THE ONE-TO-MANY RELATIONSHIP

Reading: Chapter 4 (pages 89-95)

Objectives

• Understand the one-to-many relationship between two entities
• Model a one-to-many relationship using an automated tool
• Define a database with a one-to-many relationship
DB Design Diagrams

- Once we go beyond single table DBs, diagrams are helpful in design/specification
- Variations in level of diagram
  - Entity Relationship Diagram (ERD)
    - Higher level (entities, not tables)
    - Independent of DB implementation (relational, hierarchical, etc.)
    - Falling out of favor – textbooks favor it, but industry does not
  - Relational Data Model (RDM)
    - Assumes relational implementation
    - Deals with tables and relations
- Variation in notation
  - Dependent on automated tool used

The One-to-Many Relationship

- Entities are related to other entities
- One-to-many is usually shortened to 1:m
- Example
  - Shares of stock are listed on an exchange in a country
  - One country may have many stocks listed
  - A given stock is listed in only one country
  - The connector between the two entities shows the relationship
Relationship

- Identifies the multiplicity between two entities (e.g., one-to-one, one-to-many, many-to-many)
- Illustrated in the RDM
- We use the crows feet notation for relationships
  - Many side shows a crows foot
  - One side shows a simple connector
  - Vertical lines show minimum number

Helpful if ERDs resemble OO models

ER Modeling

- Provide a visual representation of data
- Top-down approach to database design
- Identify
  - Entities (implemented as tables)
  - relationships between the entities
- Add details, such as
  - information about the entities and relationships
  - constraints on the entities, relationships, and attributes.

Our emphasis is primarily notation, with emphasis on concrete design (entity modeling is typically done through OO SW design)
Multiplicity Constraints

• Multiplicity constraints on relationships
  • Number of occurrences of one entity that may relate to a single occurrence of an associated entity

  E.g., One nation lists many stocks

E-R Notation

• Chen
• IDEF1X
• Bachman
• Crow’s Foot
• ISO
• UML

Some data modeling techniques (e.g., Chen) falling into disuse as OO modeling (e.g., UML) dominates

Access diagrams in Design View use a different notation
Hierarchical Relationships

- Form of 1-m relationship
- Occurs frequently
- Multiple 1:m relationships

Example – Shares DB

- Techniques to represent the Nations information
  1. Include Nation information in the Shares entity
  2. Provide Nation as a separate entity
- Including Nations information in the Shares entity might cause anomalies
Single Table Approach

- Add columns for nationName and exchangeRate

<table>
<thead>
<tr>
<th>ID</th>
<th>code</th>
<th>shareName</th>
<th>price</th>
<th>quantity</th>
<th>dividend</th>
<th>pe</th>
<th>nationName</th>
<th>exchangeRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FC</td>
<td>Freedomia Copper</td>
<td>$7.75</td>
<td>10524</td>
<td>$1.84</td>
<td>16</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PT</td>
<td>Patagonian Tea</td>
<td>$5.25</td>
<td>12635</td>
<td>$2.50</td>
<td>10</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>AR</td>
<td>Abyssinian Ruby</td>
<td>$1.82</td>
<td>22010</td>
<td>$1.32</td>
<td>13</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SIG</td>
<td>Sri Lankan Gold</td>
<td>$5.30</td>
<td>32980</td>
<td>$2.68</td>
<td>16</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LE</td>
<td>Indian Lead &amp; Zinc</td>
<td>3.75</td>
<td>19088</td>
<td>$3.00</td>
<td>12</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>BE</td>
<td>Burmese Elephant</td>
<td>$0.07</td>
<td>154717</td>
<td>$0.01</td>
<td>3</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BS</td>
<td>Bolivian Sheep</td>
<td>$12.75</td>
<td>231678</td>
<td>$1.78</td>
<td>11</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>NG</td>
<td>Nigerian Goose</td>
<td>$35.00</td>
<td>12323</td>
<td>$1.08</td>
<td>10</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CS</td>
<td>Canadian Sugar</td>
<td>$2.78</td>
<td>4716</td>
<td>$2.50</td>
<td>15</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ROF</td>
<td>Royal Ostrich Farms</td>
<td>33.75</td>
<td>1234923</td>
<td>$3.00</td>
<td>6</td>
<td>UK</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MG</td>
<td>Minnesota Gold</td>
<td>$3.87</td>
<td>81022</td>
<td>$1.00</td>
<td>25</td>
<td>USA</td>
<td>0.67</td>
</tr>
<tr>
<td>12</td>
<td>GP</td>
<td>Georgia Peach</td>
<td>$2.35</td>
<td>387333</td>
<td>$0.20</td>
<td>5</td>
<td>USA</td>
<td>0.67</td>
</tr>
<tr>
<td>13</td>
<td>NE</td>
<td>Narembaan Emu</td>
<td>$12.34</td>
<td>45619</td>
<td>$1.00</td>
<td>8</td>
<td>Australia</td>
<td>0.46</td>
</tr>
<tr>
<td>14</td>
<td>GD</td>
<td>Queensland Diamond</td>
<td>$6.73</td>
<td>89251</td>
<td>$0.50</td>
<td>7</td>
<td>Australia</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Update Anomalies

- Tables that contain redundant information may potentially suffer from update anomalies
- Types of update anomalies include:
  - Insertion – how do you insert details of a new nation (e.g., exchange rate) that has no stock?
  - Deletion – when we delete the last stock, we lose the information about the nation
  - Modification – changes to a nation must be made for all records containing that nation

We will deal with this issue in more detail later in the semester
Mapping to a Relational Database

- Each entity becomes a table
- The entity name becomes the table name
- Each attribute becomes a column
- Add a column to the table at the many end of a 1:m relationship
- Put the identifier of the one end in the added column

Nations Table

- We create a new table using the definition of the Nations entity

### Nations Table

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>code</th>
<th>nationName</th>
<th>exchangeRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UK</td>
<td>United Kingdom</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>USA</td>
<td>United States</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>AUS</td>
<td>Australia</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>IND</td>
<td>India</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

India exchange rate is displayed with 2 digits of precision, but stored as .0228
Foreign Keys

- The 1:m relationship between 2 entities is implemented in a DB with a foreign key
- A foreign key is a column whose values are the primary keys of another table

Foreign Key

- nationID in the new Shares table is a foreign key that refers to the Nations table
Referential Integrity Constraint

- For every value of a foreign key there is a primary key with that value
  - E.g., for every value of `nationID` in Shares there is a value of `ID` in `Nations`
- A foreign key can never be null
- A primary key must exist before the foreign key can be defined
  - Must create Nations before Shares

Relationships in Access

- To create a relationship in Access, first click on Relationships in Database Tools tab
Relationships in Access

• Then click on Create New … to bring up the Create New dialog box

Relationships in Access

• For every value of a foreign key there is a primary key with that value
Relationships in Access

• Enforce referential integrity

Design View – Multiple Tables

• Access will show the relationships in Design View
Are We on Track?

- Create a new DB with two tables (Shares and Nations)
  - Create the Nations table with the following data
  - Add a foreign key to the Shares table using data in slide 11

<table>
<thead>
<tr>
<th>ID</th>
<th>code</th>
<th>nationName</th>
<th>exchangeRate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UK</td>
<td>United Kingdom</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>USA</td>
<td>United States</td>
<td>0.67</td>
</tr>
<tr>
<td>3</td>
<td>AUS</td>
<td>Australia</td>
<td>0.46</td>
</tr>
<tr>
<td>4</td>
<td>IND</td>
<td>India</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Did You Meet the Objectives?

- Model a one-to-many relationship between two entities
- Model a one-to-many relationship using an automated tool
- Define a database with a one-to-many relationship