Ex.No.9 .04.2025	SET OPERATIONS AND AGGEREGATE FUNCTIONS

AIM

To perform various set operations, aggregate functions, group by and having clause on the relational database.

CREATE TABLE

CREATE TABLE STUDENTS_DETAILS(S_ID VARCHAR2(10), S_NAME VARCHAR2(50), CITY VARCHAR2(50));

Table created.

CREATE TABLE STUDENT_INFO(S_ID VARCHAR2(10), S_NAME VARCHAR2(50), S_CITY VARCHAR2(50));

Table created.

CREATE TABLE EMPLOYEE_INFO(NAME VARCHAR2(50), DEPT VARCHAR2(20), ADDRESS VARCHAR2(50), SALARY NUMBER(8));

Table created.

INSERTING VALUES

```
SQL> INSERT INTO STUDENTS_DETAILS VALUES('cse01','JAI','ERODE');
1 row created.

SQL> INSERT INTO STUDENTS_DETAILS VALUES('it01','KARTHI','CHENNAI');
1 row created.

SQL> INSERT INTO STUDENTS_DETAILS VALUES('ece01','AJITH','BANGALORE');
1 row created.

SQL> INSERT INTO STUDENTS_DETAILS VALUES('cse02','JEGAN','MUMBAI');
1 row created.

SQL> INSERT INTO STUDENTS_DETAILS VALUES('mtr01','SANJAI','SALEM');
1 row created.

SQL> INSERT INTO STUDENT_INFO VALUES('cse01','RAHUL','MADURAI');
1 row created.

SQL> INSERT INTO STUDENT_INFO VALUES('ece01','KAMALESH','ITALY');
1 row created.

SQL> INSERT INTO STUDENT_INFO VALUES('mec01','BABU','TRICHY');
1 row created.
```

```
SQL> INSERT INTO STUDENT_INFO VALUES('itr06','SANJAI','MORAPPUR');
1 row created.
SQL> INSERT INTO STUDENT_INFO VALUES('eie01','AJITH','BANGALORE');
1 row created.

SQL> INSERT INTO EMPLOYEE_INFO VALUES('JAI','IT','ERODE',60000);
1 row created.

SQL> INSERT INTO EMPLOYEE_INFO VALUES('KARTHI','IT','CHENNAI',50000);
1 row created.

SQL> INSERT INTO EMPLOYEE_INFO VALUES('AJITH','CSE','BANGALORE',90000);
1 row created.

SQL> INSERT INTO EMPLOYEE_INFO VALUES('JEGAN','ECE','COIMBATORE',45000);
1 row created.

SQL> INSERT INTO EMPLOYEE_INFO VALUES('SANJAI','CSE','BANGALORE',25000);
1 row created.

SQL> COMMIT;
Commit complete.
```

UNION KEYWORD

```
 SQL> SELECT S_ID, S_NAME FROM STUDENTS_DETAILS
    UNION
    SELECT S_ID, S_NAME FROM STUDENT_INFO;

S_ID       S_NAME
---------- --------------
cse01      JAI
cse01      RAHUL
cse02      JEGAN
ece01      KAMALESH
ece01      AJITH
eie01      AJITH
it01       KARTHI
itr06      SANJAI
mec01      BABU
mtr01      SANJAI
10 rows selected.
```

UNION ALL KEYWORD

```
  SQL> SELECT S_ID, S_NAME, CITY FROM STUDENTS_DETAILS
    UNION ALL
    SELECT S_ID, S_NAME, S_CIT Y FROM STUDENT_INFO;
```

```
S_ID        S_NAME      CITY
-----------  --------------   ---------
cse01      JAI                 ERODE

it01        KARTHI          CHENNAI

ece01      AJITH              BANGALORE


S_ID        S_NAME        CITY
-----------   -------------         -----
cse02      JEGAN           MUMBAI

mtr01      SANJAI          SALEM

cse01      RAHUL          MADURAI


S_ID        S_NAME       CITY
-----------  -------------         -------------
ece01      KAMALESH    ITALY

mec01     BABU             TRICHY

itr06       SANJAI          MORAPPUR


S_ID        S_NAME            CITY
-----------     ---------------        --------
eie01          AJITH                 BANGALORE.
10 rows selected.
```

INTERSECT KEYWORD

```
 SQL> SELECT * FROM STUDENTS_DETAILS
      INTERSECT
      SELECT * FROM STUDENT_INFO;
  no rows selected
```

MINUS KEYWORD

```
SQL> SELECT * FROM STUDENT_INFO
    MINUS
    SELECT * FROM STUDENTS_DETAILS;
```

```
S_ID         S_NAME          S_CITY
-----------      ----------          --------
cse01      RAHUL             MADURAI

ece01      KAMALESH       ITALY

eie01      AJITH                 BANGALORE


S_ID         S_NAME       S_CITY
-----------      ----------       ----------------
itr06            SANJAI     MORAPPUR

mec01          BABU       TRICHY
```

AGGREGATE FUNCTIONS MAX

```
SQL> SELECT MAX(SALARY) FROM EMPLOYEE_INFO;

 MAX(SALARY)
    -------
     90000

SQL> SELECT MIN(SALARY) FROM EMPLOYEE_INFO;

MIN(SALARY)
   -------
     25000

SQL> SELECT AVG(SALARY) FROM EMPLOYEE_INFO;

 AVG(SALARY)
   -------
   52601.6

SQL> SELECT SUM(SALARY) FROM EMPLOYEE_INFO;

  SUM(SALARY)
    -------
    263008

SQL> SELECT COUNT(NAME) AS no_of_employee FROM EMPLOYEE_INFO;

  NO_OF_EMPLOYEE
      ---------
           5
```

SECOND MAXIMUM SALARY

```
 SQL> SELECT MAX(SALARY) FROM EMPLOYEE_INFO
  WHERE SALARY NOT IN (SELECT MAX(SALARY) FROM EMPLOYEE_INFO);

MAX(SALARY)
- - - - - - -
   60000
```

SECOND MINIMUM SALARY

```
SQL> SELECT MIN(SALARY) FROM EMPLOYEE_INFO
   WHERE SALARY NOT IN (SELECT MIN(SALARY) FROM EMPLOYEE_INFO);

MIN(SALARY)
- - - - - - -
    45000
```

AGGREGATE FUNCTIONS WITH GROUPBY AND HAVING:

GROUP BY

```
SQL> SELECT DEPT, AVG(SALARY) AS avg_salary
     FROM EMPLOYEE_INFO
     GROUP BY DEPT;

DEPT              AVG_SALARY
-----------------    -----------
 IT                  55000
CSE                  54004
ECE                  45000
```

```
SQL> SELECT DEPT,SUM(SALARY)AS total_salary

      FROM EMPLOYEE_INFO

      GROUP BY DEPT

      HAVING AVG(SALARY)>43000;

DEPT              TOTAL_SALARY
--------------         -------------
IT                     110000
CSE                    108008
ECE                    45000
```

DISTINCT

```
SQL> SELECT DISTINCT DEPT FROM EMPLOYEE_INFO;


DEPT

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

IT

CSE

ECE
```

TO FIND THE EMPLOYEES WHO EARN SALARY HIGHER THAN THE AVG SALARY OF THEIR CITY

```
    SQL>SELECT NAME FROM EMPLOYEE_INFO e WHERE SALARY > (SELECT
AVG(SALARY) FROM EMPLOYEE_INFO WHERE ADDRESS = e.ADDRESS);

NAME
- - - - - - - - -
AJITH
```

TO FIND THE NAME OF THE PERSONS WHO HAVE HIGHER SALARY THAN THE AVERAGE SALARY OF THEIR DEPARTMENT

```
     SQL>SELECT NAME FROM EMPLOYEE_INFO E WHERE SALARY > (SELECT
AVG(SALARY) FROM EMPLOYEE_INFO WHERE DEPT = E.DEPT);

NAME
- - - - - - - - - - - - - - -
JAI KARTHI
```

CONTENTS	MARKS ALLOTED	MARKS OBTAINED
Aim,Algorithm,SQL,PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT

Thus, various set operations, aggregate computations, and grouping techniques using GROUP BY and HAVING clauses were effectively applied to the relational database.