CS 411: Database Systems

Fall 2021

Homework 2 (Due by 23:59 CT on Oct 1, 2021)

Logistics

1. This homework is due on Oct 1st, 2021 at 23:59 CT. **We DO NOT accept late homework submissions.**
2. You will be using Gradescope to submit your solutions. Answer each sub-question (e.g. “a.”, “b.” etc.) on a **new page** and submit your solution as a single PDF file on Gradescope. All registered students should have received an email invitation to Gradescope. Please submit the PDF to "Homework 2".
3. ***IMPORTANT*: Please make sure to link PDF pages with the corresponding question outline on Gradescope.**
4. The answers can be written electronically or they can be hand-written, but if we cannot read your submissions, we won’t be able to grade them.
5. If you are looking for tools to create ER diagrams etc, consider draw.io.
6. Please write down any intermediate steps to receive full credit.
7. Keep your solutions brief and clear.
8. Please use Campuswire if you have questions about the homework but **do not post answers.** Feel free to use private posts or come to office hours.

Rubric (IMPORTANT!)

1. Always underline primary keys in ER diagrams and UML diagrams.
2. For the questions about ER and UML diagrams, address as many constraints implied in the problem description as possible. Explicitly state any extra assumptions you make that cannot be derived from the description.
3. Follow the question description and do not make assumptions from the real world.
4. When drawing ER diagrams or converting an ER diagram to a relational schema, have the following design principles in mind:
    a. Try not to create unnecessary entities
    b. Try not to create tables that might suffer redundancy

Section 1. ER and Relational Schema

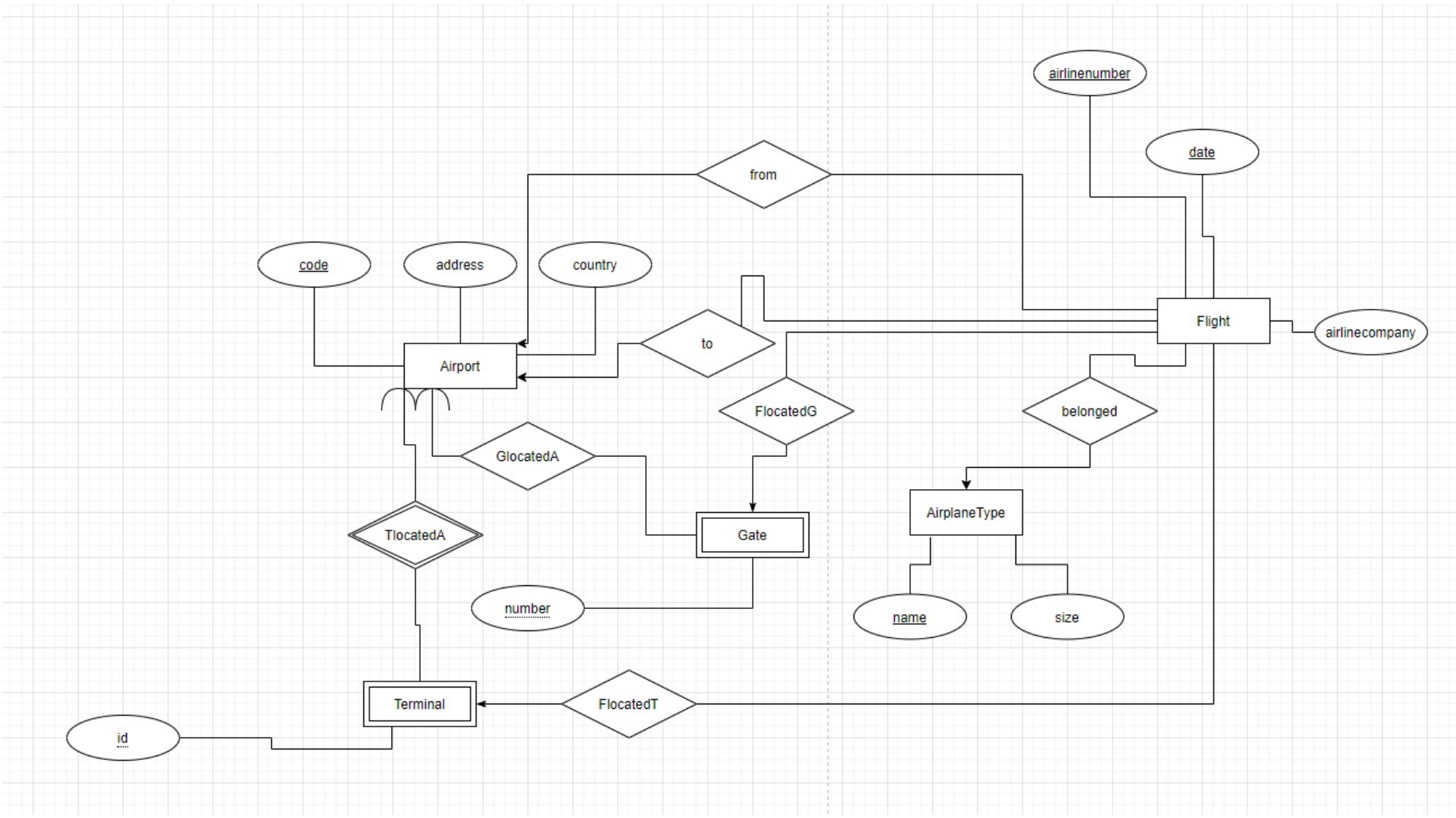
Question 1. ER Diagram (15 pts)

You are working for the O'hare Airport. Your manager tasks you with designing a database for the airport service. Given the following information, draw the ER diagram:

- The database should store information about Airport, Terminal, Gate, Flight, and Airplane Type.
- An Airport is uniquely identified by its code. Other airport attributes are address and country.
- A Terminal is uniquely identified by its ID and the airport where it is located at.
- A Gate is uniquely identified by its gate number and the airport where it is located at.
- A Flight is uniquely identified by its airline number and date of flight. Each flight also includes the airline company name as an additional attribute.
- An Airplane type is uniquely identified by its name. Airline type also includes size as another attribute.

- An Airport may have multiple Terminals, but each Terminal is located at exactly one Airport.
- An Airport may have multiple Gates, but each Gate is located at exactly one Airport.
- A Flight is at one Gate, but each Gate may have multiple Flights.
- A Flight is at one Terminal, but each Terminal may have multiple Flights.
- A Flight is of one Airplane Type, but multiple Flights can have the same Airplane Type.
- A Flight is from one Airport and to one Airport, and each Airport may have several Flights departing from and arriving at it.

Q1:

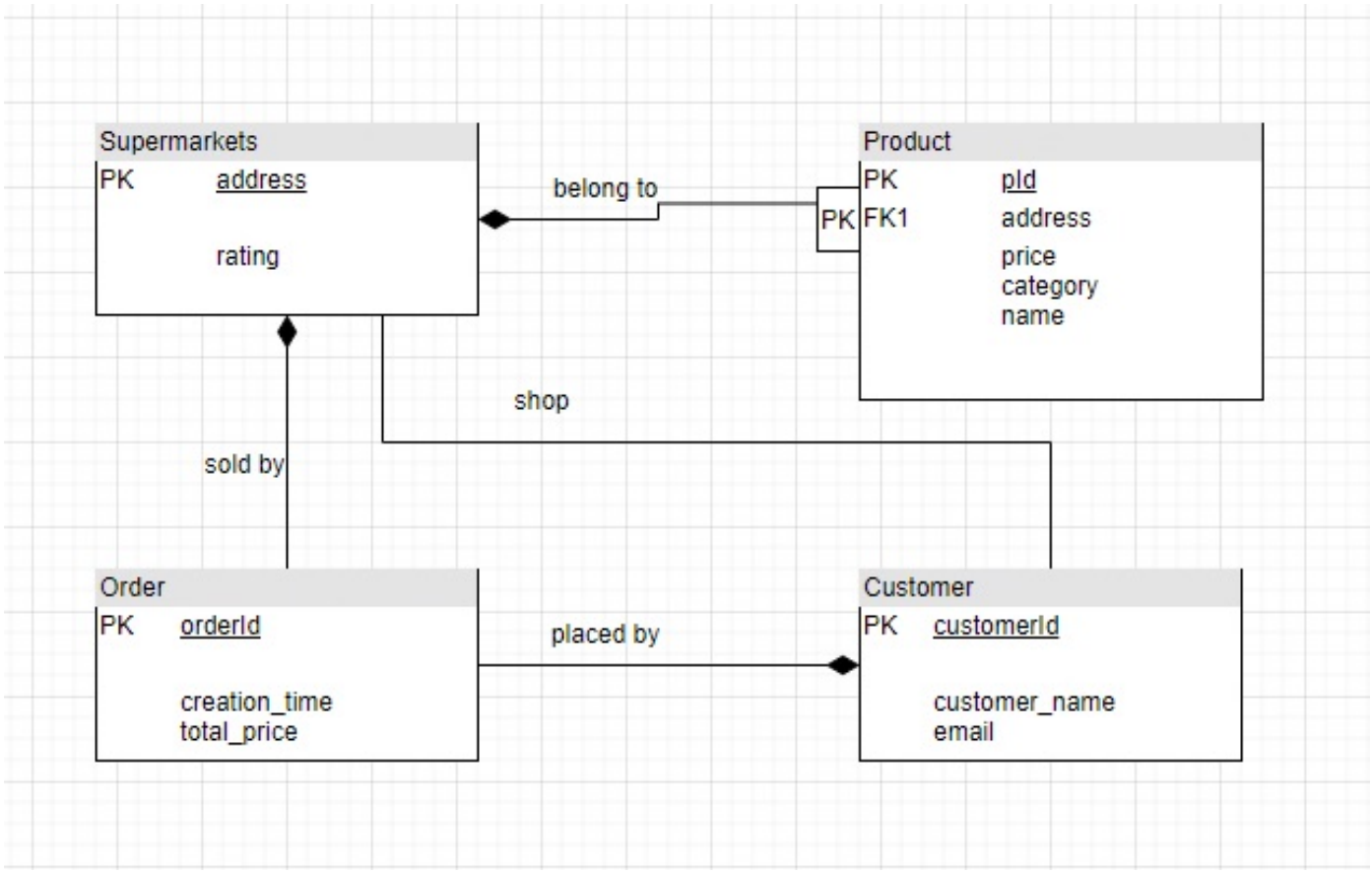

airlinenumber
date
from
code
address
country
Flight
airlinecompany
to
Airport
FlocatedG
belonged
GlocatedA
AirplaneType
TlocatedA
Gate
number
name
size
Terminal
FlocatedT
id

Question 2. UML Diagram (15 pts)

You are hired by Sam's Club to manage their database. Your first task is to design tables to store information on Stores and Customer. Given the following information, draw the UML diagram:

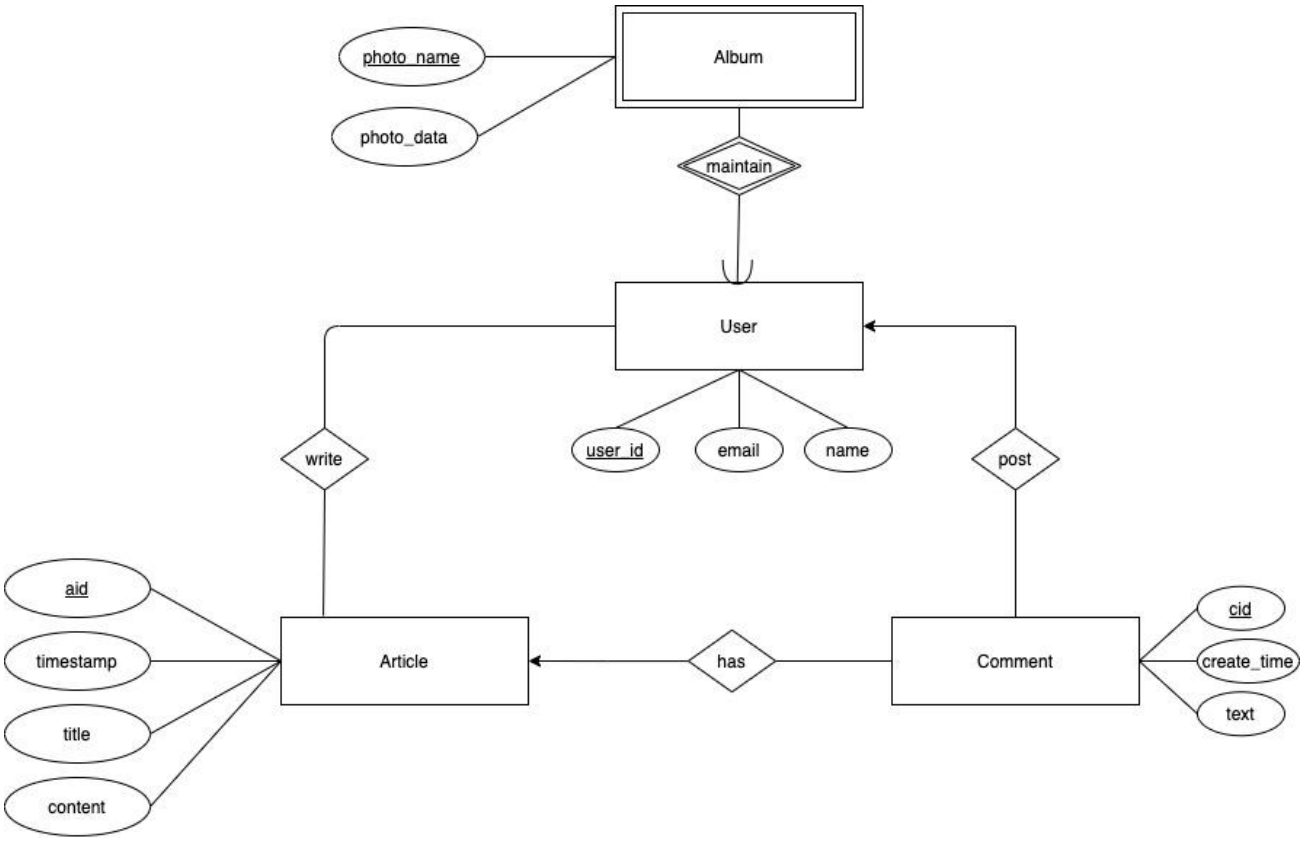
- The database has Supermarkets. Each is uniquely identified by its address, and has an attribute, rating.
- Each Product is uniquely identified by its pid and the supermarket where it belongs to, and has attributes price, category and name.
- Each Order is uniquely identified by its order id, and has attributes "creation time" and "total price". An Order is made by exactly one Supermarket, but a Supermarket may have multiple Orders.
- Each Customer is uniquely identified by its customer id, and has attributes name and email. A Customer may shop at multiple Supermarkets, and a Supermarket can have multiple Customers. A Customer may have multiple Orders, but an Order is made by exactly one Customer.

Q2:


Supermarkets
PK address
rating
belong to
Product
PK pId
PK FK1 address
price
category
name
shop
sold by
Order
PK orderId
creation_time
total_price
placed by
Customer
PK customerId
customer_name
email

Question 3. **Conversion From ERD to Relational Schema (10 points)**

Convert the below ER diagram into a relational schema, and write the DDL SQL commands for implementing it as a database schema.



Q3: DDL:

```
CREATE TABLE User (
user_id         VARCHAR(20),
name            VARCHAR(20),
email           VARCHAR(20),
PRIMARY KEY (user_id)
);

CREATE TABLE Album (
photo_name      VARCHAR(20),
photo_data      VARCHAR(max),
PRIMARY KEY (photo_name,user_id),
FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
);

CREATE TABLE Article (
aid             VARCHAR(20),
timestamp       VARCHAR(20),
   title           VARCHAR(20),
   content         VARCHAR(max),
   PRIMARY KEY (aid)
   );


   CREATE TABLE Comment (
   cid             VARCHAR(20),
   create_time     VARCHAR(20),
   text            VARCHAR(max),
   PRIMARY KEY (cid)
   );
```

Relational scheme:

```
User( PK:user_id VARCHAR(20),
name VARCHAR(20),
email VARCHAR(20))

Album( PK:photo_name VARCHAR(20),
      FK:user_id VARCHAR(20),
      photo_data VARCHAR(max))

Article( PK:aid VARCHAR(20),
timestamp VARCHAR(20),
title VARCHAR(20),
content VARCHAR(max))

Comment( PK:cid VARCHAR(20),
create_time VARCHAR(20),
text VARCHAR(20))
```

Section 2. Functional Dependencies and Normal Forms

Question 4. Attribute Closure and Functional Dependencies (20 pts)

Given the following relation:

Harry_Potter(Spell, Achievement, Wizard, Talent, Professor)

And functional dependencies:

Wizard ⟶ Talent

Wizard, Talent, Professor ⟶ Spell

Achievement ⟶ Wizard

Spell, Wizard⟶ Achievement

1. Solve X+ for every subset of the relation's attributes. **(14 points)**

2. Find all candidate keys. **(2 points)**

3. Find four non-trivial functional dependencies that are different from the ones listed in the question. **(4 points)**

Q4:
1.

ONE:

{Spell} += {Spell}

{Achievement} += {Achievement, Wizard, Talent}

{Wizard} += {Wizard, Talent}

{Talent} += {Talent}

{Professor} += {Professor}

TWO:

{Spell, Achievement} += {Spell, Achievement, Wizard, Talent}

{Spell, Wizard} += {Spell, Wizard, Talent, Achievement}

{Spell, Talent} += {Spell, Talent}

{Spell, Professor} += {Spell, Professor}

{Achievement, Wizard} += {Achievement, Wizard, Talent}

{Achievement, Talent} += {Achievement, Wizard, Talent}

{Achievement, Professor} += {Achievement, Wizard, Talent, Professor, Spell}

{Wizard, Talent} += {Wizard, Talent}

{Wizard, Professor} += {Wizard, Talent, Professor, Spell, Achievement}

{Talent, Professor} += {Talent, Professor}

THREE:

{Spell, Achievement, Wizard} += {Spell, Achievement, Wizard, Talent}

{Spell, Achievement, Talent} += {Spell, Achievement, Wizard, Talent}

{Spell, Achievement, Professor} += {Spell, Achievement, Wizard, Talent, Professor}

{Spell, Wizard, Talent} += {Spell, Wizard, Talent, Achievement}

{Spell, Wizard, Professor} += {Spell, Wizard, Talent, Achievement, Professor}

{Spell, Talent, Professor} += {Spell, Talent, Professor}

{Achievement, Wizard, Talent} += {Achievement, Wizard, Talent}

{Achievement, Wizard, Professor} += {Achievement, Wizard, Talent, Professor, Spell}

{Achievement, Talent, Professor} += {Achievement, Wizard, Talent, Professor, Spell}

{Wizard, Talent, Professor} += {Wizard, Talent, Professor, Spell, Achievement}

FOUR:

{Spell, Achievement, Wizard, Talent} += {Spell, Achievement, Wizard, Talent}

{Spell, Achievement, Wizard, Professor} += {Spell, Achievement, Wizard, Talent, Professor}

{Spell, Achievement, Talent, Professor} += {Spell, Achievement, Wizard, Talent, Professor}

{Spell, Wizard, Talent, Professor} += {Spell, Achievement, Wizard, Talent, Professor}

{Achievement, Wizard, Talent, Professor} += {Spell, Achievement, Wizard, Talent, Professor}

FIVE:

{Spell, Achievement, Wizard, Talent, Professor} += {Spell, Achievement, Wizard, Talent, Professor}

2.
{Achievement, Professor}
{Wizard, Professor}

3.

{Achievement} -> {Talent}
{Wizard, Professor} -> {Spell, Achievement}

Question 5. Normal Forms (40 pts)

a. Consider the following relations R1, R2, R3. Which normal forms (BCNF,3NF) is each relation in? Explain why the relation is or is not in each of these normal forms. List all violating FDs. **(15 pts)**

   i. R1 = (A,B,C,D,E,F) with a set of functional dependencies FD = {AB->CE; BE->F; B->AD;}

   ii. R2=(A,B,C,D,E,F) with a set of functional dependencies FD={AC->FB; F->CE; B->C;}

   iii. R3=(A,B,C,D,E) with a set of functional dependencies FD={C->BD;A->B;E->B}

b. Given a relation R (A, B, C, D, E, F) and functional dependencies FD = {A->C, F->C, D->BF, CF->AE, DE->C} **(12 pts)**

   i. Compute the minimal basis of FD. Show all steps to receive full credits. (7 pts)

   ii. Decompose the relation R into a set of relations that are in 3NF.(5pts)

c. Given a relation R (A,B,C,D,E,F) and functional dependencies FD = {D->F, B->AC, BD->E, A->F}.**(13 pts)**

   Decompose the relation R into a set of relations that are in BCNF. You must list all your steps to receive full points.

Q5:

a. Consider the following relations R1, R2, R3. Which normal forms (BCNF,3NF) is each relation in? Explain why the relation is or is not in each of these normal forms. List all violating FDs. **(15 pts)**
   i. R1 = (A,B,C,D,E,F) with a set of functional dependencies FD = {AB->CE; BE->F; B->AD;}
   ii. R2=(A,B,C,D,E,F) with a set of functional dependencies FD={AC->FB; F->CE; B->C;}
   iii. R3=(A,B,C,D,E) with a set of functional dependencies FD={C->BD;A->B;E->B}

(a)

i:

{A,B}+ = {A, B, C, E, F}

{B,E}+ = {B, E, F, A, D, C}

{B}+ = {B, A, D, C, E, F}

Thus, it is BCNF and 3NF since all LHS are super key

(ii) {A,C}+ = {A, C, F, B, E}
{F}+ = {F, C, E}
{B}+ = {B, C}
=> not BCNF

super key is {A,C,D}, while {AC->FB; F->CE} is not part of super key

So, not 3NF

∴ It's not BCNF or 3NF

(iii) {C}+ = {C, B, D}
{A}+ = {A, B}
{E}+ = {E, B}
not BCNF

Super key: {A, C, E}, while {C->BD, A->B, E->B} are not part of super key, not 3NF

∴ It's not BCNF or 3NF

b. Given a relation R (A, B, C, D, E, F) and functional dependencies FD = {A->C, F->C, D->BF, CF->AE, DE->C} **(12 pts)**

i. Compute the minimal basis of FD. Show all steps to receive full credits. (7 pts)

ii. Decompose the relation R into a set of relations that are in 3NF.(5pts)

(i) $A \to C$, $F \to C$, $D \to BF$, $F \to AE$, $DE \to C$

$F \to AE$ ⇓ $\therefore F \to C$, $F \to A$, $F \to E$

$DE \to C$ ⇓ $\therefore D \to BF$, $F \to C$, $\therefore D \to C$

$\therefore \{A \to C, \cancel{F \to C}, D \to B, D \to F, F \to A, F \to E, \cancel{D \to C}\}$

If remove $A \to C$, $\{A\}+ = \{A\}$
$\therefore A \to C$ remain

remove $F \to C$, $\{F\}+ = \{F, A, C, E\}$, so remove

remove $D \to B$, $\{D\}+ = \{D, F, A, E, C\}$, $\therefore D \to B$ remain

remove $D \to F$, $\{D\}+ = \{D, B, C\}$, $\therefore D \to F$ remain

remove $F \to A$, $\{F\}+ = \{F, E\}$, $\therefore F \to A$ remain

remove $F \to E$, $\{F\}+ = \{F, A, C\}$, $\therefore F \to E$ remain

remove $D \to C$, $\{D\}+ = \{D, F, A, E, C\}$, $\therefore$ remove

$\therefore$ Thus, the minimal basis: $\{A \to C, D \to B, D \to F, F \to A, F \to E\}$

(ii) $R_1(A, C)$, $R_2(D, B)$, $R_3(D, F)$, $R_4(F, A)$, $R_5(F, E)$

no attribute is left, done

c. Given a relation R (A,B,C,D,E,F) and functional dependencies FD = {~~D->F~~, B->AC, BD->E, ~~A->F~~}.**(13 pts)**

Decompose the relation R into a set of relations that are in BCNF. You must list all your steps to receive full points.

minimal bases : $D \to F$, $B \to A$, $B \to C$, $BD \to E$, $A \to F$

$R(A,B,C,D,E,F)$

$FD = D \to F$

$\therefore R_1(D,F)$, $R_2(D,A,B,C,E)$

$FD = B \to A$

$\therefore R_3(B,A)$, $R_4(B,C,D,E)$

$FD = B \to C$

$\therefore R_5 = (B,C)$, $R_6(B,D,E)$

$FD = BD \to E$ BCNF ✓

$\therefore$ Thus, $R_1(D,F)$, $R_2(B,A)$, $R_3(B,C)$, $R_4(B,D,E)$