Functional Dependencies

MSCI346
Lukasz Golab

Acknowledgement: slides derived from material provided by
Sliberschatz, Korth and Sudarshan, copyright 2019, www.db-book.com,
and by Prof. Wojciech Golab, ECE356

Learning Outcomes

- Introduction to database normalization:
  - features of a good relational design
  - atomic domains
  - functional dependencies and relational decomposition
  - first normal form (1NF)
- Develop a working knowledge of functional dependency theory:
  - closure of a set of functional dependencies
  - inference rules
  - ~~closure of an attribute set~~
  - ~~canonical cover~~
- Textbook sections (6th ed.): 8.1, 8.2, 8.3, 8.4

To Merge or not to Merge…

- Suppose that instead of maintaining separate *instructor* and *department* relations, we would like to merge them into a single relation *inst_dept*:

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

- What are the pros and cons of this design?

Motivation

- Redundancy in a database leads to troublesome anomalies.
  - **Update anomalies**: a repeated value may be changed in one place but not in another place.
  - **Insertion anomalies:** in order to insert one value, it becomes necessary to insert some unrelated value.
  - **Deletion anomalies:** deleting one type of information leads to the loss of an another unrelated type of information.

Motivation (Cont.)

- Consider our earlier example where *instructor* and *department* are merged into *inst_dept*.
  - If one department changes its name, we may need to update many rows in *inst_dept*.
  - If an instructor is inserted, we must include the instructor's department budget in the same row.
  - If all ECE instructors are deleted, the ECE department no longer has any representation in the database.

Different Forms of Repetition

- Example 1: repetition within one tuple

  section(*course_id, sec_id, semester, year,*
  building, room_number)

  ('ECE356', '001', 'W13', 2013, 'E2', 'E2-1303')

- Example 2: repetition between tuples

  inst_dept(*ID*, *name*, *salary*, *dept_name*, *building*, *budget*)

  ('22222', 'Einstein', '95000', 'Physics', 'Watson', 70000)
  ('33456', 'Gold', '87000', 'Physics', 'Watson', 70000)

Diagnosis and Remedy

- In the examples considered, there are two types of repetition:
  1. The value domains of some attributes are not **atomic**: one value may encode multiple pieces of information.

     Example: 'E2-1303' repeats information in 'E2'

  2. Attribute values in different tuples are related by **functional dependencies**: one subset of attributes **functionally determines** the values of another subset. (A type of constraint present in real data.)

     Example: *dept_name* $\rightarrow$ *building, budget*

- Remedies:
  1. Break up value domains to create atomic domains.
  2. Decompose relations to avoid specific types of FD's.

First Normal Form (1NF)

- A value domain is **atomic** if its elements are considered to be indivisible units.
  - Examples of non-atomic domains:
    - multi-valued and composite attributes
    - identifiers such as 'ECE356', which can be broken up into parts
- A relational schema *R* is in **first normal form** if the domains of all attributes of *R* are atomic.
- Non-atomic domains are bad because:
  - they complicate storage
  - they encourage redundancy
  - they lead to information being encoded in business logic (e.g., application parses 'ECE356' to obtain department name)
- **Assumption: from now on, all relations are in first normal form unless we say otherwise.**

A Theory of Functional Dependencies

- **Functional dependencies** (FDs) are constraints on the set of **legal relations** – ones that conform to some conceptual model of the data, which itself is guided by our informal understanding of the world).
- FDs state that the value for a certain set of attributes determines (i.e., constrains) uniquely the value for another set of attributes. An FD is a generalization of the notion of a **key**.

  Example: *dept_name* $\rightarrow$ *building, budget*

  Pronunciation:
  dept_name **functionally determines** building and budget

- **Note:** If we know the value of *dept_name* then we know that the values of *building* and *budget* are uniquely determined, but we may not know immediately what these values are.

Functional Dependencies (Cont.)

- Let R be a relation schema where

$$\alpha \subseteq R \text{ and } \beta \subseteq R$$

- The **functional dependency**

$$\alpha \rightarrow \beta$$

holds on R if and only if for any legal relation $r(R)$, whenever any two tuples t_1 and t_2 of r agree on the attributes α, they also agree on the attributes β. That is,

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

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

A	B
1	4
1	5
3	7

- On this instance, $A \rightarrow B$ does **NOT** hold, but $B \rightarrow A$ may or may not hold.

Functional Dependencies (Cont.)

- 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
  - there is no $\alpha \subset K$ such that $\alpha \rightarrow R$.
- Functional dependencies allow us to express constraints that cannot be expressed using superkeys. Consider the schema:

  inst_dept (<u>*ID*</u>, *name, salary, dept_name, building, budget*)

  We expect the following functional dependencies to hold:

  $$dept\_name \rightarrow building$$

  $$ID \rightarrow building$$

  but we would not expect the following to hold:

  $$dept\_name \rightarrow salary$$

Functional Dependencies (Cont.)

- 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 whenever $\beta \subseteq \alpha$.

Closure of a Set of Functional Dependencies

- Functional dependencies help us reason about data:
  - FDs represent constraints on legal relations
  - FDs help us identify certain forms redundancy
- Given a set F of functional dependencies, there may be certain other functional dependencies that are not in F but are logically implied by those in F.
  - For example: If F contains only $A \rightarrow B$ and $B \rightarrow C$, then we can infer that $A \rightarrow C$
- The set of **all** functional dependencies logically implied by F is the **closure** of F.
- We denote the *closure* of F by $\boldsymbol{F^+}$.
- In general $F^+ \supseteq F$ holds.

Armstrong's Axioms

- We can find F^+, the closure of F, by repeatedly applying **Armstrong's Axioms:**
  - if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ (**reflexivity**)
  - if $\alpha \rightarrow \beta$, then $\gamma\ \alpha \rightarrow \gamma\ \beta$ (**augmentation**)
  - if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ (**transitivity**)
- These axioms (more correctly called **inference rules**) are
  - **sound**
    (i.e., they generate only functional dependencies that actually hold)
  - and **complete**
    (i.e., they generate all functional dependencies that hold)

Example: Applying Armstrong's Axioms

- $R = (A, B, C, G, H, I)$
  $F = \{ \; A \rightarrow B$
  $A \rightarrow C$
  $CG \rightarrow H$
  $CG \rightarrow I$
  $B \rightarrow H \}$
- Applying the axioms, we can obtain additional members of F^+
  - $A \rightarrow H$
    - by transitivity from $A \rightarrow B$ *and* $B \rightarrow H$
  - $AG \rightarrow I$
    - by augmenting $A \rightarrow C$ with G, to get $AG \rightarrow CG$
      and then by transitivity with $CG \rightarrow I$
  - $CG \rightarrow HI$
    - by augmenting $CG \rightarrow I$ to infer $CG \rightarrow CGI$,
      and augmenting of $CG \rightarrow H$ to infer $CGI \rightarrow HI$,
      and then by transitivity

Additional Inference Rules

- The following rules can also be used to compute functional dependencies:
  - If $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\,\gamma$ holds (**union**)
  - If $\alpha \rightarrow \beta\,\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds (**decomposition**)
  - If $\alpha \rightarrow \beta$ holds and $\gamma\,\beta \rightarrow \delta$ holds, then $\alpha\,\gamma \rightarrow \delta$ holds (**pseudotransitivity**)
- Note1: The above rules can be inferred from Armstrong's axioms.
- Note2: If you're asked on an exam to prove that some FD holds <u>using Armstrong's axioms</u> then use only Armstrong's axioms and <u>do not use these additional rules</u>.