**DBMS:**

**3. What is Database?**

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated.

In one view, databases can be classified per types of content: bibliographic, full-text, numeric, and images.

The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

In computing, databases are sometimes classified per their organizational approach.

The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in several different ways.

......................................................................................................................

**4. What is Table?**

A relational database is made up of several components, of which the table is most significant.

The database table is where all the data in a database is stored, and without tables, there would not be much use for relational databases.

A database consists of one or more tables. 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.

......................................................................................................................

**5. What is Column?**

Columns are defined to hold a specific type of data, such as dates, numeric, or textual data.

In the simplest of definitions, a column is defined by its name and data type. The name is used in SQL statements when selecting and ordering data, and the data type is used to validate information stored.

......................................................................................................................

**6. What is row?**

A table can contain zero or more rows. When there are zero, it said to be empty.

There is not practical limit on the number of rows a table can hold; however, remember the table’s primary key may have some influence on this.

There is no guarantee that the rows in a table are stored in an order. Use the ORDER BY clause to do so.

......................................................................................................................

**8. Example for Inner join?**

The INNER JOIN keyword selects all rows from both tables if there is a match between the columns in both tables.

**Syntax:**

SELECT column\_name(s)

FROM table1

INNER JOIN table2

ON table1.column\_name=table2.column\_name;

Customers table:

|  |  |  |  |
| --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **Address** | **City** |
| 1 | Spandana | Obere Str. 57 | Berlin |
| 2 | Srikanth | Mataderos 2312 | Chicago |
| 3 | Apsa | Big rapids 4930 | Grand rapids |
| 4 | Dhruva | Marsh lane 2250 | Dallas |

Order table:

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | CustomerID | EmployeeID | OrderDate |
| 1 | 1 | 5 | 2012/23/04 |
| 2 | 2 | 7 | 2013/23/04 |
| 3 | 3 | 6 | 2015/23/04 |

SELECT Customers.CustomerName, Orders.OrderID

FROM Customers

INNER JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

ORDER BY Customers.CustomerName;

|  |  |
| --- | --- |
| CustomerName | OrderID |
| Spandana | 1 |
| Srikanth | 2 |
| Apsa | 3 |

......................................................................................................................

**9. Example for Left outer join?**

LEFT JOIN performs a join starting with the first (left-most) table and then any matching second (right-most) table records. LEFT JOIN and LEFT OUTER JOIN are the same.

**Syntax:**

SELECT column-names

FROM table-name1 LEFT JOIN table-name2

ON column-name1 = column-name2

WHERE condition

Customers table:

|  |  |  |  |
| --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **Address** | **City** |
| 1 | Spandana | Obere Str. 57 | Berlin |
| 2 | Srikanth | Mataderos 2312 | Chicago |
| 3 | Apsa | Big rapids 4930 | Grand rapids |
| 4 | Dhruva | Marsh lane 2250 | Dallas |

Order table:

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | CustomerID | EmployeeID | OrderDate |
| 1 | 2 | 5 | 2012/23/04 |
| 2 | 3 | 7 | 2013/23/04 |
| 3 | 4 | 6 | 2015/23/04 |

SELECT Customers.CustomerName, Orders.OrderID

FROM Customers 9

LEFT JOIN Orders

ON Customers.CustomerID=Orders.CustomerID

ORDER BY Customers.CustomerName;

|  |  |
| --- | --- |
| CustomerName | OrderID |
| Spandana | null |
| Srikanth | 1 |
| Apsa | 2 |
| Dhruva | 3 |

......................................................................................................................

**10. Example for Right outer join?**

The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.

**Syntax:**

SELECT column\_name(s)

FROM table1

RIGHT JOIN table2

ON table1.column\_name=table2.column\_name;

Order table:

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | CustomerID | EmployeeID | OrderDate |
| 1 | 2 | 5 | 2012/23/04 |
| 2 | 3 | 7 | 2013/23/04 |
| 3 | 4 | 6 | 2015/23/04 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **Address** | **City** |
| 1 | Spandana | Obere Str. 57 | Berlin |
| 2 | Srikanth | Mataderos 2312 | Chicago |
| 3 | Apsa | Big rapids 4930 | Grand rapids |
| 4 | Dhruva | Marsh lane 2250 | Dallas |

SELECT Orders.OrderID, Custmoers.CustomerName

FROM Orders

RIGHT JOIN Customers

ON Orders. Customers ID= Customers. Customers ID

ORDER BY Orders.OrderID;

|  |  |
| --- | --- |
| OrderID | CustomerName |
| 1 | Spandhana |
| 2 | Srikanth |
| 3 | Apsa |
|  | Dhruva |

......................................................................................................................

**11. Example for Max, sun, Avg?**

|  |  |  |  |
| --- | --- | --- | --- |
| **OrderId** | 1 | 2 | 3 |
| **CustomerId** | 321 | 455 | 456 |
| **Total** | 10 | 40 | 20 |
| **OrderDate** | 1/2/2014 | 3/2/2014 | 3/10/2014 |

In this example, you want to sum up the total orders for the month of March. The following SQL statement would sum up these orders.

SELECT SUM(Total) FROM Orders

WHERE OrderDate BETWEEN ‘3/1'2014' AND ‘3/31/2014' . The return value from the above SQL statement is 60.

The following SQL statement shows you how to use the AVG function.

SELECT AVG(Total) FROM Orders

WHERE OrderDate BETWEEN ‘3/1'2014' AND ‘3/31/2014'. The new value returned by the AVG function is 30.

Suppose you want to find the highest order total in your whole Order table. The following SQL statement gets the highest order.

SELECT MAX(Total) FROM Order

The result of the above statement is "40."

......................................................................................................................

**12. Example for Group by**

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

**Syntax:**

SELECT column\_name, aggregate\_function(column\_name)

FROM table name

WHERE column\_name operator value

GROUP BY column\_name;

**Orders Table:**

**OrderID CustomerID EmployeeID OrderDate ShipperID**

10248 90 5 1996-07-04 3

10249 81 6 1996-07-05 1

10250 34 4 1996-07-08 2

**Shippers table:**

**ShipperID ShipperName**

1 Speedy Express

2 United Package

3 Federal Shipping

**Employee Table:**

**EmployeeID LastName FirstName BirthDate**

1 Davolio Nancy 1968-12-08

2 Fuller Andrew 1952-02-19

3 Leverling Janet 1963-08-30

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

LEFT JOIN Shippers

ON Orders.ShipperID=Shippers.ShipperID

GROUP BY ShipperName;

**ShipperName NumberOfOrders**

Federal Shipping 68

Speedy Express 54

United Package 74

......................................................................................................................

**13. Example for Having**

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

**Syntax:**

SELECT column\_name, aggregate\_function(column\_name)

FROM table\_name

WHERE column\_name operator value

GROUP BY column\_name

HAVING aggregate\_function(column\_name) operator value;

**Orders Table:**

**OrderID CustomerID EmployeeID OrderDate ShipperID**

10248 90 5 1996-07-04 3

10249 81 6 1996-07-05 1

10250 34 4 1996-07-08 2

**Employee Table:**

**EmployeeID LastName FirstName BirthDate**

1 Davolio Nancy 1968-12-08

2 Fuller Andrew 1952-02-19

3 Leverling Janet 1963-08-30

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;

**LastName NumberOfOrders**

Buchanan 11

Callahan 27

Davolio 29

Fuller 20

......................................................................................................................

**14. 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 criterion.

**Syntax:**

SELECT column\_name,column\_name

FROM table\_name

WHERE column\_name operator value;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **Address** | **City** | **State** |
| 1 | Spandana | Obere Str. 57 | Berlin | Texas |
| 2 | Srikanth | Mataderos 2312 | Chicago | Michigan |
| 3 | Apsa | Big rapids 4930 | Grand rapids | Texas |
| 4 | Dhruva | Marsh lane 2250 | Dallas | Texas |

SELECT \* FROM Customers

WHERE State='Mexico';

**15. Example for Primary key**

The PRIMARY KEY constraint uniquely identifies each record in a database table. The Primary keys must contain UNIQUE values. A primary key column cannot contain NULL values. Most tables should have a primary key, and each table can have only ONE primary key.

......................................................................................................................

**16. Example for Foreign key**

A FOREIGN KEY in one table points to a PRIMARY KEY in another table.

………………………………………………………………………………………………………………….

**18. Finding second highest salary from row table**

select sal from

(select rownum n,a.\* from

(select distinct sal from emp order by sal desc) a)

where n = 2;