The Principle of Database System

Teacher: Li Fang (李芳)

Email: fli@sjtu.edu.cn

TA: li_feng@sjtu.edu.cn

Web Site:

http://lt-lab.sjtu.edu.cn:8080/database

Grading:

Attendance & Homework: 20%

Final examination: 80%

Time Arrangement

Week 1th~ 14th : Lectures

Tuesday 8.00AM-9.40AM E403

Thursday 8.00AM-9.40AM E531

Week 10th ~ 18th: Project

About the course

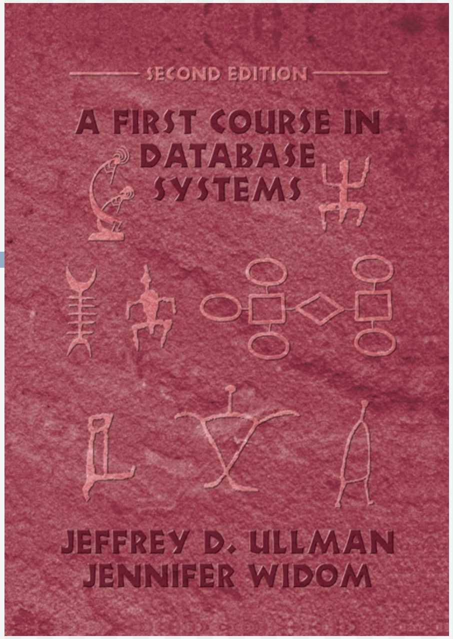
- Textbook

A first course in database systems

(second edition)

Authors:Jeffrey D.Ullman, Jennifer Widom

Stanford University



- Reference web site:

http://www-db.stanford.edu/~ullman/fcdb.html

Other reference books

1) Database System Concepts by Abraham Silberschatz, et al (机械工业出版社)

2）Database Principles, Programming, and Performance by Patrick O’Neil, et al.（高等教育出版社）

3）数据库系统实现 database System Implementation（Stanford university）中，英版（机械工业出版社）

4）数据库系统概论 （萨师煊，王珊）高等教育出版社

About Database Courses

- A first course in database systems

  theory, model, programming language and so on…

- Database system implementation

  Implementation details …

- Database software

  products from those companies:

  Oracle, Sybase, IBM, Microsoft,….

Contents of **the course**

- **Introduction** (what is DBMS? What is Database system?...)
- **Design of database** (how does one build a useful database? What kind of information is stored in database? three kinds of data models will be learned in the lecture.)
- **Database Programming**(how to query and operate on database? Relational algebra,SQL,OQL and Datalog will be discussed in the lecture.)
- **Database new research trends**

Introduction (Chapter 1)

- The history of Database System
- Overview of a database Management System
- Three Aspects of Database-System Studies

Introduction to Database Systems

What is a Database System?

- Database (data , metadata)
- Hardware (disks)
- Software (DBMS)
- People (users, database designers and database administrators DBA)

The history of Database Systems

- Early database management systems evolved from file system
- Network DBMS, Hierarchical DBMS
- **Relational database** Systems(in 1970, Ted Codd proposed tables called relations as a view of data in database)
- **Object-oriented database**
- **Web database and XML**

Purpose of Database System

- **Drawbacks of using file systems to store data:**
  - **Data redundancy and inconsistency**
    - Multiple file formats, duplication of information in different files
  - **Difficulty in accessing data**
    - Need to write a new program to carry out each new task
  - **Data isolation — multiple files and formats**
  - **Integrity problems**
    - Integrity constraints (e.g. account balance > 0) become part of program code
    - Hard to add new constraints or change existing ones

Purpose of Database System (cont.)

- **Atomicity of updates**
  - Failures may leave database in an inconsistent state with partial updates carried out
  - E.g. transfer of funds from one account to another should either complete or not happen at all
- **Concurrent access by multiple users**
  - Concurrent accessed needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - E.g. two people reading a balance and updating it at the same time
- **Security problems**

Database systems provide solutions to all these problems

Some applications of database systems

- **Databases touch all aspects of our lives**
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions …

What do you think about Database Management System?

From User's point of view:

- Collection of interrelated data
- Set of programs to access the data
- DBMS contains information about a particular enterprise
- DBMS provides an environment that is both *convenient* and *efficient* to use.

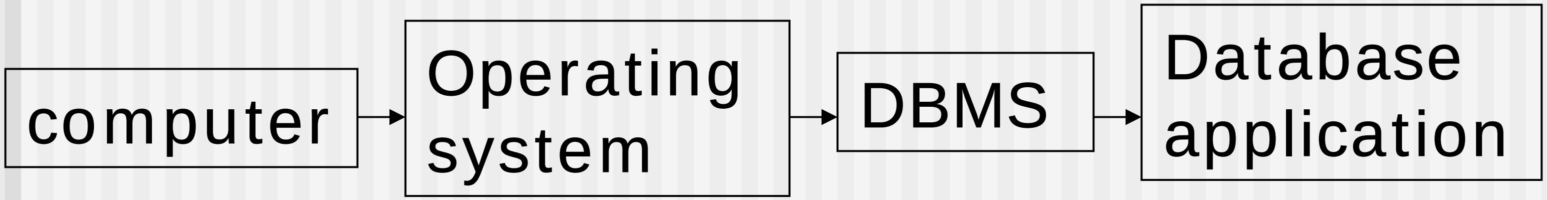
What is a DBMS (database management system) ?

From Commercial Market:

- Oracle, Sybase are among the largest software companies in the world.
- IBM offers its DB2 system
- Microsoft offers SQL-Server & Microsoft Access for DBMS on desktop

DBMS (From its Functions)

- Designing the structure of their information (e.g. relational model)
- Users can operate (query, modify) on the data
- Managing huge amount of data and support efficient,concurrent,secure, atomic access to very large amounts of data



DBMS (how to manage data)

- Persistent storage: supports the storage of very large amounts of data.
- Programming interface:to access and modify data through a powerful query language.
- Transaction management:supports concurrent access to data.
- Concurrent control:supports secure, atomic access to very large amounts of data.

DBMS (Components)

- Query processor

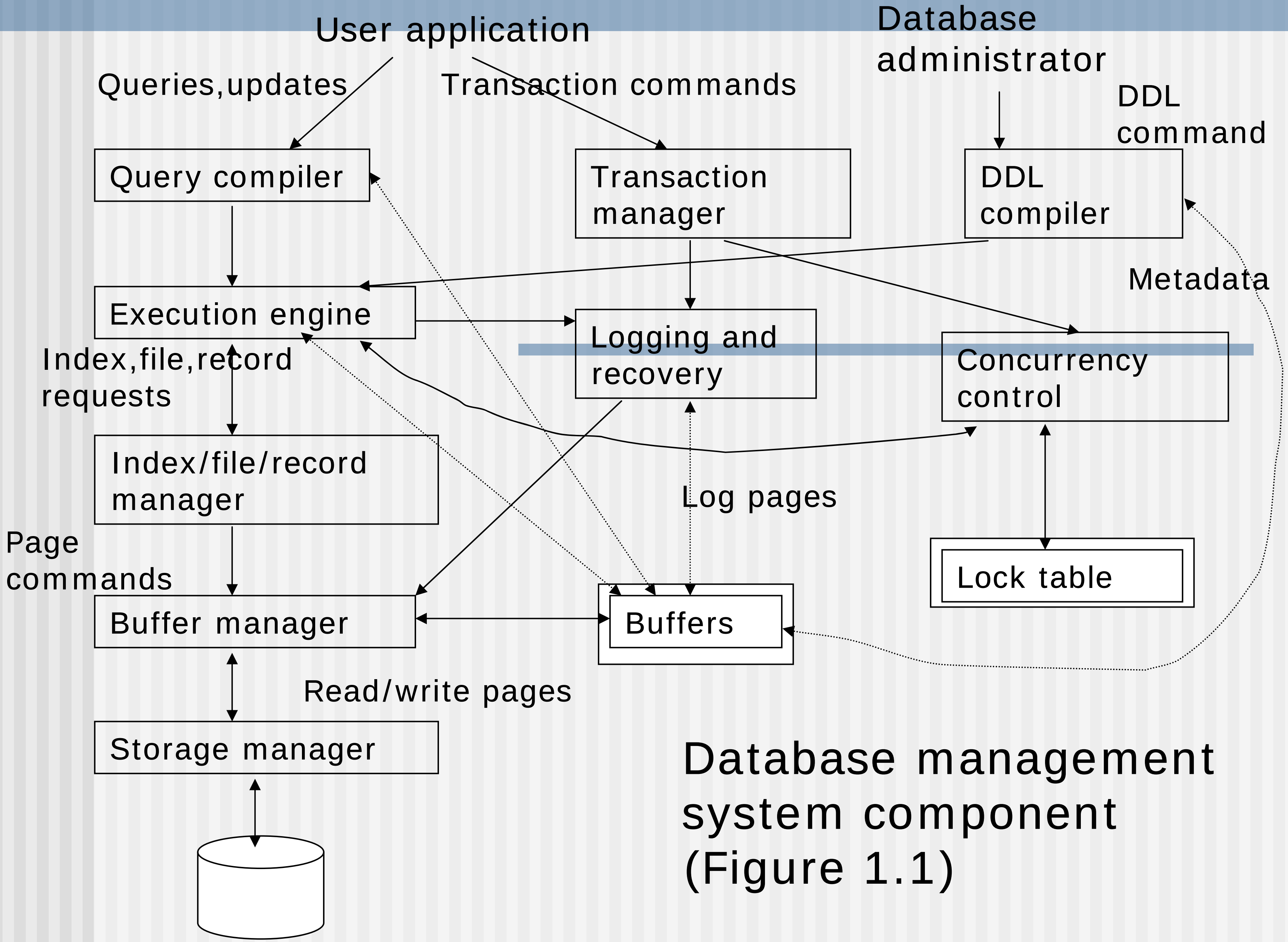
The query compiler, execution engine

- Transaction manager

Logging,concurrency control, deadlock resolution

- Storage manager

Control the placement of data on disk, and its movement between disk and main memory



Database management system component (Figure 1.1)

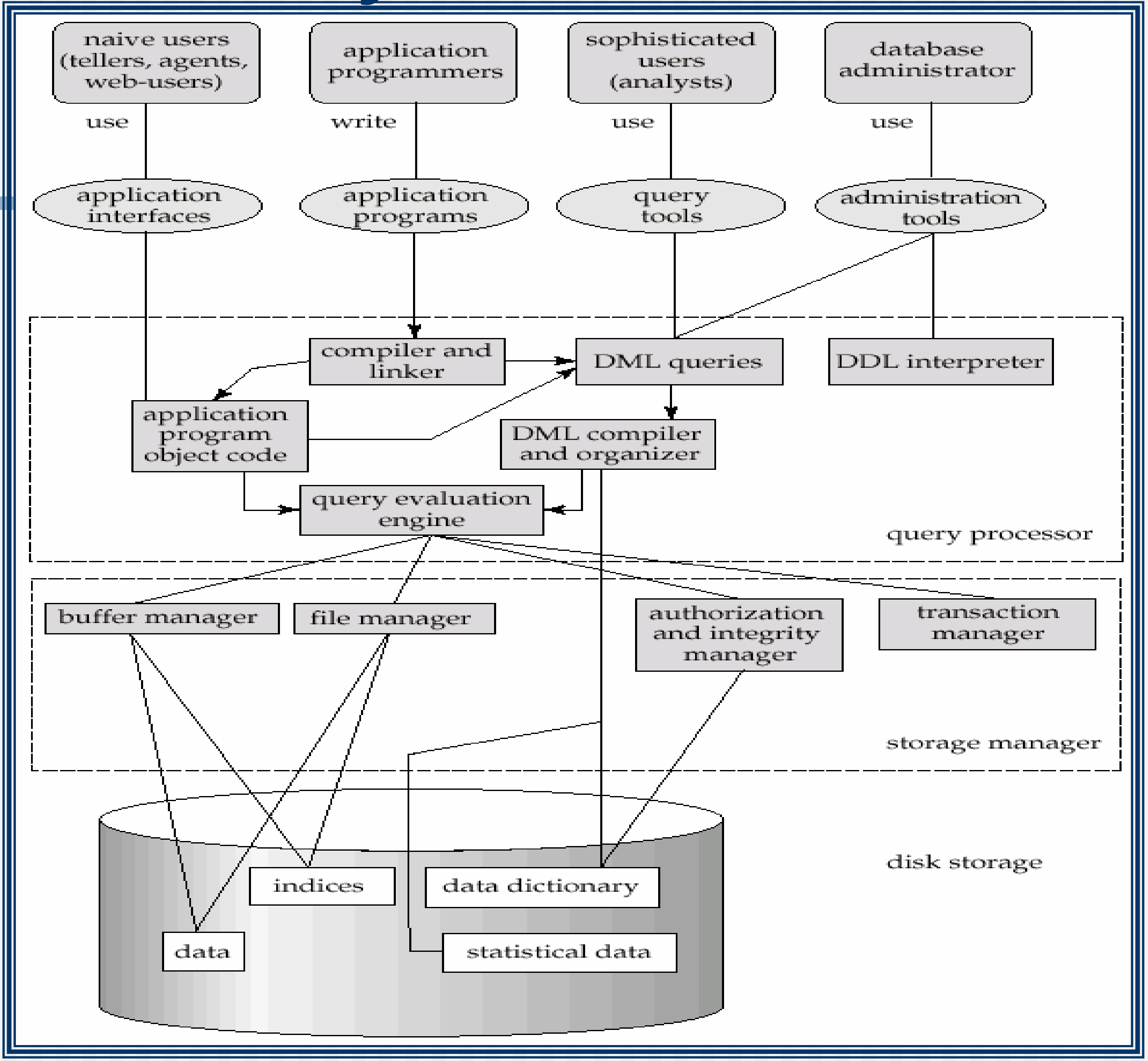
Users are differentiated by the way they expect to interact with the system

- Application programmers – interact with system through database manipulation language
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naïve users – invoke one of the permanent application programs that have been written previously
  - E.g. people accessing database over the web, bank tellers, clerical staff

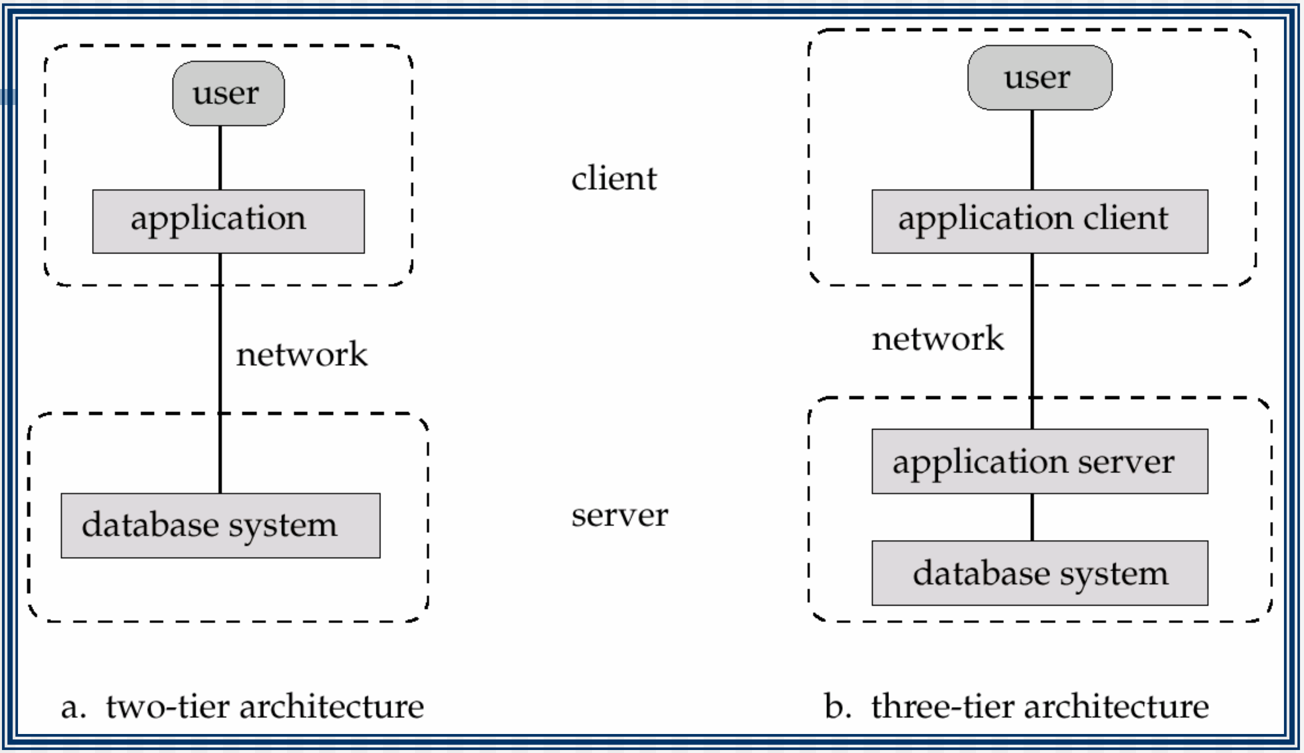
Database Administrator

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
- Database administrator's duties include:
    - Schema definition
    - Storage structure and access method definition
    - Schema and physical organization modification
    - Granting user authority to access the database
    - Specifying integrity constraints
    - Acting as liaison with users
    - Monitoring performance and responding to changes in requirements

Overall System Structure



Application Architectures



- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using “middleware”

Three Aspects to Studying DBMS's

- Modeling and design of databases.(how is the information structured?)
- Programming: queries and DB operations like update.(how does one express queries and other operations on the database)
- DBMS implementation.(how to build a DBMS)

Summary of chapter 1

- Database System & Applications
- Database management System(DBMS)

features,functions,components (storage manager, the query processor, and the transaction manager)

- Database Language (DDL, DML)
- The development of DBMS

Past time:	Now:
Network DBMS	Relational DBMS
Hierarchical DBMS	Object-Relational
Relational DBMS	Object-DBMS

Database Modeling

- Introduction
- Entity-Relationship data model (chapter 2)
- Relational data model (chapter 3)
- Other data models (chapter 4)

Introduction

- 现实世界：客观存在的世界。
- 信息世界：现实世界在人们头脑中的反映。
- 机器世界：信息世界的信息在机器世界中以数据的形式存放。

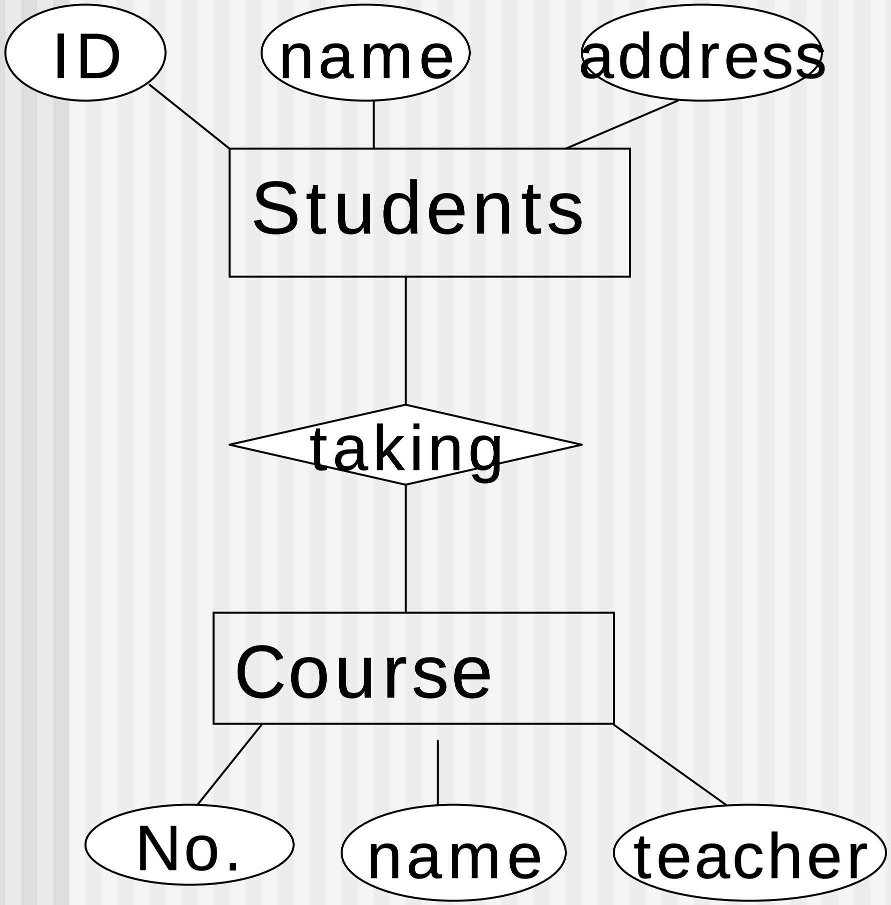
reality-》	information world-》	machine world
	E-R data model	relational model
		object-relational model
	ODL	object-oriented model

Chapter 2: Entity/Relationship Model

- Entity like objects, =things
- Entity set like class = set of similar Entity or objects
- Attribute=property of entities in an entity set, similar to fields of a struct.
- Relation=connect two or more entity set

In diagrams, entity set : rectangle; attribute: oval, relation: diamonds

Entity/Relationship Diagrams: example



Relationships

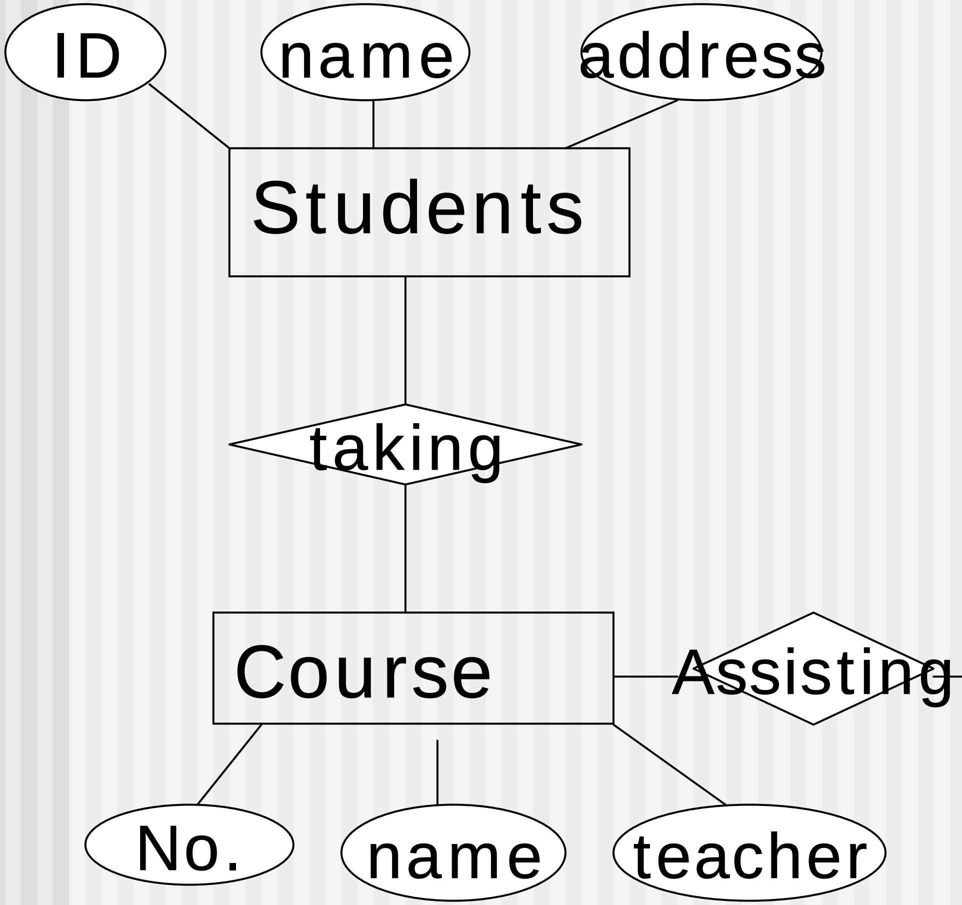
- Binary (relation between two entity sets)
- Multiway (relation between more than two entity sets)
- Multiplicity of relationships

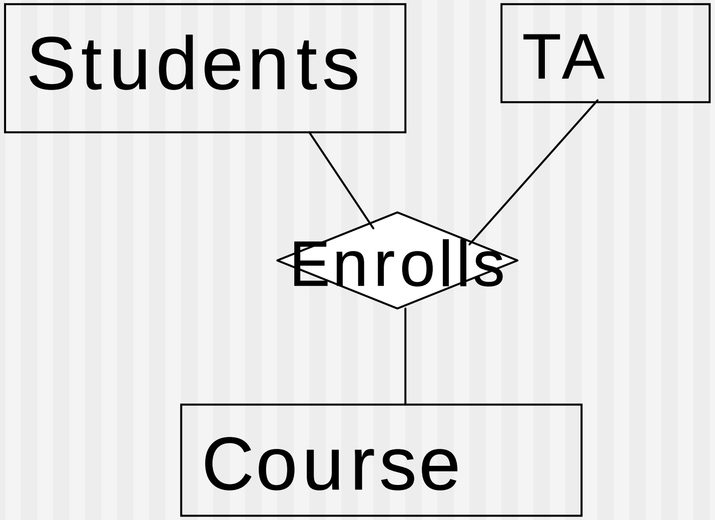
Many to many

Many to one

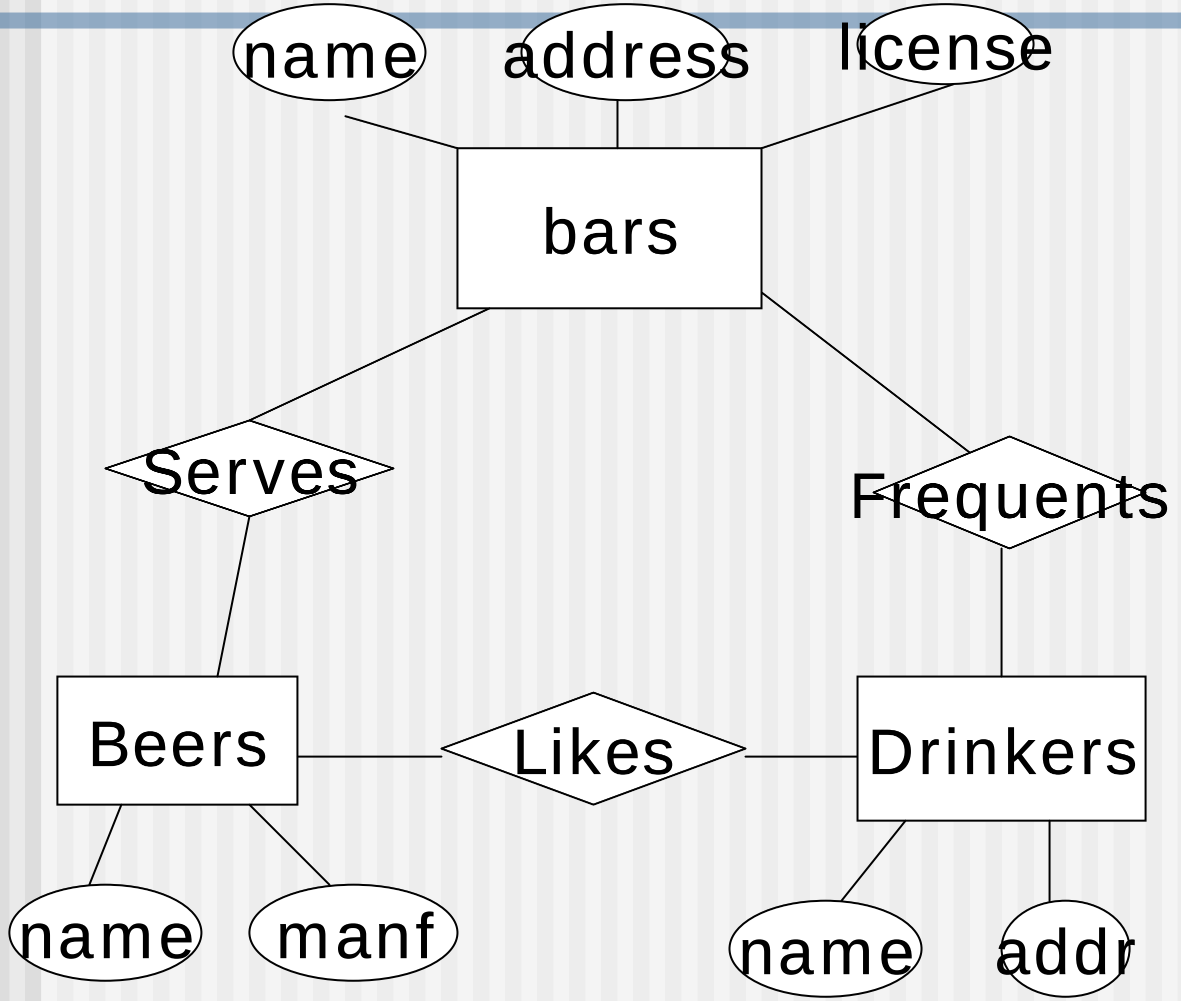
One to one

Multiway Relationships

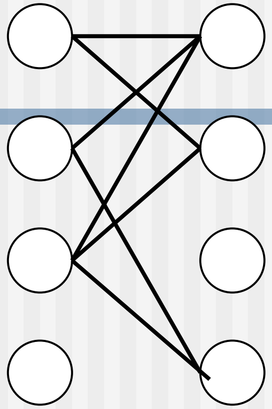




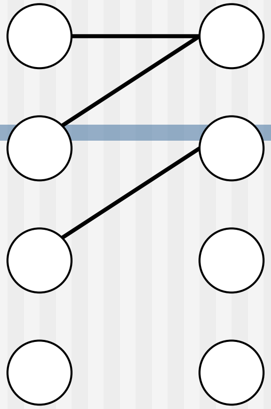
Beers-Bars-Drinkers Example



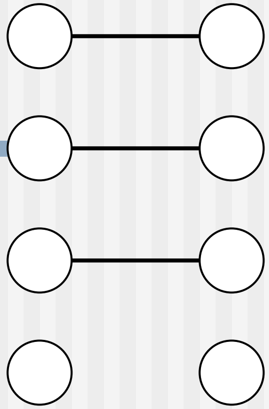
Multiplicity of Relationships



Many-many



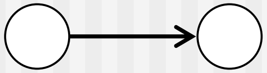
Many-one



One-one

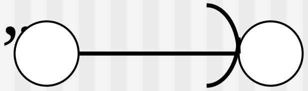
Representation of Many-One

- E/R: arrow pointing to “one.”

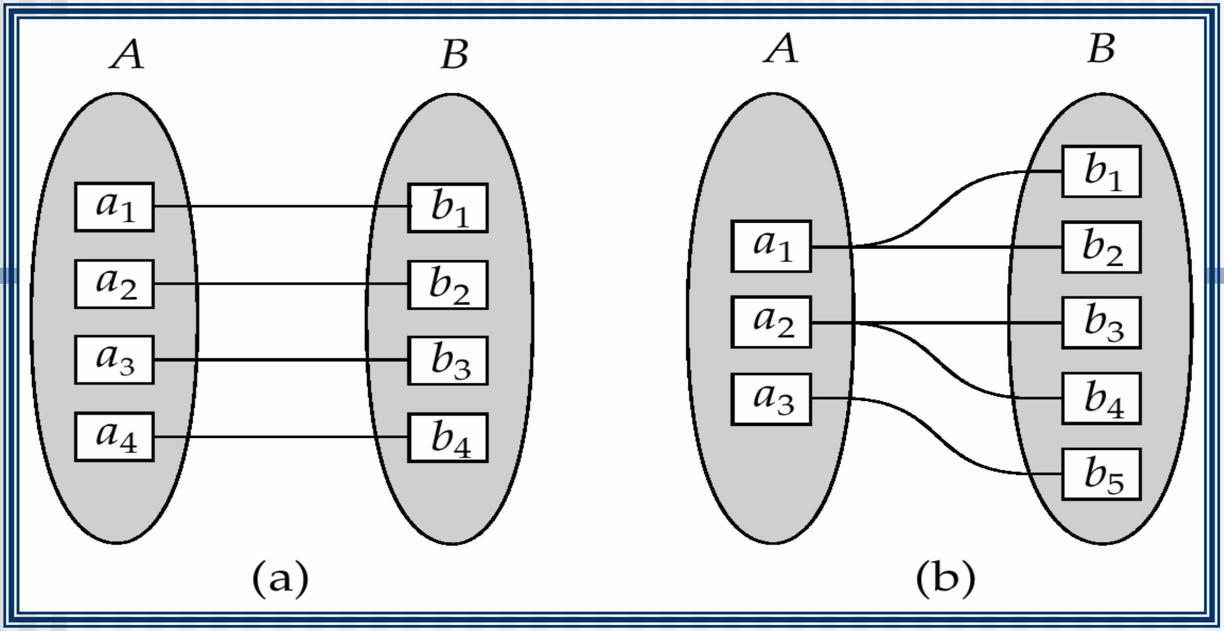


Means at most one.

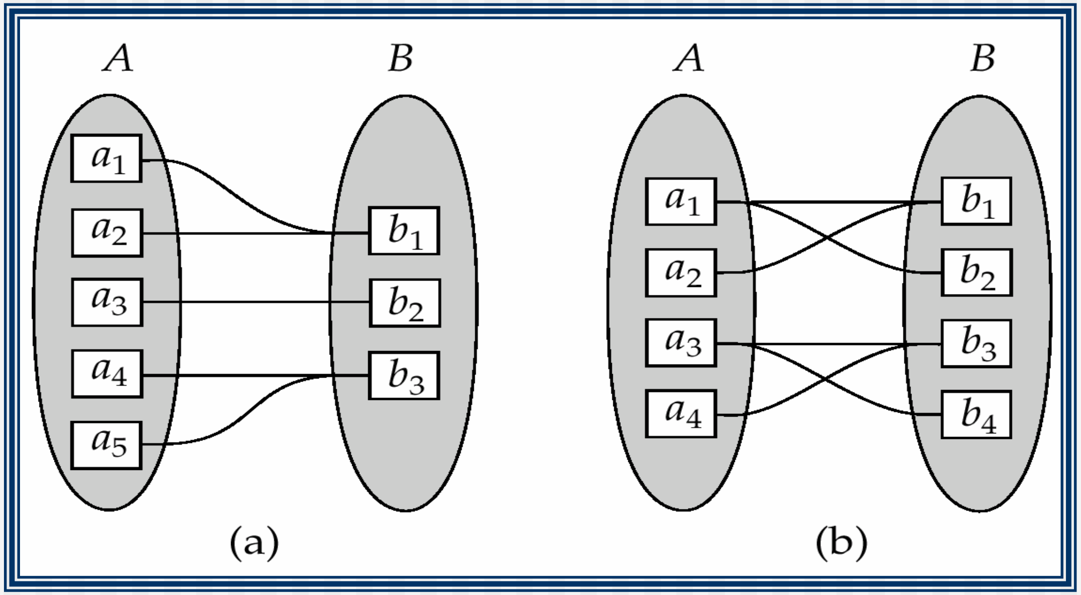
- Rounded arrow = “exactly one.”



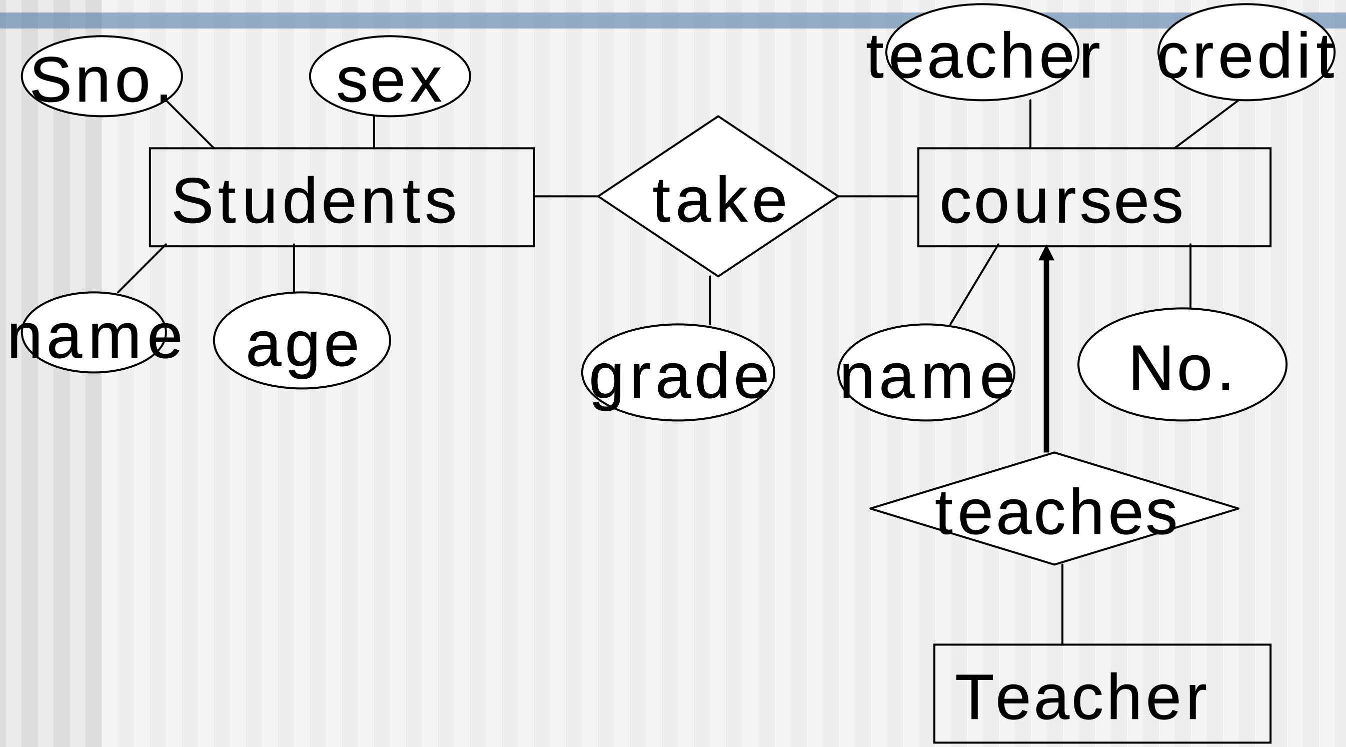
Multiplicity of Relationships



- Express the number of entities to which another entity can be associated via a relationship set.



Examples



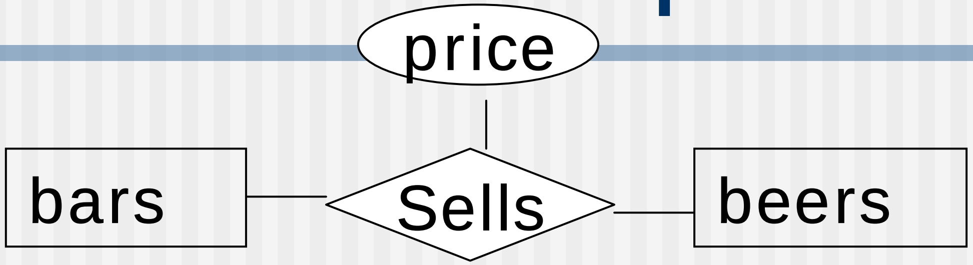
Suppose one teacher teaches only one course, but one course can be taught by many teachers

Binary relation

Many to many relationship

Many to one relationship

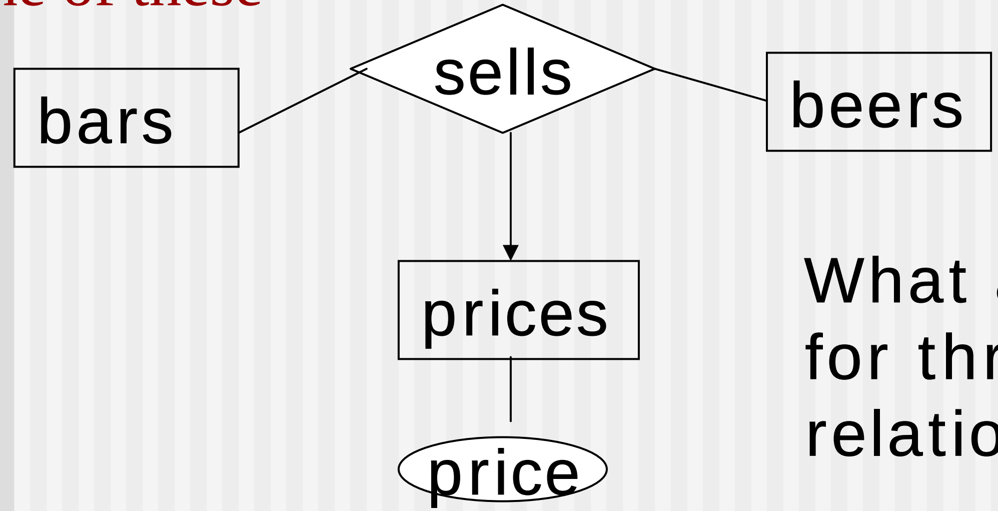
Examples:3-way relationship



Price depends jointly on bar and beer

If price depends only on beers, what should we do?

Notice arrow convention for multiway relationships: “all other E.S. determine one of these

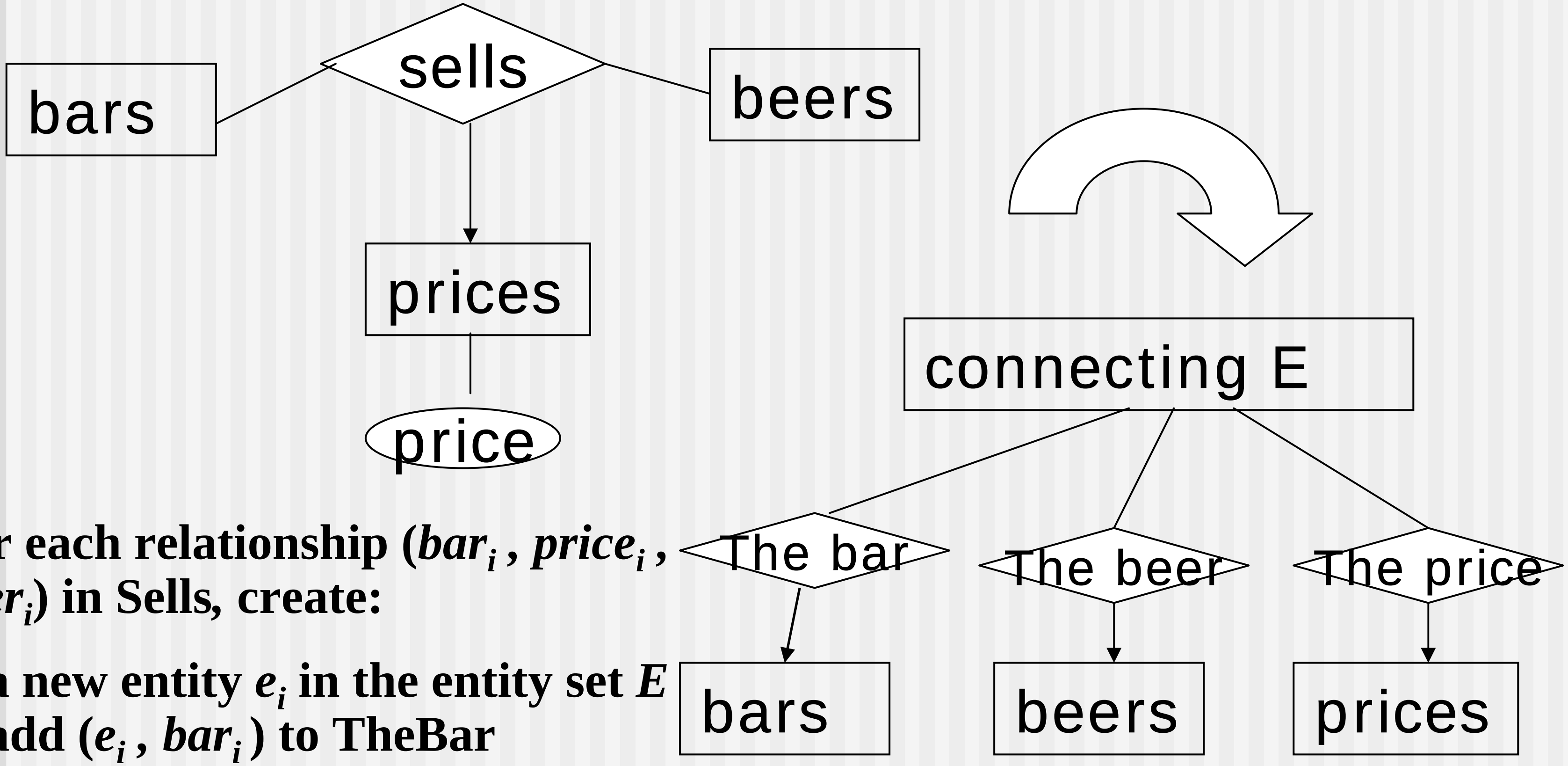


What are the situations for three entities relationship?

Converting Multiway to 2 way

- Creating a new connecting E.S. to represent the rows of a relationship set
- Many-one relationships from the connecting E.S. to the others

Converting Multiway to 2 way: example



For each relationship (*bar_i* , *$price_i$* , *$beer_i$*) in Sells, create:

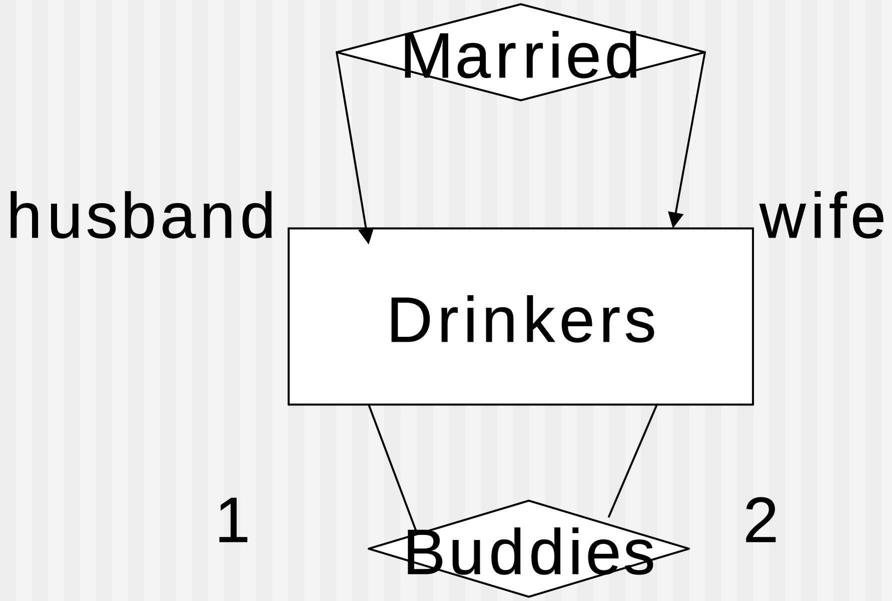
1. a new entity *e_i* in the entity set *E*
2. add (*e_i* , *bar_i*) to TheBar

3. add (*e_i* , *$price_i$*) to *ThePrice*

4. add (*e_i* , *$beer_i$*) to TheBeer

Roles

- Sometimes an E.S.participates more than once in a relationship.
- Label edges with roles to distinguish.



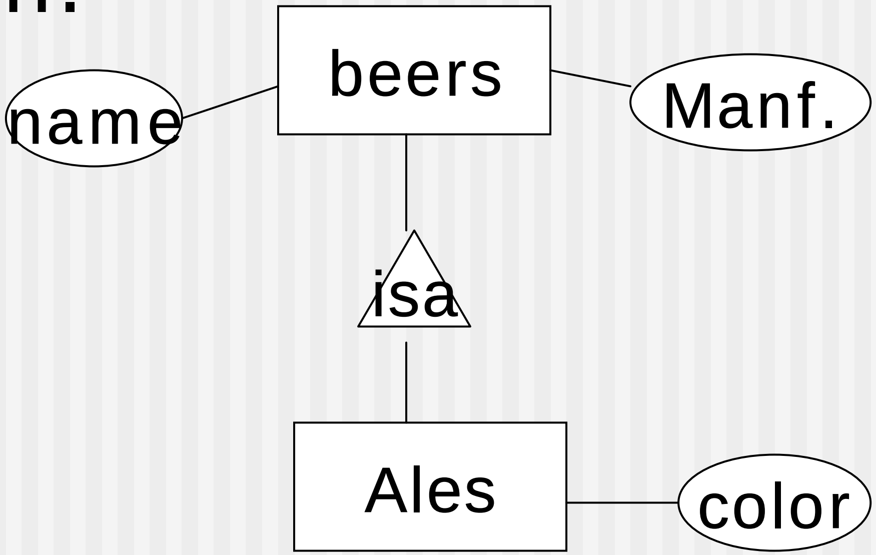
Subclasses

- Subclass = special case = fewer entities = more properties
- Example

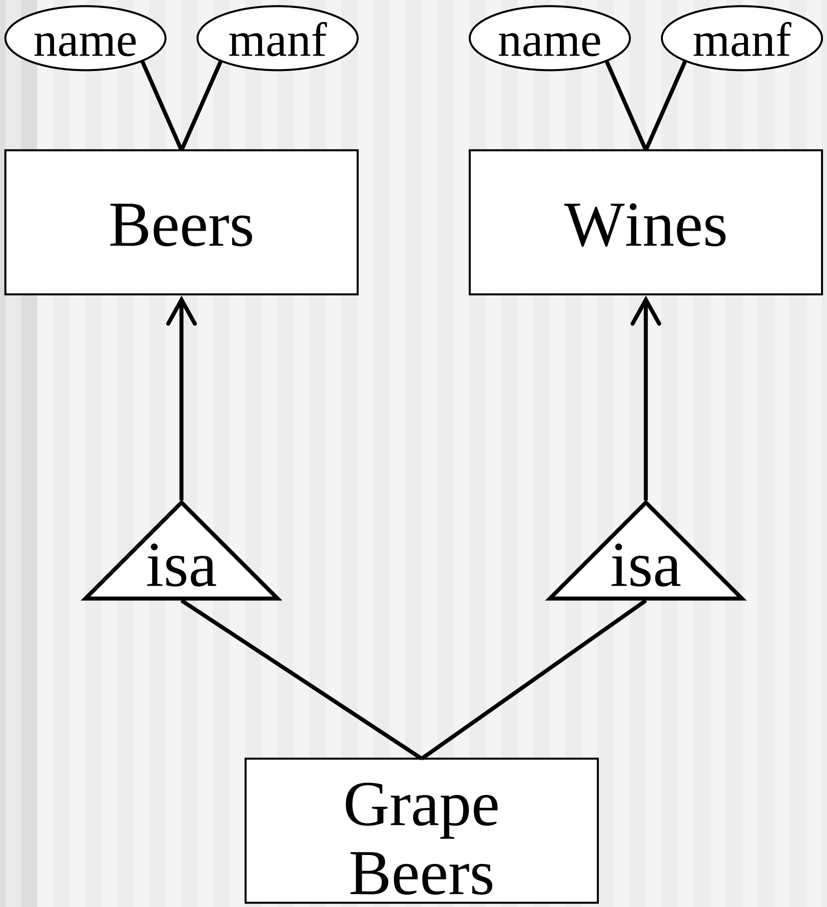
Ales are a kind of beer. In addition to the properties (= attributes and relationships) of beers, there is a “color”attribute for ales.

E/R Subclasses

- Assume subclasses form a tree (no multiple inheritance)
- Isa triangles indicate the subclass relation.



Multiple Inheritance



- Theoretically, an E.S. could be a subclass of several other entity sets.
- Problems?

Example: `manf` means vintner for wines, bottler for beers. What does `manf` mean for "grape beers"?

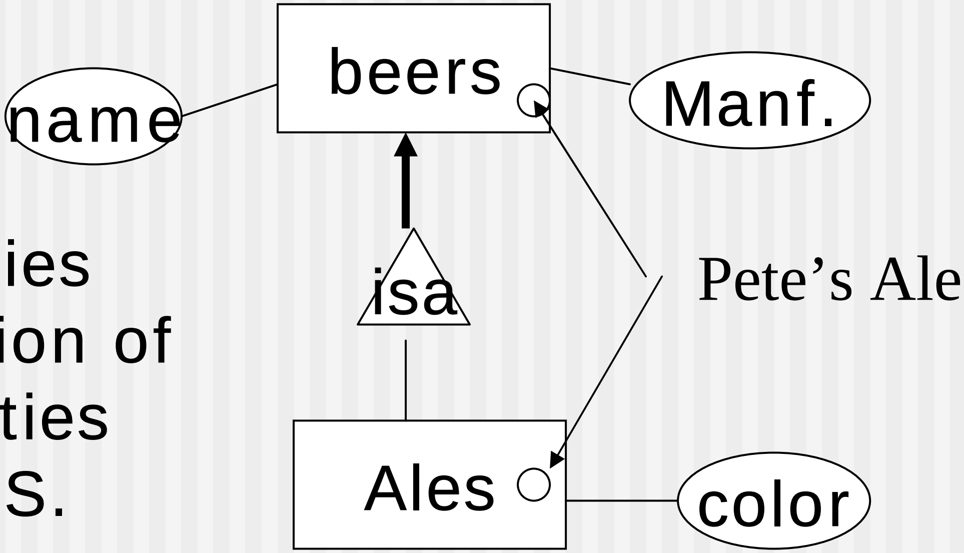
- In practice, we shall assume a tree of entity sets connected by *isa*, with all "isas" pointing from child to parent.

Different Subclass Viewpoints

- E/R viewpoint: An entity has a component in each entity set to which it logically belongs.
  - ✓ Its properties are the union of the properties of these E.S.
- Object-oriented viewpoint: An object (entity) belongs to exactly one class. It inherits properties of its superclasses.
  - ✓ It *inherits* properties of its superclasses.

Example, from E/R Viewpoint

- An entity has a component in each entity set to which it logically belongs.



Its properties are the union of the properties of these E.S.

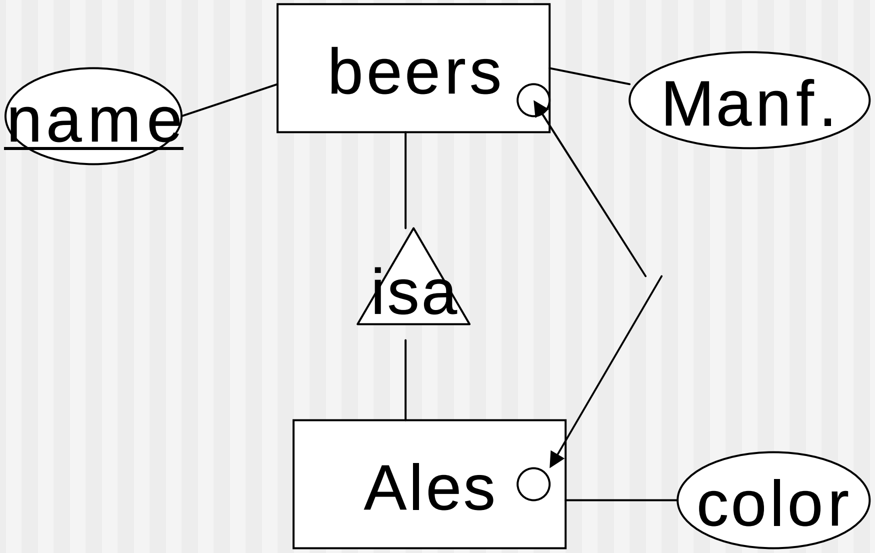
Keys

A key is a set of attributes such that no two entities agree on all these attributes.

- In E/R model, every E.S.must have a key. It could have more than one key, but one set of attributes is the “designated”key.
- In E/R diagrams, you should underline all attributes of the designated key.

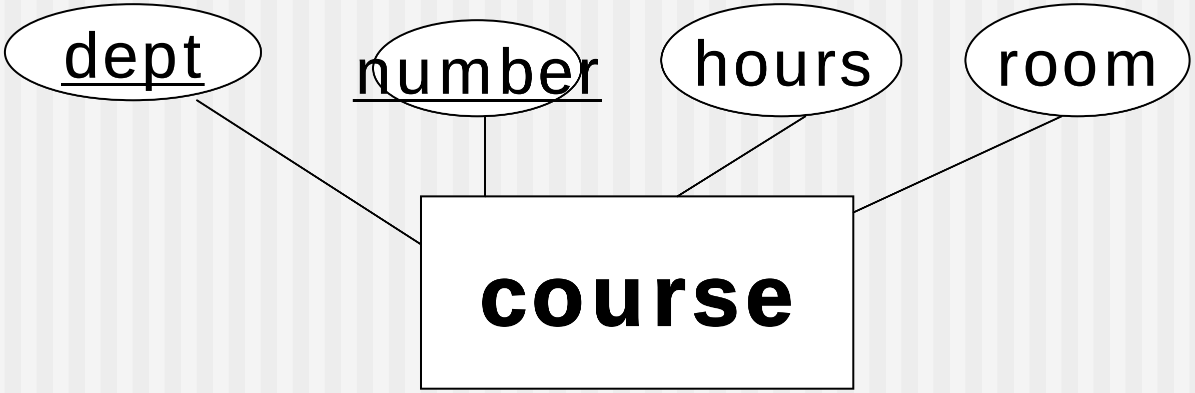
Example

Suppose name is key for Beers



Beer name is also key for Ales. In general, key at root is key for all.

Example: A Multiattribute Key



Possibly, hours+room also forms a key, but we have not designed it as such.

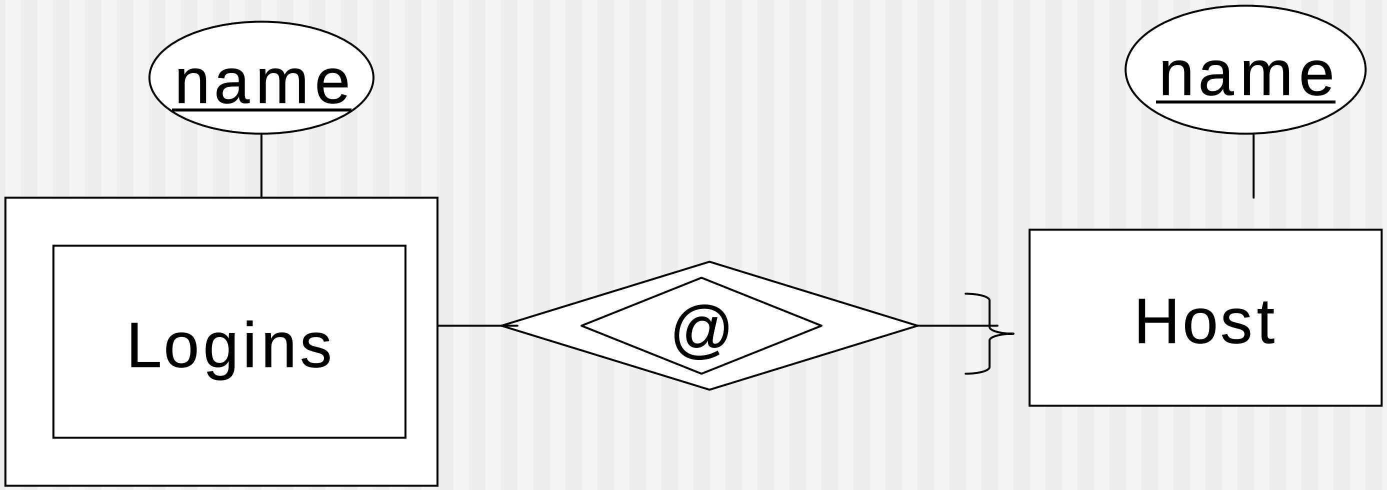
Weak Entity Sets

Sometimes an E.S. E's key comes not (completely) from its own attributes, but from the keys of one or more E.S's to which E is linked by a supporting many-one relationship.

- Called a **weak E.S**.
- Represented by putting double rectangle around E and a double diamond around each supporting relationship.
- Many-one-ness of supporting relationship (includes 1-1) essential. With many-many, we would not know which entity provided the key value.
- "Exactly one" also essential, or else we might not be able to extract key attributes by following the supporting relationship.

Example: Logins (Email addresses)

Login name= user name + host name, e.g. li-fang@cs.sjtu.edu.cn

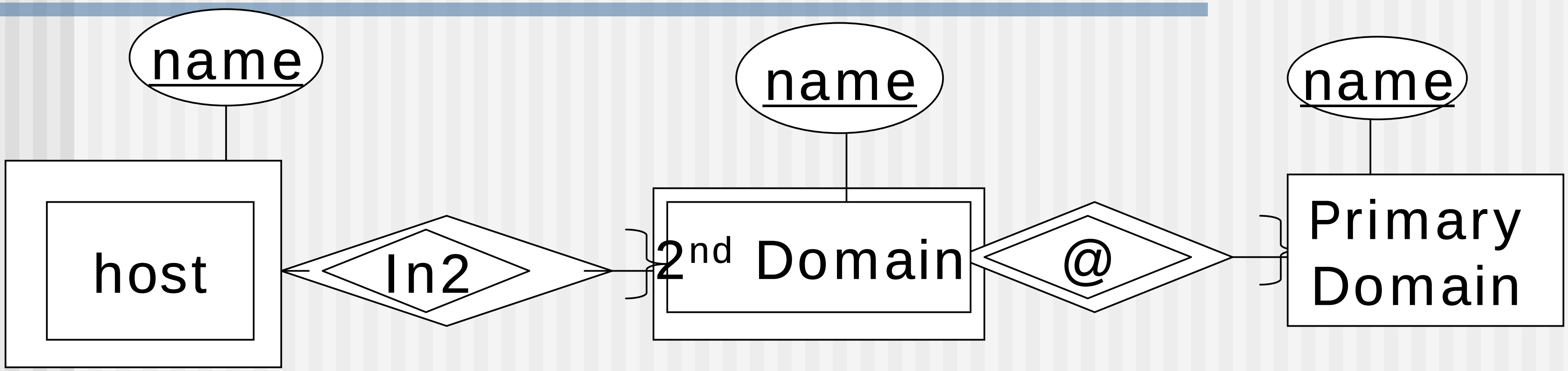


Key for a login =the user name +Host name

Example: Logins (Email addresses)

- A “Login” entity corresponds to a user name on a particular host, but the password table does not record the host, just the user name, e.g. li-fang
- Key for a login = the user name at the host (which is unique for that host only)+IP address of the host (which is unique globally).

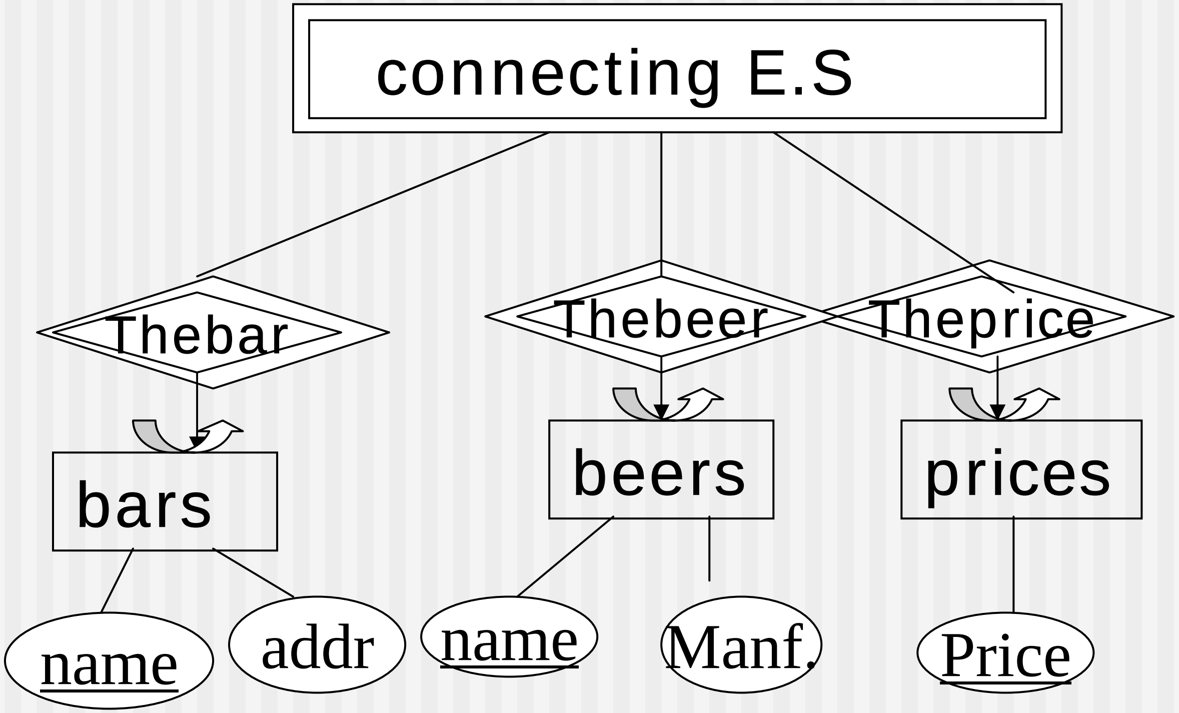
Example:Chain of “Weakness”: consider IP addresses consisting of a primary domain, subdomain and host



- Key for primary domain = its name (e.g.,sjtu)
- Key for secondary domain = its name (e.g., cs) +name of primary domain
- Key for host = its name (e.g., Elis) + name of 2nd Domain+name of primary domain

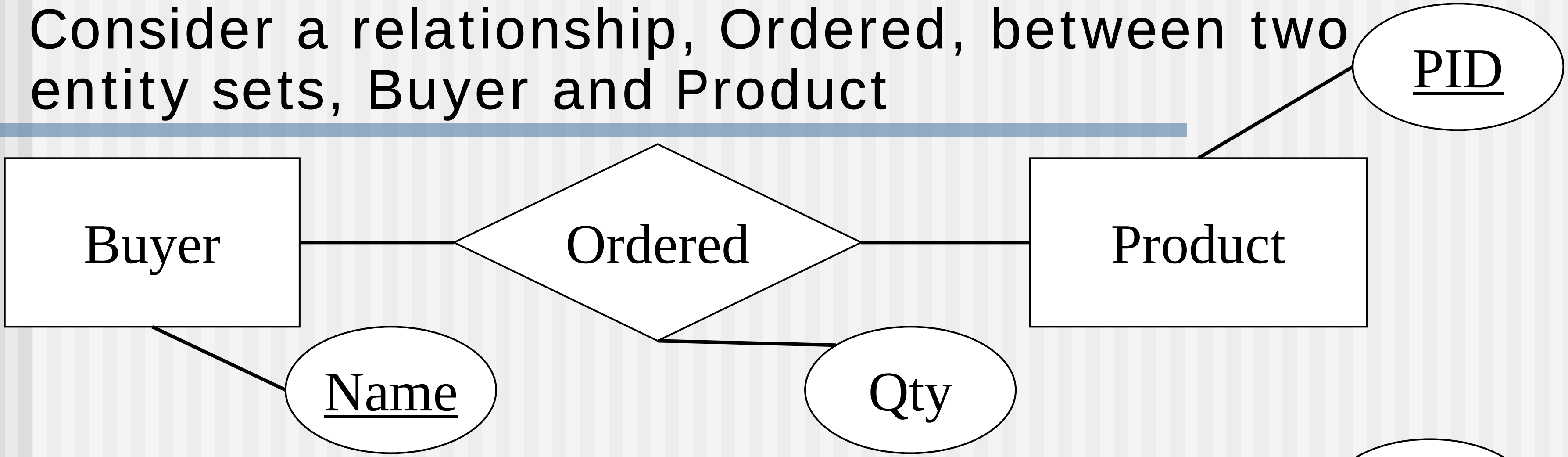
All "Connecting" Entity Sets are Weak

All "Connecting" Entity Sets Are Weak

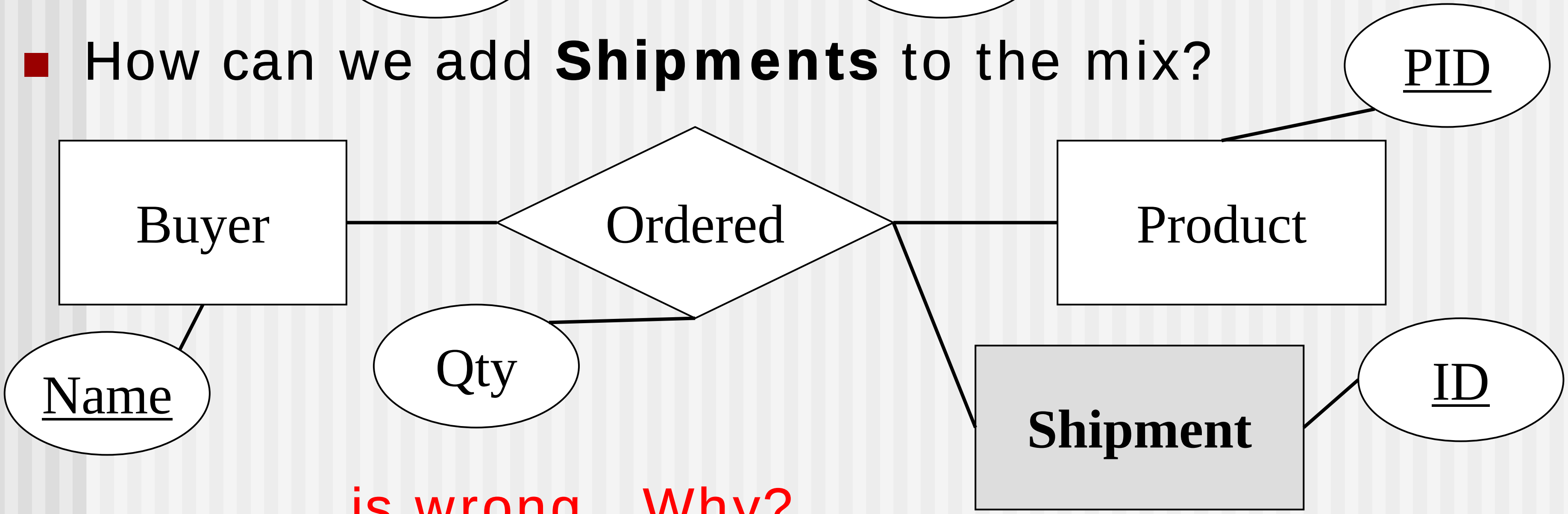


Relationship To Weak Entities

- Consider a relationship, Ordered, between two entity sets, Buyer and Product



- How can we add **Shipments** to the mix?

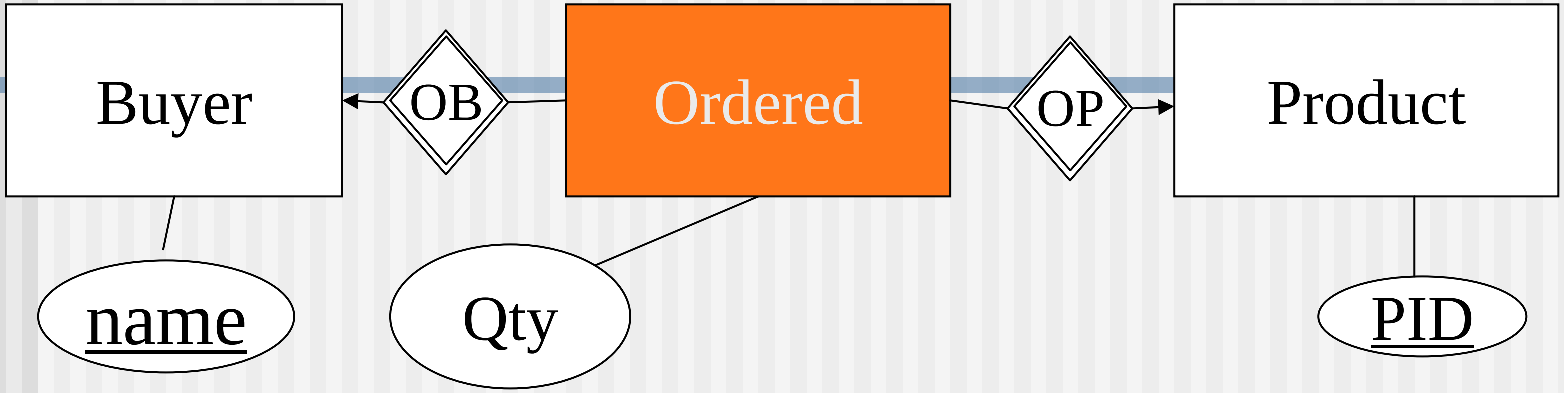


is wrong. Why?

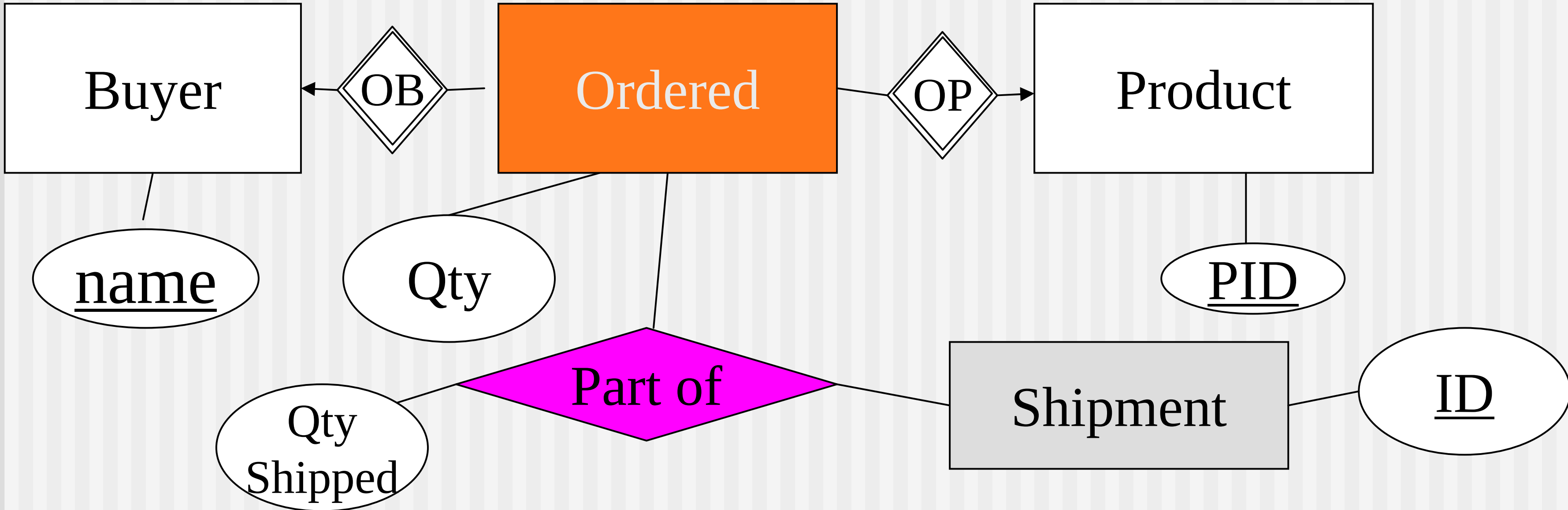
Classroom Discussion

- What is the solution to the problem?

- Solution: make Ordered into a weak entity set.



- And then add Shipment.



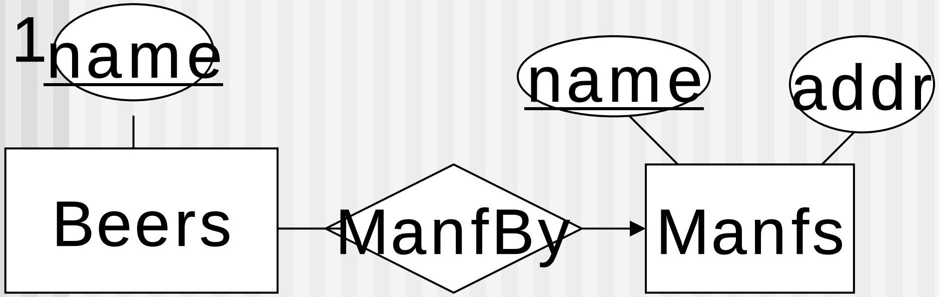
Design Principles

Setting: client has possible vague idea of what he/she wants. You must design a database that represents these thoughts and only these thoughts.

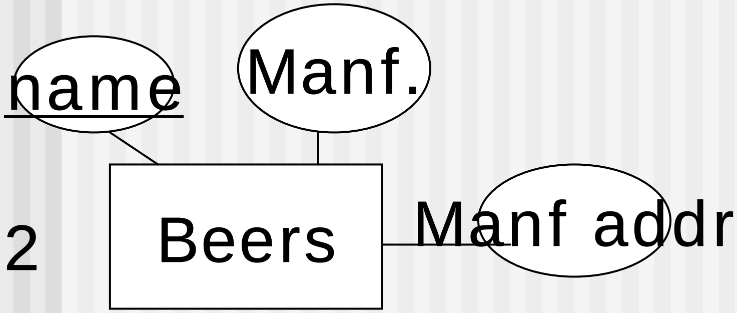
- Avoid Redundancy=saying the same thing more than once, that will waste space and encourage inconsistency.

Example: Which one is better?

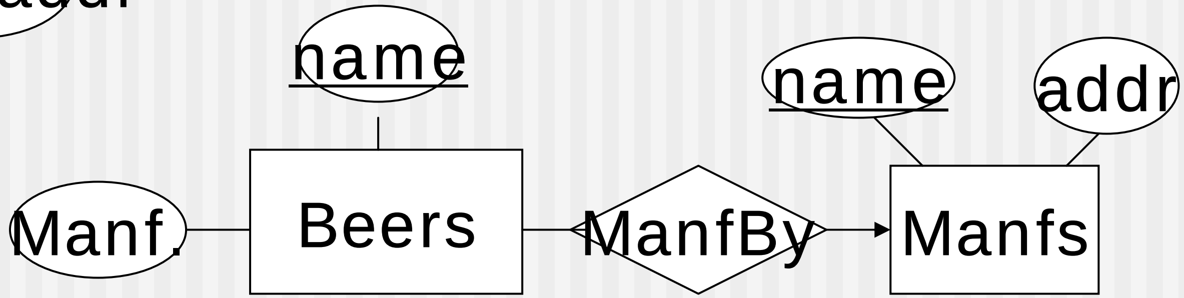
1



2



3



1:good

2:repeats manufacturer address for each beer they manufacture.

3:manufacturer’s name said twice.

Use Schema to Enforce Constraints

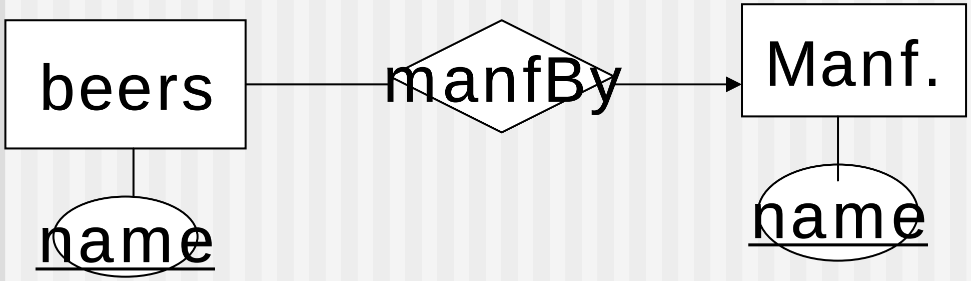
The design schema should enforce as many constraints as possible.

Example

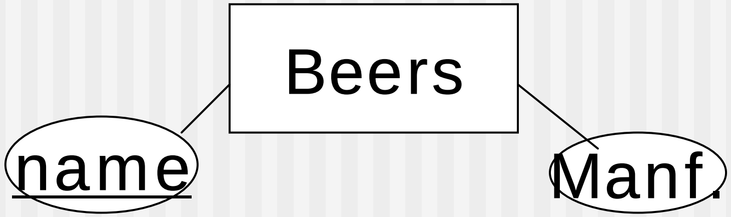
If registrar wants to associate only one instructor with a course, do not allow sets of instructors and count on departments to enter only one instructor per course.

Entity Sets Vs. Attributes

You may be unsure which concepts are worthy of being entity sets, and which are handled more simple as attributes.



Wrong ??



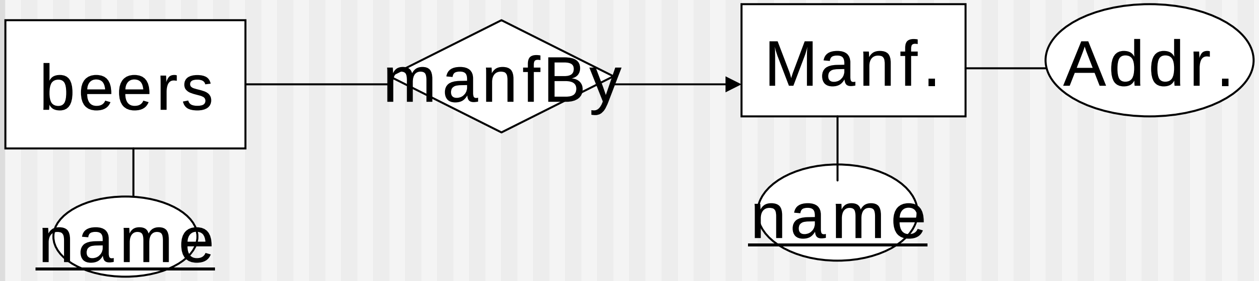
Right !!

Intuitive Rule for E.S.Vs. Attribute

Make an entity set only if it either:

1. Is more than a name of something; i.e., it has nonkey attributes or relationships with a number of different entity sets, or
2. Is the "many" in a many-one relationship.

Example



- Manf. Deserves to be an E.S. because we record addr, nonkey attribute.
- Beers deserves to be an E.S. because it is at the end of the “many” end.

Don’t overuse Weak E.S.

- There is a tendency to feel that no E.S. has its entities uniquely determined without following some relationships.
- However, in practice, we almost always create unique ID’s to compensate: social-security numbers, VIN’s, etc.

Don’t overuse Weak E.S.

- The only times weak E.S.’s seem necessary are when:

1. We can not easily create such ID’s; e.g., no one is going to accept a “species ID”as part of the standard nomenclature (species is a weak E.S supported by membership in a genus)
2. There is no global authority to create them, e.g., crews and studios.

The Modeling of Constraints

- Key constraints
- Single-value constraints
- Referential integrity constraints
- Domain constraints
- General constraints

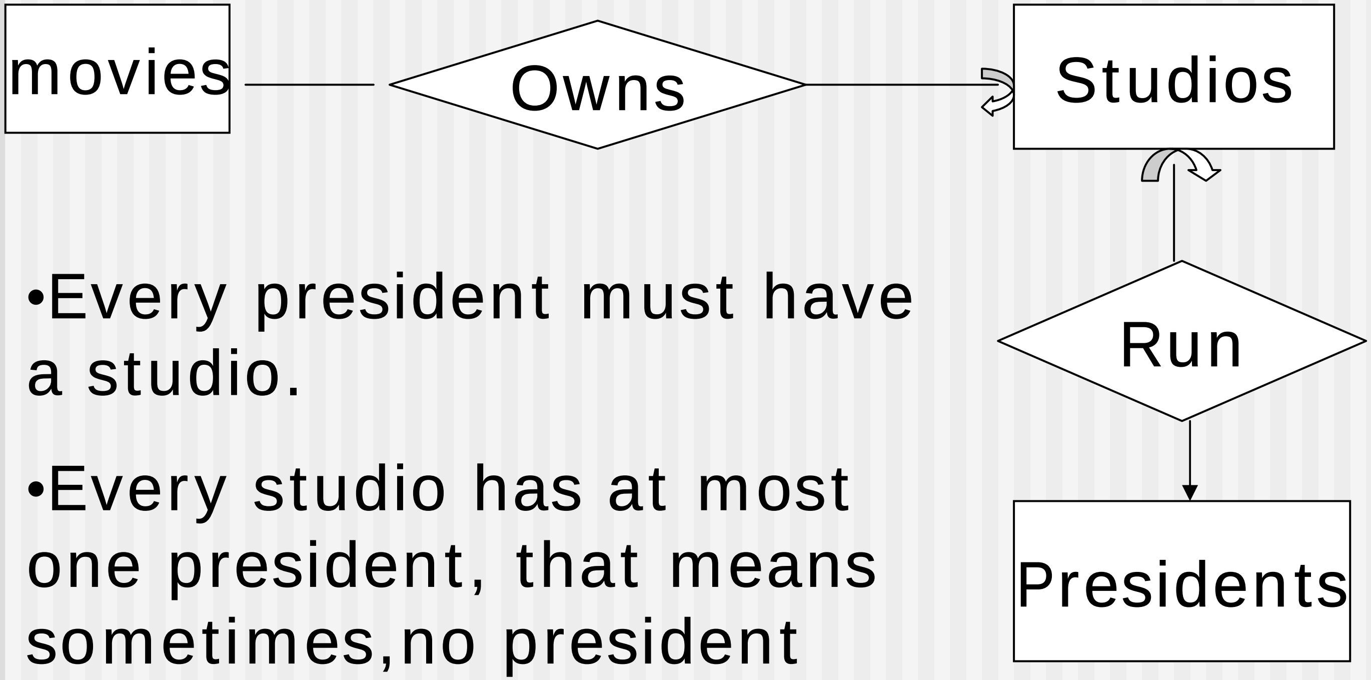
Key constraints

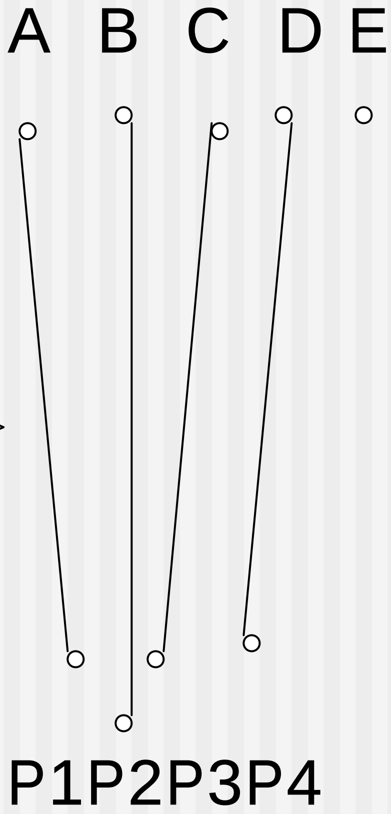
- No two entities may agree in their values for all of the attributes that constitute a key.
- A key may consist of more than one attribute.
- There can also be more than one possible key for an entity set.

Single-value constraints

- Many ways to express:
  - Each attribute of an entity set has a single value. (not null)
  - A relationship R that is many-one from entity set E to entity set F implies a single-value constraint. (at most one)

Referential integrity constraints



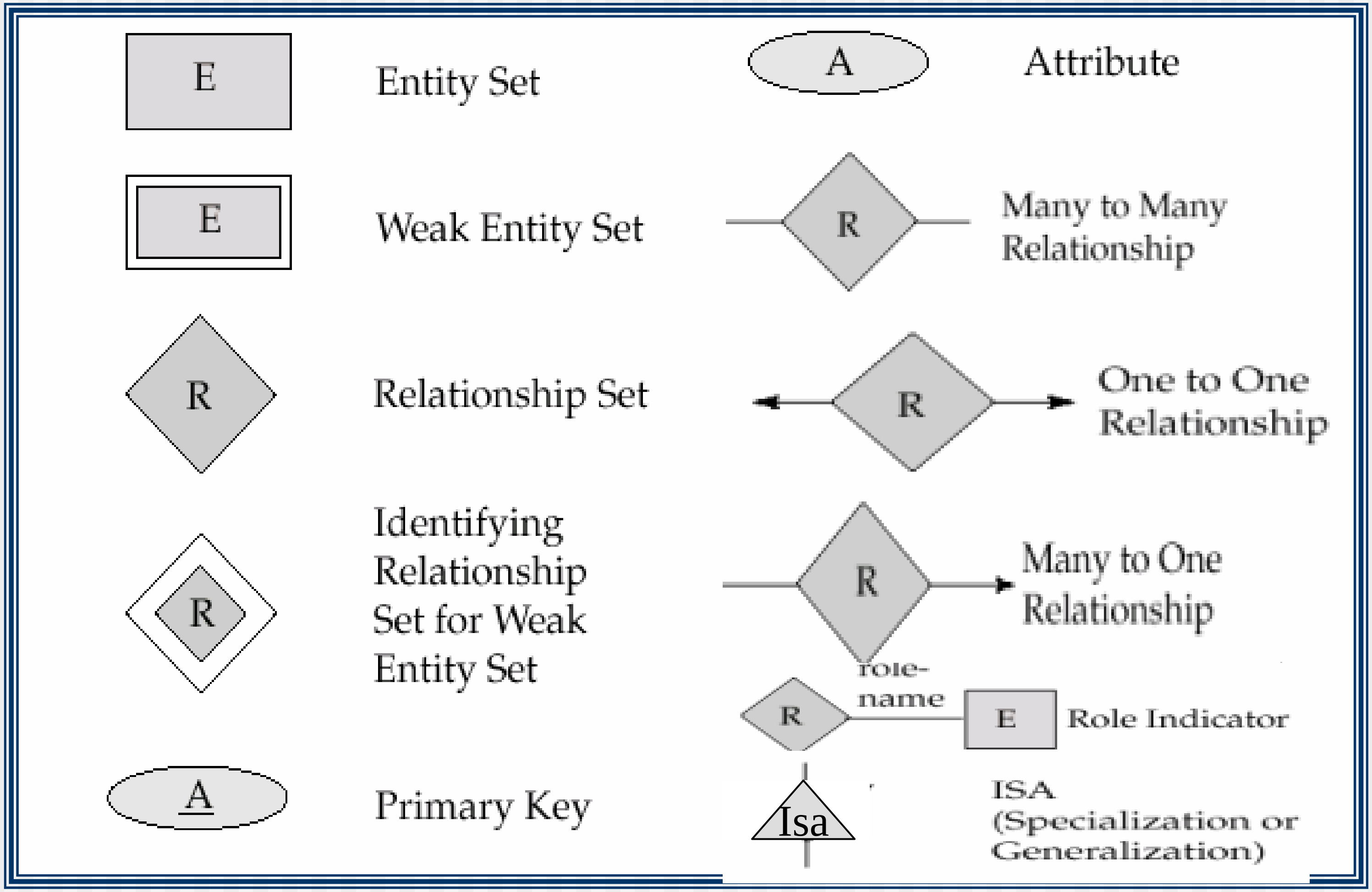


- Every president must have a studio.
- Every studio has at most one president, that means sometimes,no president
- If a studio(e.g.A) does not exist, its president(P1) should also be deleted as well.

Other constraints

- Domain constraints restrict the value of an attribute to be in a limited set.
- General constrains, such as placing a constraint on the degree of a relationship, number constraints and so on.

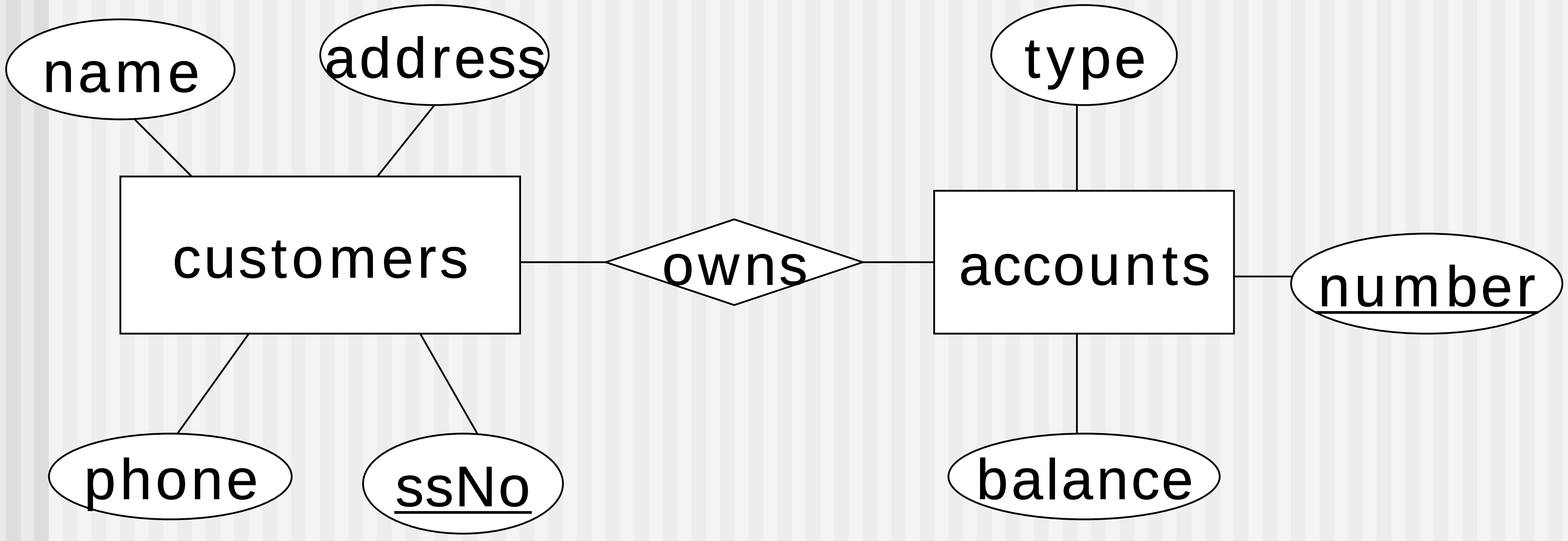
Summary of Symbols Used in E-R Notation



Exercises for E-R diagram

- Let us design a database for a bank, including information about customers and their accounts. Information about a custom includes their name, address, phone, and Social Security number. Accounts have numbers, types(e.g., savings,checking) and balances. We also need to record the customer who own an account. Be sure to include arrows where appropriate, to indicate the multiplicity of a relationship.

E-R Diagram

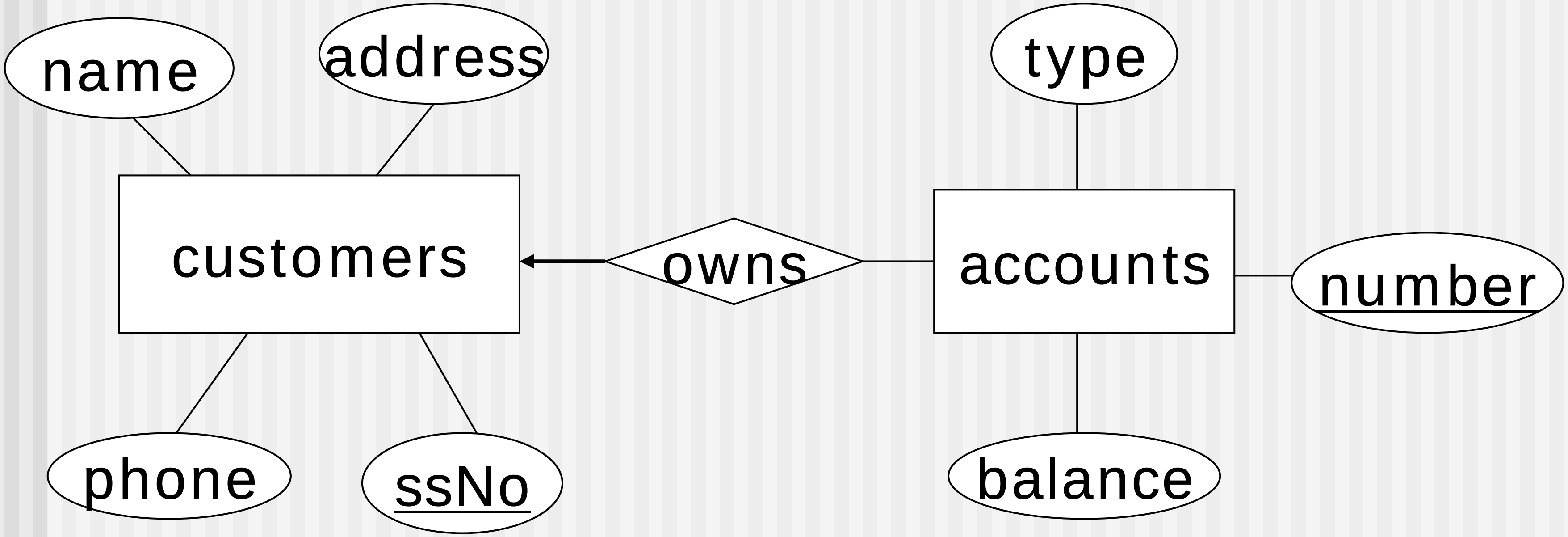

name
address
type
customers
owns
accounts
number
phone
ssNo
balance

Some modifications on the example

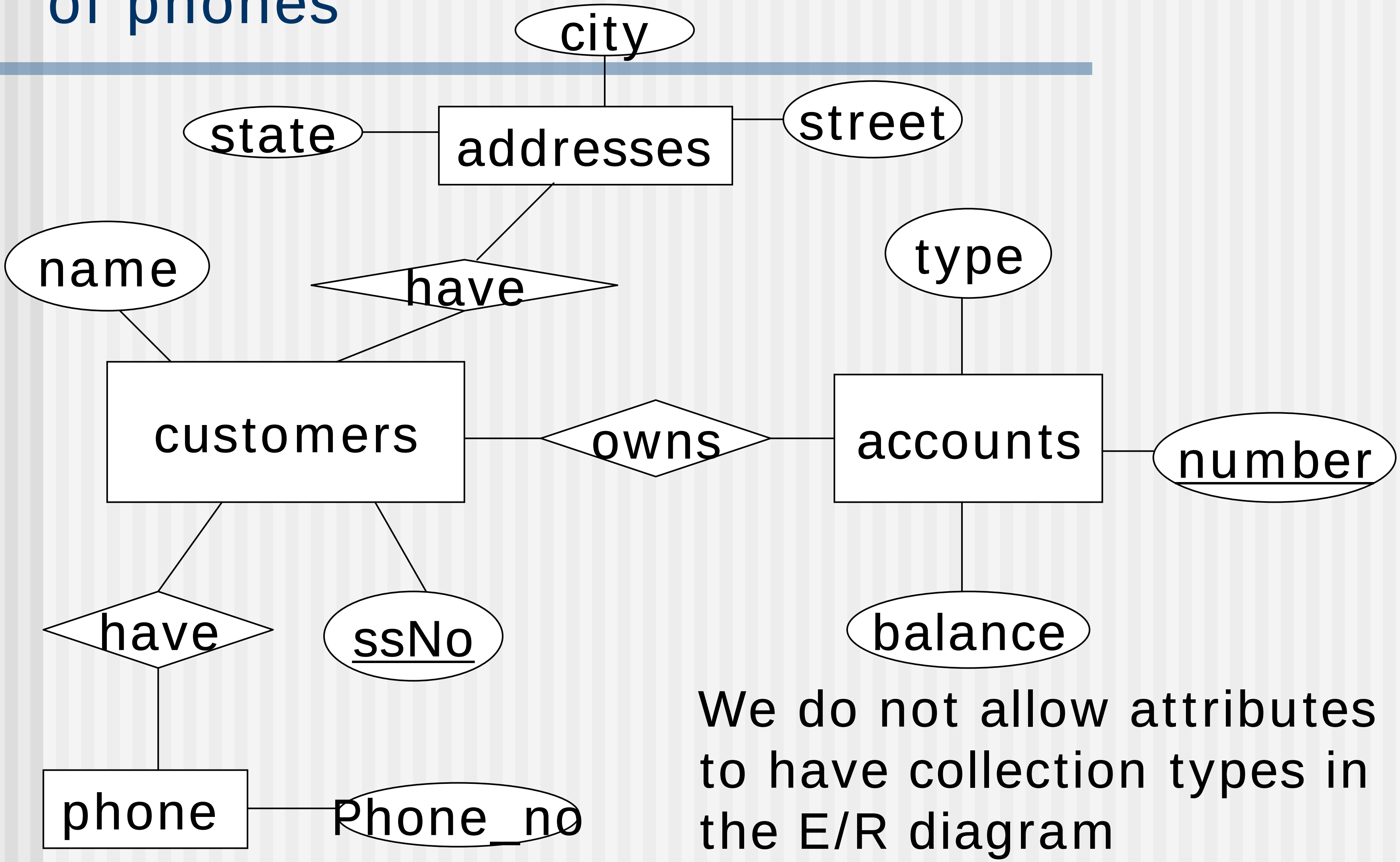
- Suppose an account can have only one customer.
- A customer can have a set of addresses(which is street-city-state triples) and a set of phones.

What are the ER diagram?

Suppose an account can have only one customer.



A customer can have a set of addresses (which is street-city-state triples) and a set of phones



We do not allow attributes to have collection types in the E/R diagram

Question?

- Modify the diagram so that customers can have a set of addresses, and at each address there is a set of phones.

Classroom Design Exercise

- Design a database suitable for a university registrar. This database should include information about students, departments, professors, courses, which students are enrolled in which courses, which professors are teaching which courses, students grades, TA’s for a course (TA’s are students), which courses a department offers, and any other information you deem appropriate.

Homework

- Exercise 2.2.1
- Exercise 2.4.1
- Exercise 2.4.2

Summary of Chapter 2

- Entity-Relationship Diagrams

Entities & Attributes & Relationships

Multiplicity, Multiway of Relationships

Weak Entity Sets

Subclasses

- Good Design

Faithfully represent

Avoid redundancy

Choose appropriate elements