18CSC303J – Database Management System
Experiment 5

Shushrut Kumar (RA1811028010049)

1. Write a SQL statement to find the total purchase amount of all orders

```
SELECT SUM(ORDER_AMOUNT) AS TOTAL_PURCHASE
FROM ORDERS_049;
```

TOTAL_PURCHASE
49600

Download CSV

2. Write a SQL statement to find the average purchase amount of all orders.

```
SELECT CAST(AVG(ORDER_AMOUNT) AS DECIMAL(10,2)) AS  TOTAL_PURCHASE
FROM ORDERS_049;
```

TOTAL_PURCHASE
4509.09

Download CSV

3. Write a SQL statement to find the number of customers who get at least a gradation for his/her performance.

```
SELECT COUNT(DISTINCT CUSTOMER_ID) AS GRADED_CUSTOMER
FROM ORDERS_049
WHERE GRADE IS NOT NULL;
```

GRADED_CUSTOMER
6

Download CSV

4. Write a SQL statement to get the maximum purchase amount of all the orders

```
SELECT MAX(ORDER_AMOUNT) AS MAX_PURCHASE_AMOUNT
FROM ORDERS_049;
```

MAX_PURCHASE_AMOUNT
10000

Download CSV

5. Write a SQL statement to get the minimum purchase amount of all the orders.

```
SELECT MIN(ORDER_AMOUNT) AS MAX_PURCHASE_AMOUNT
FROM ORDERS_049;
```

MAX_PURCHASE_AMOUNT
1100

Download CSV

6. Write a SQL statement which selects the highest grade for each of the cities of the customers.

```
SELECT CITY, MIN(GRADE) AS HIGHEST_GRADE
FROM ORDERS_049
WHERE CITY IS NOT NULL
GROUP BY CITY;
```

CITY	HIGHEST_GRADE
Chennai	A
Vadodara	B
Mumbai	A

Download CSV

3 rows selected.

7. Write a SQL statement to find the highest purchase amount ordered by each customer with their ID and highest purchase amount.

```
SELECT CUSTOMER_ID, MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY CUSTOMER_ID
ORDER BY CUSTOMER_ID;
```

CUSTOMER_ID	MAX(ORDER_AMOUNT)
3000	1500
3001	1900
3002	2100
3003	10000
3004	5400
3005	6400
3008	8900

Download CSV

7 rows selected.

8. Write a SQL statement to find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.

```
SELECT CUSTOMER_ID, ORDER_DATE, MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY CUSTOMER_ID,ORDER_DATE
ORDER BY ORDER_DATE;
```

CUSTOMER_ID	ORDER_DATE	MAX(ORDER_AMOUNT)
3000	15-AUG-12	1500
3001	16-AUG-12	1900
3002	16-AUG-12	2100
3003	16-AUG-12	3600
3003	17-AUG-12	5400
3004	17-AUG-12	5400
3005	17-AUG-12	6400
3008	17-AUG-12	8900
3003	19-AUG-12	1100
3003	20-AUG-12	10000

Download CSV

10 rows selected.

9. Write a SQL statement to find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.

```
SELECT CUSTOMER_ID, ORDER_DATE, MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY CUSTOMER_ID,ORDER_DATE
ORDER BY ORDER_DATE;
```

CUSTOMER_ID	ORDER_DATE	MAX(ORDER_AMOUNT)
3000	15-AUG-12	1500
3001	16-AUG-12	1900
3002	16-AUG-12	2100
3003	16-AUG-12	3600
3003	17-AUG-12	5400
3004	17-AUG-12	5400
3005	17-AUG-12	6400
3008	17-AUG-12	8900
3003	19-AUG-12	1100
3003	20-AUG-12	10000

Download CSV

10 rows selected.

10. Write a SQL statement to find the highest purchase amount with their ID and order date, for only those customers who have the highest purchase amount in a day is more than 2000.

```
SELECT CUSTOMER_ID, ORDER_DATE, MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY CUSTOMER_ID,ORDER_DATE
HAVING MAX(ORDER_AMOUNT)>2000
ORDER BY ORDER_DATE;
```

CUSTOMER_ID	ORDER_DATE	MAX(ORDER_AMOUNT)
3002	16-AUG-12	2100
3003	16-AUG-12	3600
3003	17-AUG-12	5400
3004	17-AUG-12	5400
3005	17-AUG-12	6400
3008	17-AUG-12	8900
3003	20-AUG-12	10000

Download CSV

7 rows selected.

11. Write a SQL statement to find the highest purchase amount with their ID and order date, for those customers who have a higher purchase amount in a day is within the range 2000 and 6000

```
SELECT CUSTOMER_ID, ORDER_DATE, MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY CUSTOMER_ID,ORDER_DATE
HAVING MAX(ORDER_AMOUNT)>2000 AND MAX(ORDER_AMOUNT)<6000
ORDER BY ORDER_DATE;
```

CUSTOMER_ID	ORDER_DATE	MAX(ORDER_AMOUNT)
3002	16-AUG-12	2100
3003	16-AUG-12	3600
3003	17-AUG-12	5400
3004	17-AUG-12	5400

Download CSV

4 rows selected.

12. Write a SQL statement to find the highest purchase amount with their ID, for only those customers whose ID is within the range 3002 and 3007.

```
SELECT CUSTOMER_ID,  MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY CUSTOMER_ID
HAVING CUSTOMER_ID>=3002 AND CUSTOMER_ID<=3007;
```

CUSTOMER_ID	MAX(ORDER_AMOUNT)
3002	2100
3004	5400
3003	10000
3005	6400

Download CSV

4 rows selected.

13. Write a SQL statement to find the highest purchase amount with their ID, for only those salesmen whose ID is within the range 5003 and 5008.

```
SELECT SALESMAN_ID,  MAX(ORDER_AMOUNT)
FROM ORDERS_049
GROUP BY SALESMAN_ID
HAVING SALESMAN_ID>=5003 AND SALESMAN_ID<=5008;
```

SALESMAN_ID	MAX(ORDER_AMOUNT)
5003	8900
5005	10000

Download CSV

2 rows selected.

14. Write a SQL statement that counts all orders for a date August 17th, 2012

```
SELECT COUNT(ORDER_AMOUNT) AS ORDER_COUNT
FROM ORDERS_049
WHERE ORDER_DATE = TO_DATE('2012-08-17','yyyy-mm-dd');
```

ORDER_COUNT
5

Download CSV

15. Write a SQL statement that counts the number of different non NULL city values for salesmen.

```
SELECT COUNT(DISTINCT CITY) AS CITY_COUNT
FROM ORDERS_049
WHERE CITY IS NOT NULL;
```

CITY_COUNT
3

Download CSV