Things You Already Know

or “The Dark Art of Database Normalisation”

Originally Presented by Dominic Gruitjers
Adapted by Philip Norton
Contents

• Database Design
• Keys
• What is Normalisation?
• Why Normalise a Database?
• How do we Normalise a Database?
• Examples
• Assessment
Database Design

• Database design is the formal process of analyzing facts about the real world into a structured database model. Database design is characterized by the following phases: requirement analysis, logical design and physical design.
  – The GIS Glossary Environmental Systems Research Institute, Inc. 1995

• If we teach algorithms before we teach programming why aren’t we teaching DB design before Access or MySQL?
Database Design

- Database Design Techniques
  - Entity-Relationship Diagrams
  - Normalisation
  - Relational Algebra
  - Relational Calculus
  - Types of databases
    - Relational
    - Hierarchical
    - Object-Relational
    - Temporal

- Database Design is largely a philosophy and often a dark art
Database Design
Rethinking Keys
Keys

• Super Keys
• Candidate Keys
• Primary Keys
• Compound/Composite/Concatenated Keys
Keys

• Super Keys
  – A combination of fields in a table which uniquely identify each row.
  – Consider the following Student table

  – Student (StuID, IDNum, Name)

  – Super Keys: {StuID, IDNum, Name}
  – {StuID, Name}
  – {StuID, IDNum}
  – {IDNum, Name}
  – {IDNum}
  – {StuNum}
Keys

• **Candidate Keys**
  
  – Minimal super-keys
  
  – In other words there are no fields included in these keys which don’t contribute to the uniqueness of the rows

  
  – Candidate Keys of the Student relation
    
    • \{StuNum, IDNum\}
    
    • \{StuNum\}
    
    • \{IDNum\}
• **Primary Keys**
  
  – A minimal candidate key
  
  – In other words one candidate key chosen as a Primary Key for its simplicity
  
  – Primary Key for the Student relation could be either:
    
    – {StuNum}
    
    – {IDNum}
Keys

• Composite/Compound/Concatenated Keys
  – Any key which includes more than one attribute or field
  – In other words, all the previous keys can also be composite keys

  – PRIMARY KEYS CAN ALSO BE COMPOSITE KEYS

  – Books (Author, Title, Pages, Cost)

  – In the absence of an ISBN number the combination of Author and Title can serve as a primary key for the Books relation – no author writes a book with the same title twice.
Feel good picture
What is Normalisation?

• “[Normalisation] is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies”

• http://en.wikipedia.org/wiki/Database_normalization
Why Normalise?

• Normalisation avoids these very bad things:
  – Repeating Groups
  – Data redundancy (NOT repetition!)
  – Anomalies
    • Update
    • Insertion
    • Deletion
  – Tricky queries

  – Plus it earns you geek points
Why Normalise?

Repeating Groups – Type 1

<table>
<thead>
<tr>
<th>Name</th>
<th>Grade</th>
<th>Sport 1</th>
<th>Sport 2</th>
<th>Sport 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roger Philips</td>
<td>10</td>
<td>Cricket</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patience Mbata</td>
<td>10</td>
<td>Netball</td>
<td>Hockey</td>
<td></td>
</tr>
<tr>
<td>Hennie Venter</td>
<td>12</td>
<td>Hockey</td>
<td>Rugby</td>
<td>Golf</td>
</tr>
<tr>
<td>Sarah Cohen</td>
<td>11</td>
<td>Golf</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why Normalise?

Repeating Groups – Type 2

<table>
<thead>
<tr>
<th>Name</th>
<th>Grade</th>
<th>Sports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roger Philips</td>
<td>10</td>
<td>Cricket</td>
</tr>
<tr>
<td>Patience Mbata</td>
<td>10</td>
<td>Netball, Hockey</td>
</tr>
<tr>
<td>Hennie Venter</td>
<td>12</td>
<td>Hockey, Rugby, Golf</td>
</tr>
<tr>
<td>Sarah Cohen</td>
<td>11</td>
<td>Golf</td>
</tr>
</tbody>
</table>
**Why Normalise?**

Data Redundancy

<table>
<thead>
<tr>
<th>Name</th>
<th>Grade</th>
<th>Sports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roger Philips</td>
<td>10</td>
<td>Cricket</td>
</tr>
<tr>
<td>Patience Mbata</td>
<td>10</td>
<td>Netball</td>
</tr>
<tr>
<td>Patience Mbata</td>
<td>10</td>
<td>Hockey</td>
</tr>
<tr>
<td>Hennie Venter</td>
<td>12</td>
<td>Hockey</td>
</tr>
<tr>
<td>Hennie Venter</td>
<td>12</td>
<td>Rugby</td>
</tr>
<tr>
<td>Hennie Venter</td>
<td>12</td>
<td>Golf</td>
</tr>
<tr>
<td>Sarah Cohen</td>
<td>11</td>
<td>Golf</td>
</tr>
</tbody>
</table>
Why Normalise?

• Update Anomaly
  – Occurs when we need to update the same data in more than one place
  – E.g.
    • When all the students change grades at the end of the year we need to assign new values to the Grade field
    • We need to update the Grade for each student in more than one place
    • Human error could lead to inconsistencies
Why Normalise?

• Deletion Anomaly
  – Occurs when a deletion causes loss of data unnecessarily
  – E.g.
    • If Patience decides to stop playing sport and focus on her academics we need to delete two records
    • In doing so we delete Patience completely from the database!
    • You’ve basically ruined her life... shame on you.
Why Normalise?

• Insertion Anomaly
  – Occurs when we add a record and it does not satisfy the design or the primary key requirements
  – E.g.
    • If a new student comes to the school we need to add them to the table.
    • The student cannot be added to the database without first choosing a sport because that would violate the primary key requirement
Feel good picture
How do we Normalise?

• “Normalisation is a methodical process informed by intuition.”
• Normalisation involves iteratively leveling relations, producing new relations in various Normal Forms
• Normal Forms
  – 1st Normal Form – 1NF
  – 2nd Normal Form – 2NF
  – 3rd Normal Form – 3NF
How do we Normalise?

• Characteristics of Normal Forms
  – 1NF
    • No repeating groups
  – 2NF
    • Relation is in 1NF
    • No partial dependencies
  – 3NF
    • Relation is in 2NF
    • No transitive dependencies
How do we Normalise?

• Dependencies
  – Partial Dependency
    • When a field’s value is dependent on only part of a composite key
    • Can you spot the partial dependency in the Student relation?
  – Transitive Dependency
    • When a field’s value is dependent on a non-key field
    • Can you spot the transitive dependency in the Student relation?
### How do we Normalise?

Consider the following data in the `Student` table relating to sports choices at a school:

<table>
<thead>
<tr>
<th>StuID</th>
<th>Name</th>
<th>Grade</th>
<th>House</th>
<th>House_Colour</th>
<th>Sport</th>
<th>Coach</th>
<th>Practice_Day</th>
<th>NumMatches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1225</td>
<td>Joe Soap</td>
<td>10</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Cricket U16</td>
<td>Lezar</td>
<td>Thursday</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rugby 1st Team</td>
<td>Meyer</td>
<td>Tuesday</td>
<td>2</td>
</tr>
<tr>
<td>1124</td>
<td>Roger Phillips</td>
<td>10</td>
<td>Oxford</td>
<td>Blue</td>
<td>Cricket U16</td>
<td>Lezar</td>
<td>Thursday</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Netball U18</td>
<td>Smid</td>
<td>Wednesday</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Girls Hockey U18</td>
<td>Bailly</td>
<td>Monday</td>
<td>5</td>
</tr>
<tr>
<td>2345</td>
<td>Patience Mbata</td>
<td>11</td>
<td>Yale</td>
<td>Red</td>
<td>Boys Hockey U19</td>
<td>Jones</td>
<td>Monday</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rugby 1st Team</td>
<td>Meyer</td>
<td>Tuesday</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Golf</td>
<td>Joubert</td>
<td>Friday</td>
<td>7</td>
</tr>
<tr>
<td>3221</td>
<td>Hennie Visser</td>
<td>12</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Golf</td>
<td>Joubert</td>
<td>Friday</td>
<td>5</td>
</tr>
<tr>
<td>2334</td>
<td>Sarah Cohen</td>
<td>11</td>
<td>Oxford</td>
<td>Blue</td>
<td>Golf</td>
<td>Joubert</td>
<td>Friday</td>
<td>5</td>
</tr>
</tbody>
</table>
How do we Normalise?

• This table can be written as

• Students (StuID, Name, Grade, House, House_Colour, Sport, Coach, Practice Day, NumMatches)
## Normalise

- **1NF – Not all it’s cracked up to be**

<table>
<thead>
<tr>
<th>StuID</th>
<th>Name</th>
<th>Grade</th>
<th>House</th>
<th>House_Colour</th>
<th>Sport</th>
<th>Coach</th>
<th>Practice Day</th>
<th>NumMatches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1225</td>
<td>Joe Soap</td>
<td>10</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Cricket U16</td>
<td>Lezar</td>
<td>Thursday</td>
<td>4</td>
</tr>
<tr>
<td>1225</td>
<td>Joe Soap</td>
<td>10</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Rugby 1st Team</td>
<td>Meyer</td>
<td>Tuesday</td>
<td>2</td>
</tr>
<tr>
<td>1124</td>
<td>Roger Phillips</td>
<td>10</td>
<td>Oxford</td>
<td>Blue</td>
<td>Cricket U16</td>
<td>Lezar</td>
<td>Thursday</td>
<td>3</td>
</tr>
<tr>
<td>2345</td>
<td>Patience Mbata</td>
<td>11</td>
<td>Yale</td>
<td>Red</td>
<td>Netball U18</td>
<td>Smid</td>
<td>Wednesday</td>
<td>8</td>
</tr>
<tr>
<td>2345</td>
<td>Patience Mbata</td>
<td>11</td>
<td>Yale</td>
<td>Red</td>
<td>Girls Hockey U18</td>
<td>Bailly</td>
<td>Monday</td>
<td>5</td>
</tr>
<tr>
<td>3221</td>
<td>Hennie Visser</td>
<td>12</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Boys Hockey U19</td>
<td>Jones</td>
<td>Monday</td>
<td>5</td>
</tr>
<tr>
<td>3221</td>
<td>Hennie Visser</td>
<td>12</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Rugby 1st Team</td>
<td>Meyer</td>
<td>Tuesday</td>
<td>2</td>
</tr>
<tr>
<td>3221</td>
<td>Hennie Visser</td>
<td>12</td>
<td>Cambridge</td>
<td>Yellow</td>
<td>Golf</td>
<td>Joubert</td>
<td>Friday</td>
<td>7</td>
</tr>
<tr>
<td>2334</td>
<td>Sarah Cohen</td>
<td>11</td>
<td>Oxford</td>
<td>Blue</td>
<td>Golf</td>
<td>Joubert</td>
<td>Friday</td>
<td>5</td>
</tr>
</tbody>
</table>
How do we Normalise?

• 1NF
  – Flattened the table
  – Chose Key
  – Eliminated Repeating Groups

• 1NF does not eliminate data redundancy
• 1NF does not generate new tables
• This relation can be written as

• Students (StuID, Name, Grade, House, House_Colour, Sport, Coach, Practice Day, NumMatches)
How do we Normalise?

• 2NF – Draw a dependency Diagram

• 2NF – Write all keys separately, then together

StuID
Sport
StuID, Sport
How do we Normalise?

• 2NF – Write all dependant fields next to their respective keys
  - StuID, Name, Grade, House, House_Col
  - Sport, Coach, Practice Day
  - StuID, Sport, NumMatches

• 2NF – Complete
  - No more partial dependencies
  - Reduces redundancy considerably
How do we Normalise?

<table>
<thead>
<tr>
<th>StuID</th>
<th>Sport</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1225</td>
<td>Cricket U16</td>
<td>4</td>
</tr>
<tr>
<td>1225</td>
<td>Rugby 1st Team</td>
<td>2</td>
</tr>
<tr>
<td>1124</td>
<td>Cricket U16</td>
<td>3</td>
</tr>
<tr>
<td>2345</td>
<td>Netball U18</td>
<td>8</td>
</tr>
<tr>
<td>2345</td>
<td>Girls Hockey U18</td>
<td>5</td>
</tr>
<tr>
<td>3221</td>
<td>Boys Hockey U19</td>
<td>5</td>
</tr>
<tr>
<td>3221</td>
<td>Rugby 1st Team</td>
<td>2</td>
</tr>
<tr>
<td>3221</td>
<td>Golf</td>
<td>7</td>
</tr>
<tr>
<td>2334</td>
<td>Golf</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sport</th>
<th>Coach</th>
<th>Practice Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cricket U16</td>
<td>Lezar</td>
<td>Thursday</td>
</tr>
<tr>
<td>Netball U18</td>
<td>Smid</td>
<td>Wednesday</td>
</tr>
<tr>
<td>Girls Hockey U18</td>
<td>Bailly</td>
<td>Monday</td>
</tr>
<tr>
<td>Boys Hockey U19</td>
<td>Jones</td>
<td>Monday</td>
</tr>
<tr>
<td>Rugby 1st Team</td>
<td>Meyer</td>
<td>Tuesday</td>
</tr>
<tr>
<td>Golf</td>
<td>Joubert</td>
<td>Friday</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StuID</th>
<th>Name</th>
<th>Grade</th>
<th>House</th>
<th>House_Col</th>
</tr>
</thead>
<tbody>
<tr>
<td>1225</td>
<td>Joe Soap</td>
<td>10</td>
<td>Cambridge</td>
<td>Yellow</td>
</tr>
<tr>
<td>1124</td>
<td>Roger Phillips</td>
<td>10</td>
<td>Oxford</td>
<td>Blue</td>
</tr>
<tr>
<td>2345</td>
<td>Patience Mpbta</td>
<td>11</td>
<td>Yale</td>
<td>Red</td>
</tr>
<tr>
<td>3221</td>
<td>Hennie Visser</td>
<td>12</td>
<td>Cambridge</td>
<td>Yellow</td>
</tr>
<tr>
<td>2334</td>
<td>Sarah Cohen</td>
<td>11</td>
<td>Oxford</td>
<td>Blue</td>
</tr>
</tbody>
</table>
Assessment

• Various forms
  – Given spread-sheet type data carry out the normalisation process
  – Identify dependencies
  – Draw dependency diagrams
  – Identify anomalies and redundancy
  – Given a relation state which normal form it’s in with reasons
  – Weigh up two designs in terms of normalisation theory