Spring 2020 2020/02/19

GEOG 580
Data Management for GIS

Entity-Relationship Data Model

PostgreSQL PostGIS

Recap

- Structured Query Language (SQL)
  - Combining queries
  - Join
  - Subqueries
  - Aggregates & Misc.

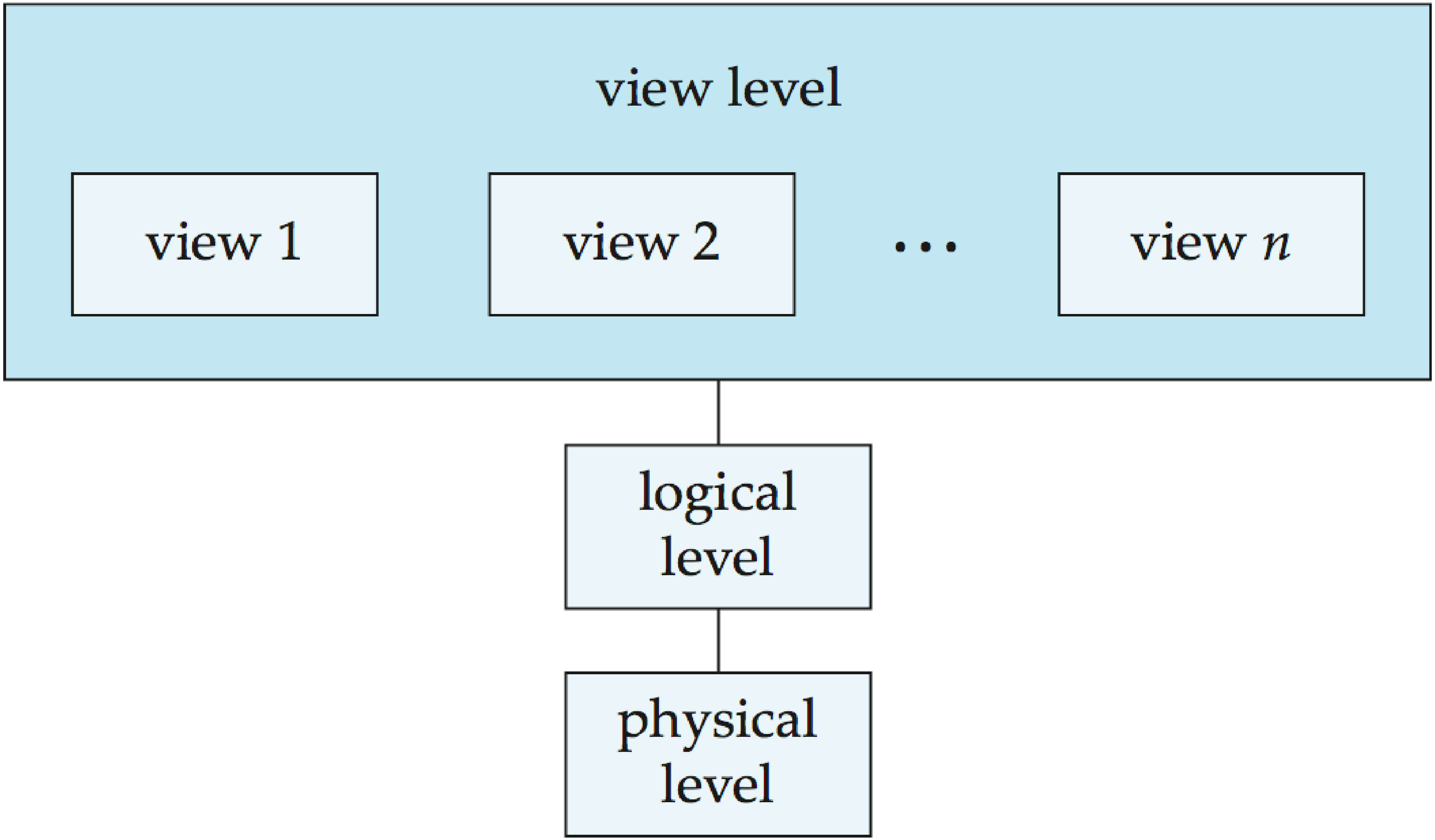
Topic

- Entity-Relationship (ER) Data Model
- Database Normalization

Entity Relationship (ER) Data Model

ER Data Model

Levels of Abstraction



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Levels of Abstraction

- **Physical level:** describes how a record (e.g., a student) is stored.

- **Logical level:** ***describes what data*** are stored in the database, and ***what relationships exist*** among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures. Implementation of the simple structures at the logical level may involve complex physical-level structures.

- **View level:** describes only part of the entire database. Application programs hide details of data types. Views can also hide information for security purposes. The system may provide many views for the same database. Many users need to access only a part of the database.

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

ER Data Model

DB Design Phase

1. Characterize fully the data needs of the prospective database users

2. Select a data model
   e.g.) relational, object-oriented, …

3. Apply the concepts of the selected data model and translates these requirements into a conceptual schema of the database

4. Logical Design: Deciding on the database schema

5. Physical Design: Deciding on the physical layout of the database

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Design Approaches: ER Data Model

- ❑ Facilitates database design by allowing specification of an enterprise schema that represents the over all logical structure of a database.

- ❑ Maps the meanings and interactions of real-world enterprises onto a conceptual schema.

- ❑ Three basic concepts:
  1. ***Entity sets***
  2. ***Relationship sets***
  3. ***Attributes***

- ❑ ER Diagram: A diagrammatic representation of the ER data model

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

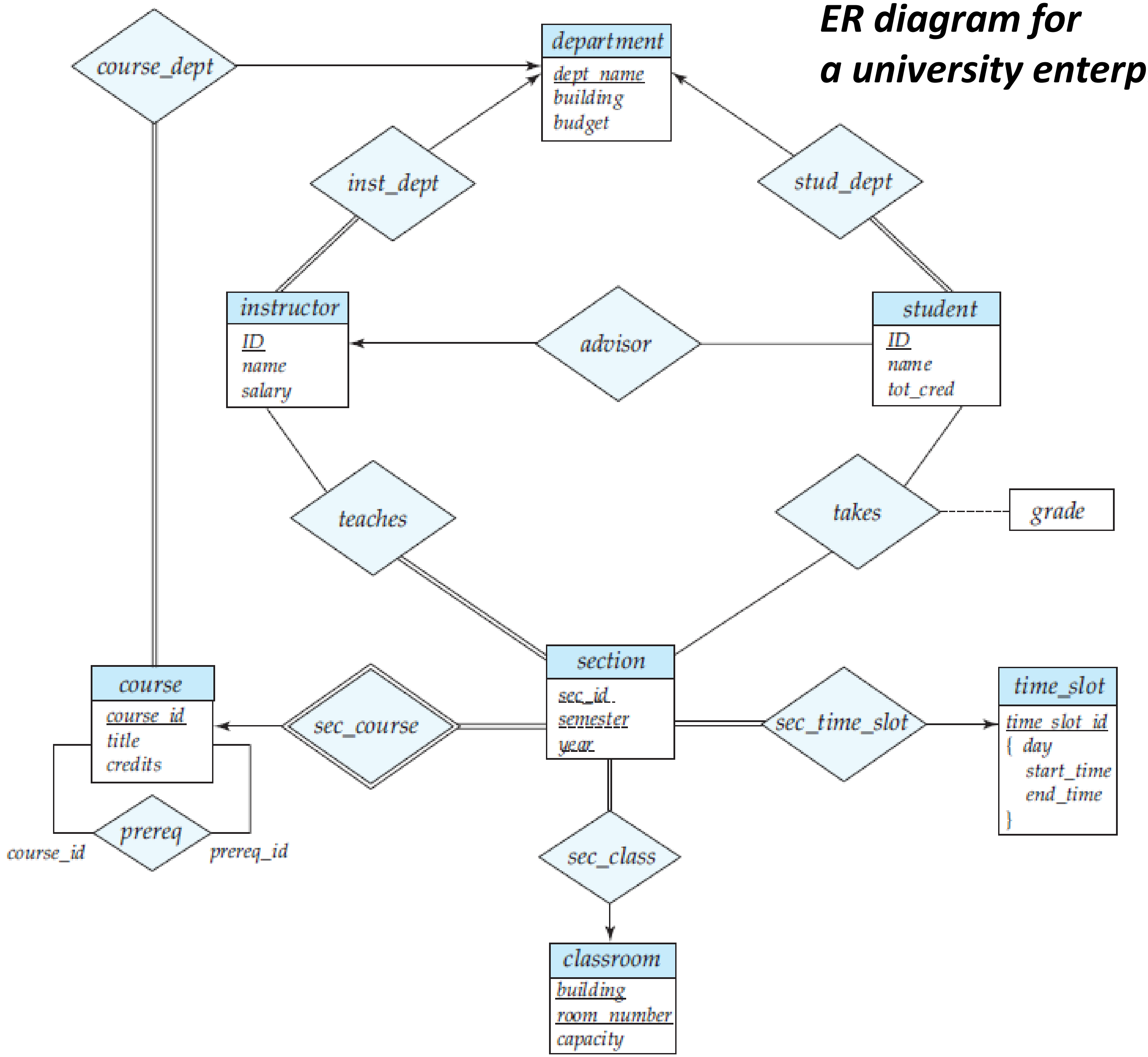
ER Data Model

- ❑ Fac
  ent cture
  of a

- ❑ Ma rises
  ont

- ❑ Thr
  1.
  2.
  3.

- ❑ ER a
  mo



ER diagram for
a university enterprise

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

ER Data Model

Entity Sets

- An ***entity*** is a “thing” or “object” in the real world that is distinguishable from all other objects.
  e.g.) *specific person, company event*

- An ***entity set*** is a set of entities of the same type that share the same **properties**, or **attributes**.
  e.g.) *instructor=(ID, name, age, salary)*
  course=(course_id, title, credits, capacity)

- Each entity has a **value** for each of its **attributes**.
  e.g.) *instructor1 = (12121, Smith, 45, 90000)*

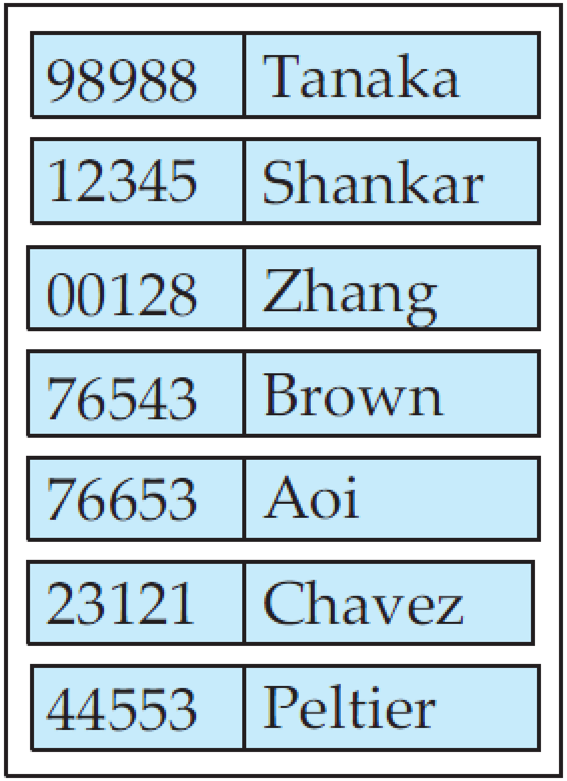
- **Primary Key**: An attribute of the entity set, which uniquely identifying each member of the set

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

ER Data Model

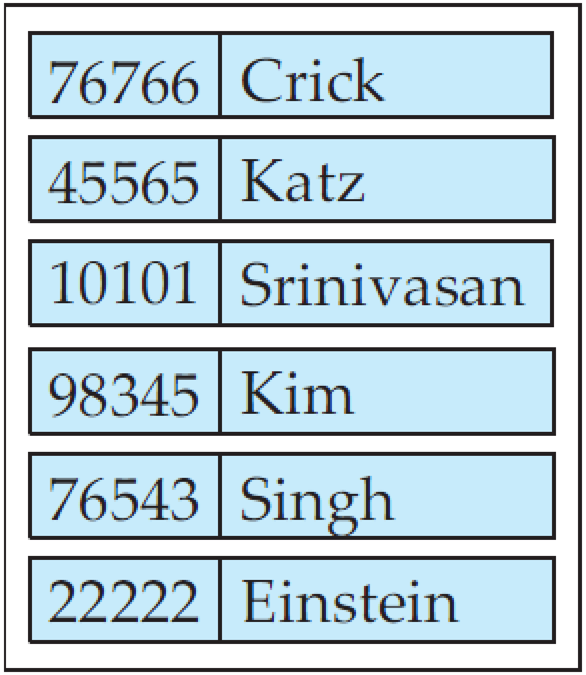
Entity Sets

Entity sets *instructor* and *student*



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

student



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

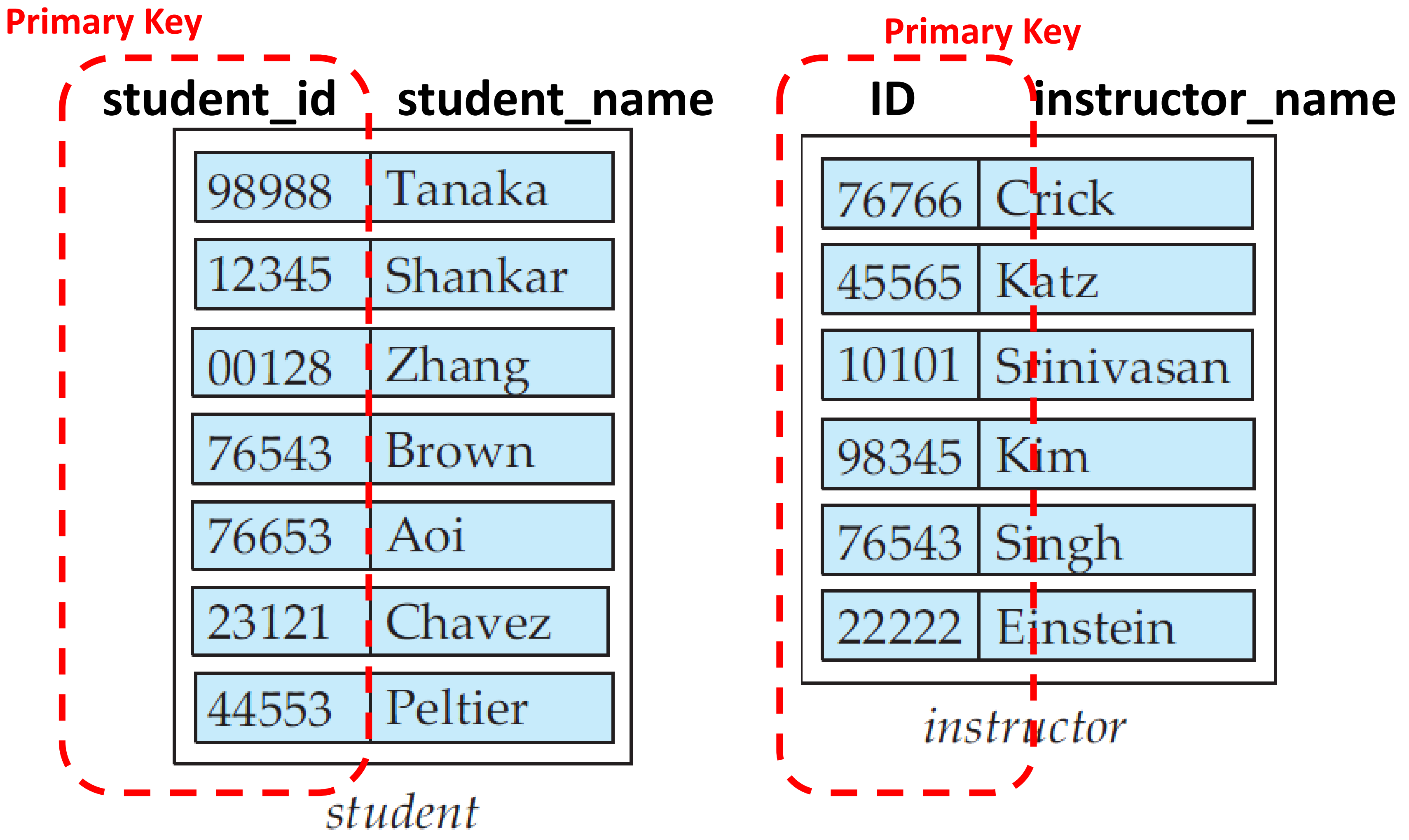
instructor

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

ER Data Model

Entity Sets

Entity sets *instructor* and *student*


Primary Key
Primary Key

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

student

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

instructor

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Relationship Sets

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

e.g.)

44553 (Peltier)	**advisor**	22222 (Einstein)
student entity	***relationship***	*instructor* entity

❑ A ***relationship set*** is a set of relationships of the same type

- A mathematical relation among n ≥ 2 entities, each taken from entity sets

$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

where $(e_1, e_2, \ldots, e_n)$ is a relationship

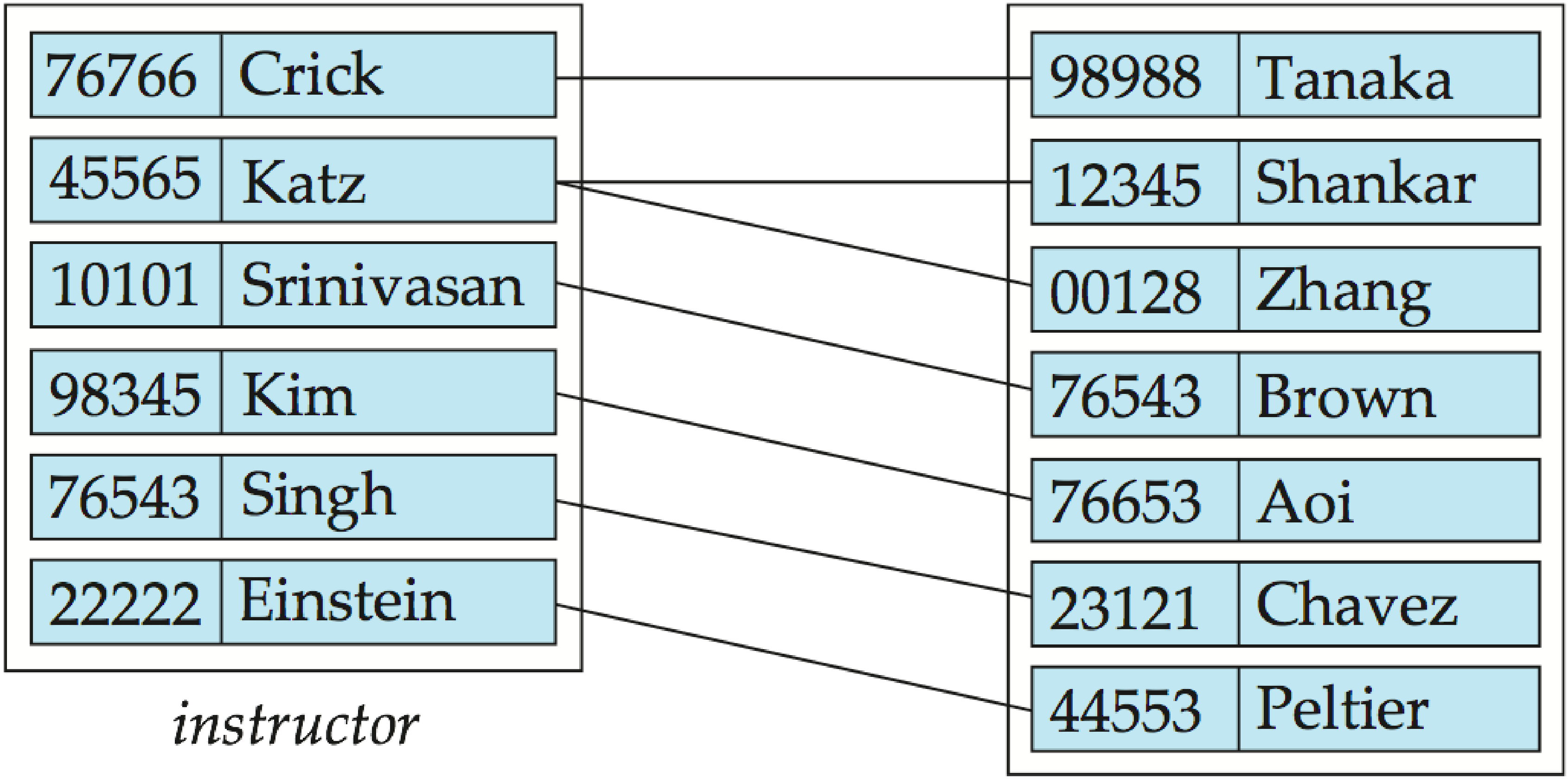
e.g.)

(44553, 22222) ∈ advisor

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

ER Data Model

Relationship Sets



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

instructor

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

student

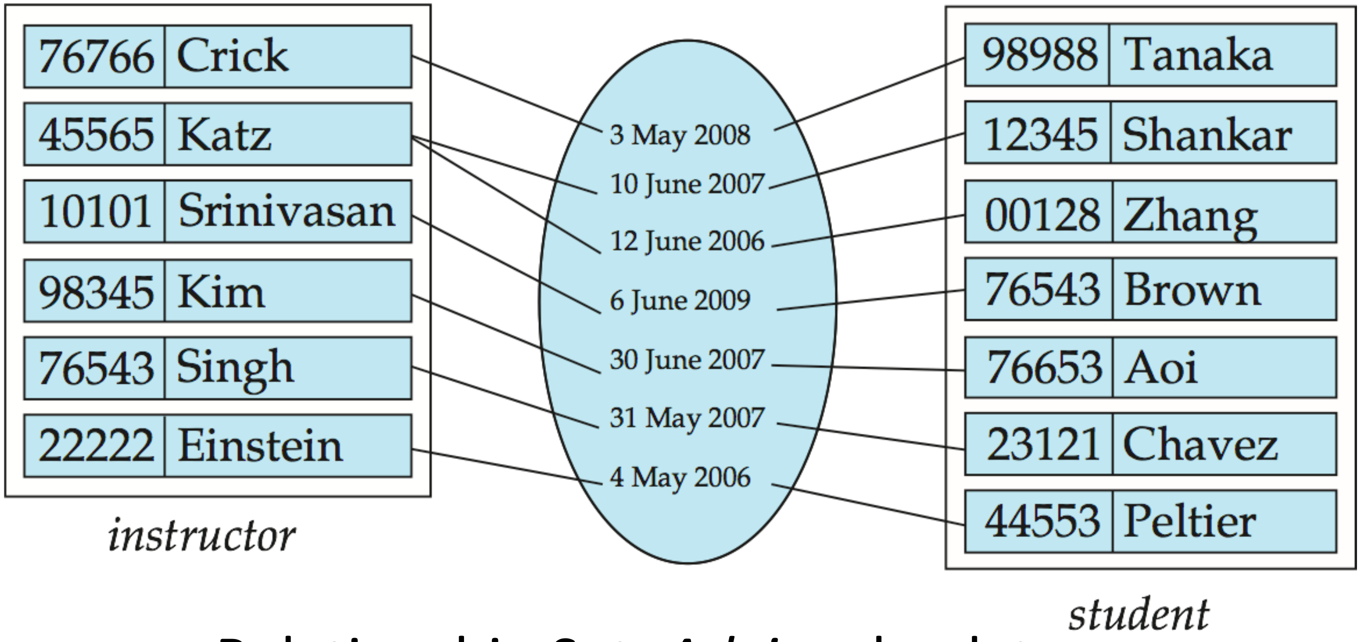
Relationship Set: *Advisor*

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Relationship Sets

- ❑ An ***attribute*** can be associated with a ***relationship set***

  e.g.) the *advisor* relationship set between *instructor* and *student* may have the attribute date which tracks when the student started being associated with the advisor.



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

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

Relationship Set: *Advisor* by date

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Relationship Sets: Degree

❑ Binary relationship

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

❑ Relationships between more than two entity sets are rare (most relationships are binary)

e.g.)

students work on research projects under the guidance of an instructor.

* relationship *proj_guide* is a ternary relationship between *instructor*, *student*, and *project*

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Relationship Sets: Cardinality Constraints

- Express the number of entities to which another entity can be associated via a relationship set

- Most useful in describing binary relationship sets

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

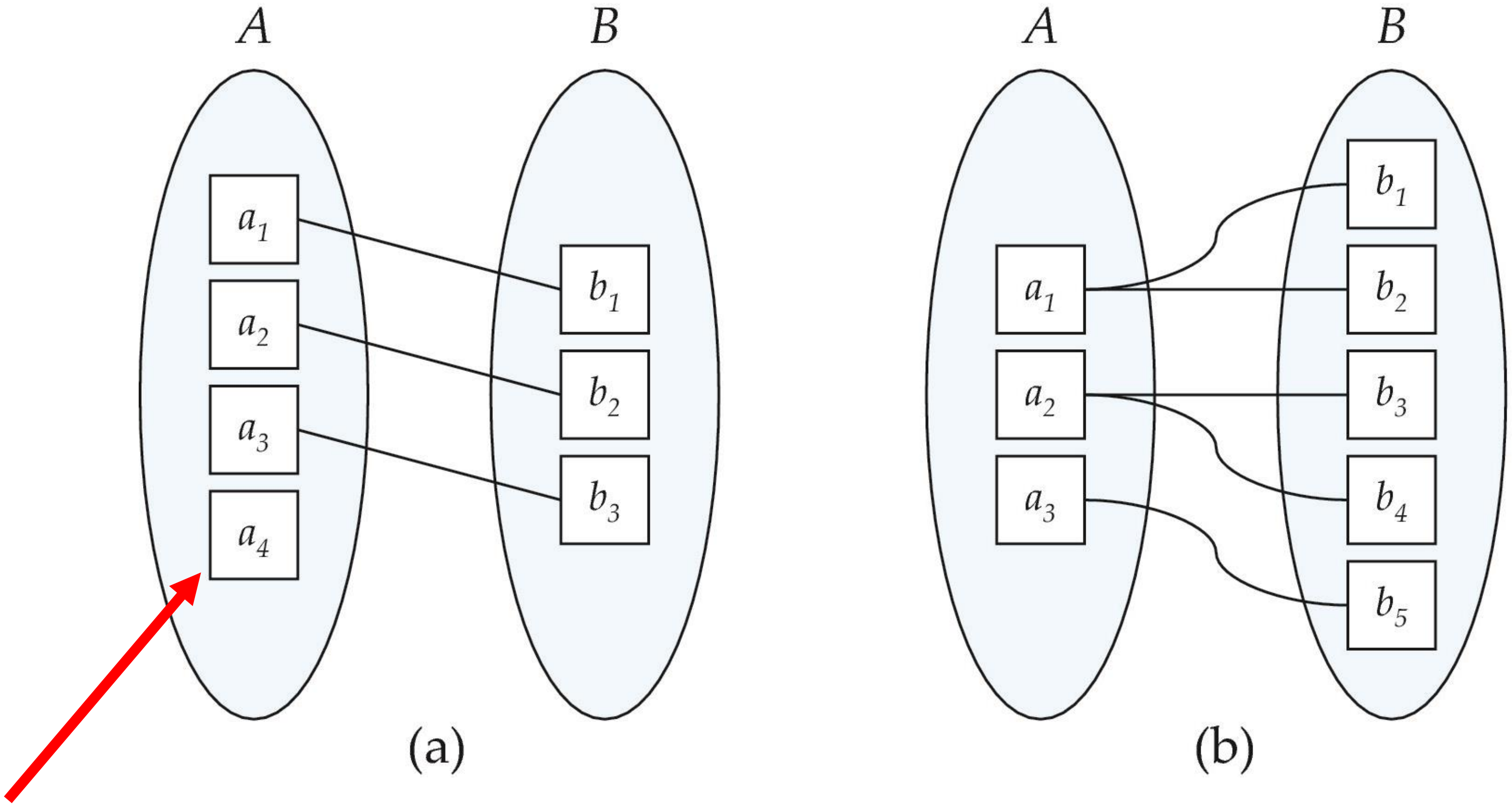
  - **one to one**
  - **one to many**
  - **many to one**
  - **many to many**

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

ER Data Model

Relationship Sets: Cardinality Constraints

one to one **one to many**



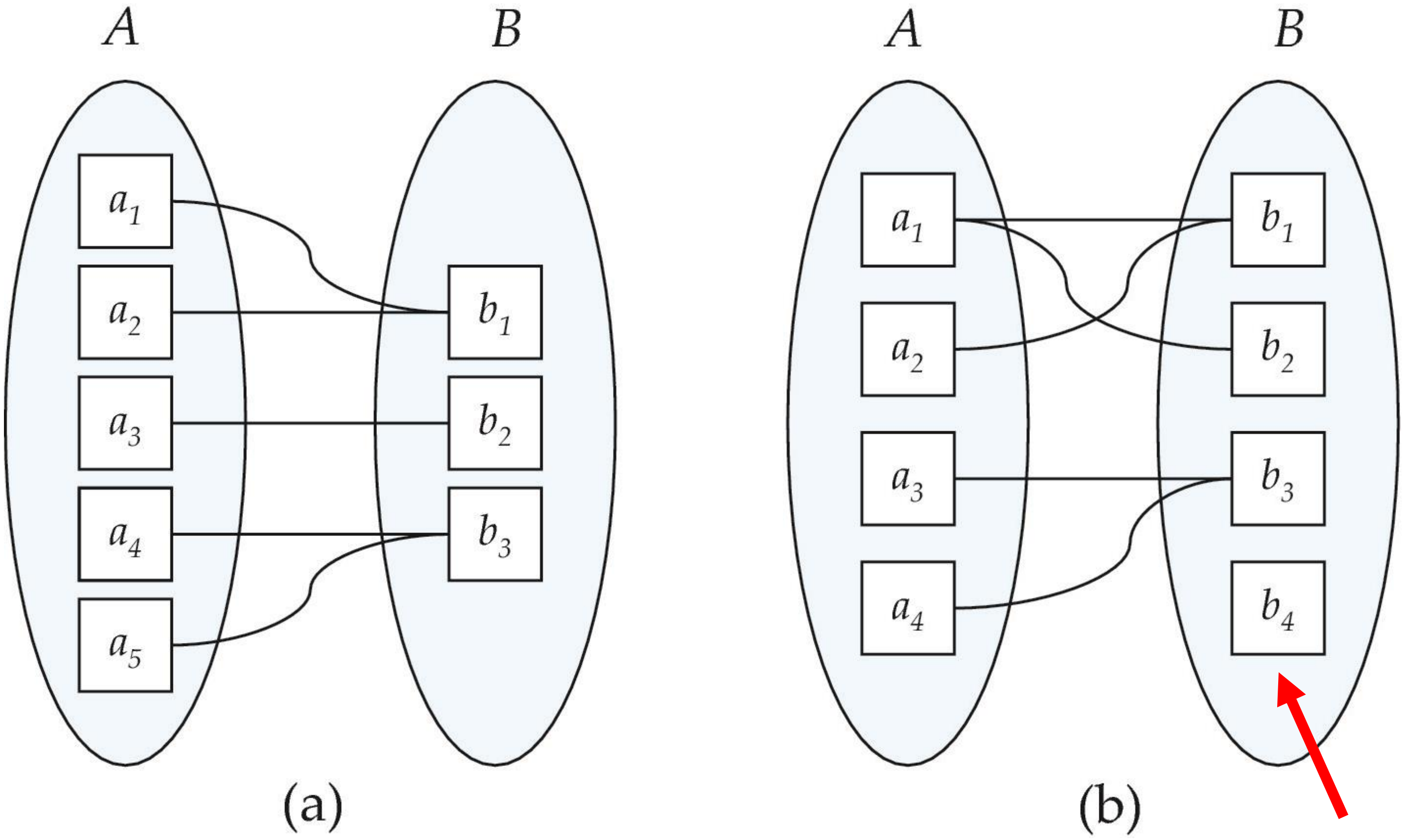
(a) (b)

* Some elements in A and B may not be mapped to any elements in the other set

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Relationship Sets: Cardinality Constraints

many to one **many to many**



* Some elements in A and B may not be mapped to any elements in the other set

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Attributes

- An attribute (or field) defines a characteristic of an entity set.

- Attribute types:
  - **Simple** and **Composite** attributes
    - Composite attributes can be divided into subparts (e.g., address)
  - **Single-valued** and **Multivalued** attributes

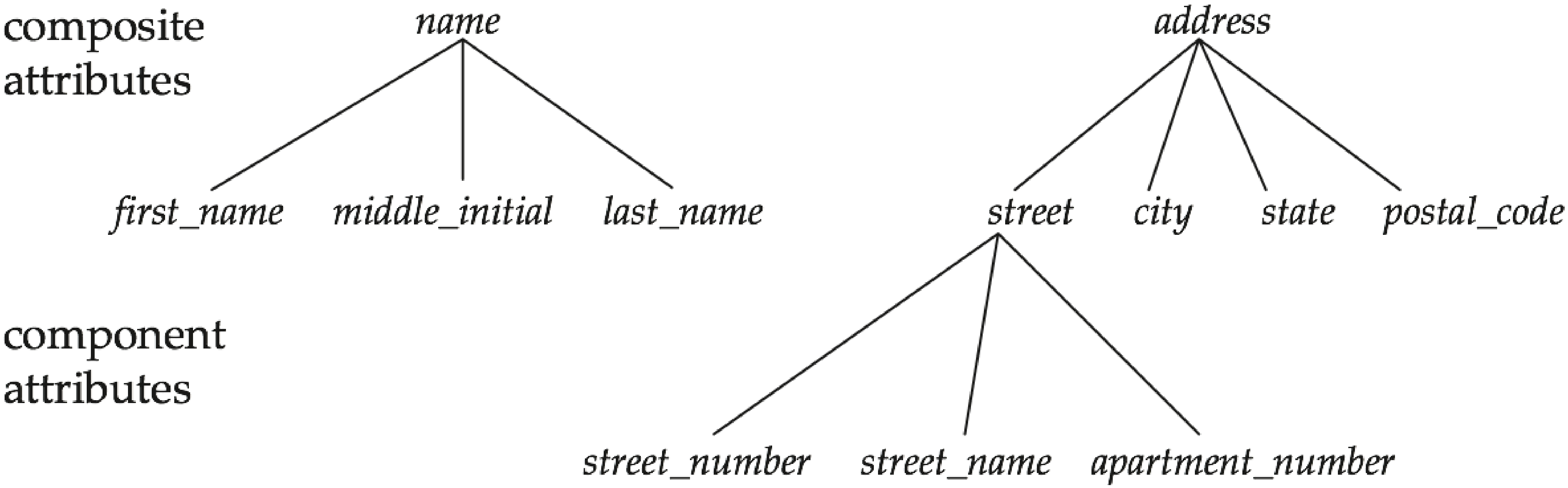
    e.g.) Multivalued attributes: phone number (home, work, cell)
  - **Derived** attributes
    - Can be computed from other attributes (e.g. age, given DOB)

- **Domain**: the set of permitted values for each attributes

  e.g.) class grade = {A, B, C, D, E, F}

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Attributes



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

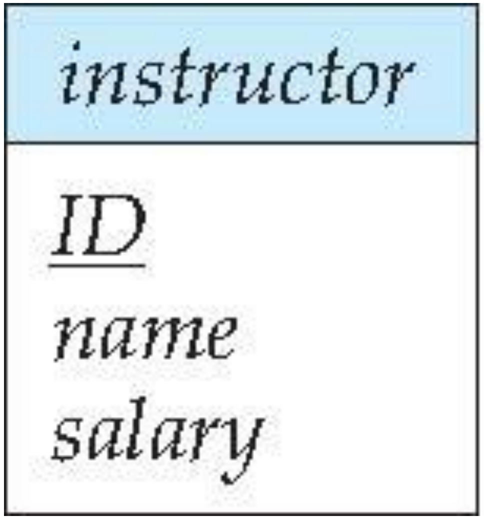
E-R Diagram

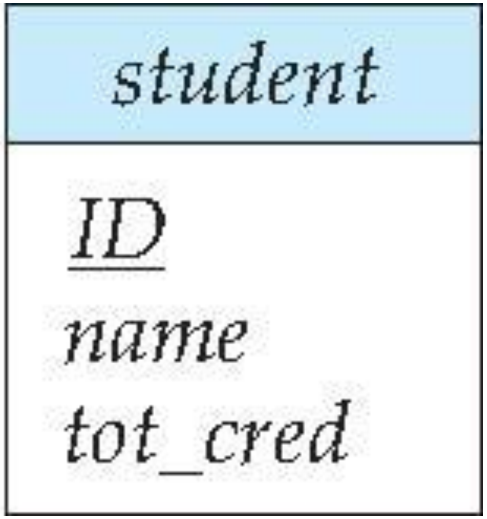
E-R Diagram

Entity Sets

❑ Entities can be represented graphically as follows:

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



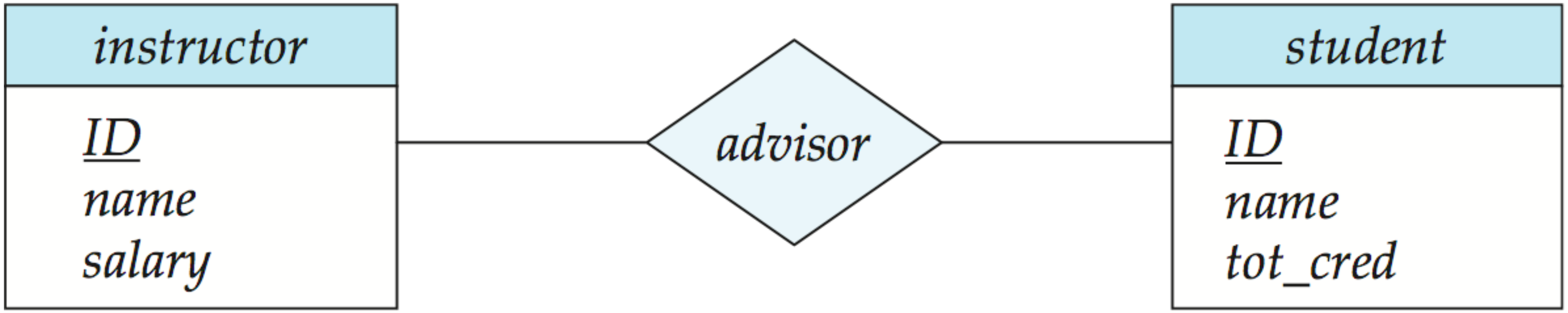


Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

E-R Diagram

Relationship Sets

- Diamonds represent relationship sets



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

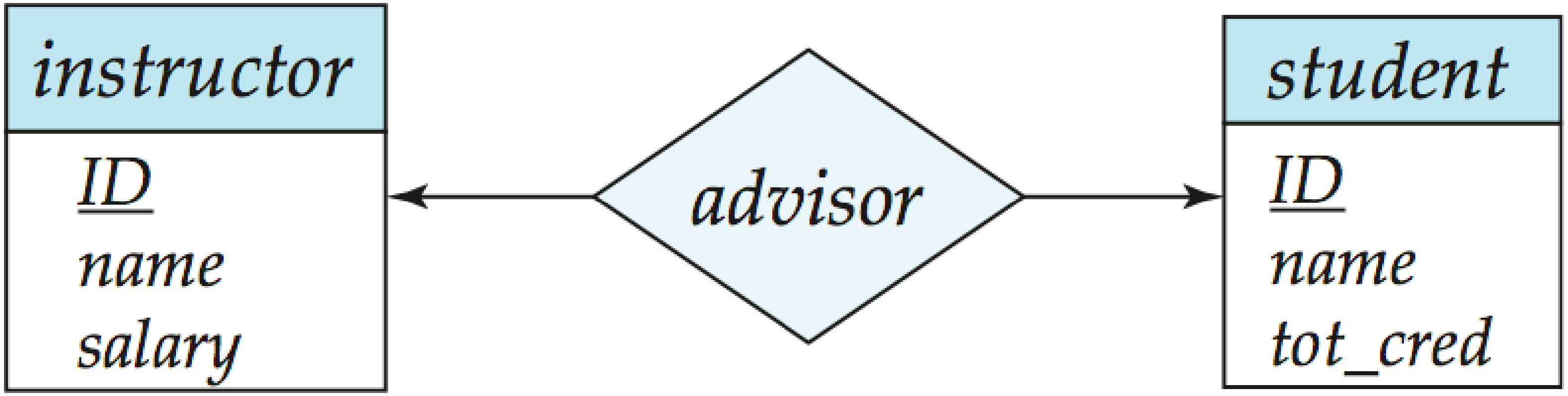
E-R Diagram

Cardinality Constraints

❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **One-to-one relationship** between an *instructor* and a *student* :

- A *student* is associated with at most one *instructor* via the relationship *advisor*
- A *student* is associated with at most one *department* via *stud_dept*



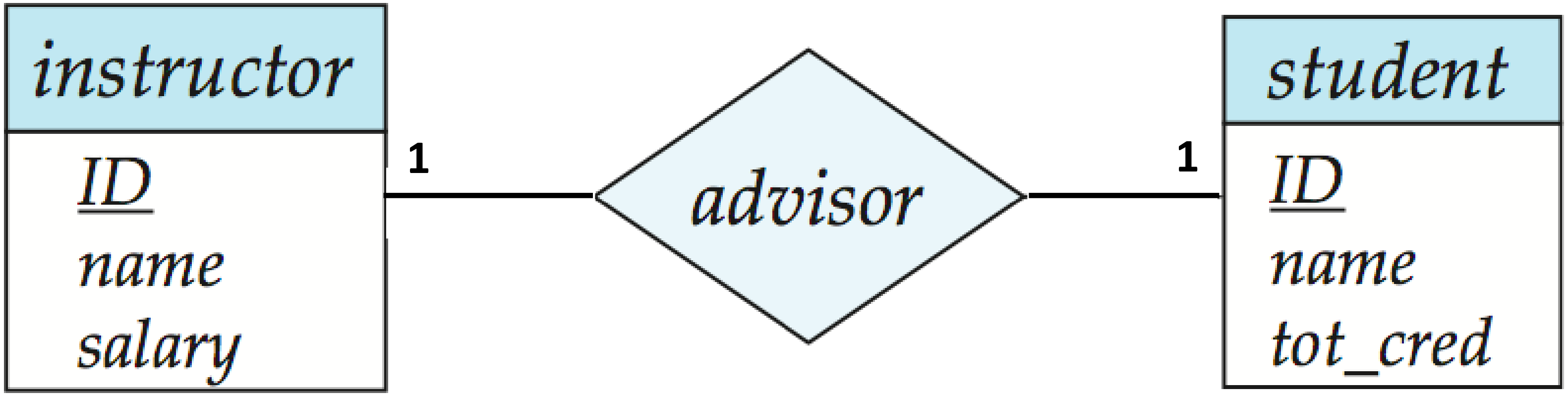
Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Cardinality Constraints

- ❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **One-to-one relationship** between an *instructor* and a *student* :

- A *student* is associated with at most one *instructor* via the relationship *advisor*
- A *student* is associated with at most one *department* via *stud_dept*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

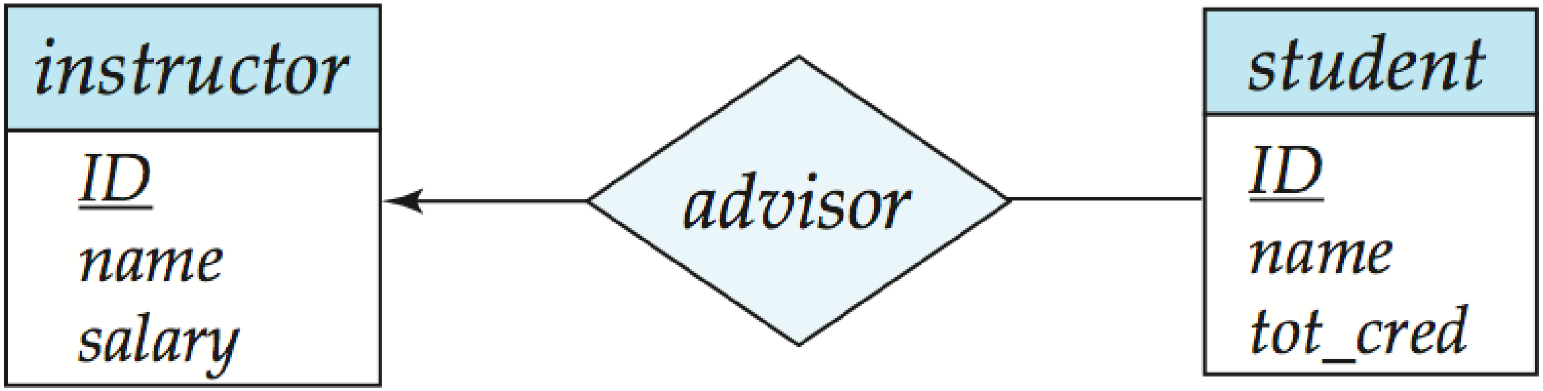
E-R Diagram

Cardinality Constraints

- Drawing either a directed line (→), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.

e.g.) **One-to-many relationship** between an *instructor* and a *student* :

- An *instructor* is associated with several (including 0) *students*
- A *student* is associated with at most one *instructor* via *advisor*



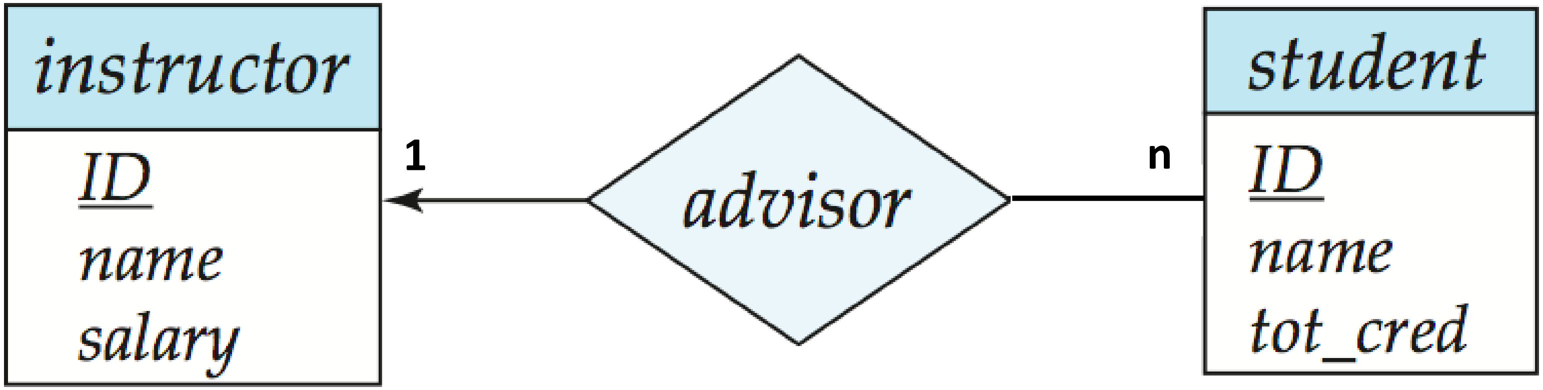
Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Cardinality Constraints

❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **One-to-many relationship** between an *instructor* and a *student* :

- An *instructor* is associated with several (including 0) *students*
- A *student* is associated with at most one *instructor* via *advisor*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

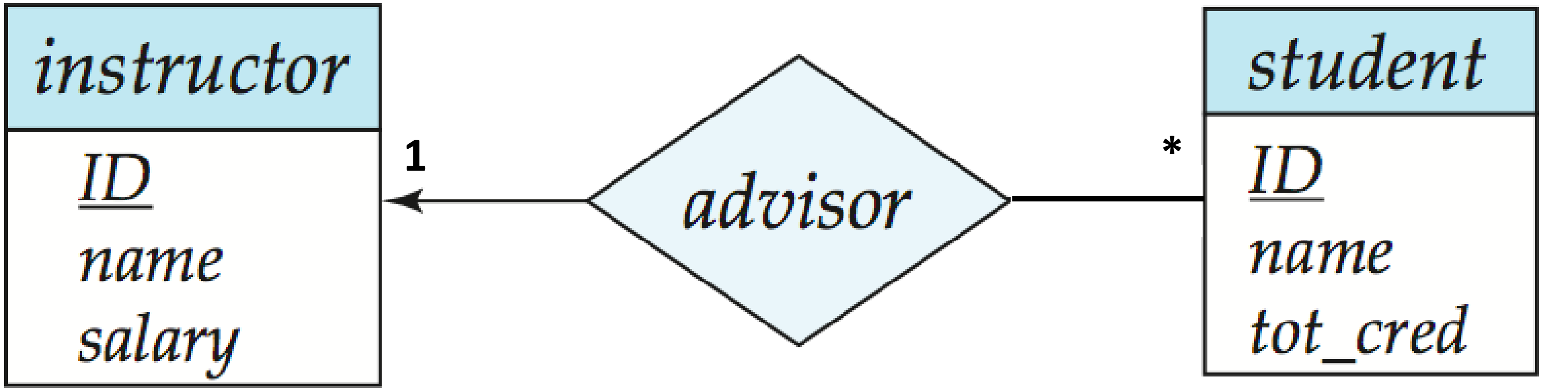
E-R Diagram

Cardinality Constraints

❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **One-to-many relationship** between an *instructor* and a *student* :

- An *instructor* is associated with several (including 0) *students*
- A *student* is associated with at most one *instructor* via *advisor*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

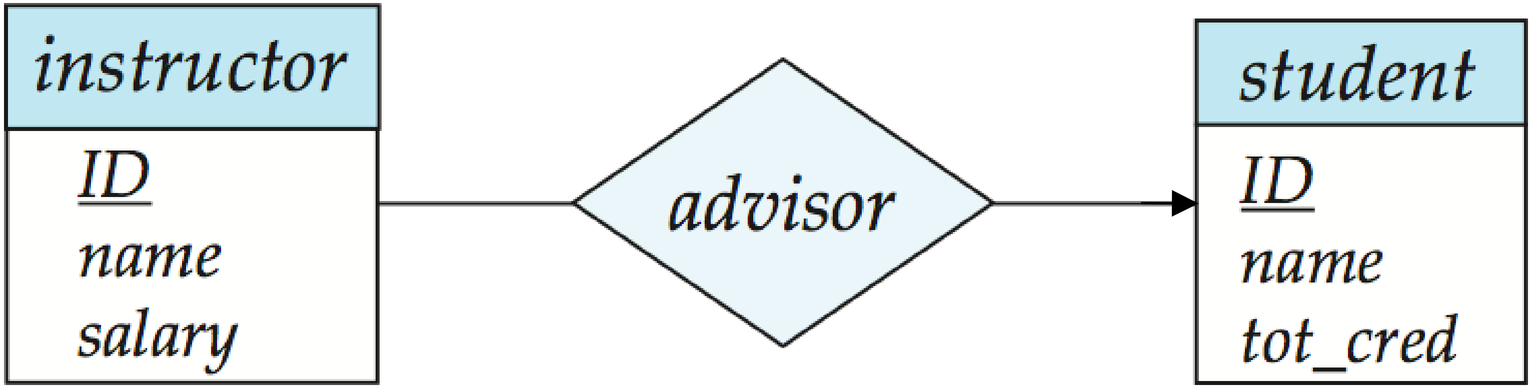
E-R Diagram

Cardinality Constraints

- ❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **Many-to-one relationship** between an *instructor* and a *student* :

- An *instructor* is associated with at most one *student* via *advisor*
- A *student* is associated with several (including 0) *instructors* via *advisor*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

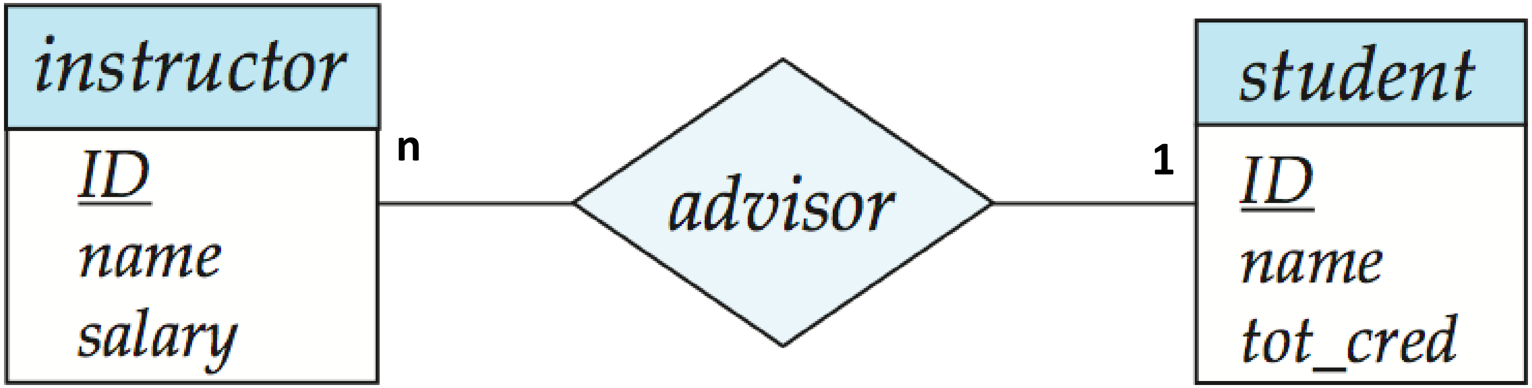
E-R Diagram

Cardinality Constraints

- ❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **Many-to-one relationship** between an *instructor* and a *student* :

- An *instructor* is associated with at most one *student* via *advisor*
- A *student* is associated with several (including 0) *instructors* via *advisor*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

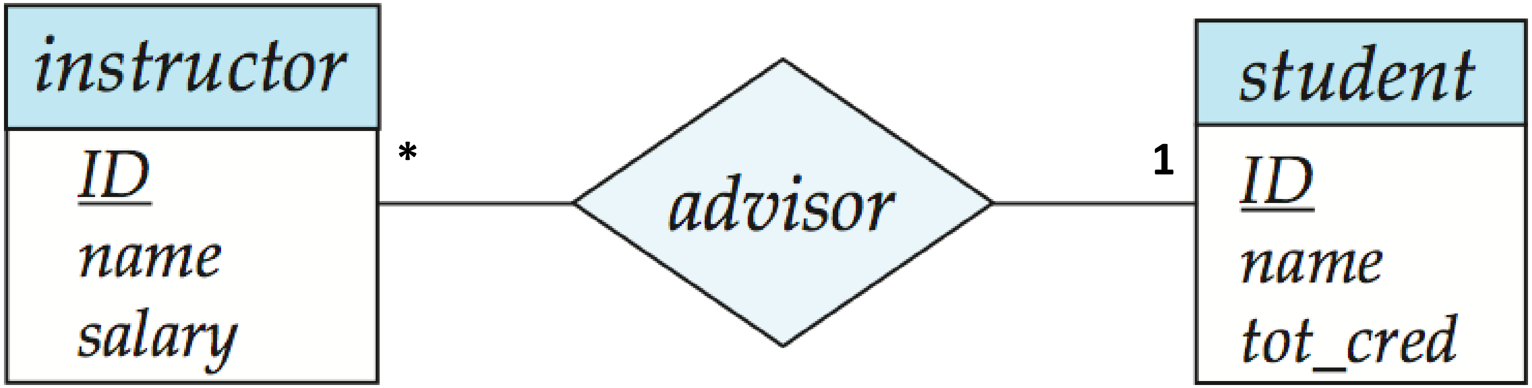
E-R Diagram

Cardinality Constraints

- ❑ Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **Many-to-one relationship** between an *instructor* and a *student* :

- An *instructor* is associated with at most one *student* via *advisor*
- A *student* is associated with several (including 0) *instructors* via *advisor*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

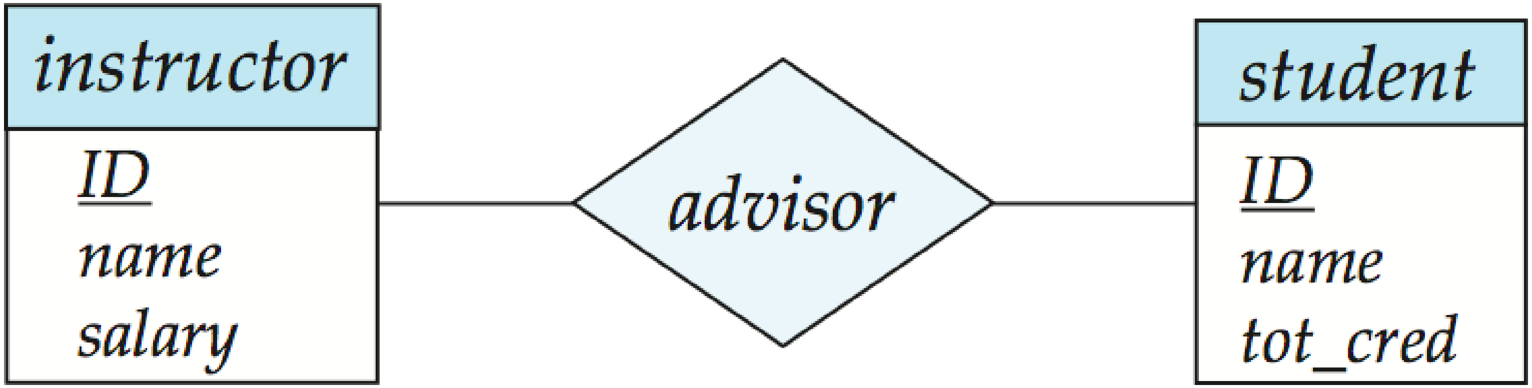
E-R Diagram

Cardinality Constraints

❑Drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

e.g.) **Many-to-many relationship** between an *instructor* and a *student* :

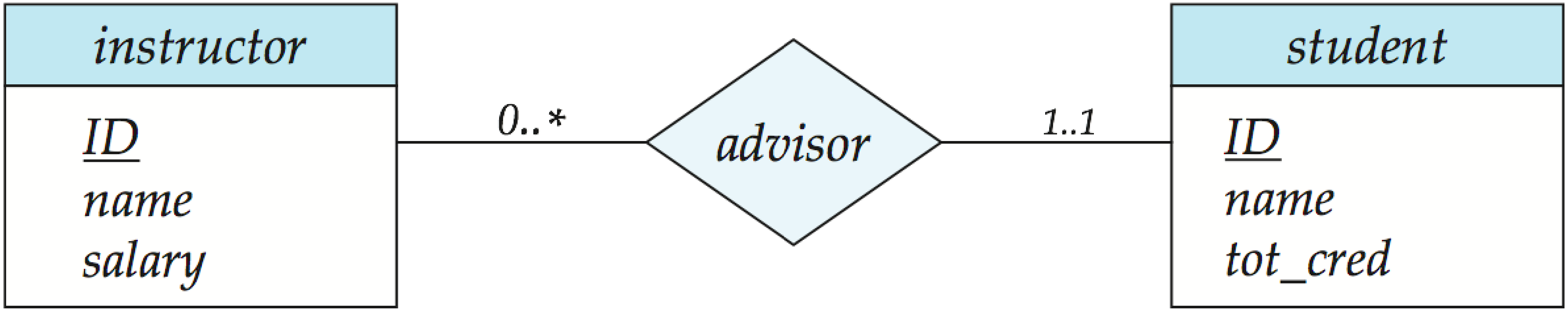
- An *instructor* is associated with several (including 0) *students* via *advisor*
- A *student* is associated with several (including 0) *instructors* via *advisor*



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

Cardinality Constraints

- A line may have an associated minimum and maximum cardinality, shown in the form *l..h*, where *l* is the minimum and *h* the maximum cardinality
  - A minimum value of 1 indicates total participation
  - A maximum value of 1 indicates that the entity participates in at most one relationship
  - A maximum value of * indicates no limit

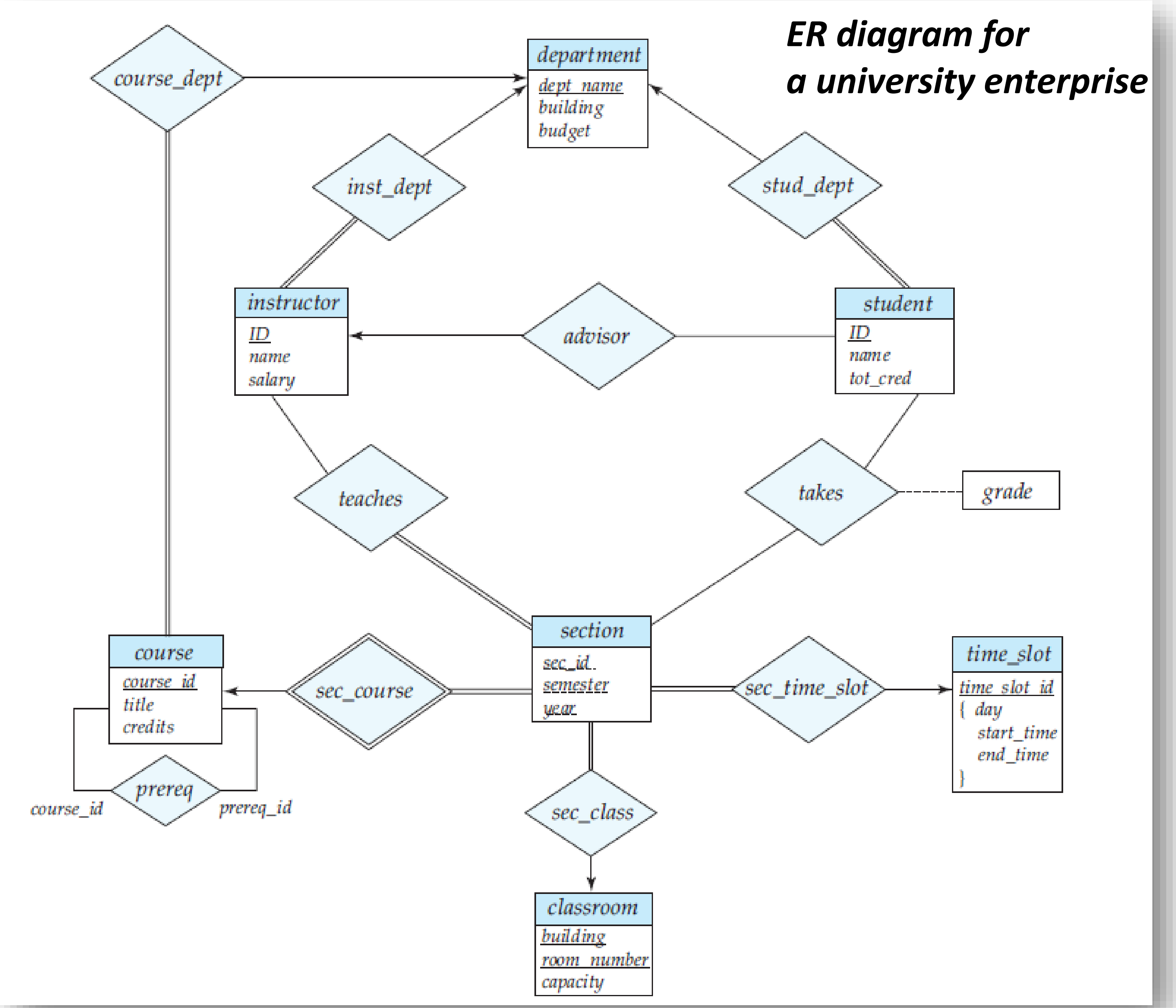


e.g.)

- Instructor can advise 0 or more students.
- A student must have 1 advisor; cannot have multiple advisor

Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

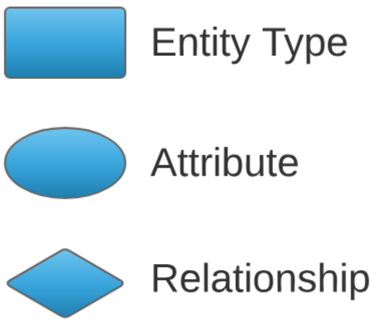
E-R Diagram



Source: Database System Concepts, Silberschatz, A., Korth, H.F., and Sudarshan, S. (2011)

E-R Diagram

ERA

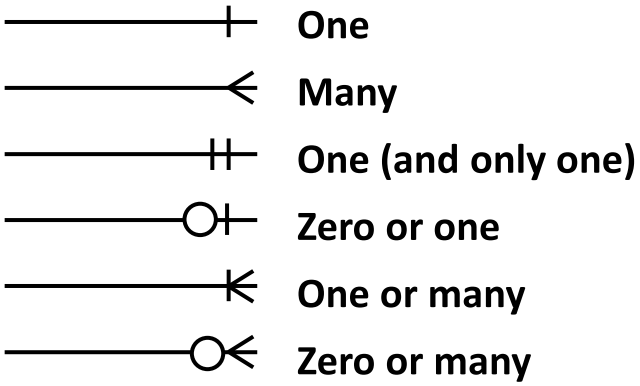


Entity Type

Attribute

Relationship

Crow's foot notation (Cardinality)



One

Many

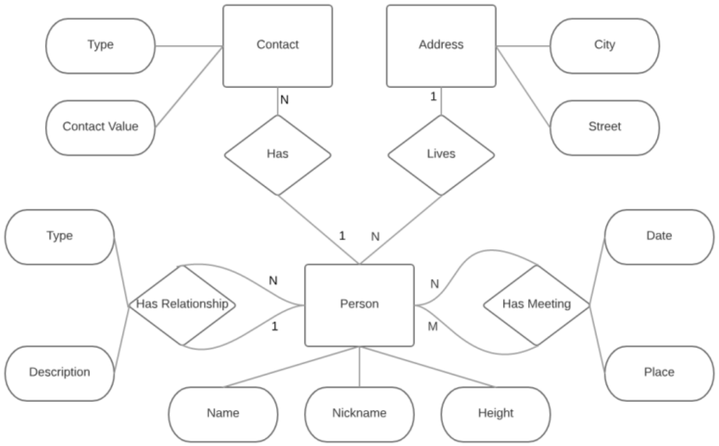
One (and only one)

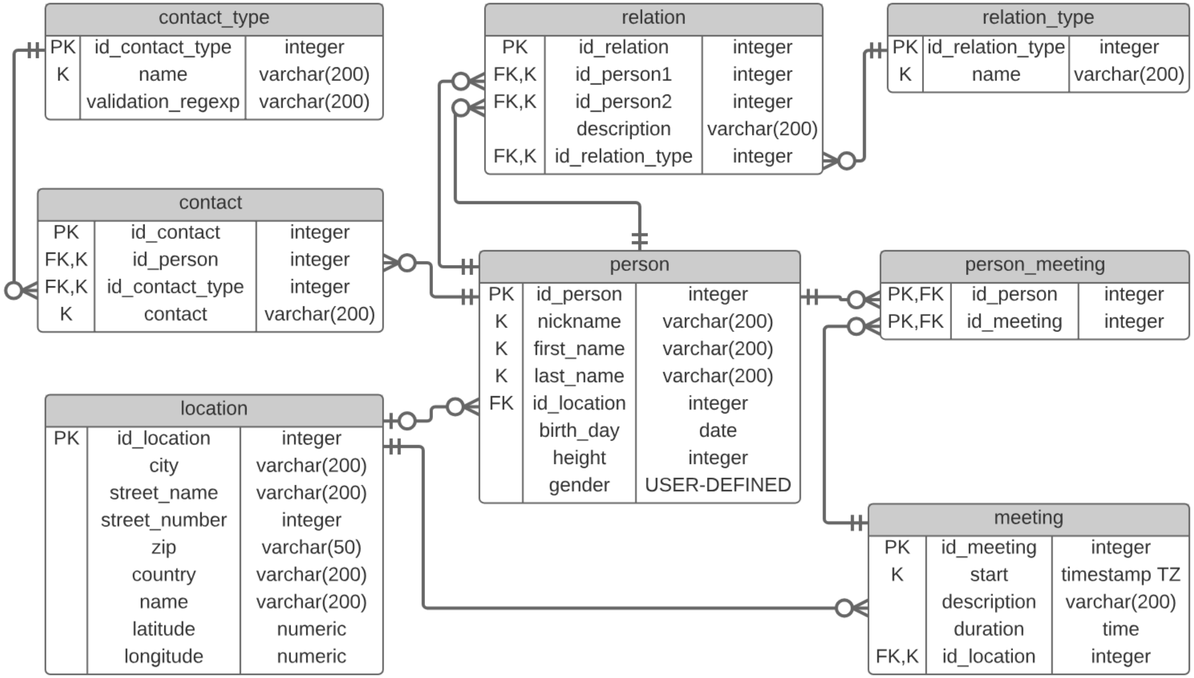
Zero or one

One or many

Zero or many

Logical Model





Physical Model with Crow's foot notation

contact_type

PK	id_contact_type	integer
K	name	varchar(200)
	validation_regexp	varchar(200)

relation

PK	id_relation	integer
FK,K	id_person1	integer
FK,K	id_person2	integer
	description	varchar(200)
FK,K	id_relation_type	integer

relation_type

PK	id_relation_type	integer
K	name	varchar(200)

contact

PK	id_contact	integer
FK,K	id_person	integer
FK,K	id_contact_type	integer
K	contact	varchar(200)

person

PK	id_person	integer
K	nickname	varchar(200)
K	first_name	varchar(200)
K	last_name	varchar(200)
FK	id_location	integer
	birth_day	date
	height	integer
	gender	USER-DEFINED

person_meeting

PK,FK	id_person	integer
PK,FK	id_meeting	integer

location

PK	id_location	integer
	city	varchar(200)
	street_name	varchar(200)
	street_number	integer
	zip	varchar(50)
	country	varchar(200)
	name	varchar(200)
	latitude	numeric
	longitude	numeric

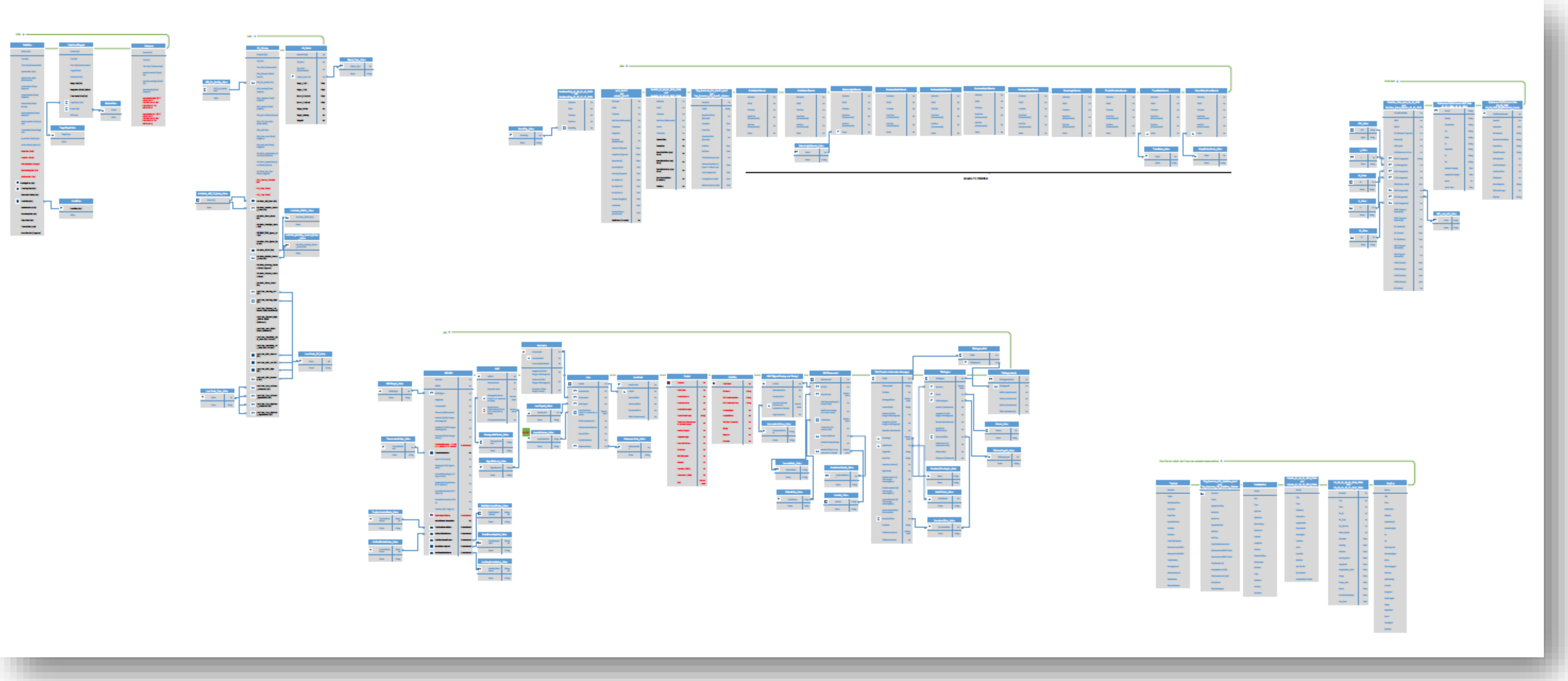
meeting

PK	id_meeting	integer
K	start	timestamp TZ
	description	varchar(200)
	duration	time
FK,K	id_location	integer

https://akela.mendelu.cz/~lysek/tmwa/articles/database-design/

E-R Diagram

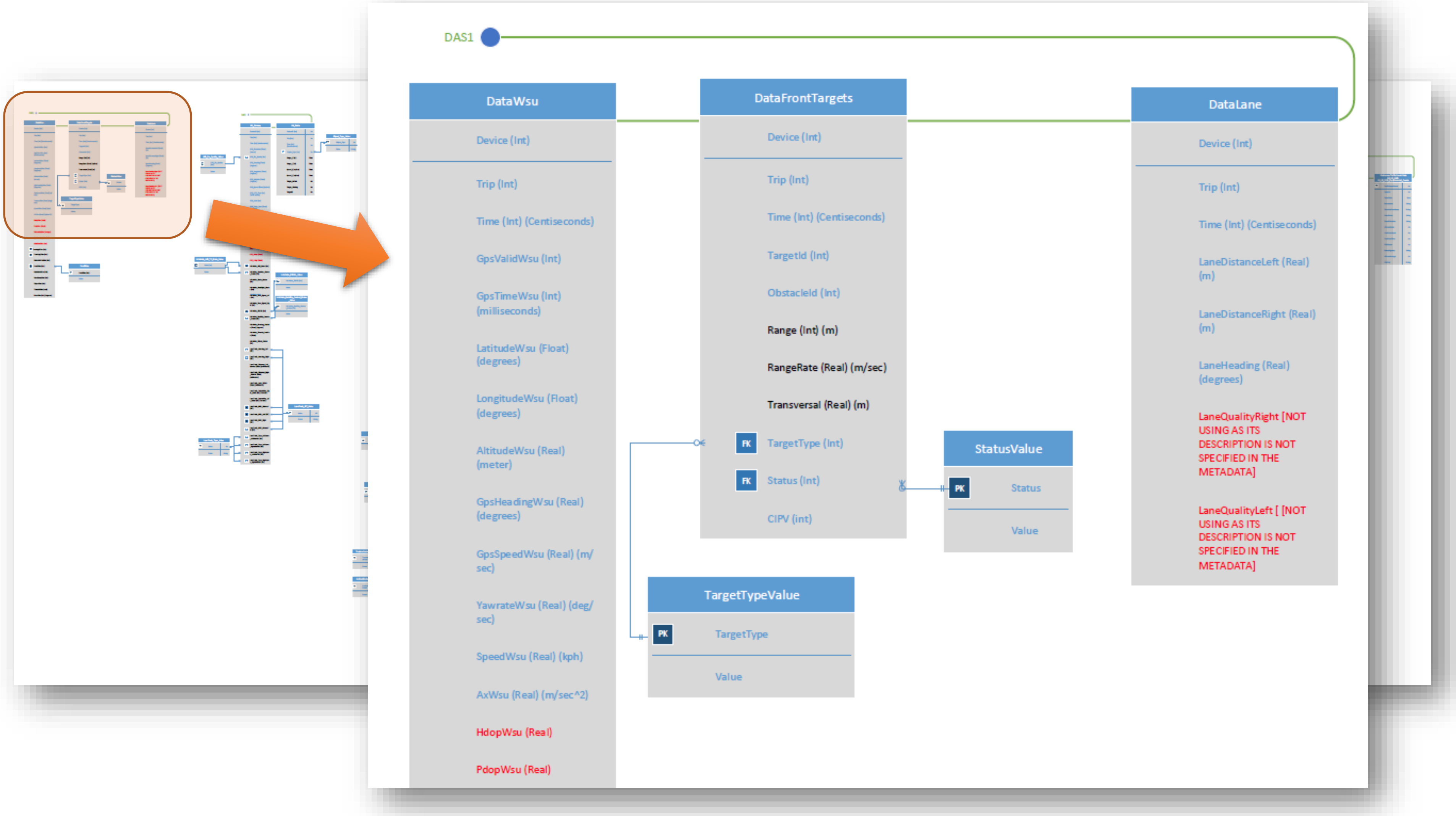
ER Diagram Example: SafeD Project



Using Microsoft Visio

E-R Diagram

ER Diagram Example: SafeD Project



DAS1

DataWsu

- Device (Int)
- Trip (Int)
- Time (Int) (Centiseconds)
- GpsValidWsu (Int)
- GpsTimeWsu (Int) (milliseconds)
- LatitudeWsu (Float) (degrees)
- LongitudeWsu (Float) (degrees)
- AltitudeWsu (Real) (meter)
- GpsHeadingWsu (Real) (degrees)
- GpsSpeedWsu (Real) (m/sec)
- YawrateWsu (Real) (deg/sec)
- SpeedWsu (Real) (kph)
- AxWsu (Real) (m/sec^2)
- HdopWsu (Real)
- PdopWsu (Real)

DataFrontTargets

- Device (Int)
- Trip (Int)
- Time (Int) (Centiseconds)
- TargetId (Int)
- ObstacleId (Int)
- Range (Int) (m)
- RangeRate (Real) (m/sec)
- Transversal (Real) (m)
- FK TargetType (Int)
- FK Status (Int)
- CIPV (int)

DataLane

- Device (Int)
- Trip (Int)
- Time (Int) (Centiseconds)
- LaneDistanceLeft (Real) (m)
- LaneDistanceRight (Real) (m)
- LaneHeading (Real) (degrees)
- LaneQualityRight [NOT USING AS ITS DESCRIPTION IS NOT SPECIFIED IN THE METADATA]
- LaneQualityLeft [[NOT USING AS ITS DESCRIPTION IS NOT SPECIFIED IN THE METADATA]

StatusValue

- PK Status
- Value

TargetTypeValue

- PK TargetType
- Value

Using Microsoft Visio

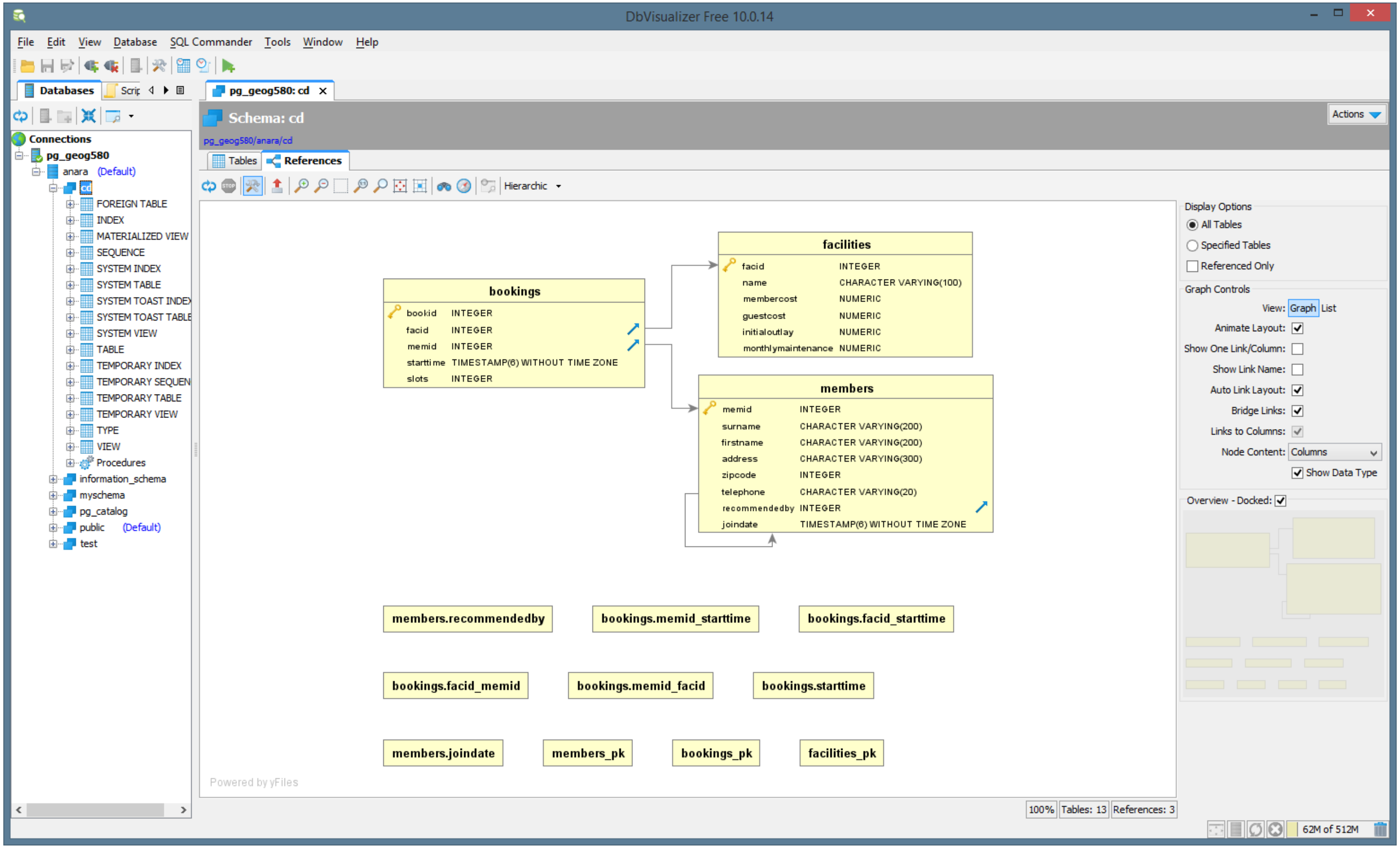
E-R Diagram

DB Design Tools

- □ https://wiki.postgresql.org/wiki/GUI_Database_Design_Tools

E-R Diagram

DbVisualizer (Free version 10.0.14)



facilities

facid	INTEGER
name	CHARACTER VARYING(100)
membercost	NUMERIC
guestcost	NUMERIC
initialoutlay	NUMERIC
monthlymaintenance	NUMERIC

bookings

bookid	INTEGER
facid	INTEGER
memid	INTEGER
starttime	TIMESTAMP(6) WITHOUT TIME ZONE
slots	INTEGER

members

memid	INTEGER
surname	CHARACTER VARYING(200)
firstname	CHARACTER VARYING(200)
address	CHARACTER VARYING(300)
zipcode	INTEGER
telephone	CHARACTER VARYING(20)
recommendedby	INTEGER
joindate	TIMESTAMP(6) WITHOUT TIME ZONE

members.recommendedby | bookings.memid_starttime | bookings.facid_starttime

bookings.facid_memid | bookings.memid_facid | bookings.starttime

members.joindate | members_pk | bookings_pk | facilities_pk

Display Options
- ● All Tables
- ○ Specified Tables
- □ Referenced Only

Graph Controls
- View: Graph List
- Animate Layout: ☑
- Show One Link/Column: □
- Show Link Name: □
- Auto Link Layout: ☑
- Bridge Links: ☑
- Links to Columns: ☑
- Node Content: Columns
- ☑ Show Data Type

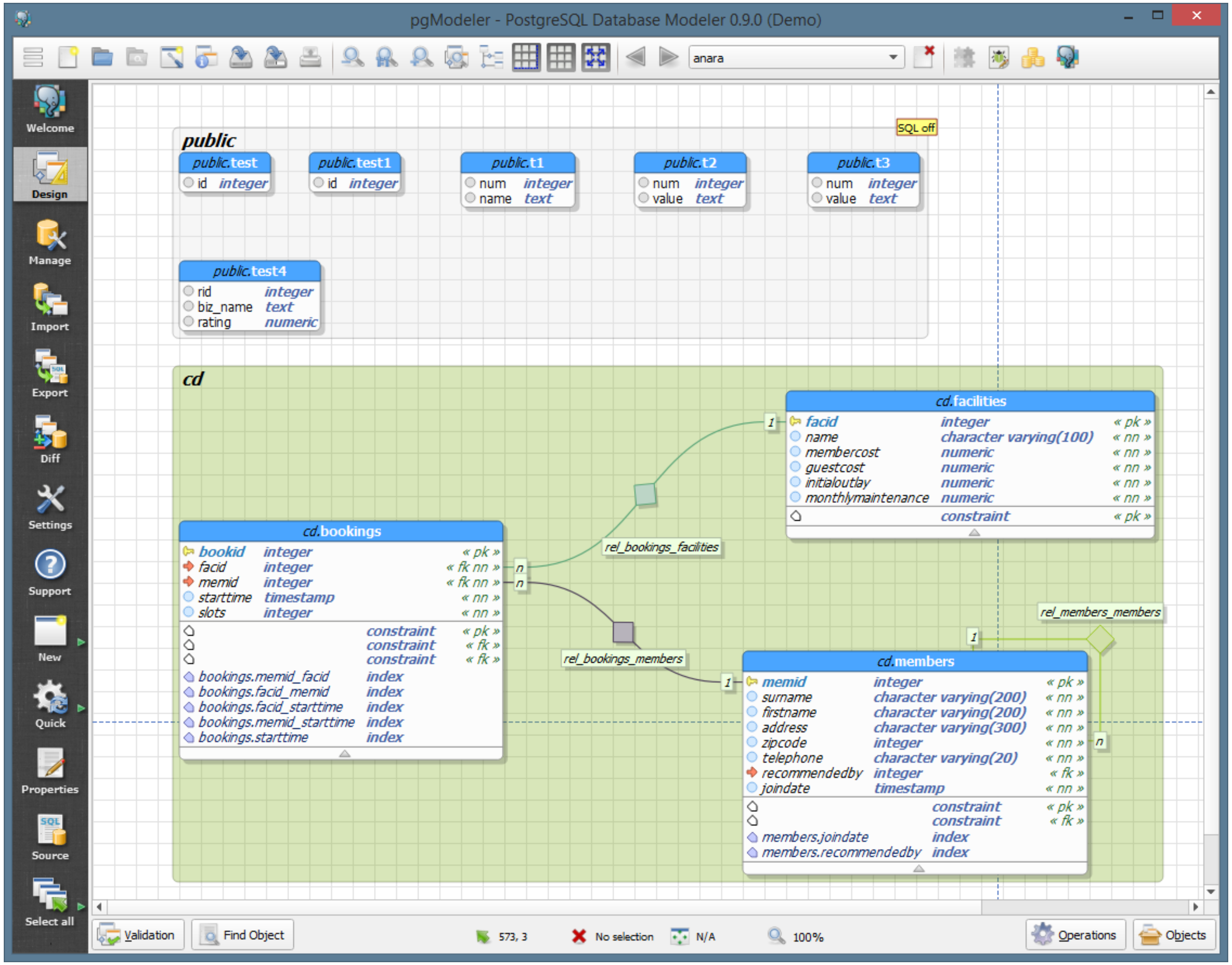
100% | Tables: 13 | References: 3

https://www.dbvis.com/

DbVis Software

E-R Diagram

DB Design Tools: pgModeler (demo ver. 0.9.0)



public.test

id	integer

public.test1

id	integer

public.t1

num	integer
name	text

public.t2

num	integer
value	text

public.t3

num	integer
value	text

public.test4

rid	integer
biz_name	text
rating	numeric

cd.facilities

facid	integer	« pk »
name	character varying(100)	« nn »
membercost	numeric	« nn »
guestcost	numeric	« nn »
initialoutlay	numeric	« nn »
monthlymaintenance	numeric	« nn »
	constraint	« pk »

cd.bookings

bookid	integer	« pk »
facid	integer	« fk nn »
memid	integer	« fk nn »
starttime	timestamp	« nn »
slots	integer	« nn »
	constraint	« pk »
	constraint	« fk »
	constraint	« fk »
bookings.memid_facid	index	
bookings.facid_memid	index	
bookings.facid_starttime	index	
bookings.memid_starttime	index	
bookings.starttime	index	

cd.members

memid	integer	« pk »
surname	character varying(200)	« nn »
firstname	character varying(200)	« nn »
address	character varying(300)	« nn »
zipcode	integer	« nn »
telephone	character varying(20)	« nn »
recommendedby	integer	« fk »
joindate	timestamp	« nn »
	constraint	« pk »
	constraint	« fk »
members.joindate	index	
members.recommendedby	index	

Production: $12.50 for 6 months!

https://www.pgmodeler.com.br/

pgModeler