DBMS

Dr. Media A. Ibrahim
Assist.Lec: Areen Hamad
Software Engineering
ISE Department
Media.ibrahim@epu.edu.iq
Areen.hamad@epu.edu.iq

2024-2025

Fundamentals of Database Concepts Database Models.

Outlines

- Introduction to Data Modeling.
- The Entity-Relationship Model.
- Attributes in the E-R Model.
- Relationships in the E-R Model.
- Mapping Cardinality.
- Keys of an Entity Set.
- Primary Keys, Super Keys and Candidate Keys.
- Entity Sets vs. Attributes.
- Weak Entity Sets vs. Strong Entity Sets.
- Multiway Relationships.
- Generalization, Specialization and Aggregation in ER Model

Modeling

- A **database model** is a type of data **model** that determines the logical structure of a **database** and fundamentally determines in which manner data can be stored, organized, and manipulated. The most popular example of a **database model** is the relational **model**, which uses a table-based format.

Instances and Schemas

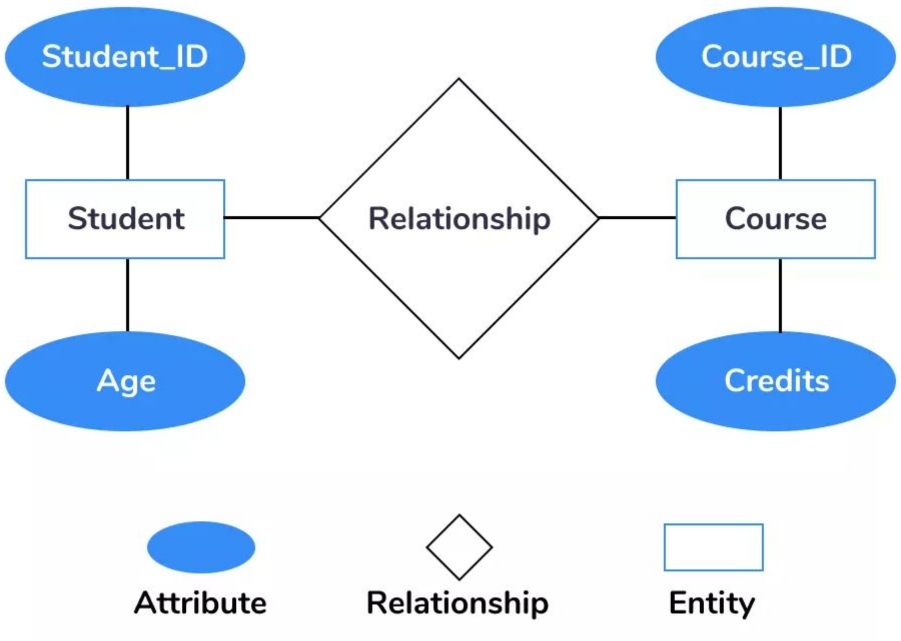
- Similar to types and variables in programming languages
  - **Schema** – the logical structure of the database
    - Example: The database consists of information about a set of customers and accounts and the relationship between them
    - Analogous to type information of a variable in a program
    - **Physical schema**: database design at the physical level
    - **Logical schema**: database design
    - the logical level
  - **Instance** – the actual content of the database at a particular point in time
    - Analogous to the value of a variable
  - **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
    - Applications depend on the logical schema
    - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Example: University Database

- **Conceptual schema:**
  - *Students(sid: string, name: string, login: string, age: integer, gpa:real)*
  - *Courses(cid: string, cname:string, credits:integer)*
  - *Enrolled(sid:string, cid:string, grade:string)*
- **Physical schema:**
  - Relations stored as unordered files.
  - Index on first column of Students.
- **External Schema (View):**
  - *Course_info(cid:string, enrollment:integer)*
  - *CS542Students(sid: string, grade:string)*

The Entity-Relationship Model

- An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modelling technique that can help define business processes and can be used as the foundation for a relational database.



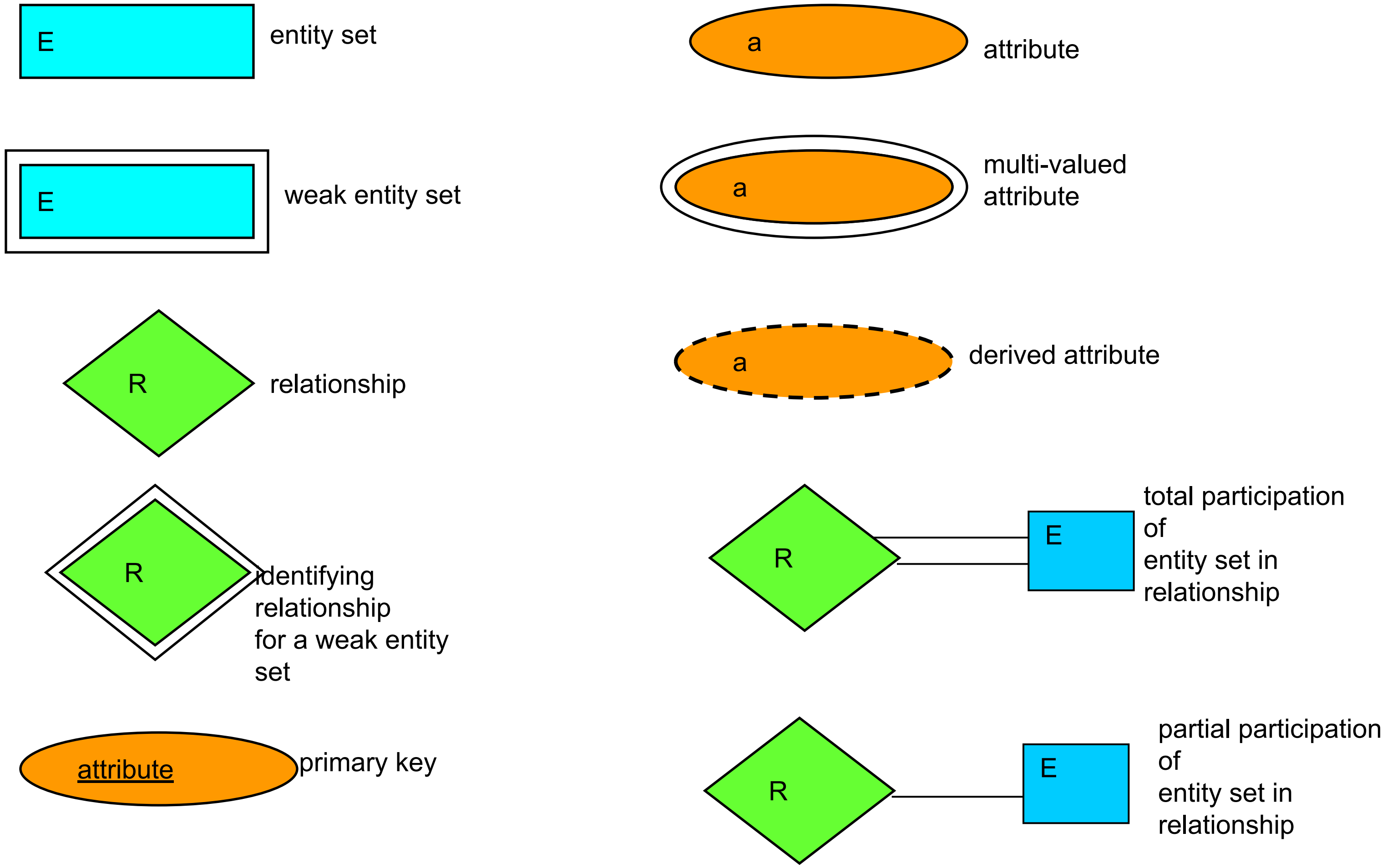
Creating an ER Model

1. Identify entities
2. Identify entity attributes and primary keys
3. Specify relationships

So the basics of ER modeling is :

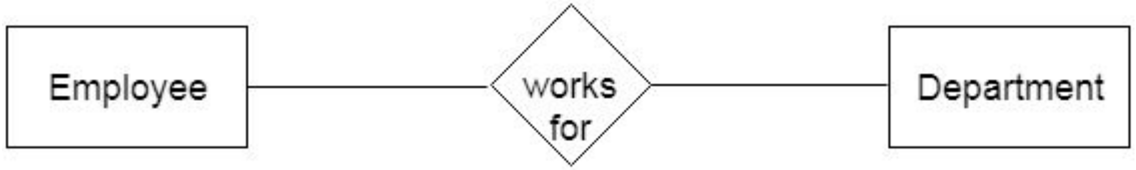
- Entities
- Relationships
- Attributes

E-R Model Notation



Entities

- Entity - distinguishable “thing” in the real world
  - Strong (or regular) entity - entities have an independent existence (e.g. staff)
  - Weak entity - existence dependent on some other entity (e.g. next of kin)
- *Entity set* = collection of similar entities.
  - Similar to a class in object-oriented language



Weak Entity Sets

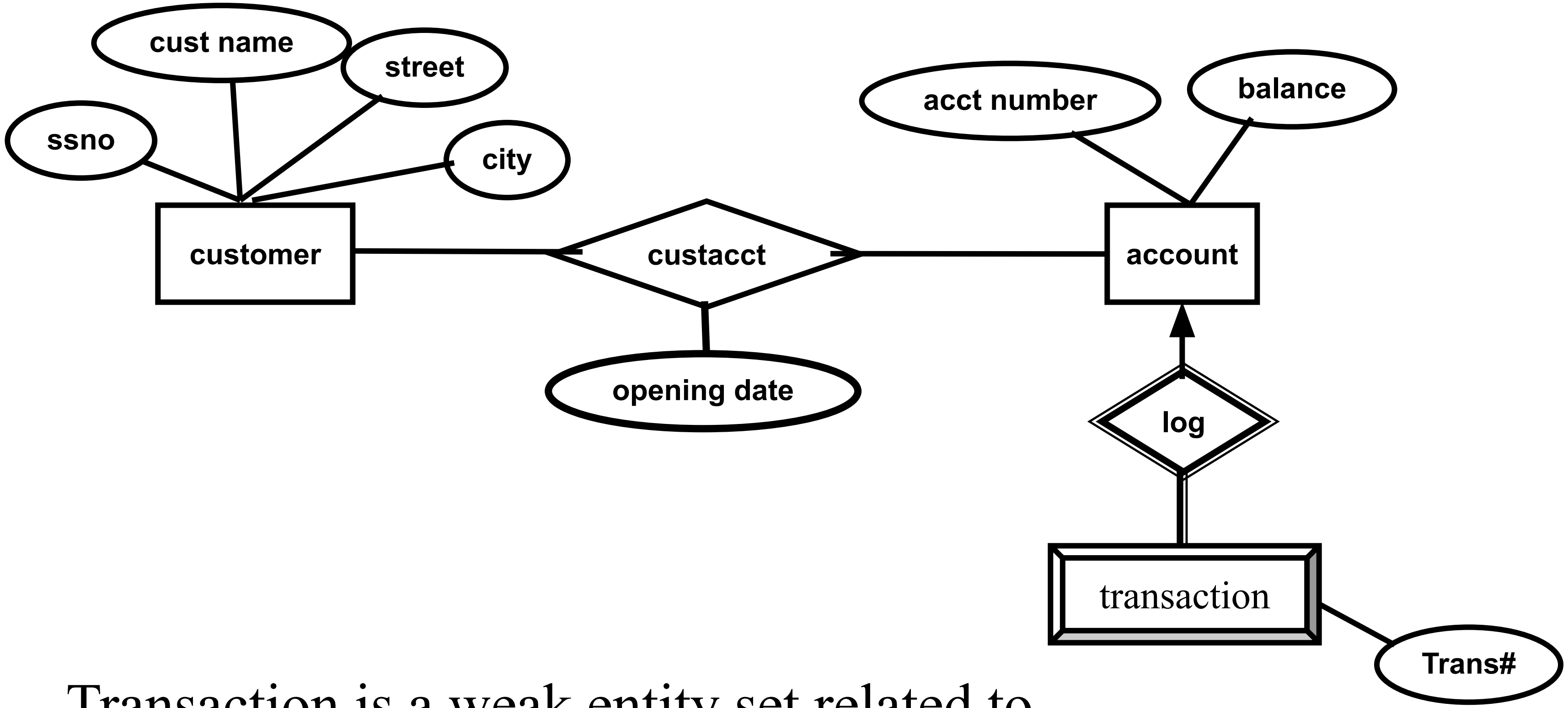
- Entity sets that do not have sufficient attributes to form a key are called weak entity sets.

- A weak entity set totally depends upon (one or more) strong entity sets via a one-to-many relationship from whom they derive their key

- A weak entity set may have a discriminator (or a partial key) that distinguish between weak entities related to the same strong entity.

- key of weak entity set = Key of owner entity set(s) + discriminator

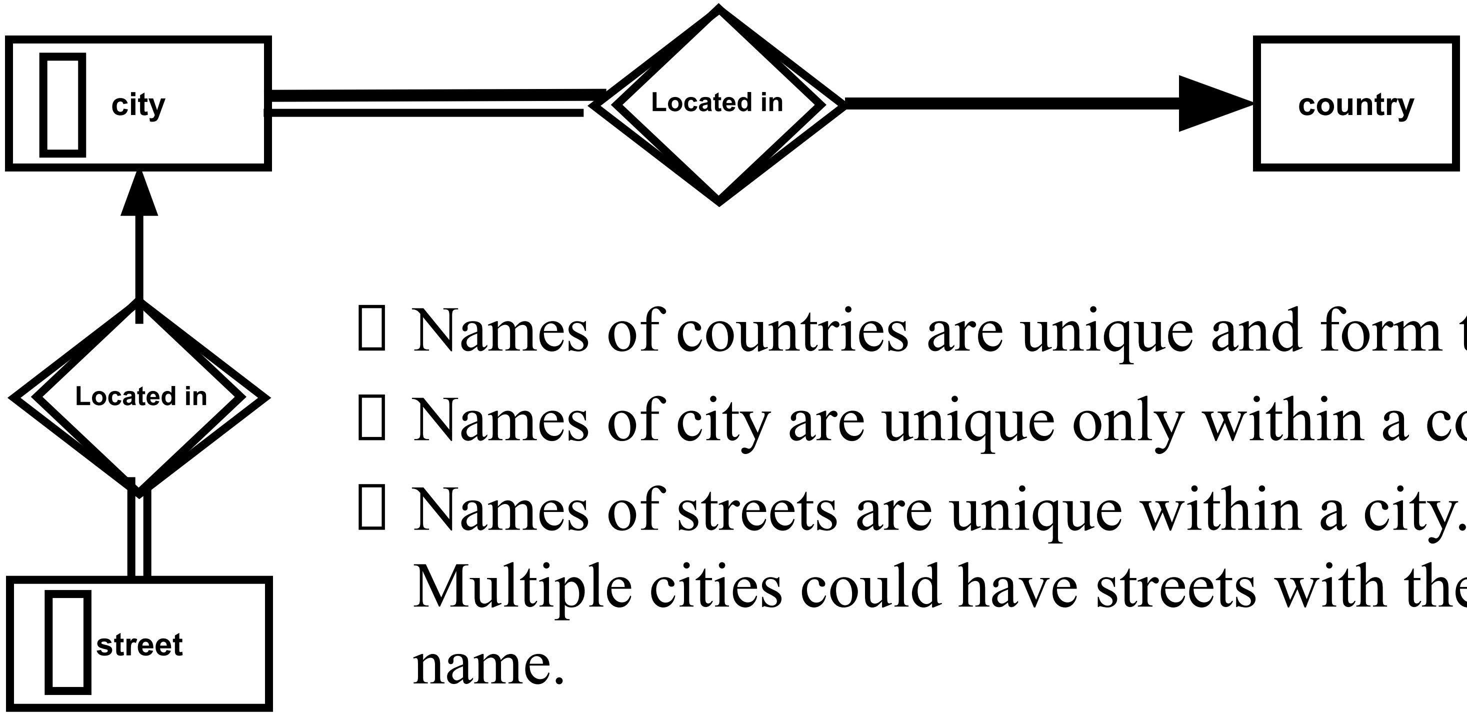
Weak Entity Sets (cont.)



– Transaction is a weak entity set related to accounts via log relationship.

– Trans# distinguish different transactions on same account

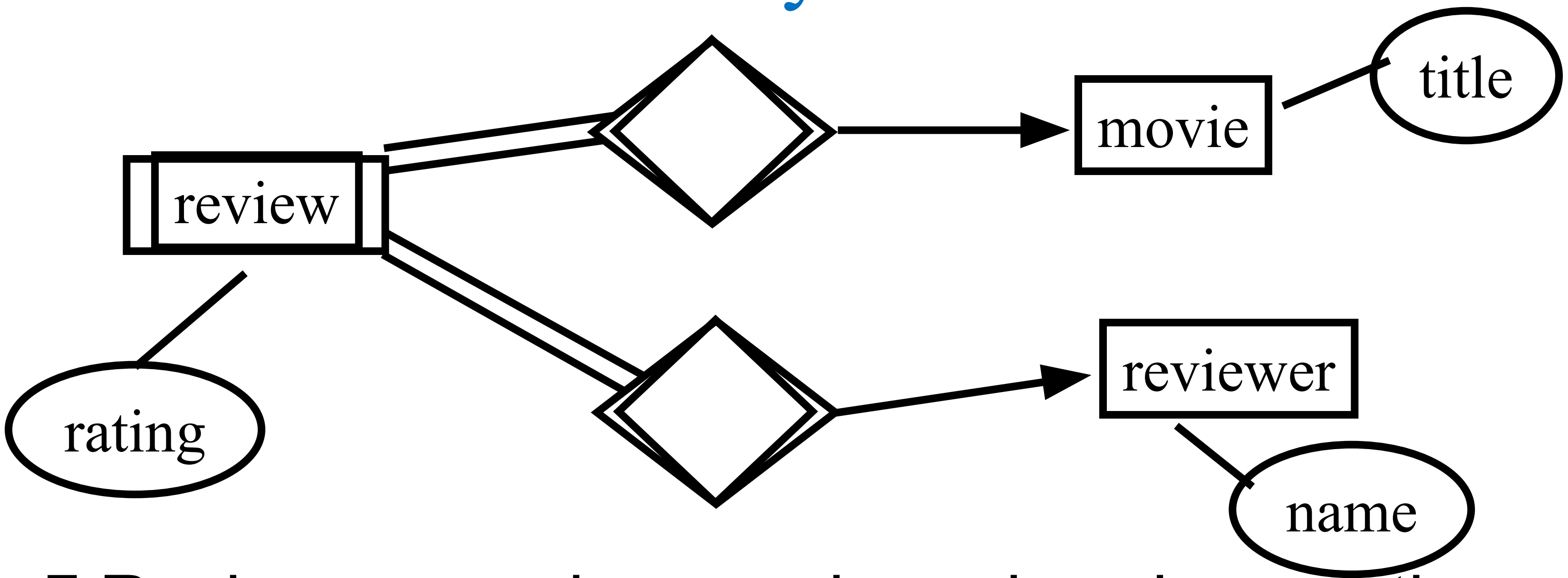
A Chain of Weak Entity Sets



- Names of countries are unique and form the key.
- Names of city are unique only within a contry
- Names of streets are unique within a city. Multiple cities could have streets with the same name.

Example illustrating that a weak entity set might itself participate as owner in an identifying relationship with another weak entity set.

A Weak Entity Set with Multiple Owner Entity Sets

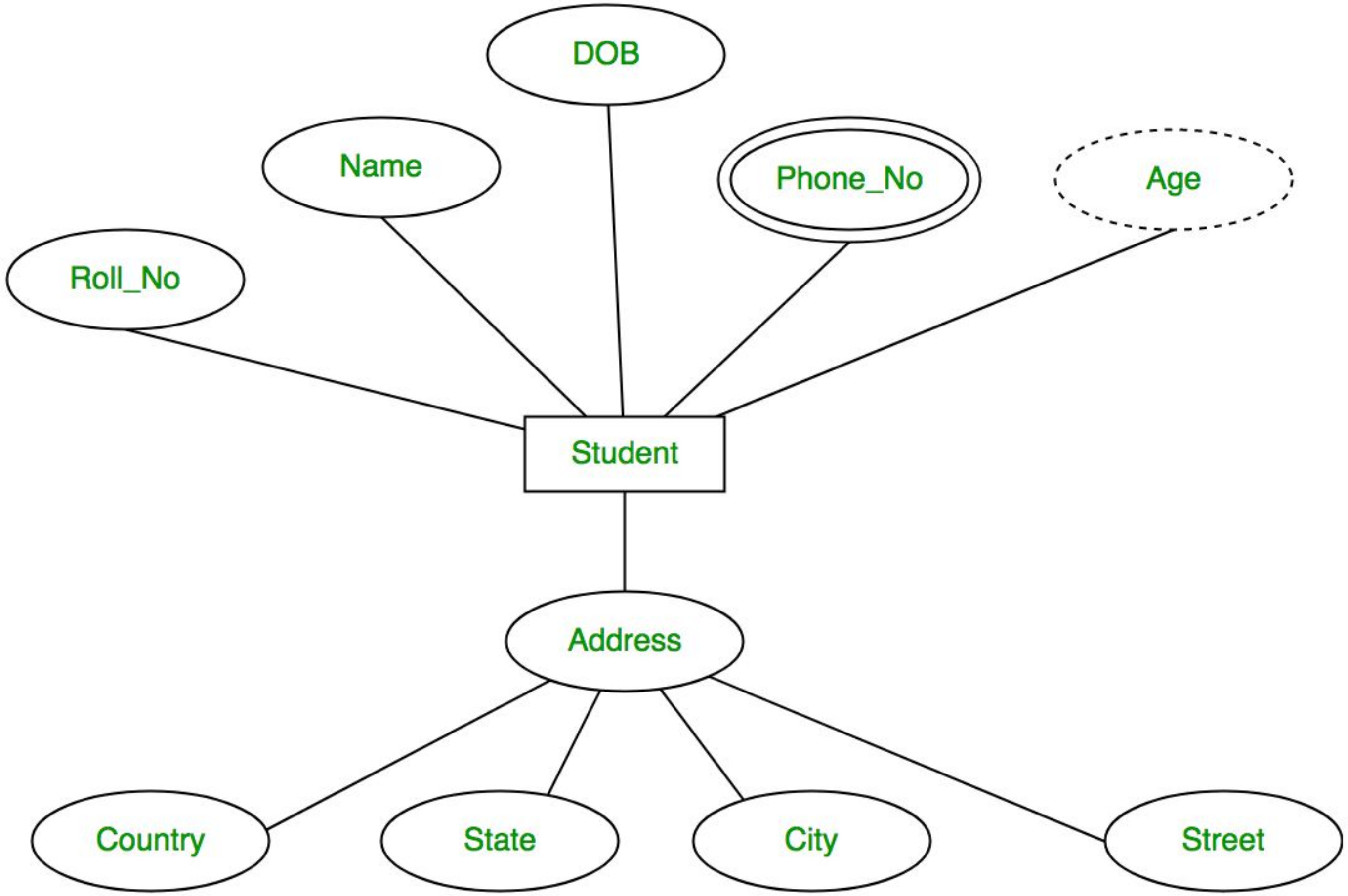


- Reviewers review movie and assign a rating
- Review is a weak entity set whose owner sets correspond to both the movie and the reviewer entity sets.
- Key for the review entity set = key of movie + key of reviewer

Attributes

- Entity types have ***Attributes*** (or properties) which associate each entity with a value from a *domain* of values for that attribute
- Attributes can be
  - simple (atomic) e.g. Surname; date of birth
  - composite e.g. address (street, town, postcode)
  - multi-valued e.g. phone number
  - base or derived e.g. D.O.B. ; age
  - key

Entity type Student with its attributes can be represented as:



Relationships

- A relationship is
  "... An association among entities (the participants).."
- Relationships link entities with each other
- ***Relationship Set***: Collection of similar relationships.
  - Same entity set could participate in different relationship sets, or in different "roles" in same set.

Relationships: constraints

- The ***degree*** of a relationship type
  - binary (connects 2 entity types)
  - unary/ recursive (connects 1 entity type with itself)
  - complex (connects 3 or more entity types)
    - Ternary (connects 3)

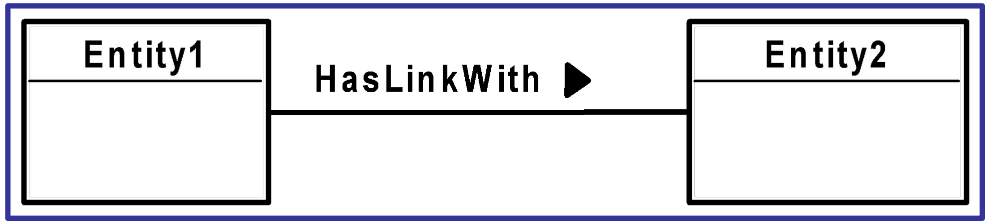
Degree

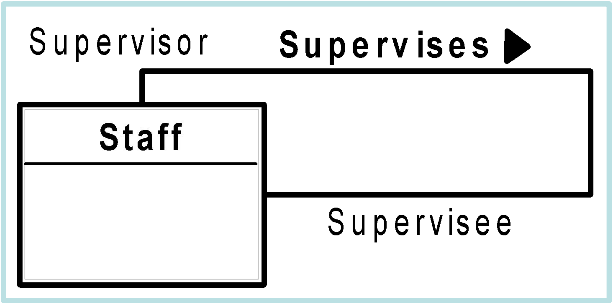
- Relationship constraints - ***cardinality***
  - one to one (1:1)
  - one to many (1:m)
  - many to many (m:n)
- Relationship constraints – ***participation***
  - full/mandatory
  - or partial/optional

Multiplicity

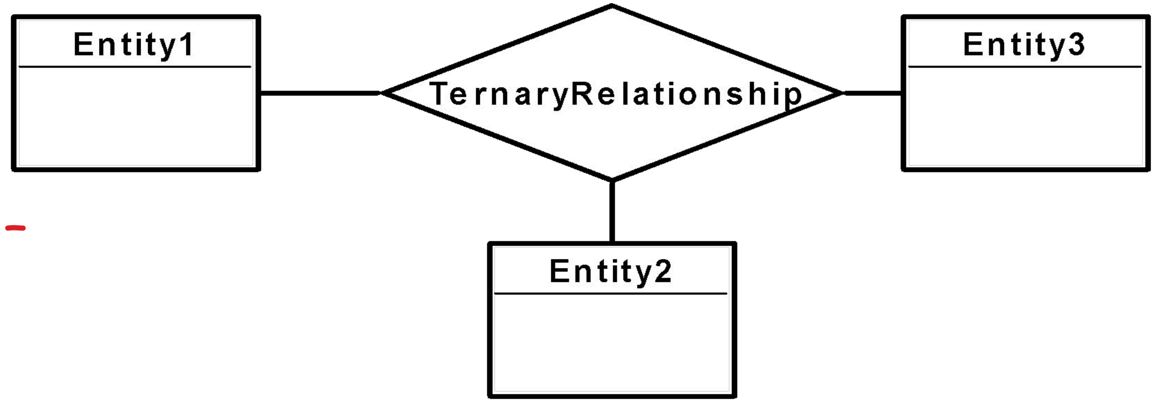
Relationships: Degree

Binary relationship





Recursive (Unary) relationship - example



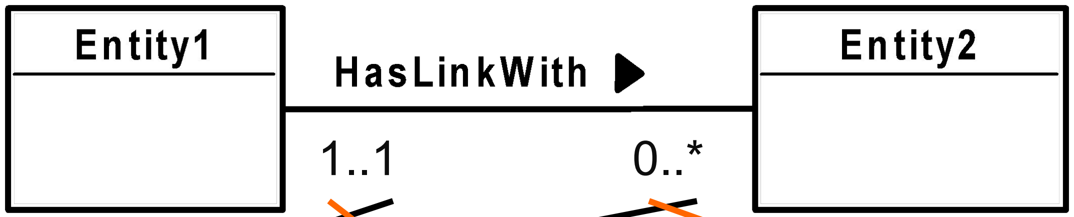
Complex relationship – here ternary

Relationships: Multiplicity

- label lines to show cardinality and participation
  - 0..1 “zero or one”
  - 0..* “zero or more”
  - 1..1 “one”
  - 1..4 “between 1 and 4”
  - 1..* “one or more”

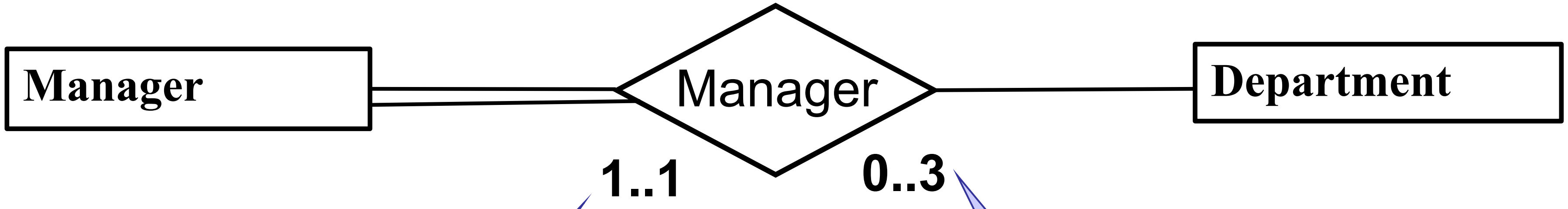
optional

mandatory



Entity1 has a 1:m relationship with Entity2;
participation for Entity2 is mandatory, for Entity1 optional.

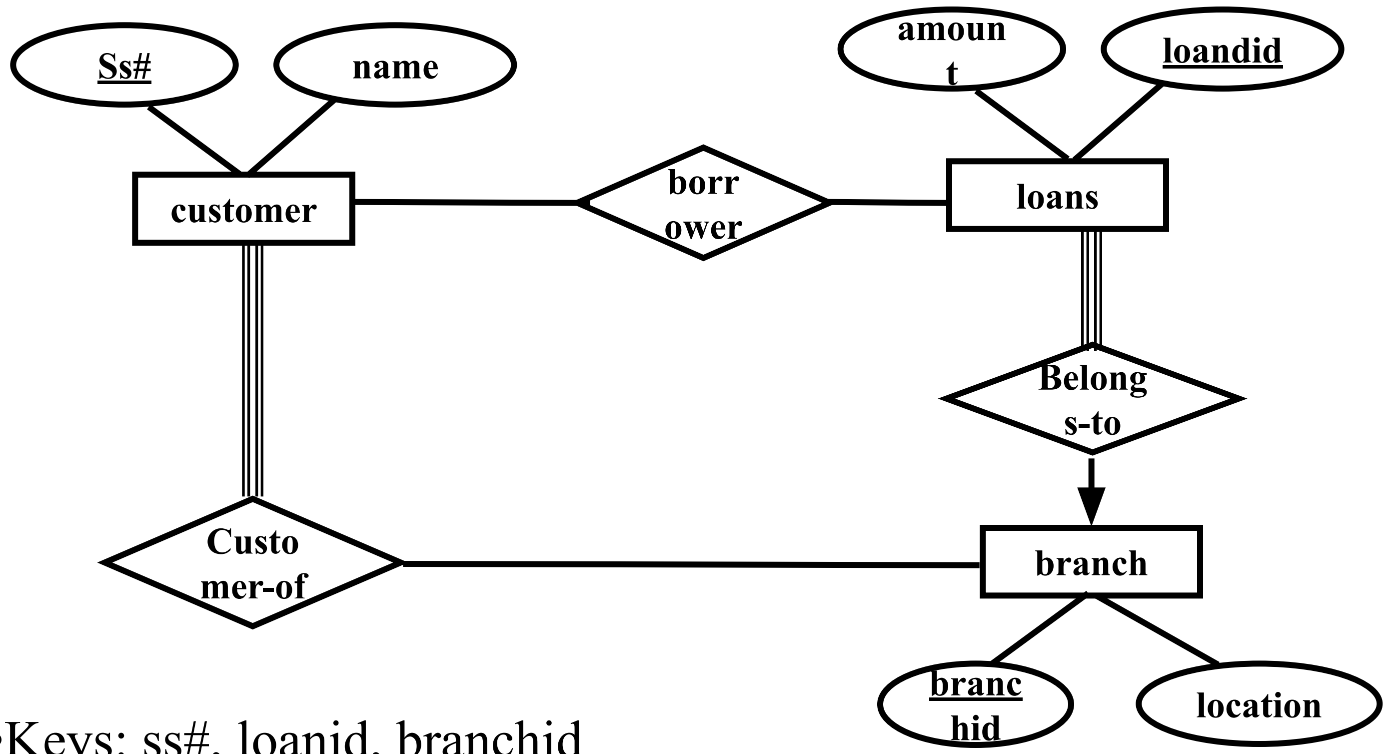
Relationships example



Each department is managed by ONE manager

Each manager manages UP TO 3 departments (but need not manage any department)

Example



•Keys: ss#, loanid, branchid

•Cardinality constraint: each loan belongs to a single branch

•Participation constraints:

•Each customer must be a customer of at least one branch

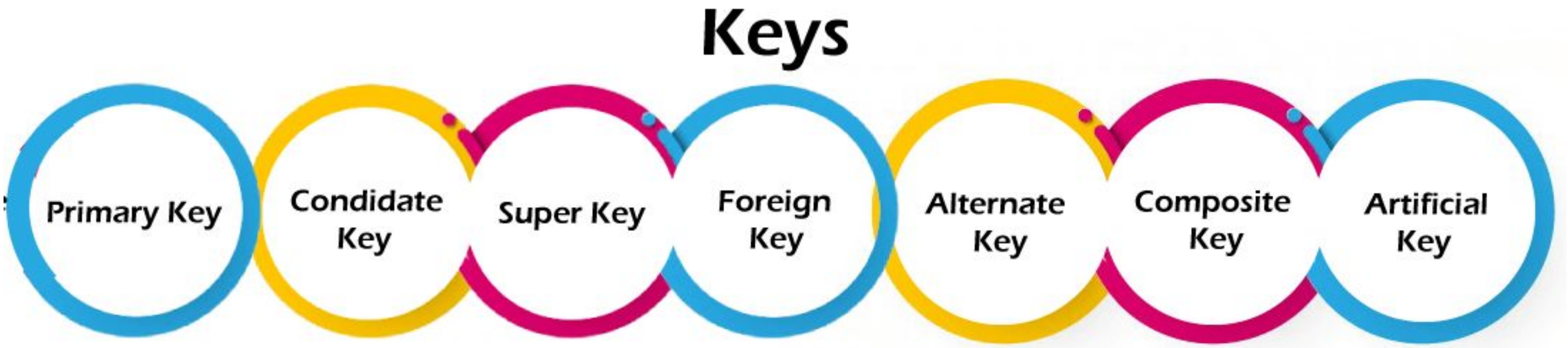
•Each loan must belong to some branch

Entity vs. Attribute

- Should *address* be an attribute of Employees or an entity (connected to Employees by a relationship)?
- Depends upon the use we want to make of address information, and the semantics of the data:
    - If we have several addresses per employee, *address* must be an entity (since attributes cannot be set-valued).
    - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, *address* must be modeled as an entity (since attribute values are atomic).

DBMS Keys

- *Key and key attributes:*
  - Key: a unique value for an entity
  - Key attributes: a group of one or more attributes that uniquely identify an entity in the entity set
  - Keys play an important role in the relational database.
  - It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.
- *Types of keys:*

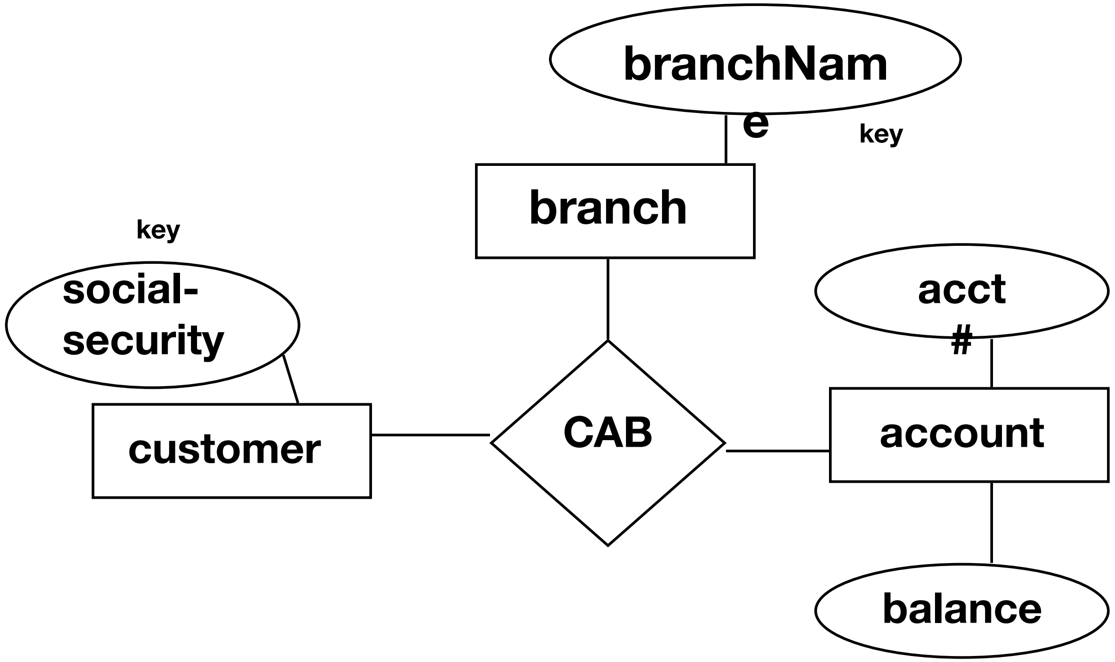


DBMS Keys

- *Super key, candidate key, primary key and Foreign key*
  - Super key: a set of attributes that allows to identify and entity uniquely in the entity set
  - Candidate key: minimal super key
    - There can be many candidate keys
  - Primary key: a candidate key chosen by the designer
    - Denoted by underlining in ER attributes
  - Foreign key:
    - Foreign keys are the column of the table used to point to the primary key of another table

Multiway Relationships

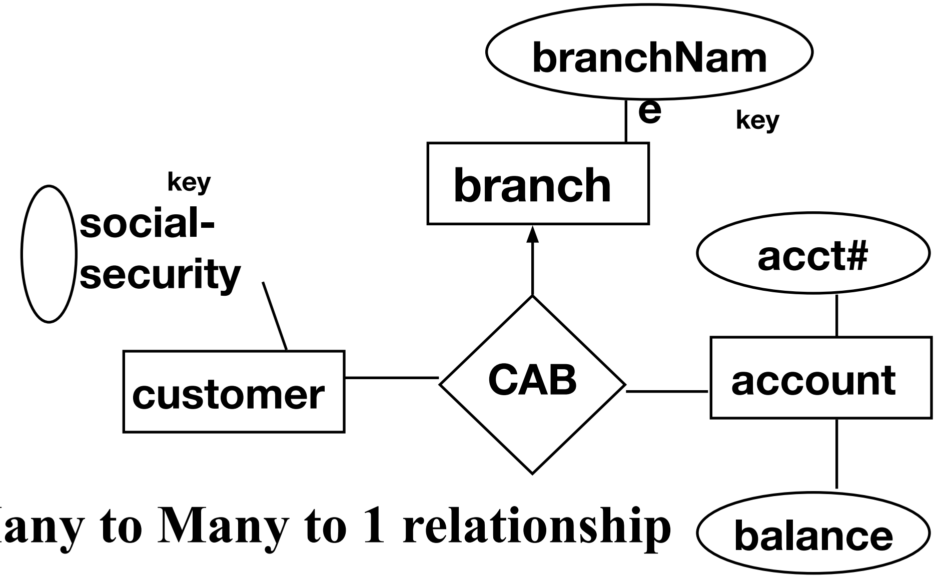
- Usually binary relationships (connecting two E.S.) suffice.
- However, there are some cases where three or more E.S. must be connected by one relationship.
- Similar to binary relationship, cardinality and participation constraints defined over multiway relationships



Customer	Account	Branch
John	1001	Irvine
Megan	1001	LA
Megan	2001	Tokyo

CAB Relationship Set

Cardinality Constraint over Multiway Relationships (cont.)



Many to Many to 1 relationship

– Interpretation:
  - Each pair of customer and account determine the branch (that is, have a single branch related to them).

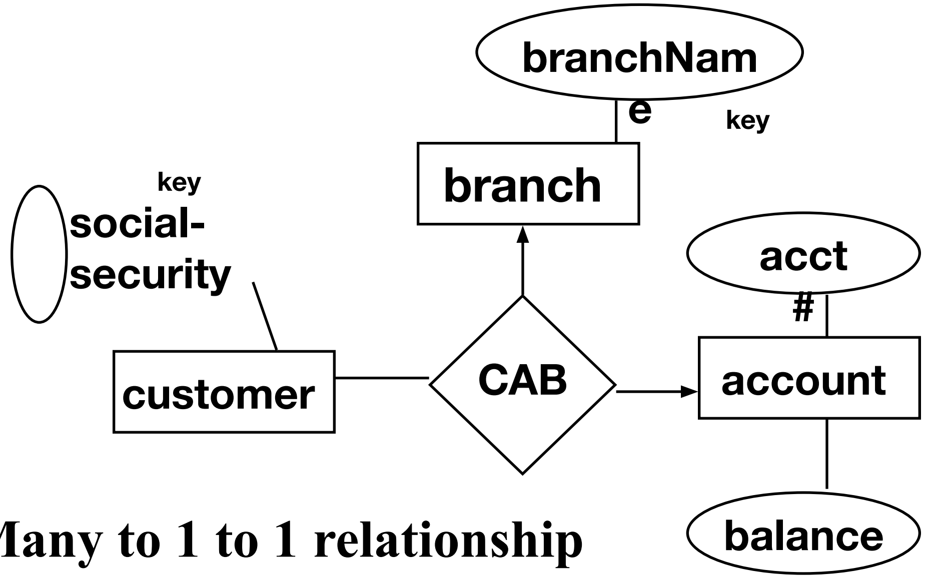
Customer	Account	branch
John	1001	Irvine
Megan	1001	Dallas
Megan	1002	Tokyo
Megan	1001	Tokyo

Illegal: Megan has account 1001 at 2 branches

Customer	Account	branch
John	1001	Irvine
Megan	1001	Dallas
Megan	1002	Tokyo
Megan	1003	Tokyo

Legal

Cardinality Constraint over Multiway Relationships (cont.)



Many to 1 to 1 relationship

– Interpretation:
  - Each (customer, branch) pair related to a single account
  - Each (customer, account) pair related to a single branch

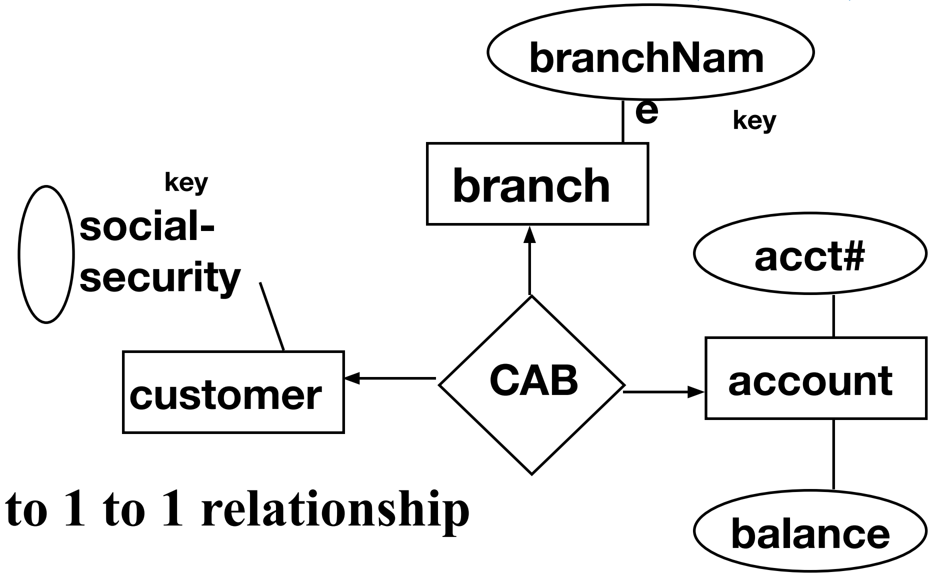
Customer	Account	branch
John	1001	Irvine
Megan	1001	Dallas
Megan	1002	Tokyo
Megan	1003	Tokyo

Illegal: Megan has 2 accounts in Tokyo Branch

Customer	Account	branch
John	1001	Irvine
Megan	1001	Dallas
Megan	1002	Tokyo
John	1002	Tokyo

Legal

Cardinality Constraint over Multiway Relationships (cont.)



1 to 1 to 1 relationship

Customer	Account	branch
John	1001	Irvine
Megan	1001	Dallas
Megan	1002	Tokyo
John	1002	Tokyo

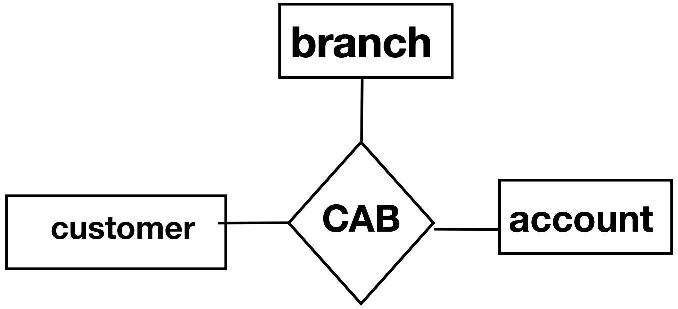
Illegal: Both John and Megan have account 1002 in Tokyo Branch

– Interpretation:
  - Each (customer, branch) related to a single account
  - Each (customer, account) pair related to a single branch
  - Each (branch, account) pair can have single customer

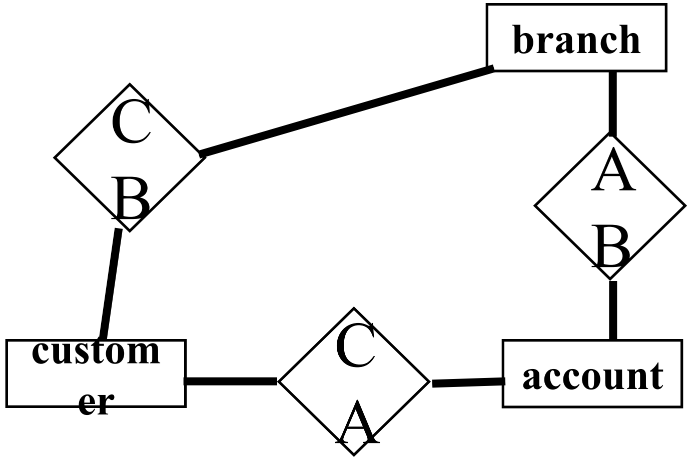
Customer	Account	branch
John	1001	Irvine
Megan	1001	Dallas
Megan	1002	Tokyo

Legal

Representing Ternary Relationship Using Binary Relationships

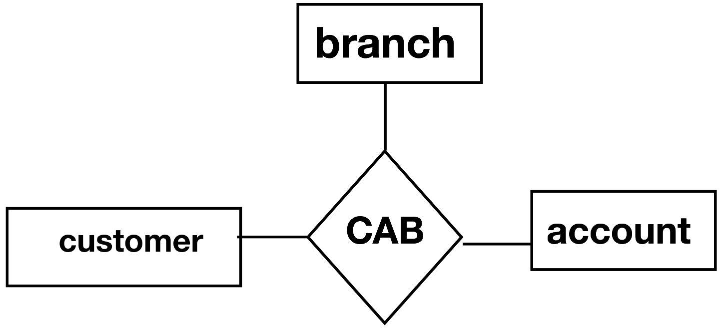


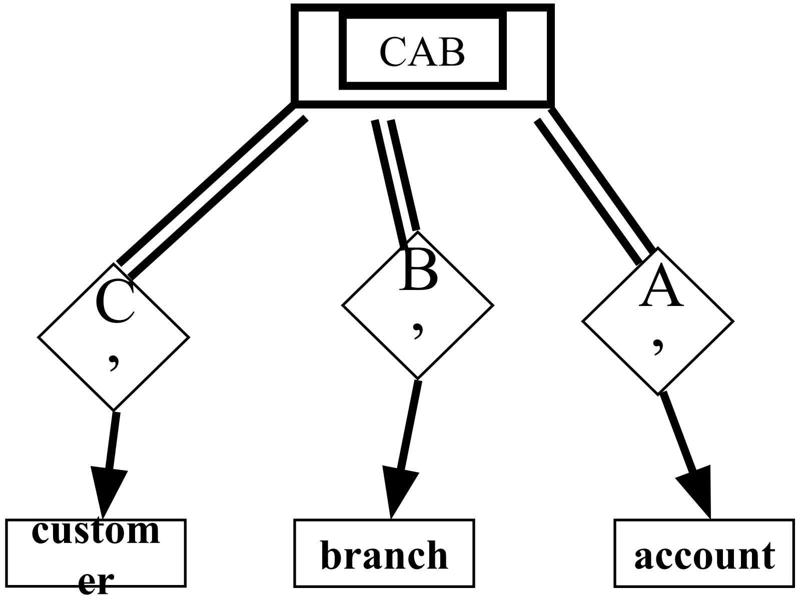
Customer	Account	branch
C2	A1	B1
C1	A2	B1
C1	A1	B2



- Above Schema using binary relationships does not correctly capture the information represented by the ternary relationship.

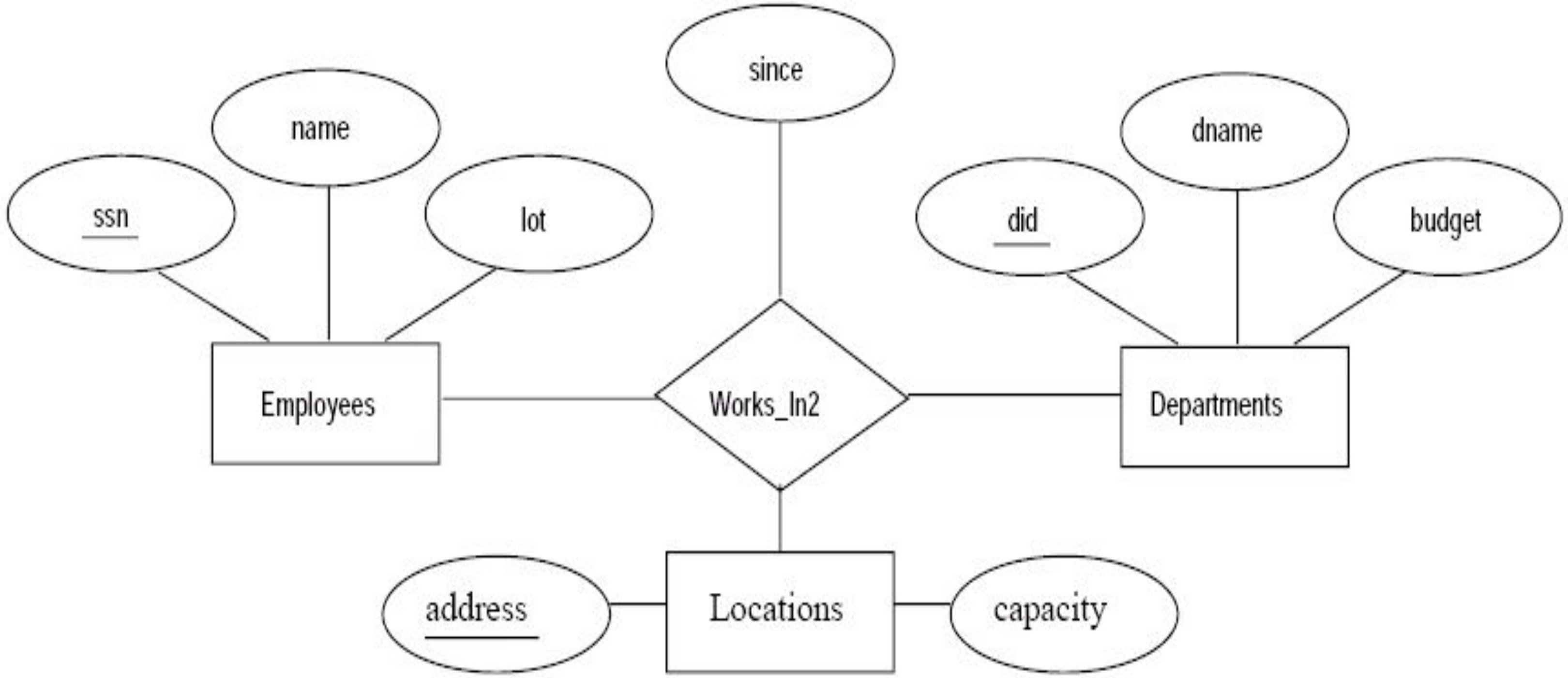
Representing Ternary Relationship Using Binary Relationships (cont.)





- The CAB relationship is represented as a weak entity set that depends upon the customer, branch and account entity sets.
- This schema using binary relationship fully captures the ternary relationship.

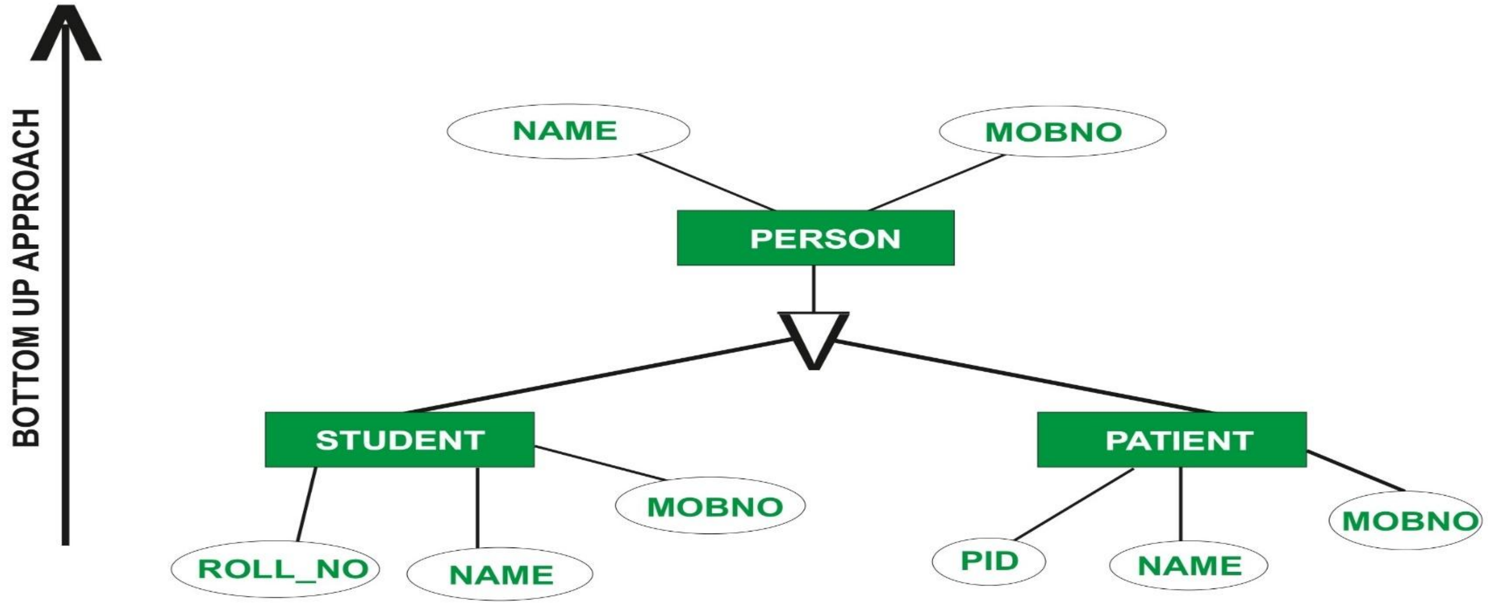
Multi-Way Relationships



Generalization, Specialization and Aggregation in ER Model

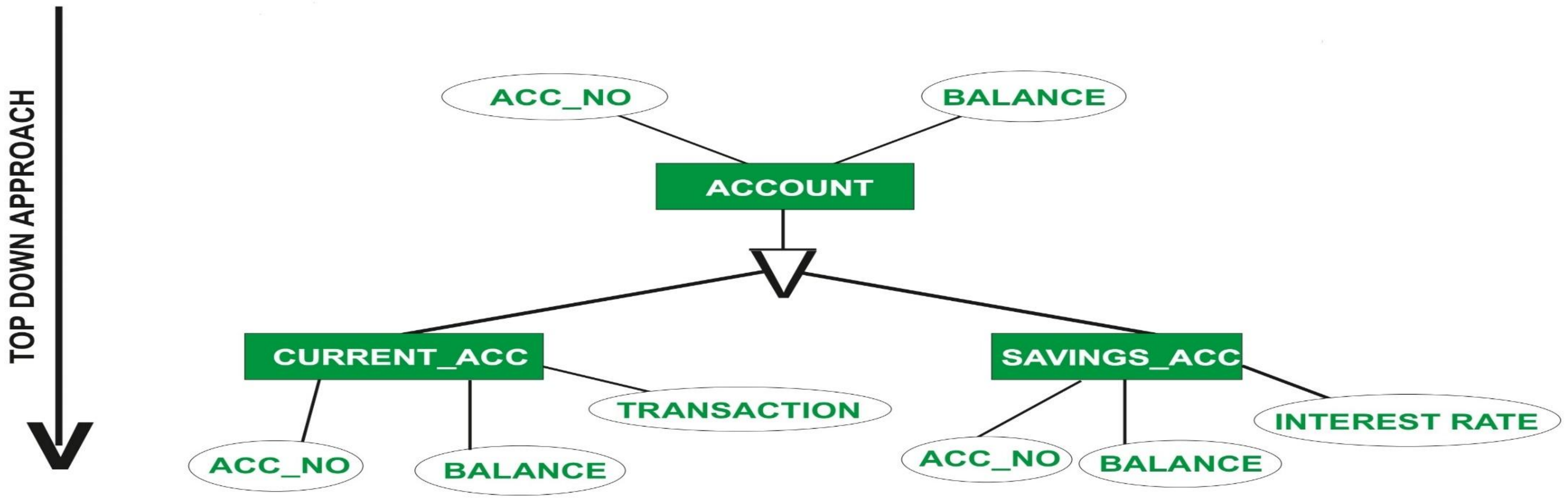
Generalization

- Generalization is like a bottom-up approach in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- Generalization is more like subclass and superclass system, but the only difference is the approach. Generalization uses the bottom-up approach.
- In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass



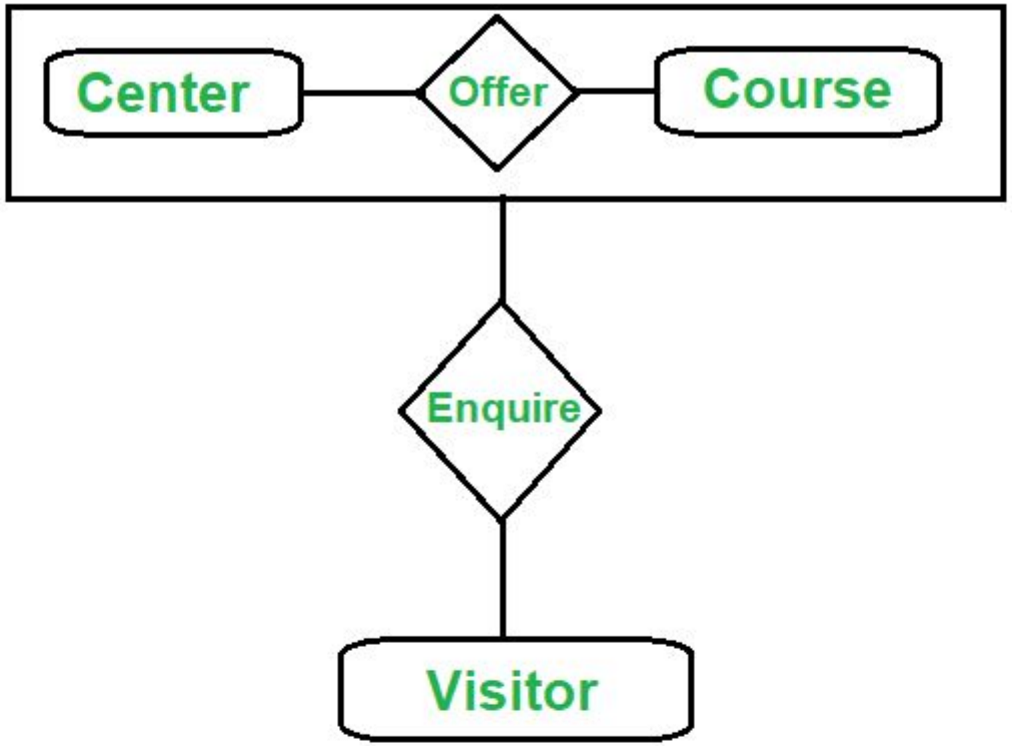
Specialization

- Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added.



Aggregation

- In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.



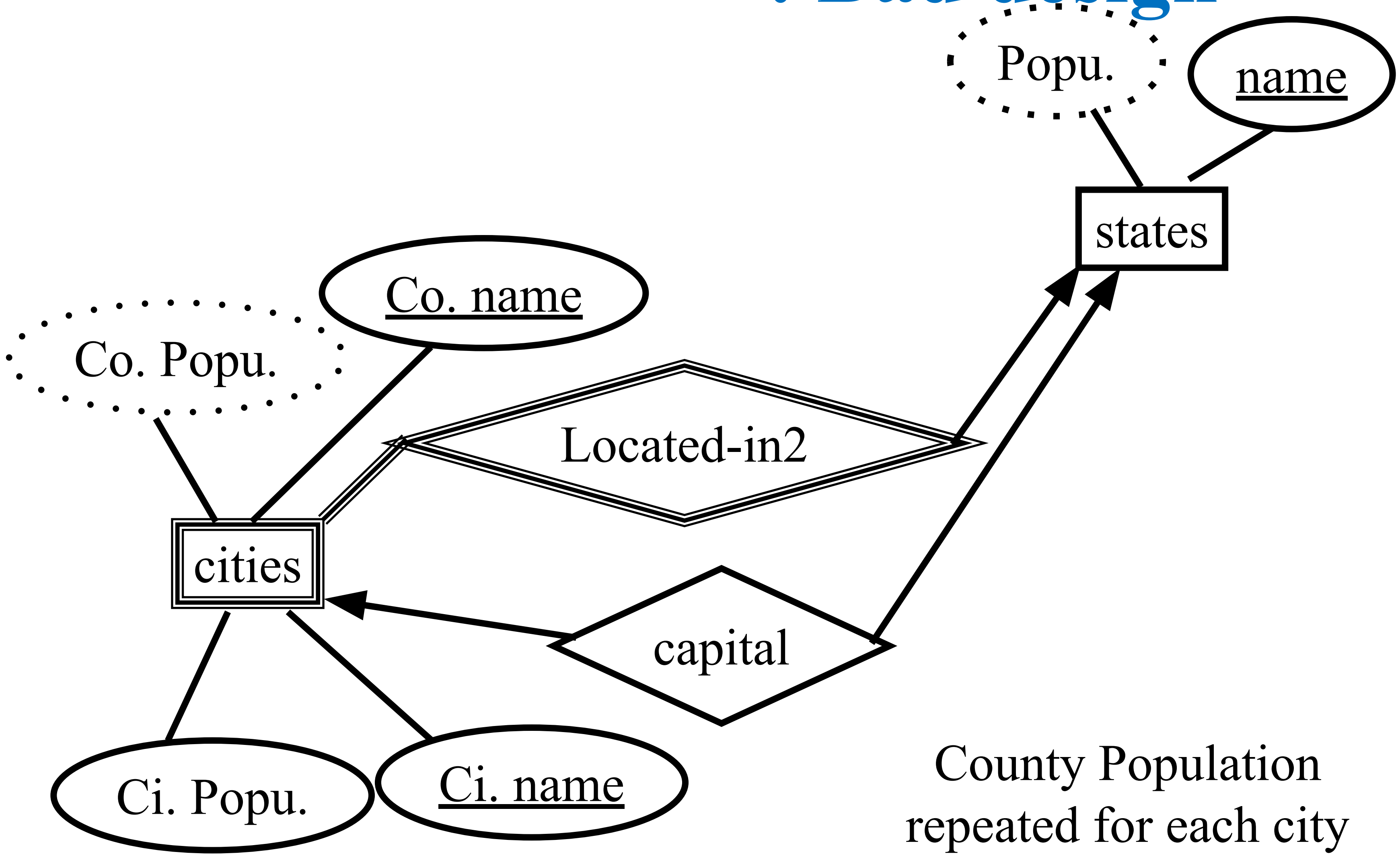
A Design Problem

- We wish to design a database representing cities, counties, and states in the US.
- For states, we wish to record the name, population, and state capital (which is a city).
- For counties, we wish to record the name, the population, and the state in which it is located.
- For cities, we wish to record the name, the population, the state in which it is located and the county in which it is located.

Uniqueness assumptions:

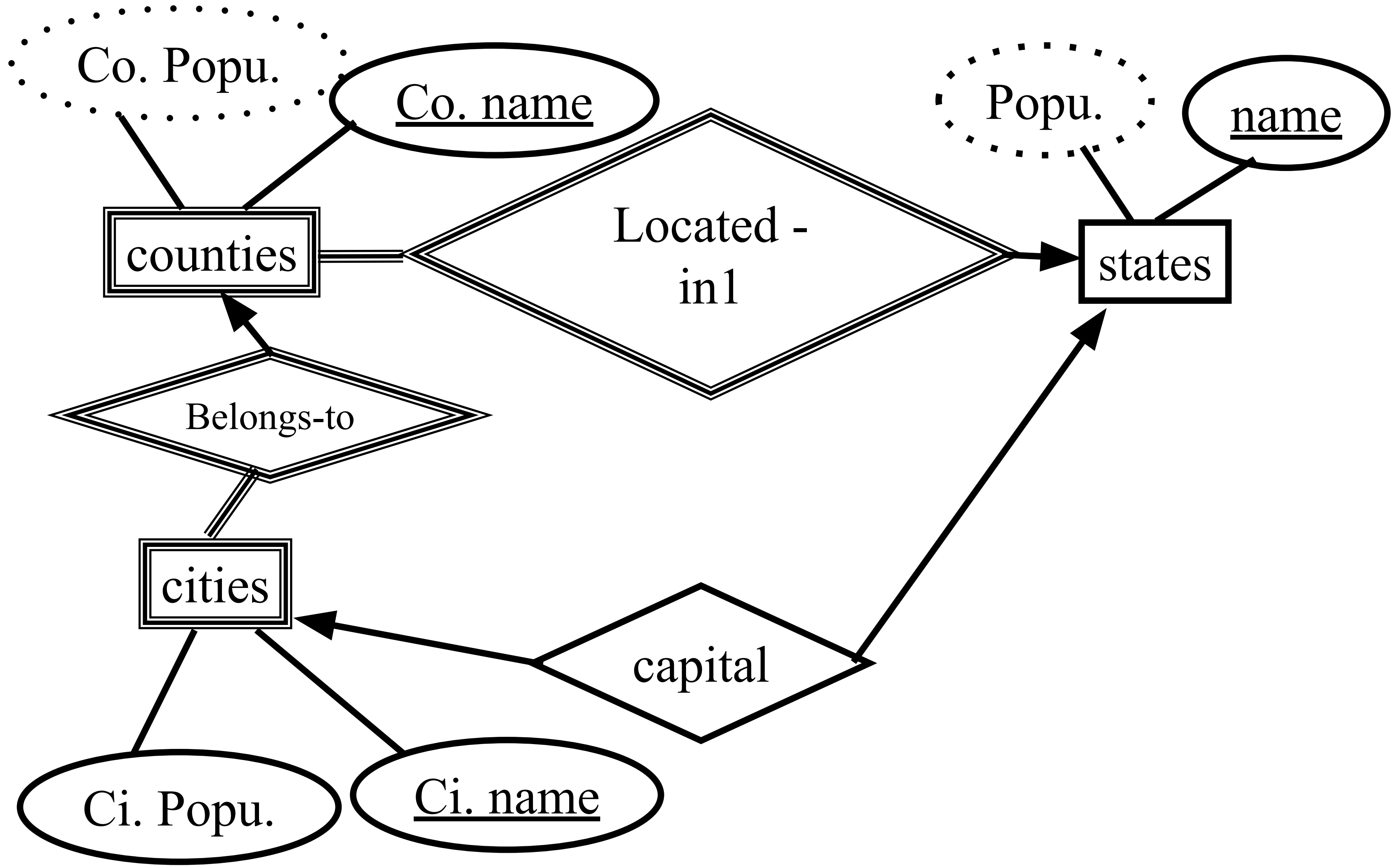
- Names of states are unique.
- Names of counties are only unique within a state (e.g., 26 states have Washington Counties).
- Cities are likewise unique only within a state (e.g., there are 24 Springfields among the 50 states).
- Some counties and cities have the same name, even within a state (example: San Francisco).
- All cities are located within a single county.

Design 1 : Bad design



County Population
repeated for each city

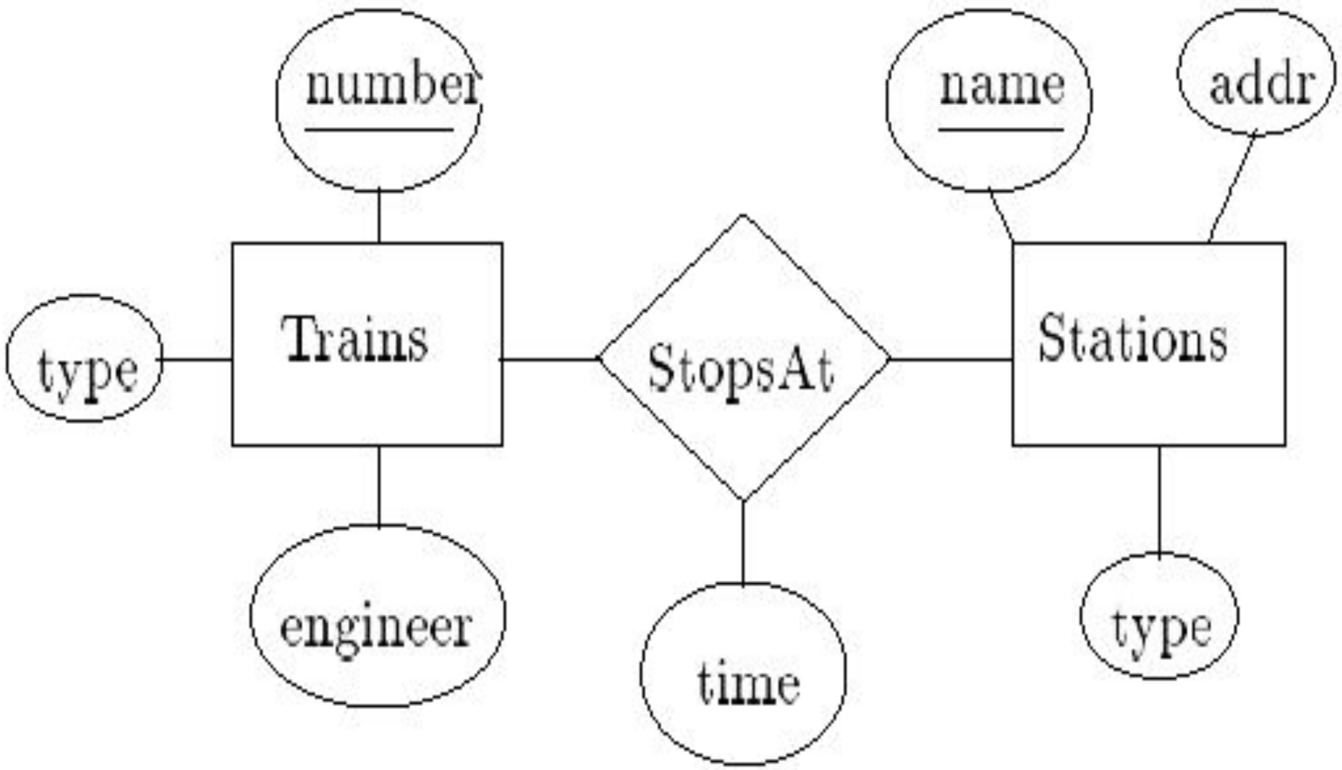
Design 2 -- good design



Another Design Problem

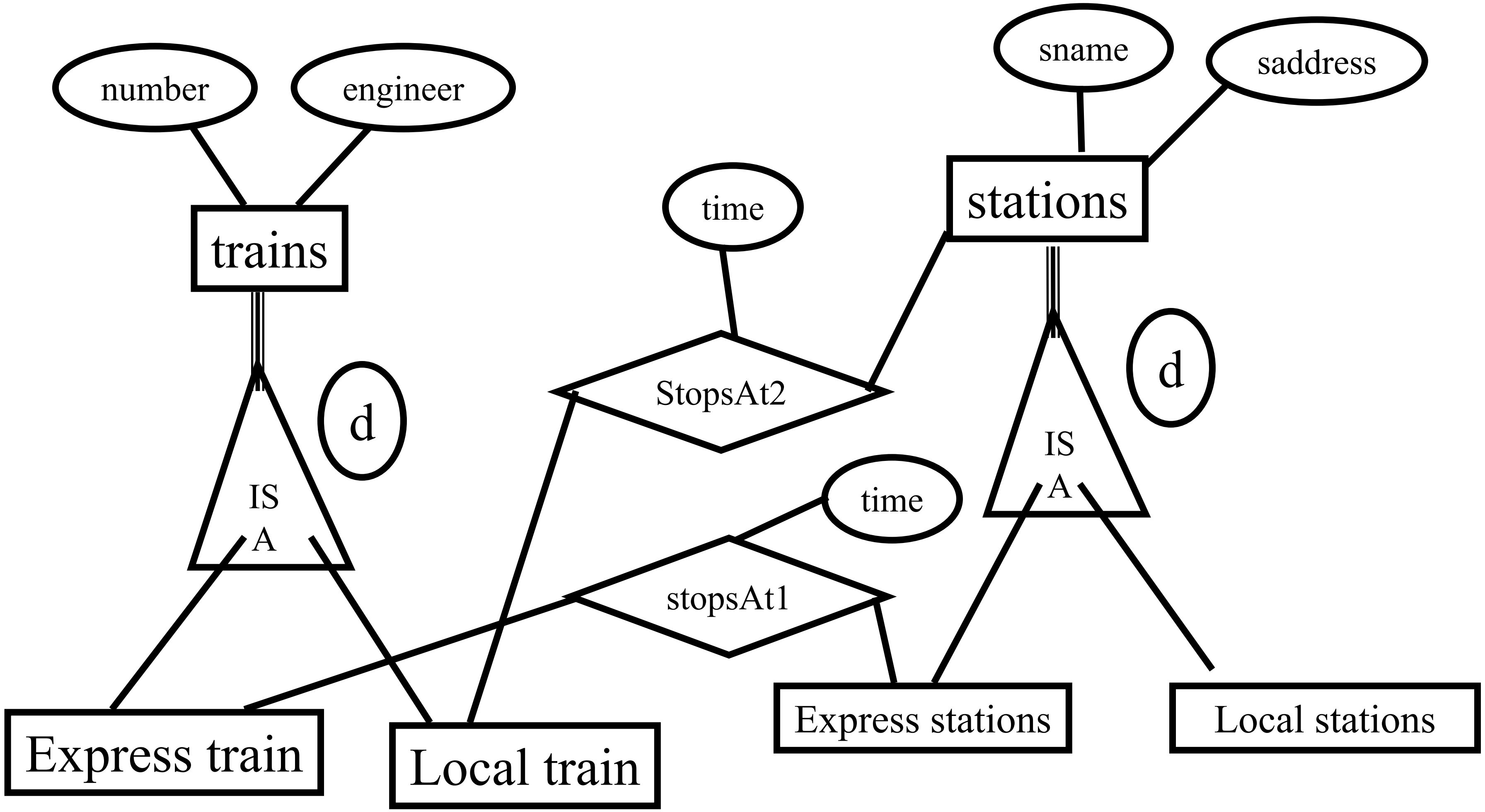
- We wish to design a database consistent with the following facts.
- Trains are either local trains or express trains, but never both.
- A train has a unique number and an engineer.
- Stations are either express stops or local stops, but never both.
- A station has a name (assumed unique) and an address.
- All local trains stop at all stations.
- Express trains stop only at express stations.
- For each train and each station the train stops at, there is a time.

Design 1: Bad design



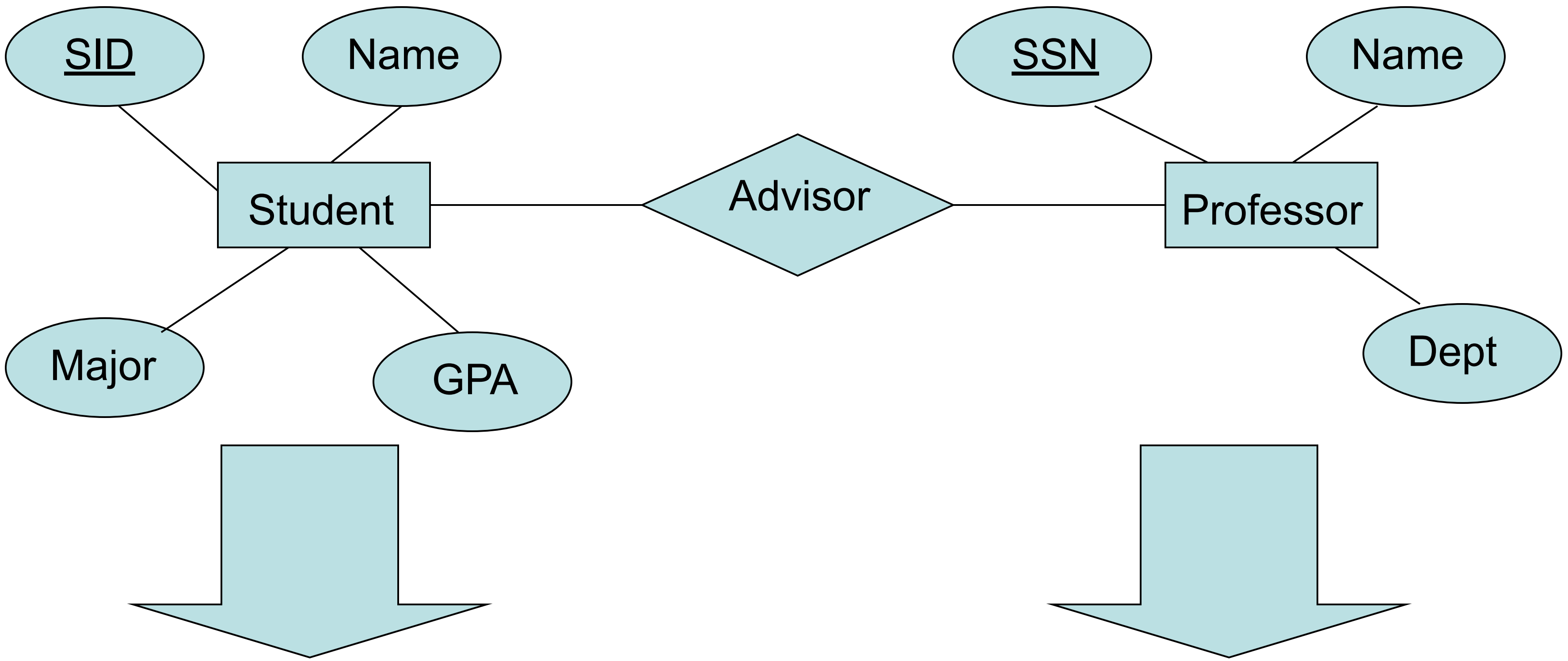
- Does not capture the constraints that express trains only stop only at express stations and local trains stop at all local stations

Design 2: Better Design



How to translate ER Model to Relational Model

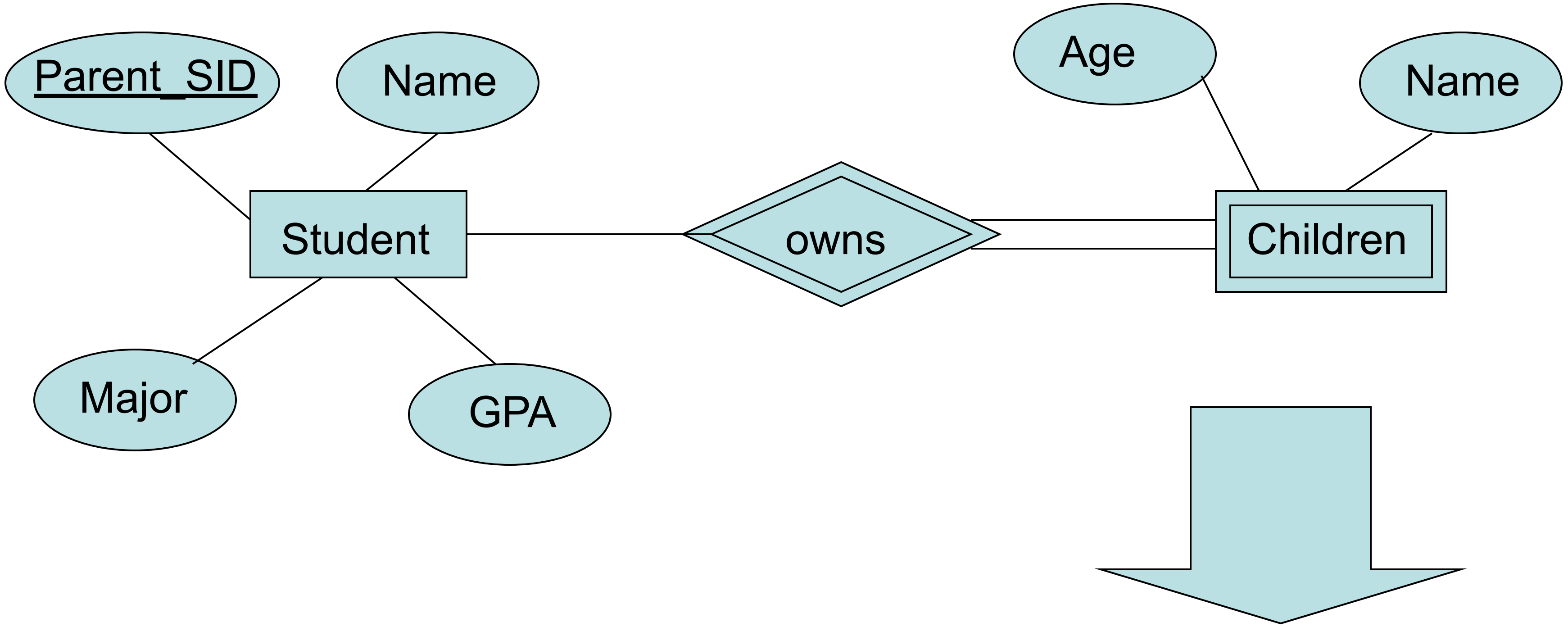
Example – Strong Entity Set



SID	Name	Major	GPA
1234	John	CS	2.8
5678	Mary	EE	3.6

SSN	Name	Dept
9999	Smith	Math
8888	Lee	CS

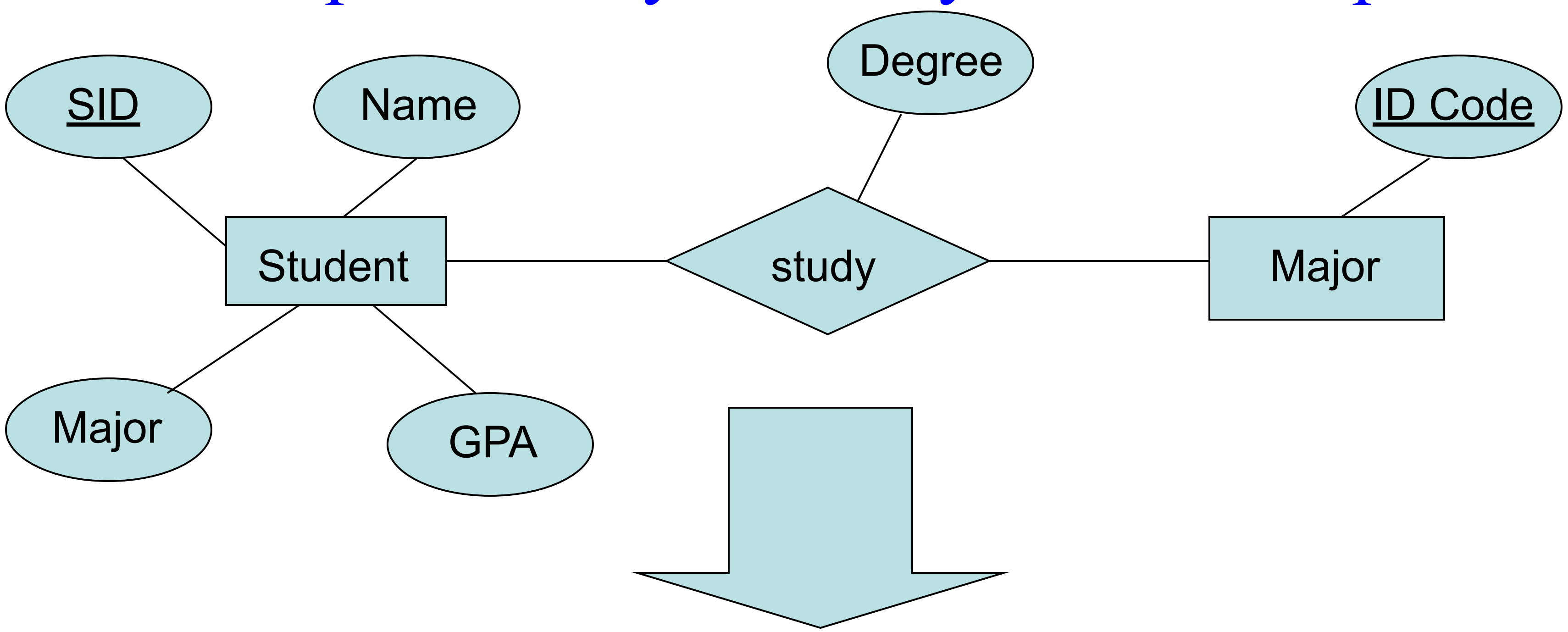
Example – Weak Entity Set



Age	Name	Parent_SID
10	Bart	1234
8	Lisa	5678

* Primary key of *Children* is *Parent_SID* + *Name*

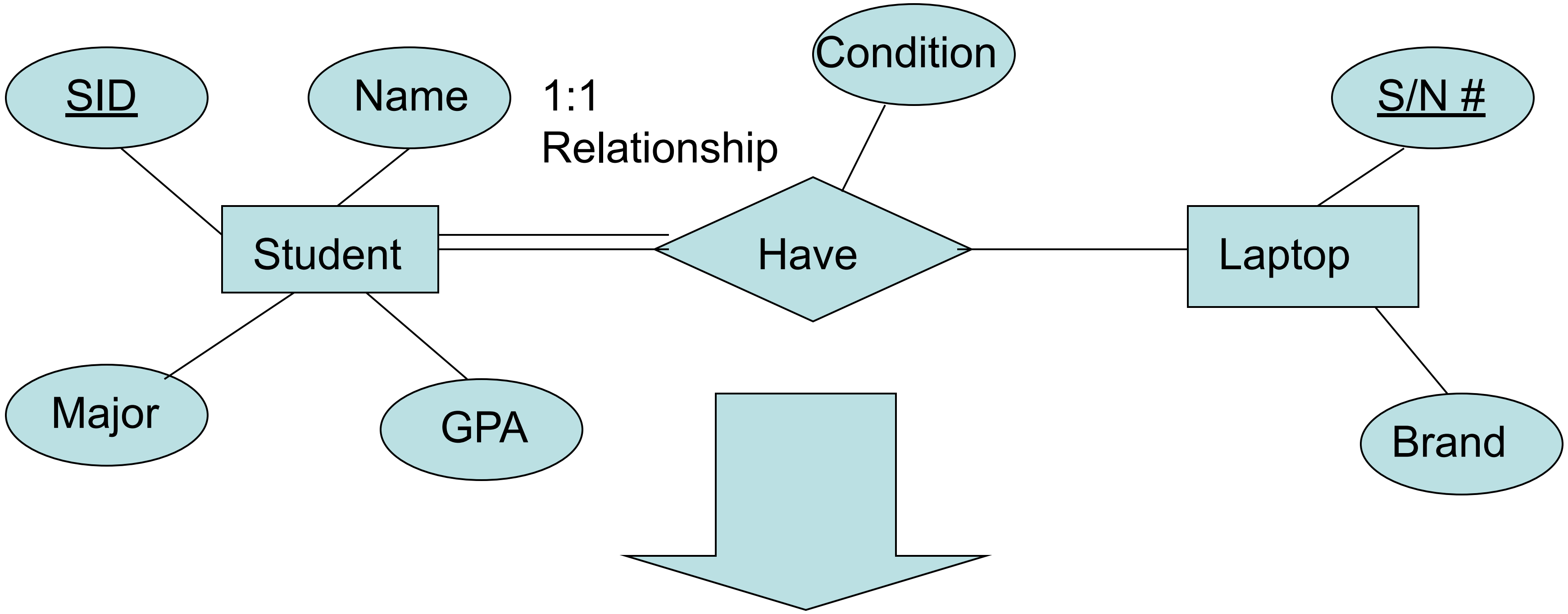
Example – Many-to-Many Relationship Set



SID	Maj_ID Co	S_Degree
9999	07	1234
8888	05	5678

* Primary key can be either *SID* or Maj_*ID_Co*

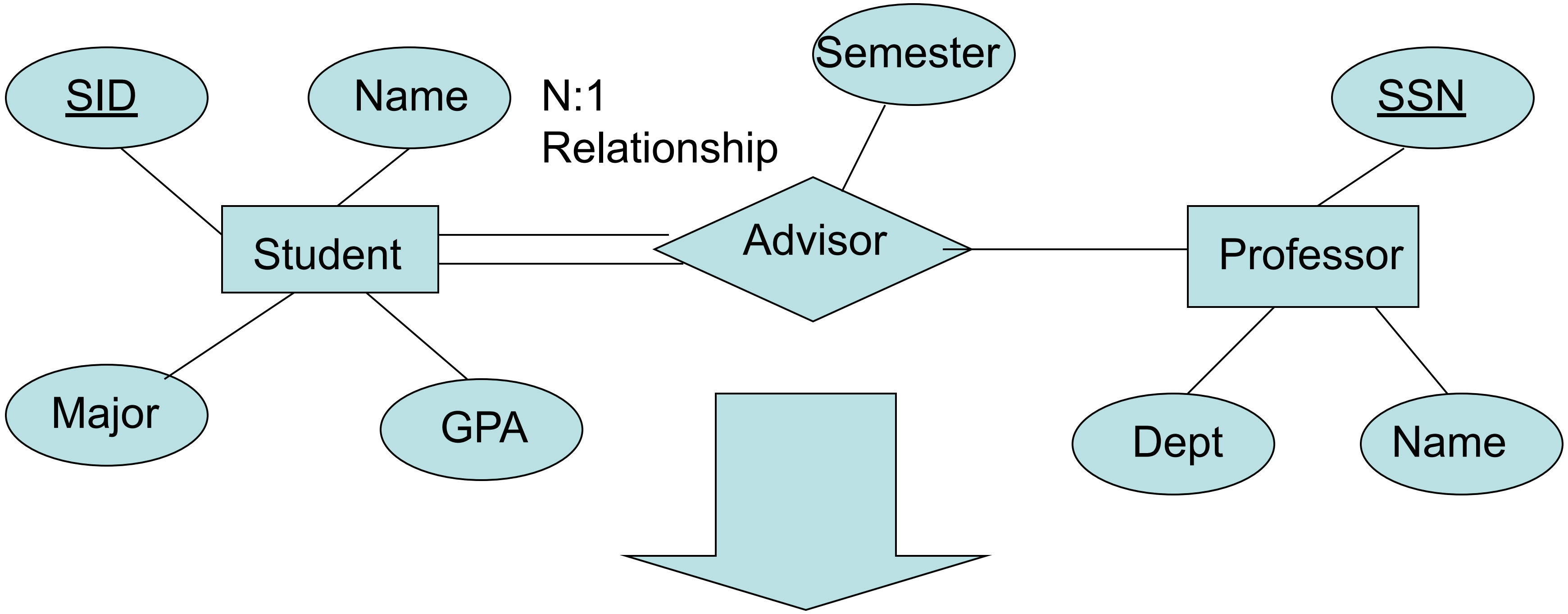
Example – One-to-One Relationship Set



SID	Name	Major	GPA	LP_S/N	Hav_Cond
9999	Bart	Economy	-4.0	123-456	Own
8888	Lisa	Physics	4.0	567-890	Loan

* Primary key can be either *SID* or *LP_S/N*

Example – Many-to-One Relationship Set

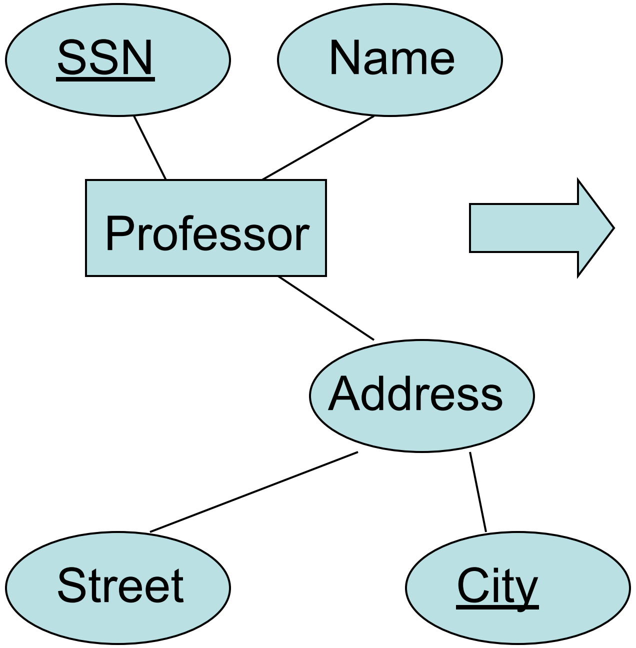


<u>SID</u>	Name	Major	GPA	Pro_SSN	Ad_Sem
9999	Bart	Economy	-4.0	123-456	Fall 2006
8888	Lisa	Physics	4.0	567-890	Fall 2005

* Primary key of this table is *SID*

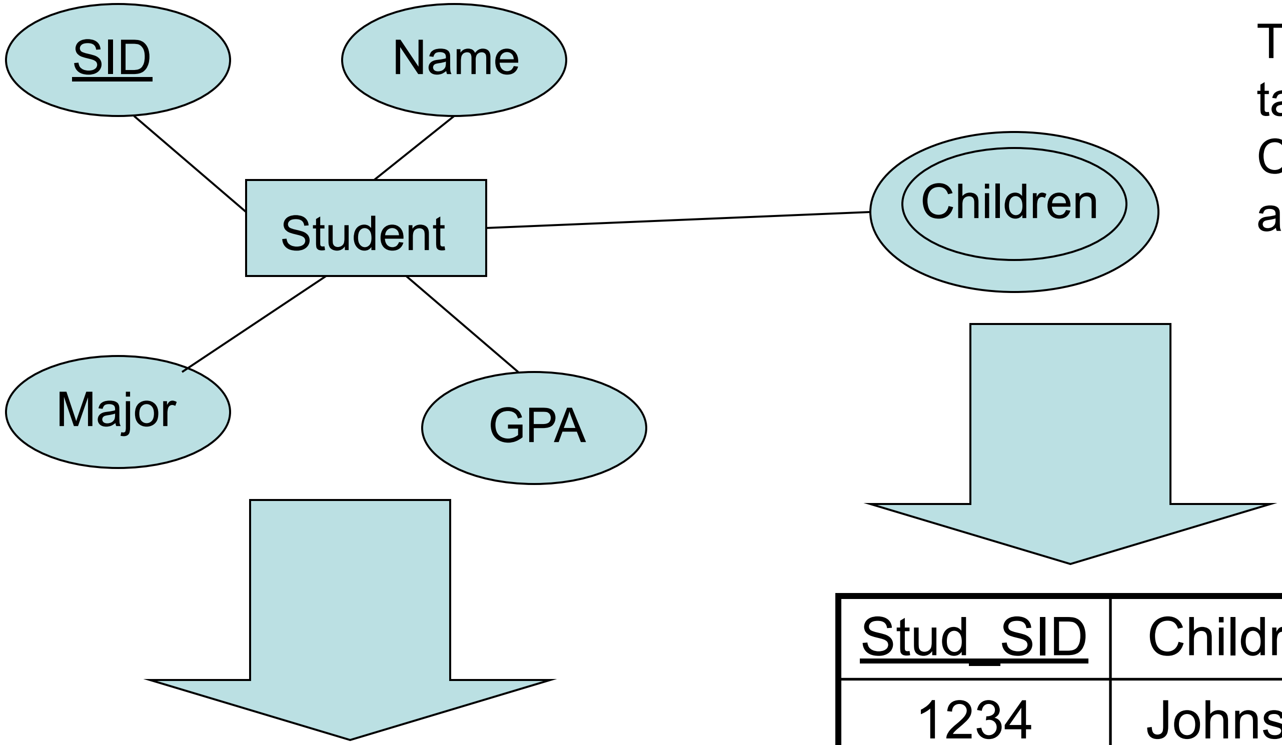
Representing Composite Attribute

- One column for each component attribute
- NO column for the composite attribute itself



<u>SSN</u>	Name	Street	City
9999	Dr. Smith	50 1st St.	Fake City
8888	Dr. Lee	1 B St.	San Jose

Example – Multivalue attribute

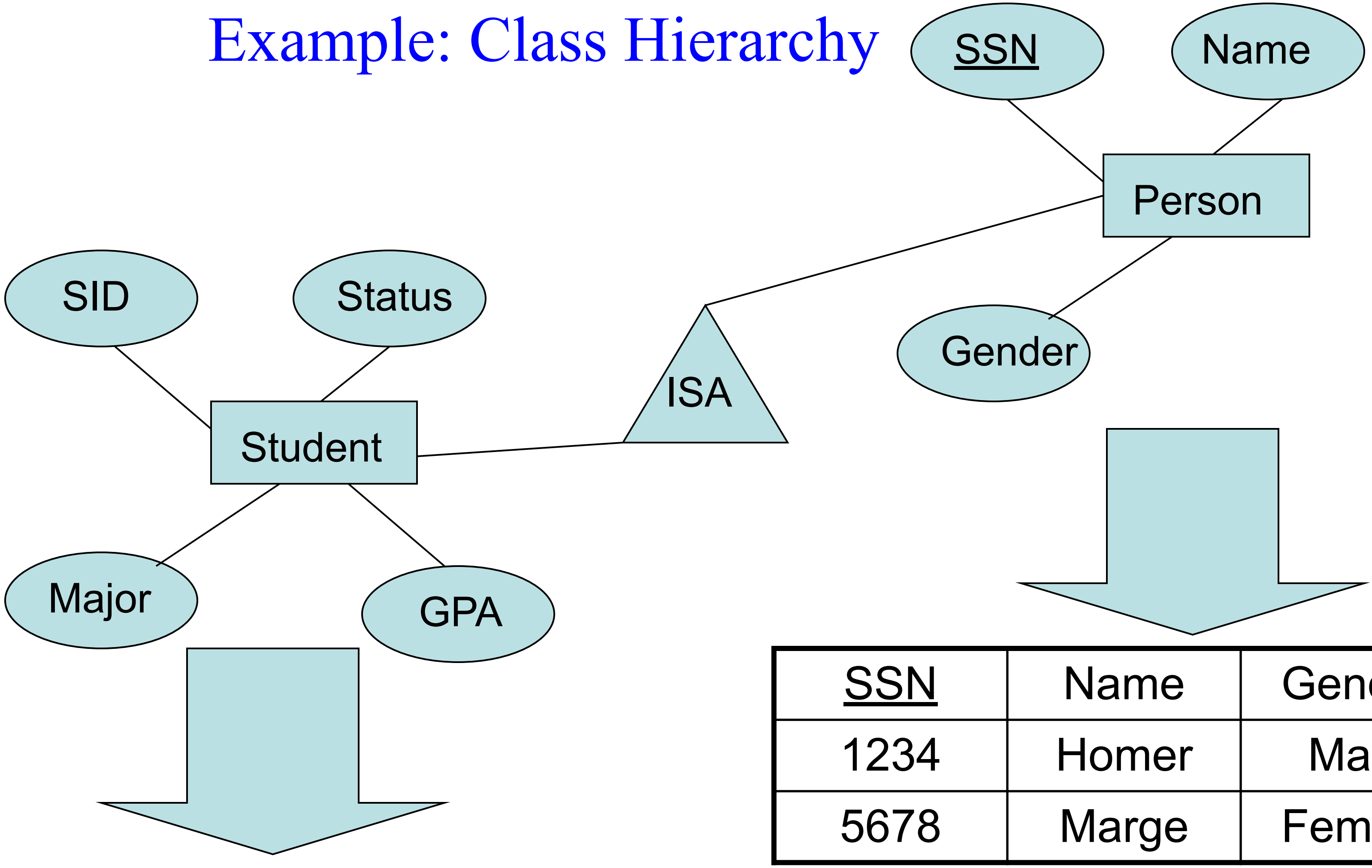


The primary key for this table is Student_SID + Children, the union of all attributes

SID	Name	Major	GPA
1234	John	CS	2.8
5678	Homer	EE	3.6

Stud_SID	Children
1234	Johnson
1234	Mary
5678	Bart
5678	Lisa
5678	Maggie

Example: Class Hierarchy



SSN	Name	Gender
1234	Homer	Male
5678	Marge	Female

SSN	SID	Status	Major	GPA
1234	9999	Full	CS	2.8
5678	8888	Part	EE	3.6

Conclusion

- An **entity relationship model**, also called an **entity-relationship (ER)** diagram, is a graphical representation of **entities** and their **relationships** to each other, typically used in computing in regard to the organization of data within databases or information systems.

Exercise:

Draw an ER diagram for an online library system with entities like books, authors, and members.

Normalize a table containing student data (name, courses, grades) to 3NF.