UMBC

AN HONORS UNIVERSITY IN MARYLAND

CMSC 461, Database Management Systems
Spring 2018

Lecture 10 - Chapter 7 Entity Relationship Model

These slides are based on "Database System Concepts" 6th edition book and are a modified version of the slides which accompany the book (http://codex.cs.yale.edu/avi/db-book/db6/slide-dir/index.html), in addition to the 2009/2012 CMSC 461 slides by Dr. Kalpakis

Dr. Jennifer Sleeman

https://www.csee.umbc.edu/~jsleem1/courses/461/spr18

Logistics

- HW2 key will be available Mon March 5th
- Start phase 2 sooner vs. later due Wed 3/7/2018
- HW3 will be released this week
- See the project page for a description of the rest of the phases

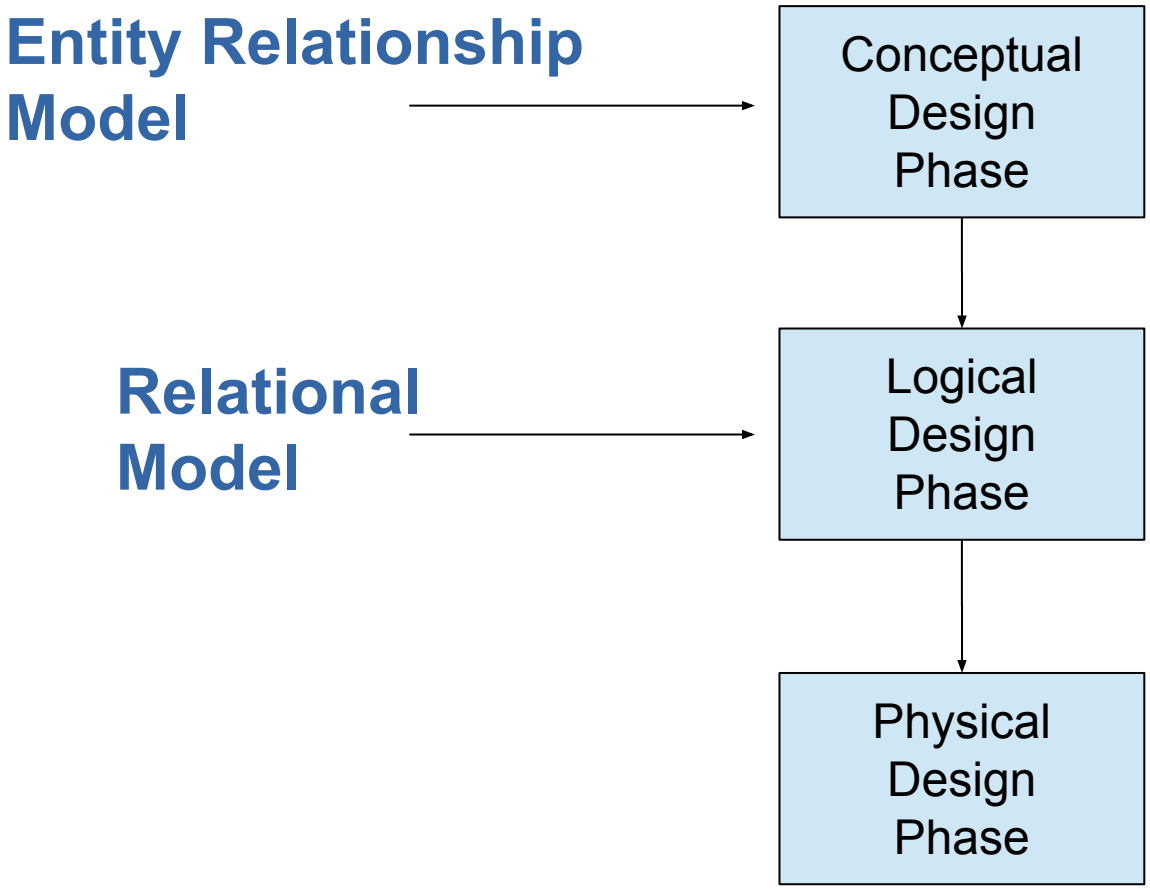
Lecture Outline

- Review and Clarification
- Reduction to Relational Schemas
- Design Issues
- Extended ER Features
- Database Design Tools

Lecture Outline

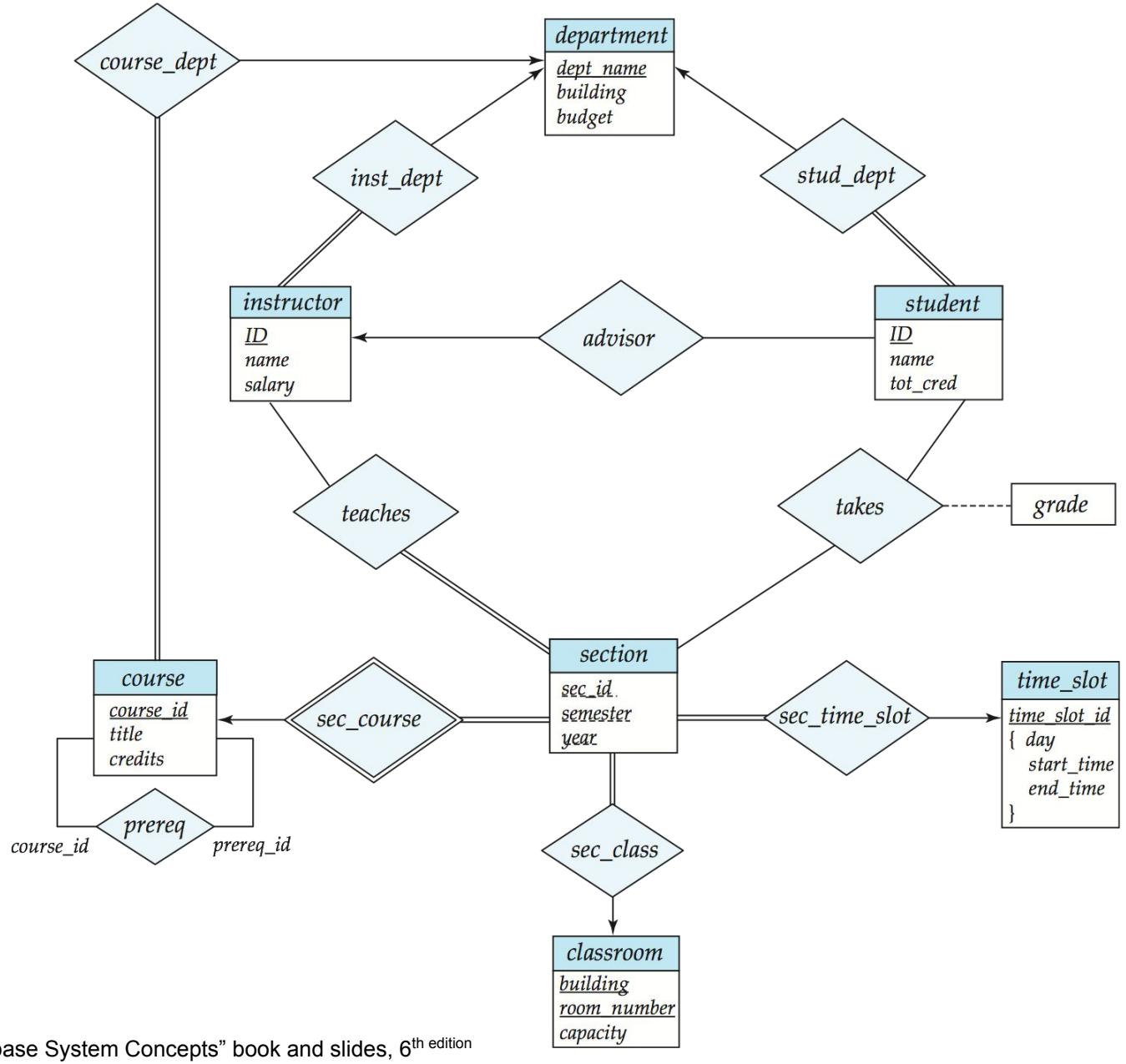
- ***Review and Clarification***
- Reduction to Relational Schemas
- Design Issues
- Extended ER Features
- Database Design Tools

Database Design Phases



Three levels of modeling

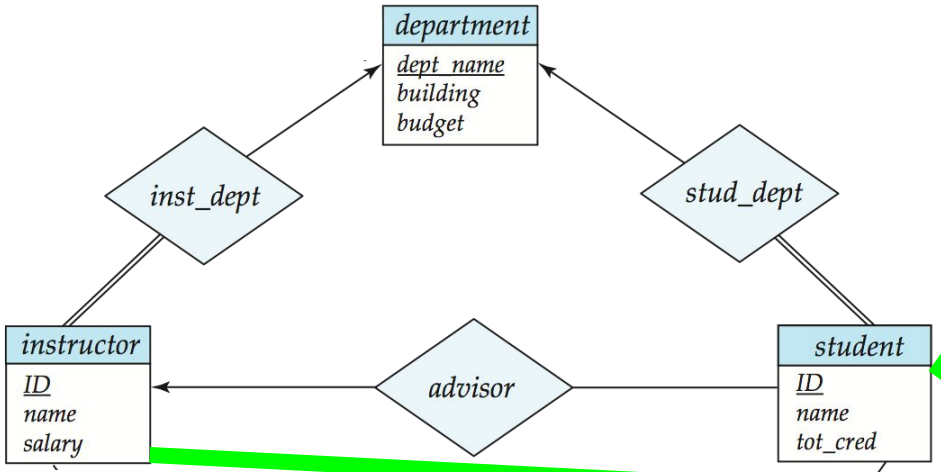
Redundancy Revisited

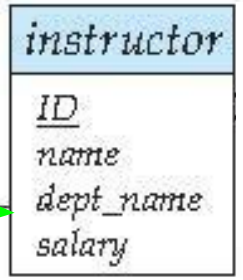


Based on and image from "Database System Concepts" book and slides, 6th edition

Redundancy Revisited

- We first identify entity sets
- Then we choose identifying attributes
- And relationship sets among entities
- This results in redundant attributes across entity sets
- Try to remove
  - During the ER modeling phase

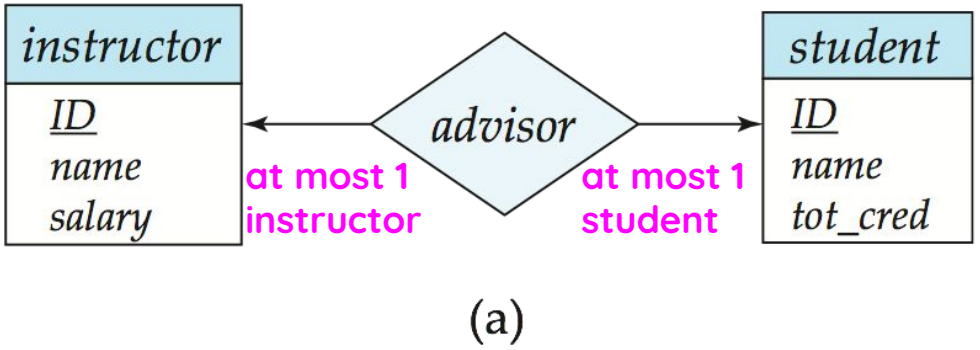




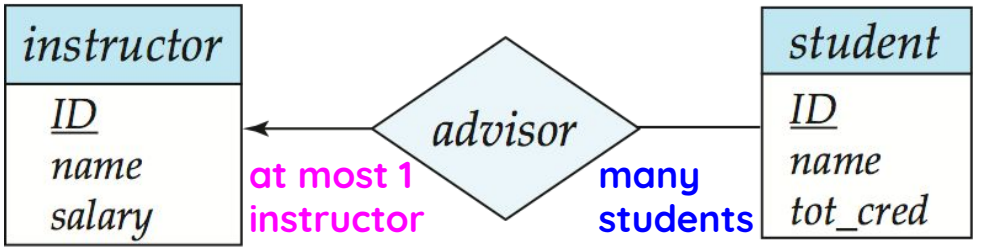
Based on and image from "Database System Concepts" book and slides, 6th edition

Cardinality Revisited

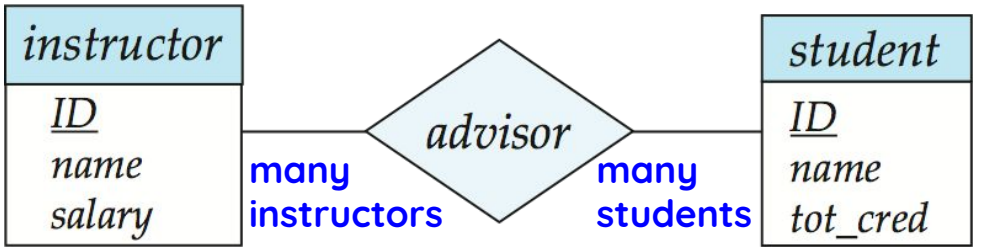
- draw directed line (→)
  - signifying “one”
- undirected line (—),
  - signifying “many”
- Between the relationship set and the entity set



(a)



(b)



(c)

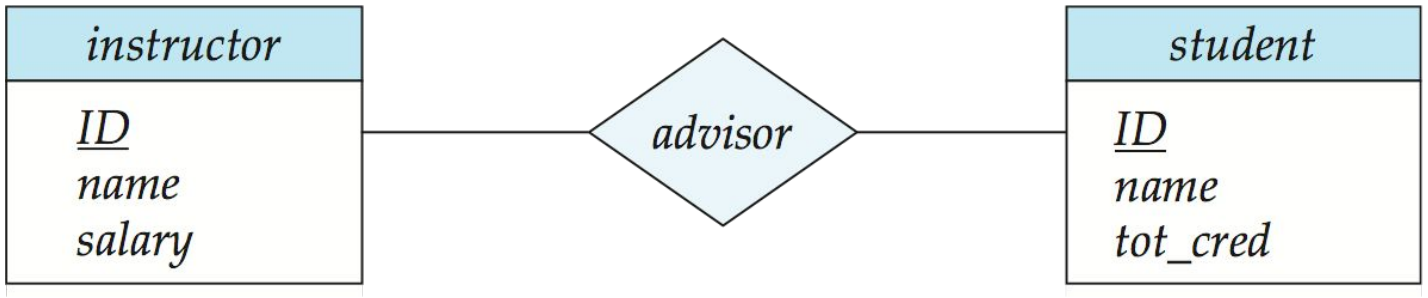
Based on and image from “Database System Concepts” book and slides, 6th edition

Total vs. Partial Participation Revisited

- There could be an entity that does not participate in a relationship set
- This means there is partial participation
- If we want to indicate that there must be total participation, use the double line:

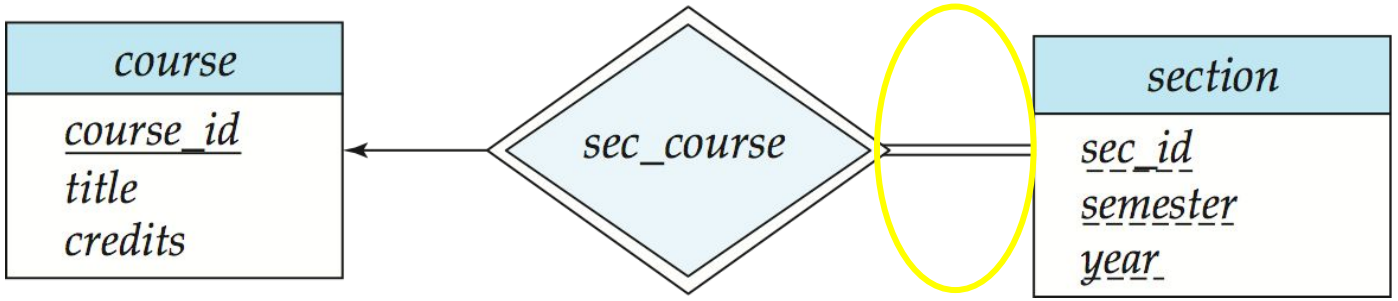
Total vs. Partial Participation Revisited

- Some entities may not participate in any relationship in the relationship set
  - Example: participation of *instructor* in *advisor* is partial

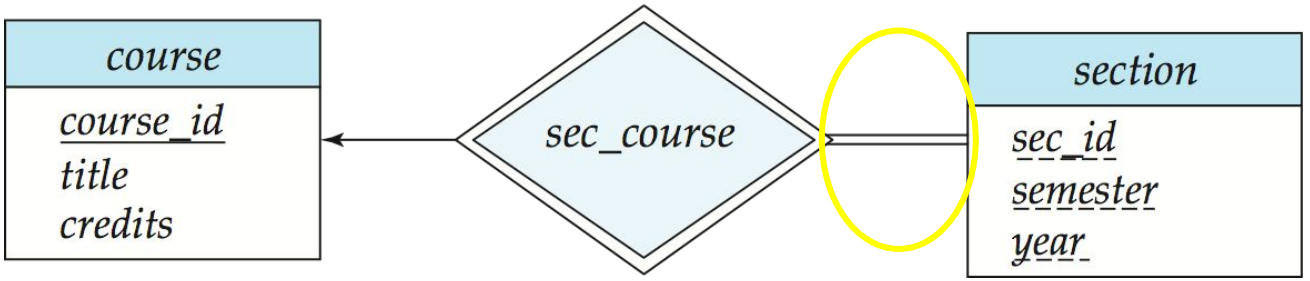


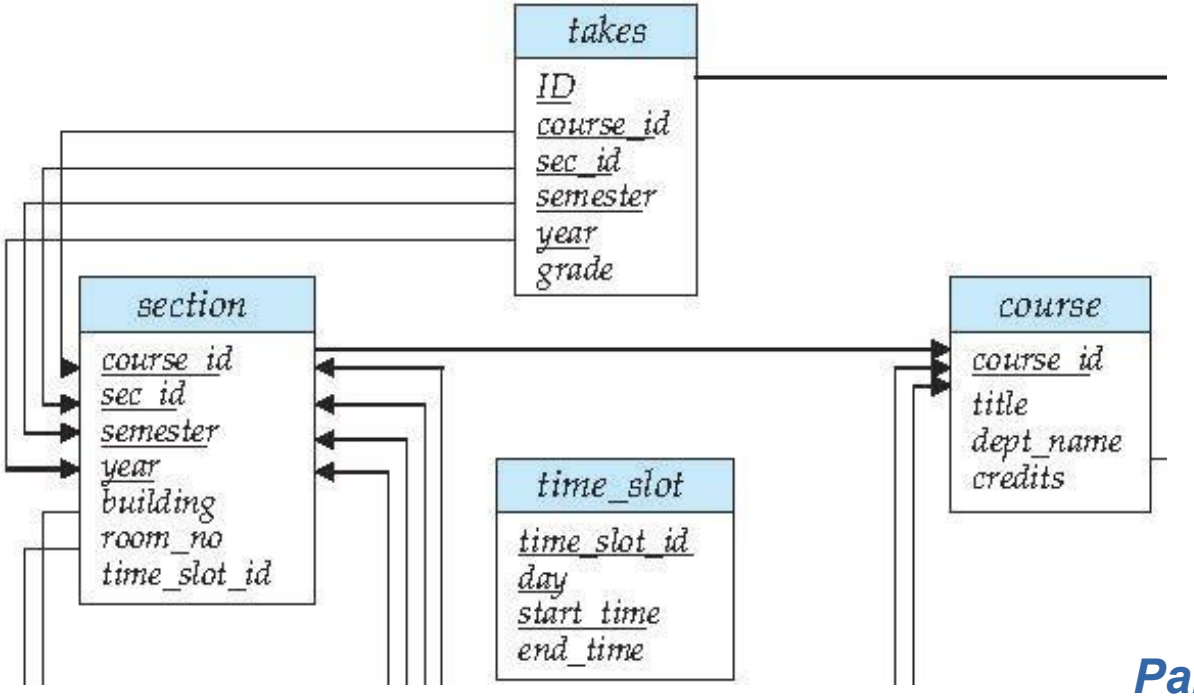
Total vs. Partial Participation Revisited

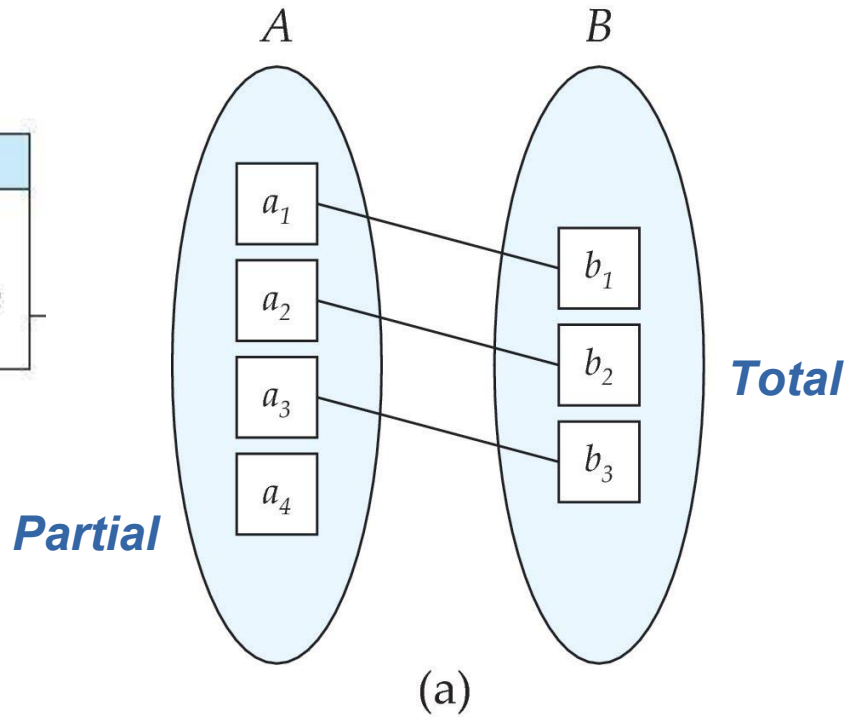
- Use double lines
- Every entity in entity set participates in at least one relationship in the relationship set
  - Example: participation of *section* in *sec_course* is total



Total vs. Partial Participation Revisited

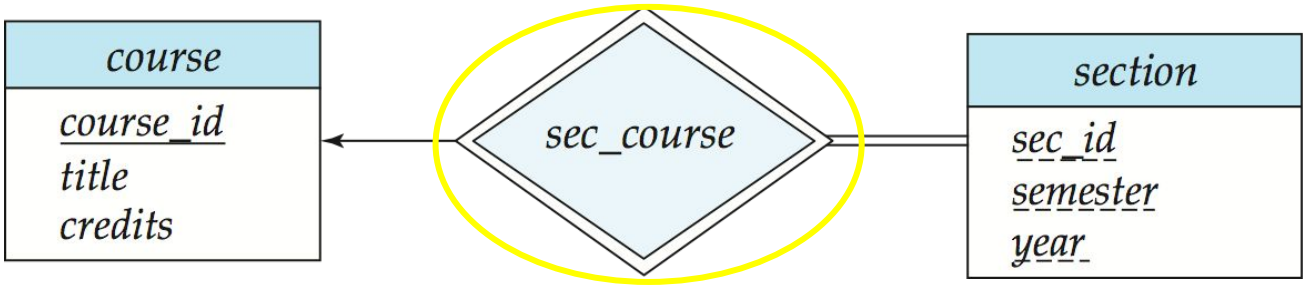


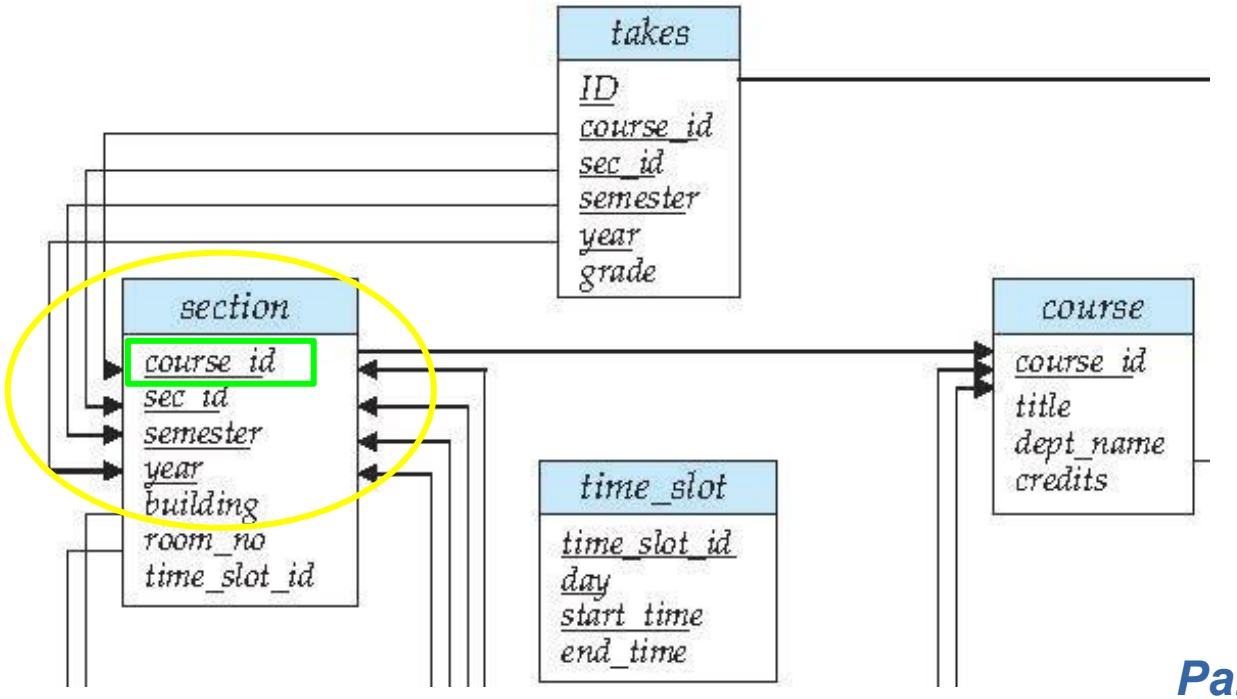


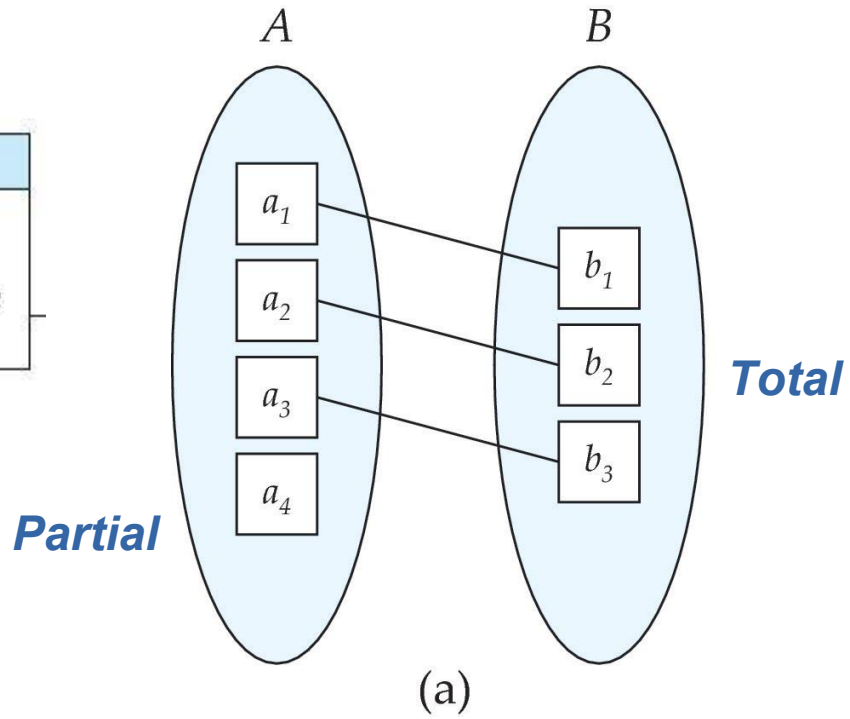


Based on and image from "Database System Concepts" book and slides, 6th edition

Strong vs Weak Entity Sets Revisited







Based on and image from "Database System Concepts" book and slides, 6th edition

Lecture Outline

- Review and Clarification
- ***Reduction to Relational Schemas***
- Design Issues
- Extended ER Features
- Database Design Tools

Reduction to Relation Schemas

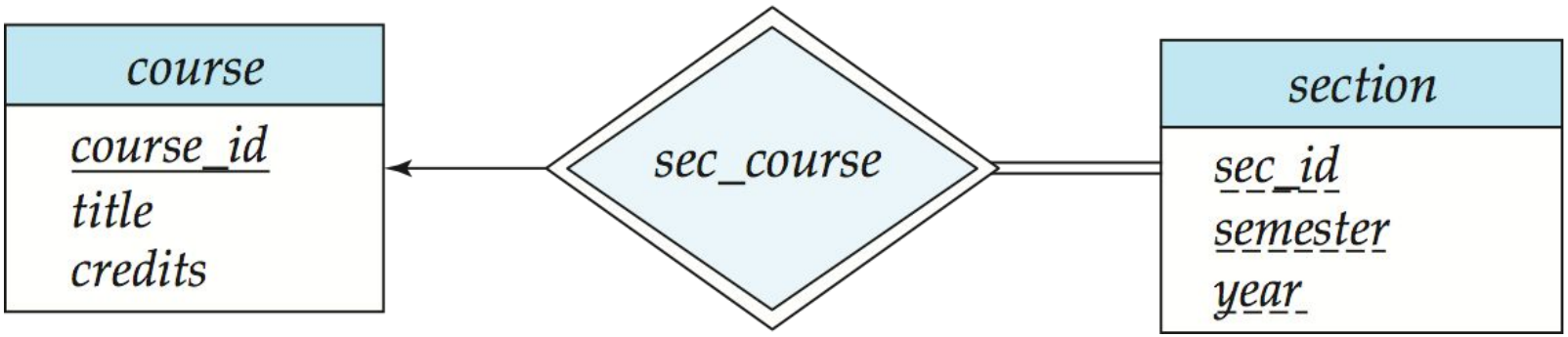
- Entity sets and relationship sets can be expressed as ***relation schemas***
  - represent the contents of the database
- Database conforms to an E-R diagram
  - represented by a collection of schemas

Reduction to Relation Schemas

- For each entity set and relationship set create a unique schema
  - name corresponds to entity set or relationship set
- For each schema create columns
  - corresponds to attributes

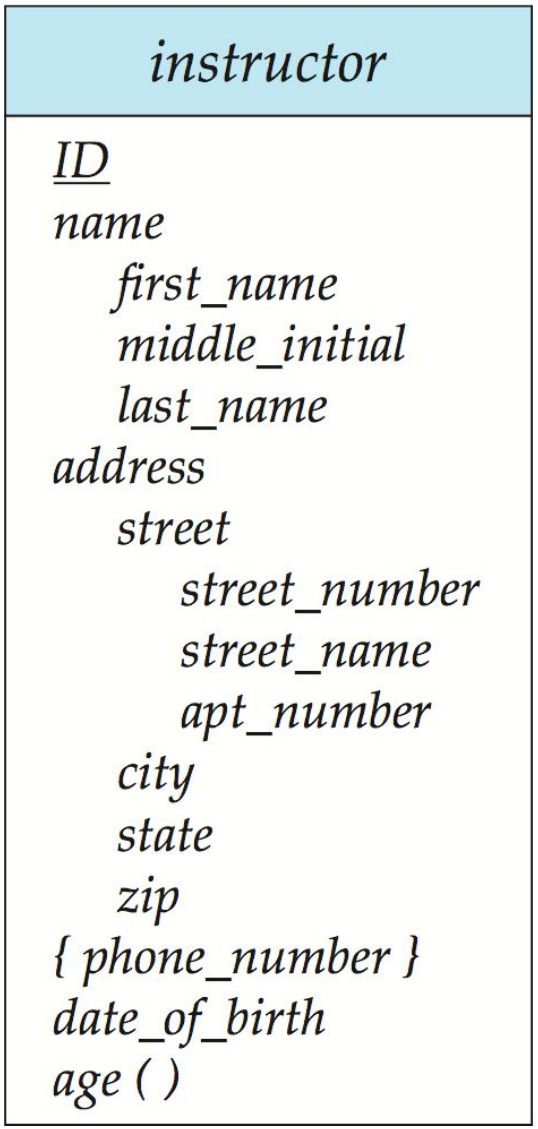
Representing Entity Sets With Simple Attributes

- A strong entity set reduces to a schema with the same attributes
  course(course_id, title, credits)
- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set
  section (course_id, sec_id, sem, year)



Based on and image from "Database System Concepts" book and slides, 6th edition

Composite and Multivalued Attributes



- Composite attributes are flattened out by creating a separate attribute for each component attribute
  - Example: given entity set *instructor* with composite attribute *name* with component attributes *first_name* and *last_name* the schema corresponding to the entity set has two attributes *name_first_name* and *name_last_name*
    - *Prefix omitted if there is no ambiguity*
- Ignoring multivalued attributes, extended instructor schema is
  - *instructor(ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth)*

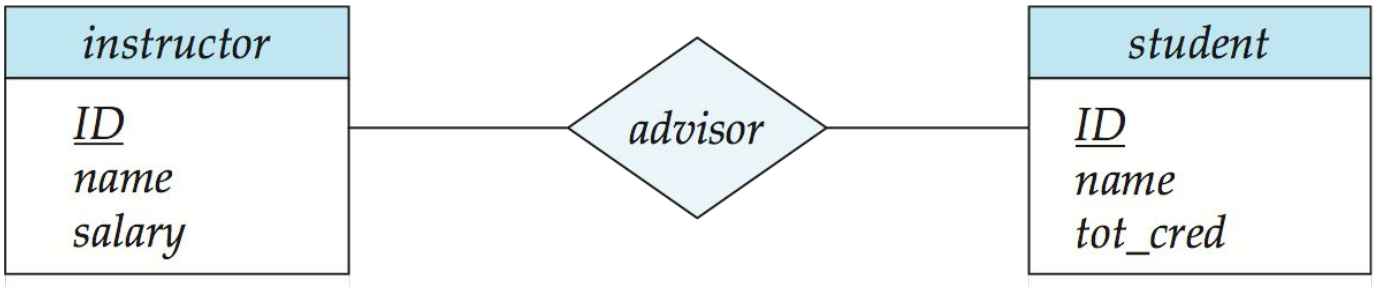
Composite and Multivalued Attributes

- A multivalued attribute *M* of an entity *E* is represented by a separate schema *EM*
  - Schema *EM* has attributes corresponding to the primary key of *E* and an attribute corresponding to multivalued attribute *M*
  - Example: Multivalued attribute *phone_number* of *instructor* is represented by a schema:
    inst_phone= (*ID*, *phone_number*)
  - Each value of the multivalued attribute maps to a separate tuple of the relation on schema *EM*
    - For example, an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples:
      (22222, 456-7890) and (22222, 123-4567)

Representing Relationship Sets

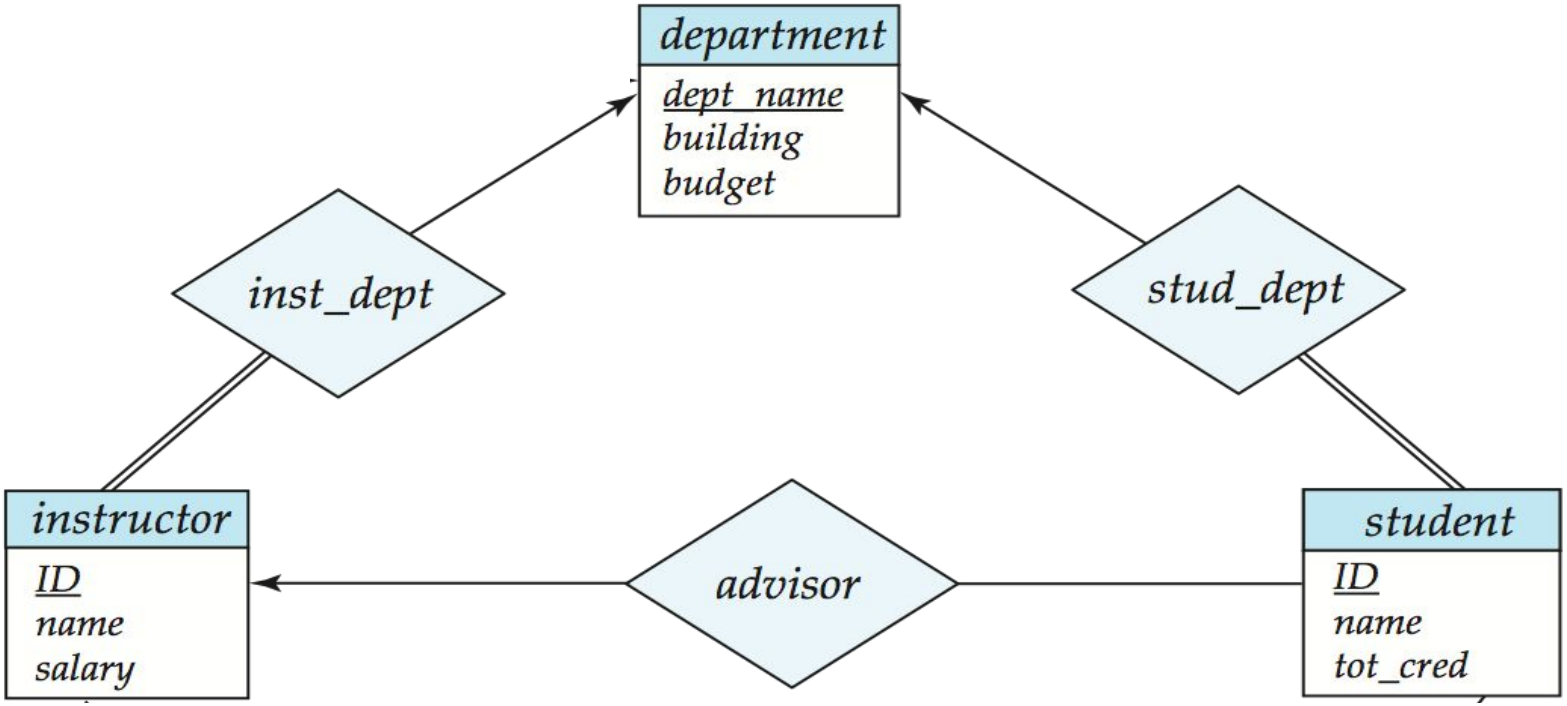
- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.
- Example: schema for relationship set *advisor*

  advisor = (*s_id*, *i_id*)



Representing Relationship Sets

- Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the "many" side, containing the primary key of the "one" side
- Example: Instead of creating a schema for relationship set *inst_dept*, add an attribute *dept_name* to the schema arising from entity set *instructor*

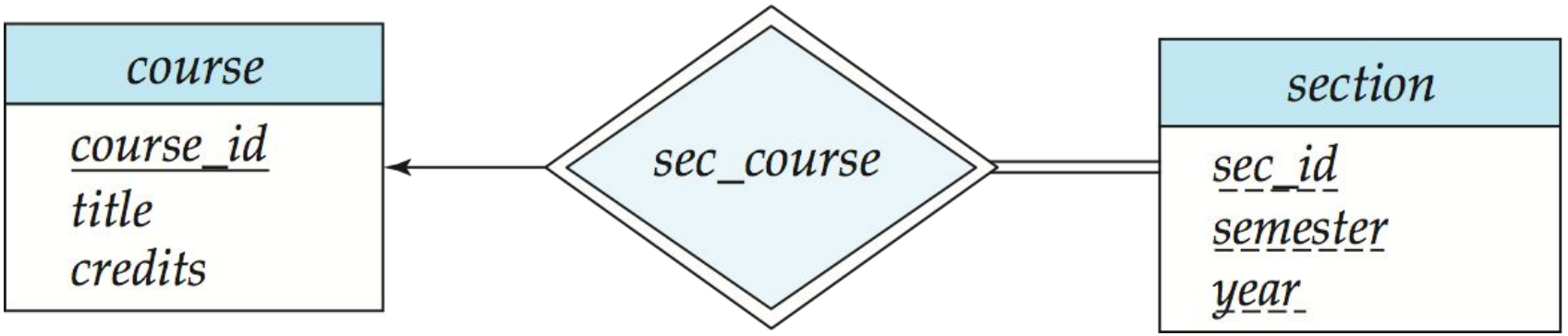


Representing Relationship Sets

- For one-to-one relationship sets, either side can be chosen to act as the “many” side
  - That is, extra attribute can be added to either of the tables corresponding to the two entity sets
- If participation is *partial* on the “many” side, replacing a schema by an extra attribute in the schema corresponding to the “many” side could result in null values

Redundancy of Schemas

- Schema for the relationship set linking a weak entity set to its corresponding strong entity set is redundant and doesn't need to be present in a relational database design



Combination of Schemas

- Consider a many-to-one relationship set AB
  - where A is an entity set and B is an entity set
- This would result in 3 schemas
  - A
  - B
  - AB
- If participation of A is total, then we can combine A and AB
  - Union of attributes between the 2 schemas

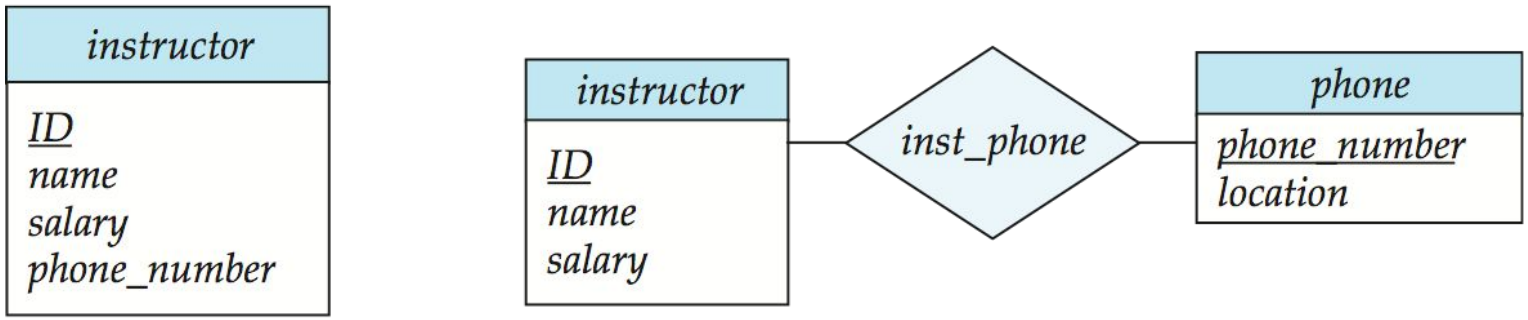
Example: A= instructor, B=department, schema *inst_dept* can be combined with the instructor schema
(ID, name, dept_name, salary)

Lecture Outline

- Review and Clarification
- Reduction to Relational Schemas
- ***Design Issues***
- Extended ER Features
- Database Design Tools

Design Issues – Entity Sets vs. Attributes

- Sometimes attributes can be seen as an entity on its own
  - Example: Use of phone as an entity allows extra information about phone numbers (plus multiple phone numbers)
- If we treat it as an attribute, constraints how many values for that attribute
- Else the concept of a phone number becomes more generalized



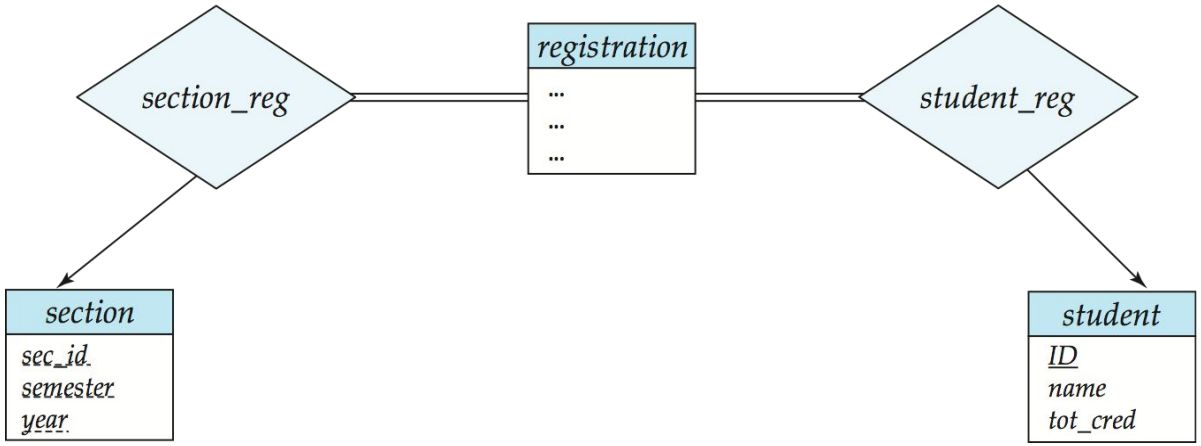
Based on and image from "Database System Concepts" book and slides, 6th edition

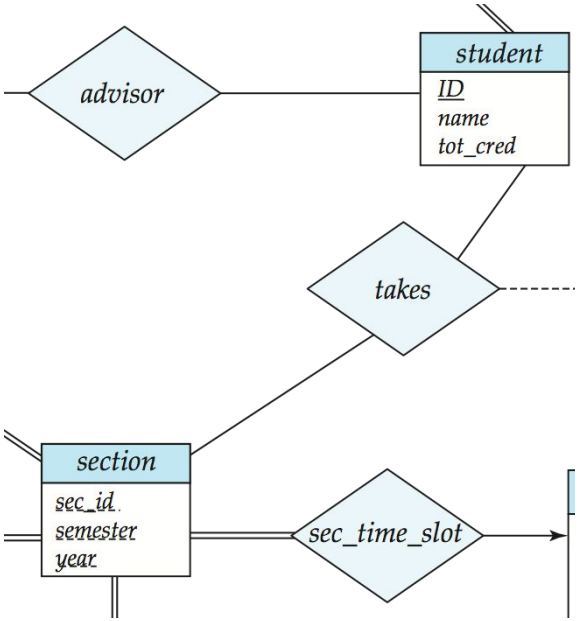
Design Issues – Entity Sets vs. Attributes

- Key Mistakes:
  - Using primary key of an entity set as an attribute of another entity set instead of a relationship
    - Student ID in Instructor relation
  - Designation of primary keys as attributes of the relationship set
    - Implied by the relationship set

Design Issues – Entity Sets vs. Relationship Sets

- Possible guideline is to designate a relationship set to describe an action that occurs between entities





Here we used a registration entity set instead of the takes relationship set.

Lecture Outline

- Review and Clarification
- Reduction to Relational Schemas
- Design Issues
- ***Extended ER Features***
- Database Design Tools

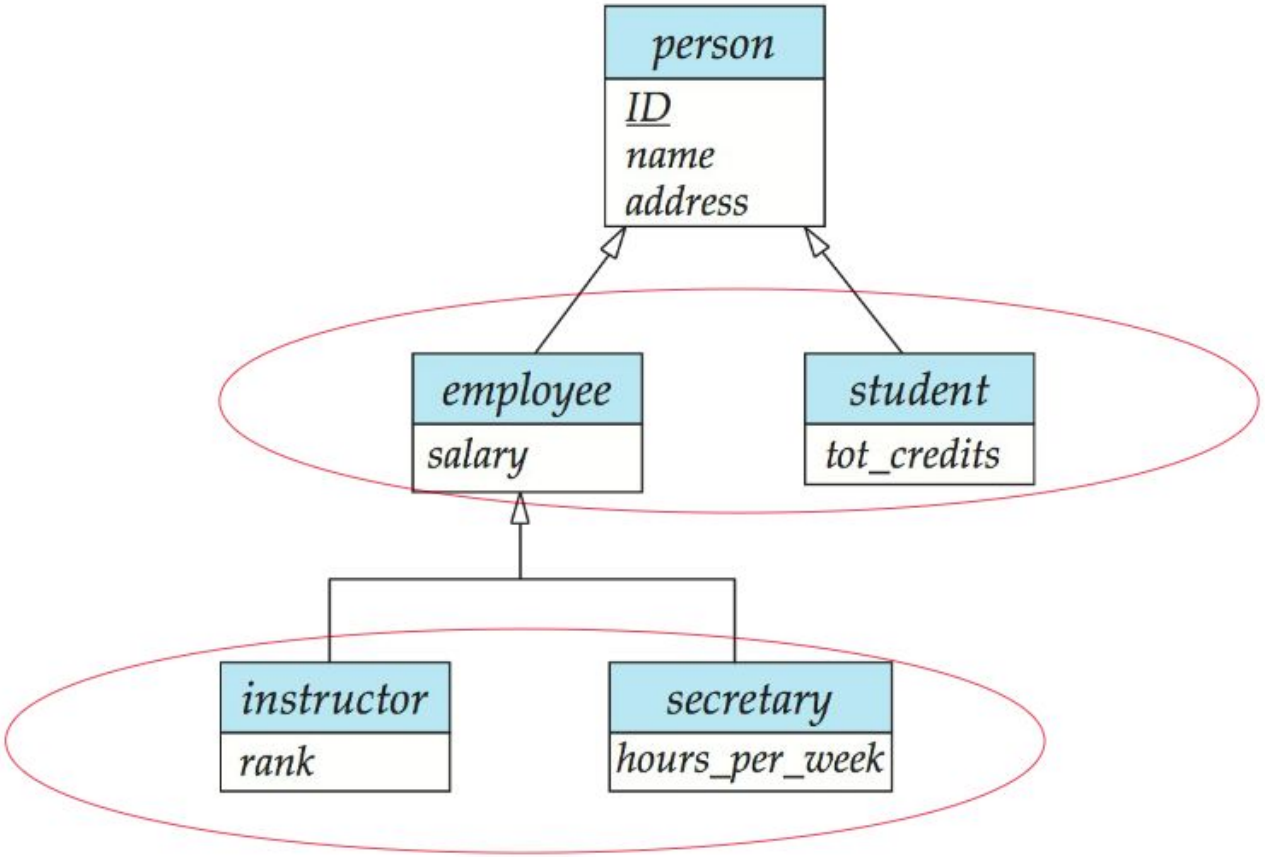
Extended ER Features

- Extensions to basic concepts
  - Specialization
  - Generalization
  - Higher/Lower level entity sets
  - Attribute Inheritance
  - Aggregation

Specialization

- ***Top down design process*** - designate subgroupings within an entity set that are distinctive from other entities in the set
- These subgroupings 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 ISA (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

Specialization



Based on and image from "Database System Concepts" book and slides, 6th edition

Specialization

- Entity set may be specialized by more than one distinguishing feature
  - Entities can belong to multiple specializations
- ***Overlapping specialization*** – multiple sets permitted
- ***Disjoint specialization*** – At most one specialized entity set

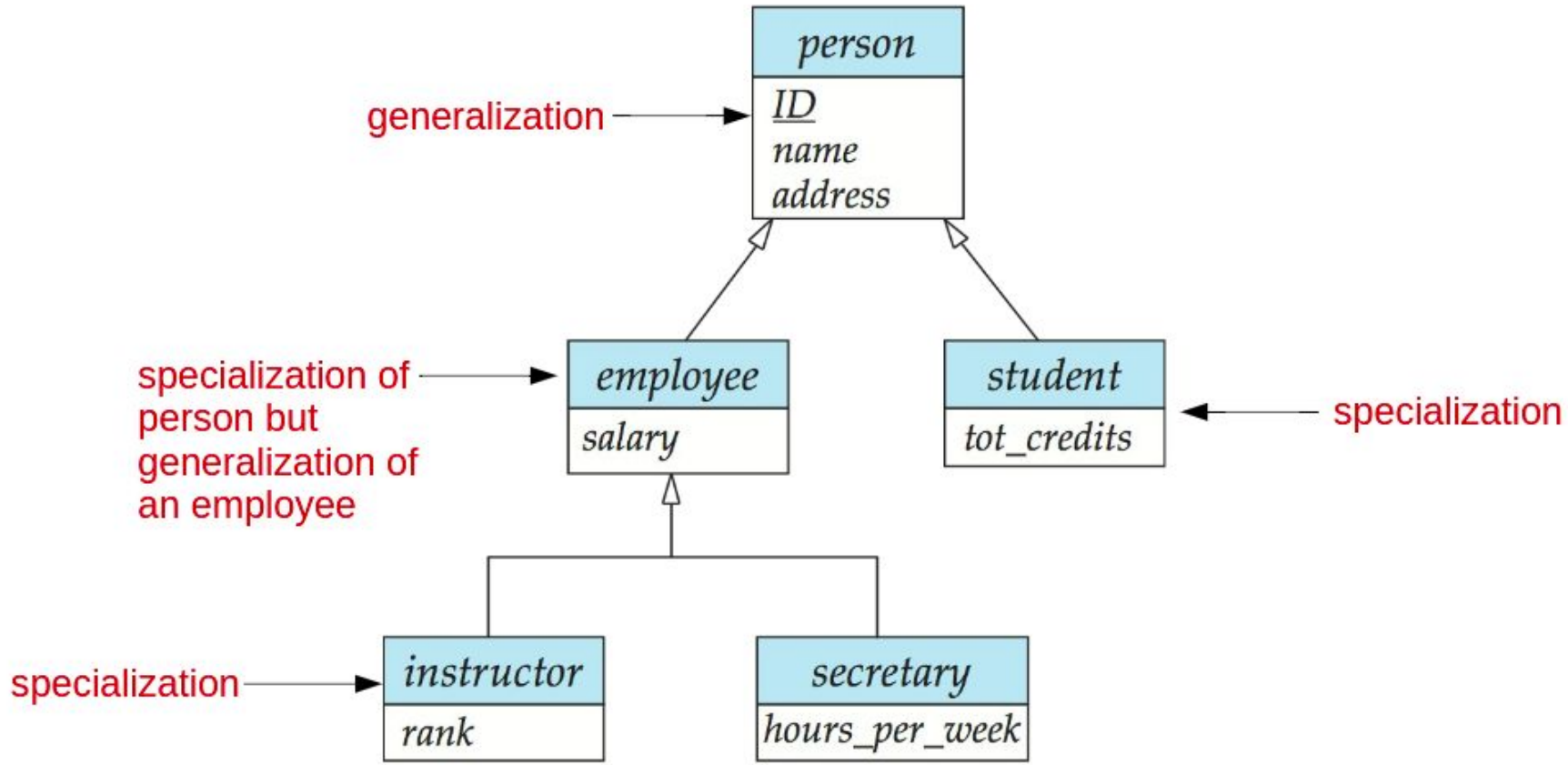
Generalization

- ***A bottom-up design process***
  - a number of entity sets that share same features
  - combine into higher-level entity set
- Specialization and generalization inversions of each other
  - represented in an E-R diagram in same way
- Specialization and generalization are used interchangeably

Generalization

- Can have multiple specializations of an entity set based on different features
  - permanent_employee vs. temporary_employee and instructor vs. secretary
- Each particular employee would be a member of
  - one of permanent_employee or temporary_employee
  - and also a member of instructor or secretary
- The ISA relationship also referred to as superclass/subclass relationship

Generalization vs Specialization

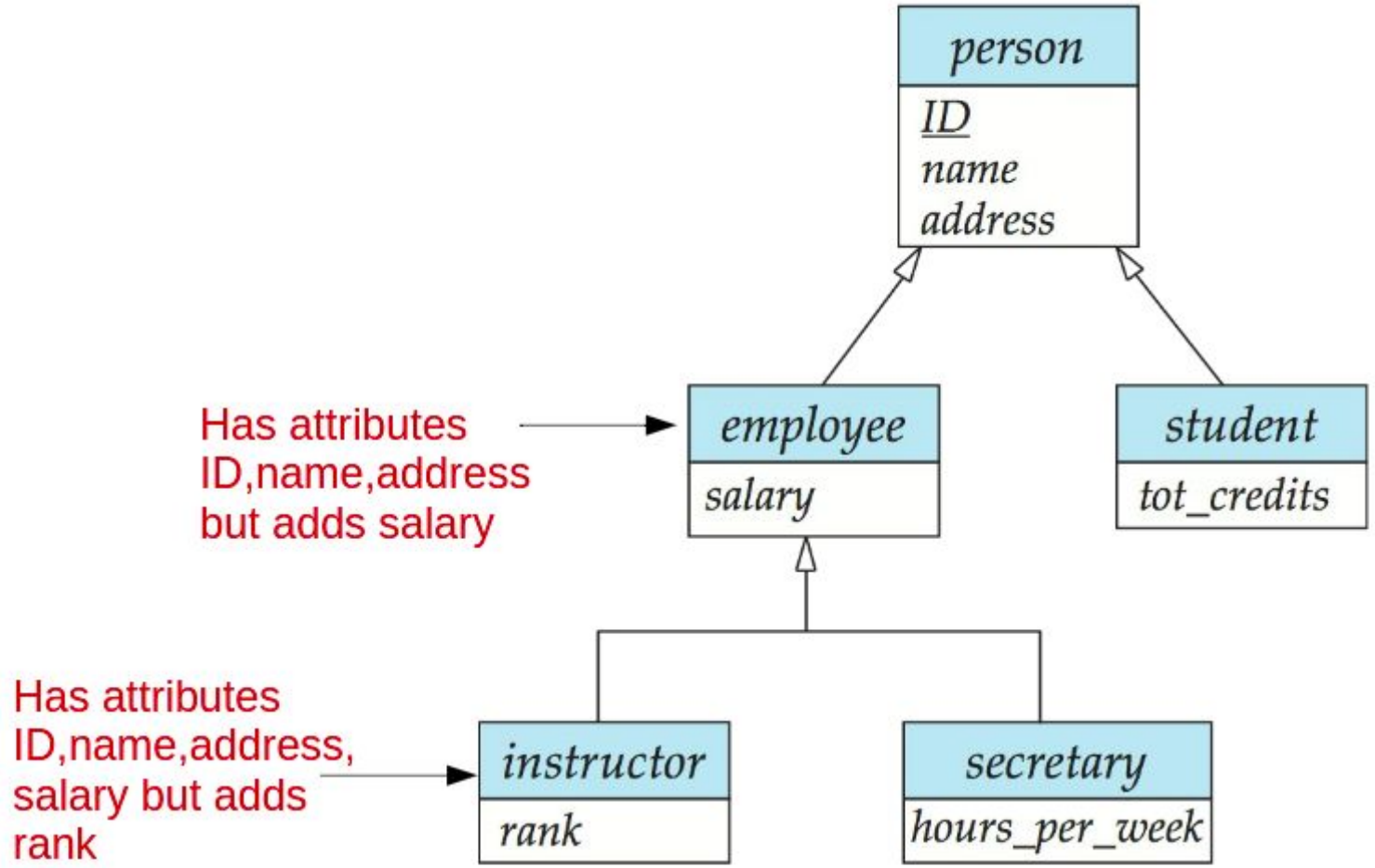


Based on and image from "Database System Concepts" book and slides, 6th edition

Attribute Inheritance

- Attributes of higher level entity sets inherited by lower level entity sets
- Participation also inherited
- Outcome:
  - High-level entity set with attributes and relationships, apply to low-level entity set
  - Lower-level entity sets with distinctive features

Attribute Inheritance

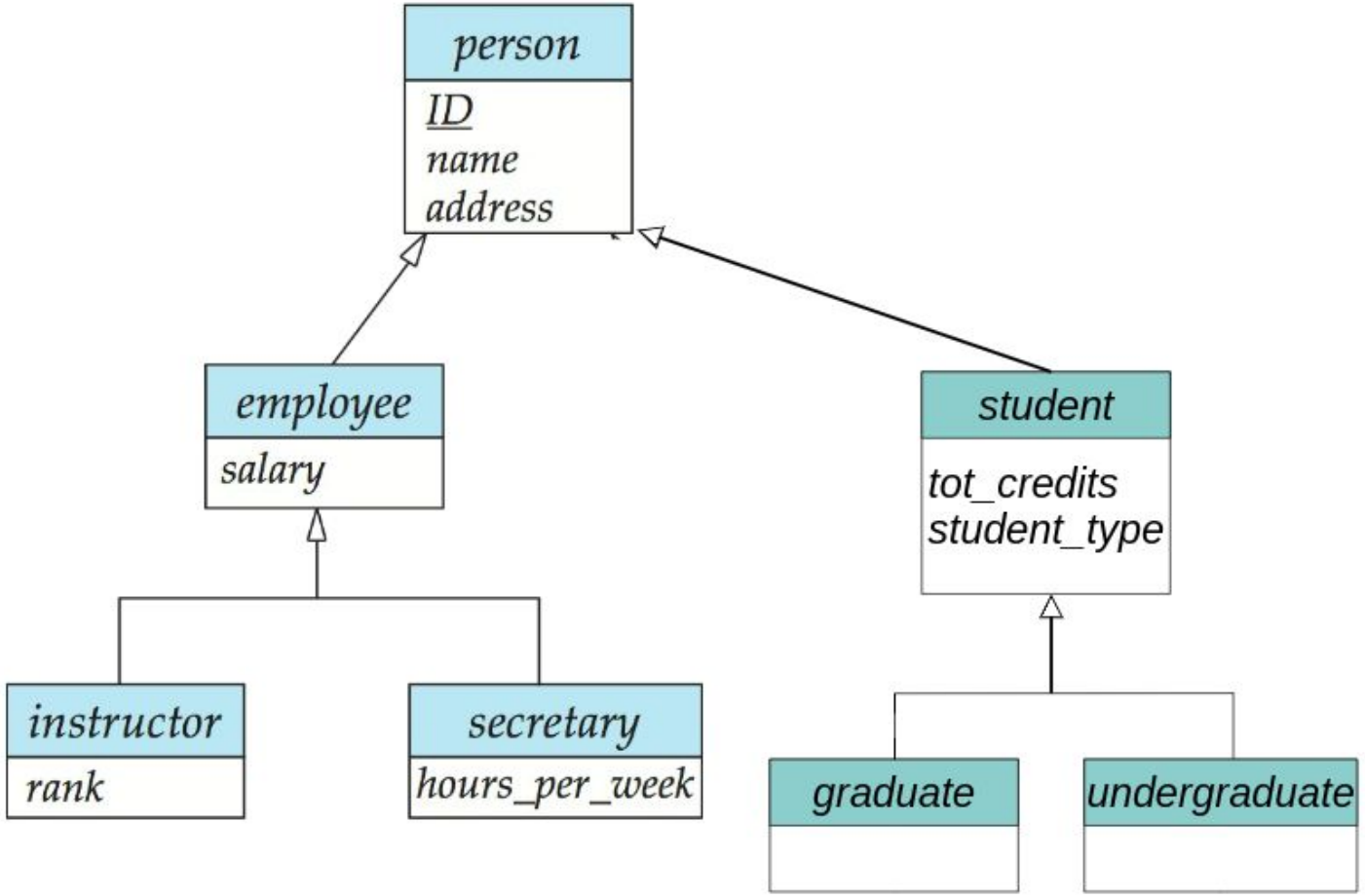


Based on and image from "Database System Concepts" book and slides, 6th edition

Specialization/Generalization Design Constraints

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

Specialization/Generalization Design Constraints



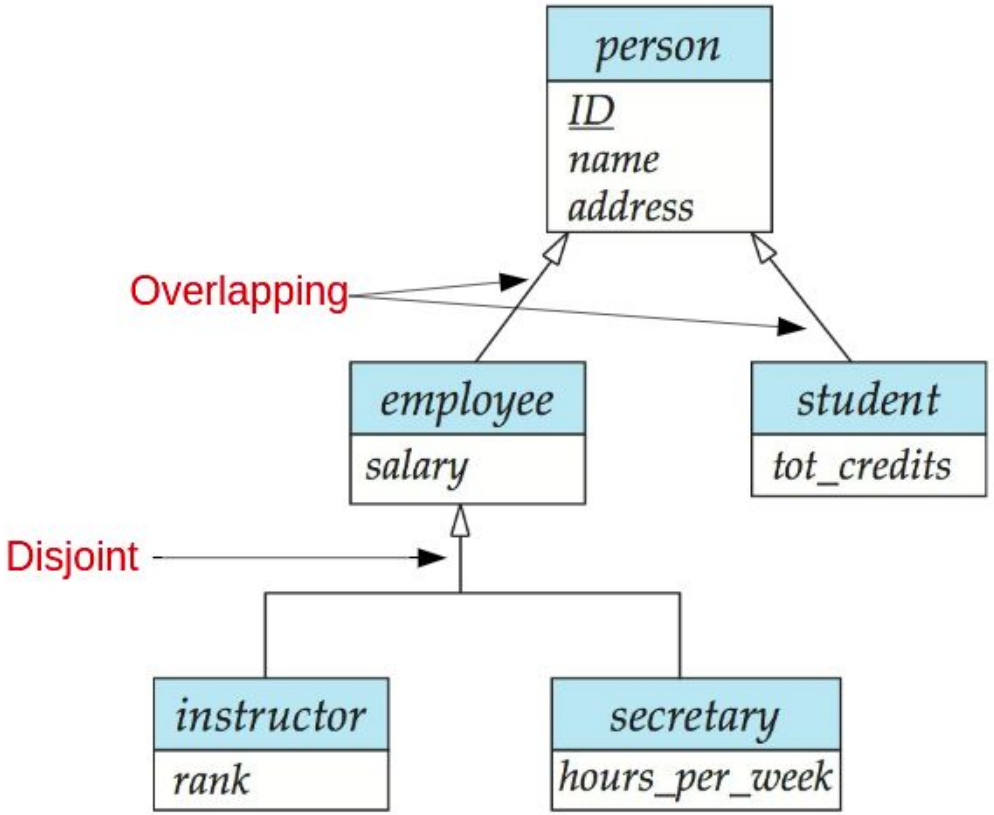
Conditioned: if
student_type='graduate' then type graduate or if
student_type='undergrad' then type undergraduate

Based on and image from "Database System Concepts" book and slides, 6th edition

Specialization/Generalization Design Constraints

- 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 E-R diagram by having multiple lower-level entity sets link to the same triangle
  - ***Overlapping***
    - an entity can belong to more than one lower-level entity set

Specialization/Generalization Design Constraints

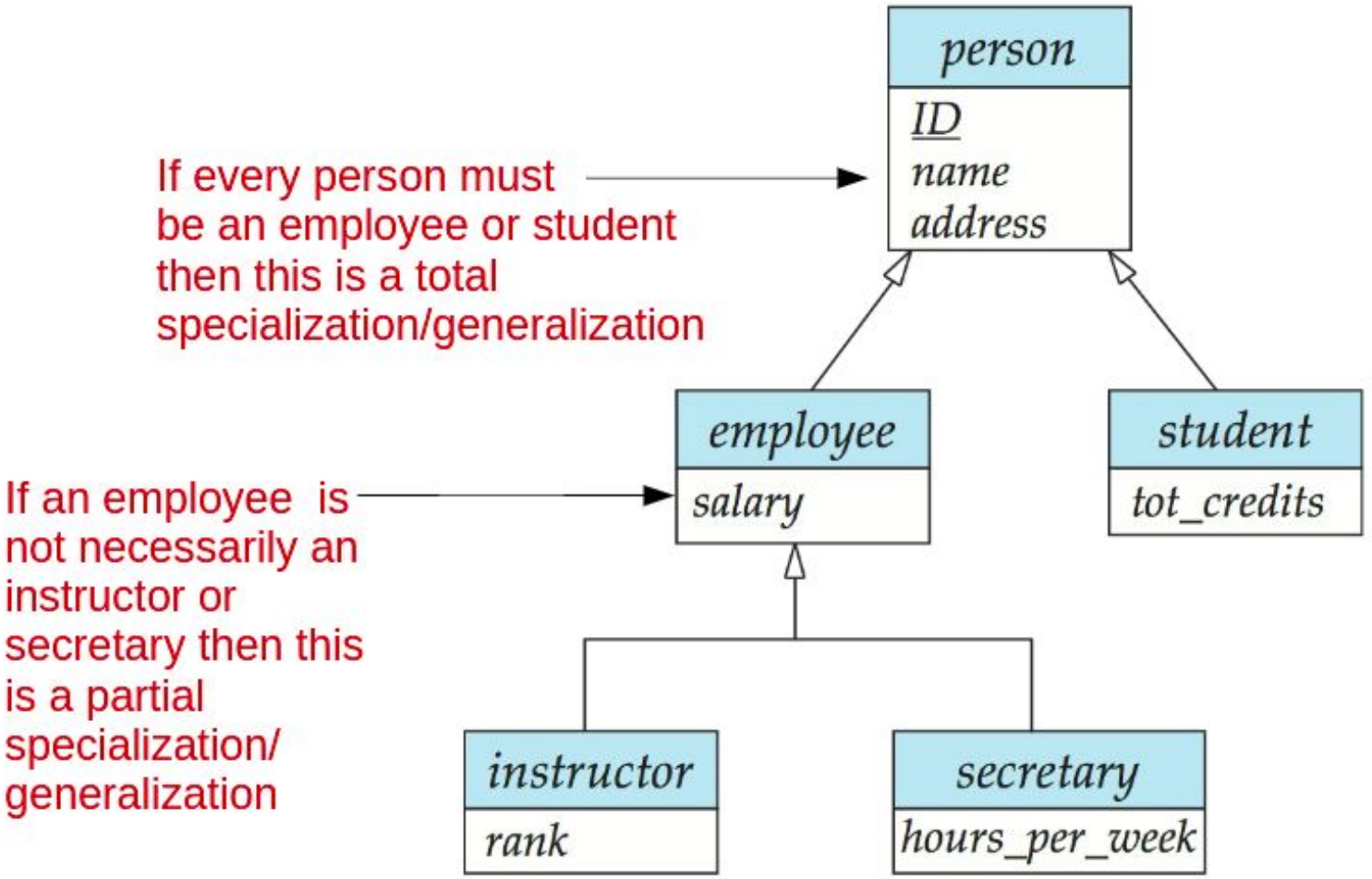


Based on and image from "Database System Concepts" book and slides, 6th edition

Specialization/Generalization Design Constraints

- 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

Specialization/Generalization Design Constraints



Based on and image from "Database System Concepts" book and slides, 6th edition

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

Specialization via Schemas

Method 2:

- Form a schema for each entity set with all local and inherited attributes

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

- If specialization is total and disjoint, the schema for the generalized entity set (*person*) not required to store information
  - Can be defined as a "view" relation containing union of specialization relations
  - But explicit schema 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

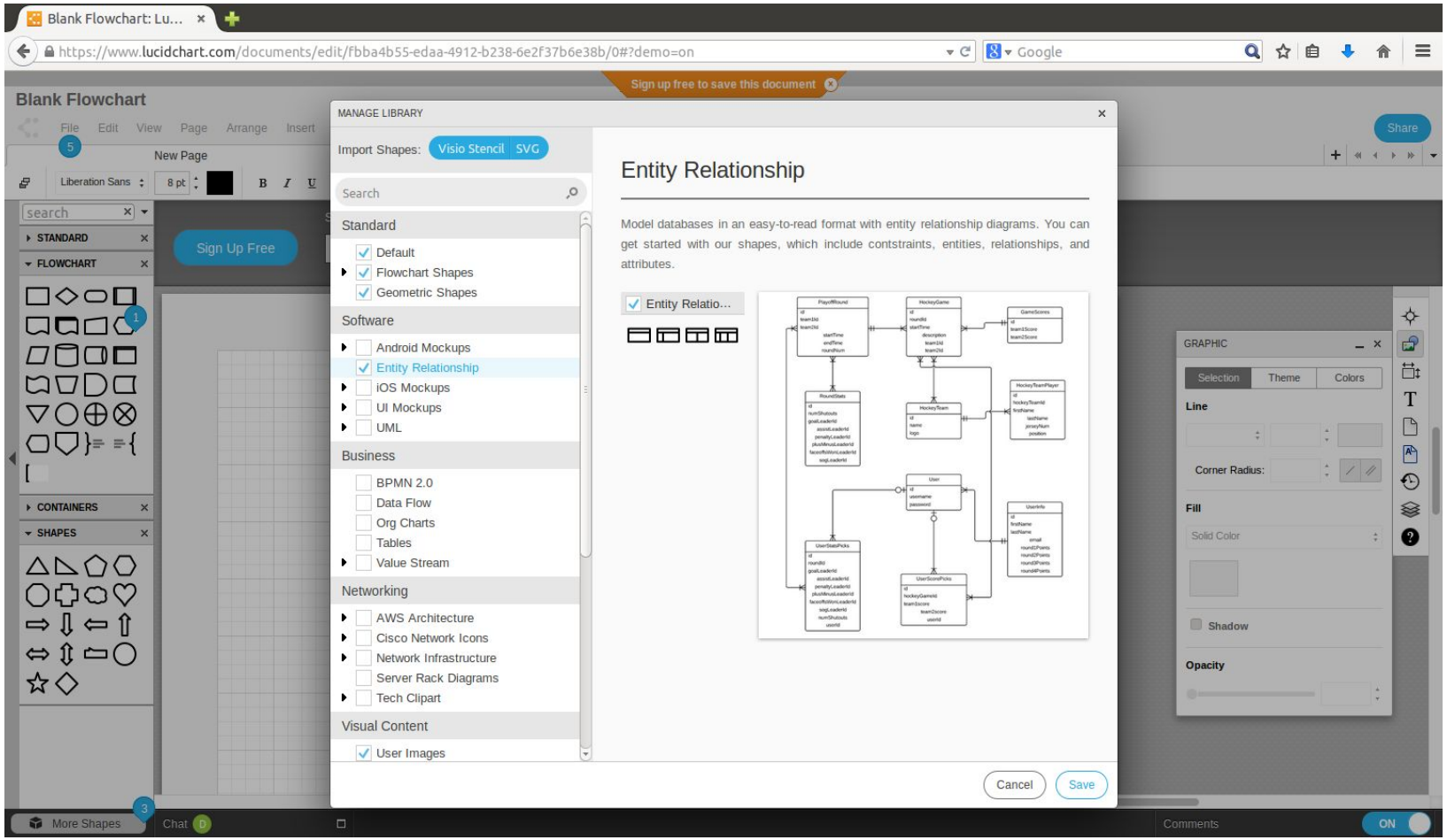
ER Design Decisions

- The use of an attribute or entity set to represent an object
- Whether a real-world concept is best expressed by an entity set or a relationship set
- The use of a strong or weak entity set
- The use of specialization/generalization – contributes to modularity in the design

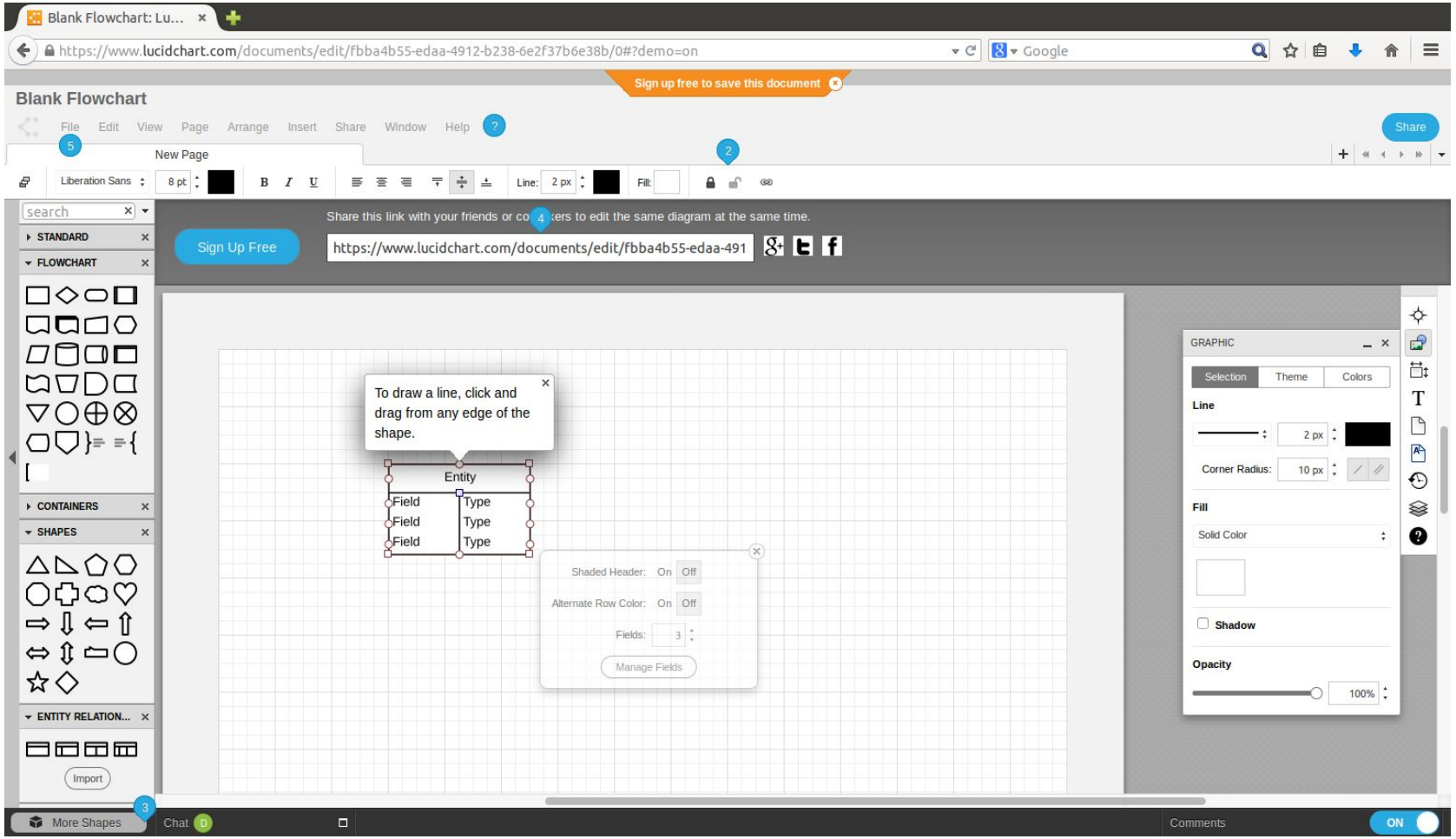
Lecture Outline

- Review and Clarification
- Reduction to Relational Schemas
- Design Issues
- Extended ER Features
- ***Database Design Tools***

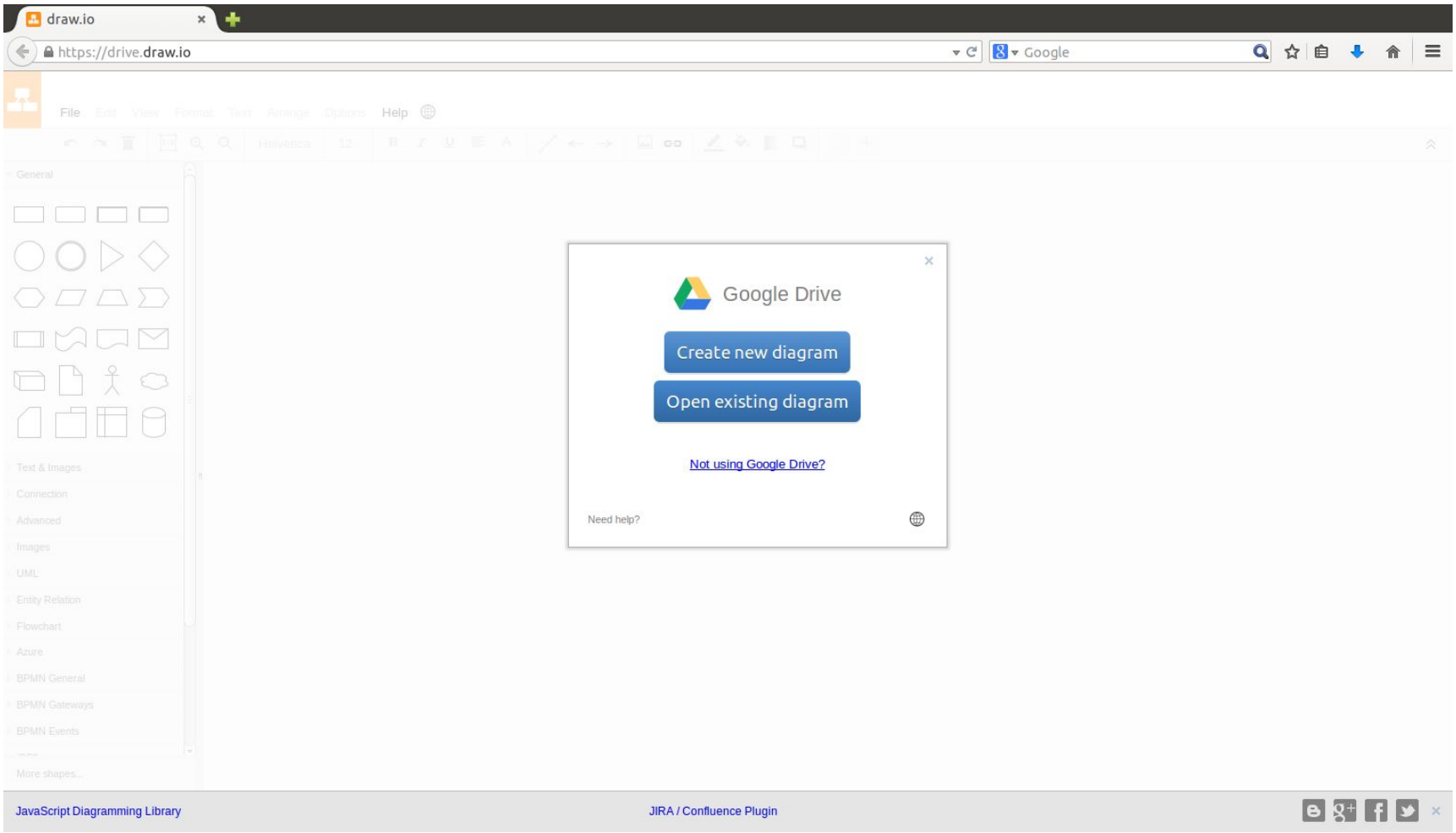
E-R Diagramming Tools - Lucid



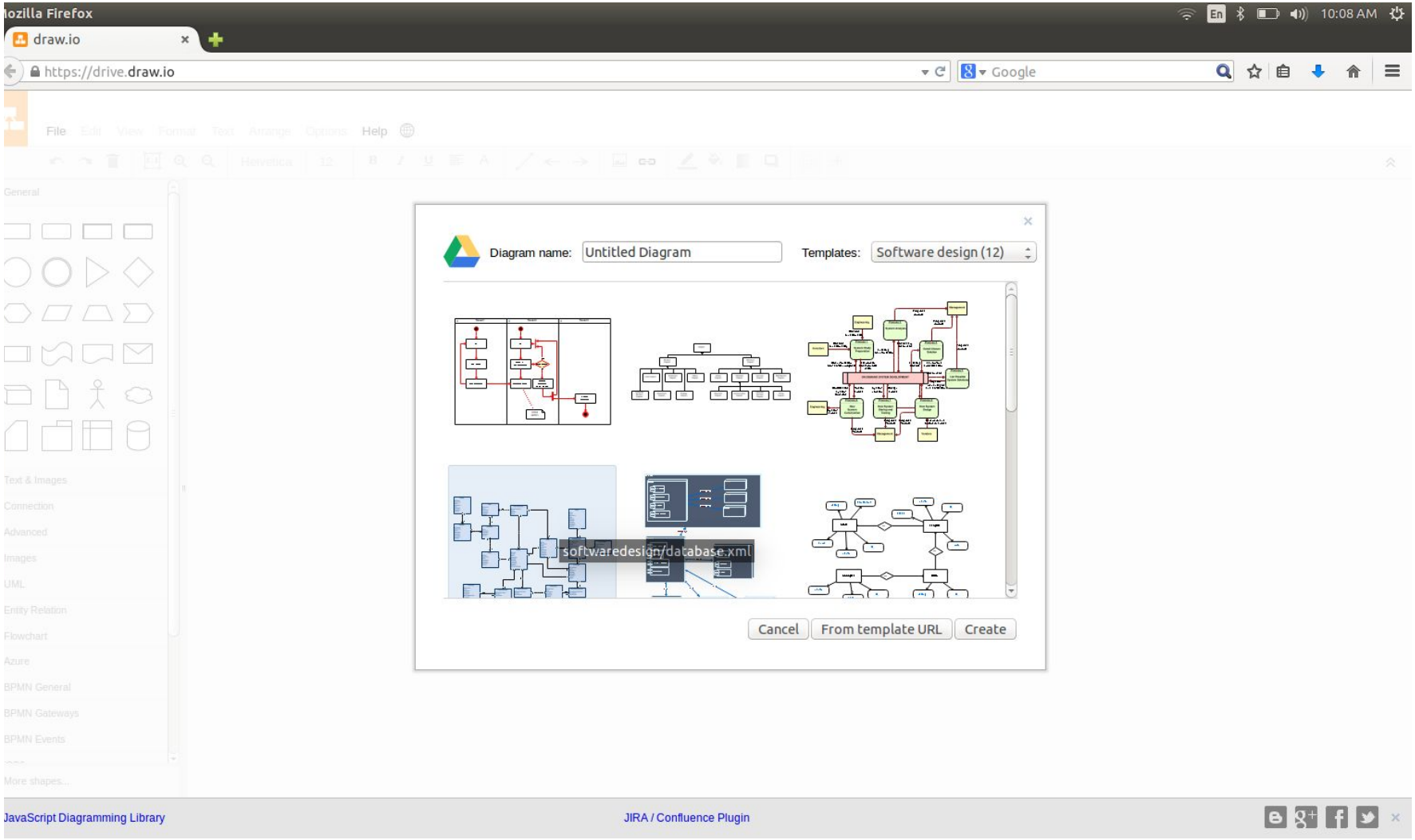
E-R Diagramming Tools - Lucid



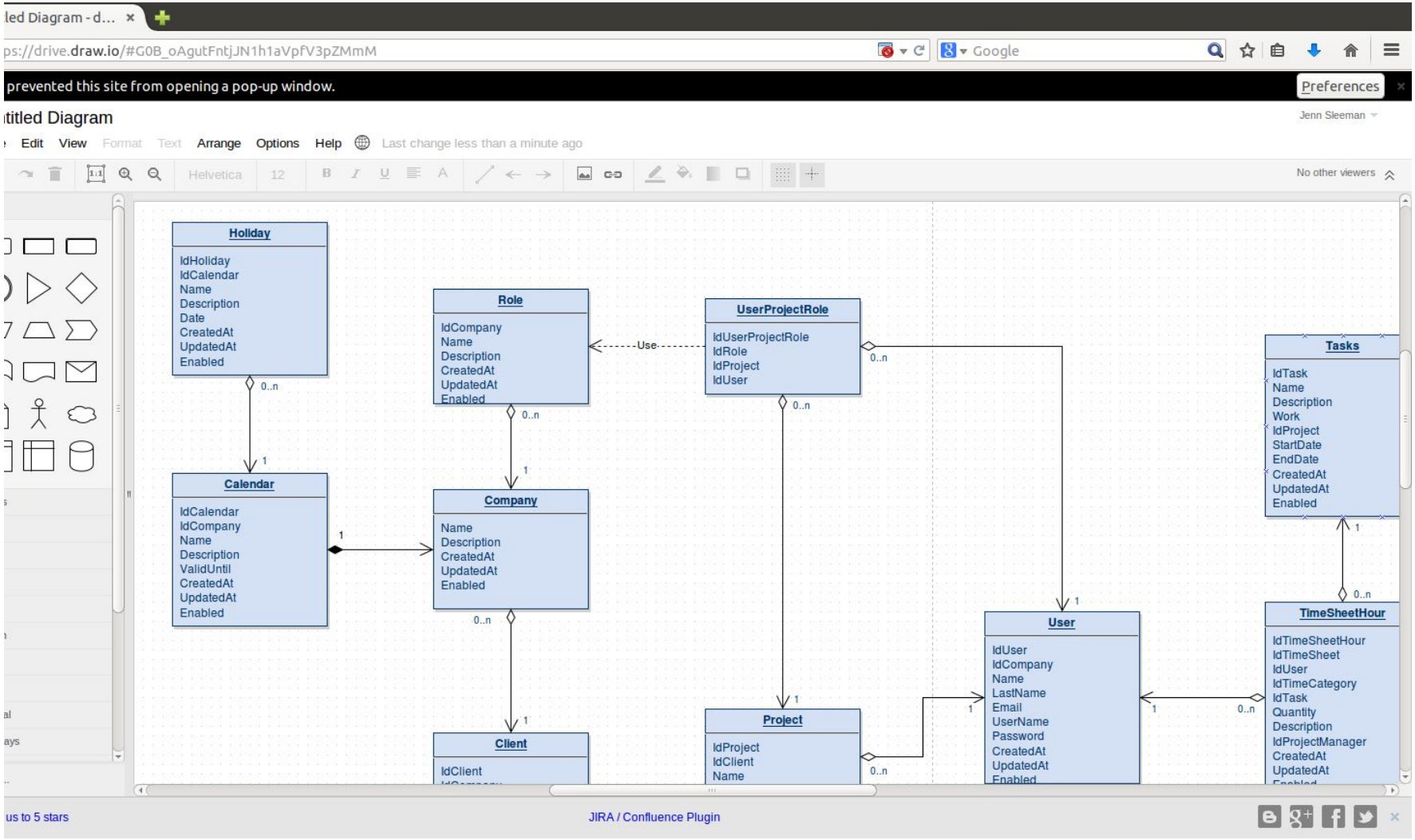
E-R Diagramming Tools – draw.io



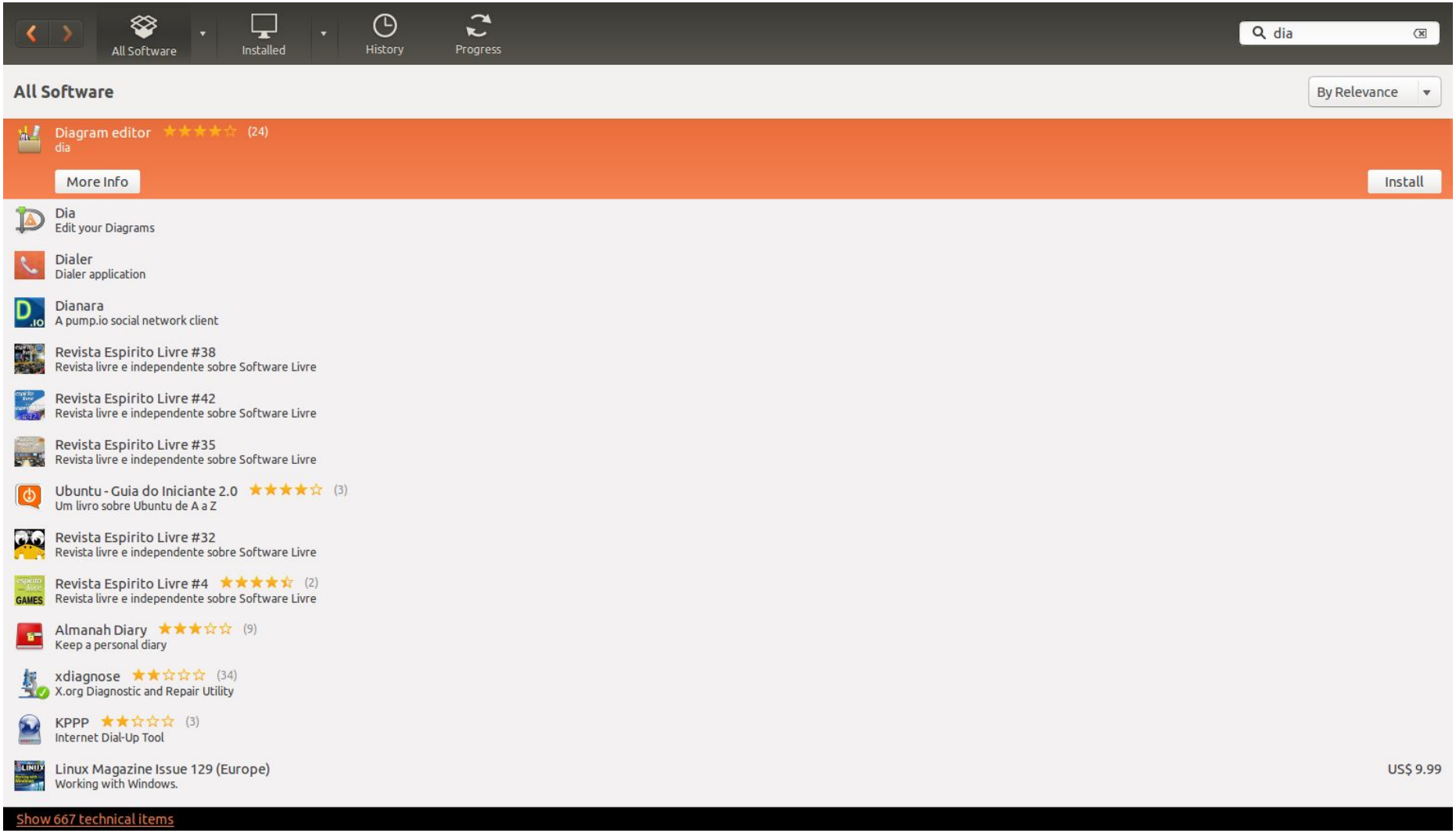
E-R Diagramming Tools – draw.io



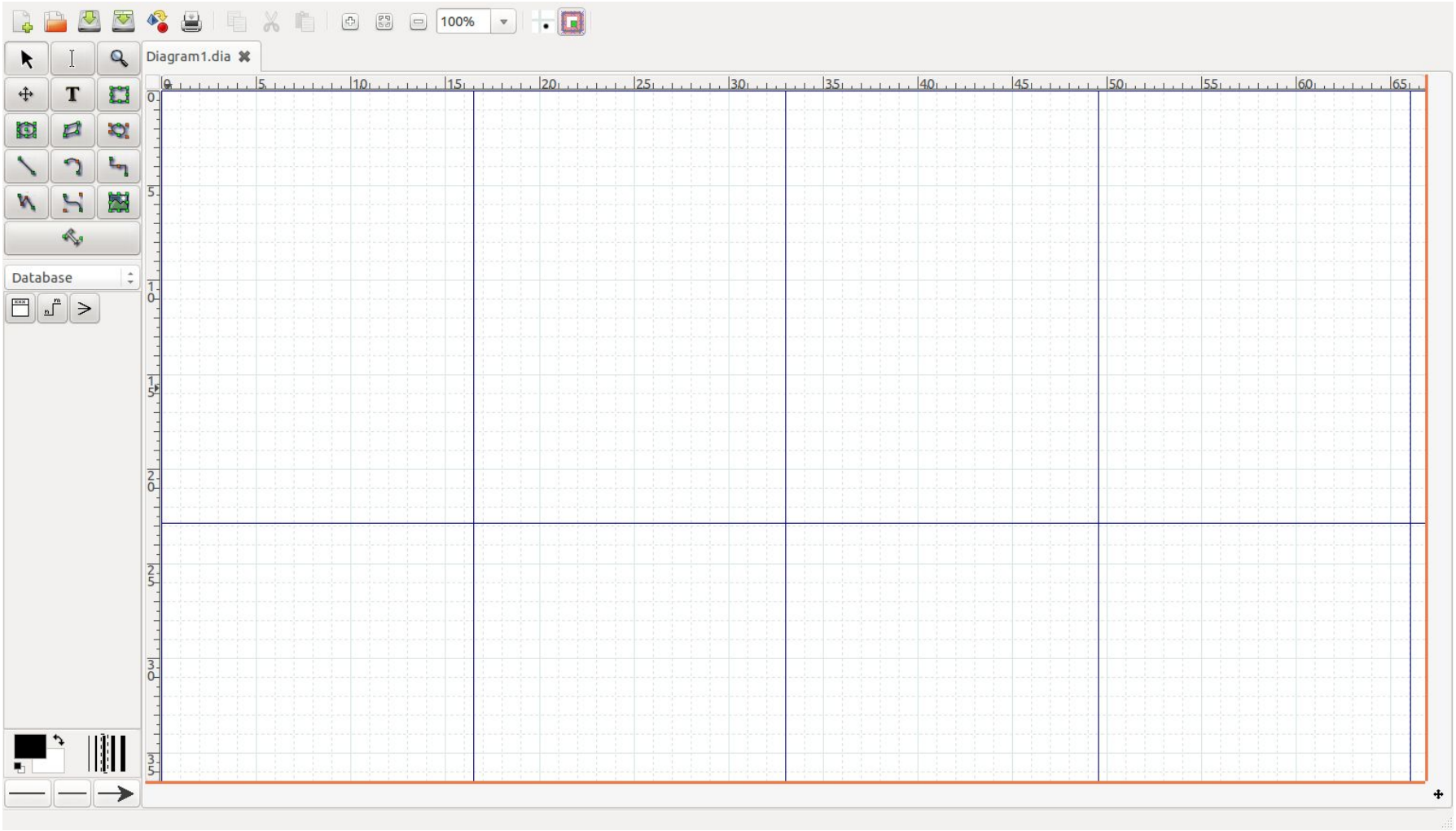
E-R Diagramming Tools – draw.io



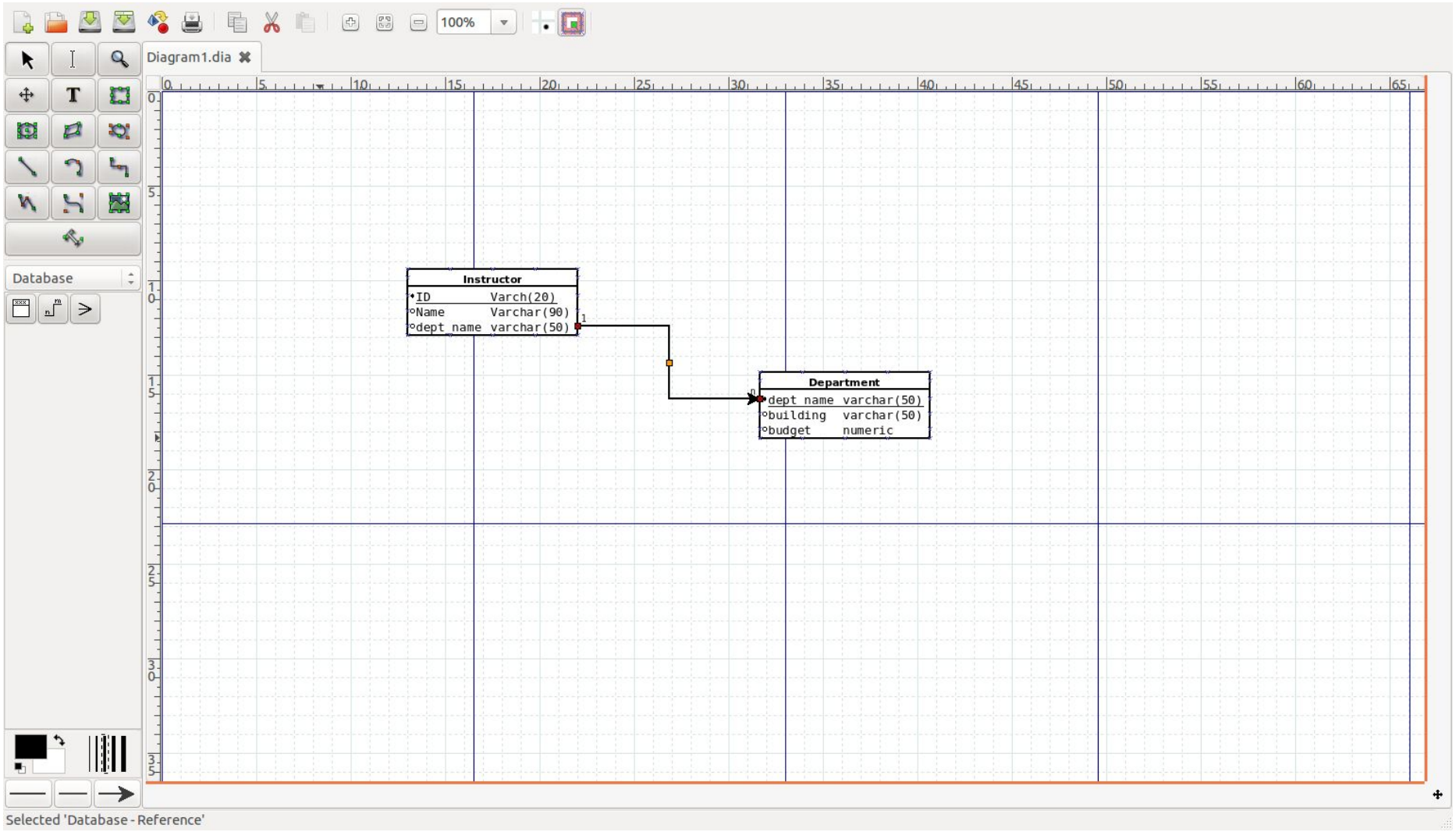
E-R Diagramming Tools – dia



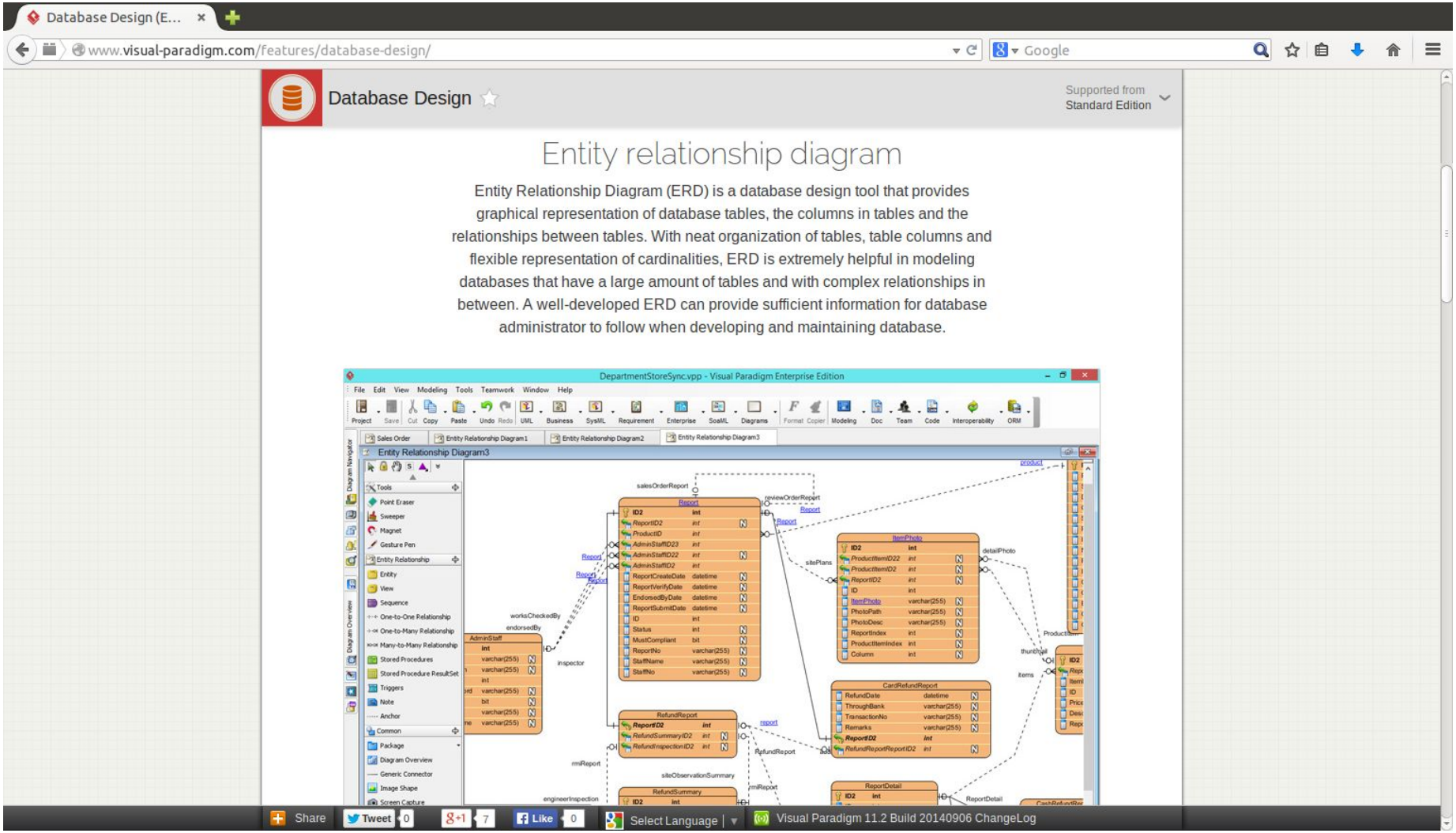
E-R Diagramming Tools – dia



E-R Diagramming Tools – dia



E-R Diagramming Tools – Visual Paradigm



Working Example - Dog Shelter

I own a dog shelter and I want to build a system that can support me in managing the dogs and the families that adopt dogs. I like to know a lot of information about the people so I can make suggestions on what type of dog would be best for the family wishing to adopt. Some of my dogs have medical conditions I need to keep track of. After an adoption I like to check in with the family at different points in time to ensure the adoption was a success.

What are the entity sets and relationship sets involved?

Working Example - Dog Shelter

What you should consider:

1. Entity Sets
2. Relationship Sets
3. Attributes (Composite, Multi-valued, Derived)
4. Weak entity sets
5. Binary and non-binary relationships
6. Cardinality
7. Relationship set roles