

UNIDAD DE TRABAJO N°3/4.- Diseño físico Relacional



Diseño físico Relacional

La mayor parte de los sistemas actuales de BB.DD. son relacionales.

En 1979 surge el primer SGBDR comercial basado en SQL (ORACLE)

LENGUAJE: SQL

El SQL actúa como interfase entre un usuario y el sistema. El SQL es un lenguaje con un conjunto de instrucciones para definir, manejar y controlar los datos de una BD relacional.

El SQL es un lenguaje no procedimental, es decir, los usuarios especifican el QUE, no el COMO,

Tipos de sublenguajes que incorpora

El DDL (Data Definition Language, Lenguaje de Definición de Datos): CREATE(DATABASE, TABLE, INDEX, VIEW, USER), ALTER, DROP..

El DML (Data Manipulation Language, Lenguaje de Manipulación de Datos) (INSERT, DELETE, UPDATE, SELECT)

El DCL (Data Control Language, Lenguaje de Control de Datos), GRANT, REVOKE



SINTAXIS DE CONEXIÓN.

MySQL: cliente de MySQL-server:

`mysql [options] [database]`

options:

`help` visualiza la ayuda

`-p|password password` con la que se conectarnos

`-P|port puerto TCPIP` remoto al que se conecta

`-h host=numero nombre host` o

`-u user` Usuario con el que se conecta

`-s socket:fichero socket` con el que se conecta.

SINTAXIS DE CONEXIÓN

Ejemplos:

#Conexión con usuario y password: `mysql -u root -p`

Enter password:*****

#conexión con usuario y password a la base de datos jardinería del host 192.168.3.100

`mysql -u root -pPassworddelusuario -h 192.168.3.100 jardineria`

#conexión con usuario y password a la base de datos jardinería del host 192.168.3.100
con puerto 15300

`mysql -u root -pPassworddelusuario -h 192.168.3.100 jardineria -P 15300`



```
MariaDB [(none)]> select user(),connection_id(), version(),database();
```

user()	connection_id()	version()	database()
root@localhost	10	10.4.14-MariaDB	NULL

```
1 row in set (0.000 sec)
```

```
MariaDB [(none)]> select user(),  
-> connection_id(),  
-> version(),  
-> database();
```

user()	connection_id()	version()	database()
root@localhost	10	10.4.14-MariaDB	NULL

```
1 row in set (0.000 sec)
```



¿Como crear una base de datos?

- **CREATE DATABASE [IF NOT EXISTS]NOMBRE__BD**

```
MariaDB [(none)]> CREATE DATABASE PRIMERA;  
Query OK, 1 row affected (0.004 sec)
```

```
MariaDB [(none)]>
```

¿COMO VER LAS BASES DE DATOS ?

```
MariaDB [(none)]> SHOW DATABASES;
```

Database
information_schema
mysql
performance_schema
phpmyadmin
primera
test

```
6 rows in set (0.018 sec)
```



CREATE

SINTAXIS:

CREATE {DATABASE|SCHEMA} [IF NOT EXISTS] nombre_db

[DEFAULT] CHARACTER SET juego_caracteres|

[DEFAULT] COLLATE nombre_colocación



USAR UNA BASE DE DATOS.- USE.-

```
mariaDB [(none)]> USE PRIMERA;  
Database changed  
mariaDB [PRIMERA]> _
```



CREAR UNA TABLA

```
CREATE [TEMPORARY] TABLE [esquema] nombre_tabla
```

```
( nombre_campo1 tipo de datos Modificadores ,
```

```
(nombre_campo2 tipo de datos Modificadores,
```

```
.....
```

```
PRIMARY KEY ( campo1, campo2, ...)
```

```
FOREIGN KEY (campo 1) REFERENCES nombre_tabla (campo) opciones,
```

```
FOREIGN KEY (campo 1) REFERENCES nombre_tabla (campo) opciones,
```

TEMPORARY: EL TOKEN ES OPCIONAL, PARA INDICAR TABLAS TEMPORALES QUE SE DESTRUIRÁN EN EL MOMENTO EN QUE EL USUARIO SE DESCONECTE



TIPOS DE DATOS.-

Numéricos: TINYINT(1byte) Binarios: BIT(1bit) o BOOL

INT

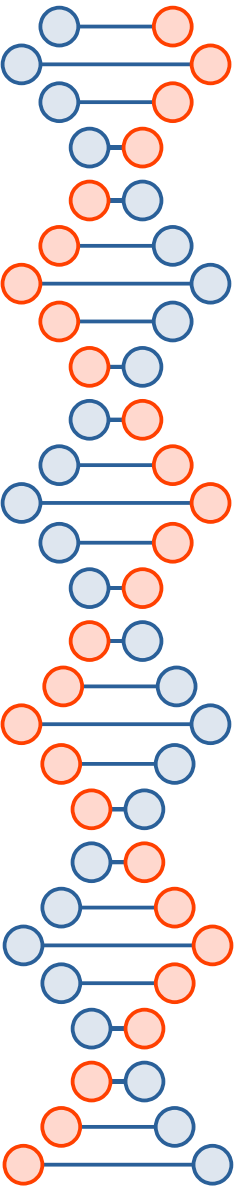
DECIMAL o DEC(parte entera, decimales)

Carácter. CHAR(n): Cadena de longitud fija. Se completa con espacios

VARCHAR(n): Cadenas de longitud variables. Se ajusta al contenido

TEXT y BLOB: grandes cantidades de texto.

Modificadores o restricciones

- 
- Null o not null
 - Default
 - Unique key
 - Auto_increment
 - Comment
 - Primary Key
 - Check
 - If not exists
 - max_rows
 - min_rows
 - Engine...



Modificadores o restricciones

```
create table if not exists pedido(  
    codigo int auto_increment primary key,  
    fecha datetime,  
    estado enum('pendiente','entregado','rechazado')  
    comment='tabla de pedido a proveedores'  
    auto_increment=10000  
    max_rows=1000000  
    engine=innodb )
```



Modificadores o restricciones

OPCIONES: especifica que hacer con las reglas de integridad referencial, en el caso de la tabla padre realizar:

ON DELETE

ON UPDATE

Las opciones ON DELETE y ON UPDATE establecen el comportamiento del gestor en caso de que las filas de la tabla padre (es decir, la tabla referenciada) se borren o actualicen. Los comportamientos pueden ser CASCADE, SET NULL Y NO ACTION.



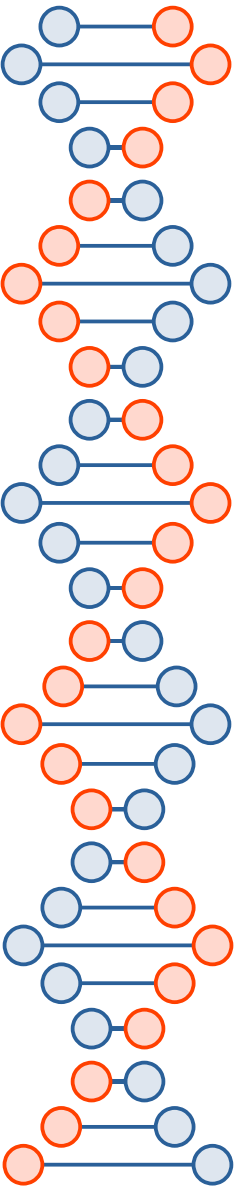
Modificadores o restricciones

Si se usa un *NO ACTION*, y se intenta un borrado o actualización sobre la tabla padre, la operación se impide, rechazando el borrado o la actualización

Si se especifica *CASCADE*, la operación se propaga a la tabla hija, es decir, si se actualiza la tabla padre, se actualizan los registros relacionados con la tabla hija, y si se borra un registro de la tabla padre, se borran aquellos registros de la tabla hija que estén referenciando al registro borrado.

Si se indica *SET NULL*, se establece a null la clave foránea afectada por un borrado o modificación de la tabla padre.

Modificadores o restricciones



```
create table clientes(  
dni varchar(8) PRIMARY KEY  
nombre varchar(50),  
direccion varchar(60)  
);
```

```
create table mascotas(  
codigo integer PRIMARY KEY,  
nombre varchar(50),  
raza varchar(50),  
cliente varchar(9) REFERENCES clientes (dni)  
ON DELETE CASCADE ON UPDATE SET NULL";
```

Si no se especifica ON DELETE u ON UPDATE por defecto se actúa como NO ACTION.



Modificación de tablas (ALTER TABLE)

`ALTER TABLE nombre_table`

`{[ADD (columna tipodato [restriccion columna][..])]`

`[MODIFY (columna tipodato[restriccion columna][..])]`

`[ADD CONSTRAINTS restricción]`

`[DROP CONSTRAINTS restricción]};`



Añadir o modificar columnas

`ALTER TABLE nombretabla {ADD|MODIFY} (columna tipo[restriccion..]);`

Eliminación de columnas:

`ALTER TABLE nombretabla DROP COLUMN nombre_columna;`

Añadir restricción de tabla

`ALTER TABLE nombretabla ADD CONSTRAINT nombrerestriccion restriccion;`

Eliminar una restricción de una tabla

`alter table nombre table DROP CONSTRAINT nombre_restricción;`

Activación y desactivación de una restricción

`alter table nombretalbe [enable validate|enable novalidate|disable] nombrerestricción`

Donde:

enable validate: activa la restricción si el conjunto cumple la restricción

enable novalidate: activa la restricción para las siguientes instrucciones

disable: desactiva la restricción



Renombrado de tablas

`RENAME TABLE nombre_tabla TO nombre_tabla_nuevo`

Consultar la estructura de una tabla

`DESCRIBE [ESQUEMA] nombre_tabla;`

`show create table nombre_tabla.`

da un listado de la tabla mostrando sus columnas, y aportando información sobre los tipos de datos, restricciones, etc.



CREACIÓN DE INDICES

Un índice es una estructura de datos que permite acceder a las filas de una tabla siguiendo el orden de los valores de alguna o algunas de sus columnas. Por tanto, las dos cosas más importantes que hay que especificar para definir un índice son la tabla sobre la que se va a crear y las columnas que se van a utilizar para establecer el orden.

```
CREATE [UNIQUE] INDEX nombre_índice ON nombre_tabla (  
    (nombre_columna orden,  
    nombre_columna orden,  
    ..... );
```

El orden puede ser ASC(ascendente) o DESC(descendente) . Si no se especifica por defecto es ASC.

Cada índice se actualiza de forma automática con cada modificación, inserción o borrado de registros de la tabla, con lo que siempre será consistente respecto a los datos de la misma.

Ejemplos:

```
CREATE INDEX X_Nombre ON SUMINISTRADOR (Snombre);
```

```
CREATE INDEX ALFA ON SUMINISTRADOR
```

```
( SNOMBRE ASC,
```

```
CIUDAD DESC);
```



CREACIÓN DE VISTAS (CREATE VIEW)

Una vista es un concepto lógico formado por una porción de una o varias tablas. Es equivalente a una "ventana" que permite ver solamente las filas y columnas necesarias para una aplicación.

Hay varias razones para utilizar vistas en lugar de las tablas sobre las que se definen:

- Seguridad. Con las vistas se puede ofrecer a los usuarios parte de la información y ocultar aquella que se considere confidencial.

- Comodidad. Al trabajar exclusivamente con los datos necesarios (que serán los que se seleccionen en la vista), la presentación resulta más sencilla y más fácil de entender.

La sentencia utilizada para crear una vista es `CREATE VIEW`, cuya sintaxis general es:

```
CREATE VIEW nombre_vista posibles_campos AS subconsulta;
```

Físicamente, una vista no ocupa espacio al ser creada, ya que no se construye realmente hasta el momento en que se hace referencia a ella mediante una operación de consulta. Una vista es una tabla virtual, donde la definición de la misma se almacena en el catálogo o diccionario del sistema. Por lo tanto, toda modificación en la tabla o tablas base sobre la que se ha definido la vista se verá reflejada en ella y viceversa.



Borrado de objetos

- DROP DATABASE nombre_DB;
- DROP TABLE Nombre_table; borra el contenido y la estructura
- TRUNCATE TABLE NOMBRE_TABLE: borrar el contenido de la tabla
- Drop index nombre_indice
- DROP VIEW nombre_vista

Ejemplo.-

CREAR LA BASE DE DATOS ALMACÉN

- La tabla SUMINISTRADOR representa proveedores o suministradores. Cada uno tiene un número de suministrador (SNUM) único, un nombre (SNOMBRE) no por fuerza único, un valor de situación (SITUACION) y una localidad (CIUDAD).

SUMINISTRADOR (Snum, Snombre, Situacion, Ciudad)

- La tabla PRODUCTO representa Artículos. Cada artículo tiene un número de artículo (PNUM) único, un nombre de artículo (PNOMBRE), un color (COLOR), un peso (PESO) y una localidad donde se almacena dicho producto (CIUDAD).

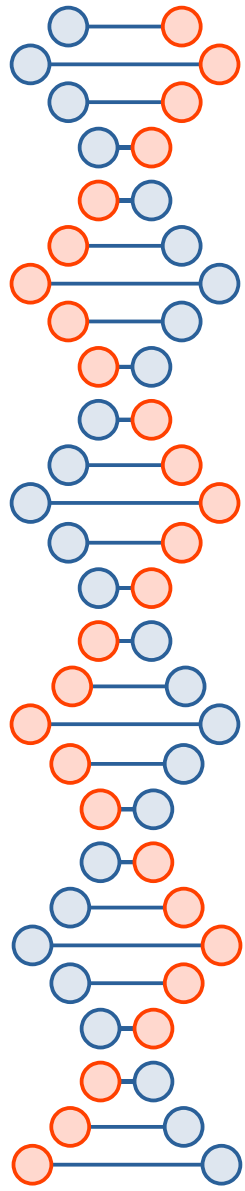
PRODUCTO (Pnum, Pnombre, Color, Peso, Ciudad)

- La tabla VENTAS representa ventas o envíos. Sirve para conectar entre sí las otras dos tablas. Cada envío tiene un número de suministrador (SNUM), un número de artículo (PNUM) y la cantidad enviada (CANTIDAD).

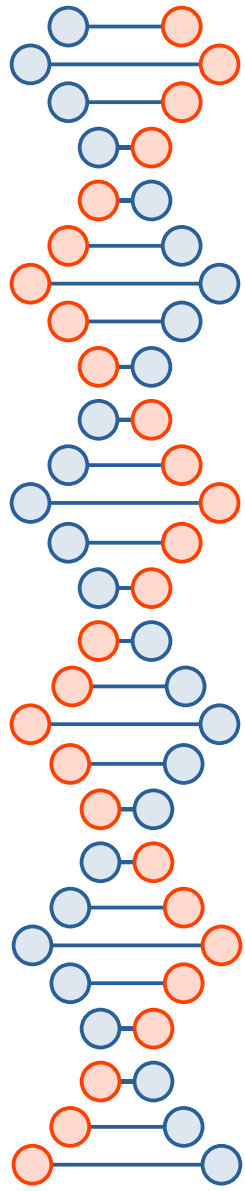
VENTAS (Snum, Pnum, Cantidad)

Los suministradores y los artículos pueden considerarse como entidades, y un envío o venta puede considerarse una interrelación entre un determinado suministrador y un artículo

Las claves primarias de la base de datos son el campo SNUM para la tabla SUMINISTRADOR, el campo PNUM para la tabla PRODUCTO, y el campo o atributo compuesto (SNUM, PNUM) para la tabla VENTAS .



```
create database if not exists almacén character set utf8;
use almacén;
create table if not exists suministrador(
    Snum varchar(5) primary key,
    Snombre varchar(20) not null,
    Situación int,
    ciudad varchar(20)
) character set utf8 collate utf8_general_ci;
create table if not exists productos(
    pnum varchar(5) primary key,
    pnombre varchar(10),
    color varchar(10),
    peso int,
    ciudad varchar(10)
) character set utf8 collate utf8_spanish2_ci;
create table if not exists ventas(
    Snum varchar(5),
    Pnum varchar(5),
    cantidad int,
    primary key(Snum, Pnum)
) character set utf8 collate utf8_spanish2_ci;
alter table ventas add foreign key (Snum) references suministrador(Snum);
alter table ventas add foreign key (Pnum) references productos(pnum);
```

insert into suministrador values

("S1","Salazar",20,"Londres"),

("S2","Jaime",10,"Paris"),

("S3","Bernal",30,"Paris"),

("S4","Corona",20,"Londres"),

("S5","Aldana",30,"Atenas");

insert into productos values

("P1","Tuerca","Rojo",12,"Londres"),

("P2","Perno","Verde",17,"Paris"),

("P3","Birlo","Azul",17,"Roma"),

("P4","Birlo","Rojo",14,"Londres"),

("P5","Leva","Azul",12,"Paris"),

("P6","Engrane","Rojo",19,"Londres");

insert into ventas values(

"S1","P1",300),

("S1","P2",200),

("S1","P3",400),

("S1","P4",200),

("S1","P5",100),

("S1","P6",100),

("S2","P1",300),

("S2","P2",400),

("S3","P2",200),

("S4","P2",200),

("S4","P4",300),

("S4","P5",400);



Consulta de datos: Cláusula Select

SELECT lista

from nombre_tabla

where condicion

group by expresiones

having condicion

order by expresion [asc|desc]

select: especifica las columnas que serán devueltas

from: indica la tabla o tablas a las que se accederá

Group by: agrupa las filas seleccionadas por la cláusula where

having: especifica una condición de selección para un grupo

order by: ordena las filas seleccionadas