

## 18CSC303J-Database Management System

### Experiment 5

**Ayush Vaishnav**

**RA1811028010041**

#### **Creating table:**

```
create table orders_041(  
    salesman_id number(4) NOT NULL,  
    city varchar(15),  
    customer_id number(4) NOT NULL,  
    order_date date NOT NULL,  
    order_amount number(8) NOT NULL,  
    grade char);  
desc orders_041;
```

```
INSERT INTO orders_041  
VALUES(5002,NULL,3000,TO_DATE('2012-08-15','yyyy-mm-dd'),1500,NULL);
```

```
INSERT INTO orders_041  
VALUES(5003,'Indore',3001,TO_DATE('2012-08-16','yyyy-mm-dd'),1900,'B');
```

```
INSERT INTO orders_041  
VALUES(5002,'delhi',3002,TO_DATE('2012-08-16','yyyy-mm-dd'),2100,'A');
```

```
INSERT INTO orders_041  
VALUES(5005,'delhi',3003,TO_DATE('2012-08-16','yyyy-mm-dd'),3600,'D');
```

```
INSERT INTO orders_041  
VALUES(5003,'mumbai',3004,TO_DATE('2012-08-17','yyyy-mm-dd'),5400,'A');
```

```
INSERT INTO orders_041  
VALUES(5002,'Indore',3005,TO_DATE('2012-08-17','yyyy-mm-dd'),6400,'C');
```

```
INSERT INTO orders_041
VALUES(5003,'Indore',3003,TO_DATE('2012-08-17','yyyy-mm-dd'),5400,'D');
```

```
INSERT INTO orders_041
VALUES(5003,NULL,3008,TO_DATE('2012-08-17','yyyy-mm-dd'),8900,'C');
```

```
INSERT INTO orders_041
VALUES(5002,'delhi',3003,TO_DATE('2012-08-17','yyyy-mm-dd'),3300,NULL);
```

```
INSERT INTO orders_041
VALUES(5010,'mumbai',3003,TO_DATE('2012-08-19','yyyy-mm-dd'),1100,NULL);
```

```
INSERT INTO orders_041
VALUES(5005,NULL,3003,TO_DATE('2012-08-20','yyyy-mm-dd'),10000,'A');
```

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

```
QUERY-> SELECT SUM(ORDER_AMOUNT) AS TOTAL_PURCHASE
FROM ORDERS_041;
```

TOTAL_PURCHASE
49600

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

```
QUERY-> SELECT CAST(AVG(ORDER_AMOUNT) AS DECIMAL(10,2)) AS
AVG_PURCHASE
FROM ORDERS_041;
```

AVG_PURCHASE
4509.09

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

QUERY-> SELECT COUNT(DISTINCT CUSTOMER\_ID) AS GRADED\_CUSTOMER  
FROM ORDERS\_041  
WHERE GRADE IS NOT NULL;

<b>GRADED_CUSTOMER</b>
6

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

QUERY-> SELECT MAX(ORDER\_AMOUNT) AS MAX\_PURCHASE\_AMOUNT  
FROM ORDERS\_041;

<b>MAX_PURCHASE_AMOUNT</b>
10000

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

QUERY-> SELECT MIN(ORDER\_AMOUNT) AS MIN\_PURCHASE\_AMOUNT  
FROM ORDERS\_041;

<b>MIN_PURCHASE_AMOUNT</b>
1100

**6. Write a SQL statement which QUERY-> SELECTs the highest grade for each of the cities of the customers.**

```
QUERY-> SELECT CITY, MAX(GRADE) AS HIGHEST_GRADE  
FROM ORDERS_041  
WHERE CITY IS NOT NULL  
GROUP BY CITY;
```

CITY	HIGHEST_GRADE
Indore	D
delhi	D
mumbai	A

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

```
QUERY-> SELECT CUSTOMER_ID, MAX(ORDER_AMOUNT)  
FROM ORDERS_041  
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

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

QUERY-> SELECT CUSTOMER\_ID, ORDER\_DATE, MAX(ORDER\_AMOUNT)  
FROM ORDERS\_041  
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

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

```
QUERY-> SELECT CUSTOMER_ID, ORDER_DATE, MAX(ORDER_AMOUNT)
FROM ORDERS_041
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

**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.**

```
QUERY-> SELECT CUSTOMER_ID, ORDER_DATE, MAX(ORDER_AMOUNT)
FROM ORDERS_041
GROUP BY CUSTOMER_ID, ORDER_DATE
HAVING MAX(ORDER_AMOUNT)>2000
ORDER BY ORDER_DATE;
```

<b>CUSTOMER_ID</b>	<b>ORDER_DATE</b>	<b>MAX(ORDER_AMOUNT)</b>
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

**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**

QUERY-> SELECT CUSTOMER\_ID, ORDER\_DATE, MAX(ORDER\_AMOUNT)  
FROM ORDERS\_041  
GROUP BY CUSTOMER\_ID, ORDER\_DATE  
HAVING MAX(ORDER\_AMOUNT) BETWEEN 2000 AND 6000  
ORDER BY ORDER\_DATE;

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

**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.**

```
QUERY-> SELECT CUSTOMER_ID, MAX(ORDER_AMOUNT)
FROM ORDERS_041
GROUP BY CUSTOMER_ID
HAVING CUSTOMER_ID BETWEEN 3002 AND 3007;
```

<b>CUSTOMER_ID</b>	<b>MAX(ORDER_AMOUNT)</b>
3002	2100
3003	10000
3004	5400
3005	6400

**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.**

```
QUERY-> SELECT SALESMAN_ID, MAX(ORDER_AMOUNT)
FROM ORDERS_041
GROUP BY SALESMAN_ID
HAVING SALESMAN_ID BETWEEN 5003 AND 5008;
```

<b>SALESMAN_ID</b>	<b>MAX(ORDER_AMOUNT)</b>
5003	8900
5005	10000

Download CSV



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

```
QUERY-> SELECT COUNT(ORDER_AMOUNT) AS ORDER_COUNT  
FROM ORDERS_041  
WHERE ORDER_DATE = TO_DATE('2012-08-17','yyyy-mm-dd');
```

ORDER_COUNT
5

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

```
QUERY-> SELECT COUNT(DISTINCT CITY) AS CITY_COUNT  
FROM ORDERS_041  
WHERE CITY IS NOT NULL;
```

CITY_COUNT
3