Using Select query

1. TO display the record of a column
   1. Syntax : select columnname from tablename;

Eg: select first\_name from employees;

1. To display the record of more one column

Syntax : select col1,col1….. from tablename ;

Eg: select first\_name,last\_name from employees;

1. To display records of all columns

Syntax : select \* from table name;

Eg: select \* from employees;

1. To give an alias name to a column

Syntax: select column name as alias\_name from tablename;

Eg: select first\_name, hire\_date as start\_date from employees;

1. To display unique values from a column

Syntax: select distinct job\_id from employees;

1. To concatenate values if two columns.

Syntax: select column1 || column2 from tablename; --> No column names as not given and without spaces

Eg: select f\_name || l\_name from employes;

--> For giving name and spaces

Syntax: select f\_name || ‘ ’ || l\_name as Name from employees; or --> name = “Employee Name”

**WHERE CLAUSE**

**-->** WHERE is used to filter the records while fetching them from a table.

1. WAQ to display first\_name ,last\_name, salary from employees where salary is more than 5000.

Syntax: select first\_name,last\_name , salary from employees where salary >5000;

1. For a range (5000-10000)

Syntax:

select first\_name ,last\_name ,salary from employees where salary>=5000 and salary<=10000;

Or,

select first\_name ,last\_name ,salary from employees where salary between 5000 and 10000;

1. Whose salary is not in the range 5000 to 10000.

Syntax:

select first\_name ,last\_name ,salary from employees where salary not between 5000 and 10000;

1. To display first \_name , job\_id where first\_name starts with ‘A’

Syntax:

Select first\_name , job\_id from employees where first\_name like ‘A%’; (case sensitive)

1. To display last\_name , job\_id where last name ends with ‘N’;

Syntax:

Select last\_name , job\_id from employees where last\_name like ‘%N’; (case sensitive)

1. To display first\_name, job\_id where first name contains ‘t’

Syntax:

Select first\_name , job\_id from employees where first\_name like ‘%t%’;

1. To display last name , job id where last name has 4th character h.

Syntax:

Select last\_name , job\_id from employees where last\_name like ‘\_\_\_h%’; (case sensitive)

**Using ORDER BY**

--> For sorting in ascending or descending order .

1. Select first\_name , job\_id from employees order by first\_name asc;

Or,

1. Select first\_name , job\_id from employees order by first\_name desc ;
2. To display salary > 10000 and in asc order .

Select first\_name ,last\_name,salary from employees where salary> 10000 order by first\_name asc;

Q>WAQ to display first\_name ,salary from employees where salary is more than prompt salary.

Syntax:

Select first\_name ,salary from employees where salary>= &salary;

**USING AGGREGATE FUNCTIONS**

1. **sum()**: To find total salary in employees table

Syntax: select sum(salary) from employees;

OR,

Select sum(salary) as “Total Salary” from employees;

1. **Count()**:--> To count total number of employees

Syntax: select count(\*) from employees;

1. **Avg()**: To find average

Syntax: select avg(salary) from employees;

1. **Min()**:

Syntax: select min(salary) from employees;

1. **Max()**

Syntax : select max(salary) from employees;

\*\*\* aggregate functions always with select query

Complex\*\*

1. To display first\_name ,salary of highest paid employees

Syntax: select first\_name,salary from employees where salary= (select max(salary) from employees);

1. To display second highest paid employee

Syntax: select first\_name , salary from employees where salary = (select max(salary) from employees where salary != ( select max(salary) from employees));

**USING GROUP BY**

--> Used to group rows having same value.

--> Provides summary of data in a table.

--> Useful for data analysis.

Sample query : select first\_name , job\_id , salary from employees;

1. To find total salary for each job\_id.

--> select job\_id, sum(salary) from employees group by job\_id;

Q)To find the total members for job\_id.

--> select job\_id , count(\*) from employees group by job\_id ;

1. To find average,minimum and maximum salary of each department .

Syntax: select job\_id ,max(salary),min(salary), avg(salary) from employees group by job\_id;

1. To find average,maximum , minimum salary of each job\_id where max salary is between 10000 and 15000.

\*\*\* no WHERE clause after using GROUP BY instead use HAVING \*\*\*

**USING HAVING CLAUSE**

**-->** Operations on grouped records returned by GROUP BY .

--> Only the groups that meet HAVING criteria are returned .

--> HAVING requires that GROUP BY is present.

Syntax:

Select column\_name from table\_name where condition group by column\_name having condition order by column\_name;

\*\*\*NOTE---> HAVING can only be used with aggregate functions.

1. To find average,maximum , minimum salary of each job\_id where max salary is between 10000 and 15000.

Syntax: select job\_id , max(salary),min(salary),avg(salary) from employees group by job\_id having max(salary) between 10000 and 15000;

1. To display department\_id , no. Of employees from each department where no>5.

Syntax: select department\_id ,count(department\_id) from employees group by department\_id having count(\*)>5;