

Lecture 3

DATABASE MANAGEMENT SYSTEM

COMPANY Database – 1/3

- ⦿ 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**.

COMPANY Database – 2/3

- ⦿ **EMPLOYEE's** information i.e. **social security no, address, salary, gender, and birthdate.**

- Each employee *works for one department but may work on several projects.*
- We keep track of the **number of hours** per week that an employee currently works on each project.
- We also keep track of the *direct supervisor of each employee.*

COMPANY Database – 3/3

- ⦿ Each employee may *have a number of* **DEPENDENTS**.
- ⦿ For each dependent, we keep track of their **name**, **sex**, **birthdate**, and **relationship** to the employee.

ER (Entity-Relational) Model

Concepts – 1/2

- ⦿ Popular high-level conceptual model
- ⦿ **Entities are specific objects in the mini-world that are represented in the database**
 - For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT
- ⦿ **Attributes are properties used to describe an entity**
 - For example an EMPLOYEE entity may have the attributes Name, SSN, Address, Sex, BirthDate

ER (Entity-Relational) Model

Concepts – 2/2

- A specific entity will have a **value** for each of its attributes.

For example a specific employee entity may have Name='John Smith', SSN='123456789', Address ='731, Houston TX', Sex='F', BirthDate='09-JAN-55'

Types of Attributes – 1/4

⦿ Simple

- Each entity has a single atomic value for the attribute. For example, SSN or Sex.

⦿ Composite

- The attribute may be composed of several components.

Name (FirstName, MiddleName, LastName)

- Composition may form a hierarchy where some components are themselves composite.

Types of Attributes – 2/4

⦿ **Multi-valued**

- An entity may have multiple values for that attribute.

PreviousDegrees of a STUDENT. Denoted as {PreviousDegrees}

⦿ **Derived**

- An attribute that can be computed from some other attribute.

Age can be computed from *BirthDate*.

Types of Attributes – 3

⦿ **Stored**

- An attribute from which a derived attribute can be computed.

BirthDate is the stored attribute for *Age*

Types of Attributes – 4/4

● **Complex attributes**

- In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels, although this is rare.

PreviousDegrees of a STUDENT is a composite multi-valued attribute

{PreviousDegrees (College, Year, Degree, Field)}

- Multiple PreviousDegrees values can exist
- Each has four subcomponent attributes: College, Year, Degree, Field

Entity Types and Key Attribute – 1/2

- ⦿ 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.

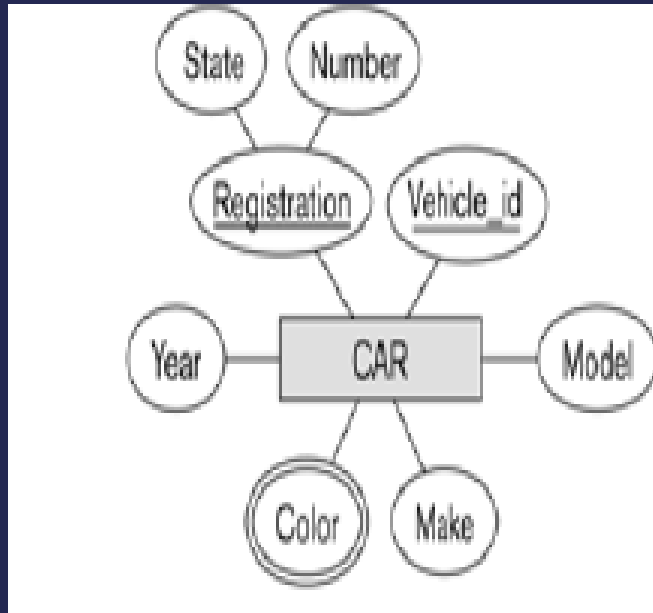
Entity Types and Key Attribute – 2/2

- ⦿ **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.

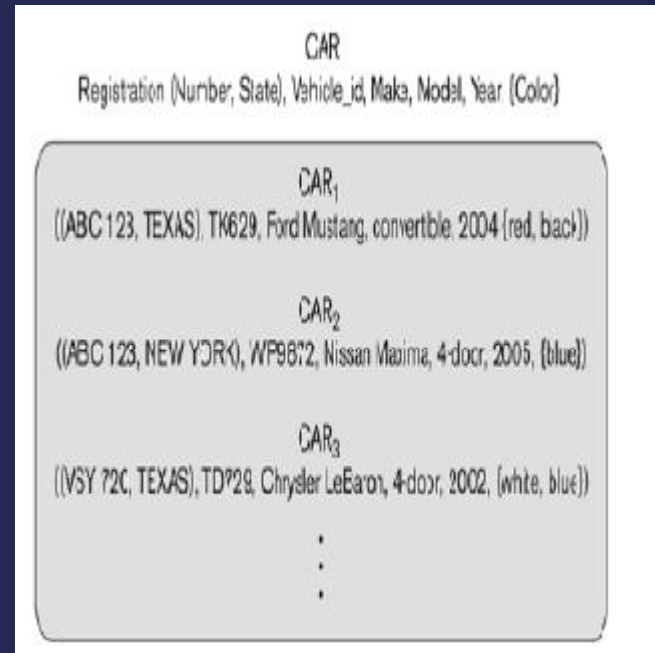
ER Diagram Notations

- ⦿ **An entity type** displayed as **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**

CAR Entity with two key attributes



ER Diagram of CAR Entity



Entity Set with three entities

Entity Type

- ⦿ Each entity type will have a **collection of entities stored in the database** called the **entity set**
- ⦿ Same name used to refer to both the entity type and the entity set
- ⦿ **Entity set is the current *state of the entities of that* type that are stored in the database**

Initial Design of Entity Types for the COMPANY Database Schema

- Based on the requirements, we can identify four initial entity types in the COMPANY database:
 - DEPARTMENT
 - Dcode, Dname, {Location}, manager (name, Startdate)
 - PROJECT
 - Project no, name, controlling department, location
 - EMPLOYEE
 - SSN, Name, address, salary, sex, birthdate, department, supervisor, {project (project no, hours worked)}
 - DEPENDENT
 - name, sex, birthdate, and relationship, employee to which related.

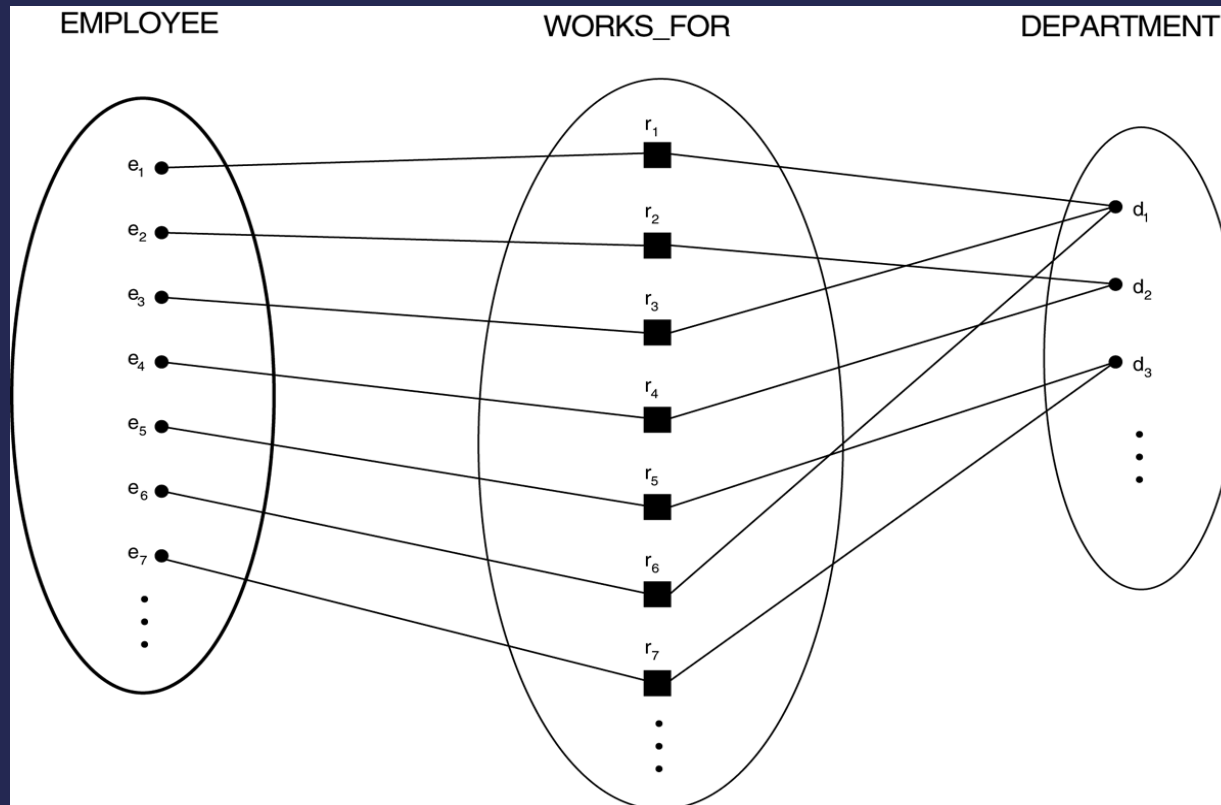
Relationship and Relationship Type – 1/2

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

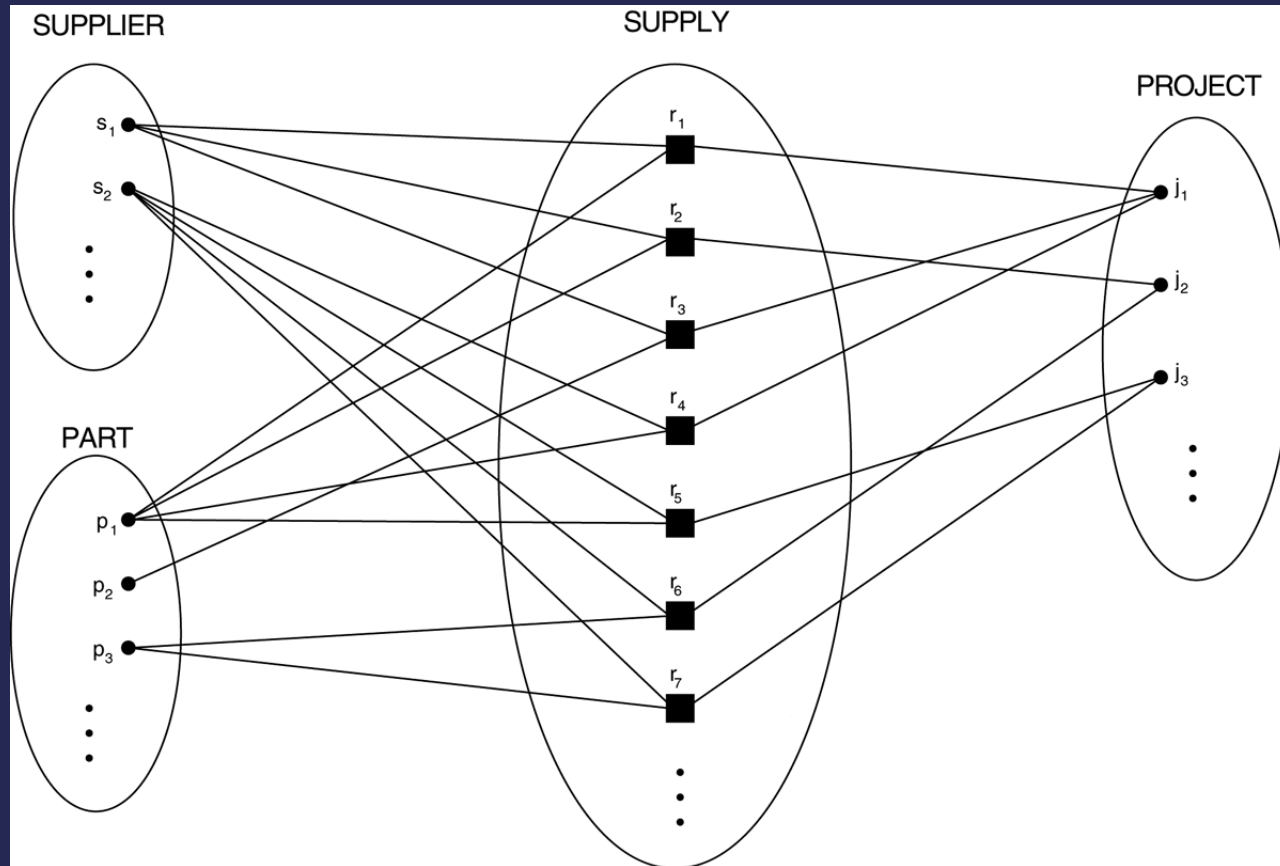
Relationship and Relationship Type – 2/2

- Relationships of the same type are grouped or typed into a **relationship type**
 - For example, the WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate, or the 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*.

Binary Relationship



Relationship of degree 3



Relationship Type and Set – 1/2

⦿ 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
- **Current *state of a relationship type***

Relationship Type and Set – 2/2

- ⦿ Each instance in the set relates individual participating entities – one from each participating entity type
- ⦿ In ER diagrams, we represent the *relationship type* as follows:
 - **Diamond-shaped box** is used to display a **relationship type**
 - Connected to the participating entity types via straight lines

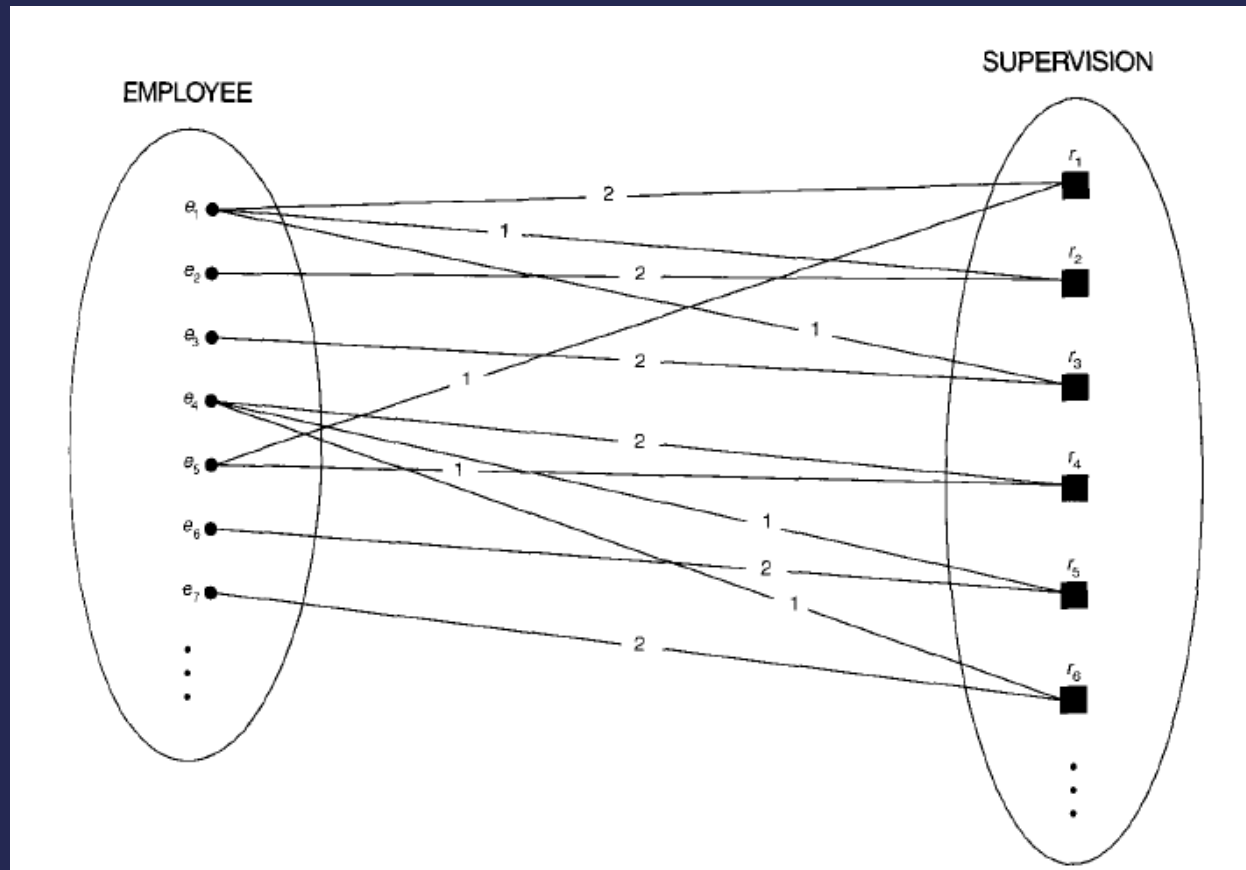
Recursive Relationships – 1/3

- ⦿ A relationship type with the same participating entity type in **distinct roles**
 - Example: the SUPERVISION relationship
 - EMPLOYEE participates twice in two distinct roles: supervisor (or boss) role and supervisee (or subordinate) role
- ⦿ Each relationship instance relates two distinct EMPLOYEE entities:
 - One employee in *supervisor* role
 - One employee in *supervisee* role

Recursive Relationships – 2

- In a **recursive relationship type**.
 - Both **participations** are **same entity** type in **different roles**.
 - For example, SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker).
- In ER diagram, need to display ***role names*** to distinguish participations.

Recursive Relationships – 3



A recursive relationship **SUPERVISION** between **EMPLOYEE** in the *supervisor* role (1) and **EMPLOYEE** in the *subordinate* role (2).

Refining the COMPANY database schema by introducing relationships – 1/2

- ⦿ By examining the requirements, six relationship types are Identified
 - All are *binary* relationships(degree 2)
- ⦿ Listed below 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)

Refining the COMPANY database schema by introducing relationships – 2/2

- ⦿ In the refined design, some attributes from the initial entity types are refined into relationships:
 - Manager of DEPARTMENT -> MANAGES
 - Works_on of EMPLOYEE -> WORKS_ON
 - Department of EMPLOYEE -> WORKS_FOR
- ⦿ In general, more than one relationship type can exist between the same participating entity types
 - MANAGES and WORKS_FOR are distinct relationship types between EMPLOYEE and DEPARTMENT
 - Different meanings and different relationship instances.