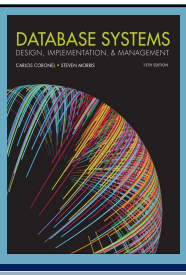


Chapter 6
Normalization of Database Tables

Learning Objectives

- After completing this chapter, you will be able to:
  - Explain normalization and its role in the database design process
  - Identify and describe each of the normal forms: 1NF, 2NF, 3NF, BCNF, and 4NF
  - Explain how normal forms can be transformed from lower normal forms to higher normal forms
  - Apply normalization rules to evaluate and correct table structures
  - Identify situations that require denormalization to generate information efficiently
  - Use a data-modeling checklist to check that the ERD meets a set of minimum requirements

Database Tables and Normalization (1 of 2)

- Normalization: evaluating and correcting table structures to minimize data redundancies
  - Reduces data anomalies
- Normal forms
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

Database Tables and Normalization (2 of 2)

- Structural point of view of normal forms
  - Higher normal forms are better than lower normal forms i.e. 2NF is better than 1NF and 3NF is better than 2NF
  - Properly designed 3NF structures meet the requirement of fourth normal form (4NF)
- Denormalization: produces a lower normal form
  - Results in increased performance and greater data redundancy
  - E.g. a 3NF will be converted to 2NF

The Need for Normalization

- Used while designing a new database structure
  - Analyzes the relationship among the attributes within each entity
  - Determines if the structure can be improved through normalization
  - Improves the existing data structure and creates an appropriate database design
- The main goal of normalization is to eliminate data anomalies by eliminating unnecessary or unwanted data redundancies. For this, you need to put together all previous concepts learned such as
  - Identifying business rules
  - Identifying and defining business and data constraints
  - Defining functional dependencies
  - Identifying entities and relationships
  - Eliminating multivalued attributes

The Normalization Process (1 of 5)

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - Each table represents a single subject
  - Each row/column intersection contains only one value and not a group of values
  - No data item will be unnecessarily stored in more than one table
  - All nonprime attributes in a table are dependent on the primary key
  - Each table has no insertion, update, or deletion anomalies

The Normalization Process (2 of 5)

- Ensures that all tables are in at least 3NF
  - Higher forms are not likely to be encountered in business environment
- Works one relation at a time
  - Identifies the dependencies of a relation (table)
  - Progressively breaks the relation up into a new set of relations

The Normalization Process (3 of 5)

Table 6.2: Normal Forms	
Normal Form	**Characteristic**
First normal form (1NF)	Table format, no repeating groups, and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	3NF and no independent multivalued dependencies

The Normalization Process (4 of 5)

Table 6.3: Functional Dependence Concepts	
Concept	**Definition**
Functional dependence	The attribute B is fully functionally dependent on the attribute A if each value of A determines one and only one value of B. Example: PROJ_NUM $\rightarrow$ PROJ_NAME (read as PROJ_NUM functionally determines PROJ_NAME) In this case, the attribute PROJ_NUM is known as the determinant attribute, and the attribute PROJ_NAME is known as the dependent attribute.
Functional dependence (generalized definition)	Attribute A determines attribute B (that is, B is functionally dependent on A) if all (generalized definition) of the rows in the table that agree in value for attribute A also agree in value for attribute B.
Fully functional dependence (composite key)	If attribute B is functionally dependent on a composite key A but not on any subset of that composite key, the attribute B is fully functionally dependent on A.

The Normalization Process (5 of 5)

- Partial dependency: functional dependence in which the determinant is only part of the primary key
  - Assumption: one candidate key
  - Straight forward
  - Easy to identify
- Transitive dependency: attribute is dependent on another attribute that is not part of the primary key
  - More difficult to identify among a set of data
  - Occur only when a functional dependence exists among nonprime attributes

Conversion to First Normal Form (1NF) (1 of 3)

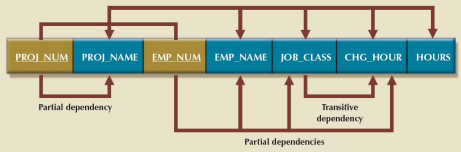
- Repeating group: group of multiple entries of same type can exist for any single key attribute occurrence
  - Reduces data redundancies
- Three step procedure
  - Eliminate the repeating groups
  - Identify the primary key
  - Identify all dependencies
- Dependency diagram: depicts all dependencies found within given table structure
  - Helps to get an overview of all relationships among table's attributes
  - Makes it less likely that an important dependency will be overlooked

Conversion to First Normal Form (1NF) (2 of 3)

- 1NF describes tabular format in which:
  - All key attributes are defined
  - There are no repeating groups in the table
  - All attributes are dependent on the primary key
- All relational tables satisfy 1NF requirements
- Some tables contain partial dependencies
  - Update, insertion, or deletion

Conversion to First Normal Form (1NF) (3 of 3)

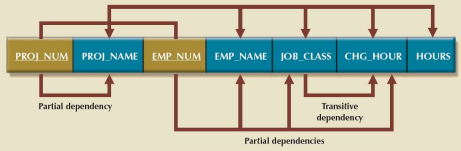
FIGURE 6.3 FIRST NORMAL FORM (1NF) DEPENDENCY DIAGRAM



Primary key attributes: PROJ_NUM, EMP_NUM
Arrows above the attributes indicate all desirable dependencies i.e. dependencies based on the primary key. In this case they are dependent on the combination of PROJ_NUM and EMP_NUM.
The arrows below the diagram indicates less desirable dependencies (partial and transitive)

Conversion to First Normal Form (1NF)

FIGURE 6.3 FIRST NORMAL FORM (1NF) DEPENDENCY DIAGRAM



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:
(PROJ_NUM → PROJ_NAME)
(EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR)

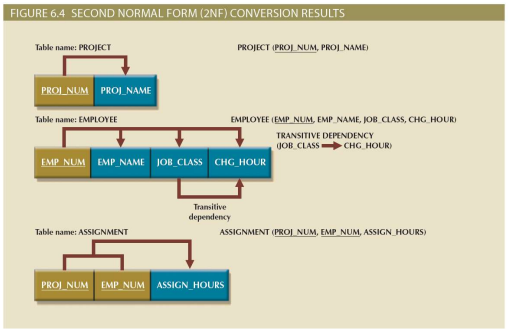
TRANSITIVE DEPENDENCY:
(JOB_CLASS → CHG_HOUR)

Conversion to Second Normal Form (2NF) (1 of 2)

- Conversion to 2NF occurs only when the 1NF has a composite primary key
  - If the 1NF has a single-attribute primary key, then the table is automatically in 2NF
- The 1NF-to-2NF conversion is simple
  - Make new tables to eliminate partial dependencies
  - Reassign corresponding dependent attributes
- Table is in 2NF when it:
  - Is in 1NF
  - Includes no partial dependencies

Conversion to Second Normal Form (2NF) (2 of 2)

FIGURE 6.4 SECOND NORMAL FORM (2NF) CONVERSION RESULTS



CENGAGE

16

Conversion to Third Normal Form (3NF) (1 of 2)

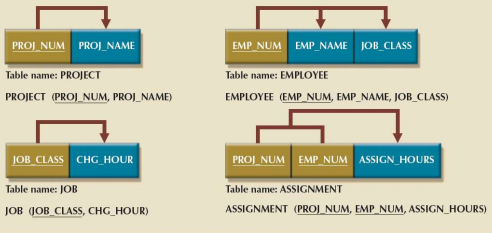
- The data anomalies created by the database organization shown in Figure 6.4 are easily eliminated
  - Make new tables to eliminate transitive dependencies
  - Reassign corresponding dependent attributes
- Table is in 3NF when it:
  - Is in 2NF
  - Contains no transitive dependencies

CENGAGE

17

Conversion to Third Normal Form (3NF) (2 of 2)

FIGURE 6.5 THIRD NORMAL FORM (3NF) CONVERSION RESULTS



CENGAGE

18

Improving the Design

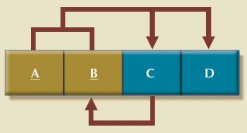
- Normalization is valuable because its use helps eliminate data redundancies
  - Evaluate PK assignments and naming conventions
  - Refine attribute atomicity
    - Atomic attribute: cannot be further subdivided
    - Atomicity: characteristic of an atomic attribute
  - Identify new attributes and new relationships
  - Refine primary keys as required for data granularity
    - Granularity: Level of detail represented by the values stored in a table's row
  - Maintain historical accuracy and evaluate using derived attributes

The Boyce-Codd Normal Form (1 of 4)

- Every determinant in the table should be a candidate key
  - Candidate key: same characteristics as primary key but not chosen to be the primary key
  - Equivalent to 3NF when the table contains only one candidate key
  - Violated only when the table contains more than one candidate key
  - Considered to be a special case of 3NF

The Boyce-Codd Normal Form (2 of 4)

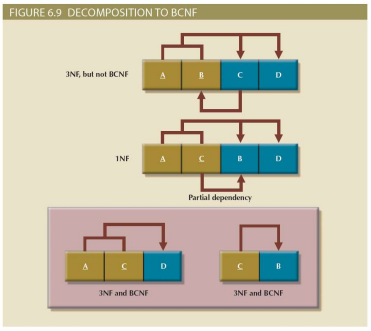
FIGURE 6.8 A TABLE THAT IS IN 3NF BUT NOT IN BCNF



- A + B ⟶ C, D
- A + C ⟶ B, D
- C ⟶ B

The Boyce-Codd Normal Form (3 of 4)

FIGURE 6.9 DECOMPOSITION TO BCNF



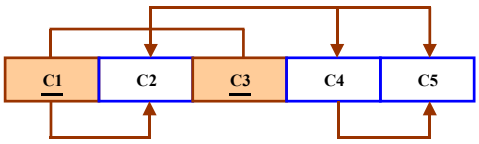
The Boyce-Codd Normal Form (4 of 4)

Table 6.5: Sample Data for a BCNF Conversion

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

- STU_ID + STAFF_ID ⟶ CLASS_CODE, ENROLL_GRADE
- CLASS_CODE ⟶ STAFF_ID

Test your knowledge



- **Create a database whose tables are at least in 2NF, showing the dependency diagrams for each table.**
- **Create a database whose tables are at least in 3NF, showing the dependency diagrams for each table.**

2NF

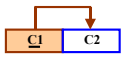


Table 1

Primary key: C1
Foreign key: None
Normal form: 3NF

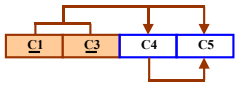


Table 2

Primary key: C1 + C3
Foreign key: C1 (to Table 1)
Normal form: 2NF, because the table exhibits the transitive dependencies C4 → C5

CENGAGE

25

3NF

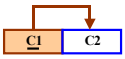


Table 1
Primary key: C1
Foreign key: None
Normal form: 3NF

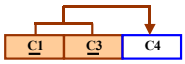


Table 2
Primary key: C1 + C3
Foreign key: C1 (to Table 1)
C4 (to Table 3)
Normal form: 3NF

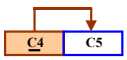


Table 3
Primary key: C4
Foreign key: None
Normal form: 3NF

CENGAGE

26

Fourth Normal Form (4NF)

- Rules
  - All attributes must be dependent on the primary key, but they must be independent of each other
  - No row may contain two or more multivalued facts about an entity
- Table is in 4NF when it:
  - Is in 3NF
  - Has no multivalued dependencies

CENGAGE

27

Summary

- Normalization is a technique used to design tables in which data redundancies are minimized
- A table is in 1NF when all key attributes are defined and all remaining attributes are dependent on the primary key
- A table is in 2NF when it is in 1NF and contains no partial dependencies
- A table is in 3NF when it is in 2NF and contains no transitive dependencies
- A table that is not in 3NF may be split into new tables until all of the tables meet the 3NF requirements
- Normalization is an important part—but only a part—of the design process
- A table in 3NF might contain multivalued dependencies that produce either numerous null values or redundant data