Experiment 4

**Name** : Mohammad Wasi

**SAP ID** : 500110709

**Batch** : AIML B8

Aim: Use of Inbuilt functions and relational algebra operation

**Objective :**

To employ SQL queries incorporating inbuilt functions and relational algebra operations to extract and analyse data, facilitating understanding of database management principles and enhancing query-writing skills.

Theory :

SQL commands and keywords play a crucial role in database management. Here's a bit more detail on some commonly used ones:

**SELECT**: This command retrieves data from one or more tables in the database. It allows you to specify which columns you want to retrieve and any filtering criteria using the WHERE clause.

**WHERE**: The WHERE clause filters rows based on specified conditions. It allows you to narrow down the result set by specifying criteria that the rows must meet.

**JOIN**: JOIN is used to combine rows from two or more tables based on a related column between them. It allows you to retrieve data from multiple tables in a single query, linking them together based on a common column.

**ORDER BY:** This clause sorts the result set in ascending or descending order based on one or more columns. It helps in organizing the data for better analysis and presentation.

**Code**

**-- Create EMP table**

CREATE TABLE EMP (

EMPNO INT PRIMARY KEY,

ENAME VARCHAR(20),

JOB VARCHAR(20),

MGR INT,

HIREDATE VARCHAR(20),

SAL INT,

COMM INT,

DEPTNO INT

);

**-- Create DEPT table**

CREATE TABLE DEPT (

DEPTNO INT PRIMARY KEY,

DNAME VARCHAR(20),

LOC VARCHAR(20)

);

**-- Insert data into EMP table**

INSERT INTO EMP VALUES (7369, 'SMITH', 'CLERK', 7902, '17-DEC-80', 500, 800, 20);

INSERT INTO EMP VALUES (7499, 'ALLEN', 'SALESMAN', 7698, '20-FEB-81', 1600, 300, 30);

INSERT INTO EMP VALUES (7521, 'WARD', 'SALESMAN', 7698, '22-FEB-81', 1250, 500, 30);

INSERT INTO EMP VALUES (7566, 'JONES', 'MANAGER', 7839, '02-APR-81', 2975, NULL, 20);

INSERT INTO EMP VALUES (7654, 'MARTIN', 'SALESMAN', 7698, '28-SEP-81', 1250, 1400, 30);

INSERT INTO EMP VALUES (7698, 'BLAKE', 'MANAGER', 7839, '01-MAY-81', 2850, NULL, 30);

INSERT INTO EMP VALUES (7782, 'CLARK', 'MANAGER', 7839, '09-JUN-81', 2450, NULL, 10);

INSERT INTO EMP VALUES (7788, 'SCOTT', 'ANALYST', 7566, '09-DEC-82', 3000, NULL, 20);

INSERT INTO EMP VALUES (7839, 'KING', 'PRESIDENT', NULL, '17-NOV-81', 5000, NULL, 10);

INSERT INTO EMP VALUES (7844, 'TURNER', 'SALESMAN', 7698, '08-SEP-81', 1500, 0, 30);

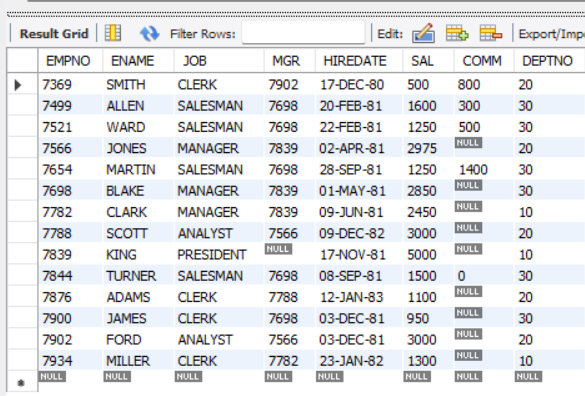
INSERT INTO EMP VALUES (7876, 'ADAMS', 'CLERK', 7788, '12-JAN-83', 1100, NULL, 20);

INSERT INTO EMP VALUES (7900, 'JAMES', 'CLERK', 7698, '03-DEC-81', 950, NULL, 30);

INSERT INTO EMP VALUES (7902, 'FORD', 'ANALYST', 7566, '03-DEC-81', 3000, NULL, 20);

INSERT INTO EMP VALUES (7934, 'MILLER', 'CLERK', 7782, '23-JAN-82', 1300, NULL, 10);

select\* from EMP;



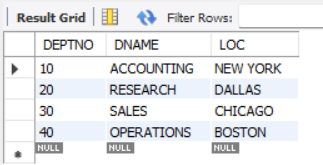
**-- Insert data into DEPT table**

INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');

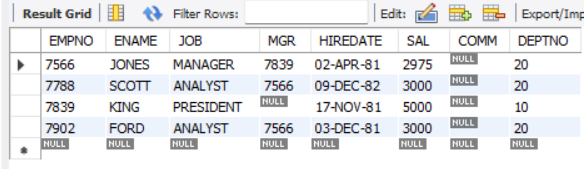
INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');



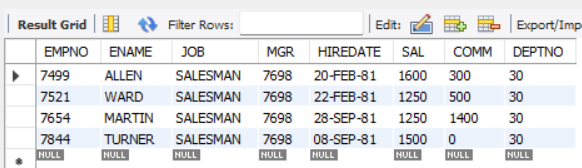
**-- 1. List the details of the employees whose salaries are more than the employee BLAKE.**

SELECT \* FROM EMP WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'BLAKE');



**-- 2. List the employees whose jobs are the same as ALLEN.**

SELECT \* FROM EMP WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'ALLEN');

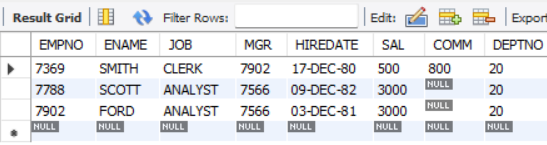


**-- 3. List the employees whose salary is the same as FORD or SMITH in descending order of names.**

SELECT \* FROM EMP

WHERE SAL IN (SELECT SAL FROM EMP WHERE ENAME IN ('FORD', 'SMITH'))

ORDER BY ENAME DESC;

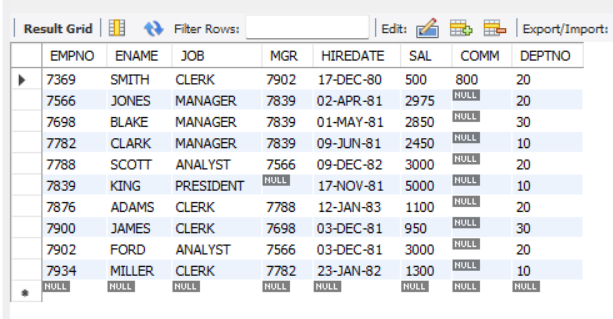


**-- 4. List the employees whose jobs are the same as MILLER or salary is more than ALLEN.**

SELECT \* FROM EMP

WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'MILLER')

OR SAL > (SELECT SAL FROM EMP WHERE ENAME = 'ALLEN');

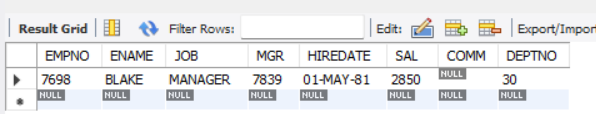


**-- 5. Find the highest paid employee of the sales department.**

SELECT \* FROM EMP

WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'SALES')

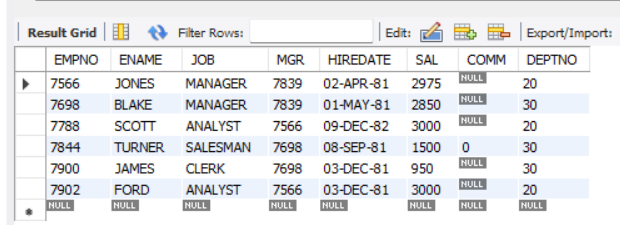
AND SAL = (SELECT MAX(SAL) FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'SALES'));



**-- 6. List the employees who are senior to the most recently hired employee working under KING.**

SELECT \* FROM EMP

WHERE HIREDATE < (SELECT MAX(HIREDATE) FROM EMP WHERE MGR = (SELECT EMPNO FROM EMP WHERE ENAME = 'KING'));



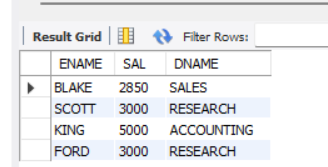
**-- 7. List the names of the employees who are getting the highest salary department-wise.**

SELECT E.ENAME, E.SAL, D.DNAME

FROM EMP E

JOIN DEPT D ON E.DEPTNO = D.DEPTNO

WHERE (E.SAL, E.DEPTNO) IN (SELECT MAX(SAL), DEPTNO FROM EMP GROUP BY DEPTNO);



**-- 8. List the employees whose salary is equal to the average of the maximum and minimum.**

SELECT \*

FROM EMP

WHERE SAL = (

SELECT AVG(SAL)

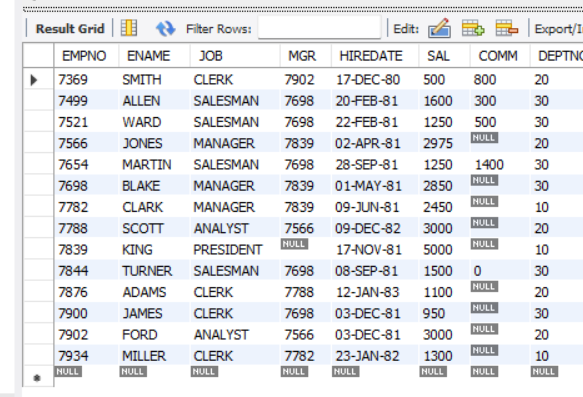
FROM (

SELECT MAX(SAL) AS MAX\_SAL, MIN(SAL) AS MIN\_SAL

FROM EMP

) sub

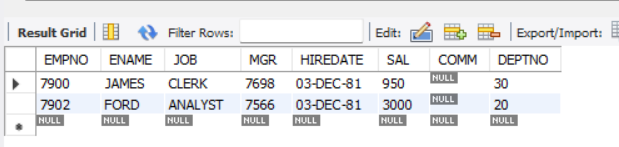
);

);

**-- 9. List the employees who joined the company on the same date.**

SELECT \* FROM EMP

WHERE HIREDATE IN (SELECT HIREDATE FROM EMP GROUP BY HIREDATE HAVING COUNT(\*) > 1);



**-- 10. Find out the employees who joined the company before their managers.**

SELECT \*

FROM EMP E1

WHERE EXISTS (

SELECT 1

FROM EMP E2

WHERE E1.MGR = E2.EMPNO

AND E1.HIREDATE < E2.HIREDATE

);