

Assignment-2

Name: Parmar Bhavik J.

➤ Create Table:

QUERY:

● Salesman Table

```
CREATE TABLE salesman(  
  salesman_id int NOT NULL PRIMARY KEY,  
  name nvarchar(50) NOT NULL,  
  city nvarchar(50),  
  commission int,  
)
```

● Customer Table

```
CREATE TABLE customer(  
  customer_id int NOT NULL PRIMARY KEY,  
  cust_name nvarchar(50),  
  city nvarchar(50),  
  grade int,  
  salesman_id int,  
)
```

```
ALTER TABLE customer  
ADD FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id);
```

● Order Table

```
CREATE TABLE orders(  
  ord_no int NOT NULL PRIMARY KEY,  
  purch_amt money NOT NULL,  
  ord_date date NOT NULL,  
  customer_id int NOT NULL,  
  salesman_id int NOT NULL,  
  FOREIGN KEY (customer_id) REFERENCES customer(customer_id),  
  FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id),  
)
```

- 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, c.cust_name, s.city  
FROM salesman s inner join customer c on s.city=c.city;
```

	name	cust_name	city
1	James	Nick	New York
2	James	Brad Davis	New York
3	Pit	Julian	London
4	Nail	Fabian	Paris
5	Lyon	Fabian	Paris
6	Pit	Brad Guzan	London
7	James	bravo	New York

- 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 customer c inner join orders o on  
c.customer_id=o.customer_id  
WHERE o.purch_amt BETWEEN 500 AND 2000;
```

	ord_no	purch_amt	cust_name	city
1	7005	948.50	Graham	Califomia
2	7008	1983.43	Fabian	Paris

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

QUERY:

```
SELECT s.name, c.cust_name AS "Costomer Name", s.city ,
s.name,s.commission
FROM salesman s inner join customer c on
s.salesman_id=c.salesman_id;
```

	name	Costomer Name	city	name	commission
1	James	Nick	New York	James	15
2	James	Brad Davis	New York	James	15
3	Nail	Graham	Paris	Nail	13
4	Nail	Julian	Paris	Nail	13
5	Lyon	Fabian	Paris	Lyon	14
6	Lauson	Cameron	San Fransisco	Lauson	12
7	Paul	Jozy	Rome	Paul	13
8	Pit	Brad Guzan	London	Pit	11
9	James	bravo	New York	James	15
10	Nail	sachin	Paris	Nail	13

- 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 s.name, c.cust_name AS "Costomer Name", s.city ,
s.name,s.commission FROM salesman s inner join customer c on
s.salesman_id=c.salesman_id WHERE s.commission>12;
```

	name	Costomer Name	city	name	commission
1	James	Nick	New York	James	15
2	James	Brad Davis	New York	James	15
3	Nail	Graham	Paris	Nail	13
4	Nail	Julian	Paris	Nail	13
5	Lyon	Fabian	Paris	Lyon	14
6	Paul	Jozy	Rome	Paul	13
7	James	bravo	New York	James	15
8	Nail	sachin	Paris	Nail	13

- 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 s.name, c.cust_name AS "Costomer Name", c.city as
"customer city" , s.name,      s.city as "salesman
city",s.commission FROM salesman s inner join customer c on
s.salesman_id=c.salesman_id WHERE S.commission>12 AND s.city !
= c.city;
```

	name	Costomer Name	customer city	name	salesman city	commission
1	Nail	Graham	California	Nail	Paris	13
2	Nail	Julian	London	Nail	Paris	13
3	Paul	Jozy	Moscow	Paul	Rome	13
4	Nail	sachin	Ahmedabad	Nail	Paris	13

- 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.grade , s.name as "Salesman Name",
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 Name	commission
1	7001	2012-10-05	150.50	Graham	200	Nail	13
2	7002	2012-09-10	270.65	Brad Guzan	NULL	Pit	11
3	7003	2012-10-05	65.26	Nick	100	James	15
4	7004	2012-08-17	110.50	Cameron	100	Lauson	12
5	7005	2012-09-10	948.50	Graham	200	Nail	13
6	7006	2012-07-27	2400.60	Brad Davis	200	James	15
7	7007	2012-09-10	5760.00	Nick	100	James	15
8	7008	2012-10-10	1983.43	Fabian	300	Lyon	14
9	7009	2012-10-10	2480.40	Cameron	100	Lauson	12
10	7010	2012-06-27	250.45	Julian	300	Nail	13
11	7011	2012-08-17	75.29	Jozy	200	Paul	13
12	7012	2012-04-25	3045.60	Nick	100	James	15

- 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 o.* , c.cust_name as "customer name" , c.city as
"customer city" , c.grade , s.name as "salesman name" , s.city
as "salesman city" , 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	purch_amt	ord_date	customer_id	salesman_id	customer name	customer city	grade	salesman name	salesman city	commission
1	7001	150.50	2012-10-05	3003	5002	Graham	California	200	Nail	Paris	13
2	7002	270.65	2012-09-10	3008	5003	Brad Guzan	London	NULL	Pit	London	11
3	7003	65.26	2012-10-05	3001	5001	Nick	New York	100	James	New York	15
4	7004	110.50	2012-08-17	3006	5006	Cameron	Berlin	100	Lauson	San Fransisco	12
5	7005	948.50	2012-09-10	3003	5002	Graham	California	200	Nail	Paris	13
6	7006	2400.60	2012-07-27	3002	5001	Brad Davis	New York	200	James	New York	15
7	7007	5760.00	2012-09-10	3001	5001	Nick	New York	100	James	New York	15
8	7008	1983.43	2012-10-10	3005	5004	Fabian	Paris	300	Lyon	Paris	14
9	7009	2480.40	2012-10-10	3006	5006	Cameron	Berlin	100	Lauson	San Fransisco	12
10	7010	250.45	2012-06-27	3004	5002	Julian	London	300	Nail	Paris	13
11	7011	75.29	2012-08-17	3007	5005	Jozy	Moscow	200	Paul	Rome	13
12	7012	3045.60	2012-04-25	3001	5001	Nick	New York	100	James	New York	15

- 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 as "cutomer name", c.city as "customer
city" , c.grade , s.name as "salesman name" , s.city as
"salesman city" FROM salesman s inner join customer c on
s.salesman_id=c.salesman_id order by c.customer_id asc;
```

	cutomer name	customer city	grade	salesman name	salesman city
1	Nick	New York	100	James	New York
2	Brad Davis	New York	200	James	New York
3	Graham	California	200	Nail	Paris
4	Julian	London	300	Nail	Paris
5	Fabian	Paris	300	Lyon	Paris
6	Cameron	Berlin	100	Lauson	San Fransisco
7	Jozy	Moscow	200	Paul	Rome
8	Brad Guzan	London	NULL	Pit	London
9	bravo	New York	100	James	New York
10	sachin	Ahmedabad	200	Nail	Paris

➤ 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 as "customer name", c.city as "customer city" ,
c.grade , s.name as "salesman name" , s.city as "salesman city"
FROM salesman s inner join customer c on
s.salesman_id=c.salesman_id
WHERE c.grade<300
order by c.customer_id asc;
```

	customer name	customer city	grade	salesman name	salesman city
1	Nick	New York	100	James	New York
2	Brad Davis	New York	200	James	New York
3	Graham	California	200	Nail	Paris
4	Cameron	Berlin	100	Lauson	San Fransisco
5	Jozy	Moscow	200	Paul	Rome
6	bravo	New York	100	James	New York
7	sachin	Ahmedabad	200	Nail	Paris

➤ 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 as "customer name" , c.city as "customer city" ,
o.ord_no , o.ord_date , o.purch_amt as "Order Amount"
FROM customer c left outer join orders o on
c.customer_id=o.customer_id
order by o.ord_date;
```

	customer name	customer city	ord_no	ord_date	Order Amount
1	bravo	New York	NULL	NULL	NULL
2	sachin	Ahmedabad	NULL	NULL	NULL
3	Nick	New York	7012	2012-04-25	3045.60
4	Julian	London	7010	2012-06-27	250.45
5	Brad Davis	New York	7006	2012-07-27	2400.60
6	Cameron	Berlin	7004	2012-08-17	110.50
7	Jozy	Moscow	7011	2012-08-17	75.29
8	Brad Guzan	London	7002	2012-09-10	270.65
9	Graham	California	7005	2012-09-10	948.50
10	Nick	New York	7007	2012-09-10	5760.00
11	Nick	New York	7003	2012-10-05	65.26
12	Graham	California	7001	2012-10-05	150.50
13	Cameron	Berlin	7009	2012-10-10	2480.40
14	Fabian	Paris	7008	2012-10-10	1983.43

- 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 as "customer name" , c.city as "customer city" ,
o.ord_no , o.ord_date , o.purch_amt as "Order Amount" , s.name as
"salesman name" , s.commission FROM customer c left outer join
orders o on c.customer_id=o.customer_id left outer join salesman s
on o.salesman_id = s.salesman_id;
```

	customer name	customer city	ord_no	ord_date	Order Amount	salesman name	commission
1	Nick	New York	7003	2012-10-05	65.26	James	15
2	Nick	New York	7007	2012-09-10	5760.00	James	15
3	Nick	New York	7012	2012-04-25	3045.60	James	15
4	Brad Davis	New York	7006	2012-07-27	2400.60	James	15
5	Graham	California	7001	2012-10-05	150.50	Nail	13
6	Graham	California	7005	2012-09-10	948.50	Nail	13
7	Julian	London	7010	2012-06-27	250.45	Nail	13
8	Fabian	Paris	7008	2012-10-10	1983.43	Lyon	14
9	Cameron	Berlin	7004	2012-08-17	110.50	Lauson	12
10	Cameron	Berlin	7009	2012-10-10	2480.40	Lauson	12
11	Jozy	Moscow	7011	2012-08-17	75.29	Paul	13
12	Brad Guzan	London	7002	2012-09-10	270.65	Pit	11
13	bravo	New York	NULL	NULL	NULL	NULL	NULL
14	sachin	Ahmedabad	NULL	NULL	NULL	NULL	NULL

- 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.*,c.cust_name as "customer name" FROM salesman s left
outer join customer c on s.salesman_id=c.salesman_id order by
s.salesman_id asc;
```

	salesman_id	name	city	commission	customer name
1	5001	James	New York	15	Nick
2	5001	James	New York	15	Brad Davis
3	5001	James	New York	15	bravo
4	5002	Nail	Paris	13	Graham
5	5002	Nail	Paris	13	Julian
6	5002	Nail	Paris	13	sachin
7	5003	Pit	London	11	Brad Guzan
8	5004	Lyon	Paris	14	Fabian
9	5005	Paul	Rome	13	Jozy
10	5006	Lauson	San Fransisco	12	Cameron
11	5007	Devillers	Africa	14	NULL
12	5008	Putin	Russia	13	NULL

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

QUERY:

```
Select s.*, c.cust_name as 'Customer Name', c.city as 'Customer City', c.grade, o.ord_no, o.ord_date, o.purch_amt from salesman s
Inner Join customer c On s.salesman_id = c.salesman_id
Inner Join orders o On s.salesman_id = o.salesman_id;
```

	salesman_id	name	city	commission	Customer Name	Customer City	grade	ord_no	ord_date	purch_amt
1	5002	Nail	Paris	13	Graham	California	200	7001	2012-10-05	150.50
2	5002	Nail	Paris	13	Julian	London	300	7001	2012-10-05	150.50
3	5002	Nail	Paris	13	sachin	Ahmedabad	200	7001	2012-10-05	150.50
4	5003	Pit	London	11	Brad Guzan	London	NULL	7002	2012-09-10	270.65
5	5001	James	New York	15	Nick	New York	100	7003	2012-10-05	65.26
6	5001	James	New York	15	Brad Davis	New York	200	7003	2012-10-05	65.26
7	5001	James	New York	15	bravo	New York	100	7003	2012-10-05	65.26
8	5006	Lauson	San Fransisco	12	Cameron	Berlin	100	7004	2012-08-17	110.50
9	5002	Nail	Paris	13	Graham	California	200	7005	2012-09-10	948.50
10	5002	Nail	Paris	13	Julian	London	300	7005	2012-09-10	948.50
11	5002	Nail	Paris	13	sachin	Ahmedabad	200	7005	2012-09-10	948.50
12	5001	James	New York	15	Nick	New York	100	7006	2012-07-27	2400.60
13	5001	James	New York	15	Brad Davis	New York	200	7006	2012-07-27	2400.60
14	5001	James	New York	15	bravo	New York	100	7006	2012-07-27	2400.60
15	5001	James	New York	15	Nick	New York	100	7007	2012-09-10	5760.00
16	5001	James	New York	15	Brad Davis	New York	200	7007	2012-09-10	5760.00
17	5001	James	New York	15	bravo	New York	100	7007	2012-09-10	5760.00
..	----	-	-	..	-	-	---	----	-----	-----

➤ 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.*, c.cust_name as 'Customer Name', c.city as 'Customer City', c.grade, o.ord_no, o.ord_date, o.purch_amt from salesman s
right outer Join customer c On s.salesman_id = c.salesman_id
left outer Join orders o On c.customer_id = o.customer_id
where o.purch_amt > 2000 And c.grade Is Not NULL;
```


	salesman_id	name	city	commission	Customer Name	Customer City	grade	ord_no	ord_date	purch_amt
1	5001	James	New York	15	Brad Davis	New York	200	7006	2012-07-27	2400.60
2	5001	James	New York	15	Nick	New York	100	7007	2012-09-10	5760.00
3	5006	Lauson	San Fransisco	12	Cameron	Berlin	100	7009	2012-10-10	2480.40
4	5001	James	New York	15	Nick	New York	100	7012	2012-04-25	3045.60

- 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.*, c.cust_name as 'Customer Name', c.city as 'Customer City', c.grade, o.ord_no, o.ord_date, o.purch_amt from salesman s right outer Join customer c On s.salesman_id = c.salesman_id left outer Join orders o On c.customer_id = o.customer_id where o.purch_amt > 2000 And c.grade Is Not NULL;
```

	salesman_id	name	city	commission	Customer Name	Customer City	grade	ord_no	ord_date	purch_amt
1	5001	James	New York	15	Brad Davis	New York	200	7006	2012-07-27	2400.60
2	5001	James	New York	15	Nick	New York	100	7007	2012-09-10	5760.00
3	5006	Lauson	San Fransisco	12	Cameron	Berlin	100	7009	2012-10-10	2480.40
4	5001	James	New York	15	Nick	New York	100	7012	2012-04-25	3045.60

- 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 AS "Order Amount" FROM customer c FULL OUTER JOIN orders o ON c.customer_id=o.customer_id WHERE c.grade IS NOT NULL;
```

	cust_name	city	ord_no	ord_date	Order Amount
1	Nick	New York	7003	2012-10-05	65.26
2	Nick	New York	7007	2012-09-10	5760.00
3	Nick	New York	7012	2012-04-25	3045.60
4	Brad Davis	New York	7006	2012-07-27	2400.60
5	Graham	California	7001	2012-10-05	150.50
6	Graham	California	7005	2012-09-10	948.50
7	Julian	London	7010	2012-06-27	250.45
8	Fabian	Paris	7008	2012-10-10	1983.43
9	Cameron	Berlin	7004	2012-08-17	110.50
10	Cameron	Berlin	7009	2012-10-10	2480.40
11	Jozy	Moscow	7011	2012-08-17	75.29
12	bravo	New York	NULL	NULL	NULL
13	sachin	Ahmedabad	NULL	NULL	NULL

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

QUERY:

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

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James	New York	15	3001	Nick	New York	100	5001
2	5001	James	New York	15	3002	Brad Davis	New York	200	5001
3	5001	James	New York	15	3003	Graham	California	200	5002
4	5001	James	New York	15	3004	Julian	London	300	5002
5	5001	James	New York	15	3005	Fabian	Paris	300	5004
6	5001	James	New York	15	3006	Cameron	Berlin	100	5006
7	5001	James	New York	15	3007	Jozy	Moscow	200	5005
8	5001	James	New York	15	3008	Brad Guzan	London	NULL	5003
9	5001	James	New York	15	3009	bravo	New York	100	5001
10	5001	James	New York	15	3010	sachin	Ahmedabad	200	5002
11	5002	Nail	Paris	13	3001	Nick	New York	100	5001
12	5002	Nail	Paris	13	3002	Brad Davis	New York	200	5001
13	5002	Nail	Paris	13	3003	Graham	California	200	5002
14	5002	Nail	Paris	13	3004	Julian	London	300	5002
15	5002	Nail	Paris	13	3005	Fabian	Paris	300	5004
16	5002	Nail	Paris	13	3006	Cameron	Berlin	100	5006
17	5002	Nail	Paris	13	3007	Jozy	Moscow	200	5005
18	5002	Nail	Paris	13	3008	Brad Guzan	London	NULL	5003
19	5002	Nail	Paris	13	3009	bravo	New York	100	5001

➤ 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 s cross join customer c
where s.city = c.city;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James	New York	15	3001	Nick	New York	100	5001
2	5001	James	New York	15	3002	Brad Davis	New York	200	5001
3	5003	Pit	London	11	3004	Julian	London	300	5002
4	5002	Nail	Paris	13	3005	Fabian	Paris	300	5004
5	5004	Lyon	Paris	14	3005	Fabian	Paris	300	5004
6	5003	Pit	London	11	3008	Brad Guzan	London	NULL	5003
7	5001	James	New York	15	3009	bravo	New York	100	5001

➤ 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 s cross join customer c
where s.city = c.city and c.grade is not null;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James	New York	15	3001	Nick	New York	100	5001
2	5001	James	New York	15	3002	Brad Davis	New York	200	5001
3	5003	Pit	London	11	3004	Julian	London	300	5002
4	5002	Nail	Paris	13	3005	Fabian	Paris	300	5004
5	5004	Lyon	Paris	14	3005	Fabian	Paris	300	5004
6	5001	James	New York	15	3009	bravo	New York	100	5001

➤ 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 s cross join customer c
where s.city != c.city and c.grade is not null;
```

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James	New York	15	3003	Graham	California	200	5002
2	5001	James	New York	15	3004	Julian	London	300	5002
3	5001	James	New York	15	3005	Fabian	Paris	300	5004
4	5001	James	New York	15	3006	Cameron	Berlin	100	5006
5	5001	James	New York	15	3007	Jozy	Moscow	200	5005
6	5001	James	New York	15	3010	sachin	Ahmedabad	200	5002
7	5002	Nail	Paris	13	3001	Nick	New York	100	5001
8	5002	Nail	Paris	13	3002	Brad Davis	New York	200	5001
9	5002	Nail	Paris	13	3003	Graham	California	200	5002
10	5002	Nail	Paris	13	3004	Julian	London	300	5002
11	5002	Nail	Paris	13	3006	Cameron	Berlin	100	5006
12	5002	Nail	Paris	13	3007	Jozy	Moscow	200	5005
13	5002	Nail	Paris	13	3009	bravo	New York	100	5001
14	5002	Nail	Paris	13	3010	sachin	Ahmedabad	200	5002
15	5003	Pit	London	11	3001	Nick	New York	100	5001
16	5003	Pit	London	11	3002	Brad Davis	New York	200	5001
17	5003	Pit	London	11	3003	Graham	California	200	5002
18	5003	Pit	London	11	3005	Fabian	Paris	300	5004
19	5003	Pit	London	11	3006	Cameron	Berlin	100	5006