|  |  |
| --- | --- |
| 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','Job','ERODE');

1 row created.

SQL> INSERT INTO STUDENTS\_DETAILS VALUES('it01','lisa','CHENNAI');

1 row created.

SQL> INSERT INTO STUDENTS\_DETAILS VALUES('ece01','jisoo','BANGALORE');

1 row created.

SQL> INSERT INTO STUDENTS\_DETAILS VALUES('cse02','rose','MUMBAI');

1 row created.

SQL> INSERT INTO STUDENTS\_DETAILS VALUES('mtr01','jino','SALEM');

1 row created.

SQL> INSERT INTO STUDENT\_INFO VALUES('cse01','jennie','MADURAI');

1 row created.

SQL> INSERT INTO STUDENT\_INFO VALUES('ece01','paul','ITALY');

1 row created.

SQL> INSERT INTO STUDENT\_INFO VALUES('mec01','kane','TRICHY');

1 row created.

SQL> INSERT INTO STUDENT\_INFO VALUES('itr06','Millier','MORAPPUR');

1 row created.

SQL> INSERT INTO STUDENT\_INFO VALUES('eie01','Kholi','BANGALORE');

1 row created.

SQL> INSERT INTO EMPLOYEE\_INFO VALUES('Dhoni','IT','ERODE',60000);

1 row created.

SQL> INSERT INTO EMPLOYEE\_INFO VALUES('Ron','IT','CHENNAI',50000);

1 row created.

SQL> INSERT INTO EMPLOYEE\_INFO VALUES('Rohit','CSE','BANGALORE',78008);

1 row created.

SQL> INSERT INTO EMPLOYEE\_INFO VALUES('JEGAN','ECE','COIMBATORE',45000);

1 row created.

SQL> INSERT INTO EMPLOYEE\_INFO VALUES('Nancy','CSE','BANGALORE',30000);

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 jennie

cse01 Rose

cse02 Job

ece01 Jisoo

ece01 Paul

eie01 Kholi

it01 lisa

itr06 Miller

mec01 Kane

mtr01 Jino

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 Job ERODE

it01 lisa CHENNAI

ece01 paul BANGALORE

S\_ID S\_NAME CITY

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

cse02 Jennie MUMBAI

mtr01 Jino SALEM

cse01 Rose MADURAI

S\_ID S\_NAME CITY

---------- ----------- ----------- ece01 KAMALESH ITALY

mec01 Kane TRICHY

itr06 Miller MORAPPUR

S\_ID S\_NAME CITY

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

eie01 Kholi 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 jennie MADURAI

ece01 paul ITALY

eie01 Kholi BANGALORE

S\_ID S\_NAME S\_CITY

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

itr06 Miller MORAPPUR

mec01 Jino TRICHY

# AGGREGATE FUNCTIONS MAX

SQL> SELECT MAX(SALARY) FROM EMPLOYEE\_INFO;

MAX(SALARY)

----------- 78008

SQL> SELECT MIN(SALARY) FROM EMPLOYEE\_INFO;

MIN(SALARY)

----------- 30000

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

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

Jennie

# 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

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

Job

Jennie

|  |  |  |
| --- | --- | --- |
| 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 functions, group by and having clause operations are performed on the relational database.