Revision

- The previous lecture looked at modeling in general, in particular at:
  - What models are used for
  - How systems modeling is different
  - Shortcomings of modeling, including:
    - Scalability
    - Granularity
    - Model as approximation of real world
    - Whose view is the model?
- The lecture before looked at the tabular representation of data in the relational model

Outline

- The relation model
- Functional dependencies
- Anomalies
- Normal forms
  - 1NF
  - 2NF
  - 3NF

The relational model

- This model is what we've been looking at.
- Depends on the notions of
  - Entities
  - Attributes
  - Relations

Tables

- Data is thought of in tabular form.
- The table is a grid of attributes and occurrences of entities
- The data types in the attributes are always the same
- There is only a single item of data in an occurrence for each attribute
- There cannot be identical entries [or tuples]

Redundancy or Repetition

- One of the aims of databases is to reduce the number of times the same piece of data is keyed into the system.
- This is a matter of quality control.
- The disadvantage of such an approach is that a system might have to get data from many tables [files] to generate information for a user.
### Functional dependencies
- A functional dependency is one which can be seen to be part of a chain of functions:
- E.g. a staff member of the university belongs to a department, Departments belong to Faculties. Therefore there is a further functional dependency which is also true – staff members are linked to a Faculty.

### Notation
- The usual way to indicate functional dependency is:
  - `Staff_member \rightarrow department`
or
  - `department \rightarrow faculty`
- Where the \rightarrow reads “functionally determines”
- The item on the left is called a "determinant"

### Features of a Key
- No null values
- Provides unique identification
- Can use elements from another table [Foreign key]

### Foreign Keys
- *A foreign key* is an attribute that completes a relationship by identifying the source primary entity
- Foreign keys provide for integrity in the data (referential integrity) and for navigating between different instances of an entity.
- Every relationship in the model must be supported by a foreign key, i.e. a foreign key is a link

### Key
- It is necessary to ensure that each instance of an entity is able to be identified. A key is used for this.
- The key may be an:
  - An assigned unique number [e.g. a record number automatically assigned]
  - A meaningful number [e.g. an IP number]
  - A combination of attribute values [names_dob]

### Example of Foreign Key
- An author table made up of a unique numeric primary key and surname and forename, where in another table the author details of a book are referred to by the number only, then the number is a foreign key.
Anomalies

- There are three types of anomalies that can occur with non-normalised data representation.
  - Insertion anomalies
  - Deletion anomalies
  - Modification anomalies

Sample Table

<table>
<thead>
<tr>
<th>Sid</th>
<th>Activity</th>
<th>Charge</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Skiing</td>
<td>200</td>
</tr>
<tr>
<td>100</td>
<td>Golf</td>
<td>65</td>
</tr>
<tr>
<td>150</td>
<td>Swimming</td>
<td>50</td>
</tr>
<tr>
<td>175</td>
<td>Squash</td>
<td>50</td>
</tr>
<tr>
<td>175</td>
<td>Swimming</td>
<td>50</td>
</tr>
<tr>
<td>200</td>
<td>Swimming</td>
<td>50</td>
</tr>
<tr>
<td>200</td>
<td>Golf</td>
<td>65</td>
</tr>
</tbody>
</table>

Deletion anomalies

- In the sample table, if Sid 175 is deleted from the table, then the accompanying data [that the student is engaged in activity Squash] will of course be lost,
- BUT that's the only occurrence of that activity in the table, so data about its cost will also be lost.

Modification anomalies

- If there is redundant data then if one occurrence of that data is changed each other occurrence must also be changed, or there will be inconsistencies. [e.g. in the sample table, if the cost of Swimming changes, it must be changed three times.

Insertion Anomalies

- The sample table uses Student ID [Sid] and Activity for the key.
- This makes it impossible to add a student until she undertakes an activity, or a new activity until a student takes it.

1st Normal Form

- Data in 1NF meets the minimal requirements of storing relation data in a table:
  - Cells must have a single value. i.e.Cells cannot contain an array
  - Ordering of columns and rows is not significant
  - No duplicate rows are permitted
Example

• Data entered into a spreadsheet format will be in 1NF. The Key is the first 3 attributes.

<table>
<thead>
<tr>
<th>Author Family Name</th>
<th>Author First Name</th>
<th>Book Title</th>
<th>Subject</th>
<th>Collection</th>
<th>Building</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>John</td>
<td>Philosophy of everything</td>
<td>Philosophy</td>
<td>Matheson</td>
<td>Main</td>
</tr>
<tr>
<td>Graves</td>
<td>Robert</td>
<td>Collected</td>
<td>Literature</td>
<td>Matheson</td>
<td>Main</td>
</tr>
<tr>
<td>Bell</td>
<td>Peter</td>
<td>Timber and iron</td>
<td>Architecture</td>
<td>Hargrave</td>
<td>Hargrave</td>
</tr>
<tr>
<td>Graves</td>
<td>Robert</td>
<td>English and Scottish ballads</td>
<td>Literature</td>
<td>Matheson</td>
<td>Main</td>
</tr>
<tr>
<td>Smart</td>
<td>Jeffrey</td>
<td>Retrospective</td>
<td>Art</td>
<td>Matheson</td>
<td>Main</td>
</tr>
<tr>
<td>Shippe</td>
<td>Tom</td>
<td>Grantchester grind</td>
<td>Fiction</td>
<td>Main</td>
<td></td>
</tr>
</tbody>
</table>

2nd Normal Form

• If an attribute is functionally dependent on a key that is a combination, both insertion and deletion anomalies can arise.
• The problem can be overcome by separating the elements of the combination key, and creating two tables.

Some problems here

• Title is now the apparent key for the table, or else a combined key of author number and title. [Mixed data types]
• The linkage between subject and collection, and between collection and building, still contain redundant data, prone to each type of anomaly.

Split the Author name from the Title

<table>
<thead>
<tr>
<th>Author No.</th>
<th>Author Last Name</th>
<th>Author First Name(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Smith</td>
<td>John</td>
</tr>
<tr>
<td>2</td>
<td>Graves</td>
<td>Robert</td>
</tr>
<tr>
<td>3</td>
<td>Bell</td>
<td>Peter</td>
</tr>
<tr>
<td>4</td>
<td>Smart</td>
<td>Jeffrey</td>
</tr>
<tr>
<td>5</td>
<td>Shippe</td>
<td>Tom</td>
</tr>
</tbody>
</table>

Fixing it

• Titles could be assigned numbers [database generated? ISBNs? something meaningful?]
• Library names are unique, and could be used as a key for a table determining which library is in which building.
• This meets the requirements of 3NF.
3rd Normal Form

- Even after separating tables to achieve 2NF it is still possible to have anomalies, where a transitive dependency occurs.
- After removing transitive dependencies, then the tables represent 3NF.
- Further forms exist, but for the purposes of this subject [and using Access] 3NF is far enough.

Some other instances…

- When data are collected very rapidly a set of tables for those data might be denormalised to make writing to the database faster.
- Similarly, reports may draw on many tables, and it may be faster, for report generation, to carry redundant data in denormalised tables, than to join many tables “on the fly”.

Denormalisation

- This is a process of deliberately failing to comply with normalized data requirements in the interest of physical database operational efficiency. i.e. Expediency overtaking exactness.

A Useful Logical Modeling Reference


Example of Denormalisation

- In a database with addresses, you would, to remove redundancy, generate a table with suburbs and their postcodes.
- When a new address is keyed, the postcode would be automatically assigned in the postcode table.
- But whenever a mailout was printed, the database would have to join the customer address data with the postcode table.
- This is inefficient, and in practice you’d probably denormalise the data, and carry the postcode in each customer’s address.

Summary

- Need for unique identifier
- Functional dependencies
- Anomalies
- 3 Normal forms [1NF, 2NF, 3NF]
- Denormalization