

Ejercicios prácticos

MySQL: Consultas con cláusulas y operadores

Actividades

1.

Completá las siguientes consultas:

- a) Se quieren ordenar los resultados por el campo “apellido” de manera ascendente, de la A a la Z.
SELECT apellido, nombre, edad FROM estudiante ORDER BY _____;
- b) Se quieren ordenar los resultados por el campo “nombre” de manera descendente, de la Z a la A.
SELECT apellido, nombre, edad FROM estudiante ORDER BY _____;
- c) Se quieren ordenar los resultados por el campo “edad” de manera ascendente, de menor a mayor
SELECT apellido, nombre, edad FROM estudiante ORDER BY _____;
- d) Se quieren ordenar los resultados por el campo “edad” de manera descendente, de mayor a menor.
SELECT apellido, nombre, edad FROM estudiante ORDER BY _____;

2.

Se tiene una tabla que almacena las facturas de una juguetería con la siguiente estructura:

factura
id INT (PK) cliente_nombre VARCHAR(50) cliente_apellido VARCHAR(50) precio_total DECIMAL(8,2) cantidad_productos INT fecha DATE

Realizá las siguientes consultas:

- a) Obtené toda la información de las facturas, ordenadas por el apellido del cliente de la A a la Z.
- b) Obtené toda la información de las facturas, ordenadas por el precio de mayor a menor.
- c) Obtené toda la información de las facturas, ordenadas por la cantidad de productos que se compraron de menor a mayor.
- d) Obtené toda la información de las facturas, ordenadas por fecha, de la más reciente a la más antigua.
- e) Obtené toda la información de las facturas, ordenadas por la cantidad de productos que se compraron de mayor a menor.

3.

Investigá como se puede ordenar por dos columnas a la vez. Teniendo en cuenta la tabla de la actividad 2, realizá las siguientes consultas:

- a) Obtené toda la información de las facturas ordenadas primero por precio total (de menor a mayor) y luego por apellido del cliente (de la A a la Z).
- b) Obtené toda la información de las facturas ordenadas primero por fecha (de la más antigua a la más nueva) y luego por el nombre del cliente (de la A a la Z).
- c) Obtené toda la información de las facturas ordenadas primero por el precio total (del mayor precio al menor precio) y luego por el apellido del cliente (de la A a la Z).
- d) Obtené toda la información de las facturas ordenadas primero por la fecha (de la más nueva a la más antigua) y luego por la cantidad de productos (de mayor a menor) .

4.

Completá las siguientes consultas para limitar los resultados obtenidos:

- a) Se quieren obtener solo las primeras 5 filas de la tabla.
SELECT profesor, tema, duracion FROM clase LIMIT _____;
- b) Se quieren obtener solo las primeras 10 filas de la tabla
SELECT profesor, tema, duracion FROM clase LIMIT _____;
- c) Se quieren obtener solo de la fila 11 a la 20 de la tabla (en total 10 filas)
SELECT profesor, tema, duracion FROM clase LIMIT _____;

- d) Se quieren obtener solo de la fila 21 a la 50 (en total 30 filas)
SELECT profesor, tema, duracion FROM clase LIMIT _____;

5.

La siguiente tabla almacena información de cursos de Acamica. Tiene la siguiente estructura:

curso
id INT (PK) nombre VARCHAR(50) tema_principal VARCHAR(50) minutos INT

Realizá las siguientes consultas:

- a) Obtené toda la información de las primeras 10 filas de la tabla.
- b) Obtené toda la información de las primeras 20 filas de la tabla.
- c) Obtené toda la información desde la fila 5 a la 20.
- d) Obtené toda la información desde la fila 31 a la 50;
- e) Obtené toda la información desde la fila 16 a la 30.
- f) Obtené toda la información de la fila 20 a la 100.

6.

La siguiente tabla almacena información de las escenas grabadas de una película. Tiene la siguiente estructura:

escena
id INT (PK) titulo VARCHAR(50) cantidad_actores INT minutos INT presupuesto DECIMAL(10,2) fecha_grabacion DATE

Completá las siguientes consultas:

- a) Obtené toda la información de las filas 1 a 10 ordenadas por título de la A a la Z.
SELECT * FROM escena ORDER BY _____ LIMIT _____;
- b) Obtené toda la información de las filas 11 a 20 ordenadas por fecha de grabación (desde la más reciente a la más pasada)
SELECT * FROM escena ORDER BY _____ LIMIT _____;
- c) Obtené toda la información de las filas 21 a la 30 ordenadas por la cantidad de minutos, desde la escena más corta a la más larga.
SELECT * FROM escena ORDER BY _____ LIMIT _____;
- d) Obtené toda la información de las 5 escenas con más cantidad de actores.
SELECT * FROM escena ORDER BY _____ LIMIT _____;
- e) Obtené toda la información de las 15 escenas que se filmaron con menor presupuesto.
SELECT * FROM escena ORDER BY _____ LIMIT _____;
- f) Obtené toda la información de las 8 escenas más recientes
SELECT * FROM escena ORDER BY _____ LIMIT _____;
- g) Obtené toda la información de las 20 escenas más largas
SELECT * FROM escena ORDER BY _____ LIMIT _____;

7.

Determiná el valor de verdad (true o false) de las siguientes sentencias teniendo en cuenta las siguientes variables:

tenedores = 12 cucharas = 4 cuchillos = 16 platos = 20

- a) tenedores > 20
- b) cuchillos = 16
- c) platos < 45
- d) tenedores > 10 AND cuchillos > 10

- e) platos = 20 AND cucharas > 17
- f) tenedores < 5 OR cucharas = 4
- g) platos > 50 OR cuchillos > 50
- h) (tenedores = 12 AND cucharas > 2) AND platos < 3
- i) platos < 100 OR (tenedores > 25 AND cuchillos > 10)
- j) (tenedores > 10 AND cucharas > 10) OR (cuchillos > 2 AND platos > 2)
- k) (tenedores = 10 OR cucharas > 10 OR platos > 10) AND (tenedores = 14 OR cuchillos < 100)

8.

Teniendo en cuenta las siguientes variables, completá con los operadores lógicos que correspondan (AND u OR) para que las sentencias tengan el valor de verdad indicado:

momias = 6 fantasmas = 3 zombies = 12 vampiros = 20
--

- a) momias > 5 ___ fantasmas > 10 = TRUE
- b) zombies = 12 ___ vampiros > 50 = FALSE
- c) vampiros > 10 ___ momias < 12 = TRUE
- d) (momias > 8 ___ zombies < 17) AND vampiros > 13 = TRUE
- e) (vampiros > 10 OR zombies > 50) AND (zombies = 11 ___ momias > 3) = TRUE
- f) (vampiros = 10 ___ zombies > 10) AND (fantasmas > 1 OR momias > 35) = FALSE

9.

Esta tabla almacena información de los animales de un zoológico. Tiene la siguiente estructura:

animal
id INT (PK) especie VARCHAR(100) edad INT peso DECIMAL(6,2) altura DECIMAL(4,2) fecha_ingreso DATE

Completá las siguientes consultas:

- a) Obtené todos los animales cuya edad sea mayor a 10 y su altura menor a 0,45
SELECT * FROM animal WHERE _____;
- b) Obtené todos los animales cuya especie sea “mono”, su altura mayor a 0,75 y su peso mayor a 24
SELECT * FROM animal WHERE _____;
- c) Obtené todos los animales cuya fecha de ingreso sea anterior al 2008 o su especie sea “elefante”
SELECT * FROM animal WHERE _____;
- d) Obtené todos los animales cuya fecha de ingreso sea posterior al 2014 o que su edad sea 17 años.
SELECT * FROM animal WHERE _____;
- e) Obtené todos los animales cuyo peso sea mayor a 15 y su altura menor a 0,5 o sino, aquellos cuya edad sea mayor a 21.
SELECT * FROM animal WHERE _____;
- f) Obtené todos los animales cuya fecha de ingreso sea posterior a 2010 y su edad sea mayor a 12 o aquellos cuya altura es menor a 1,25 y su peso menor a 0,5.
SELECT * FROM animal WHERE _____;
- g) Obtené todos animales cuya especie es “hipopotamo” y su peso es mayor a 1,75 o aquellos cuya especie es “jirafa” y su altura es mayor a 2 o sino, aquellos cuya especie es “cebra” y su edad es igual a 17.
SELECT * FROM animal WHERE _____;

10.

Netflix quiere analizar el contenido que ven sus usuarios. Para esto deben analizar la información que hay almacenada en la base de datos sobre ellos. La estructura de la tabla que almacena a los usuarios de la aplicación es la siguiente:

usuario
id INT (PK) username VARCHAR(70) edad INT pais VARCHAR(60) películas_vistas INT capitulos_vistos INT fecha_registro DATE series_abandonadas INT genero_favorito VARCHAR(50)

Completá las consultas para obtener la información que corresponde:

- Obtené todos los usuarios que nunca abandonaron una serie y que vieron más de 70 capítulos
SELECT * FROM usuario WHERE _____;
- Obtené todos los de Ecuador cuyo género favorito es "Drama" y vieron más de 30 películas o los usuarios de México cuyo género favorito es "Comedia" y nunca abandonaron una serie.
SELECT * FROM usuario WHERE _____;
- Obtené todos los usuarios que tienen más de 35 años o se han registrado a la aplicación antes del 2016. Estos usuarios además tienen que ser de Chile o de Colombia.
SELECT * FROM usuario WHERE _____;
- Obtené todos los usuarios que residen en España, Francia o Alemania y se registraron en la aplicación antes del 2015 sin abandonar ninguna serie hasta el momento.
SELECT * FROM usuario WHERE _____;

- e) Obtén todos los usuarios mayores de 30 años que vieron más de 50 películas y más de 200 capítulos o los usuarios que son menores a 20 años que nunca abandonaron una serie.

SELECT * FROM usuario WHERE _____;

11.

La siguiente tabla almacena información de eventos, sería algo así como una agenda. Tiene la siguiente estructura:

evento
id INT (PK) nombre VARCHAR(60) fecha_inicio DATETIME acompañante VARCHAR(70) importancia INT

En esta actividad, vas a practicar consultas utilizando los operadores IN, LIKE y BETWEEN.

Recordá que el tipo de dato DATETIME almacena datos de el siguiente tiempo AAAA-MM-DD HH:MM:SS.

Completá las siguientes consultas:

- a) Obtén todos los eventos cuyo nombre empieza con la palabra “clase” para poder obtener por ejemplo “clase de salsa”, “clase de yoga”, etc.
SELECT * FROM evento WHERE _____ LIKE _____;
- b) Obtén todos los eventos cuya importancia va de 5 a 10.
SELECT * FROM evento WHERE _____ BETWEEN _____;
- c) Obtén todos los eventos que tengan como acompañante una persona cuyo nombre comience con “A”.
SELECT * FROM evento WHERE _____ LIKE _____;
- d) Obtén todos los eventos que inicien entre el 25 de mayo de 2015 a las 20:00 y el 17 de septiembre de 2015 a las 17:00.
SELECT * FROM evento WHERE _____ BETWEEN _____;

- e) Obtené todos los eventos cuyo nombre sea “curso de cocina”, “odontólogo” o “cumpleaños de Marcos”.
SELECT * FROM evento WHERE _____ IN _____;
- f) Realizá una consulta para saber si hay algún evento el 1 de octubre de 2017 a las 20:00, el 25 de noviembre de 2017 a las 17:00 o el 9 de diciembre de 2017 a las 15:00.
SELECT * FROM evento WHERE _____ IN _____

12.

En la siguiente tabla se almacena información de los gastos de compras en el supermercado. Vas a practicar los operadores COUNT, SUM y AVG realizando algunas consultas para obtener datos estadísticos.

compra
id INT (PK) gasto INT productos_comprados INT fecha_compra DATE

Realizá las siguientes consultas:

- Obtené la cantidad de compras realizadas. El resultado debe tener el nombre `compras_realizadas`.
- Obtené la cantidad total de dinero gastado en todas las compras. El resultado debe tener el nombre `gasto_total`.
- Obtené el promedio de dinero gastado en todas las compras. El resultado debe tener el nombre `promedio_gasto`.
- Obtené la cantidad total de productos comprados en todas las compras. El resultado debe tener el nombre `productos_totales`.
- Obtené el promedio de productos comprados en todas las compras. El resultado debe tener el nombre `promedio_productos`.

13.

Investigá como hacer para contar, sumar, o sacar promedios de los elementos de una tabla que cumplen con cierta condición.

Imaginate que existe la siguiente tabla que almacena información de partidos de fútbol:

partido
id INT (PK) fecha DATE cantidad_pases INT cantidad_goles INT publico_local INT publico_visitante

Luego de investigar, completá las siguientes consultas:

- Obtené la cantidad de partidos en la que hubo más de 1500 pases.
SELECT _____ AS muchos_pases FROM partido WHERE _____;
- Obtené la cantidad de partidos en la que hubo más de 5 goles.
SELECT _____ AS muchos_goles FROM partido WHERE _____;
- Obtené la cantidad de partidos en la que hubo más de 35.000 personas en el público local y más de 25.000 personas en el público visitante.
SELECT _____ AS mucho_publico FROM partido WHERE _____;
- Obtené el promedio de goles que se hicieron en los partidos que tuvieron más de 2000 pases.
SELECT _____ AS goles_muchos_pases FROM partido WHERE _____;
- Obtené la cantidad de pases que se hicieron en los partidos que tuvieron menos de 3 goles.
SELECT _____ AS pases_pocos_goles FROM partido WHERE _____;
- Obtené el promedio de goles que se hicieron en los partidos que tenían menos de 20.000 personas tanto en el público local como en el visitante.
SELECT _____ AS goles_poco_aliento FROM partido WHERE _____;

- g) Obtené la cantidad de publico local que hubo en los partidos que se hicieron más de 10 goles y más de 2000 pases.

SELECT _____ AS publico_muchos_goles FROM partido WHERE _____;

14.

Esta es una tabla que almacena información de todas las cuadras de una ciudad. De cada tabla se sabe información como cuantas casas tiene, cuando fue la última vez que se podaron los árboles de esa cuadra o quién fue el podador encargado.

cuadra
id INT (PK) calle VARCHAR(100) altura INT cantidad_viviendas INT cantidad_arboles INT cantidad_negocios INT ultima_poda DATE ultimo_podador VARCHAR(70)

Vas a realizar consultas “integradoras” utilizando todas las cláusulas y operadores que conoces. Algunas de ellas quizás tengan más de una forma de resolverse. En la solución a esta actividad vas a encontrar una solución posible.

La idea de esta actividad es que te focalices en cláusulas y operadores. Por eso, en cada consulta, vas a devolver toda la información de cada cuadra. En otras palabras, en todas utilizá SELECT *.

Realizá las siguientes consultas:

- Obtené todas las cuadras que se podaron por última vez luego del 24 de noviembre de 2016 ordenadas por la que tuvo la poda más reciente a la más antigua.
- Obtené todas las cuadras que tienen entre 12 y 15 árboles y entre 40 y 50 viviendas.
- Obtené todas las cuadras que fueron podadas por un podador llamado Tomás ordenadas por la cantidad de árboles, de mayor a menor. (Tene en cuenta que “Tomás” es solo el nombre del podador y en el campo “podador” aparece el nombre y apellido de ellos).
- Obtené todas las cuadras de la calle 'Lima' que tienen entre 10 y 15 negocios o las cuadras de la calle 'Caracas' que son podadas por Juan, Tomás o Víctor.

- e) Obtené las 10 cuadras con mayor cantidad de negocios
- f) Obtené todas las cuadras que de las calles 'Brasil', 'Estados Unidos' o 'Panamá' que tienen más de 10 negocios y menos de 50 viviendas.
- g) Obtené todas las cuadras que tienen o más de 30 viviendas o más de 15 árboles y fueron podadas después del 2015.

15.

Escribí el comando necesario para cambiarle el nombre a una tabla llamada 'alumno' por 'estudiante'.

16.

Escribí el comando necesario para agregarle una columna nueva a una tabla llamada 'turno'. La nueva columna se va a llamar 'fecha' y va a ser del tipo de dato DATE.

17.

Escribí el comando necesario para agregar una columna nueva a una tabla llamada 'comida'. La nueva columna va a llamarse 'ingrediente_principal' y va a ser del tipo de dato VARCHAR(100).

18.

Escribí el comando necesario para eliminar la columna 'tema_principal' de la tabla 'curso'.

19.

Escribí el comando necesario para modificar el tipo de dato de la columna 'descripcion' de la tabla 'producto'. El nuevo tipo de dato debe ser VARCHAR(150)

```
ALTER TABLE producto  
MODIFY COLUMN descripcion VARCHAR(150);
```

20.

Escribí el comando necesario para hacer que el campo 'codigo' de la tabla 'item' sea la clave primaria de la tabla.

21.

Escribí el comando necesario para eliminar una tabla llamada 'hechizo'

22.

Escribí el comando necesario para eliminar una tabla llamada 'musica'

23.

Escribí el comando necesario para crear una nueva columna llamada 'codigo' en la tabla 'item' y luego asignarlo como clave primaria. El tipo de dato de la nueva columna debe ser INT.

Soluciones

1.

- a) `SELECT apellido, nombre, edad FROM estudiante ORDER BY apellido ASC;`
- b) `SELECT apellido, nombre, edad FROM estudiante ORDER BY nombre DESC;`
- c) `SELECT apellido, nombre, edad FROM estudiante ORDER BY edad ASC;`
- d) `SELECT apellido, nombre, edad FROM estudiante ORDER BY edad DESC;`

2.

- a) `SELECT * from factura ORDER BY cliente_apellido;`
- b) `SELECT * from factura ORDER BY precio_total DESC;`
- c) `SELECT * from factura ORDER BY cantidad_productos;`
- d) `SELECT * from factura ORDER BY fecha DESC;`
- e) `SELECT * from factura ORDER BY cantidad_productos DESC;`

Observación: recordá que la cláusula `ORDER BY` ordena por defecto de manera ascendente. Es decir, que si en la consulta no se indica el tipo de orden, va a ordenar de esa manera. Es por eso que cuando el orden que corresponde en la consulta es de es tipo ascendente, no esta especificado.

3.

- a) `SELECT * FROM factura ORDER BY precio_total, cliente_apellido;`
- b) `SELECT * FROM factura ORDER BY fecha, cliente_nombre;`
- c) `SELECT * FROM factura ORDER BY precio_total DESC, cliente_apellido;`
- d) `SELECT * FROM factura ORDER BY fecha DESC, cantidad_productos DESC;`

4.

- a) `SELECT profesor, tema, duracion FROM clase LIMIT 0,5;`
- b) `SELECT profesor, tema, duracion FROM clase LIMIT 0,10;`
- c) `SELECT profesor, tema, duracion FROM clase LIMIT 10,10;`

- d)** `SELECT profesor, tema, duracion FROM clase LIMIT 20,30;`

Observación: tanto en la consulta a) como en la b) se podría hacer al siguiente modificación:

- a)** `SELECT profesor, tema, duracion FROM clase LIMIT 5;`
b) `SELECT profesor, tema, duracion FROM clase LIMIT 10;`

Si la cláusula LIMIT recibe solo un parámetro, en este caso el 5 o el 10, por defecto, devuelve desde la primera fila de la tabla. Si indicamos "LIMIT 5" significa que queremos obtener las primeras 5 filas de la tabla.

5.

- a)** `SELECT * FROM curso LIMIT 10;`
b) `SELECT * FROM curso LIMIT 20;`
c) `SELECT * FROM curso LIMIT 4,16;`
d) `SELECT * FROM curso LIMIT 30,20;`
e) `SELECT * FROM curso LIMIT 15,15;`
f) `SELECT * FROM curso LIMIT 19,81;`

6.

- a)** `SELECT * FROM escena ORDER BY titulo LIMIT 0,10;`
b) `SELECT * FROM escena ORDER BY fecha DESC LIMIT 10,10;`
c) `SELECT * FROM escena ORDER BY minutos LIMIT 20,10;`
d) `SELECT * FROM escena ORDER BY cantidad_actores DESC LIMIT 5;`
e) `SELECT * FROM escena ORDER BY presupuesto LIMIT 15;`
f) `SELECT * FROM escena ORDER BY fecha DESC LIMIT 8;`
g) `SELECT * FROM escena ORDER BY minutos DESC LIMIT 20;`

7.

- a)** F
b) T
c) T
d) T
e) F
f) T
g) F
h) F

- i) F
- j) T
- k) T

8.

- a) momias > 5 **OR** fantasmas > 10 = TRUE
- b) zombies = 12 **AND** vampiros > 50 = FALSE
- c) Los dos operadores dan el mismo valor de verdad:
 - i) vampiros > 10 **AND** momias < 12 = TRUE
 - ii) vampiros > 10 **OR** momias < 12 = TRUE
- d) (momias > 8 **OR** zombies < 17) AND vampiros > 13 = TRUE
- e) (vampiros > 10 OR zombies > 50) AND (zombies = 11 **OR** momias > 3) = TRUE
- f) (vampiros = 10 **AND** zombies > 10) AND (fantasmas > 1 OR momias > 35) = FALSE

9.

- a) SELECT * FROM animal WHERE edad > 10 AND altura < 0,45;
- b) SELECT * FROM animal WHERE especie = 'mono' AND altura > 0,75 AND peso > 24;
- c) SELECT * FROM animal WHERE fecha_ingreso < '2008-01-01' OR especie = 'elefante';
- d) SELECT * FROM animal WHERE fecha_ingreso > '2014-01-01' OR edad = 17;
- e) SELECT * FROM animal WHERE peso > 15 AND altura < 0,5 OR edad > 21;
- f) SELECT * FROM animal WHERE (fecha_ingreso > '2010-01-01' AND edad > 12) OR (altura < 1,25 AND peso < 0,5);
- g) SELECT * FROM animal WHERE (especie = 'hipopotamo' AND peso > 1,75) OR (especie = 'jirafa' AND altura > 2) OR (especie = 'cebra' AND edad = 17);

10.

- a) SELECT * FROM usuario WHERE series_abandonadas = 0 AND capitulos_vistos > 70;
- b) SELECT * FROM usuario WHERE (pais = 'Ecuador' AND genero_favorito = 'Drama' AND peliculas_vistas > 30) OR (pais = 'México' AND genero_favorito = 'Comedia' AND series_abandonadas = 0);
- c) SELECT * FROM usuario WHERE (edad > 35 OR fecha_registro < '2016-01-01') AND (pais = 'Chile' OR pais = 'Colombia');
- d) SELECT * FROM usuario WHERE (pais = 'España' OR pais = 'Francia' OR pais = 'Alemania') AND fecha_registro < '2015-01-01' AND series_abandonadas = 0;
- e) SELECT * FROM usuario WHERE (edad > 30 AND peliculas_vistas > 50 AND capitulos_vistos > 200) OR (edad > 20 AND series_abandonadas = 0);

11.

- a) SELECT * FROM evento WHERE nombre LIKE 'clase%';
- b) SELECT * FROM evento WHERE importancia BETWEEN 5 AND 10;
- c) SELECT * FROM evento WHERE acompañante LIKE 'c%';
- d) SELECT * FROM evento WHERE fecha_inicio BETWEEN '2015-05-25 20:00:00' AND '2015-09-17 17:00:00';
- e) SELECT * FROM evento WHERE nombre IN ("curso de cocina", "odontólogo" o "cumpleaños de Marcos");
- f) SELECT * FROM evento WHERE fecha_inicio IN ('2017-01-01 20:00:00' , '2017-11-25 17:00:00', '2017-12-09 15:00:00');

12.

- a) SELECT COUNT(*) AS compras_realizadas FROM compra;
- b) SELECT SUM(gasto) AS gasto_total FROM compra;
- c) SELECT AVG(gasto) AS promedio_gastol FROM compra;
- d) SELECT SUM(productos_comprados) AS productos_totales FROM compra;
- e) SELECT AVG(productos_comprados) AS promedio_productos FROM compra;

13.

- a) SELECT COUNT(*) AS muchos_pases FROM partido WHERE cantidad_pases > 1500;
- b) SELECT COUNT(*) AS muchos_goles FROM partido WHERE cantidad_goles > 5;
- c) SELECT COUNT(*) AS mucho_publico FROM partido WHERE publico_local > 35.000 AND publico_visitante > 25.000;
- d) SELECT AVG(cantidad_goles) AS goles_muchos_pases FROM partido WHERE cantidad_pases > 2000;
- e) SELECT SUM(cantidad_pases) AS pases_pocos_goles FROM partido WHERE cantidad_goles < 3;
- f) SELECT AVG(cantidad_goles) AS goles_poco_aliento FROM partido WHERE publico_local < 20.000 AND publico_visitante < 20.000;
- g) SELECT SUM(publico_local) AS publico_muchos_goles FROM partido WHERE cantidad_goles > 10 AND cantidad_pases > 2000;

14.

- a) `SELECT * FROM cuadra WHERE ultima_poda > '2016-11-24' ORDER BY ultima_poda DESC;`
- b) `SELECT * FROM cuadra WHERE cantidad_arboles BETWEEN 12 AND 15 AND cantidad_viviendas BETWEEN 40 AND 50;`
- c) `SELECT * FROM cuadra WHERE podador LIKE 'Tomás%' ORDER BY cantidad_arboles DESC;`
- d) `SELECT * FROM cuadra WHERE (calle = 'Lima' AND negocios BETWEEN 10 AND 15) OR (calle = 'Caracas' AND podador IN ('Juan', 'Tomás', 'Victor'));`
- e) `SELECT * FROM cuadra ORDER BY cantidad_negocios DESC LIMIT 10;`
- f) `SELECT * FROM cuadra WHERE calle IN ('Brasil', 'Estados Unidos', 'Panamá') AND cantidad_negocios > 10 AND cantidad_viviendas < 50;`
- g) `SELECT * FROM cuadra WHERE (cantidad_viviendas > 30 OR cantidad_arboles > 15) AND ultima_poda > '2015-01-01';`

15.

```
ALTER TABLE alumno  
RENAME TO estudiante;
```

16.

```
ALTER TABLE turno  
ADD COLUMN fecha DATE;
```

17.

```
ALTER TABLE comida  
ADD COLUMN ingrediente_principal VARCHAR(100);
```

18.

```
ALTER TABLE curso  
DROP COLUMN tema_principal;
```

19.

```
ALTER TABLE producto  
MODIFY COLUMN descripcion VARCHAR(150);
```

20.

```
ALTER TABLE item  
ADD PRIMARY KEY codigo;
```

21.

```
DROP TABLE hechizo;
```

22.

```
DROP DATABASE musica;
```

23.

```
ALTER TABLE item  
ADD COLUMN codigo INT;
```

```
ALTER TABLE item  
ADD PRIMARY KEY codigo;
```