

Databáze

Osnova

- Co to je?
- Historie
- Dělení
 - SQL/NoSQL
 - hierarchická / síťová / relační / objektová / objektová relační
- Relační databáze
- Integrita dat
 - pravidla ukládání dat, aby se nic nepoškodilo
- Příklady známých databází

Co je to databáze

- systém souborů s pevně danou strukturou záznamů
 - příklad: jako databázi si můžeme představit tabulku, tabulka má nějaké dané sloupce (struktura) a pak do ní přidáváme soubory (řádky)
 - v jedné databázi by však takovýchto tabulek bylo většinou více
- většinou pojem zahrnuje nejen databázový software (např. PostgreSQL), ale i softwarové prostředky pro přístup k databázi a manipulaci s daty (např. pgAdmin)
 - software řízení databáze
 - běžně: databáze = databáze + software řízení databáze
- zjednodušeně: je to věc co obsahuje nějaký informace a taky to jak jsou ty informace propojeny
- vztah aplikace s databází: klient-server

Historie

- úplní předchůdci: kartotéky
 - organizace podle kritérií (abeceda, ...)
 - manuální práce
- potom mechanické řešení
 - děrné štítky
 - 1890 sčítání lidu v USA
- s rozmachem počítačů
 - strojový kód nedostatečný → první programovací jazyky (COBOL)
- první databáze na sálových počítačích (1965)
- (70. léta)
 - pracovní skupiny zpracovávali databáze
 - první relační databáze
 - vznik SQL

Dělení

- z hlediska ukládání dat a vazeb
 - hierarchická*
 - data uspořádaná ve stromové struktuře
 - (podobné jako soubory na počítači - složky mohou obsahovat více podsložek a takto se postupně větvit)
 - síťová*
 - data uspořádaná v síťové struktuře (umožňuje mít více otců)
 - **relační**
 - tabulky provázané pomocí vztahů
 - více viz [Relační databáze](#)
 - objektová*
 - informace reprezentována ve formě objektů, dobré pro komplexní data
 - objektová relační

- data v tabulkách, ale OOP principy - definování datových typů a metod
- z hlediska struktury dat a dotazovacím jazyce
 - SQL
 - využívají dotazovací jazyk SQL
 - relační databáze
 - skvělé pro aplikace vyžadující vysokou integritu dat a složité transakce
 - NoSQL
 - nerelační databáze
 - např. dokumentové (MongoDB - data uložena jako JSON)
 - skvělé pro velmi velké objemy dat a rychlejší pro jednodušší operace

(*) tyto modely se považují za spíš historické

Relační databáze

- sloupec je vlastnost a řádek je záznam
- klíče
 - primární klíč (private key, PK)
 - identifikátor daného záznamu, často číslo (+záznam => +číslo)
 - cizí klíč (foreign key, FK)
 - odkazuje na nějaký jiný záznam

Velmi zjednodušený model:

Lidé

ID člověka (PK)	Jméno	Email
1	Pepa Novák	pepa.novak@example.com
2	Ivan Ištván	ivan.istvan@example.com
3	Jaroslav Jaroměř	jaroslav.jaromer@example.com

Domácí zvířátko

ID druhu (PK)	Název	Mazel
1	Kočka	false
2	Pes	true

Vlastníci zvířátek

ID vztahu (PK)	ID člověka (FK)	ID druhu (FK)
1	2	1
2	1	2

3	2	2
---	---	---

Co lze vyčíst? Pepa vlastní psa, Ivan vlastní kočku i psa, Jaroslav nevlastní zvířátko

- dobré na popisování reálného světa (všude jsou vztahy)
- pro popis tabulek a jejich vztahů se používá ER diagram (*entity relationship*)
- entita
 - skupina objektů reálného světa (zvíře, člověk)
 - instance entity = konkrétní objekt (kočka, Pepa) = záznam v tabulce → řádek
 - vlastnosti → sloupec (barva očí, ...)
 - vlastnost entity → konkrétní hodnota (modrá)
 - identifikace entity = název tabulky (Lidé, Zvířata)
 - identifikace instance entity = primární klíč - atribut
- vztah
 - slouží ke svázání entit
 - 4 typy (čteme x na x)
 - 1-N
 - člověk si může půjčit více knih, ale právě kniha může být vždy u jednoho člověka
 - Josef Novák
 - Fimfárum
 - Odysea
 - 1-1
 - člověk má právě jeden občanský průkaz
 - Josef Novák – občanský průkaz Josefa Nováka
 - M-N
 - pacient chodí k více doktorům a doktor má více pacientů
 - relační databáze toto neumožní, je třeba přidat pomocnou tabulku (i viz ukázka s lidmi a zvířaty)
 - Pacienti
 - Doktoři
 - Pacienti chodí k doktorům (FK pacienta, FK doktora)
 - bez vztahu
 - FK se používají, aby se neukládala data v databázi duplicitně
 - pro získání vlastností entit se pak v SQL používá JOIN - tj. připojení tabulek k sobě na základě nějaké vlastnosti

Integrita dat

- data jsou uložena podle pravidel
- integritní omezení
 - zabraňování vkládání nesprávných dat / poškození záznamů
 - dvě entity nesmí mít stejný PK (entitní)
 - dodržování datových typů u vlastností (doménová)
 - existují data ke kterým referujeme (referenční)
 - třeba jestli pořád existuje tabulka zvířat
 - co uděláme když se poruší některá omezení? (aktivní referenční)
 - smažeme entitu, nastavíme NULL?

Příklady známých databází

- PostgreSQL
 - open-source, objektově-relační databázový systém
- MySQL
 - open-source relační databáze, široce používaná pro webové aplikace
- MongoDB
 - NoSQL dokumentová databáze
- Microsoft SQL Server
 - komerční relační databáze od společnosti Microsoft
- Oracle Database
 - proprietární multi-modelový systém pro správu databází
- SQLite
 - extrémně lehká databáze - jeden soubor
- Redis
 - in-memory úložiště datových struktur, používané jako databáze, cache a message broker (zprostředkovatel zpráv)
 - extrémně rychlé

Zdroje

- WEISER, Martin. *18 - Databáze.docx*. Praha: Gymnázium Litoměřická.
- Databáze. In: *Wikipedie* [online]. 2026 [cit. 15.02.2026]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Datab%C3%A1ze&oldid=25613451>
- NoSQL. In: *Wikipedie* [online]. 2023 [cit. 15.02.2026]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=NoSQL&oldid=23031616>
- Object–relational database. In: *Wikipedia* [online]. 2025 [cit. 15.02.2026]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Object%E2%80%93relational_database&oldid=1317276523
- Objektová databáze. In: *Wikipedie* [online]. 2025 [cit. 15.02.2026]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Objektov%C3%A1_datab%C3%A1ze&oldid=25416341
- Relací databáze. In: *Wikipedie* [online]. 2025 [cit. 15.02.2026]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Rela%C4%8Dn%C3%AD_datab%C3%A1ze&oldid=25315190