DBlab Homework 5A

a)innerjoin with <>, not in

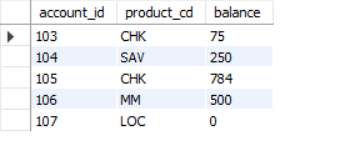
1a) innerjoin with <>: join 2 tables which does not have date ‘2004-04-04’ extract value with <> and join output with other one table which have same product\_cd.

select distinct p.name, output.account\_id, output.product\_cd from Product p inner join

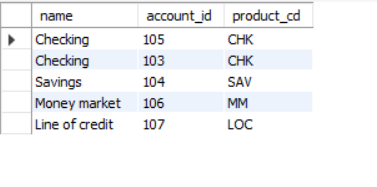
(select Distinct a.account\_id, a.product\_cd, a.balance from Account a inner join Transaction b on a.account\_id where date <> '2004-04-04') as output

on p.product\_cd = output.product\_cd;

inner level join save as output



with outer level join



Discription: In inner level query, I used ‘<>’ to extract values which does not include date'2004-04-04' from table Transaction and joined with table Account using inner join.

In outer level join I used inner join to join output and table Product.

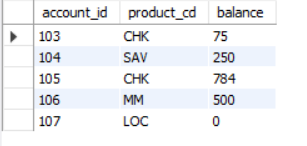
1b) inner join and not in: join 2 tables which does not have date ‘2004-04-04’, extract value with not in and join output with other one table which have same product\_cd.

select distinct p.name, output.account\_id, output.product\_cd from Product p inner join

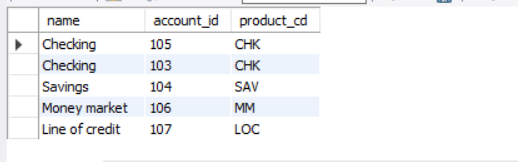
(Select Distinct a.account\_id, a.product\_cd, a.balance from Account a inner join Transaction b on a.account\_id where b.date not in('2004-04-04'))as output

on p.product\_cd =output.product\_cd ;

with inner level output



With out level



Discription: In inner level query, I used ‘not in’ to extract values which does not include date'2004-04-04' from table Transaction and joined with table Account using inner join.

In outer level join I used inner join to join output and table Product.

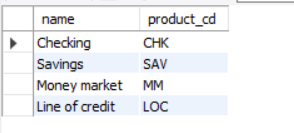
2a) left outer join with Exists:

select distinct p.name, p.product\_cd from Product p

where exists (select Distinct a.account\_id, a.product\_cd, a.balance

from Account a left outer join Transaction b

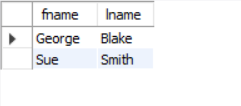
on a.account\_id =b.account\_id and a.balance <> 500);



Duscription: Inner level query, filters datavalues by using join 2 relations on join codition and constant selection operator condition. Using this output display the datavalues from outer level relation.

2b) select distinct fname, lname from Customer c left outer join Account a on c.cust\_id

where not exists( select \* from Account a, Transaction b where a.account\_id=102);



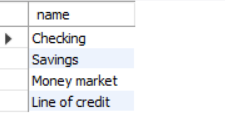
Discription: Inner level query filters datavalues by using constant selection operator condition .

I used left outer join to filter using join condition.

3a)right outer join:

Select distinct p.name from Product p right outer join Account a on p.product\_cd = a.product\_cd

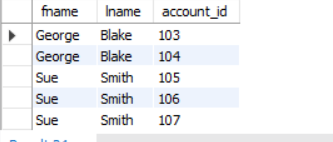
where not exists (select p.name from Product p where p.name='Saving' );



Discription: Inner level query filters datavalues by using constant selection operator condition .

I used right outer join to filter using join condition.

3b) select distinct DISTINCT c.fname, c.lname,a.account\_id from Customer c right outer join Account a on c.cust\_id= a.cust\_id where exists (select \* from Account a where not exists (select DISTINCT \* from Transactions b where b.account\_id = 107));

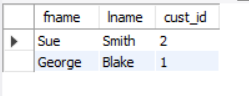


Discription: Inner level query filters datavalues by using constant selection operator condition and not exists.

I used left outer join to filter using join condition.

4a) Equijoin:

select DISTINCT c.fname, c.lname, c.cust\_id from Customer c join Account a on c.cust\_id=a.cust\_id join Product p on a.product\_cd=p.product\_cd;

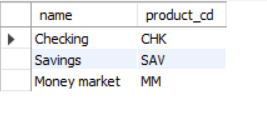


Discription: Inner level query filters datavalues by using equal operator condition .

I used join to filter using join condition.

4b)

select DISTINCT p.name, p.product\_cd from Product p join Account a on p.product\_cd = a.product\_cd join Transactions b on a.account\_id = b.account\_id;

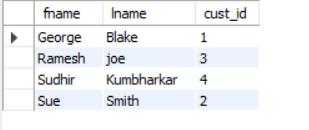


Discription: Inner level query filters datavalues by using equal operator condition .

I used join to filter using join condition.

5a) non equijoin:

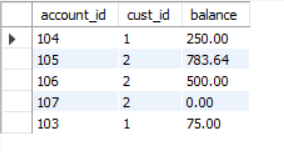
select Distinct fname, lname, c.cust\_id from Customer c, Account a, Product p where c.cust\_id <> a.cust\_id AND a.product\_cd <> p.product\_cd;



Discription: Inner level query filters datavalues by using <> selection operator condition .

I used join to filter using join condition.

5b) select Distinct a.account\_id, a.cust\_id, a.balance from Account a, Product p, Transactions b where a.product\_cd <> p.product\_cd AND a.account\_id <> b.account\_id;

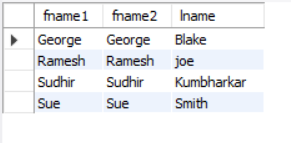


Discription: Inner level query filters datavalues by using <> selection operator condition .

I used join to filter using join condition.

6a)self join:

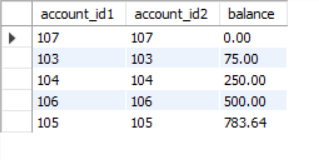
select a.fname as fname1, b.fname as fname2, a.lname from Customer a, Customer b where a.cust\_id = b.cust\_id AND a.lname=b.lname order by a.lname;



Discription: Inner level query filters datavalues by using ‘=’ selection operator condition .

I used join to filter using join condition.

6b) select a.account\_id as account\_id1, b.account\_id as account\_id2, a.balance from Account a, Account b where a.cust\_id = b.cust\_id AND a.balance = b.balance ORDER BY a.balance;

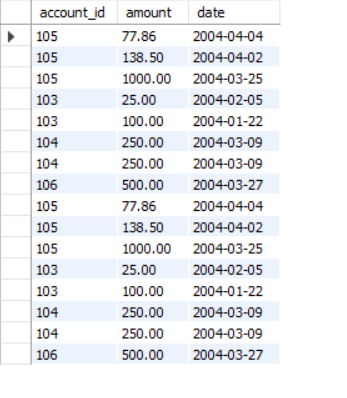


Discription: Inner level query filters datavalues by using = selection operator condition .

I used join to filter using join condition.

7a) natural join:

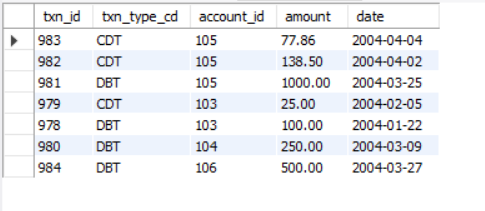
select b.account\_id, b.amount, b.date from Transactions b natural join Account a natural join Product p where a.product\_cd=p.product\_cd;



Discription: Inner level query filters datavalues by using <> selection operator condition .

I used natural join to filter using join condition.

7b) select DISTINCT b.\* from Transactions b natural join Account a natural join Product p where a.product\_cd <> 107;

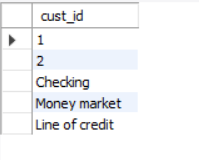


Discription: Inner level query filters datavalues by using <> selection operator condition .

I used join to filter using join condition.

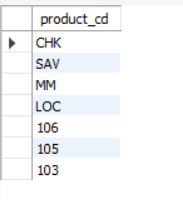
8a)union:

(select DISTINCT c.cust\_id from Customer c inner join Account a on c.cust\_id = a.cust\_id) Union (select DISTINCT name from Product p inner join Account a on p.product\_cd= a.product\_cd where p.name <> 'Savings');



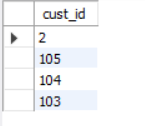
Description: I used Union to join 2 set by subqueries. In One set I used inner join to filter datavalues using join condition. In second set I used second inner join using join condition and <> operator condition.

8b) (select distinct a.product\_cd from Account a inner join Product p on a.product\_cd=p.product\_cd) union (select distinct account\_id from Transactions b join Product p where account\_id <> 104);



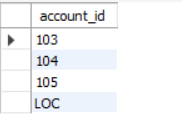
Description: I used union to combine 2 sets of datavalues. In one set I used inner join using join condition to filter datavalues. In second set, I used join and <> operator condition to filter the datavalues.

9a)union all: (select distinct a.cust\_id from Account a join customer c where a.cust\_ID <> 1 Order by cust\_id)Union all (select distinct b.account\_id from Transactions b join Product p where b.account\_id < 106);



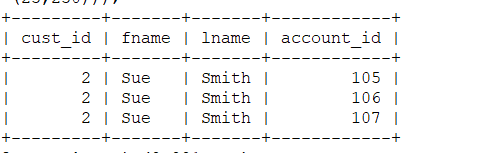
Description: I used union all to combine 2 sets of datavalues. In one set, I used join using join condition and <> operator condition to filter datavalues. In second set, I used join and <> operator condition to filter the datavalues.

9b) (select distinct account\_id from Account a where account\_id <= 105)Union all (select distinct a.product\_cd from account a join Product p on a.product\_cd = p.product\_cd where a.product\_cd IN (SELECT product\_cd FROM Product WHERE product\_cd='LOC'));



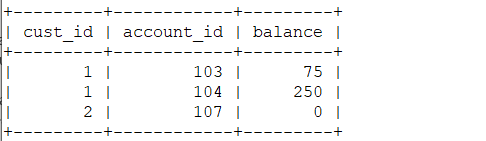
Description: I used union all to combine 2 sets of datavalues. In one set, I used <> operator condition to filter datavalues. In second set, I used join and In clause and ‘=’ operator condition to filter the datavalues.

10a) intersect: intersect: select c.cust\_id,c.fname, c.lname, a.account\_id from Customer c join Account a on c.cust\_id=a.cust\_id where a.account\_id not in ((select account\_id from Account)intersect(select account\_id from Transactions where amount in (25,250)));



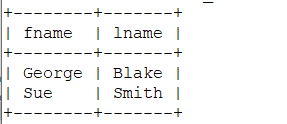
Description: I used union all to combine 2 sets of datavalues. In one set, I used <> operator condition to filter datavalues. In second set, I used join and In clause and ‘=’ operator condition to filter the datavalues.

10b) select a.cust\_id,a.account\_id, a.balance from Product p join Account a on p.product\_cd=a.product\_cd where a.account\_id not in ((select account\_id from Account)intersect(select account\_id from Transactions where amount in (1000,500)));



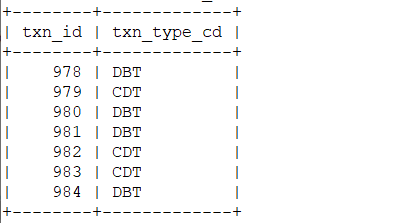
Description: I used intersect to filter datavalues from 2 sets. In one set, I used select condition to filter datavalues. In second set, I used In clause to filter the datavalues. I used ‘not in’ to skip the output datavalues from intersect used these values to associate the join condition datavalues.

11a)Except: select distinct fname,lname from Customer where not exists((select product\_cd from Product where product\_cd='SA')Except(select account\_id from Account where cust\_id =4));



Description: To skip values, I used ‘Except’ and ‘=’ operator condition. The datavalues filterd using ‘not exists’ and ‘=’ operator condition.by skiping both values, remaining datavalues are the final datavalues to display.

11b) select txn\_id, txn\_type\_cd from Transaction where not exists ((select product\_cd from Product where name= 'Sav')except (select account\_id from Account where cust\_id=5));



Description: To skip values, I used ‘Except’ and ‘=’ operator condition. The datavalues filterd using ‘not exists’ and ‘=’ operator condition.by skiping both values, remaining datavalues are the final datavalues to display.