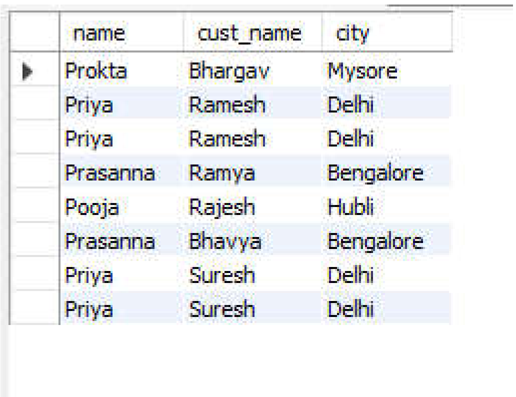
# SQL Assignment 2

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

**Query :**

```
SELECT salesman.name,customer.cust_name,customer.city FROM
customer INNER JOIN salesman ON salesman.city = customer.city;
```

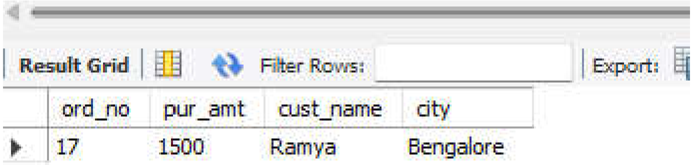


| name | cust_name | city |
|---|---|---|
| Prokta | Bhargav | Mysore |
| Priya | Ramesh | Delhi |
| Priya | Ramesh | Delhi |
| Prasanna | Ramya | Bengalore |
| Pooja | Rajesh | Hubli |
| Prasanna | Bhavya | Bengalore |
| Priya | Suresh | Delhi |
| Priya | Suresh | Delhi |

**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 orders.ord_no,orders.pur_amt,customer.cust_name,customer.city from
orders inner join customer on orders.customer_id= customer.customer_id
where orders.pur_amt between 500 and 2000;
```

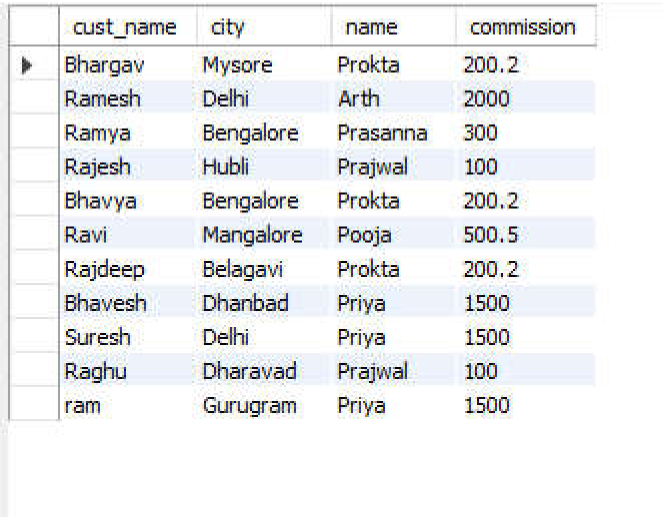


| ord_no | pur_amt | cust_name | city |
|---|---|---|---|
| 17 | 1500 | Ramya | Bengalore |

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 inner join salesman as s on s.salesman_id=c.salesman_id;
```

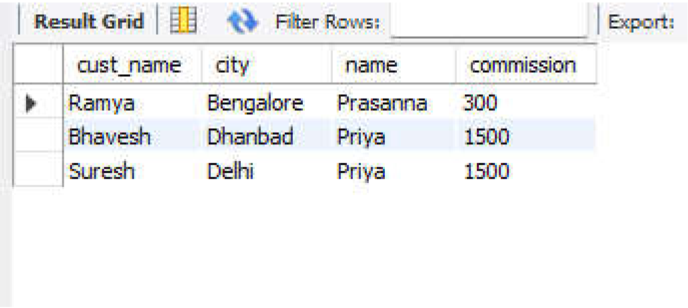


| cust_name | city | name | commission |
|---|---|---|---|
| Bhargav | Mysore | Prokta | 200.2 |
| Ramesh | Delhi | Arth | 2000 |
| Ramya | Bengalore | Prasanna | 300 |
| Rajesh | Hubli | Prajwal | 100 |
| Bhavya | Bengalore | Prokta | 200.2 |
| Ravi | Mangalore | Pooja | 500.5 |
| Rajdeep | Belagavi | Prokta | 200.2 |
| Bhavesh | Dhanbad | Priya | 1500 |
| Suresh | Delhi | Priya | 1500 |
| Raghu | Dharavad | Prajwal | 100 |
| ram | Gurugram | Priya | 1500 |

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,commission.**

**Query :**

```
select customer.cust_name,customer.city,salesman.name,salesman.commission
 from salesman,customer,orders where
 salesman.commission/orders.pur_amt>0.12 and
 salesman.salesman_id=customer.salesman_id and
 orders.customer_id=customer.customer_id;
```

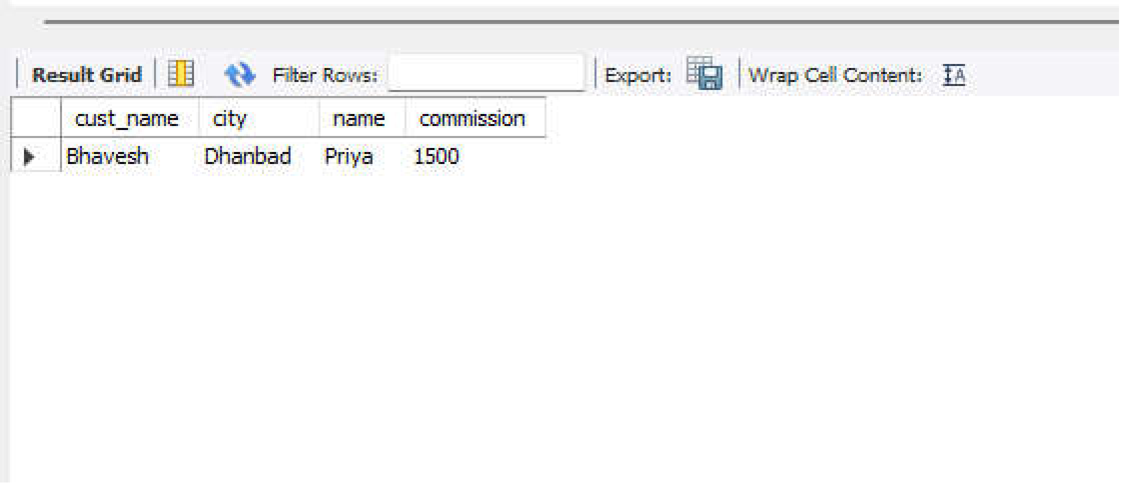


| cust_name | city | name | commission |
|---|---|---|---|
| Ramya | Bengalore | Prasanna | 300 |
| Bhavesh | Dhanbad | Priya | 1500 |
| Suresh | Delhi | Priya | 1500 |

**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 customer.cust_name,customer.city,salesman.name,salesman.commission
  from salesman,customer,orders where
  salesman.commission/orders.pur_amt>0.12 and
  salesman.salesman_id=customer.salesman_id and
  orders.customer_id=customer.customer_id and customer.city!=salesman.city;
```

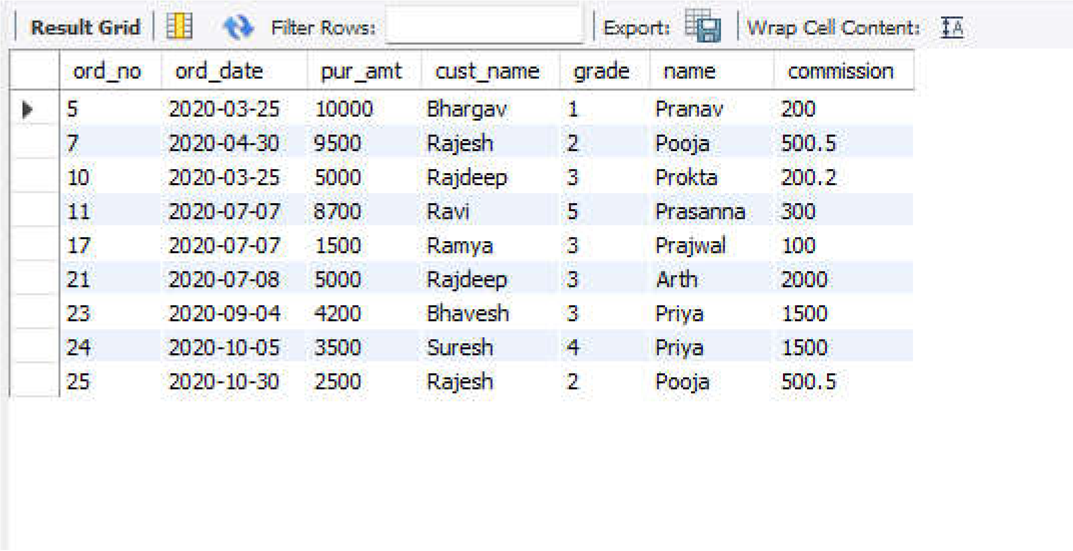


| cust_name | city | name | commission |
|---|---|---|---|
| Bhavesh | Dhanbad | Priya | 1500 |

**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.pur_amt,c.cust_name,c.grade,s.name,s.commission
  from orders as o inner join salesman as s on s.salesman_id=o.salesman_id
  inner join customer as c on c.customer_id=o.customer_id;
```

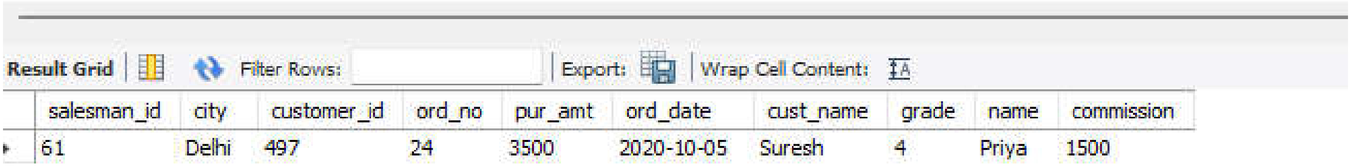


| ord_no | ord_date | pur_amt | cust_name | grade | name | commission |
|---|---|---|---|---|---|---|
| 5 | 2020-03-25 | 10000 | Bhargav | 1 | Pranav | 200 |
| 7 | 2020-04-30 | 9500 | Rajesh | 2 | Pooja | 500.5 |
| 10 | 2020-03-25 | 5000 | Rajdeep | 3 | Prokta | 200.2 |
| 11 | 2020-07-07 | 8700 | Ravi | 5 | Prasanna | 300 |
| 17 | 2020-07-07 | 1500 | Ramya | 3 | Prajwal | 100 |
| 21 | 2020-07-08 | 5000 | Rajdeep | 3 | Arth | 2000 |
| 23 | 2020-09-04 | 4200 | Bhavesh | 3 | Priya | 1500 |
| 24 | 2020-10-05 | 3500 | Suresh | 4 | Priya | 1500 |
| 25 | 2020-10-30 | 2500 | Rajesh | 2 | Pooja | 500.5 |

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 orders natural join customer natural join salesman;
```

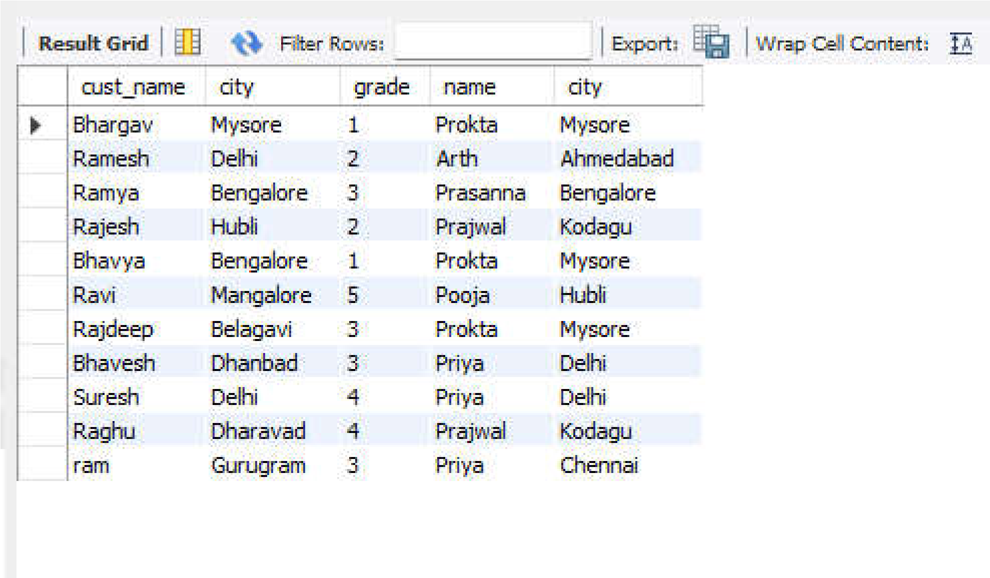


| salesman_id | city | customer_id | ord_no | pur_amt | ord_date | cust_name | grade | name | commission |
|---|---|---|---|---|---|---|---|---|---|
| 61 | Delhi | 497 | 24 | 3500 | 2020-10-05 | Suresh | 4 | Priya | 1500 |

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,c.grade,s.name,s.city from customer as c inner join salesman as s on s.salesman_id=c.salesman_id order by c.customer_id;**

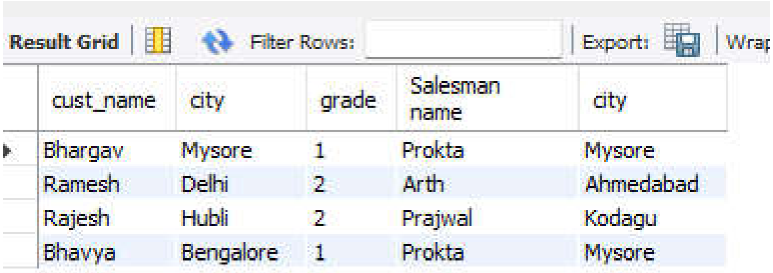


| cust_name | city | grade | name | city |
|---|---|---|---|---|
| Bhargav | Mysore | 1 | Prokta | Mysore |
| Ramesh | Delhi | 2 | Arth | Ahmedabad |
| Ramya | Bengalore | 3 | Prasanna | Bengalore |
| Rajesh | Hubli | 2 | Prajwal | Kodagu |
| Bhavya | Bengalore | 1 | Prokta | Mysore |
| Ravi | Mangalore | 5 | Pooja | Hubli |
| Rajdeep | Belagavi | 3 | Prokta | Mysore |
| Bhavesh | Dhanbad | 3 | Priya | Delhi |
| Suresh | Delhi | 4 | Priya | Delhi |
| Raghu | Dharavad | 4 | Prajwal | Kodagu |
| ram | Gurugram | 3 | Priya | Chennai |

**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,c.grade,s.name as 'Salesman name',s.city FROM
   customer as c LEFT JOIN salesman as s ON s.salesman_id = c.salesman_id
   where c.grade<3 order by c.customer_id;
```

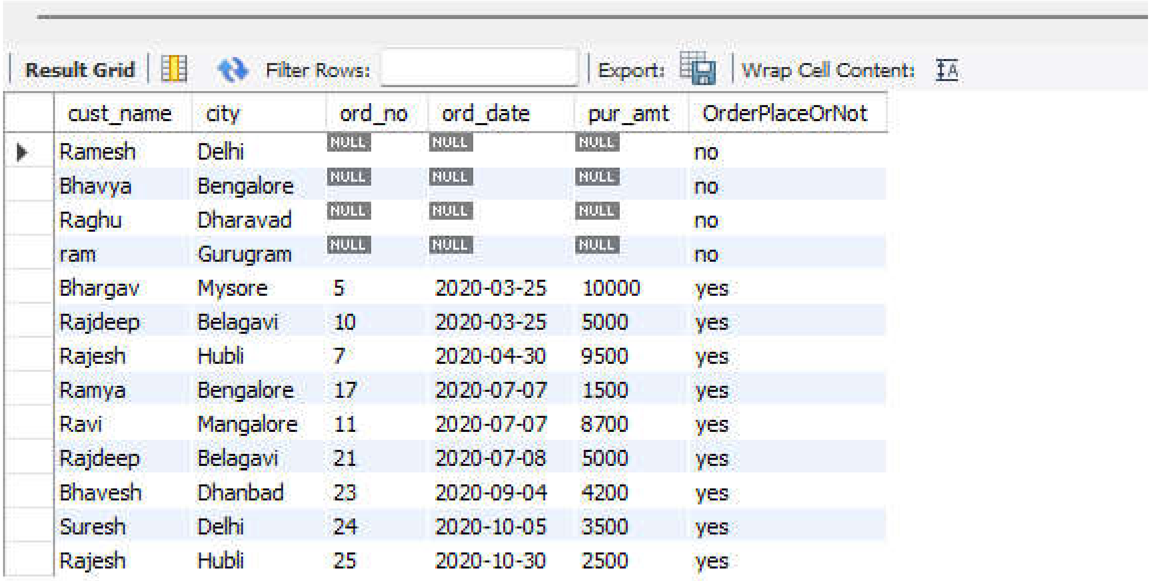


| cust_name | city | grade | Salesman name | city |
|---|---|---|---|---|
| Bhargav | Mysore | 1 | Prokta | Mysore |
| Ramesh | Delhi | 2 | Arth | Ahmedabad |
| Rajesh | Hubli | 2 | Prajwal | Kodagu |
| Bhavya | Bengalore | 1 | Prokta | Mysore |

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 distinct c.cust_name,c.city,o.ord_no,o.ord_date,o.pur_amt, case when
c.customer_id in (select distinct o.customer_id from orders) then 'yes' else 'no'
end as 'OrderPlaceOrNot'  from customer as c left join orders as o on
c.customer_id=o.customer_id order by o.ord_date;
```

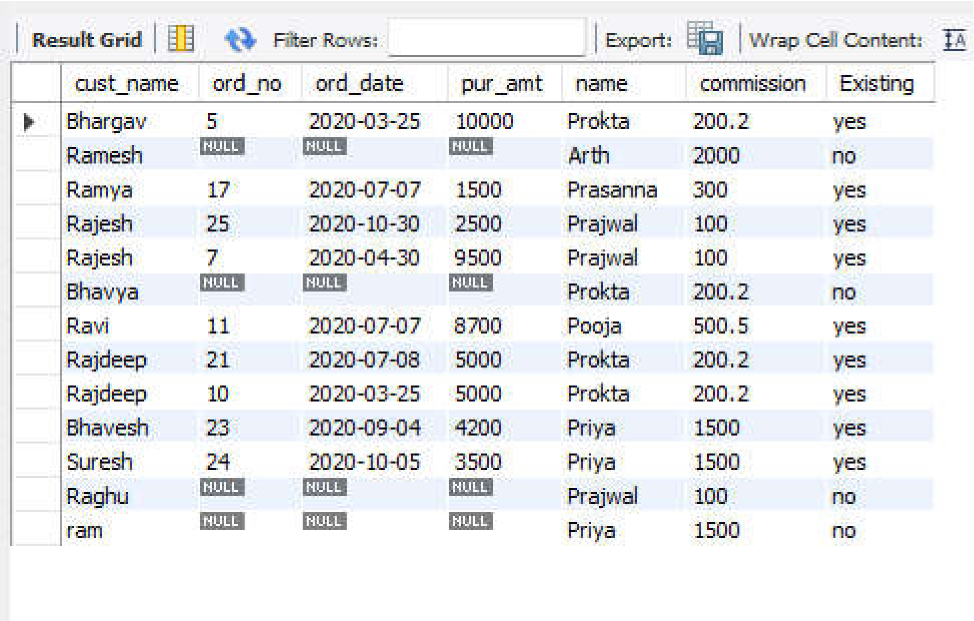


| cust_name | city | ord_no | ord_date | pur_amt | OrderPlaceOrNot |
|---|---|---|---|---|---|
| Ramesh | Delhi | NULL | NULL | NULL | no |
| Bhavya | Bengalore | NULL | NULL | NULL | no |
| Raghu | Dharavad | NULL | NULL | NULL | no |
| ram | Gurugram | NULL | NULL | NULL | no |
| Bhargav | Mysore | 5 | 2020-03-25 | 10000 | yes |
| Rajdeep | Belagavi | 10 | 2020-03-25 | 5000 | yes |
| Rajesh | Hubli | 7 | 2020-04-30 | 9500 | yes |
| Ramya | Bengalore | 17 | 2020-07-07 | 1500 | yes |
| Ravi | Mangalore | 11 | 2020-07-07 | 8700 | yes |
| Rajdeep | Belagavi | 21 | 2020-07-08 | 5000 | yes |
| Bhavesh | Dhanbad | 23 | 2020-09-04 | 4200 | yes |
| Suresh | Delhi | 24 | 2020-10-05 | 3500 | yes |
| Rajesh | Hubli | 25 | 2020-10-30 | 2500 | yes |

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,o.ord_no,o.ord_date,o.pur_amt,s.name ,s.commission,case
when c.customer_id in (select o.customer_id from orders) then 'yes' else 'no' end
as 'Existing'  from customer as c left join orders as o on
c.customer_id=o.customer_id left join salesman as s on
s.salesman_id=c.salesman_id;
```

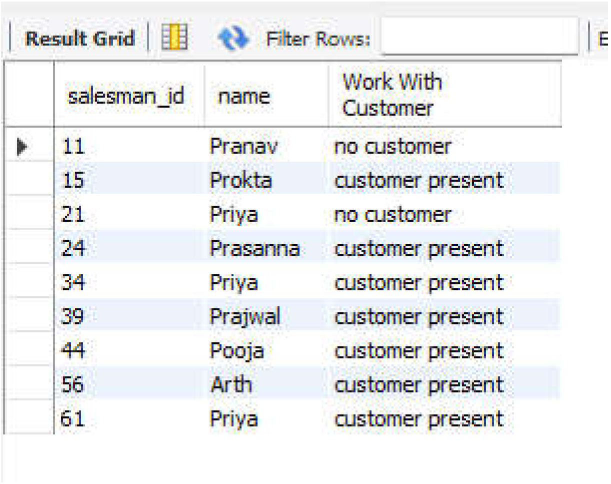


| cust_name | ord_no | ord_date | pur_amt | name | commission | Existing |
|---|---|---|---|---|---|---|
| Bhargav | 5 | 2020-03-25 | 10000 | Prokta | 200.2 | yes |
| Ramesh | NULL | NULL | NULL | Arth | 2000 | no |
| Ramya | 17 | 2020-07-07 | 1500 | Prasanna | 300 | yes |
| Rajesh | 25 | 2020-10-30 | 2500 | Prajwal | 100 | yes |
| Rajesh | 7 | 2020-04-30 | 9500 | Prajwal | 100 | yes |
| Bhavya | NULL | NULL | NULL | Prokta | 200.2 | no |
| Ravi | 11 | 2020-07-07 | 8700 | Pooja | 500.5 | yes |
| Rajdeep | 21 | 2020-07-08 | 5000 | Prokta | 200.2 | yes |
| Rajdeep | 10 | 2020-03-25 | 5000 | Prokta | 200.2 | yes |
| Bhavesh | 23 | 2020-09-04 | 4200 | Priya | 1500 | yes |
| Suresh | 24 | 2020-10-05 | 3500 | Priya | 1500 | yes |
| Raghu | NULL | NULL | NULL | Prajwal | 100 | no |
| ram | NULL | NULL | NULL | Priya | 1500 | no |

**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, case when s.salesman_id in (select salesman_id
from customer) then 'customer present' else 'no customer' end as 'Work With
Customer' from salesman as s order by s.salesman_id;
```

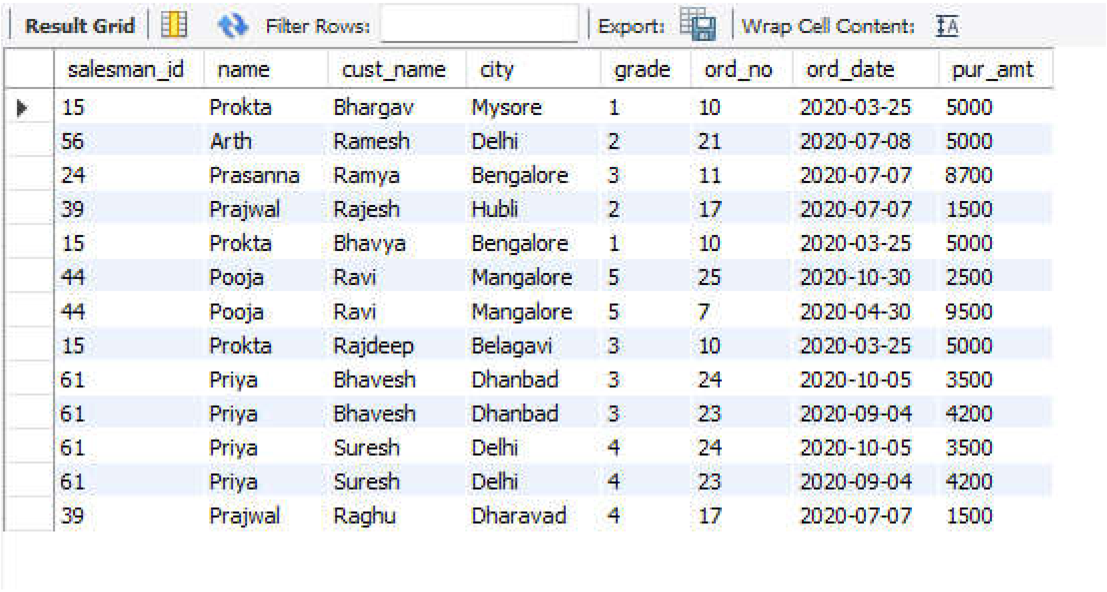


| salesman_id | name | Work With Customer |
|---|---|---|
| 11 | Pranav | no customer |
| 15 | Prokta | customer present |
| 21 | Priya | no customer |
| 24 | Prasanna | customer present |
| 34 | Priya | customer present |
| 39 | Prajwal | customer present |
| 44 | Pooja | customer present |
| 56 | Arth | customer present |
| 61 | Priya | customer present |

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

**Query :**

```
select
s.salesman_id,s.name,c.cust_name,c.city,c.grade,o.ord_no,o.ord_date,o.pur_amt from salesman as s inner join customer as c on
c.salesman_id=s.salesman_id right join orders as o on
o.salesman_id=s.salesman_id where s.salesman_id is not null;
```

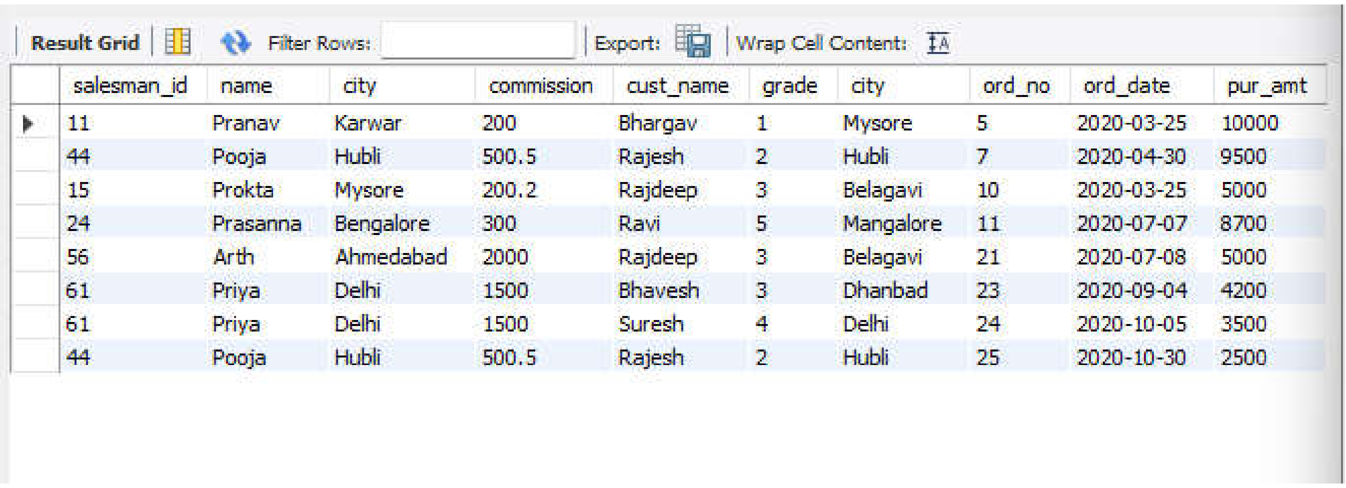


| salesman_id | name | cust_name | city | grade | ord_no | ord_date | pur_amt |
|---|---|---|---|---|---|---|---|
| 15 | Prokta | Bhargav | Mysore | 1 | 10 | 2020-03-25 | 5000 |
| 56 | Arth | Ramesh | Delhi | 2 | 21 | 2020-07-08 | 5000 |
| 24 | Prasanna | Ramya | Bengalore | 3 | 11 | 2020-07-07 | 8700 |
| 39 | Prajwal | Rajesh | Hubli | 2 | 17 | 2020-07-07 | 1500 |
| 15 | Prokta | Bhavya | Bengalore | 1 | 10 | 2020-03-25 | 5000 |
| 44 | Pooja | Ravi | Mangalore | 5 | 25 | 2020-10-30 | 2500 |
| 44 | Pooja | Ravi | Mangalore | 5 | 7 | 2020-04-30 | 9500 |
| 15 | Prokta | Rajdeep | Belagavi | 3 | 10 | 2020-03-25 | 5000 |
| 61 | Priya | Bhavesh | Dhanbad | 3 | 24 | 2020-10-05 | 3500 |
| 61 | Priya | Bhavesh | Dhanbad | 3 | 23 | 2020-09-04 | 4200 |
| 61 | Priya | Suresh | Delhi | 4 | 24 | 2020-10-05 | 3500 |
| 61 | Priya | Suresh | Delhi | 4 | 23 | 2020-09-04 | 4200 |
| 39 | Prajwal | Raghu | Dharavad | 4 | 17 | 2020-07-07 | 1500 |

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.salesman_id,s.name, s.city, s.commission ,c.cust_name,c.grade, c.city,
   o.ord_no,o.ord_date,o.pur_amt from customer c inner join orders o on
   c.customer_id = o.customer_id right join salesman s ON s.salesman_id =
   o.salesman_id WHERE c.grade IS NOT NULL AND o.pur_amt > 2000;
```

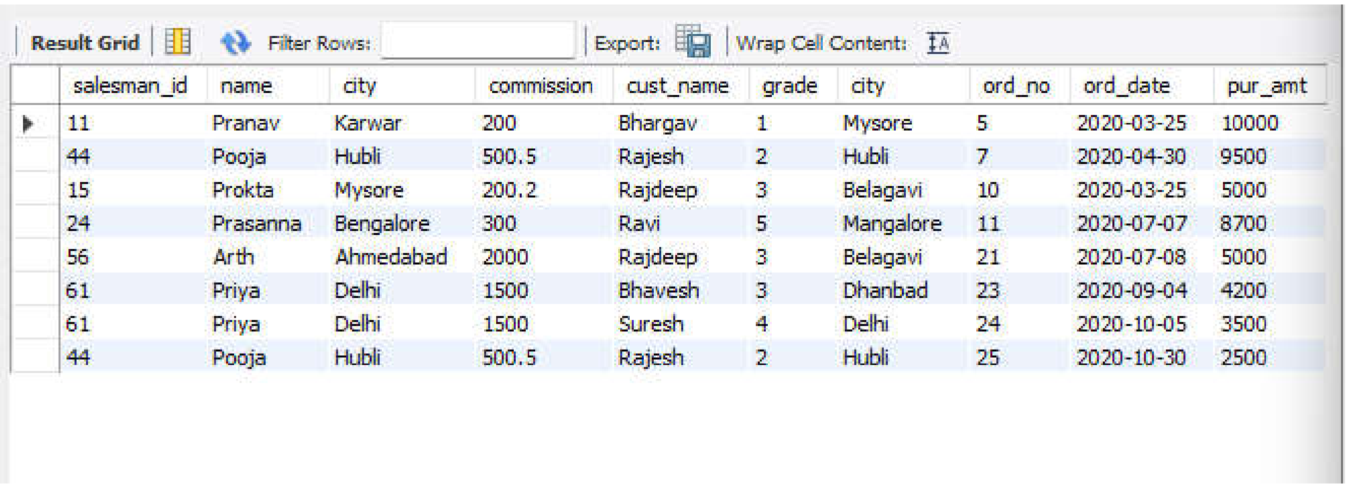


| salesman_id | name | city | commission | cust_name | grade | city | ord_no | ord_date | pur_amt |
|---|---|---|---|---|---|---|---|---|---|
| 11 | Pranav | Karwar | 200 | Bhargav | 1 | Mysore | 5 | 2020-03-25 | 10000 |
| 44 | Pooja | Hubli | 500.5 | Rajesh | 2 | Hubli | 7 | 2020-04-30 | 9500 |
| 15 | Prokta | Mysore | 200.2 | Rajdeep | 3 | Belagavi | 10 | 2020-03-25 | 5000 |
| 24 | Prasanna | Bengalore | 300 | Ravi | 5 | Mangalore | 11 | 2020-07-07 | 8700 |
| 56 | Arth | Ahmedabad | 2000 | Rajdeep | 3 | Belagavi | 21 | 2020-07-08 | 5000 |
| 61 | Priya | Delhi | 1500 | Bhavesh | 3 | Dhanbad | 23 | 2020-09-04 | 4200 |
| 61 | Priya | Delhi | 1500 | Suresh | 4 | Delhi | 24 | 2020-10-05 | 3500 |
| 44 | Pooja | Hubli | 500.5 | Rajesh | 2 | Hubli | 25 | 2020-10-30 | 2500 |

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.salesman_id,s.name, s.city, s.commission ,c.cust_name,c.grade, c.city,
   o.ord_no,o.ord_date,o.pur_amt from customer c inner join orders o on
   c.customer_id = o.customer_id right join salesman s ON s.salesman_id =
   o.salesman_id WHERE c.grade IS NOT NULL AND o.pur_amt > 2000;
```

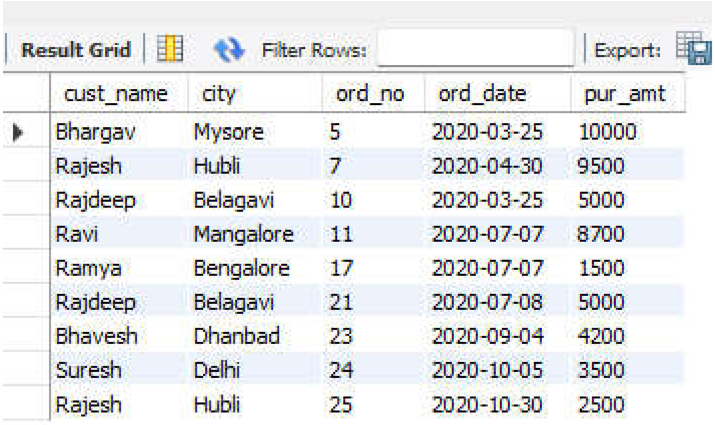


| salesman_id | name | city | commission | cust_name | grade | city | ord_no | ord_date | pur_amt |
|---|---|---|---|---|---|---|---|---|---|
| 11 | Pranav | Karwar | 200 | Bhargav | 1 | Mysore | 5 | 2020-03-25 | 10000 |
| 44 | Pooja | Hubli | 500.5 | Rajesh | 2 | Hubli | 7 | 2020-04-30 | 9500 |
| 15 | Prokta | Mysore | 200.2 | Rajdeep | 3 | Belagavi | 10 | 2020-03-25 | 5000 |
| 24 | Prasanna | Bengalore | 300 | Ravi | 5 | Mangalore | 11 | 2020-07-07 | 8700 |
| 56 | Arth | Ahmedabad | 2000 | Rajdeep | 3 | Belagavi | 21 | 2020-07-08 | 5000 |
| 61 | Priya | Delhi | 1500 | Bhavesh | 3 | Dhanbad | 23 | 2020-09-04 | 4200 |
| 61 | Priya | Delhi | 1500 | Suresh | 4 | Delhi | 24 | 2020-10-05 | 3500 |
| 44 | Pooja | Hubli | 500.5 | Rajesh | 2 | Hubli | 25 | 2020-10-30 | 2500 |

**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.pur_amt from customer c inner join orders o on c.customer_id = o.customer_id where c.grade is not null;**

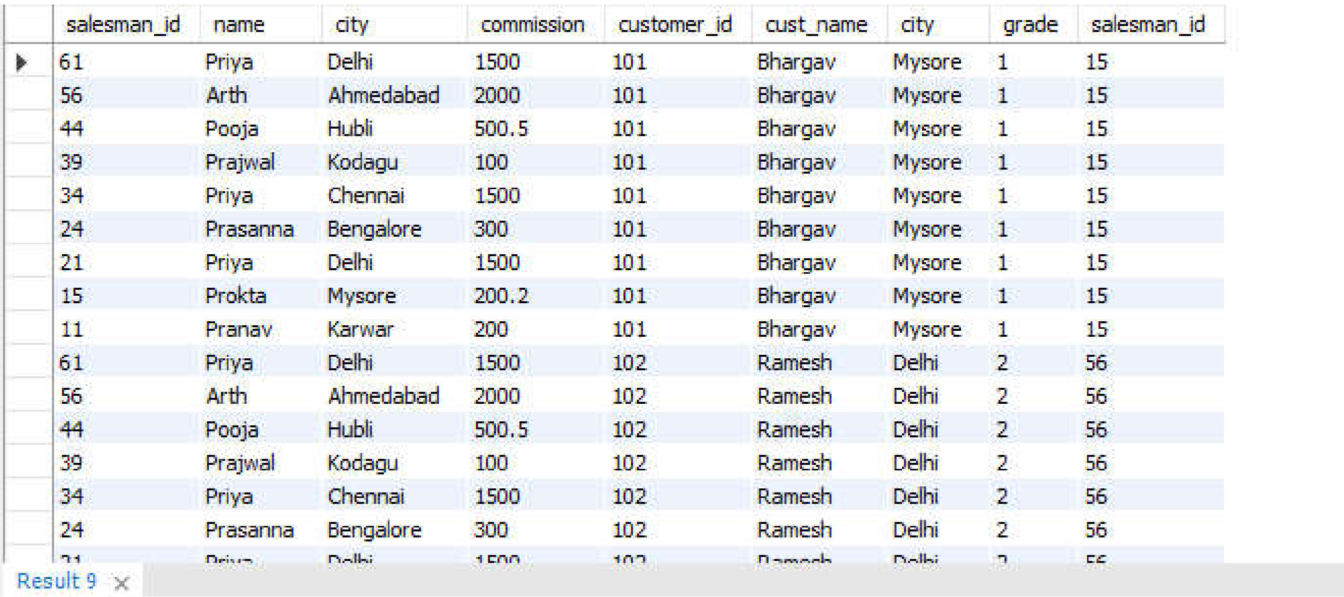


| cust_name | city | ord_no | ord_date | pur_amt |
|---|---|---|---|---|
| Bhargav | Mysore | 5 | 2020-03-25 | 10000 |
| Rajesh | Hubli | 7 | 2020-04-30 | 9500 |
| Rajdeep | Belagavi | 10 | 2020-03-25 | 5000 |
| Ravi | Mangalore | 11 | 2020-07-07 | 8700 |
| Ramya | Bengalore | 17 | 2020-07-07 | 1500 |
| Rajdeep | Belagavi | 21 | 2020-07-08 | 5000 |
| Bhavesh | Dhanbad | 23 | 2020-09-04 | 4200 |
| Suresh | Delhi | 24 | 2020-10-05 | 3500 |
| Rajesh | Hubli | 25 | 2020-10-30 | 2500 |

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

**Query :**

**select * from salesman cross join customer;**

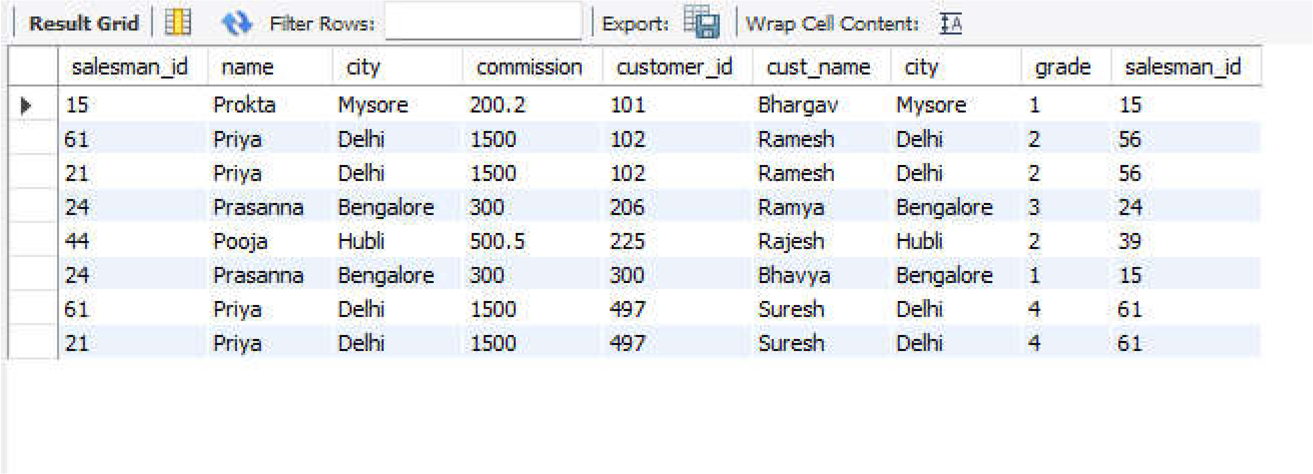


| salesman_id | name | city | commission | customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|---|---|---|---|
| 61 | Priya | Delhi | 1500 | 101 | Bhargav | Mysore | 1 | 15 |
| 56 | Arth | Ahmedabad | 2000 | 101 | Bhargav | Mysore | 1 | 15 |
| 44 | Pooja | Hubli | 500.5 | 101 | Bhargav | Mysore | 1 | 15 |
| 39 | Prajwal | Kodagu | 100 | 101 | Bhargav | Mysore | 1 | 15 |
| 34 | Priya | Chennai | 1500 | 101 | Bhargav | Mysore | 1 | 15 |
| 24 | Prasanna | Bengalore | 300 | 101 | Bhargav | Mysore | 1 | 15 |
| 21 | Priya | Delhi | 1500 | 101 | Bhargav | Mysore | 1 | 15 |
| 15 | Prokta | Mysore | 200.2 | 101 | Bhargav | Mysore | 1 | 15 |
| 11 | Pranav | Karwar | 200 | 101 | Bhargav | Mysore | 1 | 15 |
| 61 | Priya | Delhi | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 56 | Arth | Ahmedabad | 2000 | 102 | Ramesh | Delhi | 2 | 56 |
| 44 | Pooja | Hubli | 500.5 | 102 | Ramesh | Delhi | 2 | 56 |
| 39 | Prajwal | Kodagu | 100 | 102 | Ramesh | Delhi | 2 | 56 |
| 34 | Priya | Chennai | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 24 | Prasanna | Bengalore | 300 | 102 | Ramesh | Delhi | 2 | 56 |
| 21 | Priya | Delhi | 1500 | 102 | Ramesh | Delhi | 2 | 56 |

**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 cross join customer on customer.city=salesman.city;**

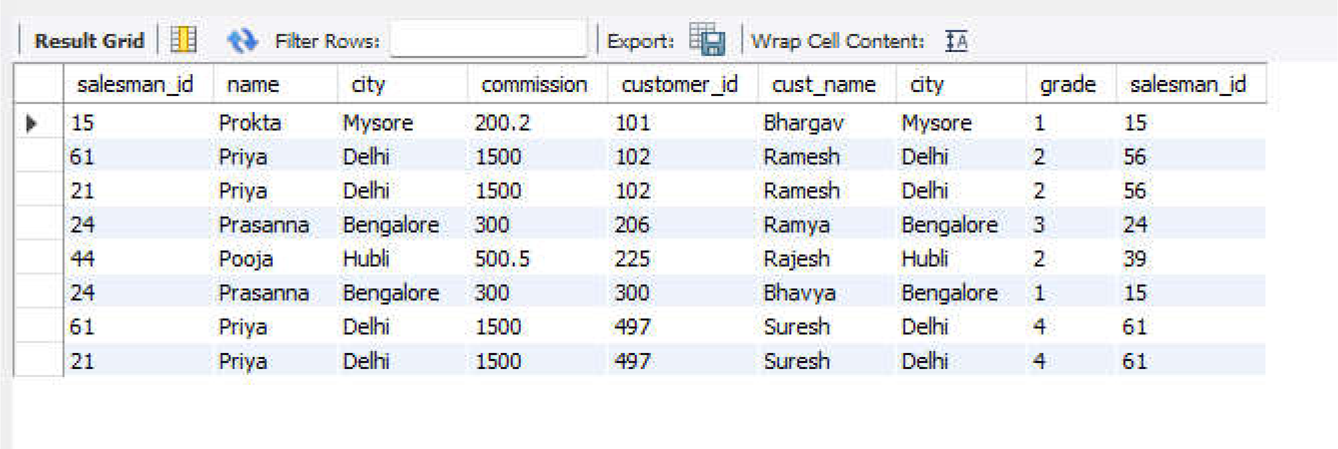


| salesman_id | name | city | commission | customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|---|---|---|---|
| 15 | Prokta | Mysore | 200.2 | 101 | Bhargav | Mysore | 1 | 15 |
| 61 | Priya | Delhi | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 21 | Priya | Delhi | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 24 | Prasanna | Bengalore | 300 | 206 | Ramya | Bengalore | 3 | 24 |
| 44 | Pooja | Hubli | 500.5 | 225 | Rajesh | Hubli | 2 | 39 |
| 24 | Prasanna | Bengalore | 300 | 300 | Bhavya | Bengalore | 1 | 15 |
| 61 | Priya | Delhi | 1500 | 497 | Suresh | Delhi | 4 | 61 |
| 21 | Priya | Delhi | 1500 | 497 | Suresh | Delhi | 4 | 61 |

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 cross join customer on customer.city=salesman.city
where customer.grade is not null;
```

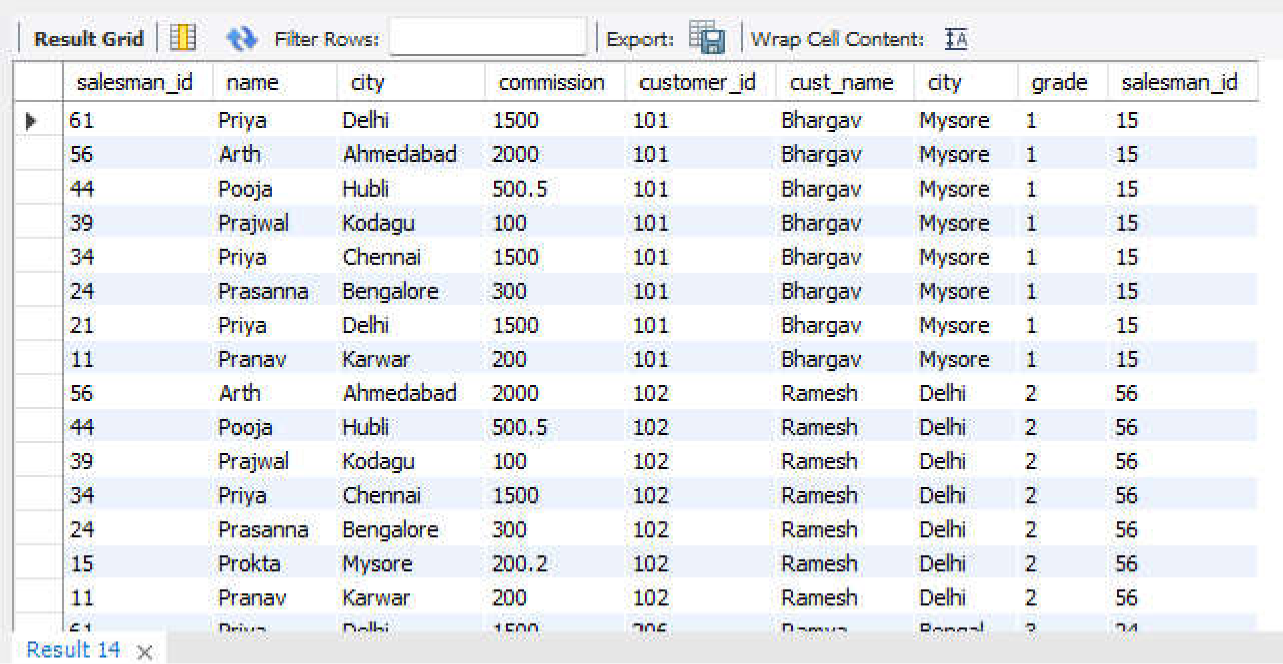


| salesman_id | name | city | commission | customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|---|---|---|---|
| 15 | Prokta | Mysore | 200.2 | 101 | Bhargav | Mysore | 1 | 15 |
| 61 | Priya | Delhi | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 21 | Priya | Delhi | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 24 | Prasanna | Bengalore | 300 | 206 | Ramya | Bengalore | 3 | 24 |
| 44 | Pooja | Hubli | 500.5 | 225 | Rajesh | Hubli | 2 | 39 |
| 24 | Prasanna | Bengalore | 300 | 300 | Bhavya | Bengalore | 1 | 15 |
| 61 | Priya | Delhi | 1500 | 497 | Suresh | Delhi | 4 | 61 |
| 21 | Priya | Delhi | 1500 | 497 | Suresh | Delhi | 4 | 61 |

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 cross join customer on customer.city!=salesman.city
   where customer.grade is not null;
```



| salesman_id | name | city | commission | customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|---|---|---|---|
| 61 | Priya | Delhi | 1500 | 101 | Bhargav | Mysore | 1 | 15 |
| 56 | Arth | Ahmedabad | 2000 | 101 | Bhargav | Mysore | 1 | 15 |
| 44 | Pooja | Hubli | 500.5 | 101 | Bhargav | Mysore | 1 | 15 |
| 39 | Prajwal | Kodagu | 100 | 101 | Bhargav | Mysore | 1 | 15 |
| 34 | Priya | Chennai | 1500 | 101 | Bhargav | Mysore | 1 | 15 |
| 24 | Prasanna | Bengalore | 300 | 101 | Bhargav | Mysore | 1 | 15 |
| 21 | Priya | Delhi | 1500 | 101 | Bhargav | Mysore | 1 | 15 |
| 11 | Pranav | Karwar | 200 | 101 | Bhargav | Mysore | 1 | 15 |
| 56 | Arth | Ahmedabad | 2000 | 102 | Ramesh | Delhi | 2 | 56 |
| 44 | Pooja | Hubli | 500.5 | 102 | Ramesh | Delhi | 2 | 56 |
| 39 | Prajwal | Kodagu | 100 | 102 | Ramesh | Delhi | 2 | 56 |
| 34 | Priya | Chennai | 1500 | 102 | Ramesh | Delhi | 2 | 56 |
| 24 | Prasanna | Bengalore | 300 | 102 | Ramesh | Delhi | 2 | 56 |
| 15 | Prokta | Mysore | 200.2 | 102 | Ramesh | Delhi | 2 | 56 |
| 11 | Pranav | Karwar | 200 | 102 | Ramesh | Delhi | 2 | 56 |
| 61 | Priya | Delhi | 1500 | 206 | Ramya | Bengal | 3 | 24 |