

IBM® DB2 Universal Database™



Kurs Visual Explain

Wersja 8

IBM® DB2 Universal Database™



Kurs Visual Explain

Wersja 8

Przed skorzystaniem z tych informacji i opisywanych przez nie produktów należy przeczytać informacje ogólne, które zawiera rozdział *Uwagi*.

Niniejszy dokument zawiera informacje dotyczące produktów firmy IBM. Są one prezentowane zgodnie z warunkami umowy licencyjnej i są chronione prawem. Informacje zawarte w tej publikacji nie zawierają żadnych gwarancji dotyczących opisywanych produktów i żadnych zapisanych w niej stwierdzeń nie należy interpretować jako takich gwarancji.

Publikacje firmy IBM można zamówić poprzez stronę WWW lub u lokalnego przedstawiciela firmy IBM.

- Aby zamówić książki poprzez stronę WWW, należy skorzystać ze strony IBM Publications Center pod adresem www.ibm.com/shop/publications/order
- Aby znaleźć najbliższego lokalnego przedstawiciela firmy IBM, należy skorzystać z informacji umieszczonych na stronie IBM Directory of Worldwide Contacts pod adresem www.ibm.com/planetwide

Aby zamówić książki DB2 w firmie IBM w Stanach Zjednoczonych lub Kanadzie, należy zadzwonić do działu DB2 Marketing and Sales pod numer 1-800-IBM-4YOU (426-4968).

Wysłanie informacji do firmy IBM daje jej prawo do ich używania i dystrybucji w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich nadawcy.

Spis treści

Informacje o tym kursie	v
Informacje dotyczące określonych środowisk	vi

Lekcja 1. Tworzenie obrazów stanu wyjaśnienia	1
Tworzenie tabel wyjaśnienia	1
Korzystanie z obrazów stanu wyjaśnienia	2
Tworzenie obrazów stanu wyjaśnienia dynamicznych instrukcji SQL	3
Tworzenie obrazów stanu wyjaśnienia statycznych instrukcji SQL	4
Co dalej	5

Lekcja 2. Wyświetlanie i korzystanie z wykresu planu dostępu	7
Wyświetlanie wykresu planu dostępu przez wybranie elementu z listy uprzednio wyjaśnionych instrukcji SQL	7
Odczytywanie symboli na wykresie planu dostępu	8
Korzystanie z suwaka powiększenia w celu powiększania fragmentów wykresu	8
Uzyskiwanie bardziej szczegółowych informacji o obiektach na wykresie	9
Odczytywanie statystyk tabel, indeksów i funkcji tabelowych	9
Uzyskiwanie szczegółowych informacji o operatorach na wykresie	10
Uzyskiwanie statystyki dotyczącej funkcji	10
Uzyskiwanie statystyki dotyczącej obszarów tabel	10
Odczytywanie statystyki kolumn występujących w instrukcji SQL	10
Uzyskiwanie informacji na temat parametrów konfiguracyjnych i opcji wiązania	11
Zmiana wyglądu wykresu	11
Co dalej	11

Lekcja 3. Udoskonalanie planu dostępu w środowisku bazy danych z pojedynczą partycją	13
Praca z wykresami planów dostępu	13
Uruchamianie zapytania bez indeksów i bez statystyk	14

Gromadzenie aktualnej statystyki tabel i indeksów przy użyciu komendy runstats	17
Tworzenie indeksów dla kolumn używanych do łączenia tabel w zapytaniu	21
Tworzenie dodatkowych indeksów dla kolumn tabeli	27
Co dalej	29

Lekcja 4. Udoskonalanie planu dostępu w środowisku partycjonowanej bazy danych	31
Praca z wykresami planów dostępu	31
Uruchamianie zapytania bez indeksów i bez statystyk	32
Gromadzenie aktualnej statystyki tabel i indeksów przy użyciu komendy runstats	35
Tworzenie indeksów dla kolumn używanych do łączenia tabel w zapytaniu	39
Tworzenie dodatkowych indeksów dla kolumn tabeli	43
Co dalej	46

Dodatek A. Pojęcia dotyczące narzędzia graficznego Visual Explain	47
Plan dostępu	47
Wykres planu dostępu	47
Węzeł wykresu planu dostępu	48
Klastrowanie	49
Kontener	49
Koszt	49
Blokowanie kursorów	50
Obszar tabel DMS	50
Dynamiczny SQL	50
Obraz stanu wyjaśnienia	51
Instrukcja SQL, którą można wyjaśnić	52
Instrukcja wyjaśniona	52
Operand	52
Operator	52
CMPEXP	54
DELETE	54
EISCAN	54
FETCH	54
FILTER	55
GENROW	55
GRPBY	56
HSJOIN	56

INSERT	57	GENROW	72
IXAND	57	GRPBY.	73
IXSCAN	58	HSJOIN	73
MSJOIN	59	INSERT	74
NLJOIN	59	IXAND	74
PIPE	60	IXSCAN	75
RETURN	60	MSJOIN	76
RIDSCN	61	NLJOIN	76
RQUERY	61	PIPE	77
SORT	61	RETURN	77
TBSCAN	62	RIDSCN	78
TEMP	63	RQUERY	78
TQUEUE	64	SORT	78
UNION	64	TBSCAN	79
UNIQUE	64	TEMP	80
UPDATE	65	TQUEUE	81
Optymalizator	65	UNION.	81
Pakiet	65	UNIQUE	81
Predykat	65	UPDATE	82
Klasa optymalizacji zapytania	66		
Selektywność predykatów	67	Dodatek C. Pojęcia związane z programem	
Połączenia gwiazdziste	68	DB2	83
Statyczny SQL	69	Bazy danych	83
Obszary tabel SMS	69	Schematy	83
Obszar tabel	69	Tabele	84
Visual Explain.	70		
		Dodatek D. Uwagi.	85
Dodatek B. Alfabetyczna lista operatorów		Znaki towarowe	88
narzędzia Visual Explain	71		
CMPEXP	71	Indeks	91
DELETE	71		
EISCAN	71	Kontakt z firmą IBM	93
FETCH	72	Informacje o produkcie	93
FILTER.	72		

Informacje o tym kursie

Ten kurs stanowi wprowadzenie do funkcji narzędzia graficznego DB2 Visual Explain. Podczas tego kursu użytkownik nauczy się, jak za pomocą narzędzia Visual Explain wyświetlać na wykresach plany dostępu dla wyjaśnionych instrukcji SQL oraz jak wykorzystywać informacje z takich wykresów do dostrajania swoich zapytań SQL w celu poprawy wydajności.

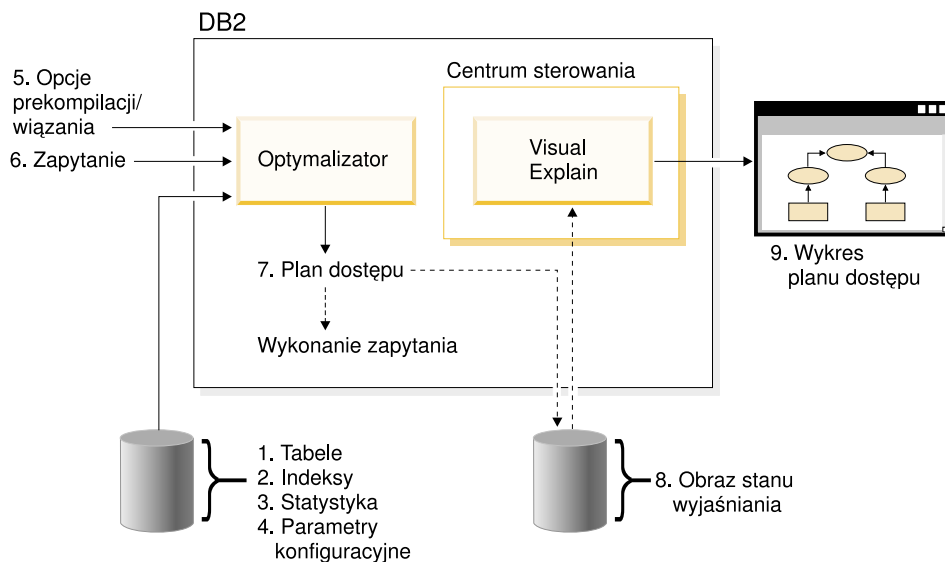
Za pomocą swojego optymalizatora program DB2 bada utworzone przez użytkownika zapytania SQL i określa najlepszą metodę dostępu do danych. Ta ścieżka dostępu do danych nazywana jest planem dostępu. W programie DB2 użytkownik może obejrzeć wyniki działania optymalizatora, wyświetlając plan dostępu, który został wybrany do zrealizowania określonego zapytania SQL. Za pomocą narzędzia graficznego Visual Explain plan dostępu może zostać wyświetlony w postaci wykresu. Wykres ten to graficzna reprezentacja biorących udział w zapytaniu obiektów bazy danych (na przykład tabel i indeksów). Wyświetlane są na nim także wykonywane na tych obiektach operacje (na przykład operacje skanowania lub sortowania), a także przepływ danych.

Efektywność dostępu do danych może zostać zwiększona przez wykonanie następujących czynności służących dostrajaniu zapytań:

1. Zmodyfikowanie projektu tabeli i zreorganizowanie danych tabeli.
2. Utworzenie odpowiednich indeksów.
3. Dostarczanie optymalizatorowi aktualnej statystyki przy użyciu komendy **runstats**.
4. Wybór odpowiednich parametrów konfiguracyjnych.
5. Wybór odpowiednich opcji wiązania.
6. Zaprojektowanie zapytań tak, aby pobierane były tylko potrzebne dane.
7. Praca z planem dostępu.
8. Utworzenie obrazów stanu wyjaśniania.
9. Wykorzystanie informacji zamieszczonych na wykresie planu dostępu.

Te czynności dotyczące poprawy wydajności odpowiadają czynnościom przedstawionym na poniższej ilustracji. (Przerywane linie wskazują czynności, które są

wymagane przez narzędzie graficzne Visual Explain).



Ten kurs obejmuje lekcje dotyczące następujących zagadnień:

- Tworzenie obrazów stanu wyjaśnienia. Czynność ta jest wymagana do wyświetlania wykresów planów dostępu.
- Wyświetlanie i posługiwanie się wykresem planu dostępu.
- Wykonywanie działań związanych z dostrajaniem i analizowanie wynikających z tych działań poprawy planu dostępu.

Uwaga: Problemy dostrajania wydajności występują w dwóch lekcjach: w lekcji środowiska baz danych z pojedynczą partycją i lekcji dotyczącej środowisk partycjonowanych baz danych.

Lekcje ilustrowane są dostarczoną z programem DB2 przykładową bazą danych SAMPLE. Jeśli nie została jeszcze utworzona przykładowa baza danych SAMPLE, należy zapoznać się z odpowiednimi instrukcjami zawartymi w podręczniku *Administration Guide*.

Informacje dotyczące określonych środowisk



Informacje oznakowane tą ikoną odnoszą się tylko do środowisk baz danych o jednej partycji.



Informacje oznakowane tą ikoną odnoszą się tylko do środowisk partycjonowanych baz danych.

Lekcja 1. Tworzenie obrazów stanu wyjaśniania

W czasie tej lekcji zostaną utworzone obrazy stanu wyjaśniania. Do przechwytywania informacji dotyczących środowiska, w którym kompilowana jest statyczna lub dynamiczna instrukcja SQL, służy funkcja wyjaśniania SQL. Korzystając z przechwyconych informacji, użytkownik może zrozumieć strukturę i potencjalną wydajność realizacji instrukcji SQL. Obraz stanu wyjaśniania to skompresowane informacje zebrane w czasie wyjaśniania instrukcji SQL. Jest on przechowywany w tabeli EXPLAIN_STATEMENT jako obiekt typu BLOB (binary large object). Zapisane są w nim następujące informacje:

- Wewnętrzna reprezentacja planu dostępu, łącznie z używanymi operatorami, tabelami i indeksami.
- Kryteria decyzyjne wykorzystywane przez optymalizator, w tym statystyki obiektów bazy danych i skumulowane koszty poszczególnych operacji.

Do wyświetlenia wykresu planu dostępu w narzędziu graficznym Visual Explain wymagane są informacje zapisane w obrazie stanu wyjaśniania.

Tworzenie tabel wyjaśniania

Aby utworzyć obrazy stanu wyjaśniania, należy upewnić się, czy dla bieżącego identyfikatora użytkownika istnieją następujące tabele:

- EXPLAIN_INSTANCE
- EXPLAIN_STATEMENT

Aby sprawdzić, czy tabele te istnieją, należy użyć komendy **DB2 list tables**. Jeśli tabele te nie istnieją, trzeba je utworzyć, postępując zgodnie z poniższymi instrukcjami:

1. Jeśli program DB2 nie został jeszcze uruchomiony, wywołaj komendę **db2start**.
2. Z poziomu procesora wiersza komend DB2 połącz się z bazą danych, której chcesz używać. W tej lekcji połącz się z przykładową bazą danych SAMPLE za pomocą komendy **connect to sample**.
3. Utwórz tabele wyjaśniania, korzystając przykładowego pliku komend o nazwie EXPLAIN.DDL. Plik ten znajduje się w katalogu sqllib/misc. Aby uruchomić plik komend, przejdź do tego katalogu i uruchom komendę **db2 -tf EXPLAIN.DDL**. Ten plik komend tworzy tabele wyjaśniania, których nazwy opatrzone są przedrostkiem identycznym z identyfikatorem połączonego użytkownika. Ten identyfikator użytkownika musi mieć uprawnienie CREATETAB w odniesieniu do bazy danych albo uprawnienie SYSADM lub DBADM.

Korzystanie z obrazów stanu wyjaśniania

Aby ułatwić użytkownikowi zrozumienie narzędzia graficznego Visual Explain, dostępne są cztery przykładowe obrazy stanu. Informacje na temat tworzenia przez użytkownika własnych obrazów stanu można znaleźć w poniższych sekcjach, jednak do przejścia tego kursu nie jest konieczne tworzenie nowych obrazów stanu.

- Tworzenie obrazów stanu wyjaśniania dynamicznych instrukcji SQL
- Tworzenie obrazów stanu wyjaśniania statycznych instrukcji SQL

Zapytanie używane dla przykładowych obrazów stanu powoduje wyświetlenie listy nazwisk, działów i dochodów wszystkich pracowników innych niż menedżerowie, których dochody są większe niż 90% najwyższej pensji menedżera.

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

Zapytanie składa się z dwóch części:

1. Podzapytanie (ujęte w nawiasy) zwraca wiersze danych, w których dochód jest większy niż 90% pensji każdego menedżera. Ponieważ podzapytanie jest poprzedzone kwalifikatorem ALL, pobierana jest tylko największa wartość z tej tabeli.
2. Główne zapytanie łączy wszystkie wiersze tabel ORG i STAFF, w których występują identyczne numery działów (DEPTNUM), JOB nie równa się 'Mgr' (kierownik), a pensja powiększona o premię jest większa od wartości zwróconej z podzapytania.

Główne zapytanie zawiera następujące trzy predykaty (porównania):

1. O.DEPTNUMB = S.DEPT
2. S.JOB <> 'Mgr'
3. S.SALARY+S.COMM > ALL (SELECT ST.SALARY*.9
FROM STAFF ST
WHERE ST.JOB='Mgr')

Predykaty te reprezentują odpowiednio:



1. predykat łączenia, który łączy tabele ORG i STAFF tam, gdzie numery działów są identyczne,
2. predykat lokalny odwołujący się do kolumny JOB tabeli STAFF,
3. predykat lokalny odwołujący się do kolumn SALARY oraz COMM tabeli STAFF i wykorzystujący wynik podzapytania.

Aby załadować przykładowe obrazy stanu, wykonaj następujące czynności:

1. Jeśli program DB2 nie został jeszcze uruchomiony, wywołaj komendę **db2start**.
2. Upewnij się, czy w bazie danych istnieją tabele wyjaśniania. W tym celu postępuj według instrukcji podanych w sekcji Tworzenie tabel wyjaśniania.
3. Nawiąż połączenie z bazą danych, która ma być używana. W tym kursie należy nawiązać połączenie z bazą danych SAMPLE. Aby nawiązać połączenie z bazą danych SAMPLE, w wierszu komend programu DB2 wydaj komendę **connect to sample**.

Jeśli nie została ona jeszcze utworzona, zapoznaj się z sekcją dotyczącą instalowania bazy danych SAMPLE w podręczniku *Administration Guide*.

4. Aby zaimportować predefiniowane obrazy stanu, uruchom plik komend programu DB2 o nazwie VESAMPL.DDL.

-  Plik ten znajduje się w katalogu sqllib\samples\ve.
-  Plik ten znajduje się w katalogu sqllib\samples\ve\inter.

Aby uruchomić ten plik komend, przejdź do odpowiedniego katalogu i wydaj komendę **db2 -tf vesampl.ddl**.

- Wspomniany plik komend należy uruchamiać, korzystając z tego samego identyfikatora użytkownika, który był używany do tworzenia tabel wyjaśniania.
- Uruchomienie tego pliku komend spowoduje zaimportowanie tylko predefiniowanych obrazów stanu. Nie spowoduje to utworzenia tabel ani danych. Opisane w dalszej części czynności związane ze strojeniem (na przykład komendy CREATE INDEX i runstats) zostaną wykonane na tabelach i danych z bazy danych SAMPLE.

Można teraz wyświetlić wykresy planów dostępu i korzystać z nich.

Tworzenie obrazów stanu wyjaśniania dynamicznych instrukcji SQL

Uwaga: Zawarte w tej sekcji informacje o tworzeniu obrazu stanu wyjaśniania służą tylko jako materiały referencyjne. Ponieważ dostępne są przykładowe obrazy stany wyjaśniania, do śledzenia tego kursu nie jest konieczne wykonywanie tego zadania.

Aby utworzyć obraz stanu wyjaśniania dynamicznej instrukcji SQL, wykonaj następujące czynności:

1. Jeśli program DB2 nie został jeszcze uruchomiony, wywołaj komendę **db2start**.
2. Upewnij się, czy w bazie danych istnieją tabele wyjaśniania. W tym celu postępuj według instrukcji podanych w sekcji Tworzenie tabel wyjaśniania.
3. Z poziomu procesora wiersza komend DB2 połącz się z bazą danych, której chcesz używać. Na przykład, aby połączyć się z bazą danych SAMPLE, wywołaj komendę **connect to sample**.

Informacje na temat tworzenia bazy danych SAMPLE można znaleźć w sekcji dotyczącej instalowania bazy danych SAMPLE w podręczniku *Administration Guide*.

4. Utwórz obraz stanu wyjaśniania dynamicznej instrukcji SQL, korzystając z jednej z poniższych komend wywoływanych z procesora wiersza komend programu DB2:
 - Aby utworzyć obraz stanu wyjaśniania bez wykonywania instrukcji SQL, wywołaj komendę **set current explain snapshot=explain**.
 - Aby utworzyć obraz stanu wyjaśniania i wykonać instrukcję SQL, wywołaj komendę **set current explain snapshot=yes**.

Ta komenda ustawia specjalny rejestr wyjaśniania. Ustawienie tego rejestru ma wpływ na wszystkie następne instrukcje SQL. Więcej informacji można znaleźć w sekcjach poświęconych bieżącym obrazom stanu wyjaśniania w podręczniku *SQL Reference*.

5. Wprowadź instrukcje SQL w procesorze wiersza komend programu DB2.
6. Aby wyświetlić wykres planu dostępu dla obrazu stanu wyjaśniania, odśwież okno Historia wyjaśnionych instrukcji (dostępne z Centrum sterowania) i kliknij dwukrotnie obraz stanu.
7. Opcjonalnie. Aby wyłączyć tworzenie obrazów stanu wyjaśniania, wywołaj komendę **set current explain snapshot=no** po wprowadzeniu instrukcji SQL.

Tworzenie obrazów stanu wyjaśniania statycznej instrukcji SQL

Uwaga: Zawarte w tej sekcji informacje o tworzeniu obrazu stanu wyjaśniania służą tylko jako materiały referencyjne. Ponieważ dostępne są przykładowe obrazy stany wyjaśniania, do śledzenia tego kursu nie jest konieczne wykonywanie tego zadania.

Aby utworzyć obraz stanu wyjaśniania statycznej instrukcji SQL, wykonaj następujące czynności:

1. Jeśli program DB2 nie został jeszcze uruchomiony, wywołaj komendę **db2start**.
2. Upewnij się, czy w bazie danych istnieją tabele wyjaśniania. W tym celu postępuj według instrukcji podanych w sekcji Tworzenie tabel wyjaśniania.
3. Z poziomu procesora wiersza komend DB2 połącz się z bazą danych, której chcesz używać. Na przykład, aby połączyć się z bazą danych SAMPLE, wywołaj komendę **connect to sample**.
4. Utwórz obraz stanu wyjaśniania statycznej instrukcji SQL, korzystając z opcji EXPLSNAP podczas wykonywania operacji BIND lub PREPARE na aplikacji. Na przykład wydaj komendę **bind plik_uzytkownika explsnap yes**.
5. Opcjonalnie. Aby wyświetlić wykres planu dostępu dla obrazu stanu wyjaśniania, odśwież okno Historia wyjaśnionych instrukcji (dostępne z Centrum sterowania) i kliknij dwukrotnie obraz stanu.

Informacje na temat korzystania z opcji EXPLSNAP w równoważnych funkcjach API można znaleźć w sekcjach poświęconych tym funkcjom, w podręczniku *Application Development Guide*.

Co dalej

W sekcji “Lekcja 2. Wyświetlanie i korzystanie z wykresu planu dostępu” na stronie 7 znajdują się informacje, jak wyświetlić wykres planu dostępu i jak zinterpretować jego zawartość.

Lekcja 2. Wyświetlanie i korzystanie z wykresu planu dostępu

W tej lekcji użyjemy okna Wykres planu dostępu do wyświetlenia i wykorzystania wykresu planu dostępu. Wykres planu dostępu to graficzna reprezentacja planu dostępu. Z jego poziomu można uzyskać dostęp do szczegółowych informacji na temat:

- tabel (i ich skojarzonych kolumn) oraz indeksów
- operatorów (takich jak operacje skanowania tabeli, sortowania i połączenia),
- obszarów tabel i funkcji.

Istnieją następujące metody wyświetlenia wykresu planu dostępu:

- wybranie elementu z listy uprzednio wyjaśnionych instrukcji;
- wybranie elementu z listy znajdujących się w pakiecie instrukcji, które można wyjaśnić;
- dynamiczne wyjaśnienie instrukcji SQL.

Ponieważ będziemy pracować z wykresami planu dostępu związanymi z przykładowymi obrazami stanu wyjaśniania, które zostały załadowane w lekcji 1, wybrane zostaną odpowiednie elementy z listy uprzednio wyjaśnionych instrukcji. Informacje na temat pozostałych metod wyświetlania wykresu planu dostępu można znaleźć w pomocy ekranowej narzędzia Visual Explain.

Wyświetlanie wykresu planu dostępu przez wybranie elementu z listy uprzednio wyjaśnionych instrukcji SQL

Aby wyświetlić wykres planu dostępu przez wybranie elementu z listy uprzednio wyjaśnionych instrukcji, wykonaj następujące czynności:

1. W Centrum sterowania rozwijaj drzewo obiektów aż do znalezienia bazy danych SAMPLE.
2. Kliknij prawym przyciskiem myszy tę bazę danych i z menu podręcznego wybierz opcję **Pokaż historię wyjaśnionych instrukcji**. Zostanie otwarte okno Historia wyjaśnionych instrukcji.
3. Wykres planu dostępu można wyświetlić tylko dla takiej instrukcji, z którą związany jest obraz stanu wyjaśniania. W wypadku instrukcji spełniających to wymaganie w kolumnie **Obraz stanu wyjaśniania** będzie wpisana wartość TAK. Kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 1 (w celu wyświetlenia kolumny **Numer zapytania** konieczne jest przewinięcie w prawo). Zostanie otwarte okno Wykres planu dostępu, związane z określoną instrukcją.

Uwaga: Wykres ten należy odczytywać od dołu do góry. Pierwszy krok zapytania wyświetlany jest w dolnej części wykresu, a ostatni krok znajduje się u góry.

Odczytywanie symboli na wykresie planu dostępu

Wykres planu dostępu przedstawia strukturę planu dostępu w postaci drzewa. *Węzły* drzewa reprezentują:

- Tabele, przedstawione jako prostokąty
- Indeksy, przedstawione jako romby
- Operatory, przedstawione jako ośmiokąty. Operatory TQUEUE, przedstawione jako równoległoboki
- Funkcje tabelowe, przedstawione jako sześciokąty.

W przypadku operatorów, liczby w nawiasach po prawej stronie typu operatora są unikalnymi identyfikatorami poszczególnych węzłów. Liczba pod typem operatora to skumulowany koszt.

Korzystanie z suwaka powiększenia w celu powiększania fragmentów wykresu

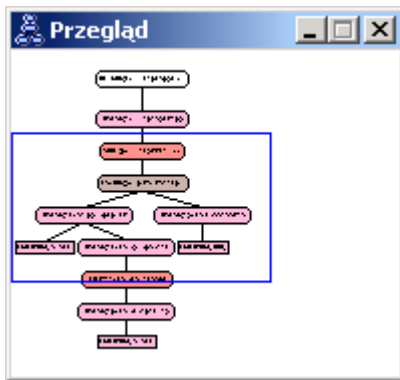
Po wyświetleniu wykresu planu dostępu pokazywany jest cały wykres, a szczegóły odróżniające poszczególne węzły mogą być niewidoczne.

Do powiększania fragmentów wykresu służy **suwak powiększenia** w oknie Wykres planu dostępu:

1. Umieść wskaźnik myszy nad niewielkim polem przewijanym na pasku suwaka powiększenia po lewej stronie wykresu.
2. Kliknij lewym przyciskiem myszy i przeciągaj suwak aż do osiągnięcia odpowiedniego stopnia powiększenia wykresu.

Aby wyświetlić inne fragmenty wykresu, należy skorzystać z pasków przewijania.

Do wyświetlenia obszernego i skomplikowanego wykresu planu dostępu można skorzystać z okna Przegląd wykresu. W tym oknie można określić wyświetloną część wykresu i powiększyć lub przewinąć wykres. Na wykresie planu dostępu widoczna jest sekcja wyświetlona w polu Powiększenie.



Aby przewijać wykres, umieść wskaźnik myszy nad podświetlonym obszarem w oknie Przegląd wykresu, naciśnij i przytrzymaj wciśnięty pierwszy przycisk myszy, a następnie przesunij mysz tak, aby wyświetlona została odpowiednia część wykresu planu dostępu.

Uzyskiwanie bardziej szczegółowych informacji o obiektach na wykresie

Istnieje możliwość uzyskania dodatkowych informacji o występujących na wykresie planu dostępu obiektach. Użytkownik może wyświetlić następujące elementy:

- Statystyki katalogu systemowego dla takich obiektów, jak:
 - tabele, indeksy i funkcje tabelowe;
 - informacje na temat operatorów, takich jak koszt, właściwości i argumenty wejściowe;
 - funkcje wbudowane lub funkcje zdefiniowane przez użytkownika;
 - obszary tabel;
 - kolumny występujące w instrukcji SQL.
- Informacje na temat parametrów konfiguracyjnych i opcji wiązania (parametrów optymalizacji).

Odczytywanie statystyk tabel, indeksów i funkcji tabelowych

Aby wyświetlić statystykę katalogu dla pokazanej na wykresie tabeli (prostokąt), indeksu (romb) lub funkcji tabelowej (sześciokąt), wystarczy kliknąć dwukrotnie odpowiadający danemu elementowi węzeł. Zostanie otwarte związane z wybranym obiektem okno Statystyka, w którym wyświetlone są informacje na temat statystyki obowiązującej w chwili tworzenia określonego obrazu stanu, a także statystyki zapisanej w tabelach katalogu systemowego.

Aby wyświetlić statystykę katalogu dla pokazanej na wykresie *wielu* tabel, indeksów lub funkcji tabelowych, zaznacz każdy element, klikając go (element zostanie podświetlony), a następnie wybierz opcje **Węzeł** → **Pokaż statystyki**. Dla każdego z zaznaczonych obiektów zostanie otwarte okno Statystyka. (Okna te mogą się nawzajem przesłaniać i może zaistnieć potrzeba przeciągnięcia niektórych z nich).

Jeśli w wierszu **STATS_TIME** w kolumnie **Wyjaśnione** znajduje się wpis **Statystyki nie zostały zaktualizowane**, oznacza to, że w czasie tworzenia planu dostępu przez optymalizator nie istniały żadne statystyki. A zatem, jeśli do utworzenia planu dostępu optymalizatorowi potrzebne były określone dane statystyczne, to zostały użyte wartości domyślne. Jeśli optymalizator użył domyślnych danych statystycznych, w kolumnie **Wyjaśnione** są one identyfikowane jako **(domyślne)**.

Uzyskiwanie szczegółowych informacji o operatorach na wykresie

Aby wyświetlić statystyki katalogu dla jednego operatora (ośmiokąt), kliknij dwukrotnie odpowiadający mu węzeł. Zostanie otwarte związane z wybranym operatorem okno **Szczegóły** dotyczące operatora, w którym dostępne są między innymi następujące informacje:

- szacowany skumulowany koszt (we/wy, instrukcje procesora i koszt łączny),
- dotychczasowa liczebność (tj. szacowana liczba przeszukiwanych wierszy),
- informacje o tabelach, które do tej pory zostały użyte i połączone w planie,
- informacje o kolumnach tych tabel, które do tej pory zostały użyte,
- informacje o zastosowanych do tej pory predyktorach, w tym o ich szacowanej selektywności,
- argumenty wejściowe poszczególnych operatorów.

Aby wyświetlić szczegółowe informacje dotyczące *wielu* operatorów, zaznacz odpowiednie operatory przez kliknięcie ich (zostaną podświetlone), a następnie wybierz opcję **Węzeł→Pokaż szczegóły**. Dla każdego z zaznaczonych obiektów zostanie otwarte okno **Statystyka**. (Okna te mogą się nawzajem przesłaniać i może zaistnieć potrzeba przeciągnięcia niektórych z nich).

Uzyskiwanie statystyki dotyczącej funkcji

Aby wyświetlić statystyki katalogu dotyczące funkcji wbudowanych i funkcji zdefiniowanych przez użytkownika, wybierz opcję **Instrukcja→Pokaż statystyki→Funkcje**. Zaznacz jedną lub więcej pozycji na liście wyświetlanej w oknie **Funkcje** i kliknij przycisk **OK**. Dla każdej zaznaczonej funkcji otwarte zostanie okno **Statystyki funkcji**.

Uzyskiwanie statystyki dotyczącej obszarów tabel

Aby wyświetlić statystyki katalogu dotyczące obszarów tabel, wybierz opcję **Instrukcja→Pokaż statystyki→Obszary tabel**. Zaznacz jedną lub więcej pozycji na liście wyświetlanej w oknie **Obszary tabel** i kliknij przycisk **OK**. Dla każdego zaznaczonego obszaru tabel otwarte zostanie okno **Statystyki obszaru tabel**.

Odczytywanie statystyki kolumn występujących w instrukcji SQL

Aby uzyskać statystykę dotyczącą kolumn, występujących w instrukcji SQL, wykonaj następujące czynności:

1. Kliknij dwukrotnie tabelę na wykresie planu dostępu. Zostanie otwarte okno **Statystyki tabel**.
2. Kliknij przycisk **Oдноśne kolumny**. Otwarte zostanie okno **Oдноśne kolumny**, zawierające listę kolumn w tabeli.

3. Zaznacz jedną lub więcej kolumn na liście i kliknij przycisk **OK**. Dla każdej zaznaczonej kolumny otwarte zostanie okno Statystyki odnośnych kolumn.

Uzyskiwanie informacji na temat parametrów konfiguracyjnych i opcji wiązania

Aby wyświetlić informacje dotyczące parametrów konfiguracyjnych i opcji wiązania (parametrów optymalizacji), w oknie Wykres planu dostępu wybierz opcje **Instrukcja**→**Pokaż parametry optymalizacji**. Zostanie otwarte okno Parametry optymalizacji zawierające informacje o wartościach parametrów obowiązujących w czasie tworzenia obrazu stanu oraz o wartościach bieżących.

Zmiana wyglądu wykresu

Aby zmodyfikować różne opcje wyświetlania wykresu, wykonaj następujące czynności:

1. W oknie Wykres planu dostępu wybierz opcje **Widok**→**Ustawienia**. Zostanie otwarty notatnik Ustawienia wykresu planu dostępu.
2. Aby zmienić kolor tła, wybierz kartę Wykres.
3. Aby zmodyfikować kolory różnych operatorów, skorzystaj z kart Podstawowe, Rozszerzone, Aktualizacja i Różne.
4. Aby zmienić kolor węzłów tabel, indeksów i funkcji, wybierz kartę Operand.
5. Aby określić typ informacji wyświetlanych w węzłach operatorów (typ kosztu lub liczności, czyli szacowanej liczby wierszy zwróconych do tej pory), wybierz kartę Operator.
6. Aby określić, czy w węzłach tabeli mają być wyświetlone nazwy schematów, czy identyfikatory użytkowników, wybierz kartę Operand.
7. Aby zdecydować, czy węzły mają być wyświetlane w postaci dwuwymiarowej, czy trójwymiarowej, wybierz kartę Węzeł.
8. Aby zaktualizować wykres zgodnie z wybranymi opcjami i zapisać ustawienia, kliknij przycisk **Zastosuj**.

Co dalej

Jeśli stosowane jest środowisko bazy danych z pojedynczą partycją, należy przejść do sekcji “Lekcja 3. Udoskonalanie planu dostępu w środowisku bazy danych z pojedynczą partycją” na stronie 13, w której zawarto informacje o wpływie różnych działań związanych ze strojeniem na plan dostępu.

Jeśli używane jest środowisko partycjonowanej bazy danych, należy przejść do sekcji “Uruchamianie zapytania bez indeksów i bez statystyk” na stronie 14, w której zawarto informacje o wpływie różnych działań związanych ze strojeniem na plan dostępu.

Lekcja 3. Udoskonalanie planu dostępu w środowisku bazy danych z pojedynczą partycją

W tej lekcji zapoznamy użytkownika z wpływem różnorodnych działań związanych ze strojeniem na plan dostępu i okna dotyczące zapytania podstawowego. Na szeregu przykładach i ilustracjach użytkownik nauczy się, w jaki sposób za pomocą komendy **runstats** i odpowiednich indeksów, można zmniejszyć szacowany łączny koszt planu dostępu, nawet dla stosunkowo prostego zapytania.

W miarę nabywania doświadczenia w pracy z narzędziem graficznym Visual Explain, użytkownik odkryje inne sposoby optymalizacji zapytań.

Praca z wykresami planów dostępu

Korzystając z czterech przykładowych obrazów stanu wyjaśniania, pokażemy, jak strojenie wpływa na poprawę wydajności bazy danych.

Z tymi obrazami stanu wyjaśniania związane są zapytania o numerach od 1 do 4. W każdym zapytaniu występuje ta sama instrukcja SQL (opisana w Lekcji 1):

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

W kolejnych iteracjach zapytania stosowanych jest coraz więcej technik strojenia. Na przykład w Zapytaniu 1 nie zastosowano dostrajania wydajności, a w Zapytaniu 4 wykorzystano najwięcej możliwości dostrajania. Różnice między poszczególnymi zapytaniami opisane są poniżej:

Zapytanie nr 1

Uruchamianie zapytania bez indeksów i bez statystyk

Zapytanie nr 2

Gromadzenie bieżących statystyk tabel i indeksów w zapytaniu

Zapytanie nr 3

Tworzenie indeksów dla kolumn używanych do łączenia tabel w zapytaniu

Zapytanie nr 4

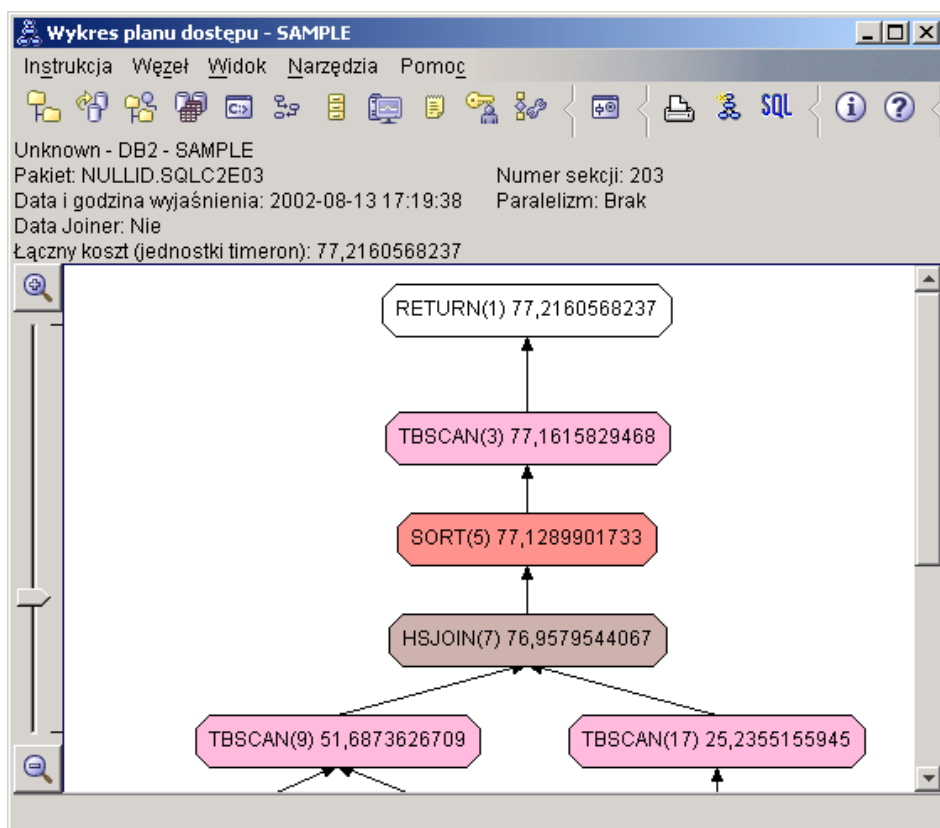
Tworzenie dodatkowych indeksów dla kolumn tabeli

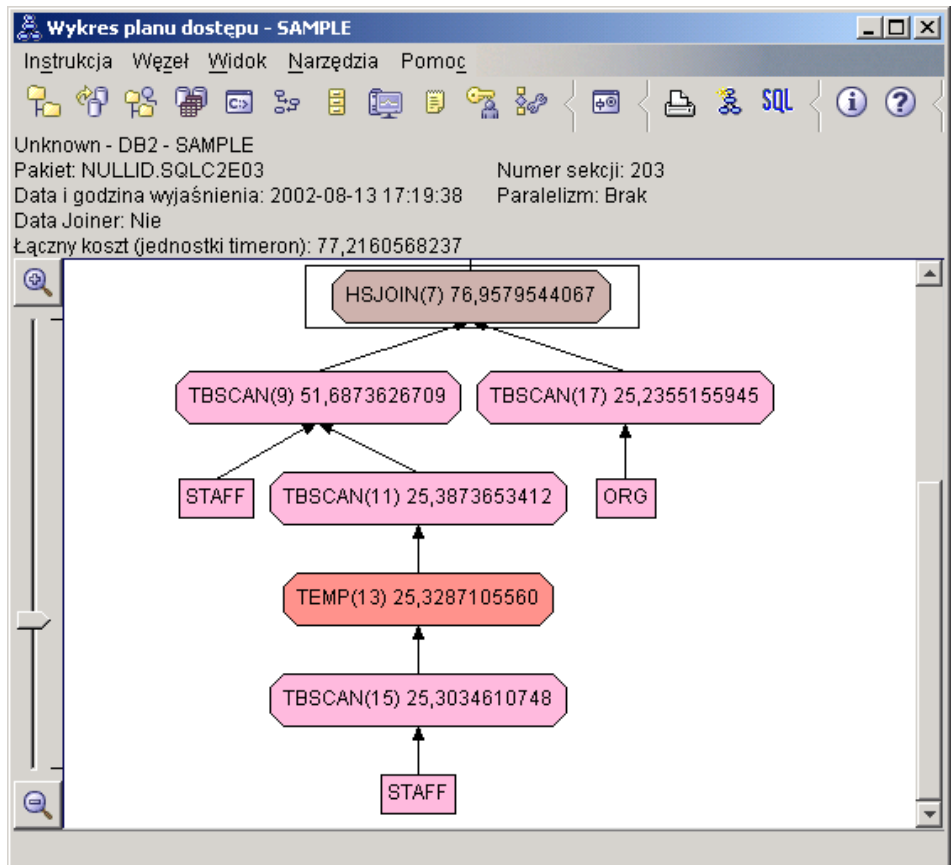
Uruchamianie zapytania bez indeksów i bez statystyk

W tym przykładzie plan dostępu dla zapytania SQL został utworzony bez korzystania z indeksów i statystyki.

Aby wyświetlić wykres planu dostępu dla Zapytania 1, wykonaj następujące czynności:

1. W Centrum sterowania rozwijaj drzewo obiektów aż do znalezienia bazy danych SAMPLE.
2. Kliknij prawym przyciskiem myszy tę bazę danych i z menu podręcznego wybierz opcję **Pokaż historię wyjaśnionych instrukcji**. Zostanie otwarte okno Historia wyjaśnionych instrukcji.
3. Kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 1 (w celu wyświetlenia kolumny **Numer zapytania** konieczne jest przewinięcie w prawo). Zostanie otwarte okno Wykres planu dostępu, związane z określoną instrukcją.





Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

1. Czy dla każdej tabeli występującej w zapytaniu istnieje aktualna statystyka?

Aby sprawdzić, czy dla każdej tabeli występującej w zapytaniu istnieje aktualna statystyka, na wykresie planu dostępu kliknij dwukrotnie każdy węzeł tabeli. Jeśli w czasie tworzenia tego obrazu stanu nie została zgromadzona żadna statystyka, w oknie Statystyki tabel, które zostanie otwarte, w wierszu, **STATS_TIME** w kolumnie **Wyjaśnione**, ukaże się tekst "Statystyki nie zostały zaktualizowane".

Jeśli bieżące statystyki nie istnieją, optymalizator używa statystyk domyślnych, które mogą różnić się od rzeczywistych. Domyślne statystyki są oznaczone słowem "domyślne" w kolumnie **Wyjaśnione** w oknie Statystyki tabel.

Zgodnie z podanymi w oknie Statystyki tabel informacjami na temat tabeli **ORG**, optymalizator użył statystyk domyślnych (tak, jak to zaznaczono obok wyjaśnionych wartości). Użyto statystyk domyślnych, ponieważ w czasie tworzenia obrazu stanu nie były dostępne statystyki rzeczywiste (tak, jak to zaznaczono w wierszu **STATS_TIME**).

Statystyka	Wyjaśnione	Bieżące
CREATE_TIME	2002-08-13 17:19:27	
STATS_TIME	Statystyki nie zostały zaktualizowane	Statystyki nie zostały zaktualizowane
CARD	60(domyślne)	
NPAGES	1(domyślne)	
FPAGES	1(domyślne)	
COLCOUNT	5(domyślne)	
OVERFLOW	0(domyślne)	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
ULOTNE	Nie(domyślne)	Nie

2. Czy w tym planie dostępu zastosowane są najbardziej efektywne metody dostępu do danych?

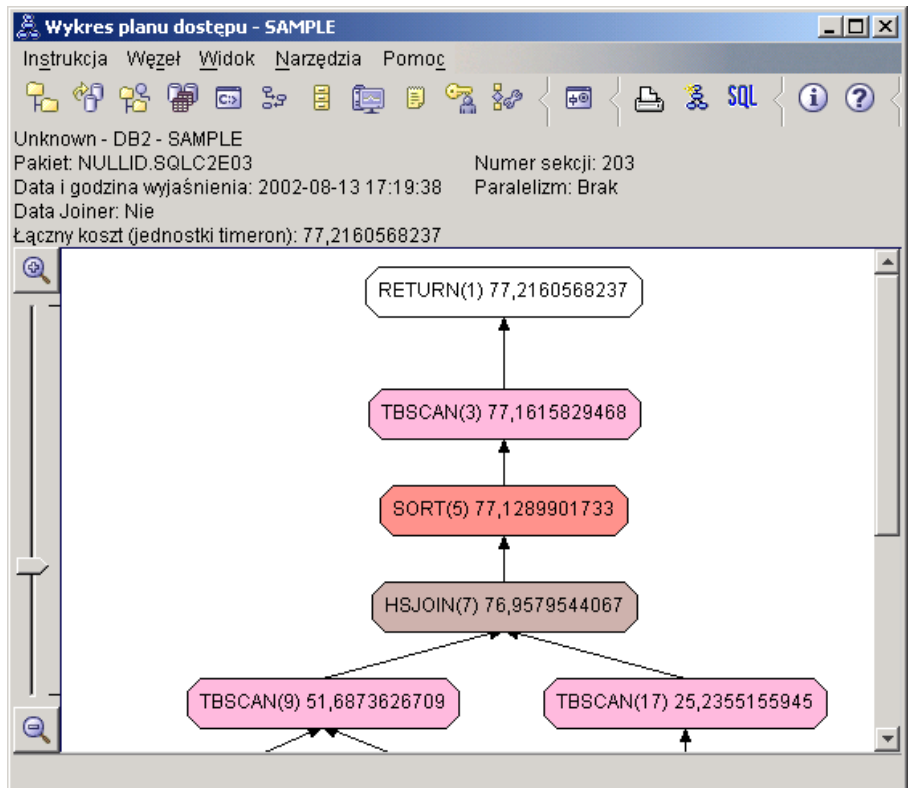
Ten plan dostępu zawiera operatory skanowania tabel, a nie indeksów. Operatory skanowania tabel są przedstawione w postaci ośmiokątów i opatrzone etykietą TBSCAN. Gdyby zostały użyte operatory skanowania indeksów, przedstawione byłyby w postaci rombów i opatrzone etykietą IXSCAN. W wypadku wyodrębniania niewielkich ilości danych użycie utworzonego dla tabeli indeksu jest bardziej ekonomiczne niż skanowanie tabeli.

3. Jak efektywny jest ten plan dostępu?

Efektywność planu dostępu można określić pod warunkiem, że jest on oparty na rzeczywistych danych statystycznych. Ponieważ w planie dostępu optymalizator wykorzystał statystyki domyślne, określenie efektywności planu dostępu nie jest możliwe.

Z reguły należy zanotować szacowany łączny koszt planu dostępu, który będzie można porównać z kosztami zmodyfikowanych planów. Koszt podany przy każdym węźle jest skumulowany i obejmuje wszystkie czynności od początku przetwarzania zapytania do danego węzła włącznie.

Podany w oknie Wykres planu dostępu łączny koszt wynosi około 77 jednostek timeron. Wartość ta wyświetlana jest w węźle **RETURN (1)** u góry wykresu. Szacowany łączny koszt jest także wyświetlany w górnym obszarze okna.



4. Co dalej?

W przykładzie z Zapytaniem 2 przyjrzymy się planowi dostępu podstawowego zapytania po uruchomieniu komendy **runstats**. Komenda **runstats** dostarcza optymalizatorowi aktualną statystykę dotyczącą wszystkich tabel występujących w zapytaniu.

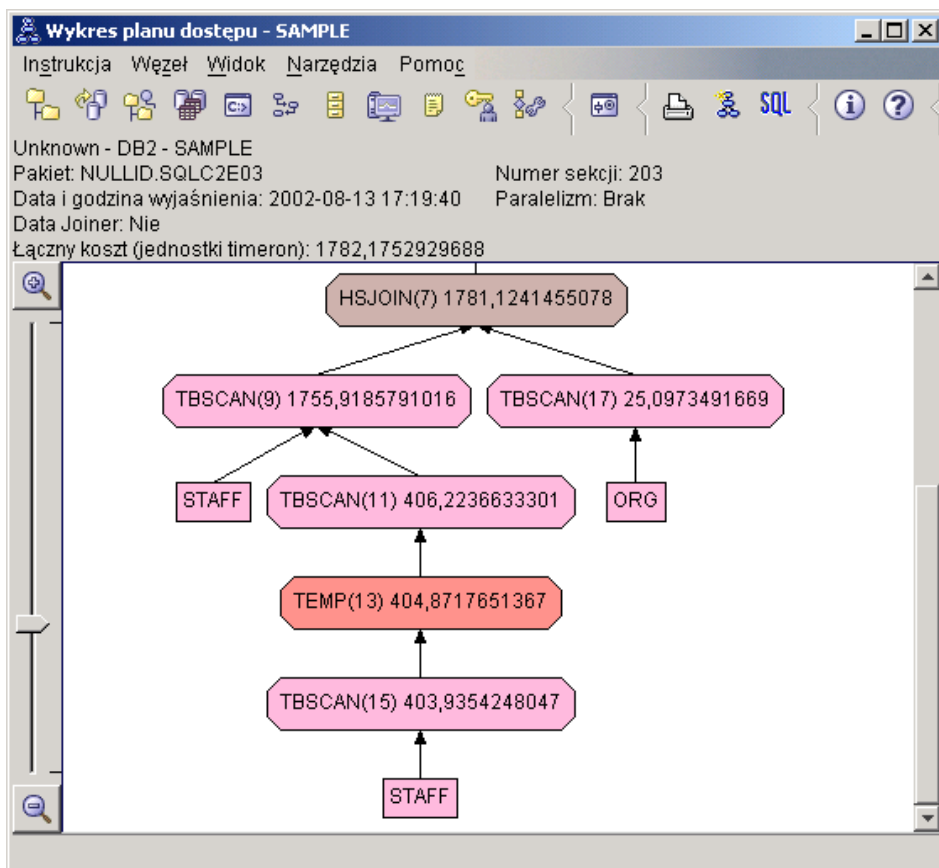
Gromadzenie aktualnej statystyki tabel i indeksów przy użyciu komendy **runstats**

Punktem wyjścia dla tego przykładu jest plan dostępu dla Zapytania 1. Plan ten zostanie zmodyfikowany przez zgromadzenie aktualnej statystyki za pomocą komendy **runstats**.

Zdecydowanie zaleca się użycie komendy **runstats** w celu zgromadzenia bieżących informacji statystycznych na temat tabel i indeksów, zwłaszcza jeśli od czasu ostatniego wykonania komendy **runstats** miały miejsce znaczące aktualizacje lub utworzono nowe indeksy. Spowoduje to przekazanie optymalizatorowi najdokładniejszych informacji, umożliwiających określenie najlepszego planu dostępu. Jeśli nie jest dostępna aktualna statystyka, optymalizator może wybrać nieefektywny plan dostępu oparty na niedokładnej statystyce domyślnej.

Należy koniecznie używać komendy **runstats** po każdej aktualizacji tabeli; w przeciwnym razie optymalizator może potraktować tabelę tak, jak gdyby była pusta. Problem ten jest wyraźnie widoczny, jeśli licznosc na stronie Szczegóły dotyczące operatora wynosi zero. W takim wypadku należy zaktualizować table, uruchomić komendę **runstats** i ponownie utworzyć obrazy stanu wyjaśniania zmodyfikowanych tabel.

Aby wyświetlić wykres planu dostępu dla Zapytania 2: w oknie Historia wyjaśnionych instrukcji kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 2. Zostanie otwarte związane z tą instrukcją okno Wykres planu dostępu.



Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

1. Czy dla każdej tabeli występującej w zapytaniu istnieje aktualna statystyka?
Z informacji przedstawionych w oknie Statystyki tabel tabeli ORG wynika, że optymalizator użył rzeczywistych danych statystycznych (wartość **STATS_TIME**

określa godzinę, o której zgromadzono te dane). Dokładność danych statystycznych zależy od tego, czy od czasu uruchomienia komendy **runstats** zawartość tabel uległa istotnym zmianom.

The screenshot shows a window titled "Statystyka tabeli - ORG" with the following information:

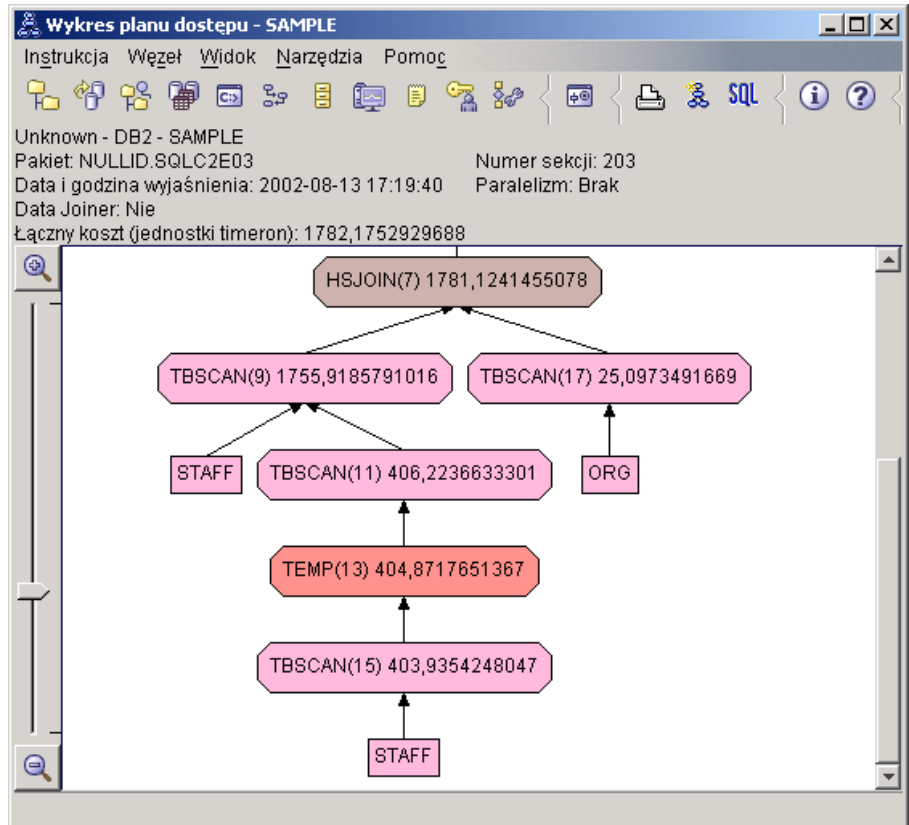
- Unknown - DB2 - SAMPLE
- Tabela: CKMWONG.ORG
- Data i godzina wyjaśnienia: 2002-08-13 17:19:40
- Bieżąca data i godzina: 2002-08-21 11:44:38

Statystyka	Wyjaśnione	Bieżące
CREATE_TIME	2002-08-13 17:19:27	
STATS_TIME	2002-08-13 17:19:40	Statystyki nie zostały zaktualizowane
CARD	8	
NPAGES	1	
FPAGES	1	
COLCOUNT	5	
OVERFLOW	0	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
ULOTNE	Nie	Nie

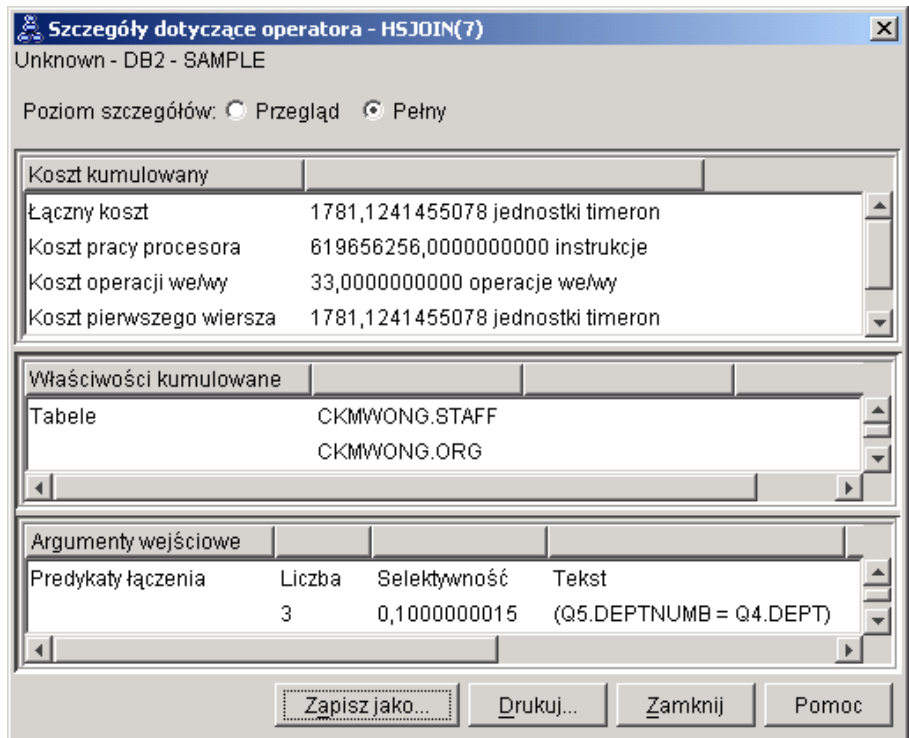
Buttons at the bottom: Odnośne kolumny, Grupy kolumn, Indeksy, Zapisz jako..., Drukuj..., Zamknij, Pomoc

2. Czy w tym planie dostępu zastosowane są najbardziej efektywne metody dostępu do danych?

Podobnie jak w przypadku Zapytania nr 1, także w planie dostępu Zapytania nr 2 występują operatory skanowania tabel (TBSCAN), a nie skanowania indeksów (IXSCAN). Mimo że istnieją aktualne statystyki, nie było stosowane skanowanie indeksów, ponieważ brak jest indeksów dla kolumn używanych w zapytaniu. Jednym ze sposobów optymalizacji zapytania może być utworzenie indeksów dla kolumn używanych do łączenia tabel (tj. kolumn występujących w predykatkach łączenia). W tym przykładzie jest to pierwszy operator łączenia scalającego ze skanowaniem: HSJOIN (7).



W oknie Szczegóły operatora dla operatora HSJOIN (7) należy popatrzeć na sekcję **Predykaty łączenia** w grupie **Argumenty wejściowe**. W kolumnie **Tekst** wymienione są kolumny używane w tej operacji łączenia. W omawianym przykładzie są to kolumny DEPTNUMB i DEPT.



3. Jak efektywny jest ten plan dostępu?

Plany dostępu oparte na aktualnych statystykach zawsze stanowią podstawę do realistycznego oszacowania kosztu (mierzonego w jednostkach timeron). Ponieważ w wypadku Zapytania 1 szacowany koszt był oparty na statystykach domyślnych, nie możemy porównywać ze sobą kosztów tych dwóch planów dostępu. To, czy koszt jest większy, czy mniejszy, nie ma znaczenia. Aby uzyskać poprawną miarę efektywności, trzeba porównać koszty planów dostępu opartych na rzeczywistych danych statystycznych.

4. Co dalej?

W przykładzie z Zapytaniem 3 przyjrzymy się skutkom dodania indeksów zbudowanych na kolumnach DEPTNUMB i DEPT. Dodanie indeksów zbudowanych na kolumnach używanych w predykatkach łączenia może poprawić wydajność.

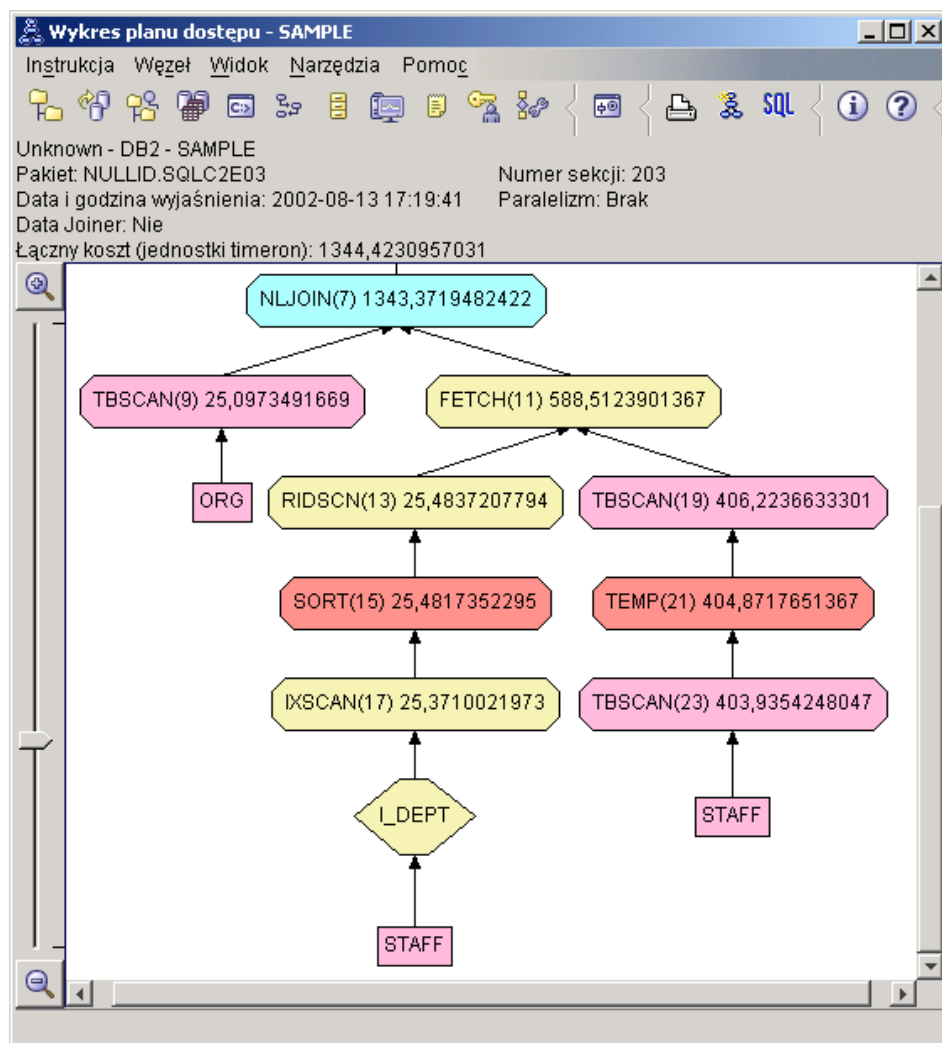
Tworzenie indeksów dla kolumn używanych do łączenia tabel w zapytaniu

Punktem wyjścia dla tego przykładu jest plan dostępu dla Zapytania 2. Plan ten zostanie zmodyfikowany poprzez utworzenie indeksów na kolumnie DEPT tabeli STAFF i na kolumnie DEPTNUMB tabeli ORG.

Uwaga: W wersji 8 zalecane indeksy mogą zostać utworzone za pomocą Kreatora optymalizacji wydajności dla obciążenia.

Aby wyświetlić wykres planu dostępu dla Zapytania 3: w oknie Historia wyjaśnionych instrukcji kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 3. Zostanie otwarte związane z tą instrukcją okno Wykres planu dostępu.

Uwaga: Nawet gdyby dla kolumny DEPTNUM został utworzony indeks, optymalizator nie użyłby go.

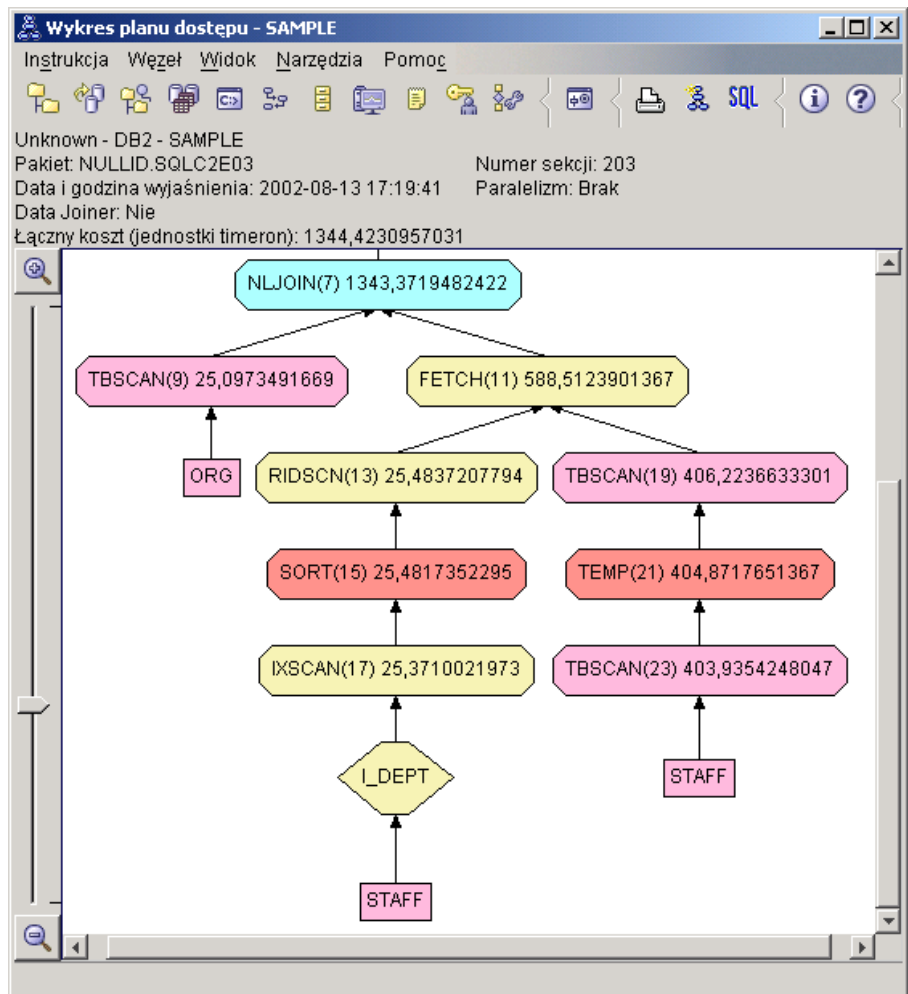


Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

1. Co zmieniło się w planie dostępu w wyniku utworzenia indeksów?

Operator łączenia scalającego ze skanowaniem HSJOIN (7), który używany był w zapytaniu 2, został zastąpiony przez operator łączenia zagnieżdżonego NLJOIN (7). Szacowany koszt zastosowania łączenia zagnieżdżonego jest mniejszy od szacowanego kosztu zastosowania łączenia scalającego ze skanowaniem, ponieważ ten typ łączenia nie wymaga tabel tymczasowych ani pomocniczych tabel sortowania.

Ponad tabelą STAFF został dodany nowy węzeł w kształcie rombu, **I_DEPT**. Węzeł ten reprezentuje indeks utworzony na kolumnie DEPT. Pokazuje on, że w celu określenia, które wiersze mają zostać pobrane, optymalizator zastosował skanowanie indeksu, a nie skanowanie tabeli.



Należy zauważyć, że w tej części wykresu planu dostępu na kolumnie DEPT został utworzony nowy indeks (I_DEPT), a do dostępu do tabeli STAFF został użyty

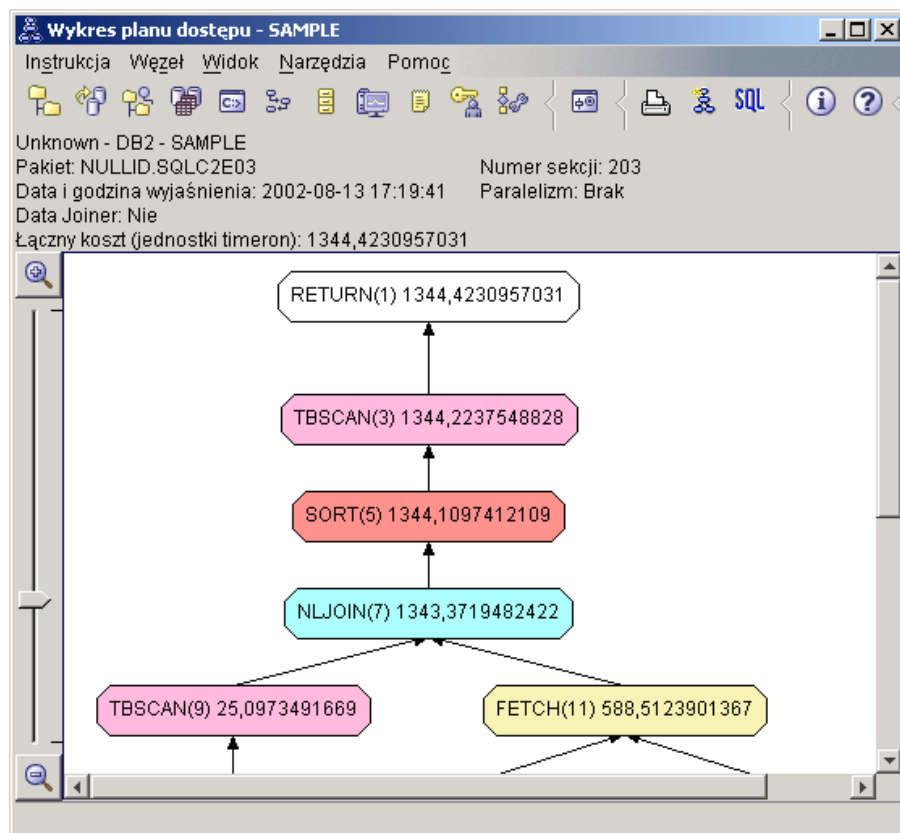
operator skanowania IXSCAN (17). W Zapytaniu nr 2 dostęp do tej tabeli był realizowany z zastosowaniem skanowania tabeli.

2. Czy w tym planie dostępu zastosowane są najbardziej efektywne metody dostępu do danych?

W wyniku dodania indeksów dostęp do tabeli STAFF realizowany został przy użyciu węzła typu IXSCAN, IXSCAN (17). W Zapytaniu nr 2 indeksy nie istniały, a zatem używano w nim operatora skanowania tabeli.

Węzeł typu FETCH, oznaczony FETCH (11), pokazuje, że optymalizator zastosował indeks nie tylko do pobrania kolumny DEPT, lecz także jako wskaźnik przy pobieraniu dodatkowych kolumn z tabeli STAFF. W tym wypadku obliczenia wykazały, że użycie skanowania indeksu i operatora pobierania jest mniej kosztowne, niż używana we wcześniejszych planach dostępu operacja pełnego skanowania tabeli.

Uwaga: Węzeł tabeli STAFF wyświetlony jest dwukrotnie w celu zilustrowania zarówno jego powiązania z indeksem zbudowanym na kolumnie DEPT, jak i z operatorem FETCH.



Plan dostępu tego zapytania ilustruje efekt utworzenia indeksów dla kolumn występujących w predykatkach łączenia. Indeksy mogą także przyspieszyć zastosowanie predykatów lokalnych. Przyjrzyjmy się predykatom lokalnym dla każdej tabeli w zapytaniu, aby sprawdzić, jak dodanie indeksów dla występujących w nich kolumn mogło wpłynąć na plan dostępu.

W oknie Szczegóły operatora dla operatora FETCH (11) należy popatrzeć na kolumny w obszarze **Właściwości kumulowane**. Kolumną użytą w predykatce dla tej operacji pobierania jest JOB, tak jak to uwidoczniło w sekcji Predykaty.

Uwaga: Selektywność tego predykatu wynosi 0,66. Oznacza to, że za pomocą tego predykatu do dalszego przetwarzania zostanie wybranych 66% wierszy.

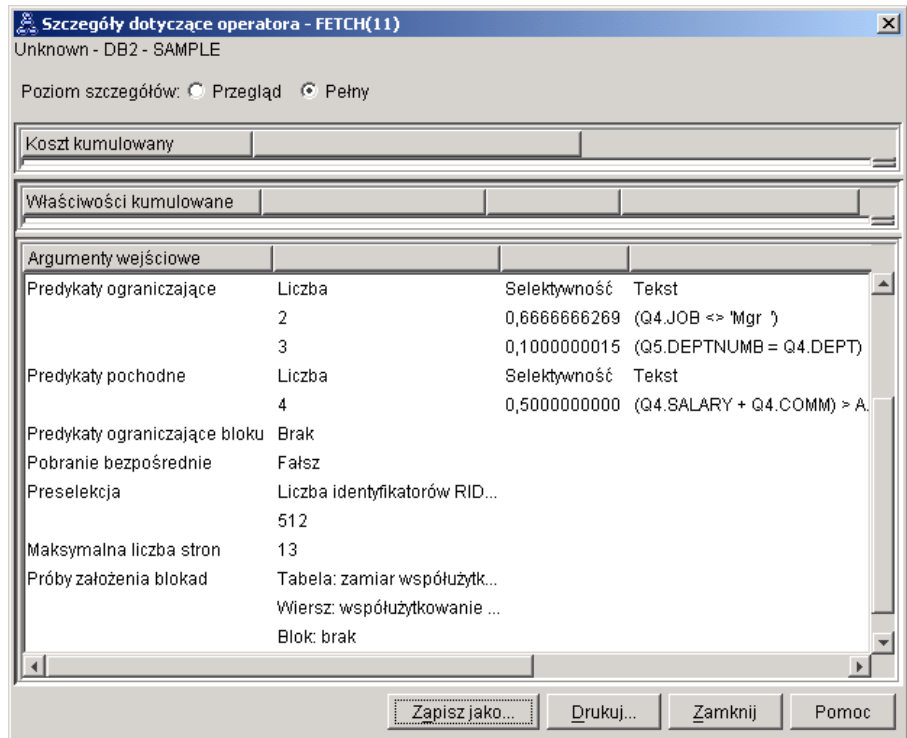
Szczegóły dotyczące operatora - FETCH(11)
Unknown - DB2 - SAMPLE

Poziom szczegółów: Przegląd Pełny

Koszt kumulowany	
Łączny koszt	588,5123901367 jednostki timeron
Koszt pracy procesora	65430928,0000000000 instrukcje
Koszt operacji we/wy	32,9478225708 operacje we/wy
Koszt pierwszego wiersza	459,1545715332 jednostki timeron

Właściwości kumulowane			
Tabele	CKMWONG.STAFF		
Kolumny	CKMWONG.STAFF.NAME CKMWONG.STAFF.ID CKMWONG.STAFF.COMM CKMWONG.STAFF.SALARY		
Kolumny sortujące	Brak		
Predykaty	Liczba	Selektywność	Tekst
	2	0,6666666269	(Q4.JOB <=> 'Mgr ')
	3	0,1000000015	(Q5.DEPTNUMB = Q4.DEPT)
	4	0,5000000000	(Q4.SALARY + Q4.COMM) > ...
Liczność	36,1666641235		
Łączna liczba użytych stron ...	20,1367607117		

Zapisz jako... Drukuj... Zamknij Pomoc



W oknie Szczegóły operatora dla operatora FETCH (11) wyświetlone są używane w tej operacji kolumny. Jak widać, w pierwszym wierszu, obok etykiety **Pobrane kolumny** pod nagłówkiem **Argumenty wejściowe**, wymieniona została kolumna DEPTNAME.

3. Jak efektywny jest ten plan dostępu?

Ten plan dostępu jest bardziej ekonomiczny niż plan z poprzedniego przykładu. Koszt kumulowany został zmniejszony z około 1,781 jednostek timeron w Zapytaniu 2 do około 982 jednostek timeron w Zapytaniu 3.

Jednak w planie dostępu dla Zapytania 3 występuje operator skanowania indeksu IXSCAN (17) i operator FETCH (11), które działają na tabeli STAFF. Wprawdzie skanowanie indeksu stosowane razem z operacją pobierania jest mniej kosztowne niż pełne skanowanie tabeli, to jednak wymaga jednego dostępu do tabeli i jednego dostępu do indeksu dla każdego pobieranego wiersza. Spróbujemy wyeliminować ten podwójny dostęp do tabeli STAFF.

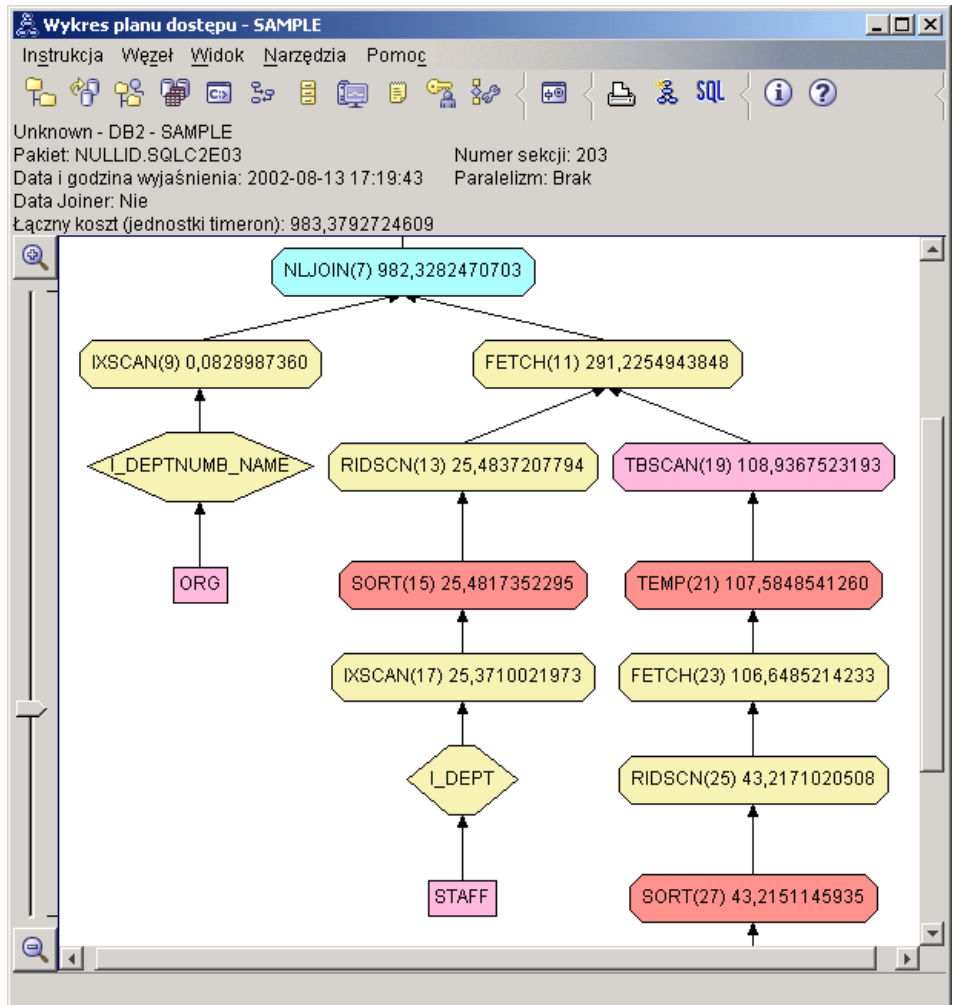
4. Co dalej?

W Zapytaniu 4 operator pobierania i skanowania indeksu został zastąpiony operatorem skanowania indeksu bez pobierania. Utworzenie dodatkowych indeksów może zmniejszyć szacowany koszt planu dostępu.

Tworzenie dodatkowych indeksów dla kolumn tabeli

Punktem wyjścia dla tego przykładu jest plan dostępu dla Zapytania nr 3. Plan ten zostanie zmodyfikowany w wyniku utworzenia indeksu na kolumnie JOB w tabeli STAFF i dodania kolumny DEPTNAME do istniejącego indeksu w tabeli ORG. (Dodanie oddzielnego indeksu mogłoby spowodować konieczność wykonania dodatkowych operacji dostępu).

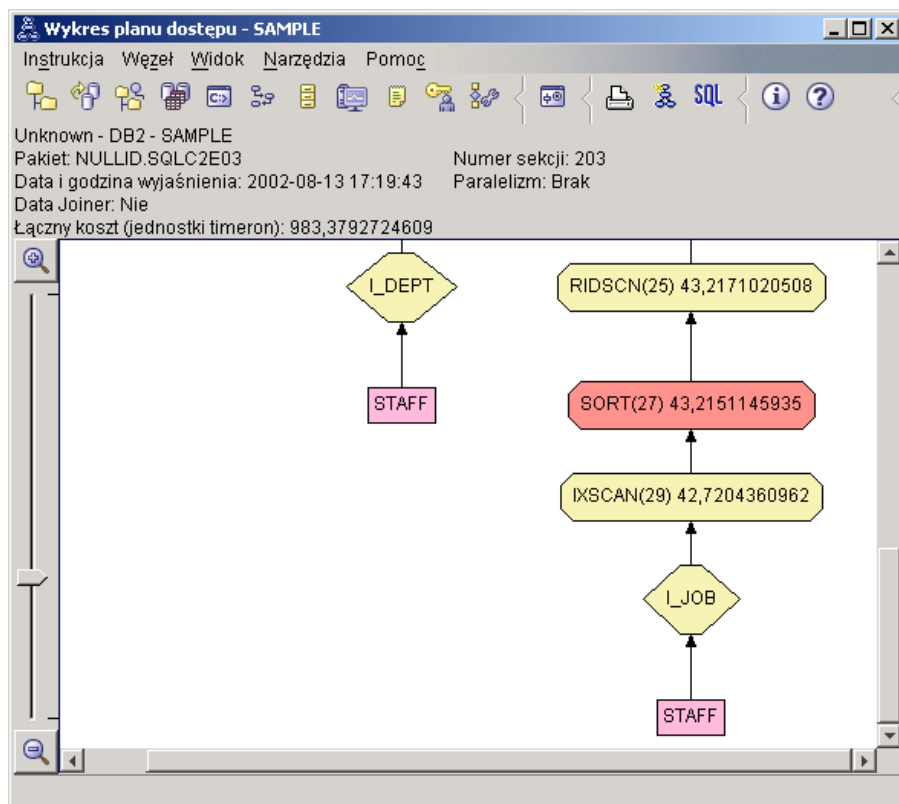
Aby wyświetlić wykres planu dostępu dla Zapytania 4: w oknie Historia wyjaśnionych instrukcji kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 4. Zostanie otwarte związane z tą instrukcją okno Wykres planu dostępu.



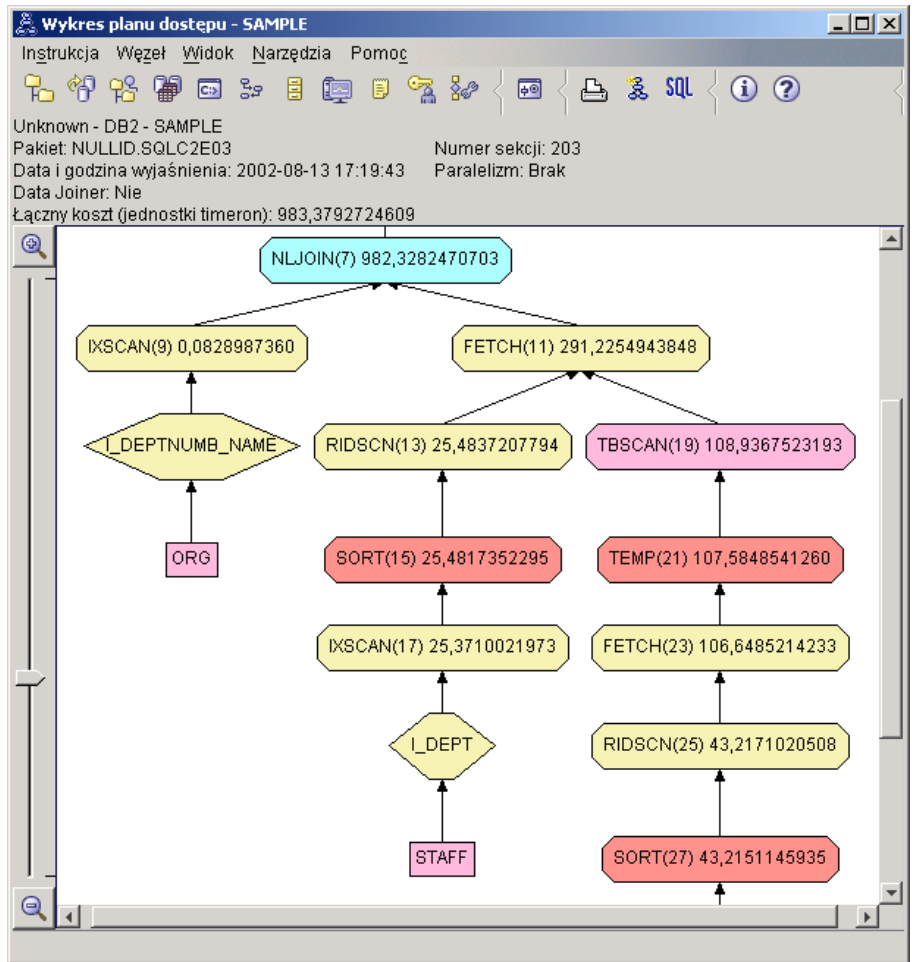
Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

1. Co zmieniło się w tym planie dostępu w wyniku utworzenia dodatkowych indeksów?

Optymalizator wykorzystał indeks utworzony dla kolumny JOB w tabeli STAFF (reprezentowany przez romb z etykietą **L_JOB**) w celu ulepszenia tego planu dostępu.



Zwróćmy uwagę, że w środkowej części wykresu planu dostępu dotychczasowe operatory skanowania indeksu i pobierania dotyczące tabeli ORG zostały zastąpione samym operatorem skanowania indeksu IXSCAN (9). Dodanie kolumny DEPTNAME do indeksu dla tabeli ORG umożliwiło optymalizatorowi wyeliminowanie dodatkowych operacji dostępu związanych z pobieraniem.



2. Jak efektywny jest ten plan dostępu?

Ten plan dostępu jest bardziej ekonomiczny niż plan z poprzedniego przykładu. Koszt kumulowany został zmniejszony z około 1,343 jednostek timeron w Zapytaniu 3 do około 982 jednostek timeron w Zapytaniu 4.

Co dalej

Szczegółowe informacje na temat innych działań, które można wykonać w celu poprawienia wydajności, dostępne są w książce *Administration Guide*. Po zapoznaniu się z tymi informacjami można powrócić do narzędzia Visual Explain, aby ocenić wpływ tych działań na wydajność.

Lekcja 4. Udoskonalanie planu dostępu w środowisku partycjonowanej bazy danych

W tej lekcji zapoznamy użytkownika z wpływem różnorodnych działań związanych ze strojeniem na plan dostępu i okna dotyczące zapytania podstawowego. Na szeregu przykładach i ilustracjach użytkownik nauczy się, w jaki sposób za pomocą komendy **runstats** i odpowiednich indeksów, można zmniejszyć szacowany łączny koszt planu dostępu, nawet dla stosunkowo prostego zapytania.

W miarę nabywania doświadczenia w pracy z narzędziem graficznym Visual Explain, użytkownik odkryje inne sposoby optymalizacji zapytań.

Praca z wykresami planów dostępu

Korzystając z czterech przykładowych obrazów stanu wyjaśniania, pokażemy, jak strojenie wpływa na poprawę wydajności bazy danych.

Z tymi obrazami stanu wyjaśniania związane są zapytania o numerach od 1 do 4. W każdym zapytaniu występuje ta sama instrukcja SQL (opisana w Lekcji 1):

```
SELECT S.ID,S.NAME,O.DEPTNAME,SALARY+COMM
FROM ORG O, STAFF S
WHERE
  O.DEPTNUMB = S.DEPT AND
  S.JOB <> 'Mgr' AND
  S.SALARY+S.COMM > ALL( SELECT ST.SALARY*.9
                        FROM STAFF ST
                        WHERE ST.JOB='Mgr' )
ORDER BY S.NAME
```

W kolejnych iteracjach zapytania stosowanych jest coraz więcej technik strojenia. Na przykład w Zapytaniu 1 nie zastosowano dostrajania wydajności, a w Zapytaniu 4 wykorzystano najwięcej możliwości dostrajania. Różnice między poszczególnymi zapytaniami opisane są poniżej:

Zapytanie nr 1

Uruchamianie zapytania bez indeksów i bez statystyk

Zapytanie nr 2

Gromadzenie bieżących statystyk tabel i indeksów w zapytaniu

Zapytanie nr 3

Tworzenie indeksów dla kolumn używanych do łączenia tabel w zapytaniu

Zapytanie nr 4

Tworzenie dodatkowych indeksów dla kolumn tabeli

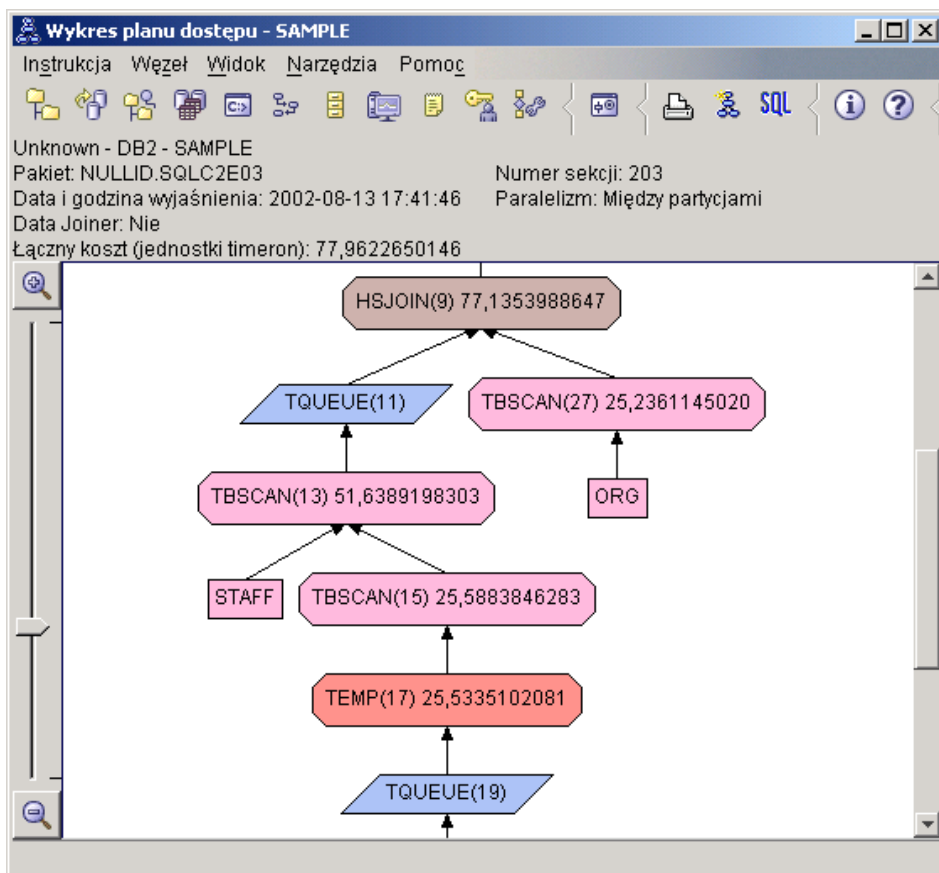
Przykłady te zostały utworzone na komputerze RS/6000 SP z 7 fizycznymi węzłami, z zastosowaniem paralelizmu między partycjami.

Uruchamianie zapytania bez indeksów i bez statystyk

W tym przykładzie plan dostępu dla zapytania SQL został utworzony bez korzystania z indeksów i statystyki.

Aby wyświetlić wykres planu dostępu dla Zapytania 1, wykonaj następujące czynności:

1. W Centrum sterowania rozwijaj drzewo obiektów aż do znalezienia bazy danych SAMPLE.
2. Kliknij prawym przyciskiem myszy tę bazę danych i z menu podręcznego wybierz opcję **Pokaż historię wyjaśnionych instrukcji**. Zostanie otwarte okno Historia wyjaśnionych instrukcji.
3. Kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 1 (w celu wyświetlenia kolumny **Numer zapytania** konieczne może być przewinięcie widoku w prawo). Zostanie otwarte okno Wykres planu dostępu, związane z określoną instrukcją.



Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

1. Czy dla każdej tabeli występującej w zapytaniu istnieje aktualna statystyka?

Aby sprawdzić, czy dla poszczególnych tabel występujących w zapytaniu istnieją bieżące statystyki, należy kliknąć dwukrotnie każdy z węzłów tabeli na wykresie planu dostępu. Jeśli w czasie tworzenia tego obrazu stanu nie została zgromadzona żadna statystyka, w oknie Statystyki tabel, które zostanie otwarte, w wierszu **STATS_TIME**, w kolumnie **Wyjaśnione**, ukaże się tekst "Statystyki nie zostały zaktualizowane".

Jeśli bieżące statystyki nie istnieją, optymalizator używa statystyk domyślnych, które mogą różnić się od rzeczywistych. Domyślne statystyki są oznaczone słowem "domyślne" w kolumnie **Wyjaśnione** w oknie Statystyki tabel.

Zgodnie z podanymi w oknie Statystyki tabel informacjami na temat tabeli ORG, optymalizator użył statystyk domyślnych (tak, jak to zaznaczono obok wyjaśnionych wartości). Użyto statystyk domyślnych, ponieważ w czasie tworzenia obrazu stanu nie były dostępne statystyki rzeczywiste (tak, jak to zaznaczono w wierszu **STATS_TIME**).

Statystyka	Wyjaśnione	Bieżące
CREATE_TIME	2002-08-13 17:40:05	
STATS_TIME	Statystyki nie zostały zaktualizowane	Statystyki nie zostały zaktualizowane
CARD	60(domyślne)	
NPAGES	1(domyślne)	
FPAGES	1(domyślne)	
COLCOUNT	5(domyślne)	
OVERFLOW	0(domyślne)	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
ULOTNE	Nie(domyślne)	Nie

2. Czy w tym planie dostępu zastosowane są najbardziej efektywne metody dostępu do danych?

Ten plan dostępu zawiera operatory skanowania tabel, a nie indeksów. Operatory skanowania tabel są przedstawione w postaci ośmiokątów i opatrzone etykietą TBSCAN. Gdyby zostały użyte operatory skanowania indeksów, przedstawione byłyby w postaci rombów i opatrzone etykietą IXSCAN. W wypadku wyodrębniania niewielkich ilości danych użycie utworzonego dla tabeli indeksu jest bardziej ekonomiczne niż skanowanie tabeli.

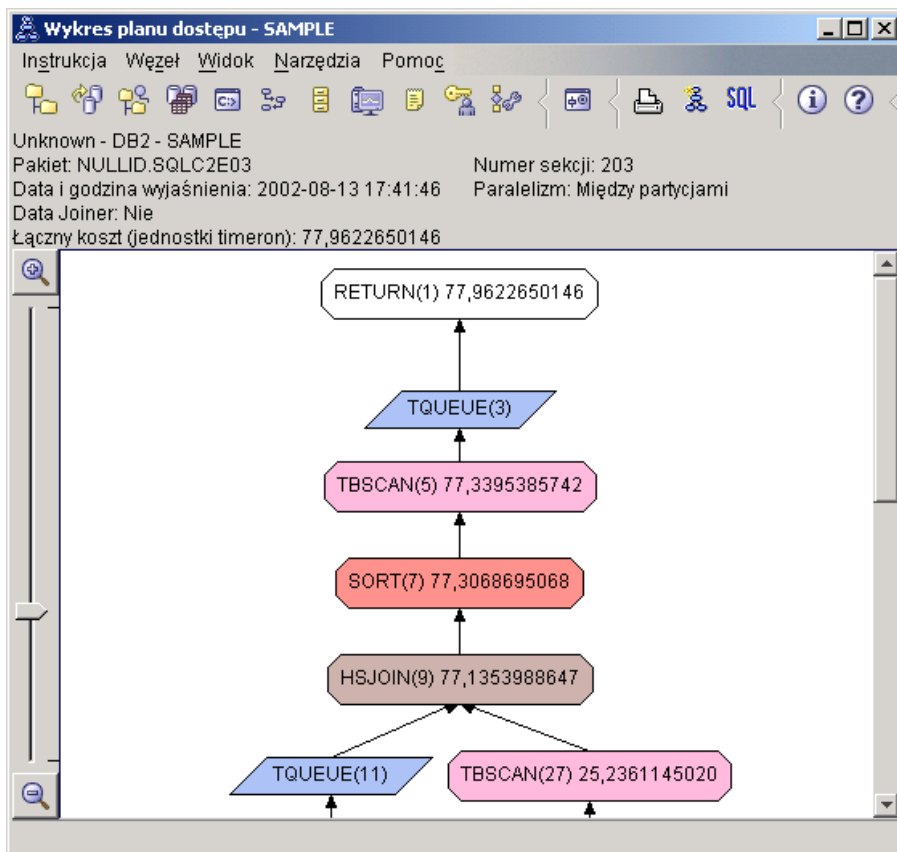
3. Jak efektywny jest ten plan dostępu?

Efektywność planu dostępu można określić pod warunkiem, że jest on oparty na rzeczywistych danych statystycznych. Ponieważ w planie dostępu optymalizator wykorzystał statystyki domyślne, określenie efektywności planu dostępu nie jest możliwe.

Z reguły należy zanotować szacowany łączny koszt planu dostępu, który będzie można porównać z kosztami zmodyfikowanych planów. Koszt podany przy każdym węźle jest skumulowany i obejmuje wszystkie czynności od początku przetwarzania zapytania do danego węzła włącznie.

Uwaga: W wypadku partycjonowanych baz danych jest to koszt skumulowany dla węzła zużywającego najwięcej zasobów.

Podany w oknie Wykres planu dostępu łączny koszt wynosi około 78 jednostek timeron. Wartość ta wyświetlana jest w węźle **RETURN (1)** u góry wykresu. Szacowany łączny koszt jest także wyświetlany w górnym obszarze okna.



4. Co dalej?

W przykładzie z Zapytaniem 2 przyjrzymy się planowi dostępu podstawowego zapytania po uruchomieniu komendy **runstats**. Komenda **runstats** dostarcza optymalizatorowi aktualną statystykę dotyczącą wszystkich tabel występujących w zapytaniu.

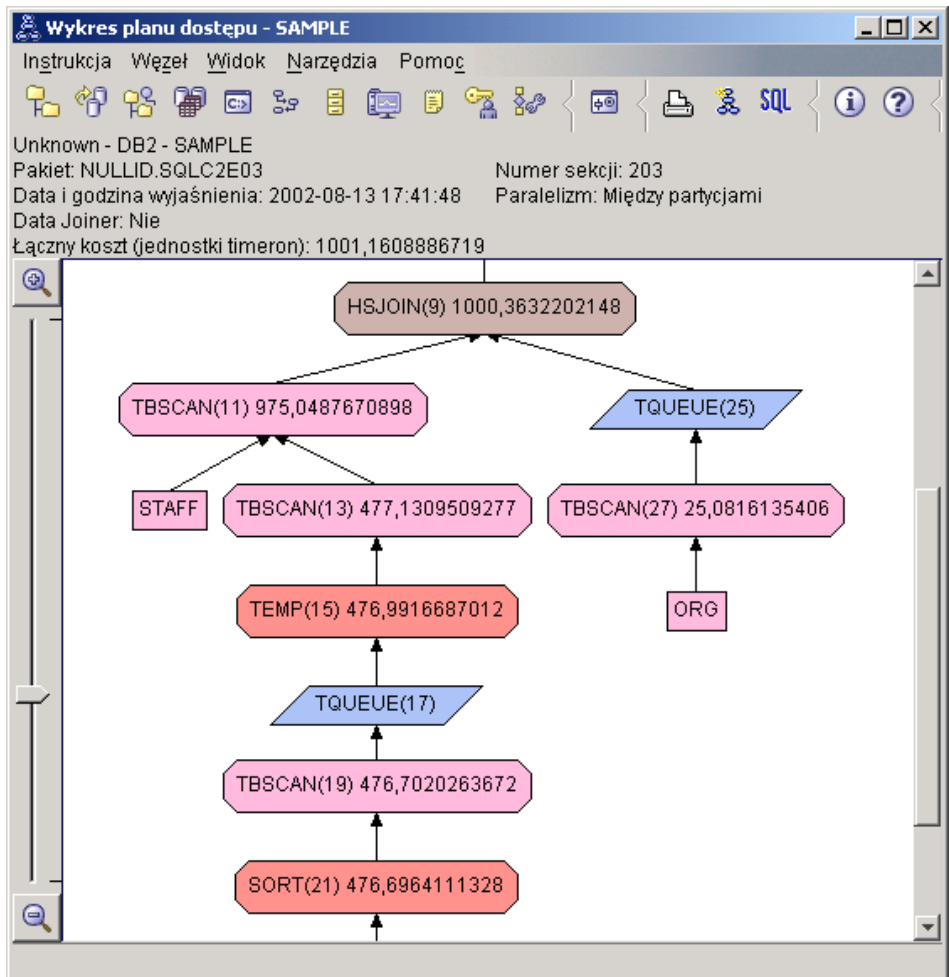
Gromadzenie aktualnej statystyki tabel i indeksów przy użyciu komendy **runstats**

Punktem wyjścia dla tego przykładu jest plan dostępu dla Zapytania 1. Plan ten zostanie zmodyfikowany przez zgromadzenie aktualnej statystyki za pomocą komendy **runstats**.

Zdecydowanie zaleca się użycie komendy **runstats** w celu zgromadzenia bieżących informacji statystycznych na temat tabel i indeksów, zwłaszcza jeśli od czasu ostatniego wykonania komendy **runstats** miały miejsce znaczące aktualizacje lub utworzono nowe indeksy. Spowoduje to przekazanie optymalizatorowi najdokładniejszych informacji, umożliwiających określenie najlepszego planu dostępu. Jeśli nie jest dostępna aktualna statystyka, optymalizator może wybrać nieefektywny plan dostępu oparty na niedokładnej statystyce domyślnej.

Należy koniecznie używać komendy **runstats** *po* każdej aktualizacji tabeli; w przeciwnym razie optymalizator może potraktować tabelę tak, jak gdyby była pusta. Problem ten jest wyraźnie widoczny, jeśli licznosc na stronie Szczegóły dotyczące operatora wynosi zero. W takim wypadku należy zaktualizować tabele, uruchomić komendę **runstats** i ponownie utworzyć obrazy stanu wyjaśniania zmodyfikowanych tabel.

Aby wyświetlić wykres planu dostępu dla Zapytania 2: w oknie Historia wyjaśnionych instrukcji kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 2. Zostanie otwarte związane z tą instrukcją okno Wykres planu dostępu.



Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

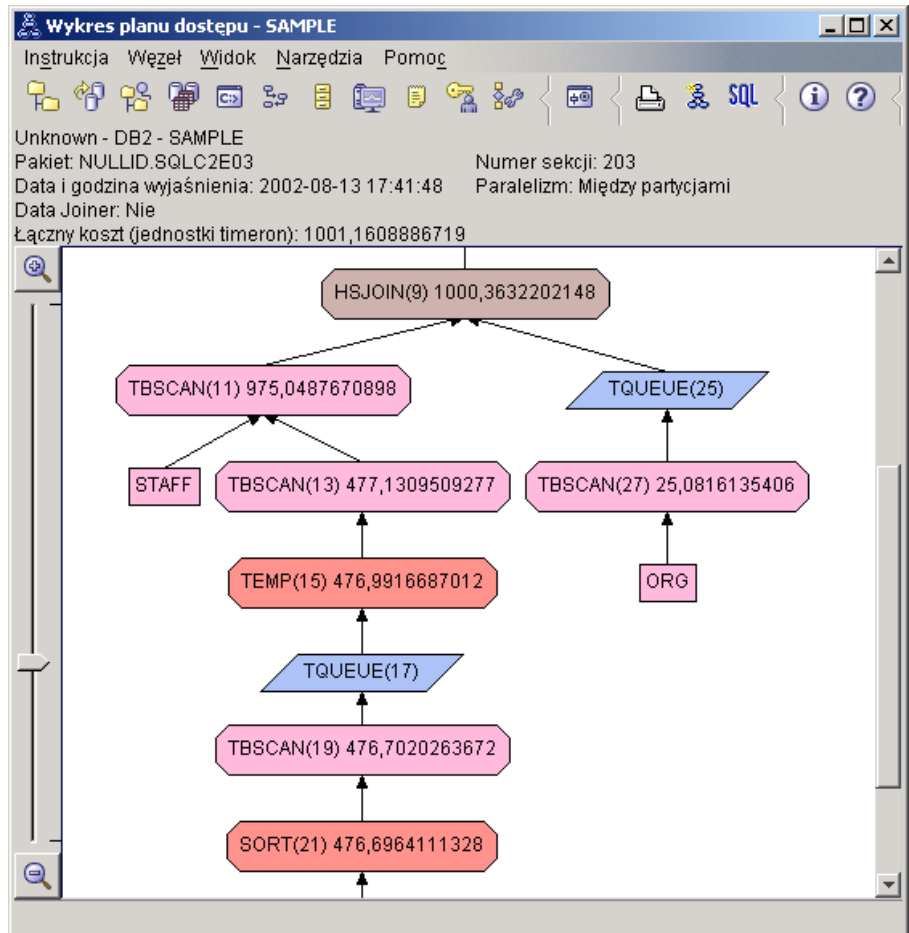
1. Czy dla każdej tabeli występującej w zapytaniu istnieje aktualna statystyka?

Z informacji przedstawionych w oknie Statystyki tabeli tabeli ORG wynika, że optymalizator użył rzeczywistych danych statystycznych (wartość **STATS_TIME** określa godzinę, o której zgromadzono te dane). Dokładność danych statystycznych zależy od tego, czy od czasu uruchomienia komendy **runstats** zawartość tabel uległa istotnym zmianom.

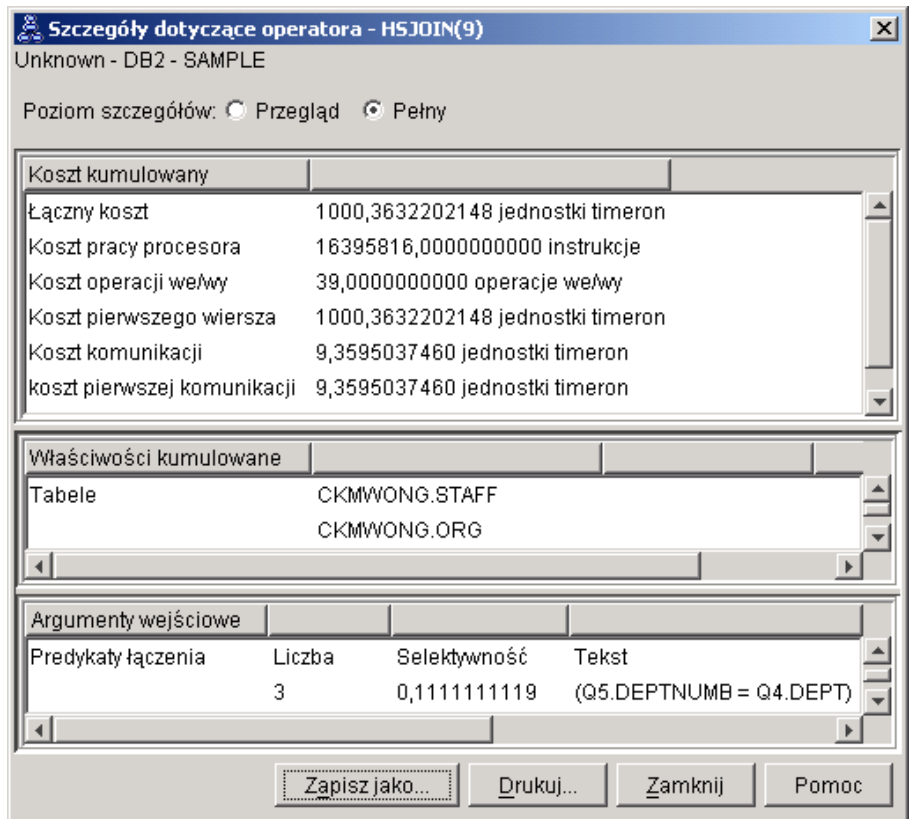
Statystyka	Wyjaśnione	Bieżące
CREATE_TIME	2002-08-13 17:40:05	
STATS_TIME	2002-08-13 17:41:48	Statystyki nie zostały zaktualizowane
CARD	2	
NPAGES	1	
FPAGES	1	
COLCOUNT	5	
OVERFLOW	0	
TABLESPACE	USERSPACE1	
INDEX_TABLESPACE		
LONG_TABLESPACE		
ULOTNE	Nie	Nie

2. Czy w tym planie dostępu zastosowane są najbardziej efektywne metody dostępu do danych?

Podobnie jak w przypadku Zapytania nr 1, także w planie dostępu Zapytania nr 2 występują operatory skanowania tabel (TBSCAN), a nie skanowania indeksów (IXSCAN). Mimo że istnieją aktualne statystyki, nie było stosowane skanowanie indeksów, ponieważ brak jest indeksów dla kolumn używanych w zapytaniu. Jednym ze sposobów optymalizacji zapytania może być utworzenie indeksów dla kolumn używanych do łączenia tabel (tj. kolumn występujących w predykatkach łączenia). W tym przykładzie jest to pierwszy operator łączenia scalającego ze skanowaniem: HSJOIN (9).



W oknie Szczegóły operatora dla operatora HSJOIN (9) należy popatrzeć na sekcję **Predykaty łączenia** w grupie **Argumenty wejściowe**. W kolumnie **Tekst** wymienione są kolumny używane w tej operacji łączenia. W omawianym przykładzie są to kolumny DEPTNUMB i DEPT.



3. Jak efektywny jest ten plan dostępu?

Plany dostępu oparte na aktualnych statystykach zawsze stanowią podstawę do realistycznego oszacowania kosztu (mierzonego w jednostkach timeron). Ponieważ w wypadku Zapytania 1 szacowany koszt był oparty na statystykach domyślnych, nie możemy porównywać ze sobą kosztów tych dwóch planów dostępu. To, czy koszt jest większy, czy mniejszy, nie ma znaczenia. Aby uzyskać poprawną miarę efektywności, trzeba porównać koszty planów dostępu opartych na rzeczywistych danych statystycznych.

4. Co dalej?

W przykładzie z Zapytaniem 3 przyjrzymy się skutkom dodania indeksów zbudowanych na kolumnach DEPTNUMB i DEPT. Dodanie indeksów zbudowanych na kolumnach używanych w predykatkach łączenia może poprawić wydajność.

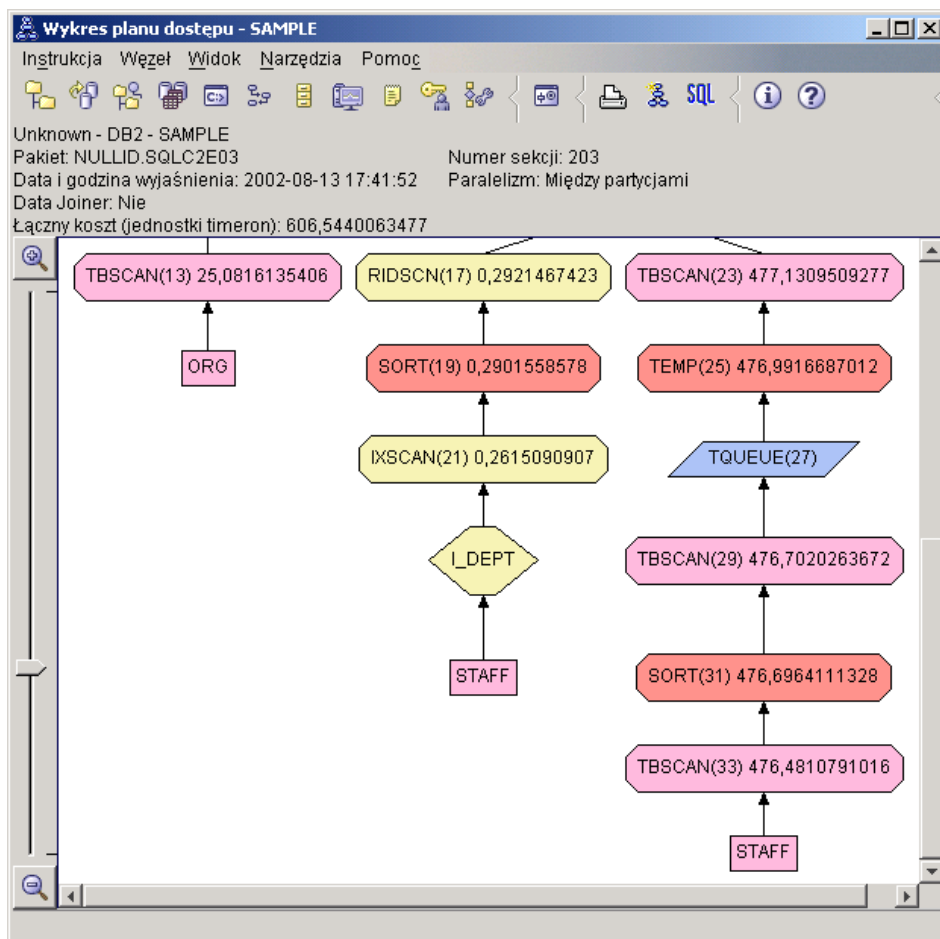
Tworzenie indeksów dla kolumn używanych do łączenia tabel w zapytaniu

Punktem wyjścia dla tego przykładu jest plan dostępu dla Zapytania 2. Plan ten zostanie zmodyfikowany poprzez utworzenie indeksów na kolumnie DEPT tabeli STAFF i na kolumnie DEPTNUMB tabeli ORG.

Uwaga: W wersji 8 zalecane indeksy mogą zostać utworzone za pomocą Kreatora optymalizacji wydajności dla obciążenia.

Aby wyświetlić wykres planu dostępu dla Zapytania 3: w oknie Historia wyjaśnionych instrukcji kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 3. Zostanie otwarte związane z tą instrukcją okno Wykres planu dostępu.

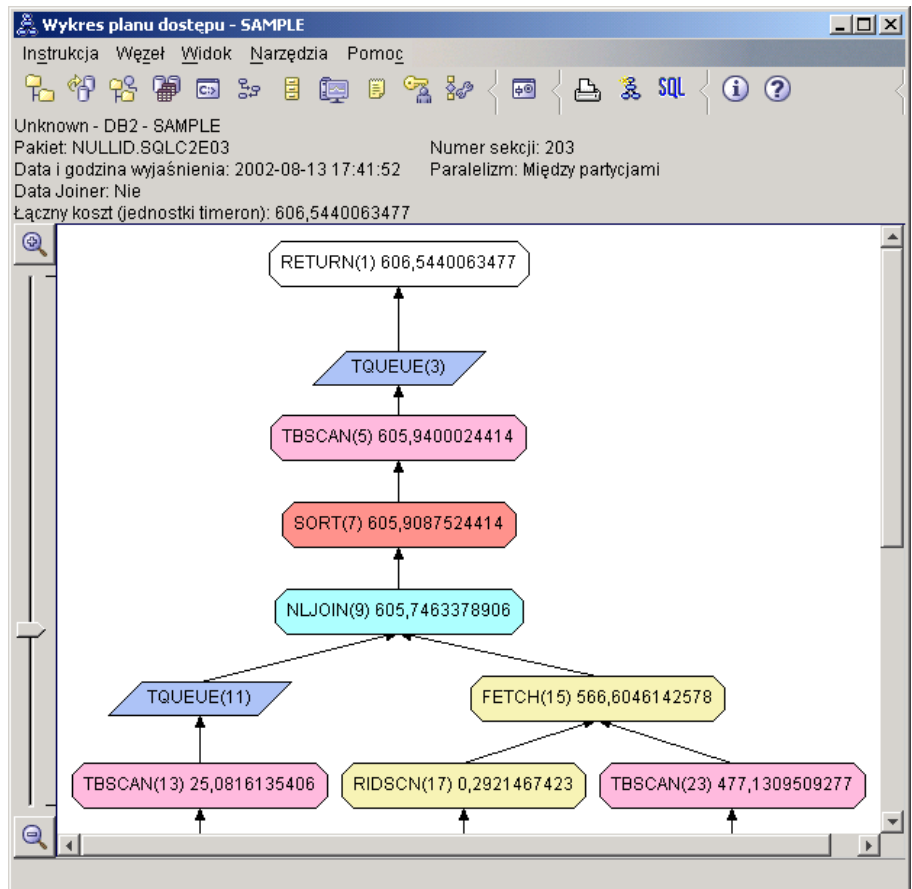
Uwaga: Nawet gdyby dla kolumny DEPTNUM został utworzony indeks, optymalizator nie użyłby go.



Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

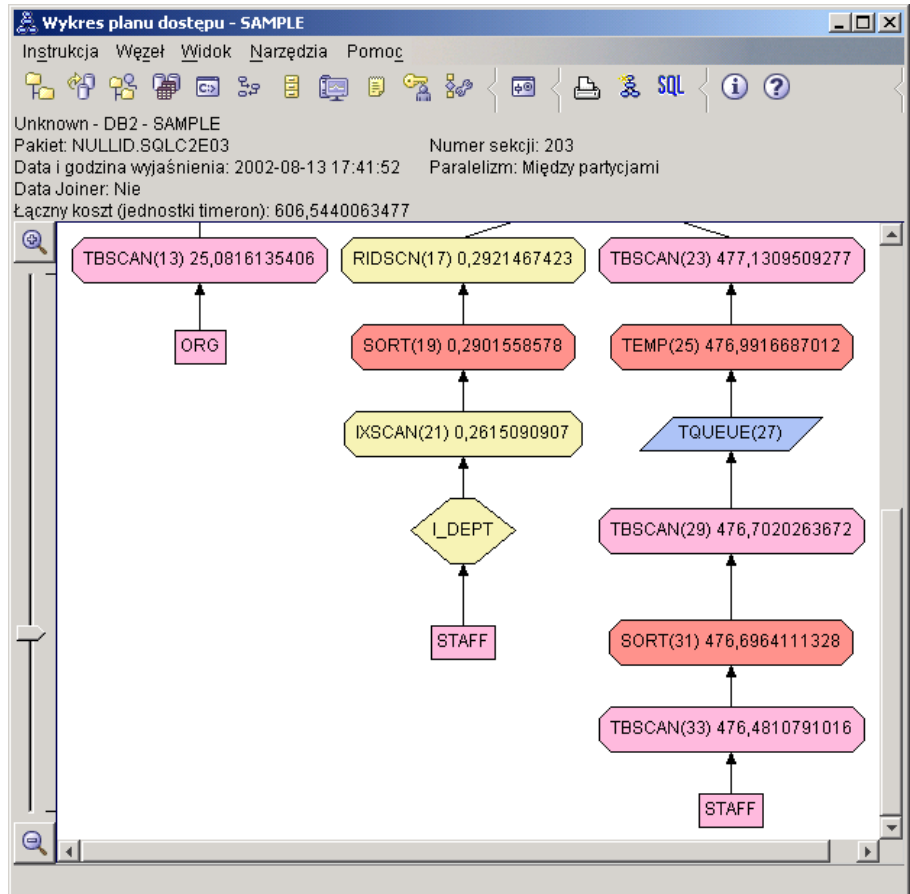
1. Co zmieniło się w planie dostępu w wyniku utworzenia indeksów?

Ponad tabelą STAFF został dodany nowy węzeł w kształcie rombu, **I_DEPT**. Węzeł ten reprezentuje indeks utworzony na kolumnie DEPT. Pokazuje on, że w celu określenia, które wiersze mają zostać pobrane, optymalizator zastosował skanowanie indeksu, a nie skanowanie tabeli.

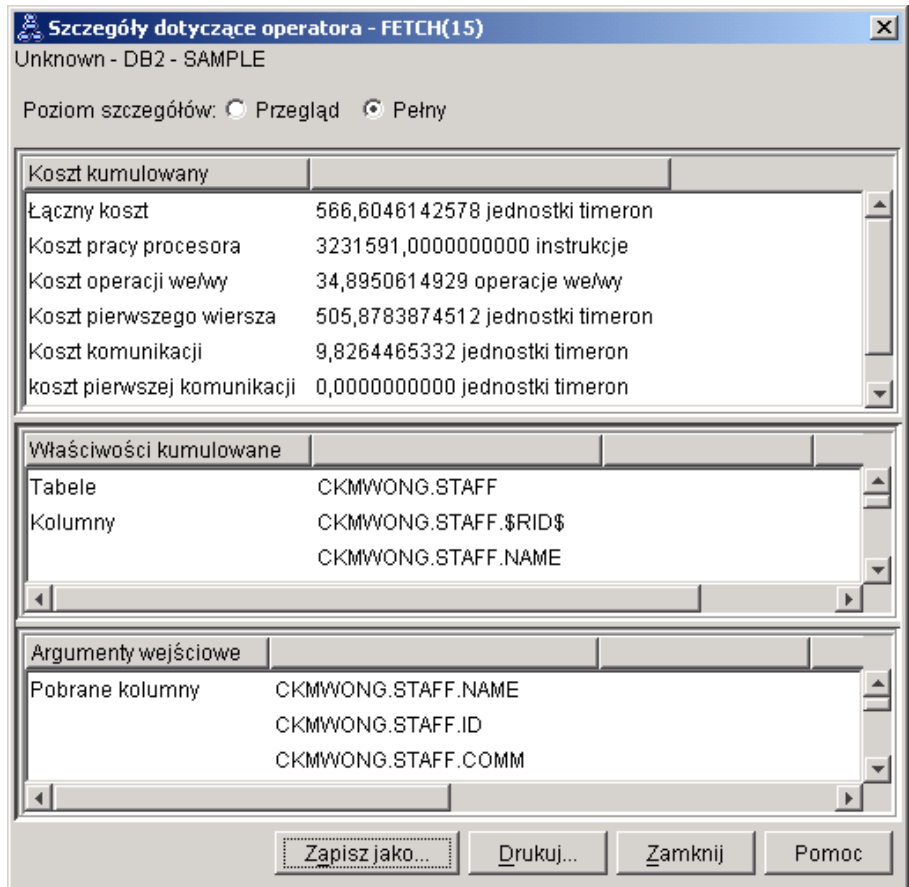


2. Czy w tym planie dostępu zastosowane są najbardziej efektywne metody dostępu do danych?

Plan dostępu dla tego zapytania ilustruje efekt utworzenia indeksu na kolumnie DEPTNUMB tabeli ORG, którym jest użycie operatorów FETCH (15) i IXSCAN (21), oraz efekt utworzenia indeksu na kolumnie DEPT tabeli STAFF. W Zapytaniu 2 indeks ten nie był używany, dlatego w tamtym przykładzie zastosowano skanowanie tabeli.



W oknie Szczegóły operatora dla operatora FETCH (15) wyświetlone są używane w tej operacji kolumny.



W tym wypadku obliczenia wykazały, że zastosowanie skanowania indeksu i operatora pobierania jest mniej kosztowne niż stosowana we wcześniejszych planach dostępu operacja pełnego skanowania tabeli.

3. Jak efektywny jest ten plan dostępu?

Ten plan dostępu jest bardziej ekonomiczny niż plan z poprzedniego przykładu. Koszt kumulowany został zmniejszony z około 1,000 jednostek timeron w Zapytaniu 2 do około 606 jednostek timeron w Zapytaniu 3.

4. Co dalej?

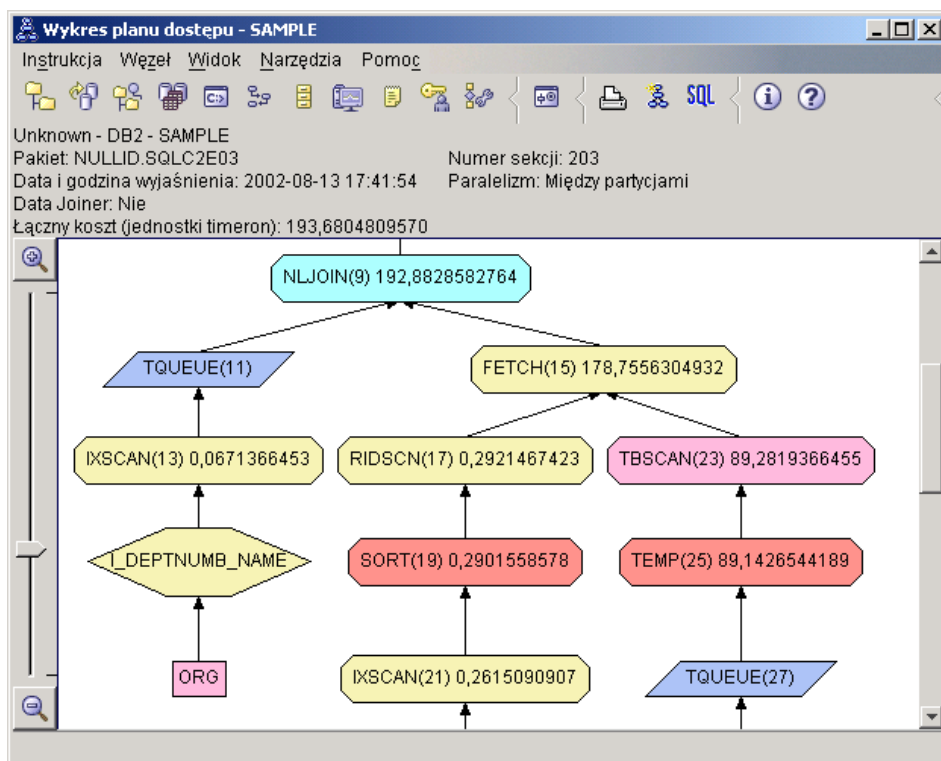
W Zapytaniu 4 operator pobierania i skanowania indeksu został zastąpiony operatorem skanowania indeksu bez pobierania. Utworzenie dodatkowych indeksów może zmniejszyć szacowany koszt planu dostępu.

Tworzenie dodatkowych indeksów dla kolumn tabeli

Punktem wyjścia dla tego przykładu jest plan dostępu dla Zapytania nr 3. Plan ten zostanie zmodyfikowany w wyniku utworzenia indeksu na kolumnie JOB w tabeli

STAFF i dodania kolumny DEPTNAME do istniejącego indeksu w tabeli ORG. (Dodanie oddzielnego indeksu mogłoby spowodować konieczność wykonania dodatkowych operacji dostępu).

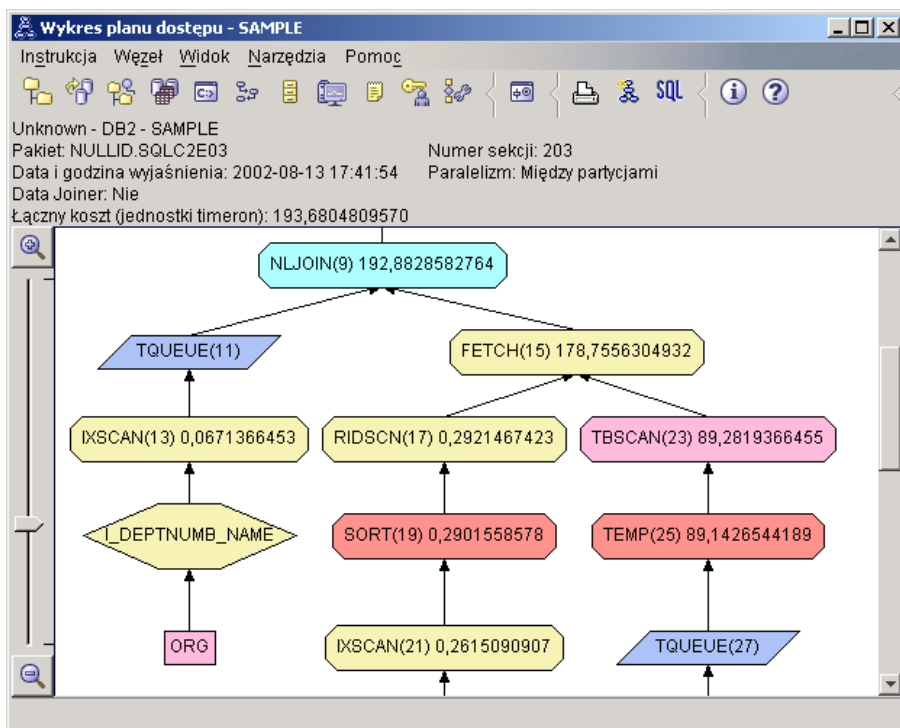
Aby wyświetlić wykres planu dostępu dla Zapytania 4: w oknie Historia wyjaśnionych instrukcji kliknij dwukrotnie wpis identyfikowany jako Zapytanie nr 4. Zostanie otwarte związane z tą instrukcją okno Wykres planu dostępu.



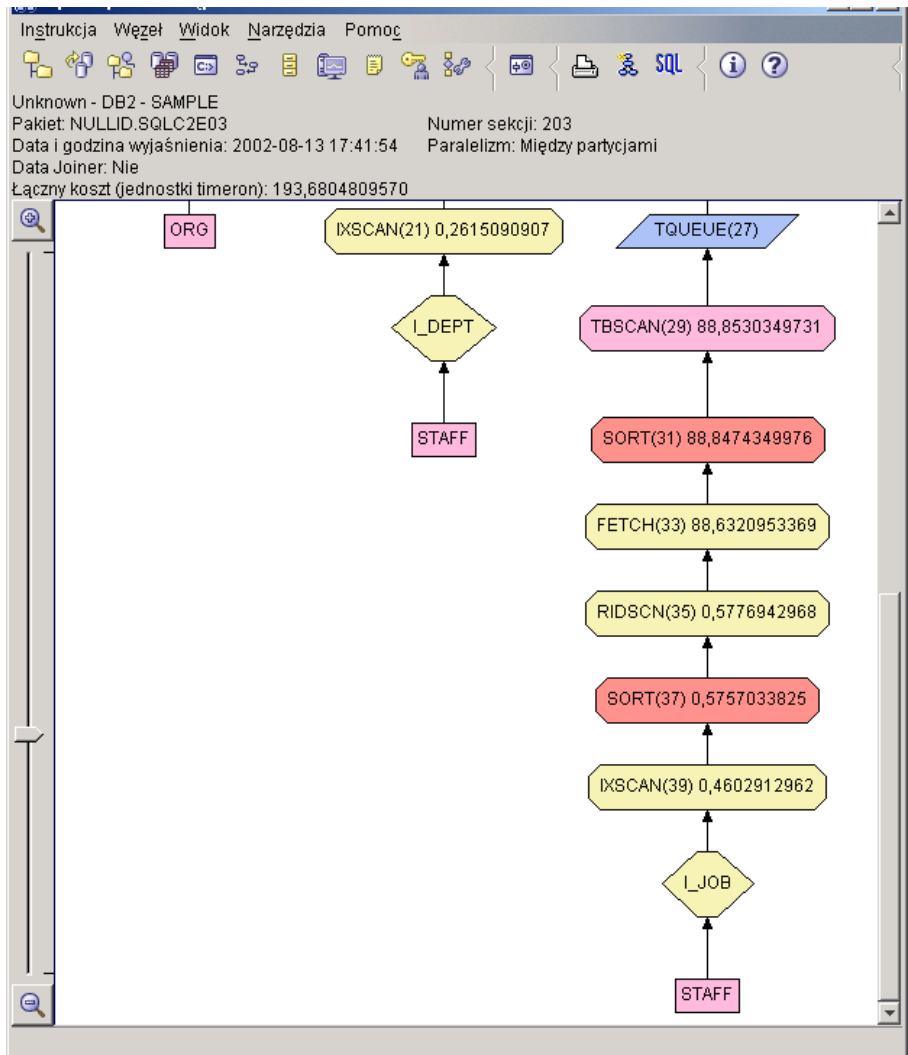
Udzielenie odpowiedzi na następujące pytania ułatwi zrozumienie metod poprawy wydajności zapytania.

1. Co zmieniło się w tym planie dostępu w wyniku utworzenia dodatkowych indeksów?

Należy zauważyć, że w środkowej części wykresu planu dostępu dotychczasowy operator skanowania tabeli, dotyczący tabeli ORG, został zastąpiony operatorem skanowania indeksu IXSCAN (13). Dodanie do tego indeksu kolumny DEPTNAME dla tabeli ORG pozwoliło na ulepszenie w optymalizatorze dostępu wymagającego zastosowania operatora skanowania tabeli.



Zwróćmy uwagę, że w dolnej części wykresu planu dostępu dotychczasowe operatory skanowania indeksu i pobierania dotyczące tabeli STAFF zostały zastąpione samym operatorem skanowania indeksu IXSCAN (39). Utworzenie indeksu JOB dla tabeli STAFF pozwoliło optymalizatorowi na wyeliminowanie dodatkowego dostępu wymagającego zastosowania operatora pobierania.



2. Jak efektywny jest ten plan dostępu?

Ten plan dostępu jest bardziej ekonomiczny niż plan z poprzedniego przykładu.

Koszt kumulowany został zmniejszony z około 606 jednostek timeron w Zapytaniu 3 do około 193 jednostek timeron w Zapytaniu 4.

Co dalej

Szczegółowe informacje na temat innych działań, które można wykonać w celu poprawienia wydajności, dostępne są w książce *Administration Guide*. Po zapoznaniu się z tymi informacjami można powrócić do narzędzia Visual Explain, aby ocenić wpływ tych działań na wydajność.

Dodatek A. Pojęcia dotyczące narzędzia graficznego Visual Explain

Plan dostępu

Do przetłumaczenia instrukcji SQL, którą można wyjaśnić, niezbędne są określone dane. *Plan dostępu* określa kolejność operacji przy uzyskiwaniu dostępu do tych danych. Umożliwia on przeglądanie statystyk wybranych tabel, indeksów lub kolumn, właściwości operatorów, informacji globalnych, takich jak statystyki obszaru tabel i funkcji, a także parametrów konfiguracyjnych wpływających na wydajność. Za pomocą narzędzia graficznego Visual Explain można wyświetlić plan dostępu instrukcji SQL w postaci graficznej.

Optymalizator generuje plan dostępu zawsze, gdy kompilowana jest instrukcja SQL podlegająca wyjaśnieniu. W przypadku instrukcji statycznych odbywa się to w czasie prekompilacji/konsolidacji, a w przypadku instrukcji dynamicznych - w czasie wykonywania.

Należy mieć świadomość, że plan dostępu jest *oszacowaniem* opartym na dostępnych informacjach. Optymalizator dokonuje oszacowania na podstawie takich informacji, jak:

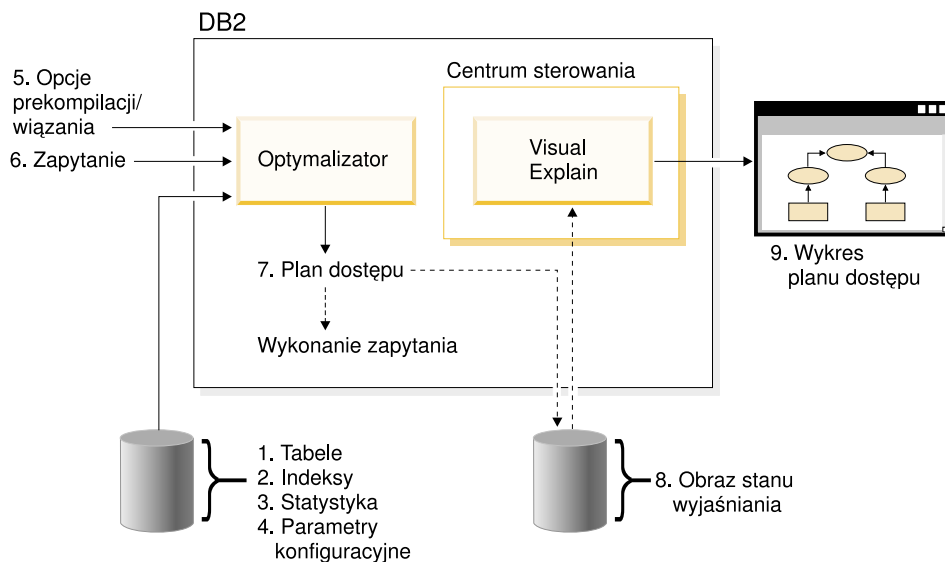
- Statystyki w systemowych tabelach katalogowych (jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats)
- Parametry konfiguracyjne
- Opcje powiązań bazy danych
- Klasa optymalizacji zapytania

Informacje o kosztach skojarzone z planem dostępu stanowią *najlepsze możliwe oszacowanie* zużycia zasobów przez zapytanie. Rzeczywisty czas przetwarzania zapytania może być zmienny w zależności od czynników, na które program DB2 nie ma wpływu (na przykład od liczby jednocześnie uruchomionych aplikacji). Rzeczywisty czas przetwarzania można zmierzyć po uruchomieniu zapytania, korzystając z monitorowania wydajności.

Wykres planu dostępu

Narzędzie graficzne Visual Explain przy tworzeniu wykresu planu dostępu wykorzystuje informacje z szeregu źródeł, tak jak to pokazano na poniższej ilustracji. W oparciu o różnorodne dane wejściowe optymalizator wybiera plan dostępu, a narzędzie graficzne Visual Explain wyświetla go na *wykresie planu dostępu*. Węzły na wykresie reprezentują tabele, indeksy oraz wszystkie wykonywane na nich operacje. Połączenia

między węzłami reprezentują przepływ danych.



Poniższa lista zadań odpowiada zadaniom przedstawionym na ilustracji. (Przerwane linie reprezentują kroki, które muszą być wykonane w programie Visual Explain).

1. Dostrojenie projektu tabeli i reorganizacja danych tabeli.
2. Utworzenie odpowiednich indeksów.
3. Zastosowanie komendy runstats w celu dostarczenia aktualnych statystyk do optymalizatora.
4. Wybranie odpowiednich parametrów konfiguracyjnych.
5. Wybranie odpowiednich opcji wiązania.
6. Zaprojektowanie zapytań służących do pobierania tylko potrzebnych danych.
7. Utworzenie planu dostępu.
8. Utworzenie obrazów stanu wyjaśniania.
9. Wyświetlenie i wykorzystanie wykresu planu dostępu.

Na przykład, aby użyć narzędzia graficznego Visual Explain, należy najpierw zaktualizować bieżącą statystykę, wykonując komendę **runstats** na tabelach i indeksach wykorzystywanych w instrukcji. Te statystyki, parametry konfiguracyjne, opcje wiązania i samo zapytanie są wykorzystywane przez optymalizator do utworzenia planu dostępu i obrazu stanu wyjaśniania w trakcie wiązania pakietu. Narzędzie graficzne Visual Explain wykorzystuje uzyskany obraz stanu wyjaśniania przy wyświetlaniu wykresu planu dostępu instrukcji.

Węzeł wykresu planu dostępu

Wykres planu dostępu ma postać drzewa złożonego z *węzłów*. Węzły te reprezentują:

- Tabele, przedstawione jako prostokąty
- Indeksy, przedstawione jako romby

- Operatory, przedstawione jako ośmiokąty. Operatory TQUEUE, przedstawione jako równoległoboki
- Funkcje tabelowe, przedstawione jako sześciokąty.

Klastrowanie

Z czasem liczne aktualizacje mogą spowodować zmianę rozmieszczenia wierszy na stronach danych, a w konsekwencji zmniejszenie stopnia *klastrowania* pomiędzy indeksem a stronami danych. Reorganizacja tabeli z uwzględnieniem wybranego indeksu powoduje ponowne zgrupowanie danych w klastry. Indeks klastrowy jest najbardziej użyteczny w przypadku kolumn z predykatami zakresów, ponieważ usprawnia dostęp sekwencyjny do danych w tabeli podstawowej. W rezultacie zmniejsza się liczba pobrań stron, gdyż podobne wartości znajdują się na tej samej stronie danych.

Zwykle tylko jeden z indeksów tabeli może charakteryzować się wysokim stopniem klastrowania.

Aby sprawdzić stopień klastrowania indeksu, należy kliknąć dwukrotnie jego węzeł, co spowoduje wyświetlenie okna Statystyka indeksów. W oknie tym wyświetlane są wartości współczynnika lub stopnia klastrowania. Jeśli wartość jest niska, należy rozważyć reorganizację danych w tabeli.

Więcej informacji można znaleźć w sekcji poświęconej reorganizacji danych w tabeli w podręczniku *Administration Guide*.

Kontener

Kontener jest to fizyczne miejsce przechowywania danych. Kontener jest skojarzony z obszarem tabel, i może być plikiem, katalogiem albo urządzeniem.

Koszt

Koszt to w kontekście narzędzia Visual Explain przewidywany całkowity poziom użycia zasobów niezbędny do wykonania planu dostępu dla instrukcji (lub elementów instrukcji). Koszt jest pochodną kosztu procesora (liczby instrukcji) i operacji we/wy (liczby operacji przeszukiwania i przesyłania stron).

Jednostką kosztu jest *cykl zegara*. Cykl zegara nie odpowiada bezpośrednio żadnemu rzeczywistemu czasowi, a jedynie stanowi pobieżne względne oszacowanie ilości zasobów (kosztu) potrzebnych menedżerowi bazy danych do wykonania dwóch planów dla tego samego zapytania.

Koszt podawany w każdym węźle operatora wykresu planu dostępu jest kosztem skumulowanym, liczonym od początku wykonywania planu dostępu do wykonania

danego operatora włącznie. Nie odzwierciedla on takich czynników, jak obciążenie systemu lub koszt zwracania wierszy danych do użytkownika.

Blokowanie kursorów

Blokowanie kursorów jest techniką ograniczającą nakład pracy poprzez umożliwienie menedżerowi bazy danych pobierania *bloków* wierszy w ramach jednej operacji. Podczas przetwarzania wiersze te są przechowywane w pamięci podręcznej. Pamięć podręczna jest przydzielana po wysłaniu przez aplikację żądania OPEN CURSOR, a zwalniana po zamknięciu kursora. Po przetworzeniu wszystkich wierszy pobierany jest następny blok wierszy.

Aby wybrać typ blokowania kursora, należy w instrukcjach **PREP** i **BIND** użyć opcji **BLOCKING** z następującymi parametrami:

UNAMBIG

Blokowane są tylko kursory jednoznaczne (ustawienie domyślne).

ALL Blokowane są zarówno kursory niejednoznaczne, jak i jednoznaczne.

NO Kursory nie są blokowane.

Więcej informacji można znaleźć w dokumencie *Administration Guidew* sekcji dotyczącej blokowania kursora.

Obszar tabel DMS

W bazie danych mogą istnieć dwa typy obszarów tabel: obszary zarządzane przez bazę danych (DMS) i obszary zarządzane przez system (SMS).

Obszarami tabel DMS zarządza menedżer bazy danych. Są one zbudowane i zoptymalizowane pod kątem jego potrzeb.

Definicja obszaru tabel DMS obejmuje listę plików (lub urządzeń), w których przechowywane są dane bazy danych w formacie obszaru tabel DMS.

Istnieje możliwość dodawania wstępnie przydzielonych plików (lub urządzeń) do istniejącego obszaru tabel DMS w celu zwiększenia jego pojemności. Menedżer bazy danych automatycznie rozdziela istniejące dane między wszystkie kontenery należące do danego obszaru tabel.

Obszary tabel DMS i SMS mogą współistnieć w tej samej bazie danych.

Dynamiczny SQL

Dynamiczne instrukcje SQL są instrukcjami SQL przygotowywanymi i wykonywanymi w aplikacji w trakcie jej pracy. W przypadku dynamicznego SQL:

- Użytkownik wydaje instrukcję SQL interaktywnie, za pomocą interfejsu CLI lub procesora wiersza komend.
- Kod źródłowy SQL jest zawarty w zmiennych języka bazowego osadzonych w aplikacji.

Podczas wykonywania dynamicznej instrukcji SQL w programie DB2 tworzony jest plan dostępu oparty na bieżącej statystyce katalogu i parametrach konfiguracyjnych. Taki plan dostępu może być inny przy każdym wykonaniu aplikacji.

Alternatywą dla dynamicznego SQL jest statyczny SQL.

Obraz stanu wyjaśniania

Za pomocą narzędzia Visual Explain można analizować zawartość obrazu stanu wyjaśniania.

Obraz stanu wyjaśniania to skompresowany zbiór informacji gromadzonych podczas wyjaśniania instrukcji SQL. Obraz jest przechowywany jako duży obiekt binarny (BLOB) w tabeli EXPLAIN_STATEMENT i zawiera następujące informacje:

- Wewnętrzną reprezentację planu dostępu, w tym jego operatory i używane tabele oraz indeksy.
- Kryteria decyzyjne wykorzystywane przez optymalizator, w tym statystyki obiektów bazy danych i skumulowane koszty poszczególnych operacji.

Obraz stanu wyjaśniania jest niezbędny do wyświetlenia graficznej reprezentacji planu dostępu instrukcji SQL. Aby obraz stanu wyjaśniania został utworzony:

1. W menedżerze bazy danych muszą istnieć tabele wyjaśniania, w których przechowywane będą obrazy stanu wyjaśniania. Aby uzyskać informacje na temat tworzenia tych tabel, patrz temat pomocy ekranowej Tworzenie tabel wyjaśniania .
2. W przypadku pakietu zawierającego statyczne instrukcje SQL przy wykonywaniu operacji prep lub bind na pakiecie należy ustawić opcję EXPLSNAP na wartość ALL lub YES. W pakiecie utworzony zostanie obraz stanu wyjaśniania dla każdej instrukcji SQL podlegającej wyjaśnieniu. Aby uzyskać więcej informacji na temat komend **BIND** i **PREP**, patrz *Command Reference*.
3. W przypadku dynamicznych instrukcji SQL należy ustawić opcję EXPLSNAP na wartość ALL przy wykonywaniu operacji bind na aplikacji, która te instrukcje wydaje, lub ustawić rejestr specjalny CURRENT EXPLAIN SNAPSHOT na wartość YES lub EXPLAIN przed interaktywnym wywołaniem tych instrukcji. Więcej informacji można znaleźć w dokumencie *SQL Reference* w sekcji dotyczącej bieżących obrazów stanu wyjaśniania.

Instrukcja SQL, którą można wyjaśnić

Instrukcja SQL, którą można wyjaśnić to instrukcja SQL na której można przeprowadzić operację wyjaśniania.

Do instrukcji SQL, które można wyjaśnić należą:

- SELECT
- INSERT
- UPDATE
- DELETE
- VALUES

Instrukcja wyjaśniona

Instrukcja wyjaśniona to instrukcja SQL, na której wykonano operację wyjaśniania. Instrukcje wyjaśnione są wyświetlane w oknie Historia wyjaśnionych instrukcji.

Operand

Operand jest obiektem, na którym wykonywana jest operacja. Na przykład, tabele i indeksy są operandami różnorodnych operatorów, takich jak TBSCAN i IXSCAN.

Operator

Operator jest czynnością, która ma zostać wykonana na danych albo danych wyjściowych tabeli lub indeksu podczas realizacji planu dostępu dla instrukcji SQL.

Na wykresie planu dostępu mogą wystąpić następujące operatory:

DELETE

Usuwa wiersze z tabeli.

EISCAN

Skanuje indeks zdefiniowany przez użytkownika w celu wygenerowania ograniczonego strumienia wierszy.

FETCH

Pobiera kolumny z tabeli przy użyciu identyfikatora konkretnego rekordu.

FILTER

Filtruje dane przez zastosowanie do nich jednego lub większej liczby predykatów.

GRPBY

Grupuje wiersze według wspólnych wartości wskazanych kolumn lub funkcji i oblicza wartości funkcji zbiorów.

HSJOIN

Reprezentuje łączenie mieszające, w którym dwie lub większa liczba tabel jest przemieszana w kolumnach łączenia.

INSERT

Wstawia wiersze do tabeli.

IXAND

Wykonuje operację AND na identyfikatorach wierszy (RID) z dwóch lub większej liczby operacji skanowania indeksów.

IXSCAN

Skanuje indeks tabeli z uwzględnieniem opcjonalnych warunków rozpoczęcia/zakończenia, generując uporządkowany strumień wierszy.

MSJOIN

Reprezentuje połączenie scalające, w którym zarówno tabela wewnętrzna, jak i zewnętrzna muszą występować w porządku predykatu połączenia.

NLJOIN

Reprezentuje zagnieżdżone połączenie w pętli, w którym tabela wewnętrzna jest używana jeden raz dla każdego wiersza tabeli zewnętrznej.

RETURN

Reprezentuje zwrot danych z zapytania do użytkownika.

RIDSCN

Skanuje listę identyfikatorów wierszy (RID) uzyskaną z jednego lub kilku indeksów.

SHIP

Pobiera dane ze zdalnego źródła bazy danych. Operator używany w systemie stowarzyszeń.

SORT

Sortuje wiersze według określonych kolumn i opcjonalnie eliminuje powielone pozycje.

TBSCAN

Pobiera wiersze, odczytując wszystkie potrzebne dane wprost ze stron danych.

TEMP

Zapisuje dane w tabeli tymczasowej, z której będą odczytywane w przyszłości (być może wielokrotnie).

TQUEUE

Przekazuje dane z tabeli między agentami bazy danych.

UNION

Przeprowadza konkatenację strumieni wierszy z wielu tabel.

UNIQUE

Eliminuje wiersze z powielonymi wartościami w określonych kolumnach.

UPDATE

Aktualizuje wiersze w tabeli.

CMPEXP

Nazwa operatora: CMPEXP

Reprezentuje: Obliczanie wyrażeń niezbędnych do uzyskania wyników pośrednich lub końcowych.

(Ten operator jest stosowany tylko w trybie debugowania.)

DELETE

Nazwa operatora: DELETE

Reprezentuje: Usunięcie wierszy z tabeli.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do usunięcia.

Wskazówka dotycząca wydajności:

- W przypadku usuwania wszystkich wierszy z tabeli, należy rozważyć użycie instrukcji DROP TABLE albo komendy **LOAD REPLACE**.

EISCAN

Nazwa operatora: EISCAN

Reprezentuje: Ten operator skanuje indeks zdefiniowany przez użytkownika w celu wygenerowania ograniczonego strumienia wierszy. W procesie skanowania wykorzystywanych jest wiele warunków rozpoczęcia/zakończenia z podanej przez użytkownika funkcji generującej zakres.

Operację tę wykonuje się w celu zawężenia zbioru wierszy spełniających kryterium przed uzyskaniem dostępu do tabeli podstawowej (w oparciu o predykaty).

Wskazówka dotycząca wydajności:

- Z czasem w wyniku aktualizacji bazy danych może nastąpić fragmentacja indeksu powodująca powstanie większej liczby stron indeksu, niż to konieczne. Sytuację tę można skorygować przez ponowne utworzenie indeksu lub reorganizację indeksu.
- Jeśli statystyki nie są aktualne, należy je zaktualizować za pomocą komendy runstats.

FETCH

Nazwa operatora: FETCH

Reprezentuje: Pobieranie kolumn z tabeli przy użyciu określonego identyfikatora wiersza (RID).

Wskazówki dotyczące wydajności:

- Należy rozbudować klucze indeksowania, tak aby obejmowały pobierane kolumny, co wyeliminuje konieczność dostępu do stron danych.
- Aby wyświetlić okno statystyki indeksu, należy odszukać indeks związany z operacją pobierania i kliknąć dwukrotnie jego węzeł. Stopień klastrowania indeksu powinien być wysoki.
- Należy zwiększyć rozmiar buforu, jeśli liczba operacji wejścia/wyjścia (I/O) związanych z operacją pobierania jest większa od liczby stron w tabeli.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą `runstats`.
Statystyki kwantyli i najczęściej występujących wartości informują o selektywności predykatów, która decyduje o tym, kiedy zamiast operatorów skanowania tabel mogą być stosowane operatory skanowania indeksów. Aby zaktualizować te statystyki, należy w odniesieniu do tabeli użyć komendy `runstats` z klauzulą `WITH DISTRIBUTION`.

FILTER

Nazwa operatora: FILTER

Reprezentuje: Zastosowanie predykatów pochodnych powodujących filtrowanie danych w oparciu o kryteria podane w tych predykatkach.

Wskazówki dotyczące wydajności:

- Należy sprawdzić, czy użyto predykatów pobierających tylko potrzebne dane. Na przykład należy sprawdzić, czy wartość selektywności predykatów odpowiada części tabeli, która ma zostać zwrócona.
- Należy upewnić się, czy wybrano klasę optymalizacji równą co najmniej 3, aby optymalizator zastąpił podzapytanie łączeniem. Jeśli nie jest to możliwe, należy spróbować ręcznie przekształcić zapytanie, w celu wyeliminowania podzapytania. Odpowiedni przykład można znaleźć w sekcji poświęconej ponownemu generowaniu zapytań przez kompilator SQL, w podręczniku *Administration Guide*.

GENROW

Nazwa operatora: GENROW

Reprezentuje: Funkcję wbudowaną, która generuje tabelę wierszy bez korzystania z danych wejściowych uzyskanych z tabel, indeksów lub operatorów.

Optymalizator może używać funkcji GENROW do generowania wierszy danych (na przykład w przypadku instrukcji INSERT lub niektórych list IN przekształcanych w łączenia).

Aby wyświetlić szacowaną statystykę tabel wygenerowanych przez funkcję GENROW, należy kliknąć dwukrotnie jej węzeł.

GRPBY

Nazwa operatora: GRPBY

Reprezentuje: Grupowanie wierszy według wspólnych wartości wyznaczonych kolumn lub funkcji. Ta operacja jest niezbędna do generowania grupy wartości lub do obliczania wartości funkcji działających na zbiorach.

Operator GRPBY może być używany także wówczas, gdy nie określono kolumn w klauzuli GROUPBY, o ile na liście SELECT występują funkcje agregujące, co wskazuje na traktowanie całej tabeli jako pojedynczej grupy przy przeprowadzaniu tej agregacji.

Wskazówki dotyczące wydajności:

- Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do grupowania.
- Aby poprawić wydajność instrukcji SELECT zawierającej pojedynczą funkcję agregującą, ale nie zawierającej klauzuli GROUP BY, należy wypróbować następujące metody:
 - W przypadku funkcji agregującej MIN(C), utworzyć indeks rosnący dla C.
 - W przypadku funkcji agregującej MAX(C), utworzyć indeks malejący dla C.

HSJOIN

Nazwa operatora: HSJOIN

Reprezentuje: Łączenie z mieszaniem, w którym kwalifikowane wiersze z tabel są mieszane, tak aby możliwe było bezpośrednie łączenie, bez wstępnego porządkowania zawartości tabel.

Łączenie jest niezbędne zawsze, gdy w klauzuli FROM występuje więcej niż jedna tabela. Łączenie mieszające jest możliwe do zrealizowania, gdy istnieje predykat łączenia, który porównuje kolumny z dwóch różnych tabel. Predykaty łączenia muszą mieć dokładnie ten sam typ danych. Łączenia z mieszaniem mogą także powstać w wyniku ponownego wygenerowania podzapytania, podobnie jak w przypadku operatora NLJOIN.

W przypadku łączenia z mieszaniem tabele wewnętrzne nie muszą być uporządkowane. Łączenie wykonuje się poprzez skanowanie tabeli wewnętrznej łączenia z mieszaniem i wygenerowanie tabeli skanowania z przemieszanych wartości kolumn łączenia. Następnie odczytywana jest tabela zewnętrzna, przy czym wartości kolumn łączenia są mieszane i wyszukiwane w tabeli skanowania wygenerowanej dla tabeli wewnętrznej.

Więcej informacji można znaleźć w sekcji poświęconej pojęciom związanym z łączeniem, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Należy użyć predykatów lokalnych (to znaczy zawierających odniesienia do jednej tabeli), aby ograniczyć liczbę wierszy, które mają być łączone.
- Należy zwiększyć rozmiar sterty sortowania, tak aby tabela skanowania zmieściła się w pamięci.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą `runstats`.

INSERT

Nazwa operatora: INSERT

Reprezentuje: Wstawianie wierszy do tabeli.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do wstawienia.

IXAND

Nazwa operatora: IXAND

Reprezentuje: Iloczyn logiczny AND na wynikach wielu operacji skanowania indeksów przy użyciu technik dynamicznych map bitowych. Operator ten umożliwia zastosowanie iloczynu logicznego predykatów do wielu indeksów w celu ograniczenia do minimum dostępu do tabeli podstawowej.

Operator ten jest stosowany w celu:

- zawężenia zbioru wierszy przed uzyskaniem dostępu do tabeli podstawowej,
- wykonania iloczynu logicznego AND na predykatkach stosowanych do wielu indeksów,
- wykonania iloczynu logicznego AND na wynikach łączeń symbolicznych, stosowanych w łączeniach gwiazdzistych.

Wskazówki dotyczące wydajności:

- Z czasem w wyniku aktualizacji bazy danych może nastąpić fragmentacja indeksu powodująca powstanie większej liczby stron indeksu, niż to konieczne. Sytuację tę można skorygować przez ponowne utworzenie indeksu lub reorganizację indeksu.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą `runstats`.
- Na ogół skanowanie indeksu jest najbardziej efektywne, gdy kryterium spełnia tylko kilka wierszy. Aby oszacować liczbę wierszy spełniających kryterium, optymalizator korzysta z dostępnych statystyk kolumn występujących w predykatach. Jeśli niektóre wartości występują częściej od innych, należy zażądać statystyki rozkładu, korzystając z klauzuli `WITH DISTRIBUTION` komendy `runstats`. Dzięki zastosowaniu statystyki rozkładu niejednorodnego optymalizator może odróżnić wartości występujące często od wartości występujących rzadko.
- Operator `IXAND` najlepiej wykorzystuje indeksy oparte na pojedynczych kolumnach, gdyż dla jego działania bardzo duże znaczenia mają klucze rozpoczęcia i zakończenia.
- W przypadku łączy gwiazdzistych należy utworzyć indeks jednokolumnowy dla każdej z najbardziej selektywnych kolumn w tabeli faktów i pokrewnych tabelach wymiarów.

IXSCAN

Nazwa operatora: IXSCAN

Reprezentuje: Skanowanie indeksu w celu wygenerowania ograniczonego strumienia wierszy. W procesie skanowania mogą być używane opcjonalne warunki rozpoczęcia/zakończenia. Skanowane mogą być także predykaty, które można indeksować, i w których występują kolumny indeksu.

Operację tę wykonuje się w celu zawężenia zbioru wierszy spełniających kryterium przed uzyskaniem dostępu do tabeli podstawowej (w oparciu o predykaty).

Więcej informacji można znaleźć w sekcji poświęconej skanowaniu indeksów, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Z czasem w wyniku aktualizacji bazy danych może nastąpić fragmentacja indeksu powodująca powstanie większej liczby stron indeksu, niż to konieczne. Sytuację tę można skorygować przez ponowne utworzenie indeksu lub reorganizację indeksu.
- Gdy uzyskiwany jest dostęp do dwóch lub większej liczby tabel, efektywność dostępu do tabeli wewnętrznej za pośrednictwem indeksu można zwiększyć, tworząc indeks dla kolumn łączenia tabeli zewnętrznej.

Więcej wytycznych dotyczących indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain.

- Jeśli statystyki nie są aktualne, trzeba zaktualizować je za pomocą komendy `runstats`.

- Na ogół skanowanie indeksu jest najbardziej efektywne, gdy kryterium spełnia tylko kilka wierszy. Aby oszacować liczbę wierszy spełniających kryterium, optymalizator korzysta z dostępnych statystyk kolumn występujących w predykatach. Jeśli niektóre wartości występują częściej od innych, należy zażądać statystyki rozkładu, korzystając z klauzuli WITH DISTRIBUTION komendy **runstats**. Dzięki zastosowaniu statystyki rozkładu niejednorodnego optymalizator może odróżnić wartości występujące często od wartości występujących rzadko.

MSJOIN

Nazwa operatora: MSJOIN

Reprezentuje: Łączenie scalające, w którym zarówno kwalifikowane wiersze z tabeli zewnętrznej, jak i wewnętrznej muszą występować w porządku predykatu łączenia. Łączenie scalające jest także zwane *łączeniem scalającym ze skanowaniem* lub *sortowanym łączeniem scalającym*.

Łączenie jest niezbędne zawsze, gdy w klauzuli FROM występuje więcej niż jedna tabela. Łączenie scalające jest możliwe do zrealizowania, gdy istnieje predykat łączenia, który porównuje kolumny z dwóch różnych tabel. Może także powstać w wyniku ponownie wygenerowanego podzapytania.

Łączenie scalające wymaga uporządkowanych danych wejściowych w łączących się kolumnach, gdyż tabele są zwykle skanowane tylko raz. Uporządkowane dane wejściowe uzyskuje się poprzez dostęp do indeksu lub posortowanej tabeli.

Więcej informacji można znaleźć w sekcji poświęconej pojęciom związanym z łączeniem, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Należy użyć predykatów lokalnych (to znaczy zawierających odniesienia do jednej tabeli), aby ograniczyć liczbę wierszy, które mają być łączone.
Wytyczne dotyczące indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

NLJOIN

Nazwa operatora: NLJOIN

Reprezentuje: Łączenie zagnieżdżone, w którym tabela wewnętrzna jest skanowana raz (zwykle przy użyciu operatora skanowania indeksu) dla każdego wiersza tabeli zewnętrznej.

Łączenie jest niezbędne zawsze, gdy w klauzuli FROM występuje więcej niż jedna tabela. Łączenie NLJOIN nie wymaga predykatu łączenia, ale na ogół działa bardziej efektywnie, gdy predykat taki istnieje.

Łączenie zagnieżdżone wykonuje się:

- Skanując tabelę wewnętrzną dla każdego użytego wiersza tabeli zewnętrznej.
- Przeszukując indeks tabeli wewnętrznej dla każdego użytego wiersza tabeli zewnętrznej.

Więcej informacji można znaleźć w sekcji poświęconej pojęciom związanym z łączeniem, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Łączenie zagnieżdżone będzie prawdopodobnie bardziej efektywne, jeśli będzie istniał indeks dla kolumn tabeli wewnętrznej (tej po prawej stronie operatora NLJOIN) występujących w predykanie łączenia. Należy sprawdzić, czy dla tabeli wewnętrznej używany jest operator TBSCAN, a nie IXSCAN. Jeśli tak, to należy rozważyć utworzenie indeksu dla kolumn łączenia.

Inny (mniej istotny) sposób na poprawę efektywności łączenia, polega na utworzeniu indeksu dla kolumn łączenia tabeli zewnętrznej, tak aby tabela zewnętrzna była uporządkowana.

Więcej wytycznych dotyczących indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .

- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

Informacje pokrewne:

- Łączenia gwiazdziste.

PIPE

Nazwa operatora: PIPE

Reprezentuje: Przekazywanie wierszy do innych operatorów bez wprowadzania w nich żadnych zmian.

(Ten operator jest stosowany tylko w trybie debugowania.)

RETURN

Nazwa operatora: RETURN

Reprezentuje: Zwrot danych z zapytania do użytkownika. Jest to ostatni operator na wykresie planu dostępu. Przedstawia on łączne skumulowane wartości i koszty planu dostępu.

Ten operator reprezentuje operację, która musi zostać wykonana.

Wskazówka dotycząca wydajności:

- Należy sprawdzić, czy użyto predykatów pobierających tylko potrzebne dane. Na przykład należy sprawdzić, czy wartość selektywności predykatów odpowiada części tabeli, która ma zostać zwrócona.

RIDSCN

Nazwa operatora: RIDSCN

Reprezentuje: Skanowanie listy identyfikatorów wierszy (RID) uzyskanej z jednego lub kilku indeksów.

Optymalizator bierze ten operator pod uwagę, gdy:

- Predykaty są połączone parametrami OR lub istnieje predykat IN. Istnieje możliwość użycia techniki zwanej logicznym sumowaniem indeksów, polegającej na łączeniu wyników wielu dostępów do indeksu dotyczących tej samej tabeli.
- Korzystne jest zastosowanie preselekcji listy dla pojedynczego dostępu do indeksu, ponieważ sortowanie identyfikatorów wierszy przed uzyskaniem dostępu do wierszy podstawowych poprawia efektywność operacji we/wy.

RQUERY

Nazwa operatora: SHIP

Reprezentuje: Operator używany w systemie stowarzyszonym do pobierania danych ze zdalnego źródła danych. Ten operator jest uwzględniany przez optymalizator, gdy instrukcja SQL SELECT jest wysyłana przy użyciu operatora SHIP do zdalnego źródła danych w celu pobrania wyników zapytania. Instrukcja SELECT jest generowana przy użyciu dialektu SQL obsługiwanego przez źródło danych i może zawierać dowolne poprawne zapytanie dozwolone w źródle danych.

Wskazówka dotycząca wydajności: Więcej informacji można znaleźć w rozdziale 4 tomu 2 podręcznika Administration Guide, zatytułowanym Federated Database Query and Network Tuning Information.

SORT

Nazwa operatora: SORT

Reprezentuje: Sortowanie wierszy w tabeli w porządku jednej lub kilku kolumn, czemu może opcjonalnie towarzyszyć eliminacja powielonych pozycji.

Sortowanie jest niezbędne, gdy nie istnieje indeks zapewniający żądany porządek sortowania, lub gdy sortowanie jest mniej kosztowne, niż skanowanie indeksu. Sortowanie jest zwykle wykonywane jako ostatnia operacja po pobraniu wymaganych wierszy albo w celu posortowania danych przed łączeniem lub grupowaniem.

W przypadku dużej liczby wierszy albo w sytuacji, gdy nie jest możliwe utworzenie potoku posortowanych danych, operacja ta wymaga kosztownego generowania tabel tymczasowych.

Więcej informacji na temat sortowania można znaleźć w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Należy rozważyć dodanie indeksu dla kolumn określających porządek sortowania. Wtyczne dotyczące indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .
- Należy sprawdzić, czy użyto predykatów pobierających tylko potrzebne dane. Na przykład należy sprawdzić, czy wartość selektywności predykatów odpowiada części tabeli, która ma zostać zwrócona.
- Rozmiar preselekcji tymczasowego systemowego obszaru tabel powinien być odpowiedni, to znaczy nie powinien być związany z operacjami we/wy. (Aby to sprawdzić, należy wybrać **Instrukcja->Pokaż statystyki->Obszary tabel** .)
- Jeśli często zachodzi potrzeba sortowania dużych ilości danych, należy rozważyć zwiększenie wartości następujących parametrów konfiguracyjnych:
 - Rozmiaru sterty sortowania (sortheap). Aby zmodyfikować ten parametr, kliknij prawym przyciskiem myszy bazę danych w Centrum sterowania, a następnie z menu podręcznego wybierz polecenie *Konfiguruj*. W notatniku, który zostanie wyświetlony, wybierz kartę **Wydajność**.
 - Próg sterty sortowania (sheapthres). Aby zmodyfikować ten parametr, kliknij prawym przyciskiem myszy instancję bazy danych w Centrum sterowania, a następnie z menu podręcznego wybierz polecenie *Konfiguruj*. W notatniku, który zostanie wyświetlony, wybierz kartę **Wydajność**.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

TBSCAN

Nazwa operatora: TBSCAN

Reprezentuje: Skanowanie tabeli (skanowanie relacji), w wyniku którego wiersze są pobierane poprzez odczyt wszystkich wymaganych danych bezpośrednio ze stron danych.

Optymalizator wybiera ten typ skanowania zamiast skanowania indeksu, gdy:

- poszukiwane wartości należą do zakresu wartości często występujących w tabeli (to znaczy konieczny jest dostęp do większości danych w tabeli),
- tabela jest niewielka,
- stopień klastrowania indeksu jest niski,
- nie istnieje indeks.

Więcej informacji na temat skanowania tabel i indeksów można znaleźć w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Skanowanie indeksu jest bardziej efektywne, niż skanowanie tabeli, jeśli tabela jest duża i nie będzie uzyskiwany dostęp do większości jej wierszy. Aby zwiększyć prawdopodobieństwo wybrania przez optymalizator operatora skanowania indeksu w takiej właśnie sytuacji, należy rozważyć dodanie indeksów dla kolumn, dla których istnieją selektywne predykaty.

Więcej wytycznych dotyczących indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .

- Jeśli indeks już istnieje, ale nie został użyty, należy sprawdzić, czy istnieją selektywne predykaty dla wszystkich początkowych kolumn indeksu. Jeśli predykaty te faktycznie istnieją, należy sprawdzić, czy stopień klastrowania indeksu jest wysoki. (Aby wyświetlić tę statystykę, należy otworzyć okno Statystyka tabeli dla tabeli wyświetlanej poniżej sortowania i wybrać przycisk *Indeksy*, co spowoduje wywołanie okna Statystyka indeksu).
- Rozmiar preselekcji obszaru tabel powinien być odpowiedni, to znaczy nie powinien być związany z operacjami we/wy. (Aby to sprawdzić, należy wybrać **Instrukcja->Pokaż statystyki->Obszary tabel**).

Więcej informacji można znaleźć w sekcji poświęconej preselekcji danych do puli buforów w podręczniku *Administration Guide*.

- Jeśli statystyki nie są aktualne, należy je zaktualizować za pomocą komendy runstats. Statystyki kwantyli i najczęściej występujących wartości informują o selektywności predykatów. Statystyk tych można użyć np. w celu stwierdzenia, kiedy operatory skanowania indeksów będą używane zamiast operatorów skanowania tabel. Aby zaktualizować te statystyki, należy w odniesieniu do tabeli użyć komendy **runstats** z klauzulą WITH DISTRIBUTION.

TEMP

Nazwa operatora: TEMP

Reprezentuje: Czynność zapisywania danych w tabeli tymczasowej, z której mają być one odczytywane przez inny operator (możliwe, że wielokrotnie). Tabela jest usuwana po przetworzeniu instrukcji SQL, o ile nie zostanie usunięta wcześniej.

Operator ten jest niezbędny do obliczania wartości podwyrażeń lub do zapamiętywania wyników pośrednich. W niektórych sytuacjach (na przykład, gdy instrukcja może być aktualizowana), może być obowiązkowy.

TQUEUE

Nazwa operatora: TQUEUE

Reprezentuje: Kolejkę tabel używaną do przekazywania danych z tabel między agentami bazy danych w sytuacji, gdy zapytanie jest przetwarzane przez wielu agentów bazy danych. Do przetwarzania zapytania używa się wielu agentów, gdy stosowany jest mechanizm paralelizmu.

Wyróżnia się następujące typy kolejek tabel:

- **Lokalne:** Kolejki tabel używane do przekazywania danych między agentami bazy danych wewnątrz jednego węzła. Lokalne kolejki tabel stosowane są w przypadku paralelizmu wewnątrz partycji.
- **Nielokalne :** Kolejki tabel służące do przekazywania danych między agentami bazy danych w różnych węzłach.

UNION

Nazwa operatora: UNION

Reprezentuje: Konkatenację strumieni wierszy z wielu tabel.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do sklejenia.

UNIQUE

Nazwa operatora: UNIQUE

Reprezentuje: Eliminowanie wierszy, które w określonych kolumnach zawierają powielone wartości.

Wskazówka dotycząca wydajności:

- Ten operator jest zbędny tylko w sytuacji, gdy istnieje indeks unikalny obejmujący odpowiednie kolumny.

Wytyczne dotyczące indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .

UPDATE

Nazwa operatora: UPDATE

Reprezentuje: Aktualizację danych w wierszach tabeli.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do aktualizacji.

Optymalizator

Optymalizator to składnik kompilatora SQL odpowiedzialny za wybór planu dostępu dla instrukcji SQL języka DML. Wybór odbywa się na drodze modelowania kosztu wykonania wielu alternatywnych planów dostępu. Następnie wybierany jest plan, którego przewidywany koszt jest najniższy.

Pakiet

Pakiet jest przechowywanym w bazie danych obiektem zawierającym informacje potrzebne do przetwarzania instrukcji SQL skojarzonych z jednym plikiem źródłowym aplikacji. Pakiet jest generowany na jeden z następujących sposobów:

- Poprzez prekompilację pliku źródłowego przy użyciu komendy **PREP**.
- Poprzez konsolidację pliku powiązań wygenerowanego przez prekompilator za pomocą komendy **BIND**.

Predykat

Predykat jest elementem warunku wyszukiwania wyrażającym wprost lub pośrednio operację porównania. Predykaty są zawarte w klauzulach rozpoczynających od słów kluczowych WHERE lub HAVING.

Na przykład, w następującej instrukcji SQL:

```
SELECT * FROM SAMPLE
  WHERE NAME = 'SMITH' AND
  DEPT = 895 AND YEARS > 5
```

predykatami są: NAME = 'SMITH'; DEPT = 895; i YEARS > 5.

Predykaty mogą należeć do następujących kategorii, uszeregowanych od najbardziej do najmniej wydajnych:

1. Nawiasy warunków rozpoczęcia i zakończenia (zawężenie) przeglądania indeksu. (Warunki te nazywane są także predykatami ograniczającymi zakres.)

2. Predykaty strony indeksu (zwane także wyszukiwalnymi w indeksie), których wynik można określić na podstawie indeksu, ponieważ kolumny używane w predykacie są częścią klucza indeksu.
3. Predykaty strony danych (nazywane także wyszukiwalnymi w danych), których wyniku nie można określić na podstawie indeksu. Można jednak określić wynik na podstawie wierszy, tak długo, jak wiersze pozostają w buforze.
4. Pozostałe predykaty wymagają zwykle wykonywania operacji we/wy wykraczających poza prosty dostęp do tabeli podstawowej i muszą być stosowane po skopiowaniu danych poza stronę bufora. Należą do nich predykaty zawierające podzapytania lub odczytujące dane LONG VARCHAR lub LOB przechowywane w plikach niezależnie od tabeli.

Konstruując predykaty należy dążyć do jak największej możliwej selektywności, tak aby zwracanych było jak najmniej wierszy.

Poniżej wymieniono najbardziej efektywne i najczęściej stosowane predykaty:

- *Predykat łączenia na podstawie prostej równości* jest wymagany przy połączeniach scalających. Ma on często postać `tabela1.kolumna = tabela2.kolumna` i pozwala na zrównanie kolumn w dwóch różnych tabelach, umożliwiając łączenie tych tabel.
- *Predykat lokalny* jest stosowany tylko do jednej tabeli.

Więcej informacji można znaleźć w sekcjach poświęconych dostępowi do danych i optymalizacji w *Administration Guide*.

Klasa optymalizacji zapytania

Klasa optymalizacji zapytania to zbiór zasad przekształcania zapytań i technik optymalizacji używanych przy kompilacji zapytań.

Podstawowe klasy optymalizacji zapytań to:

- 1 Optymalizacja ograniczona. Przydatna w przypadku występowania poważnych ograniczeń co do rozmiaru pamięci i mocy obliczeniowej. W przybliżeniu odpowiada optymalizacji realizowanej przez wersję 1.
- 2 Optymalizacja nieznaczna. Poziom optymalizacji wyższy niż w wersji 1, ale charakteryzujący się znacznie niższym kosztem optymalizacji niż poziomy od 3 wzwyż, zwłaszcza w przypadku bardzo złożonych zapytań.
- 3 Optymalizacja umiarkowana. Bliższy odpowiednik optymalizacji zapytań charakterystycznej dla DB2 for MVS/ESA.
- 5 Optymalizacja normalna. Zalecana w środowiskach mieszanych, w których występują zarówno proste transakcje, jak i złożone zapytania.

- 7 Optymalizacja normalna. Poziom ten sam, co 5., z tym że w przypadku złożonych zapytań w dynamicznym SQL stopień optymalizacji nie jest ograniczany.

Pozostałe klasy optymalizacji zapytań, przeznaczone do stosowania wyłącznie w specjalnych warunkach, to:

- 0 Optymalizacja minimalna. Należy ją stosować tylko wówczas, gdy wymagana jest minimalna optymalizacja lub gdy optymalizacja w ogóle nie jest wymagana (to znaczy w przypadku bardzo prostych zapytań, operujących na dobrze poindeksowanych tabelach).
- 9 Optymalizacja maksymalna. Zużywa dużo pamięci i zasobów przetwarzania. Klasy tej należy używać tylko w sytuacjach, gdy klasa 5. nie jest wystarczająca (tzn. w przypadku bardzo złożonych zapytań wykonywanych przez długi czas, dla których klasa 5. nie zapewnia dobrej wydajności).

Zwykle należy używać wyższej klasy optymalizacji dla zapytań statycznych, które prawdopodobnie będą wykonywane długo, a niższą klasę optymalizacji w przypadku prostych zapytań przesyłanych dynamicznie, albo zapytań wykonywanych tylko kilkakrotnie.

Aby określić klasę optymalizacji dla dynamicznych instrukcji SQL, należy wprowadzić następującą komendę w procesorze wiersza komend:

```
SET CURRENT QUERY OPTIMIZATION = n;
```

gdzie 'n' jest odpowiednią klasą optymalizacji zapytania.

Do określania klasy optymalizacji zapytań dla statycznych instrukcji SQL służy opcja QUERYOPT komendy **BIND** lub **PREP**.

Więcej informacji można znaleźć w sekcji poświęconej regulacji klasy optymalizacji w podręczniku *Administration Guide*.

Selektywność predykatów

Termin *selektywność* odnosi się do prawdopodobieństwa, że wiersz spełni predykat (tj. predykat będzie prawdziwy).

Na przykład, selektywność 0,01 (1%) w przypadku predykatu operującego na tabeli zawierającej 1.000.000 wierszy oznacza, że predykat zwróci około 10.000 wierszy (1% z 1.000.000) i odrzuci około 990.000 (liczby prognozowane).

Wskazane jest stosowanie wysoce selektywnych predykatów (o selektywności 0,10 lub mniejszej). Takie predykaty zwracają mniejszą liczbę wierszy, na których działają dalsze operatory, a zatem w przetworzenie zapytania angażują mniej zasobów procesora i we/wy.

Przykład

Załóżmy, że mamy tabelę zawierającą 1.000.000 wierszy, i że pierwotne zapytanie zawiera klauzulę ORDER BY wymagającą dodatkowej czynności sortowania. W przypadku predykatu o selektywności 0,01, sortowanie musi być wykonywane na około 10.000 wierszy. Jednak w przypadku predykatu o selektywności równej 0,50, operacja sortowania byłaby wykonywana na 500.000 wierszy, tym samym wymagając więcej czasu procesora i operacji we/wy.

Połączenia gwiazdziste

Zbiór połączeń nazywamy gwiazdzistym, gdy tabela faktów (duża tabela centralna) jest połączona z dwiema lub większą liczbą tabel wymiarów (mniejszych tabel zawierających opisy wartości kolumn tabeli faktów).

Połączenie gwiazdziste składa się z 3 głównych części:

- Semi-połączeń
- Operacji IXAND na wynikach semi-połączeń
- Operacji kompletowania semi-połączeń.

Połączenie takie ma postać dwóch lub większej liczby połączeń będących argumentami operatora IXAND.

Semi-połączenie jest specjalną postacią połączenia, którego wynikiem jest jedynie identyfikator wiersza (RID) tabeli wewnętrznej, a nie połączenie kolumn tabeli wewnętrznej i zewnętrznej.

W połączeniach gwiazdzistych semi-połączenia są źródłem identyfikatorów wierszy dla operatora IXAND. Operator IXAND grupuje efekty filtrowania uzyskiwane przy użyciu różnych połączeń. Wynik działania operatora IXAND jest przekazywany do operatora IXOR, który porządkuje identyfikatory wierszy i eliminuje wszelkie powielone wiersze, które mogły powstać w wyniku połączeń będących argumentami operatora IXAND. Wiersze z tabeli faktów są następnie pobierane przy użyciu operatora Fetch. Wreszcie zredukowana tabela faktów jest łączona ze wszystkimi tabelami wymiarów, co powoduje, że połączenia stają się kompletne.

Wskazówki dotyczące wydajności:

- Dla każdego z połączeń tabel wymiarów należy utworzyć indeks w tabeli faktów.
- Próg sterty sortowania powinien być na tyle duży, by możliwe było przydzielenie filtru bitowego operatora IXAND. W przypadku połączeń gwiazdzistych może to wymagać użycia aż 12 MB lub 3000 stron po 4 KB. W przypadku paralelizmu wewnątrz partycji filtr bitowy jest przydzielany z tego samego segmentu pamięci współużytkowanej, co sterta dbheap, ale jest ograniczony rozmiarem sterty sortheap

(i sheapthres w ramach instancji). Ilość pamięci współużytkowanej jest zatem kontrolowana przez wartości sortheap i sheapthres, a wymagana jej ilość może przekroczyć 12 MB.

- W odniesieniu do tabel wymiarów powinny być stosowane predykaty filtrujące. Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

Statyczny SQL

Statyczna instrukcja SQL jest wbudowana w aplikację. Wszystkie wbudowane instrukcje muszą zostać poddane prekompilacji i skonsolidowane w *pakiet*, zanim możliwe będzie wykonanie aplikacji.

Podczas kompilacji w programie DB2 dla każdej instrukcji tworzony jest plan dostępu, oparty na statystykach katalogu i parametrach konfiguracji obowiązujących w czasie prekompilacji i konsolidacji instrukcji.

Te plany dostępu używane są przy każdym uruchomieniu aplikacji; nie ulegają zmianie, aż do czasu ponownej konsolidacji pakietu.

Alternatywą dla statycznego SQL jest dynamiczny SQL.

Obszary tabel SMS

W bazie danych mogą istnieć dwa typy obszarów tabel: obszary zarządzane przez system (SMS) i obszary zarządzane przez bazę danych (DMS).

Obszarem tabel SMS zarządza system operacyjny, który przechowuje dane z bazy danych w obszarze przydzielonym w chwili tworzenia obszaru tabel. Definicja obszaru tabel obejmuje listę jednej lub kilku ścieżek katalogów, w których przechowywane są te dane.

System plików zarządza przydziałem pamięci na nośnikach.

Obszary tabel SMS i DMS mogą współistnieć w tej samej bazie danych.

Obszar tabel

Dużymi bazami danych łatwiej jest zarządzać, jeśli zostaną podzielone na odrębnie zarządzane części, zwane *obszarami tabel*.

Obszar tabel pozwala przypisać dane do określonych urządzeń logicznych lub części tych urządzeń. Na przykład, tworząc tabelę można nakazać, by jej indeksy lub długie kolumny z długimi lub dużymi obiektami (LOB) były zawsze przechowywane niezależnie od pozostałych danych tabeli.

Obszar tabel może obejmować jedno lub kilka fizycznych urządzeń pamięci (kontenerów), co pozwala osiągnąć większą wydajność. Zaleca się jednak, by wszystkie urządzenia lub kontenery w ramach tego samego obszaru tabel miały podobną charakterystykę wydajności.

Obszar tabel może być zarządzany na dwa sposoby: jako obszar zarządzany przez system (SMS) albo jako obszar zarządzany przez bazę danych (DMS).

Visual Explain

Uwaga: Począwszy od wersji 6, programu Visual Explain nie można już wywołać z wiersza poleceń. Można go natomiast nadal wywoływać z poziomu różnych obiektów bazy danych w Centrum sterowania. W przypadku tej wersji w dokumentacji nadal używana jest nazwa Visual Explain.

Narzędzie graficzne Visual Explain umożliwia wyświetlanie planu dostępu wyjaśnionych instrukcji SQL w postaci wykresu. Informacji odczytanych z wykresu można użyć w celu zoptymalizowania swoich zapytań SQL pod kątem wydajności.

Wykres planu dostępu zawiera szczegółowe informacje na temat:

- tabel (i ich skojarzonych kolumn) oraz indeksów
- operatorów (takich jak operacje przeglądania tabeli, sortowania i połączenia)
- obszarów tabel i funkcji.

Narzędzia graficznego Visual Explain można także używać do:

- Przeglądania statystyk używanych w czasie optymalizacji. Następnie statystyki te można porównać z bieżącym katalogiem w celu stwierdzenia, czy ponowna konsolidacja pakietu mogłaby poprawić wydajność.
- Określania, czy dostęp do tabeli był realizowany z użyciem indeksu. Jeśli nie był, to narzędzie Visual Explain może pomóc w określeniu, które kolumny warto poddać operacji indeksowania.
- Analizowania efektów stosowania różnych technik optymalizacji poprzez porównywanie dwóch wersji wykresu planu dostępu dla zapytania - wersji przed zastosowaniem techniki i po jej zastosowaniu.
- Uzyskiwania informacji o poszczególnych operacjach w planie dostępu, w tym o całkowitym prognozowanym koszcie i liczbie zwracanych wierszy (liczności).

Dodatek B. Alfabetyczna lista operatorów narzędzia Visual Explain

CMPEXP

Nazwa operatora: CMPEXP

Reprezentuje: Obliczanie wyrażeń niezbędnych do uzyskania wyników pośrednich lub końcowych.

(Ten operator jest stosowany tylko w trybie debugowania.)

DELETE

Nazwa operatora: DELETE

Reprezentuje: Usunięcie wierszy z tabeli.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do usunięcia.

Wskazówka dotycząca wydajności:

- W przypadku usuwania wszystkich wierszy z tabeli, należy rozważyć użycie instrukcji DROP TABLE albo komendy **LOAD REPLACE**.

EISCAN

Nazwa operatora: EISCAN

Reprezentuje: Ten operator skanuje indeks zdefiniowany przez użytkownika w celu wygenerowania ograniczonego strumienia wierszy. W procesie skanowania wykorzystywanych jest wiele warunków rozpoczęcia/zakończenia z podanej przez użytkownika funkcji generującej zakres.

Operację tę wykonuje się w celu zawężenia zbioru wierszy spełniających kryterium przed uzyskaniem dostępu do tabeli podstawowej (w oparciu o predykaty).

Wskazówka dotycząca wydajności:

- Z czasem w wyniku aktualizacji bazy danych może nastąpić fragmentacja indeksu powodująca powstanie większej liczby stron indeksu, niż to konieczne. Sytuację tę można skorygować przez ponowne utworzenie indeksu lub reorganizację indeksu.

- Jeśli statystyki nie są aktualne, należy je zaktualizować za pomocą komendy `runstats`.

FETCH

Nazwa operatora: FETCH

Reprezentuje: Pobieranie kolumn z tabeli przy użyciu określonego identyfikatora wiersza (RID).

Wskazówki dotyczące wydajności:

- Należy rozbudować klucze indeksowania, tak aby obejmowały pobierane kolumny, co wyeliminuje konieczność dostępu do stron danych.
- Aby wyświetlić okno statystyki indeksu, należy odszukać indeks związany z operacją pobierania i kliknąć dwukrotnie jego węzeł. Stopień klastrowania indeksu powinien być wysoki.
- Należy zwiększyć rozmiar buforu, jeśli liczba operacji wejścia/wyjścia (I/O) związanych z operacją pobierania jest większa od liczby stron w tabeli.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą `runstats`.

Statystyki kwantyli i najczęściej występujących wartości informują o selektywności predykatów, która decyduje o tym, kiedy zamiast operatorów skanowania tabel mogą być stosowane operatory skanowania indeksów. Aby zaktualizować te statystyki, należy w odniesieniu do tabeli użyć komendy `runstats` z klauzulą `WITH DISTRIBUTION`.

FILTER

Nazwa operatora: FILTER

Reprezentuje: Zastosowanie predykatów pochodnych powodujących filtrowanie danych w oparciu o kryteria podane w tych predykatkach.

Wskazówki dotyczące wydajności:

- Należy sprawdzić, czy użyto predykatów pobierających tylko potrzebne dane. Na przykład należy sprawdzić, czy wartość selektywności predykatów odpowiada części tabeli, która ma zostać zwrócona.
- Należy upewnić się, czy wybrano klasę optymalizacji równą co najmniej 3, aby optymalizator zastąpił podzapytanie łączeniem. Jeśli nie jest to możliwe, należy spróbować ręcznie przekształcić zapytanie, w celu wyeliminowania podzapytania. Odpowiedni przykład można znaleźć w sekcji poświęconej ponownemu generowaniu zapytań przez kompilator SQL, w podręczniku *Administration Guide*.

GENROW

Nazwa operatora: GENROW

Reprezentuje: Funkcję wbudowaną, która generuje tabelę wierszy bez korzystania z danych wejściowych uzyskanych z tabel, indeksów lub operatorów.

Optymalizator może używać funkcji GENROW do generowania wierszy danych (na przykład w przypadku instrukcji INSERT lub niektórych list IN przekształcanych w łączenia).

Aby wyświetlić szacowaną statystykę tabel wygenerowanych przez funkcję GENROW, należy kliknąć dwukrotnie jej węzeł.

GRPBY

Nazwa operatora: GRPBY

Reprezentuje: Grupowanie wierszy według wspólnych wartości wyznaczonych kolumn lub funkcji. Ta operacja jest niezbędna do generowania grupy wartości lub do obliczania wartości funkcji działających na zbiorach.

Operator GRPBY może być używany także wówczas, gdy nie określono kolumn w klauzuli GROUPBY, o ile na liście SELECT występują funkcje agregujące, co wskazuje na traktowanie całej tabeli jako pojedynczej grupy przy przeprowadzaniu tej agregacji.

Wskazówki dotyczące wydajności:

- Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do grupowania.
- Aby poprawić wydajność instrukcji SELECT zawierającej pojedynczą funkcję agregującą, ale nie zawierającej klauzuli GROUP BY, należy wypróbować następujące metody:
 - W przypadku funkcji agregującej MIN(C), utworzyć indeks rosnący dla C.
 - W przypadku funkcji agregującej MAX(C), utworzyć indeks malejący dla C.

HSJOIN

Nazwa operatora: HSJOIN

Reprezentuje: Łączenie z mieszaniem, w którym kwalifikowane wiersze z tabel są mieszane, tak aby możliwe było bezpośrednie łączenie, bez wstępnego porządkowania zawartości tabel.

Łączenie jest niezbędne zawsze, gdy w klauzuli FROM występuje więcej niż jedna tabela. Łączenie mieszające jest możliwe do zrealizowania, gdy istnieje predykat łączenia, który porównuje kolumny z dwóch różnych tabel. Predykaty łączenia muszą

mieć dokładnie ten sam typ danych. Łączenia z mieszaniem mogą także powstać w wyniku ponownego wygenerowania podzapytania, podobnie jak w przypadku operatora NLJOIN.

W przypadku łączenia z mieszaniem tabele wewnętrzne nie muszą być uporządkowane. Łączenie wykonuje się poprzez skanowanie tabeli wewnętrznej łączenia z mieszaniem i wygenerowanie tabeli skanowania z przemieszanych wartości kolumn łączenia. Następnie odczytywana jest tabela zewnętrzna, przy czym wartości kolumn łączenia są mieszane i wyszukiwane w tabeli skanowania wygenerowanej dla tabeli wewnętrznej.

Więcej informacji można znaleźć w sekcji poświęconej pojęciom związanym z łączeniem, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Należy użyć predykatów lokalnych (to znaczy zawierających odniesienia do jednej tabeli), aby ograniczyć liczbę wierszy, które mają być łączone.
- Należy zwiększyć rozmiar sterty sortowania, tak aby tabela skanowania zmieściła się w pamięci.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

INSERT

Nazwa operatora: INSERT

Reprezentuje: Wstawianie wierszy do tabeli.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do wstawienia.

IXAND

Nazwa operatora: IXAND

Reprezentuje: Iloczyn logiczny AND na wynikach wielu operacji skanowania indeksów przy użyciu technik dynamicznych map bitowych. Operator ten umożliwia zastosowanie iloczynu logicznego predykatów do wielu indeksów w celu ograniczenia do minimum dostępu do tabeli podstawowej.

Operator ten jest stosowany w celu:

- zawężenia zbioru wierszy przed uzyskaniem dostępu do tabeli podstawowej,
- wykonania iloczynu logicznego AND na predykatkach stosowanych do wielu indeksów,
- wykonania iloczynu logicznego AND na wynikach łączy symbolicznych, stosowanych w łączeniach gwiazdzystych.

Wskazówki dotyczące wydajności:

- Z czasem w wyniku aktualizacji bazy danych może nastąpić fragmentacja indeksu powodująca powstanie większej liczby stron indeksu, niż to konieczne. Sytuację tę można skorygować przez ponowne utworzenie indeksu lub reorganizację indeksu.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą `runstats`.
- Na ogół skanowanie indeksu jest najbardziej efektywne, gdy kryterium spełnia tylko kilka wierszy. Aby oszacować liczbę wierszy spełniających kryterium, optymalizator korzysta z dostępnych statystyk kolumn występujących w predykatkach. Jeśli niektóre wartości występują częściej od innych, należy zażądać statystyki rozkładu, korzystając z klauzuli `WITH DISTRIBUTION` komendy `runstats`. Dzięki zastosowaniu statystyki rozkładu niejednorodnego optymalizator może odróżnić wartości występujące często od wartości występujących rzadko.
- Operator `IXAND` najlepiej wykorzystuje indeksy oparte na pojedynczych kolumnach, gdyż dla jego działania bardzo duże znaczenia mają klucze rozpoczęcia i zakończenia.
- W przypadku łączy gwiazdzistych należy utworzyć indeks jednokolumnowy dla każdej z najbardziej selektywnych kolumn w tabeli faktów i pokrewnych tabelach wymiarów.

IXSCAN

Nazwa operatora: IXSCAN

Reprezentuje: Skanowanie indeksu w celu wygenerowania ograniczonego strumienia wierszy. W procesie skanowania mogą być używane opcjonalne warunki rozpoczęcia/zakończenia. Skanowane mogą być także predykaty, które można indeksować, i w których występują kolumny indeksu.

Operację tę wykonuje się w celu zawężenia zbioru wierszy spełniających kryterium przed uzyskaniem dostępu do tabeli podstawowej (w oparciu o predykaty).

Więcej informacji można znaleźć w sekcji poświęconej skanowaniu indeksów, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Z czasem w wyniku aktualizacji bazy danych może nastąpić fragmentacja indeksu powodująca powstanie większej liczby stron indeksu, niż to konieczne. Sytuację tę można skorygować przez ponowne utworzenie indeksu lub reorganizację indeksu.
- Gdy uzyskiwany jest dostęp do dwóch lub większej liczby tabel, efektywność dostępu do tabeli wewnętrznej za pośrednictwem indeksu można zwiększyć, tworząc indeks dla kolumn łączenia tabeli zewnętrznej.

Więcej wytycznych dotyczących indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain.

- Jeśli statystyki nie są aktualne, trzeba zaktualizować je za pomocą komendy `runstats`.

- Na ogół skanowanie indeksu jest najbardziej efektywne, gdy kryterium spełnia tylko kilka wierszy. Aby oszacować liczbę wierszy spełniających kryterium, optymalizator korzysta z dostępnych statystyk kolumn występujących w predykatkach. Jeśli niektóre wartości występują częściej od innych, należy zażądać statystyki rozkładu, korzystając z klauzuli WITH DISTRIBUTION komendy **runstats**. Dzięki zastosowaniu statystyki rozkładu niejednorodnego optymalizator może odróżnić wartości występujące często od wartości występujących rzadko.

MSJOIN

Nazwa operatora: MSJOIN

Reprezentuje: Łączenie scalające, w którym zarówno kwalifikowane wiersze z tabeli zewnętrznej, jak i wewnętrznej muszą występować w porządku predykatu łączenia. Łączenie scalające jest także zwane *łączeniem scalającym ze skanowaniem* lub *sortowanym łączeniem scalającym*.

Łączenie jest niezbędne zawsze, gdy w klauzuli FROM występuje więcej niż jedna tabela. Łączenie scalające jest możliwe do zrealizowania, gdy istnieje predykat łączenia, który porównuje kolumny z dwóch różnych tabel. Może także powstać w wyniku ponownie wygenerowanego podzapytania.

Łączenie scalające wymaga uporządkowanych danych wejściowych w łączących się kolumnach, gdyż tabele są zwykle skanowane tylko raz. Uporządkowane dane wejściowe uzyskuje się poprzez dostęp do indeksu lub posortowanej tabeli.

Więcej informacji można znaleźć w sekcji poświęconej pojęciom związanym z łączeniem, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Należy użyć predykatów lokalnych (to znaczy zawierających odniesienia do jednej tabeli), aby ograniczyć liczbę wierszy, które mają być łączone.
Wytyczne dotyczące indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

NLJOIN

Nazwa operatora: NLJOIN

Reprezentuje: Łączenie zagnieżdżone, w którym tabela wewnętrzna jest skanowana raz (zwykle przy użyciu operatora skanowania indeksu) dla każdego wiersza tabeli zewnętrznej.

Łączenie jest niezbędne zawsze, gdy w klauzuli FROM występuje więcej niż jedna tabela. Łączenie NLJOIN nie wymaga predykatu łączenia, ale na ogół działa bardziej efektywnie, gdy predykat taki istnieje.

Łączenie zagnieżdżone wykonuje się:

- Skanując tabelę wewnętrzną dla każdego użytego wiersza tabeli zewnętrznej.
- Przeszukując indeks tabeli wewnętrznej dla każdego użytego wiersza tabeli zewnętrznej.

Więcej informacji można znaleźć w sekcji poświęconej pojęciom związanym z łączeniem, w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Łączenie zagnieżdżone będzie prawdopodobnie bardziej efektywne, jeśli będzie istniał indeks dla kolumn tabeli wewnętrznej (tej po prawej stronie operatora NLJOIN) występujących w predykanie łączenia. Należy sprawdzić, czy dla tabeli wewnętrznej używany jest operator TBSCAN, a nie IXSCAN. Jeśli tak, to należy rozważyć utworzenie indeksu dla kolumn łączenia.

Inny (mniej istotny) sposób na poprawę efektywności łączenia, polega na utworzeniu indeksu dla kolumn łączenia tabeli zewnętrznej, tak aby tabela zewnętrzna była uporządkowana.

Więcej wytycznych dotyczących indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .

- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

Informacje pokrewne:

- Łączenia gwiazdziste.

PIPE

Nazwa operatora: PIPE

Reprezentuje: Przekazywanie wierszy do innych operatorów bez wprowadzania w nich żadnych zmian.

(Ten operator jest stosowany tylko w trybie debugowania.)

RETURN

Nazwa operatora: RETURN

Reprezentuje: Zwrot danych z zapytania do użytkownika. Jest to ostatni operator na wykresie planu dostępu. Przedstawia on łączne skumulowane wartości i koszty planu dostępu.

Ten operator reprezentuje operację, która musi zostać wykonana.

Wskazówka dotycząca wydajności:

- Należy sprawdzić, czy użyto predykatów pobierających tylko potrzebne dane. Na przykład należy sprawdzić, czy wartość selektywności predykatów odpowiada części tabeli, która ma zostać zwrócona.

RIDSCN

Nazwa operatora: RIDSCN

Reprezentuje: Skanowanie listy identyfikatorów wierszy (RID) uzyskanej z jednego lub kilku indeksów.

Optymalizator bierze ten operator pod uwagę, gdy:

- Predykaty są połączone parametrami OR lub istnieje predykat IN. Istnieje możliwość użycia techniki zwanej logicznym sumowaniem indeksów, polegającej na łączeniu wyników wielu dostępów do indeksu dotyczących tej samej tabeli.
- Korzystne jest zastosowanie preselekcji listy dla pojedynczego dostępu do indeksu, ponieważ sortowanie identyfikatorów wierszy przed uzyskaniem dostępu do wierszy podstawowych poprawia efektywność operacji we/wy.

RQUERY

Nazwa operatora: SHIP

Reprezentuje: Operator używany w systemie stowarzyszonym do pobierania danych ze zdalnego źródła danych. Ten operator jest uwzględniany przez optymalizator, gdy instrukcja SQL SELECT jest wysyłana przy użyciu operatora SHIP do zdalnego źródła danych w celu pobrania wyników zapytania. Instrukcja SELECT jest generowana przy użyciu dialektu SQL obsługiwanego przez źródło danych i może zawierać dowolne poprawne zapytanie dozwolone w źródle danych.

Wskazówka dotycząca wydajności: Więcej informacji można znaleźć w rozdziale 4 tomu 2 podręcznika Administration Guide, zatytułowanym Federated Database Query and Network Tuning Information.

SORT

Nazwa operatora: SORT

Reprezentuje: Sortowanie wierszy w tabeli w porządku jednej lub kilku kolumn, czemu może opcjonalnie towarzyszyć eliminacja powielonych pozycji.

Sortowanie jest niezbędne, gdy nie istnieje indeks zapewniający żądany porządek sortowania, lub gdy sortowanie jest mniej kosztowne, niż skanowanie indeksu. Sortowanie jest zwykle wykonywane jako ostatnia operacja po pobraniu wymaganych wierszy albo w celu posortowania danych przed łączeniem lub grupowaniem.

W przypadku dużej liczby wierszy albo w sytuacji, gdy nie jest możliwe utworzenie potoku posortowanych danych, operacja ta wymaga kosztownego generowania tabel tymczasowych.

Więcej informacji na temat sortowania można znaleźć w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Należy rozważyć dodanie indeksu dla kolumn określających porządek sortowania. Wtyczne dotyczące indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .
- Należy sprawdzić, czy użyto predykatów pobierających tylko potrzebne dane. Na przykład należy sprawdzić, czy wartość selektywności predykatów odpowiada części tabeli, która ma zostać zwrócona.
- Rozmiar preselekcji tymczasowego systemowego obszaru tabel powinien być odpowiedni, to znaczy nie powinien być związany z operacjami we/wy. (Aby to sprawdzić, należy wybrać **Instrukcja->Pokaż statystyki->Obszary tabel .**)
- Jeśli często zachodzi potrzeba sortowania dużych ilości danych, należy rozważyć zwiększenie wartości następujących parametrów konfiguracyjnych:
 - Rozmiaru sterty sortowania (sortheap). Aby zmodyfikować ten parametr, kliknij prawym przyciskiem myszy bazę danych w Centrum sterowania, a następnie z menu podręcznego wybierz polecenie *Konfiguruj*. W notatniku, który zostanie wyświetlony, wybierz kartę **Wydajność**.
 - Próg sterty sortowania (sheapthres). Aby zmodyfikować ten parametr, kliknij prawym przyciskiem myszy instancję bazy danych w Centrum sterowania, a następnie z menu podręcznego wybierz polecenie *Konfiguruj*. W notatniku, który zostanie wyświetlony, wybierz kartę **Wydajność**.
- Jeśli statystyki nie są aktualne, należy je zaktualizować komendą runstats.

TBSCAN

Nazwa operatora: TBSCAN

Reprezentuje: Skanowanie tabeli (skanowanie relacji), w wyniku którego wiersze są pobierane poprzez odczyt wszystkich wymaganych danych bezpośrednio ze stron danych.

Optymalizator wybiera ten typ skanowania zamiast skanowania indeksu, gdy:

- poszukiwane wartości należą do zakresu wartości często występujących w tabeli (to znaczy konieczny jest dostęp do większości danych w tabeli),
- tabela jest niewielka,
- stopień klastrowania indeksu jest niski,
- nie istnieje indeks.

Więcej informacji na temat skanowania tabel i indeksów można znaleźć w podręczniku *Administration Guide*.

Wskazówki dotyczące wydajności:

- Skanowanie indeksu jest bardziej efektywne, niż skanowanie tabeli, jeśli tabela jest duża i nie będzie uzyskiwany dostęp do większości jej wierszy. Aby zwiększyć prawdopodobieństwo wybrania przez optymalizator operatora skanowania indeksu w takiej właśnie sytuacji, należy rozważyć dodanie indeksów dla kolumn, dla których istnieją selektywne predykaty.

Więcej wytycznych dotyczących indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .

- Jeśli indeks już istnieje, ale nie został użyty, należy sprawdzić, czy istnieją selektywne predykaty dla wszystkich początkowych kolumn indeksu. Jeśli predykaty te faktycznie istnieją, należy sprawdzić, czy stopień klastrowania indeksu jest wysoki. (Aby wyświetlić tę statystykę, należy otworzyć okno Statystyka tabeli dla tabeli wyświetlanej poniżej sortowania i wybrać przycisk *Indeksy*, co spowoduje wywołanie okna Statystyka indeksu).
- Rozmiar preselekcji obszaru tabel powinien być odpowiedni, to znaczy nie powinien być związany z operacjami we/wy. (Aby to sprawdzić, należy wybrać **Instrukcja->Pokaż statystyki->Obszary tabel**).

Więcej informacji można znaleźć w sekcji poświęconej preselekcji danych do puli buforów w podręczniku *Administration Guide*.

- Jeśli statystyki nie są aktualne, należy je zaktualizować za pomocą komendy `runstats`. Statystyki kwantyli i najczęściej występujących wartości informują o selektywności predykatów. Statystyk tych można użyć np. w celu stwierdzenia, kiedy operatory skanowania indeksów będą używane zamiast operatorów skanowania tabel. Aby zaktualizować te statystyki, należy w odniesieniu do tabeli użyć komendy `runstats` z klauzulą `WITH DISTRIBUTION`.

TEMP

Nazwa operatora: TEMP

Reprezentuje: Czynność zapisywania danych w tabeli tymczasowej, z której mają być one odczytywane przez inny operator (możliwe, że wielokrotnie). Tabela jest usuwana po przetworzeniu instrukcji SQL, o ile nie zostanie usunięta wcześniej.

Operator ten jest niezbędny do obliczania wartości podwyrażeń lub do zapamiętywania wyników pośrednich. W niektórych sytuacjach (na przykład, gdy instrukcja może być aktualizowana), może być obowiązkowy.

TQUEUE

Nazwa operatora: TQUEUE

Reprezentuje: Kolejkę tabel używaną do przekazywania danych z tabel między agentami bazy danych w sytuacji, gdy zapytanie jest przetwarzane przez wielu agentów bazy danych. Do przetwarzania zapytania używa się wielu agentów, gdy stosowany jest mechanizm paralelizmu.

Wyróżnia się następujące typy kolejek tabel:

- **Lokalne:** Kolejki tabel używane do przekazywania danych między agentami bazy danych wewnątrz jednego węzła. Lokalne kolejki tabel stosowane są w przypadku paralelizmu wewnątrz partycji.
- **Nielokalne :** Kolejki tabel służące do przekazywania danych między agentami bazy danych w różnych węzłach.

UNION

Nazwa operatora: UNION

Reprezentuje: Konkatenację strumieni wierszy z wielu tabel.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do sklejenia.

UNIQUE

Nazwa operatora: UNIQUE

Reprezentuje: Eliminowanie wierszy, które w określonych kolumnach zawierają powielone wartości.

Wskazówka dotycząca wydajności:

- Ten operator jest zbędny tylko w sytuacji, gdy istnieje indeks unikalny obejmujący odpowiednie kolumny.

Wytyczne dotyczące indeksów można znaleźć w pomocy ekranowej narzędzia Visual Explain w sekcji Tworzenie odpowiednich indeksów .

UPDATE

Nazwa operatora: UPDATE

Reprezentuje: Aktualizację danych w wierszach tabeli.

Ten operator reprezentuje operację, która musi zostać wykonana. Aby ograniczyć koszty planu dostępu, należy skupić się na innych operatorach (takich jak operatory skanowania i łączenia), które określają zbiór wierszy przeznaczonych do aktualizacji.

Dodatek C. Pojęcia związane z programem DB2

Bazy danych

Relacyjna baza danych reprezentuje dane w postaci zbioru tabel. Tabela składa się ze zdefiniowanego zestawu kolumn i dowolnej liczby wierszy. Dane w każdej tabeli są logicznie ze sobą powiązane. Istnieje też możliwość zdefiniowania związków między tabelami. W czasie wyświetlania danych i manipulowania nimi stosowane są zasady i operacje matematyczne zwane relacjami (niektóre z tych operacji to INSERT, SELECT i UPDATE).

Baza danych zawiera swój własny opis: oprócz danych zapisany jest w niej opis jej struktury. W bazie danych znajdują się: zestaw tabel katalogu systemowego, które opisują logiczną i fizyczną strukturę danych; plik konfiguracyjny, w którym zapisane są wartości skojarzonych z bazą danych parametrów; protokół odtwarzania, w którym rejestrowane są wykonywane transakcje i transakcje, które mogą zostać zarchiwizowane.

Baza danych może być lokalna lub zdalna. Lokalna baza danych jest fizycznie położona na aktualnie używanej stacji roboczej, a baza danych, która znajduje się na innym komputerze, to zdalna baza danych.

Schematy

Schemat to unikalny identyfikator służący do zgrupowania zestawu obiektów bazy danych (na przykład tabel, widoków, indeksów lub aliasów). Jeśli na przykład użytkownik tworzy tabelę o nazwie LISTA_PŁAC, zastosowanie schematu pozwala na uniknięcie konieczności żmudnego przeszukiwania bazy danych w celu sprawdzenia, czy tabela o takiej nazwie nie została utworzona już przez innego użytkownika. Nazwa każdego obiektu musi być unikalna tylko w obrębie określonego schematu.

Nazwy większości obiektów bazy danych składają się z dwóch części. Pierwsza część to nazwa schematu, a druga to nazwa samego obiektu. W czasie tworzenia obiektu użytkownik może przypisać ten obiekt do określonego schematu. Jeśli nie zostanie określony schemat, obiekt zostanie przypisany do schematu domyślnego, którego nazwa odpowiada zazwyczaj identyfikatorowi użytkownika, który utworzył obiekt. Na przykład z użytkownikiem o nazwisku Kowalski może być związana tabela o nazwie KOWALSKI.LISTA_PŁAC.

Schemat to także obiekt w bazie danych. Jest on tworzony w chwili utworzenia pierwszego obiektu w tym schemacie. Właścicielem schematu może być określona osoba, i ten właściciel może sterować dostępem do znajdujących się w tym schemacie danych i obiektów.

Tabele

Relacyjna baza danych reprezentuje dane w postaci zbioru tabel. Tabela składa się z danych, które logicznie uporządkowane są w kolumnach i wierszach (ogólnie zwanych rekordami).

Każda tabela ma nazwę. Każda kolumna w tabeli także ma swoją nazwę. Wiersze w tabeli nie są uporządkowane w żaden szczególny sposób, lecz mogą być pobrane w kolejności wyznaczonej wartościami w określonych kolumnach. Dane w tabeli są logicznie ze sobą powiązane. Wszystkie dane bazy danych i tabel przypisane są do obszarów tabel.

Dodatek D. Uwagi

Produktów, usług lub opcji opisywanych w tym dokumencie firma IBM nie musi oferować we wszystkich krajach. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela firmy IBM. Jakakolwiek wzmianka na temat produktu, programu lub usługi firmy IBM nie oznacza, że może być zastosowany jedynie ten produkt, ten program lub ta usługa firmy IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny, pod warunkiem, że nie narusza to praw własności intelektualnej firmy IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Używanie tego dokumentu nie daje żadnych praw do tych patentów. Wnioski o przyznanie licencji można zgłaszać na piśmie pod adresem:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Zapytania dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej firmy IBM (IBM Intellectual Property Department) lub wysłać je na piśmie na adres:

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

Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego: FIRMA INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W TAKIM STANIE, W JAKIM SIĘ ZNAJDUJE "AS IS" BEZ UDZIELANIA JAKIKOLWIEK GWARANCJI (W TYM TAKŻE RĘKOJMI), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ ORAZ PRZYDATNOŚCI DO OKREŚLONEGO CELU LUB GWARANCJI, ŻE PUBLIKACJA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy drukarskie. Informacje te są okresowo aktualizowane, a zmiany te zostaną ujęte w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Jakiegokolwiek wzmianki na temat stron internetowych nie należących do firmy IBM zostały podane jedynie dla wygody użytkownika i nie oznaczają, że firma IBM w jakikolwiek sposób firmuje te strony. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do korzystania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjodawcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie z tym opisywanym) oraz (ii) wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, uiszczenie odpowiedniej opłaty.

Licencjonowany program opisany w tym dokumencie oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary te wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów innych firm zostały uzyskane od dostawców tych produktów z opublikowanych przez nich zapowiedzi lub innych powszechnie

dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych firm należy kierować do dostawców tych produktów.

Jakiegokolwiek wzmianki na temat kierunków rozwoju firmy IBM mogą ulec zmianie lub anulowaniu bez uprzedzenia i dotyczą jedynie ogólnych celów i założeń.

Publikacja ta może zawierać przykładowe dane i raporty używane w codziennej działalności biznesowej. W celu kompleksowego zilustrowania tej działalności podane przykłady zawierają nazwy osób, firm i ich produktów. Wszystkie te nazwiska/nazwy są fikcyjne i jakakolwiek ich zbieżność z prawdziwymi nazwiskami/nazwami jest całkowicie przypadkowa.

LICENCJA NA PRAWA AUTORSKIE:

Niniejsza publikacja może zawierać przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i rozpowszechniać te programy przykładowe w dowolnej formie bez uiszczania opłat, w celu rozbudowy, użytkowania, handlowym lub w celu rozpowszechniania aplikacji zgodnych z aplikacyjnym interfejsem programowym dla tego systemu operacyjnego, dla którego napisane były programy przykładowe. Programy przykładowe nie zostały gruntownie przetestowane. Firma IBM nie może zatem gwarantować lub sugerować niezawodności, użyteczności i funkcjonalności tych programów.

Każda kopia lub dowolna część programów przykładowych, albo też dowolna praca pochodna, musi zawierać poniższą informację o prawach autorskich:

© (*nazwa_firmy_użytkownika*) (*rok*). Części niniejszego kodu pochodzą z programów przykładowych firmy IBM Corp. © Copyright IBM Corp. *_rok_lub_lata_*. Wszelkie prawa zastrzeżone.

Znaki towarowe

Następujące nazwy są znakami towarowymi firmy International Business Machines Corporation w Stanach Zjednoczonych i/lub innych krajach i zostały użyte w co najmniej jednym dokumencie z biblioteki DB2:

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

Poniższe nazwy są znakami towarowymi lub zastrzeżonymi znakami towarowymi innych firm i zostały użyte w co najmniej jednym dokumencie z biblioteki DB2:

Microsoft, Windows, Windows NT oraz logo Windows są znakami towarowymi firmy Microsoft Corporation w Stanach Zjednoczonych i/lub innych krajach.

Intel i Pentium są znakami towarowymi firmy Intel Corporation w Stanach Zjednoczonych i/lub innych krajach.

Java i wszystkie znaki towarowe związane z językiem Java są znakami towarowymi firmy Sun Microsystems, Inc. w Stanach Zjednoczonych i/lub innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i innych krajach.

Inne nazwy firm, produktów i usług mogą być znakami towarowymi lub znakami usług innych firm.

Indeks

B

- blokowanie kursora
 - definicja 50
- blokowanie wierszy
 - blokowanie kursora 50

C

- CMPEXP, operator
 - definicja 54, 71

D

- DELETE, operator
 - definicja 54, 71
- dynamiczne instrukcje SQL, tworzenie obrazów stanu wyjaśniania 3
- dynamiczny SQL
 - definicja 50

E

- EISCAN, operator
 - definicja 54, 71
- EXPLAIN.DDL, plik/komenda 1
- EXPLSNAP, opcja (komendy BIND) 4

F

- FETCH, operator
 - definicja 54, 72
- FILTER, operator
 - definicja 55, 72
- funkcje wbudowane, odczytywanie statystyki 10
- funkcje zdefiniowane przez użytkownika, odczytywanie statystyki 10
- funkcje, odczytywanie statystyki 10

G

- GENROW, funkcja
 - definicja 55, 72
- gromadzenie aktualnej statystyki tabel i indeksów 17, 35
- GRPBY, operator
 - definicja 56, 73

H

- HSJOIN, operator
 - definicja 56, 73

I

- indeksy
 - klastrowanie
 - definicja 49
- INSERT, operator
 - definicja 57, 74
- instrukcje wyjaśniane
 - definicja 52
- IXAND, operator
 - definicja 57, 74
- IXSCAN, operator
 - definicja 58, 75

K

- klasy optymalizacji zapytań
 - definicja 66
- kolumny w instrukcjach SQL, odczytywanie statystyki 10
- komenda LIST TABLES 1
- komendy, EXPLAIN.DDL 1
- komendy, LIST TABLES 1
- komendy, opcja EXPLSNAP komendy BIND 4
- komendy, SET CURRENT EXPLAIN SNAPSHOT 4
- komendy, VESAMPL.DDL 2
- kontenery
 - definicja 49
- koszt
 - definicja 49

Ł

- łączenia gwiazdziste
 - definicja 68

M

- MSJOIN, operator
 - definicja 59, 76

N

- NLJOIN, operator
 - definicja 59, 76

O

- obrazy stanu wyjaśniania
 - definicja 51
- obrazy stanu wyjaśniania dynamicznych instrukcji SQL, tworzenie 3

- obrazy stanu wyjaśniania statycznych instrukcji SQL, tworzenie 4
- obrazy stanu wyjaśniania, przykłady dla Visual Explain 2
- obrazy stanu wyjaśniania, tworzenie 1
- obrazy stanu, przykłady dla Visual Explain 2
- obszary tabel
 - definicja 69
 - DMS
 - definicja 50
- obszary tabel DMS
 - definicja 50
- obszary tabel zarządzane przez system
 - definicja 69
- obszary tabel, odczytywanie statystyki 10
- opcje wiązania, uzyskiwanie informacji 11
- operandy
 - definicja 52
- operatory
 - definicja 52
 - lista 52
- operatory, uzyskiwanie szczegółowych informacji 10
- optymalizator
 - definicja 65

P

- pakiety
 - definicja 65
- parametry konfiguracyjne, uzyskiwanie informacji 11
- PIPE, operator
 - definicja 60, 77
- plan dostępu, ulepszenie 13, 31
- plany dostępu
 - definicja 47
- pliki, EXPLAIN.DDL 1
- predykaty
 - definicja 65

R

- RETURN, operator
 - definicja 60, 77
- RIDSCN, operator
 - definicja 61, 78

S

selektywność predykatów
definicja 67

SET CURRENT EXPLAIN SNAPSHOT,
komenda 4

SHIP, operator
definicja 61, 78

SORT, operator
definicja 61, 78

statyczne instrukcje SQL, tworzenie
obrazów stanu wyjaśniania 4

statyczny SQL
definicja 69

statystyka, tabel, indeksów, funkcji
tabelowych 9

suwak powiększenia, do powiększania
fragmentów wykresów planów
dostępu 8

T

tabele wyjaśniania, tworzenie 1

TBSCAN, operator
definicja 62, 79

TEMP, operator
definicja 63, 80

TQUEUE, operator
definicja 64, 81

tworzenie dodatkowych indeksów na
kolumnach tabeli 27, 43

tworzenie indeksów na kolumnach
używanych do łączenia tabel w
zapytaniu 21, 39

U

UNION, operator
definicja 64, 81

UNIQUE, operator
definicja 64, 81

UPDATE, operator
definicja 65, 82

V

VESAMPL.DDL, komenda 2

Visual Explain
opis 70

W

wyjaśnione instrukcje
definicja 52

wyjaśnione instrukcje SQL,
wybieranie 7

wykonywanie zapytania bez indeksów i
statystyki 32

wykres planu dostępu, odczytywanie
symboli 8

wykres planu dostępu, szczegółowe
informacje o obiektach 9

wykres planu dostępu, wyświetlanie i
korzystanie z 7

wykres planu dostępu, zmiana
wyglądu 11

wykresy planów dostępu
lista użytych operatorów 52

tworzenie
definicja 47

węzły
definicja 48

Z

zapytanie bez indeksów i statystyki 14

Kontakt z firmą IBM

W celu skontaktowania się z firmą IBM w Stanach Zjednoczonych zadzwoń pod jeden z następujących numerów:

- 1-800-237-5511 - dział obsługi klienta
- 1-888-426-4343 - informacje o dostępnych usługach
- 1-800-IBM-4YOU (426-4968) - dział marketingu i sprzedaży programu DB2

W celu skontaktowania się z firmą IBM w Kanadzie zadzwoń pod jeden z następujących numerów:

- 1-800-IBM-SERV (1-800-426-7378) - dział obsługi klienta
- 1-800-465-9600 - informacje o dostępnych usługach
- 1-800-IBM-4YOU (1-800-426-4968) - dział marketingu i sprzedaży programu DB2

Krajowe lub regionalne przedstawicielstwo firmy IBM można znaleźć w serwisie WWW o nazwie Directory of Worldwide Contacts pod adresem www.ibm.com/planetwide

Informacje o produkcie

Informacje dotyczące produktów z rodziny DB2 Universal Database można uzyskać telefonicznie lub w sieci WWW pod adresem www.ibm.com/software/data/db2/udb

W tym serwisie dostępne są najnowsze informacje dotyczące biblioteki technicznej, zamawiania podręczników, oprogramowania do pobrania przez klientów, grup dyskusyjnych i pakietów poprawek, a także najświeższe wiadomości i odsyłacze do zasobów WWW.

Mieszkańcy USA, którzy chcą zamawiać produkty lub uzyskać informacje natury ogólnej mogą dzwonić pod następujące numery telefonów:

- 1-800-IBM-CALL (1-800-426-2255).
- 1-800-879-2755 - zamawianie publikacji.

Informacje o możliwościach kontaktu z firmą IBM poza Stanami Zjednoczonymi dostępne są na stronie serwisu IBM Worldwide pod adresem www.ibm.com/planetwide

IBM