

## Kapitola 1: Úvod

- Účel databázových systémů
- Pohled na data
- Modely dat
- Jazyk pro definici dat (Data Definition Language; DDL)
- Jazyk pro manipulaci s daty (Data Manipulation Language; DML)
- Správa transakcí
- Správa ukládání dat
- Správce databáze
- Uživatelé databáze
- Celková struktura systému

### **Systém pro správu databáze (Database Management System DBMS)**

- Soubor dat ve vzájemném vztahu
- Sada programů pro přístup k datům
- DBMS obsahuje informace o jednom konkrétním „podnikání“
- DBMS poskytuje prostředí, které je pohodlné i účinné

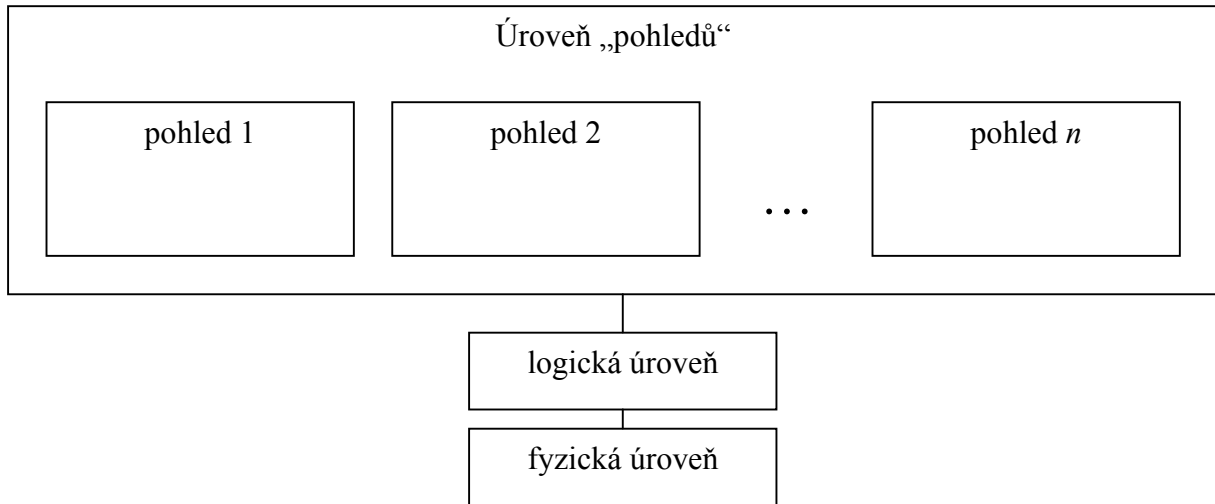
### **Účel databázových systémů**

Systémy pro správu databází byly vyvinuty kvůli zvládnutí následujících problémů při zpracování souborů v tradičních operačních systémech:

- Redundance a inkonsistence dat
- Problémy s přístupy k datům
- Izolace dat – různé soubory a formáty
- Problémy s integritou
- Jedinečnost (atomicita) aktualizací
- Současný přístup více uživatelů
- Bezpečnostní problémy

## Pohled na data

Architektura databázového systému:



## Úrovně abstrakce

- Fyzická úroveň: popisuje, jak je záznam (např. *customer*) uložen
- Logická úroveň: popisuje data uložená v databázi a vztahy mezi nimi.

**type** *customer* = **record**

*name*: string;

*street*: string;

*city*: integer;

**end;**

- Úroveň „pohledů“: aplikační programy skrývají detaily o typech dat. *Pohledy* mohou také skrývat informace (jako např. *plat*) – např. z bezpečnostních důvodů.

## Instance a schémata

- Analogie s *typy* a *proměnnými* v programovacích jazycích
- *Schéma* – logická struktura databáze (např. množina *zákazníků* a *úctů* + vztah mezi nimi)
- *Instance* – aktuální obsah databáze v konkrétním časovém okamžiku

## Nezávislost dat

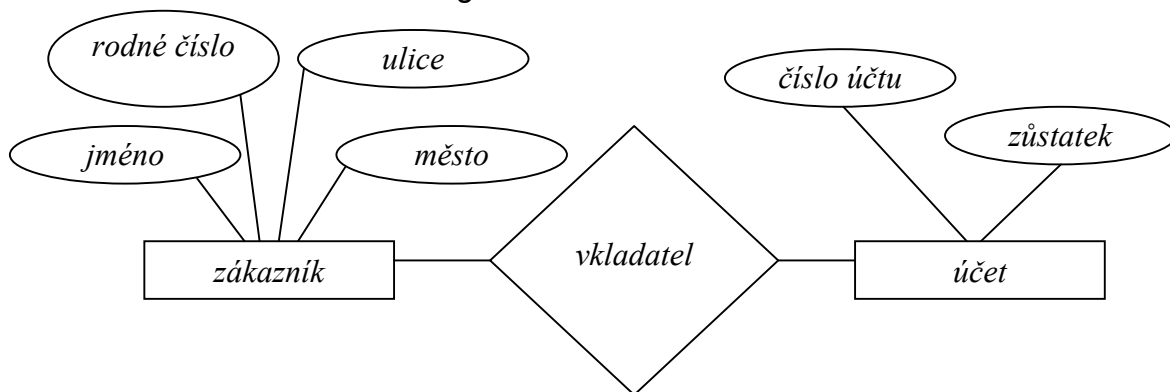
- Možnost měnit *definici schématu* na jedné úrovni bez vlivu na *definici schématu* na jiné vyšší úrovni.
- *Rozhraní* mezi různými úrovněmi a komponentami by měla být dobře definována, aby změny v některých částech neměly významný vliv na jiné části.
- Dva stupně nezávislosti dat:
  - Fyzická nezávislost dat
  - Logická nezávislost dat

## Modely dat

- Sada nástrojů pro popis:
  - dat
  - vztahů mezi daty
  - sémantiky dat
  - omezení dat (*data constraints*)
- Logické modely založené na objektech
  - *entitně-vztahový model* (entity-relationship model)
  - *objektově orientovaný model*
  - sémantický model
  - funkcionální model
- Logické modely založené na záznamech
  - *relační model* (např. MySQL, Oracle)
  - síťový model
  - hierarchický model (např. IMS)

### Entitně-vztahový model (Entity-relationship model)

Příklad entitně-vztahového diagramu



### Relační model

Příklad dat v tabulce v relačním modelu:

<i>jméno zákazníka</i>	<i>rodné číslo</i>	<i>ulice</i>	<i>město</i>	<i>číslo účtu</i>
Johnson	800327/6655	Alma	Palo Alto	A-101
Smith	735203/5312	North	Rye	A-215
Johnson	800327/6655	Alma	Palo Alto	A-201
Jones	605229/2929	Main	Harrison	A-217
Smith	735203/5312	North	Rye	A-201

<i>číslo účtu</i>	<i>zůstatek</i>
A-101	500
A-201	900
A-215	700
A-217	750

## Jazyk pro definici dat (Data definition language; DDL)

- Značení pro specifikaci definice schématu databáze
- DDL kompilátor generuje množinu tabulek uložených v *datovém slovníku* (*Data dictionary*)
- Datový slovník obsahuje *metadata* (data o datech)
- *Jazyk pro ukládání a definice dat* (*Data storage and definition language*) – speciální typ DDL, pomocí kterého se specifikuje struktura *uložení dat* a metody použité databázovým systémem k *přístupu k datům*

## Jazyk manipulace s daty (Data manipulation language; DML)

- Jazyk pro zpřístupnění a manipulaci s daty organizovanými příslušným datovým modelem
- Dvě třídy jazyků
  - Procedurální – uživatel specifikuje, která data jsou vyžadována a jak je získat
  - Neprocedurální (deklarativní) – uživatel specifikuje, která data jsou vyžadována, ale ne, jak je získat

## Správa transakcí (Transaction Management)

- *Transakce* je sada operací, které představují jednu logickou funkci v databázové aplikaci
- *Správa transakcí* zajišťuje, že databáze zůstává v konzistentním stavu nezávisle na selhání systému (např. výpadky energie a pády OS) nebo selhání transakcí.
- *Správa souběžnosti* (*Concurrency-control manager*) dohlíží na vzájemné ovlivňování mezi současně probíhajícími transakcemi, aby byla zajištěna konzistence databáze.

## Správa ukládání dat (Storage Management)

- *Správce ukládání* je programový modul, který poskytuje rozhraní mezi daty uloženými v databázi na nízké úrovni a aplikacemi a dotazy posílanými systémem.
- Správce ukládání je zodpovědný za následující úlohy:
  - interakce se správcem souborů
  - efektivní ukládání, získávání a aktualizace dat

## Správce databáze (Database Administrator)

- Koordinuje všechny aktivity v databázovém systému; má dobré znalosti o informačních zdrojích a potřebách podniku.
- Povinnosti správce databáze zahrnují:
  - Definice *schématu*
  - Definice *struktury ukládání a metody přístupu*
  - *Modifikace* schématu a fyzické organizace
  - Přiděluje uživatelům *práva přístupu* k databázi
  - Specifikuje *integritní omezení*
  - Styčný bod s uživateli
  - Monitoruje *výkon* a zodpovídá za změny v požadavcích

## Uživatelé databáze (Database Users)

- Uživatelé jsou rozděleni podle toho, jakým způsobem spolupracují se systémem
- *Aplikační programátoři* – pracují se systémem pomocí volání DML
- *Chytří uživatelé* – formulují dotazy v databázovém dotazovacím jazyku
- *Speciální uživatelé* – píšou speciální databázové aplikace, které nespádají do klasického zpracování dat
- *Naivní uživatelé* – spouští jeden z předem napsaných aplikačních programů

## Celková struktura systému

