Database Design II: Normalization Forms

DSC 301: Lecture 9

February 24, 2021

Lecture Objectives

- Review Functional Dependencies
- State purpose of normalization
  - Mention performance trade-off, thus denormalization
- State Normalization Forms
- Apply normalization (use MySQL Workbench)

Functional Dependencies (FD)

- Notation: $A \rightarrow B$ read "B is functionally dependent on A" (or A determines B)
- Generalizes concept of keys (i.e., a key is a FD)
- $A \rightarrow B$ if $t_i[B] = t_j[B]$ when $t_i[A] = t_j[A]$ for all i, j.
- Used in the normalization process (topic of this lecture)

Example 1. *Given the relation R in Table 1, verify that $A \rightarrow B$. Find another FD.*

Table 1: Functional Dependency

R =

A	B	C	D
1	1	1	1
1	1	3	1
2	1	3	4
2	1	3	1
3	6	3	1

Purpose of Normalization

A table (i.e., relation) is the basic building block of a database, therefore its design is of significant importance. How to produce good table structures?

Definition 1. ***Normalization*** *is a technique to produce a set of relations with certain desirable properties (given data requirements of the enterprise*[1]*).*

- Minimize redundancy
  - reduce opportunities for inconsistencies (typically from updates)
  - reduce storage
  - considered as an anomaly
  - NOTE: minimize does not mean eliminate (e.g., keys are redundant)
- Eliminate anomalies
- Facilitate multiple table queries

Normalization Forms

Except for the first normal form (`1NF`), all normal forms based on functional dependencies. Because higher normal forms deal with situations extremely rare[2], we only discuss the first three normal forms. In addition, there are trade-offs between data redundancy and performance. Normalizing reduces redundancy at the expense of performance. Therefore, occasionally, we **denormalize** database tables to reduce the number of *joins*.

[1] Needs of the organization

[2] Analytic (statistical) research applications beyond the scope of business operations.

A database is *normalized* if it is in third-normal form. A relation not normalized is said to be in *unnormalized form* (`UNF`). Normal forms are executed as a series of steps.

1. `1NF`: Each row is unique[3], no repeated columns, and all cells are atomic.
2. `2NF`: In `1NF` and contains no partial dependencies.
3. `3NF`: In `2NF` and contains no transitive dependencies.

First Normal Form

Each row is unique, no repeated columns, and all cells are atomic.

Example 2. *Identify why the given table is unnormalized.*

Table 2: Non-normalize Table

Students =

Name	Class 1	Class 2	Class 3
Norville Rogers	MAT480	MAT405	ART101
Fred Jones	MAT190	ENV104	BIO110
Daphne Blake	MAT190	ENV104	

Example 3. *Identify why the given table is unnormalized.*

Table 3: Non-normalize Table. Note: this may be a good spreadsheet design and great view for a report.

Students =

ID	Name	Class	Rank	Hours
1	Norville Rogers	MAT190 ENV104 BIO110	Freshman	18
2	Fred Jones	MAT480 MAT405 ART101	Senior	101
3	Daphne Blake	MAT190 ENV104	Freshman	25

[3] i.e., Key exists

Table 4: Table in `1NF`. **Note**: There still exists many update anomalies and data redundancy.

StudentID	First	Last	Class	Description	Rank	Hours
1	Norville	Rogers	MAT190	Calculus 1	Freshman	18
1	Norville	Rogers	ENV104	EnvSci.	Freshman	18
1	Norville	Rogers	BIO110	Biology	Freshman	18
2	Fred	Jones	MAT480	Math Research	Senior	101
2	Fred	Jones	MAT405	Numerical Analysis	Senior	101
2	Fred	Jones	ART101	Drawing	Senior	101
3	Daphne	Blake	MAT190	Calculus 1	Freshman	25
3	Daphne	Blake	ENV104	EnvSci.	Freshman	25

Second Normal Form

Table must be in `1NF` and no partial dependencies (i.e., every non-key column must depend on the entire key). In Table 4, (`StudentID` , `Class`) is a primary key. However, `StudentID` $\rightarrow$ `First` is a partial dependency. That is, `First` is depends on `StudentID`. In fact, so does `Last` and `Rank`. Said another way, `Rank`, for example, does NOT depend on the entire key (only part of the key). In addition, `Description` only depends on `Class` instead of the entire key. We decompose table into two tables for which there are no partial dependencies.

Table 5: Table in `2NF`. **Note**: reduced redundancy and all non-key columns (i.e., `First, Last, Rank, Hours` depend entirely on `StudentID`.

Students =

StudentID	First	Last	Rank	Hours
1	Norville	Rogers	Freshman	18
2	Fred	Jones	Senior	101
3	Daphne	Blake	Freshman	25

Note: It is important to point out that functional dependencies are not particular to an instance (i.e., the Students relation above), but a general property (constraint) on the schema. For example, in the Students relation above, technically `First` $\rightarrow$ `Last`, but actually we should not expect this dependency to hold. There are many students that have same first names and different last names or same first and last names.

Table 6: Table not in `2NF`. What is the key? `Class, StudentID`. Note that `Description` is dependent on only part of the key (i.e., `Class`).

Classes =

Class	Description	StudentID
MAT190	Calculus 1	1
MAT190	Calculus 1	3
ENV104	EnvSci.	1
ENV104	EnvSci.	3
BIO110	Biology	1
MAT480	Math Research	2
MAT405	Numerical Analysis	2
ART101	Drawing	2

Table 7: Table in `2NF`. What is the key?

Classes =

Class	Description
MAT190	Calculus 1
ENV104	EnvSci.
BIO110	Biology
MAT480	Math Research
MAT405	Numerical Analysis
ART101	Drawing

Third Normal Form

Remove transitive dependencies (non-key columns depend ONLY on key).

Table 8: Table in `3NF`. What is the key?

Student_Classes =

Class	StudentID
MAT190	1
MAT190	3
ENV104	1
ENV104	3
BIO110	1
MAT480	2
MAT405	2
ART101	2

Table 9: Normalize

TID	Tname	Dept	Building	Baddr	Office	SID	Sname	Course
1	Adams	MAT	A	12 Broadway	200	1	Jones	MAT190
1	Adams	MAT	A	12 Broadway	200	2	Rogers	MAT190
1	Adams	MAT	A	12 Broadway	200	3	Blake	MAT195
2	Bashforth	MAT	A	12 Broadway	201	3	Blake	MAT200
2	Bashforth	MAT	A	12 Broadway	201	4	King	MAT200
2	Bashforth	MAT	A	12 Broadway	201	5	Snow	MAT201
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3	Carrie	HIS	B	11 Hill	111	5	Snow	HIS101
3	Carrie	HIS	B	11 HILL	111	4	King	HIS101
3	Carrie	HIS	B	11 Hill	111	5	Fisher	HIS102
3	Carrie	HIS	B	11 Hill	111	5	Ramsey	HIS102
4	Adams	HIS	B	11 Hill	200	6	Flowers	HIS101
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

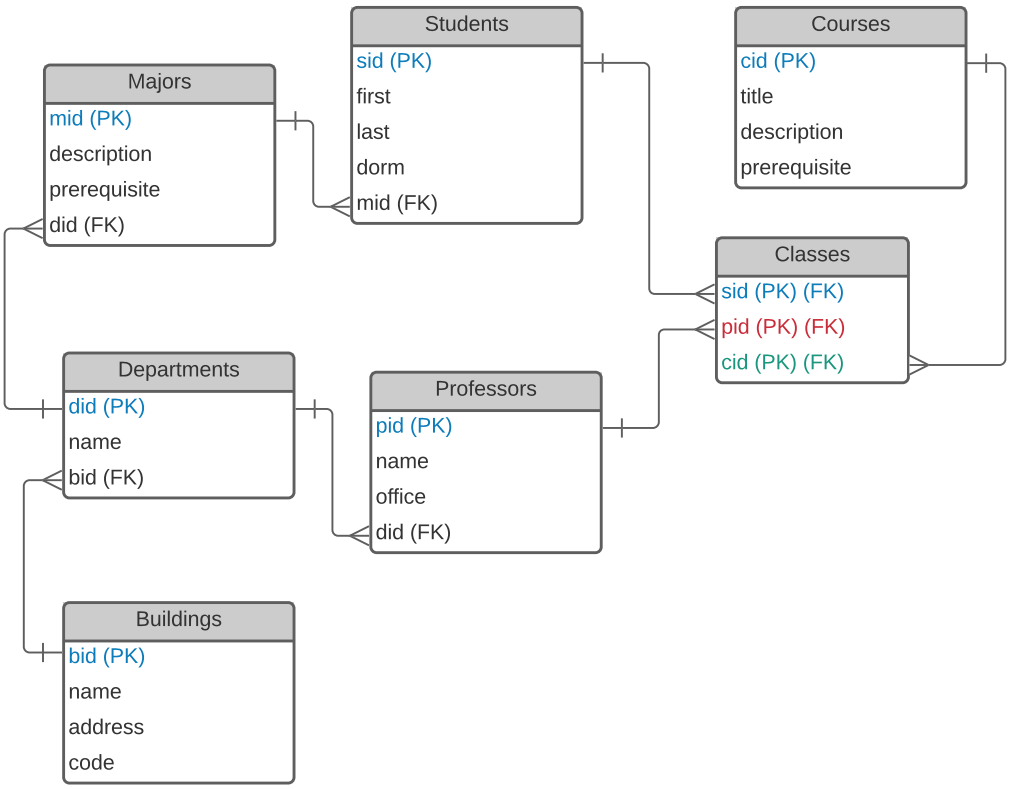

Figure 1: Teachers - Students - Classes