Databases Design. Introduction to SQL

LECTURE 4

Normalization. Normal Forms

Last lecture

Update anomalies

- Modification
- Insertion
- Deletion

Functional dependencies

- Full
- Partial
- Transitive

Today’s lecture

- We'll speak about normalization and normal forms.
- You'll know what a normal form corresponds to a “good” design
- We'll discuss how we can improve a database design

Database Design Stages

1. Subject Area Analysis
2. Conceptual Design
3. Logical Design
4. Physical Design

Normalization

- **Normalization** is used to test the correctness of a logical data model. The logical data model is a source of information for the next phase - Physical database design.

- **Normalization** – is a process of decomposing complex relations into simple relations in order to remove unwanted functional dependencies and data redundancy within complex relations.

Normal Forms

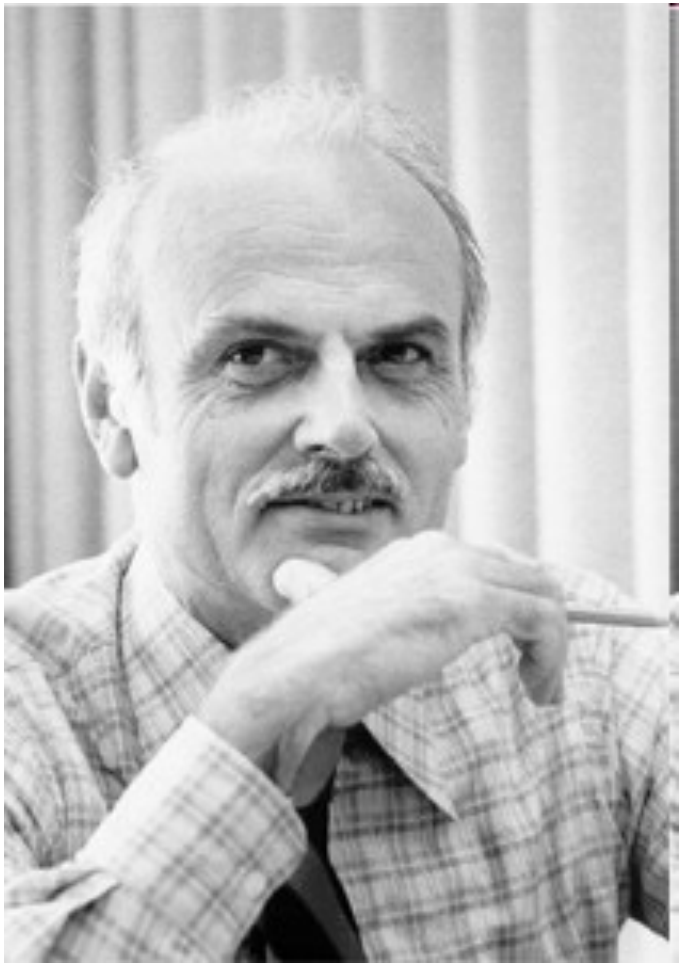
- The **Normal Forms (NF)** of relational database theory provide criteria for determining a table's degree of immunity against logical inconsistencies and anomalies.

- The normal forms are applicable to individual tables; to say that an entire database is in normal form *n* is to say that all of its tables are in normal form *n*.

History

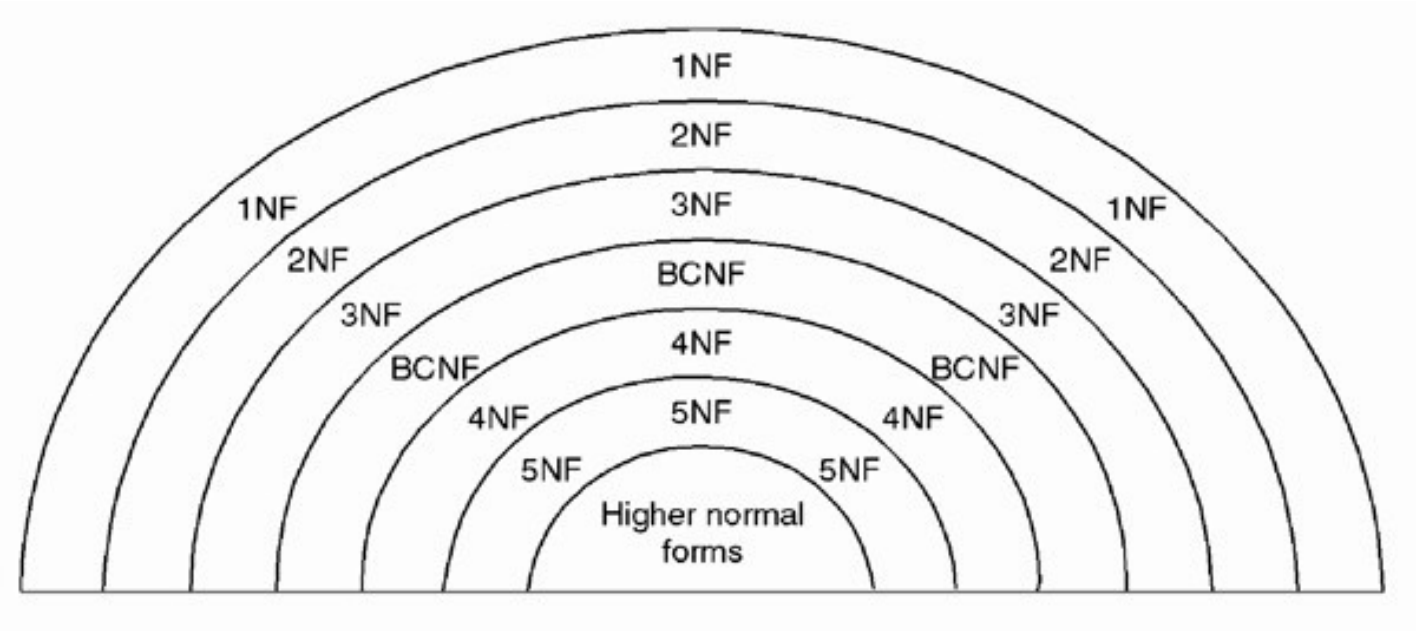
Edgar F. Codd, the inventor of the relational model, introduced the concept of normalization and what we now know as the First Normal Form (1NF) in 1970.

Codd went on to define the Second Normal Form (2NF) and Third Normal Form (3NF) in 1971, and Codd and Raymond F. Boyce defined the Boyce-Codd Normal Form (BCNF) in 1974.



Normal Forms

- Relational tables are classified into various normal forms based on the existence of various types of functional dependencies.
- Types of functional dependencies leads to over five normal forms.



“Good” design

- Normalization is executed as a series of steps. Each step corresponds to a specific normal form that has known properties.
- As normalization proceeds, the relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.
- 3NF is the standard normal form that a relation may be in to be considered a “good” design.
- 3NF tables are free of insertion, modification (update), and deletion anomalies.

First Normal Form (1NF)

- Relation is in **First Normal Form (1NF)** if each column is a single, atomic value.
- Relation is in 1NF if every cell in the table contains one and only one value.
- Relation that is not in 1NF is known as **unnormalized** or **UNF (0NF)**. A relation that is in UNF will have composite of multi-valued attributes.

First Normal Form

- The table below stores teachers and departments information.
- This table is in 1NF because every attribute is atomic.

Teacher_id	Last_name	Dep_id	Dep_name	Room
001	Teacher1	001	CET	409
002	Teacher2	001	CET	409
003	Teacher3	002	IS	803

First Normal Form

- Consider the relation below.
- It is in UNF (not in 1NF).

Teacher_id	Last_name	Degree
001	Teacher1	{M.D., PhD}
002	Teacher2	{M.D.}
003	Teacher3	{M.D., PhD}

Second Normal Form (2NF)

- Relation that is in **Second Normal Form (2NF)** is in 1NF and has no partial dependencies on the PK.

- Second normal form is associated with **modification anomaly**.

Partial dependency

- A dependency X -> Y is a **partial dependency** if there exists an attribute A that is part of X that can be removed from X and the dependency still holds.
- Example: Teachers and Courses table.
- PK is {teacher_id, course_id}

Teacher_id	Last_name	Course_id	Course_name	Credits
001	Teacher1	001	SDP1	3
001	Teacher1	002	SDP2	3
002	Teacher2	001	SDP1	3

Partial dependency

- Also suppose this relation has the following dependencies:

  FD1: {teacher_id, course_id} -> {last_name, course_name, credits}

  FD2: {teacher_id} -> {last_name}

  FD3: {course_id} -> {course_name, credits}

- FD1 is a partial dependency
- FD2 and FD3 are full dependencies.

Third Normal Form (3NF)

- Relation is in **Third Normal Form (3NF)** if it is in 2NF and it contains no attributes that are transitively dependent on the Primary Key.

- Third normal form is defined in terms of transitive dependencies and is associated with **insertion and deletion anomalies**.

Transitive dependency

- **Transitive dependency** is a condition where X, Y, and Z are attributes of a relation such that if X→Y and Y→Z, then Z is transitively dependent on X via Y.
- Example: Students and Groups table.
- PK is student_id.

Student_id	Last_name	Group_id	Group_name
001	Student1	001	Group1
002	Student2	001	Group1
003	Student3	002	Group2

Transitive dependency

- The relation has the following functional dependencies:

  {student_id} -> {last_name, group_id, group_name}

  {group_id} -> {group_name}
- This relation contains a transitive dependency because

  {student_id} -> {group_id} -> {group_name}

Decomposition

- **Decomposition** is the process of breaking down in parts or elements.
- It breaks the table into multiple tables in a database.

- It should always be lossless, because it confirms that the information in the original relation can be accurately reconstructed based on the decomposed relations.

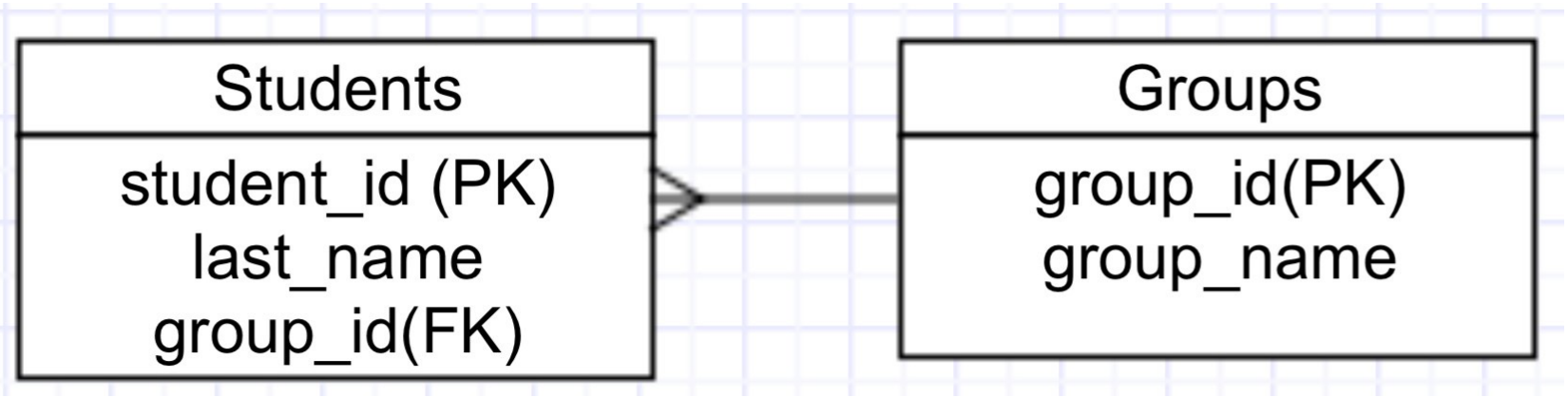
Normalization

- Let's normalize two of the relations that we have looked at.
- First, Students and Groups relation.

Student_id	Last_name	Group_id	Group_name
001	Student1	001	Group1
002	Student2	001	Group1
003	Student3	002	Group2

Normalization

- We said this relation contains a transitive dependency, and, thus, violates third normal form. We need to remove the transitive dependency from the table.
- We do this by breaking the relation into two tables: Students and Groups.



Normalization

- Now, Teachers and Courses table.
- PK of this relation was {teacher_id, course_id}.

Teacher_id	Last_name	Course_id	Course_name	Credits
001	Teacher1	001	SDP1	3
001	Teacher1	002	SDP2	3
002	Teacher2	001	SDP1	3

Normalization

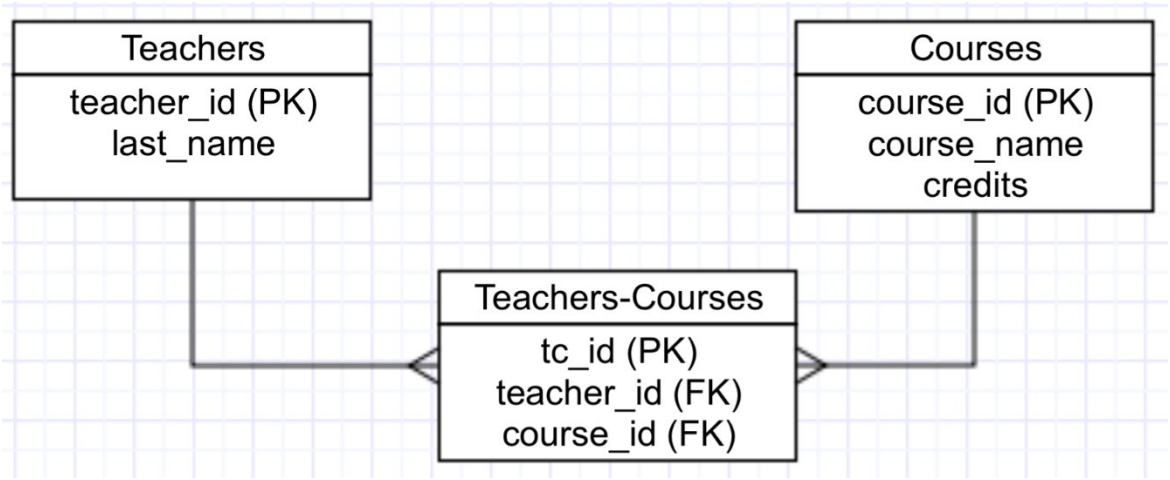
- Also, recall that the FDs for this relation were

  FD1: {teacher_id, course_id} -> {last_name, course_name, credits}

  FD2: teacher_id -> last_name

  FD3: course_id -> course_name, credits

- Because of this relation's partial dependencies, it violates 2NF. To bring it to 2NF, we create three relations.

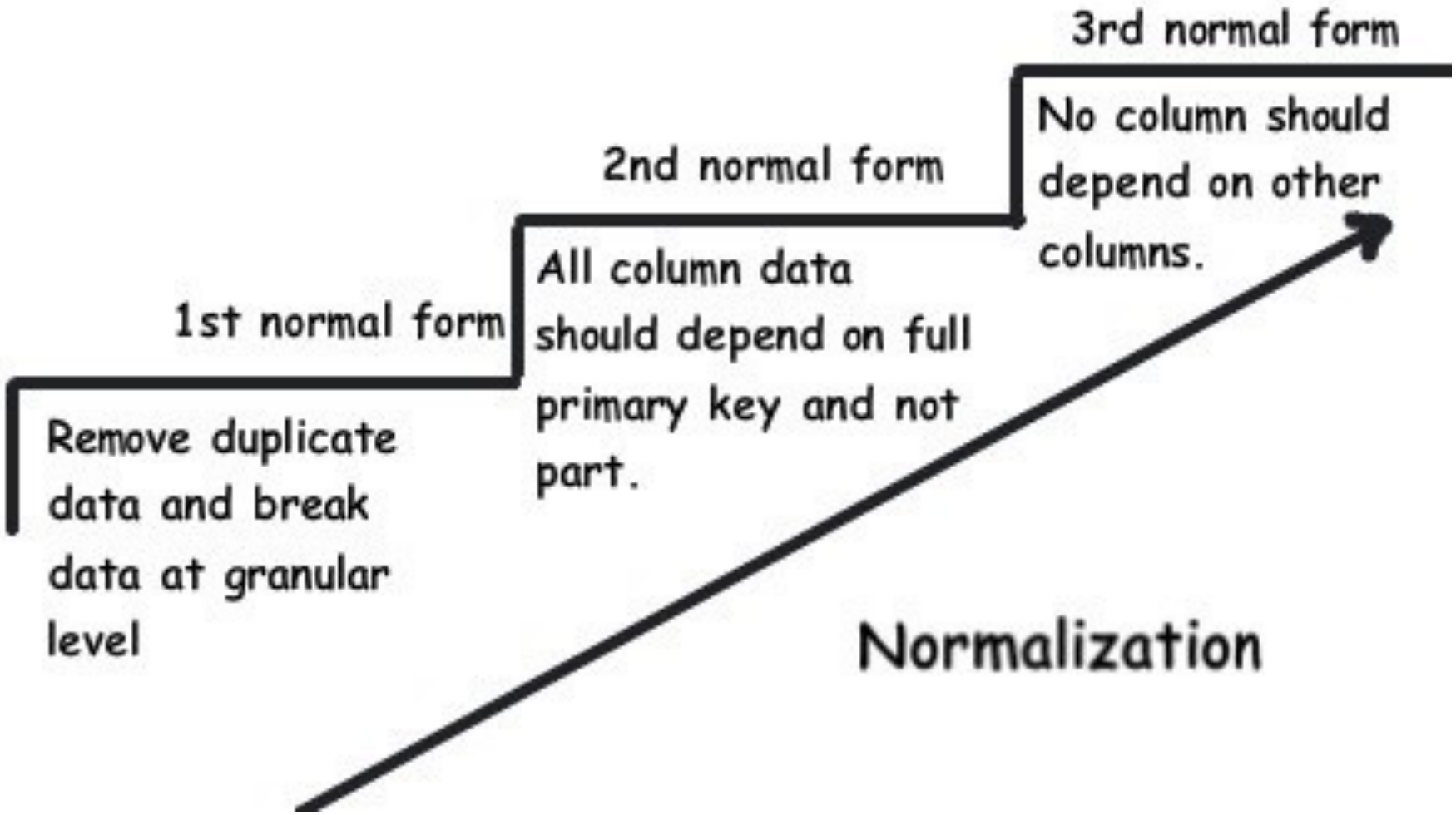


Summary

- Typically, a 3NF relation is considered a good database design – the normalizing process usually ends once 3NF is attained.

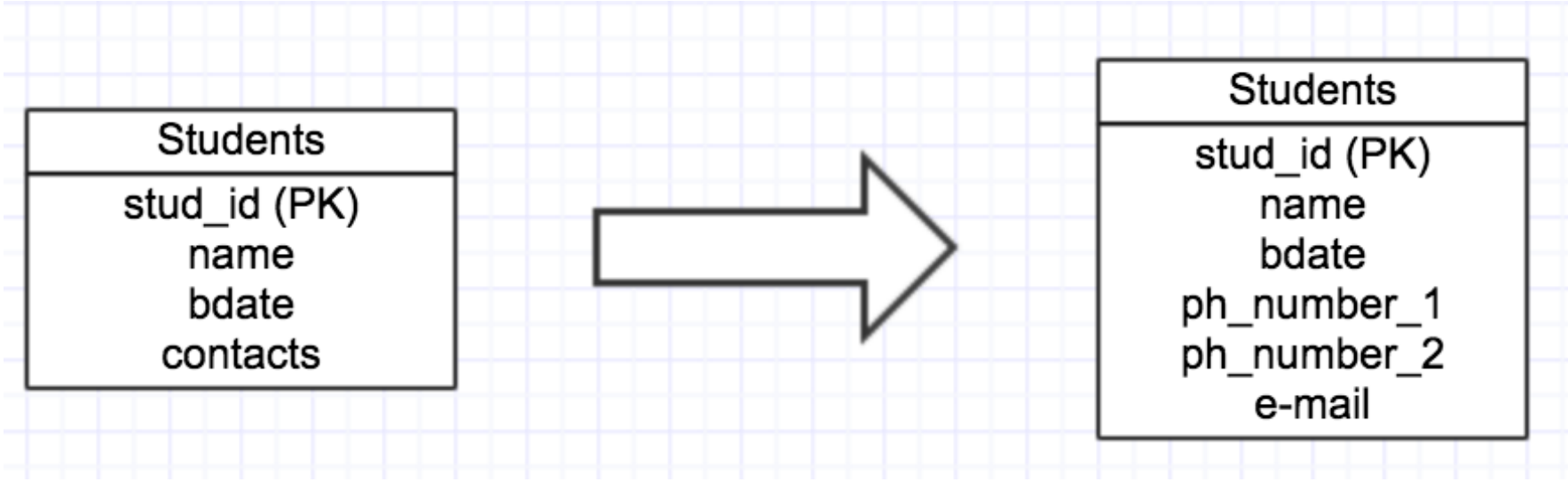
- How do we remove functional dependencies that violate a given normal form?
- We break a relation into many smaller relations while still maintaining the relationship among the data.

Summary


1st normal form
Remove duplicate data and break data at granular level
2nd normal form
All column data should depend on full primary key and not part.
3rd normal form
No column should depend on other columns.
Normalization

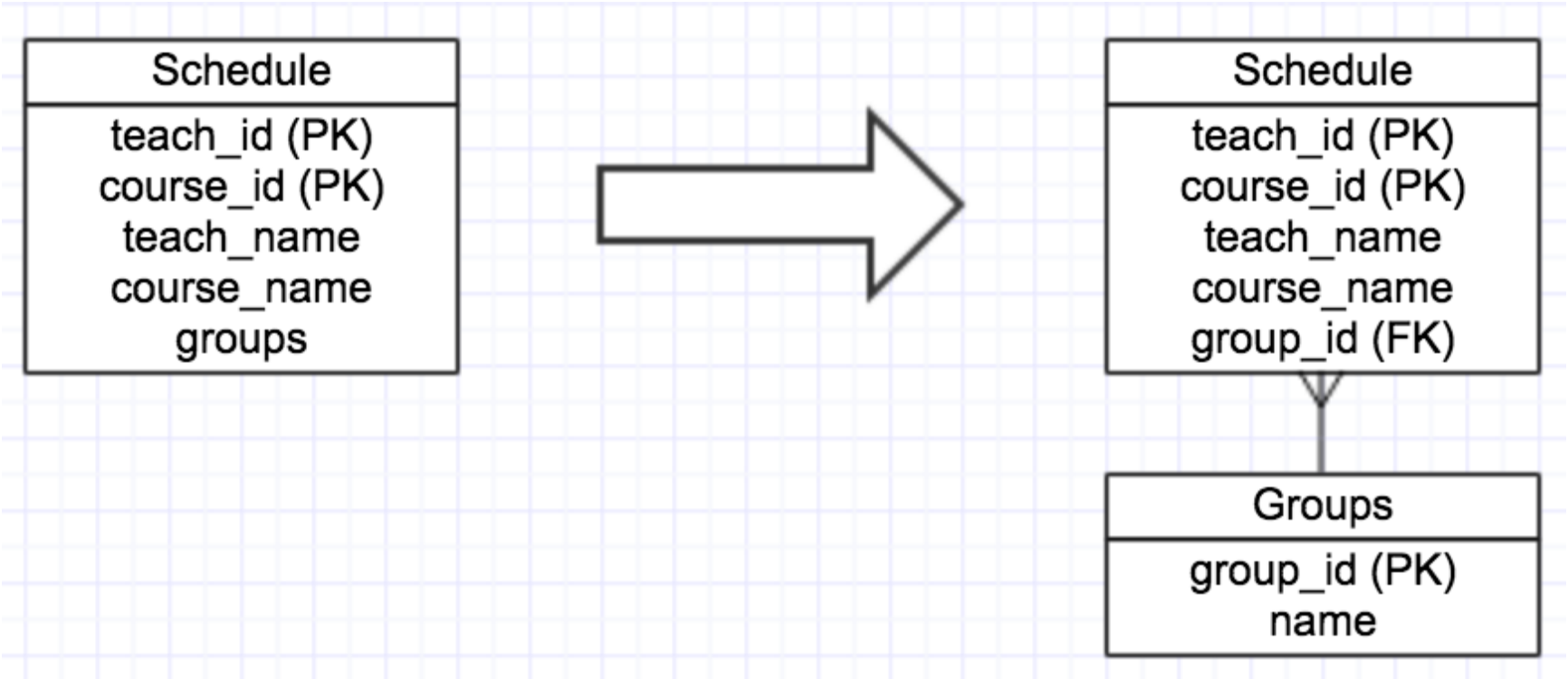
Examples for CourseWork.Part 3

Case 1: UNF to 1NF (vers. 1)

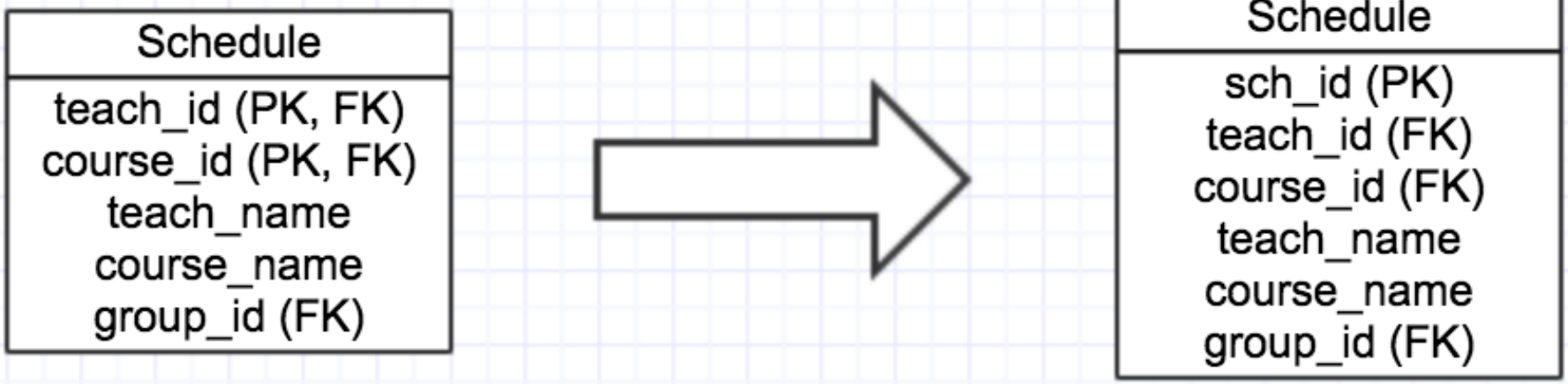


Contacts was divided like ph_number_1, ph_number_2, e-mail

Case 1: UNF to 1NF (vers. 2)



Case 2: 1NF to 2NF



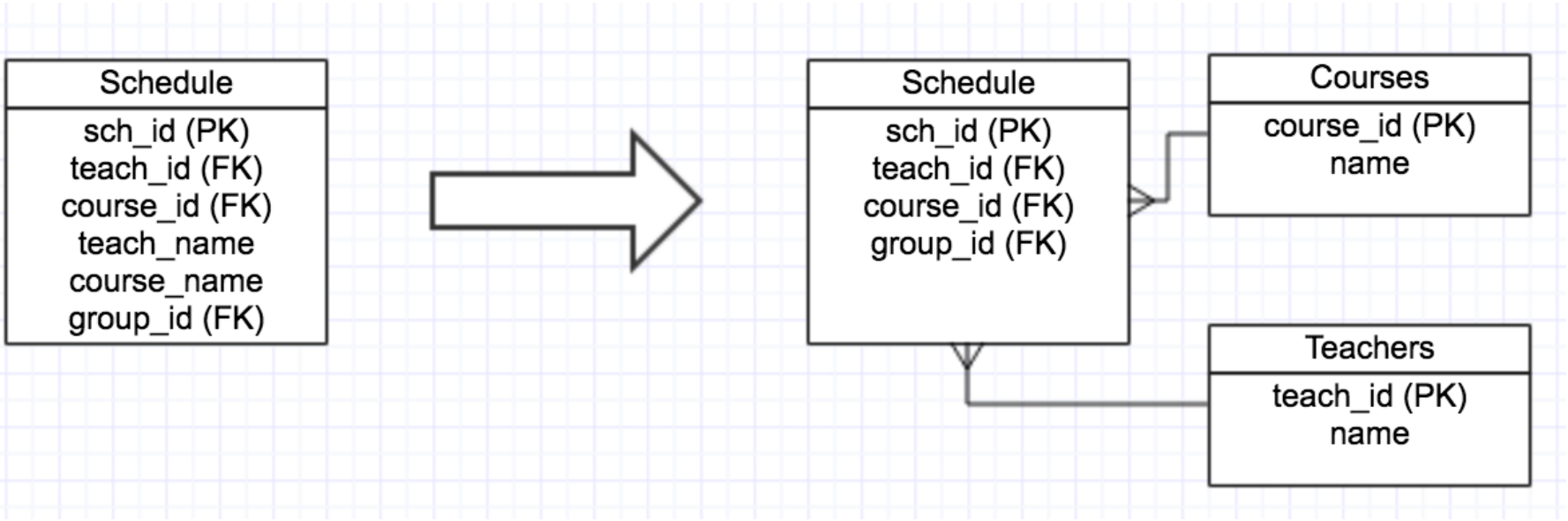
Before:

FD1: {teach_id, course_id} -> {teach_name, course_name, group_id}

FD2: {teach_id} -> {teach_name}

FD3: {course_id} -> {course_name}

Case 3: 2NF to 3NF



Before:

FD1: {sch_id} -> {teach_id, course_id, teach_name, course_name, group_id}

FD2: {teach_id} -> {teach_name}

FD3: {course_id} -> {course_name}

Books

- **Connolly, Thomas M. Database Systems**: A Practical Approach to Design, Implementation, and Management / Thomas M. Connolly, Carolyn E. Begg.- United States of America: Pearson Education

- **Garcia-Molina, H. Database system**: The Complete Book / Hector Garcia-Molina.- United States of America: Pearson Prentice Hall

- **Sharma, N. Database Fundamentals**: A book for the community by the community / Neeraj Sharma, Liviu Perniu.- Canada

- E.F. Codd, "Further Normalization of the Data Base Relational Model"

Question

Through normalization, update anomalies

a) can be maximized
b) can be eliminated
c) is usually left unchanged
d) can be minimized but not eliminated

Question

Which of the following statements concerning normal forms is true?

a) A relation that is in second normal form is also in first normal form.
b) Each normal form contains a state of independent properties, unrelated to other normal forms.
c) The lower the normal form number, the better the schema design is.
d) Schemas that are in second normal form are considered the best.

Question

For a relation to be in 3NF, it should not contain _____ attribute that is transitively dependent on _____.

a) a non-primary key, the primary key
b) a primary key, a non-primary key
c) a primary key, a foreign key
d) a non-primary key, a foreign key

Question

Consider a table with atomic attributes A, B, and C and the following functional dependencies.

A -> B

B -> C

If the primary key of this table is attribute A, then this relation satisfies which of the following normal forms?

1. First
2. Second
3. Third

a) I and II only
b) I, II and III
c) I only
d) None