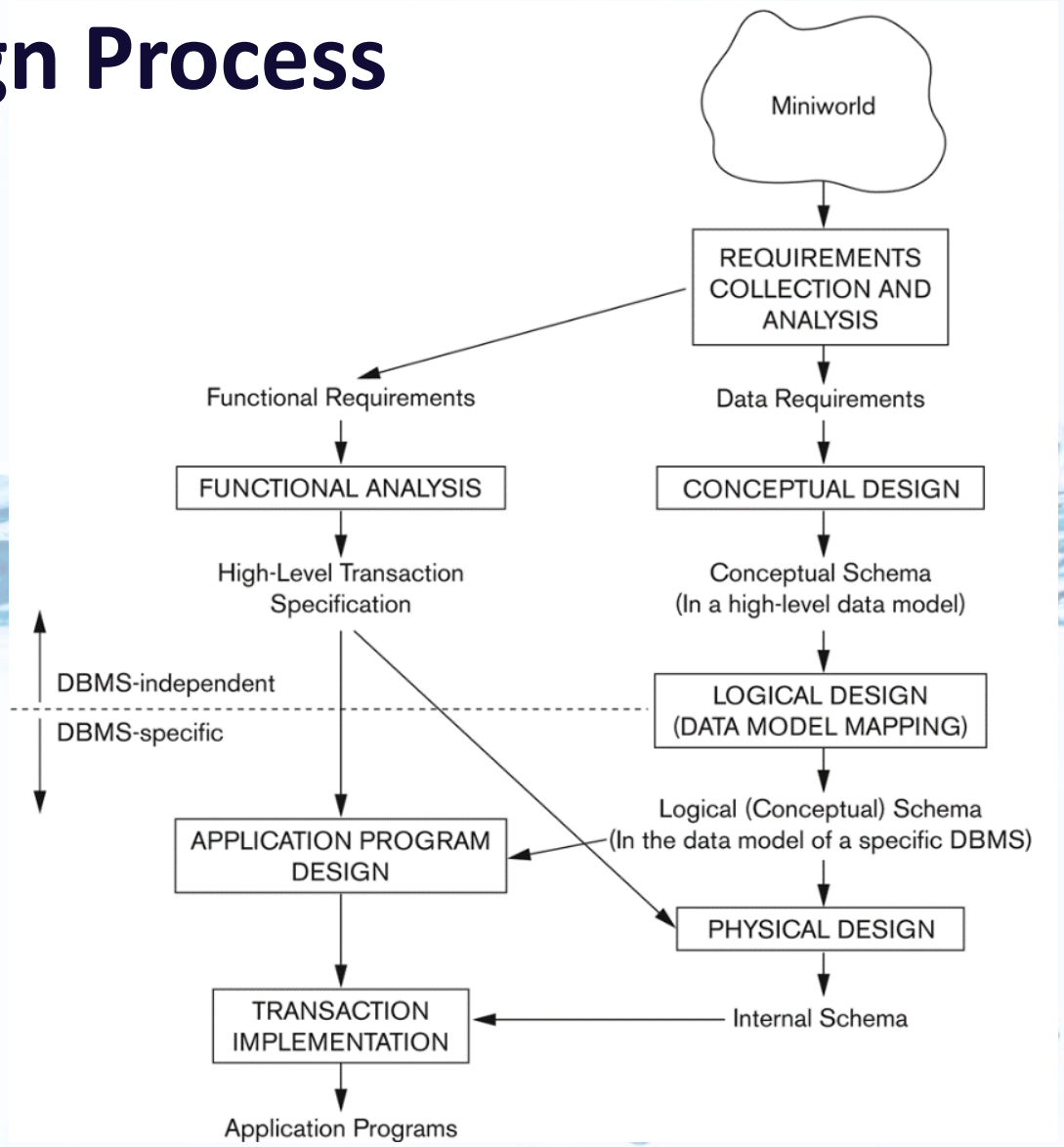
COSC 3380
Design of Database Systems

Data Modeling with Entity-Relationship (ER) Model

February 12, 2024

Overview of Database Design Process



Example: COMPANY Database

- Create a database schema based on the following (simplified) **requirements** of the COMPANY Database:
  - The company is organized into DEPARTMENTs
  - Each department has a name, number and an employee who *manages* the department
  - We keep track of the start date of the department manager
  - A department may have several locations
  - Each department *controls* a number of PROJECTs
  - Each project has a unique name, unique number and is located at a single location

Example COMPANY Database

- The database will store each EMPLOYEE's social security number, address, salary, sex, and birthdate
  - Each employee *works for* one department but may *work on* several projects
  - The DB will keep track of the number of hours per week that an employee currently works on each project
  - Also keep track of the *direct supervisor* of each employee
- Each employee may have a number of DEPENDENTs
  - For each dependent, the DB keeps a record of name, sex, birthdate, and relationship to the employee

ER Model Concepts

- **Entities and Attributes**
  - Entity is a basic concept for the ER model
  - Entities are specific things or objects in the mini-world that are represented in the database
    - Eg.: the EMPLOYEE - John Smith, the Research DEPARTMENT, the ProductX PROJECT
  - Attributes are properties used to describe an entity
    - Eg.: an EMPLOYEE entity may have the attributes of Name, SSN, Address, Gender, BirthDate

ER Model Concepts

- **Entities and Attributes**
  - A specific entity will have a value for each of its attributes
    - Eg.: a specific employee entity may have

      Name='John Smith’

      SSN='123456789’

      Address ='731, Fondren, Houston, TX’

      Gender='M’

      BirthDate='09-JAN-55‘
  - Each attribute has a *value set* (or data type) associated with it – integer, string/char, date, or enumerated type

Types of Attributes

- **Simple**
  - Each entity has a single atomic value for the attribute
  - Eg.: SSN, Gender
- **Composite**
  - The attribute may be composed of several components
    - Address (Apt#, House#, Street, City, State, ZipCode, Country)
    - Name(FirstName, MiddleName, LastName)
    - Composition may form a hierarchy where some components are themselves composite

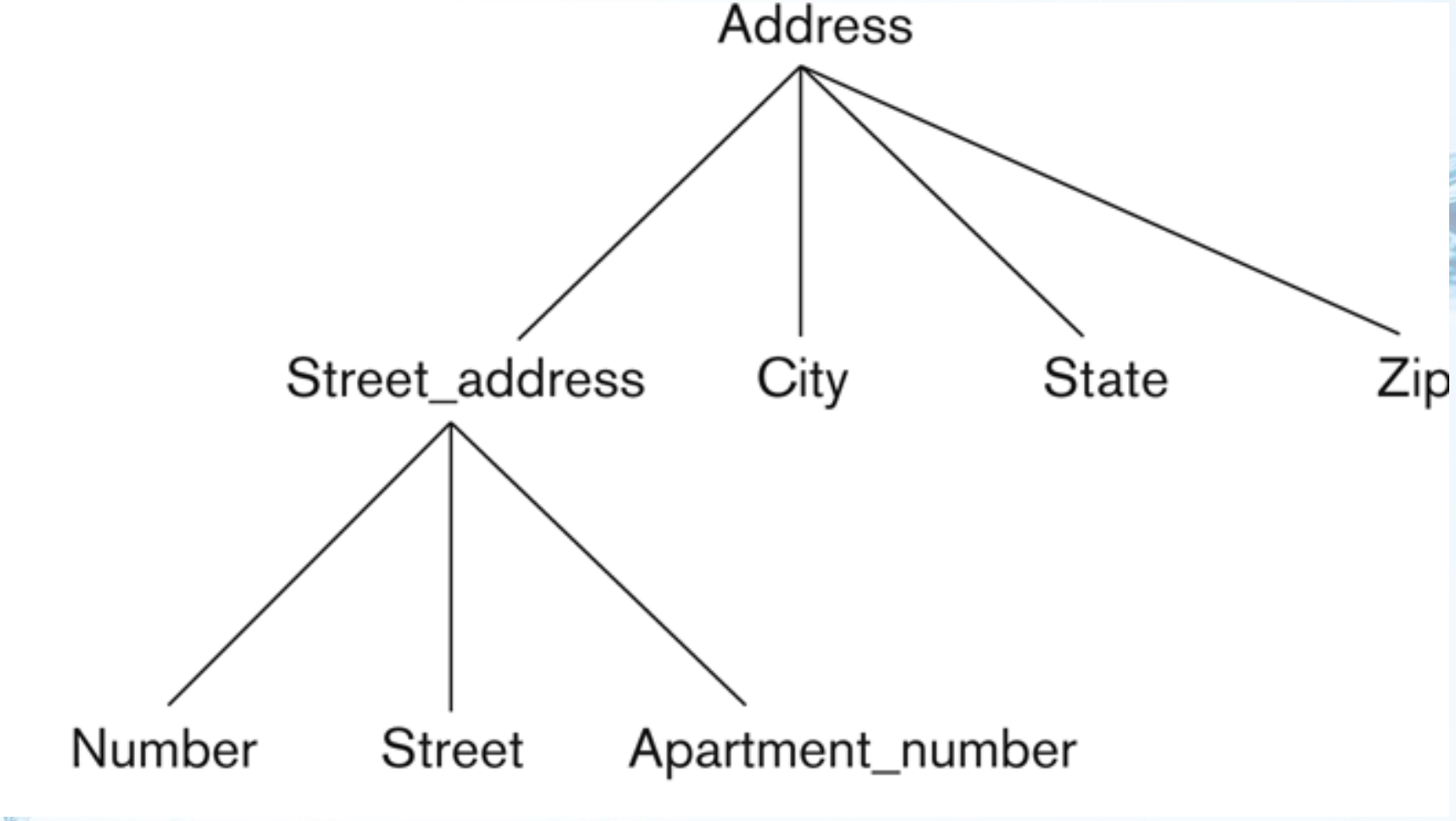
Types of Attributes

- **Multi-valued**
  - An entity may have multiple values for that attribute.
    - *Color* of a CAR
    - *Previous_Degrees* of a STUDENT
    - Denoted as {Color} or {Previous_Degrees}

Types of Attributes

- Composite and multi-valued attributes may be nested arbitrarily to any number of levels
  - Previous_Degrees of a STUDENT - composite multi-valued attribute
    - {Previous_Degrees (College, Year, Degree, Field)}
  - Multiple Previous_Degrees values can exist
  - Each has four subcomponent attributes:
    - College, Year, Degree, Field

Example of a composite attribute



Entity Types and Key Attributes

- Entities with the same basic attributes are grouped or typed into an entity type.
  - For example, the entity type EMPLOYEE and PROJECT.
- An attribute of an entity type for which each entity must have a unique value is called a key attribute of the entity type.
  - For example, SSN of EMPLOYEE.

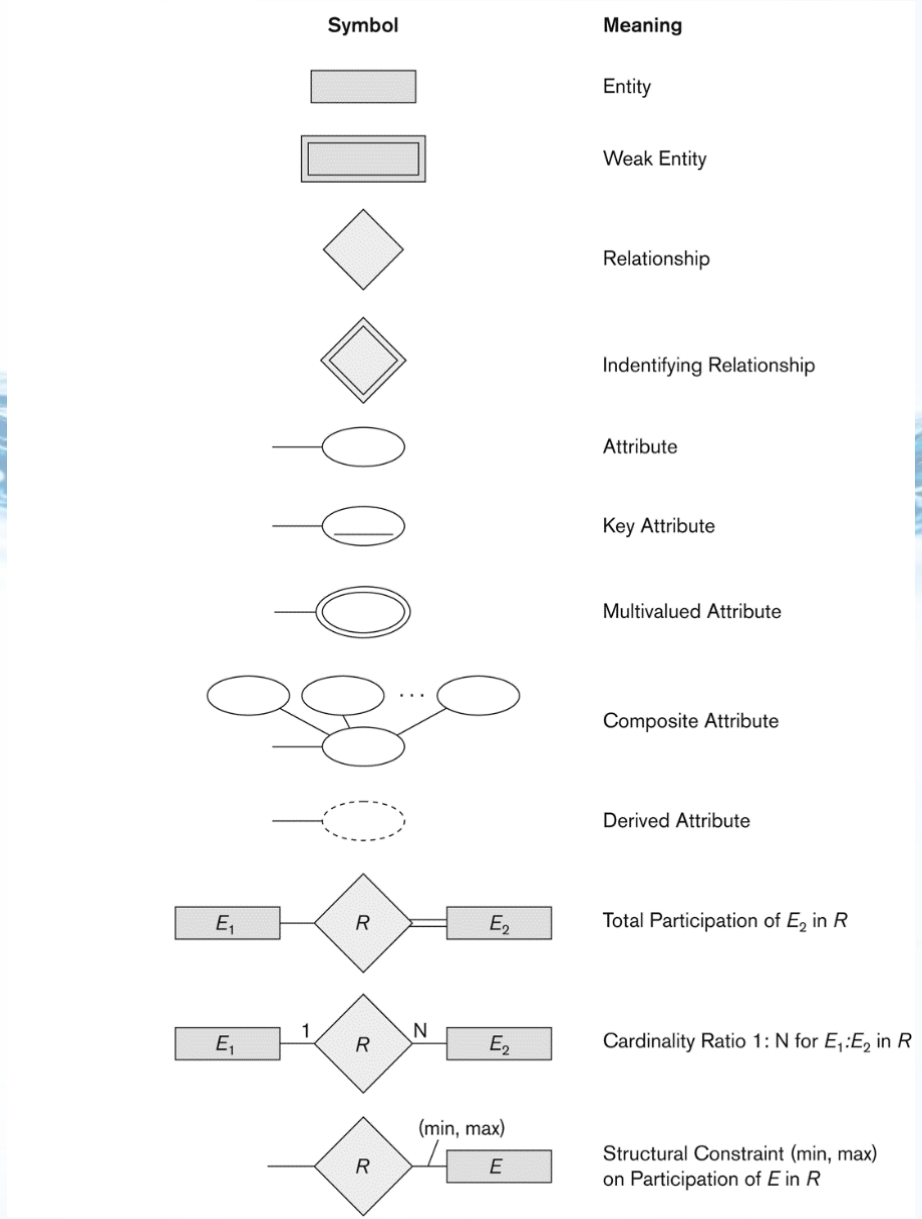
Entity Types and Key Attributes

- A key attribute may be composite.
  - VehicleTagNumber is a key of the CAR entity type with components (Number, State).
- An entity type may have more than one key.
  - The CAR entity type may have two keys:
    - VehicleIdentificationNumber (popularly called VIN)
    - VehicleTagNumber (Number, State), aka license plate number.
- Each key is <u>underlined</u>
  - This is different from the relational schema where only one “primary key” is underlined

Value Sets (Domains) of Attributes

- Each simple attribute is associated with a value set
  - ***Lastname*** has a value which is a character string of up to 30 characters
  - ***Date*** has a value consisting of MM-DD-YYYY where each letter is an integer
- A **value set** specifies the set of values associated with an attribute
  - Value sets are similar to data types in most programming languages
  - Integer, character (n), real, bit

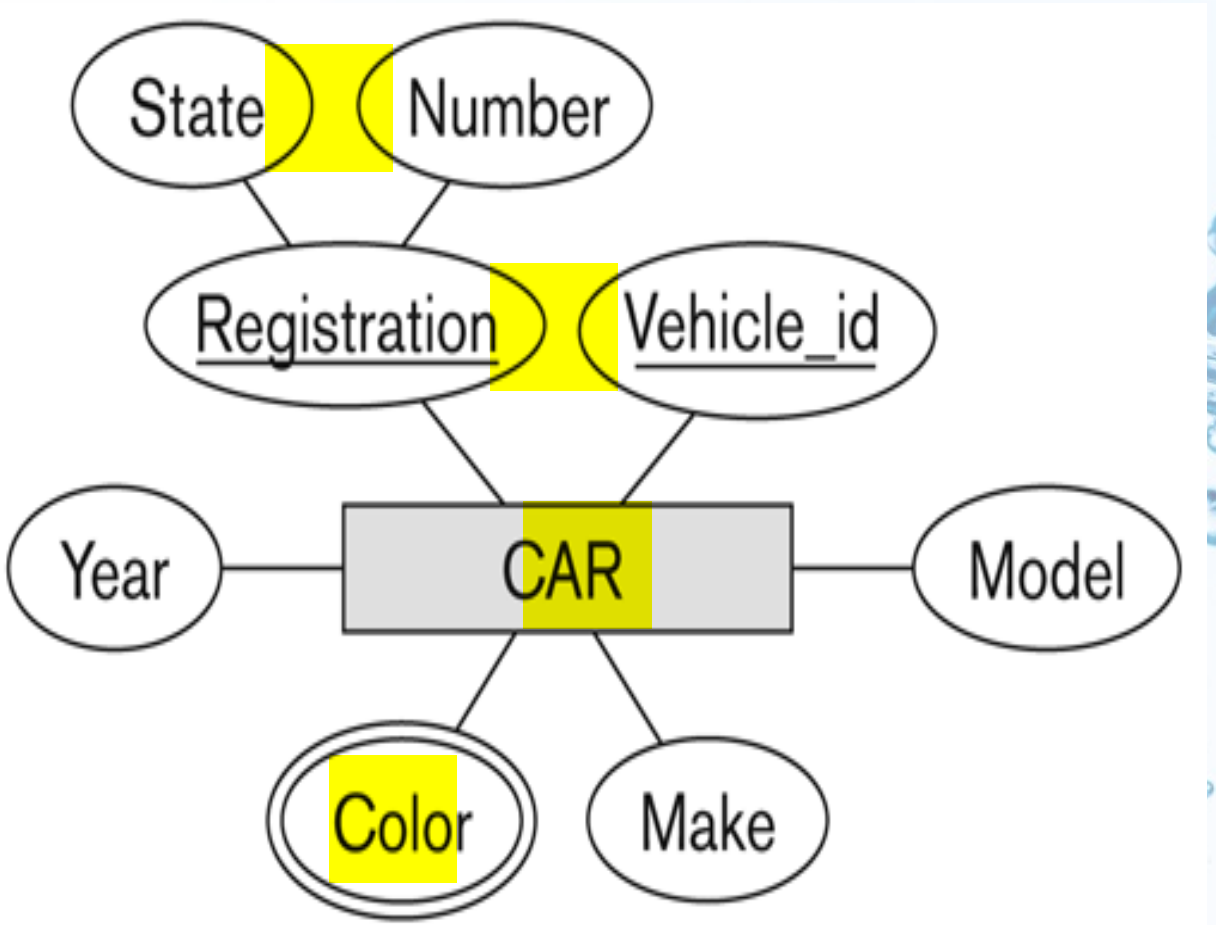
Notation for ER diagrams



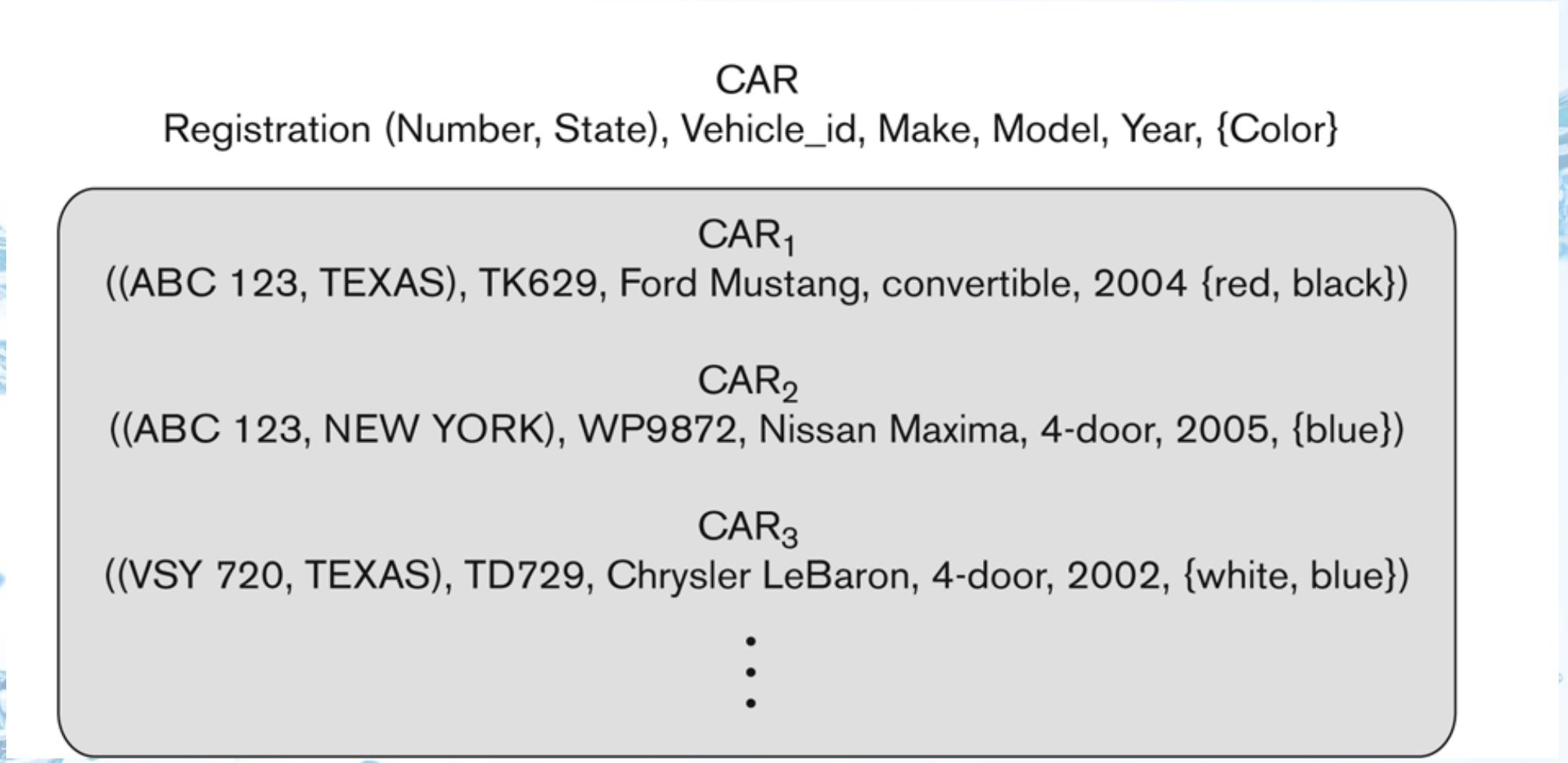
Displaying an Entity type

- In ER diagrams, an entity type is displayed in a rectangular box
- Attributes are displayed in ovals
  - Each attribute is connected to its entity type
  - Components of a composite attribute are connected to the oval representing the composite attribute
  - Each key attribute is underlined
  - Multivalued attributes displayed in double ovals

Entity Type CAR



Entity Set for CAR



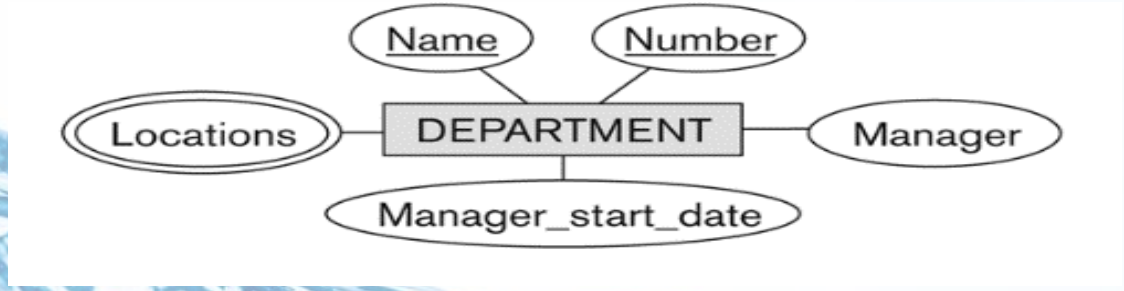
Basics for designing Databases

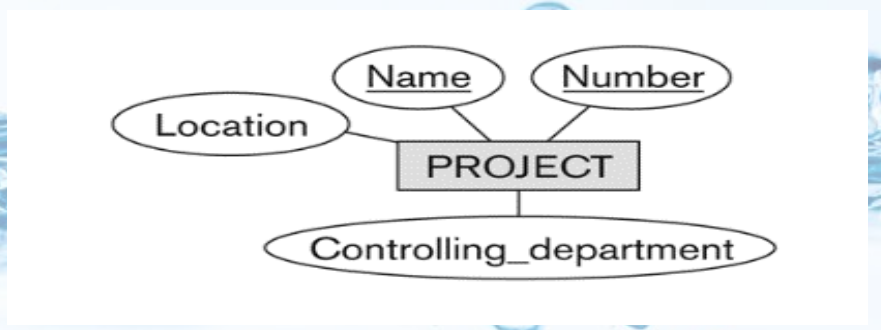
- Acquire domain knowledge
- Gather requirements
- Conceptual Schema
  - Entities
  - Attributes
    - Types of attributes
      - simple, composite, multi-valued
    - Key attributes
    - Value set
  - Data types

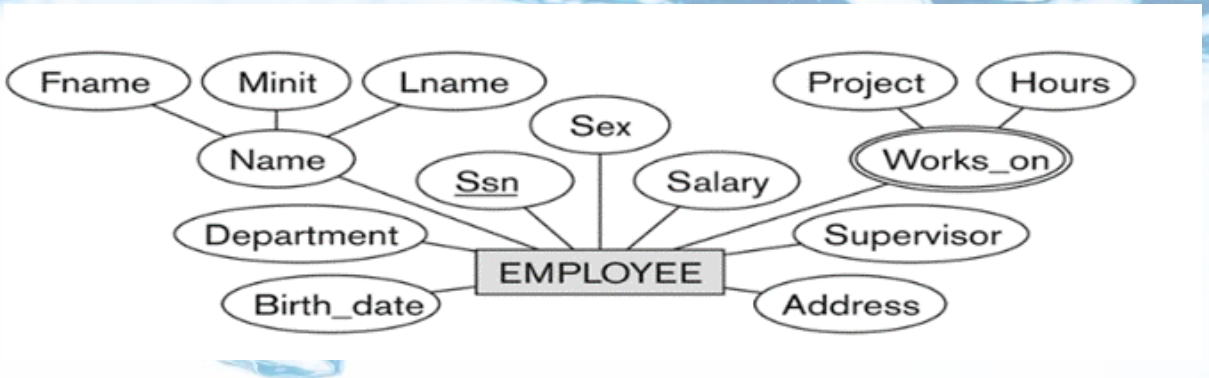
Conceptual Design of Entity Types for COMPANY

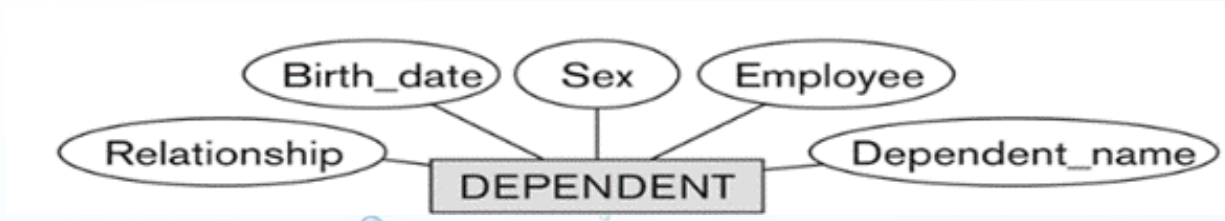
- Four initial entity types in the COMPANY database:
  - DEPARTMENT
  - PROJECT
  - EMPLOYEE
  - DEPENDENT

Initial Design of Entity Types





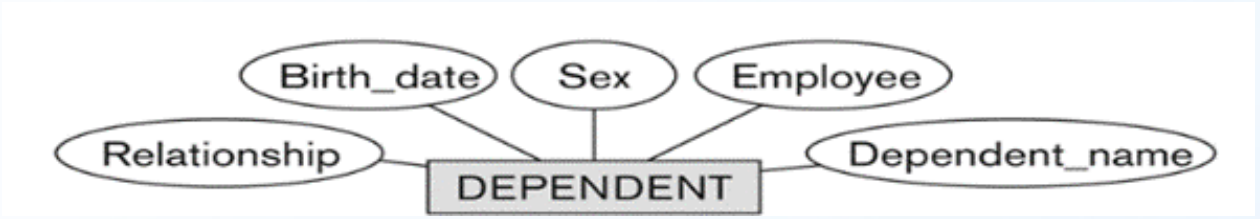




Weak Entity Types

- An entity that does not have a key attribute
  - identification-dependent on another entity type
- A weak entity must participate in an identifying relationship type with an owner or identifying entity type
- Weak entities are identified by the combination of:
  - A partial key of the weak entity type
  - The particular entity they are related to in the identifying relationship type

Weak Entity Types



- A DEPENDENT entity is identified by the dependent's first name
- The specific EMPLOYEE with whom the dependent is related
- Name of DEPENDENT is the *partial key*
- DEPENDENT is a *weak entity type*
- EMPLOYEE is its identifying entity type via the identifying relationship type DEPENDENT_OF

Refine the initial design

- The initial design is typically not complete
- Some aspects in the requirements will be represented as **relationships**
- ER model has three main concepts:
  - Entities (entity types and entity sets)
  - Attributes (simple, composite, multivalued)
  - Relationships (relationship types and relationship sets)

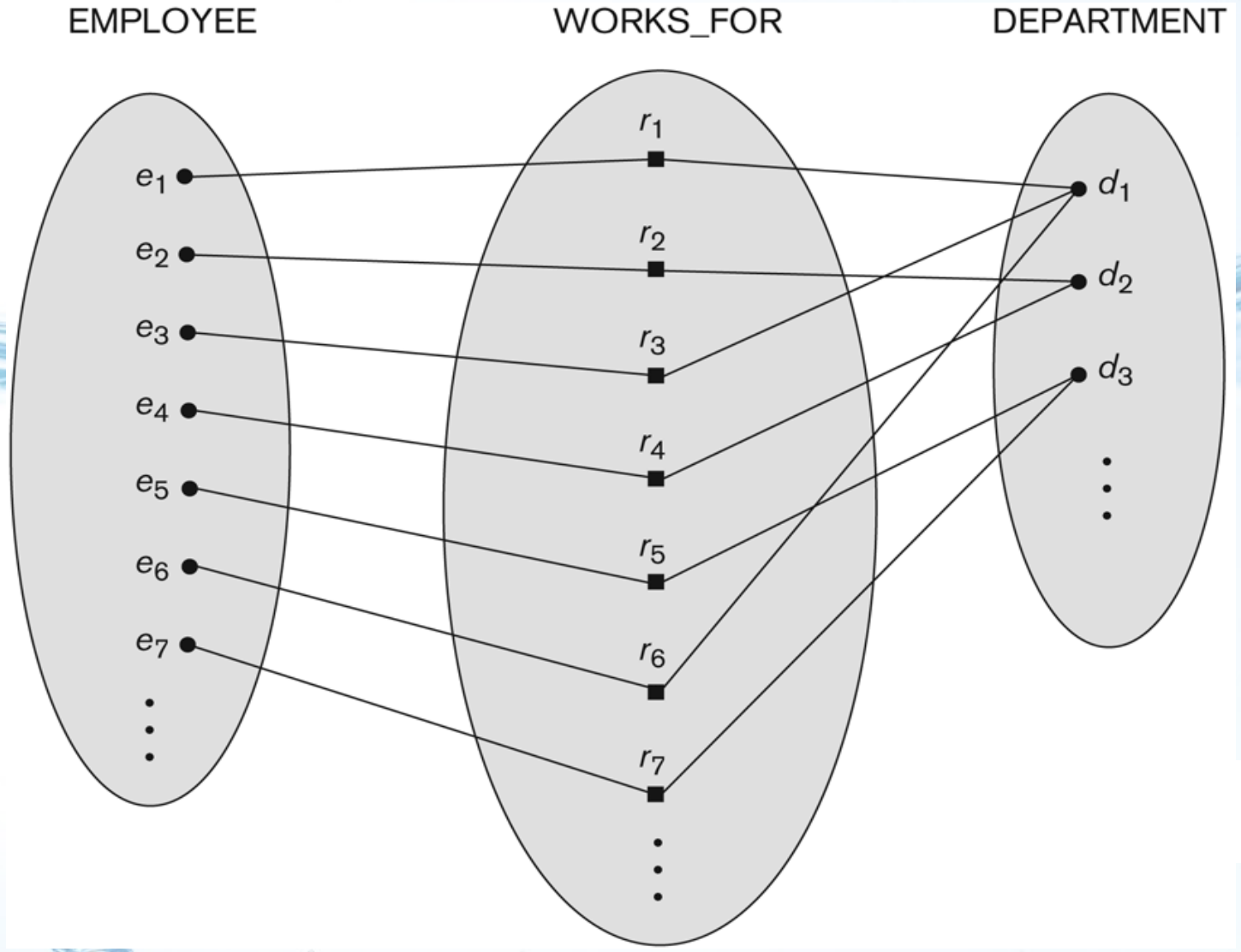
Relationships and Relationship Types

- A **relationship** relates two or more distinct entities with a specific meaning.
  - EMPLOYEE John Smith *works on* the ProductX PROJECT
  - EMPLOYEE Franklin Wong *manages* the Research DEPARTMENT.

Relationships and Relationship Types

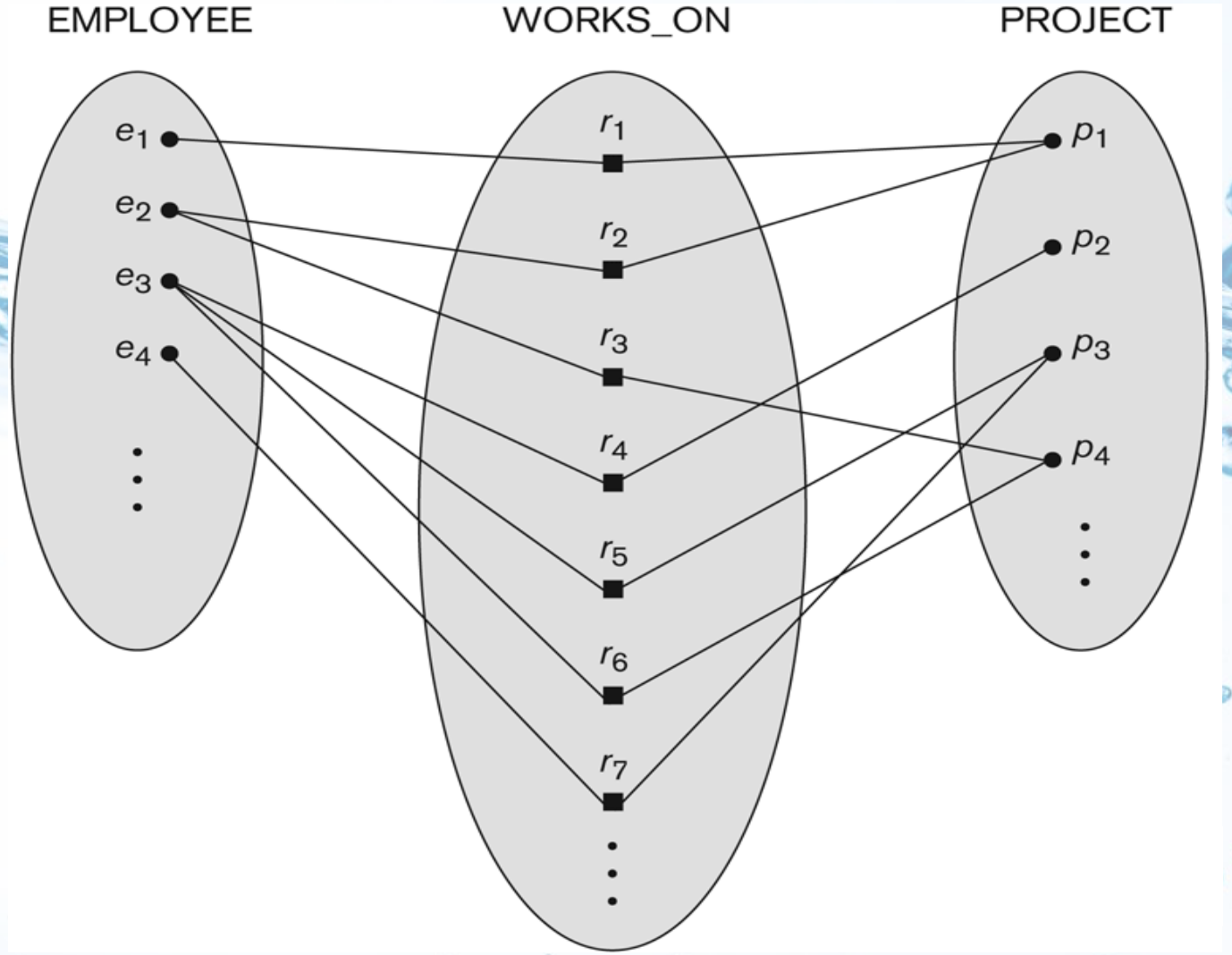
- Relationships of the same type are grouped or typed into a **relationship type**
  - WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate
  - MANAGES relationship type in which EMPLOYEEs and DEPARTMENTs participate
- The degree of a relationship type is the number of participating entity types
  - Both MANAGES and WORKS_ON are *binary* relationships

WORKS_FOR relationship



N:1

WORKS_ON relationship



M:N

Relationship type vs. relationship set

- Relationship Type:
  - Is the schema description of a relationship
  - Identifies the relationship name and the participating entity types
  - Also identifies certain relationship constraints
- Relationship Set:
  - The current set of relationship instances represented in the database
  - The current *state* of a relationship type

Relationships in COMPANY database schema

- By examining the requirements, six relationship types exist
- All are *binary* relationships (degree of 2)
- Relationships with their participating entity types:
  - WORKS_FOR (between EMPLOYEE, DEPARTMENT)
  - MANAGES (also between EMPLOYEE, DEPARTMENT)
  - CONTROLS (between DEPARTMENT, PROJECT)
  - WORKS_ON (between EMPLOYEE, PROJECT)
  - SUPERVISION (between EMPLOYEE (as subordinate), EMPLOYEE (as supervisor))
  - DEPENDENTS_OF (between EMPLOYEE, DEPENDENT)

ER Diagram – Relationship Types