Normal Forms and Decompositions

Tom Kelliher, CS 318

Apr. 1, 2002

1 Administrivia

Announcements

Assignment

Read 8.7–8.

Sample end-to-end application due Friday.

Each group must demonstrate its relations are in BCNF or 3NF (or show how to decompose the relations into BCNF or 3NF — no need to implement) by the end of the semester.

From Last Time

Entailment checking.

Outline

1. Normal Forms.

2. Properties of Decompositions.

Coming Up

Synthesis/decomposition of BCNF and 3NF.

2 Normal Forms

Normal forms eliminate degrees of redundancy.

Example relation: (SSN, Name, Address, Hobby). FDs?

Example decomposed relation: (SSN, Name, Address), (SSN, Hobby).

2.1 Boyce-Codd Normal Form

A relational schema $\mathbf{R} = (\overline{R}; \mathcal{F})$ is in BCNF if for every FD $\overline{X} \to \overline{Y} \in \mathcal{F}$ either of the following is true:

1. $\overline{Y} \subseteq \overline{X}$.
2. $\overline{X}$ is a superkey of $\mathbf{R}$.

Are either of the examples in BCNF?

2.2 Third Normal Form

A relational schema $\mathbf{R} = (\overline{R}; \mathcal{F})$ is in BCNF if for every FD $\overline{X} \to A \in \mathcal{F}$ either of the following is true:

1. $A \subseteq \overline{X}$.
2. $\overline{X}$ is a superkey of $\mathbf{R}$.
3. $A \in \overline{K}$ for some key $\overline{K}$ of $\mathbf{R}$.

Which is true: all BCNF schemas are in 3NF, vice-versa, or none of the above?

3 Properties of Decompositions

1. What is a decomposition?
2. Lossless decompositions.
3. Dependency preserving decompositions.
4. Conclusions.

3.1 Definition of a Decomposition

A decomposition of $\mathbf{R} = (\overline{R}; \mathcal{F})$ is a set of schemas:

$$\mathbf{R}_1 = (\overline{R_1}; \mathcal{F}_1), \mathbf{R}_2 = (\overline{R_2}; \mathcal{F}_2), \ldots, \mathbf{R}_n = (\overline{R_n}; \mathcal{F}_n),$$

such that the following hold:

1. $\overline{R} = \cup_{i=1}^{n} \overline{R}_i$.
2. $\mathcal{F}$ entails $\mathcal{F}_i$ for all i.

The decomposition of a relation instance is defined similarly.

3.2 Lossless Decompositions

1. We need:

   $$\mathbf{r} = \mathbf{r}_1 \bowtie \mathbf{r}_2 \bowtie \cdots \bowtie \mathbf{r}_n.$$

   Why? Consider the "ultimate" redundancy eliminating "decomposition" of the example relation.

2. This is always true:

   $$\mathbf{r} \subseteq \mathbf{r}_1 \bowtie \mathbf{r}_2 \bowtie \cdots \bowtie \mathbf{r}_n.$$

   Why?

3. So we need to show:
$$\mathbf{r}_1 \bowtie \mathbf{r}_2 \bowtie \cdots \bowtie \mathbf{r}_n \subseteq \mathbf{r}.$$

4. A binary decomposition will be lossless if either of the following is true:

   (a) $(\overline{R}_1 \cap \overline{R}_2) \rightarrow \overline{R}_1 \in \mathcal{F}^+$.

   (b) $(\overline{R}_1 \cap \overline{R}_2) \rightarrow \overline{R}_2 \in \mathcal{F}^+$.

   The justification isn't that hard, but we'll skip it.

3.3 Dependency-Preserving Decompositions

1. Consider the schema HasAccount (AccountNumber, ClientId, OfficeId) with FDs:

   (a) ClientId, OfficeId $\rightarrow$ AccountNumber

   (b) AccountNumber $\rightarrow$ OfficeId

   It has been decomposed into: (AccountNumber, OfficeId) and (AccountNumber, ClientId). What about the FDs?

2. A decomposition is dependency-preserving iff
$$\mathcal{F} = \cup_{i=1}^{n} \mathcal{F}_i$$
   How do we show this?

3. Decompositions which are not dependency-preserving require extra work on updates!

4. Consider $\mathbf{R} = (\overline{R}; \mathcal{F})$ and one of the schemas of the decomposition: $\overline{R}_i$. We define:
$$\pi_{\overline{R}_i}(\mathcal{F}) = \{\overline{X} \rightarrow \overline{Y} \mid \overline{X} \rightarrow \overline{Y} \in \mathcal{F}^+ \text{ and } \overline{X} \cup \overline{Y} \subseteq \overline{R}_i\}.$$
   The idea is to use this projection to define $\mathcal{F}_i$.

5. Computing these projections is exponential in the size of $\mathcal{F}$!

3.4 Conclusions

1. All things being equal, BCNF is preferable to 3NF.

2. Not all BCNF decompositions are dependency-preserving.

   A problem in update-intensive environments.

3. When BCNF decomposition results in a dependency-preserving set of relations, use the BCNF.

   Otherwise, consider using 3NF.