```sql
--1. Write a query to display all orders from the orders table issued by the salesman
‘Paul Adman’
select * from orders where salesman_id in (select salesman_id from salesman where
NAME='Paul Adam ' )

--2. Write a query to display all orders for the salesman who belongs to city London.
select * from orders where salesman_id in (select salesman_id from salesman where
city='London' )

--3. Write a query to find all the orders issues against the salesman who may works for
customer whose id is  3007

select * from orders where salesman_id in (select salesman_id from customer where
customer_id=3007)

--4. Write a query to display all the orders which values are greater than the average
order value for 10th October 2012.
select * from orders where purch_amt > (select avg(purch_amt) from orders where
ord_date='10/10/2012' )

--5. Write a query to find all orders attributed to a salesman in New York
select * from orders where salesman_id in (select salesman_id from salesman where
city='New York' )

--6. Write a query to display the commission of all the salesmen servicing customers in
Paris.
select * from salesman where salesman_ID in (select salesman_id from Customer where
city='Paris    ')

select * from Customer
select * from orders
select * from salesman
--7. Write a query to display all the customers whose id is 2001 bellow the salesman ID
of Mc Lyon.
select * from customer where customer_ID in (select salesman_id-2001 from salesman where
NAME='Mc Lyon ')

--8. Write a query to count the customers with grades above New York's average.
select count(customer_id)as NoOfCustomers from customer  where grade > (select AVG(grade)
from customer where city='New York                             ')

--9. Write a query to display all customers with orders on October 5, 2012
select * from customer where customer_ID in (select customer_ID from orders where
ord_date='2012-10-5')

--10. Write a query to find the name and numbers of all salesmen who had more than one
customer.
select * from salesman where salesman_ID in (select salesman_ID as counts from customer
group by salesman_ID having count(customer_ID) >1    )

--11. Write a queries to find all orders with order amounts which are on or above-average
amounts for their customers.
select * from orders a where purch_amt >= (select avg(purch_amt) from orders b where
a.customer_id=b.customer_id)
```

```
--12. Write a query to find the sums of the amounts from the orders table, grouped by
date,
--eliminating all those dates where the sum was not at least 1000.00 above
--the maximum order amount for that date.

select * from Customer
select * from orders
select * from salesman
--13. Write a query to extract the data from the customer table if and
--only if one or more of the customers in the customer table are located in London.
select * from customer where  exists  (select customer_ID from customer where city
='London')

--14. Write a query to find the salesmen who have multiple customers.
select * from salesman
where salesman_ID in (select salesman_ID as counts
from customer
group by salesman_ID having count(customer_ID) >1   )

--15. Write a query that extract the rows of all salesmen who have customers with more
than one orders.
select * from salesman
where salesman_ID in
(select salesman_ID from orders group by (salesman_id) having count(ord_no)>1)

--16. Write a query to display all orders with an amount smaller than the maximum amount
for a customers in London.
select * from orders where purch_amt <(select max(purch_amt) from orders where
customer_id in
(select customer_id from customer where city='London' ) )
select * from orders where purch_amt <
(select max(purch_amt) from orders o,customer c where c.customer_ID=o.customer_id and
city='London')

--17. Write a query to display only those customers whose grade are,
--in fact, higher than every customer in New York.
select * from customer  where grade > all (select grade from customer where
city='New York                             ')


--18. Write a SQL query to display the name of each company,
--price for their most expensive product along with their Name.
select * from company
select * from item_mast
SELECT P.pro_name AS "Product Name",
       P.pro_price AS "Price",
       C.com_name AS "Company"
   FROM item_mast P, company C
   WHERE P.pro_com = C.com_id
     AND P.pro_price =
     (
       SELECT MAX(P.pro_price)
         FROM item_mast P
         WHERE P.pro_com = C.com_id
     );
--19. Write a query in SQL to find the first name and last name of employees
--working for departments which sanction amount is second lowest.
select * from emp_department
```

```sql
select * from emp_details

select e1.emp_fname,e1.emp_lname from emp_details e1 where e1.emp_DPT
in(select DPT_code from emp_department
where DPT_ALLOTMENT=(select min(DPT_ALLOTMENT) from emp_department
where DPT_ALLOTMENT > (select min(DPT_ALLOTMENT) from emp_department)))

--20. Write a query in SQL to find the names of departments where more than two employees
are working
select DPT_Name from emp_department where DPT_code in (select emp_DPT from emp_details
group by emp_DPT having COUNT(emp_IDNO)>2)
--21. Write a SQL statement to know which salesman are working for which customer
select c.cust_name,s.NAME from customer c,salesman s where c.salesman_id =s.salesman_ID

--22. Write a SQL statement to find the list of customers who appointed a salesman
--for their jobs who does not live in the same city where their customer lives,
--and gets a commission is above 12%.
select * from customer c where c.salesman_id in (select s.salesman_id from salesman s
where s.city !=c.city and s.commission >0.12 )

--23. Write a SQL statement to make a list in ascending order for the customer
--who holds a grade less than 300 and works either through a salesman or by own.
select c.cust_name,c.customer_ID,c.grade,c.salesman_id from customer c
Left join salesman s on c.salesman_id=s.salesman_ID where c.grade< 300

--24. Write a SQL statement to make a list in ascending order for the
--salesmen who works either for one or more customer or not yet join under any of the
customers.
SELECT a.cust_name,a.city,a.grade,
b.name AS "Salesman", b.city
FROM customer a
right OUTER JOIN salesman b
ON b.salesman_id=a.salesman_id
ORDER BY b.salesman_id;

--25. Write a SQL statement to make a list for the salesmen who works either
--for one or more customer or not yet join under any of the customers
--who placed either one or more orders or no order to their supplier.
SELECT a.cust_name,a.city,a.grade,
b.name AS "Salesman",
c.ord_no, c.ord_date, c.purch_amt
FROM customer a
RIGHT OUTER JOIN salesman b
ON b.salesman_id=a.salesman_id
RIGHT OUTER JOIN orders c
ON c.customer_id=a.customer_id;

--26. Write a SQL statement to make a report with customer name, city,
--order no, order date, purchase amount for those customers from the
--existing list who placed one or more orders or which orders
--have been placed by the customer who is not on the list
SELECT a.cust_name,a.city, b.ord_no,
b.ord_date,b.purch_amt AS "Order Amount"
FROM customer a
FULL OUTER JOIN orders b
ON a.customer_id=b.customer_id
WHERE a.grade IS NOT NULL;
```

```sql
--27. Write a SQL statement to make a report with customer name, city,
--order no. order date, purchase amount for only those customers on
--the list who must have a grade and placed one or more orders or
--which order(s) have been placed by the customer who is neither in the list nor have a
grade.

--28. Write a SQL query to display the item name, price, and company name of all the
products.
select * from company
select * from item_mast
select it.pro_name,it.pro_price,c.com_name from item_mast it,company c
where c.com_id in (select com_id from company c where c.com_id=it.pro_com)
```