CS348-Homework 4

Fall 2018

Due: Tuesday November 27, 2018 11:59PM

(There will be a 10% penalty for each late calendar-day. After five calendar days, the homework will not be accepted.)

1) (15 pts) Consider the following legal instance of a relational schema *R (A, B, C):*

<u>*R:*</u>

A	B	C
a	10	T
b	20	T
c	20	F
c	30	F

a) (10 pts) Which of the following dependencies are violated by the instances of *R*?

i. A→B **Violated**

ii. B→A **Violated**

iii. C→A **Violated**

iv. AC→ B **Violated**

v. BC→A Not violated

b) (5 pts) By only observing the instance of *R* above, can you identify the functional dependencies that hold on schema R?

No, because we can only see an instance and nothing more.

2) (15 pts) Given the following Relation R and the set of Functional Dependencies (FD) S that hold on R, find a canonical cover of S. Show the steps for getting this canonical cover. R (A, B, C, D, E, G, H, I) S:

A→B

ABCD→E

ACDG→EI

EG→H

EG→I

Decomposition of ACDG → EI to ACDG → E and ACDG → I
Psuedo-transitivity property of A → B and ABCD → E results in ACD → E
The ACD → E makes the ACDG → E unnecessary and it can be removed.
The ACDG → I can also be removed, since the EG → I determines the same with less variables.
This results in the final canonical cover below.
Final Answer:

A → B
ACD → E
EG → H
EG → I

3) (20 pts) Consider the snapshot of the relation named "Lending-Schema" from your Database Design Lecture (lecture slide 7.6).

branch-name	*branch-city*	*assets*	*customer-name*	*loan-number*	*amount*
Downtown	Brooklyn	9000000	Jones	L-17	1000
Redwood	Palo Alto	2100000	Smith	L-23	2000
Perryridge	Horseneck	1700000	Hayes	L-15	1500
Downtown	Brooklyn	9000000	Jackson	L-14	1500

Now, explain in detail (with proper example using Lending-Schema) each of the following anomalies:

i) Update Anomaly.

The issue is redundancy in branch-name, branch-city, and assets. This can cause an issue if a partial update is made using one or more of the previously listed rows. For instance, if I wanted to update the amount based on the branch-name, branch-city, and/or assets there are duplicates and the wrong one may be updated. In the case of updating the amount based on the branch-name being Downtown and the assets being 9000000 then either Jones or Jackson's loan amount will change, but it is uncertain if the correct one will change.

ii) Delete Anomaly.

The issue is accidental deletion of branch-name and branch-city from the database. If you wanted to delete Hayes using the customer-name, it would also delete Perryridge completely from the database as well.

4) (15 pts) Given a relation *R* and the set of functional dependencies *S* that hold for *R* as below, *R(A, B, C, D, E)* *S:*

A→B

BC→E

ED→A

i) List all the candidate keys for R.

RHS Variables: B, E, A
NON-RHS Variables: C, D
Join both to determine candidate keys results in: (C,D,B), (C,D,E), and (C,D,A)

ii) Identify the highest normal form of R (1NF, 3NF, or BCNF).

1NF is assumed and has atomic domain.

3NF is possible and explained below:
Candidate keys: (C,D,B), (C,D,E), and (C,D,A)
Key attributes: A, B, C, D, E
Check the RHS contains only key attributes or LHS is a superkey

A → B	RHS checks out
BC → E	RHS checks out
ED → A	RHS checks out

Thus, 3NF is the highest normal form of R.

The Functional Dependency A→B is not a superkey and thus violates BCNF rules.

The highest normal form of R is **3NF**.

iii) If applicable, show the steps of your work to put R into BCNF.

Since A→B violates BCNF, split on A and B
Table1 attributes: A,B
Table1 Functional Dependencies: A → B
Table2 attributes: A,C,D,E
Table2 Functional Dependencies: AC → E, ED → A

Table2 must be split now (into table3 and table4)
Table3 attributes: A, C, E
Table3 Functional Dependencies: AC → E
Table4 attributes: A, C, D
Table4 Functional Dependencies: NONE

Table4 does not have any functional dependencies because ACD → A goes to A → A, which can be left out because it is trivial.

End result is:
Table1 attributes: A, B
Table1 Functional Dependencies: A → B
Table3 attributes: A, C, E
Table3 Functional Dependencies: AC → E

Table4 attributes: A, C, D
Table4 Functional Dependencies: NONE

iv) If applicable, show the steps of your work to put R into 3NF.

Not applicable

5) (20 pts) Given a relation *R* and the set of functional dependencies *S* that hold for *R* as below, *R (A, B, C, D, E, F)* *S:*

AB→C

AB→D

C→A

D→B

E→F

(i) Identify the candidate keys.

A possible set of candidate keys would be **(A, B, E)**

AB determines both C and D. E is never found on the right-hand side, so it must always be in the set.

Another possible set would be **(C, D, E),** since C determines A and D determines B. Also, a set could be **(A, D, E),** since D determines B and AB determines C. Lastly, **(B, C, E)** is a set of candidate keys because C determines A and AB determines D.

(ii) Find a decomposition of R into 3NF relations that is lossless-join and dependency preserving.

Candidate keys: (A,B,E), (C,D,E), (A,D,E), and (B,C,E)
Key attributes: A, B, C, D, E

F is a none key attribute and E is an attribute, thus a 3NF table can be created with:
Table1 Attributes: E, F
Table1 Functional Dependencies: E → F

The other table will have the following:
Table2 Attributes: A, B, C, D, E
Table2 Functional Dependencies: AB→C, AB→D, C→A, and D→B
This table has the attribute E, so that it can be joined with the other table.

Show the process of your work for all the above.

6) (15 pts) Given the following decomposition of the relation *R* and the set of functional dependencies *S* as below

R (A, B, C, D, E, F, G, H, I, J) *S:*

AB→C

A→DE

B→F

F→GH

D→IJ

Decomposition:

R1 (A, B, C, D, E, J)

R2 (B, F, G, H)

R3 (D, I)

Is the decomposition dependency-preserving? Justify your answer (with explanation).

Yes, the decomposition is dependency-preserving. The A, B, and C from AB → C can all be found in R1. The A, D, and E from A → DE can also all be found in R1. The B and F from B → F can all be found in R2. The F, G, and H from F → GH can be all be found in R2. Finally, by the decomposition rule we know that if D → IJ holds, then D → I and D → J will also hold.