Assignment - 2

1. write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city

Query:

```
SELECT s.name AS Salesman, c.cust_name, c.city
FROM salesman as s
JOIN customer as c
ON s.city = c.city;
```

	Salesman	cust_name	city
1	John Doe	Jack Smith	New York
2	Jane Doe	Maria Garcia	Los Angeles
3	Bob Smith	Tom Johnson	Chicago
4	Alice Chen	Sarah Davis	Houston
5	Tom Brown	David Brown	Miami
6	Sarah Lee	Emily Lee	Seattle
7	Tim Davis	John Kim	Dallas
8	Rachel Kim	Rachel Chen	San Francisco
9	David Wong	Tim Wong	Boston
10	Emily Chen	Alice Davis	Atlanta

2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city

Query:

```
select o.ord_no, o.purch_amt, c.cust_name, c.city
from orders as o
join customer as c
ON o.customer_id = c.customer_id
where purch_amt between 500 and 2000;
```

	ord_no	purch_amt	cust_name	city
1	1	1000.00	David Brown	Miami
2	2	2000.00	Sarah Davis	Houston
3	3	1500.00	Tim Wong	Boston
4	4	750.00	Tom Johnson	Chicago
5	5	1250.00	Maria Garcia	Los Angeles
6	6	500.00	Emily Lee	Seattle
7	7	1750.00	Jack Smith	New York
8	9	1750.00	Rachel Chen	San Francisco
9	10	1250.00	Alice Davis	Atlanta

3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission

Query:

```
select c.cust_name, c.city, s.name, s.commission
from customer as c
join salesman as s
on c.salesman_id = s.salesman_id
```

	cust_name	city	name	commission
1	Jack Smith	New York	Bob Smith	13
2	Maria Garcia	Los Angeles	Tim Davis	20
3	Tom Johnson	Chicago	Jane Doe	12
4	Sarah Davis	Houston	Sarah Lee	18
5	David Brown	Miami	John Doe	16
6	Emily Lee	Seattle	Tom Brown	11
7	John Kim	Dallas	Rachel Kim	9
8	Rachel Chen	San Francisco	David Wong	10
9	Tim Wong	Boston	Alice Chen	9
10	Alice Davis	Atlanta	Emily Chen	11

4. write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, and commission.

Query:

```
SELECT s.name AS "Salesman", s.commission, c.cust_name AS "Customer
Name", c.city
FROM salesman as s
JOIN customer as c
ON s.salesman_id = c.salesman_id
WHERE s.commission > 12;
```

	Salesman	commission	Customer Name	city
1	Bob Smith	13	Jack Smith	New York
2	Tim Davis	20	Maria Garcia	Los Angeles
3	Sarah Lee	18	Sarah Davis	Houston
4	John Doe	16	David Brown	Miami

5. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

Query:

```
select c.cust_name as "Customer Name", c.city as "Customer City",
s.name as "Salesman", s.city as "Salesman city", s.commission
from customer as c
join salesman as s
on s.salesman_id = c.salesman_id
where s.city != c.city and s.commission > 12
```

	Customer Name	Customer City	Salesman	Salesman city	commission
1	Jack Smith	New York	Bob Smith	Chicago	13
2	Maria Garcia	Los Angeles	Tim Davis	Dallas	20
3	Sarah Davis	Houston	Sarah Lee	Seattle	18
4	David Brown	Miami	John Doe	New York	16

6. write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission

Query:

```
select o.ord_no, o.ord_date, o.purch_amt, c.cust_name as "Customer
Name", c.city, s.name as "Salesman", s.commission
from orders as o
join customer as c
on c.customer_id = o.customer_id
join salesman as s
on s.salesman_id = o.salesman_id
```

	ord_no	ord_date	purch_amt	Customer Name	city	Salesman	commission
1	1	2020-01-01	1000.00	David Brown	Miami	John Doe	16
2	2	2020-02-01	2000.00	Sarah Davis	Houston	Sarah Lee	18
3	3	2020-03-01	1500.00	Tim Wong	Boston	Alice Chen	9
4	4	2020-04-01	750.00	Tom Johnson	Chicago	Jane Doe	12
5	5	2020-05-01	1250.00	Maria Garcia	Los Angeles	Tim Davis	20
6	6	2020-06-01	500.00	Emily Lee	Seattle	Tom Brown	11
7	7	2020-07-01	1750.00	Jack Smith	New York	Bob Smith	13
8	8	2020-08-01	2250.00	John Kim	Dallas	Rachel Kim	9
9	9	2020-09-01	1750.00	Rachel Chen	San Francisco	David Wong	10
10	10	2020-10-01	1250.00	Alice Davis	Atlanta	Emily Chen	11

7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

Query:

```
select * from salesman as s
join customer as c
on c.salesman_id = s.salesman_id
join orders as o
on o.customer_id = c.customer_id;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id	ord_no	purch_amt	ord_date	customer_id	salesman_id
1	3	Bob Smith	Chicago	13	1	Jack Smith	New York	200	3	7	1750.00	2020-07-01	1	3
2	7	Tim Davis	Dallas	20	2	Maria Garcia	Los Angeles	500	7	5	1250.00	2020-05-01	2	7
3	2	Jane Doe	Los Angeles	12	3	Tom Johnson	Chicago	770	2	4	750.00	2020-04-01	3	2
4	6	Sarah Lee	Seattle	18	4	Sarah Davis	Houston	900	6	2	2000.00	2020-02-01	4	6
5	1	John Doe	New York	16	5	David Brown	Miami	840	1	1	1000.00	2020-01-01	5	1
6	5	Tom Brown	Miami	11	6	Emily Lee	Seattle	1030	5	6	500.00	2020-06-01	6	5
7	8	Rachel Kim	San Francisco	9	7	John Kim	Dallas	650	8	8	2250.00	2020-08-01	7	8
8	9	David Wong	Boston	10	8	Rachel Chen	San Francisco	350	9	9	1750.00	2020-09-01	8	9
9	4	Alice Chen	Houston	9	9	Tim Wong	Boston	360	4	3	1500.00	2020-03-01	9	4
10	10	Emily Chen	Atlanta	11	10	Alice Davis	Atlanta	740	10	10	1250.00	2020-10-01	10	10

8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.

Query :

```
select c.cust_name, c.city, grade, s.name, s.city
from customer as c
join salesman as s
on s.salesman_id = c.salesman_id
order by c.customer_id ASC;
```

	cust_name	city	grade	name	city
1	Jack Smith	New York	200	Bob Smith	Chicago
2	Maria Garcia	Los Angeles	500	Tim Davis	Dallas
3	Tom Johnson	Chicago	770	Jane Doe	Los Angeles
4	Sarah Davis	Houston	900	Sarah Lee	Seattle
5	David Brown	Miami	840	John Doe	New York
6	Emily Lee	Seattle	1030	Tom Brown	Miami
7	John Kim	Dallas	650	Rachel Kim	San Francisco
8	Rachel Chen	San Francisco	350	David Wong	Boston
9	Tim Wong	Boston	360	Alice Chen	Houston
10	Alice Davis	Atlanta	740	Emily Chen	Atlanta

9. write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id.

Query:

```
select c.cust_name, c.city, grade, s.name, s.city
from customer as c
join salesman as s
on s.salesman_id = c.salesman_id
where grade < 300
order by c.customer_id;
```

	cust_name	city	grade	name	city
1	Jack Smith	New York	200	Bob Smith	Chicago

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 determine whether any of the existing customers have placed an order or not

Query:

```
select c.cust_name, c.city, o.ord_date, o.purch_amt
from customer as c
join orders as o
on c.customer_id = o.customer_id
order by o.ord_date;
```

	cust_name	city	ord_date	purch_amt
1	David Brown	Miami	2020-01-01	1000.00
2	Sarah Davis	Houston	2020-02-01	2000.00
3	Tim Wong	Boston	2020-03-01	1500.00
4	Tom Johnson	Chicago	2020-04-01	750.00
5	Maria Garcia	Los Angeles	2020-05-01	1250.00
6	Emily Lee	Seattle	2020-06-01	500.00
7	Jack Smith	New York	2020-07-01	1750.00
8	John Kim	Dallas	2020-08-01	2250.00
9	Rachel Chen	San Francisco	2020-09-01	1750.00
10	Alice Davis	Atlanta	2020-10-01	1250.00

11. Write a SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves

Query:

```
select c.cust_name, c.city, o.customer_id, o.ord_date, o.purch_amt,
s.name, s.commission
from orders as o
left join customer as c
on c.customer_id = o.customer_id
left join salesman as s
on s.salesman_id = o.salesman_id
```

	cust_name	city	customer_id	ord_date	purch_amt	name	commission
1	David Brown	Miami	5	2020-01-01	1000.00	John Doe	16
2	Sarah Davis	Houston	4	2020-02-01	2000.00	Sarah Lee	18
3	Tim Wong	Boston	9	2020-03-01	1500.00	Alice Chen	9
4	Tom Johnson	Chicago	3	2020-04-01	750.00	Jane Doe	12
5	Maria Garcia	Los Angeles	2	2020-05-01	1250.00	Tim Davis	20
6	Emily Lee	Seattle	6	2020-06-01	500.00	Tom Brown	11
7	Jack Smith	New York	1	2020-07-01	1750.00	Bob Smith	13
8	John Kim	Dallas	7	2020-08-01	2250.00	Rachel Kim	9
9	Rachel Chen	San Francisco	8	2020-09-01	1750.00	David Wong	10
10	Alice Davis	Atlanta	10	2020-10-01	1250.00	Emily Chen	11

12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers

Query :

```
select s.salesman_id, s.name, s.city, s.commission
from salesman as s
left join customer as c
on s.salesman_id = c.salesman_id
order by s.salesman_id;
```

	salesman_id	name	city	commission
1	1	John Doe	New York	16
2	2	Jane Doe	Los Angeles	12
3	3	Bob Smith	Chicago	13
4	4	Alice Chen	Houston	9
5	5	Tom Brown	Miami	11
6	6	Sarah Lee	Seattle	18
7	7	Tim Davis	Dallas	20
8	8	Rachel Kim	San Francisco	9
9	9	David Wong	Boston	10
10	10	Emily Chen	Atlanta	11

13. write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

Query:

```
select s.name, c.cust_name, c.city, c.grade, o.ord_no, o.ord_date,
o.purch_amt
from customer as c
left join salesman as s
on s.salesman_id = c.salesman_id
left join orders as o
on c.customer_id = o.customer_id
```

	name	cust_name	city	grade	ord_no	ord_date	purch_amt
1	Bob Smith	Jack Smith	New York	200	7	2020-07-01	1750.00
2	Tim Davis	Maria Garcia	Los Angeles	500	5	2020-05-01	1250.00
3	Jane Doe	Tom Johnson	Chicago	770	4	2020-04-01	750.00
4	Sarah Lee	Sarah Davis	Houston	900	2	2020-02-01	2000.00
5	John Doe	David Brown	Miami	840	1	2020-01-01	1000.00
6	Tom Brown	Emily Lee	Seattle	1030	6	2020-06-01	500.00
7	Rachel Kim	John Kim	Dallas	650	8	2020-08-01	2250.00
8	David Wong	Rachel Chen	San Francisco	350	9	2020-09-01	1750.00
9	Alice Chen	Tim Wong	Boston	360	3	2020-03-01	1500.00
10	Emily Chen	Alice Davis	Atlanta	740	10	2020-10-01	1250.00

14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customers. 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.

Query:

```
select s.name, c.cust_name, c.city, c.grade, o.ord_no, o.ord_date,
o.purch_amt
from customer as c
left join salesman as s
on s.salesman_id = c.salesman_id
left join orders as o
on c.customer_id = o.customer_id
where (o.purch_amt >= 2000) and (c.grade is not NULL or o.ord_no is
not NULL);
```

	name	cust_name	city	grade	ord_no	ord_date	purch_amt
1	Sarah Lee	Sarah Davis	Houston	900	2	2020-02-01	2000.00
2	Rachel Kim	John Kim	Dallas	650	8	2020-08-01	2250.00

15. Write a SQL statement to generate a list of all the salesmen who either work for one or more customers or have yet to join any of them. The customer may have placed one or more orders at or above order amount 2000, and must have a grade, or he may not have placed any orders to the associated supplier.

Query:

```
select s.name, c.cust_name, c.city, c.grade, o.ord_no, o.ord_date,
o.purch_amt
from customer as c
join salesman as s
on s.salesman_id = c.salesman_id
join orders as o
on c.customer_id = o.customer_id
where (o.purch_amt >= 2000) and (c.grade is not NULL or o.ord_no is
not NULL);
```

	name	cust_name	city	grade	ord_no	ord_date	purch_amt
1	Sarah Lee	Sarah Davis	Houston	900	2	2020-02-01	2000.00
2	Rachel Kim	John Kim	Dallas	650	8	2020-08-01	2250.00

16. Write a SQL statement to generate a report with the 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 neither is on the list nor has a grade.

Query:

```
SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt
FROM customer as c
LEFT JOIN orders as o
ON c.customer_id = o.customer_id
WHERE (c.grade IS NOT NULL AND o.ord_no IS NOT NULL)
OR (c.cust_name IS NULL AND c.grade IS NULL);
```

	cust_name	city	ord_no	ord_date	purch_amt
1	David Brown	Miami	1	2020-01-01	1000.00
2	Sarah Davis	Houston	2	2020-02-01	2000.00
3	Tim Wong	Boston	3	2020-03-01	1500.00
4	Tom Johnson	Chicago	4	2020-04-01	750.00
5	Maria Garcia	Los Angeles	5	2020-05-01	1250.00
6	Emily Lee	Seattle	6	2020-06-01	500.00
7	Jack Smith	New York	7	2020-07-01	1750.00
8	John Kim	Dallas	8	2020-08-01	2250.00
9	Rachel Chen	San Francisco	9	2020-09-01	1750.00
10	Alice Davis	Atlanta	10	2020-10-01	1250.00

17. Write a SQL query to combine each row of the salesman table with each row of the customer table

Query:

```
select *
from salesman as s
cross join customer as c
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	1	John Doe	New York	16	1	Jack Smith	New York	200	3
2	2	Jane Doe	Los Angeles	12	1	Jack Smith	New York	200	3
3	3	Bob Smith	Chicago	13	1	Jack Smith	New York	200	3
4	4	Alice Chen	Houston	9	1	Jack Smith	New York	200	3
5	5	Tom Brown	Miami	11	1	Jack Smith	New York	200	3
6	6	Sarah Lee	Seattle	18	1	Jack Smith	New York	200	3
7	7	Tim Davis	Dallas	20	1	Jack Smith	New York	200	3
8	8	Rachel Kim	San Francisco	9	1	Jack Smith	New York	200	3
9	9	David Wong	Boston	10	1	Jack Smith	New York	200	3
10	10	Emily Chen	Atlanta	11	1	Jack Smith	New York	200	3
11	1	John Doe	New York	16	2	Maria Garcia	Los Angeles	500	7
12	2	Jane Doe	Los Angeles	12	2	Maria Garcia	Los Angeles	500	7
13	3	Bob Smith	Chicago	13	2	Maria Garcia	Los Angeles	500	7
14	4	Alice Chen	Houston	9	2	Maria Garcia	Los Angeles	500	7
15	5	Tom Brown	Miami	11	2	Maria Garcia	Los Angeles	500	7
16	6	Sarah Lee	Seattle	18	2	Maria Garcia	Los Angeles	500	7

18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city

Query:

```
select *
from salesman as s
cross join customer as c
where c.city = s.city;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	1	John Doe	New York	16	1	Jack Smith	New York	200	3
2	2	Jane Doe	Los Angeles	12	2	Maria Garcia	Los Angeles	500	7
3	3	Bob Smith	Chicago	13	3	Tom Johnson	Chicago	770	2
4	4	Alice Chen	Houston	9	4	Sarah Davis	Houston	900	6
5	5	Tom Brown	Miami	11	5	David Brown	Miami	840	1
6	6	Sarah Lee	Seattle	18	6	Emily Lee	Seattle	1030	5
7	7	Tim Davis	Dallas	20	7	John Kim	Dallas	650	8
8	8	Rachel Kim	San Francisco	9	8	Rachel Chen	San Francisco	350	9
9	9	David Wong	Boston	10	9	Tim Wong	Boston	360	4
10	10	Emily Chen	Atlanta	11	10	Alice Davis	Atlanta	740	10

19. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for every customer and vice versa for those salesmen who belong to a city and customers who require a grade

Query:

```
select *
from salesman as s
cross join customer as c
where c.city = s.city AND c.grade is not NULL;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	1	John Doe	New York	16	1	Jack Smith	New York	200	3
2	2	Jane Doe	Los Angeles	12	2	Maria Garcia	Los Angeles	500	7
3	3	Bob Smith	Chicago	13	3	Tom Johnson	Chicago	770	2
4	4	Alice Chen	Houston	9	4	Sarah Davis	Houston	900	6
5	5	Tom Brown	Miami	11	5	David Brown	Miami	840	1
6	6	Sarah Lee	Seattle	18	6	Emily Lee	Seattle	1030	5
7	7	Tim Davis	Dallas	20	7	John Kim	Dallas	650	8
8	8	Rachel Kim	San Francisco	9	8	Rachel Chen	San Francisco	350	9
9	9	David Wong	Boston	10	9	Tim Wong	Boston	360	4
10	10	Emily Chen	Atlanta	11	10	Alice Davis	Atlanta	740	10

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

Query:

```
select *
from salesman as s
cross join customer as c
where c.city != s.city AND c.grade is not NULL;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	2	Jane Doe	Los Angeles	12	1	Jack Smith	New York	200	3
2	3	Bob Smith	Chicago	13	1	Jack Smith	New York	200	3
3	4	Alice Chen	Houston	9	1	Jack Smith	New York	200	3
4	5	Tom Brown	Miami	11	1	Jack Smith	New York	200	3
5	6	Sarah Lee	Seattle	18	1	Jack Smith	New York	200	3
6	7	Tim Davis	Dallas	20	1	Jack Smith	New York	200	3
7	8	Rachel Kim	San Francisco	9	1	Jack Smith	New York	200	3
8	9	David Wong	Boston	10	1	Jack Smith	New York	200	3
9	10	Emily Chen	Atlanta	11	1	Jack Smith	New York	200	3
10	1	John Doe	New York	16	2	Maria Garcia	Los Angeles	500	7
11	3	Bob Smith	Chicago	13	2	Maria Garcia	Los Angeles	500	7
12	4	Alice Chen	Houston	9	2	Maria Garcia	Los Angeles	500	7
13	5	Tom Brown	Miami	11	2	Maria Garcia	Los Angeles	500	7
14	6	Sarah Lee	Seattle	18	2	Maria Garcia	Los Angeles	500	7
15	7	Tim Davis	Dallas	20	2	Maria Garcia	Los Angeles	500	7
16	8	Rachel Kim	San Francisco	9	2	Maria Garcia	Los Angeles	500	7
17	9	David Wong	Boston	10	2	Maria Garcia	Los Angeles	500	7

Query executed successfully.