The Entity-Relationship Model
ER Model - Part 2: Conversion to SQL

By Michael Hahsler
Based on slides for CS145 Introduction to Databases (Stanford)
What you will learn about in this section

1. Relationships: multiplicity, multi-way
2. Design considerations
3. Conversion to SQL
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
No specified cardinality often means N:M, or we do not want to decide, yet.
Multi-way Relationships

How do we model “A person buys a product in a store?”
Q: What do the 1s and the N mean?

Multiplicity in Multiway Relationships

- **Product**
  - N
- **Person**
  - 1
- **Store**
  - 1
- **Purchase**
Better: many to many to many relationship
Conversion of Multi-way Relationship to New Entity + Binary Relationships?

Multi-way Relationship  -->  Entity + Binary

Multiple purchases per (product, store, person) possible here!
3. Design Principles

What’s wrong with these examples?

- Product - Purchase - Person
- Country - President - Person
Design Principles: What’s Wrong?

- Product
- Store
- date
- personName
- personAddr

Purchase

11
Design Principles: What’s Wrong? - Fixed
Examples: Entity vs. Attribute

Should address be an attribute?

Or an entity?
Examples: Entity vs. Attribute

Should address be an attribute?

• How do we handle employees with more than two addresses?

• How do we handle addresses where internal structure of the address (e.g. zip code, state) is useful?
Examples: Entity vs. Attribute

Use an entity

In general, when we want to record several values, we choose a separate entity.
From ER Diagrams to Relational Schema

Key concept:

Both *Entity sets* and *Relationships* become relations (tables in RDBMS)
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
CREATE TABLE Product(
  name     CHAR(50) PRIMARY KEY,
  price    DOUBLE,
  category VARCHAR(30)
)
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
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
)
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.
CREATE TABLE Address(
  ID      CHAR(50),
  Number CHAR(50),
  Street  CHAR(50),
  ZIPCode      CHAR(10),
  PRIMARY KEY (ID),
  FOREIGN KEY (CustID)
REFERENCES Customer,
)
How do we represent this as a relational schema?
Alternative Notations
Exercise: Add Multiplicity to your ER diagram

Also make sure to add (new concepts underlined):

- A player can only belong to one team, a play can only be in one game, a pass/run..?
- Multiple players **Tackle** a single person in a play
- Players can achieve a **Personal Record** linked to a specific Game and Play
- Players have a **weight** which changes in on vs. off-season