select a.id, b.name, a.request, a.response, a.created\_at, a.responded\_at, s.name  
from kx\_commands\_logs a join kx\_commands b on a.command\_id = b.id  
join kx\_stations s on a.station\_id = s.id  
where a.response like '500%'

**SQL**

**10-11 -14-15**

Structural query language

To query or modify data. In costumer table, there is an id number that is linked to orders table. When an id number change his address or number or anything else we do not need to change his features erverywhere. We just put his id number in order table and the other infos are in the costumer table so we can edit them just in one table.

## 2. Retrieving Data From a Single Table

Use;

Select \*

From table

--Where column a = value

Order by column b

2:

select \* from sql\_store.customers; -- sql\_store is a database

select distinct

where sate='VA' , where points >= 3000 , where birth\_date > '1990-01-01'

order of and , or : () and or   
where birth\_date >= '1990-01-01' or points>1000 and sate = ‘va’ :   
where birth\_date >= '1990-01-01' or ( points>1000 and sate = ‘va’ )

where not ( birth\_date >= '1990-01-01' or points>1000 )

where state not in ('va','fl','ga')

where points between 1000 and 3000  
where birth\_date between '1990-01-01' and '2000-01-01'

where last\_name like 'b%' , where last\_name like '%b%' : % any amount of charecters  
where last\_name like '\_\_\_\_\_y' : \_ single character  
where address like '%trail%' or address like '%avenue%'  
where phone not like '%9'

where last\_name regexp 'field'   
where last\_name regexp '^field' : ^ start with field  
where last\_name regexp 'field$' : $ end with field  
where last\_name regexp 'field|mac' : | search for field or mac  
where last\_name regexp 'field$|^mac|rose'  
where last\_name regexp '[gim]e' -- ge ie me: []  
where last\_name regexp '[a-e]e' -- ge ie me : [c-g] c to g

where phone is null , where phone is not null

order by first\_name desc

order by state desc , first\_name

[ select first\_name, last\_name, 10 as points  
from customers  
order by state desc , first\_name ]

order by quantity\*unit\_price desc : order can be based on arithmetic between columns

limit 3 : show just 3 rows of result

limit 6,3 : ignore 6 first rows and show next 3 rows

order of clauses : use - select – from – where – order by – limit

## 3. Retrieving Data From Multiple Tables

select order\_id, customers.customer\_id, first\_name, last\_name  
from orders  
join customers on orders.customer\_id = customers.customer\_id

abbriveation:

select order\_id, c.customer\_id, first\_name, last\_name  
from orders o   
 join customers c  
 on o.customer\_id = c.customer\_id

from 2 database :

use sql\_inventory;  
select \*  
from sql\_store.order\_items oi  
join products p on oi.product\_id = p.product\_id

self join:

use sql\_hr;  
select e.employee\_id, e.first\_name,m.first\_name as manager  
from employees e  
join employees m on e.reports\_to = m.employee\_id

multiple tables:

use sql\_invoicing;  
select \*  
from payments p  
join clients c on p.client\_id = c.client\_id  
join payment\_methods pm on pm.payment\_method\_id = p.payment\_method

join on multiple condition:

select \*  
from order\_items oi  
join order\_item\_notes oin   
 on oi.order\_id=oin.order\_id  
 and oi.product\_id = oin.product\_id

outer join:

select c.customer\_id , c.first\_name , o.order\_id  
 from customers c  
 left (right) join orders o on c.customer\_id=o.customer\_id  
order by c.customer\_id

on c.customer\_id = p.customer\_id : using(customer\_id)

natural join : it will detect common columns automatically

cross join : all the rows of table will be combined with all the rows of the other table

union : the output tables should have same number of columns

select customer\_id,first\_name,points, 'bronze' as type , 'sasan'  
from customers  
where points < 2000  
union  
select customer\_id,first\_name,points, 'silver' as type, 'sasan'  
from customers  
where points between 2000 and 3000

## 4. Inserting, Updating, and Deleting Data

Insert into tablea (column1.column2,..)  
values (default,null,dd,dchik) , (default,null,dd,dchik), (default,null,dd,dchik)

Hierarchical inserting?

Create table tablename as  
select \* from orders

## 5. Summarizing Data

Select max(invoice\_total) as highest ----- max-min – avg-sum-count(distinct)  
from invoices

Check the code

Group by is always after from – where and befor order by

When you group by. The number of columns in select besides aggfunc should be same as group by columns

Having is similar to where but it affect after the grouping by. But where affect before grouping by on the columns of table. The other difference is having will just be applied on the containing features of select in contrast with where

WITH ROLLUP