

Auxiliar Ocho - CC20A

Computación II - Sección 2

Oscar E. Alvarez Callaú
oalvarez@dcc.uchile.cl

Santiago - Chile, Oct/25/2007

1. Dependencias Funcionales

Son un conjunto de restricciones y definiciones semánticas que tiene los datos o atributos de un esquema de relación.

1.1. Definición

$$x \rightarrow y : \forall t_1, t_2 \in t(R) / t_1[x] = t_2[x] \implies t_1[y] = t_2[y]$$

Esta es la definición formal de una dependencia funcional

1.2. Reglas de inferencia (axiomas - Armstrong)

Conjunto de reglas base para la inferencia de otras dependencias funcionales.

Reflexiva

$$x \supseteq y \implies x \rightarrow y$$

Aumento

$$x \rightarrow y \models xz \rightarrow yz$$

Transitiva

$$x \rightarrow y, y \rightarrow z \implies x \rightarrow z$$

2. SQL (cont.)

Dada las siguientes relaciones implementadas en un DBMS(sistema administrador de base de datos) con especificación SQL2:

Arrendatario (rut,nombre,apellido)

Telefono (rut,fono), teléfonos tanto de los dueños como de los arrendatarios

Dueño (rut,nombre,apellido)

Casa (id_casa,rut,nro,calle,comuna), rut es llave externa de Dueño

Arrienda (rut,id_casa,deuda), rut es llave externa de Arrendatario

Se pide entonces(por simplicidad definimos que las intrucciones en SQL no implementan case-sensitive permitiendo que A=a):

1. Los arrendatarios que arriendan la casa ubicada en la calle Carrera no 1024, Santiago.

```
SELECT A.RUT, A.Nombre, A.Apellido
FROM Arrendatario A, Arrienda B, Casa C
WHERE A.RUT=B.RUT AND B.Id_casa=C.Id_casa
      AND C.Calle='Carrera' AND C.Nro='1024' AND C.Comuna='Santiago' ;
```

2. Cuánto le deben a María Pérez?, suponiendo que hay una sola(condición diferente a la vista en clases)

```
SELECT SUM(A.Deuda) FROM Arrienda A, Casa B, Dueo C
WHERE A.Id_casa=B.Id_casa AND B.RUT=C.RUT
      AND C.Nombre='Maria' AND C.Apellido='Perez' ;
```

- 3.Cuál es la deuda total para cada dueño?, en este caso a diferencia de la auxiliar nuestro información completa de los dueños.

```
SELECT SUM(A.Deuda), C.RUT, C.Nombre, C.Apellido
FROM Arrienda A, Casa B, Dueno C
WHERE A.Id_casa=B.Id_casa AND B.RUT=C.RUT
GROUP BY C.RUT ;
```

4. Liste todas las personas de la base de datos.

```
SELECT * FROM Arrendatario UNION SELECT * FROM Dueno ;
```

5. Indique los dueños que poseen tres o más casas

```
SELECT A.RUT, A.Nombre, A.Apellido
FROM Dueno A, Casa C
WHERE A.RUT=C.RUT
GROUP BY A.RUT
HAVING COUNT(C.Id_casa)>=3 ;
```

6. Liste los dueños que tengan deudores en todas sus casas, corrigiendo la solución vista en clase(ver las clausulas *where*)

```
SELECT DISTINCT D.RUT
FROM CASA D
WHERE NOT EXISTS ((SELECT A.RUT, A.ID_CASA
                   FROM Casa A
                   WHERE A.RUT=D.RUT)
                  EXCEPT
                  (SELECT B.RUT, B.ID_CASA
                   FROM Casa B, Arrienda C
                   WHERE B.RUT=D.RUT
                   AND C.ID_CASA=B.ID_CASA AND A.DEUDA>0))
```

7. Encuentre el máximo número de arrendatarios por casa.

```
SELECT COUNT(A.RUT)
FROM Arrienda A
GROUP BY A.Id_casa
HAVING COUNT(A.RUT) >= ALL(SELECT COUNT(B.RUT)
                           FROM Arrienda B
                           GROUP BY B.Id_casa) ;
```