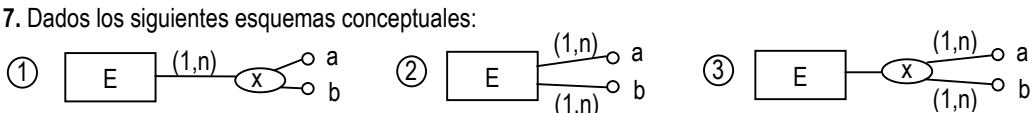


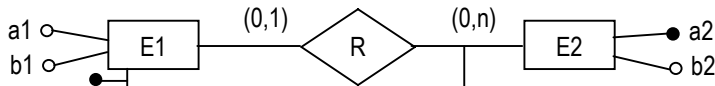
**TEST** **(2.5 puntos)**

1. ¿Cómo se busca un registro por el campo de ordenación en un fichero ordenado?  
*Mediante búsqueda binaria.*
2. ¿Cómo se busca un registro por un campo que no es el de ordenación en un fichero ordenado?  
*Mediante búsqueda lineal.*
3. ¿Qué relación existe entre ficheros e índices?  
*Que un índice es un fichero ordenado.*
4. ¿Cómo se busca un registro por el campo de hash en un fichero que utiliza hashing?  
*Mediante la función de hash.*
5. ¿Cómo se busca un registro por un campo que no es el de hash en un fichero que utiliza hashing?  
*Mediante búsqueda lineal.*
6. Se escogerá entre un índice primario o un índice secundario dependiendo de que ...  
*el campo de indexación sea o no el de ordenación del fichero.*
7. ¿En qué se diferencian los índices secundarios de los índices primarios y los índices de agrupamiento?  
*En que se pueden definir varios sobre un mismo fichero, independientemente del tipo que sea.*
8. Una de las ventajas de los árboles B+ frente a los árboles B es que ...  
*permiten dos modos de acceso: directo y secuencial.*
9. La independencia de datos (lógica-física) aparece con ...  
*los sistemas de bases de datos relacionales.*
10. El que los sistemas de gestión de bases de datos permiten definir subesquemas de la base de datos significa que ...  
*se puede hacer que cada aplicación vea solamente la parte de los datos que necesita.*
11. ¿Por qué han surgido modelos de bases de datos posteriores al modelo relacional?  
*Porque hay aplicaciones que necesitan sistemas de bases de datos cuyos requisitos (tipos de registros, de transacciones, etc.) son diferentes.*
12. En una relación (estructura de datos del modelo relacional) ...  
*no hay tuplas repetidas.*
13. Las claves ajenas necesitan unas reglas de comportamiento ante el borrado y la modificación de las tuplas de la tabla a la que hacen referencia ¿no sería necesario definir una regla ante la inserción de tuplas en dicha tabla?  
*No, porque cuando se inserta una tupla en la tabla, todavía no tiene referencias.*
14. Sean R1 y R2 las siguientes relaciones:  
R1 := LINEAS\_FAC[codfac, linea, cant]                      R2 := LINEAS\_FAC[codfac, cant, dto]  
*R1 tiene tantas tuplas como R2, o más.*
15. Sean R1 y R2 las siguientes relaciones (las claves ajenas en CLIENTES, FACTURAS y LINEAS\_FAC no aceptan nulos):  
R1 := FACTURAS JOIN CLIENTES                      R2 := FACTURAS JOIN LINEAS\_FAC  
*R1 tiene tantas tuplas como FACTURAS.*
16. En el diseño de bases de datos, el esquema lógico que se produce a partir del esquema conceptual ...  
*es menos expresivo puesto que presenta un nivel de abstracción menor.*



El esquema ③ no utiliza bien los conceptos del modelo E/R.

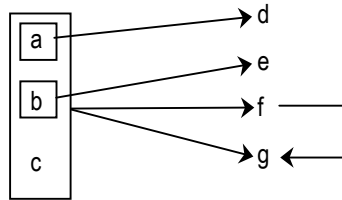
18. Dado el siguiente esquema conceptual:



La cardinalidad mínima con la que E1 participa en R debería ser 1.

19. Una dependencia funcional describe una relación entre atributos de una tabla.  
*Es cierto, ya que si  $x \rightarrow y$  entonces para cada valor de  $x$  hay asociado un solo valor de  $y$ .*

20. Dada la relación  $R(a,b,c,d,e,f,g)$  en la que existen las siguientes dependencias funcionales:



¿de qué modo se debe descomponer  $R$  para evitar posibles anomalías en inserciones, borrados y modificaciones?

$R1(a,b,c,f)$ ,  $R2(a,d)$ ,  $R3(b,e)$ ,  $R4(f,g)$

## CUESTIONES SQL

(1 punto)

1. ¿Obtienen las mismas filas las dos sentencias SQL que se muestran a continuación?

Razona la respuesta. (0.5 puntos)

```

SELECT A.DESCRIP, COUNT(*)
FROM LINEAS_FAC L, FACTURAS F,
     ARTICULOS A
WHERE L.CODFAC = F.CODFAC
AND L.CODART = A.CODART
GROUP BY A.CODART, A.DESCRIP;
  
```

```

SELECT A.DESCRIP, COUNT(*)
FROM LINEAS_FAC L, FACTURAS F,
     ARTICULOS A
WHERE L.CODFAC = F.CODFAC
AND L.CODART = A.CODART
GROUP BY A.DESCRIP;
  
```

Las dos sentencias no obtendrán las mismas filas si existen artículos que tienen la misma descripción. Esto ocurre porque en la sentencia de la derecha no se ha agrupado por el atributo  $A.CODART$  que es quien permite distinguir entre artículos ya que es la clave primaria de la tabla de  $ARTICULOS$ .

2. Sabiendo que la columna **DTO** de la tabla **FACTURAS** acepta nulos ¿se obtendrá el mismo resultado en **COL1** y **COL2** al ejecutar la siguiente sentencia SQL? Razona la respuesta. (0.5 puntos)

```

SELECT AVG(DTO) COL1, SUM(DTO)/COUNT(*) COL2
FROM FACTURAS F;
  
```

Si hay nulos en la columna  $DTO$ ,  $COL1$  y  $COL2$  no tendrán el mismo valor. En  $COL1$  tendremos el valor medio de los descuentos no nulos, mientras que en  $COL2$  tendremos el valor medio de todos los descuentos, considerando que el  $DTO$  nulo equivale a  $DTO$  cero.

## EJERCICIO 1

(2 puntos)

(a.1) Escribir una expresión del álgebra relacional que obtenga los datos de los socios cuyos préstamos corresponden todos a películas realizadas el mismo año en que el socio se dio de alta o en años posteriores (para las fechas utilizar las funciones de SQL).

```

T1 := EXTEND SOCIO ADD TO_CHAR(FECHA_ALTA, 'YYYY') AS AÑO_ALTA
T2 := T1 JOIN PRESTAMO JOIN CINTA JOIN PELICULA
T3 := (T2 WHERE AÑO >= AÑO_ALTA) [CODSOCIO]
T4 := (T2 WHERE AÑO < AÑO_ALTA) [CODSOCIO]
RDO := (T3 MINUS T4) JOIN SOCIO
  
```

(a.2) Escribir una expresión del álgebra relacional que obtenga los datos de las películas en las que han participado actores de al menos dos nacionalidades.

```

T1 := (REPARTO JOIN ACTOR) [CODPELI, NACIONALIDAD]
T2 := SUMMARIZE T1 GROUP BY (CODPELI) ADD COUNT(*) AS NUMERO
T3 := T2 WHERE NUMERO >= 2
RDO := T3 JOIN PELICULA
  
```

(b.1) Escribir una expresión del cálculo relacional que obtenga los datos de los socios cuyos préstamos corresponden todos a películas realizadas el mismo año en que el socio se dio de alta o en años posteriores (para las fechas utilizar las funciones de SQL).

```
SOCIOX WHERE ∀ PRESTAMOX
  (IF PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO THEN
    ∃ PELICULAX ∃ CINTAX
    (PELICULAX.CODPELI=CINTAX.CODPELI AND
    CINTAX.CODCINTA=PRESTAMOX.CODCINTA AND
    PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO AND
    TO_NUMBER(TO_CHAR(SOCIOX.FECHA_ALTA, 'YYYY')) <=PELICULAX.AÑO)
```

(b.2) Escribir una expresión del cálculo relacional que obtenga los datos de las películas en las que han participado actores de al menos dos nacionalidades.

```
PELICULAX WHERE ∃ ACTOR1X ∃ ACTOR2X ∃ REPARTO1X ∃ REPARTO2X
  (PELICULAX.CODPELI=REPARTO1X.CODPELI AND
  REPARTO1X.CODACTOR=ACTOR1X.CODACTOR
  AND PELICULAX.CODPELI=REPARTO2X.CODPELI AND
  REPARTO2X.CODACTOR=ACTOR2X.CODACTOR
  AND ACTOR1X.NACIONALIDAD<>ACTOR2X.NACIONALIDAD)
```

## EJERCICIO 2

(2.5 puntos)

(a) ¿A qué consulta responde la siguiente sentencia SQL? (0,5 pts.)

```
SELECT COUNT(DISTINCT TO_CHAR(F.FECHA, 'ddd'))/365.0*100.0
FROM FACTURAS F
WHERE TO_NUMBER(TO_CHAR(F.FECHA, 'yyyy')) =
  TO_NUMBER(TO_CHAR(SYSDATE, 'yyyy')) -1
AND 5000 < ( SELECT SUM(L.CANT * L.PRECIO)
  FROM LINEAS_FAC L
  WHERE L.CODFAC = F.CODFAC) ;
```

*Porcentaje de días del año pasado en que se hicieron facturas de más de 5000 pesetas.*

(b) (1 pts.) Escribir una sentencia SQL que responda a la siguiente consulta:

Obtener el código y el nombre de las poblaciones a las que se ha servido mayor número de artículos distintos.

```
SELECT PU.CODPUE, PU.NOMBRE
FROM LINEAS_FAC LF, FACTURAS F, CLIENTES C, PUEBLOS PU
WHERE LF.CODFAC=F.CODFAC
AND F.CODCLI=C.CODCLI
AND C.CODPUE=PU.CODPUE
GROUP BY PU.CODPUE, PU.NOMBRE
HAVING COUNT(DISTINCT LF.CODART) >=
  ALL(SELECT COUNT(DISTINCT LF.CODART)
  FROM LINEAS_FAC LF, FACTURAS F, CLIENTES C
  WHERE LF.CODFAC=F.CODFAC
  AND F.CODCLI=C.CODCLI
  GROUP BY PU.CODPUE);
```

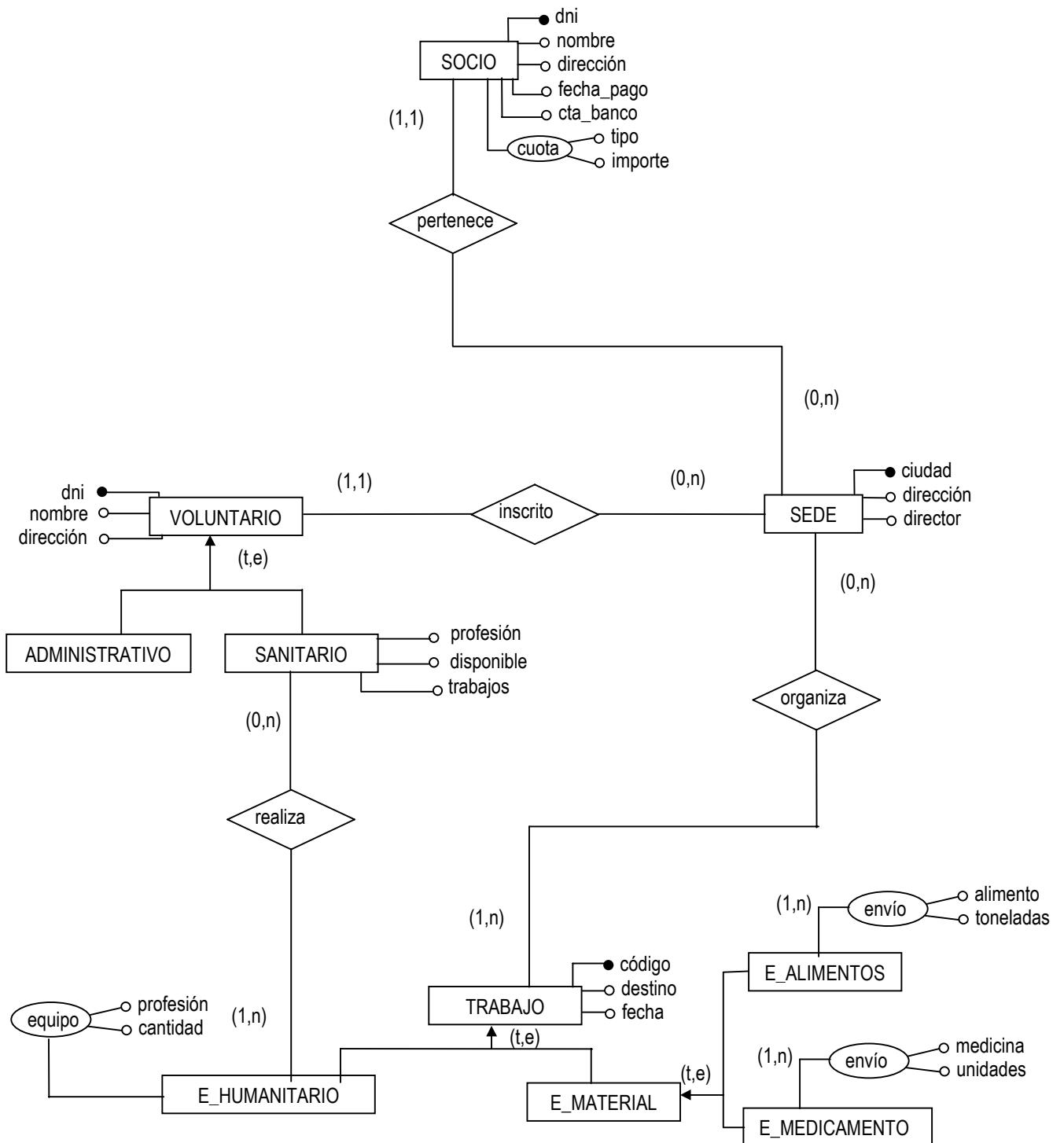
(c) (1 pts.) Escribir una sentencia SQL que responda a la siguiente consulta:

Parejas de clientes y artículos tales que el cliente ha comprado al menos 50 unidades del artículo en cada uno de los meses del año pasado.

```

SELECT C.CODCLI, A.CODART
FROM CLIENTES C, ARTICULOS A
WHERE 12 = (SELECT COUNT(COUNT(*)
FROM LINEAS_FAC L, FACTURAS F
WHERE L.CODFAC=F.CODFAC
AND L.CODART=A.CODART
AND F.CODCLI=C.CODCLI
AND TO_CHAR(F.FECHA, 'YYYY')=TO_CHAR(SYSDATE, 'YYYY') - 1
GROUP BY TO_CHAR(F.FECHA, 'MM')
HAVING SUM(LF.CANT) >=50);
    
```

**EJERCICIO 3 (2 puntos)**



**SEDE**(ciudad,dirección,director)

**CUOTA**(tipo,importe)

**SOCIO**(dni,nombre,dirección,fecha\_pago,cta\_banco,tipo\_cuota,ciudad\_sede)

**SOCIO.tipo\_cuota** es clave ajena a **CUOTA**

**SOCIO.ciudad\_sede** es clave ajena a **SEDE**

**VOLUNTARIO**(dni,nombre,dirección,tipo,ciudad\_sede)

**tipo** ∈ {'administrativo','sanitario'}

**VOLUNTARIO.ciudad\_sede** es clave ajena a **SEDE**

**SANITARIO**(dni,profesión,disponible)

**SANITARIO.dni** es clave ajena a **VOLUNTARIO**

**TRABAJO**(código,destino,fecha,tipo)

**tipo** ∈ {'humanitario','material'}

**E\_ALIMENTOS**(código,alimento,toneladas)

**E\_ALIMENTOS.código** es clave ajena a **TRABAJO**

**E\_MEDICAMENTOS**(código,medicina,unidades)

**E\_MEDICAMENTOS.código** es clave ajena a **TRABAJO**

**E\_HUMANITARIO**(código,dni)

**E\_HUMANITARIO.código** es clave ajena a **TRABAJO**

**E\_HUMANITARIO.dni** es clave ajena a **SANITARIO**

**EQUIPO**(código,profesión,cantidad)

**EQUIPO.código** es clave ajena a **TRABAJO**

**ORGANIZACIÓN**(código,ciudad\_sede)

**ORGANIZACIÓN.código** es clave ajena a **TRABAJO**

**ORGANIZACIÓN.ciudad\_sede** es clave ajena a **SEDE**