

Dotazovací jazyky. QBE, SQL, DDL, DML. Integritní omezení. Transakční zpracování, vlastnosti transakcí.

Dva typy jazyků

Se SŘBD spojena existence dvou typů jazyků:

- jazyk pro definici dat (Data Definition Language) k vytvoření definic všech uživatelských dat potřebných v aplikaci (logické schéma databáze),
- jazyk pro manipulaci dat (Data Manipulation Language) k aktualizaci a výběru dat podle požadavků,
 - dotazování v řeči schématu databáze, případně na vyšší úrovni zajišťuje

dotazovací jazyk.

Dva alternativní jazyky definované s relačním modelem

při definici relačního modelu byly navrženy pro práci s daty a strukturou databáze dva jazyky:

- procedurální relační algebra - při použití nutno specifikovat jaká data jsou zapotřebí a jak tato data vyvolat (procedura),
- neprocedurální relační kalkul - nutno specifikovat pouze jaká data jsou potřeba, SŘBD určí efektivní metodu pro vyvolání dat,

Tyto jazyky jsou základem pro relačně úplné jazyky vyšší úrovně (rel. úplné = poskytují možnosti relačního kalkulu; jsou neprocedurální):

- Structured Query Language - standardizován ISO,
- Query-by-Example (QBE) - graficky orientovaný způsob dotazování ("ukaz a klepni") pro jednodušší dotazy vyjádřené pomocí termínů několika tabulek, převádí "grafický" dotaz na příkaz SQL.

Jazyk pro manipulaci dat DML

- slouží pro vyvolání a aktualizaci dat,
- příkaz pro vyvolání a zobrazení dat z databáze:
SELECT sloupce výstupu
FROM tabulka
WHERE filtr, podmínka
GROUP BY skupiny řádků se stejnou hodnotou sloupce
HAVING filtr skupin dle podmínky
ORDERED BY řazení výstupu,
- příkaz pro přidání nových řádků:
INSERT INTO tabulka (sloupce) VALUES (hodnoty),
- modifikace existujících dat (splňujících podmínku):
UPDATE tabulka SET sloupec = NULL, výraz, (nesetříděný dotaz) WHERE vyhled. podm.
- odstranění [řádků z] tabulky [splňujících podmínku]:
DELETE FROM tabulka [WHERE podmínka].

Jazyk pro definici dat DDL

- slouží pro definici struktury databáze a kontroly přístupu k datům,
- definice databázových objektů - tabulek, pohledů:
CREATE SCHEMA
- vytvoření schématu a prázdné tabulky a jejich Integritních omezení s definicí sloupců s datovými typy a jejich Integritních omezení:

- CREATE TABLE tabulka (sloupec , [CONSTRAINT tabulkové IO]
 sloupec = název datový typ [CONSTRAINT sloupcové IO, výchozí hodnota];
- změna definice schématu, přidání/odebrání atributu(sloupce) a/nebo Integritního Omez: ALTER TABLE tab. ADD[COLUMN]/ALTER[COLUMN]/DROP COLUMN,CONSTRAINT;
 - odstranění tabulky, jak schématu, tak obsahu dat DROP TABLE tabulka;

Datové typy sloupců

- přesné numerické typy (INTEGER, NUMERIC, DECIMAL),
- aproximativní numerické typy (FLOAT, REAL, DOUBLE PRECISION),
- znakové řetězce (CHARACTER=VARCHAR),
- bitové řetězce (BIT),
- časové datové typy (DATE, TIME),
- časové intervaly.

Integritní omezení

- lokální, sloupcové Integritní Omezení umožňuje omezit množinu platných hodnot daného atributu v rámci záznamu (hodnotu v daném sloupci v rámci řádku):
 - pojmenované IO -> CREATE TABLE ... (... , CONSTRAINT jmenoConstraint ...)
 - nepojmenované
 pět typů omezení na platnou hodnotu atributu:
 NOT NULL, UNIQUE, PRIMARY KEY, REFERENCES cizí klíč, CHECK-obecná podm.
- globální integritní omezení celé tabulky (zobecnění atributových IO pro kombinace hodnot více sloupců) UNIQUE, FOREIGN KEY (=REFERENCES), CHECK

Referenční integrita

- modeluje logické vztahy mezi sloupci dvou tabulek (hlavní tabulky a závislé tabulky)
- při aktualizaci tabulky referované, referující může dojít k porušení integrity cizích klíčů:
 - hlášení chyby
 - vykonání referenční akce
 podmínky spuštění akce: ON UPDATE, ON DELETE
 CASCADE(propagace), SET NULL, SET DEFAULT, NO ACTION.

Integritní omezení na úrovni schématu

- formulace integritního omezení mimo tabulky:
 CREATE ASSERTION jméno CHECK (...)
- možné problémy při implementaci (jak kontrolovat při aktualizaci?)

Transakční zpracování

motivace:

- chránit data organizovaná pod SŘBD
 - poskytnout korektní a rychlý přístup většímu množství uživatelů,
- plní dvě funkční komponenty:
- komponenta řízení souběžného (paralelního) zpracování (concurrency control) - uživatel vidí konzistentní stavy i při paralelním zpracování
 - komponenta zpracování z chyb (recovery) stav databáze není narušen v případě chyby systému, média.

Zachování konzistence databáze = splnění všech integritních omezení definovaných schématem databáze v každém novém stavu.

Transakce konzistenci zachovává, končí v konečném čase.

Program transakce se provede buď celý nebo vůbec.

Souběžné (paralelní) zpracování transakcí realizováno proložením.

Transparence paralelního zpracování = konzistence i při souběžném běhu transakcí. Plná transparence - jakoby uživatel pracoval sám, bez interference s ostatními uživateli.

Transakce jsou základem zotavení z chyb. Atomicita vzhledem k chybám = udržení konzistence i s výskytem chyby.

Architektura pro transakční zpracování

- manažer dat - komunikace s DB - požadavky čtení/zápis
- zodpovědnost za atomicitu vzhledem k chybám
- rozvrhovač - řízení paralelního zpracování transakcí
- zajištění transparence paralelního zpracování
- manažer transakcí - vyvolává transakci, posílá požadavky na data.

Transakce

- posoupnost akcí (čtení/zápis/výpočet), se kterou se zachází jako s celkem, logická jednotka práce
- příkazy COMMIT, ROLLBACK zajišťují atomický projev transakce pro uživatele (COMMIT=potvrzení, ROLLBACK=navrácení dle žurnálu do výchozího stavu).
- stavy transakce: aktivní (počáteční), částečně potvrzený, chybný, zrušený (ukončení), potvrzený (ukončení)

Vlastnosti transakce (ACID)

- atomicita (Atomicity) - transakce se tváří jako celek, buď proběhne celá nebo vůbec,
- konzistence (Consistency) - transakce transformuje databázi z jednoho konzistentního stavu do jiného konzistentního stavu,
- nezávislost (Independence, Isolation) - dílčí efekty transakce nejsou viditelné jiným transakcím,
- trvanlivost (Durability) - efekty úspěšně ukončené transakce jsou v databázi uloženy.

Rozvrh = stanovené pořadí provádění akcí více transakcí

Problémy s paralelním zpracováním transakcí (anomálie)

- ztráta aktualizace
- dočasná aktualizace (při chybě systému) (1)
- neopakovatelné čtení (2)
- fantóm (3) (opakovaný dotaz vrátí odlišnou množinu záznamů)

odstranění anomálií pomocí uspořadatelnosti, úrovně izolace (0-3), úroveň izolace volena dle míry závažnosti nekonzistence

Sériové rozvrhy - zachování operací transakce pohromadě.

Uspořadatelnost - efekt paralelního zpracování stejný jako u sériového, tj. výsledek vede ke konzistentnímu stavu

Operace konfliktní a kompatibilní

Ekvivalence rozvrhů (komutativita operací čtení/zápisu)

Uzamykací protokoly - pravidla konstrukce transakcí