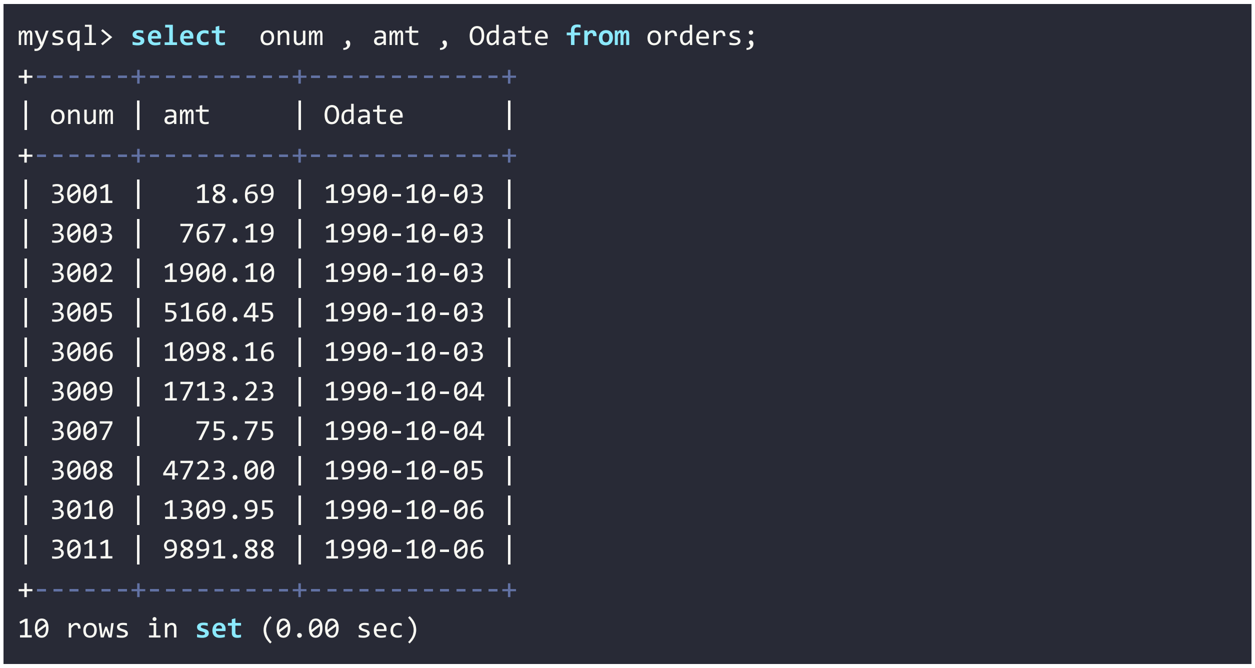
Assignment 4

—---

1) Write a select command that produces the order number, amount, and date for all rows in the Orders table.

—---

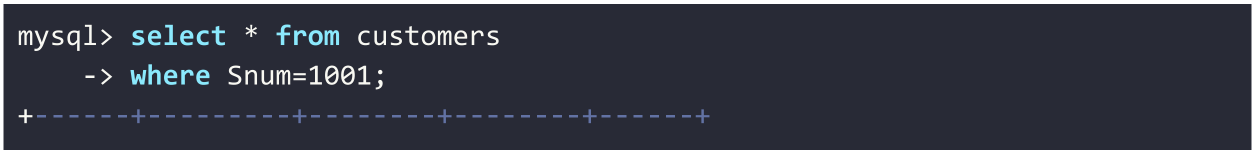


```
mysql> select  onum , amt , Odate from orders;
+------+---------+------------+
| onum | amt     | Odate      |
+------+---------+------------+
| 3001 |   18.69 | 1990-10-03 |
| 3003 |  767.19 | 1990-10-03 |
| 3002 | 1900.10 | 1990-10-03 |
| 3005 | 5160.45 | 1990-10-03 |
| 3006 | 1098.16 | 1990-10-03 |
| 3009 | 1713.23 | 1990-10-04 |
| 3007 |   75.75 | 1990-10-04 |
| 3008 | 4723.00 | 1990-10-05 |
| 3010 | 1309.95 | 1990-10-06 |
| 3011 | 9891.88 | 1990-10-06 |
+------+---------+------------+
10 rows in set (0.00 sec)
```

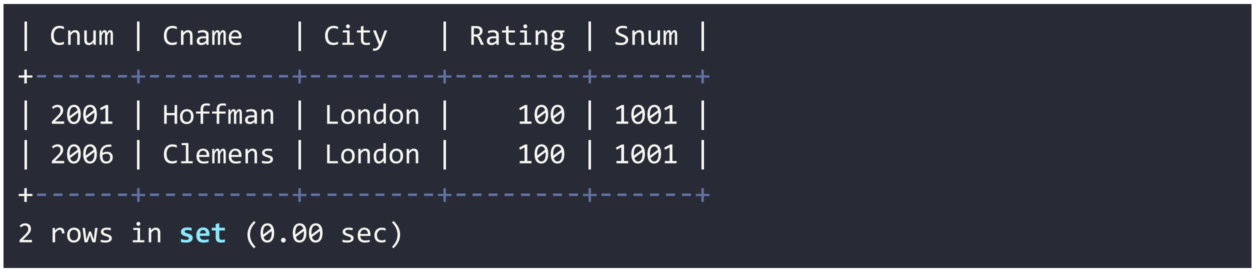
—---

2) Write a query that produces all rows from the Customers table for which the salesperson’s number is 1001.

—---



```
mysql> select * from customers
    -> where Snum=1001;
+------+---------+--------+--------+------+
```



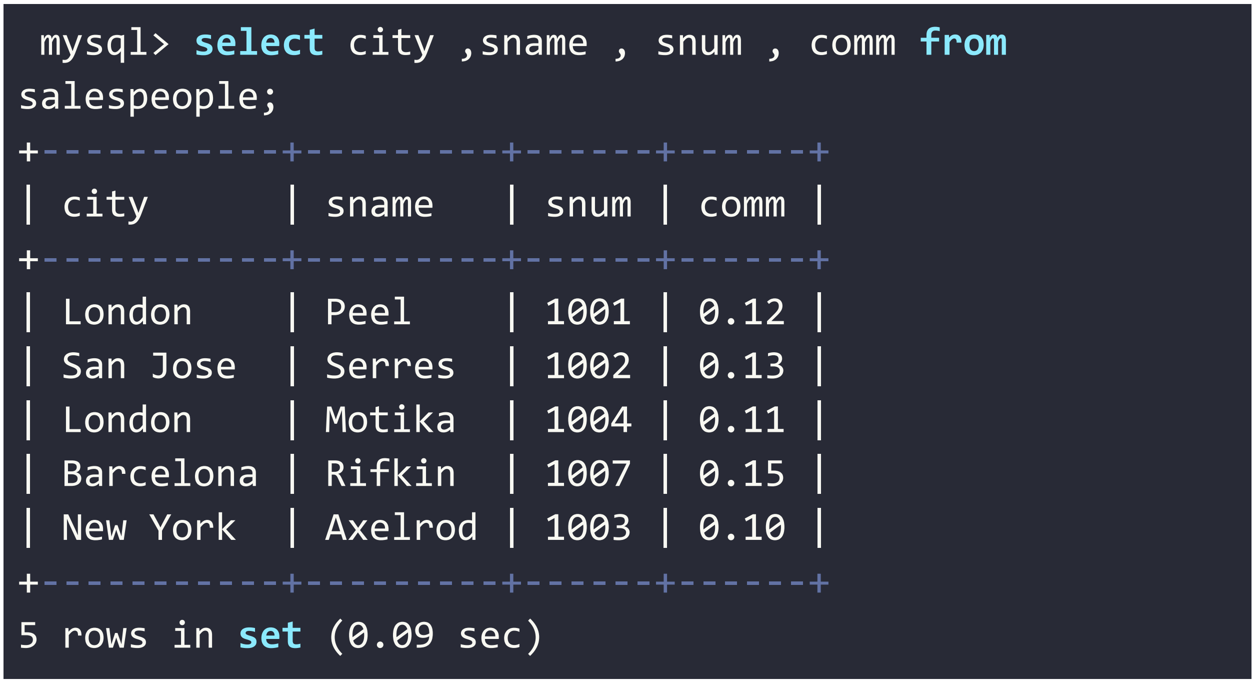
```
| Cnum | Cname   | City   | Rating | Snum |
+------+---------+--------+--------+------+
| 2001 | Hoffman | London |    100 | 1001 |
| 2006 | Clemens | London |    100 | 1001 |
+------+---------+--------+--------+------+
2 rows in set (0.00 sec)
```

—---

—---

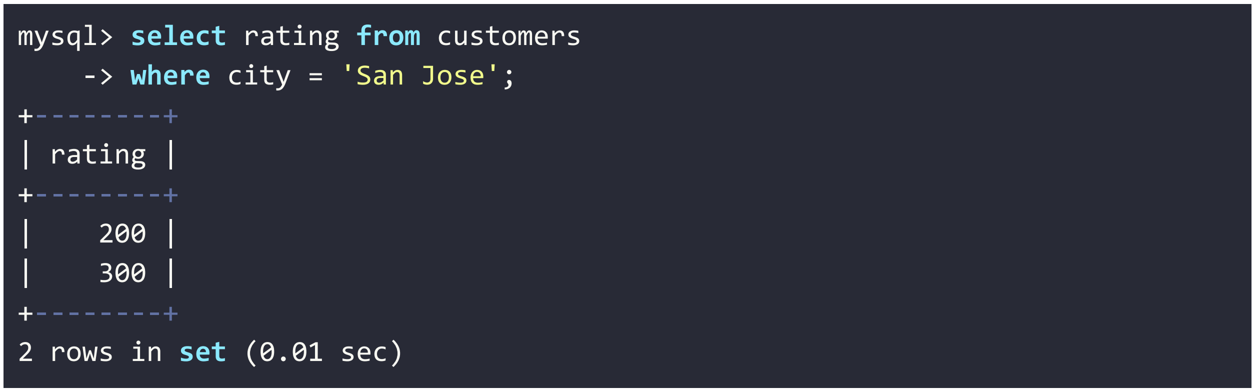
3) Write a query that displays the Salespeople table with the columns in the following order: city, sname, snum, comm

—---



```
 mysql> select city ,sname , snum , comm from
salespeople;
+-----------+---------+------+------+
| city      | sname   | snum | comm |
+-----------+---------+------+------+
| London    | Peel    | 1001 | 0.12 |
| San Jose  | Serres  | 1002 | 0.13 |
| London    | Motika  | 1004 | 0.11 |
| Barcelona | Rifkin  | 1007 | 0.15 |
| New York  | Axelrod | 1003 | 0.10 |
+-----------+---------+------+------+
5 rows in set (0.09 sec)
```

4) Write a select command that produces the rating followed by the name of each customer in San Jose.



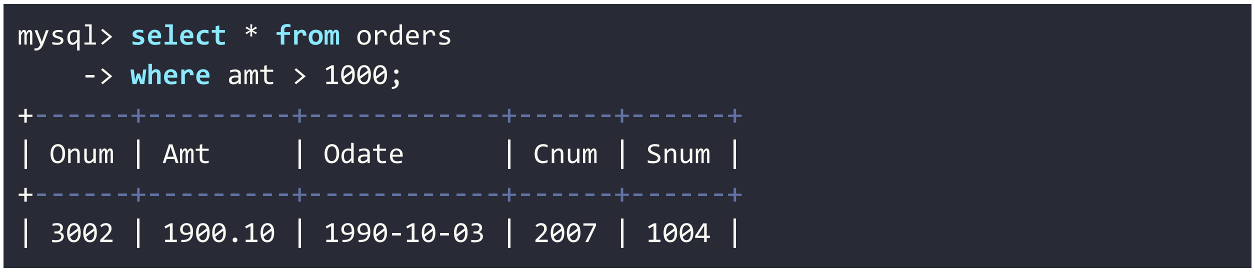
```
mysql> select rating from customers
    -> where city = 'San Jose';
+--------+
| rating |
+--------+
|    200 |
|    300 |
+--------+
2 rows in set (0.01 sec)
```

Assignment 5

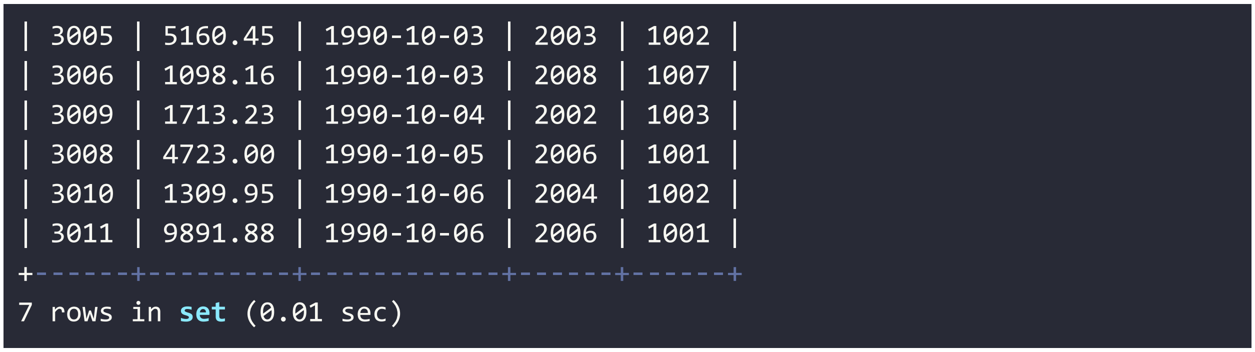
—---

1) Write a query that will give you all orders for more than Rs. 1,000.

—---



```
mysql> select * from orders
    -> where amt > 1000;
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
```

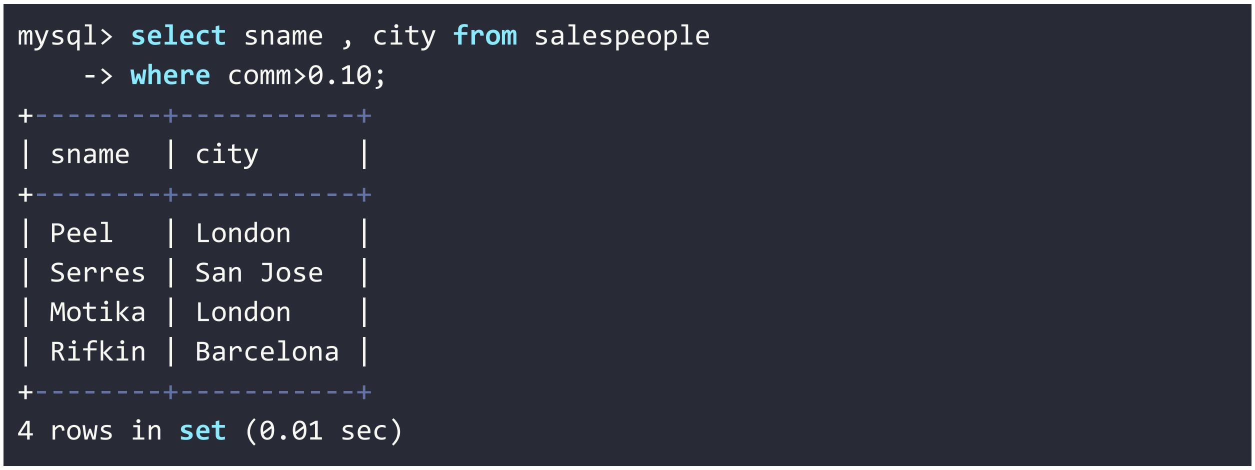


```
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+------+---------+------------+------+------+
7 rows in set (0.01 sec)
```

—---

2) Write a query that will give you the names and cities of all salespeople in London with a commission above .10.

—---

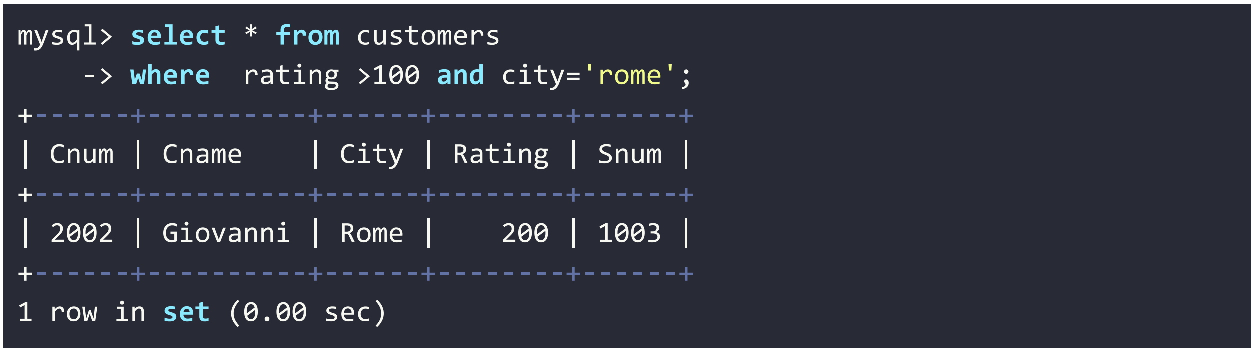


```
mysql> select sname , city from salespeople
    -> where comm>0.10;
+--------+-----------+
| sname  | city      |
+--------+-----------+
| Peel   | London    |
| Serres | San Jose  |
| Motika | London    |
| Rifkin | Barcelona |
+--------+-----------+
4 rows in set (0.01 sec)
```

—---

3) Write a query on the Customers table whose output will exclude all customers with a rating <= 100, unless they are located in Rome.

—---

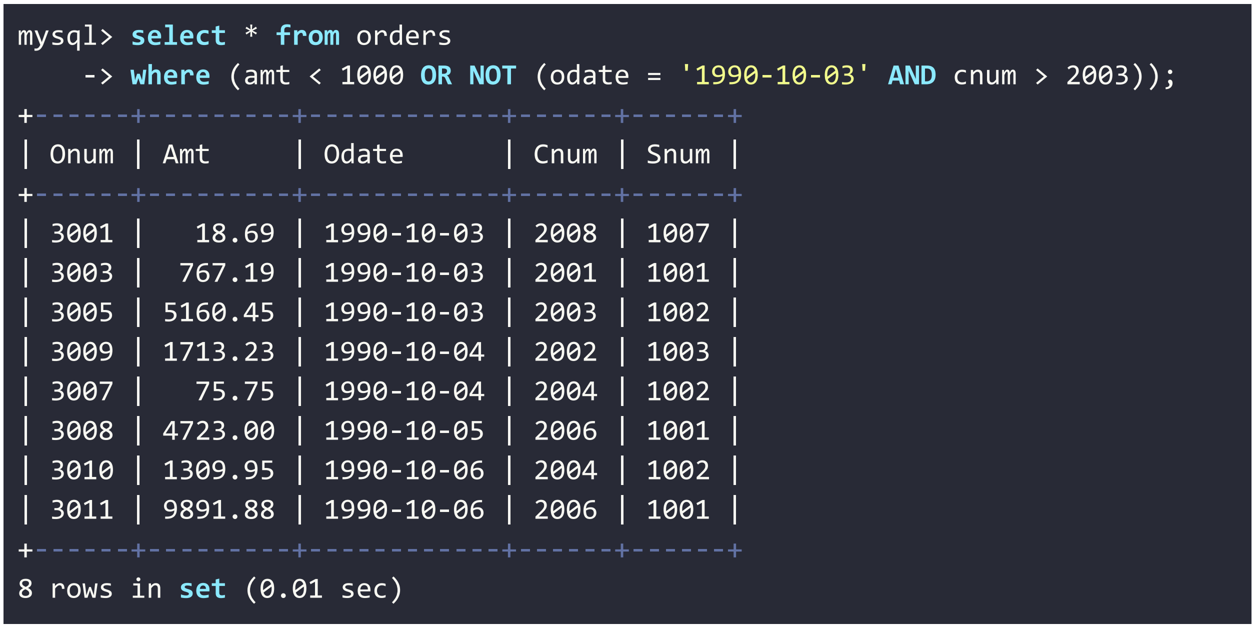


```
mysql> select * from customers
    -> where  rating >100 and city='rome';
+------+-----------+------+--------+------+
| Cnum | Cname     | City | Rating | Snum |
+------+-----------+------+--------+------+
| 2002 | Giovanni  | Rome |    200 | 1003 |
+------+-----------+------+--------+------+
1 row in set (0.00 sec)
```

—---

4) What will be the output from the following query? Select * from Orders where (amt < 1000 OR NOT (odate = ‘1990-10-03’ AND cnum > 2003));

—---

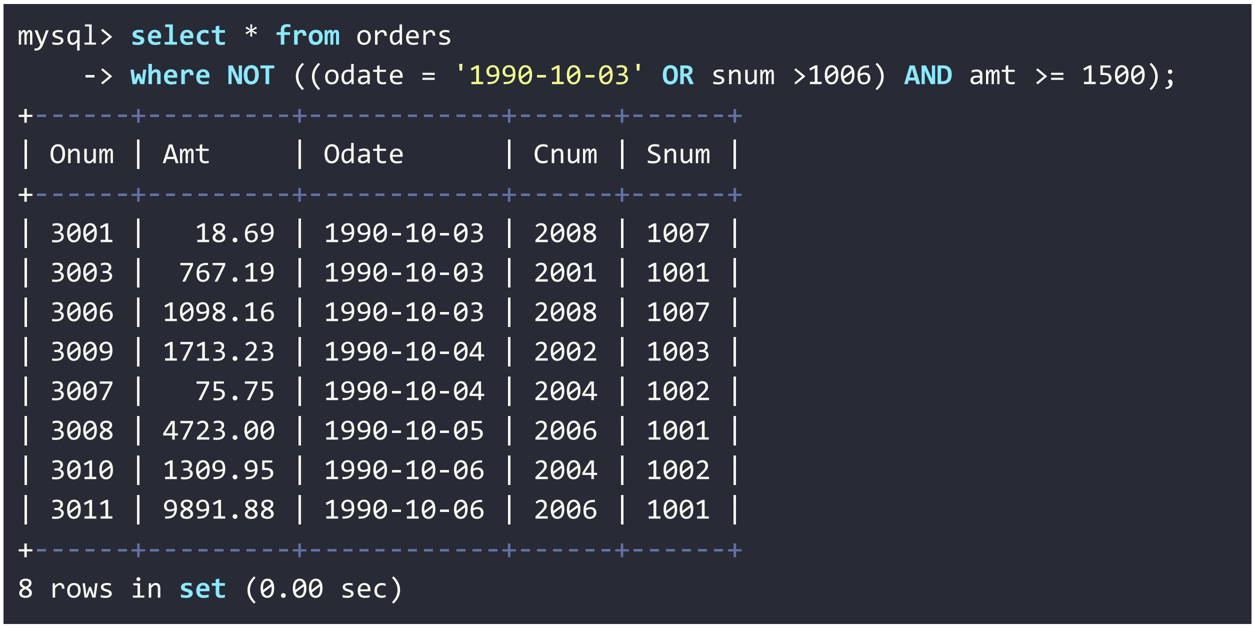


```
mysql> select * from orders
    -> where (amt < 1000 OR NOT (odate = '1990-10-03' AND cnum > 2003));
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+------+---------+------------+------+------+
8 rows in set (0.01 sec)
```

—---

5) What will be the output of the following query? Select * from Orders where NOT ((odate = ‘1990-10-03’ OR snum >1006) AND amt >= 1500);

—---

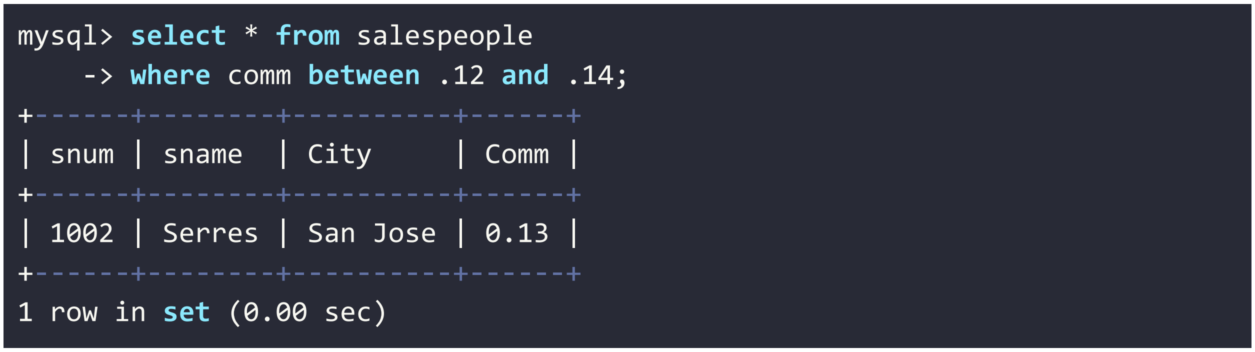


```
mysql> select * from orders
    -> where NOT ((odate = '1990-10-03' OR snum >1006) AND amt >= 1500);
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+------+---------+------------+------+------+
8 rows in set (0.00 sec)
```

—---

6) What is a simpler way to write this query? Select snum, sname, city, comm From Salespeople where (comm > .12 OR comm <.14)

—---



```
mysql> select * from salespeople
    -> where comm between .12 and .14;
+------+--------+----------+------+
| snum | sname  | City     | Comm |
+------+--------+----------+------+
| 1002 | Serres | San Jose | 0.13 |
+------+--------+----------+------+
1 row in set (0.00 sec)
```

—---

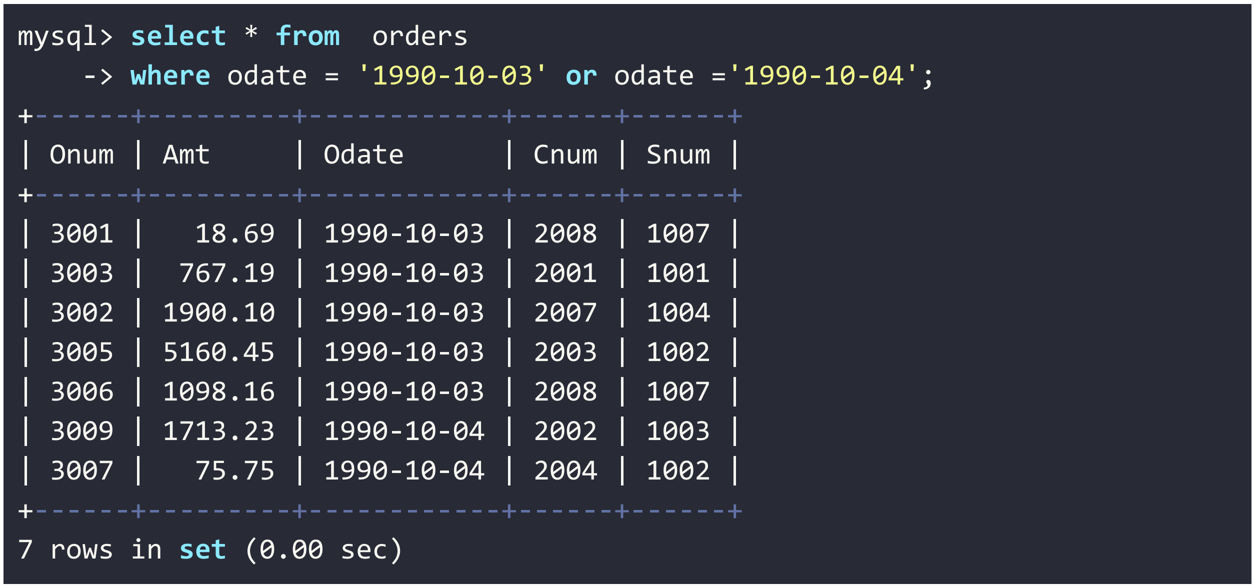
Assignment 6

—--

1) Write two different queries that would produce all orders taken on October 3rd or 4th, 1990

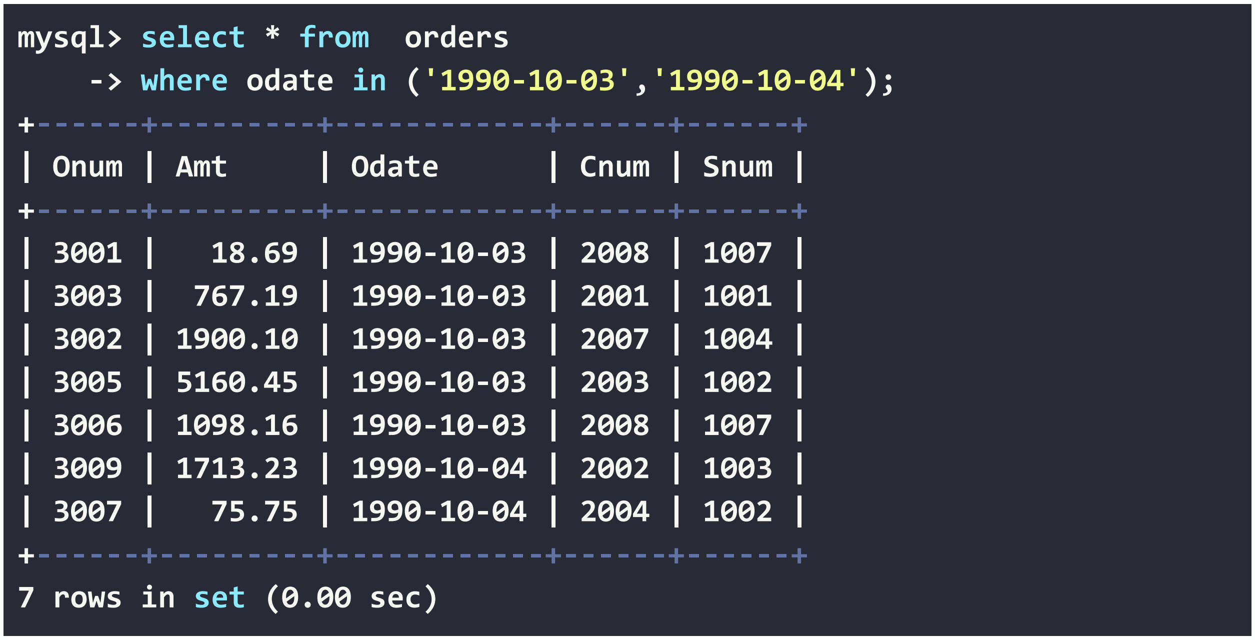
—---

1St Querry



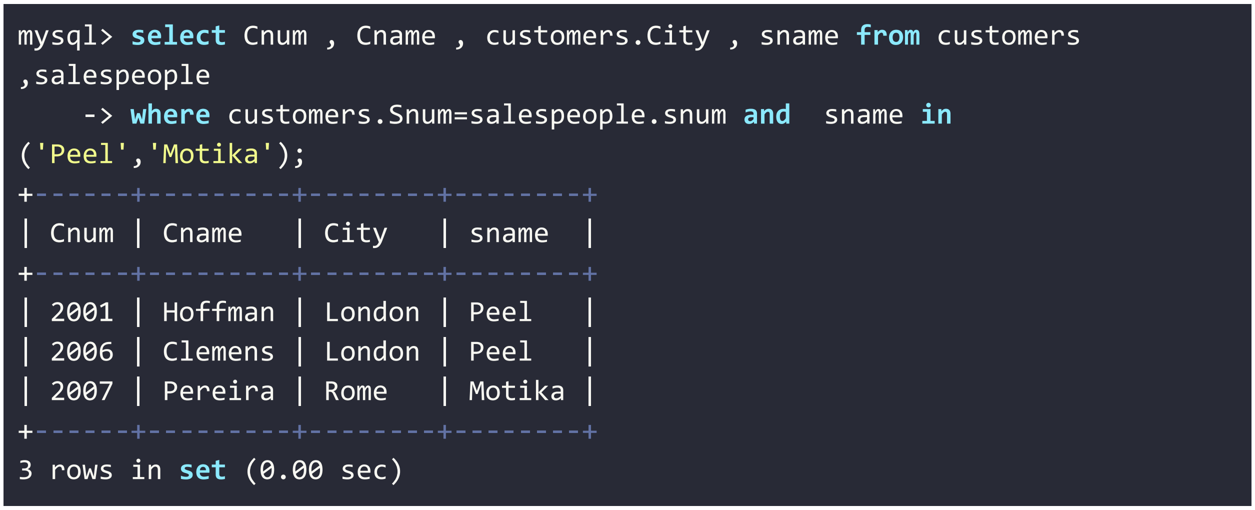
```
mysql> select * from  orders
    -> where odate = '1990-10-03' or odate ='1990-10-04';
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
+------+---------+------------+------+------+
7 rows in set (0.00 sec)
```

2nd Querry



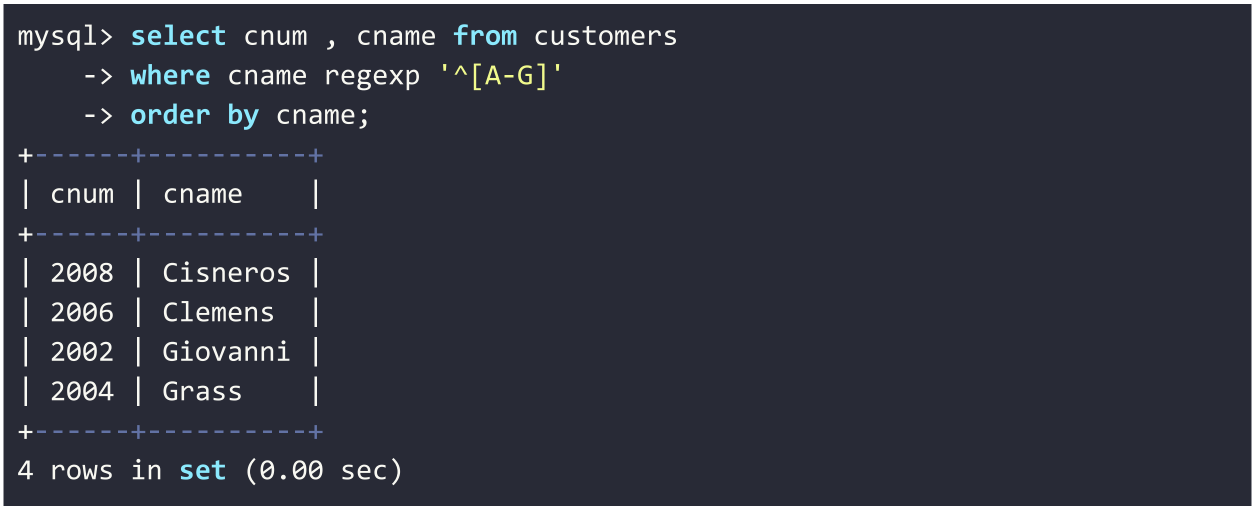
```
mysql> select * from  orders
    -> where odate in ('1990-10-03','1990-10-04');
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
+------+---------+------------+------+------+
7 rows in set (0.00 sec)
```

2) Write a query that selects all of the customers serviced by Peel or Motika. (Hint: the snum field relates the two tables to one another).



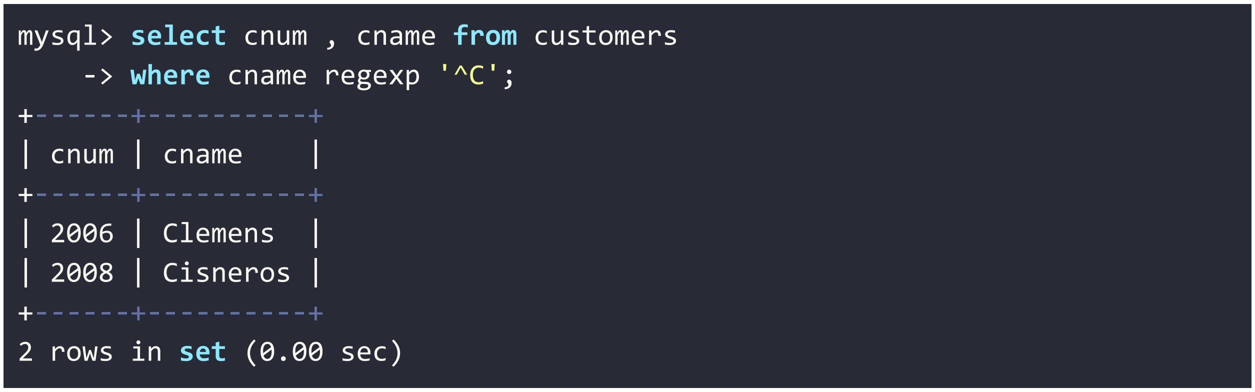
```
mysql> select Cnum , Cname , customers.City , sname from customers
,salespeople
    -> where customers.Snum=salespeople.snum and  sname in
('Peel','Motika');
+------+---------+--------+--------+
| Cnum | Cname   | City   | sname  |
+------+---------+--------+--------+
| 2001 | Hoffman | London | Peel   |
| 2006 | Clemens | London | Peel   |
| 2007 | Pereira | Rome   | Motika |
+------+---------+--------+--------+
3 rows in set (0.00 sec)
```

3) Write a query that will produce all the customers whose names begin with a letter from ‘A’ to ‘G’.



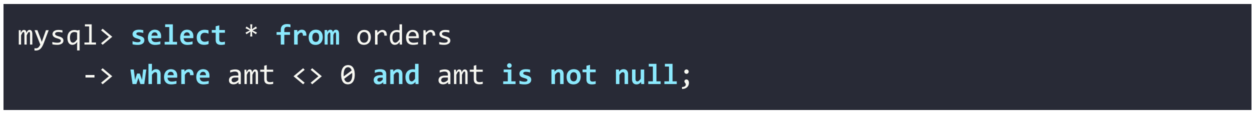
```
mysql> select cnum , cname from customers
    -> where cname regexp '^[A-G]'
    -> order by cname;
+------+----------+
| cnum | cname    |
+------+----------+
| 2008 | Cisneros |
| 2006 | Clemens  |
| 2002 | Giovanni |
| 2004 | Grass    |
+------+----------+
4 rows in set (0.00 sec)
```

4) Write a query that selects all customers whose names begin with the letter ‘C’.

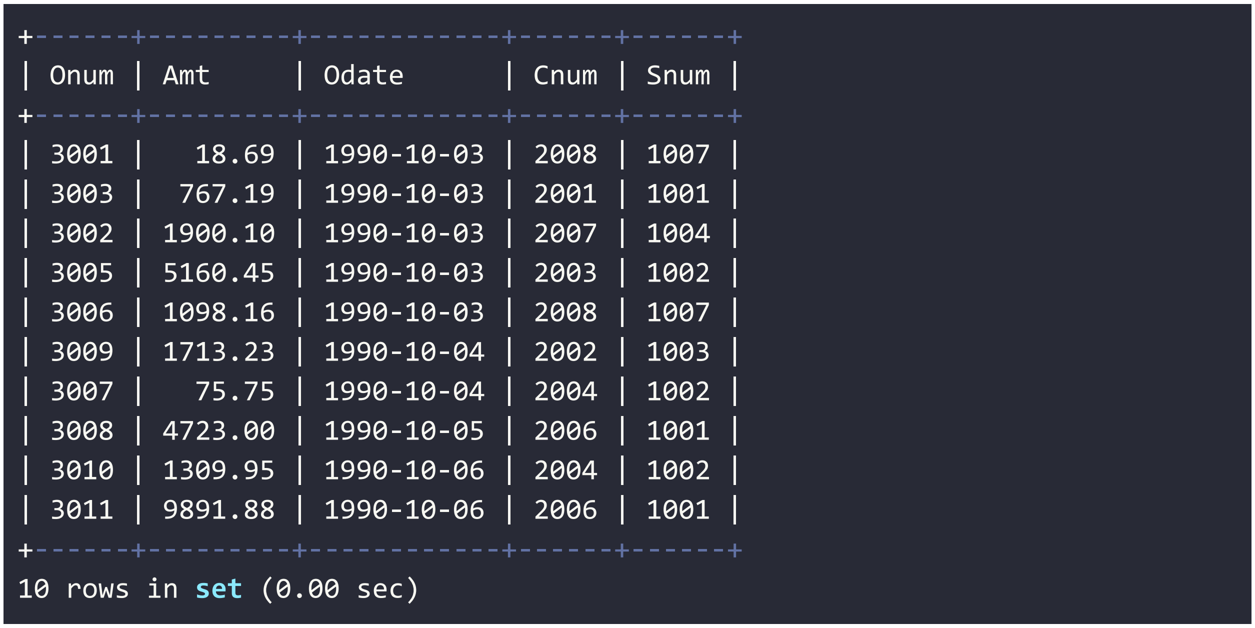


```
mysql> select cnum , cname from customers
    -> where cname regexp '^C';
+------+----------+
| cnum | cname    |
+------+----------+
| 2006 | Clemens  |
| 2008 | Cisneros |
+------+----------+
2 rows in set (0.00 sec)
```

5) Write a query that selects all orders except those with zeroes or NULLs in the amt field.



```
mysql> select * from orders
    -> where amt <> 0 and amt is not null;
```

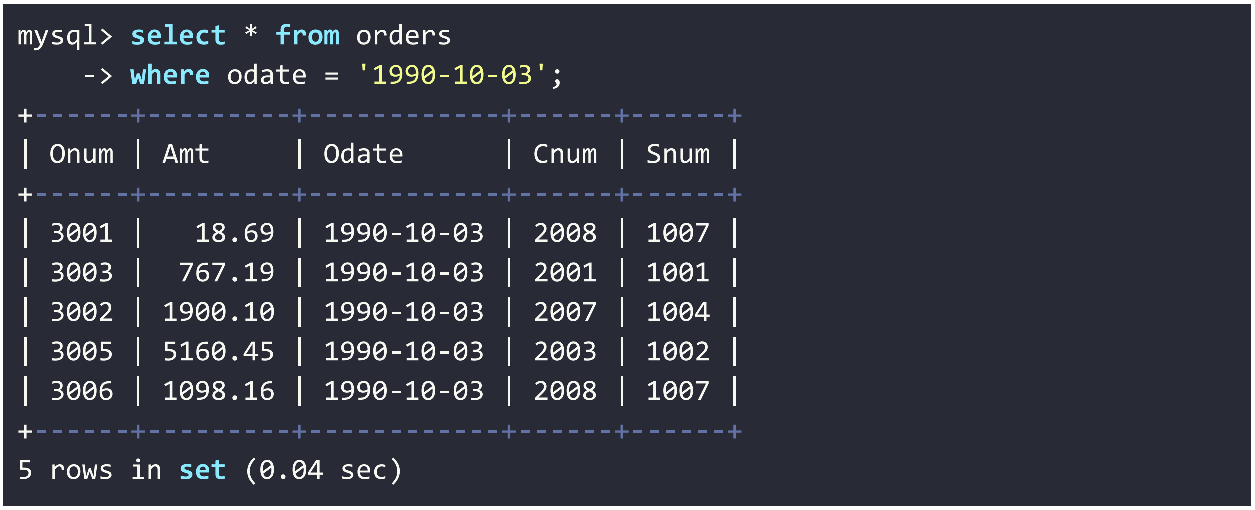


```
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+------+---------+------------+------+------+
10 rows in set (0.00 sec)
```

Assignment 7

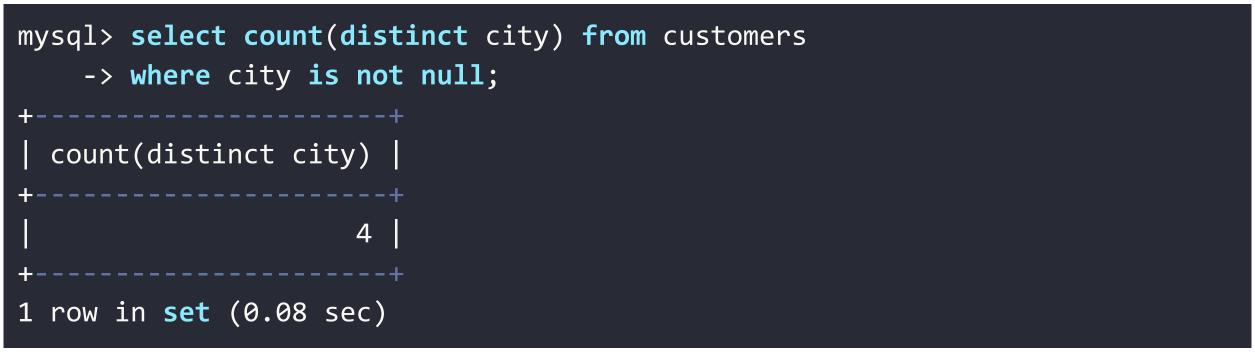
—--

1) Write a query that counts all orders for October 3.



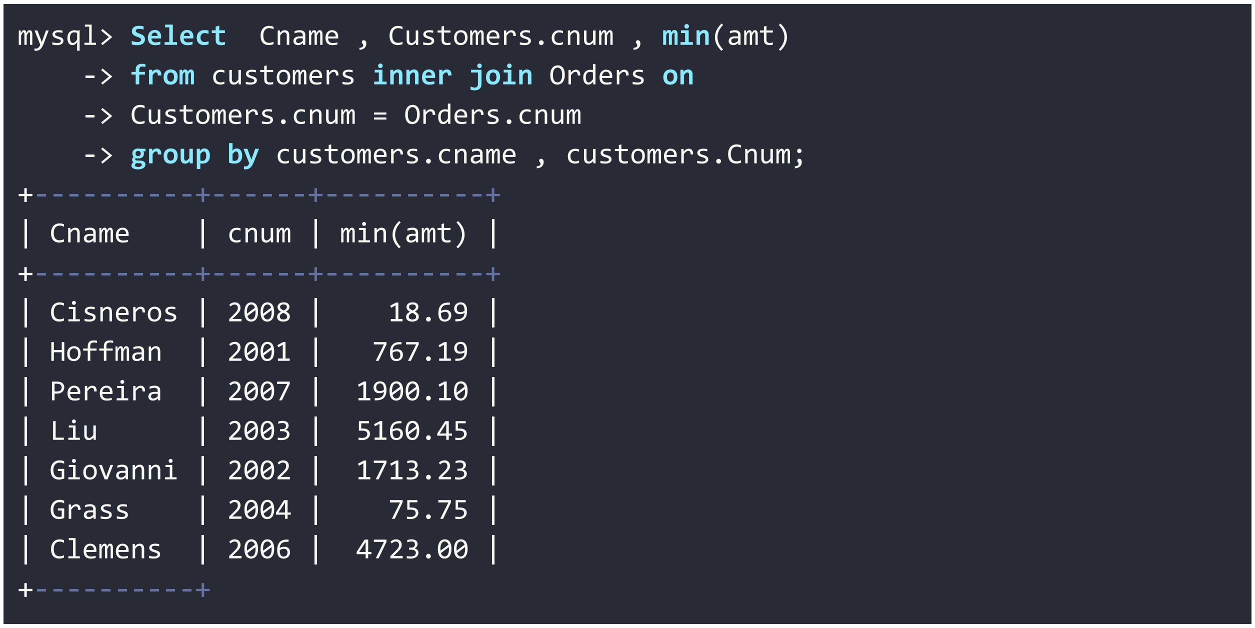
```
mysql> select * from orders
    -> where odate = '1990-10-03';
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
+------+---------+------------+------+------+
5 rows in set (0.04 sec)
```

2) Write a query that counts the number of different non-NULL city values in the Customers table.



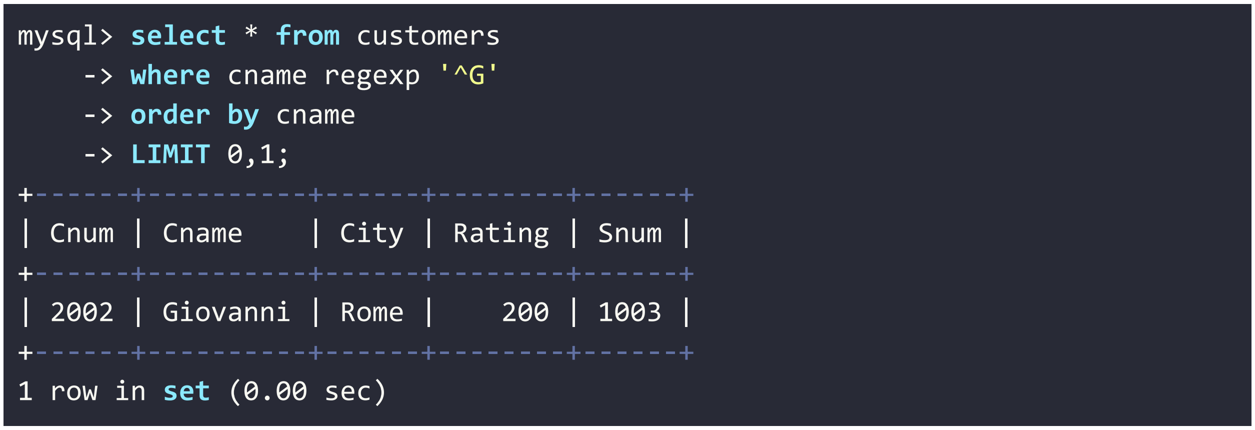
```
mysql> select count(distinct city) from customers
    -> where city is not null;
+----------------------+
| count(distinct city) |
+----------------------+
|                    4 |
+----------------------+
1 row in set (0.08 sec)
```

3) Write a query that selects each customer’s smallest order.



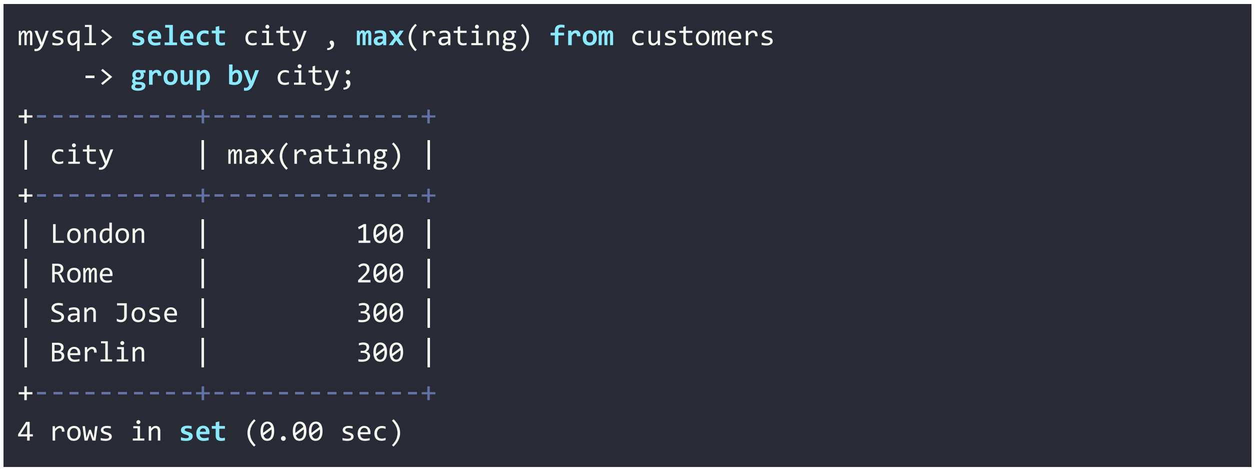
```
mysql> Select  Cname , Customers.cnum , min(amt)
    -> from customers inner join Orders on
    -> Customers.cnum = Orders.cnum
    -> group by customers.cname , customers.Cnum;
+----------+------+----------+
| Cname    | cnum | min(amt) |
+----------+------+----------+
| Cisneros | 2008 |    18.69 |
| Hoffman  | 2001 |   767.19 |
| Pereira  | 2007 |  1900.10 |
| Liu      | 2003 |  5160.45 |
| Giovanni | 2002 |  1713.23 |
| Grass    | 2004 |    75.75 |
| Clemens  | 2006 |  4723.00 |
+----------+
```

4) Write a query that selects the first customer, in alphabetical order, whose name begins with G.



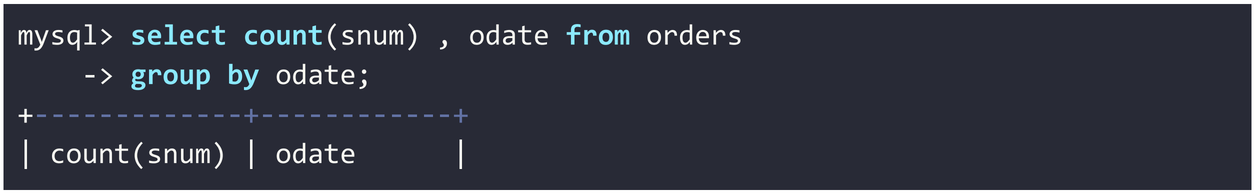
```
mysql> select * from customers
    -> where cname regexp '^G'
    -> order by cname
    -> LIMIT 0,1;
+------+----------+------+--------+------+
| Cnum | Cname    | City | Rating | Snum |
+------+----------+------+--------+------+
| 2002 | Giovanni | Rome |    200 | 1003 |
+------+----------+------+--------+------+
1 row in set (0.00 sec)
```

5) Write a query that selects the highest rating in each city.



```
mysql> select city , max(rating) from customers
    -> group by city;
+----------+-------------+
| city     | max(rating) |
+----------+-------------+
| London   |         100 |
| Rome     |         200 |
| San Jose |         300 |
| Berlin   |         300 |
+----------+-------------+
4 rows in set (0.00 sec)
```

6) Write a query that counts the number of salespeople registering orders for each day. (If a salesperson has more than one order on a given day, he or she should be counted only once.).



```
mysql> select count(snum) , odate from orders
    -> group by odate;
+-------------+------------+
| count(snum) | odate      |
+------------+------------+
|          5 | 1990-10-03 |
|          2 | 1990-10-04 |
|          1 | 1990-10-05 |
|          2 | 1990-10-06 |
+------------+------------+
4 rows in set (0.00 sec)
```

Assignment 8

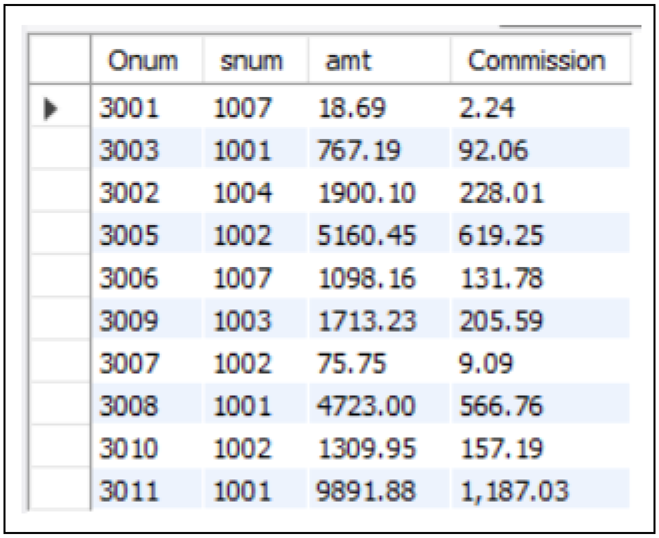
—---

1) Assume each salesperson has a 12% commission. Write a query on the orders table that will produce the order number, the salesperson number, and the amount of the salesperson's commission for that order.

—---

```
select Onum , snum , amt , format((amt*0.12), 2) as Commission from orders
```

Output :



Onum	snum	amt	Commission
3001	1007	18.69	2.24
3003	1001	767.19	92.06
3002	1004	1900.10	228.01
3005	1002	5160.45	619.25
3006	1007	1098.16	131.78
3009	1003	1713.23	205.59
3007	1002	75.75	9.09
3008	1001	4723.00	566.76
3010	1002	1309.95	157.19
3011	1001	9891.88	1,187.03

---

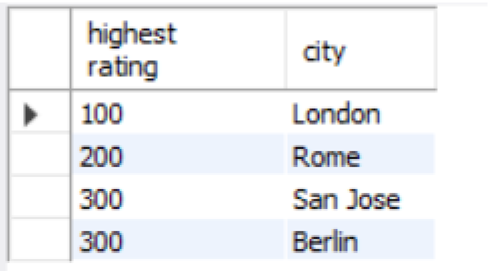
2) Write a query on the Customers table that will find the highest rating in each city.

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

SQL query :

```
select max(rating) as 'highest rating ', city from customers
group by city;
```

Output :



highest rating	city
100	London
200	Rome
300	San Jose
300	Berlin

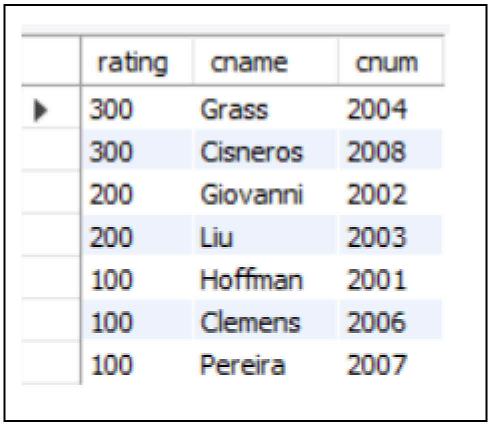
---

3) Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer’s name and number.

Query :

```
select rating, cname , cnum from customers
order by rating desc;
```

Output :



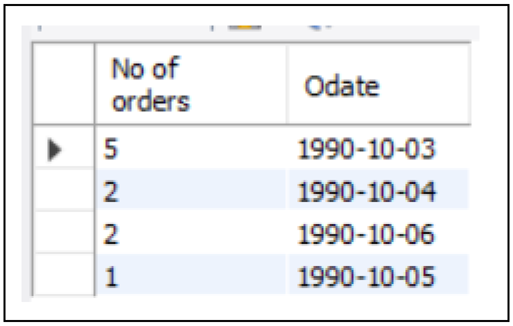
rating	cname	cnum
300	Grass	2004
300	Cisneros	2008
200	Giovanni	2002
200	Liu	2003
100	Hoffman	2001
100	Clemens	2006
100	Pereira	2007

—---

4) Write a query that totals the orders for each day and places the results in descending order.

—---

```
select count(onum) as 'No of orders', Odate
from orders
group by Odate
order by count(onum) desc;
```



No of orders	Odate
5	1990-10-03
2	1990-10-04
2	1990-10-06
1	1990-10-05

—---

Assignment 9

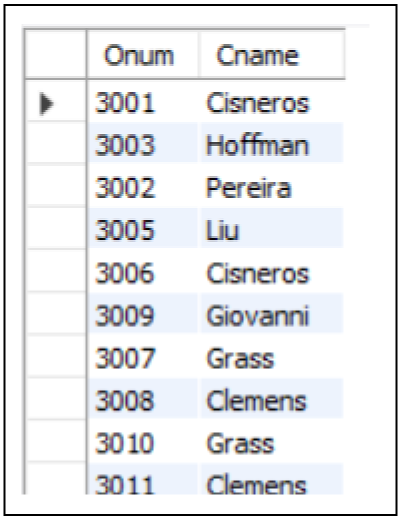
---

1) Write a query that lists each order number followed by the name of the customer who made the order.

SQL query

```
select Onum , Cname from orders, customers
where orders.cnum = customers.cnum;
```

Output:

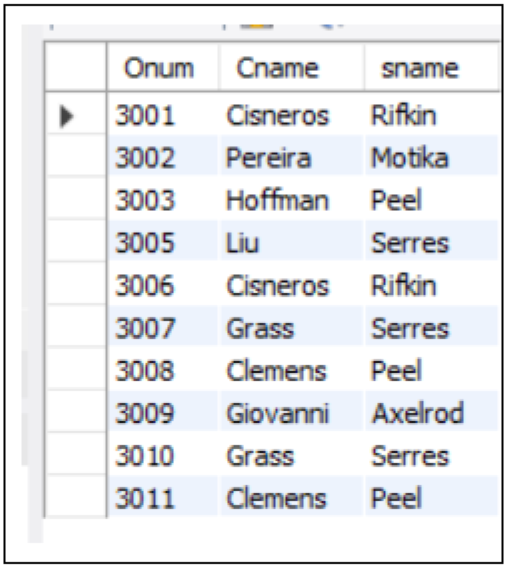


Onum	Cname
3001	Cisneros
3003	Hoffman
3002	Pereira
3005	Liu
3006	Cisneros
3009	Giovanni
3007	Grass
3008	Clemens
3010	Grass
3011	Clemens

2) Write a query that gives the names of both the salesperson and the customer for each order along with the order number.

```
select Onum , Cname , sname from orders, customers, salespeople
where orders.cnum = customers.cnum and salespeople.snum= orders.snum
order by onum;
```

Output :

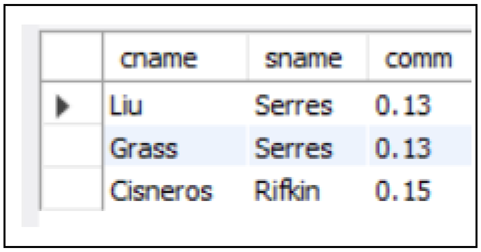


	Onum	Cname	sname
▶	3001	Cisneros	Rifkin
	3002	Pereira	Motika
	3003	Hoffman	Peel
	3005	Liu	Serres
	3006	Cisneros	Rifkin
	3007	Grass	Serres
	3008	Clemens	Peel
	3009	Giovanni	Axelrod
	3010	Grass	Serres
	3011	Clemens	Peel

3) Write a query that produces all customers serviced by salespeople with a commission above 12%. Output the customer's name, the salesperson's name, and the salesperson's rate of commission.

```
select cname , sname , comm from customers, salespeople
where customers.snum = salespeople.snum
and comm>.12;
```

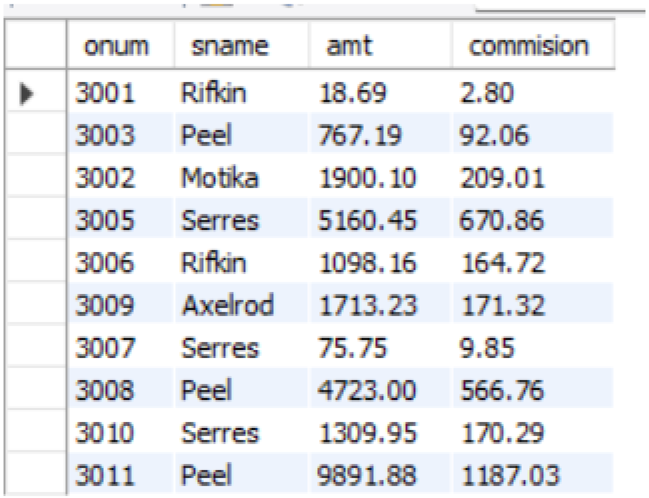
Output :



cname	sname	comm
Liu	Serres	0.13
Grass	Serres	0.13
Cisneros	Rifkin	0.15

4) Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

```
select onum, sname, amt  , amt*Comm as commision from orders, salespeople ,
customers
where salespeople.snum = orders.snum and orders.cnum=customers.cnum and
customers.rating>100;
```



onum	sname	amt	commision
3001	Rifkin	18.69	2.80
3003	Peel	767.19	92.06
3002	Motika	1900.10	209.01
3005	Serres	5160.45	670.86
3006	Rifkin	1098.16	164.72
3009	Axelrod	1713.23	171.32
3007	Serres	75.75	9.85
3008	Peel	4723.00	566.76
3010	Serres	1309.95	170.29
3011	Peel	9891.88	1187.03

—---

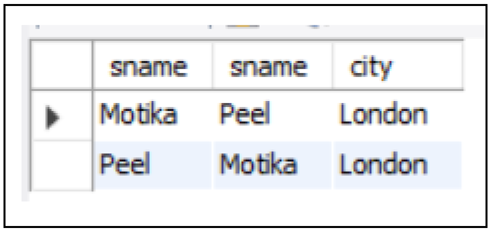
Assignment 10

—---

1) Write a query that produces all pairs of salespeople who are living in the same city. Exclude combinations of salespeople with themselves as well as duplicate rows with the order reversed.

—---

```
select  s1.sname , s2.sname , s1.city
from salespeople as s1 , salespeople as s2
where s1.snum <> s2.snum and s1.city = s2.city;
```



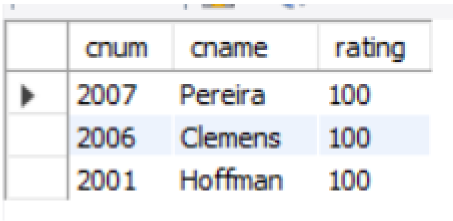
sname	sname	city
Motika	Peel	London
Peel	Motika	London

—---

2) Write a query that produces the names and cities of all customers with the same rating as Hoffman.

—---

```
select  distinct c1.cnum, c1.cname, c1.rating  from customers as c1 ,
customers as c2
where c1.cnum <> c2.cnum and c1.rating=(select rating from customers where
cname= 'Hoffman');
```



cnum	cname	rating
2007	Pereira	100
2006	Clemens	100
2001	Hoffman	100

—---

3) Write a query that selects the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.

—---

—---

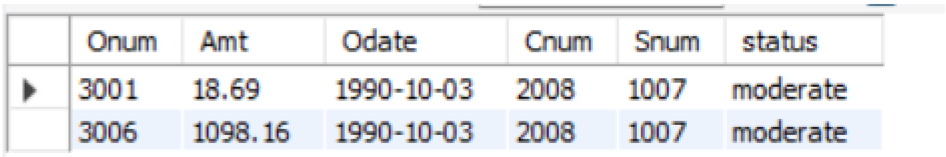
Assignment 11

—---

1) Write a query that uses a subquery to obtain all orders for the customer named Cisneros. Assume you do not know his customer number (cnum).

—---

```
select * from orders
where cnum = (select cnum from customers where cname = 'Cisneros' );
```



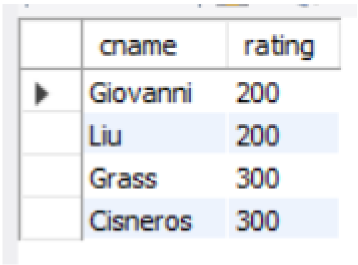
	Onum	Amt	Odate	Cnum	Snum	status
▶	3001	18.69	1990-10-03	2008	1007	moderate
	3006	1098.16	1990-10-03	2008	1007	moderate

—--

2) Write a query that produces the names and ratings of all customers who have above-average orders.

—--

```
select cname , rating from customers
where rating > (select avg(rating) from customers);
```



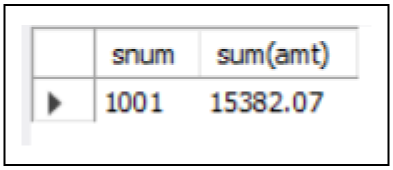
cname	rating
Giovanni	200
Liu	200
Grass	300
Cisneros	300

—--

3) Write a query that selects the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.

—--

```
select snum , sum(amt) from orders
group by snum
having sum(amt) > (select max(amt) from orders);
```

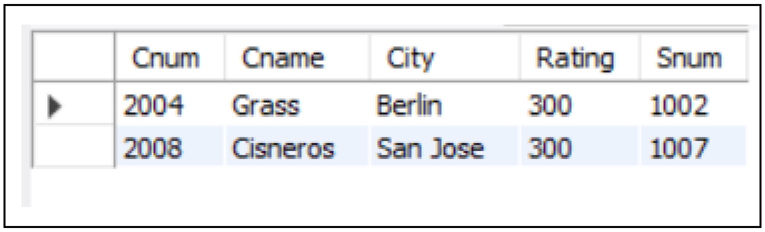


snum	sum(amt)
1001	15382.07

Assignment 12

1) Write a query that selects all customers whose ratings are equal to or greater than ANY of Serres'.

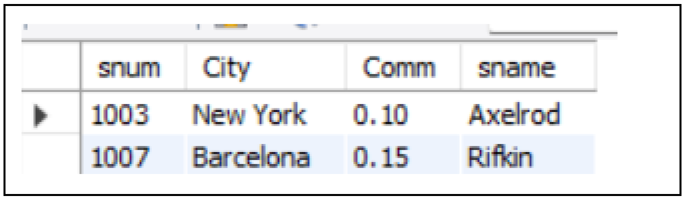
```sql
select * from customers
where rating >= ANY(select rating from customers where cname = 'cisneros' )
order by cnum;
```



	Cnum	Cname	City	Rating	Snum
▶	2004	Grass	Berlin	300	1002
	2008	Cisneros	San Jose	300	1007

2) Write a query using ANY or ALL that will find all salespeople who have no customers located in their city.

```sql
select * from salespeople
where city <> all (select city from customers);
```



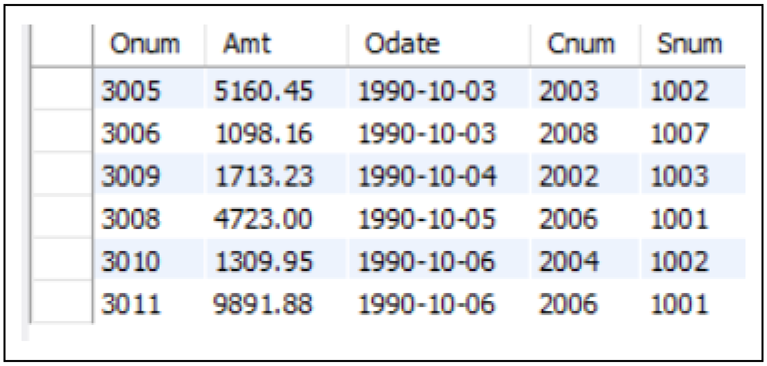
snum	City	Comm	sname
1003	New York	0.10	Axelrod
1007	Barcelona	0.15	Rifkin

—---

3) Write a query that selects all orders for amounts greater than any for the customers in London.

—---

```
select * from orders;
select * from customers;
select * from orders
where amt >  any (select amt from orders where cnum  = any (select cnum
from customers where city='London' ));
```



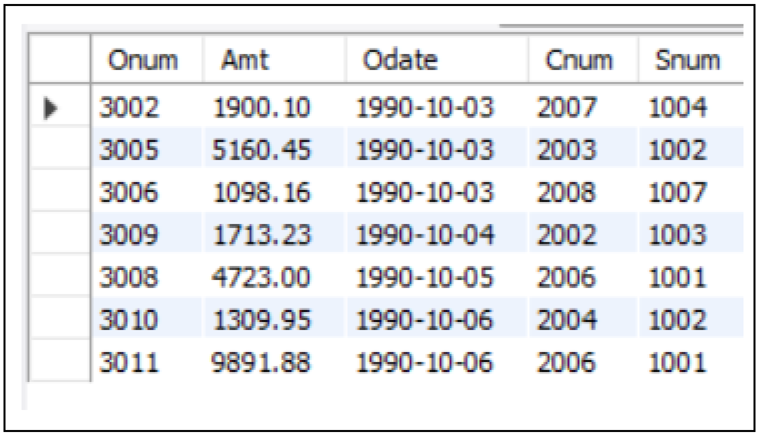
Onum	Amt	Odate	Cnum	Snum
3005	5160.45	1990-10-03	2003	1002
3006	1098.16	1990-10-03	2008	1007
3009	1713.23	1990-10-04	2002	1003
3008	4723.00	1990-10-05	2006	1001
3010	1309.95	1990-10-06	2004	1002
3011	9891.88	1990-10-06	2006	1001

—---

4) Write a query that selects all orders for amounts greater than any for the customers in London.

—---

```
select * from orders
where amt > (
select  min(amt) from orders , customers
where orders.cnum=customers.cnum and city ='London');
```



Onum	Amt	Odate	Cnum	Snum
3002	1900.10	1990-10-03	2007	1004
3005	5160.45	1990-10-03	2003	1002
3006	1098.16	1990-10-03	2008	1007
3009	1713.23	1990-10-04	2002	1003
3008	4723.00	1990-10-05	2006	1001
3010	1309.95	1990-10-06	2004	1002
3011	9891.88	1990-10-06	2006	1001

—---

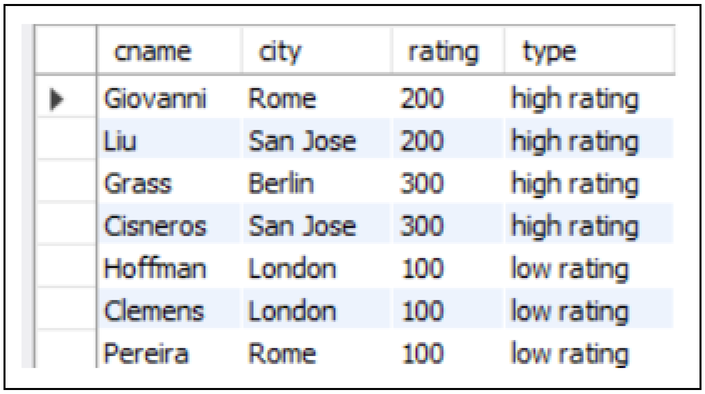
Assignment 13

—---

1) Create a union of two queries that shows the names, cities, and ratings of all customers. Those with rating of 200 or greater will also have the words “High Rating”, while the others will have the words “Low Rating”.

—---

```
select * from customers;
Select cname , city, rating ,  'high rating' as type  from customers
where rating >= 200
UNION
select cname ,city , rating, 'low rating' from customers
where rating <200;
```



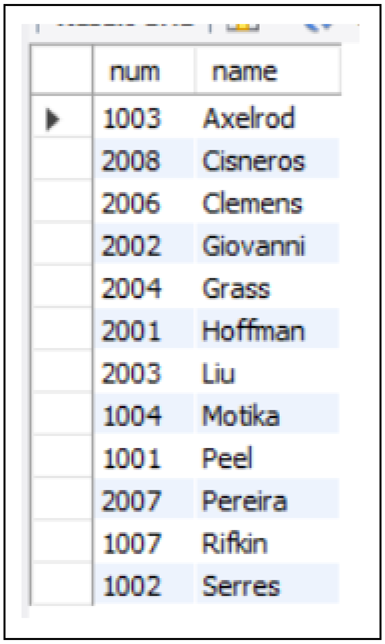
cname	city	rating	type
Giovanni	Rome	200	high rating
Liu	San Jose	200	high rating
Grass	Berlin	300	high rating
Cisneros	San Jose	300	high rating
Hoffman	London	100	low rating
Clemens	London	100	low rating
Pereira	Rome	100	low rating

—---

2) Write a command that produces the name and number of each salesperson and each customer with more than one current order. Put the results in alphabetical order.

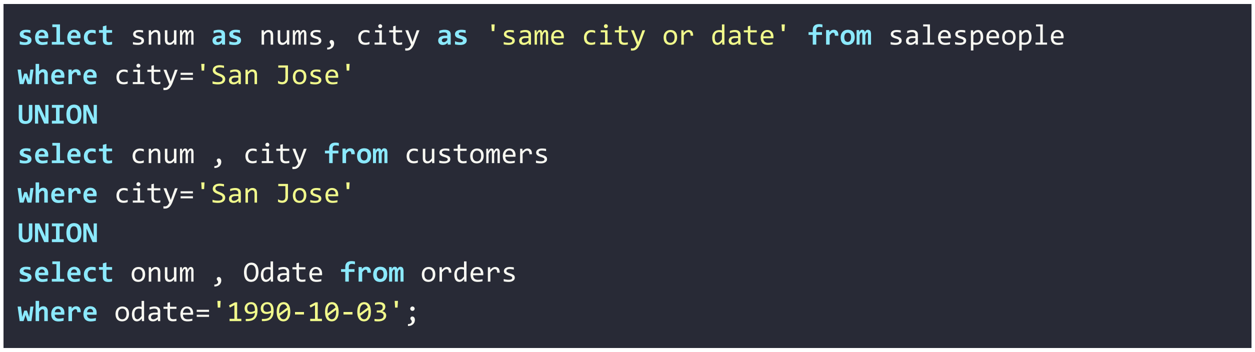
—---

```
select * from orders;
select   orders.cnum as num,cname as name from customers , orders
where orders.cnum=customers.cnum
UNION
select orders.snum , sname from salespeople , orders
where orders.snum= salespeople.snum
order by name;
```

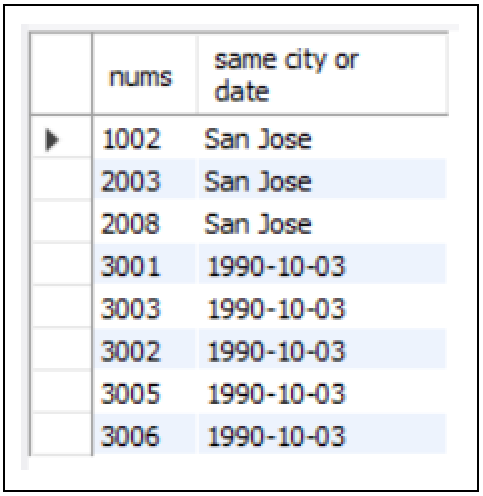


num	name
1003	Axelrod
2008	Cisneros
2006	Clemens
2002	Giovanni
2004	Grass
2001	Hoffman
2003	Liu
1004	Motika
1001	Peel
2007	Pereira
1007	Rifkin
1002	Serres

3) Form a union of three queries. Have the first select the snums of all salespeople in San Jose; the second, the cnums of all customers in San Jose; and the third the onums of all orders on October 3. Retain duplicates between the last two queries but 64 Sameer Dehadrai eliminate any redundancies between either of them and the first. (Note: in the sample tables as given, there would be no such redundancy. This is beside the point.)



```
select snum as nums, city as 'same city or date' from salespeople
where city='San Jose'
UNION
select cnum , city from customers
where city='San Jose'
UNION
select onum , Odate from orders
where odate='1990-10-03';
```

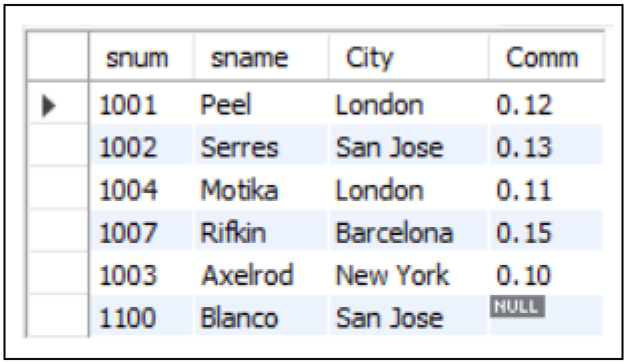


nums	same city or date
1002	San Jose
2003	San Jose
2008	San Jose
3001	1990-10-03
3003	1990-10-03
3002	1990-10-03
3005	1990-10-03
3006	1990-10-03

Assignment 14

1) Write a command that puts the following values, in their given order, into the salespeople table: city – San Jose, name – Blanco, comm – NULL, cnum – 1100.

```
insert into salespeople(city , sname , comm, snum) values
('San Jose' , 'Blanco' , null, 1100);
select * from salespeople;
```



snum	sname	City	Comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelrod	New York	0.10
1100	Blanco	San Jose	NULL

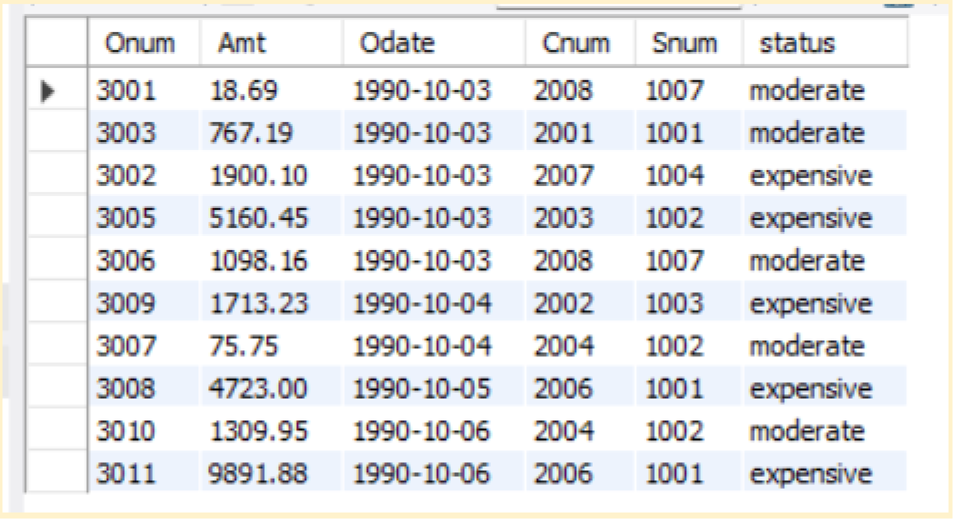
—---

2) Write a command that removes all orders from customer Clemens from the Orders table.

—---

Before deleting

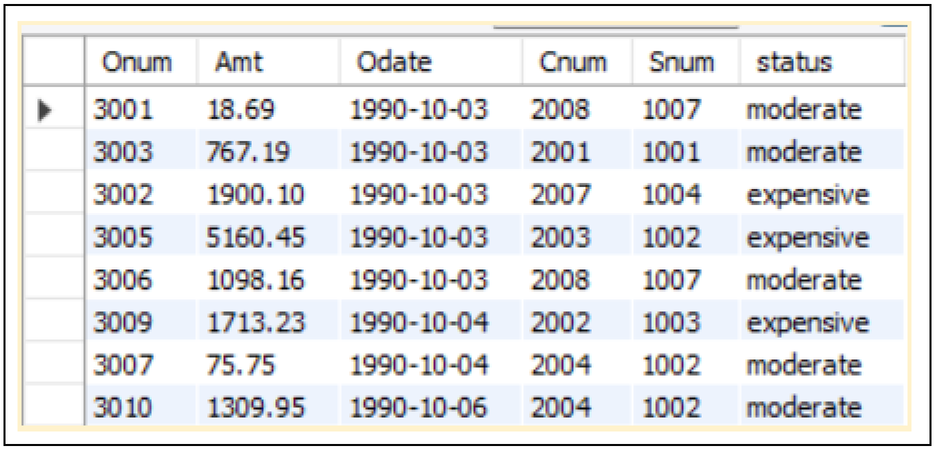
```
select * from orders;
```



	Onum	Amt	Odate	Cnum	Snum	status
▶	3001	18.69	1990-10-03	2008	1007	moderate
	3003	767.19	1990-10-03	2001	1001	moderate
	3002	1900.10	1990-10-03	2007	1004	expensive
	3005	5160.45	1990-10-03	2003	1002	expensive
	3006	1098.16	1990-10-03	2008	1007	moderate
	3009	1713.23	1990-10-04	2002	1003	expensive
	3007	75.75	1990-10-04	2004	1002	moderate
	3008	4723.00	1990-10-05	2006	1001	expensive
	3010	1309.95	1990-10-06	2004	1002	moderate
	3011	9891.88	1990-10-06	2006	1001	expensive

After deleting :

```
delete from  orders
where cnum = (select cnum from customers where cname ='Clemens');
select * from orders;
```



	Onum	Amt	Odate	Cnum	Snum	status
▶	3001	18.69	1990-10-03	2008	1007	moderate
	3003	767.19	1990-10-03	2001	1001	moderate
	3002	1900.10	1990-10-03	2007	1004	expensive
	3005	5160.45	1990-10-03	2003	1002	expensive
	3006	1098.16	1990-10-03	2008	1007	moderate
	3009	1713.23	1990-10-04	2002	1003	expensive
	3007	75.75	1990-10-04	2004	1002	moderate
	3010	1309.95	1990-10-06	2004	1002	moderate

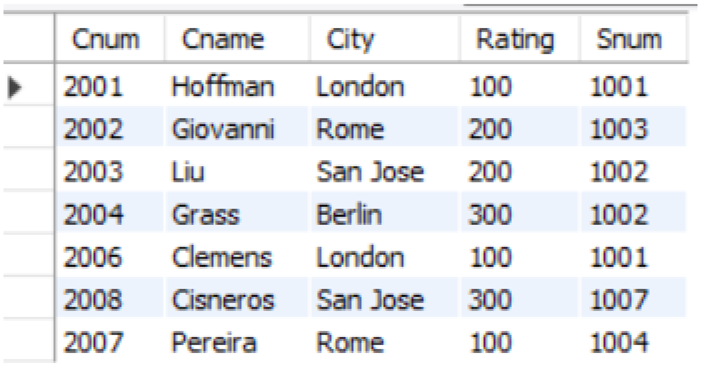
—---

3) Write a command that increases the rating of all customers in Rome by 100.

—---

Before querying

```
select * from customers;
```

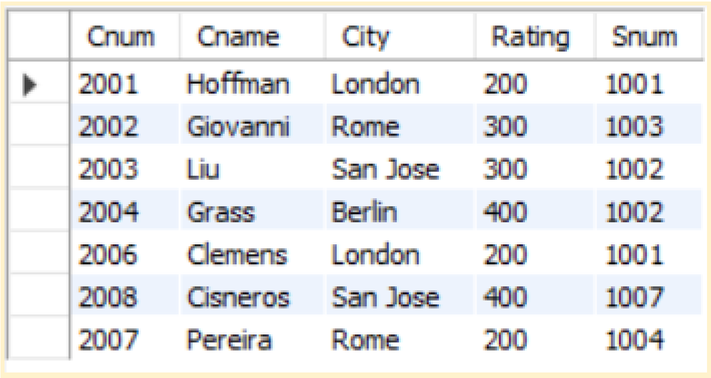


	Cnum	Cname	City	Rating	Snum
▶	2001	Hoffman	London	100	1001
	2002	Giovanni	Rome	200	1003
	2003	Liu	San Jose	200	1002
	2004	Grass	Berlin	300	1002
	2006	Clemens	London	100	1001
	2008	Cisneros	San Jose	300	1007
	2007	Pereira	Rome	100	1004

After querying :

```
update customers
set rating =rating +100;

select * from customers;
```

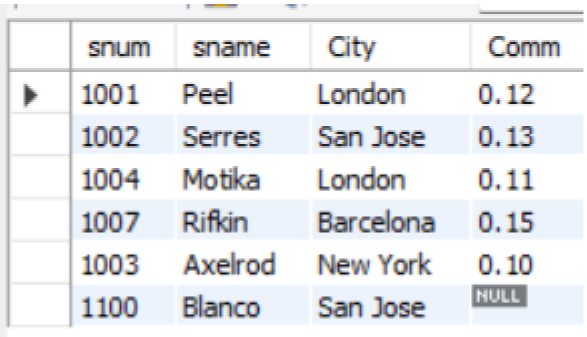


	Cnum	Cname	City	Rating	Snum
▶	2001	Hoffman	London	200	1001
	2002	Giovanni	Rome	300	1003
	2003	Liu	San Jose	300	1002
	2004	Grass	Berlin	400	1002
	2006	Clemens	London	200	1001
	2008	Cisneros	San Jose	400	1007
	2007	Pereira	Rome	200	1004

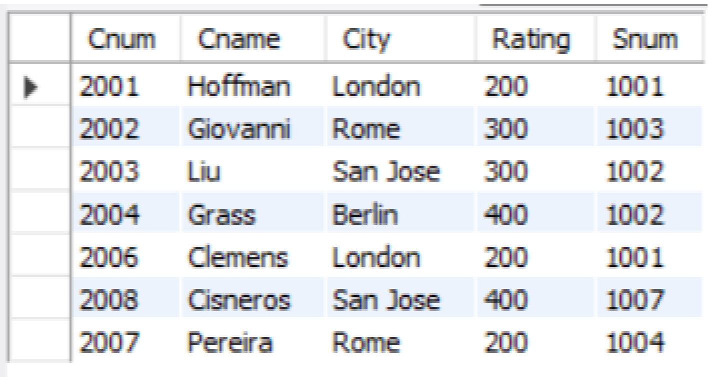
3) Write a command that increases the rating of all customers in Rome by 100.

Before querying

```
select * from salespeople;
select * from customers;
```

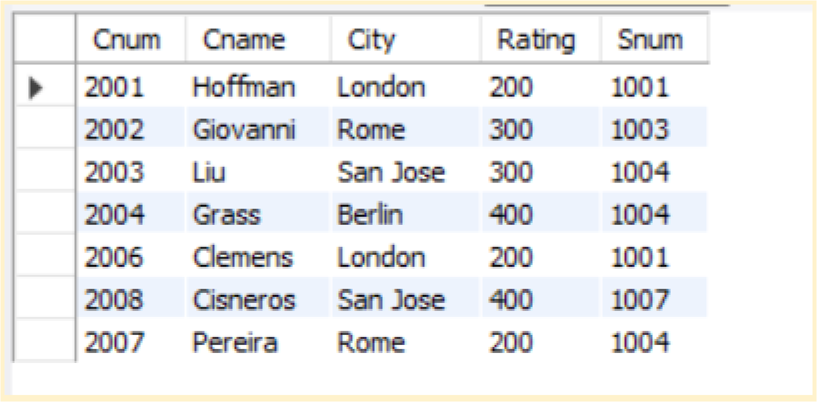


	snum	sname	City	Comm
▶	1001	Peel	London	0.12
	1002	Serres	San Jose	0.13
	1004	Motika	London	0.11
	1007	Rifkin	Barcelona	0.15
	1003	Axelrod	New York	0.10
	1100	Blanco	San Jose	NULL



	Cnum	Cname	City	Rating	Snum
▶	2001	Hoffman	London	200	1001
	2002	Giovanni	Rome	300	1003
	2003	Liu	San Jose	300	1002
	2004	Grass	Berlin	400	1002
	2006	Clemens	London	200	1001
	2008	Cisneros	San Jose	400	1007
	2007	Pereira	Rome	200	1004

After querying :



Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	200	1001
2002	Giovanni	Rome	300	1003
2003	Liu	San Jose	300	1004
2004	Grass	Berlin	400	1004
2006	Clemens	London	200	1001
2008	Cisneros	San Jose	400	1007
2007	Pereira	Rome	200	1004

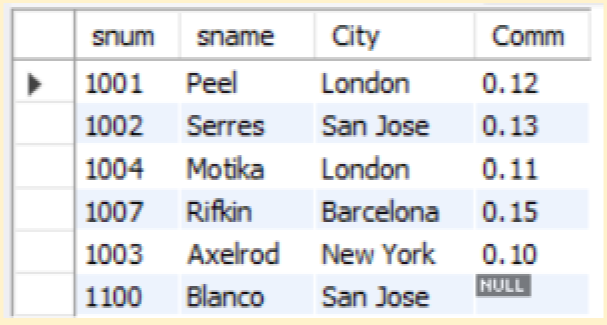
Assignment 15

---

1) Assume there is a table called Multicust, with all of the same column definitions as Salespeople. Write a command that inserts all salespeople with more than one customer into this table.

Creating table called Muticast with same structure as salespeople :

```
create table multicast select * from salespeople;
select * from multicast;
```

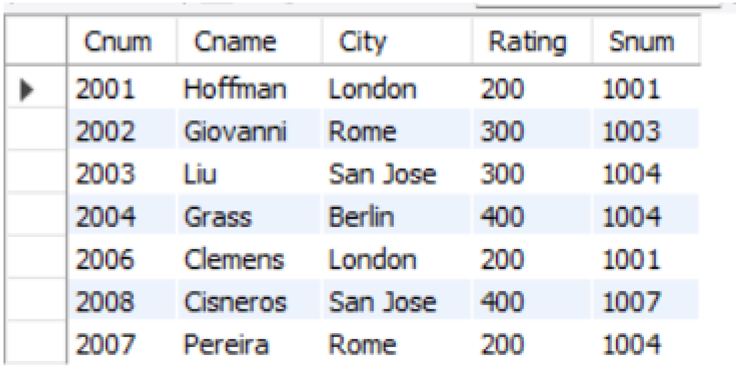


	snum	sname	City	Comm
▶	1001	Peel	London	0.12
	1002	Serres	San Jose	0.13
	1004	Motika	London	0.11
	1007	Rifkin	Barcelona	0.15
	1003	Axelrod	New York	0.10
	1100	Blanco	San Jose	NULL

2) Write a command that deletes all customers with no current orders.

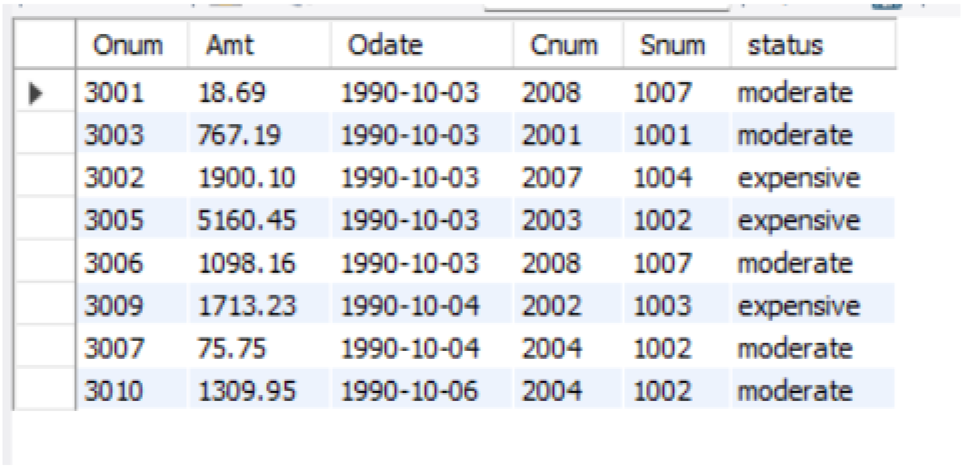
Before querying:

Customers table :



	Cnum	Cname	City	Rating	Snum
▶	2001	Hoffman	London	200	1001
	2002	Giovanni	Rome	300	1003
	2003	Liu	San Jose	300	1004
	2004	Grass	Berlin	400	1004
	2006	Clemens	London	200	1001
	2008	Cisneros	San Jose	400	1007
	2007	Pereira	Rome	200	1004

Orders table :

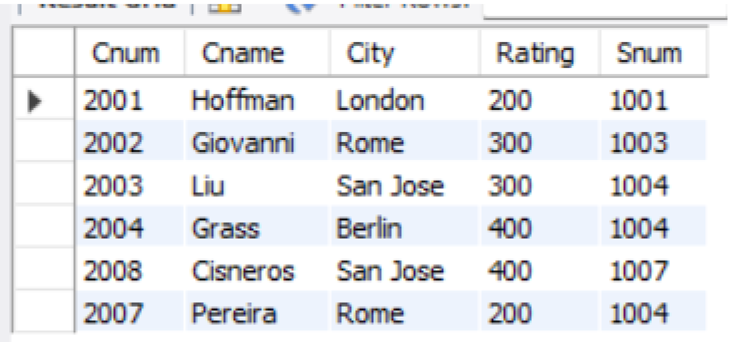


	Onum	Amt	Odate	Cnum	Snum	status
▶	3001	18.69	1990-10-03	2008	1007	moderate
	3003	767.19	1990-10-03	2001	1001	moderate
	3002	1900.10	1990-10-03	2007	1004	expensive
	3005	5160.45	1990-10-03	2003	1002	expensive
	3006	1098.16	1990-10-03	2008	1007	moderate
	3009	1713.23	1990-10-04	2002	1003	expensive
	3007	75.75	1990-10-04	2004	1002	moderate
	3010	1309.95	1990-10-06	2004	1002	moderate

After querying :

```
delete from customers
where NOT (cnum = any (select cnum from orders));
```

```
select * from customers;
```



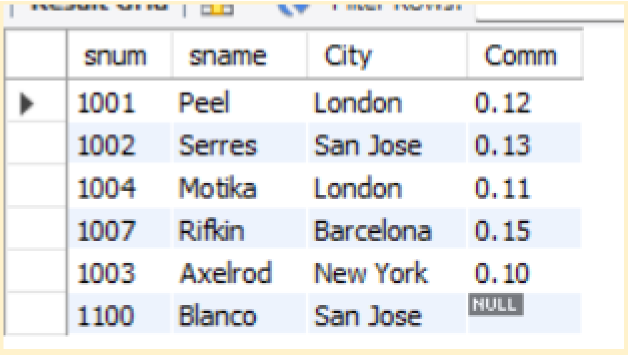
Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	200	1001
2002	Giovanni	Rome	300	1003
2003	Liu	San Jose	300	1004
2004	Grass	Berlin	400	1004
2008	Cisneros	San Jose	400	1007
2007	Pereira	Rome	200	1004

---

3) Write a command that increases by twenty percent the commissions of all salespeople with total orders above Rs. 3,000.

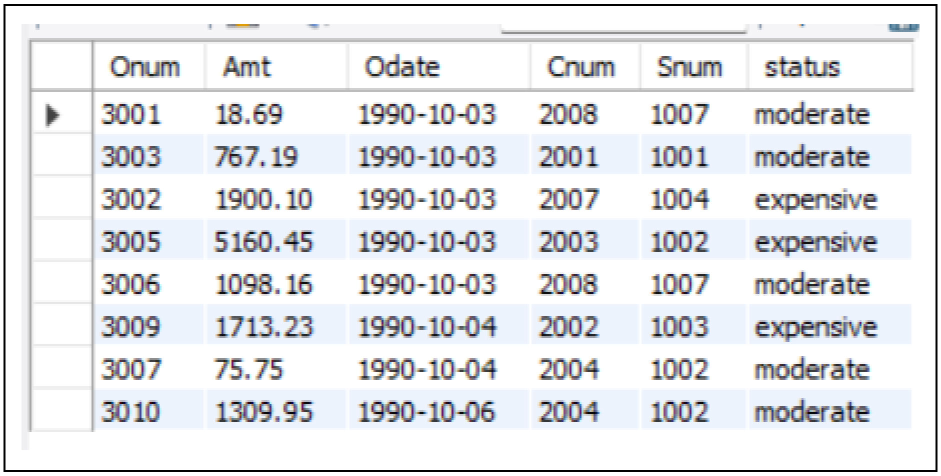
Before querying :

Salespeople Table :



snum	sname	City	Comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelrod	New York	0.10
1100	Blanco	San Jose	NULL

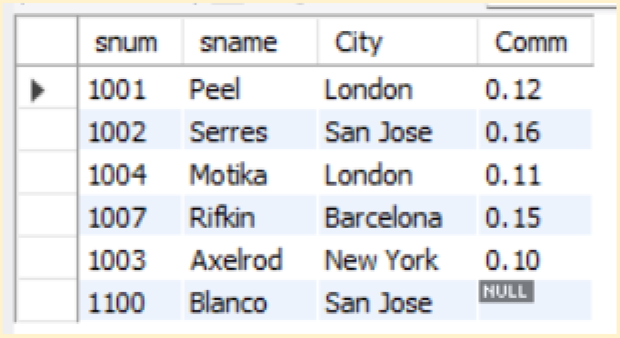
Orders table :



	Onum	Amt	Odate	Cnum	Snum	status
▶	3001	18.69	1990-10-03	2008	1007	moderate
	3003	767.19	1990-10-03	2001	1001	moderate
	3002	1900.10	1990-10-03	2007	1004	expensive
	3005	5160.45	1990-10-03	2003	1002	expensive
	3006	1098.16	1990-10-03	2008	1007	moderate
	3009	1713.23	1990-10-04	2002	1003	expensive
	3007	75.75	1990-10-04	2004	1002	moderate
	3010	1309.95	1990-10-06	2004	1002	moderate

After querying :

```
update salespeople
set comm= (comm + comm*0.2)
where snum = any (select snum from (
select snum, sum(amt) from orders
group by snum
having sum(amt) > 3000) as min_table);
select * from salespeople;
```



	snum	sname	City	Comm
▶	1001	Peel	London	0.12
	1002	Serres	San Jose	0.16
	1004	Motika	London	0.11
	1007	Rifkin	Barcelona	0.15
	1003	Axelrod	New York	0.10
	1100	Blanco	San Jose	NULL