

Relační databáze

Pojmy

- **databáze** = množina informací
- **relační databáze** - množina informací uspořádaná podle vazeb (relací) mezi nimi, data jsou uspořádána do **entit** (= tabulky), které mají dané **atributy** (= jednotlivé sloupce) a v nichž jsou v řádcích dané záznamy
- **Atribut** = sloupeček tabulky, atribut je určený svým datovým typem (celé číslo, řetězec, desetinné číslo, datum,...), každý atribut má svou doménu (= rozsah hodnot, kterých může nabývat, např. atribut výplata by neměl být záporný takže má rozsah od 0 do maximální velikosti datového typu).

Technologie

Všechny tyto databáze jsou postavené na jazyku SQL (Structured Query Language), který byl vyvinut firmou IBM. Liší se akorát v implementaci ukládání dat a přidání vlastních prvcích.

- **MySQL** - vyvinuta firmou Sun Microsystems (dnes koupená Oraclem), má dvojí licencování (GPL a zároveň i komerční licenci)
- **MariaDB**
- **PostgreSQL** - open source databáze, která je primárně vyvíjena pro unixové systémy (existují však i balíčky pro systém Windows)
- **SQLite** - databáze šířená pod licencí Public domain, specifická je tím, že nejde o klasickou databázi klient-server, ale o malou knihovnu, kterou vývojář přidá ke své aplikaci, tím pádem se dá použít k uložení dat na klientském zařízení
- **MS SQL** - komerční databázový systém od firmy Microsoft
- **Oracle** - komerční databázový systém od firmy Oracle
- **Azure SQL** a **Amazon Relational Database Services** - databázové systémy sloužící pro ukládání dat do cloudu (buď Amazon Cloud nebo Microsoft Azure)

Vazba

Vztah mezi daty v jednotlivých entitách (= tabulkách).

1:1

Pro každý záznam v tabulce první existuje **pouze jeden** záznam v tabulce druhé.

Příklad: Mám tabulku občané a tabulku rodná čísla. Každý občan má pouze jedno rodné číslo a každé rodné číslo má pouze jednoho svého občana.

1:n

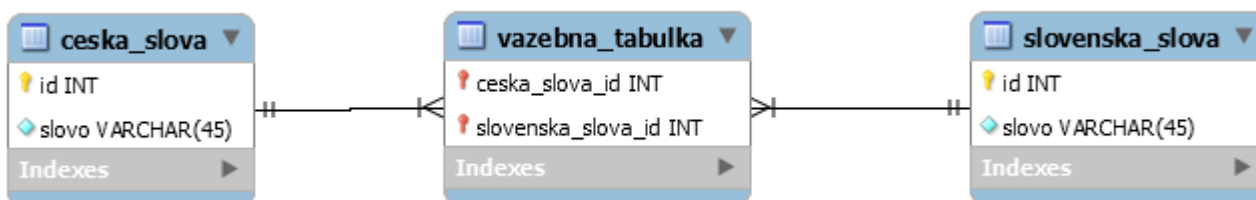
Každý záznam v první tabulce může být spojen s **více záznamy** v tabulce druhé.

Příklad: Mám tabulku psů a tabulku majitelů. Každý pes má svého pouze jednoho majitele, ale jeden majitel může mít vícero psů.

n:m

Více záznamů v první tabulce může být spojeno s **více záznamy** v tabulce druhé. Tato problematika se řeší tzv. **vazebnou tabulkou**, která obsahuje všechny relace mezi záznamy.

Příklad: Slovník. Mám tabulku obsahující všechna česká slova, mám druhou tabulku obsahující všechna slovenská slova a mám vazebnou tabulku, která obsahuje všechny možné překlady pro jednotlivá slova.



Klíče

Primární klíč

Atribut (=slopec), který jednoznačně identifikuje každý záznam tabulky, tím pádem by měl být pro každý záznam tabulky unikátní a neměl by mít nulovou hodnotu.

Příklad: Primárním klíčem každého občana by mohlo být rodné číslo, protože ho má každý unikátní.

Cizí klíč

Atribut který slouží pro vyjádření vztahu mezi entitami. Můžeme to vidět na příkladu níže. Cizí klíč nemusí být unikátní (u vazeb 1:n a n:m by to ani nebylo možné).

| Občané | | |
|--------------------|--------|----------|
| ID (primární klíč) | Jméno | Příjmení |
| 1 | Michal | Mrnuščík |
| 2 | Adolf | Ryšavý |
| 3 | David | Fosilní |
| 4 | Greg | Anderson |
| 5 | Andrej | Bubiš |
| 6 | Miloš | Tuzeman |

| Psi | | |
|--------------------|-------------------------|--------|
| ID (primární klíč) | ID Majitele (cizí klíč) | Jméno |
| 1 | 1 | Dag |
| 2 | 1 | Adéla |
| 3 | 6 | Becher |
| 4 | 5 | Rex |

Normální formy

= normy, které by bylo dobré dodržovat při návrhu tabulky

1 NF

„Každý atribut by měl obsahovat pouze **atomické hodnoty**.“

To znamená, že informace obsažené v atributu by už neměly být rozložitelné na menší jednotky.

Příklad: Neměl bych mít atribut JmenoAPrijmeni, ale měl bych používat dva atributy jeden atribut pro Jmeno a druhý pro Prijmeni, protože nikdy nevím, kdy budu potřebovat jen jméno nebo pouze příjmení.

2 NF

„Každý atribut, který není primárním klíčem, by měl být na primárním klíči závislý.“

To znamená, že pokud se mi informace v jednom atributu u více záznamů opakují (-> nejsou závislé na primárním klíči), měl bych si pro ně vytvořit speciální tabulku, která bude obsahovat tyto hodnoty, a v původní tabulce akorát dát cizí klíč odkazující na tuto tabulku.

Příklad: Mám tabulku obsahující moje zboží, které prodávám. V tabulce mám jako atribut jméno dodavatele a jeho číslo, pokud mám více zboží od stejného dodavatele dochází k opakování těchto hodnot. Proto vytvořím novou tabulku obsahující dodavatele a do původní tabulky dám cizí klíč odkazující na tabulku dodavatelů.

3 NF

„Všechny neklíčové atributy by na sobě měly být navzájem nezávislé.“

To znamená, že neklíčové atributy, které by na sobě byly závislé bychom měli odsunout do jiné tabulky.

Příklad: Mám tabulku obsahující zaměstnance a jejich pozice a jejich platy. Zaměstnanci na stejných pozicích by měli mít stejné platy (pokud to nejsou ženy), to znamená, že neklíčové atributy plat a pozice jsou na sobě závislé. Vyřeším to tak, že si vytvořím jinou tabulku obsahující pozice a k nim přidělené platy.

Základní databázové příkazy

- DDL = Data definition language
 - CREATE TABLE
 - ALTER TABLE
 - DROP TABLE
 - CREATE DATABASE
 - CREATE VIEW
 - ...
- DML = Data manipulation language
 - SELECT
 - INSERT
 - UPDATE
 - DELETE
- **SELECT** - vybere data
- **INSERT** - přidá data
- **WHERE** - podmínka pro výběr/změnu dat
- **ORDER BY** - seřadí vybraná data (ASC - vzestupně, DESC - sestupně)
- **OR, AND** - logické operátory
- **DELETE** - smaže data z databáze
- **CREATE TABLE** - vytvoří novou tabulku

From:

<https://old.gml.cz/wiki/> - **GMLWiki**

Permanent link:

<https://old.gml.cz/wiki/doku.php/informatika:maturita:15a>

Last update: **12. 03. 2021, 14.24**

