Entity Relationship Diagrams

DB Practice CSC 330 Spring 2022

README

- This is a notebook for learning and practicing modeling relational databases using Entity Relationship Diagrams, keys and bridge tables
- Though we won't get more deeply into database design, understanding relationships, you need to understand this to understand `JOIN` commands when querying relational databases
- The notebook also contains two practice exercises for you to complete on your own in a self-guided lab session.
- **Modeling** is an essential IT design skill. You can model:
  - Technical processes (e.g. with UML)
  - Human processes (e.g. with BPMN)
  - Algorithms (e.g. with pseudocode or flowcharts)
  - Entity relationships (e.g. with ERDs)

ERD notations

- Relational database design relies on table relationships
- This is especially important for `JOIN` operations
- There are two dominant notations for ERD[1]

Chen

- **Notation Example**

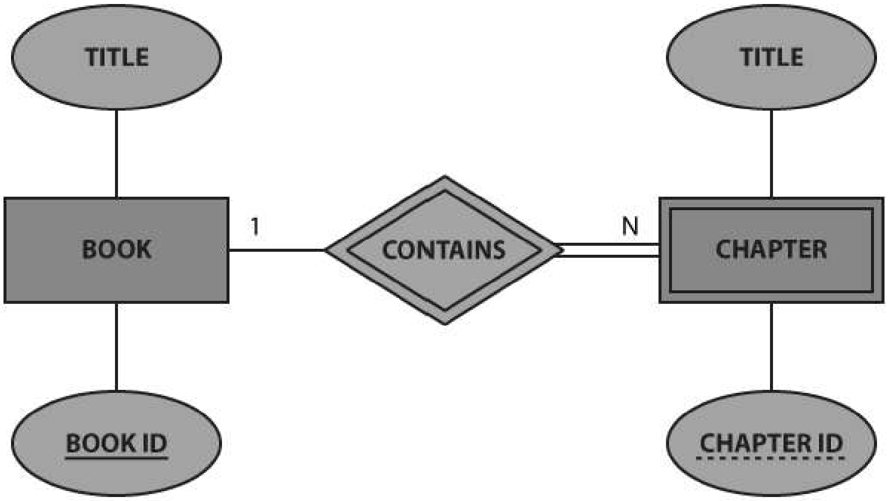


Figure 1: ERD Chen notation example (Source: Dybka, 2014)

- `BOOK` and `CHAPTER` are two entities (tables)
- A `BOOK` fully `CONTAINS` 1…N `CHAPTER` entities
- The `CHAPTER` is a fully dependent child of `BOOK`
- The `CHAPTER` totally participates in the `BOOK`
- The `BOOK` has attributes `TITLE` and `BOOK_ID`
- `BOOK_ID` is a *Primary Key* (`PK`) of `BOOK`
- The `CHAPTER` has attributes `TITLE` and `CHAPTER_ID`
- The `CHAPTER_ID` is a *Primary Key* (`PK`) of `CHAPTER`

Crow's foot--

- **Notation summary**
    - A box represents an **entity**, e.g. `book`
    - An entity has **attributes**, e.g. `book_id`, `book_title` etc.

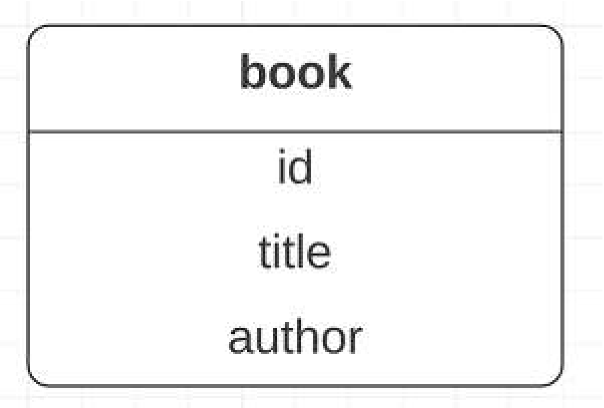


Figure 2: Entity with attributes

- The attributes can be overloaded with additional properties like *Primary Key*, *Foreign Key*, and they have types like *integer*, *text* etc.

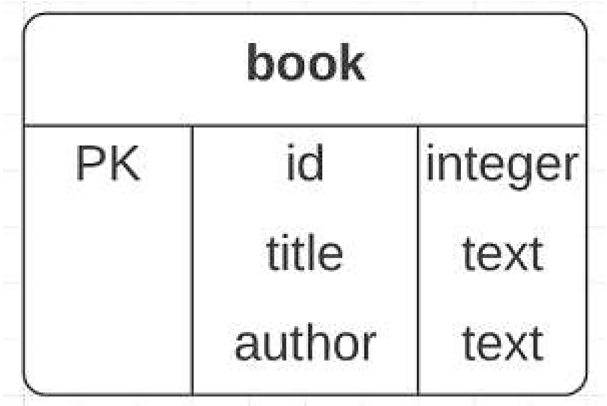


Figure 3: Entity with overloaded attributes

- Lines between entities represent a (binary) **relationship**
- Relationships have two indicators: **maximum** (aka multiplicity) and **minimum**

- Multiplicity = *maximum* number of associations between the entities

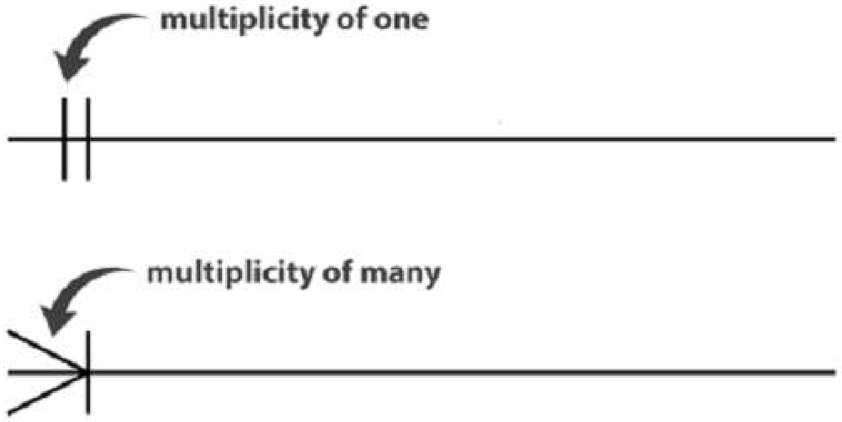


Figure 4: Multiplicity (Source: Dybka, 2016)

- Example 1: "A book has one and only one ISBN."

- Example 2: "A book has many chapters."

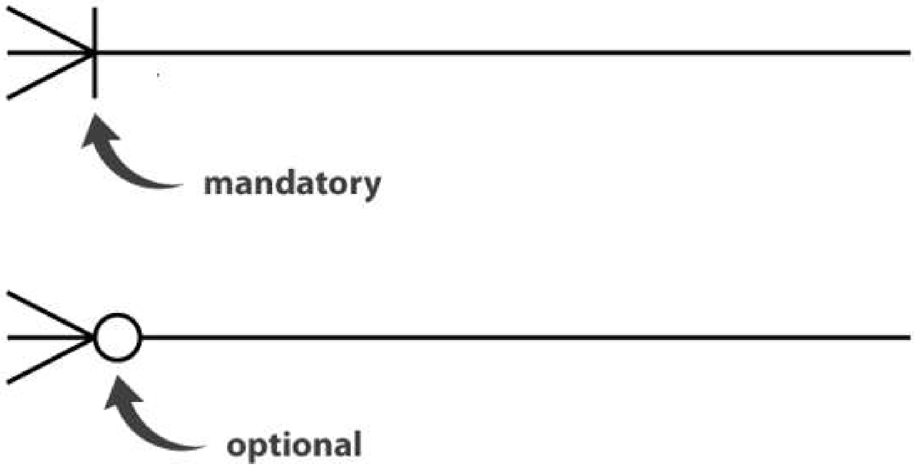
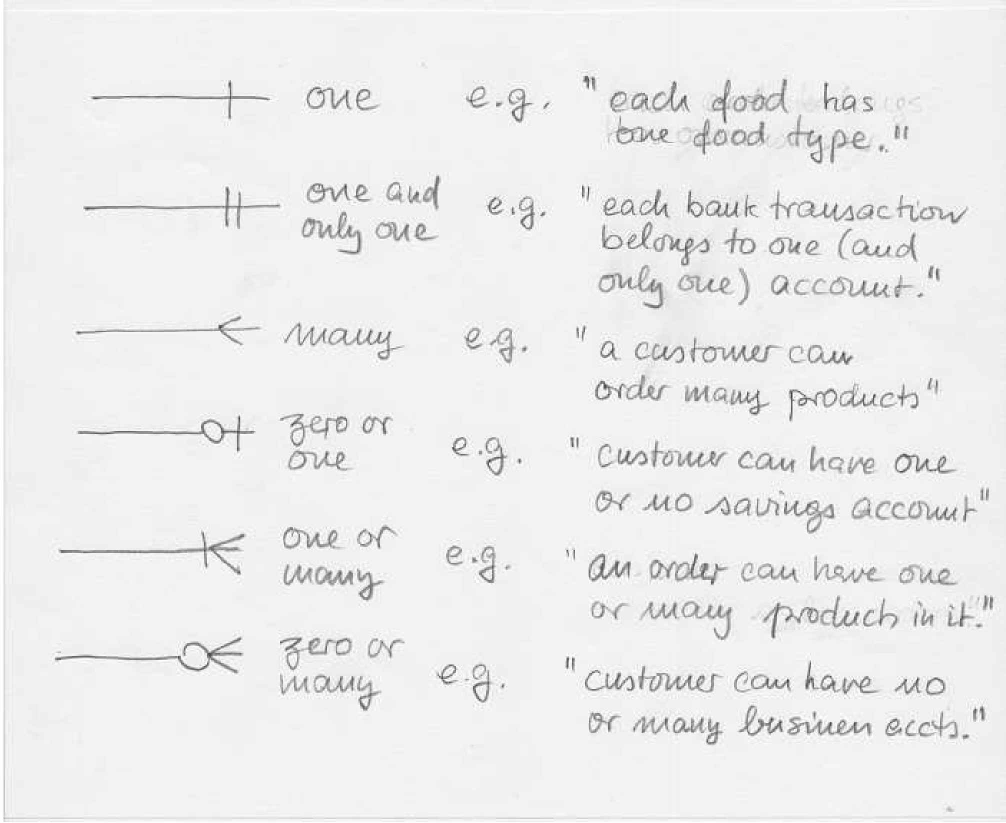


Figure 5: Minimum (Source: Dybka, 2016)

  - Example 3: "A book has at least one chapter."
- Example 4: "A book has no or many new editions."



one e.g. "each food has one food type."

one and only one e.g. "each bank transaction belongs to one (and only one) account."

many e.g. "a customer can order many products"

zero or one e.g. "customer can have one or no savings account"

one or many e.g. "an order can have one or many product in it."

zero or many e.g. "customers can have no or many business accts."

Figure 6: ERD crow's foot notation summary

- Realistic example: bank tables

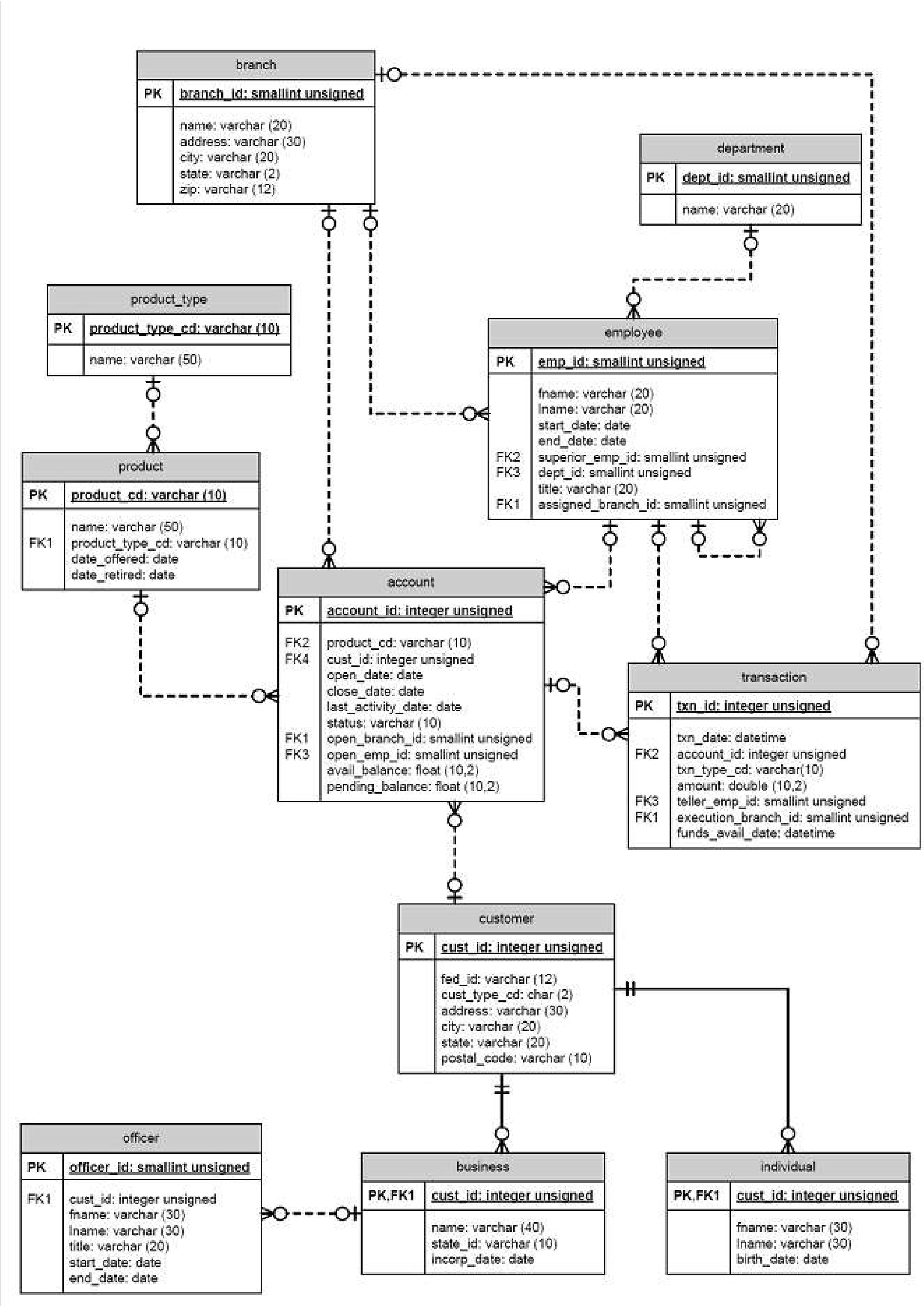

branch
PK branch_id: smallint unsigned
name: varchar (20)
address: varchar (30)
city: varchar (20)
state: varchar (2)
zip: varchar (12)
department
PK dept_id: smallint unsigned
name: varchar (20)
product_type
PK product_type_cd: varchar (10)
name: varchar (50)
employee
PK emp_id: smallint unsigned
fname: varchar (20)
lname: varchar (20)
start_date: date
end_date: date
FK2 superior_emp_id: smallint unsigned
FK3 dept_id: smallint unsigned
title: varchar (20)
FK1 assigned_branch_id: smallint unsigned
product
PK product_cd: varchar (10)
name: varchar (50)
FK1 product_type_cd: varchar (10)
date_offered: date
date_retired: date
account
PK account_id: integer unsigned
FK2 product_cd: varchar (10)
FK4 cust_id: integer unsigned
open_date: date
close_date: date
last_activity_date: date
status: varchar (10)
FK1 open_branch_id: smallint unsigned
FK3 open_emp_id: smallint unsigned
avail_balance: float (10,2)
pending_balance: float (10,2)
transaction
PK txn_id: integer unsigned
txn_date: datetime
FK2 account_id: integer unsigned
txn_type_cd: varchar(10)
amount: double (10,2)
FK3 teller_emp_id: smallint unsigned
FK1 execution_branch_id: smallint unsigned
funds_avail_date: datetime
customer
PK cust_id: integer unsigned
fed_id: varchar (12)
cust_type_cd: char (2)
address: varchar (30)
city: varchar (20)
state: varchar (20)
postal_code: varchar (10)
officer
PK officer_id: smallint unsigned
FK1 cust_id: integer unsigned
fname: varchar (30)
lname: varchar (30)
title: varchar (20)
start_date: date
end_date: date
business
PK,FK1 cust_id: integer unsigned
name: varchar (40)
state_id: varchar (10)
incorp_date: date
individual
PK,FK1 cust_id: integer unsigned
fname: varchar (30)
lname: varchar (30)
birth_date: date

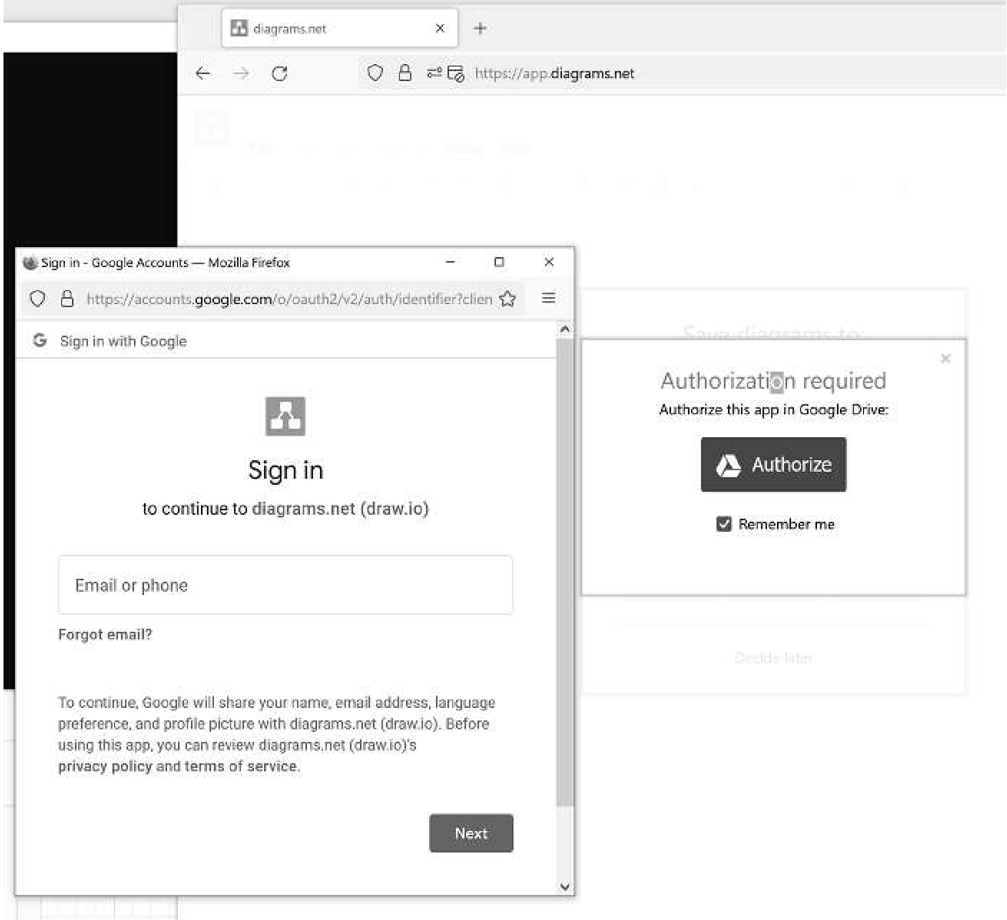
Figure 7: ERD of a bank (Source: Beaulieu, 2008)

Food database revisited

Create diagram with `draw.io`

- `[ ]` Open draw.io - you can choose where to save your diagrams
- `[ ]`

  If you choose Google Drive, you need to sign in to authorize as shown in the image.



Sign in

to continue to diagrams.net (draw.io)

To continue, Google will share your name, email address, language preference, and profile picture with diagrams.net (draw.io). Before using this app, you can review diagrams.net (draw.io)'s privacy policy and terms of service.

Authorization required

Authorize this app in Google Drive:

Figure 8: draw.io authorization dialog

- [ ] When you've authorized the storage place, you can create a new diagram. Don't bother with the templates.
- [ ] This short video shows how to create an entity and relationships between them using the crow's foot notation.

Create diagram

- [ ] *Zoom to Width* using the menu (left bottom of screen)
- [ ] Highlight and delete the diagrams on the screen
- [ ] Drag a table template onto the drawing board
- [ ]

  Draw the 4 entity diagrams for the food database

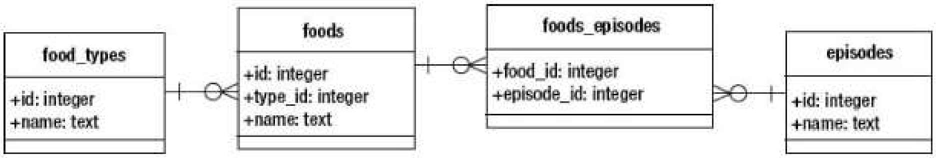


Figure 9: ERD of the food database (Source: Allen/Owens, 2010)

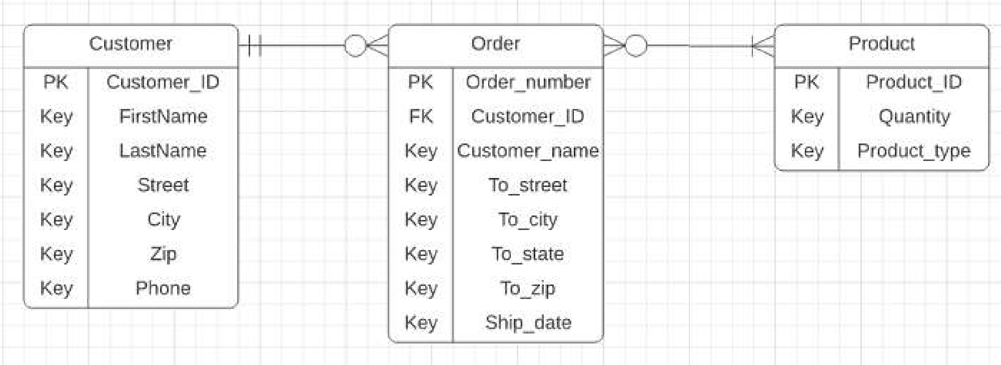
- [ ]

  Write down the relationships between `food_types` and `food`. Remember that you're not trying to describe all possible relationships but only the relationship for the purpose of this database with its narrow meaning: types of foods shown in episodes of the Seinfeld TV show.

  - `food_types` to `food`: each food type has zero or many instances of food in the show.
  - `food` to `food_types`: each food on the show is exactly one type of food.

Practice: customer orders

- [X] Develop an ER diagram with three tables: `customer`, `order`, and `product`
- [X] Identify suitable attributes
- [X] The diagram should allow for relationships like
  - "A customer submits an order"
  - "An order contains a product"
  - "An order belongs to a customer"
  - "A product is part of an order."
- [X] Use draw.io to draw the diagram
- [X] Draw relationships with the correct minimum/maximum
- [ ] Give your diagram a title
- [ ] Share the link to your solution with me via Email



Customer	
PK	Customer_ID
Key	FirstName
Key	LastName
Key	Street
Key	City
Key	Zip
Key	Phone

Order	
PK	Order_number
FK	Customer_ID
Key	Customer_name
Key	To_street
Key	To_city
Key	To_state
Key	To_zip
Key	Ship_date

Product	
PK	Product_ID
Key	Quantity
Key	Product_type

Figure 10: Customer orders products ERD

Practice exercise (due Thursday, 7 April 4 pm)

- [ ] Develop an ER diagram with relationships between **four** tables: `Student`, `Course`, `Lecturer`, and `Class` (or meeting).
- [ ] Identify suitable attributes: make sure that they are
  - measurable (you can think of a suitable data type)
  - atomic (`address` is composite, `street`, `street number` are atomic)
  - attributes (like `name`) and not entities (like `registrar`) or relationships (like `attendance`) themselves
- [ ] The diagram should allow for relationships like
  - "A student is enroled in a course."
  - "A class is attended by students."
  - "An lecturer offers a course."

    - "A class is taught by an lecturer."
    - etc.
- `[ ]` Use draw.io to draw the final ERD (see demo video)
- `[ ]` Draw relationships with the correct minimum/maximum
- `[ ]` Put `ERD assignment by [Your name] - Pledged` in the title of your diagram
- `[ ]` Upload a screenshot of your solution to Schoology

References

- Birkenkrahe (April 5, 2022). Drawing ERD in draw.io [video]. URL: youtu.be/gCranxLqZDI.
- Dybka (August 2, 2014). Chen Notation [blog]. URL: vertabelo.com.
- Dybka (August 31, 2016). Crow's Foot Notation [blog]. URL: vertabelo.com.
- Lucidchart (2017). Entity Relationship Diagram (ERD) Tutorial Part 1 [video]. URL: youtu.be/QpdhBUYk7Kk.

Footnotes:

[1] Notation reflects priorities, e.g. readibility vs. detail. It is surprising that there aren't more popular notations! Notation must faithfully represent the modeling standard. You can in fact become famous with notation - Feynman diagrams are an example: they are a diagrammatic language for complicated integrals that represent elementary particle interactions.

Author: Marcus Birkenkrahe

Created: 2022-04-05 Tue 18:19