IBM® DB2® Life Sciences Data Connect

계획, 설치 및 구성 안내서

버전 8

GA30-1512-00

IBM® DB2® Life Sciences Data Connect

계획, 설치 및 구성 안내서

버전 8

GA30-1512-00

이 정보 및 이 정보가 지원하는 제품을 사용하기 전에 반드시 *주의사항*에 나와 있는 일반 정보를 읽으십시오.

IBM 서적을 주문하려면 온라인을 통하거나 한국 IBM 담당자에게 문의하십시오.

- 책을 온라인으로 주문하려면 IBM Publications Center(www.ibm.com/shop/publications/order)를 방문하십시오.
- 한국 IBM 담당자에게 문의하려면 IBM Directory of Worldwide Contacts(www.ibm.com/planetwide)를 방문하십시오.

미국이나 캐나다의 DB2 마케팅 및 판매 부서에서 DB2 책을 주문하려면 1-800-IBM-4YOU(426-4968)로 전화하십시오.

IBM에 정보를 보내는 경우, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

목차

책 정보 vii
이 책의 사용자 vii
버전 8의 새로운 기능. vii
온라인 정보 ix
규칙. ix
구문 도표를 읽는 방법. x
의견을 보내는 방법 xii

제 1 장 DB2 Life Sciences Data Connect의 개념 1
DB2 Life Sciences Data Connect 1
IBM Life Sciences DiscoveryLink 3

제 2 장 DB2 Life Sciences Data Connect 설치 5
DB2 Life Sciences Data Connect 설치. . . 5
DB2 Life Sciences Data Connect를 설치하기 전. 7
AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치 . 8
Windows 서버에 DB2 Life Sciences Data Connect 설치. 9
DB2 Life Sciences Data Connect를 설치한 후 11

제 3 장 데이터 소스로서의 테이블 구조 파일 13
테이블 구조 파일의 정의. 13
테이블 구조 파일 유형 14
정렬된 파일 14
정렬되지 않은 파일 14
테이블 구조 파일에서의 DB2 Life Sciences Data Connect 작업 방식 15
페더레이티드 시스템에 테이블 구조 파일 추가 16
테이블 구조 파일 랩퍼 등록. 17
테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정. 18
테이블 구조 파일에 대한 서버 등록 19
테이블 구조 파일에 대한 별칭 등록 20
테이블 구조 파일 랩퍼에 대한 랩퍼 제한사항 및 고려사항 25
테이블 구조 파일 랩퍼에 대한 파일 제한사항 및 고려사항 25
테이블 구조 파일 랩퍼에 대한 파일 액세스 제어 모델 26
테이블 구조 파일 랩퍼에 대한 최적화 추가 정보 및 고려사항 27
테이블 구조 파일 랩퍼에 대한 메시지 . . . 27

제 4 장 데이터 소스로서의 Documentum. . 31
Documentum의 정의 31
페더레이티드 시스템에 Documentum 추가. . 33
Documentum 클라이언트 라이브러리에 링크 (AIX 및 Solaris 운영 환경에만 해당) . . . 35
Documentum의 클라이언트 dmcl.ini 파일 지정 36
Documentum 랩퍼 등록. 38
Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정 39
Documentum 데이터 소스에 대한 서버 등록 40
인수 40
옵션 40
사용자 맵핑(Documentum 랩퍼) 42
Documentum 데이터 소스에 대한 별칭 등록 43
컬럼 옵션. 44
별칭 컬럼 옵션 44
별칭 옵션. 45
의사 컬럼 이해 46
CREATE NICKNAME의 예 50

Documentum 데이터 소스에 대한 사용자 정의 기능 등록. 52
사용자 정의 기능 문자열 인수 규칙 . . . 53
쿼리에서 사용자 정의 기능 사용 53
사용자 정의 기능 테이블. 54
Documentum 데이터 소스에 대한 쿼리 실행 59
Documentum 랩퍼를 위한 CreateNicknameFile 유틸리티의 정의 . . . 60
CreateNicknameFile 유틸리티 설치 (Documentum 랩퍼) 61
CreateNicknameFile 유틸리티 구성 (Documentum 랩퍼) 61
Documentum 등록 테이블에서 DM_ID 오브젝트 유형 맵핑 63
반복 속성 이중 정의(Documentum 랩퍼) . . 64
Documentum 랩퍼에 대한 제한사항 및 고려사항 65
Documentum 랩퍼에 대한 액세스 제어. . . 66
Documentum 랩퍼에 대한 메시지 67

제 5 장 데이터 소스로서의 Excel 73
Excel의 정의 73
Excel 랩퍼에 대한 전제조건 75
페더레이티드 시스템에 Excel 추가 75
Excel 랩퍼 등록 76
Excel 데이터 소스에 대한 서버 등록. . . . 76
인수 정의. 77
Excel 데이터 소스에 대한 별칭 등록. . . . 77
CREATE NICKNAME 구문(Excel용) . . 78
옵션 정의. 78
Excel 데이터 소스에 대한 쿼리 실행. . . . 79
샘플 Excel 랩퍼 시나리오 80
Excel 랩퍼에 대한 랩퍼 제한사항. 83
Excel 파일 제한사항 83
Excel 랩퍼에 대한 파일 액세스 제어 모델. . 84
Excel 랩퍼에 대한 메시지 84

제 6 장 데이터 소스로서의 BLAST. . . . 91
BLAST의 정의. 91
페더레이티드 시스템에 BLAST 추가. 96
blastall 실행 파일 및 매트릭스 파일의 올바른 버전이 설치되었는지 확인 97
BLAST 디먼 구성. 97
BLAST 디먼 시작 101
BLAST 랩퍼 등록 102
BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정 103
BLAST 데이터 소스에 대한 서버 등록 . . 104
인수 104
옵션 105
BLAST 데이터 소스에 대한 별칭 등록 . . 105
별칭 컬럼 옵션. 107
별칭 옵션 108
정의 행 구문 분석 108
고정 컬럼 109
CREATE NICKNAME의 예 112
BLAST SQL 쿼리 구성 113
샘플 BLAST 쿼리 114
BLAST 랩퍼에 대한 최적화 추가 정보 . . 116
BLAST 랩퍼에 대한 메시지 116

제 7 장 데이터 소스로서의 XML 119
XML의 정의 119
페더레이티드 시스템에 XML 추가 124
XML 랩퍼 등록 125
XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정 126
XML 데이터 소스에 대한 서버 등록 . . . 127
XML 데이터 소스에 대한 별칭 등록 . . . 128
루트가 아닌 별칭에 대한 페더레이티드 뷰 작성(XML 랩퍼). 134
XML 데이터 소스에 대한 쿼리 실행 . . . 136
XML 랩퍼에 대한 제한사항 및 고려사항 138
XML 랩퍼에 대한 메시지. 138

제 8 장 비용산정 별칭 옵션 지정 145

제 9 장 별칭 대체 147

별칭 대체 147
데이터 유형 변경 148
별칭 옵션 변경. 148

주의사항 151

상표 154

관련 서적 157

색인 159

책 정보

이 책에는 다음과 같은 내용이 들어 있습니다.

- DB2 Life Sciences Data Connect에 대한 소개, IBM Life Sciences DiscoveryLink 오퍼링에 적용되는 방식, 생명 과학에 맞게 조정된 종합적인 소프트웨어 및 서비스 세트.
- DB2 Life Sciences Data Connect의 설치 명령어.
- 랩퍼를 등록하여 페더레이티드 시스템에 데이터 소스를 추가하는 명령어. SQL을 사용하여 사용자나 응용프로그램이 데이터 소스와 통신할 수 있게 하는 모듈인 랩퍼.

텍스트에 대한 기술적 변경사항은 변경사항의 왼쪽에 있는 수직선에 의해 표시됩니다.

이 책의 사용자

이 책은 관리자가 생명 과학 연구 및 개발 데이터에 대한 페더레이티드 데이터베이스 환경을 설정하고, 응용프로그램 프로그래머가 이러한 환경에 대한 응용프로그램을 개발할 수 있도록 하기 위해 작성되었습니다.

버전 8의 새로운 기능

DB2 Life Sciences Data Connect 버전 8의 새로운 기능은 다음과 같습니다.

일반

- 랩퍼 라이브러리 이름이 갱신되었습니다.
- 비관계형 데이터 소스를 위한 비용산정 별칭 옵션이 추가되었습니다.

비관계형 데이터 소스를 위한 확장 쿼리 계획

지원되는 데이터 소스에 대한 쿼리를 위해 개발된 액세스 방법을 확장시켜 DB2 Life Sciences Data Connect 랩퍼를 전역 쿼리 프로세스에 참

여하도록 재작성하였습니다. 이 새로운 계획 함수는 비관계형 랩퍼에 전송되는 쿼리의 성능을 증가시켜 줍니다.

XML 랩퍼

XML 랩퍼가 추가되었습니다. XML 랩퍼는 XML 데이터 소스에 페더레이티드 액세스를 제공합니다. XML은 BLAST, Documentum, Excel 및 테이블 구조 파일이 들어 있는 DB2 Universal Databse 버전 7 이후의 비관계형 랩퍼의 증가되는 목록을 조인합니다.

테이블 구조 파일 랩퍼

- TYPE, VERSION 및 NODE 서버 옵션이 더 이상 필요하지 않습니다.
- SORTED 별칭 옵션이 추가되었습니다.

Documentum 랩퍼

- ALL_VALUES 별칭 옵션이 추가되었습니다.
- 버전 7의 다음 사용자 정의 기능은 현재 별칭 의사 컬럼입니다.
  - GET_FILE
  - GET_FILE_DEL
  - GET_RENDITION
  - GET_RENDITION_DEL
  - HITS
  - SCORE
- RENDITION_FORMAT 사용자 정의 기능이 추가되었습니다.

Excel 랩퍼

- Excel97 및 Excel2000 데이터 소스에는 하나의 랩퍼 라이브러리 파일만 있으면 됩니다.
- TYPE, VERSION 및 NODE 서버 옵션이 더 이상 필요하지 않습니다.

온라인 정보

이 절에서는 이 제품과 관련된 웹 주소 및 전자우편 주소를 제공합니다.

http://www.ibm.com/software/data/db2/lifesciencesdataconnect/
DB2 Life Sciences Data Connect 제품 웹 사이트

http://www.ibm.com/solutions/lifesciences/discoverylink.html
DiscoveryLink 웹 사이트

http://www.ibm.com/solutions/lifesciences/
IBM Life Sciences 웹 사이트

ls@us.ibm.com
IBM Life Sciences 전자우편 주소

규칙

이 책에서는 다음과 같은 강조표시 규칙을 사용합니다.

굵은체
명령 및 그래픽 사용자 인터페이스(GUI) 제어(예: 필드 이름, 폴더 이름, 메뉴 선택사항)를 나타냅니다.

모노스페이스
입력한 텍스트나 코딩의 예를 나타냅니다.

이탤릭체
값으로 대체해야 하는 변수를 나타냅니다. 이탤릭체는 책 제목을 나타내거나 단어를 강조하기도 합니다.

대문자 오브젝트의 SQL 키워드 및 이름(예: 테이블, 뷰 및 서버)을 나타냅니다.

구문 도표를 읽는 방법

이 책에서는 다음과 같이 정의된 구조를 사용하여 구문을 설명합니다.

왼쪽에서 오른쪽으로, 위에서 아래로 행을 따라 구문 도표를 읽으십시오.

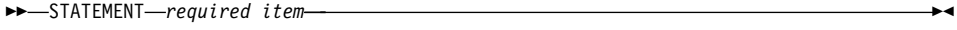
▸▸── 기호는 명령문의 시작을 나타냅니다.

──▸ 기호는 명령문 구문이 다음 행에서 계속됨을 나타냅니다.

▸── 기호는 명령문이 이전 행에서 계속됨을 나타냅니다.

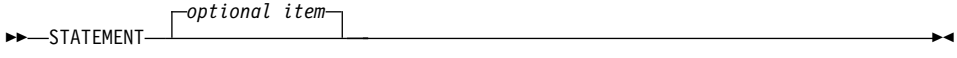
──▸◂ 기호는 명령문의 끝을 나타냅니다.

필수 항목이 수평선(주 경로)에 표시됩니다.



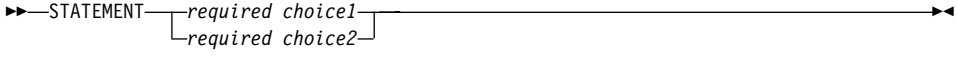
선택적 항목은 주 경로 아래에 표시됩니다.

선택적 항목이 주 경로 위에 표시되는 경우 해당 항목은 명령문의 실행에는 영향을 미치지 않으며 판독에만 사용됩니다.



둘 이상의 항목을 선택할 수 있는 경우 스택으로 표시됩니다.

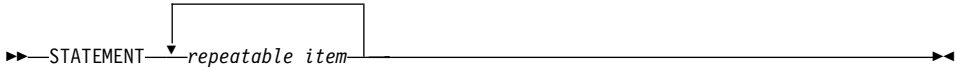
항목 중 하나를 *반드시* 선택해야 하는 경우 스택 중 한 항목이 주 경로에 표시됩니다.



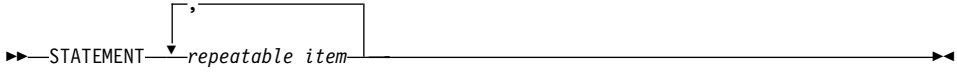
항목 중 어느 것도 선택하지 않아도 되는 경우 주 경로 아래에 스택 전체가 표시됩니다.

항목 중 하나가 디폴트값이면, 하나는 주 경로 위에 표시되고 나머지 선택사항은 아래에 표시됩니다.

주 행 위에 있는 왼쪽으로 돌아가는 화살표는 반복될 수 있는 항목을 표시합니다. 이 경우 반복되는 항목은 하나 이상의 공백으로 구분되어야 합니다.



반복 화살표에 쉼표가 들어 있는 경우 반복되는 항목은 쉼표로 구분해야 합니다.



스택 위의 반복 화살표는 스택 항목에서 하나 이상의 선택사항을 작성하거나 하나의 선택사항을 반복할 수 있음을 표시합니다.

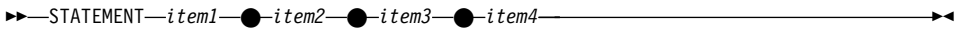
키워드는 대문자로 표시(예: FROM)되며 표시된 것과 똑같이 입력해야 합니다. 변수는 소문자로 표시됩니다(예: column-name). 이는 구문에서 사용자 제공 이름이나 값을 나타냅니다.

구두점, 괄호, 산술 연산자 또는 다른 기호들이 표시된 경우 이를 구문의 일부로서 입력해야 합니다.

때때로 하나의 변수가 여러 개의 매개변수 세트를 표시하기도 합니다. 예를 들어, 다음 도표에서 변수 parameter-block은 **parameter-block**이라는 표제를 갖는 도표의 해석으로 대체될 수 있습니다.

parameter-block:

『큰 점』(●)은 임의의 시퀀스로 지정될 수 있습니다.



위의 도표는 item2 및 item3이 어떤 순서로든 지정될 수 있음을 표시합니다. 다음 둘 다 유효합니다.

```
STATEMENT item1 item2 item3 item4
STATEMENT item1 item3 item2 item4
```

의견을 보내는 방법

고객의 피드백은 IBM이 보다 좋은 품질의 정보를 제공하는 데 도움이 됩니다. 이 책이나 다른 DB2 문서에 대한 의견을 보내 주십시오. 다음 방법 중 하나를 사용하여 의견을 보내실 수 있습니다.

- 웹을 통해 의견을 보내 주십시오. http://www.ibm.com/software/data/rcf에서 IBM Data Management 온라인 고객 의견서 양식에 액세스할 수 있습니다.
- 전자우편으로 의견을 보내실 때는 ibmkspoe@kr.ibm.com으로 보내 주십시오. 제품의 이름, 제품의 버전 번호, 책의 이름 및 부품 번호(적용 가능한 경우)도 함께 보내 주십시오. 특정 텍스트에 대한 의견을 보내는 경우 텍스트의 위치(예 : 장 및 절 제목, 테이블 번호, 페이지 번호 또는 도움말 항목 제목)도 함께 보내 주십시오.

제 1 장 DB2 Life Sciences Data Connect의 개념

이 장에서는 DB2 Life Sciences Data Connect 제품, IBM Life Sciences DiscoveryLink 오퍼링 및 생명 과학 데이터를 쿼리하기 위한 시스템 설정에 관련된 일반 단계를 소개합니다.

DB2 Life Sciences Data Connect

IBM® DB2® Life Sciences Data Connect는 DB2 페더레이티드 시스템이 분산된 소스로부터 유전적, 화학적, 생물학적 및 기타 연구 데이터를 통합할 수 있게 합니다. DB2 페더레이티드 시스템은 DB2 Universal Database™ 서버와, DB2 Universal Database 서버가 데이터를 검색하는 여러 데이터 소스로 구성된 분산 컴퓨팅 시스템입니다.

페더레이티드 시스템에서 사용자나 응용프로그램은 SQL문을 사용하여 IBM, Oracle, Sybase 및 Microsoft의 관계형 데이터베이스 그리고 테이블 구조 파일과 같은 비관계형 데이터 소스와 같은 여러 이기종 데이터 소스에 위치한 데이터를 쿼리, 검색 및 조인할 수 있습니다. 2 페이지의 그림 1에서는 연구 데이터의 여러 소스를 액세스하기 위해 DB2 Life Sciences Data Connect를 사용하는 페더레이티드 시스템에 대해 설명합니다.

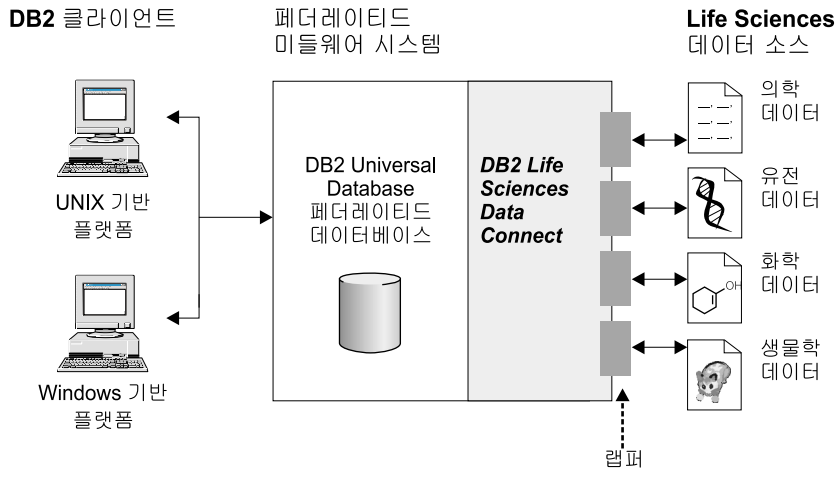


그림 1. DB2 Life Sciences Data Connect를 사용한 생명 과학 데이터 액세스

DB2 페더레이티드 시스템은 클라이언트, 클라이언트가 쿼리를 제출하는 데이터베이스(페더레이티드 데이터베이스라고 함), 페더레이티드 데이터베이스가 데이터 소스와 통신하는 인터페이스 및 데이터 소스 자체를 포함합니다.

페더레이티드 서버가 데이터 소스와 통신하는 메커니즘을 *랩퍼*라고 합니다. 랩퍼를 구현하기 위해 서버는 *랩퍼 모듈*이라고 하는 라이브러리에 저장된 루틴을 사용합니다. 이 루틴은 서버가 데이터 소스에 연결하고 반복적으로 데이터 소스에서 데이터를 검색하는 조작을 수행하게 합니다.

페더레이티드 시스템이 설정되면 데이터 소스에 있는 정보는 하나의 큰 데이터베이스에 있는 것처럼 액세스될 수 있습니다. 사용자 및 응용프로그램은 여러 데이터 소스에서 데이터를 검색하는 쿼리를 하나의 페더레이티드 데이터베이스로 보냅니다. 응용프로그램은 페더레이티드 데이터베이스에 대해서도 다른 DB2 데이터베이스에서와 마찬가지로 작업합니다.

페더레이티드 시스템에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

관련 개념:

- 3 페이지의『IBM Life Sciences DiscoveryLink』

IBM Life Sciences DiscoveryLink

DiscoveryLink 오퍼링은 여러 이기종 데이터 소스의 데이터를 통합하기 위한 생명 과학 연구 및 개발 요구사항에 맞게 조정된 미들웨어 소프트웨어 및 서비스 세트입니다.

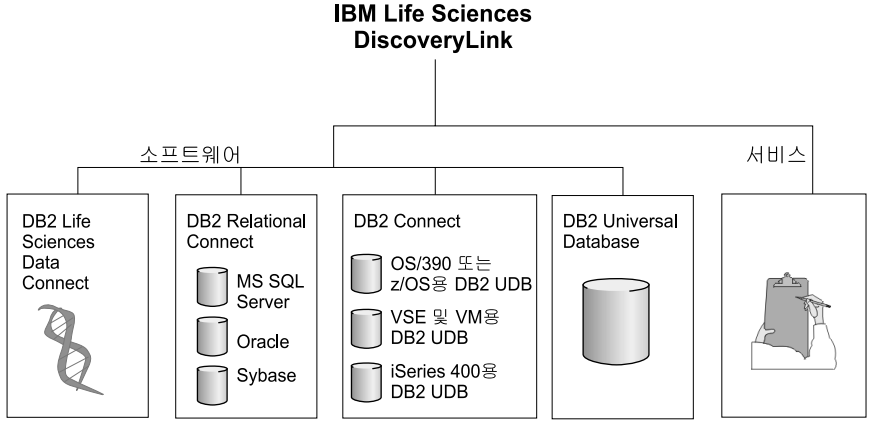


그림 2. IBM Life Sciences DiscoveryLink

예를 들어, DiscoveryLink에서 단일 SQL문을 사용하여 스위스에 있는 Oracle 데이터베이스의 단백질 시퀀스 데이터, 일본에 있는 Sybase 데이터베이스의 화학적 구조 데이터 및 사용자 LAN에 있는 테이블 구조 플랫 파일에 저장된 분광기 데이터를 통합할 수 있습니다. 데이터는 마치 하나의 가상 데이터베이스에 있는 것처럼 표시됩니다.

소프트웨어 구성요소는 다음을 포함합니다.

DB2® Life Sciences Data Connect
생명 과학 데이터에 액세스합니다.

DB2 Relational Connect
Oracle, Sybase 및 Microsoft® 관계형 데이터베이스에 액세스합니다. DB2 Relational Connect에 대한 자세한 정보는 *페더레이티드 시스템 안내서*를 참조하십시오.

DB2 Connect™

호스트 시스템에서 DB2 데이터베이스 서버에 액세스합니다. DB2 Connect에 대한 자세한 정보는 *DB2 Connect 사용자 안내서*를 참조하십시오.

DB2 Universal Database™

여러 이기종 데이터 소스에서 쿼리를 최적화하고 결과를 통합합니다. DB2 Universal Database에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

DiscoveryLink 소프트웨어 및 서비스에 대한 자세한 정보는 아래의 관련 링크 절에서 "온라인 정보"를 참조하십시오.

관련 개념:

- 1 페이지의『DB2 Life Sciences Data Connect』

제 2 장 DB2 Life Sciences Data Connect 설치

이 장에서는 각 랩퍼에 지원되는 플랫폼, UNIX 및 Windows 기반 랩퍼의 설치 지시사항 그리고 설치 완료시 사용자의 시스템에 위치하는 랩퍼 라이브러리에 대해 설명합니다.

DB2 Life Sciences Data Connect 설치

DB2 Life Sciences Data Connect를 사용하여 생명 과학 데이터를 조회하고 검색하려면 랩퍼를 설치한 다음 각 랩퍼를 등록하여 페더레이티드 시스템에 추가해야 합니다.

랩퍼에는 DB2 Universal Database가 필요로 하는 것 이상의 특수 요구사항이 없으며 DB2 Universal Database가 지원하는 시스템 구성에서 실행합니다.

표 1은 각 운영 체제의 DB2 Life Sciences Data Connect 랩퍼를 나타낸 것입니다. 각 생명 과학 랩퍼를 등록하기 위한 지시사항은 아래의 관련 링크 절에 나열된 주제에 제공되어 있습니다.

표 1. 운영 체제별 DB2 Life Sciences Data Connect 랩퍼

랩퍼	Windows	AIX	HP-UX	Linux	Solaris 운영 환경
테이블 구조 파일	X	X	X	X	X
Documentum	X	X			X
Excel	X				
BLAST	X	X		X	X
XML	X	X	X	X	X

설치 프로세스 중 사용자는 세 개의 설치 가능한 구성요소 즉, 과학, 구조화된 파일 및 응용프로그램 중에서 선택할 수 있습니다. 설치 가능한 각 구성요소와 각 구성요소에 포함된 랩퍼의 목록은 6 페이지의 표 2에 제공되어 있습니다.

표 2.

설치 가능한 구성요소 이름	설명	포함된 랩퍼
과학	과학 데이터 소스는 게놈, proteomic, bioinformatic 및 cheminformatic 정보와 같은 생명 과학 분야를 위해 개발되었습니다.	BLAST
구조화된 파일	구조화된 파일 데이터 소스에는 정의된 반복 가능한 구조를 사용하여 파일에 저장된 생명 과학 데이터가 들어 있습니다.	테이블 구조 파일, Excel, XML
응용프로그램	응용프로그램 데이터 소스는 응용프로그램을 사용하여 기초적인 생명 과학 데이터에 액세스합니다. 원시 데이터는 수많은 표준 및 비표준 형식으로 이루어질 수 있습니다.	Documentum

절차:

DB2 Life Sciences Data Connect를 설치하려면 다음의 단계를 따르십시오.

1. DB2 Life Sciences Data Connect를 설치하기 전.
2. AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect를 설치하십시오.
3. Windows 서버에 DB2 Life Sciences Data Connect를 설치하십시오.
4. DB2 Life Sciences Data Connect를 설치한 후.

관련 태스크:

- 7 페이지의『DB2 Life Sciences Data Connect를 설치하기 전』
- 8 페이지의『AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치』
- 9 페이지의『Windows 서버에 DB2 Life Sciences Data Connect 설치』
- 11 페이지의『DB2 Life Sciences Data Connect를 설치한 후』
- 61 페이지의『CreateNicknameFile 유틸리티 설치(Documentum 랩퍼)』

DB2 Life Sciences Data Connect를 설치하기 전

이 태스크는 *DB2 Life Sciences Data Connect 설치*에 대한 기본 태스크의 일부입니다.

절차:

페더레이티드 서버에 DB2 Life Sciences Data Connect를 설치하기 전에 다음을 수행하십시오.

- DB2 Universal Database Enterprise Server Edition을 페더레이티드 서버에 설치했는지 확인하십시오.
- 데이터베이스가 페더레이티드 데이터베이스 시스템 지원을 설정했는지 확인하십시오. 이 설정을 점검하려면 DB2 명령행 처리기에서 다음의 명령을 실행하십시오.

  ```
  GET DATABASE MANAGER CONFIGURATION
  ```

  이 명령은 모든 데이터베이스 매개변수 및 현재 설정을 표시합니다.
  FEDERATED 매개변수가 예로 설정되었는지 확인하십시오.

  FEDERATED 매개변수가 NO로 설정된 경우, DB2 명령행 처리기에서 다음의 명령을 실행하십시오.

  ```
  UPDATE DATABASE MANAGER CONFIGURATION USING FEDERATED YES
  ```

이 태스크의 다음 태스크는 *AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치*입니다.

관련 태스크:

- 9 페이지의『Windows 서버에 DB2 Life Sciences Data Connect 설치』
- 11 페이지의『DB2 Life Sciences Data Connect를 설치한 후』
- 8 페이지의『AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치』

AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치

이 태스크는 *DB2 Life Sciences Data Connect 설치*에 대한 기본 태스크의 일부입니다.

전제조건:

아래의 관련 태스크 절에서 ″DB2 Life Sciences Data Connect를 설치하기 전″을 참조하십시오.

절차:

AIX, HP-UX, Linux 및 Solaris 운영 환경 페더레이티드 서버에 DB2 Life Sciences Data Connect를 설치하려면 db2setup 유틸리티를 사용하십시오.

주: db2setup 유틸리티 사용시 표시되는 화면은 페더레이티드 서버에 설치된 소프트웨어 제품에 의해 좌우됩니다. 이 단계에서는 DB2 Life Sciences Data Connect가 설치되어 있지 않다고 가정합니다.

1. 루트 권한을 가진 사용자로서 로그인하십시오.
2. DB2 Life Sciences Data Connect CD-ROM을 넣고 마운트하십시오. CD-ROM을 마운트하는 방법에 대한 정보는 *UNIX용 DB2 빠른 시작*을 참조하십시오.
3. **cd** */cdrom* 명령을 입력하여 CD-ROM이 마운트된 디렉토리로 변경하십시오. 여기서, *cdrom*은 사용자 제품 CD-ROM의 마운트 지점입니다.
4. 다음 명령을 입력하십시오.

   ```
   ./db2setup
   ```

   DB2 설치 창이 열립니다.
5. 설치 프로그램에서 프롬프트에 따르십시오.

   설치가 완료되면, DB2 Life Sciences Data Connect가 다른 DB2 제품과 함께 디렉토리에 설치됩니다.

   - AIX용 DB2 서버에서 디렉토리는 /usr/opt/db2_08_01입니다.
   - Solaris 운영 환경용 DB2 서버에서 디렉토리는 /opt/IBM/db2/V8.1입니다.

- HP-UX용 DB2 서버에서 디렉토리는 /opt/IBM/db2/V8.1입니다.
- Linux용 DB2 서버에서 디렉토리는 /opt/IBM/db2/V8.1입니다.

이 태스크의 다음 태스크는 *Windows 서버에 DB2 Life Sciences Data Connect 설치*입니다.

관련 태스크:

- 7 페이지의『DB2 Life Sciences Data Connect를 설치하기 전』
- 9 페이지의『Windows 서버에 DB2 Life Sciences Data Connect 설치』
- 11 페이지의『DB2 Life Sciences Data Connect를 설치한 후』

Windows 서버에 DB2 Life Sciences Data Connect 설치

이 태스크는 *DB2 Life Sciences Data Connect 설치*에 대한 기본 태스크의 일부입니다.

전제조건:

아래의 관련 태스크 절에서 ″DB2 Life Sciences Data Connect를 설치하기 전″을 참조하십시오.

절차:

Windows 페더레이티드 서버에 DB2 Life Sciences Data Connect를 설치하려면 설치 프로그램을 사용하십시오.

1. DB2 Universal Database 설치를 수행하기 위해 작성한 사용자 어카운트를 사용하여 페더레이티드 서버에 로그온하십시오.
2. 실행 중인 모든 프로그램을 종료하여 설치 프로그램이 필요시 파일을 갱신할 수 있게 하십시오.
3. 설치 프로그램을 호출하십시오. 자동 또는 수동으로 설치 프로그램을 호출할 수 있습니다. 설치 프로그램이 자동으로 시작하는 데 실패하거나, 다른 언어로 설치를 실행하려는 경우, 수동으로 설치 프로그램을 호출하십시오.

- 자동으로 설치 프로그램을 호출하려면, DB2 Life Sciences Data Connect CD를 드라이브에 넣으십시오. 자동 실행 기능이 자동으로 설치 프로그램을 시작합니다. 시스템 언어가 결정되며, 해당 언어의 설치 프로그램이 시작됩니다.
- 수동으로 설치 프로그램을 호출하려면 다음을 수행하십시오.
  a. **시작**을 누른 다음 **실행**을 누르십시오.
  b. **열기** 필드에 다음 명령을 입력하십시오.

     ```
     x:\setup /i language
     ```

     여기서,

     x: CD-ROM 드라이브를 나타냅니다.

     language
     사용 언어를 나타냅니다(예를 들어, 영어의 경우 EN).
  c. **확인**을 누르십시오.

  설치 런치패드를 여십시오.

4. **설치**를 눌러 설치 프로세스를 시작하십시오.
5. 설치 프로그램에서 프롬프트에 따르십시오.

   설치가 완료되면 DB2 Life Sciences Data Connect가 다른 DB2 제품과 함께 설치 디렉토리에 설치됩니다. 디폴트 설치 디렉토리는 C:\Program Files\IBM\SQLLIB입니다.

이 태스크의 다음 태스크는 *DB2 Life Sciences Data Connect를 설치한 후*입니다.

관련 태스크:

- 7 페이지의『DB2 Life Sciences Data Connect를 설치하기 전』
- 11 페이지의『DB2 Life Sciences Data Connect를 설치한 후』
- 8 페이지의『AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치』

DB2 Life Sciences Data Connect를 설치한 후

이 태스크는 *DB2 Life Sciences Data Connect 설치*에 대한 기본 태스크의 일부입니다. 설치 이후에, 랩퍼 라이브러리 파일은 사용자 시스템에 위치합니다. 이 라이브러리는 랩퍼 등록 프로세스 중 사용됩니다.

절차:

설치의 유효성을 확인하려면 설치 디렉토리에 디폴트 랩퍼 라이브러리가 있는지 점검하십시오.

지원되는 운영 체제별로 각 라이브러리에 대한 디폴트 파일 이름은 Windows 플랫폼의 경우, 표 3에 나열되어 있고 UNIX 플랫폼의 경우, 표 4에 나열되어 있습니다.

표 3. Windows 플랫폼의 디폴트 랩퍼 라이브러리 이름

랩퍼	Windows
테이블 구조 파일	db2lsfile.dll
Documentum	db2lsdctm.dll
Excel97 / Excel2000	db2lsxls.dll
BLAST	db2lsblast.dll
XML	db2lsxml.dll

표 4에는 지원되는 UNIX 플랫폼의 랩퍼 라이브러리 이름이 나열되어있습니다.

표 4. UNIX 플랫폼의 디폴트 랩퍼 라이브러리 이름

랩퍼	AIX	HP-UX	Linux	Solaris 운영 환경
테이블 구조 파일	libdb2lsfile.a	libdb2lsfile.sl	libdb2lsfile.so	libdb2lsfile.so
Documentum	libdb2lsdctm.a			libdb2lsdctm.so
BLAST	libdb2lsblast.a		libdb2lsblast.so	libdb2lsblast.so
XML	libdb2lsxml.a	libdb2lsxml.sl	libdb2lsxml.so	libdb2lsxml.so

주: Windows를 제외한 모든 플랫폼에 있는 Documentum의 경우, 이러한 라이브러리는 Documentum 클라이언트 라이브러리로 링크 편집된 후에 작성됩니

다. 설치 이후에 시스템에 저장된 파일의 이름은 AIX의 경우 `libdb2lsSTdctmF.a`이고 Solaris 운영 환경의 경우 `libdb2lsSTdctmF.so`입니다.

이 태스크 다음에는 더 이상의 태스크가 없습니다.

관련 태스크:

- 7 페이지의『DB2 Life Sciences Data Connect를 설치하기 전』
- 9 페이지의『Windows 서버에 DB2 Life Sciences Data Connect 설치』
- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』
- 8 페이지의『AIX, HP-UX, Linux 및 Solaris 운영 환경 서버에 DB2 Life Sciences Data Connect 설치』
- 33 페이지의『페더레이티드 시스템에 Documentum 추가』
- 75 페이지의『페더레이티드 시스템에 Excel 추가』
- 96 페이지의『페더레이티드 시스템에 BLAST 추가』
- 124 페이지의『페더레이티드 시스템에 XML 추가』

제 3 장 데이터 소스로서의 테이블 구조 파일

이 장에서는 테이블 구조 파일의 정의와 이를 사용자의 페더레이티드 시스템에 데이터 소스로 추가하는 방법에 대해 설명하고, 테이블 구조 파일 랩퍼와 연관된 오류 메시지를 나열합니다.

테이블 구조 파일의 정의

테이블 구조 파일은 일련의 레코드로 구성된 일반 구조이며 각 레코드에는 임의의 분리문자로 구분되는 동일한 수의 필드가 들어 있습니다. 널(NULL) 값은 양 옆에 2개의 분리문자로 표시됩니다.

다음 예는 DRUGDATA1.TXT 파일의 내용을 표시한 것입니다. 여기에는 세 개의 레코드가 들어 있으며 각 레코드는 세 개의 필드를 가지며 쉼표로 구분됩니다.

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

첫 번째 필드는 약의 고유한 ID 번호입니다. 두 번째 필드는 약의 이름입니다. 세 번째 필드는 약을 생산하는 제조업체의 이름입니다.

관련 개념:

- 14 페이지의『테이블 구조 파일 유형』
- 15 페이지의『테이블 구조 파일에서의 DB2 Life Sciences Data Connect 작업 방식』
- 31 페이지의『Documentum의 정의』
- 73 페이지의『Excel의 정의』
- 91 페이지의『BLAST의 정의』
- 119 페이지의『XML의 정의』

관련 태스크:

- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』

테이블 구조 파일 유형

테이블 구조 파일은 정렬되거나 정렬되지 않을 수 있습니다.

정렬된 파일

DRUGDATA1.TXT에는 정렬된 레코드가 들어 있습니다. 파일은 약의 고유한 ID 번호인 첫 번째 필드로 정렬됩니다. 이 필드는 각 약에 대해 고유하므로 기본 키입니다. 정렬된 파일은 오름차순으로 정렬되어야 합니다.

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

정렬되지 않은 파일

DRUGDATA2.TXT에는 정렬되지 않은 레코드가 들어 있습니다. 파일에서 레코드가 나열되는 방식에는 순서가 없습니다.

```
332,DrugnameB,Manufacturer2
234,DrugnameA,Manufacturer1
333,DrugnameC,Manufacturer2
```

랩퍼는 정렬되지 않은 파일보다 정렬된 데이터 파일에서 훨씬 더 효과적으로 검색을 수행할 수 있습니다.

관련 개념:

- 13 페이지의『테이블 구조 파일의 정의』
- 15 페이지의『테이블 구조 파일에서의 DB2 Life Sciences Data Connect 작업 방식』

관련 태스크:

- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』

테이블 구조 파일에서의 DB2 Life Sciences Data Connect 작업 방식

랩퍼라는 모듈 사용을 사용하면 DB2 Life Sciences Data Connect는 테이블 구조 파일에 있는 데이터를 쿼리하는 SQL문이 일반 관계 테이블 또는 뷰에 포함되어 있는 것처럼 처리할 수 있습니다. 이는 테이블 구조 파일에 있는 데이터를 관계 데이터나 다른 테이블 구조 파일에 있는 데이터에 조인시킵니다. 이 프로세스는 그림 3에 설명되어 있습니다.

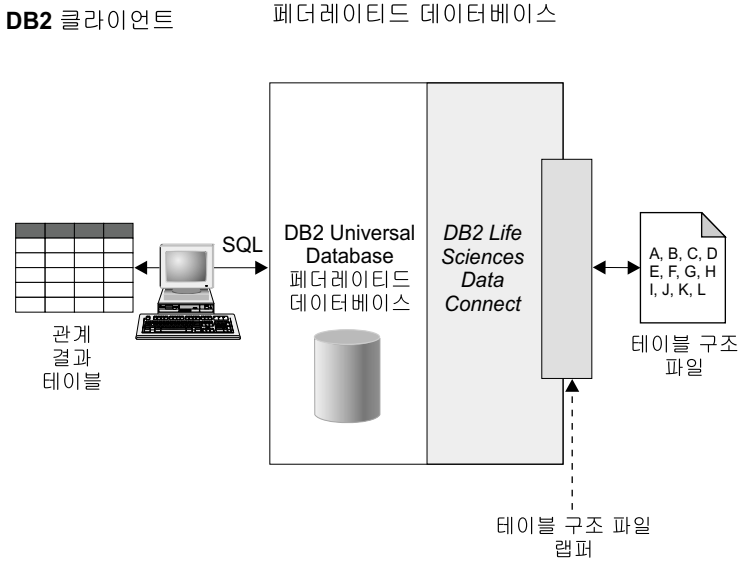


그림 3. 테이블 구조 파일 랩퍼 작업 방식

예를 들어, 테이블 구조 파일 DRUGDATA2.TXT가 연구실의 컴퓨터에 있다고 가정해보십시오. 이 데이터를 쿼리하여 다른 데이터 소스에 있는 다른 테이블과 일치시키려는 시도는 지루할 수 있습니다.

DRUGDATA2.TXT를 DB2 Life Sciences Data Connect를 사용하여 등록하면 파일은 관계 데이터 소스인 것처럼 작동합니다. 이제 다른 관계 및 비관계 데이터 소스와 함께 파일을 쿼리하고 함께 데이터를 분석할 수 있습니다.

예를 들어, 다음 쿼리를 실행할 수 있습니다.

```
SELECT * FROM DRUGDATA2 ORDER BY DCODE
```

이 쿼리는 다음과 같은 결과를 생성합니다.

코드	약	제조업체
234	DrugnameA	Manufacturer1
332	DrugnameB	Manufacturer2
333	DrugnameC	Manufacturer2

관련 개념:

- 13 페이지의『테이블 구조 파일의 정의』
- 14 페이지의『테이블 구조 파일 유형』

관련 태스크:

- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』

페더레이티드 시스템에 테이블 구조 파일 추가

절차:

테이블 구조 파일의 데이터 소스를 페더레이티드 서버에 추가하려면 다음을 수행하십시오.

1. CREATE WRAPPER 명령을 사용하여 랩퍼를 등록하십시오.
2. 선택적: 쿼리 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오.
3. CREATE SERVER 명령을 사용하여 서버를 등록하십시오.
4. 모든 테이블 구조 파일에 대해 CREATE NICKNAME 명령을 사용하여 별칭을 등록하십시오.

이 명령은 DB2 명령행 처리기에서 실행될 수 있습니다.

관련 태스크:

- 17 페이지의『테이블 구조 파일 랩퍼 등록』
- 18 페이지의『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 33 페이지의『페더레이티드 시스템에 Documentum 추가』
- 75 페이지의『페더레이티드 시스템에 Excel 추가』
- 96 페이지의『페더레이티드 시스템에 BLAST 추가』
- 124 페이지의『페더레이티드 시스템에 XML 추가』

테이블 구조 파일 랩퍼 등록

이 태스크는 *페더레이티드 시스템에 테이블 구조 파일 추가*에 대한 기본 태스크의 일부입니다. 데이터 소스에 액세스하려면 랩퍼를 등록해야 합니다. 랩퍼는 페더레이티드 서버가 통신을 하고 데이터 소스에서 데이터를 검색하는 데 사용하는 메커니즘입니다. 랩퍼는 라이브러리 파일로 시스템에 설치됩니다.

절차:

랩퍼를 등록하려면 CREATE WRAPPER문을 사용하여 테이블 구조 파일에 액세스하기 위해 사용할 랩퍼를 지정하십시오.

예를 들어, AIX에서 랩퍼를 등록하려면, 다음 명령문을 실행하십시오.

```
CREATE WRAPPER laboratory_flat_files LIBRARY 'libdb2lsfile.a'
OPTIONS(DB2_FENCED 'N');
```

이 예에서 `laboratory_flat_files`는 랩퍼용으로 선택된 이름입니다. 이 이름은 등록 중인 랩퍼의 데이터베이스 내에서 고유해야 합니다. AIX에서 테이블 구조 파일 랩퍼에 대한 필수 라이브러리 이름은 `libdb2lsfile.a`입니다.

라이브러리 이름은 디폴트로 `libdb2lsfile.a`로 설치되지만 설치 중에 사용자 정의되었을 수 있습니다. 올바른 이름은 시스템 관리자에게 문의하십시오.

지원되는 플랫폼별로 테이블 구조 파일 랩퍼에 대한 디폴트 라이브러리 파일 이름의 표는 아래의 관련 태스크 절에서 ″DB2 Life Sciences Data Connect를 설치한 후″를 참조하십시오. CREATE WRAPPER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정*입니다.

관련 태스크:

- 18 페이지의『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 38 페이지의『Documentum 랩퍼 등록』
- 76 페이지의『Excel 랩퍼 등록』
- 102 페이지의『BLAST 랩퍼 등록』
- 125 페이지의『XML 랩퍼 등록』

테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정

이 태스크는 *페더레이티드 시스템에 테이블 구조 파일 추가*에 대한 기본 태스크의 일부입니다. 테이블 구조 파일에 액세스할 때 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오. 이 변수는 페더레이티드 서버가 초기화 중에 랩퍼를 로드하는지 여부를 결정합니다.

절차:

DB2_DJ_COMM 환경 변수를 설정하려면 연관된 CREATE WRAPPER문에 지정한 랩퍼에 해당하는 랩퍼 라이브러리와 함께 `db2set` 명령을 제출하십시오.

예를 들어, 다음과 같습니다.

```
db2set DB2_DJ_COMM='libdb2lsfile.a'
```

등호(=)의 양쪽에 스페이스가 없게 하십시오.

데이터베이스 시작 중 랩퍼 라이브러리 로드와 연관된 오버헤드가 있습니다. 이 오버헤드를 방지하려면 액세스하려는 라이브러리만을 지정하십시오.

DB2_DJ_COMM 환경 변수에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

이 태스크의 다음 태스크는 *테이블 구조 파일에 대한 서버* 등록입니다.

관련 태스크:

- 17 페이지의『테이블 구조 파일 랩퍼 등록』
- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 39 페이지의『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 103 페이지의『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 126 페이지의『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

테이블 구조 파일에 대한 서버 등록

이 태스크는 *페더레이티드 시스템에 테이블 구조 파일 추가*에 대한 기본 태스크의 일부입니다. 랩퍼를 등록한 후에 해당 서버를 등록해야 합니다.

절차:

테이블 구조 파일 서버를 페더레이티드 시스템에 등록하려면 CREATE SERVER 문을 사용하십시오. 예를 들어, 다음과 같습니다.

```
CREATE SERVER biochem_lab WRAPPER laboratory_flat_files
```

이 예에서 `biochem_lab`은 테이블 구조 파일 서버에 지정된 이름입니다. 이름은 등록 중인 서버의 데이터베이스에 고유해야 합니다.

CREATE SERVER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *테이블 구조 파일에 대한 별칭 등록*입니다.

관련 태스크:

- 18 페이지의『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』
- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』
- 127 페이지의『XML 데이터 소스에 대한 서버 등록』

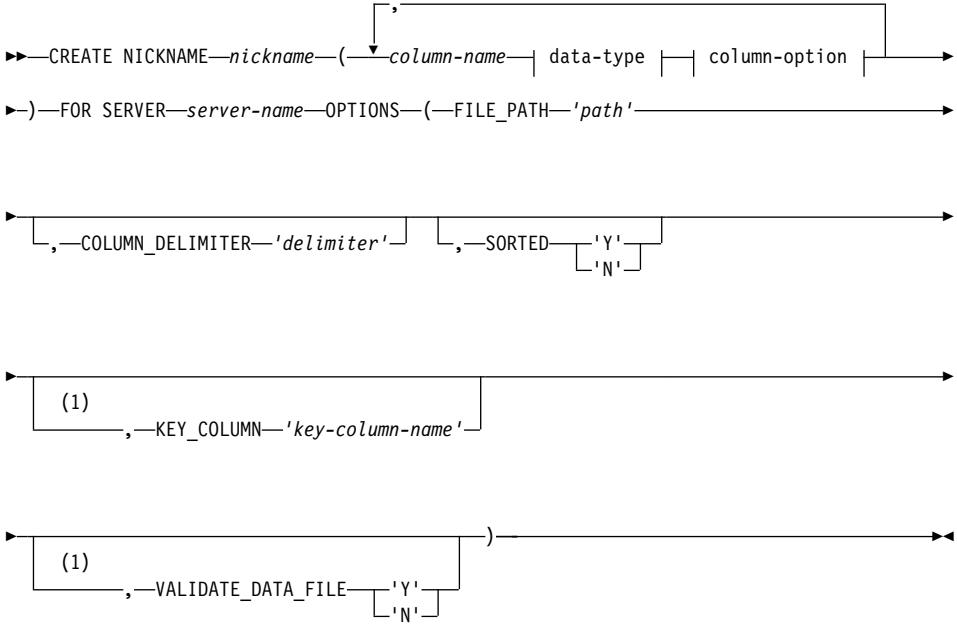
테이블 구조 파일에 대한 별칭 등록

이 태스크는 *페더레이티드 시스템에 테이블 구조 파일 추가*에 대한 기본 태스크의 일부입니다. 서버를 등록한 후에 해당 별칭을 등록해야 합니다. 별칭은 쿼리에서 테이블 구조 파일 데이터를 참조할 때 사용됩니다.

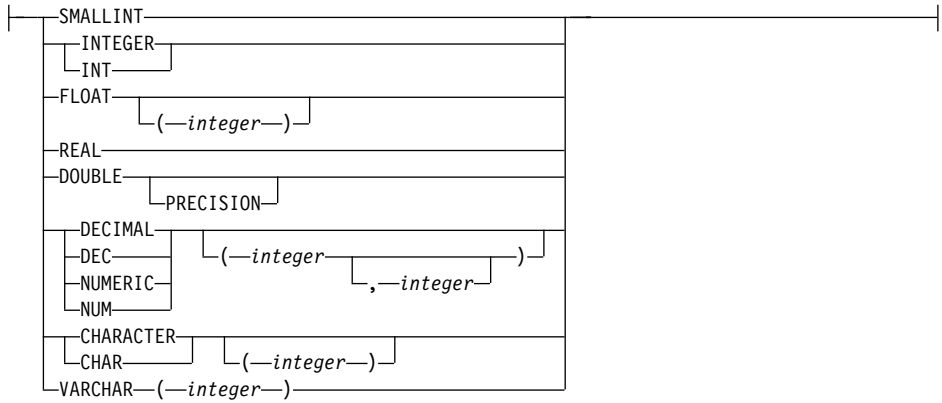
절차:

별칭을 등록하려면 액세스하려는 각 테이블 구조 파일에 대해 CREATE NICKNAME문을 사용하십시오.

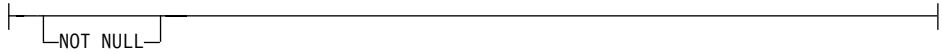
CREATE NICKNAME문의 구문은 다음과 같습니다.



data-type:



column-option:



주:

1 정렬되지 않은 파일의 경우에는 허용되지 않습니다. 정렬된 파일의 경우 선택적입니다.

nickname

액세스되는 테이블 구조 파일의 고유한 별칭. 등록 중인 스키마의 다른 모든 별칭, 테이블 및 뷰와 구별되어야 합니다.

column-name

테이블 구조 파일의 각 필드에 제공된 고유한 이름. 각 컬럼 이름 다음에는 데이터 유형이 옵니다. 유형이 CHAR, VARCHAR, SMALLINT, INTEGER, FLOAT, DOUBLE, REAL 및 DECIMAL인 컬럼만이 지원됩니다.

SMALLINT

작은 정수에 사용됩니다.

INTEGER 또는 INT

큰 정수에 사용됩니다.

FLOAT(*integer*)

*integer*의 값에 따라 단일 또는 배정밀도 부동 소수점 숫자에 사용됩니다.

*integer*의 값은 1 - 53 사이에 있어야 합니다. 1 - 24 값은 단정밀도를 나타내며 25 - 53은 배정밀도를 나타냅니다.

REAL

단정밀도 부동 소수점에 사용됩니다.

DOUBLE 또는 DOUBLE PRECISION

배정밀도 부동 소수점에 사용됩니다.

FLOAT

배정밀도 부동 소수점에 사용됩니다.

DECIMAL(*precision-integer, scale-integer***) 또는 DEC(***precision-integer, scale-integer***)**

10진수에 사용됩니다.

첫 번째 정수는 숫자의 정밀도로 총 자리수를 나타냅니다. 이 값의 범위는 1 - 31입니다.

두 번째 정수는 숫자의 스케일입니다. 즉, 소수점의 오른쪽에 있는 자리수입니다. 이 값의 범위는 0부터 숫자의 정밀도까지입니다.

정밀도 및 스케일이 지정되지 않은 경우 디폴트값인 5,0이 사용됩니다.

단어 **NUMERIC** 및 **NUM**은 **DECIMAL** 및 **DEC**와 동의어로서 사용될 수 있습니다.

CHARACTER(*integer***) 또는 CHAR(***integer***) 또는 CHARACTER 또는 CHAR**

고정 길이를 가진 *integer*에 사용되며 범위는 1 - 254입니다. 길이 스펙이 누락되면 길이는 1로 간주됩니다.

VARCHAR(*integer***)**

가변 길이를 가진 *integer*에 사용되며 범위는 1 - 32767입니다.

NOT NULL

컬럼에 널(Null) 값이 들어 있지 않게 합니다.

server-name

CREATE SERVER문을 사용하여 등록한 서버를 식별합니다. CREATE SERVER문에 대한 자세한 정보는 아래의 관련 링크 절을 참조하십시오. 이 서버는 테이블 구조 파일을 액세스하는 데 사용됩니다.

'path' 작은따옴표로 묶여 액세스되는 테이블 구조 파일로의 완전한 경로. 데이터 파일은 파이프나 다른 비표준 파일 유형보다는 표준 파일이나 기호 링크이어야 합니다. 데이터 파일은 DB2 인스턴스 소유자가 읽을 수 있어야 합니다. 인스턴스 소유자에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

SORTED

데이터 소스 파일의 정렬 여부를 지정합니다. 이 옵션은 'Y', 'y', 'n' 또는 'N' 중 하나를 승인합니다. 디폴트값은 'N'입니다.

주: 데이터 소스를 정렬된 것으로 지정한 경우,
VALIDATE_DATA_FILE을 'Y'로 설정하는 것이 바람직합니다.

'delimiter'

테이블 구조 파일의 컬럼을 구분하기 위해 사용되는 분리문자로 작은따옴표로 묶입니다. 한 문자의 분리문자만이 허용됩니다. 컬럼 분리문자가 정의되지 않으면 컬럼 분리문자의 디폴트로 쉼표가 사용됩니다. 작은따옴표를 분리문자로 사용할 수 없습니다. 컬럼 분리문자는 컬럼에 대해 유효한 데이터로서 존재할 수 없습니다. 예를 들어, 컬럼 중 하나에 쉼표가 삽입된 데이터가 들어 있는 경우 컬럼 분리문자로 쉼표를 사용할 수 없습니다.

'key-column-name'

정렬된 파일에서 키를 형성하는 파일에서의 컬럼의 이름으로, 작은따옴표로 묶입니다. 정렬된 파일에만 이 옵션을 사용하십시오.

단일 컬럼 키만이 지원됩니다. 값은 CREATE NICKNAME문에 정의된 컬럼의 이름이어야 합니다. 컬럼은 오름차순으로 정렬되어야 합니다. 정렬된 별칭에 대해 값이 지정되지 않은 경우, 별칭이 붙은 파일의 첫 번째 컬럼을 디폴트로 사용합니다. NOT NULL 옵션을 별칭 명령문의 정의에 추가하여 키 컬럼을 널(NULL) 입력 가능하지 않은 것으로 지정하는 것이 바람직합니다. 예를 들어, 다음과 같습니다.

```
CREATE NICKNAME tox (tox_id INTEGER NOT NULL, toxicity VARCHAR(100))
FOR SERVER tox_server1
OPTIONS (FILE_PATH'/tox_data.txt', SORTED 'Y')

CREATE NICKNAME weights (mol_id INTEGER, wt VARCHAR(100) NOT NULL)
FOR SERVER wt_server
OPTIONS (FILE_PATH'/wt_data.txt', SORTED 'Y', KEY_COLUMN 'WT')
```

주: 이 옵션은 대소문자를 구분합니다. 그러나 DB2는 컬럼이 큰따옴표를 사용하여 정의되지 않았으면 컬럼 이름을 대문자로 접습니다. 다음의 예에서 empno 컬럼은 DB2의 의해 대문자로 바뀌고 empno 키 컬럼은 소문자로 제출되기 때문에 올바로 작동되지 않습니다. 따라서 키로 지정된 컬럼을 찾을 수 없습니다.

```
CREATE NICKNAME depart (
empno char(6) NOT NULL)
FOR SERVER DATASTORE
OPTIONS(FILE_PATH'data.txt', SORTED 'Y', KEY_COLUMN 'empno');
```

VALIDATE_DATA_FILE

정렬된 파일의 경우, 이 옵션은 랩퍼가 키 컬럼이 오름차순으로 정렬되었음을 검증하고 널(NULL) 키를 점검하는지의 여부를 지정합니다. 이 옵션에 유효한 값은 'Y' 또는 'N'으로, 작은따옴표로 묶입니다. 등록시에만 값이 점검됩니다. 이 옵션이 지정되지 않은 경우 유효성 확인은 발생하지 않습니다.

다음의 예는 아래의 관련 링크 절에 나열된 "테이블 구조 파일의 정의"에 설명된 테이블 구조 파일 DRUGDATA1.TXT에 대한 CREATE NICKNAME문을 나타냅니다.

```
CREATE NICKNAME DRUGDATA1(Dcode Integer NOT NULL, Drug CHAR(20),
Manufacturer CHAR(20))
FOR SERVER biochem_lab OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT',
COLUMN_DELIMITER ',', SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y')
```

CREATE NICKNAME문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오. 별칭에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

이 태스크 다음에는 더 이상의 태스크가 없습니다.

관련 태스크:

- 11 페이지의『DB2 Life Sciences Data Connect를 설치한 후』
- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』

- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』
- 145 페이지의 제 8 장『비용산정 별칭 옵션 지정』

테이블 구조 파일 랩퍼에 대한 랩퍼 제한사항 및 고려사항

- Passthru 세션은 랩퍼를 사용할 때 허용되지 않습니다.
- 여러 컬럼으로 구성된 키는 허용되지 않습니다.
- 파일은 오름차순으로만 정렬되어야 합니다. 내림차순에서 정렬은 지원되지 않습니다.
- 랩퍼는 NOT NULL 제한조건을 강제로 적용하지 않지만 DB2는 강제로 적용합니다. 별칭을 작성하고 컬럼에 NOT NULL 제한조건에 첨부하여 해당 컬럼에 대해 널(NULL) 값을 갖는 행을 선택한 경우 DB2는 널(NULL) 값을 NOT NULL 컬럼에 지정할 수 없음을 나타내는 SQL0407N 오류가 발생합니다.

  이 규칙에 대한 예외는 정렬된 별칭의 경우입니다. 정렬된 별칭의 키 컬럼은 널(NULL)이 될 수 없습니다. 정렬된 별칭에 대해 널(NULL) 키 컬럼이 발견된 경우 키 컬럼이 누락되었음을 나타내는 SQL1822N 오류가 발생합니다.
- DB2 Universal Database Enterprise Server Edition에서 별칭이 작성된 테이블 구조 파일은 각 노드의 동일한 경로 이름을 사용하여 액세스할 수 있어야 합니다. 일반 경로를 갖는 모든 노드로부터 액세스할 수 있는한 파일은 DB2 Universal Database 노드에 있을 필요가 없습니다.

관련 참조:

- 25 페이지의『테이블 구조 파일 랩퍼에 대한 파일 제한사항 및 고려사항』
- 65 페이지의『Documentum 랩퍼에 대한 제한사항 및 고려사항』
- 83 페이지의『Excel 랩퍼에 대한 랩퍼 제한사항』
- 83 페이지의『Excel 파일 제한사항』
- 138 페이지의『XML 랩퍼에 대한 제한사항 및 고려사항』

테이블 구조 파일 랩퍼에 대한 파일 제한사항 및 고려사항

- 파일은 행 당 하나의 레코드로 제한됩니다.
- 각 레코드는 동일한 수의 구분된 컬럼을 가져야 합니다.

- 각 레코드는 랩퍼가 설치된 플랫폼에 대한 표준 행 종료 문자로 종료되어야 합니다.
- 컬럼 분리문자는 파일에서 일관성이 있어야 합니다.
- 널(Null) 값은 양 옆에 있는 두 개의 분리문자로 표시되거나, 널(Null)필드가 행의 마지막 필드인 경우 행 종료자가 뒤에 오는 분리문자로 표시됩니다.
- 기수(radix) 문자는 LC_NUMERIC 자국어 지원(NLS) 범주의 RADIXCHAR 항목에 의해 결정됩니다.
- 정렬된 데이터 소스는 LC_COLLATE 자국어 지원(NLS) 범주에 있는 설정에 의해 정의된 현재 로케일의 대조 시퀀스에 따라 오름차순으로 정렬되어야 합니다.
- 데이터베이스 코드 페이지는 파일의 문자 세트와 일치해야 하며, 그렇지 않으면 예상치 못한 결과를 가져올 수 있습니다.
- 다중 바이트 문자가 들어 있는 파일은 지원되지 않습니다.
- 컬럼 유형에 비해 숫자 이외의 필드가 너무 긴 경우, 초과 데이터는 절단됩니다.
- 파일의 10진수 필드에 컬럼 유형의 스케일 매개변수에 의해 허용되는 것보다 더 많은 자리수가 기수(radix) 문자 다음에 오는 경우, 초과 데이터는 절단됩니다.
- 최대 행 길이는 32768입니다.

관련 참조:

- 25 페이지의『테이블 구조 파일 랩퍼에 대한 랩퍼 제한사항 및 고려사항』
- 65 페이지의『Documentum 랩퍼에 대한 제한사항 및 고려사항』
- 83 페이지의『Excel 파일 제한사항』
- 138 페이지의『XML 랩퍼에 대한 제한사항 및 고려사항』

테이블 구조 파일 랩퍼에 대한 파일 액세스 제어 모델

데이터베이스 관리 시스템은 DB2 인스턴스 소유자의 권한을 사용하여 테이블 구조 파일에 액세스합니다. 랩퍼는 이 사용자 ID(또는 그룹 ID)로 읽을 수 있는 파일에만 액세스할 수 있습니다. 응용프로그램의 권한 부여 ID(페더레이티드 데이터베이스에 대한 연결을 설정하는 ID)는 관련되어 있지 않습니다.

관련 참조:

- 66 페이지의『Documentum 랩퍼에 대한 액세스 제어』
- 84 페이지의『Excel 랩퍼에 대한 파일 액세스 제어 모델』

테이블 구조 파일 랩퍼에 대한 최적화 추가 정보 및 고려사항

- 시스템은 정렬되지 않은 파일보다 정렬된 데이터 파일에서 훨씬 더 효과적으로 검색을 수행할 수 있습니다.
- 정렬된 파일의 경우 쿼리를 제출할 때 키 컬럼에 대한 값이나 범위를 지정하여 성능을 개선할 수 있습니다.
- 테이블 구조 파일의 별칭에 대한 통계는 SYSSTAT 및 SYSCAT 뷰를 갱신하여 수동으로 갱신해야 합니다. SYSSTAT 및 SYSCAT 뷰의 수동 갱신에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

관련 참조:

- 116 페이지의『BLAST 랩퍼에 대한 최적화 추가 정보』

테이블 구조 파일 랩퍼에 대한 메시지

이 절에서는 테이블 구조 파일의 랩퍼로 작업하는 중에 발생할 수 있는 메시지 목록을 나열하고 설명합니다. 메시지에 대한 자세한 정보는 *DB2 메시지 참조서*를 참조하십시오.

표 5. 테이블 구조 파일의 랩퍼가 발행한 메시지

오류 코드	메시지	설명
SQL0405N	숫자 리터럴 "<literal>"은 범위를 벗어나므로 유효하지 않습니다.	데이터 파일에 있는 컬럼 또는 SQL문에 있는 술어 값에 해당 데이터 유형에 대해 가능한 범위를 벗어난 값이 들어 있습니다. 데이터 파일을 정정하거나 좀 더 적합한 유형으로 컬럼을 재정의하십시오.
SQL0408N	값이 관련 지정 목표의 데이터 유형과 호환 가능하지 않습니다. 목표 이름은 "<column_name>"입니다.	데이터 파일의 컬럼에 해당 데이터 유형에 유효하지 않은 문자가 들어 있습니다. 데이터 파일을 정정하거나 좀 더 적합한 유형으로 컬럼을 재정의하십시오.

표 5. 테이블 구조 파일의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL문은 처리될 수 있습니다(이유: ″데이터 소스 경로가 널(NULL)임″).	IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL문은 처리될 수 있습니다(이유: ″키 컬럼 검색 실패″).	IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL문은 처리될 수 있습니다(이유: ″STAT가 데이터 소스에서 실패했음. 오류 번호 = <error_number>″).	적절한 디렉토리 권한을 가지고 있는지 확인하십시오. 파일이 있는지 확인하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL문은 처리될 수 있습니다(이유: ″컬럼 정보가 없음″).	IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL문은 처리될 수 있습니다(이유: ″지원되지 않는 연산자″).	IBM 고객만족센터에 문의하십시오.
SQL1816N	랩퍼 ″<wrapper_name>″은 페더레이티드 데이터베이스에 정의하려고 시도 중인 데이터 소스의 ″유형″(″<type>″ ″″)에 액세스하는 데 사용될 수 없습니다.	서버 유형이 유효하지 않습니다. CREATE SERVER문에 서버 유형을 지정하지 말아야 합니다. TYPE 키워드 및 값을 제거하고 다시 실행하십시오.
SQL1822N	데이터 소스 ″<server_name>″에서 예상치 못한 오류 코드 ″ERRNO = <error_number>″를 받았습니다. 연관된 텍스트 및 토큰은 ″파일을 읽을 수 없습니다.″입니다.	오류 번호의 값을 확인하십시오. DB2 인스턴스 소유자가 파일을 읽을 수 있는지 확인하십시오. 그런 다음 SQL문을 다시 실행하십시오.

표 5. 테이블 구조 파일의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 "<server_name>"에서 예상치 못한 오류 코드 "데이터 오류"를 받았습니다. 연관된 텍스트 및 토큰은 "데이터 소스가 비표준 파일입니다."입니다.	데이터 소스 파일은 디렉토리, 소켓 또는 FIFO입니다. 표준 파일만이 데이터 소스로서 액세스될 수 있습니다. 유효한 파일을 지정하도록 FILE_PATH 옵션을 변경한 후 SQL 명령을 재발행하십시오.
SQL1822N	데이터 소스 "<server_name>"에서 예상치 못한 오류 코드 "ERRNO = <error_number>"를 받았습니다. 연관된 텍스트 및 토큰은 "파일 열기 오류"입니다.	랩퍼가 파일을 열 수 없습니다. 오류가 발생한 이유를 판별하려면 오류 번호를 확인하십시오. 데이터 소스에서 문제점을 정정하고 SQL 명령을 재발행하십시오.
SQL1822N	데이터 소스 "<server_name>"에서 예상치 못한 오류 코드 "데이터 오류"를 받았습니다. 연관된 텍스트 및 토큰은 "키 컬럼 누락"입니다.	데이터 소스에서 검색된 레코드에 키 필드가 누락되었습니다. 키 컬럼은 널(NULL)일 수 없습니다. 데이터를 정정하거나 정렬되지 않은 별칭으로 파일을 등록하십시오.
SQL1822N	데이터 소스 "<server_name>"에서 예상치 못한 오류 코드 "데이터 오류"를 받았습니다. 연관된 텍스트 및 토큰은 "파일이 정렬되어 있지 않습니다."입니다.	파일이 키 컬럼으로 정렬되지 않았습니다. 다음 중 하나를 수행하십시오. 올바른 컬럼을 지정하도록 KEY_COLUMN 옵션을 변경하거나 데이터 파일을 다시 정렬하거나 정렬되지 않은 별칭으로 별칭을 등록하십시오.
SQL1822N	데이터 소스 "<server_name>"에서 예상치 못한 오류 코드 "데이터 오류"를 받았습니다. 연관된 텍스트 및 토큰은 "키가 정의 크기를 초과합니다."입니다.	데이터 소스에서 읽은 키 컬럼 필드가 DB2 컬럼 정의보다 크면 랩퍼 검색 루틴이 제대로 작동하지 못할 수 있습니다. 데이터를 정정하거나 별칭 정의를 정정하고 별칭을 다시 등록하십시오.
SQL1822N	데이터 소스 "<server_name>"에서 예상치 못한 오류 코드 "데이터 오류"를 받았습니다. 연관된 텍스트 및 토큰은 "데이터 파일의 행이 32K를 초과합니다."입니다.	데이터 파일의 행이 랩퍼가 허용하는 최대 행 길이를 초과했습니다. 행 길이는 32768보다 클 수 없습니다. 데이터 파일에서 행의 길이를 줄이십시오.
SQL1823N	서버 "<server_name>"의 데이터 유형 "<data_type>"에 대한 데이터 유형 맵핑이 존재하지 않습니다.	지원되지 않는 데이터 유형을 가진 별칭이 정의되었습니다. 지원되는 데이터 유형만을 사용하여 별칭을 재정의하십시오.
SQL1881N	"<option_name>"은 "<object_name>"에 대해 유효한 "<component>" 옵션이 아닙니다.	나열된 값은 나열된 오브젝트의 유효한 옵션이 아닙니다. 유효하지 않은 옵션을 제거하거나 변경한 다음 SQL문을 다시 제출하십시오.

표 5. 테이블 구조 파일의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1882N	″Nickname″ 옵션 ″COLUMN_DELIMITER″는 ″<nickname_name>″의 ″<delimiter>″로 설정될 수 없습니다.	컬럼 분리문자의 길이가 한 자를 초과합니다. 옵션을 한 자로 재정의하십시오. 그런 다음 SQL 명령을 다시 실행하십시오.
SQL1882N	″Nickname″ 옵션 ″KEY_COLUMN″은 ″<nickname_name>″의 ″<column_name>″으로 설정될 수 없습니다.	키 컬럼으로 선택된 컬럼이 이 별칭에 대해 정의되어 있지 않습니다. KEY_COLUMN 옵션을 이 별칭의 정렬된 컬럼 중 하나로 정정한 다음 SQL 명령을 재발행하십시오.
SQL1882N	″Nickname″ 옵션 ″VALIDATE_DATA_FILE″은 ″<nickname_name>″의 ″<option_value>″로 설정될 수 없습니다.	옵션 값이 유효하지 않습니다. 유효한 값은 ″Y″ 또는 ″N″입니다. 옵션을 정정하고 별칭을 다시 등록하십시오.
SQL1883N	″<option_name>″은 ″<object_name>″의 필수 ″<component>″ 옵션입니다.	SQL문에 랩퍼에 대한 필수 옵션이 누락되었습니다. 필수 옵션을 추가하고 SQL문을 다시 제출하십시오.
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = ″21″	Passthru 세션을 시도했습니다. 테이블 구조 파일 랩퍼가 passthru 세션을 지원하지 않습니다.

관련 참조:

- 67 페이지의『Documentum 랩퍼에 대한 메시지』
- 84 페이지의『Excel 랩퍼에 대한 메시지』
- 116 페이지의『BLAST 랩퍼에 대한 메시지』
- 138 페이지의『XML 랩퍼에 대한 메시지』

제 4 장 데이터 소스로서의 Documentum

이 장에서는 Documentum의 정의와 사용자의 페더레이티드 시스템에 Documentum 데이터 소스를 추가하는 방법을 설명하고, Documentum 랩퍼와 연관된 오류 메시지들을 나열합니다.

Documentum의 정의

Documentum은 체크인, 체크아웃, 작업흐름 및 버전 관리와 같은 문서 내용 및 속성의 관리를 제공하는 문서 관리 소프트웨어입니다. Documentum 제품은 관계형 데이터베이스의 맨 위에 빌드된 3-티어 클라이언트-서버 시스템입니다.

Docbase는 문서 내용, 속성, 관계, 버전, 해석, 형식, 작업흐름 및 보안을 저장하는 Documentum 저장소입니다. DQL(Documentum Query Language) 확장 SQL 다이얼렉트(dialect)가 Documentum 데이터를 쿼리하는 데 사용됩니다. Docbase는 Oracle 인스턴스 또는 DB2® 데이터베이스에 문서 내용 파일을 더한 것과 같습니다. 메타데이터는 기본 관계형 데이터베이스 관리 시스템(RDBMS)에 저장되며, 내용은 데이터베이스의 2진 대형 오브젝트(BLOB)로 또는 서버 시스템의 파일 시스템 내에 저장된 파일로 저장됩니다. Documentum에 대한 자세한 정보는 Documentum 매뉴얼을 참조하십시오.

Documentum의 랩퍼는 Documentum 데이터 소스를 DB2 페더레이티드 시스템로 추가하게 합니다. Documentum 데이터 소스를 페더레이티드 시스템에 추가함으로써 SQL문을 사용하여 오브젝트 및 등록된 테이블을 Documentum Docbase에서 액세스하고 쿼리할 수 있습니다. 그런 다음 원시(native) 데이터 소스로부터 데이터를 이동하지 않고 페더레이티드 시스템에서 다른 데이터 소스데이터 소스와 이 데이터를 통합할 수 있습니다. Documentum 랩퍼는 Documentum 서버와 인터페이스하기 위해 클라이언트 라이브러리를 사용합니다. Documentum 랩퍼는 Documentum 서버의 EDMS 98(버전 3으로도 나타냄) 및 4i의 두 가지 버전에

대한 액세스를 제공합니다. 그림 4에서는 Documentum 랩퍼 작업 방법을 설명합니다.

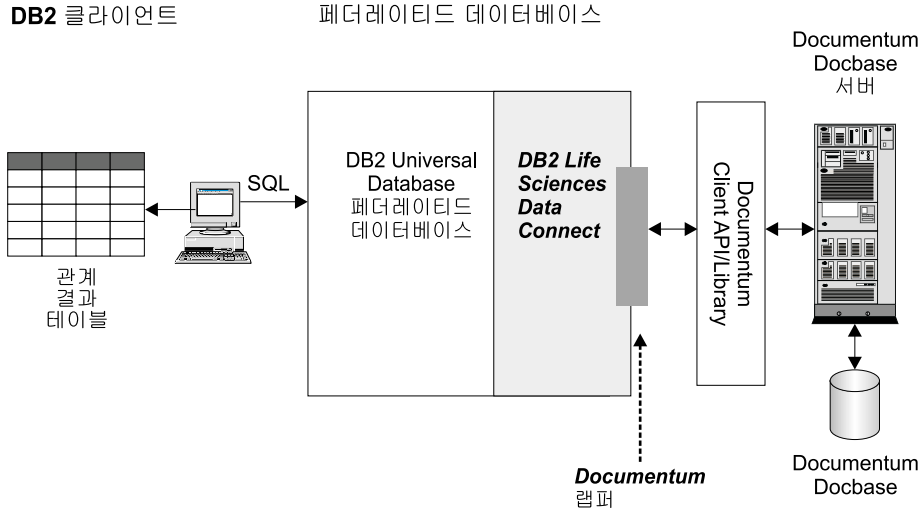


그림 4. Documentum 랩퍼 작업 방법

Documentum 랩퍼를 등록한 후에, Documentum Docbase 오브젝트 및 등록된 테이블을 관계 테이블로 맵핑할 수 있습니다. 이것은 Docbase 속성을 DB2 관계형 테이블의 컬럼 이름으로 맵핑하여 수행됩니다.

예를 들어, 표 6에서는 연관된 데이터와 함께 Documentum Docbase 디폴트 문서 유형, dm_document에 대한 속성의 서브세트를 나열합니다. 이 속성의 부속 집합이 중요하며, 이 속성을 페더레이티드 데이터베이스 시스템으로 연결하려고 함을 결정했습니다. 이 데이터 서브세트의 이름을 DrugAB_data라고 지정했습니다.

표 6. DrugAB_data

제목	주제	작성자	키워드
The effect of drug A on rabbits	Drug A	Curran, L.	rabbits, drug A
Toxicity results for drug A	Drug A	Abelite, P., McMurtrey, K.	toxicity, drug A
Drug B interactions	Drug B	DeNiro, R., Stone, S.	interactions, drug B
Chemical structure of drug B	Drug B	Boyslim, F.	structure, drug B

Documentum 랩퍼를 등록한 후에, SQL문을 사용하여 데이터를 쿼리할 수 있습니다.

다음 쿼리는 주제가 Drug A인 제목과 작성자를 표시합니다. 결과 테이블은 표 7에 표시됩니다.

```
SELECT title, authors
FROM drugAB_data
WHERE subject = 'Drug A'
```

표 7. 쿼리 결과

제목	작성자
The effect of drug A on rabbits	Curran, L.
Toxicity results for drug A	Abelite, P., McMurtrey, K.

관련 개념:

- 13 페이지의『테이블 구조 파일의 정의』
- 73 페이지의『Excel의 정의』
- 91 페이지의『BLAST의 정의』
- 119 페이지의『XML의 정의』

관련 태스크:

- 33 페이지의『페더레이티드 시스템에 Documentum 추가』

페더레이티드 시스템에 Documentum 추가

절차:

페더레이티드 서버에 Documentum 데이터 소스를 추가하려면 다음을 수행하십시오.

1. Documentum 클라이언트 라이브러리에 링크하십시오.
2. Documentum의 클라이언트 dmcl.ini 파일을 지정하십시오.
3. CREATE WRAPPER문을 사용하여 랩퍼를 등록하십시오.
4. 선택적: 쿼리 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오.
5. CREATE SERVER문을 사용하여 서버를 등록하십시오.

6. CREATE USER MAPPING문을 사용하여 데이터 소스에 대한 액세스 권한을 사용자에게 부여하십시오.
7. CREATE NICKNAME문을 사용하여 별칭을 등록하십시오.
8. CREATE FUNCTION문을 사용하여 사용자 정의 기능을 작성하십시오.

이 명령문은 DB2 명령행 처리기에서 실행될 수 있습니다. 일단 등록되면, 데이터 소스에 대한 쿼리를 실행할 수 있습니다.

관련 태스크:

- 35 페이지의『Documentum 클라이언트 라이브러리에 링크(AIX 및 Solaris 운영 환경에만 해당)』
- 36 페이지의『Documentum의 클라이언트 dmcl.ini 파일 지정』
- 38 페이지의『Documentum 랩퍼 등록』
- 39 페이지의『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 42 페이지의『사용자 맵핑(Documentum 랩퍼)』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』
- 52 페이지의『Documentum 데이터 소스에 대한 사용자 정의 기능 등록』
- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』
- 75 페이지의『페더레이티드 시스템에 Excel 추가』
- 96 페이지의『페더레이티드 시스템에 BLAST 추가』
- 124 페이지의『페더레이티드 시스템에 XML 추가』

Documentum 클라이언트 라이브러리에 링크(AIX 및 Solaris 운영 환경에만 해당)

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. Documentum 데이터 소스에 대한 액세스를 가능하게 하려면, DB2 페더레이티드 시스템이 클라이언트 라이브러리로 링크 편집되어야 합니다. 링크 편집 프로세스는 페더레이티드 서버가 통신하는 각 데이터 소스에 대한 랩퍼 라이브러리를 작성합니다. djxlinkDctm 스크립트를 실행할 때 Documentum 랩퍼 라이브러리를 작성합니다.

절차:

djxlinkDctm 스크립트를 실행하려면 다음을 실행 하십시오.

1. Documentum 클라이언트 라이브러리가 있는 디렉토리를 지정하도록 LSDC_DMCL 환경 변수를 설정하십시오.

   예를 들어, 다음과 같습니다.

   ```
   export LSDC_DMCL=/usr/documentum/product/3.1.7
   ```

2. 루트로서 다음의 명령을 입력하십시오.

   ```
   ksh djxlinkDctm
   ```

주: DB2 Universal Database FixPak을 적용한 후에 djxlinkDctm 명령을 다시 실행해야 합니다.

이 태스크의 다음 태스크는 *Documentum의 클라이언트 dmcl.ini 파일 지정*입니다.

관련 태스크:

• 36 페이지의『Documentum의 클라이언트 dmcl.ini 파일 지정』

Documentum의 클라이언트 dmcl.ini 파일 지정

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. Documentum Docbase에 대한 액세스는 Documentum 클라이언트의 dmcl.ini 파일에 의해 제어됩니다. DB2 인스턴스는 Documentum Docbase에 대한 액세스를 얻기 위해 Documentum 클라이언트의 dmcl.ini 파일에 설정된 환경 변수를 가져야 합니다.

절차:

환경 변수를 설정하려면 다음과 같이 하십시오.

1. db2dj.ini 파일을 편집하고 다음의 환경 변수 중 하나를 설정하십시오.

   ```
   DOCUMENTUM=<path>
   DMCL_CONFIG=<path>/dmcl.ini
   ```

   여기서 <path>는 사용하려는 dmcl.ini 파일이 포함된 완전한 디렉토리입니다.

   Documentum의 dmcl.ini 파일 위치에 대한 디폴트 경로는 /pkgs/documentum입니다. 두 행이 모두 포함되면, DMCL_CONFIG가 사용됩니다.

   AIX 및 Solaris 운영 환경에서 db2dj.ini 파일은 `$HOME/sqllib/cfg`에 있습니다.

   Windows에서 db2dj.ini 파일은 `x:\sqllib\cfg`에 있으며 여기서 **x:**는 sqllib 디렉토리가 있는 드라이브를 나타냅니다.

   주: DB2 인스턴스 보고서의 액세스 가능한 모든 Docbase에 대한 Docbroker의 이름이 37 페이지의 그림 5에 표시된 대로 dmcl.ini 파일에 지정되었는지 확인하십시오.

```
################ DOCUMENTUM CLIENT CONFIGURATION FILE ################
#
# Copyright Documentum 1994.
# Version 3.1 of the Documentum Server.
#
# A generated client init file for the Documentum Server.
#
# The only REQUIRED information in this file is the
# [DOCBROKER_PRIMARY] section and an entry for host.
# The host value should be the name of host on which
# your network wide DocBroker is running

[DOCBROKER_PRIMARY]
host = server16.comp2.big.com
```

그림 5. Docbroker 이름이 지정된 샘플 dmcl.ini 파일

2. 다음 명령을 실행하여 DB2 인스턴스를 재순환하십시오.

```
db2stop
db2start
```

이 태스크의 다음 태스크는 *Documentum 랩퍼 등록*입니다.

관련 태스크:

- 18 페이지의 『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 35 페이지의 『Documentum 클라이언트 라이브러리에 링크(AIX 및 Solaris 운영 환경에만 해당)』
- 39 페이지의 『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 103 페이지의 『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 126 페이지의 『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

Documentum 랩퍼 등록

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. 데이터 소스에 액세스하려면 랩퍼를 등록해야 합니다. 랩퍼는 페더레이티드 서버가 통신을 하고 데이터 소스에서 데이터를 검색하는 데 사용하는 메커니즘입니다. 랩퍼는 라이브러리 파일로 시스템에 설치됩니다.

절차:

Documentum 랩퍼를 등록하려면 CREATE WRAPPER문을 실행하십시오.

예를 들어, AIX에서 디폴트 라이브러리 파일 `libdb2lsdctm.a`로부터 `Dctm_Wrapper`라고 하는 Documentum 랩퍼를 작성하려면 다음의 명령문을 제출하십시오.

```
CREATE WRAPPER Dctm_Wrapper LIBRARY 'libdb2lsdctm.a'
OPTIONS(DB2_FENCED 'N');
```

지원되는 플랫폼별로 Documentum 랩퍼에 대한 디폴트 라이브러리 파일 이름의 표는 아래의 관련 태스크 절에서 ″DB2 Life Sciences Data Connect를 설치한 후″를 참조하십시오. CREATE WRAPPER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정*입니다.

관련 태스크:

- 11 페이지의『DB2 Life Sciences Data Connect를 설치한 후』
- 17 페이지의『테이블 구조 파일 랩퍼 등록』
- 39 페이지의『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 76 페이지의『Excel 랩퍼 등록』
- 102 페이지의『BLAST 랩퍼 등록』
- 125 페이지의『XML 랩퍼 등록』

Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. Documentum 데이터 소스에 액세스할 때 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오. 이 변수는 페더레이티드 서버가 초기화 중에 랩퍼를 로드하는지 여부를 결정합니다.

절차:

DB2_DJ_COMM 환경 변수를 설정하려면 연관된 CREATE WRAPPER문에 지정한 랩퍼에 해당하는 랩퍼 라이브러리와 함께 `db2set` 명령을 제출하십시오.

예를 들어, 다음과 같습니다.

```
db2set DB2_DJ_COMM='libdb2lsdctm.a'
```

등호(=)의 양쪽에 스페이스가 없게 하십시오.

데이터베이스 시작 중 랩퍼 라이브러리 로드와 연관된 오버헤드가 있습니다. 이 오버헤드를 방지하려면 액세스하려는 라이브러리만을 지정하십시오.

DB2_DJ_COMM 환경 변수에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

이 태스크의 다음 태스크는 *Documentum 데이터 소스에 대한 서버 등록*입니다.

관련 태스크:

- 18 페이지의『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 38 페이지의『Documentum 랩퍼 등록』
- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 103 페이지의『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 126 페이지의『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

Documentum 데이터 소스에 대한 서버 등록

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. 랩퍼를 등록한 후에 해당 서버를 등록해야 합니다.

절차:

Documentum 서버를 페더레이티드 시스템에 등록하려면 CREATE SERVER문을 사용하십시오.

예를 들어, 연관된 CREATE WRAPPER문에서 작성된 Dctm_Wrapper 랩퍼에 대해 Dctm_Server1이라고 하는 서버가 있다고 가정하십시오. AIX에서 실행하며 Oracle을 사용하여 데이터를 저장하는 Docbase가 서버에 들어 있다고 가정하십시오. 서버를 등록하려면 다음 명령을 실행하십시오.

```
CREATE SERVER Dctm_Server1
TYPE  DCTM
VERSION 3
WRAPPER Dctm_Wrapper
OPTIONS( NODE 'Dctm_Docbase',
OS_TYPE 'AIX',
RDBMS_TYPE 'ORACLE');
```

인수

TYPE 데이터 소스의 유형을 지정합니다. Documentum에 대한 유형은 DCTM입니다. 이 인수는 필수입니다.

VERSION

데이터 소스의 버전을 지정합니다. EDMS98에 대한 값은 '3'입니다. 4i에 대한 값은 '4'입니다. 이 인수는 필수입니다.

WRAPPER

이 서버와 연관된 랩퍼의 이름을 지정합니다. 이 인수는 필수입니다.

옵션

CONTENT_DIR

GET_FILE, GET_FILE_DEL, GET_RENDITION 및
GET_RENDITION_DEL 의사 컬럼에 의해 검색된 내용 파일을 저장하

기 위해 로컬로 액세스 가능한 루트 디렉토리의 이름을 지정합니다. 이러한 의사 컬럼을 사용할 수 있는 모든 사용자가 쓸 수 있어야 합니다. 디폴트값은 /tmp입니다. 이 옵션은 선택적입니다.

NODE

Documentum Docbase의 실제 이름을 지정합니다. 이 옵션은 필수입니다.

OS_TYPE

Docbase 서버의 운영 체제를 지정합니다. 올바른 값은 AIX, SOLARIS 및 WINDOWS입니다. 이 옵션은 필수입니다.

RDBMS_TYPE

Docbase가 사용하는 RDBMS를 지정합니다. 올바른 값은 DB2, INFORMIX, ORACLE, SQLSERVER 또는 SYBASE입니다. 이 옵션은 필수입니다.

TRANSACTIONS

서버 트랜잭션 모드를 지정합니다. 올바른 값은 다음과 같습니다.

- NONE -- 트랜잭션이 사용 가능하지 않습니다.
- QUERY -- 트랜잭션이 Dctm_Query 메소드에 대해서만 사용 가능합니다.
- ALL -- 트랜잭션이 Dctm_Query 메소드에 대해 사용 가능합니다. ALL은 이 릴리스에서 QUERY와 동일한 기능을 가집니다.

디폴트값은 QUERY입니다. 이 옵션은 선택적입니다.

CREATE SERVER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *사용자 맵핑(Documentum 랩퍼)*입니다.

관련 태스크:

- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 39 페이지의『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 42 페이지의『사용자 맵핑(Documentum 랩퍼)』
- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』
- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』

- 127 페이지의『XML 데이터 소스에 대한 서버 등록』

사용자 맵핑(Documentum 랩퍼)

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. 이전에 정의된 서버에 사용자를 맵핑하여 데이터 소스에 대한 액세스 권한을 부여해야 합니다.

절차:

페더레이티드 서버에 사용자를 맵핑하려면 CREATE USER MAPPING문을 사용하십시오.

예를 들어, 다음의 CREATE USER MAPPING문은 사용자 Chuck을 Dctm_Server1 서버의 사용자 Charles에 맵핑합니다.

```
CREATE USER MAPPING FOR Chuck SERVER Dctm_Server1
OPTIONS(REMOTE_AUTHID 'Charles', REMOTE_PASSWORD 'Charles_pw');
```

또한 사용자 자신의 사용자 맵핑을 정의할 수 있습니다. 다음의 예에서 USER는 이름이 USER인 사용자가 아니라 현재 사용자를 의미하는 키워드입니다.

```
CREATE USER MAPPING FOR USER SERVER Dctm_Server1
OPTIONS(REMOTE_AUTHID 'Lisa', REMOTE_PASSWORD 'Lisa_pw')
```

CREATE USER MAPPING문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *Documentum 데이터 소스에 대한 별칭 등록*입니다.

관련 태스크:

- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』

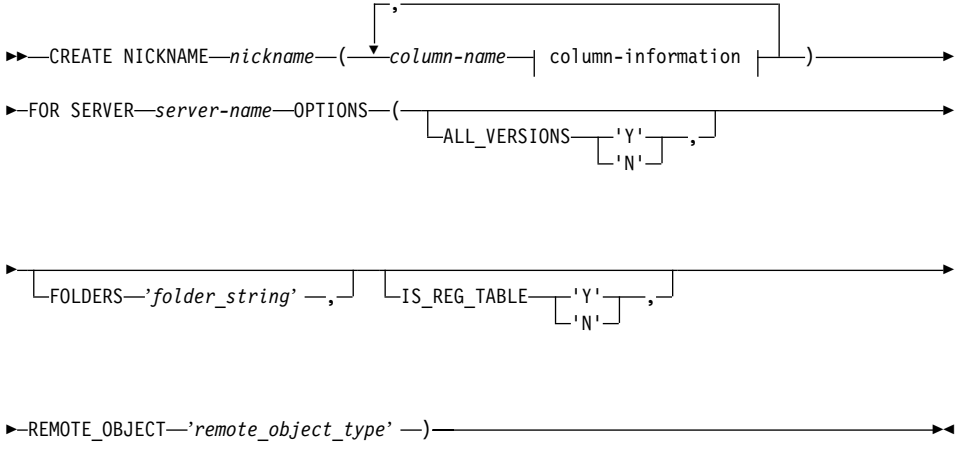
Documentum 데이터 소스에 대한 별칭 등록

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. 서버를 등록하고 서버에 사용자를 맵핑한 후에 해당 별칭을 등록해야 합니다. 별칭은 쿼리에서 Documentum 데이터 소스를 참조할 때 사용됩니다.

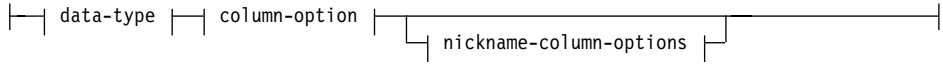
절차:

별칭을 등록하려면 CREATE NICKNAME문을 사용하여 관심이 있는 등록된 테이블이나 각 오브젝트 유형에 대한 각 Docbase의 별칭을 작성하십시오.

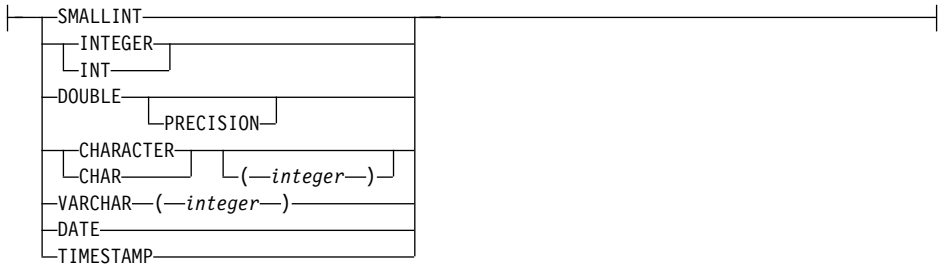
Documentum에 대한 CREATE NICKNAME문의 구문은 다음과 같습니다.



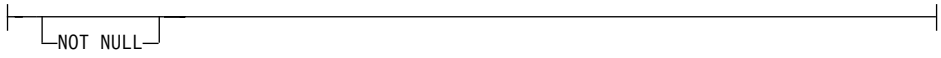
column-information:



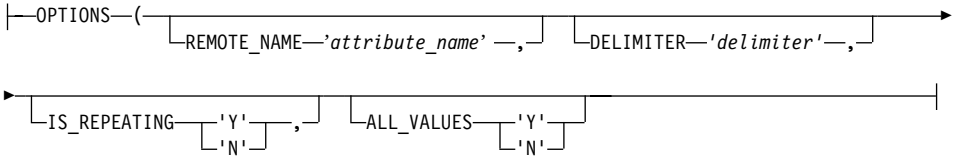
data-type:



column-option:



nickname-column-options:



CREATE NICKNAME문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

컬럼 옵션

NOT NULL

TIMESTAMP 및 DATE로 정의된 컬럼을 제외한 모든 단일 값 컬럼을 NOT NULL로 정의해야 합니다. 반복 속성은 별칭에서 NOT NULL로 정의되지 않아야 합니다.

별칭 컬럼 옵션

별칭 컬럼 옵션 값은 작은따옴표로 묶어야 합니다.

ALL_VALUES

반복되는 속성의 모든 값이 지정된 분리문자로 구분되어 리턴될 것임을 지정합니다. 이 옵션이 누락되었거나 'N'이면 반복 속성의 마지막 값만이 리턴됩니다. DELIMITER 아래에 표시된 대로 ALL_VALUES는

IS_REPEATING 옵션이 'Y'인 VARCHAR 컬럼에 대해서만 지정할 수 있습니다(IS_REG_TABLE = 'Y'이면 유효하지 않습니다).

DELIMITER

반복 속성의 여러 값을 연결할 때 사용되는 분리문자 문자열을 지정합니다. 분리문자는 하나 이상의 문자일 수 있습니다. 디폴트 분리문자는 쉼표입니다. 이 옵션은 IS_REPEATING 옵션이 'Y'로 설정된 데이터 유형이 VARCHAR인 오브젝트의 속성에 대해서만 유효합니다. 이 옵션은 선택적입니다.

IS_REPEATING

컬럼이 다중 값인지를 표시합니다. 올바른 값은 'Y' 및 'N'입니다. 디폴트값은 'N'입니다. 이 옵션은 선택적입니다.

REMOTE_NAME

상응하는 Documentum 속성 또는 컬럼의 이름을 지정합니다. 이 옵션은 리모트 속성이나 컬럼 이름을 로컬 DB2 컬럼 이름으로 맵핑합니다. 상응하는 디폴트값은 DB2 컬럼 이름입니다. 이 옵션은 선택적입니다.

별칭 옵션

별칭 옵션 값은 작은따옴표 안에 있어야 합니다.

ALL_VERSIONS

모든 오브젝트 버전의 검색 여부를 지정합니다. 올바른 값은 'y', 'Y', 'n' 및 'N'입니다. 디폴트값 'N'은 현재 오브젝트 버전만이 쿼리 처리에 포함됨을 의미합니다. 이 옵션은 IS_REG_TABLE = 'Y'일 때 유효합니다. 이 옵션은 선택적입니다.

FOLDERS

논리적으로 결합되고 구문이 정확한 하나 이상의 Documentum FOLDER 술어가 들어 있는 문자열을 지정합니다. FOLDER 술어를 지정하면 이 별칭으로 표시된 문서 세트를 지정된 폴더에 있는 문서 세트로 제한합니다.

이 옵션 지정시, 작은따옴표로 FOLDERS 옵션의 전체 값을 묶고 문자열 내에서 작은따옴표 대신 큰따옴표를 사용하십시오.

예를 들어, 다음을 삽입하려는 경우

```
FOLDER('/Tools',DESCEND) OR FOLDER('/Cars')
```

다음 FOLDERS 옵션을 지정하십시오.

```
FOLDERS 'FOLDER("/Tools",DESCEND) OR FOLDER("/Cars")'
```

이 옵션은 IS_REG_TABLE = 'Y'일 때 유효합니다. 이 옵션은 선택적입니다.

IS_REG_TABLE

REMOTE_OBJECT 옵션으로 지정한 오브젝트가 Documentum 등록 테이블인지 여부를 지정합니다. 올바른 값은 'y', 'Y', 'n' 및 'N'입니다. 디폴트값은 'N'입니다. 이 옵션은 선택적입니다.

주: ALTER NICKNAME문을 통해 이 옵션을 변경하여 Documentum 오브젝트에서 등록된 테이블로(또는 그 반대로) 별칭을 변경할 수 없습니다. 그 대신 별칭을 삭제(DROP)하거나 다시 작성(re-CREATE)해야 합니다.

REMOTE_OBJECT

별칭과 연관된 Documentum 오브젝트 유형의 이름을 지정합니다. 이름은 Documentum 오브젝트 유형 또는 등록 테이블일 수 있습니다. 등록 테이블의 경우, 테이블 소유자의 이름이 접두부로 표시되어야 합니다. 등록된 테이블이 Docbase 소유자에 속하는 경우, 소유자 이름에 대해 dm_dbo가 사용될 수 있습니다. 이 옵션은 필수입니다.

주: ALTER NICKNAME을 사용하여 REMOTE_OBJECT 옵션의 값을 변경하면 새로운 오브젝트의 구조가 원래의 오브젝트 구조와 유사하지 않을 경우 오류가 발생합니다.

의사 컬럼 이해

CREATE NICKNAME문은 여섯 개의 의사 컬럼도 정의합니다. 이러한 컬럼은 오브젝트 내용 및 기타 정보에 액세스하는 데 사용됩니다.

의사 컬럼과 이에 대한 정의는 47 페이지의 표 8에 나열되어 있습니다.

표 8. 의사 컬럼 이름 및 정의.

의사 컬럼 이름	정의
GET_FILE	VARCHAR (255)
GET_FILE_DEL	VARCHAR (255)
GET_RENDITION	VARCHAR (255)
GET_RENDITION_DEL	VARCHAR (255)
HITS	INTEGER
SCORE	DOUBLE

표 9에는 SELECT절에 대한 의사 컬럼을 나열한 것입니다.

표 9. SELECT절에 대한 의사 컬럼

의사 컬럼 이름	설명

GET_FILE	컬럼 값 외에 현재 행에 대한 내용 파일을 검색합니다.
	
	내용 파일에 대한 확장자는 Documentum 형식 이름입니다. 동일한 이름의 파일이 존재하는 경우, 겹쳐쓰여집니다.
	
	GET_FILE은 오브젝트의 기본 형식을 따르려고 시도합니다. 행에서의 해당 값은 오브젝트의 a_content_type입니다. 오브젝트가 내용 파일을 갖지 않는 경우 해당 값은 문자열 ″no_content″입니다.
	
	예를 들어, 다음과 같습니다.
	`SELECT object_name, DCTM.GET_FILE`
	`FROM ...`
	
	내용 파일은 서버의 CONTENT_DIR 옵션으로 지정된 서버 디렉토리에 위치합니다. 또한 사용자의 DB2 로컬 이름으로 지정된 서브디렉토리에 위치합니다. 서브디렉토리가 없는 경우 작성됩니다.
	
	해당 확장자는 문서의 형식 유형에 대한 Docbase에 정의된 관련 DOS 확장자입니다. 예를 들어, MS Word 문서에 대해서는 ″.doc″입니다.
	
	문자열 ″no_content″ 또는 파일의 완전한 이름을 리턴합니다.

GET_FILE_DEL	GET_FILE_DEL이 해당 쿼리에서 이전 행에 대해 검색된 파일이 있는 경우 우선 이 파일을 삭제한다는 점만을 제외하고 이 함수는 GET_FILE과 동일합니다. 문자열 ″no_content″ 또는 파일의 완전한 이름을 리턴합니다.

표 9. SELECT절에 대한 의사 컬럼 (계속)

의사 컬럼 이름	설명

GET_RENDITION	해당 렌디션(rendition)의 내용 파일, 다른 형식으로 된 원래 문서의 사본을 컬럼 값 이외에 현재 행에 대해서도 검색합니다.
	
	내용 파일에 대한 확장자는 Documentum 형식 이름입니다. 동일한 이름의 파일이 존재하는 경우, 겹쳐쓰여집니다.
	
	렌디션 형식을 정의하려면 WHERE절에서 술어를 DCTM.RENDITION_FORMAT(<format) = 1 형식으로 지정해야 합니다.
	
	예를 들어, 다음과 같습니다.
	
	`SELECT object_name, get_rendition`
	`FROM ...`
	`WHERE DCTM.RENDITION_FORMAT('pdf')=1`
	
	GET_RENDITION은 오브젝트의 이름 지정된 렌디션을 따르려고 시도합니다. 오브젝트가 내용 파일을 갖지 않아 해당 값이 문자열 ″no_content″인 경우나 렌디션(rendition)이 없어 해당 값이 문자열 ″not_found″인 경우를 제외하고는 해당 행의 값은 오브젝트의 a_content_type입니다.
	
	내용 파일은 서버의 CONTENT_DIR 옵션으로 지정된 서버 디렉토리에 위치합니다. 또한 사용자의 DB2 로컬 이름으로 지정된 서브디렉토리에 위치합니다. 서브디렉토리가 없는 경우 작성됩니다.
	
	해당 확장자는 문서의 형식 유형에 대한 Docbase에 정의된 관련 DOS 확장자입니다. 예를 들어, MS Word 문서에 대해서는 ″.doc″입니다.
	
	문자열 ″no_content″, ″not found″ 또는 파일의 완전한 이름을 리턴합니다.

GET_RENDITION_DEL	GET_RENDITION_DEL이 해당 쿼리에서 이전 행에 대해 검색된 파일이 있는 경우 우선 이 파일을 삭제한다는 점만을 제외하고 이 함수는 GET_RENDITION과 동일합니다. 문자열 ″no_content″, ″not found″ 또는 파일의 완전한 이름을 리턴합니다.

49 페이지의 표 10은 검색 절이 포함된 쿼리에 있는 SELECT절에 대한 의사 컬럼을 나열한 것입니다.

표 10. 검색 절이 포함된 쿼리에 있는 SELECT절에 대한 의사 컬럼

의사 컬럼 이름	설명

HITS	검색 기준이 일치하는 문서에서 갯수를 표시하는 정수를 리턴합니다.
	
	예를 들어, 다음과 같습니다.
	
	`SELECT r_object_id, object_name, hits`
	`FROM std_doc`
	`WHERE DCTM.SEARCH_WORDS ('''workflow'' OR ''flowchart''')=1`
	
	리턴되는 각 문서의 경우, 문서 내용 안에서 ″workflow″ 및 ″flowchart″ 단어의 갯수를 합해 HITS 값으로 리턴합니다.
	
	HITS 의사 컬럼은 문서에 하나의 내용 파일만이 있는 경우에 적합합니다. 이것은 일반적인 경우입니다. 이 의사 컬럼은 SELECT문에 대한 WHERE절 조건에 사용할 수 있습니다. 그러나 SELECT절에서도 지정되어야 합니다.

SCORE	문서의 관련 순위를 리턴합니다.
	
	이 의사 컬럼은 Documentum의 ACCRUE 개념 연산자와 결합하여 사용하십시오. 둘 다 각 리턴된 문서에서 찾은 지정된 단어의 갯수를 나타내는 숫자를 리턴합니다.
	
	예를 들어, 다음과 같습니다.
	
	`SELECT object_name, score`
	`FROM std_doc`
	`WHERE DCTM.SEARCH_TOPIC('<ACCRUE>("document","management","workflow")')=1`
	`AND SCORE >=75`
	
	명령문은 해당 내용에서 지정된 단어의 2 - 3개를 갖는 모든 문서를 리턴합니다. 문서가 오직 하나의 단어를 갖는 경우, 점수 50이 지정되므로 WHERE절 기준에 맞지 않아 리턴하지 않습니다. 세 개의 단어 중 두 개를 찾은 경우, 문서에 75의 점수가 지정됩니다. 세 개의 모든 단어를 찾은 경우, 문서의 점수는 88입니다.
	
	SCORE 의사 컬럼은 하나의 내용 파일이 있는 문서에 사용됩니다. 이것은 일반적인 경우입니다.
	
	SCORE는 WHERE에 SEARCH_WORDS() 또는 SEARCH_TOPIC() 함수가 포함된 경우에만 SELECT절에 있을 수 있습니다. WHERE절에서, ACCRUE 개념 연산자와 결합하여 사용됩니다.
	
	ACCRUE 개념 연산자에 대한 정보는 Documentum 문서를 참조하십시오.

CREATE NICKNAME의 예

다음 CREATE NICKNAME문은 별칭 std_doc을 정의합니다. Std_doc은 dm_document의 오브젝트 유형과 함께 Documentum Docbase와 연관됩니다. 표 11은 CREATE NICKNAME문을 구성하는 데 사용되는 DB2 관계 컬럼 이름으로 Documentum 속성 및 데이터 유형을 맵핑합니다.

표 11. std_doc 별칭에 대한 DB2 컬럼으로 Documentum 속성 맵핑

Documentum 속성 이름	Documentum 데이터 유형	DB2 컬럼 이름	DB2 데이터 유형	반복 여부	널(NULL) 입력 가능
object_name	string(255)	object_name	varchar	아니오	아니오
r_object_id	ID	object_id	char(16)	아니오	아니오
r_object_type	string(32)	object_type	varchar	아니오	아니오
title	string(255)	title	varchar	아니오	아니오
subject	string(128)	subject	varchar	아니오	아니오
authors	string(32)	author	varchar	예	예
keywords	string(32)	keyword	varchar	예	예
r_creation_date	time	creation_date	timestamp	아니오	예
r_modify_date	time	modified_date	timestamp	아니오	예
a_status	string(16)	status	varchar	아니오	아니오
a_content_type	string(32)	content_type	varchar	아니오	아니오
r_content_size	double	content_size	integer	아니오	아니오
owner_name	string(32)	owner_name	varchar	아니오	예

표 12는 별칭에 사용된 각 Documentum 속성을 설명한 것입니다.

표 12. std_doc 별칭에 대한 Documentum 속성의 설명

Documentum 속성 이름	설명
object_name	오브젝트의 사용자 정의 이름
r_object_id	작성 시간에 설정된 이 오브젝트에 대한 고유한 오브젝트 ID
r_object_type	오브젝트가 작성될 때 설정된 오브젝트의 유형
title	오브젝트의 사용자 정의 제목
subject	오브젝트의 사용자 정의 주제
authors	오브젝트에 대한 작성자의 사용자 정의 목록
keywords	오브젝트에 대한 사용자 정의 키워드 목록

표 12. std_doc 별칭에 대한 Documentum 속성의 설명 (계속)

Documentum 속성 이름	**설명**
r_creation_date	오브젝트가 작성된 날짜 및 시간
r_modify_date	오브젝트가 최종 수정된 날짜 및 시간
a_status	라우터 태스크가 이송될 때 서버에 의해 설정됩니다. 값은 라우터 상태에서 attached_task_status로 지정된 값에서 얻습니다.
a_content_type	오브젝트 내용의 파일 형식.
r_content_size	내용의 바이트 수. 여러 페이지 문서의 경우, 이 속성은 문서와 연관된 첫 번째 내용의 크기를 기록합니다.
owner_name	오브젝트 소유자의 이름(오브젝트를 작성한 사용자).

50 페이지의 표 11은 다음 CREATE NICKNAME문으로 변환합니다.

```
CREATE NICKNAME std_doc (
  object_name varchar(255) not null,
  object_id char(16) not null OPTIONS(REMOTE_NAME 'r_object_id'),
  object_type varchar(32) not null OPTIONS(REMOTE_NAME 'r_object_type'),
  title varchar(255) not null,
  subject varchar(128) not null,
  author varchar(32) OPTIONS(REMOTE_NAME 'authors', IS_REPEATING 'Y'),
  keyword varchar(32) OPTIONS(REMOTE_NAME 'keywords', IS_REPEATING 'Y'),
  creation_date timestamp OPTIONS(REMOTE_NAME 'r_creation_date'),
  modifed_date timestamp OPTIONS(REMOTE_NAME 'r_modify_date'),
  status varchar(16) not null OPTIONS(REMOTE_NAME 'a_status'),
  content_type varchar(32) not null OPTIONS(REMOTE_NAME 'a_content_type'),
  content_size integer not null OPTIONS(REMOTE_NAME 'r_content_size'),
  owner_name varchar(32))
FOR SERVER Dctm_Server2 OPTIONS (REMOTE_OBJECT 'dm_document', IS_REG_TABLE 'N')
```

CREATE NICKNAME문을 제출한 후에, 별칭 std_doc를 사용하여 페더레이티드 시스템을 쿼리할 수 있습니다. 또한 페더레이티드 시스템의 다른 별칭 및 테이블로 std_doc 별칭을 조인할 수 있습니다.

주: 카탈로그에서 이 별칭에 대한 컬럼의 수는 의사 컬럼으로 인해 CREATE NICKNAME문에 지정되는 것보다 여섯 개가 더 많습니다.

CreateNicknameFile 유틸리티를 사용하여 Documentum 유형을 자동으로 DB2 유형으로 맵핑하고 초기 CREATE NICKNAME문을 작성할 수 있습니다. CreateNicknameFile 유틸리티에 대한 자세한 정보는 아래의 관련 링크 절을 참조하십시오.

이 태스크의 다음 태스크는 *Documentum 데이터 소스에 대한 사용자 정의 기능 등록*입니다.

관련 태스크:

- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 42 페이지의『사용자 맵핑(Documentum 랩퍼)』
- 52 페이지의『Documentum 데이터 소스에 대한 사용자 정의 기능 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』
- 145 페이지의 제 8 장『비용산정 별칭 옵션 지정』

Documentum 데이터 소스에 대한 사용자 정의 기능 등록

이 태스크는 *페더레이티드 시스템에 Documentum 추가*에 대한 기본 태스크의 일부입니다. 여러 사용자 정의 기능을 등록하려면 CREATE FUNCTION문을 사용해야 합니다. 전체 텍스트 검색 및 쿼리 내의 문서 내용 검색과 같이 Documentum의 고유한 일부 성능을 액세스하기 위해 이 함수를 사용할 수 있습니다.

술어에 대한 사용자 정의 기능이 54 페이지의 표 13에서 나열됩니다.

DB2는 BOOLEAN 데이터 유형을 지원하지 않습니다. 그러므로 유효한 SQL문을 작성하려면, 각 사용자 정의 기능 값이 명시적으로 테스트되어야 합니다. 랩퍼 구현은 지정된 테스트 비교 연산자와 관계없이 ″DCTM.<function>(<args>) = 1″에 대한 의미론만을 지원합니다.

주: TOPIC 함수에 대한 참조는 Verity, Inc의 써드 파티 전체 텍스트 인덱스 시스템의 일부로서 제공된 Documentum 함수에 대한 것입니다.

절차:

사용자 정의 기능을 등록하려면 CREATE FUNCTION문을 사용하십시오.

모든 사용자 정의 기능은 스키마 이름 DCTM으로 등록되어야 합니다. 각 함수의 완전한 이름은 DCTM.<function_name>입니다.

다음 예는 ANY_EQ 사용자 정의 기능을 등록합니다.

```
CREATE FUNCTION DCTM.ANY_EQ (CHAR(), CHAR()) RETURNS INTEGER
AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION
```

Documentum 랩퍼를 설치하는 각 DB2 데이터베이스에 대해 각 사용자 정의 기능을 한 번 등록해야 합니다.

사용자 정의 기능의 등록을 돕기 위해 샘플 파일 create_function_mappings.ddl 이 sqllib/samples/lifesci 디렉토리에 제공됩니다. 이 파일에는 각 사용자 정의 기능에 대한 정의가 들어 있습니다. Documentum 랩퍼가 설치된 각 DB2 데이터베이스에 대한 사용자 정의 기능을 등록하기 위해 이 ddl 파일을 실행할 수 있습니다.

사용자 정의 기능 문자열 인수 규칙

문자열로 패스되는 모든 인수는 다음 규칙을 지켜야 합니다.

- 각 문자열은 작은따옴표로 묶입니다.
- 문자열 내의 작은따옴표는 두 개의 작은따옴표로 표시됩니다.

쿼리에서 사용자 정의 기능 사용

다음 예는 쿼리에서 사용자 정의 기능의 사용을 설명합니다.

'Dave Winters'라는 하나 이상의 작성자를 갖는 문서에 대한 std_doc 별칭에서 오브젝트 이름 및 작성자를 표시하려면 다음을 수행하십시오.

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1
```

'Dave Winters' 또는 'Jon Doe'라는 하나 이상의 작성자를 갖는 문서에 대한 std_doc 별칭에서 오브젝트 이름 및 작성자를 표시하려면 다음을 수행하십시오.

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_IN(authors,'Dave Winters','Jon Doe')=1
```

오브젝트 이름 및 r_object_id를 표시하고, 작성자 컬럼에 'Dave Win%'와 같은 문자열이 들어 있는 문자에 대한 std_doc 별칭에서 내용 파일을 검색하려면 다음을 수행하십시오.

```
SELECT object_name, r_object_id, get_file FROM std_doc
WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1
```

사용자 정의 기능 테이블

표 13은 술어에 대한 사용자 정의 기능을 나열한 것입니다.

표 13. 술어에 대한 사용자 정의 기능

함수 이름	설명
ANY_EQ(arg1, arg2)	지정된 값과 동일한 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 비교되는 값을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1`
ANY_NE(arg1, arg2)	지정된 값과 동일하지 않은 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 비교되는 값을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_NE(authors,'Dave Winters')=1`
ANY_LT(arg1, arg2)	지정된 값 미만의 모든 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 비교되는 값을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_LT(num_approvers,4)=1`

표 13. 술어에 대한 사용자 정의 기능 (계속)

함수 이름	설명
ANY_GT(arg1, arg2)	지정된 값보다 큰 모든 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 비교되는 값을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_GT(num_approvers,3)=1`
ANY_LE(arg1, arg2)	지정된 값 이하의 모든 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 비교되는 값을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_LE(num_approvers,2)=1`
ANY_GE(arg1, arg2)	지정된 값 이상의 모든 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 비교되는 값을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_GE(num_approvers,1)=1`
ANY_IN(arg1, arg2 - arg11)	지정된 값 목록에서 10개의 값에 대한 반복 속성을 테스트합니다. 동일한 데이터 유형의 3-11개의 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2-arg11** 쉼표로 구분되는 비교 값 목록을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_IN(authors,'Crick','Watson')=1`

표 13. 술어에 대한 사용자 정의 기능 (계속)

함수 이름	설명
ANY_LIKE(arg1, arg2)	지정된 값과 같은 모든 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 작은따옴표 안의 부속 문자열과 비교 중인 패턴을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1 OR DCTM.ANY_LIKE(keywords,'%\_%')=1` **주:** Escape절은 ANY_LIKE() 술어에서 지원되지 않습니다.
ANY_NOT_LIKE(arg1, arg2)	지정된 값과 같지 않은 모든 값에 대한 반복 속성을 테스트합니다. 다음 두 개의 필수 인수를 갖습니다. **arg1** 반복 속성을 나타내는 컬럼의 이름을 지정합니다. **arg2** 작은따옴표 안의 부속 문자열과 비교 중인 패턴을 지정합니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_NOT_LIKE(authors,'Dave Win%')=1 OR DCTM.ANY_NOT_LIKE(keywords,'%\_%')=1` **주:** Escape절은 ANY_NOT_LIKE() 술어에서 지원되지 않습니다.
ANY_NULL(arg)	IS NULL에 대한 반복 속성을 테스트합니다. 반복 속성이나 단일 값 DATE 또는 TIMESTAMP 속성의 이름인 하나의 필수 인수를 갖습니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_NULL(authors)=1`
ANY_NOT_NULL(arg)	IS NOT NULL에 대한 반복 속성을 테스트합니다. 반복 속성의 이름인 하나의 필수 인수를 갖습니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.ANY_NOT_NULL(authors)=1`

표 13. 술어에 대한 사용자 정의 기능 (계속)

함수 이름	설명

ANY_SAME_INDEX(arg1 - arg10)	각 속성의 동일한 인덱스에서 값에 대한 반복 속성을 테스트합니다. 2 - 10개의 다른 ANY_xx() 함수를 갖습니다.
	
	다음 예는 UCD와 관계되지 않은 Ken이라는 최소한 한 명의 작성자를 문서가 갖는지 여부를 검사합니다.
	
	`... WHERE DCTM.ANY_SAME_INDEX(`
	`ANY_EQ(author_name,'Ken'),`
	`DCTM.ANY_NE(author_affiliation,'UCD'))=1`

CABINET(arg) 및 CABINET_TREE(arg)	Docbase 캐비넷의 완전한 이름인 필수 인수를 하나 갖습니다.
	
	예를 들어, 다음과 같습니다.
	
	`... WHERE DCTM.CABINET('/Tools')=1`
	`... WHERE DCTM.CABINET_TREE('/MyDocs')=1`
	
	CABINET 및 CABINET_TREE의 여러 인스턴스를 사용하여 여러 캐비넷을 지정하십시오.
	
	예를 들어, 다음과 같습니다.
	
	`... WHERE DCTM.CABINET('/Tools')=1`
	`OR DCTM.CABINET_TREE('/Parts')=1`

FOLDER(arg) 및 FOLDER_TREE(arg)	Docbase 폴더 또는 캐비넷의 완전한 이름인 필수 인수를 하나 갖습니다.
	
	예를 들어, 다음과 같습니다.
	
	`... DCTM.FOLDER('/Tools/Drills')=1`
	`... DCTM.FOLDER_TREE('/MyDocs/WhitePapers')=1`
	
	FOLDER 및 FOLDER_TREE의 여러 인스턴스를 사용하여 여러 폴더를 지정하십시오.
	
	예를 들어, 다음과 같습니다.
	
	`... DCTM.FOLDER('/Tools/Drills')=1`
	`OR DCTM.FOLDER_TREE('/Animals/Horses')=1`

RENDITION_FORMAT (format)	검색할 렌디션의 형식을 설정하기 위해 GET_RENDITION 및 GET_RENDITION_DEL 의사 컬럼에 대해 작업합니다. 형식을 지정하는 한 문자의 문자열 인수를 사용합니다.
	
	다음의 예는 PDF 형식의 문서를 검색합니다.
	
	`SELECT get_rendition`
	`FROM ....`
	`WHERE DCTM.RENDITION_FORMAT('pdf')=1`

표 13. 술어에 대한 사용자 정의 기능 (계속)

함수 이름	설명
USER(1)	현재 사용자의 Documentum 작성자 ID와 값을 비교합니다. 1이어야 하는 더미 인수를 갖습니다. 예를 들어, 다음과 같습니다. `... WHERE approver = DCTM.USER(1)` **주:** Documentum 작성자 ID가 DB2 작성자 ID에 해당하게 하려면, CREATE USER MAPPING문을 사용하십시오. 사용자 맵핑에 대한 자세한 정보는 아래의 관련 링크 절을 참조하십시오.

SEARCH_WORDS(arg)	AND, OR 또는 NOT으로 구분되며 작은따옴표로 묶인 각 단어의 목록인 하나의 필수 문자열 인수를 갖고 괄호를 사용하여 우선 순위를 제어합니다. 단어는 스페이스를 포함할 수 없으며 작은따옴표로 묶여야 합니다.
	
	예를 들어, 다음과 같습니다.
	`... DCTM.SEARCH_WORDS('''yeast''`
	`AND (''bread'' OR ''cake'')`
	`AND NOT ''wedding''' )=1`

SEARCH_TOPIC(arg)	Documentum 및 Verity verbatim으로 패스되는 Verity TOPIC 쿼리문인 하나의 필수 인수를 갖습니다. 예를 들어, 다음과 같습니다. `... WHERE DCTM.SEARCH_TOPIC('"quick"')=1`

CREATE FUNCTION문에 대한 자세한 내용은 *DB2 SQL 참조서*를 참조하십시오.

이 태스크 다음에는 더 이상의 태스크가 없습니다.

관련 태스크:

• 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』

Documentum 데이터 소스에 대한 쿼리 실행

랩퍼를 등록한 후에, Documentum 데이터 소스에 대해 SQL 쿼리를 실행할 수 있습니다. 이 절에서는 쿼리의 몇가지 예를 제공합니다.

절차:

쿼리를 실행하려면 일반 테이블 이름 및 테이블 컬럼을 사용할 때와 동일한 방식으로 SQL문에 별칭과 정의된 별칭 컬럼을 사용하십시오.

다음 쿼리는 'Test Document'라는 문서에 대한 모든 Docbase 문서를 표시합니다.

```
SELECT object_name
FROM std_doc
WHERE object_name='Test Document';
```

다음 쿼리는 사용자 정의 기능 ANY_EQ를 사용하여 작성자 중 한 명이 'Joe Doe'인 모든 문서를 표시합니다.

```
SELECT object_name
FROM std_doc
WHERE DCTM.ANY_EQ(author,'Joe Doe')=1
```

다음 쿼리는 FOLDER_TREE 함수 및 SEARCH_WORDS 함수를 사용하여 텍스트 ″protein″이 들어 있는 승인된 캐비넷에서 모든 문서를 찾습니다.

```
SELECT object_name
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.SEARCH_WORDS('protein')=1
```

다음의 쿼리는 GET_FILE 의사 컬럼과 FOLDER_TREE 및 ANY_IN 사용자 정의 기능을 사용하여 작성자가 나열된 승인된 캐비넷의 모든 문서에 대한 내용이 위치한 DB2 서버에서 파일의 이름을 검색합니다.

```
SELECT object_name, object_id, get_file
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.ANY_IN(author, 'Mary Black', 'Joe Carson', 'Peter Miller')=1
```

관련 태스크:

- 79 페이지의『Excel 데이터 소스에 대한 쿼리 실행』
- 136 페이지의『XML 데이터 소스에 대한 쿼리 실행』

Documentum 랩퍼를 위한 CreateNicknameFile 유틸리티의 정의

자유롭게 다운로드하여 사용할 수 있는 CreateNicknameFile이라는 Docbasic 유틸리티를 사용하여 임의의 Docbase 오브젝트나 등록된 테이블의 완전한 정의가 들어 있는 ASCII 파일을 작성할 수 있습니다. 다음을 위해 출력 파일을 편집할 수 있습니다.

- 컬럼 및 속성에 대한 로컬 이름을 정의. 로컬 및 리모트 이름은 처음에 Docbase에 알려진 이름입니다.
- 원치않는 컬럼 및 속성을 삭제. 사전 정의된 Documentum 문서 유형(dm_document)은 EDMS98에서 59개의 속성을 가지며 4i에서는 76개의 속성을 갖습니다. 대부분의 속성에는 하위 레벨 문서 관리 및 응용프로그램 개발을 위한 메타데이터가 들어 있습니다. 관심 대상이 아닌 속성을 삭제하면 성능에 영향을 미치지 않고 SELECT * SQL문을 더 유용하게 만들 수 있습니다.
- 특정 Documentum 폴더로 이 별칭에 대한 검색을 제한하기 위해 FOLDERS 옵션의 값을 추가.
- 원하는 경우 TIMESTAMP에 대한 DATE 맵핑을 변경. 유틸리티는 가장 유용해 보이는 DQL DATE에서 DB2® DATE로의 맵핑을 생성합니다.
- 응용프로그램에 따라 CHAR 맵핑을 VARCHAR로 또는 역으로 변경.

Docbase에서 유틸리티를 설치하고 Documentum Windows® 그래픽 사용자 인터페이스로부터 실행해야 합니다. 유틸리티가 생성하는 파일은 설치된 Docbase에 고유합니다.

관련 태스크:

- 61 페이지의『CreateNicknameFile 유틸리티 설치(Documentum 랩퍼)』
- 61 페이지의『CreateNicknameFile 유틸리티 구성(Documentum 랩퍼)』
- 63 페이지의『Documentum 등록 테이블에서 DM_ID 오브젝트 유형 맵핑』

CreateNicknameFile 유틸리티 설치(Documentum 랩퍼)

CreateNicknameFile 유틸리티는 Documentum 데이터 소스에 대한 CREATE NICKNAME문을 쓸 때 도움이 됩니다.

절차:

유틸리티를 설치하려면 다음을 수행하십시오.

1. http://www.ibm.com/software/data/db2/lifesciencesdataconnect/에서 DB2 Life Sciences Data Connect 제품 웹사이트의 다운로드 섹션에서 CreateNicknameFile 유틸리티를 다운로드하십시오.
2. CreateNicknameFile.txt라는 유틸리티를 임포트하려면 EDMS98 Workspace 그래픽 사용자 인터페이스 또는 4i Desktop Client를 사용하십시오. Docbase 캐비넷 또는 폴더로 유틸리티를 임포트하거나, 원하는 이름을 부여할 수 있습니다.
3. 새로 임포트한 CreateNicknameFile.txt 오브젝트에 대한 등록 정보 대화 상자에서 **사용자가 실행할 수 있음** 상자를 체크하십시오.

관련 개념:

- 60 페이지의 『Documentum 랩퍼를 위한 CreateNicknameFile 유틸리티의 정의』

관련 태스크:

- 61 페이지의 『CreateNicknameFile 유틸리티 구성(Documentum 랩퍼)』
- 63 페이지의 『Documentum 등록 테이블에서 DM_ID 오브젝트 유형 맵핑』

CreateNicknameFile 유틸리티 구성(Documentum 랩퍼)

CreateNicknameFile 유틸리티는 Documentum 데이터 소스에 대한 CREATE NICKNAME문을 쓸 때 도움이 됩니다.

전제조건:

CreateNicknameFile 유틸리티를 구성하려면 먼저 설치해야 합니다.
CreateNicknameFile 유틸리티 설치에 대한 자세한 정보는 아래의 관련 태스크 절에서 ″CreateNicknameFile 유틸리티 설치(Documentum 랩퍼)″를 참조하십시오.

절차:

유틸리티를 설치 후 구성하려면 다음을 수행하십시오.

1. 유틸리티의 아이콘을 더블 클릭하여 유틸리티를 실행하십시오.
2. Documentum 문서/오브젝트 유형 이름을 입력하십시오. 디폴트값은 dm_document입니다.

   주: 등록된 테이블에 대한 별칭 파일을 작성하는 것이 필요한 경우 이름으로 dm_registered를 지정하십시오. dm_registered를 지정하는 경우, <owner>.<table_name> 형식으로 완전한 테이블 이름에 대해 프롬프트됩니다. Docbase 소유자가 테이블을 소유하는 경우(일반적인 경우) 소유자 이름에 대해 dm_dbo를 사용할 수 있습니다.

   유틸리티는 등록된 테이블에 대한 별칭 이름으로 이름 지정 규칙을 따릅니다. 규칙은 ″등록된 테이블″을 나타내기 위해 테이블 이름에 ″rt_″ 접두부를 붙입니다. 이 규칙을 사용하지 않으려는 경우 유틸리티가 제안한 별칭을 변경할 수 있습니다.
3. 작성 중인 별칭과 연관된 서버 이름을 입력하십시오.
4. 별칭의 이름을 입력하십시오.

   별칭 이름은 설명적이어야 하며 DB2 인스턴스 내에서 고유해야 합니다. 동일한 <object_type>이 여러 서버를 정의하는 데 필요할 수 있으므로 유틸리티는 <server_name>.<object_type>의 이름 지정 규칙을 따릅니다. 이 규칙을 따르지 않으려는 경우 유틸리티가 제안한 별칭을 변경할 수 있습니다.
5. 출력 파일의 이름을 입력하십시오.

   디폴트값은 C:\Temp\nickname.txt입니다. 출력 파일을 받은 디렉토리는 이미 존재해야 하며 유틸리티를 실행하는 사용자가 쓸 수 있어야 합니다.

프롬프트에 응답한 후, 별칭 파일이 작성되고 텍스트 편집기에서 열립니다.

관련 개념:

- 60 페이지의 『Documentum 랩퍼를 위한 CreateNicknameFile 유틸리티의 정의』

관련 태스크:

- 43 페이지의 『Documentum 데이터 소스에 대한 별칭 등록』
- 61 페이지의 『CreateNicknameFile 유틸리티 설치(Documentum 랩퍼)』

Documentum 등록 테이블에서 DM_ID 오브젝트 유형 맵핑

CreateNicknameFile 유틸리티에 의해 작성된 컬럼 정의는 해당 DB2 데이터 유형에 대한 각 데이터 유형의 올바른 맵핑을 비롯하여 Documentum 랩퍼의 요구사항을 준수합니다. 유일한 예외는 Documentum이 등록된 테이블에서 DM_ID 데이터 유형을 지원하지 않는다는 것입니다. 유틸리티는 등록된 테이블에 있는 컬럼이 문자열로 정의된 경우 오브젝트 ID를 포함하기 위해 사용되며, 16문자 길이이며 "_id"로 끝나는 이름을 갖는다고 가정합니다. DM_ID 데이터 유형의 경우, 유틸리티는 컬럼을 DB2 CHAR(16) 데이터 유형으로 맵핑합니다. 다른 모든 경우에, 모든 문자열/varchar 컬럼은 DB2 VARCHAR 데이터 유형으로 맵핑됩니다.

절차:

올바른 데이터 유형 맵핑을 위해서는 다음을 수행하십시오.

1. CreateNicknameFile 유틸리티에 의해 작성된 출력 파일에서 컬럼 데이터 유형 정의를 조사하십시오.
2. 유틸리티가 Documentum 컬럼의 데이터 유형을 올바르지 않은 DB2 데이터 유형에 맵핑한 경우, 파일을 사용하여 별칭을 DB2에 등록하기 전에 DB2 데이터 유형을 변경하십시오.

관련 개념:

- 60 페이지의 『Documentum 랩퍼를 위한 CreateNicknameFile 유틸리티의 정의』

관련 태스크:

- 61 페이지의 『CreateNicknameFile 유틸리티 설치(Documentum 랩퍼)』

• 61 페이지의『CreateNicknameFile 유틸리티 구성(Documentum 랩퍼)』

반복 속성 이중 정의(Documentum 랩퍼)

랩퍼의 쿼리 성능을 최대화하려면, 각 속성이 참(true)과 대등한 DB2 데이터 유형으로 정의되어야 합니다. 즉, Documentum 정수는 DB2 정수 등으로 정의되어야 합니다. 그러나 이 정의는 비VARCHAR 반복 속성에 대한 다중 값을 리턴하지 못하게 합니다. 이러한 컬럼의 경우, index[0]의 값만이 리턴됩니다.

가능할 때마다 랩퍼는 Docbase 오브젝트당 오직 한 행의 결과만을 리턴하므로 이 제한사항이 존재합니다. 이 제한사항은 반복 속성이 선택될 때에만 나타납니다. 그러나 동일한 리모트 반복 속성에 대한 두 번째 컬럼을 정의할 수 있지만 데이터 유형은 VARCHAR입니다.

이 컬럼 이름은 모든 관련 값의 분리문자로 분리된 목록으로 모든 값을 리턴하기 위해 SELECT 목록에서 사용됩니다(각 컬럼의 DELIMITER 옵션은 사용되는 분리문자를 지정함).

다중 값 컬럼의 로컬 이름을 표준화해야 합니다. 참(true) 데이터 유형으로 정의된 컬럼의 로컬 이름에 접두부 ″m_″을 추가하여 각 다중 값 컬럼의 로컬 이름을 표준화할 수 있습니다.

예를 들어, 데이터 유형 TIMESTAMP로 정의된 approval_dates라는 Documentum 반복 속성의 별칭 컬럼을 갖는다고 가정하십시오. m_approval_dates라는 두 번째 별칭을 작성하고 VARCHAR 데이터 유형으로 정의할 수 있습니다. 그런 다음 SELECT 목록에서 m_approval_dates를 사용하여 분리문자 분리 목록에서 모든 승인 날짜를 리턴할 수 있습니다.

참(true) 데이터 유형이 VARCHAR인 반복 속성에 대해서는 이중 정의를 사용할 필요가 없습니다.

Documentum 랩퍼에 대한 제한사항 및 고려사항

이 절에는 Documentum 랩퍼의 사용과 연관된 제한사항 및 고려사항의 목록이 들어 있습니다.

- 반복 속성 값 리턴에 관련된 제한사항: 마지막 값만이 다음에 대해 리턴됩니다.
  - 비VARCHAR 반복 속성
  - ALL_VALUES 'N'이 지정되었을 때 VARCHAR 컬럼

  이 제한을 극복하기 위해, 반복 속성 컬럼에 대한 이중 정의를 작성할 수 있습니다. 반복 속성에 대한 이중 정의 작성의 자세한 정보는 아래의 관련 링크 절을 참조하십시오.

  또한, VARCHAR로 정의된 반복 속성의 여러 값이 하나의 분리문자로 분리된 문자열로 리턴됩니다. 분리문자는 CREATE NICKNAME문에 설정된 DELIMITER 별칭 옵션의 설정에 따라 다릅니다.
- Passthru는 지원되지 않습니다.
- DB2 응용프로그램에 의해 작성된 DB2 응용프로그램에 대한 각 연결의 경우, Documentum 랩퍼는 최대 10개의 동시 Documentum 세션을 지원하며, 그러한 각 세션은 최대 10개의 Documentum 쿼리를 동시에 관리할 수 있습니다. 단일 DB2 응용프로그램은 동시에 진행 중인 여러 쿼리를 가질 수 있으며, DB2로 쿼리가 제출될 때 쿼리의 수명이 시작되며 결과 세트에서 해당 커서가 닫힐 때 종료합니다. 주어진 시간에, 해당 시간에 진행 중인 전체 쿼리 세트에서 다음 제한사항이 유지되어야 합니다.
  - 모든 쿼리에 의해 참조되는 모든 별칭은 단지 10개의 다른 Documentum 서버에 상주해야 합니다.
  - 하나의 Documentum 서버에서 단지 10개의 별칭이 참조될 수 있습니다.

  둘 이상의 쿼리에서 언급되거나 단일 쿼리에서 여러번 참조되는 별칭은 나타날 때마다 한 번만 카운트됩니다.
- Documentum 랩퍼는 클라이언트 라이브러리의 AIX용 버전 3.1.7a를 사용합니다. Documentum 4i를 사용 중인 경우, Documentum에서 클라이언트 라이브러리의 이전 버전을 획득하는 것이 필요합니다(아직 설치되지 않은 경우).

- DB2가 Boolean 유형을 지원하지 않으므로 WHERE절에 사용된 대부분의 사용자 정의 기능(USER 제외)은 정수를 리턴하도록 정의되므로 ″=1″ 검사를 수행해야 합니다.

  예를 들어, 다음과 같습니다.

  ```
  "... WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1"
  ```
- DB2의 제한사항으로 인해 사용자 정의 기능 USER는 사용되지 않은 정수 인수와 함께 정의됩니다.
- DB2의 동일한 인스턴스에 대해 실행하는 모든 서버는 동일한 Documentum dmcl.ini 구성 매개변수를 공유해야 합니다.
- 반복 속성에 대한 ANY_IN 사용자 정의 기능에서 단일 명령문에 대한 값의 최대 수는 10입니다. 그러나 여러 명령문은 OR될 수 있습니다.
- ANY_SAME_INDEX 사용자 정의 기능의 경우 반복 속성의 동일 인덱스에서 값에 대한 테스트의 최대 수는 10입니다. 테스트는 왼쪽에서 오른쪽으로 평가되는 AND 테스트이어야 합니다.
- 랩퍼는 특정 코드 페이지에 대하여 특정한 성능을 갖지 않습니다.

관련 참조:

- 25 페이지의『테이블 구조 파일 랩퍼에 대한 랩퍼 제한사항 및 고려사항』
- 25 페이지의『테이블 구조 파일 랩퍼에 대한 파일 제한사항 및 고려사항』
- 83 페이지의『Excel 랩퍼에 대한 랩퍼 제한사항』
- 83 페이지의『Excel 파일 제한사항』
- 138 페이지의『XML 랩퍼에 대한 제한사항 및 고려사항』

Documentum 랩퍼에 대한 액세스 제어

쿼리는 Docbase에서 사용자 권한에 종속됩니다. 사용자가 적어도 읽기 액세스를 갖는 문서만이 쿼리 결과에 포함됩니다.

관련 참조:

- 26 페이지의『테이블 구조 파일 랩퍼에 대한 파일 액세스 제어 모델』
- 84 페이지의『Excel 랩퍼에 대한 파일 액세스 제어 모델』

Documentum 랩퍼에 대한 메시지

이 절에서는 Documentum의 랩퍼로 작업하는 중에 발생할 수 있는 메시지 목록을 나열하고 설명합니다. 메시지에 대한 자세한 정보는 *DB2 메시지 참조서*를 참조하십시오.

표 14. Documentum의 랩퍼가 발행한 메시지

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″sqlno_crule_save_plans [100]:rc (-2144272209) 비어 있는 플랜 목록이 발견되었습니다.″)	DB2에 제출된 SQL 쿼리를 랩퍼가 처리할 수 없습니다. 구문을 정정하고 다시 제출하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″dmAPI exec 실패: [DM_QUERY_E_BAD_QUAL] 오류: ″속성 <column_name>에 대한 속성 규정자 A0가 유효한 규정자가 아닙니다.″)	올바르지 않은 Documentum 유형 또는 등록된 테이블이 REMOTE_OBJECT 별칭 옵션에 대해 입력되었습니다. 올바른 Documentum 오브젝트 유형 또는 등록된 테이블을 사용하려면 별칭을 변경하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″올바르지 않은 널(NULL) 컬럼이 지정되었습니다.″)	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″Nickname 스펙이 비어 있습니다.″)	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″출력 오브젝트가 비어 있거나 불완전합니다.″)	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.

표 14. Documentum의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "예상치 못한 컬럼 수가 요청되었습니다.")	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다(이유 "컬럼 정보가 없습니다).	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "지원되지 않는 컬럼 유형이 요청되었습니다.")	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다(이유 "올바르지 않은 컬럼 정의").	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다(이유 "일치하지 않는 유형, DB2 요청 != 별칭 유형").	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "출력 매개변수가 널(NULL)이 아닙니다.")	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "쿼리 출력 변수가 널(NULL)이 아닙니다.")	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다(이유 "올바르지 않은 시간소인 길이").	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.

표 14. Documentum의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다(이유 ″일치하지 않는 컬럼 수″).	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다(이유: ″값을 변환할 때 데이터에 액세스할 수 없음″).	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″DMCL 클라이언트를 초기화하는 데 실패했습니다.″)	Documentum 클라이언트가 초기화할 수 없습니다. 시스템 관리자에게 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″Get_User가 널(NULL)을 리턴했습니다.″)	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″Get_Local_User가 널(NULL)을 리턴했습니다.″)	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″트랜잭션 시작에 실패했습니다.″)	Documentum이 begintrans에 실패했음을 보고했습니다. 시스템 관리자에게 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″입력 매개변수가 널(NULL)이 아닙니다.″)	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″Dctm 함수는 DCTM.function(...) =1과 유사해야 합니다.″)	사용자가 =1을 Dctm 함수에 대한 술어의 RHS로 사용하지 않았습니다. 구문을 정정하고 쿼리를 다시 실행하십시오.

표 14. Documentum의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "올바르지 않은 컬럼 번호가 요청되었습니다.")	내부 프로그래밍 오류. IBM 고객만족센터에 문의하십시오.
SQL1881N	"DELIMITER"는 "<column-name>"에 대해 유효한 "COLUMN" 옵션이 아닙니다.	DELIMITER 옵션이 컬럼 <column-name>에 대해 지정되었지만 IS_REPEATING 옵션이 지정되지 않았습니다.
SQL1882N	"SERVER" 옵션 "RDBMS_TYPE"을 "<server-name>"에 대한 "<option-value>"(으)로 설정할 수 없습니다.	RDBMS_TYPE 서버 옵션에 대해 지정된 값이 유효하지 않습니다. DB2, INFORMIX, ORACLE, SQLSERVER 또는 SYBASE 중 하나이어야 합니다.
SQL1882N	"SERVER" 옵션 "TRANSACTIONS"를 "<server-name>"에 대한 "<option-value>"(으)로 설정할 수 없습니다.	TRANSACTIONS 서버 옵션에 대해 지정된 값이 유효하지 않습니다. NONE, QUERY, PASSTHRU 또는 ALL 중 하나이어야 합니다.
SQL1882N	"NICKNAME" 옵션 "IS_REG_TABLE"을 "<nickname>"에 대해 "<option-value>"(으)로 설정할 수 없습니다.	IS_REG_TABLE 별칭 옵션에 대해 지정된 값이 유효하지 않습니다. 'Y' 또는 'N' 중 하나이어야 합니다.
SQL1882N	"NICKNAME" 옵션 "ALL_VERSIONS"를 "<nickname>"에 대해 "<option-value>"(으)로 설정할 수 없습니다.	ALL_VERSIONS 별칭 옵션에 대해 지정된 값이 유효하지 않습니다. 'Y' 또는 'N' 중 하나이어야 합니다.
SQL1882N	"SERVER" 옵션 "OS_TYPE"을 "<server-name>"에 대해 "<option-value>"(으)로 설정할 수 없습니다.	OS_TYPE 서버 옵션에 대해 지정된 값이 유효하지 않습니다. AIX, HPUX, SOLARIS 또는 WINDOWS이어야 합니다.
SQL1882N	"NICKNAME" 옵션 "FOLDERS"를 "<nickname>"에 대해 "<option-value>"(으)로 설정할 수 없습니다.	FOLDERS 별칭 옵션에 대해 지정된 값이 유효하지 않습니다. IS_REG_TABLE이 'Y'인 테이블에 대해 지정할 수 없습니다.
SQL1882N	"NICKNAME" 옵션 "VERSIONS"를 "<nickname>"에 대해 "<option-value>"(으)로 설정할 수 없습니다.	VERSIONS 별칭 옵션에 대해 지정된 값이 유효하지 않습니다. 'Y' 또는 'N' 중 하나이어야 합니다. 또한 VERSIONS 'Y'는 IS_REG_TABLE이 'Y'인 테이블에 대해 지정할 수 없습니다.

표 14. Documentum의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = "올바르지 않은 컬럼 이름, IS_REG_TABLE 또는 IS_REPEATING이 별칭에 지정되었습니다."	IS_REG_TABLE, IS_REPEATING, REMOTE_NAME 옵션 및 컬럼 이름의 올바른 스펙에 대해서는 별칭 명령문을 검사하십시오.
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = "db2dj.ini에서 DOCUMENTUM 또는 DMCL_CONFIG env var가 누락되었습니다."	필수 환경 변수가 설정되지 않았습니다. db2dj.ini 파일에서 설정하십시오.
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = "디버깅을 위해 로그 파일을 여는 데 실패했습니다."	문제점 해결에 사용된 로그 파일에 액세스할 수 없습니다. 시스템 관리자에게 문의하십시오.
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = "오직 하나의 검색 조건만이 지정될 수 있습니다."	오직 하나의 사용자 정의 검색 함수가 쿼리마다 지정될 수 있습니다.
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = "내용 디렉토리를 작성하는 데 실패했습니다."	DB2 에이전트에 의해 대상 디렉토리를 쓸 수 있는지 확인하십시오.
SQL30090N	응용프로그램 실행 환경에 대해 유효하지 않은 조작입니다. 이유 코드 = "내용 파일에서 사용 권한을 변경하는 데 실패했습니다."	db2 에이전트에 의해 목표 내용 디렉토리를 쓸 수 있는지 확인하십시오.

관련 참조:

- 27 페이지의『테이블 구조 파일 랩퍼에 대한 메시지』
- 84 페이지의『Excel 랩퍼에 대한 메시지』
- 116 페이지의『BLAST 랩퍼에 대한 메시지』
- 138 페이지의『XML 랩퍼에 대한 메시지』

제 5 장 데이터 소스로서의 Excel

이 장에서는 Excel의 정의와 사용자의 페더레이티드 시스템에 Excel 데이터 소스를 추가하는 방법을 설명하고, Excel 랩퍼와 연관된 오류 메시지를 나열합니다.

Excel의 정의

Excel 스프레드시트 또는 통합 문서는 Microsoft®(MS) Excel 응용프로그램을 사용하여 작성된 파일이며 파일 확장자가 xls입니다. DB2 Life Sciences Data Connect는 Excel 97 및 Excel 2000에서 스프레드시트를 지원합니다. 그림 6에서는 Excel 랩퍼가 스프레드시트를 페더레이티드 시스템에 연결하는 방법을 설명합니다.

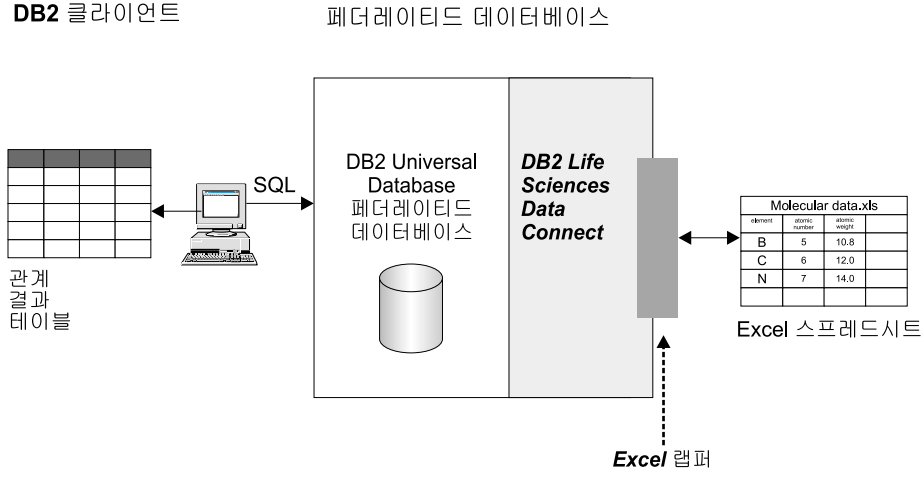


그림 6. Excel 랩퍼 작업 방법

Excel 랩퍼는 CREATE NICKNAME문을 사용하여 Excel 스프레드시트의 컬럼을 DB2® 페더레이티드 시스템의 컬럼으로 맵핑합니다. 74 페이지의 표 15에서는 Compound_Master.xls라는 파일에 저장되는 샘플 스프레드시트 데이터를 표시합니다.

표 15. Compound_Master.xls의 샘플 스프레드시트

	A	B	C	D
1	compound_A	1.23	367	tested
2	compound_G		210	
3	compound_F	0.000425536	174	tested
4	compound_Y	1.00256		tested
5	compound_Q		1024	
6	compound_B	33.5362		
7	compound_S	0.96723	67	tested
8				
9	compound_O	1.2		tested

이 정보는 보통 표준 SQL 명령을 통해서는 사용 불가능합니다. Excel 랩퍼가 설치 및 등록되면, 표준 관계형 데이터 소스에 있는 것처럼 이 정보에 액세스할 수 있습니다. 예를 들어, 분자 수가 100보다 큰 모든 화합물 데이터를 알려는 경우, 다음 SQL 쿼리를 실행합니다.

```
SELECT * FROM compound_master WHERE mol_count > 100
```

쿼리 결과는 표 16에 표시됩니다.

표 16. 쿼리 결과

COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
compound_A	1.23	367	tested
compound_G		210	
compound_F	0.000425536	174	tested
compound_Q		1024	

관련 개념:

- 13 페이지의 『테이블 구조 파일의 정의』
- 31 페이지의 『Documentum의 정의』
- 91 페이지의 『BLAST의 정의』
- 119 페이지의 『XML의 정의』

관련 태스크:

- 75 페이지의『페더레이티드 시스템에 Excel 추가』

관련 참조:

- 75 페이지의『Excel 랩퍼에 대한 전제조건』

Excel 랩퍼에 대한 전제조건

Excel 데이터 소스 랩퍼를 활용하기 위한 전제조건은 다음과 같습니다.

- Excel 랩퍼를 활용하려면 먼저 DB2 Life Sciences Data Connect가 설치된 서버에 MS Excel 응용프로그램을 설치해야 합니다.

관련 태스크:

- 75 페이지의『페더레이티드 시스템에 Excel 추가』
- 76 페이지의『Excel 랩퍼 등록』

페더레이티드 시스템에 Excel 추가

절차:

페더레이티드 시스템에 Excel 데이터 소스를 추가하려면 다음을 수행하십시오.

1. CREATE WRAPPER문을 사용하여 랩퍼를 등록하십시오.
2. CREATE SERVER문을 사용하여 서버를 등록하십시오.
3. 액세스하려는 각 Excel 스프레드시트에 대해 CREATE NICKNAME문을 사용하여 별칭을 등록하십시오.

이 명령은 DB2 명령행 처리기에서 실행될 수 있습니다.

관련 태스크:

- 76 페이지의『Excel 랩퍼 등록』
- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』
- 33 페이지의『페더레이티드 시스템에 Documentum 추가』

- 96 페이지의『페더레이티드 시스템에 BLAST 추가』
- 124 페이지의『페더레이티드 시스템에 XML 추가』

Excel 랩퍼 등록

이 태스크는 *페더레이티드 시스템에 Excel 추가*에 대한 기본 태스크의 일부입니다. 데이터 소스에 액세스하려면 랩퍼를 등록해야 합니다. 랩퍼는 페더레이티드 서버가 통신을 하고 데이터 소스에서 데이터를 검색하는 데 사용하는 메커니즘입니다. 랩퍼는 라이브러리 파일로 시스템에 설치됩니다.

절차:

Excel 데이터 소스 랩퍼를 등록하려면, CREATE WRAPPER문을 제출하십시오.

라이브러리 파일 db2lsxls.dll을 사용하여 Excel_9x_Wrapper라고 하는 Excel 97에 대한 Excel 랩퍼를 작성하려면 다음의 명령문을 제출하십시오.

```
CREATE WRAPPER Excel_9x_Wrapper LIBRARY 'db2lsxls.dll'
OPTIONS(DB2_FENCED 'N');
```

CREATE WRAPPER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *Excel 데이터 소스에 대한 서버 등록*입니다.

관련 태스크:

- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』

관련 참조:

- 75 페이지의『Excel 랩퍼에 대한 전제조건』

Excel 데이터 소스에 대한 서버 등록

이 태스크는 *페더레이티드 시스템에 Excel 추가*에 대한 기본 태스크의 일부입니다. 랩퍼를 등록한 후에 해당 서버를 등록해야 합니다.

절차:

페더레이티드 시스템에 Excel 서버를 등록하려면 CREATE SERVER문을 사용하십시오.

예를 들어, CREATE WRAPPER문을 사용하여 작성된 Excel_2000_Wrapper 랩퍼에 대해 서버를 등록하는 biochem_node1 노드 이름을 사용하여 biochem_lab이라고 하는 서버를 작성하려면 다음의 명령문을 제출하십시오.

```
CREATE SERVER biochem_lab WRAPPER Excel_2000_Wrapper;
```

인수 정의

WRAPPER

연관된 CREATE WRAPPER문에 등록한 랩퍼의 이름을 지정합니다. 이 인수는 필수입니다.

CREATE SERVER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *Excel 데이터 소스에 대한 별칭 등록*입니다.

관련 태스크:

- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 76 페이지의『Excel 랩퍼 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』
- 127 페이지의『XML 데이터 소스에 대한 서버 등록』

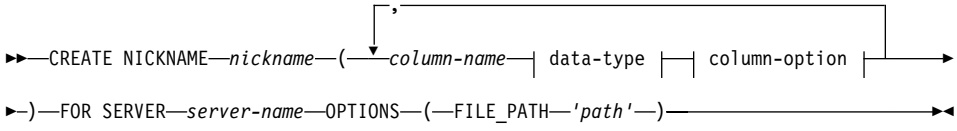
Excel 데이터 소스에 대한 별칭 등록

이 태스크는 *페더레이티드 시스템에 Excel 추가*에 대한 기본 태스크의 일부입니다. 서버를 등록한 후에 해당 별칭을 등록해야 합니다. 별칭은 쿼리에서 Excel 데이터 소스를 참조할 때 사용됩니다.

절차:

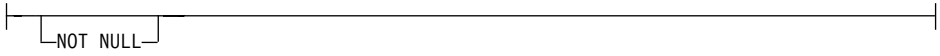
Excel 데이터 소스를 관계형 테이블로 맵핑하려면, CREATE NICKNAME문을 사용하여 별칭을 작성하십시오.

CREATE NICKNAME 구문(Excel용)



data-type:

column-option:



CREATE NICKNAME문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

FOR SERVER

연관된 CREATE SERVER문에 등록한 서버를 식별합니다. 이 서버는 Excel 스프레드시트를 액세스하기 위해 사용됩니다. 서버 이름을 지정하십시오.

옵션 정의

FILE_PATH

액세스하려는 Excel 스프레드시트의 완전한 디렉토리 경로 및 파일 이름을 지정합니다.

다음 예의 명령문은 이름이 CompoundMaster.xls인 Excel 스프레드시트 파일에서 Compounds 별칭을 작성합니다. 파일에는 페더레이티드 시스템에 정의되는 세 개의 데이터 컬럼인 Compound_ID, CompoundName 및 MolWeight가 들어 있습니다.

```
CREATE NICKNAME Compounds (
Compound_ID INTEGER,
CompoundName VARCHAR(50),
MolWeight FLOAT)
FOR SERVER biochem_lab
OPTIONS(FILE_PATH 'C:\My Documents\CompoundMaster.xls');
```

이 태스크 다음에는 더 이상의 태스크가 없습니다.

관련 태스크:

- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』
- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』
- 145 페이지의 제 8 장『비용산정 별칭 옵션 지정』

Excel 데이터 소스에 대한 쿼리 실행

이 절에서는 별칭 예인 Compounds를 사용하는 여러 샘플 Excel 스프레드시트 쿼리를 나열합니다.

절차:

쿼리를 실행하려면 일반 테이블 이름 및 테이블 컬럼을 사용할 때와 동일한 방식으로 SQL문에 별칭과 정의된 별칭 컬럼을 사용하십시오.

다음의 쿼리는 분자량이 2000보다 큰 모든 compound_ID를 표시합니다.

```
SELECT compound_ID
FROM Compounds
WHERE MolWeight > 200;
```

다음 쿼리는 화합물 이름 또는 분자량이 널(NULL)인 모든 레코드를 표시합니다.

```
SELECT *
FROM Compounds
WHERE CompoundName IS NULL
OR MolWeight IS NULL;
```

다음의 쿼리는 화합물 이름에 문자열 ase가 포함되었으며 분자량이 300 이상인 모든 레코드를 표시합니다.

```
SELECT *
FROM Compounds
WHERE CompoundName LIKE '%ase%
AND MolWeight >=300;
```

관련 태스크:

- 59 페이지의『Documentum 데이터 소스에 대한 쿼리 실행』
- 80 페이지의『샘플 Excel 랩퍼 시나리오』
- 136 페이지의『XML 데이터 소스에 대한 쿼리 실행』

샘플 Excel 랩퍼 시나리오

이 절에서는 C:\Data 디렉토리에 위치한 Excel 2000 스프레드시트에 액세스하는 Excel_2000 랩퍼의 샘플 구현을 설명합니다. 시나리오는 랩퍼를 등록하며, 서버를 등록하며, 스프레드시트를 액세스하는 데 사용할 하나의 별칭을 등록합니다. 시나리오에 표시된 명령문은 DB2 명령행 처리기를 사용하여 입력됩니다. 랩퍼를 등록한 후에, 스프레드시트에 대한 쿼리를 실행할 수 있습니다.

시나리오는 네 개의 컬럼과 아홉 개의 행이 있는 Compund_Master.xls라고 하는 복합 스프레드시트에서 시작합니다. 파일에 대한 완전한 경로 이름은 C:\Data\Compound_Master.xls입니다. 내용은 표 17에 표시됩니다.

표 17. Compound_Master.xls 샘플 스프레드시트

	A	B	C	D
1	compound_A	1.23	367	tested
2	compound_G		210	
3	compound_F	0.000425536	174	tested
4	compound_Y	1.00256		tested

표 17. Compound_Master.xls 샘플 스프레드시트 (계속)

	A	B	C	D
5	compound_Q		1024	
6	compound_B	33.5362		
7	compound_S	0.96723	67	tested
8				
9	compound_O	1.2		tested

절차:

Excel 랩퍼를 사용하여 Excel 2000 스프레드시트에 액세스하려면 다음과 같이 하십시오.

1. Excel_2000 랩퍼를 등록하십시오.

```
db2 => CREATE WRAPPER Excel_2000 LIBRARY 'db2lsxls.dll'
OPTIONS(DB2_FENCED 'N')
```

2. 서버를 등록하십시오.

```
db2 => CREATE SERVER biochem_lab WRAPPER Excel_2000
```

3. Excel 스프레드시트를 나타내는 별칭을 등록하십시오.

```
db2 => CREATE NICKNAME Compound_Master (compound_name VARCHAR(40),
weight FLOAT, mol_count INTEGER, was_tested VARCHAR(20))
FOR biochem_lab
OPTIONS ( FILE_PATH 'C:\Data\Compound_Master.xls')
```

등록 프로세스가 완료되었습니다. Excel 데이터 소스는 이제 페더레이티드 시스템의 일부이며, SQL 쿼리에서 사용될 수 있습니다.

다음 예는 Excel 데이터 소스를 사용하여 획득한 샘플 SQL 쿼리 및 결과를 표시합니다.

- 샘플 SQL 쿼리: "mol_count가 100보다 큰 모든 화합물 데이터를 제공하십시오"

```
SELECT * FROM compound_master WHERE mol_count > 100
```

  결과: 행 1, 2, 3, 5 및 7에 대한 모든 필드.

- 샘플 SQL 쿼리: "mol_count가 아직 결정되지 않은 모든 화합물에 대해 compound_name 및 mol_count를 제공하십시오."

```
SELECT compound_name, mol_count FROM compound_master
WHERE mol_count IS NULL
```

결과: 스프레드시트로부터 행 4, 6, 9의 필드 compound_name 및 mol_count.

- 샘플 SQL 쿼리: "테스트되지 않았으며 무게가 1보다 큰 화합물의 수를 카운트하십시오."

```
SELECT count(*) FROM compound_master
WHERE was_tested IS NULL AND weight > 1
```

결과: 기준과 일치하는 스프레드시트에서 단일 행 6을 나타내는 레코드 카운트 1.

- 샘플 SQL 쿼리: mol_count가 결정되었으며 평균 mol_count보다 작은 모든 화합물에 대해 compound_name 및 mol_count를 제공하십시오.

```
SELECT compound_name, mol_count
FROM compound_master
WHERE mol_count IS NOT NULL
AND mol_count < (SELECT AVG(mol_count) FROM compound_master
                 WHERE mol_count IS NOT NULL AND was_tested IS NOT NULL)
```

서브쿼리는 표 18을 리턴하는 주 쿼리로 평균 368을 리턴합니다.

표 18. 쿼리 결과

COMPOUND_NAME	MOL_COUNT
compound_A	367
compound_G	210
compound_F	174
compound_S	67

관련 태스크:

- 75 페이지의 『페더레이티드 시스템에 Excel 추가』
- 79 페이지의 『Excel 데이터 소스에 대한 쿼리 실행』

Excel 랩퍼에 대한 랩퍼 제한사항

- Excel 랩퍼는 DB2 Universal Database Enterprise Server Edition을 지원하는 Microsoft Windows 운영 체제에서만 사용 가능합니다.
- Passthru 세션은 Excel 랩퍼에서 허용되지 않습니다.
- Excel 스프레드시트 데이터는 읽기만 가능하며 쓸 수 없습니다.
- 랩퍼가 지원하는 DATE 데이터 유형의 날짜 범위는 1970년 1월 1일부터 2038년 1월 18일까지입니다.

관련 참조:

- 25 페이지의『테이블 구조 파일 랩퍼에 대한 랩퍼 제한사항 및 고려사항』
- 25 페이지의『테이블 구조 파일 랩퍼에 대한 파일 제한사항 및 고려사항』
- 65 페이지의『Documentum 랩퍼에 대한 제한사항 및 고려사항』
- 83 페이지의『Excel 파일 제한사항』
- 138 페이지의『XML 랩퍼에 대한 제한사항 및 고려사항』

Excel 파일 제한사항

- 데이터 유형은 각 컬럼 내에서 일치되어야 하며 컬럼 데이터 유형은 별칭 등록 프로세스 중 제대로 기술되어야 합니다.
- Excel 랩퍼는 Excel 워크북 내에서 기본 스프레드시트에만 액세스할 수 있습니다.
- 스프레드시트에서 공백 셀은 널(NULL)로서 해석됩니다.
- 최대 10개의 연속 공백 행이 스프레드시트에 존재할 수 있으며 데이터 세트에 포함될 수 있습니다. 11개 이상의 연속 공백 행은 데이터 세트의 끝으로 해석됩니다.
- 공백 컬럼이 스프레드시트에 존재할 수 있습니다. 그러나 이 컬럼이 사용되지 않는 경우일지라도 유효한 필드로서 등록 및 설명되어야 합니다.

관련 참조:

- 83 페이지의『Excel 랩퍼에 대한 랩퍼 제한사항』

Excel 랩퍼에 대한 파일 액세스 제어 모델

데이터베이스 관리 시스템은 DB2 데이터베이스 서비스의 LOG ON AS 등록 정보 권한으로 Excel 파일에 액세스합니다. 이 설정은 DB2 인스턴스에 대한 LOG ON 등록 정보에서 볼 수 있습니다. 등록 정보 페이지는 Windows NT 서비스 제어판을 통해 액세스됩니다.

관련 참조:

- 26 페이지의『테이블 구조 파일 랩퍼에 대한 파일 액세스 제어 모델』
- 66 페이지의『Documentum 랩퍼에 대한 액세스 제어』

Excel 랩퍼에 대한 메시지

이 절에서는 Excel의 랩퍼로 작업하는 중에 발생할 수 있는 메시지 목록을 나열하고 설명합니다. 메시지에 대한 자세한 정보는 *DB2 메시지 참조서*를 참조하십시오.

표 19. Excel의 랩퍼가 발행한 메시지

오류 코드	메시지	설명
SQL1817N	CREATE SERVER문은 페더레이티드 데이터베이스에 정의되게 하려는 데이터 소스의 ″VERSION″을 식별하지 않습니다.	VERSION 매개변수는 CREATE SERVER 문 실행 중 지정되지 않았습니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1000.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″메모리 할당 오류″입니다.	IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1001.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″알 수 없는 옵션″입니다.	DDL문에 지정된 옵션은 지원되지 않습니다. SQL문을 정정하고 다시 실행하십시오.

표 19. Excel의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1002.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "DELTA 오브젝트 작성에 실패함"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1100.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "랩퍼 옵션이 지원되지 않음"입니다.	랩퍼 OPTIONS는 이 랩퍼에서 지원되지 않습니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1200.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "<option>이 지원되지 않는 서버 옵션"입니다.	지정한 옵션은 이 랩퍼에서 지원되지 않습니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1201.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "서버 이름 확보 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1209.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 " VARCHAR 데이터 변환 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1211.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "INTEGER 데이터 변환 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1212.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "FLOAT 데이터 변환 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.

표 19. Excel의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1400.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″<option>이 지원되지 않는 사용자 옵션″입니다.	지정한 옵션은 이 랩퍼에서 지원되지 않습니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1401.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″USER 델타 오브젝트 작성에 실패함″입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1500.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″<option>이 지원되지 않는 별칭 옵션″입니다.	지정한 옵션은 이 랩퍼에서 지원되지 않습니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1501.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″필수 옵션 PATH가 지정되지 않았습니다.″입니다.	PATH 옵션이 NICKNAME을 등록하는 데 필요합니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1502.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″NICKNAME 델타 오브젝트 작성에 실패함″입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1503.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″별칭 컬럼 유형 확보 중 오류″입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1504.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″별칭 컬럼 유형 이름 확보 중 오류″입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.

표 19. Excel의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1505.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "데이터 소스 "Excel 랩퍼"에서 수신되었습니다."입니다.	지정한 <data type>은 이 랩퍼에서 지원되지 않습니다. SQL문을 정정하고 다시 실행하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1506.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "별칭 컬럼 정보 확보 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1507.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "<option> 옵션을 제거할 수 없습니다."입니다.	지정된 옵션이 필수 옵션이므로 제거할 수 없습니다.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1508.VANI"를 받았습니다. 연관된 텍스트 및 토큰은 "컬럼 이름을 변경할 수 없습니다."입니다.	컬럼 이름의 변경은 Excel 랩퍼에 의해 허용되지 않습니다.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1701.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "SQL 구문 분석 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1702.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "NICKNAME 오브젝트 액세스 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1703.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "데이터 스토리지 빌드 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.

표 19. Excel의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1704.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "SQL을 별칭 데이터에 링크 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1705.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "Excel 응용프로그램 시작에 실패했습니다."입니다.	Excel 응용프로그램의 시작에 실패했습니다. Excel이 시스템에 설치되어 랩퍼의 올바른 버전으로 등록되었는지 확인하십시오. Windows NT 서비스 제어판에서 DB2 인스턴스에 대한 LOG ON AS 등록 정보를 확인하십시오. Excel 응용프로그램이 이 권한을 사용하여 액세스됩니다. 이 사용자가 해당 권한을 가지고 있는지 확인하거나 이 등록 정보를 권한 부여된 계정으로 변경한 다음, DB2를 재시작하고 SQL 쿼리를 다시 실행하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1706.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "소스 스프레드시트를 여는 중 오류"입니다.	SQL 쿼리에서 별칭으로 참조된 스프레드시트를 여는 중 문제점이 발생했습니다. 등록 중 CREATE NICKNAME문 실행 중 지정된 PATH에 파일이 존재하는지 확인하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1707.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "DL 출력 스토리지 액세스 중 오류"입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1708.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "Excel 응용프로그램 종료에 실패했습니다."입니다.	내부 프로그램 오류가 발생했습니다. 반복된 쿼리 이후에도 이 오류가 지속되는 경우, IBM 고객만족센터에 문의하십시오.
SQL1822N	데이터 소스 "Excel 랩퍼"에서 예기치 않은 오류 코드 "-1711.<internal program code>"를 받았습니다. 연관된 텍스트 및 토큰은 "페치 중 오류, 데이터/컬럼 유형이 불일치할 수 있음"입니다.	별칭 등록 중 지정된 데이터 유형과는 다른 데이터 유형의 SQL 쿼리 중 데이터가 페치되었습니다. 소스 스프레드시트에서 데이터를 정정하거나 별칭에서 등록된 데이터 유형을 정정하십시오. 이렇게 하여 문제점이 정정되지 않는 경우, IBM 고객만족센터에 문의하십시오.

표 19. Excel의 랩퍼가 발행한 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 ″Excel 랩퍼″에서 예기치 않은 오류 코드 ″-1900.<internal program code>″를 받았습니다. 연관된 텍스트 및 토큰은 ″메모리 할당 오류″입니다.	내부 프로그램 오류가 발생했습니다. IBM 고객만족센터에 문의하십시오.

관련 참조:

- 27 페이지의『테이블 구조 파일 랩퍼에 대한 메시지』
- 67 페이지의『Documentum 랩퍼에 대한 메시지』
- 116 페이지의『BLAST 랩퍼에 대한 메시지』
- 138 페이지의『XML 랩퍼에 대한 메시지』

제 6 장 데이터 소스로서의 BLAST

이 장에서는 BLAST의 정의와 사용자의 페더레이티드 시스템에 BLAST 데이터 소스를 추가하는 방법을 설명하고, BLAST 랩퍼와 연관된 오류 메시지들을 나열합니다.

BLAST의 정의

BLAST(Basic Local Alignment Search Tool)는 NCBI(National Center for Biotechnology Information)에 의해 유지보수되는 유틸리티입니다. BLAST는 ″히트(hit)″에 대한 뉴클레오타이드 또는 아미노산 배열 데이터베이스를 스캔하는 데 사용됩니다. BLAST 히트에는 하나 이상의 HSP(high-scoring segment pair)가 들어 있습니다. HSP는 배열 조각의 쌍으로 로컬로 최대한 맞추어져 있으며 유사 점수가 임계값을 초과합니다. NCBI는 GenBank 및 SWISS-PROT과 같은 BLAST 가능한 데이터 소스에 대해 BLAST 검색을 수행하는 데 사용되는 실행 파일 blastall을 제공합니다.

BLAST 랩퍼는 다섯 가지의 BLAST 검색 유형인 BLASTn, BLASTp, BLASTx, tBLASTn 및 tBLASTx를 모두 지원합니다. 이러한 유형들은 표 20에 설명되어 있습니다.

표 20. BLAST 랩퍼가 지원하는 BLAST 검색 유형

BLAST 검색 유형	**설명**
BLASTn	뉴클레오타이드 배열을 뉴클레오타이드 배열 데이터베이스의 내용과 비교하여 영역이 원래 배열의 영역과 일치하는 배열을 찾는 BLAST 검색 유형.
BLASTp	아미노산 배열을 아미노산 배열 데이터베이스의 내용과 비교하여 영역이 원래 배열의 영역과 일치하는 배열을 찾는 BLAST 검색 유형.
BLASTx	뉴클레오타이드 배열을 아미노산 배열 데이터베이스의 내용과 비교하여 영역이 원래 배열의 영역과 일치하는 배열을 찾는 BLAST 검색 유형. 쿼리 배열은 여섯 개의 모든 읽기 프레임으로 변환되며 각 결과 배열을 사용하여 배열 데이터베이스를 검색합니다.

표 20. BLAST 랩퍼가 지원하는 BLAST 검색 유형 (계속)

BLAST 검색 유형	설명
tBLASTn	아미노산 배열을 뉴클레오타이드 배열 데이터베이스의 내용과 비교하여 영역이 원래 배열의 영역과 일치하는 배열을 찾는 BLAST 검색 유형. 배열 데이터베이스의 배열은 여섯 개의 모든 읽기 프레임으로 변환되며 결과 배열에서 쿼리 배열의 영역과 일치하는 영역을 검색합니다.
tBLASTx	뉴클레오타이드 배열을 뉴클레오타이드 배열 데이터베이스의 내용과 비교하여 영역이 원래 배열의 영역과 일치하는 배열을 찾는 BLAST 검색 유형. tBLASTx 검색에서 쿼리 배열 및 배열 데이터베이스는 여섯 개의 모든 읽기 프레임으로 변환되며 결과 배열을 비교하여 일치하는 영역을 찾습니다.

그림 7은 페더레이티드 시스템에 대한 BLAST의 작업 방식을 나타냅니다.

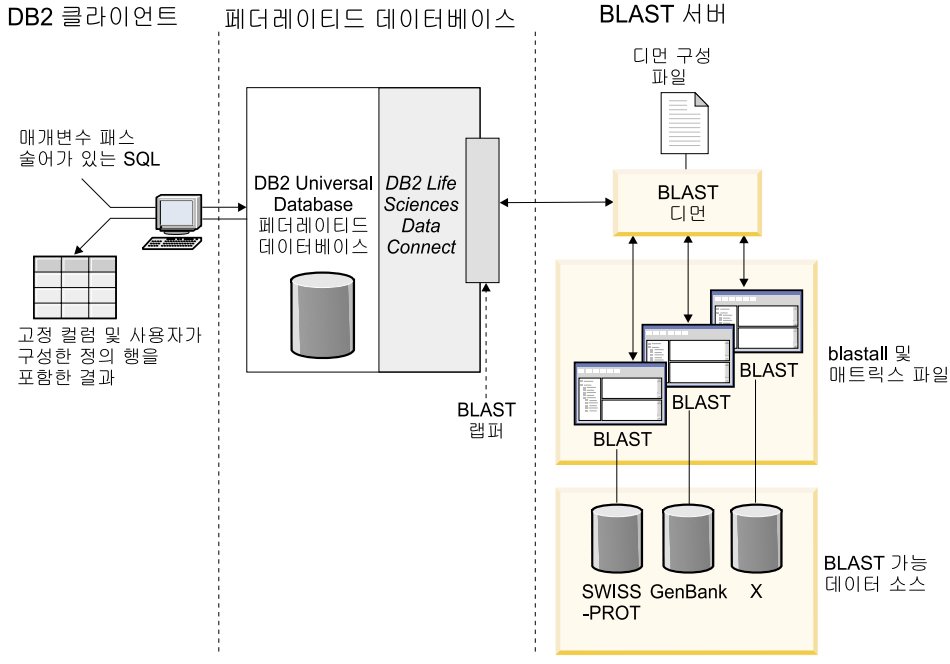


그림 7. BLAST 랩퍼의 작업 방식

클라이언트 측에서 사용자 또는 응용프로그램은 표준 BLAST 옵션에 맵핑하는 BLAST 특정 매개변수 패스 술어가 있는 SQL문을 제출합니다. 입력 술어가 있는 SQL문은 BLAST 랩퍼가 설치된 DB2® Universal Database 페더레이티드 데이터베이스 시스템으로 전송됩니다.

BLAST 랩퍼는 쿼리를 BLAST 응용프로그램이 이해할 수 있는 형식으로 변환하며 변환된 쿼리를 BLAST 서버에 전송합니다. 이 서버는 페더레이티드 시스템이 있는 머신과는 별도의 머신일 수 있습니다. 특수 디먼 프로그램이 BLAST 서버에서 실행됩니다. 디먼 구성 파일의 정보를 사용하는 이 디먼은 페더레이티드 시스템으로부터 쿼리 요청을 받고 BLAST 응용프로그램에 전송합니다. 그러면 BLAST 응용프로그램은 일반적인 방식으로 BLAST 가능한 데이터 소스에 대해 실행됩니다.

결과는 BLAST로 리턴된 다음 디먼에 리턴됩니다. 디먼은 검색된 데이터를 BLAST 랩퍼에 리턴합니다. 랩퍼는 데이터를 관계 테이블 형식으로 변환하고 이 테이블을 사용자 또는 응용프로그램에 리턴합니다. 리턴된 데이터는 두 부분으로 이루어져 있습니다.

- BLAST 사용자에게 익숙한 일련의 표준 고정 컬럼
- 사용자가 구성한 정의 행 정보

다음의 예는 관계 정보를 BLAST 가능 데이터 소스로부터 추출하는 방법을 보여줍니다. 데이터는 원시 Fasta 파일 형식에서 BLAST 가능 데이터 세트로 이동된 다음 페더레이티드 시스템의 다른 데이터 소스와 결합시킬 수 있는 관계 테이블로 다시 이동됩니다.

94 페이지의 그림 8은 네 개의 정의 행 및 뉴클레오타이드 배열 레코드가 들어 있는 샘플 Fasta 파일입니다.

```
>7:4986 PMON5744
GTTCTTCCCAGTGCCCAAGTCCATTCTGACATCAATGAAGAAGGTAAAATCCCTGCGTGATCCCTCTGCC
AAGATGTCGAAATCAGACCCGGATAAACTAGCTGCTGTCAGAATAACAGACAGCCCGGAGGAGATCGTGC
AGAAGTTCCGCAAGGCTGTGACGGACTTCACCTCGGAGGTCACCTACGACCCGGCCAGGCGAGGAGGCGT
GTCCAACTTGGTGGCCATCCACGCGGCAGTGACCGGACTCCCGGTGGAGGAGGTGGTCCGCCGAAGTGCT
GGCATCAACACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTGCACCAATTAAGA
GTGAAATTGAAAAACTGAAGAGGAACAAGGACCACCTAGAGAAGGTTTTACAAGTTGGGTCGGCAAAAGC
CAAAGAATTAGCATATCCCGTGTGCCAGGAGGTGAAGAAATTGGTGGGGTTTCTATAGGCAGTCTCACCT
AGTCCCAGAAAATGTTTTTTATCTTGTGGTCTGCTTGCACACTCAGTCTAATAAAGGCAGCTTTCCTAAG
ACGCCAACAATTCCAGTTTGGGGATGCTTAGTTTACT
>8:9747 PMON5699
AAGAAGTTCTTGTTAGAACTTTCCACCTCCGGCTTCCCCTCCACCTCTCTTACTGTCCCAACCTTCTGAG
ACGCTTTTTCTCCTCCCGAGGATTTATCTCTTTCTCTCTCTCTCTCTCTCTCTCTTTTTTTTTTTCCCCT
TTTCCCCCCCCGAGGCTGGTTTTGCTTTGGGGAGGGGGGGTTTTTTAAAGGGGCCGGGGGGGCCCCCTTT
CTCCCCCCTAATGGGGTTAATTAATAATGGGGGGGGGGGTTTTTTTTTTTTTAAACCCCTATTTGGTCCGG
CCCGGGGATTTCCCCCCCCCCCCCCTTGCCCGGTTCCGGGGCCCGGAGGAGGGGGGGAAAAGGGCGGGAA
CCTTTGGTAGTTTCCCCTCGGAAAAAAATTTTTCGGGGGGGAAAACCTCCCT
>13:6512 PMON5498
GATAAGAGGCAGAATAGAAGACTGGACTACTTCTCTCCTAAAAACACATTTAAAACTAAGCCTGAGCAAT
CTCCACCCAAATGGACCGGAAACCTTAAAAAAGAATCCTACTCCTGAAGAAAAAGAGGAGGACACATCAA
GAGGTAGAAGGGGCGATTTCATGATATAAACAACCCCATACCTCCAGAGTGGGAAGCTCCACAGACTGAA
AACTAACTGGTTCACAGAAACTCACCTACAGGAGTGAGCCCCACATCAAACCCTCGAATGTGGGGATCTG
GCACTGGTAGAAAGAGCCCCTGGAGCATCTGGCATTGAAGGCCAGTGGGGCTTGTGTGCAGGAGATCCAC
AGGACTAGGGGAAACGGAGACCCCCATTCTTAAAAGGTGCACACAGACTTTTACGTGCACTGGGTCCCAG
TGCAAAGCAAAGTCTCCATAGGAATCTGGGTCAAACCTGACTGCAGTTCTTGGAGGACCTCCTGGGAAAG
CAAGGGTGAATGTGGCTTCTTGTGGGGAAAGGACATTGGAAGCAAAGCTCTTGGGAATATTCATCAGTGT
GC
>15:8924 PMON5426
GGAGAAACTGACTCCTGAGCAGCTGCAATTCATGCGGCAGGTGCAGCTCGCCCAGTGGCAGAAGACGCTG
CCACAGCGGCGGACCCGGAACATCGTGACCGGCCTGGGCATCGGGGCGCTGGTGTTGGCAATTTGTATCC
GTTTGGACTGTAGACTCAGGGAGACCGCATTTAGGGGAACAGGAAGGGCAGCAGGGGCGTGTAGGAGGGC
AGTGTGGGGGTGGTAGAAGGAGCCCGAGATATGAAAACCTTGGCTCCTTTTAACTCTGAATCAAGCGTTT
GGTGTACCTTACGTTGTCATTTTAAAGGTGTATTTTAGTATAATTGATTAATGATTACGGAGTCGGGTGA
GGGCTCCCAGGAGCAGACGGCAGAAGATCGAATTTGGGAGGATGATCAGCAGCGGTGGTTGAGCAAGTGT
GGGAAAAGGGAATGCGCACATTCCACGTGGTTTCCTGAACCCACCTCCCCAGATGGTTACACCTTCTACT
CGGTGTCCCAGGAGCGTTTCTTGGATGAGCTGGAGGATGAGGCCAAAGCTGCTC
```

그림 8. 샘플 Fasta 파일, nucleo1

표준 formatdb 응용프로그램은 Fasta 파일을 BLAST 가능 데이터 세트로 변환합니다. 이제 데이터는 BLAST 랩퍼를 설치 및 등록하여 페더레이티드 시스템을 통해 SQL이 쿼리할 수 있도록 준비되어 있습니다.

클라이언트 측의 사용자 또는 응용프로그램이 보낸 다음의 쿼리는 BLAST 랩퍼에 의해 변환됩니다. 그런 다음 BLAST 가능 데이터 세트에 대해 실행됩니다.

```
SELECT Unique_ID, Experiment_Number, Organism_Number, HSP_Info, Score
FROM nucleo1
WHERE BlastSeq = 'ACATTCTTATAGAGTATTGCTACTCCTCCAGGATAGAGTCATCTCT
GGTCTCCAGAGCCACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTG
CACCAATACAGAAACTCACCTACAGGAGTGAGCGGGTGGTAGAAGGAGCCCGAGATATGAAA
ACCTTGTTTCAAGACCCCATTGTCACCGGGG';
```

쿼리의 결과는 BLAST 랩퍼에 의해 표 21에 표시된 관계 테이블 형식으로 변환됩니다.

표 21. BLAST는 페더레이티드 시스템에 통합될 때 관계 테이블 형식으로 결과를 리턴합니다.

고유 ID	실험 번호	유기체 번호	HSP_INFO	SCORE
PMON5744	4986	7	Identities = 57/201 (28%), Positives = 57/201(28%), Gaps = 0/201(0%)	+1.13487000000000E+002
PMON5426	8924	15	Identities = 35/201 (17%), Positives = 35/201(17%), Gaps = 0/201(0%)	+6.98754000000000E+001
PMON5498	6512	13	Identities = 26/201 (13%), Positives = 26/201(13%), Gaps = 0/201(0%)	+5.20342000000000E+001

데이터는 완전한 관계 형식으로 이루어져 있으며 연구실에서 사용하는 다른 데이터 소스의 데이터와 결합시킬 수 있습니다. 여러 데이터 소스의 결과를 결합하면 페더레이티드 시스템을 구현하기 전에는 쉽게 또는 효율적으로 알아 볼 수 없습니다.

관련 개념:

- 13 페이지의『테이블 구조 파일의 정의』
- 31 페이지의『Documentum의 정의』
- 73 페이지의『Excel의 정의』

- 119 페이지의『XML의 정의』

관련 태스크:

- 96 페이지의『페더레이티드 시스템에 BLAST 추가』

페더레이티드 시스템에 BLAST 추가

절차:

페더레이티드 서버에 BLAST 데이터 소스를 추가하려면 다음과 같이 하십시오.

1. 올바른 버전의 blastall 실행 파일과 매트릭스 파일이 설치되어 있는지 확인하십시오.
2. BLAST 디먼을 구성하십시오.
3. BLAST 디먼을 시작하십시오.
4. CREATE WRAPPER문을 사용하여 랩퍼를 등록하십시오.
5. 선택적: 쿼리 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오.
6. CREATE SERVER문을 사용하여 서버를 등록하십시오.
7. CREATE NICKNAME문을 사용하여 별칭을 등록하십시오.

이 명령문은 DB2 명령행 처리기에서 실행할 수 있습니다. BLAST 랩퍼가 페더레이티드 시스템에 추가된 후에 BLAST 데이터 소스에 대해 쿼리를 실행할 수 있습니다.

관련 태스크:

- 97 페이지의『blastall 실행 파일 및 매트릭스 파일의 올바른 버전이 설치되었는지 확인』
- 97 페이지의『BLAST 디먼 구성』
- 101 페이지의『BLAST 디먼 시작』
- 102 페이지의『BLAST 랩퍼 등록』
- 103 페이지의『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』

- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』
- 33 페이지의『페더레이티드 시스템에 Documentum 추가』
- 75 페이지의『페더레이티드 시스템에 Excel 추가』
- 124 페이지의『페더레이티드 시스템에 XML 추가』

blastall 실행 파일 및 매트릭스 파일의 올바른 버전이 설치되었는지 확인

이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. BLAST 서버 머신에 최신 버전의 blastall 실행 파일과 BLOSUM62, BLOSUM80, PAM30 및 PAM70 매트릭스 파일을 설치했는지 확인하십시오. 설치하지 않았으면 2진 파일과 매트릭스 파일을 설치해야 합니다. 매트릭스 파일은 blastall 실행 파일과 동일한 디렉토리에 있어야 합니다.

절차:

blastall 실행 파일과 매트릭스 파일의 버전 레벨을 점검하려면 다음과 같이 하십시오.

1. 명령행에서 BLAST 검색을 실행하고 출력 파일에 있는 버전 번호를 보십시오.
2. 최신 버전의 blastall 실행 파일과 매트릭스 파일이 없으면 NCBI 웹 사이트 ftp://ftp.ncbi.nih.gov/blast/executables에서 파일을 다운로드하십시오.

이 태스크의 다음 태스크는 *BLAST 디먼 구성*입니다.

관련 태스크:

- 97 페이지의『BLAST 디먼 구성』

BLAST 디먼 구성

이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. BLAST 랩퍼는 DB2 Universal Database 페더레이티드 시스템의 TCP/IP를 통해 액세스 가능한 UNIX 기반 머신에서 BLAST 디먼을 실행할 것을 요구합니다. 디먼은 랩퍼 및 DB2 Universal Database와 별도로 실행되며 랩퍼에서 BLAST

작업 요청을 기다립니다. 디먼 실행 파일 db2blast_daemon은 BLAST 서버 머신의 어느 디렉토리에나 위치할 수 있습니다.

DB2 Universal Database 설치 중 디먼 실행 파일은 페더레이티드 서버가 설치되고 있는 머신에서 AIX의 경우 /usr/opt/db2_08_01/bin 디렉토리에 있으며 다른 Unix 플랫폼의 경우 /opt/IBM/db2/V8.1/bin 디렉토리에 있습니다. 사용자의 환경에서 BLAST가 다른 머신에서 실행되는 경우, 이 머신에서 사용자가 선택한 위치로 디먼을 복사해야 합니다.

BLAST 디먼은 다음과 같은 권한을 가지고 있어야 합니다.

- BLAST 검색을 실행할 수 있도록 blastall 2진 파일에 대한 실행 액세스 권한.
- 임시 파일을 쓸 수 있는 디렉토리에 대한 쓰기 액세스 권한.
- BLAST 검색을 실행할 수 있는 최소한 하나의 BLAST 가능 데이터 소스에 대한 읽기 액세스 권한. blastall 실행 파일은 formatdb 프로그램에 의해 생성된 데이터 파일과 BLAST 인덱스 파일 모두에 대한 읽기 액세스 권한을 가지고 있어야 합니다.

BLAST 디먼에는 구성 파일이 필요합니다. 이름이 BLAST_DAEMON.config인 샘플 디먼 구성 파일은 DB2PATH/samples/lifesci 디렉토리에 위치하며 여기서 DB2PATH는 DB2 Universal Database가 설치된 디렉토리입니다. BLAST_DAEMON.config가 파일의 디폴트 이름입니다.

디먼이 액세스할 수 있는 위치로 구성 파일을 복사하고 원하면 이 파일의 이름을 바꾸고 데이터 소스에 대해 작업하도록 편집하십시오. 디폴트로 blast_daemon은 이 디먼이 시작된 작업 디렉토리에서 구성 정보를 찾습니다.

절차:

디먼을 구성하려면 구성 파일에 다음의 옵션을 지정하십시오. 경로가 필요한 옵션의 경우, 상대 경로를 지정할 수 있습니다. 상대 경로는 디먼 프로세스가 시작된 디렉토리에 상대적입니다.

DAEMON_PORT

랩퍼가 제출한 BLAST 작업 요청을 디먼이 기다리는 네트워크 포트입니다.

MAX_PENDING_REQUESTS

한 번에 디먼에서 블로킹할 수 있는 최대 BLAST 작업 요청 수입니다. 이 수는 동시에 실행 중인 BLAST 작업의 수를 나타내지 않고 한 번에 블로킹할 수 있는 작업 요청의 수만을 나타냅니다. 이 수는 5보다 큰 수로 설정하는 것이 바람직합니다. BLAST 디먼은 동시에 실행할 수 있는 BLAST 작업의 수를 제한하지 않습니다.

DAEMON_LOGFILE_DIR

디먼이 로그 파일을 작성할 디렉토리입니다. 이 파일에는 BLAST 디먼에 의해 생성된 유용한 상태 및 오류 정보가 포함됩니다.

Q_SEQ_DIR_PATH

디먼에 의해 임시 쿼리 시퀀스 데이터 파일이 작성될 디렉토리입니다. 이 임시 파일은 BLAST 작업이 완료되면 정리됩니다.

BLAST_OUT_DIR_PATH

BLAST 출력 데이터를 저장하기 위해 디먼이 임시 파일을 작성할 디렉토리입니다. 이 파일에서 데이터를 읽고 네트워크 연결을 통해 랩퍼로 데이터를 다시 전달하며 이 때 디먼은 임시 파일을 정리합니다.

BLASTALL_PATH

디먼을 실행 중인 머신에 있는 BLAST 실행 파일의 완전한 이름입니다.

데이터베이스 스펙 항목

BLAST 가능 데이터 소스의 위치를 지정합니다. 디먼이 올바로 기능을 수행하려면 데이터 소스에 대한 별칭을 작성할 때 CREATE NICKNAME 문의 DATASOURCE 옵션에 구성 파일에 사용된 각 항목 이름을 지정해야 합니다. CREATE NICKNAME문에 대한 자세한 정보는 아래의 관련 태스크 절에서 ″BLAST 데이터 소스에 대한 별칭 등록″을 참조하십시오.

구성 파일에는 형식이 다음과 같은 최소한 하나의 데이터베이스 스펙 항목이 있어야 합니다.

```
entry_name = path to BLAST-able_data_source
```

예를 들어, GenBank BLAST 가능 데이터 소스를 지정하기 위해 디먼 구성 파일에 다음의 행을 추가합니다.

```
genbank=/dsk/1/nucl_data/genbank
```

데이터베이스 스펙 항목에 표시된 경로에는 다음의 파일이 포함되어야 합니다.

- 원래의 Fasta 형식 데이터
- 세 개의 인덱스 파일
  - 뉴클레오타이드 데이터 소스의 경우, 인덱스 파일의 확장자는 다음과 같습니다.
    - .nhr
    - .nin
    - .nsq
  - 아미노산 데이터 소스의 경우, 인덱스 파일의 확장자는 다음과 같습니다.
    - .phr
    - .pin
    - .psq

데이터베이스 스펙 항목은 원본 Fasta 형식 데이터가 포함된 파일의 파일 이름을 나타내야 합니다. 세 개의 인덱스 파일은 원본 Fasta 형식 데이터가 포함된 파일과 동일한 루트 파일 이름을 가지고 있어야 합니다.

구성 파일은 줄 바꾸기 문자로 끝나야 합니다.

예:

다음의 예는 GenBank 및 SWISS-PROT에 대한 BLAST 가능 데이터 소스 스펙 및 필수 옵션과 함께 샘플 구성 파일의 내용을 나타냅니다.

```
DAEMON_PORT=4007
MAX_PENDING_REQUESTS=10
DAEMON_LOGFILE_DIR=./
Q_SEQ_DIR_PATH=./
BLAST_OUT_DIR_PATH=./
BLASTALL_PATH=./blastall
genbank=/dsk/1/nucl_data/genbank
swissprot=/dsk/1/prot_data/swissprot
```

이 태스크의 다음 태스크는 *BLAST 디먼 시작*입니다.

관련 태스크:

- 97 페이지의 『blastall 실행 파일 및 매트릭스 파일의 올바른 버전이 설치되었는지 확인』
- 101 페이지의 『BLAST 디먼 시작』
- 105 페이지의 『BLAST 데이터 소스에 대한 별칭 등록』

BLAST 디먼 시작

이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. BLAST 데이터 소스에 액세스하려면 먼저 BLAST 디먼을 실행하고 있어야 합니다.

전제조건:

BLAST 디먼을 시작하기 전에 구성 파일의 DAEMON_LOGFILE_DIR, BLAST_OUT_DIR_PATH 및 Q_SEQ_DIR_PATH 항목 아래에 나열된 모든 경로에 대해 쓰기 액세스 권한을 가지고 있어야 합니다.

절차:

디먼 설치 디렉토리에 있으며 디먼 구성 파일의 이름을 변경하지 않았고 구성 파일이 디먼 실행 파일과 동일한 디렉토리에 있는 경우에 BLAST 디먼을 시작하려면 명령행에 다음의 명령을 입력하십시오.

```
db2blast_daemon
```

실행 파일은 BLAST 디먼이 실행되는 새로운 프로세스를 시작합니다.

디먼 구성 파일의 이름을 변경했거나 디먼 구성 파일이 위치한 디렉토리에 있지 않은 경우에 BLAST 디먼을 시작하려면 디먼 실행 파일이 새로운 이름 또는 위치를 지정하도록 랩퍼 디먼 명령에 -c 옵션을 사용해야 합니다.

예를 들어, 다음의 명령은 서브디렉토리 cfg의 BLAST_D.config라고 하는 파일에서 랩퍼 디먼이 구성 정보를 찾도록 합니다.

```
db2blast_daemon -c cfg/BLAST_D.config
```

이 태스크의 다음 태스크는 *BLAST 랩퍼 등록*입니다.

관련 태스크:

- 97 페이지의『BLAST 디먼 구성』
- 102 페이지의『BLAST 랩퍼 등록』

BLAST 랩퍼 등록

이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. 데이터 소스에 액세스하려면 랩퍼를 등록해야 합니다. 랩퍼는 페더레이티드 서버가 통신을 하고 데이터 소스에서 데이터를 검색하는 데 사용하는 메커니즘입니다. 랩퍼는 라이브러리 파일로 시스템에 설치됩니다.

절차:

BLAST 랩퍼를 등록하려면 CREATE WRAPPER문을 제출하십시오.

예를 들어, AIX에서 디폴트 라이브러리 파일 `libdb2lsblast.a`로부터 `my_blast`라고 하는 BLAST 랩퍼를 작성하려면 다음의 명령문을 제출하십시오.

```
CREATE WRAPPER my_blast LIBRARY 'libdb2lsblast.a'
OPTIONS(DB2_FENCED 'N');
```

지원되는 플랫폼별로 BLAST 랩퍼에 대한 디폴트 라이브러리 파일 이름의 표는 아래의 관련 태스크 절에서 ″DB2 Life Sciences Data Connect를 설치한 후″를 참조하십시오. CREATE WRAPPER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정*입니다.

관련 태스크:

- 17 페이지의『테이블 구조 파일 랩퍼 등록』
- 38 페이지의『Documentum 랩퍼 등록』

- 76 페이지의『Excel 랩퍼 등록』
- 101 페이지의『BLAST 디먼 시작』
- 103 페이지의『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 125 페이지의『XML 랩퍼 등록』

BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정

이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. BLAST 데이터 소스에 액세스할 때 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오. 이 변수는 페더레이티드 서버가 초기화 중에 랩퍼를 로드하는지 여부를 결정합니다.

절차:

DB2_DJ_COMM 환경 변수를 설정하려면 연관된 CREATE WRAPPER문에 지정한 랩퍼에 해당하는 랩퍼 라이브러리와 함께 `db2set` 명령을 제출하십시오.

예를 들어, 다음과 같습니다.

```
db2set DB2_DJ_COMM='libdb2lsblast.a'
```

등호(=)의 양쪽에 스페이스가 없게 하십시오.

데이터베이스 시작 중 랩퍼 라이브러리 로드와 연관된 오버헤드가 있습니다. 이 오버헤드를 방지하려면 액세스하려는 라이브러리만을 지정하십시오.

DB2_DJ_COMM 환경 변수에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

이 태스크의 다음 태스크는 *BLAST 데이터 소스에 대한 서버 등록*입니다.

관련 태스크:

- 18 페이지의『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 39 페이지의『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 102 페이지의『BLAST 랩퍼 등록』

- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』
- 126 페이지의『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

BLAST 데이터 소스에 대한 서버 등록

이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. 랩퍼를 등록한 후에 해당 서버를 등록해야 합니다.

절차:

페더레이티드 시스템에 BLAST 서버를 등록하려면 CREATE SERVER문을 사용하십시오.

사용자의 환경에서 BLAST 실행 파일 및 디먼이 설치된 각 머신에 대해 BLAST 실행 파일 및 디먼의 이 인스턴스를 사용하여 실행하려는 BLAST 검색의 각 유형마다 하나의 서버를 등록해야 합니다.

예를 들어, BLASTn 검색에 사용할 CREATE WRAPPER문을 사용하여 작성된 my_blast 랩퍼에 대해 blast_server1이라고 하는 서버를 등록하려면 다음의 명령문을 제출하십시오.

```
CREATE SERVER blast_server1
TYPE blastn
   VERSION 2.1.2
   WRAPPER my_blast
   OPTIONS (NODE 'big_rs.company.com', PORT '4007')
```

인수

TYPE 제공된 서버를 사용하여 수행되는 BLAST 검색의 유형을 판별합니다. 이 인수는 필수입니다. blastn, blastp, blastx, tblastn, tblastx 값 중 하나로 설정해야 합니다.

VERSION

사용하고 있는 서버의 버전을 지정합니다. 실행 중인 blastall의 버전으로 설정해야 합니다. 이 인수는 필수입니다.

WRAPPER

CREATE WRAPPER문을 사용하여 등록한 랩퍼의 이름을 지정합니다. 이 인수는 필수입니다.

옵션

NODE

BLAST 디먼 프로세스가 실행되고 있는 시스템의 호스트 이름을 지정합니다. 이 옵션은 필수입니다.

PORT

디먼이 BLAST 작업 요청을 기다리는 포트 번호를 지정합니다. 포트 번호는 디먼 구성 파일의 daemon_port 옵션에 지정된 번호와 동일해야 합니다. 디폴트는 4007입니다. 이 옵션은 선택적입니다.

CREATE SERVER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *BLAST 데이터 소스에 대한 별칭 등록*입니다.

관련 태스크:

- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』
- 103 페이지의『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』
- 127 페이지의『XML 데이터 소스에 대한 서버 등록』

BLAST 데이터 소스에 대한 별칭 등록

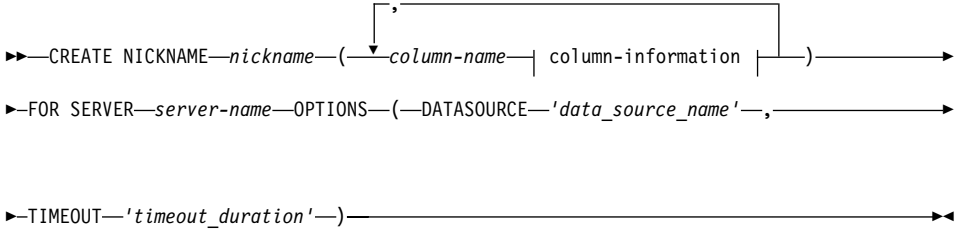
이 태스크는 *페더레이티드 시스템에 BLAST 추가*에 대한 기본 태스크의 일부입니다. 서버를 등록한 후에 해당 별칭을 등록해야 합니다. 별칭은 쿼리에서 BLAST 데이터 소스를 참조할 때 사용됩니다.

절차:

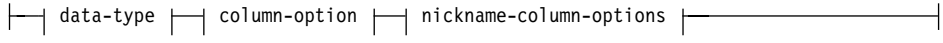
BLAST 별칭을 등록하려면 CREATE NICKNAME문을 사용하십시오. 각 BLAST 검색 유형은 별도의 서버에 의해 처리되므로 제공된 BLAST 가능 데이터 소스에 대해 실행하려는 각 BLAST 검색 유형에 대해 별도의 별칭을 정의해야 합니다.

별칭은 데이터 소스의 정의 행 부분에 대한 컬럼 정보를 지정합니다. 다른 모든 컬럼은 고정 컬럼입니다. 정의 행 구문 분석에 대한 자세한 정보는 108 페이지의 『정의 행 구문 분석』을 참조하십시오. 고정 컬럼에 대한 자세한 정보는 109 페이지의 『고정 컬럼』을 참조하십시오.

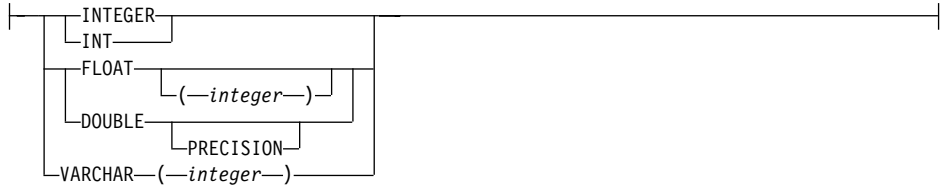
BLAST에 대한 CREATE NICKNAME문의 구문은 다음과 같습니다.



column-information:

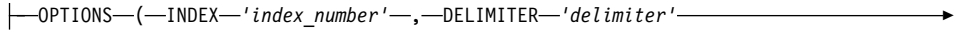


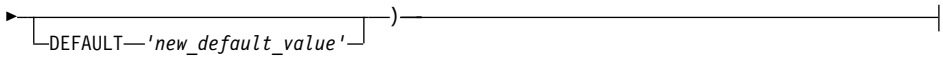
data-type:



column-option:

nickname-column-options:





CREATE NICKNAME문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

별칭 컬럼 옵션

별칭 컬럼 옵션 값은 작은따옴표로 묶어야 합니다.

INDEX

정의 행 컬럼 그룹에서 이 옵션이 나타나는 컬럼의 서수. 이 옵션은 필수입니다. 정의 행 구문 분석에 대한 자세한 정보는 108 페이지의『정의 행 구문 분석』을 참조하십시오.

DELIMITER

이 옵션이 나타나는 컬럼에 대한 정의 행 정보의 끝점을 판별하는 데 사용해야 하는 분리문자. 이 옵션의 값에 둘 이상의 문자가 나타나는 경우, 이들 문자 중 처음에 나타나는 한 문자가 이 필드의 정보 끝을 알립니다. 디폴트는 행의 끝(eol)입니다. 컬럼에 나머지 정의 행을 포함시키려는 경우 지정되는 마지막 컬럼을 제외하고 이 옵션은 필수적입니다. 정의 행 구문 분석에 대한 자세한 정보는 108 페이지의『정의 행 구문 분석』을 참조하십시오.

DEFAULT

다음의 입력 고정 컬럼에 대한 새로운 디폴트값을 지정합니다.

- E_value
- QueryStrands
- GapAlign
- NMisMatchPenalty
- NMatchReward
- Matrix
- FilterSequence
- NumberOfAlignments
- GapCost

- ExtendedGapCost
- WordSize
- ThresholdEx

이 새로운 값이 미리 설정된 디폴트값을 겹쳐씁니다. 새로운 디폴트값은 제공된 컬럼에 대해 표시된 값과 동일한 유형이어야 합니다. 입력 고정 컬럼에 대한 자세한 정보는 109 페이지의 『입력 고정 컬럼』을 참조하십시오. 이 옵션은 선택적입니다.

별칭 옵션

별칭 옵션 값은 작은따옴표 안에 있어야 합니다.

DATASOURCE

BLAST 검색이 실행될 대상 데이터 소스의 이름. 여기에 사용되는 정확한 문자열은 BLAST 디먼의 구성 파일에 제공되어야 합니다. 구성 파일에 대한 자세한 정보는 아래의 관련 태스크 절에서 ″BLAST 디먼 구성″을 참조하십시오. 이 옵션은 필수입니다.

TIMEOUT

BLAST 랩퍼가 디먼으로부터 결과를 기다릴 최대 시간(분). 디폴트는 60입니다. 이 옵션은 선택적입니다.

정의 행 구문 분석

defline이라고도 하는 정의 행은 BLAST 가능 데이터 소스의 각 시퀀스에 대한 키와 유사하며 각 BLAST 히트의 일부분으로 리턴됩니다.

결과 테이블에 정의 행 정보를 포함시키고자 하는 경우, CREATE NICKNAME 문에 정의 행 컬럼을 지정해야 합니다. 각 컬럼 스펙은 INDEX 옵션을 지정해야 합니다. 컬럼에 나머지 정의 행을 포함시키려는 경우 지정되는 마지막 컬럼을 제외하고 각 컬럼에 대해 DELIMITER 옵션을 지정해야 합니다.

INDEX 및 DELIMITER 옵션에 대한 자세한 정보는 107 페이지의 『별칭 컬럼 옵션』을 참조하십시오.

정의 행 필드의 유형은 integer, float, double 또는 varchar 중 하나이어야 합니다.

주: BLAST 히트의 액세스 수 필드에서 데이터가 발견된 경우, 이 BLAST 히트의 정의 필드에 있는 데이터 앞에 이 데이터가 삽입됩니다. 정의 필드 데이터 앞에 액세스 수 데이터가 포함된 결과 정의 행은 랩퍼에 의해 구문 분석됩니다.

CREATE NICKNAME문에 정의 행 컬럼을 지정하는 방법을 나타내는 예는 112 페이지의 『CREATE NICKNAME의 예』를 참조하십시오.

고정 컬럼

CREATE NICKNAME문은 자동으로 고정 컬럼을 작성합니다. 고정 컬럼은 CREATE NICKNAME문에는 나타나지 않지만 별칭 정의의 일부이며 SQL 쿼리에 참조할 수 있습니다. 고정 컬럼의 유형은 입력과 출력의 두 가지입니다.

입력 고정 컬럼

입력 고정 컬럼은 SQL 쿼리에서 매개변수 전달 술어로 사용됩니다. BLAST에 표준 BLAST 스위치를 전달합니다. 그러면 BLAST는 이러한 스위치를 사용하여 지정된 데이터 소스에 대해 실행됩니다. 입력 고정 컬럼은 쿼리 선택 목록에도 참조할 수 있으며 결과 테이블의 일부로 리턴될 수 있습니다. 입력 고정 컬럼은 표 22에 나열되어 있습니다.

표 22. 입력 고정 컬럼

이름	데이터 유형	설명
BlastSeq	varchar(32000)	쿼리 시퀀스를 BLAST 랩퍼에 전달합니다.
E_Value	double	입력 및 출력 매개변수. 입력 매개변수로서 이 컬럼은 blastall로부터 리턴되어야 하는 예상 값의 상한을 BLAST 랩퍼에 나타냅니다.
QueryStrands	integer	BLASTn 검색을 수행할 때 비교해야 하는 요소를 지정합니다. 값 1은 최상위 요소를 사용해야 함을 나타내고 2는 최하위 요소를 나타내며 3은 이 두 요소를 비교해야 함을 나타냅니다.
GapAlign	char(1)	BLAST 출력에서 갭이 있는 정렬이 허용되는지의 여부를 랩퍼에 나타냅니다.
Matrix	varchar(50)	아미노산 쌍들 사이의 유사도를 판별하기 위해 blastall이 사용하는 대체 매트릭스를 판별합니다. 아미노산을 아미노산과 비교하는 BLAST 검색 유형만이 이 술어를 사용합니다.

표 22. 입력 고정 컬럼 (계속)

이름	데이터 유형	설명
NMisMatchPenalty	integer	상동 부분의 뉴클레오타이드 쌍 중 하나가 일치하지 않으면 blastall이 정렬 점수에서 감하는 값을 지정합니다. 뉴클레오타이드를 뉴클레오타이드와 비교하는 BLAST 검색 유형만이 이 술어를 사용합니다.
NMatchReward	integer	일치하지 않는 상동 부분의 각 뉴클레오타이드 쌍에 대한 정렬 점수에 blastall이 더하는 값을 지정합니다. 뉴클레오타이드를 뉴클레오타이드와 비교하는 BLAST 검색 유형만이 이 술어를 사용합니다.
FilterSequence	char(1)	쿼리 시퀀스에서 생물학적으로 흥미가 없는 세그먼트를 제거하기 위해 필터를 수행할 것인지의 여부를 blastall에 나타냅니다. 검색 유형이 BLASTn인 경우, 사용되는 필터는 DUST입니다. 그렇지 않으면 필터는 SEG에 의해 수행됩니다.
NumberOfAlignments	integer	BLAST 출력에 포함시킬 HSP 정렬 수를 지정합니다.
GapCost	integer	정렬의 길이를 연장하기 위해 쿼리 시퀀스나 히트 시퀀스 중 하나에 갭을 삽입해야 하는 경우 blastall이 정렬 점수에서 감하는 값을 지정합니다.
ExtendedGapCost	integer	정렬 길이를 연장하기 위해 쿼리 시퀀스나 히트 시퀀스에 이미 삽입된 갭을 뉴클레오타이드 또는 아미노산 하나씩으로 확장해야 하는 경우 blastall이 정렬 점수에서 감하는 값을 지정합니다.
WordSize	integer	blastall이 데이터베이스에서 처음에 검색하는 초기 히트의 길이를 blastall에 나타냅니다.
ThresholdEx	integer	BLAST가 더 이상 히트 확장을 시도하지 않는 하한 점수 임계값을 나타냅니다.

각 입력 고정 컬럼에 대해 지원되는 BLAST 검색 유형 및 스위치는 표 23에 나열되어 있습니다.

표 23. 입력 고정 컬럼이 지원하는 BLAST 검색 유형 및 스위치

이름	BLAST 검색 유형	BLAST 스위치	필수 여부	디폴트
BlastSeq	n, p, x, tn, tx	-l	Y	N/A
E_Value	n, p, x, tn, tx	-e	N	10
QueryStrands	n	S	N	3

표 23. 입력 고정 컬럼이 지원하는 BLAST 검색 유형 및 스위치 (계속)

이름	BLAST 검색 유형	BLAST 스위치	필수 여부	디폴트
GapAlign	n, p, x, tn, tx	-g	N	T
Matrix	p, x, tn, tx	-n	N	BLOSUM62
NMisMatchPenalty	n	-q	N	-3
NMatchReward	n	-r	N	1
FilterSequence	n, p, x, tn, tx	-F	N	T
NumberOfAlignments	n, p, x, tn, tx	-b	N	250
GapCost	n, p, x, tn, tx	-G	N	11
ExtendedGapCost	n, p, x, tn, tx	-E	N	1
WordSize(Blastn의 경우, 7보다 작은 값은 유효하지 않음)	n, p, x, tn, tx	-W	N	11 -BLASTn 3 -BLASTp
ThresholdEx	n, p, x, tn, tx	-f	N	0

출력 고정 컬럼

출력 고정 컬럼은 쿼리 결과 테이블에 리턴되며 술어로 사용할 수 있습니다. 출력 고정 컬럼은 표 24에 나열되어 있습니다.

표 24. 출력 고정 컬럼

이름	데이터 유형	설명
Score	double	BLAST 결과에 보고되는 계산된 HSP 점수.
E_value	double	입력 및 출력 매개변수. 출력 매개변수로서 이 컬럼은 BLAST 결과에 보고되는 계산된 HSP 점수를 제공합니다.
Length	integer	BLAST 결과에 보고되는 히트 시퀀스의 길이.
HSP_Info	varchar(100)	BLAST에 의해 보고되는 제공된 HSP에 대한 정보 문자열. 이 문자열에는 쿼리 시퀀스와 히트 시퀀스 사이에 일치한 뉴클레오타이드 또는 아미노산 수에 대한 정보가 들어 있습니다.
HSP_Q_Start	integer	쿼리 시퀀스에서 첫 번째 상동 뉴클레오타이드 또는 아미노산의 숫자 자리.
HSP_Q_End	integer	쿼리 시퀀스에서 마지막 상동 뉴클레오타이드 또는 아미노산의 숫자 자리.
HSP_Q_Seq	varchar(32000)	HSP_Q_Start에서 시작하고 HSP_Q_End에서 종료하는 쿼리 시퀀스의 세그먼트.

표 24. 출력 고정 컬럼 (계속)

이름	데이터 유형	설명
HSP_H_Start	integer	히트 시퀀스에서 첫 번째 상동 뉴클레오타이드 또는 아미노산의 숫자 자리.
HSP_H_End	integer	히트 시퀀스에서 마지막 상동 뉴클레오타이드 또는 아미노산의 숫자 자리.
HSP_H_Seq	varchar(32000)	HSP_H_Start에서 시작하고 HSP_H_End에서 종료하는 히트 시퀀스의 세그먼트.
HSP_Midline	varchar(32000)	쿼리 및 히트 시퀀스의 상동 부분에서 각 위치에 있는 아미노산 또는 뉴클레오타이드 사이의 일치 정도를 나타내는 BLAST에 의한 문자열 출력.

CREATE NICKNAME의 예

다음의 CREATE NICKNAME문은 별칭 genbank를 정의합니다.

BLAST 결과의 정의 필드에 다음의 정보가 포함된 것으로 가정합니다.

```
>276342 15:8924 PMON5426
```

항목 설명:

276342

BLAST 결과의 액세스 필드.

15:8924 PMON5426

유기체 번호와 그 뒤의 실험 번호 및 고유 ID가 포함된 BLAST 결과의 정의 필드.

이 정보를 사용하면 다음의 별칭이 작성됩니다.

```
CREATE NICKNAME genbank (
     acc_num integer OPTIONS(INDEX '1', DELIMITER ' '),
     org_num integer OPTIONS(INDEX '2', DELIMITER ':'),
     exp_num  integer OPTIONS(INDEX '3', DELIMITER ' '),
     u_id varchar(10) OPTIONS(INDEX '4'))
     FOR SERVER blast_server1
      OPTIONS(DATASOURCE 'genbank', TIMEOUT '300');
```

컬럼 acc_num에는 276342가 포함되고 컬럼 org_num에는 15, 컬럼 exp_num에는 8924, 컬럼 u_id에는 PMON5426이 포함됩니다.

CREATE NICKNAME문을 제출한 후에 별칭 genbank를 사용하여 페더레이티드 시스템을 쿼리할 수 있습니다. genbank 별칭을 페더레이티드 시스템의 다른 별칭 및 테이블과 조인할 수도 있습니다.

이 태스크 다음에는 더 이상의 태스크가 없습니다.

관련 태스크:

- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 97 페이지의『BLAST 디먼 구성』
- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』
- 145 페이지의 제 8 장『비용산정 별칭 옵션 지정』

BLAST SQL 쿼리 구성

BLAST 데이터 소스에 대한 SQL에는 blastall 실행 파일에 표준 BLAST 스위치를 전달하는 데 사용되는 특수 입력 술어만이 포함되어야 합니다.

제한사항:

BLAST 랩퍼에 전달되는 모든 쿼리가 유효하려면 최소한 `BlastSeq` 입력 술어가 포함되어야 합니다. 다른 모든 술어는 선택적입니다.

절차:

BLAST 쿼리를 구성하려면 SQL문의 WHERE절에 입력 술어를 사용하십시오.

다음의 예는 세 개의 입력 술어 `BlastSeq`, `GapCost` 및 `NMisMatchPenalty`를 나타냅니다.

```
Select * from blast b where
BlastSeq = 'GTCCAGCC...' AND
GapCost = -10 AND
NMisMatchPenalty = -4;
```

각 입력 술어에 대해 지원되는 데이터 유형, 설명, BLAST 스위치 및 검색 유형의 목록은 아래의 관련 태스크 절에서 'BLAST 데이터 소스에 대한 별칭 등록'을 참조하십시오.

관련 태스크:

• 114 페이지의『샘플 BLAST 쿼리』

샘플 BLAST 쿼리

BLAST 데이터 소스에 대해 쿼리를 구성하는 방법을 설명하기 위해 아래에 몇 가지 샘플 BLAST 쿼리가 제공됩니다.

절차:

쿼리를 실행하려면 아래의 예를 지침으로 사용하십시오.

이러한 쿼리에서 각 별칭에 대해 사용되는 이름은 BLAST 검색 및 데이터 소스의 유형을 나타냅니다. 이와 같이 하면 등록 명령문을 각 샘플 쿼리와 함께 나열할 필요가 없습니다. 또한 일부 쿼리는 다른 가설 데이터 소스를 활용하므로 이들 예는 다른 데이터 소스 조인시 랩퍼의 동작을 보여줄 수 있습니다.

쿼리 1

```
select *
from blastn_genbank
where BlastSeq =
'caacccctccagccgagttgtcaatggcgaggaagctgttccccac';
```

이 SQL문이 실행되면 랩퍼는 표시된 시퀀스를 사용하여 GenBank의 BLASTn 검색을 수행합니다. 랩퍼는 입력 매개변수 컬럼 및 BLAST 결과 컬럼을 비롯하여 사용 가능한 모든 컬럼을 리턴합니다.

쿼리 2

```
select *
from blastn_genbank
where BlastSeq =
'caacccctccagccgagttgtcaatggcgaggaagctgttccccac'
and GapCost = 8 and NmisMatchPenalty = -4;
```

이 SQL문이 실행되면 랩퍼는 표시된 시퀀스를 사용하여 GenBank의 BLASTn 검색을 수행합니다. 또한 랩퍼는 두 개의 표시된 매개변수를 디먼에 전달하며 이 매개변수는 blastall 명령행에 전달됩니다. 랩퍼는 입력 매개변수 컬럼 및 BLAST 결과 컬럼을 비롯하여 사용 가능한 모든 컬럼을 리턴합니다.

쿼리 3

```
select blp.*
from blastp_swissprot blp, protein_db prdb
where prdb.keyword = 'malic enzyme'
and blp.BlastSeq = prdb.sequence;
```

이 SQL문이 실행되면 랩퍼는 가설 단백질 배열 데이터베이스로부터 리턴된 배열 수에 따라 SWISS-PROT에 대해 0회 이상의 BLASTp 검색을 수행합니다. 이 명령문은 DB2에 의해 두 개의 개별 쿼리로 나뉘고 가설 단백질 데이터베이스로부터 리턴되는 각 행에 대해 한 번의 BLASTp 검색이 실행됩니다. 랩퍼는 입력 매개변수 컬럼 및 BLAST 결과 컬럼을 비롯하여 사용 가능한 모든 컬럼을 리턴합니다.

쿼리 4

```
select Score, E_Value, HSP_Info, HSP_Q_Seq, HSP_H_Seq, HSP_Midline
from blastx_swissprot
where BlastSeq = 'gagttgtcaatggcgagg'
and GapCost = 8;
```

이 SQL문이 실행되면 랩퍼는 표시된 시퀀스를 사용하여 SWISS-PROT의 BLASTx 검색을 수행합니다. 이 경우, blastall은 여섯 개의 모든 읽기 프레임에 있는 입력 시퀀스를 변환하고 새로 작성된 여섯 개의 단백질 배열을 각각 사용하여 상동 검색을 수행합니다. 결과의 HSP에는 뉴클레오타이드 대 뉴클레오타이드 정렬이 아닌 아미노산 대 아미노산 정렬이 포함됩니다. 제공된 매개변수는 디먼으로 전달된 다음 명령행을 통해 blastall로 전달됩니다. 랩퍼는 쿼리에서 특별히 요청된 컬럼만을 다시 실행합니다.

쿼리 5

```
select tblx.Score, tblx.E_Value, tblx.HSP_Info tblx.HSP_Q_Seq,
HSP_H_Seq, HSP_Midline
from tblastx_genbank tblx, gen_exp_database gedb
```

```
where tblx.BlastSeq = gedb.sequence
and gedb.organism = 'interesting organism'
and GapCost = 8
and FilterSequence = 'F';
```

이 SQL문이 실행되면 랩퍼는 가설 유전자 발현 데이터베이스로부터 리턴된 배열 수에 따라 GenBank에 대해 0회 이상의 tBLASTx 검색을 수행합니다. 이 명령문은 DB2에 의해 두 개의 개별 쿼리로 나뉘고 가설 유전자 발현 데이터베이스로부터 리턴되는 각 행에 대해 한 번의 tBLASTx 검색이 실행됩니다. 이 경우, blastall은 여섯 개의 모든 읽기 프레임에 있는 GenBank의 모든 시퀀스와 입력 시퀀스를 변환하고 새로 작성된 여섯 개의 각 쿼리 단백질 배열과 새로 작성된 데이터베이스 단백질 배열을 모두 사용하여 상동 검색을 수행합니다. 결과의 HSP에는 뉴클레오타이드 대 뉴클레오타이드 정렬이 아닌 아미노산 대 아미노산 정렬이 포함됩니다. 제공된 매개변수는 디먼으로 전달된 다음 명령행을 통해 blastall로 전달됩니다. 랩퍼는 쿼리에서 특별히 요청된 컬럼만을 다시 실행합니다.

관련 태스크:

- 59 페이지의『Documentum 데이터 소스에 대한 쿼리 실행』
- 79 페이지의『Excel 데이터 소스에 대한 쿼리 실행』
- 136 페이지의『XML 데이터 소스에 대한 쿼리 실행』

BLAST 랩퍼에 대한 최적화 추가 정보

동일한 서버에서 랩퍼와 디먼을 모두 실행하면 잠재적인 네트워크 통신 병목 현상을 제거할 수 있습니다.

관련 참조:

- 27 페이지의『테이블 구조 파일 랩퍼에 대한 최적화 추가 정보 및 고려사항』

BLAST 랩퍼에 대한 메시지

이 절에서는 BLAST의 랩퍼에 대해 작업할 때 발생할 수 있는 메시지를 나열하고 설명합니다. 메시지에 대한 자세한 정보는 *DB2 메시지 참조서*를 참조하십시오.

표 25. BLAST의 랩퍼에 의해 발행되는 메시지

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 ″sqlno_crule_save_plans [100]:rc (-2144272209) 비어 있는 플랜 목록이 발견되었습니다.″)	DB2에 제출된 SQL 쿼리를 랩퍼가 처리할 수 없습니다. 구문을 정정하고 다시 제출하십시오.
SQL1816N	랩퍼 ″BLAST_WRAPPER″는 페더레이티드 데이터베이스에 정의하려고 시도 중인 데이터 소스의 ″type″(″<server type>″ ″″)에 액세스하는 데 사용할 수 없습니다.	CREATE SERVER문에 유효하지 않은 TYPE이 사용되었습니다. 유형은 지원되는 BLAST 유형 중 하나이어야 합니다.
SQL1817N	CREATE SERVER문이 페더레이티드 데이터베이스에 정의하려는 데이터 소스의 ″버전″을 식별하지 않습니다.	CREATE SERVER문이 버전을 지정하지 않았습니다.
SQL1822N	데이터 소스 ″Blast 랩퍼″에서 예상치 못한 오류 코드 ″지정되지 않은 오류″를 받았습니다. 연관된 텍스트와 토큰은 ″디먼에 연결할 수 없습니다.″입니다.	blast 랩퍼를 디먼에 연결할 수 없습니다. 디먼이 실행되고 있지 않을 수 있습니다. 잘못 구성되었을 가능성이 있습니다. 실행되고 있는 머신에 도달하지 못할 수 있습니다.
SQL1822N	데이터 소스 ″Blast 랩퍼″에서 예상치 못한 오류 코드 ″지정되지 않은 오류″를 받았습니다. 연관된 텍스트와 토큰은 ″Blast 디먼 시간종료가 만기되었습니다.″입니다.	CREATE NICKNAME문에 지정된 시간종료 값이 경과하기 전에 디먼으로부터 결과를 받지 못했습니다. 시간종료 값을 늘리거나 디먼에 문제가 있는지 확인하십시오.
SQL1822N	데이터 소스 ″Blast 랩퍼″에서 예상치 못한 오류 코드 ″지정되지 않은 오류″를 받았습니다. 연관된 텍스트와 토큰은 ″Blast 디먼이 실패했습니다.″입니다.	디먼이 통신을 중지했거나 리턴된 결과가 올바로 형식화되지 않았습니다.
SQL1822N	데이터 소스 ″Blast 랩퍼″에서 예상치 못한 오류 코드 ″지정되지 않은 오류″를 받았습니다. 연관된 텍스트와 토큰은 ″blast 디먼으로부터의 알 수 없는 오류″입니다.	blast 랩퍼가 인식하지 못한 디먼으로부터 오류 코드를 받았습니다. 디먼 버전이 랩퍼 버전과 호환되지 않을 수 있습니다.

표 25. BLAST의 랩퍼에 의해 발행되는 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 ″Blast 랩퍼″에서 예상치 못한 오류 코드 ″지정되지 않은 오류″를 받았습니다. 연관된 텍스트와 토큰은 ″컬럼 이름 바꾸기가 허용되지 않습니다.″입니다.	컬럼 중 하나의 이름을 바꾸려고 시도할 때 ALTER NICKNAME문이 발행되었습니다. 컬럼 이름 바꾸기는 허용되지 않습니다.
SQL1822N	데이터 소스 ″Blast 랩퍼″에서 예상치 못한 오류 코드 ″지정되지 않은 오류″를 받았습니다. 연관된 텍스트와 토큰은 ″XML 구문 분석기 오류″입니다.	Xerces 구문 분석기가 유효하지 않은 상태에 있거나 예외가 발생했습니다.
SQL1823N	″<server name>″ 서버로부터의 ″<data type name>″ 데이터 유형에 대해 데이터 유형 맵핑이 존재하지 않습니다.	지정된 데이터 유형은 이 컬럼이 지원하지 않습니다.
SQL1881N	″DEFAULT″는 ″<column-name>″에 대해 유효한 ″COLUMN″ 옵션이 아닙니다.	DEFAULT 옵션을 지원하지 않는 컬럼에 이 옵션이 사용되었습니다. 출력 전용 컬럼과 정의 행 컬럼에는 디폴트값이 없습니다.
SQL1882N	″COLUMN″ 옵션 ″DEFAULT″를 ″<column-name>″에 대해 ″<option-value>″(으)로 설정할 수 없습니다.	DEFAULT 옵션에 대해 지정된 값이 컬럼에 대해 호환되지 않는 유형이거나 올바로 형식화되지 않았습니다.

관련 참조:

- 27 페이지의『테이블 구조 파일 랩퍼에 대한 메시지』
- 67 페이지의『Documentum 랩퍼에 대한 메시지』
- 84 페이지의『Excel 랩퍼에 대한 메시지』
- 138 페이지의『XML 랩퍼에 대한 메시지』

제 7 장 데이터 소스로서의 XML

이 장에서는 XML의 정의와 사용자의 페더레이티드 시스템에 XML 데이터 소스를 추가하는 방법을 설명하고, XML 랩퍼와 연관된 오류 메시지를 나열합니다.

XML의 정의

XML(Extensible Markup Language)은 구조화된 문서 및 데이터의 일반 형식입니다. XML 파일의 파일 확장자는 xml입니다. HTML과 마찬가지로 XML은 문서에서 데이터를 구조화할 때 태그('<'와 '>'로 묶인 단어)를 활용합니다. 120 페이지의 그림 9는 샘플 XML 문서입니다.

```
<doc>
   <customer id='123'>
      <name>...</name>
      <address>...</address>
       ...
      <order>
         <amount>...</amount>
            <date>...</date>
         <item quant='12'>
            <name>...</name>
         </item>
         <item quant='4'>...</item>
          ...
      </order>
      <order>...</order>
       ...
      <payment>
         <number>...</number>
         <date>...</date>
      </payment>
      <payment>...</payment>
       ...
   </customer>
   <customer id='124'>...</customer>
</doc)
```

그림 9. 샘플 XML 문서

XML 랩퍼는 SQL을 사용할 수 있게 하여 파일에 저장된 외부 XML 문서를 쿼리할 수 있도록 합니다. 121 페이지의 그림 10은 페더레이티드 시스템에 대한 XML의 작업 방식을 나타냅니다.

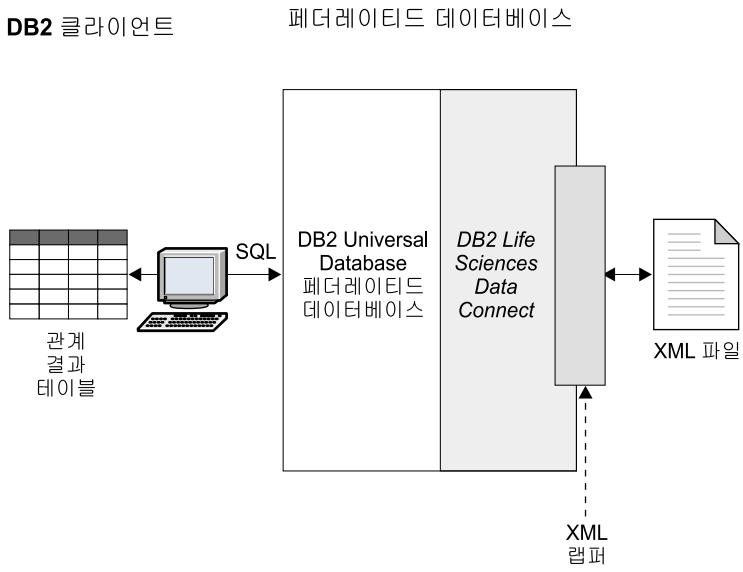


그림 10. XML 랩퍼의 작업 방식

XML 랩퍼를 사용하면 외부 데이터 소스에서 별칭 세트로 구성된 관계 스키마로 XML 데이터를 맵핑할 수 있습니다. XML 문서의 구조는 중첩 및 반복 요소를 외부 키를 사용하여 별도의 테이블로 모델화하는 관계 스키마와 논리적으로 동등합니다.

XML 문서에 해당하는 별칭은 하위 별칭이 상위 별칭에 해당하는 요소에 중첩된 요소를 모델화하는 트리로 구성됩니다.

중첩된 요소를 별도의 별칭으로 모델화해야 할 때 기본적으로 두 가지 경우가 있습니다.

- 반복 요소
- 구별되는 정체 및 풍부한 구조가 있는 요소

하위 및 상위 별칭은 랩퍼에 의해 생성된 기본/외부 키에 의해 연결됩니다.

XPath 표현식은 XML 문서를 별칭 세트로 구성된 관계 스키마에 맵핑하는 데 사용됩니다. XPath는 XML 파일의 일부분 예를 들어, XML 문서 트리 내의 노드 및 속성 그룹을 식별하기 위한 주소 지정 메커니즘입니다. 기본 XPath 구문은 파일 시스템 주소 지정과 비슷합니다.

각 별칭은 개별 쌍(tuple)을 나타내는 XML 요소를 식별하는 XPath 표현식과 각 요소로부터 컬럼 값을 추출하는 방법을 지정하는 XPath 표현식 세트에 의해 정의됩니다.

예:

다음의 예는 120 페이지의 그림 9에 있는 샘플 XML 문서를 별칭 세트에 맵핑하는 방법, 기본 및 외부 키를 사용하여 상위 및 하위 관계를 모델화하는 방법, 문서의 각 요소 내에서 개별 쌍(tuple)과 컬럼을 정의하기 위해 XPath 표현식을 사용하는 방법 및 쿼리가 페더레이티드 시스템에 등록된 후에 XML 문서에 대해 쿼리를 실행할 수 있는 방법을 보여줍니다.

샘플 XML 문서에는 각각 여러 `order` 및 `payment` 요소를 묶은 `customer` 요소 세트가 들어 있습니다.

`order` 요소는 여러 `item` 요소를 묶습니다.

요소 사이의 관계는 그림 11에 표시됩니다.

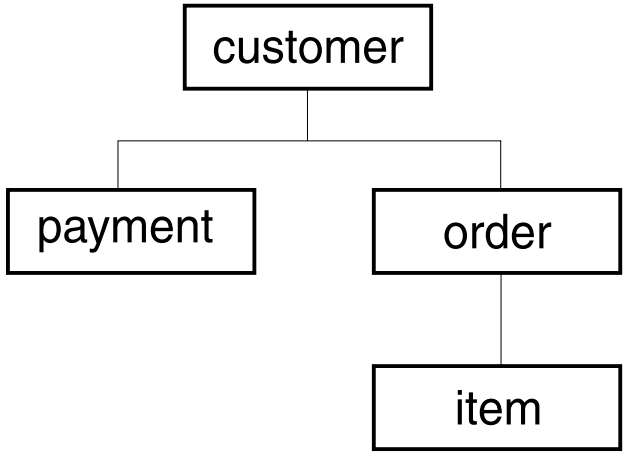


그림 11. 샘플 XML 문서의 트리 구조

이 구조에서 XML 문서는 CREATE NICKNAME문을 사용하여 네 개의 별칭을 사용하는 관계 스키마에 맵핑할 수 있습니다.

- customers

- orders
- payments
- items

별칭 사이의 관계는 차례대로 특수 기본 및 외부 키 별칭 컬럼 옵션을 사용하여 상위 또는 하위 별칭으로 각 별칭을 지정하여 정의됩니다. 각 상위 별칭은 기본 키 컬럼 옵션을 사용하여 지정된 특수 컬럼을 가지고 있어야 합니다. 외부 키 컬럼 옵션을 사용하여 상위 별칭의 기본 키 컬럼을 참조하는 특수 컬럼을 보유하여 상위의 하위를 정의합니다. 지정된 기본 및 외부 별칭 컬럼에는 랩퍼에 의해 생성된 키가 포함되므로 XML 문서의 데이터와 일치하지 않습니다. 상위가 없는 루트를 제외하고는 별칭에 정확히 하나의 상위가 있어야 하지만 여러 하위가 있을 수 있습니다.

샘플 XML 문서의 경우 `customers` 별칭에는 하나의 기본 키가 정의되며 `orders`, `payments` 및 `items` 별칭에는 각각 상위 별칭을 가리키는 외부 키가 정의됩니다. `orders` 및 `payments` 별칭에는 `customers`를 가리키는 외부 키가 있고 `items` 별칭에는 `orders`를 가리키는 외부 키가 있습니다.

개별 쌍(tuple)을 나타내는 XML 요소를 식별하기 위해 하나의 XPath 표현식이 작성됩니다. 이 예에서 모든 customer 요소는 `//customer` XPath 표현식을 사용하여 참조할 수 있으며 모든 order 요소는 `.//order` XPath 표현식을 사용하여 참조할 수 있습니다.

각 요소에서 컬럼 값을 추출하는 방법을 지정하기 위해 XPath 표현식 세트가 작성됩니다. 이 예에서 customer 요소의 `id` 속성은 현재 별칭에 정의된 컬럼이며 `./@id` XPath 표현식을 사용하여 참조할 수 있습니다. customer 요소의 이름 요소는 `.//name` XPath 표현식을 사용하여 참조할 수 있으며 customer 요소의 주소 요소는 `.//address/@street` XPath 표현식을 사용하여 참조할 수 있습니다.

XML 문서가 CREATE NICKNAME문을 사용하여 별칭 세트에 맵핑되었으면 문서의 각 요소 내에서 개별 쌍(tuple) 및 컬럼을 정의하는 XPath 표현식으로 기본 및 외부 키를 사용하여 각 별칭이 상위 또는 하위로 정의되며 XML 문서에 대해 SQL 쿼리를 실행할 수 있습니다.

별칭을 작성하는 방법과 CREATE NICKNAME문의 구문에 대한 자세한 정보는 아래의 관련 태스크 절에서 'XML 데이터 소스에 대한 별칭 등록'을 참조하십시오.

관련 개념:

- 13 페이지의『테이블 구조 파일의 정의』
- 31 페이지의『Documentum의 정의』
- 73 페이지의『Excel의 정의』
- 91 페이지의『BLAST의 정의』

관련 태스크:

- 124 페이지의『페더레이티드 시스템에 XML 추가』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』

페더레이티드 시스템에 XML 추가

절차:

페더레이티드 서버에 XML 데이터 소스를 추가하려면 다음과 같이 하십시오.

1. CREATE WRAPPER문을 사용하여 랩퍼를 등록하십시오.
2. 선택적: 쿼리 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오.
3. CREATE SERVER문을 사용하여 서버를 등록하십시오.
4. CREATE NICKNAME문을 사용하여 별칭을 등록하십시오.
5. 루트가 아닌 별칭에 대한 뷰를 작성하십시오.

이 명령문은 DB2 명령행 처리기에서 실행할 수 있습니다. XML 랩퍼가 페더레이티드 시스템에 추가된 후에 XML 데이터 소스에 대해 쿼리를 실행할 수 있습니다.

관련 태스크:

- 125 페이지의『XML 랩퍼 등록』
- 126 페이지의『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

- 127 페이지의『XML 데이터 소스에 대한 서버 등록』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』
- 134 페이지의『루트가 아닌 별칭에 대한 페더레이티드 뷰 작성(XML 랩퍼)』
- 16 페이지의『페더레이티드 시스템에 테이블 구조 파일 추가』
- 33 페이지의『페더레이티드 시스템에 Documentum 추가』
- 75 페이지의『페더레이티드 시스템에 Excel 추가』
- 96 페이지의『페더레이티드 시스템에 BLAST 추가』

XML 랩퍼 등록

이 태스크는 *페더레이티드 시스템에 XML 추가*에 대한 기본 태스크의 일부입니다. 데이터 소스에 액세스하려면 랩퍼를 등록해야 합니다. 랩퍼는 페더레이티드 서버가 통신을 하고 데이터 소스에서 데이터를 검색하는 데 사용하는 메커니즘입니다. 랩퍼는 라이브러리 파일로 시스템에 설치됩니다.

절차:

XML 랩퍼를 등록하려면 CREATE WRAPPER문을 제출하십시오.

예를 들어, AIX에서 디폴트 라이브러리 파일 `libdb2lsxml.a`로부터 `my_xml`이라고 하는 XML 랩퍼를 작성하려면 다음의 명령문을 제출하십시오.

```
CREATE WRAPPER my_xml LIBRARY 'libdb2lsxml.a'
OPTIONS(DB2_FENCED 'N');
```

지원되는 플랫폼별로 XML 랩퍼에 대한 디폴트 라이브러리 파일 이름의 표는 아래의 관련 태스크 절에서 'DB2 Life Sciences Data Connect를 설치한 후'를 참조하십시오. CREATE WRAPPER문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

이 태스크의 다음 태스크는 *XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정*입니다.

관련 태스크:

- 17 페이지의『테이블 구조 파일 랩퍼 등록』

- 38 페이지의『Documentum 랩퍼 등록』
- 76 페이지의『Excel 랩퍼 등록』
- 102 페이지의『BLAST 랩퍼 등록』
- 126 페이지의『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』

XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정

이 태스크는 *페더레이티드 시스템에 XML 추가*에 대한 기본 태스크의 일부입니다. XML 문서에 액세스할 때 성능을 개선하려면 DB2_DJ_COMM 환경 변수를 설정하십시오. 이 변수는 페더레이티드 서버가 초기화 중에 랩퍼를 로드하는지 여부를 결정합니다.

절차:

DB2_DJ_COMM 환경 변수를 설정하려면 연관된 CREATE WRAPPER문에 지정한 랩퍼에 해당하는 랩퍼 라이브러리와 함께 `db2set` 명령을 제출하십시오.

```
db2set DB2_DJ_COMM='libdb2lsxml.a'
```

등호(=)의 양쪽에 스페이스가 없게 하십시오.

데이터베이스 시작 중 랩퍼 라이브러리 로드와 연관된 오버헤드가 있습니다. 이 오버헤드를 방지하려면 액세스하려는 라이브러리만을 지정하십시오.

DB2_DJ_COMM 환경 변수에 대한 자세한 정보는 *DB2 관리 안내서*를 참조하십시오.

이 태스크의 다음 태스크는 *XML 데이터 소스에 대한 서버 등록*입니다.

관련 태스크:

- 18 페이지의『테이블 구조 파일 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 39 페이지의『Documentum 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 103 페이지의『BLAST 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 125 페이지의『XML 랩퍼 등록』

- 127 페이지의『XML 데이터 소스에 대한 서버 등록』

XML 데이터 소스에 대한 서버 등록

이 태스크는 *페더레이티드 시스템에 XML 추가*에 대한 기본 태스크의 일부입니다. 랩퍼를 등록한 후에 해당 서버를 등록해야 합니다.

절차:

페더레이티드 시스템에 XML 서버를 등록하려면 CREATE SERVER문을 사용하십시오.

```
CREATE SERVER xml_server WRAPPER my_xml
```

여기서,

WRAPPER

연관된 CREATE WRAPPER문을 사용하여 등록한 랩퍼의 이름을 지정합니다. 이 인수는 필수입니다.

주: XML 랩퍼는 TYPE 및 VERSION 키워드를 사용하지 않습니다. 이러한 키워드가 CREATE SERVER문에 사용된 경우 오류가 발생합니다.

이 태스크의 다음 태스크는 *XML 데이터 소스에 대한 별칭* 등록입니다.

관련 태스크:

- 19 페이지의『테이블 구조 파일에 대한 서버 등록』
- 40 페이지의『Documentum 데이터 소스에 대한 서버 등록』
- 76 페이지의『Excel 데이터 소스에 대한 서버 등록』
- 104 페이지의『BLAST 데이터 소스에 대한 서버 등록』
- 126 페이지의『XML 랩퍼에 대한 DB2_DJ_COMM 환경 변수 설정』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』

XML 데이터 소스에 대한 별칭 등록

이 태스크는 *페더레이티드 시스템에 XML 추가*에 대한 기본 태스크의 일부입니다. XML 데이터 소스의 트리 구조를 모델화하는 별칭을 작성해야 합니다. 트리의 상위 또는 루트를 모델화하는 상위 별칭을 작성해야 합니다. 상위 별칭에 해당하는 요소에 중첩된 요소를 모델화하는 하위 별칭을 작성해야 합니다.

상위 및 하위 별칭은 CREATE NICKNAME문에 지정된 기본 및 외부 키에 의해 연결됩니다.

각 별칭은 다음과 같은 역할을 하는 XPath 표현식에 의해 정의됩니다.

- 개별 쌍(tuple)을 나타내는 XML 요소 식별
- 각 요소에서 컬럼 값을 추출하는 방법 지정

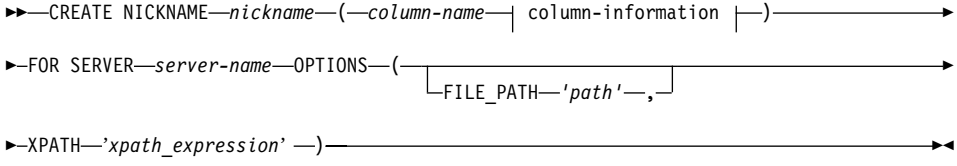
두 방법 중 한 방법으로 별칭이 XML 문서와 연관됩니다.

- 고정 방식(FILE_PATH 별칭 옵션 사용). 이 옵션을 사용하면 별칭은 특정 XML 문서의 데이터를 나타냅니다.
- 쿼리시에 지정된 파일 이름 사용(DOCUMENT 별칭 컬럼 옵션 사용). 이 옵션을 사용하면 스키마가 별칭 정의와 일치하는 XML 문서의 데이터를 나타내는 데 별칭을 사용할 수 있습니다.

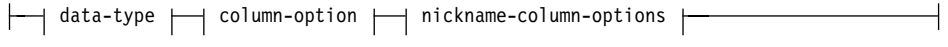
이러한 옵션에 대한 자세한 정보는 아래의 절차 절에 제공되어 있습니다.

절차:

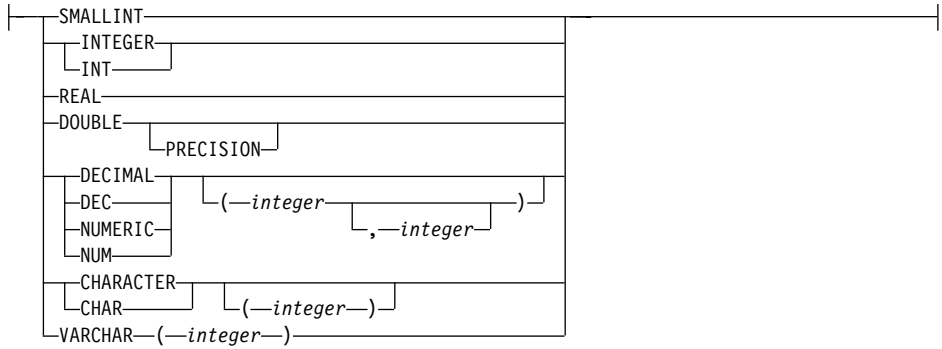
관계 테이블에 XML 데이터 소스를 맵핑하려면 CREATE NICKNAME문을 사용하여 별칭을 작성해야 합니다.



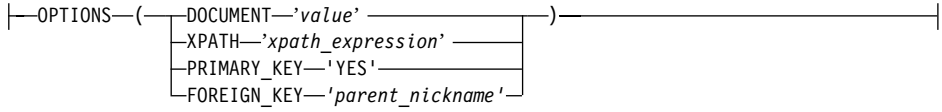
column-information:



data-type:



nickname-column-options:



column-option:

별칭 옵션

FILE_PATH

XML 문서의 파일 경로를 지정합니다. 이 별칭 옵션이 지정된 경우, DOCUMENT 별칭 컬럼 옵션을 지정해서는 안됩니다. 이 옵션은 루트 별칭(XML 문서의 최상위 레벨에 있는 요소를 식별하는 별칭)의 경우에만 승인됩니다.

XPATH

개별 쌍(tuple)을 나타내는 XML 요소를 식별한 XPath 표현식을 지정합니다. 하위 별칭에 대한 XPATH 별칭 옵션은 상위의 XPATH 별칭 옵션에 의해 지정된 경로의 컨텍스트에서 평가됩니다. 이 XPath 표현식은 XPATH 별칭 컬럼 옵션에 의해 식별된 컬럼 값을 평가하는 컨텍스트로 사용됩니다.

별칭 컬럼 옵션

DOCUMENT

XML 데이터의 종류를 지정합니다. 현재 XML 랩퍼는 FILE만을 지원합니다. 이 옵션은 루트 별칭(XML 문서의 최상위 레벨에 있는 요소를 식별하는 별칭)의 경우에만 승인됩니다. 별칭 당 하나의 컬럼만을 DOCUMENT 옵션과 함께 지정할 수 있습니다. DOCUMENT 옵션과 연관된 컬럼의 데이터 유형은 VARCHAR이어야 합니다.

FILE_PATH 별칭 옵션 대신 DOCUMENT 별칭 컬럼 옵션을 사용하는 것은 이 별칭에 해당하는 문서가 쿼리 실행 중에 제공되지 않을 것임을 의미합니다. DOCUMENT 옵션에 ″FILE″ 값이 있으면 문서가 포함된 파일의 이름이 쿼리 실행 중에 제공될 것임을 의미합니다. 다음의 CREATE NICKNAME 예는 DOCUMENT 별칭 컬럼 옵션의 사용을 설명합니다.

```
CREATE NICKNAME customers
(
   doc      VARCHAR(100)   OPTIONS(DOCUMENT 'FILE'),
   name     VARCHAR(16)    OPTIONS(XPATH './/name'),
   address  VARCHAR(30)    OPTIONS(XPATH './/address/@street'),
   cid      VARCHAR(16)    OPTIONS(PRIMARY_KEY 'YES'))
   FOR SERVER xml_server
   OPTIONS(XPATH '//customer');
```

WHERE절에 XML 문서의 위치를 지정하는 다음의 쿼리는 이제 customers 별칭에 대해 실행할 수 있습니다.

```
SELECT * FROM customers WHERE doc='/home/db2user/Customers.xml';
```

XPATH

이 컬럼에 해당하는 데이터를 찾을 수 있는 XML 문서의 XPath 표현식을 지정합니다. 이 XPath 표현식은 XPATH 별칭 옵션에 지정된 XPath 표현식을 평가한 후에 적용됩니다.

PRIMARY_KEY

상위 별칭임을 나타냅니다. 컬럼 데이터 유형은 항상 VARCHAR(16)이어야 합니다. 별칭에는 많아야 하나의 PRIMARY_KEY 컬럼 옵션이 있을 수 있습니다. ‘YES’만이 허용되는 값입니다. 이 옵션을 사용하여 지정된 컬럼은 랩퍼에 의해 생성된 키를 보유합니다. 컬럼의 값은 SELECT 목

록에서 검색할 수 없으며 이 컬럼에 대해 XPATH 옵션을 지정해서는 안 됩니다. 컬럼은 상위 및 하위 별칭을 조인하는 데에만 사용할 수 있습니다.

FOREIGN_KEY

하위 별칭임을 나타내며 해당 상위 별칭의 이름을 지정합니다. 별칭에는 많아야 하나의 FOREIGN_KEY 컬럼 옵션이 있을 수 있습니다. 이 옵션의 값은 대소문자를 구분합니다. 이 옵션을 사용하여 지정된 컬럼은 랩퍼에 의해 생성된 키를 보유합니다. 컬럼의 값은 SELECT 목록에서 검색할 수 없으며 이 컬럼에 대해 XPATH 옵션을 지정해서는 안됩니다. 컬럼은 상위 및 하위 별칭을 조인하는 데에만 사용할 수 있습니다.

별칭 예

다음의 예는 132 페이지의 그림 12의 샘플 문서에 표시된 샘플 XML 파일을 사용하여 XML 데이터 소스에 대한 별칭을 작성하기 위한 절차를 나타냅니다.

```
<doc>
   <customer id='123'>
      <name>...</name>
      <address>...</address>
       ...
      <order>
         <amount>...</amount>
            <date>...</date>
         <item quant='12'>
            <name>...</name>
         </item>
         <item quant='4'>...</item>
          ...
      </order>
      <order>...</order>
       ...
      <payment>
         <number>...</number>
         <date>...</date>
      </payment>
      <payment>...</payment>
       ...
   </customer>
   <customer id='124'>...</customer>
</doc)
```

그림 12. 샘플 XML 파일

상위 별칭 customers를 작성하려면 다음의 명령문을 지정하십시오.

```
CREATE NICKNAME customers
(
   id           VARCHAR(5)   OPTIONS(XPATH './@id')
   name         VARCHAR(16)  OPTIONS(XPATH './/name'),
   address      VARCHAR(30)  OPTIONS(XPATH './/address/@street'),
   cid          VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'))
   FOR SERVER xml_server
   OPTIONS( XPATH '//customer',
            FILE_PATH '/home/db2user/Customers.xml');
```

customer의 하위인 orders, payments 및 items에 대한 별칭을 작성하려면 다음 세 개의 별칭 명령문을 지정하십시오.

orders의 경우:

```
CREATE NICKNAME orders
(
   amount   INTEGER       OPTIONS(XPATH './amount'),
   date     VARCHAR(10)   OPTIONS(XPATH './date'),
   oid      VARCHAR(16)   OPTIONS(PRIMARY_KEY 'YES'),
   cid      VARCHAR(16)   OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
   FOR SERVER xml_server
   OPTIONS( XPATH './/order');
```

payments의 경우:

```
CREATE NICKNAME payments
(
   number   INTEGER       OPTIONS(XPATH './number'),
   date     VARCHAR(10)   OPTIONS(XPATH './date'),
   cid      VARCHAR(16)   OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
   FOR SERVER xml_server
   OPTIONS( XPATH './/payment');
```

items의 경우:

```
CREATE NICKNAME items
(
   name        VARCHAR(20)   OPTIONS(XPATH './name'),
   quantity    INTEGER       OPTIONS(XPATH './@quant'),
   oid         VARCHAR(16)   OPTIONS(FOREIGN_KEY 'ORDERS'))
   FOR SERVER xml_server
   OPTIONS( XPATH './/item');
```

이 태스크의 다음 태스크는 *루트가 아닌 별칭에 대한 페더레이티드 뷰 작성(XML 랩퍼)*입니다.

관련 태스크:

- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』
- 127 페이지의『XML 데이터 소스에 대한 서버 등록』
- 145 페이지의 제 8 장『비용산정 별칭 옵션 지정』

루트가 아닌 별칭에 대한 페더레이티드 뷰 작성(XML 랩퍼)

이 태스크는 *페더레이티드 시스템에 XML 추가*에 대한 기본 태스크의 일부입니다. XML 문서를 설명하는 별칭의 계층 구조에 대해 페더레이티드 뷰를 정의하는 것이 바람직합니다. 페더레이티드 뷰를 정의하면 루트가 포함되지 않은 XML 별칭 계층 구조의 조각들을 조인하는 쿼리와 특수 PRIMARY_KEY 및 FOREIGN_KEY 컬럼 이외의 컬럼에 대해 조인하는 쿼리가 올바로 실행됩니다.

절차:

모든 필수 술어와 루트에 대한 전체 경로가 포함된 페더레이티드 뷰를 정의하려면 다음의 단계를 수행하십시오.

1. 루트에 대한 경로에서 모든 별칭의 조인으로 루트가 아닌 각 별칭에 대한 뷰를 정의하십시오.
2. WHERE절에서 PRIMARY_KEY 및 FOREIGN_KEY 컬럼을 통해 조인 술어를 작성하십시오.
3. SELECT 목록에서 FOREIGN_KEY 별칭 컬럼 옵션을 사용하여 지정된 것을 제외하고 루트가 아닌 별칭의 모든 컬럼을 포함시키십시오.
4. SELECT 목록에 PRIMARY_KEY 옵션을 사용하여 지정된 상위 별칭의 컬럼을 포함시키십시오.

뷰 예:

다음의 예는 뷰의 사용을 설명합니다. 이 예에서는 135 페이지의 그림 13에 표시된 샘플 파일의 별칭이 `customers`, `orders`, `payments` 및 `items`로 이미 작성되었다고 가정합니다.

```
<doc>
   <customer id='123'>
      <name>...</name>
      <address>...</address>
       ...
      <order>
         <amount>...</amount>
            <date>...</date>
         <item quant='12'>
            <name>...</name>
         </item>
         <item quant='4'>...</item>
          ...
      </order>
      <order>...</order>
       ...
      <payment>
         <number>...</number>
         <date>...</date>
      </payment>
      <payment>...</payment>
       ...
   </customer>
   <customer id='124'>...</customer>
</doc)
```

그림 13. 샘플 XML 파일

루트가 아닌 별칭 order, payment 및 item에 대한 뷰는 다음과 같습니다.

order의 경우:

```
CREATE FEDERATED VIEW order_view AS
SELECT o.amount, o.date, o.oid, c.cid
FROM customers c, orders o
WHERE c.cid = o.cid;
```

payment의 경우:

```
CREATE FEDERATED VIEW payment_view AS
SELECT p.amount, p.date, c.cid
FROM customers c, payments p
WHERE c.cid = p.cid;
```

item의 경우:

```
CREATE FEDERATED VIEW item_view AS
SELECT it.quantity, it.name, o.oid
FROM customers c, orders o, items i
WHERE c.cid = o.cid AND o.oid = i.oid;
```

루트에 대한 조인 경로가 있기 때문에 이들 뷰에 제출된 쿼리가 올바로 처리됩니다.

예를 들어, 다음의 쿼리는 동일한 날짜에서 고객의 주문량과 지불 금액을 한 쌍으로 묶습니다.

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
 p.cid = o.cid;
```

이 태스크 다음에는 더 이상의 태스크가 없습니다.

관련 태스크:

- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』

XML 데이터 소스에 대한 쿼리 실행

이 절에서는 CREATE NICKNAME문을 사용하여 작성된 별칭 customers, orders 및 items를 사용하는 몇 가지 샘플 쿼리를 나열합니다.

절차:

쿼리를 실행하려면 다음의 예를 지침으로 사용하십시오.

다음의 쿼리는 모든 고객 이름을 표시합니다.

```
SELECT name FROM customers;
```

다음의 쿼리는 고객 이름이 'Smith'인 모든 레코드를 표시합니다.

```
SELECT * FROM customers where name='Smith';
```

다음의 쿼리는 각 고객의 주문에 대한 금액과 고객 이름을 표시합니다.

```
SELECT c.name, o.amount FROM customers c, orders o where c.cid=o.cid;
```

orders 및 customers 별칭 사이의 상하위 관계를 나타내려면 조인 c.cid=o.cid 가 필요합니다.

다음의 쿼리는 각 고객의 각 주문 및 품목에 대한 주문량, 품목 이름 및 고객 주소를 선택합니다.

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
WHERE c.cid=o.cid AND o.oid=i.oid;
```

상하위 관계를 유지하려면 다시 두 개의 조인이 필요합니다.

다음의 두 예는 FILE_PATH 별칭 옵션은 지정하지 않지만 DOCUMENT 별칭 컬럼 옵션을 지정하는 별칭을 사용하여 쿼리를 작성하는 방법을 보여줍니다. 별칭 customers를 작성하는 데 사용되는 해당 CREATE NICKNAME문은 아래와 같습니다.

```
CREATE NICKNAME customers
(
   doc      VARCHAR(100)   OPTIONS(DOCUMENT 'FILE'),
   name     VARCHAR(16)    OPTIONS(XPATH './/name'),
   address  VARCHAR(30)    OPTIONS(XPATH './/address/@street'),
   cid      VARCHAR(16)    OPTIONS(PRIMARY_KEY 'YES'))
   FOR SERVER xml_server
   OPTIONS(XPATH '//customer');
```

다음의 쿼리는 파일 경로가 /home/db2user/Customers.xml인 XML 파일 Customers.xml에서 모든 데이터를 선택합니다.

```
SELECT * FROM customers WHERE doc='/home/db2user/Customers.xml';
```

다음의 쿼리는 /home/db2user/Customers.xml에 위치한 XML 파일 Customers.xml로부터 수량이 1000개 이상인 주문의 고객 이름 및 주문 날짜를 선택합니다.

```
SELECT c.name, o.date FROM customers c, orders o
WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```

관련 태스크:

- 59 페이지의『Documentum 데이터 소스에 대한 쿼리 실행』
- 79 페이지의『Excel 데이터 소스에 대한 쿼리 실행』

XML 랩퍼에 대한 제한사항 및 고려사항

이 절에는 XML 랩퍼의 사용과 연관된 제한사항 및 고려사항 목록이 들어 있습니다.

- passthru 기능은 지원되지 않습니다.
- XML 문서는 읽기만 가능합니다.

관련 참조:

- 25 페이지의『테이블 구조 파일 랩퍼에 대한 랩퍼 제한사항 및 고려사항』
- 25 페이지의『테이블 구조 파일 랩퍼에 대한 파일 제한사항 및 고려사항』
- 65 페이지의『Documentum 랩퍼에 대한 제한사항 및 고려사항』
- 83 페이지의『Excel 랩퍼에 대한 랩퍼 제한사항』
- 83 페이지의『Excel 파일 제한사항』

XML 랩퍼에 대한 메시지

이 절에서는 XML의 랩퍼에 대해 작업할 때 발생할 수 있는 메시지를 나열하고 설명합니다. 메시지에 대한 자세한 정보는 *DB2 메시지 참조서*를 참조하십시오.

표 26. XML의 랩퍼에 의해 발행되는 메시지

오류 코드	메시지	설명
SQL0405N	숫자 리터럴 "<column_name>"의 값이 범위에서 벗어났기 때문에 유효하지 않습니다.	지정된 숫자 리터럴이 허용되는 범위에 있지 않습니다. CREATE NICKNAME문에서 컬럼의 데이터 유형을 점검하십시오.
SQL0408N	값이 지정 목표의 데이터 유형과 호환되지 않습니다. 목표 이름은 "<column_name>"입니다.	컬럼에 지정할 값의 데이터 유형이 지정 목표의 선언된 데이터 유형과 호환되지 않습니다. CREATE NICKNAME문에서 컬럼의 데이터 유형을 점검하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "랩퍼 오브젝트 작성 오류".)	새로운 랩퍼 오브젝트를 작성할 때 오류가 발생했습니다. DB2 지원부에 문의하십시오.

표 26. XML의 랩퍼에 의해 발행되는 메시지 (계속)

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "Xerces 초기화 오류".)	Xerces 구문 분석기의 초기화 중 예외가 발생했습니다. DB2 지원부에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "<xalan_error_message>".)	Xalan 함수를 호출할 때 오류가 발생했습니다. XML 문서를 점검하십시오. 문서가 올바로 구조화되었으면 Xalan 문서에서 오류 메시지에 대한 자세한 내용을 참조하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "XalanDOMException: 예외 코드는 <exception_code>입니다".)	XalanDOMException 예외가 발생했습니다. 예외 코드의 의미를 보려면 Xalan 문서를 참조하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "노드 값 확보 오류".)	Xalan이 유효하지 않은 노드에 액세스하려고 시도했습니다. DB2 지원부에 문의하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "XML 문서 구문 분석 오류".)	XML 문서를 구문 분석할 때 오류가 발생했습니다. XML 문서를 점검하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "XML 문서의 루트 요소 확보 오류".)	XML 문서를 구문 분석한 후에 Xalan이 루트 요소를 검색하려고 시도했지만 실패했습니다. XML 문서를 점검하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "XPath 표현식 평가시 지정되지 않은 예외 발생".)	XPath 표현식을 평가할 때 Xalan에 의해 지정되지 않은 예외가 생성되었습니다. XML 문서를 점검하고 Xalan 문서를 참조하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "노드 값 확보시 지정되지 않은 예외 발생".)	노드 값을 검색할 때 Xalan에 의해 지정되지 않은 예외가 생성되었습니다. XML 문서를 점검하고 Xalan 문서를 참조하십시오.

표 26. XML의 랩퍼에 의해 발행되는 메시지 (계속)

오류 코드	메시지	설명
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "DOM 트리 빌드 중 지정되지 않은 예외 발생".)	XML 문서에 대한 DOM 트리를 빌드 중일 때 Xalan에 의해 지정되지 않은 예외가 생성되었습니다. XML 문서를 점검하고 Xalan 문서를 참조하십시오.
SQL0901N	심각하지 않은 시스템 오류로 인해 SQL문이 실패했습니다. 후속 SQL 문은 처리될 수 있습니다. (이유 "메모리 할당 오류".)	메모리 할당시 오류가 발생했습니다.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "컬럼 데이터 유형이 지원되지 않습니다."입니다.	별칭 컬럼에 지원되지 않는 데이터 유형이 있습니다. CREATE NICKNAME문을 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "TYPE절이 지원되지 않습니다."입니다.	CREATE SERVER문에 TYPE절이 들어 있습니다. 이 절은 XML 랩퍼에 의해 지원되지 않습니다. 제거하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "VERSION절이 지원되지 않습니다."입니다.	CREATE SERVER문에 VERSION절이 들어 있습니다. 이 절은 XML 랩퍼에 의해 지원되지 않습니다. 제거하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "DOCUMENT 컬럼에 유효하지 않은 술어 사용"입니다.	쿼리에 피연산자가 올바르지 않은 술어가 들어 있습니다. 쿼리에서 술어를 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "FOREIGN_KEY 컬럼에 유효하지 않은 술어 사용"입니다.	쿼리에 피연산자가 올바르지 않은 술어가 들어 있습니다. 쿼리에서 술어를 점검하십시오.

표 26. XML의 랩퍼에 의해 발행되는 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″PRIMARY_KEY 컬럼에 유효하지 않은 술어 사용″입니다.	쿼리에 피연산자가 올바르지 않은 술어가 들어 있습니다. 쿼리에서 술어를 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″XPATH 및 DOCUMENT 옵션이 호환되지 않습니다.″입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″XPATH 및 FOREIGN_KEY 옵션이 호환되지 않습니다.″입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″XPATH 및 PRIMARY_KEY 옵션이 호환되지 않습니다.″입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″DOCUMENT 및 FOREIGN_KEY 옵션이 호환되지 않습니다.″입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″DOCUMENT 및 PRIMARY_KEY 옵션이 호환되지 않습니다.″입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.

표 26. XML의 랩퍼에 의해 발행되는 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "FOREIGN_KEY 및 PRIMARY_KEY 옵션이 호환되지 않습니다."입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "컬럼 옵션 누락"입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트 및 토큰은 "DOCUMENT 컬럼 옵션이 고유하지 않습니다."입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트 및 토큰은 "FOREIGN_KEY 컬럼 옵션이 고유하지 않습니다."입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "PRIMARY_KEY 컬럼 옵션이 고유하지 않습니다."입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. 구문을 점검하십시오.
SQL1822N	데이터 소스 "XML 랩퍼"로부터 예기치 않은 오류 코드 "<trace_point>"을(를) 받았습니다. 연관된 텍스트와 토큰은 "유효하지 않은 DOCUMENT 옵션 값"입니다.	CREATE NICKNAME문에 지정된 DOCUMENT 옵션의 값이 유효하지 않습니다. 'FILE'이어야만 합니다. CREATE NICKNAME 문을 점검하십시오.

표 26. XML의 랩퍼에 의해 발행되는 메시지 (계속)

오류 코드	메시지	설명
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″유효하지 않은 PRIMARY_KEY 옵션 값″입니다.	CREATE NICKNAME문에 지정된 PRIMARY_KEY 옵션의 값이 유효하지 않습니다. 'YES'이어야만 합니다. CREATE NICKNAME문을 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″유효하지 않은 FOREIGN_KEY 옵션 값″입니다.	CREATE NICKNAME문에 지정된 FOREIGN_KEY 옵션의 값이 유효하지 않습니다. 옵션 값과 일치하는 상위 별칭을 찾을 수 없습니다. CREATE NICKNAME문을 점검하십시오.
SQL1822N	데이터 소스 ″XML 랩퍼″로부터 예기치 않은 오류 코드 ″<trace_point>″을(를) 받았습니다. 연관된 텍스트와 토큰은 ″FILE_PATH 및 DOCUMENT 옵션이 호환되지 않습니다.″입니다.	CREATE NICKNAME문이 지정된 대로 올바르지 않습니다. FILE_PATH 및 DOCUMENT 옵션을 동시에 지정할 수 없습니다. CREATE NICKNAME 구문을 점검하십시오.
SQL1881N	″<option_name>″은(는) ″<object_name>″에 대해 유효한 ″<option_type>″이(가) 아닙니다.	지정된 옵션이 존재하지 않거나 이 특정 데이터 소스에 대해 유효하지 않을 수 있습니다. CREATE NICKNAME문을 점검하십시오.
SQL1883N	″<option_name>″은(는) ″<object_name>″에 대해 필수 ″<option_type>″ 옵션입니다.	필수 DB2 옵션이 지정되지 않았습니다. CREATE NICKNAME문을 점검하십시오.

관련 참조:

- 27 페이지의『테이블 구조 파일 랩퍼에 대한 메시지』
- 67 페이지의『Documentum 랩퍼에 대한 메시지』
- 84 페이지의『Excel 랩퍼에 대한 메시지』
- 116 페이지의『BLAST 랩퍼에 대한 메시지』

제 8 장 비용산정 별칭 옵션 지정

효율적인 실행 플랜을 작성하기 위해 옵티마이저는 여러 플랜 세트를 생성하고 각 플랜에 필요한 자원을 미리 예상합니다. 최소의 자원이 필요한 플랜이 평가에 사용됩니다.

외부 데이터 소스에 의해 실행된 쿼리 부분에 대한 예상 평가 시간은 랩퍼가 제공합니다. 이 계산에 사용된 공식은 특정 설치에 맞게 변경할 수 있는 세 개의 기본 매개변수를 사용합니다.

이러한 매개변수는 다음의 별칭 옵션으로 지정됩니다.

RESET_COST
: 외부 서버에 연결하고 결과를 다시 얻는 시간(밀리초)을 지정합니다.

ADVANCE_COST
: 각 행을 확보하는 시간(밀리초)을 지정합니다.

BIND_COST
: 랩퍼에서 외부 소스로 매개변수를 전달하는 데 필요한 시간(밀리초)을 지정합니다.

모든 값은 정수로 제공해야 합니다. 디폴트값은 *DB2 Life Sciences Data Connect 릴리스 정보 버전 8*을 참조하십시오.

절차:

비용산정 별칭 옵션을 지정하려면 다음과 같이 하십시오.

1. 설치를 분석하여 비용산정 옵션의 사용자 정의가 소속된 회사의 페더레이티드 쿼리 처리에 이득이 되는지 판별하십시오.
2. 이득이 된다고 판단되면 랩퍼의 CREATE NICKNAME문에 별칭 옵션으로 하나 이상의 비용산정 옵션을 추가하십시오.
3. CREATE NICKNAME문을 제출하십시오.

CREATE NICKNAME문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

관련 태스크:

- 20 페이지의『테이블 구조 파일에 대한 별칭 등록』
- 43 페이지의『Documentum 데이터 소스에 대한 별칭 등록』
- 77 페이지의『Excel 데이터 소스에 대한 별칭 등록』
- 105 페이지의『BLAST 데이터 소스에 대한 별칭 등록』
- 147 페이지의『별칭 대체』
- 128 페이지의『XML 데이터 소스에 대한 별칭 등록』

제 9 장 별칭 대체

이 장에서는 ALTER NICKNAME문을 사용하여 이전에 등록된 별칭을 변경하는 방법에 대해 설명합니다.

별칭 대체

ALTER NICKNAME문을 사용하여 데이터 소스 또는 뷰의 페더레이티드 데이터베이스의 표현을 수정하십시오.

제한사항:

ALTER NICKNAME문은 DB2 Life Sciences Data Connect 랩퍼에 대한 컬럼 이름을 변경할 때 사용할 수 없습니다.

절차:

변경 컬럼 값을 변경하려면 ALTER NICKNAME문을 사용하여 다음을 수행해야 합니다.

- 이 컬럼의 로컬 데이터 유형을 변경
- 이 컬럼의 옵션을 추가, 변경 또는 삭제

ALTER NICKNAME문에 대한 자세한 정보는 *DB2 SQL 참조서*를 참조하십시오.

관련 태스크:

- 148 페이지의『데이터 유형 변경』
- 148 페이지의『별칭 옵션 변경』

데이터 유형 변경

ALTER NICKNAME문을 사용하면 컬럼의 데이터 유형을 변경할 수 있습니다.

절차:

컬럼의 데이터 유형을 변경하려면 ALTER NICKNAME문을 사용하십시오.

예를 들어, 다음의 ALTER NICKNAME문은 DRUG 컬럼의 로컬 데이터 유형을 CHAR(30)으로 변경합니다. DRUG 컬럼은 원래 CREATE NICKNAME문을 사용하여 CHAR(20)으로 정의됩니다. 별칭 DRUGDATA1은 로컬 테이블 구조 파일인 drugdata1.txt를 나타냅니다.

```
ALTER NICKNAME DRUGDATA1
   ALTER COLUMN DRUG
   LOCAL TYPE CHAR(30)
```

관련 태스크:

- 147 페이지의『별칭 대체』
- 148 페이지의『별칭 옵션 변경』

별칭 옵션 변경

ALTER NICKNAME문을 사용하면 별칭 옵션을 변경할 수 있습니다.

절차:

별칭 옵션을 변경하려면 ALTER NICKNAME문을 사용하십시오.

예를 들어, 다음의 ALTER NICKNAME문은 테이블 구조 파일, drugdata1.txt의 완전한 경로를 변경합니다. 경로는 원래 CREATE NICKNAME문을 사용하여 '/user/pat/drugdata1.txt'로 정의되었습니다. 별칭 DRUGDATA1은 로컬 테이블 구조 파일인 drugdata1.txt를 나타냅니다.

```
ALTER NICKNAME DRUGDATA1
   OPTIONS (SET FILE_PATH '/usr/kelly/data/drugdata1.txt')
```

관련 태스크:

- 147 페이지의『별칭 대체』
- 148 페이지의『데이터 유형 변경』

주의사항

IBM은 다른 국가에서는 이 자료에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급하는 것이 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 사용권까지 부여하는 것은 아닙니다. 사용권에 대한 의문사항은 다음으로 문의하십시오.

135-270
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터
전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 사용권 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. **IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을『현상태대로』제공합니다.** 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및(또는) 프로그램을 사전 통지없이 언제든지 개선 및(또는) 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램 및 기타 프로그램(이 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 사용권자는 다음 주소로 문의하십시오.

135-270
서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들어, 사용권 지불 등)에 따라 사용할 수 있습니다.

이 정보에 기술된 사용권 프로그램 및 사용 가능한 모든 사용권 자료는 IBM이 IBM 기본 계약, IBM 프로그램 사용권 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서, 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 레벨 상태의 시스템에서 측정되었을 수 있으므로, 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한, 일부 성능은 추정을 통해 추측되었을 수도 있으므로, 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 사용자의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 배상 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 언급은 특별한 통지없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이 예제에는 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 포함될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권:

이 정보에는 여러 가지 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 샘플 응용프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스에 부합하는 응용프로그램의 개발, 사용, 마케팅 또는 배포를 목적으로 이들 샘플 프로그램을 복사, 수정 및 배포할 수 있으며 IBM에 대한 지불 의무는 없습니다. 이러한 예제가 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서, IBM은 이들 프로그램의 신뢰성, 서비스 가능성 또는 기능에 대해 어떠한 보증도 하지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 다음과 같은 저작권 표시가 반드시 포함되어야 합니다.

© (*귀하의 회사명*) (연도). 이 코드의 일부는 IBM Corp. 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. _연도 *입력*_. All rights reserved.

상표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 상표이며, 이러한 용어는 DB2 UDB 문서 라이브러리에 있는 문서 중 적어도 하나의 문서에 사용되었습니다.

ACF/VTAM
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
z/OS
zSeries

다음 용어는 기타 회사의 상표 또는 등록상표이며, DB2 UDB 문서 라이브러리의 최소 하나의 문서에 있는 문서 중 적어도 하나의 문서에 사용되었습니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Intel 및 Pentium은 미국 및 기타 국가에서 사용되는 Intel Corporation의 상표입니다.

Java 및 모든 Java 관련 상표는 미국 및 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

UNIX는 미국 및 기타 국가에서 Open Group의 등록 상표입니다.

기타 회사, 제품 또는 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

관련 서적

이 관련 서적에는 DB2 Life Sciences Data Connect에서 작업시 도움이 되는 DB2 Universal Database 책이 들어 있습니다.

- *DB2 Connect 사용자 안내서*(SA30-0993)
- *UNIX용 DB2 빠른 시작*(GA30-0984)
- *DB2 SQL 참조서*(SA30-0997)
- *DB2 관리 안내서: 계획*(SA30-0990)
- *DB2 관리 안내서: 구현*(SA30-0988)
- *DB2 관리 안내서: 성능*(SA30-0989)
- *DB2 메시지 참조서*(GA30-0986)
- *IBM DB2 Universal Database 페더레이티드 시스템 안내서*(GA30-1511)
- *DB2 Life Sciences Data Connect 릴리스 정보 버전 8*

색인

[라]

랩퍼
  생명 과학, 플랫폼별 5
  정의 1
  플랫폼별 디폴트 라이브러리 이름 11

[마]

메시지
  테이블 구조 파일 랩퍼 27
  BLAST 랩퍼 116
  Documentum 랩퍼 67
  Excel 랩퍼 84
  XML 랩퍼 116

[바]

별칭
  데이터 유형 변경 148
  변경 147
  별칭 옵션 변경 148
  비용산정 옵션 지정 145
별칭 비용산정 옵션 145

[사]

샘플 쿼리 59
  BLAST
    구조 105, 113
    샘플 114
  Documentum 59
  Excel 파일 79
  XML 136
생명 과학 데이터 소스 1

[차]

최적화
  테이블 구조 파일 27
  BLAST 116

[타]

테이블 구조 파일
  개요 13
  메시지 27
  예 13
  유형 14
  제한사항 및 고려사항 25
  최적화 27
  파일 액세스 제어 모델 26
  페더레이티드 시스템에 추가
    랩퍼 등록 17
    별칭 등록 20
    서버 등록 19
  DB2 Life Sciences Data Connect를 사용한 액세스 15

B

BLAST
  메시지 116
  설명 91
  페더레이티드 시스템에 추가
    랩퍼 등록 102
    별칭 등록 105
    서버 등록 104
    올바른 매트릭스 파일이 설치되어 있는지 확인 97
    올바른 blastall 실행 파일이 설치되어 있는지 확인 97
BLAST *(계속)*
  페더레이티드 시스템에 추가 *(계속)*
    BLAST 구성 파일 97
    BLAST 디먼 설정 및 구성 97
    BLAST 디먼 시작 101
    CREATE NICKNAME문 105
    CREATE SERVER문 104
    CREATE WRAPPER문 102

C

CREATE FEDERATED VIEW문
  XML 134
CREATE FUNCTION문
  Documentum 52
CREATE NICKNAME문
  테이블 구조 파일 20
  BLAST 105
  Documentum 43
  Excel 파일 77
  XML 128
CREATE SERVER문
  테이블 구조 파일 19
  BLAST 104
  Documentum 40
  Excel 파일 76
  XML 127
CREATE USER MAPPING문
  Documentum 42
CREATE WRAPPER문
  테이블 구조 파일 17
  BLAST 102
  Documentum 38
  Excel 파일 76
  XML 125

CreateNicknameFile 유틸리티,
Documentum 60
구성 61
설치 61
DM_ID 오브젝트 유형 맵핑 63

D

DB2_DJ_COMM 환경 변수 18, 39, 103, 126
DiscoveryLink 3
Documentum
메시지 67
문서에 대한 사용자 액세스 66
반복 속성 이중 정의 64
설명 31
예 31
제한사항 및 고려사항 65
페더레이티드 시스템에 추가
랩퍼 등록 38
별칭 등록 43
사용자 맵핑 42
사용자 정의 기능 등록 52
서버 등록 40
CREATE FUNCTION문 52
CREATE NICKNAME문 43
CREATE SERVER문 40
CREATE USER MAPPING문 42
CREATE WRAPPER문 38
CreateNicknameFile 유틸리티 60
Documentum 클라이언트 라이브러리에 링크(AIX 및 Solaris 운영 환경 전용) 35
Documentum의 클라이언트 dmcl.ini 파일 지정 36
CreateNicknameFile 유틸리티 60

E

Excel 파일
메시지 84
샘플 사용자 시나리오 80
설명 73
예 73
제한사항 및 고려사항 83
파일 액세스 제어 모델 84
페더레이티드 시스템에 추가
랩퍼 등록 76
별칭 등록 77
서버 등록 76
CREATE NICKNAME문 77
CREATE SERVER문 76

X

XML(eXtensible Markup Language)
메시지 138
설명 119
제한사항 및 고려사항 138
페더레이티드 시스템에 추가 125
랩퍼 등록 125
별칭 등록 128
비루트 별칭에 대한 페더레이티드 뷰 작성 134
서버 등록 127
CREATE FEDERATED VIEW문 134
CREATE NICKNAME문 128
CREATE SERVER문 127
CREATE WRAPPER문 125

IBM 한글 지원에 관한 설문

FAX : (02) 3787-0123

보내 주시는 의견은 더 나은 고객 지원 체제를 위한 귀중한 자료가 됩니다.
독자 여러분의 좋은 의견을 기다립니다.

책 제목: IBM® DB2® Life Sciences Data Connect 계획, 설치 및 구성 안내서 버전 8

책 번호: GA30-1512-00

성 명		직위/담당업무	
회 사 명		부 서 명	
주 소			
전화번호		팩스번호	
전자우편 주소			
사용중인 시스템	o 중대형 서버 o UNIX 서버 o PC 및 PC 서버		

1. IBM에서 제공하는 한글 책자와 영문 책자 중 어느 것을 더 좋아하십니까? 그 이유는 무엇입니까?
   ☐ 한글 책자 ☐ 영문 책자
   (이유:)

2. 본 책자와 해당 소프트웨어에서 사용된 한글 용어에 대한 귀하의 평가 점수는?
   ☐ 수 ☐ 우 ☐ 미 ☐ 양 ☐ 가

3. 본 책자와 해당 소프트웨어에서 번역 품질에 대한 귀하의 평가 점수는?
   ☐ 수 ☐ 우 ☐ 미 ☐ 양 ☐ 가

4. 본 책자의 인쇄 상태에 대한 귀하의 평가 점수는?
   ☐ 수 ☐ 우 ☐ 미 ☐ 양 ☐ 가

5. 한글 소프트웨어 및 책자가 지원되는 분야에 대해 귀하는 어떻게 생각하십니까?
   ☐ 한글 책자를 늘려야 함 ☐ 현재 수준으로 만족
   ☐ 그다지 필요성을 느끼지 않음

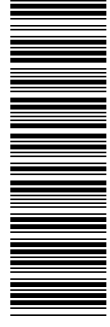
6. IBM은 인쇄물 형식(hardcopy)과 화면 형식(softcopy)의 두 종류로 책자를 제공합니다. 어느 형식을 더 좋아하십니까?
   ☐ 인쇄물 형식(hardcopy) ☐ 화면 형식(softcopy) ☐ 둘 다

☞ IBM 한글 지원 서비스에 대해 기타 제안사항이 있으시면 적어주십시오.

☺ 설문에 답해 주셔서 감사합니다.
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해 드리기 위해 최선을 다하겠습니다.

부품 번호: CT16FKO

(1P) P/N: CT16FKO



GA30-1512-00

Spine information:

IBM® DB2® Life Sciences Data Connect

DB2 LSDC 계획, 설치 및 구성 안내서

버전 8