

LESSON 4.1

98-364 Database Administration Fundamentals

Understand Normalization

Lesson Overview

4.1. Understand normalization

In this lesson, you will review:

- Normalization
- Normal Form
 - 1NF—First Normal Form
 - 2NF—Second Normal Form
 - 3NF—Third Normal Form
 - 4NF—Fourth Normal Form
 - 5NF—Fifth Normal Form

Normalization

- Applying a body of techniques to a relational database to minimize the inclusion of duplicate information.
- Normalization greatly simplifies query and update management, including security and integrity considerations, although it does so at the expense of creating a larger number of tables.
- Note: Having student_id data in more than one table is not redundant data. It is required for the related dependency.

Why use normalization?

- It reduces the amount of storage space a database uses and ensures that data is logically stored.

Normal Form

- An approach to structuring (organizing) information to avoid redundancy and inconsistency and to promote efficient maintenance, storage, and updating.
- Several “rules” or levels of normalization are accepted, each a refinement of the preceding one.
- Three forms are commonly used:
 - — First Normal Form (1NF)
 - — Second Normal Form (2NF)
 - — Third Normal Form (3NF)

1NF—First Normal Form

- Groups of records (such as Student_info) in which each field (column) contains unique value and no repetitive information.
- This form is the least structured of the forms.
- Basic rules for a database:
 - — Eliminate duplicative columns from the same table
 - — Create separate tables for each group of related data and identify each row with a unique column (the primary key).

1NF—First Normal Form (Continued)

- Table Teacher_classes: Teacher, class1, class2, class3, class4, class5, class6
- Is NOT 1NF. In this table, we have duplicated columns:

Teacher	Class1	Class2	Class3
Jones	Eng123	Eng456	Lit100

- Class1, Class2, Class3 are copies of each other (duplicated columns).
- 1NF Example: Table Teacher_classes Teacher_ID and Class_ID:

Teacher ID	Class_Id
78v22	12567

- In this table, we have no duplicated columns. In this table form, we have Unique IDs.

2NF—Second Normal Form

- 2NF more aggressively removes duplicative data.
- 2NF adds new rules which do the following:
 - — Meet all the requirements of the 1NF
 - — Remove subsets of data that apply to multiple rows of a table and place them in separate tables
 - — Create relationships between these new tables and their predecessors through the use of foreign keys

2NF—Second Normal Form (Continued)

- Example of *Student_Address* table before and after 2NF:

Before 2NF *Student_Address Table*

FirstName	LastName	Address	City	State	ZIP
-----------	----------	---------	------	-------	-----

After 2NF *Student_Address Table*

FirstName	LastName	Address	ZIP
-----------	----------	---------	-----

And ZIP Table (new table added)

City	State	ZIP
------	-------	-----

- In this 2NF, we no longer store city, state, and zip information with every student. This results in a large storage saving when you have 25,000 students.

3NF—Third Normal Form

- 3NF goes one step further.
- Meets all the requirements of 2NF plus
 - — Removes columns that are not dependent upon the primary key.
- Any column that is not directly needed is removed from the table.
- An example would be a tax on sales at a bookstore. Taxes are a calculated transaction, not a fixed amount tied to the student. If the tax rate changes, we need to change the tax rate in only one spot, not in thousands of records.

3NF—Third Normal Form (Continued)

Example: Before and after 3NF:

Before 3NF: Bookstore sales table

Student-ID	Items_Id	Quantity	Sub-Total	Tax	Total
------------	----------	----------	-----------	-----	-------

After 3NF: Bookstore sales table

Student_ID	Items_Id	Quantity
------------	----------	----------

The table items `Sub_Total`, `Tax`, and `Total` are all calculated fields. Hence, the columns that are not dependent upon the primary key are removed for 3NF.

4NF—Fourth Normal Form

- Rarely used
- 4NF has one additional requirement:
 - — Meets all the requirements of 3NF *and* it has no multivalued dependencies

Multivalued dependencies

Observe this three-column table labeled A, B, and C. For every value of A, we have respective values for B and C. B and C are independent of each other.

A	B	C
Mary	Moon	Potatoes
John	Sports cars	Beans sprouts

5NF—Fifth Normal Form

- Rarely used
- Also known as join-projection normal form (JPNF), states that no nontrivial join dependencies exist.
- Any fact should be able to be reconstructed without any anomalous results in any case, regardless of the number of tables being joined.
- Should have only candidate keys and its primary key should consist of only a single column.
- Problem
 - The size of the joins that are required to reconstruct any nontrivial data. Views and procedures can simplify them, but the underlying data still ends up very complex.
- Performance issues to consider—4NF and 5NF are often academic. In most cases, 3NF is the state normalization for databases.

5NF—Fifth Normal Form (Continued)

- Only in rare situations does a 4NF table not conform to 5NF. These are situations in which a complex real-world constraint governing the valid combinations of attribute values in the 4NF table is not implicit in the structure of that table.
- If such a table is not normalized to 5NF, the burden of maintaining the logical consistency of the data within the table must be carried partly by the application responsible for insertions, deletions, and updates to it; and there is a heightened risk that the data within the table will become inconsistent.
- 5NF design excludes the possibility of such inconsistencies.

In general, it is the best practice to keep your database in 3NF.

Lesson Review

1. What is normalization?
2. Why use normalization?
3. What forms are typically used and why?