

# APUNTES CREACIÓN DE BASE DE DATOS.

## MODELO RELACIONAL

---

### CREAR CONTENEDOR MARIADB

```
docker run -d -it -v /home/'USUARIO'/'NOMBRE_CARPETA_VM':/var/lib/mysql --name  
NOMBRE_CONTENEDOR -p 3336:3306 -e MARIADB_ROOT_PASSWORD=root mariadb
```

### CREAR CONTENEDOR MARIADB CON OTRO USUARIO A PARTE DE ROOT

```
docker run -d -it -v /home/'USUARIO'/'NOMBRE_CARPETA_VM':/var/lib/mysql --name  
NOMBRE_CONTENEDOR -p 3336:3306 -e MARIADB_USER='nombre' -e  
MARIADB_PASSWORD='contraseña' -e MARIADB_ROOT_PASSWORD=root mariadb
```

### ENTRAR AL CONTENEDOR

```
--> docker start "nombre contenedor" -> INICIA CONTENEDOR
```

```
--> docker exec -it [nombre_del_contenedor] /bin/bash -> ENTRA EN EL CONTENEDOR
```

```
--> mariadb -u root -p -> ENTRA EN SERVIDOR
```

### EJECUTAR COMANDOS DENTRO DE LA BASE DE DATOS

```
--> use "nombre_base_datos"; -> ENTRA EN BASE DE DATOS
```

```
--> create database "nombre_bdatos"; -> CREA BD
```

```
--> show databases; -> MUESTRA BD's EXISTENTES EN SERVIDOR
```

```
--> show tables; -> MUESTRA TABLAS EN LA BD
```

### CAMBIAR CLAVE A UN USUARIO DEL CONTENEDOR

```
--> passwd nombre_de_usuario (se ejecuta dentro del contenedor, como root)
```

### CAMBIAR CLAVE A ROOT

```
--> SET password=PASSWORD("clave_a_establecer");
```

### IMPORTAR .SQL A CONTENEDOR DE DOCKER

```
--> docker cp ruta/al/archivo.sql nombre_del_contenedor:/ruta/en/contenedor
```

```
--> docker exec -it nombre_del_contenedor bash
```

```
--> mysql -u usuario -p base_de_datos < /ruta/en/contenedor/archivo.sql
```

### **EJEMPLOS PARA CREAR LA BASE DE DATOS**

```
CREATE OR REPLACE DATABASE mi_base_datos;  
  
CREATE SCHEMA mi_esquema;  
  
CREATE DATABASE IF NOT EXISTS mi_nueva_base_datos;  
  
CREATE DATABASE mi_base_datos CHARACTER SET utf8;  
  
CREATE DATABASE mi_base_datos CHARACTER SET utf8 COLLATE  
utf8_general_ci;  
  
CREATE DATABASE mi_base_datos COMMENT 'Esta es mi base de datos para  
pruebas';
```

### **EJEMPLOS PARA BORRAR/MODIFICAR BASE DE DATOS**

```
DROP DATABASE IF EXISTS <bd>;  
  
DROP TABLE nombre_tabla;  
  
DELETE FROM nombre_tabla WHERE condición (ej: id=1);  
  
SHOW CREATE DATABASE <bd>; (muestra la sentencia para crear la bd)  
  
ALTER DATABASE <bd> CHARACTER SET utf8 COLLATE utf8_general_ci  
COMMENT 'Base de datos modificada';  
  
SELECT * FROM information_schema.schemata; (vistas que contienen todos  
los esquemas, tablas, columnas...etc de la bd)  
  
ALTER DATABASE <bd> RENAME TO <bd_1>;
```

### **EJEMPLOS PARA CREAR TABLAS**

```
CREATE TABLE persona (  
idPersona INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
apellido VARCHAR(255),  
nombre VARCHAR(255) NOT NULL,  
direccion VARCHAR(255),  
sexo CHAR(1) DEFAULT '1',  
ciudad VARCHAR(255)
```

);

## **EJEMPLO CON RESTRICCIONES DE BORRADO**

```
CREATE TABLE empleados (  
    id_empleado INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    id_departamento INT,  
    FOREIGN KEY (id_departamento) REFERENCES  
departamentos(id_departamento)  
    ON DELETE SET NULL(restrict, cascade)  
);
```

## **MODIFICAR VALORES DE UNA TABLA**

```
ALTER TABLE persona ADD apellido1 VARCHAR(50) NOT NULL AFTER  
nombre;  
  
ALTER TABLE persona ADD apellido2 VARCHAR(50) AFTER apellido1; (after  
nombre para que se añada después del atributo nombre)  
  
ALTER TABLE persona ADD COLUMN fecha_nacimiento DATE NOT NULL;  
  
ALTER TABLE persona DROP COLUMN IF EXISTS ciudad;  
  
ALTER TABLE persona DROP COLUMN ciudad;  
  
ALTER TABLE persona ALTER sexo DROP DEFAULT;  
  
ALTER TABLE persona ALTER sexo SET DEFAULT 'M';  
  
ALTER TABLE persona ADD COLUMN fecha_nacimiento DATE NOT NULL;  
  
ALTER TABLE persona CHANGE nombre nombre_de_usuario VARCHAR(50)  
NOT NULL;  
  
ALTER TABLE persona MODIFY nombre_de_usuario VARCHAR(100);  
  
RENAME TABLE persona TO personas;  
  
UPDATE productos SET precio = 17.99 WHERE id = 1;
```

## **RESTRICCIONES POR CAMPOS**

nombre\_empleado VARCHAR(100) NOT NULL

cantidad\_en\_stock INT DEFAULT 0

edad INT CHECK (edad >= 18)

email VARCHAR(255) UNIQUE

FOREIGN KEY (id\_cliente) REFERENCES clientes(id\_cliente)

CONSTRAINT <nombre> FOREIGN KEY (PersonID) REFERENCES  
persona(PersonID));

## **INDICES**

CREATE INDEX nombre\_indice ON nombre\_tabla(columna1 [, columna2 ...]);

DROP INDEX nombre\_indice ON nombre\_tabla;

## **VISTAS**

CREATE VIEW vista\_empleados\_altos\_salarios AS

SELECT nombre, departamento, salario

FROM empleados

WHERE salario > 50000;

## **CONSULTA DE VISTA**

SELECT nombre, salario

FROM vista\_empleados\_altos\_salarios

WHERE departamento = 'Tecnología';

## **PARTICIONES**

**LIST (parte la tabla según una lista, en este caso según el departamento al que pertenecen los empleados)**

```
CREATE TABLE empleados (  
    id_empleado INT,  
    nombre VARCHAR(100),  
    departamento VARCHAR(50)  
)  
  
PARTITION BY LIST COLUMNS(departamento) (  
    PARTITION p_tecnologia VALUES IN ('Tecnología', 'IT'),  
    PARTITION p_rrhh VALUES IN ('Recursos Humanos'),  
    PARTITION p_ventas VALUES IN ('Ventas', 'Comercial')  
);
```

**HASH (parte la tabla en función del valor de una columna, tantas particiones como se indique)**

```
CREATE TABLE clientes (  
    id_cliente INT,  
    nombre VARCHAR(100),  
)  
  
PARTITION BY HASH (id_cliente) PARTITIONS 4;
```

**KEY (como hash pero parte de forma uniforme)**

```
CREATE TABLE pedidos (  
    id_pedido INT,  
    id_cliente INT,  
    fecha_pedido DATE,  
) PARTITION BY KEY (id_cliente) PARTITIONS 8;
```

### **RANGE (se basa en un rango de valores)**

```
CREATE TABLE facturas (  
    id_factura INT,  
    fecha_factura DATE,  
)  
  
PARTITION BY RANGE (YEAR(fecha_factura)) (  
    PARTITION p_2019 VALUES LESS THAN (2020),  
    PARTITION p_2020 VALUES LESS THAN (2021),  
    PARTITION p_2021 VALUES LESS THAN (2022),  
    PARTITION p_2022 VALUES LESS THAN (2023)  
);
```

### **COLUMNAS GENERADAS O CALCULADAS**

```
CREATE TABLE usuario (  
    idUsuario INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(45),  
    primerApellido VARCHAR(45),  
    usuario VARCHAR(147) AS (CONCAT(nombre, ' ', primerApellido )) STORED  
);
```

### **USUARIOS**

```
CREATE USER 'centro'@'localhost' IDENTIFIED BY 'centro';  
  
SHOW GRANTS FOR 'centro'@'localhost';  
  
SELECT, CREATE, INSERT, DROP, UPDATE, GRANT OPTION, ALL  
PRIVILEGES, *.*  
  
GRANT SELECT ON centro_ensenanza.ASIGNATURA TO 'centro'@'localhost'; (tabla  
asignatura de la bd centro_ensenanza)  
  
REVOKE ALL PRIVILEGES ON centro_ensenanza.ALUMNO FROM  
'centro'@'localhost';
```

```
SHOW GRANTS FOR ana;
```

```
SET PASSWORD [FOR nombre_usuario[@nombre_equipo]] =  
PASSWORD('nueva_clave');
```

## **ROLES**

```
CREATE ROLE crm_dev;
```

Asignar privilegios sobre la BD crm a roles:

```
GRANT ALL ON crm.* TO crm_dev; -- Todos los permisos al rol crm_dev
```

```
GRANT SELECT ON crm.* TO crm_read;
```

Crear varios usuarios:

```
CREATE USER crm_dev1 IDENTIFIED BY 'crm_dev1';
```

Asignar los roles a los usuarios:

```
GRANT crm_dev TO crm_dev1;
```

Verificar los roles asignados:

```
SHOW GRANTS FOR crm_write;
```

## **TRANSACCIONES**

```
START TRANSACTION;
```

```
INSERT INTO cuentas (cliente_id, cantidad) VALUES (1, 100);
```

```
UPDATE total_cuentas SET total = total + 100 WHERE cliente_id = 1;
```

```
COMMIT;
```

```
mysqldump -u [usuario] -p[contraseña] [nombre_de_la_base] >  
archivo_dump.sql
```