

EXAMEN DE BASES DE DATOS – 11/06/04 – Esquema de trabajo

Cuando haya terminado el cuestionario traslade CUIDADOSAMENTE sus respuestas a la siguiente tabla, de este modo al terminar el examen podrá calcular el resultado que ha obtenido con la fórmula $(\text{Bien_contestadas} - \text{Mal_contestadas}/3) \times 0,25$.

1	2	3	4	5	6	7	8	9	10	11	12	13	14

Sea el siguiente esquema relacional al que se hará referencia como ESQUEMA de TRABAJO que mantiene información sobre casas rurales:

- Usuario**(dni: d1, nombre: d2, edad: d3, cod_pob: d4)
 - CP: {dni} VNN: {nombre, cod_pob}
- Población**(cod_pob: d4, nombre: d2, habitantes: d8)
 - CP: {cod_pob} VNN: {nombre}
- Casa_rural**(cod_cas: d5, habitaciones: d3, precio: d6, cod_pob:d4, nota_media: d7)
 - CP: {cod_cas} VNN: {habitaciones, precio, cod_pob}
 - CAj: {cod_pob} → Población
 - Borrado RESTRICTIVO y Actualización en CASCADA
- Ha_estado**(cod_cas: d5, dni: d1, veces: d10)
 - CP: {cod_cas, dni} VNN: {veces}
 - CAj: {cod_cas} → Casa_rural
 - Borrado RESTRICTIVO y Actualización en CASCADA
 - CAj: {dni} → Usuario
 - Borrado CASCADA y Actualización en CASCADA
- Opinión**(num: d9, nota: d7, cod_cas: d5, dni: d1)
 - CP:{num} VNN: {cod_cas, nota}
 - CAj: {cod_cas, dni} → Ha_estado
 - Integridad Referencial PARCIAL
 - Borrado en CASCADA y Actualización en CASCADA

Donde los distintos atributos tienen el siguiente significado

Usuario:

- dni*: número del documento nacional de identidad del usuario
- nombre*: nombre del usuario
- edad*: años que tiene el usuario
- cod_pob*: código de la población donde reside el usuario

Población:

- cod_pob*: código de la población
- nombre*: nombre de la población
- habitantes*: cantidad de habitantes de la población

Casa_rural:

cod_cas: código identificador de la casa

habitaciones: número de habitaciones de que dispone la casa

precio: euros por día y persona

cod_pob: código de la población donde está la casa rural

nota_media: nota media obtenida por la casa en las opiniones de los usuarios

Ha_estado: El usuario de dni *dni* ha estado en la casa rural de código *cod_cas* en *veces* ocasiones

Opinión: Hay una opinión con identificador *num* sobre la casa rural de código *cod_cas* que valora la casa en *nota* puntos realizada por el usuario de dni *dni*.

Y sea la siguiente extensión del esquema anterior a la que se hará referencia como BD:

Usuario			
Dni	Nombre	edad	cod_pob
1	Luisa	48	44
2	María	21	45
3	Juan	32	45

Población		
Cod_pob	Nombre	Habitantes
44	Teruel	32.000
45	Toledo	68.000
16	Cuenca	46.000
10	Cáceres	82.000

Casa_rural				
Cod_cas	Habitaciones	Precio	Cod_pob	Nota_media
c1	5	180	16	6
c2	4	100	16	10
c3	2	60	45	8
c4	8	250	10	

Ha_estado		
Cod_cas	dni	Veces
c1	1	1
c1	2	1
c2	1	4
c2	2	4
c3	1	2
c3	3	1
c4	1	1

Opinión			
Num	Nota	Cod_cas	Dni
1	6	c1	1
2	7	c1	
3	5	c1	
4	10	c2	1
5	10	c2	2
6	8	c3	

Resolver las cuestiones que aparecen a continuación.

EXAMEN DE BASES DE DATOS – 11/06/04 – Cuestionario Tipo A

1. Si se añade al esquema de trabajo la siguiente definición en SQL estándar:
`CREATE VIEW Buenos_clientes AS SELECT dni, nombre FROM Usuario U WHERE dni IN (SELECT dni FROM Opinión) AND NOT EXISTS (SELECT * FROM Opinión O WHERE O.dni = U.dni AND O.nota<5)`, ¿cuál de las siguientes afirmaciones es CIERTA?
 - a) La definición es correcta pero no se puede añadir porque el esquema ya está definido.
 - b) No es correcta la definición porque no se pueden definir vistas con subconsultas.
 - c) Se puede consultar como cualquiera de las tablas del esquema.
 - d) Se puede consultar sólo si se le añade la cláusula `WITH CHECK OPTION`.

2. Según la definición de la vista *Buenos_clientes* de la cuestión anterior, la consulta `SELECT dni, nombre FROM Buenos_clientes B WHERE (SELECT COUNT(cod_cas) FROM Ha_Estado H WHERE H.dni=B.dni) >10`, transformada al esquema lógico es:
 - a) `SELECT dni, nombre FROM Usuario U WHERE (SELECT COUNT(cod_cas) FROM Ha_Estado H WHERE H.dni=U.dni) >10`
 - b) `SELECT dni, nombre FROM Buenos_clientes B WHERE dni IN (SELECT dni FROM Opinión) AND NOT EXISTS (SELECT * FROM Opinión O WHERE O.dni = U.dni AND O.nota<5) AND (SELECT COUNT(cod_cas) FROM Ha_Estado H WHERE H.dni=B.dni) >10`
 - c) `SELECT dni, nombre FROM Buenos_clientes B WHERE (SELECT COUNT(cod_cas) FROM Ha_Estado H WHERE H.dni=B.dni) >10`
 - d) `SELECT dni, nombre FROM Usuario U WHERE dni IN (SELECT dni FROM Opinión) AND NOT EXISTS (SELECT * FROM Opinión O WHERE O.dni = U.dni AND O.nota<5) AND (SELECT COUNT(cod_cas) FROM Ha_Estado H WHERE H.dni=U.dni) >10`

3. La instrucción de SQL estándar para que la tupla de la tabla *Opinión* de *num=5* de la base de datos BD pase a tener el *dni* nulo es:
 - a) `DELETE dni FROM Opinión WHERE num=5.`
 - b) `UPDATE Opinión SET dni=NULL WHERE num=5.`
 - c) `UPDATE Opinión SET dni=NULL, num=5.`
 - d) `UPDATE Opinión SET dni=NULL WHERE dni=5.`

4. Sobre el esquema de trabajo, ¿cuál de las siguientes afirmaciones es FALSA?:
 - a) No puede haber opiniones sobre casas rurales en las que no haya estado nadie.
 - b) Para toda casa rural tiene que saberse la población en la que está.
 - c) Puede haber casas rurales en las que no haya estado ningún usuario.
 - d) Una opinión sobre una casa rural siempre es realizada por alguno de los usuarios que han estado en la casa.

5. Según el esquema de la base de datos, ¿cuál de las siguientes expresiones en Álgebra Relacional, responde a la pregunta: qué poblaciones tienen más de una casa?

- a) $(\text{Casa_rural}[\text{cod_cas}, \text{cod_pob}](\text{cod_cas}, \text{CC1}) \bowtie \text{Casa_rural}[\text{cod_cas}, \text{cod_pob}](\text{cod_cas}, \text{CC2})) \text{ DONDE } \text{CC1} \neq \text{CC2} [\text{cod_pob}]$
- b) $(\text{Casa_rural}[\text{cod_cas}, \text{cod_pob}] \bowtie \text{Casa_rural}[\text{cod_cas}, \text{cod_pob}]) [\text{cod_pob}]$
- c) $(\text{Casa_rural}[\text{cod_cas}, \text{cod_pob}](\text{cod_cas}, \text{CC1}) \bowtie \text{Casa_rural}[\text{cod_cas}, \text{cod_pob}](\text{cod_cas}, \text{CC2})) \text{ DONDE } \text{CC1} = \text{CC2} [\text{cod_pob}]$
- d) $(\text{Casa_rural}[\text{cod_cas}, \text{cod_pob}] - \text{Casa_rural}[\text{cod_cas}, \text{cod_pob}]) [\text{cod_pob}]$

6. Según el esquema de la base de datos, ¿qué información devuelve la siguiente expresión de Álgebra Relacional?:

$\text{Casa_rural}[\text{cod_cas}] - (\text{Ha_estado} \bowtie \text{Usuario})[\text{cod_cas}]$

- a) Casas rurales en las que ha estado exactamente un usuario.
- b) Casas rurales en las que no ha estado ningún usuario.
- c) Casas rurales en las que han estado más de un usuario.
- d) Casas rurales en las que han estado exactamente dos usuarios.

7. ¿Cuál sería la consecuencia de ejecutar la siguiente instrucción en SQL sobre el Esquema de Trabajo?:

```
CREATE ASSERTION R1
CHECK (NOT EXISTS (SELECT *
                    FROM Usuario U
                    WHERE EXISTS (SELECT *
                                   FROM Casa_rural C, Ha_estado H
                                   WHERE C.cod_cas = H.cod_cas
                                         AND C.cod_pob = U.cod_pob
                                         AND U.dni = H.dni
                                         AND H.veces > 2)));
```

- a) Se añadiría una restricción para impedir que un usuario esté más de dos veces en una casa rural.
- b) Se crearía una relación derivada con toda la información de los usuarios que han estado más de dos veces en una casa rural.
- c) Se añadiría una restricción para obligar a que todos los usuarios hayan estado al menos dos veces en alguna casa rural de su población.
- d) Se añadiría una restricción para impedir que un usuario esté tres o más veces en una casa rural de la población del usuario.

8. Considerando el Esquema de Trabajo, ¿Cuál de las siguientes afirmaciones es CIERTA?
- Al borrar una casa rural se borraría también todas las demás casas rurales de la misma población.
 - No se puede borrar una casa rural si existe otra casa rural en la misma población.
 - Solamente se puede borrar una población si en ella no hay casas rurales.
 - Todas las poblaciones tienen al menos una casa rural.
9. ¿Cuál de las siguientes afirmaciones sobre la Clave Ajena de *Opinión* es FALSA?:
- Si se define una restricción de valor no nulo (VNN) sobre el atributo *dni* no sería necesario indicar el tipo de Integridad Referencial.
 - Permite recoger opiniones sobre casas rurales sin conocer al usuario que realiza dichas puntuaciones.
 - Sería equivalente a dos Claves Ajenas de la forma:
CAj:{cod_cas} ? Casa_rural
Borrado en CASCADA y Actualización en CASCADA
CAj:{dni} ? Usuario
Borrado en CASCADA y Actualización en CASCADA
 - Una modificación del atributo *dni* en *Usuario*, de un usuario que ha emitido una opinión sobre alguna casa rural, se permitiría si no existe ya otro usuario con el nuevo *dni*.
10. ¿Cuál sería el estado final de las tablas *Ha_estado* y *Opinión* después de ejecutar la instrucción DELETE FROM *Ha_estado* WHERE *dni* <> 2 sobre la base de datos BD?

a)

Ha_estado		
Cod_cas	dni	Veces
c1	2	1
c2	2	4

Opinión			
Num	Nota	Cod_cas	Dni
2	7	c1	
3	5	c1	
5	10	c2	2

Las demás relaciones quedarían igual.

b)

Ha_estado		
Cod_cas	dni	Veces
c1	1	1
c2	1	4
c3	1	2
c3	3	1
c4	1	1

Opinión			
Num	Nota	Cod_cas	Dni
1	6	c1	1
2	7	c1	
3	5	c1	
4	10	c2	1
6	8	c3	

Las demás relaciones quedarían igual.

c)

Ha_estado		
Cod_cas	dni	Veces
c1	2	1
c2	2	4

Opinión			
Num	Nota	Cod_cas	Dni
5	10	c2	2

Las demás relaciones quedarían igual.

d)

Ha_estado		
Cod_cas	dni	Veces
c1	1	1
c1	2	1
c2	1	4
c2	2	4
c3	1	2
c3	3	1
c4	1	1

Opinión			
Num	Nota	Cod_cas	Dni
1	6	c1	1
2	7	c1	
3	5	c1	
4	10	c2	1
5	10	c2	2
6	8	c3	

Las demás relaciones quedarían igual.

11. Si se modifica la definición de la relación *Opinión* añadiendo un atributo “*validez*”: ¿cuál de las siguientes afirmaciones es FALSA?

- a) Si la ligadura se realiza en tiempo de compilación, será necesario recompilar las aplicaciones que usen la tabla *Opinión*.
- b) No será necesario modificar las vistas que usen la tabla *Opinión*.
- c) Si la ligadura se realiza en tiempo de ejecución, no será necesario recompilar las aplicaciones que usen la tabla *Opinión*.
- d) La independencia física garantiza que no cambiará el esquema físico.

12. Sea la transacción T1 que se ejecuta sobre la base de datos BD:

TRANSACCIÓN T1

```
INSERT INTO Opinión VALUES (10,8,'c3',2);
INSERT INTO Ha_estado VALUES ('c3',2,2);
COMMIT WORK;
```

FIN

¿Cuál de las siguientes afirmaciones es CIERTA?

- a) T1 no fallará porque el procesamiento de las transacciones cumple la propiedad de atomicidad.
- b) T1 no fallará si la integridad referencial de *Opinión* se define como INITIALLY IMMEDIATE.
- c) T1 no fallará si la integridad referencial de *Opinión* se ha definido como INITIALLY DEFERRED.
- d) T1 no fallará si incluimos la cláusula WITH NO CHECK OPTION en la definición de la transacción.

13. Considerando la transacción T1 definida anteriormente, y que U1, U2 y U3 son identificadores de usuarios del SGBD. Si ejecutamos las sentencias SQL

GRANT INSERT ON *Opinión* TO U1

GRANT INSERT ON *Ha_estado* TO U2 WITH GRANT OPTION

¿Cuál de las siguientes afirmaciones es CIERTA?

- a) U1 podrá ejecutar la transacción T1 siempre.
- b) U2 podrá ejecutar T1 si recibe permiso de U1 para insertar en *Opinión*.
- c) U1 podrá ejecutar T1 si recibe permiso de U2 para insertar en *Ha_estado*.
- d) U3 podrá ejecutar T1 si recibe permiso de U1 y U2 para insertar en *Opinión* y *Ha_estado* respectivamente.

14. Si durante la ejecución de T1 se produce una pérdida de memoria secundaria que no afecte al fichero diario ¿Cuál de las siguientes afirmaciones es CIERTA?

- a) Será necesario recuperar la copia de seguridad más reciente y repetir todas las transacciones desde esa fecha
- b) Será necesario recuperar la copia de seguridad más reciente de la base de datos y repetir todas las transacciones confirmadas desde esa fecha
- c) Desharemos la transacción T1 y todas las que dependan de ella
- d) Desharemos T1 y el resto de transacciones no confirmadas

EXAMEN DE BASES DE DATOS – 11/06/04 – Problemas

Dado el ESQUEMA de TRABAJO presentado anteriormente, resuelva los siguientes ejercicios en SQL estándar:

1. Obtener el código y el nombre de las poblaciones que no tienen ninguna casa rural con nota media inferior a 5. (0,5 puntos)
2. Obtener el dni y el nombre de los usuarios que han opinado sobre alguna casa rural de más de tres habitaciones dando una nota mayor de 5. (0,5 puntos)
3. Obtener el dni y el nombre de los usuarios que más veces han estado en una casa rural. (0,5 puntos)
4. Obtener, para todas las poblaciones que hay en la base de datos, el código y el nombre de la población, y cuántas casas rurales tiene. (0,75 puntos)
5. Suponiendo que hay opiniones sobre todas las casas rurales, obtener el código y el número de habitaciones de las casas rurales que sólo tienen opiniones con nota superior o igual a 5. (0,75 puntos)
6. Obtener el dni y el nombre de los usuarios que han estado en todas las casas rurales de alguna población. (1 punto)
7. Obtener el código y el nombre de las poblaciones que más casas rurales tienen. (1 punto)
8. El atributo *nota_media* de la relación *Casa_rural* es un atributo derivado que se obtiene al calcular el valor medio, para cada casa rural, de las notas que se le han asignado en las distintas opiniones, es decir:

$$Nota_media_casa_i = \frac{\sum notas_asignadas_casa_i}{número_opiniones_casa_i}$$

- a. Además de la inserción en la relación *Opinión*, enumere las operaciones sobre la base de datos que afectan al valor del atributo derivado. (0,75 puntos)
- b. Diseñe un disparador (*trigger*) de SQL estándar para controlar la operación de inserción en *Opinión*. (0,75 puntos)

SOLUCIONES DEL TEST:

	Respuesta Tipo A/E	Respuesta Tipo B/F	Respuesta Tipo C/G	Respuesta Tipo D/H
1	C	D	A	B
2	D	A	B	C
3	B	C	D	A
4	D	A	B	C
5	A	B	C	D
6	B	C	D	A
7	D	A	B	C
8	C	D	A	B
9	C	D	A	B
10	A	B	C	D
11	D	A	B	C
12	C	D	A	B
13	C	D	A	B
14	B	C	D	A

SOLUCIONES DEL LOS PROBLEMAS:

1. SELECT cod_pob, nombre FROM Población P
WHERE cod_pob NOT IN
(SELECT cod_pob FROM Casa_rural C WHERE nota_media <5);
2. SELECT dni, nombre FROM Usuario U
WHERE dni IN (SELECT O.dni FROM Opinión O, Casa_rural C
WHERE O.cod_cas=C.cod_cas AND C.habitaciones>3 AND O.nota>5);
3. SELECT dni, nombre FROM Usuario U
WHERE dni IN (SELECT dni FROM Ha_estado
WHERE veces = (SELECT MAX(veces) FROM Ha_estado));
4. SELECT P.cod_pob, P.nombre, COUNT(cod_cas)
FROM Población P LEFT JOIN Casa_rural C ON P.cod_pob= C.cod_pob
GROUP BY P.cod_pob, P.nombre;
5. SELECT cod_cas, habitaciones FROM Casa_rural C
WHERE NOT EXISTS (SELECT * FROM Opinión O
WHERE C.cod_cas=O.cod_cas AND nota<5));
6. SELECT dni, nombre FROM Usuario U
WHERE EXISTS (SELECT * FROM Población P
WHERE NOT EXISTS (SELECT * FROM Casa_rural C
WHERE C.cod_pob=P.cod_pob AND
NOT EXISTS (SELECT * FROM Ha_estado H
WHERE H.dni=U.dni AND
H.cod_cas=C.cod_cas)))
AND EXISTS (SELECT * FROM Casa_Rural C1 WHERE P.cod_pob =
C1.cod_pob);
7. SELECT cod_pob, nombre FROM Población P
WHERE cod_pob IN (SELECT cod_pob FROM Casa_rural C
GROUP BY cod_pob
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
FROM Casa_rural
GROUP BY cod_pob));

o bien
SELECT cod_pob, nombre FROM Población P
WHERE cod_pob IN (SELECT cod_pob FROM Casa_rural C
GROUP BY cod_pob
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
FROM Casa_rural
GROUP BY cod_pob));

8. a) Las operaciones que hay que controlar son:
- Insertar en *Opinión* → recalcular la *nota_media*
 - Borrar en *Opinión* → recalcular la *nota_media*
 - Modificar *nota* en *Opinión* → recalcular la *nota_media*
 - Modificar *cod_cas* en *Opinión* → recalcular la *nota_media*
 - Insertar en *Casa_rural* → el valor de *nota_media* debe ser nulo
 - Modificar *nota_media* en *Casa_rural* → prohibido

b) En SQL estándar:

```
CREATE TRIGGER Inserción_Opinión
AFTER INSERT ON Opinión
REFERENCING NEW ROW AS nueva
FOR EACH ROW
BEGIN ATOMIC
    DECLARE Media AS INTEGER;
    SELECT AVG(nota) INTO Media
    FROM Opinión
    WHERE cod_cas=nueva.cod_cas;
    UPDATE Casa_rural SET nota_media= media
    WHERE cod_cas=nueva.cod_cas;
END
```