

Autor: Mateusz Bieliński, numer indeksu 26752

1. Zapytanie do bazy danych można przedstawić w postaci drzewa. W takim drzewie operacje SQL (selekcja, projekcja, złączenie) są przedstawione za pomocą oznaczeń. W drzewie pogrupowane są operacje i połączone liniami. W przypadku złączenia występują także nazwy złączonych tabel.

Plan wykonania zapytania to przedstawienie operacji w kolejności ich wykonywania w danym zapytaniu. Poza operacjami są uwzględnione także metody dostępu i metody realizacji. Gdy plan wykonania zapytania nie korzysta z tymczasowych tabel jest określany jako działający w miejscu. Przy łączeniu operatorów w planie wykonania zapytania korzysta się z zasady przetwarzania potokowego, to znaczy, że wynik jednego operatora jest przekazywany na wejście drugiego.

Przykłady operacji optymalizacyjnych:

- dla zapytań gdzie występuje selekcja i złączenie wykonywanie selekcji przed złączeniem. Złączenie jest najbardziej czasochłonna operacja, więc ograniczenie liczby rekordów przekazywanych do złączenia będzie dobrym rozwiązaniem
- wykorzystywanie indeksu dla operacji selekcji
- wiązanie selekcji z iloczynem kartezjańskim
- stosowanie planu działającego w miejscu oraz przetwarzania potokowego (selekcje i projekcje)
- stosowanie klastra zamiast operatora złączenia
- staranie się przechodzenia tylko indeksów, a nie tabel
- w przypadku perspektywy zmaterializowanej – korzystanie z obliczonych wyników
- wyszukiwanie wspólnych podwyrażeń i obliczanie ich tylko raz
- wykonanie na początku projekcji, gdy nie jest możliwe działanie w miejscu – pozwoli to na zmniejszenie rozmiaru zapisywanych rekordów
- wstępne przetwarzanie pliku rekordów (sortowanie, haszowanie)
- gromadzenie i wykorzystywanie statystyk ilościowych dotyczących tabel, kolumn i indeksów
- szacowanie kosztów każdego planu wykonania zapytania i wybrania tego o najmniejszym koszcie
- zapamiętywanie planu, żeby zastosować go następnym razem w takiej samej sytuacji

2. Protokół ścisłego blokowania dwufazowego (Strict 2PL) zakłada blokowanie dostępu do obiektu w celu uniknięcia konfliktów wynikających z jednoczesnego odczytu/zapisu danych przez kilka transakcji.

Zasady protokołu:

- żeby odczytać obiekt transakcja musi uzyskać blokadę S (współdzieloną), a żeby zapisać musi uzyskać blokadę X (wyłącznie)
- jeśli jest założona blokada X, inne transakcje nie mogą założyć kolejnej blokady S lub X
- jeśli jest założona blokada S, inne transakcje nie mogą założyć kolejnej blokady X
- transakcja może ustawić się w kolejce lub zostać wycofana gdy nie może założyć blokady
- gdy transakcja się kończy zwalnianie są wszystkie blokady przez nią założone

Pierwsza faza protokołu to założenie blokad oraz dokonanie odczytu i zapisu przez transakcje.

Druga faza to wykonanie COMMIT/ROLLBACK oraz zwolnienie blokad.

3. - blokady – są zakładane przez transakcje w celu uniemożliwienia odczytu/zapisu przez inne transakcje w czasie wykonywania transakcji zakładającej blokadę
- dziennik – jest wymagany by wycofać zmiany niezakończonych transakcji lub odtworzyć bazę w przypadku awarii
- wielowersyjność – to pozwala na odczytywanie danych niezmiennych, gdy wykonywana jest transakcja
- kopia bazy danych
- zapasowa instalacja – taka sama jak główna, przejmuje jej rolę w razie awarii

4. W bazach danych występują 4 poziomy izolacji użytkowników podczas transakcji: SERIALIZABLE, REPEATABLE READS, READ COMMITTED, READ UNCOMMITTED.

SERIALIZABLE zapewnia największą izolację, w tym szeregowalność i odtwarzalność. Ten poziom wymaga, żeby były odczytywane tylko zatwierdzone zmiany oraz żeby wartości odczytane lub zmienione przez daną transakcję nie mogły być zmienione przez inne transakcje – w tym celu zakładane są blokady na odczyt i zapis do czasu zwolnienia zgodnie z protokołem Strict-2PL. Ponadto wymagane jest, żeby wynik SELECT wyliczony przez transakcję T nie zmienił się do końca jej działania – to zapewniają blokady typu S na wynik SELECT.

REPEATABLE READS – ten poziom izolacji jest podobny do SERIALIZABLE z taką różnicą, że REPEATABLE READS nie zakłada blokady typu S na wynik SELECT.

READ COMMITTED – podobnie jak w poprzednich poziomach wymagane jest odczytywanie tylko zatwierdzonych zmian, natomiast tylko zmienione wartości przez daną transakcję nie mogą być zmienione przez inne transakcje – w tym celu zakładane są blokady typu X do końca transakcji oraz blokady typu S do odczytu tylko do zakończenia instrukcji.

READ UNCOMMITTED – najmniejsza izolacja. Dana transakcja może odczytywać zatwierdzone i niezatwierdzone dane, a także nie dokonuje zmian – nie ma żadnych blokad.

Warto uwzględnić, że w przypadku Oracle występują różnice dla READ COMMITTED i SERIALIZABLE.

5. Dwu-fazowe zatwierdzanie ma miejsce w transakcjach rozproszonych w przypadku polecenia COMMIT.

Pierwsza faza to głosowanie – najpierw koordynator wysyła komunikat prepare do węzłów, a następnie węzły odpowiadają yes/no (yes - gdy węzeł jest gotowy do zatwierdzenia transakcji, no - gdy musi wycofać transakcję). Następnie w przypadku samych odpowiedzi yes koordynator zapisuje w dzienniku i rozsyła komunikat commit – zmiany zostają utrwalone. W przypadku przynajmniej jednej negatywnej odpowiedzi koordynator zapisuje i rozsyła komunikat abort – wycofanie zmian.

Druga faza to kończenie – węzły zapisują komunikat commit/abort, usuwają informacje o transakcji z pamięci RAM i wysyłają do koordynatora komunikat ack. Koordynator po

otrzymaniu komunikatów ack usuwa informacje o transakcji z pamięci RAM i zapisuje informację o zakończeniu w dzienniku.