

9. Databázové systémy

Databáze a Systém řízení báze dat (SŘBD)

Databáze je organizovaný soubor informací, který je uložen na disku a umožňuje efektivní vyhledávání, ukládání a správu dat pomocí počítače. Typicky se jedná o data v tabulkách, kde jsou záznamy (řádky) a atributy (sloupce). Databáze je navržena tak, aby data nebyla zbytečně opakovaná a aby byla spolehlivě chráněná proti výpadkům nebo ztrátě.

Systém řízení báze dat (SŘBD, anglicky DBMS) je software, který umožňuje manipulaci s databázemi – zajišťuje ukládání, vyhledávání, změny, zálohování a další správu dat. Mezi běžně používané SŘBD patří například MySQL, Oracle, MS SQL Server, PostgreSQL.

Databázový systém (DBS) je souhrnný pojem zahrnující jak samotná data (databázi), tak software pro jejich řízení (SŘBD).

Životní cyklus vývoje databázového systému

1. **Plánování:** Určují se cíle systému, shromažďují se požadavky a tvoří se plán projektu.
2. **Analýza požadavků:** Detailně se analyzují potřeby uživatelů a požadavky na data a jejich zpracování.
3. **Návrh (design):** Tvoří se konceptuální a logické modely dat (například ER diagramy), volí se vhodné SŘBD, provádí se normalizace struktury.
4. **Implementace:** Návrh se převádí do konkrétní podoby v databázi – vytvářejí se schémata tabulek, nastavují indexy a zabezpečení.
5. **Testování:** Ověřuje se funkčnost, správnost a výkon databázového systému.
6. **Nasazení:** Systém je uveden do provozu a začíná sloužit uživatelům.
7. **Údržba:** Pravidelně se provádí zálohování, aktualizace a optimalizace databáze.

Relační datový model

Relační datový model je nejrozšířenější způsob uchovávání dat, kde jsou data organizována v **relačních tabulkách**. Každá tabulka představuje určitý typ objektu (entitu), její řádky jsou konkrétní instance těchto objektů a sloupce představují jejich vlastnosti (atributy).

Základní pojmy:

Entitní typ: Skupina objektů se stejnými atributy, např. „Student“.

Entita: Konkrétní objekt, například student Jan Novák.

Vztahová množina: Kolekce vztahů mezi entitními typy, např. „Zapsán v kurzu“.

Vztah: Jednotlivé spojení mezi entitami, například „Jan Novák je zapsán v Matematice“.

Atribut: Vlastnost entity, např. „Jméno“, „Věk“.

Kardinalita: Určuje počet instancí ve vztahu (např. 1:N, N:M).

Primární klíč: Atribut či kombinace, který jednoznačně identifikuje každý záznam v tabulce.

Cizí klíč: Atribut, který odkazuje na primární klíč v jiné tabulce a vytváří vazby mezi tabulkami.

Konceptuální a logický model

Konceptuální model je abstraktní model dat, který zobrazuje entity, jejich vlastnosti a vztahy mezi nimi, nezávisle na konkrétní databázové technologii. Nejčastěji se zapisuje formou ER diagramů. Slouží k pochopení obchodních požadavků a datových struktur v nejširším smyslu.

Logický model je přesnější návrh databáze pro konkrétní datový model (například relační). Určuje tabulky, sloupce, klíče a datové typy, stále ale není závislý na konkrétní fyzické implementaci. Logický model je mostem mezi konceptuálním modelem a konečnou realizací databáze.

Normalizace, závislosti a normální formy

Normalizace je proces rozdělení tabulek a úpravy jejich struktury za účelem odstranění redundantních dat a minimalizace anomálií při vkládání, mazání a aktualizaci dat. Výsledkem jsou čistší, efektivnější a snáze spravovatelné databázové struktury.

Funkční závislost znamená, že hodnota jednoho atributu (nebo více) jednoznačně určuje hodnotu jiného atributu.

- **Jednoduchá závislost ($X \rightarrow Y$):** Každá hodnota X určuje právě jednu hodnotu Y.
- **Parciální závislost:** Atribut závisí jen na části složeného klíče.
- **Tranzitivní závislost:** Pokud $X \rightarrow Y$ a $Y \rightarrow Z$, pak $X \rightarrow Z$.

Normální formy:

- **První normální forma (1NF):** Všechny hodnoty v tabulce jsou atomické, neobsahují opakující se skupiny.
- **Druhá normální forma (2NF):** Je v 1NF a žádný neklíčový atribut není závislý jen na části složeného klíče.
- **Třetí normální forma (3NF):** Je ve 2NF a žádný neklíčový atribut není tranzitivně závislý na primárním klíči.
- **Čtvrtá (4NF) a pátá (5NF) normální forma:** Řeší složitější typy závislostí, například multihodnotové nebo join závislosti.

Význam normalizace: Zajišťuje správnost, integritu a konzistenci dat v databázi a zjednodušuje údržbu struktury i obsahu databáze.

Integritní omezení

Integritní omezení jsou pravidla, která zabezpečují správnost a konzistenci dat v databázi:

- **Entitní integrita:** Každý záznam musí mít jednoznačný a neprázdný (NOT NULL) primární klíč.
- **Referenční integrita:** Cizí klíč musí vždy odkazovat na existující hodnotu v cílové tabulce nebo být prázdný.
- **Doménová integrita:** Atributy musí obsahovat pouze hodnoty odpovídající definovanému datovému typu a rozsahu.
- Dále mohou být nastavena **unikátní omezení**, **NOT NULL** nebo **CHECK** podmínky na úrovni sloupců.

Vztah objektů a relačních tabulek

Objektově orientované programování pracuje s objekty, zatímco relační databáze s tabulkami. Proces převodu datových struktur mezi těmito světy zajišťují **ORM nástroje (Object-Relational Mapping)**, například Hibernate, Entity Framework nebo SQLAlchemy. Tyto nástroje automaticky mapují třídy a jejich vlastnosti na tabulky a sloupce v databázi, včetně vazeb mezi entitami.

Jazyk SQL

SQL (Structured Query Language) je standardizovaný jazyk pro práci s relačními databázemi. Obsahuje příkazy pro:

- **Definici dat (DDL):** CREATE TABLE, ALTER TABLE, DROP TABLE
- **Manipulaci s daty (DML):** SELECT, INSERT, UPDATE, DELETE
- **Kontrolu přístupu (DCL):** GRANT, REVOKE
- **Transakční řízení (TCL):** COMMIT, ROLLBACK, SAVEPOINT

Díky SQL lze databáze nejen vytvářet a měnit, ale také efektivně dotazovat a analyzovat uložená data.

Transakce, ACID vlastnosti a uzamykací protokoly

Transakce je logická jednotka práce, která obsahuje jeden nebo více SQL příkazů provedených jako jeden celek – buď se všechny změny provedou, nebo se žádná neprovede.

Stavy transakce:

1. **Aktivní:** Transakce právě probíhá.
2. **Částečně potvrzená:** Příkazy provedeny, ale výsledek není uložen na disk.
3. **Potvrzená (committed):** Změny jsou trvale zapsány a viditelné ostatním.
4. **Selhala:** Při provádění nastala chyba.
5. **Zrušená (rolled back):** Změny jsou vráceny zpět do původního stavu.

ACID vlastnosti zajišťují spolehlivost transakcí:

- **Atomicita:** Transakce je nedělitelná, provede se celá, nebo vůbec.
- **Konzistence:** Transakce převádí databázi z jednoho konzistentního stavu do jiného.
- **Izolace:** Probíhající transakce se navzájem neovlivňují, systém používá zámky (locks).
- **Trvalost:** Po potvrzení transakce jsou změny trvale uloženy i při výpadku systému.

Uzamykací protokoly zabraňují konfliktům při souběžných operacích:

- **Zámky na úrovni řádku:** Umožňují vysokou souběžnost, ale jsou náročné na správu.
- **Zámky na úrovni tabulky:** Omezují souběžnost, ale jsou jednodušší.
- **Zámky na úrovni databáze:** Zcela blokuji souběh, používají se výjimečně.

Paralelní zpracování

V moderních databázích často běží více transakcí současně, což zvyšuje výkon systému. Je ale nutné zajistit, aby nedocházelo k chybám z důvodu konfliktů – k tomu slouží právě uzamykací protokoly a řízení izolace transakcí.

NoSQL databáze

NoSQL databáze jsou navrženy pro specifické případy, kdy klasický relační model není vhodný (například pro velmi velké, rychle rostoucí nebo nestrukturované datové sady).

Typy NoSQL databází:

- **Dokumentové databáze:** Ukládají data ve formě dokumentů (JSON, BSON), např. MongoDB.
- **Sloupcové databáze:** Ukládají data po sloupcích, vhodné pro analýzy velkých dat (např. Apache Cassandra).
- **Klíč-hodnota databáze:** Jednoduché páry klíč-hodnota, extrémně rychlý přístup k datům (např. Redis).
- **Grafové databáze:** Ukládají data ve formě uzlů a hran, ideální pro složité síťové vztahy (např. Neo4j).

NoSQL systémy nabízejí vyšší škálovatelnost, flexibilitu schématu a efektivní práci s nestrukturovanými daty, což je často klíčové například v oblasti Big Data nebo IoT.