

Apellidos: _____
Nombre: _____
Fecha: _____ Grupo: _____

Supuesto de datos

Crea un esquema de base de datos que soporte los siguientes requisitos:

- Se identifica a una persona por el DNI, que será único. Se conoce de ella su nombre, y el municipio donde está empadronada en el momento actual.
- Una persona puede cambiar de municipio. De un municipio solo se conoce su nombre.
- De cada persona puede saberse si está soltera o casada. Si está casada, es necesario saber quién es su cónyuge.
- Una persona puede casarse y divorciarse varias veces. Una persona solo puede tener un matrimonio a la vez.

Se recomienda:

- Que el esquema de base de datos siga la 3ª forma normal y no tenga anomalías de inserción, modificación o borrado (redundancias de datos)
- Leer todos los ejercicios, para asegurarse que el modelo de datos los soporta.

Si hay dudas...

- Consulta todas las dudas que tengas al profesor (hace de *cliente* de la base de datos)
- Pide las hojas de borrador que te hagan falta

Ejercicio 1 : Vistas principales

No es importante qué tablas se creen para almacenar la información, pero para poder corregir el apartado de procedimientos deben existir las siguientes vistas:

- V_PERSONAS (dni, nombrepersona, nombremunicipio)
 - Lista **sin duplicados** de las personas con los municipios donde están empadronados en el momento actual.
 - Si no están empadronados, el campo municipio tendrá el valor NULL
- V_SOLTEROS (dni)
 - Lista de las personas que no están casadas, **sin duplicados**.

Atributo	Tipo	
nombremunicipio	varchar(1024)	El nombre del municipio
nombrepersona	varchar(1024)	El nombre de la persona
dni	varchar(1024)	El dni de la persona

```
1 CREATE OR REPLACE VIEW V_PERSONAS(dni,nombrepersona,nobremunicipio) as
2 select ...;
3
4 CREATE OR REPLACE VIEW V_SOLTEROS(dni) as
5 select ...;
```

Listado 1: Creación de vistas para el profesor

Ejercicio 2 : Procedimientos para manipulación de datos

- Creación de personas (1 punto): CREA_PERSONA(dni,nombrepersona)
 - Crea una nueva persona en la base de datos. Si el DNI ya existe lanzará una excepción con el texto DNIYAEXISTE
 - Inicialmente, la persona no estará empadronada en ningún municipio
 - Necesita la vista V_PERSONAS para poder corregirse
- Empadronamiento de personas en municipios (2 punto): EMPADRONA(dni,nobremunicipio)
 - Asigna esa persona a ese municipio. Si el DNI no existe, se lanzará una excepción con el texto DNINOEXISTE
 - Necesita la vista V_PERSONAS para poder corregirse
- Matrimonios (3 puntos): MATRIMONIO(dni1,dni2)
 - Estas personas se apuntan en el registro de matrimonios. **Nota:** El orden de los dni no es importante para el matrimonio. Que *Pepe* se case con *Lola* es lo mismo que si *Lola* se casa con *Pepe*.
 - Si no existe alguna persona se lanzará DNINOEXISTE
 - Si alguna de las personas ya formaba parte de un matrimonio, se lanzará una excepción con el texto POLIGAMIA
 - Si una persona se quiere casar consigo misma se lanzará también POLIGAMIA
 - No es importante el municipio donde están empadronadas las personas.
 - Necesita la vista V_SOLTEROS para poder corregirse

```
1 create or replace procedure CREA_PERSONA(dni varchar, nombrepersona varchar)
2 as
3 begin
4     ...
5     -- EJEMPLO DE RAISE
6     RAISE_APPLICATION_ERROR(-20001, 'DNIYAEXISTE');
7     ...
8 end;
9
10 create or replace procedure EMPADRONA(dni varchar, nombremunicipio varchar)
11 as
12 begin
13     ...
14     -- EJEMPLO DE RAISE
15     RAISE_APPLICATION_ERROR(-20001, 'DNINOEXISTE');
16     ...
17 end;
18
19 create or replace procedure MATRIMONIO(dni1 varchar, dni2 varchar)
20 as
21 begin
22     ...
23     -- EJEMPLO DE RAISE
24     RAISE_APPLICATION_ERROR(-20001, 'POLIGAMIA');
25     ...
26 end;
```

Listado 2: Procedimientos

Ejercicio 3 : Vistas adicionales

- V_MUNICIPIOS(nombremunicipio) (1 punto)
 - Lista de los municipios donde reside al menos una persona. No debe haber filas duplicadas.
- V_HABITANTES(nombremunicipio, numerohabitantes) (3 puntos)
 - Lista de los municipios donde reside al menos una persona, con el número de habitantes en el municipio. No debe haber filas duplicadas.

```
1
2 CREATE OR REPLACE VIEW V_MUNICIPIOS(nombremunicipio) as
3 select ...;
4
5 CREATE OR REPLACE VIEW V_HABITANTES(nombremunicipio, numerohabitantes) as
6 select ...;
```

Listado 3: Vistas derivadas

Normas de entrega

Las tablas, vistas y procedimientos se crearán en la base de datos del profesor, en el usuario que se indique para cada alumno. El trabajo está entregado si está grabado en la base de datos.

Corrección automática

El examen se evaluará de forma automática, con pruebas unitarias. Dichas pruebas no funcionarán si las vistas o procedimientos no siguen los nombres indicados, con los parámetros indicados.