DQL commands

-- classes in query

-- Select from where groupby having orderby limit

-- select is used to specify columns

-- from is used to specify the table

-- where is used to filter data present in table

-- select * from emp_exe where age in (28,32,45,56)

-- group by is used divide groups based oon aggregations

-- select city, count(*) from customers group by city;

-- having is used to filter data of an aggregated column data

-- select city, count(*) from customers group by city having count(*) > 1;

-- orderby is used to sort the data in asc/desc order based on any column

-- select * from customers order by customer_id desc;

-- select customer_id, postalcode from customers order by customer_id asc;

-- limit is used to limit the number of record

-- Joins

inner join - it will give only the matching records from both tables

Syntax:

select b.account_id, b.fullname, b.branch, b.gender, t.transaction_id, t.transaction_type, t.transaction_amount

from bank_info as b join transactioninfo as t

on b.account_id = t.account_id;

outer join - both matached and unmatached records will be printed. there are three types in outer joins

-- left join

select b.account_id, fullname, b.branch,t.transaction_id, transaction_amount, transaction_time

from bank_info as b left join transactioninfo as t

on b.account_id = t.account_id

-- right join

select b.account_id, fullname, b.branch,t.transaction_id, transaction_amount, transaction_time
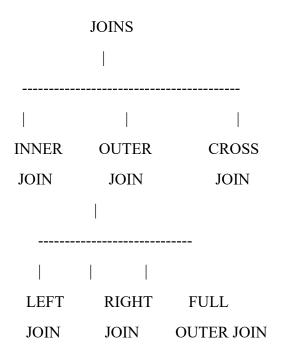
from bank_info as b right join transactioninfo as t

on b.account_id = t.account_id

-- full join

select * from bank_info as b left join transactioninfo as t on b.account_id = t.account_id

union

select * from bank_info as b right join transactioninfo as t on b.account_id = t.account_id;

Left					
1		1		1	
2		2		2	
3	UNION	3	UNION ALL	3	
4		4		4	
Right		5		1	
1				2	
2				3	
3				5	
5					

```
                JOINS
                  |
        ----------------------------------------
        |              |                  |
     INNER         OUTER            CROSS
     JOIN           JOIN             JOIN
                      |
          ----------------------------
          |         |          |
        LEFT      RIGHT      FULL
        JOIN       JOIN      OUTER JOIN
```

-- cross join - this join is also called as cartesian product because it will provide output as all possible combinations.

Syntax:

select * from bank_info, transactioninfo;