**DBMS LAB**

**Lab Assignment number 05**

**Name:** Aamir Ansari **Batch:** A **Roll no.** 01

**Aim:** Experiment to study Simple and nested queries.

**Theory:**

Query is nothing but question asked to the database.

Select statement:

Select command is used to retrieve data from database. The result is stored in the temporary table called result set.

General Syntax:

Select A1, A2, A3-----An from R1, R2------Rn where Predicate

Eg. Consider table Employee (eid, ename, salary, dno)

Select ename , salary from employee

Selecting all columns:-

Select \*from ‘tablename&gt’

Select \*from employee

Choosing selected columns:-

Select A1, A2, A3-----An from tablename

Select eid,salary from employee

Computed values in select List:-

Select eid,salary\*1.2 as increase\_sal from employee

Filtering Rows from a table:-

Where clause:-

Where clause is used to extract only those records that fulfill specified condition.

Where clause is used to filter records.

Select ‘columnname&gt’ from ‘tablename&gt’ where ‘condition&gt’

Select empid from employee where salary>20000

Distinct clause:- Eliminating duplicates

Distinct keyword is used to eliminate duplicate rows from the resultset.

Select distinct dno from employee

By default – all

Select all dno from employee

Top clause:- Top clause limits the number of rows returned in the resultset.

Syntax:- Select top n ‘columname&gt’ from ‘tablename&gt’

-- Display first three records from resultset(employee)

Select top 3 \* from emp

-- Display salaries of first two records from resultset

Select top 2 salary from emp

Operators in SQL:

Relational operators:- ,> ,’ ,’=, >=,’>(!=)

Logical Operators:- and , or ,not

Range searching Operators(between ,not between)

Between operator :- is used to specify range of values.

Syntax:- select columname from tablename where columnname between lowerlimit and

upperlimit

Eg. Display emp details from employees where salary is within range 10000 and 20000.

Select \*from employee where salary between 10000 and 20000

Select \*from employee where salary =10000 and salary =20000

Display emp details from employee where salary is not Within range between 10000 and 20000

Select \*from employee where salary not between 10000 nad 20000

Pattern matching operator:- Like ,Not Like

Like operator:

Like operator is used in a where clause to search for a specified pattern in a column.

Pattern is specified as a string in single quotes.

General Syntax:-

Select colnames from tablename where colname like ‘pattern’

SQL wild characters:-

SQL wild characters can substitute for one or more characters

When searching for data in database.

SQL wild characters must be used with Like and not like operator.

Wild character Description

% A substitute for 0 or more characters.

\_ A substitute for exactly one characters

[range] A single character within range [A-D]

[^range] A single character not within range

Eg. Select \*from emp where ename like ‘A%’

Select \*from emp where ename like ‘%A%’

Select \*from emp where eid like ‘\_\_’

Select \*from emp where eid like ‘\_\_%’

Select \*from emp where ename like ‘\_[e-f]%’

Set Membership operators:- In and Not In

In operator:-

In operator is used to check membership of value in given set

Not in :-

Not In operator is used to check absence of membership of value in given set

Syntax:

Select colname from tablename where colname in(value1,value2…..)

Eg. Select \*from emp where dno in(20,10,25)

Select \*from emp where dno not in(20,10,25)

Nested Queries:-

Nested query is a query that has another query embedded within it.The embedded query is called a subquery.

Some queries requires that data to be fetched from database and then used in comparison

Condition.

When we use Select----from----where block within where clause of another query such type of query is called Nested queries.

General syntax:

Outer query(parent query)

Select ‘columnname> from ‘tablename> where col= (select ------ from ‘tablename> where ‘condition>)

Subquery typically appears within a search condition as a part of the where or having clause of a select, update, delete.

Set comparision operators(all, some|any):-

All operator :- All operator is used to compare value of a column with all values from given set.

Some / Any :- Some / Any operator is used to compare value of a column with some of the values from given set.

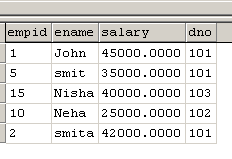
--Set comparision operators are always used with relational operators.

Eg. >all,>=all, <all, <=all, <>all, <some, =some, <>some, <=some, >some---------------etc.

=some equivalent to in, <>some equivalent to not in.

Eg. Retrive employee details of employees those are working in

It department.

select \*from emp where dno =(select dno from dept where dname ='it') (102)



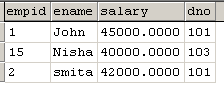
Eg. Retrive employee details of employees those are working It

department or comp department.

select \*from emp where dno in(select dno from dept where dname in ('it', 'comp') (101,102)

List employee details of employees earning salary more than salary of emp with empid 5.

select \*from emp where salary >(select salary from emp where empid=5) 35000



List employee details of employees earning salary more than salary of every emp working for dept no 101.

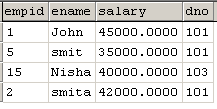
select \*from emp where salary >=all(select salary from emp where dno=101) (45000,35000,42000)



select \*from emp where salary >=(select salary from emp where dno=101) --error

List employee details of employees earning salary more than salary of some of emps working for dept no 101.

select \*from emp where salary >=some(select salary from emp where dno=101)



--List employee details of employees earning salary more than salary of some of emps working for IT department.

select \*from emp where salary >=some(select salary from emp where dno=(select dno from dept where dname='it'))

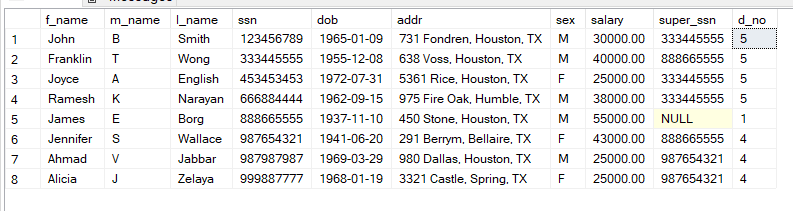
--List employee details of employees earning salary more than salary of some of emps working for IT department or comp dept.

select \*from emp where salary >=some(select salary from emp where dno in (select dno from dept where dname in ('it',’comp’)))

**Code:**

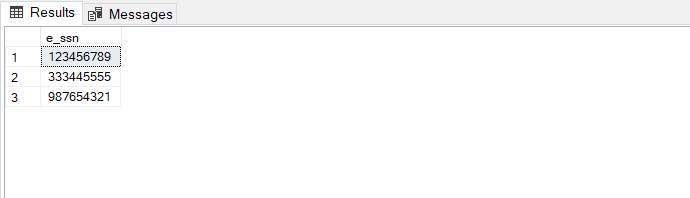
--1 Salary greater than 20,000

SELECT \* FROM Employee WHERE salary > 20000;



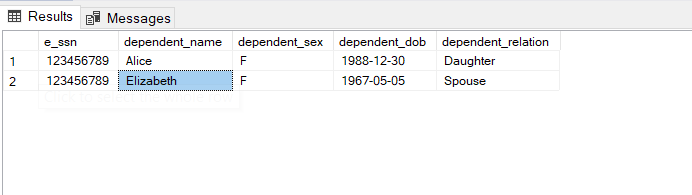
--2 Distinct ssn for employee who have dependant

SELECT DISTINCT e\_ssn FROM Dependant;



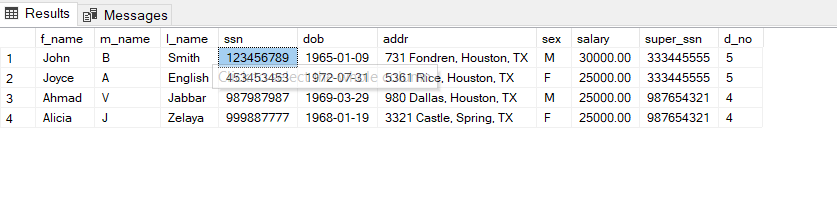
--3 First two dependants

SELECT TOP 2 \* FROM Dependant;



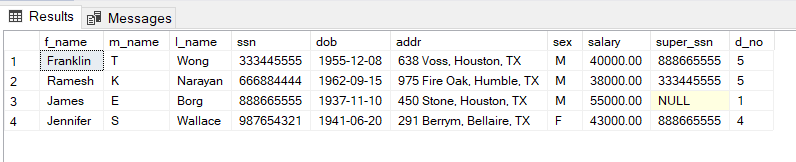
--4 Employee with salary between 10,000 and 30,000

SELECT \* FROM Employee WHERE salary BETWEEN 10000 AND 30000;



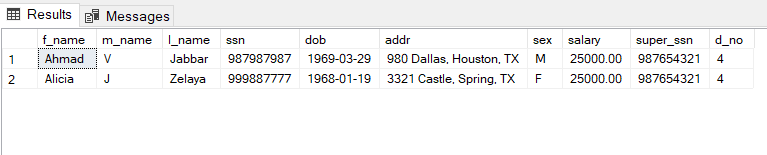
--5 Employee with salary between 10,000 and 30,000

SELECT \* FROM Employee WHERE salary NOT BETWEEN 10000 AND 30000;



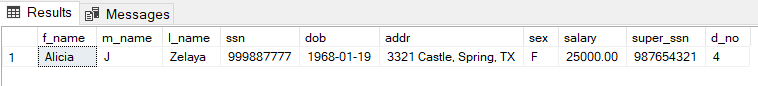
--6A Employee with name starting with A

SELECT \* FROM Employee WHERE f\_name LIKE 'A%';



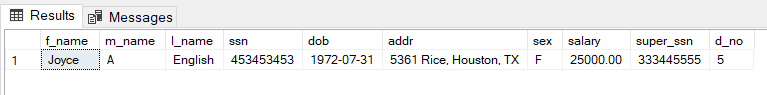
--6B Employee with name ending with A

SELECT \* FROM Employee WHERE f\_name LIKE '%a';



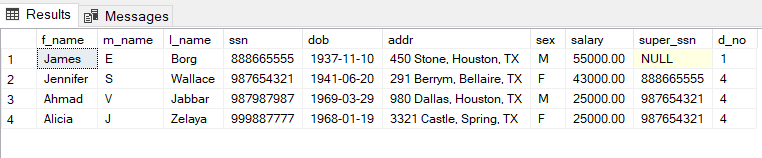
--6C Employee with name ending with A

SELECT \* FROM Employee WHERE f\_name like '%[f,e]';



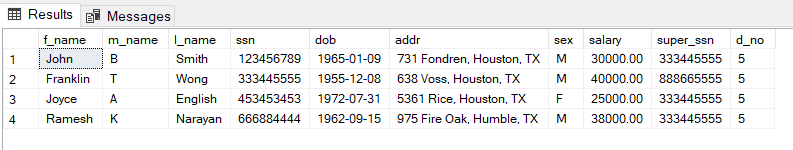
--7 Employee in department 1 and 4

SELECT \* FROM Employee WHERE d\_no = 1 OR d\_no = 4;



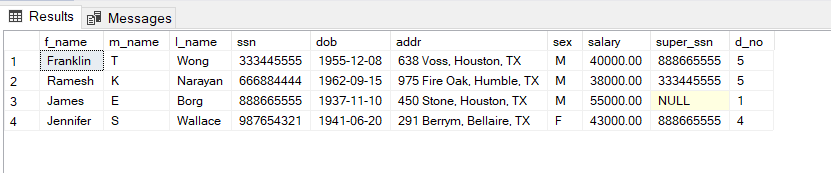
--8 Employee working in research

SELECT \* FROM Employee WHERE d\_no = (SELECT d\_no FROM Department WHERE d\_name = 'Research');



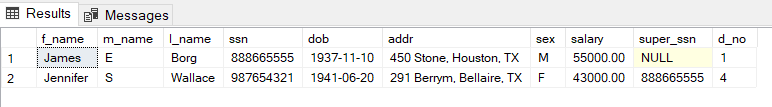
--9 Employee with more salary than a particular employee

SELECT \* FROM Employee WHERE salary > (SELECT salary FROM Employee WHERE ssn = 123456789);



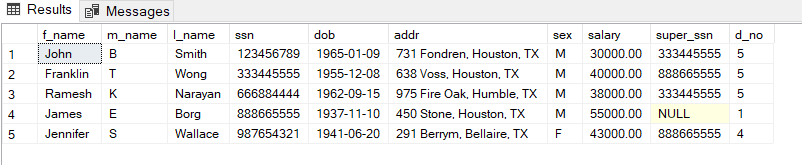
--10 Employee with more salary than max of department 5

SELECT \* FROM Employee WHERE salary > (SELECT MAX(salary) FROM Employee WHERE d\_no = 5);



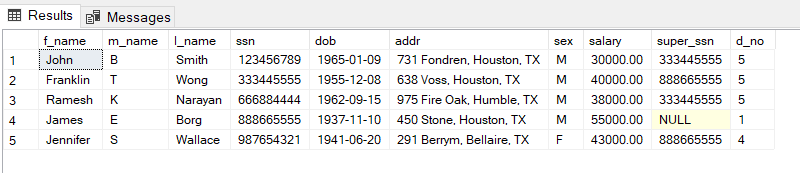
--11 Employee with more salary than min of Administration department

SELECT \* FROM Employee WHERE salary > (SELECT MIN(salary) FROM Employee WHERE d\_no = (SELECT d\_no FROM Department WHERE d\_name = 'Administration'));



--12 Employee with more salary than min of Administration or Research

SELECT \* FROM Employee WHERE salary > (SELECT MIN(salary) FROM Employee WHERE d\_no IN (SELECT d\_no FROM Department WHERE d\_name IN ('Administration', 'Research')));



**Conclusion:** We have successfully studied and implemented Simple and nested queries.