



PROYECTO: Pueblos de España

ACTIVIDAD 2.TEORIA

Responde a las siguientes preguntas, y como el resto de ejercicios puedes usar este documento (a partir de ahora) como guía, donde las respuestas las instrucciones 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 atañe a 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 decisiones de integridad referencial. Hay cuatro enfoques básicos:

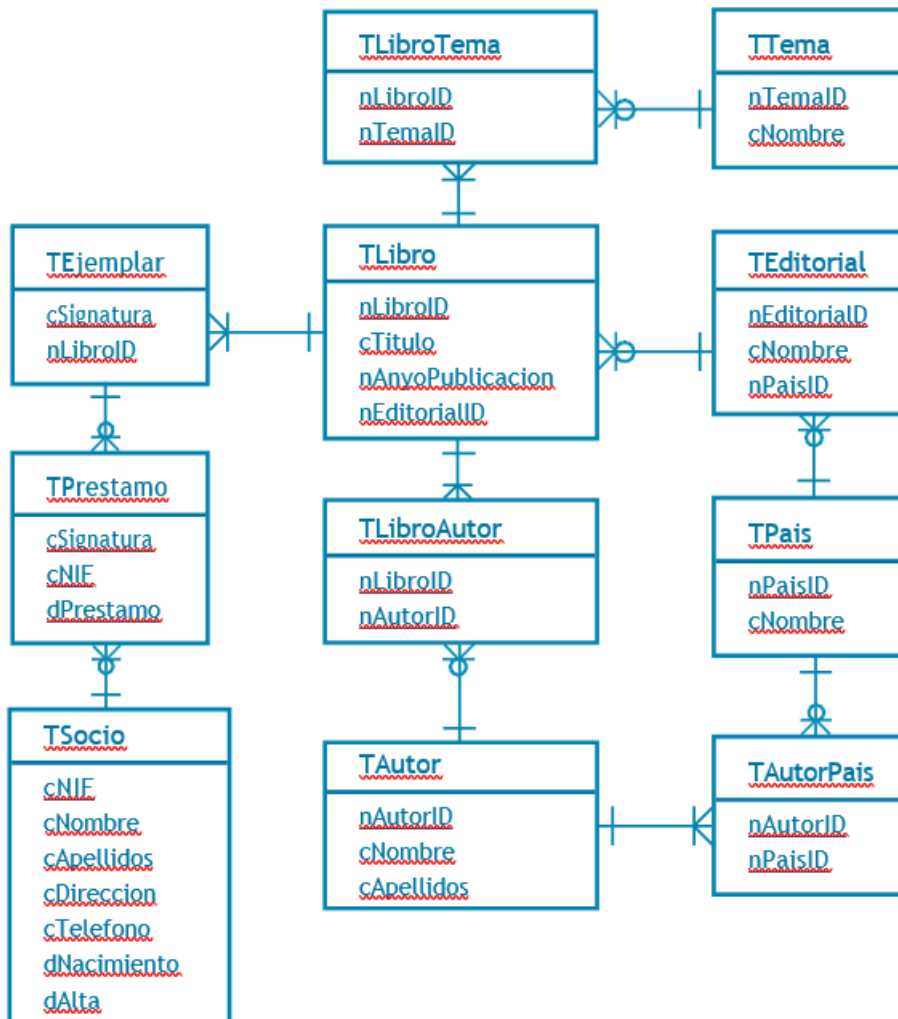
- **Prohibir la operación.** Es la decisión más restrictiva. Impide el borrado o modificación de registros que tengan coincidencias por clave ajena en otra u otras tablas. Se puede implementar de modo que, sencillamente, la operación no se lleve a cabo, o bien que, además, genere un error.
 - **Transmisión 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.





Base de Datos



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



Base de Datos



```
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 superior al 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};
```



Base de Datos



```
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 veces como 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  
);
```



ACTIVIDAD 2. PRÁCTICA

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.
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.
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.
-



Base de Datos



12. Inserta los datos de altitud de cada uno de los municipios

```
UPDATE `municipios`.malaga t1
SET t1.`altitud` =
    (SELECT ALTITUD
     from `pueblos` m2
     where t1.nombre_muni=m2.NOMBRE_ACTUAL)
where t1.altitud=1;
```

13. Lista los 10 municipios de Málaga con menor altitud

14. Lista la población de los 10 municipios de Málaga con menor altitud

15. Calcula altitud media de los 5 municipios de menor altitud

```
SELECT AVG(t.altitud)
from (select nombre_muni, altitud from malaga order by 2 limit 5) as t;
```

Si ejecutamos

```
select nombre_muni, altitud from malaga order by 2 limit 5
```

Obtenemos este resultado

nombre_muni	altitud
Fuengirola	6
Málaga	8
Rincón de la Victoria	12
Estepona	14
Marbella	18
NULL	NULL

Con **AS t**, estamos etiquetando la tabla resultante con nombre **t**, por lo que tanto en el **SELECT** como en el **WHERE** debemos usar el nombre de la “nueva tabla” que hemos creado.

16. Indica la población total de los 3 municipios de menor altitud

17. Lista los 10 municipios de Málaga con mayor altitud

18. Lista la población de los 12 municipios de Málaga con mayor altitud

19. Calcula altitud media de los 10 municipios de mayor altitud

20. Indica la población total de los 10 municipios de mayor altitud

21. ¿Cuál es la altitud media de la provincia?

22. ¿Cuál es la media poblacional de la provincia?



Base de Datos



23. Lista los pueblos de la provincia que esten por debajo de los 100 m de altitud.
24. Lista los pueblos de la provincia que esten por encima de los 100 de altitud
25. Lista los pueblos entre 10m y 50m de altitud.
26. ¿Cuál es la media de población de los pueblos de la provincia?
27. Lista los 5 pueblos de mayor extensión de la provincia de Málaga.
28. Mostrar la lista de Bases de datos.

```
show databases;
```

29. Muestra las tablas de la BBDD

```
USE municipios;  
SHOW TABLES;
```

30. Haz una Descripción de la tabla municipios.

```
DESCRIBE municipios.malaga;
```

31. Haz una Descripción de la tabla malaga
32. Modifica el campo población para que verifique que todos los municipios tienen más de 1 habitante.

Debemos de modificar la tabla con CHECK

https://www.w3schools.com/sql/sql_check.asp

Como la tabla ya esta creada, tendremos que utilizar la funcion **ALTER TABLE**

Ejemplo

```
ALTER TABLE Persons  
ADD CHECK (Age>=18);
```

La otra opción es crear reglas de restricción, como por ejemplo

```
ALTER TABLE malaga  
ADD CONSTRAINT CHK_Altitud CHECK (Altitud>=1 AND Altitud<=4000),  
ADD CONSTRAINT CHK_Poblacion CHECK (Poblacion>=0 AND Poblacion<=99999999);
```

33. Modifica el campo altitud para que verifique que todos los municipios estan entre 1 y 4000 m
 34. Lista todos los pueblos que no comiencen ni por la B, ni la A, ni la M
 35. Cuantos pueblos hay que no comiencen por la M ni por la A
 36. Haz un listado con los 10 pueblos con mayor densidad poblacional de la provincia.
-



Base de Datos



37. ¿Cuáles son los 5 pueblos con menor densidad poblacional de la provincia?
 38. ¿Cuántos pueblos comienzan por M?
 39. ¿Cuántos pueblos comienzan por A?
 40. Pon en la misma tabla, el numero de pueblos que comienzan por A, B, C, M, T y el número de cada uno.
-