

SQL ASSIGNMENT

Table SALESPeOPLE

```
mysql> use sales;
Database changed
mysql> CREATE TABLE SALESPeOPLE (
  ->   SNUM INT PRIMARY KEY,
  ->   SNAME CHAR(50),
  ->   CITY CHAR(50),
  ->   COMM FLOAT
  -> );
ERROR 1050 (42S01): Table 'salespeople' already exists
mysql> SELECT * FROM SALESPeOPLE;
Empty set (0.241 sec)

mysql> INSERT INTO SALESPeOPLE (SNUM, SNAME, CITY, COMM)
  -> VALUES
  -> (1001, 'Peel', 'London', 0.12),
  -> (1002, 'Serres', 'San Jose', 0.13),
  -> (1004, 'Motika', 'London', 0.11),
  -> (1007, 'Rafin', 'Barcelona', 0.15),
  -> (1003, 'Axelrod', 'New York', 0.10);
Query OK, 5 rows affected (0.135 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM SALESPeOPLE;
+-----+-----+-----+-----+
| SNUM | SNAME   | CITY      | COMM |
+-----+-----+-----+-----+
| 1001 | Peel    | London    | 0.12 |
| 1002 | Serres  | San Jose  | 0.13 |
| 1004 | Motika  | London    | 0.11 |
| 1007 | Rafin   | Barcelona | 0.15 |
| 1003 | Axelrod | New York  | 0.1  |
+-----+-----+-----+-----+
```

Table CUST

```
mysql> CREATE TABLE CUST (
  ->   CNUM INT PRIMARY KEY,
  ->   CNAME CHAR(50),
  ->   CITY CHAR(50),
  ->   RATING INT,
  ->   SNUM INT
  -> );
Query OK, 0 rows affected (0.487 sec)

mysql> INSERT INTO CUST (CNUM, CNAME, CITY, RATING, SNUM)
  -> VALUES
  -> (2001, 'Hoffman', 'London', 100, 1001),
  -> (2002, 'Giovanne', 'Rome', 200, 1003),
  -> (2003, 'Liu', 'San Jose', 300, 1002),
  -> (2004, 'Grass', 'Brelín', 100, 1002),
  -> (2006, 'Clemens', 'London', 300, 1007),
  -> (2007, 'Pereira', 'Rome', 100, 1004);
Query OK, 6 rows affected (0.092 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM CUST;
+-----+-----+-----+-----+-----+
| CNUM | CNAME   | CITY      | RATING | SNUM |
+-----+-----+-----+-----+-----+
| 2001 | Hoffman | London    | 100    | 1001 |
| 2002 | Giovane | Rome      | 200    | 1003 |
| 2003 | Liu     | San Jose  | 300    | 1002 |
| 2004 | Grass   | Brelín    | 100    | 1002 |
| 2006 | Clemens | London    | 300    | 1007 |
| 2007 | Pereira | Rome      | 100    | 1004 |
+-----+-----+-----+-----+-----+
6 rows in set (0.016 sec)
```

Table ORDERS

```
mysql> CREATE TABLE ORDERS (
->   ONUM INT PRIMARY KEY,
->   AMT DECIMAL(10, 2),
->   ODATE DATE,
->   CNUM INT,
->   SNUM INT
-> );
Query OK, 0 rows affected (0.530 sec)

mysql> INSERT INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM)
-> VALUES
->   (3001, 18.69, '1994-10-03', 2008, 1007),
->   (3003, 767.19, '1994-10-03', 2001, 1001),
->   (3002, 1900.10, '1994-10-03', 2007, 1004),
->   (3005, 5160.45, '1994-10-03', 2003, 1002),
->   (3006, 1098.16, '1994-10-04', 2008, 1007),
->   (3009, 1713.23, '1994-10-04', 2002, 1003),
->   (3007, 75.75, '1994-10-05', 2004, 1002),
->   (3008, 4723.00, '1994-10-05', 2006, 1001),
->   (3010, 1309.95, '1994-10-06', 2004, 1002),
->   (3011, 9891.88, '1994-10-06', 2006, 1001);
Query OK, 10 rows affected (0.301 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

1. Display snum,sname,city and comm of all salespeople.

```
mysql> Select snum, sname, city, comm
-> from salespeople;
+-----+-----+-----+-----+
| snum | sname | city | comm |
+-----+-----+-----+-----+
| 1001 | Peel | London | 0.12 |
| 1002 | Serres | San Jose | 0.13 |
| 1004 | Motika | London | 0.11 |
| 1007 | Rafin | Barcelona | 0.15 |
| 1003 | Axelrod | New York | 0.1 |
+-----+-----+-----+-----+
5 rows in set (0.009 sec)
```

2. Display all snum without duplicates from all orders.

```
mysql> Select distinct snum from orders;
+-----+
| snum |
+-----+
| 1007 |
| 1004 |
| 1001 |
| 1002 |
| 1003 |
+-----+
5 rows in set (0.296 sec)
```

3. Display names and commissions of all salespeople in london.

```
mysql> Select sname, comm from salespeople where city='London';
+-----+-----+
| sname | comm |
+-----+-----+
| Peel  | 0.12 |
| Motika | 0.11 |
+-----+-----+
2 rows in set (0.255 sec)
```

4. All customers with rating of 100.

```
mysql> Select cname from cust where rating =100;
+-----+
| cname |
+-----+
| Hoffman |
| Grass  |
| Pereira |
+-----+
3 rows in set (0.015 sec)
```

5. Produce orderno, amount and date form all rows in the order table.

```
mysql> SELECT onum, amt, odate from orders;
+-----+-----+-----+
| onum | amt      | odate      |
+-----+-----+-----+
| 3001 | 18.69    | 1994-10-03 |
| 3002 | 1900.10  | 1994-10-03 |
| 3003 | 767.19   | 1994-10-03 |
| 3005 | 5160.45  | 1994-10-03 |
| 3006 | 1098.16  | 1994-10-04 |
| 3007 | 75.75    | 1994-10-05 |
| 3008 | 4723.00  | 1994-10-05 |
| 3009 | 1713.23  | 1994-10-04 |
| 3010 | 1309.95  | 1994-10-06 |
| 3011 | 9891.88  | 1994-10-06 |
+-----+-----+-----+
10 rows in set (0.238 sec)
```

6. All customers in San Jose, who have rating more than 200.

```
mysql> Select cname from cust where city='San Jose' or rating>200;
+-----+
| cname |
+-----+
| Liu   |
| Clemens |
+-----+
2 rows in set (0.025 sec)
```

7. All customers who were either located in San Jose or had a rating above 200.

```
mysql> Select cname from cust where rating>200;
+-----+
| cname |
+-----+
| Liu   |
| Clemens |
+-----+
2 rows in set (0.017 sec)
```

8. All orders for more than \$1000.

```
mysql> Select * from orders where amt>1000;
+-----+-----+-----+-----+-----+
| ONUM | AMT   | ODATE   | CNUM | SNUM |
+-----+-----+-----+-----+-----+
| 3002 | 1900.10 | 1994-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1994-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1994-10-04 | 2008 | 1007 |
| 3008 | 4723.00 | 1994-10-05 | 2006 | 1001 |
| 3009 | 1713.23 | 1994-10-04 | 2002 | 1003 |
| 3010 | 1309.95 | 1994-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1994-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
7 rows in set (0.254 sec)
```

9. Names and citires of all salespeople in london with commission above 0.10.

```
mysql> Select sname,city from salespeople where comm>0.10 and city='London';
+-----+-----+
| sname | city |
+-----+-----+
| Peel  | London |
| Motika | London |
+-----+-----+
2 rows in set (0.242 sec)
```

10. All customers excluding those with rating <= 100 unless they are located in Rome.

```
mysql> Select cname from cust where rating<=100 or city='Rome';
+-----+
| cname |
+-----+
| Hoffman |
| Giovanne |
| Grass |
| Pereira |
+-----+
4 rows in set (0.017 sec)
```

11. All salespeople either in Barcelona or in london.

```
mysql> Select sname, city from salespeople where city in ('Barcelona','London');
+-----+-----+
| sname | city |
+-----+-----+
| Peel | London |
| Motika | London |
| Rafin | Barcelona |
+-----+-----+
3 rows in set (0.047 sec)
```

12. All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded)

```
mysql> Select sname, comm from salespeople where comm>0.10 and comm<0.12;
+-----+-----+
| sname | comm |
+-----+-----+
| Peel | 0.12 |
| Motika | 0.11 |
| Axelrod | 0.1 |
+-----+-----+
3 rows in set (0.234 sec)
```

13. All customers with NULL values in city column.

```
mysql> Select cname from cust where city is null;
Empty set (0.165 sec)
```

14. All orders taken on Oct 3Rd and Oct 4th 1994.

```
mysql> Select * from orders where odate in ('03-OCT-84','04-OCT-94');
Empty set, 2 warnings (0.259 sec)
```

15. All customers serviced by peel or Motika.

```
mysql> SELECT cname
-> FROM cust, orders
-> WHERE orders.cnum = cust.cnum
-> AND orders.snum IN (
->     SELECT snum
->     FROM salespeople
->     WHERE sname IN ('Peel', 'Motika')
-> );
+-----+
| cname |
+-----+
| Pereira |
| Hoffman |
| Clemens |
| Clemens |
+-----+
4 rows in set (0.745 sec)
```

16. All customers whose names begin with a letter from A to B.

```
mysql> select cname from cust where cname like 'A%' or cname like 'B%';
Empty set (0.011 sec)
```

17. All orders except those with 0 or NULL value in amt field.

```
mysql> SELECT onum
-> FROM orders
-> WHERE amt != 0 AND amt IS NOT NULL;
+-----+
| onum |
+-----+
| 3001 |
| 3002 |
| 3003 |
| 3005 |
| 3006 |
| 3007 |
| 3008 |
| 3009 |
| 3010 |
| 3011 |
+-----+
10 rows in set (0.243 sec)
```

18. Count the number of salespeople currently listing orders in the order table.

```
mysql> SELECT COUNT(DISTINCT snum) AS total_salespeople
-> FROM orders;
+-----+
| total_salespeople |
+-----+
| 5 |
+-----+
1 row in set (0.217 sec)
```

19. Largest order taken by each salesperson, datewise.

```
mysql> SELECT odate, snum, MAX(amt) AS max_order
-> FROM orders
-> GROUP BY odate, snum
-> ORDER BY odate, snum;
```

odate	snum	max_order
1994-10-03	1001	767.19
1994-10-03	1002	5160.45
1994-10-03	1004	1900.10
1994-10-03	1007	18.69
1994-10-04	1003	1713.23
1994-10-04	1007	1098.16
1994-10-05	1001	4723.00
1994-10-05	1002	75.75
1994-10-06	1001	9891.88
1994-10-06	1002	1309.95

10 rows in set (0.148 sec)

20. Largest order taken by each salesperson with order value more than \$3000.

```
mysql> SELECT odate, snum, MAX(amt) AS max_order
-> FROM orders
-> WHERE amt > 3000
-> GROUP BY odate, snum
-> ORDER BY odate, snum;
```

odate	snum	max_order
1994-10-03	1002	5160.45
1994-10-05	1001	4723.00
1994-10-06	1001	9891.88

3 rows in set (0.017 sec)

21. Which day had the highest total amount ordered.

```
mysql> SELECT odate, amt, snum, cnum
-> FROM orders
-> WHERE amt = (SELECT MAX(amt) FROM orders);
```

odate	amt	snum	cnum
1994-10-06	9891.88	1001	2006

1 row in set (0.065 sec)

22. Count all orders for Oct 3rd.

```
mysql> SELECT COUNT(*) FROM orders WHERE odate = '1994-10-03';
```

COUNT(*)
4

1 row in set (0.264 sec)

23. Count the number of different non NULL city values in customers table.

```
mysql> SELECT COUNT(DISTINCT city) FROM cust;
+-----+
| COUNT(DISTINCT city) |
+-----+
| 4 |
+-----+
1 row in set (0.015 sec)
```

24. Select each customer's smallest order.

```
mysql> SELECT cnum, MIN(amt) FROM orders GROUP BY cnum;
+-----+-----+
| cnum | MIN(amt) |
+-----+-----+
| 2008 | 18.69 |
| 2007 | 1900.10 |
| 2001 | 767.19 |
| 2003 | 5160.45 |
| 2004 | 75.75 |
| 2006 | 4723.00 |
| 2002 | 1713.23 |
+-----+-----+
7 rows in set (0.019 sec)
```

25. First customer in alphabetical order whose name begins with G.

```
mysql> SELECT MIN(cname) FROM cust WHERE cname LIKE 'G%';
+-----+
| MIN(cname) |
+-----+
| Giovanni |
+-----+
1 row in set (0.259 sec)
```

26. Get the output like “ For dd/mm/yy there are ____ orders.

```
mysql> SELECT CONCAT('For ', DATE_FORMAT(odate, '%d/%m/%y'), ' there are ', COUNT(*), ' Orders') AS output FROM orders GROUP BY odate;
+-----+
| output |
+-----+
| For 03/10/94 there are 4 Orders |
| For 04/10/94 there are 2 Orders |
| For 05/10/94 there are 2 Orders |
| For 06/10/94 there are 2 Orders |
+-----+
4 rows in set (0.430 sec)
```


27. Assume that each salesperson has a 12% commission. Produce order no., salesperson no., and amount of salesperson's commission for that order.

```
mysql> SELECT onum, snum, amt, amt * 0.12 AS commission FROM orders ORDER BY snum;
```

onum	snum	amt	commission
3003	1001	767.19	92.0628
3008	1001	4723.00	566.7600
3011	1001	9891.88	1187.0256
3005	1002	5160.45	619.2540
3007	1002	75.75	9.0900
3010	1002	1309.95	157.1940
3009	1003	1713.23	205.5876
3002	1004	1900.10	228.0120
3001	1007	18.69	2.2428
3006	1007	1098.16	131.7792

10 rows in set (0.269 sec)

28. Find highest rating in each city. Put the output in this form. For the city (city), the highest rating is : (rating).

```
mysql> SELECT CONCAT('For the city (', city, '), the highest rating is : (', MAX(rating), ')') AS output FROM cust GROUP BY city;
```

output
For the city (London), the highest rating is : (300)
For the city (Rome), the highest rating is : (200)
For the city (San Jose), the highest rating is : (300)
For the city (Brelina), the highest rating is : (100)

4 rows in set (0.251 sec)

29. Display the totals of orders for each day and place the results in descending order.

```
mysql> SELECT odate, COUNT(onum) AS total_orders FROM orders GROUP BY odate ORDER BY total_orders DESC;
```

odate	total_orders
1994-10-03	4
1994-10-04	2
1994-10-05	2
1994-10-06	2

4 rows in set (0.270 sec)

30. All combinations of salespeople and customers who shared a city. (ie same city).

```
mysql> SELECT sname, cname FROM salespeople, cust WHERE salespeople.city = cust.city;
+-----+-----+
| sname | cname |
+-----+-----+
| Motika | Hoffman |
| Peel   | Hoffman |
| Serres | Liu     |
| Motika | Clemens |
| Peel   | Clemens |
+-----+-----+
5 rows in set (0.249 sec)
```

31. Name of all customers matched with the salespeople serving them.

```
mysql> SELECT cname, sname FROM cust, salespeople WHERE cust.snum = salespeople.snum;
+-----+-----+
| cname | sname |
+-----+-----+
| Hoffman | Peel |
| Giovanna | Axelrod |
| Liu     | Serres |
| Grass   | Serres |
| Clemens | Rafin  |
| Pereira | Motika |
+-----+-----+
6 rows in set (0.016 sec)
```

32. List each order number followed by the name of the customer who made the order.

```
mysql> SELECT onum, cname FROM orders, cust WHERE orders.cnum = cust.cnum;
+-----+-----+
| onum | cname |
+-----+-----+
| 3002 | Pereira |
| 3003 | Hoffman |
| 3005 | Liu     |
| 3007 | Grass   |
| 3008 | Clemens |
| 3009 | Giovanna |
| 3010 | Grass   |
| 3011 | Clemens |
+-----+-----+
8 rows in set (0.017 sec)
```

33. Names of salesperson and customer for each order after the order number.

```
mysql> SELECT onum, sname, cname FROM orders, cust, salespeople WHERE orders.cnum = cust.cnum AND orders.snum = salespeople.snum;
+-----+-----+-----+
| onum | sname | cname |
+-----+-----+-----+
| 3002 | Motika | Pereira |
| 3003 | Peel   | Hoffman |
| 3005 | Serres | Liu     |
| 3007 | Serres | Grass   |
| 3008 | Peel   | Clemens |
| 3009 | Axelrod | Giovanna |
| 3010 | Serres | Grass   |
| 3011 | Peel   | Clemens |
+-----+-----+-----+
8 rows in set (0.019 sec)
```

34. Produce all customer serviced by salespeople with a commission above 12%.

```
mysql> SELECT cname, sname, comm FROM cust, salespeople WHERE comm > 0.12 AND cust.snum = salespeople.snum;
+-----+-----+
| cname | sname | comm |
+-----+-----+
| Liu   | Serres | 0.13 |
| Grass | Serres | 0.13 |
| Clemens | Rafin | 0.15 |
+-----+-----+
3 rows in set (0.060 sec)
```

35. Calculate the amount of the salesperson's commission on each order with a rating above 100.

```
mysql> SELECT sname, amt * comm AS commission FROM orders, cust, salespeople WHERE rating > 100 AND salespeople.snum = cust.snum AND salespeople.snum = orders.snum AND cust.cnum = orders.cnum;
+-----+-----+
| sname | commission |
+-----+-----+
| Serres | 670.8584753930569 |
| Axelrod | 171.32300255291165 |
+-----+-----+
2 rows in set (0.043 sec)
```

36. Find all pairs of customers having the same rating.

```
mysql> SELECT a.cname, b.cname, a.rating FROM cust a, cust b WHERE a.rating = b.rating AND a.cnum != b.cnum;
+-----+-----+-----+
| cname | cname | rating |
+-----+-----+-----+
| Pereira | Hoffman | 100 |
| Grass | Hoffman | 100 |
| Clemens | Liu | 300 |
| Pereira | Grass | 100 |
| Hoffman | Grass | 100 |
| Liu | Clemens | 300 |
| Grass | Pereira | 100 |
| Hoffman | Pereira | 100 |
+-----+-----+-----+
8 rows in set (0.046 sec)
```

37. Find all pairs of customers having the same rating, each pair coming once only.

```
mysql> SELECT a.cname, b.cname, a.rating FROM cust a, cust b WHERE a.rating = b.rating AND a.cnum < b.cnum;
+-----+-----+-----+
| cname | cname | rating |
+-----+-----+-----+
| Hoffman | Grass | 100 |
| Hoffman | Pereira | 100 |
| Liu | Clemens | 300 |
| Grass | Pereira | 100 |
+-----+-----+-----+
4 rows in set (0.306 sec)
```

38. Policy is to assign three salesperson to each customers. Display all such combination.

```
mysql> SELECT c.cname, s.sname FROM cust c JOIN (SELECT * FROM salespeople LIMIT 3) s;
+-----+-----+
| cname | sname |
+-----+-----+
| Hoffman | Motika |
| Hoffman | Serres |
| Hoffman | Peel |
| Giovanne | Motika |
| Giovanne | Serres |
| Giovanne | Peel |
| Liu | Motika |
| Liu | Serres |
| Liu | Peel |
| Grass | Motika |
| Grass | Serres |
| Grass | Peel |
| Clemens | Motika |
| Clemens | Serres |
| Clemens | Peel |
| Pereira | Motika |
| Pereira | Serres |
| Pereira | Peel |
+-----+-----+
18 rows in set (0.292 sec)
```

39. Display all customers located in cities where salesman serres has customer.

```
mysql> SELECT cname
-> FROM cust
-> WHERE city IN (
->     SELECT c.city
->     FROM cust c
->     JOIN salespeople s ON c.snum = s.snum
->     WHERE s.sname = 'Serres'
-> );
+-----+
| cname |
+-----+
| Liu |
| Grass |
+-----+
2 rows in set (0.025 sec)
```

40. Find all pairs of customers served by single salesperson.

```
mysql> SELECT DISTINCT a.cname FROM cust a, cust b WHERE a.snum = b.snum AND a.cnum != b.cnum;
+-----+
| cname |
+-----+
| Grass |
| Liu |
+-----+
2 rows in set (0.018 sec)
```

41. Produce all pairs of salespeople which are living in the same city. Exclude combinations of salespeople with themselves as well as duplicates with the order reversed.

```
mysql> SELECT a.sname, b.sname FROM salespeople a, salespeople b WHERE a.snum > b.snum AND a.city = b.city;
+-----+-----+
| sname | sname |
+-----+-----+
| Motika | Peel |
+-----+-----+
1 row in set (0.268 sec)
```

42. Produce all pairs of orders by given customer, names that customers and eliminates duplicates.

```
mysql> SELECT c.cname, a.onum, b.onum FROM orders a, orders b, cust c WHERE a.cnum = b.cnum AND a.onum > b.onum AND c.cnum = a.cnum;
```

cname	onum	onum
Grass	3010	3007
Clemens	3011	3008

```
2 rows in set (0.020 sec)
```

43. Produce names and cities of all customers with the same rating as Hoffman.

```
mysql> SELECT cname, city FROM cust WHERE rating = (SELECT rating FROM cust WHERE cname = 'Hoffman') AND cname != 'Hoffman';
```

cname	city
Grass	Brelin
Pereira	Rome

```
2 rows in set (0.017 sec)
```

44. Extract all the orders of Motika

```
mysql> SELECT onum FROM orders WHERE snum = (SELECT snum FROM salespeople WHERE sname = 'Motika');
```

onum
3002

```
1 row in set (0.016 sec)
```

45. All orders credited to the same salesperson who services Hoffman.

```
mysql> SELECT onum, sname, cname, amt FROM orders a, salespeople b, cust c WHERE a.snum = b.snum AND a.cnum = c.cnum AND a.snum = (SELECT snum FROM orders WHERE cnum = (SELECT cnum FROM cust WHERE cname = 'Hoffman'));
```

onum	sname	cname	amt
3003	Peel	Hoffman	767.19
3008	Peel	Clemens	4723.00
3011	Peel	Clemens	9891.88

```
3 rows in set (0.026 sec)
```

46. All orders that are greater than the average for Oct 4.

```
mysql> SELECT * FROM orders WHERE amt > (SELECT AVG(amt) FROM orders WHERE odate = '1994-10-04');
```

ONUM	AMT	ODATE	CNUM	SNUM
3002	1900.10	1994-10-03	2007	1004
3005	5160.45	1994-10-03	2003	1002
3008	4723.00	1994-10-05	2006	1001
3009	1713.23	1994-10-04	2002	1003
3011	9891.88	1994-10-06	2006	1001

```
5 rows in set (0.050 sec)
```

47. Find average commission of salespeople in london.

```
mysql> SELECT AVG(comm) FROM salespeople WHERE city = 'London';
+-----+
| AVG(comm) |
+-----+
| 0.11499999836087227 |
+-----+
1 row in set (0.275 sec)
```

48. Find all orders attributed to salespeople servicing customers in london.

```
mysql> SELECT snum, cnum FROM orders WHERE cnum IN (SELECT cnum FROM cust WHERE city = 'London');
+-----+-----+
| snum | cnum |
+-----+-----+
| 1001 | 2001 |
| 1001 | 2006 |
| 1001 | 2006 |
+-----+-----+
3 rows in set (0.019 sec)
```

49. Extract commissions of all salespeople servicing customers in London.

```
mysql> SELECT comm FROM salespeople WHERE snum IN (SELECT snum FROM cust WHERE city = 'London');
+-----+
| comm |
+-----+
| 0.12 |
| 0.15 |
+-----+
2 rows in set (0.015 sec)
```

50. Find all customers whose cnum is 1000 above the snum of serres.

```
mysql> SELECT cnum, cname FROM cust WHERE cnum > (SELECT snum + 1000 FROM salespeople WHERE sname = 'Serres');
+-----+-----+
| cnum | cname |
+-----+-----+
| 2003 | Liu |
| 2004 | Grass |
| 2006 | Clemens |
| 2007 | Pereira |
+-----+-----+
4 rows in set (0.248 sec)
```

51. Count the customers with rating above San Jose's average.

```
mysql> SELECT cnum, rating FROM cust WHERE rating > (SELECT AVG(rating) FROM cust WHERE city = 'San Jose');
Empty set (0.017 sec)
```