SELECT \*

FROM `acceso`

SELECT cedula, contra

FROM `acceso`

SELECT DISTINCT cedula

FROM `acceso`

SELECT COUNT(DISTINCT cedula)

FROM `acceso`

SELECT \* FROM acceso

WHERE usuario = 'oswaldo'

SELECT \*

FROM acceso

WHERE cedula = 1917

SELECT \*

FROM acceso

WHERE usuario = 'oswaldo'

AND cedula = 1917

SELECT \*

FROM acceso

WHERE usuario = 'oswaldo'

OR cedula = 1917

SELECT \*

FROM acceso

WHERE

NOT usuario = 'oswaldo'

SELECT \*

FROM acceso

WHERE usuario = 'oswaldo'

AND (cedula = 1917 OR cedula = 0)

SELECT \*

FROM acceso

WHERE NOT cedula = 1917

AND NOT usuario = 'oswaldo'

SELECT \*

FROM acceso

ORDER BY contra

SELECT \*

FROM acceso

ORDER BY contra DESC

SELECT \*

FROM acceso

ORDER BY cedula, contra

SELECT \*

FROM acceso

ORDER BY contra ASC, cedula DESC

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO cheque (beneficiario, codigo, fecha, montoletras, nmonto)

[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) ('Tom B. Erichsen', '21456794', '05 de Septiembre de 2016', 'Seiscientos Ochenta mil', '87000')

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO cheque (beneficiario, codigo, fecha)

[VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) ('Tom B. Erichsen', '21456794', '05 de Septiembre de 2016')

SELECT beneficiario, codigo, fecha

FROM cheque

WHERE montoletras

IS NULL

SELECT beneficiario, codigo, fecha

FROM cheque

WHERE montoletras

IS IS NOT NULL

UPDATE cheque

SET nmonto = '48000', montoletras = 'Cuarenta y Ocho Mil'

WHERE codigo = 21456795

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) medico

[SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) nombre = 'Orlando'

WHERE codigo = 1278

UPDATE medico

SET codigo = 8880;

DELETE FROM cheque

SELECT \*

FROM cheque

LIMIT 3

SELECT \*

FROM administrador

WHERE tipousuario='0'

LIMIT 2

SELECT MIN(contraa)

AS SmallContra

FROM administrador

SELECT MAX(contraa)

AS LargeContra

FROM administrador

SELECT COUNT(cedula)

FROM administrador

SELECT AVG(contraa)

FROM administrador

SELECT SUM(nmonto)

FROM cheque

SELECT \*

FROM cheque

WHERE beneficiario LIKE 't%'

SELECT \*

FROM cheque

WHERE beneficiario LIKE '%en'

SELECT \*

FROM cheque

WHERE beneficiario LIKE '%a%'

SELECT \*

FROM cheque

WHERE beneficiario LIKE '\_o%'

SELECT \* FROM cheque

WHERE beneficiario LIKE 't\_\_%'

SELECT \*

FROM cheque

WHERE beneficiario LIKE 'o%n'

SELECT \*

FROM cheque

WHERE beneficiario NOT LIKE 'a%'

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \*

FROM cheque

WHERE beneficiario [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) 'tom%'

SELECT \*

FROM cheque

WHERE montoletras LIKE '%en%'

SELECT \*

FROM cheque

WHERE beneficiario LIKE '\_ondon'

SELECT \*

FROM cheque

WHERE beneficiario LIKE 'O\_wa\_do'

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html)

\* FROM cheque

WHERE codigo [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ('21456794', '21456795')

SELECT \*

FROM cheque

WHERE codigo NOT IN ('21456794', '21456795')

SELECT \*

FROM Customers  
WHERE Country IN (SELECT Country FROM Suppliers)

SELECT \*

FROM cheque

WHERE nmonto

BETWEEN 50000

AND 500000

SELECT \*

FROM cheque

WHERE nmonto

NOT BETWEEN 50000

AND 500000

SELECT \*

FROM cheque

WHERE nmonto

BETWEEN 48000 AND 500000

AND codigo

NOT IN (1,2,3)

SELECT \*

FROM cheque

WHERE beneficiario

BETWEEN 'Oswaldo Marin'

AND 'Tomas Bruce Erichsen'

ORDER BY beneficiario

SELECT \*

FROM cheque

WHERE beneficiario

NOT BETWEEN 'Oswaldo Marin'

AND 'Tomas Bruce Erichsen'

ORDER BY beneficiario

SELECT \*

FROM cheque

WHERE fecha

BETWEEN '05 de Septiembre de 2016'

AND '17 de Noviembre de 2017'

SELECT codigo AS CODE, beneficiario AS Usuario

FROM cheque

SELECT montoletras AS "Descripcion", nmonto AS "monto"

FROM cheque

SELECT CustomerName, CONCAT\_WS(', ', Address, PostalCode, City, Country) AS Address  
FROM Customers;

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName  
FROM ((Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)  
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;

SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders  
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
ORDER BY Orders.OrderID;

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
CROSS JOIN Orders;

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
CROSS JOIN Orders  
WHERE Customers.CustomerID=Orders.CustomerID;

SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
FROM Customers A, Customers B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City  
ORDER BY A.City;

SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;

SELECT City FROM Customers  
UNION ALL  
SELECT City FROM Suppliers  
ORDER BY City;

SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;

SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION ALL  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;

SELECT 'Customer' AS Type, ContactName, City, Country  
FROM Customers  
UNION  
SELECT 'Supplier', ContactName, City, Country  
FROM Suppliers;

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country;

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;

SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders

FROM Orders

LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID

GROUP BY ShipperName;

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5;

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;

SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders  
FROM (Orders  
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID)  
GROUP BY LastName  
HAVING COUNT(Orders.OrderID) > 10;

SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders  
FROM Orders  
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
WHERE LastName = 'Davolio' OR LastName = 'Fuller'  
GROUP BY LastName  
HAVING COUNT(Orders.OrderID) > 25;

SELECT SupplierName  
FROM Suppliers  
WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);

SELECT SupplierName  
FROM Suppliers  
WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price = 22);

SELECT ProductName  
FROM Products  
WHERE ProductID = ANY  
  (SELECT ProductID  
  FROM OrderDetails  
  WHERE Quantity = 10);

SELECT ProductName  
FROM Products  
WHERE ProductID = ANY  
  (SELECT ProductID  
  FROM OrderDetails  
  WHERE Quantity > 99);

SELECT ProductName  
FROM Products  
WHERE ProductID = ANY  
  (SELECT ProductID  
  FROM OrderDetails  
  WHERE Quantity > 1000);

SELECT ProductName  
FROM Products  
WHERE ProductID = ALL  
  (SELECT ProductID  
  FROM OrderDetails  
  WHERE Quantity = 10);

INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers;

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)  
SELECT SupplierName, ContactName, Address, City, PostalCode, Country FROM Suppliers;

INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers  
WHERE Country='Germany';

SELECT OrderID, Quantity,  
CASE  
    WHEN Quantity > 30 THEN 'The quantity is greater than 30'  
    WHEN Quantity = 30 THEN 'The quantity is 30'  
    ELSE 'The quantity is under 30'  
END AS QuantityText  
FROM OrderDetails;

SELECT CustomerName, City, Country  
FROM Customers  
ORDER BY  
(CASE  
    WHEN City IS NULL THEN Country  
    ELSE City  
END);

SELECT ProductName, UnitPrice \* (UnitsInStock + IFNULL(UnitsOnOrder, 0))  
FROM Products;

SELECT ProductName, UnitPrice \* (UnitsInStock + COALESCE(UnitsOnOrder, 0))  
FROM Products;

-- Select all:  
SELECT \* FROM Customers;

SELECT \* FROM Customers -- WHERE City='Berlin';

-- SELECT \* FROM Customers;  
SELECT \* FROM Products;

/\*Select all the columns  
of all the records  
in the Customers table:\*/  
SELECT \* FROM Customers;

/\*SELECT \* FROM Customers;  
SELECT \* FROM Products;  
SELECT \* FROM Orders;  
SELECT \* FROM Categories;\*/  
SELECT \* FROM Suppliers;

SELECT CustomerName, /\*City,\*/ Country FROM Customers;

SELECT \* FROM Customers WHERE (CustomerName LIKE 'L%'  
OR CustomerName LIKE 'R%' /\*OR CustomerName LIKE 'S%'  
OR CustomerName LIKE 'T%'\*/ OR CustomerName LIKE 'W%')  
AND Country='USA'  
ORDER BY CustomerName;

CREATE DATABASE testDB;

SHOW DATABASES;

DROP DATABASE testdb;

CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)   
);

CREATE TABLE TestTable AS  
SELECT customername, contactname  
FROM customers;

DROP TABLE table\_name; //Elimina una tabla

TRUNCATE TABLE table\_name; //Elimina datos dentro de una tabla

ALTER TABLE Customers  
ADD Email varchar(255);

ALTER TABLE Customers  
DROP COLUMN Email;

ALTER TABLE Persons  
ADD DateOfBirth date;

ALTER TABLE Persons  
MODIFY COLUMN DateOfBirth year;

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);

ALTER TABLE Persons  
MODIFY Age int NOT NULL;

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    UNIQUE (ID)  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT UC\_Person UNIQUE (ID,LastName)  
);

ALTER TABLE Persons  
ADD UNIQUE (ID);

ALTER TABLE Persons  
ADD CONSTRAINT UC\_Person UNIQUE (ID,LastName);

ALTER TABLE Persons  
DROP INDEX UC\_Person;

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

ALTER TABLE Persons  
ADD CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName);

ALTER TABLE Persons  
DROP PRIMARY KEY;

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    CONSTRAINT FK\_PersonOrder FOREIGN KEY (PersonID)  
    REFERENCES Persons(PersonID)  
);

ALTER TABLE Orders  
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

ALTER TABLE Orders  
ADD CONSTRAINT FK\_PersonOrder  
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

ALTER TABLE Orders  
DROP FOREIGN KEY FK\_PersonOrder;

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18)  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255),  
    CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes')  
);

ALTER TABLE Persons  
ADD CHECK (Age>=18);

ALTER TABLE Persons  
ADD CONSTRAINT CHK\_PersonAge CHECK (Age>=18 AND City='Sandnes');

ALTER TABLE Persons  
DROP CHECK CHK\_PersonAge;

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes'  
);

CREATE TABLE Orders (  
    ID int NOT NULL,  
    OrderNumber int NOT NULL,  
    OrderDate date DEFAULT CURRENT\_DATE()  
);

ALTER TABLE Persons  
ALTER City SET DEFAULT 'Sandnes';

ALTER TABLE Persons  
ALTER City DROP DEFAULT;

CREATE INDEX idx\_lastname  
ON Persons (LastName);

CREATE INDEX idx\_pname  
ON Persons (LastName, FirstName);

CREATE TABLE Persons (  
    Personid int NOT NULL AUTO\_INCREMENT,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (Personid)  
);

ALTER TABLE Persons AUTO\_INCREMENT=100;

INSERT INTO Persons (FirstName,LastName)  
VALUES ('Lars','Monsen');

SELECT \* FROM Orders WHERE OrderDate='2008-11-11'

CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';

SELECT \* FROM [Brazil Customers];

CREATE VIEW [Products Above Average Price] AS  
SELECT ProductName, Price  
FROM Products  
WHERE Price > (SELECT AVG(Price) FROM Products);

SELECT \* FROM [Products Above Average Price];

CREATE OR REPLACE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName, City  
FROM Customers  
WHERE Country = 'Brazil';

DROP VIEW [Brazil Customers];