Normal Forms
Normal Forms

Goal

- Formal definition of a good database design

First Normal Form (1NF)

- A relational schema is in 1NF, if all attributes consist of atomic values which are not further decomposable
- e.g. STRING, INTEGER, but not structs, records, objects

- Example:
  Literature(Book, Author, Keyword)

Notice

- Pros and cons of nested relations which violate 1NF (NonFirstNormalForm = NF2) are discussed within the database community
- Nested Tables are part of the SQL:2003 Standard
Normal Forms (2)

Second Normal Form (2NF)

- A relational schema is in 2NF, if
  - each attribute is either part of a key candidate
    or is fully functional dependent from each key candidate
  - or
  - all non-key-candidates are dependable from the full primary key

Example

- GradedCertificate(S#, C#, Title, Lecturer, Room, Mark)
- Tuple \((s, c, t, d, r, m)\) means:
  student \(s\) got mark \(m\) in the course with number \(c\) and title \(t\) which was given in room \(r\) from lecturer \(d\)
- Functional Dependencies
  - \(\{S\#, C\#\} \rightarrow \{Mark\}\)
  - \(\{C\#\} \rightarrow \{Title\}\)
  - \(\{C\#\} \rightarrow \{Lecturer\}\)
  - \(\{Lecturer\} \rightarrow \{Room\}\)

Partly functional dependent from key candidate

Key candidate

- C#
- S#

Mark

Title

Lecturer

Room
Anomalies within GradedCertificate

- Information about a course are only available if there is a student enrolled to this course
- Lecturer is only within the database if he/she holds a course
- Changes on course names are costly (one update per student)
- If all students are leaving a course and the respective tuples are deleted, all the information about this course will be gone

Transformation into 2NF solves this anomalies

- Decomposition of GradedCertificate into two relations
- Schema in 2NF (“copy down the key“):
  - GradedCertificate(S#, C#, Mark)
  - Course(C#, Title, Lecturer, Room)

- Note: Partial functional dependencies of attributes which are not part of a key candidate are removed
Third Normal Form (3NF)

- A relational schema is in 3NF, if for all dependencies \( X \rightarrow A \) with \( X \subseteq R \), \( A \in R \), \( A \notin X \) following holds:
  - \( X \) contains a key of \( R \) or \( A \) is part of a key candidate
- The 3NF removes dependencies from non-key-attributes

Example

- Relation Course is not in 3NF, because of the dependency Lecturer \( \rightarrow \) Room and Lecturer is neither key nor Room part of a key candidate
- Following anomalies could occur
  - Lecturer and room are not available without a course
  - If a lecturer has no course all information about him and his room are deleted
- Schema in 3NF
  - GradedCertificate\( (S\#, C\#, \text{Mark}) \)
  - Course\( (C\#, \text{Title}, \text{Lecturer}) \)
  - Lecturer\( (\text{Lecturer}, \text{Room}) \)
Goals of the Synthesis Algorithm

- Decomposition of a relational schema $R$ with functional dependencies $F$ in relational schemas $R_1, \ldots, R_n$, so that following conditions are met:
  - No information loss
  - Preservation of functional dependencies
  - Relational schemas $R_1, \ldots, R_n$ meet the 3NF

Decomposition Process

1. Determine the canonical cover $F_C$ for $F$ (Left-Reduction, Right-Reduction, ...)
2. For each FD $(A \rightarrow B) \in F_C$
   - Generate a relational schema $R_A := A \cup B$ and
   - assign $R_A$ the FDs $F_A = \{C \rightarrow D \in F_C | C \cup D \subseteq R_A\}$
3. If none of the schemas generated in step 2 contain a key candidate of the original schema $R$ then create a new relation with schema $R_K = K$ and $F_K = \emptyset$, where $K$ is a candidate of $R$.
4. Remove the schemas $R_A$ that are part of another schema.
Example

- Relational schema ProfessorAddr with PersNo, Room, Rank, Name, Street, City, State, Government, ZipCode, AreaCode

- With following assumptions
  - City is main residence of a professor
  - City names are unique within a state
  - Zipcode is the same for the whole street
  - Cities and streets are located exactly in one state
  - One professor has one office (no shared offices)
Step 1 (Determine the canonical cover)

- **FD$_1$** \( \{\text{PersNo}\} \rightarrow \{\text{Room, Name, Rank, Street, Location, State}\} \)
- **FD$_2$** \( \{\text{Room}\} \rightarrow \{\text{PersNo}\} \)
- **FD$_3$** \( \{\text{Street, City, State}\} \rightarrow \{\text{ZipCode}\} \)
- **FD$_4$** \( \{\text{City, State}\} \rightarrow \{\text{AreaCode}\} \)
- **FD$_5$** \( \{\text{State}\} \rightarrow \{\text{Government}\} \)
- **FD$_6$** \( \{\text{ZipCode}\} \rightarrow \{\text{City, State}\} \)

Step 2

- From **FD$_1$** follows
  - **Professors** \( \{\text{PersNo, Name, Rank, Room, Street, City, State}\} \)
  - **FD$_1$** and **FD$_2$** are assigned
- From **FD$_2$** follows
  - **Rooms** \( \{\text{Room, PersNo}\} \)
- From **FD$_3$** follows
  - **Streets** \( \{\text{Street, City, State, ZipCode}\} \)
  - **FD$_3$** and **FD$_6$** are assigned
Step 2 (cont’d)

- From FD₄ follows
  - Cities:{City, State, AreaCode}
  - FD₄ is assigned
- From FD₅ follows
  - Governments:{State, Government}
  - FD₅ is assigned
- From FD₆ follows
  - ZipCodes:{ZipCode, City, State}
  - FD₆ is assigned

Step 3

- {PersNo} is key candidate of the original schema and is already part of relation professors

Step 4

- Remove ZipCodes, which is part of Streets
- Remove Rooms, which is part of Professors
Boyce-Codd Normal Form (BCNF)

- A relational schema is in BCNF if for all dependencies $X \rightarrow A$ with $X \subseteq R$, $A \in R$, and $A \notin X$ the following holds:
  - $X$ contains a key from $R$
- BCNF remove all dependencies between attributes which are part of a key candidate, i.e. each determinant is a key candidate!

Example

There are following key candidates: (City, State) and (City, Premier)
- Relation is in 3NF, since it contains no dependency from non-key attributes,
- But is not in BCNF, since State or Premier are only part of a key, not a key themselves
- Anomalies could occur since information about who governs which state are stored more than once
Generation of BCNF

Keep in mind

- Given a relational schema RS and a set of functional dependencies FD, then following holds:
  - There exists a decomposition RS into RS₁,…,RSₙ, which is lossless and with RSᵢ that fulfill the BCNF
  - But: there is not always dependency preserving decomposition

Generation of BCNF

- Decomposition from R into R₁ and R₂
- Generate a list of all determinants (left sides)
- For each determinant which is no key candidate generate a new relation R₁ from the functional dependency (X → A becomes R₁ := X ∪ A)
- Preserve the determinant in the original relation (R₂ := R - A)

Example

- Decomposition according FD State → Premier
  - Government(State, Premier)
  - Cities(City, State, Inhabitants)
Functional Dependencies

- Armstrong's axioms
- Transitive Closure
- Canonical Cover

Normal Form

- 1NF
- 2NF
- 3NF