

NOMBRE: _____

TEST**(1.75 puntos)**

Escoger UNA SOLA respuesta para cada pregunta (A, B o C) y escribirla en la casilla correspondiente. Cada respuesta incorrecta resta la mitad del valor de una respuesta correcta. Las preguntas que se dejan sin contestar no restan puntos.

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

- Una diferencia entre los sistemas de ficheros (SF) y los sistemas de bases de datos (SBD) es que ...
 - ... los SF pueden almacenar datos redundantes, mientras que los SBD no pueden.
 - ... los SF utilizan lenguajes navegacionales, mientras que los SBD utilizan lenguajes no navegacionales.
 - ... los SF no pueden almacenar reglas de integridad, mientras que los SBD sí pueden hacerlo.
- Un ejemplo de base de datos es ...
 - ... Oracle.
 - ... las tablas S, P y SP.
 - ... el modelo relacional.
- Algunas de las funciones de los sistemas de gestión de bases de datos son ...
 - ... controlar la seguridad, la concurrencia y la recuperación ante fallos.
 - ... eliminar los datos redundantes y mantener las prestaciones.
 - ... mejorar la productividad de los programadores.
- Se tiene un fichero ordenado sobre el que se ha definido un índice primario sobre el campo clave de ordenación (IP) y un índice secundario sobre otro campo no clave (IS).
 - En ambos índices se puede utilizar búsqueda binaria.
 - En IP sólo se puede utilizar búsqueda lineal y en IS se puede utilizar búsqueda binaria.
 - En IP se puede utilizar búsqueda binaria y en IS sólo se puede utilizar búsqueda lineal.
- Qué ventajas presentan los índices multinivel frente a los índices de un solo nivel?
 - Ninguna, cuantos más niveles, más búsquedas hay que realizar.
 - Que se reduce el número de accesos al hacer búsquedas.
 - Que en cada nivel las entradas del índice son más pequeñas, por lo que ocupa menos espacio.
- ¿Qué es la dispersión?
 - Un modo de organizar los registros dentro de un fichero.
 - Un directorio que se puede añadir a los ficheros en cualquier momento a modo de índice.
 - Una estructura de datos que funciona bien para memoria principal, pero no para disco.
- Ya que los nodos internos de un árbol B+ no guardan punteros a los registros de datos ...
 - ... se ocupa menos espacio que un árbol B equivalente, pero el acceso a los datos es igual de rápido.
 - ... el acceso a los datos es más lento que en un árbol B equivalente porque siempre hay que llegar a las hojas para encontrar el puntero a los datos.
 - ... se puede tener menos niveles que en un árbol B equivalente y, por lo tanto, llegar más rápido a los datos.
- Un árbol B+ al que le caben hasta p punteros a nodos internos en cada bloque ...
 - ... puede tener hasta $\log_2 p$ valores del campo de indexación en cada nodo.
 - ... puede tener hasta $p/2$ valores del campo de indexación en cada nodo.
 - ... puede tener hasta $p-1$ valores del campo de indexación en cada nodo.
- ¿Qué es una instantánea?
 - Una sentencia SQL que se ejecuta muy rápidamente.
 - Una relación (tabla) que se almacena en la base de datos y que es una "foto" de parte de la base de datos en un determinado momento.
 - Una consulta que se almacena en la base de datos y que se ejecuta cuando es accedida por el usuario.

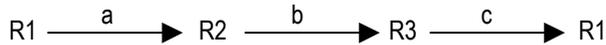
10. ¿Qué tienen en común claves primarias y claves ajenas?

- (A) ... que pueden aceptar nulos si es necesario.
- (B) ... que son identificadores de ocurrencias de entidades.
- (C) ... que son todas claves candidatas de la relación (tabla) en que se encuentran.

11. ¿Cómo se puede conseguir que un sistema de gestión de bases de datos (SGBD) relacional haga que se respete la integridad referencial?

- (A) Evitando la dependencia de datos (lógica-física).
- (B) Eso no lo hacen los SGBD, hay que hacerlo desde los programas de aplicación.
- (C) Basta con definir los atributos que son claves ajenas y a qué tablas hacen referencia.

12. Dado el siguiente diagrama referencial, donde ninguna clave ajena acepta nulos:

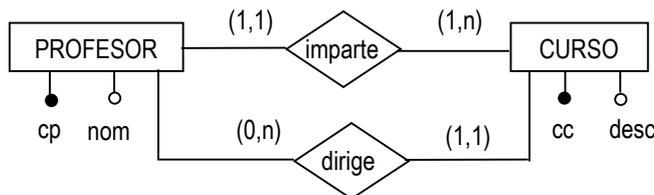


- (A) Es imposible empezar a insertar filas en cualquiera de estas tablas.
- (B) Debe existir una relación redundante, porque hay un camino de R1 a R1.
- (C) Si no aceptan nulos, la regla de borrado para todas ellas deberá ser propagar.

13. Al realizar la operación CLIENTES JOIN PUEBLOS sobre la base de datos de prácticas y teniendo en cuenta que ninguna clave ajena de la base de datos acepta nulos, ...

- (A) ... el resultado tiene tantas filas como pueblos en donde hay clientes.
- (B) ... el resultado tiene tantas filas como clientes hay en la base de datos.
- (C) ... el resultado tiene tantas filas como pueblos hay en la base de datos.

14. Dado el siguiente esquema conceptual ¿qué esquema lógico relacional le corresponde?



- (A) $PROFESOR(\underline{cp}, \underline{cc-imparte}, nom)$, $CURSO(\underline{cc}, \underline{cp-dirige}, desc)$,
 $(CURSO.cc, CURSO.cp-imparte)$ clave ajena a $PROFESOR$
- (B) $PROFESOR(\underline{cp}, nom)$, $CURSO(\underline{cc}, desc)$, $IMPARTE(\underline{cp-dirige}, \underline{cc-imparte})$,
 $IMPARTE.cp-dirige$ clave ajena a $PROFESOR$, $IMPARTE.cc-imparte$ clave ajena a $CURSO$
- (C) $PROFESOR(\underline{cp}, \underline{cc-imparte}, nom)$, $CURSO(\underline{cc}, \underline{cp-dirige}, desc)$,
 $PROFESOR.cc-imparte$ clave ajena a $CURSO$, $CURSO.cp-dirige$ clave ajena a $PROFESOR$

CUESTIÓN

(0.75 puntos)

Se tiene la siguiente relación $MENÚ(\underline{persona}, \underline{clase}, \underline{plato})$ y la siguiente información:

- existen tres clases de platos (primero, segundo y postre),
- cada persona sólo come un plato de cada clase y
- cada plato sólo pertenece a una clase (por ejemplo, la sopa es un primero).

Esto último implica la siguiente dependencia funcional: $plato \rightarrow clase$. ¿Es posible descomponer la relación $MENÚ$ en otras relaciones, de modo que se evite esta dependencia en $MENÚ$ y no se pierda información? Razona la respuesta.

AYUDA: Intenta hacer una descomposición coherente y razona sobre ella.

1. Se desea tener un listado de los nuevos clientes que se han conseguido este año. Para ello se ha escrito la siguiente sentencia SQL que busca los clientes que sólo tienen facturas en este año. ¿Es necesario el uso de DISTINCT? Razona la respuesta. ¿Es posible escribir una sentencia equivalente que no utilice subconsultas? Si es así, escríbela.

```
SELECT DISTINCT c.codcli, c.nombre
FROM facturas f, clientes c
WHERE f.codcli = c.codcli
AND TO_NUMBER(TO_CHAR(f.fecha, 'yyyy')) =
      TO_NUMBER(TO_CHAR(SYSDATE, 'yyyy'))
GROUP BY c.codcli, c.nombre
HAVING NOT EXISTS (SELECT *
                    FROM facturas ff
                    WHERE ff.codcli = c.codcli
                    AND TO_NUMBER(TO_CHAR(ff.fecha, 'yyyy')) <
                        TO_NUMBER(TO_CHAR(SYSDATE, 'yyyy')));
```

2. ¿La siguiente consulta devuelve siempre el total de las ventas (sin IVA ni descuentos) por provincia? Razona la respuesta.

```
SELECT UPPER(NVL(pr.nombre, 'SIN NOMBRE')),
       SUM(l.cant*l.precio) TOTAL_VENTAS
FROM lineas_fac l, facturas f, clientes c, pueblos pu, provincias pr
WHERE l.codfac = f.codfac
AND f.codcli = c.codcli
AND c.codpue = pu.codpue
AND pu.codpro = pr.codpro
GROUP BY pr.codpro, NVL(pr.nombre, 'SIN NOMBRE')
ORDER BY 1;
```

EJERCICIO 1**(2 puntos)**

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

```

PELICULA (codpeli, título, año, género, clasificación, fecha_estreno)
CINTA (codcinta, codpeli, idioma, fecha_entrada, fecha_retiro)
SOCIO (codsocio, nombre, dirección, teléfono)
PRESTAMO (codsocio, codcinta, fecha, pres_dev)

```

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, clasificación (todos los públicos, mayores de 13 años, etc.) y la fecha de estreno. Si la película se ha estrenado en España en los cines, esta es la fecha que se anota; si la película no se ha estrenado en los cines españoles, la fecha de estreno es la fecha en que la primera copia de la película llegó al vídeo club. De una misma película puede haber copias en idiomas distintos, por lo que en cada copia se indica el idioma y, también, la fecha en que llegó al vídeo club y la fecha en que se retiró (admite nulos). De los socios se guarda el código, nombre, dirección y teléfono. 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. Escribir una expresión del álgebra relacional que obtenga los datos de los socios que siempre toman prestadas películas que no se han estrenado en los cines españoles. (1 pto.)

2. ¿Qué consulta realiza la siguiente expresión del álgebra relacional? (0.5 pto.)

```

((PELICULA JOIN CINTA) WHERE fecha_estreno=fecha_entrada) [codpeli,título]
MINUS
((PELICULA JOIN CINTA) WHERE fecha_estreno<>fecha_entrada) [codpeli,título]

```

3. ¿Qué consulta realiza la siguiente expresión del cálculo relacional? Tener en cuenta que la diferencia de dos fechas obtiene el número de días que hay entre ellas. (0.5 pto.)

```

PELICULAX WHERE  $\forall$  SOCIOX  $\exists$  PRESTAMOX  $\exists$  CINTAX
  (IF SOCIOX.codsocio = PRESTAMOX.codsocio AND
    PRESTAMOX.codcinta = CINTAX.codcinta AND
    CINTAX.codpeli = PELICULAX.codpeli
  THEN
    PRESTAMOX.fecha - PELICULAX.fecha_estreno <= 30)

```

EJERCICIO 2**(2.5 puntos)**

(a) ¿A qué consulta responde la siguiente sentencia SQL? Ilustra la respuesta mediante un ejemplo. (0.5 pto.)

```

SELECT c.codcli, c.nombre, DECODE(MIN(iva), 7, ' X ', '') "7%",
      DECODE(MAX(iva), 16, ' X ', '') "16%"
FROM facturas f, clientes c
WHERE f.codcli = c.codcli
AND (iva=7 OR iva=16)
GROUP BY c.codcli, c.nombre;

```

(b) El impuesto en las facturas de clientes se aplica según el impuesto asociado a la provincia en la que se realiza la venta (la provincia del cliente). En el supuesto de que la nueva Ley sobre el Impuesto, puesta en vigor el 1 de enero de 2000, establezca el 4% para los clientes de Ceuta y Melilla, y el 16% para el resto de provincias, se requiere un listado en donde, para cada provincia, se muestre el número de facturas a las que no se ha aplicado el impuesto que le corresponde. Sólo es preciso sacar las provincias en donde haya alguna de estas facturas.

Escribir una sentencia SQL que obtenga el listado que se desea (1 pto.)

(c) Para proponer ofertas especiales a los buenos clientes, se necesita un listado con los datos de aquellos que, en los últimos quince meses, han hecho siempre facturas por un importe superior a 50.000 pesetas (sin tener en cuenta IVA ni descuentos).

Escribir una sentencia SQL que obtenga el listado que se desea (1 pto.)

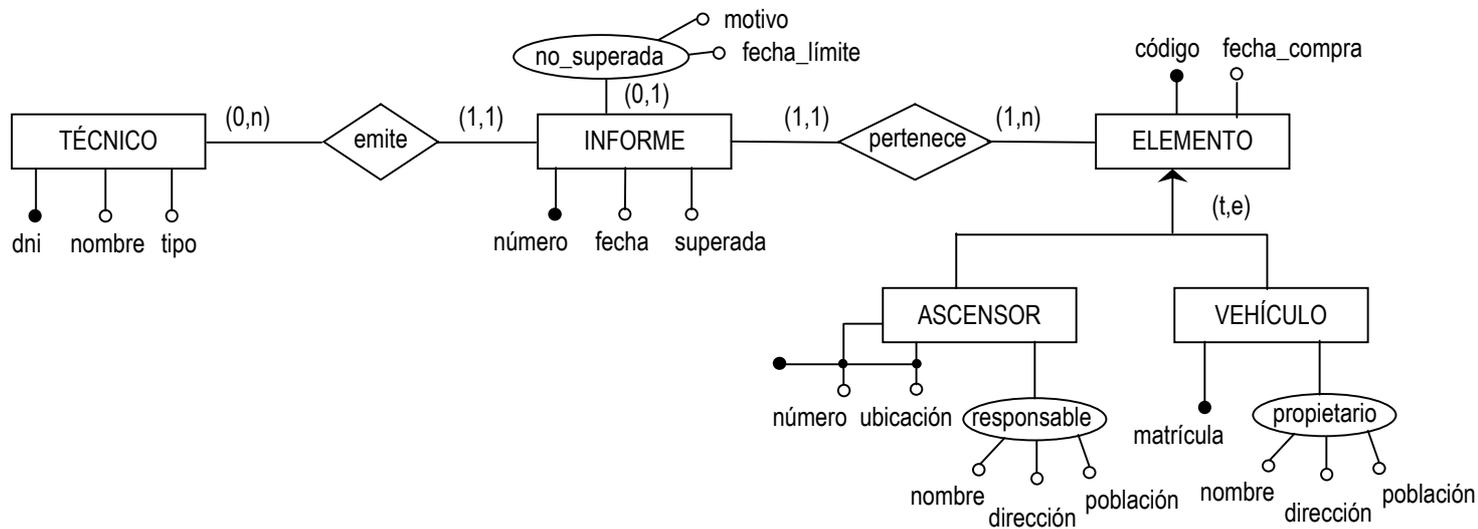
EJERCICIO 3 **(2 puntos)**

La Conselleria de Industria tiene un servicio que se encarga de realizar inspecciones técnicas de ascensores y vehículos. Cuando se realiza una inspección, el técnico que la lleva a cabo emite un informe en el que se especifica si se ha superado la inspección y, en caso contrario, se especifica el motivo por el que no la ha superado y la fecha límite para repetirla. Cuando se repite la inspección, se elabora un nuevo informe.

Se quiere construir una aplicación que, a partir de los informes de los técnicos, genere avisos automáticamente para los propietarios o responsables de los elementos que no han superado la inspección. Estos avisos se generan diez días hábiles antes de la fecha límite y tres días hábiles después, todo esto en caso de que no se haya repetido la inspección.

De los técnicos se quiere conocer el DNI, nombre y tipo (de ascensores o de vehículos). De los informes se quiere conocer el número, la fecha de elaboración y el técnico que lo ha elaborado, si se ha superado o no la inspección y, en este último caso, el motivo y la fecha límite para repetirla. Los elementos que se inspeccionan son, actualmente, ascensores y vehículos, y a todos ellos se les asigna un código que los identifica e interesa su fecha de compra. Los ascensores también se identifican por su número y ubicación (por ejemplo, el ascensor 13 del campus de Riu Sec de la UJI) y tienen un responsable que es quien debe recibir los avisos. Los vehículos también se identifican por su matrícula y tienen un propietario al que se envían los avisos. Tal y como se puede suponer, una misma persona puede ser responsable de varios ascensores (por ejemplo, una misma persona de la oficina técnica de la UJI puede ser la responsable de todos los ascensores del campus) y una misma persona puede ser propietaria de varios vehículos.

El esquema conceptual que se ha obtenido a partir de estos requerimientos es el siguiente:



1. Se pide diseñar un conjunto de relaciones en tercera forma normal que permitan almacenar toda la información descrita. En las relaciones (tablas) hay que señalar los atributos que son clave primaria y los que son clave ajena, determinando si éstas aceptan o no nulos y su comportamiento ante el borrado de tuplas en la tabla que referencian. Si se hace alguna suposición, hay que especificarla.
2. Cada vez que se genera un aviso, debe quedar constancia de ello en la base de datos registrando la fecha del mismo. ¿Dónde deberían situarse las fechas de los avisos en la base de datos de modo que ésta sea completamente flexible frente a los posibles cambios en el número de avisos a enviar?

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 codpro ▷ **PROVINCIAS**

CLIENTES codpue ▷ **PUEBLOS**

FACTURAS codcli ▷ **CLIENTES**

LINEAS_FAC codfac ▷ **FACTURAS**

LINEAS_FAC 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 pertenece. **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.