Chapter 6

Objectives: to learn
- What normalization is and what role it plays in the database design process
- About the normal forms 1NF, 2NF, 3NF, BCNF, and 4NF
- How normal forms can be transformed from lower normal forms to higher normal forms
- That normalization and ER modeling are used concurrently to produce a good database design
- That some situations require denormalization to generate information efficiently

Normalization:
- A process for assigning attributes to entities
- Reduces data redundancies
- Helps eliminate data anomalies
- Produces controlled redundancies to link tables
- Normal Forms are a series of stages done in Normalization
  - 1NF - First normal form,
  - 2NF - Second normal form,
  - 3NF - Third normal form,
  - 4NF - Fourth normal form

Database Tables & Normalization

Normal Forms (cont’)
- 2NF is better than 1NF; 3NF is better than 2NF
- For most business database design purposes, 3NF is as high as needed in normalization
- Denormalization produces a lower normal form from a higher normal form.
  - Highest level of normalization is not always most desirable
  - Increased performance but greater data redundancy

The Need for Normalization

Example: Company which manages building projects.
- The business rules are:
  - Charges its clients by billing hours spent on each contract
  - Hourly billing rate is dependent on employee’s position
- Periodically, report is generated that contains information such as displayed in Table 6.1
The Need for Normalization


Creating Entities from Tabular Data

- Structure of data set in Figure 6.1 does not handle data very well
  - Primary key - Project # contains nulls
  - Table displays data redundancies
- Report may yield different results depending on what data anomaly has occurred
  - Update - Modifying JOB_CLASS
  - Insertion - New employee must be assigned project
  - Deletion - If employee deleted, other vital data lost

The Normalization Process

- Relational database environment is suited to help designer avoid data integrity problems
  - 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, deletion anomalies
- Normalizing table structure will reduce data redundancies
The Normalization Process

- Objective of normalization is to ensure that all tables are in at least 3NF
- Normalization works one Entity at a time
- It progressively breaks table into new set of relations based on identified dependencies
- Normalization from 1NF to 2NF is three-step procedure.

<table>
<thead>
<tr>
<th>TABLE 6.2</th>
<th>Normal Forms</th>
<th>CHARACTERISTIC</th>
<th>SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1NF (First normal form)</td>
<td>Table format, no repeating groups, and PK identified</td>
<td>6.3.1</td>
<td></td>
</tr>
<tr>
<td>2NF (Second normal form)</td>
<td>1NF and no partial dependencies</td>
<td>6.3.2</td>
<td></td>
</tr>
<tr>
<td>3NF (Third normal form)</td>
<td>2NF and no transitive dependencies</td>
<td>6.3.3</td>
<td></td>
</tr>
<tr>
<td>BCNF (Boyce-Codd normal form)</td>
<td>Every determinant is a candidate key, special case of 3NF</td>
<td>6.3.4</td>
<td></td>
</tr>
<tr>
<td>4NF (Fourth normal form)</td>
<td>3NF and no independent multivalued dependencies</td>
<td>6.4.2</td>
<td></td>
</tr>
</tbody>
</table>

Conversion to First Normal Form

- Step 1: Eliminate the Repeating Groups
  - A Repeating group is group of multiple entries of same type existing for any single key attribute occurrence
  - Present data in tabular format, where each cell has single value and there are no repeating groups
  - Eliminate repeating groups, eliminate nulls by making sure that each repeating group attribute contains an appropriate data value. Repeating groups must be eliminated
- Step 2: Identify the Primary Key
  - Must uniquely identify attribute values
  - New key can be composed of multiple attributes
- Step 3: Identify All Dependencies
  - Dependencies are depicted with a diagram

Step 1: Conversion to 1NF

- Step 1: Eliminate the Repeating Groups
- Step 2: Identify the Primary Key
- Step 3: Identify All Dependencies

Step 1 - Eliminate the Repeating Groups

- A Repeating group is group of multiple entries of same type existing for any single key attribute occurrence
- Present data in tabular format, where each cell has single value and there are no repeating groups
- Eliminate repeating groups, eliminate nulls by making sure that each repeating group attribute contains an appropriate data value. Repeating groups must be eliminated
Step 2 - Conversion to 1NF

• Step 2 - Identify the Primary Key
  – Review (from Chapter 3) Determination and attribute dependence.
  – All attribute values in the occurrence are ‘determined’ by the Primary Key. The Primary Key Must uniquely identify the attribute(s).
  – Resulting Composite Key: PROJ_NUM and EMP_NUM

Step 3 - Conversion to 1NF

• Step 3 - Identify All Dependencies
  – Depicts all dependencies found within given table structure
  – Helpful in getting bird’s-eye view of all relationships among table’s attributes
    1. Draw desirable dependencies based on PKey
    2. Draw less desirable dependencies
      – Partial
      – Transitive

Resulting First Normal Form

• First normal form describes tabular format:
  – All key attributes are defined
  – No repeating groups in the table
  – All attributes are dependent on primary key
• All relational tables satisfy 1NF requirements
• Some tables contain other dependencies and should be used with caution
  – Partial dependencies - an attribute dependent on only part of the primary key
  – Transitive dependencies – an attribute dependent on another attribute that is not part of the primary key.
Conversion to Second Normal Form

- Step 1: Eliminate Partial Dependencies
  - Start with 1NF format and convert by:
    - Write each part of the composite key on its own line.
    - Write the original (composite) key on last line
  - Each component will become key in new table
- Step 2: Assign Dependent Attributes
  - From the original 1NF determine which attributes are dependent on which key attributes
- Step 3: Name the tables to reflect its contents & function

Completed Conversion to 2NF

- Each key component establishes a new table
- Table is in second normal form (2NF) when:
  - It is in 1NF and
  - It includes no partial dependencies:
    - No attribute is dependent on only portion of primary key
  - Note: it is still possible to exhibit transitive dependency
    - Attributes may be functionally dependent on nonkey attributes

Conversion to Third Normal Form

- Step 1: Eliminate Transitive Dependencies
  - Write its determinant as PK for new table.
  - And leave it in the original table
- Step 2: Reassign Corresponding Dependent Attributes
  - Identify attributes dependent on each determinant identified in Step 1, and list on new table.
- Step 3: Name the new table(s) to reflect its contents and function

Completed Conversion to 3NF
Resulting Third Normal Form

- A table is in third normal form (3NF) when both of the following are true:
  - It is in 2NF
  - It contains no transitive dependencies

Improving the Design

- Table structures should be cleaned up to eliminate initial partial and transitive dependencies
- Normalization cannot, by itself, be relied on to make good designs
- It reduces data redundancy and builds controlled redundancy.
- The higher the NF,
  - the more entities one has,
  - the more flexible the database will be,
  - the more joins (and less efficiency) you have.

Surrogate Key Considerations

- When primary key is considered to be unsuitable, designers use surrogate keys
- System-assigned primary keys may not prevent confusing entries, but do prevent violation of entity integrity.
- Example: data entries in Table 6.4 are inappropriate because they duplicate existing records

<table>
<thead>
<tr>
<th>Code</th>
<th>Job Description</th>
<th>Hourly Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>501</td>
<td>Programmer</td>
<td>$11.50</td>
</tr>
<tr>
<td>502</td>
<td>Programmer</td>
<td>$11.75</td>
</tr>
</tbody>
</table>
Improving the Design

- Identifying new attributes

Higher-Level Normal Forms

- Tables in 3NF perform suitably in business transactional databases
- Higher-order normal forms are useful on occasion
- Two special cases of 3NF:
  - Boyce-Codd normal form (BCNF)
  - Fourth normal form (4NF)

The Boyce-Codd Normal Form (BCNF)

- Every determinant in table is a candidate key
  - Has same characteristics as primary key, but for some reason, not chosen to be primary key
- When table contains only one candidate key, the 3NF and the BCNF are equivalent
- BCNF can be violated only when table contains more than one candidate key
  - example:
    Section(coursename, sectionno, courseno, time, days

The Boyce-Codd Normal Form (BCNF)

- Most designers consider the BCNF as a special case of 3NF
- Table is in 3NF when it is in 2NF and there are no transitive dependencies
- Table can be in 3NF and fail to meet BCNF
  - No partial dependencies, nor does it contain transitive dependencies
  - A nonkey attribute is the determinant of a key attribute
The Boyce-Codd Normal Form (BCNF)

- When part of the key is dependent on another non-key attribute, i.e. another candidate key.

- Non-Boyce-Codd Normal Form
  - Can only exist with a composite Primary Key
  - Example Enroll entity: Enroll(Stu_ID, Staff_ID, Class_Code, Enroll_Grade)

- Resulting BCNF with two entities
  - Enroll with composite PK Stu_ID & Class_code.
  - Class with Class_code as its PK.
Fourth Normal Form (4NF)

- Table is in fourth normal form (4NF) when both of the following are true:
  - It is in 3NF
  - No multiple sets of multivalued dependencies

- 4NF is largely academic if tables conform to following two rules:
  - All attributes dependent on primary key, independent of each other
  - No row contains two or more multivalued facts about an entity

Two Examples of multi-valued dependencies

- StudentID, StName, Phones (Home, Work, Cell, Fax)
- StudentID, Addresses (permanent, mailing, current)

Convert multi-valued phones using two additional tables in 3NF

- Student (StudentID, StName, .........)
- StuPhones (StudentID, PhoneType, Phone#)
- Phones (PhoneType, Description)

Example: Tracking employee’s volunteer service

Conversion tables for volunteer service

Denormalization

- Creation of normalized relations is important database design goal
- Processing requirements should also be a goal
- If tables are decomposed to conform to normalization requirements:
  - Number of database tables expands
  - Causing additional processing
  - Loss of system speed
Denormalization

- Conflicts are often resolved through compromises that may include denormalization
- Defects of 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
- Use denormalization cautiously
  - Understand why—under some circumstances—unnormalized tables are a better choice

Normalization and Database Design

- Normalization should be part of the design process
- Make sure that proposed entities meet required normal form before table structures are created
- Many real-world databases have been improperly designed or burdened with anomalies
- You may be asked to redesign and modify existing databases

Data-Modeling Checklist

- Data modeling translates specific real-world environment into a data model
- Data-modeling checklist helps ensure that data-modeling tasks are successfully performed

Normalization and Database Design

- ER diagram
  - Identify relevant entities, their attributes, and their relationships
  - Identify additional entities and attributes
- Normalization procedures
  - Focus on characteristics of specific entities
  - Micro view of entities within ER diagram
- Difficult to separate normalization process from ER modeling process
Summary

- Normalization is a technique used to minimize data redundancies.
- Normalization is an important part of the design process.
- Whereas ERD's provide a macro view, normalization provides micro view of entities.
  - Focuses on characteristics of specific entities.
  - May yield additional entities.
- Difficult to separate normalization from E-R diagramming – do both techniques concurrently.

Summary

- First three normal forms (1NF, 2NF, and 3NF) are most commonly encountered.
- Table is in 1NF when:
  - All key attributes are defined.
  - All remaining attributes are dependent on primary key.
- Table is in 2NF when it is in 1NF and contains no partial dependencies.
- Table is in 3NF when it is in 2NF and contains no transitive dependencies.

Summary

- Table that is not in 3NF may be split into new tables until all of the tables meet 3NF requirements.
- Table in 3NF may contain multivalued dependencies.
  - Numerous null values or redundant data.
- Convert 3NF table to 4NF by:
  - Splitting table to remove multi valued dependencies.
- Tables are sometimes denormalized to yield less I/O, which increases processing speed.
Improving the Design

- Contracting Company Example