**Database**

* 1. What is a database?

Database is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organized in the form of tables, rows and columns.

* 1. What is a table?

In relational **databases**, and flat file **databases**, a **table** is a set of data elements (values) using a model of vertical columns (identifiable by name) and horizontal rows, the cell being the unit where a row and column intersect. A **table** has a specified number of columns, but can have any number of rows.

create table CustomersShoppingList ('CustomerID','CustomerName', 'CustomerContact', 'PurchasedItem', 'ItemsPrice');

* 1. What is a column?

The columns in a table are the set of facts that we keep track of about that type of object. A column is also called an attribute.

* 1. What is a row?

Each row in a database table represents one instance of the type of object described in that table. A row is also called a record.

* 1. What is inner join?

The INNER JOIN keyword selects records that have matching values in both tables.

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

Example:

* 1. What is left outer join?

In some databases Left outer join is also called left join. The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

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

* 1. What is right outer join?

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

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

* 1. Example of Max, Sum and avg?

The MAX() function returns the largest value of the selected column.

SELECT MAX(column\_name)  
FROM table\_name  
WHERE condition;

The SUM() function returns the total sum of a numeric column.

The following SQL statement finds the sum of the "Quantity" fields in the "OrderDetails" table:

SELECT SUM(Quantity)  
FROM OrderDetails;

The AVG() function returns the average value of a numeric column.

The following SQL statement finds the average price of all products:

SELECT AVG(Price)  
 FROM Products;

* 1. Example of group by?

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

The following SQL statement lists the number of customers in each country:

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

The following SQL statement lists the number of customers in each country, sorted high to low:

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

The following SQL statement lists the number of orders sent by each shipper:

SELECT Shippers.ShipperName, COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders  
LEFT JOIN Shippers ON Orders.ShipperID=Shippers.ShipperID  
GROUP BY ShipperName;

* 1. Example of having?

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions. The following SQL statement lists the number of customers in each country. Only include countries with more than 5 customers:

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

Here customer from the country is repeated will be counted which has to be >5. Then that repeated country is written as count.

The following SQL statement lists the number of customers in each country, sorted high to low (Only include countries with more than 5 customers):

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

Here customer from the country is repeated will be counted which has to be >5. Then that repeated country is written as count and the customer with highest count is placed in descending order.

The following SQL statement lists the employees that have registered more than 10 orders:

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;

The following SQL statement lists if the employees "Davolio" or "Fuller" have registered more than 25 orders:

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;

* 1. Example for where condition?

The WHERE clause is used to filter records. The WHERE clause is used to extract only those records that fulfill a specified condition.

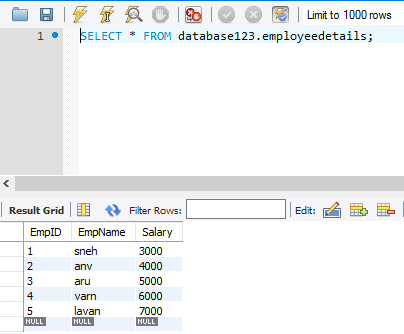
The following SQL statement selects all the customers from the country "Mexico", in the "Customers" table:

SELECT \* FROM Customers  
WHERE Country='Mexico';

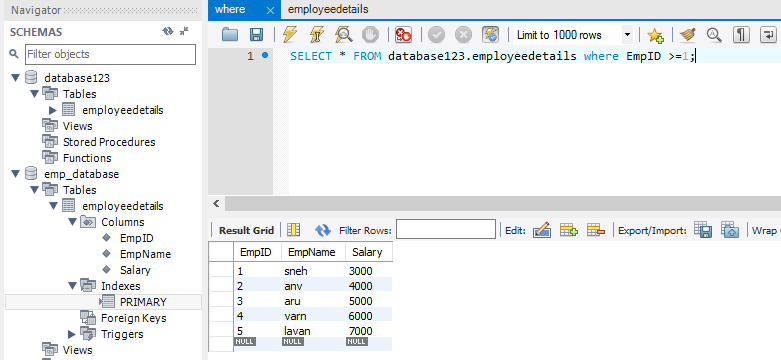
SQL requires single quotes around text values (most database systems will also allow double quotes). However, numeric fields should not be enclosed in quotes:

SELECT \* FROM Customers  
WHERE CustomerID=1;

Example:



SELECT \* FROM database123.employeedetails where EmpID >= 1;



* 1. Examples of primary key?

The PRIMARY KEY constraint uniquely identifies each record in a database table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only one primary key, which may consist of single or multiple fields.

The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:

SQL primary key on create table:

MySQL

CREATE TABLE Persons(IDint NOT NULL,LastName varchar(255) NOT NULL,  
FirstNamevarchar(255),Ageint,PRIMARY KEY (ID));

SQL server/Oracle/Access

CREATE TABLE Persons(IDint NOT NULL PRIMARY KEY,LastName varchar(255) NOT NULL,FirstNamevarchar(255),Ageint);

MySQL/SQL Server/Oracle/MS Access

CREATE TABLE Persons(IDint NOT NULL,LastName varchar(255) NOT NULL,  
FirstNamevarchar(255),Ageint,CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName));

SQL primary key on alter table:

To create a PRIMARY KEY constraint on the "ID" column when the table is already created, use the following SQL:

MySQL/SQL server/Oracle/Access

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

To allow naming of a PRIMARY KEY constraint, and for defining a PRIMARY KEY constraint on multiple columns, use the following SQL syntax:

MySQL/SQL server/Oracle/Access

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

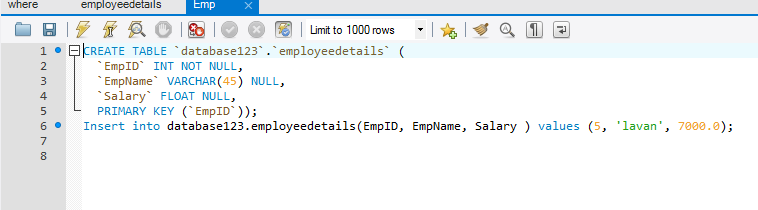
To drop a PRIMARY KEY constraint, use the following SQL:

MySQL

ALTER TABLE Persons  
DROP PRIMARY KEY;

SQL Server/Oracle/Access

ALTER TABLE Persons  
DROP CONSTRAINT PK\_Person;



* 1. Example of foreign key?

A FOREIGN KEY is a key used to link two tables together. A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table. The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

SQL foreign key on create table:

The following SQL creates a FOREIGN KEY on the "PersonID" column when the "Orders" table is created:

MySQL

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

SQL Server/Oracle/Access

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

MySQL/SQL Server/Access/Oracle

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

SQL foreign key on alter table:

To create a FOREIGN KEY constraint on the "PersonID" column when the "Orders" table is already created, use the following SQL:

MySQL/SQL Server/Oracle/MS Access

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

MySQL/SQL Server/Oracle/MS Access

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

Drop a foreign key constraint:

To drop a FOREIGN KEY constraint, use the following SQL:

ALTER TABLE Orders  
DROP FOREIGN KEY FK\_PersonOrder;

SQL Server/Oracle/Access

ALTER TABLE Orders  
DROP CONSTRAINT FK\_PersonOrder;

* 1. Finding second highest salary from a row table?

Example: Created database database123 and created an employee table as employeedetails.

