

Ejercicios sobre Álgebra y Cálculo Relacional

1. Sobre las relaciones:

PROVEEDOR S(CODPRO,NOMPRO,STATUS,CIUDAD)
PIEZA P(CODPIE,NOMPIE,COLOR,PESO,CIUDAD)
PROYECTO J(CODPJ,NOMPJ,CIUDAD)
VENTAS SPJ(CODPRO,CODPIE,CODPJ,CANTIDAD,FECHA)

Realizar las siguientes consultas en Álgebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- a) Encontrar los códigos de los proveedores que suministran alguna pieza a J1.
- b) Encontrar los suministros cuya cantidad supere las 100 unidades.
- c) Encontrar los nombres de proveedores, piezas y proyectos que se encuentren en la misma ciudad.
- d) Encontrar los nombres de las piezas suministradas por los proveedores de Londres.
- e) Encontrar todas las parejas de ciudades tales que la primera sea la de un proveedor y la segunda la de un proyecto entre los cuales haya algún suministro.
- f) Encontrar los códigos de las piezas suministradas a algún proyecto por un proveedor que se encuentre en la misma ciudad que el proyecto.
- g) Encontrar los códigos de los proyectos que tienen al menos un proveedor que no se encuentre en su misma ciudad.
- h) Mostrar todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos.
- i) Mostrar todas las ciudades de los proveedores en las que no fabriquen piezas.
- j) Mostrar todas las ciudades de los proveedores en las que además se fabriquen piezas.
- k) Encontrar los códigos de los proyectos que usan una pieza que vende S1.
- l) Encontrar la cantidad más pequeña enviada en algún suministro.
- m) Encontrar los códigos de los proyectos que no utilizan una pieza roja suministrada por un proveedor de Londres.
- n) Encontrar los códigos de los proyectos que tienen como único proveedor a S1.
- \tilde{n}) Encontrar los códigos de las piezas que se suministran a todos los proyectos de París.
- o) Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos.
- p) Encontrar los códigos de los proyectos a los que S1 suministra todas las piezas existentes.
- q) Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.

2. Sobre las relaciones:

```
LISTA_BODA(REF#,DESCRIPCION,PRECIO)
INVITACIONES(NOMBRE,DIRECCION,CIUDAD)
CONFIRMAN(NOMBRE,NUMERO)
RESERVA_REGALO(NOMBRE,REF#,FECHA)
```

Realizar las siguientes consultas en Álgebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- Encontrar los regalos (descripción) que no han sido reservados.
- Encontrar la dirección de los invitados que confirman la asistencia de más de dos personas.
- Encontrar el nombre y la referencia del regalo más caro ya reservado.

3. Se tiene una base de datos con la siguiente estructura:

```
HOMBRES(NomH,Edad)
MUJERES(NomM,Edad)
HSimM(NomH,NomM). El hombre NomH cae simpático a la mujer NomM.
MSimH(NomM,NomH). La mujer NomM cae simpática al hombre NomH.
MATRIM(NomH,NomM). NomH y NomM están casados.
```

Realizar las siguientes consultas en Álgebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- Hallar las parejas de hombres y mujeres que se caen mutuamente simpáticos, con edades entre 20 y 30 años y que no estén casados entre sí.
- Hallar las mujeres casadas a las que no cae simpático su marido.
- Hallar los hombres a los que no les cae simpática ninguna mujer.
- Hallar las mujeres casadas a las que no les cae simpático ningún hombre casado.

4. Se tiene una BD que almacena la información de una empresa de transporte de materiales de construcción con el siguiente esquema:

| |
|--|
| CONDUCTOR(<u>DNI</u> ,NOMBRE,DIREC,PROV) |
| VEHICULO(<u>MATRICULA</u> ,CARGA_MAX,FECHA_COMPRA) |
| ruta(<u>RUTA#</u> ,CIUDAD_SAL,CIUDAD_LLEG,KM) |
| VIAJE(<u>VIAJE#</u> ,DNI,MATRICULA) |
| PROG_VIAJE(<u>VIAJE#</u> , <u>RUTA#</u> , <u>DIA_SEM</u> , <u>HORA_SAL</u> , <u>HORA_LLEG</u>) |

Realizar las siguientes consultas en Álgebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- a) Encontrar entre qué dos ciudades se realiza el viaje más largo.
- b) Listar los nombres de los conductores que hayan llevado todos los camiones de la empresa.
- c) Encontrar qué días de la semana se hacen viajes entre Granada y Sevilla por la mañana (antes de las 13h.).
- d) Encontrar las rutas que se hacen todos los días de la semana, suponiendo que hay viajes todos los días.

5. Se tiene una BD con el siguiente esquema:

| |
|--|
| REPRESENTANTE(<u>DNI</u> ,NOMBRE,DIREC,PROVINCIA) |
| ZONA_REP(<u>DNI</u> , <u>COD_ZONA</u> ,POBLACION,PROVINCIA) |
| PEDIDOS(<u>DNI</u> , <u>COD_ART</u> ,CANTIDAD,POBLACION) |
| ARTICULO(<u>COD_ART</u> ,NOMBRE,COLOR,PROV_FAB) |

Realizar las siguientes consultas en Álgebra Relacional y en Cálculo Relacional Orientado a Tuplas:

- a) Listar las provincias que son visitadas por todos los representantes.
- b) Encontrar los representantes que venden fuera de su provincia artículos fabricados en su provincia.
- c) Obtener las poblaciones de Granada que hayan superado los 50.000 euros de facturación y quién realizó el pedido.
- d) Mostrar las zonas que incluyen a una sola población.
- e) Encontrar el código del artículo vendido en mayor cantidad.

6. Se tiene una BD sobre la información de un congreso con un esquema formado por las siguientes tablas:

| |
|--|
| INSCRIPCIONES(<u>INS#</u> ,NOMBRE,PROCEDENCIA) |
| SESIONES(<u>SES#</u> ,TÍTULO,COORD#) |
| ARTICULOS(<u>ART#</u> ,TÍTULO) |
| PROGRAMA(<u>SES#</u> , <u>ART#</u> ,FECHA,HORA_INICIO,HORA_FIN,SALA,PONENTE#) |

Teniendo en cuenta que tanto los coordinadores como los ponentes han de estar inscritos en el congreso, resolver las siguientes consultas mediante Álgebra Relacional y mediante Cálculo Relacional Orientado a Tuplas:

- a) Mostrar los nombres de los ponentes que coordinan su propia sesión.
- b) Seleccionar los coordinadores que coordinan una única sesión.
- c) Mostrar el título de los artículos que se exponen en primer y último lugar.

3)

a) $\pi_{\text{codpro}} (\sigma_{\text{cod.pro} = 34} (\text{Ventas}))$

b) $\sigma_{\text{ant-renta} > 100} (\text{Ventas})$

c) $\pi_{\text{codpro}, \text{codpie}, \text{codpj}} (\sigma_{\text{ciudad} = p. \text{ciudad} \wedge j. \text{ciudad} \wedge (3 \neq p \neq 3)})$

d) $\pi_{\text{codpie}, \text{nombre}} (\sigma_{\text{ciudad} = \text{Londres}} (3 \neq p)) \times \pi_{\text{nombre}} (\pi_{\text{cod.pro} = \text{Ventas}} \bowtie \sigma_{\text{ciudad} = \text{Londres}} (3)) \bowtie \pi_{\text{ciudad}}$

e) Solución clase: $\pi_{\text{ciudad}, \text{E-ciudad}} (\sigma_{\text{codpro} \bowtie (3 \neq p \neq 3)})$

f) Solución clase: $\pi_{\text{codpie}} ((\sigma_{3 \neq p \neq 3}) \bowtie \pi_{\text{cod, ciudad}} (3)) \bowtie \pi_{\text{cod, ciudad}} (3)$

g) $\pi_{\text{codpj}} (\sigma_{\text{ciudad} = p. \text{ciudad}} ((\sigma_{3 \neq p \neq 3}) \bowtie \pi_{\text{codpj, ciudad}} (3)) \bowtie \pi_{\text{codpro, ciudad}} (3))$

h) $\pi_{\text{ciudad}} (P) \cup \pi_{\text{ciudad}} (3)$

i) $\pi_{\text{ciudad}} (3) - \pi_{\text{ciudad}} (P)$

j) $\pi_{\text{ciudad}} (3) - (\pi_{\text{ciudad}} (3) - \pi_{\text{ciudad}} (P)) \rightarrow \text{ciudades donde no se han retirado peticiones}$ Solución de clase: $\pi_{\text{ciudad}} (3) \cap \pi_{\text{ciudad}} (P)$

k) $\pi_{\text{codpj}} (\sigma_{\text{codpro} = 84} (3 \neq p))$ // complejizándolo un poco, encontrar proyectos que usen peticiones que administre 84, aunque no la administre él.

l) $p(3 \neq p) = 3 \neq p \neq 2$ $\pi_{\text{codpj}} ((\pi_{\text{codpie}} (\sigma_{\text{codpro} = 84} (3 \neq p))) \bowtie 3 \neq p)$

$\pi_{\text{contradict}} (3 \neq p) - \pi_{\text{contradict}} (\sigma_{\text{3ps.contradict} > \text{3ps2.contradict}} (3 \neq p \neq 2))$

Solución clase (has que tiene el que nos dimos las operaciones)

$A \neq B = 3 \neq p$

$\pi_{\text{contradict}} (A) - \pi_{\text{contradict}} (\sigma_{A. \text{contradict} > B. \text{contradict}} (A \neq B))$

m) $\pi_{\text{codpj}} (\sigma_{\text{ciudad} = \text{Londres} \wedge \neg (\text{color} = \text{rojo})} ((3 \neq p \neq 2) \bowtie \pi_{\text{codpro, ciudad}} (3)) \bowtie \pi_{\text{codpie, color}} (P))$

Solución de clase: 3ps no tendrán en cuenta los que no tienen peticiones asociadas

$\pi_{\text{codpj}} (3) - \pi_{\text{codpj}} ((\sigma_{\text{color} = \text{rojo}} (P) \bowtie 3 \neq p) \bowtie \sigma_{\text{ciudad} = \text{Londres}} (3))$

n)

ñ) $\pi_{\text{codpie}} (\sigma_{\text{ciudad} = \text{Paris}} (3 \neq p \neq 2))$

- o) • código propietarios
- misma peticion a todos los proyectos

p)