

CMSC 461, Database Management Systems Spring 2018

Lecture 9 - 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

Logistics

- Homework #2 due today
- Phase 2 due Monday 3/5/2018
- Homework #3 due 3/12/2018

Lecture Outline

- E-R Modeling
- Entity Sets & Relationship Sets
- Attributes
- Cardinality
- Keys
- E-R Diagram
- Design Issues
- Database Design Tools

Lecture Outline

- *E-R Modeling (review)*
- Entity Sets & Relationship Sets
- Attributes
- Cardinality
- Keys
- E-R Diagram
- Design Issues
- Database Design Tools

Design Process

- Need modeling real world applications
- Real world applications are complex
 - They are hard to model
 - Not as easy to define relations, attributes and constraints

Design Phases

- **Concept Design Phase**
 - Choose a data model
 - Translate requirements into conceptual schema (*E-R model*)

Design Alternatives

- How to represent types of “things” (*entities*)
- How to relate “things” (*relations*)
- Avoid 2 major pitfalls:
 - Redundancy
 - repeated information
 - in the schema
 - Incompleteness
 - Will make certain aspects of enterprise hard if not impossible to model

E-R Modeling

- Useful for **mapping** meanings and interactions of real-world enterprises to conceptual schema
- 3 basic concepts:
 - **Entity sets**
 - **Relationship sets**
 - **Attributes**

Lecture Outline

- E-R Modeling
- *Entity Sets & Relationship Sets (review)*
- Attributes
- Cardinality
- Keys
- E-R Diagram
- Design Issues
- Database Design Tools

Entities

- An **entity** is an object that exists and is distinguishable from other objects.
 - Example: specific person, company, event, plant

Attributes

- Entities are represented by a set of **attributes**
 - Example: people have names and addresses
- Attributes are descriptive properties
- For each attribute each entity has a **value**

Entity Sets

instructor_ID instructor_name

76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

student-ID student_name

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Relationship

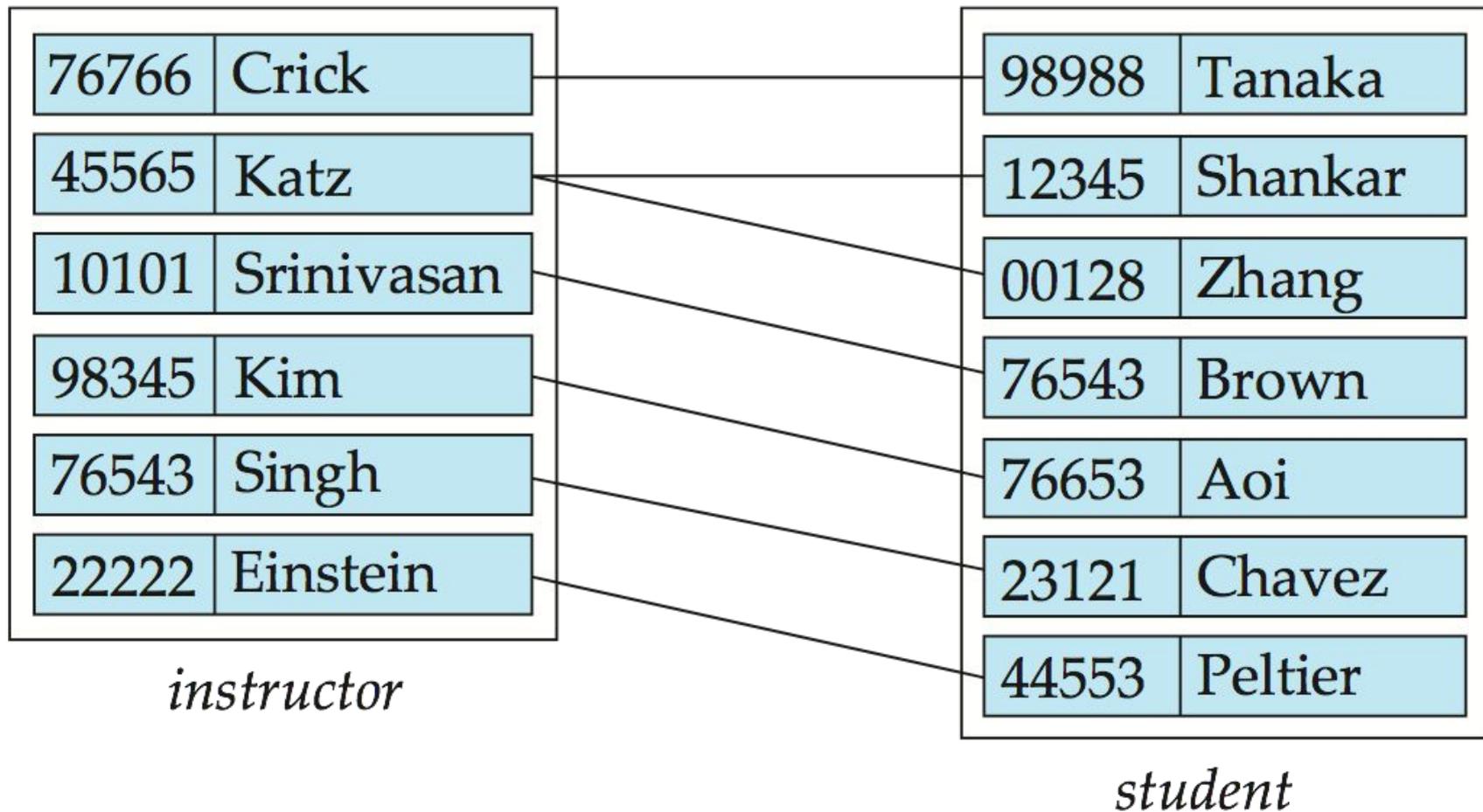
- A **relationship** is an association among several entities

Example:

Dr. Johnson *advises* *Jordan*

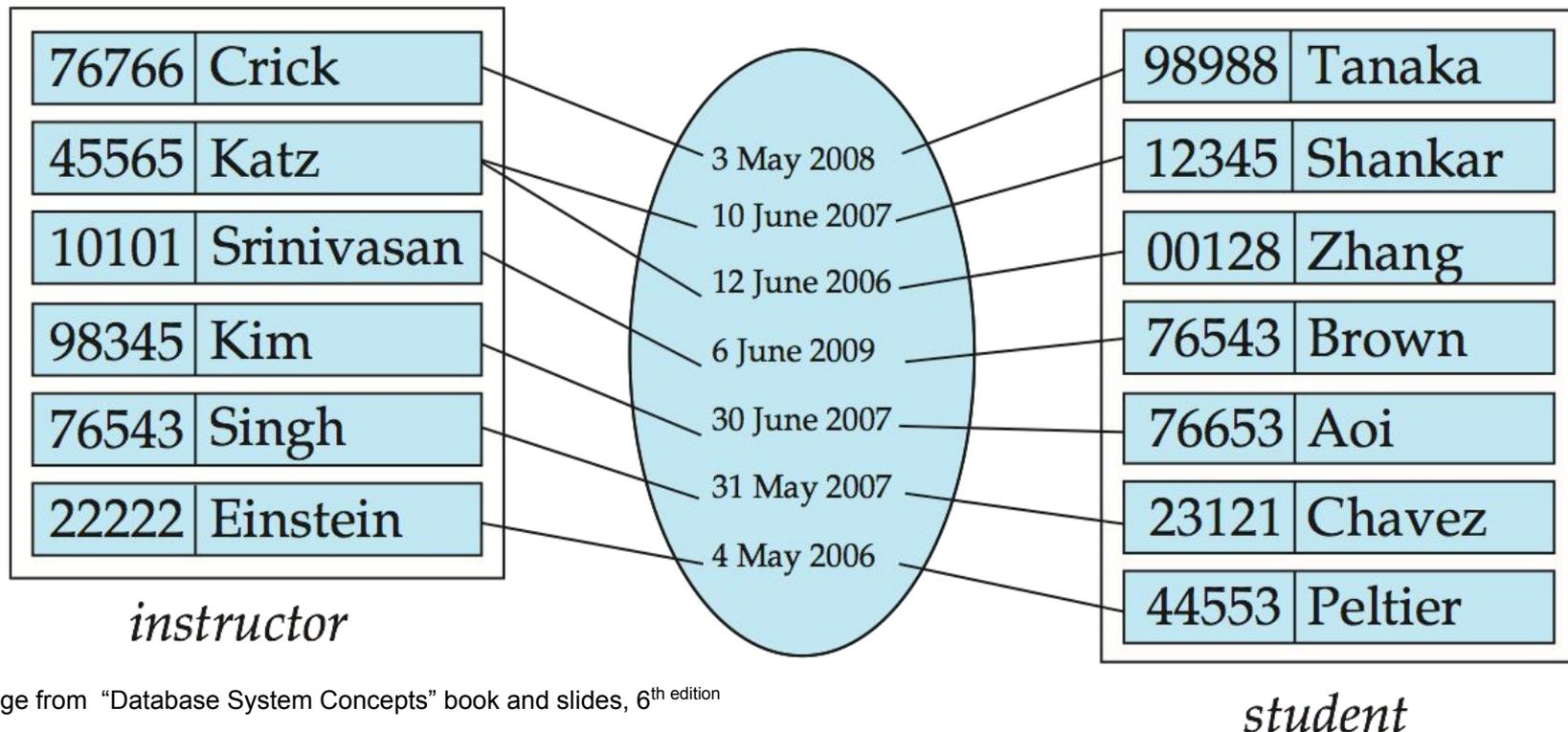
This defines an *advisor* relationship

Relationship Sets - Advisors



Relationship Sets

- An attribute can also be property of a relationship set (*descriptive attribute*)
- For instance, the advisor relationship set between entity sets instructor and student may have the attribute date which tracks when the student started being associated with the advisor



Design of a Relationship

- ***Binary relationship*** involves two entity sets (or degree two)
 - most relationship sets in a database system are binary

Lecture Outline

- E-R Modeling
- Entity Sets & Relationship Sets
- ***Attributes (review)***
- Cardinality
- Keys
- E-R Diagram
- Design Issues
- Database Design Tools

Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example:

instructor = (ID, name, street, city, salary)

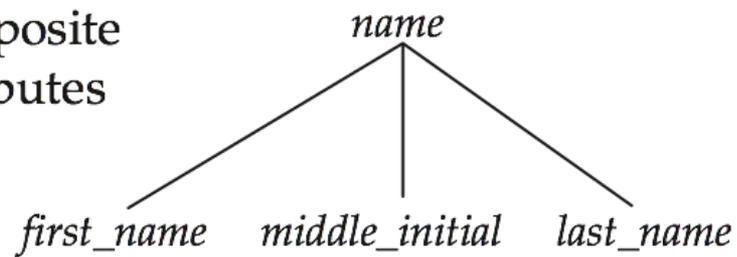
course= (course_id, title, credits)

Attributes

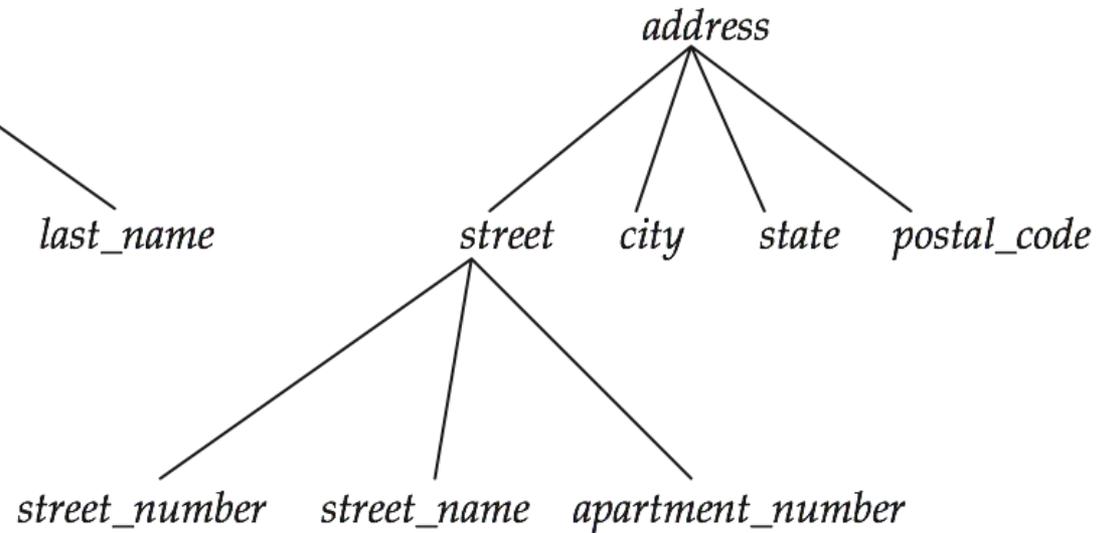
- **Domain** – the set of permitted values for each attribute
- Attribute types:
 - **Simple** and *composite* attributes
 - Example: Name (first, middle, last)
 - **Single-valued** and *multivalued* attributes
 - Example: multivalued attribute:
phone_numbers
- **Derived** attributes
 - Can be computed from other attributes
 - Example: age, given date_of_birth

Composite Attributes

composite
attributes



component
attributes



Lecture Outline

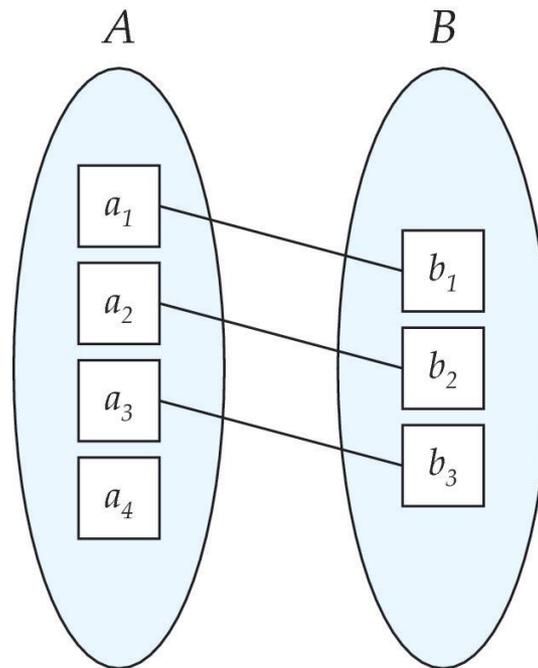
- E-R Modeling
- Entity Sets & Relationship Sets
- Attributes
- *Cardinality (review)*
- Keys
- E-R Diagram
- Design Issues
- Database Design Tools

Mapping Cardinality Constraints

- For a binary relationship set the mapping cardinality must be one of the following types:
 - One to one
 - One to many
 - Many to one
 - Many to many

Mapping Cardinality

Entity in A is associated with at most 1 entity in B and entity in B is associated with at most 1 entity in A



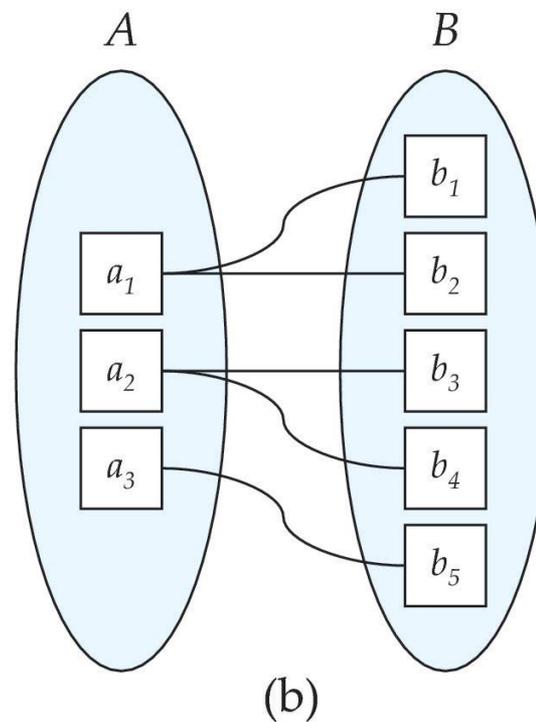
(a)

One to one

Mapping Cardinality

An entity in A is associated with any number (zero or more) of entities in B.

An entity in B is associated with AT MOST one entity in A.

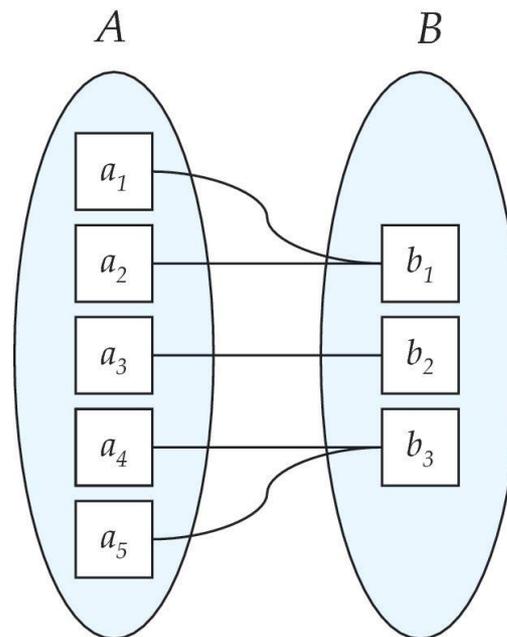


(b)
One to many

Mapping Cardinality

An entity in A is associated with AT MOST one entity in B.

An entity in B is associated with any number (zero or more) entities in A.

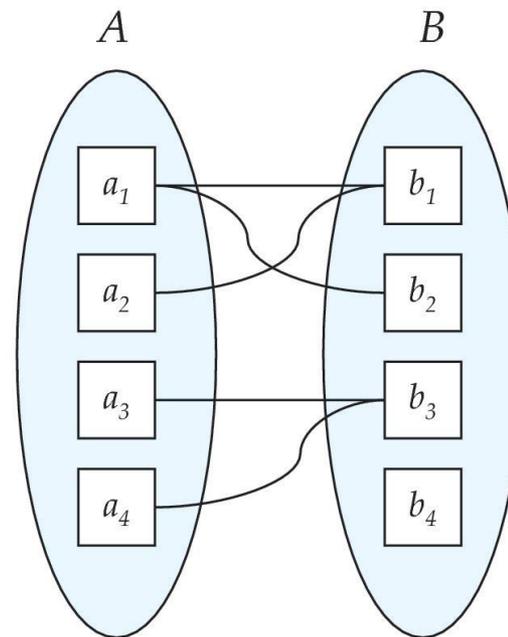


(a)

Many to one

Mapping Cardinality

An entity in A is associated with any number of entities in B , and B in A

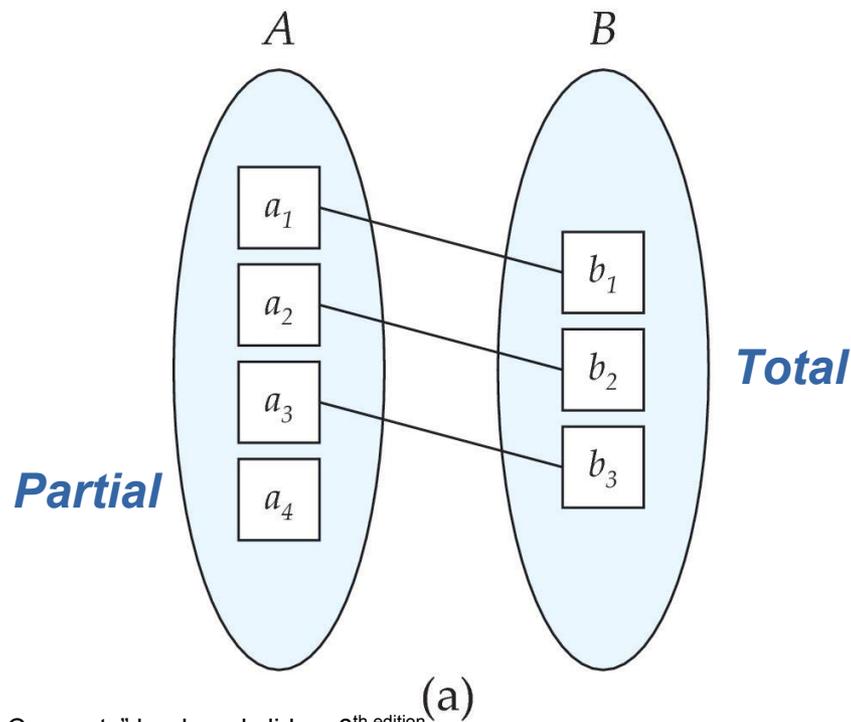


(b)

Many to many

Participation Constraints

- If every entity in entity set E participates in at least 1 relationship in the relationship set R then R is said to be **total**
- The relationship set R is said to be **partial** if only some entities in entity set E participate



Lecture Outline

- E-R Modeling
- Entity Sets & Relationship Sets
- Attributes
- Cardinality
- *Keys (review)*
- E-R Diagram
- Design Issues
- Database Design Tools

Keys

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity
- A **candidate key** of an entity set is a minimal super key
 - ID is candidate key of instructor
 - course_id is candidate key of course
- Although several candidate keys may exist, one of the candidate keys is selected to be the primary key

Recall Our Discussion of: Keys

- Let R be a set of attributes
- Let $K \subseteq R$
- K is a **superkey** of R if values for K are sufficient to identify a unique tuple of each possible relation $r(R)$
 - Example: $\{ID\}$ and $\{ID, name\}$ are both superkeys of *instructor*
- Superkey K is a **candidate key** if K is minimal
 - Example: $\{ID\}$ is a candidate key for *Instructor*

Super Keys & Candidate Keys

What is/are the super key(s)?

What is/are the candidate key(s)?

<i>ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

Teaches
Table

Keys

- A **super key** of an entity set is a set of one or more attributes whose values uniquely determine each entity
- A **candidate key** of an entity set is a minimal super key
 - ID is candidate key of instructor
 - course_id is candidate key of course
- Although several candidate keys may exist, one of the candidate keys is selected to be the primary key

Keys For Relationship Sets

- Keys can be used to uniquely identify relationships
- The combination of primary keys of the participating entity sets forms a **super key** of a relationship set.
 - (s_id, i_id) is the super key of advisor
 - NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.

Keys For Relationship Sets

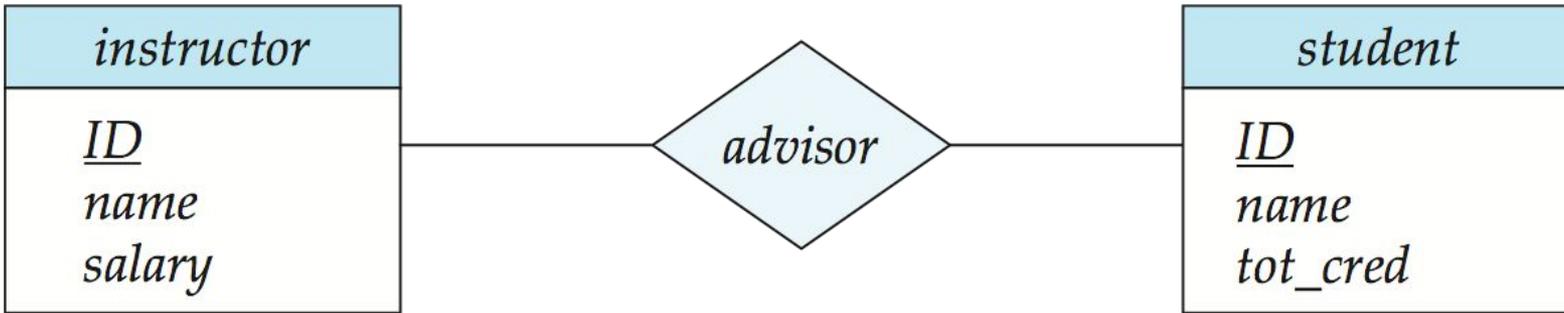
- Must consider the mapping cardinality of the relationship set when deciding what are the candidate keys
- Need to consider semantics of relationship set in selecting the primary key in case of more than one candidate key

Redundant Attributes

- Start design by identifying entity sets
- Choose identifying attributes
- Then choose relationship sets among entities
- Could result in redundant attributes across entity sets
- Try to remove
 - Only during the ER modeling phase

Overview E-R Diagram

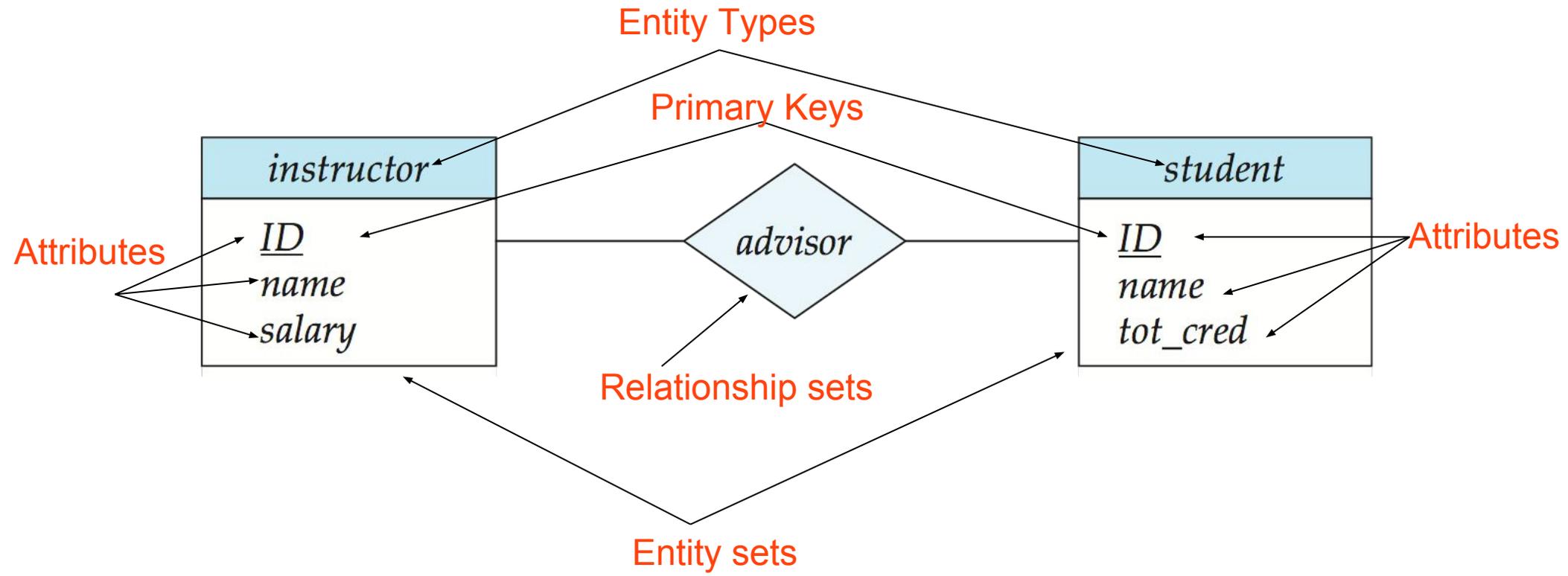
- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Attributes listed inside entity rectangle
- Underline indicates primary key attributes



Lecture Outline

- E-R Modeling
- Entity Sets & Relationship Sets
- Attributes
- Cardinality
- Keys
- *E-R Diagram*
- Design Issues
- Database Design Tools

E-R Diagram



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?

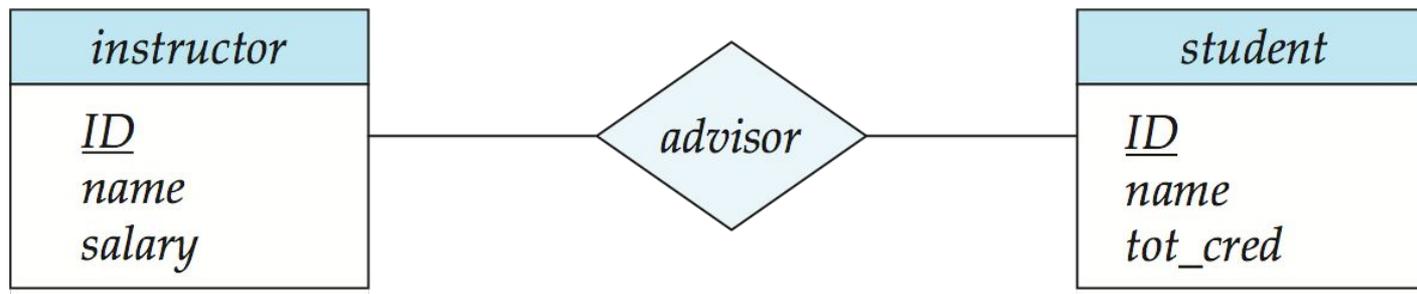
Working Example - Dog Shelter

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?

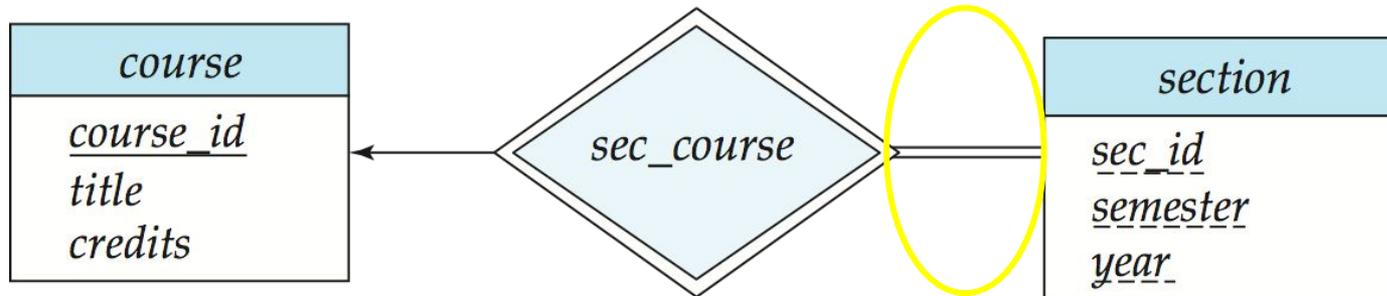
Partial Participation of an Entity Set in a Relationship Set

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

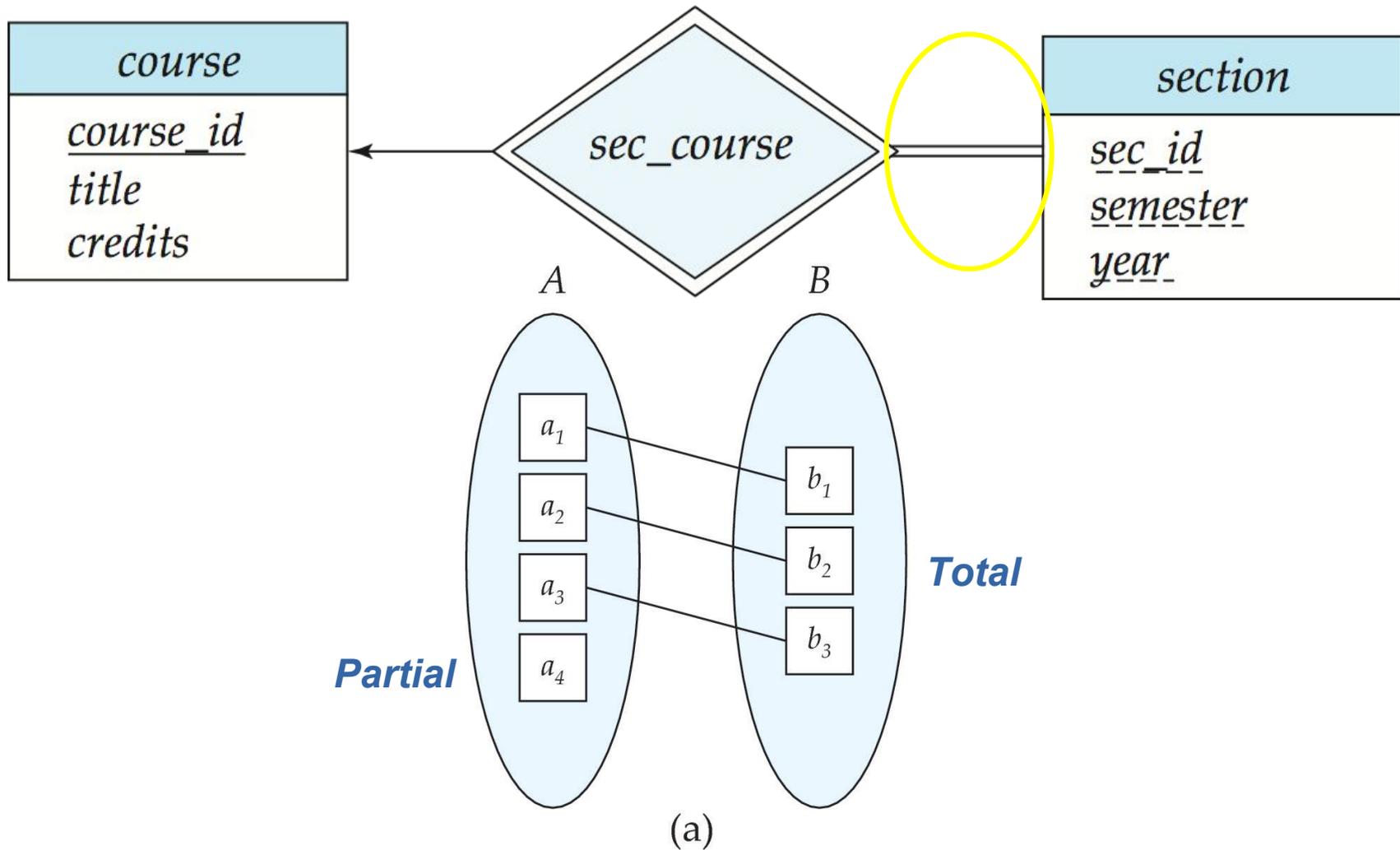


Total Participation of an Entity Set in a Relationship Set

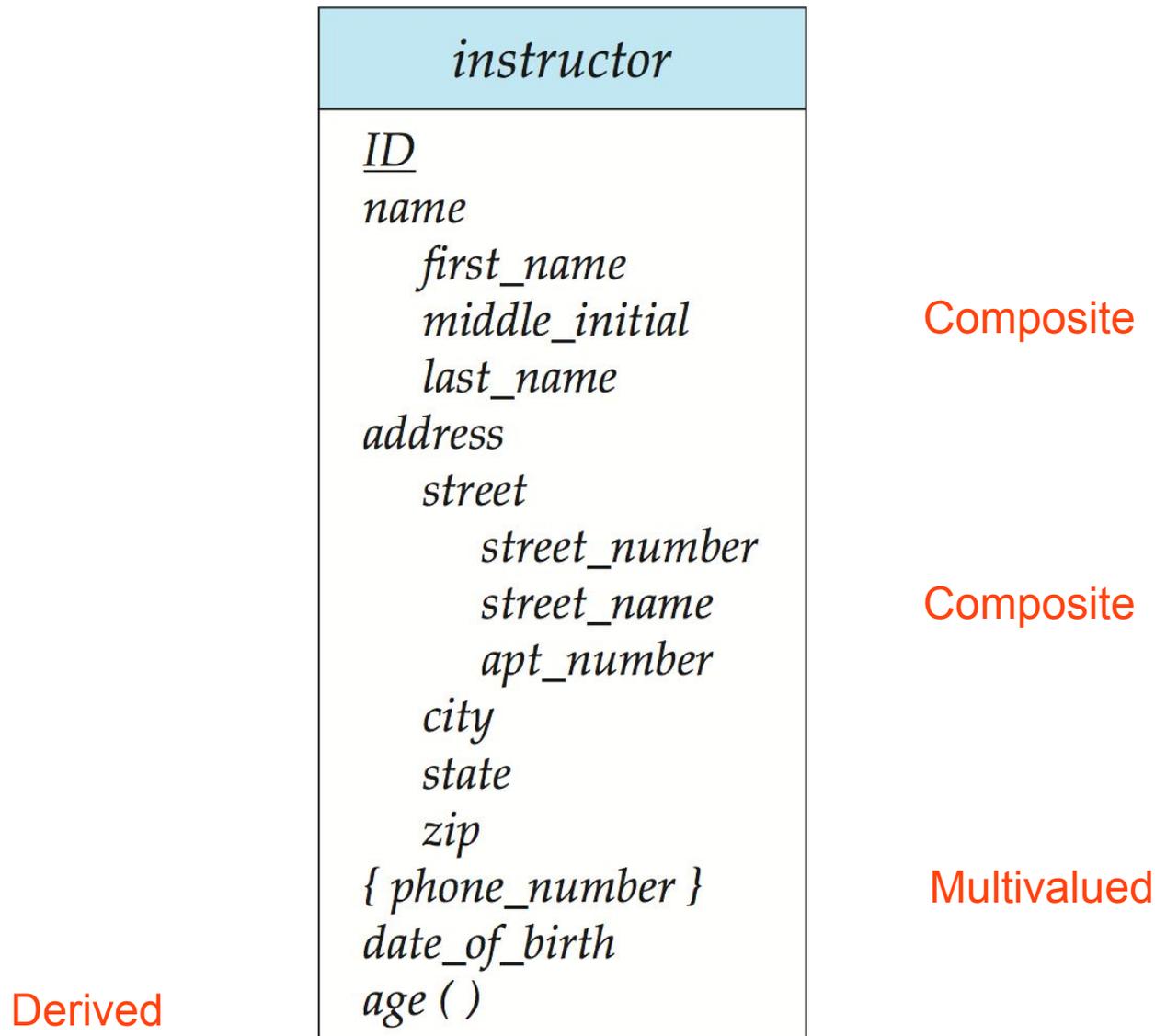
- Use double lines
- Every entity in entity set participates in at least one relationship in the relationship set
 - Example: participation of *section* in *sec_course* is total



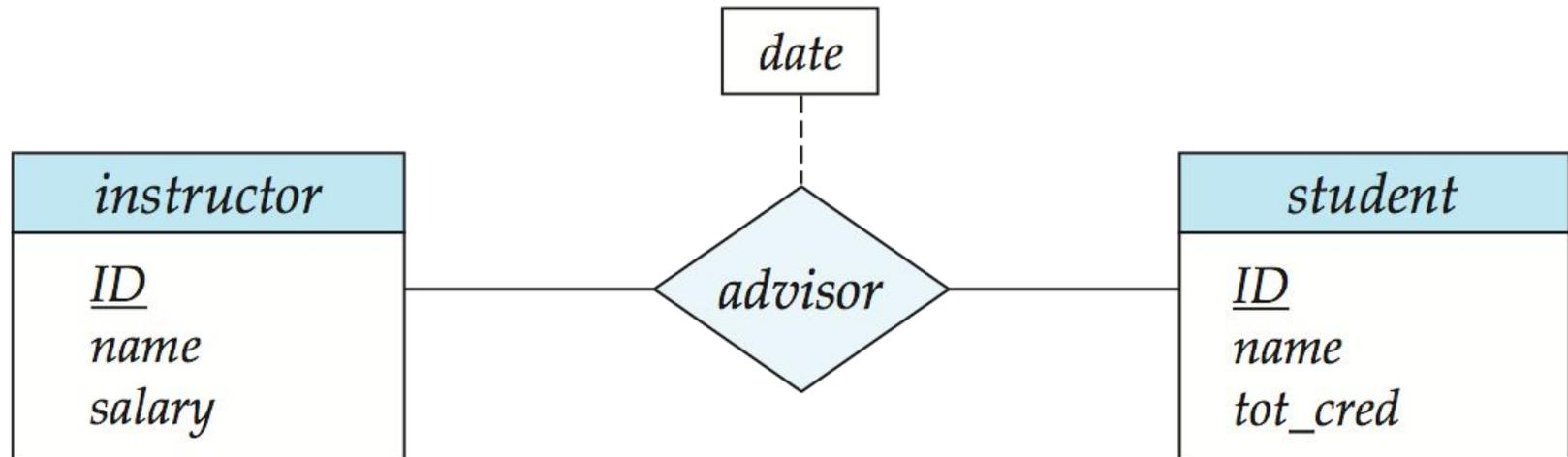
E-R Diagram



Entity With Composite, Multivalued, and Derived Attributes



Relationship Sets with Attributes

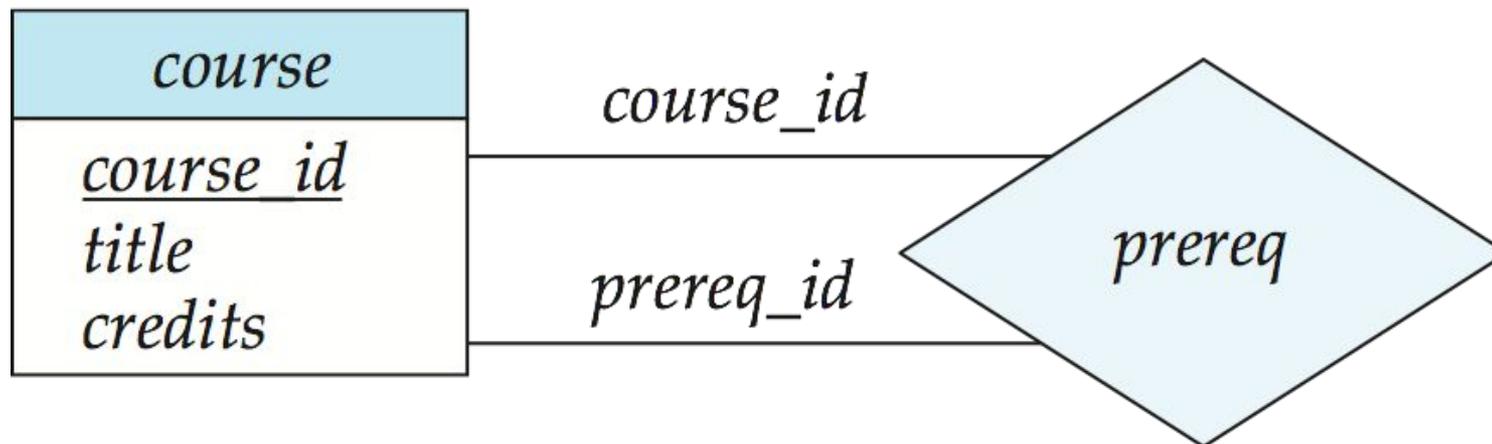


Why have attributes on the relationship set?

What would be the likely primary key of *advisor*?

Roles

- Relationships not required to be distinct
 - Each occurrence of entity set plays a “role” in relationship
- “course_id” and “prereq_id” are called roles
- Label the lines that connect diamonds and rectangles



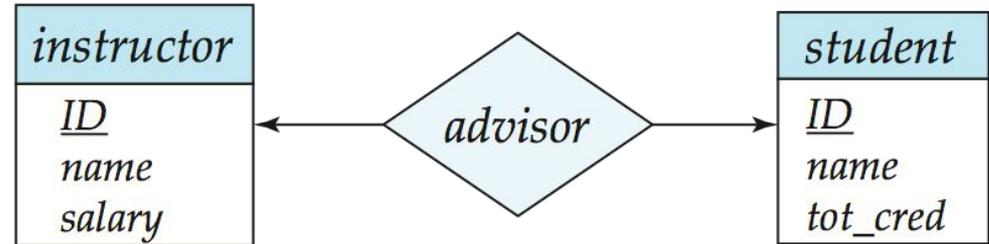
Working Example - Dog Shelter

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.

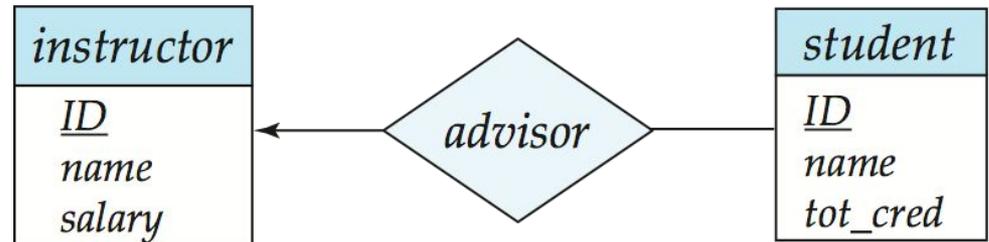
What are some of the attributes we might have for the entity sets and relationship sets, any roles?

Cardinality Constraints

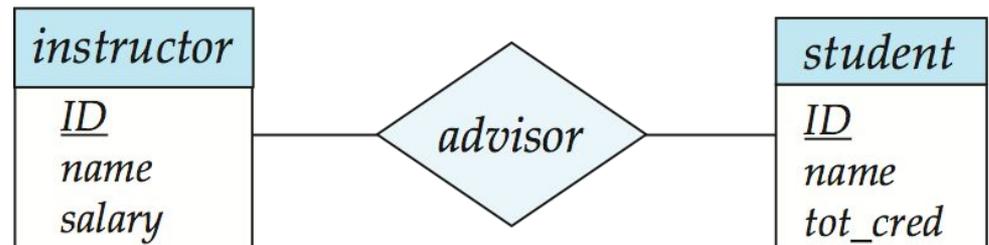
- draw directed line (\rightarrow)
 - signifying “one”
- undirected line (—),
 - signifying “many”
- Between the relationship set and the entity set



(a)



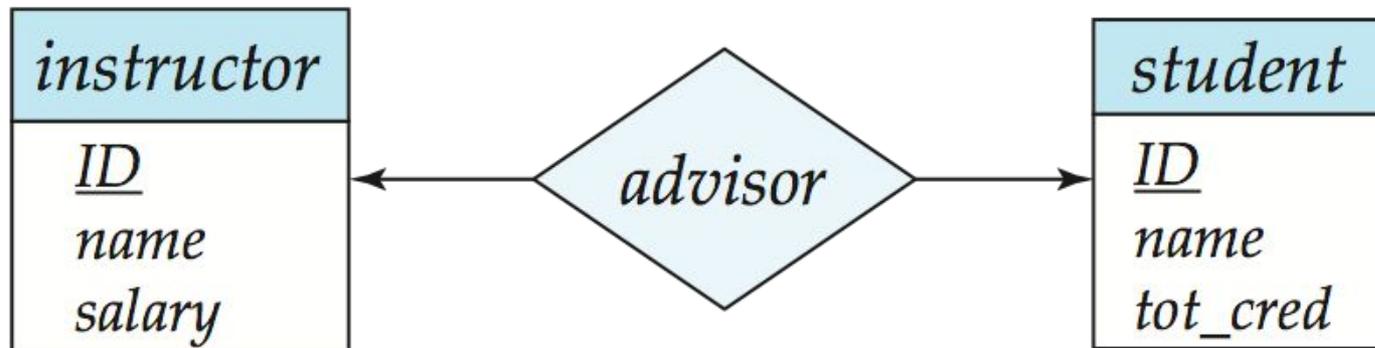
(b)



(c)

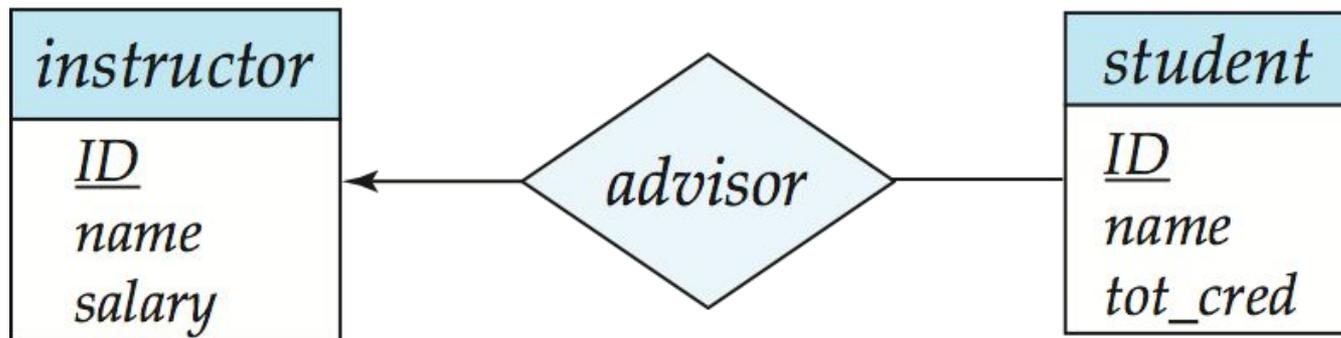
One-to-One Relationship

- Between an instructor and a student
 - an instructor is associated with at most one student via advisor
 - and a student is associated with at most one instructor via advisor



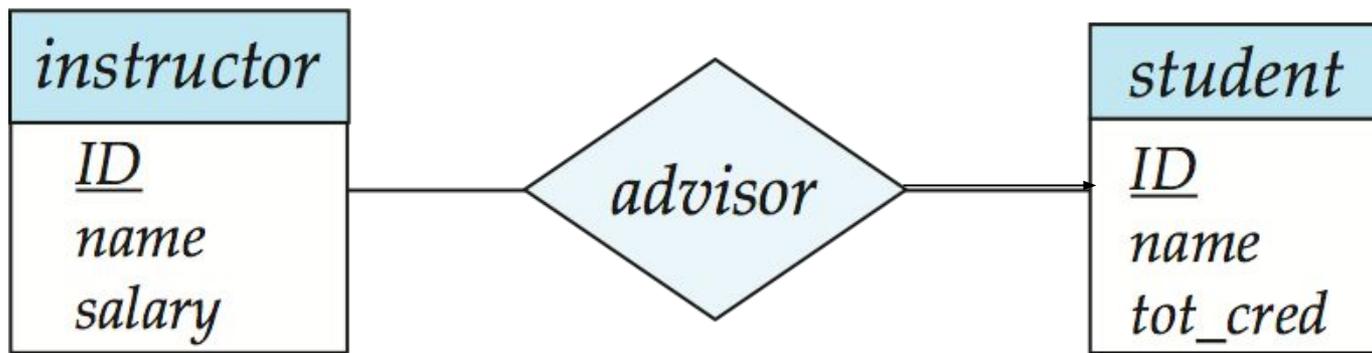
One-to-Many Relationship

- one-to-many relationship between instructor and student
 - an instructor is associated with 0 to many students via advisor
 - a student is associated with at most one instructor via advisor



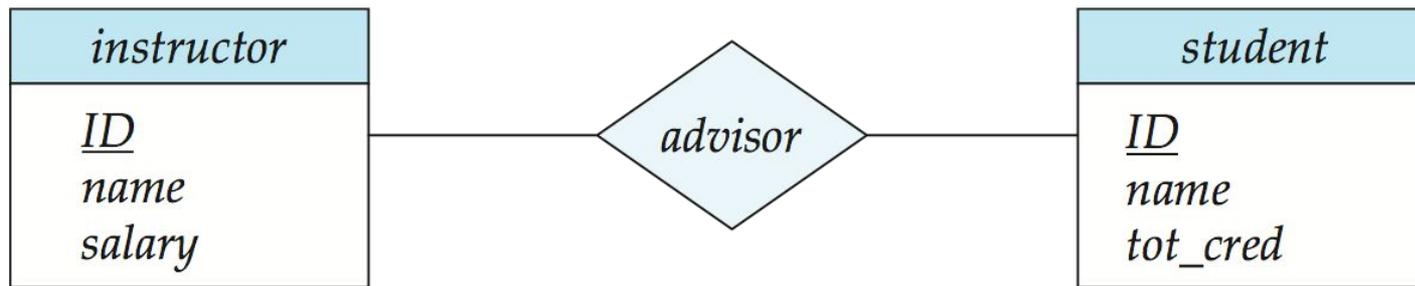
Many-to-One Relationship

- many-to-one relationship between instructor and student,
 - an instructor is associated with at most one student via advisor,
 - a student is associated with 0 to many instructors via advisor



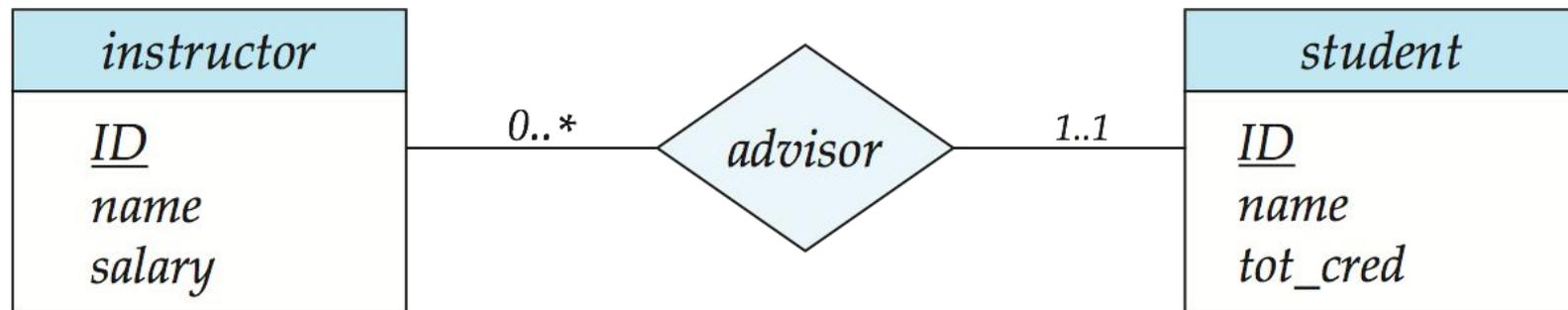
Many-to-Many Relationship

- An instructor is associated with 0 to many students via advisor
- A student is associated with 0 to many instructors via advisor

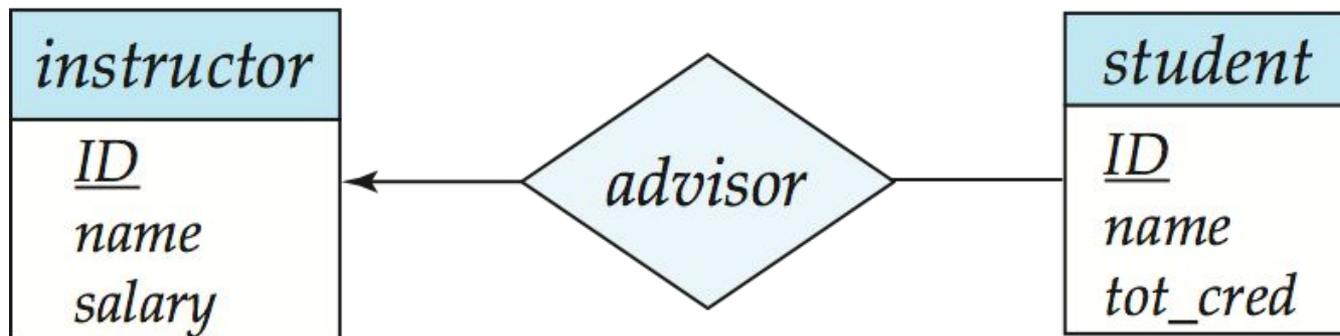


Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints

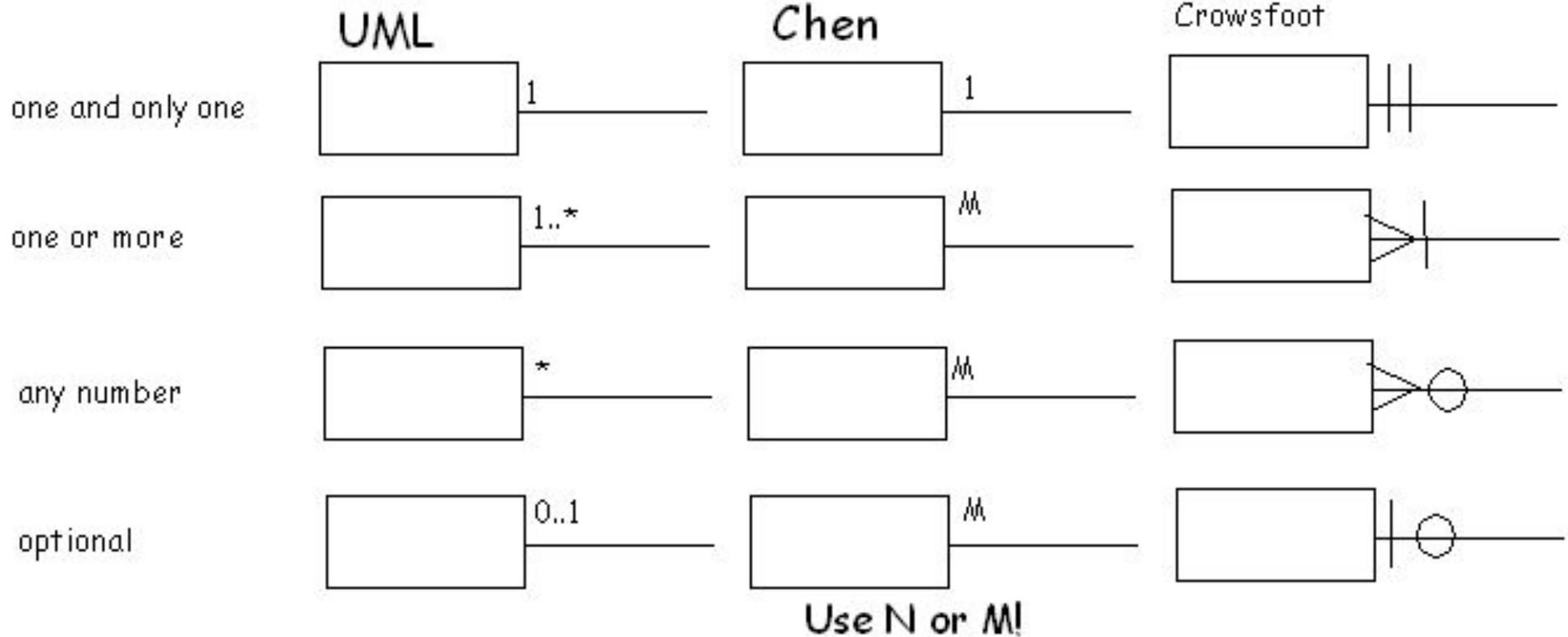


Each student has exactly one advisor
An instructor can advise 0 to many students



Alternative Notation for Cardinality Limits

Notations for showing cardinality (multiplicity)



In-Class Exercise

- Is it one-to-one, one-to-many, many-to-one, or many-to-many?
- Owner and Car
- Person and Job
- Person and Weight
- Person and Mother (Birth)
- Husband and Wife

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 is the cardinality of the relationships?

Weak Entity Sets

- **Weak entity set** - An entity set that does not have a primary key
- **Strong entity set** - An entity set that has a primary key
- **Identifying entity set** – entity set that is associated with a weak entity set
 - Weak entity set is ***existence dependent*** on identifying entity set
 - Identifying entity set “owns” weak entity set

Weak Entity Sets

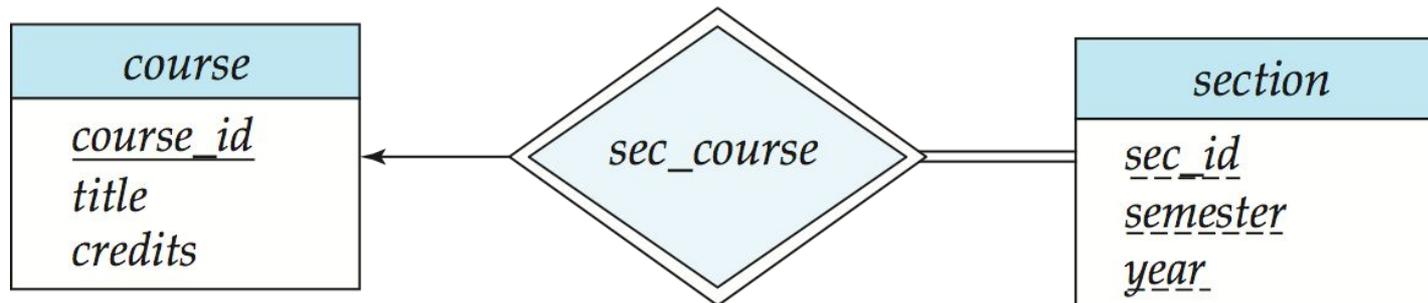
- **Identifying relationship** - Relationship associating weak entity set with identifying entity set
 - Many-to-one from weak entity set to identifying entity set
 - Participation of weak entity set in relationship set is total
 - No descriptive attributes
 - depicted using a double diamond

Weak Entity Sets

- **Discriminator** - *set of attributes that allow distinction among entities in weak entity set*
 - *Also called partial key*
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator

Weak Entity Sets

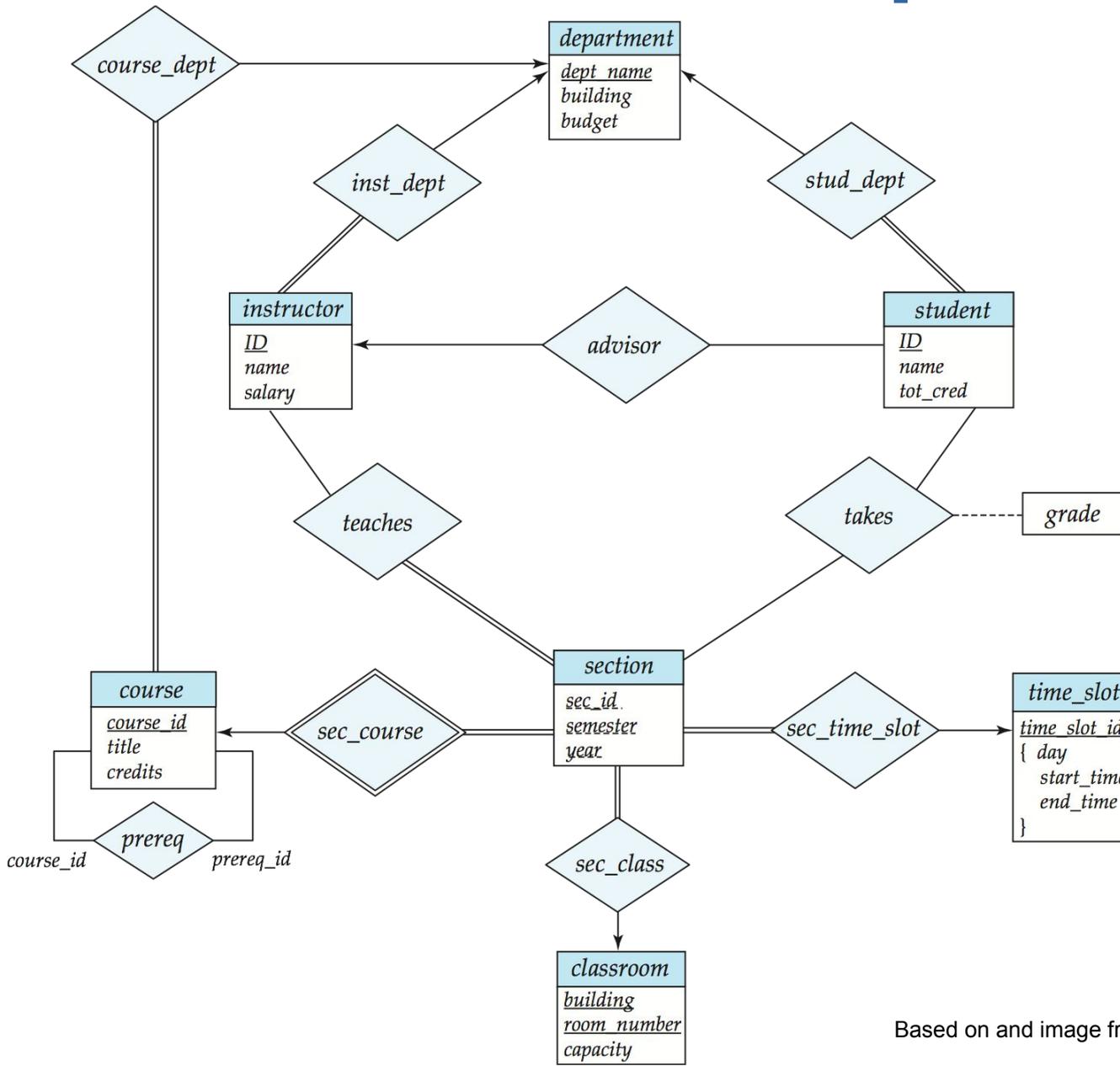
- We underline the discriminator of a weak entity set with a dashed line
- We put the identifying relationship of a weak entity in a double diamond
- Primary key for *section* – (*course_id*, *sec_id*, *semester*, *year*)



Weak Entity Sets

- Note: the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship
- If *course_id* were explicitly stored, *section* could be made a strong entity, but then the relationship between *section* and *course* would be duplicated by an implicit relationship defined by the attribute *course_id* common to *course* and *section*

E-R Diagram for University Enterprise



- Name an example of a weak entity set in the university schema?

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.

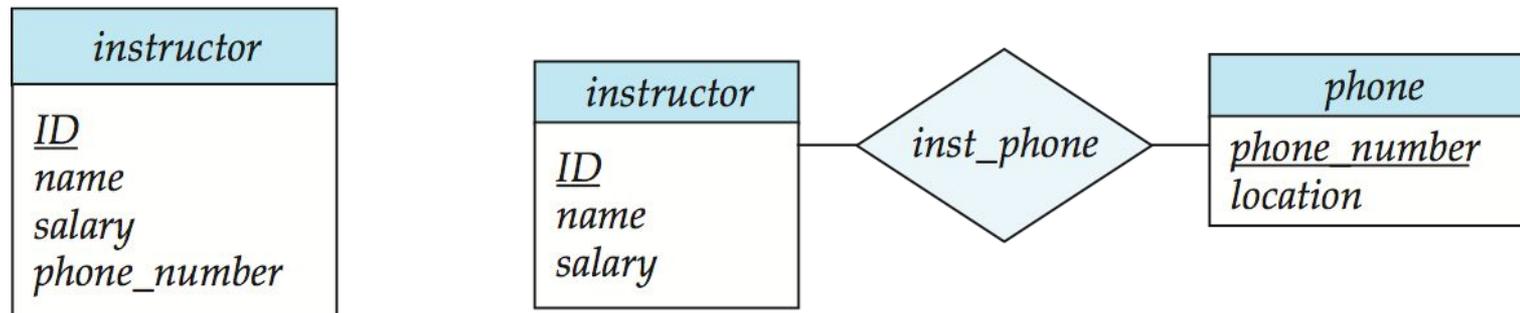
Are there any weak entity sets?

Lecture Outline

- E-R Modeling
- Entity Sets & Relationship Sets
- Attributes
- Cardinality
- Keys
- E-R Diagram
- ***Design Issues***
- Database Design Tools

Design Issues – Entity Sets vs. Attributes

- Use of phone as an entity allows extra information about phone numbers (plus multiple phone numbers)

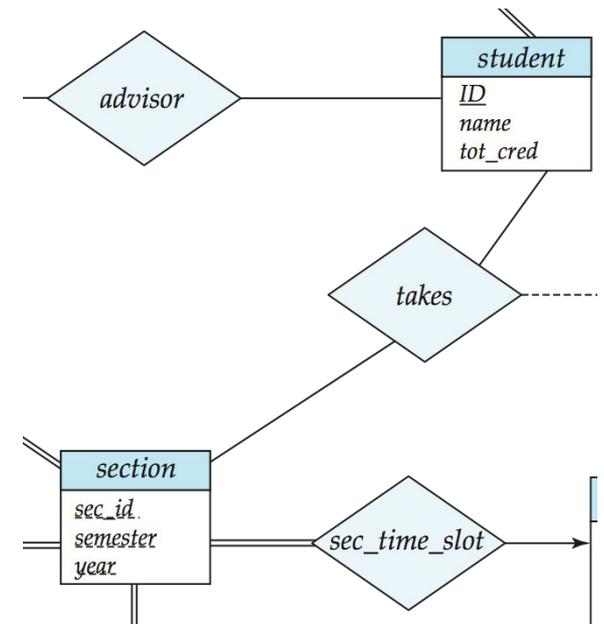
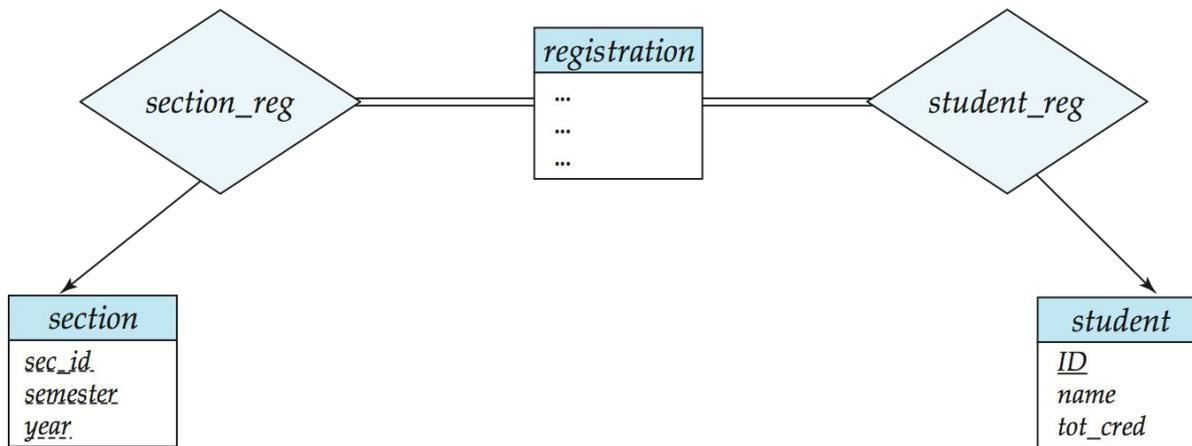


Design Issues – Entity Sets vs. Attributes

- Key Mistakes:
 - Using primary key of an entity set as an attribute of another entity set
 - * Student ID in Instructor relation
 - Designation of primary keys as attributes of the relation 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

- E-R Modeling
- Entity Sets & Relationship Sets
- Attributes
- Cardinality
- Keys
- E-R Diagram
- Design Issues
- ***Database Design Tools***

E-R Diagramming Tools - Lucid

The screenshot displays the Lucidchart web application interface for creating Entity Relationship (E-R) diagrams. The browser address bar shows the URL: <https://www.lucidchart.com/documents/edit/fbba4b55-edaa-4912-b238-6e2f37b6e38b/0#?demo=on>. The main workspace is titled "Blank Flowchart" and includes a menu bar with options like File, Edit, View, Page, Arrange, and Insert. A "Sign Up Free" button is visible in the top right corner.

The "MANAGE LIBRARY" panel is open, showing various shape categories. The "Entity Relationship" category is selected, and a preview of an E-R diagram is displayed. The diagram includes the following entities and relationships:

- Player** (Entity): Attributes include `id`, `firstName`, `lastName`, `height`, `weight`, `position`, `number`.
- HockeyGame** (Entity): Attributes include `id`, `homeTeam`, `awayTeam`, `score`, `date`, `time`.
- GameScore** (Entity): Attributes include `id`, `gameId`, `score`.
- HockeyTeam** (Entity): Attributes include `id`, `name`, `logo`.
- HockeyTeamPlayer** (Entity): Attributes include `id`, `teamId`, `playerId`, `position`.
- User** (Entity): Attributes include `id`, `username`, `password`.
- UserScore** (Entity): Attributes include `id`, `userId`, `score`.
- UserScorePoints** (Entity): Attributes include `id`, `userId`, `score`, `points`.

The relationships shown in the diagram are:

- Player** is associated with **HockeyGame** (1-to-many relationship).
- HockeyGame** is associated with **GameScore** (1-to-many relationship).
- HockeyTeam** is associated with **HockeyTeamPlayer** (1-to-many relationship).
- User** is associated with **UserScore** (1-to-many relationship).
- User** is associated with **UserScorePoints** (1-to-many relationship).

The "GRAPHIC" panel on the right shows styling options for the diagram elements, including Selection, Theme, Colors, Line, Corner Radius, Fill, and Opacity.

E-R Diagramming Tools - Lucid

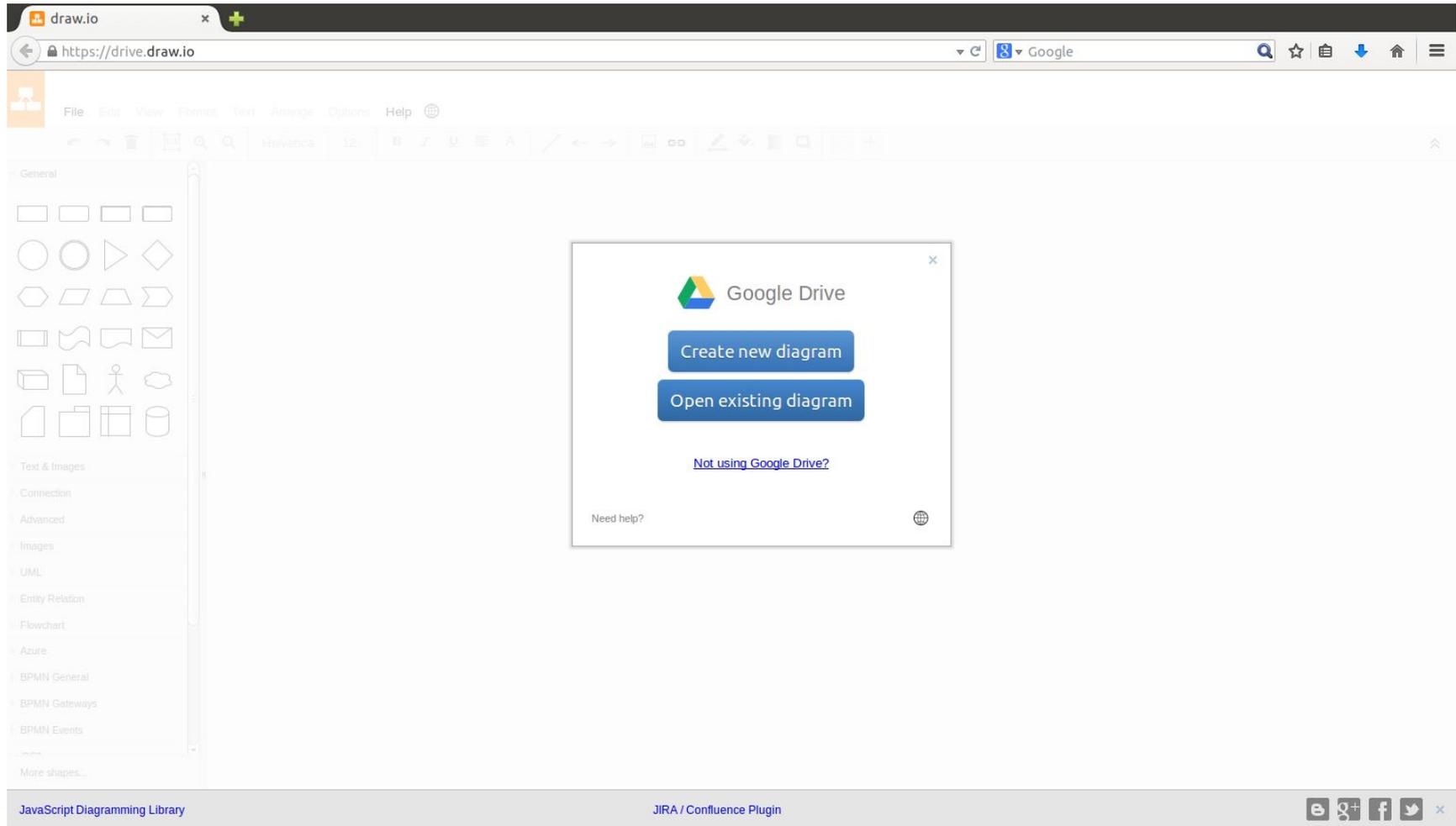
The screenshot displays the Lucidchart web application interface for creating an Entity-Relationship (E-R) diagram. The browser address bar shows the URL: <https://www.lucidchart.com/documents/edit/fbba4b55-edaa-4912-b238-6e2f37b6e38b/0#?demo=on>. The page title is "Blank Flowchart".

The interface includes a menu bar with options: File, Edit, View, Page, Arrange, Insert, Share, Window, and Help. A "Sign up free to save this document" banner is visible. The main workspace is a grid where an E-R diagram is being constructed. The diagram shows an entity named "Entity" with three fields and two types. A tooltip above the diagram reads: "To draw a line, click and drag from any edge of the shape." A "Manage Fields" dialog box is open, showing options for "Shaded Header" (On/Off), "Alternate Row Color" (On/Off), and "Fields" (set to 3).

The left sidebar contains a search bar and several categories of shapes: STANDARD, FLOWCHART, CONTAINERS, SHAPES, and ENTITY RELATION... The right sidebar shows a "GRAPHIC" panel with settings for Line (2 px), Corner Radius (10 px), Fill (Solid Color), Shadow, and Opacity (100%).

At the bottom, there are buttons for "More Shapes", "Chat", and "Comments" (ON).

E-R Diagramming Tools – draw.io



E-R Diagramming Tools – draw.io

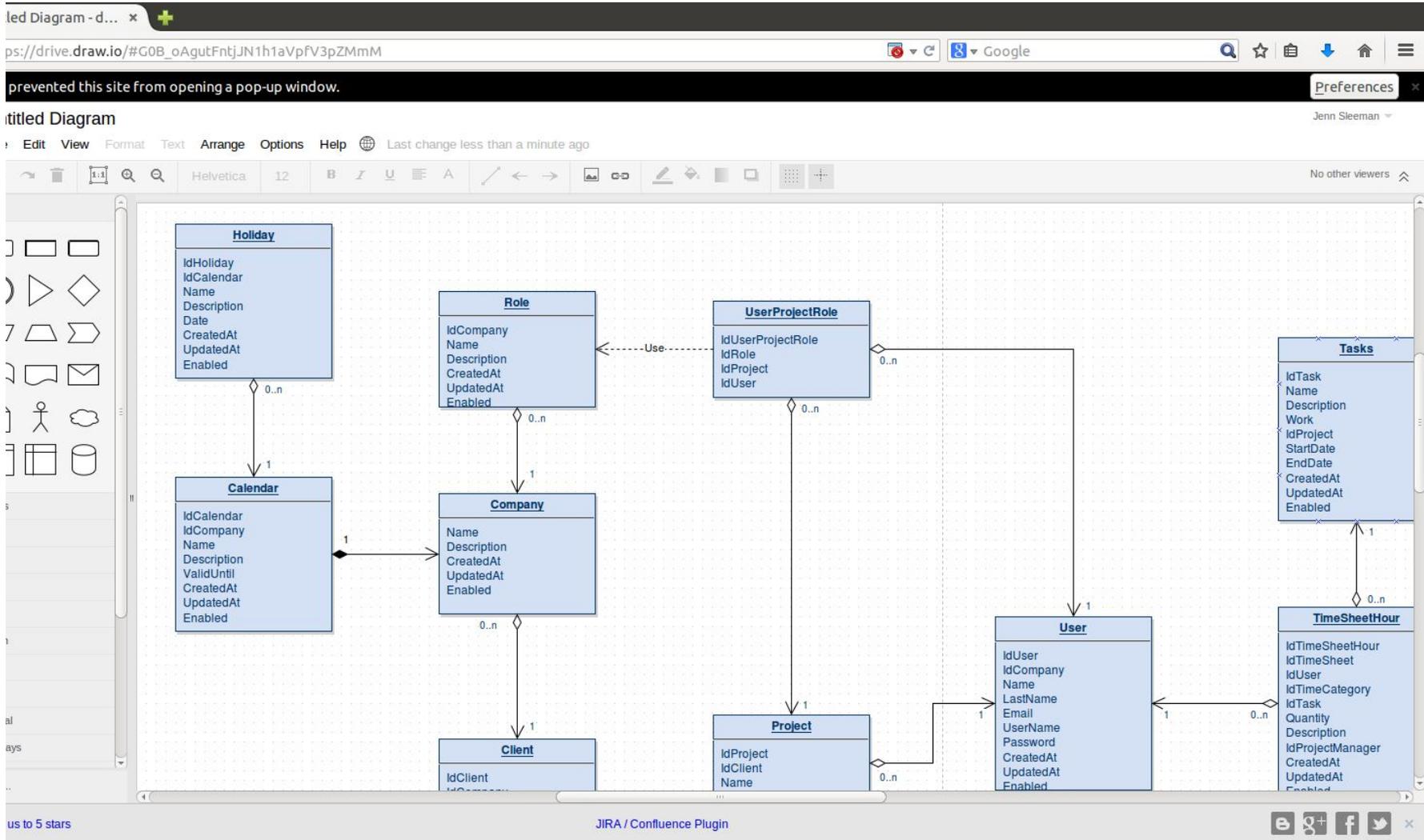
The screenshot shows the draw.io web application running in a Mozilla Firefox browser. The browser's address bar displays the URL <https://drive.draw.io>. The application interface includes a menu bar (File, Edit, View, Format, Text, Arrange, Options, Help) and a toolbar with various drawing tools. On the left side, there is a sidebar with categories of shapes: General, Text & Images, Connection, Advanced, Images, UML, Entity Relation, Flowchart, Azure, BPMN General, BPMN Gateways, BPMN Events, and More shapes... The main workspace is currently empty, but a dialog box is open in the center. The dialog box has a title bar with a close button (X) and contains the following elements:

- Diagram name:
- Templates:
- A grid of diagram templates, including:
 - A complex flowchart with multiple nodes and connections.
 - A hierarchical tree diagram.
 - A large, detailed network diagram with many nodes and connections.
 - A diagram showing a database schema with tables and relationships.
 - A diagram showing a software design with components and their interactions.
 - A diagram showing a flowchart with a central box labeled "software design/database.xml".
 - A diagram showing a network diagram with nodes and connections.
- Buttons at the bottom: , ,

At the bottom of the application, there is a footer with the following text and icons:

- JavaScript Diagramming Library
- JIRA / Confluence Plugin
- Social media icons for e, g+, f, and t.

E-R Diagramming Tools – draw.io



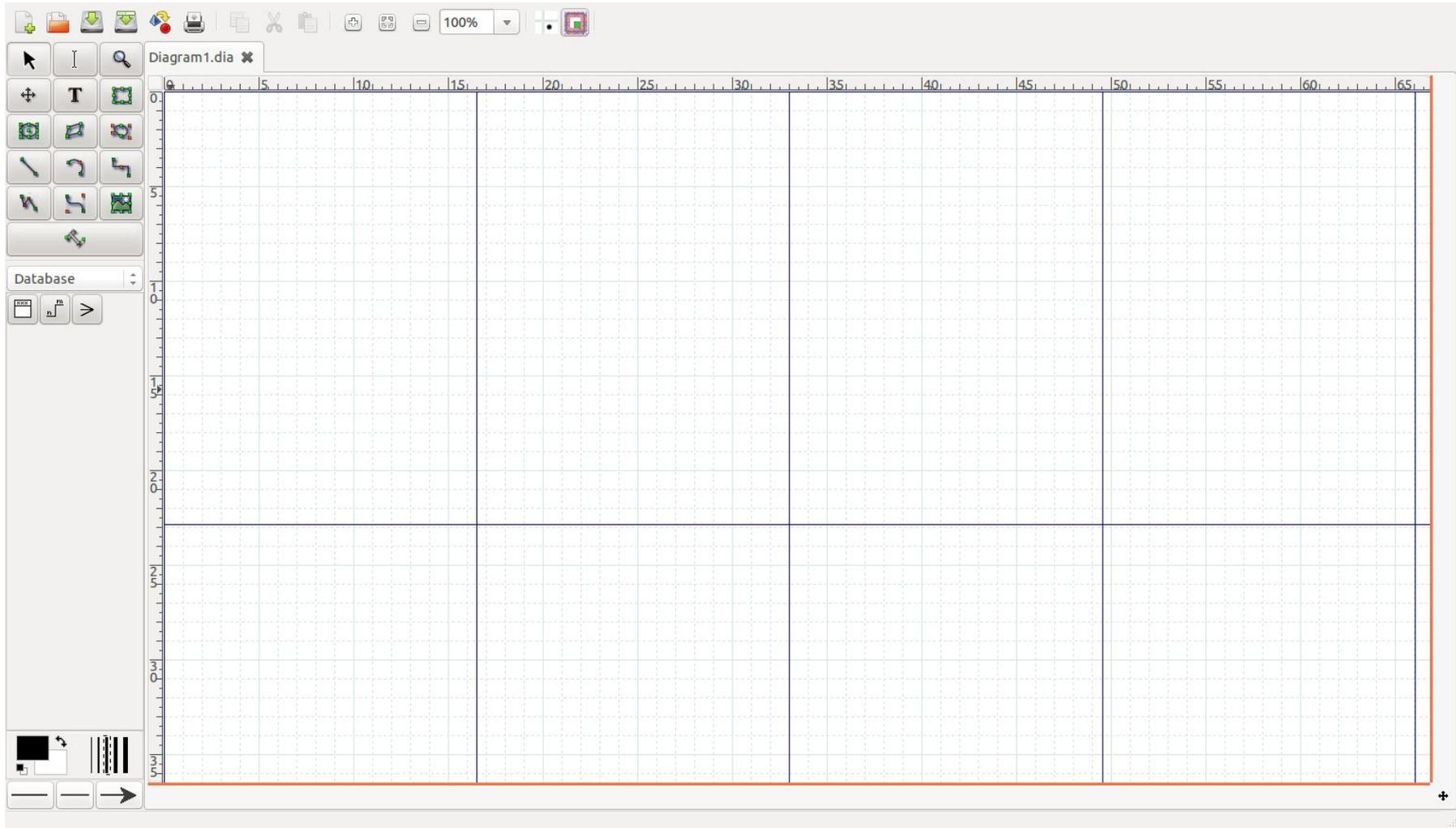
E-R Diagramming Tools – dia

The screenshot shows the Linux Software Center interface. At the top, there is a navigation bar with icons for 'All Software', 'Installed', 'History', and 'Progress'. A search bar on the right contains the text 'dia'. Below the navigation bar, the 'All Software' section is displayed, with a 'By Relevance' dropdown menu. The search results for 'dia' are listed below, with the top result, 'Diagram editor dia', highlighted in orange. This result includes a 'More Info' button and an 'Install' button. The list of results includes:

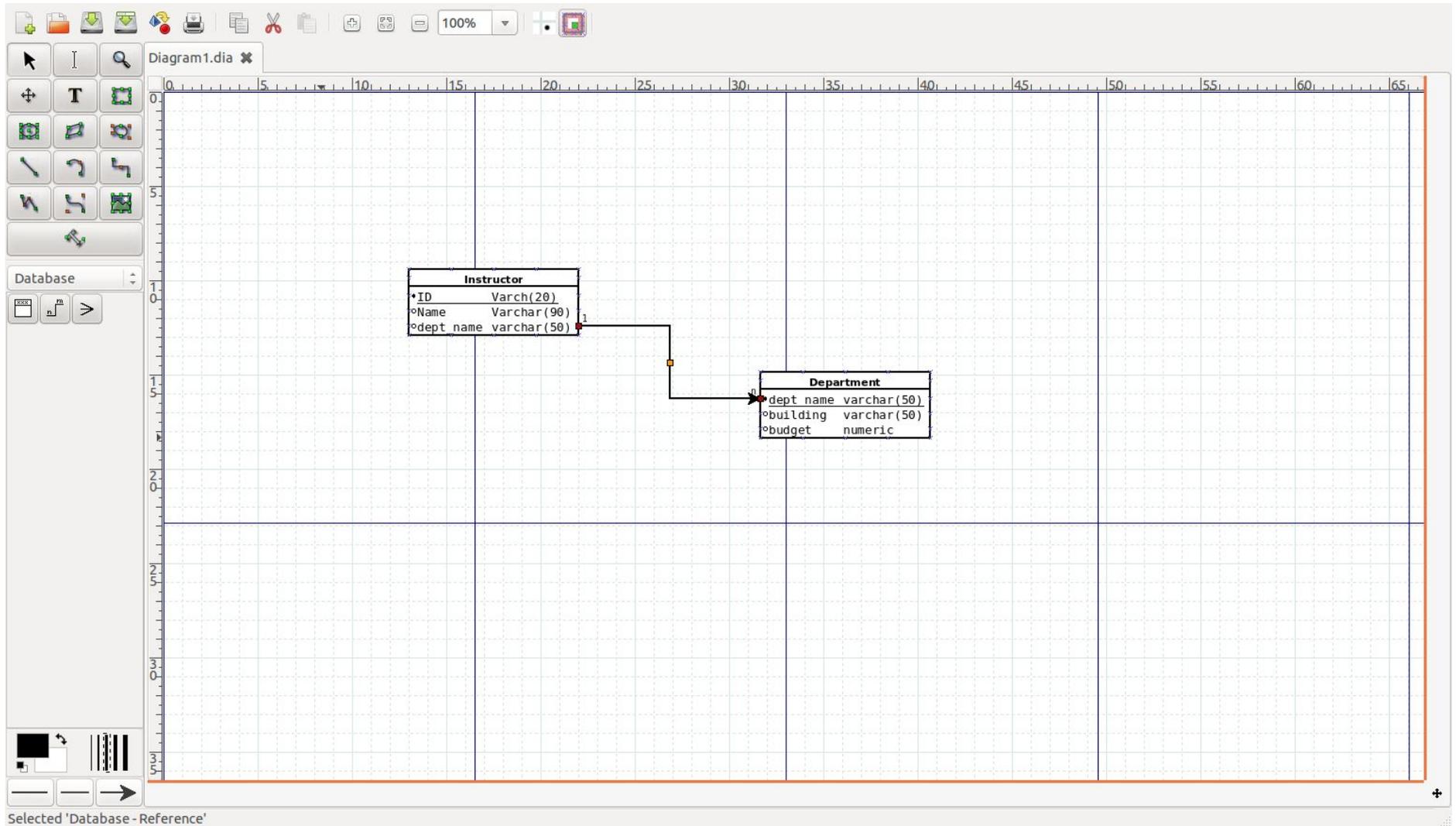
- Diagram editor dia** (24 ratings) - [More Info](#) [Install](#)
- Dia** - Edit your Diagrams
- Dialer** - Dialer application
- Dianara** - A pump.io social network client
- Revista Espirito Livre #38** - Revista livre e independente sobre Software Livre
- Revista Espirito Livre #42** - Revista livre e independente sobre Software Livre
- Revista Espirito Livre #35** - Revista livre e independente sobre Software Livre
- Ubuntu - Guia do Iniciante 2.0** (3 ratings) - Um livro sobre Ubuntu de A a Z
- Revista Espirito Livre #32** - Revista livre e independente sobre Software Livre
- Revista Espirito Livre #4** (2 ratings) - Revista livre e independente sobre Software Livre
- Almanah Diary** (9 ratings) - Keep a personal diary
- xdiagnose** (34 ratings) - X.org Diagnostic and Repair Utility
- KPPP** (3 ratings) - Internet Dial-Up Tool
- Linux Magazine Issue 129 (Europe)** - Working with Windows. US\$ 9.99

At the bottom left, there is a link to [Show 667 technical items](#).

E-R Diagramming Tools – dia



E-R Diagramming Tools – dia



E-R Diagramming Tools – Visual Paradigm

Database Design (E... x +

www.visual-paradigm.com/features/database-design/

Database Design ☆

Supported from Standard Edition

Entity relationship diagram

Entity Relationship Diagram (ERD) is a database design tool that provides graphical representation of database tables, the columns in tables and the relationships between tables. With neat organization of tables, table columns and flexible representation of cardinalities, ERD is extremely helpful in modeling databases that have a large amount of tables and with complex relationships in between. A well-developed ERD can provide sufficient information for database administrator to follow when developing and maintaining database.

DepartmentStoreSync.vpp - Visual Paradigm Enterprise Edition

Entity Relationship Diagram3

AdminStaff

- varchar(255) N
- varchar(255) N
- int N
- varchar(255) N
- varchar(255) N

Report

- ID2 int N
- ReportID2 int N
- ProductID int N
- AdminStaffID23 int N
- AdminStaffID2 int N
- ReportCreateDate datetime N
- ReportVerifyDate datetime N
- EndoncedByDate datetime N
- ReportSubmitDate datetime N
- ID int N
- Status int N
- MustCompliant bit N
- ReportNo varchar(255) N
- StaffName varchar(255) N
- StaffNo varchar(255) N

ItemPhoto

- ID2 int N
- ProductItemID2 int N
- ProductItemID2 int N
- ReportID2 int N
- ID int N
- ItemPhoto varchar(255) N
- PhotoPath varchar(255) N
- PhotoDesc varchar(255) N
- ReportIndex int N
- ProductItemIndex int N
- Column int N

CardRefundReport

- RefundDate datetime N
- ThroughBank varchar(255) N
- TransactionNo varchar(255) N
- Remarks varchar(255) N
- ReportID2 int N
- RefundReportReportID2 int N

RefundSummary

- ReportID2 int N
- RefundSummaryID2 int N
- RefundInspectorID2 int N

ReportDetail

- ID2 int N

Relationships:

- worksCheckedBy: AdminStaff (1) to Report (M)
- endorseBy: AdminStaff (1) to Report (M)
- inspector: AdminStaff (1) to Report (M)
- mmReport: Report (1) to RefundSummary (M)
- siteObservationSummary: Report (1) to RefundSummary (M)
- Report: Report (1) to ItemPhoto (M)
- Report: Report (1) to CardRefundReport (M)
- Report: Report (1) to RefundSummary (M)
- Report: Report (1) to ReportDetail (M)
- Product: Product (1) to ItemPhoto (M)
- Product: Product (1) to CardRefundReport (M)
- Product: Product (1) to RefundSummary (M)
- Product: Product (1) to ReportDetail (M)

Visual Paradigm 11.2 Build 20140906 ChangeLog