



Database

Lesson 9. Normalization

Viet-Trung Tran

<https://is.hust.edu.vn/~trungtv/>

Learning Map

Sequence	Title
1	Introduction to Databases
2	Relational Databases
3	Relational Algebra
4	Structured Query Language – Part 1
5	Structured Query Language – Part 2
6	Constraints and Triggers
7	Entity Relationship Model
8	Functional Dependency
9	Normalization
10	Storage - Indexing
11	Query Processing
12	Transaction Management – Part 1
13	Transaction Management – Part 2

Outline

- Introduction
- Normal Forms
- Normalization

Objectives

- Upon completion of this lesson, students will be able to:
 - Know why we need normalization in relational DB
 - Identify normal forms such as 1st NF, 2nd NF, 3rd NF
 - Know how to normalize a relational DB into 3NF

Keywords

Keyword	Description
1st Normal Form	the domain of an attribute must include only atomic (simple, indivisible) values and the value of any attribute in a tuple must be a single value from the domain of that attribute.
2nd Normal Form	A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on <i>any candidate key</i> .
3rd Normal Form	A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on <i>any candidate key</i> .
Normalization	Normalization is the process of removing anomalies and redundancies from DB

1. Introduction

- Motivation
- Full & Partial Dependency
- Transitive Dependency

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

<u>student_id</u>	<u>full_name</u>	<u>dob</u>	<u>subject_id</u>	<u>name</u>	<u>result</u>
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 Databases subject name, you need to update in two different places (not efficient)

<u>student_id</u>	<u>full_name</u>	<u>dob</u>	<u>subject_id</u>	<u>name</u>	<u>result</u>
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

<u>student_id</u>	full_name	dob	<u>subject_id</u>	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

- Normalization is the process of removing anomalies and redundancies from DB

1.2. Full & Partial Dependency

<u>student_id</u>	full_name	dob	<u>subject_id</u>	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

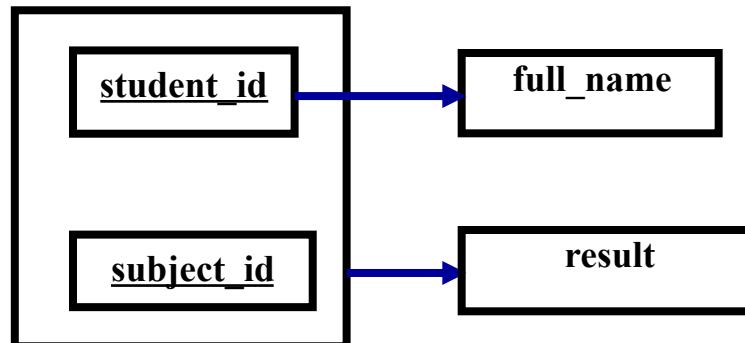
Key: (student_id, subject_id)

Full Key Dependency:

{student_id, subject_id} → result

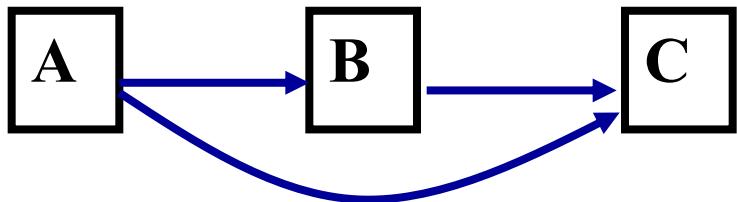
Partial Key Dependency:

student_id → full_name



1.3. Transitive dependency

- If $A \rightarrow B$ and $B \rightarrow C$
 - Attribute A must be the determinant of C.
 - Attribute A transitively determines attribute C or
 - C is transitively dependent on A



2. Normal Forms

- Introduction
- 1st Normal Form
- 2nd Normal Form
- 3rd Normal Form

2.1. Introduction

- Each form was designed to eliminate one or more of the anomalies: First NF; Second NF; Third NF
- Unnormalised Form (UNF)
 - A table that contains one or more repeating groups. I.e., its cell may contain multiple values

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

Multi Value
Or repeating
groups

2.2. First Normal Form (1NF)

- A cell in a relation contains one and only one value.
 - Disallows composite attributes, multivalued attributes or nested relations

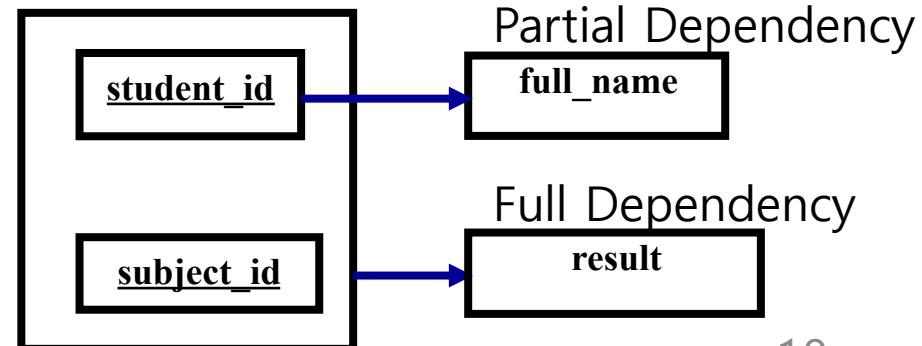
<u>student_id</u>	full_name	dob	<u>subject_id</u>	name	result
1234	David Beckham	12/21/1997	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4843	Data integration	B
1234	David Beckham	12/21/1997	IT4868	Web mining	C
1497	Tony Blair	03/01/1999	IT3090	Databases	A
1238	Theresa May	08/06/1998	IT4868	Web mining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

Full functional dependency

- Given $R(U)$, F is a set of FDS in R . $X, Y \subseteq U$. Y is fully dependent on X iff:
 - $X \rightarrow Y \subset F^+$
 - $\exists X' \subset X : X' \rightarrow Y \in F^+$

2.3. Second Normal Form (2NF)

- Based on the concept of full functional dependency
- A prime attribute
 - It is an attribute that is member of some candidate key
- 2NF relation is
 - in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key



2.3. Example

- Sales(sid, sname, city, item, price)
- $F = \{sid \rightarrow (sname, city), (sid, item) \rightarrow price\}$
- PK (sid,item)
- sname, city are partially dependent on PK
- Sales is not in 2NF

2.4. Third Normal Form (3NF)

- A relation that is
 - In 2NF and in which no non-primary-key attribute is transitively dependent on the primary key
 - I.e, all non-prime attributes are fully & directly dependent on the PK.

2.4. Example

S (sid, sname, city)

Sales(sid, item, price)

F = {sid → sname, city, sid -> item, price}

- S, Sales are in 3NF
 - ItemInfo(item, price, discount).
 - F = {item→price, price→discount}
- The attribute discount is not directly dependent on item
- ItemInfo is not in 3NF

3. Normalization

- Properties of relational decompositions
- An algorithm decomposes a universal relation into 3NF
- Some examples

3.1. Properties of relational decompositions

- A single universal relation schema $R = \{A_1, A_2, \dots, A_n\}$ that includes all the attributes of the DB
- F is a set of FDs holds on R
- Using the FDs, the algorithms decompose the universal relation schema R into a set of relation schemas $D = \{R_1, R_2, \dots, R_m\}$; D is called a decomposition of R .

3.1. Properties of relational decompositions

- Properties:
 - Attribute preservation
 - Each attribute in R will appear in at least one relation schema R_i in the decomposition so that no attributes are *lost*
 - Dependency preservation
 - Each FD $X \rightarrow Y$ specified in F either appeared directly in one of the R_i in the decomposition D or could be inferred from the dependencies that appear in some R_i .
 - Lossless join
 - $r = \Pi_{R1}(r) \bowtie \Pi_{R2}(r) \bowtie \dots \bowtie \Pi_{Rm}(r)$

3.1. Properties of relational decompositions

- An example
 - Suppose we have a relation:
 $\text{Learn}(\text{student_id}, \text{full_name}, \text{dob}, \text{subject_id}, \text{name}, \text{result})$
 - We split it into two relations:
 $\text{Student}(\text{student_id}, \text{full_name}, \text{dob})$
 $\text{Subject}(\text{subject_id}, \text{name})$
 - This decomposition does not warrant:
 - Attribute preservation: Lost information about "result"
 - Dependency preservation condition, for instance, $(\text{student_id}, \text{subject_id}) \rightarrow \text{result}$ is lost.
 - Lossless join property, i.e., we can join these two relations

3.2. An algorithm decomposes a universal relation into 3NF

- Input: An universal relation R and a set of FDs F on the attributes of R.
 - Find a minimal cover G for F
 - For each left-hand-side X of a FD that appears in G, create a relation schema in D with attributes $\{X \cup \{A_1\} \cup \{A_2\} \dots \cup \{A_k\}\}$, where $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_k$ are the only dependencies in G with X as the left-hand-side (X is the key of this relation);
 - If none of the relation schemas in D contains a key of R, then create one more relation schema in D that contains attributes that form a key of R.
 - Place any remaining attributes (that have not been placed in any relation) in a single relation schema to ensure the attribute preservation property.

3.2. An algorithm decomposes a universal relation into 3NF

- If none of the relation schemas in D contains a key of R, then create one more relation schema in D that contains attributes that form a key of R.

3.3. Some examples

- Example 1:
 - Given $R = \{A, B, C, D, E, F, G\}$, $F = \{A \rightarrow B; ABCD \rightarrow E; EF \rightarrow G; ACDF \rightarrow EG\}$
 - A minimal cover of F is $G = \{A \rightarrow B, ACD \rightarrow E, EF \rightarrow G\}$
 - Find a minimal key: $K = ACDF$
 - We have $R_1(AB)$, $R_2(ACDE)$, $R_3(EFG)$
 - Since K is not a subset of R_i , we have a new relation $R_4(ACDF)$
 - In conclusion, we have a decomposition $D = \{R_1, R_2, R_3, R_4\}$

3.3. Some examples

- Example 2:
 - Given $R(\text{student_id}, \text{name}, \text{birthday}, \text{advisor}, \text{department}, \text{semester}, \text{course}, \text{grade})$
 - $F = \{ \text{student_id} \rightarrow (\text{name}, \text{birthday}); \text{advisor} \rightarrow \text{department}; (\text{student_id}, \text{semester}, \text{course}) \rightarrow (\text{grade}, \text{advisor}, \text{department}) \}$
 - We denote like this: student_id (A), name (B), birthday (C), advisor (D), department (E), semester (F), course (G), grade (H)
 - F is rewritten as $\{A \rightarrow BC; D \rightarrow E; AFG \rightarrow HDE\}$
 - A minimal cover of F is $G = \{A \rightarrow B; A \rightarrow C; D \rightarrow E; AFG \rightarrow H\}$
 - Find a minimal key: $K = AFG$
 - We have $R1(ABC)$, $R2(DE)$, $R3(AFGH)$
 - Since K is a subset of $R3$, we have a decomposition $D = \{R1, R2, R3\}$ or $\{R1(\text{student_id}, \text{name}, \text{birthday}), R2(\text{advisor}, \text{department}), R3(\text{student_id}, \text{semester}, \text{course}, \text{grade})\}$

Remarks

- Motivation of normalization
- Full & Partial Dependency
- Transitive dependency
- 1NF, 2 NF, 3 NF
- Properties of relational decompositions
- An algorithm decomposes a universal relation into 3NF

Quiz

No	Question (Multiple Choice)	Answer (1,2,3,4)	Commentary
1	<p>How many kinds of anomalies have we just studied?</p> <ol style="list-style-type: none">1.2.3.4.	3	Insert anomalies, Update anomalies, Delete anomalies
2	<p>A relation is under the form of 3NF must satisfy:</p> <ol style="list-style-type: none">1. A cell in a relation contains one and only one value2. All non-primary-key attributes fully depend on the primary key3. All non-primary-key attributes directly depend on the primary key4. 1, 2, 3 together	4	<p>A relation is under the form of 3NF must satisfy:</p> <ul style="list-style-type: none">- Each cell contains only an atomic value (1NF)- All non-primary-key attributes fully depend on the primary key (2NF)- All non-primary-key attributes directly depend on the primary key (3NF)
3			



Next lesson: Storage & Indexing

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Database Systems: The Complete Book. Pearson Prentice Hall. the 2nd edition. 2008: Chapter 7
- Nguyễn Kim Anh, Nguyên lý các hệ cơ sở dữ liệu, NXB Giáo dục. 2004: Chương 7