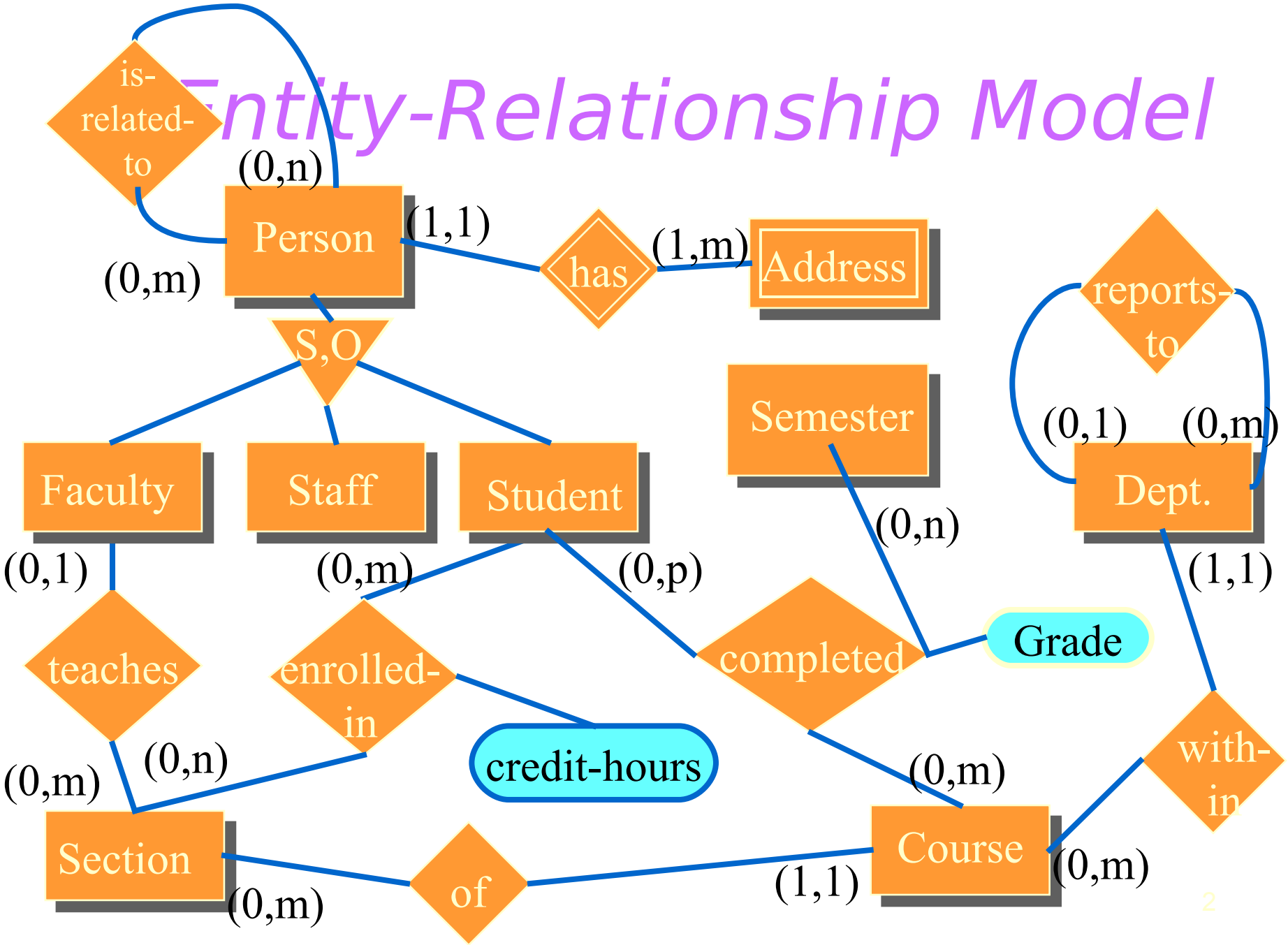
ER to Relational Conversion

Northern Illinois University

September 2000

Entity-Relationship Model



Entity-Relationship Model

ENTITIES

- Person
  - SSN (Identifier)
  - Name
  - Birth-Date
  - Beginning Date
- Address
  - Type (discriminator)
  - Street
  - City
  - State
  - Zip
- Faculty
  - SSN (Identifier)
  - Contact hours
  - Tenure status
- Staff
  - SSN (Identifier)
  - Position
- Student
  - SSN (Identifier)
  - Overall GPA
  - Major

ENTITIES

Entity-Relationship Model

- Dept.
  - Dept-Code (ID)
  - Dept-Name
  - Dept-Address
  - Dept-Chair
- Course
  - Crse-Code (ID)
  - Crse-Title
  - Crse-Max-Credit-Hours
  - Crse-Var-Hours-Code
  - Crse-Fee
- Section
  - Sect-Code (ID)
  - Sect-Credit-Hours
  - Sect-Meet-Time
  - Sect-Meet-Day
- Semester
  - Sem-Yr (ID)
  - Sem-Session (ID)

Entity-Relationship Model

RELATIONSHIPS with attributes

- Student enrolled-in Section
  - Credit-hours
    - In a variable credit section this attribute would be used to hold the credit hours for which a specific student is enrolled.
- Completed
  - Grade
    - A student is allowed to take a course more than once.

ER to Relational Conversion

1. Consider all strong entities not subtypes (do not consider “date” entities here)
2. Consider sub-type entities
   - two methods
3. Consider weak entities
4. Consider One-to-many binary relationships

ER to Relational Conversion

5 Consider many-to-many binary relationships

6 Consider relationships greater than binary (other than those involving “date” entities)

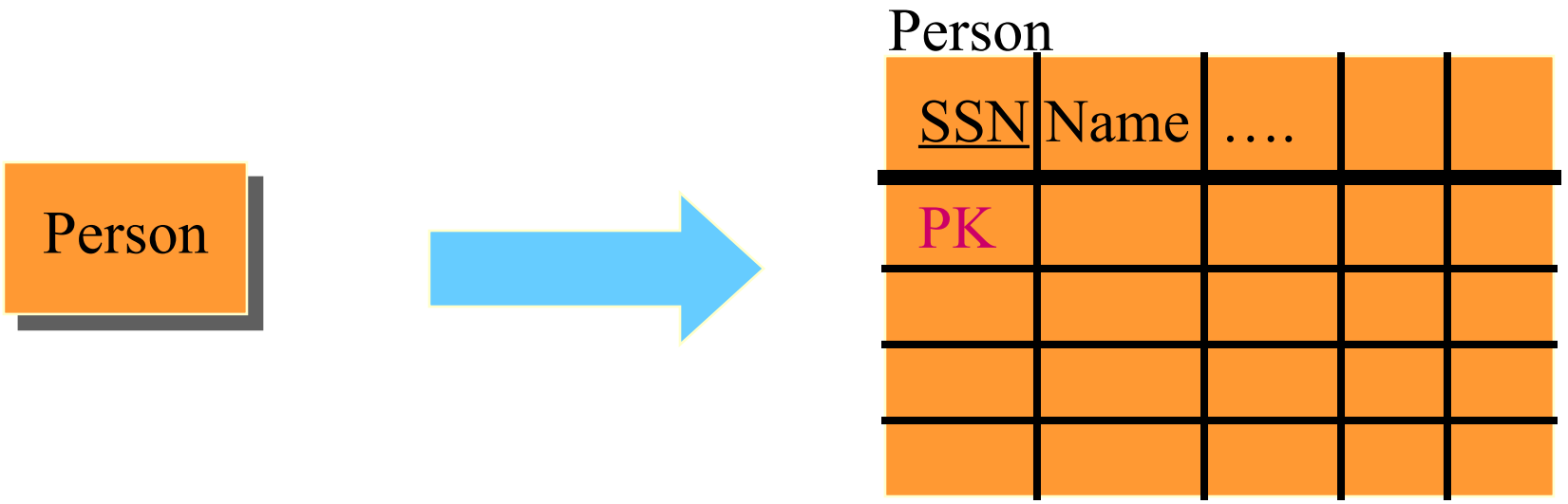
7 Consider relationships greater than binary involving a “date” entity

8 Consider recursive relationships

Consider All Strong Entities not Subtypes

- create a new relation
- name of the relation is the name of the entity
- attributes of entity become attributes of relation
- primary key of relation is entity identifier

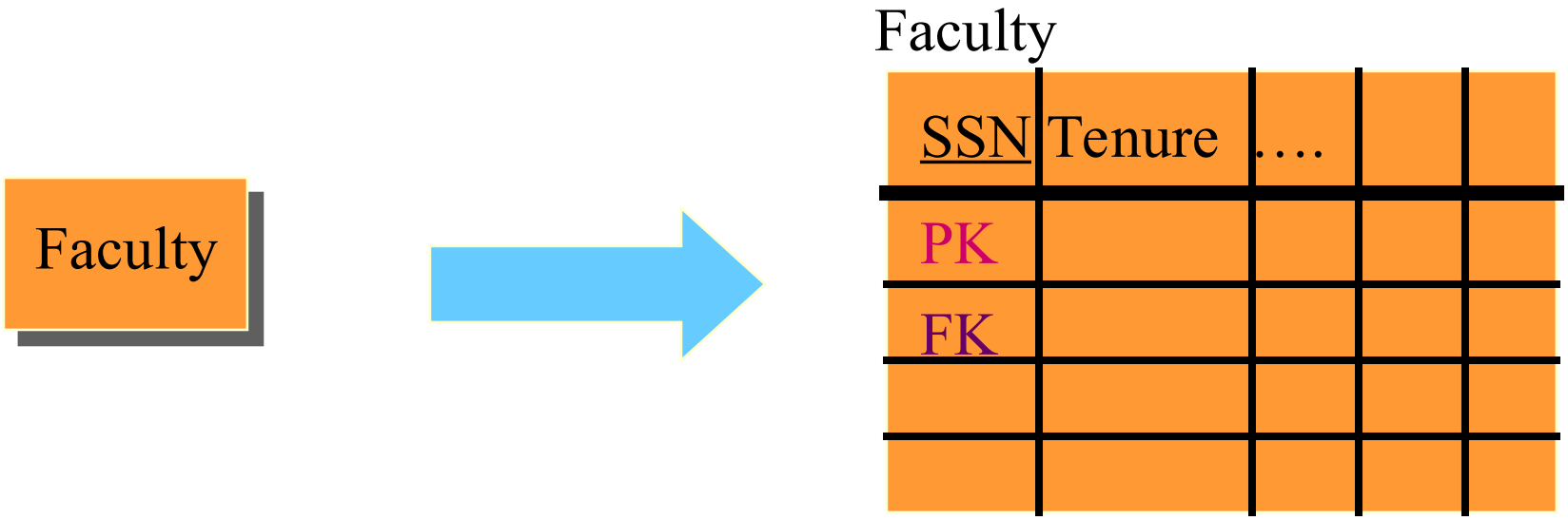
Consider All Strong Entities not Subtypes



Consider Sub-type Entities (First Method)

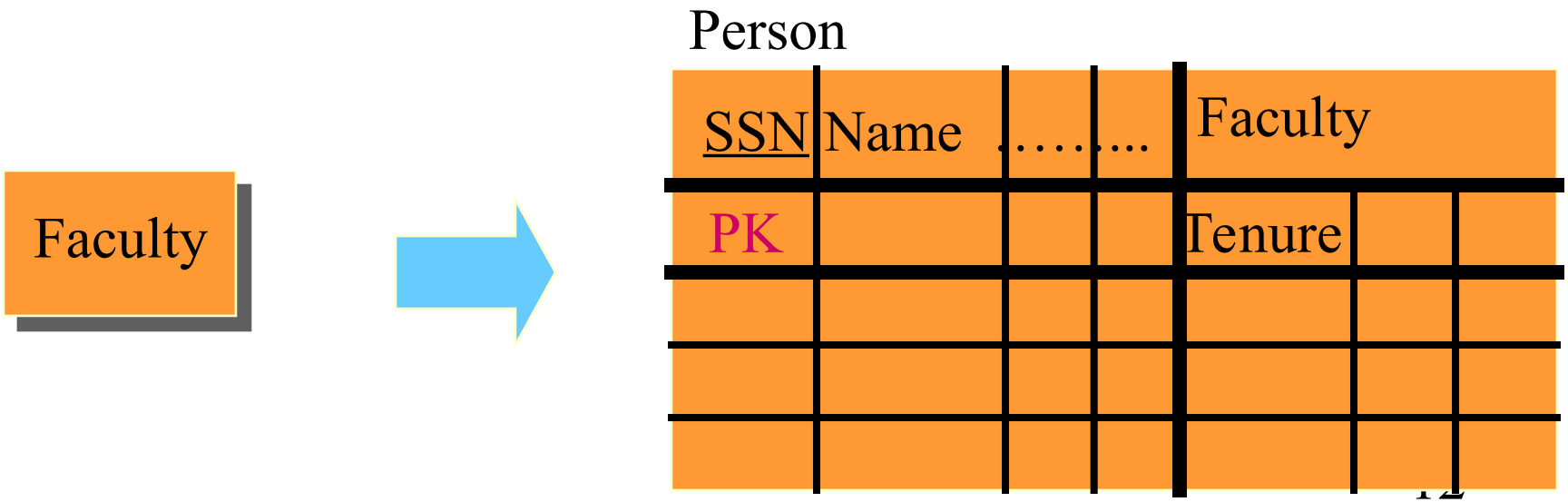
- treat as a strong entity
- primary key is the entity identifier
- primary key is also a foreign key referencing the relation created from the supertype entity

Consider Sub-type Entities (First Method)



Consider Sub-type Entities (Second Method)

- combine into the relation created from the supertype entity as a composite attribute



Consider Sub-type Entities

- may combine the two methods within the conversion of the sub-types of a single ISA

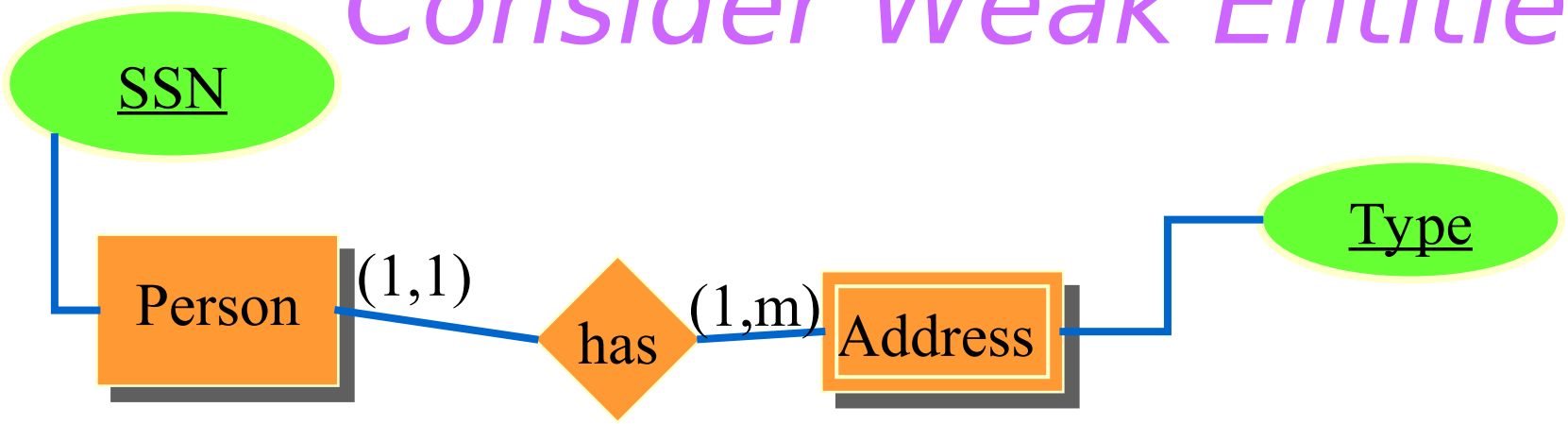
Consider Weak Entities

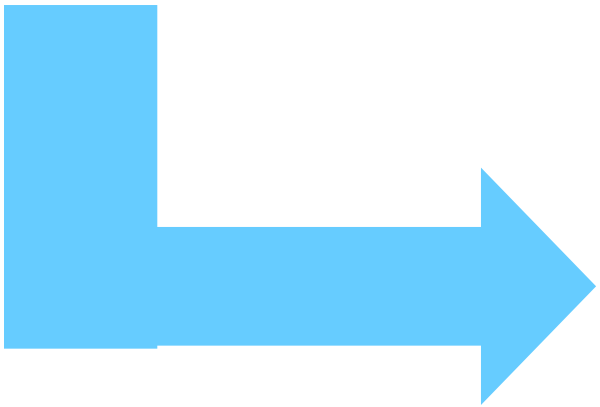
- create a new relation
- name of the relation is the name of the weak entity
- attributes of entity become attributes of relation

Consider Weak Entities

- primary key of the relation is the concatenation of the primary key of the relation created from the strong entity and the discriminator of the weak entity
- the attribute which is the primary key of the relation created from the strong entity is also a foreign key

Consider Weak Entities





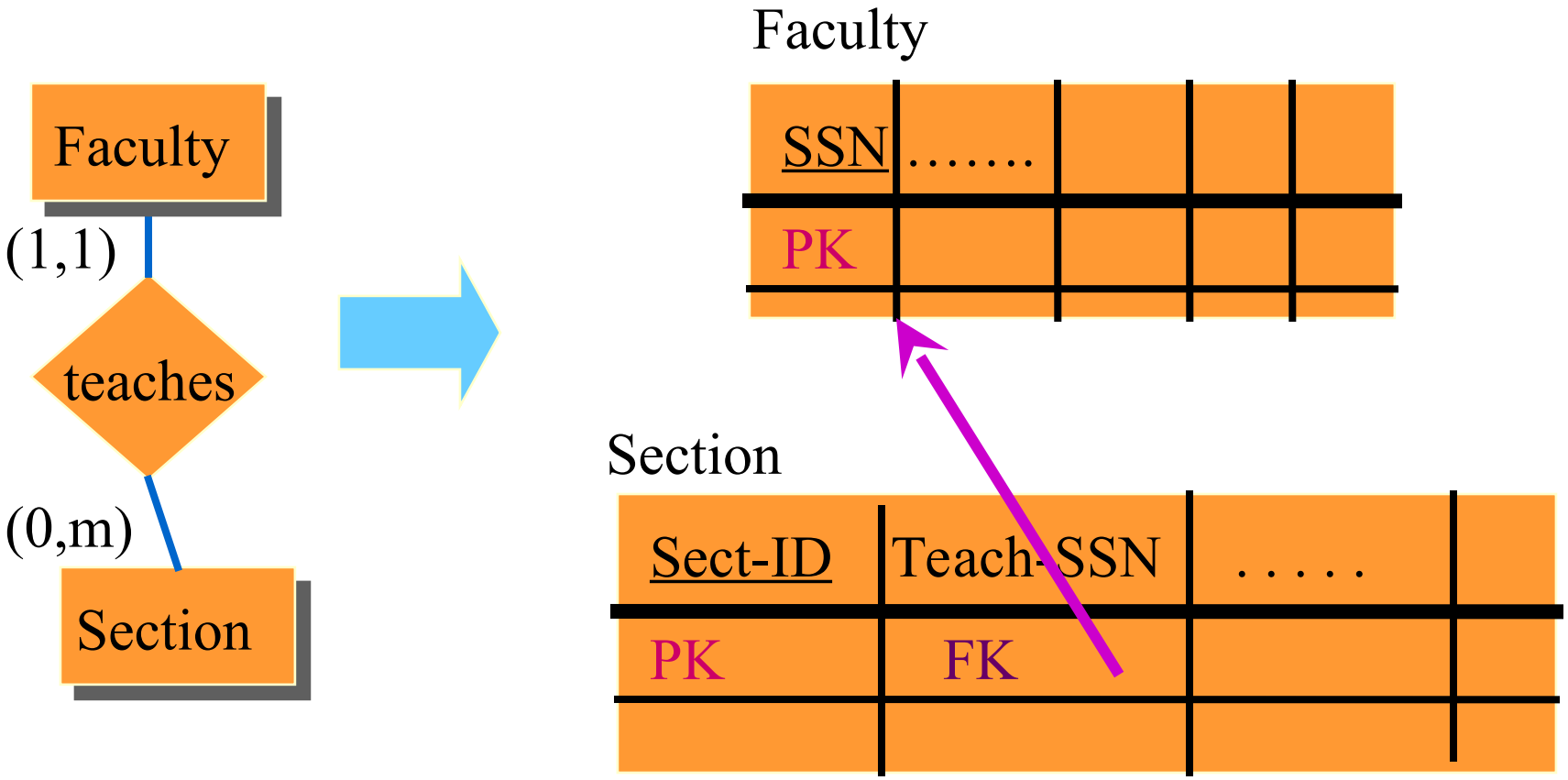
Address

SSN	Type	…		
PK	PK			
FK				

Consider One-to-many Binary Relationships

- The primary key of the relation created from the “one” entity becomes a foreign key in the relation created from the “many” entity.

Consider One-to-many Binary Relationships



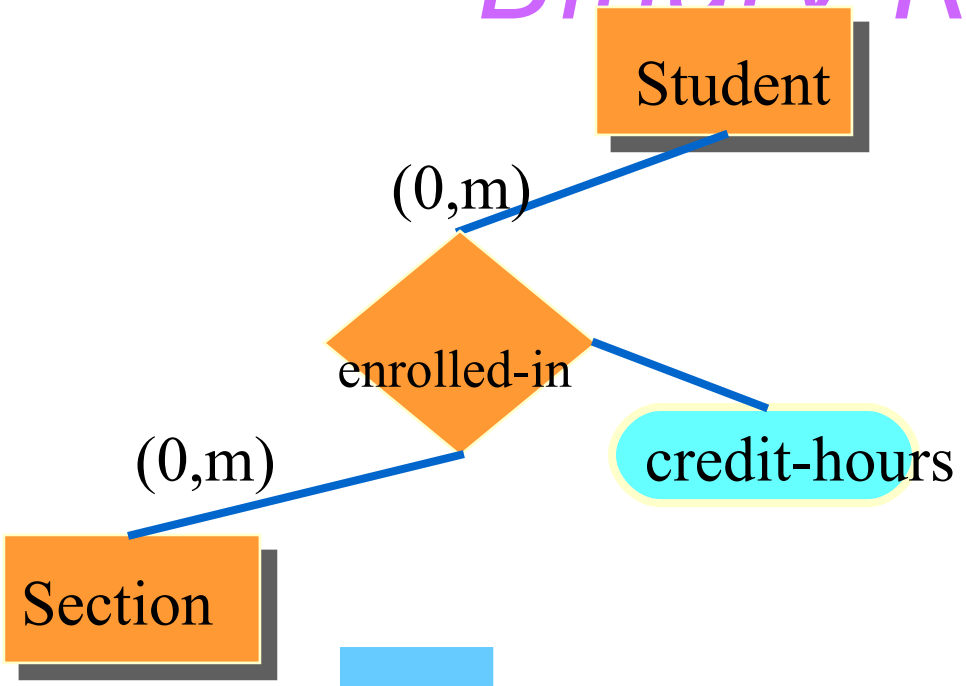
Consider Many-to-many Binary Relationships

- Create a new relation for the relationship whose primary key is the concatenation of the entity-ids of the related entities.
- The primary key attributes are also foreign keys into the relations created from the related entities.

Consider Many-to-many Binary Relationships

- The name of the new relation should reflect the relationship name.
- The intersection data of the relationship become non prime attributes of the relation.

Consider Many-to-many Binary Relationships



enrolled-in

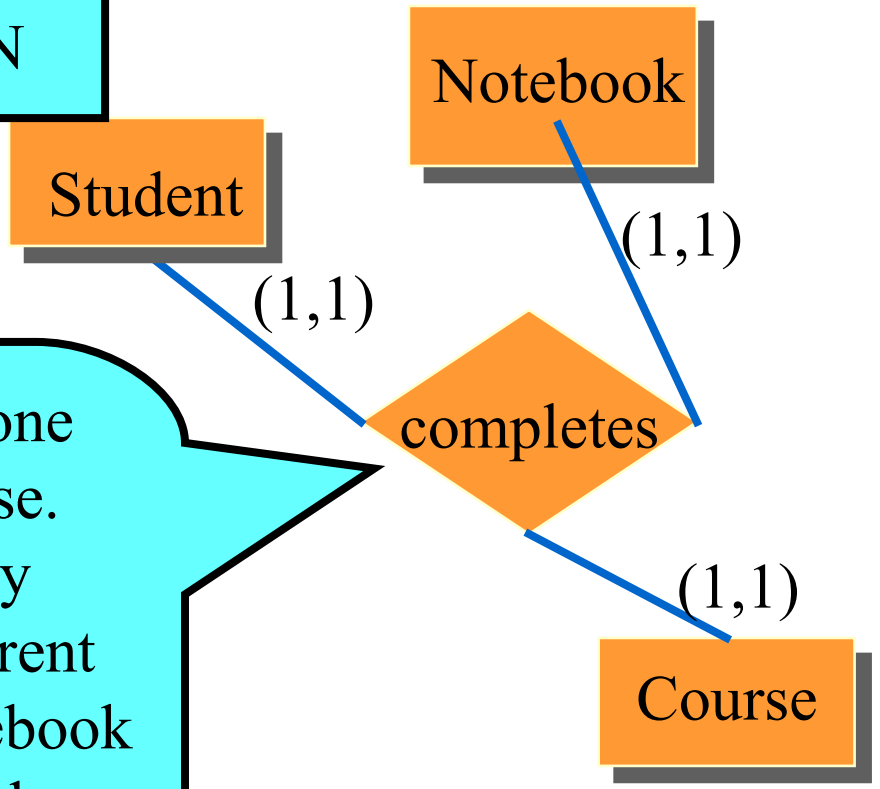
SSN	Sect-ID	Credit-hours
PK	PK	
FK	FK	

Consider Relationships Greater than Binary

- Create a new relation for the relationship.
- The primary key of the new relation depends upon the cardinalities of the relating entities.

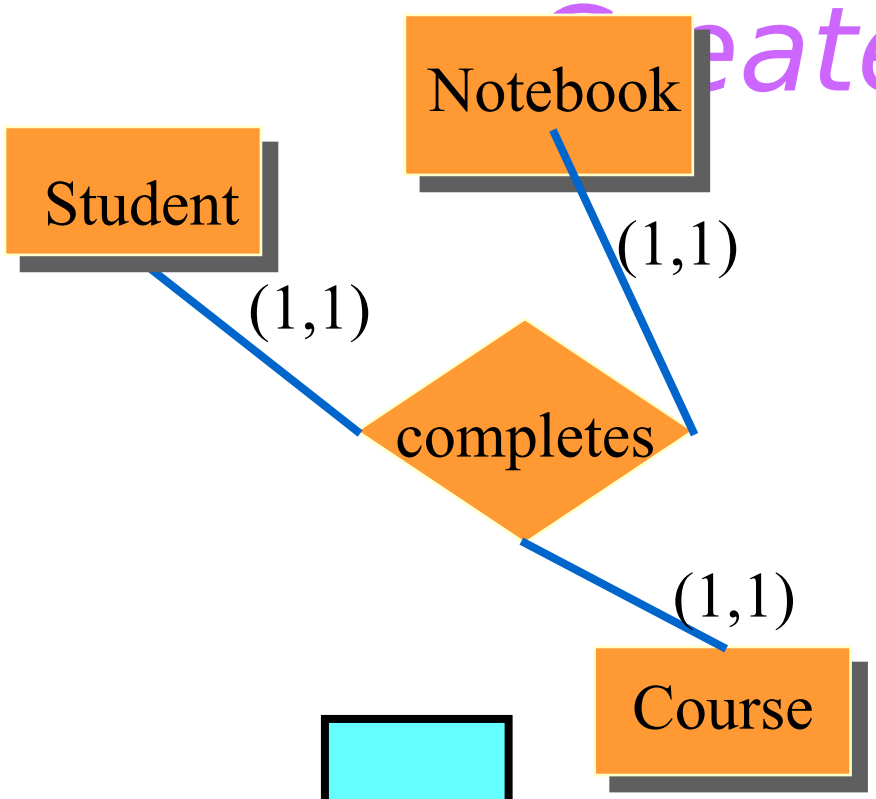
Consider Relationships ter than Binary

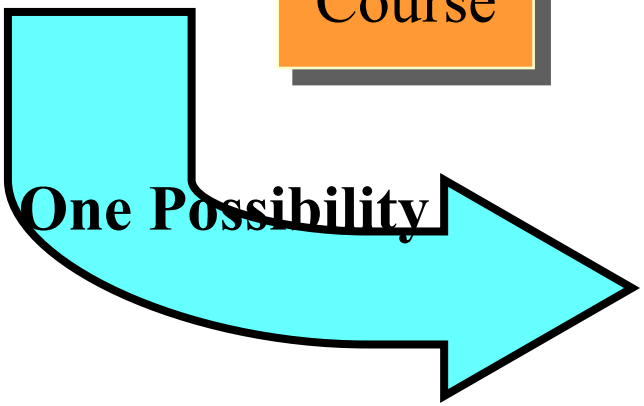
SSN, Crse-ID → Notebook-ID
SSN, Notebook-ID → Crse-ID
Crse-ID, Notebook-ID → SSN



A student used exactly one notebook for each course. He/she may be in many courses with many different notebooks. But each notebook belongs to one student and one course.

Consider Relationships Greater than Binary



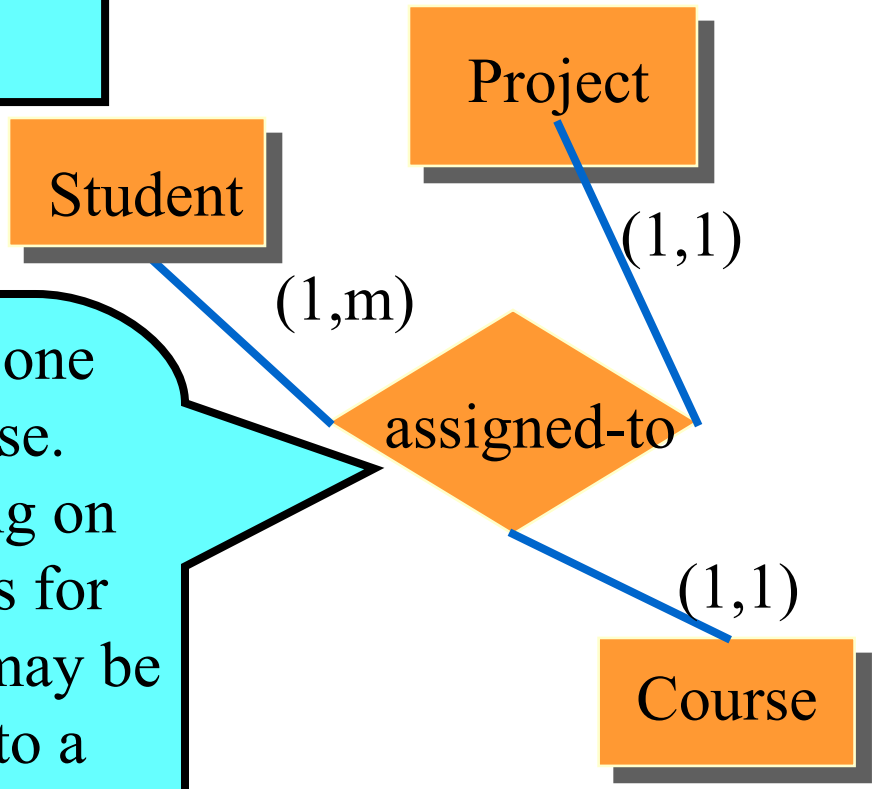


Completes

SSN	Crse-ID	Notebook-ID
PK	PK	
FK	FK	FK

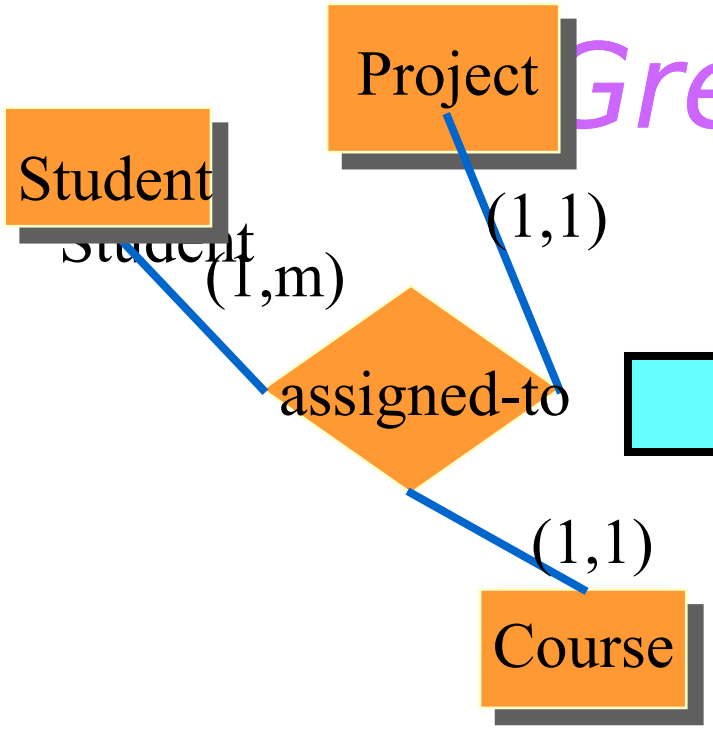
Consider Relationships Greater than Binary

SSN, Crse-ID → Proj-ID
SSN, Proj-ID → Crse-ID



A student is assigned to one project within each course. A student may be working on many projects but each is for a different course. There may be many students assigned to a project but each project is for a given course.

Consider Relationships Greater than Binary



OR

assigned-to

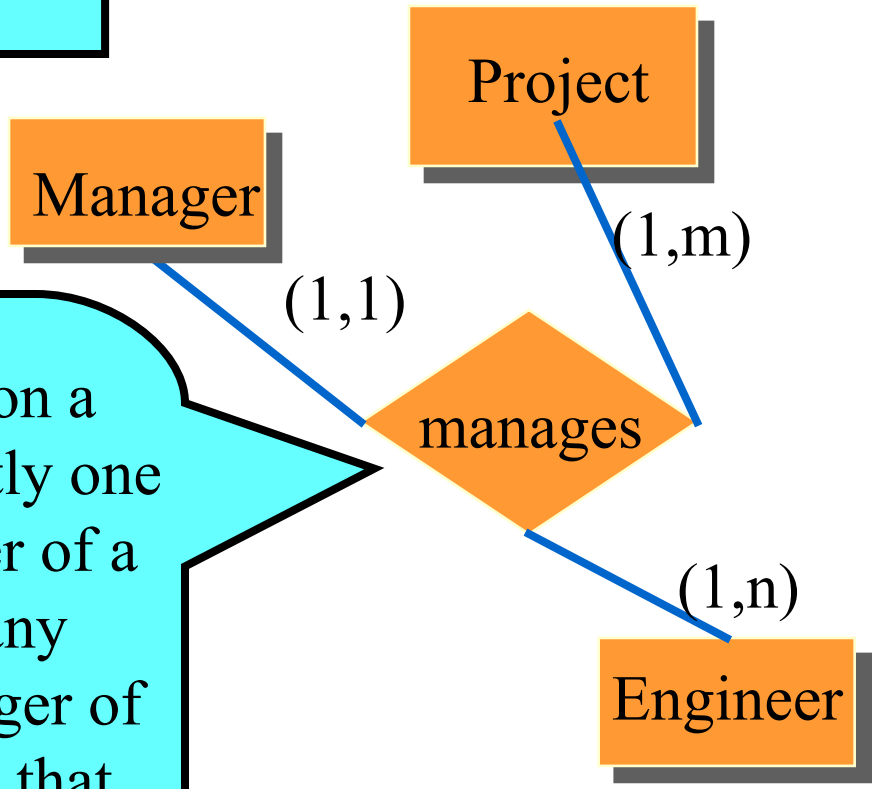
SSN	Proj-ID	Crse-ID
PK	PK	
FK	FK	FK

assigned-to

SSN	Crse-ID	Proj-ID
PK	PK	
FK	FK	FK

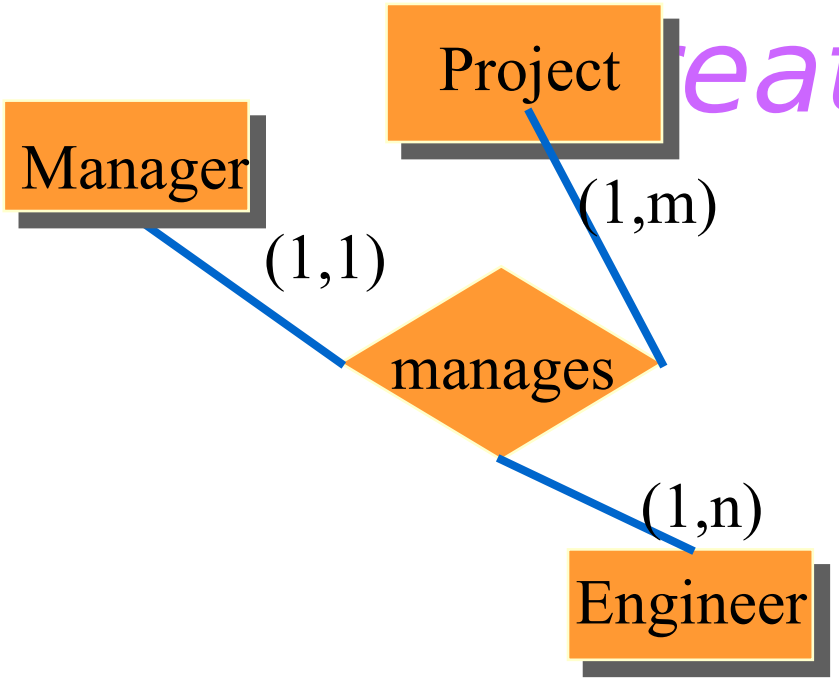
Consider Relationships Greater than Binary

Proj-ID, Engin-ID → Mgr-ID



Each engineer working on a particular project has exactly one manager, but each manager of a project may manage many engineers, and each manager of an engineer may manage that engineer on many projects.

Consider Relationships Greater than Binary

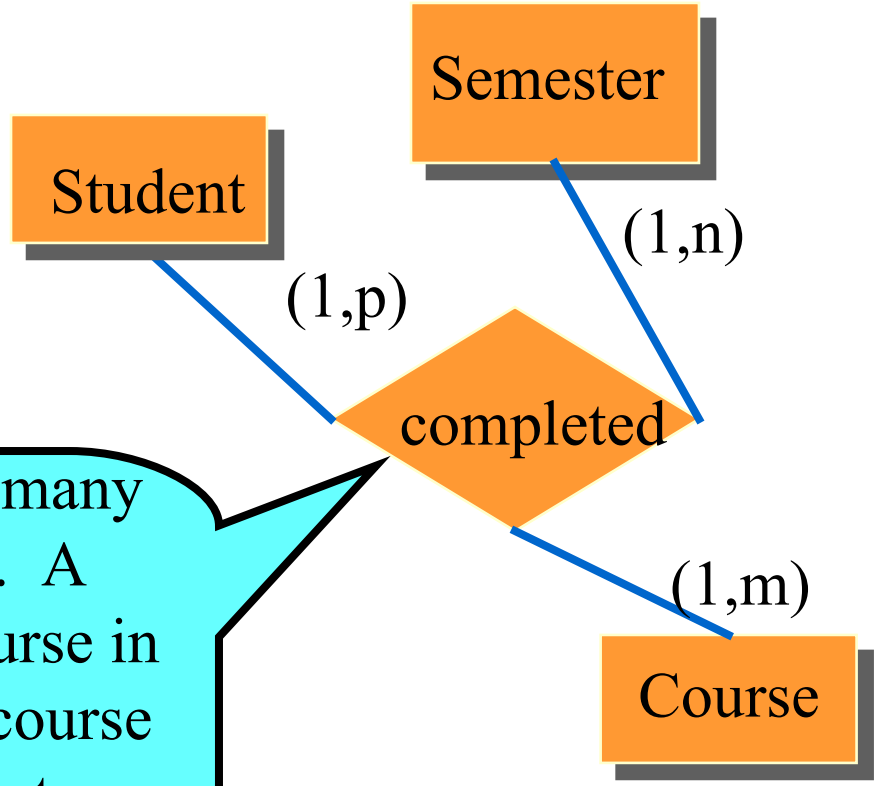


manages

Proj-ID	Engin-ID	Mgr-ID
PK	PK	
FK	FK	FK

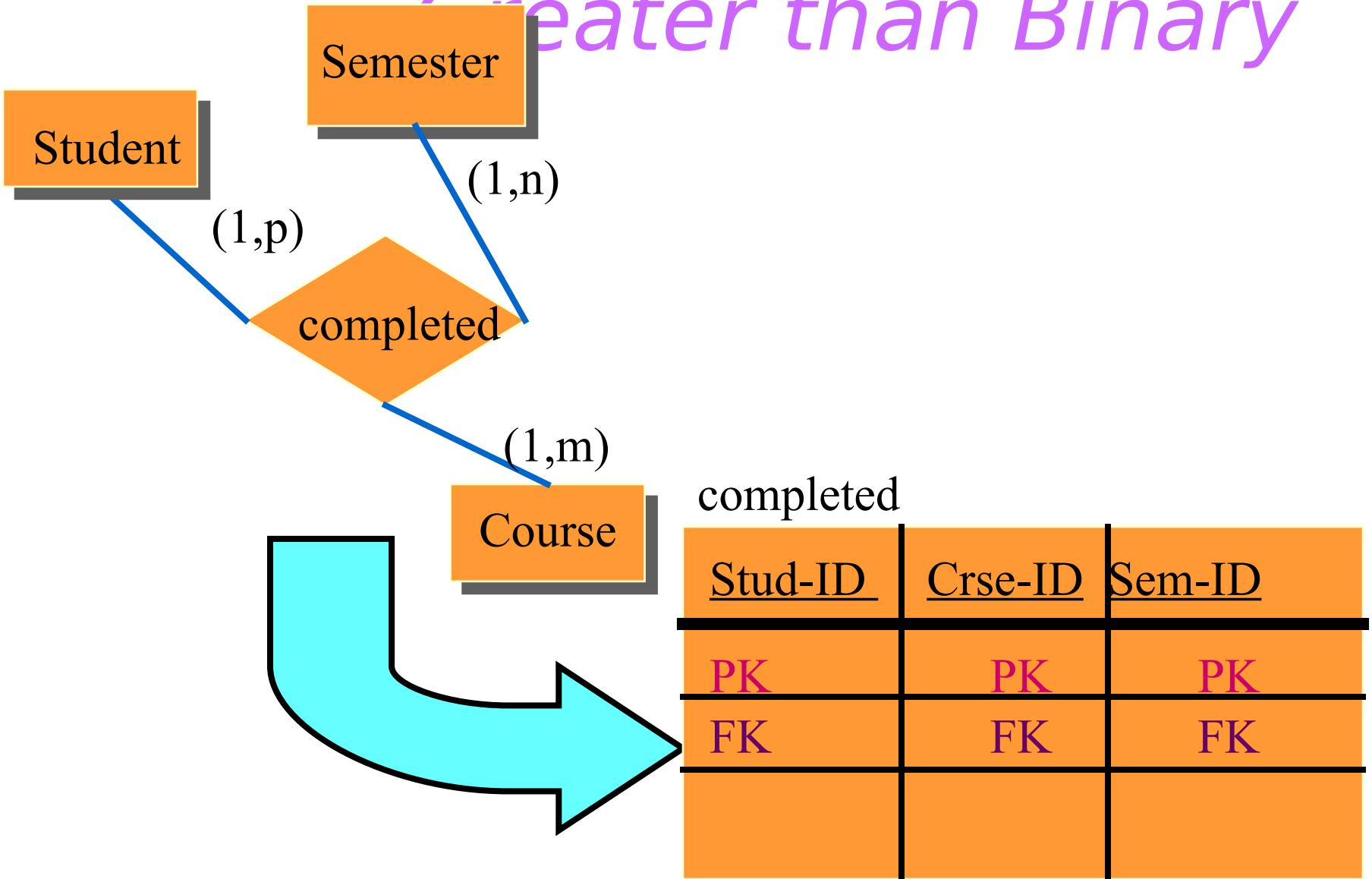
Consider Relationships Greater than Binary

No functional dependencies between entities.

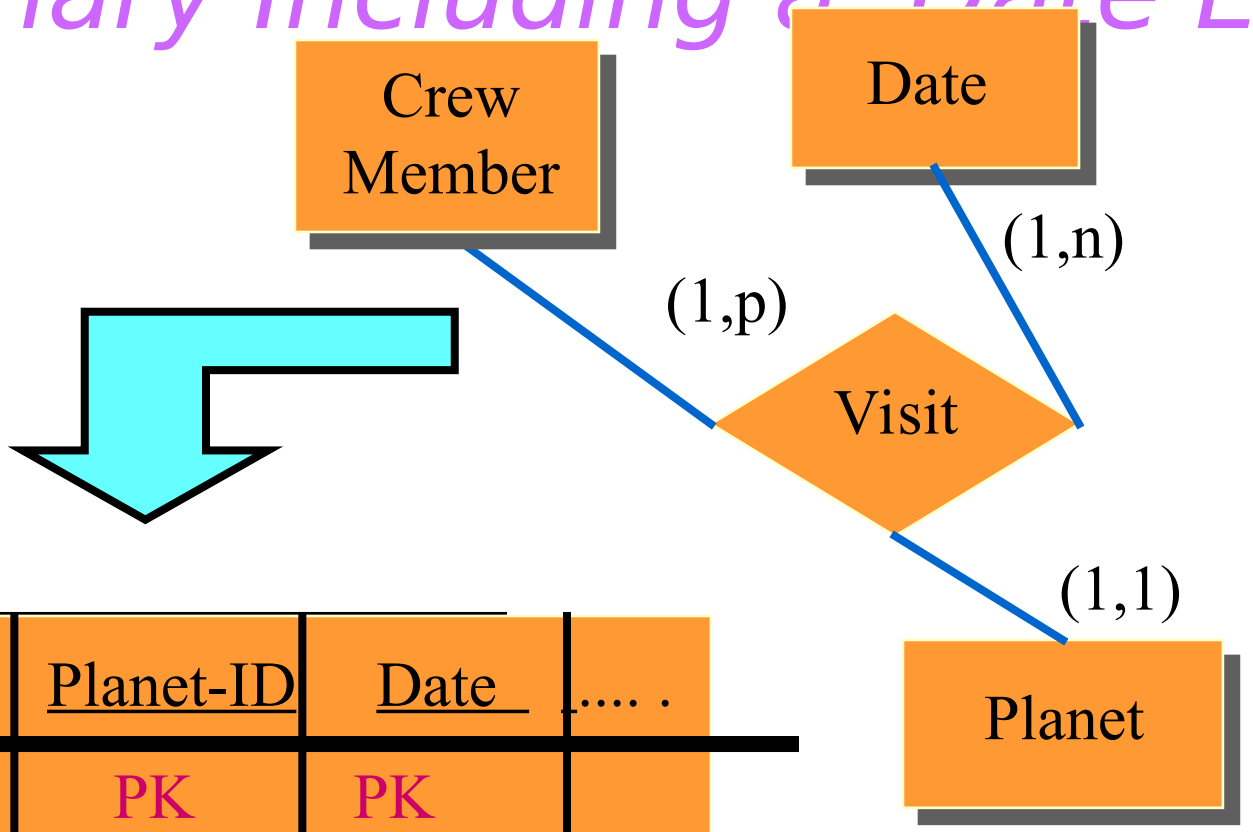


A student can complete many courses in a semester. A student may repeat a course in different semesters. A course can have many students enrolled in it in a semester.

Consider Relationships Greater than Binary

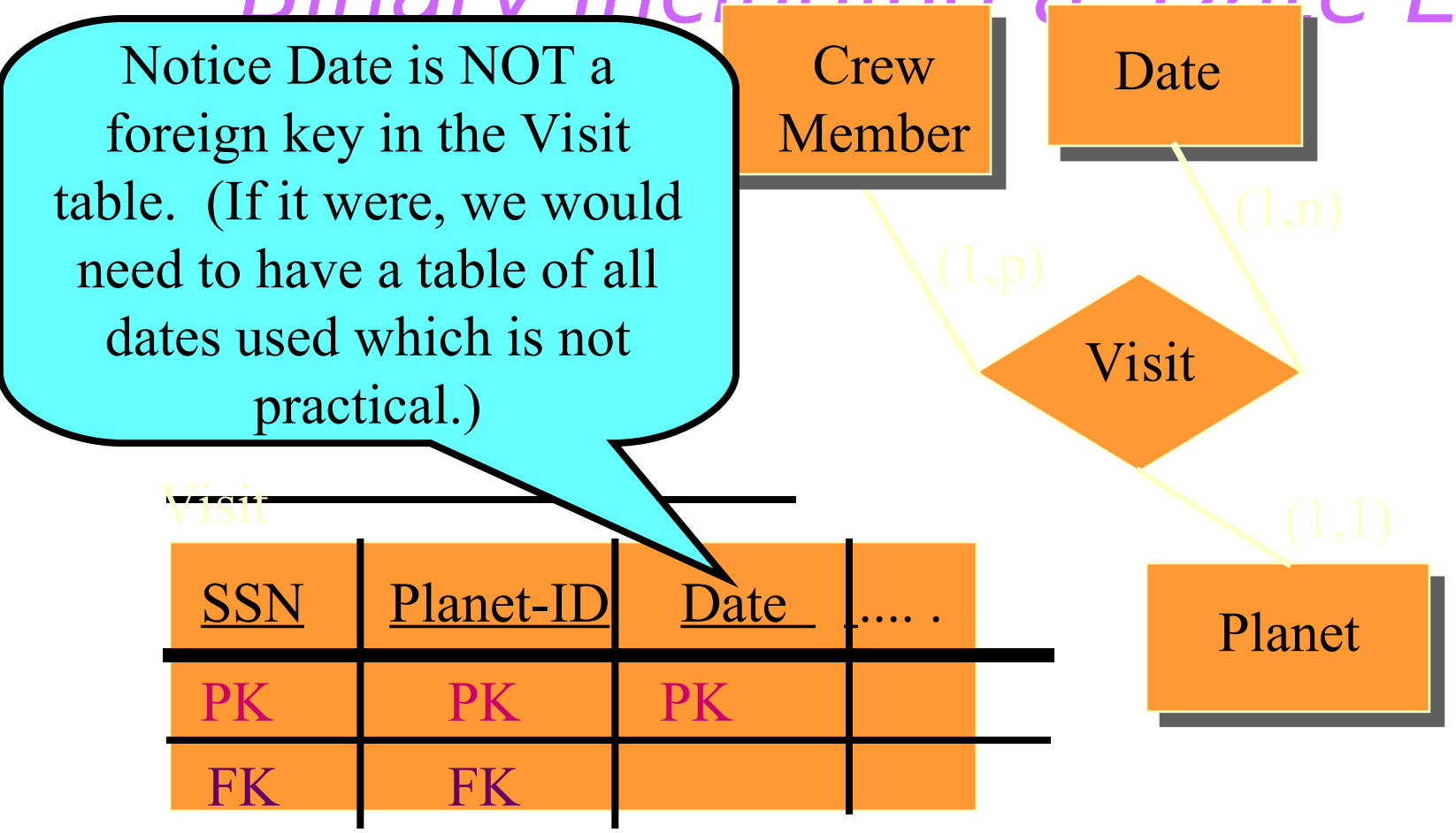


Consider Relationships Greater than Binary Including a Date Entity



Consider Relationships Greater than Binary Including a Date Entity

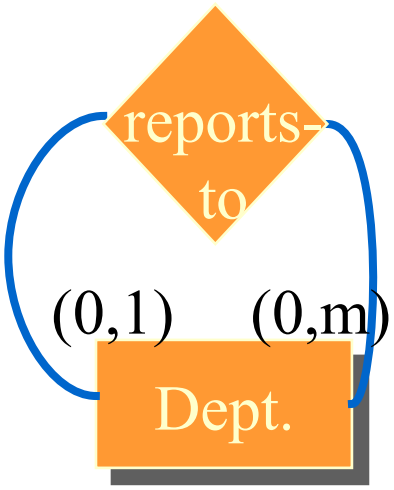
Notice Date is NOT a foreign key in the Visit table. (If it were, we would need to have a table of all dates used which is not practical.)

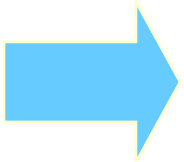


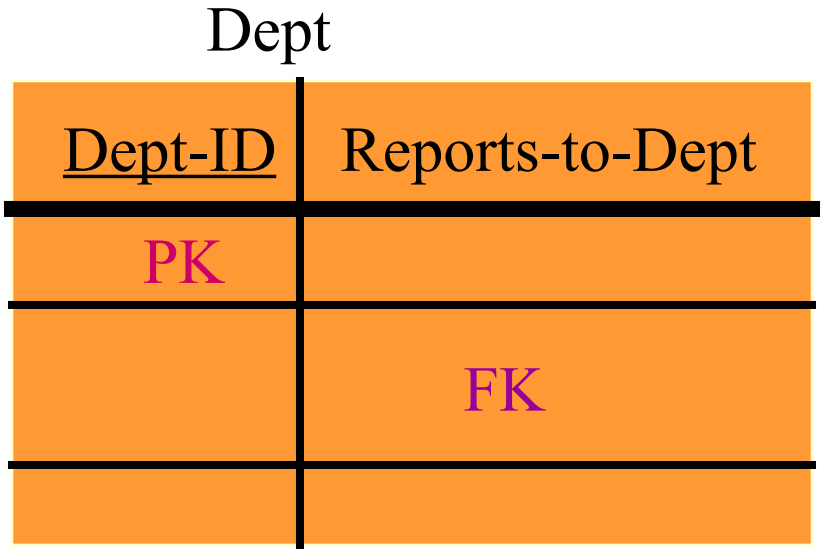
Consider Recursive Relationships

- Treat as the comparable type of relationship
  - one-to-many
    - use a foreign key
  - many-to-many
    - create a new relation for the relationship

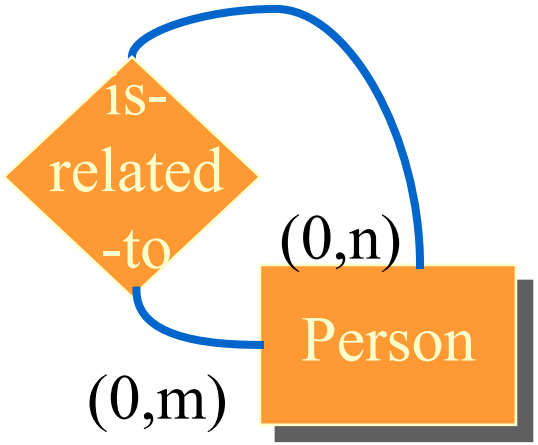
Consider Recursive Relationships

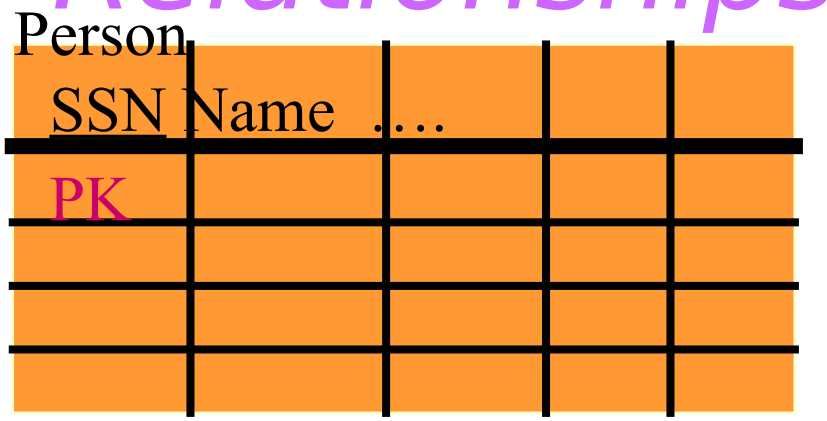


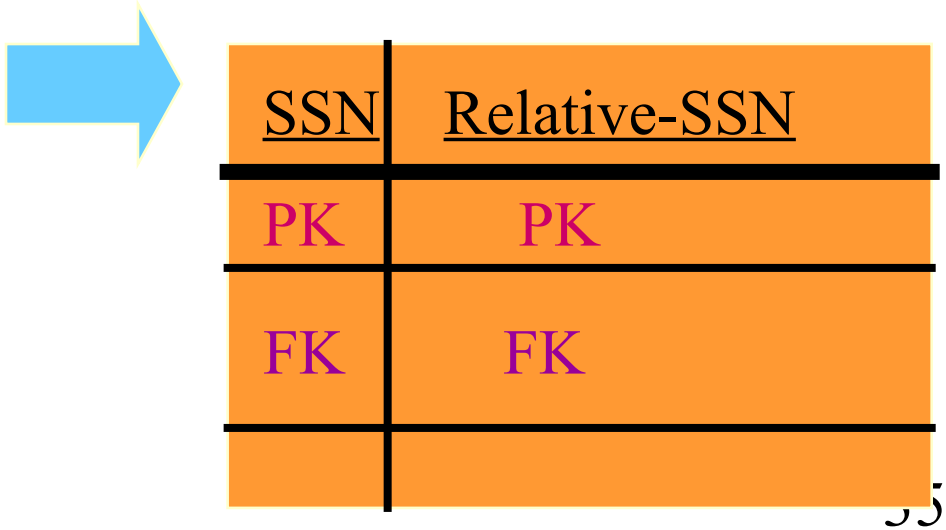




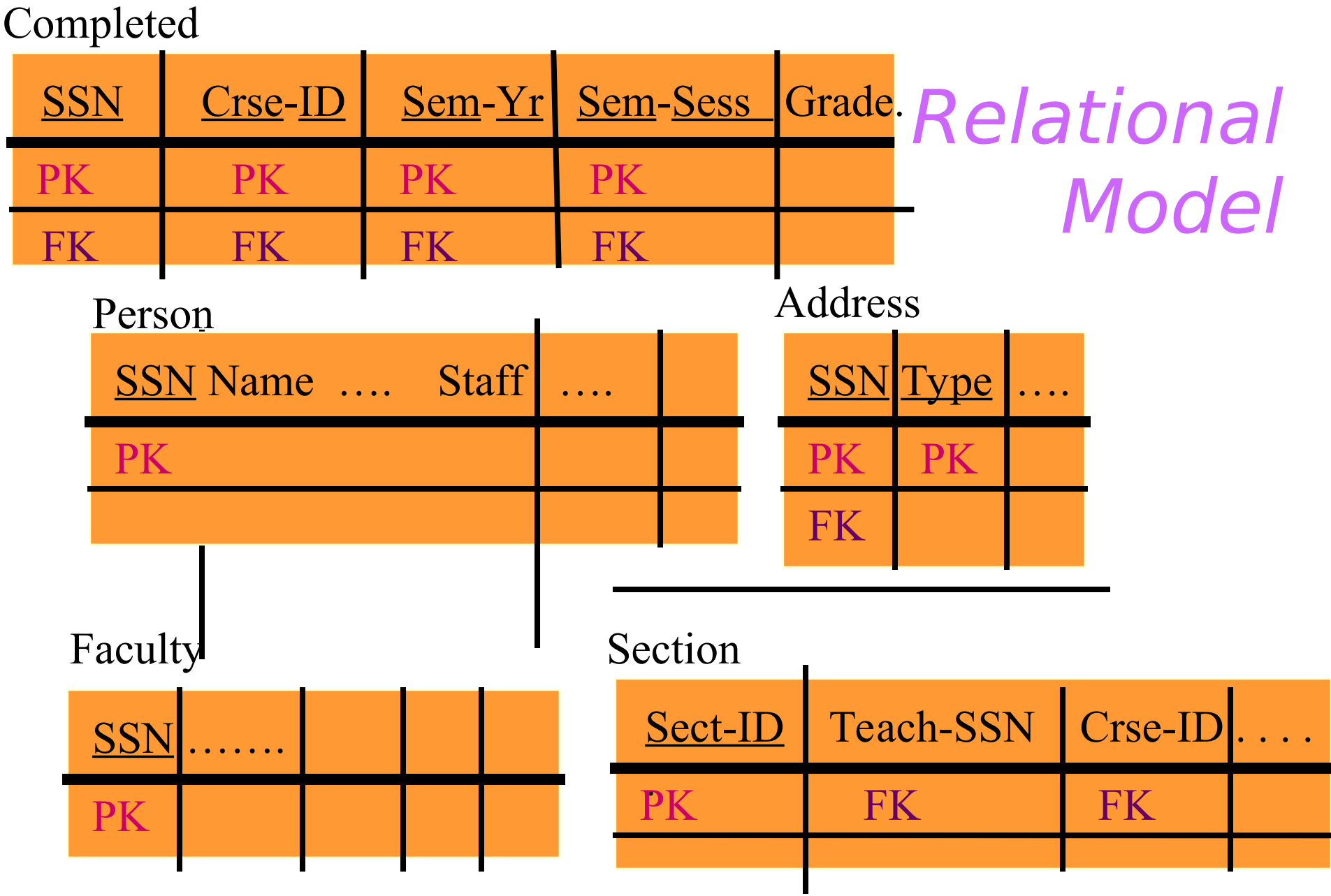
Consider Recursive Relationships





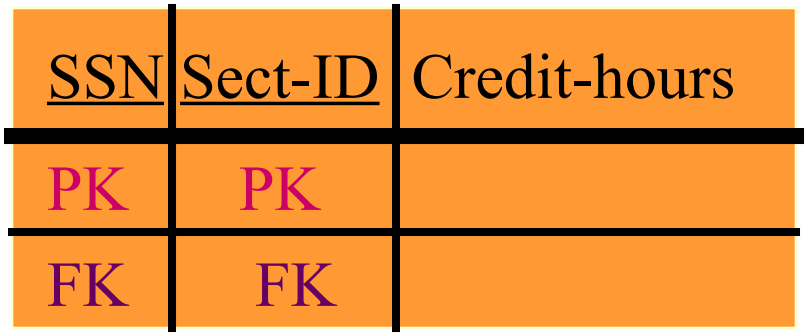


Relational Model

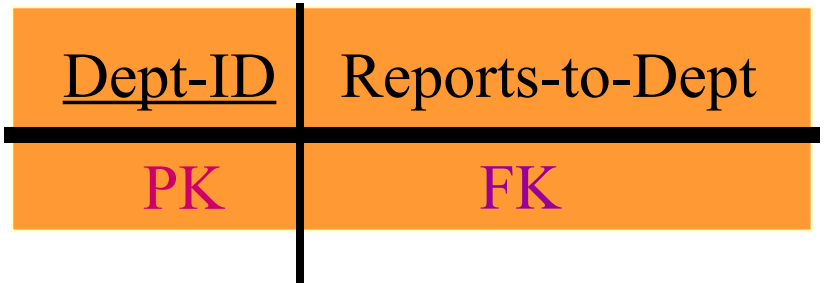


Relational Model

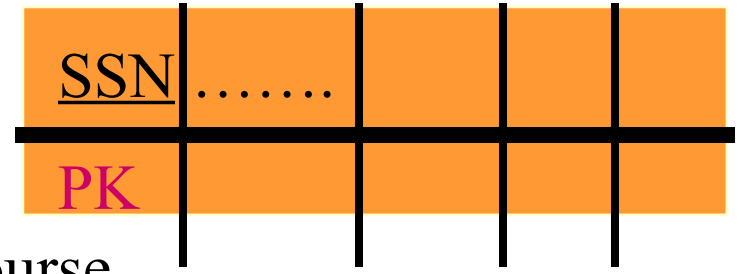
enrolled-in



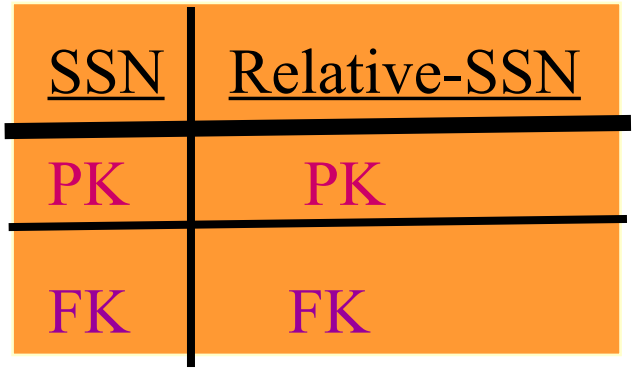
Dept



Student



is-related-to



Course