

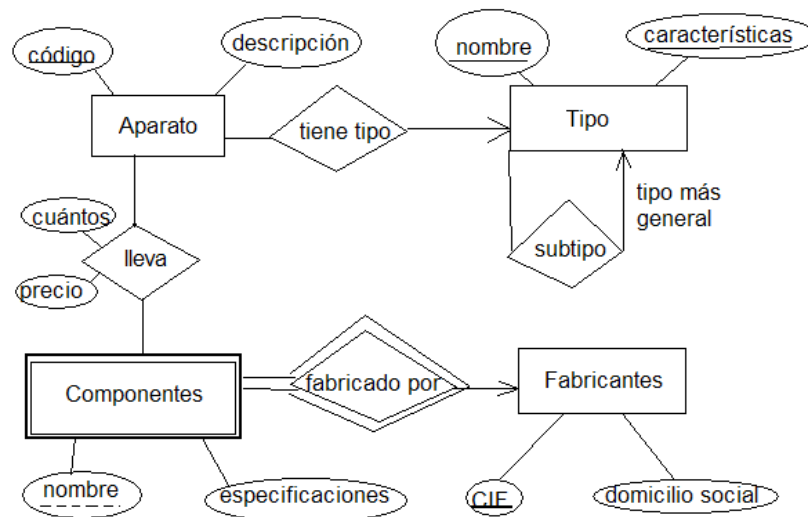
Bases de Datos y Sistemas de la Información

1) [2] Una empresa de aparatos electrónicos desea informatizar sus datos. Escribir el diagrama entidad-relación que responda a la siguiente especificación verbal:

- Cada aparato electrónico viene determinado por un *código* único y una *descripción*. Además cada aparato corresponde a un *tipo* de electrodomésticos (a lo sumo).
- Cada tipo de electrodoméstico (televisor, mp3, lavadora, etc) tiene un *nombre* y unas *características* (un campo de texto). Se supone que no hay dos tipos con el mismo nombre y características. Algunos tipos pueden formar parte de otro tipo más general (“mp3” de “aparato de música”), pero en este caso sólo forman parte de un único tipo.
- Los *componentes* son las piezas que forman el aparato. Vienen dados por un *nombre* (por ejemplo transformador) y unas *especificaciones* (un campo de texto).
- También nos interesa conocer datos de los *fabricantes* de componentes: su *CIF* (único) y su *domicilio social*.
- Cada aparato puede llevar cualquier cantidad de componentes. Interesa saber para cada aparato qué componentes lleva y qué fabricante suministra cada componente. Un aparato puede llevar muchas unidades de un mismo componente (interesa saber cuántas), pero en este caso todas estarán suministradas por el mismo fabricante y con un mismo precio.

Escribir además una instrucción en SQL para crear la tabla asociada a la relación descrita en el apartado e).

Solución 1: Considerando componentes como una entidad débil que queda desambiguada por el fabricante:



La relación *lleva* tendrá es en este caso de muchos a muchos entre componentes y aparato. Pero se debe tener en cuenta que componentes debe incluir en su clave la de fabricantes.

```

create table lleva (
    nombre varchar(10) not null,
    cif varchar(10) not null,
    código varchar(10) not null,
    cuántos int,
    precio number(8,2),
    primary key(nombre,código),
    foreign key (nombre,cif) referentes componentes,
    foreign key (codigo) references aparato );
    
```

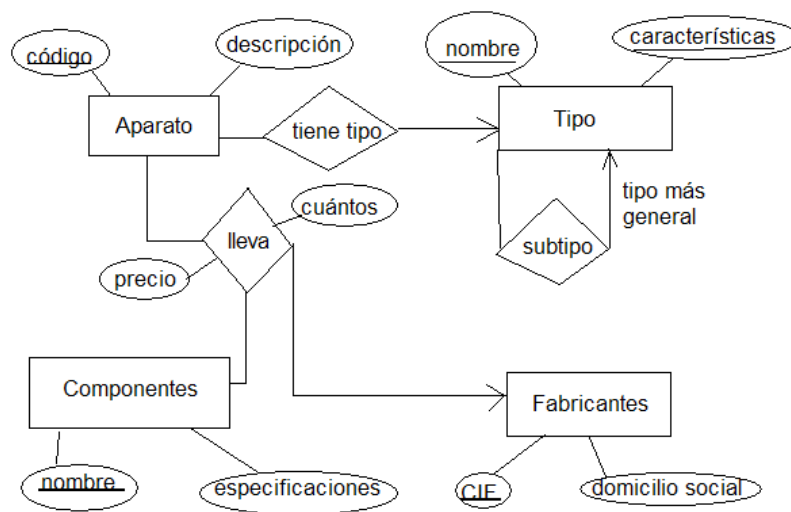
En principio la clave está formada por la de aparato (código) y por la de componentes, que al ser débil incorpora la de fabricantes. Tendríamos así

```

primary key(nombre,cif,código),
    
```

Pero según dice el enunciado, fijado un componente (nombre) y un aparato (código) hay un único fabricante (CIF), es por ello que el CIF se ha excluido de la clave primaria.

Solución 2: En el examen se dijo que se podía considerar el nombre del componente como clave. En este caso:



En este caso *lleva* es una relación entre 3 entidades. Para determinar la clave se tiene en cuenta que por la parte de fabricantes tenemos una cardinalidad 1, por lo que no es necesario incluir CIF como parte de la clave:

```
create table lleva (
  nombre varchar(10) not null,
  cif varchar(10) not null,
  código varchar(10) not null,
  cuántos int,
  precio number(8,2),
  primary key (nombre,código),
  foreign key (nombre) references componentes,
  foreign key (CIF) references fabricantes,
  foreign key (codigo) references aparato );
```

2) [5 puntos] Don Bertoldo Pomposo, administrador de fincas, tiene un sistema de Bases de Datos para llevar los datos de su negocio. Las tablas relacionales que forman la base de datos son:

- propietarios (DNI, nombreYapellidos) : datos de los propietarios de los pisos. Cada propietario puede tener varios pisos, pero cada piso tiene un único propietario.
- pisos (DNIProp, nombreUrb, piso, letra, cantidad) : Indica que el propietario con DNIProp (que debe existir en *propietarios*) tiene un piso en la urbanización nombreUrb, piso y letra, y que debe la *cantidad* indicada. Esta *cantidad* es el valor acumulado total de lo que se debe, que será 0 si se está al corriente en los pagos o un entero positivo en caso contrario.

- a) [1] Escribir una consulta SQL para obtener un listado de urbanizaciones, ordenadas por nombre con el número total de pisos morosos (que deben una cantidad mayor que 0) que contiene cada una. Debe quedar algo del estilo:

Urbanización	Morosos
Alcornocales	9
San Chicharro	0
Zarzales del Duque	7

```
select nombreUrb 'Urbanización', sum(if(cantidad>0,1,0)) 'Morosos'
  from pisos
 group by nombreUrb
 order by nombreUrb;
```

- b) [1] Escribir una consulta SQL para saber el nombre y apellidos de propietarios que tienen piso sólo en la urbanización *Los Mendrugas*.

```
Select nombreyapellidos
from propietarios P, pisos D
where D.DNIProp=P.DNI and nombreUrb='Los Mendrugas' and
      not exists (select *
                  from pisos D2
                  where D2.DNIProp = P.DNI and
                    nombreUrb<>'Los mendrugos');
```

- c) [1] Escribir una consulta SQL para obtener el DNI del propietario que más dinero debe en total (entre todos los pisos que posee). Si hay varios que deben el máximo la consulta debe mostrarlos todos.

```
create view DNITotal(DNI,Total) as
(select DNIProp, sum(cantidad)
 from pisos
 group by DNIProp);

select DNI
from DNITotal
where total >=ALL (select total from DNITotal);
```

- d) [1] Hay que pasar una derrama de 500 euros a los pisos de la urbanización 'Balcón del Fangal'. Don Bertoldo, que además de administrador es propietario de varios pisos en esta urbanización, decide pasar la cuota a todos los vecinos excepto a sí mismo. Escribir una sentencia SQL para que la cantidad que deben los pisos de dicha urbanización se incremente en 500 excepto para aquellos pisos propiedad de Don Bertoldo. (Nota: sólo hay un 'Bertoldo Pomposo' en propietarios).

```
update pisos
set cantidad = cantidad+500
where nombreUrb='Balcón del Fangal' and DNIProp <>
      (select DNI from propietarios
       where nombreyapellidos='Bertoldo Pomposo')
```

- e) [1] Escribir la consulta b) en algebra relacional y cálculo relacional de tuplas.

Álgebra relacional:

```
EnMendrugas    ←  $\Pi_{DNIProp} (\sigma_{nombreUrb='Los Mendrugas'}(pisos) )$ 
EnOtra         ←  $\Pi_{DNIProp} (\sigma_{nombreUrb<>'Los Mendrugas'}(pisos) )$ 
SoloMendrugas  ←  $\rho_{DNI} (EnMendrugas \setminus EnOtra)$ 
Resultado      ←  $\rho_{nombreyapellidos} (propietarios \times SoloMendrugas)$ 

(|x| = unión natural)
```

Cálculo de tuplas:

```
{ t |  $\exists u (u \in propietarios \wedge t[nombreyapellidos] = u[nombreyapellidos] \wedge$ 
 $\exists v (v \in pisos \wedge v[DNIProp] = u[DNI] \wedge v[nombreUrb]='Los Mendrugas' \wedge$ 
 $\neg \exists w (w \in pisos \wedge w[DNIProp] = u[DNI] \wedge w[nombreUrb]<>'Los Mendrugas'))$  }
```

- 3) [3 puntos] Una red hotelera mantiene un sistema centralizado de reservas de acuerdo a las siguientes especificaciones:

- Cada habitación (H) tiene asignado un código que indica de forma única el hotel (O) y el tipo de habitación (T)
- El precio de la habitación (P) depende del hotel y del tipo de habitación.
- El código de reserva (R) está determinado por la habitación y la fecha (F).

- Cada código de reserva determina la información del cliente (C), la habitación y la fecha.

Se pide :

- [0.5] Definir el esquema R y el conjunto de dependencias funcionales F para representar el enunciado. (0,25 puntos)
- [0.5] Calcular un recubrimiento mínimo de F.
- [0.5] Calcular todas las claves candidatas.
- [1] Realizar una descomposición en 3FN que conserve las dependencias funcionales y la información.
- [0.5] Indicar justificadamente si los subesquemas obtenidos están en FNBC.

Solucion

- $R(H, O, T, P, F, R, C)$ y $DF = \{H \rightarrow OT, OT \rightarrow P, HF \rightarrow R, R \rightarrow CHF\}$
- Hallamos la cobertura mínima:
 - Reducimos los lados derechos y buscamos atributos redundantes en los izquierdos
 - No hay d.f redundantes
$$Fc = \{H \rightarrow O, H \rightarrow T, OT \rightarrow P, HF \rightarrow R, R \rightarrow C, R \rightarrow H, R \rightarrow F\}$$

- Claves candidatas R y HF, ya que:
 Todos los atributos están en los consecuentes.
 Se descartan P y C que no están en los antecedentes. Probamos con el resto:
 $(H)^+_{F=} \{HOTP\}$
 $(O)^+_{F=} \{O\}$
 $(T)^+_{F=} \{T\}$
 $(F)^+_{F=} \{F\}$
 $(R)^+_{F=} \{HOTPFRC\} \leftarrow$ Clave candidata
 $(HF)^+_{F=} \{HOTPFRC\} \leftarrow$ Clave candidata
 $(HO)^+_{F=}, (HT)^+_{F=}, (HP)^+_{F=}$ se descartan porque O, T y P están en el cierre de H
 $(HC)^+_{F=} \{HOTPC\}$

Se descartan las combinaciones de 3 o más elementos porque no producen nuevas claves candidatas o incluyen alguna que incluya a R o a HF.

- La relación no está en 3FN.
 $H \rightarrow OT$ y $OT \rightarrow P$ invalidan que lo esté ya que ni H ni OT son superclaves y ni OT ni P forman parte de una clave candidata (HR o R)

Descomposición en 3FN usando F agrupada que es el DF original:

$R = (H, O, T, P, F, R, C)$ y $Fc = \{H \rightarrow O, H \rightarrow T, OT \rightarrow P, HF \rightarrow R, R \rightarrow C, R \rightarrow H, R \rightarrow F\}$

Descomponemos en:

R1(H, O, T)	F1 = {H → O, H → T}
R2(O, T, P)	F2 = {OT → P}
R3(H, F, R)	F3 = {HF → R, R → H, R → F}
R4(R, C, H, F)	F4 = {HF → R, R → C, R → H, R → F}

Alguna de las claves candidatas iniciales está incluida ya en alguna descomposición de las relaciones.

- R1, R2, R3 y R4 están además en FNBC, ya que en todas ellas el antecedente es superclave. Por ejemplo en R4, $HF \rightarrow R$, se tiene HF es superclave de R4, y lo mismo sucede con $R \rightarrow C$, $R \rightarrow H$, $R \rightarrow F$, ya que R también es superclave de R4.