수 있습니다.

페이지 결함 및 처리된 행 수는 Optimizer가 선택한 액세스 유형에 따라 다릅니다. 액세스 방식에 대한 자세한 정보는 3 페이지의 제 2 장『iSeries용 DB2 UDB에 대한 자료 액세스: 자료 액세스 경로 및 방식』을 참조하십시오.

일반적인 조회 최적화 추가 정보

조회가 가능한 한 빠르게 실행될 수 있도록 하기 위한 추가 정보는 다음과 같습니다.

- Optimizer에 선택 가능한 값(키 범위 추정치)들을 제공하기 위해 가장 왼쪽의 키 열이 선택한 술부와 일치하는 색인을 작성합니다.
- 결합 조회의 경우, 결합 열과 일치하는 색인들을 작성하여 Optimizer가 일치하는 평균 행 수를 판별할 수 있도록 합니다.
- 조회와 관계 있는 열만 지정함으로써 관계 없는 맵핑을 최소합니다. 예를 들면, SQL SELECT문에 SELECT *를 지정하지 않고 조회해야 하는 열만 지정합니다. 또한, 열을 갱신할 필요가 없는 경우에는 FOR FETCH ONLY를 지정해야 합니다.
- 조회가 표 스캔 액세스 방식을 자주 사용하는 경우, RGZPFM(실제 파일(PF) 멤버 재구성) 명령을 사용하여 표에서 삭제된 행을 제거하거나, CHGPF(실제 파일 변경) REUSEDLT (*YES) 명령을 사용하여 삭제된 행을 재사용하십시오.

삽입된 SQL의 경우에는 다음의 사전컴파일 옵션 사용을 고려하십시오.

- 조회 Optimizer가 임시로 자료 사본을 작성하도록 하여 보다 향상된 성능을 얻으려면 ALWCPYDTA(*OPTIMIZE)를 지정하십시오.
- 열린 자료 경로가 다음 호출을 위해 계속 열려 있도록 하려면 CLOSWLCSR(*ENDJOB) 또는 CLOSQLCSR(*ENDACTGRP)을 사용하십시오.
- OPEN, EXECUTE 또는 DESCRIBE문이 실행될 때까지 SQL문의 유효성 검사가 지연되도록 하려면 DLYPRP(*YES)를 지정하십시오. 이 옵션을 사용하면 불필요한 유효성 검사가 제거되므로 성능이 향상됩니다.
- 읽기 전용 커서에 대해 행 블록화를 허용하려면 ALWBLK(*ALLREAD)를 사용하십시오.

OPNQRYF(열린 조회 파일) 조회의 경우에는 다음 매개변수 사용을 고려하십시오.

- 조회 Optimizer가 자료의 임시로 자료 사본을 작성하도록 하여 성능이 향상되는 경우에는 ALWCPYDTA(*OPTIMIZE)를 사용하여 조회 Optimizer가 임시로 자료 사본을 작성하도록 하십시오.
- 조회 Optimizer가 임시 색인을 작성하는 대신 기존 색인을 사용하는 쪽을 선택하도록 하려면 OPTIMIZE(*FIRSTIO)를 사용하십시오.

액세스 계획 유효성 검사

액세스 계획은 각 조회 요구를 만족시키는 데 필요한 조치들을 설명하는 제어 구조입니다. 액세스 계획에는 자료 및 자료 추출 방법에 대한 정보가 포함됩니다. 어떤 조회든지 최적화가 발생할 때마다 조회 Optimizer는 요구된 자료에 액세스하는 방법에 대한 최적화된 계획을 개발합니다. 이 정보는 소위 최소 계획이라고 하는 계획에 보관됩니다. 최소 계획과 QDT(조회 정의 템플리트)는 Optimizer와의 인터페이스 및 액세스 계획 수립에 사용됩니다.

- 동적 SQL의 경우, 액세스 계획은 작성만 되고 저장되지는 않습니다. 신규 액세스 계획은 PREPARE문이 실행될 때마다 작성됩니다.
- 정적이고 삽입된 SQL이 들어 있는 iSeries 프로그램의 경우, 액세스 계획은 삽입된 SQL문이 들어 있는 프로그램 또는 패키지의 *연관된 공간*에 저장됩니다.
- OPNQRYF의 경우, 액세스 계획은 작성만 되고 저장되지는 않습니다. 신규 액세스 계획은 OPNQRYF 명령이 처리될 때마다 작성됩니다.
- Query/400의 경우, 액세스 계획은 조회 정의 오브젝트의 일부로 저장됩니다.

결합 최적화

결합 조작은 좋은 성능을 얻기 위해 특별한 주의가 필요한 복잡한 기능입니다. 이 절에서는 iSeries용 DB2 Universal Database가 결합 조회를 구현하는 방법과 조회 Optimizer에 의해 최적화 선택사항이 작성되는 방법에 대해 설명합니다. 또한, 성능 문제를 최소화하거나 해결하는 데 유용한 설계 추가 정보 및 기법도 설명합니다. 다음과 같은 내용이 설명됩니다.

- 내포된 루프 결합 구현
  - 색인이 내포된 루프 결합
  - 도달순 결합이 내포된 루프 결합
- 해시 결합
- 결합 2차 다이얼에 대한 비용 추정 및 색인 선택
- 결합 조회의 성능 향상에 대한 추가 정보

내포된 루프 결합 구현

iSeries용 DB2 Universal Database는 **내포된 루프** 결합 방식을 제공합니다. 이 방식의 경우, 결합시 표 처리는 순서대로 이루어집니다. 이 순서를 **결합 순서**라고 합니다. 최종 결합 순서에 있는 첫 번째 표를 **1차 표**라고 하며, 다른 표들은 **2차 표**라고 합니다. 각 결합 표의 위치를 **다이얼**이라고 합니다. 내포된 루프는 2차 표의 색인 또는 2차 표의 표 스캔(도달순)을 사용하여 구현됩니다. 일반적으로 색인을 사용하여 결합이 구현됩니다. 2차 표가 사용자 정의 표 함수일 경우 결합은 표 스캔을 사용하여 구현됩니다.

색인이 내포된 루프 결합 구현: 결합 도중 iSeries용 DB2 Universal Database는 다음 작업을 실행합니다.

1. 술부에 의해 선택된 첫 번째 1차 표의 행에 1차 표에 대해 로컬로 액세스합니다.
2. 1차 표의 결합 열에서 키 값을 빌드합니다.
3. 색인 스캔 키 위치지정을 사용하여, 2차 표의 결합 조건 또는 로컬 행 선택 열과 키가 일치하는 색인을 사용하는 첫 번째 2차 표에 대해 결합 조건을 만족시키는 첫 번째 행을 찾습니다.
4. 적용 가능한 경우, 비트맵 선택을 적용합니다.
5. 남은 선택을 첫 번째 2차 다이얼에 대해 로컬로 적용하여 행이 선택되는지 여부를 판별합니다.

   2차 다이얼 행이 선택되지 않은 경우에는 결합 조건을 만족시키는 다음 행을 찾습니다. 모든 2차 표에서 결합 조건과 나머지 선택을 모두 만족시키는 행이 선택될 때까지 1-5단계를 반복합니다.
6. 결과 결합 행을 리턴합니다.

7. 마지막 2차 표를 다시 처리하여 해당 다이얼의 결합 조건을 만족시키는 다음 행을 찾습니다.

   이러한 처리 도중 결합 조건을 만족시키는 행을 더 이상 선택할 수 없게 되면, 처리가 논리적인 이전 다이얼로 백업되고 결합 조건을 만족시키는 다음 행을 읽으려고 시도합니다.
8. 1차 표에서 선택한 모든 행이 처리되면 처리를 종료합니다.

내포된 루프 결합 특성

내포된 루프 결합의 특징은 다음과 같습니다.

- 순서화 또는 그룹화가 지정되어 있고 모든 열이 단일 표에 있으며 이 표가 1차 표가 될 수 있으면, 이 표는 1차 표가 되고 표에 대한 색인을 사용하여 처리됩니다.
- 순서화 및 그룹화가 두 개 이상의 표에 대해 지정된 경우, iSeries용 DB2 Universal Database는 조회 처리를 다음과 같이 두 부분으로 나눕니다.
  1. 순서화 또는 그룹화 처리를 생략하고 결합 선택을 수행한 후, 그 결과 행을 임시 작업 표에 기록합니다. 그러면 Optimizer가 결합 조회의 모든 표를 1차 표에 대한 후보로 간주합니다.
  2. 임시 작업 표의 자료에 대한 순서화 또는 그룹화 처리가 실행됩니다.

  SQL ALWCPYDTA(*OPTIMIZE) 사전컴파일러 매개변수, 또는 OPNQRYF KEYFLD 및 ALWCPYDTA(*OPTIMIZE) 매개변수가 지정되면 조회 Optimizer가 조회를 다음의 두 부분으로 나누어 성능을 향상시킬 수도 있습니다.
- 색인을 사용하여 각 2차 다이얼에서 결합 조건을 만족시키는 모든 행을 찾습니다. 행 검색은 2차 표에서 임의 순서로 수행됩니다. 이 임의 디스크 I/O 시간은 대개 조회 처리 시간의 대부분을 차지합니다. 이전의 2차 다이얼 각각에 대해 결합 조건을 만족시키는 1차 및 이전의 2차 다이얼에서 선택한 각 행에 대해 주어진 2차 다이얼이 한 번 탐색되었기 때문에 상당수의 탐색은 이후 다이얼에 대해 실행될 수 있습니다. 이후 다이얼 처리가 비효율적이면 조회 처리 시간이 상당히 증가할 수 있습니다. 따라서 결합 조회를 위한 성능 고려 사항에 주의를 기울이면 결합 조회의 실행 시간을 수시간에서 수분까지 줄일 수 있습니다.
- 2차 다이얼에서 선택한 모든 행이 색인을 통해 다시 한 번 액세스됩니다. 효율적인 색인을 찾을 수 없으면 임시 색인이 작성됩니다. 일부 결합 조회는 결합 키 모두에 대해 색인이 존재하는 경우에도 2차 다이얼에 대한 임시 색인을 빌드합니다. 장시간 실행하는 조회의 2차 다이얼의 경우 효율성이 매우 중요하므로, 조회 Optimizer는 이 다이얼에 대해 로컬 행 선택을 전달하는 항목만 들어 있는 임시 색인을 빌드하도록 선택할 수 있습니다. 이러한 행 선택 사전처리로 데이터베이스 관리자는 행이 다이얼에 대해 일치할 때마다 행 선택을 처리하는 대신 한 번의 허용으로 행 선택을 처리할 수 있습니다.

표 스캔을 사용하여 도달순 결합이 내포된 루프 결합 구현:

결합 도중 iSeries용 DB2 Universal Database는 다음 작업을 실행합니다.

1. 술부에 의해 선택된 첫 번째 1차 표의 행에 1차 표에 대해 로컬로 액세스합니다.
2. 첫 번째 2차 표에 대해 결합 조건을 만족시키는 첫 번째 행을 찾기 위해 2차 표를 스캔합니다. 이 때, 2차 표의 로컬 행 선택 열 또는 결합 조건과 일치하는 행을 찾는 표 스캔을 사용합니다.

3. 남은 선택을 첫 번째 2차 다이얼에 대해 로컬로 적용하여 행이 선택되는지 여부를 판별합니다. 2차 다이얼 행이 선택되지 않은 경우에는 결합 조건을 만족시키는 다음 행을 찾습니다. 모든 2차 표에서 결합 조건과 나머지 선택을 모두 만족시키는 행이 선택될 때까지 1 - 3단계를 반복합니다.
4. 결과 결합 행을 리턴합니다.
5. 마지막 2차 표를 다시 처리하여 해당 다이얼의 결합 조건을 만족시키는 다음 행을 찾습니다. 이러한 처리 도중 결합 조건을 만족시키는 행을 더 이상 선택할 수 없게 되면, 처리가 논리적인 이전 다이얼로 백업되고 결합 조건을 만족시키는 다음 행을 읽으려고 시도합니다.
6. 1차 표에서 선택한 모든 행이 처리되면 처리를 종료합니다.

해시 결합

해시 결합 방식은 내포 루프 결합과 유사합니다. 그러나 해시 결합의 경우에는 색인을 사용하여 2차 표에서 일치하는 행을 찾는 대신, 해당 표에 대해 로컬 선택으로 선택한 모든 행이 들어 있는 해시 임시 결과표가 작성됩니다. 해시 표의 구조는 동일한 결합 값을 가진 행이 동일한 해시 표 파티션(클러스터된)으로 로드되는 구조입니다. 주어진 결합 값에 대한 행 위치는 결합 값에 해시 함수를 적용함으로써 찾을 수 있습니다.

내포된 루프 결합과 비교한 해시 결합의 장점

해시 결합은 내포된 루프 결합과 비교할 때 몇 가지 장점이 있습니다.

- 해시 임시 결과표의 구조가 색인의 구조보다 단순하므로 해시 표를 빌드하고 탐색하는 데 필요한 CPU 처리가 적습니다.
- 해시 결과표의 행에는 조회에 필요한 모든 자료가 들어 있으므로 해시 표 탐색시 임의 I/O로 표의 자료 공간을 액세스할 필요가 없습니다.
- 결합 값이 클러스터되는 것처럼, 주어진 결합 값에 대해 일치하는 모든 행에 단일 I/O 요구로 액세스할 수 있습니다.
- SMP 병렬화를 사용하여 해시 임시 결과표를 빌드할 수 있습니다.
- 색인과는 달리, 해시 표의 항목들은 기준 표의 열 값에 대한 변경사항을 반영하기 위해 갱신되지 않습니다. 해시 표가 있다고 해서 서버의 다른 갱신 작업을 처리하는 비용이 증가되지는 않습니다.

해시 결합을 사용할 수 없는 조회

다음과 같은 조회에는 해시 결합을 사용할 수 없습니다.

- 조회의 모든 부속 조회가 내부 결합으로 변환되지 않는 경우, 부속 조회를 실행하는 조회
- UNION 또는 UNION ALL을 실행하는 조회
- 왼쪽 외부 또는 예외 결합을 실행하는 조회
- DDS로 작성된 결합 논리 파일을 사용하는 조회
- ALWCPYDTA 사전컴파일러 매개변수에 대해 *NO 또는 *YES 매개변수가 지정되었을 때 자료에 대한 라이브 액세스를 필요로 하는 조회. 해시 결합은 ALWCPYDTA(*OPTIMIZE)로 실행되는 조회에만 사용

됩니다. 이 매개변수는 사전컴파일러 명령인 STRSQL CL 명령 또는 OPNQRYF CL 명령 중 하나로 지정할 수 있습니다. Client Access/400 ODBC 드라이버 및 조회 관리 드라이버는 항상 이 모드를 사용합니다.

- 조회를 실행하는 데 임시 결과표가 필요한 경우, OPTIMIZE(*YES)로 해시 결합을 사용할 수 있습니다.
- 해시 결합은 읽기 트리거가 있는 표 또는 실제 파일에 관련된 조회에는 사용할 수 없습니다.
- SQL ROLLBACK HOLD문이나 ROLLBACK CL 명령의 결과로 커서 위치가 복원되어야 하는 조회. *NONE 이외의 확약 제어 레벨을 사용하는 SQL 어플리케이션의 경우에는 ALWBLK 사전컴파일러 매개변수에 대한 값으로서 *ALLREAD를 지정해야 합니다.

해시 결합 및 병렬 처리

조회 변경 속성 CL 명령(CHGRYA)을 사용하여 변경할 수 있는 조회 속성. DEGREE는 Optimizer의 해시 결합 사용 선택을 작동할 수 있게 하거나 작동 불가능하게 하지 않습니다. 그러나 조회 속성 DEGREE가 *OPTIMIZE, *MAX 또는 *NBRTASKS 중 하나로 설정된 경우에는 해시 결합 조회가 SMP 병렬화를 사용할 수 있습니다.

해시 결합을 효과적으로 사용할 수 있는 조회 유형

해시 결합은 임시 색인이 빌드되었던 것과 동일한 대부분의 경우에 사용됩니다. 해시 결합을 사용하여 구현될 가능성이 가장 큰 결합 조회는 다음 중 하나를 만족하는 결합 조회입니다.

- 여러 결합 표의 모든 행이 결과 행 생성에 포함되는 경우
- 결합 결과와 관계된 표의 행 수를 줄이는 결합 표에 대해 유효 비결합 선택이 지정된 경우

모든 행을 처리하는 해시 결합 예

다음은 조회된 표의 모든 행을 처리하는 결합 조회의 예입니다.

```
    SELECT *
      FROM EMPLOYEE, EMP_ACT
      WHERE EMPLOYEE.EMPNO = EMP_ACT.EMPNO
  OPTIMIZE FOR 99999999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE EMP_ACT)) FORMAT(FORMAT1)
   JFLD((1/EMPNO 2/EMPNO *EQ))
   ALWCPYDTA(*OPTIMIZE)
```

이 조회는 다음 단계를 사용하여 구현됩니다.

1. EMPNO라는 키를 가진 표 EMP_ACT에 대해 임시 해시 표가 빌드됩니다. 이것은 해당 조회가 열려 있는 경우에 발생합니다.
2. EMPLOYEE 표에서 검색된 각 행에 대해, 임시 해시 표에서 일치하는 결합 값을 탐색합니다.
3. 일치하는 행이 발견된 각각에 대해 결과 행이 리턴됩니다.

이 해시 결합 조회를 SQL 프로그램에서 설명하기 위해 PRTSQLINF CL 명령으로 작성된 메세지는 다음과 같습니다.

```
   SQL402A  Hashing algorithm used to process join.
   SQL402B  Table EMPLOYEE used in hash join step 1.
SQL402B  Table EMP_ACT used in hash join step 2.
```

로컬 선택으로 제한된 조회에 대한 해시 결합 예

다음은 조회된 결합 표가 로컬 선택으로 크게 줄어든 결합 조회의 예입니다.

```
    SELECT EMPNO, LASTNAME, DEPTNAME
      FROM EMPLOYEE, DEPARTMENT
     WHERE EMPLOYEE.WORKDEPT = DEPARTMENT.DEPTNO
       AND EMPLOYEE.HIREDATE BETWEEN 1996-01-30 AND 1995-01-30
       AND DEPARTMENT.DEPTNO IN ('A00', 'D01', 'D11', 'D21', 'E11')
  OPTIMIZE FOR 99999999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE DEPARTMENT))
   FORMAT(FORMAT2)
   QRYSLT('1/HIREDATE *EQ %RANGE(''1996-01-30'' ''1995-01-30'')
   *AND 2/DEPTNO *EQ %VALUES(''A00'' ''D01'' ''D11'' ''D21''
   ''E11''')
   JFLD((1/WORKDEPT 2/DEPTNO *EQ))
   ALWCPYDTA(*OPTIMIZE)
```

이 조회는 다음 단계를 사용하여 구현됩니다.

1. 선택 술부, DEPTNO IN(‘A00’, ‘D01’, ‘D11’, ‘D21’, ‘E11’)과 일치하는 행이 들어 있는 DEPTNO라는 키 값을 가진 표 DEPARTMENT에 대해 임시 해시 표가 빌드됩니다. 이것은 해당 조회가 열려 있는 경우에 발생합니다.
2. 선택 술부(HIREDATE BETWEEN 1996-01-30 및 1995-01-30)와 일치하는 EMPLOYEE 표에서 검색된 각 행에 대해, 임시 해시 표에서 일치하는 결합 값을 탐색합니다.
3. 일치하는 행이 발견된 각각에 대해 결과 행이 리턴됩니다.

이 해시 결합 조회를 SQL 프로그램에서 설명하기 위해 PRTSQLINF CL 명령으로 작성된 메세지는 다음과 같습니다.

```
   SQL402A  Hashing algorithm used to process join.
   SQL402B  Table EMPLOYEE used in hash join step 1.
   SQL402B  Table DEPARTMENT used in hash join step 2.
```

순서화, 그룹화, 비동등 선택 또는 결과 열이 선택된 조회에 대한 해시 결합 예

서로 다른 표의 열에서 파생된 피연산자로 순서화, 그룹화 비동등 선택이 지정되거나 결과 열이 서로 다른 표의 열에서 파생된 경우에는 해시 결합이 이루어지고 결합의 결과 행이 임시 표에 기록됩니다. 그 다음에는 두 번째 단계로서, 임시 표를 사용하여 조회가 완료됩니다.

다음은 서로 다른 표의 열에서 파생된 피연산자로 지정된 결합 조회 예입니다.

```
    SELECT EMPNO, LASTNAME, DEPTNAME
      FROM EMPLOYEE, DEPARTMENT
     WHERE EMPLOYEE.WORKDEPT = DEPARTMENT.DEPTNO
           AND EMPLOYEE.EMPNO > DEPARTMENT.MGRNO
  OPTIMIZE FOR 99999999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE DEPARTMENT)
   FORMAT(FORMAT2)
   JFLD((1/WORKDEPT 2/DEPTNO *EQ) (1/EMPNO 2/MGRNO
   *GT))
```

이 조회는 다음 단계를 사용하여 구현됩니다.

1. DEPTNO라는 키를 가진 표 DEPARTMENT에 대해 임시 해시 표가 빌드됩니다. 이것은 해당 조회가 열려 있는 경우에 발생합니다.
2. EMPLOYEE 표에서 검색된 각 행에 대해, 임시 해시 표에서 일치하는 결합 값을 탐색됩니다.
3. 일치하는 행이 발견된 각각에 대해 결과 행이 임시 표에 기록됩니다.
4. 결합 결과 행이 모두 임시 표에 기록된 후에는 EMPNO > MGRNO로 선택된 행들이 임시 표에서 읽히고 어플리케이션으로 리턴됩니다.

이 해시 결합 조회를 SQL 프로그램에서 설명하기 위해 PRTSQLINF CL 명령으로 작성된 메세지는 다음과 같습니다.

```
   SQL402A  Hashing algorithm used to process join.
   SQL402B  Table EMPLOYEE used in hash join step 1.
   SQL402B  Table DEPARTMENT used in hash join step 2.
   SQL402C  Temporary result table created for hash join query.
```

결합 최적화 알고리즘

조회 Optimizer는 결합 열, 결합 연산자, 로컬 행 선택, 색인 사용 및 결합 조회를 위한 다이얼 순서화를 판별해야 합니다.

결합 열 및 결합 연산자는 다음에 따라 결정됩니다.

- 조회의 결합 열 스펙
- 결합 순서
- 다른 행 선택과 결합 열의 상호작용
- 사용되는 색인

다이얼에 대해 구현되지 않는 결합 스펙은 이후 다이얼에서 처리될 수 있을 때까지 연기되거나, 해당 다이얼에 대해 내부 결합이 실행되고 있던 경우에는 행 선택으로 처리됩니다.

주어진 다이얼에 대해, 해당 다이얼에 대한 결합 열로 사용할 수 있는 유일한 결합 스펙은 *이전* 다이얼로 결합되는 결합 스펙입니다. 예를 들어 2차 다이얼의 경우, 결합 조건을 만족시키는 데 사용할 수 있는 유일한 결

합 스펙은 1차 다이얼의 열을 참조하는 결합 스펙입니다. 마찬가지로, 3차 다이얼은 1차 및 2차 다이얼의 열을 참조하는 결합 스펙만 사용할 수 있으며, 이런 식으로 계속됩니다. 이후의 다이얼을 참조하는 결합 스펙은 참조된 다이얼이 처리될 때까지 연기됩니다.

주어진 다이얼에 대해 대개는 한 가지 유형의 연산자만 구현됩니다. 예를 들어, 하나의 내부 결합 스펙이 결합 연산자 '='을 가지고 있고 다른 결합 스펙은 결합 연산자 '>'을 가지고 있는 경우, Optimizer는 '=' 연산자를 사용하여 결합 구현을 시도합니다. '>' 결합 스펙은 '=' 스펙에 대해 일치하는 행을 찾은 후에 행 선택으로 처리됩니다. 또한, 동일한 연산자를 사용하는 복수 결합 스펙들은 함께 구현됩니다.

주: OPNQRYF의 경우에는 왼쪽 외부 또는 예외 결합 중 하나에 대해 한가지 유형의 결합만 허용됩니다. 즉, 모든 결합 조건에 대한 결합 연산자는 동일해야 합니다.

2차 다이얼을 액세스하기 위해 기존 색인을 찾는 경우, 조회 Optimizer는 색인의 가장 왼쪽 키 열을 조사합니다. 주어진 다이얼 및 색인에 대해서는 가장 왼쪽의 키 열을 사용하는 결합 스펙이 사용될 수 있습니다. 예를 들면 다음과 같습니다.

```
    DECLARE BROWSE2 CURSOR FOR
SELECT * FROM EMPLOYEE, EMP_ACT
      WHERE EMPLOYEE.EMPNO = EMP_ACT.EMPNO
        AND EMPLOYEE.HIREDATE = EMP_ACT.EMSTDATE
     OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE, EMP_ACT)) FORMAT(FORMAT1)
   JFLD((1/EMPNO 2/EMPNO *EQ)(1/HIREDATE 2/EMSTDATE
   *EQ))
```

키 열 EMPNO, PROJNO 및 EMSTDATE를 갖는 EMP_ACT에 대한 색인의 경우, 결합 조작은 열 EMPNO에 대해서만 실행됩니다. 결합이 수행된 다음, 열 EMSTDATE를 사용하여 색인 스캔 키 선택이 수행됩니다.

또한, 2차 다이얼에 대해 최적의 색인 사용을 선택한 경우에는 조회 Optimizer는 로컬 행 선택도 사용해야 합니다. 앞의 예를 로컬 술부로 표현하면 다음과 같습니다.

```
        DECLARE BROWSE2 CURSOR FOR
SELECT * FROM EMPLOYEE, EMP_ACT
      WHERE EMPLOYEE.EMPNO = EMP_ACT.EMPNO
        AND EMPLOYEE.HIREDATE = EMP_ACT.EMSTDATE
        AND EMP_ACT.PROJNO = '123456'
     OPTIMIZE FOR 99999 ROWS
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE, EMP_ACT)) FORMAT(FORMAT2)
   QRYSLT('2/PROJNO *EQ ''123456''')
   JFLD((1/EMPNO 2/EMPNO *EQ)(1/HIREDATE 2/EMSTDATE
   *EQ))
```

키 열 EMPNO, PROJNO 및 EMSTDATE를 갖는 색인은 세 개의 키 열 모두에 대해 결합과 선택을 하나의 조작으로 결합하여 최대한 사용됩니다.

임시 색인 작성시 가장 왼쪽의 키 열은 해당 다이얼 위치에서 사용할 수 있는 결합 열입니다. 이 다이얼에 대한 모든 로컬 행 선택은 포함을 위한 항목을 임시 색인으로 선택할 때 처리됩니다. 임시 색인은 선택/생략 키 논리 파일에 대해 위해 작성된 색인과 유사합니다. 앞의 예에 대한 임시 색인은 EMPNO 및 EMSTDATE라는 키 열을 가지게 됩니다.

OS/400® 조회 Optimizer는 액세스 경로 사용 판별시 결합과 로컬 행 선택의 조합을 시도하기 때문에 기존 색인을 사용하면서도 임시 색인의 장점과 동일한 장점을 거의 모두 성취할 수 있습니다. 위의 예에서는 어떤 구현을 사용하든지 기존 색인이 사용되거나 임시 색인이 작성됩니다. 임시 색인은 색인 작성 도중 적용된 PROJNO의 로컬 행 선택과 함께 빌드되며, 이 임시 색인은 EMPNO 및 EMSTDATE라는 키 열(결합 선택과 일치)을 갖게 됩니다. 그렇지 않고 기존 색인이 EMPNO, PROJNO, EMSTDATE(또는 PROJNO, EMP_ACT, EMSTDATE 또는 EMSTDATE, PROJNO, EMP_ACT 등)라는 키 열과 함께 사용된 경우에는 로컬 행 선택은 결합 선택과 **동시에** 적용될 수 있습니다(결합 선택보다 이전이 아니라 임시 색인이 작성되었을 때 또는 결합 선택 후, 색인의 첫 번째 키 열만 결합 열과 일치할 때).

기존 색인을 사용한 구현은 결합 및 선택 처리가 임시 색인을 빌드하기 위한 오버헤드 없이 결합되기 때문에 더 빠른 성능을 제공할 가능성이 커집니다. 그러나 기존 색인을 사용할 때는 로컬 선택이 한 번이 아니라 여러 번 실행되기 때문에 임시 색인보다 I/O 처리가 약간 느릴 수 있습니다. 일반적으로는, 동등 선택을 가장 왼쪽의 키로 사용하는 열과 결합 열의 조합에 대해 키 열과 함께 사용할 수 있는 기존 색인을 가지는 것이 좋습니다.

결합 순서 최적화

결합 순서는 결합 논리 파일이 참조될 경우 고정됩니다. 또한, OPNQRYF JORDER(*FILE) 매개변수가 지정되거나 조회 옵션 파일(QAQQINI) FORCE_JOIN_ORDER 매개변수가 *YES인 경우에도 결합 순서가 고정됩니다. 그 밖의 경우에는 표의 순서를 판별하기 위해 다음의 결합 순서화 알고리즘이 사용됩니다.

1. 1차 다이얼 후보로 각 개별 표에 대한 액세스 방식을 판별합니다.
2. 로컬 행 선택을 기반으로 하는 각 표에 대해 리턴되는 행 수를 추정합니다.

   행 순서화 또는 그룹화 처리가 사용되는 결합 조회가 1단계에서 처리되는 경우에는 순서화 또는 그룹화 열이 있는 표가 1차 표가 됩니다.
3. 1차 및 첫 번째 2차 표로서 후보가 되는 표들의 결합 조합 각각에 대해 액세스 방식, 비용 및 리턴되는 예상 행 수를 판별합니다.

   4개 표의 내부 결합에 추정되는 결합 순서 결합은 다음과 같습니다.

   1-2 2-1 1-3 3-1 1-4 4-1 2-3 3-2 2-4 4-2 3-4 4-3
4. 결합 비용이 가장 적게 드는 조합을 선택합니다.

   비용이 거의 동일한 경우에는 선택하는 행 수가 가장 작은 조합을 선택합니다.
5. 이전의 2차 표로 결합된 나머지 표 각각에 대해 비용, 액세스 방식 및 예상 행 수를 판별합니다.
6. 각 표에 대해 비용이 가장 적게 드는 액세스 방식을 선택합니다.
7. 결합 비용이 가장 적게 드는 2차 표를 선택합니다.

   비용이 거의 동일한 경우에는 선택하는 행 수가 가장 작은 조합을 선택합니다.

8. 비용이 가장 적게 드는 결합 순서가 판별될 때까지 45 페이지의 4-45 페이지의 7단계를 반복합니다.

주: 다이얼 32 이후, Optimizer는 다른 방법을 사용하여 파일 결합 순서를 결정하는데, 이는 최저 비용이 되지 않을 수도 있습니다.

조회 내에 왼쪽 또는 오른쪽 외부 결합 또는 오른쪽 예외 결합이 있으면, 결합 순서는 고정되지 않습니다. 단, ON절의 모든 from-column은 왼쪽이나 오른쪽 외부 또는 예외 결합 이전의 다이얼에서 발생해야 합니다. 예를 들면 다음과 같습니다.

```
FROM A INNER JOIN B ON A.C1=B.C1
LEFT OUTER JOIN C ON B. C2=C.C2
```

이 조회에 사용할 수 있는 결합 순서는 다음과 같습니다.

1-2-3, 2-1-3 또는 2-3-1

오른쪽 외부 또는 오른쪽 예외 결합은 왼쪽 외부 및 왼쪽 예외로 구현되는데 각각 파일이 플립됩니다. 예를 들면 다음과 같습니다.

```
FROM A RIGHT OUTER JOIN B ON A.C1=B.C1
```

이것은 B LEFT OUTER JOIN A ON B.C1=A.C1로 구현됩니다. 사용할 수 있는 결합 순서는 2-1뿐입니다.

결합 논리 파일이 참조되거나 결합 순서가 지정된 표 순서로 강제 적용될 경우, 조회 Optimizer가 모든 다이얼을 지정된 순서로 루프한 후, 최저 비용 액세스 방식을 판별합니다.

결합 2차 다이얼에 대한 비용 추정 및 색인 선택

45 페이지의 3단계와 45 페이지의 5단계에서 조회 Optimizer는 비용을 추정하여 주어진 다이얼 조합에 대한 액세스 방식을 선택해야 합니다. 선택 사항은 색인이 사용되어야 한다는 점을 제외하고는 행 선택을 위한 선택 사항과 유사합니다.

여러 가지의 가능한 액세스 선택 사항들을 비교할 때 조회 Optimizer는 각 후보에 숫자로 된 비용 값을 할당하고 이 값을 사용하여 처리 시간이 가장 적게 소요되는 구현을 판별해야 합니다. 이 비용 계산 값은 CPU 및 I/O 시간의 조합으로 이루어지며 다음과 같은 가정에 기초합니다.

- 표 페이지 및 색인 페이지가 보조 기억장치에서 검색되어야 합니다. 예를 들어, 조회 Optimizer는 SETOBJACC CL 명령의 결과로 전체 표가 활동 메모리로 로드된다는 사실을 인식하지 못합니다. 이 명령을 사용하면 조회의 성능을 크게 향상시킬 수는 있지만 조회 Optimizer가 표의 메모리 상주 상태를 이용하기 위해 조회 구현을 변경하지는 않습니다.
- 조회가 서버에서 실행되는 유일한 프로세스입니다. 동일한 자원을 사용하는 다른 프로세스로 인해 발생하는 서버 CPU 사용 또는 I/O 대기는 어떤 경우에도 허용되지 않습니다. CPU 관련 비용은 조회를 실행하는 서버의 관련 처리 속도에 따라 정해집니다.
- 열의 값은 표에 균일하게 분배됩니다. 예를 들어 표에 있는 행의 10%가 동일한 값을 가지는 경우에는 표의 열 번째 행마다 해당 값이 들어 있는 것으로 간주합니다.

- 열의 값은 행에 있는 다른 열의 값과 독립적입니다. 예를 들어 A라는 열에 표의 50%의 행에서 1이라는 값이 있고, 또 B라는 열에 50%의 행에서 2라는 값이 있는 경우, A = 1이고 B = 2인 행을 선택하는 조회가 표에서 25%의 행이 선택할 것으로 기대됩니다.

2차 다이얼에 대한 결합 비용 연산의 기본 요소는 이전 다이얼 모두에서 선택된 행의 수와 이전 다이얼에서 선택된 각 행과 평균적으로 일치하는 행의 수입니다. 이 두 가지 요소 모두 주어진 다이얼에 대해 대응 행 수를 추정하여 구할 수 있습니다.

결합 연산자가 같은 것 이외의 다른 것인 경우, 대응하는 예상 수는 다음의 디폴트 필터 요소를 기준으로 합니다.

- 33% - 보다 작다, 보다 크다, 보다 작거나 같다, 보다 크거나 같다
- 90% - 같지 않다
- 25% - BETWEEN 범위(OPNQRYF %RANGE)
- 10% - 각 IN 리스트 값(OPNQRYF %VALUES)

예를 들어, 결합 연산자가 보다 작다이면 일치하는 예상 행 수는 .33 * (다이얼의 행 수)입니다. 현재 다이얼에 대해 사용 중인 결합 스펙이 없는 경우에는 카테시안(Cartesian)을 연산자로 가정합니다. 카테시안의 경우에는 일치하는 행 수가 다이얼의 모든 행이 됩니다. 단, 로컬 행 선택을 색인에 적용할 수 없는 경우에 한합니다.

결합 연산자가 같다인 경우에는 예상 행 수가 주어진 값에 대해 중복되는 행의 평균 수입니다.

iSeries는 색인 유지보수(색인에 있는 키 값의 삽입 및 삭제)를 수행하고, 색인의 지정된 키 열에 대한 고유 값 수의 실행 계수를 유지보수합니다. 이러한 통계는 색인 오브젝트와 함께 바인드되어 항상 유지보수됩니다. 조회 Optimizer는 조회 최적화시 이 통계를 사용합니다. 이러한 통계를 유지보수하면 색인 유지보수에 상당한 양의 오버헤드가 추가되는 일이 없습니다. 이 통계 정보는 다음와 같은 색인에만 사용할 수 있습니다.

- 다양한 길이의 문자 키가 들어 있지 않은 색인

  주: 다양한 길이의 문자 열을 결합 열로 사용한 경우에는 CRTLF CL 명령을 사용하여 다양한 길이의 문자 열을 고정된 문자 열로 맵핑하는 색인을 작성할 수 있습니다. 다양한 길이의 자료에 대해 정의된 고정 길이 문자 키가 들어 있는 색인은 중복 값 평균 수 통계를 제공합니다.
- 버전 2 릴리스 3 또는 그 이상의 버전이 설치된 iSeries 서버에서 작성되거나 리빌드된 색인

  주: 조회 Optimizer는 OS/400 이전 버전에서 작성된 색인을 사용하여 결합 키 값이 갖는 평균 중복 값 수가 높은지 또는 낮은지를 추정합니다. 색인이 하나의 결합 키로만 정의된 경우에는, 색인의 *크기*를 기준으로 추정이 이루어집니다. 대부분의 경우 색인의 추가 키는 해당 색인에 일치하는 행 추정을 유효하지 않게 만듭니다. 일부 결합 조회의 성능은 이러한 색인을 리빌드함으로써 향상될 수 있습니다.

중복 값의 평균 수 통계는 색인의 가장 왼쪽 키 중 처음 4개에 대해서만 유지보수됩니다. 5개 이상의 결합 열을 지정하는 조회의 경우에는, 가장 왼쪽의 4개 키 열 안에서 사용할 수 있는 *중복 값의 평균 수* 통계로 색인

을 찾을 수 있도록 복수의 추가 색인을 작성하는 것이 유용합니다. 이는 결합 열의 일부가 약간 고유한 경우 (*중복 값의 평균 수가 낮은 경우*)에 특히 중요합니다.

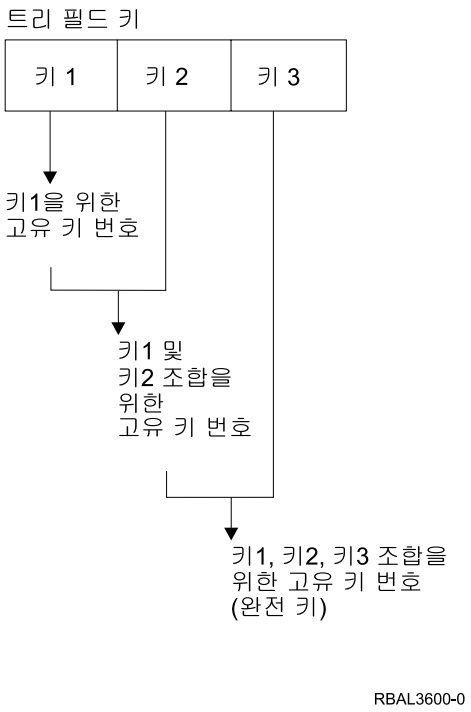


그림 1. 3 키 색인에 대한 평균 중복 값 수

이 통계는 색인 리빌드 및 작성의 일부로서 유지보수됩니다.

동등 결합에 대한 평균 중복 값 수 또는 다른 결합 연산자에 대한 디폴트 필터 값을 사용하여 대응 행 수를 구합니다. 다음 식은 이전 다이얼에서 결합 행 수를 계산하는 데 사용됩니다.

```
NPREV =  Rp * M2 * FF2 * ..... *Mn * FFn .....
```

NPREV
이전의 모든 다이얼에 있는 결합 행 수

Rp 1차 다이얼에서 선택된 행 수

M2 다이얼 2에 대응하는 행 수

FF2 다이얼 2에 대해 로컬인 술부로 위의 M2를 사용하여 아직 적용되지 않은 술부에 대한 필터링 감소 요소

Mn 다이얼 n에 대응하는 행 수

FFn 다이얼 n에 대해 로컬인 술부로 위의 Mn을 사용하여 아직 적용되지 않은 술부에 대한 필터링 감소 요소

주: 현재 다이얼 이전의 2차 다이얼 각각에 대해 대응 행 수(Mn)와 필터 감소 필터 요소(FFn) 쌍을 곱합니다.

이전 다이얼에서 결합 행의 수를 계산하고 나면 Optimizer는 액세스 방식에 대한 비용을 생성할 준비가 되었습니다.

표에서 얻은 임시 색인 또는 해시 임시 결과표: 조회 Optimizer가 분석한 첫 번째 액세스 방식 선택은 *표의 임시 색인 또는 해시 임시 결과표*를 빌드하는 것입니다. 표에서 빌드된 임시 색인 또는 해시 표를 통해 결합 2차 다이얼 액세스 비용을 계산하는 기본 공식은 다음과 같습니다.

```
JSCOST  =  CRTDSI +
           NPREV *((MATCH * FF * KeyAccess)
           + (MATCH * FF * FCost)) *
           FirstIO
```

JSCOST

결합 2차 비용

CRTDSI

임시 색인 또는 해시 임시 결과표를 빌드하는 비용

NPREV

이전의 모든 다이얼에서 얻은 결합 행 수

MATCH

대응하는 행 수(대개는 평균 중복 값)

KeyAccess

색인 또는 해시 표에서 키 액세스 비용

FF 해당 다이얼의 로컬 술부에 대한 필터링 요소(전이 폐쇄로 인해 이전 다이얼에서 실행된 선택 제외)

FCost 표에서 행에 액세스하는 비용

FirstIO

첫 번째 버퍼 검색을 위해 최적화하려는 최적화 목표로 인해 비시동 비용을 줄이는 감소율. 자세한 정보는 34 페이지의 『조회 비용 추정』을 참조하십시오.

이 2차 다이얼 액세스 방식은 사용할 수 있는 색인을 찾을 수 없는 경우나 임시 색인 또는 해시 표의 성능이 기존 색인보다 더 나은 경우에 사용됩니다. 이 방식이 기존 색인을 사용하는 것보다 더 좋은 이유는, 다음 조건을 만족하는 경우 색인 또는 해시 표가 작성되면 행 선택이 완료되기 때문입니다.

- 대응 행 수(MATCH)가 큽니다.
- 이전의 모든 다이얼에서 얻은 결합 행 수(NPREV)가 큽니다.
- 일부 필터 감소(FF < 100%)가 있습니다.

색인에서 얻은 임시 색인 또는 해시 표: 이 액세스 방식을 선택하기 위한 기본 비용 공식은 한 가지점만 제외하고는 표에서 빌드된 임시 색인 또는 해시 표를 사용할 때와 같습니다. 그 차이점은 임시 색인을 빌드하는 비용 CRTDSI가 기존 색인을 통한 행 선택을 포함시키도록 연산된다는 점입니다. 이 액세스 방식은 동일한 이유로 결합 2차 다이얼 액세스에 사용됩니다. 그러나 색인에서의 작성 비용이 더 적게 들 수 있습니다.

기존 색인 사용: 마지막 액세스 방식은 기존 색인을 사용하는 것입니다. 기존 색인을 통한 결합 2차 다이얼의 액세스 비용을 계산하는 기본 공식은 다음과 같습니다.

```
JSCOST  =  NPREV *((MATCH * KeyAccess)
         + (MATCH * FCost)) *
           FirstIO
```

JSCOST

결합 2차 비용

NPREV

이전의 모든 다이얼에서 얻은 결합 행 수

MATCH

해당 색인에서 찾을 수 있는 대응하는 키 수(대개는 평균 중복 값)

KeyAccess

색인에서 키에 액세스하는 비용

FCost 표에서 행에 액세스하는 비용

FirstIO

첫 번째 버퍼 검색을 위해 최적화하려는 최적화 목표로 인해 비시동 비용을 줄이는 감소율. 자세한 정보는 34 페이지의 『조회 비용 추정』을 참조하십시오.

I/O 최적화가 사용되는 경우(즉, OPNQRYF OPTIMIZE(*FIRSTIO))에는, 전체 비용이 줄어들기 때문에 이것이 가장 가능성 있는 액세스 방식입니다. 또한, 이전의 모든 다이얼에서 얻은 결합 행 수(NPREV)와 대응하는 키 수(MATCH)가 작은 경우에도 이것이 가장 효율적인 방식입니다.

조회 Optimizer는 다음과 같은 경우에 가장 왼쪽의 선행 키로서 결합 열의 서브세트만 갖는 색인 사용을 고려합니다.

- 평균 중복 값 수 통계에서 중복 값을 갖는 행의 평균 수가 아주 적은지 여부를 판별할 수 있는 경우
- 이전 다이얼에서 선택된 행 수가 적은 경우

전이 폐쇄를 통해 생성되는 술부

결합 조회의 경우에는 조회 Optimizer가 추가 선택을 생성하기 위해 약간 특수한 처리를 할 수도 있습니다. 조회에 속한 술부 세트가 추가 술부를 논리적으로 추론하는 경우, 조회 Optimizer는 추가 술부를 생성합니다. 그 목적은 결합 최적화 도중 추가 정보를 제공하기 위한 것입니다.

전이 폐쇄로 인해 추가되는 술부의 예:

```
SELECT * FROM EMPLOYEE, EMP_ACT
  WHERE EMPLOYEE.EMPNO = EMP_ACT.EMPNO
    AND EMPLOYEE.EMPNO = '000010'
```

Optimizer는 이 조회를 다음과 같이 수정합니다.

```
SELECT * FROM EMPLOYEE, EMP_ACT
  WHERE EMPLOYEE.EMPNO = EMP_ACT.EMPNO
    AND EMPLOYEE.EMPNO = '000010'
    AND EMP_ACT.EMPNO  = '000010'
```

OPNQRYF 예 :

```
OPNQRYF FILE((EMPLOYEE EMP_ACT)) FORMAT(FORMAT1)
   QRYSLT('1/EMPNO *EQ ''000010''')
   JFLD((1/EMPNO 2/EMPNO *EQ))
```

Optimizer는 이 조회를 다음과 같이 수정합니다.

```
OPNQRYF FILE((EMPLOYEE EMP_ACT)) FORMAT(FORMAT1)
   QRYSLT('1/EMPNO *EQ ''000010'' *AND
   2/EMPNO *EQ ''000010''')
   JFLD((1/EMPNO 2/EMPNO *EQ))
```

다음 규칙은 다른 결합 다이얼에 추가되는 술부를 판별합니다.

- 영향을 받는 다이얼은 결합 연산자가 같아야 합니다.
- 술부가 **분리 가능**하다는 것은 해당 술부의 잘못된 조건으로 인해 행이 생략될 수 있다는 의미입니다.
- 술부의 한 피연산자는 동등 결합 열이고 다른 피연산자는 상수 또는 호스트 변수입니다.
- 술부 연산자는 LIKE 또는 IN(OPNQRYF %WLDCRD, %VALUES 또는 *CT)이 아닙니다.
- 술부는 OR을 사용하여 다른 술부에 연결되지 않습니다.
- 다이얼 결합 유형은 내부 결합입니다.

조회 Optimizer는 WHERE 절(OPNQRYF QRYSLT 매개변수)에 술부가 이미 존재하는지 여부에 상관 없이 신규 술부를 생성합니다.

일부의 술부는 중복됩니다. 이것은 조회의 다른 술부에 대한 이전 평가로 술부가 제공하는 결과가 이미 판별된 경우에 발생합니다. 중복 술부는 사용자가 지정한 것일 수도 있고 술부 조작 도중 조회 Optimizer가 생성한 것일 수도 있습니다. 술부 연산자가 =, >, >=, <, <=, 또는 BETWEEN(OPNQRYF *EQ, *GT, *GE, *LT, *LE 또는 %RANGE)인 중복 술부들은 최상의 선택 범위를 반영하도록 단일 술부로 병합됩니다.

조회에 대한 복수 결합 유형

복수 결합 유형(내부, 왼쪽 외부, 오른쪽 외부, 왼쪽 예외 및 오른쪽 예외)이 JOIN 구문을 사용하는 조회에서 지정될 수 있기는 하지만, iSeries 사용권 내부 코드는 전체 조회에 대해 하나의 결합 유형(내부, 왼쪽 외부 또는 왼쪽 예외 결합 유형)만 지원할 수 있습니다. 따라서, Optimizer는 조회에 대한 전반적인 결합 유형을 무엇으로 해야 하는지 결정하고 올바른 의미론을 갖기 위해 파일을 재정렬해야 합니다.

주: 이 섹션의 내용은 OPNQRYF에는 적용되지 않습니다.

Optimizer는 각 다이얼 및 전체 조회에 대한 결합 유형을 판별하기 위해 지정될 수 있는 행 선택과 더불어 결합 범주를 평가합니다. 이 정보가 알려지면 Optimizer는 조회 내에서 발생할 수 있는 다양한 결합 유형을 시뮬레이트하기 위해 표의 관련 행 번호를 사용하여 추가 선택을 생성합니다.

왼쪽 외부 또는 예외 결합의 경우에 대해 일치하지 않는 행에는 널(null) 값이 리턴되기 때문에, WHERE 절에서 지정될 수 있는 추가의 결합 범주를 포함하여 해당 다이얼에 대해 지정된 분리 가능한 선택은 대응하지 않는 모든 행을 제외시킵니다(IS NULL 술부에 대한 선택이 아닌 경우). 따라서 IS NULL 술부가 지정된 경우에는 해당 다이얼에 대한 결합 유형이 내부 결합(또는 예외 결합)으로 변경되게 됩니다.

다음 예에서는 표 EMPLOYEE와 DEPARTMENT간에 왼쪽 외부 결합이 지정되어 있습니다. WHERE 절에는 DEPARTMENT 표에도 적용되는 두 가지 선택 술부가 있습니다.

```
SELECT EMPNO, LASTNAME, DEPTNAME, PROJNO
 FROM CORPDATA.EMPLOYEE XXX LEFT OUTER JOIN CORPDATA.DEPARTMENT YYY
        ON XXX.WORKDEPT = YYY.DEPTNO
      LEFT OUTER JOIN CORPDATA.PROJECT ZZZ
        ON XXX.EMPNO = ZZZ.RESPEMP
 WHERE XXX.EMPNO = YYY.MGRNO AND
        YYY.DEPTNO IN ('A00', 'D01', 'D11', 'D21', 'E11')
```

첫 번째 선택 술부인 XXX.EMPNO = YYY.MGRNO는 결합 범주에 추가되어 "내부 결합"의 결합 조건으로서 평가되는 추가 결합 조건입니다. 두 번째는 대응하지 않는 행들을 제외시키는 분리 가능 선택 술부입니다. 이러한 선택 술부 중 어느 하나로 인해 DEPARTMENT 표에 대한 결합 유형이 왼쪽 외부 결합에서 내부 결합으로 변경될 수 있습니다.

EMPLOYEE와 DEPARTMENT 표 간의 결합이 내부 결합으로 변경되었다 해도 전체 조회는 PROJECT 표에 대한 결합 조건을 만족시키기 위해 여전히 왼쪽 외부 결합 상태로 있어야 합니다.

주: 복수 결합 유형을 지정할 때는 대응하지 않는 행의 경우 선택을 조회에 추가해야만 복수 결합 유형이 지원되므로 주의해야 합니다. 이는 해당 개별 다이얼의 결합 유형을 기준으로 대응하지 않는 행을 선택하거나 제외시키게 되는 임의의 선택이 적용되기 전에는 결합 범주를 만족시키는 결과 행의 수가 매우 커질 수 있음을 의미합니다.

JOIN 구문의 사용 방법에 대한 자세한 정보는 SQL 프로그래밍 개념 서적에 있는 둘 이상의 표에서 자료 결합을 참조하거나 SQL 참조서를 참조하십시오.

결합 조회 성능 문제점 소스

위에서 설명한 최적화 알고리즘은 대부분의 결합 조회에 유용하지만 소수의 조회 성능이 저하될 수도 있습니다. 조회 성능 저하는 다음과 같은 경우에 발생합니다.

- 잠재적 결합 열에 대해 평균 중복 값 평균 수 통계를 제공하는 색인을 사용할 수 없는 경우

  주: 46 페이지의 『결합 2차 다이얼에 대한 비용 추정 및 색인 선택』에서는 색인 통계에 대한 제한사항을 피하거나 잠재적 결합 열에 대한 추가 색인을 작성(존재하지 않을 경우)하는 방법에 대한 제안사항을 제공합니다.

- 표에 로컬 선택을 적용할 때(선택 열에 대한 색인이 존재하지 않기 때문에) 선택되는 행의 수를 추정하기 위해 조회 Optimizer가 디폴트 필터 요소를 사용하는 경우.

  선택 열에 대한 색인을 작성하면 조회 Optimizer가 키 범위 추정을 사용하여 보다 정확한 필터링 추정을 실행할 수 있습니다.

- 결합 열에 대해 선택된 특정 값이 표에 있는 모든 결합 열에 대한 평균 중복 값 수보다 훨씬 큰 대응 열을 산출하는 경우(즉, 자료가 균일하게 분배되지 않은 경우).

  로컬 행 선택에 대응하는 선택/생략 스펙을 사용하여 색인이 있는 논리 파일을 빌드하려면 DDS를 사용하십시오. 그러면 조회 Optimizer가 선택된 키에 대해 대응하는 행의 수를 보다 정확하게 추정할 수 있습니다.

주: Optimizer는 자료가 균일하게 분배되지 않은 선택/생략 색인에서 판별을 더 잘 할 수 있습니다.

- 조회 Optimizer가 응답 세트에서 검색될 행 수에 대해 잘못된 가정을 하는 경우.

  SQL 프로그램의 경우, 사전컴파일 옵션 ALWCPYDTA(*YES)를 지정하면 해당 프로그램의 조회가 기존 색인을 사용할 가능성이 커집니다. 마찬가지로, ALWCPYDTA(*OPTIMIZE)를 지정하면 해당 프로그램의 조회가 임시 색인을 작성할 가능성이 커집니다. 또한, SQL 절 OPTIMIZE FOR n ROWS를 사용하여 조회 Optimizer에 영향을 미칠 수도 있습니다.

  OPNQRYF 명령의 경우, OPTIMIZE 키워드에 대한 잘못된 성능 옵션이 지정되는 경우가 있습니다. 기존 색인의 사용 가능성을 높이려면 *FIRSTIO를 지정하십시오. 임시 색인의 작성 가능성을 높이려면 *ALLIO를 지정하십시오.

결합 조회의 성능 향상을 위한 추가 정보

성능이 떨어지는 결합 조회를 조사하거나, 결합 조회를 사용하는 신규 어플리케이션을 작성하려는 경우에는 다음의 체크 리스트가 유용합니다.

표 2. 결합 조회를 사용하는 어플리케이션 작성을 위한 체크 리스트

작업	설명
데이터베이스 설계를 검사합니다. 모든 결합 열 및/또는 행 선택 열에 대해 사용할 수 있는 색인이 있는지 확인합니다. CRTLF를 사용하는 경우에는, 색인이 비공유인지 확인합니다.	이러한 작업을 하면 조회 Optimizer가 평균 중복 값의 수를 판별할 수 있기 때문에 보다 효율적인 액세스 방식을 선택할 수 있습니다. 상당수의 조회는 기존 색인을 사용하여 조회를 구현하고 임시 색인 작성 비용을 없앨 수 있습니다.
조회를 검사하여, Optimizer가 각 다이얼의 선택 가능성을 보다 잘 알 수 있도록 다른 다이얼에 일부 복합 술부를 추가해야 하는지 여부를 확인합니다.	조회 Optimizer는 OR로 연결된 술부나 분리 불가능 술부, 또는 술부 연산자가 LIKE 또는 IN인 술부에 대해서는 술부를 추가하지 않으므로 이러한 술부를 추가하여 조회를 수정하는 것이 좋습니다
CRTLF CL 명령을 사용하는 조회의 선택/생략 스펙과 일치하는 선택/생략 스펙이 있는 색인을 작성합니다.	이 단계는 통계 특성이 전체 표에 대해 균일하지 않은 경우에 유용합니다. 예를 들어 높은 중복 요소를 가지는 하나의 값이 있고 나머지 열 값들은 고유한 경우, 선택/생략 색인을 작성하면 Optimizer가 해당 키에 대한 값의 분배를 불균일하게 함으로써 선택한 값에 대한 올바른 최적화를 실행 할 수 있게 됩니다.
ALWCPYDTA(*OPTIMIZE) 또는 ALWCPYDTA(*YES)를 지정합니다.	조회가 임시 색인을 작성하는 경우, 그리고 Optimizer가 기존 색인만 사용하는 경우의 처리 시간이 더 효율적이라고 판단되는 경우에는 ALWCPYDTA(*YES)를 지정합니다. 조회가 임시 색인을 작성하지 않는 경우, 그리고 임시 색인을 작성하는 경우의 처리 시간이 더 효율적이라고 판단되는 경우에는 ALWCPYDTA(*OPTIMIZE)를 지정합니다. 또한, 해당 어플리케이션에 전체 결과 행을 읽으려는 목적이 있음을 Optimizer에게 알리려면 OPTIMIZE FOR n ROWS를 지정합니다. 이 경우 n은 큰 수로 설정해야 합니다. 조회가 끝나기 전에 n을 작은 수로 설정할 수도 있습니다.
OPNQRYF의 경우, OPTIMIZE(*FIRSTIO) 또는 OPTIMIZE(*ALLIO)를 지정합니다.	조회가 임시 색인을 작성 중인 경우, 그리고 Optimizer가 기존 색인만 사용하는 경우의 처리 시간이 더 효율적이라고 판단되는 경우에는 OPTIMIZE(*FERSTIO)를 지정합니다. 조회가 임시 색인을 작성하지 않으며 임시 색인을 작성하는 경우의 처리 시간이 더 효율적이라고 판단되는 경우에는 OPTIMIZE(*ALLIO)를 지정합니다.

표 2. 결합 조회를 사용하는 어플리케이션 작성을 위한 체크 리스트 (계속)

작업	설명

결합 논리 파일을 사용하거나 조회 옵션 파일(QAQQINI) FORCE_JOIN_ORDER 매개변수를 *YES로 합니다. OPNQRYF 사용자는 JORDER(*FILE)를 지정할 수 있습니다.	하나의 표가 모든 2차 표와 함께 연속적으로 결합되는 결합을 때로 **성형 결합**이라고 합니다. 모든 2차 결합 술부에 특정 표에 대한 열 참조가 포함되는 성형 결합의 경우에는, 해당 표가 제1 결합 위치에 있을 때 성능 장점이 있습니다. 예 A에서, 모든 표는 표 EMPLOYEE로 결합됩니다. 조회 Optimizer는 결합 순서를 자유롭게 결정할 수 있습니다. 조회는 예 B에 표시된 것과 같이 결합 옵션 파일(QAQQINI) FORCE_JOIN_ORDER 매개변수 *YES 또는 OPNQRYF JORDER(*FILE)을 사용하여 EMPLOYEE가 제1 결합 위치가 되도록 변경되어야 합니다. 이 예에서 결합 유형은 리턴된 디폴트 값이 없는 결합(이러한 결합을 내부 결합이라고 함)임을 주의하십시오. 이 표가 첫 번째 위치로 강제되는 이유는 임의 I/O 처리를 피하기 위해서 입니다. EMPLOYEE가 제1 결합 위치에 있지 않으면 결합 프로세스 도중 EMPLOYEE의 모든 행이 반복적으로 검사될 수 있습니다. EMPLOYEE의 *크기*가 매우 크면, 많은 임의 I/O 처리가 발생하게 되어 성능을 저하시킵니다. EMPLOYEE가 첫 번째 위치에 있도록 하면 임의 I/O 처리는 최소화됩니다.
	
	예 A: 성형 결합 조회
	`DECLARE C1 CURSOR FOR`
	`SELECT * FROM DEPARTMENT, EMP_ACT, EMPLOYEE,`
	`PROJECT`
	`WHERE DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT`
	`AND EMP_ACT.EMPNO=EMPLOYEE.EMPNO`
	`AND EMPLOYEE.WORKDEPT=PROJECT.DEPTNO`
	
	예 B: FORCE_JOIN_ORDER를 통해 순서를 강제로 적용한 성형 결합 조회
	`DECLARE C1 CURSOR FOR`
	`SELECT * FROM EMPLOYEE, DEPARTMENT, EMP_ACT,`
	`PROJECT`
	`WHERE DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT`
	`AND EMP_ACT.EMPNO=EMPLOYEE.EMPNO`
	`AND EMPLOYEE.WORKDEPT=PROJECT.DEPTNO`
	
	예 A: 성형 결합 조회(OPNQRYF)
	`OPNQRYF FILE((DEPARTMENT EMP_ACT EMPLOYEE`
	`PROJECT)) FORMAT(FORMAT1)`
	`JFLD((1/DEPTNO 3/WORKDEPT *EQ)`
	`(2/EMPNO 3/EMPNO *EQ)`
	`(3/WORKDEPT 4/DEPTNO *EQ))`
	
	예 B: JORDER(*FILE) 매개변수를 사용하는 성형 결합 조회(OPNQRYF)
	`OPNQRYF FILE((EMPLOYEE DEPARTMENT EMP_ACT`
	`PROJECT)) FORMAT(FORMAT1)`
	`JFLD((2/DEPTNO 1/WORKDEPT *EQ)`
	`(3/EMPNO 1/EMPNO *EQ)`
	`(1/WORKDEPT 4/DEPTNO *EQ))`
	`JORDER(*FILE)`
	
	주: EMPLOYEE의 열을 ORDER BY 절(OPNQRYF KEYFLD 매개변수)로 지정하는 경우에도 EMPLOYEE를 제1 결합 위치에 두게 될 수 있습니다. 그러면 조회 Optimizer는 나머지 표에 대해 최상의 순서를 선택할 수 있게 됩니다.

조회 Optimizer가 정렬 루틴을 사용할 수 있도록 ALWCPYDTA (*OPTIMIZE)를 지정합니다.	순서화가 지정되어 있고 모든 키 열이 단일 다이얼에서 얻어지는 경우, ALWCPYDTA(*OPTIMIZE)를 지정하면 조회 Optimizer가 가능한 모든 결합 순서를 고려할 수 있게 됩니다.

표 2. 결합 조회를 사용하는 어플리케이션 작성을 위한 체크 리스트 (계속)

작업	설명
한 표의 모든 행들이 다른 표의 각 행에 결합되는 것을 방지하기 위해 결합 술부를 지정합니다.	이러한 결합 술부를 지정하면 결합 팬아웃을 줄이게 되어 성능 향상됩니다. 모든 2차 표에는 '결합처' 열로서 해당 열을 참조하게 하는 결합 술부가 최소한 하나는 있어야 합니다.

세 개 이상의 표에서 자료 선택시 성능 향상을 위한 추가 정보

고려 중인 선택문이 둘 이상의 표에 액세스하는 경우에는 115 페이지의 제 5 장『대형 표에 빨리 액세스하기 위해 색인 사용』에 제안된 모든 권장사항이 적용됩니다. 다음의 제안은 특히 몇 개의 표에 액세스하는 선택문에 대한 것입니다. 세 개 이상의 표를 포함하는 결합에서는 결합 열에 대한 중복 정보를 제공하려는 경우도 있습니다. 결합 요구시 작업할 추가 정보를 Optimizer에게 제공하려는 경우가 이에 해당합니다. 그러면 Optimizer는 결합을 실행하기 위한 최상의 방법을 판별할 수 있습니다. 추가 정보는 중복적인 것처럼 보일 수도 있으나 Optimizer에게는 유용한 정보입니다. 예를 들면 다음과 같이 코딩하지 않습니다.

```
EXEC SQL
  DECLARE EMPACTDATA CURSOR FOR
  SELECT LASTNAME, DEPTNAME, PROJNO, ACTNO
       FROM CORPDATA.DEPARTMENT, CORPDATA.EMPLOYEE,
              CORPDATA.EMP_ACT
       WHERE DEPARTMENT.MGRNO = EMPLOYEE.EMPNO
              AND EMPLOYEE.EMPNO = EMP_ACT.EMPNO
END-EXEC.
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE(CORPDATA/DEPARTMENT CORPDATA/EMPLOYEE CORPDATA/EMP_ACT)
  FORMAT(FORMAT1)
   JFLD((1/MGRNO 2/EMPNO *EQ) (2/EMPNO 3/EMP_ACT *EQ))
```

Optimizer에 더 많은 자료와 코드를 제공하려면 다음과 같이 지정하십시오.

```
EXEC SQL
  DECLARE EMPACTDATA CURSOR FOR
  SELECT LASTNAME, DEPTNAME, PROJNO, ACTNO
       FROM CORPDATA.DEPARTMENT, CORPDATA.EMPLOYEE,
              CORPDATA.EMP_ACT
       WHERE DEPARTMENT.MGRNO = EMPLOYEE.EMPNO
              AND EMPLOYEE.EMPNO = EMP_ACT.EMPNO
              AND DEPARTMENT.MGRNO = EMP_ACT.EMPNO
END-EXEC.
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE(CORPDATA/DEPARTMENT CORPDATA/EMPLOYEE CORPDATA/EMP_ACT)
  FORMAT(FORMAT1)
   JFLD((1/MGRNO 2/EMPNO *EQ) (2/EMPNO 3/EMP_ACT *EQ)
   (1/MGRNO 3/EMPNO *EQ))
```

그룹화 최적화

이 섹션에서는 iSeries용 DB2 Universal Database가 그룹화 기법을 구현하는 방법과 조회 Optimizer가 최적화 선택을 하는 방법을 설명합니다. 조회 Optimizer에는 그룹화를 구현하기 위한 두 가지 선택사항이 있는데, 이는 해시 구현과 색인 구현입니다.

그룹화 해시 구현

이 기법은 기본 해시 액세스 방식을 사용하여 선택된 표 행의 그룹화 또는 요약화를 실행합니다. 선택된 각 행에 대해, 지정된 그룹화 값이 해시 기능을 통해 실행됩니다. 계산된 해시 값과 그룹화 값은 그룹화 값에 해당되는 해시 표의 항목을 빠르게 찾는 데 사용됩니다. 현재의 그룹화 값이 이미 해시 표에서 하나의 행을 가지는 경우에는, 요구된 그룹화 열 연산(SUM 또는 COUNT 등)을 기준으로 현재 표의 행 값 해시 표 값을 검색하여 요약(갱신)됩니다. 현재의 그룹화 값에 대해 해시 표 항목을 찾을 수 없는 경우에는 해시 표에 신규 항목이 삽입되어 현재의 그룹화 값으로 초기화됩니다.

이러한 구현에서 첫 번째 그룹 결과를 수신하는 데 필요한 시간은 다른 그룹화 구현의 경우보다 더 길어질 가능성이 큽니다. 이는 먼저 해시 표를 빌드하여 채워야 하기 때문입니다. 해시 표를 완전히 채우고 나면 데이터베이스 관리자가 이 표를 사용하여 그룹화 결과를 리턴하기 시작합니다. 결과를 리턴하기 전에 데이터베이스 관리자는 해시 표의 요약 항목에 지정된 그룹화 선택 범주 또는 순서화를 적용해야 합니다.

그룹화 해시 방식이 가장 효율적인 경우

그룹화 해시 방식은 통합 비율이 높을 때 가장 효과적입니다. **통합 비율**이란 계산된 그룹화 결과에 대한 선택된 표 행의 비율을 말합니다. 모든 데이터베이스 표 행이 자체의 고유 그룹화 값을 가지는 경우에는 해시 표가 매우 커질 수 있습니다. 따라서 해시 액세스 방식의 속도가 느려지게 됩니다.

Optimizer는 먼저 지정된 그룹화 열에 있는 고유 값 수(즉, 데이터베이스 표에 있는 예상 그룹 수)를 판별하여 통합 비율을 추정합니다. 그런 다음 Optimizer는 표의 전체 행 수와 지정된 선택 범주를 검사하고 이 검사 결과를 사용하여 통합 비율을 추정합니다.

그룹화 열에 대한 색인은 Optimizer의 비율 추정을 보다 정확하게 합니다. 색인에는 키 열에 대한 평균 중복 값 수를 포함하는 통계가 포함되어 있기 때문에 정확도를 높일 수 있습니다.

Optimizer는 또한 그룹의 예상 수 추정치를 사용하여 해시 표의 파티션 수를 계산합니다. 이미 언급한 바와 같이, 해시 액세스 방식은 해시 표의 균형이 잘 이루어져 있을 때 가장 효율적입니다. 해시 표 파티션 수는 해시 표간에 항목들이 분배되는 방법과 이 분배의 균일성에 직접적인 영향을 미칩니다.

해시 기능은 그룹화 값이 숫자 이외의 자료 유형(정수(2진) 자료 유형 예외)을 가지는 열로 이루어질 때 그 성능이 보다 향상됩니다. 또한, 가변 길이와 연관되지 않는 그룹화 값 열 및 널(null) 열 속성을 지정하면 해시 기능의 성능이 보다 효율적일 수 있습니다.

색인 그룹화 구현

이 구현은 색인 스캔 키 선택 또는 색인 스캔 키 위치지정 액세스 방식을 사용하여 그룹화를 실행합니다. 여기에는 그룹화 열 모두가 연속적인 가장 왼쪽의 키 열로서 들어 있는 색인이 필요합니다. 데이터베이스 관리자는 색인을 통해 개별 그룹에 액세스하고 요구된 요약 기능을 실행합니다.

정의를 통해 색인에는 함께 그룹화되는 키 값이 모두 들어 있으므로 첫 번째 그룹 결과는 해시 방식보다 더 짧은 시간 안에 리턴될 수 있습니다. 이는 해시 방법에 필요한 임시 결과가 없기 때문입니다. 이 구현은 어플리케이션이 모든 그룹 결과를 검색할 필요가 없는 경우나, 그룹화 열과 일치하는 색인이 이미 존재하는 경우에 유용합니다.

그룹화가 색인을 사용하여 구현되고 그룹화 열을 만족시키는 기존의 영구 색인이 없는 경우에는 임시 색인이 작성됩니다. 조회 내에서 지정된 그룹화 열은 이 색인에서 키 열로 사용됩니다.

그룹화 열 제거를 통한 그룹화 최적화

모든 그룹화 열은 그룹화 열 리스트에서 제거될 수 있는지 여부를 판별하기 위해 평가됩니다. 동등 연산자가 지정된 상태에서 분리 가능 선택 술부를 가지는 그룹화 열만이 고려될 수 있습니다. 이는 해당 열이 단일 값에만 대응될 수 있도록 해주며 고유 그룹을 판별하는 데 도움을 주지는 않습니다. 이러한 처리는 Optimizer가 더 많은 색인을 고려하여 조회를 구현하고 임시 색인 또는 해시 표에 키 열로 추가되는 열 수를 줄일 수 있도록 해줍니다.

다음 예는 Optimizer가 그룹화 열을 제거할 수 있는 조회를 보여줍니다.

```
    DECLARE DEPTEMP CURSOR FOR
      SELECT EMPNO, LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
  WHERE EMPNO = '000190'
        GROUP BY EMPNO, LASTNAME, WORKDEPT
```

OPNQRYF 예 :

```
OPNQRYF FILE(EMPLOYEE) FORMAT(FORMAT1)
  QRYSLT('EMPNO *EQ ''000190''')
   GRPFLD(EMPNO LASTNAME WORKDEPT)
```

이 예에서는 EMPNO = '000190'라는 선택 술부로 인해 Optimizer가 EMPNO를 그룹화 열 리스트에서 제거할 수 있습니다. 키 열로 지정된 LASTNAME 및 WORKDEPT를 가지는 색인만 조회를 구현하도록 고려될 수 있으며, 임시 색인 또는 해시가 필요한 경우에는 EMPNO가 사용되지 않습니다.

주: EMPNO가 그룹화 열 리스트에서 제거될 수 있기는 하지만, 세 개의 그룹화 열 모두를 갖는 영구 색인이 존재하는 경우에는 Optimizer가 해당 색인을 사용하기로 선택할 수도 있습니다.

읽기 트리거를 제거하여 그룹화 최적화

읽기 트리거가 있는 표 또는 실제 파일에 관련된 조회에서, Group By 트리거는 항상 Group By 처리보다 먼저 임시 파일을 포함하므로 조회의 속도를 느리게 합니다.

주: 읽기 트리거는 ADDPFTRG 명령이 TRGTIME (*AFTER) 및 TRGEVENT (*READ)와 함께 표에 사용되었을 때 추가됩니다.

조회는 읽기 트리거가 제거될 때 더 빠르게 실행됩니다(RMVPFTRG TRGTIME (*AFTER) TRGEVENT (*READ)).

추가 그룹화 열 추가를 통한 그룹화 최적화

그룹화 열을 제거하는 데 적용되는 것과 동일한 논리가 조회에 추가 그룹화 열을 추가하는 데도 사용될 수 있습니다. 이는 그룹화를 구현하는 데 색인을 사용할 수 있는지 여부를 판별하려는 경우에만 이루어집니다.

다음 예는 Optimizer가 추가 그룹화 열을 추가할 수 있는 조회를 보여줍니다.

```
    CREATE INDEX X1 ON EMPLOYEE
          (LASTNAME, EMPNO, WORKDEPT)

DECLARE DEPTEMP CURSOR FOR
      SELECT LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
  WHERE EMPNO = '000190'
        GROUP BY LASTNAME, WORKDEPT
```

OPNQRYF 예 :

```
OPNQRYF FILE ((EMPLOYEE)) FORMAT(FORMAT1)
  QRYSLT('EMPNO *EQ ''000190''')
   GRPFLD(LASTNAME WORKDEPT)
```

이러한 조회 요구의 경우, Optimizer는 조회를 위해 X1을 고려할 때 EMPNO를 추가 그룹화 열로 추가할 수 있습니다.

색인 건너뜀 키 처리를 사용한 그룹화 최적화

색인 건너뜀 키 처리는 기존 색인을 사용하는 키순 구현 알고리즘으로 그룹화할 때 사용될 수 있습니다. 색인 건너뜀 키 처리 알고리즘은 다음 작업을 실행합니다.

1. 색인을 사용하여 그룹에 대한 위치를 지정합니다.
2. 그룹에 대해 선택 범주와 일치하는 첫 번째 행을 찾은 다음, 지정된 경우에는 그룹에서 널(null) 값이 아닌 첫 번째 MIN 또는 MAX 값을 찾습니다.
3. 그룹을 사용자에게 리턴합니다.
4. 다음 그룹으로의 "건너뜀"을 실행하여 처리를 반복합니다.

이러한 과정은 그룹에 대한 모든 색인 키 값을 잠재적으로 처리하지 않음으로써 성능을 향상시키게 됩니다.

색인 건너뜀 키 처리는 다음의 조회에 사용됩니다.

- 다음과 같은 경우 키순 그룹화 구현을 사용하는 단일 표 조회
  - 조회에 열 함수가 없는 경우
  - 또는, 조회에 하나의 MIN 또는 MAX 열 함수만 있고 MIN 또는 MAX의 피연산자가 그룹화 열 다음의 색인에 있는 다음 키 열인 경우. 이 경우 조회에는 다른 그룹화 함수가 있을 수 없습니다. MIN 함수의 경우에는 키 열이 오름차순 키이어야 하며, MAX 함수의 경우에는 키 열이 내림차순 키이어야 합니다. 조회가 전체 표 그룹화 조회인 경우에는 MIN 또는 MAX의 피연산자가 첫 번째 키 열이어야 합니다.

예 1, SQL 사용:

```
CREATE INDEX IX1 ON EMPLOYEE (SALARY DESC)

DECLARE C1 CURSOR FOR
   SELECT MAX(SALARY) FROM EMPLOYEE;
```

예 1, OPNQRYF 명령 사용:

```
OPNQRYF FILE(EMPLOYEE) FORMAT(FORMAT1)
   MAPFLD((MAXSAL '%MAX(SALARY)'))
```

이 경우 조회 Optimizer는 색인 IX1을 사용하기로 선택합니다. SLIC 실행 시간 코드는 SALARY에 대해 널(null)이 아닌 첫 번째 값을 찾을 때까지 색인을 스캔합니다. SALARY가 널이 아니라는 가정하에 실행시 코드는 첫 번째 색인 키로 위치지정되고 해당 키 값을 급여의 MAX로서 리턴합니다. 그 다음에는 더 이상의 색인 키가 처리되지 않습니다.

예 2, SQL 사용:

```
CREATE INDEX IX2 ON EMPLOYEE (DEPT, JOB,SALARY)

DECLARE C1 CURSOR FOR
SELECT DEPT, MIN(SALARY)
FROM EMPLOYEE
WHERE JOB='CLERK'
GROUP BY DEPT
```

예 2, OPNQRYF 명령 사용:

```
OPNQRYF FILE(EMPLOYEE) FORMAT(FORMAT2)
QRYSLT('JOB *EQ ''CLERK''')
GRPFLD((DEPT))
MAPFLD((MINSAL '%MIN(SALARY)'))
```

이 경우 조회 Optimizer는 색인 IX2를 사용하기로 선택합니다. SLIC 실행시 코드는 JOB이 ‘CLERK’인 DEPT의 첫 번째 그룹으로 위치지정되고 SALARY를 리턴합니다. 그런 다음 코드는 JOB이 ‘CLERK’인 다음 DEPT 그룹으로 건너뜁니다.

- 다음을 만족하는 결합 조회
  - 모든 그룹화 열이 단일 표에서 추출되어야 합니다.
  - 각 다이얼에 대해 해당 다이얼을 참조하는 MIN 또는 MAX 열 함수의 피연산자는 하나여야 하며 조회에 다른 열 함수는 있을 수 없습니다.
  - MIN 또는 MAX 함수 피연산자가 그룹화 열과 동일한 다이얼에 있는 경우, 단일 표 조회와 동일한 규칙이 사용됩니다.
  - MIN 또는 MAX 함수 피연산자가 서로 다른 다이얼에 있는 경우, 해당 다이얼에 대한 결합 열은 그룹화 열 중 하나로 결합되고 해당 다이얼에 대한 색인에는 결합 열 다음에 MIN 또는 MAX 피연산자가 와야 합니다.

    예 1, SQL 사용:

```
CREATE INDEX IX1 ON DEPARTMENT(DEPTNAME)

CREATE INDEX IX2 ON EMPLOYEE(WORKDEPT, SALARY)

DECLARE C1 CURSOR FOR
   SELECT DEPTNAME, MIN(SALARY)
      FROM DEPARTMENT, EMPLOYEE
      WHERE DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT
      GROUP BY DEPARTMENT.DEPTNO;
```

예 1, OPNQRYF 명령 사용:

```
OPNQRYF FILE(DEPARTMENT EMPLOYEE) FORMAT(FORMAT1)
JFLD((1/DEPTNO 2/WORKDEPT *EQ))
GRPFLD((1/DEPTNO))
MAPFLD((MINSAL '%MIN(SALARY)'))
```

순서화 최적화

이 섹션에서는 iSeries용 DB2 Universal Database가 순서화 기법을 구현하는 방법과 조회 Optimizer에 의해 최적화 선택사항이 작성되는 방법에 대해 설명합니다. 조회 Optimizer는 색인 순서화 또는 정렬을 사용하여 순서화를 구현합니다.

정렬 순서화 구현

정렬 알고리즘은 행을 정렬 공간으로 읽어 들인 다음 지정된 순서화 키를 기준으로 행을 정렬합니다. 그 다음에는 순서화된 정렬 공간에서 사용자에게로 행이 리턴됩니다.

색인 순서화 구현

색인 순서화 구현에는 순서화 열 모두가 연속적인 가장 왼쪽의 키 열로 들어 있는 색인이 필요합니다. 데이터베이스 관리자는 색인 순서로 색인을 통해 개별 행에 액세스하고, 그 결과로 행을 순서대로 리퀘스터에게 리턴합니다.

이 구현은 어플리케이션이 모든 순서화 결과를 검색할 필요가 없는 경우나, 순서화 열과 일치하는 색인이 이미 존재하는 경우에 유용합니다. 순서화가 색인을 사용하여 구현되고 순서화 열을 만족시키는 기존의 영구 색인이 없는 경우에는 임시 색인이 작성됩니다. 조회 내에서 지정된 순서화 열은 이 색인에서 키 열로 사용됩니다.

순서화 열 제거를 통한 순서화 최적화

모든 순서화 열은 순서화 열 리스트에서 제거될 수 있는지 여부를 판별하기 위해 평가됩니다. 동등 연산자가 지정된 상태에서 분리 가능 선택 술부를 가지는 순서화 열만이 고려될 수 있습니다. 이는 해당 열이 단일 값에만 대응될 수 있도록 해주며 순서대로 판별하는 데 도움을 주지는 않습니다.

이러한 처리는 조회 구현시 Optimizer가 더 많은 색인을 고려하고 임시 색인에 키 열로서 추가되는 열의 수를 줄일 수 있도록 해줍니다. 다음의 SQL 예는 Optimizer가 순서화 열을 제거할 수 있는 조회를 보여 줍니다.

```
DECLARE DEPTEMP CURSOR FOR
  SELECT EMPNO, LASTNAME, WORKDEPT
  FROM CORPDATA.EMPLOYEE
  WHERE EMPNO = '000190'
  ORDER BY EMPNO, LASTNAME, WORKDEPT
```

OPNQRYF 예 :

```
OPNQRYF FILE(EMPLOYEE) FORMAT(FORMAT1)
  QRYSLT('EMPNO *EQ ''000190''')
  KEYFLD(EMPNO LASTNAME WORKDEPT)
```

이 예에서는 EMPNO = '000190'라는 선택 술부로 인해 Optimizer가 EMPNO를 순서화 열 리스트에서 제거할 수 있습니다. 키 열로 지정된 LASTNAME 및 WORKDEPT만 가지는 색인이 조회를 구현하기 위해 고려될 수 있으며, 임시 색인이 필요한 경우에는 EMPNO가 사용되지 않습니다.

주: EMPNO가 그룹화 열 리스트에서 제거될 수 있기는 하지만, 세 개의 순서화 열 모두를 가지는 영구 색인이 존재하는 경우에는 Optimizer가 해당 색인을 사용하기로 선택할 수도 있습니다.

추가 순서화 열 추가를 통한 순서화 최적화

순서화 열을 제거하는 데 적용되는 것과 동일한 논리가 조회에 추가 그룹화 열을 추가하는 데도 사용될 수 있습니다. 이는 순서화를 구현하는 데 색인을 사용할 수 있는지 여부를 판별하려는 경우에만 이루어집니다.

다음 예는 Optimizer가 추가 순서화 열을 추가할 수 있는 조회를 보여줍니다.

```
CREATE INDEX X1 ON EMPLOYEE (LASTNAME, EMPNO, WORKDEPT)

DECLARE DEPTEMP CURSOR FOR
  SELECT LASTNAME, WORKDEPT
  FROM CORPDATA.EMPLOYEE
  WHERE EMPNO = '000190'
  ORDER BY LASTNAME, WORKDEPT
```

OPNQRYF 예 :

```
OPNQRYF FILE ((EMPLOYEE)) FORMAT(FORMAT1)
  QRYSLT('EMPNO *EQ ''000190''')
  KEYFLD(LASTNAME WORKDEPT)
```

이러한 조회 요구의 경우, Optimizer는 조회를 위해 X1을 고려할 때 EMPNO를 추가 순서화 열로 추가할 수 있습니다.

보기 구현

보기는 다음 두 가지 방법 중 하나를 사용하여 조회 Optimizer에 의해 구현합니다.

- Optimizer가 조회 선택문을 보기의 선택문과 결합합니다(보기 합성).
- Optimizer가 보기 결과를 임시 표에 배치한 다음 조회의 보기 참조를 임시 표로 대체합니다(보기 구체화).

이는 내포된 표 표현식과, 주가 있는 표현식을 제외한 일반적인 표현식에도 적용됩니다.

보기 합성 구현

보기 합성 구현은 조회 선택문을 보기의 선택문과 결합하여 신규 조회를 생성합니다. 그 다음에는 신규의 결합된 선택문 조회가 기준이 되는 기본 표에 대해 직접 실행됩니다.

이 하나의 합성문은 자료에 대한 허용이 한 번만 필요하기 때문에 보기가 포함된 조회에서 선호되는 구현입니다.

예:

```
CREATE VIEW  D21EMPL AS
  SELECT * FROM CORPDATA.EMPLOYEE
  WHERE WORKDEPT='D21'
```

SQL 사용:

```
  SELECT LASTNAME, FIRSTNME, SALARY
  FROM D21EMPL
  WHERE JOB='CLERK'
```

OPNQRYF 사용:

```
OPNQRYF FILE(D21EMPL)
  FORMAT(FORMAT1)
  QRYSLT('JOB *EQ ''CLERK''')
```

이 예에서 조회 Optimizer는 다음 예와 비슷한 신규 조회를 생성합니다.

```
SELECT LASTNAME, FIRSTNME, SALARY
  FROM CORPDATA.EMPLOYEE
  WHERE WORKDEPT='D21' AND JOB='CLERK'
```

조회에는 사용자의 조회가 선택한 열과, 조회에서 참조되는 기본 표, 그리고 보기 및 사용자 조회 모두의 선택이 포함됩니다.

주: 사용자는 조회 Optimizer가 생성하는 신규 합성 조회를 볼 수 없습니다. 사용자와 데이터베이스 성능 분석 툴은 보기에 대한 원래의 조회만 볼 수 있습니다.

보기 구체화 구현

보기 구체화 구현은 보기 조회를 실행한 다음 그 결과를 임시 결과표에 배치합니다. 그 다음에는 사용자 조회의 보기 참조가 임시 표로 대체되고 조회는 임시 결과표에 대해 실행됩니다.

보기 구체화는 보기 합성 작성이 불가능한 경우에도 이루어집니다. 보기 구체화가 필요한 조회 유형은 다음과 같습니다.

- 보기의 완전 외부 선택이 그룹화를 포함하며, 조회가 그룹화를 포함하고, HAVING 또는 선택 리스트에서 보기의 열 함수에서 파생된 열을 참조하는 경우
- 조회가 결합 조회이고 보기의 완전 외부 선택이 그룹화 또는 DISTINCT를 포함하는 경우
- 보기의 완전 외부 선택이 DISTINCT를 포함하고 조회가 UNION, 그룹화 또는 DISTINCT를 사용하며, 다음 중 하나에 해당되는 경우
  - 조회만 공유된 가중치 NLSS 표를 갖는 경우
  - 보기만 공유된 가중치 NLSS 표를 갖는 경우
  - 조회와 보기 모두 공유된 가중치 NLSS 표를 갖지만 두 표가 서로 다른 경우
- 조회가 열 함수를 포함하고 보기의 완전 외부 선택이 DISTINCT를 포함하는 경우

- 보기가 액세스 계획을 포함하지 않는 경우. 이러한 경우는 보기가 보기를 참조하고 위에 열거한 이유 중 하나로 인해 보기 합성이 작성되지 않는 경우에 발생합니다. 이는 내포된 표 표현식과 일반적인 표현식에는 적용되지 않습니다.

임시 결과 표가 작성되었으므로, ALWCPYDTA(*OPTIMIZE)가 허용되는 액세스 방식이 조회를 구현하는 데 사용될 수 있습니다. 이러한 액세스 방식으로는 해시 그룹화, 해시 결합 및 비트맵이 있습니다.

예:

```
CREATE VIEW AVGSALVW AS
SELECT WORKDEPT, AVG(SALARY) AS AVGSAL
  FROM CORPDATA.EMPLOYEE
  GROUP BY WORKDEPT
```

SQL 예:

```
  SELECT D.DEPTNAME, A.AVGSAL
  FROM CORPDATA.DEPARTMENT D, AVGSALVW A
  WHERE D.DEPTNO=A.WORKDEPT
```

OPNQRYF 예 :

```
OPNQRYF FILE(CORPDATA/DEPARTMENT AVGSALVW)
  FORMAT(FORMAT1)
  JFLD((1/DEPTNO 2/WORKDEPT *EQ))
```

이 경우에는 결합 조회가 그룹화 보기를 참조하기 때문에 보기 합성이 작성되지 않습니다. 그 대신 AVGSALVW의 결과가 임시 결과표(*QUERY0001)에 배치됩니다. 보기 참조 AVGSALVW는 임시 결과표로 대체됩니다. 그런 다음 신규 조회가 실행됩니다. 생성된 조회는 다음과 비슷합니다.

```
SELECT D.DEPTNAME, A.AVGSAL
  FROM CORPDATA.DEPARTMENT D, *QUERY0001 A
  WHERE D.DEPTNO=A.WORKDEPT
```

주: 사용자는 조회 Optimizer가 생성하는 신규 조회를 볼 수 없습니다. 사용자와 데이터베이스 성능 분석 툴은 보기에 대한 원래의 조회만 볼 수 있습니다.

조회의 분리 가능 선택(부속 조회 술부 제외)은 가능한 경우에 항상 보기 구체화 프로세스에 추가됩니다. 그러면 더 작은 임시 결과표가 생성되고 보기 구체화시 기존 색인을 사용할 수 있습니다. 동일한 보기 또는 조회의 일반적인 표 표현식에 대한 둘 이상의 참조가 있는 경우에는 이러한 과정이 실행되지 않습니다. 다음은 보기 구체화에 분리 가능 선택이 추가되는 예입니다.

```
SELECT D.DEPTNAME,A.AVGSAL
  FROM CORPDATA.DEPARTMENT D, AVGSALVW A
  WHERE D.DEPTNO=A.WORKDEPT
  A.WORKDEPT LIKE 'D%' AND AVGSAL>10000
```

OPNQRYF 예 :

```
OPNQRYF FILE(CORPDATA/DEPARTMENT AVGSALVW)
  FORMAT(FORMAT1)
  JFLD((1/DEPTNO 2/WORKDEPT *EQ))
  QRYSLT('1/WORKDEPT *EQ %WLDCRD(''D*'') *AND 2/AVGSAL *GT 10000')
```

조회의 분리 가능 선택은 신규 조회의 보기 결과에 추가되어 임시 결과표를 생성합니다.

```
SELECT WORKDEPT, AVG(SALARY) AS AVGSAL
  FROM CORPDATA.EMPLOYEE
  WHERE WORKDEPT LIKE 'D%'
  GROUP BY WORKDEPT
  HAVING AVG(SALARY)>10000
```

제 4 장 조회 최적화 툴을 사용한 조회 성능 최적화

조회 최적화 툴을 사용하여 자료 검색 시간을 줄일 수 있습니다. 툴의 결과를 사용하여 다음 작업을 실행할 수 있습니다.

- 서버 선택한 자료 액세스 방식을 변경할 수 있습니다. 5 페이지의 『자료 액세스 방식: 요약』을 참조하십시오.
- 올바른 색인을 작성하고 이를 효율적으로 사용할 수 있습니다. 115 페이지의 제 5 장 『대형 표에 빨리 액세스하기 위해 색인 사용』을 참조하십시오.

조회 최적화는 반복적인 프로세스입니다. 조회 최적화를 위해 필요한 경우 다음 작업을 실행하십시오.

조회에 대한 통계 정보 수집

조회에 대한 통계를 수집하는 방법은 여러가지가 있습니다. 다음은 통계가 수집될 수 있는 방법의 샘플입니다.

- 66 페이지의 『SQL 어플리케이션의 성능 확인』
- 67 페이지의 『작업 기록부에서 조회 Optimizer 디버그 메세지 점검』
- 75 페이지의 『PRTSQLINF 명령을 사용하여 삽입된 SQL문에 대한 정보 수집』
- 77 페이지의 『데이터베이스 모니터를 사용하여 조회에 대한 통계 정보 수집』
- 88 페이지의 『메모리 상주 데이터베이스 모니터 API를 사용하여 조회에 대한 통계 수집』
- 93 페이지의 『Visual Explain을 사용하여 조회의 효율성 보기』
- 91 페이지의 『iSeries Navigator의 SQL 성능 모니터를 사용하여 데이터베이스 성능 모니터링』

조회 처리 제어

- 95 페이지의 『조회 속성 변경(CHGQRYA) 명령을 사용하여 조회 속성 변경』
- 95 페이지의 『조회 옵션 파일 QAQQINI를 사용하여 조회를 동적으로 제어』
- 104 페이지의 『iSeries용 DB2 UDB 예측 조회 감독자를 사용하여 장기 실행 조회 제어』
- 108 페이지의 『조회에 대한 병렬 처리 제어』

툴간 차이 비교

113 페이지의 『조회 최적화 툴: 비교 표』를 통해 다음 사항을 알 수 있습니다.

- 조회에 대해 각각의 툴이 생성할 수 있는 정보
- 프로세스에서 특정 툴이 조회를 분석하는 시기
- 조회를 향상시키기 위해 각각의 툴이 실행할 수 있는 타스크

추가 정보 및 기법

조회 최적화에 대한 경험이 있는 경우에는 37 페이지의 『일반적인 조회 최적화 추가 정보』를 참조할 수 있습니다.

또한, 다음 주제에서는 어플리케이션의 조회 성능 최적화를 위한 프로그래밍 추가 정보 및 기법을 설명합니다.

- 133 페이지의 제 6 장『데이터베이스 성능 향상을 위한 어플리케이션 설계 추가 정보』
- 141 페이지의 제 7 장『데이터베이스 성능 향상을 위한 프로그래밍 기법』
- 147 페이지의 제 8 장『일반 iSeries용 DB2 UDB 성능 고려사항』

SQL 어플리케이션의 성능 확인

다음 명령을 사용하여 SQL 어플리케이션의 성능을 확인할 수 있습니다.

DSPJOB

작업 표시(DSPJOB) 명령을 OPTION(*OPNF) 매개변수와 함께 사용하여 작업에서 실행 중인 어플리케이션이 사용하는 색인 및 표를 표시할 수 있습니다.

또한 DSPJOB를 OPTION(*JOBLCK) 매개변수와 함께 사용하여 오브젝트 및 행 잠금 경합을 분석할 수 있습니다. 즉, 잠기는 오브젝트 및 행과 잠금을 보류시키는 작업의 이름이 표시됩니다.

프로그램이 실행 중인 분리 레벨, 트랜잭션 도중 잠겨지는 행의 수, 지연 중인 DDL 기능을 표시하려면 DSPJOP 명령에 OPTION(*CMTCTL) 매개변수를 지정하십시오. 표시되는 분리 레벨은 디폴트 분리 레벨입니다. SQL 프로그램에 사용되는 실제 분리 레벨은 CRTSQLxxx 명령의 COMMIT 매개변수로 지정합니다.

PRTSQLINF

SQL 정보 인쇄(PRTSQLINF) 명령을 사용하면 프로그램에 삽입된 SQL문, SQL 패키지 또는 서비스 프로그램에 대한 정보를 인쇄할 수 있습니다. 이 정보에는 SQL문, 명령문 실행 도중 사용되는 액세스 계획 및 오브젝트에 대한 소스 멤버를 사전컴파일하는 데 사용되는 명령 매개변수 리스트가 포함됩니다. SQL문 관련 정보를 인쇄하는 것에 대한 자세한 정보는 75 페이지의『PRTSQLINF 명령을 사용하여 삽입된 SQL문에 대한 정보 수집』에 있는 PRTSQLINF 절을 참조하십시오.

STRDBMON

데이터베이스 모니터 시작(STRDBMON) 명령을 사용하여, 실행되는 모든 SQL문에 대한 정보를 파일로 캡처할 수 있습니다. 자세한 정보는 77 페이지의『데이터베이스 모니터를 사용하여 조회에 대한 통계 정보 수집』을 참조하십시오.

CHGQRYA

조회 속성 변경(CHGQRYA) 명령을 사용하여 조회 Optimizer에 대한 조회 속성을 변경할 수 있습니다. 이 명령을 사용하여 변경될 수 있는 속성 중에는 예상 조회 감독자, 병렬 및 조회 옵션이 있습니다.

STRDBG

디버그 시작(STRDBG) 명령을 사용하여 작업을 디버그 모드로 넣을 수 있고, 선택적으로, 20개의 프로그램과 20개의 클래스 파일 및 20개의 서비스 프로그램 만큼을 디버그 모드에 추가할 수 있습니다. 또한 디버깅 세션의 특정 속성을 지정합니다. 예를 들면, 기간계 라이브러리의 데이터베이스 파일이 디버그 모드에 있는 동안 갱신될 수 있는지 여부를 지정할 수 있습니다.

작업 기록부에서 조회 Optimizer 디버그 메세지 점검

조회 Optimizer의 디버그 메세지는 조회 구현에 관한 정보용 메세지를 작업 기록부에 발행합니다. 이 메세지는 조회 최적화 프로세스 도중 발생한 일을 설명합니다. 예를 들면, 다음과 같은 정보를 알 수 있습니다.

- 색인이 사용되었거나 사용되지 않은 이유
- 임시 결과가 필요한 이유
- 결합 및 블록화가 사용되는지 여부
- Optimizer가 제안한 색인의 유형
- 작업의 조회 상태
- 사용된 색인
- 커서의 상태

Optimizer는 SQL, 호출 레벨 인터페이스, ODBC, OPNQRYF 및 SQL 조회 관리자 등을 포함하여 최적화 대상이 되는 모든 조회에 대해 자동으로 메세지를 기록합니다.

디버그 메세지 보기

메세지를 보려면 다음 방법 중 하나를 사용하여 작업을 디버그 모드에 놓으십시오.

- 다음 명령을 사용하십시오.

  ```
  STRDBG PGM(Library/program) UPDPROD(*YES)
  ```

  STRDBG는 실행되는 모든 SQL문에 대한 정보를 작업 기록부에 위치시킵니다.
- QAQQINI 파일이 존재하는 사용자 라이브러리에 대해 조회 속성 변경(CHGQRYA) 명령에 QRYOPTLIB 매개변수를 설정하십시오. QAQQINI 파일의 매개변수를 MESSAGES_DEBUG로 설정하고 값을 *YES로 설정하십시오. 이 옵션은 조회 최적화 정보를 작업 기록부에 위치시킵니다.

명령 입력 패널에서 F10을 누르면 메세지 텍스트가 표시됩니다. 두 번째 레벨의 텍스트를 보려면 F1(도움말)을 누르십시오. 두 번째 레벨의 텍스트에서는 때로 조회 성능 향상을 위한 힌트가 제공되기도 합니다.

특정 디버그 메세지의 의미를 보려면 『조회 최적화 성능 정보 메세지』 및 73 페이지의 『조회 최적화 성능 정보 메세지 및 열린 자료 경로』를 참조하십시오.

iSeries Navigator를 사용하여 작업 기록부를 검토하는 데 관련한 정보는 SQL 프로그래밍 개념 책에서 작업 기록부 보기를 참조하십시오.

조회 최적화 성능 정보 메세지

데이터베이스 관리자가 작업 기록부에 기록한 정보용 메세지를 사용하여 프로그램에서 주어진 SQL문의 구조 및 성능을 평가할 수 있습니다. 이 메세지는 디버그 모드에서 실행시 SQL 프로그램 또는 대화식 SQL에 대해 발행됩니다. 데이터베이스 관리자는 적당한 경우에 다음 메세지를 송신할 수 있습니다. 앰퍼샌드 변수(&1, &X)는 작업 기록부에 메세지가 나타날 때 오브젝트명 또는 일부 다른 대체 값 중 하나가 들어 있는 대체 변수입니다. 메세지는 다음과 같습니다.

- 69 페이지의『CPI4321 - &1 파일에 대한 액세스 경로가 빌드되었습니다』
- 70 페이지의『CPI4322 - 키순 파일 &1로부터 액세스 경로가 빌드되었습니다』
- 70 페이지의『CPI4323 - OS/400 조회 액세스 계획이 리빌드되었습니다』
- 70 페이지의『CPI4324 - &1 파일에 대해 임시 파일이 빌드되었습니다』
- 71 페이지의『CPI4325 - 조회를 위해 임시 결과 파일이 빌드되었습니다』
- 71 페이지의『CPI4326 - &1 파일이 결합 위치 &11에서 처리되었습니다』
- 71 페이지의『CPI4327 - &13 파일이 결합 위치 &10에서 처리되었습니다』
- 71 페이지의『CPI4328 - &4 파일의 액세스 경로가 조회에 사용되었습니다』
- 71 페이지의『CPI4329 - &1 파일에 도달순 액세스가 사용되었습니다』
- 71 페이지의『CPI432A - &1 파일에 대한 조회 최적화 시간이 종료되었습니다』
- 72 페이지의『CPI432B - 부속 선택이 결합 조회로서 처리되었습니다』
- 72 페이지의『CPI432C - &1 파일에 대해 모든 액세스 경로가 고려되었습니다』
- 72 페이지의『CPI432D - 추가 액세스 경로 이유 코드가 사용되었습니다』
- 72 페이지의『CPI432E - 선택 열이 서로 다른 속성에 맵핑되었습니다』
- CPI432F 파일 &1에 대한 액세스 경로 제안사항.
- CPI4330 &1 파일의 병렬 &10 스캔에 &6 타스크가 사용되었습니다.
- CPI4331 &1 파일에 대해 작성된 병렬 색인에 &6 타스크가 사용되었습니다.
- CPI4332 조회에 &1 호스트 변수가 사용되었습니다.
- CPI4333 결합 처리에 해시 알고리즘이 사용되었습니다.
- CPI4334 조회가 재사용 가능한 ODP로서 구현되었습니다.
- CPI4335 해시 결합 단계 &1에 대한 Optimizer 디버그 메세지는 다음과 같습니다.
- CPI4336 그룹화 처리가 생성되었습니다.
- CPI4337 해시 결합 단계 &1에 대해 임시 해시 표가 빌드되었습니다.
- 72 페이지의『CPI4338 - &2 파일의 비트맵 처리에 &1 액세스 경로가 사용되었습니다』
- CPI4339 &1 라이브러리에서 조회 옵션이 검색되었습니다.
- CPI433A 조회 옵션 파일을 검색할 수 없습니다.
- CPI433C &1 라이브러리를 찾을 수 없습니다.
- CPI4341 분배 조회를 실행 중입니다.
- CPI4342 조회를 위한 분배 결합을 실행 중입니다.
- CPI4345 조회를 위해 임시 분배 결과 파일 &4가 빌드되었습니다.
- CPI4346 조회 결합 단계 &2의 &1에 대한 Optimizer 디버그 메세지는 다음과 같습니다.
- CPI4347 여러 단계에서 처리 중인 조회.
- CPI4348 커서와 연관된 ODP가 하드 닫기 상태입니다.
- CPI4349 호스트 변수 값의 빠른 이전 화면정리를 할 수 없습니다.

- CPI434A &1 조회 &2에 대한 Optimizer 디버그 메세지 시작.
- CPI434B &1 조회 &2에 대한 디버그 메세지 종료.
- CPI434C OS/400 조회 액세스 계획이 리빌드되지 않았습니다.
- 73 페이지의『SQL7910 - 모든 SQL 커서가 닫혔습니다』
- 74 페이지의『SQL7911 - ODP가 재사용되었습니다』
- 74 페이지의『SQL7912 - ODP가 작성되었습니다』
- 74 페이지의『SQL7913 - ODP가 삭제되었습니다』
- 74 페이지의『SQL7914 - ODP가 삭제되지 않았습니다』
- 74 페이지의『SQL7915 - SQL문에 대한 액세스 계획이 빌드되었습니다』
- 74 페이지의『SQL7916 - 조회에 블록화가 사용되었습니다』
- 74 페이지의『SQL7917 - 액세스 계획이 갱신되지 않았습니다』
- 75 페이지의『SQL7918 - 재사용할 수 있는 ODP가 삭제되었습니다』
- 75 페이지의『SQL7919 - FETCH 또는 삽입된 SELECT에 자료 변환이 요구되었습니다』
- 75 페이지의『SQL7939 - INSERT 또는 UPDATE에 자료 변환이 요구되었습니다』

이 메세지들은 조회가 실행된 방식에 대한 피드백을 제공하고, 일부 경우에는 조회가 더 빠르게 실행되도록 하기 위한 개선 사항을 표시합니다.

메세지에는 메세지의 원인, 오브젝트명 참조 및 가능한 사용자 응답에 대한 정보를 제공하는 메세지 도움말이 들어 있습니다.

메세지가 송신된 시기가 반드시 연관된 기능이 실행된 시기를 표시하는 것은 아닙니다. 일부 메세지는 조회 실행 시작시 한꺼번에 송신되기도 합니다.

다음 메세지의 원인 및 이에 대한 사용자 응답은 부연 설명이 제공됩니다. 실제 메세지 도움말은 보다 완전하며 각 메세지의 의미 및 이에 대한 응답을 판별하려고 하는 경우에 사용됩니다.

다음 섹션에서는 각 메세지에 대해 가능한 사용자 조치를 설명합니다.

CPI4321 - &1 파일에 대한 액세스 경로가 빌드되었습니다

이 메세지는 조회를 처리하기 위해 임시 색인이 작성되었음을 표시합니다. 지정된 표의 모든 행을 읽음으로써 신규 색인이 작성됩니다.

각 조회 실행시 색인 작성에 상당한 시간이 소요될 수 있습니다. 다음 조건을 만족하는 논리 파일(CRTLF) 또는 SQL 색인(CREATE INDEX SQL문)을 작성하십시오.

- 메세지 도움말에서 명명한 표에 대해
- 메세지 도움말에서 명명한 키 열 포함
- 메세지 도움말에서 지정한 오름차순 또는 내림차순 사용
- 메세지 도움말에서 지정한 정렬 순서표 사용

상수가 있는 조회의 술부와 일치하거나 부분적으로 일치하는 선택 또는 생략 범주를 사용하여 논리 파일을 작성할 것을 고려하십시오. 데이터베이스 관리자는 조회에서 선택 또는 생략 논리 파일의 사용이 명시적으로 지정되지 않은 경우에도 이 논리 파일의 사용을 고려합니다.

일부 조회의 경우에는 Optimizer가 기존 색인을 사용할 수 있을 때도 색인을 작성하기로 결정할 수 있습니다. 이러한 경우는 조회가 색인에 대한 키 열로서 순서화 열을 포함하며 지정된 유일한 행 선택이 서로 다른 열을 사용하는 경우에 발생합니다. 행 선택 결과 대략 20% 이상의 행이 리턴되면 Optimizer는 더 빨리 자료에 액세스하기 위해 새로운 색인을 작성할 수도 있습니다. 새로운 색인은 읽어야 하는 자료 양을 최소화합니다.

CPI4322 - 키순 파일 &1로부터 액세스 경로가 빌드되었습니다

이 메세지는 기존의 키순 표 또는 색인의 액세스 경로에서 임시 색인이 작성되었음을 나타냅니다.

일반적으로 이 조치는 표에 있는 자료의 서브세트만 읽어야 하기 때문에 시간이나 자원을 많이 사용하지 않습니다. 이 조치는 일반적으로 순서화, 그룹화 또는 결합 기준을 위해 색인을 작성하는 동안, Optimizer가 선택에 기존 색인을 사용할 수 있도록 하는 데 사용됩니다. 때로는 메세지 도움말에 언급된 색인 요구사항을 만족시키는 논리 파일 또는 SQL 색인을 작성함으로써 훨씬 더 빨리 액세스할 수도 있습니다.

자세한 내용은 앞에서 언급한 메세지 CPI4321을 참조하십시오.

CPI4323 - OS/400 조회 액세스 계획이 리빌드되었습니다

이 메세지가 송신되는 이유는 여러 가지가 있을 수 있습니다. 특정 이유는 메세지 도움말에서 제공됩니다.

대개는 조회되는 표의 환경이 변경되어 현재의 액세스 계획의 필요가 없어질 때 이 메세지가 송신됩니다. 표의 환경이 변경되는 예로는 조회에 필요한 색인이 서버에 없는 경우를 들 수 있습니다.

액세스 계획에는 조회 실행 방법에 대한 지침과 조회 실행을 위한 색인 리스트가 포함되어 있습니다. 필요한 색인을 더 이상 사용할 수 없는 경우에는 조회가 다시 최적화되고 새로운 액세스 계획이 작성되어 기존 계획을 대체합니다.

실행시 조회를 다시 최적화하고 새로운 액세스 계획을 빌드하는 프로세스는 iSeries용 DB2 UDB의 기능입니다. 이 기능을 통해 조회는 사용자가 간섭하지 않고도 데이터베이스의 가장 최근 상태를 사용하여 가능한 한 효율적으로 실행될 수 있습니다.

이 메세지가 드물게 나타나는 경우에는 조치가 필요하지 않습니다. 예를 들면 이 메세지는 복원 후 SQL 패키지가 처음으로 실행되는 경우나 내재 리빌드를 정당화하는 변경 사항(예: 새로운 색인이 작성된 경우)이 발생했음을 Optimizer가 감지하는 경우에 송신됩니다. 그러나 지나치게 많은 리빌드는 지나치게 조회 처리를 추가로 발생시킬 수 있으므로 되도록 피해야 합니다. 리빌드 횟수가 지나치게 많으면 어플리케이션 설계에 문제가 있거나 데이터베이스 관리 방법이 비효율적임을 나타냅니다. CPI434C를 참조하십시오.

CPI4324 - &1 파일에 대해 임시 파일이 빌드되었습니다

조회 처리가 시작되기 전에, 조회 실행을 단순화하기 위해 지정된 표의 자료가 임시 실제 표로 복사되어야 합니다. 메세지 도움말에는 이 메세지가 송신된 이유가 포함되어 있습니다.

지정된 표가 행을 거의 선택하지 않는 경우(보통 1,000행 미만), 조회 구현의 행 선택 부분은 자원과 시간을 많이 사용하지 않습니다. 그러나, 조회에 허용 가능한 것보다 많은 시간과 자원이 소요되는 경우에는 임시 표가 필요하지 않도록 조회를 변경하십시오.

이를 위한 한 가지 방법은 조회를 복수 단계로 구분하는 것입니다. 표에 필요한 행만 선택한 다음, 이 표의 행을 나머지 조회를 위해 사용하려면, INSERT문을 부속 선택과 함께 사용할 것을 고려하십시오.

CPI4325 - 조회를 위해 임시 결과 파일이 빌드되었습니다

조회의 중간 결과가 들어 있는 임시 결과표가 작성되었습니다. 결과는 내부 임시 표(구조)에 저장됩니다. 이는 결과를 처리하고 저장하는 방법에 있어 Optimizer가 제공하는 유연성을 사용할 수 있도록 합니다. 메세지 도움말에는 임시 결과표가 필요한 이유가 포함되어 있습니다.

일부 경우, 임시 결과표를 작성하면 조회를 더 빠르게 실행할 수 있습니다. 반면 임시 결과표로 복사되는 행 수가 많은 다른 조회의 경우에는 시간이 많이 소요될 수도 있습니다. 조회에 허용 가능한 것보다 많은 시간과 자원이 소요되는 경우에는 임시 결과표가 필요하지 않도록 조회를 변경하십시오.

CPI4326 - &1 파일이 결합 위치 &11에서 처리되었습니다

이 메세지는 표의 자료에 액세스하는 데 색인이 사용된 경우 지정된 표의 결합 위치를 알려 줍니다. **결합 위치**란 표들이 결합되는 순서와 관련되어 있습니다. 세부사항은 결합 최적화 섹션을 참조하십시오.

CPI4327 - &13 파일이 결합 위치 &10에서 처리되었습니다

이 메세지는 표 액세스 스캔 방식을 사용하여 표에서 행을 선택할 때 표의 이름 및 결합 위치를 알려줍니다.

결합 위치에 대한 정보 및 결합 성능 추가 정보는 앞서 언급한 메세지 CPI4326을 참조하십시오.

CPI4328 - &4 파일의 액세스 경로가 조회에 사용되었습니다

이 메세지는 조회에 사용된 기존 색인의 이름을 알려 줍니다.

색인이 사용된 이유가 메세지 도움말에 제공됩니다.

CPI4329 - &1 파일에 도달순 액세스가 사용되었습니다

지정된 표에 있는 자료에 액세스하는 데 어떤 색인도 사용되지 않았습니다. 행은 도달 순서에 따라 차례로 스캔되었습니다.

색인이 없는 경우에는 행 선택에 있는 열 중 하나와 일치하는 키 열이 포함된 색인을 하나 작성할 수 있습니다. 색인 작성은 행 선택(WHERE 절)이 표에서 20% 미만의 행을 선택하는 경우에만 적용해야 합니다. 강제로 기존 색인을 사용하도록 하려면, 조회의 ORDER BY절을 변경하여 색인의 첫 번째 키 열을 지정하거나, 조회가 첫 번째 I/O 환경에서 실행 중인지 확인하십시오.

CPI432A - &1 파일에 대한 조회 최적화 시간이 종료되었습니다

Optimizer는 조회 최적화에 소요된 시간이 조회 실행 추정 시간 및 조회된 표의 열 수에 상응하는 내부 값을 초과할 때 색인 고려를 중단합니다. 일반적으로는 표의 행 수가 많을수록 고려되는 색인 수도 많아집니다.

조회 실행 추정 시간이 초과되면, Optimizer는 더 이상 색인을 고려하지 않고 현재의 최적 방법을 사용하여 조회를 구현합니다. 최적의 성능을 얻기 위한 색인을 찾거나, 색인을 작성해야 합니다. 조회를 실행하기 위한 실제 시간이 추정 실행 시간을 초과할 경우, 이는 Optimizer가 최적 색인을 고려하지 않았다는 것을 가리킬 수 있습니다.

메세지 도움말에는 Optimizer가 시간 종료되기 전에 고려되었던 색인 리스트가 들어 있습니다. 이 색인 리스트를 보고, 최적 색인이 고려되기 전에 Optimizer가 시간 종료되었는지 판별할 수 있습니다.

최적화에 색인이 고려되도록 하려면 해당 색인과 연관된 논리 파일을 조회할 표로 지정하십시오. Optimizer는 조회 또는 SQL문에서 지정된 표의 색인을 먼저 고려합니다. 단, SQL 색인은 조회할 수 없습니다.

더 이상 필요 없는 색인은 삭제할 수 있습니다.

CPI432B - 부속 선택이 결합 조회로서 처리되었습니다

둘 이상의 SQL 부속 선택이 조회 Optimizer에 의해 결합되고 결합 조회로 처리되었습니다. 일반적으로 이 처리 방식은 성능이 우수한 옵션입니다.

CPI432C - &1 파일에 대해 모든 액세스 경로가 고려되었습니다

Optimizer가 지정된 표에 대해 빌드된 모든 색인을 고려했습니다. Optimizer가 표에 대한 모든 색인을 점검한 결과 표에 대한 현재 최상의 액세스 경로를 판별했습니다.

메세지 도움말에는 색인 리스트가 포함되어 있습니다. 각 색인과 함께 이유 코드도 추가되어 있습니다. 이유 코드는 색인이 사용되었거나 사용되지 않은 이유를 설명합니다.

CPI432D - 추가 액세스 경로 이유 코드가 사용되었습니다

이 메세지가 나타나기 바로 전에 메세지 CPI432A 또는 CPI432C가 발행되었습니다. 메세지의 길이 제한으로 인해 메세지 CPI432A 및 CPI432C에 사용되는 일부 이유 코드는 CPI432D의 메세지 도움말에서 설명합니다. 메세지 CPI432A 또는 CPI432C에서 리턴된 정보를 해석하려면 이 메세지의 도움말을 사용하십시오.

CPI432E - 선택 열이 서로 다른 속성에 맵핑되었습니다

이 메세지는 조회 Optimizer가 조회의 선택 스펙을 하나 이상 분석하는 데 색인의 사용을 고려하지 못했음을 표시합니다. 조회의 처리를 소수 행으로만 제한하기 위해 사용할 수 있는 색인이 있었던 경우에는 이 조회의 성능이 영향을 받습니다.

비교 값 및 비교 열의 속성은 일치해야 합니다. 그렇지 않으면 이 속성들이 일치하도록 하는 변환이 발생합니다. 일반적으로는 가장 작은 속성을 가지는 값이 다른 값의 속성에 맵핑되도록 하는 변환이 발생합니다. 비교 값의 속성과 호환되도록 비교 열의 속성을 맵핑해야 하는 경우에는 Optimizer가 이 선택을 구현하는 데 더 이상 색인을 사용할 수 없습니다.

CPI4338 - &2 파일의 비트맵 처리에 &1 액세스 경로가 사용되었습니다

Optimizer가 하나 이상의 색인을 조회 선택(WHERE 절)과 함께 사용하여 비트맵을 빌드하기로 선택합니다. 이 결과 비트맵은 실제로 선택됨을 표시합니다.

개념적으로 비트맵은 기준이 되는 표의 한 행마다 한 비트를 포함합니다. 선택된 행에 해당되는 비트는 '1'로 설정됩니다. 다른 모든 비트는 '0'으로 설정됩니다.

비트맵이 빌드되고 나면, 적절한 때에 조회에서 선택되지 않은 표의 행에 맵핑되는 것을 막기 위해 이 비트맵이 사용됩니다. 비트맵의 사용은 도달 순서 또는 1차 색인과의 조합에 비트맵이 사용되는지 여부에 따라 결정됩니다.

비트맵 처리가 도달 순서와 함께 사용되면 메세지 CPI4327 또는 CPI4329가 이 메세지보다 먼저 나타납니다. 이 경우, 비트맵은 조회가 선택한 표의 행만을 선택적으로 맵핑하는 데 도움이 됩니다.

비트맵 처리가 1차 색인과 함께 사용되면 메세지 CPI4326 또는 CPI4328이 이 메세지보다 먼저 나타납니다. 표에서 행을 맵핑하기 전에 1차 색인에 의해 선택된 행이 비트맵과 대조하여 검사됩니다. 세부사항은 비트맵 처리 액세스 방식을 참조하십시오.

조회 최적화 성능 정보 메세지 및 열린 자료 경로

다음의 SQL 실행 메세지 중 일부는 열린 자료 경로와 관계가 있습니다.

열린 자료 경로(ODP)란 커서가 열려 있거나 다른 SQL문이 실행될 때 작성된 내부 오브젝트를 말합니다. 열린 자료 경로는 자료에 대한 직접적 링크를 제공하여 I/O 조작이 발생할 수 있도록 합니다. ODP는 OPEN, INSERT, UPDATE, DELETE 및 SELECT INTO문에 사용되어 자료에 대한 각각의 조작을 실행합니다.

SQL 커서가 닫혀 있고 SQL문이 이미 실행된 경우에도 데이터베이스 관리자는 대부분의 경우 다음에 해당 명령문이 실행될 때 이를 재사용하기 위해 SQL 조작의 연관된 ODP를 저장합니다. 따라서 SQL CLOSE문이 SQL 커서를 닫더라도 ODP는 계속 사용할 수 있는 상태로 남아 다음에 커서가 열릴 때 다시 사용됩니다. 따라서 SQL문 실행에 소요되는 처리 및 응답 시간을 상당히 줄일 수 있습니다.

SQL문이 반복적으로 실행될 때 ODP를 재사용하는 기능은 더 빠른 성능을 얻는 데 있어 중요한 고려사항입니다.

SQL 실행시 발생되는 정보용 메세지는 다음과 같습니다.

SQL7910 - 모든 SQL 커서가 닫혔습니다

이 메세지는 작업의 호출 스택에 SQL문을 실행한 프로그램이 더 이상 존재하지 않을 때 송신됩니다.

CLOSQLCSR(*ENDJOB) 또는 CLOSQLCSR(*ENDACTGRP)이 지정되지 않은 경우, 프로그램 호출간에 ODP를 재사용하기 위한 SQL 환경은 SQL문을 실행한 활동 프로그램이 완료될 때까지만 존재합니다.

*ENDJOB 또는 *ENDACTGRP 커서와 연관된 ODP를 제외한 모든 ODP는 호출 스택에 있는 SQL 프로그램이 모두 완료되고 SQL 환경이 종료되는 때에 삭제됩니다.

이 완료 프로세스에는 커서 닫기, ODP 삭제, 준비된 명령문 제거 및 잠금 해제 등이 포함됩니다.

실행할 수 있는 SQL문을 어플리케이션의 첫 번째 프로그램에 배치하면 해당 어플리케이션의 기간 동안 SQL 환경이 활동 상태로 있게 됩니다. 그러면 프로그램이 반복적으로 호출되는 경우 다른 SQL 프로그램의 ODP가 재사용될 수 있습니다. CLOSQLCSR(*ENDJOB) 또는 CLOSQLCSR(*ENDACTGRP) 또한 지정할 수 있습니다.

SQL7911 - ODP가 재사용되었습니다

이 메세지는 명령문이 마지막으로 실행되었을 때나 해당 커서에 대해 CLOSE문이 실행되었을 때 ODP가 삭제되지 않았음을 표시합니다. ODP를 다시 사용할 수 있습니다. 이것은 불필요한 OPEN 및 CLOSE 조작을 제거함으로써 자원을 매우 효율적으로 사용할 수 있음을 표시합니 다.

SQL7912 - ODP가 작성되었습니다

다시 사용할 수 있는 ODP를 찾지 못했습니다. 명령문이 처음으로 실행된 경우나 프로세스에 대해 커서가 처음으로 열린 경우에는 항상 ODP가 작성되어야 합니다. 그러나, 이 메세지가 명령문을 실행하거나 커서를 열 때마다 나타나면 137 페이지의 『데이터베이스 어플리케이션 설계 추가 정보: 비ILE 프로그램 호출에 대해 커서 위치 보유』에서 권장하는 추가 정보를 해당 어플리케이션에 적용해야 합니다.

SQL7913 - ODP가 삭제되었습니다

작업마다 한 번씩만 실행되는 프로그램의 경우에는 이 메세지가 정상적입니다. 그러나, 이 메세지가 명령문을 실행하거나 커서를 열 때마다 나타나면 137 페이지의 『데이터베이스 어플리케이션 설계 추가 정보: 비ILE 프로그램 호출에 대해 커서 위치 보유』에서 권장하는 추가 정보를 해당 어플리케이션에 적용해야 합니다.

SQL7914 - ODP가 삭제되지 않았습니다

명령문을 재실행하거나 커서를 다시 열면 ODP를 다시 사용할 수 있어야 합니다.

SQL7915 - SQL문에 대한 액세스 계획이 빌드되었습니다

iSeries용 DB2 UDB 사전컴파일러는 필수 표가 누락된 경우에도 프로그램 오브젝트의 작성을 허용합니다. 이 경우, 액세스 계획의 바인딩은 프로그램이 처음으로 실행될 때 이루어집니다. 이 메세지는 액세스 계획이 작성되어 프로그램 오브젝트에 저장되었음을 표시합니다.

SQL7916 - 조회에 블록화가 사용되었습니다

SQL은 이 명령문을 실행할 때 한 번에 한 행을 요구하지 않고 데이터베이스 관리자에게 복수 행을 요구합니다.

SQL7917 - 액세스 계획이 갱신되지 않았습니다

데이터베이스 관리자가 해당 명령문에 대한 액세스 계획을 리빌드했는데도 프로그램이 신규 액세스 계획으로 갱신되지 않았습니다. 프로그램의 액세스 계획에 대한 공유 잠금이 적용된 프로그램을 다른 작업이 현재 실행하고 있습니다.

해당 작업이 프로그램의 액세스 계획에 대한 배타적 잠금을 획득해야만 신규 액세스 계획으로 프로그램을 갱신할 수 있습니다. 배타적 잠금은 공유 잠금이 해제되어야만 획득할 수 있습니다.

이 경우에도 명령문은 여전히 실행되고 신규 액세스 계획이 사용됩니다. 그러나 프로그램이 갱신되기 전까지는 명령문 실행시 액세스 계획이 계속해서 리빌드됩니다.

SQL7918 - 재사용할 수 있는 ODP가 삭제되었습니다

이 명령문에 대해 재사용 가능한 ODP가 존재하지만, 작업의 라이브러리 리스트 또는 대체 스펙이 조회를 변경했습니다.

명령문이 현재 기존 ODP에 있는 것과는 다른 표와 관련되어 있거나 다른 대체 스펙을 사용합니다. 기존 ODP는 재사용할 수 없으며 신규 ODP를 작성해야 합니다. ODP를 재사용하려면, 라이브러리 리스트 또는 대체 스펙을 변경하지 마십시오.

SQL7919 - FETCH 또는 삽입된 SELECT에 자료 변환이 요구되었습니다

자료를 호스트 변수에 맵핑할 때 자료 변환이 요구되었습니다. 명령문을 다음에 실행하면, 자료 변환이 요구되지 않았던 경우보다 속도가 더 느려집니다. 명령문은 실행되었지만 자료 변환을 제거하면 성능이 더 향상될 수 있습니다. 예를 들면, 특정 길이의 문자 스트링을 길이가 다른 호스트 변수 문자 스트링에 맵핑하는 것과 같은 자료 변환이 이 메세지의 발생 원인일 수 있습니다. 또한, 숫자 값을 유형이 다른 호스트 변수에 맵핑하는 경우(십진수를 정수로 맵핑하는 경우)에도 이 오류가 발생할 수 있습니다. 대부분의 변환을 방지하려면 페치되는 열과 유형 및 길이가 동일한 호스트 변수를 사용하십시오.

SQL7939 - INSERT 또는 UPDATE에 자료 변환이 요구되었습니다

INSERT 또는 UPDATE 값의 속성이 이 값을 수신하는 열의 속성과 다릅니다. 값을 변환해야 하기 때문에 이 값을 해당 열로 직접 이동할 수 없습니다. INSERT 또는 UPDATE 값의 속성이 이 값을 수신하는 열의 속성과 일치하면 성능이 향상될 수 있습니다.

PRTSQLINF 명령을 사용하여 삽입된 SQL문에 대한 정보 수집

PRTSQLINF 명령은 프로그램에 삽입된 SQL문, SQL 패키지(대개 리모트 조회에 대한 액세스 계획을 저장하는 데 사용되는 오브젝트) 또는 서비스 프로그램에 대한 정보를 수집합니다. 그런 다음 해당 정보를 스풀 파일에 보관합니다. PRTSQLINF는 다음에 대한 정보를 제공합니다.

- 실행되는 SQL문
- 실행 도중 사용되는 액세스 계획 유형. 여기에는 조회가 구현되는 방법, 사용된 색인, 결합 순서, 정렬 여부, 데이터베이스 스캔 사용 여부, 색인 작성 여부에 대한 정보가 포함됩니다.
- 오브젝트에 대한 소스 멤버를 사전컴파일하는 데 사용되는 명령 매개변수 리스트

이러한 정보를 수집하려면 저장된 액세스 계획에 대해 PRTSQLINF를 실행하거나 iSeries Navigator에서 PRTSQLINF 기능을 사용하십시오. 즉, 명령을 사용하기 전에 조회를 실행하거나 최소한 준비(SQL의 PREPARE 명령 사용)해야 합니다. PREPARE의 결과로 작성된 색인은 비교적 내용이 빈약하고 처음 실행 후 변경될 수 있으므로 조회를 실행하는 것이 가장 좋습니다. 저장된 액세스 계획에 대한 PRTSQLINF의 요구사항은 해당 명령을 OPNQRYF와 함께 재사용될 수 없음을 의미합니다.

PRTSQLINF가 디버그 메세지에서 볼 수 있는 정보와 유사한 내용을 출력하기는 하지만 저장된 액세스 계획에 대해서는 실행해야 합니다. 조회 Optimizer는 작업이 디버그 모드에 있는 경우에 현재의 조회 처리에 대한 정보 메세지를 자동으로 기록합니다. 따라서, PRTSQLINF가 소급하여 작동하는 데 반해 조회 디버그 메세지는 실행시 작동됩니다. 조회 감독자 조회 메세지 CPA4259의 2차 레벨 텍스트에서 이 정보를 볼 수도 있습니다. 메세지는 다음과 같습니다.

- SQL400A 결합 결과를 포함하기 위해 임시 분배 결과 파일 &1이(가) 작성되었습니다. 결과 파일이 지정되었습니다.
- SQL400B 결합 결과를 포함하기 위해 임시 분배 결과 파일 &1이(가) 작성되었습니다. 결과 파일이 브로드캐스트되었습니다.
- SQL400C 분배 조회 단계 &1 및 &2에 대해 Optimizer 디버그 메세지가 나타납니다.
- SQL400D GROUP BY 처리가 생성되었습니다.
- SQL400E 분배 부속 조회를 처리하는 동안 임시 분배 결과 파일 &1이(가) 작성되었습니다.
- SQL4001 임시 결과가 작성되었습니다.
- SQL4002 재사용 가능한 ODP 정렬이 사용되었습니다.
- SQL4003 UNION.
- SQL4004 SUBQUERY.
- SQL4005 조회 Optimizer가 표 &1에 대해 시간 종료되었습니다.
- SQL4006 표 &1에 대해 모든 색인이 고려되었습니다.
- SQL4007 결합 위치 &1 표 &2에 대한 조회 구현.
- SQL4008 표 &2에 대해 색인 &1이(가) 사용되었습니다.
- SQL4009 표 &1에 대해 색인이 작성되었습니다.
- SQL401A 분배 표가 있는 조회에 대해 그룹화 기준을 처리 중입니다.
- SQL401B 그룹화 기준을 처리하는 동안 임시 분배 결과 표 &1이(가) 작성되었습니다.
- SQL401C 조회에 대해 분배 결합을 수행 중입니다.
- SQL401D 표 &2이(가) 지정되었으므로 임시 분배 결과 표 &1이(가) 작성되었습니다.
- SQL401E 표 &2이(가) 브로드캐스트되었으므로 임시 분배 결과 표 &1이(가) 작성되었습니다.
- SQL401F 표 &1이(가) 분배 결합에 사용되었습니다.
- SQL4010 표 &1에 대한 표 스캔 액세스.
- SQL4011 색인 스캔 키 행 위치지정이 표 &1에서 사용되었습니다.
- SQL4012 표 &2에 대한 색인 &1에서 색인이 작성되었습니다.
- SQL4013 액세스 계획이 빌드되지 않았습니다.
- SQL4014 &1 결합 열 쌍이 이 결합 위치에 대해 사용되었습니다.
- SQL4015 시작 컬럼 &1.&2, 종료 컬럼 &3.&4, 결합 연산자 &5, 결합 술부 &6.
- SQL4016 부속 선택이 결합 조회로 수행되었습니다.
- SQL4017 호스트 변수가 재사용 가능한 ODP로 구현되었습니다.
- SQL4018 호스트 변수가 재사용할 수 없는 ODP로 구현되었습니다.
- SQL4019 호스트 변수가 파일 관리 행 위치지정 재사용 가능 ODP로 구현되었습니다.
- SQL402A 해싱 알고리즘이 결합을 처리하는 데 사용되었습니다.
- SQL402B 표 &1이(가) 결합 단계 &2에서 사용되었습니다.

- SQL402C 해시 결합 결과를 위해 임시 표가 작성되었습니다.
- SQL402D 라이브러리 &1의 조회 옵션 파일 &2에서 조회 속성이 대체되었습니다.
- SQL4020 예상 조회 실행 시간은 &1 초입니다.
- SQL4021 액세스 계획이 마지막에 &2에 &1에 저장되었습니다.
- SQL4022 액세스 계획이 SRVQRY 속성이 활동 상태로 저장되었습니다.
- SQL4023 병렬 표 사전페치가 사용되었습니다.
- SQL4024 병렬 색인 사전로드 액세스 방식이 사용되었습니다.
- SQL4025 병렬 표 사전로드 액세스 방식이 사용되었습니다.
- SQL4026 색인만 액세스가 표 번호 &1에서 사용되었습니다.
- SQL4027 액세스 계획이 시스템에 설치된 DB2 UDB 대칭 다중처리와 함께 저장되었습니다.
- SQL4028 조회에 분배 표가 들어 있습니다.
- SQL4029 그룹화를 처리하기 위해 해싱 알고리즘이 사용되었습니다.
- SQL4030 표 &2의 병렬 스캔을 위해 &1 타스크가 지정되었습니다.
- SQL4031 표 &2에서 병렬 색인 작성을 위해 &1 타스크가 지정되었습니다.
- SQL4032 색인 &1이(가) 표 &2의 비트맵 처리에 사용되었습니다.
- SQL4033 &1 타스크가 &2을(를) 사용하여 병렬 비트맵 작성을 위해 지정되었습니다.
- SQL4034 결합을 처리하기 위해 복수 결합 클래스가 사용되었습니다.
- SQL4035 표 &1이(가) 결합 클래스 &2에서 사용되었습니다.

데이터베이스 모니터를 사용하여 조회에 대한 통계 정보 수집

데이터베이스 모니터 통계 정보에서는 조회에 대한 가장 완벽한 정보를 얻을 수 있습니다. 서버의 특정 조회 또는 모든 조회에 대해 성능 통계 정보를 수집할 수 있습니다. 통계 정보를 수집하는 몇 가지 방법은 다음과 같습니다.

- 79 페이지의 『데이터베이스 모니터 시작(STRDBMON) 명령』 및 79 페이지의 『데이터베이스 모니터 종료(NDDBMON) 명령』 사용
- 91 페이지의 『iSeries Navigator의 SQL 성능 모니터를 사용하여 데이터베이스 성능 모니터링』 사용
- 성능 모니터 시작(STRPFRMON) 명령을 STRDBMON 매개변수와 함께 사용
- 메모리 상주 데이터베이스 모니터 API 사용. 88 페이지의 『메모리 상주 데이터베이스 모니터 API를 사용하여 조회에 대한 통계 수집』을 참조하십시오.

데이터베이스 모니터 사용에 대한 예는 82 페이지의 『데이터베이스 모니터 예』을 참조하십시오.

주: 데이터베이스 모니터는 사용될 경우 상당한 CPU 및 디스크 기억장치 오버헤드를 생성할 수 있습니다.

서버의 특정 작업 또는 모든 작업을 모니터할 수 있습니다. 수집된 통계 정보는 명령에서 지정한 출력 데이터베이스 표에 위치됩니다. 두 개의 모니터를 다음과 같이 사용하여 서버의 각 작업을 동시에 모니터할수도 있습니다.

- 하나의 모니터는 특정의 한 작업에 대해서만 시작되도록 합니다.
- 또 하나의 모니터는 서버의 모든 작업에 대해 시작되도록 합니다.

두 개의 모니터가 한 작업을 모니터할 때는 각 모니터가 서로 다른 출력표에 행을 기록합니다. 출력 데이터베이스 표에 있는 행은 각 행의 고유 식별 번호로 식별할 수 있습니다.

수집할 수 있는 통계 자료의 종류

데이터베이스 모니터는 조회 Optimizer 디버그 메세지(STRDBG) 및 SQL 정보 인쇄(PRTSQLINF) 명령으로 제공되는 것과 동일한 정보를 제공합니다. 다음은 데이터베이스 모니터에 의해 수집되는 추가 정보의 샘플입니다.

- 시스템 및 작업명
- SQL문 및 부속 선택 번호
- 시작 및 끝 시간소인
- 예상 처리 시간
- 조회 대상 표의 총 행 수
- 선택된 행의 수
- 선택된 예상 행 수
- 결합된 예상 행 수
- 제안된 색인에 대한 키 열
- 총 최적화 시간
- 결합 유형 및 방법
- ODP 구현

성능 통계 사용 방법

성능 통계를 사용하여 여러 가지 보고서를 생성할 수 있습니다. 예를 들면, 보고서에서 다음과 같은 조회를 표시할 수 있습니다.

- 많은 서버 자원을 사용하는 조회
- 실행 시간이 매우 긴 조회
- 조회 감독자의 시간 제한으로 인해 실행되지 않은 조회
- 실행 중 임시 색인을 작성하는 조회
- 실행 중 조회 정렬을 사용하는 조회
- 조회 Optimizer가 제안한 키가 포함되는 키순 논리 파일을 작성할 경우 더 빨리 실행될 수 있었던 조회

주: 종료 요구로 취소된 조회는 일반적으로 완전한 성능 통계 세트를 생성하지 않습니다. 단, 조회가 런타임 또는 다단계 조회 정보의 예외와 함께, 최적화된 방법에 대한 모든 정보를 포함합니다.

데이터베이스 모니터 시작(STRDBMON) 명령

STRDBMON 명령은 서버의 특정 작업 또는 모든 작업에 대해 데이터베이스 성능 통계 수집을 시작합니다. 수집된 통계는 명령에서 지정한 출력 데이터베이스 표 멤버에 위치됩니다. 출력 표 또는 멤버가 없는 경우에는 모델 표 QSYS/QAQQDBMN의 표 및 형식 정의를 기준으로 작성됩니다. 출력 표 또는 멤버가 있는 경우, 출력 표의 행 형식은 QQQDBMN이라는 이름을 가져야 합니다.

행을 기록하기 전에 정보의 멤버를 지우거나 기존 표의 끝에 새 정보를 첨부할 수 있도록 대체/첨부 옵션을 지정할 수 있습니다.

또한, 행이 출력 표에 기록되도록 강제하기 전에 모니터되는 각 작업의 행 버퍼에 보유되는 행 수를 제어할 수 있도록 강제 행 기록 옵션을 지정할 수도 있습니다. FRCRCD(1)과 같이 강제 행 기록 값을 1로 지정하면 모니터 행이 작성되자마자 기록부에 나타납니다. FRCRCD(1)은 또한 행의 실제 순서가 시간 순서가 되도록 할 가능성이 가장 큽니다(반드시 그런 것은 아님). 그러나, FRCRCD(1)은 모니터되는 작업의 성능에 가장 부정적인 영향을 미칠 수도 있습니다. FRCRCD 매개변수 값을 크게 지정하면 모니터 성능에 미치는 영향이 줄어들 수 있습니다.

STRDBMON 명령의 TYPE 매개변수에 *DETAIL을 지정하면 요약 행뿐만 아니라 세부 행도 수집되게 됩니다. 이는 비SQL 조회 즉, QQQ1000 행을 생성하지 않는 조회의 경우에만 유용합니다. 비SQL 조회에서 리턴되는 행 수 및 해당 행을 리턴하는 데 걸린 총 시간을 판별하는 유일한 방법은 세부 행을 수집하는 것입니다. 현재 유일한 세부 행은 153 페이지의 부록 A『데이터베이스 모니터: DDS』에 있는 QQQ3019입니다. 세부 행에는 귀중한 정보가 들어 있긴 하지만, 리턴된 각 행 블록에 대해 약간의 성능을 저하시킵니다. 따라서 세부 행을 사용할 때는 주의깊게 모니터해야 합니다.

모든 작업에 대해 모니터가 시작되는 경우에는, 작업 대기행렬에서 대기하고 있는 작업이나 모니터 기간 동안 시작된 작업에 대해 작업이 시작되자마자 수집된 통계가 생깁니다. 특정 작업에 대해 모니터가 시작되는 경우에는 명령 발행시 해당 작업이 서버에서 활동하고 있어야 합니다. 다음과 같이 두 개의 모니터만으로 동시에 서버의 각 작업을 모니터할 수 있습니다.

- 특정의 한 작업에 대해서만 시작되는 모니터
- 서버의 모든 작업에 대해 시작되는 모니터

두 개의 모니터가 한 작업을 모니터하고 각 모니터가 서로 다른 출력표에 기록할 때는 모니터 행이 해당 작업에 대한 두 기록부 모두에 기록됩니다. 두 모니터가 동일한 출력표를 선택한 경우, 모니터 행이 출력표에 중복 기록되지는 않습니다.

데이터베이스 모니터 종료(NDDBMON) 명령

ENDDBMON 명령은 서버의 특정 작업 또는 모든 작업에 대한 데이터베이스 모니터를 종료합니다. 모든 작업에 대한 모니터를 종료하려고 하는 경우에는 모든 작업에 대해 STRDBMON 명령을 먼저 발행한 상태이어야 합니다. 이 명령에서 특정 작업을 지정하는 경우에는 해당 작업에 대해서만 특별히 명시적으로 시작된 모니터가 있어야 합니다.

예를 들어, 다음 이벤트 순서를 살펴 보십시오.

1. 서버의 모든 작업에 대해 모니터가 시작됩니다.
2. 특정 작업에 대해 모니터가 시작됩니다.
3. 모든 작업에 대해 모니터가 종료됩니다.

이 순서에서 특정 작업 모니터는 모니터의 명시적인 시작이 해당 작업에 대해 이루어졌기 때문에 계속 실행됩니다. 이 모니터는 해당 특정 작업에 대한 ENDDBMON 명령이 발행될 때까지 계속 실행됩니다.

다음 순서를 살펴 보십시오.

1. 서버의 모든 작업에 대해 모니터가 시작됩니다.
2. 특정 작업에 대해 모니터가 시작됩니다.
3. 특정 작업에 대해 모니터가 종료됩니다.

이 순서에서 모든 작업에 대한 모니터는 특정 작업에 대한 모니터가 시작되는 경우에도 모든 작업에 대한 ENDDBMON 명령이 발행될 때까지 계속 실행됩니다.

다음 순서를 살펴 보십시오.

1. 특정 작업에 대해 모니터가 시작됩니다.
2. 서버의 모든 작업에 대해 모니터가 시작됩니다.
3. 모든 작업에 대해 모니터가 종료됩니다.

이 순서에서 특정 작업에 대한 모니터는 해당 작업에 대한 ENDDBMON 명령이 발행될 때까지 계속 실행됩니다.

다음 순서를 살펴 보십시오.

1. 특정 작업에 대해 모니터가 시작됩니다.
2. 서버의 모든 작업에 대해 모니터가 시작됩니다.
3. 특정 작업에 대해 모니터가 종료됩니다.

이 순서에서는 특정 작업을 포함한 모든 작업에 대해 모니터가 계속 실행됩니다.

모든 작업에 대한 모니터가 종료되면 서버의 모든 작업이 트리거되어 출력표를 닫지만, 모니터되는 작업 모두가 최종 성능 행을 기록부에 기록하기 전에 ENDDBMON 명령이 완료될 수 있습니다. 모니터링이 완료되기 전에 모니터된 모든 작업이 출력표에서 더 이상 잠금 상태를 유지하지 않는지 확인하려면 WRKOBJLCK(오브젝트 잠금에 대한 작업) 명령을 사용하십시오.

데이터베이스 모니터 성능 행

데이터베이스 표의 행은 행 식별 번호로 고유하게 식별됩니다. 파일 기반의 모니터(STRDBMON)에 있는 정보는 부록 A에 정의되어 있는 논리 형식 세트를 기준으로 하여 작성됩니다. 이 논리 형식은 디버그 메세지 및 PRSQLINF 메세지와 밀접하게 관련됩니다. 이 부록에는 각 논리 형식에 대해 어느 실제 열이 사용되는지 및 어떤 정보가 들어 있는지도 나타나 있습니다. 논리 형식을 사용하여 모니터로부터 추출할 수 있는 정보를 식별할 수 있습니다. 이 행들은 여러 개의 서로 다른 논리 파일에서 정의되는 데 이 파일은 서버 함께 제공되지 않

으므로 원하는 경우에는 사용자가 직접 작성해야 합니다. 논리 파일은 162 페이지의 『선택적 데이터베이스 모니터 논리 파일 DDS』에 표시된 DDS를 사용하여 작성할 수 있습니다. 이 표에서는 각 숫자 다음에 열 설명이 나옵니다.

주: 데이터베이스 모니터 논리 파일은 일부 선택/생략 기준이 들어 있는 키순 논리 파일입니다. 따라서 데이터베이스 모니터가 활동 중인 동안 이 표와 연관된 일부 유지보수 오버헤드가 있습니다. 사용자는 데이터베이스 모니터가 활동하는 동안, 특히 모든 작업을 모니터링할 경우, 이러한 오버헤드를 최소화시키려 할 수 있습니다. 모든 작업을 모니터할 때, 생성되는 행 수가 아주 많을 수 있습니다. 결과를 처리하는 데는 논리가 필요하지 않습니다. 단순히 표에 대한 정보의 추출을 보다 쉽고 직접적으로 만듭니다.

유지보수 오버헤드 최소화

데이터베이스 모니터 논리 파일과 연관된 유지보수 오버헤드를 최소화할 수 있는 방법은 다음과 같습니다.

- 데이터베이스 모니터가 완료될 때가지 데이터베이스 모니터 논리 파일을 작성하지 않습니다.
- 동적 선택/생략 범주를 사용(논리 파일의 DDS에서 DYNSLT 키워드 사용)하여 데이터베이스 모니터 논리 파일을 작성합니다.
- CRTLF 명령에 리빌드 색인 유지보수를 지정(MAINT 매개변수에 *REBLD 옵션 지정)하여 데이터베이스 모니터 논리 파일을 작성합니다.

실행시 유지보수 오버헤드를 최소화하면 데이터베이스 모니터 논리 파일이 작성되거나 열릴 때까지의 유지보수 비용만 지연시키게 됩니다. 데이터베이스 모니터가 활동 중인 동안 시간을 소모할 것인지 또는 데이터베이스 모니터가 완료된 후에 소모할 것인지 여부에 대한 선택이기 때문입니다.

조회 Optimizer 색인 어드바이저

조회 Optimizer는 조회의 행 선택을 분석하고, 디폴트 값을 기준으로 영구 색인을 작성하면 성능이 향상될지 여부를 판별합니다. 영구 색인이 성능을 향상시킬 수 있다고 판단되면 Optimizer는 제안된 색인을 작성하는 데 필요한 키 열을 리턴합니다.

색인 어드바이저 정보는 데이터베이스 모니터 논리 파일 QQQ3000, QQQ3001 및 QQQ3002에서 찾을 수 있습니다. 어드바이저 정보는 열 QQIDXA, QQIDXK 및 QQIDXD에 저장됩니다. QQIDXA 열에 'Y'라는 값이 들어 있으면 Optimizer가 열 QQIDXD에 표시된 키 열을 사용하여 색인을 작성하라고 조언하는 것입니다. 이 색인을 작성하는 목적은 조회 성능을 향상시키기 위해서 입니다.

열 QQIDXD에 들어 있는 키 열 리스트에는 Optimizer가 제안하는 1차 및 2차 키 열이 나열되어 있습니다. 1차 키 열은 해당 조회 선택을 기준으로 선택되는 키 수를 크게 줄이는 열입니다. 2차 키 열은 선택되는 키의 수를 줄일 수도 있고 줄이지 않을 수도 있는 열입니다.

Optimizer는 1차 키 열에 한개의 추가 2차 키 열을 더한 조합에 대해 색인 스캔 키 위치지정을 수행할 수 있습니다. 따라서, 첫 번째 2차 키 열은 선택 가능성이 가장 높은 2차 키 열이어야 합니다. Optimizer는 나머지 2차 키 열 중 하나와 색인 스캔 키 선택을 사용합니다. 색인 스캔 키 선택은 색인 스캔 키 위치지정만큼 빠르지는 않지만 선택되는 행 수를 줄일 수 있습니다. 그러므로 선택 가능성이 높은 2차 키 열이 포함되어야 합니다.

열 QQIDXK에는 열 QQIDXD에서 나열된 1차 키 열 수가 들어 있습니다. 이는 가장 왼쪽에서 제안된 키 열입니다. 나머지 키 열은 2차 키 열로 간주되고 해당 조회를 기준으로 한 예상 선택 가능성 순서로 열거됩니다. 예를 들어, QQIDXK에 4라는 값이 들어 있고 QQIDXD가 7개의 키 열을 지정하면 QQIDXK에서 지정한 첫 번째 4개의 키 열이 1차 키 열이 됩니다. 나머지 3개의 키 열은 제안된 2차 키 열이 됩니다.

2차 키 열의 진짜 선택 가능성을 판별하고 색인 작성 시 해당 키 열을 포함해야 하는지 여부를 판별하는 것은 사용자가 해야 할 일입니다. 색인을 빌드할 때 1차 키 열은 가장 왼쪽의 키 열이 되어야 하고, 그 다음에는 사용자가 선택한 2차 키 열 중 하나가 와야 하며, 선택 가능성별로 우선순위가 매겨져야 합니다. 조회 Optimizer 색인 어드바이저는 수동으로 쉽게 디버그될 수 없는 조회 내의 복합 선택 분석을 돕기 위해서만 사용되어야 합니다.

주: 제안된 색인을 작성하고 조회를 다시 실행한 후 조회 Optimizer가 제안된 색인을 사용하지 않기로 하는 경우도 있습니다. 선택 기준은 조회 Optimizer의 고려 대상이 되지만, 결합, 순서화 및 그룹화 기준은 그렇지 않습니다.

데이터베이스 모니터 예

사용자가 SQL문을 사용하는 어플리케이션 프로그램을 갖고 있고 해당 조회를 분석하고 조회의 성능을 조정하려 한다고 가정합시다. 성능 분석의 첫 번째 단계는 자료를 수집하는 것입니다. 다음 예는 STRDBMON 및 ENDDBMON을 사용하여 자료를 수집하고 분석하는 방법을 보여 줍니다.

성능 자료는 현재 작업에서 실행 중인 어플리케이션에 대해 LIB/PERFDATA에서 수집됩니다. 성능 자료를 수집하고 분석을 준비하는 순서는 다음과 같습니다.

1. STRDBMON FILE(LIB/PERFDATA). 이 표가 아직 없는 경우에는 QSYS/QAQQDBMN의 골격 표에서 표 하나가 작성됩니다.
2. 어플리케이션을 실행합니다.
3. ENDDBMON
4. 162 페이지의 『선택적 데이터베이스 모니터 논리 파일 DDS』에 표시된 DDS를 사용하여 LIB/PERFDATA에 대한 논리 파일을 작성합니다. 논리 파일을 반드시 작성해야 하는 것은 아닙니다. 모든 정보는 STRDBMON 명령에서 지정한 기본 표에 있기 때문입니다. 논리 파일은 단지 자료를 쉽게 볼 수 있도록 해줄 뿐입니다.

위의 단계를 마치면 자료 분석을 할 수 있습니다. 다음 예에서는 이러한 자료를 사용하는 방법을 설명합니다. 어플리케이션에 가장 적합한 정보를 제공하는 조회를 작성하려면 실제 및 논리 파일 DDS에 대해 충분한 지식을 습득하여 수집되는 모든 자료를 이해해야 합니다.

데이터베이스 모니터 성능 분석 예 1

표 스캔으로 SQL 어플리케이션에서 구현되는 조회가 어떤 조회인지를 판별하십시오. 두 개의 논리 파일, 즉 SQL문에 대한 정보가 들어 있는 QQQ1000과, 조회 성능 표 스캔에 대한 자료가 들어 있는 QQQ3000을 결합하면 정확한 정보를 얻을 수 있습니다. 다음 SQL 조회를 사용할 수 있습니다.

```
SELECT A.QQTLN, A.QQTFN, A.QQTOTR, A.QQIDXA, C.QQrcdr,
       (B.QQETIM - B.QQSTIM) AS TOT_TIME, B.QQSTTX
  FROM   LIB/QQQ3000 A, LIB/QQQ1000 B, LIB/QQQ3019 C
```

```
  WHERE  A.QQJFLD = B.QQJFLD
    AND  A.QQUCNT = B.QQUCNT
AND A.QQJFLD = C.QQJFLD AND A.QQUCNT = C.QQUCNT
```

이 조회의 출력 샘플은 표 3에 있습니다. 이 예에서 키는 다음의 결합 범주입니다.

```
  WHERE  A.QQJFLD = B.QQJFLD
    AND  A.QQUCNT = B.QQUCNT
```

여러 조회에 대한 많은 자료가 표 LIB/PERFDATA의 여러 행에 들어 있습니다. 단일 조회에 대한 자료가 표에서 10개 이상의 행에 들어 있는 것이 일반적입니다. 논리 파일을 정의한 다음 해당 표들을 함께 결합하는 조합을 사용하면 하나 이상의 조회에서 모든 자료를 함께 결합할 수 있습니다. 열 QQJFLD는 작업에 공통인 모든 자료 명령을 고유하게 식별하고 열 QQUCNT는 조회 레벨에서 고유합니다. 논리 파일의 컨텍스트에서 참조할 때 이 둘을 조합하면 조회 구현이 조회 명령문 정보에 연결됩니다.

표 3. 표 스캔을 실행한 SQL 조회의 출력

라이브러리 명	표 이름	총 행 수	색인 제안	리턴된 행	TOT_TIME	명령문 텍스트
LIB1	**TBL1**	**20000**	**Y**	**10**	**6.2**	`SELECT * FROM LIB1/TBL1 WHERE FLD1 = 'A'`
LIB1	**TBL2**	**100**	**N**	**100**	**0.9**	`SELECT * FROM LIB1/TBL2`
LIB1	**TBL1**	**20000**	**Y**	**32**	**7.1**	`SELECT * FROM LIB1/TBL1 WHERE FLD1 = 'B' AND FLD2 > 9000`

SQL을 사용하지 않는 조회의 경우에는 SQL 정보 행(QQQ1000)이 작성되지 않습니다. 그러면 LIB/PERFDATA의 어떤 행이 어떤 조회에 적합한지를 판별하기가 어려워집니다. SQL을 사용하는 경우에는 해당 조회에 대해 성능 행과 일치하는 실제 SQL문 텍스트가 행 QQQ1000에 들어 있습니다. 명령문 텍스트는 SQL을 통해서만 캡처할 수 있습니다. OPNQRYF 명령을 사용하여 실행된 조회의 경우에는 OPNID 매개변수가 캡처되어 조회에 행을 연결하는 데 사용할 수 있습니다. OPNID는 행 QQQ3014의 열 QQOPID에 들어 있습니다.

데이터베이스 모니터 성능 분석 예 2

어떤 SQL 어플리케이션이 표 스캔을 사용하여 구현되는지를 보여 준 앞의 예에서와 마찬가지로, 다음 예는 표 스캔을 사용하여 구현되는 모든 조회를 보여 줍니다.

```
SELECT A.QQTLN, A.QQTFN, A.QQTOTR, A.QQIDXA,
       B.QQOPID, B.QQTTIM, C.QQCLKT, C.QQRCDR, D.QQROWR,
       (D.QQETIM - D.QQSTIM) AS TOT_TIME, D.QQSTTX
FROM   LIB/QQQ3000 A INNER JOIN LIB/QQQ3014 B
         ON (A.QQJFLD = B.QQJFLD AND
             A.QQUCNT = B.QQUCNT)
       LEFT OUTER JOIN LIB/QQQ3019 C
         ON (A.QQJFLD = C.QQJFLD AND
             A.QQUCNT = C.QQUCNT)
       LEFT OUTER JOIN LIB/QQQ1000 D
       ON (A.QQJFLD = D.QQJFLD AND
           A.QQUCNT = D.QQUCNT)
```

이 예에서 표 스캔을 실행한 모든 조회의 출력은 84 페이지의 표 4에 표시되어 있습니다.

주: SQL을 사용하여 실행된 조회가 아닌 경우, 표 QQQ1000에서 선택된 열은 널(null) 디폴트 값을 리턴합니다. 이 예에서는 문자 자료에 대한 디폴트 값이 공백이고 숫자 자료에 대한 디폴트 값이 별표(*)라고 가정합니다.

표 4. 표 스캔을 실행한 모든 조회의 출력

라이브러리명	표 이름	총 행 수	색인 제안	조회 OPNID	ODP 열기 시간	시계 시간	리턴된 레코드 수	리턴된 행 수	TOT_TIME	명령문 텍스트
LIB1	**TBL1**	**20000**	**Y**		**1.1**	**4.7**	**10**	**10**	**6.2**	`SELECT * FROM LIB1/TBL1 WHERE FLD1 = 'A'`
LIB1	**TBL2**	**100**	**N**		**0.1**	**0.7**	**100**	**100**	**0.9**	`SELECT * FROM LIB1/TBL2`
LIB1	**TBL1**	**20000**	**Y**		**2.6**	**4.4**	**32**	**32**	**7.1**	`SELECT * FROM LIB1/TBL1 WHERE FLD1 = 'A' AND FLD2 > 9000`
LIB1	**TBL4**	**4000**	**N**	**QRY04**	**1.2**	**4.2**	**724**	*****	*****	*****

SQL문 텍스트가 필요없는 경우에는 표 QQQ1000에 대한 결합이 필요없습니다. QQQ3014 및 QQQ3019 행의 자료로 선택한 총 시간과 행을 판별할 수 있습니다.

데이터베이스 모니터 성능 분석 예 3

다음 단계에는 표 스캔 자료에 대한 추가 분석이 포함될 수 있습니다. 앞의 예 1과 2에는 색인 제안이라는 제목의 열이 들어 있습니다. 이 열에 있는 Y(예)는 조회 Optimizer에서 얻은 것으로서, 색인을 사용하여 자료에 액세스하면 조회가 보다 효율적으로 실행될 수 있음을 의미합니다. 색인을 참조하는 조회의 경우에는, 조회를 통해 선택되는 행의 수가 표의 총 행 수보다 적습니다. 이것은 표 스캔이 최적의 방법이 아닐 수도 있음을 의미합니다. 결국, 실행 시간이 길면 성능 조정을 통해 성능을 향상시킬 수 있는 조회가 강조표시됩니다.

다음의 논리 단계는 Optimizer가 제시하는 제안된 색인을 보기 위한 것입니다. 여기에는 다음과 같은 조회가 사용될 수 있습니다.

```
SELECT A.QQTLN, A.QQTFN, A.QQIDXA, A.QQIDXD,
       A.QQIDXK, B.QQOPID, C.QQSTTX
FROM   LIB/QQQ3000 A INNER JOIN LIB/QQQ3014 B
         ON (A.QQJFLD = B.QQJFLD AND
             A.QQUCNT = B.QQUCNT)
         LEFT OUTER JOIN LIB/QQQ1000 C
         ON (A.QQJFLD = C.QQJFLD AND
             A.QQUCNT = C.QQUCNT)
WHERE  A.QQIDXA = 'Y'
```

이 예는 예 1의 내용 중 두 부분을 약각 수정한 것입니다. 첫째, 선택된 열 수가 변경되었습니다. 가장 중요한 것은, 조회 Optimizer가 제안한 색인을 작성할 때 사용할 수 있는 키 열 리스트가 들어 있는 열 QQIDXD 선택입니다. 둘째, 조회 선택이 출력을 Optimizer가 색인 작성을 제안(A.QQIDXA = ‘Y’)한 표 스캔 조회로만 제한하고 있습니다. 이 조회로 출력되는 결과의 예는 85 페이지의 표 5와 같습니다.

표 5. 권장된 키 열이 있는 출력

라이브러리 명	표 이름	색인 제안	제안된 키 열	제안된 1차 키	조회 OPNID	명령문 텍스트
LIB1	**TBL1**	**Y**	**FLD1**	**1**		`SELECT * FROM LIB1/TBL1 WHERE FLD1 = 'A'`
LIB1	**TBL1**	**Y**	**FLD1, FLD2**	**1**		`SELECT * FROM LIB1/TBL1 WHERE FLD1 = 'B' AND FLD2 > 9000`
LIB1	**TBL4**	**Y**	**FLD1, FLD4**	**1**	**QRY04**	

이런 경우에는 Optimizer가 제안한 영구 색인을 작성하는 것이 합리적인지 여부를 사용자가 판별해야 합니다. 이 예에서는 세 가지 조회가 각각 FLD1의 1차 또는 가장 왼쪽 키 열을 사용하기 때문에 LIB1/TBL1에 대한 색인을 하나 작성하면 세 가지 조회를 모두 만족시킬 수 있습니다. 키 열 FLD1, FLD2를 사용하여 LIB1/TBL1에 대한 색인을 하나 작성하면 두 번째 또는 그 다음의 조회까지도 성능을 향상시킬 수 있습니다. 제안된 색인을 작성할 것인지 여부를 결정할 때는 해당 조회가 실행되는 빈도와 표에 대한 추가 색인을 유지보수하는 데 필요한 오버헤드를 고려해야 합니다.

FLD1, FLD2에 대한 영구 색인을 작성할 경우 순서는 다음과 같습니다.

1. 성능 모니터를 다시 시작합니다.
2. 어플리케이션을 재실행합니다.
3. 성능 모니터를 종료합니다.
4. 자료를 다시 평가합니다.

색인이 제안된 세 가지 조회는 더 이상 표 스캔을 실행하지 않습니다.

추가 데이터베이스 모니터 예

다음은 성능 모니터 통계에서 정보를 추출하는 방법을 보여 주는 추가 설명 또는 예입니다. 모든 예는 자료가 LIB/PERFDATA에서 수집되었으며 자료가 기록된 논리 파일이 작성되었다는 가정을 기초로 하고 있습니다.

1. 동적 재계획을 실행하는 조회는 몇 개입니까?

```
SELECT COUNT(*)
  FROM   LIB/QQQ1000
  WHERE  QQDYNR <> 'NA'
```

2. 동적 재계획에 사용된 명령문 텍스트는 무엇이며 그것이 사용된 이유는 무엇입니까?

```
SELECT QQDYNR, QQSTTX
  FROM   LIB/QQQ1000
  WHERE  QQDYNR <> 'NA'
```

   주: 동적 재계획의 이유 코드를 정의하려면 열 QQDYNR에 대한 설명을 참조해야 합니다.

3. LIB1/TBL1에 대해 작성된 색인은 모두 몇 개입니까?

```
SELECT COUNT(*)
  FROM   LIB/QQQ3002
  WHERE  QQTLN = 'LIB1'
    AND  QQTFN = 'TBL1'
```

4. LIB1/TBL1에 대해 작성된 모든 색인에서 사용된 키 열은 무엇이며 연관된 SQL문 텍스트는 무엇입니까?

```
SELECT A.QQTLN, A.QQTFN, A.QQIDXD, B.QQSTTX
  FROM   LIB/QQQ3002 A, LIB/QQQ1000 B
  WHERE  A.QQJFLD = B.QQJFLD
    AND  A.QQUCNT = B.QQUCNT
    AND  A.QQTLN  = 'LIB1'
    AND  A.QQTFN  = 'TBL1'
```

주: 이 조회는 SQL을 사용하여 실행된 조회의 키 열만 보여 줍니다.

5. LIB1/TBL1에 대해 작성된 모든 색인에 사용된 키 열은 무엇이며 연관된 SQL문 텍스트 또는 조회 열기 ID는 무엇입니까?

```
SELECT A.QQTLN, A.QQTFN, A.QQIDXD,
       B.QQOPID,C.QQSTTX
  FROM   LIB/QQQ3002 A INNER JOIN LIB/QQQ3014 B
         ON (A.QQJFLD = B.QQJFLD AND
             A.QQUCNT = B.QQUCNT)
         LEFT OUTER JOIN LIB/QQQ1000 C
         ON (A.QQJFLD = C.QQJFLD AND
             A.QQUCNT = C.QQUCNT)
  WHERE  A.QQTLN  = 'LIB1'
    AND  A.QQTFN  = 'TBL1'
```

주: 이 조회는 서버의 모든 조회에서 얻은 키 열을 보여 줍니다.

6. 어떤 유형의 SQL문이 실행되고 있습니까? 또한 가장 자주 실행되는 조회는 무엇입니까?

```
SELECT QQSTOP, COUNT(*)
  FROM   LIB/QQQ1000
  GROUP BY QQSTOP
  ORDER BY 2 DESC
```

7. 시간이 가장 많이 소요되는 SQL 조회는 무엇입니까? 또한, 이 조회를 사용하는 사용자는 누구입니까?

```
SELECT (QQETIM - QQSTIM), QQUSER, QQSTTX
  FROM   LIB/QQQ1000
  ORDER BY 1 DESC
```

8. 시간이 가장 많이 소요되는 조회는 무엇입니까?

```
SELECT (A.QQTTIM + B.QQCLKT), A.QQOPID, C.QQSTTX
  FROM   LIB/QQQ3014 A LEFT OUTER JOIN LIB/QQQ3019 B
         ON (A.QQJFLD = B.QQJFLD AND
             A.QQUCNT = B.QQUCNT)
         LEFT OUTER JOIN LIB/QQQ1000 C
         ON (A.QQJFLD = C.QQJFLD AND
             A.QQUCNT = C.QQUCNT)
  ORDER BY 1 DESC
```

주: 이 예는 세부 자료가 행 QQQ3019에 수집되어 있다고 가정한 것입니다.

9. 각 SQL 조회의 자료를 논리적으로 함께 그룹화하여 모든 SQL 조회에 대한 자료를 표시하십시오.

```
SELECT A.*
  FROM   LIB/PERFDATA A, LIB/QQQ1000 B
  WHERE  A.QQJFLD = B.QQJFLD
    AND  A.QQUCNT = B.QQUCNT
```

주: 이 조회는 관심있는 자료를 보다 이해하기 쉬운 형식으로 형식화하는 보고서에서 사용됩니다. 예를 들면, 이유 코드의 정의를 인쇄하는 보고서를 통해 모든 이유 코드 열이 확장될 수 있습니다(즉, 실제 열 QQRCOD = 'T1'은 조회 대상 표에 대한 색인이 없기 때문에 표 스캔이 실행 되었음을 의미합니다).

10. 키 길이가 2000바이트를 초과하거나 순서화에 120개 이상의 키 열이 지정되었기 때문에 임시 표를 사용하여 구현되는 조회의 수는 얼마나 됩니까?

```
SELECT COUNT(*)
  FROM   LIB/QQQ3004
  WHERE  QQRCOD = 'F6'
```

11. 재사용할 수 없는 ODP를 사용하여 구현된 SQL 조회는 무엇입니까?

```
SELECT B.QQSTTX
  FROM   LIB/QQQ3010 A, LIB/QQQ1000 B
  WHERE  A.QQJFLD = B.QQJFLD
    AND  A.QQUCNT = B.QQUCNT
    AND  A.QQODPI = 'N'
```

12. 조회 감독자가 중지한 모든 조회 예상 시간은 얼마입니까?

```
SELECT QQEPT, QQOPID
  FROM   LIB/QQQ3014
  WHERE  QQGVNS = 'Y'
```

주: 이 예는 세부 자료가 행 QQQ3019에 수집되어 있다고 가정한 것입니다.

13. 조회 예상 시간이 실제 시간보다 긴 조회는 무엇입니까?

```
SELECT A.QQEPT, (A.QQTTIM + B.QQCLKT), A.QQOPID,
       C.QQTTIM, C.QQSTTX
  FROM   LIB/QQQ3014 A LEFT OUTER JOIN LIB/QQQ3019 B
         ON (A.QQJFLD = B.QQJFLD AND
             A.QQUCNT = B.QQUCNT)
         LEFT OUTER JOIN LIB/QQQ1000 C
         ON (A.QQJFLD = C.QQJFLD AND
             A.QQUCNT = C.QQUCNT)
  WHERE  A.QQEPT/1000 > (A.QQTTIM + B.QQCLKT)
```

주: 이 예는 세부 자료가 행 QQQ3019에 수집되어 있다고 가정한 것입니다.

14. UNION 존재 검사를 실행하는 조회에 PTF가 적용되어야 합니다. 어떤 조회든지 UNION을 실행하는 경우에는 PTF가 적용되어야 합니다. 이 기능을 실행하는 조회가 있습니까?

```
SELECT COUNT(*)
  FROM   QQQ3014
  WHERE  QQUNIN = 'Y'
```

주: 결과가 0보다 크면 PTF가 적용되어야 합니다.

15. 사용자가 시스템 관리자이고 다음 릴리스로 업그레이드할 계획을 갖고 있다고 합시다. 이런 경우 다음과 같이 두 릴리스를 비교하는 것이 좋습니다.
    - 현재 릴리스의 어플리케이션에서 자료를 수집하고 이 자료를 LIB/CUR_DATA에 저장합니다.
    - 다음 릴리스로 이동합니다.

- 신규 릴리스의 어플리케이션에서 자료를 수집하고 이 자료를 LIB/NEW_DATA라는 다른 표에 저장합니다.
- 결과를 비교하기 위해 프로그램을 작성합니다. 자료를 관련시키기 위해 두 표에 있는 행 사이의 명령문 텍스트를 비교해야 합니다.

메모리 상주 데이터베이스 모니터 API를 사용하여 조회에 대한 통계 수집

메모리 상주 데이터베이스 모니터(DBMon)는 데이터베이스 성능을 모니터링하기 위한 또다른 방법을 제공하는 툴입니다. 이 툴은 SQL 성능 모니터링에만 사용하도록 고안된 것으로서 프로그래머 및 성능 분석자에게 유용합니다. DBMon 모니터는 새로운 API 세트의 도움을 받아 데이터베이스 모니터링 통계를 얻고, 메모리에서 사용자 대신 이를 관리합니다. 이 메모리 기반 모니터는 CPU 오버헤드를 줄일 뿐만 아니라 그 결과로 표의 *크기*도 줄여 줍니다.

데이터베이스 모니터 시작(STRDBMON) 명령은 성능 정보 수집시 서버 자원을 제한할 수 있습니다. 이 오버헤드는 주로 정보 수집시 성능 정보가 데이터베이스 표에 직접 기록되기 때문에 일어납니다. 메모리 기반 콜렉션 모드는 메모리에 성능 결과를 수집하고 관리함으로써 소요되는 서버 자원을 줄여줍니다. 그러면 모니터는 서버 전체 성능(또는 개별 SQL문의 성능)에 미치는 영향을 최소화하고 데이터베이스 성능 통계를 수집할 수 있습니다.

DBMon 모니터는 STRDBMON 모니터와 동일한 정보를 많이 수집하지만 성능 통계는 메모리에 보유됩니다. 일부 세부사항을 희생해서라도, SQL문에 대한 정보를 요약하여 수집되는 정보의 양을 줄입니다. 목적은, 성능 자료가 분석을 위해 결과표에 덤프될 때까지 자료의 조작 또는 변환을 지연시키면서 통계를 가능한 한 빨리 메모리에 보유하기 위해서입니다.

DBMon 모니터는 STRDBMON 모니터를 대체하기 위한 것이 아닙니다. DBMon 모니터의 세부사항을 잃으면 SQL문을 완전히 분석할 수 없게 되는 경우도 있습니다. 이러한 경우에는 STRDBMON 모니터를 사용해야 합니다.

DBMon 모니터는 정보를 일련의 행 형식으로 결합하고 축적하여 메모리의 자료를 관리합니다. 이것은 각각의 고유한 SQL문에 대해 명령문을 실행할 때마다 정보를 축적하고 가장 비용이 많이 드는 명령문 실행을 위해서만 세부 정보를 수집함을 의미합니다.

각 SQL문은 모니터가 다음을 기준으로 식별합니다.

- 명령문명
- 패키지(또는 프로그램)
- 준비된 명령문이 들어 있는 라이브러리
- 사용되는 커서명

완전히 동적인 명령문의 경우, 명령문 텍스트는 별도의 장소에 보유되고 명령문 식별은 포인터를 통해 내부에서 처리됩니다.

이러한 시스템은 각 SQL 조작을 표에 기록하는 오버헤드가 커지지 않도록 하면서, 통계를 메모리에 보유하기 위해 일부 세부사항은 희생합니다. 이렇게 하는 이유는 가능한 한 빨리 통계를 메모리에 보유함으로써, 나중에 표에 대한 자료를 덤프할 때 자료 조작 또는 자료 변환에 소요될 시간을 예약해 두기 위해서입니다.

DBMon은 정보를 새로운 행 형식으로 결합하고 축적하여 메모리에 있는 자료를 관리합니다. 따라서, 각각의 고유한 SQL문에 대해 명령문을 실행할 때마다 정보를 축적하고 서버는 가장 비용이 많이 드는 명령문 실행을 위해서만 세부 정보를 수집함을 의미합니다.

모니터는 명령문명, 패키지(또는 프로그램), 준비된 명령문이 들어 있는 라이브러리, 사용되는 커서명을 기준으로 각 SQL을 식별합니다. 완전한 동적 명령문의 경우,

- 명령문 텍스트는 별도의 장소에 보유되고,
- 명령문 식별은 포인터를 통해 내부에서 처리됩니다.

DBMon 모니터에 대한 API 지원

신규 API 세트가 DBMon 모니터를 지원할 수 있습니다. API는 다음 활동 각각을 지원합니다.

- 신규 모니터 시작
- 통계를 표에 덤프
- 모니터 자료를 메모리에서 지우기
- 모니터 상태 조회
- 신규 모니터 종료

신규 모니터를 시작하면, 시스템이 모니터하는 각 작업의 로컬 주소 공간에 정보가 저장됩니다. 각 명령문이 완료되면 시스템은 로컬 작업 공간의 정보를 공통의 시스템 공간으로 이동합니다. 이 공통의 시스템 공간에 들어갈 수 있는 양보다 더 많은 명령문이 실행되는 경우에는, 최근에 실행된 적이 없는 명령문이 드롭(drop)됩니다.

다음 주제에서는 데이터베이스 모니터 API에 대한 자세한 정보를 알려줍니다.

- 『메모리 상주 데이터베이스 모니터 외부 API 설명』
- 90 페이지의 『메모리 상주 데이터베이스 모니터 외부 표 설명』
- 90 페이지의 『샘플 SQL 조회』
- 90 페이지의 『메모리 상주 데이터베이스 모니터 행 식별』

메모리 상주 데이터베이스 모니터 외부 API 설명

메모리 상주 데이터베이스 모니터는 API 세트에 의해 제어됩니다. 자세한 정보는 iSeries Information Center의 **프로그래밍** 범주에 있는 OS/400 API 정보를 참조하십시오.

표 6. 외부 API 설명

QQQSSDBM	SQL 모니터를 시작하는 API
QQQCSDBM	SQL 모니터 메모리를 지우는 API
QQQDSDBM	SQL 모니터의 내용을 표에 덤프하는 API

표 6. 외부 API 설명 (계속)

QQQESDBM	SQL 모니터를 종료하는 API
QQQQSDBM	데이터베이스 모니터의 상태를 조회하는 API

메모리 상주 데이터베이스 모니터 외부 표 설명

메모리 상주 데이터베이스 모니터는 STRDBMON 모니터가 사용하는 논리 파일이 있는 단일 표를 사용하지 않고 고유 표 세트를 사용합니다. 메모리 상주 데이터베이스 모니터 표는 STRDBMON 모니터의 제안된 논리 파일과 아주 일치합니다.

주: 버전 4 릴리스 1부터, 새로 캡처된 정보는 메모리 상주 모니터에 나타나지 않으며, 이들 파일에 대한 파일 형식은 변경되지 않았어도 파일 기반 모니터에 대한 파일 형식은 변경되었습니다.

표 7. 외부 표 설명

QAQQQRYI	조회(SQL) 정보
QAQQTEXT	SQL문 텍스트
QAQQ3000	표 스캔
QAQQ3001	사용된 색인
QAQQ3002	작성된 색인
QAQQ3003	정렬
QAQQ3004	임시 표
QAQQ3007	Optimizer 시간 종료/ 고려된 모든 색인
QAQQ3008	부속 조회
QAQQ3010	호스트 변수 값

샘플 SQL 조회

STRDBMON 모니터의 경우, 모니터된 모든 자료가 저장된 표에서 정보를 추출하는 것은 사용자가 할 일입니다. 이 사용자가 선택한 어떤 조회 인터페이스를 통해서든지 이를 실행할 수 있습니다.

SQL 모니터를 지원하는 iSeries Navigator를 사용하는 경우에는 GUI를 통해 직접 결과를 분석할 수 있습니다. 어떤 표에서는 정보를 추출하기 위해 사용하거나 수정할 수 있는 여러 가지 조회가 함께 제공되어 있습니다. 이러한 조회 리스트에 대해서는 iSeries용 DB2 UDB 웹 사이트,Common queries on analysis of DB Performance Monitor data  를 참조하십시오.

메모리 상주 데이터베이스 모니터 행 식별

결합 키 열 QQKEY는 여러 표를 함께 결합하는 것을 단순화합니다. 이 열은 결합 필드(QQJFLD)와 데이터베이스 모니터가 사용한 고유 조회 카운터(QQCNT)를 대체합니다. 결합 키 열에는 해당 조회에 대한 모든 정보를 각 표에서 수신할 수 있도록 하는 고유 식별자가 들어 있습니다.

이 결합 키 열은 조회의 개별 단계에 대한 특정 정보를 식별하는 데 여전히 필요한 세부 열 모두를 대체하지는 않습니다. QDT(조회 정의 템플리트) 번호 또는 부속 선택 번호는 각 세부 단계에 대한 정보를 식별합니다. 조회 프로세스의 각 단계에 속하는 행이 어떤 행인지를 식별하려면 이 열을 사용하십시오.

- QQQDTN - 조회 정의 템플리트 번호
- QQQDTL - 조회 정의 템플리트 부속 선택 번호(부속 조회)
- QQMATN - 구체화된 조회 정의 템플리터 번호(보기)
- QQMATL - 구체화된 조회 정의 템플리트 부속 선택 번호(보기/부속 조회)
- QQMATULVL - 구체화된 조회 정의 템플리터 결합 번호(보기/결합)

모니터된 조회에 부속 조회, 합집합 연산 또는 보기 조작이 들어 있는 경우에 이 열을 사용하십시오. 모든 조회 유형은 원래의 조회 요구를 만족시키기 위해 여러 QDT를 생성할 수 있습니다. 서버는 이 열을 사용하여 각 QDT에 대한 정보를 분리하고 동시에 각 QDT가 이 원래 조회(QQKEY)에 속하는 것으로 식별될 수 있도록 합니다.

iSeries Navigator의 SQL 성능 모니터를 사용하여 데이터베이스 성능 모니터링

SQL 성능 모니터를 사용하여 SQL문이 사용하는 자원을 추적할 수 있습니다. 특정 자원이나 많은 자원을 모니터할 수 있습니다. 자원 사용 정보는 시스템 및 SQL문이 제대로 수행되고 있는지 또는 미세 성능 조정이 필요한지 여부를 판별하는 데 도움이 됩니다. 자원을 모니터하기 위해 선택할 수 있는 두 가지 유형의 모니터가 있습니다.

SQL 성능 모니터 요약

요약 SQL 성능 모니터는 시스템 인터페이스에서 찾을 수 있는 메모리 상주 데이터베이스 모니터의 iSeries Navigator 버전입니다. 이름에서 알 수 있듯이 이 모니터는 메모리에 상주하며 수집한 자료의 요약을 보유하기만 합니다. 모니터가 멈추거나 종료되면 분석을 위해 이 자료가 하드 디스크에 기록됩니다. 모니터는 정보를 메모리에 저장하므로 시스템 성능에 미치는 영향을 최소화할 수 있습니다. 그러나 세부사항이 누락될 수 있습니다. 자세한 내용은 88 페이지의 『메모리 상주 데이터베이스 모니터 API를 사용하여 조회에 대한 통계 수집』을 참조하십시오.

SQL 성능 모니터 세부사항

세부사항 SQL 성능 모니터는 시스템 인터페이스에서 찾을 수 있는 데이터베이스 모니터의 iSeries Navigator 버전입니다. 이 모니터는 자세한 자료를 실시간으로 하드 디스크에 저장하며 모니터를 멈추거나 종료하지 않아도 결과를 분석할 수 있습니다. 모니터가 생성한 자료를 기반으로 Visual Explain을 실행하도록 선택할 수도 있습니다. 이 모니터는 실시간으로 자료를 저장하므로 시스템의 성능에 영향을 줄 수 있습니다. 자세한 내용은 79 페이지의 『데이터베이스 모니터 시작(STRDBMON) 명령』을 참조하십시오.

SQL 성능 모니터 사용에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 92 페이지의 『SQL 성능 모니터 작성』
- 92 페이지의 『SQL 성능 모니터 자료 저장(모니터 멈춤)』
- 93 페이지의 『SQL 성능 모니터 자료 분석』

SQL 성능 모니터 작성

신규 SQL 성능 모니터를 작성하면 모니터의 새 인스턴스가 시스템에 작성됩니다. 시스템에서 여러 개의 모니터 인스턴스가 동시에 실행될 수 있습니다. 그러나 모든 작업을 모니터링하는 모니터 인스턴스는 단 하나입니다. 모든 작업에 대한 정보를 수집할 때 모니터는 이전에 시작된 작업이나 모니터가 작성된 후에 시작된 새 작업에 대한 정보를 수집합니다. 그러나 성능 모니터를 종료하면 모니터의 인스턴트도 종료되기 때문에 실행이 계속되지 않습니다. 반면 멈춰진 모니터는 다시 시작할 수 있습니다. SQL 성능 모니터를 작성하려는 경우:

1. **iSeries Navigator** 창에서 서버 → **데이터베이스**를 펼치십시오.
2. **SQL 성능 모니터**를 마우스 오른쪽 단추로 클릭하고 **신규**를 선택하십시오.
3. **요약** 또는 **세부사항**을 선택하십시오.
4. **이름** 필드에 모니터에 제공할 이름을 지정하십시오.
5. 모니터가 수집한 정보를 저장할 라이브러리를 **라이브러리** 필드에 지정하십시오.
6. 모니터가 사용할 수 있는 시스템 메모리의 최대 *크기*를 지정하려면 **기억장치(MB)** 필드에 MB 단위로 *크기*를 지정하십시오.
7. **모니터된 작업** 탭을 클릭하십시오.
8. 사용 가능한 모든 작업을 모니터하려고 할 때 모든 작업을 모니터링하는 다른 모니터가 없는 경우 **모두**를 선택하십시오. 한 번에 단 하나의 모니터만 모든 작업을 모니터할 수 있습니다.
9. 특정 작업만 모니터하려면 **사용할 수 있는 작업** 리스트에서 모니터할 작업을 선택하고 **선택**을 클릭하십시오. 모니터할 각 작업에 대해 이 단계를 반복하십시오. 한 번에 단 하나의 활동 모니터만 개별 작업을 모니터할 수 있습니다.
10. 모니터하지 않을 작업을 선택했거나 선택한 작업이 이미 모니터되고 있는 경우 **선택한 작업** 리스트에서 해당 작업을 선택하고 **제거**를 클릭하십시오.
11. 요약 모니터를 선택한 경우 **수집할 자료** 탭을 클릭하십시오.
12. **수집할 자료** 탭에서 수집할 자료 유형을 선택하십시오. 모든 자료 유형을 수집하려면 **모두 선택**을 클릭하십시오.
13. **확인**을 클릭하십시오. 성능 모니터가 시작되고 종료되거나 멈춰질 때까지 실행됩니다.

SQL 성능 모니터 자료 저장(모니터 멈춤)

실행될 때 실시간으로 자료를 저장하는 세부사항 SQL 성능 모니터와는 달리 요약 SQL 모니터가 수집한 자료는 메모리에 저장되므로 사용하려면 저장해야 합니다. 요약 SQL 성능 모니터를 저장하려면 다음을 수행하십시오.

1. **iSeries Navigator** 창에서, 서버 → **데이터베이스** → **SQL 성능 모니터**를 펼치십시오.
2. 오른쪽 분할 창에서 성능 모니터를 선택하십시오.
3. 이 모니터를 다시 시작하려면 저장할 모니터를 마우스 오른쪽 단추로 클릭하고 **멈춤**을 선택하십시오.
4. 이 모니터를 다시 시작하지 않으려면 저장할 모니터를 마우스 오른쪽 단추로 클릭하고 **종료**를 선택하십시오.

모니터가 수집한 자료가 저장됩니다.

SQL 성능 모니터 자료 분석

성능 모니터를 사용하여 자원을 수집한 후 모니터가 수집한 자료를 분석할 수 있습니다. 요약 SQL 성능 모니터를 사용하는 경우에는 먼저 모니터를 멈추거나 종료해야 합니다. 세부사항 SQL 성능 모니터는 계속 실행할 수 있습니다.

1. **iSeries Navigator** 창에서, 서버 → **데이터베이스** → **SQL 성능 모니터**를 펼치십시오.
2. 오른쪽 분할 창에서 성능 모니터를 마우스 오른쪽 단추로 클릭하고 **결과 분석**을 선택하십시오.
3. 보려는 자료의 수집 주기를 **수집 주기** 리스트에서 선택하십시오. 보고자 하는 자료를 선택하십시오. **수집 주기** 리스트는 모든 모니터에 적용되지만 요약 모니터만 복수 주기를 사용할 수 있습니다. 가져온 모니터는 **사용할 수 없는 정보**를 표시합니다.
4. 보고자 하는 자료를 선택하십시오.
5. 다른 방법으로 자료를 보려면 사용할 뷰에 해당하는 탭을 클릭하고 원하는 조회를 선택하십시오. 각 뷰 유형에 대한 정보를 보려면 뷰를 선택하고 F1을 누르십시오.
6. 조회를 수정하려면 **선택한 조회 수정**을 클릭하십시오. **SQL 스크립트 실행** 창을 나가기 전에 조회를 실행해야 합니다. 이 조회를 저장하려면 **SQL 스크립트 실행** 창에서 저장해야 합니다.
7. 원하는 각각의 뷰를 선택한 후 **확인** 또는 **뷰 결과**를 클릭하십시오.

Visual Explain을 사용하여 조회의 효율성 보기

iSeries Navigator의 **Visual Explain** 툴을 사용하여 SQL문의 구현을 그래픽으로 표시하는 조회 그래프를 작성할 수 있습니다. 정적 및 동적 SQL문에 대한 정보를 보기 위해 이 툴을 사용할 수 있습니다. Visual Explain은 SELECT, INSERT, UPDATE 및 DELETE와 같은 SQL문 유형을 지원합니다.

Visual Explain은 사용자 조회에서 가장 비싸거나 가장 시간이 걸리는 조작을 찾는 데 사용될 수 있습니다. 다음과 같이 하여 조회 성능을 향상시킬 수 있습니다.

- SQL문 다시 쓰기
- 조회 속성 및 환경 설정 변경
- 조회 Optimizer가 권장하는 색인 작성

Visual Explain을 사용하여 다음을 수행할 수 있습니다.

- 최적화 중 사용된 통계를 열람합니다.
- 자료를 액세스하기 위해 색인이 사용되었는지 여부를 판별합니다. 색인이 사용되지 않았다면, Visual Explain이 어떤 열이 색인화되는 데 유익할지 판별하는 데 도움을 줄 수 있습니다.
- 구현의 전후 그림을 비교하여 조정 기술을 수행한 결과를 봅니다.
- 전체 예상 비용 및 예상되는 리턴 행 수를 포함하여, 조회 그래프에 있는 각 조작(아이콘)에 대한 정보를 확보합니다.

Visual Explain은 데이터베이스 모니터 표에 저장된 자료에 대해 작업합니다. 이 표는 STRDBMON 명령의 실행 결과가 들어 있는 표입니다. 메모리 상주 모니터의 결과로 생긴 표에 대해서는 작업하지 않습니다. Visual

Explain 툴을 호출하는 방법은 두 가지입니다. 첫 번째는 가장 일반적인 방법으로, iSeries Navigator를 통한 방법입니다. 두 번째는 Visual Explain API를 통한 방법입니다. iSeries Information Center의 프로그래밍 범주에 있는 OS/400 API 정보를 참조하시시오. iSeries Navigator의 다음 창에서 Visual Explain을 시작할 수 있습니다.

- 사용할 수 있는 SQL 성능 모니터의 리스트를 펼치십시오. SQL 성능 모니터를 마우스 오른쪽 버튼으로 클릭하고, **설명 가능한 명령문 나열** 옵션을 선택하십시오. 그러면 **SQL 성능 모니터에 설명 가능한 명령문** 창이 열립니다.
- 설명하고자 하는 SQL문을 강조표시하고(마우스 왼쪽 버튼을 클릭), Visual Explain 실행 버튼을 클릭하십시오.
- **SQL 스크립트 실행** 창에 SQL문을 입력하십시오. 명령문을 선택한 다음 컨텍스트 메뉴에서 **설명...**을 선택하거나, **Visual Explain** 풀다운 메뉴에서 **실행 및 설명...**을 선택하십시오.

데이터베이스 모니터 표(서버에서 STRDBMON 명령을 실행하여 생긴 결과)는 iSeries Navigator를 통해 설명될 수 있습니다. 먼저, 데이터베이스 모니터 표를 iSeries Navigator로 가져와야 합니다. 이를 수행하려면, SQL 성능 모니터를 마우스 오른쪽 버튼으로 클릭하고 **가져오기** 옵션을 선택하십시오. 성능 모니터의 이름(iSeries Navigator에서 알려질 이름) 및 데이터베이스 모니터 표의 규정된 이름을 지정하십시오. 모니터 유형으로 반드시 자세히를 선택하십시오. 자세히는 파일 기반(STRDBMON) 모니터를 나타내고, 요약은 메모리 상주 모니터(Visual Explain에서 지원되지 않음)를 나타냅니다. 모니터가 가져오기 되면, iSeries Navigator에서 Visual Explain을 시작하기 위한 단계를 수행하십시오.

조회 Optimizer 디버그 메세지에 있는 모든 정보(즉, 표 이름, 예상된 행 수, 색인 통지 정보)는 속성 섹션에서 Visual Explain을 통해 보여집니다. 실제적으로, Visual Explain과 파일 기반 모니터(STRDBMON)에는 Optimizer 디버그 메세지에 있는 정보보다 더 많은 정보가 들어 있습니다.

- 조회되는 표의 긴 이름과 짧은 이름을 알 수 있습니다.
- 임시 색인을 작성하기 위해 선택할 때 사용된 열을 알 수 있습니다(대부분의 임시 색인은 내용이 부족하거나 선택/생략 색인임을 기억하십시오).
- 조회가 실행되었을 때의 환경에 관한 속성을 알 수 있습니다. 예를 들어, ALWCPYDTA 설정이 무엇이었는지 보여줍니다.
- 메모리 풀의 이름과 *크기*를 보여줍니다.
- INI 파일이 사용된 조회를 보여줍니다.

Visual Explain의 또다른 이점은 조회 내에 있는 부속 선택을 구별할 수 있는 기능입니다. 부속 조회가 있는 조회를 실행할 경우, 어느 Optimizer 디버그 메세지가 어느 부속 선택, 외부 조회 또는 부속 조회에 해당하는지 판별하기 어려운 경우가 있습니다. Visual Explain이 이 문제를 처리합니다.

조회 속성 변경(CHGQRYA) 명령을 사용하여 조회 속성 변경

특정 작업 도중 실행할 조회의 여러 가지 속성 유형을 CHGQRYA 명령이나 iSeries Navigator 인터페이스를 사용하여 수정할 수 있습니다. 수정할 수 있는 속성 유형은 다음과 같습니다.

- 예측 조회 감독자
- 조회 병렬화
- 비동기 작업
- 리모트에 CHGQRYA 적용
- 조회 옵션 파일 매개변수

조회를 시작하기 전에 서버는 예상 조회 경과 시간에 대한 조회 시간 제한을 검사합니다. 또한, 서버는 0이라는 시간 제한을 사용하여 몇 번의 반복을 통해 실행할 필요없이 조회에 대한 성능을 최적화합니다.

사용자는 조회 메세지 CPA4259를 검사하여 예측된 실행시간 및 조회가 실행할 조작을 알 수 있습니다. 조회가 취소되는 경우에는 디버그 메세지가 작업 기록부에 기록됩니다.

조회의 예상 또는 예측 실행시간(경과 실행 시간)이 과도한 경우, iSeries용 DB2 Universal Database 예측 조회 감독자가 조회의 시작을 중단할 수 있습니다. 감독자는 조회가 실행되는 도중이 아니라 조회가 실행되기 *전에* 작동합니다. iSeries의 대화식 또는 일괄처리 작업에서 이를 사용할 수 있습니다. 또한, 모든 iSeries용 DB2 Universal Database 조회 인터페이스와 함께 사용할 수도 있으며 SQL 조회와 함께 사용할 수도 있습니다. 세부사항은 104 페이지의 『iSeries용 DB2 UDB 예측 조회 감독자를 사용하여 장기 실행 조회 제어』를 참조하십시오.

조회 옵션 파일 QAQQINI를 사용하여 조회를 동적으로 제어

조회 옵션 파일 QAQQINI 지원은 조회가 CHGQRYA 명령 및 QAQQINI 파일을 통해 실행되는 환경을 동적으로 수정하거나 대체할 수 있는 기능을 제공합니다.

조회 옵션 파일 QAQQINI는 조회 Optimizer가 사용하는 일부 속성을 설정하는 데 사용됩니다. 실행되는 각 조회에 대해, 조회 옵션 값이 CHGQRYA CL 명령의 QRYOPTLIB 매개변수에 지정된 라이브러리의 QAQQINI 파일에서 검색되어, 조회를 최적화하거나 구현하는 데 사용됩니다.

QAQQINI 파일을 통해 수정할 수 있는 환경 속성은 다음과 같습니다.

- APPLY_REMOTE
- ASYNC_JOB_USAGE
- COMMITMENT_CONTROL_LOCK_LIMIT
- FORCE_JOIN_ORDER
- IGNORE_LIKE_REDUNDANT_SHIFTS
- MESSAGES_DEBUG
- OPEN_CURSOR_CLOSE_COUNT

| • OPEN_CURSOR_THRESHOLD
| • OPTIMIZE_STATISTIC_LIMITATION
| • OPTIMIZATION_GOAL
| • PARALLEL_DEGREE
| • PARAMETER_MARKER_CONVERSION
| • QUERY_TIME_LIMIT
| • REOPTIMIZE_ACCESS_PLAN
| • SQLSTANDARDS_MIXED_CONSTANT
| • SQL_SUPPRESS_WARNINGS
| • SQL_TRANSLATE_ASCII_TO_JOB
| • STAR_JOIN
| • SYSTEM_SQL_STATEMENT_CACHE
| • UDF_TIME_OUT
| • VISUAL_EXPLAIN_DIAGRAM

조회 옵션 파일 QAQQINI를 현재 보유하고 있거나 앞으로 이 파일을 포함할 라이브러리를 지정하려면 『QAQQINI 파일 지정』을 참조하십시오.

사용자만의 QAQQINI 파일을 작성하려면 『QAQQINI 조회 옵션 파일 작성』을 참조하십시오.

QAQQINI 파일 지정

조회 옵션 파일 QAQQINI가 현재 들어 있거나 앞으로 들어 있게 될 라이브러리를 지정하려면 CHGQRYA 명령을 QRYOPTLIB(조회 옵션 라이브러리) 매개변수와 함께 사용하십시오. 조회 옵션 파일은 각 조회의 QRYOPTLIB 매개변수에 지정된 라이브러리에서 검색되고 해당 작업 또는 사용자 세션 기간 동안 유효한 상태로 남아 있거나 CAHQRYA 명령이 QRYOPTLIB 매개변수를 변경할 때까지 유효한 상태로 남아 있습니다.

CHGQRYA 명령이 실행되지 않았거나, 실행되었지만 QRYOPTLIB 매개변수가 지정되지 않은 경우, 라이브러리 QUSRSYS에서 QAQQINI 파일이 존재하고 있는지 탐색됩니다. 조회 옵션 파일이 조회에서 발견되지 않으면 어떤 속성도 수정되지 않습니다. 서버는 QUSRSYS에 INI 파일 없이 제공되므로, INI 파일이 없음을 나타내는 메세지를 수신할 수 있습니다. 이 메세지는 오류가 아니라 단지 모든 디폴트 값이 들어 있는 QAQQINI 파일 사용되고 있다는 것을 표시하기 위한 것입니다. 작업에 대한 QRYOPTLIB 매개변수 초기 값은 QUSRSYS입니다.

QAQQINI 조회 옵션 파일 작성

각 서버의 라이브러리 QSYS에는 QAQQINI 템플리트 파일이 제공되어 있습니다. QSYS의 QAQQINI 파일은 모든 사용자 지정 QAQQINI 파일을 작성할 때 템플리트로 사용하기 위한 것입니다. 사용자 자신의

QAQQINI 파일을 작성하려면, CRTDUPOBJ 명령을 사용하여, CHGQRYA QRYOPTLIB 매개변수에서 지정될 라이브러리에 있는 QAQQINI 파일의 사본을 작성하십시오. 파일 이름은 여전히 QAQQINI이어야 합니다. 예를 들면 다음과 같습니다.

```
CRTDUPOBJ OBJ(QAQQINI)
FROMLIB(QSYS)
OBJTYPE(*FILE)
TOLIB(MYLIB)
DATA(*YES)
```

시스템 제공 트리거가 QSYS의 QAQQINI 파일에 연결되어 있으므로, QAQQINI 파일을 복사하는 유일한 방법은 반드시 CRTDUPOBJ CL 명령을 통해야 합니다. CPYF와 같이 다른 방법이 사용되면, 트리거가 손상되어 옵션 파일이 검색될 수 없거나 옵션 파일이 갱신될 수 없다는 오류가 신호됩니다.

트리거 프로그램이 QAQQINI 파일에 연결되어 있으므로, 파일을 작성하기 위해 CRTDUPOBJ CL이 사용되면 다음의 CPI321A 정보용 메세지가 작업 기록부에 6번 표시됩니다. 이것은 오류가 아닙니다. 단지 정보용 메세지일 뿐입니다.

CPI321A 정보 메세지: 라이브러리 &1에 있는 트리거 QSYS_TRIG_&1___QAQQINI___00000&N이 라이브러리 &1의 파일 QAQQINI에 추가되었습니다. 앰퍼샌드 변수(&1, &N)는 라이브러리명 또는 숫자 값이 들어 있는 대체 변수입니다.

주: QSYS의 파일 QAQQINI는 수정하지 않을 것을 권장합니다. 이것은 QUSRSYS로 복제되거나 사용자 지정 라이브러리로 사용할 원본 템플리트이기 때문입니다.

QAQQINI 조회 옵션 파일 형식

조회 옵션 파일

```
A                                       UNIQUE
A                   R QAQQINI           TEXT('Query options + file')
A                     QQPARM      256A  VARLEN(10) +
                                        TEXT('Query+
                                              option parameter') +
                                        COLHDG('Parameter')
A                     QQVAL       256A  VARLEN(10) +
                                        TEXT('Query option +
                                              parameter value') +
                                        COLHDG('Parameter Value')
A                     QQTEXT      1000G VARLEN(100) +
                                        TEXT('Query +
                                              option text') +
                                        ALWNULL +
                                        COLHDG('Query Option' +
                                               'Text') +
                                        CCSID(13488) +
                                        DFT(*NULL)
A                   K QQPARM
```

라이브러리 QSYS에 제공된 QAQQINI 파일은 다음 행으로 사전에 채워져 있습니다.

표 8. QAQQINI 파일 레코드. 설명

QQPARM	QQVAL
APPLY_REMOTE	*DEFAULT
ASYNC_JOB_USAGE	*DEFAULT
COMMITMENT_CONTROL_LOCK_LIMIT	*DEFAULT
FORCE_JOIN_ORDER	*DEFAULT
IGNORE_LIKE_REDUNDANT_SHIFTS	*DEFAULT
MESSAGES_DEBUG	*DEFAULT
OPEN_CURSOR_CLOSE_COUNT	*DEFAULT
OPEN_CURSOR_THRESHOLD	*DEFAULT
OPTIMIZATION_GOAL	*DEFAULT
OPTIMIZE_STATISTIC_LIMITATION	*DEFAULT
PARALLEL_DEGREE	*DEFAULT
PARAMETER_MARKER_CONVERSION	*DEFAULT
QUERY_TIME_LIMIT	*DEFAULT
REOPTIMIZE_ACCESS_PLAN	*DEFAULT
SQLSTANDARDS_MIXED_CONSTANT	*DEFAULT
SQL_SUPPRESS_WARNINGS	*DEFAULT
SQL_TRANSLATE_ASCII_TO_JOB	*DEFAULT
STAR_JOIN	*DEFAULT
SYSTEM_SQL_STATEMENT_CACHE	*DEFAULT
UDF_TIME_OUT	*DEFAULT
VISUAL_EXPLAIN_DIAGRAM	*DEFAULT

조회 옵션 파일에서 옵션 설정

QAQQINI 파일 조회 옵션은 INSERT, UPDATE 또는 DELETE SQL문을 사용하여 수정할 수 있습니다.

다음 예의 경우, QAQQINI 파일은 라이브러리 MyLib에서 이미 작성된 것입니다. MyLib/QAQQINI의 기존 행을 갱신하려면 UPDATE SQL문을 사용하십시오. 다음 예는 조회 Optimizer가 디버그 메세지를 출력하도록 MESSAGES_DEBUG = *YES를 설정한 것입니다.

```
UPDATE MyLib/QAQQINI SET QQVAL='*YES'
WHERE QQPARM='MESSAGES_DEBUG'
```

MyLib/QAQQINI의 기존 행을 삭제하려면 DELETE SQL문을 사용하십시오. 다음 예는 QAQQINI 파일에서 QUERY_TIME_LIMIT 행을 삭제합니다.

```
DELETE FROM MyLib/QAQQINI
WHERE QQPARM='QUERY_TIME_LIMIT'
```

MyLib/QAQQINI에 새로운 행을 삽입하려면 INSERT SQL문을 사용하십시오. 다음 예는 *NOMAX라는 값을 가지는 QUERY_TIME_LIMIT 행을 QAQQINI 파일에 추가합니다.

```
 INSERT INTO MyLib/QAQQINI
VALUES('QUERY_TIME_LIMIT','*NOMAX','New time limit set by DBAdmin')
```

QAQQINI 조회 옵션

다음 표는 QAQQINI 명령에서 지정할 수 있는 조회 옵션을 요약한 것입니다.

표 9. QAQQINI 명령에서 지정된 조회 옵션

매개변수	값	설명
APPLY_REMOTE	*DEFAULT	디폴트 값은 *NO로 설정됩니다.
	*NO	작업의 CHGQRYA 속성이 리모트 작업에 적용되지 않습니다. 리모트 작업은 서버에서 해당 작업과 연관된 속성을 사용합니다.
	*YES	분배된 표를 포함하는 데이터베이스 조회 처리에 사용되는 리모트 작업에 작업의 조회 속성이 적용됩니다. *SYSVAL이 지정된 속성의 경우에는, 리모트 서버의 시스템 값이 리모트 작업의 사용됩니다. 이 옵션은, 작업에 CHGQRYA가 사용된 경우, 리모트 작업에 CHGQRYA 명령을 사용할 권한이 있어야 합니다.
ASYNC_JOB_USAGE	*DEFAULT	디폴트 값은 *LOCAL로 설정됩니다.
	*LOCAL	데이터베이스 조회가 실행되는 서버에 대해 로컬로 존재하는 표만 포함시키는 데이터베이스 조회에 비동기 작업이 사용됩니다. 또한, 분배된 표를 포함하는 조회의 경우, 이 옵션은 필요한 통신이 비동기 상태가 되는 것을 허용합니다. 그러면 분배된 표의 조회에 관련된 각 서버가 조회의 해당 부분을 다른 서버와 동시에(병렬로) 실행할 수 있습니다.
	*DIST	분배된 표를 포함하는 데이터베이스 조회에 비동기 작업이 사용됩니다.
	*ANY	비동기 작업이 어떤 데이터베이스 조회에나 사용될 수 있습니다.
	*NONE	데이터베이스 조회 처리에 비동기 작업을 사용하는 것이 허용되지 않습니다. 또한, 분배된 표를 포함하는 조회의 모든 처리가 동시에 발생합니다. 따라서, 시스템간 병렬 처리는 발생하지 않습니다.
COMMIT_CONTROL_ LOCK_LIMIT	*DEFAULT	*DEFAULT는 500,000,000와 동일합니다.
	정수 값	새로운 값을 설정한 후 시작된 확약 트랜잭션에 잠겨질 수 있는 최대 레코드 수. 유효 정수 값은 1-500,000,000입니다.
FORCE_JOIN_ORDER	*DEFAULT	디폴트 값은 *NO로 설정됩니다.
	*NO	Optimizer가 결합 표의 순서를 재조정하는 것이 허용됩니다.
	*SQL	단지, SQL JOIN 구문을 사용하는 조회에 대해 결합 순서를 강제합니다. 이는 V4R4M0 이전의 Optimizer에 대한 작동을 모방합니다.
	*PRIMARY nnn	숫자 값 nnn(nnn은 선택적이며 디폴트 값은 1임)에 의해 나열된 파일에 대한 결합 위치를 결합에 대한 1차 위치(또는 다이얼)로 강제 수행합니다. 그런 다음, Optimizer가 비용을 기준으로 하여 나머지 모든 파일에 대한 결합 순서를 결정합니다.
	*YES	조회 Optimizer가 최적화 프로세스의 일부로 결합 표의 순서를 재조정하는 것이 허용되지 않습니다. 결합은 조회에서 표가 지정된 순서대로 발생합니다.

표 9. *QAQQINI 명령에서 지정된 조회 옵션 (계속)*

매개변수	값	설명
IGNORE_LIKE_ REDUNDANT_SHIFTS	*DEFAULT	디폴트 값은 *ALWAYS로 설정됩니다.
	*ALWAYS	SQL LIKE 술부나 OPNQRYF 명령 %WLDCRD 내장 함수를 처리할 때 DBCS 개방 피연산자에 대해 중복 시프트 문자가 무시됩니다. 이 옵션을 사용하면 조회 Optimizer가 DBCS 개방 , DBCS 선택 필드 또는 DBCS 전용 피연산자와 관련된 SQL LIKE 또는 OPNQRYF %WLDCRD 술부에 대한 키 행 위치 지정을 수행하기 위해 색인을 사용하지 못하도록 제한할 수 있습니다.
	*OPTIMIZE	SQL LIKE 술부 또는 OPNQRYF 명령 %WLDCRD 내장 함수를 처리할 때 이들 술부에 대한 키 행 위치 지정을 수행하기 위해 색인이 사용되는지 여부에 따라 DBCS 개방 피연산자에 대해 중복 시프트 문자가 무시되거나 무시되지 않을 수 있습니다. 이 옵션을 사용하면 조회 Optimizer가 DBCS 개방 , DBCS 선택 필드 또는 DBCS 전용 피연산자와 관련된 SQL LIKE 또는 OPNQRYF %WLDCRD 술부에 대한 키 행 위치 지정을 고려할 수 있습니다.
MESSAGE_DEBUG	*DEFAULT	디폴트 값은 *NO로 설정됩니다.
	*NO	디버그 메세지가 표시되지 않습니다.
	*YES	조회 Optimizer의 디버그 메세지를 모두 발행합니다.
OPEN_CURSOR_CLOSE_ COUNT	*DEFAULT	*DEFAULT는 0과 동등합니다. 세부사항은 정수 값을 참조하십시오.
	정수 값	OPEN_CURSOR_CLOSE_COUNT는 OPEN_CURSOR_THRESHOLD와 함께 사용되어 작업 내의 열기 커서 수를 관리합니다. 열기 커서 및 의사 닫힘 커서가 포함된 열기 커서의 수가 OPEN_CURSOR_THRESHOLD에 의해 지정된 값에 도달하면, 가장 이전에 사용된 커서가 먼저 닫히면서 의사 닫힘 커서가 완전히 닫힙니다. 이 값은 닫힐 커서의 수를 판별합니다. 이 매개변수에 대해 유효한 값은 1 - 65536입니다. 이 매개변수의 값은 OPEN_CURSOR_THREHOLD 매개변수에 있는 수 보다 적거나 같아야 합니다. OPEN_CURSOR_THRESHOLD이 *DEFAULT인 경우, 이 값은 무시됩니다. OPEN_CURSOR_THRESHOLD가 지정되고 이 값이 *DEFAULT인 경우 닫힌 커서의 수는 OPEN_CURSOR_THRESHOLD의 10%에 해당하는 값을 정수 값으로 반올림한 값입니다.
OPEN_CURSOR_ THRESHOLD	*DEFAULT	*DEFAULT는 0과 동등합니다. 세부사항은 정수 값을 참조하십시오.
	정수 값	OPEN_CURSOR_THRESHOLD는 OPEN_CURSOR_CLOSE_COUNT와 함께 사용되어 작업 내의 열기 커서 수를 관리합니다. 열기 커서 및 의사 닫힘 커서가 포함된 열기 커서의 수가 이 임계값에 도달하면 가장 이전에 사용된 커서가 먼저 닫히면서 의사 닫힘 커서가 완전히 닫힙니다. 닫힐 커서의 수는 OPEN_CURSOR_CLOSE_COUNT에 의해 판별됩니다. 이 매개변수에 대해 유효한 사용자 입력 값은 1 - 65536입니다. 값이 0(디폴트 값)일 경우 이는 임계값이 없으며 작업 내의 열기 커서 수를 기준으로 완전한 닫기가 강제로 수행되지 않음을 나타냅니다.
OPTIMIZATION_GOAL	*DEFAULT	최적화 목표는 인터페이스(ODBC, SQL 사전컴파일러 옵션, OPTIMIZE FOR nnn ROWS절)에 의해 판별됩니다.
	*FIRSTIO	모든 조회는 출력의 첫번 페이지를 가능한 한 빨리 리턴하는 목표로 최적화됩니다. 이 목표는 출력 자료의 첫번 페이지를 본 다음 조회를 중단할 수 있는 사용자가 출력의 제어를 제어할 때 잘 이루어집니다. OPTIMIZE FOR nnn ROWS절을 사용하여 코드된 조회는 이 절에 의해 지정된 목표를 이행합니다.
	*ALLIO	모든 조회는 전체 조회를 최소한의 경과 시간으로 실행을 완료하는 목표로 최적화됩니다. 이는 조회의 출력이 파일이나 보고서로 작성되거나 인터페이스가 출력 자료의 대기열에 들어갈 경우에 유용한 옵션입니다. OPTIMIZE FOR nnn ROWS절을 사용하여 코드된 조회는 이 절에 의해 지정된 목표를 이행합니다.

표 9. QAQQINI 명령에서 지정된 조회 옵션 (계속)

매개변수	값	설명
OPTIMIZE_STATISTIC_ LIMITATION	*DEFAULT	색인 통계를 수집하는 데 소요되는 시간을 조회 Optimizer가 판별합니다.
	*NO	조회 Optimizer가 색인 통계 자료를 수집하지 않습니다. 기본 통계 자료가 최적화에 사용됩니다(이 옵션은 꼭 필요한 경우에만 사용하십시오).
	*PERCENTAGE 정수 값	통계 자료를 수집하는 동안 탐색되는 색인의 최대 백분율을 지정합니다. 유효한 값은 1-99입니다.
	*MAX_NUMBER_ OF_RECORDS_ ALLOWED 정수 값	통계 자료 수집이 허용되는 표 중에서, 행 수가 가장 많은 표의 크기를 지정합니다. 지정한 값보다 많은 행이 있는 표에서는 Optimizer가 통계 자료를 수집하지 않고 디폴트 값을 사용합니다.
PARALLEL_DEGREE	*DEFAULT	디폴트 값은 *SYSVAL로 설정됩니다.
	*SYSVAL	사용된 처리 옵션이 시스템 값의 현재 값인 QQRYDEGREE로 설정됩니다.
	*IO	데이터베이스 조회 Optimizer가 조회에 I/O 병렬 처리를 사용하기로 선택하는 경우 타스크를 얼마든지 사용할 수 있습니다. SMP 병렬 처리는 허용되지 않습니다.
	*OPTIMIZE	조회 Optimizer는 조회 또는 데이터베이스 파일 키순 액세스 경로 빌드, 리빌드 또는 유지보수를 처리하기 위해 I/O 또는 SMP 병렬 처리에 타스크를 얼마든지 사용하도록 선택할 수 있습니다. SMP 병렬 처리는 시스템 피처인 OS/400용 DB2 대칭 다중처리가 설치된 경우에만 사용될 수 있습니다. 병렬 처리의 사용과 사용되는 타스크의 수는 서버에서 사용할 수 있는 프로세서의 수, 이 작업이 작업이 실행되는 풀에서 갖는 사용 가능한 활동 메모리 양, 조회 또는 데이터베이스 파일 키순 액세스 경로 빌드 또는 리빌드에 대한 예상 경과 시간이 CPU 처리 또는 I/O 자원에 의해 제한되는지 여부에 따라 판별됩니다. 조회 Optimizer는 해당 작업이 풀에서 사용할 수 있는 메모리 양을 기준으로 경과 시간을 최소화하는 구현 방법을 선택합니다.
	*MAX	조회 Optimizer가 I/O 또는 SMP 병렬 처리 중 하나를 사용하여 조회를 처리하기로 선택합니다. SMP 병렬 처리는 시스템 피처인 OS/400용 DB2 대칭 다중처리가 설치된 경우에만 사용됩니다. 조회 Optimizer에 의한 선택사항은 Optimizer가 풀에 있는 모든 활동 메모리가 조회 또는 데이터베이스 파일 키순 액세스 경로 빌드, 리빌드 또는 유지보수를 처리하는 데 사용될 수 있다고 가정하는 점만 제외하고는, 매개변수 값 *OPTIMIZE에 대해 작성한 선택사항과 유사합니다.
	*NONE	데이터베이스 조회 처리 또는 데이터베이스 표 색인 빌드, 리빌드 또는 유지보수에 병렬 처리가 허용되지 않습니다.
	*NUMBER_OF _TASKS nn	단일 조회에 사용될 수 있는 최대 타스크 수를 나타냅니다. 타스크의 수는 이 값이나 표와 연관된 디스크 암(arm)의 수로 완료됩니다.
PARAMETER_MARKER_ CONVERSION	*DEFAULT	디폴트 값은 *YES로 설정됩니다.
	*NO	매개변수 마커로 상수를 적용할 수 없습니다.
	*YES	매개변수 마커로 상수를 적용할 수 있습니다.
QUERY_TIME_LIMIT	*DEFAULT	디폴트 값은 *SYSVAL로 설정됩니다.
	*SYSVAL	해당 작업의 조회 시간 제한이 시스템 값인 QQRYTIMLMT에서 얻어집니다.
	*NOMAX	예상 경과 시간(초)의 최대 수가 없습니다.
	정수 값	조회를 실행하는 데 필요한 예상 경과 시간(초)에 대해 검사되는 최소값을 지정합니다. 예상 경과 시간이 이 값보다 크면 조회가 시작되지 않습니다. 유효한 값의 범위는 0부터 2147352578까지입니다.

표 9. QAQQINI 명령에서 지정된 조회 옵션 (계속)

매개변수	값	설명
REOPTIMIZE_ACCESS_PLAN	*DEFAULT	기존 조회가 다시 최적화되도록 강제 수행하지 마십시오. 단, Optimizer가 이 최적화가 필요하다고 판단할 경우 조회가 다시 최적화됩니다.
	*NO	기존 조회가 다시 최적화되도록 강제 수행하지 마십시오. 단, Optimizer가 이 최적화가 필요하다고 판단할 경우 조회가 다시 최적화됩니다.
	*YES	기존 조회가 다시 최적화되도록 강제 수행하십시오.
	*FORCE	기존 조회가 다시 최적화되도록 강제 수행하십시오.
	*ONLY_REQUIRED	종속적인 이유를 위해 계획이 다시 최적화되는 것을 허용하지 않습니다. 이 경우, 기존 계획이 여전히 유효한 작업 가능한 계획이므로 기존 계획을 계속 사용하십시오. 이것은 재최적화 계획이 이끌어낼 수 있는 모든 성능 이점을 얻지 못함을 나타낼 수 있습니다. 종속적인 이유는 파일 크기 변경, 새로운 색인 등을 포함합니다. 비종속적인 이유는 기존 액세스 계획에서 사용되는 색인의 삭제, 조회 파일이 삭제되고 재작성되는 것 등을 포함합니다.
SQLSTANDARDS_MIXED_ CONSTANT	*DEFAULT	디폴트 값은 *YES로 설정됩니다.
	*YES	SQL IGC 상수가 IGC-OPEN 상수로 취급됩니다.
	*NO	IGC 상수의 자료가 SO-DBCS 자료-SI만 포함하는 경우 상수는 IGC-ONLY로 취급되며, 그렇지 않을 경우에는 IGC-OPEN으로 취급됩니다.
SQL_SUPPRESS_WARNINGS	*DEFAULT	디폴트 값은 *NO로 설정됩니다.
	*YES	명령문 실행 후 SQLCA의 SQLCODE를 조사합니다. SQLCODE > 0인 경우 호출자에게 경고가 리턴되지 않도록 SQLCA를 변경하십시오. SQLCODE를 0으로 설정하면, SQLSTATE는 '00000'이 되고 SQLWARN은 ' '이 됩니다.
	*NO	SQL 경고가 호출자에게 리턴됨을 지정합니다.
SQL_TRANSLATE_ASCII_ TO_JOB	*DEFAULT	디폴트 값은 *NO로 설정됩니다.
	*YES	ASCII SQL문 텍스트를 iSeries 작업의 CCSID로 변환합니다.
	*NO	ASCII SQL문 텍스트를 ASCII CCSID와 연관된 EBCIDIC CCSID로 변환합니다.
STAR_JOIN	*DEFAULT	디폴트 값은 *NO로 설정됩니다.
	*NO	EVI 성형 결합 최적화 지원을 사용할 수 없습니다.
	*FORCE 정수 값	모든 해시 결합 조회에 대해 EVI 성형 결합 최적화 알고리즘을 시도합니다. Distinct List 선택이 EVI가 작성된 열에 존재하는 해시 결합 단계의 경우, Optimizer는 N번째(즉, 정수값) 색인까지의 비용에 관계 없이 해당 EVI가 계획에 추가되도록 허용합니다. 정수 값은 Optimizer가 선택한 계획에 허용될 색인 수를 나타냅니다. Optimizer는 정수값에 도달하거나 색인이 더 이상 존재하지 않을 때까지 EVI가 Distinct List 선택에 대해 빌드되도록 허용합니다. 정수 값에 허용되는 값 범위는 1 - 65534입니다. 정수 값에 값이 지정되지 않으면 65535가 사용됩니다.
	*COST	조회 최적화에서 EVI 성형 결합 지원 사용을 고려하도록(비용을 계산하도록) 허용합니다. Distinct List 선택의 사용 여부는 이 선택을 사용할 경우 어느 정도의 이점이 있는지에 따라 Optimizer가 결정합니다.
SYSTEM_SQL_STATEMENT_ CACHE	*DEFAULT	디폴트 값은 *YES로 설정됩니다.
	*YES	SQL 준비 요구가 처리될 때 SQL 시스템 전반 명령문 캐시를 조사합니다. 일치하는 명령문이 이미 캐시에 있으면 해당 준비 결과를 사용하십시오. 이렇게 하면 어플리케이션이 수행 준비를 더 잘 할 수 있습니다.
	*NO	SQL 준비 요구를 처리할 때 SQL 시스템 전반 명령문 캐시를 조사하지 않도록 지정합니다. 액세스 계획 및 QDT가 스크래치에서 빌드됩니다.

표 9. QAQQINI 명령에서 지정된 조회 옵션 (계속)

매개변수	값	설명
UDF_TIME_OUT	*DEFAULT	대기 시간이 데이터베이스에 의해 판별됩니다. 디폴트는 30초입니다.
	*MAX	UDF가 완료되기까지 데이터베이스가 대기하는 최대 시간
	정수 값	UDF가 완료될 때까지 데이터베이스가 대기해야 하는 시간을 초 단위로 지정합니다. 지정된 값이 데이터베이스 최대 대기 시간보다 크면, 데이터베이스는 최대 대기 시간을 사용합니다. 최소값은 1이고 최대값은 시스템이 정의한 값입니다.
VISUAL_EXPLAIN_ DIAGRAM	*DEFAULT	디폴트 값은 *YES로 설정됩니다.
	*BASIC	다중 액세스 방식이 하나의 색인을 통해 수행되는 경우 이 옵션은 단 하나의 자료 액세스 방식 및 해당 자료 액세스 방식을 나타내는 하나의 아이콘을 표시합니다. 임시 색인을 작성할 때 수행된 자료 액세스 방식은 표시하지 않습니다.
	*DETAIL	다중 액세스 방식이 한 색인을 통해 수행되는 경우 이 옵션은 수행된 각각의 자료 액세스 방식에 대한 아이콘을 표시합니다. 마찬가지로 이 옵션은 임시 색인을 작성할 때 수행된 자료 액세스 방식을 표시합니다.

QAQQINI 조회 옵션 파일 권한 요구사항

QAQQINI는 *PUBLIC *USE 권한과 함께 제공됩니다. 이는 사용자가 조회 옵션 파일을 볼 수만 있고 변경하지는 못하도록 하기 위한 것입니다. 시스템 또는 데이터베이스 관리자만 QAQQINI 조회 옵션 파일에 대한 *CHANGE 권한을 갖는 것이 좋습니다.

조회 옵션 파일은 CHGQRYA CL 명령의 QRYOPTLIB 매개변수에 지정된 라이브러리에 상주하며 항상 조회 Optimizer가 사용합니다. 이는 사용자가 조회 옵션 라이브러리 및 파일에 대한 권한이 없는 경우에도 마찬가지입니다. 이로 인해 시스템 관리자는 추가 보안 메카니즘을 갖게 됩니다.

QAQQINI 파일이 라이브러리 QUSRSYS에 상주하는 경우, 서버의 모든 조회 사용자가 조회 옵션을 사용할 수 있습니다. 조회 옵션을 아무나 삽입, 삭제 또는 갱신하지 못하도록 하려면 시스템 관리자는 해당 파일에 대한 갱신 권한을 *PUBLIC에서 삭제해야 합니다. 이렇게 하면 사용자들이 해당 파일의 자료를 변경할 수 없습니다.

QAQQINI 파일이 사용자 라이브러리에 상주하고 이 라이브러리가 CHGQRYA 명령의 QRYOPTLIB 매개변수에 지정되어 있으면, 조회 옵션은 이 사용자의 작업에 대한 모든 조회 실행에 영향을 미칩니다. 특정 라이브러리에서 조회 옵션을 검색하지 못하도록 하려면 시스템 관리자가 CHGQRYA CL 명령에 대한 권한을 취소하면 됩니다.

QAQQINI 파일의 시스템 제공 트리거

조회 옵션 파일인 QAQQINI 파일은 해당 파일에 대한 변경사항을 처리하기 위해 시스템이 제공하는 트리거 프로그램을 사용합니다. 파일 QAQQINI에서 트리거를 삭제하거나 추가할 수는 없습니다.

QAQQINI 파일을 갱신(INSERT, DELETE 또는 UPDATE 조작)할 때 오류가 발생하면 다음의 SQL0443 진단 메세지가 발행됩니다.

```
Trigger program or external routine detected an error.
```

iSeries용 DB2 UDB 예측 조회 감독자를 사용하여 장기 실행 조회 제어

조회에 대한 예상 또는 예측 실행 시간(경과 실행 시간)이 과도한 경우에는 iSeries용 DB2 Universal Database 예측 조회 감독자가 조회의 시작을 중단할 수 있습니다. 감독자는 조회가 실행되는 도중이 아니라 조회가 실행되기 *전에* 작동합니다. 감독자는 iSeries의 모든 대화식 또는 일괄처리 작업에서 사용될 수 있습니다. 또한, 모든 iSeries용 DB2 Universal Database 조회 인터페이스와 함께 사용할 수도 있으며 SQL 조회와 함께 사용할 수도 있습니다.

조회가 시작되기 전에 조회를 예측하고 중단하는 감독자 기능은 다음과 같은 이유로 매우 중요합니다.

- 장기 실행 조회를 조작하고 어떤 결과를 얻기 전에 조회를 비정상적으로 종료하면 서버 자원이 낭비됩니다.
- 조회 내의 일부 조작은 요구 종료(ENDRQS) CL 명령으로 인터럽트될 수 있습니다. 임시 색인을 작성하거나, GROUP BY 절 없이 열 함수를 사용하는 조회를 작성하는 경우가 이러한 유형의 조회에 대한 예입니다. 사용자가 원하는 것보다 대기 시간이 더 긴 경우에는 이러한 조작을 시작하지 않아야 합니다.

iSeries용 DB2 Universal Database의 감독자는 조회의 예상 실행 시간을 기준으로 사용합니다. 조회의 예상 실행 시간이 사용자가 정의한 시간 제한을 초과하는 경우에는 조회의 초기화가 중단됩니다.

감독자가 사용할 시간 제한을 정의하려면 다음 중 하나를 실행하십시오.

- 조회 속성 변경(CHGQRYA) CL 명령에 조회 시간 제한(QRYTIMLMT) 매개변수를 사용합니다. 이것은 조회 Optimizer가 시간 제한을 찾으려고 시도하는 첫 번째 위치입니다.
- 조회 옵션 파일에서 조회 시간 제한 옵션을 설정합니다. 이것은 조회 Optimizer가 시간 제한을 찾으려고 시도하는 두 번째 위치입니다.
- QQRYTIMLMT 시스템 값을 설정합니다. 각 작업이 CHGQRYA CL 명령에 값 *SYSVAL을 사용하도록 허용하고 조회 옵션 파일을 *DEFAULT로 설정합니다. 이것은 조회 Optimizer가 시간 제한을 찾으려고 시도하는 세 번째 위치입니다.

| 조회 감독자가 조회 Optimizer와 함께 작업하는 방법에 대한 세부사항은 『조회 감독자 작동 방법』을 참조하
| 십시오.

| 예상 조회 감독자를 사용하기 전에, 예상 조회 감독자를 효율적으로 사용하는 데 대한 정보를 보려면 105 페
| 이지의 『조회 감독자 구현 고려사항』, 105 페이지의 『사용자 어플리케이션에 대한 조회 감독자 고려사항: 시
| 간 제한 설정』 및 106 페이지의 『조회 감독자 조회 메세지에 대한 디폴트 응답 제어』를 참조하십시오.

| 예상 조회 감독자를 사용하여 조회 성능을 테스트할 수도 있습니다. 106 페이지의 『조회 감독자를 사용한 성
| 능 테스트』를 참조하십시오.

| 마지막으로 시간 제한을 넘어 실행될 것으로 예상되는 조회를 최소하는 방법을 보려면 105 페이지의 『조회 취
| 소』를 참조하십시오.

조회 감독자 작동 방법

감독자는 조회 Optimizer와 함께 작동합니다. iSeries용 DB2 Universal Database가 조회를 실행하도록 사용자가 요구하면 다음 사항이 발생합니다.

1. Optimizer가 조회 액세스 계획을 평가합니다.

   Optimizer는 평가의 일부로서 조회 실행 시간을 예측 또는 예상합니다. 그러면 조회할 자료에 액세스하여 검색하는 최적의 방법을 판별할 수 있습니다.
2. 예상 실행시간을 현재 작업 또는 사용자 세션에 적용되는 사용자 정의 조회 시간 제한과 비교합니다.
3. 조회에 예측된 실행시간이 조회 시간 제한 이하인 경우에는 조회 감독자가 개입하거나 사용자에게 메세지를 송신할 필요없이 조회가 실행되도록 합니다.
4. 조회 시간 제한이 초과하는 경우에는 조회 메세지 CPA4259가 사용자에게 송신됩니다. 이 메세지는 예상 조회 처리 시간인 XX초가 YY초라는 시간 제한을 초과함을 알려 줍니다.

   주: 사용자가 이 메세지에 응답하지 못하고 조회 결과가 *항상* 종료되도록 메세지에 대한 디폴트 응답을 구성할 수 있습니다.
5. 디폴트 메세지 응답이 사용되지 않는 경우에는 사용자가 다음 중 하나를 선택하여 실행할 수 있습니다.
   - 조회가 실제로 실행되기 전에 조회 요구를 종료합니다.
   - 예측된 실행 시간이 감독자의 시간 제한을 초과하는 경우에도 조회를 계속 실행합니다.

조회 취소

조회 실행 시간이 설정된 시간 제한보다 길 것으로 예상되는 경우, 감독자는 조회 메세지 CPA4259를 발행합니다. 사용자는 다음 중 한 가지 방법으로 메세지에 응답할 수 있습니다.

- C를 입력하여 조회를 취소합니다. 이탈 메세지 CPF427F가 SQL 실행 시간 코드에 발행됩니다. SQL은 SQLCODE -666을 리턴합니다.
- I를 입력하여 시간 제한을 무시하고 조회가 완료될 때까지 계속 실행합니다.

조회 감독자 구현 고려사항

Optimizer가 생성한 시간 제한은 *단지* 예상 값이라는 사실에 주의해야 합니다. 실제 조회 실행시간은 예상 값보다 길 수도 있고 짧을 수도 있지만 이 두 값은 거의 동일해야 합니다. 서버 전체에 대한 시간 제한을 설정할 때는 대개 조회가 실행될 수 있는 최대 허용 시간을 설정하는 것이 가장 좋습니다. 시간 제한을 너무 낮게 설정하면 일부 조회가 완료되지 않고 따라서 어플리케이션이 완료되지 못할 위험이 있습니다. 조회 구성요소를 사용하여 조회 요구를 내부에서 실행하는 기능은 여러 가지가 있습니다. 이러한 요구는 사용자 정의 시간 제한과도 비교됩니다.

사용자 어플리케이션에 대한 조회 감독자 고려사항: 시간 제한 설정

현재 작업 이외의 작업에 시간 제한 설정

현재 작업이 아닌 작업에 시간 제한을 설정할 수 있습니다. CHGQRYA 명령에 JOB 매개변수를 사용하여 탐색할 조회 옵션 파일 라이브러리(QRYOPTLIB) 또는 해당 작업에 대한 특정 QRYTIMLMT를 지정하면 이 작업을 실행할 수 있습니다.

시간 제한을 사용하여 시스템 자원의 균형 조정

소스 작업이 CHGQRYA 명령을 사용한 다음에는 목표 작업에 대한 감독자의 영향이 소스 작업과 상관없습니다. 조회 시간 제한은 해당 작업 또는 사용자 세션의 기간 동안 유효한 상태로 남아 있거나 CHGQRYA 명령으로 시간 제한이 변경될 때까지 유효한 상태로 남아 있습니다. 프로그램 제어 하에서 사용자는 실행되는 어플리케이션 기능, 하루 시간 또는 사용할 수 있는 시스템 자원 양에 따라 조회 시간 제한을 달리 설정할 수 있습니다. 따라서, 임시 조회 요구사항과 시스템 자원의 균형을 조정하려고 할 때 상당한 유연성을 제공합니다.

조회 감독자 조회 메세지에 대한 디폴트 응답 제어

시스템 관리자는 다음과 같은 CHGJOB CL 명령을 사용하여 대화식 사용자가 데이터베이스 조회의 조회 메세지를 무시할 수 있는지 여부를 제어할 수 있습니다.

- CHGJOB CL 명령의 INQMSGRPY 매개변수에 대해 *DFT 값이 지정된 경우, 대화식 사용자는 조회 메세지를 볼 수 없으며 조회는 즉시 취소됩니다.
- CHGJOB CL 명령의 INQMSGRPY 매개변수에 대해 *RQD 값이 지정된 경우, 대화식 사용자는 조회 메세지를 보고 해당 조회에 응답해야 합니다.
- CHGJOB CL 명령의 INQMSGRPY 매개변수에 대해 *SYSRPYL 값이 지정된 경우, 시스템 응답 리스트를 사용하여 대화식 사용자가 조회 메세지를 볼지 여부와 응답이필요한지 여부를 판별합니다. *SYSRPYL 매개변수에 대한 자세한 정보는 iSeries Information Center의 **프로그래밍** 범주에 있는 CL 명령 정보를 참조하십시오. 시스템 응답 리스트의 항목은 사용자 프로파일명, 사용자 ID 또는 프로세스명을 기준으로 서로 다른 디폴트 응답을 사용자 정의하는 데 사용할 수 있습니다. 조회 메세지 CPA4259의 메세지 자료에는 규정된 작업명을 사용할 수 있습니다. 그러면, 키워드 CMPDTA를 사용하여 프로세스 또는 사용자 프로파일에 적용되는 시스템 응답 리스트 항목을 선택할 수 있습니다. 사용자 프로파일명은 10개의 문자로 구성되며 위치 51에서 시작합니다. 프로세스명은 10개의 문자로 구성되며 위치 27에서 시작합니다.
- 다음 예는 디폴트 응답 C가 사용자 프로파일이 'QPGMR'인 작업에 대한 모든 요구를 취소하도록 하는 응답 리스트 요소를 추가합니다.

  ```
  ADDRPYLE  SEQNBR(56) MSGID(CPA4259) CMPDTA(QPGMR      51) RPY(C)
  ```

  다음 예는 디폴트 응답 C가 프로세스명이 'QPADEV0011'인 작업에 대한 모든 요구를 취소하도록 하는 응답 리스트 요소를 추가합니다.

  ```
  ADDRPYLE  SEQNBR(57) MSGID(CPA4259) CMPDTA(QPADEV0011 27) RPY(C)
  ```

조회 감독자를 사용한 성능 테스트

다음과 같이 조회 감독자를 사용하여 조회의 성능을 테스트할 수 있습니다.

1. CHGQRYA 명령을 사용하거나 INI 파일에서 조회 시간 제한을 0(QRYTIMLMT(0))으로 설정하십시오. 이렇게 하면 감독자는 조회 실행에 예상된 시간이 조회 시간 제한을 초과하였다는 조회 메세지를 내보내게 됩니다.
2. 조회 메세지에 메세지 도움말을 프롬트하고 PRTSQLINF(SQL 정보 인쇄) 명령을 실행하여 찾는 것과 동일한 정보를 찾습니다.

조회 감독자는 사용자가 몇 번의 조회를 반복 실행하지 않고도 성능을 최적화할 수 있도록 합니다.

또한, 조회가 취소된 경우, 조회 Optimizer가 액세스 계획을 평가하여 Optimizer 디버그 메세지를 작업 기록부에 송신합니다. 이러한 과정은 작업이 디버그 모드에 있지 않은 경우에도 일어납니다. 그러면 작업 기록부에 있는 Optimizer 조정 메세지를 검토하여, 최적의 조회 성능을 얻기 위해 추가 조정이 필요한지 여부를 확인할 수 있습니다. 이는 사용자가 실제로 조회 실행을 완료하지 않고도 Optimizer를 통해 무엇이 더 잘 수행하는지 판별하기 위해 다른 속성, 색인 및/또는 구문으로 조회를 다양하게 변경해 볼 수 있게 합니다. 이는 실제 자료 조회가 실제적으로 수행되지 않으므로 시스템 자원 사용을 절약합니다. 조회 대상 표에 많은 행이 포함되어 있는 경우에는, 시스템 자원에 저장된 자료가 많음을 나타냅니다.

성능 테스트에 이러한 기법을 사용할 때는 조회가 실행되기 전에 모든 조회 요구가 중단되므로 주의해야 합니다. 이는 특히 단일 조회 단계에서 구현되지 않는 조회에 중요한 사항입니다. 이러한 유형의 조회에서는 개별적인 여러 조회 요구가 발행된 다음, 최종 결과가 리턴되기 전에 각 조회 결과가 종합됩니다. 중간 단계 중 어느 한 단계에서 조회를 중단하면 사용자는 해당 중간 단계에 대한 성능 정보만 얻을 수 있으며 전체 조회에 대한 정보는 얻을 수 없게 됩니다.

조회 시간 제한 설정 예

조회 옵션 파일 QAQQINI를 사용하는 현재 작업 또는 사용자 세션에 조회 시간 제한을 설정하려면, CHGQRYQ 명령의 QRYOPTLIB 매개변수를 QAQQINI 파일이 존재하는 사용자 라이브러리로 지정한 다음, 매개변수를 QUERY_TIME_LIMIT로 지정하고 값을 유효한 조회 시간 제한으로 설정합니다. 조회 옵션 파일 설정에 대한 자세한 정보는 95 페이지의『조회 옵션 파일 QAQQINI를 사용하여 조회를 동적으로 제어』를 참조하십시오.

조회 시간 제한을 45초로 설정하려면 다음의 CHGQRYA 명령을 사용하십시오.

```
CHGQRYA   JOB(*) QRYTIMLMT(45)
```

이렇게 하면 조회 시간 제한이 45초로 설정됩니다. 사용자가, 예상 실행 시간이 45초 이하인 조회를 실행할 때는 아무런 개입 없이 조회가 실행됩니다. 시간 제한은 해당 작업 또는 사용자 세션 기간 동안 유효한 상태로 남아 있거나 CHGQRYA 명령으로 시간 제한을 변경할 때까지 유효한 상태로 남아 있습니다.

예를 들어, 조회 Optimizer가 조회 실행시간을 135초로 예상하였다면, 135초라는 예상 실행시간이 조회 시간 제한인 45초를 초과한다는 메세지가 사용자에게 송신됩니다.

현재 작업 이외의 작업에 대해 조회 시간 제한을 설정하거나 변경하려면, JOB 매개변수를 사용하여 CHGQRYA 명령을 실행합니다. 작업 123456/USERNAME/JOBNAME에 대해 조회 시간 제한을 45초로 설정하려면 다음의 CHGQRYA 명령을 사용하십시오.

```
CHGQRYA   JOB(123456/USERNAME/JOBNAME) QRYTIMLMT(45)
```

이렇게 하면 작업 123456/USERNAME/JOBNAME에 대한 조회 시간 제한이 45초로 설정됩니다. 예상 실행 시간이 45초 이하인 조회를 작업 123456/USERNAME/JOBNAME이 실행하려고 하면 조회는 아무런 개입 없이 실행됩니다. 조회의 예상 실행시간이 45초를 초과하는 경우(예: 50초)에는 50초라는 예상 실행시간이 조회 시간 제한인 45초를 초과한다는 메세지가 사용자에게 송신됩니다. 시간 제한은 작업 123456/USERNAME/JOBNAME의 기간 동안 유효한 상태로 남아 있거나 CHGQRYA 명령으로 작업 123456/USERNAME/JOBNAME에 대한 시간 제한을 변경할 때까지 유효한 상태로 남아 있습니다.

조회 시간 제한을 QQRYTIMLMT 시스템 값으로 설정하거나 변경하려면 다음의 CHGQRYA 명령을 사용하십시오.

```
CHGQRYA QRYTIMLMT(*SYSVAL)
```

QQRYTIMLMT 시스템 값은 해당 작업 또는 사용자 세션 기간 동안 유효한 상태로 남아 있거나 CHGQRYA 명령으로 시간 제한을 변경할 때까지 유효한 상태로 남아 있습니다. 이것은 CHGQRYA 명령의 디폴트 작동입니다.

주: 조회 시간 제한은 INI 파일에 설정되거나 SYSVAL 명령을 사용하여 설정될 수 있습니다.

조회에 대한 병렬 처리 제어

병렬 처리를 설정 및 설정 해제할 수 있습니다. DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에는, SMP(대칭 다중처리)를 설정 및 설정 해제할 수도 있습니다.

- 시스템 범위 제어의 경우에는 시스템 값 QQRYDEGREE를 사용합니다.
- 작업 레벨 제어의 경우에는, CHGQRYA 명령에 DEGREE 매개변수를 사용하거나 조회 옵션 파일 QAQQINI의 PARALLEL_DEGREE 옵션을 사용합니다.

서버 또는 특정 작업에 병렬화가 작동되었더라도, 작업에서 실행되는 개별 조회가 실제로는 병렬 방식을 사용하지 않을 수도 있습니다. 이것은 기능 제한 때문일 수도 있으며 Optimizer가 더 빠른 실행을 위해 비병렬 방법을 선택했기 때문일 수도 있습니다. 병렬 액세스 방식 각각의 성능 특징 및 제한사항을 설명한 이전 섹션을 참조하십시오. 사용할 수 있는 병렬 방식은 다음과 같습니다.

- 병렬 표 스캔 방식
- 병렬 색인 스캔 키 선택 방식
- 병렬 색인 스캔 키 위치지정 방식
- 병렬 색인 전용 액세스 방식(병렬 및 비병렬)
- 병렬 해싱 방식(병렬 및 비병렬)
- 병렬 비트맵 처리 방식

병렬 액세스 방식을 사용하여 처리되는 조회는 주 기억장치, CPU 및 디스크 자원을 많이 사용하기 때문에, 병렬 처리를 사용하는 조회의 수를 제한하고 제어해야 합니다.

조회에 대한 시스템 범위 병렬 처리 제어

QQRYDEGREE 시스템 값을 사용하여 서버의 병렬 처리를 제어할 수 있습니다. 시스템 값의 현재 값은 다음의 CL 명령을 사용하여 표시하거나 수정할 수 있습니다.

- WRKSYSVAL - 시스템 값에 대한 작업
- CHGSYSVAL - 시스템 값 변경
- DSPSYSVAL - 시스템 값 표시
- RTVSYSVAL - 시스템 값 검색

QQRYDEGREE의 특수 값은 서버의 모든 작업에 대해 병렬 처리가 디폴트로 허용되는지 여부를 제어합니다. 가능한 값은 다음과 같습니다.

***NONE**

데이터베이스 조회 처리에 병렬 처리가 허용되지 않습니다.

***IO**

조회에 I/O 병렬 처리가 허용됩니다.

***OPTIMIZE**

조회 Optimizer가 조회를 처리하기 위해 I/O 또는 SMP 병렬 처리에 타스크를 얼마든지 사용하기로 선택할 수 있습니다. SMP 병렬 처리는 DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용될 수 있습니다. 조회 Optimizer는 해당 작업이 풀(pool)의 메모리를 사용할 수 있는 양을 기준으로 경과 시간을 최소화하기 위해 병렬 처리를 선택하여 사용합니다.

***MAX**

조회 Optimizer가 조회를 처리하는 데 I/O 또는 SMP 병렬 처리를 사용하기로 선택할 수 있습니다. SMP 병렬 처리는 DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용할 수 있습니다. 조회 Optimizer가 선택하는 방법은 Optimizer가 풀(pool)의 모든 활동 메모리를 조회를 처리하는 데 사용할 수 있다고 가정하는 점만 제외하고는 매개변수 값 *OPTIMIZE에서 선택되는 방법과 유사합니다.

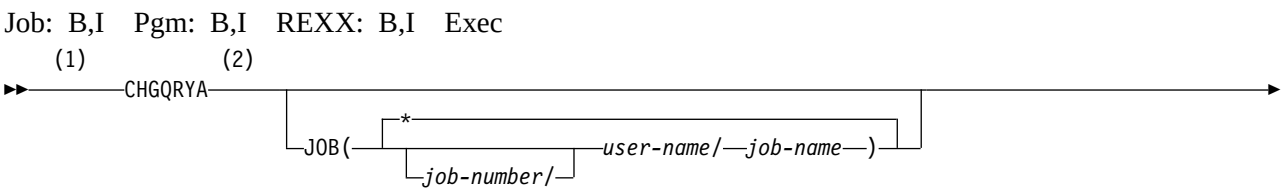
QQRYDEGREE 시스템 값의 디폴트 값은 *NONE이므로, 서버에서 디폴트로 작업에 대한 병렬 조회 처리가 실행되기를 원하는 경우에는 이 값을 변경해야 합니다.

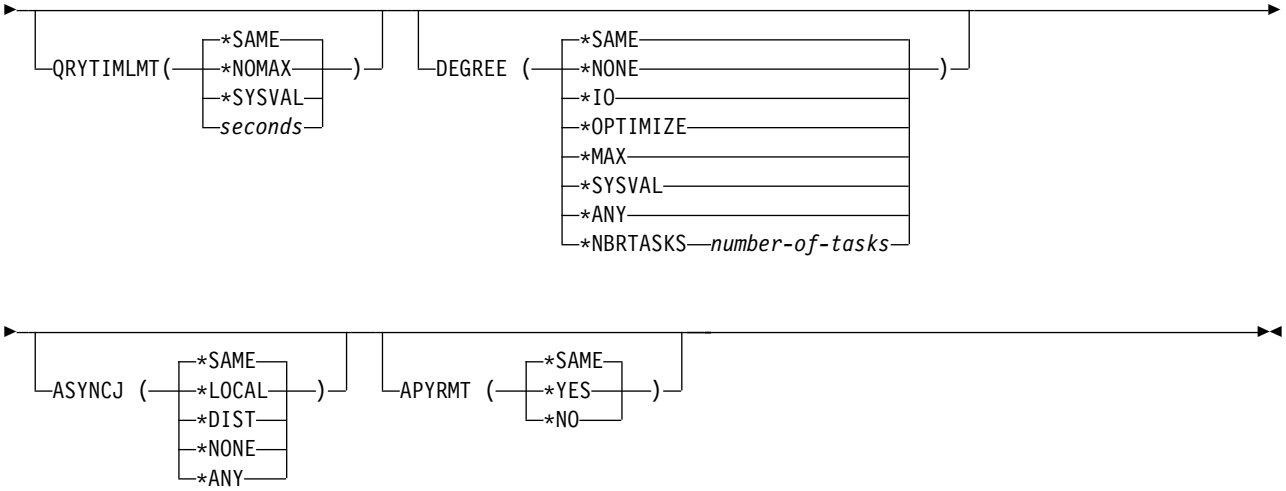
이 시스템 값을 변경하면 DEGREE 조회 속성이 *SYSVAL인 서버에서 앞으로 실행될 작업 또는 현재 실행 중인 작업 모두에 영향을 주게 됩니다. 그러나, 이미 시작된 조회나 재사용 가능한 ODP를 사용하는 조회는 영향을 받지 않습니다.

조회에 대한 작업 레벨 병렬 처리 제어

조회 속성 변경(CHGQRYA) 명령의 DEGREE 매개변수를 사용하거나 INI 파일에서 작업 레벨의 조회 병렬 처리를 제어할 수도 있습니다. 허용되는 병렬 처리 옵션과 작업에서 데이터베이스 조회 실행 지정시 사용될 수 있는 타스크의 수(선택 사항)를 지정할 수 있습니다. 대화식 작업의 CHGQRYA 명령에 프롬트하여 DEGREE 조회 속성의 현재 값을 표시할 수 있습니다.

DEGREE 속성을 변경해도 이미 시작된 조회나 재사용 가능한 ODP를 사용하는 조회는 영향을 받지 않습니다.





주:

1 값 *ANY는 값 *IO와 같습니다.

2 이 점 앞의 모든 매개변수는 위치지정 형식으로 지정할 수 있습니다.

DEGREE 키워드의 매개변수 값은 다음과 같습니다.

***SAME**
병렬 차수 조회 속성이 변경되지 않습니다.

***NONE**
데이터베이스 조회 처리에 병렬 처리가 허용되지 않습니다.

***IO**
데이터베이스 조회 Optimizer가 조회에 I/O 병렬 처리를 사용하기로 선택하는 경우 타스크를 얼마든지 사용할 수 있습니다. SMP 병렬 처리는 허용되지 않습니다.

***OPTIMIZE**
조회 Optimizer가 조회를 처리하기 위해 I/O 또는 SMP 병렬 처리에 타스크를 얼마든지 사용하기로 선택할 수 있습니다. SMP 병렬 처리는 DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용할 수 있습니다. 병렬 처리 사용 여부와 사용되는 타스크의 수는 서버에서 사용할 수 있는 프로세서의 수, 작업이 실행되는 풀(pool)에서 해당 작업이 사용할 수 있는 활동 메모리 양 CPU 처리 또는 I/O 자원이 조회의 예상 경과 시간을 제한하는지 여부에 따라 판별됩니다. 조회 Optimizer는 해당 작업이 풀에서 사용할 수 있는 메모리 양을 기준으로 경과 시간을 최소화하는 구현 방법을 선택합니다.

***MAX**
조회 Optimizer가 조회를 처리하는 데 I/O 또는 SMP 병렬 처리를 사용하기로 선택할 수 있습니다. SMP 병렬 처리는 DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용할 수 있습니다. 조회 Optimizer가 선택하는 방법은 풀(pool)의 모든 활동 메모리를 조회를 처리하는 데 사용할 수 있다고 가정하는 점만 제외하고는, 매개변수 값 *OPTIMIZE에서 선택하는 방법과 유사합니다.

***NBRTASKS** *number-of-tasks*

조회 Optimizer가 SMP 병렬 처리를 사용하여 조회를 처리하기로 선택하는 경우에 사용되는 타스크 수를 지정합니다. I/O 병렬화 또한 허용됩니다. SMP 병렬 처리는 DB2 UDB 대칭형 멀티프로세싱 피처가 설치된 경우에만 사용할 수 있습니다.

서버에서 사용할 수 있는 프로세서 수보다 타스크 수를 적게하면 주어진 조회를 실행하기 위해 사용되는 프로세서 수가 제한됩니다. 반면, 타스크 수가 프로세서 수보다 많은 경우에는 서버에서 사용할 수 있는 모든 프로세서를 조회 실행에 사용할 수 있게 됩니다. 그러나, 타스크의 수가 너무 많으면 활동 메모리의 과도한 실행과 모든 타스크를 관리하기 위한 오버헤드 비용으로 인해 성능이 저하될 수 있습니다.

***SYSVAL**

사용되는 처리 옵션이 시스템 값의 현재 값인 QQRYDEGREE로 설정되도록 지정합니다.

작업에 대한 DEGREE 속성의 초기 값은 *SYSVAL입니다.

통계 관리자를 사용하여 조회 분석

조회 Optimizer는 통계 정보 및 기타 요소를 사용하여 조회에 대한 최적의 액세스 계획을 결정할 수 있습니다. 값으로서, 이 통계 정보는 정확하고 완벽해야 합니다. 조회 Optimizer는 표의 통계 정보를 기준으로 액세스 계획을 선택하므로 정보를 최신으로 유지해야 합니다. 많은 플랫폼에서 통계 수집은 데이터베이스 관리자의 책임 하에 수행되는 수동 프로세스입니다. iSeries 서버에서 데이터베이스 통계 수집 프로세스는 자동으로 처리되며, 통계를 수동으로 관리할 수 있는 경우가 있을 수 있지만 원칙적으로는 수동으로 갱신할 필요가 없습니다. 이번 릴리스에 포함된 iSeries Navigator의 데이터베이스 통계 기능을 사용하면 표에 대한 통계 정보를 관리할 수 있습니다.

주: 수동으로 통계를 수집하기로 결정하고 iSeries Navigator에서 통계를 시스템이 자동 갱신하지 못하도록 수동으로 유지보수하도록 설정한 경우 또는 자동 갱신 프로세스의 속도를 높이려면 다음 경우에 통계를 갱신해야 합니다.

- 표가 로드되거나 재구성될 경우
- 많은 수의 행이 삽입, 갱신 또는 삭제된 경우
- 새 컬럼이 표에 추가되었을 경우
- Visual Explain의 Statistics Advisor에서 통계를 작성하거나 갱신하도록 권장하는 경우

통계 관리자 API

iSeries Navigator의 통계 기능을 구현하는 데에는 다음 API가 사용됩니다.

- 요구된 통계 콜렉션 취소(QDBSTCRS, QdbstCancelRequestedStatistics)는 요구되었지만 아직 완료되지 않았거나 성공적으로 완료되지 않은 통계 콜렉션을 즉시 취소합니다.
- 통계 콜렉션 삭제(QDBSTDS, QdbstDeleteStatistics)는 이미 완료된 통계 콜렉션을 즉시 삭제합니다.
- 요구된 통계 콜렉션 나열(QDBSTLRS, QdbstListRequestedStatistics)은 백그라운드 통계 콜렉션을 요구했지만 아직 완료되지 않은 모든 열, 열의 조합 및 파일 멤버를 나열합니다.
- 통계 콜렉션 세부사항 나열(QDBSTLDS,)은 하나의 통계 콜렉션에 대한 추가 통계 자료를 나열합니다.

- 통계 콜렉션 나열(QDBSTLS, QdbstListStatistics)은 사용 가능한 통계가 있는 지정된 파일 멤버에 대한 모든 열 및 열 조합을 나열합니다.
- 통계 콜렉션 요구(QDBSTRS, QdbstRequestStatistics)를 사용하면 특정 파일 멤버의 지정된 열 집합에 대한 하나 이상의 통계 콜렉션을 요구할 수 있습니다.
- 통계 콜렉션 갱신(QDBSTUS, QdbstUpdateStatistics)을 사용하면 속성을 갱신하고 기존 단일 통계 콜렉션 자료를 화면정리할 수 있습니다.

iSeries Navigator를 사용하여 통계 정보 관리

iSeries Navigator 통계 기능을 사용하여 표에 대한 통계 정보를 관리할 수 있습니다. 필요한 통계 정보가 있으면 이를 통해 특정 표에 대해 조회가 잘못 수행된 원인을 평가할 수 있습니다. iSeries Navigator를 사용하여 통계를 관리하는 방법을 알려면 다음 주제 중 하나를 선택하십시오.

- 『iSeries Navigator를 사용하여 통계 작성』
- 『iSeries Navigator를 사용하여 표나 별명에 대한 통계 자료 보기』
- 113 페이지의 『iSeries Navigator를 사용하여 통계 갱신』

iSeries Navigator를 사용하여 통계 작성

iSeries Navigator를 사용하여 새 통계를 작성하려면, 다음 단계를 따르십시오.

1. iSeries Navigator를 여십시오.
2. iSeries Navigator 창에서, 사용할 서버를 펼치십시오.
3. **데이터베이스**를 펼치십시오.
4. 표 또는 별명이 저장된 라이브러리가 있는 데이터베이스를 펼치십시오.
5. 표나 별명을 마우스 오른쪽 단추로 클릭한 후 **통계 자료**를 선택하십시오.
6. **통계 자료** 대화 상자에서 **신규**를 클릭하십시오.
7. 사용 가능한 열 리스트의 **신규 통계** 대화 상자에서 통계를 수집할 열을 선택하십시오. **추가**를 클릭하십시오.

iSeries Navigator를 사용하여 표나 별명에 대한 통계 자료 보기

iSeries Navigator를 사용하여 표나 별명에 대한 통계를 보려면 다음 단계를 수행하십시오.

1. iSeries Navigator를 여십시오.
2. iSeries Navigator 창에서, 사용할 서버를 펼치십시오.
3. **데이터베이스**를 펼치십시오.
4. 표 또는 별명이 저장된 라이브러리가 있는 데이터베이스를 펼치십시오.
5. 표나 별명을 마우스 오른쪽 단추로 클릭한 후 **통계 자료**를 선택하십시오.

이 대화 상자에서 통계 자료 세부사항을 보거나 통계를 갱신할 수 있습니다.

iSeries Navigator를 사용하여 통계 갱신

iSeries Navigator를 사용하여 표나 별명에 대한 통계를 갱신하려면 다음 단계를 수행하십시오.

1. iSeries Navigator를 여십시오.
2. iSeries Navigator 창에서, 사용할 서버를 펼치십시오.
3. **데이터베이스**를 펼치십시오.
4. 표 또는 별명이 저장된 라이브러리가 있는 데이터베이스를 펼치십시오.
5. 표나 별명을 마우스 오른쪽 단추로 클릭한 후 **통계 자료**를 선택하십시오.
6. **통계 자료** 대화 상자에서 **갱신**을 클릭하십시오.

조회 최적화 툴: 비교 표

PRTSQLINF	STRDBG 또는 CHGQRYA	파일 기반 모니터	메모리 기반 모니터	Visual Explain
조회를 실행하지 않고도 사용할 수 있음(액세스 계획 작성 후)	조회 실행시에만 사용할 수 있음	조회 실행시에만 사용할 수 있음	조회 실행시에만 사용할 수 있음	조회 설명시에만 사용할 수 있음
실행 여부와는 상관없이 SQL 프로그램의 모든 조회에 대해 표시됨	실행되는 조회에 대해서만 표시됨	실행되는 조회에 대해서만 표시됨	실행되는 조회에 대해서만 표시됨	설명되는 조회에 대해서만 표시됨
호스트 변수 구현에 대한 정보	호스트 변수 구현에 대한 제한된 정보	호스트 변수, 구현 및 값에 대한 모든 정보	호스트 변수, 구현 및 값에 대한 모든 정보	호스트 변수, 구현 및 값에 대한 모든 정보
프로그램, 패키지 또는 서비스 프로그램이 있는 SQL 사용자만 사용할 수 있음	모든 조회 사용자(OPNQRYF, SQL, QUERY/400)가 사용할 수 있음	모든 조회 사용자(OPNQRYF, SQL, QUERY/400)가 사용할 수 있음	SQL 인터페이스에만 사용할 수 있음	iSeries Navigator 데이터베이스 및 API 인터페이스를 통해 사용할 수 있음
메세지가 스풀 파일로 인쇄됨	메세지가 작업 기록부에 표시됨	성능 행이 데이터베이스 표에 기록됨	성능 정보가 메모리에 수집된 다음 데이터베이스 표에 기록됨	iSeries Navigator를 통해 정보가 가시적으로 표시됨
메세지를 부속 조회 또는 합집합 연산을 사용하는 조회와 연결하기가 쉬움	메세지를 부속 조회 또는 합집합 연산을 사용하는 조회와 연결하기가 어려움	모든 조회, 부속 조회 및 구체화된 보기를 고유하게 식별	반복되는 조회 요구를 요약	조회 및 연관 정보의 구현을 보기 쉬움

제 5 장 대형 표에 빨리 액세스하기 위해 색인 사용

iSeries용 DB2 Universal Database는 표에 액세스하기 위해 다음과 같은 두가지 기본적인 방법 즉, 표 스캔 (순차) 및 색인 기준 (직접) 검색을 제공합니다. 색인 기준 검색은 대개 표 스캔보다 효율적입니다. 그러나, 아주 많은 비율의 페이지를 검색할 경우 표 스캔이 색인 기준 검색보다 더 효율적입니다.

iSeries용 DB2 Universal Database는 표의 자료를 액세스하기 위해 색인을 사용할 수 없는 경우, 표의 모든 자료를 읽어야 합니다. 아주 큰 표는 특수한 성능 문제를 야기하는 데 표에 있는 모든 자료를 검색해야 하므로 많은 비용이 소요됩니다. 다음 주제는 iSeries용 DB2 Universal Database가 사용할 수 있는 색인을 활용할 수 있는 코드 설계에 도움이 되는 제안을 제공합니다.

- 『효율적인 색인을 위한 코딩: 숫자 변환 회피』
- 116 페이지의 『효율적인 색인을 위한 코딩: 연산식 회피』
- 116 페이지의 『효율적인 색인을 위한 코딩: 문자 스트링 채우기 회피』
- 117 페이지의 『효과적인 색인을 위한 코딩: % 또는 _로 시작하는 유사 패턴의 사용을 피함』
- 117 페이지의 『효율적인 색인을 위한 코딩: iSeries용 DB2 UDB가 색인을 사용하지 않는 인스턴스 인식』

색인 사용에 대한 추가 정보:

정렬 순서 표에 대한 색인 작업 방법에 대한 정보는 118 페이지의 『효율적인 색인을 위한 코딩: 색인을 정렬 순서와 함께 사용』을 참조하십시오.

효율적인 색인의 코딩 예는 119 페이지의 『색인 예』를 참조하십시오.

색인을 작성하는 여러 가지 방법에 대한 정보는 iSeries Information Center에서 색인 작성 주제를 참조하십시오.

효율적인 색인을 위한 코딩: 숫자 변환 회피

열 값과 호스트 변수(또는 상수 값)를 비교할 때, 동일한 자료 유형 및 속성을 지정하십시오. iSeries용 DB2 Universal Database는 호스트 변수 또는 상수 값의 정밀도가 열의 정밀도 보다 더 큰 경우 명명된 열에 대한 색인을 사용하지 않습니다. 비교되고 있는 두 개의 항목이 다른 자료 유형을 갖는 경우, iSeries용 DB2 Universal Database는 한 값 또는 다른 값을 변환해야 하는 데 이런 경우 제한된 기계 정밀도로 인해 결과가 부정확할 수 있습니다. 비교되고 있는 열과 상수에 대한 문제를 피하려면 다음을 사용하십시오.

- 동일한 자료 유형
- 동일한 스케일(적용 가능한 경우)
- 동일한 정밀도(적용 가능한 경우)

예를 들면, EDUCLVL은 반어 정수 값(SMALLINT)입니다. SQL을 사용하는 경우 다음과 같이 지정하십시오.

```
... WHERE EDUCLVL < 11 AND
                  EDUCLVL >= 2
```

또는

```
... WHERE EDUCLVL < 1.1E1 AND
                  EDUCLVL > 1.3
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
... QRYSLT('EDUCLVL *LT 11 *AND ENUCLVL *GE 2')
```

또는

```
... QRYSLT('EDUCLVL *LT 1.1E1 *AND EDUCLVL *GT 1.3')
```

색인이 EDUCLVL 열에 대해 작성된 경우 상수의 정밀도가 열의 정밀도보다 더 높으므로 Optimizer는 두 번째 예의 색인을 사용하지 않습니다. 첫 번째 예에서, Optimizer는 정밀도가 같으므로 색인 사용을 고려합니다.

효율적인 색인을 위한 코딩: 연산식 회피

행 선택 술부의 열에 비교되는 피연산자로서 연산식을 사용하지 마십시오. Optimizer는 연산식에 비교되고 있는 열에 대해 색인을 사용하지 않습니다. 이는 열에 대한 색인을 쓸모없게 만들지 않을 수도 있지만, 색인에 대한 추정 및 색인 스캔 키 위치지정 사용을 못하게 합니다. 무엇보다도, 조회 최적화에 유용한 통계를 사용하고 추출하는 기능을 잃게 됩니다. 예를 들어, SQL을 사용하는 경우, 다음과 같이 지정하십시오.

```
... WHERE SALARY > 16500
```

또는

```
... WHERE SALARY > 15000*1.1
```

효율적인 색인을 위한 코딩: 문자 스트링 채우기 회피

고정 길이 문자 스트링 열 값을 호스트 변수 또는 상수 값과 비교할 때 동일한 자료 길이를 사용하십시오. iSeries용 DB2 Universal Database는 상수 값 또는 호스트 변수가 열 길이 보다 더 긴 경우 색인을 사용하지 않습니다.예를 들어, EMPNO는 CHAR(6)이고 DEPTNO는 CHAR(3)입니다. 예를 들어, SQL을 사용하는 경우, 다음과 같이 지정하십시오.

```
... WHERE EMPNO > '000300' AND
                  DEPTNO < 'E20'
```

또는

```
... WHERE EMPNO > '000300 ' AND
                  DEPTNO < 'E20 '
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
... QRYSLT('EMPNO *GT "000300" *AND DEPTNO *LT "E20"')
```

또는

```
... QRYSLT('EMPNO *GT "000300" *AND DEPTNO *LT "E20"')
```

효과적인 색인을 위한 코딩: % 또는 _로 시작하는 유사 패턴의 사용을 피함

LIKE(OPNQRYF %WLDCRD) 술부 패턴으로 사용될 때, 퍼센트 기호(%), 밑줄(_)은 선택하려는 행의 열 값과 유사한 문자 스트링을 지정합니다. 다음과 같이 문자 스트링의 끝 또는 중간에 문자를 표시하는 데 사용될 때 색인을 활용할 수 있습니다. 예를 들어, SQL을 사용하는 경우, 다음과 같이 지정하십시오.

```
... WHERE LASTNAME LIKE 'J%SON%'
```

OPNQRYF 명령을 사용할 때, 다음과 같이 지정하십시오.

```
... QRYSLT('LASTNAME *EQ %WLDCRD(''J*SON*'')')
```

그러나, 문자 스트링의 시작에서 사용될 경우, 색인 스캔-키 위치지정을 사용하여 스캔되는 행의 수를 제한하기 위해 iSeries용 DB2 Universal Database가 LASTNAME 열에 정의되어 있을 수 있는 색인을 사용하지 못하게 할 수 있습니다. 단, 색인 스캔 키 선택은 사용할 수 있습니다. 예를 들어, 다음 조회에서 색인 스캔 키 선택은 사용할 수 있으나, 색인 스캔 키 위치지정은 사용할 수 없습니다.

SQL에서:

```
... WHERE LASTNAME LIKE '%SON'
```

OPNQRYF에서:

```
... QRYSLT('LASTNAME *EQ %WLDCRD(''*SON'')')
```

이상적으로, %를 사용하는 패턴을 사용하지 않음으로써 술부에서 키 처리를 수행할 때 최적 성능을 확보할 수 있습니다. 조회 또는 어플리케이션에 대해 제어를 수행할 수 있는 경우, 색인 스캔 키 위치지정을 사용할 수 있도록, 검색할 부분 스트링을 확보하기 위한 시도를 해야 합니다.

예를 들어, 이름 ″Smithers″를 찾고 있는데 ″S%″만 입력했다면, 이 조회는 ″S″로 시작하는 모든 이름을 리턴합니다. 그런 다음, ″Smi%″로 시작하는 모든 이름을 리턴하도록 조회를 조정할 것입니다. 따라서, 강제로 부분 스트링을 사용하면, 더 좋은 성능이 장기간 나타날 수 있습니다.

효율적인 색인을 위한 코딩: iSeries용 DB2 UDB가 색인을 사용하지 않는 인스턴스 인식

iSeries용 DB2 Universal Database는 다음과 같은 인스턴스에서 색인을 사용하지 않습니다.

- 갱신이 예상되는 열의 경우. 예를 들어 SQL를 사용하는 경우, 사용자의 프로그램은 다음을 포함합니다.

```
EXEC SQL
  DECLARE DEPTEMP CURSOR FOR
  SELECT EMPNO, LASTNAME, WORKDEPT
      FROM CORPDATA.EMPLOYEE
    WHERE (WORKDEPT = 'D11' OR
```

```
              WORKDEPT = 'D21') AND
             EMPNO = '000190'
      FOR UPDATE OF EMPNO, WORKDEPT
  END-EXEC.
```

OPNQRYF 명령을 사용하는 경우, 예를 들면 다음과 같습니다.

```
OPNQRYF FILE((CORPDATA/EMPLOYEE)) OPTION(*ALL)
   QRYSLT('(WORKDEPT *EQ ''D11'' *OR WORKDEPT *EQ ''D21'')
   *AND EMPNO *EQ ''000190''')
```

사원의 부서를 갱신하려 하지 않아도, iSeries용 DB2 Universal Database는 WORKDEPT 키가 있는 색인을 사용할 수 없습니다.

iSeries용 DB2 Universal Database는 색인 내에 사용된 갱신 가능한 모든 열이 같은 연산자가 있는 분리 가능한 선택 술부로서 조회 내에 사용되는 경우 색인을 사용할 수 있습니다. 앞의 예에서, iSeries용 DB2 Universal Database는 EMONO 키가 있는 색인을 사용합니다.

iSeries용 DB2 Universal Database는 FOR UPDATE OF 열 리스트가 갱신하려는 열 *WORKDEPT*를 명명하는 경우에만 보다 효율적으로 작동할 수 있습니다. 따라서, 열을 갱신하려 하지 않으면 FOR UPDATE 열 리스트의 열을 지정하지 마십시오.

동적 SQL로 인해 갱신할 수 있는 커서가 있거나 FOR UPDATE 절이 지정되지 않은 경우, 프로그램에 UPDATE문이 들어 있으므로 모든 열을 갱신할 수 있습니다.

- 동일한 행의 다른 열과 비교되는 열의 경우. 예를 들어, SQL을 사용할 때, 프로그램은 다음을 포함할 수 있습니다.

```
  EXEC SQL
 DECLARE DEPTDATA CURSOR FOR
    SELECT WORKDEPT, DEPTNAME
        FROM CORPDATA.EMPLOYEE
      WHERE WORKDEPT = ADMRDEPT
  END-EXEC.
```

OPNQRYF 명령을 사용하는 경우, 예를 들면 다음과 같습니다.

```
OPNQRYF FILE(EMPLOYEE) FORMAT(FORMAT1)
   QRYSLT('WORKDEPT *EQ ADMRDEPT')
```

*WORKDEPT*에 대한 색인 및 *ADMRDEPT*에 대한 색인이 있을지라도 iSeries용 DB2 Universal Database는 어느 색인도 사용하지 않습니다. 표의 모든 행을 살펴야 하므로 색인의 장점이 없습니다.

효율적인 색인을 위한 코딩: 색인을 정렬 순서와 함께 사용

다음 섹션은 정렬 순서 표로 색인 작업 방법에 대한 유용한 정보를 제공합니다.

- 119 페이지의 『효율적인 색인을 위한 코딩: 선택, 결합 또는 그룹화와 함께 색인 정렬 순서 사용』
- 119 페이지의 『효율적인 색인을 위한 코딩: 순서화』

정렬 순서 표 작업 방법에 대한 자세한 정보는 SQL 참조서의 "정렬 순서" 주제를 참조하십시오.

효율적인 색인을 위한 코딩: 선택, 결합 또는 그룹화와 함께 색인 정렬 순서 사용

기존 색인을 사용하기 전에 iSeries용 DB2 Universal Database는 열의 속성(선택, 결합 또는 그룹화)이 기존 색인의 키 열 속성에 대응하는지 확인합니다. 정렬 순서 표는 비교되어야 하는 추가적 속성입니다.

조회(SRTSEQ 및 LANGID 매개변수로 지정된)와 연관된 정렬 순서 표는 기존 색인이 빌드되었던 정렬 순서 표에 대응하여야 합니다. iSeries용 DB2 Universal Database는 정렬 순서 표를 비교합니다. 대응하지 않는 경우, 기존 색인은 사용될 수 없습니다.

그러나 여기에는 예외가 있습니다. 조회와 연관된 정렬 순서 표가 고유 가중치 정렬 표인 경우, (*HEX 포함) iSeries용 DB2 Universal Database는 정렬 순서 표가 다음 연산자를 사용하는 선택, 결합 또는 그룹화 열에 대해 지정되지 않을지라도 작동합니다.

- 같음(=) 연산자
- 같지 않음(^= 또는 <>) 연산자
- LIKE 술부(OPNQRYF %WLDCRD 및 *CT)
- IN 술부(OPNQRYF %VALUES)

이러한 조건이 참인 경우, iSeries용 DB2 Universal Database는 키 열이 열에 대응하는 기존 색인을 사용할 수 있으며 이 경우 색인은 다음과 같습니다.

- 색인은 정렬 순서 표를 포함하지 않거나
- 색인은 고유 가중치 정렬 순서 표를 포함합니다.

주: 표는 조회와 연관된 고유 가중치 정렬 순서 표에 대응하지 않아도 됩니다.

주: 복수 색인이 표에 대해 사용될 때 비트맵 처리에는 특별한 고려사항이 있습니다. 둘 이상의 색인이 조회 선택에서 참조되는 색인간에 공통 키 열을 갖는 경우 해당 색인은 같은 정렬 순서 표를 사용하거나 정렬 순서 표를 사용하지 않아야 합니다.

효율적인 색인을 위한 코딩: 순서화

Optimizer가 순서화 요구를 만족시키기 위해 정렬 수행을 선택하지 않는 경우 색인과 연관된 정렬 순서 표는 조회와 연관된 정렬 순서 표에 대응하여야 합니다.

정렬이 사용되는 경우, 정렬되는 동안 변환이 실행됩니다. 정렬이 정렬 순서 요구사항을 처리하므로 iSeries용 DB2 Universal Database는 선택 범주에 부합되는 기존 색인을 사용할 수 있습니다.

색인 예

다음의 색인 예는 효과적인 색인 작성에 도움을 줍니다.

예로서 세 개의 색인이 작성되었다고 가정하십시오.

색인 HEXIX가 정렬 순서로 *HEX를 사용하여 작성되었다고 가정합니다.

```
CREATE INDEX HEXIX ON STAFF (JOB)
```

색인 UNQIX가 고유 가중치 정렬 순서를 사용하여 작성되었다고 가정합니다.

```
CREATE INDEX UNQIX ON STAFF (JOB)
```

색인 SHRIX가 공유 가중치 정렬 순서를 사용하여 작성되었다고 가정합니다.

```
CREATE INDEX SHRIX ON STAFF (JOB)
```

- 정렬 순서 표를 사용하지 않고 동등 선택
- 고유 가중치 정렬 순서 표를 사용하여 동등 선택
- 공유 가중치 정렬 순서 표를 사용하여 동등 선택
- 고유 가중치 정렬 표를 사용하여 보다 큰 선택
- 고유 가중치 정렬 순서 표를 사용하여 결합 선택
- 공유 가중치 정렬 순서 표를 사용하여 결합 선택
- 정렬 순서 표를 사용하지 않고 순서화
- 고유 가중치 정렬 순서 표를 사용하여 순서화
- 공유 가중치 정렬 순서 표를 사용하여 순서화
- ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표를 사용하여 순서화
- 정렬 순서 표를 사용하지 않고 그룹화
- 고유 가중치 정렬 순서 표를 사용하여 그룹화
- 공유 가중치 정렬 순서 표를 사용하여 그룹화
- 고유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화
- ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화
- 공유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화
- ALWCPYDTA(*OPTIMIZE) 및 공유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화
- 고유 가중치 정렬 순서 표를 사용하여 다른 열에 대해 순서화 및 그룹화
- ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표를 사용하여 다른 열에 대해 순서화 및 그룹화
- ALWCPYDTA(*OPTIMIZE) 및 공유 가중치 정렬 순서 표를 사용하여 다른 열에 대해 순서화 및 그룹화

색인 예: 정렬 순서 표를 사용하지 않고 동등 선택

정렬 순서 표(SRTSEQ(*HEX))를 사용하지 않고 동등 선택

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   SRTSEQ(*HEX)
```

iSeries용 DB2 Universal Database는 색인 HEXIX 또는 색인 UNQIX를 사용할 수 있습니다.

색인 예: 고유 가중치 정렬 순서 표를 사용하여 동등 선택

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 동등 선택

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 HEXIX 또는 색인 UNQIX를 사용할 수 있습니다.

색인 예: 공유 가중치 정렬 순서 표를 사용하여 동등 선택

공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDSHR) LANGID(ENU))를 사용하여 동등 선택

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 SHRIX만을 사용할 수 있습니다.

색인 예: 고유 가중치 정렬 표를 사용하여 보다 큰 선택

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 보다 큰 선택

```
SELECT * FROM STAFF
 WHERE JOB > 'MGR'
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *GT ''MGR''')
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 UNQIX만을 사용할 수 있습니다.

색인 예: 고유 가중치 정렬 순서 표를 사용하여 결합 선택

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 결합 선택

```
SELECT * FROM STAFF S1, STAFF S2
 WHERE S1.JOB = S2.JOB
```

또는 JOIN 구문을 사용하여 같은 조회

```
SELECT *
 FROM STAFF S1 INNER JOIN STAFF S2
       ON S1.JOB = S2.JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE(STAFF STAFF)
  FORMAT(FORMAT1)
   JFLD((1/JOB 2/JOB *EQ))
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 각 조회에 색인 HEXIX 또는 색인 UNQIX를 사용할 수 있습니다.

색인 예: 공유 가중치 정렬 순서 표를 사용하여 결합 선택

공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDSHR) LANGID(ENU))를 사용하여 결합 선택

```
SELECT * FROM STAFF S1, STAFF S2
 WHERE S1.JOB = S2.JOB
```

또는 JOIN 구문을 사용하여 같은 조회

```
SELECT *
 FROM STAFF S1 INNER JOIN STAFF S2
       ON S1.JOB = S2.JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE(STAFF STAFF) FORMAT(FORMAT1)
   JFLD((1/JOB 2/JOB *EQ))
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 각 조회에 대해 색인 SHRIX만을 사용할 수 있습니다.

색인 예: 정렬 순서 표를 사용하지 않고 순서화

정렬 순서 표(SRTSEQ(*HEX))를 사용하지 않고 순서화

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
 ORDER BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   KEYFLD(JOB)
   SRTSEQ(*HEX)
```

iSeries용 DB2 Universal Database는 색인 HEXIX만을 사용할 수 있습니다.

색인 예: 고유 가중치 정렬 순서 표를 사용하여 순서화

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 순서화

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
 ORDER BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   KEYFLD(JOB) SRTSEQ(*LANGIDUNQ) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 UNQIX만을 사용할 수 있습니다.

색인 예: 공유 가중치 정렬 순서 표를 사용하여 순서화

공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDSHR) LANGID(ENU))를 사용하여 순서화

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
 ORDER BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   KEYFLD(JOB) SRTSEQ(*LANGIDSHR) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 SHRIX만을 사용할 수 있습니다.

색인 예: ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표를 사용하여 순서화

ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 순서화

```
SELECT * FROM STAFF
 WHERE JOB = 'MGR'
 ORDER BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF))
   QRYSLT('JOB *EQ ''MGR''')
   KEYFLD(JOB)
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
   ALWCPYDTA(*OPTIMIZE)
```

iSeries용 DB2 Universal Database는 선택에 대해 색인 HEXIX 또는 색인 UNQIX를 사용할 수 있습니다 정렬이 *LANGIDUNQ 정렬 순서 표를 사용하는 동안 순서화가 실행됩니다.

색인 예: 정렬 순서 표를 사용하지 않고 그룹화

정렬 순서 표(SRTSEQ(*HEX))를 사용하지 않고 그룹화

```
SELECT JOB FROM STAFF
 GROUP BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT2)
   GRPFLD((JOB))
   SRTSEQ(*HEX)
```

iSeries용 DB2 Universal Database는 색인 HEXIX 또는 색인 UNQIX를 사용할 수 있습니다.

색인 예: 고유 가중치 정렬 순서 표를 사용하여 그룹화

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 그룹화

```
  SELECT JOB FROM STAFF
   GROUP BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT2)
   GRPFLD((JOB))
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 HEXIX 또는 색인 UNQIX를 사용할 수 있습니다.

색인 예: 공유 가중치 정렬 순서 표를 사용하여 그룹화

공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDSHR) LANGID(ENU))를 사용하여 그룹화

```
  SELECT JOB FROM STAFF
   GROUP BY JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT2)
   GRPFLD((JOB))
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 SHRIX만을 사용할 수 있습니다.

다음 예는 3개의 추가 색인이 열 JOB 및 SALARY에 대해 작성된다고 가정합니다. 예 앞에 CREATE INDEX 문이 옵니다.

색인 HEXIX2가 정렬 순서로 *HEX를 사용하여 작성되었다고 가정합니다.

```
   CREATE INDEX HEXIX2 ON STAFF (JOB, SALARY)
```

색인 UNQIX2가 고유 가중치 정렬 순서를 사용하여 작성되었다고 가정합니다.

```
   CREATE INDEX UNQIX2 ON STAFF (JOB, SALARY)
```

색인 SHRIX2가 공유 가중치 정렬 순서를 사용하여 작성되었다고 가정합니다.

```
   CREATE INDEX SHRIX2 ON STAFF (JOB, SALARY)
```

색인 예: 고유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 동일한 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY JOB, SALARY
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(JOB SALARY)
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 UNQIX2를 사용하여 그룹화 및 순서화 요구사항 모두를 만족시킬 수 있습니다. 색인 UNQIX2가 없다면 iSeries용 DB2 Universal Database는 *LANGIDUNQ의 정렬 순서 표를 사용하여 색인을 작성합니다.

색인 예: ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화

ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 동일한 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY JOB, SALARY
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(JOB SALARY)
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
   ALWCPYDTA(*OPTIMIZE)
```

iSeries용 DB2 Universal Database는 UNQIX2를 사용하여 그룹화 및 순서화 요구사항 모두를 만족시킬 수 있습니다. 색인 UNQIX2가 없다면 iSeries용 DB2 Universal Database는

- *LANGIDUNQ의 정렬 순서 표를 사용하여 색인을 작성하거나
- 그룹화 및 순서화에 부합하도록 색인 HEXIX2를 사용합니다.

색인 예: 공유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화

공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDSHR) LANGID(ENU))를 사용하여 동일한 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY JOB, SALARY
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(JOB SALARY)
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 SHRIX2를 사용하여 그룹화 및 순서화 요구사항 모두를 만족시킬 수 있습니다. 색인 SHRIX2가 없다면 iSeries용 DB2 Universal Database는 *LANGIDSHR의 정렬 순서 표를 사용하여 색인을 작성합니다.

색인 예: ALWCPYDTA(*OPTIMIZE) 및 공유 가중치 정렬 순서 표를 사용하여 동일한 열에 대해 순서화 및 그룹화

ALWCPYDTA(*OPTIMIZE) 및 공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDSHR) LANGID(ENU)를 사용하여 동일한 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY JOB, SALARY
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(JOB SALARY)
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
   ALWCPYDTA(*OPTIMIZE)
```

iSeries용 DB2 Universal Database는 SHRIX2를 사용하여 그룹화 및 순서화 요구사항 모두를 만족시킬 수 있습니다. 색인 SHRIX2가 없다면 iSeries용 DB2 Universal Database는 *LANGIDSHR의 정렬 순서 표를 사용하여 색인을 작성합니다.

색인 예: 고유 가중치 정렬 순서 표를 사용하여 다른 열에 대해 순서화 및 그룹화

고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 다른 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY SALARY, JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(SALARY JOB)
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
```

iSeries용 DB2 Universal Database는 색인 HEXIX2 또는 색인 UNQIX2를 사용하여 그룹화 요구사항을 만족시킬 수 있습니다. 그룹화 결과가 들어 있는 임시 결과가 작성됩니다. 임시 색인이 빌드되어 순서화 요구사항을 만족시킵니다. *LANGIDUNQ 정렬 순서 표를 사용하여 임시 결과에 대해 작성됩니다.

색인 예: ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표를 사용하여 다른 열에 대해 순서화 및 그룹화

ALWCPYDTA(*OPTIMIZE) 및 고유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 다른 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY SALARY, JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(SALARY JOB)
   SRTSEQ(*LANGIDUNQ) LANGID(ENU)
   ALWCPYDTA(*OPTIMIZE)
```

iSeries용 DB2 Universal Database는 색인 HEXIX2 또는 색인 UNQIX2를 사용하여 그룹화 요구사항을 만족시킬 수 있습니다. 정렬이 순서화 요구사항에 부합되도록 실행됩니다.

색인 예: ALWCPYDTA(*OPTIMIZE) 및 공유 가중치 정렬 순서 표를 사용하여 다른 열에 대해 순서화 및 그룹화

ALWCPYDTA(*OPTIMIZE) 및 공유 가중치 정렬 순서 표(SRTSEQ(*LANGIDUNQ) LANGID(ENU))를 사용하여 다른 열에 대해 순서화 및 그룹화

```
SELECT JOB, SALARY FROM STAFF
 GROUP BY JOB, SALARY
 ORDER BY SALARY, JOB
```

OPNQRYF 명령을 사용할 때는 다음과 같이 지정하십시오.

```
OPNQRYF FILE((STAFF)) FORMAT(FORMAT3)
   GRPFLD(JOB SALARY)
   KEYFLD(SALARY JOB)
   SRTSEQ(*LANGIDSHR) LANGID(ENU)
   ALWCPYDTA(*OPTIMIZE)
```

iSeries용 DB2 Universal Database는 색인 HEXIX2를 사용하여 그룹화 요구사항을 만족시킬 수 있습니다. 정렬이 순서화 요구사항에 부합되도록 실행됩니다.

코드화 벡터 색인이란?

코드화 벡터 색인(EVI)은 의사결정 지원 및 조회 보고 환경에서 빠른 자료 액세스를 제공하기 위해 조회 Optimizer와 데이터베이스 엔진에서 사용되는 색인 오브젝트입니다. EVI는 기존 색인 오브젝트에 대한 보충 대체(2진 기수 트리 구조 - 논리 파일 또는 SQL 색인)이며, 비트맵 색인화의 변형입니다. EVI는 *크기*가 작고 비교적 단순하므로, 병렬로 처리될 수도 있는 표의 고속 *스캔*에 제공됩니다.

EVI는 두 개의 구성요소로 저장되는 자료 구조입니다.

- 기호 표는 표에 나타난 각 고유 키 값에 대한 통계 및 설명 정보를 포함합니다. 각 고유 키에는 1, 2 또는 4바이트 *크기*의 고유 코드가 할당됩니다.
- 벡터는 표에 있는 행과 같은 순서 위치에 나열된 코드 배열입니다. 벡터는 표에 있는 실제 행에 대한 포인터를 포함하고 있지 않습니다.

EVI의 장점

- 기억장치가 덜 요구됨
- 더 좋은 빌드 시간
- 더 정확한 통계를 조회 Optimizer에 제공

EVI의 단점

- 순서화 및 그룹화에 사용할 수 없음
- 결합에서 사용이 제한됨
- 일부 추가 유지보수 특성

EVI 작동 방법

Optimizer는 조회에 대한 비용 정보를 수집하기 위해 기호 표를 사용합니다. Optimizer가 조회를 처리하는 데 EVI를 사용하기로 결정하면, 데이터베이스 엔진은 이 벡터를 사용하여 표에 있는 각 행에 대해 한 비트를 포함하는 동적 비트맵을 빌드합니다. 이 행이 조회 선택을 만족시키면, 이 비트는 On으로 설정됩니다. 행이 조회 선택을 만족시키지 않은 경우, 비트는 Off로 설정됩니다. 중간 동적 비트맵은 비트맵 색인처럼 임시 조회를 만족시키기 위해 모두 AND 및 OR될 수 있습니다. 예를 들어, 사용자가 특정 기간 동안 특정 영역에 대한 판매 자료를 보려 한다면, 데이터베이스의 영역 열 및 4분기 열에 대해 EVI를 정의할 수 있습니다. 조회가 실행되면, 데이터베이스 엔진이 두 개의 EVI를 사용하여 동적 비트맵을 빌드한 후, 이 비트맵들을 모두 AND하여 두 선택 기준에 대한 관련 행만 들어 있는 비트맵을 생성합니다. 이 AND 기능은 서버가 읽고 테스트해야 하는 행의 수를 크게 줄여줍니다. 동적 비트맵은 조회가 실행되는 동안에만 존재합니다. 조회가 완료되면, 동적 비트맵은 제거됩니다.

EVI는 언제 사용되는가?

통계를 수집하고자 할 때, 전체 표 스캔이 선택되었을 때, 조회의 선택도가 20%-70%이고 동적 비트맵을 사용한 건너뛰 순차 액세스를 사용하여 스캔을 가속화할 때 또는 별 스키마 결합 조회에 별 스키마 결합이 사용되도록 계획되었을 때 코드화 벡터 색인이 고려되어야 합니다. 코드화 벡터 색인은 다음을 사용하여 작성되어야 합니다.

- 예상되는 고유 값의 수가 적은 단일 키 열
- 비교적 고정적인 키 열(자주 변하지 않음)
- WITH n DISTINCT VALUES절을 사용하여 예상되는 고유 값의 최대 수
- 별 스키마 모델용 외부 키 열에 대한 단일 키

EVI가 사용되는 경우 일반 색인이 EVI를 유지보수합니다. 이 유지보수 방법은 독창적입니다. 다음 표는 EVI가 유지보수되는 방법 및 EVI 유지보수 특성에 따라 EVI가 가장 효과적으로 되는 조건과 EVI가 가장 비효과적으로 되는 조건을 보여줍니다.

일반 색인 유지보수

색인이 작성되어 사용될 때마다, 유지보수 때문에 I/O 속도의 감소 가능성이 생기므로, 추가 색인의 작성 및 사용에 대한 유지보수 비용을 고려하는 것은 기본적인 일입니다. MAINT(*IMMED) 및 EVI를 사용하는 기준 색인의 경우에는 행을 삽입, 갱신 또는 삭제할 때 유지보수가 발생합니다.

색인의 유지보수를 줄이려면 다음을 고려하십시오.

- 지정 표에 대한 색인의 수를 최소화
- 일괄처리 삽입, 갱신 및 삭제 중 색인 드롭핑
- SMP를 사용하여 병렬로, 한번에 한개씩 색인을 작성
- 여러 CPU를 사용하여 복수 일괄처리 작업으로 여러 색인을 동시 작성
- SMP를 사용하여 병렬로 색인 유지보수

성능을 위해 색인을 작성하는 일의 목표는 유지보수할 색인의 수를 최소화시키면서, 통계 및 구현을 위한 색인의 최대 수를 균형 조절하는 것입니다.

EVI 유지보수

EVI를 사용할 때, 색인 유지보수에 대해 고유 요구가 존재합니다. 다음 표는 EVI가 유지보수되는 방법 및 EVI 유지보수 특성에 따라 EVI가 가장 효과적으로 되는 조건과 EVI가 가장 비효과적으로 되는 조건을 보여줍니다.

표 10. EVI 유지보수 고려사항

	조건	특성
가장 효과적	이미 있는 고유 키 값을 삽입할 때	• 최소 오버헤드 • 기호 표 키 값이 찾아지고 통계가 갱신됨 • 기존 바이트 코드와 함께, 새로운 행에 대해 벡터 요소가 추가됨
	새로운 고유 키 값을 삽입할 때 - 바이트 코드 범위 내에서 순서대로	• 최소 오버헤드 • 기호 표 키 값이 추가되고, 바이트 코드가 할당되며, 통계가 할당됨 • 새로운 바이트 코드와 함께, 새로운 행에 대해 벡터 요소가 추가됨
	새로운 고유 키 값을 삽입할 때 - 바이트 코드 범위 내에서 순서 없이	• 오버플로우 영역 임계값 내에 있는 경우 최소 오버헤드 • 기호 표 키 값이 오버플로우 영역에 추가되고, 바이트 코드가 할당되며, 통계가 할당됨 • 새로운 바이트 코드와 함께, 새로운 행에 대해 벡터 요소가 추가됨 • 오버플로우 영역 임계값에 도달할 경우, 상당한 오버헤드 • 액세스 경로가 유효화됨 - 사용할 수 없음 • EVI가 화면정리되고, 오버플로우 영역 키가 통합되고, 새로운 바이트 코드가 할당됨(기호 표와 벡터 요소가 갱신됨)
가장 비효과적	새로운 고유 키 값을 삽입할 때 - 바이트 코드 범위 밖에	• 상당한 오버헤드 • 액세스 경로화 비유효화됨 - 사용할 수 없음 • EVI가 화면정리되고, 다음 바이트 코드 크기가 사용되며, 새로운 바이트 코드가 할당됨(기호 표와 벡터 요소가 갱신됨)

EVI 사용에 대한 권장사항

코드화 벡터 색인은 의사결정 지원 및 조회 보고 환경에서 빠른 자료 액세스를 제공하기 위한 강력한 툴입니다. 그러나 EVI를 효과적으로 사용하려면, 다음 지침에 따라 EVI를 구현해야 합니다.

다음에 EVI 작성:

- 읽기 전용 표 또는 최소한의 INSERT, UPDATE, DELETE 활동을 갖는 표.
- WHERE절에서 사용되는 키 열 - SQL 요구의 로컬 선택 술부.
- 비교적 작은 고유 값 세트를 갖는 단일 키 열.

- 비교적 작은 고유 값 세트로 결과되는 복수 키 열.
- 정적 또는 비교적 정적인 고유 값 세트를 갖는 키 열.
- 여러 번 중복되는 비고유 키 열.

최대 바이트 코드 *크기*가 예상되는 EVI 작성:

- CREATE ENCODED VECTOR INDEX문에 ″WITH n DISTINCT VALUES″절을 사용하십시오.
- 불확실하다면, 65,535 보다 큰 수를 사용하여 4바이트 코드를 작성함으로써, 바이트 코드 *크기*를 전환하는 EVI 유지보수 오버헤드를 피하십시오.

자료를 로드할 때:

- EVI를 드롭(drop)하고, 자료를 로드하고, EVI를 작성하십시오.
- EVI 바이트 코드 *크기*는 표에 있는 실제 고유 키 값의 수를 기준으로 하여 자동으로 할당됩니다.
- 기호 표는 모든 키 값을 순서대로 포함하며, 오버플로우 영역에는 키를 포함하지 않습니다.

SMP 및 별렬 색인 작성 및 유지보수 고려:

SMP(Symmetrical Multiprocessing)는 색인을 병렬로 빌드하고 유지보수하기 위한 유용한 툴입니다. OS/400의 선택적 SMP 피처 사용의 결과는 빠른 색인 빌드 시간 및 색인을 병렬로 유지보수하면서 빠른 I/O 속도를 나타냅니다. *OPTIMIZE 또는 *MAX의 SMP 등급 값을 사용하면, 색인을 빌드하거나 유지보수하는 데 추가 복수 타스크와 추가 서버 자원이 사용됩니다. *MAX 등급 값의 경우, 색인 작성에 선형 확장성을 예상하십시오. 예를 들어, 4 프로세서 서버에서 색인을 작성할 경우, 1 프로세서 서버의 4배 만큼 빠르게 될 수 있습니다.

오버플로우 영역에 있는 값 검사:

필드 설명 표시(DSPFD) 명령(또는 V4R5M0 Operations Navigator - 데이터베이스)를 사용하여 오버플로우 영역에 몇 개의 값이 있는지 검사할 수 있습니다. DSPFD 명령이 실행되면, 오버플로우 영역에 있는 고유 키 값의 초기 및 실제 수의 세부사항에 대해 오버플로우 영역 매개변수를 검사하십시오.

CHGLF를 사용하여 색인 액세스 경로를 빌드:

액세스 경로 리빌드 강제 속성을 *YES*(FRCRBDAP(*YES))로 설정하고 논리 파일 변경(CHGLF) 명령을 사용하십시오. 이 명령은 색인을 드롭핑하고 재작성할 때와 같은 일을 수행하지만, 색인이 빌드되는 방법에 대해 알아야 할 필요가 없습니다. 이 명령은 원래 색인 정의를 사용할 수 없는 응용프로그램에 대해 또는 액세스 경로를 화면정리하는 경우 특히 효과적입니다.

제 6 장 데이터베이스 성능 향상을 위한 어플리케이션 설계 추가 정보

이 섹션에는 데이터베이스 성능을 최대화하기 위해 SQL 어플리케이션 설계시 적용할 수 있는 다음과 같은 설계 추가 정보가 들어 있습니다.

- 『데이터베이스 어플리케이션 설계 추가 정보: 라이브 자료 사용』
- 135 페이지의『데이터베이스 어플리케이션 설계 추가 정보: 열기 조작 수 감소』
- 137 페이지의『데이터베이스 어플리케이션 설계 추가 정보: 커서 위치 보유』

데이터베이스 어플리케이션 설계 추가 정보: 라이브 자료 사용

용어 **라이브 자료**는 데이터베이스 관리자가 자료를 사본을 만들지 않고 자료를 검색할 때 사용되는 액세스 유형을 의미합니다. 이 액세스 유형을 사용하면, 프로그램으로 리턴되는 자료는 항상 데이터베이스에 있는 자료의 현재 값을 반영합니다. 프로그래머는 데이터베이스 관리자가 자료 사본을 사용할지 또는 자료를 직접 검색할지를 제어할 수 있습니다. 이것은 사전컴파일러 명령 또는 SQL(STRSQL) 명령에 대해 자료 복사 허용(ALWCPYDTA) 매개변수를 지정하여 실행됩니다.

ALWCPYDTA(*NO)를 지정하면 데이터베이스 관리자는 계속 라이브 자료를 사용해야 합니다. 라이브 자료 액세스를 사용하면 커서를 다시 열고 닫을 필요없이 검색 중인 자료를 화면정리하므로 성능면에서 장점이 있습니다. 이러한 장점을 나타내는 어플리케이션의 예는 표시 화면에 리스트를 작성하는 어플리케이션입니다. 표시 화면에 한 번에 리스트의 20개 요소만을 표시할 수 있다면 초기 20개 요소가 표시된 후 어플리케이션 프로그래머는 다음 20행이 표시되도록 요구할 수 있습니다. OS/400 오퍼레이팅 시스템과 다른 오퍼레이팅 시스템을 위해 고안된 일반 SQL 어플리케이션은 다음과 같이 구조화될 수 있습니다.

```
  EXEC SQL
DECLARE C1 CURSOR FOR
  SELECT EMPNO, LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
    ORDER BY EMPNO
  END-EXEC.

  EXEC SQL
    OPEN C1
  END-EXEC.

*    PERFORM FETCH-C1-PARA  20 TIMES.

     MOVE EMPNO to LAST-EMPNO.

  EXEC SQL
    CLOSE C1
  END-EXEC.

*    Show the display and wait for the user to indicate that
*    the next 20 rows should be displayed.

  EXEC SQL
    DECLARE C2 CURSOR FOR
```

```
  SELECT EMPNO, LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
    WHERE EMPNO > :LAST-EMPNO
    ORDER BY EMPNO
  END-EXEC.

  EXEC SQL
    OPEN C2
  END-EXEC.

*    PERFORM FETCH-C21-PARA  20 TIMES.

*    Show the display with these 20 rows of data.

  EXEC SQL
    CLOSE C2
  END-EXEC.
```

위 예에서, 리스트를 계속하고 현재 자료를 얻으려면 추가 커서가 열려 있어야 함에 유의하십시오. 이로 인해, iSeries 서버에서 처리 시간을 증가시키는 추가 ODP가 작성됩니다. 위의 예 대신 프로그래머는 다음 SQL문을 사용하여 ALWCPYDTA(*NO)를 지정하여 어플리케이션을 설계할 수 있습니다.

```
  EXEC SQL
DECLARE C1 CURSOR FOR
  SELECT EMPNO, LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
    ORDER BY EMPNO
  END-EXEC.

  EXEC SQL
    OPEN C1
  END-EXEC.

*    Display the screen with these 20 rows of data.

*    PERFORM FETCH-C1-PARA  20 TIMES.

*    Show the display and wait for the user to indicate that
*    the next 20 rows should be displayed.

*    PERFORM FETCH-C1-PARA  20 TIMES.

  EXEC SQL
    CLOSE C1
  END-EXEC.
```

위 예에서, FOR 20 ROWS 절이 복수 열 FETCH문에 대해 사용된다면 조회를 더욱 잘 실행할 수 있습니다. 그러면 20개의 행을 하나의 조작으로 검색할 수 있습니다.

데이터베이스 어플리케이션 설계 추가 정보: 열기 조작 수 감소

SQL 자료 조작어 명령문은 자료에 대한 열린 자료 경로(ODP)를 작성하기 위해 데이터베이스 열기 연산을 실행하여야 합니다. 열린 자료 경로는 표에 대한 모든 입/출력 조작이 실행되는 경로입니다. 어떤 의미에서 열린 자료 경로는 SQL 어플리케이션을 표에 연결합니다. 프로그램에서 열기 조작 수는 성능에 상당한 영향을 미칠 수 있습니다. 데이터베이스 열기 조작은 다음에 대해 발생합니다.

- OPEN문
- SELECT INTO문
- VALUES절이 있는 INSERT문
- WHERE 조건이 있는 UPDATE문
- 연산자 또는 함수를 참조하는 WHERE CURRENT OF 커서 및 SET™ 절이 있는 UPDATE문
- 표현식이 있는 SET문
- 표현식이 있는 VALUES INTO문
- WHERE 조건이 있는 DELETE문

선택문이 있는 INSERT문은 두개의 열기 조작을 필요로 합니다. 양식 중에도 하나의 부속 선택에 대해 하나의 열기 조작이 필요한 부속 조회 양식도 있습니다.

열기 수를 최소화하기 위해 iSeries용 DB2 Universal Database는 열린 자료 경로(ODP)를 열어 두어 명령문이 다시 실행되면 ODP를 재사용합니다

- ODP는 호스트 변수를 사용하여 서브세트 임시 색인을 작성하였습니다. OS/400 데이터베이스 지원은 SQL 문에 지정된 행 선택과 대응하는 행에 대해서만 항목이 있는 임시 색인을 빌드하도록 선택할 수 있습니다. 호스트 변수가 행 선택에 사용된 경우, 임시 색인은 호스트 변수에 들어 있는 다른 값에 필요한 항목을 갖지 않습니다.
- 순서화가 호스트 변수 값에 대해 지정되었습니다.
- 대체 데이터베이스 파일(OVRDBF) 또는 삭제 대체(DLTOVR) CL 명령이 ODP가 열린 이후 발행되었으며, 이는 SQL문 실행에 영향을 줍니다.

  주: 참조 중인 표의 이름에 영향을 미치는 대체만이 제공된 프로그램 호출 내에서 ODP를 닫습니다.
- 결합은 임시 결합에 결합의 중간 단계가 포함되어야 하는 복합 결합입니다.
- 일부 경우는 복잡한 정렬을 포함하는데, 이때 임시 파일이 요구되며 다시 사용되지 않을 수 있습니다.
- 최종 열기 조작 이후 라이브러리 리스트가 변경되어 시스템 명명 모드의 규정화되지 않은 조회에 의해 선택된 표가 변경됩니다.
- 해시 결합을 사용하여 결합이 이행되었습니다.

| 내장된 정적 SQL의 경우, iSeries용 DB2 Universal Database는 동일한 명령문에 의해 열린 ODP를 재사용
| 합니다. 프로그램에서 나중에 코드화된 동일 명령문은 다른 명령문으로부터 ODP를 재사용하지 않습니다. 동
| 일 명령문이 프로그램에서 여러 번 실행되어야 하는 경우, 서브루틴에서 한 번 코드화하여 서브루틴을 호출하
| 여 명령문을 실행하십시오.

iSeries용 DB2 Universal Database에 의해 열린 ODP는 다음과 같은 경우에 닫힙니다.

- CLOSE, INSERT, UPDATE, DELETE 또는 SELECT INTO문이 완료되고 ODP는 임시 결과 또는 서브세트 임시 색인을 필요로 합니다.
- 자원 재생(RCLRSC) 명령이 발행됩니다. RCLRSC는 호출 스택에서 첫 번째 COBOL 프로그램이 종료되거나 COBOL 프로그램이 STOP RUN COBOL문을 발행할 때 발행됩니다. RCLRSC는 CLOSQLCSR(*ENDJOB)을 사용하여 사전컴파일러된 프로그램에 대해 작성된 ODP를 닫지 않습니다. 디폴트가 아닌 활성 그룹과의 RCLRSC 상호 작용에 대해서는 다음 서적을 참조하십시오.
  - *WebSphere Development Studio: ILE C/C++ Programmer's Guide*
  - *WebSphere Development Studio: ILE COBOL Programmer's Guide*
  - *WebSphere Development Studio: ILE RPG Programmer's Guide*
- 호출 스택에 SQL문이 포함된 최종 프로그램이 있는 경우, CLOSQLCSR(*ENDJOB)을 사용하여 사전컴파일된 프로그램 또는 CLOSQLCSR(*ENDACTGRP)을 사용하여 사전컴파일된 모듈에 대해 작성된 ODP를 제외한 모든 ODP가 닫힙니다.
- CONNECT(유형 1)문이 활성 그룹에 대해 어플리케이션 서버를 변경할 때 활성 그룹에 대해 작성된 모든 ODP가 닫힙니다.
- DISCONNECT문이 어플리케이션 서버에 대한 연결을 종료할 때 해당 어플리케이션 서버에 대한 모든 ODP가 닫힙니다.
- 해제된 연결이 COMMIT로 종료될 때 해당 어플리케이션 서버에 대한 모든 ODP가 닫힙니다.
- 조회 옵션 파일(QAQQINI) 매개변수 OPEN_CURSOR_THRESHOLD에 의해 지정된 열기 커서에 대한 임계값에 이르렀을 때

iSeries용 DB2 Universal Database가 다음과 같은 방법으로 ODP를 열어 둘지 여부를 제어할 수 있습니다.

- SQL문을 발행하는 프로그램이 항상 호출 스택이 있도록 어플리케이션 설계
- CLOSQLCSR(*ENDJOB) 또는 CLOSQLCSR(*ENDACTGRP) 매개변수를 사용
- 조회 옵션 파일(QAQQINI)의 OPEN_CURSOR_THRESHOLD 및 OPEN_CURSOR_CLOSE_COUNT 매개변수를 지정

iSeries용 DB2 Universal Database는 SET 절의 표현식에 오프레이터 또는 함수가 들어 있을 때 각각의 UPDATE WHERE CURRENT OF의 첫 번째 실행에 대해 열기 조작을 실행합니다. 호스트 언어 코드에서 함수 또는 연산을 코드화하여 열기 조작을 피할 수 있습니다.

예를 들면, 다음 UPDATE 절은 iSeries용 DB2 Universal Database가 열기 조작을 실행하도록 합니다.

```
  EXEC SQL
 FETCH EMPT  INTO  :SALARY
  END-EXEC.

  EXEC SQL
 UPDATE CORPDATA.EMPLOYEE
   SET SALARY = :SALARY + 1000   WHERE CURRENT OF EMPT
  END-EXEC.
```

반면에, 다음과 같은 코드화 기법을 사용하여 열기를 피할 수 있습니다.

```
  EXEC SQL
 FETCH EMPT  INTO  :SALARY
END EXEC.

ADD 1000 TO SALARY.

  EXEC SQL
 UPDATE CORPDATA.EMPLOYEE
   SET SALARY = :SALARY   WHERE CURRENT OF EMPT
  END-EXEC.
```

여러 가지 방법으로 SQL문에서 전체 열기를 수행할 지 여부를 판별할 수 있습니다. 기본 방법은 디버그가 사용 중일 때 발행된 메세지를 검토하거나 데이터베이스 모니터를 사용하는 것입니다. CL 명령 작업 추적(TRCJOB) 또는 저널 표시(DSPJRN)를 사용할 수도 있습니다.

데이터베이스 어플리케이션 설계 추가 정보: 커서 위치 보유

| 커서 위치를 보유하여 성능을 향상할 수 있습니다. 커서 위치는 비ILE 프로그램 호출이나 ILE 프로그램 호출
| 에 대해 보유할 수 있습니다. 또한, 여기에는 모든 프로그램 호출에 대한 커서 위치를 보유하기 위한 일반 규
| 칙이 있습니다.

데이터베이스 어플리케이션 설계 추가 정보: 비ILE 프로그램 호출에 대해 커서 위치 보유

비ILE 프로그램 호출의 경우, SQL 커서 닫기(CLOSQLCSR) 매개변수를 사용하여 다음 범위를 지정할 수 있습니다.

- 커서
- 준비된 명령문
- 잠금

적절히 사용될 때 CLOSQLCSR 매개변수는 필요한 SQL OPEN, PREPARE 및 LOCK문 수를 줄일 수 있습니다. 또한 프로그램 호출시 커서 위치를 보유할 수 있어 어플리케이션을 단순화할 수 있습니다.

***ENDPGM**

이것은 모든 비ILE 사전컴파일러에 대한 디폴트입니다. 이 옵션을 사용하면 열린 프로그램이 호출 스택에 있는 동안에만 커서를 열어 두어 액세스할 수 있습니다. 프로그램이 종료되면 SQL 커서는 더 이상 사용될 수 없습니다. 준비된 명령문 또한 프로그램이 종료될 때 유실됩니다. 그러나 잠금은 호출 스택에 있는 최종 SQL 프로그램이 완료될 때까지 남아 있습니다.

***ENDSQL**

이 옵션의 경우, 프로그램에 의해 작성된 SQL 커서 및 준비된 프로그램이 호출 스택에 있는 최종 SQL 프로그램이 완료될 때까지 열려 있습니다. 다른 프로그램에 의해 사용될 수 없으며, 동일한 프로그램에 대한 다른 호출에 의해서만 사용될 수 있습니다. 잠금은 호출 스택에 있는 최종 SQL 프로그램이 완료될 때까지 남아 있습니다.

***ENDJOB**

이 옵션의 경우 작업 기간 동안 SQL 커서, 준비된 명령문 및 잠금 상태로 유지할 수 있습니다. 스택에 있는 최종 SQL 프로그램이 완료될 때, *ENDJOB에 의해 작성된 모든 SQL 자원이 계속 활동합니다. 잠금이 활동 중입니다. CLOSE, COMMIT 및 ROLLBACK문에 의해 명백히 닫히지 않은 SQL 커서가 계속 열려 있습니다. 준비된 명령문을 동일한 프로그램에 대한 후속 호출에 대해 계속 사용할 수 있습니다.

데이터베이스 어플리케이션 설계 추가 정보: ILE 프로그램간 호출시 커서 위치 보유

ILE 프로그램 호출의 경우, SQL 커서 닫기(CLOSQLCSR) 매개변수를 사용하여 다음 범위를 지정할 수 있습니다.

- 커서
- 준비된 명령문
- 잠금

적절히 사용될 때 CLOSQLCSR 매개변수는 필요한 SQL OPEN, PREPARE 및 LOCK문 수를 줄일 수 있습니다. 또한 프로그램 호출시 커서 위치를 보유할 수 있어 어플리케이션을 단순화할 수 있습니다.

***ENDACTGRP**

이것은 ILE 사전컴파일러에 대한 디폴트입니다. 이 옵션의 경우, SQL 커서 및 준비된 명령문은 프로그램이 실행 중인 활동 그룹이 종료될 때까지 계속 열려있습니다. 다른 프로그램에 의해 사용될 수 없으며, 동일한 프로그램에 대한 다른 호출에 의해서만 사용될 수 있습니다. 잠금은 활성 그룹이 종료될 때까지 남아 있습니다.

***ENDMOD**

이 옵션의 경우, 열린 모듈이 활동 중인 경우에만 커서가 열려 있어 액세스할 수 있습니다. 모듈이 종료되면 SQL 커서는 더 이상 사용될 수 없습니다. 준비된 명령문 또한 모듈이 종료될 때 유실됩니다. 그러나 잠금은 호출 스택에 있는 최종 SQL 프로그램이 완료될 때까지 남아 있습니다.

데이터베이스 어플리케이션 설계 추가 정보: 모든 프로그램 호출에 대한 커서 위치 보유 일반 규칙

CLOSQLCSR(*ENDPGM) 또는 CLOSQLCSR(*ENDMOD)을 사용하여 컴파일된 프로그램을 사용하는 경우 자료에 액세스하기 위해 프로그램 또는 모듈을 호출할 때마다 커서가 열려야 합니다. SQL 프로그램 또는 모듈이 여러번 호출되므로 재사용 가능한 ODP를 활용하려는 경우, 커서는 프로그램 또는 모듈이 존재하기 전에 명백히 닫혀야 합니다.

CLOSQLCSR 매개변수를 사용하고 *ENDSQL, *ENDJOB 또는 *ENDACTGRP를 지정하는 경우 호출시마다 OPEN 및 CLOSE문을 실행할 필요가 없습니다. 명령문을 더 적게 실행할 수 있을 뿐아니라, 프로그램 또는 모듈 호출간에 커서 위치를 유지할 수 있습니다.

다음 SQL문 예는 CLOSQLCSR 매개변수를 사용하는 장점을 보여 줍니다.

```
EXEC SQL
   DECLARE DEPTDATA CURSOR FOR
     SELECT EMPNO, LASTNAME
      FROM CORPDATA.EMPLOYEE
       WHERE WORKDEPT = :DEPTNUM
END-EXEC.

EXEC SQL
   OPEN DEPTDATA
END-EXEC.

EXEC SQL
   FETCH DEPTDATA INTO :EMPNUM, :LNAME
END-EXEC.

EXEC SQL
   CLOSE DEPTDATA
END-EXEC.
```

이 프로그램이 다른 SQL 프로그램에서 여러번 호출되는 경우 재사용 가능한 ODP를 사용할 수 있습니다. 따라서 SQL이 이 프로그램에 대한 호출 간에 활동 중인 한 OPEN문은 데이터베이스 열기 조작을 요구하지 않습니다. 그러나, 커서가 각각의 OPEN문 다음에 있는 첫 번째 결과 행에 계속 위치지정되면 FETCH문이 항상 첫 번째 행을 리턴합니다.

다음 예에서, CLOSE문이 다음과 같이 제거되었습니다.

```
EXEC SQL
   DECLARE DEPTDATA CURSOR FOR
     SELECT EMPNO, LASTNAME
      FROM CORPDATA.EMPLOYEE
       WHERE WORKDEPT = :DEPTNUM
END-EXEC.

    IF CURSOR-CLOSED IS = TRUE THEN
EXEC SQL
   OPEN DEPTDATA
END-EXEC.

EXEC SQL
   FETCH DEPTDATA INTO :EMPNUM, :LNAME
END-EXEC.
```

이 프로그램이 *ENDJOB 옵션 또는 *ENDACTGRP 옵션으로 사전컴파일되고 활성 그룹이 활동 중인 경우 커서 위치가 유지됩니다. 커서 위치는 다음과 같은 경우에도 유지됩니다.

- 프로그램이 *ENDSQL 옵션으로 사전컴파일됨
- SQL이 프로그램 호출 간에 활동 중임

이러한 전략으로 인해 프로그램에 대한 각각의 호출이 커서에 있는 다음 행을 검색합니다. 연속 자료 요구의 경우, OPEN문이 필요없으며 실제로 -502 SQLCODE에 대해 실패됩니다. 오류를 무시하거나 코드를 추가하여 OPEN을 건너뛸 수 있습니다. 우선 FETCH문을 사용하여 실행할 수 있으며 FETCH 조작이 실패한 경우에만 OPEN문을 실행하여 실행합니다.

또한 이 기법은 준비된 명령문에도 적용됩니다. 프로그램은 먼저 EXECUTE를 시도한 후 실패한 경우, PREPARE를 실행합니다. 결과는 올바른 CLOSQLCSR 옵션이 선택되었다는 가정에서 PREPARE가 프로그램에 대한 첫 번째 호출에 대해서만 필요합니다. 물론, 명령문이 프로그램에 대한 호출간에 변경할 수 있으면, 모든 경우에 PREPARE를 실행해야 합니다.

기본 프로그램은 첫 번째 호출에서만 특수 매개변수를 송신하여 이를 제어할 수도 있습니다. 이 특수 매개변수 값은 첫 번째 호출임을, 서브프로그램은 OPEN, PREPARE 및 LOCK을 실행하여야 합니다.

주: COBOL 프로그램을 사용할 경우, STOP RUN문을 사용하지 마십시오. 호출 스택에서 첫 번째 COBOL 프로그램이 종료되거나 STOP RUN문을 실행할 때 자원 재생(RCLRSC) 조작이 실행됩니다. 이 조작은 SQL 커서를 닫습니다. *ENDSQL 옵션은 요구대로 작동되지 않습니다.

제 7 장 데이터베이스 성능 향상을 위한 프로그래밍 기법

다음 코딩 추가 정보는 SQL 조회의 성능을 향상시키는 데 도움이 됩니다.

- 『데이터베이스 성능 향상을 위한 프로그래밍 기법: OPTIMIZE 절 사용』
- 142 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: FETCH FOR n ROWS 사용』
- 143 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: INSERT FOR n ROWS 사용』
- 144 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: 데이터베이스 관리자 블록화 제어』
- 145 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: SELECT문을 사용하여 선택되는 열 수 최적화』
- 145 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: SQL PREPARE 명령문을 사용하여 중복 유효성 검사 제거』
- 146 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: REFRESH(*FORWARD)를 사용하여 대화식으로 표시되는 자료 페이징』

데이터베이스 성능 향상을 위한 프로그래밍 기법: OPTIMIZE 절 사용

어플리케이션이 커서에 대한 전체 결과표를 검색하지 않을 경우, OPTIMIZE 절을 사용하면 성능을 향상시킬 수 있습니다. 조회 Optimizer는 Optimize 절에 지정된 값을 사용하여 행의 서브세트를 검색하기 위한 예상 비용을 수정합니다.

다음 조회가 1000개의 행을 리턴한다고 가정하십시오.

```
  EXEC SQL
DECLARE C1 CURSOR FOR
  SELECT EMPNO, LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
      WHERE WORKDEPT = 'A00'
     ORDER BY LASTNAME
    OPTIMIZE FOR 100 ROWS
END EXEC.
```

주: 위의 OPTIMIZE절에 사용될 수 있는 값은 1-9999999 또는 ALL입니다.

Optimizer는 다음 비용을 연산합니다.

최적화율 = n행에 대한 최적화 값 / 응답 세트의 예상 행 수

```
Cost using a temporarily created index:

               Cost to retrieve answer set rows
            +  Cost to create the index
            +  Cost to retrieve the rows again
                  with a temporary index        * optimize ratio

Cost using a SORT:

               Cost to retrieve answer set rows
```

```
          +  Cost for SORT input processing
          +  Cost for SORT output processing  * optimize ratio


Cost using an existing index:

             Cost to retrieve answer set rows
             using an existing index          * optimize ratio
```

앞의 예에서 색인을 정렬하거나 작성하는 데 드는 예상 비용은 최적화율에 의해 조정되지 않습니다. 따라서 Optimizer는 최적화와 사전처리 요구사항의 균형을 맞출 수 있습니다. 최적화 수가 결과표의 행 수보다 큰 경우에는 예상 비용에 어떤 조정도 일어나지 않습니다. 조회에 OPTIMIZE 절이 지정되지 않은 경우에는, 명령문 유형, ALWCPYDTA 값 또는 출력 장치에 따라 디폴트 값이 사용됩니다.

명령문 유형	ALWCPYDTA(*OPTIMIZE)	ALWCPYDTA(*YES 또는 *NO)
DECLARE CURSOR	결과표의 행 수	3% 또는 결과표의 행 수
삽입된 선택	2	2
INTERACTIVE 선택을 표시 화면으로 출력	3% 또는 결과표의 행 수	3% 또는 결과표의 행 수
INTERACTIVE 선택을 프린터 또는 데이터베이스 표로 출력	결과표의 행 수	결과표의 행 수

OPTIMIZE 절은 조회 최적화에 다음과 같은 영향을 미칩니다.

- 기존 색인을 사용하도록 합니다(작은 수를 지정하여).
- 응답 세트에서 가능한 행 수를 큰 수로 지정하여 색인 작성을 가능하게 하거나, 정렬 또는 해시를 실행하도록 합니다.

데이터베이스 성능 향상을 위한 프로그래밍 기법: FETCH FOR n ROWS 사용

많은 FETCH문을 연속해서 실행하는 어플리케이션의 성능은 FETCH FOR n ROWS를 사용하여 향상시킬 수 있습니다. 이 절을 사용하면, 표에 있는 자료의 여러 행을 검색하고 검색한 행을 호스트 구조 배열 또는 행 기억장치 영역에 단일 FETCH와 함께 넣을 수 있습니다. 호스트 구조 배열 또는 행 기억장치 영역의 선언에 대한 자세한 정보는 SQL 참조서 또는 호스트 언어를 사용한 SQL 프로그래밍 서적의 각 장을 프로그래밍하십시오.

FOR n ROWS 절 없이 FETCH문을 사용하는 SQL 어플리케이션의 성능은 복수행 FETCH문을 사용하여 여러 행을 검색함으로써 향상시킬 수 있습니다. 호스트 구조 배열 또는 행 기억장치 영역이 FETCH로 채워진 다음에는, 어플리케이션이 해당 배열 또는 기억장치 영역의 자료를 루프하여 개별 행 각각을 처리할 수 있습니다. 이 명령문은 SQL 실행이 한 번만 호출되고 모든 자료가 어플리케이션 프로그램으로 동시에 리턴되었기 때문에 더욱 빠르게 실행됩니다.

사용자가 어플리케이션 프로그램을 변경하여 SQL 실행시 표에서 검색되는 행을 데이터베이스 관리자가 블록화하도록 할 수 있습니다. 자세한 정보는 144 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: 데이터베이스 관리자 블록화 제어』를 참조하십시오.

FETCH FOR n ROWS 사용시 SQL 블록화 성능 향상을 위해 몇 가지 기술을 사용할 수 있습니다.

다음 표에서는 프로그램이 100개의 행을 어플리케이션에 FETCH하려고 시도 했습니다. 표에서 블록화를 실행할 수 있을 때의 SQL 실행 및 데이터베이스 관리자에 대한 호출 횟수의 차이를 주의하여 보십시오.

표 11. FETCH문을 사용한 호출 횟수

	블록화를 사용하지 않는 데이터베이스 관리자	블록화를 사용하는 데이터베이스 관리자
단일 행 FETCH문	100회의 SQL 호출로 100회의 데이터베이스 호출	100회의 SQL 호출로 1회의 데이터베이스 호출
복수 행 FETCH문	1회의 SQL 실행 호출로 100회의 데이터베이스 호출	1회의 SQL 실행 호출로 1회의 데이터베이스 호출

데이터베이스 성능 향상을 위한 프로그래밍 기법: FETCH FOR n ROWS 사용시 SQL 블록화 성능 향상

FETCH FOR n ROWS를 사용할 때는 다음 사항에 대해 특별히 성능 고려를 해야 합니다. 만족되어야 합니다. 다음을 사용하여 SQL 블록화 성능을 향상시킬 수 있습니다.

- 호스트 구조 배열의 속성 정보, 또는 행 기억장치 영역과 연관된 설명자의 속성 정보는 검색되는 열의 속성과 일치해야 합니다.
- 어플리케이션은 한 번의 복수 행 FETCH 호출로 가능한 한 많은 행을 검색해야 합니다. 복수 행 FETCH 요구에 대한 블록화 요소는 시스템 페이지 크기 또는 OVRDBF 명령의 SEQONLY 매개변수에 의해 제어되지 않으며, 복수 행 FETCH 요구에서 요구되는 행 수에 의해 제어됩니다.
- 한 프로그램 내에서 동일한 커서에 대한 단일/복수 행 FETCH 요구가 혼용되어서는 안됩니다. 한 커서에 대한 한 번의 FETCH가 복수 행 FETCH로 취급된 경우에는 해당 커서에 대한 모든 FETCH가 복수 행 FETCH로 취급됩니다. 이 경우, 단일 행 FETCH 요구는 각각 한 행의 복수 행 FETCH로 취급됩니다.
- PRIOR, CURRENT 및 RELATIVE 화면이동 옵션을 복수 행 FETCH문과 함께 사용해서는 안됩니다. 어플리케이션이 커서를 임의로 이동할 수 있도록 하려면, 데이터베이스 관리자가 어플리케이션과 동일한 커서 위치를 유지해야 합니다. 따라서, SQL 실행시 이러한 옵션이 지정된 상태에서의 화면이동 커서에 대한 모든 FETCH 요구는 복수 행 FETCH 요구로 취급됩니다.

데이터베이스 성능 향상을 위한 프로그래밍 기법: INSERT FOR n ROWS 사용

많은 INSERT문을 연속해서 실행하는 어플리케이션의 성능은 INSERT FOR n ROWS를 사용하여 향상시킬 수 있습니다. 이 절을 사용하면 호스트 구조에 있는 자료의 행을 목표 표에 하나 이상 삽입할 수 있습니다. 이 배열은 구조의 요소가 목표 표의 열에 해당되는 구조 배열이어야 합니다.

INSERT...VALUES문(n ROWS 없이)을 루프하는 SQL 어플리케이션의 성능은 INSERT n ROWS문을 표에 복수 행을 삽입하여 향상시킬 수 있습니다. 어플리케이션이 행이 있는 호스트 배열을 채우기 위해 루프한 다음에는 단일 INSERT n ROWS문을 실행하여 전체 배열을 표에 삽입할 수 있습니다. 이 명령문은 SQL 실행이 한 번만 호출되고 모든 자료가 목표 표에 동시에 삽입되었기 때문에 보다 빠르게 실행됩니다.

다음 표에서는 프로그램이 100개의 행을 표에 INSERT하려고 시도했습니다. 블록화를 실행할 수 있을 때의 SQL 실행 및 데이터베이스 관리자에 대한 호출 횟수의 차이를 주의하여 보십시오.

표 12. INSERT문 사용시 호출 횟수

	블록화를 사용하지 않는 데이터베이스 관리자	블록화를 사용하는 데이터베이스 관리자
단일 행 INSERT문	100회의 SQL 실행 호출로 100회의 데이터베이스 호출	100회의 SQL 실행 호출로 1회의 데이터베이스 호출
복수 행 INSERT문	1회의 SQL 실행 호출로 100회의 데이터베이스 호출	1회의 SQL 실행 호출로 1회의 데이터베이스 호출

데이터베이스 성능 향상을 위한 프로그래밍 기법: 데이터베이스 관리자 블록화 제어

성능을 향상시키기 위해, SQL 실행시 가능한 한 언제든지 데이터베이스 관리자 블록화를 통해 행을 검색하고 삽입하려는 시도가 이루어집니다.

원하는 경우에는 사용자가 블록화를 제어할 수 있습니다. SQL문이 포함된 어플리케이션 프로그램을 호출하기 전에 CL 명령인 OVRDBF(데이터베이스 파일 대체)에 SEQONLY 매개변수를 사용하십시오. CRTSQLxxx 명령에 ALWBLK 매개변수를 지정할 수도 있습니다.

다음과 같은 경우에는 데이터베이스 관리자가 블록화를 허용하지 않습니다.

- 커서를 갱신 또는 삭제할 수 있는 경우
- 행과 피드백 정보를 합한 길이가 32767을 초과하는 경우. 피드백 정보의 최소 *크기*는 11바이트입니다. 피드백 *크기*는 커서가 사용한 색인에 있는 키 열의 바이트 수와 널(null) 허용 키 열(있는 경우)의 수에 따라 증가됩니다.
- COMMIT(*CS)가 지정되고 ALWBLK(*ALLREAD)가 지정되지 않은 경우
- COMMIT(*ALL)가 지정되고 다음을 만족하는 경우
  - SELECT INTO문 또는 블록화된 FETCH문이 사용되지 않음
  - 조회가 열 함수를 사용하지 않거나 열 기준 그룹화를 지정하지 않음
  - 임시 결과 표를 작성할 필요가 없음
- COMMIT(*CHG)가 지정되고 ALWBLK(*ALLREAD)가 지정되지 않은 경우
- 커서에는 최소한 한개의 부속 조회 및 가장 바깥쪽 부속 조회가 제공한 부속 조회에 대한 상관 참조 또는 가장 바깥쪽 부속 조회가 처리한 IN, = ANY 또는 < > ALL 부속 조회 술부 연산자가 있는 부속 조회가 들어 있는데, 이는 상관 참조로 처리되며 이 부속 조회는 분리할 수 없습니다.

SQL 실행시 다음과 같은 경우에는 데이터베이스 관리자와 함께 행이 자동으로 블록화됩니다.

- INSERT

  INSERT문에 선택문이 포함된 경우에는 삽입된 행이 블록화되고 해당 블록이 가득 찰 때까지 목표 표에 실제로 삽입되지는 않습니다. SQL 실행시 블록화된 삽입에 대해 블록화가 자동으로 일어납니다.

주: VALUES 절이 있는 INSERT가 지정된 경우, SQL을 실행하면 삽입을 실행하는 데 사용되는 내부 커서는 프로그램이 종료될 때까지 실제로 닫히지 않을 수 있습니다. 동일한 INSERT 명령문이 다시 실행되는 경우에는 완전 열기가 필요하지 않으며 어플리케이션은 더 빠르게 실행됩니다.

- OPEN

  다음 조건이 모두 만족되는 경우에는 행 검색시 OPEN문 하에서 블록화가 이루어집니다.

  – 커서가 FETCH문에만 사용됩니다.

  – 프로그램에 No EXECUTE 또는 EXECUTE IMMEDIATE문이 있거나, ALWBLK(*ALLREAD)가 지정되었거나, 커서가 FOR FETCH ONLY 절과 함께 선언됩니다.

  – COMMIT(*CHG) 및 ALWBLK(*ALLREAD)가 지정되고 COMMIT(*CS) 및 ALWBLK(*ALLREAD)가 지정되었거나, COMMIT(*NONE)가 지정되었습니다.

데이터베이스 성능 향상을 위한 프로그래밍 기법: SELECT문을 사용하여 선택되는 열 수 최적화

SELECT문의 선택 리스트에서 지정하는 열의 수에 따라 데이터베이스 관리자가 기준 표에서 자료를 검색하고 해당 자료를 어플리케이션 프로그램의 호스트 변수로 맵핑합니다. 지정하는 열의 수를 최소로 하면 처리 장치의 자원 사용이 보존됩니다. SELECT *을 코드화하는 것이 편리하기는 하지만, 어플리케이션에 실제로 필요한 열을 명시적으로 코드화하는 것이 훨씬 좋습니다. 이는 특히 색인 전용 액세스만 원하거나, 모든 열이 정렬 조작에 관여하는 경우(SELECT DISTINCT 및 SELECT UNION에 대해 발생 하는것처럼)에 중요한 사항입니다.

이는 조회 내에 있는 열의 수를 최소화하고 따라서 색인이 모든 자료에 대한 요구를 완전히 충족시키기 위해 사용될 수 있다는 가능성을 높이므로, 색인만 액세스를 고려할 때 중요한 사항이 되기도 합니다.

데이터베이스 성능 향상을 위한 프로그래밍 기법: SQL PREPARE 명령문을 사용하여 중복 유효성 검사 제거

SQL PREPARE문이 실행될 때 발생하는 처리는 사전컴파일 처리 도중 발생하는 처리와 유사합니다. 준비되는 명령문에 대해서는 다음 처리가 발생합니다.

- 구문이 검사됩니다.
- 오브젝트 사용이 유효한지 여부를 확인하기 위해 명령문의 유효성을 검사합니다.
- 액세스 계획이 빌드됩니다.

명령문이 다시 실행되거나 열리면, 데이터베이스 관리자는 액세스 계획이 여전히 유효한지 여부를 다시 검사합니다. 이러한 열기 처리 유효성 검사의 대부분은 PREPARE 처리 도중 발생한 유효성 검사와 중복됩니다. DLYPRP(*YES) 매개변수는 해당 프로그램의 PREPARE문이 동적 명령문의 유효성을 완전히 검사할지 여부를 지정합니다. 동적 명령문이 열리거나 실행되면 유효성 검사가 완료됩니다. 이 매개변수는 중복 유효성 검사를 제거하므로 PREPARE SQL문을 사용하는 프로그램의 성능을 크게 향상시킵니다. 이 사전컴파일 옵션을 지정하는 프로그램은 명령문이 유효한지 확인하기 위해 OPEN 또는 EXECUTE 명령문을 실행한 다음

SQLCODE 및 SQLSTATE를 검사해야 합니다. 명령문에 대해 OPEN이 발행되기 전에 PREPARE문에 INTO 절이 사용되거나 DESCRIBE문이 동적 명령문을 사용하는 경우에는 DLYPRP(*YES)로 성능을 향상시킬 수 없습니다.

데이터베이스 성능 향상을 위한 프로그래밍 기법: REFRESH(*FORWARD)를 사용하여 대화식으로 표시되는 자료 페이징

대형 표에서, 표에서 직접 가장 최근의 자료를 동적으로 검색하는 STRSQL 명령의 REFRESH(*ALWAYS) 매개변수로 인해 대개 페이징 성능이 저하됩니다. REFRESH(*FORWARD)를 지정하면 페이징 성능을 향상시킬 수 있습니다.

REFRESH(*FORWARD)를 사용하여 자료를 대화식으로 표시할 경우, 표시 화면을 통해 정방향으로 페이징할 때 임시 선택문의 결과가 임시 표로 복사됩니다. 해당 표를 공유하는 다른 사용자는 사용자가 선택문 결과를 표시하는 동안 행에 대한 변경을 할 수 있습니다. 이미 표시된 행에 대해 정방향 또는 역방향으로 페이징하면 갱신된 표의 행이 아닌 임시 표의 행이 표시됩니다.

화면정리 옵션은 세션 서비스 표시 화면에서 변경할 수 있습니다.

제 8 장 일반 iSeries용 DB2 UDB 성능 고려사항

어플리케이션을 코드화할 때, 다음과 같은 일반 추가 정보가 성능을 최적화하는 데 도움이 됩니다.

- 『긴 오브젝트명을 사용할 때 데이터베이스 성능에 미치는 영향』
- 『데이터베이스 성능에 미치는 사전컴파일 옵션의 영향』
- 148 페이지의 『데이터베이스 성능에 영향을 미치는 ALWCPYDTA 매개변수』
- 149 페이지의 『데이터베이스에서 VARCHAR 및 VARGRAPHIC 자료 사용에 대한 추가 정보』

긴 오브젝트명을 사용할 때 데이터베이스 성능에 미치는 영향

긴 오브젝트명은 SQL문에 사용될 때 시스템 오브젝트명으로 내부적으로 변환됩니다. 이러한 변환으로 성능에 약간의 영향을 줄 수 있습니다.

라이브러리명을 갖는 긴 오브젝트명을 규정화하면 짧은 이름으로의 변환이 사전컴파일시에 발생합니다. 이러한 경우, 명령문이 실행될 때 성능에 영향을 주지 않습니다. 그렇지 않은 경우, 변환은 실행시에 실행되며 성능에 약간 영향을 줍니다.

데이터베이스 성능에 미치는 사전컴파일 옵션의 영향

여러 사전컴파일 옵션이 성능이 향상된 SQL 프로그램을 작성하는 데 사용될 수 있습니다. 여러 사전컴파일 옵션은 옵션을 사용하면 어플리케이션의 기능에 영향을 주므로 유일한 옵션입니다. 이러한 이유로, 이러한 매개변수에 대한 디폴트 값은 이전 릴리스로부터 어플리케이션을 마이그레이션하는 값입니다. 그러나, 다른 옵션을 지정하여 성능을 향상시킬 수 있습니다. 다음 표는 이러한 사전컴파일 옵션 및 그로 인한 성능 향상을 보여줍니다.

이러한 옵션 중의 일부는 대부분의 어플리케이션에 적합합니다. CRTDUPOBJ 명령을 사용하여 SQL CRTSQLxxx 명령 사본을 작성하고 CHGCMDDFT 명령을 사용하여 사전컴파일 매개변수에 대한 최적 값을 사용자 정의하십시오. DSPPGM, DSPSRVPGM, DSPMOD 또는 PRTSQLINF 명령을 사용하여 기존 프로그램 오브젝트에 대해 사용되는 사전컴파일 옵션을 표시할 수 있습니다.

사전컴파일 옵션	최적 값	향상 내용	고려사항	관련 주제
ALWCPYDTA	*OPTIMIZE(디폴트)	순서화 또는 그룹화 범주가 선택 범주와 상충하는 조회	조회가 열려 있을 때 자료가 복사될 수 있습니다.	148 페이지의 『데이터베이스 성능에 영향을 미치는 ALWCPYDTA 매개변수』를 참조하십시오.
ALWBLK	*ALLREAD(디폴트)	추가 읽기 전용 커서가 블록화를 사용합니다.	ROLLBACK HOLD는 읽기 전용 커서의 위치를 변경할 수 없습니다. 위치 지정된 갱신 또는 삭제의 동적 처리가 실패할 수 있습니다.	144 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: 데이터베이스 관리자 블록화 제어』를 참조하십시오.

사전컴파일 옵션	최적 값	향상 내용	고려사항	관련 주제
CLOSQLCSR	*ENDJOB, *ENDSQL 또는 *ENDACTGRP	커서 위치가 프로그램 호출시에 보유될 수 있습니다.	프로그램 호출이 종료될 때 SQL 커서의 내재 닫기가 실행되지 않습니다.	137 페이지의 『데이터베이스 어플리케이션 설계 추가 정보: 비ILE 프로그램 호출에 대해 커서 위치 보유』를 참조하십시오.
DLYPRP	*YES	SQL PREPARE문을 사용하는 프로그램이 더욱 신속히 실행될 수 있습니다.	준비된 명령문의 유효성 검사 완료는 명령문이 실행 또는 열릴 때까지 지연됩니다.	145 페이지의 『데이터베이스 성능 향상을 위한 프로그래밍 기법: SQL PREPARE 명령문을 사용하여 중복 유효성 검사 제거』를 참조하십시오.
TGTRLS	*CURRENT(디폴트)	사전컴파일은 현재 릴리스에서 사용할 수 있는 성능 향상을 활용하는 코드를 생성합니다.	프로그램 오브젝트는 이전 릴리스의 서버에 대해 사용될 수 없습니다.	

데이터베이스 성능에 영향을 미치는 ALWCPYDTA 매개변수

일부 복합 조회는 색인을 사용 또는 작성하는 대신 조회를 평가하기 위해 정렬 또는 해시 방식을 사용하여 보다 잘 실행될 수 있습니다. 정렬 또는 해시를 사용하여 데이터베이스 관리자는 행 선택을 순서화 및 그룹화 프로세스로부터 분리할 수 있습니다. 비트맵 처리는 이 매개변수를 통해 부분적으로 제어될 수도 있습니다. 이렇게 분리하여 선택에 대해 가장 효율적인 색인을 사용할 수 있습니다. 예를 들면, 다음과 같은 SQL문을 고려하십시오.

```
  EXEC SQL
DECLARE C1 CURSOR FOR
  SELECT EMPNO, LASTNAME, WORKDEPT
        FROM CORPDATA.EMPLOYEE
     WHERE WORKDEPT = 'A00'
    ORDER BY LASTNAME
  END-EXEC.
```

위의 SQL문은 OPNQRYF 명령을 사용하여 다음과 같은 방법으로 기록됩니다.

```
OPNQRYF FILE(CORPDATA/EMPLOYEE)
  FORMAT(FORMAT1)
        QRYSLT(WORKDEPT *EQ ''A00'')
        KEYFLD(LASTNAME)
```

위 예에서 ALWCPYDTA(*NO) 또는 ALWCPYDTA(*YES)가 지정될 때, 데이터베이스 관리자는 색인이 있는 경우 LASTNAME이라는 열이 있는 첫 번째 색인으로부터 색인을 작성할 것입니다. 표에 있는 행은 WHERE 조건에 대응하는 행만을 선택하기 위해 색인을 사용하여 스캔됩니다.

ALWCPYDTA(*OPTIMIZE)가 지정되면, 데이터베이스 관리자는 WORKDEPT의 첫 번째 색인 열과 함께 색인을 사용합니다. 그런 다음 WHERE 조건에 대응하는 모든 행을 복사합니다. 마지막으로 LASTNAME의 값으로 복사된 행을 정렬합니다. 즉시 사용된 색인이 선택될 행을 찾으므로 행 선택 처리가 매우 효율적입니다.

ALWCPYDTA(*OPTIMIZE)는 조회를 처리하는 데 필요한 총 시간을 최적화합니다. 그러나, 첫 번째 행을 수신하는 데 요구되는 시간은 결과 표의 첫 번째 행을 리턴하기 전에 자료를 복사하여야 하므로 증가될 수 있습니다. 응답 시간의 초기 변경은 대화식 표시장치를 표시하는 어플리케이션 또는 조회의 처음 몇 행만을 검색하는 어플리케이션의 경우 중요할 수 있습니다. iSeries용 DB2 Universal Database 조회 Optimizer는 OPTIMIZE절을 사용하여 정렬을 피할 수 있습니다. 자세한 내용은 77 페이지의 『데이터베이스 모니터를 사용하여 조회에 대한 통계 정보 수집』을 참조하십시오.

또한 결합 조작을 포함하는 조회는 결합 순서가 ORDER BY 스펙에 관계없이 최적화될 수 있으므로 ALWCPYDTA(*OPTIMIZE)로부터 이점을 얻을 수 있습니다.

데이터베이스에서 VARCHAR 및 VARGRAPHIC 자료 사용에 대한 추가 정보

가변 길이 열(VARCHAR 또는 VARGRAPHIC) 지원을 사용하여 표의 열 수를 가변 길이로 표를 정의할 수 있습니다. VARCHAR 또는 VARGRAPHIC 지원을 사용하는 경우, 표의 *크기*를 줄일 수 있습니다.

가변 길이 열의 자료는 두 개의 영역 즉, 고정 길이 또는 ALLOCATE 영역과 넘침 영역에 내부적으로 저장됩니다. 디폴트 값이 지정되면, 할당된 길이는 최소한 값의 *크기*와 같습니다. 다음과 같은 점을 사용하여 사용자의 기억장치 영역을 사용하는 최선의 방법을 판별할 수 있습니다.

가변 길이 자료가 있는 표를 정의할 때, ALLOCATE 영역 폭을 결정하여야 합니다. 1차 목표가 다음과 같은 경우:

- **공간 저장:** ALLOCATE(0)를 사용하십시오.
- **성능 향상:** ALLOCATE 영역은 열에 대해 최소한 90% - 95%의 값을 통합할 수 있을 만큼 넓어야 합니다.

공간 저장 및 성능의 평형을 유지하는 것이 중요합니다. 다음 전자 전화번호부 예에서, 다음 자료가 사용됩니다.

- 8,600가지의 이름은 성, 이름 및 중간 이름으로 식별됩니다.
- 마지막, 처음 및 중간 열은 가변 길이입니다.
- 가장 짧은 성은 2자이며 가장 긴 성은 22자입니다.

이 예에서는 가변 길이 열를 사용하여 공간을 저장할 수 있는 방법을 보여줍니다. 고정 길이 열 표는 최대 공간을 사용합니다. 할당 *크기*를 적절하게 연산한 표는 디스크 공간을 적게 사용합니다. 할당 *크기*가 없이 정의되었던 표(모든 자료가 넘침 영역에 저장)는 디스크 공간을 가장 적게 사용합니다.

지원 종류	마지막 이름 최대/할당	처음 이름 최대/할당	중간 이름 최대/할당	총 실제 파일 크기	공간 넘침 행 수
고정 길이	22	22	22	567 K	0
가변 길이	40/10	40/10	40/7	408 K	73
가변 길이 디폴트	40/0	40/0	40/0	373 K	8600

많은 어플리케이션에서, 성능이 고려되어야 합니다. 디폴트 ALLOCATE(0)를 사용하는 경우, 디스크 장치 통신량이 2배가 됩니다. ALLOCATE(0)는 행의 고정 길이 부분 읽기와 공간 넘침 읽기를 위해 두 개의 읽기가 필요합니다. 주의 깊게 ALLOCATE를 선택한 가변 길이 구현은 넘침 및 간격을 최소화하고 성능을 최대화합니다. 표의 *크기*는 고정 길이 구현보다 28% 더 작습니다. 행의 1%가 넘침 영역에 있으므로 두 개의 읽기를 요구하는 액세스가 최소화됩니다. 가변 길이 구현은이 고정 길이 구현과 동일하게 실행됩니다.

ALLOCATE 키워드를 사용하는 표를 작성하려는 경우:

```
CREATE TABLE PHONEDIR
        (LAST    VARCHAR(40) ALLOCATE(10),
         FIRST   VARCHAR(40) ALLOCATE(10),
         MIDDLE  VARCHAR(40) ALLOCATE(7))
```

호스트 변수를 사용하여 가변 길이 열를 삽입 또는 갱신하려면 호스트 변수는 가변 길이여야 합니다. 공백을 고정 길이 호스트 변수에서 절단할 수 없으므로, 고정 길이 호스트 변수를 사용하면 더 많은 행이 공간 넘침으로 넘치게 됩니다. 이렇게 되면 표의 *크기*가 증가됩니다.

다음 예에서, 고정 길이 호스트 변수가 표에 행을 삽입하기 위해 사용됩니다.

```
01  LAST-NAME PIC X(40).
   ...
    MOVE "SMITH" TO LAST-NAME.
  EXEC SQL
      INSERT INTO PHONEDIR
       VALUES(:LAST-NAME, :FIRST-NAME, :MIDDLE-NAME, :PHONE)
  END-EXEC.
```

호스트 변수 LAST-NAME은 가변 길이가 아닙니다. 35개의 공백이 이어지는 스트링 “SMITH”는 VARCHAR 열 LAST에 삽입됩니다. 값은 할당 *크기* 10보다 더 깁니다. 35개의 후미 공백 중의 30개 공백이 넘침 영역에 있습니다.

다음 예에서, 가변 길이 호스트 변수가 표에 행을 삽입하기 위해 사용됩니다.

```
01  VLAST-NAME.
    49 LAST-NAME-LEN PIC S9(4) BINARY.
    49 LAST-NAME-DATA PIC X(40).
   ...
    MOVE "SMITH" TO LAST-NAME-DATA.
    MOVE 5 TO LAST-NAME-LEN.
  EXEC SQL
      INSERT INTO PHONEDIR
     VALUES(:VLAST-NAME, :VFIRST-NAME, :VMIDDLE-NAME, :PHONE)
  END-EXEC.
```

호스트 변수 VLAST-NAME은 가변 길이입니다. 자료의 실제 길이가 5로 설정됩니다. 값이 할당된 길이보다 더 짧습니다. 이것은 열의 고정된 부분에 위치될 수 있습니다.

가변 길이 호스트 변수 사용에 대한 자세한 정보는 호스트 언어를 사용한 SQL 프로그래밍을 참조하십시오.

가변 길이 열이 들어 있는 표에 대해 RGZPFM 명령을 실행하면 성능을 향상시킬 수 있습니다. 사용되고 있지 않은 넘침 영역의 부분은 RGZPFM 명령에 의해 채워집니다. 이것은 넘침 행에 대한 읽기 시간을 줄이고, 참조의 국지성을 증가하며 일련의 일괄 처리에 대해 최적의 순서를 작성합니다.

가변 길이 열에 적합한 최대 길이를 선택하십시오. 너무 긴 길이를 선택하면 프로세스 액세스 그룹(PAG)이 증가됩니다. 대형 PAG는 성능을 저하시킵니다. 대형 최대 길이는 SEQONLY(*YES)의 효율성을 저하시킵니다. 2000바이트를 넘는 가변 길이 열은 키 열로 적합하지 않습니다.

부록 A. 데이터베이스 모니터: DDS

이 부록에는 다음과 같은 데이터베이스 모니터 실제 및 논리 파일을 작성하는 데 사용되는 DDS가 들어 있습니다.

- 『데이터베이스 모니터 실제 파일 DDS』
- 162 페이지의『선택적 데이터베이스 모니터 논리 파일 DDS』

데이터베이스 모니터 실제 파일 DDS

다음 그림은 QSYS/QAQQDBMN 성능 통계 실제 파일(PF)을 작성하는 데 사용되는 DDS를 표시합니다.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* Database Monitor physical file row format
     A*
     A          R QQQDBMN                   TEXT('Database Monitor +
                                              Base Table')
     A            QQRID         15P         TEXT('Row +
                                              ID') +
                                            EDTCDE(4) +
                                            COLHDG('Row' 'ID')
     A            QQTIME          Z         TEXT('Time row was +
                                              created') +
                                            COLHDG('Time' +
                                             'Row' +
                                             'Created')
     A            QQJFLD        46H         TEXT('Join Column') +
                                            COLHDG('Join' 'Column')
     A            QQRDBN        18A         TEXT('Relational +
                                              Database Name') +
                                            COLHDG('Relational' +
                                             'Database' 'Name')
     A            QQSYS          8A         TEXT('System Name') +
                                            COLHDG('System' 'Name')
     A            QQJOB         10A         TEXT('Job Name') +
                                            COLHDG('Job' 'Name')
     A            QQUSER        10A         TEXT('Job User') +
                                            COLHDG('Job' 'User')
     A            QQJNUM         6A         TEXT('Job Number') +
                                            COLHDG('Job' 'Number')
     A            QQUCNT        15P         TEXT('Unique Counter') +
                                            ALWNULL +
                                            COLHDG('Unique' 'Counter')
     A            QQUDEF       100A         VARLEN TEXT('User Defined +
                                                    Column') +
                                            ALWNULL +
                                            COLHDG('User' 'Defined' +
                                             'Column')
     A            QQSTN         15P         TEXT('Statement Number') +
                                            ALWNULL +
                                            COLHDG('Statement' +
                                             'Number')
     A            QQQDTN        15P         TEXT('Subselect Number') +
```

```
                                        ALWNULL +
                                        COLHDG('Subselect' +
                                               'Number')
A            QQQDTL        15P          TEXT('Nested level of +
                                          subselect') +
                                        ALWNULL +
                                        COLHDG('Nested' +
                                               'Level of' +
                                               'Subselect')
A            QQMATN        15P          TEXT('Subselect of +
                                          materialized view') +
                                        ALWNULL +
                                        COLHDG('Subselect' +
                                               'Number of' +
                                               'Materialized View')
A            QQMATL        15P          TEXT('Nested level of +
                                          Views subselect') +
                                        ALWNULL +
                                        COLHDG('Nested Level' +
                                               'of View''s' +
                                               'Subselect')
A            QQTLN         10A          TEXT('Library of +
                                          Table Queried') +
                                        ALWNULL +
                                        COLHDG('Library of' +
                                               'Table' +
                                               'Queried')
A            QQTFN         10A          TEXT('Name of  +
                                          Table Queried') +
                                        ALWNULL +
                                        COLHDG('Name of' +
                                               'Table' +
                                               'Queried')
A            QQTMN         10A          TEXT('Member of +
                                          Table Queried') +
                                        ALWNULL +
                                        COLHDG('Member of' +
                                               'Table' +
                                               'Queried')
A            QQPTLN        10A          TEXT('Base Library') +
                                        ALWNULL +
                                        COLHDG('Library of' +
                                               'Base' +
                                               'Table')
A            QQPTFN        10A          TEXT('Base Table') +
                                        ALWNULL +
                                        COLHDG('Name of' +
                                               'Base' +
                                               'Table')
A            QQPTMN        10A          TEXT('Base Member') +
                                        ALWNULL +
                                        COLHDG('Member of' +
                                               'Base' +
                                               'Table')
A            QQILNM        10A          TEXT('Library of +
                                          Index Used') +
                                        ALWNULL +
                                        COLHDG('Library of' +
```

```
                                        'Index' +
                                        'Used')
A            QQIFNM        10A          TEXT('Name of +
                                         Index Used') +
                                        ALWNULL +
                                        COLHDG('Name of' +
                                         'Index' +
                                         'Used')
A            QQIMNM        10A          TEXT('Member of +
                                         Index Used') +
                                        ALWNULL +
                                        COLHDG('Member of' +
                                         'Index' +
                                         'Used')
A            QQNTNM        10A          TEXT('NLSS Table') +
                                        ALWNULL +
                                        COLHDG('NLSS' 'Table')
A            QQNLNM        10A          TEXT('NLSS Library') +
                                        ALWNULL +
                                        COLHDG('NLSS' 'Library')
A            QQSTIM          Z          TEXT('Start timestamp') +
                                        ALWNULL +
                                        COLHDG('Start' 'Time')
A            QQETIM          Z          TEXT('End timestamp') +
                                        ALWNULL +
                                        COLHDG('End' 'Time')
A            QQKP            1A         TEXT('Index scan-key positioning') +
                                        ALWNULL +
                                        COLHDG('Key' 'Positioning')
A            QQKS            1A         TEXT('Key selection') +
                                        ALWNULL +
                                        COLHDG('Key' 'Selection')
A            QQTOTR        15P          TEXT('Total rows in table') +
                                        ALWNULL +
                                        COLHDG('Total' +
                                         'Rows in' +
                                         'Table')
A            QQTMPR        15P          TEXT('Number of rows in +
                                         temporary') +
                                        ALWNULL +
                                        COLHDG('Number' +
                                         'of Rows' +
                                         'in Temporary')
A            QQJNP         15P          TEXT('Join Position') +
                                        ALWNULL +
                                        COLHDG('Join' 'Position')
A            QQEPT         15P          TEXT('Estimated processing +
                                         time') +
                                        ALWNULL +
                                        COLHDG('Estimated' +
                                         'Processing' +
                                         'Time')
A            QQDSS           1A         TEXT('Data space +
                                         Selection')
                                        ALWNULL +
                                        COLHDG('Data' 'Space' +
                                         'Selection')
A            QQIDXA          1A         TEXT('Index advised') +
```

```
                                    ALWNULL +
                                    COLHDG('Index' 'Advised')
A           QQORDG          1A      TEXT('Ordering') +
                                    ALWNULL +
                                    COLHDG('Ordering')
A           QQGRPG          1A      TEXT('Grouping') +
                                    ALWNULL +
                                    COLHDG('Grouping')
A           QQJNG           1A      TEXT('Join') +
                                    ALWNULL +
                                    COLHDG('Join')
A           QQUNIN          1A      TEXT('Union') +
                                    ALWNULL +
                                    COLHDG('Union')
A           QQSUBQ          1A      TEXT('Subquery') +
                                    ALWNULL +
                                    COLHDG('Subquery')
A           QQHSTV          1A      TEXT('Host Variables') +
                                    ALWNULL +
                                    COLHDG('Host' 'Variables')
A           QQRCDS          1A      TEXT('Row Selection') +
                                    ALWNULL +
                                    COLHDG('Row' 'Selection')
A           QQRCOD          2A      TEXT('Reason Code') +
                                    ALWNULL +
                                    COLHDG('Reason' 'Code')
A           QQRSS          15P      TEXT('Number of rows +
                                      selected or sorted') +
                                    ALWNULL +
                                    COLHDG('Number of' +
                                     'Rows' +
                                     'Selected')
A           QQREST         15P      TEXT('Estimated number +
                                     of rows selected') +
                                    ALWNULL +
                                    COLHDG('Estimated' +
                                     'Rows' +
                                     'Selected')
A           QQRIDX         15P      TEXT('Number of entries +
                                     in index created') +
                                    ALWNULL +
                                    COLHDG('Entries in' +
                                     'Index' +
                                     'Created')
A           QQFKEY         15P      TEXT('Estimated keys for +
                                     index scan-key positioning') +
                                    ALWNULL +
                                    COLHDG('Estimated' +
                                     'Entries for' +
                                     'index scan-key positioning')
A           QQKSEL         15P      TEXT('Estimated keys for +
                                     key selection') +
                                    ALWNULL +
                                    COLHDG('Estimated' +
                                     'Entries for' +
                                     'Key Selection')
A           QQAJN          15P      TEXT('Estimated number +
                                     of joined rows') +
```

```
                                        ALWNULL +
                                        COLHDG('Estimated' +
                                               'Joined' +
                                               'Rows')
A            QQIDXD      1000A          VARLEN(48) +
                                        TEXT('Columns +
                                              for the index advised') +
                                        ALWNULL +
                                        COLHDG('Advised' 'Key' +
                                               'Columns')
A            QQC11          1A          ALWNULL
A            QQC12          1A          ALWNULL
A            QQC13          1A          ALWNULL
A            QQC14          1A          ALWNULL
A            QQC15          1A          ALWNULL
A            QQC16          1A          ALWNULL
A            QQC18          1A          ALWNULL
A            QQC21          2A          ALWNULL
A            QQC22          2A          ALWNULL
A            QQC23          2A          ALWNULL
A            QQI1          15P          ALWNULL
A            QQI2          15P          ALWNULL
A            QQI3          15P          ALWNULL
A            QQI4          15P          ALWNULL
A            QQI5          15P          ALWNULL
A            QQI6          15P          ALWNULL
A            QQI7          15P          ALWNULL
A            QQI8          15P          ALWNULL
A            QQI9          15P          TEXT('Thread +
                                              Identifier') +
                                        ALWNULL +
                                        COLHDG('Thread' +
                                               'Identifier')
A            QQIA          15P          ALWNULL
A            QQF1          15P          ALWNULL
A            QQF2          15P          ALWNULL
A            QQF3          15P          ALWNULL
A            QQC61          6A          ALWNULL
A            QQC81          8A          ALWNULL
A            QQC82          8A          ALWNULL
A            QQC83          8A          ALWNULL
A            QQC84          8A          ALWNULL
A            QQC101        10A          ALWNULL
A            QQC102        10A          ALWNULL
A            QQC103        10A          ALWNULL
A            QQC104        10A          ALWNULL
A            QQC105        10A          ALWNULL
A            QQC106        10A          ALWNULL
A            QQC181        18A          ALWNULL
A            QQC182        18A          ALWNULL
A            QQC183        18A          ALWNULL
A            QQC301        30A          VARLEN(10) ALWNULL
A            QQC302        30A          VARLEN(10) ALWNULL
A            QQC303        30A          VARLEN(10) ALWNULL
A            QQ1000      1000A          VARLEN(48) ALWNULL
A            QQTIM1         Z           ALWNULL
A            QQTIM2         Z           ALWNULL
A*
```

```
A* New columns added for Visual Explain
A*
A            QVQTBL       128A         VARLEN(10) +
                                       TEXT('Queried Table, +
                                             Long Name') +
                                       ALWNULL +
                                       COLHDG('Queried' +
                                              'Table' +
                                              'Long Name')
A            QVQLIB       128A         VARLEN(10) +
                                       TEXT('Queried Library, +
                                             Long Name') +
                                       ALWNULL +
                                       COLHDG('Queried' +
                                              'Library' +
                                              'Long Name')
A            QVPTBL       128A         VARLEN(10) +
                                       TEXT('Base Table, +
                                             Long Name') +
                                       ALWNULL +
                                       COLHDG('Base' +
                                              'Table' +
                                              'Long Name')
A            QVPLIB       128A         VARLEN(10) +
                                       TEXT('Base Library, +
                                             Long Name') +
                                       ALWNULL +
                                       COLHDG('Base' +
                                              'Library' +
                                              'Long Name')
A            QVINAM       128A         VARLEN(10) +
                                       TEXT('Index Used, +
                                             Long Name') +
                                       ALWNULL +
                                       COLHDG('Index' +
                                              'Used' +
                                              'Long Name')
A            QVILIB       128A         VARLEN(10) +
                                       TEXT('Index Used, +
                                             Libary Name') +
                                       ALWNULL +
                                       COLHDG('Index' +
                                              'Used' +
                                              'Library' +
                                              'Name')
A            QVQTBLI        1A         TEXT('Table Long +
                                             Required')
                                       ALWNULL +
                                       COLHDG('Table' +
                                              'Long' +
                                              'Required')
A            QVPTBLI        1A         TEXT('Base Long +
                                             Required')
                                       ALWNULL +
                                       COLHDG('Base' +
                                              'Long' +
                                              'Required')
A            QVINAMI        1A         TEXT('Index Long +
```

```
                                      Required')
                                    ALWNULL +
                                    COLHDG('Index' +
                                           'Long' +
                                           'Required')
A           QVBNDY          1A      TEXT('I/O or CPU +
                                      Bound') +
                                    ALWNULL +
                                    COLHDG('I/O or CPU' +
                                           'Bound')
A           QVJFANO         1A      TEXT('Join +
                                      Fan out') +
                                    ALWNULL +
                                    COLHDG('Join' +
                                           'Fan' +
                                           'Out')
A           QVPARPF         1A      TEXT('Parallel +
                                      Pre-Fetch') +
                                    ALWNULL +
                                    COLHDG('Parallel' +
                                           'Pre-Fetch')
A           QVPARPL         1A      TEXT('Parallel +
                                      Preload') +
                                    ALWNULL +
                                    COLHDG('Parallel' +
                                           'Preload')
A           QVC11           1A      ALWNULL
A           QVC12           1A      ALWNULL
A           QVC13           1A      ALWNULL
A           QVC14           1A      ALWNULL
A           QVC15           1A      ALWNULL
A           QVC16           1A      ALWNULL
A           QVC17           1A      ALWNULL
A           QVC18           1A      ALWNULL
A           QVC19           1A      ALWNULL
A           QVC1A           1A      ALWNULL
A           QVC1B           1A      ALWNULL
A           QVC1C           1A      ALWNULL
A           QVC1D           1A      ALWNULL
A           QVC1E           1A      ALWNULL
A           QVC1F           1A      ALWNULL
A           QWC11           1A      ALWNULL
A           QWC12           1A      ALWNULL
A           QWC13           1A      ALWNULL
A           QWC14           1A      ALWNULL
A           QWC15           1A      ALWNULL
A           QWC16           1A      ALWNULL
A           QWC17           1A      ALWNULL
A           QWC18           1A      ALWNULL
A           QWC19           1A      ALWNULL
A           QWC1A           1A      ALWNULL
A           QWC1B           1A      ALWNULL
A           QWC1C           1A      ALWNULL
A           QWC1D           1A      ALWNULL
A           QWC1E           1A      ALWNULL
A           QWC1F           1A      ALWNULL
A           QVC21           2A      ALWNULL
A           QVC22           2A      ALWNULL
```

```
A            QVC23          2A         ALWNULL
A            QVC24          2A         ALWNULL
A            QVCTIM        15P         TEXT('Cumulative +
                                         Time') +
                                       ALWNULL +
                                       COLHDG('Estimated' +
                                        'Cumulative' +
                                        'Time')
A            QVPARD        15P         TEXT('Parallel Degree, +
                                       Requested') +
                                       ALWNULL +
                                       COLHDG('Parallel' +
                                        'Degree' +
                                        'Requested')
A            QVPARU        15P         TEXT('Parallel Degree, +
                                       Used') +
                                       ALWNULL +
                                       COLHDG('Parallel' +
                                        'Degree' +
                                        'Used')
A            QVPARRC       15P         TEXT('Parallel Limited, +
                                       Reason Code') +
                                       ALWNULL +
                                       COLHDG('Parallel' +
                                        'Limited' +
                                        'Reason Code')
A            QVRCNT        15P         TEXT('Refresh Count') +
                                       ALWNULL +
                                       COLHDG('Refresh' +
                                        'Count')
A            QVFILES       15P         TEXT('Number of, +
                                       Tables Joined')
                                       ALWNULL +
                                       COLHDG('Number of' +
                                        'Tables' +
                                        'Joined')
A            QVP151        15P         ALWNULL
A            QVP152        15P         ALWNULL
A            QVP153        15P         ALWNULL
A            QVP154        15P         ALWNULL
A            QVP155        15P         ALWNULL
A            QVP156        15P         ALWNULL
A            QVP157        15P         ALWNULL
A            QVP158        15P         ALWNULL
A            QVP159        15P         ALWNULL
A            QVP15A        15P         TEXT('Decomposed' +
                                         'Subselect Number') +
                                       ALWNULL +
                                       COLHDG('Decomposed'
                                        'Subselect' +
                                        'Number')
A            QVP15B        15P         TEXT('Number of' +
                                      'Decomposed + Subselects') +
                                       ALWNULL +
                                       COLHDG('Number of' +
                                        'Decomposed' + Subselects')
A            QVP15C        15P         TEXT('Decomposed' +
                                      'Subselect + Reason code') +
```

```
                                    ALWNULL +
                                    COLHDG('Decomposed' +
                                           'Reason' +
                                           'Code')
A           QVP15D        15P       TEXT('Number of first' +
                                         'Decomposed + Subselect') +
                                    ALWNULL +
                                    COLHDG('Starting' + 'Decomposed' +
                                           'Subselect'
A           QVP15E        15P       TEXT('Materialized Union' +
                                         'Level')
                                    ALWNULL +
                                    COLHDG('Materialized' +
                                           'Union' +
                                           'Level')
A           QVP15F        15P       ALWNULL
A           QVC41          4A       ALWNULL
A           QVC42          4A       ALWNULL
A           QVC43          4A       ALWNULL
A           QVC44          4A       ALWNULL
A           QVC81          8A       ALWNULL
A           QVC82          8A       ALWNULL
A           QVC83          8A       ALWNULL
A           QVC84          8A       ALWNULL
A           QVC85          8A       ALWNULL
A           QVC86          8A       ALWNULL
A           QVC87          8A       ALWNULL
A           QVC88          8A       ALWNULL
A           QVC101        10A       ALWNULL
A           QVC102        10A       ALWNULL
A           QVC103        10A       ALWNULL
A           QVC104        10A       ALWNULL
A           QVC105        10A       ALWNULL
A           QVC106        10A       ALWNULL
A           QVC107        10A       ALWNULL
A           QVC108        10A       ALWNULL
A           QVC1281      128A       VARLEN(10) ALWNULL
A           QVC1282      128A       VARLEN(10) ALWNULL
A           QVC1283      128A       VARLEN(10) ALWNULL
A           QVC1284      128A       VARLEN(10) ALWNULL
A           QVC3001      300A       VARLEN(32) ALWNULL
A           QVC3002      300A       VARLEN(32) ALWNULL
A           QVC3003      300A       VARLEN(32) ALWNULL
A           QVC3004      300A       VARLEN(32) ALWNULL
A           QVC3005      300A       VARLEN(32) ALWNULL
A           QVC3006      300A       VARLEN(32) ALWNULL
A           QVC3007      300A       VARLEN(32) ALWNULL
A           QVC3008      300A       VARLEN(32) ALWNULL
A           QVC5001      500A       VARLEN(32) ALWNULL
A           QVC5002      500A       VARLEN(32) ALWNULL
A           QVC1000     1000A       VARLEN(48) ALWNULL
A           QWC1000     1000A       VARLEN(48) ALWNULL
```

선택적 데이터베이스 모니터 논리 파일 DDS

다음 예는 표시된 DDS를 사용하여 작성할 수 있는 여러가지 선택적 논리 파일을 보여줍니다. 각 예 다음에 이어지는 표에 여기에 대한 설명이 있습니다. 이러한 표는 서버와 함께 제공되는 것이 아니며 실행하려면 표를 작성해야 합니다. 이들 파일은 선택적이며 모니터 자료를 분석하는 데 필수적이지 않습니다.

데이터베이스 모니터 논리 표 1000 - SQL 정보에 대한 요약

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A*
     A* DB Monitor logical table 1000 - Summary Row for SQL Information
     A*
     A          R QQQ1000                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
```

```
|       A            QQUSER
|       A            QQJNUM
|       A            QQTHRD                    RENAME(QQI9) +
|                                                COLHDG('Thread' +
|                                                       'Identifier')
|       A            QQUCNT
|       A            QQRCNT                    RENAME(QQI5) +
|                                                COLHDG('Refresh' +
|                                                       'Counter')
|       A            QQUDEF
|       A*
|       A* Information about the SQL statement executed
|       A*
|       A            QQSTN
|       A            QQSTF                     RENAME(QQC11) +
|                                                COLHDG('Statement' +
|                                                       'Function')
|       A            QQSTOP                    RENAME(QQC21) +
|                                                COLHDG('Statement' +
|                                                       'Operation')
|       A            QQSTTY                    RENAME(QQC12) +
|                                                COLHDG('Statement' 'Type')
|       A            QQPARS                    RENAME(QQC13) +
|                                                COLHDG('Parse' 'Required')
|       A            QQPNAM                    RENAME(QQC103) +
|                                                COLHDG('Package' 'Name')
|       A            QQPLIB                    RENAME(QQC104) +
|                                                COLHDG('Package' 'Library')
|       A            QQCNAM                    RENAME(QQC181) +
|                                                COLHDG('Cursor' 'Name')
|       A            QQSNAM                    RENAME(QQC182) +
|                                                COLHDG('Statement' 'Name')
|       A            QQSTIM
|       A            QQSTTX                    RENAME(QQ1000) +
|                                                COLHDG('Statement' 'Text')
|       A            QQSTOC                    RENAME(QQC14) +
|                                                COLHDG('Statement' +
|                                                       'Outcome')
|       A            QQROWR                    RENAME(QQI2) +
|                                                COLHDG('Rows' 'Returned')
|       A            QQDYNR                    RENAME(QQC22) +
|                                                COLHDG('Dynamic' 'Replan')
|       A            QQDACV                    RENAME(QQC16) +
|                                                COLHDG('Data' 'Conversion')
|       A            QQTTIM                    RENAME(QQI4) +
|                                                COLHDG('Total' 'Time' +
|                                                       'Milliseconds')
|       A            QQROWF                    RENAME(QQI3) +
|                                                COLHDG('Rows' 'Fetched')
|       A            QQETIM
|       A            QQTTIMM                   RENAME(QQI6) +
|                                                COLHDG('Total' 'Time')
|                                                       'Microseconds')
|       A            QQSTMTLN                  RENAME(QQI7) +
|                                                COLHDG('Total' +
|                                                       'Statement' +
|                                                       'Length')
|       A            QQIUCNT                   RENAME(QQI1) +
```

```
                                        COLHDG('Insert' 'Unique')
                                               'Count')     A*
      A            QQADDTXT                  RENAME(QWC14) +
                                             COLHDG('Additional' 'SQL')
                                               'Text')
      A*
      A*
      A* Additional information about the SQL statement executed
      A*
      A            QVSQCOD                   RENAME(QQI8) +
                                             COLHDG('SQL' +
                                               'Return' +
                                               'Code')
      A            QVSQST                    RENAME(QQC81) +
                                             COLHDG('SQLSTATE')
      A            QVCLSCR                   RENAME(QVC101) +
                                             COLHDG('CLOSQLCSR' +
                                               'Setting')
      A            QVALWCY                   RENAME(QVC11) +
                                             COLHDG('ALWCPYDTA' +
                                               'Setting')
      A            QVPSUDO                   RENAME(QVC12) +
                                             COLHDG('Pseudo' +
                                               'Open')
      A            QVPSUDC                   RENAME(QVC13) +
                                             COLHDG('Pseudo' +
                                               'Close')
      A            QVODPI                    RENAME(QVC14) +
                                             COLHDG('ODP' +
                                               'Implementation')
      A            QVDYNSC                   RENAME(QVC21) +
                                             COLHDG('Dynamic' +
                                               'Replan' +
                                               'Subtype Code')
      A            QVCMMT                    RENAME(QVC41) +
                                             COLHDG('Commit' +
                                               'Level')
      A            QVBLKE                    RENAME(QVC15) +
                                             COLHDG('Blocking' +
                                               'Enabled')
      A            QVDLYPR                   RENAME(QVC16) +
                                             COLHDG('Delay' +
                                               'Prep')
      A            QVEXPLF                   RENAME(QVC1C) +
                                             COLHDG('SQL' +
                                               'Statement' +
                                               'Explainable')
      A            QVNAMC                    RENAME(QVC17) +
                                             COLHDG('Naming' +
                                               'Convention')
      A            QVDYNTY                   RENAME(QVC18) +
                                             COLHDG('Type of' +
                                               'Dynamic' +
                                               'Processing')
      A            QVOLOB                    RENAME(QVC19) +
                                             COLHDG('Optimize' +
                                               'LOB' +
                                               'Data Types')
```

```
|       A            QVUSRP                    RENAME(QVC1A) +
|                                              COLHDG('User' +
|                                                     'Profile')
|       A            QVDUSRP                   RENAME(QVC1B) +
|                                              COLHDG('Dynamic' +
|                                                     'User' +
|                                                     'Profile')
|       A            QVDFTCL                   RENAME(QVC1281) +
|                                              COLHDG('Default' +
|                                                     'Collection')
|       A            QVPROCN                   RENAME(QVC1282) +
|                                              COLHDG('Procedure' +
|                                                     'Name on' +
|                                                     'CALL')
|       A            QVPROCL                   RENAME(QVC1283) +
|                                              COLHDG('Procedure' +
|                                                     'Library on' +
|                                                     'CALL')
|       A            QVSPATH                   RENAME(QVC1000) +
|                                              COLHDG('SQL' +
|                                                     'Path')
|       A            QVSPATHB                  RENAME(QwC1000) +
|                                              COLHDG('SQL' +
|                                                     'Path' +
|                                                     'Continued')
|       A            QVSPATHC                  RENAME(QVC5001) +
|                                              COLHDG('SQL' +
|                                                     'Path' +
|                                                     'Continued')
|       A            QVSPATHD                  RENAME(QVC5002) +
|                                              COLHDG('SQL' +
|                                                     'Path' +
|                                                     'Continued')
|       A            QVSPATHE                  RENAME(QVC3001) +
|                                              COLHDG('SQL' +
|                                                     'Path' +
|                                                     'Continued')
|       A            QVSPATHF                  RENAME(QVC3002) +
|                                              COLHDG('SQL' +
|                                                     'Path' +
|                                                     'Continued')
|       A            QVSPATHG                  RENAME(QVC30013) +
|                                              COLHDG('SQL' +
|                                                     'Path' +
|                                                     'Continued')
|       A            QVSCHEM                   RENAME(QVC1284) +
|                                              COLHDG('SQL' +
|                                                     'Schema)
|       A*
|       A* Environmental information about the SQL statement executed
|       A*
|       A            QVDFMT                    RENAME(QVC42) +
|                                              COLHDG('Date' +
|                                                     'Format')
|       A            QVDSEP                    RENAME(QWC11) +
|                                              COLHDG('Date' +
|                                                     'Separator')
|       A            QVTFMT                    RENAME(QVC43) +
```

```
|                                         COLHDG('Time' +
|                                                'Format')
|       A            QVTSEP               RENAME(QWC12) +
|                                         COLHDG('Time' +
|                                                'Separator')
|       A            QVDPNT               RENAME(QWC13) +
|                                         COLHDG('Decimal' +
|                                                'Point')
|       A            QVSRTSQ              RENAME(QVC104) +
|                                         COLHDG('Sort' +
|                                                'Sequence' +
|                                                'Table')
|       A            QVSRTSL              RENAME(QVC105) +
|                                         COLHDG('Sort' +
|                                                'Sequence' +
|                                                'Library')
|       A            QVLNGID              RENAME(QVC44) +
|                                         COLHDG('Language' +
|                                                'ID')
|       A            QVCNTID              RENAME(QVC23) +
|                                         COLHDG('Country' +
|                                                'ID')
|       A            QVFNROW              RENAME(QQIA) +
|                                         COLHDG('FIRST n' +
|                                                'ROWS Value')
|       A            QVOPTRW              RENAME(QQF1) +
|                                         COLHDG('OPTIMIZE FOR' +
|                                                'n ROWS Value')
|       A            QVRAPRC              RENAME(QVC22) +
|                                         COLHDG('SQL Access' +
|                                                'Plan Rebuild' +
|                                                'Reason Code')
|       A            QVNOSV               RENAME(QVC24) +
|                                         COLHDG('Access Plan' +
|                                                'Not Saved' +
|                                                'Reason Code')
|       A            QVCTXT               RENAME(QVC81) +
|                                         COLHDG('Transaction' +
|                                                'Context' +
|                                                'ID')
|       A            QVAGMRK              RENAME(QVP152) +
|                                         COLHDG('Activation' +
|                                                'Group' +
|                                                'Mark')
|       A            QVCURTHR             RENAME(QVP153) +
|                                         COLHDG('Open Cursor' +
|                                                'Threshold')
|       A            QVCURCNT             RENAME(QVP154) +
|                                         COLHDG('Open Cursor' +
|                                                'Close' +
|                                                'Count')
|       A            QVLCKLMT             RENAME(QVP155) +
|                                         COLHDG('Commit' +
|                                                'Lock' +
|                                                'Limit')
|       A            QVSQLMIXED           RENAME(QWC15) +
|                                         COLHDG('SQL' +
|                                                'Mixed' +
```

```
                                        'Constants')
     A            QVSQLSUPP             RENAME(QWC16) +
                                        COLHDG('SQL' +
                                               'Suppress' +
                                               'Warnings')
     A            QVSQLASCII            RENAME(QWC17) +
                                        COLHDG('SQL' +
                                               'Translate' +
                                               'ASCII')
     A            QVSQLCACHE            RENAME(QWC18) +
                                        COLHDG('SQL' +
                                               'Statement' +
                                               'Cache')
     A          K QQJFL
     A          S QQRID                 CMP(EQ 1000)
```

표 13. QQQ1000 - SQL 정보에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQRCNT	QQI5	고유 화면정리 카운터
QQUDEF	QQUDEF	사용자 정의 열
QQSTN	QQSTN	명령문 번호(명령문에 따라 고유함)
QQSTF	QQC11	명령 함수: • S - 선택 • U - 갱신 • I - 삽입 • D - 삭제 • L - 자료 정의어 • O - 기타

표 13. QQQ1000 - SQL 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQSTOP	QQC21	명령문 조작: • AL - 표 변경 • CA - 호출 • CC - 콜렉션 작성 • CD - 유형 작성 • CF - 함수 작성 • CG - 트리거 작성 • CI - 색인 작성 • CL - 닫기 • CM - 확약 • CN - 연결 • CO - 주석 달기 • CP - 프로시듀어 작성 • CS - 별명/동의어 작성 • CT - 표 작성 • CV - 보기 작성 • DE - 설명 • DI - 단절 • DL - 삭제 • DM - 매개변수 마커 설명 • DP - 프로시듀어 선언 • DR - 드롭(drop) • DT - 표 설명 • EI - 즉시 실행 • EX - 실행 • FE - 페치 • FL - 자유 위치 지정자 • GR - 부여 • HC - 하드 닫기 • HL - 보류 위치 지정자 • IN - 삽입 • JR - 재사용된 서버 작업 • LK - 잠금 • LO - 레이블 • MT - 텍스트 계속 • OP - 열기 • PD - 준비 및 설명 • PR - 준비 • RB - 롤백 저장점 • RE - 해제 • RO - 롤백

표 13. QQQ1000 - SQL 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQSTOP(계속)	QQC21	• RS - 해제 저장점 • RT - 표 이름 변경 • RV - 취소 • SA - 저장점 • SC - 연결 설정 • SI - 선택 위치 • SP - 경로 설정 • SR - 결과 세트 설정 • SS - 현재 스키마 설정 • ST - 트랜잭션 설정 • SV - 변수 설정 • UP - 갱신 • VI - 들어갈 값
QQSTTY	QQC12	명령문 유형: • D - 동적 명령문 • S - 정적 명령문
QQPARS	QQC13	분석 필요(Y/N)
QQPNAM	QQC103	현재 SQL문이 들어있는 패키지명 또는 프로그램명
QQPLIB	QQC104	패키지가 들어 있는 라이브러리명
QQCNAM	QQC181	이 SQL문에 대응하는 커서명(적용되는 경우)
QQSNAM	QQC182	SQL문에 대한 명령문명(적용되는 경우)
QQSTIM	QQSTIM	이 명령문이 입력된 시간
QQSTTX	QQ1000	명령문 텍스트
QQSTOC	QQC14	명령문 결과 • S - 성공함 • U - 실패함
QQROWR	QQI2	리턴된 결과 행수

표 13. QQQ1000 - SQL 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQDYNR	QQC22	동적 재계획(액세스 계획 리빌드) • NA - 재계획 없음 • NR - 새 릴리스에 대해 SQL QDT 리빌드 • A1 - 표 또는 멤버가 액세스 계획이 최종 빌드되었을 때 참조된 표 또는 멤버와 동일한 오브젝트가 아닙니다. 오브젝트가 다른 몇가지 이유는 다음과 같습니다. – 오브젝트가 삭제되고 리빌드되었음. – 오브젝트가 저장되고 복원되었음. – 라이브러리 리스트가 변경되었음. – 오브젝트의 이름이 변경되었음. – 오브젝트가 이동되었음. – 오브젝트가 다른 오브젝트로 대체되었음. – 이번이 조회가 들어 있는 오브젝트가 복원된 이후 해당 조회를 처음으로 실행하는 경우임. • A2 - 액세스 계획이 재사용 가능한 ODP(열린 자료 경로)를 사용하도록 빌드되었는데 Optimizer가 이 호출에 대해 재사용할 수 없는 ODP를 사용하기로 선택하였습니다. • A3 - 액세스 계획이 재사용할 수 없는 ODP(열린 자료 경로)를 사용하도록 빌드되었으며 Optimizer가 이 호출에 대해 재사용 가능한 ODP를 사용하기로 선택하였습니다. • A4 - 표 멤버의 행 수가 액세스 계획이 최종 빌드된 이후로 10% 이상 변경되었습니다. • A5 - 이 조회의 표 중 하나에 대해 신규 색인이 존재합니다. • A6 - 이 액세스 계획에 사용되었던 색인이 더 이상 존재하기 않거나 더 이상 유효하지 않습니다. • A7 - 시스템 프로그래밍 변경으로 인해 OS/400 조회에 리빌드된 액세스 계획이 필요합니다. • A8 - 현재 작업의 CCSID가 액세스 계획을 최종 작성한 작업의 CCSID와 다릅니다. • A9 - 다음 중 하나 이상의 값이 현재 작업에 대해 이 액세스 계획을 최종 작성했을 때의 작업에 대한 값과 다릅니다. – 날짜 형식 – 날짜 분리자 – 시간 형식 – 시간 분리자 • AA - 지정된 정렬 순서표가 이 액세스 계획이 작성되었던 때 사용되었던 정렬 순서표와 다릅니다. • AB - 기억장치 풀이 변경되었거나 CHGQRYA 명령의 DEGREE 매개변수가 변경되었습니다. • AC - 시스템 피처 DB2 복수 시스템이 설치 또는 제거되었습니다. • AD - 정도 조회 속성 값이 변경되었습니다. • AE - 보기가 상위 레벨 언어에 의해 열리거나 구체화되는 중입니다. • AF - 사용자 정의 유형 또는 사용자 정의 기능이 액세스 계획에서 참조된 유형 또는 기능과 동일한 오브젝트가 아니거나 SQL 경로가 액세스 계획이 빌드될 때와 동일하지 않습니다.
QQDYNR(계속)	QQC22	• B0 - 지정된 옵션이 조회 옵션 파일의 결과로서 변경되었습니다. • B1 - 액세스 계획이 현재 작업과 다른 확약 제어 레벨로 생성되었습니다. • B2 - 액세스 계획이 이전 액세스 계획과 다른 정적 커서 응답 설정 크기로 생성되었습니다.

표 13. QQQ1000 - SQL 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQDACV	QQC16	자료 변환 • N - 아니오 • 0 - 적용 가능하지 않음. • 1 - 길이가 일치하지 않음. • 2 - 숫자 유형이 일치하지 않음. • 3 - C 호스트 변수가 NUL 종료됨. • 4 - 호스트 변수 또는 열이 가변 길이이며 나머지는 가변 길이가 아님. • 5 - CCSID 변환. • 6 - DRDA 및 널(null) 허용, 가변 길이, 부분적 행에 포함, 파생 표현 식 또는 충분하지 못한 호스트 변수를 사용하여 블록화된 페치. • 7 - 자료, 시간 또는 시간소인 열. • 8 - 호스트 변수가 너무 많음. • 9 - 삽입 대상 표가 SQL 표가 아님.
QQTTIM	QQI4	이 명령문에 대한 총 시간(밀리초 단위). 페치의 경우, 여기에는 커서의 OPEN에 대한 모든 페치가 들어 있습니다.
QQROWF	QQI3	커서에 대해 페치된 총 행
QQETIM	QQETIM	시간 SQL 요구 완료
QQTTIMM	QQI6	이 명령문에 대한 총 시간(밀리초 단위). 페치의 경우, 여기에는 커서의 OPEN에 대한 모든 페치가 들어 있습니다.
QQSTMTLN	QQI7	SQL문의 길이
QQIUCNT	QQI1	INSERT와 연관된 QDT에 대한 고유 조회 계수. QQUCNT에는 명령문의 WHERE 부분과 연관된 QDT에 대한 고유 조회 계수가 들어 있습니다.
QVSQCOD	QQI8	SQL 리턴 코드
QVSQST	QQC81	SQLSTATE
QVCLSCR	QVC101	닫기 커서. 가능한 값은 다음과 같습니다. • *ENDJOB - 작업이 종료될 때 SQL 커서가 닫힙니다. • *ENDMOD - 모듈이 종료될 때 SQL 커서가 닫힙니다. • *ENDPGM - 프로그램이 종료될 때 SQL 커서가 닫힙니다. • *ENDSQL - 호출 스택에 대한 첫 번째 SQL 프로그램이 종료될 때 SQL 커서가 닫힙니다. • *ENDACTGRP - 활성 그룹이 종료될 때 SQL 커서가 닫힙니다.
QVALWCY	QVC11	ALWCPYDTA 설정(Y/N/O) • Y - 자료 사본을 사용할 수 있음 • N - 자료 사본을 사용할 수 없음 • O - Optimizer가 성능을 위해 자료의 사본을 사용하도록 선택할 수 있음
QVPSUDO	QVC12	Open을 트리거할 수 있는 SQL 조작에 대한 의사 Open(Y/N). • OP - Open • IN - Insert • UP - Update • DL - Delete • SI - Select Into • SV - Set • VI - Values into 모든 조작에 대해 공백이 될 수 있습니다.

표 13. QQQ1000 - SQL 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVPSUDC	QVC13	Close를 트리거할 수 있는 SQL 조작에 대한 의사 Close(Y/N). • CL - Close • IN - Insert • UP - Update • DL - Delete • SI - Select Into • SV - Set • VI - Values into 모든 조작에 대해 공백이 될 수 있습니다.
QVODPI	QVC14	ODP 구현 • R - 재사용 가능한 ODP • N - 재사용할 수 없는 ODP • ' ' - 사용되지 않는 열
QQDYNSC	QVC21	동적 재계획, 부속 유형 이유 코드
QVCMMT	QVC41	확약 제어 레벨. 가능한 값은 다음과 같습니다. • NC • UR • CS • CSKL • RS • RR
QVBLKE	QVC15	블로킹 유형. 가능한 값은 다음과 같습니다. • S - 단일 행, ALWBLK(*READ) • F - 한 행을 강제, ALWBLK(*NONE) • L - 제한된 블록, ALWBLK(*ALLREAD)
QVDLYPR	QVC16	지연 준비(Y/N)
QVEXPLF	QVC1C	SQL문이 설명 가능함(Y/N)
QVNAMC	QVC17	명명 규칙. 가능한 값은 다음과 같습니다. • N - 시스템 명명 규칙 • S - SQL 명명 규칙
QVDYNTY	QVC18	동적 처리의 유형 • E - 확장 동적 처리 • S - 시스템 광역 캐시 • L - 로컬 준비된 명령문
QVOLOB	QVC19	LOB 자료 유형 최적화(Y/N)
QVUSRP	QVC1A	컴파일된 프로그램이 실행될 때 사용된 사용자 프로파일. 가능한 값은 다음과 같습니다. • N = 사용자 프로파일은 명명 규칙에 의해 판별됨. *SQL의 경우, USRPRF(*OWNER)가 사용됨. *SYS의 경우, USRPRF(*USER)가 사용됨. • U = USRPRF(*USER)가 사용됨. • O = USRPRF(*OWNER)가 사용됨.
QVDUSRP	QVC1B	동적 SQL문에 대해 사용된 사용자 프로파일. • U = USRPRF(*USER)가 사용됨. • O = USRPRF(*OWNER)가 사용됨.
QVDFTCL	QVC1281	디폴트 콜렉션명.
QVPROCN	QVC1282	SQL에 대한 CALL 프로시듀어명.
QVPROCL	QVC1283	SQL에 대한 CALL 프로시듀어 라이브러리.
QVSPATH	QVC1000	정적 SQL문에 대한 사용자 정의 유형, 기능 및 프로시듀어를 찾기 위해 사용된 경로.
QVSPATHB	QWC1000	필요한 경우, SQL 경로의 연속. SQL 경로의 1001-2000바이트를 포함합니다.
QVSPATHC	QWC5001	필요한 경우, SQL 경로의 연속. SQL 경로의 2001-2500바이트를 포함합니다.
QVSPATHD	QWC5002	필요한 경우, SQL 경로의 연속. SQL 경로의 2501-3000바이트를 포함합니다.

표 13. QQQ1000 - SQL 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVSPATHE	QWC3001	필요한 경우, SQL 경로의 연속. SQL 경로의 3001-3300바이트를 포함합니다.
QVSPATHF	QWC3002	필요한 경우, SQL 경로의 연속. SQL 경로의 3301-3600바이트를 포함합니다.
QVSPATHG	QWC3003	필요한 경우, SQL 경로의 연속. SQL 경로의 3601-3900바이트를 포함합니다.
QVSCHEM	QVC1284	SQL 현재 스키마
QVDFMT	QVC42	자료 형식. 가능한 값은 다음과 같습니다. • ISO • USA • EUR • JIS • MDY • DMY • YMD
QVDSEP	QWC11	날짜 분리자. 가능한 값은 다음과 같습니다. • "/" • "." • "," • "-" • " "
QVTFMT	QVC43	시간 형식. 가능한 값은 다음과 같습니다. • ISO • USA • EUR • JIS • HMS
QVTSEP	QWC12	시간 분리자. 가능한 값은 다음과 같습니다. • ":" • "." • "," • " "
QVDPNT	QWC13	소수점. 가능한 값은 다음과 같습니다. • "." • ","
QVSRTSQ	QVC104	정렬 순서 표
QVSRTSL	QVC105	정렬 순서 라이브러리
QVLNGID	QVC44	언어 ID
QVCNTID	QVC23	국가 ID
QVFNROW	QQIA	FIRST n ROWS 절에 지정되는 값
QVOPTRW	QQF1	OPTIMIZE FOR n ROWS 절에 지정되는 값
QVRAPRC	QVC22	SQL 액세스 계획의 리빌드 이유 코드. 가능한 이유는 다음과 같습니다. (이유 리스트를 추가하십시오.)
QVNOSV	QVC24	이유 코드를 저장하지 않은 액세스 계획. 가능한 이유는 다음과 같습니다. (이유 리스트를 추가하십시오.)
QVCTXT	QVC81	트랜잭션 문맥 ID
QVAGMRK	QVP152	활성 그룹 마크
QVCCURTHR	QVP153	열기 커서 임계값
QVCCURCNT	QVP154	열기 커서 닫은 횟수
QVLCKLMT	QVP155	확약 제어 잠금 제한
QVSQLMIXED	QWC15	SQL 혼합 상수 사용(Y/N)
QVSQLSUPP	QWC16	SQL 경고 메시지 억제(Y/N)
QVSQLASCII	QWC17	ASCII를 작업에 변환(Y/N)
QVSQLCACHE	QWC18	시스템 범위 SQL문 캐시 사용(Y/N)

데이터베이스 모니터 논리 표 3000 - 표 스캔에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3000 - Summary Row for Table Scan
     A*
     A          R QQQ3000                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                RENAME(QVP15E) +
                                           COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
                                            COLHDG('Decomposed' +
                                                   'Reason' +
                                                   'Code')
     A            QDQDTS                    RENAME(QVP15D) +
                                            COLHDG('Starting' +
                                                   'Decomposed' +
                                                   'Subselect')
     A            QQTLN
     A            QQTFN
     A            QQTMN
     A            QQPTLN
     A            QQPTFN
     A            QQPTMN
     A            QQTOTR
     A            QQREST
     A            QQAJN
     A            QQEPT
     A            QQJNP
     A            QQJNDS                    RENAME(QQI1) +
                                            COLHDG('Data Space' +
                                                   'Number')
```

```
|       A           QQJNMT                  RENAME(QQC21) +
|                                           COLHDG('Join' 'Method')
|       A           QQJNTY                  RENAME(QQC22) +
|                                           COLHDG('Join' 'Type')
|       A           QQJNOP                  RENAME(QQC23) +
|                                           COLHDG('Join' 'Operator')
|       A           QQIDXK                  RENAME(QQI2) +
|                                           COLHDG('Advised' +
|                                                  'Primary' +
|                                                  'Keys')
|       A           QQDSS
|       A           QQIDXA
|       A           QQRCOD
|       A           QQIDXD
|       A           QVQTBL
|       A           QVQLIB
|       A           QVPTBL
|       A           QVPLIB
|       A           QVBNDY
|       A           QVRCNT
|       A           QVJFANO
|       A           QVFILES
|       A           QVPARPF
|       A           QVPARPL
|       A           QVPARD
|       A           QVPARU
|       A           QVPARRC
|       A           QVCTIM
|       A           QVSKIPS                 RENAME(QQC11) +
|                                           COLHDG('Skip' +
|                                                  'Sequential')
|       A           QVTBLSZ                 RENAME(QQI3) +
|                                           COLHDG('Actual' +
|                                                  'Table' 'Size')
|       A           QVTSFLDS                RENAME(QVC3001) +
|                                           COLHDG('Columns for' +
|                                                  'Data Space' +
|                                                  'Selection')
|       A           QVDVFLD                 RENAME(QQC14) +
|                                           COLHDG('Derived' +
|                                                  'Column' +
|                                                  'Selection')
|       A           QVDVFLDS                RENAME(QVC3002) +
|                                           COLHDG('Columns for' +
|                                                  'Derived' +
|                                                  'Selection')
|       A           QVRDTRG                 RENAME(QQC18) +
|                                           COLHDG('Read' +
|                                                 'Trigger')
|       A           QVCARD                  RENAME(QVP157) +
|                                           COLHDG('Cardinality')
|       A           QVUTSP                  RENAME(QVC1281) +
|                                           COLHDG('Specific' +
|                                                  'Name')
|       A           QVULSP                  RENAME(QQC1282) +
|                                           COLHDG('Specific' +
```

```
                                                'Schema')
      A           K QQJFLD
      A           S QQRID                     CMP(EQ 3000)
```

표 14. QQQ3000 - 표 스캔에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 표에 대한 기준 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QQTOTR	QQTOTR	표의 총 행
QQREST	QQREST	선택된 예상 행 수
QQAJN	QQAJN	결합된 예상 행 수
QQEPT	QQEPT	예상 처리 시간(초 단위)
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI1	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합

표 14. QQQ3000 - 표 스캔에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QQIDXK	QQI2	색인 스캔 키 위치지정을 사용하는 권장된 열 수
QQDSS	QQDSS	자료 공간 선택 • Y - 예 • N - 아니오
QQIDXA	QQIDXA	권장된 색인 • Y - 예 • N - 아니오
QQRCOD	QQRCOD	이유 코드 • T1 - 색인이 없습니다. • T2 - 색인이 있지만 사용할 수 없습니다. • T3 - Optimizer가 사용할 수 있는 색인에 대한 표 스캔을 선택하였습니다.
QQIDXD	QQIDXD	권장된 색인에 대한 열
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름
QVPTBL	QVPTBL	기준 표, 긴 이름
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QVBNDY	QVBNDY	I/O 또는 CPU 바인드. 가능한 값은 다음과 같습니다. • I - I/O 바인드 • C - CPU 바인드
QVRCNT	QVRCNT	고유 화면정리 카운터

표 14. QQQ3000 - 표 스캔에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(Y/N)
QVPARD	QVPARD	요구된 병렬 정도
QVPARU	QVPARU	사용된 병렬 정도
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QVSKIPS	QQC11	건너뜀 순차적 표 스캔(Y/N)
QVTBLSZ	QQI3	조회되는 표의 크기
QVTSFLDS	QVC3001	자료 공간 선택에 사용되는 열
QVDVFLD	QQC14	파생된 열 선택(Y/N)
QVDVFLDS	QVC3002	파생된 열 선택에 대해 사용된 열
QVRDTRG	QQC18	트리거 읽기(Y/N)
QVCard	QVP157	사용자 정의 표 함수 기능
QVUTSP	QVC1281	사용자 정의 표 함수 고유명
QVULSP	QVC1282	사용자 정의 표 함수 고유 스키마

데이터베이스 모니터 논리 표 3001 - 사용된 색인에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3001 - Summary Row for Index Used
     A*
     A          R QQQ3001                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
```

```
|       A            QQMATN
|       A            QQMATL
|       A            QQMATULVL               RENAME(QVP15E) +
|                                            COLHDG('Materialized' +
|                                                   'Union' +
|                                                   'Level')
|       A            QDQDTN                   RENAME(QVP15A) +
|                                             COLHDG('Decomposed' +
|                                                    'Subselect' +
|                                                    'Number')
|       A            QDQDTT                   RENAME(QVP15B) +
|                                             COLHDG('Number of' +
|                                                    'Decomposed' +
|                                                    'Subselects')
|       A            QDQDTR                   RENAME(QVP15C) +
|                                             COLHDG('Decomposed' +
|                                                    'Reason' +
|                                                    'Code')
|       A            QDQDTS                   RENAME(QVP15D) +
|                                             COLHDG('Starting' +
|                                                    'Decomposed' +
|                                                    'Subselect')
|       A            QQTLN
|       A            QQTFN
|       A            QQTMN
|       A            QQPTLN
|       A            QQPTFN
|       A            QQPTMN
|       A            QQILNM
|       A            QQIFNM
|       A            QQIMNM
|       A            QQTOTR
|       A            QQREST
|       A            QQFKEY
|       A            QQKSEL
|       A            QQAJN
|       A            QQEPT
|       A            QQJNP
|       A            QQJNDS                   RENAME(QQI1) +
|                                             COLHDG('Data Space' +
|                                                    'Number')
|       A            QQJNMT                   RENAME(QQC21) +
|                                             COLHDG('Join' 'Method')
|       A            QQJNTY                   RENAME(QQC22) +
|                                             COLHDG('Join' 'Type')
|       A            QQJNOP                   RENAME(QQC23) +
|                                             COLHDG('Join' 'Operator')
|       A            QQIDXPK                  RENAME(QQI2) +
|                                             COLHDG('Advised' +
|                                                    'Primary' +
|                                                    'Keys')
|       A            QQKP
|       A            QQKPN                    RENAME(QQI3) +
|                                             COLHDG('Number of' 'Key' +
|                                                    'Positioning' +
|                                                    'Columns')
|       A            QQKS
|       A            QQDSS
```

```
|       A            QQIDXA
|       A            QQRCOD
|       A            QQIDXD
|       A            QQCST                    RENAME(QQC11) +
|                                               COLHDG('Index' +
|                                                      'Is a' +
|                                                      'Constraint')
|       A            QQCSTN                   RENAME(QQ1000) +
|                                               COLHDG('Constraint' +
|                                                      'Name')
|       A*
|       A            QVQTBL
|       A            QVQLIB
|       A            QVPTBL
|       A            QVPLIB
|       A            QVINAM
|       A            QVILIB
|       A            QVBNDY
|       A            QVRCNT
|       A            QVJFANO
|       A            QVFILES
|       A            QVPARPF
|       A            QVPARPL
|       A            QVPARD
|       A            QVPARU
|       A            QVPARRC
|       A            QVCTIM
|       A            QVKOA                    RENAME(QVC14) +
|                                               COLHDG('Index' +
|                                                      'Only' +
|                                                      'Access')
|       A            QVIDXM                   RENAME(QQC12) +
|                                               COLHDG('Index' +
|                                                      'fits in' +
|                                                      'Memory')
|       A            QVIDXTY                  RENAME(QQC15) +
|                                               COLHDG('Index' +
|                                                      'Type')
|       A            QVIDXUS                  RENAME(QVC12) +
|                                               COLHDG('Index' +
|                                                      'Usage')
|       A            QVIDXN                   RENAME(QQI4) +
|                                               COLHDG('Number' +
|                                                      'Index' +
|                                                      'Entries')
|       A            QVIDXUQ                  RENAME(QQI5) +
|                                               COLHDG('Number' +
|                                                      'Unique' +
|                                                      'Values')
|       A            QVIDXPO                  RENAME(QQI6) +
|                                               COLHDG('Percent' +
|                                                      'Overflow')
|       A            QVIDXVZ                  RENAME(QQI7) +
|                                               COLHDG('Vector' +
|                                                      'Size')
|       A            QVIDXSZ                  RENAME(QQI8) +
|                                               COLHDG('Index' +
|                                                      'Size')
```

```
|       A           QVIDXPZ                 RENAME(QQIA) +
|                                           COLHDG('Index' +
|                                                  'Page' +
|                                                  'Size')
|       A           QQPSIZ                  RENAME(QVP154) +
|                                           COLHDG('Pool' +
|                                                  'Size')
|       A           QQPID                   RENAME(QVP155) +
|                                           COLHDG('Pool' +
|                                                  'ID')
|       A           QVTBLSZ                 RENAME(QVP156) +
|                                           COLHDG('Base' +
|                                                  'Table' +
|                                                  'Size')
|       A           QVSKIPS                 RENAME(QQC16) +
|                                           COLHDG('Skip' +
|                                                  'Sequential')
|       A           QVIDXTR                 RENAME(QVC13) +
|                                           COLHDG('Tertiary' +
|                                                  'Indexes'
|                                                  'Exist')
|       A           QVTSFLDS                RENAME(QVC3001) +
|                                           COLHDG('Columns for' +
|                                                  'Data Space' +
|                                                  'Selection')
|       A           QVDVFLD                 RENAME(QVC12 +
|                                           COLHDG('Derived' +
|                                                  'Column' +
|                                                  'Selection')
|       A           QVDVFLDS                RENAME(QVC3002) +
|                                           COLHDG('Columns for' +
|                                                  'Derived' +
|                                                  'Selection')
|       A           QVSKEYP                 RENAME(QVC3003) +
|                                           COLHDG('Key' +
|                                                  'Positioning' +
|                                                  'Columns')
|       A           QVSKEYS                 RENAME(QVC3004) +
|                                           COLHDG('Key' +
|                                                  'Selection' +
|                                                  'Columns')
|       A           QVJKEYS                 RENAME(QVC3005) +
|                                           COLHDG('Join' +
|                                                  'Selection' +
|                                                  'Columns')
|       A           QVOKEYS                 RENAME(QVC3006) +
|                                           COLHDG('Ordering' +
|                                                  'Columns')
|       A           QVGKEYS                 RENAME(QVC3007) +
|                                           COLHDG('Grouping' +
|                                                  'Columns')
|       A           QVRDTRG                 RENAME(QQC18) +
|                                           COLHDG('Read' +
|                                                  'Trigger')
|       A           QVCard                  RENAME(QVP157) +
|                                           COLHDG('Cardinality')
|       A           QVUTSP                  RENAME(QVC1281) +
|                                           COLHDG('Specific +
```

```
                                        'Name')
A            QVULSP                     RENAME(QVC1282) +
                                        COLHDG('Specific +
                                        'Schema')
A          K QQJFLD
A          S QQRID                      CMP(EQ 3001)
```

표 15. QQQ3001 - 사용된 색인에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 표에 대한 기준 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QQILNM	QQILNM	액세스하기 위해 사용되는 색인의 라이브러리명
QQIFNM	QQIFNM	액세스하기 위해 사용되는 색인명
QQIMNM	QQIMNM	액세스하기 위해 사용되는 색인의 멤버명
QQTOTR	QQTOTR	기준 표의 총 행
QQREST	QQREST	선택된 예상 행 수
QQFKEY	QQFKEY	색인 스캔 키 위치 지정을 통해 선택된 열
QQKSEL	QQKSEL	색인 스캔 키 선택을 통해 선택된 열
QQAJN	QQAJN	결합된 예상 행 수
QQEPT	QQEPT	예상 처리 시간(초 단위)

표 15. QQQ3001 - 사용된 색인에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI1	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QQIDXPK	QQI2	색인 스캔 키 위치지정을 사용하는 권장된 키 열 수
QQKP	QQKP	색인 스캔 키 위치지정 • Y - 예 • N - 아니오
QQKPN	QQI3	사용된 색인에 대해 색인 스캔 키 위치지정을 사용하는 열 수
QQKS	QQKS	색인 스캔 키 선택 • Y - 예 • N - 아니오
QQDSS	QQDSS	자료 공간 선택 • Y - 예 • N - 아니오
QQIDXA	QQIDXA	권장된 색인 • Y - 예 • N - 아니오

표 15. QQQ3001 - 사용된 색인에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQRCOD	QQRCOD	이유 코드 • I1 - 행 선택 • I2 - 순서화/그룹화 • I3 - 행 선택 및 순서화/그룹화 • I4 - 내포된 루프 결합 • I5 - 비트맵 처리를 사용하는 행 선택
QQIDXD	QQIDXD	권장된 색인에 대한 열
QQCST	QQC11	색인이 제한 조건임(Y/N)
QQCSTN	QQ1000	제한 조건명
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름
QVPTBL	QVPTBL	기준 표, 긴 이름
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QVINAM	QVINAM	사용된 색인(또는 제한 조건)의 이름, 긴 이름
QVILIB	QVILIB	사용된 색인의 라이브러리, 긴 이름
QVBNDY	QVBNDY	I/O 또는 CPU 바인드. 가능한 값은 다음과 같습니다. • I - I/O 바인드 • C - CPU 바인드
QVRCNT	QVRCNT	고유 화면정리 카운터
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(Y/N)
QVPARD	QVPARD	요구된 병렬 정도
QVPARU	QVPARU	사용된 병렬 정도
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QVKOA	QVC14	색인 전용 액세스(Y/N)
QVIDXM	QQC12	색인이 메모리에 적합함(Y/N)

표 15. QQQ3001 - 사용된 색인에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVIDXTY	QQC15	색인 유형. 가능한 값은 다음과 같습니다. • B - 2진 기수 색인 • C - 제한 조건(2진 기수) • E - 코드화 벡터 색인(EVI) • X - 조회 작성 임시 색인
QVIDXUS	QVC12	색인 사용. 가능한 값은 다음과 같습니다. • P - 1차 색인 • T - 3차 (및) 색인
QVIDXN	QQI4	색인 항목 수
QVIDXUQ	QQI5	고유 키 값의 수
QVIDXPO	QQI6	퍼센트 넘침
QVIDXVZ	QQI7	벡터 크기
QVIDXSZ	QQI8	색인 크기
QVIDXPZ	QQIA	색인 페이지 크기
QQPSIZ	QVP154	풀(pool) 크기
QQPID	QVP155	풀(pool) ID
QVTBLSZ	QVP156	표 크기
QVSKIPS	QQC16	건너뜀 순차적 표 스캔(Y/N)
QVIDXTR	QVC13	3차 색인이 존재함(Y/N)
QVTSFLDS	QVC3001	자료 공간 선택에 사용되는 열
QVDVFLD	QQC14	파생된 열 선택(Y/N)
QVDVFLDS	QVC3002	파생된 열 선택에 대해 사용된 열
QVSKEYP	QVC3003	색인 스캔 키 위치지정에 사용되는 열
QVSKEYS	QVC3004	색인 스캔 키 선택에 사용되는 열
QVJKEYS	QVC3005	결합 선택에 사용되는 열
QVOKEYS	QVC3006	순서화에 사용되는 열
QVGKEYS	QVC3007	그룹화에 사용되는 열
QVRDTRG	QQC18	트리거 읽기(Y/N)
QVCard	QVP157	사용자 정의 표 함수 기능
QVUTSP	QVC1281	사용자 정의 표 함수 고유명
QVULSP	QVC1282	사용자 정의 표 함수 고유 스키마

데이터베이스 모니터 논리 표 3002 - 작성된 색인에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3002 - Summary Row for Index Created
     A*
     A          R QQQ3002                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
```

```
|       A            QQRDBN
|       A            QQSYS
|       A            QQJOB
|       A            QQUSER
|       A            QQJNUM
|       A            QQTHRD                    RENAME(QQI9) +
|                                                COLHDG('Thread' +
|                                                       'Identifier')
|       A            QQUCNT
|       A            QQUDEF
|       A            QQQDTN
|       A            QQQDTL
|       A            QQMATN
|       A            QQMATL
|       A            QQMATULVL                RENAME(QVP15E) +
|                                               COLHDG('Materialized' +
|                                                      'Union' +
|                                                      'Level')
|       A            QDQDTN                    RENAME(QVP15A) +
|                                                COLHDG('Decomposed' +
|                                                       'Subselect' +
|                                                       'Number')
|       A            QDQDTT                    RENAME(QVP15B) +
|                                                COLHDG('Number of' +
|                                                       'Decomposed' +
|                                                       'Subselects')
|       A            QDQDTR                    RENAME(QVP15C) +
|                                                COLHDG('Decomposed' +
|                                                       'Reason' +
|                                                       'Code')
|       A            QDQDTS                    RENAME(QVP15D) +
|                                                COLHDG('Starting' +
|                                                       'Decomposed' +
|                                                       'Subselect')
|       A            QQTLN
|       A            QQTFN
|       A            QQTMN
|       A            QQPTLN
|       A            QQPTFN
|       A            QQPTMN
|       A            QQILNM
|       A            QQIFNM
|       A            QQIMNM
|       A            QQNTNM
|       A            QQNLNM
|       A            QQSTIM
|       A            QQETIM
|       A            QQTOTR
|       A            QQRIDX
|       A            QQREST
|       A            QQFKEY
|       A            QQKSEL
|       A            QQAJN
|       A            QQEPT
|       A            QQJNP
|       A            QQJNDS                    RENAME(QQI1) +
|                                                COLHDG('Data Space' +
|                                                       'Number')
```

```
|       A            QQJNMT                  RENAME(QQC21) +
|                                            COLHDG('Join' 'Method')
|       A            QQJNTY                  RENAME(QQC22) +
|                                            COLHDG('Join' 'Type')
|       A            QQJNOP                  RENAME(QQC23) +
|                                            COLHDG('Join' 'Operator')
|       A            QQIDXK                  RENAME(QQI2) +
|                                            COLHDG('Advised' +
|                                                   'Primary' 'Keys')
|       A            QQKP
|       A            QQKPN                   RENAME(QQI3) +
|                                            COLHDG('Number' 'Key' +
|                                                   'Positioning' +
|                                                   'Columns')
|       A            QQKS
|       A            QQDSS
|       A            QQIDXA
|       A            QQRCOD
|       A            QQIDXD
|       A            QQCRTK                   RENAME(QQ1000) +
|                                             COLHDG('Key Columns' +
|                                                   'of Index' +
|                                                   'Created')
|       A            QVQTBL
|       A            QVQLIB
|       A            QVPTBL
|       A            QVPLIB
|       A            QVINAM
|       A            QVILIB
|       A            QVBNDY
|       A            QVRCNT
|       A            QVJFANO
|       A            QVFILES
|       A            QVPARPF
|       A            QVPARPL
|       A            QVPARD
|       A            QVPARU
|       A            QVPARRC
|       A            QVCTIM
|       A            QVTIXN                  RENAME(QQC101) +
|                                            COLHDG('Name of' +
|                                                   'Index' +
|                                                   'Created')
|       A            QVTIXL                  RENAME(QQC102) +
|                                            COLHDG('Library of' +
|                                                   'Index' +
|                                                   'Created')
|       A            QVTIXPZ                 RENAME(QQI4) +
|                                            COLHDG('Page Size' +
|                                                   'of Index' +
|                                                   'Created')
|       A            QVTIXRZ                 RENAME(QQI5) +
|                                            COLHDG('Row Size' +
|                                                   'of Index' +
|                                                   'Created')
|       A            QVTIXACF                RENAME(QQC14) +
|                                            COLHDG('ACS'  +
|                                                   'Table' +
```

```
|                                          'Used')
|       A            QVTIXACS              RENAME(QQC103) +
|                                          COLHDG('Alternate' +
|                                                 'Collating' +
|                                                 'Sequence' +
|                                                 'Table')
|       A            QVTIXACL              RENAME(QQC104) +
|                                          COLHDG('Alternate' +
|                                                 'Collating' +
|                                                 'Sequence' +
|                                                 'Library')
|       A            QVTIXRU               RENAME(QVC13) +
|                                          COLHDG('Index +
|                                                 'Created is' +
|                                             'Reusable')
|       A            QVTIXSP               RENAME(QVC14) +
|                                          COLHDG('Index +
|                                                 'Created is' +
|                                                 'Sparse')
|       A            QVTIXTY               RENAME(QVC1F) +
|                                          COLHDG('Type of' +
|                                                 'Index' +
|                                             'Created')
|       A            QVTIXUQ               RENAME(QVP15F) +
|                                          COLHDG('Number of' +
|                                                 'Unique Values' +
|                                                 'Index Created')
|       A            QVTIXPO               RENAME(QVC15) +
|                                          COLHDG('Permanent' +
|                                               'Index' +
|                                                 'Created')
|       A            QVTIXFX               RENAME(QVC16) +
|                                          COLHDG('Index' +
|                                                 'From' +
|                                                 'Index')
|       A            QVTIXPD               RENAME(QVP151) +
|                                          COLHDG('Parallel' +
|                                                 'Degree ' +
|                                                 'Requested')
|       A            QVTIXPU               RENAME(QVP152) +
|                                          COLHDG('Parallel' +
|                                                 'Degree ' +
|                                                 'Used')
|       A            QVTIXPRC              RENAME(QVP153) +
|                                          COLHDG('Parallel' +
|                                                 'Degree ' +
|                                                 'Limited')
|       A            QVKOA                 RENAME(QVC17) +
|                                          COLHDG('Index' +
|                                                 'Only' +
|                                                 'Access')
|       A            QVIDXM                RENAME(QVC18) +
|                                          COLHDG('Index' +
|                                                 'fits in' +
|                                                 'Memory')
|       A            QVIDXTY               RENAME(QVC1B) +
|                                          COLHDG('Index' +
|                                                 'Type')
```

```
|       A            QVIDXN                    RENAME(QQI6) +
|                                              COLHDG('Entries in' +
|                                                     'Index' +
|                                                     'Used')
|       A            QVIDXUQ                   RENAME(QQI7) +
|                                              COLHDG('Number' +
|                                                     'Unique' +
|                                                     'Values')
|       A            QVIDXPO                   RENAME(QVP158) +
|                                              COLHDG('Percent' +
|                                                     'Overflow')
|       A            QVIDXVZ                   RENAME(QVP159) +
|                                              COLHDG('Vector' +
|                                                     'Size')
|       A            QVIDXSZ                   RENAME(QQI8) +
|                                              COLHDG('Size of' +
|                                                     'Index' +
|                                                     'Used')
|       A            QVIDXPZ                   RENAME(QVP156) +
|                                              COLHDG('Page Size' +
|                                                     'Index' +
|                                                     'Used')
|       A            QQPSIZ                    RENAME(QVP154) +
|                                              COLHDG('Pool' +
|                                                     'Size')
|       A            QQPID                     RENAME(QVP155) +
|                                              COLHDG('Pool' +
|                                                     'ID')
|       A            QVTBLSZ                   RENAME(QVP157) +
|                                              COLHDG('Table' +
|                                                     'Size')
|       A            QVSKIPS                   RENAME(QVC1C) +
|                                              COLHDG('Skip' +
|                                                     'Sequential')
|       A            QVTSFLDS                  RENAME(QVC3001) +
|                                              COLHDG('Columns for' +
|                                                     'Data Space' +
|                                                     'Selection')
|       A            QVDVFLD                   RENAME(QVC1E +
|                                              COLHDG('Derived' +
|                                                     'Column' +
|                                                     'Selection')
|       A            QVDVFLDS                  RENAME(QVC3002) +
|                                              COLHDG('Columns for' +
|                                                     'Derived' +
|                                                     'Selection')
|       A            QVSKEYP                   RENAME(QVC3003) +
|                                              COLHDG('Columns Used' +
|                                                     'for Key' +
|                                                     'Positioning')
|       A            QVSKEYS                   RENAME(QVC3004) +
|                                              COLHDG('Columns Used' +
|                                                     'for Key' +
|                                                     'Selection')
|       A            QVRDTRG                   RENAME(QQC18) +
|                                              COLHDG('Read' +
```

```
                                                'Trigger')
     A          K QQJFLD
     A          S QQRID                    CMP(EQ 3002)
```

표 16. QQQ3002 - 작성된 색인에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 표에 대한 기준 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QQILNM	QQILNM	액세스하기 위해 사용되는 색인의 라이브러리명
QQIFNM	QQIFNM	액세스하기 위해 사용되는 색인명
QQIMNM	QQIMNM	액세스하기 위해 사용되는 색인의 멤버명
QQNTNM	QQNTNM	NLSS 라이브러리
QQNLNM	QQNLNM	NLSS 표
QQSTIM	QQSTIM	시작 시간소인
QQETIM	QQETIM	종료 시간소인
QQTOTR	QQTOTR	표의 총 행
QQRIDX	QQRIDX	선택된 색인의 항목 수
QQREST	QQREST	선택된 예상 행 수
QQFKEY	QQFKEY	색인 스캔 키 위치 지정을 통해 선택된 키

표 16. QQQ3002 - 작성된 색인에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQKSEL	QQKSEL	색인 스캔 키 선택을 통해 선택된 키
QQAJN	QQAJN	결합된 예상 행 수
QQEPT	QQEPT	예상 처리 시간(초 단위)
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI1	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QQIDXK	QQI2	색인 스캔 키 위치지정을 사용하는 권장된 키 열 수
QQKP	QQKP	색인 스캔 키 위치지정 • Y - 예 • N - 아니오
QQKPN	QQI3	사용된 색인에 대해 색인 스캔 키 위치지정을 사용하는 열 수
QQKS	QQKS	색인 스캔 키 선택 • Y - 예 • N - 아니오
QQDSS	QQDSS	자료 공간 선택 • Y - 예 • N - 아니오
QQIDXA	QQIDXA	권장된 색인 • Y - 예 • N - 아니오

표 16. QQQ3002 - 작성된 색인에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQRCOD	QQRCOD	이유 코드 • I1 - 행 선택 • I2 - 순서화/그룹화 • I3 - 행 선택 및 순서화/그룹화 • I4 - 내포된 루프 결합
QQIDXD	QQIDXD	권장된 색인에 대한 키 열
QQCRTK	QQ1000	작성된 색인에 대한 키 열
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름
QVPTBL	QVPTBL	기준 표, 긴 이름
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QVINAM	QVINAM	사용된 색인(또는 제한 조건)의 이름, 긴 이름
QVILIB	QVILIB	사용된 색인의 라이브러리, 긴 이름
QVBNDY	QVBNDY	I/O 또는 CPU 바인드. 가능한 값은 다음과 같습니다. • I - I/O 바인드 • C - CPU 바인드
QVRCNT	QVRCNT	고유 화면정리 카운터
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QVTIXN	QQC101	작성된 색인명
QVTIXL	QQC102	작성된 색인의 라이브러리
QVTIXPZ	QQI4	작성된 색인의 페이지 크기
QVTIXRZ	QQI5	작성된 색인의 행 크기
QVTIXACS	QQC103	작성된 색인의 대체 배열 순서 표
QVTIXACL	QQC104	작성된 색인의 대체 배열 순서 라이브러리
QVTIXRU	QVC13	작성된 색인을 재사용할 수 있음(Y/N)
QVTIXSP	QVC14	작성된 색인이 희소 색인임(Y/N)

표 16. QQQ3002 - 작성된 색인에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVTIXTY	QVC1F	작성된 색인의 유형.가능한 값은 다음과 같습니다. • B - 2진 기수 색인 • E - 코드화 벡터 색인(EVI)
QVTIXUQ	QVP15A	작성된 색인이 EVI인 경우 작성된 색인의 고유 값 수
QVTIXPO	QVC15	영구 색인이 작성됨(Y/N)
QVTIXFX	QVC16	색인의 색인(Y/N)
QVTIXPD	QVP151	요구된 병렬 정도(색인 작성)
QVTIXPU	QVP152	사용된 병렬 정도(색인 작성)
QVTIXPRC	QVP153	이유 병렬 처리가 제한되었음(색인 작성)
QVKOA	QVC17	색인 전용 액세스(Y/N)
QVIDXM	QVC18	색인이 메모리에 적합함(Y/N)
QVIDXTY	QVC1B	색인 유형. 가능한 값은 다음과 같습니다. • B - 2진 기수 색인 • C - 제한 조건(2진 기수) • E - 코드화 벡터 색인(EVI) • T - 3차 (및) 색인
QVIDXN	QQI6	색인 항목의 수, 색인 사용
QVIDXUQ	QQI7	고유 키 값의 수, 색인 사용
QVIDXPO	QVP158	퍼센트 넘침, 색인 사용
QVIDXVZ	QVP159	벡터 크기, 색인 사용
QVIDXSZ	QQI8	사용된 색인의 크기
QVIDXPZ	QVP156	색인 페이지 크기
QQPSIZ	QVP154	풀(pool) 크기
QQPID	QVP155	풀(pool) ID
QVTBLSZ	QVP157	표 크기
QVSKIPS	QVC1C	건너뛰 순차적 표 스캔(Y/N)
QVTSFLDS	QVC3001	자료 공간 선택에 사용되는 열
QVDVFLD	QVC1E	파생된 열 선택(Y/N)
QVDVFLDS	QVC3002	파생된 열 선택에 대해 사용되는 열
QVSKEYP	QVC3003	색인 스캔 키 위치지정에 사용되는 열
QVSKEYS	QVC3004	색인 스캔 키 선택에 사용되는 열
QVRDTRG	QQC18	트리거 읽기(Y/N)

데이터베이스 모니터 논리 표 3003 - 조회 정렬에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3003 - Summary Row for Query Sort
     A*
     A          R QQQ3003                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
```

```
|       A            QQTIME
|       A            QQJFLD
|       A            QQRDBN
|       A            QQSYS
|       A            QQJOB
|       A            QQUSER
|       A            QQJNUM
|       A            QQTHRD                   RENAME(QQI9) +
|                                             COLHDG('Thread' +
|                                                    'Identifier')
|       A            QQUCNT
|       A            QQUDEF
|       A            QQQDTN
|       A            QQQDTL
|       A            QQMATN
|       A            QQMATL
|       A            QQMATULVL               RENAME(QVP15E) +
|                                             COLHDG('Materialized' +
|                                                    'Union' +
|                                                    'Level')
|       A            QDQDTN                   RENAME(QVP15A) +
|                                             COLHDG('Decomposed' +
|                                                    'Subselect' +
|                                                    'Number')
|       A            QDQDTT                   RENAME(QVP15B) +
|                                             COLHDG('Number of' +
|                                                    'Decomposed' +
|                                                    'Subselects')
|       A            QDQDTR                   RENAME(QVP15C) +
|                                             COLHDG('Decomposed' +
|                                                    'Reason' +
|                                                    'Code')
|       A            QDQDTS                   RENAME(QVP15D) +
|                                             COLHDG('Starting' +
|                                                    'Decomposed' +
|                                                    'Subselect')
|       A            QQSTIM
|       A            QQETIM
|       A            QQRSS
|       A            QQSSIZ                   RENAME(QQI1) +
|                                             COLHDG('Size of' +
|                                                    'Sort' +
|                                                    'Space')
|       A            QQPSIZ                   RENAME(QQI2) +
|                                             COLHDG('Pool' +
|                                                    'Size')
|       A            QQPID                    RENAME(QQI3) +
|                                             COLHDG('Pool' +
|                                                    'ID')
|       A            QQIBUF                   RENAME(QQI4) +
|                                             COLHDG('Internal' +
|                                                    'Buffer' +
|                                                    'Length')
|       A            QQEBUF                   RENAME(QQI5) +
|                                             COLHDG('External' +
|                                                    'Buffer' +
|                                                    'Length')
|       A            QQRCOD
```

```
|       A            QQRCSUB                   RENAME(QQI7)
|       A            QVBNDY
|       A            QVRCNT
|       A            QVPARPF
|       A            QVPARPL
|       A            QVPARD
|       A            QVPARU
|       A            QVPARRC
|       A            QQEPT
|       A            QVCTIM
|       A            QQAJN
|       A            QQJNP
|       A            QQJNDS                    RENAME(QQI6) +
|                                              COLHDG('Data Space' +
|                                                     'Number')
|       A            QQJNMT                    RENAME(QQC21) +
|                                              COLHDG('Join' 'Method')
|       A            QQJNTY                    RENAME(QQC22) +
|                                              COLHDG('Join' 'Type')
|       A            QQJNOP                    RENAME(QQC23) +
|                                              COLHDG('Join' 'Operator')
|       A            QVJFANO
|       A            QVFILES
|       A          K QQJFLD
|       A          S QQRID                     CMP(EQ 3003)
```

표 17. QQQ3003 - 조회 정렬에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQSTIM	QQSTIM	시작 시간소인

표 17. QQQ3003 - 조회 정렬에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQETIM	QQETIM	종료 시간소인
QQRSS	QQRSS	선택 또는 정렬된 행 수
QQSSIZ	QQI1	정렬 간격의 크기
QQPSIZ	QQI2	풀(pool) 크기
QQPID	QQI3	풀(pool) ID
QQIBUF	QQI4	내부 정렬 버퍼 길이
QQEBUF	QQI5	외부 정렬 버퍼 길이
QQRCOD	QQRCOD	이유 코드 • F1 - 조회에는 둘 이상의 표에서 나온 그룹화 열(GROUP BY)이 들어 있거나 재순서화될 수 없는 결합 조회의 2차 표에서 나온 그룹화 열이 들어 있음 • F2 - 조회에는 둘 이상의 표에서 나온 순서화 열(ORDER BY)이 들어 있거나 재순서화될 수 없는 결합 조회의 2차 표에서 나온 순서화 열이 들어 있음. • F3 - 그룹화 및 순서화 열이 호환되지 않습니다. • F4 - DISTINCT가 조회에 대해 지정되었습니다. • F5 - UNION이 조회에 대해 지정되었습니다. • F6 - 정렬을 사용하여 구현해야 하는 조회입니다. 2000바이트를 초과하거나 120개 이상의 키 열이 순서화에 지정되었습니다. • F7 - 조회 Optimizer가 조회의 결과를 순서화하기 위해 색인보다는 정렬 사용을 선택함. • F8 - I/O 대기 시간을 최소화하기 위해 지정된 행 선택을 실행하십시오. • FC - 조회에 그룹화 필드가 들어 있고, 조회 내에 있는 최소한 한개의 실제 파일(PF)에 대한 읽기 트리거가 있습니다.
QQRCSUB	QQI7	결합에 대한 이유 부속 코드 • 51 - UNION 및 ORDER BY를 포함한 조회 • 52 - UNION ALL을 포함한 조회
QVBNDY	QVBNDY	I/O 또는 CPU 바인드. 가능한 값은 다음과 같습니다. • I - I/O 바인드 • C - CPU 바인드
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQAJN	QQAJN	결합된 예상 행 수

표 17. QQQ3003 - 조회 정렬에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수

데이터베이스 모니터 논리 표 3004 - 임시 표에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3004 - Summary Row for Temp Table
     A*
     A          R QQQ3004                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
```

```
|                                          COLHDG('Thread' +
|                                                 'Identifier')
|       A            QQUCNT
|       A            QQUDEF
|       A            QQQDTN
|       A            QQQDTL
|       A            QQMATN
|       A            QQMATL
|       A            QQMATULVL             RENAME(QVP15E) +
|                                          COLHDG('Materialized' +
|                                                 'Union' +
|                                                 'Level')
|       A            QDQDTN                 RENAME(QVP15A) +
|                                           COLHDG('Decomposed' +
|                                                  'Subselect' +
|                                                  'Number')
|       A            QDQDTT                 RENAME(QVP15B) +
|                                           COLHDG('Number of' +
|                                                  'Decomposed' +
|                                                  'Subselects')
|       A            QDQDTR                 RENAME(QVP15C) +
|                                           COLHDG('Decomposed' +
|                                                  'Reason' +
|                                                  'Code')
|       A            QDQDTS                 RENAME(QVP15D) +
|                                           COLHDG('Starting' +
|                                                  'Decomposed' +
|                                                  'Subselect')
|       A            QQTLN
|       A            QQTFN
|       A            QQTMN
|       A            QQPTLN
|       A            QQPTFN
|       A            QQPTMN
|       A            QQSTIM
|       A            QQETIM
|       A            QQDFVL                 RENAME(QQC11) +
|                                           COLHDG('Default' +
|                                                  'Values')
|       A            QQTMPR
|       A            QQRCOD
|       A            QVQTBL
|       A            QVQLIB
|       A            QVPTBL
|       A            QVPLIB
|       A            QVTTBLN                RENAME(QQC101) +
|                                           COLHDG('Temporary' +
|                                                  'Table' +
|                                                  'Name')
|       A            QVTTBLL                RENAME(QQC102) +
|                                           COLHDG('Temporary' +
|                                                  'Table' +
|                                                  'Library')
|       A            QVBNDY
|       A            QVRCNT
|       A            QVPARPF
|       A            QVPARPL
|       A            QVPARD
```

```
|       A            QVPARU
|       A            QVPARRC
|       A            QQEPT
|       A            QVCTIM
|       A            QQAJN
|       A            QQJNP
|       A            QQJNDS                  RENAME(QQI6) +
|                                            COLHDG('Data Space' +
|                                                   'Number')
|       A            QQJNMT                  RENAME(QQC21) +
|                                            COLHDG('Join' 'Method')
|       A            QQJNTY                  RENAME(QQC22) +
|                                            COLHDG('Join' 'Type')
|       A            QQJNOP                  RENAME(QQC23) +
|                                            COLHDG('Join' 'Operator')
|       A            QVJFANO
|       A            QVFILES
|       A            QVTTRSZ                 RENAME(QQI2) +
|                                            COLHDG('Row Size' +
|                                                   'Temporary' +
|                                                   'Table')
|       A            QVTTSIZ                 RENAME(QQI3) +
|                                            COLHDG('Table Size' +
|                                                   'Temporary' +
|                                                   'Table')
|       A            QVTTRST                 RENAME(QQC12) +
|                                            COLHDG('Temporary' +
|                                                   'Result')
|       A            QVTTDST                 RENAME(QQC13) +
|                                            COLHDG('Distributed' +
|                                                   'Table')
|       A            QVTTNOD                 RENAME(QVC3001) +
|                                            COLHDG('Data' +
|                                                   'Nodes')
|       A            QMATDLVL                RENAME(QQI7) +
|                                            COLHDG('Materialized' +
|                                                   'Subquery') +
|                                                   'Level')
|       A            QMATDULVL               RENAME(QQI8) +
|                                            COLHDG('Materialized' +
|                                                   'Union') +
|                                                   'Level')
|       A            QQUNVW                  RENAME(QQC14) +
|                                            COLHDG('Union' +
|                                                   'In A View')
|       A          K QQJFLD
|       A          S QQRID                   CMP(EQ 3004)
```

표 18. QQQ3004 - 임시 표에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명

표 18. QQQ3004 - 임시 표에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 표에 대한 기준 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QQSTIM	QQSTIM	시작 시간소인
QQETIM	QQETIM	종료 시간소인
QQDFVL	QQC11	디폴트 값이 임시 표에 표시됩니다. • Y - 예 • N - 아니오
QQTMPR	QQTMPR	임시 표의 행 수

표 18. QQQ3004 - 임시 표에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQRCOD	QQRCOD	이유 코드. 가능한 값은 다음과 같습니다. • F1 - 조회에는 둘 이상의 표에서 나온 그룹화 열(GROUP BY)이 들어 있거나 재순서화될 수 없는 결합 조회의 2차 표에서 나온 그룹화 열이 들어 있음 • F2 - 조회에는 둘 이상의 표에서 나온 순서화 열(ORDER BY)이 들어 있거나 재순서화될 수 없는 결합 조회의 2차 표에서 나온 순서화 열이 들어 있음. • F3 - 그룹화 및 순서화 열이 호환되지 않습니다. • F4 - DISTINCT가 조회에 대해 지정되었습니다. • F5 - UNION이 조회에 대해 지정되었습니다. • F6 - 정렬을 사용하여 구현해야 하는 조회입니다. 2000바이트를 초과하거나 120개 이상의 키 열이 순서화에 지정되었습니다. • F7 - 조회 Optimizer는 조회의 결과를 순서화하기로 선택하였습니다. • F8 - I/O 대기 시간을 최소화하기 위해 지정된 행 선택을 실행하십시오. • F9 - 조회 Optimizer가 색인이 아닌 해시 알고리즘을 사용하여 그룹화를 실행하기로 선택하였습니다. • FA - 조회에 임시 표를 요구하는 결합 조건이 들어 있습니다. • FB - 조회 Optimizer가 조회로 특정 상관 그룹을 구현하기 위해 런타임 임시 파일을 작성합니다. • FC - 조회에 그룹화 필드가 들어 있고, 조회 내에 있는 최소한 한개의 실제 파일(PF)에 대한 읽기 트리거가 있습니다. • FD - 조회 Optimizer는 정적 커서 요청에 대한 런타임 임시 파일을 작성합니다. • H1 - 표가 결합 논리 파일이며 해당 결합 유형이 조회에 지정된 결합 유형과 일치하지 않습니다. • H2 - 논리 표에 대해 지정된 형식이 둘 이상의 기준 표를 참조합니다. • H3 - 표가 임시 표에 SQL 보기의 결과가 포함되어야 하는 복합 SQL 보기입니다. • H4 - 갱신 허용 조회의 경우, 부속 선택이 갱신 중인 열 중 하나에 대응하는 이 표에 있는 열을 참조합니다. • H5 - 갱신 허용 조회의 경우, 부속 선택이 갱신 중인 표에 기초한 SQL 보기를 참조합니다. • H6 - 삭제 허용 조회의 경우, 부속 선택이 삭제될 행의 표, SQL 보기 또는 삭제될 행의 표에 기초한 색인을 참조합니다. • H7 - 사용자 정의 표 함수가 구체화되었습니다.
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름
QVPTBL	QVPTBL	기준 표, 긴 이름

표 18. QQQ3004 - 임시 표에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QVTTBLN	QQC101	임시 표 이름
QVTTBLL	QQC102	임시 표 라이브러리
QVBNDY	QVBNDY	I/O 또는 CPU 바인드. 가능한 값은 다음과 같습니다. • I - I/O 바인드 • C - CPU 바인드
QVRCNT	QVRCNT	고유 화면정리 카운터
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(Y/N)
QVPARD	QVPARD	요구된 병렬 정도
QVPARU	QVPARU	사용된 병렬 정도
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQAJN	QQAJN	결합된 예상 행 수
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합

표 18. QQQ3004 - 임시 표에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVTTRSZ	QQI2	임시 표의 행 크기(바이트)
QVTTSIZ	QQI3	임시 표의 크기(바이트)
QVTTRST	QQC12	조회 결과가 들어 있는 임시 결과 표(Y/N)
QVTTDST	QQC13	분배 표(Y/N)
QVTTNOD	QVC3001	임시 표의 자료 노드
QMATDLVL	QQI7	구체화된 부속 조회 QDT 레벨
QMATDULVL	QQI8	구체화된 결합 QDT 레벨
QQUNVW	QQC14	보기에서 결합(Y/N)

데이터베이스 모니터 논리 표 3005 - 잠긴 표에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3005 - Summary Row for Table Locked
     A*
     A          R QQQ3005                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                RENAME(QVP15E) +
                                           COLHDG('Materialized' +
                                                  'Union' +
                                                   'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
```

```
                                                 'Subselect' +
                                                 'Number')
     A            QDQDTT                     RENAME(QVP15B) +
                                             COLHDG('Number of' +
                                                 'Decomposed' +
                                                 'Subselects')
     A            QDQDTR                     RENAME(QVP15C) +
                                             COLHDG('Decomposed' +
                                                 'Reason' +
                                                 'Code')
     A            QDQDTS                     RENAME(QVP15D) +
                                             COLHDG('Starting' +
                                                 'Decomposed' +
                                                 'Subselect')
     A            QQTLN
     A            QQTFN
     A            QQTMN
     A            QQPTLN
     A            QQPTFN
     A            QQPTMN
     A            QQLCKF                     RENAME(QQC11) +
                                             COLHDG('Lock' +
                                                 'Indicator')
     A            QQULCK                     RENAME(QQC12) +
                                             COLHDG('Unlock' +
                                                 'Request')
     A            QQRCOD
     A            QVQTBL
     A            QVQLIB
     A            QVPTBL
     A            QVPLIB
     A            QQJNP
     A            QQJNDS                     RENAME(QQI6) +
                                             COLHDG('Data Space' +
                                                 'Number')
     A            QQJNMT                     RENAME(QQC21) +
                                             COLHDG('Join' 'Method')
     A            QQJNTY                     RENAME(QQC22) +
                                             COLHDG('Join' 'Type')
     A            QQJNOP                     RENAME(QQC23) +
                                             COLHDG('Join' 'Operator')
     A            QVJFANO
     A            QVFILES
     A            QVRCNT
     A          K QQJFLD
     A          S QQRID                      CMP(EQ 3005)
```

표 19. QQQ3005 - 잠긴 표에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명

표 19. QQQ3005 - 잠긴 표에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 표에 대한 기준 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QQLCKF	QQC11	잠금 인디케이터 • Y - 예 • N - 아니오
QQULCK	QQC12	잠금 해제 요구 • Y - 예 • N - 아니오
QQRCOD	QQRCOD	이유 코드 • L1 - 보유 잠금을 사용하여 *ALL 또는 *CS에 대한 UNION • L2 - 보유 잠금을 사용하여 *ALL 또는 *CS에 대한 DISTINT • L3 - 보유 잠금을 사용하여 *ALL 또는 *CS에 대한 중복 키 없음 • L4 - 보유 잠금을 사용하여 *ALL 또는 *CS에 대해 필요한 임시 • L5 - 보유 잠금을 사용하여 *ALL 또는 *CS에 대한 시스템 표 • L6 - 보유 잠금을 사용하여 *ALL 또는 *CS에 대해 Orderby > 2000바이트 • L9 - 알 수 없음 • LA - 보유 잠금을 사용하여 *ALL 또는 *CS에 대해 사용자 정의된 표 함수
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름

표 19. QQQ3005 - 잠긴 표에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVPTBL	QVPTBL	기준 표, 긴 이름
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVRCNT	QVRCNT	고유 화면정리 카운터

데이터베이스 모니터 논리 표 3006 - 액세스 계획 리빌드에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3006 - Summary Row for Access Plan Rebuilt
     A*
     A          R QQQ3006                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
```

```
|       A            QQJOB
|       A            QQUSER
|       A            QQJNUM
|       A            QQTHRD                  RENAME(QQI9) +
|                                            COLHDG('Thread' +
|                                                   'Identifier')
|       A            QQUCNT
|       A            QQUDEF
|       A            QQQDTN
|       A            QQQDTL
|       A            QQMATN
|       A            QQMATL
|       A            QQMATULVL               RENAME(QVP15E) +
|                                            COLHDG('Materialized' +
|                                                   'Union' +
|                                                   'Level')
|       A            QDQDTN                  RENAME(QVP15A) +
|                                            COLHDG('Decomposed' +
|                                                   'Subselect' +
|                                                   'Number')
|       A            QDQDTT                  RENAME(QVP15B) +
|                                            COLHDG('Number of' +
|                                                   'Decomposed' +
|                                                   'Subselects')
|       A            QDQDTR                  RENAME(QVP15C) +
|                                            COLHDG('Decomposed' +
|                                                   'Reason' +
|                                                   'Code')
|       A            QDQDTS                  RENAME(QVP15D) +
|                                            COLHDG('Starting' +
|                                                   'Decomposed' +
|                                                   'Subselect')
|       A            QQINLN
|       A            QQINFN
|       A            QQRCOD
|       A            QVSUBRC                 RENAME(QQC21) +
|                                            COLHDG('Subtype' +
|                                                   'Reason' +
|                                                   'Code')
|       A            QVRCNT
|       A            QVRPTS                  RENAME(QQTIM1) +
|                                            COLHDG('Timestamp' +
|                                                   'Last' +
|                                                   'Rebuild')
|       A            QRQDOPT                 RENAME(QQC11) +
|                                            COLHDG('Access' +
|                                                   'Plan' +
|                                                   'Reoptimized')
|      A             QRCODES                 RENAME(QVC22) +
|                                            COLHDG('Previous' +
|                                                   'Reason' +
|                                                   'Code')
|      A             QVSUBRCS                RENAME(QVC23) +
|                                            COLHDG('Previous' +
|                                                   'Reason' +
|                                                   'Subcode')
|       A          K QQJFLD
|       A          S QQRID                   CMP(EQ 3006)
```

표 20. QQQ3006 - 액세스 계획 리빌드에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호

표 20. QQQ3006 - 액세스 계획 리빌드에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQRCOD	QQRCOD	액세스 계획이 리빌드된 이유 코드 • A1 - 표 또는 멤버가 액세스 계획이 최종 빌드되었을 때 참조된 표 또는 멤버와 동일한 오브젝트가 아닙니다. 오브젝트가 다른 이유는 다음과 같습니다. – 오브젝트가 삭제되고 리빌드되었음. – 오브젝트가 저장되고 복원되었음. – 라이브러리 리스트가 변경되었음. – 오브젝트의 이름이 변경되었음. – 오브젝트가 이동되었음. – 오브젝트가 다른 오브젝트로 대체되었음. – 이번이 조회가 들어 있는 오브젝트가 복원된 이후 해당 조회를 처음으로 실행하는 경우임. • A2 - 액세스 계획이 재사용 가능한 ODP(열린 자료 경로)를 사용하도록 빌드되었는데 Optimizer가 이 호출에 대해 재사용할 수 없는 ODP를 사용하기로 선택하였습니다. • A3 - 액세스 계획이 재사용할 수 없는 ODP(열린 자료 경로)를 사용하도록 빌드되었으며 Optimizer가 이 호출에 대해 재사용 가능한 ODP를 사용하기로 선택하였습니다. • A4 - 표에서 행 수가 액세스 계획이 최종 빌드된 이후로 10% 이상 변경되었습니다. • A5 - 조회 표 중 하나에 대해 신규 색인이 존재합니다. • A6 - 이 액세스 계획에 사용되었던 색인이 더 이상 존재하기 않거나 더 이상 유효하지 않습니다. • A7 - 시스템 프로그래밍 변경으로 인해 OS/400 조회에 리빌드된 액세스 계획이 필요합니다. • A8 - 현재 작업의 CCSID가 액세스 계획을 최종 작성한 작업의 CCSID와 다릅니다. • A9 - 다음 중 하나 이상의 값이 현재 작업에 대해 이 액세스 계획을 최종 작성했을 때의 작업에 대한 값과 다릅니다. – 날짜 형식 – 날짜 분리자 – 시간 형식 – 시간 분리자 • AA - 지정된 정렬 순서표가 이 액세스 계획이 작성되었던 때 사용되었던 정렬 순서표와 다릅니다. • AB - 기억장치 풀이 변경되었거나 CHGQRYA 명령의 DEGREE 매개변수가 변경되었습니다. • AC - 시스템 피처 DB2 복수 시스템이 설치 또는 제거되었습니다. • AD - 정도 조회 속성 값이 변경되었습니다. • AE - 보기가 상위 레벨 언어에 의해 열리거나 구체화되는 중입니다. • AF - 사용자 정의 유형 또는 사용자 정의 기능이 액세스 계획에서 참조된 유형 또는 기능과 동일한 오브젝트가 아니거나 SQL 경로가 액세스 계획이 빌드될 때와 동일하지 않습니다. • B0 - 지정된 옵션이 조회 옵션 파일의 결과로서 변경되었습니다. • B1 - 액세스 계획이 현재 작업과 다른 확약 제어 레벨로 생성되었습니다. • B2 - 액세스 계획이 이전 액세스 계획과 다른 정적 커서 응답 설정 크기로 생성되었습니다.
QVSUBRC	QQC21	액세스 계획 리빌드 이유 코드가 A7인 경우, 이 2바이트 16진 값은 A7에 대한 어떤 특정 이유가 리빌드를 강제했는지 식별합니다.
QVRCNT	QVRCNT	고유 화면정리 카운터
QVRPTS	QQTIM1	최종 액세스 계획 리빌드의 시간 소인
QRQDOPT	QQC11	이 계획을 위해 필수인 최적화 • Y - 예. 계획이 확실히 최적화되었습니다. • N - 아니오. 계획이 REOPTIMIZE_ACCESS_PLAN 매개변수 값의 QAQQINI 옵션때문에 다시 최적화되지 않았습니다.

표 20. QQQ3006 - 액세스 계획 리빌드에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QRCODES	QVC22	이전 이유 코드
QVSUBRCS	QVC23	이전 이유 부속 코드

데이터베이스 모니터 논리 표 3007 - Optimizer 시간 종료에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3007 - Summary Row for Optimizer Timed Out
     A*
     A          R QQQ3007                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
                                            COLHDG('Decomposed' +
                                                   'Reason' +
                                                   'Code')
     A            QDQDTS                    RENAME(QVP15D) +
                                            COLHDG('Starting' +
                                                   'Decomposed' +
                                                   'Subselect')
     A            QQTLN
     A            QQTFN
     A            QQTMN
     A            QQPTLN
     A            QQPTFN
     A            QQPTMN
     A            QQIDXN                    RENAME(QQ1000) +
                                            COLHDG('Index' +
```

```
|                                                 'Names')
|       A           QQTOUT                    RENAME(QQC11) +
|                                               COLHDG('Optimizer' +
|                                                      'Timed Out')
|       A           QQISRN                    RENAME(QQC301) +
|                                               COLHDG('Index' +
|                                                      'Reason' +
|       A           QVQTBL
|       A           QVQLIB
|       A           QVPTBL
|       A           QVPLIB
|       A           QQJNP
|       A           QQJNDS                    RENAME(QQI6) +
|                                               COLHDG('Data Space' +
|                                                      'Number')
|       A           QQJNMT                    RENAME(QQC21) +
|                                               COLHDG('Join' 'Method')
|       A           QQJNTY                    RENAME(QQC22) +
|                                               COLHDG('Join' 'Type')
|       A           QQJNOP                    RENAME(QQC23) +
|                                               COLHDG('Join' 'Operator')
|       A           QVJFANO
|       A           QVFILES
|       A           QVRCNT
|       A         K QQJFLD
|       A         S QQRID                     CMP(EQ 3007)
```

표 21. QQQ3007 - Optimizer 시간 종료에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호

표 21. QQQ3007 - Optimizer 시간 종료에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 표에 대한 기준 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QQIDXN	QQ1000	색인명
QQTOUT	QQC11	Optimizer 시간 종료 • Y - 예 • N - 아니오
QQISRN	QQC301	시간 종료된 색인에 의해 사용된 고유 이유 코드 리스트(각각의 색인은 이와 연관된 대응하는 사용 코드를 가짐)
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름
QVPTBL	QVPTBL	기준 표, 긴 이름
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 번호
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적

표 21. QQQ3007 - Optimizer 시간 종료에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVRCNT	QVRCNT	고유 화면정리 카운터

데이터베이스 모니터 논리 표 3008 - 부속 조회 처리에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3008 - Summary Row for Subquery Processing
     A*
     A          R QQQ3008                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
                                            COLHDG('Decomposed' +
                                                   'Reason' +
                                                   'Code')
     A            QDQDTS                    RENAME(QVP15D) +
```

```
                                        COLHDG('Starting' +
                                               'Decomposed' +
                                               'Subselect')
      A             QQORGQ              RENAME(QQI1) +
                                        COLHDG('Original' +
                                               'Number' +
                                               'of QDTs')
      A             QQMRGQ              RENAME(QQI2) +
                                        COLHDG('Number' +
                                               'of QDTs' +
                                               'Merged')
      A             QQFNLQ              RENAME(QQI3) +
                                        COLHDG('Final' +
                                               'Number' +
                                               'of QDTs')
      A               QVRCNT
      A           K QQJFLD
      A           S QQRID               CMP(EQ 3008)
```

표 22. QQQ3008 - 부속 조회 처리에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQORGQ	QQI1	원래 QDT 수
QQMRGQ	QQI2	병합된 QDT의 수
QQFNLQ	QQI3	최종 QDT 수
QVRCNT	QVRCNT	고유 화면정리 카운터

데이터베이스 모니터 논리 표 3010 - 호스트 변수 & ODP 구현에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3010 - Summary for HostVar & ODP Implementation
     A*
     A          R QQQ3010                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQRCNT                    RENAME(QQI5) +
                                            COLHDG('Refresh' +
                                                   'Counter')
     A            QQUDEF
     A            QQODPI                    RENAME(QQC11) +
                                            COLHDG('ODP' +
                                                   'Implementation')
     A            QQHVI                     RENAME(QQC12) +
                                            COLHDG('Host Variable' +
                                                   'Implementation')
     A            QQHVAR                    RENAME(QQ1000) +
                                            COLHDG('Host Variable' +
                                                   'Values')
     A          K QQJFLD
     A          S QQRID                     CMP(EQ 3010)
```

표 23. QQQ3010 - 호스트 변수 & ODP 구현에 대한 요약

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQRCNT	QQI5	고유 화면정리 카운터
QQUDEF	QQUDEF	사용자 정의 열

표 23. QQQ3010 - 호스트 변수 & ODP 구현에 대한 요약 (계속)

논리 열 이름	실제 열 이름	설명
QQODPI	QQC11	ODP 구현 • R - 재사용 가능한 ODP • N - 재사용할 수 없는 ODP • ' ' - 사용되지 않는 열
QQHVI	QQC12	호스트 변수 구현 • I - 인터페이스 제공 값(ISV) • V - 상수로 취급되는 호스트 변수(V2) • U - 표 관리 행 위치지정(UP)
QQHVAR	QQ1000	호스트 변수 값

데이터베이스 모니터 논리 표 3014 - 일반 QQ 정보에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3014 - Summary Row for Generic QQ Information
     A*
     A          R QQQ3014                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
                                            COLHDG('Decomposed' +
                                                   'Reason' +
                                                   'Code')
```

```
|       A           QDQDTS                  RENAME(QVP15D) +
|                                           COLHDG('Starting' +
|                                                  'Decomposed' +
|                                                  'Subselect')
|       A           QQREST
|       A           QQEPT
|       A           QQQTIM                  RENAME(QQI1) +
|                                           COLHDG('ODP' +
|                                                  'Open' 'Time')
|       A           QQORDG
|       A           QQGRPG
|       A           QQJNG
|       A           QQJNTY                  RENAME(QQC22) +
|                                           COLHDG('Join' +
|                                                  'Type')
|       A           QQUNIN
|       A           QQSUBQ
|       A           QQSSUB                  RENAME(QWC1F)
|                                           COLHDG('Scalar' +
|                                                  'Subselects)
|       A           QQHSTV
|       A           QQRCDS
|       A           QQGVNE                  RENAME(QQC11) +
|                                           COLHDG('Query' +
|                                                  'Governor' +
|                                                  'Enabled')
|       A           QQGVNS                  RENAME(QQC12) +
|                                           COLHDG('Stopped' +
|                                                  'by Query' +
|                                                  'Governor')
|       A           QQOPID                  RENAME(QQC101) +
|                                           COLHDG('Query' +
|                                                  'Open ID')
|       A           QQINLN                  RENAME(QQC102) +
|                                           COLHDG('Query' +
|                                                  'Options' +
|                                                  'Library')
|       A           QQINFN                  RENAME(QQC103) +
|                                           COLHDG('Query' +
|                                                  'Options' +
|                                                  'File')
|       A           QQEE                    RENAME(QQC13) +
|                                           COLHDG('Early' +
|                                                  'Exit' +
|                                                  'Indicator')
|       A           QVRCNT
|       A           QVOPTIM                 RENAME(QQI5) +
|                                           COLHDG('Optimization' +
|                                                  'Time')
|       A           QVAPRT                  RENAME(QQTIM1) +
|                                           COLHDG('Access Plan' +
|                                                  'Rebuild'
|                                                  'Timestamp')
|       A           QVOBYIM                 RENAME(QVC11) +
|                                           COLHDG('Ordering' +
|                                                  'Implementation')
|       A           QVGBYIM                 RENAME(QVC12) +
|                                           COLHDG('Grouping' +
```

```
                                          'Implementation')
     A            QVJONIM                 RENAME(QVC13) +
                                          COLHDG('Join' +
                                          'Implementation')
     A            QVDIST                  RENAME(QVC14) +
                                          COLHDG('Distinct' +
                                          'Query')
     A            QVDSTRB                 RENAME(QVC15) +
                                          COLHDG('Distributed' +
                                          'Query')
     A            QVDSTND                 RENAME(QVC3001) +
                                          COLHDG('Distributed' +
                                          'Nodes')
     A            QVNLSST                 RENAME(QVC105) +
                                          COLHDG('Sort' +
                                          'Sequence' +
                                          'Table')
     A            QVNLSSL                 RENAME(QVC106) +
                                          COLHDG('Sort' +
                                          'Sequence' +
                                          'Library')
     A            QVALWCY                 RENAME(QVC16) +
                                          COLHDG('ALWCPYDTA' +
                                          'Setting')
     A            QVVAPRC                 RENAME(QVC21) +
                                          COLHDG('Access Plan' +
                                          'Rebuilt' +
                                          'Code')
     A            QVVAPSC                 RENAME(QVC22) +
                                          COLHDG('Access Plan' +
                                          'Rebuilt' +
                                          'Subcode')
     A            QVIMPLN                 RENAME(QVC3002) +
                                          COLHDG('Implementation' +
                                          'Summary')
     A            QVUNIONL                RENAME(QWC16) +
                                          COLHDG('Last' +
                                          'Part of' +
                                          'Union')
     A            DCMPFNLBLT              RENAME(QQC14) +
                                          COLHDG('Decomposed' +
                                          'Final Cursor' +
                                          'was Built')
     A            DCMPFNLTMP              RENAME(QQC15) +
                                          COLHDG('This is' +
                                          'Decomposed' +
                                          'Final Cursor')
     A            QQPSIZ                  RENAME(QVP154) +
                                          COLHDG('Pool' +
                                          'Size')
     A            QQPID                   RENAME(QVP155) +
                                          COLHDG('Pool' +
                                          'ID')
     A*
     A* CHGQRYA or INI environment attributes used during execution of query
     A*
     A            QVMAXT                  RENAME(QQI2) +
                                          COLHDG('Query' +
```

```
|                                          'Time' +
|                                          'Limit')
|       A            QVPARA                RENAME(QVC81) +
|                                          COLHDG('Specified' +
|                                                 'Parallel' +
|                                                 'Option')
|       A            QVTASKN               RENAME(QQI3) +
|                                          COLHDG('Mamimum' +
|                                                 'Number of' +
|                                                 'Tasks')
|       A            QVAPLYR               RENAME(QVC17) +
|                                          COLHDG('Apply' +
|                                                 'CHGQRYA' +
|                                                 'Remotely')
|       A            QVASYNC               RENAME(QVC82) +
|                                          COLHDG('Asynchronous' +
|                                                 'Remote' +
|                                                 'Job Usage')
|       A            QVFRCJO               RENAME(QVC18) +
|                                          COLHDG('Join' +
|                                                 'Order' +
|                                                 'Forced')
|       A            QVDMSGS               RENAME(QVC19) +
|                                          COLHDG('Display' +
|                                                 'DEBUG' +
|                                                 'Messages')
|       A            QVPMCNV               RENAME(QVC1A) +
|                                          COLHDG('Parameter' +
|                                                 'Marker' +
|                                                 'Conversion')
|       A            QVUDFTL               RENAME(QQI4) +
|                                          COLHDG('UDF' +
|                                                 'Time' +
|                                                 'Limit')
|       A            QVOLMTS               RENAME(QVC1283) +
|                                          COLHDG('Query' +
|                                                 'Optimizer' +
|                                                 'Limitations')
|       A            QVREOPT               RENAME(QVC1E) +
|                                          COLHDG('Reoptimize' +
|                                                 'Access' 'Plan')
|       A            QVOPALL               RENAME(QVC87) +
|                                          COLHDG('Optimize' +
|                                                 'All' 'Indexes')
|       A            QVDFQDTF              RENAME(QQC14) +
|                                          COLHDG('Final' +
|                                                 'Decomposed' +
|                                                 'QDT Built')
|       A            QVDFQDT               RENAME(QQC15) +
|                                          COLHDG('Final' +
|                                                 'Decomposed' +
|                                                 'QDT')
|       A            QVRDTRG               RENAME(QQC18) +
|                                          COLHDG('Read' +
|                                                 'Trigger')
|       A            QVSTRJN               RENAME(QQC81) +
|                                          COLHDG('Star' +
|                                                 'Join')
```

```
A            OPTGOAL                 RENAME(QVC23) +
                                     COLHDG('Optimization' +
                                            'Goal')
A            DIAGLIKE                RENAME(QVC24) +
                                     COLHDG('Visual' +
                                            'Explain' +
                                            'Diagram')
A            UNIONVIEW               RENAME(QQC23) +
                                     COLHDG('Union' +
                                            'in a' +
                                            'View')
A            SUBQTYPE                RENAME(QQC21)
                                     COLHDG('Type of' +
                                            'Subselect)
A          K QQJFLD
A          S QQRID                   CMP(EQ 3014)
```

표 24. QQQ3014 - 일반 QQ 정보에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQREST	QQREST	선택된 예상 행 수
QQEPT	QQEPT	예상 처리 시간(초 단위)
QQQTIM	QQI1	커서를 여는 데 소비된 시간(밀리초 단위)
QQORDG	QQORDG	순서화(Y/N)
QQGRPG	QQGRPG	그룹화(Y/N)
QQJNG	QQJNG	결합 조회(Y/N)

표 24. QQQ3014 - 일반 QQ 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQUNIN	QQUNIN	통합 조회(Y/N)
QQSUBQ	QQSUBQ	부속 조회(Y/N)
QQSSUB	QWC1F	스칼라 부속 선택(Y/N)
QQHSTV	QQHSTV	호스트 변수(Y/N)
QQRCDS	QQRCDS	행 선택(Y/N)
QQGVNE	QQC11	조회 감독자 사용 가능(Y/N)
QQGVNS	QQC12	조회 감독자가 조회를 중단함(Y/N)
QQOPID	QQC101	조회 열기 ID
QVINLN	QQC102	조회 옵션 라이브러리명
QVINFN	QQC103	조회 옵션 파일명
QQEE	QQC13	조회 초기 나감 값
QVRCNT	QVRCNT	고유 화면정리 카운터
QVOPTIM	QQI5	Optimizer에 소비된 시간(밀리초 단위)
QVAPRT	QQTIM1	액세스 계획 리빌드 시간소인, 액세스 계획이 리빌드되었던 최종 시간
QVOBYIM	QVC11	순서화 구현. 가능한 값은 다음과 같습니다. • I - 색인 • S - 정렬
QVGBYIM	QVC12	그룹화 구현. 가능한 값은 다음과 같습니다. • I - 색인 • H - 해시 그룹화
QVJONIM	QVC13	결합 구현. 가능한 값은 다음과 같습니다. • N - 내포된 루프 결합 • H - 해시 결합 • C - 내포된 루프와 해시의 조합
QVDIST	QVC14	고유 조회(Y/N)
QVDSTRB	QVC15	분배 조회(Y/N)
QVDSTND	QVC3001	분배 노드
QVNLSST	QVC105	정렬 순서 표
QVNLSSL	QVC106	정렬 순서 라이브러리
QVALWC	QVC16	ALWCPYDTA 설정
QVVAPRC	QVC21	액세스 계획이 리빌드된 이유 코드
QVVAPSC	QVC22	액세스 계획이 리빌드된 부속 코드
QVIMPLN	QVC3002	조회 구현 요약. 조회되고 있는 각각의 표에 대해 사용된 색인명 및 자료 공간 수를 표시합니다.
QVUNIONL	QWC16	UNION의 마지막 파트(마지막 QDT)(Y/N)

표 24. QQQ3014 - 일반 QQ 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
DCMPFNLBLT	QWC14	분해된 최종 임시 커서가 빌드되었습니다(Y/N).
DCMPFNLTMP	QWC15	분해된 최종 임시 커서(최종 임시 QDT)입니다. (Y/N)
QVMAXT	QQI2	조회 시간 제한
QVPARA	QVC81	병렬 정도 • *SAME - 현재 설정을 변경하지 않음. • *NONE - 병렬 처리가 허용되지 않음. • *I/O - I/O 처리에 임의 수의 타스크를 사용할 수 있음. SMP 병렬 처리는 허용되지 않습니다. • *OPTIMIZE - Optimizer가 I/O 또는 SMP 병렬 처리에 사용할 타스크 수를 선택. • *MAX - Optimizer가 I/O 또는 SMP 병렬 처리 사용을 선택. • *SYSVAL - 조회를 처리하기 위해 현재 시스템 값을 사용. • *ANY - *I/O와 같은 의미를 가짐. • *NBRTASKS - SMP 병렬 처리를 위한 타스크 수가 열 QVTASKN에 지정됨.
QVTASKN	QQI3	최대 타스크 수
QVAPLYR	QVC17	원격으로 CHGQRYA 적용(Y/N)
QVASYNC	QVC82	비동기 작업 사용 • *SAME - 현재 설정을 변경하지 않음. • *DIST - 비동기 작업을 분배 표가 있는 조회에 사용할 수 있음 • *LOCAL - 비동기 작업을 로컬 표가 있는 조회에만 사용할 수 있음 • *ANY - 비동기 작업을 모든 데이터베이스 조회에 대해 사용할 수 있음 • *NONE - 비동기 작업이 허용되지 않음
QVFRCJO	QVC18	강제 결합 순서(Y/N)
QVDMSGS	QVC19	인쇄 디버그 메세지(Y/N)
QVPMCNV	QVC1A	매개변수 마커 변환(Y/N)
QVUDFTL	QQI4	사용자 정의 기능 시간 제한
QVOLMTS	QVC1281	Optimizer 제한사항. 가능한 값은 다음과 같습니다. • *PERCENT 다음에 퍼센트 값이 있는 2바이트의 정수가 옴. • *MAX_NUMBER_OF_RECORDS 다음에 최대 행 수를 나타내는 정수 값이 옴.
QVREOPT	QVC1E	요구된 액세스 계획 다시 최적화. 가능한 값은 다음과 같습니다. • 'O' - 액세스 계획이 전적으로 요구될 때에만 다시 최적화합니다. 주관적인 이유로 다시 최적화하지 마십시오. • 'Y' - 예. 액세스 계획을 다시 최적화하도록 합니다. • 'N' - 아니오. Optimizer가 필요로 하지 않는 이상, 액세스 계획을 다시 최적화하지 않습니다. 주관적인 이유로 다시 최적화할 수는 있습니다.

표 24. QQQ3014 - 일반 QQ 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVOPALL	QVC87	요구된 모든 색인을 최적화 • *SAME - 현재 설정을 변경하지 않음. • *YES - 모든 색인을 검사함 • *NO - Optimizer가 시간 종료를 허용함 • *TIMEOUT - Optimizer가 시간 종료를 강제함
QVDFQDTF	QQC14	최종 분해된 QDT 빌드 인디케이터(Y/N)
QVDFQDT	QQC15	최종 분해된 QDT 인디케이터(Y/N)
QVRDTRG	QQC18	트리거 읽기를 포함하는 파일 중 하나(Y/N)
QVSTRJN	QQC81	성형 결합 최적화가 요구되었습니다. • *NO - 성형 결합 최적화가 수행되지 않습니다. • *COST - Optimizer가 성형 결합 최적화를 위해 EVI를 사용할 수 있는지 여부를 판별합니다. • *FORCE - Optimizer가 성형 결합 최적화에 사용할 수 있는 EVI를 추가합니다.
OPTGOAL	QVC23	바이트 1 = 최적화 목표. 가능한 값은 다음과 같습니다. • 'F' - 첫 번째 I/O, 행으로 가득 찬 첫 번째 화면을 리턴하는 조회를 가능한 빨리 최적화합니다. • 'A' - 모든 I/O, 모든 행을 리턴하는 조회를 가능한 빨리 최적화합니다.
DIAGLIKE	QVC24	바이트 1 = Visual Explain 다이어그램 유형. 가능한 값은 다음과 같습니다. • 'D' - 상세 • 'B' - 기본 바이트 2 - LIKE 중복 시프트를 무시합니다. 가능한 값은 다음과 같습니다. • 'O' - 최적화, 조회 Optimizer가 무시할 중복 시프트를 판별합니다. • 'A' - 모두, 모든 중복 시프트가 무시됩니다.
UNIONVIEW	QQC23	바이트 1 = 이 QDT는 보기 내에 포함된 UNION의 일부입니다. (Y/N) 바이트 2 = 이 QDT는 보기 내에 포함된 UNION의 마지막 부속 선택입니다. (Y/N)
SUBQTYPE	QQC21	부속 선택의 유형. 가능한 값은 다음과 같습니다. • 'SS' - 스칼라 부속 선택 • 'SU' - Set 부속 선택으로 갱신 • 'SQ' - 부속 조회

데이터베이스 모니터 논리 표 3015 - 통계 정보에 대한 요약

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3015 - Summary Row for Statistics Information
     A*
     A          R QQQ3015                   PTABLE(*CURLIB/QAQQDBMN)
```

```
|       A            QQRID
|       A            QQTIME
|       A            QQJFLD
|       A            QQRDBN
|       A            QQSYS
|       A            QQJOB
|       A            QQUSER
|       A            QQJNUM
|       A            QQTHRD                  RENAME(QQI9) +
|                                              COLHDG('Thread' +
|                                                     'Identifier')
|       A            QQUCNT
|       A            QQUDEF
|       A            QQQDTN
|       A            QQQDTL
|       A            QQMATN
|       A            QQMATL
|       A            QQMATULVL               RENAME(QVP15E) +
|                                              COLHDG('Materialized' +
|                                                    'Union' +
|                                                    'Level')
|       A            QDQDTN                  RENAME(QVP15A) +
|                                              COLHDG('Decomposed' +
|                                                     'Subselect' +
|                                                     'Number')
|       A            QDQDTT                  RENAME(QVP15B) +
|                                              COLHDG('Number of' +
|                                                     'Decomposed' +
|                                                     'Subselects')
|       A            QDQDTR                  RENAME(QVP15C) +
|                                              COLHDG('Decomposed' +
|                                                     'Reason' +
|                                                     'Code')
|       A            QDQDTS                  RENAME(QVP15D) +
|                                              COLHDG('Starting' +
|                                                     'Decomposed' +
|                                                     'Subselect')
|       A            QQTLN
|       A            QQTFN
|       A            QQTMN
|       A            QQPTFN
|       A            QQPTMN
|       A            QVQTBL
|       A            QVQLIB
|       A            QVPTBL
|       A            QVPLIB
|       A            QQNTNM
|       A            QQNLNM
|       A            QVSTATUS                RENAME(QQC11) +
|                                              COLHDG('Statistic' +
|                                                     'Status')
|       A            QVSTATIMP               RENAME(QQi2) +
|                                              COLHDG('Statistic' +
|                                                     'Importance')
|       A            QVSTATCOL               RENAME(QQ1000) +
|                                              COLHDG('Column' +
|                                                     'Names')
|       A            QVSTATID                RENAME(QVC1000) +
```

```
                                        COLHDG('Statistic' +
                                               'Identifier')
     A          K QQJFLD
     A          S QQRID                  CMP(EQ 3015)
```

표 25. QQQ3015 - 통계 정보에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQTLN	QQTLN	조회된 표 라이브러리
QQTFN	QQTFN	조회된 표 이름
QQTMN	QQTMN	조회된 표 멤버명
QQPTLN	QQPTLN	기준 표의 라이브러리명
QQPTFN	QQPTFN	조회된 기본 표 이름
QQPTMN	QQPTMN	기준 표의 멤버명
QVQTBL	QVQTBL	조회된 표, 긴 이름
QVQLIB	QVQLIB	조회된 표의 라이브러리, 긴 이름
QVPTBL	QVPTBL	기준 표, 긴 이름
QVPLIB	QVPLIB	기준 표의 라이브러리, 긴 이름
QQVTNM	QQNTNM	NLSS 표
QQNLNM	QQNLNM	NLSS 라이브러리
QVSTATUS	QQC11	통계 상태. 가능한 값은 다음과 같습니다. • 'N' - 통계 없음 • 'S' - 실효된 통계 • ' ' - 알 수 없음

표 25. QQQ3015 - 통계 정보에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVSTATIMP	QQI2	통계의 중요도
QVSTATCOL	QQ1000	권장된 통계에 대한 열
QVSTATID	QVC1000	통계 ID

데이터베이스 논리 표 3018 - STRDBMON/ENDDBMON에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3018 - Summary Row for STRDBMON/ENDDBMON
     A*
     A          R QQQ3018                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQJOBT                    RENAME(QQC11)+
                                            COLHDG('Job' +
                                                   'Type')
     A            QQCMDT                    RENAME(QQC12) +
                                            COLHDG('Command' +
                                                   'Type')
     A            QQJOBI                    RENAME(QQC301) +
                                            COLHDG('Job' +
                                                   'Info')
     A          K QQJFLD
     A          S QQRID                     CMP(EQ 3018)
```

표 26. QQQ3018 - STRDBMON/ENDDBMON에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID

표 26. QQQ3018 - STRDBMON/ENDDBMON에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJOBT	QQC11	모니터된 작업 유형 • C - 현재 • J - 작업명 • A - 모두
QQCMDT	QQC12	명령 유형 • S - STRDBMON • E - ENDDBMON
QQJOBI	QQC301	모니터된 작업 정보 • * - 현재 작업 • 작업 번호/사용자/작업명 • *ALL - 모든 작업

데이터베이스 모니터 논리 표 3019 - 검색된 행에 대한 상세 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3019 - Detail Row for Rows Retrieved
     A*
     A          R QQQ3019                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
```

```
                                        COLHDG('Decomposed' +
                                               'Reason' +
                                               'Code')
      A             QDQDTS              RENAME(QVP15D) +
                                        COLHDG('Starting' +
                                               'Decomposed' +
                                               'Subselect')
      A             QQCPUT              RENAME(QQI1) +
                                        COLHDG('Row' +
                                               'Retrieval' +
                                               'CPU Time')
      A             QQCLKT              RENAME(QQI2) +
                                        COLHDG('Row' +
                                               'Retrieval' +
                                               'Clock Time')
      A             QQSYNR              RENAME(QQ13) +
                                        COLHDG('Synch' +
                                               'Reads')
      A             QQSYNW              RENAME(QQ14) +
                                        COLHDG('Synch' +
                                               'Writes')
      A             QQASYR              RENAME(QQ15) +
                                        COLHDG('Asynch' +
                                               'Reads')
      A             QQASYW              RENAME(QQ16) +
                                        COLHDG('Asynch' +
                                               'Writes')
      A             QQRCDR              RENAME(QQ17) +
                                        COLHDG('Rows' +
                                               'Returned')
      A             QQGETC              RENAME(QQ18) +
                                        COLHDG('Number' +
                                               'of Gets')
      A           K QQJFLD
      A           S QQRID               CMP(EQ 3019)
```

표 27. QQQ3019 - 검색된 행에 대한 상세 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호

표 27. QQQ3019 - 검색된 행에 대한 상세 행 (계속)

논리 열 이름	실제 열 이름	설명
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QQCPUT	QQI1	모든 행을 리턴하는 CPU 시간(밀리초 단위)
QQCLKT	QQI2	모든 행을 리턴하는 시간(밀리초 단위)
QQSYNR	QQI3	동기화 데이터베이스 읽기 수
QQSYNW	QQI4	동기화 데이터베이스 쓰기 수
QQASYR	QQI5	비동기 데이터베이스 읽기 수
QQASYW	QQI6	비동기 데이터베이스 쓰기 수
QQRCDR	QQI7	리턴된 행 수
QQGETC	QQI8	리턴된 행을 검색하기 위한 호출 수

데이터베이스 모니터 논리 표 3021 - 작성된 비트맵에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3021 - Summary Row for Bitmap Created
     A*
     A* New row added for Visual Explain
     A*
     A          R QQQ3021                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
```

```
A            QDQDTT                    RENAME(QVP15B) +
                                       COLHDG('Number of' +
                                              'Decomposed' +
                                              'Subselects')
A            QDQDTR                    RENAME(QVP15C) +
                                       COLHDG('Decomposed' +
                                              'Reason' +
                                              'Code')
A            QDQDTS                    RENAME(QVP15D) +
                                       COLHDG('Starting' +
                                              'Decomposed' +
                                              'Subselect')
A            QVRCNT
A            QVPARPF
A            QVPARPL
A            QVPARD
A            QVPARU
A            QVPARRC
A            QQEPT
A            QVCTIM
A            QQREST
A            QQAJN
A            QQJNP
A            QQJNDS                    RENAME(QQI6) +
                                       COLHDG('Data Space' +
                                              'Number')
A            QQJNMT                    RENAME(QQC21) +
                                       COLHDG('Join' 'Method')
A            QQJNTY                    RENAME(QQC22) +
                                       COLHDG('Join' 'Type')
A            QQJNOP                    RENAME(QQC23) +
                                       COLHDG('Join' 'Operator')
A            QVJFANO
A            QVFILES
A            QVBMSIZ                   RENAME(QQI2) +
                                       COLHDG('Bitmap' +
                                              'Size')
A            QVBMCNT                   RENAME(QVP151) +
                                       COLHDG('Number of' +
                                              'Bitmaps'
                                              'Created')
A            QVBMIDS                   RENAME(QVC3001) +
                                       COLHDG('Internal' +
                                              'Bitmap' 'IDs')
A          K QQJFLD
A          S QQRID                     CMP(EQ 3021)
```

표 28. QQQ3021 - 작성된 비트맵에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명

표 28. QQQ3021 - 작성된 비트맵에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQREST	QQREST	예상 선택 행 수
QQAJN	QQAJN	결합된 예상 행 수
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 수/원래 표 위치
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합

표 28. QQQ3021 - 작성된 비트맵에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVBMSIZ	QQI2	비트맵 크기
QVBMCNT	QVP151	작성된 비트맵의 수
QVBMIDS	QVC3001	내부 비트맵 ID

데이터베이스 모니터 논리 표 3022 - 비트맵 병합에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3022 - Summary Row for Bitmap Merge
     A*
     A* New row added for Visual Explain
     A*
     A          R QQQ3022                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
```

```
A            QQMATULVL                 RENAME(QVP15E) +
                                       COLHDG('Materialized' +
                                              'Union' +
                                              'Level')
A            QDQDTN                    RENAME(QVP15A) +
                                       COLHDG('Decomposed' +
                                              'Subselect' +
                                              'Number')
A            QDQDTT                    RENAME(QVP15B) +
                                       COLHDG('Number of' +
                                              'Decomposed' +
                                              'Subselects')
A            QDQDTR                    RENAME(QVP15C) +
                                       COLHDG('Decomposed' +
                                              'Reason' +
                                              'Code')
A            QDQDTS                    RENAME(QVP15D) +
                                       COLHDG('Starting' +
                                              'Decomposed' +
                                              'Subselect')
A            QVRCNT
A            QVPARPF
A            QVPARPL
A            QVPARD
A            QVPARU
A            QVPARRC
A            QQEPT
A            QVCTIM
A            QQREST
A            QQAJN
A            QQJNP
A            QQJNDS                    RENAME(QQI6) +
                                       COLHDG('Data Space' +
                                              'Number')
A            QQJNMT                    RENAME(QQC21) +
                                       COLHDG('Join' 'Method')
A            QQJNTY                    RENAME(QQC22) +
                                       COLHDG('Join' 'Type')
A            QQJNOP                    RENAME(QQC23) +
                                       COLHDG('Join' 'Operator')
A            QVJFANO
A            QVFILES
A            QVBMSIZ                   RENAME(QQI2) +
                                       COLHDG('Bitmap' +
                                              'Size')
A            QVBMID                    RENAME(QVC101) +
                                       COLHDG('Internal' +
                                              'Bitmap' 'ID')
A            QVBMIDMG                  RENAME(QVC3001) +
                                       COLHDG('Bitmaps' +
                                              'Merged')
A          K QQJFLD
A          S QQRID                     CMP(EQ 3022)
```

표 29. QQQ3022 - 비트맵 병합에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별

표 29. QQQ3022 - 비트맵 병합에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMQTULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQREST	QQREST	예상 선택 행 수
QQAJN	QQAJN	결합된 예상 행 수
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 수/원래 표 위치
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합

표 29. QQQ3022 - 비트맵 병합에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVBMSIZ	QQI2	비트맵 크기
QVBMID	QVC101	내부 비트맵 ID
QVBMIDMG	QVC3001	병합된 비트맵의 ID

데이터베이스 모니터 논리 표 - 작성된 임시 해시 표에 대한 요약

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3023 - Summary for Temp Hash Table Created
     A*
     A* New row added for Visual Explain
     A*
     A          R QQQ3023                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
```

```
|       A            QQMATULVL                RENAME(QVP15E) +
|                                             COLHDG('Materialized' +
|                                                    'Union' +
|                                                    'Level')
|       A            QDQDTN                   RENAME(QVP15A) +
|                                             COLHDG('Decomposed' +
|                                                     'Subselect' +
|                                                     'Number')
|       A            QDQDTT                   RENAME(QVP15B) +
|                                             COLHDG('Number of' +
|                                                     'Decomposed' +
|                                                     'Subselects')
|       A            QDQDTR                   RENAME(QVP15C) +
|                                             COLHDG('Decomposed' +
|                                                     'Reason' +
|                                                     'Code')
|       A            QDQDTS                   RENAME(QVP15D) +
|                                             COLHDG('Starting' +
|                                                     'Decomposed' +
|                                                     'Subselect')
|       A            QVRCNT
|       A            QVPARPF
|       A            QVPARPL
|       A            QVPARD
|       A            QVPARU
|       A            QVPARRC
|       A            QQEPT
|       A            QVCTIM
|       A            QQREST
|       A            QQAJN
|       A            QQJNP
|       A            QQJNDS                   RENAME(QQI8) +
|                                             COLHDG('Data Space' +
|                                                     'Number')
|       A            QQJNMT                   RENAME(QQC21) +
|                                             COLHDG('Join' 'Method')
|       A            QQJNTY                   RENAME(QQC22) +
|                                             COLHDG('Join' 'Type')
|       A            QQJNOP                   RENAME(QQC23) +
|                                             COLHDG('Join' 'Operator')
|       A            QVJFANO
|       A            QVFILES
|       A            QVHTRC                   RENAME(QVC1F) +
|                                             COLHDG('Hash' +
|                                                     'Table' +
|                                                     'Reason Code')
|       A            QVHTENT                  RENAME(QQI2) +
|                                             COLHDG('Hash' +
|                                                     'Table' +
|                                                     'Entries')
|       A            QVHTSIZ                  RENAME(QQI3) +
|                                             COLHDG('Hash' +
|                                                     'Table' +
|                                                     'Size')
|       A            QVHTRSIZ                 RENAME(QQI4) +
|                                             COLHDG('Hash' +
|                                                     'Table' +
|                                                     'Row' 'Size')
```

```
A            QVHTKSIZ                  RENAME(QQI5) +
                                       COLHDG('Hash' +
                                              'Key' +
                                              'Size')
A            QVHTESIZ                  RENAME(QQI6) +
                                       COLHDG('Hash' +
                                              'Element' +
                                              'Size')
A            QVHTPSIZ                  RENAME(QQI7) +
                                       COLHDG('Pool' +
                                              'Size')
A            QVHTPID                   RENAME(QQI8) +
                                       COLHDG('Pool' +
                                              'ID')
A            QVHTNAM                   RENAME(QVC101) +
                                       COLHDG('Hash' +
                                              'Table' +
                                              'Name')
A            QVHTLIB                   RENAME(QVC102) +
                                       COLHDG('Hash' +
                                              'Table' +
                                              'Library')
A            QVHTCOL                   RENAME(QVC3001) +
                                       COLHDG('Hash' +
                                              'Table' +
                                              'Columns')
A          K QQJFLD
A          S QQRID                   CMP(EQ 3023)
```

표 30. QQQ3023 - 작성된 임시 해시 표에 대한 요약

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드

표 30. QQQ3023 - 작성된 임시 해시 표에 대한 요약 (계속)

논리 열 이름	실제 열 이름	설명
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQREST	QQREST	예상 선택 행 수
QQAJN	QQAJN	결합된 예상 행 수
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 수/원래 표 위치
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수

표 30. QQQ3023 - 작성된 임시 해시 표에 대한 요약 (계속)

논리 열 이름	실제 열 이름	설명
QVHTRC	QVC1F	해시 표 이유 코드 • J - 해시 결합을 위해 작성됨 • G - 해시 그룹화를 위해 작성됨
QVHTENT	QQI2	해시 표 항목
QVHTSIZ	QQI3	해시 표 크기
QVHTRSIZ	QQI4	해시 표 행 크기
QVHTKSIZ	QQI5	해시 표 키 크기
QVHTESIZ	QQIA	해시 표 요소 크기
QVHTPSIZ	QQI7	해시 표 풀(pool) 크기
QVHTPID	QQI8	해시 표 풀(pool) ID
QVHTNAM	QVC101	해시 표 내부 이름
QVHTLIB	QVC102	해시 표 라이브러리
QVHTCOL	QVC3001	해시 표 작성에 사용된 열

데이터베이스 모니터 논리 표 - 고유한 처리에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3025 - Summary Row for Distinct Processing
     A*
     A* New row added for Visual Explain
     A*
     A          R QQQ3025                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
```

```
                                        COLHDG('Number of' +
                                               'Decomposed' +
                                               'Subselects')
      A             QDQDTR              RENAME(QVP15C) +
                                        COLHDG('Decomposed' +
                                               'Reason' +
                                               'Code')
      A             QDQDTS              RENAME(QVP15D) +
                                        COLHDG('Starting' +
                                               'Decomposed' +
                                               'Subselect')
      A             QVRCNT
      A             QVPARPF
      A             QVPARPL
      A             QVPARD
      A             QVPARU
      A             QVPARRC
      A             QQEPT
      A             QVCTIM
      A             QQREST
      A           K QQJFLD
      A           S QQRID               CMP(EQ 3025)
```

표 31. QQQ3025 - 고유한 처리에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)

표 31. QQQ3025 - 고유한 처리에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQREST	QQREST	예상 선택 행 수

데이터베이스 모니터 논리 표 3027 - 부속 조회 병합에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3027 - Summary Row for Subquery Merge
     A*
     A* New row added for Visual Explain
     A*
     A          R QQQ3027                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
                                            COLHDG('Decomposed' +
                                                   'Reason' +
                                                   'Code')
     A            QDQDTS                    RENAME(QVP15D) +
                                            COLHDG('Starting' +
                                                   'Decomposed' +
```

```
|                                                 'Subselect')
|       A           QVRCNT
|       A           QVPARPF
|       A           QVPARPL
|       A           QVPARD
|       A           QVPARU
|       A           QVPARRC
|       A           QQEPT
|       A           QVCTIM
|       A           QQREST
|       A           QQAJN
|       A           QQJNP
|       A           QQJNDS                  RENAME(QQI6) +
|                                           COLHDG('Data Space' +
|                                                  'Number')
|       A           QQJNMT                  RENAME(QQC21) +
|                                           COLHDG('Join' 'Method')
|       A           QQJNTY                  RENAME(QQC22) +
|                                           COLHDG('Join' 'Type')
|       A           QQJNOP                  RENAME(QQC23) +
|                                           COLHDG('Join' 'Operator')
|       A           QVJFANO
|       A           QVFILES
|       A           QVIQDTN                 RENAME(QVP151) +
|                                           COLHDG('Subselect' +
|                                                  'Number' +
|                                                  'Inner')
|       A           QVIQDTL                 RENAME(QVP152) +
|                                           COLHDG('Subselect' +
|                                                  'Level' +
|                                                  'Inner')
|       A           QVIMATN                 RENAME(QVP153) +
|                                           COLHDG('View' +
|                                                  'Number' +
|                                                  'Inner')
|       A           QVIMATL                 RENAME(QVP154) +
|                                           COLHDG('View' +
|                                                  'Level' +
|                                                  'Inner' +
|                                                  'Subselect')
|       A           QVIMATUL                RENAME(QSP155) +
|                                           COLHDG('Materialized' +
|                                                  'Union' +
|                                                  'of Inner')
|       A           QVSUBOP                 RENAME(QQC101) +
|                                           COLHDG('Subquery' +
|                                                  'Operator')
|       A           QVSUBTYP                RENAME(QVC21) +
|                                           COLHDG('Subquery' +
|                                                  'Type')
|       A           QVCORRI                 RENAME(QQC11) +
|                                           COLHDG('Correlated' +
|                                                  'Columns' +
|                                                  'Exist')
|       A           QVCORRC                 RENAME(QVC3001) +
|                                           COLHDG('Correlated' +
```

```
                                                'Columns')
     A          K QQJFLD
     A          S QQRID                      CMP(EQ 3027)
```

표 32. QQQ3027 - 부속 조회 병합에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	외부 부속 조회에 대한 부속 선택 번호
QQQDTL	QQQDTL	외부 부속 조회에 대한 부속 선택 레벨
QQMATN	QQMATN	외부 부속 조회에 대한 구체화된 보기 부속 선택 번호
QQMATL	QQMATL	외부 부속 조회에 대한 구체화된 보기 부속 선택 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQREST	QQREST	예상 선택 행 수
QQAJN	QQAJN	결합된 예상 행 수
QQJNP	QQJNP	결합 위치(사용할 수 있는 경우)
QQJNDS	QQI6	자료 공간 수/원래 표 위치
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합

표 32. QQQ3027 - 부속 조회 병합에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수
QVIQDTN	QVP151	내부 부속 조회에 대한 부속 선택 번호
QVIQDTL	QVP152	내부 부속 조회에 대한 부속 선택 레벨
QVIMATN	QVP153	내부 부속 조회에 대한 구체화된 보기 부속 선택 번호
QVIMATL	QVP154	내부 부속 조회에 대한 구체화된 보기 부속 선택 레벨
QVIMATUL	QVP155	내부 부속 조회에 대한 구체화된 보기 결합 레벨
QVSUBOP	QQC101	부속 조회 연산자. 가능한 값은 다음과 같습니다. • EQ - 같음 • NE - 같지 않음 • LT - 이하임 • LT - 보다 작음 • GE - 이상임 • GT - 보다 큼 • IN • LIKE • EXISTS • NOT - IN, LIKE 또는 EXISTS에 우선할 수 있음

표 32. QQQ3027 - 부속 조회 병합에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVSSUBTYP	QVC21	부속 조회 유형. 가능한 값은 다음과 같습니다. • SQ - 부속 조회 • SS - 스칼라 부속 선택 • SU - Set 갱신
QVCORRI	QQC11	상관된 열 존재(Y/N)
QVCORRC	QVC3001	대응하는 QDT 수와 상관된 열 리스트

데이터베이스 모니터 논리 표 3028 - 그룹화에 대한 요약 행

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A* DB Monitor logical table 3028 - Summary Row for Grouping
     A*
     A* New row added for Visual Explain
     A*
     A          R QQQ3028                   PTABLE(*CURLIB/QAQQDBMN)
     A            QQRID
     A            QQTIME
     A            QQJFLD
     A            QQRDBN
     A            QQSYS
     A            QQJOB
     A            QQUSER
     A            QQJNUM
     A            QQTHRD                    RENAME(QQI9) +
                                            COLHDG('Thread' +
                                                   'Identifier')
     A            QQUCNT
     A            QQUDEF
     A            QQQDTN
     A            QQQDTL
     A            QQMATN
     A            QQMATL
     A            QQMATULVL                 RENAME(QVP15E) +
                                            COLHDG('Materialized' +
                                                  'Union' +
                                                  'Level')
     A            QDQDTN                    RENAME(QVP15A) +
                                            COLHDG('Decomposed' +
                                                   'Subselect' +
                                                   'Number')
     A            QDQDTT                    RENAME(QVP15B) +
                                            COLHDG('Number of' +
                                                   'Decomposed' +
                                                   'Subselects')
     A            QDQDTR                    RENAME(QVP15C) +
                                            COLHDG('Decomposed' +
                                                   'Reason' +
                                                   'Code')
     A            QDQDTS                    RENAME(QVP15D) +
                                            COLHDG('Starting' +
                                                   'Decomposed' +
```

```
|                                                 'Subselect')
|       A             QVRCNT
|       A             QVPARPF
|       A             QVPARPL
|       A             QVPARD
|       A             QVPARU
|       A             QVPARRC
|       A             QQEPT
|       A             QVCTIM
|       A             QQREST
|       A             QQAJN
|       A             QQJNP
|       A             QQJNDS                  RENAME(QQI6) +
|                                             COLHDG('Data Space' +
|                                                    'Number')
|       A             QQJNMT                  RENAME(QQC21) +
|                                             COLHDG('Join' 'Method')
|       A             QQJNTY                  RENAME(QQC22) +
|                                             COLHDG('Join' 'Type')
|       A             QQJNOP                  RENAME(QQC23) +
|                                             COLHDG('Join' 'Operator')
|       A             QVJFANO
|       A             QVFILES
|       A             QVGBYIM                 RENAME(QQC11) +
|                                             COLHDG('Grouping' +
|                                                    'Implementation')
|       A             QVGBYIT                 RENAME(QQC15) +
|                                             COLHDG('Index' +
|                                                    'Type')
|       A             QVGBYIX                 RENAME(QQC101) +
|                                             COLHDG('Grouping' +
|                                                    'Index')
|       A             QVGBYIL                 RENAME(QQC102) +
|                                             COLHDG('Grouping' +
|                                                    'Index' +
|                                                    'Library')
|       A             QVGBYIXL                RENAME(QVINAM) +
|                                             COLHDG('Grouping' +
|                                                    'Index' +
|                                                    'Long Name')
|       A             QVGBYILL                RENAME(QVILIB) +
|                                             COLHDG('Grouping' +
|                                                    'Library' +
|                                                    'Long Name')
|       A             QVGBYHV                 RENAME(QQC12) +
|                                             COLHDG('Having' +
|                                                    'Selection' +
|                                                    'Exists')
|       A             QVGBYHW                 RENAME(QQC13) +
|                                             COLHDG('Having to' +
|                                                    'Where' +
|                                                    'Conversion')
|       A             QVGBYN                  RENAME(QQI2) +
|                                             COLHDG('Estimated' +
|                                                    'Number of' +
|                                                    'Groups')
|       A             QVGBYNA                 RENAME(QQI3) +
|                                             COLHDG('Average' +
```

```
                                                'Rows per' +
                                                'Group')
      A            QVGBYCOL                  RENAME(QVC3001) +
                                             COLHDG('Grouping' +
                                                'Columns')
      A            QVGBYMIN                  RENAME(QVC3002) +
                                             COLHDG('MIN' +
                                                'Columns')
      A            QVGBYMAX                  RENAME(QVC3003) +
                                             COLHDG('MAX' +
                                                'Columns')
      A            QVGBYSUM                  RENAME(QVC3004) +
                                             COLHDG('SUM' +
                                                'Columns')
      A            QVGBYCNT                  RENAME(QVC3005) +
                                             COLHDG('COUNT' +
                                                'Columns')
      A            QVGBYAVG                  RENAME(QVC3006) +
                                             COLHDG('AVG' +
                                                'Columns')
      A            QVGBYSTD                  RENAME(QVC3007) +
                                             COLHDG('STDDEV' +
                                                'Columns')
      A            QVGBYVAR                  RENAME(QVC3008) +
                                             COLHDG('VAR' +
                                                'Columns')
      A          K QQJFLD
      A          S QQRID                     CMP(EQ 3028)
```

표 33. QQQ3028 - 그룹화에 대한 요약 행

논리 열 이름	실제 열 이름	설명
QQRID	QQRID	행 식별
QQTIME	QQTIME	시간 행이 작성되었음
QQJFLD	QQJFLD	결합 열(작업에 대해 고유함)
QQRDBN	QQRDBN	관계형 데이터베이스명
QQSYS	QQSYS	시스템명
QQJOB	QQJOB	작업명
QQUSER	QQUSER	작업 사용자
QQJNUM	QQJNUM	작업 번호
QQTHRD	QQI9	스레드 ID
QQUCNT	QQUCNT	고유 계수(조회에 따라 고유함)
QQUDEF	QQUDEF	사용자 정의 열
QQQDTN	QQQDTN	고유 부속 선택 번호
QQQDTL	QQQDTL	부속 선택 내포 레벨
QQMATN	QQMATN	구체화된 보기 부속 선택 번호
QQMATL	QQMATL	구체화된 보기 내포 레벨
QQMATULVL	QVP15E	구체화된 보기 결합 레벨
QDQDTN	QVP15A	분해된 조회 부속 선택 번호로서, 분해된 모든 부속 선택 중에서 고유함
QDQDTT	QVP15B	분해된 부속 선택의 전체 수
QDQDTR	QVP15C	분해된 조회 부속 선택 이유 코드

표 33. *QQQ3028 - 그룹화에 대한 요약 행 (계속)*

논리 열 이름	실제 열 이름	설명
QDQDTS	QVP15D	첫 번째 분해된 부속 선택에 대한 분해된 조회 부속 선택 번호
QVRCNT	QVRCNT	고유 화면정리 카운터
QVPARPF	QVPARPF	병렬 사전페치(Y/N)
QVPARPL	QVPARPL	병렬 사전로드(색인 사용)
QVPARD	QVPARD	요구된 병렬 정도(색인 사용)
QVPARU	QVPARU	사용된 병렬 정도(색인 사용)
QVPARRC	QVPARRC	이유 병렬 처리가 제한되었음(색인 사용)
QQEPT	QQEPT	예상 처리 시간(초 단위)
QVCTIM	QVCTIM	예상 누적 시간(초 단위)
QQREST	QQREST	예상 선택 행 수
QQAJN	QQAJN	결합된 예상 행 수
QQJNP	QQJNP	결합 위치
QQJNDS	QQI1	자료 공간 수/원래 표 위치
QQJNMT	QQC21	결합 메소드(사용할 수 있는 경우) • NL - 내포된 루프 • MF - 선택이 있는 내포된 루프 • HJ - 해시 결합
QQJNTY	QQC22	결합 유형(사용할 수 있는 경우) • IN - 내부 결합 • PO - 왼쪽 부분 외부 결합 • EX - 예외 결합
QQJNOP	QQC23	결합 연산자(사용할 수 있는 경우) • EQ - 같음 • NE - 같지 않음 • GT - 보다 큼 • GE - 보다 크거나 같음 • LT - 보다 작음 • LE - 보다 작거나 같음 • CP - 카테시안 적
QVJFANO	QVJFANO	결합 팬아웃. 가능한 값은 다음과 같습니다. • N - 팬아웃이 허용되고 결합 팬아웃의 각각의 대응하는 열이 리턴되는 정상 결합 상태. • D - 고유한 팬아웃. 결합 팬아웃이 허용되나 모든 결합 팬아웃 열은 리턴되지 않음. • U - 고유 팬아웃. 결합 팬아웃이 허용됨. 결합 팬아웃이 발생하는 경우 오류 상태.
QVFILES	QVFILES	결합된 표 수

표 33. QQQ3028 - 그룹화에 대한 요약 행 (계속)

논리 열 이름	실제 열 이름	설명
QVGBYI	QQC11	Groupby 구현 • ' ' - 그룹화되지 않음 • I - 색인 • H - 해시
QVGBYIT	QQC15	색인 유형. 가능한 값은 다음과 같습니다. • B - 2진 기수 색인 • C - 제한 조건(2진 기수) • E - 코드화 벡터 색인(EVI) • X - 조회 작성 임시 색인
QVGBYIX	QQC101	그룹화에 사용된 색인 또는 제한 조건
QVGBYIL	QQC102	그룹화에 사용된 라이브러리의 색인
QVGBYIXL	QVINAM	그룹화에 사용된 긴 이름의 색인 또는 제한 조건
QVGBYILL	QVILIB	그룹화에 사용된 라이브러리의 긴 이름의 색인 또는 제한 조건
QVGBYHV	QQC12	Having 선택이 있음(Y/N)
QVGBYHW	QQC13	Having to Where 변환(Y/N)
QVGBYN	QQI2	예상 그룹 수
QVGBYNA	QQI3	각 그룹의 평균 행 수
QVGBYCOL	QVC3001	그룹화 열
QVGBYMIN	QVC3002	MIN 열
QVGBYMAX	QVC3003	최대 열
QVGBYSUM	QVC3004	SUM 열
QVGBYCNT	QVC3005	COUNT 열
QVGBYAVG	QVC3006	평균 열
QVGBYSTD	QVC3007	STDDEV 열
QVGBYVAR	QVC3008	VAR 열

부록 B. 메모리 상주 데이터베이스 모니터: DDS

이 부록에는 메모리 상주 데이터베이스 모니터 실제 및 논리 파일을 작성하는 데 사용되는 다음과 같은 DDS가 들어 있습니다.

외부 표 설명(QAQQQRYI) - SQL 정보에 대한 요약 행

표 34. QAQQQRYI - SQL 정보에 대한 요약 행

열 이름	설명
QQKEY	단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)
QQTIME	행 작성 시간
QQJOB	작업명
QQUSER	작업 사용자
QQJNUM	작업 번호
QQTHID	스레드 Id
QQUDEF	사용자 정의 열
QQPLIB	프로그램 또는 패키지가 들어 있는 라이브러리명
QQCNAM	커서명
QQPNAM	현재의 SQL문이 들어 있는 패키지 또는 프로그램명
QQSNAM	SQL문에 대한 명령문명(해당되는 경우)
QQCNT	명령문 사용 계수
QQAVGT	평균 실행시간(밀리초 단위)
QQMINT	최소 실행시간(밀리초 단위)
QQMAXT	최대 실행시간(밀리초 단위)
QQOPNT	실행 비용이 가장 높은 경우의 열기 시간(밀리초 단위)
QQFETT	실행 비용이 가장 높은 경우의 페치 시간(밀리초 단위)
QQCLST	실행 비용이 가장 높은 경우의 닫기 시간(밀리초 단위)
QQOTHT	실행 비용이 가장 높은 경우의 기타 시간(밀리초 단위)
QQLTU	명령문이 최근에 사용된 시간
QQMETU	비용이 가장 높은 사용 시간
QQAPRT	액세스 계획 리빌드 시간
QQFULO	완전 열기 횟수
QQPSUO	의사 열기 횟수

표 34. *QAQQQRYI - SQL 정보에 대한 요약 행 (계속)*

열 이름	설명
QQTOTR	**비결합의 경우, 표의 총 행 수**
QQRROW	**리턴되는 결과 행 수**
QQRROW	**명령 함수** S - 선택 - 갱신 I - 삽입 D - 삭제 L - 자료 정의어 O - 기타

표 34. QAQQQRYI - SQL 정보에 대한 요약 행 (계속)

열 이름	설명
QQSTOP	**명령문 조작** • AL - 표 변경 • CA - 호출 • CC - 콜렉션 작성 • CD - 유형 작성 • CF - 함수 작성 • CG - 트리거 작성 • CI - 색인 작성 • CL - 닫기 • CM - 확약 • CN - 연결 • CO - 주석 닫기 • CP - 프로시듀어 작성 • CS - 별명/동의어 작성 • CT - 표 작성 • CV - 보기 작성 • DE - 설명 • DI - 단절 • DL - 삭제 • DM - 매개변수 마커 설명 • DP - 프로시듀어 선언 • DR - 드롭(drop) • DT - 표 설명 • EI - 즉시 실행 • EX - 실행 • FE - 페치 • FL - 자유 위치 지정자 • GR - 부여 • HC - 하드 닫기 • HL - 보류 위치 지정자 • IN - 삽입 • JR - 재사용된 서버 작업 • LK - 잠금 • LO - 레이블 • MT - 텍스트 계속 • OP - 열기 • PD - 준비 및 설명 • PR - 준비 • RB - 롤백 저장점 • RE - 해제 • RO - 롤백

표 34. *QAQQQRYI - SQL 정보에 대한 요약 행 (계속)*

열 이름	설명
QQSTOP(계속)	• RS - 해제 저장점 • RT - 표 이름 변경 • RV - 취소 • SA - 저장점 • SC - 연결 설정 • SI - 선택 위치 • SP - 경로 설정 • SR - 결과 세트 설정 • SS - 현재 스키마 설정 • ST - 트랜잭션 설정 • SV - 변수 설정 • UP - 갱신 • VI - 들어갈 값
QQODPI	**ODP 구현** R - 재사용 가능 ODP(ISV) N - 재사용 불가능 ODP(V2)
QQHVI	**호스트 변수 구현** I - 인터페이스 제공 값(ISV) V - 상수로 취급되는 호스트 변수(V2) U - 표 관리 행 위치지정(UP)

표 34. QAQQQRYI - SQL 정보에 대한 요약 행 (계속)

열 이름	설명	
QQAPR	**액세스 계획 리빌드**	
	A1	표 또는 멤버가 액세스 계획이 최종 빌드 되었을 때 참조된 표 또는 멤버와 동일한 오브젝트가 아닙니다. 오브젝트가 다른 이유는 다음과 같습니다. • 오브젝트가 삭제되고 리빌드되었음. • 오브젝트가 저장되고 복원되었음. • 라이브러리 리스트가 변경되었음. • 오브젝트의 이름이 변경되었음. • 오브젝트가 이동되었음. • 오브젝트가 다른 오브젝트로 대체되었음. • 이번이 조회가 들어 있는 오브젝트가 복원된 이후 해당 조회를 처음으로 실행하는 경우임.
	A2	재사용 가능한 ODP(열린 자료 경로)를 사용하도록 액세스 계획이 빌드되었는데 Optimizer가 해당 호출에 재사용 불가능한 ODB를 사용하기로 선택하였습니다.
	A3	재사용 불가능한 ODP(열린 자료 경로)를 사용하도록 액세스 계획이 빌드되었는데 Optimizer가 해당 호출에 재사용 가능한 ODB를 사용하기로 선택하였습니다.
	A4	표 행 수가 액세스 계획이 마지막으로 빌드된 이후로 10% 이상 변경되었습니다.
	A5	조회의 표 중 하나에 대해 신규 색인이 존재합니다.
	A6	해당 액세스 계획에 사용된 색인이 더 이상 존재하지 않거나 더 이상 유효하지 않습니다.
	A7	시스템 프로그래밍 변경으로 인해 OS/400 조회에 리빌드된 액세스 계획이 필요합니다.
	A8	현재 작업의 CCSID가 액세스 계획을 최종 작성한 작업의 CCSID와 다릅니다.
	A9	현재 작업에서 다음 중 하나 이상의 값이 현재 작업에 대해 해당 액세스 계획을 마지막으로 작성했을 때의 작업에 대한 값과 다릅니다. • 날짜 형식 • 날짜 분리자 • 시간 형식 • 시간 분리자
	AA	지정된 정렬 순서표가 해당 액세스 계획을 작성할 때 사용된 정렬 순서표와 다릅니다.
	AB	기억장치 풀이 변경되었거나 CHGQRYA 명령의 DEGREE 매개변수가 변경되었습니다.
	AC	시스템 피처 DB2 다중 시스템이 설치 또는 제거되었습니다.
	AD	정도 조회 속성 값이 변경되었습니다.
	AE	보기가 상위 레벨 언어에 의해 열리거나 구체화되는 중입니다.
	AF	사용자 정의 유형 또는 사용자 정의 기능이 액세스 계획에서 참조된 유형 또는 기능과 동일한 오브젝트가 아니거나 SQL 경로가 액세스 계획이 빌드될 때와 동일하지 않습니다.
	B0	지정된 옵션이 조회 옵션 파일 QAQQINI의 결과로 변경되었습니다.
	B1	액세스 계획이 현재 작업과 다른 확약 제어 레벨로 생성되었습니다.
	B2	액세스 계획이 이전 액세스 계획과 다른 정적 커서 응답 설정 크기로 생성되었습니다.

표 34. QAQQQRYI - SQL 정보에 대한 요약 행 (계속)

열 이름	설명	
QQDACV	**자료 변환**	
	N	아니오.
	0	적용할 수 없음.
	1	길이가 일치하지 않음.
	2	숫자 유형이 일치하지 않음.
	3	C 호스트 변수가 널(null)로 종료됨.
	4	호스트 변수 또는 열이 가변 길이인데 다른 쪽은 가변 길이가 아님.
	5	CCSID 변환.
	6	DRDA 및 NULL 허용, 가변 길이가 부분 행에 들어 있거나, 표현식을 도출했거나, 충분한 호스트 변수 없이 페치를 블록화했습니다.
	7	자료, 시간 또는 시간소인 열.
	8	호스트 변수가 너무 많습니다.
	9	삽입 목표 표가 SQL 표가 아닙니다.
QQCTS	**명령문 표 스캔 사용 계수**	
QQCIU	**명령문 색인 사용 계수**	
QQCIC	**명령문 색인 작성 계수**	
QQCSO	**명령문 정렬 사용 계수**	
QQCTF	**명령문 임시 표 계수**	
QQCIA	**명령문 색인 제안 계수**	
QQCAPR	**명령문 액세스 계획 리빌드 계수**	
QQARSS	**평균 결과 세트 크기**	
QQC11	**예약**	
QQC12	**예약**	
QQC21	**예약**	
QQC22	**예약**	
QQI1	**예약**	
QQI2	**예약**	
QQC301	**예약**	
QQC302	**예약**	
QQC1000	**예약**	

외부 표 설명(QAQQTEXT) - SQL문에 대한 요약 행

표 35. QAQQTEXT - SQL문에 대한 요약 행

열 이름	설명
QQKEY	**단일 조회의 행을 행 식별과 함께 링크하는 데 사용되는 결합 열(조회마다 고유)**
QQTIME	**행 작성 시간**
QQSTTX	**명령문 텍스트**
QQC11	**예약**
QQC12	**예약**
QQC21	**예약**
QQC22	**예약**

표 35. QAQQTEXT - SQL문에 대한 요약 행 (계속)

열 이름	설명
QQQI1	**예약**
QQI2	**예약**
QQC301	**예약**
QQC302	**예약**
QQ1000	**예약**

외부 표 설명(QAQQ3000) - 도달순에 대한 요약 행

표 36. QAQQ3000 - 도달순에 대한 요약 행

열 이름	설명
QQKEY	**단일 조회의 행을 행 식별과 함께 링크하는 데 사용되는 결합 열(조회마다 고유)**
QQTIME	**행 작성 시간**
QQQDTN	**QDT 번호(ODT마다 고유)**
QQQDTL	**QDT 부속 조회 내포 레벨**
QQMATN	**구체화된 보기 QDT 번호**
QQMATL	**구체화된 보기 내포 레벨**
QQTLN	**라이브러리**
QQTFN	**표**
QQPTLN	**실제 라이브러리**
QQPTFN	**실제 표**
QQTOTR	**표의 총 행 수**
QQREST	**예상 선택 행 수**
QQAJN	**예상 결합 행 수**
QQEPT	**예상 처리 시간(초 단위)**
QQJNP	**결합 위치(사용할 수 있는 경우)**
QQJNDS	**자료 공간 수**
QQJNMT	**결합 방법(사용할 수 있는 경우)** NL - 내포된 루프 MF - 선택이 있는 내포된 루프 HJ - 해시 결합
QQJNTY	**결합 유형(사용할 수 있는 경우)** IN - 내부 결합 PO - 왼쪽 부분 외부 결합 EX - 예외 결합

표 36. QAQQ3000 - 도달순에 대한 요약 행 (계속)

열 이름	설명
QQJNOP	**결합 연산자(사용할 수 있는 경우)** EQ - 같음 NE - 같지 않음 GT - 보다 큼 GE - 보다 크거나 같음 LT - 보다 작음 LE - 보다 작거나 같음 CP - 카테시안 적
QQDSS	**자료 공간 선택** Y - 예 N - 아니오
QQIDXA	**색인 제안** Y - 예 N - 아니오
QQRCOD	**이유 코드** T1 - 색인이 없습니다. T2 - 색인이 있지만 사용할 수 없습니다. T3 - Optimizer가 사용할 수 있는 색인에 대한 표 스캔을 선택하였습니다.
QQLTLN	**라이브러리 길이**
QQLTFN	**표 길이**
QQLPTL	**실제 라이브러리 길이**
QQLPTF	**표 길이**
QQIDXD	**제안된 색인의 키 열**
QQC11	**예약**
QQC12	**예약**
QQC21	**예약**
QQC22	**예약**
QQI1	**색인 스캔 키 위치지정을 사용하는 제안된 키 열의 수**
QQI2	**예약**
QQC301	**예약**
QQC302	**예약**
QQ1000	**예약**

외부 표 설명(QAQQ3001) - 기존 색인 사용에 대한 요약 행

표 37. QQQ3001 - 기존 색인 사용에 대한 요약 행

열 이름	설명
QQKEY	**단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)**
QQTIME	**행 작성 시간**
QQQDTN	**QDT 번호(QDT마다 고유)**
QQQDTL	**RQDT 부속 조회 내포 레벨 관계형 데이터베이스명**

표 37. QQQ3001 - 기존 색인 사용에 대한 요약 행 (계속)

열 이름	설명
QQMATN	**구체화된 보기 QDT 번호**
QQMATL	**구체화된 보기 내포 레벨**
QQTLN	**라이브러리**
QQTFN	**표**
QQPTLN	**실제 라이브러리**
QQPTFN	**실제 표**
QQILNM	**색인 라이브러리**
QQIFNM	**색인**
QQTOTR	**표의 총 행 수**
QQREST	**예상 선택 행 수**
QQFKEY	**키 위치지정 키의 수**
QQKSEL	**키 선택 키의 수**
QQAJN	**결합 위치(사용할 수 있는 경우)**
QQEPT	**예상 처리 시간(초 단위)**
QQJNP	**결합 위치(사용할 수 있는 경우)**
QQJNDS	**자료 공간 수**
QQJNMT	**결합 방법(사용할 수 있는 경우)** NL - 내포된 루프 MF - 선택이 있는 내포된 루프 HJ - 해시 결합
QQJNTY	**결합 유형(사용할 수 있는 경우)** IN - 내부 결합 PO - 왼쪽 부분 외부 결합 EX - 예외 결합
QQJNOP	**결합 연산자(사용할 수 있는 경우)** EQ - 같음 NE - 같지 않음 GT - 보다 큼 GE - 보다 크거나 같음 LT - 보다 작음 LE - 보다 작거나 같음 CP - 카테시안 적
QQIDXK	**색인 스캔 키 위치지정을 사용하는 제안된 키 수**
QQKP	**색인 스캔 키 위치지정** Y - 예 N - 아니오
QQKPN	**키 위치지정 열의 수**
QQKS	**색인 스캔 키 선택** Y - 예 N - 아니오

표 37. QQQ3001 - 기존 색인 사용에 대한 요약 행 (계속)

열 이름	설명
QQDSS	**자료 공간 선택** Y - 예 N - 아니오
QQIDXA	**색인 제안** Y - 예 N - 아니오
QQRCOD	**이유 코드** I1 - 행 선택 I2 - 순서화/그룹화 I3 - 행 선택 및 순서화/그룹화 I4 - 내포된 루프 결합 I5 - 비트맵 처리를 사용하는 행 선택
QQCST	**제한사항 인디케이터** Y - 예 N - 아니오
QQCSTN	**제한사항명**
QQLTLN	**라이브러리 길이**
QQLTFN	**표 길이**
QQLPTL	**실제 라이브러리 길이**
QQLPTF	**표 길이**
QQLILN	**색인 라이브러리 - 긴**
QQLIFN	**색인 - 긴**
QQIDXD	**제안된 색인의 키 열**
QQC11	**예약**
QQC12	**예약**
QQC21	**예약**
QQC22	**예약**
QQI1	**예약**
QQI2	**예약**
QQC301	**예약**
QQC302	**예약**
QQ1000	**예약**

외부 표 설명(QAQQ3002) - 작성된 색인에 대한 요약 행

표 38. QQQ3002 - 작성된 색인에 대한 요약 행

열 이름	설명
QQKEY	**단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)**

표 38. QQQ3002 - 작성된 색인에 대한 요약 행 (계속)

열 이름	설명
QQTIME	**행 작성 시간**
QQQDTN	**QDT 번호(QDT마다 고유)**
QQQDTL	**RQDT 부속 조회 내포 레벨 관계형 데이터베이스명**
QQMATN	**구체화된 보기 QDT 번호**
QQMATL	**구체화된 보기 내포 레벨**
QQTLN	**라이브러리**
QQTFN	**표**
QQPTLN	**실제 라이브러리**
QQPTFN	**실제 표**
QQILNM	**색인 라이브러리**
QQIFNM	**색인**
QQNTNM	**NLSS 표**
QQNLNM	**NLSS 라이브러리**
QQTOTR	**표의 총 행 수**
QQRIDX	**작성된 색인의 항목 수**
QQREST	**예상 선택 행 수**
QQFKEY	**색인 스캔 키 위치지정 키 수**
QQKSEL	**색인 스캔 키 선택 키 수**
QQAJN	**예상 결합 행 수**
QQJNP	**결합 위치(사용할 수 있는 경우)**
QQJNDS	**자료 공간 수**
QQJNMT	**결합 방법(사용할 수 있는 경우)** NL - 내포된 루프 MF - 선택이 있는 내포된 루프 HJ - 해시 결합
QQJNTY	**결합 유형(사용할 수 있는 경우)** IN - 내부 결합 PO - 왼쪽 부분 외부 결합 EX - 예외 결합
QQJNOP	**결합 연산자(사용할 수 있는 경우)** EQ - 같음 NE - 같지 않음 GT - 보다 큼 GE - 보다 크거나 같음 LT - 보다 작음 LE - 보다 작거나 같음 CP - 카테시안 적
QQIDXK	**색인 스캔 키 위치지정을 사용하는 제안된 키 수**
QQEPT	**예상 처리 시간(초 단위)**

표 38. QQQ3002 - 작성된 색인에 대한 요약 행 (계속)

열 이름	설명
QQKP	**색인 스캔 키 위치지정** Y - 예 N - 아니오
QQKPN	**색인 스캔 키 위치지정 열 수**
QQKS	**색인 스캔 키 선택** Y - 예 N - 아니오
QQDSS	**자료 공간 선택** Y - 예 N - 아니오
QQIDXA	**색인 제안** Y - 예 N - 아니오
QQCST	**제한사항 인디케이터** Y - 예 N - 아니오
QQCSTN	**제한사항명**
QQRCOD	**이유 코드** I1 - 행 선택 I2 - 순서화/그룹화 I3 - 행 선택 및 순서화/그룹화 I4 - 내포된 루프 결합 I5 - 비트맵 처리를 사용하는 행 선택
QQTTIM	**색인 작성 시간**
QQLTLN	**라이브러리 길이**
QQLTFN	**표 길이**
QQLPTL	**실제 라이브러리 길이**
QQLPTF	**표 길이**
QQLILN	**색인 라이브러리 길이**
QQLIFN	**색인 길이**
QQLNTN	**NLSS 표 길이**
QQLNLN	**NLSS 라이브러리 길이**
QQIDXD	**제안된 색인의 키 열**
QQCRTK	**작성된 색인의 키 열**
QQC11	**예약**
QQC12	**예약**
QQC21	**예약**
QQC22	**예약**
QQI1	**예약**
QQI2	**예약**

표 38. QQQ3002 - 작성된 색인에 대한 요약 행 (계속)

열 이름	설명
QQC301	**예약**
QQC302	**예약**
QQ1000	**예약**

외부 표 설명(QAQQ3003) - 조회 정렬에 대한 요약 행

표 39. QQQ3003 - 조회 정렬에 대한 요약 행

열 이름	설명
QQKEY	**단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)**
QQTIME	**행 작성 시간**
QQQDTN	**QDT 번호(QDT마다 고유)**
QQQDTL	**RQDT 부속 조회 내포 레벨 관계형 데이터베이스명**
QQMATN	**구체화된 보기 QDT 번호**
QQMATL	**구체화된 보기 내포 레벨**
QQTTIM	**정렬 시간**
QQRSS	**선택 또는 정렬된 행 수**
QQSIZ	**정렬 간격 크기**
QQPSIZ	**풀(pool) 크기**
QQPID	**풀(pool) ID**
QQIBUF	**내부 정렬 버퍼 길이**
QQEBUF	**외부 정렬 버퍼 길이**
QQRCOD	**이유 코드** **F1** 조회에 둘 이상의 표에서 나온 그룹화 열(Group By)이 들어 있거나, 재순서화할 수 없는 결합 조회의 2차 표에서 나온 그룹화 열이 들어 있습니다. **F2** 조회에 둘 이상의 표에서 나온 순서화 열(Order By)이 들어 있거나, 재순서화할 수 없는 결합 조회의 2차 표에서 나온 순서화 열이 들어 있습니다. **F3** 그룹화 및 순서화 열이 호환되지 않습니다. **F4** 조회에 DISTINCT가 지정되었습니다. **F5** 조회에 UNION이 지정되었습니다. **F6** 정렬을 사용하여 구현해야 하는 조회입니다. 2000바이트를 초과하거나 120개의 열을 초과하는 키 길이가 순서화에 지정되었습니다. **F7** 조회 Optimizer가 색인이 아닌 정렬을 사용하여 조회 결과를 정렬하기로 선택하였습니다. **F8** 지정된 행 선택을 실행하여 I/O 대기 시간을 최소화하십시오. **FC** 조회에 그룹화 필드가 들어 있고, 조회 내에 있는 최소한 한개의 실제 파일에 대한 읽기 트리거가 있습니다.
QQC11	**예약**
QQC12	**예약**

표 39. QQQ3003 - 조회 정렬에 대한 요약 행 (계속)

열 이름	설명
QQC21	예약
QQC22	예약
QQI1	예약
QQI2	예약
QQC301	예약
QQC302	예약
QQ1000	예약

외부 표 설명(QAQQ3004) - 임시 표에 대한 요약 행

표 40. QQQ3004 - 임시 표에 대한 요약 행

열 이름	설명
QQKEY	단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)
QQTIME	행 작성 시간
QQQDTN	**QDT** 번호(QDT마다 고유)
QQQDTL	**RQDT** 부속 조회 내포 레벨 관계형 데이터베이스명
QQMATN	구체화된 보기 **QDT** 번호
QQMATL	구체화된 보기 내포 레벨
QQTLN	라이브러리
QQTFN	표
QQTTIM	임시 표 작성 시간
QQTMPR	임시 표의 행 수

표 40. QQQ3004 - 임시 표에 대한 요약 행 (계속)

열 이름	설명	
QQRCOD	**이유 코드**	
	F1	조회에 둘 이상의 표에서 나온 그룹화 열(Group By)이 들어 있거나, 재순서화할 수 없는 결합 조회의 2차 표에서 나온 그룹화 열이 들어 있습니다.
	F2	조회에 둘 이상의 표에서 나온 순서화 열(Order By)이 들어 있거나, 재순서화할 수 없는 결합 조회의 2차 표에서 나온 순서화 열이 들어 있습니다.
	F3	그룹화 및 순서화 열이 호환되지 않습니다.
	F4	조회에 DISTINCT가 지정되었습니다.
	F5	조회에 UNION이 지정되었습니다.
	F6	정렬을 사용하여 구현해야 하는 조회입니다. 2000바이트를 초과하거나 120개의 열을 초과하는 키 길이가 순서화에 지정되었습니다.
	F7	조회 Optimizer가 색인이 아닌 정렬을 사용하여 조회 결과를 정렬하기로 선택하였습니다.
	F8	지정된 행 선택을 실행하여 I/O 대기 시간을 최소화하십시오.
	F9	조회 Optimizer가 조회에 대한 그룹화를 수행하기 위해 액세스 경로 보다는 해싱 알고리즘을 사용하기로 선택했습니다.
	FA	조회에 임시 파일을 요구하는 결합 조건이 들어 있습니다.
	FB	조회 Optimizer가 조회로 특정 상관 그룹을 구현하기 위해 런타임 임시 파일을 작성합니다.
	FC	조회에 그룹화 필드가 들어 있고, 조회 내에 있는 최소한 한개의 실제 파일에 대한 읽기 트리거가 있습니다.
	FD	조회 Optimizer는 정적 커서 요청에 대한 런타임 임시 파일을 작성합니다.
	H1	표가 결합 논리 파일이며, 그 결합 유형이 조회에 지정된 결합 유형과 일치하지 않습니다.
	H2	논리 표에 대해 지정된 형식이 둘 이상의 기준 표를 참조합니다.
	H3	표가 SQL 보기의 임시 결과를 요구하는 복합 SQL 보기입니다.
	H4	갱신 허용 조회의 경우, 부속 선택은 갱신 중인 열 중 하나와 일치하는 이 표에 있는 열을 참조합니다.
	H5	갱신 허용 조회의 경우, 부속 선택은 갱신 중인 표를 기준으로 한 SQL 보기를 참조합니다.
	H6	삭제 허용 조회의 경우, 부속 선택은 행이 삭제될 표, SQL 보기 또는 행이 삭제될 표를 기준으로 한 색인을 참조합니다.
	H7	사용자 정의 표 함수가 구체화되었습니다.
QQDFVL	**임시로 디폴트 값을 제시할 수 있는지 여부** Y - 예 N - 아니오	
QQLTLN	**라이브러리 길이**	
QQLTFN	**표 길이**	
QQC11	**예약**	

표 40. QQQ3004 - 임시 표에 대한 요약 행 (계속)

열 이름	설명
QQC12	예약
QQC21	예약
QQC22	예약
QQI1	예약
QQI2	예약
QQC301	예약
QQC302	예약
QQ1000	예약

외부 표 설명(QAQQ3007) - Optimizer 정보에 대한 요약 행

표 41. QQQ3007 - Optimizer 정보에 대한 요약 행

열 이름	설명
QQKEY	단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)
QQTIME	행 작성 시간
QQQDTN	**QDT** 번호(QDT마다 고유)
QQQDTL	**RQDT** 부속 조회 내포 레벨 관계형 데이터베이스명
QQMATN	구체화된 보기 **QDT** 번호
QQMATL	구체화된 보기 내포 레벨
QQTLN	라이브러리
QQTFN	표
QQPTLN	실제 라이브러리
QQPTFN	표
QQTOUT	**Optimizer** 시간 종료 Y - 예 N - 아니오
QQIRSN	이유 코드
QQLTLN	라이브러리 길이
QQLTFN	표 길이
QQPTL	실제 라이브러리 길이
QQPTF	표 길이
QQIDXN	색인명
QQC11	예약
QQC12	예약
QQC21	예약
QQC22	예약
QQI1	예약
QQI2	예약
QQC301	예약

표 41. QQQ3007 - Optimizer 정보에 대한 요약 행 (계속)

열 이름	설명
QQC302	예약
QQ1000	예약

외부 표 설명(QAQQ3008) - 부속 조회 처리에 대한 요약 행

표 42. QQQ3008 - 부속 조회 처리에 대한 요약 행

열 이름	설명
QQKEY	단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)
QQTIME	행 작성 시간
QQQDTN	QDT 번호(QDT마다 고유)
QQQDTL	RQDT 부속 조회 내포 레벨 관계형 데이터베이스명
QQMATN	구체화된 보기 QDT 번호
QQMATL	구체화된 보기 내포 레벨
QQORGQ	구체화된 보기 QDT 번호
QQMRGQ	구체화된 보기 내포 레벨
QQC11	예약
QQC12	예약
QQC21	예약
QQC22	예약
QQI1	예약
QQI2	예약
QQC301	예약
QQC302	예약
QQ1000	예약

외부 표 설명(QAQQ3010) - 호스트 변수 및 ODP 구현에 대한 요약 행

표 43. QQQ3010 - 호스트 변수 및 ODP 구현에 대한 요약 행

열 이름	설명
QQKEY	단일 조회의 행을 함께 링크하는 데 사용되는 결합 열(조회마다 고유함)
QQTIME	행 작성 시간
QQHVAR	호스트 변수 값
QQC11	예약
QQC12	예약
QQC21	예약
QQC22	예약
QQI1	예약
QQI2	예약
QQC301	예약

표 43. QQQ3010 - 호스트 변수 및 ODP 구현에 대한 요약 행 (계속)

열 이름	설명
QQC302	**예약**

색인

[가]

[나]

[다]

[라]

[마]

[바]

[사]

[아]

[자]

[차]

[카]

[타]

[파]

[하]

A

C

D

E

I

J

O

P

Q

R

S

Printed in U.S.A