PRACTICA EXAMEN

Bases de datos

Nombre: Daniela Beatriz Martínez Montes
Profesor: Oscar Flores
Grupo 3.-B

1.- Creamos una base de datos llamada “Tienda”.

```
mysql> create database tienda;
Query OK, 1 row affected (0.00 sec)
```

2.-Utilizamos el comando “Show databases” para ver las bases de datos existentes y verificamos que nuestra base de datos llamada “Tienda” se creó correctamente.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| arduino            |
| automoviles        |
| biblioteca         |
| empresa            |
| escuela            |
| examenp1           |
| mysql              |
| performance_schema |
| programacion_eventos |
| sakila             |
| segundo_b          |
| sys                |
| tienda             |
| universidad        |
| viajes             |
| videoclub          |
| world              |
+--------------------+
```

3.- Para crear nuestros registros en la base de datos “Tienda”, tenemos que seleccionarla, con el comando “Use”, de esta manera indicamos que es en la base de datos que trabajaremos

```
mysql> use tienda;
Database changed
```

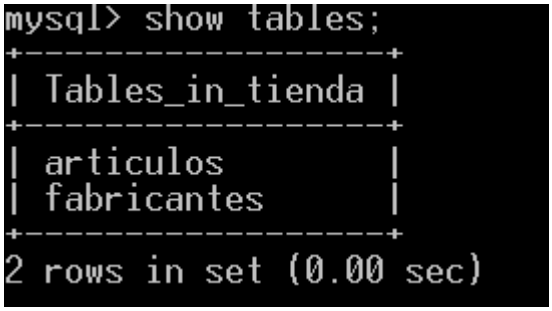
4.- Creamos una tabla llamada “Fabricantes” con el comando “Create table” seguido de paréntesis, donde se encontrarán los tipos de atributos que tendrá la tabla, son los siguientes:
“Clave_fabricante”, de tipo entero (int) positivo (unsigned), auto incrementable (auto_increment) y que sea llave primaria.
“nombre”, de tipo varchar de 30.
La llave primaria seria “clave_fabricante” (primary key(clave_fabricante)), debido a que todo dato auto_incremente debe ser llave primaria .

```
mysql> create table fabricantes(
    -> clave_fabricante int unsigned auto_increment,
    -> nombre varchar (30),
    -> primary key(clave_fabricante));
Query OK, 0 rows affected (0.55 sec)
```

Creamos otra tabla con el nombre de “Articulos”:
“Clave_articulos”, de tipo entero positivo, auto incrementable y que sea llave primaria.
“Nombre”, de tipo varchar de 30.
“Precio”, de tipo entero.
“Fabricante_clave”, Será llave foránea de “fabricantes”, es por eso que se utiliza el comando “constraint foreign key” que hace un vínculo entre los datos de otra tabla, pero con una restricción “foreign key”, luego toma de referencia la tabla “fabricantes” que es la tabla de nuestra llave foránea.

```
mysql> create table articulos(
    -> clave_articulo int unsigned auto_increment,
    -> nombre varchar (30),
    -> precio int,
    -> primary key(clave_articulo),
    -> fabricante_clave int unsigned,
    -> constraint foreign key (fabricante_clave)
    -> references fabricantes(clave_fabricante));
Query OK, 0 rows affected (0.77 sec)
```

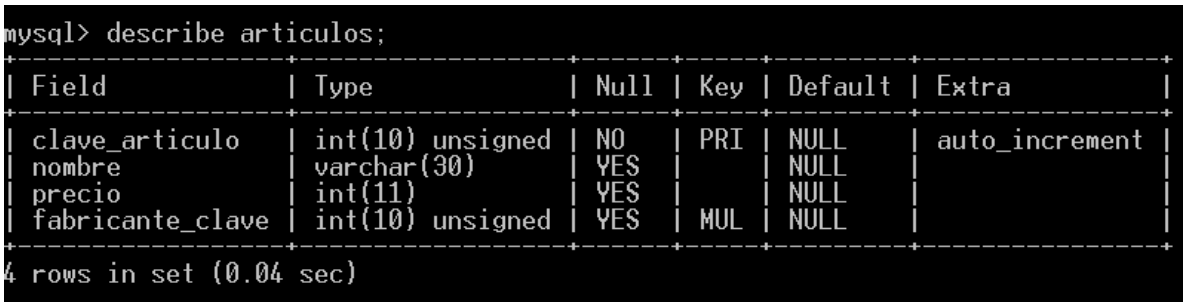
5.- “Show tables”, sirve para mostrar las tablas de la base de datos.



```
mysql> show tables;
+-------------------+
| Tables_in_tienda  |
+-------------------+
| articulos         |
| fabricantes       |
+-------------------+
2 rows in set (0.00 sec)
```

6.-

“Describe”, es un comando para mostrar todos los atributos de una tabla.



```
mysql> describe articulos;
+------------------+------------------+------+-----+---------+----------------+
| Field            | Type             | Null | Key | Default | Extra          |
+------------------+------------------+------+-----+---------+----------------+
| clave_articulo   | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| nombre           | varchar(30)      | YES  |     | NULL    |                |
| precio           | int(11)          | YES  |     | NULL    |                |
| fabricante_clave | int(10) unsigned | YES  | MUL | NULL    |                |
+------------------+------------------+------+-----+---------+----------------+
4 rows in set (0.04 sec)
```

7.-

“insert into” es un comando para insertar los datos a las tablas ya realizadas, como el ejemplo de arriba donde se inserta en el campo “nombre”, los valores que sean necesarios, valores que serán ingresados después de ejecutar el comando “values” que es el que almacena los datos en memoria.

Cuando se insertan los datos en la tabla “articulos”, en el campo de “fabricante_clave” se toma como referencia el “id_fabricantes”, esto quiere decir que el fabricante “Kingston” lleva como id el valor de 1, por ser el primer fabricante ingresado, es por eso que se ingresan números en el campo “fabricante_clave”, de la tabla artículos.

```
mysql> INSERT INTO articulos(nombre,precio,fabricante_clave)
    -> VALUES('Teclado',100,3),
    -> ('Disco Duro 300 GB',500,5),
    -> ('Mouse',80,3),
    -> ('Memoria USB',140,4),
    -> ('Memoria RAM',290,1),
    -> ('Disco duro extraible 250 GB',650,5),
    -> ('Memoria USB',279,1),
    -> ('DVD ROM',450,2),
    -> ('CD ROM',200,2),
    -> ('Tarjeta de red',180,3);
Query OK, 10 rows affected (0.07 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

En el caso de la tabla fabricantes, solo se ingresan en el nombre de quienes fabricaron el producto

```
mysql> INSERT INTO fabricantes(nombre)
    -> VALUES('Kingston'),('Adata'),('Logitech'),('Lexar'),('Seagate');
Query OK, 5 rows affected (0.08 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

8.- a) “select*from”, hace referencia a mostrar todos los datos insertados en una tabla, en este caso, todos los artículos de la tienda.

```
mysql> select * from articulos;
+----------------+---------------------------+--------+------------------+
| clave_articulo | nombre                    | precio | fabricante_clave |
+----------------+---------------------------+--------+------------------+
|              1 | Teclado                   |    100 |                3 |
|              2 | Disco Duro 300 GB         |    500 |                5 |
|              3 | Mouse                     |     80 |                3 |
|              4 | Memoria USB               |    140 |                4 |
|              5 | Memoria RAM               |    290 |                1 |
|              6 | Disco duro extraible 250 GB |    650 |                5 |
|              7 | Memoria USB               |    279 |                1 |
|              8 | DVD ROM                   |    450 |                2 |
|              9 | CD ROM                    |    200 |                2 |
|             10 | Tarjeta de red            |    180 |                3 |
+----------------+---------------------------+--------+------------------+
10 rows in set (0.00 sec)
```

b) Si queremos ser más específicos al momento de mostrar datos de una tabla, se escribe el campo que se desea mostrar, en este caso, “select nombre from articulos”, que hace referencia a mostrar únicamente los nombres de los artículos de la tabla.

```
mysql> select nombre from articulos;
+-----------------------------+
| nombre                      |
+-----------------------------+
| Teclado                     |
| Disco Duro 300 GB           |
| Mouse                       |
| Memoria USB                 |
| Memoria RAM                 |
| Disco duro extraible 250 GB |
| Memoria USB                 |
| DVD ROM                     |
| CD ROM                      |
| Tarjeta de red              |
+-----------------------------+
10 rows in set (0.00 sec)
```

c)

Si se desea mostrar dos datos de la tabla, solo es agregar los campos a mostrar separados con una coma.

```
mysql> select nombre,precio from articulos;
+-----------------------------+--------+
| nombre                      | precio |
+-----------------------------+--------+
| Teclado                     |    100 |
| Disco Duro 300 GB           |    500 |
| Mouse                       |     80 |
| Memoria USB                 |    140 |
| Memoria RAM                 |    290 |
| Disco duro extraible 250 GB |    650 |
| Memoria USB                 |    279 |
| DVD ROM                     |    450 |
| CD ROM                      |    200 |
| Tarjeta de red              |    180 |
+-----------------------------+--------+
10 rows in set (0.00 sec)
```

d) Cuando se desea mostrar datos sin repeticiones, se ejecuta el comando “distinct”. En este caso se ingresaron dos “Memoria USB” pero solo se muestra una.

```
mysql> select distinct nombre from articulos;
+-----------------------------+
| nombre                      |
+-----------------------------+
| Teclado                     |
| Disco Duro 300 GB           |
| Mouse                       |
| Memoria USB                 |
| Memoria RAM                 |
| Disco duro extraible 250 GB |
| DVD ROM                     |
| CD ROM                      |
| Tarjeta de red              |
+-----------------------------+
9 rows in set (0.09 sec)
```

e)

Una forma de mostrar datos aún más específicos, es por medio de condiciones, en este caso “where” que hace referencia a mostrar sólo los datos o el dato que queramos, en este caso es obtener todos los datos del artículo cuya clave de producto es ‘5’.

```
mysql> select * from articulos
    -> where clave_articulo=5;
+----------------+------------+--------+------------------+
| clave_articulo | nombre     | precio | fabricante_clave |
+----------------+------------+--------+------------------+
|              5 | Memoria RAM |    290 |                1 |
+----------------+------------+--------+------------------+
1 row in set (0.11 sec)
```

f)
Como el ejemplo anterior, utilizando “where” es obtener todos los datos del artículo cuyo nombre del producto es ‘’Teclado”.

```
mysql> select * from articulos
    -> where  nombre='Teclado';
+----------------+---------+--------+------------------+
| clave_articulo | nombre  | precio | fabricante_clave |
+----------------+---------+--------+------------------+
|              1 | Teclado |    100 |                3 |
+----------------+---------+--------+------------------+
```

g)
Para mostrar dos o más datos por medio de condiciones, lo único que se debe agregar es la instrucción “or”. En este caso es obtener todos los datos de la Memoria RAM y memorias USB.

```
mysql> select * from articulos
    -> where  nombre='Memoria Ram' or  nombre='Memoria USB';
+----------------+-------------+--------+------------------+
| clave_articulo | nombre      | precio | fabricante_clave |
+----------------+-------------+--------+------------------+
|              4 | Memoria USB |    140 |                4 |
|              5 | Memoria RAM |    290 |                1 |
|              7 | Memoria USB |    279 |                1 |
+----------------+-------------+--------+------------------+
3 rows in set (0.00 sec)
```

h)

Otra manera de mostrar datos, por decir, todos los datos que inicien con determinada letra, se obtienen con el comando “like”, seguido de la letra y un porcentaje, tal como se muestra en el ejemplo en el que se desea obtener todos los datos de los artículos que empiezan con ‘M’.

```
mysql> select * from articulos
    -> where nombre like 'M%';
+----------------+-------------+--------+------------------+
| clave_articulo | nombre      | precio | fabricante_clave |
+----------------+-------------+--------+------------------+
|              3 | Mouse       |     80 |                3 |
|              4 | Memoria USB |    140 |                4 |
|              5 | Memoria RAM |    290 |                1 |
|              7 | Memoria USB |    279 |                1 |
+----------------+-------------+--------+------------------+
4 rows in set (0.00 sec)
```

i)

En este ejemplo se requiere obtener el nombre de los productos donde el precio sea $100.

```
mysql> select nombre from articulos
    -> where precio=100;
+----------+
| nombre   |
+----------+
| Teclado |
+----------+
1 row in set (0.00 sec)
```

j)

En cuanto a mostrar datos “mayores a”, solo es cuestión de agregar <> (mayo, menor), dependiendo de lo que se requiera, en este caso es obtener el nombre de los productos donde el precio sea mayor a $ 200.

```
mysql> select nombre from articulos
    -> where precio>200;
+----------------------------+
| nombre                     |
+----------------------------+
| Disco Duro 300 GB          |
| Memoria RAM                |
| Disco duro extraible 250 GB |
| Memoria USB                |
| DVD ROM                    |
+----------------------------+
5 rows in set (0.00 sec)
```

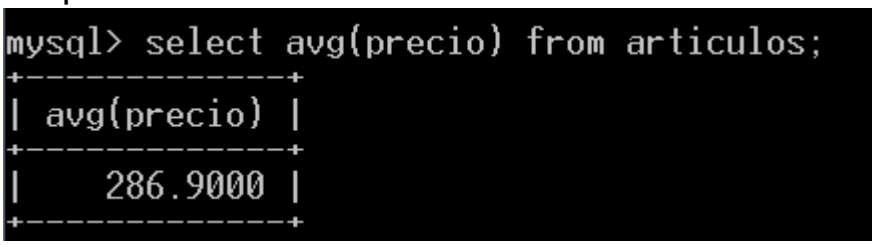
k)

Para mostrar datos entre un parámetro y otro, se utiliza en comando “between” y “and” como se muestra en el ejemplo anterior que se requiere obtener todos los datos de los artículos cuyo precio este entre $100 y $350.

```
mysql> select * from articulos
    -> where precio between 100 and 350;
+----------------+----------------+--------+------------------+
| clave_articulo | nombre         | precio | fabricante_clave |
+----------------+----------------+--------+------------------+
|              1 | Teclado        |    100 |                3 |
|              4 | Memoria USB    |    140 |                4 |
|              5 | Memoria RAM    |    290 |                1 |
|              7 | Memoria USB    |    279 |                1 |
|              9 | CD ROM         |    200 |                2 |
|             10 | Tarjeta de red |    180 |                3 |
+----------------+----------------+--------+------------------+
6 rows in set (0.05 sec)
```

l)

Para obtener el valor medio de un dato, se utiliza “avg(“nombre del campo”)”. En este caso es obtener el precio medio de todos los productos.



m) Si se desea realizar el comando anterior pero adjunto con una condición, el resultado sería como el ejemplo anterior, es que obtener el precio medio de los artículos cuyo código de fabricante sea 2.

n)

Para ordenar datos ya sea alfabetica o numericamente se utiliza en comando “order by” donde se hace referencia a los datos que se desean ordenar, en este caso es obtener el nombre y precio de los artículos ordenados por Nombre.

```
mysql> select nombre,precio from articulos
    -> order by nombre;
+-----------------------------+--------+
| nombre                      | precio |
+-----------------------------+--------+
| CD ROM                      |    200 |
| Disco Duro 300 GB           |    500 |
| Disco duro extraible 250 GB |    650 |
| DVD ROM                     |    450 |
| Memoria RAM                 |    290 |
| Memoria USB                 |    140 |
| Memoria USB                 |    279 |
| Mouse                       |     80 |
| Tarjeta de red              |    180 |
| Teclado                     |    100 |
+-----------------------------+--------+
10 rows in set (0.11 sec)
```

o) O bien, si la idea es obtener datos de forma descendente, solo hay que agregar el comando “desc” después de “order by -campo-” que en este caso se requiere obtener todos los datos de los productos ordenados descendentemente por Precio.

```
mysql> select *  from articulos
    -> order by precio desc;
+----------------+-----------------------------+--------+------------------+
| clave_articulo | nombre                      | precio | fabricante_clave |
+----------------+-----------------------------+--------+------------------+
|              6 | Disco duro extraible 250 GB |    650 |                5 |
|              2 | Disco Duro 300 GB           |    500 |                5 |
|              8 | DVD ROM                     |    450 |                2 |
|              5 | Memoria RAM                 |    290 |                1 |
|              7 | Memoria USB                 |    279 |                1 |
|              9 | CD ROM                      |    200 |                2 |
|             10 | Tarjeta de red              |    180 |                3 |
|              4 | Memoria USB                 |    140 |                4 |
|              1 | Teclado                     |    100 |                3 |
|              3 | Mouse                       |     80 |                3 |
+----------------+-----------------------------+--------+------------------+
```

p)

De igual forma, para ordenar datos de forma ascendente con el comando “asc”, adjuntando otro comando para ordenar otro valor ascendente como en el ejemplo donde se pretende obtener el nombre y precio de los artículos cuyo precio sea mayor a $ 250 y ordenarlos descendentemente por precio y luego ascendentemente por nombre.

```
mysql> select nombre,precio  from articulos
    -> where precio>250
    -> order by precio desc, nombre asc;
+-----------------------------+--------+
| nombre                      | precio |
+-----------------------------+--------+
| Disco duro extraible 250 GB |    650 |
| Disco Duro 300 GB           |    500 |
| DVD ROM                     |    450 |
| Memoria RAM                 |    290 |
| Memoria USB                 |    279 |
+-----------------------------+--------+
5 rows in set (0.00 sec)
```

q)

Obtener un listado completo de los productos, incluyendo por cada artículo los datos del artículo y del fabricante, en este caso utilizamos de igual manera el comando select, notamos que abreviamos a.(atributo),f.(atributo) con la sentencia from, indicamos que estos pertenecen a artículos y fabricantes respectivamente, con where pedimos que nos devuelva resultados donde la llave foránea de artículos, sea igual a la llave primaria de fabricantes

```
mysql> select a.clave_articulo,a.nombre,a.precio,f.nombre as fabricante
    -> from articulos a,fabricantes f
    -> where a.fabricante_clave=f.clave_fabricante;
+----------------+-----------------------------+--------+------------+
| clave_articulo | nombre                      | precio | fabricante |
+----------------+-----------------------------+--------+------------+
|              1 | Teclado                     |    100 | Logitech   |
|              2 | Disco Duro 300 GB           |    500 | Seagate    |
|              3 | Mouse                       |     80 | Logitech   |
|              4 | Memoria USB                 |    140 | Lexar      |
|              5 | Memoria RAM                 |    290 | Kingston   |
|              6 | Disco duro extraible 250 GB |    650 | Seagate    |
|              7 | Memoria USB                 |    279 | Kingston   |
|              8 | DVD ROM                     |    450 | Adata      |
|              9 | CD ROM                      |    200 | Adata      |
|             10 | Tarjeta de red              |    180 | Logitech   |
+----------------+-----------------------------+--------+------------+
10 rows in set (0.04 sec)
```

r)
Obtener la clave de producto, nombre del producto y nombre del fabricante de todos los productos en venta, aquí sucede lo mismo que en el anterior, solo que mostramos resultados diferentes

```
mysql> select a.clave_articulo,a.nombre,f.nombre as fabricante
    -> from articulos a,fabricantes f
    -> where a.fabricante_clave=f.clave_fabricante;
+----------------+-----------------------------+------------+
| clave_articulo | nombre                      | fabricante |
+----------------+-----------------------------+------------+
|              1 | Teclado                     | Logitech   |
|              2 | Disco Duro 300 GB           | Seagate    |
|              3 | Mouse                       | Logitech   |
|              4 | Memoria USB                 | Lexar      |
|              5 | Memoria RAM                 | Kingston   |
|              6 | Disco duro extraible 250 GB | Seagate    |
|              7 | Memoria USB                 | Kingston   |
|              8 | DVD ROM                     | Adata      |
|              9 | CD ROM                      | Adata      |
|             10 | Tarjeta de red              | Logitech   |
+----------------+-----------------------------+------------+
10 rows in set (0.18 sec)
```

s) Obtener el nombre y precio de los artículos donde el fabricante sea Logitech ordenarlos alfabéticamente por nombre del producto, sucede lo mismo que en los casos anteriores, solo que aquí solo mostraremos nombre y precio, y solo los productos que son del fabricante logitech

```
mysql> select a.nombre,a.precio
    -> from articulos a,fabricantes f
    -> where a.fabricante_clave=f.clave_fabricante and f.nombre='Logitech'
    -> order by a.nombre;
+----------------+--------+
| nombre         | precio |
+----------------+--------+
| Mouse          |     80 |
| Tarjeta de red |    180 |
| Teclado        |    100 |
+----------------+--------+
3 rows in set (0.04 sec)
```

t) Obtener el nombre, precio y nombre de fabricante de los productos que son marca Lexar o Kingston ordenados descendentemente por precio. Con la sentencia IN, indicamos que solo nos muestre atributos de Lexar y Kingston

```
mysql> select a.nombre,a.precio,f.nombre as fabricante
    -> from articulos a,fabricantes f
    -> where f.clave_fabricante=a.fabricante_clave and  f.nombre in('Lexar','Kin
gston')
    -> order by precio desc;
+-------------+--------+------------+
| nombre      | precio | fabricante |
+-------------+--------+------------+
| Memoria RAM |    290 | Kingston   |
| Memoria USB |    279 | Kingston   |
| Memoria USB |    140 | Lexar      |
+-------------+--------+------------+
3 rows in set (0.00 sec)
```

u)

Para insertar datos en una tabla, se utiliza el comando “Insert into” ya antes utilizado, en este caso, se requiere añadir un nuevo producto: Clave del producto 11, Altavoces de $120 del fabricante 2.

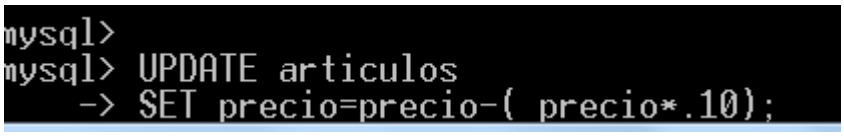
```
mysql> INSERT INTO articulos(nombre,precio,fabricante_clave)
    -> VALUES('Altavoces',120,2);
Query OK, 1 row affected (0.38 sec)
```

v)

Para cambiar nombres o valores de los datos de las tablas, se utiliza el comando “update nombre_tabla” y el comando “set” y el parámetro que se va a modificar igualándolo al nuevo valor, luego se crea la condición del valor a modificar como en el ejemplo, donde requiere cambiar el nombre del producto 6 a ‘Impresora Laser’.

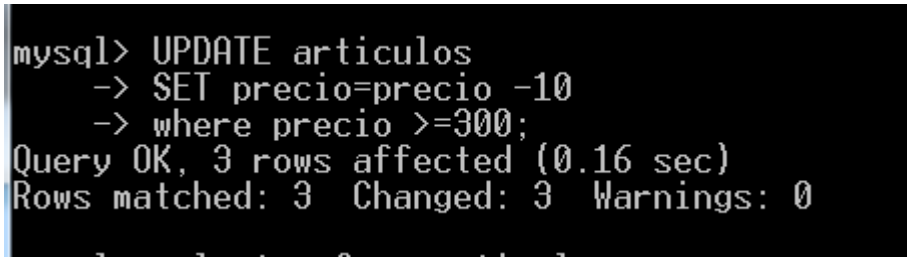
```
mysql> UPDATE articulos
    -> SET nombre='Impresora Laser'
    -> where clave_articulo=6;
Query OK, 1 row affected (0.24 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

w)
Aplicar un descuento del 10% a todos los productos, utilizamos UPDATE pare realizar cambios, y set para definir que valores cambiaremos, que este caso hacemos una operación ahí mismo, decimos que al precio se le reste precio *.01, es decir un 10 de descuento



x)

Para realizar cambios en los datos, se utiliza el comando “update” seguido del nombre de la tabla, luego se utiliza “set” para definir a cuales valores se cambiara el valor. En este ejemplo se desea aplicar un descuento de $ 10 a todos los productos cuyo precio sea mayor o igual a $ 300.



y)

Para borrar datos de una tabla se utiliza el comando “Delete” seguido del nombre de la tabla, seguido de la condición del dato que se desea eliminar en esta situación es borra el producto numero 6