

Bases de Datos y Sistemas de Información
Curso 2004-5. Ingeniería Superior
Primer Parcial.

- 1) (2.5 puntos) Dados los esquemas de relación
Clientes(numCl, nombre, apellido, teléfono, códigoPostal)
Pedidos(numPed, numCl, fechaPed, numArtículos)
- a) Plantear una consulta SQL con una reunión (join) que nos diga el nombre y teléfono de los clientes con código postal 28039 que tengan algún pedido anterior al 01-01-2005.
 - b) Plantear la misma consulta utilizando subconsultas correlacionadas (con EXISTS)
 - c) Plantear la misma consulta con subconsultas no correlacionadas (con IN)
 - d) Listar todos los clientes que tengan un total de más de 20 artículos es sus pedidos. Los que no tengan más de 20 artículos también deben aparecer, mostrando un valor nulo en lugar de la suma total de artículos.
 - e) Mostrar con un orden descendente todos los números de clientes con el número de pedidos que ha realizado (que es cero si no se han realizado pedidos) sin usar reunión externa (outer join).

Sol

- a) SELECT DISTINCT NOMBRE, TELEFONO
FROM CLIENTES NATURAL INNER JOIN PEDIDOS
WHERE CODIGOPOSTAL = '28039'
AND FECHAPED < '01-01-2005'
- b) SELECT NOMBRE, TELEFONO
FROM CLIENTES C
WHERE CODIGOPOSTAL = '28039'
AND EXISTS (SELECT NUMPED
FROM PEDIDOS
WHERE NUMCL = C.NUMCL
AND FECHAPED < '01-01-2005')
- c) SELECT NOMBRE, TELEFONO
FROM CLIENTES C
WHERE CODIGOPOSTAL = '28039'
AND NUMCL IN (SELECT NUMCL
FROM PEDIDOS
WHERE FECHAPED < '01-01-2005')
- d) SELECT NOMBRE, C.NUMCL, TELEFONO, CL20.TOTALART
FROM CLIENTES C
NATURAL LEFT OUTER JOIN (
SELECT NUMCL, SUM(NUMARTICULOS) AS TOTALART
FROM PEDIDOS
GROUP BY NUMCL
HAVING SUM(NUMARTICULOS)>20)AS CL20
- e) SELECT NUMCL, COUNT(*)
FROM PEDIDOS
GROUP BY NUMCL
UNION
SELECT NUMCL, 0

```

FROM CLIENTES C
WHERE NOT EXISTS(SELECT NUMPED
                  FROM PEDIDOS
                  WHERE NUMCL = C.NUMCL)

ORDER BY 1 DESC

```

- 2) (2 puntos) Mostrar la consulta a) con consultas del álgebra relacional, calculo relacional de tuplas, cálculo relacional de dominios y QBE

Álgebra relacional:

Salida $\leftarrow \pi_{\text{nombre, teléfono}}(\sigma_{\text{códigoPostal}=28039}(\text{Clientes}) \bowtie \sigma_{\text{fechaPed} < 01-01-2005}(\text{Pedidos}))$

Cálculo Relacional de tuplas

$$\{t \mid \exists u($$

$$((u \in \text{Clientes})$$

$$\wedge$$

$$(t[\text{nombre}] = u[\text{nombre}])$$

$$\wedge$$

$$(t[\text{teléfono}] = u[\text{teléfono}])$$

$$\wedge$$

$$(t[\text{códigoPostal}] = 28039)$$

$$\wedge$$

$$(\exists v \in \text{Pedidos} (v[\text{numCl}] = u[\text{numCl}])$$

$$\wedge v[\text{fechaPed} < 01-01-2005)$$

$$)$$

Cálculo relacional de Dominios

$$\{ \langle \text{nombre, teléfono} \rangle \mid$$

$$\exists \text{numCl, apellido, códigoPostal} (\langle \text{numCl, nombre, apellido, teléfono,}$$

$$\text{códigoPostal} \rangle \in \text{Clientes}$$

$$\wedge \langle \text{códigoPostal} \rangle = 28039)$$

$$\wedge$$

$$(\exists \text{numPed, fechaPed, numArticulos} (\langle \text{numPed, numCl, fechaPed, numArticulos} \rangle$$

$$\in \text{Pedidos}$$

$$\wedge \langle \text{fechaPed} \rangle < 01-01-2005)$$

$$)$$

QBE

<i>Clientes</i>	<i>nombre</i>	<i>numCl</i>	<i>apellido</i>	<i>teléfono</i>	<i>código Postal</i>
	P._x	P._n		P._y	28039

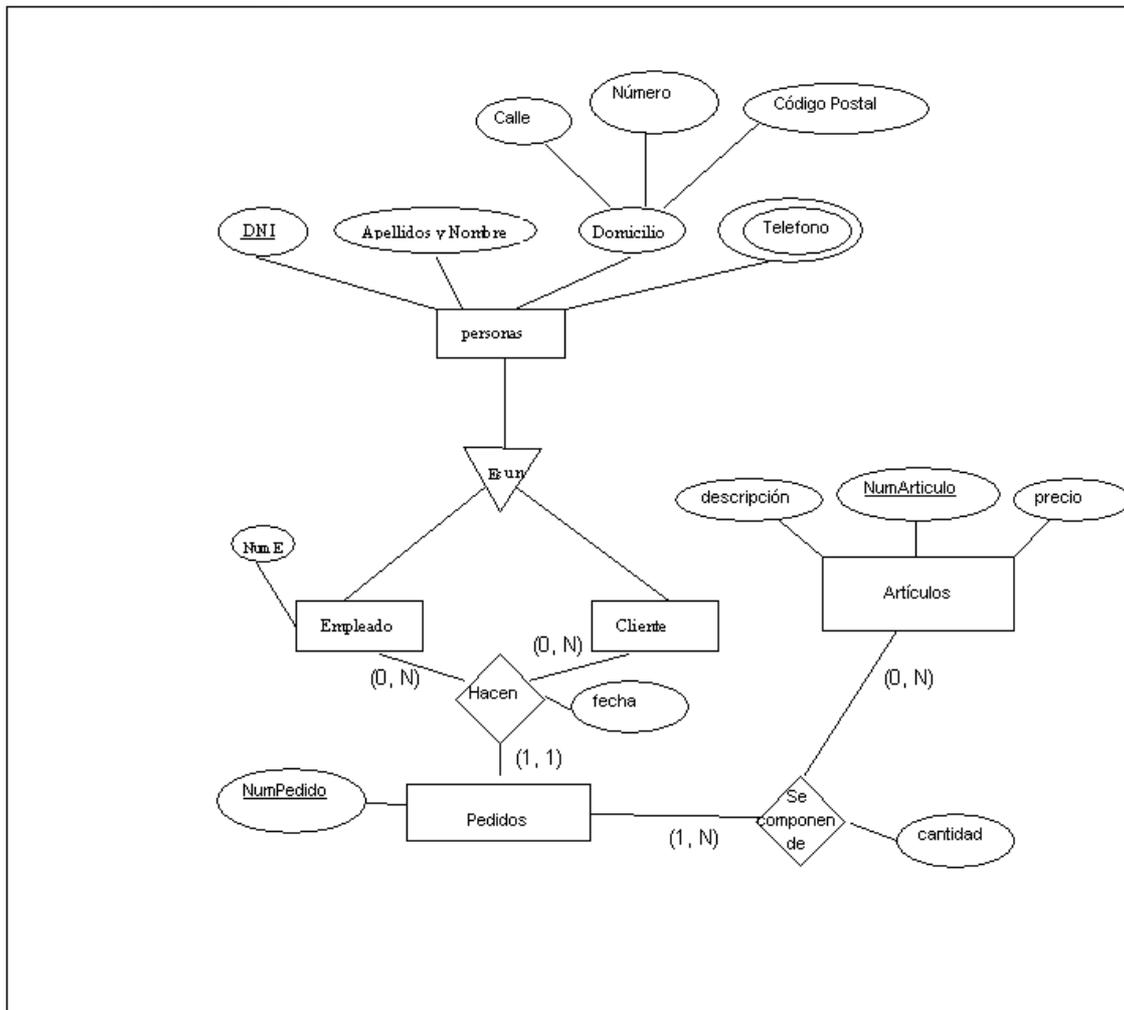
<i>Pedidos</i>	<i>numCl</i>	<i>fechaPed</i>
	_n	_z

<i>Condiciones</i>
01-01-2005 > _z

- 3) (2 puntos) La empresa PCS se dedica a vender ordenadores personales. Necesita un sistema de información para conocer el DNI, apellidos y nombre, distintos teléfonos y domicilio (con calle, número y código postal) de sus empleados y clientes. Los Empleados además tienen un número de empleado. Se debe conocer el DNI del cliente, del empleado y fecha de cada pedido, código y cuantía de cada artículo de un pedido, y una descripción y precio de cada artículo.
- Construir un modelo EER para la modelar los datos de la empresa.
 - Explicar las relaciones de cardinalidad mínimo-máximo.
 - Traducir a modelo relacional
 - Dar las restricciones de integridad referencial oportunas.

Sol:

a) EER y b) cardinalidad mínimo-máximo:



c) Modelo relacional

Reestructuración: Atributos multivalorados, Atributos compuestos, Generalizaciones.

Resultado:

Personas(DNI, NumEmpleado, Nombre, Apellido1, Apellido2, NumEmpl*)

Teléfonos(DNI, número)

Domicilios(Calle, número, códigoPostal, DNI)

Pedidos(NumPedido, DNIEmp, DNICli, fecha)

ArtPedidos(NumPedido, NumArtículo, Cantidad)

Articulos(NumArtículo, descripción, precio)

e) Restricciones de integridad referencial

$\Pi_{(DNI)}(\text{Teléfonos}) \subseteq \Pi_{(DNI)}(\text{Persona})$

$\Pi_{(DNI)}(\text{Domicilio}) \subseteq \Pi_{(DNI)}(\text{Persona})$

$\Pi_{(DNIEmp)}(\text{Pedidos}) \subseteq \Pi_{(DNI)}(\text{Persona})$

$\Pi_{(DNICli)}(\text{Pedidos}) \subseteq \Pi_{(DNI)}(\text{Persona})$

$\Pi_{(NumPedido)}(\text{Pedidos}) \subseteq \Pi_{(DNI)}(\text{Persona})$

$\Pi_{(NumPedido)}(\text{ArtPedidos}) \subseteq \Pi_{(NumPedido)}(\text{Pedidos})$

$\Pi_{(NumArtículo)}(\text{ArtPedidos}) \subseteq \Pi_{(NumArtículo)}(\text{Artículos})$

- 4) (1 punto) En el modelo de la empresa PCS se está estudiando la posibilidad de incluir en Pedido un total del importe, que se calcula sumando para cada artículo de un pedido el precio del artículo por la cuantía de artículos de ese tipo en el pedido. Dar una opinión razonada sobre la redundancia sabiendo que:
- Actualmente se tienen 1000 pedidos y hay 50 artículos en venta
 - Se realizan 30 pedidos nuevos cada día
 - En Internet se pregunta por el precio total de un pedido 40 veces al día.

Sol

Supongamos que en media hay N artículos por cada pedido. Los cálculos del coste de las dos operaciones con y sin redundancia, sin tener en cuenta la integridad referencial (que se podría incluir) es:

Coste sin redundancia

Op1:

Pedido: 1 escrituras

ArtPedidos: N escrituras

Op2:

ArtPedidos: N lecturas

Total Coste: $30 \cdot 2 \cdot (N+1) + 40 \cdot N = 60 + 100N$

Coste con redundancia

Op1:

Pedido: 1 escrituras

ArtPedidos: N escrituras

Artículos: N lecturas (para calcular el precio del pedido)

Op2:

ArtPedidos: 1 lectura

Total Coste: $30 \cdot 2 \cdot (N+1) + 30N + 40 = 100 + 90N$

Si hay más de 4 artículos por pedido en media, elegimos con redundancia.

- 5) (1.5 puntos) Dado el conjunto de dependencias funcionales $S = \{A \rightarrow B, C \rightarrow DA, D \rightarrow A, AB \rightarrow C\}$ en $R(A, B, C, D)$
- (1 punto) Encuéntrese un conjunto mínimo de dependencias funcionales equivalente
 - (0.5 puntos) Dar las claves candidatas de R.

Sol:

a) Paso 1: $G:=F$

Paso 2: $G = \{A \rightarrow B, C \rightarrow D, C \rightarrow A, D \rightarrow A, AB \rightarrow C\}$

Paso 3 (eliminación de atributos redundantes):

Veamos si A o B es redundante en $AB \rightarrow C$

$A \notin B^+_G = \{B\}$ luego A no es redundante en $AB \rightarrow C$

$B \in A^+_G = \{A, B, C, D\}$ luego B es redundante en $AB \rightarrow C$, y

$G = \{A \rightarrow B, C \rightarrow D, C \rightarrow A, D \rightarrow A, A \rightarrow C\}$

Paso 3: Eliminación de dependencias funcionales redundantes

$B \notin A^+_{\{C \rightarrow D, C \rightarrow A, D \rightarrow A, A \rightarrow C\}} = \{A, C\}$ luego $A \rightarrow B$ no es redundante

$D \notin C^+_{\{A \rightarrow B, C \rightarrow A, D \rightarrow A, A \rightarrow C\}} = \{C, A, B\}$ luego $C \rightarrow D$ no es redundante

$A \in C^+_{\{A \rightarrow B, C \rightarrow D, D \rightarrow A, A \rightarrow C\}} = \{C, D, A\}$ luego $C \rightarrow A$ es redundante y

$G = \{A \rightarrow B, C \rightarrow D, D \rightarrow A, A \rightarrow C\}$

$A \notin D^+_{\{A \rightarrow B, C \rightarrow D, A \rightarrow C\}} = \{D\}$ luego $D \rightarrow A$ no es redundante

$C \notin A^+_{\{A \rightarrow B, C \rightarrow D, D \rightarrow A\}} = \{A, B\}$ luego $A \rightarrow C$ no es redundante

Luego un conjunto mínimo de dependencias funcionales equivalente a S es

$\{A \rightarrow B, C \rightarrow D, D \rightarrow A, A \rightarrow C\}$

b) $A^+_S = \{A, B, C, D\}$

$B^+_S = \{B\}$

$C^+_S = \{C, D, A, B\}$

$D^+_S = \{D, A, B, C\}$

Las claves candidatas son A, C y D

- 6) (1 punto) Se tiene una relación caballosGenealógicos con atributos Padre e Hijo que implementa un árbol genealógico de caballos pura sangre (relacionando la entidad Caballos consigo misma). Expresar con SQL recursivo y con reglas Datalog una consulta para saber todos los pura sangre que son descendientes de "Rayo veloz"

Sol:

a) SQL recursivo:

```
with recursive descendientes(hijo, padre) as (  
  select hijo, padre  
  from caballosGenealógicos  
  where padre = "Rayo veloz"  
  union  
  select C.hijo, C.padre  
  from caballosGenealógicos C, descendientes D  
  where C.padre = D.hijo  
)  
select hijo  
from descendientes
```

b) Datalog

descendiente(X,Y) :- caballosGenealógicos(X, Y)

descendiente(X,Y) :- caballosGenealógicos(Z, Y), descendiente(X, Z).

Otra forma:

descRV(D) :- caballosGenealógicos("Rayo Veloz", Y)

descRV(D) :- caballosGenealógicos(P, D), descRV(P)