Lecture 5

Relational Database Design Using

ER-to-Relational Schema Mapping

Dr. Vandana Kushwaha

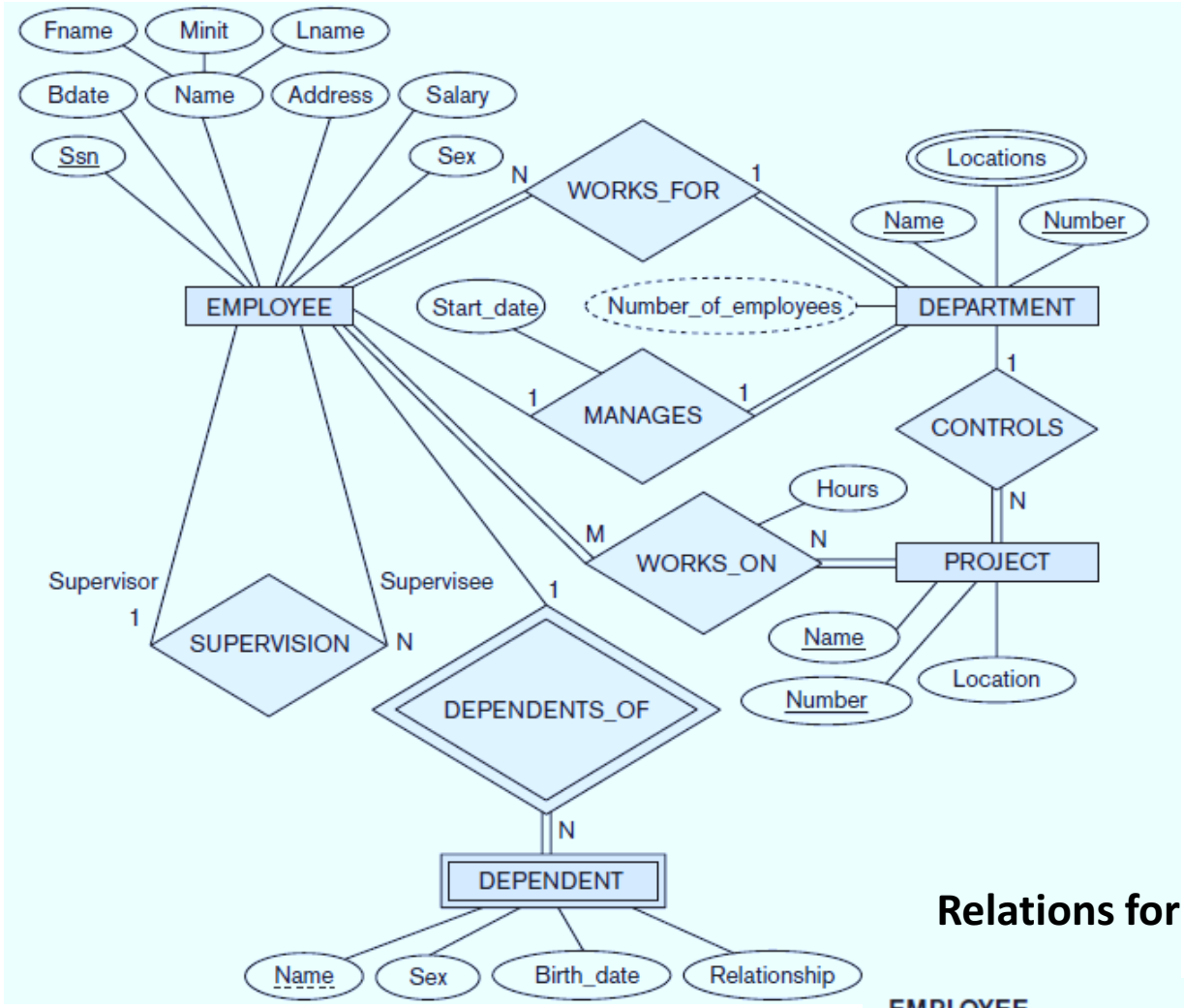
Department of Computer Science

Institute of Science, BHU, Varanasi

Mapping of Entity Types

Mapping of Regular Entity Types

- For each **Regular (strong) entity type *E*** in the **ER schema**, create a **relation *R*** that includes **all the simple attributes** of ***E*.**
- Include only the **simple component attributes** of a **composite attribute.**
- Choose **one** of the **key attributes** of ***E*** as the **primary key** for ***R*.**
- If the chosen **key** of ***E*** is a **composite**, then the set of simple attributes that form it will together form the **primary key** of *R*.
- If **multiple keys** were **identified** for ***E*** during the **conceptual design**, the information describing the attributes that form each additional key is kept in order to specify **secondary (unique) keys** of **relation *R*.**
- **Example**: We create the relations **EMPLOYEE, DEPARTMENT**, and **PROJECT** correspond to the regular entity types EMPLOYEE, DEPARTMENT, and PROJECT.



Relations for Strong Entity Sets

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

Mapping of Weak Entity Types

- For each **Weak entity type *W*** in the **ER schema** with **owner entity type *E***, create a **relation *R*** and include all attributes of ***W*** as attributes of ***R*.**
- In addition, **include** the **primary key attribute(s)** of the **owner entity type(s)** as **foreign key attributes** of ***R*.**
- The **primary key** of ***R*** is the **combination of the primary key(s)** of the **owner(s)** and the **partial key** of the **weak entity type *W*.**
- **Example:** Create the **relation DEPENDENT** in this step to correspond to the **weak entity type DEPENDENT.**
- We include the **primary key Ssn** of the **EMPLOYEE relation**—which corresponds to the **owner entity type**—as a **foreign key attribute** of **DEPENDENT.**
- The **primary key** of the **DEPENDENT relation** is the **combination {Essn, Dependent_name},** because **Dependent_name** is the **partial key** of **DEPENDENT.**

DEPENDENT

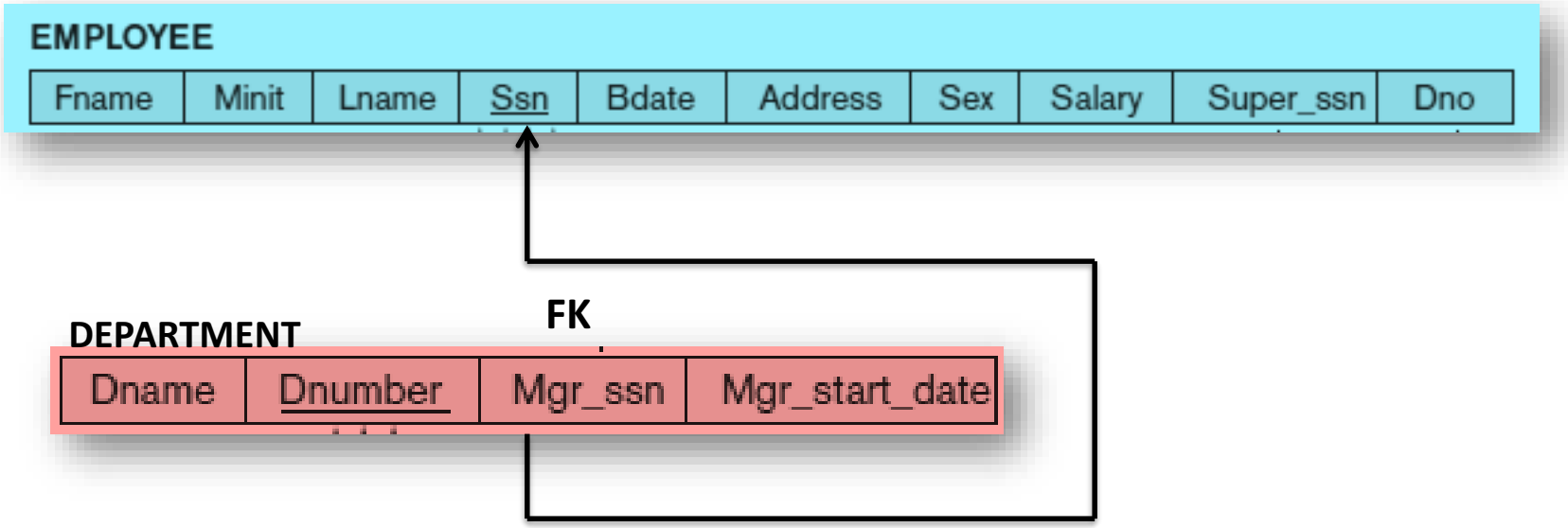
Essn	Dependent_name	Sex	Bdate	Relationship

Mapping of Relationship Types

Mapping of Binary 1:1 Relationship

- For each **Binary 1:1 relationship type *R*** in the **ER schema**, identify the **relations *S* and *T*** that correspond to the **entity types participating** in ***R*.**
- Choose **one** of the **relations- *S***, and **include as a foreign key** in ***S*** the **primary key of *T*.**
- It is better to **choose an entity type** with ***total participation*** in ***R*** in the **role of *S*.**
- Include **all the attributes** of the **1:1 relationship type *R*** as **attributes** of ***S*.**
- **Example**: Map the **1:1 relationship type MANAGES** by choosing the participating entity type **DEPARTMENT** to serve in the **role of *S*** because its **participation in the MANAGES** relationship type is **total** (every department has a manager).
- **Include** the **primary key** of the **EMPLOYEE relation** as **foreign key** in the **DEPARTMENT relation** and **rename** it **Mgr_ssn**.
- Also **include** the **simple attribute Start_date** of the **MANAGES relationship** type in the **DEPARTMENT relation** and rename it **Mgr_start_date.**

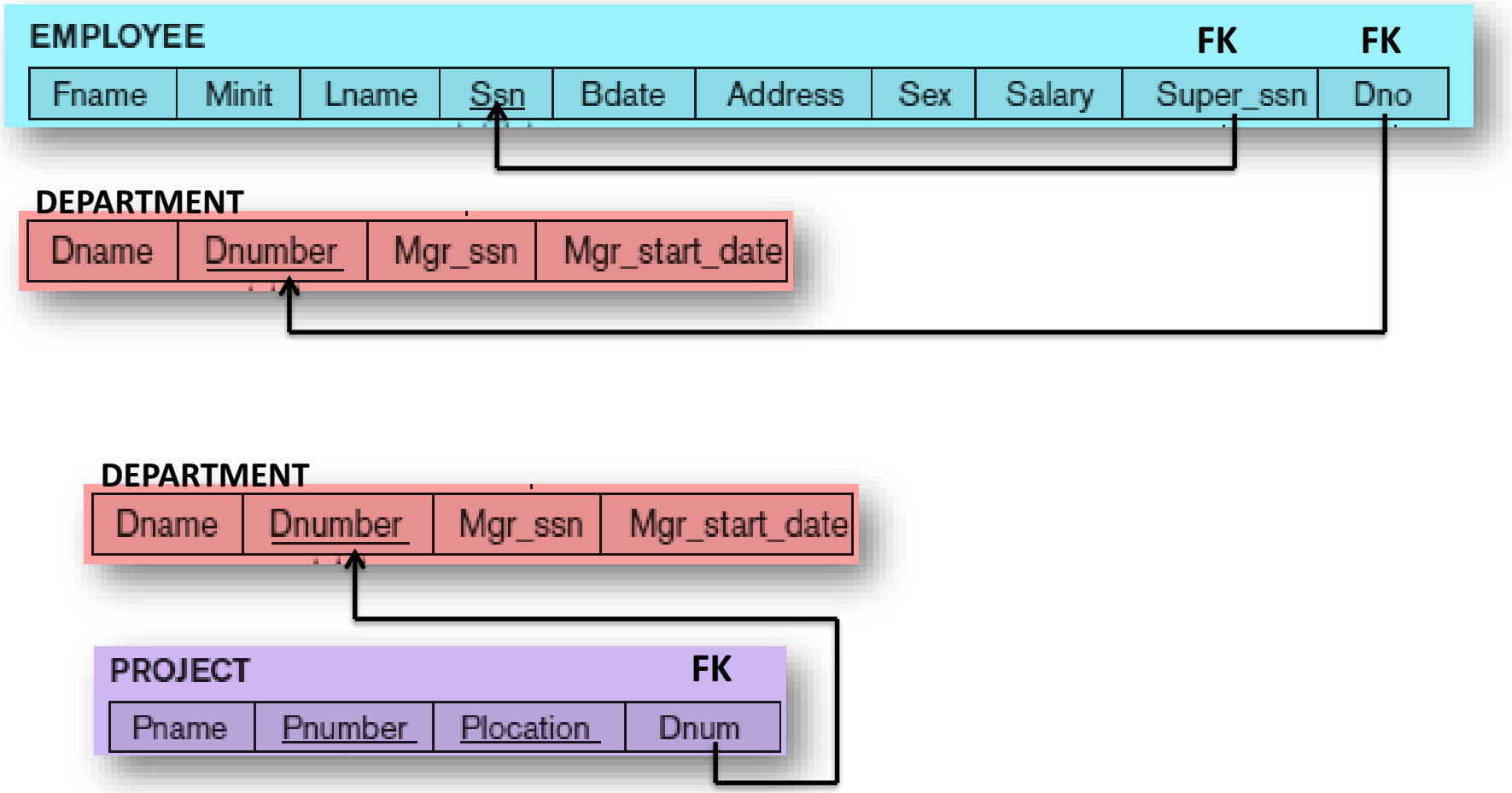
Mapping of Binary 1:1 Relationship



Mapping of Binary 1:N Relationship

- For each regular **Binary 1:N relationship type *R***, identify the **relation *S*** that represents the **participating entity type** at the ***N-side*** of the **relationship type R.**
- Include as **foreign key** in ***S*** the **primary key** of the **relation *T*** that represents the other **entity type** participating in ***R*.**
- **Include** any **attributes** of the **1:N relationship type** as **attributes of *S*.**
- **Example:** Map the **1:N relationship** types **WORKS_FOR, CONTROLS**, and **SUPERVISION** .
- For **WORKS_FOR** we include the **primary key Dnumber** of the **DEPARTMENT relation** as **foreign key** in the **EMPLOYEE relation** and call it **Dno.**
- For **SUPERVISION** we include the **primary key** of the **EMPLOYEE relation** as **foreign key** in the **EMPLOYEE relation itself**—because the relationship is **recursive**—and call it **Super_ssn.**
- The **CONTROLS relationship** is mapped to the **foreign key attribute Dnum** of **PROJECT**, which references the **primary key Dnumber** of the **DEPARTMENT** relation.

Mapping of Binary 1:N Relationship

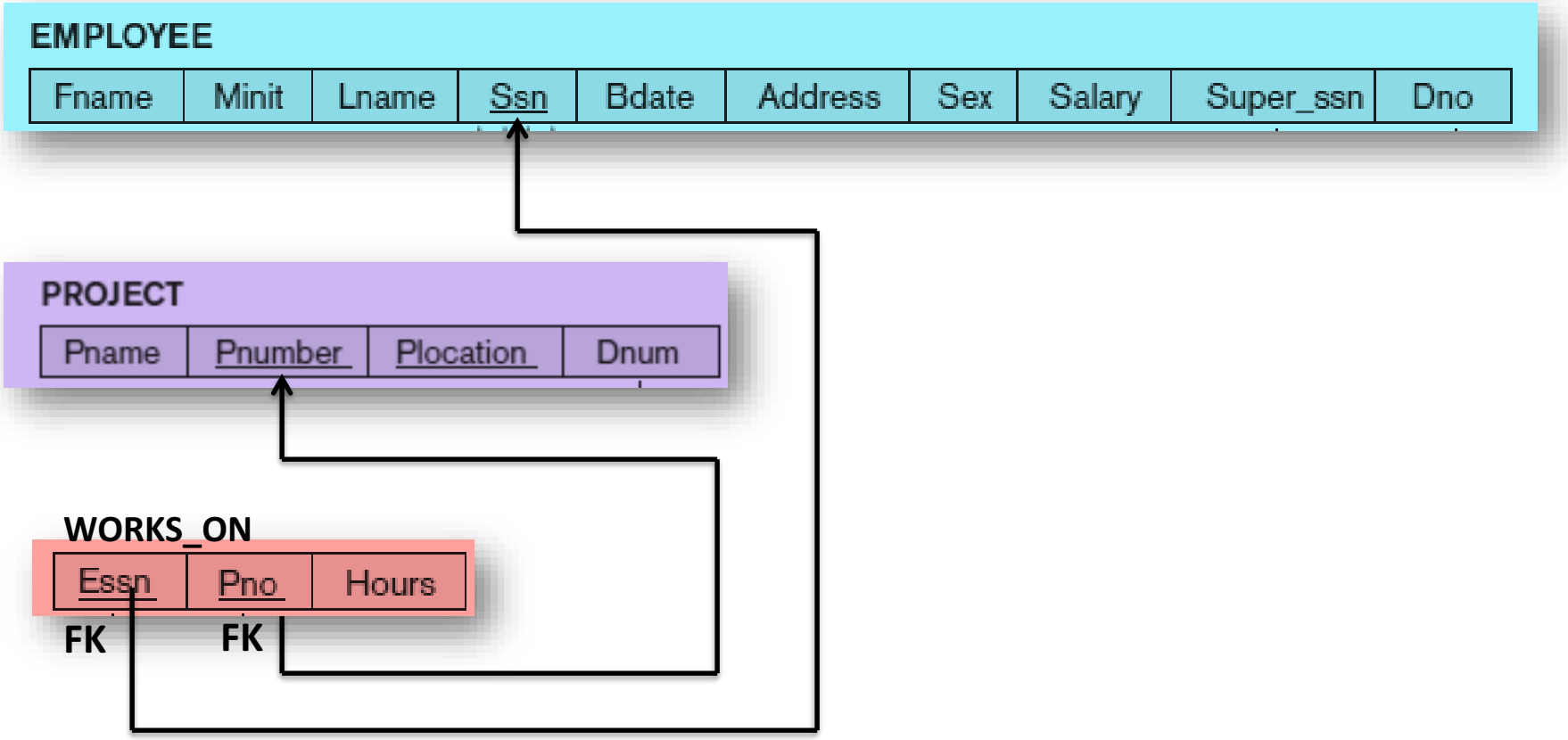


Mapping of Binary M:N Relationship

- For each **binary M:N relationship type *R***, create a **new relation *S*** to **represent *R*.**
- Include as **foreign key attributes in *S*** the **primary keys** of the **relations** that represent the **participating entity types**; their ***combination*** will form the **primary key of *S*.**
- Also **include any attributes** of the **M:N relationship type** as **attributes of *S*.**
- **Example:** Map the **M:N relationship type WORKS_ON** by creating the **relation WORKS_ON.**
- Include the **primary keys** of the **PROJECT** and **EMPLOYEE relations** as **foreign keys in WORKS_ON** and rename them Pno and Essn, respectively.
- **Pno** and **Essn** jointly form the **primary key** of relationship type **WORKS_ON**

Mapping of Binary M:N Relationship

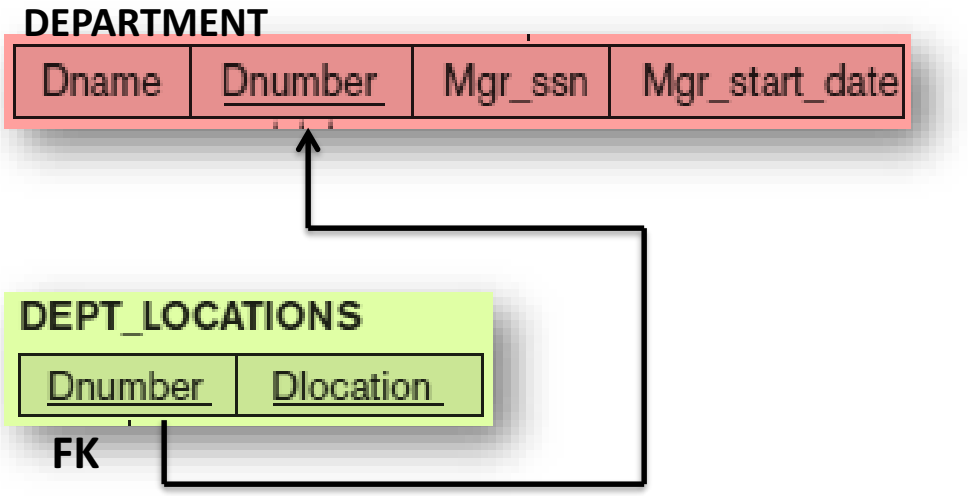
- Also **include an attribute Hours** in **WORKS_ON** to represent the **Hours attribute** of the **relationship type.**
- The **primary key** of the **WORKS_ON relation** is the **combination** of the **foreign key attributes {Essn, Pno}.**



Mapping of Multivalued Attributes

- For each **Multivalued attribute *A*, create** a **new relation *R*.**
- This **relation *R*** will include an **attribute** corresponding to ***A***, plus the **primary key attribute *K***—as a **foreign key** in ***R***—of the **relation** that has ***A* as a multivalued attribute.**
- The **primary key of *R*** is the **combination of *A*** and ***K***.
- If the **multivalued attribute** is **composite**, we include its **simple components**.
- **Example:** Create a **relation DEPT_LOCATIONS.**
- The **attribute Dlocation** represents the **multivalued attribute** LOCATIONS of DEPARTMENT, while **Dnumber**—as **foreign key**—represents the **primary key** of the **DEPARTMENT relation.**
- The **primary key** of **DEPT_LOCATIONS** is the **combination of {Dnumber, Dlocation}.**

Mapping of Multivalued Attributes



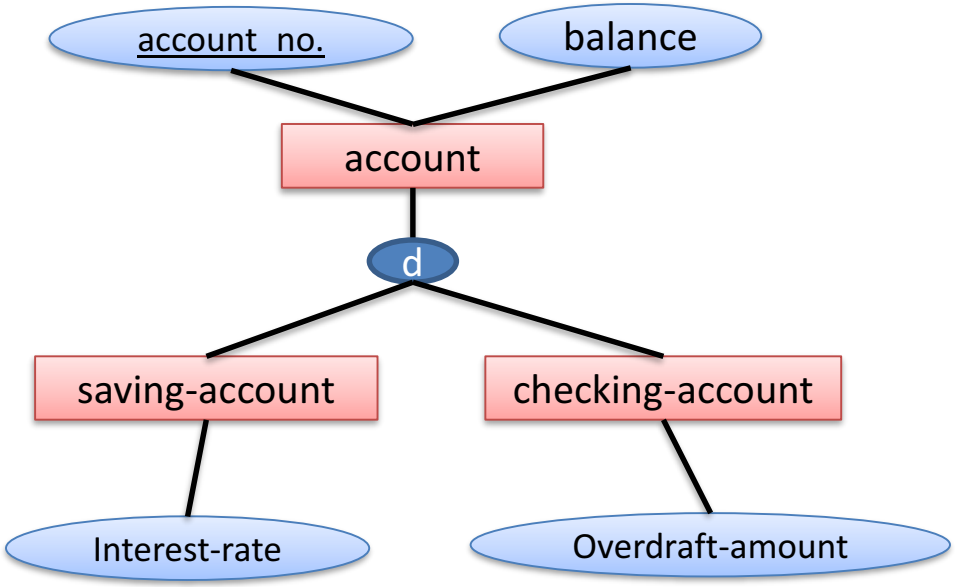
Mapping of Composite Attributes

- We handle **Composite attributes** by **creating** a **separate attribute** for **each of the component attributes.**
- We do **not create** a **separate column** for the **composite attribute itself.**
- **Example:**
- ***Address*** is a **composite attribute** of **entity set *employee***, and the **components of *address*** are ***street*** and ***city*.**
- The table generated from **employee** would then **contain columns *address-street*** and ***address-city*; there is no separate column for *address*.**

Mapping EER Model Constructs to Relations

Mapping of Specialization or Generalization

- There are **two different methods** for transforming to a tabular form an **E-R diagram** that includes **Generalization/Specialization.**
- Consider the following **Generalization** of ***saving-account*** and ***checking-account*** **entity type** into a higher level ***account*** **entity type**.



Method 1:

- Create a **table** for the **higher-level entity set.**
- For each **lower-level entity set**, create a **table** that includes a column for each of the attributes of that entity set **plus** a column for each attribute of the **primary key** of the **higher-level entity set.**
- Thus, for the **E-R diagram**, we have **three tables:**
  - ***account(account-number, balance)***
  - ***savings-account(account-number, interest-rate)***
  - ***checking-account(account-number , overdraft-amount)***

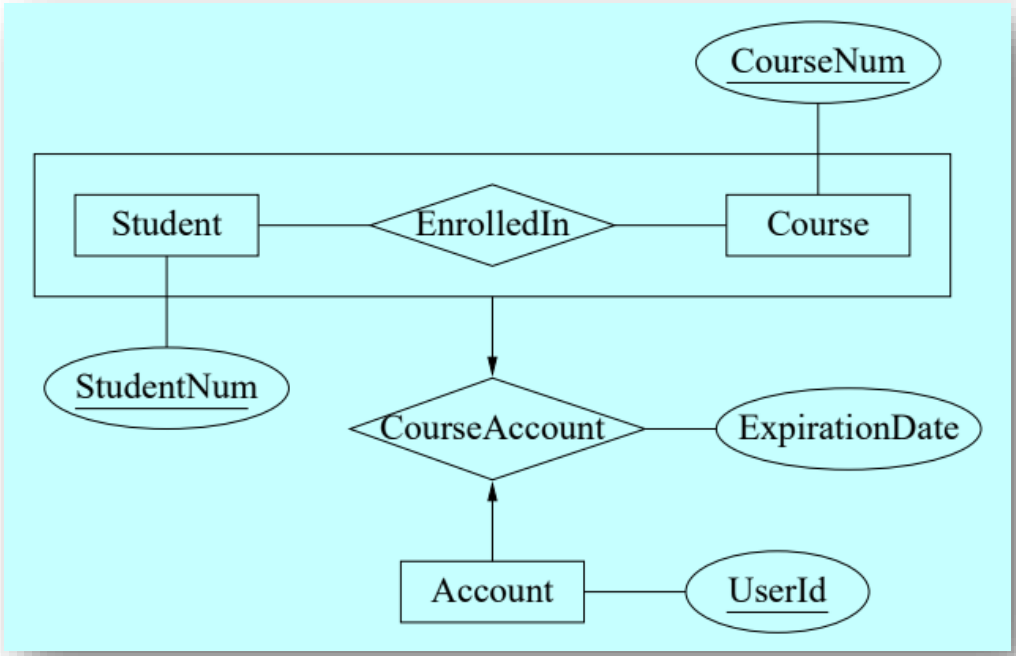
Method 2:

- An **alternative representation** is possible, if the **Generalization** is **disjoint and complete.**
- That is, if no entity is a member of two lower-level entity sets directly below a higher-level entity set, and
- If every entity in the higher level entity set is also a member of one of the lower-level entity sets.
- Here, **do not create a table** for the **higher-level entity set.**
- Instead, for **each lower-level entity set**, create a **table** that includes a column for each of the attributes of that entity set plus a column for ***each*** **attribute** of the **higher-level entity set.**

Method 2:

- Then, for the **E-R diagram**, we have **two tables:**
  - ***savings-account(account-number, balance, interest-rate)***
  - ***checking-account(account-number, balance, overdraftamount)***
- The ***savings-account*** and ***checking-account*** relations corresponding to these tables both have ***account-number*** as the **primary key.**

Relation mapping for Aggregation



Student

StudentNum

Course

CourseNum

Account

UserId

EnrolledIn

StudentNum	CourseNum

CourseAccount

UserId	StudentNum	CourseNum	ExpirationDate