DS 1300 - The Entity-Relationship Model (ER Model)

By Michael Hahsler
Based on slides for CS145 Introduction to Databases (Stanford)
Introduction to Database Design
Database Design

• Database design: Why do we need it?
  • Agree on structure of the database before deciding on a particular implementation

• Consider issues such as:
  • What entities to model
  • How entities are related
  • What constraints exist in the domain
  • How to achieve good designs

• Several formalisms exist
  • We discuss one flavor of ER diagrams

Data Scientists use it to decide what data to collect and/or how to organize data for analysis.
Database Design Process

1. Requirements Analysis

• What is going to be stored?
• How is it going to be used?
• What are we going to do with the data?
• Who should access the data?
2. 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 cannot participate
Database Design Process

1. Requirements Analysis
2. Conceptual Design
3. Logical, Physical, Security, etc.

3. Implementation:

• Logical Database Design

• Physical Database Design

• Security Design
ER is a visual syntax for DB design which is precise enough for technical points, but abstracted enough for non-technical people.
Impact of the ER model

• The ER model is one of the most cited articles in Computer Science
  • “The Entity-Relationship model – toward a unified view of data”
    Peter Chen, 1976

• Used by companies big and small
1. ER Basics: Entities & Relations
Entities and Entity Sets

- **Entities & entity types** are the primitive units of the ER model

  - **Entities** are the individual objects (instances), which are members of entity types
  - **Entity type** are the *classes* or *types* of objects in our model
  - Example: Person is an entity type while Michael is an entity.
  - *We use entity types in ER models*
Entities and Entity Types

- An entity type has **attributes** represented by ovals attached to an entity type.

Shapes *are* important. Colors used here *are not.*
Entities vs. Entity Sets

Example:

<table>
<thead>
<tr>
<th>Product</th>
<th>Entity type</th>
<th>Name: Xbox</th>
<th>Category: Game Console</th>
<th>Price: $250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Entity</td>
<td>Name: My Little Pony Doll</td>
<td>Category: Toy</td>
<td>Price: $25</td>
</tr>
</tbody>
</table>

Entities are **not** explicitly represented in ER diagrams!
A **key** is a **minimal** set of attributes that uniquely identifies an entity.

Denote elements of the primary key by underlining.

Here, \{name, category\} is **not** a key (it is not *minimal*).

If it were, what would it mean?

The ER model forces us to designate a single **primary key**, though there may be multiple candidate keys. Often, we introduce an **artificial key** attribute (also called a **synthetic or surrogate key**).
Entity Types Define Relations

<table>
<thead>
<tr>
<th>Product</th>
<th>name</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>Electronics</td>
<td>$9.99</td>
<td></td>
</tr>
<tr>
<td>GizmoLite</td>
<td>Electronics</td>
<td>$7.50</td>
<td></td>
</tr>
<tr>
<td>Gadget</td>
<td>Toys</td>
<td>$5.50</td>
<td></td>
</tr>
</tbody>
</table>
The R in ER: Relationships

- A relationship type is between two entity types

How to read a relationship in both directions:
1. A product is made by a company
2. A company makes a product
What is a Relationship?

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?

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?

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?

• 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 (to right) is** \{Product.name, Company.name\}

This follows from our mathematical definition of a relationship (it is a set)
Relationships and Attributes

Relationships may have attributes as well.

For example: “since” records when company started making a product.

Note: For each product/company pair there is automatically only a single since value since there can only be one unique product/company pair in makes.
A: A person can only buy a specific product once per day (date)

Modeling something as a relationship makes it unique. **What if this is not appropriate?**
Decision: Relationship vs. Entity?

What about this way?

Now we can have multiple purchases per product, person pair!

We can always use a new entity instead of a relationship. For example, to permit multiple instances of each entity combination!
Note on Relationships vs. Relation

**ER Model:** How do Entity types relate to each other

**Math:** A Relation

**Relational Algebra:** A table with data (a set)

Relations are used to implement entity types and certain relationship types!

<table>
<thead>
<tr>
<th>Product</th>
<th>name</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>Electronics</td>
<td>$9.99</td>
<td></td>
</tr>
<tr>
<td>GizmoLite</td>
<td>Electronics</td>
<td>$7.50</td>
<td></td>
</tr>
<tr>
<td>Gadget</td>
<td>Toys</td>
<td>$5.50</td>
<td></td>
</tr>
</tbody>
</table>
How to Create an ER Diagram

“Rules of thumb" for mapping natural language descriptions into ER diagrams:

<table>
<thead>
<tr>
<th>English grammar structure</th>
<th>ER structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common noun</td>
<td>Entity type</td>
</tr>
<tr>
<td>Proper noun</td>
<td>Entity</td>
</tr>
<tr>
<td>Verb</td>
<td>Relationship type</td>
</tr>
<tr>
<td>Adjective</td>
<td>Attribute for entity</td>
</tr>
<tr>
<td>Adverb</td>
<td>Attribute for relationship</td>
</tr>
</tbody>
</table>
Example: How to Create an ER Diagram

Here is what the person in charge said:

“Our company is called PowerSeller and we sell health products on Ebay. Our products are made by different manufacturers. Products belong to different product categories (e.g., supplements, cosmetics, etc.) and each product is sold at a fixed price. We use customer IDs for our customers, and we know for all of them the shipping address and name, and for most we also know a phone number.”
Example: How to Create an ER Diagram

Here is what the person in charge said:

“Our company is called PowerSeller and we sell health products on Ebay. Our products are made by different manufacturers. Products belong to different product categories (e.g., supplements, cosmetics, etc.) and each product is sold at a fixed price. We use customer IDs for our customers, and we know for all of them the shipping address and name, and for most we also know a phone number.”
Example: How to Create an ER Diagram

Identify entity types and attributes (nouns):
• company, PowerSeller
• health product, product, product category, price, manufacturer
• customerID, customer, shipping address, name, phone number

Identify relationship types (verbs):
• know
• belongs to
• sell
• make
Example: How to Create an ER Diagram

**Identify Entity types (bold) and attributes:**
- company, PowerSeller
- health product, product, product category, price, manufacturer
- customerID, customer, shipping address, name, phone number

**Identify relationship types:**
- know
- belongs to
- sell, buy
- make
Exercise: Draw an ER diagram to store football information

Teams play each other in Games. Each pair of teams can play each other multiple times.

Players belong to Teams (assume no trades / changes).

A Game is made up of Plays that result in a yardage gain/loss, and potentially a touchdown.

A Play will contain either a Pass from one player to another, or a Run by one player.
From ER Diagrams to Relational Schema

• Key concept:

Both *Entity sets* and *Relationships* become relations (tables in RDBMS)
Multiplicity of ER Relationships

Using Chen’s Notation

One-to-one:

Many-to-one:

One-to-many:

Many-to-many:
How to read a relationship in both directions:
1. A product is made by a **one** company
2. A company makes **many** product
From ER Diagrams to Relational Schema

• An entity set becomes a relation (multiset of tuples / table)
  
  − Each tuple is one entity
  
  − Each tuple is composed of the entity’s attributes, and has the same primary key

<table>
<thead>
<tr>
<th>name</th>
<th>price</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo1</td>
<td>99.99</td>
<td>Camera</td>
</tr>
<tr>
<td>Gizmo2</td>
<td>19.99</td>
<td>Edible</td>
</tr>
</tbody>
</table>
CREATE TABLE Product(
  name CHAR(50) PRIMARY KEY,
  price DOUBLE,
  category VARCHAR(30)
)
From ER Diagrams to Relational Schema (N:M)

- A relation between entity sets $A_1, \ldots, A_N$ also becomes a multiset of tuples / a table
  - Each row/tuple is one relation, i.e. one unique combination of entities $(a_1, \ldots, a_N)$
  - Each row/tuple is
    - composed of the union of the entity sets’ attributes
    - has the entities’ primary keys as foreign keys
    - has the union of the entity sets’ keys as primary key

<table>
<thead>
<tr>
<th>Purchased</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>firstname</td>
<td>lastname</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Gizmo1</td>
<td>Bob</td>
<td>Joe</td>
<td>01/01/15</td>
<td></td>
</tr>
<tr>
<td>Gizmo2</td>
<td>Joe</td>
<td>Bob</td>
<td>01/03/15</td>
<td></td>
</tr>
<tr>
<td>Gizmo1</td>
<td>JoeBob</td>
<td>Smith</td>
<td>01/05/15</td>
<td></td>
</tr>
</tbody>
</table>
CREATE TABLE Purchased(
    name CHAR(50),
    firstname CHAR(50),
    lastname CHAR(50),
    date DATE,
    PRIMARY KEY (name, firstname, lastname),
    FOREIGN KEY (name)
        REFERENCES Product,
    FOREIGN KEY (firstname, lastname)
        REFERENCES Person
)

<table>
<thead>
<tr>
<th>name</th>
<th>firstname</th>
<th>lastname</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo1</td>
<td>Bob</td>
<td>Joe</td>
<td>01/01/15</td>
</tr>
<tr>
<td>Gizmo2</td>
<td>Joe</td>
<td>Bob</td>
<td>01/03/15</td>
</tr>
<tr>
<td>Gizmo1</td>
<td>JoeBob</td>
<td>Smith</td>
<td>01/05/15</td>
</tr>
</tbody>
</table>
From ER Diagrams to Relational Schema (1:N)

- A 1:N relationship can be implemented without an extra table.
- Add the primary key of the “1 side” to the table for the “N side” entity.

<table>
<thead>
<tr>
<th>ID</th>
<th>Number</th>
<th>Street</th>
<th>ZIPCode</th>
<th>CustID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>123</td>
<td>Main St</td>
<td>75000</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6660</td>
<td>Willow Dr</td>
<td>86123</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Nowhere Pl</td>
<td>99999-1234</td>
<td>2</td>
</tr>
</tbody>
</table>
From ER Diagrams to Relational Schema (1:N)

CREATE TABLE Address(
  ID CHAR(50),
  Number CHAR(50),
  Street CHAR(50),
  ZIPCode CHAR(10),
  PRIMARY KEY (ID),
  FOREIGN KEY (CustID) REFERENCES Customer,
)

<table>
<thead>
<tr>
<th>ID</th>
<th>Number</th>
<th>Street</th>
<th>ZIPCode</th>
<th>CustID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>123</td>
<td>Main St</td>
<td>75000</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6660</td>
<td>Willow Dr</td>
<td>86123</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Nowhere Pl</td>
<td>99999-1234</td>
<td>2</td>
</tr>
</tbody>
</table>
Exercise: Create the tables the following ER diagram

How do we represent this as a relational schema?
Exercise: Create the tables for your ER Diagram for the football example

Teams play each other in Games. Each pair of teams can play each other multiple times.

Players belong to Teams (assume no trades / changes).

A Game is made up of Plays that result in a yardage gain/loss, and potentially a touchdown.

A Play will contain either a Pass from one player to another, or a Run by one player.