Normalization

- Evaluating and correcting table structures to minimize data redundancies
- Reduces data anomalies
- Assigns attributes to tables based on determination
- Normal forms
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

work on one relation (table) at a time: identify dependencies, then 'normalize' - progressively break it down into smaller relations (tables), based on the dependencies we identify in the original relation so that "only the PK, the whole PK and nothing but the PK" acts as a determinant!

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.
Functional dependence (Generalized definition)	Attribute A determines attribute B if all 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.

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
- Improves the existing data structure and creates an appropriate database design

Conversion to First Normal Form

- **Repeating group**: Group of multiple entries of same type can exist for any single key attribute occurrence
  - Existence proves the presence of data redundancies
- Enable reducing data redundancies
- Steps
  - Eliminate the repeating groups
  - Identify the primary key
  - Identify all dependencies

Normalization Process

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - Each table represents a single subject
  - 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 is void of insertion, update, and deletion anomalies

wholly; nothing but

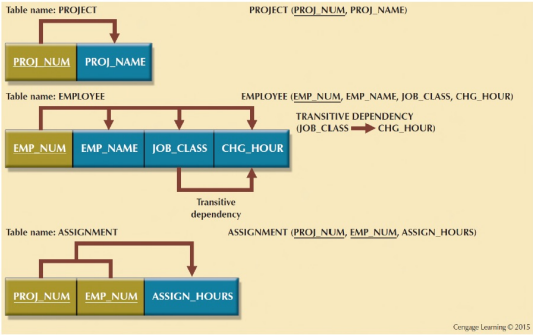
In other words, "fill in the blanks" so that there are no nulls. Now we have a relation (table), with a value in each cell.

Conversion to First Normal Form

- 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
    - Subject to data redundancies and various anomalies

Normalization is a systematic process that yields progressively higher 'normal forms' (NFs) for each entity (table) in our db. We want **at least** 3NF for each table; in RL, we stop **at** 3NF.

Figure 6.4 - Second Normal Form (2NF) Conversion Results



We eliminate partial dependencies by creating separate tables of such dependencies, and removing the dependent attributes from the starter table.

Denormalization

- Defects in unnormalized tables
  - Data updates are less efficient because tables are larger
  - Indexing is more cumbersome
  - No simple strategies for creating virtual tables known as views

Conversion to Third Normal Form

- Steps
  - Make new tables to eliminate transitive dependencies
    - **Determinant**: Any attribute whose value determines other values within a row
  - Reassign corresponding dependent attributes
- Table is in 3NF when it:
  - Is in 2NF
  - Contains no transitive dependencies

Denormalization

- Design goals
  - Creation of normalized relations
  - Processing requirements and speed
- Number of database tables expands when tables are decomposed to conform to normalization requirements
- Joining a larger number of tables:
  - Takes additional input/output (I/O) operations and processing logic
  - Reduces system speed