# Database Management Systems

## Module 22: Relational Database Design/2

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

*ppd@cse.iitkgp.ac.in*

# Module Recap





- Identified the features of good relational design
- Familiarized with the First Normal Form

# Module Objectives





- To Introduce Functional Dependencies

# Module Outline





- Functional Dependencies

# Functional Dependencies

# Goal: Devise a Theory for Good Relations



- Decide whether a particular relation $R$ is in "good" form.
- In the case that a relation $R$ is not in "good" form, decompose it into a set of relations $\{R_1, R_2, \ldots, R_n\}$ such that
  - each relation is in good form
  - the decomposition is a lossless-join decomposition
- The theory is based on:
  - Functional dependencies
  - Multivalued dependencies
  - Other dependencies

# Functional Dependencies



- Constraints on the set of legal relations
- Require that the value for a certain set of attributes determines uniquely the value for another set of attributes
- A functional dependency is a generalization of the notion of a *key*

# Functional Dependencies (2)



- Let $R$ be a relation schema
  $\alpha \subseteq R$ and $\beta \subseteq R$
- The functional dependency or FD
  $\alpha \rightarrow \beta$
  holds on $R$ if and only if for any legal relations $r(R)$, whenever any two tuples $t_1$ and $t_2$ of $r$ agree on the attributes $\alpha$, they also agree on the attributes $\beta$. That is,

$$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$

- Example: Consider $r(A, B)$ with the following instance of $r$.

| A | B |
|---|---|
| 1 | 4 |
| 1 | 5 |
| 3 | 7 |

- On this instance, $A \rightarrow B$ does **NOT** hold, but $B \rightarrow A$ does hold. So we cannot have tuples like (2, 4), or (3, 5), or (4, 7) added to the current instance.

# Functional Dependencies (3)



- $K$ is a superkey for relation schema $R$ if and only if $K \rightarrow R$
- $K$ is a candidate key for $R$ if and only if
  - $K \rightarrow R$ and
  - for no $\alpha \subset K$, $\alpha \rightarrow R$
- Functional dependencies allow us to express constraints that cannot be expressed using superkeys. Consider the schema:

  *inst_dept(ID, name, salary, dept_name, building, budget)*
- We expect these functional dependencies to hold:

  *dept_name* $\rightarrow$ *building*

  *dept_name* $\rightarrow$ *budget*

  *ID* $\rightarrow$ *budget*

  but would not expect the following to hold:

  *dept_name* $\rightarrow$ *salary*

# Functional Dependencies (4)



- We use functional dependencies to:
  - test relations to see if they are legal under a given set of functional dependencies.
    - If a relation $r$ is legal under a set $F$ of functional dependencies, we say that $r$ **satisfies** $F$
  - specify constraints on the set of legal relations
    - We say that $F$ **holds on** $R$ if all legal relations on $R$ satisfy the set of functional dependencies $F$
- **Note**: A specific instance of a relation schema may satisfy a functional dependency even if the functional dependency does not hold on all legal instances
  - For example, a specific instance of instructor may, by chance, satisfy
    $$name \rightarrow ID$$
  - In such cases we do not say that $F$ holds on $R$

# Functional Dependencies (5)



- A functional dependency is trivial if it is satisfied by all instances of a relation
  - Example:
    - $ID, name \rightarrow ID$
    - $name \rightarrow name$
- In general, $\alpha \rightarrow \beta$ is trivial if $\beta \subseteq \alpha$.

# Functional Dependencies (6)



- Functional dependencies are:

| StudentID | Semester | Lecture | TA |
|---|---|---|---|
| 1234 | 6 | Numerical Methods | John |
| 1221 | 4 | Numerical Methods | Smith |
| 1234 | 6 | Visual Computing | Bob |
| 1201 | 2 | Numerical Methods | Peter |
| 1201 | 2 | Physics II | Simon |

  - $StudentID \rightarrow Semester$
    $StudentID, Lecture \rightarrow TA$
    $\{StudentID, Lecture\} \rightarrow \{TA, Semester\}$

# Functional Dependencies (7)



- Functional dependencies are:

| Employee ID | Employee Name | Department ID | Department Name |
|---|---|---|---|
| 0001 | John Doe | 1 | Human Resources |
| 0002 | Jane Doe | 2 | Marketing |
| 0003 | John Smith | 1 | Human Resources |
| 0004 | Jane Goodall | 3 | Sales |

  - $EmployeeID \rightarrow EmployeeName$
    $EmployeeID \rightarrow DepartmentID$
    $DepartmentID \rightarrow DepartmentName$

# Functional Dependencies (8): Armstrong's Axioms



- Given a set of Functional Dependencies $F$, we can infer new dependencies by the **Armstrong's Axioms**:
  - **Reflexivity**: if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$
  - **Augmentation**: if $\alpha \rightarrow \beta$, then $\gamma\alpha \rightarrow \gamma\beta$
  - **Transitivity**: if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$
- These axioms can be repeatedly applied to generate new FDs and added to $F$
- A new FD obtained by applying the axioms is said to the **logically implied** by $F$
- The process of generations of FDs terminate after finite number of steps and we call it the **Closure Set** $F^+$ for FDs $F$. This is the set of **all** FDs logically implied by $F$
- Clearly, $F \subseteq F^+$
- These axioms are
  - **Sound** (generate only functional dependencies that actually hold), and
  - **Complete** (eventually generate all functional dependencies that hold)
- Prove the axioms from definitions of FDs
- Prove the soundness and completeness of the axioms

# Functional Dependencies (9): Closure of a Set of FDs



- $F = \{A \rightarrow B, B \rightarrow C\}$
- $F^+ = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

# Module Summary



- Introduced the notion of Functional Dependencies

**Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.**
**Edited and new slides are marked with "PPD".**