

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# OBJEKTOVĚ - RELAČNÍ DATABÁZE

# Klady relačních databázových systémů

2

- ❑ Relační systémy nabízí možnosti práce s rozsáhlými soubory dat s relativně jednoduchou strukturou (tabulkami) a se silnými dotazovacími prostředky, které reprezentuje standard SQL89 a SQL92
- ❑ Relační technologie dospěla ke svému vrcholu v souběžném zpracování požadavků v různých paralelních architekturách
- ❑ Je rozšířená na většině hardwarových platform,
- ❑ Poskytuje "brány" (gateways) mezi jednotlivými relačními databázovými systémy

# Nedostatky relačních DB systémů

3

- ❑ Relační systémy se ukázaly jako nevhodné pro aplikace, vyžadující mnohem bohatší datové typy než poskytuje relační model – například
  - ❑ modelování objektů v systémech pro návrh (napr. CAD)
  - ❑ geografické IS.
  - ❑ speciální aplikace objevující se v medicíně, (EKG, rentgenové snímky)
  - ❑ ve výzkumu Země (seismická data, snímky ze satelitu).
- ❑ V 90-tých letech do databázové technologie začaly pronikat prvky objektově orientovaných (dále OO) jazyků. Sváděla k tomu představa ukládat objekty do databáze a využít současně mnoha užitečných prvků OO technologie.

# OO datový model

4

- ❑ Objektově orientovaný datový model (ODM) představuje zcela nový přístup a není postaven jako rozšíření relačního datového modelu.
- ❑ Do jisté míry zde jde o renezanci původního síťového datového modelu, který je doplněn o možnost práce s objekty tak, jak je známe z objektového programování.
- ❑ OO DBS jsou na trhu - např. Gemstone, Versant, Caché. Již před 15 lety existovaly názory, že OO DBS zcela vytlačí relační systémy. Dosud se tak nestalo.

# Výhody objektového datového modelu

5

- ❑ Pro OO modelování je charakteristická především bohatost typů objektů, které jsou k dispozici.
- ❑ Ukazuje se, že pomocí těchto typů objektů se snadno modelují a následně snadno implementují objekty používané právě v podnikových systémech. Tyto objekty jsou složité nejen z hlediska struktury, ale i z hlediska vzájemných vztahů.
- ❑ Relační databáze založené na normalizovaných tabulkách umožňují modelovat jednoduše, ovšem za cenu mnohdy složitého a neefektivního přístupu k odpovídajícím datům. Proto mohou OO systémy nabývat pro vývoj budoucích aplikací zásadního významu.

# Požadované vlastnosti OO DBS

6

- ❑ Systém musí být databázový - musí podporovat perzistenci, správu sekundární paměti, paralelizmus, zotavení a prostředky pro kladení „ad hoc dotazů.
- ❑ Systém musí být objektově orientovaný – musí podporovat:
  - ❑ komplexní objekty, nepožaduje se 1NF
  - ❑ uživatelem rozšiřitelné typy
  - ❑ identifikaci objektu,
  - ❑ zapouzdření,
  - ❑ typy nebo třídy,
  - ❑ dědičnost,
  - ❑ polymorfizmus,
  - ❑ pozdní vazbu.

# Proč objektově-relační DBS ?

7

- ❑ O-R technologie, se zdá být vhodným kompromisem mezi relační a objektovou technologií. Cílem je:
  - ❑ obdržet maximum z rozsáhlých investic do relační technologie (vybudování databází, zkušenosti vývojářů i uživatelů)
  - ❑ využít výhody v pružnosti a produktivitě OO modelování,
  - ❑ integrovat databázové služby do nových aplikací.
- ❑ O-R datový model ve svých principech zůstává původním relačním datovým modelem.
- ❑ Jde o doplnění relačního datového modelu o možnost práce s některými datovými strukturami, které známe z oblasti objektově orientovaných programovacích jazyků.

# Nové črty O-R DBS

8

- ❑ Přidávají možnosti ukládat objekty do relační databáze.
- ❑ Výrobci relačních databází potřebovali rozšířit možnosti jejich aplikace do oblastí požadujících integraci klasických tabulkových dat a objektů speciálních typu, jako jsou např. časové řady, prostorová data, či binární objekty mezi které patří audio, video, obrázky, či aplety.
- ❑ Zapouzdřením metod a datových struktur může O-R server vyvolat složité operace pro prohledávání a transformaci těchto složitých multimediálních dat.



# Rozšiřitelnost

9

- ❑ Protože je nemožné, aby každý výrobce DBS byl schopen implementovat širokou škálu různých typů dat a odpovídajících přístupových metod, realizovaly O-R DBS již dávno požadovanou ideu rozšiřitelných relačních DBS.
- ❑ Rozšiřitelnost zde znamená dát možnost přidávání nových datových typů, ale také potřebných programů (funkcí) pro efektivní vyhledávání dat v souladu s jejich vnitřní strukturou.
- ❑ "Objektovost" by měla u O-R technologie zahrnovat odpovídající objektový jazyk vyšší úrovně. Jako nejvhodnější se ukázal standard SQL:2003 (který revidoval SQL-99 a přidal několik nových možností.)

# Rozšíření relačního modelu

10

- ❑ **Strukturované typy atributů** – hodnotou atributu může být celá relace (vnořená relace)
- ❑ **Reference** (odkazy) - umožňují sdílení řádků mezi tabulkami a umožňují uživatelům vyjádřit složité spojení tabulek pomocí mnohem jednoduššího vyjádření cesty.
- ❑ **Metody** – podobně jako u OO jazyků
- ❑ **Identifikátory řádků** (řádek v tabulce představuje objekt) – lze odlišit řádky i v případě, že všechny ostatní atributy (kromě identifikátoru) jsou shodné. Identifikátor řádku je obecně neviditelný pro uživatele, lze jej vidět pouze za určitých okolností.

# Vnořené relace – rekurzivní definice

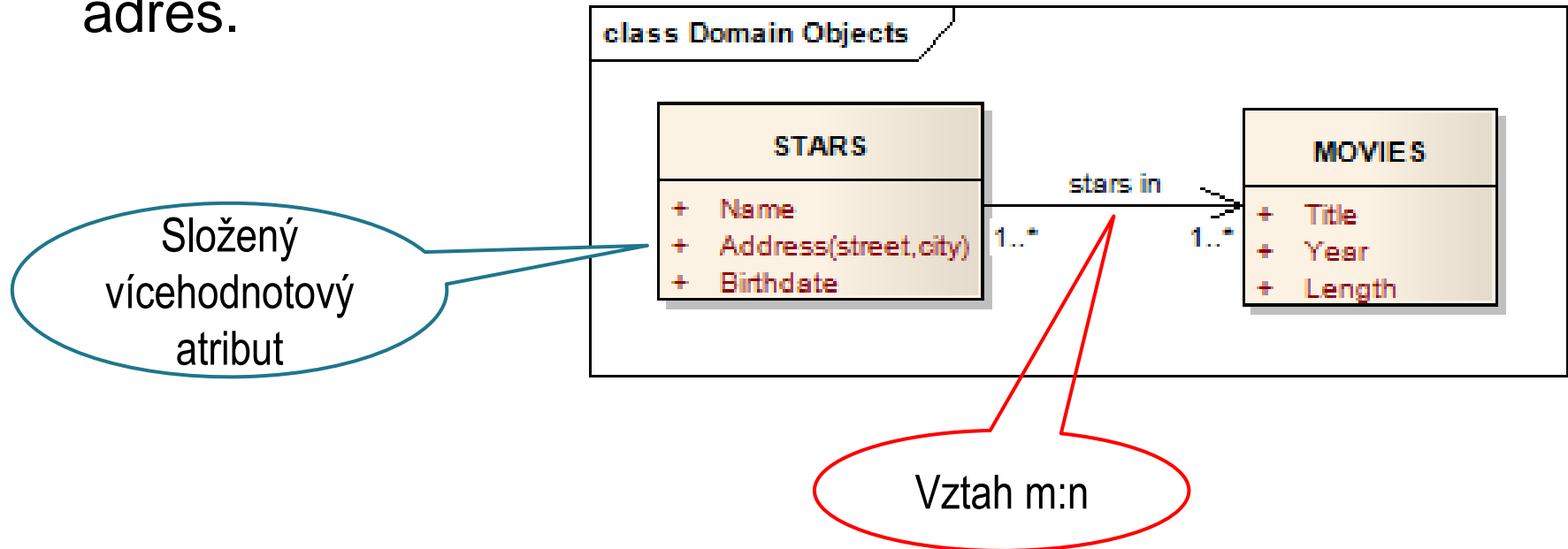
11

- ❑ **Základ:** Atribut může být atomického typu (integer, real, string,...)
- ❑ **Indukce:** Typem relace může být libovolné schéma, sestávající z názvu jednoho nebo více atributů z nichž každý je přípustného typu. **Navíc,**
  - ❑ typem atributu může být schéma.
  - ❑ typem atributu může být odkaz na řádek s daným schématem nebo množina odkazů na řádky s daným schématem.

# Příklad 1

12

- ❑ Máme vytvořit O-R schéma pro uložení informací o hercích (jméno, adresa, datum narození), přičemž každý herec bude mít uvedeny i základní informace o filmech, ve kterých hraje (název filmu, rok, délka). Někteří z herců mají více adres.



# Příklad – řešení A – vnořené relace

13

Name	Address	Birthdate	Movies																		
Fisher	<table><tr><td>street</td><td>city</td></tr><tr><td>Maple</td><td>H'wood</td></tr><tr><td>Locust</td><td>Malibu</td></tr></table>	street	city	Maple	H'wood	Locust	Malibu	9/9/50	<table><tr><td>title</td><td>year</td><td>lenght</td></tr><tr><td>Star Wars</td><td>1977</td><td>124</td></tr><tr><td>Empire</td><td>1980</td><td>127</td></tr><tr><td>Return</td><td>1983</td><td>133</td></tr></table>	title	year	lenght	Star Wars	1977	124	Empire	1980	127	Return	1983	133
street	city																				
Maple	H'wood																				
Locust	Malibu																				
title	year	lenght																			
Star Wars	1977	124																			
Empire	1980	127																			
Return	1983	133																			
Hamill	<table><tr><td>street</td><td>city</td></tr><tr><td>Oak</td><td>NY</td></tr></table>	street	city	Oak	NY	8/8/54	<table><tr><td>title</td><td>year</td><td>lenght</td></tr><tr><td>Star Wars</td><td>1977</td><td>124</td></tr><tr><td>Empire</td><td>1980</td><td>127</td></tr><tr><td>Return</td><td>1983</td><td>133</td></tr></table>	title	year	lenght	Star Wars	1977	124	Empire	1980	127	Return	1983	133		
street	city																				
Oak	NY																				
title	year	lenght																			
Star Wars	1977	124																			
Empire	1980	127																			
Return	1983	133																			

# Příklad – řešení B – vnořené relace a odkazy

14

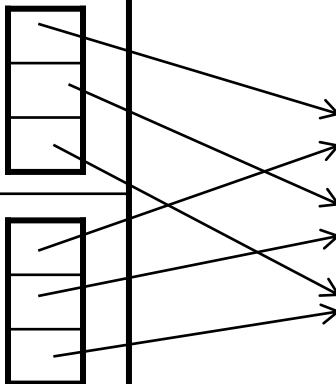
V případě řešení A předchozího příkladu nastala redundance v údajích, kterou O-R model umožňuje řešit pomocí odkazů.

## Stars

Name	Address		Birthdate	Movies
Fisher	street	city	9/9/50	<div><div></div><div></div><div></div></div>
	Maple	H'wood		
	Locust	Malibu		
Hamill	street	city	8/8/54	<div><div></div><div></div><div></div></div>
	Oak	NY		

## Movies

title	year	length
Star Wars	1977	124
Empire	1980	127
Return	1983	133



# Transformace konceptuálního modelu do O-R modelu

15

- ❑ Entitní typ → Strukturovaný typ
- ❑ Atribut → Typ sloupce
  - Vícehodnotový atribut → ARRAY / MULTISSET
  - Složený atribut → ROW / Strukturovaný typ sloupce
- ❑ Vztah
  - 1:1 → REF / REF
  - 1:N → REF / [ARRAY nebo MULTISSET ]
  - M:N → ARRAY / [ARRAY nebo MULTISSET ]
- ❑ Agregace → REF / [ARRAY nebo MULTISSET ]
- ❑ Kompozice → REF / [ARRAY nebo MULTISSET ]
- ❑ Generalizace → Hierarchie typů

*MULTISSET je kolekce prvků, které se mohou opakovat*

# OO rozšíření jazyka SQL

16

- ❑ Základní rozšíření jazyka SQL, které přidává objektové vlastnosti jsou uživatelem definované typy (UDT); v zásadě jsou to definice třídy s atributy a metodami.
- ❑ UDT se používají dvěma různými způsoby:
  - ❑ UDT může být typem tabulky,
  - ❑ UDT může být typem atributu v tabulce.



# Vytvoření UDT

17

❑ CREATE TYPE <TypeName> AS (<attribute declarations>

❑ **Příklady:**

❑ CREATE TYPE Address\_UDT AS (  
    Street CHAR(50), City CHAR(20) )

❑ CREATE TYPE Movie\_UDT AS (  
    Title CHAR(30), Year INTEGER, Genre CHAR(10))

❑ CREATE TYPE Star\_UDT AS (  
    Name CHAR(50), Address Address\_UDT, Birthdate DATE,  
    REF (Movie\_UDT) SCOPE Student ARRAY[30] )

# Typ reference a identita objektu

18

- ❑ Typ reference je možné použít na jednoznačnou identifikaci řádků tabulky a na definování vztahu mezi tabulkami
- ❑ Reference umožňují sdílení řádků mezi tabulkami a umožňují uživatelům vyjádřit složité spojení tabulek pomocí mnohem jednoduššího vyjádření cesty.
- ❑ Tabulka může mít sloupec typu reference který slouží na identifikaci jejích řádků (OID). Tímto sloupcem může být systémem generovaná a udržovaná hodnota.
- ❑ Je-li T uživatelem definovaný typ, potom  $REF(T)$  je typ, odkazující na řádek typu T. Reference může mít zadán rozsah – daný názvem relace, na jejíž řádky se odkazujeme.
  - ❑ například  $REF(T) SCOPE(R)$

# Vytvoření identifikátoru pro tabulku

19

- ❑ Abychom se mohli odkazovat na řádky tabulky, musí tabulka mít pro každý řádek vytvořen identifikátor (ID).
- ❑ V příkazu CREATE TABLE, pokud je typ tabulky uživatelem definovaný, může být zahrnut následující element:

REF IS <název atributu> <jak je vytvořen>

- ❑ Název atributu je název sloupce s identifikátorem
- ❑ Jak je vytvořen může udávat:
  - SYSTEM GENERATED
  - DERIVED (systém použije primární klíč na generování ID)

# Reference - příklad

20

```
CREATE TYPE StarType AS (  
  Name char(30), address AddressType,  
  bestMovie REF(movieType) SCOPE Movies)
```

Vytvoření ID pro tabulku

```
CREATE TYPE MovieType AS (  
  title char(30), year INTEGER, genre CHAR(10))  
CREATE TABLE Movies OF MovieType(  
  REF IS MovieID SYSTEM GENERATED,  
  PRIMARY KEY (title, year))
```

- ❑ *REF IS SYSTEM GENERATED* indikuje, že aktuální hodnoty příslušného typu REF jsou vytvořeny systémem.

# Typ ROW

21

- ❑ Definuje řádek dat - je tvořen posloupností dvojic jméno/typ.
  - ❑ Příklad:  
ROW (ulice varchar(200), město(20) varchar, PSČ varchar(5))
- ❑ Umožňuje, aby celé řádky mohly být:
  - ❑ uložené v proměnných,
  - ❑ předané jako parametry do procedur a funkcí ,
  - ❑ vrácené jako návratové hodnoty při volání funkcí.
- ❑ Umožňuje také, aby sloupec tabulky obsahoval hodnoty typu ROW. Typ ROW je v podstatě tabulka vnořená do tabulky.

# Metody

22

- ❑ Deklarace metody je podobná jako u uložené funkce – každá vrací hodnotu určitého typu.
- ❑ Metoda vyžaduje
  - ❑ Deklaraci v CREATE TYPE
  - ❑ Separátní definici v příkaze CREATE METHOD

Příklad:

```
CREATE TYPE AddressType AS (Street char (50), city  
    char (20))  
METHOD houseNumber() RETURNS Char(10))  
CREATE METHOD houseNumber() RETURNS Char(10)  
FOR AddressType  
Begin ..... END
```

# Příklad

23

- ❑ Máme vyjádřit vztah (M:N) mezi filmy a herci použitím referencí.
- ❑ Nejdříve musíme redefinovat tabulku MovieStar, aby se na ni dalo odkazovat  
CREATE TABLE MovieStar OF StarType (  
REF IS starID SYSTEM GENERATED  
PRIMARY KEY name)
- ❑ Vyjádříme vztah  
CREATE TABLE StarsIn(  
Star REF(StarType) SCOPE movieStar,  
Movie REF(movieType) SCOPE Movies)

# Operace s O-R daty

24

- ❑ Všechny běžné relevantní operátory jazyka SQL lze použít na tabulky, mající UDT.
- ❑ Některé operátory mají modifikovanou syntaxi
- ❑ Jsou definovány nové operátory



# Sledování referencí

25

- ❑ Předpokládejme, že  $x$  je hodnota typu  $\text{REF}(T)$ . Potom  $x$  odkazuje na některý řádek  $t$  typu  $T$ . Můžeme získat samotnou hodnotu  $t$  nebo její komponenty dvěma způsoby:
- ❑ Operátor  $\rightarrow$  má v podstatě stejný význam jako v jazyce C – tj. je-li  $x$  odkaz na řádek  $t$ , a je-li  $a$  atribut řádku  $t$ , tak  $x \rightarrow a$  je hodnota atributu  $a$  v řádku  $t$ .
- ❑ Operátor  $\text{DEREF}$ , který se aplikuje na odkaz a vrací odkazovaný řádek

# Příklad

26

- ❑ Máme najít filmy, ve kterých hraje Brad Pitt (tabulka StarsIn)

```
SELECT Deref(movie)
FROM StarsIn
Where star->name= 'Brad Pitt'
```

# Typ kolekce

27

- ❑ Kolekce jsou konstruktory typu, používané na definování kolekce jiných typů.
- ❑ Používají se na uložení více hodnot v jednom sloupci a mohou vyústit do vhnízděné tabulky
- ❑ Typ kolekce může být
  - ARRAY
  - MULTISSET
  - LIST
  - SET

# Kolekce

28

- ❑ ARRAY - jednorozměrné pole s daným maximálním počtem prvků
- ❑ MULTISSET – neuspořádané kolekce která může obsahovat duplicity
- ❑ LIST – uspořádané kolekce, která může obsahovat duplicity
- ❑ SET – neuspořádané kolekce, která neobsahuje duplicity