Entity Relationship Model

Prepared by Soky KAK, PhD

Outline

- ER Model Basic Concept
- ER Diagram Representation
- Generalization

Purpose of E/R Model

- ✓ The E/R model allows us to sketch database schema designs.
  - Includes some constraints, but not operations.
- ✓ Designs are pictures called *entity-relationship diagrams*.
- ✓ Later: convert E/R designs to relational DB designs.

Framework for E/R

- Design is a serious business.
- The “boss” knows they want a database, but they don’t know what they want in it.
- Sketching the key components is an efficient way to develop a working database.

Entity

- ✓ An entity is an object that exists and is distinguishable from other objects. Example: specific person, company, event, plant.
- ✓ It may be an object with physical existence (person, house) or it may be an object with conceptual existence (company, job, university course)
- ✓ Entity has attributes. Example: people have names and addresses.

Entity examples

- ✓ Each person in an organization is an entity.
- ✓ If the organization employs 150 people, the it means the organization has 150 entities of person.

Entity Sets

- ✓ An *Entity set* is collection of similar entities or a named collection of entities having the same attributes.
- ✓ *Attribute* = property of (the entities of) an entity set.
  - Attributes are simple values, e.g. integers or character strings, not structs, sets, etc.

Example

- ✓ Persons:
- ✓ {

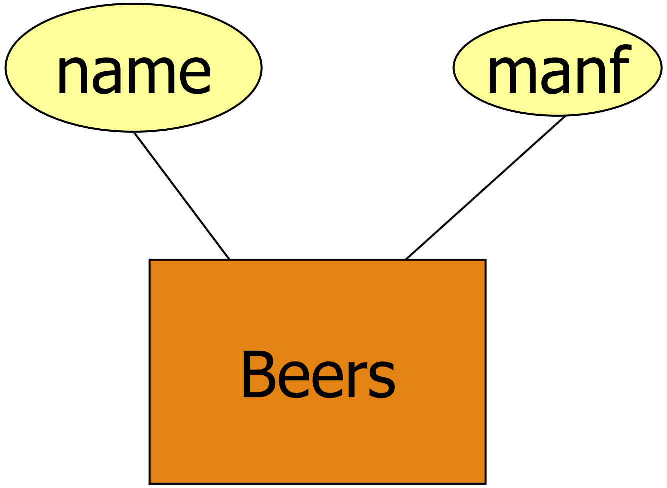
<“001”, “Jonh”, “Male”, “01-Jan-1999”>,

<“002”, “Smith”, “Female”, “07-Jan-1979”>

- ✓ The entity set has two entities.

Example:

- ✓ Entity set Beers has two attributes, name and manf (manufacturer).
- ✓ Each Beers entity has values for these two attributes, e.g. (Cambodia, Khmer Brewery)



Entity Type

- ✓ An entity type is a named set of attributes of similar entities.
- ✓ Examples of entity types: Person, Book and Course.

  Person: {person_id, name, dob, address}

  Book: {book_id,title, author, publisher, price}

  Course: {course_id, title, teacher}

- ✓ Note: each id columns (person_id, book_id, course_id) is used to uniquely identify each entity.

Entity types

✓There are 6 types of entity type:

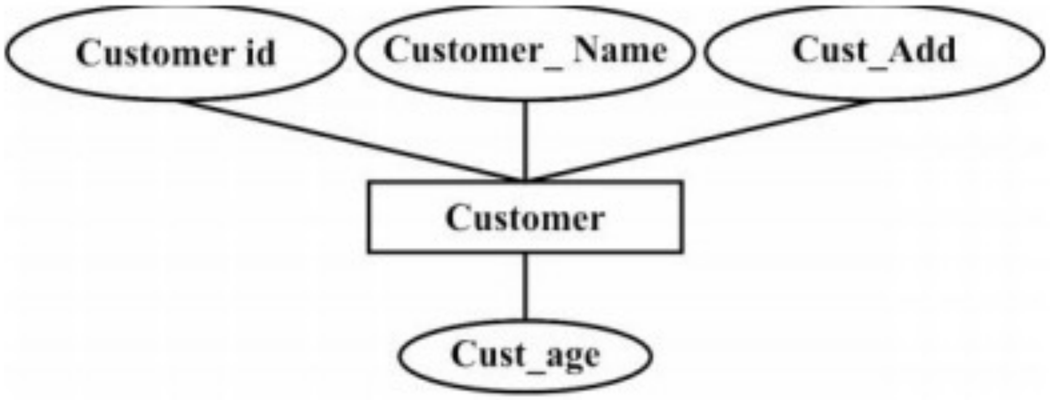
- Traditional Entity
- Strong Entity
- Weak Entity
- Associative/Composite Entity
- Recursive Entities
- Subtype/Supertype Entity.

Traditional Entity

- ✓ The traditional entity, also known as the simple entity or normal entity.
- ✓ This generally only has one primary key associated with it.
- ✓ This is never associated with many-to-many type relationships.

Strong Entity

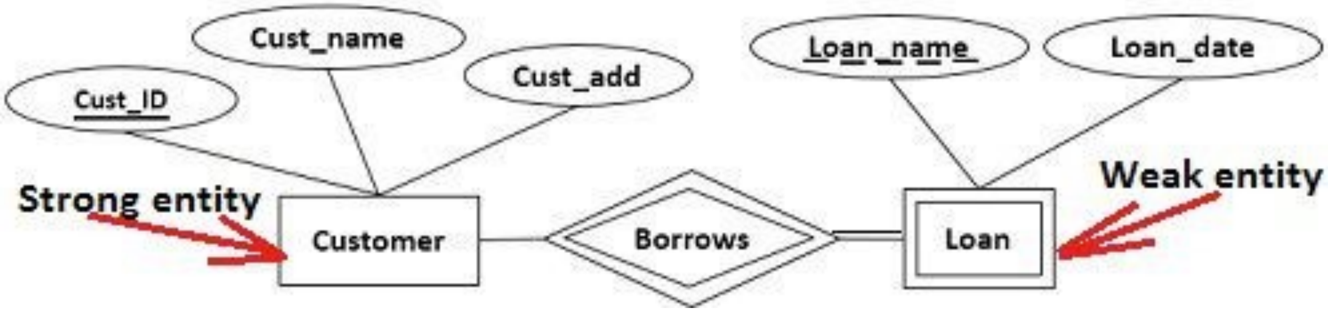
- ✓ An entity type which has own distinct primary key that used to identify specific uniquely from another entity type is called as Strong Entity type.
- ✓ An Entity type which is independent on some other entity type is called Strong Entity type



Customer Entity

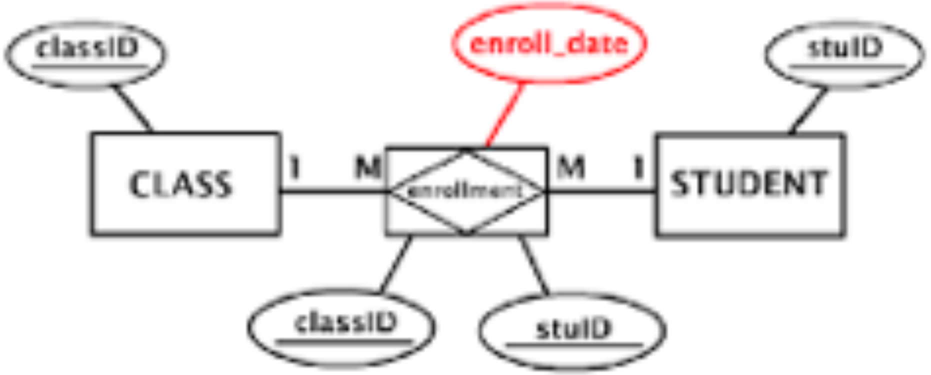
Weak Entity

- ✓ Entity type which is dependent on some other entity type is called as Weak entity type.
  - Weak entity type is dependent on a strong entity and cannot exist on its own.
  - It does not have a unique identifier that has partial identifier.
  - Partial identifier is represented by double-line



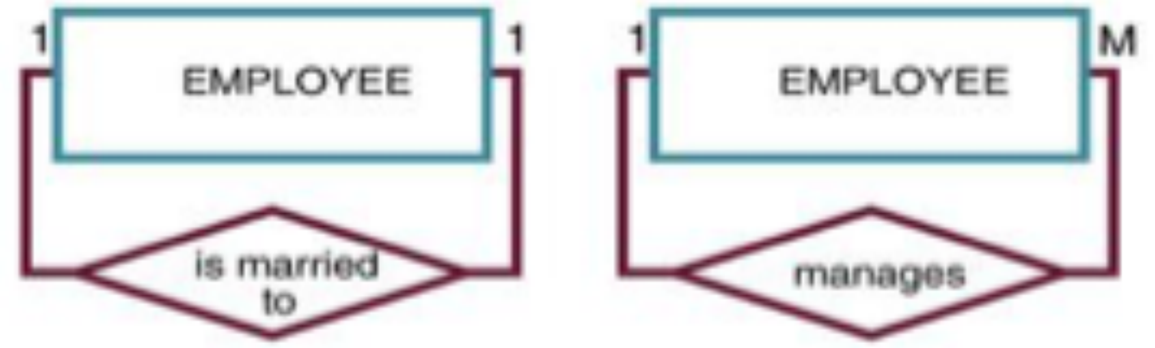
Composite entity

- ✓ A composite entity is also known as a “bridge” entity which is used to handle the many-to-many relationships that the traditional/simple entity could not handle.
- ✓ This entity lies between the two entities that are of interest.
- ✓ This composite entity shares the primary keys from both the connecting tables.



Recursive Entity

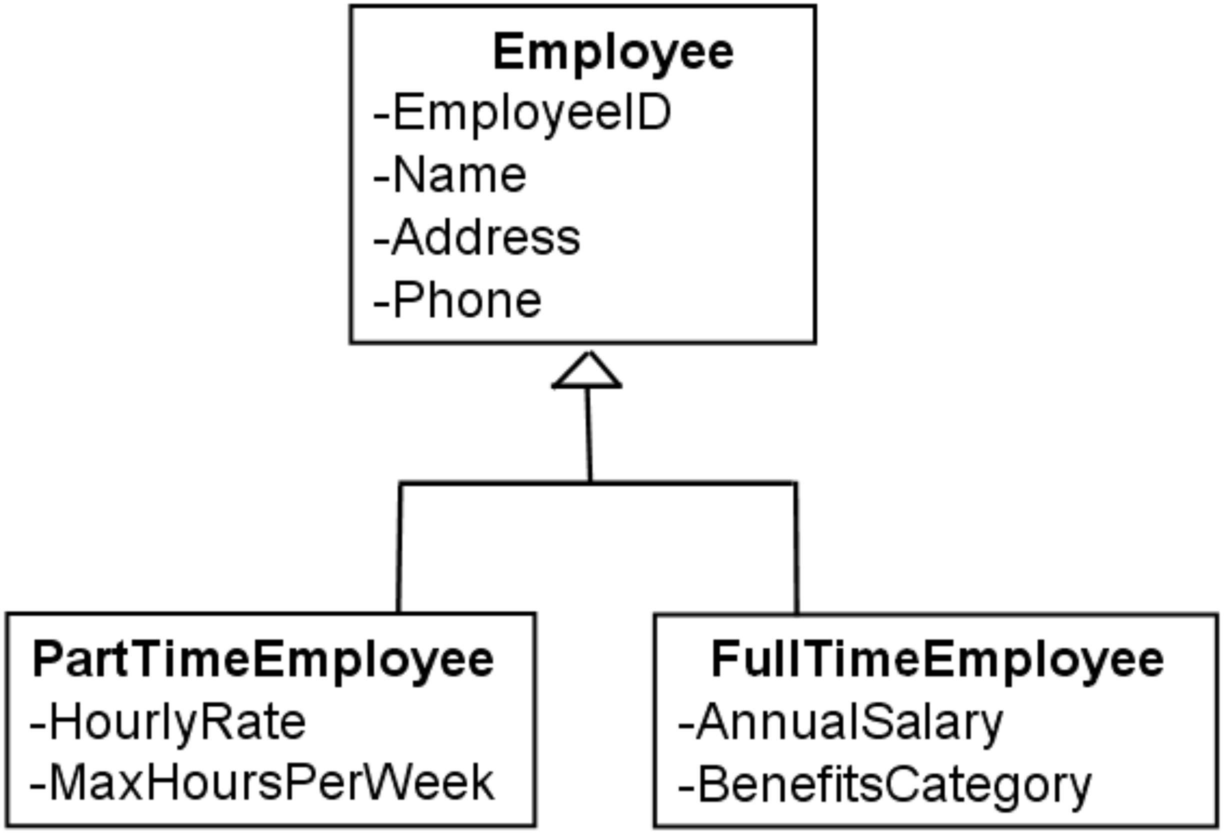
- ✓ A recursive entity is one in which a relationship can exist between occurrences/entities of the same entity set.
- ✓ A recursive entity is found within a unary relationship



Supertype/Subtype entity

- ✓ The supertype (parent) entity is the top most entity that shares the information down to the subtypes (children).
- ✓ The subtypes inherit all the information from the supertype entity.
- ✓ When moving down the hierarchical supertype to the subtype, that is known as specialization.
- ✓ When moving from the subtype to the supertype, that is known as generalization.

Example


Employee
-EmployeeID
-Name
-Address
-Phone
PartTimeEmployee
-HourlyRate
-MaxHoursPerWeek
FullTimeEmployee
-AnnualSalary
-BenefitsCategory

Attributes

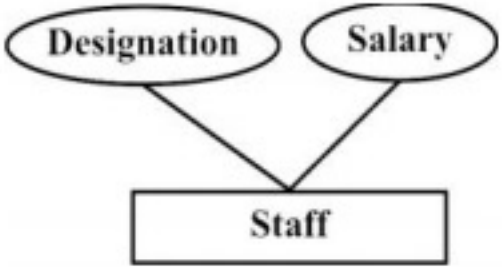
- ✓ Entities are represented by means of their properties, called attributes.
- ✓ All attributes have values. For example, a student entity may have name, class, and age as attributes.

Types of Attributes

- ✓ Simple attribute
- ✓ Composite attribute
- ✓ Derived attribute
- ✓ Single-value attribute
- ✓ Multi-value attribute

Simple attribute

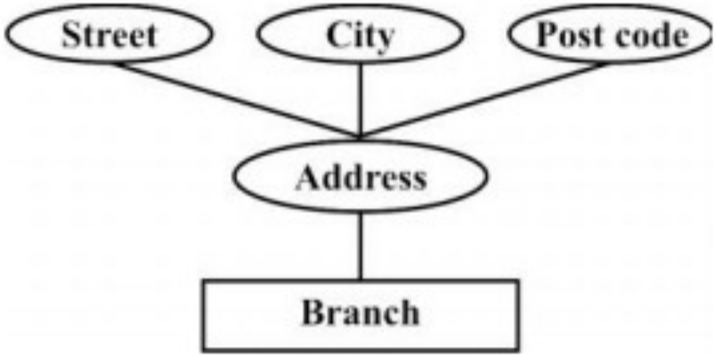
- ✓ Simple attributes are atomic values, which cannot be divided further.
- ✓ For example, a student's phone number is an atomic value of 10 digits.



Simple Attributes

Composite attribute

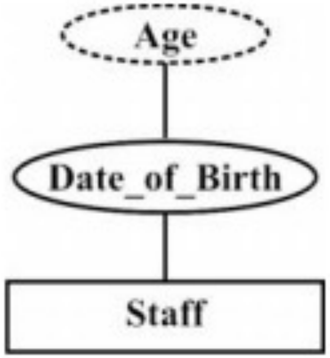
- ✓ Composite attributes are made of more than one simple attribute.
- ✓ For example, a student's complete name may have first_name and last_name.



Composite Attributes

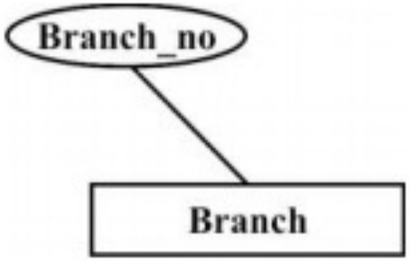
Derived attribute

- ✓ Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.
- ✓ For example, average_salary in a department should not be saved directly in the database, instead it can be derived.
- ✓ For another example, age can be derived from data_of_birth.



Single-value attribute

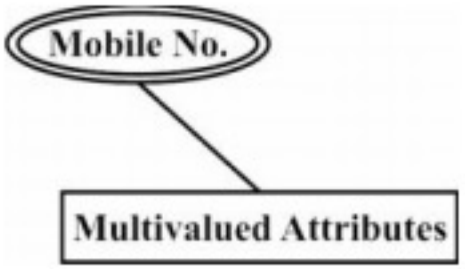
- ✓ Single-value attributes contain single value.
- ✓ For example − Social_Security_Number, ID No.



Single Valued attributes

Multi-value attribute

- Multi-value attributes may contain more than one values.
- For example, a person can have more than one phone number, email_address, etc.



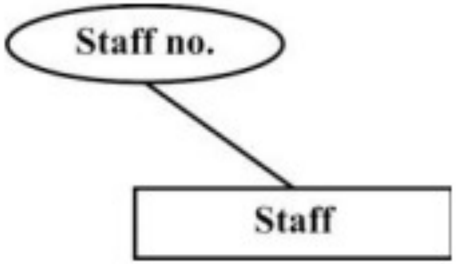
Multivalued Attributes

Null attribute

- ✓ The attribute which take NULL value when entity does not have the value to it.
- ✓ The Null attribute is an attribute their value is unknown, unassigned and missing information

Key attribute

- ✓ This attribute has the unique value for an entity which is used to identified given row in the table is called as key attribute of an entity.



Domain

- ✓ A domain is a set of values an attribute of an entity set can have in an entity.
- ✓ Sample domains:

Attribute	Domain
First name	Set of 30 character strings
color	Red, green, silver, white
price	3000 to 10000
rating	0 to 10

Keys

- Key plays an important role in relational database; it is used for identifying unique rows from table.
- A key is a minimal set of attributes whose values uniquely identify an entity in an entity set.
- It also establishes relationship among tables.

Entity Set	Key
Persons	Person_id
Books	Book_id
Courses	Course_id
Student Marks	Roll_number, subject_id

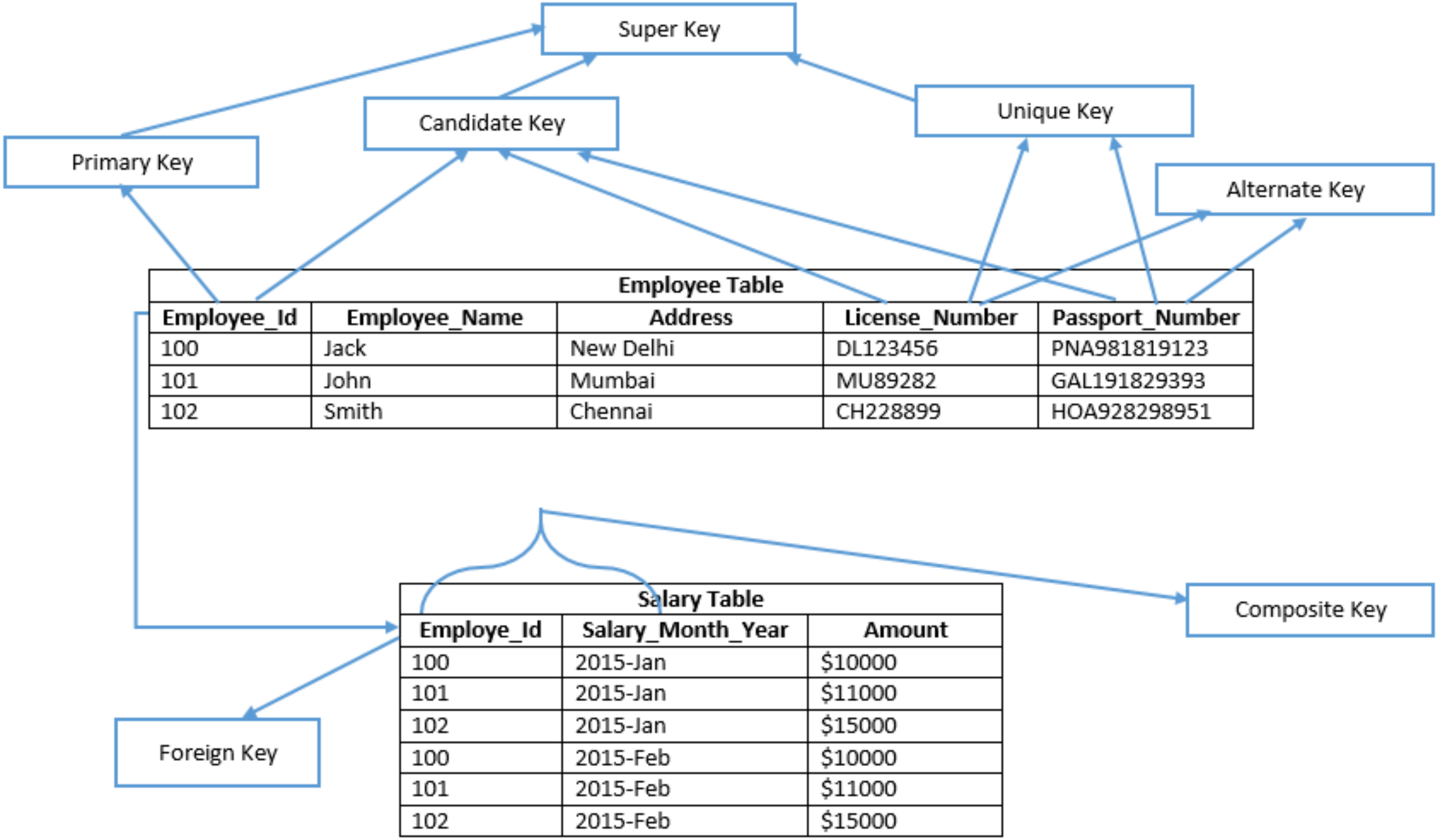
Type of Keys

- ✓ Primary Key – A primary is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.
- ✓ Super Key – A super key is a set of one of more columns (attributes) to uniquely identify rows in a table.
- ✓ Candidate Key – A super key with no redundant attribute is known as candidate key

Type of Keys (Cont.)

- ✓ Alternate Key – Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.
- ✓ Composite Key – A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.
- ✓ Foreign Key – Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

Example of Keys



Super Key

Candidate Key

Unique Key

Primary Key

Alternate Key

Composite Key

Foreign Key

Employee Table

Employee_Id	Employee_Name	Address	License_Number	Passport_Number
100	Jack	New Delhi	DL123456	PNA981819123
101	John	Mumbai	MU89282	GAL191829393
102	Smith	Chennai	CH228899	HOA928298951

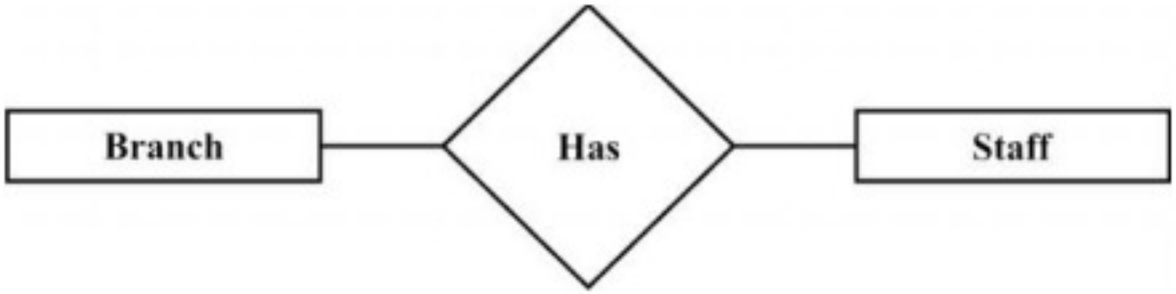
Salary Table

Employe_Id	Salary_Month_Year	Amount
100	2015-Jan	$10000
101	2015-Jan	$11000
102	2015-Jan	$15000
100	2015-Feb	$10000
101	2015-Feb	$11000
102	2015-Feb	$15000

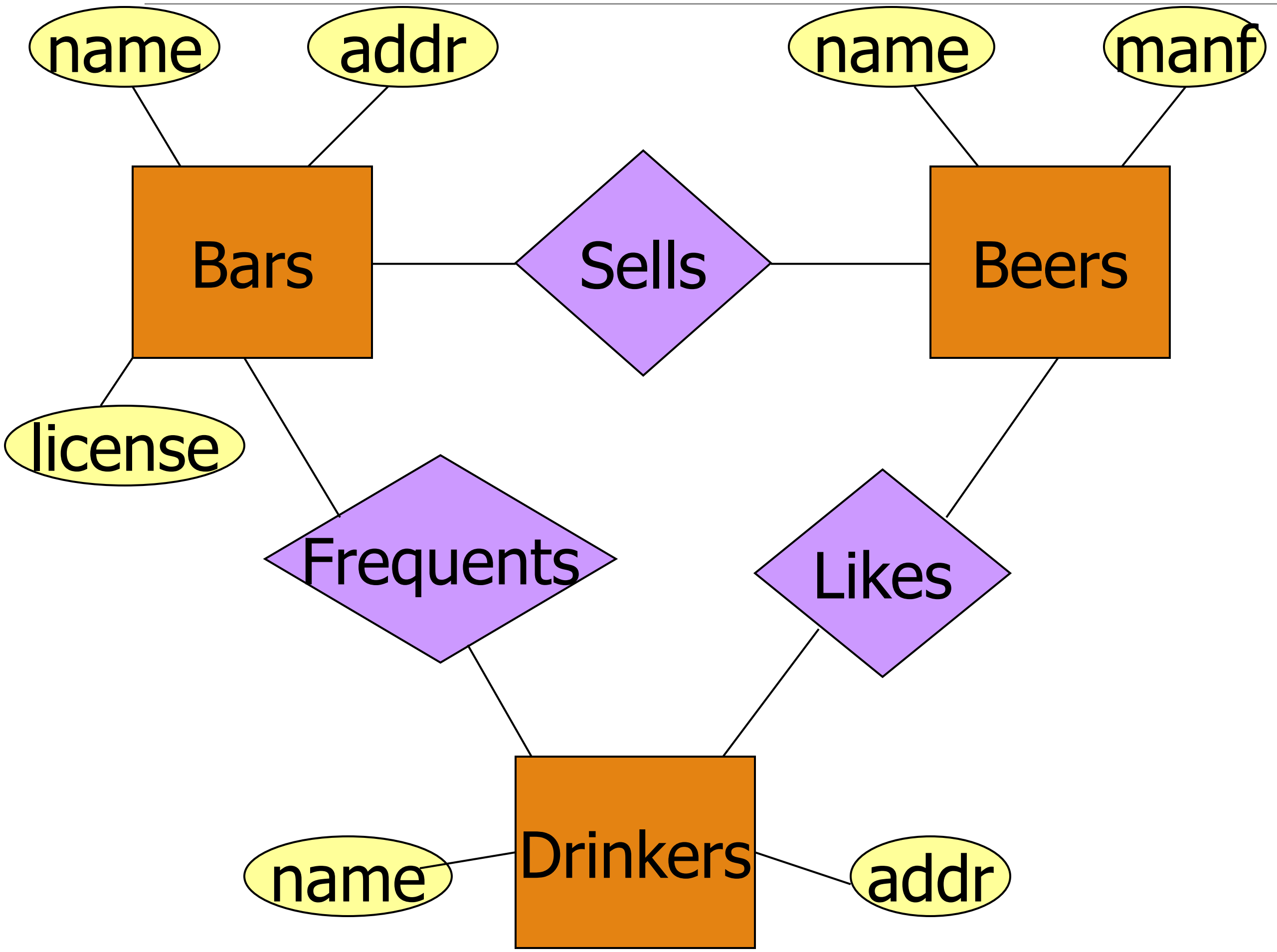
Relationship

- ✓ A relationship connects two or more entity sets.
- ✓ It is represented by a diamond, with lines to each of the entity sets involved.

Example : Branch has a staff



Example: Relationship



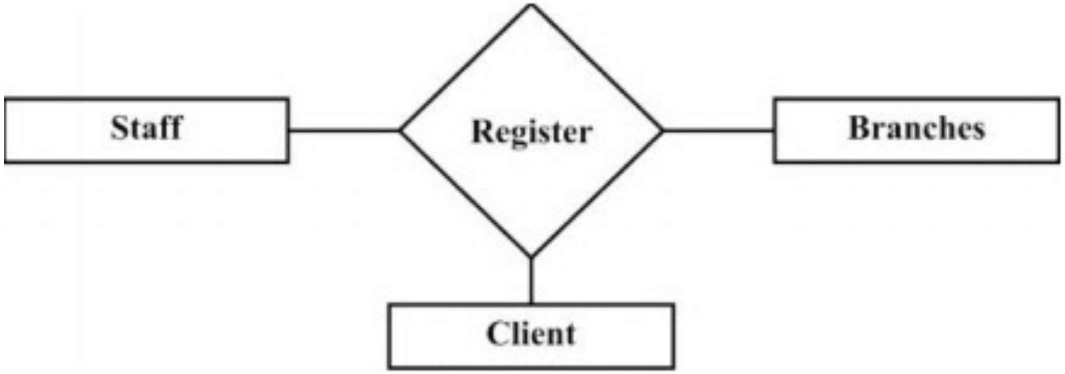
Bars sell some beers.

Drinkers like some beers.

Drinkers frequent some bars.

Degree of relationship

- ✓ It is the number of entities participated in a particular relational model.
- ✓ There are two type of degree of relationship.
  - Binary = degree 2
  - Ternary = degree 3
  - n-ary = degree n



Staff registers a Client at a Branch is a Ternary relationship

Relationship Type

- ✓ A relationship type represents collection of associations of similar type.
- ✓ Similar types of associations means associations having the same attributes from participating entity types or sets.
- ✓ Attribute set (person_id, book_id) represents a relationship type.
- ✓ A name of relationship type is Borrower(person_id, book_id)

Relationship Set

- ✓ The collection of similar relationship is known as Relationship set.
- ✓ For example relationship set Borrowers of relationship type Borrower(person_id, book_id) could be:

{<1001, 12345>,<1001,12456>,<1002,12398>}

Example: Relationship Set

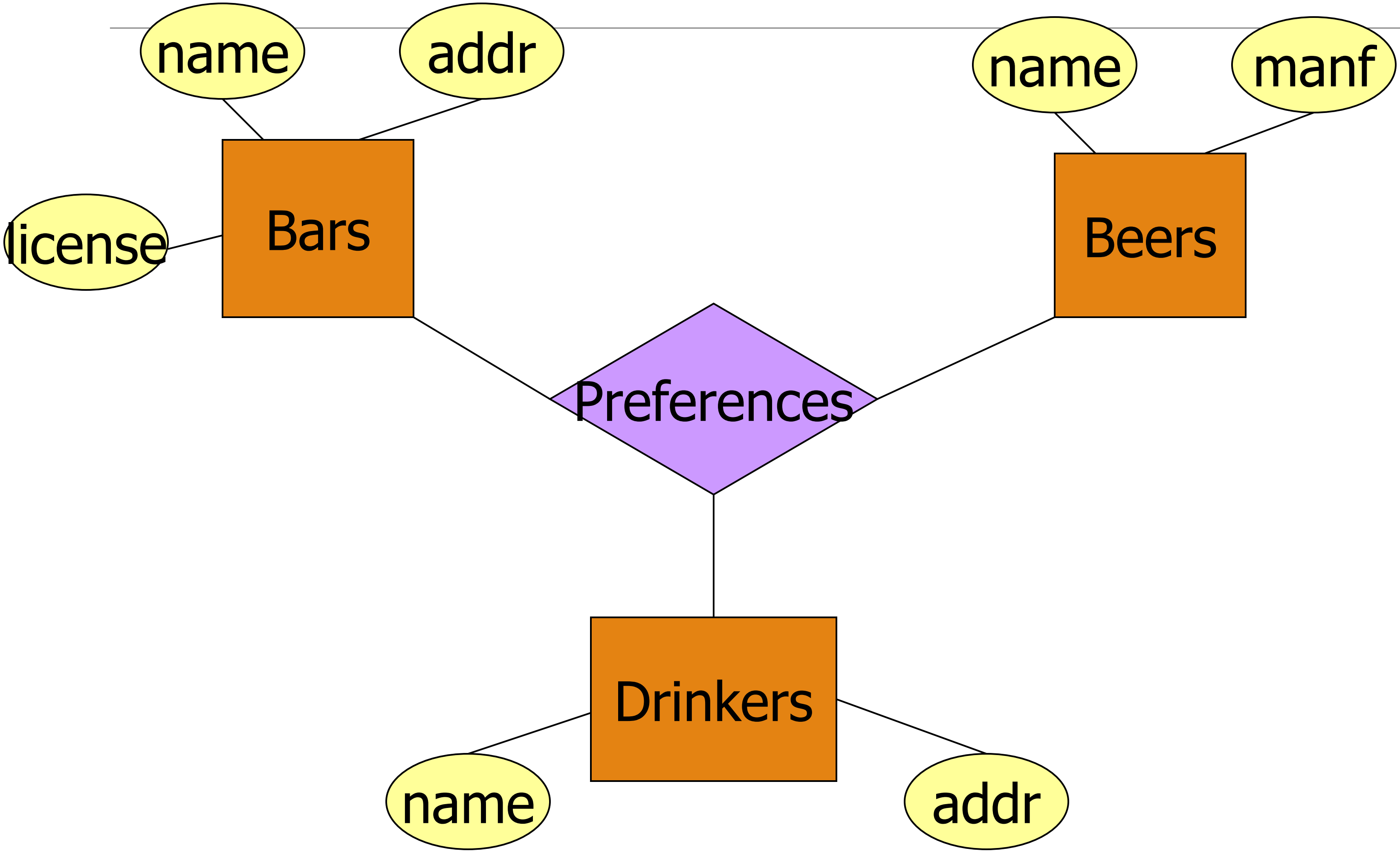
- ✓ For the relationship Sells, we might have a relationship set like:

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

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.

Example: 3-Way Relationship



A Typical Relationship Set

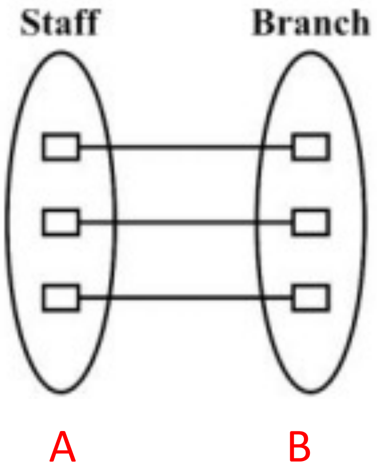
Bar	Drinker	Beer
Joe’s Bar	Ann	Miller
Sue’s Bar	Ann	Bud
Sue’s Bar	Ann	Pete’s Ale
Joe’s Bar	Bob	Bud
Joe’s Bar	Bob	Miller
Joe’s Bar	Cal	Miller
Sue’s Bar	Cal	Bud Lite

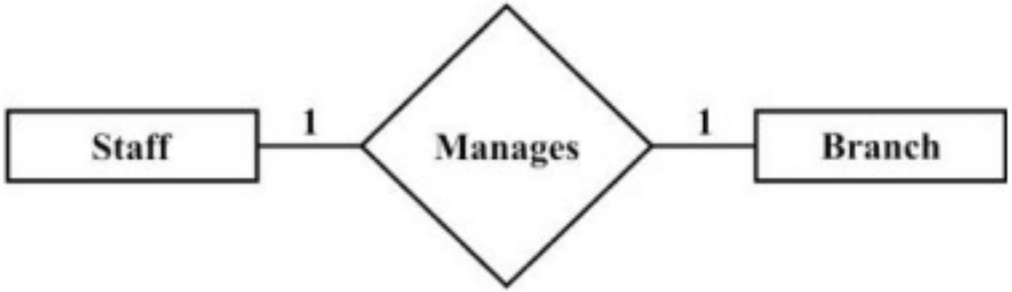
Binary Relationships

- ✓ Binary relationships are relationships between two entity types.
- ✓ Types of binary relationships:
  - One-to-One (1:1) relationship
  - One-to-Many(1:*) or Many-to-One (*:1) relationship
  - Many-to-Many(*:*) relationship

One-to-One Relationship (1:1)

- ✓ In this type of Mapping Constraint One record of an entity is related to the one record of an another entity.
- ✓ That is one row on an table is related to an one row of another table
  - Ex. Each branch is managed by one member of the staff that’s means Branch Manager.

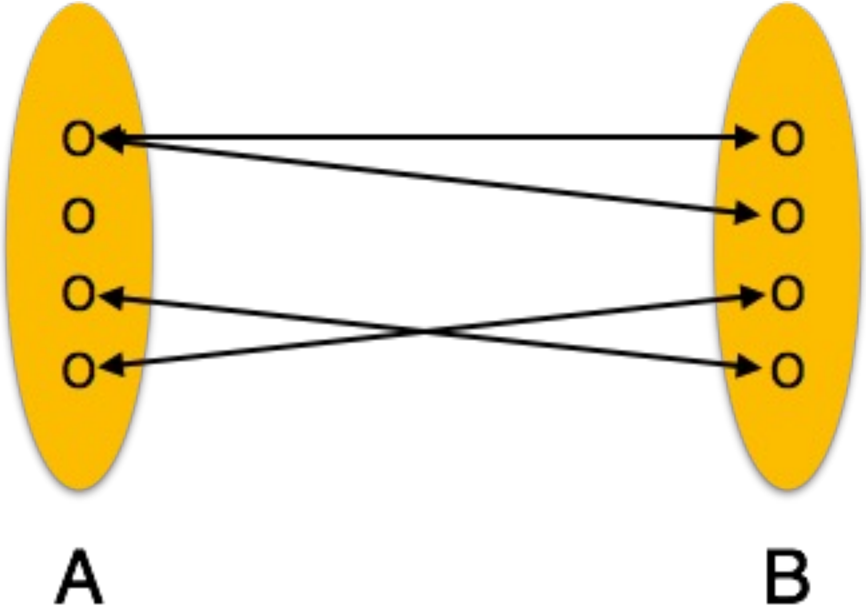




One to one

One-to-Many Relationship (1:*)

- ✓ One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



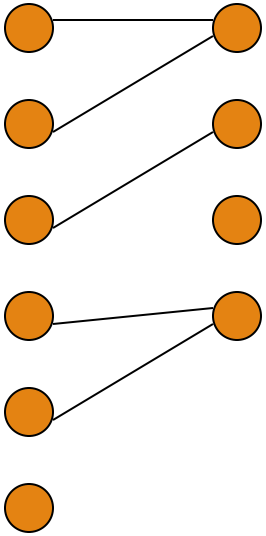
One-to-Many example

- ✓ Lecturer teaches more than one Subjects.

Many-to-One Relationships (*:1)

- ✓ Some binary relationships are *many-to-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.

In Pictures:

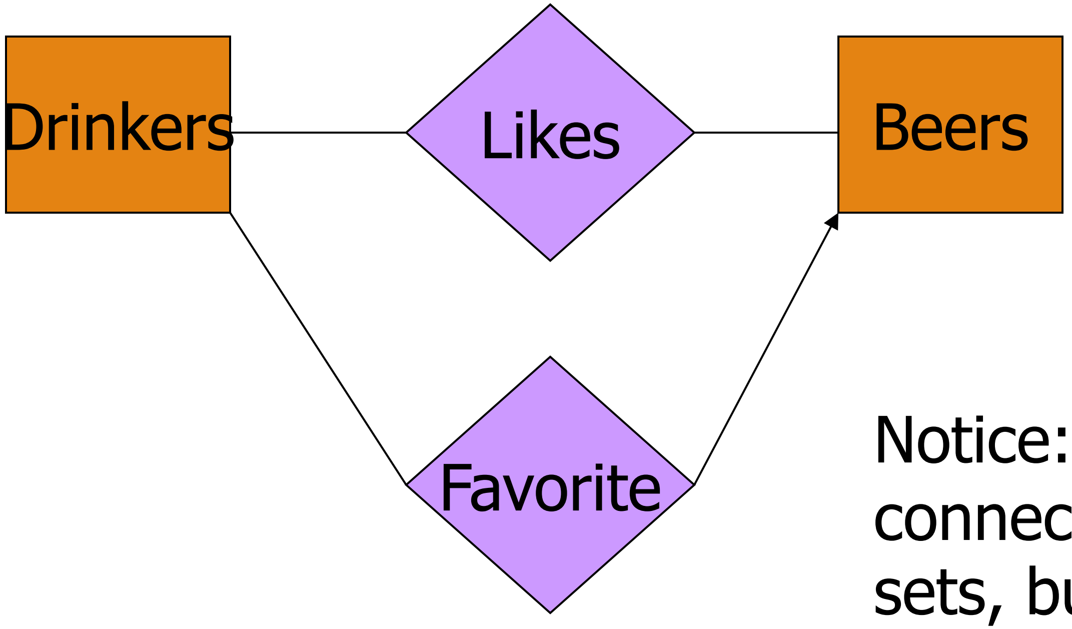


many-one

Example: Many-to-One Relationship

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

Example: Many-One Relationship

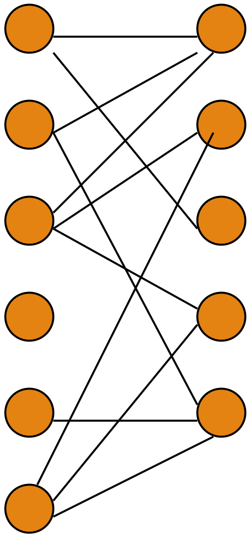


Notice: two relationships connect the same entity sets, but are different.

Many-to-Many Relationships (*:*)

- ✓ Focus: binary relationships, such as Sells between Bars and Beers.
- ✓ In a *many-to-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.

In Pictures:



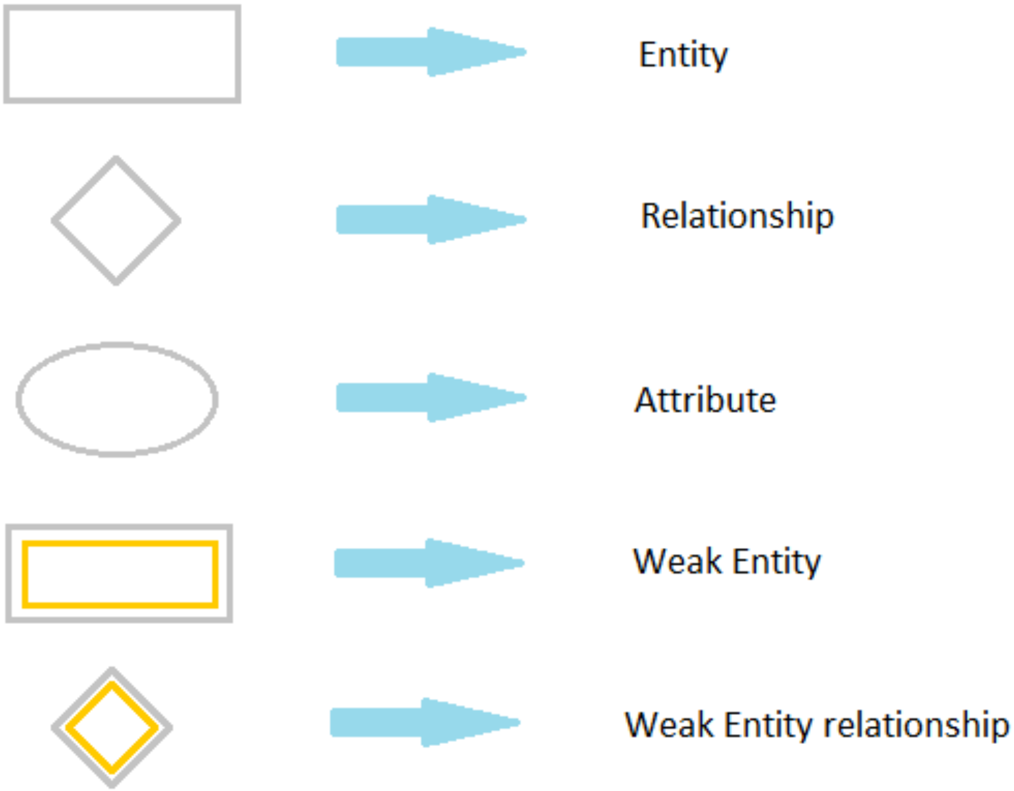
many-many

Entity Relationship (ER) Diagram

- ✓ An ER diagram is a pictorial representation of the information that can be captured by a database.
- ✓ Such a "picture" serves two purposes:
  - It allows database professionals to describe an overall design concisely yet accurately.
  - (Most of) it can be easily transformed into the relational schema.

E/R Diagrams

- In an entity-relationship diagram:
  - Rectangles represent entity sets.
  - Diamonds represent relationship sets.
  - Lines link attributes to entity sets and entity sets to relationship sets.
  - Underline indicates primary key attributes
  - Ellipses represent an attributes
  - Double Lines represent total participation of an entity in a relationship set
  - Double rectangle represent a weak entity sets



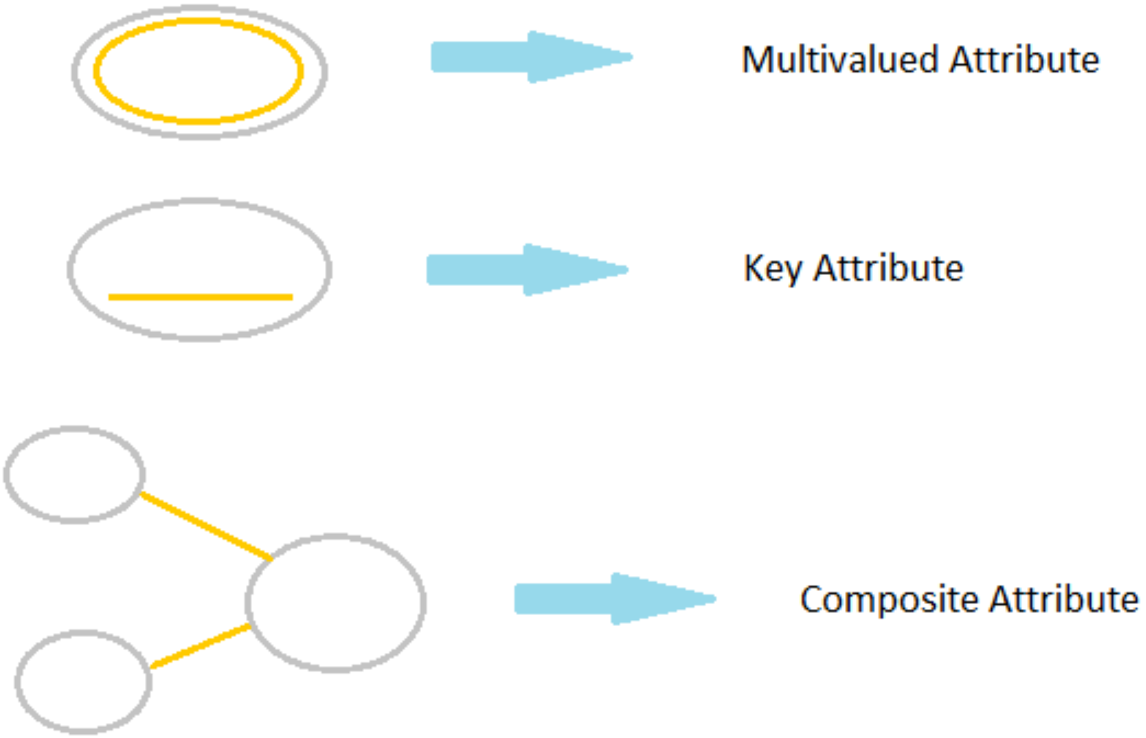
Entity

Relationship

Attribute

Weak Entity

Weak Entity relationship

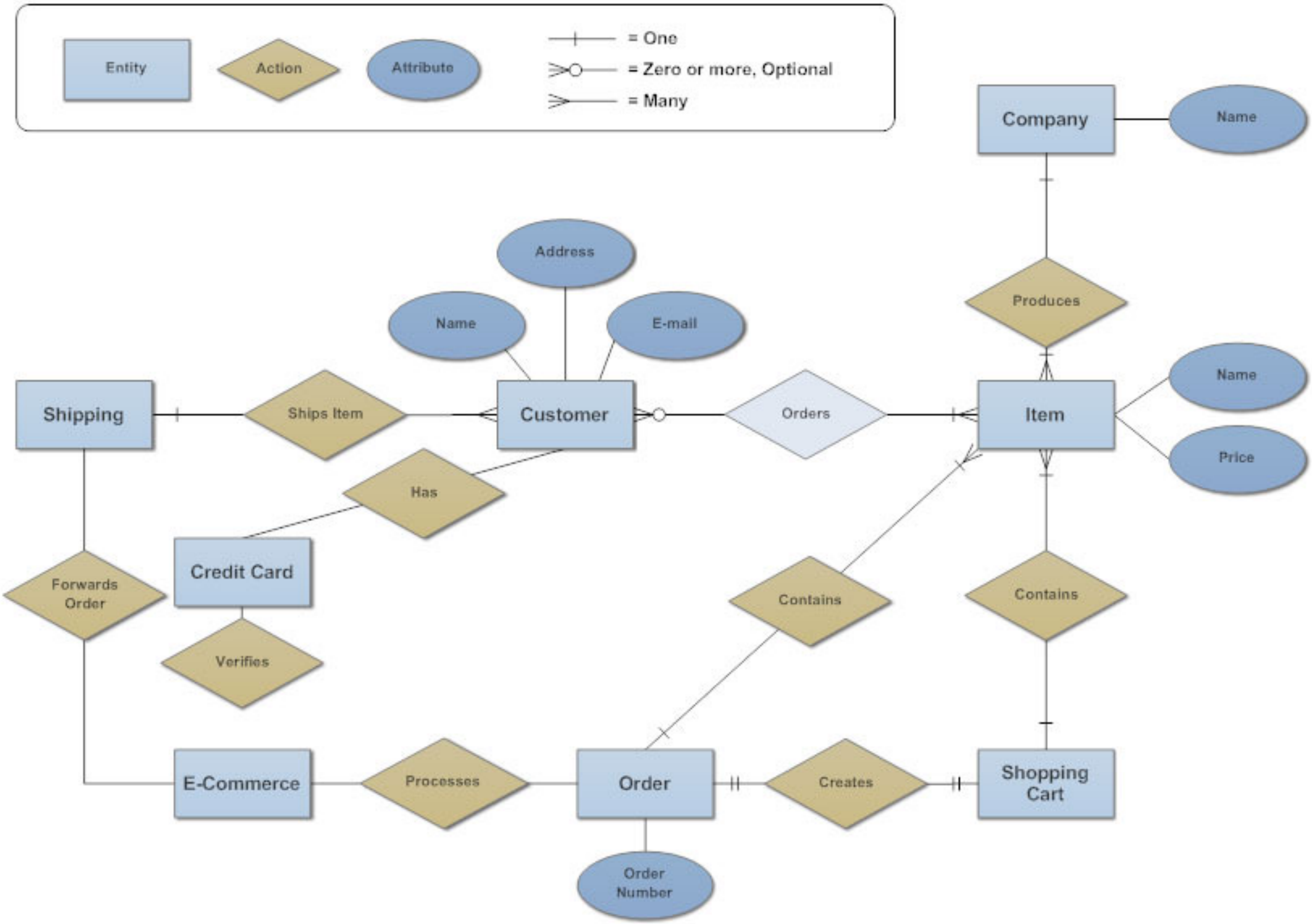


Multivalued Attribute

Key Attribute

Composite Attribute

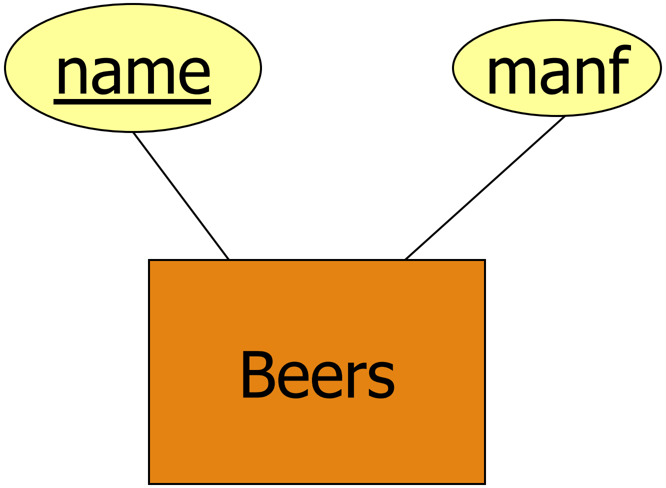
Entity Relationship Diagram - Internet Sales Model



From E/R Diagrams to Relations

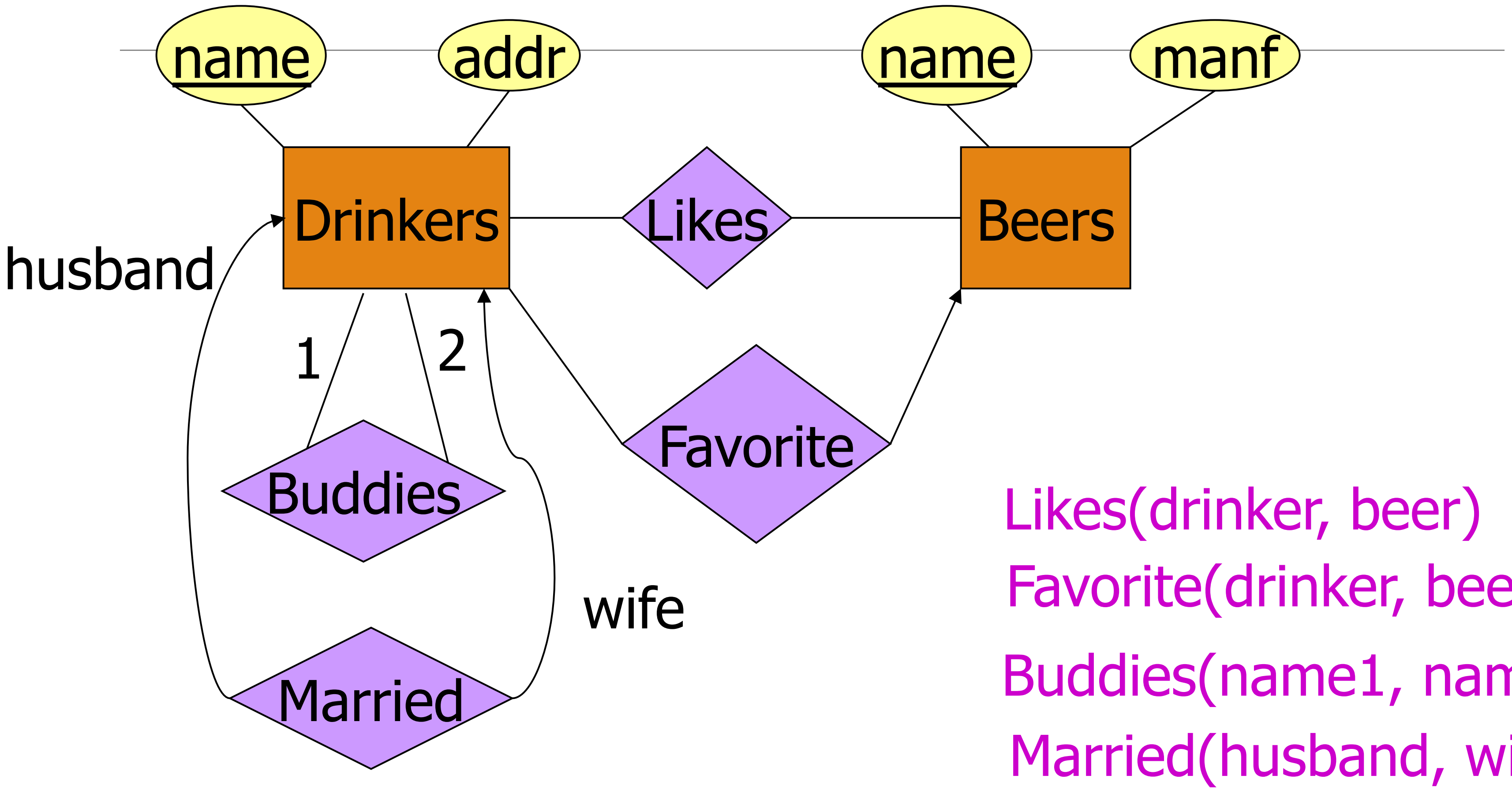
- ✓ Entity set -> relation.
  - Attributes -> attributes.
- ✓ Relationships -> relations whose attributes are only:
  - The keys of the connected entity sets.
  - Attributes of the relationship itself.

Entity Set -> Relation



Relation: Beers(name, manf)

Relationship -> Relation



Likes(drinker, beer)

Favorite(drinker, beer)

Buddies(name1, name2)

Married(husband, wife)

Further readings

- ✓ http://archive.mu.ac.in/myweb_test/syllFybscit/DBMS.pdf