

KIET GROUP OF INSTITUTIONS

DEPARTMENT OF COMPUTER APPLICATIONS

LAB ASSIGNMENT 9

DBMS Lab (KCA – 252)

Assignments on Aggregate Function

PRASSAN VARSHNEY

ROLL_NUMBER – 36

Consider the following table to solve the queries.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

1. write a SQL query to calculate total purchase amount of all orders. Return total purchase amount.
SELECT SUM(purch_amt) AS Total_Purchase_Amount
FROM orders;
2. write a SQL query to calculate average purchase amount of all orders. Return average purchase amount.
SELECT AVG(purch_amt) AS Average_Purchase_Amount
FROM orders;
3. write a SQL query to find the number of salespeople.
SELECT COUNT(DISTINCT salesman_id) AS Number_of_Salespeople
FROM orders;
4. write a SQL query to find the maximum purchase amount.
SELECT MAX(purch_amt) AS Maximum_Purchase_Amount
FROM orders;
5. write a SQL query to find the minimum purchase amount.
SELECT MIN(purch_amt) AS Minimum_Purchase_Amount
FROM orders;
6. write a SQL query to find the highest purchase amount ordered by each customer. Return customer ID, maximum purchase amount.
SELECT customer_id, MAX(purch_amt) AS Maximum_Purchase_Amount
FROM orders
GROUP BY customer_id;

7. write a SQL query to find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.
 SELECT ord_date, MAX(purch_amt) AS Highest_Purchase_Amount
 FROM orders
 GROUP BY ord_date;
8. write a SQL query to find the highest purchase amount on '2012-08-17' by each salesperson. Return salesperson ID, purchase amount.
 SELECT salesman_id, MAX(purch_amt) AS Purchase_Amount
 FROM orders
 WHERE ord_date = '2012-08-17'
 GROUP BY salesman_id;
9. write a SQL query to find highest order (purchase) amount by each customer in a particular order date. Filter the result by highest order (purchase) amount above 2000.00. Return customer id, order date and maximum purchase amount.
 SELECT customer_id, ord_date, MAX(purch_amt) AS Maximum_Purchase_Amount
 FROM orders
 GROUP BY customer_id, ord_date
 HAVING MAX(purch_amt) > 2000.00;
10. write a SQL query to find the maximum order (purchase) amount in the range 2000, 6000 (Begin and end values are included.) by combination of each customer and order date. Return customer id, order date and maximum purchase amount.
 SELECT customer_id, ord_date, MAX(purch_amt) AS Maximum_Purchase_Amount
 FROM orders
 WHERE purch_amt BETWEEN 2000 AND 6000
 GROUP BY customer_id, ord_date;
11. write a SQL query to find the maximum order (purchase) amount by each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Return customer id and maximum purchase amount.
 SELECT customer_id, MAX(purch_amt) AS Maximum_Purchase_Amount
 FROM orders
 WHERE customer_id BETWEEN 3002 AND 3007
 GROUP BY customer_id;
12. write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders
 SELECT COUNT(ord_no) AS Number_of_Orders
 FROM orders
 WHERE ord_date = '2012-08-17';

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

13. write a SQL query to count the number of customers.

```
SELECT COUNT(customer_id) AS Number_of_Customers
FROM customer;
```

14. write a SQL query to find the number of customers who got at least a gradation for his/her activity.

```
SELECT COUNT(customer_id) AS Number_of_Customers_With_Gradation
FROM customer
WHERE grade IS NOT NULL;
```

15. write a SQL query to find the highest grade of the customers for each of the city. Return city, maximum grade.

```
SELECT city, MAX(grade) AS Maximum_Grade
FROM customer
GROUP BY city;
```

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

16. write a SQL query to count number of salespeople who belongs to a city. Return number of salespeople.

```
SELECT city, COUNT(salesman_id) AS Number_of_Salespeople
FROM salesman
GROUP BY city;
```

Sample table: item_mast

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200	15
102	Key Board	450	16
103	ZIP drive	250	14
104	Speaker	550	16
105	Monitor	5000	11
106	DVD drive	900	12
107	CD drive	800	12
108	Printer	2600	13
109	Refill cartridge	350	13
110	Mouse	250	12

17. write a SQL query to count number of products where product price is higher than or equal to 350. Return number of products.

```
SELECT COUNT(pro_id) AS Number_of_Products
```

```
FROM item_mast
WHERE pro_price >= 350;
```

Consider a table named **Employee(Eid, Name, Dept, Salary, DOJ)**
Solve the following queries.

18. Display the latest date on which an employee had joined.

```
SELECT MAX(DOJ) AS Latest_Joining_Date
FROM Employee;
```

19. Display the 1st date on which an employee had joined.

```
SELECT MIN(DOJ) AS First_Joining_Date
FROM Employee;
```

20. List out how many numbers of departments are there.

```
SELECT COUNT(DISTINCT Dept) AS Number_of_Departments
FROM Employee;
```

21. Display how many numbers of employees are there in MCA Department.

```
SELECT COUNT(Eid) AS Number_of_Employees_in_MCA
FROM Employee
WHERE Dept = 'MCA';
```

22. Display department wise maximum average salary.

```
SELECT Dept, MAX(AVG_Salary) AS Maximum_Average_Salary
FROM (
    SELECT Dept, AVG(Salary) AS AVG_Salary
    FROM Employee
    GROUP BY Dept
) AS Dept_Salaries
GROUP BY Dept;
```

23. Display the total number of employees in the organization.

```
SELECT COUNT(Eid) AS Total_Number_of_Employees
FROM Employee;
```

24. Display department wise the numbers of employees working.

```
SELECT Dept, COUNT(Eid) AS Number_of_Employees
FROM Employee
GROUP BY Dept;
```