Chapter 6 - Normalization Review

• Objectives: to learn
  – About the normal forms 1NF, 2NF, 3NF, BCNF, and 4NF
  – How normal forms can be transformed from lower normal forms to higher normal forms

• Outline: Review
  – Conversion to 1NF
  – Conversion to 2NF
  – Conversion to 3NF
  – Conversion to BCNF
  – Conversion to 4NF

Database Tables & Normalization

• Normalization:
  – A formal process for assigning attributes to entities

• Normal Forms are a series of stages done in the normalization process
  – The higher the NF,
    – the more entities one has,
    – the more flexible the database will be,
    – the more joins (and less efficiency) you have.

Conversion to First Normal Form

• Normalization from 1NF to 2NF is three-step procedure.
  – Step 1: Eliminate the Repeating Groups
    • Eliminate nulls: each repeating group attribute contains an appropriate data value
  – Step 2: Identify the Primary Key
    • Must uniquely identify attribute values
    • New key may be composed of multiple attributes
  – Step 3: Identify All Dependencies
    • Dependencies are depicted with a diagram
Conversion to 1NF - step 3

- **Step 3 - Identify All Dependencies**
  - Identify the desirable dependencies based on Primary Key
  - Identify the less desirable dependencies
    - **Partial dependencies**
      - based on part of composite primary key
      - Only possible where the Primary key is composite
    - **Transitive dependencies**
      - When one nonprime attribute depends on another nonprime attribute

Showing the dependencies (1NF) – step 3

- **Relational Schemas**: Identifies P.Key & attributes
- **Dependency Diagrams**: Identifies dependencies

Conversion to Second Normal Form

- **Step 1: Eliminate Partial Dependencies**
  - Write each part of the composite key on its own line.
  - Write the original (composite) key* on last line
- **Step 2: Assign Dependent 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 non-key attributes

---

*If 3 or more attributes make up a composite key, you’ll need to write down all various combinations.

**A combination may be eliminated if there are no dependent attributes and no M:N relationship existing with the other PK components.*
Final: Second Normal Form

Conversion to Third Normal Form

• Step 1: Eliminate Transitive Dependencies
  a. Write its determinant as PK for new table.
  b. AND leave the determinant in the Original Table
• Step 2: Reassign Corresponding Dependent Attributes
• Step 3: Name the new table(s) to reflect its contents and function

PROJECT (PROJ_NUM, PROJ_NAME)
EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)
ASSIGN (PROJ_NUM, EMP_NUM, HOURS)
JOB (JOB_CLASS, CHG_HOUR)

Improving the Design

• Additional issues to address and possibly change, in order to produce a good normalized set of tables:
  - Evaluate PK Assignments (BCNF)
  - Evaluate Multi-value variables (4 NF)
  - Evaluate Naming Conventions
  - Refine Attribute Atomicity
  - Identify New Attributes
  - Identify New Relationships
  - Refine Primary Keys as Required for Data Granularity
  - Maintain Historical Accuracy
  - Evaluate Using Derived Attributes

The Boyce-Codd Normal Form (BCNF)

• A table may be in 3NF, but not BCNF.
• BCNF is violated only when a non-key attribute is the determinant of a key attribute
  - example:
    Section (coursename, sectionno, courseno, time, days
  - Occurs most often when the wrong attribute was chosen as part of the composite Primary Key.
• Normalize to BCNF by exchanging the key attribute with the non-key-attribute.
  - This produces a partial dependency that then must be resolved.
Fourth Normal Form (4NF)

- A table in 3NF with no multiple sets of multi-valued dependencies is in 4NF.
- Examples of multi-valued dependencies
  - StudentID, StName, Phones(Home, Work, Cell, Fax)
  - StudentID, Addresses(permanent, mailing, current)
- Convert multi-valued phones into another table in 3NF & remove the phone data from the original.
  - Student(StudentID, StName, .........)
  - StuPhones(StudentID, PhoneType, Phone#)
- Reduce redundant descriptive data by creating a third table.
  - Phones(PhoneType, Description)

Summary

- Normalization is a technique used to minimize data redundancies.
- A Table is in 1NF when:
  - All key attributes are defined
  - All remaining attributes are dependent on 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 is in BCNF when it is in 3NF and contains no non-key attribute that determines a key attribute.
  - Convert to BCNF by exchanging the determinate with the key attribute and then resolving the partial dependency with a new table.
- A Table is in 4NF when it is in 3NF table with no multi-valued dependencies:
  - Convert to 4NF by splitting into two tables to remove multi-valued dependencies.