



MASARYKOVA UNIVERZITA

Systemy pro ukládání dat

Jaroslav Šmarda

Systemy pro ukládání dat

- Souborová organizace dat
- Databázový systém
- Relační datový model
- Konceptuální schéma (E-R diagram)
- Tabulky v relační databázi
- Jazyk SQL

Ukládání dat

- Zpracování aplikace
 - Při zpracování data v operační paměti
 - Po skončení data zmizí
- Potřeba trvalého uložení dat
 - datová persistence
- Pro trvalé uložení dat slouží vnější paměť
- Vnější paměť:
 - Harddisky
 - elektromechanické zařízení
 - Flash, SSD (solid-state drive)
 - mikročip
 - CD, DVD

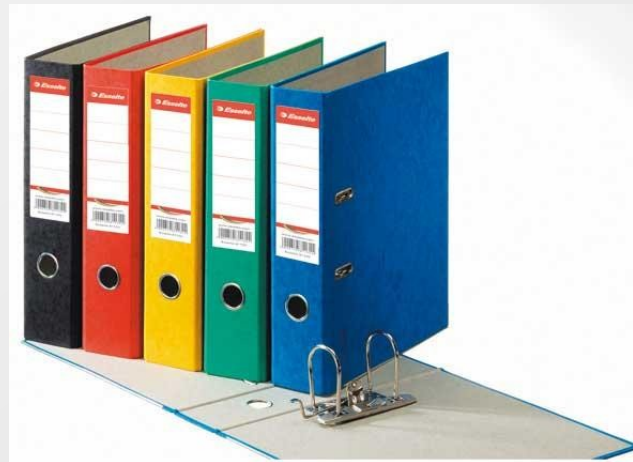
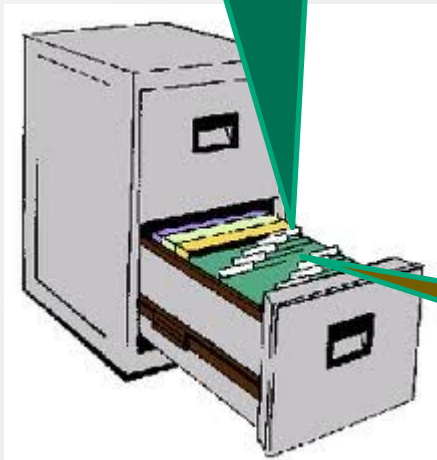
Souborová organizace dat

Soubor

anglicky file

původně pořadač

Sekvenční organizace dat



Posloupnost datových záznamů



Soubory dat

☞ Soubor

- ☞ je kolekce (collection) záznamů (record), které jsou uloženy jako jeden celek (jednotka) operačního systému
- ☞ O soubory se stará operační systém počítače
- ☞ Soubory jsou uloženy ve složkách (adresářích) operačního systému

Záznamy v souboru

- Záznam (record)
 - popisuje entitu (osoba, věc, místo)
 - entita má vlastnosti – atributy (číslo, jméno, příjmení,..)
- Záznam je:
 - posloupnost polí (field) – datových položek
 - každé pole obsahuje jeden atribut
 - Zpravidla záznamy pevné délky s pevným počtem polí v definovaném pořadí

Soubory

☞ Binární

- ☞ určeny pro zpracování aplikacemi, ne pro čtení uživatelem
- ☞ zpravidla záznamy pevné délky
- ☞ datové položky (především numerické) uloženy binárně (integer, long, float, double,...)

☞ Textové

- ☞ dají se číst v textovém editoru
- ☞ textové řádky proměnné délky zakončené speciálním znakem (nový řádek)

Binární soubory

Free Hex Editor Neo (Administrator)

File Edit View Select Operations Bookmarks NTFS Streams Tools History Window Help

Selection

Information	
Total Size	No selection bytes
Fragments	No selection

Details

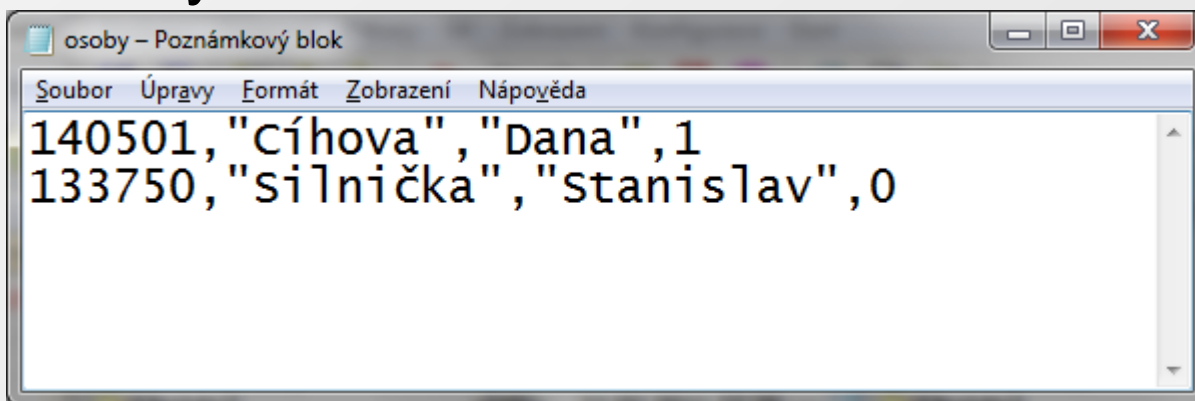
Ready Offset: 0x00000086 (134) Size: 0x00000050 (80): 0,08 KB Hex bytes, 16, Default ANSI OVR

Textové soubory

☞ Příklady:

☞ XML

☞ CSV (Comma Separated Values) – čárkou oddělované hodnoty



The screenshot shows a Notepad window titled "osoby - Poznámkový blok". The menu bar includes "Soubor", "Úpravy", "Formát", "Zobrazení", and "Nápověda". The text area contains two lines of CSV data:

```
140501,"Cíhova","Dana",1  
133750,"silnička","Stanislav",0
```

Soubory

☞ Nevýhody:

- ☞ Každá aplikace musí znát vnitřní strukturu souborů
- ☞ Není řešen paralelní přístup více aplikací současně ke stejnému souboru
- ☞ Pokud každý typ záznamu v jednom souboru, pak v případě hodně typů otevírání mnoha souborů – složité, neefektivní
- ☞ Pokud všechny typy záznamů v jednom souboru, pak složité, nepřehledné, náchylné k chybám

☞ Výhoda:

- ☞ Jednoduché, rychlé

Databáze

- Databáze je:
 - kolekce (collection) datových položek (data items), které jsou spravovány (managed) jako jedna jednotka
- Databázový objekt
 - pojmenovaná datová struktura uložená v databázi
- Databázový model
 - způsob, jakým databázový systém organizuje data, aby odrážela realitu
- DBMS (DataBase Management System)
 - Software od dodavatele databázového systému, který komplexně řeší práci s daty

Databázové systémy

- Oracle,
- Microsoft Access, Microsoft SQL Server,
- MySQL,
- IBM DB2,
- Sybase

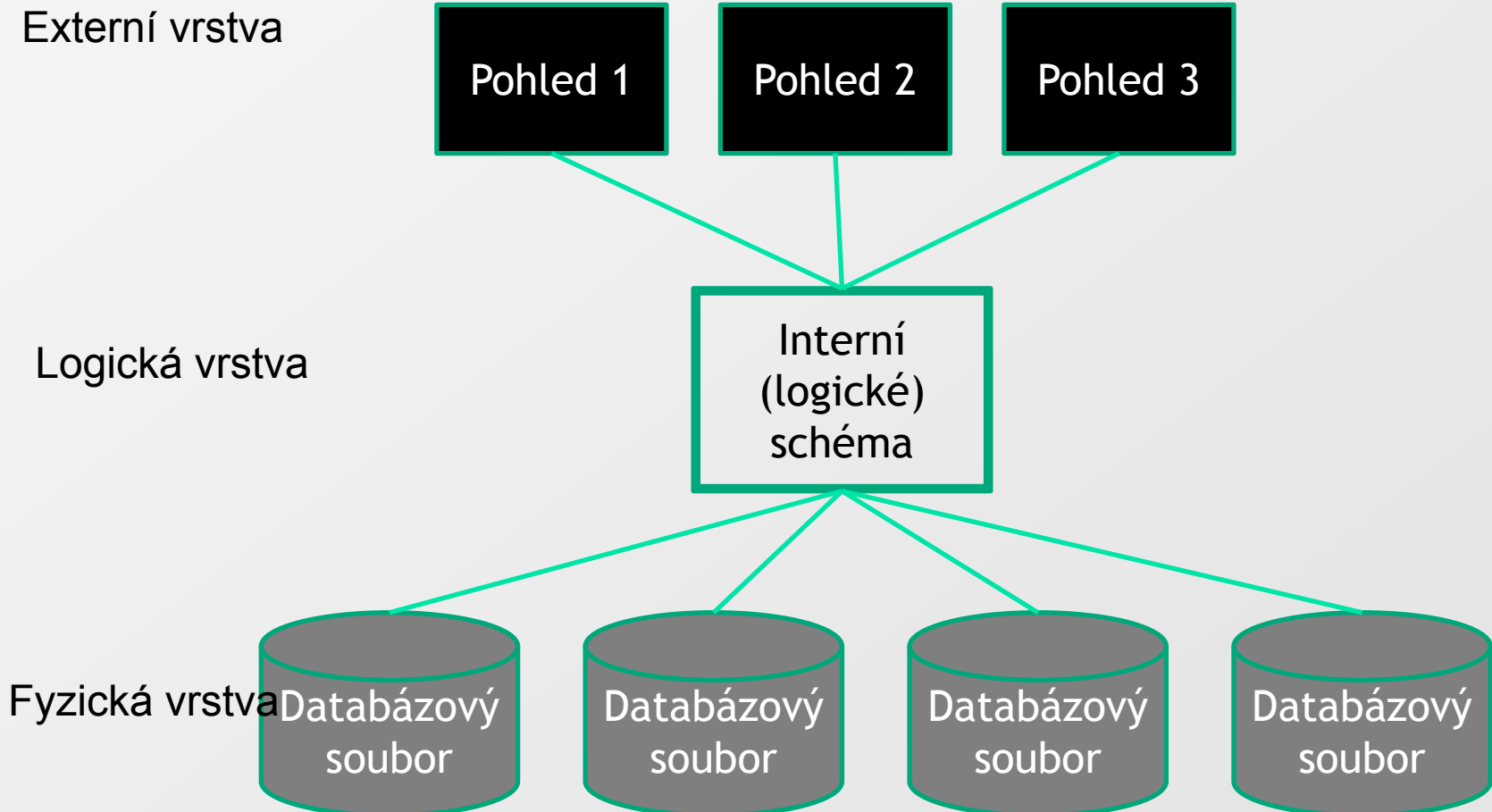
DBMS (DataBase Management System)

- Poskytuje funkce pro práci s daty:
 - Přesun dat z/do fyzických souborů na disku
 - Řízení paralelního přístupu (více uživatelů současně) k datům a řešení konfliktů
 - Správa transakcí - provede všechny operace (např. bankovní) nebo žádnou
 - Podpora pro dotazovací jazyk
 - Bezpečnostní autentizační mechanismus, který zabrání neoprávněným přístupům

Vrstvy datové abstrakce

- Data uložena jen jednou,
- ale různým uživatelům (osoba nebo aplikace) různé pohledy (user views) na data

Vrstvy datové abstrakce



Fyzická vrstva

- ☞ databáze uložena ve více souborech
- ☞ jen v Microsoft Accessu celá databáze v jednom souboru
- ☞ uživatel nepotřebuje znát, jak je databáze fyzicky uložena
- ☞ u velkých instalací je specialista – databázový administrátor (DBA)

Logická vrstva (logické schéma)

- Abstraktní datové struktury nad fyzickými soubory
- Schéma:
 - Kolekce (collection) všech typů datových položek ukládaných v databázi
 - V relačním modelu množina dvoudimenzionálních tabulek

Externí vrstva

- Množina uživatelských pohledů – subschéματα
- Uživatelé a aplikace přistupují k databázi prostřednictvím dotazů - View

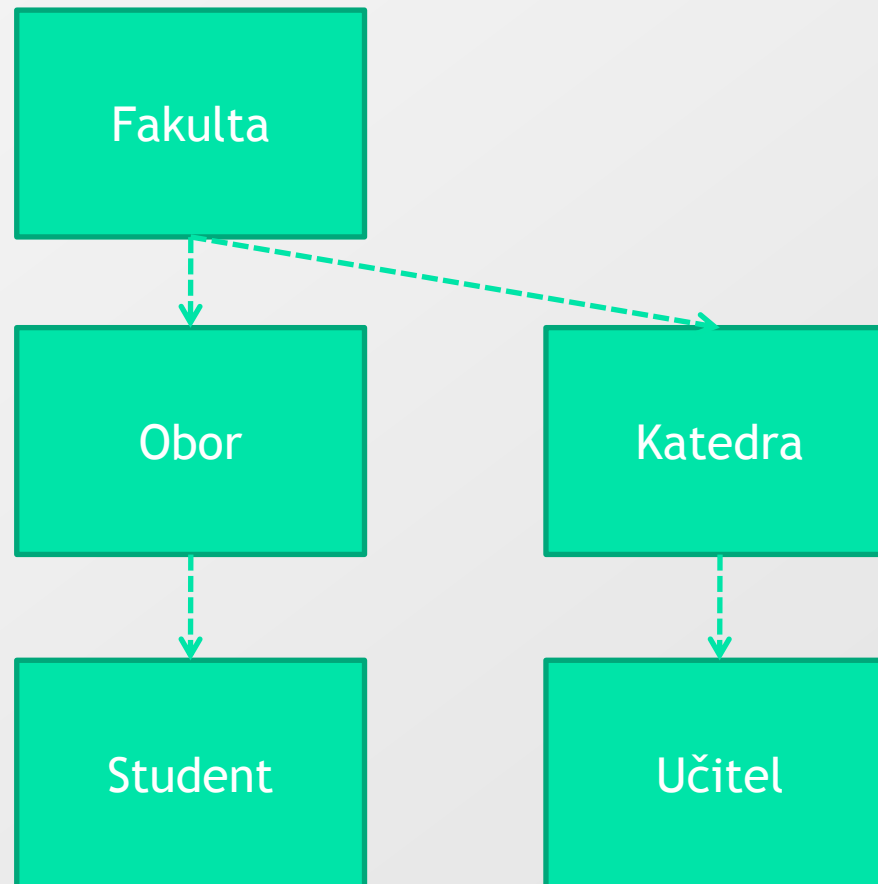
Fyzická datová nezávislost

- Lze změnit uložení v souborech a uživatelé a aplikace to nepoznají
- Fyzická datová nezávislost:
 - Lze přesunout databázový soubor na jiný disk nebo do jiného adresáře
 - Lze rozdělit nebo spojit databázové soubory
 - Lze přejmenovat databázové soubory

Databázové modely

- Hierarchický
 - Nejstarší
 - Nejznámější IMS
 - Pojmy:
 - záznamy (record) - nebo uzly,
 - ukazatele,
 - relace 1:N

Hierarchický databázový model



Databázové modely

☞ Síťový

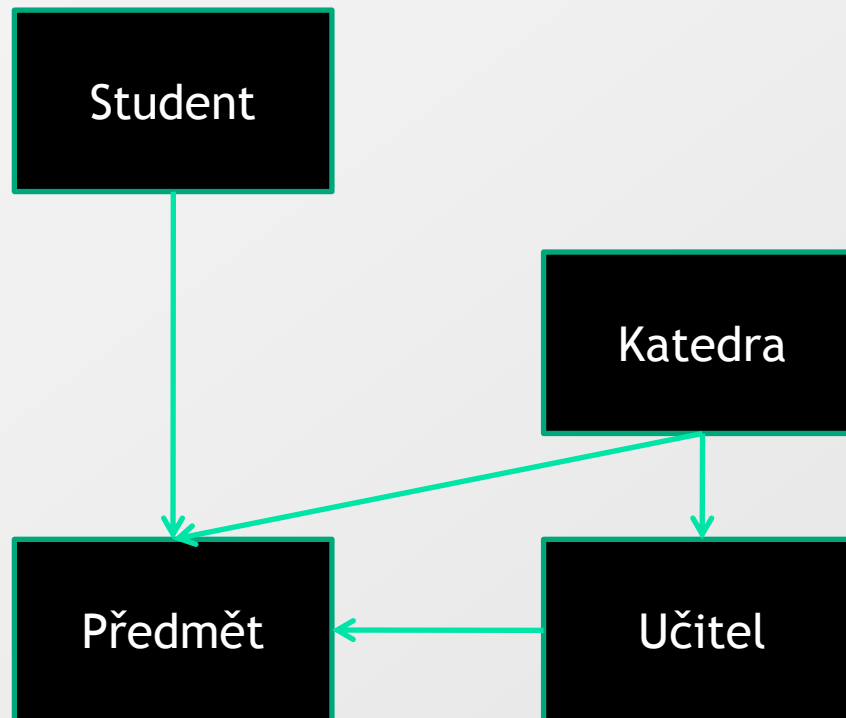
☞ Nejznámější IDMS (Integrated Database Management System)

☞ Pojmy:

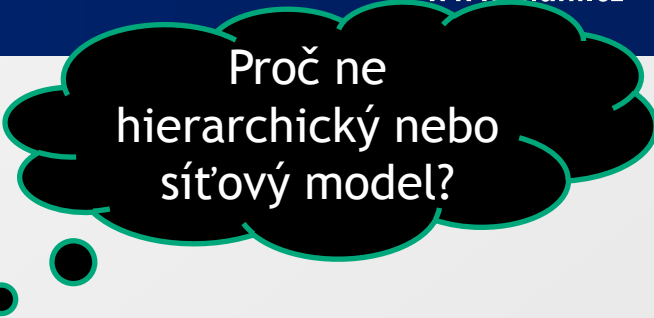
☞ záznamy (record),

☞ relace vlastník–člen (owner-member) 1:N nebo množina (set)

Sít'ový databázový model



Databázový model

A black thought bubble with a green outline, containing the text 'Proč ne hierarchický nebo síťový model?'. It is connected to the main text by three small black circles of increasing size.

Proč ne
hierarchický nebo
síťový model?

- Relaçní databázový model:
 - Hierarchický a síťový – všechny možné vztahy musí být předdefinovány (data jail)
 - Vyhledávání přes ukazatele pomalé
 - Pokud se hierarchická / síťová databáze poruší, těžko se opravuje

Relační datový model - příklad

Studenti

Student_ID	Prijm	Jmeno	Od_kdy
618920	Bendová	Dana	01.10.2008

Predmety

Predmet_ID	Nazev	Ucitel_ID
VIKBA18	Informační systémy	140572

Znamky

Predmet_ID	Student_ID	Znamka
VIKBA18	341652	A
VIKBA18	618920	B

Relační databázový model

- Autor relačního modelu: Dr. E. F. Codd, 1970
- Data ve 2-D (dvou-dimenzionálních) tabulkách
- Místo předdefinovaných cest v síťovém modelu libovolné spojování tabulek (join) do pohledů (view)
- K propojení tabulek se používají běžné údaje (například osobní číslo zaměstnance, číslo faktury atd.), nikoliv ukazatele

Relační datový model - výhody

- Definice a práce s tabulkami je jednoduchá
- Data lze získat prostřednictvím jednoduchých dotazů
- Data lze dobře zabezpečit
 - např. oprávnění k tabulkám a sloupcům tabulek
- ANSI a ISO standardizace modelu
- Hodně dodavatelů databázových systémů
- Konverze mezi databázovými systémy je poměrně jednoduchá
- Relační databázové systémy jsou stabilní systémy

Návrh databáze

- Konceptuální
 - Technologicky nezávislý
 - Použitelný pro libovolný datový model i pro systém souborů
- Logický
 - Překlad konceptuálního schématu do prvků relačního datového modelu (tabulek)
- Fyzický
 - Mapování prvků relačního datového modelu (tabulek) do fyzických souborů

Konceptuální databázový návrh

- Proces, který vytváří **technologicky nezávislý** datový model,
 - který může být implementován v **libovolném** databázovém systému a dokonce i v souborovém systému
- Výsledkem je E-R diagram

Konceptuální schéma obsahuje

⇒ Entity

- ⇒ Osoby, věci, místa, události,...
- ⇒ „věci“ z reálného světa, jejichž záznamy potřebujeme ukládat v databázi
- ⇒ Ve schématu zobrazeny jako obdélníky
- ⇒ Každá entita představuje **třídu** (množinu) entit
- ⇒ Jeden výskyt entity je **instance** entity

⇒ Atributy (entit)

- ⇒ Jeden fakt, který určitým způsobem charakterizuje entitu
- ⇒ Jednoznačný identifikátor (unique identifier):
 - ⇒ Atribut (nebo skupina atributů) jednoznačně charakterizující entitu

Konceptuální schéma obsahuje

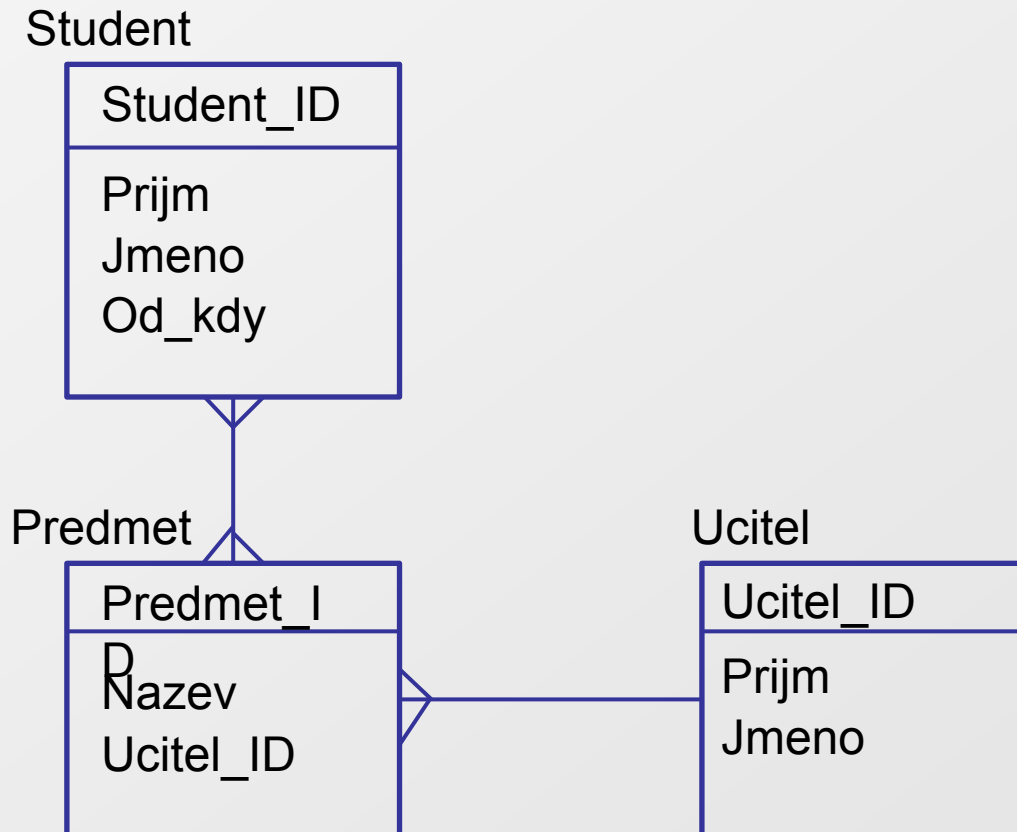
☞ Relace

- ☞ asociace mezi entitami
- ☞ relace je podstatou datového modelu - databáze je ukládání provázaných dat
- ☞ „lepidlo“, které drží relační model
- ☞ v grafickém schématu čára

Mohutnost (cardinality) relace

- Maximální mohutnost (cardinality) relace:
 - maximální počet instancí jedné entity, které mohou být asociovány s entitou na opačné straně čáry
 - Možnosti:
 - Jedna (One)
 - Mnoho (Many)
- Minimální mohutnost relace:
 - minimální počet instancí jedné entity, které mohou být asociovány s entitou na opačné straně čáry
 - Možnosti:
 - Nula (Zero)
 - Jedna (One)

Příklad E-R diagramu



Základní typy relací

- 1:1 (One : One)
 - Není moc častá
- 1:N (One : Many)
 - v grafu na straně N (Many) tzv. vraní noha (crow's foot)
- M:N (Many : Many)
 - V grafu na stranách M i N tzv. vraní noha (crow's foot)

Logický návrh databáze

➤ Mapování:

Konceptuální
schéma
(E-R diagram)

Relační
datový model
(kolekce 2-D
tabulek)

Tabulky v relačním modelu

- 2-D struktury, skládají se z řádků a sloupců

Student_ID	Prijm	Jmeno	Od_kdy
341652	Cíhová	Marta	01.10.2009
618920	Bendová	Dana	01.10.2008

- Řádek – jedna entita
- Sloupec – jeden atribut
- Normalizace tabulek:
 - více tabulek s méně sloupci tak, aby nevznikaly duplicity sloupců

Sloupce a datové typy v relační databázi

- ☒ Sloupec (nebo pole)
 - ☐ nejmenší pojmenovaná datová jednotka, se kterou se pracuje v relační databázi
 - ☐ Má přiřazen datový typ
 - ☐ Např. NUMERIC, DECIMAL, INTEGER, VARCHAR, CHAR, FLOAT, DOUBLE, BOOLEAN, DATE, TIME, TIMESTAMP, ...
- ☐ Datový typ
 - ☐ Vymezuje obor hodnot ve sloupci
 - ☐ Určuje operace, které je možno provést
- ☐ Primární klíč (Primary Key)
 - ☐ Sloupec nebo sloupce, které jednoznačně identifikují entitu

SQL (Structured Query Language)

- SQL je programovací jazyk navržený pro práci s relačními databázemi
- Historie:
 - Předchůdce SEQUEL (Structured English Query Language), IBM konec 70 tých let
 - Standardizace SQL: ANSI 1986, ISO 1987

Příkazy SQL

- Dotazovací jazyk (DQL – Data Query Language)
 - SELECT
- Manipulační jazyk (DML – D. Manipulation L.)
 - INSERT, UPDATE, DELETE
- Definiční jazyk (DDL – D. Definition L.)
 - CREATE
- Řídící jazyk (DCL – D. Control L.)
 - GRANT (povolit), REVOKE (odvolat)

Dotazovací jazyk – příkaz SELECT

❏ SELECT

- ❏ sloupce, které mají být ve výsledné tabulce
- ❏ znak * pro všechny sloupce tabulky

❏ FROM

- ❏ tabulky nebo pohledy (view), ze kterých se vybírají data

❏ WHERE

- ❏ podmínky pro výběr řádků výsledku
- ❏ operátory = <= > >= <> AND OR a závorky ()

❏ ORDER BY

- ❏ pořadí řádků v tabulce (sloupce, podle kterých budou řádky seřazeny)

❏ GROUP BY

- ❏ sloupce, podle kterých budou řádky ve výsledku agregovány

Příklad příkazu SELECT

Student_ID	Prijm	Jmeno	Od_kdy
341652	Cíhová	Marta	01.10.2009
618920	Bendová	Dana	01.10.2008

```
SELECT Jmeno, Prijm  
FROM Studenti  
WHERE DATEPART(YEAR, Od_kdy)<2009;
```

Jmeno	Prijm
Dana	Bendová

Spojení (JOIN) tabulek

Student_ID	Prijm	Jmeno	Od_kdy
341652	Cíhová	Marta	01.10.2009
618920	Bendová	Dana	01.10.2008
257694	Peterka	Jiří	01.10.2009

Predmet_ID	Student_ID	Znamka
VIKBA18	341652	A
VIKBA18	618920	B

```
SELECT Prijm, Predmet_ID, Znamka  
FROM Studenti JOIN Znamky  
ON Studenti.Student_ID =Znamky.Student_ID  
ORDER BY Prijm;
```

Prijm	Predmet_ID	Znamka
Bendová	VIKBA18	B
Cíhová	VIKBA18	A

Standardní (vnitřní – INNER) a vnější (OUTER) spojení (JOIN)

- Vnitřní spojení (INNER JOIN)
 - Výsledek obsahuje jen ty řádky, které jsou v obou spojovaných tabulkách
- Vnější spojení (OUTER JOIN)
 - Výsledek obsahuje i řádky, které jsou jen v jedné z tabulek, sloupce z druhé tabulky mají hodnotu NULL (prázdný)

Příklad OUTER JOIN

Student_ID	Prijm	Jmeno	Od_kdy
341652	Cíhová	Marta	01.10.2009
618920	Bendová	Dana	01.10.2008
257694	Peterka	Jiří	01.10.2009

Predmet_ID	Student_ID	Znamka
VIKBA18	341652	A
VIKBA18	618920	B

```
SELECT Prijm, Predmet_ID, Znamka
FROM Studenti OUTER JOIN Znamky
ON Studenti.Student_ID =Znamky.Student_ID
ORDER BY Prijm;
```

Prijm	Predmet_ID	Znamka
Bendová	VIKBA18	B
Cíhová	VIKBA18	A
Peterka