```
create table customer(
test(# customer_id int,
test(# cust_name varchar(50),
test(# city varchar(50),
test(# grade int,
test(# salesman_id int);
CREATE TABLE

test=# insert into customer values(3002,'Nick Rimandi','New York',100,5001);
INSERT 0 1
test=# insert into customer values(3007,'Brad Davis','New York',200,5001);
INSERT 0 1
test=# insert into customer values(3005,'Graham Zusi','California',200,5002);
INSERT 0 1
test=# insert into customer values(3008,'Julian Green','London',300,5002);
INSERT 0 1
test=# insert into customer values(3004,'Fabian Johnson','Paris',300,5006);
INSERT 0 1
test=# insert into customer values(3009,'Geoff Cameron','Berlin',100,5003);
INSERT 0 1
test=# insert into customer values(3003,'Jozy Altidor','Moscow',200,5007);
INSERT 0 1
insert into customer(customer_id,cust_name,city,salesman_id) values(3001,'Brad
Guzan','London',5005);
INSERT 0 1

test=# select * from customer;
 customer_id |   cust_name    |    city    | grade | salesman_id
-------------+----------------+------------+-------+-------------
        3002 | Nick Rimandi   | 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
(8 rows)

----------------------------------------------------------------------------------------------------------------------------
test=# create table salesman(salesman_id int, name varchar(30), city varchar(20), commission
float);
test=# insert into salesman values (5001,'James Hoog','New York',0.15);
INSERT 0 1
test=# insert into salesman values (5002,'Nail Knite','Paris',0.13);
INSERT 0 1
test=# insert into salesman values (5005,'Pit Alex','London',0.11);
INSERT 0 1
test=# insert into salesman values (5006,'Mc Lyon','Paris',0.14);
INSERT 0 1
test=# insert into salesman values (5007,'Paul Adam','Rome',0.13);
```

```
INSERT 0 1
test=# insert into salesman values (5003,'Lauson Hen','San Jose',0.12);
INSERT 0 1
test=# select * from 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
(6 rows)
--------------------------------------------------------------------------------------------------------------------------
```

1. From the following tables write a SQL query to find the salesperson and customer who belongs to same city. Return Salesman, cust_name and city

```
test=# select salesman.name as "salesman",
test-# customer.cust_name,customer.city
test-# from salesman,customer
test-# where salesman.city = customer.city;
  salesman  |   cust_name    |   city
------------+----------------+----------
 Pit Alex   | Brad Guzan     | London
 James Hoog | Nick Rimando   | New York
 Mc Lyon    | Fabian Johnson | Paris
 Nail Knite | Fabian Johnson | Paris
 James Hoog | Brad Davis     | New York
 Pit Alex   | Julian Green   | London
(6 rows)
--------------------------------------------------------------------------------------------------------------------------
```

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

```
test=# select o.ord_no,o.purch_amt,c.cust_name,c.city from orders o,customer c where o.customer_id = c.customer_id and o.purch_amt between 500 and 2000;
 ord_no | purch_amt |   cust_name    |    city
--------+-----------+----------------+------------
  70010 |   1983.43 | Fabian Johnson | Paris
  70007 |     948.5 | Graham Zusi    | California
(2 rows)
--------------------------------------------------------------------------------------------------------------------------
```

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

```
test=# select c.cust_name AS "Customer Name",c.city,s.name as "Salesman",s.commission from customer c INNER JOIN salesman s ON c.salesman_id = s.salesman_id;
 Customer Name  |    city    |  Salesman  | commission
----------------+------------+------------+------------
```

```
 Brad Guzan     | London     | Pit Alex   |       0.11
 Nick Rimando   | New York   | James Hoog |       0.15
 Jozy Altidor   | Moscow     | Paul Adam  |       0.13
 Fabian Johnson | Paris      | Mc Lyon    |       0.14
 Graham Zusi    | California | Nail Knite |       0.13
 Brad Davis     | New York   | James Hoog |       0.15
 Julian Green   | London     | Nail Knite |       0.13
 Geoff Cameron  | Berlin     | Lauson Hen |       0.12
(8 rows)
```

----

4. From the following tables write a SQL query to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, commission.

```
test=# select c.cust_name AS "Customer Name", c.city, s.name AS "Salesman",s.commission from
customer c Inner Join salesman s on c.salesman_id = s.salesman_id where s.commission>.12;
 Customer Name  |    city    |  Salesman  | commission
----------------+------------+------------+------------
 Nick Rimando   | New York   | James Hoog |       0.15
 Jozy Altidor   | Moscow     | Paul Adam  |       0.13
 Fabian Johnson | Paris      | Mc Lyon    |       0.14
 Graham Zusi    | California | Nail Knite |       0.13
 Brad Davis     | New York   | James Hoog |       0.15
 Julian Green   | London     | Nail Knite |       0.13
(6 rows)
```

----

5. From the following tables write a SQL query to find those salespersons do not live in the same city where their customers live and received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, salesman city, commission.

```
test=# select c.cust_name AS "Customer Name",c.city,s.name AS "SalesMan", s.city,s.commission
from customer c INNER JOIN salesman s ON c.salesman_id = s.salesman_id WHERE
s.commission>.12 And c.city <> s.city;
 Customer Name |    city    |  SalesMan  | city  | commission
---------------+------------+------------+-------+------------
 Jozy Altidor  | Moscow     | Paul Adam  | Rome  |       0.13
 Graham Zusi   | California | Nail Knite | Paris |       0.13
 Julian Green  | London     | Nail Knite | Paris |       0.13
(3 rows)
```

----

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

```
test=# create table orders(ord_no integer, purch_amt float, ord_date date, customer_id integer,
salesman_id integer);
CREATE TABLE
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70001,150.5,'2012-10-05',3005,5002);
INSERT 0 1
```

```
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70009,270.65,'2012-09-10',3001,5005   );
test(# );
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70002,65.26,'2012-10-05',3002,5001);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70004,110.5,'2012-08-17',3009,5003);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70007,948.5,'2012-09-10',3005,5002);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70005,2400.6,'2012-07-27',3007,5001);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70008,5760,'2012-09-10',3002,5001);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70010,1983.43,'2012-10-10',3004,5006);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70003,2480.4,'2012-10-10',3009,5003);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70012,250.45,'2012-06-27',3008,5002);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70011,75.29,'2012-08-17',3003,5007);
INSERT 0 1
test=# insert into orders(ord_no,purch_amt,ord_date,customer_id,salesman_id) values
(70013,3045.6,'2012-04-25',3002,5001);
INSERT 0 1

test=# select * from 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
(12 rows)
```

```
test=# select o.ord_no,o.ord_date,o.purch_amt,c.cust_name AS "Customer Name",c.grade,s.name
AS "Salesman", s.commission from orders o INNER JOIN customer c ON o.customer_id =
c.customer_id INNER JOIN salesman s ON o.salesman_id = s.salesman_id;
 ord_no |  ord_date  | purch_amt | Customer Name  | grade |  Salesman  | commission
--------+------------+-----------+----------------+-------+------------+------------
  70001 | 2012-10-05 |     150.5 | Graham Zusi    |   200 | Nail Knite |       0.13
  70009 | 2012-09-10 |    270.65 | Brad Guzan     |       | Pit Alex   |       0.11
  70002 | 2012-10-05 |     65.26 | Nick Rimando   |   100 | James Hoog |       0.15
  70004 | 2012-08-17 |     110.5 | Geoff Cameron  |   100 | Lauson Hen |       0.12
  70007 | 2012-09-10 |     948.5 | Graham Zusi    |   200 | Nail Knite |       0.13
  70005 | 2012-07-27 |    2400.6 | Brad Davis     |   200 | James Hoog |       0.15
  70008 | 2012-09-10 |      5760 | Nick Rimando   |   100 | James Hoog |       0.15
  70010 | 2012-10-10 |   1983.43 | Fabian Johnson |   300 | Mc Lyon    |       0.14
  70003 | 2012-10-10 |    2480.4 | Geoff Cameron  |   100 | Lauson Hen |       0.12
  70012 | 2012-06-27 |    250.45 | Julian Green   |   300 | Nail Knite |       0.13
  70011 | 2012-08-17 |     75.29 | Jozy Altidor   |   200 | Paul Adam  |       0.13
  70013 | 2012-04-25 |    3045.6 | Nick Rimando   |   100 | James Hoog |       0.15
(12 rows)
```

Case Study:

```
test=# create table sales(customer_id varchar(4), order_date date, product_id integer);
CREATE TABLE
test=# insert into sales (customer_id,order_date,product_id) values ('A', '2021-01-01',2);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('A', '2021-01-07',2);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('A', '2021-01-10',3);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('A', '2021-01-11',3);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('B', '2021-01-1',2);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('B', '2021-01-02',2);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('B', '2021-01-04',1);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('B', '2021-01-11',1);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('B', '2021-01-16',3);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('B', '2021-02-01',3);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('C', '2021-01-01',3);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('C', '2021-01-01',3);
INSERT 0 1
test=# insert into sales (customer_id,order_date,product_id) values ('C', '2021-01-07',3);
INSERT 0 1
```

```
test=# select * from sales;
 customer_id | order_date | product_id
-------------+------------+------------
 A           | 2021-01-01 |          1
 A           | 2021-01-01 |          2
 A           | 2021-01-07 |          2
 A           | 2021-01-10 |          3
 A           | 2021-01-11 |          3
 B           | 2021-01-01 |          2
 B           | 2021-01-02 |          2
 B           | 2021-01-04 |          1
 B           | 2021-01-11 |          1
 B           | 2021-01-16 |          3
 B           | 2021-02-01 |          3
 C           | 2021-01-01 |          3
 C           | 2021-01-01 |          3
 C           | 2021-01-07 |          3
------------------------------------------------------------------------------------------------------------------------

test=# create table menu(product_id integer, product_name varchar(5), price integer);
CREATE TABLE
test=# insert into menu(product_id,product_name,price) values ('1','sushi',10);
INSERT 0 1
test=# insert into menu(product_id,product_name,price) values ('2','curry',15);
INSERT 0 1
test=# insert into menu(product_id,product_name,price) values ('3','ramen',12);
INSERT 0 1
test=# select * from menu;
 product_id | product_name | price
------------+--------------+-------
          1 | sushi        |    10
          2 | curry        |    15
          3 | ramen        |    12
(3 rows)
------------------------------------------------------------------------------------------------------------------------

test=# create table members(customer_id varchar(4), join_date date);
CREATE TABLE
test=# insert into members(customer_id,join_date) values ('A', '2021-01-07');
INSERT 0 1
test=# insert into members(customer_id,join_date) values ('B', '2021-01-09');
INSERT 0 1
test=# select * from members;
 customer_id | join_date
-------------+------------
 A           | 2021-01-07
 B           | 2021-01-09
(2 rows)
------------------------------------------------------------------------------------------------------------------------
```

1. What is the total amount each customer spent at the restaurant?

```
test=# select s.customer_id, sum(price) AS total_sales from sales s join menu m on s.product_id = m.product_id GROUP BY customer_id;

 customer_id | total_sales
-------------+-------------
 B           |          74
 C           |          36
 A           |          64
(3 rows)
------------------------------------------------------------------------------------------------------------------------------
```

2. How many days has each customer visited the restaurant?

```
test=# select customer_id, COUNT(DISTINCT(order_date)) AS visit_count from sales GROUP BY customer_id;
 customer_id | visit_count
-------------+-------------
 A           |           4
 B           |           6
 C           |           2
(3 rows)
------------------------------------------------------------------------------------------------------------------------------
```

3. What was the first item from the menu purchased by each customer?

CREATE VIEW:-

```
test=# create view combined_sales as select s.customer_id, order_date, mm.join_date, product_name, price, case when join_date<=order_date then 'Y' else 'N' end as mm, RANK() OVER(PARTITION by s.customer_id ORDER BY order_date) ranking from sales s left join menu m on s.product_id = m.product_id left join members mm on s.customer_id = mm.customer_id;
CREATE VIEW

test=# select cs.customer_id, product_name from combined_sales cs join(select customer_id, MIN(order_date) as mordt from combined_sales group by customer_id) as cs2 on cs.order_date =cs2.mordt and cs.customer_id =cs2.customer_id;
 customer_id | product_name
-------------+--------------
 A           | sushi
 A           | curry
 B           | curry
 C           | ramen
 C           | ramen
(5 rows)
------------------------------------------------------------------------------------------------------------------------------
```

4. What is the most purchased item on the menu and how many times was it purchased by all customers?

```
test=# SELECT (COUNT(s.product_id)) AS most_purchased, product_name FROM sales s JOIN menu
m ON s.product_id = m.product_id GROUP BY s.product_id, product_name ORDER BY
most_purchased DESC;
 most_purchased | product_name
----------------+--------------
              7 | ramen
              4 | curry
              3 | sushi
(3 rows)
```

---

5. Which item was the most popular for each customer?

```
test=# Create view pop_item as (select customer_id , product_name, count(product_name) as
ct,dense_rank() OVER (PARTITION by customer_id ORDER BY count(product_name) desc) as
ranking from combined_sales group by customer_id,product_name);
CREATE VIEW

test=# select customer_id,product_name,ct from pop_item where ranking=1;
 customer_id | product_name | ct
-------------+--------------+----
 A           | ramen        |  2
 A           | curry        |  2
 B           | sushi        |  2
 B           | curry        |  2
 B           | ramen        |  2
 C           | ramen        |  3
(6 rows)
```

---

6. Which item was purchased first by the customer after they became a member?

```
test=# create view first_item as (select customer_id, product_name, MIN(order_date) as mordt,
dense_rank() OVER (PARTITION by customer_id ORDER BY MIN(order_date)) as ranking from
combined_sales where mm='Y' group by customer_id, product_name);
CREATE VIEW

test=# select customer_id,product_name,mordt from first_item where ranking=1;
 customer_id | product_name |   mordt
-------------+--------------+------------
 A           | curry        | 2021-01-07
 B           | sushi        | 2021-01-11
(2 rows)
```

---

7. Which item was purchased just before the customer became a member

```
test=# create view last_item as (select customer_id, product_name,order_date, dense_rank() over
(partition by customer_id order by order_date desc) as ranking from combined_sales where
mm='N' and join_date is not null);
CREATE VIEW
```

```
test=# select customer_id,product_name,order_date from last_item where ranking=1;
 customer_id | product_name | order_date
-------------+--------------+------------
 A           | sushi        | 2021-01-01
 A           | curry        | 2021-01-01
 B           | sushi        | 2021-01-04
(3 rows)
```

---

8. What is the total items and amount spent for each member before they became a member?

```
test=# SELECT customer_id, count(DISTINCT(product_name)) as cpdt, SUM(price) AS amt from
combined_sales cs where mm='N' and join_date is not null group by customer_id;
 customer_id | cpdt | amt
-------------+------+-----
 A           |    2 |  25
 B           |    2 |  40
(2 rows)
```

---

9. If each $1 spent equates to 10 points and sushi has a 2x points multiplier - how many points would each customer have?

```
test=# SELECT customer_id, SUM(case when cs.product_name ='sushi' then ct*2*10 else ct*10
end) as cost from(select customer_id,product_name, sum(price) as ct from combined_sales cs
GROUP BY customer_id, product_name) as cs group by customer_id;
 customer_id | cost
-------------+------
 B           |  940
 C           |  360
 A           |  740
(3 rows)
```

---

10. In the first week after a customer joins the program (including their join date) they earn 2x points on all items, not just sushi how many points do customer A and B have at the end of January?