Chapter Eight Data Modeling and Analysis

- Define data modeling and explain its benefits.
- Recognize and understand the basic concepts and constructs of a data model.
- Read and interpret an entity relationship data model.
- Explain when data models are constructed during a project and where the models are stored.
- Discover entities and relationships.
- Construct an entity - relationship context diagram.
- Discover or invent keys for entities and construct a key - based diagram.
- Construct a fully attributed entity relationship diagram and describe all data structures and attributes to the repository or encyclopedia.
- Normalize a logical data model to remove impurities that can make a database unstable, inflexible, and non - scalable.
- Describe a useful tool for mapping data requirements to business operating locations.

Data modeling – a technique for organizing and documenting a system’s data. Sometimes called database modeling.

Entity relationship diagram (ERD) – a data model utilizing several notations to depict data in terms of the entities and relationships described by that data.

Entity – a class of persons, places, objects, events, or concepts about which we need to capture and store data.

- Named by a singular noun
  - **Persons:** agency, contractor, customer, department, division, employee, instructor, student, supplier.
  - **Places:** sales region, building, room, branch office, campus.
  - **Objects:** book, machine, part, product, raw material, software license, software package, tool, vehicle model, vehicle.
  - **Events:** application, award, cancellation, class, flight, invoice, order, registration, renewal, requisition, reservation, sale, trip.
  - **Concepts:** account, block of time, bond, course, fund, qualification, stock.
**Entity instance** – a single occurrence of an entity.

<table>
<thead>
<tr>
<th>Entity instance</th>
<th>2144</th>
<th>Arnold</th>
<th>Betty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3122</td>
<td>Taylor</td>
<td>John</td>
</tr>
<tr>
<td></td>
<td>3843</td>
<td>Simmons</td>
<td>Lisa</td>
</tr>
<tr>
<td></td>
<td>9644</td>
<td>Macy</td>
<td>Bill</td>
</tr>
<tr>
<td></td>
<td>2837</td>
<td>Leah</td>
<td>Heather</td>
</tr>
<tr>
<td></td>
<td>2203</td>
<td>Wrench</td>
<td>Tim</td>
</tr>
</tbody>
</table>

**Data type** – a property of an attribute that identifies what type of data can be stored in that attribute.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Logical Business Meaning</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER</td>
<td>Any number, real or integer</td>
<td>For integers, specify the range. For real numbers, specify the range and precision.</td>
</tr>
<tr>
<td>TEXT</td>
<td>A string of characters, inclusive of numbers. When numbers are included in a TEXT attribute, it means that we do not expect to perform arithmetic or comparisons with those numbers.</td>
<td>Maximum size of attribute. Actual values are usually infinite; however, users may specify certain narrative restrictions.</td>
</tr>
<tr>
<td>MEMO</td>
<td>Same as TEXT but of an indeterminate size. Some business systems require the ability to attach potentially lengthy notes to a give database record.</td>
<td>Variation on the MMDDYYYY format.</td>
</tr>
<tr>
<td>DATE</td>
<td>Any date in any format.</td>
<td>For AM/PM times: HHMMT</td>
</tr>
<tr>
<td>TIME</td>
<td>Any time in any format.</td>
<td>For military (24-hour) times: HHMM</td>
</tr>
<tr>
<td>YES/NO</td>
<td>An attribute that can assume only one of these two values.</td>
<td>(YES, NO)</td>
</tr>
<tr>
<td>VALUE SET</td>
<td>A finite set of values. In most cases, a coding scheme would be established. (e.g., F=Freshman, SO=Sophomore, JR=Junior, SR=Senior).</td>
<td>(values1, values2, valuesn)</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Any picture or image.</td>
<td>(M=Male, F=Female)</td>
</tr>
</tbody>
</table>

**Domain** – a property of an attribute that defines what values an attribute can legitimately take on.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Domain</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER</td>
<td>Domain</td>
<td>Examples</td>
</tr>
<tr>
<td>TEXT</td>
<td>Maximum size of attribute. Actual values are usually infinite; however, users may specify certain narrative restrictions.</td>
<td>Text(30)</td>
</tr>
<tr>
<td>DATE</td>
<td>Variation on the MMDDYYYY format.</td>
<td>MMDDYYYY</td>
</tr>
<tr>
<td>TIME</td>
<td>For AM/PM times: HHMMT</td>
<td>HHMM</td>
</tr>
<tr>
<td>YES/NO</td>
<td>(YES, NO)</td>
<td>YES, NO</td>
</tr>
<tr>
<td>VALUE SET</td>
<td>(values1, values2, valuesn)</td>
<td>(M=Male, F=Female)</td>
</tr>
</tbody>
</table>

**Default value** – the value that will be recorded if a value is not specified by the user.

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Interpretation</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM NULL</td>
<td>For an instance of the attribute, if the user does not specify a value, then use this value.</td>
<td>NONE</td>
</tr>
<tr>
<td>REQUIRED NULL</td>
<td>For an instance of the attribute, if the user does not specify a value, then leave it blank.</td>
<td>REQUIRED NULL</td>
</tr>
</tbody>
</table>

**Example**

<table>
<thead>
<tr>
<th>Student</th>
<th>Last Name</th>
<th>First Name</th>
<th>Middle Initial</th>
<th>Address</th>
<th>City</th>
<th>State or Province</th>
<th>Country</th>
<th>Postal Code</th>
<th>Phone Number</th>
<th>Area Code</th>
<th>Exchange Number</th>
<th>Number Within Exchange</th>
<th>Date of Birth</th>
<th>Gender</th>
<th>Race</th>
<th>Major</th>
<th>Grade Point Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macy</td>
<td>Wrench</td>
<td>John</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Key** – an attribute, or a group of attributes, that uniquely identifies an instance of an entity. Synonyms include composite key and compound key.

**Candidate key** – one of a number of keys that may serve as the primary key of an entity. Also called a candidate identifier.

**Primary key** – a candidate key that will most commonly be used to uniquely identify a single entity instance.

**Alternate key** – a candidate key that is not selected to become the primary key is called an alternate key. A synonym is secondary key.
Subsetting criteria – an attribute(s) whose finite values divide all entity instances into useful subsets. Sometimes called inversion entry.

Relationship – a natural business association that exists between one or more entities.

Cardinality – the minimum and maximum number of occurrences of one entity that may be related to a single occurrence of the other entity. Because all relationships are bidirectional, cardinality must be defined in both directions for every relationship.

Degree – the number of entities that participate in the relationship. A relationship between two entities is called a binary relationship. A relationship between different instances of the same entity is called a recursive relationship. A relationship between three entities is called a 3-ary or ternary relationship.
Relationships may exist between more than two entities and are called \( N \)-ary relationships. The example ERD depicts a ternary relationship.

**Foreign key** — a primary key of an entity that is used in another entity to identify instances of a relationship.
- A foreign key is a primary key of an entity that is contributed to (duplicated in) another entity to identify instances of a relationship.
- A foreign key always matches the primary key in the other entity.
- A foreign key may or may not be unique (generally not).
- The entity with the foreign key is called the child.
- The entity with the matching primary key is called the parent.

**Nonidentifying relationship** — a relationship in which each participating entity has its own independent primary key
- Primary key attributes are not shared.
- The entities are called strong entities.

**Identifying relationship** — a relationship in which the parent entity’s key is also part of the primary key of the child entity.
- The child entity is called a weak entity.

**Associative entity** — an entity that inherits its primary key from more than one other entity (called parents).
- Each part of that concatenated key points to one and only one instance of each of the connecting entities.
Data Modeling Concepts: Nonspecific Relationships

Nonspecific relationship – a relationship where many instances of an entity are associated with many instances of another entity. Also called many-to-many relationship.

Nonspecific relationships must be resolved. Most nonspecific relationships can be resolved by introducing an associative entity.

Resolving Nonspecific Relationships (continued)

Many-to-many relationships can be resolved with an associative entity.

Generalization – a concept wherein the attributes that are common to several types of an entity are grouped into their own entity.

Supertype – an entity whose instances store attributes that are common to one or more entity subtypes.

Subtype – an entity whose instances may inherit common attributes from its entity supertype. And then add other attributes that are unique to the subtype.
Strategic Data Modeling
- Many organizations select IS development projects based on strategic plans.
  - Includes vision and architecture for information systems
  - Identifies and prioritizes develop projects
  - Includes enterprise data model as starting point for projects

Data Modeling during Systems Analysis
- Data model for a single information system is called an application data model.
  - Context data model includes only entities and relationships.

Logical Model Development Stages
1. Context Data model
   - To establish project scope
2. Key-base data model
   - Eliminate nonspecific relationships
   - Add associative entities
   - Include primary and alternate keys
   - Precise cardinalities
3. Fully attributed data model
   - All remaining attributes
   - Subsetting criteria
4. Normalized data model

JRP and Interview Questions for Data Modeling
- Is each business activity or event handled the same way, or are there special circumstances?
- Discover cardinalities
- What events occur that imply associations between subjects?
- Discover relationships?
- Are all instances of each subject the same?
- Discover generalization hierarchies
- How often does the data change?
- Discover data timing needs
- Are there any restrictions on who can see or use the data?
- Discover security and control needs
- What characteristics describe each subject?
- Discover attributes and domains
- Are there any characteristics of a subject that divide all instances of the subject into useful subsets?
- Discover entity subsetting criteria
- What unique characteristic (or characteristics) distinguishes an instance of each subject from other instances of the same subject?
- Discover entity keys
- What are the subjects of the business?
- Discover system entities
- Discover data entities

Candidate Questions
(see Table 8-4 in text for a more complete list)

Purpose | Candidate Questions
--- | ---
Discover system entities | What are the subjects of the business?
Discover entity keys | What unique characteristic (or characteristics) distinguishes an instance of each subject from other instances of the same subject?
Discover attributes and domains | What characteristics describe each subject?
Discover security and control needs | Are there any restrictions on who can see or use the data?
Discover generalization hierarchies | Are all instances of each subject the same?
Discover relationships? | What events occur that imply associations between subjects?
Discover cardinalities | Is each business activity or event handled the same way, or are there special circumstances?

Automated Tools for Data Modeling

Entity Discovery for SoundStage

<table>
<thead>
<tr>
<th>Business Definition</th>
<th>Entity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A monthly or quarterly event whereby special product offerings are made available to members.</td>
<td>Promotion</td>
</tr>
<tr>
<td>An inventoried product available for promotion and sale to members.</td>
<td>Product</td>
</tr>
<tr>
<td>A business event to which the Member Services System must respond.</td>
<td>Transaction</td>
</tr>
<tr>
<td>An order generated for a member as part of a monthly promotion, or an order initiated by a member.</td>
<td>Member order</td>
</tr>
<tr>
<td>A contract whereby a member agrees to purchase a certain number of products within a certain time. After fulfilling that agreement the member becomes eligible for bonus credits that are redeemable for free or discounted products.</td>
<td>Agreement</td>
</tr>
</tbody>
</table>

Member: An active member of one or more clubs. Not a target system objective is to re-enroll inactive members as opposed to deleting them.

Member order: An order generated for a member as part of a monthly promotion, or an order initiated by a member. Note: The current system only supports orders generated from promotions; however, customer initiated orders have been given a high priority in an added option in the proposed system.

Transaction: A business event to which the Member Services System must respond.

Product: An inventoried product available for promotion and sale to members. Note: System improvement objectives include (1) compatibility with new bar code system being developed for the warehouse, and (2) adaptability to a rapidly changing mix of products.

Promotion: A monthly or quarterly event whereby special product offerings are made available to members.
The Context Data Model

The Key-based Data Model

The Key-based Data Model With Generalization

The Fully-Attributed Data Model

What is a Good Data Model?

- A good data model is simple.
  - Data attributes that describe any given entity should describe only that entity.
  - Each attribute of an entity instance can have only one value.

- A good data model is essentially nonredundant.
  - Each attribute, other than foreign keys, describes at most one entity.
  - Look for the same attribute recorded more than once under different names.

- A good data model should be flexible and adaptable to future needs.

Data analysis – a technique used to improve a data model for implementation as a database.
Goal is a simple, nonredundant, flexible, and adaptable database.

Normalization – a data analysis technique that organizes data into groups to form nonredundant, stable, flexible, and adaptive entities.
First normal form (1NF) – an entity whose attributes have no more than one value for a single instance of that entity
- Any attributes that can have multiple values actually describe a separate entity, possibly an entity and relationship.

Second normal form (2NF) – an entity whose nonprimary-key attributes are dependent on the full primary key.
- Any nonkey attributes that are dependent on only part of the primary key should be moved to any entity where that partial key is actually the full key. This may require creating a new entity and relationship on the model.

Third normal form (3NF) – an entity whose nonprimary-key attributes are not dependent on any other non-primary key attributes.
- Any nonkey attributes that are dependent on other nonkey attributes must be moved or deleted. Again, new entities and relationships may have to be added to the data model.

Derived attribute – an attribute whose value can be calculated from other attributes or derived from the values of other attributes.
Third Normal Form Example 2

SoundStage 3NF Data Model

Data-to-Location -CRUD Matrix