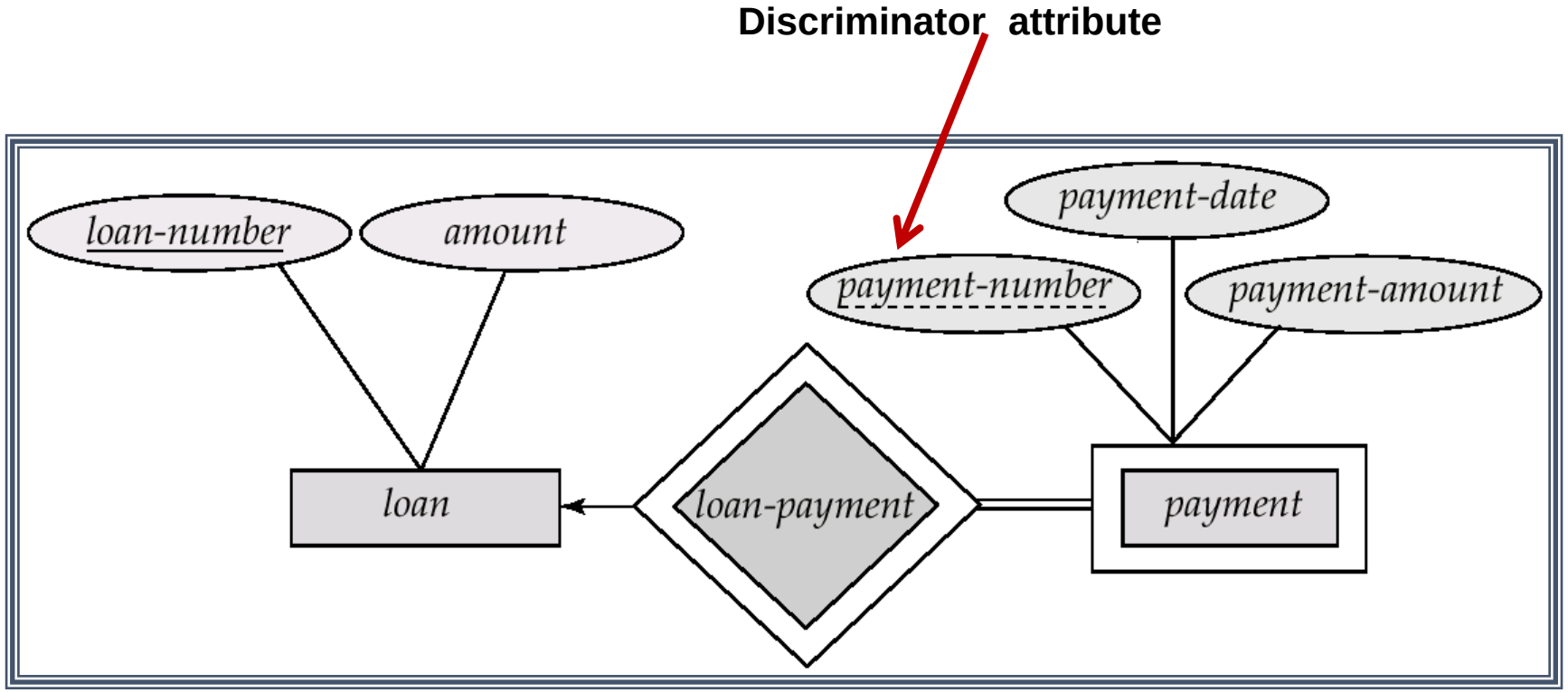
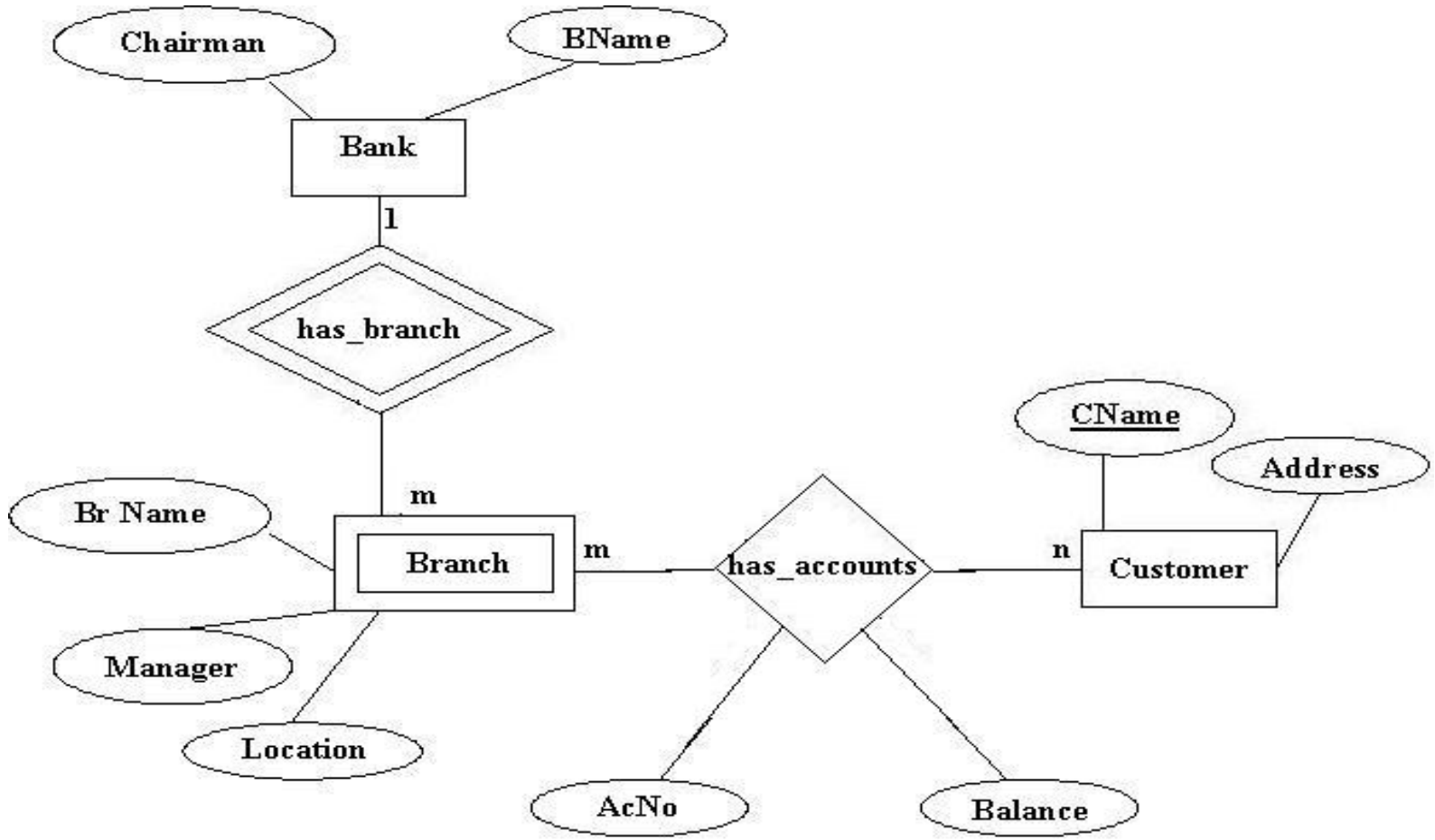
Weak Entity Sets

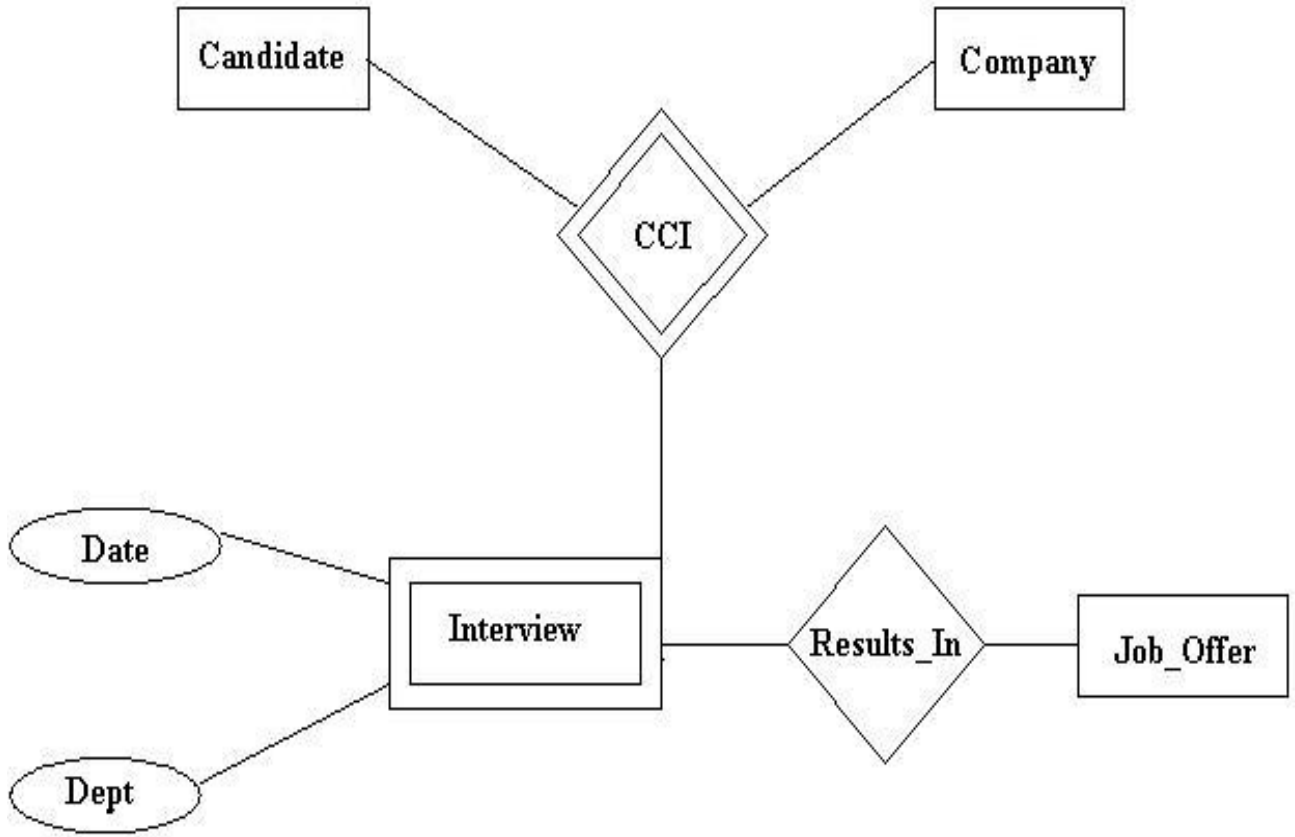
- An entity set that does not have a primary key
- The existence of a weak entity W depends on the existence of a *identifying entity M*

Weak Entity - example



Key for payment : loan-number + payment-number


Chairman
BName
Bank
1
has_branch
m
Br Name
Branch
m
has_accounts
n
Customer
CName
Address
Manager
Location
AcNo
Balance

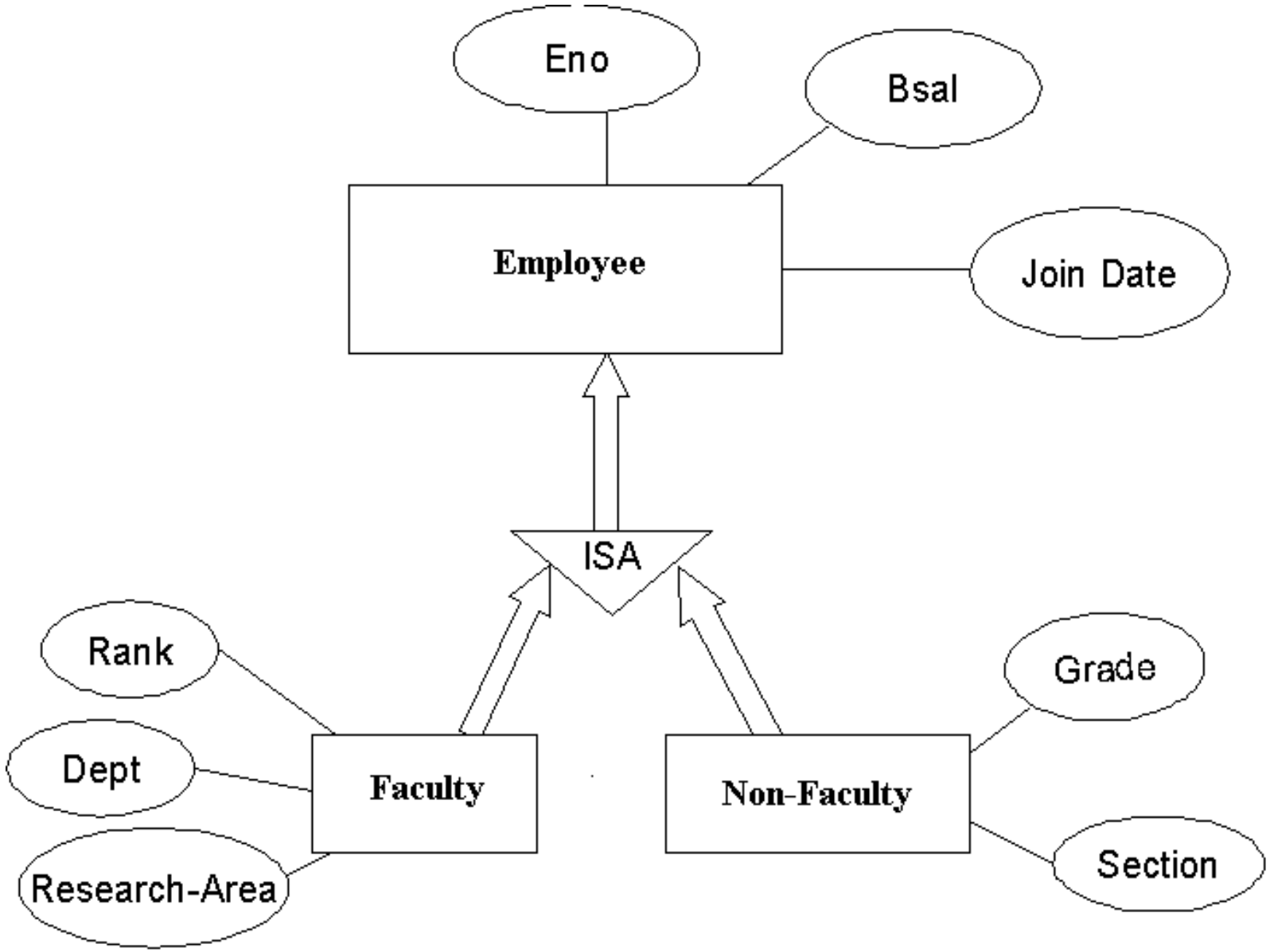

Candidate
Company
CCI
Date
Interview
Results_In
Job_Offer
Dept

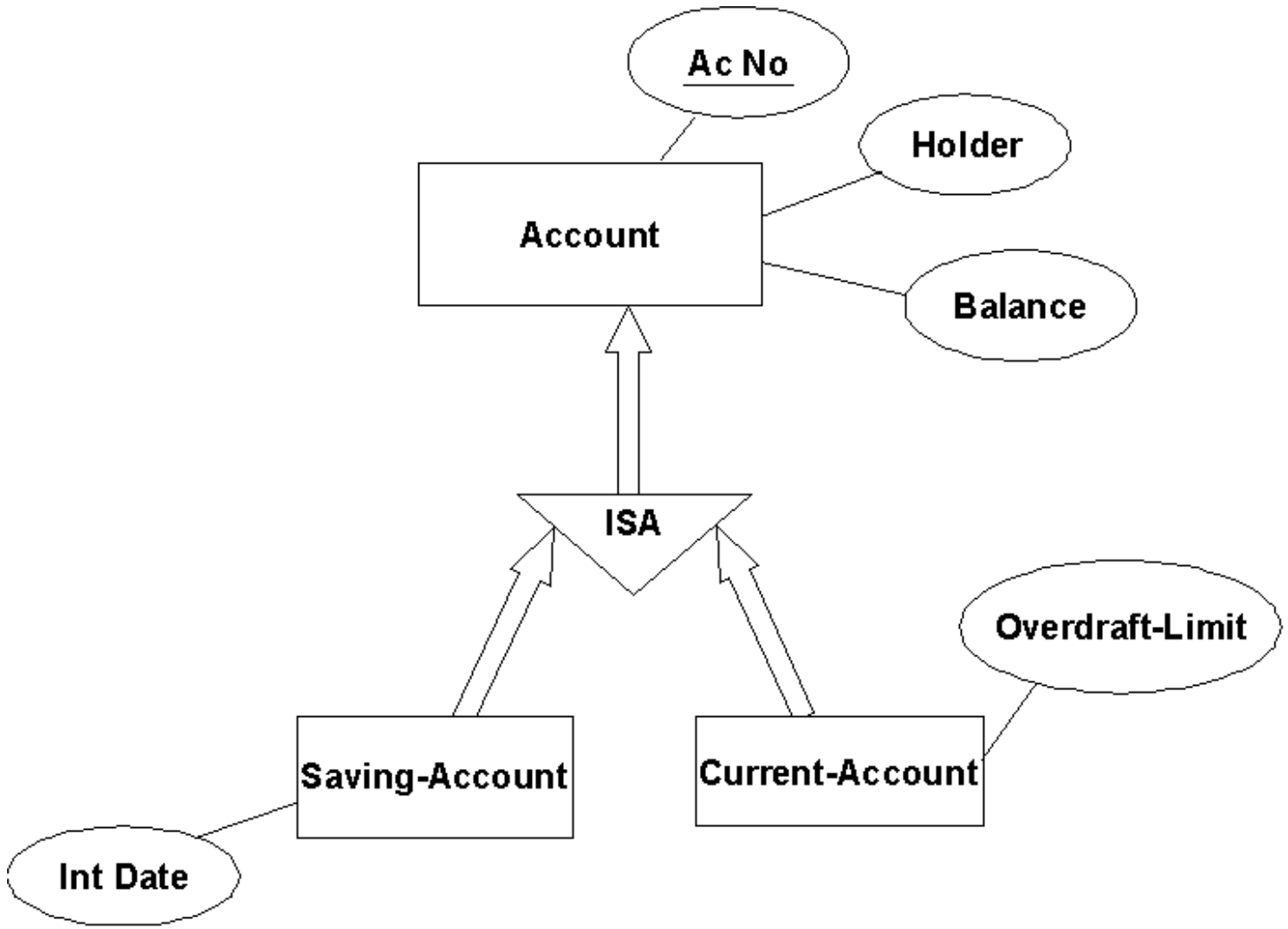
EXTENDED E-R MODEL

- extensions to capture more meaning
- concepts of generalization, aggregation and sub-set hierarchies added
  - Similar to OO concepts : inheritance, composite objects

Generalization

- to generalize from two or more entity sets and factor out commonality
- Example : given two entities Faculty and Student, we can define a ‘general’ entity called Person
- Common attributes are factored out
- Also called as IS-A relationship
- Implies a bottom-up design process


Eno
Bsal
Employee
Join Date
ISA
Rank
Dept
Research-Area
Faculty
Non-Faculty
Grade
Section

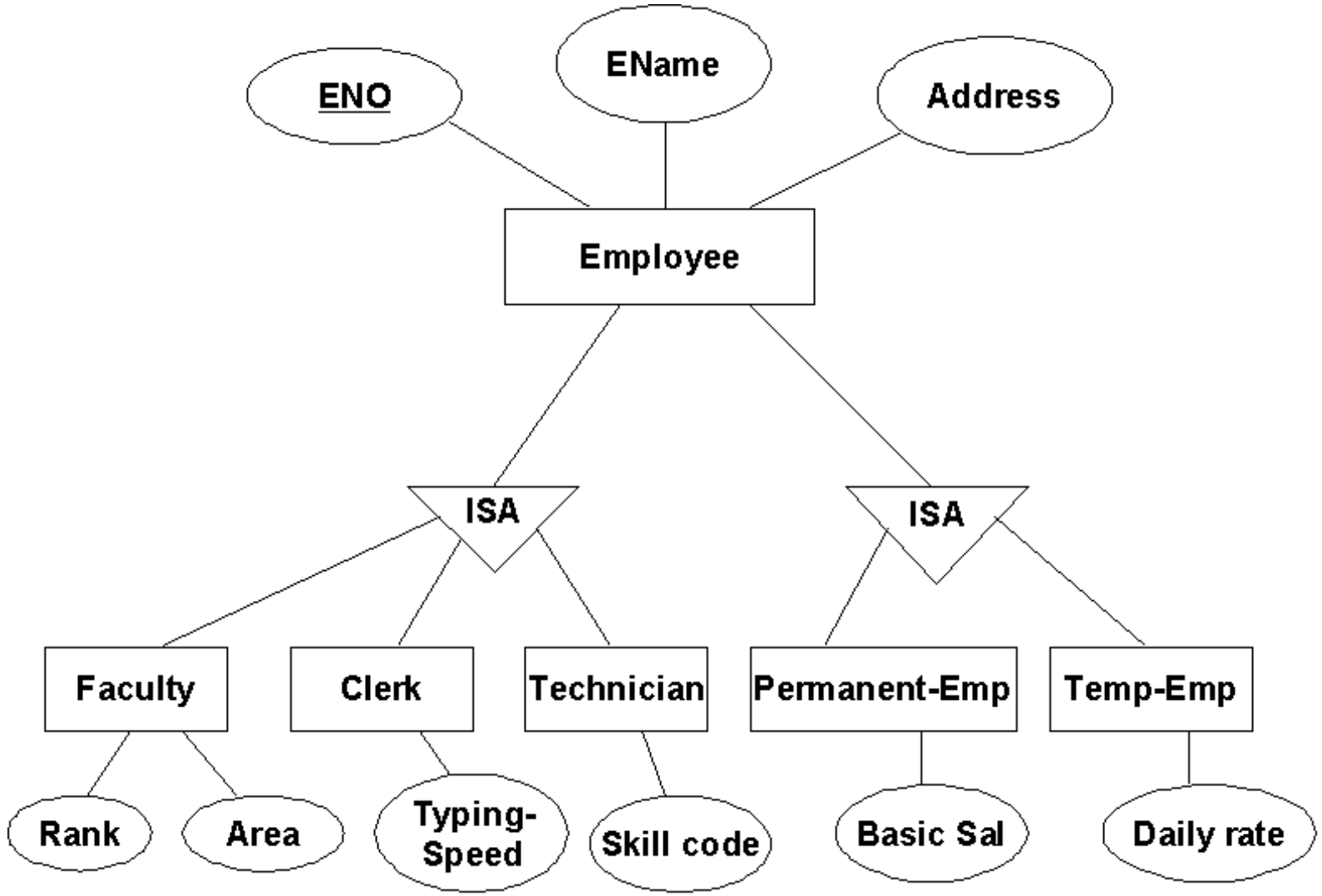

Ac No
Holder
Account
Balance
ISA
Overdraft-Limit
Saving-Account
Current-Account
Int Date

Specialization

- also called subset hierarchy; we create special cases for a given entity
  - Teacher as a special case of Employee
- this is also IS-A relationship

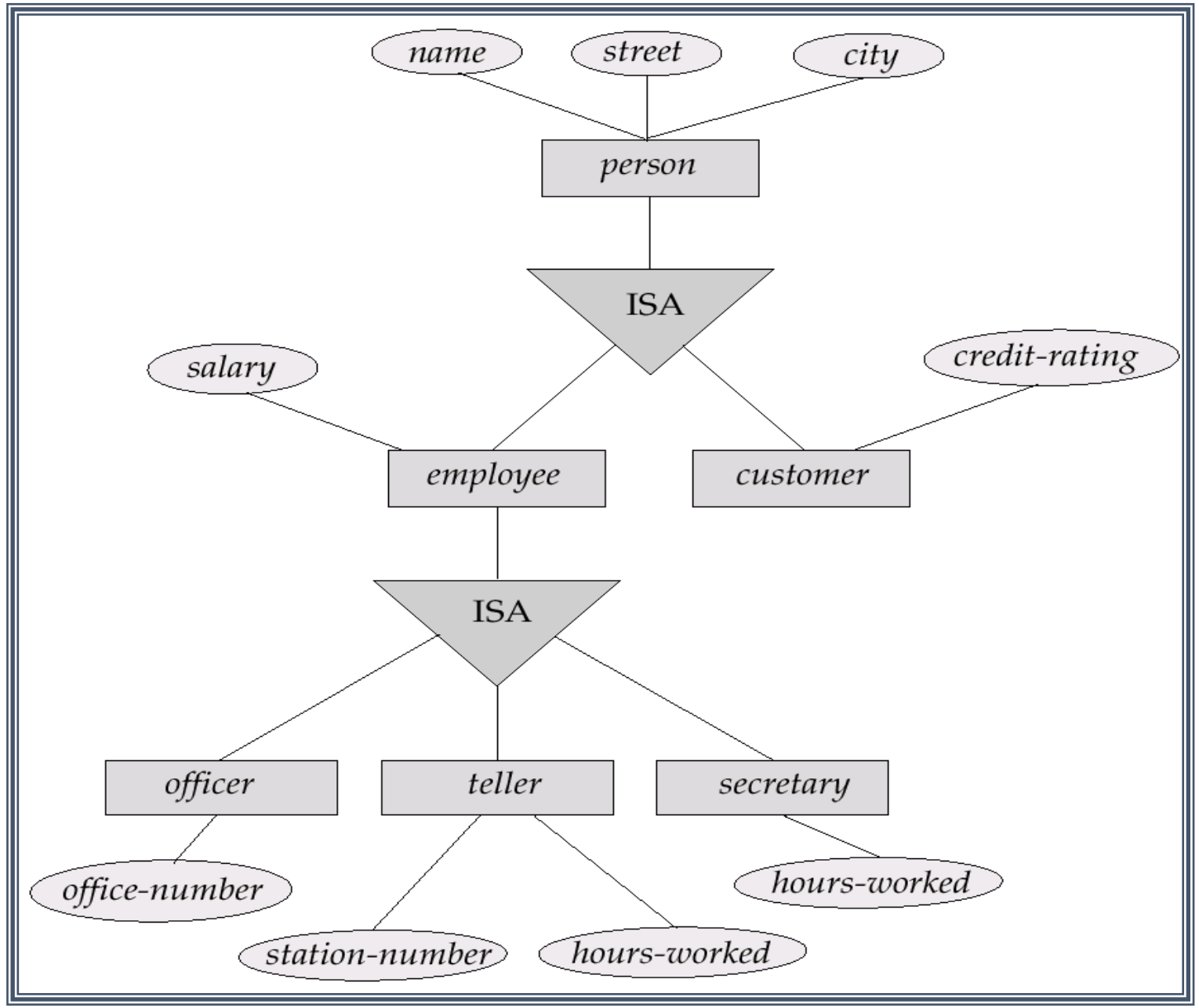
Specialization.....

- specialization allows classification into subsets based on some attribute
- we may have several specializations of same entity
- the subsets may have additional attributes
- Top-down design process


ENO
EName
Address
Employee
ISA
ISA
Faculty
Clerk
Technician
Permanent-Emp
Temp-Emp
Rank
Area
Typing-Speed
Skill code
Basic Sal
Daily rate

Inheritance

- inheritance present in both Generalization and specialization
  - Direction important : bottom-up in generalization, top-down in Specialization
  - Important to distinguish the two cases
  - E.g., every instance in E need not be present in subsets of E
    - Called Completeness constraint


name
street
city
person
ISA
salary
credit-rating
employee
customer
ISA
officer
teller
secretary
office-number
station-number
hours-worked
hours-worked

Constraints on a Specialization/Generalization

- Constraint on which entities can be members of a given lower-level entity set.
  - condition-defined
    - E.g. all customers over 65 years are members of *senior-citizen* entity set;
    - *senior-citizen* ISA *person*.

Constraints …

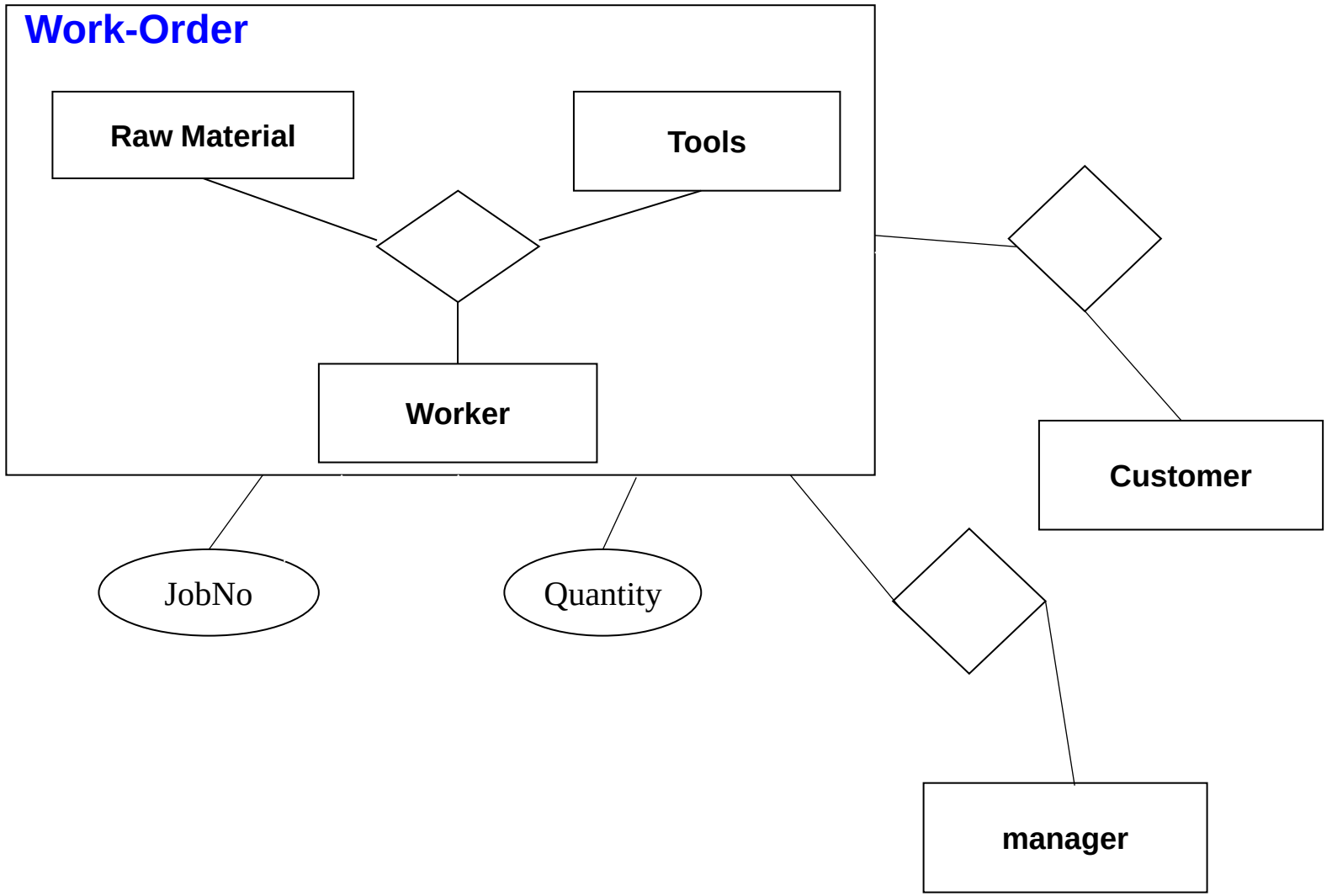
- Constraint on specialization.
  - Disjoint sub-types
    - Faculty, Student are disjoint
  - Overlapping sub-types
    - Student, Scholar (having scholarships) overlap

Aggregation

- for building complex entity from existing entities
  - having attribute whose value is another entity
  - Contains instances of other entities within

Examples :

- Work-order entity defined as consisting of entities Raw-material, Tools and Workers;
- Work-order itself related with Customer entity
- Aggregation notation not explicitly provided in Extended E-R model


Work-Order
Raw Material
Tools
Worker
Customer
JobNo
Quantity
manager

Converting ER model to Relations

- A few simple rules
  - Entities and relationships to tables
  - A few refinements
  - Dealing with weak entities
  - Dealing with generalizations/specializations

- Let us illustrate with an example

Converting E-R Schema to Tables

- Primary keys allow entity sets and relationship sets to be expressed uniformly as *tables*
- For each entity set and relationship set there is a unique table
- Each table has columns corresponding to attributes

Representing Entities

- A strong entity maps to a table with the same attributes.
- Composite attributes are flattened out by creating a separate attribute for each component attribute in same or as a separate table
- A multivalued attribute X of an entity E is represented by a separate table EX (key of E, X)
  - Each value of the multivalued attribute maps to a separate row of the table EX

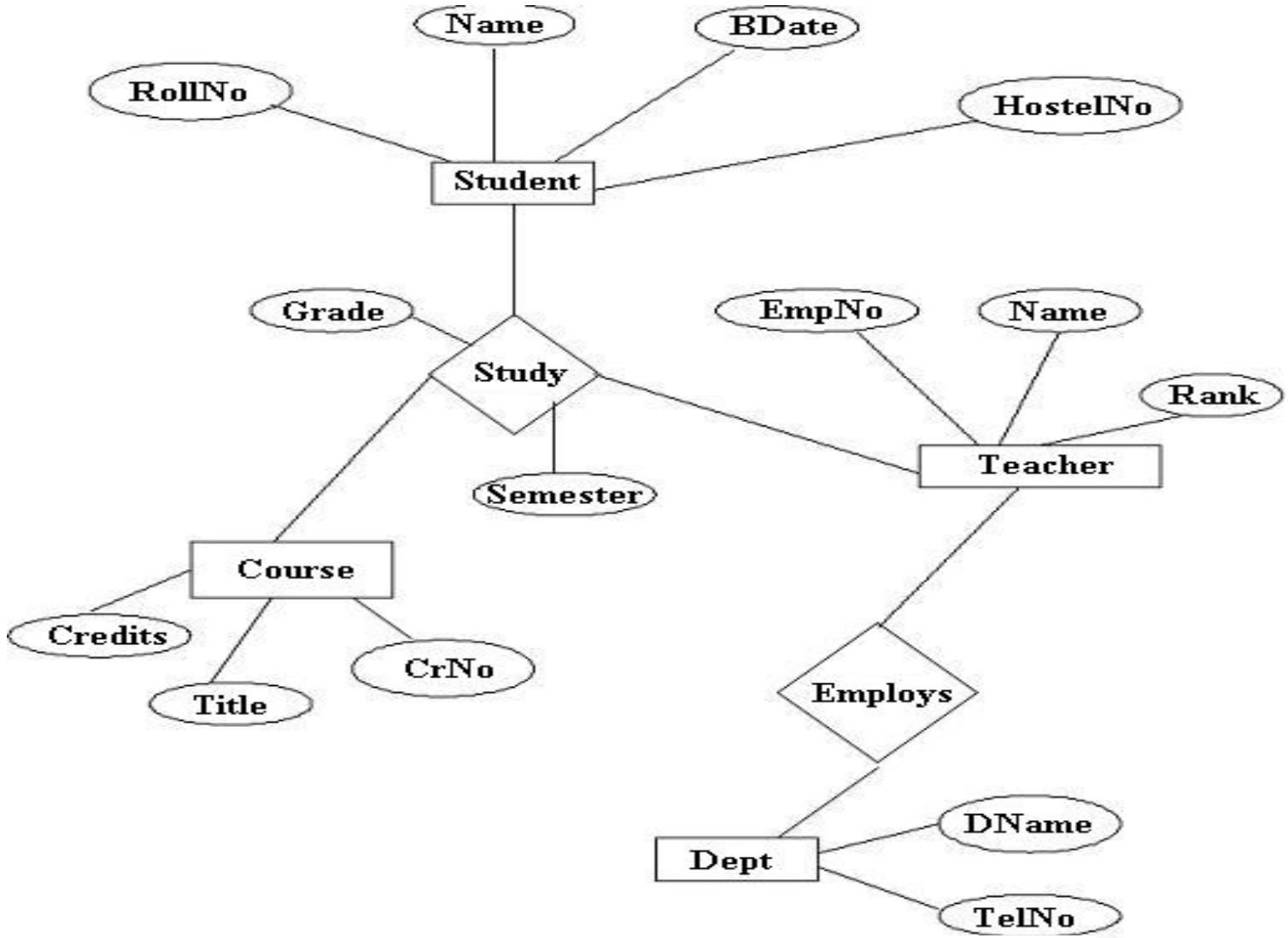
Representing relationships

- **For one-to-one relationship Between E1 and E2, either of their tables can represent the relationship by including the key of the other**
  - **Consider Department Head relationship with Person**
- **A many-to-many relationship : also as a table containing**
  - **primary keys of the two participating entities,**
  - **any attributes of the relationship**

Relationships …

- Many-to-one and one-to-many relationships that are total on the many-side :
  - Works relationship from Dept to Emp with tables for EMP and Dept
  - No separate table for relationship
  - Add to the many side the primary key of the one side (dept_id to EMP)
- If participation is *partial* on the many side, null values will need to be allowed
- The relationship between weak and its identifying strong entity, being one-to-many, modeled like this

Convert to Relational model



Representing Specializations

- Choice based on how data will be used
- Method 1:
  - Form a table for the higher level entity
  - Form a table for each lower level entity set, include primary key of higher level entity set and local attributes

table	table attributes
person	*name, street, city*
customer	*name, credit-rating*
employee	*name, salary*

  - Drawback: getting information about, e.g., *employee* requires accessing two tables

Representing Specialization as Tables (Cont.)

- Method 2:
  - Form a table for each entity set with all local and inherited attributes

table	table attributes
person	*name, street, city*
customer	*name, street, city, credit-rating*
employee	*name, street, city, salary*

- If specialization is total, table for generalized entity (*person*) not required to store information
  - Can be defined as a “view” relation containing union of specialization tables
  - But explicit table may still be needed for foreign key constraints

Relations Corresponding to Aggregation

- **To represent aggregation, create a table containing**
  - **primary key of the aggregated relationship,**
  - **the primary key of the associated entity set**
  - **Any descriptive attributes**

Exercise : Understand and convert to relational data model