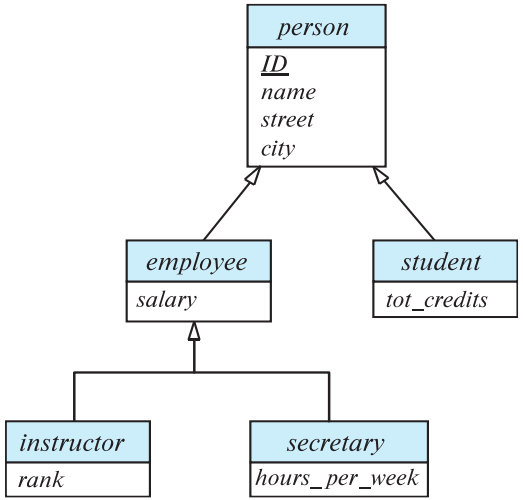
Extended ER Features

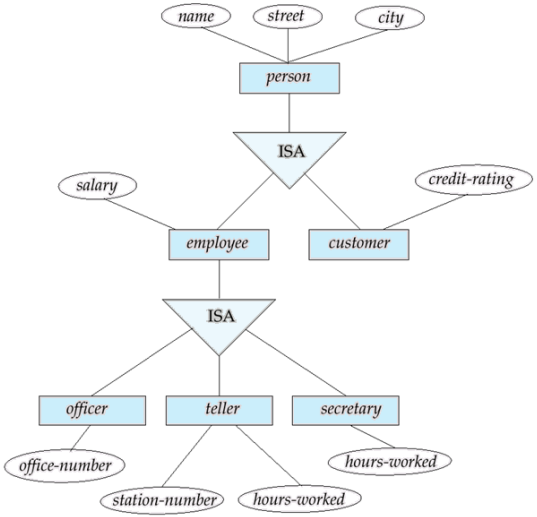
Specialization

- Top-down design process, we designate sub-groupings within an entity set that are distinctive from other entities in the set
- These sub-groupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set
- Depicted by a triangle component labeled *Is-A* (e.g., ***instructor*** "*is a*" ***person***)
- **Attribute inheritance:** A lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked

Example

- **Overlapping: *employee*** and ***student***
- **Disjoint: *instructor*** and ***secretary***
- Total and partial





Representing Specialization via Schemas

Method 1

- Form a schema for the higher-level entity
- Form a schema for each lower-level entity set, include primary key of higher-level entity set and local attributes

schema	attributes
person	ID, name, street, city
student	ID, tot_cred
employee	ID, salary

- **Drawback:** Getting information about, an ***employee*** requires accessing two relations, the one corresponding to the low-level schema and the one corresponding to the high-level schema

schema	attributes
person	ID, name, street, city
student	ID, name, street, city, tot_cred
employee	ID, name, street, city, salary

Method 2:

- Form a schema for each entity set with all local and inherited attributes
- 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
- **Drawback***: name, street* and *city* may be stored redundantly for people who are both ***students*** and ***employees***

Generalization

- A bottom-up design process, combine a number of entity sets that share the same features into a higher-level entity set
- Specialization and generalization are simple inversions of each other; they are represented in an ER diagram in the same way
- The terms specialization and generalization are used interchangeably
- Can have multiple specializations of an entity set based on different features
- E.g., ***permanent-employee*** vs. ***temporary-employee***, in addition to ***officer*** vs. ***secretary*** vs. ***teller***
- Each particular employee would be
  - A member of one of ***permanent-employee*** or ***temporary-employee***
  - And also a member of one of ***officer***, ***secretary***, or ***teller***
- The *Is-A* relationship also referred to as superclass - subclass relationship

Design 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*** *Is-A* ***person***
  - User-defined
- Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization
  - Disjoint
    - An entity can belong to only one lower-level entity set
    - Noted in ER diagram by writing disjoint next to the *Is-A* triangle
  - Overlapping
    - An entity can belong to more than one lower-level entity set

Completeness Constraint

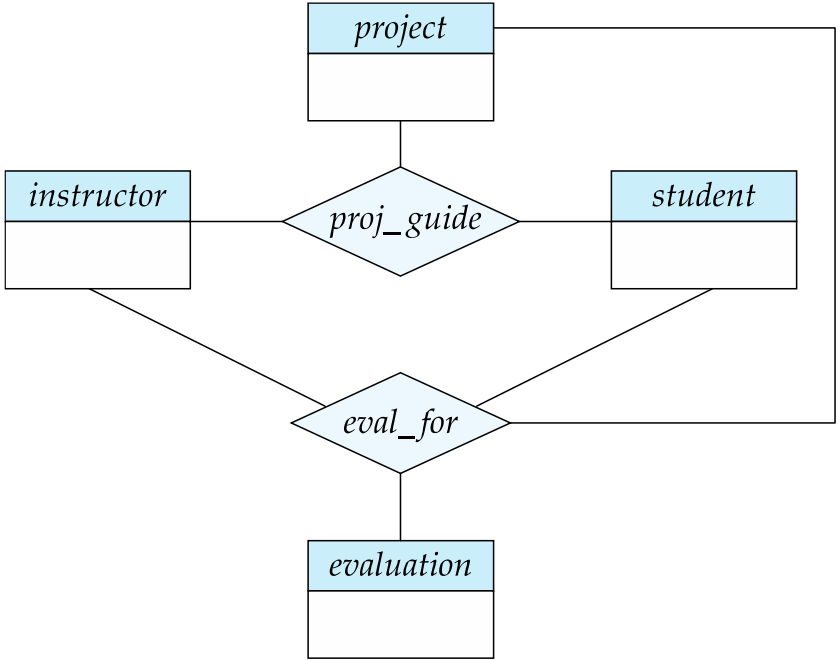
- Specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization
  - **Total:** An entity must belong to one of the lower-level entity sets
  - **Partial:** An entity need not belong to one of the lower-level entity sets
- Partial generalization is the default

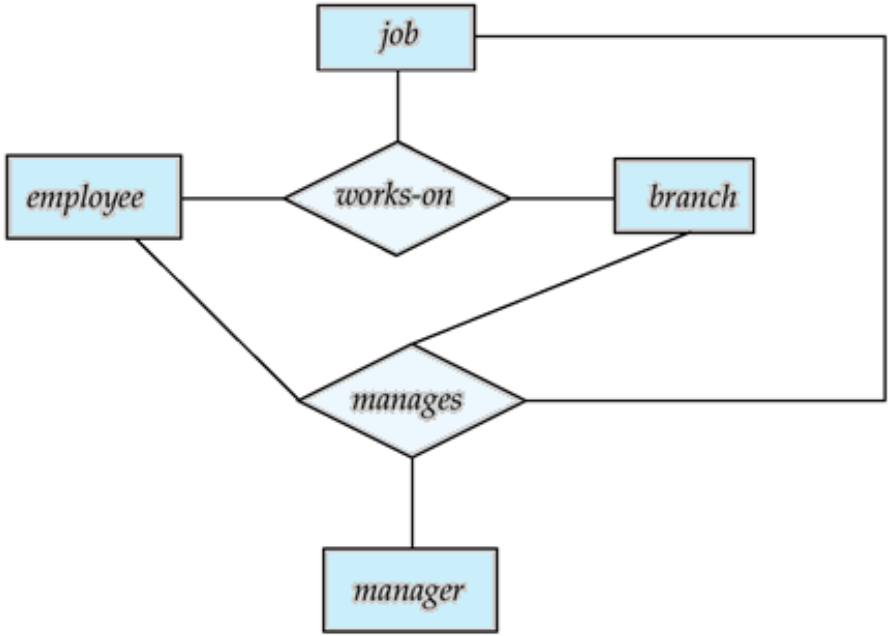
- We can specify total generalization in an ER diagram by adding the keyword total in the diagram and drawing a dashed line from the keyword to the corresponding hollow arrow-head to which it applies (for a total generalization), or to the set of hollow arrow-heads to which it applies (for an overlapping generalization)

- The ***student*** generalization is total: All ***student*** entities must be either graduate or undergraduate
  - Because the higher-level entity set arrived at through generalization is generally composed of only those entities in the lower-level entity sets, the completeness constraint for a generalized higher-level entity set is usually total

Aggregation

- Consider the ternary relationship ***proj_guide***, which we saw earlier
- Suppose we want to evaluations of a ***student*** by a ***guide*** on a ***project***

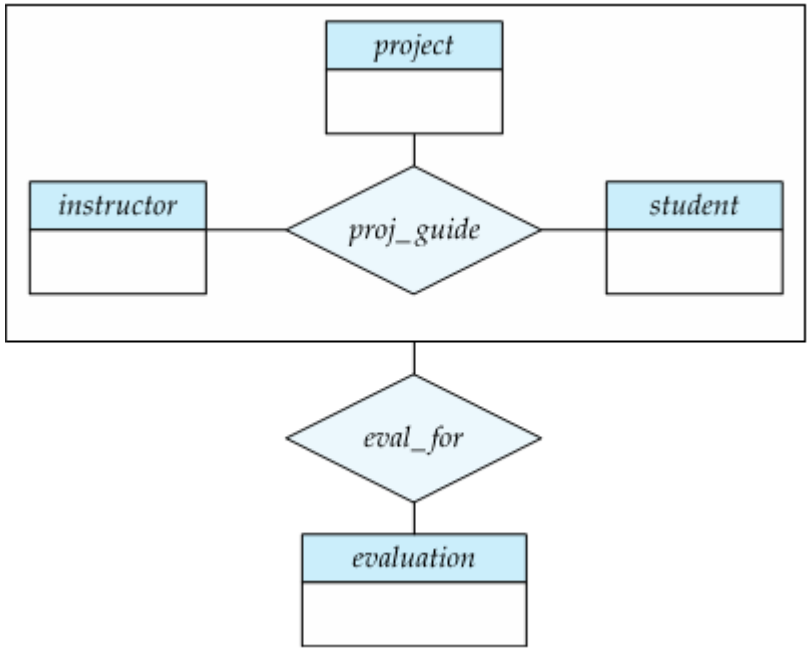


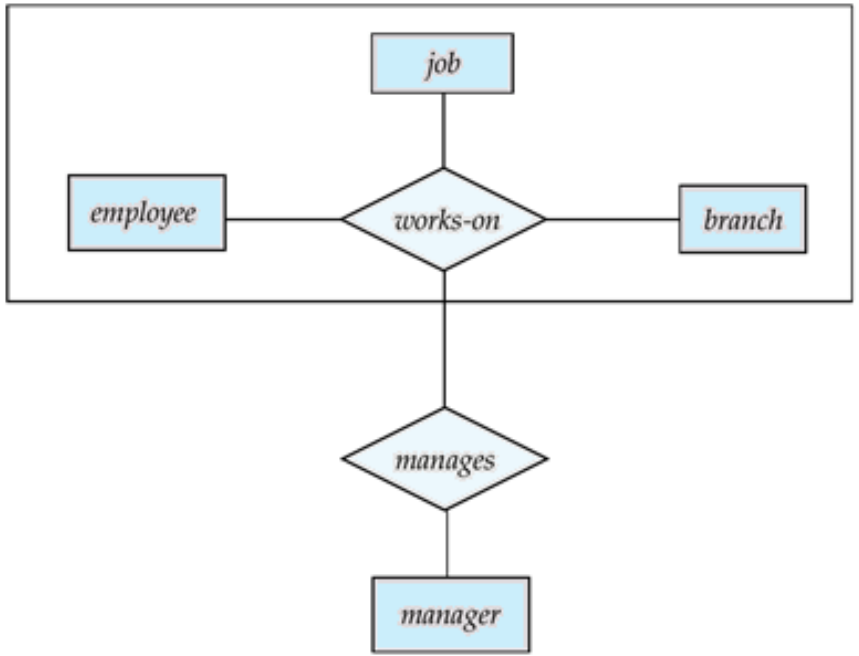


Aggregation

- Relationship sets ***eval_for*** and ***proj_guide*** represent overlapping information
  - Every ***eval_for*** relationship corresponds to a ***proj_guide*** relationship
  - However, some ***proj_guide*** relationships may not correspond to any ***eval_for*** relationships
    - So we can't discard the ***proj_guide*** relationship
- Eliminate this redundancy via ***aggregation***
  - Treat relationship as an abstract entity
  - Allows relationships between relationships
  - Abstraction of relationship into new entity
- Eliminate this redundancy via ***aggregation*** without introducing redundancy, the following diagram represents:
  - A ***student*** is guided by a particular ***instructor*** on a particular ***project***
  - A ***student, instructor, project*** combination may have an associated ***evaluation***

ER Diagram with Aggregation





Reduction to Relational Schemas

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

- **Example1:**
- The schema ***eval_for*** is:

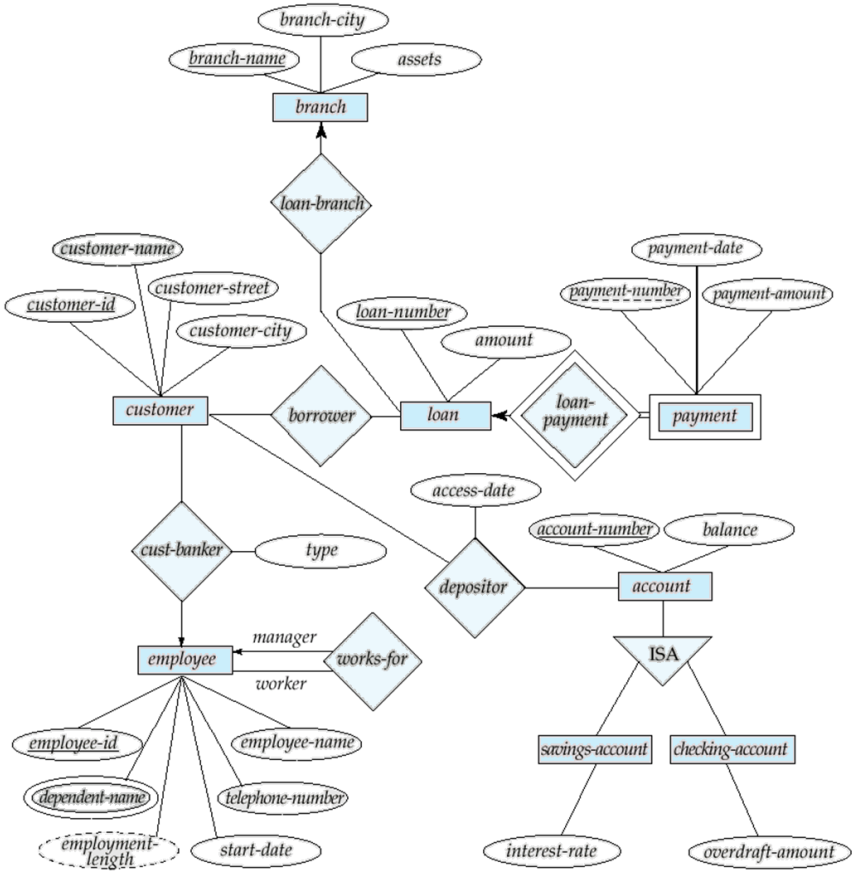
eval_for (s_ID, project_id, i_ID, evaluation_id)

- The schema ***proj_guide*** is redundant
- **Example2:**
- To represent aggregation ***manages*** between relationship ***works-on*** and entity set ***manager***, create a table:

manages(employee-id, branch-name, title, manager-name)

- Table ***works-on*** is redundant provided we are willing to store null values for attribute ***manager-name*** in table ***manages***

ER Diagram for a Banking Enterprise



Next Lecture

ER Diagram Design Issues

Thank you for your attention...

Any question?

Contact:
Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13
Phone: +91-9477678768
E-mail: shrutilipi@nitk.edu.in