**Capítulo 1**

MySQL: MySQL permite almacenar y acceder a los datos a través de múltiples motores de almacenamiento, incluyendo InnoDB, CSV y NDB. MySQL también es capaz de replicar datos y particionar tablas para mejorar el rendimiento y la durabilidad.

**Crear una base de datos**

CREATE DATABASE <nombre de la base de datos>;

Ejemplo:

CREATE DATABASE EJEMPLOMYSQL;

**Usar o seleccionar la base de datos**

Use < nombre de la base de datos >;

Ejemplo:

Use EJEMPLOMYSQL;

**Ver una lista de todas las bases de datos**

Show databases;

**CRUD Tablas**

**Crear una tabla / entidad**

CREATE TABLE <Nombre de la tabla en minúscula> (

<identificador>

<datos>

);

**Ver tablas**

Para ver todas las tablas en general

show tables;

Para ver una tabla con su estructura.

describe persona;

**Modificar tabla**

ALTER TABLE < Nombre de la tabla > rename < Nuevo nombre >;

Ejemplo:

alter table animal rename persona;

**Eliminar tabla**

DROP TABLE <Nombre de la tabla>;

drop table animal;

**CRUD Columnas**

**Crear columna**

<Nom\_ columna> <Tipo de dato> <’Tamaño’>;

Ejemplo: Se crea una base de datos con el nombre animal, se crea al id con el tipo de dato entero y auto incrementable, se colocan datos de tipo string y se asigna el id como llave primaria.

Ejemplo;

CREATE TABLE animal (

id int auto\_increment,

tipo varchar(25),

estado enum(‘vivo’,’muero’),

primary key (id)

);

**Modificar columna**

Modifica el tipo de dato

ALTER TABLE <Tabla> modify columna <Nombre de la columna> <Tipo de dato nuevo>

Ejemplo:

alter table persona modify column tipo int;

Cambiar el nombre del atributo

ALTER TABLE <Tabla> change <Nombre actual> <Nuevo nombre> <Tipo de dato>

Ejemplo:

alter table animal change edad fechanacimiento int(3);

Agregar a la tabla una nueva columna

Ejemplo:

ALTER TABLE <Tabla> ADD <Nombre\_columna> <Tipo de dato> <Tamaño>;

alter table animal ADD edad int(3);

**Ver columna**

SHOW COLUMNS FROM < Nombre de tabla>;

Ejemplo:

SHOW COLUMNS FROM animal;

**Eliminar columna**

ALTER TABLE <tabla> DROP COLUMN <Nombre\_columna>

Ejemplo:

ALTER TABLE animal DROP COLUMN fechanacimiento;

**Inserciones y modificaciones**

**Comentarios**

Una sola línea

--<Texto>

Ejemplo:

--No modificar esta tabla

**CRUD** **Inserciones**

**Crear Inserciones**

Añadir datos dentro de una tabla y las columnas dependiendo de su posición y tipo de dato.

INSERT INTO <Tabla> VALUES (<Para Id se pone Null>, ‘primer atributo’, ‘segundo atributo’);

Ejemplo:

INSERT INTO animal VALUES (NULL,'Mamífero','extinto');

order by es para ordenar.

ASC de manera ascendente.

DESC de manera descendente.

WHERE <campo> <> “Texto”

Like “%Texto%”

WHERE <campo> = “texto”

WHERE (campo like “%texto%”) or (campo like “%texto%”)

**Ver Inserciones**

SELECT <Campos> from <Table>;

Ejemplo:

select tipo,estado from animal;

select tipo,estado from animal **order by** estado ASC;

**Editar Inserciones**

UPDATE ‘Tabla’ SET ‘’ ‘Columna1’ = ‘NuevoAtrubuto’, ‘Columna 2’ = ‘NuevoAtrubuto’ WHERE ‘Table’. ‘Nombre del id’ = <Numero del id>;

Ejemplo:

UPDATE `animal` SET `tipo` = 'Mamífero' WHERE `animal`.`id` = 1;

Ejemplo:

UPDATE `animal` SET `tipo` = 'Mamífero', `estado`= 'casado' WHERE `animal`.`id` = 1;

UPDATE `animal` SET `tipo` = 'Ma1', `estado` = 'viv1' WHERE `animal`.`id` = 1;

UPDATE `animal` SET `id` = '22', `tipo` = 'Ma', `estado` = 'ext' WHERE `animal`.`id` = 2;

**Eliminar Inserciones**

DELETE FROM `Tabla` WHERE `Table` . `Nombre del id` = <Numero del id>

Ejemplo:

DELETE FROM `animal` WHERE `animal`.`id` = 22;

**Llaves foráneas**

foreign key (Nombre de la FK creada previamente) reference <Tabla invocada>(Nombre de la PK de tabla invocada)

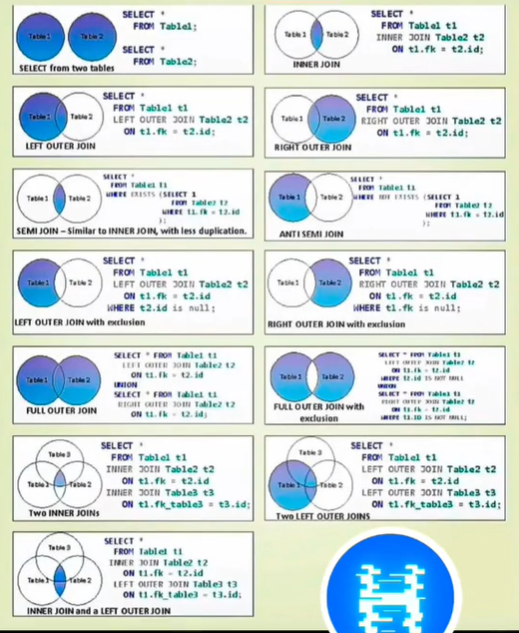
**Ejemplo**

Tabla alumno (

id\_Carrera int,

foreign key (id\_Carrera) references Carrera(idCarrera));

**Uniones**



**Tipos de datos**

-Numéricos.

-tynyint:

-Entero.

-Valor de -128 a 127.

-Rango de 0 a 225.

-Bit o bool:

10

-Solo almacena un valor de 0 o 1.

Smallint:

-Entero.

-Valor de -32768 a 32767.

-Rango de 0 a 65535.

Mediumint:

-Entero.

-Valor de -8.388.608 a 8.388.607.

-Rango de 0 a16777215.

BigInt:

-Entero.

-Valor de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.

-Rango de 0 a 18.446.744.073.709.551.615.11

Float:

-Real

-valor de -3.402823466E+38 a -1.175494351E-38, 0.

-Rango de 1.175494351E-38 a 3.402823466E+38.

Real, Double.

-Real

-Valor de -1.7976931348623157E+308 a -2.2250738585072014E-308, 0.

-Rango de 2.2250738585072014E-308 a 1.7976931348623157E+308

-Tipos de fecha.

-Date: coloca el día, mes y año.

-rango: 01-01-1001 a 31-12-9999

12

Datetime: coloca el día, mes, año, día, hora, minutos y segundos.

-Rango de 01-01-1001-0h-0m-0s a 31-12-9999-23h-59m-60s

TimeStamp: El formato depende.

-Rango 01-01-1970 a 2037

Time: formato de HH-MM-SS.

-Rango de –838 horas a 59 minutos a 59 segundos o contrario.

Year:

-Rango de 1910 a 2155.

13

Tipos de cadena.

Char(n):

-Rango de 0 a 255 caracteres.

VarChar(n):

-Rango de 0 a 255 caracteres

TinyText y TinyBlob:

-Rango de 255 caracteres.

Blob y Text:

-Rango de 65535 caracteres.

MediumBlob y MediumText:

-Rango de 16.777.215 caracteres.

14

LongBlob y LongText:

-Rango de caracteres 4.294.967.295.

Enum:

-Para valores inconsistente “m” o “f”.

-Rango de 65535 valores distintos

Set:

- La lista puede tener un máximo de 64 valores.