

Databáze

Databáze je propracovaný systém pro *ukládání dat* a jejich *následné zpracování*. Obsahuje data uložená na *paměťovém médiu*. Tato data mezi sebou mají určité *vztahy* a jsou určitým způsobem *členěna*. V širším pojetí spadají do pojmu databáze i *nástroje*, které s daty pracují (ukládají je, mění je a mažou je).

Na internetu se databáze používají při *programování webových aplikací*. Například tento slovníček je tvořen databází jednotlivých pojmů a skriptem, který s nimi pracuje (vypisuje, řadí a propojuje). Tato data by samozřejmě bylo možné uložit i do souboru, databáze však obvykle fungují mnohem rychleji, bývají optimalizovány pro přístup více uživatelů (návštěvníků stránek) a obsahují mechanismy, které práci s daty usnadňují. Opravdu složité aplikace, jako např. *redakční systém* či *internetový obchod* se bez využívání databáze neobejdou.

- Typy databáze:
 - Relační (níže)
 - Objektové - umožňují skladování dat s libovolnou strukturou
 - Speciální - (např Hierarchická - data uspořádána ve stromové struktuře, Síťová)

Relační databáze

Relační databáze = *kolekce dat*, která slouží pro popis reálného světa (např. evidence školní knihovny, sklad chemikálií, evidence studentů). Nechová jen jako běžné úložiště dat, ale díky *aplikační logice systému*, který databázi řídí, je schopen svá data rychle *třídít*, *vyhledávat* v nich a *řadit* je podle různých kritérií.

Relace = samostatná tabulka, která se skládá ze sloupců a řádků. Každý sloupec (atribut) má předem definovaný datový typ (např. celé číslo, řetězec, datum, logická hodnota ...). Složitějším příkladem relace pak je propojení více takovýchto tabulek.

Index = slouží ke zrychlení vyhledávacích a dotazovacích procesů v databázi, definování unikátní hodnoty sloupce tabulky.

Primární klíč = pole nebo kombinace polí, jednoznačně identifikující každý záznam v databázové tabulce. Žádné pole, které je součástí primárního klíče, nesmí obsahovat hodnotu NULL. Každá tabulka má mít definovaný právě jeden primární klíč.

Cizí klíč = (v prostředí relačních databází) u tabulky vytvoří spojení jednoho nebo více jejích sloupců se sloupcem nebo sloupci jiné („cizí“) tabulky

Entita = prvky reálného světa (např. člověk, stroj, vyučovaný předmět, město), který je popsán svými charakteristikami (vlastnostmi). Ty se většinou označují jako *atributy* (např. jméno, příjmení, stav, plat, hmotnost).

- Vztahy mezi entitami:
 - **1:1** - např. jeden člověk má uloženy stejné osobní údaje na městském úřadě stejně jako na správě sociálního zabezpečení
 - **1:N** - např. jeden člověk vlastní *n* platebních karet (jedna karta může mít pouze jednoho vlastníka)
 - **M:N** - bez omezení, např. jedna kniha má více autorů, zároveň ale autoři napsali více knih než tu jednu

Referenční integrita

Referenční integrita je nástroj databázového stroje, který pomáhá udržovat vztahy v relačně propojených databázových tabulkách.

Definuje se jako vztah mezi dvojicí tabulek. Tabulka, v níž je pravidlo referenční integrity uvedeno, se nazývá *podřízená tabulka*. Tabulka, jejíž jméno je v integritním omezení uvedeno, je *nadřízená tabulka*. V nadřízené i podřízené tabulce je určen sloupec nebo skupina sloupců tvořící *klíč*. Pravidlo referenční integrity vyžaduje, aby pro každý záznam v podřízené tabulce, jehož klíč nemá hodnotu NULL, existoval v nadřízené tabulce záznam se stejným klíčem. Klíč v nadřízené tabulce musí být unikátním indexem, klíč v podřízené tabulce musí být nějakým indexem.

Při manipulaci s daty se referenční integrita projevuje takto:

- Přidáváte-li záznam do podřízené tabulky (resp. přepisujete-li hodnotu sloupce z klíče), kontroluje se existence stejného klíče v nadřízené tabulce.
- Při smazání hodnoty sloupce z klíče (zrušením záznamu nebo přepsáním hodnoty) v nadřízené tabulce se kontroluje, zda v podřízené tabulce není záznam se stejnou hodnotou klíče.

System řízení báze dat (SŘBD)

Databázový systém, softwarové vybavení, které zajišťuje práci s databází, tzn. tvoří rozhraní mezi aplikačními programy a uloženými daty. Musí být jednak schopen efektivně pracovat s velkým množstvím dat, ale také musí být schopný řídit a definovat strukturu těchto perzistentních dat.

Relational DataBase Management System (RDMS)

Síťové služby (servery), zpracovávající současně více požadavků na data od mnoha uživatelů do více databází.

- Typy:
 - *freeware, shareware* - MySQL, PostgreSQL, Firebird
 - *licencované* - Oracle, MS SQL server, Sybase

Structured Query Language (SQL)

Strukturovaný dotazovací jazyk sloužící pro tvorbu univerzálních dotazů v databázích, zakládání tabulek, ošetření přístupu k datům, sdílení dat nebo třeba pro zabezpečení databází. SQL se většinou používá v kombinaci s některým výkonnějším programovacím jazykem.

SQL management software - univerzální nástroje pro správu dat SQL databází - *PHPMyAdmin, Adminer*

MS Access = nástroj pro správu relačních databází, který umí pracovat s daty MS-Jet, MS SQL server, Oracle a je součástí kancelářského software MS Office, jednoduché uživatelské prostředí, bez znalostí SQL jazyka

Základy jazyka vznikly v 70. letech 20. století při výzkumu relační databáze firmou IBM - jazyk *SEQUEL*. Dalšími vývoji (např. spolupráce s firmou Oracle) vznikl jazyk dnešní podoby.

- Typy dotazů:
 - Dotazy manipulující s daty:

- SELECT - příkaz vrací množinu záznamů z jedné a nebo více tabulek
- INSERT - přidá do tabulky relační databáze nový záznam
- UPDATE - příkaz upravuje data (záznamy) v relační databázi. Může
- DELETE - příkaz, sloužící k odstranění záznamů z tabulky relační databáze
- Dotazy pro definici dat:
 - CREATE - je příkaz , který slouží k vytváření databázových objektů.
 - CREATE TABLE - nejběžnější, vytvoření tabulky
 - ALTER - je příkaz , který slouží ke změně databázových objektů.
 - DROP - je příkaz , který slouží k odstranění databázových objektů
- Dotazy pro pro definici práv k datům:
 - GRAND - příkaz, kterým lze v relačních databázích nastavit přístupová práva k jednotlivým tabulkám
 - REVOKE - příkaz, kterým lze odebrat přístupová práva k jednotlivým tabulkám

Aplikační model databázové aplikace (obrázek propojení tabulek) - [klikni zde](#)

Příklady SQL

Vytvoření tabulky

```
CREATE TABLE Adresar (Jmeno varchar(50), Mail varchar(50), Adresa
varchar(50), );
```

Přidání hodnoty

```
INSERT INTO Adresar (jmeno, email, adresa) VALUES ('Karel',
'karel@praha', 'Praha 5');
```

Výpis tabulky

```
SELECT * FROM Adresar;
```

Podmíněný výběr

```
SELECT * FROM Adresar WHERE Jmeno LIKE 'Karel';
```

Příklady MySQL

```
// navazani vlakna k MySQL
$spojeni=@mysql_connect("host","user","password");

// výběr uživatelské databáze
@mysql_select_db("userdb",$spojeni);
$result = mysql_query("SELECT * FROM zbozi", $spojeni);
```

Externí odkazy

- [Wikipedia - Databáze](#)
- [Wikipedia - Relační databáze](#)

