

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 ERD¹

Chen

- **Notation Example**

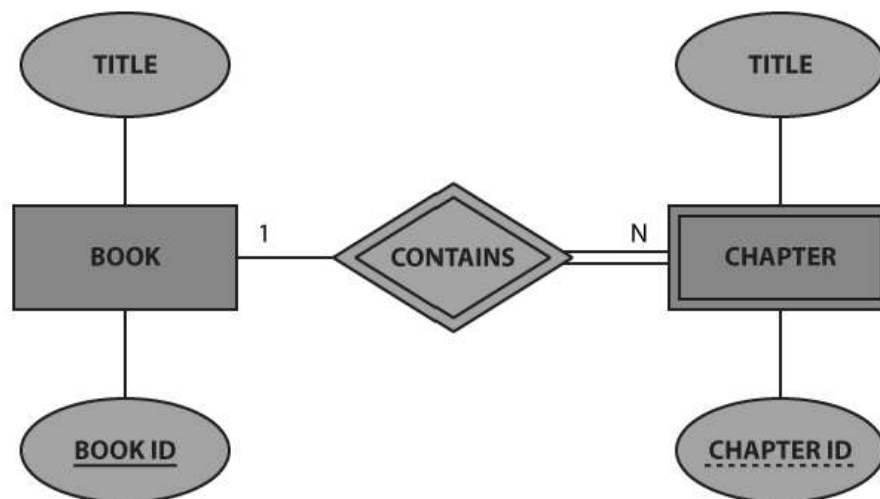


Figure 1: ERD Chen notation example (Source: Dybka, 2014)

- 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.

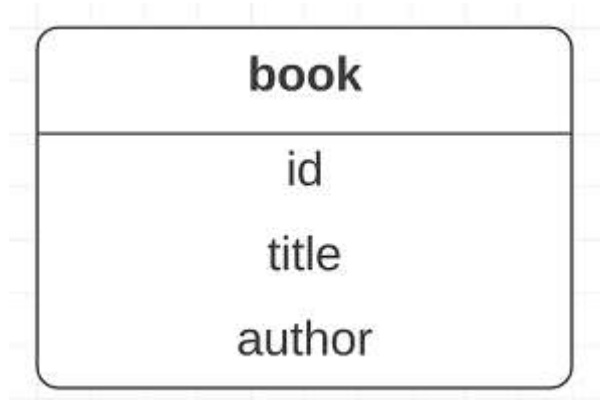


Figure 2: Entity with attributes

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

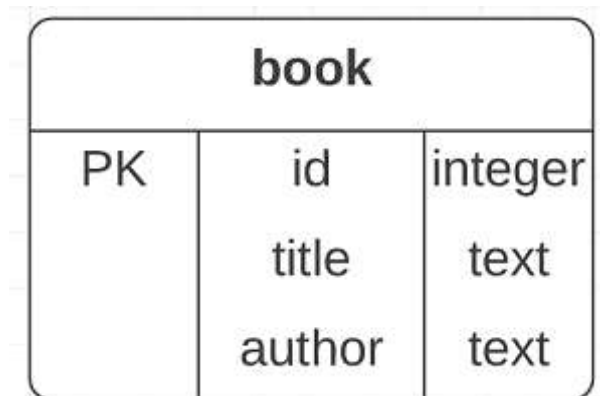


Figure 3: Entity with overloaded attributes

- Lines between entities represent a (binary) **relationship**
- Relationships have two indicators: **maximum** (aka multiplicity) and **minimum**
- Multiplicity = *maximum* number of associations between the entities

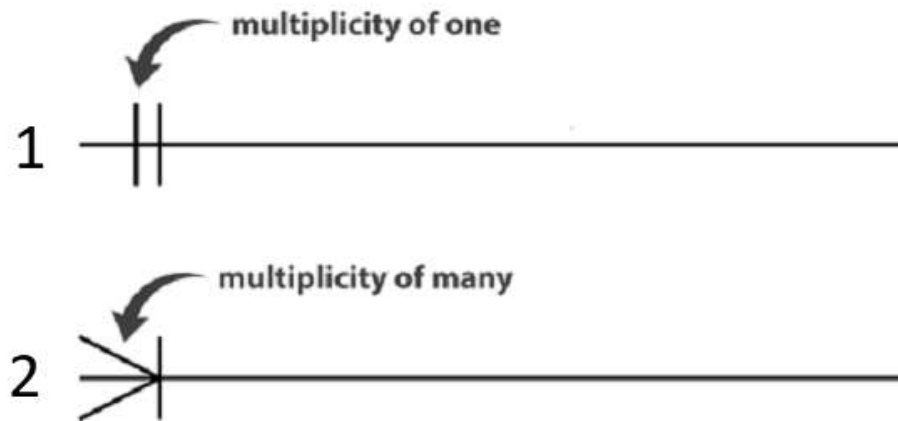


Figure 4: Multiplicity (Source: Dybka, 2016)

- 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."

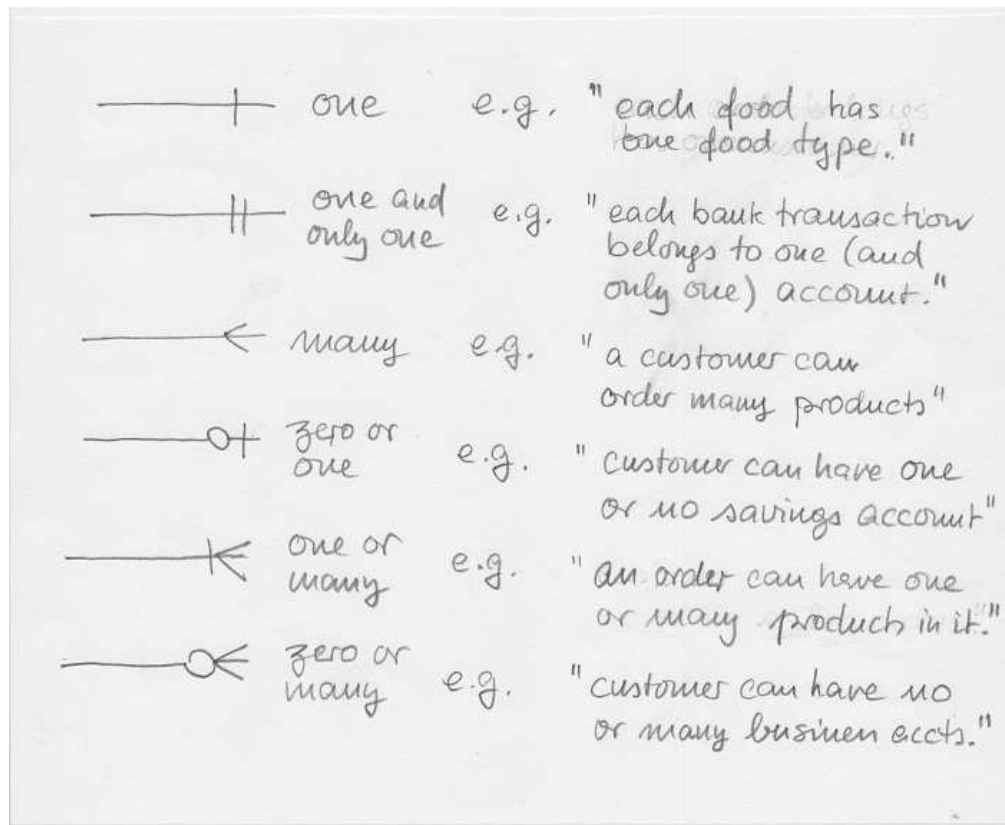


Figure 6: ERD crow's foot notation summary

- Realistic example: bank tables

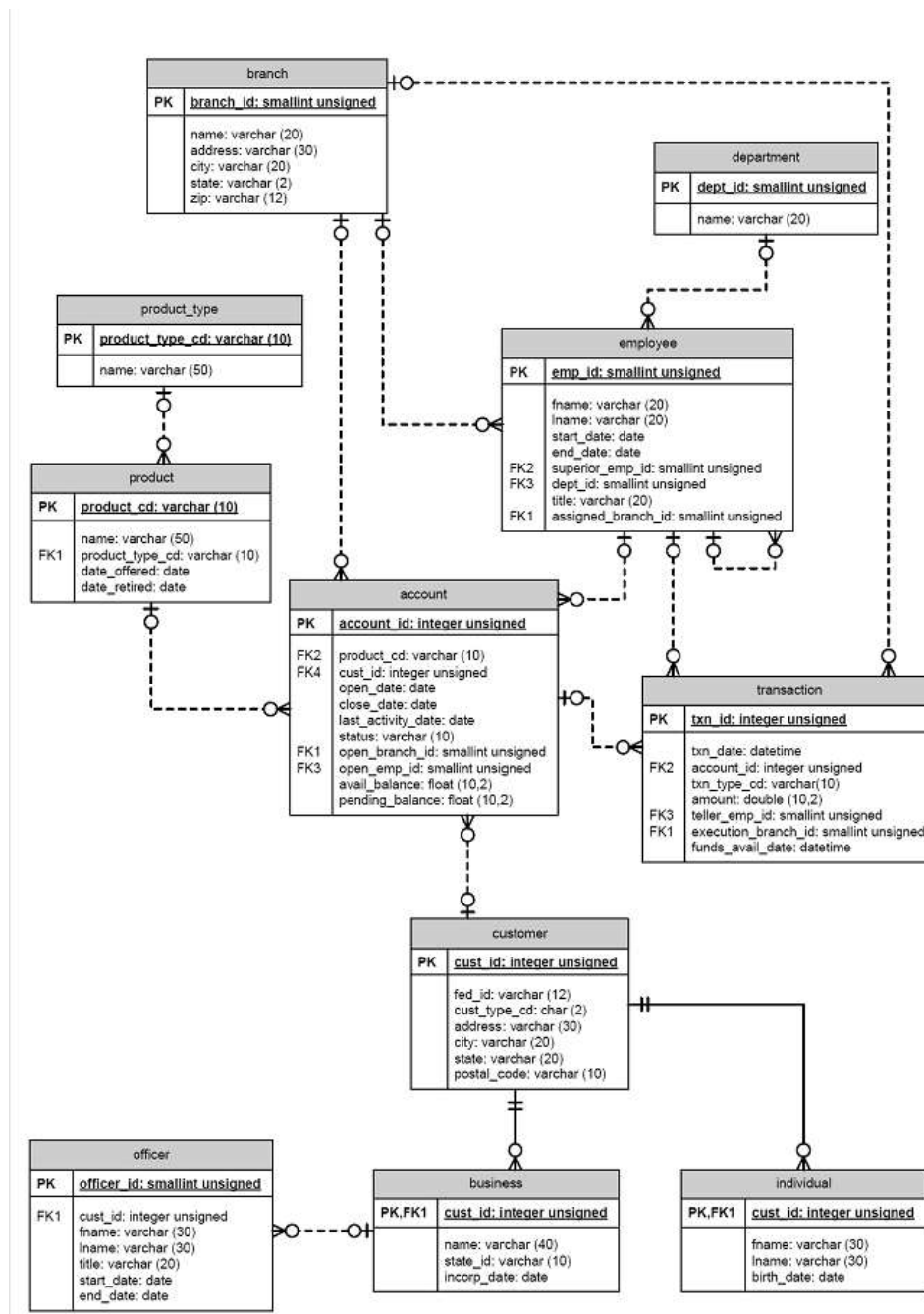


Figure 7: ERD of a bank (Source: Beaulieu, 2008)

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.

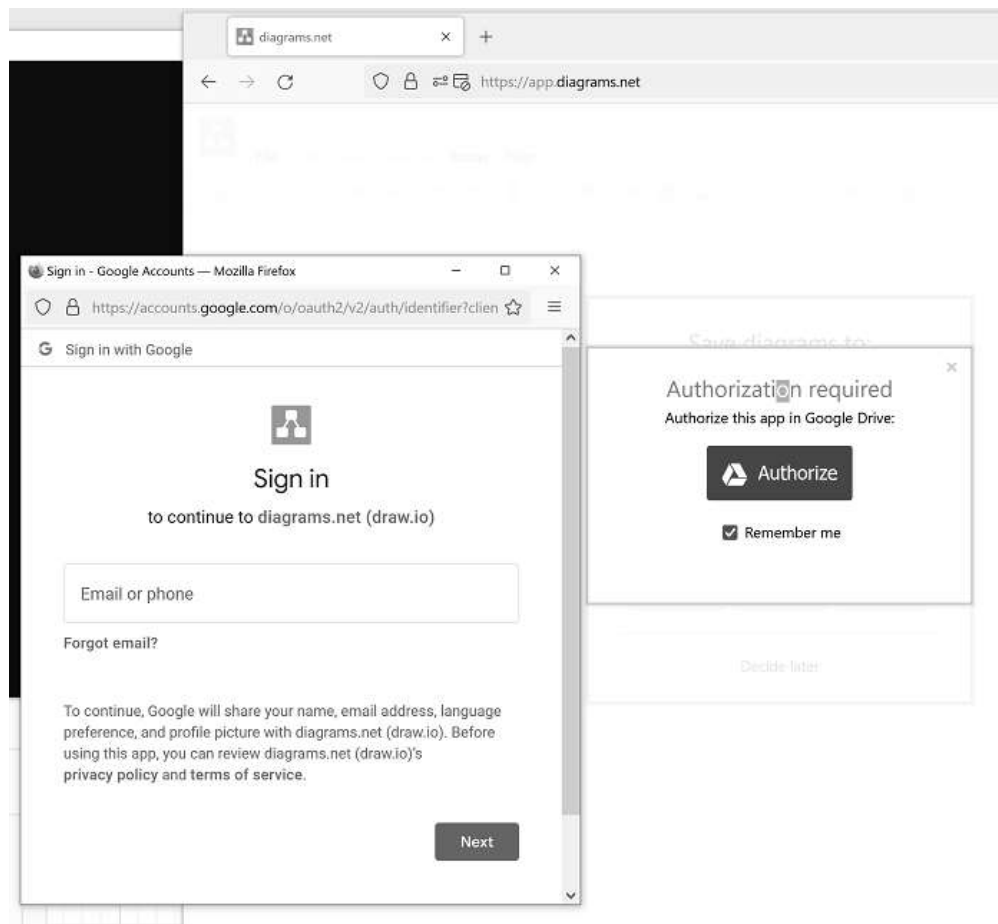


Figure 8: draw.io authorization dialog

- [] 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

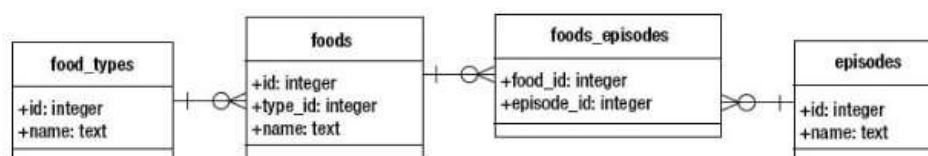


Figure 9: ERD of the food database (Source: Allen/Owens, 2010)

- []

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

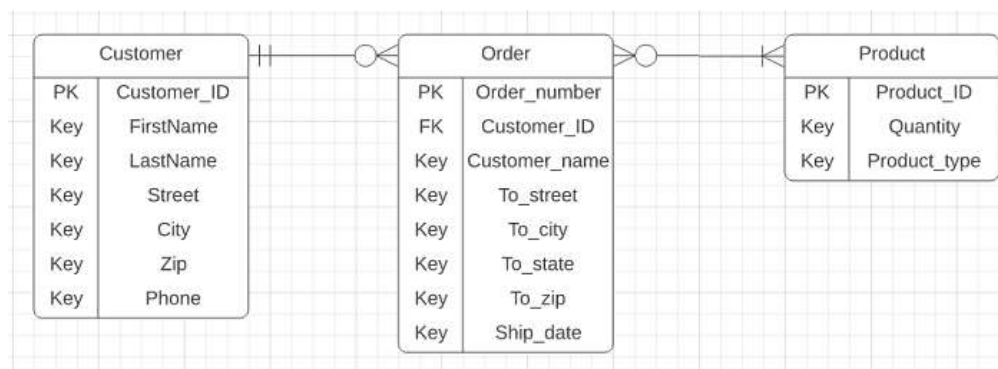


Figure 10: Customer orders products ERD

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 enrolled 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. readability 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 18:25