

Databáze

1. Definice

- **Definice databáze:** Databáze je systém pro ukládání, organizování, spravování a manipulaci s daty. Umožňuje efektivní ukládání a rychlý přístup k velkým množstvím informací.
- **Účel databáze:**
 - Umožňuje spravovat data na centralizovaném místě.
 - Podporuje efektivní dotazování, třídění a analýzu dat.
 - Poskytuje bezpečnostní funkce, jako jsou přístupové kontroly a šifrování.

2. Typy databází

2.1 Dle struktury dat:

- **Hierarchické databáze:**
 - Data jsou organizována ve stromové struktuře, kde každá položka má jednoznačného nadřazeného (rodič).
 - Příkladem je XML nebo starší IBM IMS.
- **Síťové databáze:**
 - Data jsou organizována do grafu, kde každý záznam může mít více nadřazených a podřazených záznamů.
 - Příkladem je systém CODASYL.
- **Relational (relační) databáze:**
 - Data jsou organizována do tabulek, které jsou navzájem propojené pomocí klíčů (primární klíč, cizí klíč).
 - Příklady: MySQL, PostgreSQL, Oracle, SQL Server.
- **NoSQL databáze:**
 - Nevyužívají tabulkovou strukturu a jsou určeny pro specifické úkoly (např. dokumentové, grafové, klíč-hodnota).
 - Příklady: MongoDB, Cassandra, Redis, Neo4j.

2.2 Dle přístupu k datům:

- **SQL (Structured Query Language) databáze:**
 - Používají standardní dotazovací jazyk SQL pro komunikaci s databází.
 - Vhodné pro aplikace, kde je potřeba silná konzistence dat.
- **NoSQL databáze:**
 - Nerequire SQL a umožňují flexibilnější struktury dat (např. dokumenty, klíč-hodnota).
 - Vhodné pro aplikace s dynamickými nebo nejednotnými datovými strukturami.

3. Relační databáze

- **Relační model:**
 - Data jsou uspořádána do tabulek, které mají řádky (záznamy) a sloupce (atributy).
 - Každá tabulka má primární klíč, což je unikátní identifikátor pro každý záznam.
 - Tabulky mohou být propojeny pomocí cizího klíče.

- **Tabulka (Table):**
 - Základní jednotka v relační databázi. Každá tabulka obsahuje konkrétní typ informací (např. zákazníci, objednávky).
- **Primární klíč (Primary Key):**
 - Atribut nebo kombinace atributů, které jednoznačně identifikují každý záznam v tabulce.
- **Cizí klíč (Foreign Key):**
 - Atribut v jedné tabulce, který odkazuje na primární klíč v jiné tabulce. Umožňuje propojení tabulek.
- **Normalizace:**
 - Proces organizování dat v databázi tak, aby byla minimalizována redundance a závislosti mezi atributy.
 - Normalizační stupně (1NF, 2NF, 3NF, BCNF).

4. SQL – Structured Query Language

4.1 Základní příkazy SQL:

- **SELECT:** Používá se k dotazování na data v tabulkách.
`SELECT * FROM tabulka WHERE podmínka;`
- **INSERT:** Používá se k přidávání nových záznamů do tabulky.
`INSERT INTO tabulka (sloupec1, sloupec2) VALUES (hodnota1, hodnota2);`
- **UPDATE:** Používá se k aktualizaci existujících záznamů v tabulce.
`UPDATE tabulka SET sloupec = nová_hodnota WHERE podmínka;`
- **DELETE:** Používá se k odstraňování záznamů z tabulky.
`DELETE FROM tabulka WHERE podmínka;`

4.2 Pokročilé dotazy SQL:

- **Join (spojování tabulek):**
 - **INNER JOIN:** Spojuje záznamy, které mají shodné hodnoty v obou tabulkách.
 - **LEFT JOIN:** Spojuje všechny záznamy z levé tabulky s odpovídajícími záznamy v pravé tabulce (pokud existují).
- **Skupinové funkce (Aggregation Functions):**
 - **COUNT():** Počítá počet řádků.
 - **SUM():** Sčítá hodnoty.
 - **AVG():** Vypočítá průměr.
 - **GROUP BY:** Skupinu hodnot podle specifického sloupce.
- **Podmínky (WHERE, HAVING):**
 - **WHERE:** Používá se k filtrování dat.
 - **HAVING:** Používá se k filtrování agregovaných dat.

5. Transakce a správa databáze

- **Transakce:**

- Transakce je sekvence jednoho nebo více SQL příkazů, které se provedou jako celek.
- Zajistí, že všechny operace probíhají úspěšně, nebo žádná z nich.
- **ACID vlastnosti:**
 - **Atomicita:** Transakce je buď celá provedena, nebo vůbec.
 - **Konzistence:** Transakce vede databázi ze stavu konzistentního do jiného konzistentního stavu.
 - **Izolace:** Transakce probíhají nezávisle na sobě.
 - **Trvalost:** Po dokončení transakce jsou změny trvalé.
- **Uzamykání (Locking):**
 - Mechanismus pro zajištění, že transakce nebudou ovlivňovat data navzájem.
 - **Shared Lock:** Povolení pro čtení.
 - **Exclusive Lock:** Povolení pro zápis.

6. NoSQL databáze

6.1 Druhy NoSQL databází:

- **Dokumentové databáze:**
 - Ukládají data ve formátu dokumentů (např. JSON, BSON).
 - Příklad: MongoDB.
- **Klíč-hodnota databáze:**
 - Ukládají data jako páry klíč-hodnota.
 - Příklad: Redis.
- **Sloupcové databáze:**
 - Ukládají data v tabulkách, ale místo řádků využívají sloupce.
 - Příklad: Apache Cassandra.
- **Grafové databáze:**
 - Ukládají data jako uzly a hrany v grafu.
 - Příklad: Neo4j.

7. Optimalizace a zálohování databází

7.1 Optimalizace výkonu:

- **Indexy:** Zlepšují výkon při hledání v tabulkách.
- **Denormalizace:** Ukládání redundantních dat pro zrychlení dotazů (proti normalizaci).
- **Cacheování:** Skladování často používaných dat v rychlé paměti.

7.2 Zálohování databáze:

- **Full Backup:** Záloha celé databáze.
- **Incremental Backup:** Záloha pouze změněných dat od poslední zálohy.
- **Point-in-Time Recovery:** Obnova databáze do konkrétního bodu v čase.