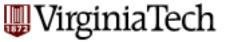


CS 4604: Introduction to Database Management Systems

B. Aditya Prakash

Lecture #16: 3NF

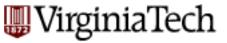


Preserving FDs in a Decomposition

- 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
 - Each professor teaches <= 1 course per semester
 Professor Semester Year → CourseNumber DepartmentName
 - Each course is taught either in the fall every year or in the spring every year

CourseNumber DepartmentName → Semester

- What are the keys?
 - {Professor, Semester, Year}
 - {CourseNumber, DepartmentName, Professor, Year}



Preserving FDs in a Decomposition

- Decomposing using
- CourseNumber DepartmentName -> Semester
 - Teach1 (CourseNumber, DepartmentName, Semester)
 - Teach2 (CourseNumber, DepartmentName, Professor, Year)
- Are both in BCNF?
- How do you enforce
- Professor Semester Year → CourseNumber DepartmentName?
 - Only by joining Teach1 and Teach2, which is expensive
- So BCNF is not necessarily dependency preserving!



"Elegant" Workaround

■ Let's define the problem away ©



Third Normal Form

- A relation R is in Third Normal Form (3NF) iff for every non-trivial FD A1 A2..An → B for R, one of the following two conditions is true:
 - A1 A2 ...An is a superkey for R
 - B is **prime** i.e. B is an attribute in some key for R

:::::::::BCNF

- Note B should be in a key not a superkey
- NP-Complete to test if a relation is in 3NF



Third Normal Form

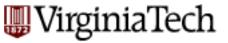
- A relation R is in Third Normal Form (3NF) iff for every non-trivial FD A1 A2..An → B for R, one of the following two conditions is true:
 - A1 A2 ...An is a superkey for R
 - B is prime i.e. B is an attribute in some key for R

- What happened to the first two? ©
 - They were defined, but not very useful today



Third Normal Form

- Teach(C, D, P, S, Y) has FDs
 - $-PSY \rightarrow CD$
 - $-CD \rightarrow S$
- Keys are {P, S, Y} and {C, D, P, Y}
- CD → S violates BCNF
- However, Teach is in 3NF because S is part of a key



More 3NF Examples

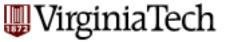
- Consider the relation Teach(CourseNumber, DepartmentName, Professor, Semester, Year)
- The relation models which courses a professor teaches in which semester
- 1. Each professor teaches <= 1 course per semester.

 $PSY \rightarrow CD$

2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught can change from year to year.

 $CDY \rightarrow S$

- Keys?
 - {P, S, Y} and {C, D, P, Y}
- In 3NF?
 - Yes



More 3NF Examples

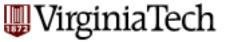
- Consider the relation Teach(CourseNumber, DepartmentName, Professor, Semester, Year)
- The relation models which courses a professor teaches in which semester
- 1. Each professor teaches <= 1 course per semester.

$$PSY \rightarrow CD$$

2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught can change from year to year.

$$CDY \rightarrow S$$

- 3. Every time it is offered, each course is taught by at most one professor CDYS → P
- Keys?
 - {P, S, Y} and {C, D, Y, P} and {C, D, Y, S}
- In 3NF?
 - Yes



More 3NF Examples

- Consider the relation Teach(CourseNumber, DepartmentName, Professor, Semester, Year)
- The relation models which courses a professor teaches in which semester
- 1. Each professor teaches <= 1 course per semester.

$$PSY \rightarrow CD$$

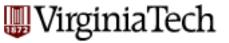
2. In a year, each course is taught either in the fall or in the spring. The semester a course is taught can change from year to year.

$$CDY \rightarrow S$$

3. Over all offerings, each course is taught by at most one professor.

$$CD \rightarrow P$$

- Keys?
 - {P, S, Y} and {C, D, Y}
- In 3NF?
 - Still Yes!



Decomposition into 3NF

- We can always decompose a relational schema R into a set S of schemas that are dependencypreserving, i.e.
 - each relation in S is in 3NF
 - the decomposition of R into S is lossless-join
 - the decomposition into S is dependency-preserving,
 i.e., for each FD that holds in R, there is a relation in S that allows that FD to be checked
- Then why bother with BCNF?
 - Unfortunately, can't guarantee no anomalies above!



3NF Synthesis Algorithm

- Let F be the set of all FDs of R
- We will compute a lossless-join, dependencypreserving decomposition of R into S, where every relation in S is in 3NF
- 1. Find a minimal basis for F, say G
- 2. For every FD X → A in G, use X U A as the schema for one of the relations in S
- 3. If the attributes in none of the relations in S form a superkey for R, add another relation to S whose schema is a key for R



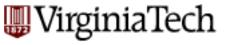
Computing a Minimal Basis

- See step 3 of Algorithm 3.12 on page 82 of your textbook
- Start with a set F of FDs and compute a minimal basisG
- 1. If there is an FD D in F that follows from the other FDs in F, remove D from F
- Let Y→B be an FD in F with at least two attributes in Y and let Z be Y with one of its attributes removed. If Z→B follows from the FDs in F, replace Y→B by Z→B
- Repeat the first two steps until no more changes can be made to F

3NF Synthesis Algorithm

Let F be the set of all FDs of R

- Surprisingly Polynomial!
- We will compute a lossless-join, dependencypreserving decomposition of R into S, where every relation in S is in 3NF
- 1. Find a minimal basis for F, say G
- 2. For every FD X → A in G, use X U A as the schema for one of the relations in S
- 3. If the attributes in none of the relations in S form a superkey for R, add another relation to S whose schema is a key for R



3NF Synthesis Algorithm

Correctness? (Tricky Proof)

- Let F be the set of all FDs of R
- We will compute a lossless-join, dependencypreserving decomposition of R into S, where every relation in S is in 3NF
- 1. Find a minimal basis for F, say G
- 2. For every FD X → A in G, use X U A as the schema for one of the relations in S
- 3. If the attributes in none of the relations in S form a superkey for R, add another relation to S whose schema is a key for R



Example

Example:

$$R(A, B, C)$$

F: {A \rightarrow B, C \rightarrow B}

• Q1: what is the cover?

• Q2: what is the decomposition to 3NF?



Example

Example:

```
R(A, B, C)
F: {A\rightarrowB, C\rightarrowB }
```

• Q1: what is the cover?

A1: 'F' is the cover

Q2: what is the decomposition to 3NF?

A2: R1(A,B), R2(C,B), ... [is it lossless??]



Example

Example:

```
R(A, B, C)
F: {A\rightarrowB, C\rightarrowB }
```

• Q1: what is the cover?

A1: 'F' is the cover

• Q2: what is the decomposition to 3NF?

A2: R1(A,B), R2(C,B), R3(A,C)



Next Lecture

- Multivalued Dependencies
- 4NF