1. From sql\_store, choose columns name 'first\_name, points and then add 1000 to points in a new column named 'discount factor'
2. From customer table, suppose we have 2 persons with similar state, we only want to see unique states
3. From table products select columns name, unit\_price, then multiply column unit\_price by 1.1 and name it as new\_price
4. Where clause, filtering data: From customer table choose only those with points > 3000. Now choose those not in state ‘va’.
5. Where clause using more conditions using (and, or, not). Do following:

-- choose bothbirth after 1990-01-01 and pointes more than 1000

-- where birthdate after '1990-01-01' or points bigger than 1000 and state='va'

-- Which operation does first: and, or, ?

1. NOT or negate. From customer choose those which are not with birth after 1990-01-01 or point bigger than 1000.
2. In operator. From customer choose those who lives in states va,fl,ga
3. In operator. From products choose those which are equal to 49,38,72
4. Between, choose from table customer whose who have points between 1000 and 3000.
5. Like. From customer choose

--any person whom lastname starts with ‘b’

--any person whom lastname starts with ‘brush’

--any person whom lastname has ‘b’ somewhere, does not matter where

--any person whom lastname has ‘y’ at the end

--any person whom lastname has ‘y’ at the end and 5 character before y.

--any person whom lastname has ‘y’ at the end and ‘b’ at begginig and total word has 6 character.

1. NOT Like. From customer choose those who have phone number without ‘9’.
2. Regexp (regular expression): From customer table and last name columns:

Find last name ends with field.

Find last name starts with fields.

Find last name has fields or mac.

Find last name which has ‘e’ and before ‘e’ characters ‘g’ or ‘i’ or ‘m’

Find last name which has ‘e’ and before ‘e’ any characters between ‘a-h’

-- ^beginning

-- $ end

-- | or

-- [bdcs]

-- [h-m]

1. IS NULL. From orders, which order has no shipment data?
2. Order by, from customer table sort data by state column descending

-- sort by state descending and first\_name ascending

1. Choose table order\_item, and make a new column named total\_price which is quantity\* unit\_price. And then sort them by total\_price descending
2. LIMIT. From customer table, show the top 3 people with higher points
3. Inner join. Join two tables ‘orders’ and ‘customer’ with column ‘customer\_id’ as a new table and show the columns order\_id, customer\_id, first\_name from this new table.
4. Joining across database. Select order\_items table in sql\_store and join it with product table in sql\_inventory dataset. Join them in column name product\_id.
5. Do the same as Ex.18 but start doing from inventory data set. You should get similar results.
6. Join table payment and payment\_method from database ‘invocing’ such that the payment\_method is known.
7. Left join: In sql\_store select 2 tables, products and order\_items. Join 2 tables where you find out for ‘ALL’ product\_id the quantity of order. Present your final table with column product\_id, name and quantity.
8. Using instead of On in join: join tables payments and client on client\_id. Then join it to table payment\_method to payments\_method\_id. See when you can use using in this exercise.
9. Join with rows, Union: From custome\_id table, select columns ’customer\_id, first\_name,points and give points>3000 as Gold and for 2000<points<3000 ‘silver and for less that 2000 ‘bronze’. These medals should be given in new columns named type.
10. Insert two rows in the products table, name persons faezeh o ali.
11. Creating a copy of a table: from invoices table, create a copy of order table named order\_archived where only oderd\_date before 2019-01-01 is copied.
12. Updating single row. Update the first row in invoice table to new value for payment\_total to 45 and payment\_date to 2019-03-01
13. Update multiple row. Change all the rows with cliend\_id=3 to payment\_total+5000. Here you have to change the setting to un-safe mode
14. Using subqueries in update: suppose we have the previous exercise but we do not know client\_id, but we only know names of client from client table. solve it.
15. Deleting rows. Delete row in table invoice correspond to client\_id ‘Myworks’ in client\_id table