CS 317 & CS 387 Database Information Systems

Lecture 02 – ER Models

Why use models?

- We build models of complex systems because we cannot comprehend any such system in its entirety – has many aspects to it.
- Need to develop a common understanding of the problem and the solution – communicate easily and efficiently.
- Cannot afford a trial-and-error approach
  - Allows us to study without building a real system

Which aspects to model?

- The choice of which model we use has a profound influence on how a problem is attacked and how a solution is shaped
- No single model is sufficient; every complex system is best approached through a set of independent models
- The quality of a model is defined by it's closeness to reality.

DATA MODEL

- Represents operational data about real world events & entities
- Also known as a domain model.
- Model may be at various levels:
  - logical or physical
  - external, conceptual, internal

Data Model……

- A good model
  - is easy to understand
  - has few concepts
  - Is consistent
  - Enables operations on the model to execute efficiently.
- Must capture meaning of data (data semantics) which help us in interpreting the data

Data Model……

- Semantics captured through data types, inter-relationships and data integrity constraints
  - permitted values
  - uniqueness
  - existence dependence
  - restrictions on some operations such as insertions, deletions

Why learn about Modeling?

- The way in which data is stored is very important for subsequent access and manipulation by SQL.
- Properties of a good data model:
  - It is easy to write correct and easy to understand queries.
  - Minor changes in the problem domain do not change the schema.
  - Major changes in the problem domain can be handled without too much difficulty.
  - Can support efficient database access.

The next few weeks

- Relational Data Models
  - The Entity-Relationship (ER) model
  - The relational model
  - Converting E/R diagram to relational designs.
  - Functional and multi-valued dependencies
- At this point, you will know how to
  - Identify all entities and relationships and describe them using an E/R diagram .
  - Convert the E/R model to a number of relations in a relational schema.
  - Use normalization to eliminate redundancy and bad choices in the relational schema.

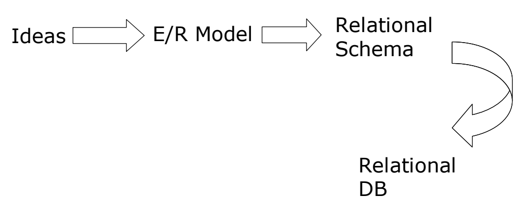
Basic DB Terminology

- **Data model :** describes high-level conceptual structuring of data
  - Example: Data is set of student records, each with ID, name, address, and courses
  - Example: Data is a graph where nodes represent proteins and edges represent chemical bonds between proteins
- **Schema** describes how data is to be structured and stored in a database
  - Defined during creation of the database
  - Schemas rarely change

Terminology

- **<u>Data</u>** is actual “instance” of database
  - Updated continuously
  - Changes rapidly

The modeling sequence



Purpose of E/R Model

- The *<u>E/R model</u>* allows us to sketch database designs.
  - Kinds of data and how they connect.
  - Not how data changes.
- Designs are pictures called *entity-relationship diagrams*.
- Later: convert E/R designs to relational DB designs.

Concepts

- *Entity* = “thing” or object.
- *Entity set* = collection of similar entities.
  - Similar to a class in object-oriented languages.
- *Attribute* = property of (the entities of) an entity set.
  - Attributes are simple values, e.g. integers or character strings.

ENTITY

- an object that exists
- distinguishable from other objects
- could be concrete or abstract
- Examples : this course on DBIS, Ganesh as a student, etc

ENTITY SET

- a set of similar entities
- need not be disjoint with other entity sets
  - e.g., supplier and consumer may have common entities
- example : set of all books in a library
- entity set also called entity type or entity class
- an entity is an occurrence or an instance of some entity type

ENTITY SET……

- we often use the words ‘entity’ to mean ‘entity-set’
- entity sets are named using singular common nouns :
  - Book
  - Student
  - Course

Analogy to the OO Model

- Entity Set ≅ Class in the OO Model
- Entity ≅ Object

Instance of an Entity Set

Name	*Id*	*Address*
Hermione Grainger	HG	Gryffindor Tower
Draco Malfoy	DM	Slytherin Tower
Harry Potter	HP	Gryffindor Tower
Ron Weasley	RW	Gryffindor Tower

- For each entity set, the instance stores a specific set of entities.
- Each entity is a tuple containing specific values for each attribute.

ATTRIBUTE

- an entity has a set of attributes
- attribute defines property of an entity
  - entity **Book** has **Price** attribute
- it is given a name
- attribute has value for each entity
- value may change over time
- same set of attributes are defined for ALL entities in an entity set

ATTRIBUTE….

- Example : entity set BOOK has the following attributes

  TITLE ISBN
  ACC-NO AUTHOR
  PUBLISHER YEAR
  PRICE

- a particular book has value for each of the above attributes

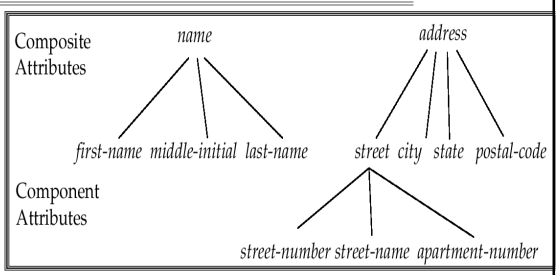
ATTRIBUTE….

- an attribute may be **multi-valued**, i.e., it has more than one value for a given entity;
  - A book may have many authors
  - An account may have multiple owners (joint)
- an attribute which uniquely identifies entities of a set is called **candidate key attribute** of that entity set
  - Eg: ISBN number for a book
  - PAN number for a citizen of India etc.

Composite Attributes

- Composite attribute: As the name implies it is made up of multiple components:
  - Eg: Name = First name + Last Name
  - Address = Street name + Number + City etc.
  -

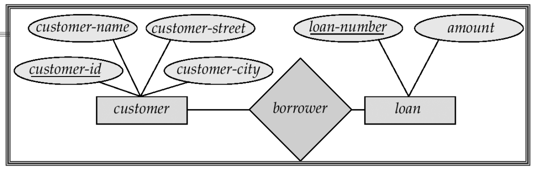
Composite Attributes



Derived Attributes

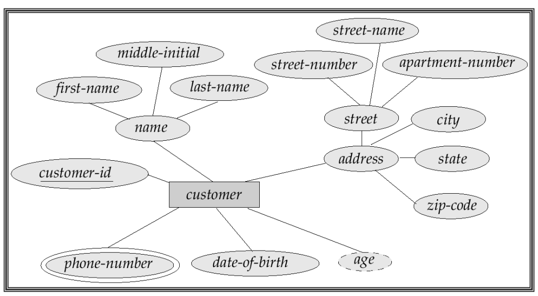
- A derived attribute is based **entirely** on another attribute.
  - For example, an employee's monthly salary is based on the employee's annual salary.
  - Age is derived out of date of birth

E-R Diagrams



- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Ellipses/circles** represent attributes
  - **Double ellipses** represent multi-valued attributes.
  - **Dashed ellipses** denote derived attributes.
- **Underline** indicates primary key attributes (will study later)

E-R Diagram With Composite, Multivalued, and Derived Attributes



PRIMARY KEYS

- To distinguish occurrences of entities
- Distinction made using values of some attribute(s)
- Set of one/more attributes which, taken collectively, uniquely identify an entity in an entity set is called its candidate key
  - Roll-number for a student
  - Acc-no for a book

PRIMARY KEYS…..

- No subset of it is a candidate key
- An entity may have multiple candidate keys
- Primary key is a candidate key chosen by designer as the principal means of identification

EXAMPLE : A COLLEGE

(some entities and their attributes)

- STUDENT : rollno, name, hostel-no., date-of-birth
- COURSE : courseno, name, credits
- TEACHER : empno, name, rank, room-no, telephone
- DEPT : name, telephone

Ex : identify primary keys of above entities.

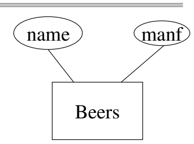
EXAMPLE : A COLLEGE

- STUDENT : <u>rollno</u>, name, hostel-no., date-of-birth
- COURSE : <u>courseno</u>, name, credits
- TEACHER : <u>empno</u>, name, rank, room-no, telephone
- DEPT : <u>name</u>, telephone

EXAMPLE : A COLLEGE…

- Perception of reality and focus of design could have indicated more entities
  - HOSTEL, SEMESTER
  - Or, teacher could only be an attribute
- EXERCISE : Identify entities in a hospital and give a few instances of each

Example

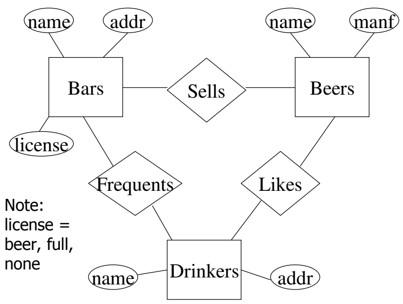


- Entity set <u>Beers</u> has two attributes, <u>*name*</u> and <u>*manf*</u> (manufacturer).
- Each Beers entity has values for these two attributes, e.g. (Bud, Anheuser-Busch)

Relationships

- A <u>**relationship set**</u> connects two or more entity sets.
  - It is represented by a diamond, with lines to each of the entity sets involved.
- A relationship R between entity sets E and F relates specific entities in E to some entities in F.
- R is a tuple (e, f) where e is in E and f is in F.
- An instance of R is simply the “concatenation” of the attributes lists for all pairs of tuples(e, f).

Example

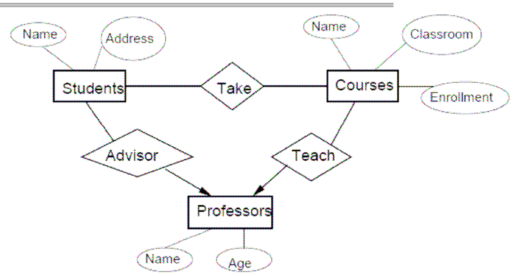


Bars sell some beers.

Drinkers like some beers.

Drinkers frequent some bars.

Another Example



Attributes types: strings, numbers, or "enums"(A Professor 's Age could be "old,""much older," or "still alive!").

Instances Vs Models

- The current *"value" of an entity set* is the set of entities that belong to it.
  - Example: the set of all bars in our database.
  - Each entity is a tuple containing specific values for each attribute.
- The *"value" of a relationship set* is a set of lists of currently related entities, one from each of the related entity sets.

RELATIONSHIP SET

- A *relationship* set is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

  $\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$

  where $(e_1, e_2, \ldots, e_n)$ is a relationship

- But the terms 'relationship' and 'relationship set' often used interchangeably

RELATIONSHIP SET….

- binary relationship set `TAKE` between `STUDENT` and `COURSE`
- relationship `TAKE` could be ternary among `STUDENT`, `COURSE` and `TEACHER`

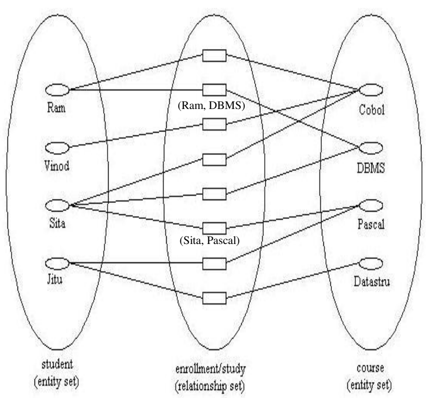
RELATIONSHIP SET….

- Relationships typically named using verbs
  - `Take`
  - `Enroll`
  - `Order`

EXERCISE : identify relationships and their attributes in the hospital example and give a few instances of each

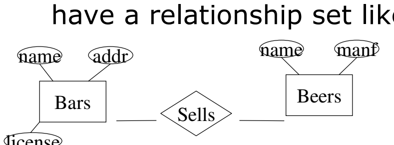
DEPICTING A RELATIONSHIP

- entity sets as a collection
- entity instances by small ovals
- relationship instances by small rectangle with connections to involved entities



Example

- For the relationship **Sells**, we might have a relationship set like:



Bar	Beer
Joe's Bar	Bud
Joe's Bar	Miller
Sue's Bar	Bud
Sue's Bar	Pete's Ale
Sue's Bar	Bud Lite

PRIMARY KEY FOR REPATIONSHIPS

- Made of primary keys of <u>all participating entities</u>
  - e.g., primary key of **<u>TAKE</u>** is
    (rollno, courseno)
  - How about for `Sells`?

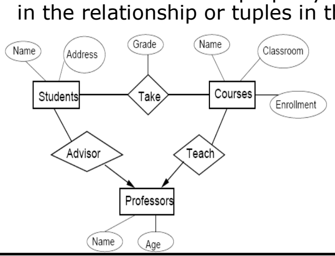
Relationship instance (Take)

Student	*Address*	*Course*	*Enrollment*	*Grade*
Hermione Grainger	Gryffindor Tower	Potions	∞	A-
Draco Malfoy	Slytherin Tower	Potions	∞	B
Harry Potter	Gryffindor Tower	Potions	∞	A
Ron Weasley	Gryffindor Tower	Potions	∞	C

What entity would *Grade* be an attribute of?

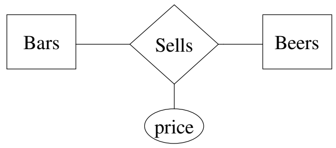
Attributes of a Relationship

- It is useful/essential to attach attributes to relationships.
- Such an attribute is a property of the entity-pairs in the relationship or tuples in the relationship set.



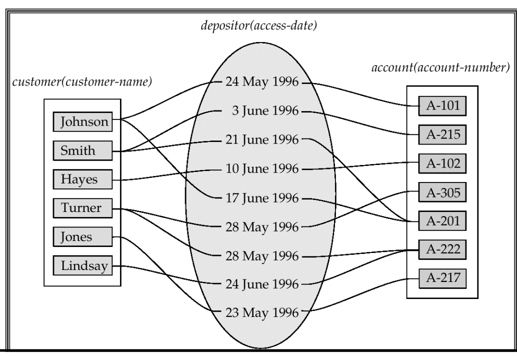
Grade is a function
Of *Student* & *Course*

Another Example



Price is a function of both the bar and the beer, not of one alone.

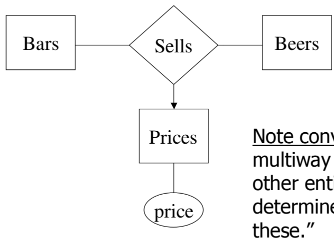
Banking example



Equivalent Diagrams Without Attributes on Relationships

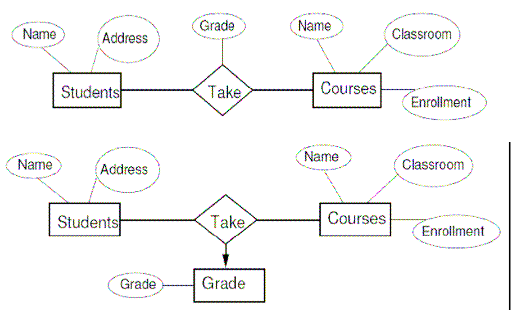
- Create an entity set representing values of the attribute.
- Make that entity set participate in the relationship.

Example



Note convention: arrow from multiway relationship = "all other entity sets together determine a unique one of these."

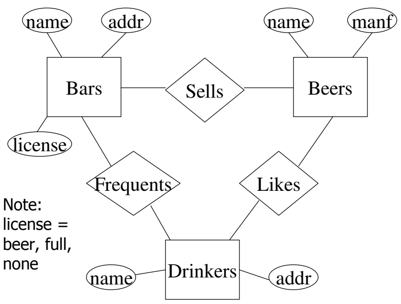
Academics Example



Multiway Relationships

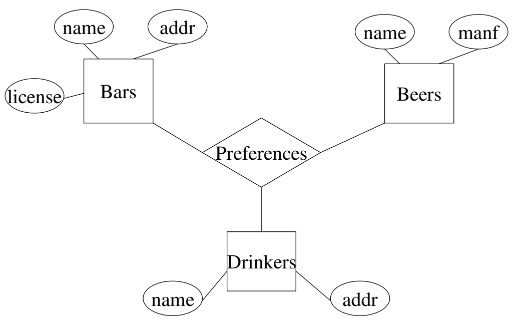
- Sometimes, we need a relationship that connects more than two entity sets.
- Suppose that drinkers will only drink certain beers at certain bars.
  - Our three binary relationships Likes, Sells, and Frequents do not allow us to make this distinction.
  - But a 3-way relationship would.

Originally we had



Note:
license =
beer, full,
none

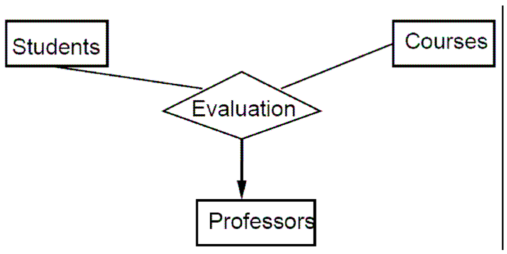
Using Ternary Relationships



The Relationship Set

Bar	Drinker	Beer
Joe’s Bar	Ann	Miller
Sue’s Bar	Ann	Bud
Sue’s Bar	Ann	Pete’s Ale
Joe’s Bar	Bob	Bud
Joe’s Bar	Bob	Miller
Joe’s Bar	Cal	Miller
Sue’s Bar	Cal	Bud Lite

Academics Example



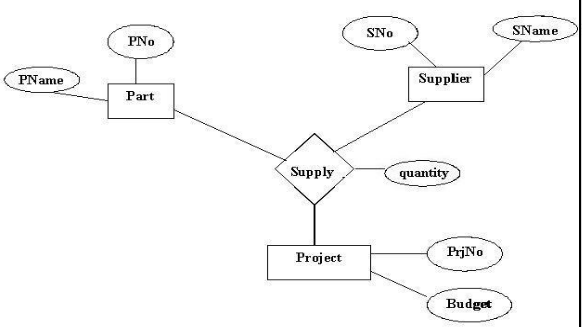
Scenario: >= 1 professor can teach a course but each student evaluate each professor.

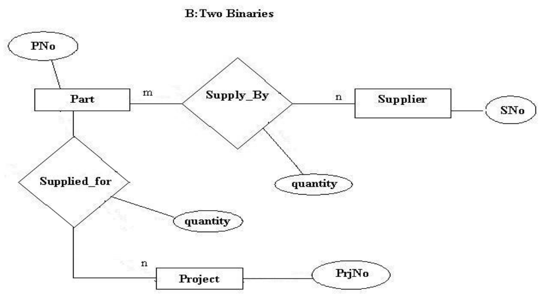
TERNARY RELATIONSHIPS

- Be sure that your model reflects real-world correctly
- Ternary (or, of higher order) relationships are harder to understand
- A ternary relationship **is not the same** as two binary relationships

Exercise : Compare the following E-R Diagram with the one on next page using sample data

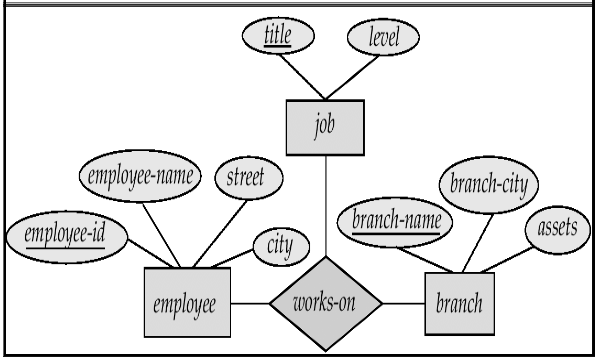
A: Ternary





Can you point out "loss of data" in going from A to B above?
Will adding a relationship between supplier and project help?

E-R Diagram with a Ternary Relationship



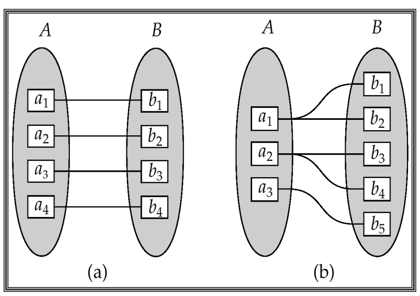
Cardinalities in Relationships

- Mapping cardinality of a relationship
  - 1 –1
  - 1 – many
  - many – 1
  - Many-many

One-One Relationships

- In a *one-one* relationship, each entity of either entity set is related to at most one entity of the other set.
- Example: Relationship Best-seller between entity sets Manfs (manufacturer) and Beers.
  - A beer cannot be made by more than one manufacturer, and no manufacturer can have more than one best-seller (assume no ties).

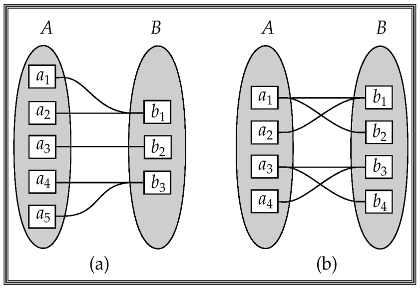
One-One and One-Many



Many-One Relationships

- Some binary relationships are *many - one* from one entity set to another.
- Each entity of the first set is connected to at most one entity of the second set.
- But an entity of the second set can be connected to zero, one, or many entities of the first set.

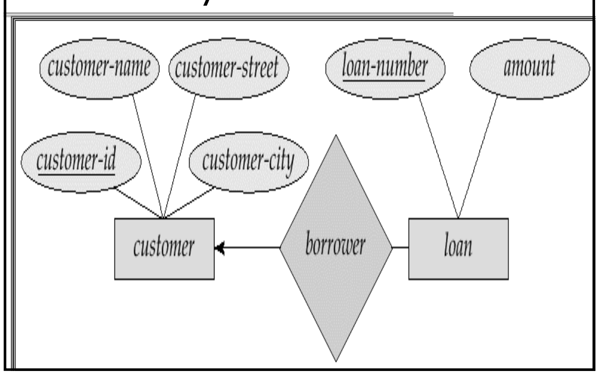
Many-one and many-many



Representing "Multiplicity"

- Show a many-one relationship by an arrow entering the "one" side.
- Show a one-one relationship by arrows entering both entity sets.
- **Rounded arrow** = "exactly one," i.e., each entity of the first set is related to exactly one entity of the target set.

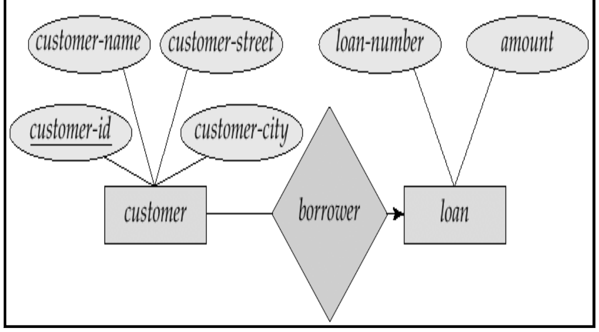
1- many



Many-1 Example

- Favorite, from Drinkers to Beers is many-one.
- A drinker has at most one favorite beer.
- But a beer can be the favorite of any number of drinkers, including zero.

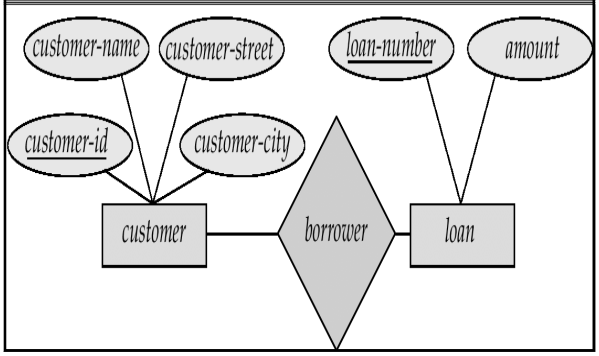
Many – 1: Example 2



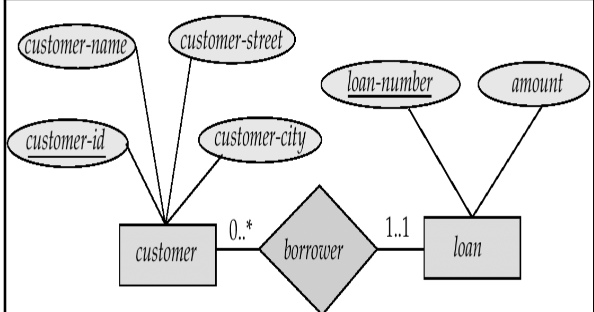
Many-Many Relationships

- Focus: binary relationships, such as Sells between Bars and Beers.
- In a *many-many* relationship, an entity of **either** set can be connected to many entities of the other set.
  - E.g., a bar sells many beers; a beer is sold by many bars.

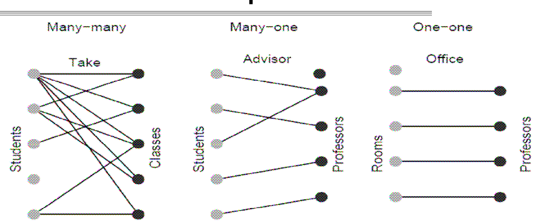
Many - many



Alternative Cardinality Specification

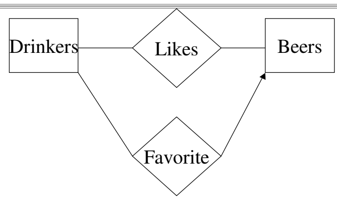


More examples



- Advisor is many-to-one from Students to Professors because each student can have at most one advisor.
- Alternative: Advisor is one-to-many from Professors to Students to because each professor can advise many students.

Another Example of cardinality



- Likes is a Many-Many relationship while
- Favorite is a ?????

There can be only 1 favorite beer for a drinker but
A given beer may be favorite for many drinkers => Many-1

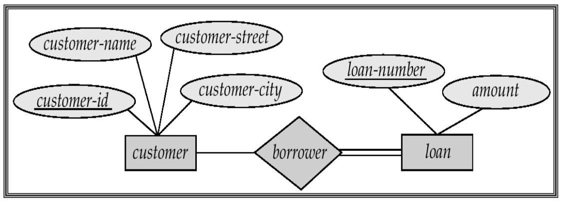
Example

- Consider Best-seller between Manfs and Beers.
- Some beers are not the best-seller of any manufacturer, so a rounded arrow to Manfs would be inappropriate.
- But a beer manufacturer has to have a best-seller.

In the E/R Diagram

Participation of an Entity Set in a Relationship Set

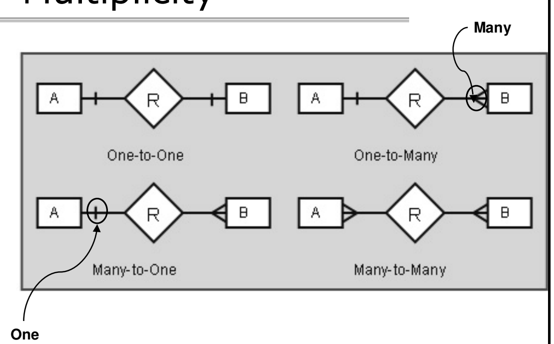
- **Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set**
  - **E.g. participation of *loan* in *borrower* is total**
    - **every loan must have a customer associated to it via borrower**



Participation …

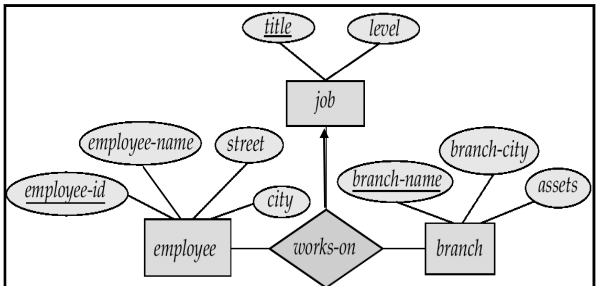
- **Partial participation**: some entities may not participate in any relationship in the relationship set
  - E.g. participation of *customer* in *borrower* is partial

Another notation for Multiplicity



Cardinality Constraints on Ternary Relationship

- We allow **at most one arrow out of a ternary** (or greater degree) relationship to indicate a cardinality constraint



- E.g. an arrow from *works-on* to *job* indicates each employee works on at most one job at any branch.

Why?

- If there is more than one arrow, there are two ways of defining the meaning. E.g a ternary relationship *R* between *A*, *B* and *C* with arrows to *B* and *C* could mean

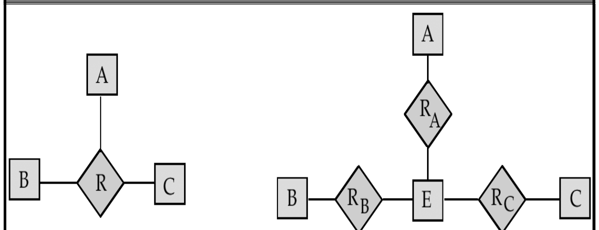
1. each *A* entity is associated with a unique entity from *B* and *C* or
2. each pair of entities from (*A, B*) is associated with a unique *C* entity, and each pair (*A, C*) is associated with a unique *B*

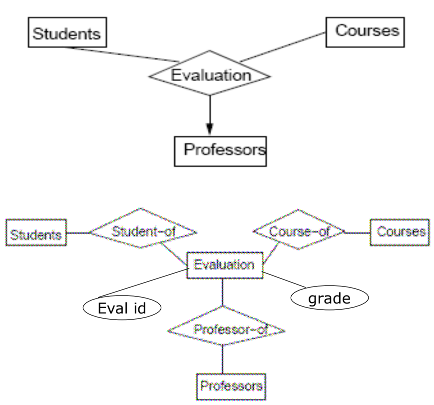
- ***To avoid confusion we outlaw more than one arrow***

Converting Ternary to Binary

- Create a new connecting entity set.
- Introduce relationships from the connecting entity set to each of the entities in the original relationship.
- If an entity set plays more than one role, create a relationship for each role.

In pictures





Relationship Sets

Student	*Course*	*Professor*	*Grade*
Hermione Grainger	Potions	Snape	F-
Draco Malfoy	Potions	Snape	A*
Harry Potter	Potions	Lupin	A+
Ron Weasley	Potions	Lupin	B+

Evaluation Before Conversion

Evaluation

Eval_Id	*Grade*
e1	F-
e2	A*
e3	A+
e4	B+

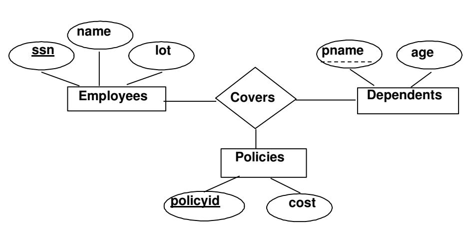
Student_of

Eval_Id	Student
e1	Hermione Grainger
e2	Draco Malfoy
e3	Harry Potter
e4	Ron Weasley

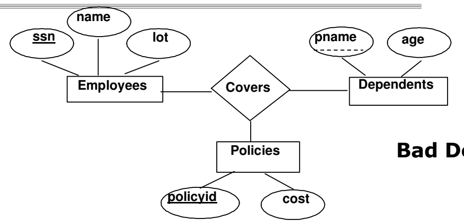
Details of the conversion

- Create an entity in the new `Evaluation` entity set for each instance (row) in the *ternary* `Evaluation` relationship.
- In the `Student-of` relationship, relate each entity in the `Evaluation` entity set with the corresponding student entity.
- Similarly for `Course-of` and `Professor-of` relationships

Binary vs. Ternary Relationships

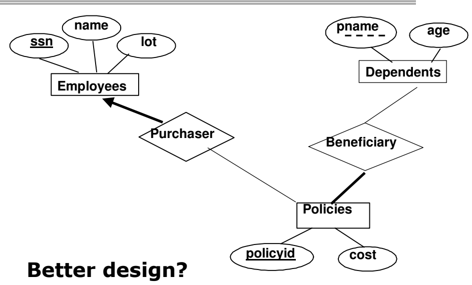


Binary vs. Ternary Relationships



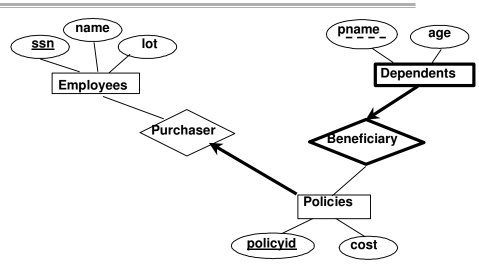
Bad Design

- What if each policy is owned by just 1 employee?
- What if each dependent should be tied to only 1 covering policy?



Better design?

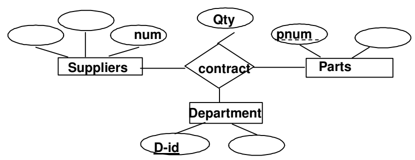
Binary vs. Ternary Relationships



Even Better Design

Binary vs. Ternary Relationships

- Previous example illustrated when two binary relationships better than one ternary relationship.
- How about ternary relation Contracts :
  - entity sets Parts, Departments and Suppliers,
  - and has descriptive attribute *qty*.

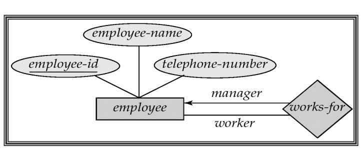


Binary vs. Ternary Relationships

- Ternary relation `Contracts` relates entity sets `Parts`, `Departments` and `Suppliers`, and has attribute *qty*.
- What about following binary relationships :
  - S "can-supply" P,
  - D "needs" P, and
  - D "deals-with" S
- No combination of binary relationships is an adequate substitute:
  - Together 3 binary relationships don't imply that D has agreed to buy P from S.
  - Also, how could we record *qty*?

Self Relationship

- Sometimes entities in a entity set may relate to other entities in the same set. Thus self relationship



- Here employees mange some other employees
- The labels "manger" and "worker" are called *roles* the self relationship

More examples on self-relationship

- People to people
1. Parent – children
2. Manager – employee
3. Husband – wife

- Word to word
  - Root – synonym

Roles

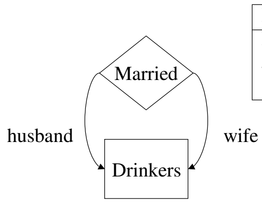
- Sometimes an entity set appears more than once in a relationship.

- Label the edges between the relationship and the entity set with names called *roles*.

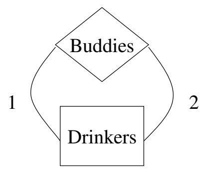
Example

Relationship Set

Husband	Wife
Bob	Ann
Joe	Sue
…	…



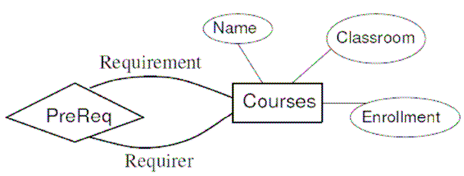
Example



Relationship Set

Buddy1	Buddy2
Bob	Ann
Joe	Sue
Ann	Bob
Joe	Moe
…	…

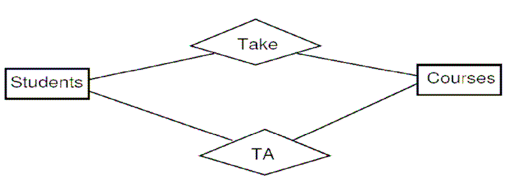
Courses Example



Parallel Relationships

Can there be more than one relationship between the same pair of entities?

Example: TA and Take relationship between Students and Classes



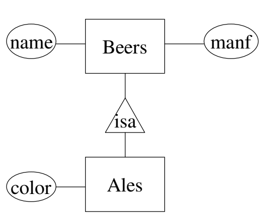
Subclasses

- *Subclass* = special case = fewer entities = more properties.
- Example: Ales are a kind of beer.
  - Not every beer is an ale, but some are.
  - Let us suppose that in addition to all the *properties* (attributes and relationships) of beers, ales also have the attribute color.

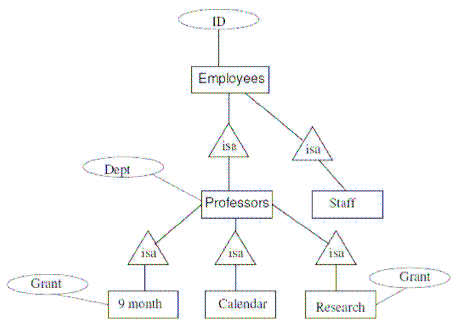
Subclasses in E/R Diagrams

- Assume subclasses form a tree.
  - I.e., no multiple inheritance.
- Is-a triangles indicate the subclass relationship.
  - Point to the superclass.

Example



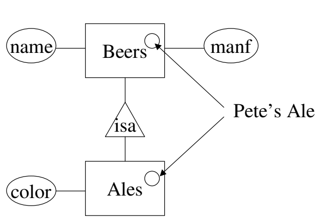
Example of a sub class tree



E/R Vs. Object-Oriented Subclasses

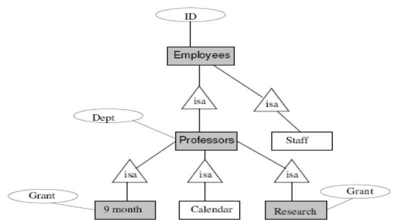
- In OO, objects are in one class only.
  - Subclasses inherit from superclasses.
- In contrast, E/R entities have representatives in all subclasses to which they belong.
  - Rule: if entity *e* is represented in a subclass, then *e* is represented in the superclass.

Example



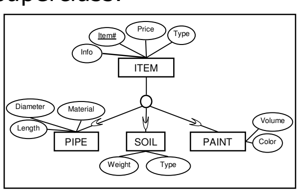
Components of an Entity

Prof. Fingers InMany Pies teaches in one semester every year and does not teach in the other semester. In the other semester, his research grant pays his salary. Which entity sets does he have components in?



Attribute Inheritance

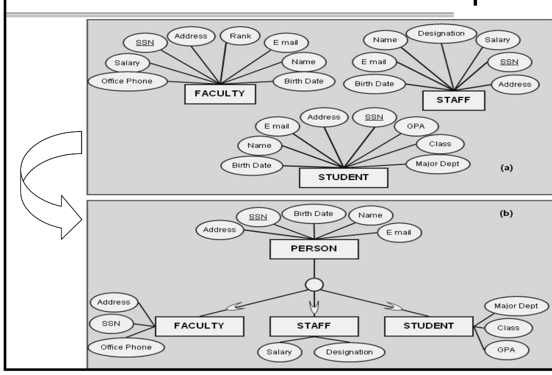
- It is the property by which subclass entities inherit values for all attributes of the superclass.



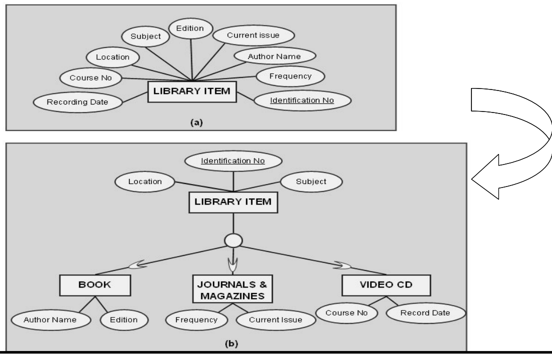
Instance of an ITEM entity
Length: 10”
Diameter: 2”
Material: Cast Iron
Item #: 12599
Price: $0.5 /lb
Info: Rust Proof
Type: Pipe

Inherited Attributes

Generalization - Example



Specialization - Example



Participation and Disjoint Constraints

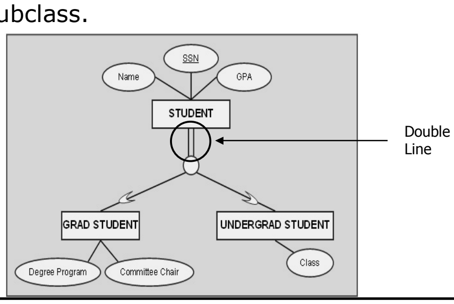
- Constraints are intuitive and help us manifest business rules and incorporate them into the EER design.

- Participation Constraints
  - Dictate whether every member of a superclass must participate as a member of a subclass.
  - May be Total Participation or Partial Participation.

- Disjoint Constraints
  - Define whether it is possible for an instance of a superclass to be a member or one or more subclasses simultaneously.
  - May be Disjoint Rule or Overlap Rule.

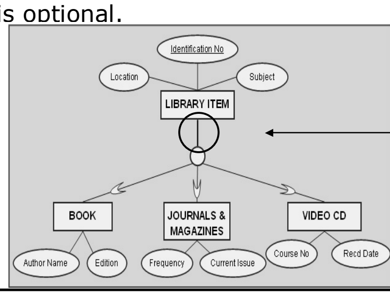
Total Participation Rule

- Every member or instance of a superclass must be a member of at least one subclass.



Partial Participation Rule

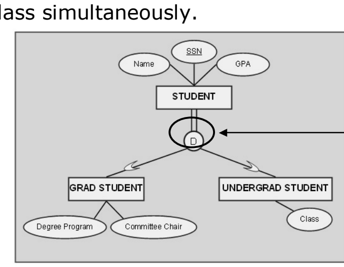
- Member of a superclass does not have to be member of any subclass. Membership is optional.



Single Line

Disjoint Rule

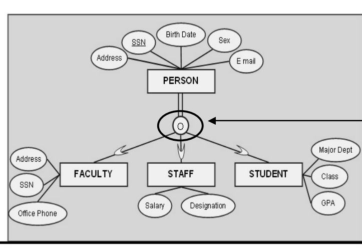
- If an instance of superclass is a member of any subclass, it cannot be an instance of any other subclass simultaneously.



Letter "D"

Overlap Rule

- If an instance of a superclass is a member of any subclass, it can also belong to any other subclass simultaneously.



Letter "O"

Constraints : putting together

<u>Participation Constraints</u>

Total Participation Rule

PartialParticipation Rule

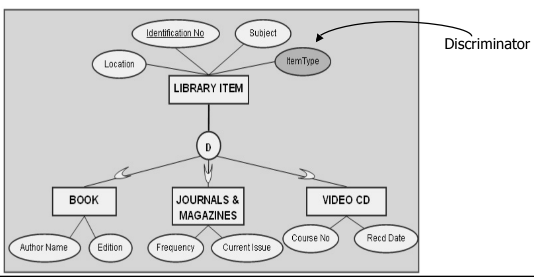
<u>Disjoint Constraints</u>

Disjoint Rule

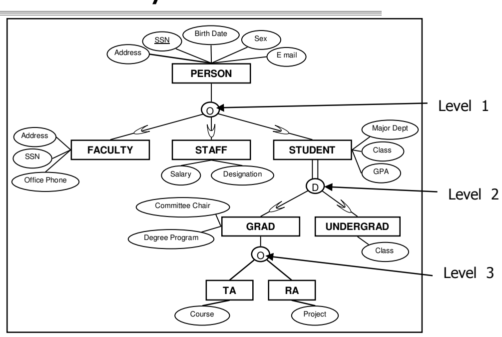
Overlap Rule

Subclass Discriminators

- An attribute of a superclass that discriminates a new entry to the superclass into appropriate subclasses.



Superclass/Subclass Hierarchy

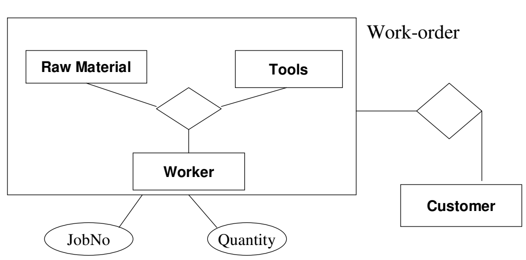


Aggregation

- **Treats a relationship as an entity**
  - **used to express a relationship among relationships**
- For building complex entity from existing entities (or existing entities and relationships)
- Two ways of defining complex entities :
  - create an attribute whose value is another entity
  - define an entity as containing a group of related entities

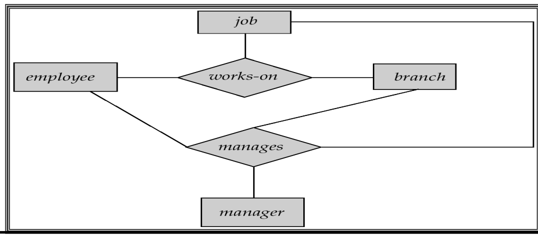
Examples :

- Work-order object (entity) defined as consisting of entities Raw-material, Tools and Workers;
- Work-order itself related with Customer entity
- Aggregation notation not explicitly provided in Extended E-R model



Aggregation

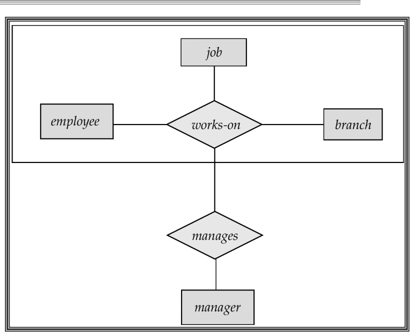
- Consider the ternary relationship *works-on*, which we saw earlier
- Suppose we want to record managers for tasks performed by an employee at a branch

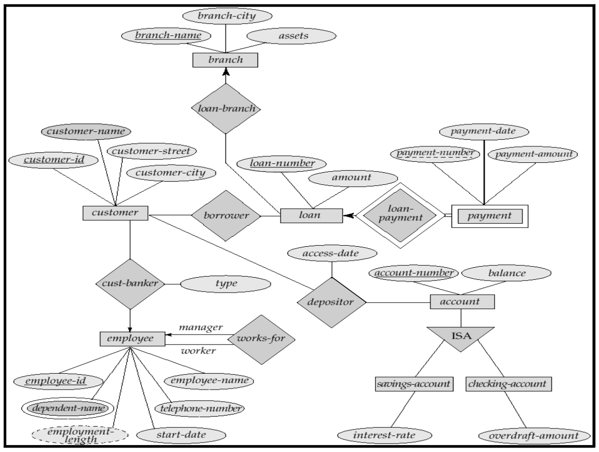


Aggregation (Cont.)

- Relationship sets *works-on* and *manages* represent overlapping information
  - Every *manages* relationship corresponds to a *works-on* relationship
  - However, some *works-on* relationships may not correspond to any *manages* relationships
    - So we can't discard the *works-on* relationship
- Eliminate this redundancy via *aggregation* as shown in the following diagram :
  - An employee works on a particular job at a particular branch
  - An employee, branch, job combination may have an associated manager

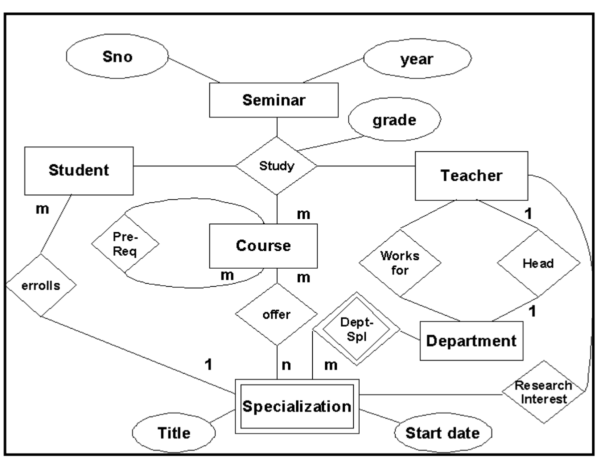
E-R Diagram With Aggregation



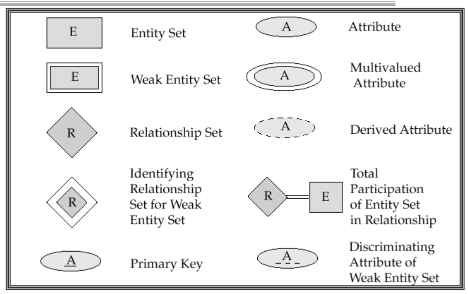


EXERCISE (Post-Graduate studies)

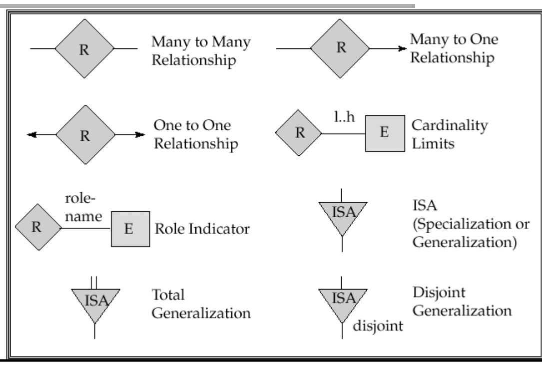
Students join a particular specialization offered by a department. A specialization with same title (e.g., MICROCOMPUTERS) may be offered by one/more depts independently. Teachers are appointed to a specific dept, and given a room and telephone. Depts have some teacher as its head. Courses are offered under various specializations. A teacher may teach many courses and a course may be taught by many. A student studies a course under a teacher during some semester (e.g., semester 1 of 2003), and is awarded a grade. A teacher's research interest may lie in one/more specializations. Courses have one/more/zero prerequisites



Summary of Symbols Used in E-R Notation



Summary of Symbols (Cont.)



Alternative E-R Notations