PRACTICAL – 03

NAME : Aryan Kashikar
Roll n0: D2 32

Aim : Implementation of queries using Aggregate Function and demonstrate the use of Group By clause on underlying tables in the database.

Problem Definition

- Create table using DDL Script

Products (product_id Integer, product_type_id Integer, Name Varchar2(30), Price Integer)

Note: product_id is Primary Key

```
SQL> CREATE TABLE Products (
  2    product_id INT PRIMARY KEY,
  3    product_type_id INT,
  4    Name VARCHAR2(30),
  5    Price INT
  6  );

Table created.
```

- Insert following 09 tupules using DML scripts
  - insert into products values (1,1,'Simple TV',1000.00);
  - insert into products values (2,1,'LED TV',1500.00);
  - insert into products values (3,1,'LCD TV',2000.00);
  - insert into products values (4,2,'Mobile Phone',1000.00);
  - insert into products values (5,2,'Smart Phone',2000.00);
  - insert into products values (6,2,'Jio Phone',3000.00);
  - insert into products values (7,3,'Simple WM',1500.00);
  - insert into products values (8,3,'Automated WM',2000.00);
  - insert into products values (9,3,'Semi WM',2500.00);

```
SQL> INSERT INTO Products VALUES (1, 1, 'Simple TV', 1000.00);

1 row created.

SQL> INSERT INTO Products VALUES (2, 1, 'LED TV', 1500.00);

1 row created.

SQL> INSERT INTO Products VALUES (3, 1, 'LCD TV', 2000.00);

1 row created.

SQL> INSERT INTO Products VALUES (4, 2, 'Mobile Phone', 1000.00);

1 row created.

SQL> INSERT INTO Products VALUES (5, 2, 'Smart Phone', 2000.00);

1 row created.

SQL> INSERT INTO Products VALUES (6, 2, 'Jio Phone', 3000.00);

1 row created.

SQL> INSERT INTO Products VALUES (7, 3, 'Simple WM', 1500.00);

1 row created.

SQL> INSERT INTO Products VALUES (8, 3, 'Automated WM', 2000.00);

1 row created.

SQL> INSERT INTO Products VALUES (9, 3, 'Semi WM', 2500.00);

1 row created.
```

OUTPUT:

```
SQL> SELECT * FROM Products;

PRODUCT_ID PRODUCT_TYPE_ID NAME                           PRICE
---------- --------------- ------------------------------ ----------
         1               1 Simple TV                      1000
         2               1 LED TV                         1500
         3               1 LCD TV                         2000
         4               2 Mobile Phone                   1000
         5               2 Smart Phone                    2000
         6               2 Jio Phone                      3000
         7               3 Simple WM                      1500
         8               3 Automated WM                   2000
         9               3 Semi WM                        2500

9 rows selected.
```

Tasks-01:

1. Count the number of products

```
SQL> SELECT COUNT(*) AS product_count FROM Products;

PRODUCT_COUNT
-------------
            9
```

2. Count the number of products and sum of price of products

```
SQL> SELECT COUNT(*) AS product_count, SUM(Price) AS total_price FROM Products;

PRODUCT_COUNT TOTAL_PRICE
------------- -----------
            9       16500
```

3. Count the number of products_type_id

```
SQL> SELECT COUNT(DISTINCT product_type_id) AS type_count FROM Products;

TYPE_COUNT
----------
         3
```

4. Count the number of distinct products_type_id

```
SQL> SELECT COUNT(DISTINCT product_type_id) AS distinct_type_count FROM Products;

DISTINCT_TYPE_COUNT
-------------------
                  3
```

5. Calculate the average price of the product

```
SQL> SELECT AVG(Price) AS average_price FROM Products;

AVERAGE_PRICE
-------------
   1833.33333
```

6. Calculate the average price of the distinct product

```
SQL> SELECT AVG(DISTINCT Price) AS average_distinct_price FROM Products;

AVERAGE_DISTINCT_PRICE
----------------------
                  2000
```

7. Calculate maximum and minimum price of the product

```
SQL> SELECT MAX(Price) AS maximum_price, MIN(Price) AS minimum_price FROM Products;

MAXIMUM_PRICE MINIMUM_PRICE
------------- -------------
         3000          1000
```

8. Find the count of number of ROWID

```
SQL> SELECT COUNT(ROWID) AS rowid_count FROM Products;

ROWID_COUNT
-----------
          9
```

9. Find maximum and minimum product name

```
SQL> SELECT MAX(Name) AS maximum_name, MIN(Name) AS minimum_name FROM Products;

MAXIMUM_NAME                   MINIMUM_NAME
------------------------------ ------------------------------
Smart Phone                    Automated WM
```

10. Calculate standard deviation of price

```
SQL> SELECT STDDEV(Price) AS price_standard_deviation FROM Products;

PRICE_STANDARD_DEVIATION
------------------------
              661.437828
```

11. Calculate variance of price

```
SQL> SELECT VARIANCE(Price) AS price_variance FROM Products;

PRICE_VARIANCE
--------------
        437500
```

12. Calculate average price group by product_type_id

```
SQL> SELECT product_type_id, AVG(Price) AS average_price FROM Products GROUP BY product_type_id;

PRODUCT_TYPE_ID AVERAGE_PRICE
--------------- -------------
              1          1500
              2          2000
              3          2000
```

13. Calculate Variance on price group by product_type_id

```
SQL> SELECT product_type_id, VARIANCE(Price) AS price_variance FROM Products GROUP BY product_type_id;

PRODUCT_TYPE_ID PRICE_VARIANCE
--------------- --------------
              1         250000
              2        1000000
              3         250000
```

14. Calculate Variance on price group by product_type_id order by Variance

```
SQL> SELECT product_type_id, VARIANCE(Price) AS price_variance
  2  FROM Products
  3  GROUP BY product_type_id
  4  ORDER BY price_variance;

PRODUCT_TYPE_ID PRICE_VARIANCE
--------------- --------------
              1         250000
              3         250000
              2        1000000
```

15. Calculate average price group by product_type_id and having average price greater than 1500.

```
SQL> SELECT product_type_id, AVG(Price) AS average_price
  2  FROM Products
  3  GROUP BY product_type_id
  4  HAVING AVG(Price) > 1500;

PRODUCT_TYPE_ID AVERAGE_PRICE
--------------- -------------
              2          2000
              3          2000
```

16. Calculate average price of the products whose price is less that Rs.2000 and group by product_type_id and having average price greater than 1500.

```
SQL> SELECT product_type_id, AVG(Price) AS average_price
  2  FROM Products
  3  WHERE Price < 2000
  4  GROUP BY product_type_id
  5  HAVING AVG(Price) > 1500;

no rows selected
```

Task-02: Execute the following Aggregate and Group By queries on Employee Table of

Scott username

```
SQL> connect SCOTT/TIGER
Connected.
SQL> spool d:\pract3scott.txt
SQL> SELECT * FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

1. Find the highest sal of EMP table

```
SQL> SELECT MAX(sal) AS highest_salary FROM EMP;

HIGHEST_SALARY
--------------
          5000
```

2. Find details of highest paid employee.

```
SQL> SELECT * FROM EMP WHERE sal = (SELECT MAX(sal) FROM EMP);

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

3. Find the highest paid employee of department 20

```
SQL> SELECT * FROM EMP WHERE deptno = 20 AND sal = (SELECT MAX(sal) FROM EMP WHERE deptno = 20);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

4. Find the total sal given to the MGR

```
SQL> SELECT SUM(sal) AS total_salary FROM EMP WHERE job = 'MANAGER';

TOTAL_SALARY
------------
        8275
```

5. Find the total annual sal to distribute job wise in the year 81.

```
SQL> SELECT job, SUM(sal) AS total_salary FROM EMP WHERE hiredate LIKE '%81' GROUP BY job;

JOB       TOTAL_SALARY
--------- ------------
SALESMAN          5600
CLERK              950
PRESIDENT         5000
MANAGER           8275
ANALYST           3000
```

6. Display the average salaries of all the clerks.

```
SQL> SELECT job, AVG(sal) AS average_salary FROM EMP WHERE job = 'CLERK' GROUP BY job;

JOB       AVERAGE_SALARY
--------- --------------
CLERK             1037.5
```

7. List the employeein dept 20 whose sal is >the average sal 0f dept 10 emps

```
SQL> SELECT e.* FROM EMP e, (SELECT AVG(sal) AS avg_sal FROM EMP WHERE deptno = 10) d WHERE e.deptno = 20 AND e.sal > d.avg_sal;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

8. Display the number of employee for each job group deptno wise

```
SQL> SELECT deptno, job, COUNT(*) AS employee_count FROM EMP GROUP BY deptno, job;

    DEPTNO JOB       EMPLOYEE_COUNT
---------- --------- --------------
        20 CLERK                  2
        30 SALESMAN               4
        20 MANAGER                1
        30 CLERK                  1
        10 PRESIDENT              1
        30 MANAGER                1
        10 CLERK                  1
        10 MANAGER                1
        20 ANALYST                2

9 rows selected.
```

9. List the manage rno and the number of employees working for those mgrs in the ascending Mgrno.

```
SQL> SELECT mgr AS manager_number, COUNT(*) AS employee_count FROM EMP GROUP BY mgr ORDER BY mgr;

MANAGER_NUMBER EMPLOYEE_COUNT
-------------- --------------
          7566              2
          7698              5
          7782              1
          7788              1
          7839              3
          7902              1
                            1

7 rows selected.
```

10. List departmentwise employee count

```
SQL> SELECT deptno, COUNT(*) AS employee_count FROM EMP GROUP BY deptno;

    DEPTNO EMPLOYEE_COUNT
---------- --------------
        30              6
        20              5
        10              3
```

11. List the department, details where at least two emps are working

```
SQL> SELECT deptno, COUNT(*) AS employee_count FROM EMP GROUP BY deptno HAVING COUNT(*) >= 2;

    DEPTNO EMPLOYEE_COUNT
---------- --------------
        30              6
        20              5
        10              3
```

12. List the names of the emps who are getting the highest sal dept wise.

```
SQL> SELECT deptno, ename, sal FROM EMP WHERE (deptno, sal) IN (SELECT deptno, MAX(sal) FROM EMP GROUP BY deptno);

    DEPTNO ENAME            SAL
---------- ---------- ---------
        30 BLAKE           2850
        20 SCOTT           3000
        10 KING            5000
        20 FORD            3000
```

13. List the emps whose sal is greater than or equal to the average of max and minimum

```
SQL> SELECT * FROM EMP WHERE sal >= (SELECT (MAX(sal) + MIN(sal)) / 2 FROM EMP);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
```

14. List the no. of emps in each department where the no. is more than 3.

```
SQL> SELECT deptno, COUNT(*) AS employee_count FROM EMP GROUP BY deptno HAVING COUNT(*) > 3;

    DEPTNO EMPLOYEE_COUNT
---------- --------------
        30              6
        20              5
```

15. Find out how may Managers are their in the company.

```
SQL> SELECT COUNT(*) AS manager_count FROM EMP WHERE job = 'MANAGER';

MANAGER_COUNT
-------------
            3
```

16. Check whether all the emps numbers are indeed unique

```
SQL> SELECT COUNT(*) AS duplicate_count FROM (SELECT empno, COUNT(*) FROM EMP GROUP BY empno HAVING COUNT(*) > 1);

DUPLICATE_COUNT
---------------
              0
```

17. Find all the emps who earn the more than minimum Salary of department 50 in ascending order of deptno.

```
SQL> SELECT * FROM EMP WHERE sal > (SELECT MIN(sal) FROM EMP WHERE deptno = 50) ORDER BY deptno ASC;

no rows selected
```

18. Find out all the emps who earn highest salary in each job type. Sort in descending salary order.

```
SQL> SELECT job, ename, sal FROM EMP e WHERE sal = (SELECT MAX(sal) FROM EMP WHERE job = e.job) ORDER BY sal DESC;

JOB       ENAME             SAL
--------- ---------- ----------
PRESIDENT KING             5000
ANALYST   SCOTT            3000
ANALYST   FORD             3000
MANAGER   JONES            2975
SALESMAN  ALLEN            1600
CLERK     MILLER           1300

6 rows selected.
```

19. List the Deptno where there are no emps.

```
SQL> SELECT DISTINCT deptno FROM EMP WHERE deptno NOT IN (SELECT DISTINCT deptno FROM EMP);

no rows selected
```

20. List the No.of emp's and Avg salary within each department for each job.

```
SQL> SELECT deptno, job, COUNT(*) AS employee_count, AVG(sal) AS average_salary FROM EMP GROUP BY deptno, job;

    DEPTNO JOB       EMPLOYEE_COUNT AVERAGE_SALARY
---------- --------- -------------- --------------
        20 CLERK                  2            950
        30 SALESMAN               4           1400
        20 MANAGER                1           2975
        30 CLERK                  1            950
        10 PRESIDENT              1           5000
        30 MANAGER                1           2850
        10 CLERK                  1           1300
        10 MANAGER                1           2450
        20 ANALYST                2           3000

9 rows selected.
```

21. Find the maximum average salary drawn for each job except for 'President'.

```
SQL> SELECT job, MAX(average_salary) AS max_average_salary FROM (SELECT job, AVG(sal) AS average_salary FROM EMP WHERE job != 'PRESIDENT' GROUP BY job) GROU
P BY job;

JOB       MAX_AVERAGE_SALARY
--------- ------------------
CLERK                 1037.5
SALESMAN                1400
MANAGER           2758.33333
ANALYST                 3000
```

22. List the highest paid emp.

```
SQL> SELECT * FROM EMP WHERE sal = (SELECT MAX(sal) FROM EMP);

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

23. List the details of most recently hired emp of dept 30.

```
SQL> SELECT * FROM EMP WHERE deptno = 30 AND hiredate = (SELECT MAX(hiredate) FROM EMP WHERE deptno = 30);

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
```

24. Find the count of employee, average salary and sum of the salary

```
SQL> SELECT COUNT(*) AS employee_count, AVG(sal) AS average_salary, SUM(sal) AS total_salary FROM EMP;

EMPLOYEE_COUNT AVERAGE_SALARY TOTAL_SALARY
-------------- -------------- ------------
            14     2073.21429        29025
```

25. Find the count of employee, average salary and sum of the salary and group by department number wise in the ascending order

```
SQL> SELECT deptno, COUNT(*) AS employee_count, AVG(sal) AS average_salary, SUM(sal) AS total_salary
  2  FROM EMP
  3  GROUP BY deptno
  4  ORDER BY deptno ASC;

    DEPTNO EMPLOYEE_COUNT AVERAGE_SALARY TOTAL_SALARY
---------- -------------- -------------- ------------
        10              3     2916.66667         8750
        20              5           2175        10875
        30              6     1566.66667         9400
```

26. Find the count of employee, average salary and sum of the salary and group by department number wise and job wise and in the ascending order of dept number and Job.

```
SQL> SELECT deptno, job, COUNT(*) AS employee_count, AVG(sal) AS average_salary, SUM(sal) AS total_salary
  2  FROM EMP
  3  GROUP BY deptno, job
  4  ORDER BY deptno ASC, job ASC;

    DEPTNO JOB       EMPLOYEE_COUNT AVERAGE_SALARY TOTAL_SALARY
---------- --------- -------------- -------------- ------------
        10 CLERK                 1           1300         1300
        10 MANAGER               1           2450         2450
        10 PRESIDENT             1           5000         5000
        20 ANALYST               2           3000         6000
        20 CLERK                 2            950         1900
        20 MANAGER               1           2975         2975
        30 CLERK                 1            950          950
        30 MANAGER               1           2850         2850
        30 SALESMAN              4           1400         5600

9 rows selected.
```