Chapter 14 & 15 - Objectives

- The purpose of normalization.
  - The potential problems associated with redundant data in base relations.
- The concept of functional dependency, which describes the relationship between attributes.
  - How to identify functional dependencies for a given relation.
  - How functional dependencies identify the primary key for a relation.
- How to undertake the process of normalization.
  - How to identify the most commonly used normal forms, namely 1NF, 2NF, 3NF, Boyce–Codd Normal Form (BCNF), and 4NF.

Contents

- The purpose of normalization
  - Data redundancy
- Functional dependency and Transitive dependency
- Process of normalization
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce–Codd Normal Form (BCNF)
  - Multi-valued dependencies and Fourth Normal Form (4NF)

Purpose of Normalization

- **Normalization** is a technique for producing a set of suitable relations that support the data requirements of an enterprise.
- Characteristics of a suitable set of relations include:
  - the minimal number of attributes necessary to support the data requirements of the enterprise;
  - attributes with a close logical relationship are found in the same relation;
  - minimal redundancy with each attribute represented only once with the important exception of attributes that form all or part of foreign keys.
Data Redundancy and Update Anomalies

- Major aim of relational database design is to group attributes into relations to minimize data redundancy.
- Potential benefits for implemented database include:
  - Reduction in the file storage space required by the base relations thus minimizing costs.
  - Updates to the data stored in the database are achieved with a minimal number of operations thus reducing the opportunities for data inconsistencies.

- Problems associated with data redundancy are illustrated by comparing the Staff and Branch relations with the StaffBranch relation.

<table>
<thead>
<tr>
<th>Staff</th>
<th>Branch</th>
<th>StaffBranch</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>B1</td>
<td>11</td>
</tr>
<tr>
<td>s1</td>
<td>B2</td>
<td>12</td>
</tr>
<tr>
<td>s1</td>
<td>B3</td>
<td>13</td>
</tr>
<tr>
<td>s2</td>
<td>B4</td>
<td>14</td>
</tr>
<tr>
<td>s2</td>
<td>B5</td>
<td>15</td>
</tr>
<tr>
<td>s2</td>
<td>B6</td>
<td>16</td>
</tr>
<tr>
<td>s2</td>
<td>B7</td>
<td>17</td>
</tr>
</tbody>
</table>

Data Redundancy and Update Anomalies

- StaffBranch relation has redundant data; the details of a branch are repeated for every member of staff.

<table>
<thead>
<tr>
<th>Staff</th>
<th>Branch</th>
<th>StaffBranch</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>B1</td>
<td>11</td>
</tr>
<tr>
<td>s1</td>
<td>B2</td>
<td>12</td>
</tr>
<tr>
<td>s1</td>
<td>B3</td>
<td>13</td>
</tr>
<tr>
<td>s2</td>
<td>B4</td>
<td>14</td>
</tr>
<tr>
<td>s2</td>
<td>B5</td>
<td>15</td>
</tr>
<tr>
<td>s2</td>
<td>B6</td>
<td>16</td>
</tr>
<tr>
<td>s2</td>
<td>B7</td>
<td>17</td>
</tr>
</tbody>
</table>

- In contrast, the branch information appears only once for each branch in the Branch relation and only the branch number (branchNo) is repeated in the Staff relation, to represent where each member of staff is located.
Data Redundancy and Update Anomalies

- Relations that contain redundant information may potentially suffer from update anomalies.

- Types of update anomalies include
  - Insertion
  - Deletion
  - Modification

Functional Dependencies

- Important concept associated with normalization are functional dependency and transitive dependency

- Functional dependency describes relationship between attributes.

- For example, if A and B are attributes of relation R, B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R.

Characteristics of Functional Dependencies

- Property of the meaning or semantics of the attributes in a relation.

- Diagrammatic representation.

  - The determinant of a functional dependency refers to the attribute or group of attributes on the left-hand side of the arrow.
An Example Functional Dependency

Characteristics of Functional Dependencies

• Determinants should have the minimal number of attributes necessary to maintain the functional dependency with the attribute(s) on the right hand-side.

• This requirement is called full functional dependency.

• Full functional dependency indicates that if A and B are attributes of a relation, B is fully functionally dependent on A, if B is functionally dependent on A, but not on any proper subset of A.

Characteristics of Functional Dependencies

• Main characteristics of functional dependencies used in normalization:
  - There is a one-to-one relationship between the attribute(s) on the left-hand side (determinant) and those on the right-hand side of a functional dependency.
  - The determinant has the minimal number of attributes necessary to maintain the dependency with the attribute(s) on the right hand-side.
  - Holds for all time.

Transitive Dependencies

• Transitive dependency describes a condition where A, B, and C are attributes of a relation such that if $A \rightarrow B$ and $B \rightarrow C$, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

• Important to recognize a transitive dependency because its existence in a relation can potentially cause update anomalies.
The Process of Normalization

- Formal technique for analyzing a relation based on its primary key and the functional dependencies between the attributes of that relation.

- Often executed as a series of steps. Each step corresponds to a specific normal form, which has known properties.

Identifying Functional Dependencies

- Identifying all functional dependencies between a set of attributes is relatively simple if...
  - the meaning of each attribute and the relationships between the attributes are well understood.

- This information should be provided by the enterprise in the form of...
  - discussions with users and/or users’ requirements specification.
  - common sense and experience!

Example - Identifying a set of functional dependencies for the StaffBranch relation

- Examine semantics of attributes in StaffBranch relation (see Slide 7). Assume that position held and branch determine a member of staff’s salary.

- With sufficient information available, identify the functional dependencies for the StaffBranch relation as:
  - staffNo → sName, position, salary, branchNo, bAddress
  - branchNo → bAddress
  - bAddress → branchNo
  - branchNo, position → salary
  - bAddress, position → salary
Example - Using sample data to identify functional dependencies.

- Consider the data for attributes denoted A, B, C, D, and E in the Sample relation (see next slide).

- Important to establish that sample data values shown in relation are representative of all possible values that can be held by attributes A, B, C, D, and E.
  - Assume true despite the relatively small amount of data shown in this relation.

Example - Using sample data to identify functional dependencies.

A (fd2) ▫ B
C (fd1) ▫ C
D (fd3) ▫ A, B
A, B ▫ E (fd4)

Identifying the Primary Key for a Relation using Functional Dependencies

- Main purpose of identifying a set of functional dependencies for a relation is to specify the set of integrity constraints that must hold on a relation.

- An important integrity constraint to consider first is the identification of candidate keys, one of which is selected to be the primary key for the relation.
Example - Identifying Primary Key for Sample Relation

- Sample relation has four functional dependencies (see Slide 23).
- The determinants in the Sample relation are A, B, C, and (A, B).
- However, the only determinant that functionally determines all the other attributes of the relation is (A, B).
- (A, B) is identified as the primary key for this relation.

The Process of Normalization

- As normalization proceeds, the relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.

Contents

- The purpose of normalization
  - Data redundancy
- Functional dependency and Transitive dependency
- Process of normalization
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce–Codd Normal Form (BCNF)
  - Multi-valued dependencies and Fourth Normal Form (4NF)
Unnormalized Form (UNF)

- A table that contains one or more repeating groups.
- To create an unnormalized table
  - Transform the data from the information source (e.g. form) into table format with columns and rows.

Contents

- The purpose of normalization
  - Data redundancy
- Functional dependency and Transitive dependency
- Process of normalization
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce–Codd Normal Form (BCNF)
  - Multi-valued dependencies and Fourth Normal Form (4NF)

First Normal Form (1NF)

- A relation in which the intersection of each row and column contains one and only one value.

UNF to 1NF

- Nominate an attribute or group of attributes to act as the key for the unnormalized table.
- Identify the repeating group(s) in the unnormalized table which repeats for the key attribute(s).
- Remove the repeating group by
  - Placing the repeating data along with a copy of the original key attribute(s) into a separate relation.
  - or sometimes by
  - Entering appropriate data into the empty columns of rows containing the repeating data.
Second Normal Form (2NF)

- A relation is in 2NF if it is in 1NF and every non-primary-key attribute is *fully functionally dependent* on the primary key.

- Full functional dependency indicates that if
  - A and B are attributes of a relation, (then)
  - B is fully dependent on A if B is functionally dependent on A but not on any proper subset of A.

- (B is *partial* dependent on A if B is functionally dependent on some proper subset of A.)

1NF to 2NF

- Identify the primary key for the 1NF relation.
- Identify the functional dependencies in the relation.
- *Remove every partial dependency* exist on the primary key by placing then in a new relation along with a copy of their determinant.
Review of Normalization (UNF to BCNF)

<table>
<thead>
<tr>
<th>propertyNo</th>
<th>pAddress</th>
<th>iDate</th>
<th>iTime</th>
<th>comments</th>
<th>staffNo</th>
<th>sName</th>
<th>carReg</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG4</td>
<td>6 Lawrence St.</td>
<td>13-Oct-05</td>
<td>10:00</td>
<td>Need to replace window</td>
<td>SG07</td>
<td>Ann Becks</td>
<td>M251 JGR</td>
</tr>
<tr>
<td></td>
<td>Glasgow</td>
<td></td>
<td></td>
<td>in good order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>22-Apr-04</td>
<td>09:00</td>
<td></td>
<td>SG12</td>
<td>David Ford</td>
<td>M535 HDR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1-Oct-04</td>
<td>12:00</td>
<td></td>
<td>SG14</td>
<td>David Ford</td>
<td>N721 HFR</td>
</tr>
<tr>
<td>PG16</td>
<td>2 Nov., Dr.</td>
<td>22-Apr-04</td>
<td>13:00</td>
<td>Damp in living room</td>
<td>SG14</td>
<td>David Ford</td>
<td>M251 JGR</td>
</tr>
<tr>
<td></td>
<td>Glasgow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7-Oct-04</td>
<td>14:00</td>
<td>Good condition</td>
<td>SG07</td>
<td>Ann Becks</td>
<td>N721 HFR</td>
</tr>
</tbody>
</table>

Example: Identifying functional dependencies

<table>
<thead>
<tr>
<th>propertyNo</th>
<th>iDate</th>
<th>iTime</th>
<th>pAddress</th>
<th>comments</th>
<th>staffNo</th>
<th>sName</th>
<th>carReg</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>13-Oct-05</td>
<td>10:00</td>
<td>6 Lawrence St.</td>
<td>Need to replace window in good order</td>
<td>SG07</td>
<td>Ann Becks</td>
<td>M251 JGR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Glasgow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f2</td>
<td>22-Apr-04</td>
<td>09:00</td>
<td>6 Lawrence St.</td>
<td>Damp in living room</td>
<td>SG14</td>
<td>David Ford</td>
<td>M535 HDR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Glasgow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f3</td>
<td>1-Oct-04</td>
<td>12:00</td>
<td>6 Lawrence St.</td>
<td>Damp in bathroom</td>
<td>SG14</td>
<td>David Ford</td>
<td>N721 HFR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Glasgow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f4</td>
<td>2 Nov., Dr.</td>
<td>13:00</td>
<td>2 Nov., Dr.</td>
<td>Replace living room</td>
<td>SG14</td>
<td>David Ford</td>
<td>M535 HDR</td>
</tr>
<tr>
<td></td>
<td>Glasgow</td>
<td></td>
<td>Glasgow</td>
<td>Good condition</td>
<td>SG07</td>
<td>Ann Becks</td>
<td>N721 HFR</td>
</tr>
</tbody>
</table>

Example: Normalization (UNF to 2NF)

StaffPropertyInspection 1NF

PropertyInspection Property 2NF

Contents

- The purpose of normalization
  - Data redundancy
- Functional dependency and Transitive dependency
- Process of normalization
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce–Codd Normal Form (BCNF)
  - Multi-valued dependencies and Fourth Normal Form (4NF)
Third Normal Form (3NF)

- A relation that is in 1NF and 2NF and in which no attribute is transitively dependent on the primary key.

- Transitive dependency is a condition where
  - A, B and C are attributes of a relation such that if A → B and B → C, then C is transitively dependent on A through B.

2NF to 3NF

- Identify the primary key in the 2NF relation.
- Identify functional dependencies in the relation.
- Remove transitive dependencies exist on the primary key by placing them in a new relation along with a copy of their dominant.

General Definitions of 2NF and 3NF

- Second normal form (2NF)
  - A relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on any candidate key.

- Third normal form (3NF)
  - A relation that is in second normal form and in which no non-primary-key attribute is transitively dependent on any candidate key.

Example: Normalization (2NF to 3NF)
**Summary**

Unnormalized Form (UNF)

- Remove repeating groups

First Normal Form (1NF)

- "Requiring existence of 'the key' ensures that the table is in 1NF."

Second Normal Form (2NF)

- "Requiring that non-key attributes be dependent on 'the whole key' ensures 2NF."

Third Normal Form (3NF)

- "Further requiring that non-key attributes be dependent on 'nothing but the key' ensures 3NF."

@2012 by Chukiat Worasucheep

---

**Exercise**

- Is this table in 1NF or 2NF or 3NF?

<table>
<thead>
<tr>
<th>Tournament</th>
<th>Year</th>
<th>Winner</th>
<th>Winner Date of Birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indiana Invitational</td>
<td>1998</td>
<td>Al Fredrickson</td>
<td>21 July 1975</td>
</tr>
<tr>
<td>Cleveland Open</td>
<td>1998</td>
<td>Hub Allenstein</td>
<td>28 September 1968</td>
</tr>
<tr>
<td>Des Moines Masters</td>
<td>1999</td>
<td>Al Fredrickson</td>
<td>21 July 1975</td>
</tr>
<tr>
<td>Indiana Invitational</td>
<td>1999</td>
<td>Chip Masterson</td>
<td>14 March 1977</td>
</tr>
</tbody>
</table>

@2012 by Chukiat Worasucheep

---

**Exercise (cont.)**

<table>
<thead>
<tr>
<th>Tournament</th>
<th>Year</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indiana Invitational</td>
<td>1998</td>
<td>Al Fredrickson</td>
</tr>
<tr>
<td>Cleveland Open</td>
<td>1998</td>
<td>Hub Allenstein</td>
</tr>
<tr>
<td>Des Moines Masters</td>
<td>1999</td>
<td>Al Fredrickson</td>
</tr>
<tr>
<td>Indiana Invitational</td>
<td>1999</td>
<td>Chip Masterson</td>
</tr>
</tbody>
</table>

**Player Dates of Birth**

<table>
<thead>
<tr>
<th>Player</th>
<th>Date of Birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip Masterson</td>
<td>14 March 1977</td>
</tr>
<tr>
<td>Al Fredrickson</td>
<td>21 July 1975</td>
</tr>
<tr>
<td>Bob Albertson</td>
<td>28 September 1968</td>
</tr>
</tbody>
</table>

@2012 by Chukiat Worasucheep

---

**Contents**

- The purpose of normalization
  - Data redundancy
- Functional dependency and Transitive dependency
- Process of normalization
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce–Codd Normal Form (BCNF)
- Multi-valued dependencies and Fourth Normal Form (4NF)

@2012 by Chukiat Worasucheep
Boyce–Codd Normal Form (BCNF)

- BCNF has additional constraints compared with the definition of 3NF.
- Boyce–Codd normal form (BCNF)
  - A relation is in BCNF if and only if every determinant is a candidate key.

Example: Tennis court

- A tennis club that has one hard court (Court 1) and one grass court (Court 2)
- A booking is defined by its Court and the period for which the Court is reserved
- Additionally, each booking has a Rate Type associated with it. There are four distinct rate types:
  - SAVER, for Court 1 bookings made by members
  - STANDARD, for Court 1 bookings made by non-members
  - PREMIUM-A, for Court 2 bookings made by members
  - PREMIUM-B, for Court 2 bookings made by non-members

Example: Tennis court (cont.)

<table>
<thead>
<tr>
<th>Today's Court Bookings</th>
<th>Court</th>
<th>Start Time</th>
<th>End Time</th>
<th>Rate Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>08:30</td>
<td>10:30</td>
<td></td>
<td>SAVER</td>
</tr>
<tr>
<td>1</td>
<td>11:00</td>
<td>12:00</td>
<td></td>
<td>SAVER</td>
</tr>
<tr>
<td>1</td>
<td>14:00</td>
<td>15:30</td>
<td></td>
<td>STANDARD</td>
</tr>
<tr>
<td>2</td>
<td>10:00</td>
<td>11:30</td>
<td></td>
<td>PREMIUM-H</td>
</tr>
<tr>
<td>2</td>
<td>11:30</td>
<td>13:30</td>
<td></td>
<td>PREMIUM-H</td>
</tr>
<tr>
<td>2</td>
<td>15:00</td>
<td>16:30</td>
<td></td>
<td>PREMIUM-A</td>
</tr>
</tbody>
</table>
Example: Tennis court (cont.)

- Since the Court Bookings table contains no partial FDs, and no transitive dependencies, the table adheres to both 2NF and 3NF.
- However the table does not adhere to BCNF.
- This is because of the dependency Rate Type → Court, in which the determining attribute (Rate Type) is neither a candidate key nor a superset of a candidate key.
- Dependency Rate Type → Court is respected as a Rate Type should only ever apply to a single Court.

Review of Normalization (UNF to BCNF)

<table>
<thead>
<tr>
<th>StaffPropertyInspection</th>
<th>1NF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropertyInspection</td>
<td>2NF-</td>
</tr>
<tr>
<td>Staff</td>
<td>3NF</td>
</tr>
</tbody>
</table>

Example: Tennis court - Adjusted Tables

<table>
<thead>
<tr>
<th>Rate Types</th>
<th>Court Member ID</th>
<th>Today's Booking</th>
<th>Rate Type</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANDARD</td>
<td>1</td>
<td>SavFR</td>
<td>SavFR</td>
<td>09:30</td>
<td>10:30</td>
</tr>
<tr>
<td>STANDARD</td>
<td>2</td>
<td>SavFR</td>
<td>SavFR</td>
<td>11:00</td>
<td>12:00</td>
</tr>
<tr>
<td>PREMIUM A</td>
<td>3</td>
<td>PREMIUM A</td>
<td>10:00</td>
<td>11:30</td>
<td></td>
</tr>
<tr>
<td>PREMIUM A</td>
<td>4</td>
<td>PREMIUM A</td>
<td>11:30</td>
<td>13:30</td>
<td></td>
</tr>
<tr>
<td>PREMIUM B</td>
<td>5</td>
<td>PREMIUM B</td>
<td>15:00</td>
<td>16:30</td>
<td></td>
</tr>
</tbody>
</table>
Fourth Normal Form (4NF)

- Although BCNF removes anomalies due to functional dependencies, another type of dependency called a multi-valued dependency (MVD) can also cause data redundancy.

  - Multi-valued Dependency (MVD)
    - Dependency between attributes (for example, A, B, and C) in a relation, such that for each value of A there is a set of values for B and a set of values for C. However, the set of values for B and C are independent of each other.

Fourth Normal Form (4NF)

- MVD between attributes A, B, and C in a relation using the following notation:
  - A \rightarrow\rightarrow B
  - A \rightarrow\rightarrow C

- A relation is in 4NF if it is in Boyce-Codd Normal Form and contains no multi-valued dependencies.
What We’ve Learnt Today

• The purpose of normalization
  ▫ Data redundancy
• Functional dependency and Transitive dependency
• Process of normalization
  ▫ First Normal Form (1NF)
  ▫ Second Normal Form (2NF)
  ▫ Third Normal Form (3NF)
  ▫ Boyce–Codd Normal Form (BCNF)
  ▫ Multi-valued dependencies and Fourth Normal Form (4NF)

©2012 by Chukiat Worasucheep