|  |
| --- |
| **CIS 450/550: Database and Information Systems** |
| **Homework 3 ‒ ER and Relational Design Theory** |

Due by by 23:59:59 EDT on June 17, 2019

This homework is about relational database design, from ER diagrams to relational schemas and their normalization. You should read the assigned readings from Chapters 2, 3 and 9 (see syllabus), and submit your answers via Gradescope.

For ER diagrams, you can create an electronic version of each diagram using draw.io, Powerpoint, Word, Visio, OpenOce Draw, OmniGrae, Google Drive Document or any other software that you are familiar with, and export it to PDF format. Alternatively, you can draw by hand, take a picture, embed in a document, and export to PDF -- but be careful that it is legible. If it is not, we will not grade it!

### Part 1: ER Modeling (30 points)

*You are living with your friends in an apartment, and have been adding all expenses incurred along with the payee name to an excel sheet; periodically, you settle up the balances. You realize that this is too tedious for everyone, and that you need an application to make your life easier. Being a pro-developer you refuse to use existing apps such as Splitwise or BillMonk, and would like to develop one yourself to manage the expenses.*

1. (15 points)

Model the data that supports this application using an ER diagram. The elements your model must include are the following:

* *Users* have a unique id, name and email.
* A user can create a *group*, for which they becomes the owner and to which they can add other users. The group should have a unique name among the existing groups.
* Users can add *expenses*. An expense is defined as an amount which you have paid and would like to split with others in a group. Each expense should have a unique id, title (a brief description of what the expense is about), date, amount, picture and notes (the last two are optional).
* To avoid bogus expenses, only owners of the group can add expenses.
* Each member in the group can have a nickname relative to that group.

A close up of a logo

Description automatically generated

Expense(picture(O), notes(O))

2. (5 points)

If you feel that certain constraints cannot be represented in the ER diagram, mention them and the reason why they cannot be represented.

Answer:

The ER diagram cannot identify if the owner add itself as a member, because the ternary relationship has owner\_id, member\_id, group\_name as primary key, and the owner\_id and member\_id can be the same when create table, with the ER diagram, but according to instruction, in the same group the owner cannot be the member.

3. (10 points)

After you release your app, it becomes extremely popular among students. Hence, you would like to make the following changes to your schema and raise lots of money. Show only the part of your schema that changes.

* Normal Members can create only one group, but Premium Members pay an amount (say $5) and can create more than one group.
* You now feel that only owners adding expense is too stringent for users. Hence, you would like to allow any member of the group to add expenses.

A close up of a logo

Description automatically generated

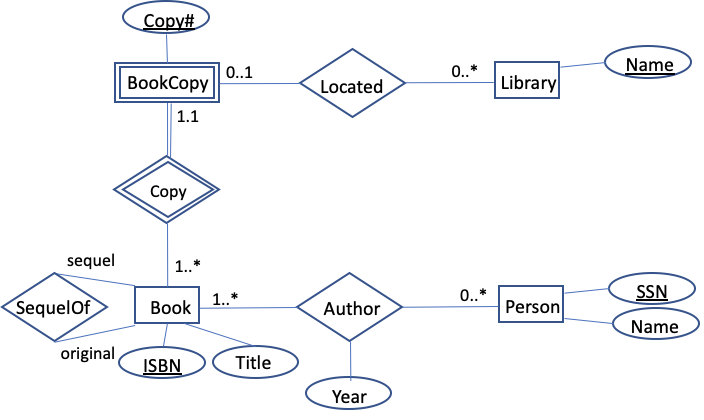
A close up of a logo

Description automatically generated

### Part 2: ER Design to Relations (20 points)

4. (20 points)

Translate the following ER diagram to the relational model by specifying the resulting relations, their attributes, keys, and foreign keys using SQL DDL.



CREATE TABLE Person(

SSN int(9),

Name varchar(20),

PRIMARY KEY (SSN)

);

CREATE TABLE Book(

ISBN int(13),

Title varchar(20),

PRIMARY KEY (ISBN)

);

CREATE TABLE Author(

ISBN int(13),

SSN int(9),

Year int(4),

PRIMARY KEY (ISBN, SSN),

FOREIGN KEY (ISBN) REFERENCES Book(ISBN),

FOREIGN KEY (SSN) REFERENCES Person(SSN)

);

CREATE TABLE SequelOf (

Orign\_ISBN int(9),

Seq\_ISBN int(9),

PRIMARY KEY (Orign\_ISBN),

FOREIGN KEY (Orign\_ISBN) REFERENCES Book(ISBN),

FOREIGN KEY (Seq\_ISBN) REFERENCES Book(ISBN)

);

CREATE TABLE BookCopy (

ISBN int(13),

Copy# int(10),

PRIMARY KEY (ISBN, Copy#),

FOREIGN KEY (ISBN) REFERENCES Book(ISBN)

);

CREATE TABLE Library (

Name varchar(20),

PRIMARY KEY(Name)

);

CREATE TABLE Located (

Lib\_name varchar(20),

Bk\_ISBN int(13),

Bk\_Copy# int(10),

PRIMARY KEY (Bk\_ISBN, Bk\_Copy#),

FOREIGN KEY (Bk\_Copy#) REFERENCES BookCopy(Copy#),

FOREIGN KEY (Bk\_ISBN) REFERENCES Book(ISBN),

FOREIGN KEY (Bk\_Copy#) REFERENCES BookCopy(Copy#)

);

Lib\_name is not a primary key of Located

### Part 2: Relational Design, BCNF and 3NF (50 points)

5. (30 points)

Consider a database of ship voyages with the following attributes: S (ship name), T (type of ship), V (voyage identifier), C (cargo carried by one ship on one voyage), P (port), and D (day). We assume that a voyage consists of a sequence of events where one ship picks up a single cargo, and delivers it to a sequence of ports. A ship can visit only one port in a single day. Thus the following functional dependencies may be assumed:

{S→T, V→SC, SD→PV}

1. (5 points)

What is the key(s) of STVCPD? Make sure they are minimal, i.e. not a superset of a key.

Answer:

The keys are SD or VD

Step 1: Considering the above 3 FDs, only D has appeared in the left but not right. So, D must in the key. S and V has appeared in the both sides of the FDs. So, we can consider putting S, V in the key. T, C, P has only appeared in the right side. This means they cannot determine others but can be determined by others. So, we do not need to consider them in the key. There is no element that never appears in the FDs.

Step 2: Considering (D)+, considering the three FDs, it equals to {D}, it does not include all the attributes, so it cannot be a key.

Considering (SD)+, according to the three FDs, it equals to { STVCPD }, it includes all the attributes, so it can be a key.

Considering (VD)+, according to the three FDs, it equals to { STVCPD }, it includes all the attributes, so it can be a key.

Considering other three attributes, we just prove they do not need to be considered in keys ((TD)+={TD}, ((CD)+={CD}, ((PD)+={PD})

Step 3: from step 1 and 2, we prove the SD or VD is the minimal

So, the key for R={ STVCPD } is SD or VD

1. (10 points)

Find a lossless-join decomposition of STVCPD into BCNF.

Answer:

R={ STVCPD }

F= {S→T, V→SC, SD→PV}

Key=SD(or VD)

Step 1: the left side of S→T is S, S is not the superkey of R, so it does not in BCNF, we need to split it out according to BCNF algorithm:

R1={ST}

F1={ S→T }

Key=S

R2={ SVCPD }

F2={ V→SC, SD→PV }

Key=SD

Step 2: the left side of V→SC is V, V is not the superkey of R2, so it does not in BCNF, we need to split it out according to BCNF algorithm:

R3={VSC}

F3={ V→SC }

Key=V

R4={VPD}

F4={VD🡪PV}? VD->P

Key=VD

Thus, one of the BCNF lossless-join decomposition of R is R1{ST}, R3{VSC}, R4{VPD},but it is not a dependency-preserving decomposition.

1. (10 points)

Find a lossless-join, dependency-preserving decomposition into 3NF.

Answer:

R={ STVCPD }

F={S→T, V→SC, SD→PV}

Step 1: the minimal cover of R={ STVCPD } is F={S→T, V→SC, SD→PV}. And the key of R is SD or VD

Step 2: split it into small subsets according to F:

R1{ST}, R2{VSC}, R3{SDPV}

Step 3: each of these new subset R1, R2, R3, does not have the same left side, so we do not combine them together.

Step 4: the left side of R3{SDPV} is SD, is the key of R, so we do not need to add a new subset that contains the key of R.

Thus, a 3NF lossless-join, dependency-preserving decomposition is R1{ST}, R2{VSC}, R3{SDPV}

1. (5 points)

Explain why there is no lossless-join, dependency-preserving decomposition into BCNF.

ANSWER:

The issue is that V→S and SD→V; you need to keep SDV together to be able to check the dependency (it is not implied by the others), but V→S is a violation of BCNF.

Answer:

We have tried the first way in 5b, and it is not a dependency-preserving decomposition

Way2:

R={ STVCPD }

F= {S→T, V→SC, SD→PV}

Key=SD(or VD)

Step 1: split on S→T of R:

R1={ST}

F1={ S→T }

Key=S

R2={ SVCPD }

F2={ V→SC, SD→PV }

Key=VD

Step 2: split on SD→PV of R2:

R3={SDPV}

F3={ SD→PV }

Key=SD

R4={SCD}

F4={SD🡪C}

Key=SD

Not a dependency-preserving decomposition

Way3:

R={ STVCPD }

F= {S→T, V→SC, SD→PV}

Key=SD(or VD)

Step 1: split on V→SC of R:

R1={VSC}

F1={ V→SC }

Key=V

R2={ TVPD }

F2={ V→T, VD→P }

Key=VD

Step 2: split on V→T of R2:

R3={VT}

F3={ V→T }

Key=V

R4={VPD}

F4={VD🡪P}

Key=VD

Not a dependency-preserving decomposition

Thus, we have considered all the situations, and none of them is a dependency-preserving decomposition. There is not a lossless-join, dependency-preserving decomposition into BCNF for R

6. (20 points)

Consider a relation R = ABCD and the dependencies F= {A→B, B→D, CD→AB}.

1. (5 points) List all the keys for R (make sure they are minimal, i.e. not a superset of some other key).

Answer:

Considering the attributes ABCD and FDs, only C has appeared in the left side but not right side. So, C must be in the key. And (C)+={C}, so C is not the whole key.

Considering (AC)+, according to { A→B, B→D}, (AC)+={ABCD}, so AC is a key.

Considering (BC)+, according to { B→D, CD→AB}, (BC)+={ABCD}, so BC is a key

Considering (CD)+, according to { CD→AB}, (DC)+={ABCD}, so AC is a key.

Thus, the key for R is AC or BC or CD

1. (5 points) Is R in BCNF?

Answer:

No, R is not in BCNF.

No matter key is AC or BC or CD, the left sides for A→B and B→D are A and B, they are not the superkey of R.

1. (5 points) Is R in 3NF?

Answer:

Yes, R is in 3NF.

The primary attributes for R are A, B, C, D.

For A→B, B→D, CD→AB, the right side of them are B, D, and AB. They all meet the requirement in 3NF that the right side are the primary attributes(some key).

1. (5 points) Give a minimal cover for the following set of dependencies:

F’= {A→B, B→BD, A→D, CD→AB}

Answer:

Step 1: all the attributes in the right side should be in a single format

{A→B, B→B, B→D, A→D, CD→A, CD→B}

Step 2: get rid of redundant like B→B; A→B + B→D => A→D; CD→A + A→B => CD→B

{A→B, B→D, CD→A}

Thus, the minimal cover is {A→B, B→D, CD→A}