Quiz 3: Database Systems I

Instructor: Hassan Khosravi

Spring 2012 CMPT 354

1. [10] Show that each of the following are **not** valid rules about FD's by giving a small example relations that satisfy the given FD's (following the "if") but not the FD that allegedly follows (after the "then").

   a. (3) If A->B then B->A

      If attribute A represented Social Security Number and B represented a person's name, then we would assume A->B but B->A would not be valid because there may be many people with the same name and different Social Security Numbers.

A	B
1	2
2	2

   b. (3) If AB->C and A->C, then B->C

      Let attribute A represent Social Security Number, B represent gender and C represent name. Surely Social Security Number and gender can uniquely identify a person's name (i.e. AB→C). A Social Security Number can also uniquely identify a person's name (i.e. A→C). However, gender does not uniquely determine a name (i.e. B→C is not valid)

A	B	C
1	1	1
2	1	2

   c. (4) If AB->C , then A->C or B->C

      Let attribute A represent latitude and B represent longitude. Together, both attributes can uniquely determine C, a point on the world map (i.e. AB->C). However, neither A nor B can uniquely identify a point (i.e. A->C and B->C are not valid).

A	B	C
1	1	1
2	1	2
1	2	3

2. [10] Suppose we have relation R(A,B,C,D,E), with some set of FD's, and we wish to project those FD's onto relation S(A,B,C). Give the FD's that hold in S if the FD's for R are:
    a. AB->D, AC->E, BC->D, D->A, and E->B.

We need to compute the closures of all subsets of {ABC}, although there is no need to think about the empty set or the set of all three attributes. Here are the calculations for the remaining six sets:

{A}+=A
{B}+=B
{C}+=C
{AB}+=ABD
{AC}+=ABCDE
{BC}+=ABCDE

We ignore D and E, so a basis for the resulting functional dependencies for ABC is: AC->B and BC->A.

3. [10] Let R(A,B,C,D,E)be decomposed into relations with the following three sets of attributes: {A,B,C}, {B,C,D}, and {A,C,E}. For the following sets of FD's, use the chase test to tell whether the decomposition of R is lossless. If it is not lossless, give an example of an instance of R that returns more than R when projected onto the decomposed relations and rejoined.

A->D, D->E, and B->D

This is the initial tableau:

A	**B**	**C**	**D**	**E**
a	b	c	d_1	e_1
a_1	b	c	d	e_1
a	b_1	c	d_1	e

This is the final tableau after applying FDs A→D, D→E and B→D.

A	**B**	**C**	**D**	**E**
a	b	c	d	e
a_1	b	c	d	e
a	b_1	c	d	e

Since there is an unsubscripted row, the decomposition for R is lossless for this set of FDs.

4. [10] Consider a relation Stocks(B, O, I, S, Q, D), whose attributes may be thought of informally as broker, office (of the broker), investor, stock, quantity (of the stock owned by the investor), and dividend (of the stock). Let the set of FD's for Stocks be S->D, I->B, IS->Q, and B->O.

(2) What are all the keys for Stocks

IS is the only key for the Stocks relation.

(3) Verify that the given FD's are their own minimal basis.

The first step to verify that the given FDs are their own minimal basis is to check
$\{S^+\} = \{S\}$
$\{I^+\} = \{I\}$
$\{IS^+\} = \{ISDBO\}$
$\{B^+\} = \{B^+\}$
The second step to verify that the given FDs are their own minimal basis is to check to see if any of the left sides of an FD can have one or more attributes removed without losing the dependencies. However, this is not the case for the one FD that contains two attributes on the left side.
Thus, the given set of FDs has been verified to be the minimal basis.

a. (5) Use the 3NF synthesis algorithm to find a lossless-join, dependency-preserving decomposition of R into 3NF relations. Are any of the relations not in BCNF?

Since the only key is IS, the given set of FDs has some dependencies that violate 3NF. We also know that the given set of FDs is a minimal basis. Thus the decomposed relations are SD, IB, ISQ and BO. Since the relation ISQ contains a key, we do not need to add an additional relation. The final set of decomposed relations is SD, IB, ISQ and BO.

5. [10] For each of the following relations schemas and dependencies.

   a. R(A,B,C,D,E) with MVD's A->>B and AB->>C and FD's A->D and AB->E

      Do the following:

      i. (4) Find all the 4NF violations.

         From the FDs A→D and AB→E, we can deduce that the only key is ABC. The MVDs A→→B, AB→→C and the derived MVDs A→→D and AB→→E all violate 4NF.

      ii. (6) Decompose the relations into a collection of relation schemas in 4NF.

         We must separate out the attributes of these dependencies, first decomposing into AB and ACDE. However, there is still a 4NF violation in the latter from the MVD A→→D because A is not a superkey. Thus we further decompose ACDE into relations AD and ACE. There are no more 4NF violations for the three decomposed relations so we are done. The final set of relations is AB, AD and ACE

6. [10] Use the chase test to tell whether the following dependencies hold in a relations R(A, B, C, D, E) with the dependencies A->>BC, B->D, and C->>E.

   a. (4) A->>D

      Our starting tableau is:

A	B	C	D	E
a	b_1	c_1	d	e_1
a	b	c	d_2	e

Applying MVD A→→BC we get:

A	B	C	D	E
a	b_1	c_1	d	e_1
a	b	c	d_2	e
a	b	c	d	e_1
a	b_1	c_1	d_2	e

Applying FD B→D we get:

A	B	C	D	E
a	b_1	c_1	d	e_1
a	b	c	d	e
a	b	c	d	e_1
a	b_1	c_1	d	e

Applying MVD C→→E we get:

A	B	C	D	E
a	b_1	c_1	d	e_1
a	b	c	d	e
a	b	c	d	e_1
a	b_1	c_1	d	e
a	b_1	c_1	d	e
a	b	c	d	e_1
a	b	c	d	e
a	b_1	c_1	d	e_1

Since a row of all unsubscripted attributes exists, then the MVD A→→D holds in the relation.

(6) A->E

Our starting tableau is:

A	B	C	D	E
a	b_1	c_1	d_1	e_1
a	b_2	c_2	d_2	e_2

Applying MVD A→→BC we get:

A	B	C	D	E
a	b_1	c_1	d_1	e_1
a	b_2	c_2	d_2	e_2
a	b_1	c_1	d_2	e_2
a	b_2	c_2	d_1	e_1

Applying FD B→D we get:

A	B	C	D	E
a	b_1	c_1	d_1	e_1
a	b_2	c_2	d_1	e_2
a	b_1	c_1	d_1	e_2
a	b_2	c_2	d_1	e_1

Applying MVD C→→E we get:

A	B	C	D	E
a	b_1	c_1	d_1	e_1
a	b_2	c_2	d_1	e_2
a	b_1	c_1	d_1	e_2
a	b_2	c_2	d_1	e_1
a	b_1	c_1	d_1	e_2
a	b_2	c_2	d_1	e_1
a	b_1	c_1	d_1	e_1
a	b_2	c_2	d_1	e_2

Using the chase test, it appears that the FD A→E does not hold in the relation.