Database Systems, CSCI 4380-01
Homework # 2 Answers
Due Monday September 21, 2020 at 11:59:59 PM

Homework Statement. This homework is worth 5% of your total grade. If you choose to skip it, Midterm #1 will be worth 5% more. Remember, practice is extremely important to do well in this class. I recommend that not only you solve this homework, but also work on homeworks from past semesters. Link to those is already provided in Teams, which I am repeating here:

`http://cs.rpi.edu/~sibel/DBS_Past_Materials/`

This homework aims to test normalization theory.

Question 1. You are given the following relation:

`EventInformation(eventname, edate, starttime, duration, URL, description, host, panelistname, panelistemail, participantid, participantname, participantemail, participantaddress, ticketprice)`

This is a relation containing information about different events. Each event can have multiple names, panelists and participants. There can be multiple events on a given date, but only one event can occur on a given `edate` and `starttime`. For such an event, there is a unique `duration, URL, description` and `host`.

Two panelists from different events can have the same `panelistname` or `panelistemail`, but for panelist in a specific event, `panelistname` is unique and given their `panelistname` for an event, their `panelistemail` is fixed. (This means that however unlikely, two participants in the same event may share an email but not name).

`participantname`, `participantemail`, `participantaddress` are not guaranteed to be unique in the database, but `participantid` is unique for the whole relation. Given a unique `participantid`, their `participantname`, `participantemail` and `participantaddress` is fixed. The `ticketprice` value is unique for a unique participant and an event as different people can be charged different amounts for the same event (in the same way Amazon charges different people different amounts for the same product!).

List all relevant functional dependencies for this relation based on the above description.

Based on your functional dependencies, check if this relation is in BCNF or in 3NF. Show your work.

Answer.

`EventInformation(eventname, edate, starttime, duration, URL, description, host, panelistname, panelistemail, participantid, participantname, participantemail, participantaddress, ticketprice)`

`edate starttime` → `duration URL description host`

`edate starttime panelistname` → `panelistemail`

`participantid` → `participantname participantemail participantaddress`

`edate starttime participantid` → `ticketprice`

Key: `eventname edate starttime panelistname participantid`

It is not in BCNF or 3NF. All functional dependencies violate both normal forms.

Question 2. You are given the following set of functional dependencies for relation $R(A, B, C, D, E, F, G)$.

$$\mathcal{F} = \{AC \rightarrow D, AC \rightarrow E, BE \rightarrow F, AFG \rightarrow B\}$$

Is the decomposition of R into $R1(A, B, C, F, G)$ and $R2(A, B, C, D, E)$ a dependency preserving decomposition?

To do this, find the projection of these functional dependencies to decomposed relations $R1$ and $R2$ below as $\mathcal{F}_1$ and $\mathcal{F}_2$. Show some details of your work.

The find if the union of these functional dependencies and check if they are equivalent to the original set F.

Answer.

$$\mathcal{F} = \{AC \rightarrow DE, BE \rightarrow F, AFG \rightarrow B\}$$

Note: When computing dependency projections, it only makes sense to consider attributes that appear in the left hand side and their combinations that are in the given relation.

$R1(A, B, C, F, G)$ $\mathcal{F}_1 = \{AFG \rightarrow B, ABC \rightarrow F\}$

$AB+ = \{A, B\}$ $AC+ = \{A, C, D, E\}$ $ABC+ = \{A, B, C, D, E, F\}$ $AFG+ = \{A, F, G, B\}$

$R2(A, B, C, D, E)$ $\mathcal{F}_2 = \{AC \rightarrow DE\}$

$BE+ = \{B, E, F\}$ $ABE+ = \{A, B, E, F\}$

$$\mathcal{F} = \{AC \rightarrow DE, BE \rightarrow F, AFG \rightarrow B\}$$

$$\mathcal{F}' = \mathcal{F}_1 \cup \mathcal{F}_2 = \{AC \rightarrow DE, AFG \rightarrow B, ABC \rightarrow F\}$$

In $\mathcal{F}'$, $BE+ = \{B, E\}$, hence $BE \rightarrow F$ is not preserved. This is not a dependency preserving decomposition.

Question 3. You are given the following set of functional dependencies for relation $R(A, B, C, D, E, F, G)$:

$$\mathcal{F} = \{AC \rightarrow BD, BC \rightarrow E, BE \rightarrow DF, AG \rightarrow EB\}$$

$R1(A, C, B, D)$
$R2(A, B, C, E, G)$
$R3(B, E, F)$
$R4(A, G, E)$

Is the following decomposition lossless? Show your work with Chase decomposition algorithm.

Answer.

A	B	C	D	E	F	G
a	b	c	d	e1	f1	g1
a	b	c	d2	e	f2	g
a3	b	c3	d3	e	f	g3
a	b4	c4	d4	e	f4	g

Apply $AC \to BD$ to rows 1 and 2:

A	B	C	D	E	F	G
a	b	c	d	e1	f1	g1
a	b	c	d	e	f2	g
a3	b	c3	d3	e	f	g3
a	b4	c4	d4	e	f4	g

Apply: $BC \to E$ to rows 1 and 2:

A	B	C	D	E	F	G
a	b	c	d	e	f1	g1
a	b	c	d	e	f2	g
a3	b	c3	d3	e	f	g3
a	b4	c4	d4	e	f4	g

Apply: $AG \to EB$ to rows 2 and 4:

A	B	C	D	E	F	G
a	b	c	d	e	f1	g1
a	b	c	d	e	f2	g
a3	b	c3	d3	e	f	g3
a	b	c4	d4	e	f4	g

Apply $BE \to DF$ to all rows

A	B	C	D	E	F	G
a	b	c	d	e	f	g1
a	b	c	d	e	f	g
a3	b	c3	d	e	f	g3
a	b	c4	d	e	f	g

Rows 2 and 4 have no subscripts, hence this decomposition is lossless.

Question 4. You are given the following set of functional dependencies for relation $R(A, B, C, D, E, F, G, H)$:

$$\mathcal{F} = \{AD \to CE, C \to D, BEF \to G, AG \to C\}$$

(a) Find keys, check if it is in 3NF or not.

(b) If it is not in 3NF, use 3NF decomposition to find relations in 3NF.

(c) For each decomposed relation, the find the functional dependencies that are projected into the relation. Check if it is in BCNF or not.

Answer.

$R(A, B, C, D, E, F, G, H)$:

$$\mathcal{F} = \{AD \to CE, C \to D, BEF \to G, AG \to C\}$$

(ABFH should be in all keys as they never appear on the right hand side of a functional dependency.)

Keys: ABDFH, ABCFH, ABEFH, ABGFH

This relation is already in 3NF because C,D,E,G are all prime attributes. All functional dependencies have a prime attribute on the right hand side.

We are done!

Question 5. Convert the following set of functional dependencies to minimal basis. Show only the main steps:

$$\mathcal{F} = \{AC \to BD, BC \to BE, ABC \to E\}$$

Answer.

Step 1:

$F = \{AC \to B, AC \to D, BC \to B, BC \to E, ABC \to E\}$

Step 2:

$F = \{AC \to B, AC \to D, BC \to E, ABC \to E\}$

Step 3:

Can we remove $ABC \to E$?

$F2 = \{AC \to B, AC \to D, BC \to E\}$

in $F2$: $ABC+ = \{A, B, C, E, D\}$, since E is in the closure, yes, we we can remove it.

Step 4:

Try to remove any attribute on the left, but cannot. Hence, this is in minimal basis:

$F = \{AC \to B, AC \to D, BC \to E\}$

Question 6. You are given the following relation and the set of functional dependencies. In this model, clubs can have multiple offiers but a person can be the offier of only one club.

Use BCNF decomposition to find a set of relations that are in BCNF.

`Clubs(clubname, datefounded, url, contactemail, memberid, membername, officername, officerposition)`

We will shorten the attributes for simplicity to:

Note: VERSION 1: This is the original set of fds in the question. I will solve it using these and then the corrected set to show both results.

`Clubs(cname, df, url, email, mid, mname, oname, oposition)`

cname $\to$ df url email
cname mid $\to$ mname

`cname oname` → `oposition`
`oname` → `cname`

Answer.

Keys: `mid oname`

All violate BCNF. I will use `cname oname` → `oposition`

`cname oname+` = cname, oname, oposition

`C1(cname, oname, oposition)`

`oname` → `cname, oposition`
Key: `oname` In BCNF

`C2(cname, df, url, email, mid, mname, oname)`

`cname` → `df url email`
`cname mid` → `mname`
`oname` → `cname`

Key: `mid oname`

All violate BCNF! I will take out `oname` → `cname`

`C21(oname, cname)`, `oname` → `cname` Key: `oname`, in BCNF

`C22(df, url, email, mid, mname, oname)`

`oname` → `df url email`
`oname mid` → `mname`

Key: `oname, mid`. The first fd violates BCNF. I will take it out!

`C221(df, url, email, oname)`

`oname` → `df url email`, key: `oname` in BCNF.

`C221(mid, mname, oname)`

`oname mid` → `mname` key: `oname, mid` in BCNF.

Final result:

```
C1(cname, oname, oposition)
C21(oname, cname)
C221(df, url, email, oname)
C221(mid, mname, oname)
```

Note that C21 is redundant given C1, so we can potentially remove it later. But this is a valid answer to this query.

VERSION 2: This is the minor change made after it was pointed out that the third fd is not minimal. Let's see how it changes the result if we do the same order of operations:

`Clubs(cname, df, url, email, mid, mname, oname, oposition)`

```
cname → df url email
cname mid → mname
oname → oposition
oname → cname
```

Answer.

Keys: `mid oname`

All violate BCNF. I will use `oname → oposition`

`oname+` = cname, oname, oposition

`C1(cname, oname, oposition)`

`oname → cname, oposition`
Key: `oname` In BCNF

`C2(df, url, email, mid, mname, oname)`

```
oname → df url email
oname mid → mname
```

Key: `oname, mid`. The first fd violates BCNF. I will take it out!

`C21(df, url, email, oname)`

`oname → df url email`, key: `oname` in BCNF.

`C22(mid, mname, oname)`

`oname mid → mname` key: `oname, mid` in BCNF.

Final result:

```
C1(cname, oname, oposition)
C21(df, url, email, oname)
C22(mid, mname, oname)
```

Basically, the same result except I did not have the one extra relation!

Now let's see if the results change if we apply the fds in a different ordering for experimentation, I will not list all steps.

`Clubs(cname, df, url, email, mid, mname, oname, oposition)`

```
cname → df url email
cname mid → mname
oname → cname, oposition
```

Keys: `mid oname`

Use `cname → df url email`

`C1(cname,df,url,email) cname → df url email`

`C2(cname, mid, mname, oname, oposition)`

`cname mid` → `mname`
`oname` → `cname, oposition`

Take out `cname mid` → `mname`

`C21(cname,mid,mname)`, `cname mid` → `mname`, Key: cname, mid, in BCNF

`C22(cname, mid, oname, oposition)`, `oname` → `cname, oposition`, key; oname, mid, not in BCNF

Take out the last fd!

`C221(cname, oname, oposition)`, `oname` → `cname, oposition`, key; oname, in BCNF

`C222(oname, mid)`, no fds, key: oname, mid, in BCNF

This is the final result, indeed it is different:

```
C1(cname,df,url,email)
C21(cname,mid,mname)
C221(cname, oname, oposition)
C222(oname, mid)
```