

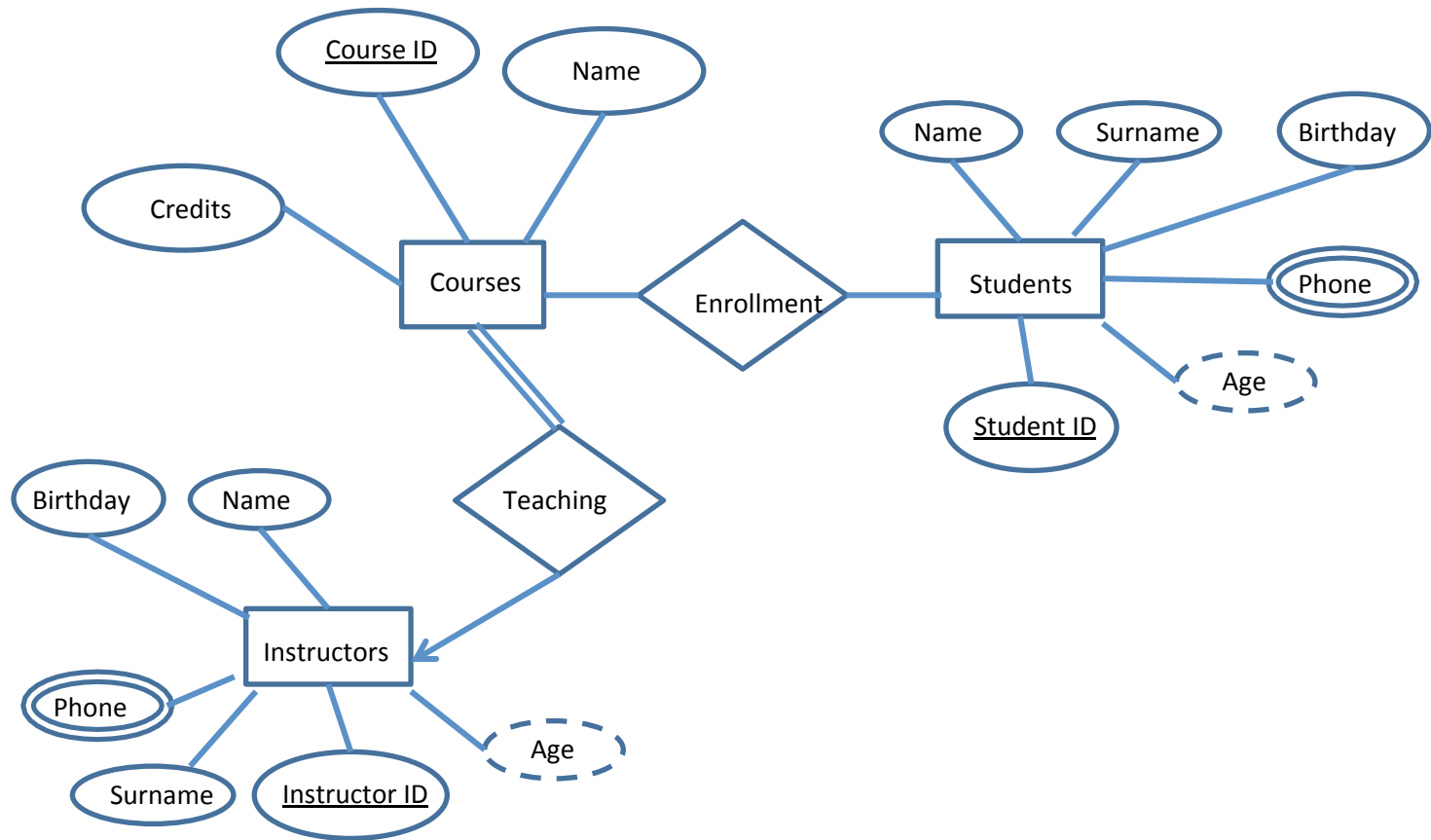
DATABASE DESIGN AND ENTITY- RELATIONSHIP DIAGRAMS

DATABASE DESIGN

- Goal of database design is to determine entities and relations that will be stored in a database together with their attributes and properties
- In order to determine this data structure we will use Entity-Relationship (E/R) diagrams

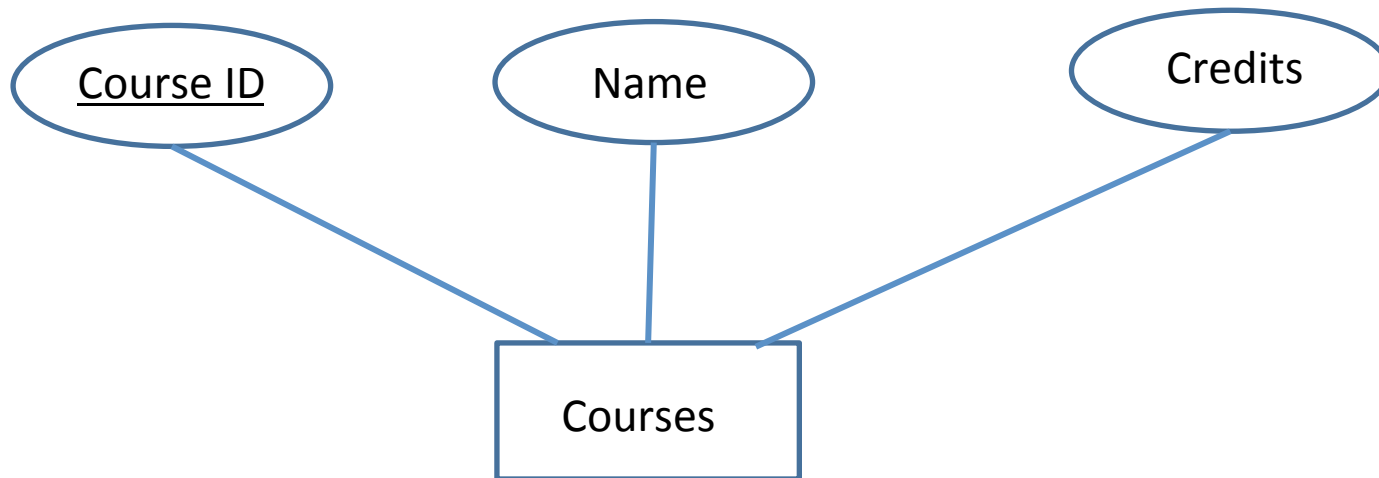
ENTITY-RELATIONSHIP DIAGRAMS

- Entity relationship (E/R) diagrams are used to design and show logical structure of databases graphically



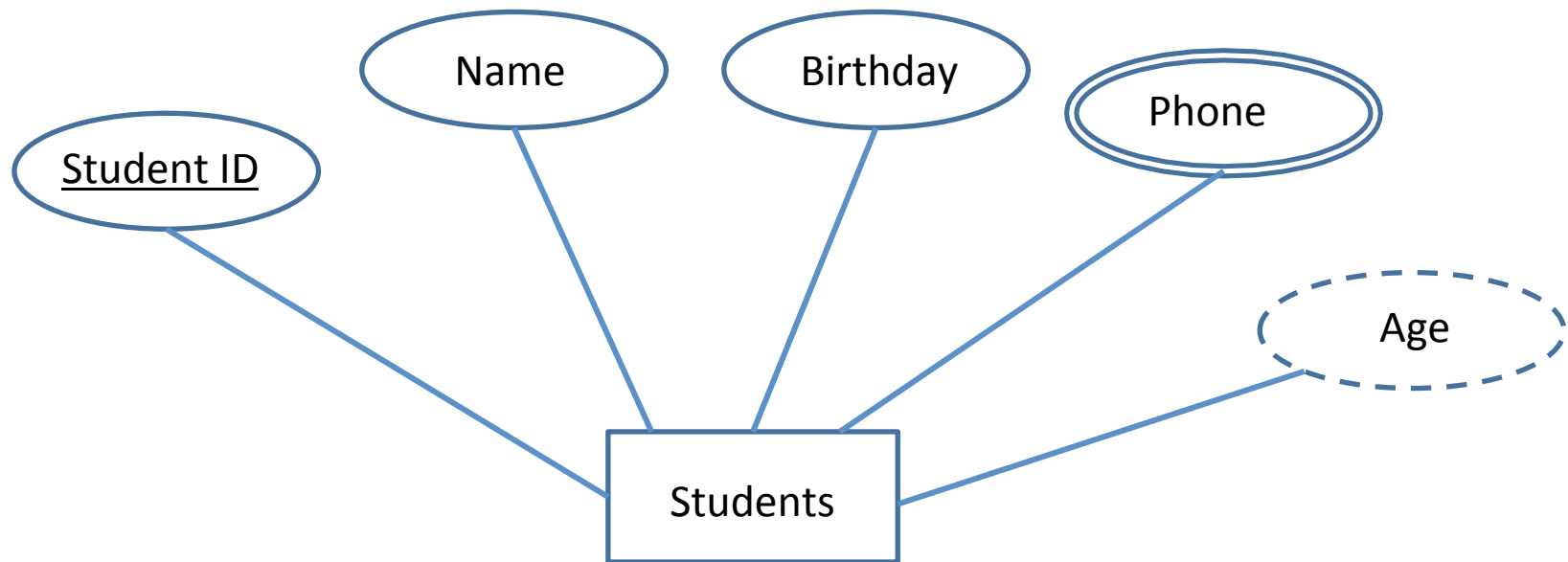
COMPONENTS OF E/R DIAGRAMS

- **Rectangles:** entities
- **Ellipses:** attributes
- **Lines:** links
- Primary keys are underlined



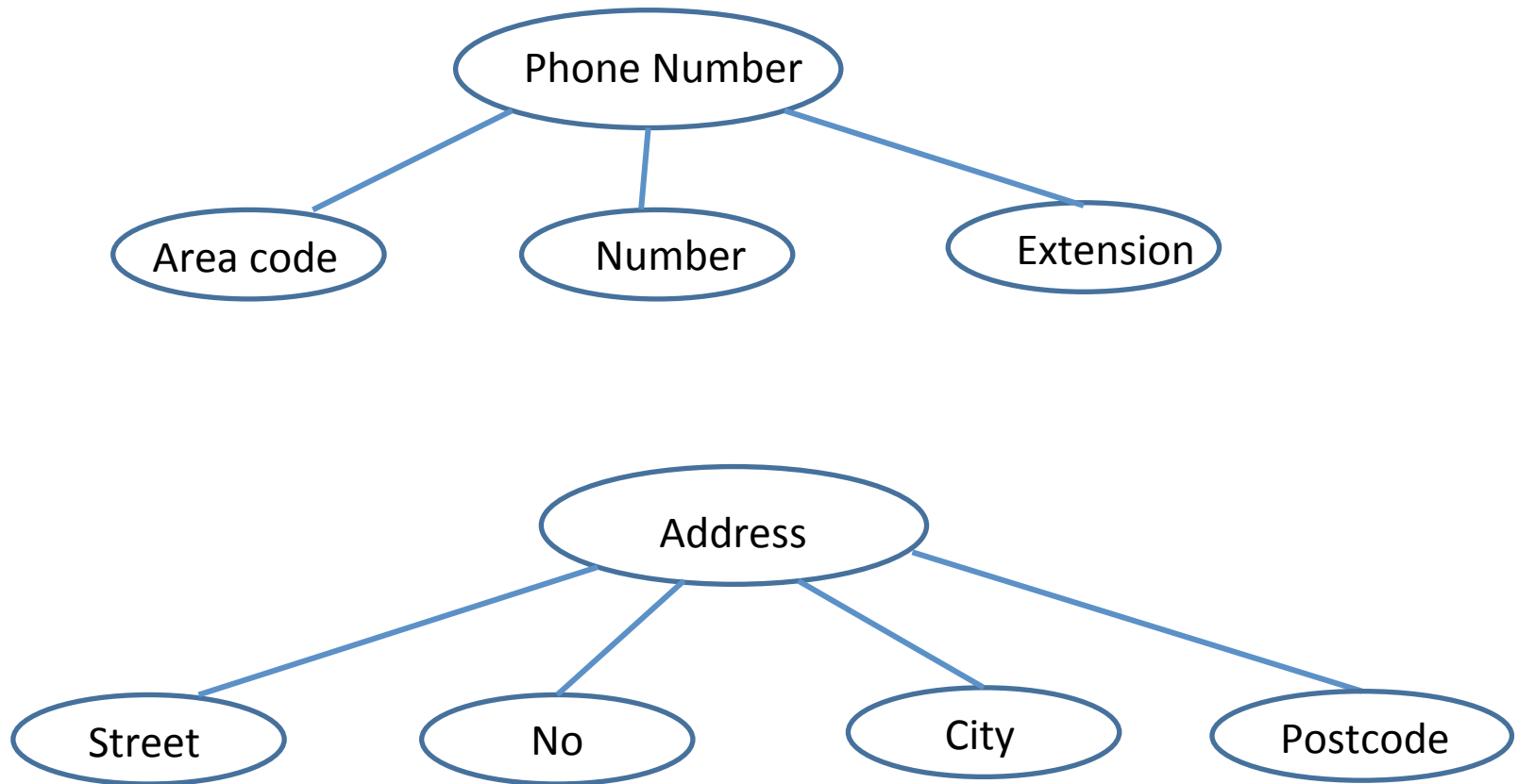
COMPONENTS OF E/R DIAGRAMS

- **Double ellipses:** multi-valued attributes
- **Dahed ellipses:** derived attributes



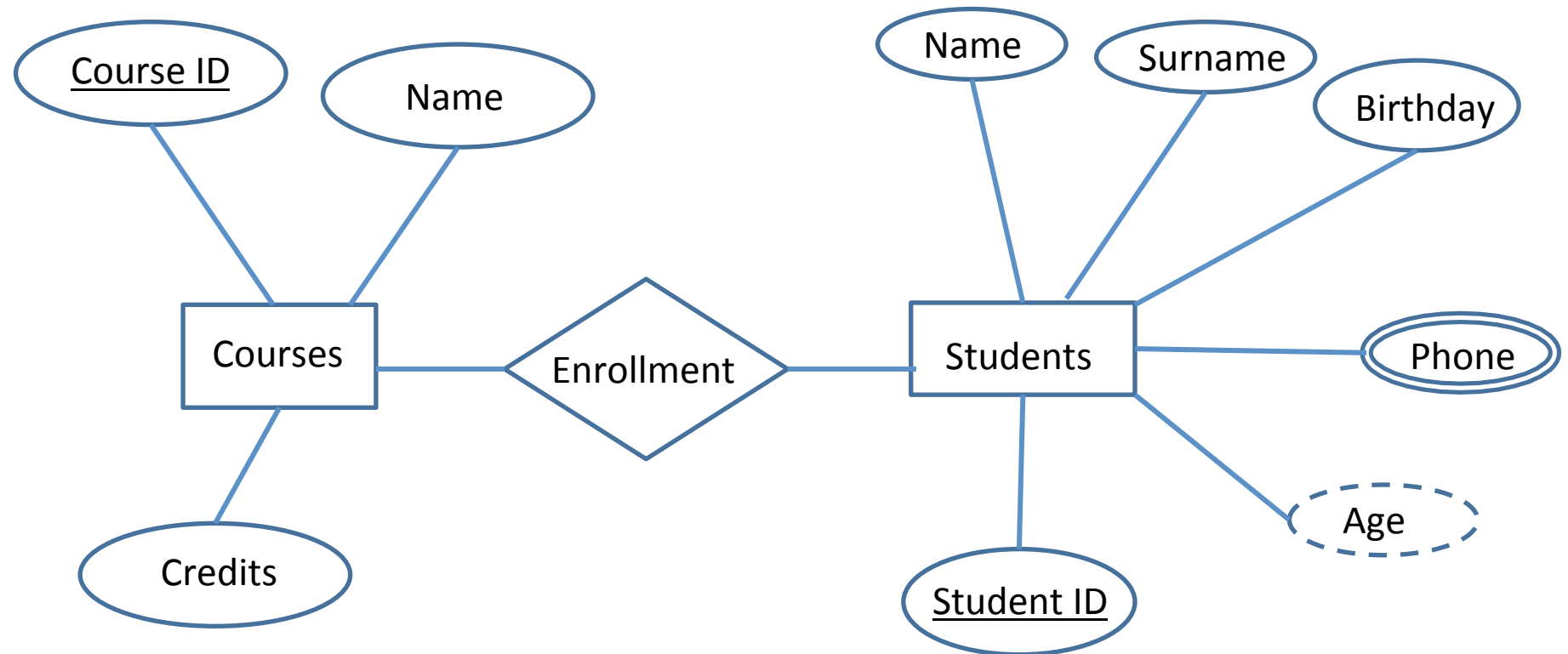
COMPONENTS OF E/R DIAGRAMS

- **Composite attributes**



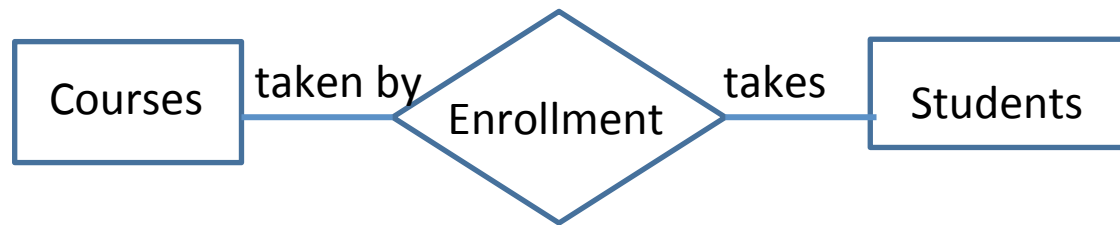
COMPONENTS OF E/R DIAGRAMS

- **Diamonds:** relations



COMPONENTS OF E/R DIAGRAMS

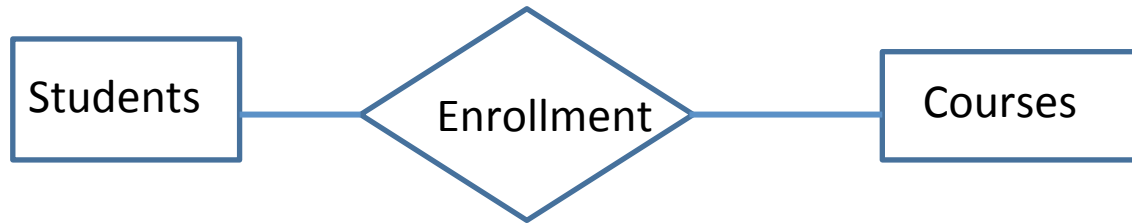
- We may indicate the role played by each entity set
- Roles are shown on the line connecting entities to relation



COMPONENTS OF E/R DIAGRAMS

Arrows show mapping cardinality for binary relations

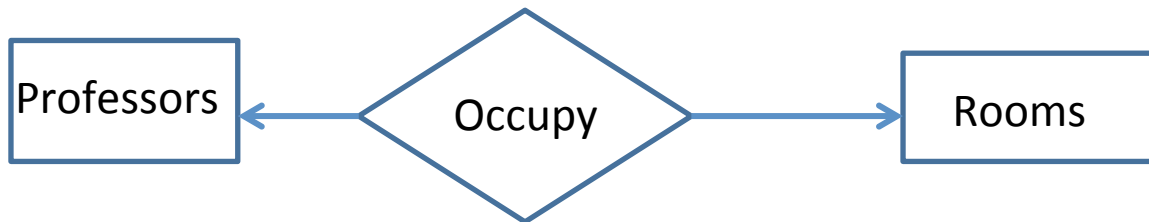
- **Many-to-many:** no arrows



- **One-to-many:** arrow pointing to parent entity

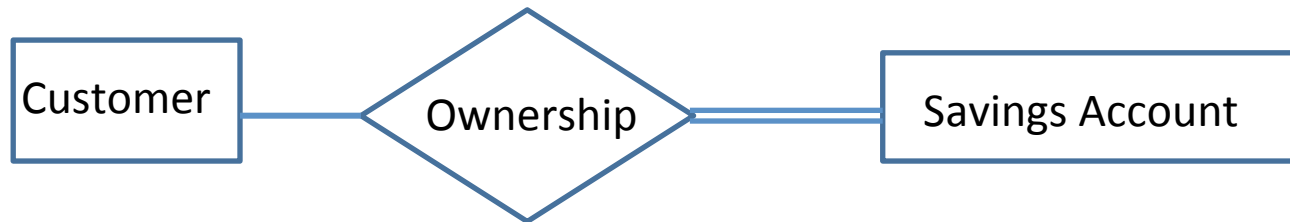


- **One-to-one:** arrows pointing to both entities



COMPONENTS OF E/R DIAGRAMS

- **Double lines:** total participation
 - Every savings account must have an owner



- Every credit card must belong to a customer

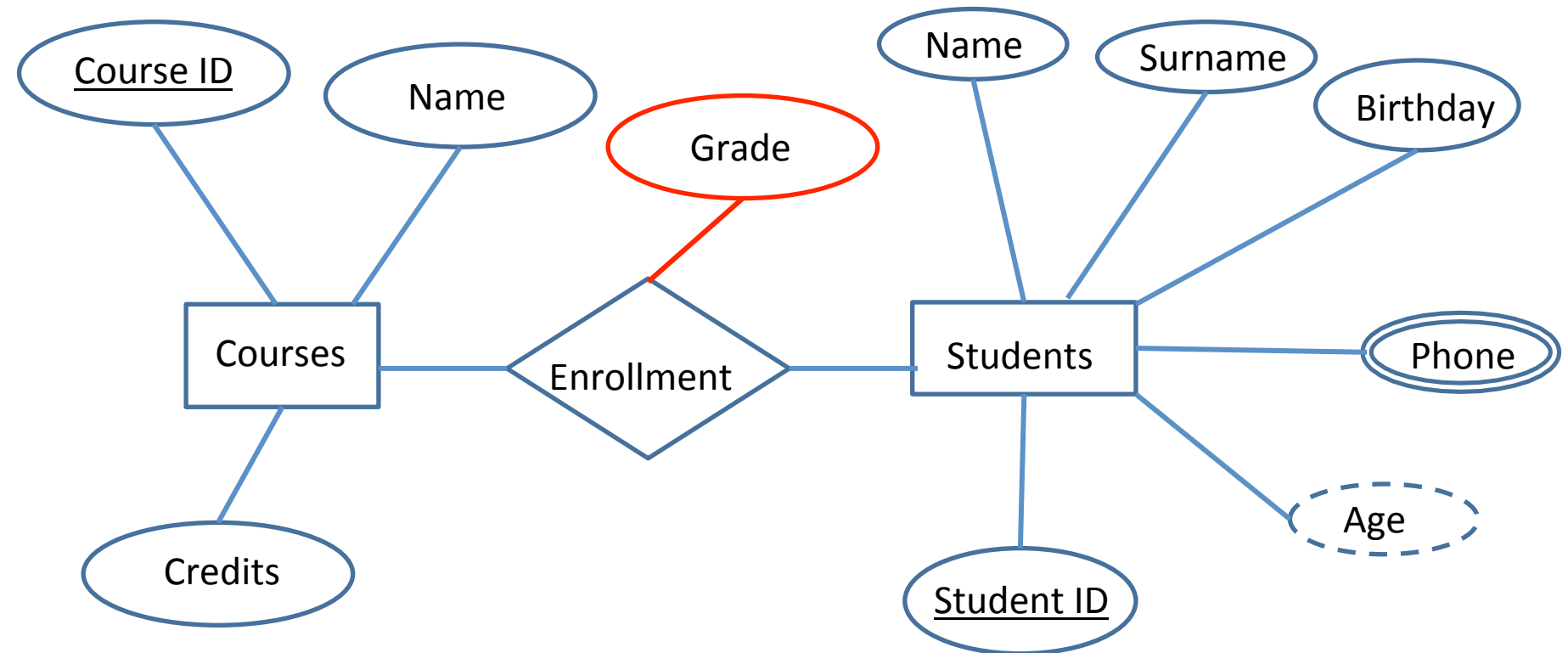


- Every husband must have a wife and vice versa



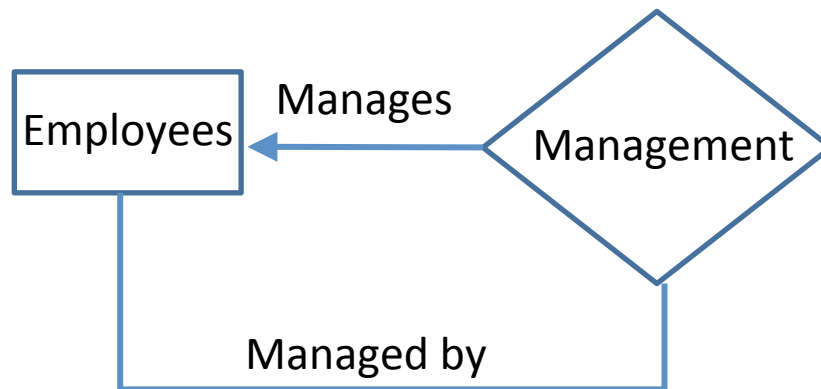
COMPONENTS OF E/R DIAGRAMS

- Attributes of relations

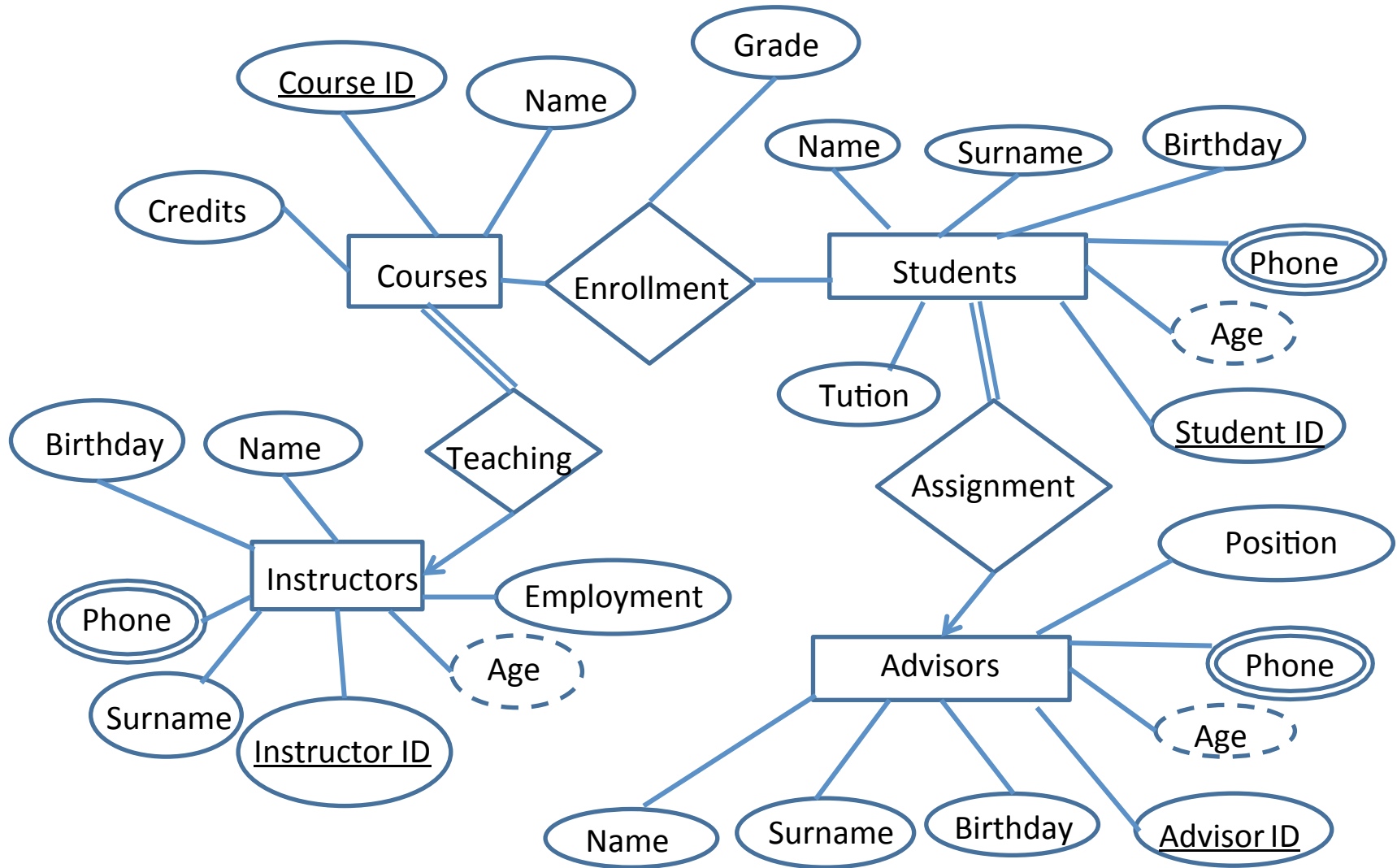


RECURSIVE RELATIONS

- An entity set can be put into a relation with itself
- Such relations are called recursive relations



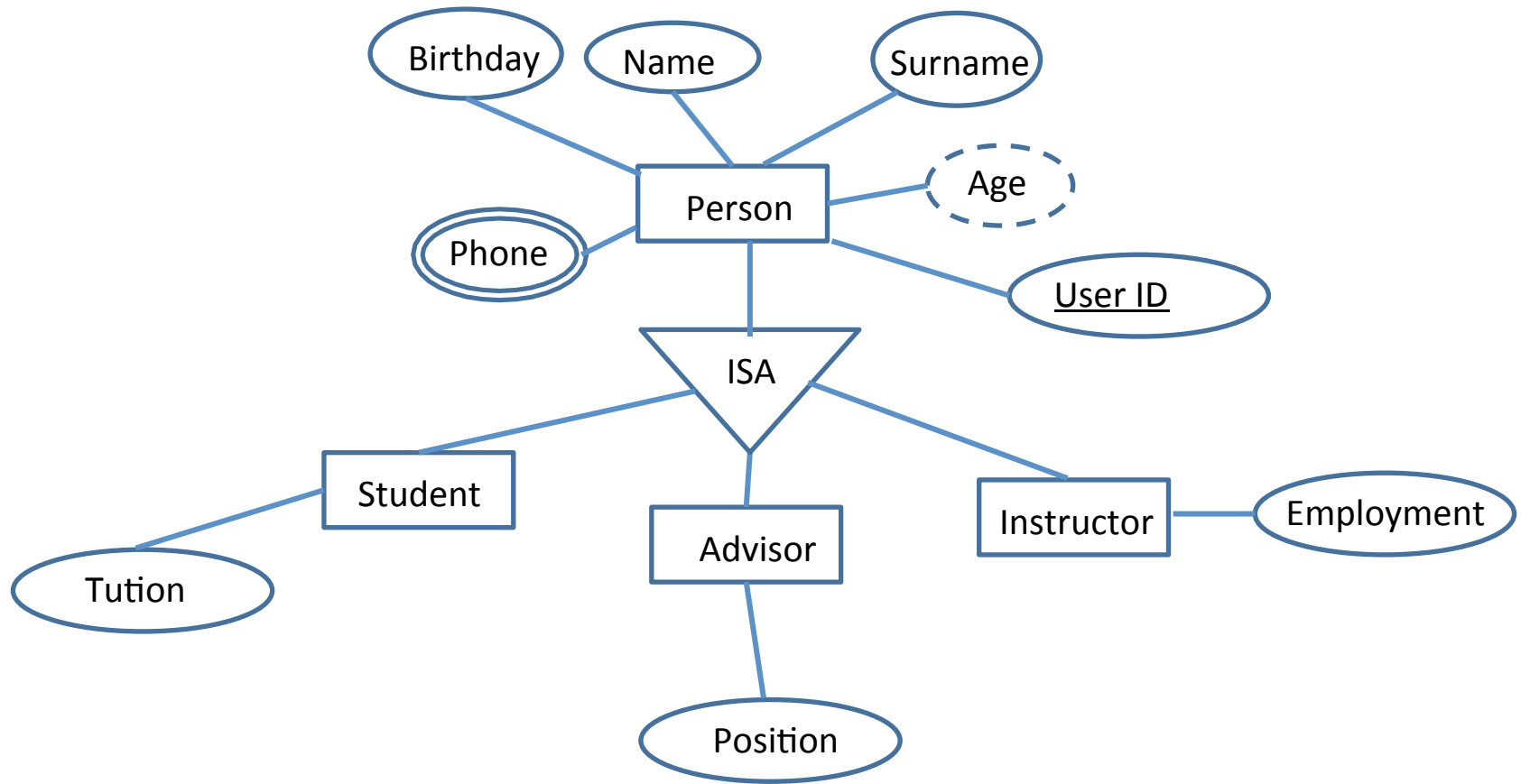
ENTITY-RELATIONSHIP DIAGRAM



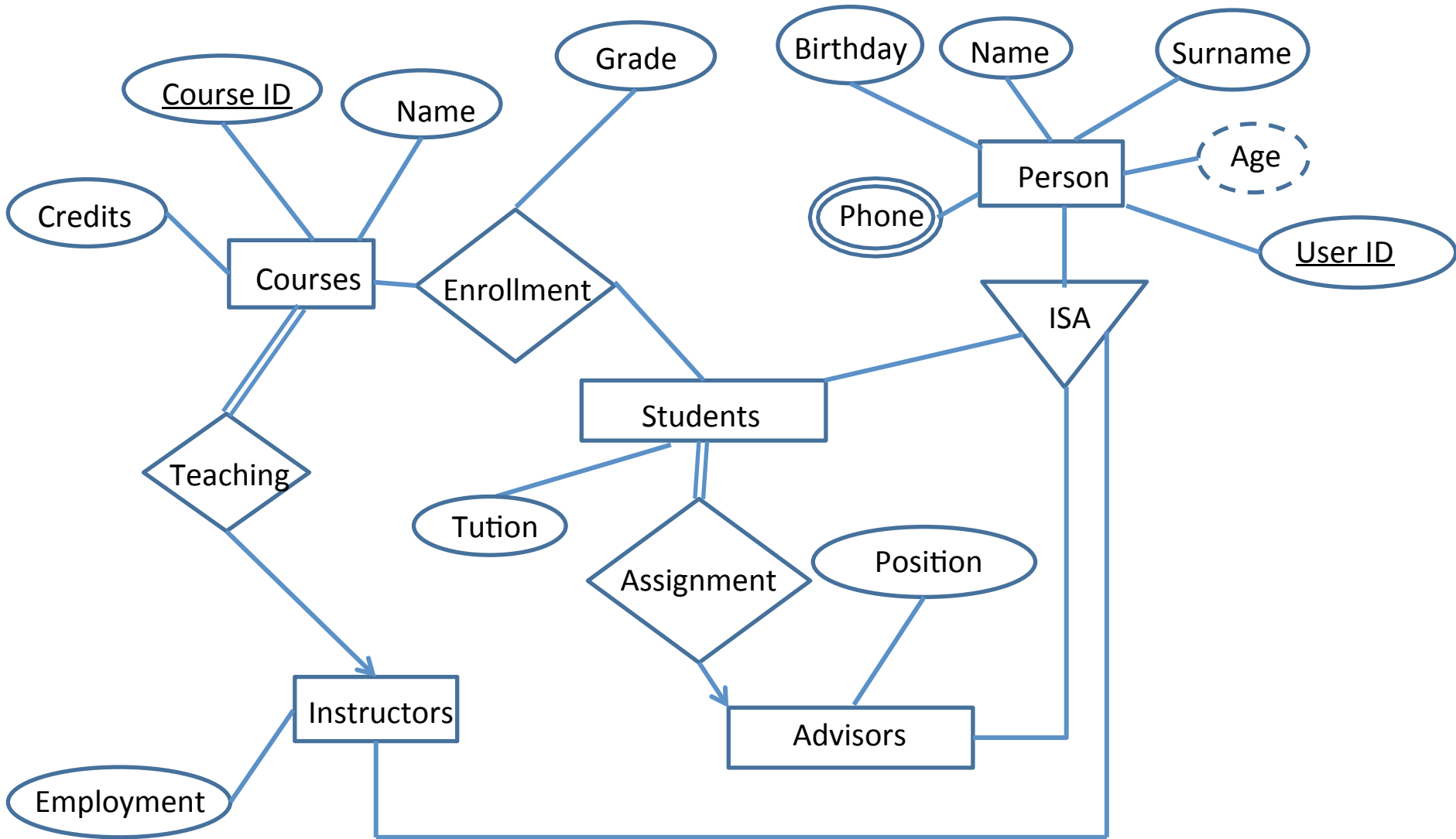
ISA RELATIONS

- Suppose that there are different entity sets having several common attributes and few distinct attributes
- Example: Students, advisors and instructors
- ISA relations allow making generalizations by combining such entity sets under a higher level entity set

ISA RELATIONS



ENTITY-RELATIONSHIP DIAGRAM



WEAK ENTITY SET

- Sometimes we may need to define entities whose existence depends on other entities
 - Courses
 - Attributes: ID, Name, credits, description
 - Sections
 - Attributes: section num. (discriminator), number of students, time, place
- A section cannot exist in database without associated course
- Section: WEAK ENTITY
- Course: STRONG ENTITY

WEAK ENTITY SET

- **Strong entity:** an entity that can exist independently
- **Weak entity:** an entity that cannot exist without an associated strong entity
- Association between the weak entity and the corresponding strong entity must be established using a relation
- Participation of the weak entity must be **total**

WEAK ENTITY SET

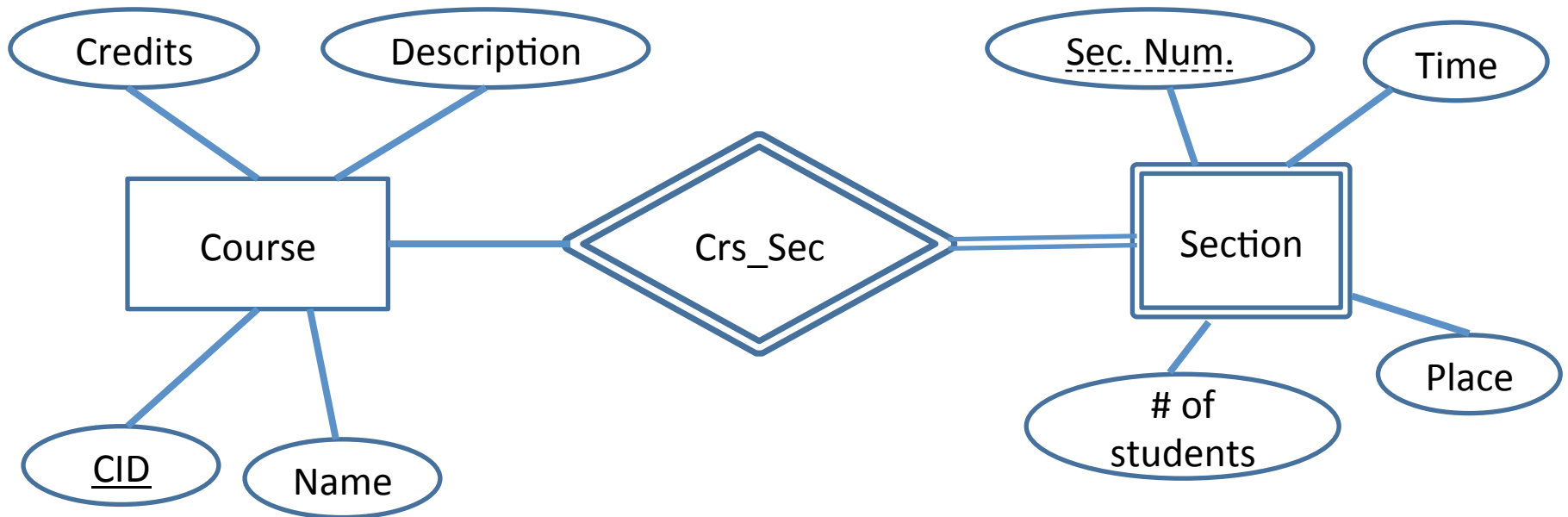
- Strong entities have their own primary keys
- Weak entities may not have their own primary keys
- They may have **discriminators**
- In this case, primary key can be indirectly obtained by combining the primary key of the associated strong entity with the discriminator

WEAK ENTITY SET

- Example:
 - Courses (strong)
 - Attributes: **ID**, Name, credits, description
 - Sections (weak)
 - Attributes: **section num. (discriminator)**, number of students, time, place
- The pair (**ID, section number**) can be used as the primary key of the entity set sections

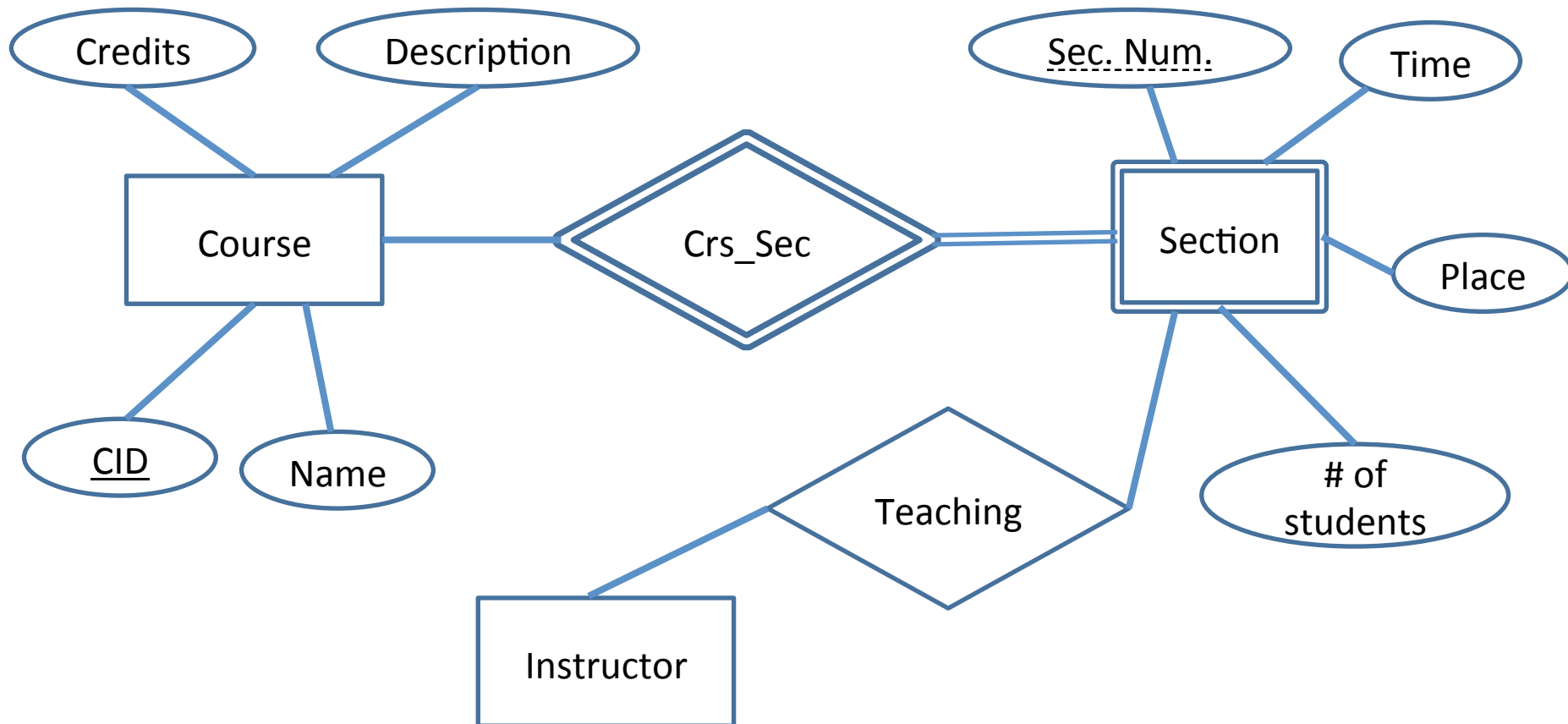
WEAK ENTITY SET

- In E/R diagram
 - **Weak entity set:** double square
 - **Associated relation:** double diamond
 - **Discriminator:** underlined with dashed line



WEAK ENTITY SET

- A weak entity set can participate into a relation other than the relation associated with its strong entity set
- Do not use double diamonds for this relation.
- Denote it using single diamond



CASE STUDY: COMPANY DATABASE

- The company is organized into departments. Each department has a name, number, location, phone number and a single employee who manages the department. A department may have more than phone number.
- Departments control projects. A department may control more than one project while sometimes it may not be involved in any project at all. However, a project must always be controlled by a single department. Each project has a unique name, unique number and a description.
- We store each employee's social security number (SSN), name, address, salary, gender and birth date. Name is composed of first name and last name.
- Each employee always works for one department but may work on several or no projects. Each department must always have at least one employee working for it. The same applies to projects. We keep track of the number of hours per week that an employee currently works on each project.
- An employee can be manager of only one department.
- Each employee has a supervisor whose is also an employee.
- Each employee may have a number of dependents. For each dependent, we keep track of his/her SSN, name, gender, birth date, and relationship to the employee. Assume that only one of the parents of a depended works in the company

CASE STUDY: COMPANY DATABASE

