MySQL Alura

Clase 1: Conceptos basicos, crear y eliminar bases de datos

Columnas = Campos

Filas = Registros

USE world;/\*Usar base de datos\*/

SELECT \* FROM city;/\*Usar todo de city\*/

SELECT \* FROM country;

SELECT \* FROM countrylanguage;

/\*CREATE DATABASE jugos;\*//\*Creamos una base de datos, está localizada en C:\ProgramData\MySQL\MySQL Server 8.0\Data Podria copiar y pegar en Este Equipo para encontrarlo\*/

/\*DROP SCHEMA jugos;\*//\*Eliminamos jugos. Incluso vemos como se elimina del apartado SCHEMAS y de la carpeta Data. Tambien podria usar click derecho en esa base de datos y seleccionar drop para eliminarlo\*/

/\*jugos2 lo creamos dando click derecho en SCHEMAS y Create Schema y le aclaramos que se utiliza utf8 general ci \*/

/\*Borré jugos2 con click derecho\*/

Clase 2 : Creando una tabla

Para crear una tabla, tenemos que especificar el tipo de los campos (TINYINT, DOUBLE, FLOAT, etc.)

Los atributos de los campos numericos pueden ser:

- SIGNED o UNSIGNED : Con signo o sin signo.

- ZEROFILL : Los espacios vacios los llena con cero. La cantidad depende de cuantos espacios especificamos.

- AUTO\_INCREMENT: Incremento secuencial ej 1,2,3,4,5... tambien puede ser 2,4,6,8...etc.

- OUT OF RANGE: Error que se ejecuta cuando nos salimos de los limites establecidos.

-DATE: 1000-01-01 hasta 9999-12-31 (formato YYYY-MM-DD)

-DATETIME: Lo mismo que el anterior pero ademas almacena la hora

1000-01-01 00:00:00 hasta 9999-12-31 23:59:59 (formato YYYY-MM-DD hh:mm:ss)

-TIMESTAMP: Toma en cuenta la zona horaria del pais.

1970-01-01 00:00:01 UTC hasta 2038-01-19 UTC

-TIME: -838:59:59 hasta 839:59:59

-YEAR: 1901 hasta 2155 (Puede escribirse con 2 numeros o 4).

-CHAR Valor fijo. Llena los espacios vacios

-VARCHAR Valor variable. No llena los espacios vacios, se adapta al contenido

-BINARY Valor fijo con numeros binario

-VARBINARY Valor variable con numeros binarios

-BLOB Binarios largos -> TINYBLOB,MEDIUMBLOB,LONGBLOB

-TEXT Texto largos -> TINYTEXT,MEDIUMTEXT,LONGTEXT

-ENUM Define opciones en una lista

-SET y COLLATE: Caracteres que se aceptan utf8,utf16,etc.

-GEOMETRY - POLYGON -LINESTRING -POINT

EMPRESA DE JUGOS:

/\*Empresa de jugos

la información más importante de almacenar con respecto al cliente es:

- DNI - Nombre -Dirección - Edad -Sexo -Límite de crédito -Volumen mínimo de jugo que pueden comprar -Si ya realizaron la primera compra.

\*/

USE jugos;

CREATE TABLE TBCLIENTES(/\*Creamos tabla con nombre TBCLIENTES\*/

DNI varchar(20), /\*Recordar que varchar se adapta al contenido\*/

NOMBRE varchar(150), /\*Puede usar hasta 150 caracteres, pero recordemos que como es variable, si ponemos 5 letras, se adapta a esas 5 letras\*/

DIRECCION1 varchar(150),

DIRECCION2 varchar(150),

BARRIO varchar(50),

CIUDAD varchar(50),

ESTADO varchar(50),

CP varchar(10),

EDAD smallint(50),

SEXO varchar(1),

LIMITE\_\_CREDITO float,

VOLUMEN\_COMPRA float,

PRIMERA\_COMPRA BIT(1)

);

Ahora queremos hacer una tabla haciendo referencia a Productos dentro de jugos. Esta la vamos a crear con la ayuda del workbench con click derecho en tables

CREATE TABLE `jugos`.`tbproductos` (

`producto` VARCHAR(20) NULL,

`nombre` VARCHAR(150) NULL,

`envase` VARCHAR(50) NULL,

`volumen` VARCHAR(20) NULL,

`sabor` VARCHAR(50) NULL,

`precio` FLOAT NULL);

Creamos una nueva tabla para vendedores en jugos

USE JUGOS;

CREATE TABLE TABLA\_DE\_VENDEDORES(

MATRICULA VARCHAR(5),

NOMBRE VARCHAR(100),

PORCENTAJE\_COMISION FLOAT

);

Recordar que para borrar tablas se puede hacerles click derecho ‘drop table’. O bien escribimos

DROP TABLE NOMBRE\_TABLA;

CLASE 4

Descargamos el excel para el ejercicio, donde hay muchos datos para la tbproductos. Vamos a poner a mano datos para un producto.

INSERT INTO TBPRODUCTOS(/\*meter dentro de las llaves de tbproductos los valores\*/

producto, nombre, envase, volumen, sabor, precio

) VALUES (‘773912’,’clean’,’botella pet’, ‘1 litro’, ‘naranja’, 8.01);

/\*Recordar que los barchar van con comillas simples, los numeros como float no\*/

/\*Vemos resultado con\*/

SELECT \* FROM TBPRODUCTOS;

Agregar a mano vendedor a tabla\_de\_vendedores

USE JUGOS;

INSERT INTO tabla\_de\_vendedores(

matricula, nombre, porcentaje\_comision

) VALUES ('0023', 'Joan Geraldo de la Fonseca',0.1);

SELECT \* FROM tabla\_de\_vendedores;

Agregar un par mas de vendedores pero a la vez

USE JUGOS;

INSERT INTO tabla\_de\_vendedores(

matricula, nombre, porcentaje\_comision

) VALUES ('00235', 'Márcio Almeida Silva',0.08),

('00236', 'Cláudia Morais',0.08);

SELECT \* FROM tabla\_de\_vendedores;

Agregue un par de productos mas

USE jugos;

INSERT INTO tbproductos(

producto, nombre, envase, volumen, sabor,

precio) VALUES ('695594', 'Festival de Sabores', 'Botella PET',

'1.5 Litros', 'Asaí', 18.51);

INSERT INTO tbproductos(

producto, nombre, envase, volumen, sabor,

precio) VALUES ('1041119', 'Línea Citrus', 'Botella de Vidrio',

'700 ml', 'Lima', 4.90);

SELECT \* FROM tbproductos;

Pero en estos dos ultimos he cometido errores de tipeo y me di cuenta despues de subirlos al servidor.

Tenemos que aclarar que queremos actualizar la tabla y poner los nuevos datos en las llaves que quiero cambiar.

USE jugos;

UPDATE tbproductos SET envase= ‘Botella PET’

WHERE producto = ‘773912’; /\*Pero tenemos que clarar en donde\*/

UPDATE tbproductos SET precio = 28.51 WHERE producto = ‘695594’; /\*Pero tenemos que clarar en donde\*/

UPDATE tbproductos SET sabor = ‘Lima/Limón’ WHERE producto = ‘1041119’; /\*Pero tenemos que clarar en donde\*/

La primera vez. Para que nos permita hacer estos cambios tenemos que ir a edit, preferencias. En SQL editor vamos a desmarcar el Safe updates.

Y luego de volver a reiniciar y probar.

Ejercicio actualizar cosas de vendedores

USE JUGOS;

UPDATE tabla\_de\_vendedores SET PORCENTAJE\_COMISION=0.11 WHERE MATRICULA='00236';

UPDATE tabla\_de\_vendedores SET NOMBRE='Joan Geraldo de la Fonseca Junior' WHERE MATRICULA='0023';

UPDATE tabla\_de\_vendedores SET MATRICULA='00233' WHERE MATRICULA='0023';

SELECT \* FROM tabla\_de\_vendedores;

Eliminar registros de una tabla

Hay que tener cuidado con dejar solo “DELETE FROM tbproductos” porque se vaciará toda la tabla sin borrar los campos. Recordar la diferencia con DROP, que esta eliminaba la tabla de la base de datos.

DELETE FROM tbproductos WHERE producto='773912';

delete FROM tabla\_de\_vendedores WHERE NOMBRE='Joan Geraldo de la Fonseca Junior';

Es importante establecer una llave primaria para evitar la duplicacion de informacion en los registros.

En la llave primaria es aquel campo donde ninguno de los registros se puede repetir. Esto sirve mucho para que no se repitan numeros identificadores que tienen que ser unicos.

El ALTER TABLE se refiere a la tabla que queremos alterar.

ADD PRIMARY KEY(...) es la alteracion que queremos hacerle, que es agregar una llave primaria. En los parentesis va el campo que queremos evitar duplicacion.

ALTER TABLE tbproductos ADD PRIMARY KEY(producto);

Si insertamos un numero de producto duplicado, nos mostrará un error que lo da a entender.

Agregar columnas

ALTER TABLE tbclientes ADD COLUMN(FECHA\_NACIMIENTO date);

Recordar especificar el tipo de variable luego del nombre de la nueva columna. En este ejemplo es del tipo date.

---------------------------------------------------------

Ejercicio donde pongo casi todo en practica

use jugos;

DELETE FROM tabla\_de\_vendedores; /\*Vaciar registros de tabla\*/

ALTER table tabla\_de\_vendedores ADD primary key(MATRICULA);/\*Para que no se pueda repetir los datos\*/

ALTER TABLE tabla\_de\_vendedores ADD COLUMN(FECHA\_ADMISION date);/\*Agregar columna\*/

ALTER TABLE tabla\_de\_vendedores ADD COLUMN(DE\_VACACIONES BIT(1));/\*Agregar columna\*/

/\*Agregar vendedores y sus valores\*/

INSERT INTO tabla\_de\_vendedores(

MATRICULA, NOMBRE, PORCENTAJE\_COMISION, FECHA\_ADMISION, DE\_VACACIONES)values('00235','Márcio Almeida Silva',0.08,'2014-08-15',0),

('00236','Cláudia Morais',0.08,'2013-09-17',1),

(

/\*Matrícula -\*/ '00237',

/\*Nombre:\*/ 'Roberta Martins',

/\*Comision:\*/ 0.11,

/\*Fecha de admisión:\*/ '2017-03-18',

/\*¿Está de vacaciones?\*/ 1

),

(

/\*Matrícula - \*/'00238',

/\*Nombre:\*/ 'Péricles Alves',

/\*Comision:\*/ 0.11,

/\*Fecha de admisión:\*/ '2016-08-12',

/\*¿Está de vacaciones?\*/ 0

);

select \* FROM tabla\_de\_vendedores;/\*Mostrar en pantalla\*/

Clase 5

Podemos seleccionar los campos de distintas formas

SELECT \* FROM nomTable; (Selecciona todos los campos y registros de la tabla)

SELECT MATRICULA, NOMBRE, PORCENTAJE\_COMISION, FECHA\_ADMISION, DE\_VACACIONES FROM tabla\_de\_vendedores; (Selecciona especificamente estos campos)

Podriamos seleccionar algunos y solo se mostraran esos. Por ejemplo

SELECT MATRICULA, NOMBRE, FECHA\_ADMISION, FROM tabla\_de\_vendedores;

Incluso podemos elegir el orden en la cual mostrarse simplemente escribiendo el orden

SELECT FECHA\_ADMISION, MATRICULA, NOMBRE , FROM tabla\_de\_vendedores;

Tambien podemos darles un apodo a cada campo para ayudarnos en su escritura, por ejemplo

SELECT FECHA\_ADMISION AS FECHA, MATRICULA AS MAT, NOMBRE AS NOM , FROM tabla\_de\_vendedores;

Tambien podemos seleccionar si queremos ver los primeros registros

SELECT FECHA\_ADMISION AS FECHA, MATRICULA AS MAT, NOMBRE AS NOM , FROM tabla\_de\_vendedores LIMIT 3;

Con eso muestra los 3 primeros productos de la tabla.

Podemos seleccionar cualquier registro/s pero ademas filtrando. Por ejemplo

SELECT \* FROM tabla\_de\_vendedores WHERE NOMBRE = ‘Agustin Perea’;

Aquí lo que hicimos es que muestre todos campos de la tabla de vendedores donde en los registros tenga alguien ese nombre. Obviamente tambien podriamos decidir que mostrar, ya sea la matricula, fecha, etc como habiamos hecho antes.

Podemos seleccionar cualquier registro/s pero filtrando con condicionales.

SELECT \* FROM TBCLIENTE WHERE EDAD > 27;

SELECT \* FROM TBCLIENTE WHERE EDAD <> 27;/\*Asi es el diferente en MySQL\*/

SELECT \* FROM TBCLIENTE WHERE NOMBRE > 'Erica Carvajo';/\*Se muestran todos los nombres mayores a "E " en el alfabeto\*/

/\*Si queremos seleccionar un numero fijo de tipo double, no se va a poder, tendriamos que usar el tipo decimal para hacer eso. \*/

ALTER TABLE tbproducto MODIFY PRECIO\_LISTA decimal(4,2);

SELECT \* FROM TBPRODUCTO WHERE PRECIO\_LISTA= 28.51;

/\*Pero de todas formas con float podriamos hacer la alternativa de buscar dentro de un rango de numeros flotantes. Por ejemplo\*/

SELECT \* FROM TBproducto WHERE PRECIO\_LISTA >=28.48 AND PRECIO\_LISTA <=28.52;

/\*Que podemos hacer con\*/

SELECT \* FROM TBproducto WHERE precio\_lista BETWEEN 28.48 AND 28.52;

/\*Para filtrar por fechas podriamos hacer lo mismo que venimos haciendo. Pero si por ejemplo queremos seleccionar los que nacieron en un año en especifico, podemos hacer\*/

select \* from tbcliente WHERE year(FECHA\_NACIMIENTO)= 1995; /\*Obviamente podemos hacer lo mismo con dia y mes.\*/

/\*Podriamos utilizar otros condicionales obviamente\*/

SELECT \* FROM TBproducto WHERE envase = 'Botella de Vidrio' OR envase = 'Lata';

/\*Tambien condiciones compuestas\*/

SELECT \* FROM TBproducto WHERE (envase = 'Botella de Vidrio' OR envase = 'Lata') OR (precio\_lista BETWEEN 28.48 AND 28.52) ;