

## 8 - Operadores Relacionales = <> < <= > >=



### Problema:

Un comercio que vende artículos de computación registra los datos de sus artículos en una tabla con ese nombre.

1- Elimine "articulos", si existe:

```
drop table if exists articulos;
```

2- Cree la tabla, con la siguiente estructura:

```
create table articulos(  
  codigo integer,  
  nombre varchar(20),  
  descripcion varchar(30),  
  precio float,  
  cantidad integer  
);
```

3- Vea la estructura de la tabla (describe).

4- Ingrese algunos registros:

```
insert into articulos (codigo, nombre, descripcion, precio,cantidad)  
values (1,'impresora','Epson Stylus C45',400.80,20);  
insert into articulos (codigo, nombre, descripcion, precio,cantidad)  
values (2,'impresora','Epson Stylus C85',500,30);  
insert into articulos (codigo, nombre, descripcion, precio,cantidad)  
values (3,'monitor','Samsung 14',800,10);  
insert into articulos (codigo, nombre, descripcion, precio,cantidad)  
values (4,'teclado','ingles Biswal',100,50);  
insert into articulos (codigo, nombre, descripcion, precio,cantidad)  
values (5,'teclado','español Biswal',90,50);
```

5- Seleccione todos los registros de la tabla. (select).

6- Muestre los datos de las impresoras.

7- Seleccione los artículos cuyo precio sea mayor o igual a 500:

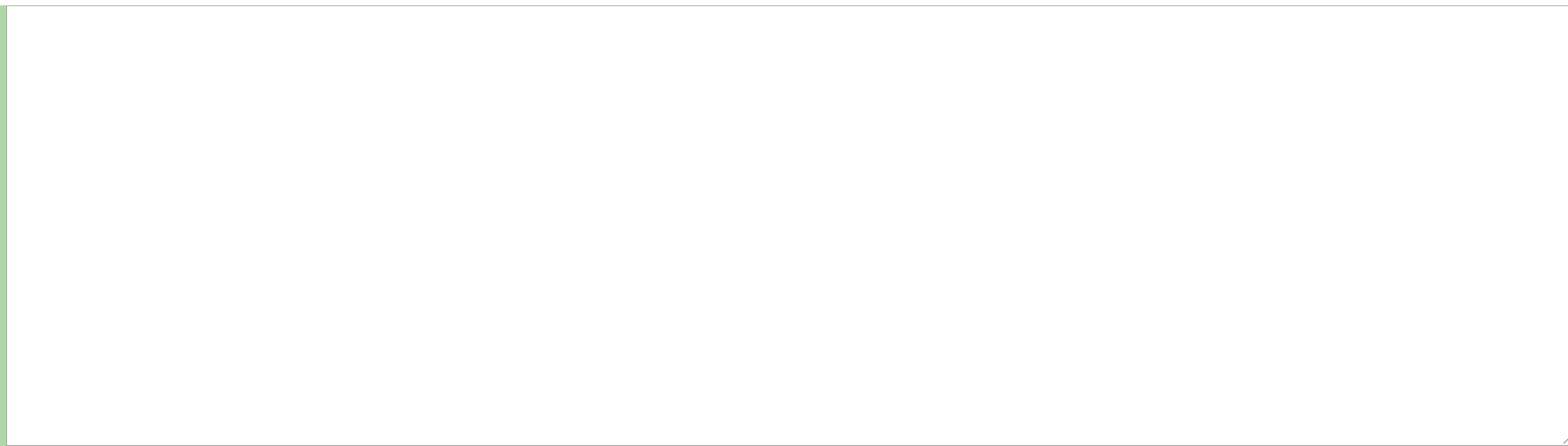
```
select * from articulos  
where precio>=500;
```

8- Seleccione los artículos cuya cantidad sea menor a 30:

```
select nombre,descripcion, precio,cantidad  
from articulos  
where cantidad<20;
```

9- Selecciones el nombre y descripción de los artículos que no cuesten \$100:

```
select nombre, descripcion  
from articulos  
where precio<>100;
```



Ejecutar los comandos SQL

Borrar comando.

Finalizar

### Otros problemas:

A) Un video club que alquila películas en video almacena la información de sus películas en alquiler en una tabla denominada "peliculas".

1- Elimine la tabla, si existe.

2- Cree la tabla eligiendo el tipo de dato adecuado para cada campo:

```
create table peliculas(  
  titulo varchar(20),  
  actor varchar(20),  
  duracion integer,  
  cantidad integer  
);
```

3- Vea la estructura de la tabla:  
describe peliculas;

4- Ingrese los siguientes registros:

```
insert into peliculas (titulo, actor, duracion, cantidad)  
values ('Mision imposible','Tom Cruise',120,3);  
insert into peliculas (titulo, actor, duracion, cantidad)  
values ('Mision imposible 2','Tom Cruise',180,2);  
insert into peliculas (titulo, actor, duracion, cantidad)  
values ('Mujer bonita','Julia R.',90,3);  
insert into peliculas (titulo, actor, duracion, cantidad)  
values ('Elsa y Fred','China Zorrilla',90,2);
```

5- Seleccione las películas cuya duración no supere los 90 minutos:

```
select *from peliculas  
where duracion<=90;
```

6- Seleccione todas las películas en las que el actor no sea 'Tom Cruise':

```
select * from peliculas  
where actor<>'Tom Cruise';
```

7- Seleccione todas las películas de las que haya más de 2 copias:

```
select * from peliculas  
where cantidad >2;
```

B) Trabaje con la tabla "agenda" en la que registra los datos de sus amigos.

1- Elimine "agenda", si existe.

2- Cree la tabla, con los siguientes campos: apellido (cadena de 30), nombre (cadena de 20), domicilio (cadena de 30) y telefono (cadena de 11).

3- Visualice la estructura de la tabla.

4- Ingrese los siguientes registros:

Mores,Alberto,Colon 123,4234567,  
Torres,Juan,Avellaneda 135,4458787,  
Lopez,Mariana,Urquiza 333,4545454,  
Lopez,Fernando,Urquiza 333,4545454.  
Suarez,Mariana,Sarmiento 643,4445544.  
Lopez,Ana,Sucre 309,4252587.

5- Seleccione todos los registros de la tabla.

6- Seleccione los amigos cuyo apellido sea 'Lopez'.

7- Seleccione los registros cuyo nombre NO sea 'Mariana'.

8- Seleccione los registros cuyo domicilio sea 'Colon 123'.

9- Muestre los datos de quienes tengan el teléfono '4545454'.

[Retornar](#)