|  |
| --- |
| **What is Database ?**  A database is a collection that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images. |
| **What is Table ?**  Each **table** is made up of rows and columns. If you think of a **table** as a grid, the column go from left to right across the grid and each entry of data is listed down as a row. |
| **What is Column ?**  A column is a set of data values of a particular Simple type, one for each row of the table |
| **What is Row ?**  A primary key uniquely identifies each row in a relational. |
|  |
| **Example for Inner join ?**  SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  FROM Orders  INNER JOIN Customers  ON Orders.CustomerID=Customers.CustomerID; |
| **Example for Left outer join ?**  SELECT Customers.CustomerName, Orders.OrderID  FROM Customers  LEFT JOIN Orders  ON Customers.CustomerID=Orders.CustomerID  ORDER BY Customers.CustomerName; |
| **Example for Right outer join ?**  SELECT *column\_name(s)*  FROM *table1*  RIGHT OUTER JOIN *table2*  ON *table1.column\_name*=*table2.column\_name*; |
| **Example for Max, sun, Avg** |
| **Example for Group by ?**  SELECT Shippers.ShipperName,COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders  LEFT JOIN Shippers  ON Orders.ShipperID=Shippers.ShipperID  GROUP BY ShipperName; |
| **Example for Having ?**  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; |
| **Example for Where condition ?**  SELECT \* FROM Customers  WHERE Country='Mexico'; |
| **Example for Primary key ?**  CREATE TABLE Persons  (  P\_Id int NOT NULL PRIMARY KEY,  LastName varchar(255) NOT NULL,  FirstName varchar(255),  Address varchar(255),  City varchar(255)  ) |
| **Example for Foreign key ?**  CREATE TABLE Orders  (  O\_Id int NOT NULL,  OrderNo int NOT NULL,  P\_Id int,  PRIMARY KEY (O\_Id),  FOREIGN KEY (P\_Id) REFERENCES Persons(P\_Id)  ) |
|  |
| **Finding second highest salary from row table ?**  select \* FROM ( select EmployeeID, **Salary** ,rank() over (order by **Salary** DESC) ranking from Employee ) WHERE ranking = N; The rank function will assign a ranking to each row starting from 1. |
|  |
|  |
|  |