

Bases de Datos**Segundo Parcial – Modelo X**

06-06-2002

*Razonar las respuestas**Contestar TODAS las preguntas en hojas separadas**Entregar las preguntas en orden, así como los apartados de las preguntas**TEST: Respuesta correcta +3 puntos, incorrecta -1, en blanco +0**Sólo hay una respuesta correcta. En caso de duda, marcar la “más correcta”**Total de preguntas 10 (30 puntos)**Es necesario superar el mínimo del test (15 puntos) para poder aprobar el examen**Es necesario obtener al menos 2,25 puntos en el conjunto de las dos preguntas para poder aprobar el examen***TEST** (4,5 puntos)**PRIMERA PREGUNTA** (3,5 puntos)

Para la base de datos del anexo, escribir una expresión en SQL para las siguientes consultas (convertirla a SQL básico en caso de usar características no estándar o de SQL92):

- Código, nombre y marca de los vehículos de los que sólo se ha realizado una única venta
Escribir también esta consulta en QBE
- Código y nombre de las marcas que han vendido al menos un vehículo en todos los concesionarios de su mismo país.
- Código y nombre de los vehículos cuyo número de unidades almacenados en cada concesionario es superior a la media de unidades por vehículo almacenadas en el concesionario.

SEGUNDA PREGUNTA (2,00 puntos)

Encontrar una descomposición de producto sin pérdida (PSP), que conserve las dependencias (CD), y normalizada en 3FN o BCNF lo mejor posible (mínimo número de esquemas y de redundancia, etc.) del esquema de relaciones $R = (A, B, C, D, E, F, G, H, I, J, K)$ y el conjunto de dependencias F :

$$F = \left\{ \begin{array}{l} E \rightarrow J B \\ G \rightarrow E K \\ B J \rightarrow C G \\ D \rightarrow A F I \end{array} \right\}$$

Indicar la forma normal de cada relación y mostrar que la descomposición encontrada efectivamente es de producto sin pérdida y conserva las dependencias.

ANEXO

Base de datos relacional que almacena información referente a concesionarios de vehículos de diferente nombre y modelo (que pueden ser de distintas marcas), y que los venden a clientes.

Marca(CM, nombre_Marca, país, tipo)

// Datos de las marcas que fabrican los vehículos: código, nombre, país de origen y tipo de vehículos que fabrica (“familiar”, “deportivo”, “industrial”).

//Ej: (1, “SEAT”, “España”, “familiar”)

Vehículo(CV, nombre_Vehículo, modelo, potencia, precio, CM)

// Datos de cada vehículo: código, nombre general del vehículo, modelo concreto, potencia, precio de venta y marca que fabrica el vehículo

// Ej: (22, “1430”, “especial”, 70, 22000, 1)

Concesionario(CC, nombre_Concesionario, ciudad_Concesionario, país_Concesionario)

// Datos de los concesionarios que venden vehículos

// Ej: (333, “Autos Pepe”, Oviedo, España)

Cliente(DNI, nombre_Cliente, ciudad_Cliente, país_Cliente)

// Datos de los clientes

// Ej: (4444, “Deivid Jaseljof”, “Los Angeles”, “EE.UU.”)

Venta(CC, CV, DNI, color, descuento)

// Datos de cada venta: concesionario que vendió el vehículo, vehículo vendido, cliente al que se le vendió, color concreto de la unidad del vehículo y descuento aplicado en tanto por ciento.

// Ej: (333, 22, 4444, “amarillo”, 10)

Almacén(CC, CV, unidades)

// Datos de los vehículos que están en el almacén de cada concesionario: concesionario, vehículo almacenado y número de unidades que el concesionario tiene en el almacén

// Ej: (333, 22, 5)

Nota 1 : DEBEN usarse las abreviaturas:

Vehículo, veh

Concesionario , conc

Cliente , cli

Nombre, nom

Nota 2 : Supónganse los dominios acostumbrados para cada atributo

Nota 3 : Nombres alternativos para las tablas:

Marca, m

Vehículo, g

Concesionario, o

Cliente, c

Venta, v

Almacén, a

TEST

1. Dada una integridad referencial, con una relación referenciada r_1 con clave primaria K y una relación que referencia r_2 con clave externa a
 - (a) r_1 tiene que estar en forma normal BCNF al menos
 - (b) r_1 y r_2 deben ser relaciones compatibles para la diferencia
 - (c) r_2 debe ser una relación que provenga de una entidad débil del modelo E-R
 - (d) Ninguna de las otras es correcta**
 - (e) Siempre debe cumplirse $?_K(r_1) \subseteq ?_a(r_2)$

2. Dados un F y un recubrimiento canónico de F , F_c , siempre se cumple [Nota: \implies implica lógicamente, $\not\implies$ no implica lógicamente]
 - (a) $F^+ \not\implies F_c^+$
 - (b) Ninguna de las otras es correcta
 - (c) $F \implies F_c$
 - (d) $F \not\implies F_c$ y $F_c \implies F$
 - (e) $F \implies F_c$ y $F_c \implies F$**

3. Dado $R = (A, B, C, D)$ y $F = \{ B \twoheadrightarrow C, BC \twoheadrightarrow DA, A \twoheadrightarrow B \}$
 - (a) R NO está en BCNF ni en 3FN porque F no es un recubrimiento canónico
 - (b) Ninguna de las otras es correcta
 - (c) La forma normal más alta en que está R es 3NF
 - (d) R NO está en 3NF
 - (e) La forma normal más alta en que está R es BCNF**

4. En una descomposición de producto sin pérdida de R en R_1 y R_2 , con $r(R)$, $r_1(R_1)$, $r_2(R_2)$
 - (a) Es posible que R_1 y R_2 no tengan ningún atributo en común
 - (b) R_1 y R_2 deben tener como mucho un atributo en común
 - (c) Basta con que R_1 y R_2 tengan más de un atributo en común para que la descomposición sea de producto sin pérdida
 - (d) Ninguna de las otras es correcta
 - (e) $r_1 \times r_2$ NO puede tener más tuplas que r**

5. La conservación de dependencias en una descomposición es deseable porque
 - (a) La conservación reduce el grado de redundancia inicial
 - (b) Si no se conservasen, la BD estaría en un estado inconsistente
 - (c) Si no se conservasen, no se podría insertar nueva información en la BD
 - (d) Nos garantiza que todas las relaciones de la descomposición estén en BCNF
 - (e) Ninguna de las otras es correcta**

6. Dentro de los limitantes de integridad expresables en alguna versiones vistas del SQL
- (a) **Ninguna de las otras es correcta**
// Puesto que en ediciones anteriores del libro venía UNIQUE KEY, y no sólo UNIQUE, que es la forma estandarizada en SQL-2
 - (b) **Las claves candidato se indican con la cláusula UNIQUE**
 - (c) Las claves primarias se indican con la cláusula FOREIGN KEY
 - (d) Los disparadores se indican con la sentencia CREATE ASSERTION
 - (e) Las dependencias funcionales se indican con la cláusula FUNCTIONAL DEPENDENCY
7. Visto $F = \{ BD \twoheadrightarrow CAE, CA \twoheadrightarrow B, B \twoheadrightarrow E \}$, puede afirmarse
- (a) **E es un atributo ajeno en $BD \twoheadrightarrow CAE$**
 - (b) A es un atributo ajeno en $CA \twoheadrightarrow B$
 - (c) C es un atributo ajeno en $BD \twoheadrightarrow CAE$
 - (d) A es un atributo ajeno en $CA \twoheadrightarrow B$
 - (e) Ninguna de las otras es correcta
8. En SQL, una expresión equivalente a las que usan la operación de intersección en álgebra relacional
- (a) No puede expresarse en la práctica, pues CONTAINS no forma parte de ningún estándar de SQL
 - (b) Necesita utilizar obligatoriamente el operador MINUS (EXCEPT) al menos
 - (c) Tendría un tiempo de ejecución tan excesivo que no se puede usar en la práctica
 - (d) Se puede crear fácilmente usando el operador FORALL
 - (e) **Ninguna de las otras es correcta**
9. El cierre X^+ de un conjunto de atributos X contenido en R bajo un conjunto de dependencias F
- (a) Se calcula partiendo necesariamente de un recubrimiento canónico de F
 - (b) Puede tener menos atributos que X
 - (c) **Si X no tiene atributos primos, X^+ no puede tener atributos primos**
 - (d) Es igual a R si todos los atributos de X son primos
 - (e) Ninguna de las otras es correcta
10. En una integridad referencial con una relación referenciada r1 con clave primaria K y una relación que referencia r2 con clave externa alfa
- (a) **Una eliminación en r1 puede conllevar una eliminación en r2**
 - (b) Una eliminación en r2 puede conllevar una actualización en r1
 - (c) Una inserción en r1 puede conllevar una inserción en r2
 - (d) Una inserción en r2 puede conllevar una actualización en r1
 - (e) Ninguna de las otras es correcta

Bases de Datos**Segundo Parcial – Modelo X****PROPUESTA DE SOLUCIÓN DE EJERCICIOS****PRIMERA PREGUNTA (3,5 puntos)**

Para la base de datos del anexo, escribir una expresión en SQL para las siguientes consultas (convertirla a SQL básico en caso de usar características no estándar o de SQL92):

**a) Código, nombre y marca de los vehículos de los que sólo se ha realizado una única venta
Escribir también esta consulta en QBE**

Cada venta es una tupla en la relación venta, con un código de vehículo determinado. Una estrategia es buscar aquellos vehículos que tienen más de una tupla (es decir, como mínimo dos) y restarlos de los que tienen al menos una tupla.

En SQL, al tener operadores de agregados, también podemos agrupar las ventas por código de vehículo y contar simplemente los grupos con una sola tupla. A partir de la relación venta con productos naturales se obtiene fácilmente la marca, nombre, etc. Añadimos el nombre del vehículo y de la marca en la agrupación para poder sacarlos en el resultado (sólo pueden salir aquellos atributos o expresiones que toman siempre el mismo valor dentro del grupo: funciones de agregados y atributos que salen en la cláusula GROUP BY, en esencia).

```
SELECT v.CV, g.nom_veh, m.nom_marca
FROM marca AS m, vehiculo AS g, venta AS v
WHERE m.CM = y.CM AND g.CV = v.CV
GROUP BY v.CV, g.nom_veh, m.nom_marca
HAVING COUNT(*) = 1
```

En QBE se puede utilizar una estrategia similar a las 2 expuestas anteriormente.

b) Código y nombre de las marcas que han vendido al menos un vehículo en todos los concesionarios de su mismo país.

Conjunto de concesionarios donde ha vendido al menos un vehículo la marca t.CM

CONTAINS

Conjunto de concesionarios del mismo país que la marca t.CM (t.país_marca)

La lista de todas las posibles t.CM la podemos sacar de la propia relación marca

```
SELECT t.CM, t.nom_marca
FROM marca AS t
WHERE
( SELECT CC
  FROM vehiculo AS g, venta AS v
  WHERE g.CV = v.CV AND
        g.CM = t.CM
)
CONTAINS
( SELECT CC
  FROM concesionario AS c
  WHERE c.pais_conc = t.pais_marca
)
```

Para pasarlo a ANSI-86 -> NOT EXISTS (A EXCEPT B) y luego el EXCEPT lo expresamos con el NOT IN.

c) Código y nombre de los vehículos cuyo número de unidades almacenados en cada concesionario es superior a la media de unidades por vehículo almacenadas en el concesionario.

Dado un vehículo, tomamos las unidades que hay almacenadas de ese vehículo en un concesionario determinado t.CC (de la relación almacén). Si esas unidades son mayores que la media de unidades almacenadas en el concesionario t.CC, ese vehículo nos interesa.

Para saber la media de unidades almacenadas en el concesionario t.CC puede hacerse “a mano”, calculando el conjunto de tuplas de almacén de ese concesionario, y obteniendo el número acumulado de unidades de ese conjunto y dividiendo por el número total de tuplas de ese conjunto. También se puede simplemente aplicar el operador de media a las unidades en ese conjunto, puesto que únicamente saldrá una tupla en el resultado.

```
SELECT
FROM vehiculo AS g, almacen AS t
WHERE t.CV = g.CV AND
      t.unidades >
      ( SELECT AVG(unidades)
        FROM almacen AS abis
        WHERE abis.CC = t.CC
      )
```

SEGUNDA PREGUNTA (2,00 puntos)

Encontrar una descomposición de producto sin pérdida (PSP), que conserve las dependencias (CD), y normalizada en 3FN o BCNF lo mejor posible (mínimo número de esquemas y de redundancia, etc.) del esquema de relaciones R = (A, B, C, D, E, F, G, H, I, J, K) y el conjunto de dependencias F:

$$F = \left\{ \begin{array}{l} E \rightarrow JB \\ G \rightarrow EK \\ BJ \rightarrow CG \\ D \rightarrow AFI \end{array} \right\}$$

Indicar la forma normal de cada relación y mostrar que la descomposición encontrada efectivamente es de producto sin pérdida y conserva las dependencias.

H estará en cualquier clave, ya que no aparece en F. Puede comprobarse que hay 3 claves candidato:

GDH
JDBH
EDH

Que en esencia se derivan de las dependencias $E \rightarrow JB$, $JB \rightarrow G$, $G \rightarrow E$.

Obsérvese que de esas dependencias se deriva $E \rightarrow G$, $JB \rightarrow E$ y $G \rightarrow JB$.

Para normalizar, inicialmente se comprueba que F sea un recubrimiento canónico (como es el caso), y cada dependencia genera una relación. Si hay dependencias repetidas en las relaciones generadas se intentan eliminar de todas ellas menos en una, y si no hay ninguna clave candidato del esquema original se añade una clave candidato a la descomposición (como en este caso). Se calculan las restricciones de cada R_i y se dice la forma normal en la que está cada R_i (con este algoritmo 3FN o BCNF):

$R_1 = (E, J, B)$	$F_1 = \{ E \rightarrow JB, \mathbf{JB \rightarrow E} \}$	Claves E y JB, por tanto BCNF
$R_2 = (G, E, K)$	$F_2 = \{ G \rightarrow EK, \mathbf{E \rightarrow G} \}$	Claves G y E, por tanto BCNF
$R_3 = (B, J, C, G)$	$F_3 = \{ JB \rightarrow CG, \mathbf{G \rightarrow JB} \}$	Claves JB y G, por tanto BCNF
$R_4 = (D, A, F, I)$	$F_4 = \{ D \rightarrow AFI \}$	Clave D, por tanto BCNF
$R_5 = (G, D, H)$	$F_5 = \{ \}$	Clave GDH, por tanto BCNF

(R5 podría ser cualquier otra clave candidato del esquema original)