

- 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".

```
KD3_86716_Abhay@>select cname,city,rating,'high_rating' from customers where rating>=200 union select cname,city,rating,'low rating' from customers where rating < 200 ;
```

cname	city	rating	high_rating
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

7 rows in set (0.00 sec)

- 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.

```
KD3_86716_Abhay@>SELECT sname,  
-> snum,  
-> cname  
-> FROM salespeople  
-> JOIN customers using(snum)  
-> WHERE snum IN  
-> (SELECT snum  
-> FROM customers  
-> GROUP BY snum  
-> HAVING count(snum)>1)  
-> ORDER BY sname,  
-> cname;
```

sname	snum	cname
Peel	1001	Clemens
Peel	1001	Hoffman
Serres	1002	Grass
Serres	1002	Liu

4 rows in set (0.00 sec)

KD3\_86716\_Abhay@>

- 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 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 besides the point).

```
KD3_86716_Abhay@>select snum from salespeople where city='san jose' union select cnum from customers where city='san jose' union all select onum from orders where odate='1990-10-03';
```

snum
1002
2003
2008
3001
3003
3002
3005
3006

```
8 rows in set (0.00 sec)
```