Contents

Data Query Language [DQL]

It is used to retrieve the data from the database from the particular object (nothing but Table)

List of DQL statements:

1) select
2) projection
3) selection
4) Joins

SELECT

* Select clause is used to retrieve the data from the table by selecting only the columns.
* From the select clause we can pass the arguments like * (Alldetails), column name (particular col), expression (Result of that expression)
* Select clause is used to display the result based on the given argument.

PROJECTION

The process of retrieving the data from the table by selecting only columns

```
SELECT */[DISTINCT] COL_NAME/EXPRESSION[ALIAS]

FROM TABLE_NAME;
```

Order of execution:

→ From clause executes first.
→ From the FROM clause we can pass an argument (Table name)
→ The job of FROM clause is to go to database, search for the table and put the table under execution
→ After execution from clause, select clause will execute.
→ From the Select clause we can pass arguments like column name, *, expression
→ The job of select clause is go to the table which is under execution, select the result and display the output.

FROM CLAUSE — Tablename: EMP → Database

SELECT CLAUSE — COL-name: ENAME → output

EMPNO	ENAME	SAL	JOB
1	Miller	5000	clerk
2	Adams	3000	clerk
3	King	2000	President

Result

* Miller
Adams
King

① WAQtd the names of employees from the employee Table

```
SELECT ENAME
FROM EMP;
```

② WAQtd the salaries of the employees

```
SELECT SAL
FROM EMP;
```

③ waqtd the hiredate of the employees

```
SELECT HIREDATE
FROM EMP;
```

④ waqtd the commission of the employees

```
SELECT COMM
FROM EMP;
```

⑤ waqtd the empnumber and names from the employee

```
SELECT EMPNO, ENAME
FROM EMP;
```

⑥ waqtd the names of employee, salary, dept no for all the employees

```
SELECT ENAME, SAL, DEPTNO
FROM EMP;
```

⑦ waqtd employeenumber, name, Job, commission for all the employees

```
SELECT EMPNO, ENAME, JOB, COMM
FROM EMP;
```

⑧ waqtd department names present in department Table

```
SELECT DEPTNAME DNAME
FROM DEPT;
```

⑨ waqtd all the details of the employee

```
SELECT *
FROM EMP;
```

⑩ waqtd all the details of the department Table

```
SELECT *
FROM DEPT;
```

EMPLOYEE TABLE:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

DEPARTMENT TABLE

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEWYORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Distinct Clause

* Distinct clause is used to remove repeated and duplicated values from the Result table.
* Distinct clause is used as first argument in the select clause along with the distinct clause we can use multiple column names
* Distinct clause is optional.

① Waqtd Distinct Salary

```
SELECT DISTINCT SAL
FROM EMP;
```

② Waqtd Distinct Employee name and Salary

```
SELECT DISTINCT ENAME,SAL
FROM EMP;
```

③ Waqtd Employee number, Employee name and Salary of the Employes

```
SELECT EMPNO, EMPNAME,SAL
FROM EMP;
```

④ Waqtd Distinct department number

```
SELECT DISTINCT DEPTNO
FROM DEPT;
```

Expression

Expression statement gives result.

Statement

Statement is a combination of operators and operand

① Waqtd Annual Salary of the Employee

```
SELECT SAL*12
FROM EMP;
```

② Wapto Halfterm Salary of the Employee

```
SELECT SAL*6
FROM EMP;
```

③ Wapto Name and Salary of the Employee with penalty of 100

```
SELECT ENAME, SAL-100
FROM EMP;
```

④ Wapto all the Details of the Employee along with their Annual Salary

```
SELECT EMP.*, (SAL*12) AS ANNUAL_SAL
FROM EMP;
```

Note: Along with * we are not supposed to use any other column name or Expression Eventhough if we want to use column name or expression along with * make sure that tablename has been given before * (star) as a reference

Alias

An alternative name given to an Object

Alias name is not permanent, we can provide Alias name in three ways.

→ By using AS keyword
→ Without using AS keyword
→ By using doubleQuotes (" ")

① Wapto Salary as Annual-Salary by using three methods

```
SELECT SAL*12
AS ANNUAL_SAL
FROM EMP;
        (or)
SELECT SAL*12 ANNUAL_SAL
FROM EMP;
        (or)
SELECT SAL*12 "ANNUAL_SAL"
FROM EMP;
```

Selection

The process of retreiving the data from the table by selecting both rows as well as columns.

Syntax

```
SELECT * / [DISTINCT] COLUMN-NAME / EXPRESSION [ALIAS]
FROM TABLE_NAME
WHERE < FILTER_CONDITION >;
```

Order of Execution

First from FROM
Second WHERE [ROW BY ROW]
Third one SELECT

What is where clause?

Where clause is used to filter the condition from the Table
Where clause executes row by row
In where clause we must write only filter condition.

Note: 1) From clause execute first
2) From FROM clause we can pass arguments (i.e TableName)
3) The job of the FROM clause to go to the database and search for the table and put the table under execution.
4) After the from clause execution, WHERE clause executes
5) from the WHERE clause we can pass arguments like only FILTER condition.
6) The job of WHERE clause is to go to the table which is under execution and start executing row by row to filter the condition.
7) After the execution of where clause, SELECT clause will execute.
8) The job of the SELECT clause is select the result and display as the output.

Wap to name and salary of employee earning salary more than 2000

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL>2000;
```

Wap to name and salary of employee earning salary less than 5000.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL <5000
```

waqtd name and deptno of employee working in deptno 20

```
SELECT ENAME,DEPTNO
FROM EMP
WHERE DEPTNO=20;
```

waqtd details of employee whose commission is 1400

```
SELECT *
FROM EMP
WHERE COMM=1400;
```

waqtd details of employee whose empno is 7566

```
SELECT *
FROM EMP
WHERE EMPNO=7566;
```

waqtd Annual salary of employee whose name is Smith

```
SELECT SAL*12
FROM EMP
WHERE ENAME ='SMITH';
```

waqtd name of employee work as clerk

```
SELECT ENAME
FROM EMP
WHERE JOB='CLERK';
```

waqtd salary of employee working as salesman

```
SELECT SAL
FROM EMP
WHERE JOB ='SALESMAN';
```

waqtd who earns more than 2000

```
SELECT *
FROM EMP
WHERE SAL >2000;
```

waqtd who was hired after 01-Jan-81

```
SELECT *
FROM EMP
WHERE HIREDATE >'01-JAN-81';
```

Order of WHERE Clause

FROM CLAUSE → Tablename: EMP → Database

WHERE CLAUSE → SAL>1000

SELECT CLAUSE → ENAME / RESULT → Output

EMPNO	ENAME	SAL
1	Hari	2000
2	Ram	3000
3	Dev	4000
4	Krish	5000
5	Devi	100

Result

ENAME
Hari
Ram
Dev
Krish

Operators in sql

Arithematic operator[+,-,*,/,%]
Concat operator[||]
Relational operator [>,<,<=,>=]
Comparison operator [= , !=]
Logical operator [AND,OR,NOT]
Special operator [IN, NOT IN, BETWEEN, NOT BETWEEN, IS, IS NOT, LIKE, NOTLIKE]
Subquery operator[ALL, ANY, EXISTS, NOTEXIST

What is concat operator?
It is used to merge two strings.

Waqtd ename and sal for all employees by using concat operator

```
SELECT ENAME ||SAL
FROM EMP;
```

Relational operator

Waqtd name and sal of employee, earning salary less than 2000

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL<2000;
```

Waqtd name & salary of employee who earns salary >3000

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL>3000;
```

Waqtd name and salary whose salary less than or equal to 2975

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL <=2975;
```

Comparison Operator

Waqtd details of employee whose empnumber is 7902

```
SELECT *
FROM EMP
WHERE EMPNO=7902;
```

Waqtd names of employee if the employee is not working in Deptno=20 [Answer should be 2 methods]

```
SELECT ENAME
FROM EMP
WHERE NOT DEPTNO=20;
```

```
SELECT ENAME
FROM EMP
WHERE DEPTNO!=20;
```

```
SELECT ENAME
FROM EMP
WHERE DEPTNO<20 OR DEPTNO>20;
```

Logical Operator

To check multiple conditions we use logical operator

AND operator

- In the given condition if all the conditions are satisfy we get True
- In the given condition if any one not satisfied we get False

wagtd name, Sal and Job of emp if emp is working as manager, earning salary more than 2000

```
SELECT ENAME, SAL, JOB
FROM EMP
WHERE JOB = 'MANAGER' AND SAL > 2000;
```

wagtd name, Job and hiredate if emp working in dept 30 and ~~earning~~ working as Analyst and salary less than 5000

```
SELECT ENAME, JOB, HIREDATE
FROM EMP
WHERE JOB = 'ANALYST' AND SAL < 5000;
```

wagtd details of employee along with Annual Salary working as salesman in dept 10.

```
SELECT EMP.*, (SAL*12)
FROM EMP
WHERE JOB = 'SALESMAN' AND DEPTNO = 10;
```

OR operator

- In the given condition if all the conditions are satisfied, we get result True.
- In the given condition if any one of the condition satisfied, we get result True.
- In the given condition if no condition satisfied we get result False.

wagtd name and Salary of employee if employee working in deptNO 10 or 20

```
SELECT ENAME, SAL
FROM EMP
WHERE DEPTNO = 10 OR DEPTNO = 20;
```

wagtd name and Job of employee if employee work as Salesman / Manager

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB = 'SALESMAN' OR JOB = 'MANAGER';
```

Waqtd all the details of employee working deptNO; 10,20,30, or 40;

```
SELECT *
FROM EMP
WHERE DEPTNO= 10 OR DEPTNO=20 OR DEPTNO=30 OR DEPTNO=40;
```

NOT operator

It is a unary operator or a Reverse Operator

Waqtd name, deptno of employee if they not working in deptno=10;

```
SELECT ENAME, DEPTNO
FROM EMP
WHERE NOT DEPTNO=10;
```

Special operators

1) In
2) Not in
3) Between
4) Notbetween
5) Is
6) Is not
7) Like
8) Notlike

In Operator

It is a multivalued operator which can accept multiple values at RHS.

Syntax

```
COL_NAME IN (V1,V2,V3 ... Vn);
```

Waqtd name and deptno of employes working in deptno 10,20,30,40.

```
SELECT ENAME, DEPTNO
FROM EMP
WHERE DEPTNO IN (10,20,30);
```

Waqtd name and JOB of employees working as Salesman, Analyst, manager, clerk.

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB IN ('SALESMAN','ANALYST','MANAGER','CLERK');
```

Waqtd all the details of employee working as clerk and earning salary less than 1500

```
SELECT *
FROM EMP
WHERE JOB IN ('CLERK') AND SAL < 1500;
```

waqtd details of employee working as manager in deptno 30

```
SELECT *
FROM EMP
WHERE JOB IN ('MANAGER') AND DEPTNO IN (30);
```

waqtd details of employee along with their annual sal working in dept 30 job is Salesman and their Annual Sal has to be greater than 14000

```
SELECT EMP.*, SAL*12
FROM EMP
WHERE JOB IN ('SALESMAN') AND DEPTNO IN(30) AND SAL*12 >14000;
```

Not In Operator

It is similar to In operator but instead of selecting the values, the given values will be rejected and remaining values will be displayed

syntax:

```
COLNAME NOT IN (V1,V2,V3 .. Vn);
```

waqtd name and deptno of employee if employee not working as Manager & Salesman

```
SELECT ENAME, DEPTNO
FROM EMP
WHERE JOB NOT IN ('MANAGER','SALESMAN');
```

waqtd name, Job and deptno of employee if employee working as clerk not in deptno 10 & 30.

```
SELECT ENAME,JOB, DEPTNO
FROM EMP
WHERE JOB NOT IN ('CLERK') AND DEPTNO NOT IN (10,30);
```

BETWEEN OPERATOR

It is used when there is a range of values (starting value, Ending value)

syntax:

```
COLNAME BETWEEN LOWERRANGE AND HIGHERRANGE
```

waqtd name and salary of employee earning salary in the range of 800 & 3000.
(As it is values)

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL BETWEEN 800 AND 3000;
```

waqtd name and sal of Employee in the range between 800 & 3000.
(increment start, decrement end value)

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL BETWEEN 801, AND 2999;
```

Wagtd name and sal of employee hiredate in the range of 1982 and 1985

```
SELECT ENAME,SAL
FROM EMP
WHERE HIREDATE BETWEEN ('01-JAN-1982' AND '31-DEC-1985');
```

Wagtd name and hiredate of employee hired in the range between 1982 & 1987

```
SELECT ENAME,HIREDATE
FROM EMP
WHERE HIREDATE BETWEEN ('01-JAN-1983' AND '31-DEC-1987');
```

Wagtd name of employee hired during the year 2024.

```
SELECT ENAME
FROM EMP
WHERE HIREDATE BETWEEN ('01-JAN-2024' AND '31-DEC-2024');
```

Not Between Operator

It is similar to between operator, but instead of selecting the range get rejected, remaining values will be displayed.

Syntax:

```
COL-NAME NOT BETWEEN HIGHERRANGE AND LOWER RANGE
```

Wagtd name, sal of employee where sal not in range of 800 & 3000

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL NOT BETWEEN (800 AND 3000);
```

Wagtd name, sal of employee where sal not in range between 800 & 3000

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL NOT BETWEEN (801 AND 3000);
```

Wagtd name and sal of employee where hired in range of 1982 and 1985

```
SELECT ENAME,SAL
FROM EMP
WHERE HIREDATE NOT BETWEEN ('01-JAN-1982' AND '31-DEC-1985');
```

Wagtd name and hiredate of employee hired in between range of 1982 & 1987

```
SELECT ENAME,HIREDATE
FROM EMP
WHERE HIREDATE NOT BETWEEN ('01-JAN-1983' AND '31-DEC-1987');
```

Wagtd name of employee hired during the not in the year 2024

```
SELECT ENAME
FROM EMP
WHERE HIREDATE NOT BETWEEN ('01-JAN-2024' AND '31-DEC-2024');
```

Is operator

It is used to compare with Null

Syntax: `COL_NAME IS NULL;`

Wapto name and sal of emp if the emp not getting sal

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL IS NULL;
```

Wapto name and comm of emp if the emp not getting commission

```
SELECT ENAME, COMM
FROM EMP
WHERE COMM IS NULL;
```

Wapto details of emp if emp don't have reporting manager (MGR)

```
SELECT *
FROM EMP
WHERE MGR IS NULL;
```

Is-Not operator

It is used to compare with not Null.

Syntax `COL_NAME IS NOT NULL;`

Wapto name and salary of emp if emp is earning salary

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL IS NOT NULL;
```

Wapto details of emp earning sal but not commission

```
SELECT *
FROM EMP
WHERE SAL IS NOT NULL AND COMM IS NULL;
```

Wapto earning salary as well as commission

```
SELECT *
FROM EMP
WHERE SAL IS NOT NULL AND COMM IS NOT NULL;
```

Like operator

It is used for pattern matching.

Syntax:

```
COLNAME LIKE PATTERN
```

Note:

To find the pattern we use percentile (%) and Underscore (_)

percentile used to fetch remaining characters

Underscore used to fetch particular missing character

waqtd whose name starts with 'A'

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'A%';
```

waqtd whose name ends with 'A'

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%A';
```

waqtd name of emp whose name consists character 'A'

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%A' OR ENAME LIKE 'A%' OR ENAME LIKE '%A%';
```

waqtd name of emp whose name consists character 'A' twice

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%AA%' OR ENAME LIKE 'A%A' OR ENAME LIKE '%A%A' OR ENAME LIKE '%AA' OR ENAME LIKE 'A%A%';
```

waqtd name of emp whose name consists of consecutive L's

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%LL%';
```

waqtd whose name starts with M and ends with character R.

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'M%' AND ENAME LIKE '%R';
```

waqtd whose name starts with M or J.

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'M%' OR ENAME LIKE 'J%';
```

waqtd name of emp whose name has exact four characters.

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '____';
```

waqtd name of emp whose name last but second character is 'M'

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%M_';
```

waqtd name and sal of emp whose sal ends with 50.

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL LIKE '%50';
```

NOT LIKE operator

It is similar to LIKE operator but instead of selecting, get rejected and remaining pattern will be displayed

syntax: `COL_NAME NOT LIKE PATTERN`

waqtd whose name does not start with 'A'

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE 'A%';
```

waqtd whose name does not end with 'A'

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE '%A';
```

waqtd whose name does not consists character 'A'

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE 'A%' AND ENAME NOT LIKE '%A' AND ENAME NOT LIKE '%A%';
```

waqtd whose name not consists character 'A' twice

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE 'A%A' AND ENAME NOT LIKE '%A%A' AND ENAME NOT LIKE '%AA'
AND ENAME NOT LIKE '%AA%' AND ENAME NOT LIKE 'A%A%';
```

waqtd emp name not consists consecutive L's.

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE '%LL%' AND ENAME NOT LIKE 'LL%' AND ENAME NOT LIKE '%LL';
```

waqtd emp name not start with M and end with R

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE 'M%J';
```

waqtd emp name who not start with M/J.

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE 'M%' AND ENAME NOT LIKE 'J%';
```

waqtd emp name whose name not contain exact 4 characters

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE '____';
```

waqtd emp name not last but second character is 'M'.

```
SELECT ENAME
FROM EMP
WHERE ENAME NOT LIKE '%M_';
```

Waqtd name and sal that not have sal ends with 50.

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL NOT LIKE '%50';
```

Functions

Function is a block of code or set of instructions given to perform specific task.

There are 2 types of functions:

1) User defined function
2) Built-In function

Userdefined

Sql is not a userdefined function, it is predefined

Built in functions are of 2 types

1) Single Row
2) Multi Row

Single Row

If we pass single input we get single output.

To the single row function we can pass 'n' number of inputs we get 'n' number of outputs

They are also known as scalar functions.

Multi Row

To the multirow function if we pass 'n' number inputs we get only single output.

It takes all the inputs as one shot and provides one output.

List of multirow function

1) MAX( )
2) MIN( )
3) AVG ( )
4) SUM ( )
5) COUNT ( )

MAX( )

It is used to find maximum values of column

MIN( )

It is used to find minimum values of column.

AVG( )

It is used to get the average value of column

SUM()

It is used to get the summation of values of column

COUNT()

It is used to get number of values of column

Characteristics of MultiRow Function:-

1) It can accept only one argument that is col name or Expression.
2) Multirow function will not accept NULL.
3) Along with multirow function we are not supposed to use any other col-name in SELECT clause.
4) We are not supposed to use MRF in WHERE clause.
5) Count function is the only MRF which can accept * (star) as an Argument.
6) Multiple MRF can be used in select clause.

Note: MRF are the group functions in sql.
MRF also called as Aggregate functions.

Wap to maximum sal of Employee

```
SELECT MAX(SAL)
FROM EMP;
```

Wap to minimum comm of Employee

```
SELECT MIN(COMM)
FROM EMP;
```

Wap to Average sal of the Employee table

```
SELECT AVG (SAL)
FROM EMP;
```

Wap to total salary given to Emp table

```
SELECT SUM(SAL)
FROM EMP;
```

Wap to no of salaries present in Emp table

```
SELECT COUNT (SAL)
FROM EMP;
```

Wap to no of comm in Emp table.

```
SELECT COUNT (COMM)
FROM EMP;
```

Wap to name and Comm of Emp, Earning Comm in dept 30.

```
SELECT ENAME, COMM
FROM EMP
WHERE DEPTNO IN 30 AND COMM IS NOT NULL;
```

Wqatd maximum salary given to Employee, if emp has character 's' in the name and working as manager in dept 10 with salary of more than 18000.

```
SELECT MAX(SAL)
FROM EMP
WHERE ENAME LIKE 'S%' OR ENAME LIKE '%S' AND JOB IN 'MANAGER' AND
DEPT NO IN 10 AND SAL>18000;
```

Grouping And Filtering

Group By clause

It is used to Group the records from the Table.

It executes Row By Row.

Characteristics of Group By Clause

It is used to Group the records based on the given arguments.

In Group By clause, column name or expression are used as argument

The column name or expression used for grouping can be used in SELECT clause

In Group By clause where clause is optional.

Group By clause can be executed with or without using WHERE clause

Syntax

```
SELECT MRF/EXPRESSION
FROM TABLE_NAME
[WHERE < FILTER CONDITION >]
GROUP BY COL_NAME / EXPRESSION;
```

Order of execution

1) FROM clause will execute first.
2) From the FROM clause we pass argument (Tablename)
3) Job of FROM clause is go to the database, Search for the table and put the table under execution.
4) After FROM clause execution, WHERE clause executed, but WHERE clause is optional.
5) After FROM clause execution, GROUP BY clause can be executed.
6) The Job of GROUP BY clause is to go to the table which is under execution and start executing row by row to group the records.
7) After the execution of Group By clause we get Groups.
8) After the execution of Group By clause SELECT clause is executed.
9) The job of SELECT clause is to goto Group; start execution to select the result and display it.

Group By Group

FROM CLAUSE —Table name / EMP→ DataBase

WHERE CLAUSE —filter / condition→ (optional)

GROUP BY CLAUSE —ROW-BY ROW→ Groups

SELECT CLAUSE —DEPTNO / GROUP BY GROUP→ Output

EMPNO	ENAME	SAL	DEPTNO
1	Srinu	100	10
2	Bhanu	200	20
3	Renu	300	30
4	Ram	1000	20

RESULT:

EMPNO	ENAME	SAL	DEPTNO
1	Srinu	100	10

EMPNO	ENAME	SAL	DEPTNO
2	Bhanu	200	20
4	Ram	1000	20

EMPNO	ENAME	SAL	DEPTNO
3	Renu	300	30

1) Waqtd number of emp working in each dept no

```
SELECT COUNT(*),DEPTNO
FROM EMP
GROUP BY DEPTNO;
```

2) Waqtd maximum sal given to each job

```
SELECT MAX(SAL),JOB
FROM EMP
GROUP BY JOB;
```

3) Waqtd number of distinct sal in employee table

```
SELECT COUNT(DISTINCT(SAL))
FROM EMP;
```

4) Waqtd number of emp working in each department except analyst

```
SELECT COUNT[ENAME],DEPTNO
FROM EMP
WHERE JOB!='ANALYST'
GROUP BY DEPTNO;
```

5) Waqtd number of emp working in each job, if emp has character 'A' in their names.

```
SELECT COUNT(ENAME),JOB
FROM EMP
WHERE ENAME LIKE '%A%'
GROUP BY JOB;
```

6) Waqtd number of emp working in each department except president

```
SELECT COUNT(ENAME),DEPTNO
FROM EMP
WHERE JOB !='PRESIDENT'
GROUP BY DEPTNO
```

7) Waqtd total sal needed to pay all the employees in each job.

```
SELECT SUM(SAL),JOB
FROM EMP
GROUP BY JOB;
```

8) Waqtd number of emp working as Manager in each department.

```
SELECT COUNT(ENAME),DEPTNO
FROM EMP
WHERE JOB='MANAGER'
GROUP BY DEPTNO;
```

9) wagtd average Sal needed to pay all the Employees in each department excluding the Emp of dept no 20

```
SELECT AVG(SAL), DEPTNO
FROM EMP
WHERE DEPTNO != 20
GROUP BY DEPTNO;
```

10) wagtd total salary needed to pay and no of Salesman in each department no.

```
SELECT SUM(SAL), COUNT(ENAME), DEPTNO
FROM EMP
WHERE JOB = 'SALESMAN'
GROUP BY DEPTNO;
```

Having Clause

→ It is used to Group filter condition.
→ Having clause is depends upon GROUP BY clause result.
→ Having clause cannot be executed without using GROUP BY clause.
→ We can use multirowfunctions in having clause.
→ After Execution of GROUP BY clause, Having clause will be executed.
→ After Execution of GROUP BY clause, Any clause executes that executes Group by Group Having clause executes Group By Group.

Syntax

```
SELECT GROUPFUNC/GROUP EXPRESSION
FROM TABLE_NAME
[WHERE <FILTER_CONDITION>]
GROUP BY COL_NAME / EXPRESSION
HAVING < GROUP_FILTER_CONDITION >;
```

Order of Execution

1) FROM → Row By Row
2) WHERE → Row By Row
3) HAVING → Group By Group
4) SELECT → Group By Group

→ FROM Clause Execute first.
→ From the FROM clause we can pass argument (Tablename)
→ The Job of FROM clause is to go to the Database and search for the table and put the table Under Execution.

→ After Execution of FROM clause we can use GROUP BY clause
→ From the GROUP BY clause we call the arguments
→ The job of GROUP By clause is to go to the table which is under Execution, and start executing row by row to group
→ After Execution of GROUPBY clause we get Groups.
→ After execution of GROUPBy clause, Having Clause Executed
→ From the Having clause, we pass an argument Group.Filter Condition.
→ The job of Having clause is to go to the Group and start Executing Group By Group and filter the Group Condition
→ After execution of Having clause, SELECT Clause will Execute
→ The job of SELECT clause to go to Group and start Executing Group By Group and Select table result and display it.

1. FROM CLAUSE —Tablename: Emp→ Database

2. GROUP BY CLAUSE —COL-name: DEPTNO – ROW BY ROW→

3. HAVING CLAUSE —GROUP: FILTER COUNT(*)→

4. SELECT ——→ Output
GROUP BY GROUP

EMPNO	ENAME	DEPTNO
1	Ganesh	10
2	Ramesh	20
3	Suresh	30
4	Prakash	10
5	Devi	10
6	Pavani	20
7	Chandni	20
8	Ravali	20
9	Pravali	30
10	Deepali	30

Result:

Displaying ENAMES working in each department having atleast 3.

DEPT 10 - GROUP

EMPNO	ENAME	DEPTNO
1	Ganesh	10
4	Prakash	10
5	Devi	10

DEPT20-GROUP

EMPNO	ENAME	DEPTNO
2	Ramesh	20
6	Pavani	20
7	Chandni	20
8	Ravali	20

DEPT30-GROUP

EMPNO	ENAME	DEPTNO
3	Suresh	30
9	Pravali	30
10	Deepali	30

1) Grouping based on DEPTNO
2) Then COUNT of EMP in DEPTNO
3) That count should be greater than or equal to 3 ⇒ DEPT10 ≥ 3 = TRUE
DEPT20 ≥ 3 = TRUE
DEPT30 ≥ 3 = TRUE

1) Waqtd no of Employees working in each department having atleast 3 employees in each dept no.

```
SELECT COUNT(*), DEPTNO
FROM EMP
GROUP BY DEPTNO
HAVING COUNT(*) ≥3;
```

2) Waqtd the names that are repeated

```
SELECT COUNT(*), ENAME
FROM EMP
GROUP BY ENAME
HAVING COUNT(*)>1;
```

3) Waqtd the names that are present 2 times.

```
SELECT ENAME, COUNT(*)
FROM EMP
GROUP BY ENAME
HAVING COUNT(*)=2;
```

4) Waqtd no of employees working each dept having exactly 4 employees in each deptno.

```
SELECT COUNT(*), DEPTNO
FROM EMP
GROUP BY DEPTNO
HAVING COUNT(*)=4;
```

5) Waqtd the sal that are repeated

```
SELECT COUNT(*), SAL
FROM EMP
GROUP BY SAL
HAVING COUNT(*)>1;
```

6) Order By clause

Order By clause is used to display the result either in ascending order or descending order.

Order By clause is optional

Order By clause can be used after the execution of all the clauses based on the concept.

Syntax for Ascending Order

```
ORDER BY COL_NAME;
```

Syntax for Descending Order

```
ORDER BY COLNAME DESC;
```

1) waqtd no of emp working in each department having atleast number by using Ascending Order.

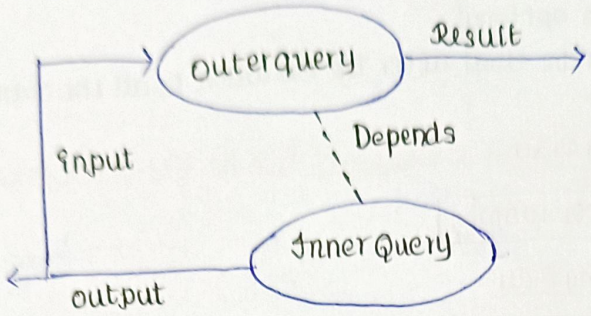
```
SELECT COUNT(*), DEPTNO
FROM EMP
GROUP BY DEPTNO
ORDER BY DEPTNO;
```

2) waqtd no of emp working in each department number by using Descending Order.

```
SELECT COUNT(*),DEPTNO
FROM EMP
GROUP BY DEPTNO
ORDER BY DEPTNO DESC;
```

SubQuery

A Query written inside another query is called as SubQuery.
SubQuery is also known as Bottom-Top Approach.



→ Inner query can also we call as 'Subquery'
→ Inner query will execute first and produces output
→ The output of innerquery can be taken as an input to the outer query.
→ The outer query will execute and generates result, therefore the outerquery is dipends on inner query

Why and when we use Subquery???

Case(1):
If there is unknown present in the question, to find that unknown we use subquery case 1.

1) Wqtd name and salary of employee earning salary more than Smith.

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL>(SELECT SAL FROM EMP WHERE ENAME='SMITH');
```

2) Wqtd name and salary of employee earning salary more than king

```
SELECT ENAME,SAL
FROM EMP
WHER SAL> (SELECT SAL
FROM EMP
WHERE ENAME='king');
```

3) Wqtd name and hiredate of emp hired after miller.

```
SELECT ENAME,HIREDATE
FROM EMP
WHERE HIREDATE >(SELECT HIREDATE FROM EMP WHERE ENAME='MILLER');
```

4) Waqtd name and deptno of emp working in the same department as Smith.

```
SELECT ENAME, DEPTNO
FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM EMP
WHERE ENAME = 'SMITH');
```

5) Waqtd details of emp working in the same designation as Jones.

```
SELECT * FROM EMP
WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'JONES');
```

6) Waqtd name and sal of emp earning sal more than Smith but less than King.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'SMITH')
AND SAL < (SELECT SAL FROM EMP WHERE ENAME = 'KING');
```

7) Waqtd details of emp working in same department as Smith earning sal more than king hired after Martin working in the same job as Turner.

```
SELECT * FROM EMP
WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'SMITH')
AND SAL > (SELECT SAL FROM EMP WHERE ENAME = 'KING') AND HIREDATE >
(SELECT HIREDATE FROM EMP WHERE ENAME = 'MARTIN')
AND JOB = (SELECT JOB FROM EMP WHERE ENAME = 'TURNER');
```

8) Waqtd emp details earning sal more than smith working in the same department as Jones.

```
SELECT *
FROM EMP
WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'SMITH')
AND DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'JONES');
```

9) Waqtd details of emp earning sal more than 2000 working in the same designation as Blake.

```
SELECT * FROM EMP
WHERE SAL > 2000 AND JOB = (SELECT JOB FROM EMP WHERE ENAME = 'BLAKE');
```

10) waqtd details of emp hired before martin earning sal less than 3000

```
SELECT * FROM EMP
WHERE HIREDATE < (SELECT HIREDATE FROM EMP WHERE ENAME='MARTIN')
AND SAL < 3000;
```

case (2)

Whenever the data is present in multiple table to fetch the data we use subquery.

1) waqtd deptname of smith

```
SELECT DNAME FROM DEPT
WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME='SMITH');
```

2) waqtd dname and loc of emp whose name is king

```
SELECT DNAME, LOC
FROM DEPT
WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME='KING');
```

3) waqtd name of emp working in the loc 'chicago'

```
SELECT ENAME FROM EMP WHERE DEPTNO=(SELECT DEPTNO FROM DEPT WHERE LOC='CHICAGO');
```

4) waqtd name and sal of emp working in Sales department

```
SELECT ENAME, SAL FROM EMP
WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME='SALES');
```

5) waqtd details of the emp working in the same department as Smith earning sal more than Adams, working in the same JOB as Miller, hired before martin working in the location 'Boston'.

```
SELECT * FROM EMP WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME='MILLER') AND
DEPTNO = (SELECT DEPTNO FROM DEPT WHERE LOC = 'NEWYORK');
```

6) wagtd details of emp working in same designation as Miller and working in the location Newyork.

```
SELECT * FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='MILLER')
AND DEPTNO=(SELECT DEPTNO FROM DEPT WHERE LOC='NEWYORK');
```

Maximum And Minimum

1) wagtd name of emp earning max salary

```
SELECT ENAME FROM EMP SAL WHERE SAL=(SELECT MIN(SAL) FROM EMP);
```

2) wagtd name and sal of emp earning min salary

```
SELECT ENAME,SAL FROM EMP WHERE SAL=(SELECT MIN(SAL) FROM EMP) ;
```

3) wagtd the greatest employee number.

```
SELECT ENAME, EMPNO FROM EMP WHERE EMPNO=(SELECT MAX(EMPNO) FROM EMP);
```

NOTE:

In subquery corresponding column name not be same but their datatypes must be same.

In subquery we are not supposed to use more than one argument in the innerquery SELECT clause

4) wagtd name and sal of emp having least employee number

```
SELECT ENAME,SAL FROM EMP WHERE EMPNO=(SELECT MIN(EMPNO) FROM EMP);
```

⑤ Waqtd name and comm of emp having highest commission

```
SELECT ENAME,COMM FROM EMP WHERE COMM=(SELECT MAX(COMM) FROM EMP);
```

⑥ Waqtd name and hiredate of emp hired before all the emp (First emp)

```
SELECT ENAME, HIREDATE FROM EMP WHERE HIREDATE=(SELECT MIN(HIREDATE)
FROM EMP);
```

⑦ Waqtd name and hiredate of emp hired at the last

```
SELECT ENAME, HIREDATE FROM EMP WHERE HIREDATE=(SELECT MAX(HIREDATE)
FROM EMP);
```

⑧ Waqtd name and comm of emp who earns minimum commission

```
SELECT ENAME, COMM FROM EMP WHERE COMM=(SELECT MIN(COMM) FROMEMP);
```

⑨ Waqtd name, sal, comm of emp who earns maximum commission

```
SELECT ENAME,SAL,COMM FROM EMP WHERE COMM=(SELECT MAX(COMM) FROM EMP);
```

⑩ Waqtd details of emp earning least annual salary

```
SELECT * FROM EMP WHERE SAL*12 =(SELECT MIN(SAL*12) FROM EMP);
```

Types of Subqueries

1) Single Row Subquery
2) Multi Row Subquery

Single Row Subquery:-

If a query returns exact one value we can call it as single row subquery
If a subquery returns exact one value we can use normal operator (=)
or special operator (in operator)

Wapto dname of Smith

```
SELECT DNAME FROM EMP WHERE DEPTNO IN(SELECT DEPTNO FROM EMP
WHERE ENAME='SMITH');
```

(or)

```
SELECT DNAME FROM EMP WHERE DEPTNO =(SELECT DEPTNO FROM EMP
WHERE ENAME ='SMITH');
```

Multi Row Subquery

If a subquery returns more than one value we can call it as multirow subquery.
If a subquery returns morethan we are not supposed to use normal operator(=) we must use special operator (in operator)

Wapto dname of smith and king

```
SELECT DNAME FROM EMP WHERE DEPTNO IN(SELECT DEPTNO FROM EMP
WHERE ENAME ='SMITH' OR ENAME='KING');
```

SubQuery operator

ALL operator
ANY operator
EXIST operator
NOT EXIST operator

All operator

All operator is a special operator used along with the relational operator to compare the value at RHS
All operator returns True, if all the condition at RHS have satisfied
All operator acts as an AND operator

1) Wapto name and Annualsal of emp earning annual sal more than all the salesman

```
SELECT ENAME,SAL*12
FROM EMP
WHERE SAL*12 > ALL (SELECT SAL*12
FROM EMP WHERE JOB = 'SALESMAN');
```

2) Wapto name and salary of emp earning salary more than all managers

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL> ALL (SELECT SAL
FROM EMP
WHERE JOB ='MANAGER');
```

3) Wapto name and sal of emp earning sal less than all the managers.

```
SELECT ENAME,SAL
FROM EMP
WHERE SAL < ALL (SELECT SAL
FROM EMP
WHERE JOB ='MANAGER');
```

Any operator

Any operator is a special operator which is used to along with relational operator to compare the value AT RHS.

Any operator returns TRUE, if any operator condition at RHS satisfied

Any operator acts like OR operator.

1) Wapto name and Annualsal of emp earning annualsal more than Any of salesman

```
SELECT ENAME, SAL*12 FROM EMP WHERE SAL*12 > ANY (SELECT SAL*12
FROM EMP WHERE JOB ='SALESMAN');
```

2) Wapto name and salary of emp earning salary more than any managers

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ANY (SELECT SAL
FROM EMP WHERE JOB ='MANAGER');
```

3) Wapto name and sal of emp earning sal less than any managers.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL < ANY (SELECT SAL
FROM EMP WHERE JOB ='MANAGER');
```

Note: It is difficult to find whether query belongs to single row subquery or multirow subquery.
We suggest to use special operator in (In operator) instead of normal operator (=)

Nested Subquery

A subquery written inside another query is known as Nested subquery.
We can nest upto 255 queries

waqtd second maximum salary

```
SELECT MAX(SAL) FROM EMP
WHERE SAL<(SELECT MAX(SAL)
FROM EMP);
```

waqtd third maximum salary

```
SELECT MAX(SAL)
FROM EMP
WHERE SAL<(SELECT MAX(SAL)
FROM EMP
WHERE SAL<(SELECT MAX(SAL)
FROM EMP));
```

waqtd fourth maximum salary

```
SELECT MAX(SAL)
FROM EMP
WHERE SAL<(SELECT MAX(SAL) FROM EMP
WHERE SAL<(SELECT MAX(SAL) FROM EMP
WHERE SAL<(SELECT MAX(SAL) FROM EMP)));
```

waqtd fifth maximum salary

```
SELECT MAX(SAL)
FROM EMP
WHERE SAL< (SELECT MAX(SAL) FROM EMP
WHERE SAL< (SELECT MAX(SAL) FROM EMP
WHERE SAL <(SELECT MAX(SAL) FROM EMP
WHERE SAL <(SELECT MAX(SAL) FROM EMP))));
```

waqtd name of emp earning second max salary

```
SELECT ENAME
FROM EMP
WHERE SAL=(SELECT MAX(SAL) FROM EMP
WHERE SAL< (SELECT MAX(SAL) FROM EMP));
```

Waqtd Second min salary

```
SELECT MIN(SAL)
FROM EMP
WHERE SAL>(SELECT MIN(SAL)
FROM EMP);
```

waqtd third min Salary

```
SELECT MIN(SAL)
FROM EMP
WHERE SAL> (SELECT MIN(SAL) FROM EMP
WHERE SAL> (SELECT MIN(SAL) FROM EMP));
```

waqtd fourth min Salary

```
SELECT MIN(SAL)
FROM EMP
WHERE SAL >(SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP)));
```

waqtd fifth min Salary

```
SELECT MIN(SAL)
FROM EMP
WHERE SAL>(SELECT MIN(SAL) FROM EMP
WHERE SAL> (SELECT MIN(SAL) FROM EMP
WHERE SAL> (SELECT MIN(SAL) FROM EMP
WHERE SAL> (SELECT MIN(SAL) FROM EMP))));
```

Waqtd dname, loc Of Emp Earning third mal Salary

```
SELECT DNAME, LOC FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO FROM EMP
WHERE SAL=(SELECT MAX(SAL) FROM EMP
WHERE SAL<(SELECT MAX(SAL) FROM EMP
WHERE SAL<(SELECT MAX(SAL) FROM EMP))));
```

waqtd dname, deptno of Emp Earning 11th Men Salary (Research, 20)

```
SELECT DNAME, DEPTNO FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO FROM EMP
WHERE SAL = (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP))))))))));
```

Employee Manager Relation

case(1):

To find reporting manager of employee

MGR = EMPNO

EMPNO	ENAME	MGR
1	A	2
2	B	3
3	C	4
4	D	1

1) wap to get name of smith manager

```
SELECT ENAME FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE ENAME = 'SMITH');
```

2) wap to get sal of Adams manager

```
SELECT SAL FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE ENAME = 'ADAMS');
```

3) Waqtd details of miller manager

```
SELECT * FROM EMP
WHERE EMPNO =(SELECT MGR FROM EMP
WHERE ENAME='MILLER');
```

4) Waqtd loc of Jones Manager

```
SELECT LOC FROM DEPT
WHERE DEPTNO=(SELECT DEPTNO FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE ENAME='JONES'));
```

5) Waqtd name of adams Manager's manager

```
SELECT ENAME FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE ENAME='ADAMS'));
```

6) Waqtd dname and loc of Allen's Manager's Manager

```
SELECT DNAME,LOC FROM EMP DEPT
WHERE DEPTNO IN(SELECT DEPTNO FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE ENAME='ALLEN')));
```

7) Waqtd dname of Smith manager's manager's manager

```
SELECT DNAME FROM EMP DEPT
WHERE DEPTNO IN(SELECT DEPTNO FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE EMPNO=(SELECT MGR FROM EMP
WHERE ENAME='SMITH'))));
```

8) wapto details of dept table of Smith managers manager's managers managers manager.

```
SELECT * FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE ENAME = 'SMITH'))))));
```

case(a):

To find the employee who are reporting to manager

EMPNO = MGR

EMPNO	ENAME	MGR
1	A	2
2	B	3
3	C	4
4	D	1

1) wapto name of employee's reporting to king

```
SELECT ENAME FROM EMP
WHERE MGR = (SELECT EMPNO FROM EMP
WHER ENAME='KING');
```

2) wapto dname of emp reporting to king

```
SELECT DNAME FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO FROM EMP
WHERE MGR = (SELECT EMPNO FROM EMP
WHERE ENAME = 'KING'));
```

3) Wrqtd dname of Jones manager

```
SELECT DNAME FROM DEPT
WHERE DEPTNO IN (SELECT DEPTNO FROM EMP
WHERE MGR EMPNO = (SELECT MGR FROM EMP
WHERE ENAME = 'JONES'));
```

4) Wrqtd Millers managers Salary

```
SELECT SAL FROM EMP
WHERE EMPNO = (SELECT MGR FROM EMP
WHERE ENAME = 'MILLER');
```

5) Wrqtd details of emp reporting to Jones

```
SELECT * FROM EMP
WHERE MGR = (SELECT EMPNO FROM EMP
WHERE ENAME = 'JONES');
```

6) Wrqtd number of emp reporting to Ford's manager

```
SELECT COUNT(*) FROM EMP
WHERE MGR = (SELECT EMPNO FROM EMP
WHERE ENAME = 'FORD');
```

7) Wagtd number of emp reporting to King

```
SELECT COUNT(*) FROM EMP
WHERE MGR=(SELECT EMPNO FROM EMP
WHERE ENAME='KING');
```

8) Wagtd number of emp reporting to Smith

```
SELECT COUNT(*) FROM EMP
WHERE MGR=(SELECT EMPNO FROM EMP
WHERE ENAME='SMITH');
```