

Úvod do informačních technologií

databáze

Martin Trnečka

Katedra informatiky
Univerzita Palackého v Olomouci

Data

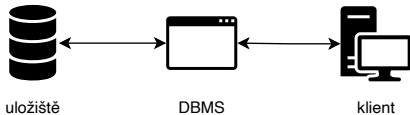
- data = informace
 - jak uložit?
 - jak spravovat? (čtení, zápis, vícenásobný přístup, oprávnění, analýza)
 - vlastní implementace vs. existující řešení
- typ dat
- struktura dat

Databáze

- kolekce dat
- „perzistentní“ uložení
- formální model vs. implementace
- DataBase Management System (DBMS)

DataBase Management System (DBMS)

- systém báze řízení dat, systém pro správu dat
- implementuje konkrétní *model dat*
- služby:
 - uložení dat, transakční zpracování a zabránění vzniku chyb (žurnálování)
 - integritní omezení (konzistence dat)
 - víceuživatelský přístup (neblokující)
 - bezpečnost dat (oprávnění, šifrování)
- *databázové schéma* = struktura dat
- různé role: vývojář, administrátor, uživatel
- jazyk pro def. dat (DDL), jazyk pro modifikaci dat (DML), dotazovací jazyk (QL)
- příklady: MariaDB, PostgreSQL, SQLite, MongoDB, Microsoft SQL Server, Oracle



Modely dat

■ souborový

- historicky první databáze
- záznamy uloženy v souborech a adresářích
- velmi jednoduché = snadno implementovatelné
- absence abstrakce a mnohdy DBMS

■ síťový

- dnes již málo používaný
- záznamy provázány odkazy
- grafová struktura
- znovu-objevení v podobě grafových databází (zmíníme později)

Modely struktury dat

■ hierarchický

- varianta/zjednodušení síťového modelu
- stromová struktura
- vztah potomek-rodíč
- 1:N, rodič může mít více potomků, potomek má právě jednoho rodiče

■ relační

- relace
- data uložena v tabulkách
- nejpoužívanější
- obvykle ve spojení s objektovým modelem = objektově-relační databáze

■ objektový

- data chápeme jako objekty

■ další modely

Odbočka: Modely dat

- populární dělení:
 - SQL = relační databáze
 - NoSQL = nerelační databáze

CSV formát

- data uloženy v souborech (obvykle přípona .csv)
- pevná struktura (hlavička)
- záznamy uloženy na řádcích
- hodnoty odděleny oddělovačem (obvykle čárka, ale i jiný znak)
- žádná abstrakce, omezené dotazování
- univerzální formát, velká podpora

Klíč-hodnota

- data ve formátu: klíč \rightarrow hodnota
 - klíč = omezený datový typ
 - hodnota = libovolná (i strukturovaná)
- operace: ulož, přečti, smaž
- omezené (zejména dotazování), ale velmi populární
- použití:
 - uchování nastavení
 - kešování
 - menší databáze
- návrh klíče:
 - jedinečný identifikátor (číselný, textový)
 - hash
 - časové razítko (time-stamp)
- zavedení struktury \rightarrow namespace

Semi-strukturovaná data

- poměrně vágní pojem
- data mají strukturu, ale ne tabulkovou
- samo-popisující → struktura dat je součástí dat samotných
- příklady:
 - XML
 - JSON
 - YAML
 - BSON

XML

- Extensible Markup Language (XML)
- textový formát
- původně zamýšlen pro službu WWW → neujal se
- univerzální, platformově nezávislý a značně rozšířený formát
- příklad:

```
<?xml version="1.0" encoding="UTF-8"?>
<poznamka>
  <od>Dave Barry</od>
  <nadpis>Database</nadpis>
  <telo>the information you lose when your memory crashes</telo>
</poznamka>
```

XML

- podobnost s jazykem HTML
 - XML je více striktní
 - XML je obecnější
- atributy v XML
- omezení struktury dat → XML schéma
- validita
- podpora (knihovny) v programovacích jazycích

Příklad: XML schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="poznamka">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="od" type="xs:string" />
        <xs:element name="nadpis" type="xs:string" />
        <xs:element name="telo" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Odbočka: XML

- transformace XML → jazyk XSLT
- dotazování → jazyk XPath
- parsování
 - DOM (Document Object Model)
 - SAX (Simple API for XML)

JSON

- JavaScript Object Notation
- odlehčený formát (kontrast s XML)
- syntaxe:
 - data ve tvaru "jméno" : "hodnota"
 - data se oddělují čárkou
 - složené závorky pro *objekty*
 - hranaté závorky pro *pole*
- omezení struktury dat → JSON schéma

```
{  
  "firstName": "Oliver",  
  "lastName": "Queen",  
  "alterEgo": [  
    {"name": "The Hood"},  
    {"name": "The Arrow"},  
    {"name": "Green Arrow"}  
  ]  
}
```

Grafové databáze

- uzly = záznamy, hrany = relace
- velmi flexibilní
- ideální v situacích kdy data mají povahu sítě
- například: sociální sítě, sémantický web, doprava

Dokumentové databáze

- data = sada dokumentů
- ideální v situacích kdy záznamy nemají pevnou strukturu
- vztahy mezi dokumenty na základě identifikátoru (primární klíč)
- například: webové stránky

Relační databáze

- revoluce
- benefit:
 - eliminace duplicit
 - zajištění konzistence dat
- data uložena v tabulkách
- sloupec = vlastnosti (atributy) záznamu
- doména atributu = přípustné hodnoty
- různá omezení atributu (typ, jedinečnost, ...)
- schéma databáze
- řádek = záznam v databázi (relace)
- řádky nesmí porušovat databázové schéma (lze chápat jako výhodu i nevýhodu)
- normalizace databáze → snížení redundance a zvýšení integrity dat
- příklad

Jazyk SQL

- dotazovací jazyk
- výběr dat

```
SELECT jmeno_sloupce [, jmeno_sloupce, jmeno_sloupce, ...] FROM tabulka WHERE podmínky
```

- * pro výběr všech sloupců
- WHERE část je nepovinná

- agregace dat

```
funkce(jmeno_sloupce)
```

- například: min, max, sum, avg

Jazyk SQL

- setřídění, limitace, seskupení (píše se za WHERE podmínku)

```
ORDER BY jmeno_sloupce  
LIMIT n  
GROUP BY jmeno_sloupce
```

- spojení dat (vnitřní)

```
SELECT tab_1.sloupec_1, tab_2.sloupec_2 FROM tabulka_1 JOIN tabulka_2 ON tab_1.sloupec_1 =  
    tab_2.sloupec_2 WHERE podmínky
```

- vnitřní, levé, pravé, plné, kartézské

Jazyk SQL

■ vytvoření tabulky

```
CREATE TABLE zakaznik (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    jmeno TEXT,  
    telefon INTEGER NOT NULL UNIQUE);
```

■ vložení řádku do tabulky

```
INSERT INTO zakaznik (id, jmeno, telefon) VALUES (  
    '11',  
    'Tomáš',  
    '123'  
);
```

Jazyk SQL

■ smazání řádku z tabulky

```
DELETE FROM zakaznik WHERE id=11;
```

■ smazání tabulky

```
DROP TABLE zakaznik;
```

Big data

- „buzzword“
- data mimo možnosti běžného zpracování
- různé definice
- různé nároky: objem, rychlost, různorodost, ...
- speciální prostředky: např. různé platformy nebo distribuované databáze