CS353 Spring 2019
Homework 5 Solutions

Q.1 (20 points) Consider the following relation ABC:

A	**B**	**C**	**Tuple #**
10	b1	c1	#1
10	b2	c2	#2
11	b4	c1	#3
12	b3	c4	#4
13	b1	c1	#5
14	b3	c4	#6

Given the above instance of the relation, which of the following dependencies, *may hold* in the above relation? If the dependencey cannot hold, explain why *by specifying the tuples that cause the violation*.

a. i. A → C, ii. B → C, iii. C → B, iv. B → A, v. C → A
b. Does the above relation have a potential candidate key? If it does, what is it? If not, why not?

Solution:

a.

i. A → C cannot hold because Tuples #1 and #2 agree on their A values but they do not agree on their C values.
ii. B → C may hold because only Tuples #1 and #5 agree on their B values and they also agree on their C values.
iii. C → B cannot hold because Tuples #1 and #3 agree on their C values and but they do not agree on their C values.
iv. B → A cannot hold because Tuples #1 and #5 agree on their B values but they do not agree on their A values.
v. C → A cannot hold because Tuples #1, #3, and #5 agree on their C values but they do not agree on their A values.

b.

- A alone cannot be a candidate key because Tuples #1 and #2 have the same A value.
- B alone cannot be a candidate key because Tuples #1 and #5 have the same B value.
- C alone cannot be a candidate key because Tuples #1, #3 and #5 have the same C value.
- (B, C) cannot be a candidate key because Tuples #1 and #5 have the same (B, C) value.
- (A, B) is a potential candidate key because all tuples in the current instance have unique (A, B) value.
- (A, C) is another potential candidate key because all tuples in the current instance have unique (A, C) value.

Q.2 (20 points) Consider the attribute set ABCDEFGH and the following set, *F*, of FDs:

$ABH \rightarrow C$	$F \rightarrow AD$
$A \rightarrow D$	$E \rightarrow F$
$C \rightarrow E$	$BH \rightarrow E$
$BGH \rightarrow F$	

Find the canonical (minimal) cover, *G*, for the set *F*.

Solution: (Apply the algorithm the same as covered in the lectures).

In BGH → F, G is extraneous because $(BH)^+$=ABCDEFH contains F. Hence, replace BGH → F with BH → F.

In ABH → C, A is extraneous because $(BH)^+$=ABCDEFH contains A. Hence, replace ABH → C with BH → C.
In F → AD, D is extraneous because F^+=FAD contains D (under (*F* – { F → AD}) U { F → A}; using F → A and A→ D). Hence, replace F → AD with F → A.
We combine BH → F, BH → F and BH → E to BH → CEF to make each left hand side in F_c unique.
Hence, we are left with

$BH \rightarrow CEF$	$F \rightarrow A$
$A \rightarrow D$	$E \rightarrow F$
$C \rightarrow E$	

In BH → CEF, F is extraneous because $(BH)^+$=ABCDEFH contains F (under (*F* – { BH → CEF}) U { BH → CE}). Hence, replace BH → CEF with BH → CE.
In BH → CE, E is extraneous because $(BH)^+$=ABCDEFH contains F (under (*F* – { BH → CE}) U { BH → C}). Hence, replace BH → CE with BH → C.

The final canonical cover is

$BH \rightarrow C$	$F \rightarrow A$
$A \rightarrow D$	$E \rightarrow F$
$C \rightarrow E$	

Q.3 (20 points) Consider the following relation R(A, B, C, D, E) with the following dependencies:

$AB \rightarrow C, CD \rightarrow E, DE \rightarrow B$

a) Is AB a candidate key of this relation? If not, is ABD? Explain your answer.

b) Is this relation in BCNF? Explain your answer. If not, decompose it into BCNF relations.

c) Is this relation in 3NF? Explain your answer. If not, decompose it into 3NF relations.

Solution:

a)

$AB^+ = AB$

$AB^+ = ABC$ because of $AB \rightarrow C$

Because AB doesn't determine all the attributes of the relation, it is not a candidate key. However, ABD is a candidate key since

$ABD^+ = ABD$

$ABD^+ = ABCD$ because of $AB \rightarrow C$

$ABD^+ = ABCDE$ because of $CD \rightarrow E$.

b)

Because the only candidate key is ABD, $DE \rightarrow B$ violates BCNF since the left hand side of the dependency doesn't contain a super key. $AB \rightarrow C$ violates BCNF because the left hand side of the dependency doesn't contain a super key. $CD \rightarrow EC$ violates BCNF because the left hand side of the dependency doesn't contain a super key.

The BCNF decomposition using $AB \rightarrow C$

R(A, B, C, D, E): R1(A, B, C), R2(A, B, D, E)

R1 is in BCNF because AB is a candidate key ($AB \rightarrow C$). R2 is in not BCNF since ABD is a candidate key but $DE \rightarrow B$ violates BCNF because DE is not a candidate key.

R2(A, B, D, E) is decomposed into R21 (A, D, E), R22 (D, E, B), which are both in BCNF (DE is the candidate key for R22 because $DE \rightarrow B$ and ADE is the candidate key for R21 becuase no FDs hold.

c)

$AB \rightarrow C, CD \rightarrow E, DE \rightarrow B$

Because the only candidate key is ABD, $AB \rightarrow C$ violates 3NF since the left hand side of the dependency doesn't contain a super key and the right hand side is not a part of the candidate key. $CD \rightarrow E$ violates 3NF because the left hand side of the dependency doesn't contain a super key and the right hand side is not a part of the candidate key. $DE \rightarrow B$ doesn't violate 3NF because the left hand side of the dependency doesn't contain a super key but the right hand side is a part of the candidate key ABD.

The canonical cover of F = { $AB \rightarrow C, CD \rightarrow E, DE \rightarrow B$ } is the same as the given set of FDs. There are no extraneous attributes.

The result of the for loop in the algorithm is

R1: (A, B, C), R2: (C, D, E), R3: (D, E, B)

Because none of the relations contained a candidate key (A, B, D), we add R4: (A, B, D).

None of the relations are contained in another relation schema. Hence, the final set of relations is

R1: (A, B, C), R2: (C, D, E), R3: (D, E, B), R4: (A, B, D).

Q.4 (20 points)

Consider the relation R, that hold schedules of courses and sections at a university;

R(Course_no, Sec_no, Offering_dept, Credit_hours, Course_level, Instructor_ssn, Semester, Year, Days_hours, Room_no, No_of_students).

Suppose that the following dependencies hold on R:

Course_no → {Offering_dept, Credit_hours, Course_level}

{Course_no, Sec_no, Semester, Year} → {Days_hours, Room_no, No_of_students, Instructor_ssn}

{Room_no, Days_hours, Semester, Year} → {Instructor_ssn, Course_no, Sec_no}

Try to determine which sets of attributes form keys of R. How would you normalize this relation?

Solution:

Let's make attribute names shorter.

R(C, S, O, H, L, I, M, Y, D, R, N)

C → OHL
CSMY → DRNI
RDMY → ICS

C+ → C
C+ → COHL
CSMY+ = CSMY
CSMY+ = CSOHLMY
CSMY+ = CSOHLMYDRNI

CSMY is a candidate key for the relation.

RDMY+ = RDMY
RDMY+ = ICSRDMY
RDMY+ = ICSRDMYOHL
RDMY+ = ICSRDMYOHLNI

RDMY is another candidate key for the relation.

C → OHL violates BCNF because c is not a candidate key for the relation.

We can decompose it into BCNF relations using C → OHL as follows.

R1(COHL), R2(CSIMYDRN)

R1 is in BCNF because C is a candidate key (C → OHL).

R2 has two candidate keys RDMY and CSMY. R2 is in BCNF because CSMY → DRNI and RDMY → ICS do not violate BCNF since the left hand sides are candidate keys.

Q5. (20 points) Consider the following relation for published books:

BOOK(Book_title, Author_name, Book_type, List_price, Author_affil, Publisher)

Author_affil refers to the affiliation of the author. Suppose the following dependencies exist:

Book_title → Publisher, Book_type
Book_type → List_price
Author_name → Author_affil

a) What normal form this relation is in? Explain your answer.

b) Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.

Solution:

Book_title+ = Book_title

Book_title+ = Book_title, Publisher, Book_type

Book_title+ = Book_title, Publisher, Book_type, List_price

(Book_title, Author_name) + = Book_title, Publisher, Book_type, List_price, Author_name, Author_affil

The candidate key for the relation is (Book_title, Author_name).

The relations is not in BCNF because the following dependencies violate BCNF since the LHS of dependencies do not contain key.

Book_title → Publisher, Book_type

Book_type → List_price

Author_name → Author_affil

The relation is not in 3NF either because the same dependencies violate 3NF since the LHS of dependencies do not contain key and the RHS of dependencies are not a part of the key.

A possible BCNF decomposition using Book_title → Publisher, Book_type is

BOOK1(Book_title, Publisher, Book_type)

BOOK2(Book_title, Author_name, List_price, Author_affil)

BOOK1 is in BCNF because Book_title is the candidate key (Book_title → Publisher, Book_type)

BOOK2 is not in BCNF because (Book_title, Author_name) is the candidate key and Author_name → Author_affil violates BCNF since the LHS of the dependency does not contain key and the RHS of the dependency is not a part of the key.

BOOK2 can be further decomposed into

BOOK21(Book_title, Author_name, List_price)

BOOK22(Author_name, Author_affil)

which are both in BCNF.