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.

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 $R = (R; \mathcal{F})$ is in BCNF if for every FD $X \rightarrow Y \in \mathcal{F}$ either of the following is true:

1. $Y \subseteq X$.
2. $X$ is a superkey of $R$.

Are either of the examples in BCNF?

2.2 Third Normal Form

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

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

3.1 Definition of a Decomposition

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

$$R_1 = (\bar{R}_1; F_1), R_2 = (\bar{R}_2; F_2), \ldots, R_n = (\bar{R}_n; F_n),$$

such that the following hold:

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

The decomposition of a relation instance is defined similarly.

3.2 Lossless Decompositions

1. We need:

$$r = r_1 \bowtie r_2 \bowtie \cdots \bowtie r_n.$$ 

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

2. This is always true:

$$r \subseteq r_1 \bowtie r_2 \bowtie \cdots \bowtie r_n.$$ 

Why?
3. So we need to show: \( r_1 \bowtie r_2 \bowtie \cdots \bowtie r_n \subseteq r. \)

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 \((\text{AccountNumber, ClientId, OfficeId})\) with FDs:

   (a) \( \text{ClientId, OfficeId} \rightarrow \text{AccountNumber} \)

   (b) \( \text{AccountNumber} \rightarrow \text{OfficeId} \)

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

2. A decomposition is dependency-preserving iff

   \[ \mathcal{F} = \bigcup_{i=1}^{n} \mathcal{F}_i \]

   How do we show this?

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

4. Consider \( R = (\overline{R}; \mathcal{F}) \) and one of the schemas of the decomposition: \( \overline{R}_i \). We define:

   \[ \pi_{\overline{R}_i}(\mathcal{F}) = \{ X \rightarrow Y \mid X \rightarrow Y \in \mathcal{F}^+ \text{ and } X \cup Y \subseteq \overline{R}_i \}. \]

   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.