Computer Science 202

Database Systems:
Normalisation

Objectives

- To learn to identify good and poor table structures.
- To learn that poor table structures can be repaired through normalization.
- To learn normalization techniques.

Table Normalisation

- Tables are the basic unit within a relational database
- Normalisation is the process of assigning attributes to entities
  - Reduction of redundant data
  - Elimination of possible data anomalies
  - Use of controlled redundancies in order to link tables

Problems with un-normalised data

- Data Updates are less efficient and possibly error prone due to inconsistencies
- Problematic for creating views
- Indexing of data is more cumbersome

Normalisation and DB Design

- Should be part of the design process
- Entity Relation diagrams provide a high-level view
- Normalisation provides for a much lower-level view of the interaction between entities
- ER diagramming and Normalisation are closely related

Stages of Normalisation

- 1NF – First Normal Form
- 2NF – Second Normal Form
- 3NF – Third Normal Form
- 4NF – Fourth Normal Form

Conversion of DATA to 1NF

- Repeating groups to be eliminated
- Proper primary key must be selected/created
  - uniquely identifies entity rows
- Dependencies can be identified
  - Most desirable - those based on Primary Key
  - Less desirable
    - Partial – Based on part of Primary Key
    - Transitive – one non key attribute depends on another

Conversion to 2NF

- Start with 1NF data
- Write each key component on a separate line
- Write original Key on the last line
- Each Key component should be used as a new table
- Write dependant attributes after the Key on each line

Conversion to 3NF

- Start with 2NF data
- Create separate tables in order to remove transitive dependencies

Conversion to 4NF

- Data is already in 3NF
- Multiple sets of multi-valued dependencies are removed

4th Normal Form

Worked Example

Worked Example - observations

- PROJ_NUM intended to be the primary key ?
- Strong likelihood of data anomalies
  - Update
    - Modifying JOB_CLASS
  - Insertion
    - New employee assigned to a project
  - Deletion
    - Removal of employee will cause loss of other data

Worked Example : 1NF

- Primary Key
  - Use a composite key consisting of **PROJ_NUM** and **EMP_NUM**
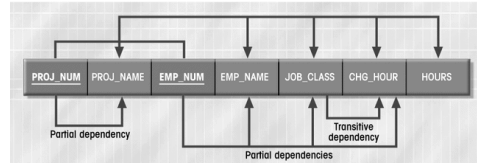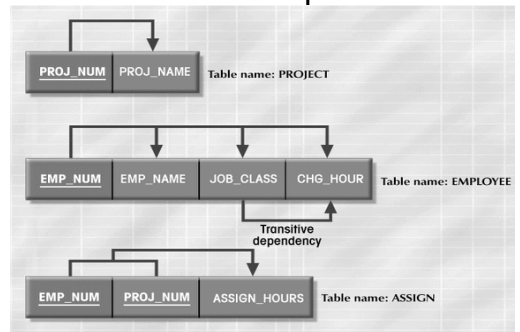
Worked Example: 2NF

- Split the Primary key up and create new tables

PROJECT (PROJ_NUM, PROJ_NAME)

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)

ASSIGN (PROJ_NUM, EMP_NUM, HOURS)

Worked Example: 2NF

Worked Example: 3NF

- Remove transitive dependencies
- JOB_CLASS → CHG_HOUR

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

DeNormalisation

- Normalisation is one of many database design goals
- Normalised table requirements
  - Additional processing
  - Loss of system speed
- Normalisation purity is difficult to sustain due to conflict in:
  - Design efficiency
  - Information requirements
  - Processing
- Some tables may be denormalised after optimal normal form has been reached in the interests of ease of implementation and speed of data processing in order to avoid multiple joins

Boyce-Codd Normal Form (BCNF)

- Every determinant in the table is a candidate key
  - Every attribute whose value determines another attribute value in a row is a determinant
  - A 3NF table with only a single candidate key is already in BCNF
- Removal of transient dependencies to another table