SQL ASSIGNMENT 2

CREATE TABLE [salesman]

(

Salesman\_id INT NOT NULL IDENTITY(5001,1) PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

City VARCHAR(50),

Commission DECIMAL(2,2));

INSERT INTO [salesman]

VALUES('James Hoog','New York',0.15),

('Nail Knite','Paris',0.13),

('Pit Alex ','London ',0.11),

('Mc Lyon','Paris',0.14),

('Paul Adam','Rome',0.13),

('Lauson Hen','San Jose',0.12)

('Lason ','San Jose',0.18);

SELECT \* FROM salesman



CREATE TABLE [custmer](

customer\_id INT NOT NULL IDENTITY(3001,1) PRIMARY KEY,

cust\_name VARCHAR(50) NOT NULL,

city VARCHAR(50),

grade INT ,

salesman\_id INT NOT NULL);

INSERT INTO [custmer]

VALUES( 'Nick Rimando', 'New York', 100 ,5001),

( 'Brad Davis', 'New York', 200 ,5001),

( 'Graham Zusi', 'California', 200 ,5002),

( 'Julian Green ', 'London', 300 ,5002),

( 'Fabian Johnson', 'Paris', 300 ,5006),

( 'Geoff Cameron', 'Berlin', 100 ,5003),

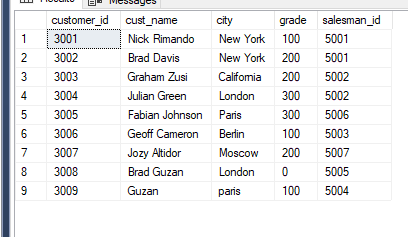
( 'Jozy Altidor', 'Moscow', 200 ,5007),

( 'Brad Guzan', 'London', '',5005);

INSERT INTO [custmer]

VALUES( 'Guzan', 'paris', 100,5004);

SELECT \* FROM customer



CREATE TABLE [orders](

ord\_no INT NOT NULL IDENTITY(70001,1) PRIMARY KEY ,

purch\_amt DECIMAL(10,4),

ord\_date DATE,

customer\_id INT NOT NULL,

salesman\_id INT NOT NULL);

INSERT INTO [orders]

VALUES(150.5 ,'2012-10-05' ,3005, 5002),

( 270.65,'2012-09-10',3001,5005),

(65.26,'2012-10-05',3002,5001),

(110.5,'2012-08-17',3009,5003),

(948.5 , '2012-09-10',3005 ,5002),

(2400.6,'2012-07-27' ,3007 , 5001),

( 5760,'2012-09-10',3002, 5001),

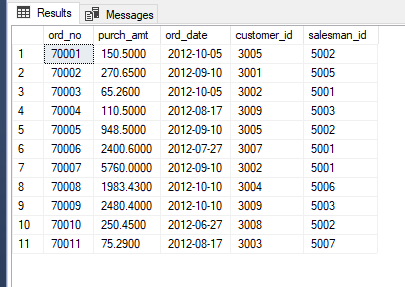
(1983.43, '2012-10-10',3004, 5006),

(2480.4 ,'2012-10-10' ,3009, 5003),

( 250.45,'2012-06-27', 3008 , 5002),

(75.29 ,'2012-08-17',3003, 5007);

SELECT \* FROM [orders]



**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,custmer.cust\_name,salesman.city FROM salesman INNER JOIN custmer ON salesman.City=custmer.city;

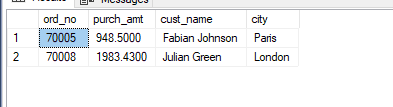


**2.** write a SQL query to find those orders where the order amount exists between 500

and 2000. Return ord\_no, purch\_amt, cust\_name, city

SELECT orders.ord\_no,orders.purch\_amt,custmer.cust\_name,custmer.city FROM custmer INNER JOIN orders ON orders.customer\_id=custmer.customer\_id

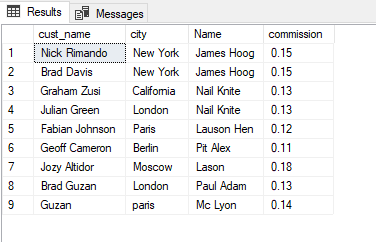
WHERE orders.purch\_amt BETWEEN 500 AND 2000;



**3.** write a SQL query to find the salesperson(s) and the customer(s) he represents.

Return Customer Name, city, Salesman, commission

SELECT custmer.cust\_name,custmer.city,salesman.Name,salesman.commission FROM custmer INNER JOIN salesman ON salesman.salesman\_id=custmer.salesman\_id;

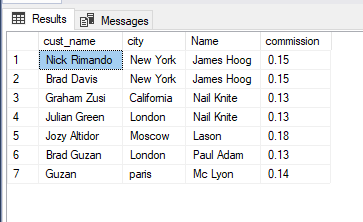


**4.** write a SQL query to find salespeople who received commissions of more than 12

percent from the company. Return Customer Name, customer city, Salesman,

commission.

SELECT c.cust\_name,c.city,s.Name,s.commission FROM custmer c INNER JOIN salesman s ON c.salesman\_id=s.Salesman\_id WHERE s.Commission>0.12;



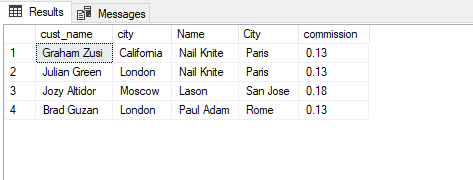
**5.** write a SQL query to locate those salespeople who do not live in the same city where

their customers live and have received a commission of more than 12% from the

company. Return Customer Name, customer city, Salesman, salesman city,

commission

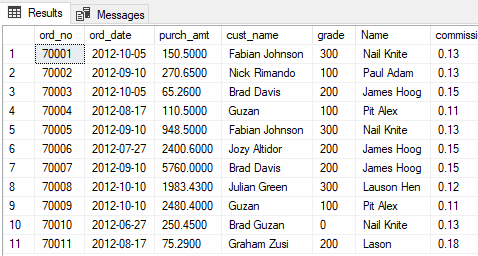
SELECT c.cust\_name,c.city,s.Name,s.City,s.commission FROM custmer c INNER JOIN salesman s ON c.salesman\_id=s.Salesman\_id WHERE s.City != c.city AND s.Commission>.12;



**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 o.ord\_no,o.ord\_date,o.purch\_amt,c.cust\_name,c.grade,s.Name,s.commission FROM orders o,custmer c, salesman s WHERE o.customer\_id =c.customer\_id AND o.salesman\_id =s.Salesman\_id;



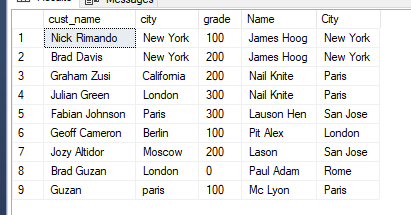
**7.** Write a SQL statement to join the tables salesman, customer and orders so that the

same column of each table appears once and only the relational rows are returned.

**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 c.cust\_name,c.city,c.grade,s.Name,s.City FROM custmer c INNER JOIN salesman s ON c.salesman\_id=s.Salesman\_id ORDER BY c.customer\_id ASC;

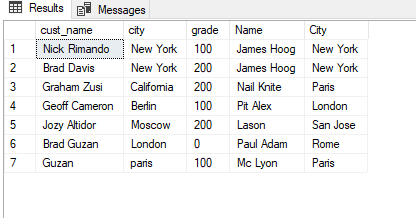


**9**. write a SQL query to find those customers with a grade less than 300. Return

cust\_name, customer city, grade, Salesman, salesmancity. The result should be

ordered by ascending customer\_id.

SELECT c.cust\_name,c.city,c.grade,s.Name,s.City FROM custmer c INNER JOIN salesman s ON c.salesman\_id=s.Salesman\_id WHERE c.grade <300 ORDER BY c.customer\_id ASC ;

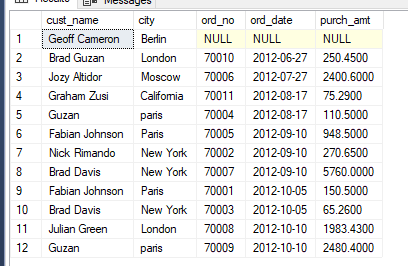


**10.** Write a SQL statement to make a report with customer name, city, order number,

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 c.cust\_name,c.city,o.ord\_no,o.ord\_date,o.purch\_amt FROM custmer c LEFT OUTER JOIN orders o ON c.customer\_id=o.customer\_id ORDER BY o.ord\_date ASC;



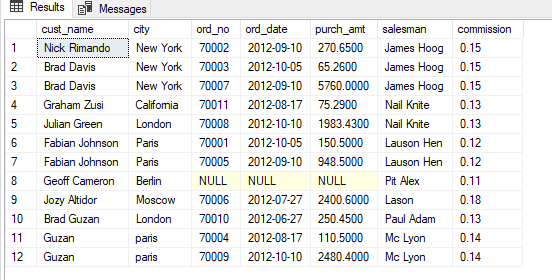
**11.** Write a SQL statement to generate a report with customer name, city, order number,

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 c.cust\_name,c.city,o.ord\_no,o.ord\_date,o.purch\_amt,s.Name AS salesman,s.commission FROM custmer c LEFT OUTER JOIN orders o ON c.customer\_id=o.customer\_id LEFT OUTER JOIN salesman s ON c.salesman\_id=s.salesman\_id;



**12.** Write a SQL statement to generate a list in ascending order of salespersons who

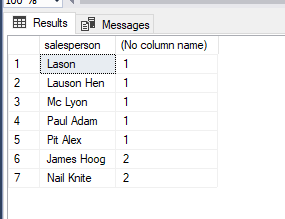
work either for one or more customers or have not yet joined any of the customers

SELECT s.Name AS salesperson,COUNT(c.customer\_id) FROM salesman s LEFT OUTER JOIN custmer c

ON s.Salesman\_id=c.salesman\_id

group by s.Name

ORDER BY COUNT(c.customer\_id) ASC;



**13.** write a SQL query to list all salespersons along with customer name, city, grade,

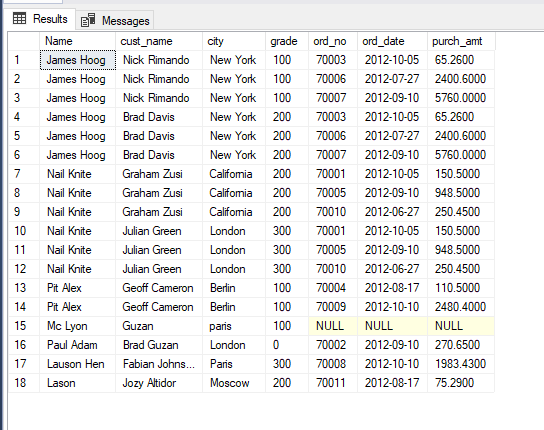
order number, date, and amount.

SELECT s.Name,c.cust\_name,c.city,c.grade,o.ord\_no,o.ord\_date,o.purch\_amt

FROM salesman s LEFT OUTER JOIN custmer c

ON s.Salesman\_id=c.salesman\_id

LEFT OUTER JOIN orders o ON s.Salesman\_id=o.salesman\_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

he may not have placed any order to the associated supplier.

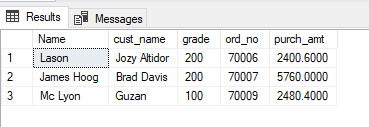
SELECT s.Name ,c.cust\_name,c.grade,o.ord\_no,o.purch\_amt

FROM salesman s

LEFT OUTER JOIN custmer c

ON s.Salesman\_id = c.salesman\_id

LEFT OUTER JOIN orders o ON o.customer\_id = c.customer\_id WHERE o.purch\_amt >2000 AND c.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.

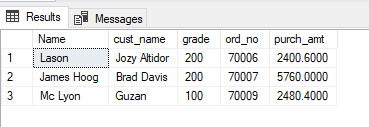
SELECT s.Name ,c.cust\_name,c.grade,o.ord\_no,o.purch\_amt

FROM salesman s

LEFT OUTER JOIN custmer c

ON s.Salesman\_id = c.salesman\_id

LEFT OUTER JOIN orders o ON o.customer\_id = c.customer\_id WHERE o.purch\_amt >2000 AND c.grade is not null;



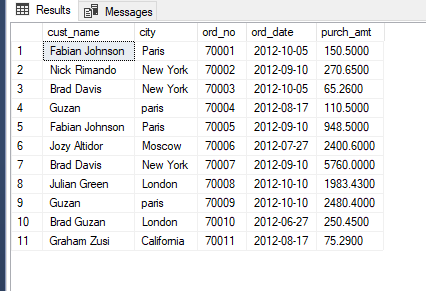
**16.** Write a SQL statement to generate a report with the customer name, city, order no.

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.

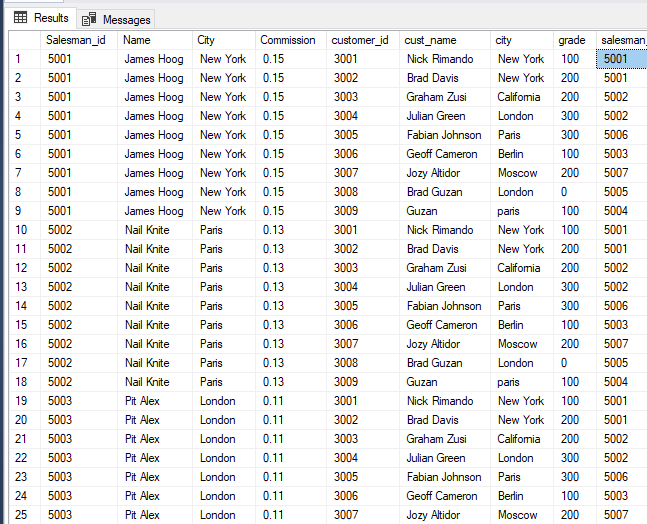
SELECT c.cust\_name,c.city,o.ord\_no,o.ord\_date,o.purch\_amt FROM custmer c INNER JOIN orders o ON c.customer\_id=o.customer\_id WHERE c.grade IS NOT NULL ;



**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 custmer;

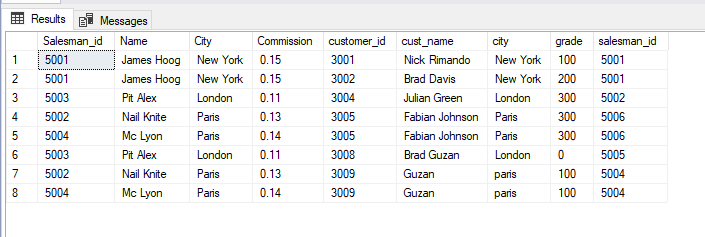


**18.** Write a SQL statement to create a Cartesian product between salesperson and

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 custmer WHERE salesman.City=custmer.city;

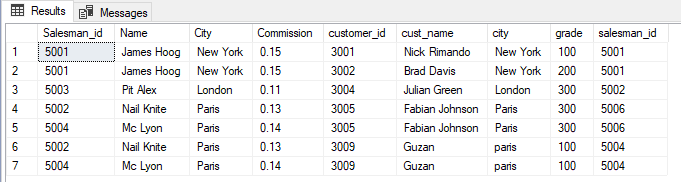


**19.** Write a SQL statement to create a Cartesian product between salesperson and

customer, i.e. each salesperson will appear for every customer and vice versa for

those salesmen who belong to a city and customers who require a grade

SELECT \* FROM salesman CROSS JOIN custmer WHERE salesman.City =custmer.city AND custmer.grade != 0;



**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 custmer WHERE salesman.City != custmer.city AND custmer.grade != 0;

