COL 362 & COL 632

Relational - Normalization

17 Jan 2023

Course Project

- Teams of up-to 3 members
  - Deadline for team formation: ~~21st January 2023.~~ 22nd January 2023
  - Form link will be posted on Piazza
  - FAQ on the website, and timelines for each project
- Two kinds:
  - **Database backed web-application**
    - Choose your own domain for application (some ideas will be given)
    - Collect data, design the schema, store and manage the database
    - Develop a “cool” web application – Django, Flask, RoR, …
    - Define query workloads, demo
  - **Database management system internals**

Course Project

- Teams of up-to 3 members
  - Deadline for team formation: 21st January 2023.
  - Form link will be posted on Piazza
  - FAQ on the website, and timelines for each project
- Two kinds:
  - **Database backed web-application**
  - **Database management system internals**
    - Implement a novel functionality within PostgreSQL 8.3.23
    - It should have end-to-end integration (SQL front end, query plan / processing, optimizations, transaction support)
    - Collect data, develop a demo (need not be an application)
    - Present clear metrics for success – speed? quality? model theory?

Armstrong's Axioms

- Reflexivity

  If B is a subset of A, then A → B

- Augmentation

  If A → B, then AC → BC

- Transitivity

  If A → B and B → C, then A → C

Additionally…

- Union Rule
  If $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds

- Decomposition Rule
  If $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds

- Pseudotransitivity rule
  If $\alpha \rightarrow \beta$ holds and $\gamma\beta \rightarrow \delta$ holds, then $\alpha\gamma \rightarrow \delta$ holds

HW

Closure of FDs

Minimal Basis

- **Given**: *F*, the set of FDs
- **Output**: F^+, the ***closure* of F**, containing all FDs derivable from F

- Example: R (A, B, C)

AB → C

C → ED

Basis (i.e., F)

Is this set F^+

AB → ED

AB → E

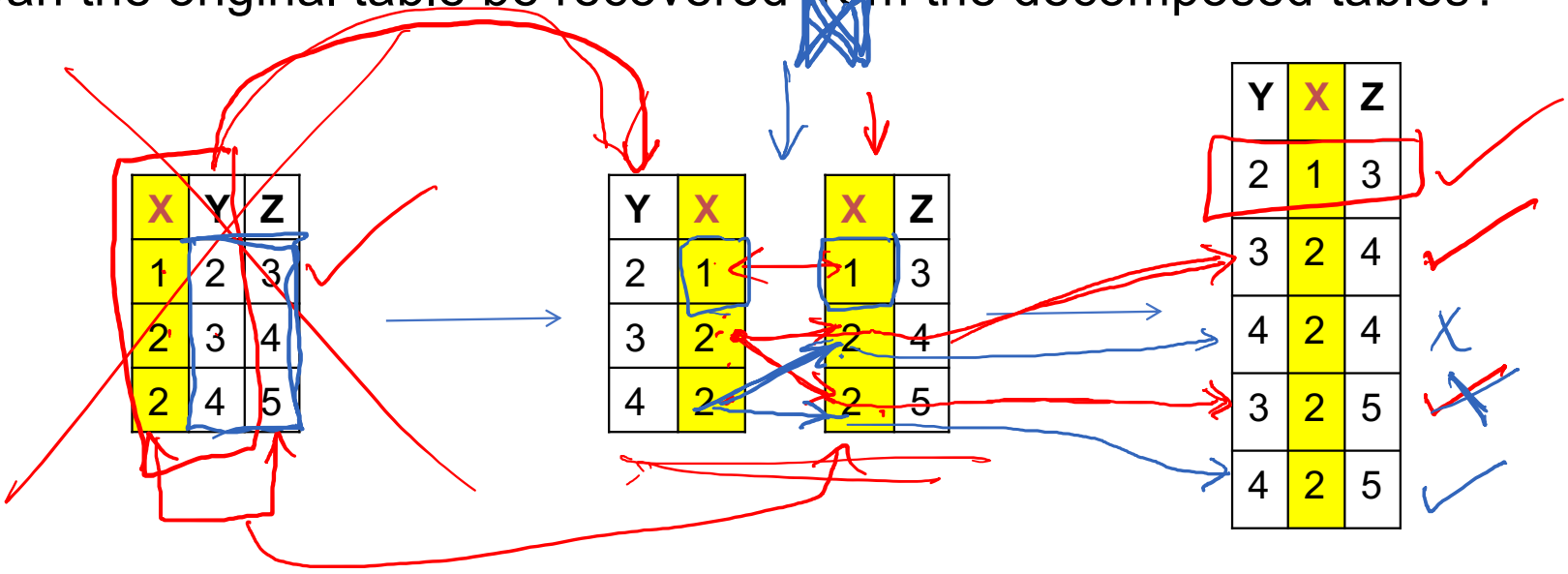
ABC → ED

C → E

C → D

Relation decomposition

- Breaking up a relation into two or more
- Tuples are ***projected***
- **Lossy** and **lossless** decomposition
  - Can the original table be recovered from the decomposed tables?



Lossless Decomposition

- We can use functional dependencies to show when certain decomposition are lossless.
- For the case of $R = (R_1, R_2)$, we require that for all possible relations r on schema R

$$r = \Pi_{R1}(r) \bowtie \Pi_{R2}(r)$$

- A decomposition of R into R_1 and R_2 is **lossless decomposition** if **at least one** of the following dependencies is in F^+:
  - $R_1 \cap R_2 \rightarrow R_1$
  - $R_1 \cap R_2 \rightarrow R_2$
- The above functional dependencies are a **sufficient** condition for lossless join decomposition; the dependencies are a **necessary** condition only if all constraints are functional dependencies

Dependency Preservation

- Testing functional dependency constraints each time the database is updated can be costly
- It is useful to design the database in a way that constraints can be tested efficiently.
- If testing a functional dependency can be done by considering **just one relation**, then the cost of testing this constraint is low
- When decomposing a relation, it is possible that it is no longer possible to do the testing without having to perform a Cartesian Product.
- A decomposition that makes it computationally hard to enforce functional dependency is said to be **NOT dependency preserving**.

Normalization – Goals

- Let R be a relation scheme with a set F of functional dependencies.
- Decide whether a relation scheme R is in “good” form.
- In the case that a relation scheme R is **not in “good” form**, need to decompose it into a set of relation scheme {R1, R2, ..., Rn} such that:
  - Each relation scheme is in good form
  - The decomposition is a lossless decomposition
  - Preferably, the decomposition should be dependency preserving.

First normal form

- 1NF (First normal form)
  - A relation is in 1NF iff every tuple contains an atomic value for each attribute
  - Follows directly from definition of relation
  - Relation contains a key

Second normal form (1/2)

No non-prime attribute in the table is functionally dependent on a proper subset of any candidate key

Name DOB MTitle Year → Address

Name	DOB	Address	MTitle	Year	Language
Priyanka Chopra	1992	Mumbai	Don	2006	Hindi
Priyanka Chopra	1992	Mumbai	Don II	2011	Hindi
Tom Cruise	1962	LA	MI-IV	2011	English
Anthony Hopkins	1937	LA	Thor: Ragnarok	2017	English
Bill Nighy	1949	LA	Valkyrie	2008	English

What are we missing here?

Name	DOB	Address
Priyanka Chopra	1992	Mumbai
Anthony Hopkins	1937	LA
Bill Nighy	1949	LA
Tom Cruise	1962	LA

MTitle	Year	Language
Don	2006	Hindi
Don II	2011	Hindi
MI-IV	2011	English
Valkyrie	2008	English
Thor: Ragnarok	2017	English

Second normal form (2/2)

Name	DOB	Address	MTitle	Year	Language
Priyanka Chopra	1992	Mumbai	Don	2006	Hindi
Priyanka Chopra	1992	Mumbai	Don II	2011	Hindi
Anthony Hopkins	1937	LA	MI-IV	2011	English
Anthony Hopkins	1937	LA	Valkyrie	2017	English
Bill Nighy	1949	LA	Valkyrie	2008	English

No non-prime attribute in the table is functionally dependent on a proper subset of any candidate key

ID	Name	DOB	Address
1	Priyanka Chopra	1992	Mumbai
2	Anthony Hopkins	1937	LA
3	Bill Nighy	1949	LA
4	Tom Cruise	1962	LA

AID	MID
1	1
1	2
2	3
3	4
4	5
4	3

ID	MTitle	Year	Language
1	Don	2006	Hindi
2	Don II	2011	Hindi
3	MI-IV	2011	English
4	Valkyrie	2008	English
5	Thor: Ragnarok	2017	English

Third normal form

- 3NF (third normal form)
  - For a non-trivial FD X → Y,
    X is a superkey or *Y is prime*

Name	**DOB**	**Address**	**Country**
Priyanka Chopra	1992	Mumbai	India
Anthony Hopkins	1937	LA	USA
Bill Nighy	1949	LA	USA

Address → Country
Name, DOB → Address
Name, DOB → Country

Name	**DOB**	**AID**
Priyanka Chopra	1992	1
Anthony Hopkins	1937	2
Bill Nighy	1949	2

ID	**Address**	**Country**
1	Mumbai	India
2	LA	USA

Boyce-Codd normal form

For all **non-trivial** FDs X → Y in F^+

X is a superkey

Addresses the following additional scenarios:

- Multiple candidate keys with intersecting elements
- All attributes are part of some key

Title	Theatre	City

Keys: Title, City
Theatre, Title
FDs:
Theatre → City
Title, City → Theatre
Theatre, Title → City

Lossless decomposition

- Algorithm:
  - If X → Y is a BCNF violation, then form two relations:
    with attributes from X U Y
    with attributes from X U (*all-X-Y)*