5. Области (домейни)

Лекционен курс “Бази от данни”

Определения

Най-малката семантична единица данни (наричат се още скаларни) в релационния модел е индивидуална стойност на данни – напр. ISBN номер на книга.

От гледна точка на релационния модел тези стойности са неделими (atomic).

Да отбележим, че в примера ISBN номерът има своя вътрешна структура, т.е. може да бъде разделен на съставящи го елементи – цифри, тирета, букви.
Но в този случай той ще загуби своето **значение**.

Област - именувано множество от скаларни стойности, всички от един и същ тип.

Областта е множеството от всички възможни стойности, които дефинираните върху нея атрибути могат да имат.

Пример: областта на годините, в които са публикувани съответните книги – множество от цели числа между 1600 и 2011, т.е. в този диапазон стойността би имала смисъл, извън него годината на публикуване би била нереална.

Областите са един вид пулове от стойности, от които се вземат актуалните стойности на атрибутите.

Или по-точно, всеки атрибут трябва да бъде дефиниран върху точно един прилежащ домейн.

Сравнения, ограничени върху областите

Какво е значението на областите?

Ако два атрибута вземат своите стойности от един и същ домейн, тогава сравнението (и по този начин обединяването, сливането и т.н. върху тези атрибути) има смисъл.

Например:

- Сравнението между възрастта на определен служител и бюджета на отдела, в който той работи, е математически възможно, защото те са числа, т.е. от един и същ тип;
- Но логически (в реалния свят) това сравнение е абсурдно – тези два атрибута имат различна семантика;
- Дефинирайки ги върху различни домейни, можем да предотвратим съхраняването на нереални ситуации в БД.

```
SELECT ….
FROM EMP, DEPT
WHERE EMP.AGE = DEPT.BUDGET
```

```
SELECT ….
FROM EMP, DEPT
WHERE EMP.DEPT_ID = DEPT.DEPT_ID
```

Дефиниране на данни

Трябва да се отбележи, че домейните са от концептуално естество:

- Те не се съхраняват като множество от стойности в БД – те са част от дефиницията на БД;
- Всяка дефиниция на атрибутите включва референция към кореспондиращ домейн - системата ще има информация за съвместимостта на атрибутите;
- Напр. област (домейн) може да бъде дефинирана посредством оператора
  CREATE DOMAIN domain data-type;
  (DDL на SQL).

```
CREATE DOMAIN YEAR_PUB INT;
CREATE DOMAIN BOOK_TITLE VARCHAR(50);

CREATE BASE RELATION BOOK
(
  BOOK_ISBN VARCHAR(20) NOT NULL,
  TITLE  DOMAIN (BOOK_TITLE),
  PRICE INT,
  YEAR_PUBLISHED DOMAIN(YEAR_PUB)
);
```

- При създаване на нова област (CREATE DOMAIN) СУБД генерира нова входна точка в системния каталог, която специфицира тази нова област;

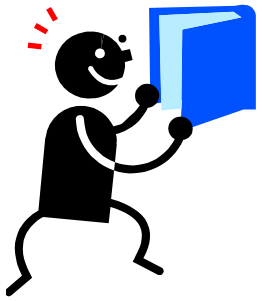
- CREATE BASE RELATION - включва референции към кореспондиращите области и създава нова входна точка в каталога на релациите.

Съставни области

- Една съставна област е дефинирана като *Декартово произведение* на множество от прости области;
- Напр. съставната област DATE може да бъде дефинирана като Декартовото произведение на простите области DAY, MONTH, YEAR

DAY = (1, 2, ..., 31)
MONTH = (1, 2, ..., 12)
YEAR = (00, 01, ..., 99)

DATE = (01 01 00
02 01 00
...
31 01 00
....
31 12 99)



- Атрибут, дефиниран върху DATE областта, може да има 37 200 (31 * 12 * 100) стойности
- Не всички са валидни!

```
CREATE DOMAIN DAY CHAR(2);
CREATE DOMAIN MONTH CHAR(2);
CREATE DOMAIN YEAR CHAR(2);
CREATE DOMAIN DATE (MONTH, DAY, YEAR);

CREATE TABLE EMPLOYEE
(
  EMP# ..... NOT NULL,
  ENAME..... ,
  HIREDATE DOMAIN (DATE) (HIREMONTH, HIREDAY, HIREYEAR),
  DEPT#..... ,
  SALARY.....
);
```

DML операторите ще бъдат в състояние да референцират съставния атрибут HIREDATE и простите HIREDAY, HIREMONTH, HIREYEAR.

Напр.:

```
SELECT *
FROM   EMPLOYEE
WHERE  HIREYEAR < ‘05';
```

```
SELECT *
FROM   EMPLOYEE
 WHERE HIREDATE = '010105';
```

Изтриване на области

След като има оператор за създаване на домейн, би трябвало да има и такъв за изтриването му:

DESTROY DOMAIN domain;

Изтрива областта от каталога.