Name: Soumyadeep Ganguly

Roll. No: 24MDT0082

Course: M.Sc. in Data Science

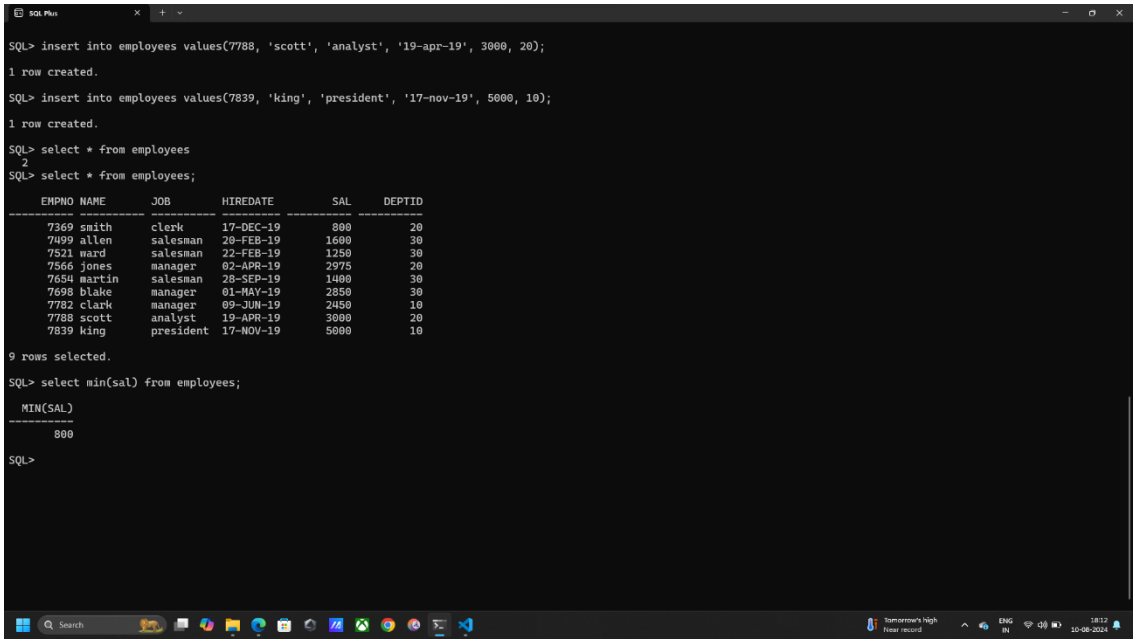
Assessment 2

PMDS506P Database Management systems.

Q1. Create a table EMPLOYEES with the attributes EMPNO, NAME, JOB, HIREDATE, SAL, DEPTID and populate it with the following data.

1. **Find the minimum salary being given by the company.**

SELECT MIN(SAL) FROM EMPLOYEES;



```
SQL> insert into employees values(7788, 'scott', 'analyst', '19-apr-19', 3000, 20);

1 row created.

SQL> insert into employees values(7839, 'king', 'president', '17-nov-19', 5000, 10);

1 row created.

SQL> select * from employees
  2
SQL> select * from employees;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- ---------- ---------- ----------
      7369 smith      clerk      17-DEC-19        800         20
      7499 allen      salesman   20-FEB-19       1600         30
      7521 ward       salesman   22-FEB-19       1250         30
      7566 jones      manager    02-APR-19       2975         20
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30
      7782 clark      manager    09-JUN-19       2450         10
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10

9 rows selected.

SQL> select min(sal) from employees;

  MIN(SAL)
----------
       800

SQL>
```

2. **List all employees with salary more than 1000 and who belong to department with id 30.**

SELECT * FROM EMPLOYEES WHERE SAL>1000 AND DEPTID = 30;

```
SQL Plus
  2
SQL> select * from employees;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7369 smith      clerk      17-DEC-19        800         20
      7499 allen      salesman   20-FEB-19       1600         30
      7521 ward       salesman   22-FEB-19       1250         30
      7566 jones      manager    02-APR-19       2975         20
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30
      7782 clark      manager    09-JUN-19       2450         10
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10

9 rows selected.

SQL> select min(sal) from employees;

  MIN(SAL)
----------
       800

SQL> select * from employees where sal>1000 and deptid = 30;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7499 allen      salesman   20-FEB-19       1600         30
      7521 ward       salesman   22-FEB-19       1250         30
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30

SQL>
```

3. **Display the table entries with ascending order of their salary.**

SELECT * FROM EMPLOYEES ORDER BY SAL ASC;

```
SQL Plus
SQL> insert into employees values(7839, 'king', 'president', '17-nov-19', 5000, 10);

1 row created.

SQL> SELECT * FROM EMPLOYEES;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7369 smith      clerk      17-DEC-19        800         20
      7499 allen      salesman   20-FEB-19       1600         30
      7521 ward       salesman   22-FEB-19       1250         30
      7566 jones      manager    02-APR-19       2975         20
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30
      7782 clark      manager    09-JUN-19       2450         10
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10

9 rows selected.

SQL> SELECT * FROM EMPLOYEES ORDER BY SAL ASC;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7369 smith      clerk      17-DEC-19        800         20
      7521 ward       salesman   22-FEB-19       1250         30
      7654 martin     salesman   28-SEP-19       1400         30
      7499 allen      salesman   20-FEB-19       1600         30
      7782 clark      manager    09-JUN-19       2450         10
      7698 blake      manager    01-MAY-19       2850         30
      7566 jones      manager    02-APR-19       2975         20
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10

9 rows selected.

SQL>
```

4. **Display all entries with salary > 2000 with the entries ordered in descending order with respect to the department ids.**

   SELECT * FROM EMPLOYEES WHERE SAL > 2000 ORDER BY DEPTID DESC;

```
SQL> SELECT * FROM EMPLOYEES ORDER BY DEPTID WHERE SAL > 2000;
SELECT * FROM EMPLOYEES ORDER BY DEPTID WHERE SAL > 2000
                                           *
ERROR at line 1:
ORA-00933: SQL command not properly ended


SQL> SELECT * FROM EMPLOYEES ORDER BY DEPTID DESC WHERE SAL > 2000;
SELECT * FROM EMPLOYEES ORDER BY DEPTID DESC WHERE SAL > 2000
                                              *
ERROR at line 1:
ORA-00933: SQL command not properly ended


SQL> SELECT * FROM EMPLOYEES WHERE SAL > 2000 ORDER BY DEPTID DESC;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7698 blake      manager    01-MAY-19       2850         30
      7788 scott      analyst    19-APR-19       3000         20
      7566 jones      manager    02-APR-19       2975         20
      7839 king       president  17-NOV-19       5000         10
      7782 clark      manager    09-JUN-19       2450         10

SQL>
```

5. **List the names of the employee being given the maximum salary.**

   SELECT * FROM EMPLOYEES
   WHERE SAL = (SELECT MAX(SAL) FROM EMPLOYEES);

```
                                        *
ERROR at line 1:
ORA-00934: group function is not allowed here


SQL> SELECT *, MAX(SAL) FROM EMPLOYEES;
SELECT *, MAX(SAL) FROM EMPLOYEES
        *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> SELECT EMPNO, NAME, JOB, MAX(SAL) FROM EMPLOYEES;
SELECT EMPNO, NAME, JOB, MAX(SAL) FROM EMPLOYEES
       *
ERROR at line 1:
ORA-00937: not a single-group group function


SQL> SELECT * FROM EMPLOYEES WHERE SAL = (SELECT MAX(SAL) FROM EMPLOYEES);

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7839 king       president  17-NOV-19       5000         10

SQL>
```

6. **Find the average salary being given by the company.**

SELECT AVG(SAL) FROM EMPLOYEES;

```
SQL Plus
SELECT *, MAX(SAL) FROM EMPLOYEES
        *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> SELECT EMPNO, NAME, JOB, MAX(SAL) FROM EMPLOYEES;
SELECT EMPNO, NAME, JOB, MAX(SAL) FROM EMPLOYEES
       *
ERROR at line 1:
ORA-00937: not a single-group group function


SQL> SELECT * FROM EMPLOYEES WHERE SAL = (SELECT MAX(SAL) FROM EMPLOYEES);

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7839 king       president  17-NOV-19       5000         10

SQL> SELECT AVG(SAL) FROM EMPLOYEES;

  AVG(SAL)
----------
2369.44444

SQL>
```

7. **Find the total count of all the employees.**

SELECT COUNT(EMPNO) FROM EMPLOYEES;

```
SQL Plus
ORA-00937: not a single-group group function


SQL> SELECT * FROM EMPLOYEES WHERE SAL = (SELECT MAX(SAL) FROM EMPLOYEES);

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7839 king       president  17-NOV-19       5000         10

SQL> SELECT AVG(SAL) FROM EMPLOYEES;

  AVG(SAL)
----------
2369.44444

SQL> SELECT COUNT(EMPNO) FROM EMPLOYEES;

COUNT(EMPNO)
------------
           9

SQL>
```

8. **Find the total salary of all the employees.**

SELECT SUM(SAL) FROM EMPLOYEES;

```
---------- ---------- ---------- --------- ---------- ----------
      7839 king       president  17-NOV-19       5000         10

SQL> SELECT AVG(SAL) FROM EMPLOYEES;

  AVG(SAL)
----------
2369.44444

SQL> SELECT COUNT(EMPNO) FROM EMPLOYEES;

COUNT(EMPNO)
------------
           9

SQL> SELECT SUM(SAL) FROM EMPLOYEES;

  SUM(SAL)
----------
     21325

SQL>
```

9. **Find the variance of the salary of the employees and display it.**

SELECT VARIANCE(SAL) FROM EMPLOYEES;

```
SQL> SELECT VAR(SAL) FROM EMPLOYEES;
SELECT VAR(SAL) FROM EMPLOYEES
       *
ERROR at line 1:
ORA-00904: "VAR": invalid identifier


SQL> SELECT VARIENCE(SAL) FROM EMPLOYEES;
SELECT VARIENCE(SAL) FROM EMPLOYEES
       *
ERROR at line 1:
ORA-00904: "VARIENCE": invalid identifier


SQL> SELECT VARIANCE(SAL) FROM EMPLOYEES;

VARIANCE(SAL)
-------------
   1646215.28

SQL>
```

10. **Find the total salary of all the employees of each department.**

SELECT DEPTID, SUM(SAL) FROM EMPLOYEES GROUP BY DEPTID;

```
VARIANCE(SAL)
-------------
   1646215.28

SQL> SELECT SUM(SAL) FROM EMPLOYEES GROUP BY DEPTID;

  SUM(SAL)
----------
      6775
      7100
      7450

SQL> SELECT DEPTID, SUM(SAL) FROM EMPLOYEES GROUP BY DEPTID;

    DEPTID   SUM(SAL)
---------- ----------
        20       6775
        30       7100
        10       7450

SQL>
```

11. **Find the average salary of all the employees of each department**

SELECT DEPTID, AVG(SAL) FROM EMPLOYEES GROUP BY DEPTID;

```
----------
      6775
      7100
      7450

SQL> SELECT DEPTID, SUM(SAL) FROM EMPLOYEES GROUP BY DEPTID;

    DEPTID   SUM(SAL)
---------- ----------
        20       6775
        30       7100
        10       7450

SQL> SELECT DEPTID, AVG(SAL) FROM EMPLOYEES GROUP BY DEPTID;

    DEPTID   AVG(SAL)
---------- ----------
        20 2258.33333
        30       1775
        10       3725

SQL>
```

12. Find the count of the employees in each job type where the salary >1000.

SELECT JOB, COUNT(EMPNO) FROM EMPLOYEES WHERE SAL>1000 GROUP BY JOB;

```
            3
            3
            1
            1

SQL> SELECT JOB, \COUNT(EMPNO) FROM EMPLOYEES WHERE SAL>1000 GROUP BY JOB;
SELECT JOB, \COUNT(EMPNO) FROM EMPLOYEES WHERE SAL>1000 GROUP BY JOB
            *
ERROR at line 1:
ORA-00911: invalid character


SQL> SELECT JOB, COUNT(EMPNO) FROM EMPLOYEES WHERE SAL>1000 GROUP BY JOB;

JOB        COUNT(EMPNO)
---------- ------------
salesman              3
manager               3
analyst               1
president             1

SQL>
```

13. Find the department id whose total number of employees is greater than 3.

SELECT DEPTID, COUNT(EMPNO) FROM EMPLOYEES

GROUP BY DEPTID HAVING COUNT(EMPNO) >3;

```
3;

    DEPTID COUNT(EMPNO)
---------- ------------
        30            4

SQL> SELECT DEPTID, COUNT(EMPNO) FROM EMPLOYEES GROUP BY DEPTID;

    DEPTID COUNT(EMPNO)
---------- ------------
        20            3
        30            4
        10            2

SQL> SELECT DEPTID, COUNT(EMPNO) FROM EMPLOYEES GROUP BY DEPTID HAVING COUNT(EMPNO) >3;

    DEPTID COUNT(EMPNO)
---------- ------------
        30            4

SQL>
```

14. Find the next Monday of the hiredates of the employees who are clerks.

SELECT EMPNO, hiredate, NEXT_DAY(hiredate, 'MONDAY') AS next_monday FROM employees WHERE job = 'clerk';

```
SQL> SELECT DEPTID, COUNT(EMPNO) FROM EMPLOYEES GROUP BY DEPTID;

    DEPTID COUNT(EMPNO)
---------- ------------
        20            3
        30            4
        10            2

SQL> SELECT DEPTID, COUNT(EMPNO) FROM EMPLOYEES GROUP BY DEPTID HAVING COUNT(EMPNO) >3;

    DEPTID COUNT(EMPNO)
---------- ------------
        30            4

SQL> SELECT EMPNO, hiredate, NEXT_DAY(hiredate, 'MONDAY') AS next_monday FROM employees W
HERE job = 'clerk';

     EMPNO HIREDATE  NEXT_MOND
---------- --------- ---------
      7369 17-DEC-19 23-DEC-19

SQL>
```

Q2. Create the following table named as sailors, and answer the following.

a. **List the distinct ratings among all the sailors.**

select distinct rating from sailors;

```
SQL> INSERT INTO SAILORS VALUES(4, 'Zorba', 8, 18, 5400.003);

1 row created.

SQL> INSERT INTO SAILORS VALUES(5, 'Julius', 5, 25, 34123.888);

1 row created.

SQL> select * from sailors;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         1 Dustin              7         45   2345.645
         2 Rusty              10         35   1345.763
         3 Horatio             5         35   3456.726
         4 Zorba               8         18   5400.003
         5 Julius              5         25  34123.888

SQL> select distict rating from sailors;
select distict rating from sailors
       *
ERROR at line 1:
ORA-00904: "DISTICT": invalid identifier


SQL> select distinct rating from sailors;

    RATING
----------
         7
        10
         5
         8

SQL>
```

b. **Write a query to round the salary of each sailor to the nearest whole number.**

select sailor_id, name, round(salary) from sailors;

```
 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         1 Dustin              7         45   2345.645
         2 Rusty              10         35   1345.763
         3 Horatio             5         35   3456.726
         4 Zorba               8         18   5400.003
         5 Julius              5         25  34123.888

SQL> select distict rating from sailors;
select distict rating from sailors
       *
ERROR at line 1:
ORA-00904: "DISTICT": invalid identifier


SQL> select distinct rating from sailors;

    RATING
----------
         7
        10
         5
         8

SQL> select sailor_id, name, round(salary) from sailors;

 SAILOR_ID NAME       ROUND(SALARY)
---------- ---------- -------------
         1 Dustin              2346
         2 Rusty               1346
         3 Horatio             3457
         4 Zorba               5400
         5 Julius             34124

SQL>
```

c. Write a query to truncate the salary of each sailor to two decimal places.

select sailor_id, name, trunc(salary, 2) from sailors;

```
        10
         5
         8

SQL> select sailor_id, name, round(salary) from sailors;

 SAILOR_ID NAME       ROUND(SALARY)
---------- ---------- -------------
         1 Dustin              2346
         2 Rusty               1346
         3 Horatio             3457
         4 Zorba               5400
         5 Julius             34124

SQL> select * from sailors;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         1 Dustin              7         45   2345.645
         2 Rusty              10         35   1345.763
         3 Horatio             5         35   3456.726
         4 Zorba               8         18   5400.003
         5 Julius              5         25  34123.888

SQL> select sailor_id, name, trunc(salary, 2) from sailors;

 SAILOR_ID NAME       TRUNC(SALARY,2)
---------- ---------- ---------------
         1 Dustin             2345.64
         2 Rusty              1345.76
         3 Horatio            3456.72
         4 Zorba                 5400
         5 Julius            34123.88

SQL> |
```

d. Write a query to find the square of the ratings of the sailors and display it along with their names.

select name, rating*rating as sqr_rating from sailors;

```
         3 Horatio             3457
         4 Zorba               5400
         5 Julius             34124

SQL> select * from sailors;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         1 Dustin              7         45   2345.645
         2 Rusty              10         35   1345.763
         3 Horatio             5         35   3456.726
         4 Zorba               8         18   5400.003
         5 Julius              5         25  34123.888

SQL> select sailor_id, name, trunc(salary, 2) from sailors;

 SAILOR_ID NAME       TRUNC(SALARY,2)
---------- ---------- ---------------
         1 Dustin             2345.64
         2 Rusty              1345.76
         3 Horatio            3456.72
         4 Zorba                 5400
         5 Julius            34123.88

SQL> select name, rating*rating as sqr_rating from sailors;

NAME       SQR_RATING
---------- ----------
Dustin             49
Rusty             100
Horatio            25
Zorba              64
Julius             25

SQL> |
```

e. Write suitable query to display all sailors with even numbered sailor id.

SELECT * FROM SAILORS WHERE MOD(SAILOR_ID,2) = 0;

```
SQL> select sailor_id, name, trunc(salary, 2) from sailors;

 SAILOR_ID NAME       TRUNC(SALARY,2)
---------- ---------- ---------------
         1 Dustin             2345.64
         2 Rusty              1345.76
         3 Horatio            3456.72
         4 Zorba                 5400
         5 Julius            34123.88

SQL> select name, rating*rating as sqr_rating from sailors;

NAME       SQR_RATING
---------- ----------
Dustin             49
Rusty             100
Horatio            25
Zorba              64
Julius             25

SQL> select * from sailors where sailor_id%2 = 0;
select * from sailors where sailor_id%2 = 0
                                        *
ERROR at line 1:
ORA-00911: invalid character


SQL> select * from sailors where mod(sailor_id,2) = 0;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ---------
         2 Rusty              10         35  1345.763
         4 Zorba               8         18  5400.003

SQL>
```

f) Write an SQL query to find the latest date from the two dates, 11-05-1988 and 12 06-1989.

SELECT GREATEST(TO_DATE('11-05-1988', 'DD-MM-YYYY'), TO_DATE('12-06-1989', 'DD-MM-YYYY')) AS latest_date FROM DUAL;

```
SQL> select * from empview;

NAME               SAL     DEPTID
---------- ---------- ----------
sahrikh          1600         30
ward             1250         30
martin           1400         30
blake            2850         30
clark            2450         10
king             5000         10
SOUMYA           6000         10

7 rows selected.

SQL> SELECT GREATEST(TO_DATE('11-05-1988', 'DD-MM-YYYY'), TO_DATE('12-06-1989', 'DD-MM-YYYY'))
 AS latest_date
  2  FROM DUAL;

LATEST_DA
---------
12-JUN-89

SQL>
```

g) Write three SQL query to round of the date 12/05/2023 16:47:00 to the nearest date, month and year formats separately.

SELECT ROUND(TO_DATE('12/05/2023 16:47:00', 'DD/MM/YYYY HH24:MI:SS'), 'DD') AS rounded_date FROM DUAL;

```
martin          1400        30
blake           2850        30
clark           2450        10
king            5000        10
SOUMYA          6000        10

7 rows selected.

SQL> SELECT GREATEST(TO_DATE('11-05-1988', 'DD-MM-YYYY'), TO_DATE('12-06-1989', 'DD-MM-YYYY'))
 AS latest_date
  2  FROM DUAL;

LATEST_DA
---------
12-JUN-89

SQL> SELECT ROUND(TO_DATE('12/05/2023 16:47:00', 'DD/MM/YYYY HH24:MI:SS'), 'DD') AS rounded_da
te
  2  FROM DUAL;

ROUNDED_D
---------
13-MAY-23

SQL>
```

h) Find the number of months between the two dates 23-05-1985 and 24-11-2020.

SELECT MONTHS_BETWEEN(TO_DATE('24-11-2020', 'DD-MM-YYYY'), TO_DATE('23-05-1985', 'DD-MM-YYYY')) AS MONTHS_DIFFERENCE FROM DUAL;

```
SQL> SELECT GREATEST(TO_DATE('11-05-1988', 'DD-MM-YYYY'), TO_DATE('12-06-1989', 'DD-MM-YYYY'))
 AS latest_date
  2  FROM DUAL;

LATEST_DA
---------
12-JUN-89

SQL> SELECT ROUND(TO_DATE('12/05/2023 16:47:00', 'DD/MM/YYYY HH24:MI:SS'), 'DD') AS rounded_da
te
  2  FROM DUAL;

ROUNDED_D
---------
13-MAY-23

SQL> SELECT MONTHS_BETWEEN(TO_DATE('24-11-2020', 'DD-MM-YYYY'), TO_DATE('23-05-1985', 'DD-MM-Y
YYY')) AS months_difference
  2  FROM DUAL;

MONTHS_DIFFERENCE
-----------------
       426.032258

SQL>
```

i. **Write a query to display the remainder when each sailors's salary is divided by 100.**

select sailor_id, name, salary, mod(salary, 100) from sailors;

```
SQL> select name, rating*rating as sqr_rating from sailors;

NAME       SQR_RATING
---------- ----------
Dustin             49
Rusty             100
Horatio            25
Zorba              64
Julius             25

SQL> select * from sailors where sailor_id%2 = 0;
select * from sailors where sailor_id%2 = 0
                                        *
ERROR at line 1:
ORA-00911: invalid character


SQL> select * from sailors where mod(sailor_id,2) = 0;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         2 Rusty              10         35   1345.763
         4 Zorba               8         18   5400.003

SQL> select sailor_id, name, salary, mod(salary, 100) from sailors;

 SAILOR_ID NAME           SALARY MOD(SALARY,100)
---------- ---------- ---------- ---------------
         1 Dustin       2345.645          45.645
         2 Rusty        1345.763          45.763
         3 Horatio      3456.726          56.726
         4 Zorba        5400.003            .003
         5 Julius      34123.888          23.888

SQL>
```

j. **Change all lower-case letters in the names of the sailors to uppercase and display Ans: UPDATE sailors SET name = UPPER (name);**

update sailors set name = upper(name);

```
ORA-00911: invalid character


SQL> select * from sailors where mod(sailor_id,2) = 0;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         2 Rusty              10         35   1345.763
         4 Zorba               8         18   5400.003

SQL> select sailor_id, name, salary, mod(salary, 100) from sailors;

 SAILOR_ID NAME           SALARY MOD(SALARY,100)
---------- ---------- ---------- ---------------
         1 Dustin       2345.645          45.645
         2 Rusty        1345.763          45.763
         3 Horatio      3456.726          56.726
         4 Zorba        5400.003            .003
         5 Julius      34123.888          23.888

SQL> update sailors set name = upper(name);

5 rows updated.

SQL> select * from sailors;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         1 DUSTIN              7         45   2345.645
         2 RUSTY              10         35   1345.763
         3 HORATIO             5         35   3456.726
         4 ZORBA               8         18   5400.003
         5 JULIUS              5         25  34123.888

SQL>
```

k. Find the sailor with the longest name using SQL query.

SELECT NAME FROM SAILORS ORDER BY LENGTH(NAME) DESC FETCH FIRST 1 ROW ONLY;

```
---------- ---------- ---------- ---------------
         1 Dustin       2345.645          45.645
         2 Rusty        1345.763          45.763
         3 Horatio      3456.726          56.726
         4 Zorba        5400.003            .003
         5 Julius      34123.888          23.888

SQL> update sailors set name = upper(name);

5 rows updated.

SQL> select * from sailors;

 SAILOR_ID NAME           RATING        AGE     SALARY
---------- ---------- ---------- ---------- ----------
         1 DUSTIN              7         45   2345.645
         2 RUSTY              10         35   1345.763
         3 HORATIO             5         35   3456.726
         4 ZORBA               8         18   5400.003
         5 JULIUS              5         25  34123.888

SQL> select name form sailors order by length(name) desc fetch first 1 row only;
select name form sailors order by length(name) desc fetch first 1 row only
                 *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> select name from sailors order by length(name) desc fetch first 1 row only;

NAME
----------
HORATIO

SQL>
```

L. Concatenate the name and age of a sailors and display the results.

SELECT CONCAT(NAME, AGE) AS NAME_AGE FROM SAILORS;

```
         3 HORATIO             5         35   3456.726
         4 ZORBA               8         18   5400.003
         5 JULIUS              5         25  34123.888

SQL> select name form sailors order by length(name) desc fetch first 1 row only;
select name form sailors order by length(name) desc fetch first 1 row only
                 *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> select name from sailors order by length(name) desc fetch first 1 row only;

NAME
----------
HORATIO

SQL> select concat(name, ' ', age) as name_age from sailors;
select concat(name, ' ', age) as name_age from sailors
       *
ERROR at line 1:
ORA-00909: invalid number of arguments


SQL> select concat(name, age) as name_age from sailors;

NAME_AGE
--------------------------------------------------
DUSTIN45
RUSTY35
HORATIO35
ZORBA18
JULIUS25

SQL>
```

Q3. From the table created in Q1.

a) Create a view Empview with the columns name salary and deptno.

CREATE VIEW EMPVIEW AS SELECT NAME, SAL, DEPTID FROM EMPLOYEES;
SELECT * FROM EMPVIEW;

```
SQL Plus
      7369 smith      clerk      17-DEC-19        800         20
      7499 allen      salesman   20-FEB-19       1600         30
      7521 ward       salesman   22-FEB-19       1250         30
      7566 jones      manager    02-APR-19       2975         20
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30
      7782 clark      manager    09-JUN-19       2450         10
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10

9 rows selected.

SQL> commit;

Commit complete.

SQL> CREATE VIEW EMPVIEW AS SELECT NAME, SAL, DEPTID FROM EMPLOYEES;

View created.

SQL> SELECT * FROM EMPVIEW;

NAME               SAL     DEPTID
---------- ---------- ----------
smith             800         20
allen            1600         30
ward             1250         30
jones            2975         20
martin           1400         30
blake            2850         30
clark            2450         10
scott            3000         20
king             5000         10

9 rows selected.

SQL>
```

b) Create a view Empview 1 in read only mode.

CREATE VIEW EMPVIEW1

AS SELECT NAME, SAL, DEPTID FROM EMPLOYEES WITH READ ONLY;

SELECT * FROM EMPVIEW1;

```
SQL Plus
SQL> SELECT * FROM EMPVIEW;

NAME               SAL     DEPTID
---------- ---------- ----------
smith             800         20
allen            1600         30
ward             1250         30
jones            2975         20
martin           1400         30
blake            2850         30
clark            2450         10
scott            3000         20
king             5000         10

9 rows selected.

SQL> CREATE VIEW EMPVIEW1 AS SELECT NAME, SAL, DEPTID FROM EMPLOYEES WITH READ ONLY;

View created.

SQL> SELECT * FROM EMPVIEW1;

NAME               SAL     DEPTID
---------- ---------- ----------
smith             800         20
allen            1600         30
ward             1250         30
jones            2975         20
martin           1400         30
blake            2850         30
clark            2450         10
scott            3000         20
king             5000         10

9 rows selected.

SQL>
```

c) Add a tuple to your view Empview and Empview1 and check the outputs.

Inserting to EMPVIEW:

INSERT INTO EMPVIEW VALUES ('SOUMYA', 6000, 10);

Inserting to EMPVIEW1:

INSERT INTO EMPVIEW1 VALUES ('JOHN', 3000, 20);

```
clark            2450         10
scott            3000         20
king             5000         10

9 rows selected.

SQL> INSERT INTO EMPVIEW VALUES ('SOUMYA', 6000, 10);

1 row created.

SQL> SELECT * FROM EMPVIEW;

NAME              SAL     DEPTID
---------- ---------- ----------
smith             800         20
allen            1600         30
ward             1250         30
jones            2975         20
martin           1400         30
blake            2850         30
clark            2450         10
scott            3000         20
king             5000         10
SOUMYA           6000         10

10 rows selected.

SQL> INSERT INTO EMPVIEW1 VALUES ('JOHN', 3000, 20);
INSERT INTO EMPVIEW1 VALUES ('JOHN', 3000, 20)
*
ERROR at line 1:
ORA-42399: cannot perform a DML operation on a read-only view


SQL>
```

d) Check the table EMPLOYEES whether the tuple you added to Empview is added or not.

SELECT * FROM EMPLOYEES;

```
jones            2975         20
martin           1400         30
blake            2850         30
clark            2450         10
scott            3000         20
king             5000         10
SOUMYA           6000         10

10 rows selected.

SQL> INSERT INTO EMPVIEW1 VALUES ('JOHN', 3000, 20);
INSERT INTO EMPVIEW1 VALUES ('JOHN', 3000, 20)
*
ERROR at line 1:
ORA-42399: cannot perform a DML operation on a read-only view


SQL> select * from employees;

     EMPNO NAME       JOB        HIREDATE         SAL     DEPTID
---------- ---------- ---------- --------- ---------- ----------
      7369 smith      clerk      17-DEC-19        800         20
      7499 allen      salesman   20-FEB-19       1600         30
      7521 ward       salesman   22-FEB-19       1250         30
      7566 jones      manager    02-APR-19       2975         20
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30
      7782 clark      manager    09-JUN-19       2450         10
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10
           SOUMYA                                6000         10

10 rows selected.

SQL>
```

e) Delete tuples from the view Empview where deptno is 20;

DELETE FROM EMPVIEW WHERE DEPTID = 20;

```
SQL Plus
      7654 martin     salesman   28-SEP-19       1400         30
      7698 blake      manager    01-MAY-19       2850         30
      7782 clark      manager    09-JUN-19       2450         10
      7788 scott      analyst    19-APR-19       3000         20
      7839 king       president  17-NOV-19       5000         10
           SOUMYA                                6000         10

10 rows selected.

SQL> DELETE FROM Empview WHERE deptno = 20;
DELETE FROM Empview WHERE deptno = 20
                           *
ERROR at line 1:
ORA-00904: "DEPTNO": invalid identifier


SQL> DELETE FROM Empview WHERE deptid = 20;

3 rows deleted.

SQL> select * from empview;

NAME              SAL     DEPTID
---------- ---------- ----------
allen            1600         30
ward             1250         30
martin           1400         30
blake            2850         30
clark            2450         10
king             5000         10
SOUMYA           6000         10

7 rows selected.

SQL> |
```

f) Modify the tuple with name Allen as Sahrikh in the view Empview.

UPDATE EMPVIEW SET NAME = 'sahrikh' WHERE NAME = 'allen';

```
SQL Plus
3 rows deleted.

SQL> select * from empview;

NAME              SAL     DEPTID
---------- ---------- ----------
allen            1600         30
ward             1250         30
martin           1400         30
blake            2850         30
clark            2450         10
king             5000         10
SOUMYA           6000         10

7 rows selected.

SQL> UPDATE EMPVIEW SET NAME = 'sahrikh' WHERE NAME = 'allen';

1 row updated.

SQL> select * from empview;

NAME              SAL     DEPTID
---------- ---------- ----------
sahrikh          1600         30
ward             1250         30
martin           1400         30
blake            2850         30
clark            2450         10
king             5000         10
SOUMYA           6000         10

7 rows selected.

SQL> |
```