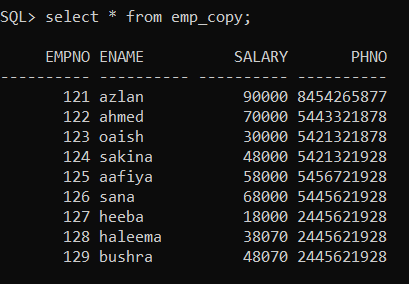
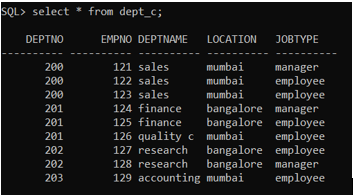
# **Name: Abdurrahman Qureshi**

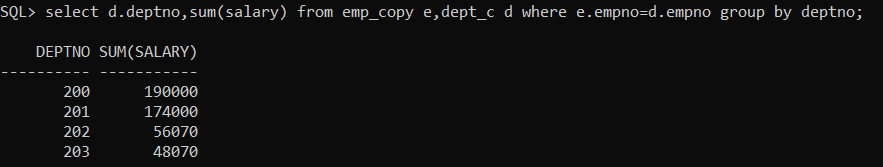
# **Roll No: 210451**

Practical No: 8

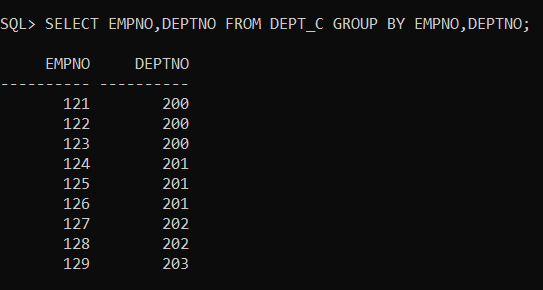
EXECUTE QUERIES USING THE SELECT COMMAND WITH WHERE, HAVING,GROUP BY AND ORDER BY CLAUSES.

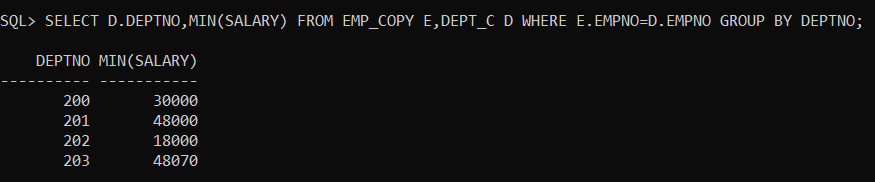
1. SELECT D.DEPTNO,SUM(SALARY) FROM EMP E,DEPT D WHERE E.EMPNO=D.EMPNO



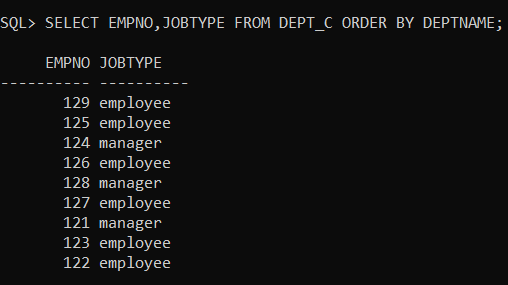
B.SELECT EMPNO,DEPTNO FROM DEPT GROUP BY EMPNO,DEPTNO



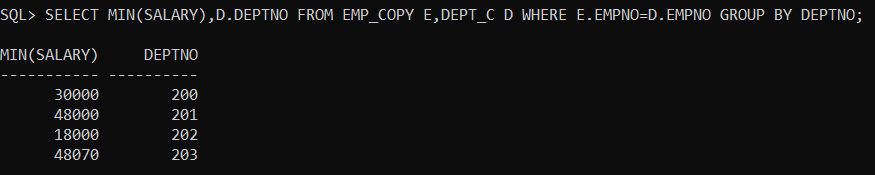
C.SELECT D.DEPTNO,MIN(SALARY) FROM EMP\_COPY E,DEPT\_C D WHERE EMPNO=D.EMPNO GROUP BY DEPTNO;



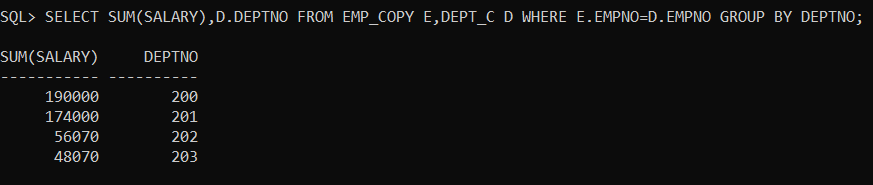
D.SELECT EMPNO,JOBTYPE FROM DEPT ORDER BY DEPTNAME;



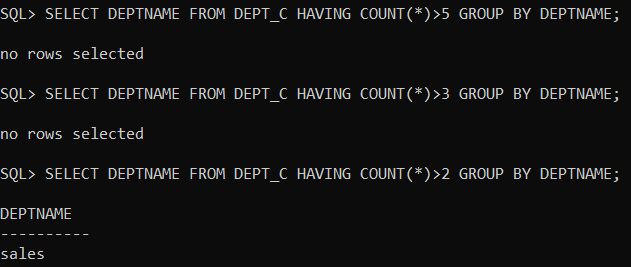
A. DISPLAY MINIMUM SALARYOF EMPLOYEE FROM EVERY DEPARTMENT.



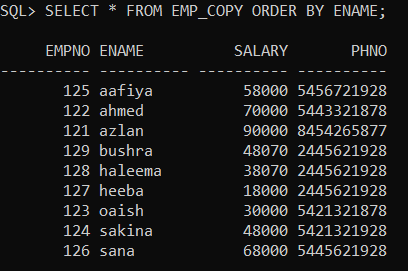
B. DISPLAY TOTAL SALARY OF EVERY DEPARTMENT.



C. DISPLAY THE DEPARTMENT HAVING TOTAL EMPLOYEES MORE THAN 5.

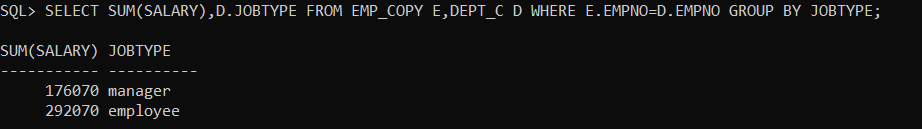


D. DISPLAY DETAILS OF EMPLOYEES WITH THE EMPLOYEE NAME IN ASCENDING ORDER.

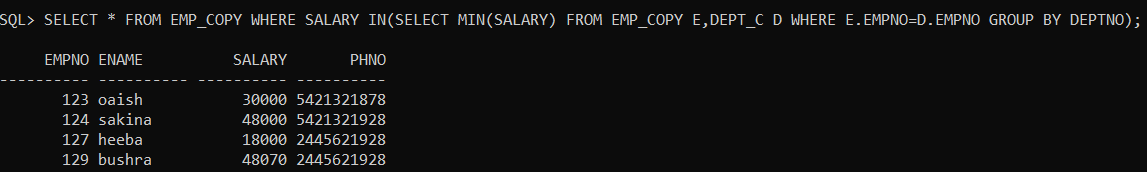


X|V.EXERCISE.

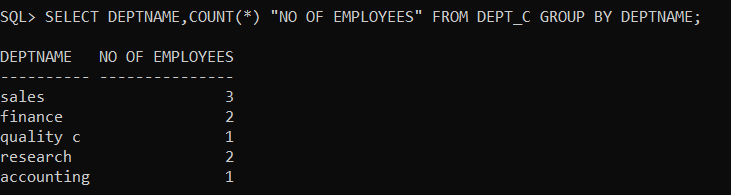
A. DISPLAY TOTAL SALARY SPENT FOR EACH JOB CATEGORY.



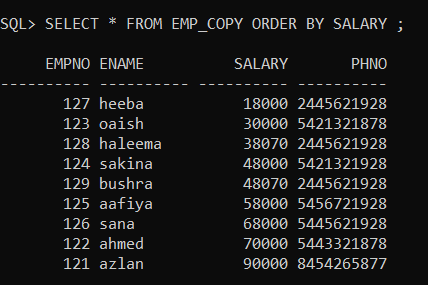
B. DISPLAY LOWEST PAID EMPLOYEE DETAILS UNDER EACH DEPARTMENT.



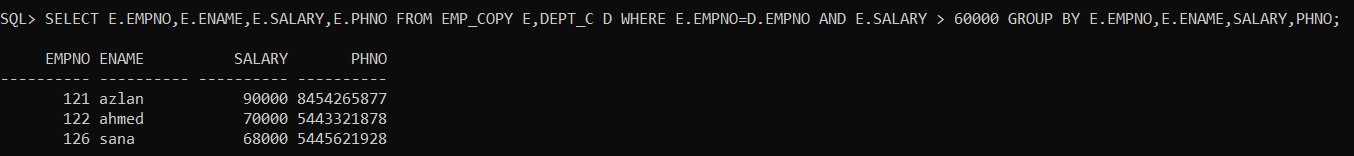
C. DISPLAY NUMBER OF EMPLOYEES WORKING IN EACH DEPARTMENT AND THEIR DEPARTMENT NAME.



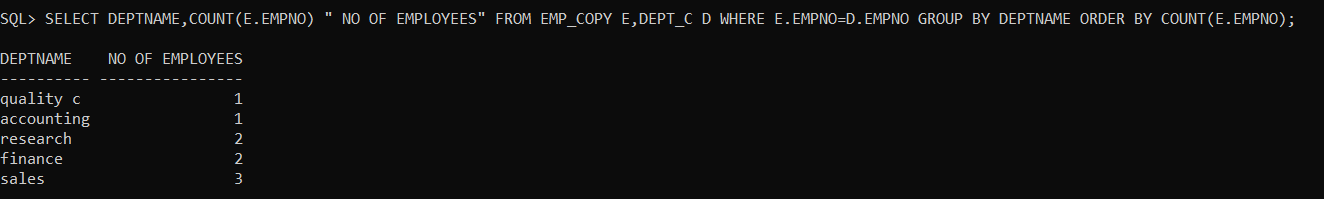
D.DISPLAY DETAILS OF EMPLOYEES WITH SALARY IN INCREASING ORDER.



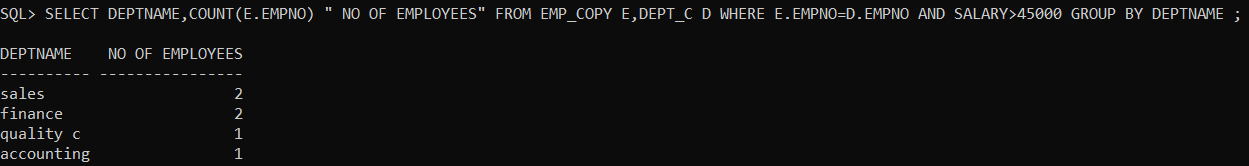
E. DISPLAY THE DETAILS OF EMPLOYEES EARNING SALARY GREATER THAN 60000 FROM EVERY DEPARTMENT.



F. LIST THE NUMBER OF EMPLOYEES FROM EVERY DEPARTMENT IN ASCENDING ORDER.



G. LIST THE EMPLOYEES FROM EVERY DEPARTMENT GETTING SALARY MORE THAN 45000.

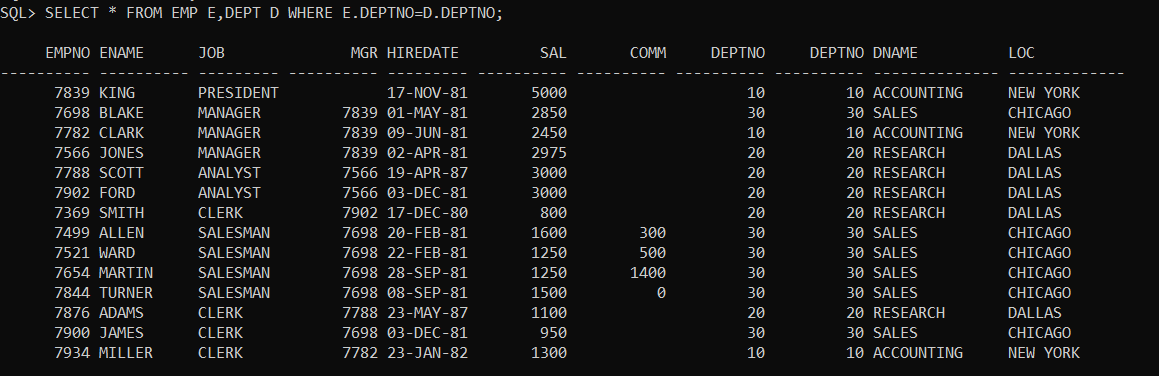
****

# **Name: Abdurrahman Qureshi**

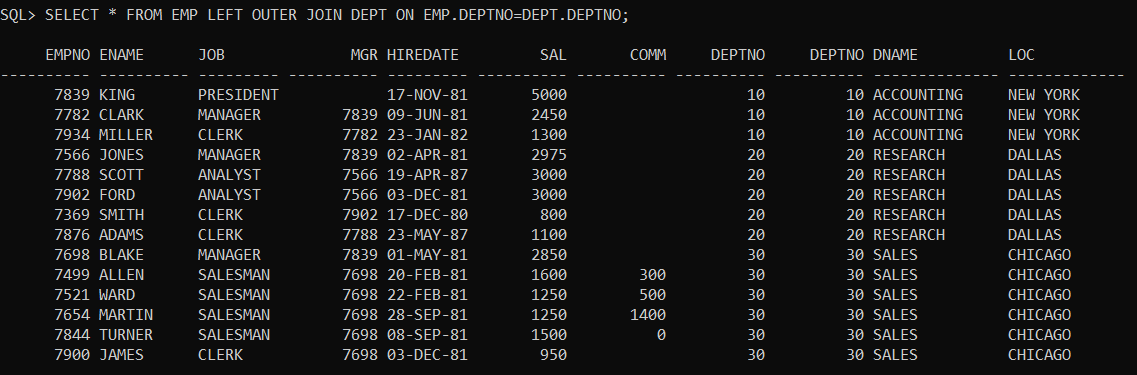
# **Roll No: 210451**

Practical No: 9

1.SELECT \* FROM EMP,DEPT WHERE EMP.DEPT NO =DEPT.DEPTNO.

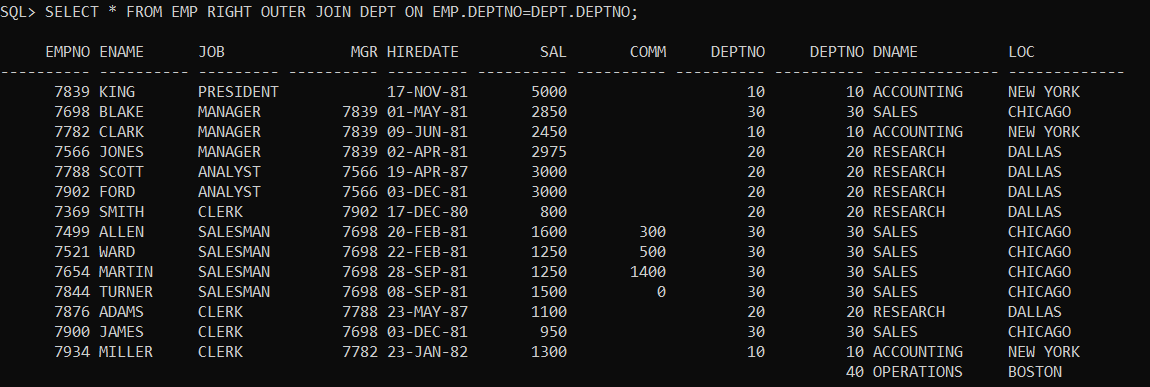


2.SELECT \* FROM EMP LEFT OUTER JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;

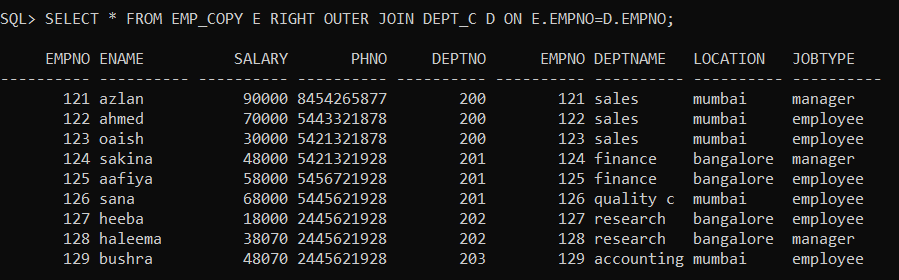


3. SELECT \* FROM EMP RIGHT OUTER JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;

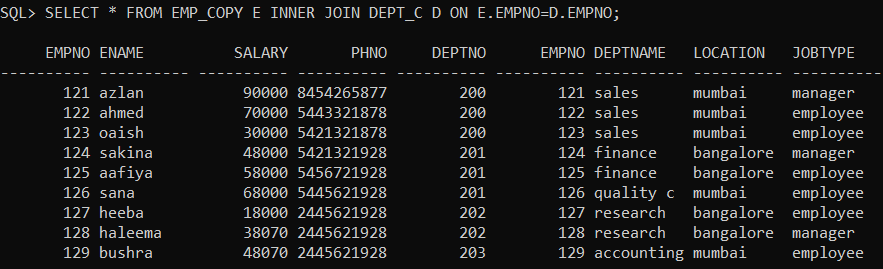
4. SELECT \* FROM EMP FULL OUTER JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;



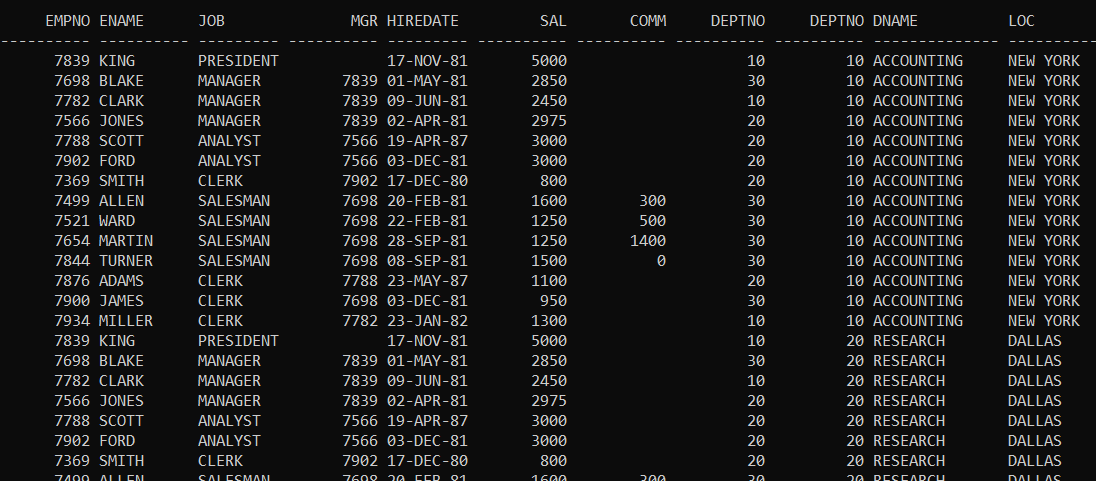
1.PERFORM RIGHT JOIN ON EMP AND DEPT TABLE.

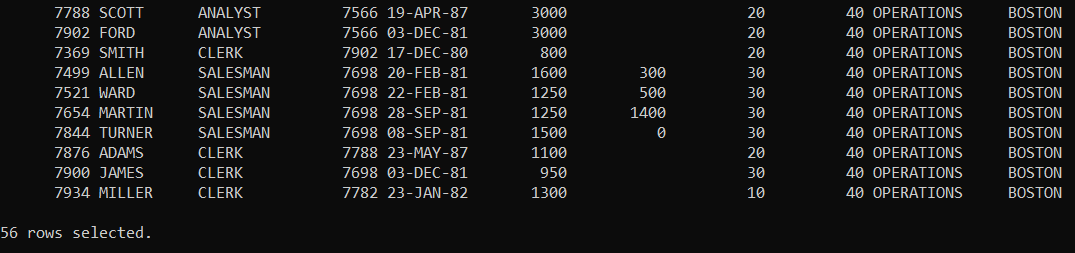


2. PERFORM INNER JOIN ON EMP AND DEPT TABLE.



3. PERFORM CROSS JOIN WITH SUITABLE EXAMPLE.



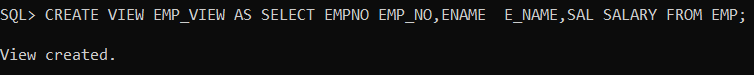


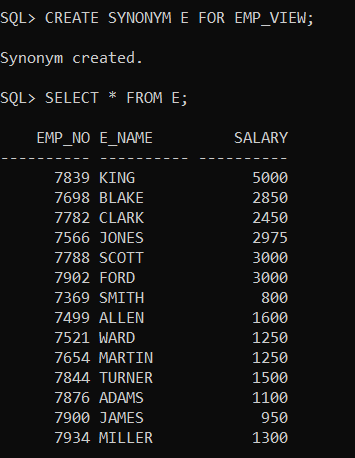
# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

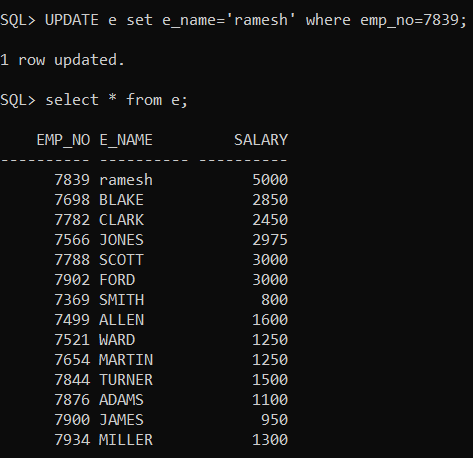
Practical No: 10

1.CREATE VIEW EMP\_VIEW AS SELECT EMP\_NO,E\_NAME,SALARY FROM EMP;

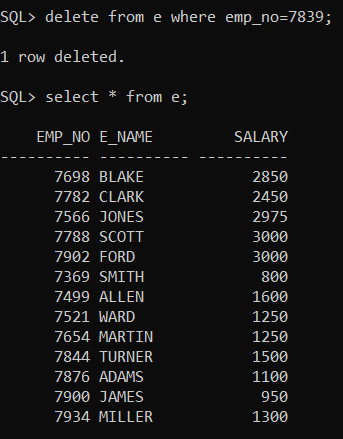




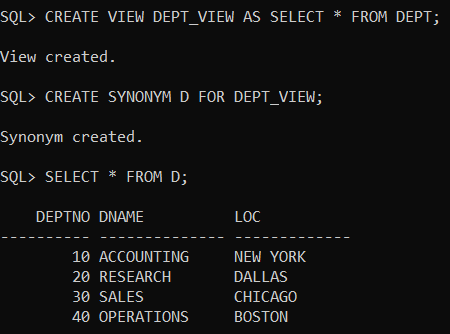
2.UPDATE EMP\_VIEW SET E\_NAME=’RAMESH’ WHERE EMP\_NO = 1001;



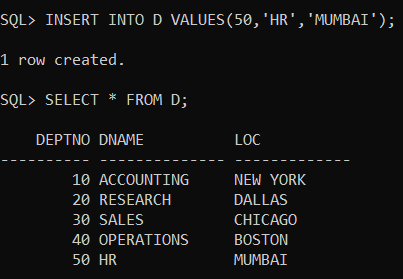
3.delete from emp\_view where emp\_no=1005;



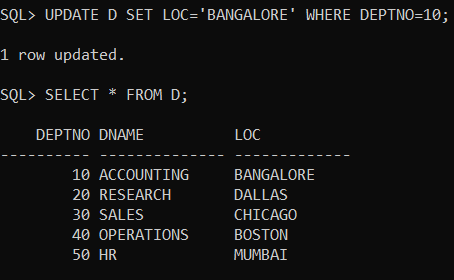
1.CREATE DEPT\_VIEW ON DEPT TABLE;



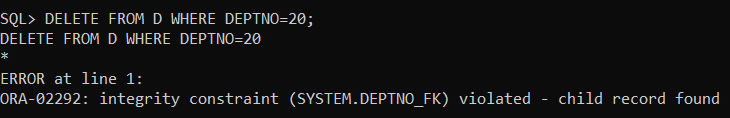
2.INSERT NEW RECORD IN DEPT\_VIEW VIEW;



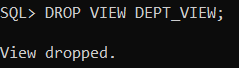
3.MODIFY LOCATION OF DEPNO 10 OF DEPT\_VIEW;



4.DELETE THE RECORD OF DEPTNO 20 FROM DEPT\_VIEW VIEW;



5.DELETE THE VIEW DEPT\_VIEW;

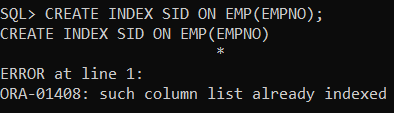


# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

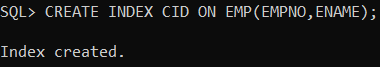
Practical No: 11

1.CREATE INDEX SID ON EMP(EMPNO);



WE CANNOT INDEX PRIMARY KEY AS THEY ARE AUTOMATICALLY INDEXED DURING THE TIME OF TABLE CREATION.

2.CREATE INDEX CID ON EMP(EMPNO,ENAME);



3.CREATE SEQUENCE EMP\_SEQUENCE

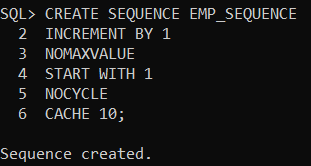
INCREMENT BY 1

START WITH 1

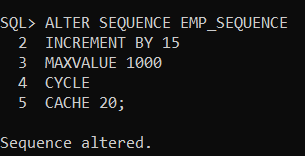
NOMAXVALUE

NOCYCLE

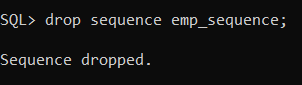
CACHE 10;



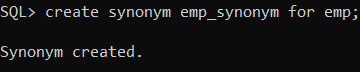
4.ALTER SEQUENCE:



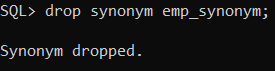
5. drop sequence emp\_sequence;



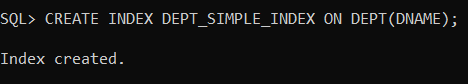
6. create synonym emp\_sys for emp;



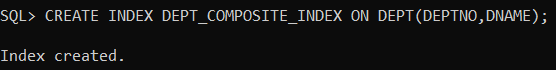
7. drop synonym emp\_sys;



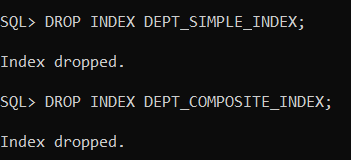
1. CREATE SIMPLE INDEX DEPT\_SIMPLE\_INDEX ON DEPT TABLE;



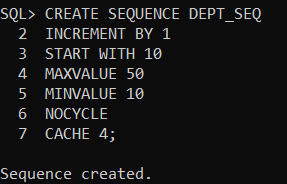
2.CREATE COMPOSITE INDEX DEPT\_COMPOSITE\_INDEX ON DEPT TABLE;



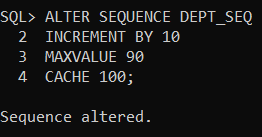
3.DROP BOTH THE INDEXES:



4.CREATE SEQUNECE DEPT\_SEQUENCE ON DEPT TABLE;



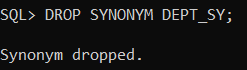
5.ALTER SEQUENCE



6.CREATE SYNONYM DEPT\_SY ON DEPT TABLE;



7.DROP SYNONYM;

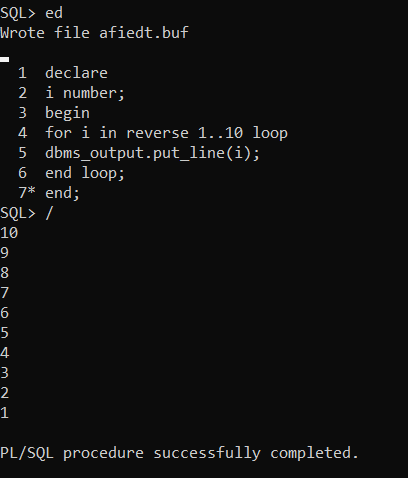


# **Name: Abdurrahman Qureshi**

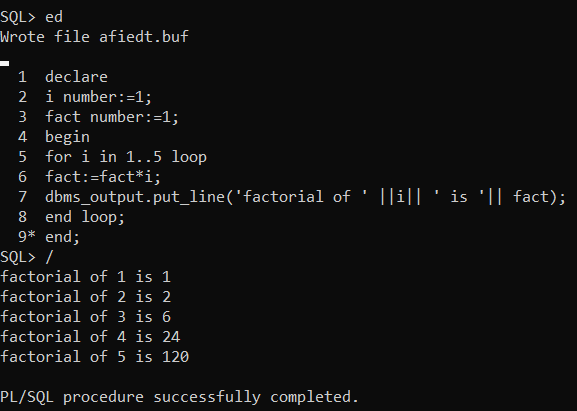
# **Roll No: 210451**

Practical No: 12

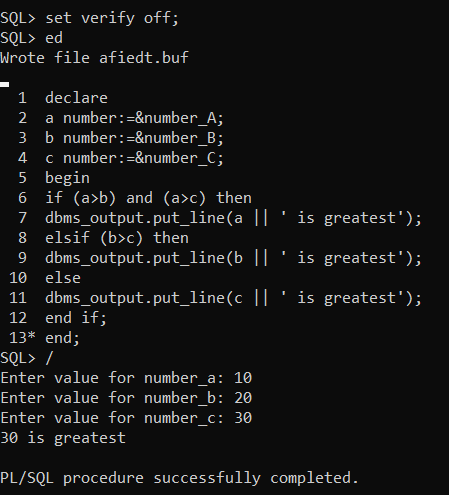
1.write a pl/sql program to display 1 to 10 numbers in reverse order using for loop:



2.write a pl/sql program to find factorial of number 5 using while loop:



1.WRITE A PL/SQL PROGRAM TO ACCEPT THREE NUMBERS AND SIPLAY THE LARGEST NUMBER:



2. WRITE A PL/SQL PROGRAM TO DISPLAY EVEN NUMBERS BETWEEN 1 TO 100:

