2. Mark each pair of relational algebraic expressions below as either EQUIVALENT or DIFFERENT. In each of the case you mark as DIFFERENT, provide an example of relations that show the expressions are not equivalent.

For the expressions, we use the following symbols:

- θ1 θ2 ..... To denote predicates
- L1, L2 .....to denote lists of attributes
- E1,E2..... To denote relation algebra expressions
- |><| to denote natural join operation
- $|><|_{\theta}$ to denote theta(conditional) join operation
- =X, X=, =X= to denote left, right, full outer join operation
- A ξ F to denote group by operation, where A are the group by dimensional attributes and F are the aggregates measures

1). σ ( θ 1 and θ 2 )(E)== σ θ 1( σ θ 2(E)) E or D

2). σ θ 1 ( σ θ 2 (E))== σ θ 2( σ θ 1(E)) E or D

3). E1 full outer theta join E2 == E2 full outer theta join E1 E or D

4). (E1 full outer join E2) full outer join E3) ==E1 full outer join (E2 full outer join E3)

5). (E1 ∩ E2) ∩ E3==E1 ∩ (E2 ∩ E3) E or D

4. Give an SQL schema definition for the employee database .Define an appropriate domain for each attribute and an appropriate primary key for each relation schema

Employee(employee_name, street, city)

Works(employee_name,company_name,salary)

Company(company_name,city)

Manages(employee_name,manager_name)

3.18 Give an SQL schema definition for the employee dat
an appropriate domain for each attribute and an a
each relation schema.
**Answer:**

```
create domain   company_names char(20)
create domain   city_names char(30)
create domain   person_names char(20)

create table    employee
(employee_name  person_names,
 street         char(30),
 city           city_names,
 primary key    (employee_name))

create table    works
(employee_name  person_names,
 company_name   company_names,
 salary         numeric(8, 2),
 primary key    (employee_name))

create table    company
(company_name   company_names,
 city           city_names,
 primary key    (company_name))

create table    manages
(employee_name  person_names,
 manager_name   person_names,
 primary key    (employee_name))
```

5. Answer the following questions.

1). Do the following queries produce the same results? If not, what are the difference? And how would you change the queries to produce the same result?

Q1. Select * from R;

Q2. (select * from R) intersect (select * from R);

Translate the following queries to relational algebraic expressions based on the following relations:

Branch(branch_name, branch_city,assets)

Customer(customer_name,customer_street,customer_city)

Account(account_number,branch_name,balance)

Loan(loan_number,branch_name,amount)

Depositor(customer_name,account_number)

Borrower(customer_name,loan_number)

2). Find the names of all customers who have a loan at the perryridge branch.

3). Find the names of all customers who have a loan at the perryridge branch but don't have

an account at any branch of the bank.

- Find the names of all customers who have a loan at the Perryridge branch.

$$\Pi_{customer\_name}(\sigma_{branch\_name=\text{“Perryridge”}}(\sigma_{borrower.loan\_number = loan.loan\_number}(borrower \times loan)))$$

- Find the names of all customers who have a loan at the Perryridge branch but do not have an account at any branch of the bank.

$$\Pi_{customer\_name}(\sigma_{branch\_name = \text{“Perryridge”}}(\sigma_{borrower.loan\_number = loan.loan\_number}(borrower \times loan))) - \Pi_{customer\_name}(depositor)$$

6.Based on the sales data we have been using in class, compute the following query(write the query in SQL)

For each product, count for each month of 1995 how many sales of the previous and how many sales of the following month had quantity greater than that month's average sales.

| PRODUCT | MONTH | BEFORE_TOT | AFTER_TOT |
|---|---|---|---|
| ======== | ====== | ========== | ========== |

第六题答案：

```
create view v1 as
select product,month, avg(quant) as avg_q
from sales
where year=1995
group by product, month

create view v2 as
select v1.product as product, v1.month as month, count(sales.quant) as PREVIOUS_TOT
from sales,v1
where sales.year=1995 and sales.product=v1.product and sales.month=v1.month-1 and sales.quant>v1.avg_q
group by v1.product ,v1.month

create view v3 as
select v1.product as product, v1.month as month, count(sales.quant) as FOLLOWING_TOT
from sales,v1
where sales.year=1995 and sales.product=v1.product and sales.month=v1.month+1 and sales.quant>v1.avg_q
group by v1.product ,v1.month

select v2.product, v2.month,v2.PREVIOUS_TOT,v3.FOLLOWING_TOT
from v2,v3
where v2.product=v3.product and v2.month=v3.month
```

7. Answer the following questions on attribute set closures, Functional Dependencies and

Decompositions:

1).Given R(A,B,C) and the following functional dependencies, computer A+.

AB→C

BC→A

AC→B

2). Suppose that we decompose the schema R=(A,B,C,D,E) into (A,B,C) and (A,D,E). Show that this composition is a lossless-join decomposition if the following set F of functional dependencies holds:

A→BC

CD→E

B→D

E→A

3). Given the schema R=(A,B,C,D,E) and the same set F of functional dependencies as above,show how the following functional dependencies are derived:

A→E

A→ABCDE

第七题第三小题答案：
1. A->BC and B->D,
so A->CD,
and CD->E,
so A->E.

2. A->BC so A->B, so A->D,
and A->A AND A->BC AND A->E and A->D,
so A->ABCDE

8. Given the following:

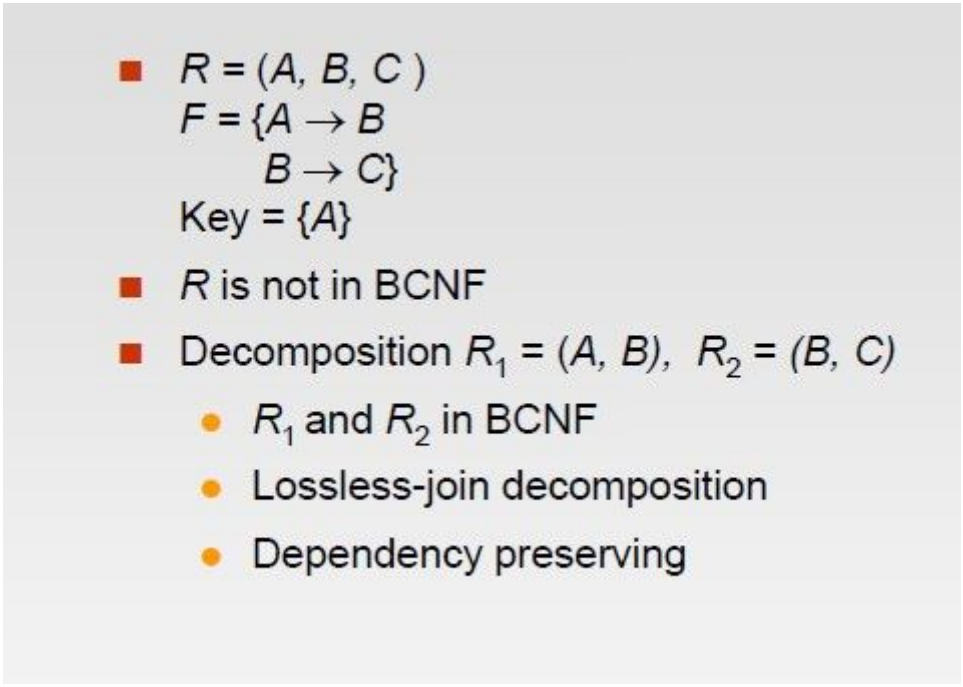
R=(A,B,C) and the functional dependencies below

A→B

B→C

1). Show R is not in BCNF

2) decompose R-i.e., list the decomposed relations R1,R2,R3..




3).show the newly decomposed relations are in BCNF

4). Show it was a lossless-join decomposition

5). Show the decomposition was dependency preserving