25 SOICT
YEARS ANNIVERSARY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Lesson 5
Database design: Bottom-up Approach

Part 1: Functional Dependency

Learning objective

•*Upon completion of this lesson, students will be able to:*

1. Recall the concept of functional dependency, Armstrong's axioms and secondary rules
2. Identify closure of a FD set, closure of a set of attributes
3. Find a minimal key of a relation under a set of FDs
4. Identify the equivalence of sets of FDs and find the minimal cover of a set of FDs

Outline

1. Functional Dependency
2. Armstrong's Axioms and secondary rules
3. Closure of a FD set, closure of a set of attributes
4. A minimal key
5. Equivalence of sets of FDs
6. Minimal Sets of FDs

1. Functional Dependency

- 1.1. Introduction
- 1.2. Definition

1.1. Motivation

- Designing DB: one of the most difficult tasks
- One simplest design approach is to use a big table and store all data
- But what's the problem with this?
  - Anomalies
  - Redundancies

1.1. Motivation: Insertion Anomalies

- PK: (student_id, subject_id)
- We can not insert a new subject if we do not have a student assigned to it yet
- We can not insert a null value into PK attributes

student_id	full_name	dob	subject_id	name	result
1234	David Beckham	12/21/1997	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4843	Data integration	B
1234	David Beckham	12/21/1997	IT4868	Web mining	C
1497	Tony Blair	03/01/1999	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4868	Web mining	B
1542	Margaret Thatcher	05/08/1997	IT2000	Introduction to ICT	C

1.1. Motivation: Update anomalies

- An instance where the same information must be updated in several different places
- If you update the name of subject "**Databases**", you need to update in two different places (not efficient)

student_id	full_name	dob	subject_id	name	result
1234	David Beckham	12/21/1997	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4843	Data integration	B
1234	David Beckham	12/21/1997	IT4868	Web mining	C
1497	Tony Blair	03/01/1999	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4868	Web mining	B
1542	Margaret Thatcher	05/08/1997	IT2000	Introduction to ICT	C

1.1. Motivation: Deletion Anomalies

- Where deleting one piece of data inadvertently causes other data to be lost
- If we delete student **Margaret Thatcher**, then we will lose information about subject **"Introduction to ICT"**

student_id	full_name	dob	subject_id	name	result
1234	David Beckham	12/21/1997	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4843	Data integration	B
1234	David Beckham	12/21/1997	IT4868	Web mining	C
1497	Tony Blair	03/01/1999	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4868	Web mining	B
1542	Margaret Thatcher	05/08/1997	IT2000	Introduction to ICT	C

1.2. Definition

- FD is the single most important concept in relational schema design theory

  - Suppose that R = {A_1, A_2, ... , A_n} , X and Y are non-empty subsets of R.

  - A functional dependency (FD), denoted by X $\rightarrow$ Y, specifies a constraint on the possible tuples that can form a relation state r of R. The constraint is that, for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$, they must also have $t_1[Y] = t_2[Y]$.

    - X: the left-hand side of the FD

    - Y: the right-hand side of the FD

1.2. Definition

- This means that the values of the X component of a tuple uniquely (or functionally) determine the values of the Y component.

- A FD X → Y is trivial if X ⊇ Y

- If X is a candidate key of R, then X → R

1.2. Definition

- Examples
  - AB → C

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c2	d1
a2	b1	c3	d1

subject

subject_id	name	credit	percentage_final_exam
IT3090	Databases	3	0.7
IT4843	Data integration	3	0.7
IT4868	Web mining	2	0.6
IT2000	Introduction to ICT	2	0.5
IT3020	Discrete Mathematics	2	0.7
IT3030	Computer Architectures	3	0.7

- subject_id → name,
- subject_id → credit,
- subject_id → percentage_final_exam,
- subject_id → {name, credit}

2. Armstrong's axioms

2.1. Armstrong's axioms

2.2. Secondary rules

2.3. An example

2.1. Armstrong's axioms

- Given
  - $R = \{A_1, A_2, \ldots, A_n\}$, X, Y, Z, W are subsets of R.
  - XY denoted for $X \cup Y$
- Reflexivity
  - If $Y \subseteq X$ then $X \rightarrow Y$
- Augmentation
  - If $X \rightarrow Y$ then $XZ \rightarrow YZ$
- Transitivity
  - If $X \rightarrow Y$, $Y \rightarrow Z$ then $X \rightarrow Z$

2.2. Secondary rules

- Union
  - If $X{\rightarrow}Y$, $X{\rightarrow}Z$ then $X{\rightarrow}YZ$.

- Pseudo-transitivity
  - If $X{\rightarrow}Y$, $WY{\rightarrow}Z$ then $XW{\rightarrow}Z$.

- Decomposition
  - If $X{\rightarrow}Y$, $Z \subseteq Y$ then $X{\rightarrow}Z$

2.3. An example

- Given a set of FDs: F = {AB $\rightarrow$ C, C $\rightarrow$ A}
- Prove: BC $\rightarrow$ ABC
  - From C $\rightarrow$ A, we have BC $\rightarrow$ AB (Augmentation)
  - From AB $\rightarrow$ C, we have AB $\rightarrow$ ABC (Augmentation)
  - And we can conclude BC $\rightarrow$ ABC (Transitivity)

3. Closure of a FD set, closure of a set of attributes

3.1. Closure of a FD set

3.2. Closure of a set of attributes

3.3. A problem

3.1. Closure of a FD set

- Suppose that F = {A $\to$ B, B $\to$ C} on R(A, B, C,…). We can infer many

  FDs such as:

  A $\to$ C, AC $\to$ BC,…

- **Definition**

  - Formally, the set of all dependencies that include F as well as all

    dependencies

    that can be inferred from F is called the closure of F, denoted by F$^+$.

- F $\vDash$ X $\to$ Y to denote that the FD X $\to$ Y is inferred from the set of FDs F.

3.2. Closure of a set of attributes

- Problem
  - We have F, and $X \rightarrow Y$, we have to check if $F \vDash X \rightarrow Y$ or not

- Should we calculate F^+? $\Rightarrow$ Closure of a set of attributes

- Definition
  - For each such set of attributes X, we determine the set X^+ of attributes that are functionally determined by X based on F; X^+ is called the closure of X under F.

3.2. Closure of a set of attributes

- To find the closure of an attribute set X^+ under F

    - **Input:** A set F of FDs on a relation schema R, and a set of attributes X, which is a subset of R.
    $X^0 := X;$
    repeat
        for each functional dependency $Y \rightarrow Z$ in F do
            if $X^{i-1} \supseteq Y$ then $X^i := X^{i-1} \cup Z;$

        else $X^i := X^{i-1}$
    until (X^i unchanged);
    $X^+ := X^i$

3.2. Closure of a set of attributes

- An example
  - Given R = {A, B, C, D, E, F} and F = {AB $\rightarrow$ C, BC $\rightarrow$ AD, D $\rightarrow$ E, CF $\rightarrow$ B}.

    Calculate $(AB)^+_F$
    - X^0 = AB
    - X^1 = ABC (from AB $\rightarrow$ C)
    - X^2 = ABCD (from BC $\rightarrow$ AD)
    - X^3 = ABCDE (from D $\rightarrow$ E)
    - X^4 = ABCDE
    - $(AB)^+_F$=ABCDE

3.3. A Problem

- $X \to Y$ can be inferred from F if and only if $Y \subseteq X^+_F$

- $F \vDash X \to Y \Leftrightarrow Y \subseteq X^+_F$

- An example

  - Let R = {A, B, C, D, E}, F = {A $\to$ B, B $\to$ CD, AB $\to$ CE}.

  Consider whether or not $F \vDash A \to C$

    - $(A)^+_F$ = ABCDE $\supseteq$ {C}

4. Minimal key

4.1. Definition

4.2. An algorithm to find a minimal key

4.3. An example

4.1. Definition

- Minimal key

  - Given $R = \{A_1, A_2, \ldots, A_n\}$, a set of FDs F

  - K is considered as a minimal key of R if:

    - $K \subseteq R$

    - $K \rightarrow R \in F^+$

    - Với $\forall K' \subset K$, thì $K' \rightarrow R \notin F^+$

  - $K^+ = R$ and $K \backslash \{A_i\} \rightarrow R \notin F^+$

4.2. An algorithm to find a minimal key

- To find a minimal key

  - Input: $R = \{A_1, A_2, \ldots, A_n\}$, a set of FDs F
    - Step^0 $K^0 = R$
    - Step^i If $(K^{i-1} \setminus \{A_i\}) \rightarrow R$ then $K^i = K^{i-1} \setminus \{A_i\}$
                   else $K^i = K^{i-1}$
    - Step^{n+1} $K = K^n$

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

4.3. An example

- Given R = {A, B, C, D, E}, F = {AB $\rightarrow$ C, AC $\rightarrow$ B, BC $\rightarrow$ DE}.

- Find a minimal key
  - Step^0: K^0= R = ABCDE
  - Step^1: Check if or not (K^0\{A}) $\rightarrow$ R (i.e, BCDE $\rightarrow$ R).
  - (BCDE)$^+$= BCDE $\neq$ R. Vậy K^1 = K^0 = ABCDE
  - Step^2: Check if or not (K^1\{B}) $\rightarrow$ R (i.e, ACDE $\rightarrow$ R).
  - (ACDE)$^+$ = ABCDE = R. So, K^2 = K^1\{B} = ACDE
  - Step^3: K^3 = ACDE
  - Step^4: K^4 = ACE
  - Step^5: K^5 = AC

- We infer that AC is a minimal key

4.4. An other algorithm to find a minimal key

Input: $U = \{A_1, A_2, \ldots, A_n\}$, F

Output: a minimal key

- **Step 1:**

  VT = set of all attributes on the left-side of FD in F

  VP = set of all attributes on the right-side of FD in F

  X = U \ VP: set of attributes **that must be in K**

  Y = VP \ VT: set of attributes **that must NOT be in** K

  Z = VP $\cap$ VT: set of attrbutes **that may be in K**

- **Step 2**: If $(X)^+ = U$ then X is the minimal key: K = X. End!

- **Step 3**: If $(X)^+ \neq U$ then

  - $K^0 = X \cup Z$

  - Repeat: check if we can remove any attribute from Z (similar to slide 22)

  - **K = K^i**

5. Equivalence of Sets of FDs

5.1. Definition

5.2. An example

5.1. Definition

- Definition:
  - A set of FDs F is said to cover another set of FDs G if every FD in G is also

    in F^+ (every dependency in G can be inferred from F).

- Check if F and G are equivalent:
- Two sets of FDs F and G are equivalent if $F^+ = G^+$.
  - Therefore, equivalence means that every FD in G can be inferred from F, and every FD in F can be inferred from G;
  - That is, G is equivalent to F if both the conditions - G covers F and F covers G - hold.

5.2. An example

- Prove that F = {A $\to$ C, AC $\to$ D, E $\to$ AD, E $\to$ H} and G = {A $\to$ CD, E $\to$ AH} are equivalent
  - For each FD of F, prove that it is in G^+
    - A $\to$ C: $(A)^+_G$ = ACD $\supseteq$ C, so A $\to$ C $\in$ G^+
    - AC $\to$ D: $(AC)^+_G$ = ACD $\supseteq$ D, so AC $\to$ D $\in$ G^+
    - E $\to$ AD: $(E)^+_G$ = EAHCD $\supseteq$ AD, so E $\to$ AD $\in$ G^+
    - E $\to$ H: $(E)^+_G$ = EAHCD $\supseteq$ H, so E $\to$ H $\in$ G^+
    - $\Rightarrow F^+ \subseteq G^+$
  - For each FD of G, prove that it is in F^+ (the same)
    - $\Rightarrow G^+ \subseteq F^+$
  - $\Rightarrow F^+ = G^+$

6. A minimal cover of a set of FDs

6.1. Definition

6.2. An algorithm to find a minimal cover of a set of FDs

6.3. An example

6.1. Definition

- Minimal Sets of FDs
    - A set of FDs F to be minimal if it satisfies:
        - Every dependency in F has a single attribute for its right-hand side.
        - We cannot replace any dependency $X \rightarrow A$ in F with a dependency $Y \rightarrow A$, where Y is a proper subset of X, and still have a set of dependencies that is equivalent to F.
        - We cannot remove any dependency from F and still have a set of dependencies that is equivalent to F.
    - A set of dependencies in a standard or canonical form and with no redundancies

6.2. An algorithm to find a minimal cover of a set of FDs

- Finding a Minimal Cover F for a Set of FDs G

  - Input: A set of FDs G.

    - 1. Set F := G.

    - 2. Replace each functional dependency $X \rightarrow \{A_1, A_2, ..., A_n\}$ in F by the n FDs $X \rightarrow A_1$, $X \rightarrow A_2$, ..., $X \rightarrow A_n$.

    - 3. For each FD $X \rightarrow A$ in F

      - for each attribute B that is an element of X

      - if $\{\{F - \{X \rightarrow A\}\} \cup \{(X - \{B\}) \rightarrow A\}\}$ is equivalent to F

      - then replace $X \rightarrow A$ with $(X - \{B\}) \rightarrow A$ in F.

    - 4. For each remaining functional dependency $X \rightarrow A$ in F

      - if $\{F - \{X \rightarrow A\}\}$ is equivalent to F, then remove $X \rightarrow A$ from F.

6.3. An example

- G = {B → A, D → A, AB → D}.

- We have to find the minimal cover of G.
  - All above dependencies are in canonical form
  - In step 2, we need to determine if AB → D has any redundant attribute
    - on the left-hand side; that is, can it be replaced by B → D or A → D?
    - Since B → A then AB → D may be replaced by B → D.
    - We now have a set equivalent to original G, say G_1: {B → A, D → A, B → D}.
  - In step 3, we look for a redundant FD in G_1. Using the transitive rule on
    - B → D and D → A, we conclude B → A is redundant.
    - Therefore, the minimal cover of G is {B → D, D → A}

Remark

- Functional dependencies
- Armstrong axioms and their secondary rules
- Closure of a set of FDs
- Closure of a set of attributes under a set of FDs
- An algorithm to find a minimal key
- Equivalence of sets of FDs
- Finding a minimal set of a set of FDs

Summary

1. **Functional Dependency**
    - A FD $X \rightarrow Y$: the values of the X component of a tuple uniquely (or functionally) determine the values of the Y component

2. **Armstrong 's Axioms and secondary rules**
    - Reflexivity, Augmentation, Transitivity
    - Union, Pseudo-transitivity, Decomposition

3. **Closure of a FD set, closure of a set of attributes**
    - All dependencies that can be inferred from F, include F, is called the closure of F, denoted by F^+
    - A set of attributes are functionally determined by X based on F

4. **A minimal key**
    - A minimal set of attributes can determine R

5. **Equivalence of sets of functional dependencies**
    - F is equivalent to G if every dependency in G can be inferred from F, and every dependency in F can be inferred from G

6. **A minimal cover of a set of FDs**
    - A set of dependencies in a standard or canonical form and with no redundancies

Thank you for your attention!

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

soict.hust.edu.vn/   fb.com/groups/soict

References

- Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd edition, Mc Graw Hill, 2003.
- Elmasri and Navathe, Fundamentals of Database Systems, 6th edition, Addison-Wesley, 2011.

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY