

Databáze, přednáška 1

Motivace, ER modelování a relační model dat

Tomáš Urbanec

Katedra informatiky, Univerzita Palackého v Olomouci

26. září 2025

Úvod

Kontakty

- Přednášející: Tomáš Urbanec
- Kancelář: 5.044
- Email: tomas.urbanec@upol.cz
- Stránky katedry: inf.upol.cz/lide/tomas-urbanec
- Konzultační hodiny:
 - ▶ středa: 12:15 - 12:45
 - ▶ **pátek: 9:45 - 10:15**
- web: apollo.inf.upol.cz/~urbanec/teaching/2025/xdata/databases.html

Obsah kurzu

- Modelování databáze
- Relační model dat
- SQL (PostgreSQL)
- Dokumentový model dat
- MongoDB
- Fulltextové vyhledávání
- Elasticsearch

Přednášky

- ① Databáze a relační model dat
 - ▶ motivace,
 - ▶ ER modelování a relační model dat
- ② Relační model podruhé,
 - ▶ relační algebra a dotazy v SQL
- ③ Relační model potřebí
 - ▶ integritní omezení
 - ▶ návrh databáze v relačním modelu
- ④ Dokumentový model
 - ▶ motivace a základy dokumentového modelu
- ⑤ Dokumentový model podruhé,
 - ▶ MongoDB a základní dotazy
- ⑥ Fulltextové vyhledávání a shrnutí
 - ▶ Elasticsearch a základní dotazy

Zdroje

- web předmětu
- materiály dr. Laštovičky k obdobnému kurzu pro denní studium
- knihovna PřF a KI
 - ▶ široká nabídka knih k relačnímu modelu a relačním databázím

Motivace a vývoj databází

Proč existuje více typů databází?

- různé aplikace mají odlišné požadavky na ukládání a práci s daty
- kompromis mezi rychlostí, flexibilitou, škálovatelností a konzistencí
- rostoucí objem dat a nové potřeby

Stručná historie databázových modelů:

- 60. léta: **hierarchický model** - stromová struktura, odkazy
- 70. léta: **síťový model** - flexibilnější vazby mezi záznamy
- 70. léta (dodnes): **relační model** (Codd, 1970) - relace, poté tabulky a SQL
- 90. léta: **objektově-orientované DB** - kombinace s OOP
- nové milénium:
 - **NoSQL databáze** - dokumentové, klíč–hodnota, grafové, ...
 - **NewSQL a multimodelové DB** - kombinace relačního základu s NoSQL
 - **Distribuované databáze** - škálovatelnost, dostupnost, ale má to svou cenu

Průvodní příklad

- Eshop s elektrem
- Snadno si představíme
- Skutečný problém
- Snadno a účelně využijeme všechny představené modely i technologie
- Uvidíme, v čem jsou silné a v čem ne.

Objektovo vztahové (ER) modelování

Objektovo vztahové (ER) modelování

- **Objektovo vztahové modelování** je metoda návrhu databází.
 - ▶ Častěji nazývané **Entity–Relationship (ER) modelování**.
- Grafická a konceptuální reprezentace dat a jejich vztahů.
- Pomáhá pochopit a navrhnout strukturu databáze ještě před její implementací.
- Hlavní stavební bloky:
 - ▶ **Objekty** – objekty reálného světa (např. Zákazník, Produkt, Nákup)
 - ▶ **Vlastnosti** – vlastnosti entit (např. jméno zákazníka, cena produktu)
 - ▶ **Vztahy** – propojení mezi entitami (např. „Zákazník provedl Nákup“)

Základní pojmy

- **Samostatné jevy**

- ▶ To, co jsme schopni pozorovat přímo.
- ▶ Př. (konkrétní) zákazník, produkt, platba, nákup, sleva, ...
- ▶ Modelovány **objekty** (entitami)

- **Průvodní jevy**

- ▶ Samostatně neexistují, jsou navázány na samostatné jevy.
- ▶ **Vlastnost**
 - ★ Průvodní jev pozorovatelný na objektu
 - ★ Př. datum nákupu, stav platby, výše slevy, ...
 - ★ Modelovány **atributy**
- ▶ **Vztah**
 - ★ Průvodní jev pozorovatelný mezi více objekty
 - ★ Př. (daný) Zákazník *provedl* (danou) platbu,
 - ★ Př. (daná) Sleva byla *aplikována* na (daný) nákup, ...
 - ★ Modelovány **vztahy**

Základní pojmy

- **Typ objektů** - pojmenovaná množina všech objektů s danou určující vlastností.
 - ▶ Př. typ objektů *Produkt* je množina všech věcí, co prodáváme.
 - ▶ Př. typ objektů *Zákazník* je množina všech lidí, kteří u nás mají účet.
 - ▶ $\text{Zákazník} = \{x \mid x \text{ má účet v eshopu}\}$
- **Typ vztahů** - pojmenovaná relace mezi typy objektů s danou určující vlastností.
 - ▶ Př. typ vztahů *Provedl* je relace mezi typem objektů *Zákazník* a typem objektů *Nákup*
 - ▶ $\text{Provedl} = \{\langle x, y \rangle \in \text{Zákazník} \times \text{Nákup} \mid x \text{ provedl } y\}$
- **Atribut** - pojmenované zobrazení z typu objektů do spočetné množiny (*domény atributu*).
 - ▶ Př. atribut *Cena* udává ceny *Produktu*.
 - ▶ $\text{Cena}: \text{Produkt} \rightarrow \mathbb{N}$

Kardinalita typu vztahů

- **Kardinalita typu vztahů** je výraz tvaru $x : y$, kde $x, y \in \{0-1, 1, N\}$.
- Kardinalita zachycuje některé naše předpoklady o realitě, které musí být zachovány i v modelu.
- Př. *Provedl* mezi *Zákazník* a *Nákup*
 - ▶ Každý nákup provedl právě jeden zákazník.
 - ▶ Naopak ale každý zákazník může provést N nákupů.
 - ▶ *Provedl* mezi *Zákazník* a *Nákup* je tedy kardinality $1 : N$.

Kardinalita typu vztahů

- Máme-li typ vztahů T s kardinalitou $x : y$ mezi typy objektů O_1 a O_2 , pak
 - ▶ Pokud $x = 1$, pak pro každé $o_2 \in O_2$ existuje právě jedno $o_1 \in O_1$, kde $\langle o_1, o_2 \rangle \in T$.
 - ▶ Pokud $y = 1$, pak pro každé $o_1 \in O_1$ existuje právě jedno $o_2 \in O_2$, kde $\langle o_1, o_2 \rangle \in T$.
 - ▶ Pokud $x = 0-1$, pak pro každé $o_2 \in O_2$ existuje nejvýše jedno $o_1 \in O_1$, kde $\langle o_1, o_2 \rangle \in T$.
 - ▶ Pokud $y = 0-1$, pak pro každé $o_1 \in O_1$ existuje nejvýše jedno $o_2 \in O_2$, kde $\langle o_1, o_2 \rangle \in T$.
- Tedy platí následující
 - ▶ Pokud $x = 1$, pak T^{-1} je zobrazení $O_2 \rightarrow O_1$.
 - ▶ Pokud $y = 1$, pak T je zobrazení $O_1 \rightarrow O_2$.
 - ▶ Pokud $x = y = 1$, pak T je bijekce.
 - ▶ Pokud $x = 0-1$, pak T^{-1} je částečné zobrazení $O_2 \rightarrow O_1$.
 - ▶ Pokud $y = 0-1$, pak T je částečné zobrazení $O_1 \rightarrow O_2$.

Objektovo vztahový model

- Objektovo vztahový model je dán orientovaným grafem, kde
 - 1 každý vrchol značí typ objektů, typ vztahů, nebo atribut.
 - 2 každý vrchol má název.
 - 3 do vrcholu pro typ vztahů vedou hrany pouze z vrcholů pro typ objektů.
 - 4 z vrcholu pro typ vztahů vedou hrany pouze do vrcholů pro typ objektů.
 - 5 z vrcholu atributu vede právě jedna hrana a to do vrcholu pro typ objektů.
 - 6 vrchol pro typ objektů má za sousedy pouze vrcholy pro typ vztahů nebo atribut.
 - 7 hrana mezi vrcholem pro typ objektů a typ vztahů je ohodnocena prvkem množiny $\{0-1, \infty, N\}$.
- My se omezíme na binární typy vztahů, tj. pouze mezi dvěma typy objektů.
 - ▶ Pak z a do každého vrcholu pro typ vztahů vede právě jedna hrana a to z a do vrcholů pro typy objektů.

Objektovo vztahový model - ER diagramy

- Vrcholy pro objektový typ zakreslujeme obdélníky.
- Vrcholy pro vztahové typy zakreslujeme kosočtverci.
- Vrcholy atributů zakreslujeme elipsami.
- Kardinalitu typu vztahů zapisujeme na odpovídající hrany.
- Příklad: *Zákazník*, *Nákup* a *Provedl*

Jednoznačná identifikace objektů

- K jednoznačné identifikaci objektu v rámci typu objektů slouží takzvané **klíče**.
- Nechť $\{a_1, \dots, a_n\}$ je množina atributů typu objektů O a pro každé i , kde $1 \leq i \leq n$ je D_i doména atributu a_i .
 - ▶ Množinu $K = \{a_1, \dots, a_n\}$ nazveme **nadklíčem** typu objektů O , pokud pro každý přípustný stav reality je zobrazení $f: O \rightarrow D_1 \times \dots \times D_n$ dané předpisem $f(o) = \langle a_1(o), \dots, a_n(o) \rangle$ injektivní.
 - ▶ Nadklíč typu O nazveme **kandidátní klíč** O , pokud žádná jeho vlastní podmnožina již není nadklíč O .
 - ▶ **Primární klíč** typu O je jeden zvolený z kandidátních klíčů O .
- Klíče poskytují jednoznačnou identifikaci objektů daného typu.
 - ▶ Tj. jsou-li si dva objekty rovny na všech attributech z nějakého klíče, pak jsou stejné.
 - ▶ Kandidátní klíče jsou minimální nadklíče. Tj. nelze je dále zmenšovat.
 - ▶ Primární klíč volíme z kandidátních klíčů dle situace.
- Názvy atributů z primárního klíče v ER diagramu podtrhneme.

Příklad

- Navrhne si základ eshopu
- Produkty, Zákazníci, Platby, Nákupy
- Cvičení
 - ▶ Zkuste si návrh rozšířit.
 - ▶ Zkuste u všech objektových typů určit nadklíče a kandidátní klíče.
 - ▶ Podívejte se na cvičení v materiálech dr. Laštovičky

Relační model dat

Relační model dat

- **Relační model** je teoretický základ pro relační databáze.
- Matematický model založený na pojmu relace (tabulky).
- Poskytuje formální rámec pro ukládání a manipulaci s daty.
- Hlavní prvky:
 - ▶ **Relace (tabulka)**: množina n -tic (řádků) nad schématem (množinou atributů),
 - ▶ **Atribut**: pojmenovaný sloupec s doménou hodnot,
 - ▶ **Klíče**: určují jednoznačnost řádků a vztahy mezi tabulkami,
 - ▶ **Operace**: relační algebra (např. projekce, selekce, spojení).
- Tím se budeme zabývat.

Relační databáze

- **Relační databáze** je systém (přibližně) implementující relační model dat.
- Softwarová aplikace spravující data uložená v tabulkách.
 - ▶ Př. PostgreSQL, MariaDB, MySQL, SQLite, Oracle, ...
- Umožňuje efektivně pracovat s daty, zajišťuje jejich konzistenci, integritu a bezpečnost.
- Hlavní vlastnosti:
 - ▶ data organizovaná v tabulkách (relacích),
 - ▶ využití klíčů pro vazby mezi tabulkami,
 - ▶ podpora SQL jako dotazovacího jazyka,
 - ▶ transakční zpracování, podpora souběžného přístupu.
- Na tom si to ukážeme (PostgreSQL).

Základní pojmy relačního modelu

- **Atribut** je vhodný řetězec písmen a podtržítka
 - ▶ Př. jméno, datum, věk, ID...
- **Doména atributu** D_y je spočetná množina hodnot, které může atribut nabývat.
 - ▶ Př. $D_{\text{věk}} = \mathbb{N}$, $D_{\text{počet_nohou}} = \{1, \dots, 100\}$, ...
- **(Tabulkové) schéma** je konečná podmnožina množiny atributů.
 - ▶ Př. $S = \{\text{jméno, datum, věk, ID}\}$, ...
- **Buňka** (*komponenta*) je dvojice $\langle y, d \rangle$, kde y je atribut a $d \in D_y$.
 - ▶ Př. $\langle \text{jméno, Jan} \rangle$, $\langle \text{věk, 10} \rangle$, ...
- **Řádek** (*n-tice*) nad schématem S je množina buněk, která každému atributu z S přiřazuje právě jednu hodnotu.
 - ▶ Př. $\{\langle \text{jméno, Jan} \rangle, \langle \text{věk, 10} \rangle, \langle \text{ID, 1} \rangle \langle \text{datum, 2025/09/26} \rangle\}$
- **Vlastnost (řádků)** nad S je výroková formule $V(r)$ kde r je řádek nad S .

Základní pojmy relačního modelu

- **(Datová) tabulka** (*relace*) nad S je konečná množina řádků nad S .
 - ▶ Př. $\{\{\langle \text{jméno}, \text{Jan} \rangle, \langle \text{věk}, 10 \rangle, \langle \text{ID}, 1 \rangle \langle \text{datum}, 2025/09/26 \rangle\}, \{\langle \text{jméno}, \text{Tom} \rangle, \langle \text{ID}, 42 \rangle, \langle \text{věk}, 20 \rangle, \langle \text{datum}, 2020/20/20 \rangle\}\}$
- **Určující vlastnost** tabulky T je vlastnost $V(r)$, která přesně charakterizuje množinu řádků T .
 - ▶ Př. “Návštěva s ID ID dne datum byl člověk ve věku věk let jménem jméno.”
- **Tabulková proměnná** nad schématem S je proměnná, jejíž hodnotou je tabulka nad S .
 - ▶ Př. Tabulková proměnná návštěva nad S .
- **Základní tabulka** je tabulková proměnná, jejíž hodnotu lze přímo měnit.
 - ▶ Př. Tatáž tabulková proměnná návštěva nad S .
- **Grafická tabulka** je zobrazení datové tabulky v tabulkovém formátu s pořadím řádků a atributů.

SQL

- **SQL (Structured Query Language)** je standardní jazyk pro práci s relačními databázemi.
- Deklarativní jazyk pro vytváření a manipulaci s tabulkami a daty.
- Umožňuje ukládat, vyhledávat, měnit a mazat data v databázi.
- Praktický nástroj pro implementaci návrhu databáze (např. podle ER modelu)
- Hlavní oblasti:
 - ▶ **DDL (Data Definition Language):** definice struktury databáze (CREATE, ALTER, DROP)
 - ▶ **DML (Data Manipulation Language):** manipulace s daty (INSERT, UPDATE, DELETE)
 - ▶ **DQL (Data Query Language):** dotazování nad daty (SELECT)
 - ▶ **DCL (Data Control Language):** řízení přístupů (GRANT, REVOKE)

SQL tabulka

- Podobná datové tabulce, ale může určovat pořadí sloupců i řádků, mít duplicitní řádky a některé buňky bez hodnoty.
- SQL tabulka je zároveň datovou tabulkou, pokud splňuje:
 - ▶ názvy sloupců jsou jedinečné,
 - ▶ nejsou duplicitní řádky,
 - ▶ nevyužíváme pořadí řádků ani sloupců,
 - ▶ všechny buňky mají hodnotu.
- **Skalární typ** je typ, který dává jméno množině hodnot, např. integer, text.
- Atribut (sloupec) SQL tabulky má doménu určenou skalárním typem.

PostgreSQL

- Pokročilý open-source relační databázový systém.
- Databázový server, který implementuje SQL standard a rozšiřuje jej o další funkce.
- Umožňuje efektivně ukládat a spravovat velké objemy dat, poskytuje transakční spolehlivost a rozšiřitelnost.
- Hlavní vlastnosti:
 - ▶ open-source, multiplatformní
 - ▶ silná podpora standardu SQL
 - ▶ rozšiřitelné typy, funkce a indexy
 - ▶ podpora transakcí a integrity dat
 - ▶ bohatý ekosystém nástrojů (např. psql, rozšíření pro VS Code)
- Lze provozovat na svém stroji (relativně snadné)
- Budete jej mít dostupný na serveru katedry (informace na webu a mailem)
- Cvičení
 - ▶ Zprovozněte si PostgreSQL (lokálně i na serveru)

Práce s tabulkami v PostgreSQL

- Praktická ukázka základní práce s tabulkami v PostgreSQL
- Cvičení
 - ▶ Zkuste si ukázané dotazy sami.
 - ▶ Podívejte se na web předmětu, kde jsou další informace a příklady.
 - ▶ Podívejte se na cvičení v materiálech dr. Laštovičky.

Převod ER modelu do relačního modelu

Changelog

- ① 26.9.2025
 - ▶ chyba na slidu 17 (nadklíč místo kandidátní klíč)
 - ▶ drobnosti