Entity Relationship Model

✓ The Entity Relationship Model
  • Database design
  • Entity, Entity Set, Attribute, Relationship

• E/R Design Considerations
  • Constraints: Key, Referential, Degree
  • Relationship Conditions: Multiplicity of Relationships, Multiway Relationships

• More E/R Concepts
  • Combining Relations, Constraints, Subclasses, Weak Entity Sets

• Conversion to SQL
Database Design Process

Requirements Analysis

• What data is going to be stored?

• What are we going to do with the data?

• Who should access the data?
Database Design Process

Conceptual Design

• A high-level description of the database
• Sufficiently precise that technical people can understand it
• But, not so precise that non-technical people can not participate

✓Where E/R Model Fits
Database Design Process

Detailed Design

• A Logical Database Design

• Physical Database Design

• Security Design
Entity Relationship Model

• Principal element types

  • **Entity**: Individual object

  • **Entity Sets**: Collection of similar entities forms an entity set
    ER model is a static concept, involving the structure of data and **not the operations on data**

  • **Attributes**: Entity sets have attributes, showing properties of the entities in that set

  • **Relationships**: Connections among two or more entity sets
Entity Relationship Model

**Example:** Movie Database

- Each movie is an entity, and the set of all movies constitutes an entity set.
- The stars are entities, and the set of stars is an entity set.
- A studio is another kind of entity, and the set of studios is a third entity set.

- The entity set Movies might be given attributes such as title and length.

- Movies and Stars are two entity sets, we could have a relationship Stars-in that connects movies and stars.

- Entities are **not** explicitly represented in E/R diagrams!
ER Diagrams

A graph representing entity sets, attributes, and relationships

- **Entity sets** are represented by rectangles.
- **Attributes** are represented by ovals.
- **Relationships** are represented by diamonds.
Keys

• A **key** is a set of attributes that uniquely identifies an entity.

• Every entity must have a key, although in some cases (we will see later) the key actually belongs to another entity set.

• Denote elements of the primary key by **underlining**.

• There is no notation for representing the situation where there are several keys for an entity set.

• We underline only the primary key.

• Multiple underlining: they are each member of the key.
Keys

• The attributes **title** and **year** together form the key for **Movies**.
What is a Relationship?

• Let $A$ and $B$ be sets
  
  $A$={1,2,3}, $B$={a,b,c,d}

• $A \times B$ (the cross-product) is the set of all pairs $(a,b)$
  
  $A \times B = \{(1,a), (1,b), (1,c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d)\}$

• We define a relationship to be a subset of $A \times B$
  
  $R = \{(1,a), (2,c), (2,d), (3,b)\}$
What is a Relationship?

- $A = \{1, 2, 3\}$
  $B = \{a, b, c, d\}$

- $A \times B = \{(1,a), (1,b), (1,c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d)\}$

- $R = \{(1,a), (2,c), (2,d), (3,b)\}$
A relationship between entity sets P and C is a subset of all possible pairs of entities in P and C, with tuples uniquely identified by P and C’s keys.
What is a Relationship?

### Company

<table>
<thead>
<tr>
<th>name</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>Electronics</td>
<td>$700</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Electronics</td>
<td>$300</td>
</tr>
<tr>
<td>Office</td>
<td>Software</td>
<td>$120</td>
</tr>
</tbody>
</table>

### Product

<table>
<thead>
<tr>
<th>name</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone 8</td>
<td>Electronics</td>
<td>$700</td>
</tr>
<tr>
<td>iPad 4</td>
<td>Electronics</td>
<td>$300</td>
</tr>
<tr>
<td>Office</td>
<td>Software</td>
<td>$120</td>
</tr>
</tbody>
</table>

### Company C × Product P

<table>
<thead>
<tr>
<th>C.name</th>
<th>P.name</th>
<th>P.category</th>
<th>P.price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>iPhone 8</td>
<td>Electronics</td>
<td>$700</td>
</tr>
<tr>
<td>Apple</td>
<td>iPad 4</td>
<td>Electronics</td>
<td>$300</td>
</tr>
<tr>
<td>Apple</td>
<td>Office</td>
<td>Software</td>
<td>$120</td>
</tr>
<tr>
<td>Microsoft</td>
<td>iPhone 8</td>
<td>Electronics</td>
<td>$700</td>
</tr>
<tr>
<td>Microsoft</td>
<td>iPad 4</td>
<td>Electronics</td>
<td>$300</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Office</td>
<td>Software</td>
<td>$120</td>
</tr>
</tbody>
</table>

### Makes

<table>
<thead>
<tr>
<th>C.name</th>
<th>P.name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>iPhone 8</td>
</tr>
<tr>
<td>Apple</td>
<td>iPad 4</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Office</td>
</tr>
</tbody>
</table>
What is a Relationship?

• There can only be **one relationship for every unique combination of entities**

• This also means that **the relationship is uniquely determined by the keys of its entities**

• Example: the key for Makes is
  
  \{Product.name, Company.name\}

<table>
<thead>
<tr>
<th>C.name</th>
<th>P.name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>iPhone 8</td>
</tr>
<tr>
<td>Apple</td>
<td>iPad 4</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Office</td>
</tr>
</tbody>
</table>
Entity Relationship Model

- The Entity Relationship Model
  - Database design
  - Entity, Entity Set, Attribute, Relationship, Key

✓ E/R Design Considerations
  - Constraints: Referential Constraints, Degree Constraints
  - Relationship Conditions: Multiplicity of Relationships, Multiway Relationships

- More E/R Concepts
  - Combining Relations, Constraints, Subclass, Weak Entity Sets

- Conversion to SQL
Referential Integrity

• A value appearing in one context must also appear in another

• A rounded arrow head pointing to $F$
  • Not only the relationship is many-one from $E$ to $F$
  • But also the entity of set $F$ related to a given entity of set $E$ is required to exist

E/R diagram showing referential integrity constraints
Degree Constraints

- A bounding number on the edges that connect a relationship to an entity set

- Indicates limits on the number of entities that can be connected to any one entity of the related entity set

Representing a constraint on the number of stars per movie

From Chapter 4, The complete book
Multiplicity of Binary ER Relationships

- If each member of E can be connected by R to at most one member of F: R is many-one from E to F.

- If R is both many-one from E to F and many-one from F to E: R is one-one.

- If R is neither many-one from E to F nor from F to E: R is many-many.
Multiplicity of ER Relations

- Usage of arrows

\[ X \rightarrow Y \text{ means} \]

There exists a function mapping from \( X \) to \( Y \)

(Recall the definition of a function)
Multiway Relationships

- Relationships involving more than two entity sets

- A multiway relationship in an E/R diagram is represented by lines from the relationship diamond to each of the involved entity sets.

A ternary (three-way) relationship

From Chapter 4, The complete book
Roles in Relationships

• It is possible that one entity set appears two or more times in a single relationship.

• Each line to the entity set represents a different role that the entity set plays in the relationship.

• We label the edges between the entity set and relationship by names (roles).

A relationship with roles
Attributes in Relationships

- It is never necessary to place attributes on relationships.
- Could be replaced by a new entity set with the attributes.

A relationship with an attribute

From Chapter 4, The complete book
Attributes in Relationships

Moving the attribute to an entity set

From Chapter 4, The complete book
A Note on Different Notations

Each product made by at most one company.

Referential Integrity.

Each product made by exactly one company.
Acknowledgements

I have used materials from the following resources in preparation of this course:

• **Database Systems: The Complete Book**
• Database Systems (Kifer, Bernstein, Lewis)
• Database System Concepts: [https://www.db-book.com](https://www.db-book.com)
• Course offerings
  • CMPT 354 (Jiannan Wang - SFU): [https://sfu-db.github.io/cmpt354/](https://sfu-db.github.io/cmpt354/)
  • W 4111 (Eugene Wu - Columbia): [https://w4111.github.io/](https://w4111.github.io/)
  • CS 186 (Joe Hellerstein - Berkeley): [https://sites.google.com/site/cs186fall17/](https://sites.google.com/site/cs186fall17/)