FICHEROS Y BASES DE DATOS 2º ITIG 10/9/2003

1. ¿Qué tienen en común los índices secundarios y los árboles B+?

Que ambos son índices.

2. ¿Es aconsejable definir un índice sobre todos y cada uno de los campos de los registros de un fichero?

Depende, ya que las actualizaciones de los datos requerirán más tiempo.

3. ¿Cuál de las siguientes afirmaciones es cierta? / ¿En qué condiciones será posible eliminar un equipo de la base de datos?

Sólo se pueden borrar equipos cuyos ciclistas no han participado en ninguna etapa.

4. Dadas las expresiones E1 y E2:

```
E1: PUESTO[dorsal]
    MINUS
    (PUESTO WHERE posicion>5)[dorsal]
```

```
E2: (PUESTO WHERE posicion<=5)[dorsal]
```

La expresión E1 obtiene sólo un subconjunto de los ciclistas que obtiene la expresión E2.

5. Tras finalizar la vuelta ciclista, se desea obtener los datos de los ciclistas que han ganado todos los puertos de una misma etapa. Dadas las siguientes expresiones:

```
E1: CICLISTAX WHERE ∃PUERTOX ∀ETAPAX
          (IF PUERTOX.numetapa = ETAPAX.numetapa
           THEN CICLX.dorsal = PUERTOX.dorsal)
```

```
E2: CICLISTAX WHERE ∃PUERTOX NOT ∃ETAPAX
          (PUERTOX.numetapa = ETAPAX.numetapa
           AND CICLX.dorsal <> PUERTOX.dorsal)
```

¿Cuál de las siguientes afirmaciones es cierta?

Ambas expresiones obtienen el resultado deseado.

6. Escribir una expresión del álgebra relacional que obtenga los datos del ciclista/s que ha quedado más veces entre las diez primeras posiciones.

```
T1 := SUMMARIZE (PUESTO WHERE posición <= 10) GROUPBY(dorsal)
      ADD COUNT(*) AS veces

T2 := SUMMARIZE T1 GROUPBY() ADD MAX(veces) AS veces

RDO := (T1 JOIN T2)[dorsal] JOIN CICLISTA
```

7. Escribir una expresión del cálculo relacional que obtenga los datos de los ciclistas que siempre han quedado entre las diez primeras posiciones.

```
CICLISTAX WHERE ∀PUESTOX ( IF PUESTOX.dorsal = CICLISTAX.dorsal
                           THEN PUESTOX.posición <= 10 )
```

8. Dada la siguiente consulta "¿cuántos equipos han tenido ciclistas llevando el maillot amarillo?" y dada la siguiente sentencia:

```
SELECT  ...
FROM   LLEVA L, CICLISTA C, MAILLOT M
WHERE  L.dorsal = C.dorsal  AND  L.codigo = M.codigo
AND    M.color = 'amarillo' ;
```

¿Qué expresión debe aparecer en la cláusula SELECT?

```
COUNT(DISTINCT C.nomequipo)
```

9. Se pretende obtener el nombre y el director de cada equipo que está participando en la vuelta ciclista, junto con el número de maillots que ha llevado en total en las etapas que se han corrido. Antes de empezar la vuelta, ningún equipo ha llevado ningún maillot, por lo que todos aparecerán con un número de maillots igual a cero. Dada la siguiente sentencia SELECT ¿qué debe aparecer en la cláusula WHERE?

```
SELECT E.nomequipo, E.director, COUNT(L.dorsal)
FROM   LLEVA L, CICLISTA C, EQUIPO E
WHERE  ...
GROUP BY E.nomequipo, E.director;
```

```
L.dorsal (+) = C.dorsal AND C.nomequipo = E.nomequipo
```

10. Se desea obtener el nombre de los equipos que han tenido dos o más ciclistas entre las cinco primeras posiciones de una misma etapa.

```
SELECT DISTINCT C.nomequipo
FROM   PUESTO P, CICLISTA C
WHERE  P.dorsal = C.dorsal
...
```

¿Cómo debe continuar la sentencia para obtener el resultado deseado?

```
AND P.posicion <= 5
GROUP BY C.nomequipo, P.numetapa
HAVING COUNT(*) >= 2;
```

11. Escribir una sentencia SELECT que obtenga el nombre de los equipos que no tienen ciclistas de más de 25 años. Ten en cuenta que el atributo CICLISTA.añonacim admite nulos.

```
SELECT nomequipo
FROM   CICLISTA
WHERE  TO_NUMBER(TO_CHAR(SYSDATE,'yyyy')) - añonacim <= 25
MINUS
SELECT nomequipo
FROM   CICLISTA
WHERE  TO_NUMBER(TO_CHAR(SYSDATE,'yyyy')) - añonacim > 25;
```

12. Escribir una sentencia SELECT que muestre el nombre de los equipos tienen más de dos ciclistas que han quedado entre las tres primeras posiciones alguna vez y que han tenido un mismo maillot, sea cual sea, más de cinco veces.

```
SELECT DISTINCT C.nomequipo
FROM   LLEVA L, CICLISTA C
WHERE  L.dorsal = C.dorsal
AND    C.nomequipo IN (SELECT C.nomequipo
                       FROM   PUESTO P, CICLISTA C
                       WHERE  P.dorsal = C.dorsal
                       AND    P.posicion <= 3
                       GROUP BY C.nomequipo
                       HAVING COUNT(DISTINCT C.dorsal) > 2)
GROUP BY C.nomequipo, L.codigo
HAVING COUNT(*) > 5;
```

13. Obtener el esquema de la base de datos correspondiente a las entidades **CONVOCATORIA** y **ORGANISMO** del esquema conceptual, teniendo también en cuenta la relación que existe entre ellas: **promueve**.

CODPOSTAL(codpostal, población)

ORGANISMO(nombre, dirección, codpostal, teléfono)
ORGANISMO.codpostal es clave ajena a CODPOSTAL: no admite nulos; regla borrado: restringir.

CONVOCATORIA(programa, número, fecha_pub, fecha_lim, web, dog_boe, fecha_resol, nombre_organismo)
CONVOCATORIA.fecha_resol admite nulos.
CONVOCATORIA.nombre_organismo es clave ajena a ORGANISMO: no admite nulos; regla de borrado: restringir/propagar.

14. Añadir al esquema de la base de datos la/s tabla/s correspondiente/s a la entidad **INVESTIGADOR** del esquema conceptual.

AREA(área, departamento)

INVESTIGADOR(correoe, nombre, teléfono, despacho, área, grupo, categoría, año_ini_categ)
INVESTIGADOR.área es clave ajena a AREA: no admite nulos; regla de borrado: restringir/(propagar).

15. Añadir al esquema de la base de datos las tablas correspondientes a la entidad **SOLICITUD** del esquema conceptual, teniendo también en cuenta las relaciones en las que participa: **presenta, principal y participa**.

SOLICITUD(título, num_reg, fecha_pres, fecha_ini, fecha_fin, presup, aprobada, programa_conv, número_conv, correoe_inv_principal)
SOLICITUD.num_reg es clave alternativa.
SOLICITUD.aprobada admite nulos y su dominio es {'sí', 'no'}
(SOLICITUD.programa_conv, SOLICITUD.número_conv) es clave ajena a CONVOCATORIA: no admite nulos; regla de borrado: restringir/propagar.
SOLICITUD.correoe_inv_principal es clave ajena a INVESTIGADOR: no admite nulos; regla de borrado: restringir/(propagar).

PARTICIPA(título, correoe_inv, horas)
PARTICIPA.título es clave ajena a SOLICITUD: no admite nulos; regla de borrado: propagar.
PARTICIPA.correoe_inv es clave ajena a INVESTIGADOR: no admite nulos; regla de borrado: propagar/(restringir).

16. Modificar el esquema de la base de datos (es decir, las tablas) para que se refleje:

(a) Quien es el investigador que dirige cada grupo de investigación (sólo es uno).

Una posible solución es incluir una nueva columna en INVESTIGADOR llamada 'director' que admita nulos y que sólo pueda tomar el valor 'sí': si un investigador es director de su grupo, el atributo toma el valor 'sí'; si no lo es, el atrbuto es nulo. Estableciendo la combinación (grupo, director) como clave alternativa, nos aseguramos de que el director es uno por cada grupo.

(b) Quien es el investigador que dirige cada departamento (sólo es uno).

La solución propuesta en (a) puede utilizarse también aquí, aunque ya no es posible establecer una clave alternativa que nos permita tener un solo director por departamento, ya que el atributo departamento no está en la tabla INVESTIGADOR, está en AREA.

Otra posible solución es crear una nueva tabla de departamentos con una clave ajena al investigador que dirige cada uno de ellos:
DEPARTAMENTO(departamento, correoe_dir)
DEPARTAMENTO.correoe_dir es clave ajena a INVESTIGADOR: no admite nulos; regla de borrado: restringir
AREA(área, departamento)
AREA.departamento es clave ajena a DEPARTAMENTO: no admite nulos; regla de borrado: propagar/(restringir).