ER Diagrams

Conceptual Data modeling

Entity

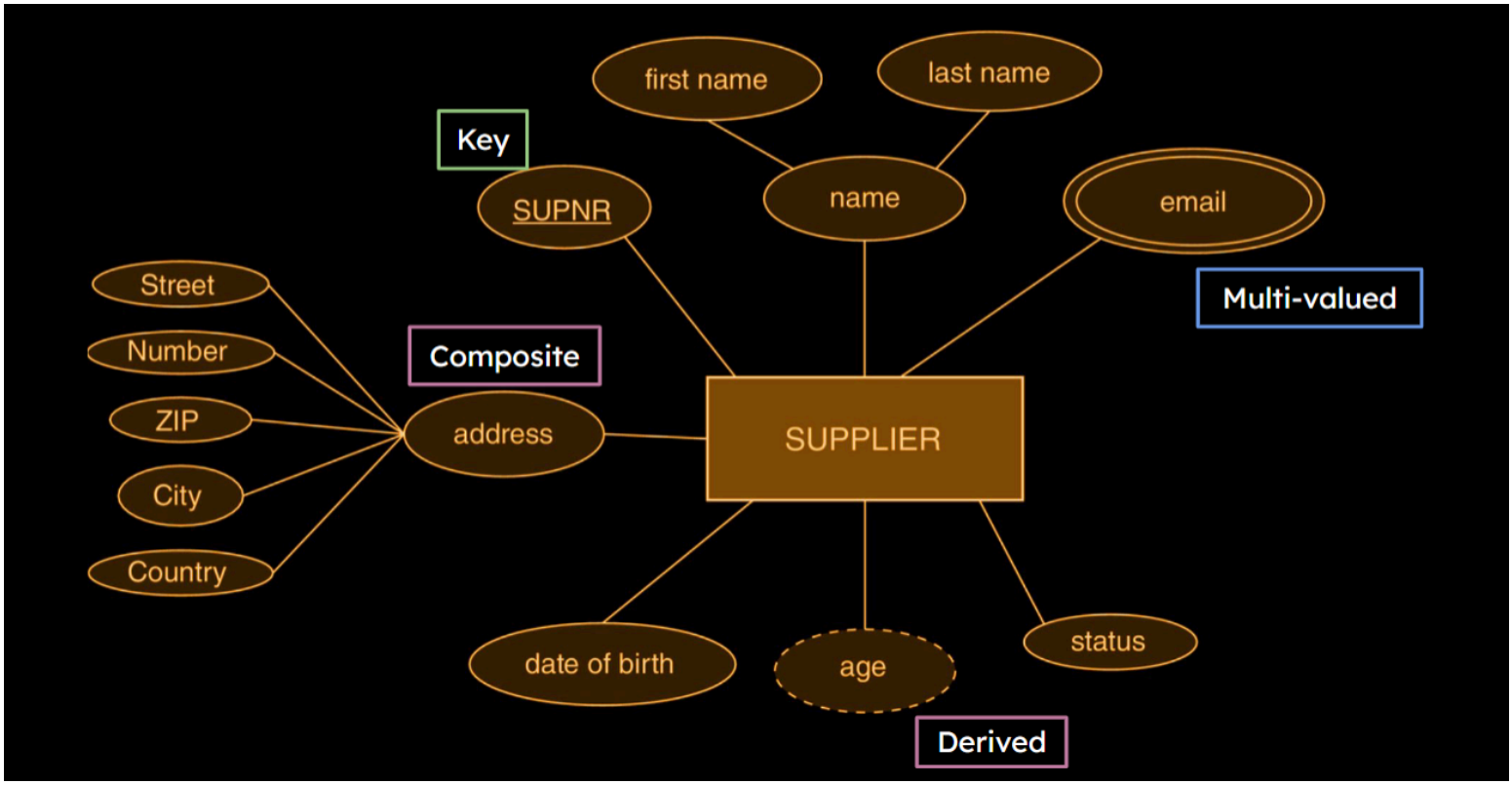
- Representted as a rectagle in the diagram
- An entity is an instance

Entity keys

- Key
  A Key of a entity, marked with a underline for primary key. ER diagrams do not show secondary keys
- Composite
  A attribute with more attributes under it
- Multi-valued
  An attribute type that can have multiple values

Usually, multivalued attributes in a database schema (DDL) are represented by making distinct tables just for storing these attributes. Every table has a foreign key that references both the multivalued attribute itself and the primary key of the entity to which it belongs. This method facilitates efficient multivalued data querying and manipulation while lowering redundancy and preserving data integrity.

- Derived
  Methods for derived attributes.
  Option 1: Create an attribute and maintain it like with SQL triggers.
  Option 2: Create a view that computes it.



Relationships

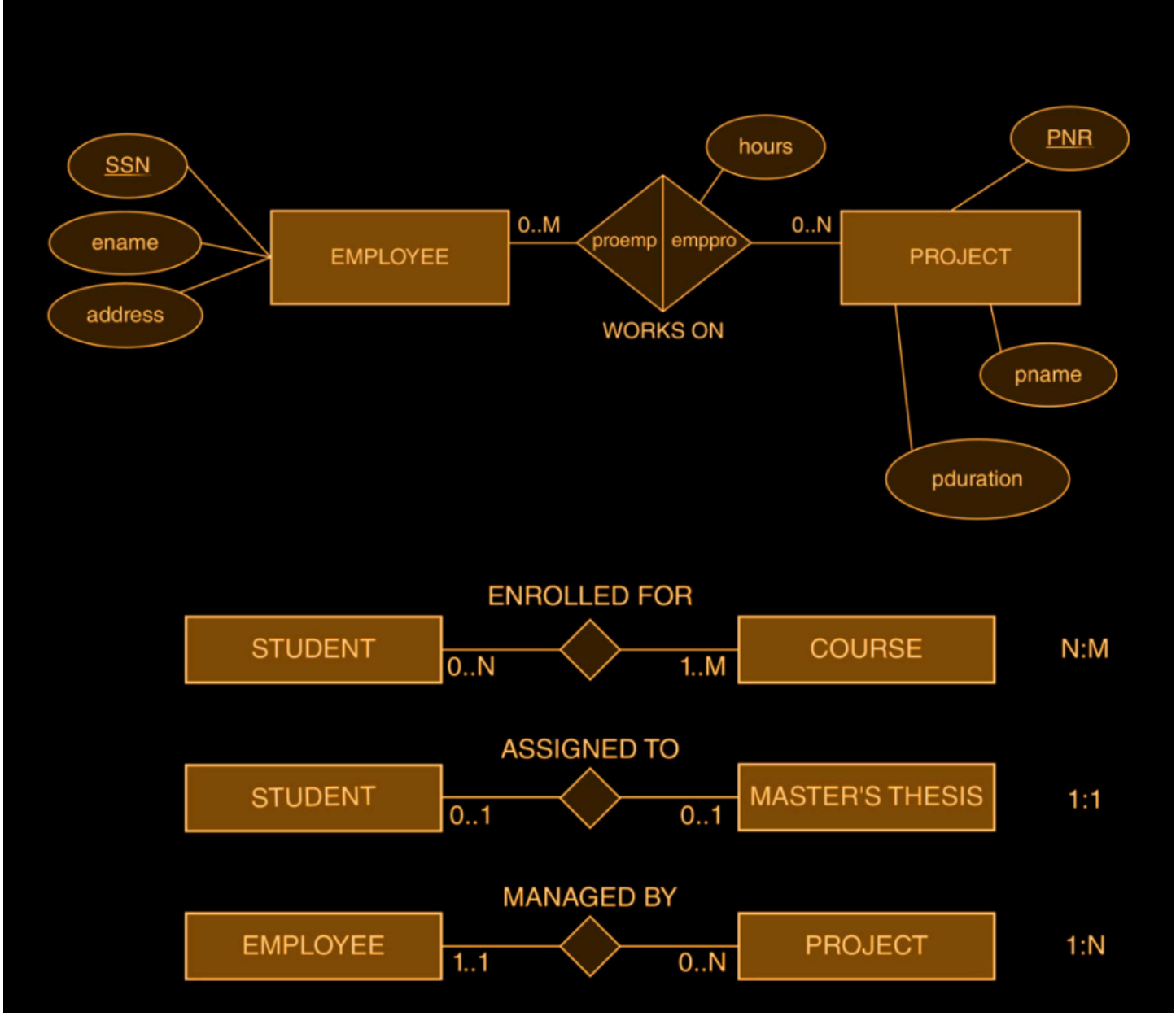
A relation is between two or more Entities (with roles)

We use the `REFERENCES` keyword to define a relationship in SQL.

Cardinality

A relationship can har cardinalities

- minimum 0 or 1
- maximum 1 or N / * / M / ...



CHECK Statement

Add constraints directly in the DDL with the `CHECK` statement.

Specialization & Generalization

The process of defining a set of subclasses of an entity type. The set of subclasses that form a specialization is defined on the basis of some distinguishing characteristic of the entities in the superclass

Example

consider an ARTIST superclass with subclasses SINGER and ACTOR. The specialization process defines an “IS A” relationship. In other words, a singer is an artist. Also, an actor is an artist. The opposite does not apply.

Generalization, also called abstraction, is the reverse process of Specialization.

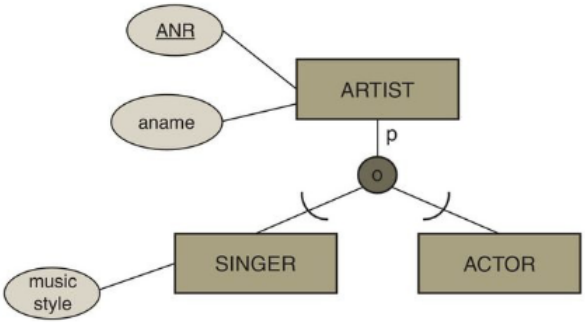
Disjointness and completeness constraint

Disjointness constraints

The disjointness constraint specifies what subclasses an entity of the superclass can belong to

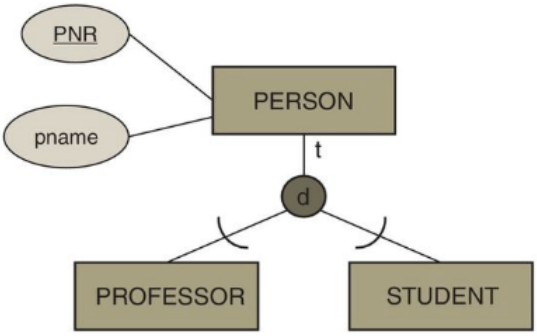
- A **disjoint** specialization is a specialization where an entity can be a member of at most one of the subclasses (Defined d in the ER)

- An **overlap** specialization is a specialization where the same entity may be a member of more than one subclass (Defined o in the ER)
  Completeness constraint
  The completeness constraint indicates whether all entities of the superclass should belong to one of the subclasses or not.
- **Total** specialization is a specialization where every entity in the superclass must be a member of some subclass (Defined t in the ER)
- A partial specialization allows an entity to only belong to the superclass and to none of the subclasses. (Defined p in the ER)
  Examples



Example of partial (p) specialization with overlap (o).

The specialization is partial since not all artists are singers or actors; The specialization is overlap since some artists can be both singers and actors.

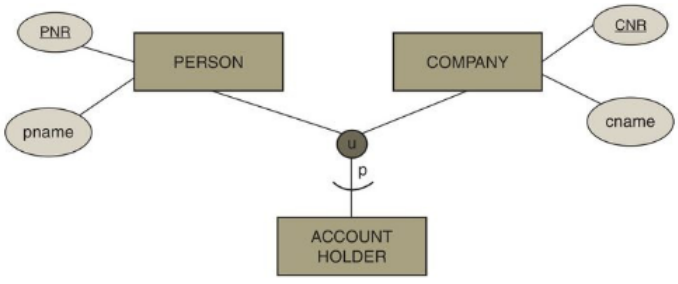


Example of total (t) and disjoint (d) specialization.

The specialization is total, since according to our model all people are either students or professors. The specialization is disjoint, since a student cannot be a professor at the same time.

Categorization

A *category* is a subclass that has several possible superclasses. Each superclass represents a different entity type. The category then represents a collection of entities that is a subset of the union of the superclasses.
Categorization is represented in the EER by the letter u from Union.



EER categorization.

Shows how the superclasses PERSON and COMPANY have been categorized into an ACCOUNT HOLDER subclass. In

other words, the account holder entities are a subset of the union of the person and company entities; The categorization is partial as represented by the letter “p”. This implies that not all persons or companies are account holders