Comp115 Spring 2017, HW2 Answer Key

Problem 1 - 20%

Suppose that we have the following three tuples in a legal instance of a relation schema S with four attributes $WXYZ$ (listed in order): (1, 2, 3, 4), (5, 2, 3, 4), (6, 4, 3, 5)

1. Which of the following dependencies can you infer does **not** hold over the schema S? State your reasoning.

   (a) $W \rightarrow Y$

   We **cannot** infer that it does **not** hold over the schema S. $1 \rightarrow 3$, $5 \rightarrow 3$ and $6 \rightarrow 3$ do not violate.

   (b) $XZ \rightarrow W$

   We **can** infer that it does **not** hold over the schema S. (2,4) $\rightarrow$ 1 and (2, 4) $\rightarrow$ 5 violate it. The left hand side cannot point to different values.

   (c) $Y \rightarrow Z$

   We **can** infer that it does **not** hold over the schema S. $3 \rightarrow 4$ and $3 \rightarrow 5$ violate it. The left hand side cannot point to different values.

   (d) $WX \rightarrow Z$

   We **cannot** infer that it does **not** hold over the schema S. (1, 2) $\rightarrow$ 4, (5, 2) $\rightarrow$ 4 and (6, 4) $\rightarrow$ 5 do not violate.

2. Can you identify any dependencies that **do** hold over S?

   We cannot identify dependencies that hold over S only with any amount of data. You can always add more data that could invalidate your functional dependencies.

Problem 2 - 30%

Suppose you are given a relation R with four attributes $WXYZ$. For each of the following sets of FDs, assuming those are the only dependencies that hold for R, do the following:

(a) Identify the candidate key(s) for R

(b) Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF)

(c) If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies

1. $WX \rightarrow Y$, $WX \rightarrow Z$, $XY \rightarrow WX$

   (a) Candidate keys: WX, XY

   (b) It is in BCNF because WX and XY are superkeys for all.

   (c) No need for decomposition.

2. $X \rightarrow YZ$, $Y \rightarrow W$, $W \rightarrow X$

   (a) Candidate keys: W, X, Y

(b) It is in BCNF because W, X and W are superkeys for all.

(c) No need for decomposition.

3. $XY \rightarrow WX$, $XY \rightarrow YZ$, $Z \rightarrow X$

   (a) Candidate keys: XY

   (b) It is in 3NF. For the 3rd relation ($Z \rightarrow X$), Z is not a super key, so R cannot be in BCNF. However, X is part of the key XY, so R is in 3NF.

   (c) XYZ, XYW, ZX

4. $WX \rightarrow YZ$, $Y \rightarrow W$

   (a) Candidate keys: WX, YX

   (b) It is in 3NF. For the 2nd relation ($Y \rightarrow W$), Y is not a superkey, so R cannot be in BCNF. However, W and Y are both part of keys WX, so R is in 3NF.

   (c) YW, XYZ Dependencies lost

5. $YZ \rightarrow X$, $X \rightarrow W$, $W \rightarrow Y$

   (a) Candidate keys: YZ, XZ

   (b) It is in 2NF.

   (c) XWY, ZX Dependencies lost

Problem 3 - 50%

Consider the attribute set R = $ABCDEGH$ and the FD set

$$\text{F} = \{\ AC \rightarrow E\ ,\ AE \rightarrow C,\ BE \rightarrow D,\ CG \rightarrow A,\ G \rightarrow E,\ D \rightarrow G\ \}$$

1. For each of the attribute sets above, do the following:

   (a) Compute which dependencies hold over the set.

   (b) Identify the candidate key(s).

   (c) Decompose it into a collection of BCNF relations if it is not in BCNF.

      (i) ACE

         (a) R1 = ACE, the FDs are : AC $\rightarrow$ E, AE $\rightarrow$ C

         (b) Candidate keys : AC, AE

         (c) This is already in BCNF.

      (ii) $ABCDE$

         (a) R2 = ABCDE, the FDs are : AC $\rightarrow$ E, AE $\rightarrow$ C, BE $\rightarrow$ D

         (b) Candidate keys : ABC, ABE

         (c) This is not in BCNF. Decompose it as : ACE, BED

(iii) *ABC*

(a) R3 = ABC. No dependencies are preserved.

(b) Candidate key: ABC

(c) It is in BCNF.

(iv) *ACDEG*

(a) R4 = ACDEG, the FDs are : AC $\rightarrow$ E, AE $\rightarrow$ C, D $\rightarrow$ G, G $\rightarrow$ E, CG $\rightarrow$ A

(b) Candidate keys : AD, CD

(c) ACE, AED, DG

2. Which of the following decompositions of R = *ABCDEGH* with the same set dependencies F, is (i) dependency-preserving? (ii) lossless-join?

   (a) { *ACE, BCED, ACDG* }

      i. Yes, it is dependency preserving - each set in the decomposition contains only complete functional dependencies.

      ii. No, the intersections of the decompositions is not a superkey.

   (b) { *AC, CG, ABCD, ED* }

      i. No, there are no dependencies preserved

      ii. No, the intersections of the decompositions is not a superkey