Maxigang
www.wuolah.com/student/Maxigang

FBDFinalJunio2014resuelto.pdf

Exámenes teóricos resueltos

2º Fundamentos de Bases de Datos

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Fundamentos de Bases de Datos. Julio 2014
Teoría

Nombre Alumno:

Profesor del Grupo:

Parcial/es a los que se presenta:

Parcial 1

Marca con V o F las siguientes afirmaciones. **Dos errores eliminan un acierto**.

1. El principal objetivo de evitar la redundancia en una BD es ahorrar espacio en disco (F)
2. Cuando se diseña una BD es fundamental conocer las características técnicas del servidor sobre el que se va a implantar (F)
3. El término integridad hace referencia a la veracidad de los datos que se almacenan, esto es, a su correspondencia con la realidad (V)
4. La independencia física permite reorganizar las estructuras del nivel interno sin que se vean afectados los programas de aplicaciones (V)
5. En el nivel externo se plasma la perspectiva que tiene cada usuario de la BD (V)
6. UPDATE … es un comando del DDL (F)
7. Cuando se pasa un diagrama E/R a tablas, las claves candidatas no se tienen en cuenta (F)
8. La independencia física es posible gracias a la correspondencia externa/conceptual (F)
9. Gracias a la transformación conceptual/interna se puede mantener la independencia física (V)
10. La elaboración del esquema conceptual es tarea del programador de aplicaciones (F)
11. Una entidad que no tiene clave primaria es siempre una entidad débil (V)
12. La forma de implantar la cardinalidad de una relación de un diagrama E/R en una tabla, es mediante la correcta elección de las claves candidatas y primarias (V)
13. Una relación de cardinalidad muchos-muchos siempre genera una tabla con clave primaria compuesta (V)
14. Una relación de cardinalidad muchos-uno puede generar una tabla con clave primaria compuesta (V)
15. CREATE TABLE es un comando del DDL (V)
16. En una jerarquía, todas las entidades del conjunto de entidades genérico deben estar en un conjunto de entidades específico (F)
17. En una jerarquía, todas las entidades de un conjunto específico deben estar en el conjunto de entidades genérico (V)
18. Todas las tablas procedentes de entidades débiles tienen claves externas (V)
19. Un atributo no puede ser clave primaria y externa a la vez (F)
20. La diferencia entre una clave candidata y una primaria es que la candidata no tiene por qué ser minimal (F)
21. Una clave externa puede tomar el valor nulo (V)
22. Una clave primaria puede tomar el valor nulo parcialmente (F)
23. La regla de integridad de entidad exige que no existan tuplas duplicadas en una relación (F)
24. La dependencia existencial sólo se da entre entidades débiles y fuertes (F)
25. Todas las restricciones de integridad se pueden mantener eligiendo convenientemente claves candidatas, primarias y externas (F)

Parcial 2

Marca con V o F en el margen las siguientes afirmaciones. **Dos errores eliminan un acierto**.

1. Las páginas que componen un archivo almacenado no tienen por qué estar consecutivas en disco (V)
2. El índice no denso mejora el barrido ordenado completo del fichero por la clave física (F)
3. El acceso directo a registros no permite realizar la lectura secuencial de datos en un rango(V)
4. El hashing dinámico es muy eficaz porque la tabla hash va en memoria principal (F)
5. Se pueden montar tantos índices densos como se necesiten (V)
6. En un índice multinivel el índice de primer nivel (nodo hoja) puede ser denso o no denso (V)
7. El acceso directo a registros garantiza siempre que encuentro una tupla con una sóla lectura de bloque (F)
8. El acceso directo a bloques o cubos produce menos lecturas en disco que el acceso directo a registros (V)
9. El índice denso es adecuado para consultas por rangos de valores del campo clave (V)
10. Los bloques usados para almacenar los datos de la BD pueden ser de distinto tamaño dependiendo del tamaño de los registros que se almacenen en ellos (F)
11. El índice no denso permite realizar preguntas de tipo existencial sin acceder al fichero de datos (F)
12. El hashing dinámico es el método de acceso que mejor distribuye los datos en disco y, por tanto, el que menos desperdicio ocasiona (V)
13. Se pueden montar tantos índices no densos como sea necesario (F)
14. Para montar un índice denso, los registros tienen que estar ordenados físicamente por algún campo (F)
15. El índice denso ocupa el mismo tamaño que el propio fichero que indexa (F)
16. En ficheros no ordenados físicamente, no se pueden montar índices no densos (V)
17. El índice no denso es mucho menor que el denso cuando caben varios registros en un bloque (V)
18. Las sentencias CREATE TABLE y CREATE INDEX de SQL generan nuevos conjuntos de páginas (archivos almacenados) en el nivel interno (V)
19. El rendimiento de un índice no denso desciende considerablemente cuando se realizan inserciones o borrados (F)
20. En acceso directo a registros, si se produce una colisión, habrá un hueco en el fichero maestro que nunca se va a aprovechar. (V)
21. El índice denso no es rentable cuando se actualiza o se inserta con mucha frecuencia (V)
22. El índice no denso es el único mecanismo de indexación posible cuando los datos están ordenados físicamente (F)
23. En hashing dinámico hace falta una estimación del número de datos a insertar para dimensionar la tabla hash (V)
24. El gestor de disco forma parte del SGBD (F)
25. Lo normal es que cada archivo almacenado del nivel interno se almacene en un fichero físico separado (F)

Tiempo total de realización: ?.

Fundamentos de Bases de Datos. Julio 2014
Ejercicio Práctico

Nombre Alumno:

Profesor del Grupo:

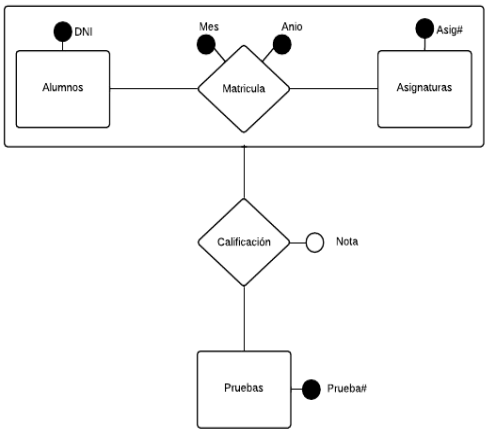
Parcial/es a los que se presenta:

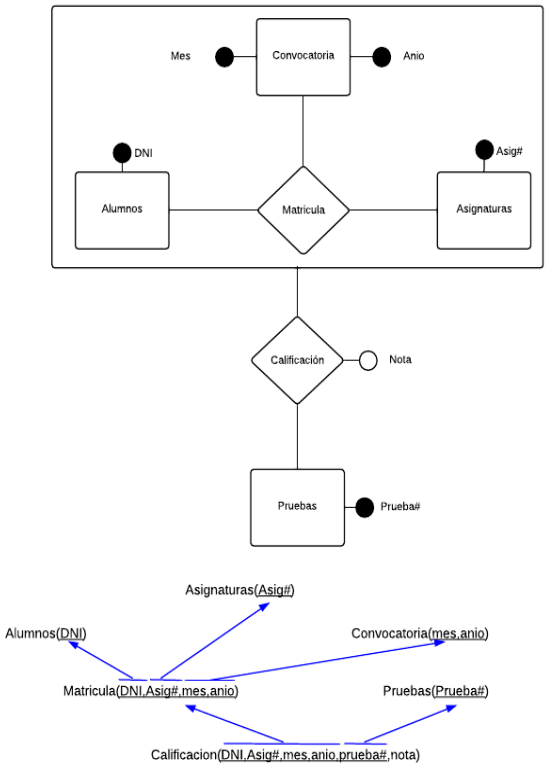
Parcial 1

1. Queremos gestionar las calificaciones obtenidas por cada alumno para las diversas pruebas programadas para cada asignatura y convocatoria, conforme a la siguiente especificación del problema:
    a. Tenemos tres convocatorias por año: junio, septiembre y diciembre. Cada convocatoria se identifica mediante el nombre y año de la convocatoria, p.e. (junio,2014).
    b. Cada matrícula está identificada por el alumno, la asignatura y la convocatoria.
    c. Existen diferentes tipos de pruebas para evaluar el rendimiento del alumno en cada asignatura: examen teórico, examen práctico, asistencia a clase, entregas de problemas, etc.
    d. Cada una de estas pruebas se evalúa mediante un valor numérico entre 0 y 10.
    e. Tenemos que poder registrar la calificación que ha obtenido cada alumno en cada convocatoria de cada asignatura para cada prueba evaluada, es decir, por cada matrícula y para cada prueba.
    f. Para cada alumno, convocatoria, asignatura y prueba sólo podemos tener una calificación.

Se pide:

a. Dibujar el esquema Entidad/Relación que represente adecuadamente dicha información. (**6 pts**.)
b. Elaborar el esquema relacional a que dá lugar indicando las claves candidatas y externas correspondientes. (**4 pts.**)





Parcial 2

Consideremos el siguiente esquema de base de datos:

HOTELES(Hot#,Nombre-h, Dirección, categoría)
HABITACIONES(Hot#,num-hab,tipo,n-camas,precio,disponible)
CLIENTE(DNI, Nombre-cl, Direccion-cl, Localidad, tipo-tarjeta,numero-tarjeta)
ACTIVIDAD(Act#,tipo-act, descripción)
OFERTA-ACTIVIDAD(Act#,Hot#, nmax-personas, fecha-inicio,fecha-final)
RESERVA(DNI,Hot#,num-hab,fecha-llegada,fecha-salida)
PIDE-ACTIVIDAD(DNI,Act#,Hot#,fecha,num-personas)

Aclaraciones semánticas:

- La categoría de un hotel es numérica en indica 1, 2 3 .. estrellas
- Disponible en HABITACION es un campo de que toma los valores 'si' o 'no' y que nos dice si una habitación se puede asignar a un cliente en este momento.
- El tipo de actividad puede ser: INFANTIL, ADULTO ó TODOS, la descripción de una actividad es su nombre, por ejemplo BUCEO, EQUITACIÓN etc.
- Las fecha-inicio y fecha-final de la tabla OFERTA-ACTIVIDAD nos dice entre qué fechas un hotel oferta una actividad determinada, por ejemplo BUCEO desde 1-07-2014 hasta el 31-08-2014
- Se asume que para que un cliente pueda pedir una actividad en un hotel debe tener una reserva que incluya la fecha pedida y el hotel debe ofertar la actividad en dicha fecha.

Las relaciones de llave exterior son las siguientes:
- HABITACIONES.Hot# llave exterior a HOTELES.Hot#
- RESERVA.(Hot#,num-hab) llave exterior a HABITACIONES(Hot#,num-hab)
- OFERTA-ACTIVIDAD.Hot# llave exterior a HOTELES.Hot#
- OFERTA-ACTIVIDAD.Act# llave exterior a ACTIVIDAD.Act#
- PIDE-ACTIVIDAD.(Act#,Hot#) llave exterior a OFERTA-ACTIVIDAD.(Act#,Hot#)
- RESERVA.DNI llave exterior a CLIENTE.DNI
- PIDE-ACTIVIDAD. DNI llave exterior a CLIENTE.DNI

1) Resolver en **Cálculo Relacional** la consulta:

Encontrar los clientes que han reservado en el hotel de nombre 'Estrella de Mar', pero que no han solicitado ninguna actividad. (1.5)

```
range R in reserva
range H in hotel
range PA in pide_actividad

select R.dni
where (exists H) H.nombre-h='Estrella de Mar' and
(exist R)  R.hot#=H.hot# and
not (exists PA) (pide-actividad(PA) and PA.dni=R.dni))}
```

2) Resolver en **Álgebra Relacional** la consulta**:**

Encontrar los DNI de los clientes que han pedido todas actividades de tipo infantil que se ofertan para al menos 10 personas. (1.5)

$$\pi_{dni,act\#}(pide-actividad) \div \pi_{act\#}(\sigma_{nmax>=10} oferta-actividad\,[?][?][?]\sigma_{tipo=INFANTIL}(actividad))$$

El operador que hay entre oferta-actividad y actividad es un reunion natural

3) **Resolver en SQL** los siguientes enunciados:

a) El hotel 'Estrella de Mar' quiere saber la información de todas sus habitaciones que estén disponibles. Se pide

a.1) Crear una vista de usuario para este requerimiento. (1.5)

```
CREATE VIEW Disponibles_estrella AS
SELECT ho.nombre_h,ha.hot#,num-hab,tipo,n_camas,precio,disponible
FROM hoteles ho, habitaciones ha
WHERE ho.nombre_h='Estrella de Mar' AND ho.hot#=ha.hot# AND
ha.disponible='si'
```

a.2) Usando la vista anterior, escribe la sentencia para actualizar a 'disponibles' las habitaciones que hayan quedado libres cada día. ¿Sería posible dicha actualización? Razona la respuesta. (1.5)

```
UPDATE Disponibles_estrella SET disponible='si'
WHERE hot#,num_hab IN
(SELECT r.hot#,r.num_hab
FROM hoteles ho, reservas r
WHERE ho.nombre_h='Estrella de Mar' AND ho.hot#=r.hot# AND
r.fecha_llegada>sysdate OR r.fecha_salida<sysdate
)
```

b) El hotel 'Estrella de Mar' quiere saber para cada día la cantidad de personas apuntadas a cada actividad durante el mes de Agosto del 2014. Dar la consulta que resuelve este requerimiento. (2)

```
SELECT p.fecha,p.act#,a.descripcion,sum(p.num_personas)
FROM hoteles h, pide_actividad p, actividad a
WHERE h.nombre_h='Estrella de Mar' AND h.hot#=p.hot# AND p.act#=a.act# AND
p.fecha >='01/08/2014' AND p.fecha <= '31/08/2014'
GROUP BY p.fecha,p.act#,a.descripcion
ORDER BY p.fecha,p.act#;
```

c) Encontrar los datos de los clientes de Toledo que han reservado todos los hoteles de cinco estrellas (2) *es una división, esta es una forma

```
SELECT * FROM Clientes c WHERE c.localidad='Toledo'
AND NOT EXISTS( SELECT * FROM Hoteles h WHERE h.categoria='5 Estrellas' AND
NOT EXISTS(SELECT * FROM Reserva r WHERE r.hot#=h.hot# AND r.DNI=c.DNI)
);
```

```
SELECT * FROM Clientes c
WHERE c.localidad='Toledo'
AND NOT EXISTS (
    ( SELECT Hot#  FROM Hoteles h WHERE h.categoria=5)
    minus
    (SELECT Hot# FROM Reserva r WHERE r.hot#=h.hot# AND r.DNI=c.DNI)
);
```