Database Management Systems

Lab Manual

Submitted in the partial fulfillment of the
requirements for the award of Degree of

Bachelor of Engineering

in

Computer Science and Engineering

By

NAME: Sriharini Margapuri

ROLL NO: 1005-21-733065

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY COLLEGE OF ENGINEERING (A)
Osmania University, Hyderabad – 500 007 2022-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY COLLEGE OF ENGINEERING (A)
Osmania University, Hyderabad – 500 007

CERTIFICATE

This is to certify that **Sriharini Margapuri** bearing

Roll no: **1005-21-733065** studying B.E. V Semester has successfully completed

"**Database Management Systems Lab**" for the academic year 2023-24.

Internal Examiner **External Examiner**

TABLE OF CONTENTS

SUBQUERIES:

1. Display all employees who do not have any reportees.

```
SQL> --Q1

SQL> SELECT ENAME FROM EMP WHERE EMPNO NOT IN (SELECT MGR FROM EMP WHERE MGR
IS NOT NULL);


ENAME

----------

ADAMS

WARD

ALLEN

JAMES

SMITH

MILLER

MARTIN

TURNER


8 rows selected.
```

2. List employees who have at least 2 reporting.

```
SQL> --Q2

SQL> SELECT ENAME FROM EMP WHERE EMPNO IN (SELECT MGR FROM EMP WHERE MGR IS
NOT NULL GROUP BY MGR HAVING COUNT(MGR) >= 2);


ENAME

----------

KING

BLAKE

JONES
```

3. List the department names who have more than 5 employees.

```
SQL> --Q3

SQL> SELECT DNAME FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP GROUP BY
DEPTNO HAVING COUNT(DEPTNO) > 5);


DNAME

--------------

SALES
```

4. List the department names having at least 3 salesman.

```
SQL> --Q4

SQL> SELECT DNAME FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP WHERE JOB
= 'SALESMAN' GROUP BY DEPTNO HAVING COUNT(DEPTNO) >= 3);


DNAME

--------------

SALES
```

5. List the employees from research and accounting having at least 2 reporting.

```
SQL> --Q5

SQL> SELECT ENAME FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE
DNAME = 'RESEARCH' OR DNAME = 'ACCOUNTING') AND EMPNO IN (SELECT MGR FROM EMP
GROUP BY MGR HAVING COUNT(MGR) >=2);


ENAME

----------

KING

JONES
```

6. List the employees working in research department.

```
SQL> --Q6

SQL> SELECT ENAME FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE
DNAME = 'RESEARCH');


ENAME

----------

JONES

FORD

SMITH

SCOTT

ADAMS
```

7. List the employees who are located in New York and Chicago.

```
SQL> --Q7

SQL> SELECT ENAME FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE LOC
= 'NEW YORK' OR LOC = 'CHICAGO');


ENAME

----------

KING

BLAKE

CLARK

MARTIN

ALLEN

TURNER

JAMES

WARD

MILLER
```

```
9 rows selected.
```

8. Display the department names in which analysts are working.

```
SQL> --Q8

SQL> SELECT DNAME FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP WHERE JOB
= 'ANALYST');


DNAME

--------------

RESEARCH
```

9. Display employees who are reporting to Jones.

```
SQL> --Q9

SQL> SELECT ENAME FROM EMP WHERE MGR IN (SELECT EMPNO FROM EMP WHERE ENAME =
'JONES');


ENAME

----------

FORD

SCOTT
```

CURSORS:

1. **Write a program to accept a mgr and display who are working under that mgr.**

```
SQL> --Q1
SQL> DECLARE
  2  C_ENAME EMP.ENAME%TYPE;
  3  V_MGR EMP.MGR%TYPE := &MGR;
  4  CURSOR C_MANAGES IS SELECT ENAME FROM EMP WHERE MGR = V_MGR;
  5  BEGIN
  6  OPEN C_MANAGES;
  7  LOOP
  8  FETCH C_MANAGES INTO C_ENAME;
  9  EXIT WHEN C_MANAGES%NOTFOUND;
 10  DBMS_OUTPUT.PUT_LINE(C_ENAME);
 11  END LOOP;
 12  CLOSE C_MANAGES;
 13  END;
 14  /
Enter value for mgr: 7566
old   3: V_MGR EMP.MGR%TYPE := &MGR;
new   3: V_MGR EMP.MGR%TYPE := 7566;
FORD
SCOTT

PL/SQL procedure successfully completed.
```

2. Write a program to accept a year and display the emps belonging to that year.

```
SQL> --Q2

SQL> DECLARE

  2  C_ENAME EMP.ENAME%TYPE;

  3  V_YEAR NUMBER := &YEAR;

  4  CURSOR C_HIRED IS SELECT ENAME FROM EMP WHERE EXTRACT(YEAR FROM
HIREDATE) = V_YEAR;

  5  BEGIN

  6  OPEN C_HIRED;

  7  LOOP

  8  FETCH C_HIRED INTO C_ENAME;

  9  EXIT WHEN C_HIRED%NOTFOUND;

 10  DBMS_OUTPUT.PUT_LINE(C_ENAME);

 11  END LOOP;

 12  CLOSE C_HIRED;

 13  END;

 14  /

Enter value for year: 1981

old   3: V_YEAR NUMBER := &YEAR;

new   3: V_YEAR NUMBER := 1981;

KING

BLAKE

CLARK

JONES

MARTIN

ALLEN

TURNER

JAMES

WARD
```

```
FORD


PL/SQL procedure successfully completed.
```

3. Write a program to accept the grade and display emps belonging to that grade.

```
SQL> --Q3
SQL> DECLARE
  2  C_ENAME EMP.ENAME%TYPE;
  3  V_GRADE SALGRADE.GRADE%TYPE := &GRADE;
  4  CURSOR C_GRADE IS SELECT ENAME FROM EMP JOIN SALGRADE ON SAL BETWEEN
LOSAL AND HISAL WHERE GRADE = V_GRADE;
  5  BEGIN
  6  OPEN C_GRADE;
  7  LOOP
  8  FETCH C_GRADE INTO C_ENAME;
  9  EXIT WHEN C_GRADE%NOTFOUND;
 10  DBMS_OUTPUT.PUT_LINE(C_ENAME);
 11  END LOOP;
 12  CLOSE C_GRADE;
 13  END;
 14  /
Enter value for grade: 3
old   3: V_GRADE SALGRADE.GRADE%TYPE := &GRADE;
new   3: V_GRADE SALGRADE.GRADE%TYPE := 3;
ALLEN
TURNER


PL/SQL procedure successfully completed.
```

TRIGGERS:

1. Write a database trigger to halt the transaction of emp table if the deptno does not exist in the dept table.

```
SQL> --Q1
SQL> CREATE OR REPLACE TRIGGER T1
  2  BEFORE INSERT OR UPDATE ON EMP1
  3  FOR EACH ROW
  4  DECLARE
  5  V_DEPTNO DEPT.DEPTNO%TYPE;
  6  BEGIN
  7  SELECT DEPTNO INTO V_DEPTNO FROM DEPT WHERE DEPTNO = :NEW.DEPTNO;
  8  EXCEPTION
  9  WHEN NO_DATA_FOUND THEN
 10  RAISE_APPLICATION_ERROR(-20000, 'DEPARTMENT DOES NOT EXIST.');
 11  END;
 12  /

Trigger created.

SQL> UPDATE EMP1 SET DEPTNO = 50 WHERE ENAME = 'MILLER';
UPDATE EMP1 SET DEPTNO = 50 WHERE ENAME = 'MILLER'
       *
ERROR at line 1:
ORA-20000: DEPARTMENT DOES NOT EXIST.
ORA-06512: at "SYSTEM.T1", line 7
ORA-04088: error during execution of trigger 'SYSTEM.T1'
```

2. Write a database trigger to add Rs. 500 if the inserting salary is less than Rs. 1000.

```
SQL> --Q2

SQL> CREATE OR REPLACE TRIGGER T2

  2  BEFORE INSERT ON EMP1

  3  FOR EACH ROW

  4  BEGIN

  5  IF :NEW.SAL < 1000 THEN

  6  :NEW.SAL := :NEW.SAL + 500;

  7  END IF;

  8  END;

  9  /


Trigger created.


SQL> INSERT INTO EMP1(EMPNO, ENAME, SAL, DEPTNO) VALUES (1111, 'HARINI', 600,
10);


1 row created.


SQL> SELECT * FROM EMP1 WHERE EMPNO = 1111;


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      1111 HARINI                                          1100                    10
```

3. Write a database trigger display the message when the inserting hiredate is greater than system date.

```
SQL> --Q3

SQL> CREATE OR REPLACE TRIGGER T3

  2  BEFORE INSERT ON EMP1

  3  FOR EACH ROW

  4  BEGIN

  5  IF :NEW.HIREDATE > SYSDATE THEN

  6  RAISE_APPLICATION_ERROR(-20000, 'INVALID HIREDATE.');

  7  END IF;

  8  END;

  9  /


Trigger created.


SQL> INSERT INTO EMP1(EMPNO, ENAME, SAL, DEPTNO, HIREDATE) VALUES (1112,
'HARINI', 500, 10, '01-JAN-24');

INSERT INTO EMP1(EMPNO, ENAME, SAL, DEPTNO, HIREDATE) VALUES (1112,
'HARINI', 500, 10, '01-JAN-24')

            *

ERROR at line 1:

ORA-20000: INVALID HIREDATE.

ORA-06512: at "SYSTEM.T3", line 3

ORA-04088: error during execution of trigger 'SYSTEM.T3'
```

4. Write a database trigger to add Rs. 500 if the inserting salary is less than Rs. 1000.

```
SQL> --Q4

SQL> CREATE OR REPLACE TRIGGER T4

  2  BEFORE INSERT ON EMP1

  3  FOR EACH ROW

  4  BEGIN

  5  IF :NEW.SAL < 1000 THEN

  6  :NEW.SAL := :NEW.SAL + 500;

  7  END IF;

  8  END;

  9  /


Trigger created.


SQL> INSERT INTO EMP1(EMPNO, ENAME, SAL, DEPTNO) VALUES (1112, 'HARINI', 700,
10);


1 row created.


SQL> SELECT * FROM EMP1 WHERE EMPNO = 1112;


        EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM
DEPTNO

---------- ---------- --------- ---------- --------- ---------- ---------- ----------

      1112 HARINI                                          1200                    10
```

5. Write a database trigger to halt the transaction on Sunday on emp table.

```
SQL> --Q5

SQL> CREATE OR REPLACE TRIGGER T5

  2  BEFORE UPDATE ON EMP1

  3  FOR EACH ROW

  4  BEGIN

  5  IF TO_CHAR(SYSDATE, 'FMDAY') = 'SUNDAY' THEN

  6  RAISE_APPLICATION_ERROR(-20000, 'CANNOT COMMIT TRANSACTIONS ON
SUNDAYS');

  7  END IF;

  8  END;

  9  /


Trigger created.


SQL> UPDATE EMP1 SET SAL = 500 WHERE EMPNO = 1111;

UPDATE EMP1 SET SAL = 500 WHERE EMPNO = 1111

       *

ERROR at line 1:

ORA-20000: CANNOT COMMIT TRANSACTIONS ON SUNDAYS

ORA-06512: at "SYSTEM.T5", line 3

ORA-04088: error during execution of trigger 'SYSTEM.T5'
```

6. Write a procedure to accept deptno as input and print the details of emps along with grade?

```
SQL> --Q6
SQL> CREATE OR REPLACE PROCEDURE P6 (E NUMBER) IS
  2  V_EMPNO EMP.EMPNO%TYPE;
  3  V_ENAME EMP.ENAME%TYPE;
  4  V_JOB EMP.JOB%TYPE;
  5  V_MGR EMP.MGR%TYPE;
  6  V_HIREDATE EMP.HIREDATE%TYPE;
  7  V_SAL EMP.SAL%TYPE;
  8  V_COMM EMP.COMM%TYPE;
  9  V_DEPTNO EMP.DEPTNO%TYPE;
 10  BEGIN
 11  SELECT EMPNO INTO V_EMPNO FROM EMP WHERE EMPNO = E;
 12  SELECT ENAME INTO V_ENAME FROM EMP WHERE EMPNO = E;
 13  SELECT JOB INTO V_JOB FROM EMP WHERE EMPNO = E;
 14  SELECT HIREDATE INTO V_HIREDATE FROM EMP WHERE EMPNO = E;
 15  SELECT SAL INTO V_SAL FROM EMP WHERE EMPNO = E;
 16  SELECT COMM INTO V_COMM FROM EMP WHERE EMPNO = E;
 17  SELECT DEPTNO INTO V_DEPTNO FROM EMP WHERE EMPNO = E;
 18  DBMS_OUTPUT.PUT_LINE('EMPNO: ' || V_EMPNO);
 19  DBMS_OUTPUT.PUT_LINE('ENAME: ' || V_ENAME);
 20  DBMS_OUTPUT.PUT_LINE('JOB: ' || V_JOB);
 21  DBMS_OUTPUT.PUT_LINE('HIREDATE: ' || V_HIREDATE);
 22  DBMS_OUTPUT.PUT_LINE('SAL: ' || V_SAL);
 23  DBMS_OUTPUT.PUT_LINE('COMM: ' || V_COMM);
 24  DBMS_OUTPUT.PUT_LINE('DEPTNO: ' || V_DEPTNO);
 25  END;
 26  /
```

```
Procedure created.


SQL> EXECUTE P6(7566);
EMPNO: 7566
ENAME: JONES
JOB: MANAGER
HIREDATE: 02-APR-81
SAL: 2975
COMM:
DEPTNO: 20

PL/SQL procedure successfully completed.
```

PROCEDURES:

1. Write a program to calculate the area of a triangle by accepting the 3 sides.

```
SQL> --Q1
SQL> CREATE OR REPLACE PROCEDURE P1 (S1 NUMBER, S2 NUMBER, S3 NUMBER) IS
  2  AREA NUMBER;
  3  S NUMBER;
  4  BEGIN
  5  S := (S1 + S2 + S3)/2;
  6  AREA := SQRT(S*(S-S1)*(S-S2)*(S-S3));
  7  DBMS_OUTPUT.PUT_LINE('AREA: ' || AREA);
  8  END;
  9  /

Procedure created.

SQL> EXECUTE P1(3, 4, 5);
AREA: 6

PL/SQL procedure successfully completed.
```

2. Write a program to accept the temp in Centigrade and convert it into Fahrenheit. (C=(F-32)/1.8)

```
SQL> --Q2
SQL> CREATE OR REPLACE PROCEDURE P2 (C FLOAT) IS
  2  F FLOAT;
  3  BEGIN
  4  F := (1.8*C)+32;
  5  DBMS_OUTPUT.PUT_LINE('FAHRENHEIT: ' || F);
  6  END;
```

```
  7  /

Procedure created.

SQL> EXECUTE P2(32);
FAHRENHEIT: 89.6

PL/SQL procedure successfully completed.
```

3. Write a program to accept a string and check whether it is a palindrome or not.

```
SQL> --Q3
SQL> CREATE OR REPLACE PROCEDURE P3 (S VARCHAR2) IS
  2  B NUMBER;
  3  C VARCHAR2(1);
  4  D VARCHAR2(50);
  5  BEGIN
  6  B := LENGTH(RTRIM(S));
  7  WHILE B>= 1 LOOP
  8  C := SUBSTR(S, B, 1);
  9  D := CONCAT(D, C);
 10  B := B-1;
 11  END LOOP;
 12  IF (S = D) THEN
 13  DBMS_OUTPUT.PUT_LINE(S || ' IS A PALINDROME.');
 14  ELSE
 15  DBMS_OUTPUT.PUT_LINE(S || ' IS NOT A PALINDROME.');
 16  END IF;
 17  END;
 18  /
```

```
Procedure created.


SQL> EXECUTE P3('RACECAR');
RACECAR IS A PALINDROME.


PL/SQL procedure successfully completed.
```

4. Write a program to accept a year and display all Sundays along with the date.

```
SQL> --Q4
SQL> CREATE OR REPLACE PROCEDURE P4 (Y NUMBER) IS
  2  B NUMBER := SUBSTR(Y, 3, 2);
  3  SD DATE := CONCAT('01-JAN-', B);
  4  ED DATE := CONCAT('31-DEC-', B);
  5  BEGIN
  6  WHILE(SD <= ED) LOOP
  7  IF(TO_CHAR(SD, 'FMDAY') = 'SUNDAY') THEN
  8  DBMS_OUTPUT.PUT_LINE(SD || ' SUNDAY');
  9  END IF;
 10  SD := SD + 1;
 11  END LOOP;
 12  END;
 13  /

Procedure created.


SQL> EXECUTE P4(2023);
01-JAN-23 SUNDAY
08-JAN-23 SUNDAY
```

15-JAN-23 SUNDAY

22-JAN-23 SUNDAY

29-JAN-23 SUNDAY

05-FEB-23 SUNDAY

12-FEB-23 SUNDAY

19-FEB-23 SUNDAY

26-FEB-23 SUNDAY

05-MAR-23 SUNDAY

12-MAR-23 SUNDAY

19-MAR-23 SUNDAY

26-MAR-23 SUNDAY

02-APR-23 SUNDAY

09-APR-23 SUNDAY

16-APR-23 SUNDAY

23-APR-23 SUNDAY

30-APR-23 SUNDAY

07-MAY-23 SUNDAY

14-MAY-23 SUNDAY

21-MAY-23 SUNDAY

28-MAY-23 SUNDAY

04-JUN-23 SUNDAY

11-JUN-23 SUNDAY

18-JUN-23 SUNDAY

25-JUN-23 SUNDAY

02-JUL-23 SUNDAY

09-JUL-23 SUNDAY

16-JUL-23 SUNDAY

23-JUL-23 SUNDAY

30-JUL-23 SUNDAY

06-AUG-23 SUNDAY

```
13-AUG-23 SUNDAY
20-AUG-23 SUNDAY
27-AUG-23 SUNDAY
03-SEP-23 SUNDAY
10-SEP-23 SUNDAY
17-SEP-23 SUNDAY
24-SEP-23 SUNDAY
01-OCT-23 SUNDAY
08-OCT-23 SUNDAY
15-OCT-23 SUNDAY
22-OCT-23 SUNDAY
29-OCT-23 SUNDAY
05-NOV-23 SUNDAY
12-NOV-23 SUNDAY
19-NOV-23 SUNDAY
26-NOV-23 SUNDAY
03-DEC-23 SUNDAY
10-DEC-23 SUNDAY
17-DEC-23 SUNDAY
24-DEC-23 SUNDAY
31-DEC-23 SUNDAY

PL/SQL procedure successfully completed.
```

SQL:

SET 1:

1. Display all employees and corresponding managers.

```
SQL> --Q1
SQL> SELECT E1.ENAME, E2.ENAME FROM EMP E1 JOIN EMP E2 ON E1.MGR = E2.EMPNO;

ENAME      ENAME
---------- ----------
BLAKE      KING
CLARK      KING
JONES      KING
MARTIN     BLAKE
ALLEN      BLAKE
TURNER     BLAKE
JAMES      BLAKE
WARD       BLAKE
MILLER     CLARK
FORD       JONES
SCOTT      JONES
SMITH      FORD
ADAMS      SCOTT

13 rows selected.
```

2. Display all the departments where employee salary greater than average salary of that department.

```
SQL> --Q2

SQL> SELECT DISTINCT DEPT.DNAME FROM DEPT JOIN EMP ON DEPT.DEPTNO =
EMP.DEPTNO WHERE DEPT.DEPTNO IN (SELECT DEPTNO FROM EMP E1 WHERE SAL >
(SELECT AVG(SAL) FROM EMP E2 WHERE E2.DEPTNO = E1.DEPTNO));


DNAME

--------------

ACCOUNTING

SALES

RESEARCH
```

3. Display all employees where ename starts with J and ends with S.

```
SQL> --Q3

SQL> SELECT ENAME FROM EMP WHERE ENAME LIKE 'J%S';


ENAME

----------

JONES

JAMES
```

4. Display employees where length of ename is 5.

```
SQL> --Q4

SQL> SELECT ENAME FROM EMP WHERE LENGTH(ENAME) = 5;


ENAME

----------

BLAKE

CLARK

JONES
```

```
ALLEN

JAMES

SMITH

SCOTT

ADAMS


8 rows selected.
```

5. Display all employees whose salary is greater than the manager's salary.

```
SQL> --Q5

SQL> SELECT E1.ENAME FROM EMP E1 JOIN EMP E2 ON E1.MGR = E2.EMPNO WHERE
E1.SAL > E2.SAL;


ENAME

----------

FORD

SCOTT
```

6. Display nth highest and nth lowest salary in emp table.

```
SQL> --Q6

SQL> SELECT MAX(SAL) FROM EMP WHERE SAL IN (SELECT SAL FROM EMP ORDER BY SAL
DESC OFFSET &N-1 ROWS) UNION SELECT MIN(SAL) FROM EMP WHERE SAL IN (SELECT
SAL FROM EMP ORDER BY SAL OFFSET &N-1 ROWS);

Enter value for n: 2

Enter value for n: 2

old   1: SELECT MAX(SAL) FROM EMP WHERE SAL IN (SELECT SAL FROM EMP ORDER BY
SAL DESC OFFSET &N-1 ROWS) UNION SELECT MIN(SAL) FROM EMP WHERE SAL IN
(SELECT SAL FROM EMP ORDER BY SAL OFFSET &N-1 ROWS)

new   1: SELECT MAX(SAL) FROM EMP WHERE SAL IN (SELECT SAL FROM EMP ORDER BY
SAL DESC OFFSET 2-1 ROWS) UNION SELECT MIN(SAL) FROM EMP WHERE SAL IN (SELECT
SAL FROM EMP ORDER BY SAL OFFSET 2-1 ROWS)
```

```
  MAX(SAL)
----------
      3000
       950
```

SET 2:

1. Increase commission $10 for employees who are located in New York.

```
SQL> --Q1

SQL> UPDATE EMP1 SET COMM = COMM + 10 WHERE DEPTNO IN (SELECT DEPTNO FROM
DEPT WHERE LOC = 'NEW YORK');


3 rows updated.


SQL> SELECT * FROM EMP1 WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE LOC =
'NEW YORK');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

2. Reduce the commission amount from employee salary for each employee who joined after Allen joined.

```
SQL> --Q2

SQL> UPDATE EMP1 SET COMM = COMM - 5 WHERE HIREDATE > (SELECT HIREDATE FROM
EMP WHERE ENAME = 'ALLEN');


12 rows updated.


SQL> SELECT * FROM EMP WHERE HIREDATE > (SELECT HIREDATE FROM EMP WHERE ENAME
= 'ALLEN');
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
```

```
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7788 SCOTT      ANALYST         7566 09-DEC-82       3000                    20
      7876 ADAMS      CLERK           7788 12-JAN-83       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

12 rows selected.
```

3. Increase 1% salary for employee who are making lowest salary in dept 10.

```
SQL> --Q3

SQL> UPDATE EMP1 SET SAL = SAL * 1.01 WHERE SAL IN (SELECT MIN(SAL) FROM EMP
WHERE DEPTNO = 10) AND DEPTNO = 10;


1 row updated.


SQL> SELECT * FROM EMP WHERE SAL IN (SELECT MIN(SAL) FROM EMP WHERE DEPTNO =
10) AND DEPTNO = 10;


     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

4. Increase $100 for employees who is making more than average salary of their department.

```
SQL> --Q4

SQL> UPDATE EMP1 E1 SET E1.SAL = E1.SAL + 100 WHERE E1.SAL IN (SELECT
MAX(E2.SAL) FROM EMP1 E2 WHERE E2.DEPTNO = E1.DEPTNO);
```

```
4 rows updated.

SQL> SELECT * FROM EMP1 E1 WHERE E1.SAL IN (SELECT MAX(E2.SAL) FROM EMP1 E2
WHERE E2.DEPTNO = E1.DEPTNO);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5100                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3100                    20
      7788 SCOTT      ANALYST         7566 09-DEC-82       3100                    20
```

5. Increase $250 commission for Blake’s team.

```
SQL> --Q5

SQL> UPDATE EMP1 SET COMM = COMM + 250 WHERE MGR IN (SELECT EMPNO FROM EMP1
WHERE ENAME = 'BLAKE');

5 rows updated.

SQL> SELECT * FROM EMP1 WHERE MGR IN (SELECT EMPNO FROM EMP1 WHERE ENAME =
'BLAKE');
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1650         30
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        550         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500        250         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        750         30
```

SET 3:

1. Display all employees who joined in year 1981.

```
SQL> --Q1
SQL> SELECT ENAME FROM EMP WHERE EXTRACT(YEAR FROM HIREDATE) = 1981;

ENAME
----------
KING
BLAKE
CLARK
JONES
MARTIN
ALLEN
TURNER
JAMES
WARD
FORD

10 rows selected.
```

2. Display comm in emp table. Display zero in place of null.

```
SQL> --Q2
SQL> SELECT COALESCE(COMM, 0) FROM EMP;

COALESCE(COMM,0)
----------------
               0
               0
               0
```

```
              0
           1400
            300
              0
              0
            500
              0
              0

COALESCE(COMM,0)
----------------
              0
              0
              0

14 rows selected.
```

3. Display all employees with how many years they have been serving the company.

```
SQL> --Q3
SQL> SELECT ENAME, ROUND((SYSDATE-HIREDATE)/365, 0) FROM EMP;

ENAME      ROUND((SYSDATE-HIREDATE)/365,0)
---------- -------------------------------
KING                                    42
BLAKE                                   43
CLARK                                   43
JONES                                   43
MARTIN                                  42
ALLEN                                   43
```

```
TURNER                                    42
JAMES                                     42
WARD                                      43
FORD                                      42
SMITH                                     43
SCOTT                                     41
ADAMS                                     41
MILLER                                    42

14 rows selected.
```

4. Display all employees where employees hired before 01-Jan-1981.

```
SQL> --Q4
SQL> SELECT ENAME FROM EMP WHERE HIREDATE < '01-JAN-81';

ENAME
----------
SMITH
```

5. Display the record in emp table where mgr is 7698, 7566 and sal should be greater than 1500.

```
SQL> --Q5
SQL> SELECT * FROM EMP WHERE (MGR = 7698 OR MGR = 7566) AND SAL > 1500;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7788 SCOTT      ANALYST         7566 09-DEC-82       3000                    20
```

SET 4:

1. Display all the departments where department does not have an employee.

```
SQL> --Q1

SQL> SELECT DNAME FROM DEPT LEFT OUTER JOIN EMP ON DEPT.DEPTNO = EMP.DEPTNO
WHERE DEPT.DEPTNO NOT IN (SELECT DEPTNO FROM EMP GROUP BY DEPTNO HAVING
COUNT(*) != 0);


DNAME

--------------

OPERATIONS
```

2. Display all the departments where department has at least one employee.

```
SQL> --Q2

SQL> SELECT DISTINCT DNAME FROM DEPT LEFT OUTER JOIN EMP ON DEPT.DEPTNO =
EMP.DEPTNO WHERE DEPT.DEPTNO IN (SELECT DEPTNO FROM EMP GROUP BY DEPTNO
HAVING COUNT(*) >= 1);


DNAME

--------------

ACCOUNTING

SALES

RESEARCH
```

3. Display all the employees those who are not managers.

```
SQL> --Q3

SQL> SELECT ENAME FROM EMP WHERE EMPNO NOT IN (SELECT MGR FROM EMP WHERE MGR
IS NOT NULL);


ENAME

----------

ADAMS
```

```
WARD

ALLEN

JAMES

SMITH

MILLER

MARTIN

TURNER


8 rows selected.
```

4. Display ename, deptno from emp table with format of {ename} belongs to {deptno}.

```
SQL> --Q4

SQL> SELECT ENAME || ' BELONGS TO ' || DEPTNO FROM EMP;


ENAME||'BELONGSTO'||DEPTNO

----------------------------------------------------------------

KING BELONGS TO 10

BLAKE BELONGS TO 30

CLARK BELONGS TO 10

JONES BELONGS TO 20

MARTIN BELONGS TO 30

ALLEN BELONGS TO 30

TURNER BELONGS TO 30

JAMES BELONGS TO 30

WARD BELONGS TO 30

FORD BELONGS TO 20

SMITH BELONGS TO 20

SCOTT BELONGS TO 20

ADAMS BELONGS TO 20
```

```
MILLER BELONGS TO 10


14 rows selected.
```

5. Display total number of employees hired for 1980, 1981, 1982. The output should be in one record.

```
SQL> --Q5

SQL> SELECT COUNT(*) FROM EMP WHERE HIREDATE BETWEEN '01-JAN-1980' AND '31-
DEC-1982';


  COUNT(*)

----------

        13
```

SET 5:

1. Write a program for multiplication table.

```
SQL> --Q1
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  I NUMBER := 1;
  3  N NUMBER := &N;
  4  BEGIN
  5  WHILE(I<=10) LOOP
  6  DBMS_OUTPUT.PUT_LINE(N || ' x ' || I || ' = ' || N*I);
  7  I := I + 1;
  8  END LOOP;
  9  END;
 10  /
Enter value for n: 3
old   3: N NUMBER := &N;
new   3: N NUMBER := 3;
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

PL/SQL procedure successfully completed.
```

2. Write a program to accept a number and find how many digits it contains.

```
SQL> --Q2
SQL> DECLARE
  2  N NUMBER := &N;
  3  C NUMBER;
  4  BEGIN
  5  C:= LENGTH(N);
  6  DBMS_OUTPUT.PUT_LINE('NUMBER OF DIGITS IN ' || N || ' IS ' || C);
  7  END;
  8  /
Enter value for n: 2142
old   2: N NUMBER := &N;
new   2: N NUMBER := 2142;
NUMBER OF DIGITS IN 2142 IS 4
PL/SQL procedure successfully completed.
```

3. WAP to accept the empno and display ename, sal, hiredate and calculate ta, da, hra, lic, gross, exp and print all the emp details. Ta is 30% of sal, da is 20% of sal, hra is 15% of sal, lic is 5% of sal.

```
SQL> --Q3
SQL> SELECT ENAME, SAL, HIREDATE, (SAL*0.30) AS TA, (SAL*0.20) AS DA,
(SAL*0.15) AS HRA, (SAL*0.05) AS LIC FROM EMP WHERE EMPNO = &EMPNO;
Enter value for empno: 7566
old   1: SELECT ENAME, SAL, HIREDATE, (SAL*0.30) AS TA, (SAL*0.20) AS DA,
(SAL*0.15) AS HRA, (SAL*0.05) AS LIC FROM EMP WHERE EMPNO = &EMPNO
new   1: SELECT ENAME, SAL, HIREDATE, (SAL*0.30) AS TA, (SAL*0.20) AS DA,
(SAL*0.15) AS HRA, (SAL*0.05) AS LIC FROM EMP WHERE EMPNO = 7566

ENAME             SAL HIREDATE          TA         DA        HRA        LIC
---------- ---------- --------- ---------- ---------- ---------- ----------
JONES            2975 02-APR-81      892.5        595     446.25     148.75
```

4. WAP to accept a empno and display empno, based on experience calculate the bonus and store it into the bonus table.

a. If exp < 5 years then bonus is 1 month salary.
b. If exp between 5 and 9 years then bonus is 20% of annual salary.
c. If exp > 9 years then bonus is 1 month salary plus 25% of annual salary.

```
SQL> --Q4
SQL> DECLARE
  2  V EMP.EMPNO%TYPE := &EMPNO;
  3  S EMP.SAL%TYPE;
  4  H EMP.HIREDATE%TYPE;
  5  B FLOAT;
  6  BEGIN
  7  SELECT HIREDATE INTO H FROM EMP WHERE EMPNO = V;
  8  SELECT SAL INTO S FROM EMP WHERE EMPNO = V;
  9  IF((SYSDATE-H)/365 < 5) THEN
 10  B := S;
 11  ELSIF ((SYSDATE-H)/365 > 5 AND (SYSDATE-H) < 9) THEN
 12  B := 0.20*(12*S);
 13  ELSIF ((SYSDATE-H) > 9) THEN
 14  B := S + (0.25*(12*S));
 15  END IF;
 16  DBMS_OUTPUT.PUT_LINE(V || ' BONUS: ' || B);
 17  END;
 18  /
Enter value for empno: 7566
old   2: V EMP.EMPNO%TYPE := &EMPNO;
new   2: V EMP.EMPNO%TYPE := 7566;
7566 BONUS: 11900

PL/SQL procedure successfully completed.
```

SET 6:

1. **Display ename, job, dname, deptno for each employee by using inline view.**

```
SQL> --Q1

SQL> SELECT E.ENAME, E.JOB, D.DNAME, E.DEPTNO FROM EMP E, (SELECT DNAME,
DEPTNO FROM DEPT) D WHERE E.DEPTNO = D.DEPTNO;


ENAME      JOB       DNAME              DEPTNO
---------- --------- -------------- ----------
KING       PRESIDENT ACCOUNTING             10
BLAKE      MANAGER   SALES                  30
CLARK      MANAGER   ACCOUNTING             10
JONES      MANAGER   RESEARCH               20
MARTIN     SALESMAN  SALES                  30
ALLEN      SALESMAN  SALES                  30
TURNER     SALESMAN  SALES                  30
JAMES      CLERK     SALES                  30
WARD       SALESMAN  SALES                  30
FORD       ANALYST   RESEARCH               20
SMITH      CLERK     RESEARCH               20
SCOTT      ANALYST   RESEARCH               20
ADAMS      CLERK     RESEARCH               20
MILLER     CLERK     ACCOUNTING             10

14 rows selected.
```

2. Find the oldest and latest hiredates.

```
SQL> --Q2

SQL> SELECT MAX(HIREDATE), MIN(HIREDATE) FROM EMP;


MAX(HIRED MIN(HIRED
--------- ---------
12-JAN-83 17-DEC-80
```

3. Write a query to supply the alias to table and retrieve the data from emp, dept.

```
SQL> --Q3

SQL> SELECT E.*, D.* FROM EMP E JOIN DEPT D ON E.DEPTNO = D.DEPTNO;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
DEPTNO DNAME          LOC
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
---------- -------------- -------------

      7839 KING       PRESIDENT            17-NOV-81       5000                    10
10 ACCOUNTING     NEW YORK

      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
30 SALES          CHICAGO

      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
10 ACCOUNTING     NEW YORK

      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
20 RESEARCH       DALLAS

      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
30 SALES          CHICAGO

      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
30 SALES          CHICAGO

      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
30 SALES          CHICAGO

      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
30 SALES          CHICAGO

      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
30 SALES          CHICAGO

      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
20 RESEARCH       DALLAS
```

```
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
20 RESEARCH       DALLAS

      7788 SCOTT      ANALYST         7566 09-DEC-82       3000                    20
20 RESEARCH       DALLAS

      7876 ADAMS      CLERK           7788 12-JAN-83       1100                    20
20 RESEARCH       DALLAS

      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
10 ACCOUNTING     NEW YORK


14 rows selected.
```

4. Select empno, ename, job, deptno from emp using self join.

```
SQL> --Q4

SQL> SELECT E1.EMPNO, E1.ENAME, E2.JOB, E2.DEPTNO FROM EMP E1 JOIN EMP E2 ON
E1.EMPNO = E2.EMPNO;


     EMPNO ENAME      JOB           DEPTNO
---------- ---------- --------- ----------
      7839 KING       PRESIDENT         10
      7698 BLAKE      MANAGER           30
      7782 CLARK      MANAGER           10
      7566 JONES      MANAGER           20
      7654 MARTIN     SALESMAN          30
      7499 ALLEN      SALESMAN          30
      7844 TURNER     SALESMAN          30
      7900 JAMES      CLERK             30
      7521 WARD       SALESMAN          30
      7902 FORD       ANALYST           20
      7369 SMITH      CLERK             20
      7788 SCOTT      ANALYST           20
      7876 ADAMS      CLERK             20
      7934 MILLER     CLERK             10
```

14 rows selected.

5. Select empno, ename, mgr, comm from emp using self join.

```
SQL> --Q5

SQL> SELECT E1.EMPNO, E1.ENAME, E2.MGR, E2.COMM FROM EMP E1 JOIN EMP E2 ON
E1.EMPNO = E2.EMPNO;


     EMPNO ENAME              MGR       COMM
---------- ---------- ---------- ----------
      7839 KING
      7698 BLAKE            7839
      7782 CLARK            7839
      7566 JONES            7839
      7654 MARTIN           7698       1400
      7499 ALLEN            7698        300
      7844 TURNER           7698          0
      7900 JAMES            7698
      7521 WARD             7698        500
      7902 FORD             7566
      7369 SMITH            7902
      7788 SCOTT            7566
      7876 ADAMS            7788
      7934 MILLER           7782


14 rows selected.
```

SET 7:

1. Display nth highest and nth lowest salary in emp table.

```
SQL> --Q1

SQL> SELECT MAX(SAL) FROM EMP WHERE SAL IN (SELECT SAL FROM EMP ORDER BY SAL
DESC OFFSET &N-1 ROWS) UNION SELECT MIN(SAL) FROM EMP WHERE SAL IN (SELECT
SAL FROM EMP ORDER BY SAL OFFSET &N-1 ROWS);

Enter value for n: 3

Enter value for n: 3

old   1: SELECT MAX(SAL) FROM EMP WHERE SAL IN (SELECT SAL FROM EMP ORDER BY
SAL DESC OFFSET &N-1 ROWS) UNION SELECT MIN(SAL) FROM EMP WHERE SAL IN
(SELECT SAL FROM EMP ORDER BY SAL OFFSET &N-1 ROWS)

new   1: SELECT MAX(SAL) FROM EMP WHERE SAL IN (SELECT SAL FROM EMP ORDER BY
SAL DESC OFFSET 3-1 ROWS) UNION SELECT MIN(SAL) FROM EMP WHERE SAL IN (SELECT
SAL FROM EMP ORDER BY SAL OFFSET 3-1 ROWS)


  MAX(SAL)

----------

      3000

      1100
```

2. Display all the departments where department has 3 employees.

```
SQL> --Q2

SQL> SELECT DNAME FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP GROUP BY
DEPTNO HAVING COUNT(DEPTNO) = 3);


DNAME

--------------

ACCOUNTING
```

3. Display sum of salary for each department. The output should be in one record.

```
SQL> --Q3

SQL> SELECT SUM(SAL) FROM EMP GROUP BY DEPTNO;


  SUM(SAL)

----------

      8750

      9400

     10875
```

4. Display all the departments where department does have at least one employee.

```
SQL> --Q4

SQL> SELECT DNAME FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP GROUP BY
DEPTNO HAVING COUNT(DEPTNO) >= 1);


DNAME

--------------

ACCOUNTING

RESEARCH

SALES
```

SET 8:

1. Display all the records in emp table.

```
SQL> --Q1
SQL> SELECT * FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```
14 rows selected.
```

2. Display all the records in emp table where employee belongs to deptno 10.

```
SQL> --Q2
SQL> SELECT * FROM EMP WHERE DEPTNO = 10;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10

```
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

3. Display all the records in emp table where employee does not belong to deptno 30.

```
SQL> --Q3
SQL> SELECT * FROM EMP WHERE DEPTNO != 30;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7788 SCOTT      ANALYST         7566 09-DEC-82       3000                    20
      7876 ADAMS      CLERK           7788 12-JAN-83       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

8 rows selected.
```

4. Display total number of records in emp table.

```
SQL> --Q4
SQL> SELECT COUNT(*) FROM EMP;

  COUNT(*)
----------
        14
```

5. Display emp table with salary descending order.

```
SQL> --Q5
SQL> SELECT * FROM EMP ORDER BY SAL DESC;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7934	MILLER	CLERK	7782	23-JAN-82	1300		10
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7369	SMITH	CLERK	7902	17-DEC-80	800		20

```
14 rows selected.
```

6. Display first five records in employee table.

```
SQL> --Q6
SQL> SELECT * FROM EMP FETCH FIRST 5 ROWS ONLY;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
```