FICHEROS Y BASES DE DATOS **2º ITIG** **23/9/2000**

NOMBRE _____

Las preguntas del examen 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. Cada respuesta correcta vale 0,2 puntos. Cada respuesta incorrecta resta la mitad del valor de una respuesta correcta (0,1 puntos). Las preguntas que se dejan sin contestar no restan puntos.

GRUPO 1 — 1 PUNTO

1	2	3	4	5

GRUPO 2 — 3 PUNTOS

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

GRUPO 3 — 3 PUNTOS

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

GRUPO 4 — 3 PUNTOS

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

IMPORTANTE: Hay varias versiones del examen, por lo que es imprescindible entregar esta hoja de respuestas junto a las preguntas del test al terminar el examen. Si alguna hoja de respuestas no va acompañada del test correspondiente, el examen no podrá ser corregido y se calificará como suspenso.

Firma:

FICHEROS Y BASES DE DATOS **2º ITIG** **23/9/2000**

NOMBRE _____

Firma:

GRUPO 1 1 PUNTO

1. Se ha creado un árbol B y un árbol B+ como índices sobre un campo clave de un fichero desordenado.

(A) El árbol B tendrá los mismos o más niveles que el árbol B+.

(B) El árbol B+ tendrá los mismos o más niveles que el árbol B.

(C) Ambos árboles tienen el mismo número de niveles, pero el B+ apunta a más registros.

2. ¿En qué situación debe doblar su tamaño el directorio de un fichero de datos que utiliza la dispersión (*hashing*) extensible?

(A) Cuando el fichero se ha llenado y ha de utilizar listas de desborde para insertar nuevos registros de datos.

(B) Cuando todas sus entradas apuntan a bloques distintos y se debe insertar un registro de datos en un bloque que está lleno.

(C) Cuando todos los nodos hoja del árbol se encuentran al mismo nivel.

3. ¿Cuándo se desdobla un bloque de un fichero de datos que utiliza la dispersión (*hashing*) lineal?

(A) Cuando le toca el turno.

(B) Cuando se llena.

(C) Cuando su lista de desborde ocupa un bloque entero.

4. ¿Quién realiza la gestión de ficheros en un sistema de bases de datos?

(A) El sistema operativo.

(B) El sistema de gestión de bases de datos.

(C) El programador de aplicaciones.

5. El lenguaje SQL:

(A) Es un lenguaje navegacional.

(B) Es un lenguaje orientado a objetos (relaciones, tuplas)

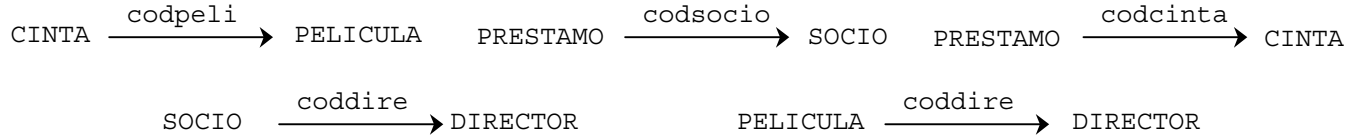
(C) Es un lenguaje de cuarta generación.

GRUPO 2 — 3 PUNTOS

Las relaciones que forman la base de datos de un vídeo club son las siguientes:

```
PELICULA(codpeli,título,año,género,coddire,idioma,país)
CINTA(codcinta,codpeli,idioma)
SOCIO(codsocio,nombre,dirección,teléfono,género,coddire)
PRESTAMO(codsocio,codcinta,fecha,pres_dev)
DIRECTOR(coddire,nombre,año_nac,país)
```

En las relaciones anteriores, son claves primarias los atributos y grupos de atributos que aparecen subrayados. Las claves ajenas se muestran en los siguientes diagramas referenciales:



El vídeo club posee copias (CINTA) de películas (PELICULA) que presta (PRESTAMO) a sus socios (SOCIO). De las películas se guarda el código, título, año de realización, género cinematográfico y director. Cada película está rodada en un idioma y en un país, y puede haber copias en otros idiomas distintos (porque ha sido doblada), por lo que en cada copia se indica el idioma en el que se encuentra. De los directores se guarda el nombre, año de nacimiento y país en que vive (DIRECTOR). De los socios se guarda el código, nombre, dirección, teléfono y su director y género favoritos, si es que los tiene. Por último, de los préstamos interesa la cinta que se presta, el socio al cual se presta y la fecha. Además, mientras una cinta está prestada, el atributo pres_dev tiene el valor '*prestada*' y al finalizar el préstamo su valor es '*devuelta*'.

1. ¿Cuáles de las siguientes expresiones responden a la consulta 'directores que sólo han dirigido películas rodadas en su mismo país'?

```
E1: DIRECTORX WHERE ∀PELICULAX (PELICULAX.coddire<>DIRECTORX.coddire
                                OR PELICULAX.país=DIRECTORX.país)
E2: DIRECTORX WHERE ∀PELICULAX (IF PELICULAX.país=DIRECTORX.país
                                THEN PELICULAX.coddire=DIRECTORX.coddire)
E3: DIRECTORX WHERE NOT ∃PELICULAX (PELICULAX.país<>DIRECTORX.país
                                AND PELICULAX.coddire=DIRECTORX.coddire)
```

(A) Las expresiones E1 y E2.

(B) Las expresiones E1 y E3.

(C) Las expresiones E2 y E3.

2. ¿A qué consulta responde la siguiente expresión del cálculo relacional?

```
SOCIOX WHERE ∀PRESTAMOX ∃CINTAX ∃PELICULAX
 (IF PRESTAMOX.codsocio=SOCIOX.codsocio THEN PRESTAMOX.codcinta=CINTAX.codcinta
     AND CINTAX.codpeli=PELICULAX.codpeli AND CINTAX.idioma=PELICULAX.idioma)
```

(A) Socios que siempre toman prestadas películas en el mismo idioma.

(B) Socios que han tenido préstamos de películas en todos los idiomas.

(C) Socios que sólo toman películas en versión original.

3. ¿A qué consulta responde la siguiente expresión del cálculo relacional?

```
DIRECTORX WHERE ∃PELICULAX (DIRECTORX.coddire=PELICULAX.coddire
AND ∀PELICULAY ∃DIRECTORY (PELICULAY.coddire=DIRECTORY.coddire AND
PELICULAX.año-DIRECTORX.año_nac<=PELICULAY.año-DIRECTORY.año_nac))
```

(A) Directores que han realizado películas después del director más joven.

(B) Directores que han hecho alguna película siendo los más jóvenes.

(C) Directores de la misma edad que han empezado a rodar películas más jóvenes.

4. Se quiere responder la consulta 'directores y géneros tales que a los socios a los que les gusta el mismo director, también les gusta el mismo género cinematográfico'. ¿Cuál de las siguientes sentencias responde a esta consulta?

(A)
```
SOCIOX.coddire,SOCIOX.género WHERE ∀SOCIOY
  (IF SOCIOX.coddire=SOCIOY.coddire THEN SOCIOX.género=SOCIOY.género)
```
(B)
```
SOCIOX.coddire,SOCIOX.género WHERE ∃SOCIOY
  (IF SOCIOX.coddire=SOCIOY.coddire THEN SOCIOX.género=SOCIOY.género)
```
(C)
```
SOCIOX.coddire,SOCIOX.género WHERE ∃SOCIOY
  (SOCIOX.coddire=SOCIOY.coddire AND SOCIOX.género=SOCIOY.género)
```

5. ¿A qué consulta responde la siguiente expresión del álgebra relacional?

```
T1 := (DIRECTOR JOIN PELICULA)[coddire]
T2 := ((DIRECTOR TIMES PELICULA) WHERE DIRECTOR.país<>PELICULA.país)[coddire]
RDO := T1 INTERSECT T2
```

(A) Directores que sólo han dirigido películas rodadas en su mismo país.

(B) Directores que han dirigido películas en, al menos, dos países.

(C) Directores que sólo han dirigido películas en países que no son el suyo.

6. Para responder a la consulta 'socios que han tomado prestadas películas de todos los directores' mediante el álgebra relacional, obtendremos los socios que han visto películas de tantos directores como directores hay en la base de datos. Para ello, construimos la siguiente expresión:

```
T1 := (PRESTAMO JOIN CINTA JOIN PELICULA[proy1])[proy2]
T2 := SUMMARIZE T1 GROUPBY(atrib1) ADD COUNT(*) AS directores
T3 := SUMMARIZE (PELICULA[proy3]) GROUPBY(atrib2) ADD COUNT(*) AS directores
RDO := T2 oper T3
```

Los atributos que deben aparecer en `proy1` son:

(A) `codpeli,coddire`

(B) `codpeli`

(C) No hay que hacer proyección.

7. Continuando con la expresión del ejercicio 6, los atributos que deben aparecer en `proy2` son:

(A) `codsocio`

(B) `codsocio,coddire`

(C) No hay que hacer proyección.

8. Continuando con la expresión del ejercicio 6, los atributos que deben aparecer en `atrib1` son:

(A) No debe aparecer ningún atributo.

(B) `codsocio`

(C) `codsocio,coddire`

9. Continuando con la expresión del ejercicio 6, los atributos que deben aparecer en `proy3` son:

(A) `coddire`

(B) `codpeli,coddire`

(C) No hay que hacer proyección.

10. Continuando con la expresión del ejercicio 6, los atributos que deben aparecer en `atrib2` son:

(A) No debe aparecer ningún atributo.

(B) `codpeli`

(C) `codpeli,coddire`

11. Continuando con la expresión del ejercicio 6, el operador que debe aparecer en `oper` es:

(A) `JOIN`

(B) `TIMES`

(C) `MINUS`

12. Se quiere responder a la consulta 'películas que tienen copias en todos los idiomas'. Para ello se plantea la siguiente expresión del álgebra relacional: `CINTA[proy1] DIVIDEBY CINTA[proy2]`

Los atributos que deben aparecer en `proy1` son:

(A) `codpeli,idioma`

(B) No hace falta ninguna proyección.

(C) `idioma`

13. Continuando con el ejercicio 12, los atributos que deben aparecer en `proy2` son:

(A) `codpeli, idioma`

(B) No hace falta ninguna proyección

(C) `idioma`

14. Sabiendo que las claves ajenas de la base de datos no aceptan nulos ¿cuántas filas tendrá el resultado de realizar la siguiente expresión del álgebra relacional?

```
PELICULA JOIN DIRECTOR[coddire,nombre] JOIN CINTA[codcinta,codpeli]
```

(A) Tantas filas como tiene `DIRECTOR`.

(B) Tantas filas como tiene `PELICULA`.

(C) Tantas filas como tiene `CINTA`.

15. El resultado de la expresión `SOCIO JOIN PRESTAMO JOIN CINTA JOIN PELICULA` ...

(A) ... obtiene los préstamos de películas en versión original donde el género y el director le gustan al socio.

(B) ... obtiene los socios que han tomado prestada alguna copia de alguna película.

(C) ... obtiene las películas que se han prestado, sea cual sea el idioma y los gustos del socio.

GRUPO 3 3 PUNTOS

1. Dada la siguiente consulta 'obtener los clientes que han comprado en más de 10 ocasiones un mismo artículo' y dadas las sentecias S1 y S2:

S1:
```
SELECT DISTINCT f.codcli
FROM   lineas_fac lf, facturas f
WHERE  lf.codfac = f.codfac
GROUP BY f.codcli, lf.codart
HAVING COUNT(DISTINCT f.codfac)>10;
```

S2:
```
SELECT f.codcli, lf.codart
FROM   lineas_fac lf, facturas f
WHERE  lf.codfac = f.codfac
GROUP BY f.codcli
HAVING COUNT(DISTINCT f.codfac)>10;
```

(A) La sentencia S1 responde a la consulta, mientras que la sentencia S2 no lo hace.

(B) La sentencia S2 responde a la consulta, mientras que la sentencia S1 no lo hace.

(C) Ninguna de las dos sentencias responde a la consulta.

2. Dada la siguiente consulta: 'obtener los artículos a los que siempre se ha aplicado el mismo descuento' y dadas las sentencias S3 y S4:

S3:
```
SELECT codart,dto
FROM   lineas_fac
GROUP BY codart
HAVING COUNT(DISTINCT NVL(dto,0))=1;
```

S4:
```
SELECT codart
FROM   lineas_fac
GROUP BY codart,dto
HAVING COUNT(DISTINCT NVL(dto,0))=1;
```

(A) La sentencia S3 responde a la consulta, mientras que la sentencia S4 no lo hace.

(B) La sentencia S4 responde a la consulta, mientras que la sentencia S3 no lo hace.

(C) Ninguna de las dos sentencias responde a la consulta.

3. Dada la consulta: 'obtener los clientes que no compran artículos de más de 10.000 pesetas' y dadas las sentencias S5 y S6:

S5:
```
SELECT f.codcli
FROM   lineas_fac lf, facturas f
WHERE  lf.codfac = f.codfac
AND    lf.precio > 10000
GROUP BY f.codcli
HAVING COUNT(lf.codart) = 0;
```

S6:
```
SELECT f.codcli
FROM   lineas_fac lf, facturas f
WHERE  lf.codfac = f.codfac
AND    lf.precio <= 10000
MINUS
SELECT f.codcli
FROM   lineas_fac lf, facturas f
WHERE  lf.codfac = f.codfac
AND    lf.precio > 10000;
```

(A) La sentencia S5 responde a la consulta, mientras que la sentencia S6 no lo hace.

(B) La sentencia S6 responde a la consulta, mientras que la sentencia S5 no lo hace.

(C) Las dos sentencias responden a la consulta.

4. Dada la consulta: 'obtener los artículos que han sido comprados por 10 o más clientes y en más de 30 facturas' y dada la sentencia S7:

S7:
```
SELECT lf.codart
FROM   lineas_fac lf, facturas f
WHERE  lf.codfac = f.codfac
GROUP BY lf.codart
HAVING ???
```

¿Cuál de las siguientes expresiones es la que debe aparecer en la parte del HAVING?

(A) `COUNT(DISTINCT f.codcli) >= 10 AND COUNT(DISTINCT f.codfac) > 30`

(B) `COUNT(DISTINCT f.codcli) >= 10 AND COUNT(f.codfac) > 30`

(C) `COUNT(f.codcli) >= 10 AND COUNT(f.codfac) > 30`

4. Dada la consulta: 'obtener los clientes con los artículos que siempre compran sin descuento', se ha empezado a escribir la sentencia S8 para resolverla en SQL. Las siguientes cuestiones tratan de completar esta sentencia para que consiga su objetivo.

```
S8: SELECT f.codcli,lf.codart
    FROM   lineas_fac lf, facturas f
    WHERE  lf.codfac = f.codfac
```

En el `WHERE` falta ...

(A) `AND NVL( lf.dto,0 ) = 0`

(B) `AND COUNT( DISTINCT lf.dto ) = 0`

(C) no falta nada.

5. Continuando con la sentencia S8, las columnas que deben aparecer en la parte del GROUP BY son:

(A) `f.codcli`

(B) `f.codcli, lf.codart`

(C) `lf.codart`

6. Continuando con la sentencia S8, la expresión de la parte del HAVING debe ser:

(A) `COUNT( DISTINCT NVL( lf.dto,0 ) )  = 1  AND NVL( lf.dto,0 ) = 0`

(B) `MIN( NVL( lf.dto,0 ) ) = MAX( NVL( lf.dto,0 ) )`

(C) `SUM( NVL( lf.dto,0 ) ) = 0`

7. ¿A qué consulta responde la sentencia S9?

```
S9: SELECT codcli
    FROM   facturas
    GROUP BY codcli
    HAVING COUNT(*)/(MAX(fecha)-MIN(fecha)) >=
                ALL(SELECT COUNT(*)/(MAX(fecha)-MIN(fecha))
                    FROM fecturas
                    GROUP BY codcli);
```

(A) Clientes que han comprado con más frecuencia.

(B) Clientes que entre su primera y última factura han comprado más veces.

(C) Clientes que han comprado en tantos días como al menos todos los demás.

8. Teniendo en cuenta que no hay artículos sin ventas y dadas las sentencias S10 y S11:

```
S10: SELECT codart
     FROM   lineas_fac
     WHERE  cant > 5
     MINUS
     SELECT codart
     FROM   lineas_fac
     WHERE  cant <= 5;
```

```
S11: SELECT a.codart
     FROM   articulos a
     WHERE  5 < ALL (SELECT l.cant
                     FROM   lineas_fac l
                     WHERE  l.codart = a.codart);
```

(A) Las sentencias S10 y S11 son equivalentes, obtienen el mismo resultado.

(B) Las sentencias S10 y S11 no son equivalentes, obtienen distintos resultados.

(C) Para obtener el mismo resultado con las sentencias S10 y S11 habría que corregir un error de sintaxis en el operador `ALL`.

9. Dadas las sentencias S12 y S13:

S12:
```
SELECT f.codcli
FROM   facturas f
WHERE  0 = ALL
    (SELECT l.dto
     FROM   lineas_fac l
     WHERE  l.codfac=f.codfac)
GROUP BY f.codcli
HAVING  MAX(f.iva)=MIN(f.iva);
```

S13:
```
SELECT f.codcli
FROM   facturas f,lineas_fac l
WHERE  f.codfac=l.codfac
GROUP BY f.codcli
HAVING  MAX(f.iva)=MIN(f.iva)
AND     MAX(l.dto)=0;
```

(A) Las sentencias S12 y S13 son equivalentes, obtienen el mismo resultado.

(B) Las sentencias S12 y S13 no son equivalentes, obtienen distintos resultados.

(C) Para obtener el mismo resultado con las sentencias S12 y S13 habría que corregir un error de sintaxis en el operador `ALL`.

10. Dada la consulta: 'obtener el número de facturas que tienen menos de tres líneas' y dadas las sentencias S14 y S15:

S14:
```
SELECT COUNT(*)
FROM   facturas f,lineas_fac l
WHERE  f.codfac = l.codfac
AND    l.linea < 3
GROUP BY f.codfac;
```

S15:
```
SELECT COUNT(COUNT(*))
FROM   facturas f,lineas_fac l
WHERE  f.codfac=l.codfac
GROUP BY f.codfac
HAVING  COUNT(l.linea)<3;
```

(A) La sentencia S14 responde a la consulta, mientras que la sentencia S15 no lo hace.

(B) La sentencia S15 responde a la consulta, mientras que la sentencia S14 no lo hace.

(C) Las dos sentencias responden a la consulta.

11. Para responder a la consulta 'obtener los clientes con facturas de la provincia de Castellón, incluyendo también aquellos de la provincia de Valencia que tengan alguna factura sin iva', se ha empezado a escribir la siguiente sentencia:

S16:
```
SELECT DISTINCT c.codcli, c.nombre, p.codpro
FROM   clientes c, facturas f, pueblos p
WHERE  c.codcli = f.codcli AND
       c.codpue = p.codpue AND ...
```

En el `WHERE` falta la condición:

a) `p.codpro in ('12', decode(nvl(f.iva,0),0,'46',0));`

b) `(p.codpro = '12' or (p.codpro = '46' and f.iva = 0 or f.iva is null));`

c) `(p.codpro in ('12','46') and nvl(f.iva, 0) = 0);`

12. La siguiente sentencia rata de responder a la consulta 'contar el número de facturas que se han hecho cada año desde 1990'.

```
S17: SELECT TO_CHAR(fecha,'yyyy'),COUNT(*)
     FROM   facturas
     GROUP BY TO_CHAR(fecha,'yyyy')
     HAVING TO_CHAR(fecha,'yyyy') > '1990';
```

(A) La condición del `HAVING` no es necesaria, se debería poner en el `WHERE`.

(B) La condición del `HAVING` debería ser `TO_NUMBER(TO_CHAR(fecha,'yyyy'))>1990` y por tanto en el `SELECT` y en el `GROUP BY` debería aparecer `TO_NUMBER(TO_CHAR(fecha,'yyyy'))`

(C) La condición del `HAVING` debería ser `MIN(TO_CHAR(fecha,'yyyy'))= '1990'`

13. Dadas las sentencias S18 y S19:

```
S18: SELECT c.nombre,COUNT(*)
     FROM   facturas f,clientes c
     WHERE  f.codcli = c.codcli
     GROUP BY c.codcli,c.nombre;
```

```
S19: SELECT c.nombre,COUNT(*)
     FROM   facturas f,clientes c
     WHERE  f.codcli = c.codcli
     GROUP BY c.nombre;
```

(A) Ambas sentencias devuelven siempre el mismo resultado.

(B) Ambas sentencias no devolverán nunca el mismo resultado porque se está agrupando por columnas distintas.

(C) Ambas sentencias pueden devolver el mismo resultado si cada cliente tiene un nombre distinto.

14. Teniendo en cuenta que los descuentos y el iva son atributos que aceptan nulos:

```
SELECT COUNT(*) a, COUNT(dto) b, COUNT(DISTINCT dto) c,
       COUNT(NVL(dto,0)) d, COUNT(DISTINCT NVL(dto,0)) e
FROM   lineas_fac;
```

Las columnas que se obtienen en el resultado siempre cumplen:

(A) $a \geq b \geq c$; $d = a$; $e = c$

(B) $a \geq b = c$; $d = b$; $e \geq c$

(C) $a = b \geq c$; $d \leq a$; $e \leq c$

15. La siguiente sentencia SQL:

```
SELECT COUNT(COUNT(*))
FROM   lineas_fac
GROUP BY codfac
HAVING  MAX(NVL(dto,0)) = MIN(NVL(dto,0));
```

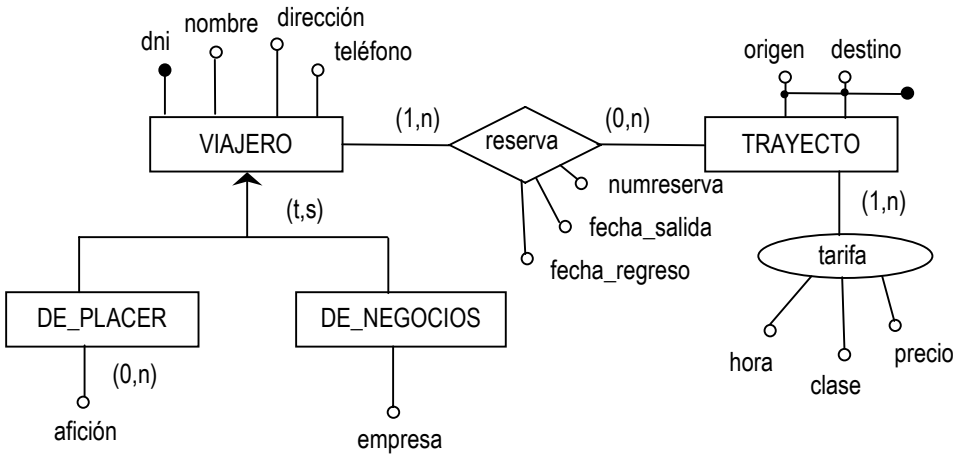
(A) Devuelve varias filas, cada una indica el número de líneas de las facturas en las que se ha aplicado el mismo descuento.

(B) Devuelve una sola fila, que es el número de facturas que en todas sus líneas tienen el mismo descuento.

(C) Devuelve varias filas: tantas como descuentos iguales se han aplicado en las líneas de factura.

GRUPO 4 3 PUNTOS

El siguiente esquema conceptual describe la información de reservas de tren de una agencia de viajes:



Las reservas de billetes de tren las realizan los mismos viajeros, de placer o de negocios. Cuando un viajero hace una reserva debe indicar el trayecto que desea realizar y las fechas de salida y de regreso. También debe indicar la clase en la que quiere viajar (primera, turista, etc.) y la hora de salida, ya que el precio varía en función éstas. Cada trayecto tiene una tarifa de precios distinta: la tarifa del trayecto Castellón – Barcelona no es la misma que la del trayecto Castellón – París. Una vez realizada la reserva, se informa al viajero de su número de reserva, que es distinto para cada una de ellas.

1. ¿Cuál de las siguientes representaciones es la más adecuada para la jerarquía VIAJERO?

(A) VIAJERO_DE_PLACER(<u>dni</u>,nombre,dirección,teléfono)
AFICION(<u>dni,afición</u>) , dni es clave ajena a VIAJERO_DE_PLACER
VIAJERO_DE_NEGOCIOS(<u>dni</u>,nombre,dirección,teléfono,empresa)

(B) VIAJERO(<u>dni</u>,nombre,dirección,teléfono,tipo) donde tipo∈ {de_placer, de_negocios}
AFICION(<u>dni,afición</u>) , dni es clave ajena a VIAJERO
EMPRESA(<u>dni,empresa</u>) , dni es clave ajena a VIAJERO

(C) Las dos descomposiones son igualmente adecuadas y no pueden plantear ningún problema.

2. La tabla RESERVA(numreserva,dni,origen,destino,fecha_salida,fecha_regreso,clase,hora) representa la relación de muchos a muchos 'reserva' que aparece en el esquema. ¿Qué atributos forman su clave primaria?

(A) numreserva.

(B) numreserva y dni.

(C) numreserva, dni, origen y destino.

3. El atributo dni de la tabla RESERVA es una clave ajena al viajero que ha realizado la reserva. Esta clave ajena ...

(A) No puede aceptar nulos.

(B) Debe aceptar nulos obligatoriamente.

(C) Habrá que preguntar a la agencia de viajes si debe o no aceptar nulos.

4. ¿De qué modo se debe representar la entidad TRAYECTO?

(A) TRAYECTO(<u>origen,destino</u>,clase,hora,precio)

(B) TRAYECTO(origen,destino)
TARIFA(clase,hora,precio,origen,destino) , (origen,destino) clave ajena a TRAYECTO

(C) TRAYECTO(origen,destino)
TARIFA(clase,hora,origen,destino,precio) , (origen,destino) clave ajena a TRAYECTO

5. Se sabe que la tarifa de precios es la misma para los trayectos que realizan el mismo recorrido en uno y otro sentido (la tarifa es igual para el trayecto Madrid – Valencia que para el trayecto Valencia – Madrid). Para tener en cuenta este hecho en la base de datos, sin que se almacene información redundante, se puede añadir un atributo denominado 'sentido' que tomará valores en el conjunto {de origen a destino, de destino a origen}. ¿En qué tabla se debe añadir este atributo?

(A) TRAYECTO

(B) TARIFA

(C) RESERVA

6. La agencia tiene una serie de descuentos aplicables a los viajeros en función de lo que se han gastado en los últimos seis meses. Por ejemplo, el descuento de tipo A es el que se aplica a los viajeros que han gastado más de 220.000 pesetas en los últimos seis meses. Se han incluido tres atributos en la/s tabla/s que almacenan los datos de los viajeros (tipo,importe,dto) indicando el tipo de descuento, el importe tope que marca el tipo de descuento y el descuento que se aplica al tipo correspondiente.

(A) Al añadir estos atributos aparecen dependencias funcionales no deseadas: tipo → importe, tipo → dto

(B) Al incluir estos tres atributos, tipo debe entrar a formar parte de la clave primaria de la/s tabla/s con los datos de los viajeros.

(C) Ya que el descuento se aplica al viajero, el único atributo que se debe incluir en la/s tabla/s de viajeros es el descuento. Los atributos tipo e importe no hay que almacenarlos en la base de datos.

7. Un centro universitario tiene tres titulaciones, siendo posible que un alumno se matricule en varias titulaciones. Cada titulación tiene unas propiedades específicas, por lo que hay que representarlas mediante una jerarquía ¿de qué tipo deberá ser ésta?

(A) Total y exclusiva.

(B) Total y superpuesta.

(C) Parcial y exclusiva.

8. De las reglas de las claves ajenas ¿qué estrategia predomina respecto a las otras (las deja sin efecto)?

(A) Restringir.

(B) Propagar.

(C) Anular.

9. Dado el siguiente esquema relacional:

DEPARTAMENTO(código,ubicación,director), director es una clave alternativa

EMPLEADO(dni,nombre,dirección,coddep), coddep es clave ajena a DEPARTAMENTO y no acepta nulos

PROYECTO(código,título,presupuesto,responsable), responsable es clave ajena a DEPARTAMENTO

¿Qué se puede decir acerca de la siguiente afirmación: 'un departamento puede tener como director a un empleado de otro departamento'?

(A) No es cierta.

(B) Es cierta.

(C) Del esquema no se puede obtener esta información.

10. Continuando con el esquema relacional del ejercicio 9:

¿Qué se puede decir acerca de la siguiente afirmación: 'un responsable de proyecto ha de ser un solo departamento y cada departamento sólo puede ser responsable de un proyecto'?

(A) No es cierta.

(B) Es cierta.

(C) Del esquema no se puede obtener esta información.

11. Continuando con el esquema relacional del ejercicio 9:

¿Qué se puede decir acerca de la siguiente afirmación: 'un empleado puede trabajar en varios proyectos de su mismo departamento'?

(A) No es cierta.

(B) Es cierta.

(C) Del esquema no se puede obtener esta información.

12. Continuando con el esquema relacional del ejercicio 9:

¿Qué se puede decir acerca de la siguiente afirmación: 'los empleados que son directores pueden serlo de varios departamentos'?

(A) No es cierta.

(B) Es cierta.

(C) Del esquema no se puede obtener esta información.

13. Continuando con el esquema relacional del ejercicio 9:

¿Qué operación puede violar la integridad referencial?

(A) El borrado de un departamento cuyo director dirige también otro departamento.

(B) El borrado de un director de un departamento sin empleados y con proyectos.

(C) El borrado de un proyecto en el que trabajan empleados de un departamento.

14. Para la clave ajena de EMPLEADO a DEPARTAMENTO, la regla de borrado podría ser:

(A) Restringir o anular, pero no propagar.

(B) Propagar o restringir, pero no anular.

(C) Propagar o anular, pero no restringir.

En la siguiente pregunta todas las respuestas son verdaderas, pero si no se contesta, no cuenta.

15. ¿Qué opinas sobre el nuevo tipo de examen?

(A) Lo prefiero frente al tipo anterior.

(B) Sigo prefiriendo el tipo anterior.

(C) Hasta que no sepa lo que he sacado, no sabré si prefiero este tipo o el anterior.

Descripción de la Base de Datos de Prácticas

La base de datos que se describe a continuación se utiliza en la primera parte del examen y también en la segunda parte, por lo que se deberá conservar esta hoja para ambas partes.

La base de datos consta de las siguientes tablas (es la base de datos de prácticas):

PROVINCIAS(codpro,nombre)

PUEBLOS(codpue,nombre,codpro)

CLIENTES(codcli,nombre,direccion,codpostal,codpue)

ARTICULOS(codart,descrip,precio,stock,stock_min)

FACTURAS(codfac,fecha,codcli,iva,dto)

LINEAS_FAC(codfac,linea,cant,codart,dto,precio)

Las claves primarias son los atributos y conjuntos de atributos que aparecen subrayados; las claves ajenas son las siguientes:

PUEBLOS $\xrightarrow{codpro}$ **PROVINCIAS**

CLIENTES $\xrightarrow{codpue}$ **PUEBLOS**

FACTURAS $\xrightarrow{codcli}$ **CLIENTES**

LINEAS_FAC $\xrightarrow{codfac}$ **FACTURAS**

LINEAS_FAC $\xrightarrow{codart}$ **ARTICULOS**

La información contenida en estas tablas corresponde a una empresa de venta de artículos eléctricos. En **ARTICULOS** se tiene el código y la descripción de cada artículo, su precio de venta actual, el número de unidades del artículo que se tienen en stock y el stock mínimo que se desea mantener. **CLIENTES** contiene los datos de los clientes: código, nombre, dirección, código postal y pueblo al que pertence. **PUEBLOS** contiene los nombres de los pueblos de los clientes, con una referencia a la provincia a la que pertenecen (**PROVINCIAS**). **FACTURAS** contiene las cabeceras de las facturas correspondientes a las compras realizadas por los clientes. Cada factura tiene un código, la fecha en que se realiza, así como el IVA y el descuento que se le aplica. Las líneas de cada factura se encuentran en **LINEAS_FAC**. En cada una de ellas se especifica la cantidad de unidades del artículo que se compra, el precio de venta y el descuento que se aplica sobre dicho precio.