Słowniczek pojęć bazodanowych

Krótkie, samodzielne definicje wszystkich pojęć używanych na tej półce, od A do Ż.

Skróty użyte na tej stronie (16)

Skrót	Co znaczy
ACID	Atomicity, Consistency, Isolation, Durability: cztery obietnice transakcji w bazie danych
CAP	Client Access Point: punkt dostępu klienta, czyli nazwa i adres IP roli klastra, pod którymi użytkownicy łączą się z usługą niezależnie od tego, który węzeł ją prowadzi
CSV	w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami
CTE	Common Table Expression: nazwany kawałek zapytania SQL pisany w WITH
DDL	Data Definition Language: polecenia SQL zmieniające strukturę (CREATE, ALTER, DROP)
DML	Data Manipulation Language: polecenia SQL zmieniające dane (INSERT, UPDATE, DELETE)
JSON	JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami
MSSQL	Microsoft SQL Server: silnik bazy danych Microsoftu, na nim w Zielonej stoi księgowość Optima
MVCC	Multi-Version Concurrency Control: baza trzyma kilka wersji wiersza, żeby czytający nie blokował piszącego
PESEL	Powszechny Elektroniczny System Ewidencji Ludności: jedenastocyfrowy numer identyfikacyjny osoby w Polsce, przykład danych, które filtr danych osobowych powinien wykrywać
RAM	Random Access Memory: pamięć operacyjna
RPO	Recovery Point Objective: ile danych wolno stracić przy awarii
RTO	Recovery Time Objective: jak długo wolno nie działać przy awarii
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
T-SQL	Transact-SQL: dialekt SQL używany w Microsoft SQL Server, wywodzący się z języka zapytań oryginalnego silnika Sybase
UUID	Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator

Po co to istnieje

Ta strona jest punktem wejścia, nie punktem końcowym — kiedy w innej stronie pada nieznany termin, wracamy tutaj po jedno zdanie wyjaśnienia, a nie po wykład. Definicje są tu celowo krótkie i uproszczone; pełne rozwinięcie z przykładami, poleceniami i tabelami różnic między silnikami znajduje się na stronach tematycznych, do których poniższe hasła odsyłają.

A

ACID

Skrót od czterech własności, jakie transakcja gwarantuje: atomowość, spójność, izolację i trwałość. Każda z tych czterech liter to osobna obietnica silnika wobec aplikacji, a razem opisują, dlaczego można bezpiecznie grupować kilka operacji w jedną, niepodzielną całość. Rozwinięcie: Czym jest transakcja: ACID rozłożone na cztery obietnice.

Agregacja

Zwinięcie wielu wierszy w jeden przez policzenie z nich pojedynczej wartości — sumy, średniej, licznika albo wartości skrajnej. Bez `GROUP BY` cała tabela jest jedną grupą i funkcja agregująca zwraca dokładnie jeden wiersz; z `GROUP BY` agregacja liczy się osobno w każdej grupie wierszy. Rozwinięcie: Funkcje agregujące i GROUP BY.

Algebra relacyjna

Formalny, matematyczny opis operacji wykonywanych na relacjach — selekcji (wybór wierszy), projekcji (wybór kolumn), iloczynu kartezjańskiego i ich pochodnych. SQL nie jest bezpośrednim zapisem tej algebry, ale `JOIN` odpowiada w niej złączeniu, a `CROSS JOIN` jest dosłownie iloczynem kartezjańskim.

Alias

Tymczasowa nazwa nadana tabeli albo kolumnie na czas jednego zapytania, widoczna tylko w tym zapytaniu i nigdzie zapisana w bazie. Alias skraca odwołania do długich nazw tabel, pozwala odróżnić dwa wystąpienia tej samej tabeli przy złączeniu tabeli z samą sobą i nadaje czytelną nazwę kolumnie policzonej wyrażeniem.

ALTER TABLE

Polecenie DDL zmieniające schemat relacji, czyli definicję tabeli obowiązującą dla wszystkich obecnych i przyszłych wierszy naraz — dodanie kolumny, zmianę typu, dołożenie ograniczenia. W przeciwieństwie do `INSERT` / `UPDATE` / `DELETE`, które zmieniają dane, `ALTER TABLE` bywa najbardziej ryzykowną operacją na żywej bazie, bo część jego wariantów wymaga zablokowania całej tabeli na czas trwania zmiany. Rozwinięcie: ALTER TABLE: co blokuje tabelę, a co nie.

Anomalia współbieżności

Sytuacja, w której wynik zapytania zależy od przypadkowego przeplotu operacji dwóch równolegle działających transakcji — czegoś, czego nie dałoby się zaobserwować, gdyby te transakcje wykonały się jedna po drugiej. Standard SQL nazywa formalnie trzy takie anomalie: brudny odczyt, niepowtarzalny odczyt i fantom. Rozwinięcie: Anomalie współbieżności: brudny odczyt, niepowtarzalny odczyt, fantom.

Anomalia aktualizacji

Sytuacja, w której zmiana jednego faktu wymaga poprawienia go we wszystkich miejscach, w których został powtórzony w niezormalizowanej tabeli, z ryzykiem, że jedno z nich zostanie pominięte. Jest to jeden z trzech klasycznych powodów, dla których normalizujemy dane do kolejnych postaci normalnych.

Anomalia usuwania

Sytuacja, w której usunięcie jednego faktu przypadkiem kasuje też inny, logicznie od niego niezależny fakt, bo oba są zmieszane w jednym wierszu niezormalizowanej tabeli. Klasyczny przykład to usunięcie jedynego zamówienia klienta, które razem z zamówieniem kasuje jedyny zapis mówiący, że taki klient w ogóle istnieje.

Anomalia wstawiania

Sytuacja, w której nie da się zapisać pewnego faktu bez jednoczesnego zapisania innego, niezwiązanego z nim faktu, bo oba są zmieszane w jednej, niezormalizowanej tabeli. Typowy przykład to niemożność zapisania nowego klienta, który jeszcze nie złożył żadnego zamówienia, gdy dane klienta istnieją wyłącznie jako kolumny wiersza zamówienia.

Antyzłączenie

Wzorzec `LEFT JOIN` połączony z warunkiem `WHERE kolumna_z_prawej IS NULL`, który znajduje wiersze lewej tabeli nieposiadające żadnego dopasowania po prawej stronie. Nazwa mówi wprost, o co chodzi — to złączenie, które szuka braku złączenia.

Atomowość

Pierwsza z czterech liter ACID — gwarancja, że transakcja wykona się w całości albo wcale, bez stanu pośredniego widocznego dla reszty systemu. Jeśli w środku transakcji coś zawiedzie, silnik wycofuje wszystkie jej dotychczasowe operacje, tak jakby nigdy się nie odbyły. Rozwinięcie: Czym jest transakcja: ACID rozłożone na cztery obietnice.

Atrybut

Nazwana kolumna relacji. W czystym modelu matematycznym atrybuty, podobnie jak krotki, nie mają ustalonej kolejności i odwołujemy się do nich po nazwie, ale SQL tej czystości nie zachowuje — kolumny tabeli mają zdefiniowaną kolejność zapisaną przy `CREATE TABLE`.

Autocommit

Tryb pracy połączenia, w którym każde pojedyncze polecenie SQL dostaje własną, niejawną transakcję zatwierdzaną automatycznie zaraz po wykonaniu, bez jawnego `BEGIN` i `COMMIT`. To domyślny tryb w większości narzędzi klienckich, wygodny dla pojedynczych zapytań, ale kosztowny przy wielu operacjach z rzędu. Rozwinięcie: BEGIN, COMMIT, ROLLBACK, SAVEPOINT.

AUTO_INCREMENT

Mechanizm MySQL/MariaDB generujący kolejną liczbę całkowitą dla klucza sztucznego przy każdym `INSERT`, bez potrzeby ręcznego liczenia największej dotychczasowej wartości. Odpowiednikiem w PostgreSQL jest `SERIAL` / `IDENTITY`, w MSSQL `IDENTITY`, a w Oracle sekwencja połączona z `IDENTITY` albo wyzwalaczem. Rozwinięcie: Sekwencje, IDENTITY, AUTO_INCREMENT.

Autoextend

Ustawienie pliku danych albo tablespace'u pozwalające mu automatycznie powiększyć się na dysku, gdy zapełni się jego dotychczasowy rozmiar, zamiast zwrócić błąd braku miejsca. Wygodne jako zabezpieczenie przed nagłym zatrzymaniem bazy, ale bez górnego limitu potrafi w milczeniu zapełnić cały dysk systemowy. Rozwinięcie: Rozmiar bazy: pliki, tablespace'y, autoextend.

B

Backup pełny

Kopia zapasowa obejmująca wszystkie dane bazy w danym momencie, niezależna od jakiejkolwiek wcześniejszej kopii — odtworzenie z samego backupu pełnego zawsze wystarcza, żeby odzyskać bazę do stanu z chwili jego wykonania. Jest podstawą każdego łańcucha backupów, bo backupy różnicowe i przyrostowe odnoszą się właśnie do niego. Rozwinięcie: Backup pełny, różnicowy, przyrostowy, logi.

Backup różnicowy

Kopia zapasowa obejmująca wyłącznie zmiany, jakie zaszły od ostatniego backupu pełnego, niezależnie od tego, ile backupów różnicowych powstało między nimi. Odtworzenie wymaga dokładnie dwóch kopii: ostatniego backupu pełnego i najnowszego różnicowego. Rozwinięcie: Backup pełny, różnicowy, przyrostowy, logi.

Backup przyrostowy

Kopia zapasowa obejmująca wyłącznie zmiany, jakie zaszły od ostatniego backupu — pełnego albo przyrostowego, cokolwiek było ostatnie w łańcuchu. Odtworzenie wymaga backupu pełnego i całego łańcucha kolejnych backupów przyrostowych aż do wybranego punktu, co czyni go szybszym do wykonania, ale wolniejszym i bardziej wrażliwym na uszkodzenie przy odtwarzaniu niż backup różnicowy. Rozwinięcie: Backup pełny, różnicowy, przyrostowy, logi.

Baza danych

Zorganizowany zbiór danych trzymany trwale na dysku — same dane, ich struktura i ograniczenia, bez procesów, które je obsługują. Fizycznie jest to zestaw plików: danych, indeksów i logów transakcyjnych. Rozwinięcie: Baza danych, silnik, instancja, schemat: cztery pojęcia, które się myli.

BCNF

Postać normalna Boyce'a-Codda — wersja trzeciej postaci normalnej domknięta ściślej, wymagająca, żeby dla każdej zależności funkcyjnej w tabeli kolumna po lewej stronie była kandydatem na klucz. Różnica wobec 3NF ujawnia się dopiero w tabelach z kilkoma nakładającymi

się kluczami kandydującymi, więc w codziennej pracy projektowej rzadko trzeba sięgać dalej niż 3NF. Rozwinięcie: Normalizacja do 3NF i świadoma denormalizacja.

B-drzewo

Zrównoważone drzewo wyszukiwań, w którym każdy węzeł może mieć kilkaset dzieci, a droga od korzenia do dowolnego liścia ma zawsze tę samą długość. To najczęściej używana struktura wewnętrzna indeksu — liście B-drzewa trzymają posortowane wartości kolumny razem z odsyłaczem do wiersza i są połączone w listę, co pozwala szybko obsługiwać zapytania zakresowe. Rozwinięcie: B-drzewo od środka.

Binlog

Binary log — dziennik zapisu MySQL/MariaDB rejestrujący zmiany danych w formie umożliwiającej ich odtworzenie na innym serwerze. Jest podstawą replikacji w tych silnikach oraz odtwarzania do punktu w czasie razem z backupem pełnym.

Bitmap Index Scan

Dwuetapowa metoda dostępu, w której silnik najpierw przechodzi po indeksie i buduje w pamięci mapę bitową stron tabeli zawierających pasujące wiersze, a dopiero potem czyta te strony w kolejności fizycznej. To kompromis między skanem indeksu a skanem sekwencyjnym, sprawdzający się przy średniej selektywności warunku.

Blokada

Rezerwacja obiektu bazy danych — wiersza, strony albo tabeli — przez transakcję, ograniczająca to, co inne transakcje mogą w tym samym czasie z tym obiektem zrobić. Blokady są mechanizmem, którym silnik faktycznie egzekwuje poziom izolacji zadeklarowany przez transakcję. Rozwinięcie: Blokady: współdzielone, wyłączne, eskalacja.

Blokada współdzielona

Blokada zakładana przy odczycie, którą może trzymać jednocześnie wiele transakcji na tym samym obiekcie — pozwala czytać równolegle, ale nie pozwala nikomu zapisać obiektu, dopóki wszystkie blokady współdzielone nie zostaną zwolnione. Rozwinięcie: Blokady: współdzielone, wyłączne, eskalacja.

Blokada wyłączna

Blokada zakładana przy zapisie, którą może trzymać w danym momencie tylko jedna transakcja na danym obiekcie — dopóki ją trzyma, żadna inna transakcja nie może ani zapisać, ani (w zależności od poziomu izolacji) odczytać tego obiektu. Rozwinięcie: Blokady: współdzielone, wyłączne, eskalacja.

C

CAP (twierdzenie)

Dla systemu rozproszonego, w którym dane są replikowane na kilku węzłach, twierdzenie CAP mówi, że w chwili przerwania sieci między węzłami system musi wybrać między dalszym odpowiadaniem z potencjalnie nieaktualnymi danymi a odmową odpowiedzi do czasu pewności co do spójności. Odporność na partycję sieci nie jest w praktyce opcją do odrzucenia, więc realny wybór dotyczy tego, co system robi w chwili awarii, a nie stałej deklaracji „dwóch z trzech liter”.

CHECK

Ograniczenie sprawdzające przy każdej próbie zapisu, czy wartość w kolumnie spełnia zadany warunek logiczny — na przykład że cena jest większa od zera. Silnik odrzuca wiersz naruszający `CHECK` niezależnie od tego, czy trafia on do tabeli przez aplikację, import CSV czy ręczny `INSERT`. Rozwinięcie: Ograniczenia: NOT NULL, UNIQUE, CHECK, FOREIGN KEY, DEFAULT.

Checkpoint

Punkt kontrolny, w którym silnik zapisuje na dysk wszystkie dotąd zmodyfikowane w pamięci strony danych, tak żeby po ewentualnej awarii odtwarzanie musiało cofnąć się tylko do ostatniego checkpointu, a nie do samego początku dziennika transakcyjnego. Częstsze checkpointy skracają czas odzyskiwania po awarii kosztem dodatkowego obciążenia dysku w trakcie normalnej pracy.

Ciągłość łańcucha backupów

Właściwość zestawu backupów mówiąca, że od dowolnego backupu przyrostowego albo różnicowego da się dojść z powrotem do backupu pełnego bez brakującego ogniwa. Skasowanie jednego backupu przyrostowego w środku łańcucha unieważnia wszystkie kolejne po nim, bo żaden

z nich osobno nie zawiera pełnych danych.

Część kotwicząca i część rekurencyjna

Dwa elementy, z których zawsze składa się zapytanie rekurencyjne, sklejone `UNION ALL`. Część kotwicząca uruchamia się raz i daje punkt startowy, a część rekurencyjna uruchamia się wielokrotnie, dokładając kolejny poziom wyniku, dopóki cokolwiek zwraca. Rozwinięcie: CTE: WITH i zapytania rekurencyjne.

Collation

Reguła określająca, jak silnik porównuje i sortuje wartości tekstowe — czy rozróżnia wielkość liter i jak traktuje znaki diakrytyczne przy sortowaniu alfabetycznym. Dwie bazy z tym samym kodowaniem znaków, ale różnym collation, mogą zwrócić różną kolejność tych samych nazw w `ORDER BY`.

CROSS JOIN

Złączenie bez żadnego warunku `ON`, zwracające pełny iloczyn kartezjański dwóch tabel — każdy wiersz pierwszej zestawiony z każdym wierszem drugiej. Różni się od przypadkowego, zapomnianego warunku złączenia tylko intencją: tu iloczyn kartezjański jest świadomie zamówionym narzędziem. Rozwinięcie: CROSS JOIN i SELF JOIN.

CTE

Common Table Expression — nazwany blok zapytania zdefiniowany klauzulą `WITH`, do którego reszta zapytania odwołuje się po nazwie jak do zwykłej tabeli. CTE istnieje tylko na czas jednego zapytania i nie zostaje w bazie jako osobny obiekt. Rozwinięcie: CTE: WITH i zapytania rekurencyjne .

D

Dane

Konkretne wartości zapisane w kolumnach i wierszach tabeli — treść, którą struktura bazy (schemat, typy, ograniczenia) opisuje i porządkuje, ale sama danymi nie jest. Ta sama struktura

tabeli może być wypełniona zupełnie różnymi danymi bez żadnej zmiany w definicji.

DDL

Data Definition Language — podzbiór SQL odpowiedzialny za definiowanie i zmianę struktury bazy: `CREATE`, `ALTER`, `DROP`. W przeciwieństwie do DML, polecenia DDL w większości silników zatwierdzają się natychmiast i niejawnie, nawet w środku otwartej transakcji.

Deadlock

Sytuacja, w której dwie albo więcej transakcji wzajemnie czekają na blokady trzymane przez siebie nawzajem, tworząc cykl, z którego żadna nie może się wydostać bez interwencji z zewnątrz. Silnik wykrywa taki cykl automatycznie i przerywa jedną z transakcji błędem, żeby uwolnić resztę.

Rozwinięcie: Zakleszczenie (deadlock): jak powstaje i jak je czytać z logu.

DELETE

Polecenie DML usuwające wiersze z tabeli na podstawie warunku `WHERE`, bez zmiany definicji samej tabeli. Bez `WHERE` `DELETE` usuwa wszystkie wiersze, ale — w odróżnieniu od `TRUNCATE` — robi to wiersz po wierszu, w ramach transakcji, którą da się wycofać. Rozwinięcie: DELETE vs TRUNCATE vs DROP.

Denormalizacja

Świadome cofnięcie części normalizacji — celowe wprowadzenie z powrotem redundancji w zamian za szybszy odczyt, kosztem trudniejszego utrzymania spójności danych. Nie jest błędem projektowym ani zapomnieniem o normalizacji, tylko decyzją podjętą po tym, jak w pełni znormalizowany schemat już istnieje, zwykle w odpowiedzi na zmierzony problem z wydajnością.

Rozwinięcie: Normalizacja do 3NF i świadoma denormalizacja.

Dialekt SQL

Wariant SQL właściwy dla konkretnego silnika — nadzbiór rdzenia standardu rozszerzony o funkcje i składnię, których nie ma nigdzie indziej. T-SQL, PL/pgSQL i PL/SQL to trzy różne dialekty, z których każdy zawiera więcej niż sam standard SQL. Rozwinięcie: Skąd się wziął SQL i dlaczego każdy silnik mówi nim inaczej.

Dirty read

Brudny odczyt — anomalia współbieżności, w której transakcja widzi dane zapisane przez inną transakcję, która jeszcze nie zatwierdziła swoich zmian. Jeśli ta druga transakcja wykona `ROLLBACK`, pierwsza transakcja przez chwilę operowała na danych, które nigdy nie istniały. Rozwinięcie: Anomalie współbieżności: brudny odczyt, niepowtarzalny odczyt, fantom.

DISTINCT

Deduplikacja wyniku zapytania — usunięcie wierszy, które po policzeniu wszystkich kolumn z listy `SELECT` są identyczne z którymś wcześniejszym wierszem. Działa na gotowym wyniku, po tym, jak `SELECT` policzył wszystkie kolumny, i często jest objawem błędu w warunku złączenia, a nie prawidłowym rozwiązaniem problemu. Rozwinięcie: DISTINCT: i dlaczego zwykle jest objawem, nie rozwiązaniem.

DML

Data Manipulation Language — podzbiór SQL odpowiedzialny za operowanie na danych wewnątrz istniejącej struktury: `INSERT`, `UPDATE`, `DELETE`, `SELECT` (w niektórych podziałach `SELECT` zalicza się osobno do DQL). Polecenia DML zmieniają instancję relacji, nigdy jej schemat.

Dokument

Samoopisujący się rekord, zwykle w formacie JSON albo jego binarnej odmianie, trzymający wszystkie dane jednej encji w jednej strukturze, łącznie z zagnieżdżonymi listami, bez rozbijania ich na osobne tabele połączone kluczami obcymi. Model dokumentowy sprawdza się, gdy struktura danych jest zmienna między rekordami albo gdy encja jest zawsze czytana i zapisywana w całości.

DROP

Polecenie DDL usuwające definicję tabeli razem z jej schematem relacji — usuwając ją, usuwa niejako i wszystkie wiersze, które ta definicja mogłaby kiedykolwiek zawierać. To operacja o poziom wyżej niż `TRUNCATE`, bo po `DROP` tabela przestaje istnieć jako obiekt bazy, a nie tylko traci zawartość. Rozwinięcie: DELETE vs TRUNCATE vs DROP.

Druga postać normalna (2NF)

Wymóg, żeby każda kolumna niebędąca częścią klucza zależała od całego klucza głównego, a nie tylko od jego fragmentu — dotyczy tabel z kluczem złożonym z kilku kolumn. Kolumna zależna wyłącznie od części klucza złożonego, a nie od całości, jest przykładem zależności częściowej, którą 2NF eliminuje. Rozwinięcie: Normalizacja do 3NF i świadoma denormalizacja.

Dziedzina

Zbiór dopuszczalnych wartości dla danej kolumny — w praktyce to, co nazywamy typem danych kolumny. Dziedzina ogranicza, co w ogóle może się znaleźć w danej pozycji krotki: kolumna zadeklarowana jako data nie przyjmie tekstu. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

E

Eskalacja blokad

Automatyczna zamiana wielu drobnych blokad wierszowych na jedną, obejmującą całą tabelę, gdy liczba pojedynczych blokad przekroczy próg ustalony przez silnik. Chroni pamięć silnika przed przechowywaniem milionów pojedynczych blokad, kosztem tego, że inne transakcje tracą dostęp do całej tabeli, nie tylko do kilku wierszy. Rozwinięcie: Blokady: współdzielone, wyłączne, eskalacja.

EXCEPT

Operacja zwracająca wiersze z pierwszego zapytania, które nie występują w drugim — różnica zbiorów w sensie matematycznym. Zamiana kolejności zapytań po obu stronach `EXCEPT` zmienia wynik na coś zupełnie innego. Rozwinięcie: INTERSECT i EXCEPT / MINUS.

EXISTS

Operator sprawdzający wyłącznie, czy podzapytanie zwróciło choćby jeden wiersz, bez patrzenia na to, jakie wartości ten wiersz zawiera. Silnik może przerwać wykonanie podzapytania od razu po znalezieniu pierwszego pasującego wiersza, co czyni `EXISTS` zwykle tańszym niż `IN` z dużym podzapytaniem. Rozwinięcie: Podzapytania: skalarne, wierszowe, skorelowane, EXISTS.

EXPLAIN

Polecenie zwracające plan wykonania zapytania zamiast jego wyniku — pokazuje, jakich metod dostępu, złączeń i sortowań silnik zamierza użyć, zanim jeszcze faktycznie je wykona. W większości silników wariant z dodatkowym słowem (`ANALYZE`, `ACTUAL`) uruchamia zapytanie naprawdę i porównuje szacunki optymalizatora z rzeczywistością. Rozwinięcie: Plan wykonania: jak go wywołać i jak go czytać.

F

Failover

Automatyczne albo ręczne przełączenie roli głównego serwera bazy danych na serwer zapasowy w chwili awarii tego pierwszego. Jakość failover mierzy się dwoma liczbami: ile danych może przy nim zostać utraconych i ile czasu trwa samo przełączenie. Rozwinięcie: Replikacja: synchroniczna, asynchroniczna, logiczna, fizyczna.

Fantom

Anomalia współbieżności, w której transakcja powtarza to samo zapytanie z warunkiem `WHERE` i za drugim razem widzi nowe wiersze, których nie było wcześniej, bo inna transakcja w międzyczasie je wstawiła i zatwierdziła. W odróżnieniu od niepowtarzalnego odczytu, który dotyczy zmiany wartości w już widzianym wierszu, fantom dotyczy pojawienia się całkiem nowych wierszy. Rozwinięcie: Anomalie współbieżności: brudny odczyt, niepowtarzalny odczyt, fantom.

Fillfactor

Ustawienie mówiące silnikowi, jaki procent każdej strony indeksu wypełnić danymi przy jego budowie, zostawiając resztę jako celowo puste miejsce zarezerwowane na przyszłe wstawienia. Domyślne sto procent jest najlepsze pod kątem rozmiaru indeksu zaraz po zbudowaniu, ale najgorsze pod kątem liczby podziałów stron przy kolejnych zapisach. Rozwinięcie: Fragmentacja i przebudowa indeksów.

Fragmentacja

Stan, w którym fizyczne rozmieszczenie danych indeksu albo tabeli na dysku przestaje odpowiadać ich logicznemu porządkowi — puste miejsce po usuniętych wierszach (fragmentacja wewnętrzna) albo rozjazd między logiczną kolejnością stron a ich fizycznym położeniem (fragmentacja

zewnętrzna). Oba rodzaje rosną wraz z liczbą operacji `UPDATE` i `DELETE` i wymagają okresowej przebudowy albo reorganizacji indeksu. Rozwinięcie: Fragmentacja i przebudowa indeksów.

FULL OUTER JOIN

Złączenie zewnętrzne zachowujące w wyniku wiersze bez dopasowania z obu tabel naraz, nie tylko z jednej — tam, gdzie brakuje dopasowania, kolumny drugiej strony wypełniają się `NULL`-ami symetrycznie w obie strony. Rozwinięcie: FULL OUTER JOIN.

Funkcja agregująca

Funkcja licząca jedną wartość z wielu wierszy — sumę, średnią, licznik, minimum, maksimum albo sklejony tekst. Użyta bez `GROUP BY` traktuje całą tabelę jako jedną grupę; użyta z `GROUP BY` liczy się osobno w każdej grupie. Rozwinięcie: Funkcje agregujące i GROUP BY.

Funkcja okna

Funkcja licząca wartość dla każdego wiersza na podstawie zbioru powiązanych wierszy — jego okna — bez zmniejszania liczby wierszy w wyniku, w odróżnieniu od funkcji agregującej z `GROUP BY`. Klauzula `OVER (...)` definiuje to okno: `PARTITION BY` dzieli wiersze na grupy bez ich zwijania, a `ORDER BY` wewnątrz `OVER` ustala kolejność przetwarzania. Rozwinięcie: Funkcje okna — OVER, PARTITION BY, ROW_NUMBER, RANK, LAG/LEAD.

G

GRANT

Polecenie nadające kontu bazodanowemu konkretne uprawnienie do konkretnego obiektu — odczyt tabeli, wykonanie procedury, tworzenie obiektów w schemacie. Odwrotnością `GRANT` jest `REVOKE`, odbierające wcześniej nadane uprawnienie. Rozwinięcie: Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień.

Granulacja blokady

Poziom, na jakim silnik zakłada blokadę — pojedynczy wiersz, strona danych obejmująca kilka-kilkanaście wierszy, albo cała tabela. Im drobniejsza granulacja, tym więcej transakcji może pracować na tej samej tabeli naraz bez wchodzenia sobie w drogę, kosztem większego zużycia pamięci na śledzenie pojedynczych blokad. Rozwinięcie: Blokady: współdzielone, wyłączne, eskalacja.

GROUP BY

Klauzula dzieląca wiersze zapytania na grupy według wartości w wymienionych kolumnach, tak żeby funkcje agregujące liczyły się osobno w każdej grupie, a nie dla całej tabeli naraz. Liczba grup w wyniku odpowiada liczbie różnych kombinacji wartości kolumn grupujących. Rozwinięcie: Funkcje agregujące i GROUP BY.

H

Hash join

Metoda złączenia, w której silnik buduje w pamięci tablicę haszującą z mniejszej z dwóch stron złączenia, a potem przechodzi raz po większej stronie, sprawdzając każdy jej wiersz w tej tablicy. Sprawdza się przy dużych zbiorach po obu stronach, kiedy nie ma przydatnego indeksu na kolumnie złączenia. Rozwinięcie: Skan sekwencyjny, skan indeksu, lookup — trzy sposoby dotarcia do wiersza.

HAVING

Klauzula filtrująca gotowe grupy powstałe po `GROUP BY`, w odróżnieniu od `WHERE`, który filtruje pojedyncze wiersze zanim jeszcze powstaną grupy. Warunek w `HAVING` może odwoływać się do wyniku funkcji agregującej, czego `WHERE` zrobić nie potrafi, bo w momencie jego działania żadna agregacja jeszcze się nie policzyła. Rozwinięcie: HAVING: dlaczego nie da się tego zrobić w WHERE.

Histogram rozkładu wartości

Statystyka dzieląca zakres wartości kolumny na przedziały o zbliżonej liczbie wierszy, dzięki czemu optymalizator potrafi oszacować, ile wierszy spełni warunek nierówności, nie tylko warunek równości. Bez histogramu silnik znałby tylko liczbę różnych wartości w kolumnie, nie ich rozłożenie wzdłuż zakresu. Rozwinięcie:

Statystyki i optymalizator kosztowy.

I

Iloczyn kartezjański

Zbiór wszystkich możliwych par, jakie da się utworzyć, biorąc po jednym elemencie z każdego z dwóch zbiorów. Dla dwóch tabel to każdy wiersz pierwszej zestawiony z każdym wierszem drugiej — liczba wierszy wyniku to zawsze iloczyn liczby wierszy obu tabel, stąd nazwa. Rozwinięcie: Czym jest złączenie: iloczyn kartezjański i warunek, który go ogranicza.

Index-only scan

Sposób wykonania zapytania, w którym silnik czyta wyłącznie strony indeksu i buduje z nich wynik, pomijając odczyt tabeli w ogóle. Możliwy tylko wtedy, gdy indeks jest pokrywający dla danego zapytania, czyli zawiera wszystkie potrzebne kolumny. Rozwinięcie: Indeks złożony, pokrywający, częściowy, funkcyjny.

Indeks

Osobna struktura danych, przechowywana obok tabeli, trzymająca posortowane wartości jednej lub kilku kolumn razem z odsyłaczem do miejsca, gdzie leży pełny wiersz w tabeli. Indeks nie jest kopią tabeli — jest dodatkowym, mniejszym zestawem danych zbudowanym po to, żeby dla danej wartości odnaleźć właściwy wiersz bez przeglądania reszty. Rozwinięcie: Czym jest indeks — i dlaczego to nie jest „przyspieszacz tabeli".

Indeks częściowy

Indeks obejmujący tylko podzbiór wierszy tabeli, wybrany warunkiem `WHERE` podanym już przy tworzeniu indeksu, a nie w każdym zapytaniu, które z niego korzysta. Mniejszy zakres oznacza mniejszy indeks, tańszy w utrzymaniu, bo silnik aktualizuje go tylko przy zmianach dotyczących wierszy spełniających ten warunek. Rozwinięcie: Indeks złożony, pokrywający, częściowy, funkcyjny.

Indeks funkcyjny

Indeks zbudowany nie na gołej wartości kolumny, tylko na wyniku wyrażenia policzonego z tej kolumny — silnik przechowuje w indeksie już policzone wartości wyrażenia. Pozwala zachować sargowalność zapytania, które w `WHERE` filtruje po tym samym wyrażeniu, na przykład po wielkich literach nazwy. Rozwinięcie: Indeks złożony, pokrywający, częściowy, funkcyjny.

Indeks klastrowy

Indeks, w którym liście B-drzewa nie trzymają odsyłacza do wiersza, tylko sam wiersz w całości — fizyczna kolejność danych w tabeli jest wtedy identyczna z logiczną kolejnością indeksu. Tabela może mieć co najwyżej jeden indeks klastrowy, bo dane fizycznie mogą być posortowane tylko na jeden sposób naraz. Rozwinięcie: B-drzewo od środka.

Indeks nadmiarowy

Indeks, którego zakres zastosowań w całości pokrywa inny, już istniejący indeks na tej samej tabeli — istnieje, zajmuje miejsce i płaci koszt zapisu, ale żadne zapytanie nie potrzebuje go osobno. Indeks na jednej kolumnie jest nadmiarowy, jeśli na tej samej tabeli istnieje już indeks złożony zaczynający się od tej samej kolumny.

Indeks pokrywający

Indeks, który zawiera wszystkie kolumny potrzebne zapytaniu — zarówno te w `WHERE`, jak i te w liście `SELECT` — dzięki czemu silnik odpowiada na zapytanie w całości z samego indeksu, bez ani jednego dodatkowego odczytu wiersza z tabeli. Rozwinięcie: Indeks złożony, pokrywający, częściowy, funkcyjny.

Indeks złożony

Indeks zbudowany na więcej niż jednej kolumnie naraz, w którym wartości są sortowane najpierw według pierwszej kolumny, w obrębie każdej jej wartości według drugiej, i tak dalej. Kolejność kolumn w definicji decyduje o tym, jakie zapytania w ogóle mogą z tego indeksu skorzystać — reguluje to zasada najbardziej lewego prefiksu. Rozwinięcie: Indeks złożony, pokrywający, częściowy, funkcyjny.

Information_schema

Zestaw widoków zdefiniowany w standardzie SQL, jednakowo nazwany i ustrukturyzowany w każdym silniku, który go implementuje. Daje przenośność — to samo zapytanie o listę tabel działa bez zmian na kilku różnych silnikach — kosztem tego, że szczegóły specyficzne dla danego silnika w ogóle w nim nie istnieją.

INNER JOIN

Złączenie zwracające wyłącznie te pary wierszy z dwóch tabel, dla których warunek w `ON` dał prawdę — każdy wiersz bez dopasowania po drugiej stronie znika z wyniku całkowicie. To domyślny, najczęściej używany rodzaj złączenia i punkt odniesienia dla wszystkich pozostałych.

Rozwinięcie: INNER JOIN.

INSERT

Polecenie DML dodające nowe wiersze do tabeli, bez zmiany jej schematu. Jest jedynym poleceniem DML, które tworzy wiersze — `UPDATE` i `DELETE` operują wyłącznie na wierszach, które już istnieją. Rozwinięcie: INSERT: jeden wiersz, wiele wierszy, INSERT ... SELECT.

Instancja

Uruchomiona, działająca w pamięci kopia silnika — procesy systemowe plus struktury w pamięci RAM, które razem obsługują żądania klientów. Instancję się startuje i zatrzymuje; bazę danych się otwiera i zamyka, i to rozróżnienie jest fundamentalne w Oracle, a subtelne w PostgreSQL i MySQL.

Rozwinięcie: Baza danych, silnik, instancja, schemat: cztery pojęcia, które się myli.

INTERSECT

Operacja zwracająca wyłącznie te wiersze, które pojawiają się jednocześnie w wyniku pierwszego i drugiego zapytania, porównywane po wszystkich kolumnach naraz. Wiersz musi być identyczny w obu zapytaniach, żeby trafić do wyniku. Rozwinięcie: INTERSECT i EXCEPT / MINUS.

Izolacja

Trzecia litera ACID — stopień, w jakim niezatwierdzone zmiany jednej transakcji są niewidoczne dla innych transakcji działających równolegle. Wyższy poziom izolacji chroni przed większą liczbą anomalii współbieżności, kosztem większej pracy silnika i mniejszej przepustowości pod obciążeniem. Rozwinięcie: Poziomy izolacji i co który naprawdę gwarantuje.

J

JOIN

Operacja zestawiająca wiersze jednej tabeli z wierszami drugiej, tworząc z nich szersze wiersze wynikowe złożone z kolumn obu stron. Samo słowo nie mówi jeszcze, które pary wierszy trafią do wyniku — o tym decyduje dopiero warunek złączenia podany w `ON`. Rozwinięcie: Czym jest złączenie: iloczyn kartezjański i warunek, który go ogranicza.

K

Kardynalność relacji

Liczba krotek w relacji, czyli w praktyce liczba wierszy tabeli. Nie należy mylić jej z kardynalnością relacji między tabelami (1:1, 1:N, N:M) — to inne użycie tego samego słowa.

Katalog natywny

Własny, wewnętrzny zestaw tabel systemowych danego silnika, nieopisany żadnym zewnętrznym standardem — `pg_catalog` w PostgreSQL, widoki `ALL_` / `USER_` / `DBA_` w Oracle, schemat `sys` w MSSQL. Daje dostęp do wszystkiego, co silnik o sobie wie, kosztem przenośności między silnikami.

Katalog systemowy

Zbiór tabel i widoków opisujących strukturę samej bazy — jakie ma tabele, kolumny, indeksy, użytkowników i uprawnienia. Jest tworzony i aktualizowany automatycznie przy każdej komendzie DDL, ale czytamy go zwykłym `SELECT`, dokładnie tak jak każdą inną tabelę. Rozwinięcie: Katalog systemowy: skąd baza wie, co w sobie ma.

Klucz

Minimalny zbiór kolumn, którego wartości jednoznacznie identyfikują wiersz w tabeli, tak że żadne dwa wiersze nie mogą mieć w tych kolumnach tej samej kombinacji wartości. Słowo „minimalny" jest tu istotne — jeśli po odjęciu którejś kolumny jednoznaczność wciąż się zachowuje, zbiór jeszcze

nie był kluczem w ścisłym sensie. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz główny

Ten jeden klucz kandydujący spośród kilku możliwych, który wybieramy jako oficjalny identyfikator wiersza. Silnik dodatkowo wymusza na nim `NOT NULL`, bo pusty identyfikator nie identyfikuje niczego. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz kandydujący

Superklucz, z którego nie da się odjąć żadnej kolumny bez utraty jednoznaczności — czyli superklucz minimalny. Tabela może mieć kilku kandydatów naraz, jeśli kilka różnych kolumn każda z osobna jednoznacznie identyfikuje wiersz. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz naturalny

Identyfikator zaczerpnięty z danych, które i tak istnieją w świecie rzeczywistym opisywanym przez tabelę — NIP, PESEL, adres e-mail. Nie wymaga dodatkowej kolumny wyłącznie na potrzeby bazy, ale bywa kosztowny w utrzymaniu, bo dane ze świata rzeczywistego potrafią się zmieniać. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz obcy

Kolumna albo zestaw kolumn w jednej tabeli, którego wartości muszą odpowiadać wartościom klucza w innej tabeli, albo być puste, jeśli kolumna dopuszcza `NULL`. To mechanizm, który fizycznie realizuje relację między tabelami. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz sztuczny

Identyfikator wygenerowany przez sam system, bez żadnego znaczenia biznesowego — liczba z sekwencji, `IDENTITY`, `AUTO_INCREMENT` albo UUID. Zaletą jest stabilność wartości niezależnej od zmian w danych źródłowych, wadą to, że sam w sobie nic nie mówi o wierszu i nie chroni przed duplikatem biznesowym. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz unikalny

Ograniczenie mówiące, że wartości w danej kolumnie się nie powtarzają, bez pretensji do bycia głównym identyfikatorem wiersza. Tabela może mieć wiele ograniczeń unikalności naraz, a w większości silników taka kolumna dopuszcza wielokrotne wystąpienie `NULL`. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

Klucz złożony

Klucz zbudowany z więcej niż jednej kolumny naraz, gdzie żadna z kolumn osobno nie jest unikalna, ale ich kombinacja już tak. Typowym przykładem jest klucz główny tabeli łączącej, złożony z dwóch kolumn wskazujących po jednym wierszu z każdej z dwóch łączonych tabel.

Kolumna

Nazwana pozycja w wierszu tabeli, ograniczona zadeklarowanym typem danych i dziedziną wartości, jakie może przyjąć. Kolumny mają ustaloną kolejność w definicji tabeli, w odróżnieniu od czysto matematycznego pojęcia atrybutu. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

Krotka

Pojedynczy element relacji, w praktyce jeden wiersz tabeli — uporządkowany zestaw wartości, po jednej z każdej dziedziny relacji. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

L

LEFT JOIN

Złączenie zewnętrzne zachowujące w wyniku wszystkie wiersze lewej tabeli, niezależnie od tego, czy znalazły dopasowanie w prawej — brakujące kolumny prawej strony wypełniają się `NULL`-ami. Rozwinięcie: LEFT JOIN i RIGHT JOIN.

LIMIT

Klauzula ograniczająca liczbę wierszy zwracanych przez zapytanie do zadanej liczby, stosowana logicznie na samym końcu, po sortowaniu. Każdy silnik ma własną składnię tego samego pomysłu:

`LIMIT` w PostgreSQL i MySQL, `TOP` w MSSQL, `ROWNUM`/`FETCH FIRST` w Oracle. Rozwinięcie: ORDER BY, LIMIT/TOP/ROWNUM/FETCH — porcjowanie wyniku.

Liczba zmiennoprzecinkowa

Typ przechowujący liczby z ułamkiem w postaci przybliżonej, w ograniczonej precyzji binarnej — szybki w obliczeniach, ale niezdolny do dokładnego zapisania wielu wartości dziesiętnych. Typ dokładny dziesiętny przechowuje cyfry wprost, bez przybliżenia, kosztem nieco wolniejszych obliczeń.

Liść (B-drzewa)

Najniższy poziom B-drzewa, na którym leżą właściwe wartości indeksowanej kolumny razem ze wskaźnikiem do wiersza tabeli, albo — w indeksie klastrowym — razem z samym wierszem. Liście są połączone między sobą w listę dwukierunkową, co pozwala drzewu obsługiwać zakresy, nie tylko pojedyncze wartości. Rozwinięcie: B-drzewo od środka.

Lock_timeout

Limit czasu, przez jaki sesja czeka na uzyskanie blokady, zanim silnik przerwie oczekiwanie i zwróci błąd zamiast czekać w nieskończoność. Zamienia ryzyko zablokowania całej kolejki zapytań w kontrolowany błąd, który da się obsłużyć i spróbować ponownie w spokojniejszym momencie.

Logiczna kolejność wykonania

Ustalona przez standard SQL kolejność, w jakiej silnik logicznie przetwarza klauzule zapytania, niezależnie od tego, w jakiej kolejności fizycznie odczytuje dane. To model do rozumowania o zapytaniu — optymalizator wolno mu zmienić kolejność faktycznych operacji, o ile wynik końcowy wyjdzie taki, jakby trzymał się modelu. Rozwinięcie: SELECT: kolejność zapisu a kolejność wykonania.

Lookup

Doskok do konkretnego wiersza w tabeli po znalezieniu jego pozycji w indeksie — drugi krok skanu indeksu, następujący po nawigacji po B-drzewie. Koszt lookupu rośnie z liczbą pasujących wierszy, bo każdy z nich to osobny, rozproszony odczyt strony tabeli. Rozwinięcie: Skan sekwencyjny, skan indeksu, lookup — trzy sposoby dotarcia do wiersza.

M

Magazyn klucz-wartość

Najprostszy z modeli danych — baza to w istocie ogromna mapa, gdzie każdemu kluczowi odpowiada jedna wartość, odczytywana i zapisywana wyłącznie po tym kluczu, bez języka zapytań w stylu SQL. Sprawdza się jako warstwa bardzo szybkiego dostępu do danych o prostej strukturze — cache, sesje, liczniki — nie jako główne źródło prawdy dla danych o złożonych powiązaniach.

Magazyn szeroko-kolumnowy

Model, w którym dane grupowane są wierszami zidentyfikowanymi kluczem, ale każdy wiersz może mieć inny zestaw kolumn, a dane fizycznie są przechowywane kolumna po kolumnie, nie wiersz po wierszu. Sprawdza się przy bardzo dużym wolumenie zapisów rozłożonych na wiele serwerów naraz, kosztem złożonych zapytań łączących wiele encji.

MERGE

Polecenie łączące w jednym poleceniu logikę wstawienia i aktualizacji — wstawia nowy wiersz, jeśli pasujący jeszcze nie istnieje, albo aktualizuje istniejący, jeśli już jest. Standard SQL definiuje `MERGE`, ale nie każdy silnik go implementuje pod tą samą nazwą i z tą samą składnią. Rozwinięcie: MERGE / UPSERT — jak to robi każdy silnik.

Merge Join

Metoda złączenia, w której obie strony są posortowane po kolumnie złączenia, a silnik przesuwa się po obu jednocześnie, zestawiając kolejne pasujące wartości bez budowania żadnej dodatkowej struktury pomocniczej. Sprawdza się, kiedy obie strony i tak są już posortowane albo kiedy wynik zapytania sam musi wyjść posortowany po tej kolumnie.

Metoda dostępu

Konkretny sposób, w jaki silnik fizycznie dociera do wierszy tabeli potrzebnych do policzenia wyniku zapytania — pełne przejście po tabeli, nawigacja po indeksie, albo połączenie kilku technik naraz. To samo zapytanie logiczne nie narzuca jednej metody dostępu — wybiera ją optymalizator na podstawie statystyk. Rozwinięcie: Plan wykonania: jak go wywołać i jak go czytać.

Migawka

Spójny widok danych bazy z konkretnego momentu w czasie, wykorzystywany zarówno przez mechanizm MVCC do obsługi współbieżności, jak i przez odtwarzanie do punktu w czasie. Każda transakcja w momencie swojego rozpoczęcia dostaje właśnie taką migawkę i widzi dane takimi, jakie były zatwierdzone w chwili jej powstania. Rozwinięcie: MVCC: dlaczego czytający nie blokuje piszącego.

Model danych

Sposób, w jaki system organizuje dane koncepcyjnie i w jaki się je odpytuje — jako tabele powiązane kluczami, jako samodzielne dokumenty, jako pary klucz-wartość. Wybór modelu danych determinuje, jakie pytania da się zadać łatwo, a jakie trudno albo wcale.

MVCC

Multi-Version Concurrency Control — mechanizm, w którym silnik nie zmienia wiersza w miejscu przy każdym `UPDATE`, tylko tworzy jego nową wersję obok starej. Dzięki temu odczyt nie musi czekać na blokadę trzymaną przez piszącego, bo sięga po wersję sprzed rozpoczęcia zapisu, która wciąż istnieje osobno. Rozwinięcie: MVCC: dlaczego czytający nie blokuje piszącego.

N

Nested loop

Metoda złączenia, w której dla każdego wiersza tabeli zewnętrznej silnik osobno przeszukuje tabelę wewnętrzną w poszukiwaniu dopasowań. Sprawdza się najlepiej, kiedy tabela zewnętrzna ma niewiele wierszy, a tabela wewnętrzna ma indeks na kolumnie złączenia. Rozwinięcie: Skan sekwencyjny, skan indeksu, lookup — trzy sposoby dotarcia do wiersza.

Nieokreślona kolejność

Właściwość tabeli relacyjnej wynikająca z tego, że jest ona zbiorem wierszy, nie listą — a zbiór z definicji nie ma kolejności. Fizyczna kolejność wierszy na dysku może się zmienić z dowolnego powodu technicznego, więc zapytanie bez `ORDER BY` nigdy nie ma gwarancji, w jakiej kolejności zwróci wiersze.

Normalizacja

Proces przekształcania struktury tabel tak, żeby każdy pojedynczy fakt był zapisany w bazie dokładnie w jednym miejscu, a nie powtarzany w wielu wierszach naraz. Eliminuje trzy klasyczne anomalie — wstawiania, aktualizacji i usuwania — które pojawiają się w jednej, szerokiej, niezormalizowanej tabeli. Rozwinięcie: Normalizacja do 3NF i świadoma denormalizacja.

NULL

Znacznik braku informacji w kolumnie — nie liczba zero, nie pusty ciąg znaków, tylko stwierdzenie „tej wartości tu nie ma albo nie jest znana". To rozróżnienie ma konsekwencje w każdym miejscu, gdzie `NULL` wchodzi w interakcję z logiką, arytmetyką albo porównaniem. Rozwinięcie: NULL: trzecia wartość logiczna i pułapki, które z niej wynikają.

O

Ograniczenie

Reguła zapisana raz w definicji tabeli, której silnik pilnuje przy każdej próbie zapisu, niezależnie od tego, przez jaką drogę dane trafiają do tabeli. Zamiast polegać na tym, że każda aplikacja i każdy skrypt migracyjny sam zadba o poprawność danych, silnik sprawdza ją na poziomie samego zapisu. Rozwinięcie: Ograniczenia: NOT NULL, UNIQUE, CHECK, FOREIGN KEY, DEFAULT.

Ograniczenie odroczone

Ograniczenie sprawdzane nie od razu po każdym poleceniu, tylko dopiero przy `COMMIT` całej transakcji. Przydatne, gdy dwie tabele odwołują się do siebie nawzajem albo gdy w ramach jednej transakcji trzeba chwilowo przestawić kolejność wierszy tak, że stan pośredni narusza regułę, ale stan końcowy już nie.

Online DDL

Zestaw technik pozwalających zmienić strukturę tabeli bez blokowania jej na czas trwania całej operacji — albo od razu jako operację czysto metadanową, albo przez zbudowanie nowej wersji tabeli w tle i podmienienie jej na końcu w jednej krótkiej blokadzie. Rozwinięcie: ALTER TABLE: co

blokuje tabelę, a co nie.

ON DELETE CASCADE

Klauzula klucza obcego mówiąca, że usunięcie wiersza rodzica ma automatycznie usunąć wszystkie wiersze potomne odwołujące się do niego, zamiast zwracać błąd naruszenia klucza obcego.

Rozwinięcie: Ograniczenia: NOT NULL, UNIQUE, CHECK, FOREIGN KEY, DEFAULT.

Optymalizator kosztowy

Komponent silnika, który dla danego zapytania rozważa kilka alternatywnych planów wykonania, szacuje koszt każdego z nich na podstawie zapamiętanych statystyk o danych i wybiera plan o najniższym szacowanym koszcie. To on decyduje między skanem sekwencyjnym a skanem indeksu, między złączeniem hash a nested loop. Rozwinięcie: Statystyki i optymalizator kosztowy.

Optymalizator regułowy

Starszy model, w którym silnik wybiera plan według ustalonej z góry hierarchii preferencji, bez patrzenia na to, ile faktycznie wierszy dany warunek zwróci. Oracle wycofał go formalnie w wersji 10g na rzecz optymalizatora kosztowego, a reguła nie odróżnia warunku zwracającego jeden wiersz od warunku zwracającego milion. Rozwinięcie: Statystyki i optymalizator kosztowy.

ORDER BY

Klauzula sortująca wynik zapytania według wskazanych kolumn albo wyrażeń, wykonywana logicznie jako jeden z ostatnich kroków, po `SELECT` i przed `LIMIT`. Bez `ORDER BY` kolejność wierszy w wyniku nie jest niczym gwarantowana. Rozwinięcie: ORDER BY, LIMIT/TOP/ROWNUM/FETCH — porcjowanie wyniku.

P

Paginacja kluczowa

Alternatywa dla stronicowania przez `OFFSET`, w której kolejna strona nie liczy się od początku wyniku, tylko od ostatniego wiersza poprzedniej strony — po angielsku keyset pagination albo

metoda seek. Unika kosztu rosnącego wraz z numerem strony, jaki płaci `OFFSET`, bo silnik nie musi przeliczać i odrzucać wszystkich wcześniejszych wierszy.

Partycjonowanie

Podział jednej logicznej tabeli na kilka fizycznie osobnych fragmentów, zwykle według zakresu wartości kolumny (na przykład daty) albo według listy wartości, tak żeby zapytanie dotyczące jednego fragmentu mogło pominąć resztę bez ich odczytu. Aplikacja i tak widzi jedną tabelę — podział jest szczegółem fizycznym, niewidocznym w zwykłym `SELECT`.

Pierwsza postać normalna (1NF)

Wymóg, żeby każda kolumna w każdym wierszu przechowywała dokładnie jedną, niepodzielną wartość, bez list i grup powtarzalnych upakowanych w jedno pole. Kolumna trzymająca kilka pozycji sklejonych w jeden tekst łamie ten wymóg, bo baza nie potrafi ani przefiltrować po pojedynczej pozycji, ani jej zsumować bez wcześniejszego rozbicia. Rozwinięcie: Normalizacja do 3NF i świadoma denormalizacja.

PITR

Point-In-Time Recovery, odtwarzanie do punktu w czasie — odtworzenie bazy nie do stanu ostatniego backupu, tylko do dowolnej sekundy między backupem pełnym a chwilą awarii, przez odtworzenie backupu i doegranie zarejestrowanych po nim logów transakcyjnych. Rozwinięcie: Odtwarzanie do punktu w czasie (PITR).

Plan wykonania

Drzewo operacji, które optymalizator wybrał jako sposób obliczenia wyniku konkretnego zapytania — każdy węzeł to jedna operacja, z liśćmi reprezentującymi dostęp do surowych danych i korzeniem reprezentującym ostateczny wynik. Ten sam SQL da inny plan zależnie od tego, jakie indeksy istnieją i jakie statystyki silnik ma o danych. Rozwinięcie: Plan wykonania: jak go wywołać i jak go czytać.

Podzapytanie

Pełne zapytanie `SELECT` umieszczone wewnątrz innego zapytania zamiast wewnątrz osobnego polecenia. Miejsce, w którym podzapytanie stoi, decyduje o tym, co wolno mu zwrócić — w prostym warunku `WHERE` musi być skalarne, w `FROM` staje się tabelą pochodną, a przy `IN` czy `EXISTS` może

zwracać wiele wierszy. Rozwinięcie: Podzapytania: skalarne, wierszowe, skorelowane, EXISTS.

Podzapytanie skalarne

Podzapytanie, które zwraca dokładnie jedną wartość — jeden wiersz, jedna kolumna. Tylko takie podzapytanie wolno postawić po jednej stronie zwykłego operatora porównania, bo tylko wtedy porównanie ma jednoznaczny sens. Rozwinięcie: Podzapytania: skalarne, wierszowe, skorelowane, EXISTS.

Podzapytanie skorelowane

Podzapytanie, które odwołuje się do kolumny zapytania zewnętrznego, przez co nie da się go policzyć raz z góry. Silnik logicznie wykonuje je od nowa dla każdego wiersza zewnętrznego zapytania, podstawiając za każdym razem inną wartość. Rozwinięcie: Podzapytania: skalarne, wierszowe, skorelowane, EXISTS.

Poziom izolacji

Ustawienie transakcji określające, na które anomalie współbieżności transakcja pozostaje podatna, a przed którymi jest chroniona. Im wyższy poziom izolacji, tym mniej anomalii transakcja może zaobserwować, ale tym więcej pracy silnik musi wykonać, żeby tę ochronę zapewnić. Rozwinięcie: Poziomy izolacji i co który naprawdę gwarantuje.

Predykat

Wyrażenie logiczne policzone osobno dla każdego wiersza, które zwraca prawdę, fałsz albo — gdy w grę wchodzi `NULL` — wartość nieznaną. `WHERE` zostawia w wyniku wyłącznie wiersze, dla których predykat dał prawdę. Rozwinięcie: WHERE — filtrowanie wierszy, operatory, LIKE, IN, BETWEEN.

PRIMARY KEY

Deklaracja w `CREATE TABLE` wybierająca jedną kolumnę albo zestaw kolumn jako klucz główny tabeli, z automatycznie wymuszonym `NOT NULL` i unikalnością. Tabela relacyjna może mieć co najwyżej jeden `PRIMARY KEY`. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

R

Ramka okna

Podzbiór wierszy w obrębie partycji, na którym funkcja okna faktycznie liczy swój wynik dla danego wiersza. Domyślnie, przy obecności `ORDER BY` w `OVER`, ramka sięga od początku partycji do bieżącego wiersza włącznie, co tłumaczy zachowanie sumy narastającej. Rozwinięcie: Funkcje okna — OVER, PARTITION BY, ROW_NUMBER, RANK, LAG/LEAD.

Relacja

W sensie matematycznym relacja to zbiór krotek o jednakowej strukturze — dokładnie to, co w praktyce nazywamy tabelą. To samo słowo wraca też w innym znaczeniu, jako relacja między tabelami, i to jest inne pojęcie. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

Relacja między tabelami

Powiązanie logiczne mówiące, jak wiersze jednej tabeli odnoszą się do wierszy drugiej. Kluczową własnością takiej relacji jest jej kardynalność — jeden do jednego, jeden do wielu albo wiele do wielu — i to ona decyduje, jak relację zaprojektować. Rozwinięcie: Relacje: 1:1, 1:N, N:M i tabela łącząca.

Replikacja asynchroniczna

Tryb replikacji, w którym transakcja na serwerze głównym zatwierdza się od razu, bez czekania na potwierdzenie, że zmiana dotarła do serwera zapasowego. Daje najlepszą wydajność zapisu, ale w razie awarii serwera głównego kilka ostatnich transakcji może zostać utraconych. Rozwinięcie: Replikacja: synchroniczna, asynchroniczna, logiczna, fizyczna.

Replikacja fizyczna

Replikacja przenosząca na serwer zapasowy identyczną, bajt w bajt, kopię plików danych, budowaną z tego samego dziennika zapisu z wyprzedzeniem, którego używa serwer główny do własnego odzyskiwania po awarii. Serwer zapasowy w tym trybie musi mieć tę samą wersję silnika i zwykle tę samą architekturę. Rozwinięcie: Replikacja: synchroniczna, asynchroniczna, logiczna,

fizyczna.

Replikacja logiczna

Replikacja przenosząca na serwer docelowy nie surowe bloki dysku, tylko logiczny opis zmian — które wiersze zostały wstawione, zaktualizowane albo usunięte — co pozwala replikować tylko wybrane tabele i między różnymi wersjami, a nawet różnymi silnikami. Rozwinięcie: Replikacja: synchroniczna, asynchroniczna, logiczna, fizyczna.

Replikacja synchroniczna

Tryb replikacji, w którym transakcja na serwerze głównym czeka na potwierdzenie zapisu ze strony serwera zapasowego, zanim sama zgłosi zatwierdzenie aplikacji. Gwarantuje brak utraty danych w razie awarii serwera głównego, kosztem wolniejszego zatwierdzania każdej transakcji. Rozwinięcie: Replikacja: synchroniczna, asynchroniczna, logiczna, fizyczna.

REVOKE

Polecenie odbierające kontu bazodanowemu wcześniej nadane uprawnienie do obiektu. Jest dokładnym odwróceniem `GRANT` i, tak jak on, nie wymaga usuwania ani odtwarzania samego konta. Rozwinięcie: Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień.

ROLLBACK

Polecenie wycofujące wszystkie zmiany wykonane w otwartej transakcji, tak jakby nigdy się nie odbyły. W połączeniu z `SAVEPOINT` pozwala wycofać tylko część transakcji, zamiast całej. Rozwinięcie: BEGIN, COMMIT, ROLLBACK, SAVEPOINT.

ROW_NUMBER

Funkcja okna nadająca każdemu wierszowi w obrębie partycji kolejny, unikalny numer, zgodnie z kolejnością wskazaną w `ORDER BY` wewnątrz `OVER`. Klasyczne zastosowanie to znalezienie dokładnie jednego, najnowszego wiersza dla każdej wartości klucza grupującego. Rozwinięcie: Funkcje okna — OVER, PARTITION BY, ROW_NUMBER, RANK, LAG/LEAD.

RPO

Recovery Point Objective — maksymalna ilość danych, wyrażona czasem, jaką organizacja akceptuje utracić w razie awarii. RPO równe piętnastu minutom znaczy, że backupy i logi muszą być na tyle częste, żeby żadna awaria nie kosztowała więcej niż piętnaście minut niezapisanych zmian. Rozwinięcie: RPO i RTO: dwie liczby, od których zaczyna się każdy plan backupu.

RTO

Recovery Time Objective — maksymalny czas, w jakim system musi zostać przywrócony do działania po awarii, liczony od chwili jej wystąpienia do chwili, gdy aplikacje znów mogą z bazy korzystać. RTO determinuje, czy wystarczy zwykłe odtworzenie z backupu, czy potrzebny jest gotowy serwer zapasowy z replikacją. Rozwinięcie: RPO i RTO: dwie liczby, od których zaczyna się każdy plan backupu.

S

Sargowalność

Właściwość warunku w `WHERE` mówiąca, czy optymalizator może go wykorzystać do przeszukania indeksu, zamiast czytać każdy wiersz tabeli po kolei. Warunek przestaje być sargowalny, kiedy kolumna indeksowana trafia pod funkcję albo wyrażenie po lewej stronie porównania. Rozwinięcie: WHERE — filtrowanie wierszy, operatory, LIKE, IN, BETWEEN.

Savepoint

Nazwany punkt pośredni wewnątrz otwartej transakcji, do którego można wycofać się poleceniem `ROLLBACK TO`, nie tracąc przy tym zmian wykonanych przed tym punktem. Pozwala obsłużyć błąd w środku długiej transakcji bez wycofywania całej pracy od początku. Rozwinięcie: BEGIN, COMMIT, ROLLBACK, SAVEPOINT.

Schema-on-read

Podejście, w którym silnik przy zapisie nie narzuca żadnej konkretnej struktury, a to, czy zapisane dane są zgodne z oczekiwaniami, ujawnia się dopiero w momencie, gdy aplikacja próbuje odczytać konkretne pole. Tak działa MongoDB — dwa dokumenty w tej samej kolekcji mogą mieć zupełnie inny zestaw pól.

Schema-on-write

Podejście, w którym silnik wymusza zgodność zapisywanych danych ze strukturą tabeli w chwili zapisu — niezgodny `INSERT` jest odrzucany, zanim dane trafią na dysk. Tak działa każdy silnik relacyjny.

Schemat

Przestrzeń nazw wewnątrz bazy danych, która grupuje tabele, widoki i inne obiekty, żeby dwie tabele o tej samej nazwie mogły istnieć obok siebie bez konfliktu. Schemat nie jest osobnym zbiorem plików — to etykieta na obiektach wewnątrz jednej bazy. Rozwinięcie: Baza danych, silnik, instancja, schemat: cztery pojęcia, które się myli.

Schemat relacji

Nazwa relacji razem z listą jej atrybutów i ich dziedzin — to, co zapisujemy w `CREATE TABLE`, czyli definicja tabeli, zanim jeszcze trafi do niej choćby jeden wiersz. Odróżnia się od instancji relacji, czyli faktycznego zestawu krotek znajdujących się w tabeli w danym momencie. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

Selektywność

Stosunek liczby różnych wartości, jakie przyjmuje kolumna, do liczby wierszy tabeli. Im wyższa selektywność, tym mniejszy odsetek wierszy przypada przeciętnie na jedną wartość, i tym więcej silnik zyskuje, mogąc odnaleźć tylko te wiersze zamiast przeglądać wszystkie. Rozwinięcie: Czym jest indeks — i dlaczego to nie jest „przyspieszacz tabeli".

Sekwencja

Obiekt bazy generujący kolejne, niepowtarzalne liczby całkowite, niezależny od żadnej konkretnej tabeli ani kolumny. Jest niesilnikowo-specyficznym, przenośnym mechanizmem stojącym za `IDENTITY` i podobnymi klauzulami generującymi wartości klucza sztucznego. Rozwinięcie: Sekwencje, IDENTITY, AUTO_INCREMENT.

SELECT

Polecenie zwracające dane z jednej lub kilku tabel, jedyne polecenie DQL w SQL. Kolejność, w jakiej piszemy jego klauzule, różni się od kolejności, w jakiej silnik je logicznie wykonuje. Rozwinięcie: SELECT: kolejność zapisu a kolejność wykonania.

SELF JOIN

Złączenie tabeli z samą sobą, wymagające nadania jej dwóch różnych aliasów, żeby odróżnić, które wystąpienie tabeli ma się na myśli w danym miejscu. Sama tabela fizycznie istnieje raz — aliasy istnieją tylko na czas jednego zapytania. Rozwinięcie: CROSS JOIN i SELF JOIN.

Seq scan

Skan sekwencyjny — sposób odczytu, w którym silnik czyta kolejno każdą stronę danych tabeli od początku do końca i dla każdego wiersza sprawdza warunek z osobna. Koszt takiego przejścia rośnie liniowo z liczbą wierszy tabeli, niezależnie od tego, ile z nich ostatecznie spełni warunek. Rozwinięcie: Skan sekwencyjny, skan indeksu, lookup — trzy sposoby dotarcia do wiersza.

Silnik bazodanowy

Oprogramowanie, które umie czytać i zapisywać zbiór danych zgodnie z regułami — SQL-em, transakcjami, uprawnieniami. Silnik to produkt instalowany raz, z którego można uruchomić wiele działających instancji naraz. Rozwinięcie: Baza danych, silnik, instancja, schemat: cztery pojęcia, które się myli.

Skos zapisu

Write skew — anomalia współbieżności nienazwana wprost w standardzie SQL obok trzech oficjalnych, w której dwie transakcje osobno czytają ten sam zestaw wierszy, każda z osobna zmienia inny wiersz na podstawie tego odczytu, i obie się zatwierdzają — mimo że wykonane razem, jedna po drugiej, dałyby inny, poprawny wynik. Poziomy izolacji chroniące przed trzema klasycznymi anomaliami nie zawsze chronią przed skosem zapisu. Rozwinięcie: Anomalie współbieżności: brudny odczyt, niepowtarzalny odczyt, fantom.

Spójność

Druga litera ACID — gwarancja, że transakcja przenosi bazę z jednego stanu zgodnego z regułami (ograniczeniami, kluczami obcymi) do innego takiego stanu, nigdy przez stan pośredni naruszający

te reguły na trwałe. Rozwinięcie: Czym jest transakcja: ACID rozłożone na cztery obietnice.

SQL

Structured Query Language — deklaratywny język do opisywania, jakie dane mają się znaleźć w wyniku, oraz do definiowania i kontrolowania struktury, w której te dane leżą. Deklaratywny znaczy, że zapytanie opisuje pożądany wynik, a nie kroki prowadzące do jego policzenia. Rozwinięcie: Skąd się wziął SQL i dlaczego każdy silnik mówi nim inaczej.

Standaryzacja SQL

Proces, w którym niezależna organizacja spisuje wspólną definicję języka, żeby polecenia napisane pod jeden silnik dało się uruchomić na innym. Standard SQL jest rewidowany regularnie od 1986 roku, a kolejne wersje dokładają zapytania rekurencyjne, funkcje okna, obsługę JSON i inne rozszerzenia. Rozwinięcie: Skąd się wziął SQL i dlaczego każdy silnik mówi nim inaczej.

Statystyki

Zestaw liczb opisujących rozkład danych w tabeli i jej kolumnach, zbierany przez silnik i odczytywany przez optymalizator przy planowaniu każdego zapytania — liczba wierszy, liczba stron na dysku, liczba różnych wartości w kolumnie i histogram ich rozkładu. Nieaktualne statystyki są jedną z najczęstszych przyczyn nagle złego planu wykonania. Rozwinięcie: Statystyki i optymalizator kosztowy.

Stopień relacji

Liczba atrybutów, czyli kolumn, relacji. Tabela o pięciu kolumnach ma stopień piąty, niezależnie od tego, ile ma wierszy.

Superklucz

Dowolny zbiór kolumn, którego wartości jednoznacznie identyfikują wiersz, bez wymogu minimalności. Cała tabela zawsze jest superkluczem, bo dwa identyczne pod każdym względem wiersze są w praktyce nieodróżnialne. Rozwinięcie: Klucze: główny, obcy, unikalny, naturalny, sztuczny.

T

Tabela

Praktyczna realizacja relacji — zbiór wierszy o jednakowej strukturze kolumn, zdefiniowanej raz w `CREATE TABLE`. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

Tabela łącząca

Tabela, której jedynym zadaniem jest przechowywanie par kluczy obcych wskazujących po jednym wierszu z każdej z dwóch tabel biorących udział w relacji wiele do wielu. Każdy wiersz tabeli łączącej reprezentuje jedno konkretne skojarzenie. Rozwinięcie: Relacje: 1:1, 1:N, N:M i tabela łącząca.

Tabela pochodna

Podzapytanie postawione w miejscu, gdzie normalnie stoi nazwa tabeli, czyli w klauzuli `FROM`. Silnik traktuje jego wynik jak zwykłą tabelę tymczasową, istniejącą tylko na czas tego jednego zapytania, i wymaga dla niej aliasu. Rozwinięcie: Podzapytania: skalarne, wierszowe, skorelowane, EXISTS.

Tablespace

Logiczny kontener grupujący jeden lub więcej plików fizycznych na dysku, w którym silnik trzyma dane tabel i indeksów przypisanych do tego kontenera. Pozwala rozdzielić dane różnych baz albo obiektów na różne dyski fizyczne bez zmiany niczego w samym SQL-u odwołującym się do tabel. Rozwinięcie: Rozmiar bazy: pliki, tablespace'y, autoextend.

Transakcja

Jednostka pracy złożona z jednej albo wielu operacji na bazie danych, która dla całej reszty systemu wygląda jak jedna, niepodzielna czynność — albo wszystkie operacje w niej się zapiszą, albo żadna. Nie jest sposobem przyspieszenia niczego ani mechanizmem uprawnień, tylko granicą, wewnątrz której baza gwarantuje, że nic niedokończonego nie zostanie na trwałe zapisane. Rozwinięcie: Czym jest transakcja: ACID rozłożone na cztery obietnice.

Trójwartościowa logika

System logiczny SQL, w którym każdy warunek daje jeden z trzech wyników — prawda, fałsz albo nieznana — a ten ostatni pojawia się zawsze, gdy w porównaniu uczestniczy `NULL`. Klasyczna logika dwuwartościowa zna tylko prawdę i fałsz; SQL dokłada trzecią możliwość, bo baza musi umieć reprezentować niekompletną wiedzę. Rozwinięcie: NULL: trzecia wartość logiczna i pułapki, które z niej wynikają.

Trwałość

Czwarta litera ACID — gwarancja, że zmiany zatwierdzonej transakcji przetrwają nawet awarię zasilania czy restart silnika zaraz po zatwierdzeniu. Zapewnia ją zapis do dziennika transakcyjnego na dysku, zanim silnik potwierdzi zatwierdzenie aplikacji. Rozwinięcie: Czym jest transakcja: ACID rozłożone na cztery obietnice.

Trzecia postać normalna (3NF)

Wymóg, żeby żadna kolumna niebędąca kluczem nie zależała od innej kolumny niebędącej kluczem, tylko bezpośrednio od klucza głównego. Zależność przechodzącą przez pośrednią kolumnę nazywamy zależnością przechodnią, i to ją 3NF eliminuje. Rozwinięcie: Normalizacja do 3NF i świadoma denormalizacja.

TRUNCATE

Polecenie usuwające całą zawartość tabeli naraz, bez sprawdzania wierszy pojedynczo i bez możliwości odwołania się do `WHERE`, ale zostawiające samą definicję tabeli nietkniętą. Działa szybciej niż `DELETE` bez warunku, bo nie generuje osobnego wpisu w dzienniku transakcyjnym dla każdego usuwanego wiersza. Rozwinięcie: DELETE vs TRUNCATE vs DROP.

Typ danych

Deklaracja z góry narzucająca, jakiego rodzaju wartości wolno przechowywać w danej kolumnie i ile miejsca silnik ma na nie zarezerwować — konkretna, techniczna realizacja pojęcia dziedziny. Silnik wymusza typ przy każdym zapisie, bo bez tego wymuszenia porównania i sortowanie nie miałyby jednoznacznego znaczenia. Rozwinięcie: CREATE TABLE: typy danych i ich odpowiedniki między silnikami.

U

UNION

Operacja sklejająca wyniki dwóch niezależnych zapytań w jeden, ustawiając je jeden pod drugim, w przeciwieństwie do złączenia, które zestawia tabele obok siebie. Sam `UNION`, bez `ALL`, dodatkowo usuwa duplikaty powstałe po sklejeniu. Rozwinięcie: UNION vs UNION ALL.

UNION ALL

Wariant sumy zbiorów, który skleja wyniki obu zapytań bez żadnego sprawdzania duplikatów, po prostu dokładając wiersze drugiego zapytania pod wierszami pierwszego. Jest tańszy niż goły `UNION`, bo pomija kosztowne porównywanie wszystkich wierszy w poszukiwaniu powtórzeń. Rozwinięcie: UNION vs UNION ALL.

UPDATE

Polecenie DML zmieniające wartości jednej lub więcej kolumn w wierszach, które już istnieją w tabeli, bez zmiany liczby wierszy. `WHERE` w `UPDATE` działa dokładnie tak samo jak `WHERE` w `SELECT` — to ten sam predykat sprawdzany dla każdego wiersza z osobna. Rozwinięcie: UPDATE: i dlaczego zawsze piszemy WHERE zanim napiszemy SET.

Upsert

Operacja wstawiająca nowy wiersz, jeśli pasujący jeszcze nie istnieje, albo aktualizująca istniejący, jeśli już jest — nazwa to zlepek słów update i insert. Naiwna implementacja tej logiki po stronie aplikacji staje się wyścigiem przy współbieżności, dlatego silniki oferują dedykowane, atomowe polecenia do tego celu. Rozwinięcie: MERGE / UPSERT — jak to robi każdy silnik.

UUID

Uniwersalny, 128-bitowy identyfikator, praktycznie niepowtarzalny bez potrzeby centralnego licznika. Przydatny, gdy identyfikatory muszą być generowane niezależnie w wielu miejscach naraz, kosztem większego rozmiaru niż zwykła liczba całkowita i gorszej lokalności fizycznej w indeksie B-drzewa.

V

VACUUM

Proces porządkujący w silnikach z wielowersyjną kontrolą współbieżności, który usuwa martwe wersje wierszy pozostawione przez `UPDATE` i `DELETE`, odzyskując miejsce na dysku i ograniczając fragmentację wewnętrzną. Bez regularnego działania tego procesu tabele o dużym ruchu zapisu rosną, mimo że logicznie nie przybywa w nich danych. Rozwinięcie: MVCC: dlaczego czytający nie blokuje piszącego.

W

WAL

Write-Ahead Log, dziennik zapisu z wyprzedzeniem — sekwencyjny plik, do którego silnik zapisuje opis każdej zmiany danych, zanim ta zmiana trafi do właściwych plików tabel i indeksów na dysku. Nazwa opisuje wprost zasadę: pisz najpierw do dziennika, dopiero potem do głównych struktur danych. Rozwinięcie: Log transakcyjny: WAL, redo, transaction log.

Warunek stopu

To, co kończy rekurencję w zapytaniu rekurencyjnym — moment, w którym część rekurencyjna przestaje zwracać nowe wiersze. Przy danych z cyklem brak jawnego zabezpieczenia w postaci limitu głębokości prowadzi do nieskończonej pętli. Rozwinięcie: CTE: WITH i zapytania rekurencyjne.

Warunek złączenia

Wyrażenie logiczne, które dla każdej pary wierszy z iloczynu kartezjańskiego rozstrzyga, czy ta para ma sens i ma zostać w wyniku. Bez tego warunku zostaje surowy iloczyn kartezjański ze wszystkimi bezsensownymi skojarzeniami. Rozwinięcie: Czym jest złączenie: iloczyn kartezjański i warunek, który go ogranicza.

Węzeł wewnętrzny

Strona B-drzewa, która nie przechowuje samych wierszy, tylko klucze graniczne i wskaźniki do węzłów niższego poziomu — mówi, w którą stronę zejść, żeby trafić do właściwego zakresu wartości. Rozwinięcie: B-drzewo od środka.

WHERE

Klauzula filtrująca pojedyncze wiersze na podstawie predykatu, wykonywana logicznie zaraz po `FROM`, przed jakąkolwiek agregacją. Wiersze, dla których predykat dał fałsz albo wartość nieznaną, są odrzucane tak samo. Rozwinięcie: WHERE — filtrowanie wierszy, operatory, LIKE, IN, BETWEEN.

Wiersz

Praktyczna realizacja krotki — pojedynczy rekord tabeli, zestaw wartości po jednej dla każdej kolumny. Rozwinięcie: Model relacyjny: tabela, wiersz, kolumna, dziedzina.

Współbieżność

Sytuacja, w której więcej niż jedna transakcja działa na tej samej bazie danych w tym samym przedziale czasu, potencjalnie czytając i zmieniając te same wiersze. Sama w sobie nie jest problemem — silniki są projektowane właśnie po to, żeby obsługiwać wiele sesji naraz — problemem jest to, co się dzieje, gdy dwie transakcje wchodzą sobie w drogę. Rozwinięcie: Czym jest transakcja: ACID rozłożone na cztery obietnice.

Wyzwalacz

Trigger — blok kodu proceduralnego, który silnik uruchamia automatycznie w reakcji na zdarzenie na tabeli, takie jak `INSERT`, `UPDATE` czy `DELETE`, bez jawnego wywołania przez aplikację. Wygodny do egzekwowania reguł biznesowych i audytu, ale ukrywa logikę poza zapytaniem, które ją uruchomiło, co bywa trudne do debugowania.

Wzorzec LIKE

Ciąg znaków, w którym `%` zastępuje dowolny fragment tekstu, także pusty, a `_` zastępuje dokładnie jeden znak. Wzorzec zaczynający się od stałego prefiksu i kończący `%` zachowuje

sargowalność w większości silników, w odróżnieniu od wzorca zaczynającego się od `%`. Rozwinięcie: WHERE — filtrowanie wierszy, operatory, LIKE, IN, BETWEEN.

Z

Zakleszczenie

Deadlock — sytuacja, w której dwie albo więcej transakcji wzajemnie czekają na blokady trzymane przez siebie nawzajem, tworząc cykl, z którego żadna nie może się wydostać bez interwencji z zewnątrz. Różni się od zwykłego oczekiwania na blokadę tym, że zwykłe oczekiwanie się kończy, gdy transakcja blokująca w końcu zatwierdzi albo wycofa swoje zmiany. Rozwinięcie: Zakleszczenie (deadlock): jak powstaje i jak je czytać z logu.

Zapytanie deklaratywne

Zapytanie opisujące, jak wynik ma wyglądać, a nie krok po kroku, jak go policzyć. To, w jakiej kolejności silnik faktycznie przejdzie po danych i którego indeksu użyje, zostaje jego decyzją, widoczną dopiero w planie wykonania. Rozwinięcie: Skąd się wziął SQL i dlaczego każdy silnik mówi nim inaczej.

Zapytanie rekurencyjne

Zapytanie zbudowane na CTE, które odwołuje się samo do siebie, dokładając kolejne poziomy wyniku aż do spełnienia warunku stopu — typowe zastosowanie to przechodzenie hierarchii o nieznanej z góry głębokości, jak struktura przełożonych. Rozwinięcie: CTE: WITH i zapytania rekurencyjne.

Zasada najbardziej lewego prefiksu

Reguła mówiąca, że indeks złożony na kilku kolumnach da się wykorzystać dla warunku dotyczącego pierwszej kolumny samej w sobie oraz dla warunku na kolejne kolumny razem z pierwszą, ale nie dla warunku, który pomija pierwszą kolumnę. Wynika to wprost z tego, jak wpisy indeksu są posortowane — najpierw według pierwszej kolumny. Rozwinięcie: Indeks złożony, pokrywający, częściowy, funkcyjny.

Złączenie zewnętrzne

OUTER JOIN — złączenie, które w przeciwieństwie do INNER JOIN zachowuje w wyniku wiersze bez dopasowania z co najmniej jednej strony, uzupełniając brakujące kolumny wartościami `NULL`. LEFT JOIN i RIGHT JOIN to dwa warianty tego złączenia, różniące się tym, którą stronę traktują jako tę do zachowania w całości. Rozwinięcie: LEFT JOIN i RIGHT JOIN.

Revision #43
Created 2026-09-24 23:07:52 UTC by Tomasz Konieczny
Updated 2026-10-26 14:12:03 UTC by Tomasz Konieczny