Fall 2010 91.573 Database Homework 7 Solution

Due: November 26, 2010

8.1

**Answer:** A decomposition \( \{R_1, R_2\} \) is a lossless-join decomposition if \( R_1 \cap R_2 \rightarrow R_1 \) or \( R_1 \cap R_2 \rightarrow R_2 \). Let \( R_1 = (A, B, C) \), \( R_2 = (A, D, E) \), and \( R_1 \cap R_2 = A \). Since \( A \) is a candidate key (see Practice Exercise 8.6), Therefore \( R_1 \cap R_2 \rightarrow R_1 \).

8.6

**Answer:** Note: It is not reasonable to expect students to enumerate all of \( F^+ \). Some shorthand representation of the result should be acceptable as long as the nontrivial members of \( F^+ \) are found.

Starting with \( A \rightarrow BC \), we can conclude: \( A \rightarrow B \) and \( A \rightarrow C \).

Since \( A \rightarrow B \) and \( B \rightarrow D \), \( A \rightarrow D \) (decomposition, transitive)

Since \( A \rightarrow CD \) and \( CD \rightarrow E \), \( A \rightarrow E \) (union, decomposition, transitive)

Since \( A \rightarrow A \), we have \( A \rightarrow ABCDE \) from the above steps (reflexive)

Since \( E \rightarrow A \), \( E \rightarrow ABCDE \) (union)

Since \( CD \rightarrow E \), \( CD \rightarrow ABCDE \) (transitive)

Since \( B \rightarrow D \) and \( BC \rightarrow CD, BC \rightarrow ABCDE \) (augmentative, transitive)

Also, \( C \rightarrow C, D \rightarrow D, BD \rightarrow D, \) etc.

Therefore, any functional dependency with \( A, E, BC, \) or \( CD \) on the left hand side of the arrow is in \( F^+ \), no matter which other attributes appear in the FD. Allow * to represent any set of attributes in \( R \), then \( F^+ \) is \( BD \rightarrow B, BD \rightarrow D, C \rightarrow C, D \rightarrow D, BD \rightarrow BD, B \rightarrow D, B \rightarrow B, B \rightarrow BD, \) and all FDs of the form \( A^* \rightarrow \alpha, BC^* \rightarrow \alpha, CD^* \rightarrow \alpha, E^* \rightarrow \alpha \) where \( \alpha \) is any subset of \( \{A, B, C, D, E\} \). The candidate keys are \( A, BC, CD, \) and \( E \).

8.28 (use Chase Test)
Show that the following decomposition of the schema \( R \) of Practice Exercise 8.1 is not a lossless-join decomposition:

\[
(A, B, C) \\
(C, D, E).
\]

**Hint:** Give an example of a relation \( r \) on schema \( R \) such that

\[
\Pi_{A, B, C} (r) \Join \Pi_{C, D, E} (r) \neq r
\]

**Answer:** Following the hint, use the following example of \( r \):

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>( b_1 )</td>
<td>( c_1 )</td>
<td>( d_1 )</td>
<td>( e_1 )</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>( b_2 )</td>
<td>( c_1 )</td>
<td>( d_2 )</td>
<td>( e_2 )</td>
</tr>
</tbody>
</table>

With \( R_1 = (A, B, C) \), \( R_2 = (C, D, E) \):

a. \( \Pi_{R_1} (r) \) would be:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>( b_1 )</td>
<td>( c_1 )</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>( b_2 )</td>
<td>( c_1 )</td>
</tr>
</tbody>
</table>

b. \( \Pi_{R_2} (r) \) would be:

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_1 )</td>
<td>( d_1 )</td>
<td>( e_1 )</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>( d_2 )</td>
<td>( e_2 )</td>
</tr>
</tbody>
</table>

c. \( \Pi_{R_1} (r) \Join \Pi_{R_2} (r) \) would be:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>( b_1 )</td>
<td>( c_1 )</td>
<td>( d_1 )</td>
<td>( e_1 )</td>
</tr>
<tr>
<td>( a_1 )</td>
<td>( b_1 )</td>
<td>( c_1 )</td>
<td>( d_2 )</td>
<td>( e_2 )</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>( b_2 )</td>
<td>( c_1 )</td>
<td>( d_1 )</td>
<td>( e_1 )</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>( b_2 )</td>
<td>( c_1 )</td>
<td>( d_2 )</td>
<td>( e_2 )</td>
</tr>
</tbody>
</table>

Clearly, \( \Pi_{R_1} (r) \Join \Pi_{R_2} (r) \neq r \). Therefore, this is a lossy join.
a. \[ B \rightarrow BD \] (third dependency)
\[ BD \rightarrow ABD \] (fourth dependency)
\[ ABD \rightarrow ABCD \] (first dependency)
\[ ABCD \rightarrow ABCDE \] (second dependency)

Thus, \( B^+ = ABCDE \)

b. Prove (using Armstrong's axioms) that \( AF \) is a superkey.

\[ A \rightarrow BCD \] (Given)
\[ A \rightarrow ABCD \] (Augmentation with A)
\[ BC \rightarrow DE \] (Given)
\[ ABCD \rightarrow ABCDE \] (Augmentation with ABCD)
\[ A \rightarrow ABCDE \] (Transitivity)
\[ AF \rightarrow ABCDEF \] (Augmentation with F)

c. We see that \( D \) is extraneous in dep. 1 and 2, because of dep. 3.

Removing these two, we get the new set of rules

\[ A \rightarrow BC \]
\[ BC \rightarrow E \]
\[ B \rightarrow D \]
\[ D \rightarrow A \]

Now notice that \( B^+ \) is \( ABCDE \), and in particular, the FD \( B \rightarrow E \) can be determined from this set. Thus, the attribute \( C \) is extraneous in the third dependency. Removing this attribute, and combining with the third FD, we get the final canonical cover as:

\[ A \rightarrow BC \]
\[ B \rightarrow DE \]
\[ D \rightarrow A \]

Here, no attribute is extraneous in any FD.
d. We see that there is no FD in the canonical cover such that the set of attributes is a subset of any other FD in the canonical cover. Thus, each each FD gives rise to its own relation, giving
\[ r_1(A, B, C) \]
\[ r_2(B, D, E) \]
\[ r_3(D, A) \]

Now the attribute \( F \) is not dependent on any attribute. Thus, it must be a part of every superkey. Also, none of the relations in the above schema have \( F \), and hence, none of them have a superkey. Thus, we need to add a new relation with a superkey.
\[ r_4(A, F) \]

e. We start with
\[ r(A, B, C, D, E, F) \]

We see that the relation is not in BCNF because of the first FD. Hence, we decompose it accordingly to get
\[ r_1(A, B, C, D) \quad r_2(A, E, F) \]

Now we notice that \( A \rightarrow E \) is an FD in \( F^+ \), and causes \( r_2 \) to violate BCNF. Once again, decomposing \( r_2 \) gives
\[ r_1(A, B, C, D) \quad r_2(A, F) \quad r_3(A, E) \]

This schema is now in BCNF.

f. Can you get the same BCNF decomposition of \( r \) as above, using the canonical cover?
If we use the functional dependencies in the preceding canonical cover directly, we cannot get the above decomposition. However, we can infer the original dependencies from the canonical cover, and if we use those for BCNF decomposition, we would be able to get the same decomposition.
**Answer:** First we note that the dependencies given in Practice Exercise 8.1 form a canonical cover. Generating the schema from the algorithm of Figure 8.12 we get

\[ R' = \{(A, B, C), (C, D, E), (B, D), (E, A)\}. \]

Schema \((A, B, C)\) contains a candidate key. Therefore \(R'\) is a third normal form dependency-preserving lossless-join decomposition.

Note that the original schema \(R = (A, B, C, D, E)\) is already in 3NF. Thus, it was not necessary to apply the algorithm as we have done above. The single original schema is trivially a lossless join, dependency-preserving decomposition.