Data Modeling: Part 1

Entity Relationship (ER) Model

MBA 8473

Cognitive Objectives (Module 2)

32. Explain the three-step process of data-driven information system (IS) development
33. Examine the purpose of data modeling in information management
34. Explain Data Model quality issues.
35. Provide examples of four common difficulties in a poorly designed data model: (1) redundancies, (2) update, (3) deletion anomalies, and (4) insertion anomalies.
36. Explain entity-relationship (ER) model and identify its building blocks: Entity, Identifier, Relationship, and Attribute.
37. Explain three properties of an entity-relationship: (1) Degree of a relationship, (2.1) Minimum and Maximum Cardinalities, and (2.2) Optional and Mandatory Associations
38. Do an ER Model or Diagram.
39. Explain relational model, and identify its building blocks, and underlying core principles: (1) Tables and attributes, (2) Primary keys, and entity integrity, (3) Foreign keys and referential Integrity.
40. Apply rules for mapping E-R models to the relational model
41. Apply normalization theory to assess the quality of a relational model
Organization of Concepts

Data Modeling
- The Rationale
  - Shared Understanding
  - Uncover Relationships
  - Prevent Anomalies
    - Insert
    - Update
    - Delete
- E-R Model
  - Building Blocks
    - Entity
    - Attributes
    - Relationships
    - Identifiers
  - Types of Relationships
    - Degree of a Relationship
    - Min. & Max. Cardinalities
    - 1:1, 1:M, M:N
    - Optional & Mandatory
    - Recursive
    - Supertype - Subtype
- Relational Model
  - Building Blocks
    - Tables
    - Attributes
    - Primary Key
    - Foreign key
  - Integrity
    - Primary Key
    - Referential
    - Domain
  - Data Quality
    - Normalization Theory

Data-Driven IS Development (C.O. 32)

Problem Domain

Conceptual Design

Logical Design

Physical Design
Why is Data Modeling important for managing organizational information? (C.O. 33)

- The emphasis of data modeling is on representing reality, business complexity.
- A database must mirror the real world. Only then can it answer questions about the real world!
- Ideally it should be a graphical representation of both business “reality” and the actual database.

The goal is to identify the facts to be stored in the database.

A high quality Data Model should offer (C.O. 34)

- A well-formed data model (completeness)
- A high fidelity (clarity plus reliability) image
Difficult 1: Redundant Data (C.O. 35)

Consider the following table that stores data about auto parts and suppliers. This seemingly harmless table contains many potential problems.

<table>
<thead>
<tr>
<th>Part#</th>
<th>Description</th>
<th>Supplier</th>
<th>Address</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Coil</td>
<td>Dynar</td>
<td>45 Eastern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>101</td>
<td>Muffler</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>102</td>
<td>Wheel Cover</td>
<td>A1 Auto</td>
<td>7441 E. 4th Street</td>
<td>Detroit</td>
<td>MI</td>
</tr>
<tr>
<td>103</td>
<td>Battery</td>
<td>Dynar</td>
<td>45 E. 8th Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>104</td>
<td>Radiator</td>
<td>United Parts</td>
<td>346 Taylor Drive</td>
<td>Austin</td>
<td>TX</td>
</tr>
<tr>
<td>105</td>
<td>Manifold</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>106</td>
<td>Converter</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>107</td>
<td>Tail Pipe</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
</tbody>
</table>

Suppose you want to add another part?

Memory space is wasted by duplicating data about the supplier. Every time a new part is entered for a particular supplier, all of the supplier data is repeated. Imagine the problems if several suppliers supply hundreds of auto parts each.

Difficult 2: Update Anomaly (C.O. 35)

What if GlassCo moves to Olympia? How many rows have to change in order to ensure that the new address is recorded.

<table>
<thead>
<tr>
<th>Part#</th>
<th>Description</th>
<th>Supplier</th>
<th>Address</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Coil</td>
<td>Dynar</td>
<td>45 Eastern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>101</td>
<td>Muffler</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>102</td>
<td>Wheel Cover</td>
<td>A1 Auto</td>
<td>7441 E. 4th Street</td>
<td>Detroit</td>
<td>MI</td>
</tr>
<tr>
<td>103</td>
<td>Battery</td>
<td>Dynar</td>
<td>45 E. 8th Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>104</td>
<td>Radiator</td>
<td>United Parts</td>
<td>346 Taylor Drive</td>
<td>Austin</td>
<td>TX</td>
</tr>
<tr>
<td>105</td>
<td>Manifold</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>106</td>
<td>Converter</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>107</td>
<td>Tail Pipe</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
</tbody>
</table>

Again, imagine the issues surrounding modifications of hundreds of rows of data for one supplier. When changes are made, they must be made to all copies of the data. Think about the confusion that results from changing only a subset of the duplicate data.
**Difficulty 3: Deletion Anomaly (C.O. 35)**

Suppose you no longer carried part number 102 and decided to delete that row from the table?

<table>
<thead>
<tr>
<th>Part#</th>
<th>Description</th>
<th>Supplier</th>
<th>Address</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Coil</td>
<td>Dynar</td>
<td>45 Eastern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>101</td>
<td>Muffler</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>102</td>
<td>Wheel Cover</td>
<td>A1 Auto</td>
<td>7441 E. 4th Street</td>
<td>Detroit</td>
<td>MI</td>
</tr>
<tr>
<td>103</td>
<td>Battery</td>
<td>Dynar</td>
<td>45 Eastern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>104</td>
<td>Radiator</td>
<td>United Parts</td>
<td>346 Taylor Drive</td>
<td>Austin</td>
<td>TX</td>
</tr>
<tr>
<td>105</td>
<td>Manifold</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>106</td>
<td>Converter</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>107</td>
<td>Tail Pipe</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
</tbody>
</table>

**Now, looking at the remaining data below, what is the address of A1 Auto? (C.O. 35)**

<table>
<thead>
<tr>
<th>Part#</th>
<th>Description</th>
<th>Supplier</th>
<th>Address</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Coil</td>
<td>Dynar</td>
<td>45 Eastern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>101</td>
<td>Muffler</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>104</td>
<td>Radiator</td>
<td>United Parts</td>
<td>346 Taylor Drive</td>
<td>Austin</td>
<td>TX</td>
</tr>
<tr>
<td>105</td>
<td>Manifold</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>106</td>
<td>Converter</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>107</td>
<td>Tail Pipe</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
</tbody>
</table>

A deletion anomaly means that we lose more information than we want. We lose facts about more than one subject with one deletion.
Difficulty 4: Insertion Anomaly (C.O. 35)

Next, you want to add a new supplier, CarParts, but you have not yet ordered parts from that supplier. What do you add?

<table>
<thead>
<tr>
<th>Part#</th>
<th>Description</th>
<th>Supplier</th>
<th>Address</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Coil</td>
<td>Dynar</td>
<td>45 Eastern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>101</td>
<td>Muffler</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>103</td>
<td>Battery</td>
<td>Dynar</td>
<td>45 Estern Ave.</td>
<td>Denver</td>
<td>CO</td>
</tr>
<tr>
<td>104</td>
<td>Radiator</td>
<td>United</td>
<td>346 Taylor Drive</td>
<td>Austin</td>
<td>TX</td>
</tr>
<tr>
<td>105</td>
<td>Manifold</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>106</td>
<td>Converter</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>107</td>
<td>Tail Pipe</td>
<td>GlassCo</td>
<td>1638 S. Front</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>???</td>
<td>?????????</td>
<td>CarParts</td>
<td>101 Mariposa</td>
<td>Orlando</td>
<td>FL</td>
</tr>
</tbody>
</table>

This situation is called an insertion anomaly. Stated reversely, we cannot add a fact about one subject until we have additional data about another subject.

Turning to The Building Blocks of an ER (Entity Relationship) Model (C.O. 36)

- Entity
  - Attribute
  - Identifier

- Relationship
Entities and Their Instances (C.O. 36)

• What is an Entity?
  – Some “thing” or object in the environment that occurs more than once and we have a recognized business interest in capturing information about this “thing”
  – Classes of real-world objects
    • PERSON, STUDENT, EMPLOYEE
  – Typically represented by a rectangle
  – An instance is a particular occurrence of an entity

<table>
<thead>
<tr>
<th>Entity</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT</td>
<td>AL, BILLY, CHARLES</td>
</tr>
</tbody>
</table>

Attributes (C.O. 36)

• Attributes describe an entity
  • Attributes such as Student_name, Student_social_sec#, … so on, describe an entity called Student for a given purpose)
    – Some recommendations:
      • Attribute names must be unique within a data model
        – Not recommended: Student (…, soc_sec#, …); Professor (…, soc_sec#, …)
        – Recommended: Student (…, stud_soc_sec#, …); Professor (…, prof_soc_sec#, …)
      • Attribute names must be meaningful
        – Not recommended: ss#
        • Recommended: soc_sec# or social_security_number
      • Differentiate between attribute name and attribute value
        – Attribute name: social_security_number
        – Attribute value: 234-56-7890, 234567890
Identifiers (C.O. 36)

• (Why?) Every instance of an entity must be uniquely identified
  – Entity: Student
  – Instances (Name attribute only):
    • John Smith, Jane Doe, John Smith, …
    • We need some attribute other than the Name to differentiate the first John Smith from the second John Smith
• (What?) An identifier can be one attribute or a collection of attributes
  – One attribute as an identifier of the entity called STUDENT:
    social_security_number
  – A collection of attributes as an identifier of the entity called
    REGISTRATION: soc_sec_number+course_number
• (How to denote an identifier) An underlined attribute (or a leading asterisk) denotes an identifier:
  – social_security_number or *social_security_number and NOT
    social_security_number
• Discuss how to recognize poor identifiers.

Entity Relationship Properties (C.O. 37)

• In the real world, entities are related to other entities.
• Relationships are most often depicted using a diamond shaped rectangle or simply by a straight line.
• An “appropriate” label describes the relationship between entities
Example of an entity relationship (37)

This relationship captures a business fact:
A department has employee(s) or stated otherwise, an employee belongs to a department.

What it does not tell us yet: how many of one entity relates to how many of the other. For example, how many employee belongs to a department? Such information is called “Cardinality” of the relationship.

Cardinality of Relationships [C.O. 37(2)]

- Minimum Cardinality
  - The minimum number of instances of one entity that must participate in a relationship
    - What are the minimum number of employees that must be in a department? Example: 0, 1, 5, N?

- Maximum Cardinality
  - The maximum number of instances of one entity that can participate in a relationship
    - What are the maximum number of employees that can be in a department? Example: 1, 10, 40, M?
Examples of minimum and maximum cardinalities for each entity (37.2)

In this scheme:
A DEPARTMENT is allowed to have 1 or Many EMPLOYEES, so in this example
minimum cardinality of EMPLOYEE is 1 and maximum cardinality of EMPLOYEE is
M (or Many).
(Note: A department of 0 employee is not allowed and will not exist in the database)

In this scheme:
An EMPLOYEE is allowed to belong to one and only one DEPARTMENT, i.e., only one
department can participate in this relationship. So, minimum cardinality of
DEPARTMENT is 1 and maximum cardinality of DEPARTMENT is also 1. (Note: (1)
An employee without a department is not allowed and will not exist in the database. (2)
No two departments can ‘have’ the same employee in the database!)

Examples of minimum and maximum cardinality (37.2)

As a practice we show both entity (or side)
cardinalities in one diagram
ER Model Draws upon Cardinality-Based Classification of Relationships (C.O. 37.2.1 and 37.2.2)

- Broadly classified in terms of maximum cardinality
- A 1:1 [one-to-one] relationship
- A 1:M [one-to-many] relationship
- A M:N [many-to-many] relationship

[Illustration follows]

One-to-one (1:1) Relationship (C.O.37.2.1 contd.)

- One and only one employee is assigned one office space only.
- A particular office space is assigned to one and only one employee only.
One-to-Many (1:M) Relationship (C.O.37.2.1 contd.)
(Example 1)

DEPT
*deptname
depthfloor
depthphone

EMP
*empno
empfname
empsalary

• A department can have many employees, but an employee can be a member of only one department.

One-to-Many (1:M) Relationship (C.O.37.2.1 contd.)
(Example 1)

CUSTOMER
*Number
Name
Address

ORDER
*Order Number
Order Date
Promised Date

- A customer can place many orders, but has at least one order.
- An order can be placed by one and only one customer.
- This is a one-to-many with minimum cardinalities of one (as stated)
- What implications does this have for collection of customer data?
Optional and Mandatory Association

- An instance of an entity can exist without being related to an instance of the other entity. Then the minimum cardinality is 0. Then the relationship becomes an optional association.
  - A particular customer may not have placed an order and can still be tracked for different purposes.

Many-to-Many (M:M) Relationships (C.O.38 contd.)

- A student can take many classes.
- Each class can have many students.
Doing an Entity-Relationship Model or Diagram (C.O. 38)

• Recap that:
  – Each entity represented by a rectangle.
  – Each relationship represented by a diamond or a straight line.
  – Attributes shown in ovals or as a list.
  – ER Model draws upon Cardinality-Based Classification of Relationships

In-class exercise 1 (c.o. 38)

• A customer can place an order for one or more products. Customers that have not placed any orders can be included in the customer database for purposes of marketing research. Products that have not been ordered can be part of the product database. An order has to be associated with at least one product but can be associated with many products.

• Develop an E-R model for the described scenario. Where necessary, please make any assumptions. It is normally okay to do this, as complex scenarios are unlikely to be described completely. However, it is very important to explicitly state these assumptions.
In-class exercise 2 (c.o. 38)

- In a city called eCity there are various types of licenses that are given out to different businesses. A business can ask for more than one license type. The city mayor wants to keep track of licenses given out to each business. Develop an ER model for the situation. List your assumptions.

Appendix

- Degree of a relationship between entities
- Some popular notations for cardinality
- Some shots from an Access implementation.
- Examples of small ER models.
Degree of a relationship between entities

- **Degree of a relationship** is determined by the number of participating entities
  - unary [one entity - also called recursive relationships]
  - binary [two participating entities, most frequently found]
  - ternary [three participating entities]

Two Examples of Unary (or Recursive) of Relationship (37.1) (A recursive relationship relates an entity to itself.)

Example 1

Example 2
Example of Binary Relationship (37.1)

- Two entities are involved
  - DEPARTMENT
  - EMPLOYEE

Example of Ternary Relationship (37.1)

- Three entities are involved
  - MOTHER
  - FATHER
  - CHILD
Cardinality Notations  
(c.o. 38)

<table>
<thead>
<tr>
<th>Cardinality Interpretation</th>
<th>Minimum Instances</th>
<th>Maximum Instances</th>
<th>Graphic Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exactly one</td>
<td>1</td>
<td>1</td>
<td><img src="example" alt="Graphic" /></td>
</tr>
<tr>
<td>Zero or one</td>
<td>0</td>
<td>1</td>
<td><img src="example" alt="Graphic" /></td>
</tr>
<tr>
<td>One or more</td>
<td>1</td>
<td>Many (&gt;1)</td>
<td><img src="example" alt="Graphic" /></td>
</tr>
<tr>
<td>Zero, one or more</td>
<td>0</td>
<td>Many (&gt;1)</td>
<td><img src="example" alt="Graphic" /></td>
</tr>
<tr>
<td>More than one</td>
<td>&gt;1</td>
<td>&gt;1</td>
<td><img src="example" alt="Graphic" /></td>
</tr>
</tbody>
</table>

MS Access will let you represent an Entity as a Table. You can create a table in three ways as shown above. “Create Table Using Wizard” comes with some already designed tables for some everyday entities. For example, I wanted to create one Table for DEPARTMENT and one for EMPLOYEE. The wizard already has a ready to use version of EMPLOYEE table. You should check to see whether it is good for your purpose.
Note: A table called EMPLOYEE has been created. I have entered one instance of the entity as data. This is a Data Sheet view. One should go to Design view (just click on View and choose Design) to change data specifications.

This is a Design view of the table called EMPLOYEE. Note the choices under Data Type for EmployeeID.
Example 1: Student-Course

Design a database to keep track of what courses a student takes, the grade he or she receives and information on what professors teach the courses.

Entities: Student, Course, Professor

Relationships:   Student takes Course
                N :  M
                Professor teaches Course
                N :  M

E-R Model (Diagram)
Example 2: Supplier-Part ERD
(c.o. 38 continues)

Supplier sends Part: Relationship