

EXAMEN DE FICHEROS Y BASES DE DATOS – F47  
21 DE JUNIO DE 2002

NOMBRE Y APELLIDOS: \_\_\_\_\_

ATENCIÓN: Lee estas instrucciones hasta el final antes de empezar a hacer el examen.

- Las preguntas del examen que son de tipo test se deben contestar en esta hoja de respuestas.
- Cada pregunta tiene tres respuestas: dos son falsas y sólo una es verdadera.
- Escoger una respuesta para cada pregunta (A, B o C) y escribirla de forma legible en la casilla correspondiente. No escribir en las casillas de color gris.
- La puntuación de cada respuesta correcta en las preguntas de tipo test es de 0,5. Cada respuesta incorrecta resta la mitad del valor de una respuesta correcta. Las preguntas que se dejan sin contestar no restan puntos.
- La puntuación de cada respuesta correcta de las preguntas que no son de tipo test (casillas en gris) es de 0,75.
- Para agilizar la corrección del examen, contesta la pregunta 8 en un folio, las preguntas 13 y 14 en otro folio, y las preguntas de la 15 a la 17 en otro folio.

1	2	3	4		
5	6	7	8		
9	10	11	12	13	14
15	16	17			

IMPORTANTE: Debido a que hay varios modelos de examen, es imprescindible entregar las hojas con los enunciados de las preguntas junto a esta hoja de respuestas al terminar el examen.

Muy bien, ahora ya puedes proceder con el examen. Suerte.

LAS PREGUNTAS DE LA 5 A LA 14 SE BASAN EN LA SIGUIENTE ESPECIFICACIÓN

---

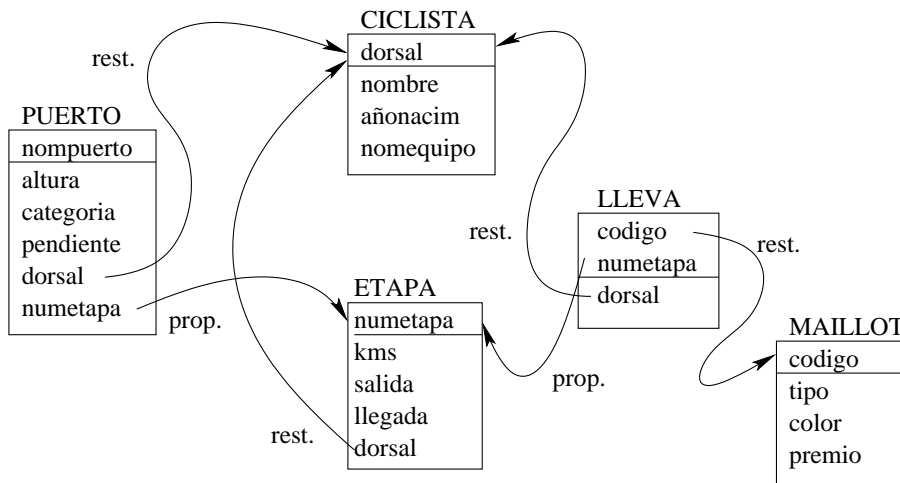
El siguiente esquema relacional representa una base de datos que almacena información sobre una vuelta ciclista. Ningún atributo acepta nulos, a menos que se especifique lo contrario.

**EQUIPO**(nomequipo, director) : Datos de los distintos equipos ciclistas que participan en la vuelta: nombre del equipo y nombre de su director.

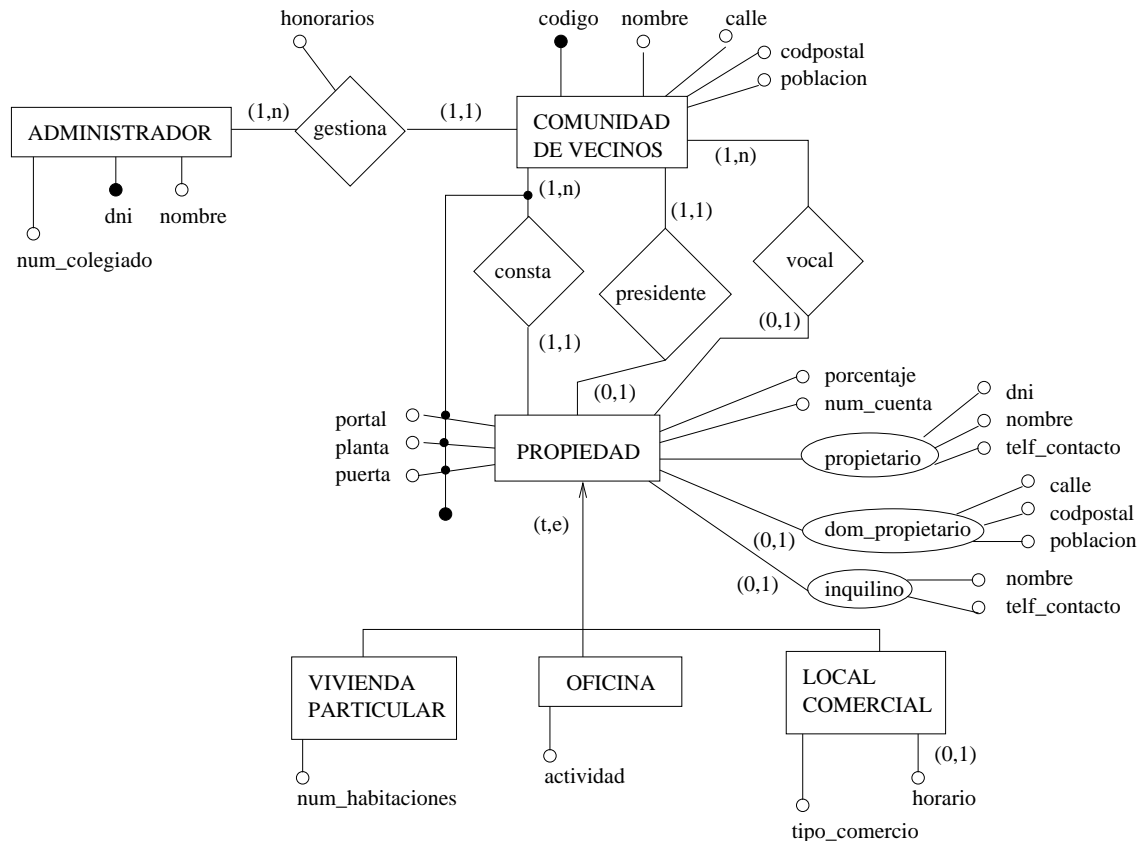
**CICLISTA**(dorsal, nombre, año, nomequipo) : Datos de los ciclistas que componen los distintos equipos: número del dorsal, nombre del ciclista, año de nacimiento del ciclista y nombre del equipo al que pertenece. **CICLISTA**.nomequipo es clave ajena a **EQUIPO**; regla de borrado: propagar.

**ETAPA**(numetapa, kms, salida, llegada, dorsal) : Datos de las etapas que componen la vuelta ciclista: número de la etapa (las etapas se numeran consecutivamente: 1, 2, ...), kilómetros que tiene la etapa, nombre de la población de donde sale la etapa, nombre de la población donde se encuentra la meta de la etapa y número del dorsal del ciclista que ha ganado la etapa. **ETAPA**.salida y **ETAPA**.llegada están definidas sobre el mismo dominio. **ETAPA**.dorsal es clave ajena a **CICLISTA**; acepta nulos (aún no se ha corrido la etapa); regla de borrado: restringir.

**PUERTO**(nompuerto, altura, categoría, pendiente, numetapa, dorsal) : Datos de los puertos de montaña que visita la vuelta ciclista: nombre del puerto, altura máxima, categoría del puerto (primera, especial, etc.), pendiente media del puerto, número de la etapa donde se pasa por él y número del dorsal que ha ganado el puerto al pasar en primera posición. **PUERTO**.numetapa es clave ajena a **ETAPA**; regla de borrado: propagar. **PUERTO**.dorsal es clave ajena a **CICLISTA**; acepta nulos (aún no se ha corrido la etapa que pasa por el puerto); regla de borrado: restringir.



El esquema conceptual que se muestra a continuación corresponde a una firma de abogados dedicada a la administración de fincas.



La firma tiene varios abogados que ejercen de administradores de una o más comunidades de vecinos, tarea por la que cada uno de ellos cobra unos honorarios anuales. Cada comunidad de vecinos es gestionada por un solo administrador y tiene un código que la identifica. Además, la comunidad tiene un nombre y una dirección (calle, código postal y población). Cada comunidad consta de una serie de propiedades que pueden ser de tres tipos: vivienda particular, local comercial y oficina.

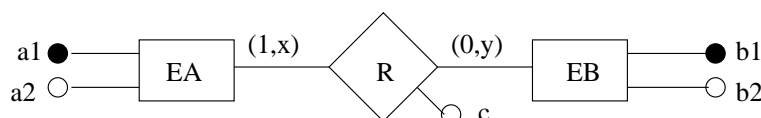
Cada propiedad se identifica dentro de su comunidad por su número de portal, planta y puerta. Además, interesa conocer los datos personales del propietario (DNI, nombre y teléfono de contacto). Si el propietario no habita la propiedad, también interesa conocer su dirección completa y los datos del inquilino (nombre y teléfono de contacto), si es que está habitada. Cada propiedad tiene un porcentaje de participación en los gastos de la comunidad y una cuenta bancaria en la que el propietario desea que se le domicilie el pago de los recibos.

Cada comunidad de vecinos tiene además un presidente y varios vocales, elegidos entre todos los propietarios, y que son quienes se encargan de tratar directamente con el administrador los distintos problemas que puedan surgir.

EXAMEN DE FICHEROS Y BASES DE DATOS – F47  
21 DE JUNIO DE 2002

NOMBRE Y APELLIDOS: \_\_\_\_\_

1. Se ha diseñado un fichero disperso utilizando la dispersión extensible. ¿Cuándo aumentará el tamaño del directorio?
  - (A) Cuando el número de registros en un bloque supere el 69 %.
  - (B) Cuando se llene un bloque con profundidad local menor a la global.
  - (C) Cuando se llene un bloque con profundidad local igual a la global.
2. Dado un árbol B de orden 100 y dado un determinado nodo interno que tiene 95 punteros. ¿Cuántos valores del campo de indexación hay en dicho nodo?
  - (A) 47 valores.
  - (B) 94 valores.
  - (C) 95 valores.
3. Los nodos de un árbol B+ ...
  - (A) ... siempre están por encima de la mitad de su capacidad.
  - (B) ... siempre están por encima del 69 % de su capacidad.
  - (C) ... siempre están entre el 50 % y el 69 % de su capacidad
4. Un índice, sea del tipo que sea, también se puede construir sobre una combinación de campos. Por ejemplo, sobre un fichero de personas se puede construir un índice combinado sobre  $\langle \text{apellido1}, \text{apellido2}, \text{nombre} \rangle$  siendo éstos tres campos del fichero. Las entradas del índice estarán ordenadas del mismo modo que ordenaría la cláusula `ORDER BY apellido1, apellido2, nombre` en una consulta de SQL. ¿En qué tipo de consultas podemos utilizar el índice para acelerar el acceso a los datos?
  - (A) En consultas en las que se especifica sólo apellido1, en consultas en las que se especifica sólo apellido1 y apellido2, y en consultas en las que se especifica apellido1, apellido2 y nombre.
  - (B) En consultas en las que se especifique sólo apellido1, o sólo apellido2, o sólo nombre. También en consultas en las que se especifique sólo apellido1 y apellido2, o sólo apellido1 y nombre, o sólo apellido2 y nombre. Y también en consultas en las que se especifiquen los tres campos.
  - (C) Ninguna. Un índice tan raro no puede servir para muchas cosas.
5. Dado el siguiente esquema conceptual:



donde  $x \in \{1, n\}$  e  $y \in \{1, n\}$  y dadas las siguientes tablas: EA(a1, a2), EB(b1, b2) y R(a1, b1, c).

- (A) La clave ajena R.a1 no acepta nulos y la clave ajena R.b1 sí los acepta.
- (B) Ambas claves ajenas R.a1 y R.b1 no aceptan nulos.
- (C) Sólo sabremos si las claves ajenas R.a1 y R.b1 aceptan nulos una vez se decida qué atributos forman la clave primaria de R (será en función de las cardinalidades máximas).
6. Dada la siguiente expresión del álgebra relacional: ( ( SUMMARIZE ETAPA GROUPBY() ADD MAX(kms) AS kms ) JOIN ETAPA ) JOIN CICLISTA ¿A cuál de las siguientes expresiones es equivalente?
- (A) CICLISTAX WHERE  $\forall$  ETAPAX (IF ETAPAX.dorsal=CICLISTAX.dorsal THEN NOT  $\exists$  ETAPAY (ETAPAY.kms<ETAPAX.kms))
- (B) CICLISTAX WHERE  $\exists$  ETAPAX (ETAPAX.dorsal=CICLISTAX.dorsal AND  $\forall$  ETAPAY (ETAPAX.kms>ETAPAY.kms))
- (C) CICLISTAX WHERE  $\forall$  ETAPAX (ETAPAX.dorsal=CICLISTAX.dorsal AND NOT  $\exists$  ETAPAY (ETAPAY.kms<ETAPAX.kms))
7. Dada la siguiente expresión del cálculo relacional: CICLISTAX WHERE NOT  $\exists$  ETAPAX (ETAPAX.dorsal=CICLISTAX.dorsal AND  $\exists$  LLEVAX (LLEVAX.dorsal=CICLISTAX.dorsal) ) ¿A cuál de las siguientes expresiones es equivalente?
- (A) (LLEVA[dorsal] INTERSECT ETAPA[dorsal]) JOIN CICLISTA
- (B) (LLEVA[dorsal] JOIN ETAPA[dorsal]) JOIN CICLISTA
- (C) (LLEVA[dorsal] MINUS ETAPA[dorsal]) JOIN CICLISTA
8. Escribe una expresión del álgebra relacional que obtenga el nombre de los equipos que tienen la media de edad mayor de todos los equipos.
9. Se desea conocer el nombre de cada ciclista mayor de 25 años junto con el número de puertos que ha ganado en toda la vuelta. El resultado debe mostrar todos los ciclistas mayores de 25 años, aunque no hayan ganado ningún puerto. Partimos de la siguiente sentencia:
- ```
SELECT c.dorsal, c.nombre
FROM puerto p, ciclista c
WHERE p.dorsal = c.dorsal
AND TO_NUMBER(TO_CHAR(SYSDATE, 'yyyy'))-c.añonacim > 25
GROUP BY c.dorsal, c.nombre;
```
- (A) Para obtener el resultado deseado, la primera condición de la cláusula WHERE debería ser “p.dorsal (+) = c.dorsal” y la cláusula SELECT debería ser “SELECT c.dorsal, c.nombre, COUNT(\*)”
- (B) Para obtener el resultado deseado, la cláusula WHERE debería ser “WHERE p.dorsal = c.dorsal (+)”, la restricción sobre la edad se debería situar en el HAVING y la cláusula SELECT debería ser “SELECT c.dorsal, c.nombre, COUNT(DISTINCT p.nompuerto)”
- (C) Para obtener el resultado deseado, la primera condición de la cláusula WHERE debería ser “p.dorsal (+) = c.dorsal” y la cláusula SELECT debería ser “SELECT c.dorsal, c.nombre, COUNT(p.nompuerto)”

10. Dada la siguiente sentencia ¿cuál de las siguientes afirmaciones es cierta?

```
SELECT DISTINCT l1.maillot
FROM lleva l1
WHERE NOT EXISTS (SELECT *
                  FROM lleva l2
                  WHERE l2.maillot = l1.maillot
                  AND l1.dorsal <> l2.dorsal);
```

- (A) Obtiene los maillots que sólo han sido llevados por un ciclista.
- (B) Obtiene los maillots que sólo han sido llevados por más de un ciclista.
- (C) Obtiene los maillots que nunca han sido llevados por ningún ciclista.

11. Dada la siguiente sentencia SQL:

```
SELECT COUNT(DISTINCT dorsal)
FROM etapa
WHERE dorsal IS NULL;
```

- (A) Obtiene el número de etapas que restan para terminar la vuelta ciclista.
- (B) Obtiene el valor 0.
- (C) Obtiene el valor 1.

12. Dadas las sentencias que se muestran a continuación:

```
S1:SELECT c1.dorsal, c1.nombre, COUNT(*)
FROM etapa e, ciclista c1
WHERE 5 < (SELECT COUNT(*)
          FROM ciclista c2
          WHERE c2.nomequipo = c1.nomequipo)
AND e.dorsal = c1.dorsal
GROUP BY c1.dorsal, c1.nombre;
```

```
S2:SELECT c.dorsal, c.nombre, COUNT(*)
FROM etapa e, ciclista c
WHERE e.dorsal = c.dorsal
GROUP BY c.dorsal, c.nombre;
HAVING COUNT(DISTINCT c.nomequipo) > 5;
```

- (A) Ambas sentencias obtienen siempre el mismo resultado.
- (B) La sentencia S1 obtiene un número de filas mayor o igual que el número de filas que obtiene la sentencia S2.
- (C) La sentencia S2 obtiene un número de filas mayor o igual que el número de filas que obtiene la sentencia S1.

13. Escribe una sentencia SQL que obtenga los datos de los ciclistas que sólo han ganado etapas con algún puerto de montaña.

14. Escribe una sentencia SQL que obtenga los datos del ciclista ganador de la etapa con el puerto más alto.

Todas las relaciones base que presentes en los siguientes apartados deben estar en tercera forma normal. En ellas debes indicar las claves primarias, claves alternativas, claves ajenas y la regla de borrado para éstas (si hay varias reglas posibles, especificálas).

Cualquier suposición que hagas debes especificarla. No se considera correcta la introducción de nueva información en el esquema de la base de datos, tan sólo deben aparecer aquellos atributos que se muestran en el esquema conceptual.

15. Obtener el esquema de la base de datos correspondiente a las entidades **ADMINISTRADOR** y **COMUNIDAD DE VECINOS** del esquema conceptual, teniendo también en cuenta la relación que existe entre ellas: **gestiona**.
16. Añadir al esquema de la base de datos la entidad **PROPIEDAD** con su jerarquía y la relación **consta**. Nótese que es posible que un propietario posea varias propiedades.
17. Añadir al esquema de la base de datos la información de las relaciones **presidente** y **vocal**. Es importante que el propio esquema de la base de datos recoja la restricción de que un presidente y un vocal sólo pueden ejercer este papel en su propia comunidad.