Third Normal Form

T. M. Murali

November 3, 2010

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester.
  2. Each course is taught either in the fall every year or in the spring every year.

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. Each course is taught either in the fall every year or in the spring every year.

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. $\mathtt{PSY} \rightarrow \mathtt{CD}$
  2. Each course is taught either in the fall every year or in the spring every year. $\mathtt{CD} \rightarrow \mathtt{S}$

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
    1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
    2. Each course is taught either in the fall every year or in the spring every year. $\texttt{CD} \rightarrow \texttt{S}$
- What are the keys?

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. Each course is taught either in the fall every year or in the spring every year. $\texttt{CD} \rightarrow \texttt{S}$
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. PSY $\rightarrow$ CD
  2. Each course is taught either in the fall every year or in the spring every year. CD $\rightarrow$ S
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$
- Decomposing using CD $\rightarrow$ S yields

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. `PSY` $\rightarrow$ `CD`
  2. Each course is taught either in the fall every year or in the spring every year. `CD` $\rightarrow$ `S`
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$
- Decomposing using `CD` $\rightarrow$ `S` yields
  `Teach1(C, D, S)` and `Teach2(C, D, P, Y)`.

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. Each course is taught either in the fall every year or in the spring every year. $\texttt{CD} \rightarrow \texttt{S}$
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$
- Decomposing using $\texttt{CD} \rightarrow \texttt{S}$ yields `Teach1(C, D, S)` and `Teach2(C, D, P, Y)`. Are both in BCNF?

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. `PSY` $\rightarrow$ `CD`
  2. Each course is taught either in the fall every year or in the spring every year. `CD` $\rightarrow$ `S`
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$
- Decomposing using `CD` $\rightarrow$ `S` yields `Teach1(C, D, S)` and `Teach2(C, D, P, Y)`. Are both in BCNF?
- How do you enforce `PSY` $\rightarrow$ `CD`?

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. `PSY` $\rightarrow$ `CD`
  2. Each course is taught either in the fall every year or in the spring every year. `CD` $\rightarrow$ `S`
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$
- Decomposing using `CD` $\rightarrow$ `S` yields `Teach1(C, D, S)` and `Teach2(C, D, P, Y)`. Are both in BCNF?
- How do you enforce `PSY` $\rightarrow$ `CD`? Only by joining `Teach1` and `Teach2`, which is expensive.

Preserving FDs in a Decomposition

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Do not assume that each course is taught by at most one professor.
- University introduces two new rules:
  1. Each professor teaches ≤ 1 course per semester. PSY $\rightarrow$ CD
  2. Each course is taught either in the fall every year or in the spring every year. CD $\rightarrow$ S
- What are the keys? $\{P, S, Y\}$ and $\{C, D, P, Y\}$
- Decomposing using CD $\rightarrow$ S yields `Teach1(C, D, S)` and `Teach2(C, D, P, Y)`. Are both in BCNF?
- How do you enforce PSY $\rightarrow$ CD? Only by joining `Teach1` and `Teach2`, which is expensive.
- The BCNF decomposition algorithm does not preserve FDs.

Third Normal Form

- A relation R is in *Third Normal Form* (3NF) if and only if for every non-trivial FD $A_1\ A_2\ \ldots\ A_n \rightarrow B$ for R, one of the following two conditions is true:
  1. $\{A_1, A_2, \ldots, A_n\}$ is a superkey for R or
  2. B is *prime*, i.e., B is an attribute in some key for R.

Third Normal Form

- A relation R is in *Third Normal Form* (3NF) if and only if for every non-trivial FD $A_1\ A_2\ \ldots\ A_n \rightarrow B$ for R, one of the following two conditions is true:
  1. $\{A_1, A_2, \ldots, A_n\}$ is a superkey for R or
  2. B is *prime*, i.e., B is an attribute in some key for R.
- `Teach(C, D, P, S, Y)` has FDs $\texttt{PSY} \rightarrow \texttt{CD}$ and $\texttt{CD} \rightarrow \texttt{S}$
- Keys are $\{P, S, Y\}$ and $\{C, D, P, Y\}$.
- $\texttt{CD} \rightarrow \texttt{S}$ violates BCNF.
- However, `Teach` is in 3NF because `S` is a part of a key.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
  1. Each professor teaches ≤ 1 course per semester. $\mathtt{PSY} \rightarrow \mathtt{CD}$
  2. In a year, each course is taught either in the fall or in the spring.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught in can change from year to year.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
    1. Each professor teaches ≤ 1 course per semester. PSY $\rightarrow$ CD
    2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught in can change from year to year. CDY $\rightarrow$ S. Keys are $\{P, S, Y\}$ and $\{C, D, P, Y\}$. Still in 3NF.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught in can change from year to year. $\texttt{CDY} \rightarrow \texttt{S}$. Keys are $\{P, S, Y\}$ and $\{C, D, P, Y\}$. Still in 3NF.
  3. Add the following constraint and repeat the analysis: Every time it is offered, each course is taught by at most one professor.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught in can change from year to year. $\texttt{CDY} \rightarrow \texttt{S}$. Keys are $\{P, S, Y\}$ and $\{C, D, P, Y\}$. Still in 3NF.
  3. Add the following constraint and repeat the analysis: Every time it is offered, each course is taught by at most one professor. $CDYS \rightarrow P$ (new), $PSY \rightarrow CD$ (old), and $CDY \rightarrow S$ (old). Keys are $\{P, S, Y\}$ and $\{C, D, Y\}$. Still in 3NF.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
  1. Each professor teaches ≤ 1 course per semester. `PSY` $\rightarrow$ `CD`
  2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught in can change from year to year. `CDY` $\rightarrow$ `S`. Keys are $\{P, S, Y\}$ and $\{C, D, P, Y\}$. Still in 3NF.
  3. Add the following constraint and repeat the analysis: Every time it is offered, each course is taught by at most one professor. $CDYS \rightarrow P$ (new), $PSY \rightarrow CD$ (old), and $CDY \rightarrow S$ (old). Keys are $\{P, S, Y\}$ and $\{C, D, Y\}$. Still in 3NF.
  4. Modify the previous constraint: Over all offerings, each course is taught by at most one professor.

More 3NF Examples

- Consider the relation `Teach(CourseNumber, DepartmentName, Professor, Semester, Year)`
- The relation models which courses a professor teaches in which semester.
- Modify the second rule and repeat the analysis:
  1. Each professor teaches ≤ 1 course per semester. $\texttt{PSY} \rightarrow \texttt{CD}$
  2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught in can change from year to year. $\texttt{CDY} \rightarrow \texttt{S}$. Keys are $\{P, S, Y\}$ and $\{C, D, P, Y\}$. Still in 3NF.
  3. Add the following constraint and repeat the analysis: Every time it is offered, each course is taught by at most one professor. $CDYS \rightarrow P$ (new), $PSY \rightarrow CD$ (old), and $CDY \rightarrow S$ (old). Keys are $\{P, S, Y\}$ and $\{C, D, Y\}$. Still in 3NF.
  4. Modify the previous constraint: Over all offerings, each course is taught by at most one professor. $CD \rightarrow P$ (new), $PSY \rightarrow CD$ (old), and $CDY \rightarrow S$ (old). Keys are $\{P, S, Y\}$ and $\{C, D, Y\}$. Still in 3NF.

Decomposition into 3NF

- We can always decompose a relational schema R into a set $\mathcal{S}$ of schemas that are *dependency-preserving*, i.e.,
  - each relation in $\mathcal{S}$ is in 3NF,
  - the decomposition of R into $\mathcal{S}$ is lossless-join,
  - the decomposition into $\mathcal{S}$ is *dependency-preserving*, i.e., for each FD that holds in R, there is a relation in $\mathcal{S}$ that allows that FD to be checked.

Decomposition into 3NF

- We can always decompose a relational schema R into a set $\mathcal{S}$ of schemas that are *dependency-preserving*, i.e.,
  - each relation in $\mathcal{S}$ is in 3NF,
  - the decomposition of R into $\mathcal{S}$ is lossless-join,
  - the decomposition into $\mathcal{S}$ is *dependency-preserving*, i.e., for each FD that holds in R, there is a relation in $\mathcal{S}$ that allows that FD to be checked.
- However, the relations may not be in BCNF and may contain some redundancy.

3NF Synthesis Algorithm

- Let $\mathcal{F}$ be the set of all FDs of R.
- We will compute a lossless-join, dependency-preserving decomposition of R into $\mathcal{S}$, where every relation in $\mathcal{S}$ is in 3NF.

1. Find a minimal basis for $\mathcal{F}$, say $\mathcal{G}$.
2. For every FD $X \rightarrow A$ in $\mathcal{G}$, use $X \cup A$ as the schema for one of the relations in $\mathcal{S}$.
3. If the attributes in none of the relations in $\mathcal{S}$ form a superkey for R, add another relation to $\mathcal{S}$ whose schema is a key for R.

3NF Synthesis Algorithm

- Let $\mathcal{F}$ be the set of all FDs of R.
- We will compute a lossless-join, dependency-preserving decomposition of R into $\mathcal{S}$, where every relation in $\mathcal{S}$ is in 3NF.

1. Find a minimal basis for $\mathcal{F}$, say $\mathcal{G}$.
2. For every FD $X \rightarrow A$ in $\mathcal{G}$, use $X \cup A$ as the schema for one of the relations in $\mathcal{S}$.
3. If the attributes in none of the relations in $\mathcal{S}$ form a superkey for R, add another relation to $\mathcal{S}$ whose schema is a key for R.

- Read Chapter 3.5.3 of the textbook to understand why this algorithm works.

3NF Synthesis Algorithm

- Let $\mathcal{F}$ be the set of all FDs of R.
- We will compute a lossless-join, dependency-preserving decomposition of R into $\mathcal{S}$, where every relation in $\mathcal{S}$ is in 3NF.

1. Find a minimal basis for $\mathcal{F}$, say $\mathcal{G}$.
2. For every FD $X \rightarrow A$ in $\mathcal{G}$, use $X \cup A$ as the schema for one of the relations in $\mathcal{S}$.
3. If the attributes in none of the relations in $\mathcal{S}$ form a superkey for R, add another relation to $\mathcal{S}$ whose schema is a key for R.

- Read Chapter 3.5.3 of the textbook to understand why this algorithm works.

Computing a Minimal Basis

- See step 3 of Algorithm 3.12 on page 82 of your textbook.
- Start with a set $\mathcal{F}$ of FDs and compute a mimimal basis $\mathcal{G}$.

Computing a Minimal Basis

- See step 3 of Algorithm 3.12 on page 82 of your textbook.
- Start with a set $\mathcal{F}$ of FDs and compute a mimimal basis $\mathcal{G}$.

1. If there is an FD D in $\mathcal{F}$ that follows from the other FDs in $\mathcal{F}$, remove D from $\mathcal{F}$.
2. Let $Y \rightarrow B$ be an FD in $\mathcal{F}$ with at least two attributes in Y and let Z be Y with one of its attributes removed. If $Z \rightarrow B$ follows from the FDs in $\mathcal{F}$, replace $Y \rightarrow B$ by $Z \rightarrow B$.
3. Repeat the first two steps until no more changes can be made to $\mathcal{F}$.

Example of 3NF Synthesis Algorithm

- (Handout 3, Problem 1, part 10) Apply the 3NF synthesis algorithm to the `Inventory` relation. Are all the relations in BCNF?

Example of 3NF Synthesis Algorithm

- (Handout 3, Problem 1, part 10) Apply the 3NF synthesis algorithm to the `Inventory` relation. Are all the relations in BCNF?
- (Handout 3, Problem 2, part 11) Apply the 3NF synthesis algorithm to the `Concerts` relation. Are all the resulting relations in BCNF?

Example of 3NF Synthesis Algorithm

- (Handout 3, Problem 1, part 10) Apply the 3NF synthesis algorithm to the `Inventory` relation. Are all the relations in BCNF?
- (Handout 3, Problem 2, part 11) Apply the 3NF synthesis algorithm to the `Concerts` relation. Are all the resulting relations in BCNF? Resulting schema is exactly the one we get by applying the BCNF normalisation algorithm to the FD `City Year` $\rightarrow$ `Venue Month Date`.

Example of 3NF Synthesis Algorithm

- (Handout 3, Problem 1, part 10) Apply the 3NF synthesis algorithm to the `Inventory` relation. Are all the relations in BCNF?
- (Handout 3, Problem 2, part 11) Apply the 3NF synthesis algorithm to the `Concerts` relation. Are all the resulting relations in BCNF? Resulting schema is exactly the one we get by applying the BCNF normalisation algorithm to the FD `City Year` → `Venue Month Date`.
- Apply BCNF normalisation algorithm to `Concerts2`. Are the resulting relations in 3NF? Is the decomposition dependency-preserving?

Example of 3NF Synthesis Algorithm

- (Handout 3, Problem 1, part 10) Apply the 3NF synthesis algorithm to the `Inventory` relation. Are all the relations in BCNF?
- (Handout 3, Problem 2, part 11) Apply the 3NF synthesis algorithm to the `Concerts` relation. Are all the resulting relations in BCNF? Resulting schema is exactly the one we get by applying the BCNF normalisation algorithm to the FD `City Year` → `Venue Month Date`.
- Apply BCNF normalisation algorithm to `Concerts2`. Are the resulting relations in 3NF? Is the decomposition dependency-preserving?
- Suppose we apply the BCNF normalisation algorithm to `Concerts` using the FD `Song` → `Album` and perform no more decompositions.

Example of 3NF Synthesis Algorithm

- (Handout 3, Problem 1, part 10) Apply the 3NF synthesis algorithm to the `Inventory` relation. Are all the relations in BCNF?
- (Handout 3, Problem 2, part 11) Apply the 3NF synthesis algorithm to the `Concerts` relation. Are all the resulting relations in BCNF? Resulting schema is exactly the one we get by applying the BCNF normalisation algorithm to the FD `City Year` → `Venue Month Date`.
- Apply BCNF normalisation algorithm to `Concerts2`. Are the resulting relations in 3NF? Is the decomposition dependency-preserving?
- Suppose we apply the BCNF normalisation algorithm to `Concerts` using the FD `Song` → `Album` and perform no more decompositions. Resulting relations have schemas {`Song, Album`} and {`Song, City, Venue, Year, Month, Date`}. This decomposition is in 3NF but is not dependency-preserving.