

Konceptuální modelování databází - proč?

- Nejdůležitější přínosy:
 - ▶ společné chápání objektů aplikace uživateli a projektanty,
 - ▶ integrace různých uživatelských pohledů,
 - ▶ výsledek je vstupem pro realizaci databáze,
 - ▶ slouží jako dokumentace.
- Důsledky vypuštění konceptuální úrovně :
 - ▶ Příliš nízká úroveň pohledu na data:
 - ⇒ obtížná komunikace se zadavatelem (zákazníkem),
 - ⇒ neumožní realizaci větší databáze.
 - ▶ V rozsáhlejší databázi je velmi těžké se zorientovat.

Návrhy IS “postaru”

Funkční a datová analýza relativně oddělená.

- **Funkční schéma** – výsledek funkční analýzy a návrhu.
 - ▶ Kdo bude používat aplikaci? – Kategorie uživatelů systému.
 - ▶ Pracovní postupy v organizaci, které mají být počítačově podporovány.
 - ▶ Události, která spouští počítačově podporovaný pracovní postup.
- **Datové schéma** – výsledek datové analýzy a návrhu.

Problém: udržení konzistence funkční a datové analýzy.

- Funkční a datová analýza není tak striktně oddělená.
- Objektový přístup zdůrazňuje zapouzdření dat, pracuje s voláním metod na objektech. Masivně využívá konstrukce jako agregace, kompozice, dědění.
- UML notace je objektově orientovaná.

Varování

V této přednášce nebudeme diskutovat všechny konstrukty diagramu tříd.

Omezíme se na ty konstrukty, které lze víceméně přímo realizovat v **relačním** databázovém stroji pomocí **deklarativních integritních omezení**.

Prvky konceptuálního modelu

- orientace na třídy (entity) a vztahy mezi nimi
- entity (třídy)
 - atributy a další IO
 - instance (objekty)
- asociace (vztahy)
 - ▶ obecný vztah (asociace)
 - kardinalita, parcialita, role
 - ▶ ISA-hierarchie (podtyp jako specializace)
 - nepřehánět !!! (zejména pro relační model)
 - ▶ agregace, kompozice (entita je rozšířením jiné, je na ní závislá)

Tvorba datového modelu – postup

- Identifikace **entit (entitních typů)** jako tříd objektů stejného typu.

příklady entit

FILM, ZÁKAZNÍK, ZAMĚSTNANEC, KOPIE

- Identifikace **vztahů (vztahových typů)**, do kterých entity mohou vstupovat :

příklady vztahu

ZAKAZNÍK (entita)
MÁ_PŮJČEN (vztah)
FILM (entita)

- Identifikace atributů popisujících blíže vlastnosti entit a vztahů.

Atributy a IO

Příklady atributů:

- **příjmení** (atribut) zaměstnance (entita),
- **rodné číslo** (atribut) zaměstnance (entita),
- **datum** (atribut vztahu), do kdy má zákazník (entita) půjčenou (vztah) kopii filmu (entita).

Příklady integritních omezení (IO):

- Doménou atributu **vaha** (entity zákazník) je integer.
- Atribut **vaha** (entity zákazník) musí mít jednu hodnotu (pro jednoho zákazníka).
- Atribut **rodné číslo** je identifikátorem entity zákazník.
- Atribut **datum** (vztahu půjčeno) může mít nejvýše jednu hodnotu.
- Atribut **herec** entity film může mít mnoho hodnot.

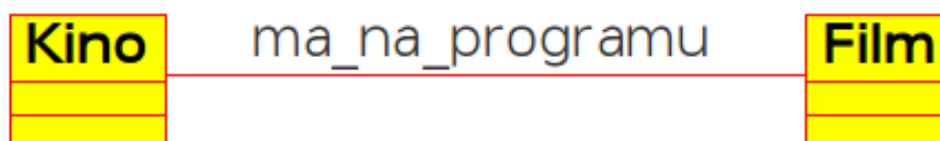
Entita a vztah - notace UML

Lineární zápis:

Entity: Film, Kino

Relace: ma_na_programu (Film, Kino)

Grafický zápis:



Pokračování

Prezentace pokračuje v dalším souboru.