CS44800 Fall 21
Assignment 2 solutions

October 7, 2021

1: Functional Dependencies

Given R=(A,B,C,D,E) and the set of functional dependencies F = {AB→D, AC→E, A→B, BC→E, A→E}

1. Find the Closure of F.
   Answer: You can skip the repetitive ones.

A+ = {ABDE}	AB+ = {ABDE}	ABC+ = {ABCDE}	ABCD+ = {ABCDE}
B+ = {B}	AC+ = {ABCDE}	ABD+ = {ABDE}	ABCE+ = {ABCDE}
C+ = {C}	AD+ = {ABDE}	ABE+ = {ABDE}	ABDE+ = {ABDE}
D+ = {D}	AE+ = {ABDE}	ACD+ = {ABCDE}	ACDE+ = {ABCDE}
E+ = {E}	BC+ = {BCE}	ACE+ = {ABCDE}	BCDE+ = {BCDE}
	BD+ = {BD}	ADE+ = {ABDE}	ABCDE+ = {ABCDE}
	BE+ = {BE}	BCD+ = {BCDE}	
	CD+ = {CD}	BCE+ = {BCE}	
	CE+ = {CE}	BDE+ = {BDE}	
		CDE+ = {CDE}	

2. Are there any extraneous dependencies (dependencies that can be derived from other dependencies) in F?

   Answer: Yes. AC→ E because it can be determined by other dependencies in F.

3. Rewrite F into a canonical cover F_C

   Answer: A→BDE
   BC → E

2: Lossless Join Decomposition

Q2.1:

A decomposition is lossless if: $R_1 \cap R_2 \rightarrow R_1$ or $R_1 \cap R_2 \rightarrow R_2$.

$R_1 \cap R_2$ = AB

$(AB)^+$ = ABCDEF AB is candidate key

Therefore AB→ ABCDEF

$R_1 \cap R_2 \rightarrow R_1$ = ABC which is subset of ABCDEF and therefore $R_1 \cap R_2$ is lossless

decomposition. $R_{1,2}$ = (A,B,C,D)

$R_{1,2} \cap R_3 = D$

$(D)^+ = D$

Since $R_{1,2} \cap R_3 = D$ and neither $R_{1,2} \cap R_3 \rightarrow R_{1,2}$ or $R_{1,2} \cap R_3 \rightarrow R_3$ are satisfied.

R is not a lossless decomposition.

Q2.2

$R_1 \cap R_2 = A$

A is a candidate key since, $(A)^+$= {A, B, C, D, E}

Therefore A → ABCDE

And since ABC is a subset of ABCDE

Therefore, $R_1 \cap R_2 \rightarrow R_1$ or (A, B, C)

Therefore, this decomposition R is a lossless decomposition.

3: Normalization

Q3.1

In 3NF, there are no transitive dependencies and in BCNF, if for every non trivial functional dependency, the key for that functional dependency is a superkey for that relation. 3NF basically is a less strong version of BCNF, where you do not decompose if you would not be able to check all original FD's only in the decomposed relations.

The advantage to BCNF is that the relation ends up fully decomposed. It's the most space-efficient lossless decomposition method.

The benefit of 3NF is that it provides a way to ensure no transitive dependencies, and you can always check all functional dependencies in the decomposed relations WITHOUT joining (Dependency Preserving).

Q3.2

Relation: R=CSJDPQV

FDs: C→CSJDPQV, SD→P, JP→C,J→S

Find minimal cover: {C→J, C→D, C→Q, C→V, JP→C, J→S, SD→P}

Combine LHS: {C→JDQV, JP→C, J→S, SD→P}

New relations: CJDQV, JPC, JS, SDP

Since CJDQV is a superkey we are done!

Q3.3

We start by finding an FD which violates BCNF:

A→BC violates BCNF because it is a non-trivial FD and A is not a super key for R

Therefore, we decompose R into:

R_1 = (A, B, C)

R_2 = (A, D, E, H)

The next FD: E→HA does not violate BCNF for R_1 but does violate BCNF for R_2 so we decompose R_2 into:

R_3 = (E, H, A)

R_4 = (D, E)

We are left with:

R_1 = (A, B, C)

R_3 = (E, H, A)

R_4 = (D, E)

As our decomposition, which each are not violated by any FD so this is our complete decomposition.

1. What are candidate keys for R given the functional dependencies?

   Answer: {A,C}

4: Multivalued Dependencies

Q4.1) Why do we need multi-valued dependency if we already preserve all the functional dependencies? Please give an example (other than the one below) of a situation that we need to further decompose using multi-valued dependencies.

Answer: Some relation schemas, even though they are in BCNF, do not seem to be sufficiently normalized, in the sense that they still suffer from the problem of repetition of information. To deal with this problem, a new form of constraint is defined, called a multivalued dependency. As we did for functional dependencies, we shall use multivalued dependencies to define a normal form for relation schemas. This normal form, called fourth normal form (4NF), is more restrictive than BCNF. (Source: section 7.6 of the textbook)
Any appropriate example is accepted as a correct answer. We refer you to the section 7.6 of the textbook for an excellent example.

Q4.2.1) List functional dependencies that hold for R.

Answer: ID→ Age, Name

Q4.2.2) List multivalued dependencies that hold for R.

Answer:
ID→ MovieLiked
ID→ SnackLiked

5: Student Activities Database

There were many good answers to this question; we have provided one example, courtesy of Sai Chikyala, that is reasonably comprehensive but still reasonably concise.

5.

1. R = {ClubID: INT, ClubName: VARCHAR, ClubMember: VARCHAR, ClubTitle: VARCHAR, EventID: INT, EventName: VARCHAR, EventSponsorClub: INT, EventLocation: VARCHAR, EventDateTime: DATETIME, EventAttendee: VARCHAR, StudentEmail: VARCHAR, StudentName: VARCHAR, StudentCampusAddress: VARCHAR, StudentPrimaryMajor: VARCHAR, StudentClassYear: YEAR}
2. ClubMember and EventAttendee can be represented by StudentEmail instead. The attributes just represent a way to identify students, which can be done with student email. EventSponsoringClub can be represented by ClubID instead.
   ClubID->ClubName
   A club id can only have one corresponding club name.
   ClubID->->StudentEmail
   A club can have multiple members.
   EventID->EventName, EventLocation, EventDateTime
   An event id can only have one corresponding event name, location, and date/time.
   EventID->ClubID
   An event is sponsored by a single club.
   ClubID->->EventID
   A club can have multiple events.
   EventID->->StudentEmail
   An event can have multiple attendees.
   StudentEmail->StudentName, StudentCampusAddress, StudentPrimaryMajor, StudentClassYear
   A student email has a single student name, address, primary major, and class year associated to it.
   StudentEmail->ClubTitle
   A student can have a single title in a club. A club title for a normal member is just 'Member'.
3. BCNF
   R(ClubID, ClubName, ClubTitle, EventID, EventName, EventLocation, EventDateTime, StudentEmail, StudentName, StudentCampusAddress, StudentPrimaryMajor, StudentClassYear)

   StudentEmail->StudentName, StudentCampusAddress, StudentPrimaryMajor, StudentClassYear violates BCNF.

   R(ClubID, ClubName, ClubTitle, EventID, EventName, EventLocation, EventDateTime, StudentEmail)
   Student(Email, Name, CampusAddress, PrimaryMajor, ClassYear)
   This satisfies the constraint that you cannot decompose the student table any further.

   EventID->EventName, EventLocation, EventDateTime violates BCNF.

   R(ClubID, ClubName, ClubTitle, EventID, StudentEmail)