Chapter 5
Advanced Data Modeling
Learning Objectives

- In this chapter, you will learn:
  - About the extended entity relationship (EER) model
  - How entity clusters are used to represent multiple entities and relationships
  - The characteristics of good primary keys and how to select them
  - How to use flexible solutions for special data-modeling cases
Extended Entity Relationship Model (EERM)

- Result of adding more semantic constructs to the original entity relationship (ER) model
- **EER diagram (EERD)**: Uses the EER model
Entity Supertypes and Subtypes

- **Entity supertype**: Generic entity type related to one or more entity subtypes
  - Contains common characteristics
- **Entity subtype**: Contains unique characteristics of each entity subtype
- Criteria to determine the usage
  - There must be different, identifiable kinds of the entity in the user’s environment
  - The different kinds of instances should each have one or more attributes that are unique to that kind of instance
Specialization Hierarchy

- Depicts arrangement of higher-level entity supertypes and lower-level entity subtypes
- Relationships are described in terms of “is-a” relationships
- Subtype exists within the context of a supertype
- Every subtype has one supertype to which it is directly related
- Supertype can have many subtypes
Specialization Hierarchy

- Provides the means to:
  - Support attribute inheritance
  - Define a special supertype attribute known as the subtype discriminator
  - Define disjoint/overlapping constraints and complete/partial constraints
Figure 5.2 - Specialization Hierarchy

This figure illustrates the specialization hierarchy with the following components:

- **Supertype**: EMPLOYEE
- **Dependent**: DEPENDENT
- **Attributes**: EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_HIRE_DATE, EMP_TYPE, DPNT_NUM, DPNT_FNAM, DPNT_LNAME, DPNT_RELATION
- **Constraints**: Disjoint/Overlapping (d/o), Partial/Complete
- **Discriminator**: EMP_TYPE
- **Subtypes**: PILOT, MECHANIC, ACCOUNTANT
- **Unique Attributes**: PIL_LICENSE, PIL_RATINGS, PIL_MED_TYPE, MEC_TITLE, MEC_CERT, ACT_TITLE, ACT_CPA_DATE

The diagram shows the relationships and attributes between the supertype and subtypes, with arrows indicating the inheritance and uniqueness of attributes.
Inheritance

- Enables an entity subtype to inherit attributes and relationships of the supertype
- All entity subtypes inherit their primary key attribute from their supertype
- At the implementation level, supertype and its subtype(s) maintain a 1:1 relationship
- Entity subtypes inherit all relationships in which supertype entity participates
- Lower-level subtypes inherit all attributes and relationships from its upper-level supertypes
Subtype Discriminator

- Attribute in the supertype entity that determines to which entity subtype the supertype occurrence is related
- Default comparison condition is the equality comparison
Disjoint and Overlapping Constraints

- **Disjoint subtypes**: Contain a unique subset of the supertype entity set
  - Known as **nonoverlapping subtypes**
  - Implementation is based on the value of the subtype discriminator attribute in the supertype

- **Overlapping subtypes**: Contain nonunique subsets of the supertype entity set
  - Implementation requires the use of one discriminator attribute for each subtype
Figure 5.4 - Specialization Hierarchy with Overlapping Subtypes
Table 5.1 - Discriminator Attributes with Overlapping Subtypes

<table>
<thead>
<tr>
<th>DISCRIMINATOR ATTRIBUTES</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>The Employee is a member of the Professor subtype.</td>
</tr>
<tr>
<td>N</td>
<td>The Employee is a member of the Administrator subtype.</td>
</tr>
<tr>
<td>Y</td>
<td>The Employee is both a Professor and an Administrator.</td>
</tr>
</tbody>
</table>
Completeness Constraint

- Specifies whether each supertype occurrence must also be a member of at least one subtype

- Types
  - **Partial completeness**: Not every supertype occurrence is a member of a subtype
  - **Total completeness**: Every supertype occurrence must be a member of any
Table 5.2 - Specialization Hierarchy Constraint Scenarios

<table>
<thead>
<tr>
<th>TYPE</th>
<th>DISJOINT CONSTRAINT</th>
<th>OVERLAPPING CONSTRAINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial</td>
<td>Supertype has optional subtypes. Subtype discriminator can be null. Subtype sets are unique.</td>
<td>Supertype has optional subtypes. Subtype discriminators can be null. Subtype sets are not unique.</td>
</tr>
<tr>
<td>Total</td>
<td>Every supertype occurrence is a member of only one subtype. Subtype discriminator cannot be null. Subtype sets are unique.</td>
<td>Every supertype occurrence is a member of at least one subtype. Subtype discriminators cannot be null. Subtype sets are not unique.</td>
</tr>
</tbody>
</table>
## Specialization and Generalization

**Specialization**
- Top-down process
- Identifies lower-level, more specific entity subtypes from a higher-level entity supertype
- Based on grouping unique characteristics and relationships of the subtypes

**Generalization**
- Bottom-up process
- Identifies a higher-level, more generic entity supertype from lower-level entity subtypes
- Based on grouping common characteristics and relationships of the subtypes
Entity Cluster

- Virtual entity type used to represent multiple entities and relationships in ERD
- Avoid the display of attributes to eliminate complications that result when the inheritance rules change
Figure 5.5 - Tiny College ERD Using Entity Clusters
Primary Keys

- Single attribute or a combination of attributes, which uniquely identifies each entity instance
  - Guarantees entity integrity
  - Works with foreign keys to implement relationships
Natural Keys or Natural Identifier

- Real-world identifier used to uniquely identify real-world objects
  - Familiar to end users and forms part of their day-to-day business vocabulary
  - Also known as natural identifier
  - Used as the primary key of the entity being modeled
Desirable Primary Key Characteristics

- Non intelligent
- No change over time
- Preferably single-attribute
- Preferably numeric
- Security-compliant
Use of Composite Primary Keys

- Identifiers of composite entities
  - Each primary key combination is allowed once in M:N relationship

- Identifiers of weak entities
  - Weak entity has a strong identifying relationship with the parent entity
Use of Composite Primary Keys

- When used as identifiers of weak entities, represent a real-world object that is:
  - Existence-dependent on another real-world object
  - Represented in the data model as two separate entities in a strong identifying relationship
Figure 5.6 - The M:N Relationship between STUDENT and CLASS

Table name: STUDENT
(first four fields)

<table>
<thead>
<tr>
<th>STU_NUM</th>
<th>STU_LNAME</th>
<th>STU_FNAME</th>
<th>STU_INIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>321452</td>
<td>Bowser</td>
<td>William</td>
<td>C</td>
</tr>
<tr>
<td>324257</td>
<td>Smithson</td>
<td>Anne</td>
<td>K</td>
</tr>
<tr>
<td>324258</td>
<td>Brewer</td>
<td>Juliette</td>
<td></td>
</tr>
<tr>
<td>324269</td>
<td>Oblonski</td>
<td>Walker</td>
<td>H</td>
</tr>
<tr>
<td>324273</td>
<td>Smith</td>
<td>John</td>
<td>D</td>
</tr>
<tr>
<td>324274</td>
<td>Katanga</td>
<td>Raphael</td>
<td>P</td>
</tr>
<tr>
<td>324291</td>
<td>Robertson</td>
<td>Gerald</td>
<td>T</td>
</tr>
<tr>
<td>324299</td>
<td>Smith</td>
<td>John</td>
<td>B</td>
</tr>
</tbody>
</table>

Table name: ENROLL

<table>
<thead>
<tr>
<th>CLASS_CODE</th>
<th>STU_NUM</th>
<th>ENROLLGRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10014</td>
<td>321452</td>
<td>C</td>
</tr>
<tr>
<td>10014</td>
<td>324257</td>
<td>B</td>
</tr>
<tr>
<td>10018</td>
<td>321452</td>
<td>A</td>
</tr>
<tr>
<td>10018</td>
<td>324257</td>
<td>B</td>
</tr>
<tr>
<td>10021</td>
<td>321452</td>
<td>C</td>
</tr>
<tr>
<td>10021</td>
<td>324257</td>
<td>C</td>
</tr>
</tbody>
</table>

Table name: CLASS
(first three fields)

<table>
<thead>
<tr>
<th>CLASS_CODE</th>
<th>CRG_CODE</th>
<th>CLASS_SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>10012</td>
<td>ACCT-211</td>
<td>1</td>
</tr>
<tr>
<td>10013</td>
<td>ACCT-211</td>
<td>2</td>
</tr>
<tr>
<td>10014</td>
<td>ACCT-211</td>
<td>3</td>
</tr>
<tr>
<td>10015</td>
<td>ACCT-212</td>
<td>1</td>
</tr>
<tr>
<td>10016</td>
<td>ACCT-212</td>
<td>2</td>
</tr>
<tr>
<td>10017</td>
<td>CIS-220</td>
<td>1</td>
</tr>
<tr>
<td>10018</td>
<td>CIS-220</td>
<td>2</td>
</tr>
<tr>
<td>10019</td>
<td>CIS-220</td>
<td>3</td>
</tr>
<tr>
<td>10020</td>
<td>CIS-420</td>
<td>1</td>
</tr>
<tr>
<td>10021</td>
<td>QM-261</td>
<td>1</td>
</tr>
<tr>
<td>10022</td>
<td>QM-261</td>
<td>2</td>
</tr>
<tr>
<td>10023</td>
<td>QM-362</td>
<td>1</td>
</tr>
<tr>
<td>10024</td>
<td>QM-362</td>
<td>2</td>
</tr>
<tr>
<td>10025</td>
<td>MATH-243</td>
<td>1</td>
</tr>
</tbody>
</table>
Surrogate Keys

- Primary key used to simplify the identification of entity instances are useful when:
  - There is no natural key
  - Selected candidate key has embedded semantic contents or is too long
- Require ensuring that the candidate key of entity in question performs properly
  - Use unique index and not null constraints
Table 5.4 - Data Used to Keep Track of Events

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME_START</th>
<th>TIME_END</th>
<th>ROOM</th>
<th>EVENT_NAME</th>
<th>PARTY_OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/17/2014</td>
<td>11:00AM</td>
<td>2:00PM</td>
<td>Allure</td>
<td>Burton Wedding</td>
<td>60</td>
</tr>
<tr>
<td>6/17/2014</td>
<td>11:00AM</td>
<td>2:00PM</td>
<td>Bonanza</td>
<td>Adams Office</td>
<td>12</td>
</tr>
<tr>
<td>6/17/2014</td>
<td>3:00PM</td>
<td>5:30PM</td>
<td>Allure</td>
<td>Smith Family</td>
<td>15</td>
</tr>
<tr>
<td>6/17/2014</td>
<td>3:30PM</td>
<td>5:30PM</td>
<td>Bonanza</td>
<td>Adams Office</td>
<td>12</td>
</tr>
<tr>
<td>6/18/2014</td>
<td>1:00PM</td>
<td>3:00PM</td>
<td>Bonanza</td>
<td>Boy Scouts</td>
<td>33</td>
</tr>
<tr>
<td>6/18/2014</td>
<td>11:00AM</td>
<td>2:00PM</td>
<td>Allure</td>
<td>March of Dimes</td>
<td>25</td>
</tr>
<tr>
<td>6/18/2014</td>
<td>11:00AM</td>
<td>12:30PM</td>
<td>Bonanza</td>
<td>Smith Family</td>
<td>12</td>
</tr>
</tbody>
</table>

Cengage Learning © 2015
Design Case 1: Implementing 1:1 Relationships

- Foreign keys work with primary keys to properly implement relationships in relational model

- Rule
  - Put primary key of the parent entity on the dependent entity as foreign key

- Options for selecting and placing the foreign key:
  - Place a foreign key in both entities
  - Place a foreign key in one of the entities
Table 5.5 - Selection of Foreign Key in a 1:1 Relationship

<table>
<thead>
<tr>
<th>CASE</th>
<th>ER RELATIONSHIP CONSTRAINTS</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>One side is mandatory and the other side is optional.</td>
<td>Place the PK of the entity on the mandatory side in the entity on the optional side as a FK, and make the FK mandatory.</td>
</tr>
<tr>
<td>II</td>
<td>Both sides are optional.</td>
<td>Select the FK that causes the fewest nulls, or place the FK in the entity in which the (relationship) role is played.</td>
</tr>
<tr>
<td>III</td>
<td>Both sides are mandatory.</td>
<td>See Case II, or consider revising your model to ensure that the two entities do not belong together in a single entity.</td>
</tr>
</tbody>
</table>
Figure 5.7 - The 1:1 Relationship between Department and Employee
Design Case 2: Maintaining History of Time-Variant Data

- **Time-variant data**: Data whose values change over time and for which a history of the data changes must be retained
  - Requires creating a new entity in a 1:M relationship with the original entity
  - New entity contains the new value, date of the change, and other pertinent attribute
Figure 5.8 - Maintaining Salary History
Figure 5.9 - Maintaining Manager History
Figure 5.10 - Maintaining Job History
Design Case 3: Fan Traps

- **Design trap**: Occurs when a relationship is improperly or incompletely identified
  - Represented in a way not consistent with the real world
- **Fan trap**: Occurs when one entity is in two 1:M relationships to other entities
  - Produces an association among other entities not expressed in the model
Figure 5.11 - Incorrect ERD with Fan Trap Problem
Figure 5.12 - Corrected ERD After Removal of the Fan Trap
Design Case 4: Redundant Relationships

- Occur when there are multiple relationship paths between related entities
- Need to remain consistent across the model
- Help simplify the design
Figure 5.13 - A Redundant Relationship