

Entity-Relationship model

- Chapter 2 in the book
- Chen 1976

Database design

- Goal: specification of database schema
 - Conceptual schema of data - Independent of the DBMS.
- Methodology:
 - Use E-R model to get a high-level graphical view of essential components (data) of the considered world and how they are related.
- E-R model: World viewed as set of:
 - Entities
 - Relations between entities

Entity

- Entity: a concrete object involved in the considered world.
 - Examples: a particular person (John), a particular class (cs387), a particular movie (Miss Congeniality), an academic department (CSIS)
- Entity type / Entity set: Set of similar objects.
 - Analogy: Entity/Variable - Entity Type/Variable type
 - Examples: students, classes, movies, departments

Entity type attribute

- **Attribute:** To each entity we associate a set of attributes that characterize it.
 - Examples: *fname* is an attribute for a student.
enrollment is an attribute for a class.
- An **Entity type** is describe by a set of attributes.
 - Example: *Student(fname, lname, address, hobbies, age)*
- **Domain of an attribute:** Possible values of an attribute.
 - Examples:
fname is a *string*.
age is in $\{0, 1, 2, \dots, 149, 150\}$.
salary is a positive real number.

- Different types of attributes

- **simple**

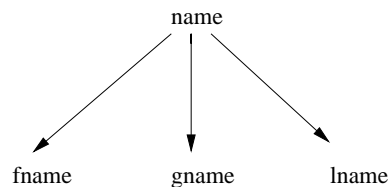
- Example: *fname*

- **composite**

- Example: Consider the entity type

Student(name, address, hobbies, age)

name is composed of the first name, the middle name and the last name of a student.



- **single-valued**

- one value

- **multi-valued**

- Consider the entity type

Student(fname, lname, address, hobbies, age)

and a particular student such that *fname* = *John*, *lname* = *Smith*, *address* = *Paris*, *age* = 10, and he can have zero, one or several hobbies. *hobbies* is a multi-valued attribute.

- **derived**

- Example: The *age* can be computed from the date of birth.

Key

Attributes of an entity type

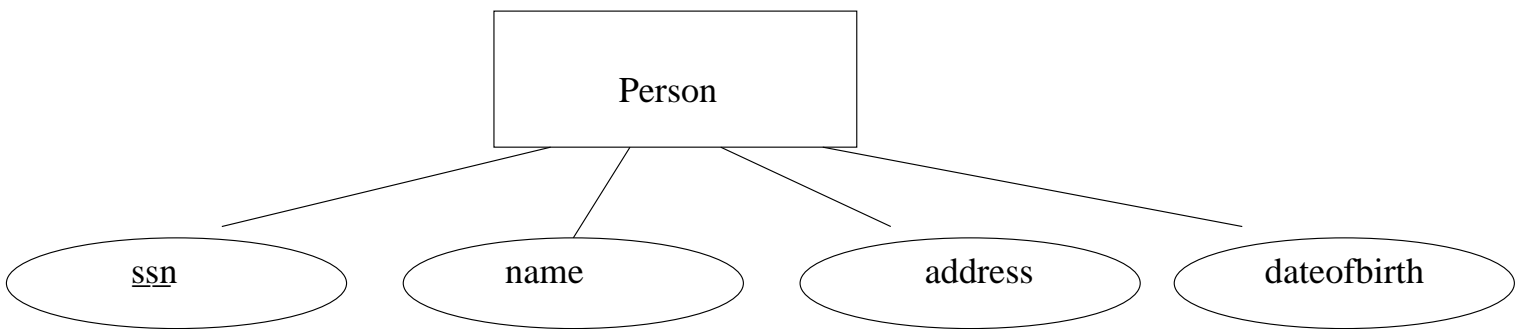
- No two entities in an entity type can have the same values for all attributes.
- A **key** (or **superkey**) is a set of attributes that uniquely identifies an entity.
 - Any superset of a superkey is a superkey.
- A **candidate key** is a **minimum** set of attributes that uniquely identifies an entity.
 - No proper subset of the candidate key is a superkey.
- Examples:
 - A social security number is enough to distinguish one entity customer from another.
Customer(ssn, fname, lname, address)
 - A name could be a superkey if no two persons have the same name.

- Notes:
 - There may exist more than one superkey.
 - The entire set of attributes of an entity type is always a superkey.
- A **primary key** is a candidate key chosen by the database designer as the principal means of identifying entities within an entity type.
 - Attributes of a primary key must be attributes that never or rarely change.
- An entity type with a primary key is said a **strong entity type**.

Entity type schema

- **Entity type schema:**
 - Entity type name,
 - Attributes (and their domains),
 - Key constraints.

Entity type graphical representation



Representation of entities in a table

- We represent an entity type by a table (\Rightarrow relation).
- We represent entities by rows in the table.
- Attributes are columns of the table.
- Problems ?
- Will be discussed in a following chapter.

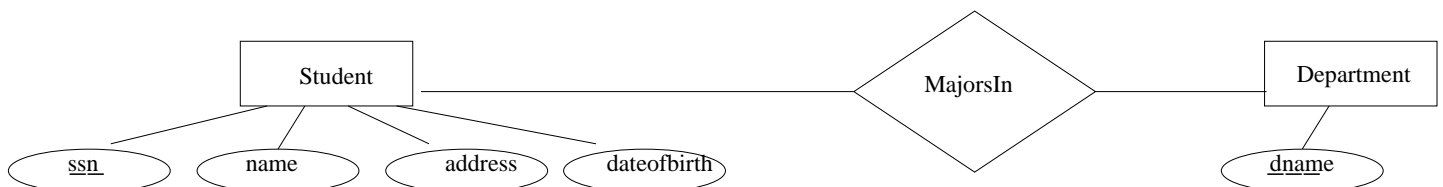
Graphical representation

- Rectangles: entity types
- Double rectangles: weak entity types
- Ellipses: attributes
- Double ellipses: multivalued attributes
- Dashed ellipses: derived attributes
- Diamonds: relationship types
- Lines: link attributes to entity types and entity types to relationship types

Relationship

- **Relationship:** Relates 2 or more entities.
 - Formally a relationship among entities e_1, e_2, \dots, e_n is a n -tuple: (e_1, e_2, \dots, e_n) .
 - Example: John *majors* in Computer Science. Bill is the *father of* Susan.
- **Relationship type:** set of similar relationships.
 - Formally a relationship type is a relation on entity types (set of relationships).
 - Example: *Student* (Entity type) related to *Department* (Entity type) by *MajorsIn* (Relationship type).

Graphical representation:



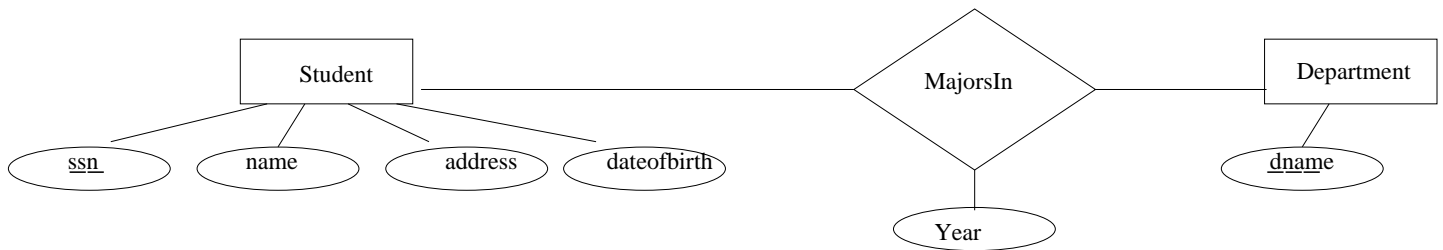
Relationship type attributes

- Attribute of a relationship type describes the relationship.

Examples:

- John majors in CS since 2000. John and CS are related. 2000 describes the relationship (SINCE).

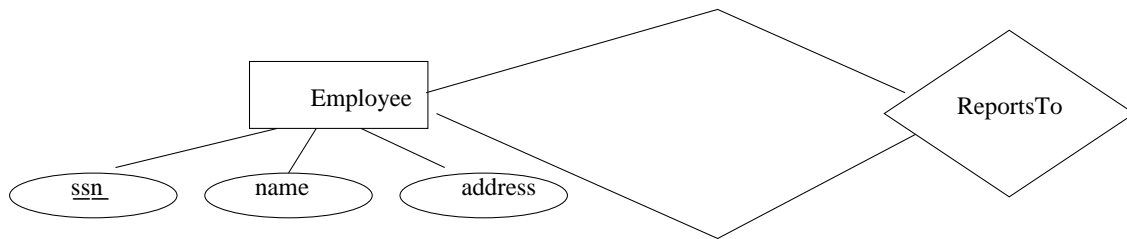
Graphical representation:



- The depositor relationship type between a *customer* and an *account* may have the attribute *access-date*.

Degree of a relationship type

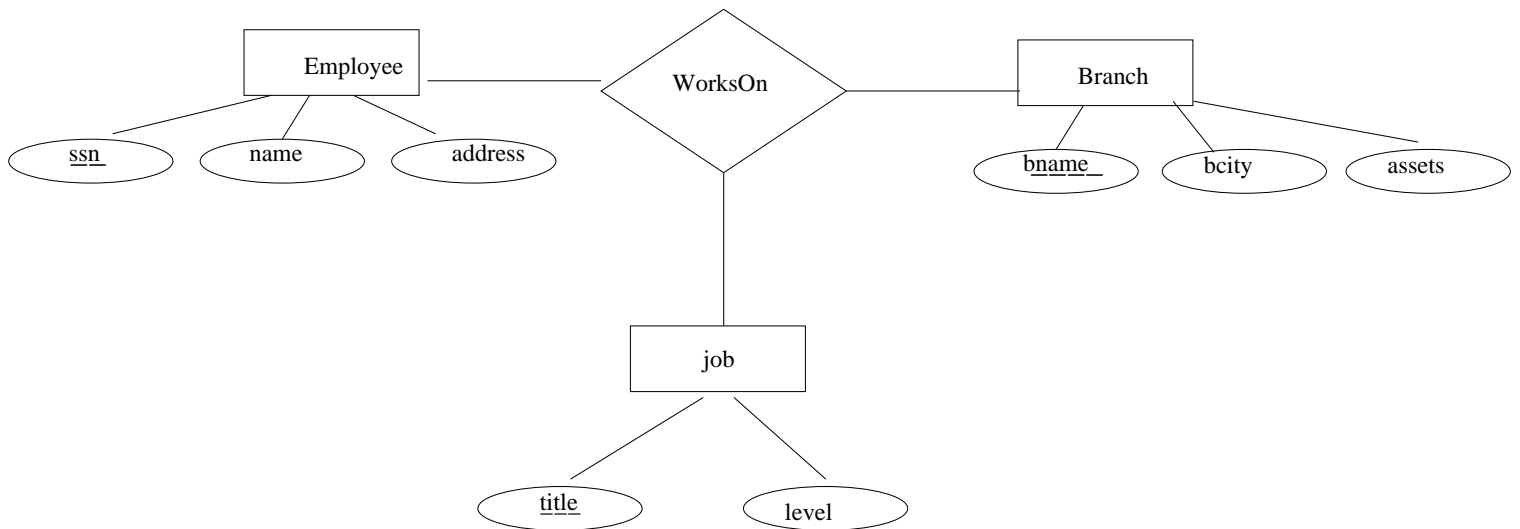
- A relationship type that involves 1 entity type is said **unary**.
 - Example: *ReportsTo* relationship type. A subordinate reports to his/her supervisor.



- A relationship type that involves 2 entity types is said **binary**.
 - Example: *MajorsIn*

- A relationship type may involve more than two entity types (rare use).

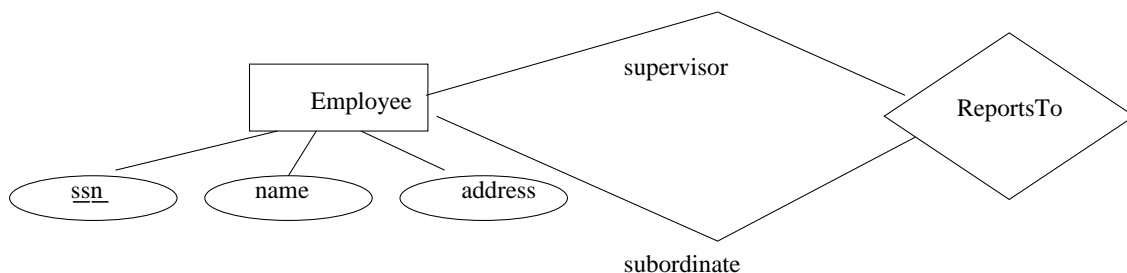
Example of a ternary relation: Bank *employees* can have responsibilities at multiple branches with different *jobs* at different *branches*. Then there is a ternary relationship type between entity types: *employees*, *jobs* and *branch*.



Role

- The function that an entity plays in a relationship is called that entity's **role**.
- Roles are in general implicit. However they are useful when the meaning of a relationship needs clarification.
- Roles labeled the edges representing a relationship.
- Consider the *ReportsTo* relationship type and “Bob reports to Mary”.

“Bob reports to Mary” is represented by a tuple (*Bob, Mary*). Bob and Mary are both elements of the same entity type *Employee*. *ReportsTo* has roles *Subordinate* and *Supervisor*.



Relationship type cardinalities

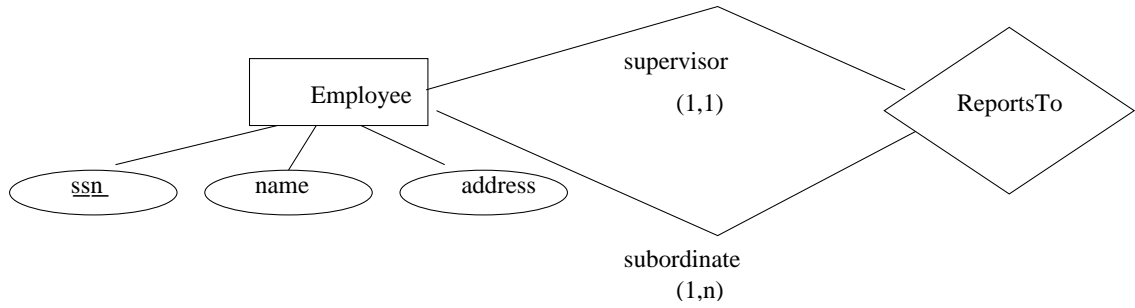
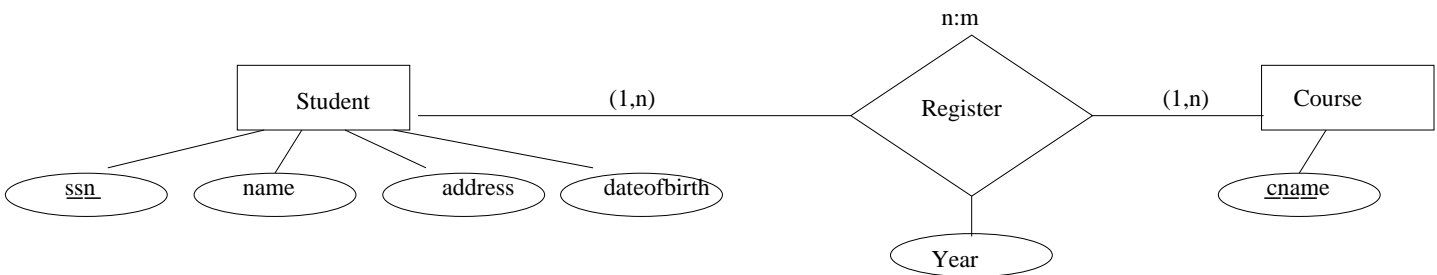
- Constraints on number of entities that can be associated with an entity under a relationship type.
 - Analogy: one-to-one (injective) function.
- Consider entity types A and B and a binary relationship type R from A to B .
 - **one-to-many (1:n)** An entity in A is associated with any number (0 or more) of entities in B . An entity in B is associated with at most one entity in A .

Example: An employee works in one company and a company has many employees.
 - **one-to-one (1:1)** An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A .

Example: A department has only one chairman and a chairman can be the chairman of only one department.
 - **many-to-many (n:m)** An entity in A is associated with any number (0 or more) of entities in B and an entity in B is associated with any number (0 or more) of entities in A .

Example: A student registers in different courses and a course has many students.

Examples of cardinalities

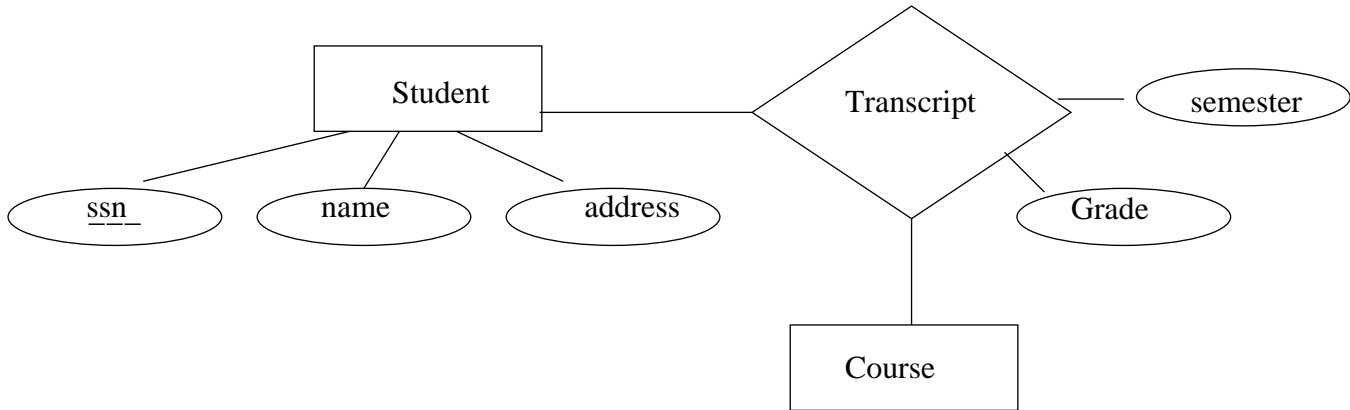
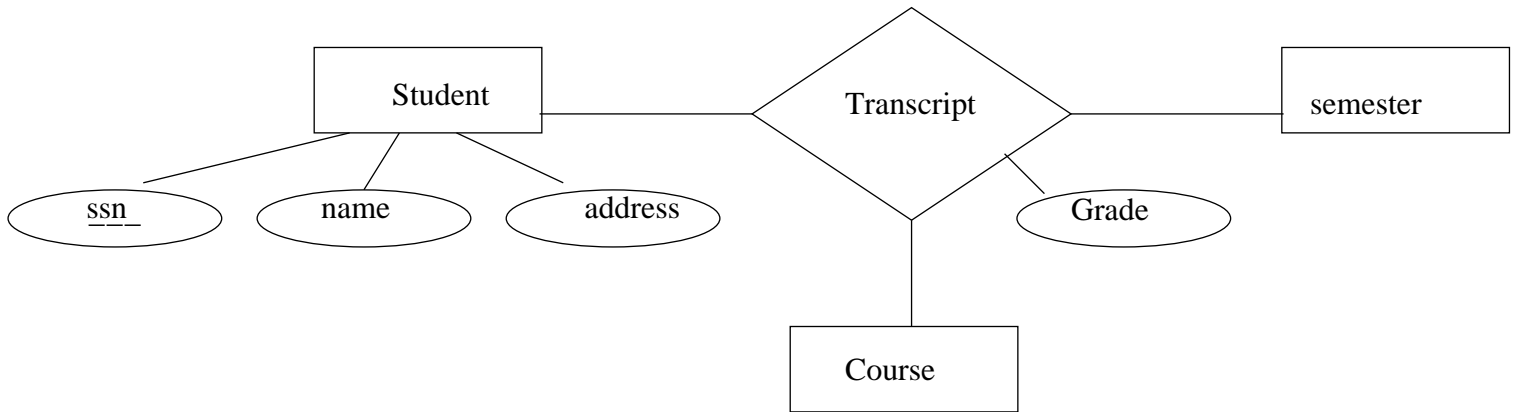


How to design an E-R model?

- There is no real receipts.
- What are the entity types and their attributes?
- What are the relationship types and their attributes?
- Implies deciding whether something is an attribute or an entity, or an entity or a relationship.
- Analyze of relationship types (cardinalities).
- Define the keys.
- Construct the E-R diagram.

Entity or attribute?

- Sometimes an object can be represented either by an entity type or an attribute.



Entity or relationship?

- Sometimes an object can be represented either by an entity type or a relationship type.

