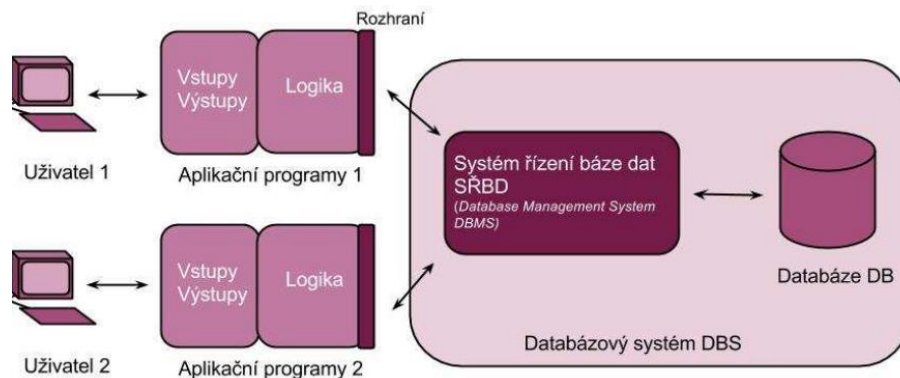


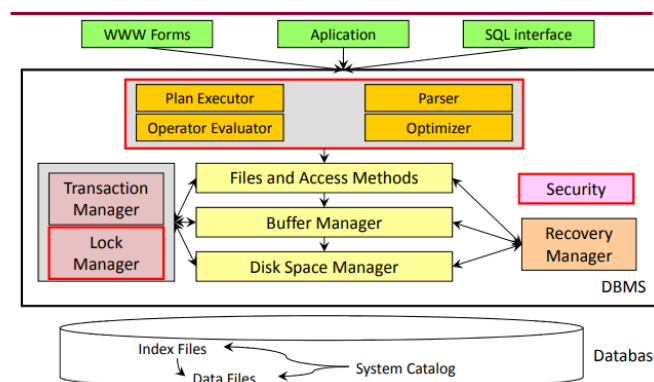
27. Architektura relačních databázových systémů, datový model, konceptuální modelování

- **Databázový systém umožňuje transformovat data na informace.**
- Proč využívat databáze a ne souborový systém:
 - Izolace dat
 - Duplicita a redundance dat
 - Nebezpečí nekonzistence stejných dat v různých souborech
 - Závislost mezi aplikací a daty – těžko se provádí změny
- Co jsou databázové systémy:
 - Centralizované úložiště dat (řeší izolaci a získávání informací, definice dat není již součástí aplikace)
 - Vytvoření mezivrstvy (není důležité kde jsou data uložena)
 - Kontrola přístupu a manipulace s daty i mimo aplikace
 - jednotné zabezpečení integrity dat (**integritní omezení**)
 - kontrolovaný přístup více uživatelů (**transakce**)
 - přístup k libovolné podmnožině dat (**přístupová práva, pohledy**)
- Proč relační databáze:
 - existují kvalitní implementace
 - jednoduchost modelu
 - dotazovací jazyky vyšší úrovně

Architektura relačních databázových systémů



Architektura DBS [3]



- **Databázový systém DBS = DB + SŘBD**

Databáze DB:

- sdílená kolekce dat popisující aktivity jedné nebo více organizací,
- definice dat tzv. **systémový katalog** je uložený spolu s vlastními daty

Systém řízení báze dat SŘBD:

- softwarový systém, který umožňuje uživatelům definovat, vytvářet a udržovat databázi a poskytuje k ní kontrolovaný přístup

DBS:

- **poskytuje efektivní, spolehlivé, praktické a bezpečné úložiště a přístup k obrovskému množství perzistentních dat pro více uživatelů**

Datový model

Návrh datové vrstvy:

- sada konceptů **na popis dat, vztahů mezi daty a omezeními kladenými na data** v dané organizaci.
- určen ke komunikaci mezi zákazníkem a návrhářem databáze
- Obsahuje:
 - popis struktury
 - integritní omezení IO – explicitně vyjádřené podmínky, které má databáze splňovat
- Přístupy
 - **bottom-up** – vhodné pro menší případně existující databáze (od detailů po celek)
 - **top-down** – vhodné pro větší a nově vytvářené databáze (první od hlavních věcí pak detaily)

Rozdíl mezi datovým a konceptuálním modelem relačních databází spočívá v jejich úrovni abstrakce a zaměření. Datový model se více zaměřuje na konkrétní strukturu a technické aspekty dat, zatímco konceptuální model se zaměřuje na obecné popisy dat a jejich vztahů bez ohledu na implementaci. Konceptuální model slouží jako základ pro návrh fyzického datového modelu, který specifikuje konkrétní implementační detaily.

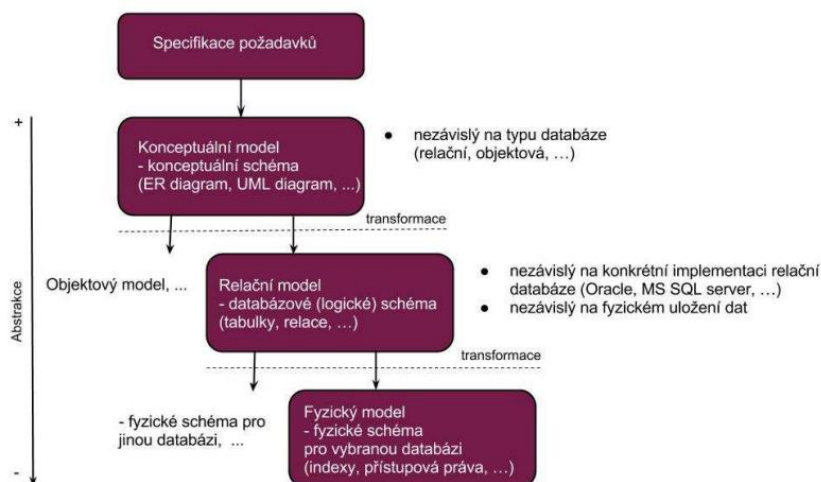
Konceptuální modelování

Charakteristiky konceptuálního modelu

Konceptuální model:

- **zachycuje požadavky** klienta
- zachycuje doménu problému ve formě entit (objektů) a vztahů mezi nimi
- je úplně nezávislý na implementačních detailech (použitý SŘDB, hardwarová platforma, programovací jazyků a databázovém modelu (relační, hierarchický, síťový, objektový)
- forma: je vhodné využít existující konvence, diagram

Schéma konceptuálního modelu:



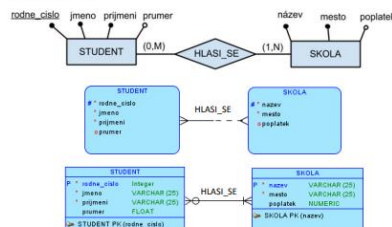
Pojmy:

- **Model** je soubor pojmů, nástrojů a technik pro modelování
- Výsledkem modelování je **schéma**
- Schéma se často zobrazuje formou **diagramu**

Základní konceptuální modely

1. Entitně-relační model využívající ER(A) diagram – Chen (1976)

- není standardizován, existuje mnoho notací
 - Chenova (původní)
 - Barkrova (Oracle)
 - Information Engineering



2. UML model využívající UML diagram tříd (class diagram)

- standardizován



Konstrukty konceptuálního modelu

- Entita
 - Atribut
 - identifikační klíč
- Vztah
- Integritní omezení
 - entitní
 - doménové
 - null
 - jedinečnost
 - vztahové
 - kardinalita
 - členství
 - slabá/silná entita
 - enterprise

Popis konstruktů:

Entita

- má nezávislou existenci
- je to abstrakce, název pro množinu podobných věc
- je daná **jménem** a **množinou vlastností** (atributů)
- **Instance entity** je konkrétní výskyt (prvek) entit (je jednoznačně odlišitelná)
- **Entitní množina** je množina všech potřebných prvků entity (př. STUDENT* je množina např. všech studentů FM)

Atribut

- je vlastnost entity, která nás zajímá a jejíž hodnotu chceme mít v DB uloženo
- má **datový typ**
- může být **proměnlivý** (věk), nebo **stálý** (datum narození) je vhodné volit spíš atributy stál
- Druhy atributů:
 - atomický (jednoduchý)
 - složený (např. adresa = psč, město, ulice)
 - jednohodnotový
 - vícehodnotový (např. telefonní číslo)
 - odvozený (věk, celkový počet pracovníků, kauce = 3x měsíční renta)

Vztah

- je vazba, asociace mezi entitami
- stupeň vztahu = počet entit ve vztahu
- binární, ternární, vazba sama na sebe (rekurzivní)
- atribut vztahu
- Instance vztahu - př. Alice se hlásí na TUL

Integritní omezení IO – entitní

- **Doména atributu** – množina přípustných hodnot
 - rozsahy, kusovníky
 - datový typ a délka atributu
 - defaultní hodnota
- Každý atribut může být **povinný** tj. musí mít zadanou hodnotu, nebo **nepovinný** (NULL resp. absence hodnoty) – značka NN
- **Jedinečnost** (unique) – značka U
- **Identifikační klíč**
 - je atribut nebo kombinace atributů, jejichž hodnoty jednoznačně odliší jednu instanci dané entity od druhé
- **Primární klíč**
 - nejvhodnější identifikační klíč
- **Složený klíč**
 - identifikační klíč složený z více atributů

Integritní omezení IO – vztahové

- **Kardinalita** – popisuje počet možných relací pro každou participující entitu
 - 1:1
 - 1:M
 - M:N
- **Členství ve vztahu** (parcialita, existenční závislost) – popisu jestli se všechny entity musí nacházet ve vztahu
 - **totální** (povinný) **musí**
 - **parciální** (nepovinný) **může**

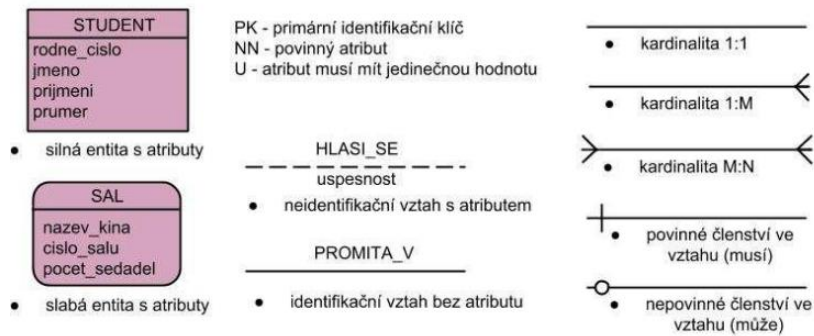
Parcialita – může/musí

- **Silná/slabá entita**
 - **Silná entita** – je identifikačně a existenčně nezávislá
 - **Slabá entita** – je identifikačně a existenčně závislá na jiné entitě.

Pokud máme slabou entitu, nelze její instance rozlišit pouze podle vlastních atributů. Identifikujeme jí tedy pomocí toho, že je v povinném tzv. **identifikačním** vztahu k jiné rodičovské entitě. Její primární klíč bude složený a bude obsahovat i primární klíč rodičovské entity.

příkl. kópie-kniha, sál-kino

- založená na notaci v Case Studiu



- Čtení relací z obou stran

Každý
název první entity
může/musí (parcialita na straně druhé entity)
název relace
1/více (kardinalita na straně druhé entity)
název druhé entity

Konceptuální návrh databáze

1. Identifikovat entity
2. Identifikovat vztahy (využití matice každý s každým)
3. Určit kardinalitu, členství a identifikační závislosti vztahů
4. Identifikovat atributy a přiřadit je k entitám a vztahům
5. Určit domény atributů
6. Identifikovat identifikační klíče
7. Identifikovat hierarchii v rozšířeném modelu ISA ER
8. Nakreslit ER diagram
9. Zkonzultovat navržený diagram se zákazníkem