Databases Design. Introduction to SQL

LECTURE 2

Conceptual Design

Database Design Stages

1. Subject Area Analysis
2. Conceptual Design
3. Logical Design
4. Physical Design

Conceptual Modeling

ER model (entity-relationship model) is a way of graphically representing the logical relationships of entities in order to create a database.

The ER model was first proposed by Peter Chen of Massachusetts Institute of Technology (MIT) in the 1970s.

To design an ER model you should know …

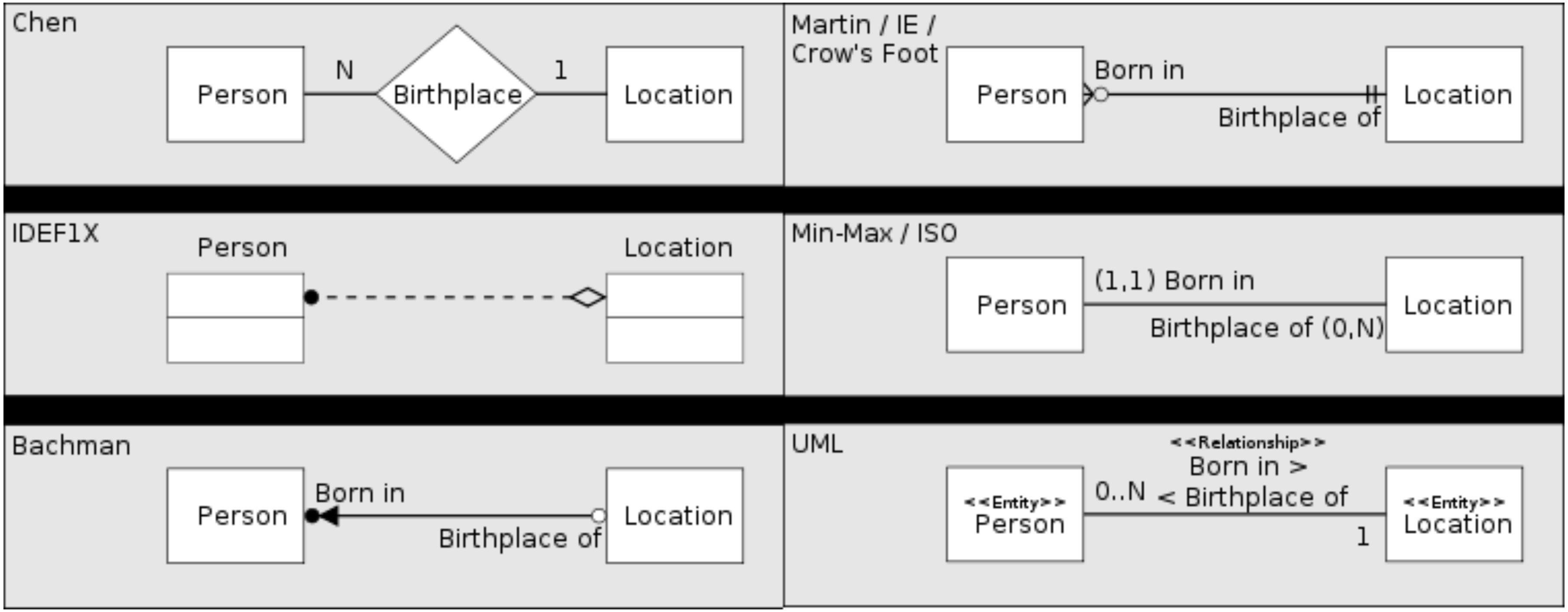
- Entities
- Attributes
- Relationships

Different types of Notations

- Chen’s Notation
- Bachman notation
- IDEF1X
- Martin notation (Crow’s foot)
- min, max-notation
- UML class diagram

Different types of Notations

Various methods of representing the same one to many relationship. In each case, the diagram shows the relationship between a person and a place of birth: each person must have been born at one, and only one, location, but each location may have had zero or more people born at it.



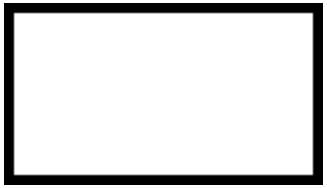
Example: University db

- Entities:
  Students
  Teachers
  Subjects

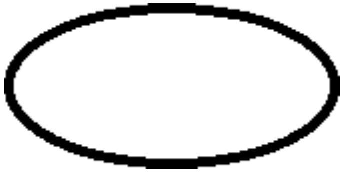
- Attributes
  Students (stud_id, name, email, group)
  Teachers (teach_id, name, email, department)
  Subjects (subject_id, name, credits)

First notation - Chen’s notation

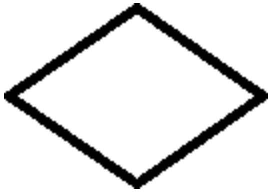
- Entity (rectangle shape)



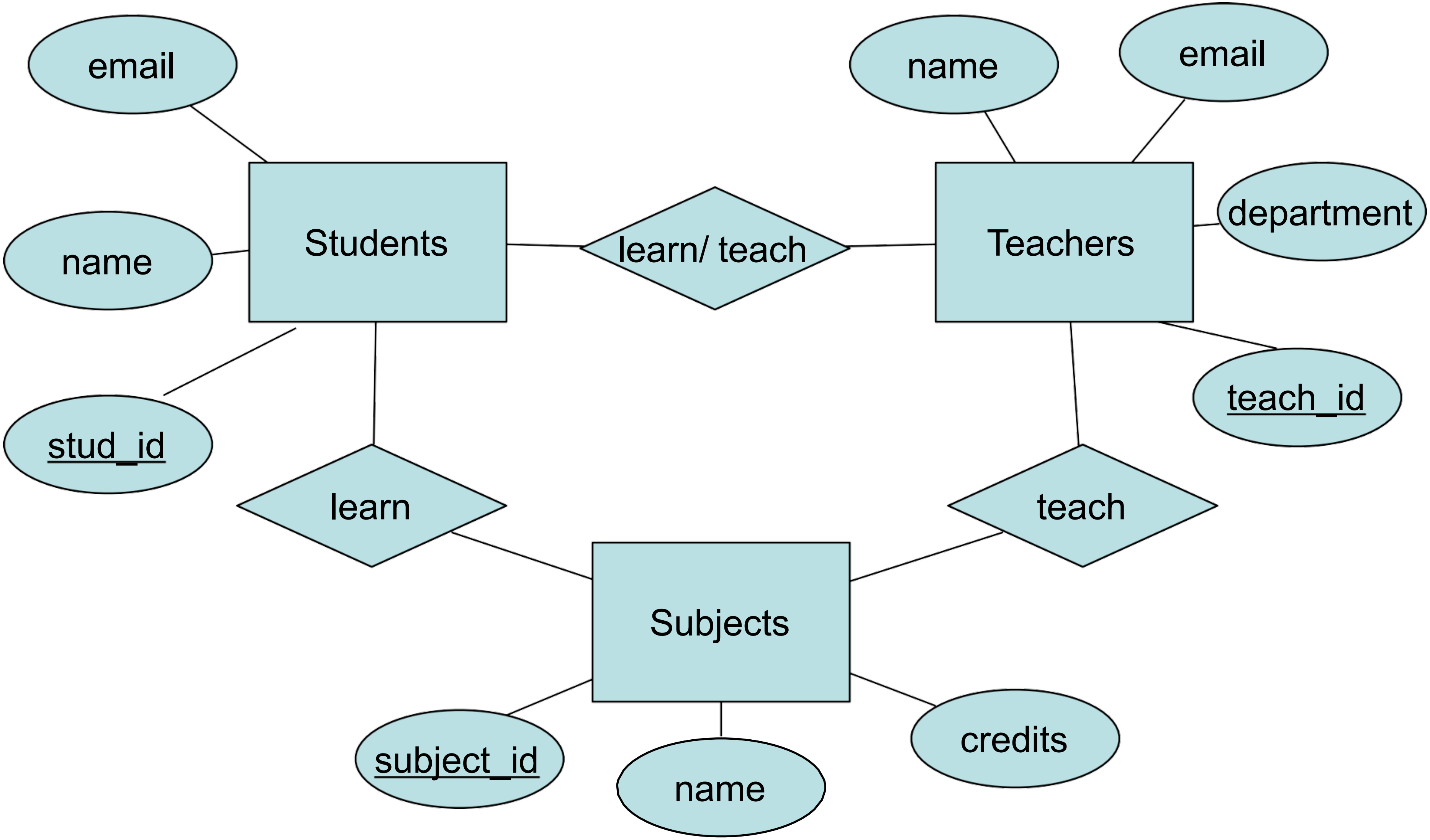
- Attribute (oval shape)



- Relationship (rhombus shape)



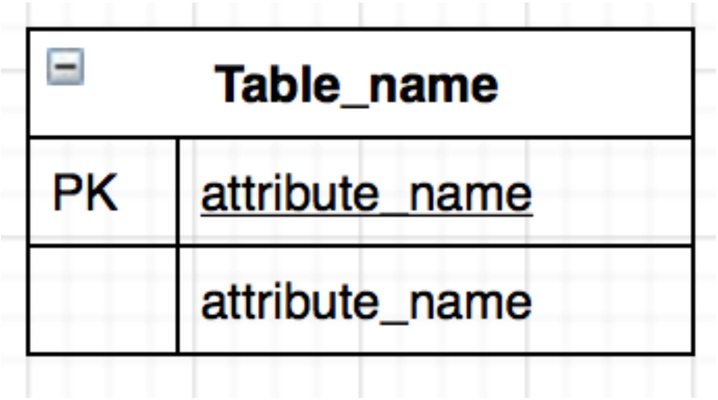
ER-diagram with Chen’s notation



Crow’s foot notation

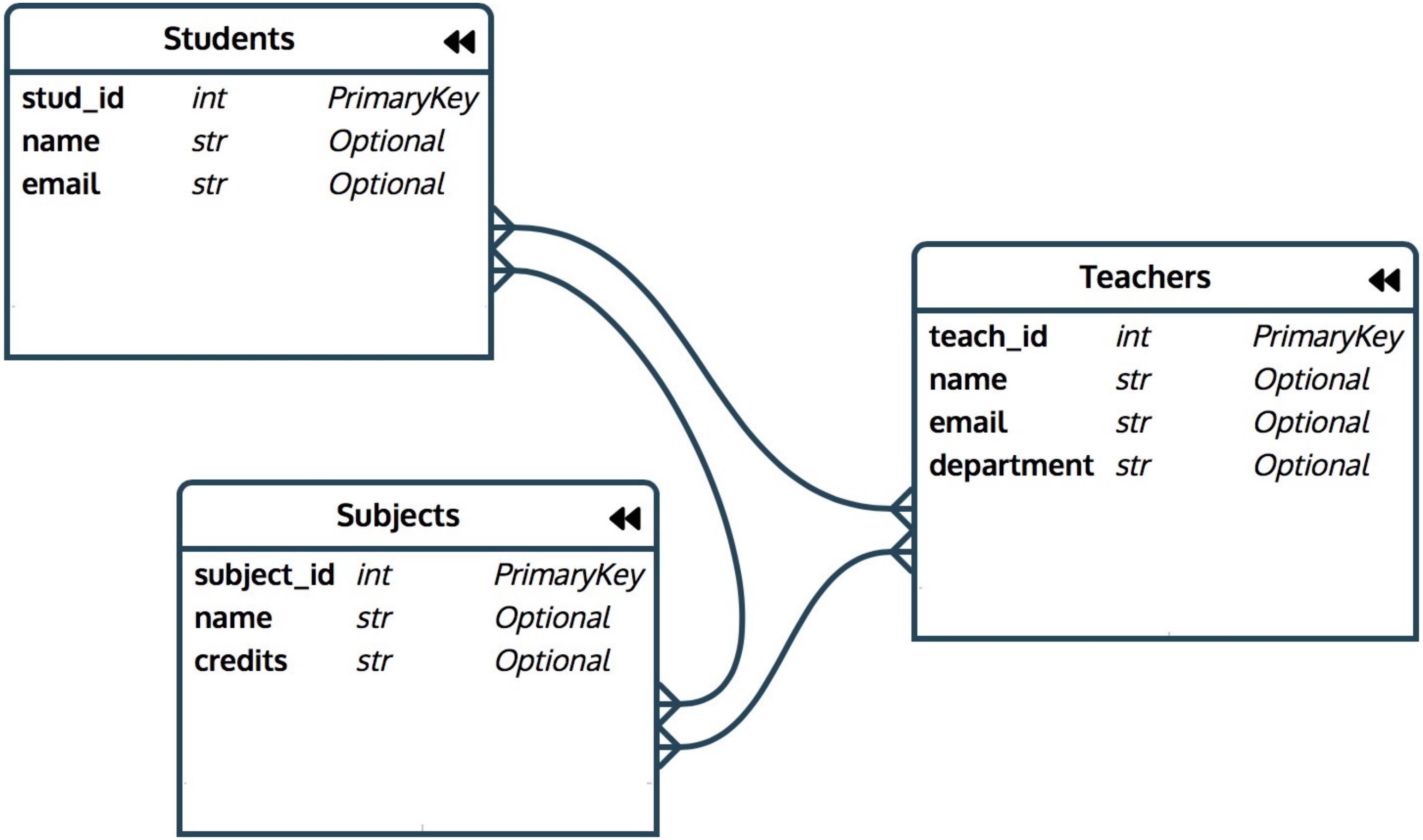
Crow's foot diagrams represent:

- entities as boxes;
- relationships as lines between the boxes;
- different shapes at the ends of these lines represent the relative cardinality of the relationship (the *dash* represents "one ", the *crow's foot* represents "many" or "infinite")



Table_name	
PK	attribute_name
	attribute_name

ER-diagram with Crow’s foot notation



Students

stud_id	*int*	*PrimaryKey*
name	*str*	*Optional*
email	*str*	*Optional*

Teachers

teach_id	*int*	*PrimaryKey*
name	*str*	*Optional*
email	*str*	*Optional*
department	*str*	*Optional*

Subjects

subject_id	*int*	*PrimaryKey*
name	*str*	*Optional*
credits	*str*	*Optional*

Relationships

Multiplicity is the number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship

Relationship types:

- one-to-one (1:1)
- one-to-many (1:*)
- many-to-many (*:*)

Foreign key

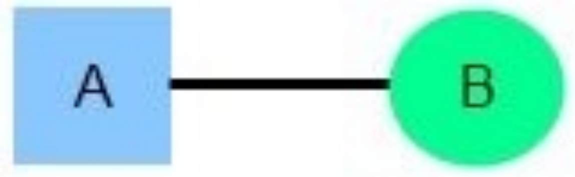
Foreign key is a key used to link two tables together.

Foreign key is an attribute in one table that refers to the Primary key in another table.

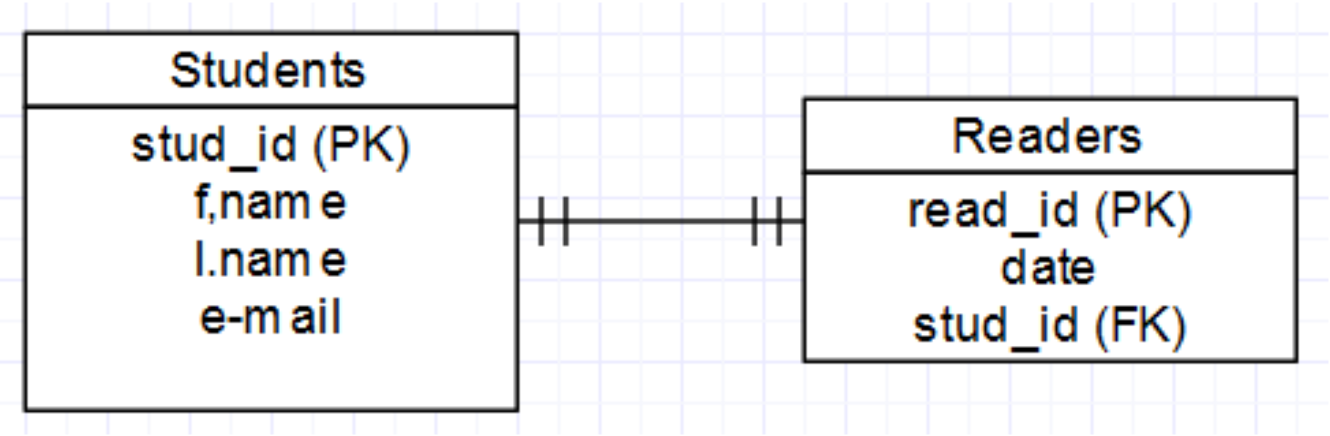
The table containing the foreign key is called the **child table**, and the table containing the Primary key is called the referenced or **parent table**.

One-to-one

One instance of an entity (A) is associated with one other instance of another entity (B).



Example of one-to-one

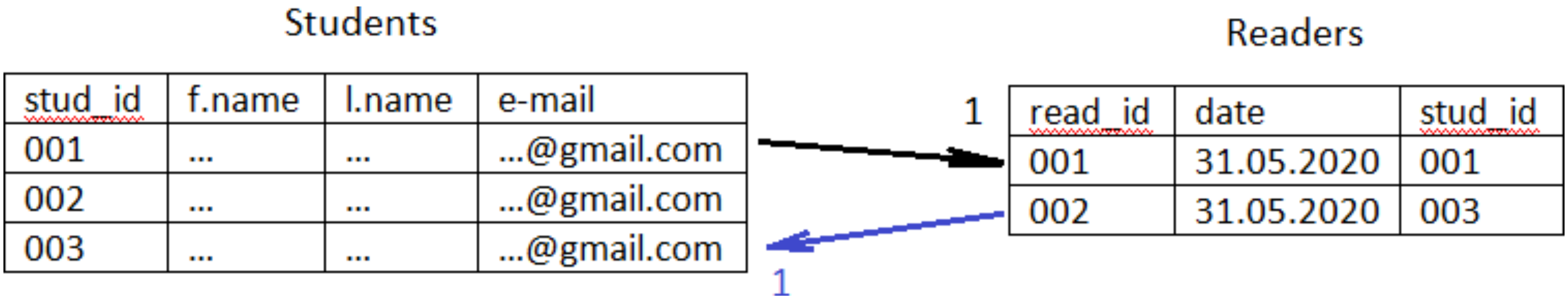


Students

stud_id (PK)
f,name
l.name
e-mail

Readers

read_id (PK)
date
stud_id (FK)



Students

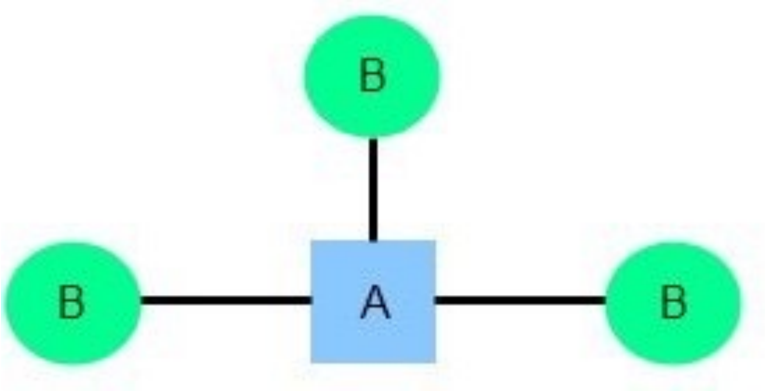
stud_id	f.name	l.name	e-mail
001	...	...	...@gmail.com
002	...	...	...@gmail.com
003	...	...	...@gmail.com

Readers

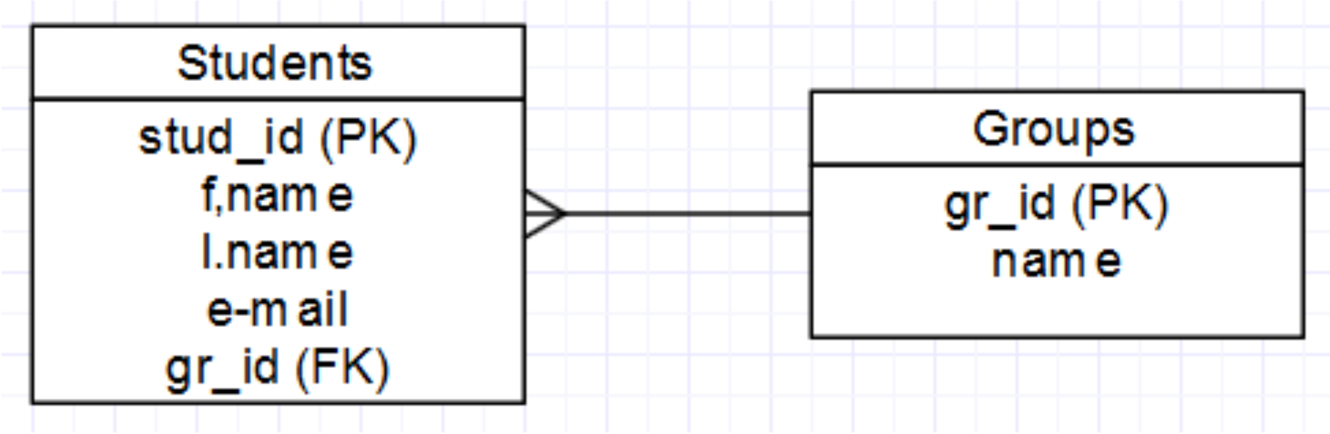
read_id	date	stud_id
001	31.05.2020	001
002	31.05.2020	003

One-to-many

One instance of an entity (A) is associated with one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A.



Example of one-to-many

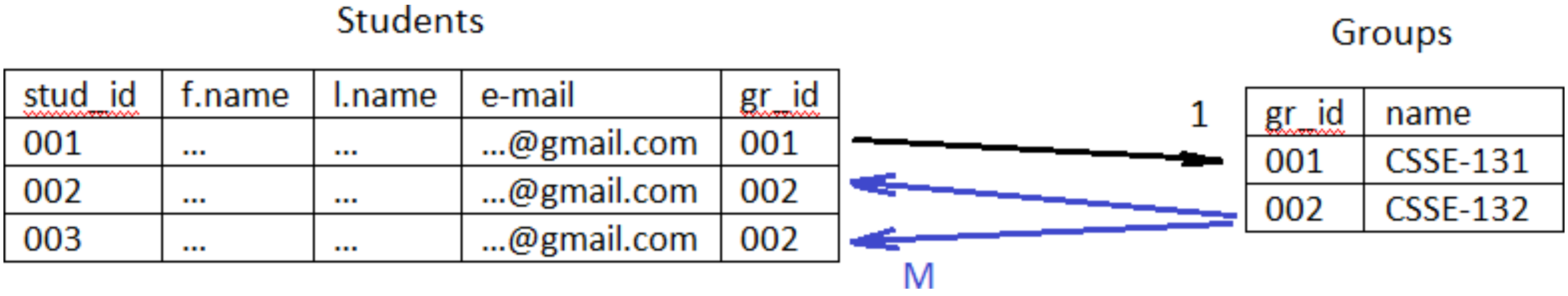


Students

stud_id (PK)
f,name
l.name
e-mail
gr_id (FK)

Groups

gr_id (PK)
name



Students

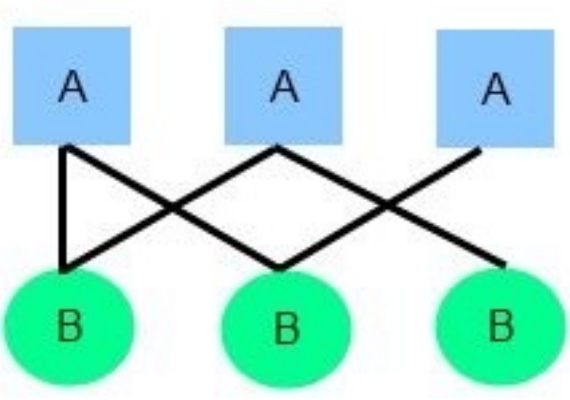
stud_id	f.name	l.name	e-mail	gr_id
001	...	...	...@gmail.com	001
002	...	...	...@gmail.com	002
003	...	...	...@gmail.com	002

Groups

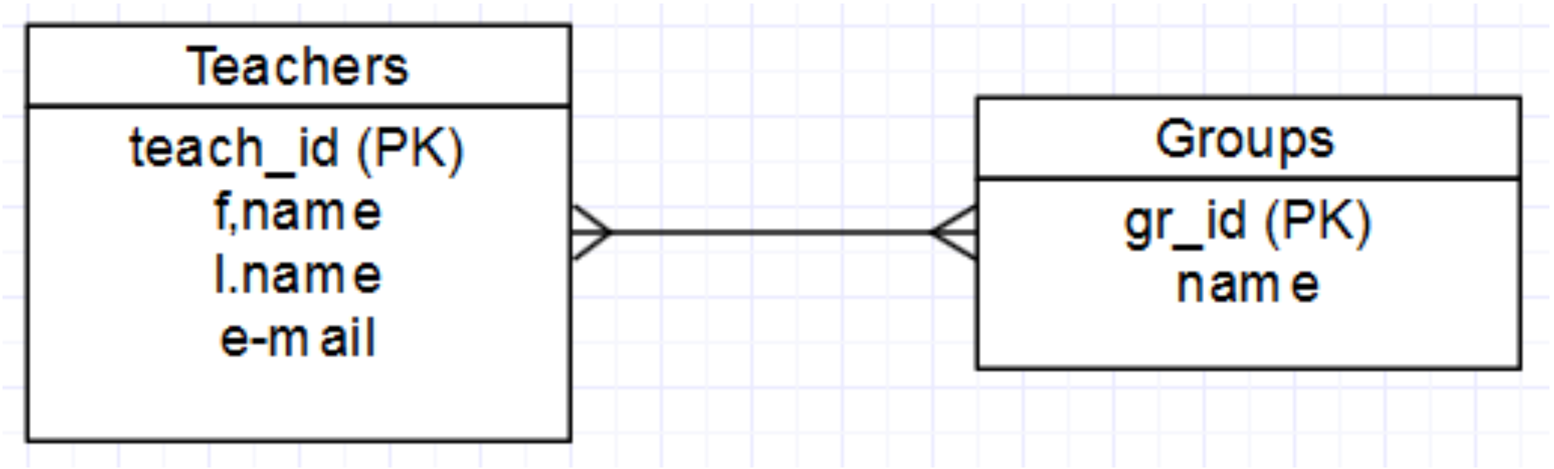
gr_id	name
001	CSSE-131
002	CSSE-132

Many-to-many

One instance of an entity (A) is associated with one or many instances of another entity (B), and one instance of entity B is associated with one or many instances of entity A.



Example of many-to-many

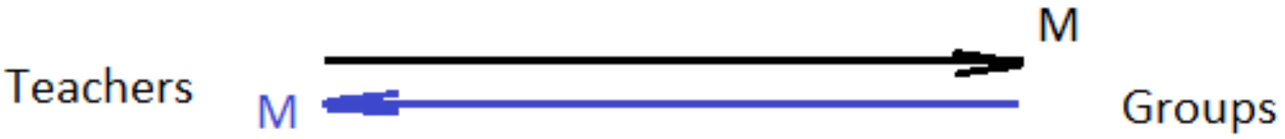


Teachers

teach_id (PK)
f,name
l.name
e-mail

Groups

gr_id (PK)
name



Teachers

M

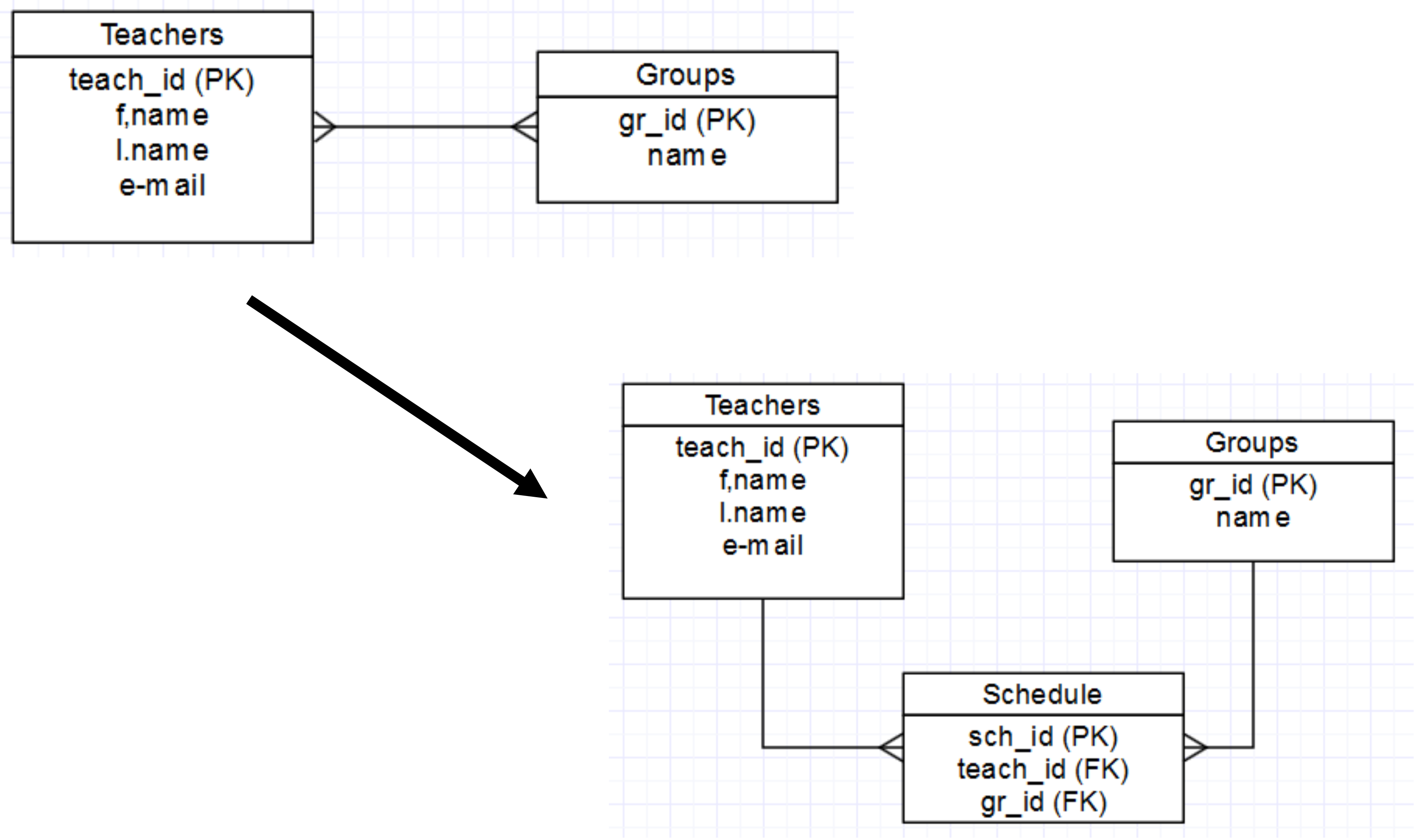
M

Groups

teach_id	f.name	l.name	e-mail
001	…	…	…@gmail.com
002	…	…	…@gmail.com
003	…	…	…@gmail.com

gr_id	name
001	CSSE-131
002	CSSE-132

Example of many-to-many



Teachers

teach_id (PK)
f,name
l.name
e-mail

Groups

gr_id (PK)
name

Teachers

teach_id (PK)
f,name
l.name
e-mail

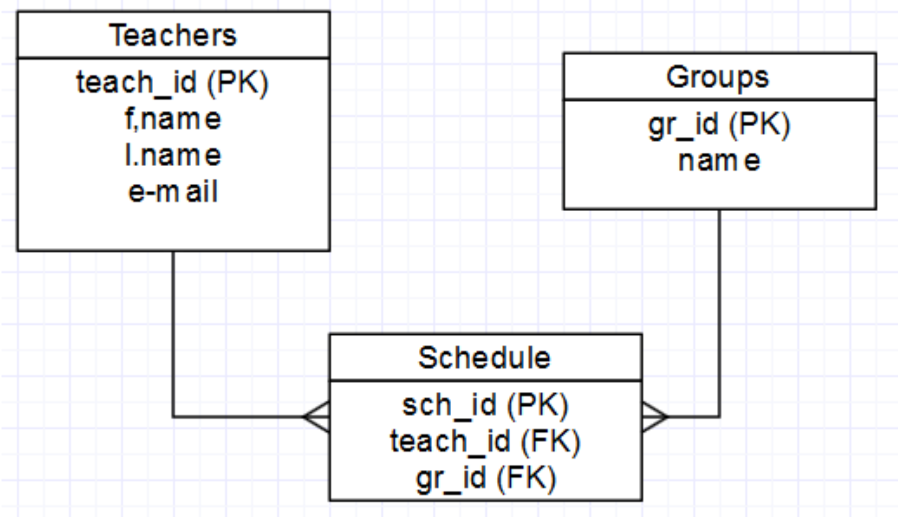
Groups

gr_id (PK)
name

Schedule

sch_id (PK)
teach_id (FK)
gr_id (FK)

Example of many-to-many



Teachers

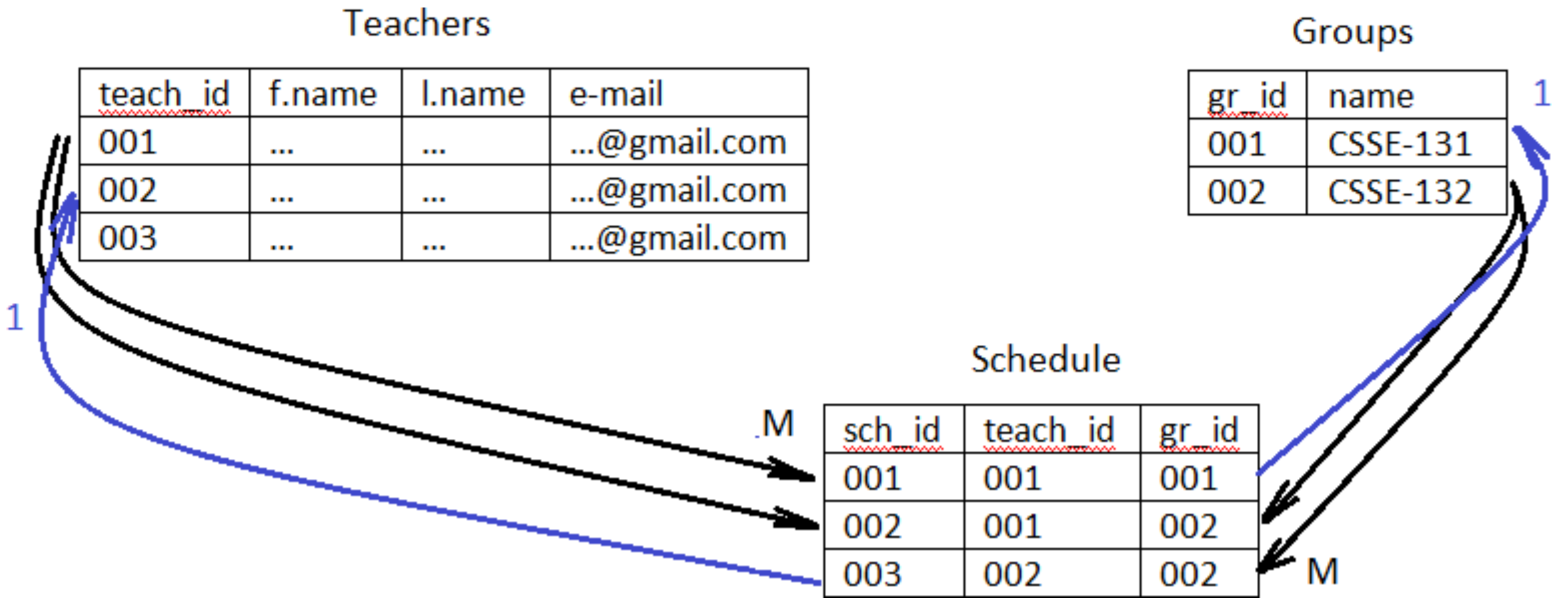
teach_id (PK)
f,name
l.name
e-mail

Groups

gr_id (PK)
name

Schedule

sch_id (PK)
teach_id (FK)
gr_id (FK)



Teachers

teach_id	f.name	l.name	e-mail
001	…	…	…@gmail.com
002	…	…	…@gmail.com
003	…	…	…@gmail.com

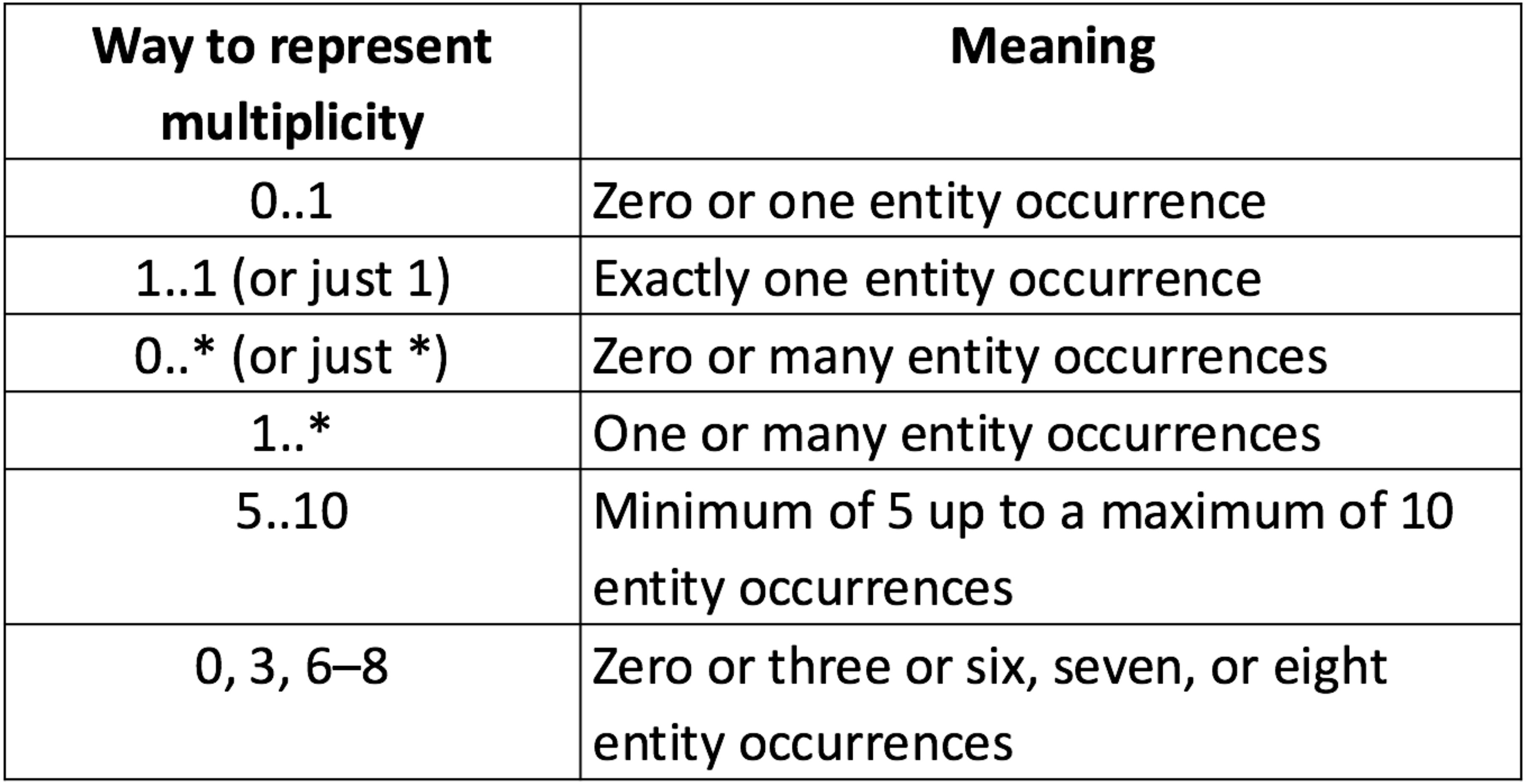
Groups

gr_id	name
001	CSSE-131
002	CSSE-132

Schedule

sch_id	teach_id	gr_id
001	001	001
002	001	002
003	002	002

Another representation ways



Way to represent multiplicity	Meaning
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero or three or six, seven, or eight entity occurrences

Crow’s foot relationships

Symbols are used to represent cardinality:

- the *ring* represents "zero"
- the *dash* represents "one"
- the *crow's foot* represents "many" or "infinite"

Sometimes these symbols are used in pairs. The inner component of the notation represents the minimum, and the outer component represents the maximum.

- *ring* and *dash* → **minimum zero, maximum one (optional)**
- *dash* and *dash* → **minimum one, maximum one (mandatory)**
- *ring* and *crow's foot* → **minimum zero, maximum many (optional)**
- *dash* and *crow's foot* → **minimum one, maximum many (mandatory)**

Tools

- Gliffy.com
- Lucidchart.com
- Creately.com
- Draw.io
- MS Visio
- Erwin
- etc.

Books

- **Connolly, Thomas M. Database Systems**: A Practical Approach to Design, Implementation, and Management / Thomas M. Connolly, Carolyn E. Begg.- United States of America: Pearson Education

- **Garcia-Molina, H. Database system**: The Complete Book / Hector Garcia-Molina.- United States of America: Pearson Prentice Hall

- **Sharma, N. Database Fundamentals**: A book for the community by the community / Neeraj Sharma, Liviu Perniu.- Canada