**PRACTICAL - 6**

**Aim: To solve various queries related to grouping and aggregate functions**

**Theoretical Description:**

* **Aggregate Functions**: SUM(), AVG(), MAX(), etc., are used to perform calculations on groups of rows.
* **GROUP BY**: Groups rows that have the same values in specified columns, typically used with aggregate functions.
* **HAVING**: Filters groups based on the result of aggregate functions, applied after GROUP BY.
* **WHERE**: Filters rows before grouping or aggregation.

**Query-1: 1)**

**Insert the following values into product table.**

**SQL Statement:**

CREATE TABLE product (Detorder\_no Varchar(6),Product\_no Varchar(6),Qty\_order INT);

INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19001', 'P00001', 10);

INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19001', 'P00002', 3);

INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19002', 'P00001', 4);

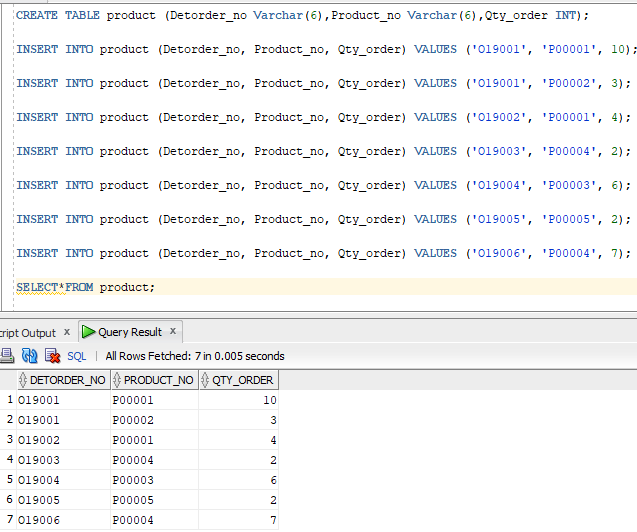
INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19003', 'P00004', 2);

INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19004', 'P00003', 6);

INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19005', 'P00005', 2);

INSERT INTO product (Detorder\_no, Product\_no, Qty\_order) VALUES ('O19006', 'P00004', 7);

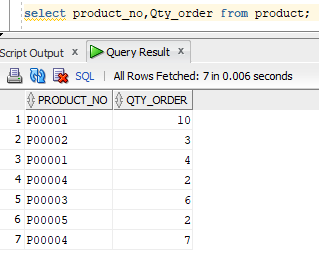
**Output:**

****

**Query-2: Retrieve the product numbers and total quantity ordered for each product from the product table.**

**SQL Statement:** select product\_no,Qty\_order from product;

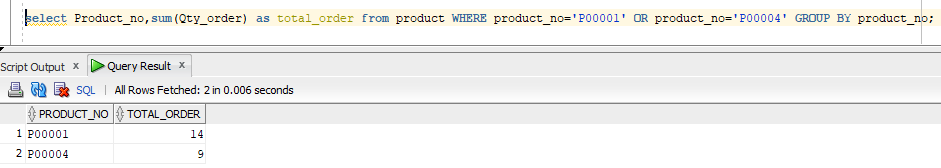
**Output:**

****

**Query-3: Retrieve the product no and the total quantity ordered for product’s ‘P00001’ and ‘P00004’ from product table.**

**SQL Statement:** select Product\_no,sum(Qty\_order) as total\_order from product WHERE product\_no='P00001' OR product\_no='P00004' GROUP BY product\_no;

**Output:**

****

**Query-4: Insert the following values into emp\_company.**

**SQL Statement:**

CREATE TABLE emp\_company (ENAME Varchar(10),CNAME Varchar(10),SALARY INT);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Anil' ,'ACC',1500);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Shankar' ,'TATA',2000);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Jay' ,'WIPRO',1800);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Sunil' ,'WIPRO',1700);

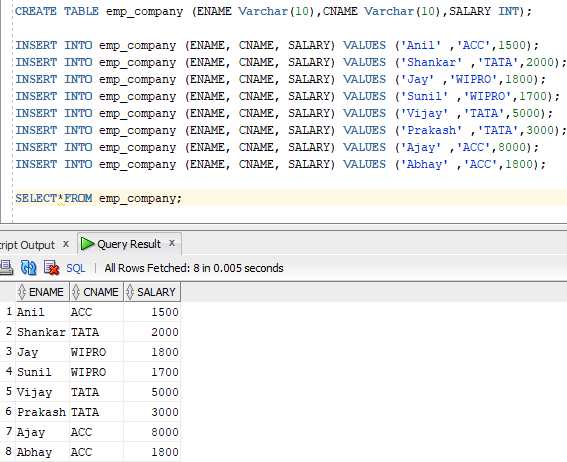
INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Vijay' ,'TATA',5000);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Prakash' ,'TATA',3000);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Ajay' ,'ACC',8000);

INSERT INTO emp\_company (ENAME, CNAME, SALARY) VALUES ('Abhay' ,'ACC',1800);

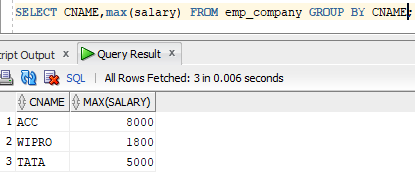
**Output:**

****

**Query-5: List the name of company and maximum salary in that company.**

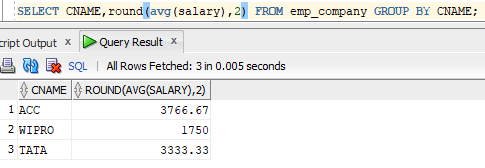
**SQL Statement:** SELECT CNAME,max(salary) FROM emp\_company GROUP BY CNAME;

**Output:**

****

**Query-6: Find out the average salary of each company.**

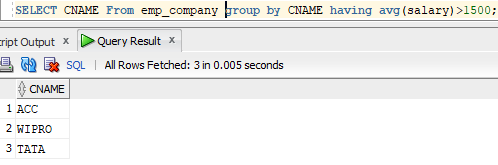
**SQL Statement:** SELECT CNAME,round(avg(salary),2) FROM emp\_company GROUP BY CNAME;

**Output:   
**

**Query-7: Find out the name of companies having average salary more than 1500.**

**SQL Statement:** SELECT CNAME From emp\_company group by CNAME having avg(salary)>1500;

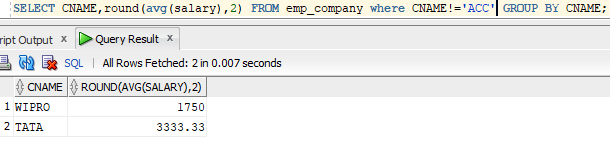
**Output:**

****

**Query-8: Find out the average salary of each company except ‘ACC’.**

**SQL Statement:** SELECT CNAME,round(avg(salary),2) FROM emp\_company where CNAME!='ACC' GROUP BY CNAME;

**Output:**

****

**// [IF any Questions are given to you]**

**Question-1:**

**Answer:**

**Question-2:**

**Answer:**