



# Бази от данни

Обектно-релационен модел на данни

# Обектно-релационен модел

---

- ▶ След появата на ОО СУБД-та в началото на 90-те, те не успяват да се наложат и към средата на 90-те години отпадат като начин на реализиране на СУБД
- ▶ Така преходът от релационни към ОО СУБД не се реализира
- ▶ Въпреки това ОО подход оказва голямо влияние върху релационните СУБД и в резултат на това се появява обектно-релационния модел.
- ▶ Този модел е в основата на един от най-съвремените стандарти SQL99



# От релации към обектни-релации

---

- ▶ В обектно-релационния модел релацията отново е основна концепция. За разлика от релационния модел обаче, в обектно-релационния модел са добавени следните характеристики:
  - ▶ Въвеждат се структурни типове за атрибутите – подобно на ODL
  - ▶ Въвеждат се методи, които могат да се прилагат към кортежите на релацията
  - ▶ Въвеждат се идентификатори на кортежите
  - ▶ Въвеждат се референции към кортежи



# Вгнездени релации

---

- ▶ Релации, които са разширени с първата характеристики (позволяват се структурни типове за атрибутите), често се наричат вгнездени релации.
- ▶ В модела допускащ вгнездени релации атрибутите могат да имат неатомарен тип. Релационните схеми също се разглеждат като възможен тип на атрибутите.
- ▶ Като резултат се получава следната индуктивна дефиниция за тип на атрибут и схема на релация:
  - ▶ База – всеки атрибут може да има атомарен тип;
  - ▶ Стъпка – схемата на една релация се състои от имена за всеки един от атрибутите, като всеки атрибут може да има произволен тип, включително схема на релация.



# Вложени релации

---

- ▶ При релационни модел, не се указва типа на атрибутите в схемата на релацията (те не влияят на релационните концепции)
- ▶ При вгнездените релации, ако типът на атрибута е атомарен също не го указваме, но ако е неатомарен (релационна схема), имената на атрибутите на тази схема ще се задават в скоби след името на атрибута. Тези атрибути от своя страна също могат да имат тип релационна схема и по този начин да получим вгнездена релационна схема на произволни ниски нива.



# Вгнездени релации - пример

- ▶ Например по-долу е показан екземпляр на вгнездена релация – Stars
- ▶ Тя има следната схема:
  - ▶ Stars(name, address(street, city), birthdate, movies(title, year, length))
  - ▶ Чрез атрибута movies се указват филмите, в които актьора е участвал.
  - ▶ Чрез атрибута stars се указват адресите, на актьора.

name	address		birthdate	movies		
Fisher	street	city	9/9/99	title	year	length
	Maple	Hwood		Star Wars	1977	124
	Locust	Malibu		Empire	1980	127
				Return	1983	133
Hamill	street	city	8/8/88	title	year	length
	Oak	Bwood		Star Wars	1977	124
				Empire	1980	127
				Return	1983	133

# Вгнездени релации - пример

- ▶ От представения пример се вижда, че записите за филмите в които участват повече от един актьор се повтарят. Това е излишество.
- ▶ За да се разреши този проблем в модела се въвеждат референции към кортежи.
- ▶ Всеки атрибут може да има референция към кортеж или множество от кортежи от дадена схема.

name	address		birthdate	movies		
Fisher	street	city	9/9/99	title	year	length
	Maple	Hwood		Star Wars	1977	124
	Locust	Malibu		Empire	1980	127
				Return	1983	133
Hamill	street	city	8/8/88	title	year	length
	Oak	Bwood		Star Wars	1977	124
				Empire	1980	127
				Return	1983	133

# Вгнездени релации - референции

- ▶ Ако атрибутът  $A$  има тип референция към кортеж от схема  $R$ , то в схемата, в която участва  $A$  отбелязваме това по следния начин  $A(*R)$
- ▶ В ODL аналог на това понятие е връзка от тип  $R$
- ▶ Ако атрибутът  $A$  има тип референция към множество кортеж от схема  $R$ , то в схемата, в която участва  $A$  отбелязваме това по следния начин  $A(\{*R\})$
- ▶ В ODL аналог на това понятие е връзка от тип  $\text{Set}\langle R \rangle$
- ▶ С примера по-долу се елиминира излишеството в горната релация

name	address		birthdate	movies		
Fisher	street	city	9/9/99			
	Maple	Hwood				
	Locust	Malibu				
Hamill	street	city	8/8/88			
	Oak	Bwood				

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



# Сравнение на ОО с ОР модел

---

- ▶ **Обектно-ориентирания (ОО) модел**, представен с ODL и **Обектно-релационния (ОР) модел**, който разглеждаме много си приличат.
- ▶ **Обекти и кортежи** – обектът е структура съдържащ компоненти за атрибути и връзки на обекта. Кортежът също може да се разглежда като структура, но с компоненти само за атрибутите. В обектно-релационни модел обаче, кортежът се разглежда като структура, с компоненти за атрибутите и указатели към други кортежи.
- ▶ **Разширения на класове и екземпляри на релации** – В ODL всички обекти съществуват в разширението на класа. В обектно-релационния модел може да има няколко релации с идентични схеми, но е различни екземпляри.
- ▶ **Методи** – Както и в обектно-ориентирания модел, така и в обектно-релационния модел се позволява възможности за дефиниране и използване на методи.



# Сравнение на ОО с ОР модел

---

- ▶ **Система за типизация** – Системите за типизация в ODL и в обектно-релационния модел са подобни. И двете системи поддържат атомарни типове и конструктори за създаване на нови типове. Конструкторите се различават в реализациите, но навсякъде се поддържат множества и мултимножества. Също така, типът множество от структури играе централна роля и в двата модела. В ODL това е типът на всеки клас, а в обектно-релационния модел това е типът на всяка релация.
- ▶ **Референции и идентификатори на обекти** – В чистия обектно-ориентиран модел идентификаторите на обектите са скрити от потребителя и той няма достъп до тях. В обектно-релационен модел референциите към кортежи се задават като част от типа на атрибут и при определени условия, те подлежат на манипулация. Тази възможност позволява значителна гъвкавост, но е сериозен източник на грешки.
- ▶ **Проблем за съвместимост** – Разликите между двата модела са много малки. Въпреки това обектно-релационен модел се наложи при разработката на СУБД, за разлика от обектно-ориентирания модел. Обектно-ориентираните системи се появиха прекалено късно, едва когато релационните системи се наложиха на пазара. При въвеждането на ОР СУБД-та доставчиците, гарантираха съвместимост с чисто релационните СУБД. От друга страна миграцията към чисти ОО СУБД от релационно СУБД изисква пълно пренаписване на системата.



# Преобразуване на ODL към ОР схеми

---

- ▶ Проблемите, които се появиха при преобразуване на ODL към чисто релационна схема произтича от това, че в ODL се допускат по-богати конструкции – атрибути от неатомарен тип, връзки, методи.
- ▶ С разширенията, които добавихме в ОР модел, преобразуването на ODL към ОР схема се улеснява значително.
- ▶ Неатомарните типове в ODL директно могат да се образуват към съответните им еквивалентни обектно-релационни типове.
- ▶ В някои случаи, обаче, конкретната ОР схема може да не поддържа определен неатомарен тип и тогава се налага да се използва подходящо моделиране, подобно на това при преобразуване на ODL към чисто релационна схема.



# Преобразуване на ODL към ОР схеми

---

- ▶ Връзките в обектно-релационния модел могат да се представят чрез релации, основавайки се на ключове, но за тяхното представяне могат да се използват и референции към кортежи.
- ▶ При преобразуване на ODL към чисто релационна схема изключихме от разглеждане методите, но това ограничение отпада в обектно-ориентирания модел, където се поддържат методи.



# Дефинирани от потребителя типове (UDT)

---

- ▶ DB2: UDT – User-Defined Types
- ▶ UDT – са разширение в SQL с което се реализират предимствата на обектно-ориентирания модел.
- ▶ UDT може да се използва по два начина:
  - ▶ За създаване на тип на атрибут в таблица
  - ▶ За създаване на тип структура



# Дефиниране на UDT като атрибут

---

- ▶ Стандартът SQL-99 позволява на потребителите да дефинират собствени типове.
- ▶ Най-простият начин е да се преименува вече съществуващ примитивен тип (INT, DATE и т.н.)
- ▶ Например:

```
CREATE TYPE T AS INTEGER;
```

- ▶ Целта на това преименуване е да не могат да се правят сравнения между стойности от един и същи тип, но с различно значение.
- ▶ Например KPH – километри в час и MPH – мили в час. И двете са цели числа, но не може да се сравняват. Аналогично CM и INCH
- ▶ При създаване на таблица KPH, MPH, CM, INCH могат да бъдат указани като тип на колона.



# Дефиниране на UDT като тип структура

---

- ▶ UDT може да се използва за създаване на тип структура (подобно на дефинирането на клас в ODL)
- ▶ Например:

```
CREATE TYPE ADDRESS_TYPE AS (  
    STREET CHAR(50),  
    CITY CHAR(20)  
);
```

```
CREATE TYPE STAR_TYPE AS (  
    NAME CHAR(30),  
    ADDR ADDRESS_TYPE  
);
```



# Деклариране на методи в UDT

---

- ▶ Нека разширим дефиницията на типа ADDRESS\_TYPE с метод HOUSE\_NUM, който връща номера на имота в адреса. Например ако имаме адрес 123 Maple Str., този метод ще върне 123
- ▶ В дефиницията на типа се поставя само декларация на метода, на следващия слайд е показана дефиницията на самия метод
- ▶ Пример:

```
CREATE TYPE ADDRESS_TYPE AS (  
    STREET CHAR(50),  
    CITY CHAR(20)  
)  
METHOD HOUSE_NUM() RETURNS CHAR(10);
```





# Дефиниране на методи в UDT

---

- ▶ Ако методът има параметри те трябва да се запишат ето така (A INT, B CHAR(5) )

```
CREATE METHOD HOUSE_NUM()  
RETURNS CHAR(10)  
FOR ADDRESS_TYPE  
BEGIN  
  
.....  
END;
```



## Пример в DB2 за дефиниране на структура ред

---

```
CREATE TYPE myRowType AS ROW  
(  
    key INTEGER,  
    data VARCHAR(100)  
);
```



# Пример в DB2 за дефиниране на масив

---

## ► Масив

```
CREATE TYPE arrType AS  
INTEGER ARRAY[10];
```

## ► Асоциативен

```
CREATE TYPE arrType2 AS  
INTEGER ARRAY[VARCHAR(100)];
```

```
CREATE TYPE arrType3 AS  
myRowType ARRAY[VARCHAR(100)];
```



## Пример в DB2 за дефиниране на референция

---

### ► Scalar anchoring

```
DECLARE empSalary ANCHOR  
employee.salary;
```

### ► Row anchoring

```
DECLARE emp ANCHOR ROW employee;
```



# Обектно-релационен модел (обобщение)

---

- ▶ В обектно-релационния модел релацията отново е основна концепция.
- ▶ За разлика от релационния модел обаче, в обектно-релационния модел са добавени следните характеристики:
  - ▶ Въвеждат се структурни типове за атрибутите – подобно на ODL
  - ▶ Въвеждат се методи, които могат да се прилагат към кортежите на релацията
  - ▶ Въвеждат се идентификатори на кортежите
  - ▶ Въвеждат се референции към кортежи
- ▶ Обектно-релационни модел се наложи при разработката на СУБД, за разлика от обектно-ориентирания модел.
- ▶ При въвеждането на ОР СУБД-та доставчиците, гарантираха съвместимост с чисто релационните СУБД.

