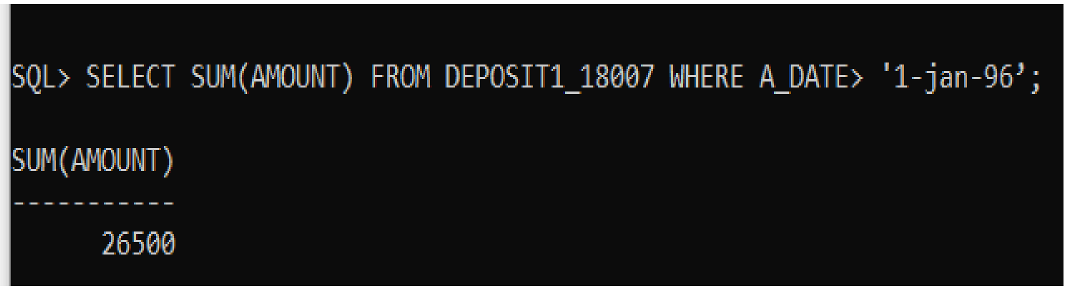
PRACTICAL NO-8

AIM:TO APPLY THE CONCEPT OF AGGREGATING DATA USING GROUP FUNCTIONS.

Query statement1:List total deposit of customer having account date after 1-jan-96.

- Query :SELECT SUM(AMOUNT) FROM DEPOSIT1_18007 WHERE A_DATE> '1-jan-96';

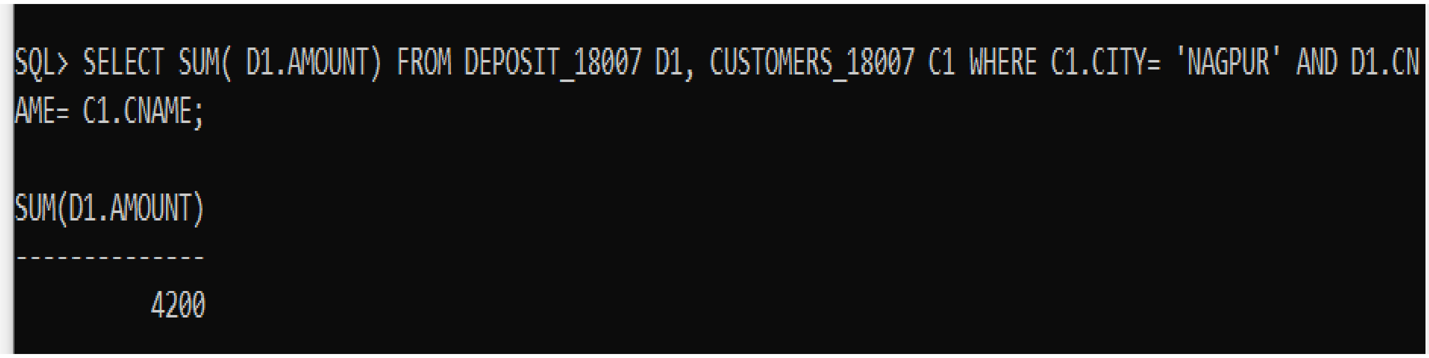


```
SQL> SELECT SUM(AMOUNT) FROM DEPOSIT1_18007 WHERE A_DATE> '1-jan-96';

SUM(AMOUNT)
-----------
      26500
```

Query statement 2:List total deposit of customers living in city Nagpur.

- Query :SELECT SUM(D1.AMOUNT) FROM DEPOSIT_18007 D1, CUSTOMERS_18007 C1 WHERE C1.CITY= 'NAGPUR' AND D1.CNAME= C1.CNAME;

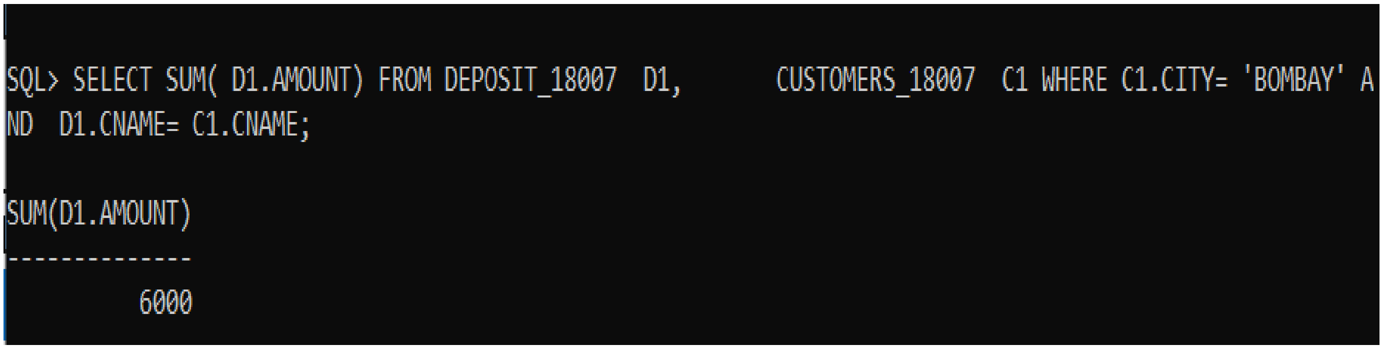


```
SQL> SELECT SUM( D1.AMOUNT) FROM DEPOSIT_18007 D1, CUSTOMERS_18007 C1 WHERE C1.CITY= 'NAGPUR' AND D1.CN
AME= C1.CNAME;

SUM(D1.AMOUNT)
--------------
          4200
```

Query statement 3: List maximum deposit of customers living in bombay.

- Query :SELECT SUM(D1.AMOUNT) FROM DEPOSIT_18007 D1, CUSTOMERS_18007 C1 WHERE C1.CITY= ‘BOMBAY’ AND D1.CNAME= C1.CNAME;

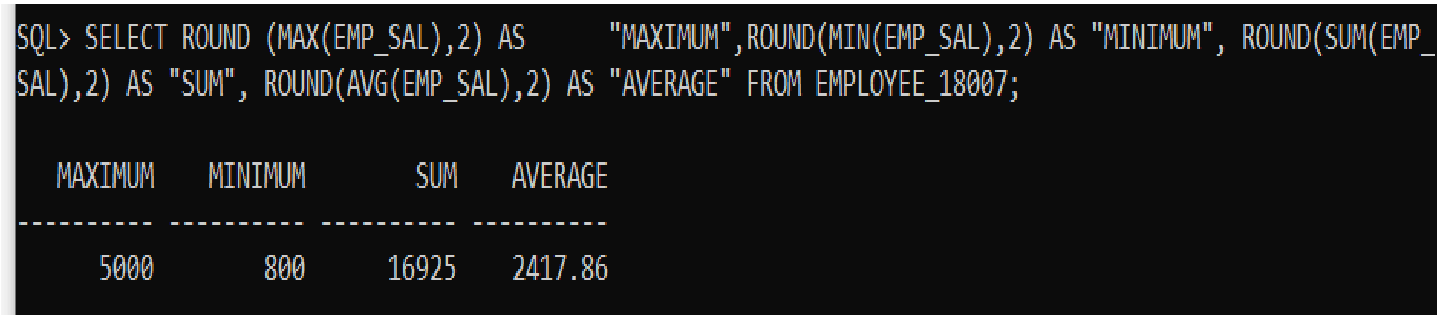


```
SQL> SELECT SUM( D1.AMOUNT) FROM DEPOSIT_18007  D1,      CUSTOMERS_18007  C1 WHERE C1.CITY= 'BOMBAY' A
ND  D1.CNAME= C1.CNAME;

SUM(D1.AMOUNT)
--------------
          6000
```

Query statement 4:Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

- Query :SELECT ROUND (MAX(EMP_SAL),2) AS "MAXIMUM",ROUND(MIN(EMP_SAL),2) AS "MINIMUM", ROUND(SUM(EMP_SAL),2) AS "SUM", ROUND(AVG(EMP_SAL),2) AS "AVERAGE" FROM EMPLOYEE_18007;

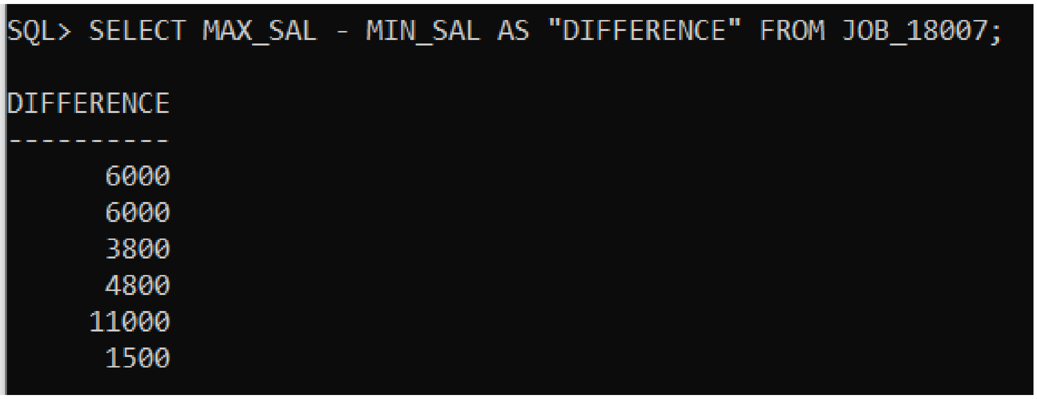


```
SQL> SELECT ROUND (MAX(EMP_SAL),2) AS      "MAXIMUM",ROUND(MIN(EMP_SAL),2) AS "MINIMUM", ROUND(SUM(EMP_
SAL),2) AS "SUM", ROUND(AVG(EMP_SAL),2) AS "AVERAGE" FROM EMPLOYEE_18007;

   MAXIMUM    MINIMUM        SUM    AVERAGE
---------- ---------- ---------- ----------
      5000        800      16925    2417.86
```

Query statement 5:Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

- Query :SELECT MAX_SAL - MIN_SAL AS “DIFFERENCE” FROM JOB_18007;

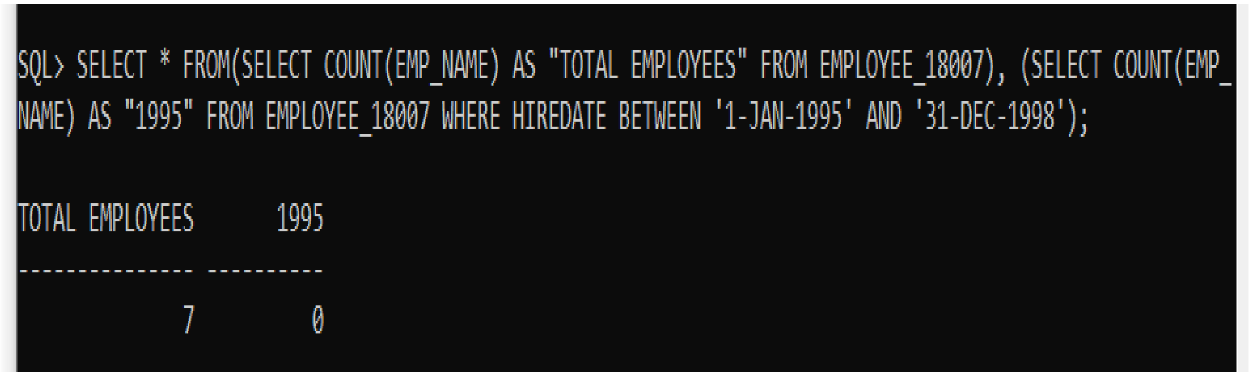


```
SQL> SELECT MAX_SAL - MIN_SAL AS "DIFFERENCE" FROM JOB_18007;

DIFFERENCE
----------
      6000
      6000
      3800
      4800
     11000
      1500
```

Query statement 6:Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998.

- Query :SELECT * FROM(SELECT COUNT(EMP_NAME) AS "TOTAL EMPLOYEES" FROM EMPLOYEE_18007), (SELECT COUNT(EMP_NAME) AS "1995" FROM EMPLOYEE_18007 WHERE HIREDATE BETWEEN '1-JAN-1995' AND '31-DEC-1998');

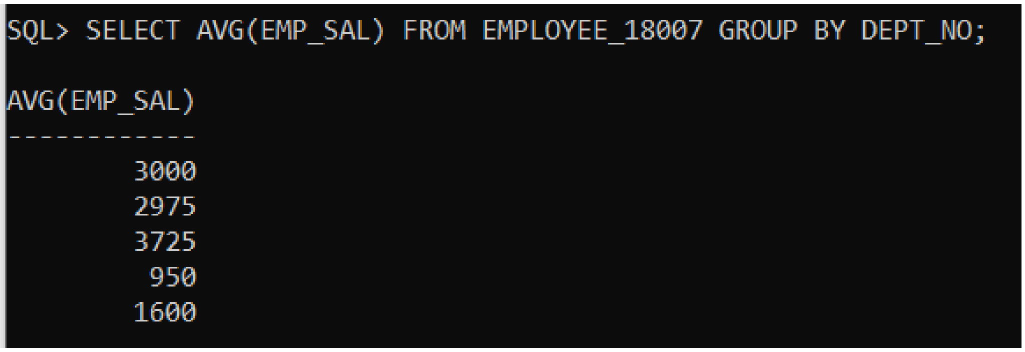


```
SQL> SELECT * FROM(SELECT COUNT(EMP_NAME) AS "TOTAL EMPLOYEES" FROM EMPLOYEE_18007), (SELECT COUNT(EMP_
NAME) AS "1995" FROM EMPLOYEE_18007 WHERE HIREDATE BETWEEN '1-JAN-1995' AND '31-DEC-1998');

TOTAL EMPLOYEES       1995
--------------- ----------
              7          0
```

Query statement 7:Find the average salaries for each department without displaying the respective department numbers.

- Query :SELECT AVG(EMP_SAL) FROM EMPLOYEE_18007 GROUP BY DEPT_NO;

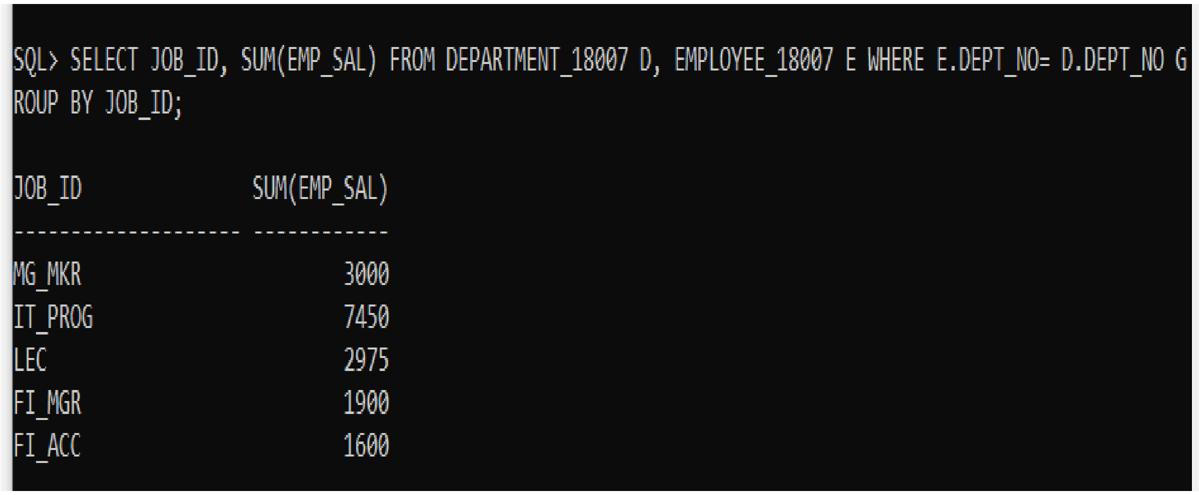


```
SQL> SELECT AVG(EMP_SAL) FROM EMPLOYEE_18007 GROUP BY DEPT_NO;

AVG(EMP_SAL)
------------
        3000
        2975
        3725
         950
        1600
```

Query statement 8 :Write a query to display the total salary being paid to each job title, within each department.

- Query : SELECT JOB_ID, SUM(EMP_SAL) FROM DEPARTMENT_18007 D, EMPLOYEE_18007 E WHERE E.DEPT_NO= D.DEPT_NO GROUP BY JOB_ID;

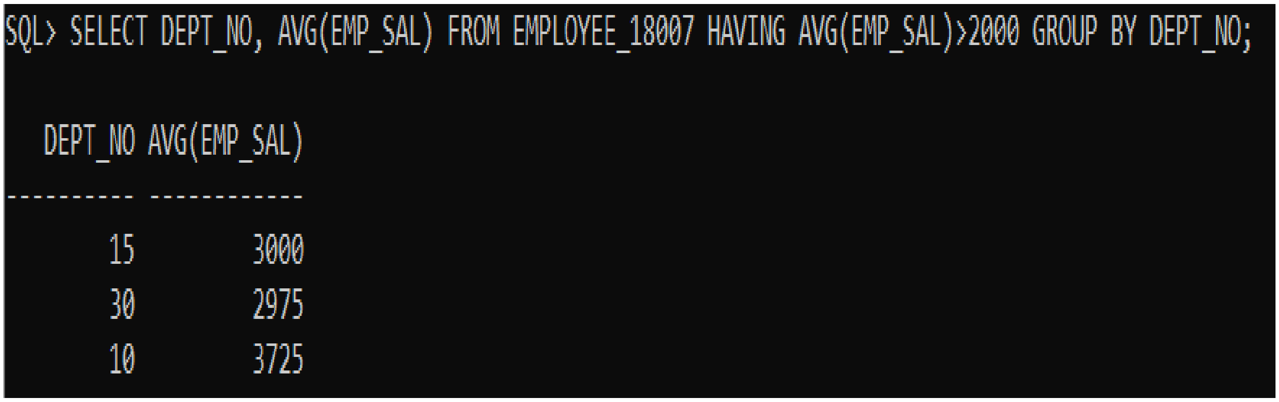


```
SQL> SELECT JOB_ID, SUM(EMP_SAL) FROM DEPARTMENT_18007 D, EMPLOYEE_18007 E WHERE E.DEPT_NO= D.DEPT_NO G
ROUP BY JOB_ID;

JOB_ID               SUM(EMP_SAL)
-------------------- ------------
MG_MKR                       3000
IT_PROG                      7450
LEC                          2975
FI_MGR                       1900
FI_ACC                       1600
```

Query statement 9 : Find the average salaries > 2000 for each department without displaying the respective department numbers.

- Query : SELECT DEPT_NO, AVG(EMP_SAL) FROM EMPLOYEE_18007 HAVING AVG(EMP_SAL)>2000 GROUP BY DEPT_NO;

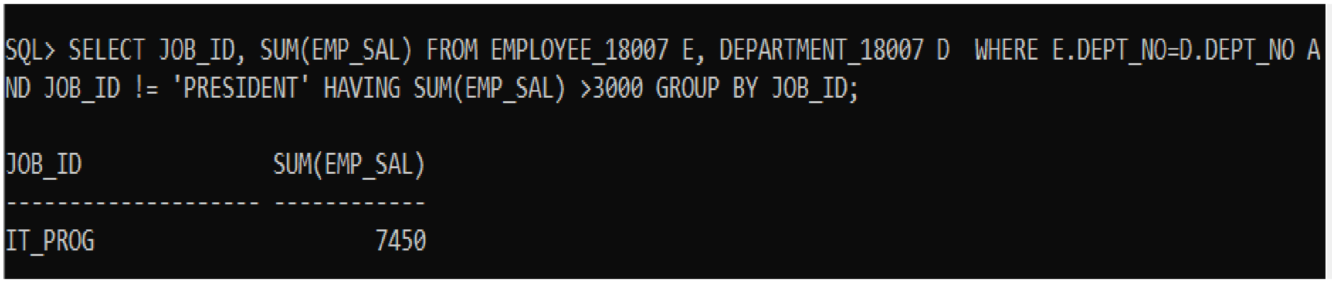


```
SQL> SELECT DEPT_NO, AVG(EMP_SAL) FROM EMPLOYEE_18007 HAVING AVG(EMP_SAL)>2000 GROUP BY DEPT_NO;


   DEPT_NO AVG(EMP_SAL)
---------- -----------
        15        3000
        30        2975
        10        3725
```

Query statement 10 :Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary.

- Query : SELECT JOB_ID, SUM(EMP_SAL) FROM EMPLOYEE_18007 E, DEPARTMENT_18007 D WHERE E.DEPT_NO=D.DEPT_NO AND JOB_ID != 'PRESIDENT' HAVING SUM(EMP_SAL) >3000 GROUP BY JOB_ID;

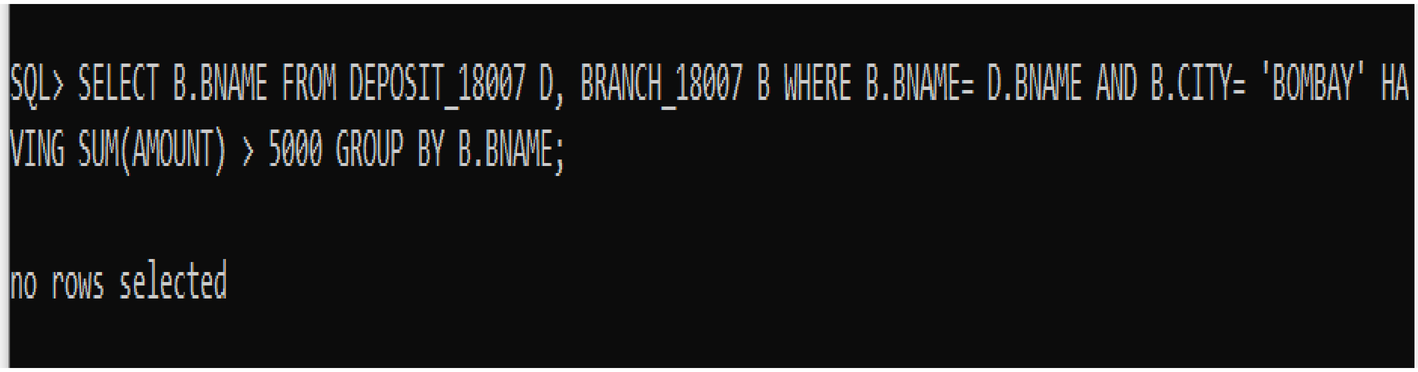


```
SQL> SELECT JOB_ID, SUM(EMP_SAL) FROM EMPLOYEE_18007 E, DEPARTMENT_18007 D  WHERE E.DEPT_NO=D.DEPT_NO A
ND JOB_ID != 'PRESIDENT' HAVING SUM(EMP_SAL) >3000 GROUP BY JOB_ID;

JOB_ID               SUM(EMP_SAL)
-------------------- -----------
IT_PROG                     7450
```

Query statement 11:List the branches having sum of deposit more than 5000 and located in city bombay.

- Query : SELECT B.BNAME FROM DEPOSIT_18007 D, BRANCH_18007 B WHERE B.BNAME= D.BNAME AND B.CITY= 'BOMBAY' HAVING SUM(AMOUNT) > 5000 GROUP BY B.BNAME;



```
SQL> SELECT B.BNAME FROM DEPOSIT_18007 D, BRANCH_18007 B WHERE B.BNAME= D.BNAME AND B.CITY= 'BOMBAY' HA
VING SUM(AMOUNT) > 5000 GROUP BY B.BNAME;

no rows selected
```