PROYECTO: Pueblos de España

ACTIVIDAD 2

Responde a las siguientes preguntas, y como el resto de ejercicios puedes usar este documento (a partir de ahora) como guia, donde las respuestas las instruciones deben estar en modo texto (no imagen), y las respuestas como capturas en modo imagen.

Conceptos teóricos.

1. Definición de BBDD

- *Creación de una BBDD*

```
CREATE DATABASE <base_datos>;
```

```
CREATE DATABASE BDBiblioteca;
```

- *Borrado de una BBDD*

```
DROP DATABASE <base_datos>;
```

```
DROP DATABASE BDBiblioteca;
```

2. Integridad referencial

La existencia de tablas relacionadas mediante clave ajena ocasiona una problemática que atañea la integridad de la información implicada. ¿Es posible borrar el registro correspondiente a un socio de la biblioteca que tiene préstamos registrados? ¿Se puede cambiar la clave primaria de una editorial relacionada con libros? Diseñar una base de datos también implica tomar decisionesde integridad referencial. Hay cuatro enfoques básicos:

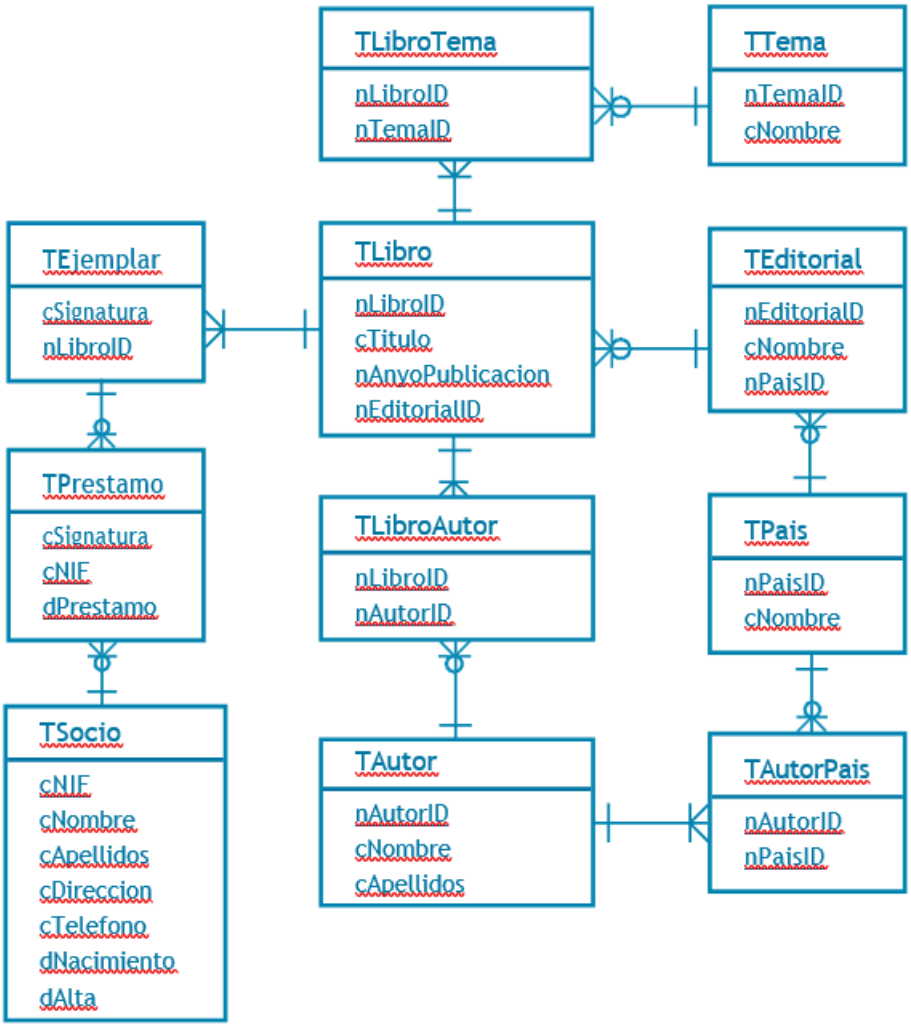
- ***Prohibir la operación.*** Es la decisión más restrictiva. Impide el borrado o modificación de registros que tengan coincidencias por clave ajena en otra u otras tablas. Se puede implementar de modo que, sencillamente, la operación no se lleve a cabo, o bien que, además, genere un error.
- ***Transmisión en cascada.*** Se arrastra la modificación o el borrado a las tablas con registros relacionados (es decir, si se eliminase en TAutor el registro

correspondiente a un autor con libros también se eliminarían automáticamente los registros de TLibroAutor que vinculan TAutor con TLibro).

- ***Puesta a nulo.*** Se permite la operación de modificación o borrado, pero poniendo a NULL los valores de las claves ajenas correspondientes (se podría borrar en TEditorial una editorial con libros, pero en los registros correspondientes de TLibro el campo nEditorialID quedaría a NULL). En este caso, las claves ajenas no pueden contar con la restricción NOT NULL.

  Esta alternativa no es recomendable, ya que contradice una regla de eficiencia básica:evitar la proliferación de valores nulos.

- ***Uso de valor por defecto.*** Se permite la operación de modificación o borrado, pero asignando a las claves ajenas correspondientes su valor por defecto. Implica que ese valor esté definido.



3. Definición de Tablas

- *Creación de una tabla*

```
CREATE TABLE <tabla> (
    <campo1> <tipo>[(<longitud>)] [NOT NULL][UNIQUE][PRIMARY KEY]
    [CHECK <condición>] [DEFAULT <valor>][, [
    <campo2> <tipo>[(<longitud>)] [NOT NULL][UNIQUE][PRIMARY KEY]
    CHECK <condición>][DEFAULT <valor>], ]
    ...
    <campoN> <tipo>[(<longitud>)] [NOT NULL][UNIQUE][PRIMARY KEY]
    [CHECK
    <condición>][DEFAULT
    <valor>]][, PRIMARY KEY
    (<lista_campos>)]
    [, FOREIGN KEY (<lista_campos>) REFERENCES <tabla>
    (<lista_campos>)[{ON UPDATE [NO ACTION|SET DEFAULT|SET
    NULL|CASCADE]
    [ON DELETE [NO ACTION|SET DEFAULT|SET NULL|CASCADE]]
        }|
        {ON DELETE [NO ACTION|SET DEFAULT|SET
        NULL|CASCADE] [ON UPDATE [NO ACTION|SET
        DEFAULT|SET NULL|CASCADE]]
        }]]
    [, FOREIGN KEY...]
    ...
    [, FOREIGN KEY...]
);
```

```
CREATE TABLE TPais (
    nPaisID NUMERIC(3,0) NOT NULL UNIQUE PRIMARY KEY,
    cNombre VARCHAR(30) NOT NULL
);
```

Ejemplos

```
CREATE TABLE TEditorial (
    nEditorialID IDENTITY PRIMARY KEY,cNombre VARCHAR(40)
    NOT NULL, nPaisID NUMERIC(3,0) DEFAULT 724
    , FOREIGN KEY (nPaisID) REFERENCES TPais (nPaisID)
    ON UPDATE CASCADE
    ON DELETE NO ACTION
);
```

A nEditorialID se le ha asignado el tipo de datos IDENTITY. De ese modo, el SGBDR será responsable de asignarle valores válidos, únicos y correlativos. En cuanto a nPaisID, con función de clave ajena, en el caso de que se intente borrar un nPaisID de TPais, se impedirá dicho borrado; pero, en cambio, se permitirá la actualización, que se propagará a TEditorial en cascada

```
CREATE TABLE TSocio (
     cNIF CHAR(9) NOT NULL UNIQUE PRIMARY KEY,
     cNombre VARCHAR(30) NOT NULL,
     cApellidos VARCHAR(60) NOT NULL,
     cDireccion VARCHAR(100),
     cTelefono CHAR(12) NOT NULL,
     dNacimiento DATE NOT NULL,
     dAlta DATE NOT NULL CHECK dAlta >= "2003-09-01"
);
```

Mediante la cláusula `CHECK` se ha indicado una condición que debe cumplir el campo dAlta: que la fecha de alta de todo socio sea igual o superioral 1 de septiembre de 2003.

- ***Modificación de una tabla***. La sentencia `ALTER TABLE` permite añadir un campo a una tabla, modificar sus características o borrarlo.

  – Añadir un campo:

```
ALTER
TABLE
<tabla>
          ADD <campo> <tipo>[(<longitud>)][NOT
               NULL][UNIQUE][PRIMARY KEY]
 [CHECK <condición>][DEFAULT <valor>];
```

```
ALTER TABLE TSocio
     ADD cEmail VARCHAR(256);
```

Modificar un campo:

```
ALTER TABLE <tabla>
     ALTER <campo> {TYPE <tipo>[(<longitud>)]|DROP DEFAULT};
```

```
ALTER TABLE TSocio
     ALTER cNombre VARCHAR(40);
```

La cláusula `TYPE` permite variar el tipo de datos o su longitud. `DROP DEFAULT` elimina el valor por defecto.

– Borrar un campo:

```
ALTER TABLE <tabla>
     DROP <campo>;
```

```
ALTER TABLE TLibro
     DROP nAnyoPublicacion;
```

- Eliminación de una tabla.

```
DROP TABLE <tabla>;
```

```
DROP TABLE TTema;
```

4. Definición de tipos de datos

El usuario de una base de datos puede crear tipos a medida que podrá reutilizar tantas vecescomo desee.

```
CREATE TYPE <nuevo_tipo > AS <tipo_estándar> [(<longitud>)];
```

Base de Datos

```
CREATE TYPE typNombre AS
    VARCHAR(30);
```

```
CREATE TYPE typCodPostal AS
    CHAR(5);
```

```
CREATE TABLE TCliente (
     cNIF CHAR(9) NOT NULL UNIQUE PRIMARY KEY,
     cNombre typNombre NOT NULL,
     cDireccion VARCHAR(100),
     cCodPostal typCodPostal,
     cTelefono CHAR(12) NOT NULL
);
```

1. Crea una tabla llamada málaga, cuyos campos sean nombre de municipio y población.

```
CREATE TABLE `municipios`.`Malaga` (
  `nombre_muni` VARCHAR(80) NOT NULL,
  `Poblacion` DECIMAL(9,0) NOT NULL,
  PRIMARY KEY (`nombre_muni`));
```

2. Introduce los datos de la provincia de málaga en la nueva tabla málaga.

```
INSERT INTO `malaga`(`nombre_muni`, `Poblacion`)
SELECT `municipios`.`NOMBRE_ACTUAL`, `municipios`.`POBLACION_MUNI`
FROM municipios
WHERE provincia="Malaga";
```

3. Borra todos los datos de la tabla Málaga.

```
DELETE  FROM malaga;
```

4. Como tenemos el use sobre la BBDD, volvemos a cargar los datos de la provincia.

```
INSERT INTO `malaga`(`nombre_muni`, `Poblacion`)
SELECT nombre_actual,poblacion_muni
FROM municipios;
```

5. Saca listado por orden descendente de población.
6. Saca listado por orden descendente de población de los municipios:
    a. menores de 1000
    b. entre 1000 y 5000.
    c. Entre 5000 y 10000
    d. Entre 10000 y 20000
    e. Entre 20000 y 50000
    f. Mayores de 50000
7. Vamos a incluir el campo no nulo de superficie. ¿tienes algun problema?
8. Vamos a incluir primero el campo superficie, pero con valores nulos.
9. Introduce los valores de superficie en KM2
10. Cambia a no nulo el campo superficie.

```
ALTER TABLE `municipios`.`municipios`
CHANGE COLUMN `SUPERFICIE` `SUPERFICIE` DECIMAL(10,4) NOT NULL ;
```

11. Añade en la tabla el campo altitud con valor 1 por defecto.
12. Inserta los datos de altitud de cada uno de los municipios
13. Lista los 10 municipios de Málaga con menor altitud
14. Lista los 10 municipios de Málaga con mayor altitud
15. ¿Cuál es la altitud media de la provincia?
16. Lista los pueblos de la provincia que esten por debajo de los 100 m de altitud.
17. Lista los pueblos de la provincia que esten por encima de los 100 de altitud
18. Lista los pueblos entre 10m y 50m de altitud.
19. ¿Cuál es la media de población de los pueblos de la provincia?
20. Lista los 5 pueblos de mayor extensión de la provincia de málaga.