What is DBMS?

It is database management system used to store,retrive,update,delete data with the help of queries.

It is Organised collection of Data.

SQL-Structured Query language

It has % commands as Follows

1) **DDL-Data definition Language**
2) **DQL-Data Query Language**
3) **DML-Data manipulation language**
4) **DCL-Data Control Language**
5) **TCL-Transaction Control language**

Data Defination Language-DDL:

It has 6 commands

1) **Create**-Used to create database,table
   Ex. create database database_name;
   Create table table_name;
2) **Alter**-Used to change structure of database
   Ex. Alter table

3) **Drop-**Used to delete object or database
   Ex. Drop table table_name;
4) **Truncate**-Used to remove all records from table
   Ex. Truncate table table_name;
5) **Comment**-used to Add comment
6) **Rename**-Used to change name of table.

Data Query Language:

It is also known as data retrival lang.

Select: Used to retrive data from database

Ex select * from table_name;

Data Manupulation Language:

It has 5 commands.

1) **Insert:** Insert data to database
   Ex.insert into table_name(col 1 datatype,col2 datatype,….col n datatype) values(value1,value2,….valuen);
2) **Delete-**Used to delete records from table or database

   Ex. Delete from table_name [where condition];

3) Update-Used to update existing data within table
   Ex.
4) **Lock-**It is Table control concurrency
5) **Call**-call PL/SQL

Data Control Language:

It has 2 commands

1) **Grant**-Access privilege the database
2) **Revoke-**Withdraw the access privileges

Transaction Control Language:

It has3 commands

1) **Commit**-commit the transaction
2) **Savepoint**-set the savepoint within transaction
3) **Rollback-**To get Savepoint

Use Keyword

Ex.use base1;

Insert

Syntax:insert into table_name(column1 datatype,column 2 datatype,….)values (value1 ,value2,…);

insert into employee (dept, position,Email_Id,phone_number,Birthday,Location) values('Testing','software Tester','sachinsh912@gmail.com',9623582648,04/3/1998,'Kolhapur');

select *from employee;

```
-------------+-----------------+-------------------------+--------------+---------------------+----------+

| dept        | position        | Email_Id                | phone_number | Birthday            | Location |

+-------------+-----------------+-------------------------+--------------+---------------------+----------
+

| Design      | Desig manager   | pqrse123@gmail.com      | 7447677768   | 05/9/87             |
Satara   |

| Development | Sr.Developer    | abc89@gmail.com         | 9875184565   | 08/06/1987          |
Mumbai   |

| Sales       | Sales Manager   | ushabansode123@gmail.com | 9623303863   | 02/06/1998
| pune     |
| Excutive    | CEO            | snehabahir123@gmail.com  | 9309792716   | 03/08/1995          |
Sangli   |

| Testing     | software Tester | sachinsh912@gmail.com    | 9623582648   |
0.000667334000500500 | Kolhapur |

+-------------+-----------------+-------------------------+--------------+---------------------+----------
+
```

Update

Syntax: update table_name set column 1=value 1,column 2=value 2,…. Where condition;

update employee set Birthday='09/9/98' where dept='Testing';

```
+-------------+-----------------+-------------------------+--------------+------------+----------+

| dept        | position        | Email_Id                | phone_number | Birthday   | Location |

+-------------+-----------------+-------------------------+--------------+------------+----------+

| Design      | Desig manager   | pqrse123@gmail.com      | 7447677768   | 05/9/87    | Satara
|

| Development | Sr.Developer    | abc89@gmail.com         | 9875184565   | 08/06/1987 |
Mumbai   |

| Sales       | Sales Manager   | ushabansode123@gmail.com | 9623303863   | 02/06/1998 |
pune     |

| Excutive    | CEO             | snehabahir123@gmail.com  | 9309792716   | 03/08/1995 | Sangli
|

| Testing     | software Tester | sachinsh912@gmail.com    | 9623582648   | 09/9/98    |
Kolhapur |

+-------------+-----------------+-------------------------+--------------+------------+----------+
```

Alter

Syntax: ALTER TABLE *table_name*
ADD *column_name datatype*;

ALTER TABLE *table_name*
DROP COLUMN *column_name;*

alter table employee add Address varchar (30);

```
+-------------+-----------------+-------------------------+--------------+------------+----------+---------
+

| dept        | position        | Email_Id                | phone_number | Birthday   | Location |
Address |
+-------------+-----------------+-------------------------+--------------+------------+----------+---------
+

| Design      | Desig manager   | pqrse123@gmail.com      | 7447677768   | 05/9/87    | Satara
| NULL    |

| Development | Sr.Developer    | abc89@gmail.com         | 9875184565   | 08/06/1987 |
Mumbai   | NULL    |

| Sales       | Sales Manager   | ushabansode123@gmail.com | 9623303863   | 02/06/1998 |
pune     | NULL    |

| Excutive    | CEO             | snehabahir123@gmail.com  | 9309792716   | 03/08/1995 | Sangli
| NULL    |

| Testing     | software Tester | sachinsh912@gmail.com    | 9623582648   | 09/9/98    |
Kolhapur | NULL    |
```

Drop

Syntax: DROP TABLE *table_name*;

drop table employee;

Truncate

Syntax: TRUNCATE TABLE *table_name*;

Truncate table employee;

Delete

Syntax: DELETE FROM *table_name* WHERE *condition*;

delete from employee where email_id='abc89@gmail.com';

```
+----------+-----------------+-------------------------+--------------+------------+----------+---------+

| dept     | position        | Email_Id                | phone_number | Birthday   | Location | Address |

+----------+-----------------+-------------------------+--------------+------------+----------+---------+

| Design   | Desig manager   | pqrse123@gmail.com      | 7447677768   | 05/9/87    | Satara   |
NULL    |

| Sales    | Sales Manager   | ushabansode123@gmail.com | 9623303863   | 02/06/1998 | pune
| NULL    |

| Excutive | CEO             | snehabahir123@gmail.com  | 9309792716   | 03/08/1995 | Sangli   |
NULL    |

| Testing  | software Tester | sachinsh912@gmail.com    | 9623582648   | 09/9/98    | Kolhapur
| NULL    |

+----------+-----------------+-------------------------+--------------+------------+----------+---------+
```

Rename:

Syntax: rename old_table_name to new_table_name;

alter old_table_name rename to new_table_name;

Ex: rename table employee to Emp;

Select command

Count()

Syntax: SELECT COUNT(*column_name*)
FROM *table_name*
WHERE *condition*;

select count(emp_id) from emp where emp_id=1;

```
+---------------+
| count(emp_id) |
+---------------+
|             1 |
```

Avg()

Syntax: SELECT AVG(*column_name*)
FROM *table_name*
WHERE *condition*;

select avg(salary) from emp;

```
+-------------+
| avg(salary) |
+-------------+
|       31500 |
+-------------+
```

Sum()

Syntax: SELECT SUM(*column_name*)
FROM *table_name*
WHERE *condition*;

select sum(salary) from emp;

```
+-------------+

| sum(salary) |

+-------------+

|      126000 |

+-------------+
```

Min()

Syntax: SELECT MIN(*column_name*)
FROM *table_name*
WHERE *condition*;

```
select min(salary) from emp ;

+-------------+

| min(salary) |

+-------------+

| 20000       |

+-------------+
```

Max()

Syntax: SELECT MAX(*column_name*)
FROM *table_name*
WHERE *condition*;

```
select max(salary) from emp;

+-------------+

| max(salary) |

+-------------+

| 44000       |

+-------------+
```

Distinct

Syntax: SELECT DISTINCT *column1*, *column2, ...*
FROM *table_name*;

ADD

Syntax: SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition1* AND *condition2* AND *condition3 ...*;

select *from emp where dept='design' and location='satara';

```
+--------+---------------+--------------------+--------------+----------+----------+---------+--------+---
-----+
| dept   | position      | Email_Id           | phone_number | Birthday | Location | Address |
Emp_id | Salary |
+--------+---------------+--------------------+--------------+----------+----------+---------+--------+---
-----+
| Design | Desig manager | pqrse123@gmail.com | 7447677768   | 05/9/87  | Satara   | abc     |
1      | 22000  |
```

OR

Syntax: SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition1* OR *condition2* OR *condition3 ...*;

select *from emp where dept='design' or location='satara';

```
+--------+---------------+--------------------+--------------+------------+----------+---------+--------+-
-------+
| dept   | position      | Email_Id           | phone_number | Birthday   | Location | Address |
Emp_id | Salary |
+--------+---------------+--------------------+--------------+------------+----------+---------+--------+-
-------+
| Design | Desig manager | pqrse123@gmail.com | 7447677768   | 05/9/87    | Satara   | abc     |
1      | 22000  |
| Sales  | Sale manager  | ytfhhk@gamail.com  | 7845985745   | 09/10/1789 | satara   | sanj    | 6
| 80000  |
```

NOT

Syntax: SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE NOT *condition*;

select *from emp where not (dept='testing');

```
+------------+--------------------+-------------------------+--------------+------------+----------+-------
--+--------+--------+
| dept       | position           | Email_Id                | phone_number | Birthday   | Location |
Address | Emp_id | Salary |
+------------+-------------------+------------------------+-------------+------------+----------+-------
--+--------+--------+

| Design     | Desig manager      | pqrse123@gmail.com       | 7447677768   | 05/9/87    | Satara
| abc     | 1      | 22000  |

| Sales      | Sales Manager      | ushabansode123@gmail.com | 9623303863   | 02/06/1998 |
pune     | abc     | 2      | 20000  |

| Excutive   | CEO                | snehabahir123@gmail.com  | 9309792716   | 03/08/1995 |
Sangli   | xyz     | 3      | 40000  |

| Sales      | Sale manager       | ytfhhk@gamail.com        | 7845985745   | 09/10/1789 | satara   |
sanj    | 6      | 80000  |

| Production | production manager | samiksha@gamail.com      | 7845874595   | 09/10/1989 |
sangali  | sangj   | 7      | 70000  |
```

Limit

Syntax: select column_name(s)

From table_name

Where condition

Limit number;

select dept from emp where address='abc' limit 2;

```
+--------+

| dept   |

+--------+

| Design |

| Sales  |

+--------+
```

In Function

Syntax:select column_name(s)

From table_name

Where column_name in(value1, value2,…….)

Ex; select location from emp where salary in (20000,40000);

```
+----------+

| location |

+----------+
| pune     |

| Sangli   |
```

Between Operator

Syntax:
```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

select position from emp where salary between 40000 and 90000;

```
+-----------------+

| position        |

+-----------------+

| CEO             |

| software Tester |

| QA              |
```

JOIN

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

INNER JOIN

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

Syntax:

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

LEFT JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

Syntax:
```
SELECT column_name(s)
FROM table1
```

LEFT JOIN *table2*
ON *table1.column_name* = *table2.column_name*;

RIGHT JOIN

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

Syntax: SELECT *column_name(s)*
FROM *table1*
RIGHT JOIN *table2*
ON *table1.column_name* = *table2.column_name*;

FULL OUTER JOIN

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Syntax: SELECT *column_name(s)*
FROM *table1*
FULL OUTER JOIN *table2*
ON *table1.column_name* = *table2.column_name*
WHERE *condition*;

Self Join

A self join is a regular join, but the table is joined with itself.

Syntax: SELECT *column_name(s)*
FROM *table1 T1*, *table1 T2*
WHERE *condition*;

UNION Operator

The UNION operator is used to combine the result-set of two or more SELECT statements.

- Every SELECT statement within UNION must have the same number of columns
- The columns must also have similar data types
- The columns in every SELECT statement must also be in the same order

Syntax: SELECT *column_name(s)* FROM *table1*
UNION
SELECT *column_name(s)* FROM *table2*;

GROUP BY Statement

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

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.

Syntax: SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
ORDER BY *column_name(s);*

select dept from emp group by dept;

```
+----------+

| dept     |

+----------+

| Design   |

| Sales    |

| Excutive |

| Testing  |

+----------+
```

ORDER BY

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

Syntax: SELECT *column1*, *column2, ...*
FROM *table_name*
ORDER BY *column1, column2, ...* ASC|DESC;

Select * from emp order by location;

```
+----------+----------------+------------------------+-------------+-----------+----------+---------+-
-------+--------+
| dept     | position        | Email_Id                | phone_number | Birthday   | Location | Address |
Emp_id | Salary |

+----------+-----------------+-------------------------+--------------+------------+----------+---------+-
-------+--------+

| Testing  | software Tester | sachinsh912@gmail.com    | 9623582648   | 09/9/98    | Kolhapur
| pqr     | 4      | 44000  |

| Testing  | QA              | ushaban2gmail.com        | 8957485478   | 9/09/1998  | miraj    | abc
| 5      | 90000  |

| Sales    | Sales Manager   | ushabansode123@gmail.com | 9623303863   | 02/06/1998 | pune
| abc     | 2      | 20000  |

| Excutive | CEO             | snehabahir123@gmail.com  | 9309792716   | 03/08/1995 | Sangli   |
xyz     | 3      | 40000  |

| Design   | Desig manager   | pqrse123@gmail.com       | 7447677768   | 05/9/87    | Satara   |
abc     | 1      | 22000  |

+----------+-----------------+-------------------------+--------------+------------+----------+---------+-
-------+--------+
```

WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

Syntax: SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition*;

select * from emp where dept='testing';

```
+---------+-----------------+----------------------+--------------+-----------+----------+---------+------
--+--------+

| dept    | position        | Email_Id             | phone_number | Birthday  | Location | Address |
Emp_id | Salary |

+---------+-----------------+----------------------+--------------+-----------+----------+---------+------
--+--------+

| Testing | software Tester | sachinsh912@gmail.com | 9623582648   | 09/9/98   | Kolhapur |
pqr     | 4      | 44000  |

| Testing | QA              | ushaban2gmail.com     | 8957485478   | 9/09/1998 | miraj    | abc     | 5
| 90000  |
+---------+-----------------+----------------------+--------------+-----------+----------+---------+------
--+--------+
```

HAVING Clause

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

Syntax: SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
GROUP BY *column_name(s)*
HAVING *condition*
ORDER BY *column_name(s);*

select count(address),dept from emp group by dept having count(address)<5;

```
+----------------+----------+

| count(address) | dept     |

+----------------+----------+

|              1 | Design   |

|              1 | Sales    |

|              1 | Excutive |

|              2 | Testing  |

+----------------+----------+
```

Transaction Control Language Commands

Commits:

Ex.commits;

Savepoint

Syntax: savepoint savepoint_name;

Ex.savepoint updation;

Rollback

Syntax:rollback to savepoint;

Ex rollback to updation;

Copy

Syntax:

- select column(s)_name from table_name where condition[];
- insert into new_table_name select * from old_table_name where condition[];

  Ex:

LIKE Operator

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (_) represents one, single character

Syntax: SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *columnN* LIKE *pattern*;

select *from emp where dept like 's%';

```
+-------+---------------+-------------------------+--------------+------------+----------+---------+------
--+--------+

| dept  | position      | Email_Id                | phone_number | Birthday   | Location | Address |
Emp_id | Salary |

+-------+---------------+-------------------------+--------------+------------+----------+---------+------
--+--------+

| Sales | Sales Manager | ushabansode123@gmail.com | 9623303863   | 02/06/1998 | pune     |
abc     | 2      | 20000  |

| Sales | Sale manager  | ytfhhk@gamail.com        | 7845985745   | 09/10/1789 | satara   | sanj
| 6      | 80000  |

+-------+---------------+-------------------------+--------------+------------+----------+---------+------
--+--------+
```

SELECT TOP Clause

The SELECT TOP clause is used to specify the number of records to return.

Syntax: SELECT TOP *number*|*percent* *column_name(s)*
FROM *table_name*
WHERE *condition*;

MySQL Server

SELECT *column_name(s)*
FROM *table_name*
WHERE *condition*
LIMIT *number*;

Rand

Syntax: select column(s)_name from table_name order by rand();

Ex; **select Emp_id from emp order by rand()limit 2;**

```
+--------+

| Emp_id |

+--------+

| 1      |

| 2      |

+--------+
```

Cast

Syntax: select cast(123 as varchar(20)) [result_name] from [source]
Ex; select s_no , cast(maths as float) float_maths from marks;

Least

Syntax: select least(1,2,3,……..n);

Ex; select least(1,2,3,4,……) as least_of_number;

Subquery

Syntax: select column(s)_name from table_name1 where condition[]comparison_operator(select column(s)_name from table_name from table_name where condition[]);

Ex: **select *from emp where salary >(select avg(salary) from emp);**

```
+------------+-------------------+--------------------+-------------+-----------+----------+---------+-
-------+--------+
| dept       | position           | Email_Id           | phone_number | Birthday   | Location | Address |
Emp_id | Salary |

+------------+--------------------+--------------------+--------------+------------+----------+---------+-
-------+--------+

| Testing    | QA                 | ushaban2gmail.com   | 8957485478   | 9/09/1998  | miraj    | abc
| 5      | 90000  |

| Sales      | Sale manager       | ytfhhk@gamail.com   | 7845985745   | 09/10/1789 | satara   |
sanj    | 6      | 80000  |

| Production | production manager | samiksha@gamail.com | 7845874595   | 09/10/1989 |
sangali  | sangj   | 7      | 70000  |
```

Substring

Syntax: select substring(column_name,1,n) from table_name;

Ex: **select substring(dept,1,4)from emp;**

```
+---------------------+

| substring(dept,1,4) |

+---------------------+

| Desi                |

| Sale                |

| Excu                |

| Test                |

| Test                |

| Sale                |

| Prod                |
```

Upper And Lower

Syntax: select upper/lower(column_name)from table_name

Ex; **select upper(dept)from emp**;

```
+-------------+

| upper(dept) |

+-------------+
| DESIGN      |
| SALES       |
| EXCUTIVE    |
| TESTING     |
| TESTING     |
| SALES       |
| PRODUCTION  |
```

select lower (dept) from emp;

```
+--------------+
| lower (dept) |
+--------------+
| design       |
| sales        |
| excutive     |
| testing      |
| testing      |
| sales        |
| production   |
+--------------+
```

Second-Highest Value of an Integer

Syntax: select max(column_name) from table_name where column_name not in(select max(column_name) from table_name);

Ex; **select max(salary) from emp where emp_id not in (select max(salary)from emp);**

```
+-------------+
| max(salary) |
+-------------+
| 90000       |
+-------------+
```