COMP 3005: Database Management Systems (Due: Friday November 19, 2021 (11:59 PM))

Assignment #4

Instructor: Ahmed El-Roby *Name:* , *ID:*

Instructions: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- The accepted format for your submission is pdf only.
- If you use the tex file, make sure you edit line 28 to add your name and ID. Only write your solution and do not change anything else in the tex file. If you do, you will be penalized.
- No late submissions are allowed.

Q 1: (3 points)

In class, we showed that functional dependencies are transitive. That is, if $X \to Y$ and $Y \to Z$, then $X \to Z$. Assume a new proposed rule: If $X \to Y$ and $Z \to Y$, then $X \to Z$. Prove that this rule is incorrect.
Proof by counterexample:
Assume the following relation schema *(X, Y, Z)* with the following two tuples: (x_1, y_1, z_1) and (x_1, y_1, z_2). Also assume that $X \to Y$, $Z \to Y$, and $X \to Z$. However, this instance of the relation schema has two tuples with the same values for attribute X, but different values for attribute Z. So, $X \to Z$ does not hold.

Q 2: (3 points)

How can you use functional dependencies to represent the constraint that a relationship between two entity sets X and Y is one-to-many from X to Y.

$PK(Y) \to PK(X)$

Q 3: (8 points)

Consider the following relation $R = \{A, B, C, D, E\}$ and the following set of functional dependencies
$F = \{$
$A \to BC$
$CD \to E$
$B \to D$
$E \to A\}$
Compute B^+. Is R in BCNF? If not, give a lossless decomposition of R into BCNF. Show your work for all previous questions.
$B^+ = \{B, D\}$
The candidate keys are A, BC, CD, and E. See below:
$A \to BC$, so $A \to B$ and $A \to C$. Since $A \to B$ and $B \to D$, then $A \to D$
Since $A \to CD$ and $CD \to E$, then $A \to E$
Since $A \to A$, and from above $A \to BCDE$, then $A \to ABCDE$
Since $E \to A$, then $E \to ABCDE$
Since $CD \to E$, then $CD \to ABCDE$
Since $B \to D$ and $BC \to CD$, then $BC \to ABCDE$

Now, from F, $B \to D$ is non-trivial and B is not superkey. Using the algorithm discussed in class, we derive the relations $R_1(A, B, C, E)$ and $R_2(B, D)$

Q 4: (4 points)

Give a lossless, dependency-preserving decomposition into 3NF of schema R in Q3.

F already forms a canonical cover. Using the algorithm dicussed in class, R is decomposed into $R_1(A, B, C)$, $R_2(C, D, E)$, $R_3(B, D)$, and $R_4(E, A)$. No redundant relations exist.

Q 5: (4 points)

Assume the following decomposition of R in Q3: $R_1(A, B, E)$ and $R_2(C, D, E)$. Is this decomposition lossy or lossless? Why? Show your work in detail.
The decomposition is lossless because $R_1 \cap R_2 \rightarrow R_1$ and $R_1 \cap R_2 \rightarrow R_2$.

Q 6: (22 points)

Consider the following relation $R(A, B, C, D, E, G)$ and the set of functional dependencies
$F = \{$
$A \rightarrow BCD$
$BC \rightarrow DE$
$B \rightarrow D$
$D \rightarrow A\}$

Note: Show the steps for each answer.

(a) Compute B^+. (4 points)
$B \rightarrow BD$ (third dependency)
$BD \rightarrow ABD$ (fourth dependency)
$ABD \rightarrow ABCD$ (first dependency)
$ABCD \rightarrow ABCDE$ (second dependency)
$B^+ = ABCDE$

(b) Prove (using Armstrong's axioms) that AG is superkey. (4 points)
$A \rightarrow BCD$ (Given)
$A \rightarrow ABCD$ (Augmentation with A)
$BC \rightarrow DE$ (Given)
$ABCD \rightarrow ABCDE$ (Augmentation with $ABCD$)
$A \rightarrow ABCDE$ (Transitivity)
$AG \rightarrow ABCDEG$ (Augmentation with G)

(c) Compute F_c. (6 points)
D is extraneous in the first and second functional dependencies because of the third functional dependency. So, the new set of functional dependencies become:
$A \rightarrow BC$
$BC \rightarrow E$
$B \rightarrow D$
$D \rightarrow A\}$
We also note that C is extraneous in the second functional dependency because $B \rightarrow E$ can be inferred using F (look at B^+). So, the final canonical cover becomes:
$F_c =$
$A \rightarrow BC$
$B \rightarrow E$
$B \rightarrow D$
$D \rightarrow A$
Second and third functional dependencies can be also combined to be $B \rightarrow DE$. But this is optional.

(d) Give a 3NF decomposition of the given schema based on a canonical cover. (4 points)
Depending whether second and third dependencies were unioned. We could have two answers based on the algorithm discussed in class: $R_1(A, B, C)$
$R_2(B, E)$
$R_3(B, D)$
$R_4(D, A)$
OR

$R_1(A, B, C)$
$R_2(B, D, E)$
$R_3(D, A))$
Neither of the above solutions has a candidate key because G is not dependent on any attribute. So, we create another relation $R_{4OR5}(A, G)$.

(e) Give a BCNF decomposition of the given schema based on F. Use the first functional dependency as the violator of the BCNF condition. (4 points)

Starting with R, we see that it is not in BCNF because of the first functional dependency (A is not superkey). So, we decompose R based on the algorithm discussed in class into:
$R_1(A, B, C, D)$ and $R_2(A, E, G)$.
We then notice that $A \rightarrow E$ is in F^+ and causes R_2 to violate BCNF. So, we decompose R_2 further into: $R_3(A, G)$ and $R_4(A, E)$. We now see that R_1, R_3, and R_4 are all in BCNF.

Q 7: (6 points)

Given the following set of functional dependencies:
$A \rightarrow BC$
$B \rightarrow AC$
$C \rightarrow AB$
Show that it is possible to find more than one unique canonical cover for this set.
Consider the first functional dependency. We an verify that C is extraneous in $A \rightarrow BC$ and delete it. Subsequently, we can similarly check that A is extraneous in $B \rightarrow AC$ and delete it, and that B is extraneous in $C \rightarrow AB$ and delete it, resulting in a canonical cover $A \rightarrow B$, $B \rightarrow C$, and $C \rightarrow A$.
However, we an also verify that B is extraneous in $A \rightarrow BC$ and delete it. Subsequently, we an similarly check that C is extraneous in $B \rightarrow AC$ and delete it, and that A is extraneous in $C \rightarrow AB$ and delete it, resulting in a canonical cover $A \rightarrow C$, $B \rightarrow A$, and $C \rightarrow B$.

Q 8 (7 points)

Consider the schema $R = (A, B, C, D, E, G)$ and the set F of functional dependencies:
$A \rightarrow BC$
$BD \rightarrow E$
$CD \rightarrow AB$
Use the BCNF decomposition algorithm to find a BCNF decomposition of R. Start with $A \rightarrow BC$. Explain your steps. Is this decomposition lossy or lossless? Is it dependency-preserving?
First use $A \rightarrow BC$ to decompose R into $R_1 = (A, B, C)$ and$R_2 = (A, D, E, G)$. Then use the inferred $AD \rightarrow E$ to decompose R_2 into $R_3 = (A, D, E)$ and $R_4 = (A, D, G)$. The resulting decomposition is (A, B, C), (A, D, E), (A, D, G).
It is lossless since when the algorithm decomposes a schema using $\alpha \rightarrow \beta$, the attributes in α appear in both resulting schemas ($\alpha \cap \beta = \phi$) and can be used to join two schemas without introducing more tuples, since $\alpha \rightarrow \alpha\beta$.
It is not dependency preserving. It preserves $A \rightarrow BC$, $AD \rightarrow E$, but not $BD \rightarrow E$.

Q 9: (3 points)

As discussed in class, SQL does not support functional dependency constraints. But it supports materialized views. Assume that the DBMS maintains the materialized view immediately. Given a relation $R(W, X, Y, Z)$, how would you use materialized views to enforce the functional dependency $W \rightarrow Z$?

```
create materialized view V as
select distinct W, Z
from R;
alter table V add constraint V_pk primary key (W);
```