1. Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to same city.

```
SELECT salesman.name ,
customer.cust_name, customer.city
FROM salesman inner join customer
on salesman.city=customer.city;
```

2. Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.

```
SELECT  orders.ord_no,orders.purch_amt,
customer.cust_name,customer.city
FROM quiz3.orders inner join quiz3.customer
on customer.customer_id= customer.customer_id
AND orders.purch_amt BETWEEN 500 AND 2000;
```

3. Write a SQL statement to know which salesman are working for which customer.

```
SELECT customer.cust_name,
salesman.city, salesman.name ,salesman.commission
FROM quiz3.customer
INNER JOIN quiz3.salesman
ON salesman.salesman_id=customer.salesman_id
```

4. Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.

```
SELECT customer.cust_name, customer.city, salesman.name, salesman.commission
FROM quiz3.customer
```

```
INNER JOIN quiz3.salesman

ON salesman.salesman_id=customer.salesman_id

WHERE salesman.commission>.12;
```

5. Write a SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in same city where there customer lives, and gets a commission is above 12%.

```
SELECT customer.cust_name ,

customer.city, salesman.name , salesman.city,salesman.commission

FROM quiz3.customer

INNER JOIN quiz3.salesman

ON salesman.salesman_id=customer.salesman_id

WHERE salesman.commission>.12

AND salesman.city<>customer.city;
```

6. Write a SQL statement to find the details of a order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.

```
SELECT a.ord_no,a.ord_date,a.purch_amt,

b.cust_name AS "Customer Name", b.grade,

c.name AS "Salesman", c.commission

FROM orders a

INNER JOIN customer b

ON a.customer_id=b.customer_id

INNER JOIN salesman c

ON a.salesman_id=c.salesman_id;
```

7. Write a SQL statement to make a join within the tables salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come.

```
ELECT *
FROM orders
NATURAL JOIN customer
NATURAL JOIN salesman;
```

8. Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.

```
SELECT a.cust_name,a.city,a.grade,
b.name AS "Salesman",b.city
FROM customer a
LEFT JOIN salesman b
ON a.salesman_id=b.salesman_id
order by a.customer_id;
```

9. 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 a.cust_name,a.city,a.grade,
b.name AS "Salesman", b.city
FROM customer a
LEFT OUTER JOIN salesman b
ON a.salesman_id=b.salesman_id
WHERE a.grade<300
ORDER BY a.customer_id;
```

10. Write a SQL statement to make a report with customer name, city, order number, order date and order amount in ascending order according to the order date to find that either any of the existing customer have placed no order or placed one or more orders.

```
SELECT a.cust_name,a.city, b.ord_no,
b.ord_date,b.purch_amt AS "Order Amount"
FROM customer a
LEFT OUTER JOIN orders b
ON a.customer_id=b.customer_id
order by b.ord_date;
```

11. Write a SQL statement to make a report with customer name, city, order number, order date, order amount salesman name and commission to find that either any of the existing customer have placed no order or placed one or more orders by their salesman or by own.

```
SELECT a.cust_name,a.city, b.ord_no,
b.ord_date,b.purch_amt AS "Order Amount",
c.name,c.commission
FROM customer a
LEFT OUTER JOIN orders b
ON a.customer_id=b.customer_id
LEFT OUTER JOIN salesman c
ON c.salesman_id=b.salesman_id;
```

12. 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 customer.

```
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;
```

13. 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 customer 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;
```

14. Write a SQL statement to make a list for the salesmen who either work for one or more customer or yet to join any of the customer. The customer, may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated 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
WHERE c.purch_amt>=2000
AND a.grade IS NOT NULL;
```

15. 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 order(s) have been placed by the customer who are not in 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;
```

16. Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for only those customers in the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who are neither in the list not have a grade.

```
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;
```

17. write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa

```
SELECT *
FROM salesman a
CROSS JOIN customer b;
```

18. Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa for those customer who belongs to a city.

```
ELECT *
FROM salesman a
CROSS JOIN customer b
WHERE a.city IS NOT NULL;
```

19. Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa for those salesmen who belongs to a city and the customers who must have a grade.

```
SELECT *
FROM salesman a
CROSS JOIN  customer b
WHERE a.city IS NOT NULL
AND b.grade IS NOT NULL;
```

20. Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa for those salesmen who must belongs a city which is not the same as his customer and the customers should have a own grade.

```
SELECT *
FROM salesman a
```

```
CROSS JOIN customer b
WHERE a.city IS NOT NULL
AND b.grade IS NOT NULL
AND  a.city<>b.city;
```