SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

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

select salesman.name , customer.cust_name ,customer.city
from salesman join customer
on salesman.city = customer.city

/*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*/

select orders.ord_no,orders.purch_amt,customer.cust_name,customer.city
from orders, customer
where customer.customer_id = orders.customer_id
```

133 %

Results | Messages

	name	cust_name	city
1	Pit Alex	Brad Guzan	London
2	James Hoog	Nick Rimando	New York
3	Nail Knite	Fabian Johnson	Paris
4	Mc Lyon	Fabian Johnson	Paris
5	James Hoog	Brad Davis	New York

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
from salesman join customer
on salesman.city = customer.city

/*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*/

select orders.ord_no,orders.purch_amt,customer.cust_name,customer.city
from orders, customer
where customer.customer_id = orders.customer_id
and orders.purch_amt between 500 AND 2000

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

select customer.cust_name , customer.city , salesman.name , salesman.commission
```

133 %

Results | Messages

	ord_no	purch_amt	cust_name	city
1	70007	948.5	Graham Zusi	California
2	70010	1983.43	Fabian Johnson	Paris

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
and orders.purch_amt between 500 AND 2000

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

select customer.cust_name , customer.city , salesman.name , salesman.commission
from customer
inner join salesman
on customer.salesman_id=salesman.salesman_id

/*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.*/

select customer.cust_name , customer.city , salesman.name , salesman.commission
```

133 %

Results | Messages

	cust_name	city	name	commission
1	Brad Guzan	London	Pit Alex	0.1100
2	Nick Rimando	New York	James Hoog	0.1500
3	Jozy Altidor	Moscow	Paul Adam	0.1300
4	Fabian Johnson	Paris	Mc Lyon	0.1400
5	Graham Zusi	California	Nail Knite	0.1300
6	Brad Davis	New York	James Hoog	0.1500
7	Julian Green	London	Nail Knite	0.1300
8	Geoff Cameron	Berlin	Lauson Hen	0.1200

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
from customer
inner join salesman
on customer.salesman_id=salesman.salesman_id

/*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.*/

select customer.cust_name , customer.city , salesman.name , salesman.commission
from customer
inner join salesman
on customer.salesman_id = salesman.salesman_id
where salesman.commission>.12;

/*5. write a SQL query to locate those salespeople who do not live in the same city where
```

133 %

Results | Messages

	cust_name	city	name	commission
1	Nick Rimando	New York	James Hoog	0.1500
2	Jozy Altidor	Moscow	Paul Adam	0.1300
3	Fabian Johnson	Paris	Mc Lyon	0.1400
4	Graham Zusi	California	Nail Knite	0.1300
5	Brad Davis	New York	James Hoog	0.1500
6	Julian Green	London	Nail Knite	0.1300

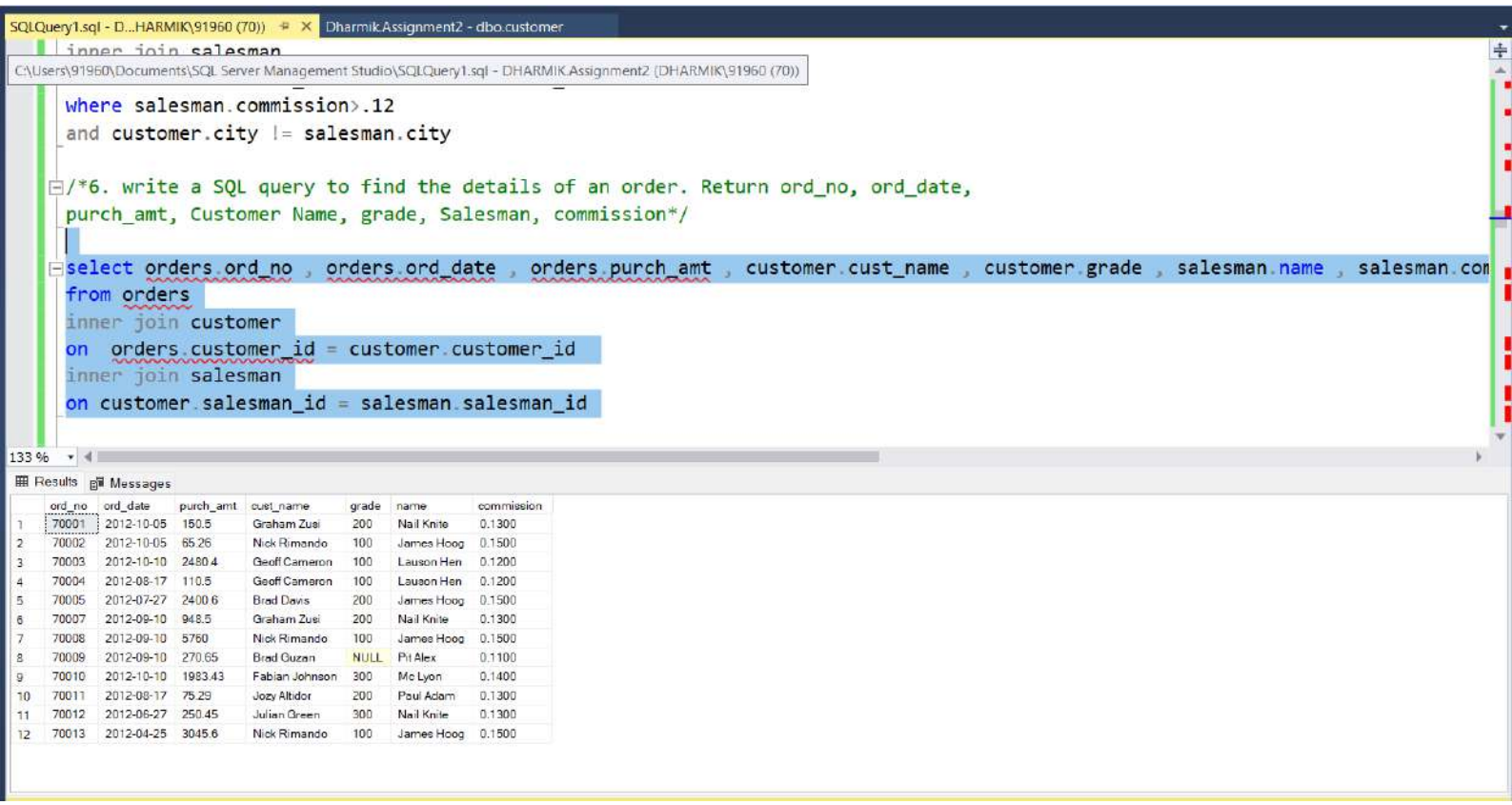
SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
their customers live and have received a commission of more than 12% from the
company. Return Customer Name, customer city, Salesman, salesman city,
commission*/

select customer.cust_name , customer.city , salesman.name , salesman.city ,salesman.commission
from customer
inner join salesman
on customer.salesman_id = salesman.salesman_id
where salesman.commission>.12
and customer.city != salesman.city

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

select orders.ord_no , orders.ord_date , orders.purch_amt , customer.cust_name , customer.grade , salesman.name , salesman.com
```

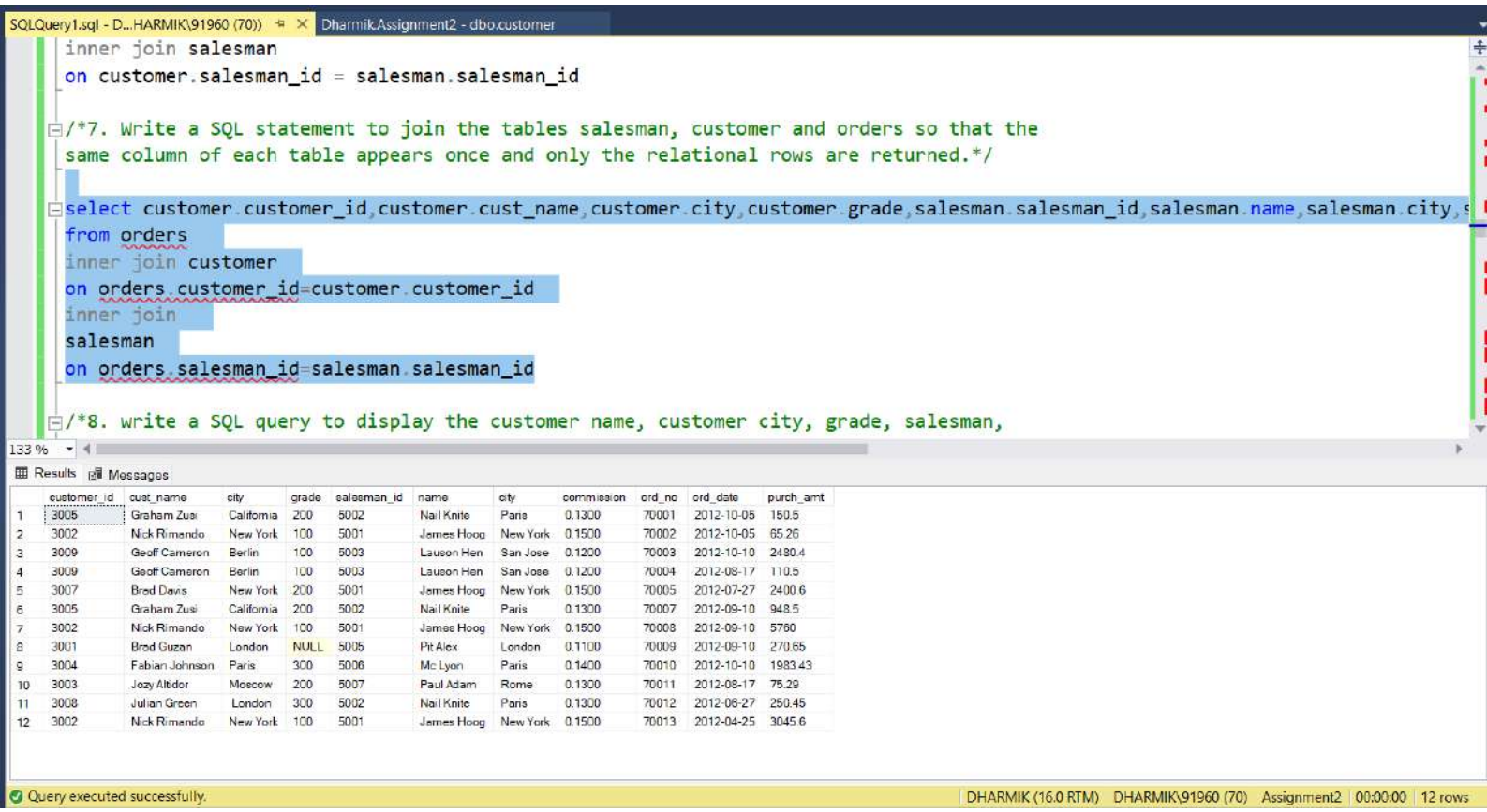
133 %

Results | Messages

	cust_name	city	name	city	commission
1	Jozy Altidor	Moscow	Paul Adam	Rome	0.1300
2	Graham Zusi	California	Nail Knite	Paris	0.1300
3	Julian Green	London	Nail Knite	Paris	0.1300



	ord_no	ord_date	purch_amt	cust_name	grade	name	commission
1	70001	2012-10-05	150.5	Graham Zusi	200	Nail Knite	0.1300
2	70002	2012-10-05	65.26	Nick Rimando	100	James Hoog	0.1500
3	70003	2012-10-10	2480.4	Geoff Cameron	100	Lauson Hen	0.1200
4	70004	2012-08-17	110.5	Geoff Cameron	100	Lauson Hen	0.1200
5	70005	2012-07-27	2400.6	Brad Davis	200	James Hoog	0.1500
6	70007	2012-09-10	948.5	Graham Zusi	200	Nail Knite	0.1300
7	70008	2012-09-10	5760	Nick Rimando	100	James Hoog	0.1500
8	70009	2012-09-10	270.65	Brad Guzan	NULL	Pit Alex	0.1100
9	70010	2012-10-10	1983.43	Fabian Johnson	300	Mc Lyon	0.1400
10	70011	2012-08-17	75.29	Jozy Altidor	200	Paul Adam	0.1300
11	70012	2012-06-27	250.45	Julian Green	300	Nail Knite	0.1300
12	70013	2012-04-25	3045.6	Nick Rimando	100	James Hoog	0.1500



	customer_id	cust_name	city	grade	salesman_id	name	city	commission	ord_no	ord_date	purch_amt
1	3005	Graham Zusi	California	200	5002	Nail Knite	Paris	0.1300	70001	2012-10-05	150.5
2	3002	Nick Rimando	New York	100	5001	James Hoog	New York	0.1500	70002	2012-10-05	65.26
3	3009	Geoff Cameron	Berlin	100	5003	Lauson Hen	San Jose	0.1200	70003	2012-10-10	2480.4
4	3009	Geoff Cameron	Berlin	100	5003	Lauson Hen	San Jose	0.1200	70004	2012-08-17	110.5
5	3007	Brad Davis	New York	200	5001	James Hoog	New York	0.1500	70005	2012-07-27	2400.6
6	3005	Graham Zusi	California	200	5002	Nail Knite	Paris	0.1300	70007	2012-09-10	948.5
7	3002	Nick Rimando	New York	100	5001	James Hoog	New York	0.1500	70008	2012-09-10	5760
8	3001	Brad Guzan	London	NULL	5005	Pit Alex	London	0.1100	70009	2012-09-10	270.65
9	3004	Fabian Johnson	Paris	300	5006	Mc Lyon	Paris	0.1400	70010	2012-10-10	1983.43
10	3003	Jozy Altidor	Moscow	200	5007	Paul Adam	Rome	0.1300	70011	2012-08-17	75.29
11	3008	Julian Green	London	300	5002	Nail Knite	Paris	0.1300	70012	2012-06-27	250.45
12	3002	Nick Rimando	New York	100	5001	James Hoog	New York	0.1500	70013	2012-04-25	3045.6

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
on orders.salesman_id=salesman.salesman_id

/*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.*/

select  customer.cust_name , customer.city , customer.grade , salesman.name , salesman.city
from customer
join salesman
on customer.salesman_id = salesman.salesman_id
order by customer.customer_id

/*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.*/
```

133 %

Results | Messages

	cust_name	city	grade	name	city
1	Brad Guzan	London	NULL	Pit Alex	London
2	Nick Rimando	New York	100	James Hoog	New York
3	Jozy Altidor	Moscow	200	Paul Adam	Rome
4	Fabian Johnson	Paris	300	Mc Lyon	Paris
5	Graham Zusi	California	200	Nail Knite	Paris
6	Brad Davis	New York	200	James Hoog	New York
7	Julian Green	London	300	Nail Knite	Paris
8	Geoff Cameron	Berlin	100	Lauson Hen	San Jose

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
order by customer.customer_id

/*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.*/

select customer.cust_name , customer.city , customer.grade , salesman.name , salesman.city
from customer
left join salesman
on customer.salesman_id = salesman.salesman_id
where customer.grade<300
order by customer.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
```

133 %

Results | Messages

	cust_name	city	grade	name	city
1	Nick Rimando	New York	100	James Hoog	New York
2	Jozy Altidor	Moscow	200	Paul Adam	Rome
3	Graham Zusi	California	200	Nail Knite	Paris
4	Brad Davis	New York	200	James Hoog	New York
5	Geoff Cameron	Berlin	100	Lauson Hen	San Jose

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

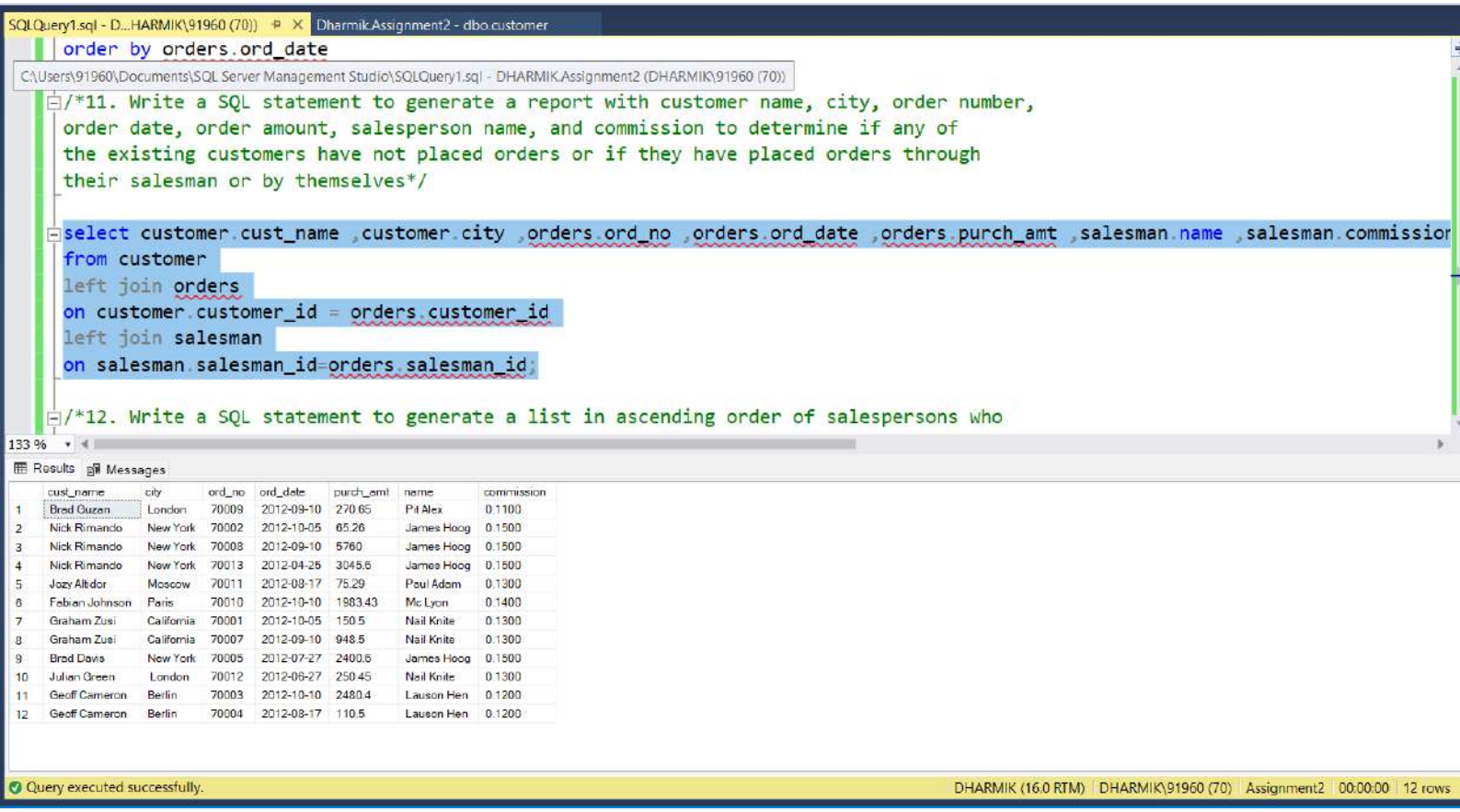
```
/*10. Write a SQL statement to make a report with customer name, city, order number,
order date, and order amount in ascending order according to the order date to
determine whether any of the existing customers have placed an order or not*/

select customer.cust_name , customer.city , orders.ord_no , orders.ord_date , orders.purch_amt
from customer
left join orders
on customer.customer_id = orders.customer_id
order by orders.ord_date

/*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*/
```

133 %

Results | Messages

	cust_name	city	ord_no	ord_date	purch_amt
1	Nick Rimando	New York	70013	2012-04-25	3045.6
2	Julian Green	London	70012	2012-06-27	250.45
3	Brad Davis	New York	70005	2012-07-27	2400.6
4	Jozy Altidor	Moscow	70011	2012-08-17	75.29
5	Geoff Cameron	Berlin	70004	2012-08-17	110.5
6	Brad Guzan	London	70009	2012-09-10	270.65
7	Graham Zusi	California	70007	2012-09-10	948.5
8	Nick Rimando	New York	70008	2012-09-10	5760
9	Graham Zusi	California	70001	2012-10-05	150.5
10	Nick Rimando	New York	70002	2012-10-05	65.26
11	Fabian Johnson	Paris	70010	2012-10-10	1983.43
12	Geoff Cameron	Berlin	70003	2012-10-10	2480.4

Query executed successfully. | DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00:00:00 | 12 rows


SQLQuery1.sql - D...HARMIK\91960 (70))
Dharmik.Assignment2 - dbo.customer
order by orders.ord_date
C:\Users\91960\Documents\SQL Server Management Studio\SQLQuery1.sql - DHARMIK.Assignment2 (DHARMIK\91960 (70))
/*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*/
select customer.cust_name ,customer.city ,orders.ord_no ,orders.ord_date ,orders.purch_amt ,salesman.name ,salesman.commissior
from customer
left join orders
on customer.customer_id = orders.customer_id
left join salesman
on salesman.salesman_id=orders.salesman_id;
/*12. Write a SQL statement to generate a list in ascending order of salespersons who
133 %
Results
Messages
cust_name city ord_no ord_date purch_amt name commission
1 Brad Guzan London 70009 2012-09-10 270.65 Pit Alex 0.1100
2 Nick Rimando New York 70002 2012-10-05 65.26 James Hoog 0.1500
3 Nick Rimando New York 70008 2012-09-10 5760 James Hoog 0.1500
4 Nick Rimando New York 70013 2012-04-25 3045.6 James Hoog 0.1500
5 Jozy Altidor Moscow 70011 2012-08-17 75.29 Paul Adam 0.1300
6 Fabian Johnson Paris 70010 2012-10-10 1983.43 Mc Lyon 0.1400
7 Graham Zusi California 70001 2012-10-05 150.5 Nail Knite 0.1300
8 Graham Zusi California 70007 2012-09-10 948.5 Nail Knite 0.1300
9 Brad Davis New York 70005 2012-07-27 2400.6 James Hoog 0.1500
10 Julian Green London 70012 2012-06-27 250.45 Nail Knite 0.1300
11 Geoff Cameron Berlin 70003 2012-10-10 2480.4 Lauson Hen 0.1200
12 Geoff Cameron Berlin 70004 2012-08-17 110.5 Lauson Hen 0.1200
Query executed successfully.
DHARMIK (16.0 RTM) DHARMIK\91960 (70) Assignment2 00:00:00 12 rows

SQLQuery1.sql - D...HARMIK\91960 (70)) | DharmikAssignment2 - dbo.customer

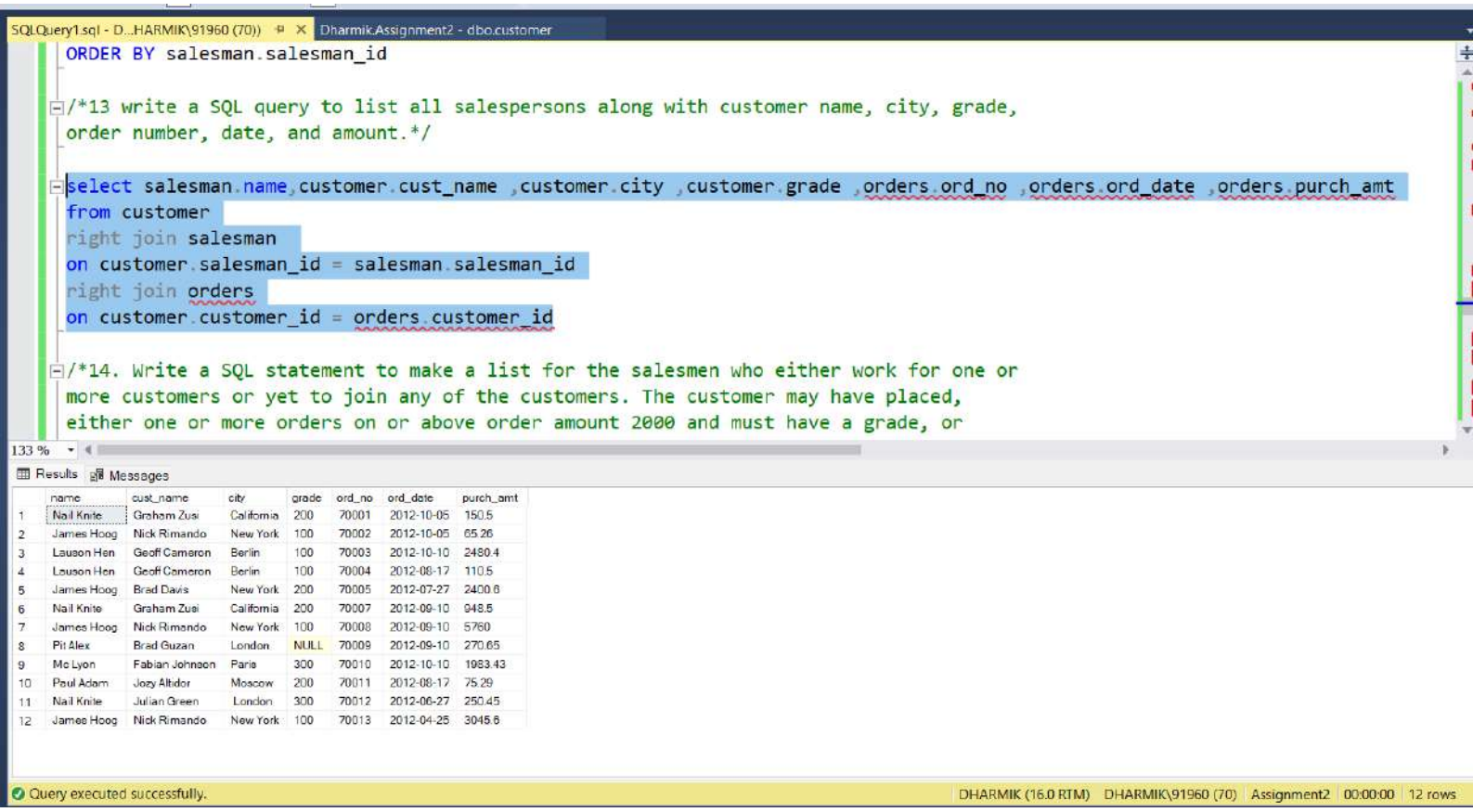
```
on salesman.salesman_id=orders.salesman_id;

/*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*/

SELECT Salesman.name
FROM customer
RIGHT OUTER JOIN salesman
ON salesman.salesman_id=customer.salesman_id
ORDER BY salesman.salesman_id

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

select salesman.name,customer.cust_name ,customer.city ,customer.grade ,orders.ord_no ,orders.ord_date ,orders.purch_amt
```

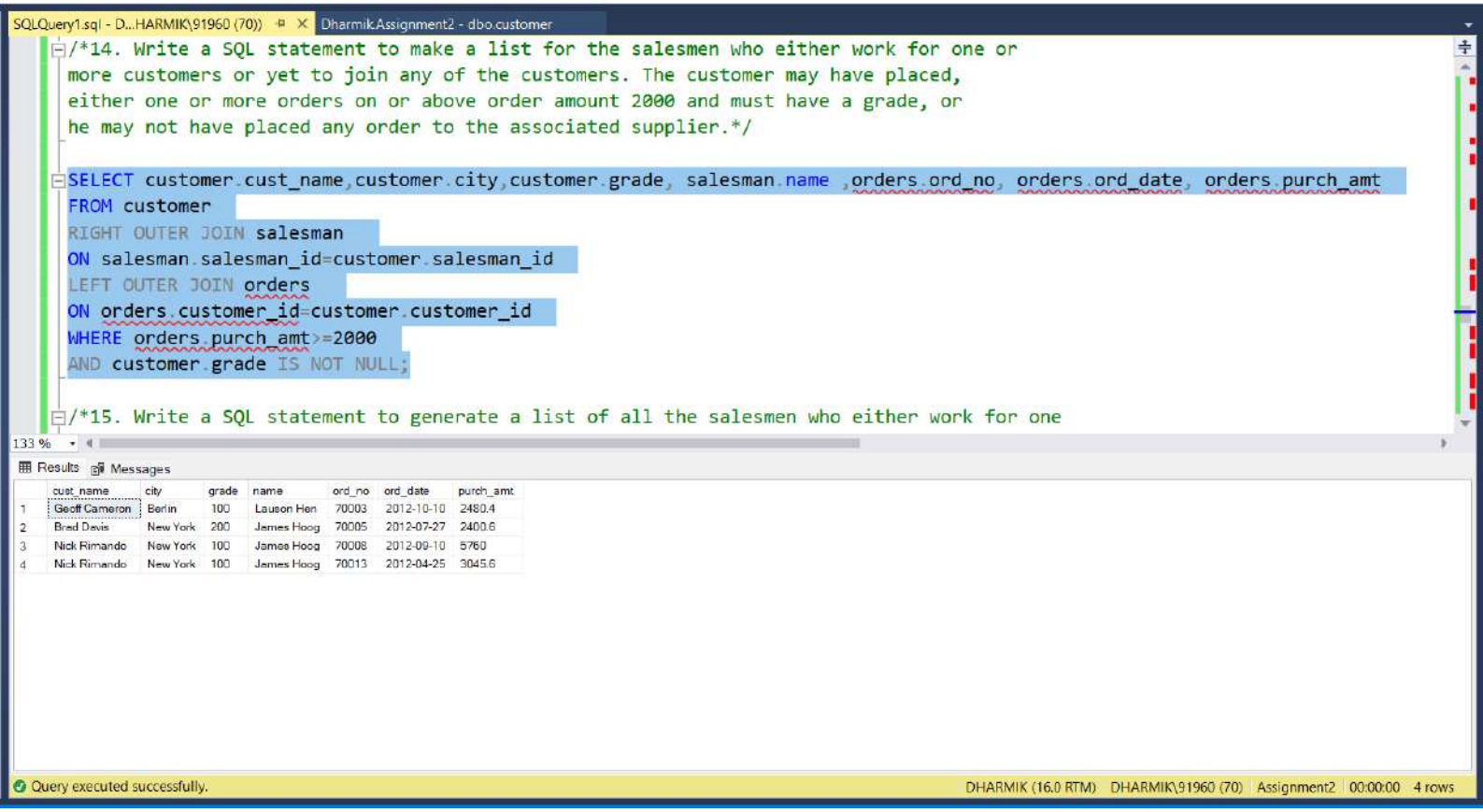
133 %

Results | Messages

	name
1	James Hoog
2	James Hoog
3	Nail Knite
4	Nail Knite
5	Lauson Hen
6	Pit Alex
7	Mc Lyon
8	Paul Adam

Query executed successfully. DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00:00:00 | 8 rows


SQLQuery1.sql - D...HARMIK\91960 (70))
Dharmik.Assignment2 - dbo.customer
ORDER BY salesman.salesman_id
/*13 write a SQL query to list all salespersons along with customer name, city, grade,
order number, date, and amount.*/
select salesman.name,customer.cust_name ,customer.city ,customer.grade ,orders.ord_no ,orders.ord_date ,orders.purch_amt
from customer
right join salesman
on customer.salesman_id = salesman.salesman_id
right join orders
on customer.customer_id = orders.customer_id
/*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
133 %
Results
Messages
name cust_name city grade ord_no ord_date purch_amt
1 Nail Knite Graham Zusi California 200 70001 2012-10-05 150.5
2 James Hoog Nick Rimando New York 100 70002 2012-10-05 65.26
3 Lauson Hen Geoff Cameron Berlin 100 70003 2012-10-10 2480.4
4 Lauson Hen Geoff Cameron Berlin 100 70004 2012-08-17 110.5
5 James Hoog Brad Davis New York 200 70005 2012-07-27 2400.6
6 Nail Knite Graham Zusi California 200 70007 2012-09-10 948.5
7 James Hoog Nick Rimando New York 100 70008 2012-09-10 5760
8 Pit Alex Brad Guzan London NULL 70009 2012-09-10 270.65
9 Mc Lyon Fabian Johnson Paris 300 70010 2012-10-10 1983.43
10 Paul Adam Jozy Altidor Moscow 200 70011 2012-08-17 75.29
11 Nail Knite Julian Green London 300 70012 2012-06-27 250.45
12 James Hoog Nick Rimando New York 100 70013 2012-04-25 3045.6
Query executed successfully.
DHARMIK (16.0 RTM)
DHARMIK\91960 (70)
Assignment2
00:00:00
12 rows


SQLQuery1.sql - D...HARMIK\91960 (70))
DharmikAssignment2 - dbo.customer
/*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.*/
SELECT customer.cust_name,customer.city,customer.grade, salesman.name ,orders.ord_no, orders.ord_date, orders.purch_amt
FROM customer
RIGHT OUTER JOIN salesman
ON salesman.salesman_id=customer.salesman_id
LEFT OUTER JOIN orders
ON orders.customer_id=customer.customer_id
WHERE orders.purch_amt>=2000
AND customer.grade IS NOT NULL;
/*15. Write a SQL statement to generate a list of all the salesmen who either work for one
133 %
Results
Messages
cust_name city grade name ord_no ord_date purch_amt
1 Geoff Cameron Berlin 100 Lauson Hen 70003 2012-10-10 2480.4
2 Brad Davis New York 200 James Hoog 70005 2012-07-27 2400.6
3 Nick Rimando New York 100 James Hoog 70008 2012-09-10 5760
4 Nick Rimando New York 100 James Hoog 70013 2012-04-25 3045.6
Query executed successfully.
DHARMIK (16.0 RTM)
DHARMIK\91960 (70)
Assignment2
00:00:00
4 rows

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
AND customer.grade IS NOT NULL;

/*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.*/

SELECT customer.cust_name,customer.city,customer.grade, salesman.name ,orders.ord_no, orders.ord_date, orders.purch_amt
FROM customer
RIGHT OUTER JOIN salesman
ON salesman.salesman_id=customer.salesman_id
LEFT OUTER JOIN orders
ON orders.customer_id=customer.customer_id
WHERE orders.purch_amt>=2000
AND customer.grade IS NOT NULL;
```

133 %

Results | Messages

	cust_name	city	grade	name	ord_no	ord_date	purch_amt
1	Geoff Cameron	Berlin	100	Lauson Hen	70003	2012-10-10	2480.4
2	Brad Davis	New York	200	James Hoog	70005	2012-07-27	2400.6
3	Nick Rimando	New York	100	James Hoog	70008	2012-09-10	5760
4	Nick Rimando	New York	100	James Hoog	70013	2012-04-25	3045.6

Query executed successfully. DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00:00:00 | 4 rows

SQLQuery1.sql - D...HARMIK\91960 (70)) | DharmikAssignment2 - dbo.customer

Execute (F5)

```
/*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.*/

SELECT customer.cust_name,customer.city, orders.ord_no,
orders.ord_date,orders.purch_amt
FROM customer
FULL OUTER JOIN orders
ON customer.customer_id=orders.customer_id
WHERE customer.grade IS NOT NULL

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

133 %

Results | Messages

	cust_name	city	ord_no	ord_date	purch_amt
1	Nick Rimando	New York	70002	2012-10-05	65.26
2	Nick Rimando	New York	70008	2012-09-10	5760
3	Nick Rimando	New York	70013	2012-04-25	3045.6
4	Jozy Altidor	Moscow	70011	2012-08-17	75.29
5	Fabian Johnson	Paris	70010	2012-10-10	1983.43
6	Graham Zusi	California	70001	2012-10-05	150.5
7	Graham Zusi	California	70007	2012-09-10	948.5
8	Brad Davis	New York	70005	2012-07-27	2400.6
9	Julian Green	London	70012	2012-06-27	250.45
10	Geoff Cameron	Berlin	70003	2012-10-10	2480.4
11	Geoff Cameron	Berlin	70004	2012-08-17	110.5

Query executed successfully. DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00:00:00

SQLQuery1.sql - D...HARMIK\91960 (70)) | DharmikAssignment2 - dbo.customer

```
FULL OUTER JOIN orders
ON customer.customer_id=orders.customer_id
WHERE customer.grade IS NOT NULL

/*17. Write a SQL query to combine each row of the salesman table with each row of the
customer table*/
select *
from salesman
cross join customer

/*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
```

133 %

Results | Messages

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.1500	3001	Brad Guzan	London	NULL	5005
2	5001	James Hoog	New York	0.1500	3002	Nick Rimando	New York	100	5001
3	5001	James Hoog	New York	0.1500	3003	Jozy Altidor	Moscow	200	5007
4	5001	James Hoog	New York	0.1500	3004	Fabian Johnson	Paris	300	5006
5	5001	James Hoog	New York	0.1500	3005	Graham Zusi	California	200	5002
6	5001	James Hoog	New York	0.1500	3007	Brad Davis	New York	200	5001
7	5001	James Hoog	New York	0.1500	3008	Julian Green	London	300	5002
8	5001	James Hoog	New York	0.1500	3009	Geoff Cameron	Berlin	100	5003
9	5002	Nail Knite	Paris	0.1300	3001	Brad Guzan	London	NULL	5005
10	5002	Nail Knite	Paris	0.1300	3002	Nick Rimando	New York	100	5001
11	5002	Nail Knite	Paris	0.1300	3003	Jozy Altidor	Moscow	200	5007
12	5002	Nail Knite	Paris	0.1300	3004	Fabian Johnson	Paris	300	5006
13	5002	Nail Knite	Paris	0.1300	3005	Graham Zusi	California	200	5002
14	5002	Nail Knite	Paris	0.1300	3007	Brad Davis	New York	200	5001
15	5002	Nail Knite	Paris	0.1300	3008	Julian Green	London	300	5002
16	5002	Nail Knite	Paris	0.1300	3009	Geoff Cameron	Berlin	100	5003
17	5003	Lauson Hen	San Jose	0.1200	3001	Brad Guzan	London	NULL	5005
18	5003	Lauson Hen	San Jose	0.1200	3002	Nick Rimando	New York	100	5001

Query executed successfully. | DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00:00:00 | 48 rows

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
/*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*/

SELECT *
FROM salesman
CROSS JOIN customer
WHERE salesman.city = customer.city;

/*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
```

133 %

Results | Messages

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5005	Pit Alex	London	0.1100	3001	Brad Guzan	London	NULL	5005
2	5001	James Hoog	New York	0.1500	3002	Nick Rimando	New York	100	5001
3	5002	Nail Knite	Paris	0.1300	3004	Fabian Johnson	Paris	300	5006
4	5006	Mc Lyon	Paris	0.1400	3004	Fabian Johnson	Paris	300	5006
5	5001	James Hoog	New York	0.1500	3007	Brad Davis	New York	200	5001

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```sql
/*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*/

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

/*20. Write a SQL statement to make a Cartesian product between salesman and
```

133 %

Results | Messages

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.1500	3002	Nick Rimando	New York	100	5001
2	5001	James Hoog	New York	0.1500	3003	Jozy Altidor	Moscow	200	5007
3	5001	James Hoog	New York	0.1500	3004	Fabian Johnson	Paris	300	5006
4	5001	James Hoog	New York	0.1500	3005	Graham Zusi	California	200	5002
5	5001	James Hoog	New York	0.1500	3007	Brad Davis	New York	200	5001
6	5001	James Hoog	New York	0.1500	3008	Julian Green	London	300	5002
7	5001	James Hoog	New York	0.1500	3009	Geoff Cameron	Berlin	100	5003
8	5002	Nail Knite	Paris	0.1300	3002	Nick Rimando	New York	100	5001
9	5002	Nail Knite	Paris	0.1300	3003	Jozy Altidor	Moscow	200	5007
10	5002	Nail Knite	Paris	0.1300	3004	Fabian Johnson	Paris	300	5006
11	5002	Nail Knite	Paris	0.1300	3005	Graham Zusi	California	200	5002
12	5002	Nail Knite	Paris	0.1300	3007	Brad Davis	New York	200	5001
13	5002	Nail Knite	Paris	0.1300	3008	Julian Green	London	300	5002
14	5002	Nail Knite	Paris	0.1300	3009	Geoff Cameron	Berlin	100	5003
15	5003	Lauson Hen	San Jose	0.1200	3002	Nick Rimando	New York	100	5001
16	5003	Lauson Hen	San Jose	0.1200	3003	Jozy Altidor	Moscow	200	5007
17	5003	Lauson Hen	San Jose	0.1200	3004	Fabian Johnson	Paris	300	5006
18	5003	Lauson Hen	San Jose	0.1200	3005	Graham Zusi	California	200	5002
19	5003	Lauson Hen	San Jose	0.1200	3007	Brad Davis	New York	200	5001

Query executed successfully.

DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00

SQLQuery1.sql - D...HARMIK\91960 (70)) | Dharmik.Assignment2 - dbo.customer

```
AND customer.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 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*/

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

133 %

Results | Messages

	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.1500	3003	Jozy Altidor	Moscow	200	5007
2	5001	James Hoog	New York	0.1500	3004	Fabian Johnson	Paris	300	5006
3	5001	James Hoog	New York	0.1500	3005	Graham Zusi	California	200	5002
4	5001	James Hoog	New York	0.1500	3008	Julian Green	London	300	5002
5	5001	James Hoog	New York	0.1500	3009	Geoff Cameron	Berlin	100	5003
6	5002	Nail Knite	Paris	0.1300	3002	Nick Rimando	New York	100	5001
7	5002	Nail Knite	Paris	0.1300	3003	Jozy Altidor	Moscow	200	5007
8	5002	Nail Knite	Paris	0.1300	3005	Graham Zusi	California	200	5002
9	5002	Nail Knite	Paris	0.1300	3007	Brad Davis	New York	200	5001
10	5002	Nail Knite	Paris	0.1300	3008	Julian Green	London	300	5002
11	5002	Nail Knite	Paris	0.1300	3009	Geoff Cameron	Berlin	100	5003
12	5003	Lauson Hen	San Jose	0.1200	3002	Nick Rimando	New York	100	5001
13	5003	Lauson Hen	San Jose	0.1200	3003	Jozy Altidor	Moscow	200	5007
14	5003	Lauson Hen	San Jose	0.1200	3004	Fabian Johnson	Paris	300	5006
15	5003	Lauson Hen	San Jose	0.1200	3005	Graham Zusi	California	200	5002
16	5003	Lauson Hen	San Jose	0.1200	3007	Brad Davis	New York	200	5001

Query executed successfully. | DHARMIK (16.0 RTM) | DHARMIK\91960 (70) | Assignment2 | 00:0