

CS122 Using Relational Databases and SQL

Introduction to Entity-Relationship Diagram

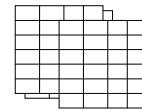
Chengyu Sun
California State University, Los Angeles

Database Design

Problem in Real World

Tables in RDBM

Some Restaurant
Terminal ID: NC2HHRY
Merchant ID: 4492414532566624
VISA *****1234 srv:1
SALE inv:000032
Batch: 000244
Date: JUN 17, 06 Time: 18:44
 AUTH:00559B
Base: \$36.70
Tip:
Total: Chengyu Sun



Entity-Relationship (ER) Model

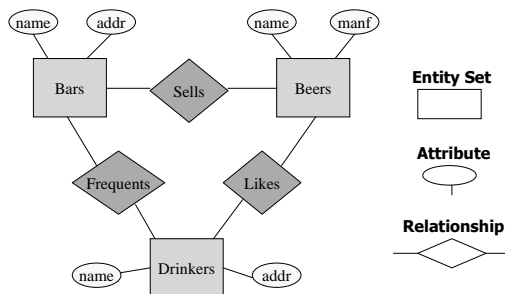
Problem → ER Model → Tables

- ◆ Sort of an object-oriented approach
- ◆ A graphical representation of the design – ER Diagram
- ◆ Easily converted to relational model

Sample Problem

- ◆ Design a database to keep track information about bars, beers, and drinkers
- ◆ Beers – name, manufacturer
 - Name, manufacturer
- ◆ Bars – name, address
 - Bars sell beers
- ◆ Drinkers – name, address
 - Drinkers go to bars and likes beers

ER Diagram

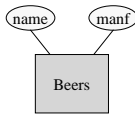


Entity Set and Attributes

- ◆ Entity Set is a collection of entities
 - E.g. Bars, Beers, Drinkers, Products, ...
- ◆ Attributes are the properties of an entity set
 - For example
 - Attributes of Bars: name, address
 - Attributes of Products: id, description, category, price
 - Must have simple values like numbers or strings

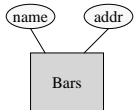
Instances of An Entity Set

Entity Set



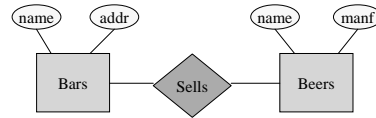
Instances of the Entity Set

(Bud, Anheuser-Busch)
 (Miller, Miller Brewing)
 (Bud Lite, Anheuser-Busch)



(Joe's Bar, 113 Main St)
 (Sue's Bar, 20 East St)

Relationship



Types of Relationships

- ◆ Many-to-Many
- ◆ Many-to-One (One-to-Many)
- ◆ One-to-One

Many-to-Many Relationship



- ◆ **Sells** is a many-to-many relationship
 - One bar can sell many beers
 - One beer can be sold in many bars

Many-to-One Relationship



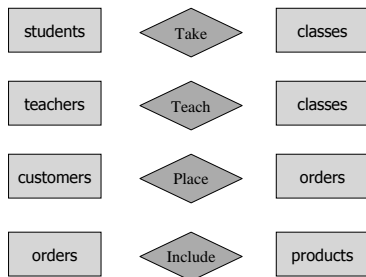
- ◆ **Favorite** is a many-to-one relationship
 - One drinker only has one favorite beer
 - One beer can be the favorites of many drinkers
- ◆ *An arrow is used to indicate the "one" side*

One-to-One Relationship



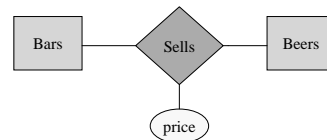
- ◆ **Bestseller** is a one-to-one relationship
 - One manufacturer only has one bestselling beer
 - One beer can only be the bestseller of one manufacturer
- ◆ *Arrows on both sides*

Relationship Examples



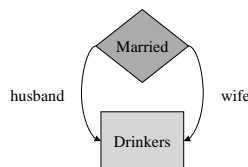
Attributes of Relationships

- ◆ Sometimes it's useful to attach an attribute to a relationship.

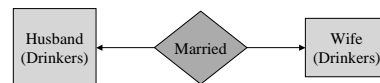


Roles

- ◆ An entity set may appear in the same relationship more than once.
- ◆ Label the edges with names called Roles



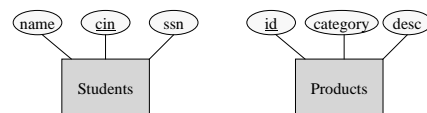
Another Way to Look at Roles



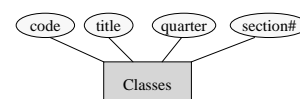
Keys

- ◆ A key is an attribute or a set of attributes that *uniquely* identify an entity in an entity set.
- ◆ Each entity set must have a key
- ◆ If there are multiple keys, choose one of them as the *primary key*

Keys in ER Diagram



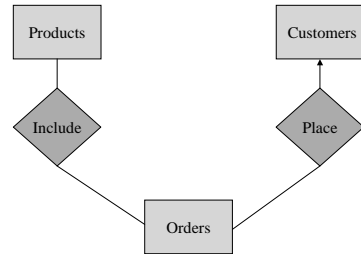
What's the key for Classes??



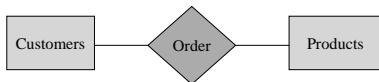
Design the Store Database

- ◆ Keep track information about
 - Products
 - Customers
 - Orders

Design #1: Order as an Entity Set



Design #2: Order as a Relationship

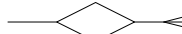
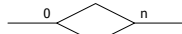
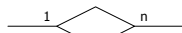


Quick Summary about ER Diagram

- ◆ Entity Sets
 - Attributes
 - ◆ Must have simple values
 - Keys
- ◆ Relationships
 - Many-to-many, many-to-one, one-to-one
 - Attributes

Alternative Notations and Things We Won't Cover

Different ways to represent relationships



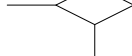
Weak entity set



Subclass



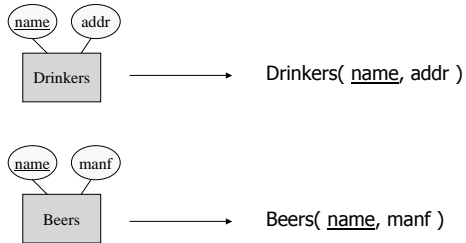
Multi-way Relationship



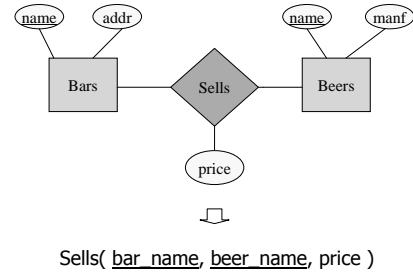
Convert ER Diagram to Relations

- ◆ Entity sets
- ◆ Relationships

Converting Entity Sets



Converting Relationships



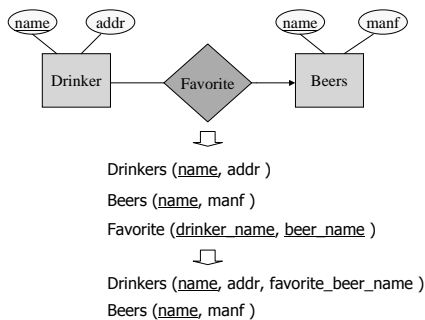
Converting Relationships – General Rules

- ◆ The resulting relation includes
 - All key attributes from the entity sets involved in the relationship
 - All the attributes of the relationship itself

Converting Relationships – Combining Relations

- ◆ The relations converted from *many-to-one* and *one-to-one* relationships can be absorbed into the relation of the "many" side.

Example of Combining Tables



The Store Database

- ◆ Convert the ER diagram from Design #1
- ◆ Convert the ER diagram from Design #2
- ◆ Which one is better??

Normal Forms

- ◆ Formal ways to evaluate the “goodness” of a database design
- ◆ Covered in CS422