Orders Table :

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. Calculate total purchase amount of all orders. Return total purchase amount.

- **SQL Command:**

```
SELECT SUM(purch_amt) AS total_purchase_amount
FROM orders
```

total_purchase_amount
17541.18

2. Calculate average purchase amount of all orders. Return average purchase amount.

- **SQL Command:**

```
SELECT AVG(purch_amt) AS average_purchase_amount
FROM orders
```

average_purchase_amount
1461.7650000000000000

3. Count the number of unique salespeople. Return number of salespeople.

- **SQL Command:**

```
SELECT COUNT(DISTINCT(salesman_id)) AS number_of_salespeople
FROM orders
```

number_of_salespeople
6

4. Find the maximum purchase amount.

- **SQL Command:**

```
SELECT MAX(purch_amt)
FROM orders
```

max
5760.00

5. Find the minimum purchase amount.

- **SQL Command:**

```
SELECT MIN(purch_amt)
FROM orders
```

min
65.26

6. Find the highest purchase amount ordered by each customer. Return customer ID, maximum purchase amount.

- **SQL Command:**

```
SELECT customer_id, MAX(purch_amt) AS max_purch_amt
FROM orders
GROUP BY customer_id
ORDER BY customer_id
```

customer_id	max_purch_amt
3001	270.65
3002	5760.00
3003	75.29
3004	1983.43
3005	948.50
3007	2400.60
3008	250.45
3009	2480.40

7. Find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.

- **SQL Command:**

```
SELECT customer_id, ord_date, MAX(purch_amt) AS
highest_purch_amt
FROM orders
GROUP BY ord_date, customer_id
ORDER BY ord_date, customer_id
```

customer_id	ord_date	highest_purch_amt
3002	2012-04-25	3045.60
3008	2012-06-27	250.45
3007	2012-07-27	2400.60
3003	2012-08-17	75.29
3009	2012-08-17	110.50
3001	2012-09-10	270.65
3002	2012-09-10	5760.00
3005	2012-09-10	948.50
3002	2012-10-05	65.26
3005	2012-10-05	150.50
3004	2012-10-10	1983.43
3009	2012-10-10	2480.40

8. Find the highest purchase amount on '2012–08–17' by each salesperson. Return salesperson ID, purchase amount.

- **SQL Command:**

```
SELECT salesman_id, MAX(purch_amt)
FROM orders
WHERE ord_date = '2012-08-17'
GROUP BY salesman_id
```

salesman_id	max
5003	110.50
5007	75.29

9. 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.

- **SQL Command:**

```
SELECT customer_id, ord_date, MAX(purch_amt)
FROM orders
GROUP BY customer_id, ord_date
HAVING MAX(purch_amt) > 2000
ORDER BY ord_date, customer_id
```

customer_id	ord_date	max
3002	2012-04-25	3045.60
3007	2012-07-27	2400.60
3002	2012-09-10	5760.00
3009	2012-10-10	2480.40

10. 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.

- **SQL Command:**

```
SELECT customer_id, ord_date, MAX(purch_amt)
FROM orders
GROUP BY customer_id, ord_date
HAVING MAX(purch_amt) BETWEEN 2000 AND 6000
ORDER BY ord_date, customer_id
```

customer_id	ord_date	max
3002	2012-04-25	3045.60
3007	2012-07-27	2400.60
3002	2012-09-10	5760.00
3009	2012-10-10	2480.40

11. Find the maximum order (purchase) amount by the combination of each customer and order date. Filter the rows for maximum order (purchase) amount is either 2000, 3000, 5760, 6000. Return customer id, order date and maximum purchase amount.

- **SQL Command:**

```
SELECT customer_id, ord_date, MAX(purch_amt)
FROM orders
GROUP BY customer_id, ord_date
HAVING MAX(purch_amt) IN (2000, 3000, 5760, 6000)
ORDER BY ord_date, customer_id
```

customer_id	ord_date	max
3002	2012-09-10	5760.00

12. 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.

- **SQL Command:**

```
SELECT customer_id, MAX(purch_amt)
FROM orders
WHERE customer_id BETWEEN 3002 AND 3007
GROUP BY customer_id
ORDER BY customer_id
```

customer_id	max
3002	5760.00
3003	75.29
3004	1983.43
3005	948.50
3007	2400.60

13. Find the maximum order (purchase) amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Filter the rows for maximum order (purchase) amount is higher than 1000. Return customer id and maximum purchase amount.

- **SQL Command:**

```
SELECT customer_id, MAX(purch_amt)
FROM orders
WHERE customer_id BETWEEN 3002 AND 3007
GROUP BY customer_id
HAVING MAX(purch_amt) > 1000
ORDER BY customer_id
```

customer_id	max
3002	5760.00
3004	1983.43
3007	2400.60

14. Find the maximum order (purchase) amount generated by each salesperson. Filter the rows for the salesperson ID is in the range 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount.

- **SQL Command:**

```
SELECT salesman_id, MAX(purch_amt)
FROM orders
WHERE salesman_id BETWEEN 5003 AND 5008
GROUP BY salesman_id
ORDER BY salesman_id
```

salesman_id	max
5003	2480.40
5005	270.65
5006	1983.43
5007	75.29

15. Count all the orders generated on ‘2012–08–17’. Return number of orders.

- **SQL Command:**

```
SELECT COUNT(*)
FROM orders
WHERE ord_date = ‘2012-08-17’
```

count
2

16. Count number of orders by the combination of each order date and salesperson. Return order date, salesperson id.

- **SQL Command:**

```
SELECT ord_date, salesman_id, COUNT(*)
FROM orders
GROUP BY salesman_id, ord_date
ORDER BY ord_date, salesman_id
```

ord_date	salesman_id	count
2012-04-25	5001	1
2012-06-27	5002	1
2012-07-27	5001	1
2012-08-17	5003	1
2012-08-17	5007	1
2012-09-10	5001	1
2012-09-10	5002	1
2012-09-10	5005	1
2012-10-05	5001	1
2012-10-05	5002	1
2012-10-10	5003	1
2012-10-10	5006	1