

Lesson 6: Normalization



Database Normalization

- This is the process by which table structures are reviewed and correct to minimize data redundancy and reduce the chance of a data anomolous data.
- The higher the normal form, the better the design.
- Denormalization is the inverse of the process and is typically done to present data and information to end users or non-technical users.

Database Normalization Continued

- Each table represents a single subject.
- No data will be unnecessarily stored in more than one table.
- All nonkey attributes are functionally dependent, rather all attributes are determined by the value of the primary key.
- Tables are void of insert, update and delete data anomalies ensuring data integrity and consistency.
- Generally we strive for third normal form (3NF).

Normal Forms

Normal Form	Characteristic
First Normal Form (1NF)	Table format, no repeating groups, primary key identified.
Second Normal Form (2NF)	1NF and no partial dependencies
Third Normal Form (3NF)	1NF and no transitive dependencies
Boyce–Codd normal Form (BCNF)	Every determinant is a candidate key (special case of 3NF)
Fourth Normal Form	3NF and no independent multivalued dependencies

Table 6.2 in the book (page 207)

Revisting Functional Dependence (Lesson 3)

- A **key** consists of one or more attributes of a table (aka relation).
 - Keys are based on determination: The value of A can be used to look up the value of B. OR If someone knows my Employee ID, they can look up my First Name.
 - Relationships based on determination are known as **Functional Dependence**.

Functional Dependence Notation

Breaking Down the Notation
player_id → first_name
DETERMINANT → DEPENDENT
The player_id functionally determines first_name
The first_name is functionally dependent on player_id

Dependencies

- **Partial Dependency** - a condition in which an attribute is dependent on only a subset of the primary key.
- **Transitive Dependency** - a condition in which one attribute is dependent on another attribute that isn't part of the primary key.
- We're going to use dependency diagrams to visualize our data dependencies.

First Normal Form (1NF)

- First, make sure that the data is in table format, with rows and columns.
- Next check for repeating groups, that is multiple rows exist for a single key occurrence.
- Identify the primary key, whether it's a single attribute or multiple attributes.
- Identify any dependencies.
- Let's look at an example.

Second Normal Form (2NF)

- We're going to handle any partial dependencies.
- A **partial dependency** occurs when an attribute is dependent on one key attribute (applicable to composite keys).
- Let's look at an example.

Third Normal Form (3NF)

- A **transitive dependency** occurs when an attribute is dependent on a non-key attribute.
- Eliminate the transitive dependencies by creating a new table and reassigning attributes.
- Let's keep going with our example.

Boyce-Codd Third Normal Form (BCNF)

- Every determinant in a table is a candidate key.
 - These candidate keys were not chosen as the primary key for one reason or another.
 - Only violated if a table contains more than one candidate key.

Normalization as Part of the Design

- We want to include the process of normalization within the design phase so that we don't back ourselves into a corner.
 - Create your data model and ERD, then normalize, then finalize and implement.
- DBMSs provide many advantages, so let's ensure we use normalization to reap the benefits.