Schema Design

schema, n. – a representation of a plan or theory in the form of an outline or model.

Data Modeling

- How do we represent real world relationships and properties in our program?
  - …in a way that makes writing the program easy
  - …while remaining flexible for future changes
  - …oh, it also has to be fast (enough).

Designing a Schema

What we'll focus on today

- Analysis
  - What does my program need to output?
  - What data will I need to produce that output?
- Conceptual Design
  - Conceptual entities and their relationships
- Logical Design
  - In an SQL database: What are my tables, keys, and relationships?
  - In a program: What are my functions and data structures?
- Physical Design
  - Javascript code, CREATE TABLE statements

Designing a Schema

- Analysis
  - What does my program need to output?
  - What data will I need to produce that output?

Example: A Journal
Analysis

- I want a program to keep my journal in.
- I want to be able to enter the text of each journal entry.
- I want to be able to see journal entries chronologically.

Designing a Schema

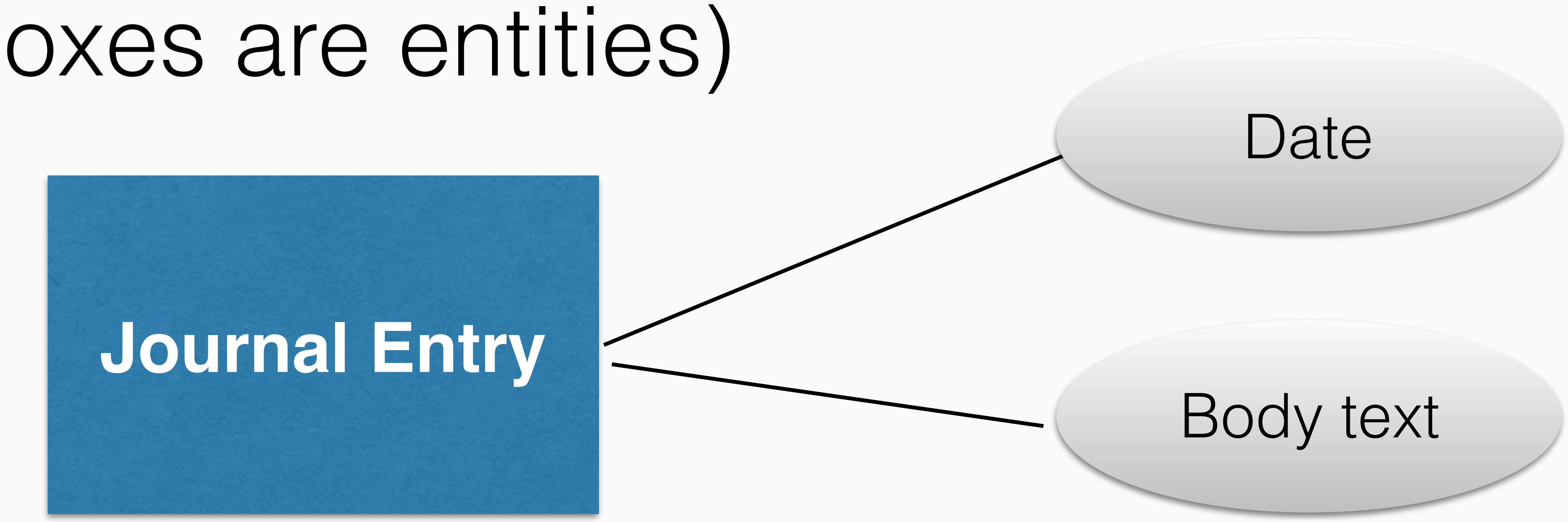
- Conceptual Design
  - Conceptual entities
  - Their relationships

Example: A Journal

Conceptual Design

(circles are properties)

(boxes are entities)

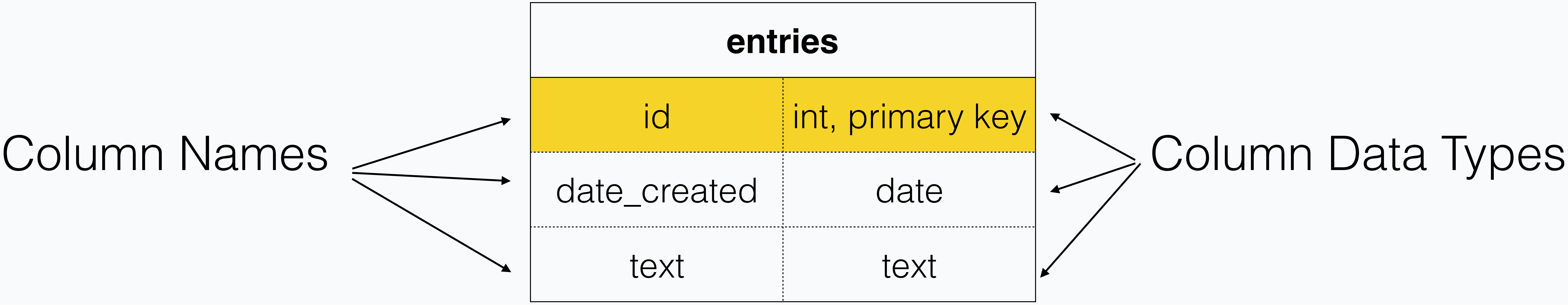


Designing a Schema

- Logical Design
  - In an SQL database: What are my tables, keys, and relationships?
  - In a program: What are my functions and data structures?

Example: A Journal

Logical Design



Column Names

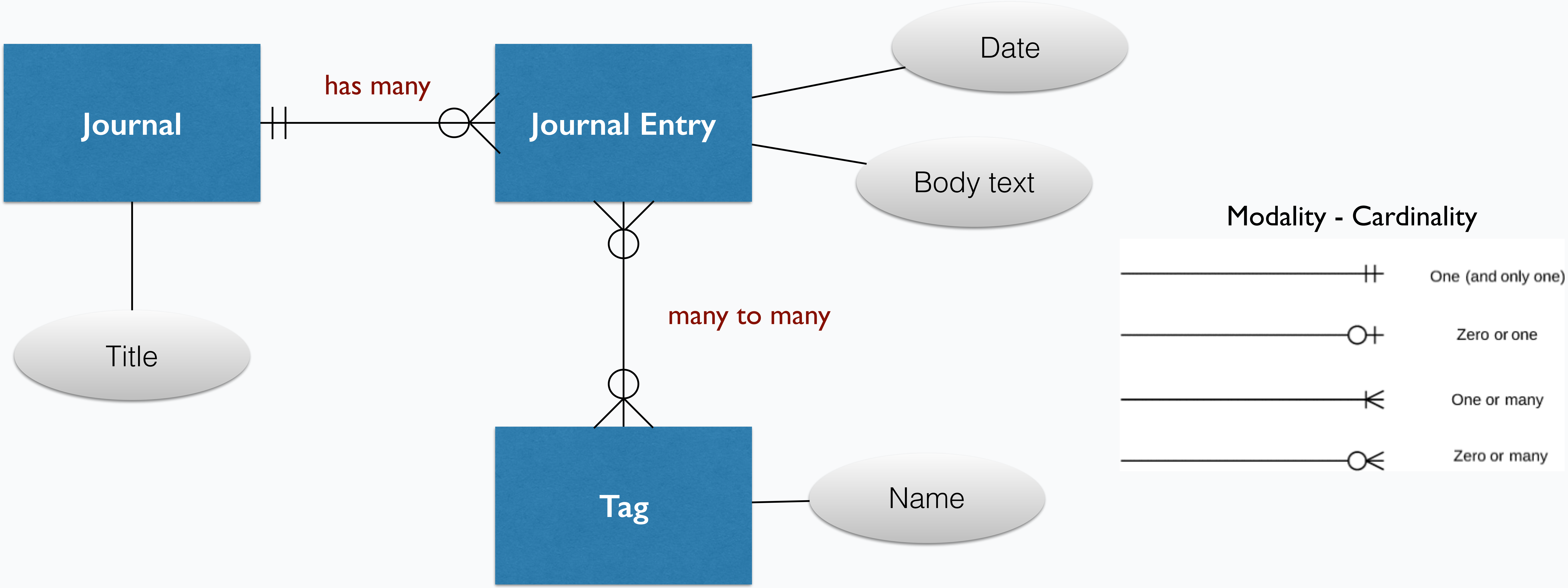
entries	
id	int, primary key
date_created	date
text	text

Column Data Types

All done??

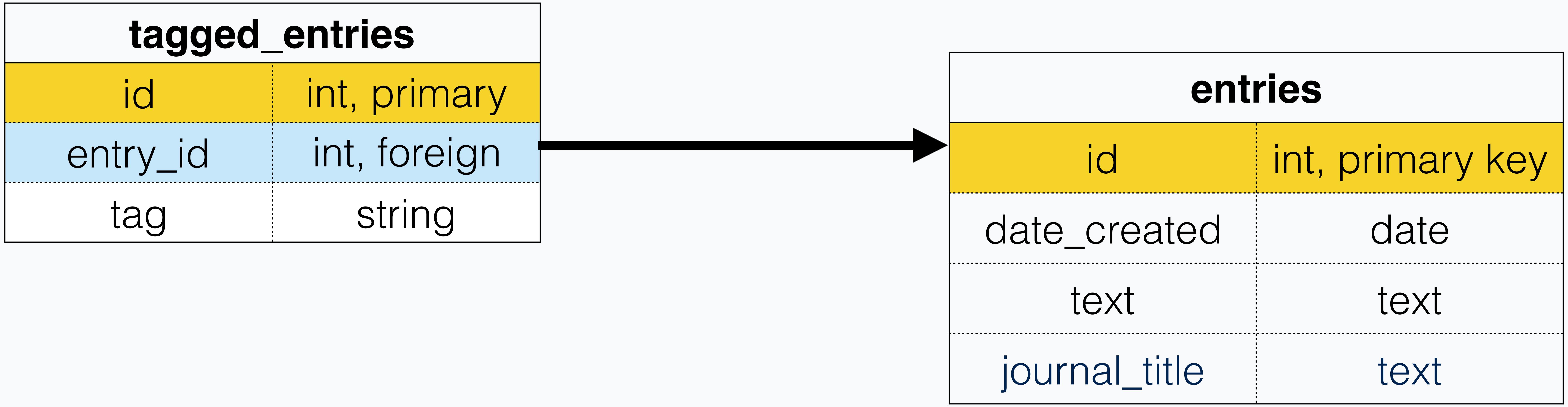
- **New requirements**
  - I want to be able to have multiple journals
  - I want to be able to #tag entries and find all entries with a particular #tag
- Let’s do it again!
  - Analysis –> Conceptual Design –> Logical Design

Example: A Journal
Conceptual Design, Take 2



Example: A Journal

Logical Design, Take 2?



tagged_entries	
id	int, primary
entry_id	int, foreign
tag	string

entries	
id	int, primary key
date_created	date
text	text
journal_title	text

But Wait!!!
Redundancy

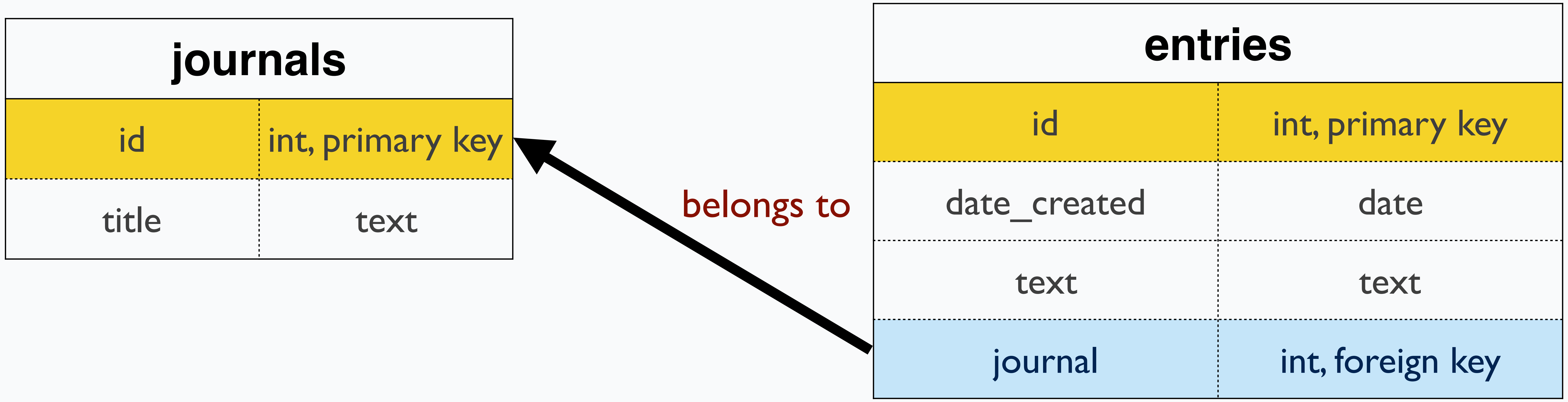
- Database organization that minimizes data redundancy and improves data integrity
- How do I change the name of “happy times” to “time that passes”?

select * from entries;

id	date_created	text	journal_title
0	2017-01-01	I am happy	happy times
1	2017-01-02	I am very happy	happy times
2	2017-01-18	Despair fills me	happy times
3	2017-01-19	Sadness is my life	an anatomy of pain

Normalization

- Focuses on optimal storage
- Eliminate repeating groups in individual tables
- Create a separate table for each set of related data
- Identify each set of related data with a primary key

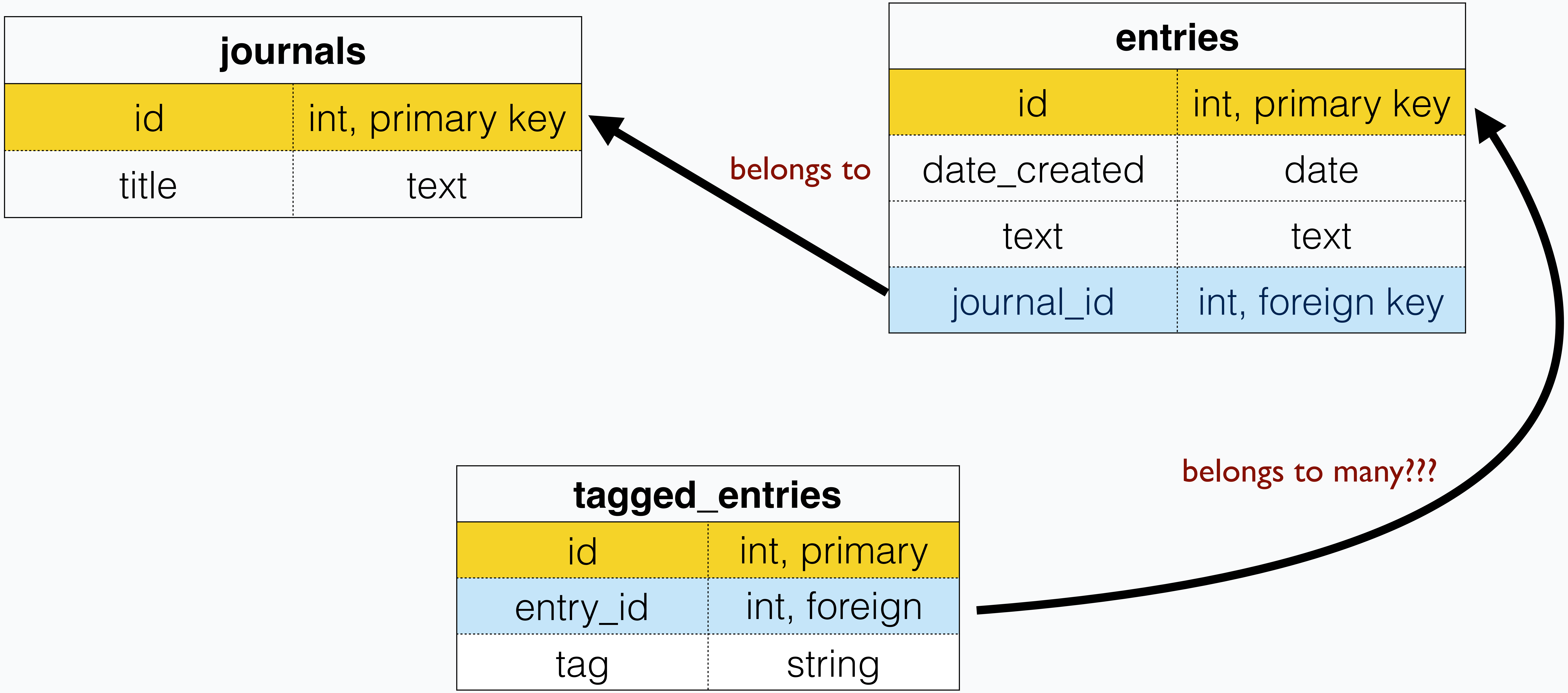


journals	
id	int, primary key
title	text

belongs to

entries	
id	int, primary key
date_created	date
text	text
journal	int, foreign key

Example: A Journal
Am I Normalized?



journals

id	int, primary key
title	text

entries

id	int, primary key
date_created	date
text	text
journal_id	int, foreign key

tagged_entries

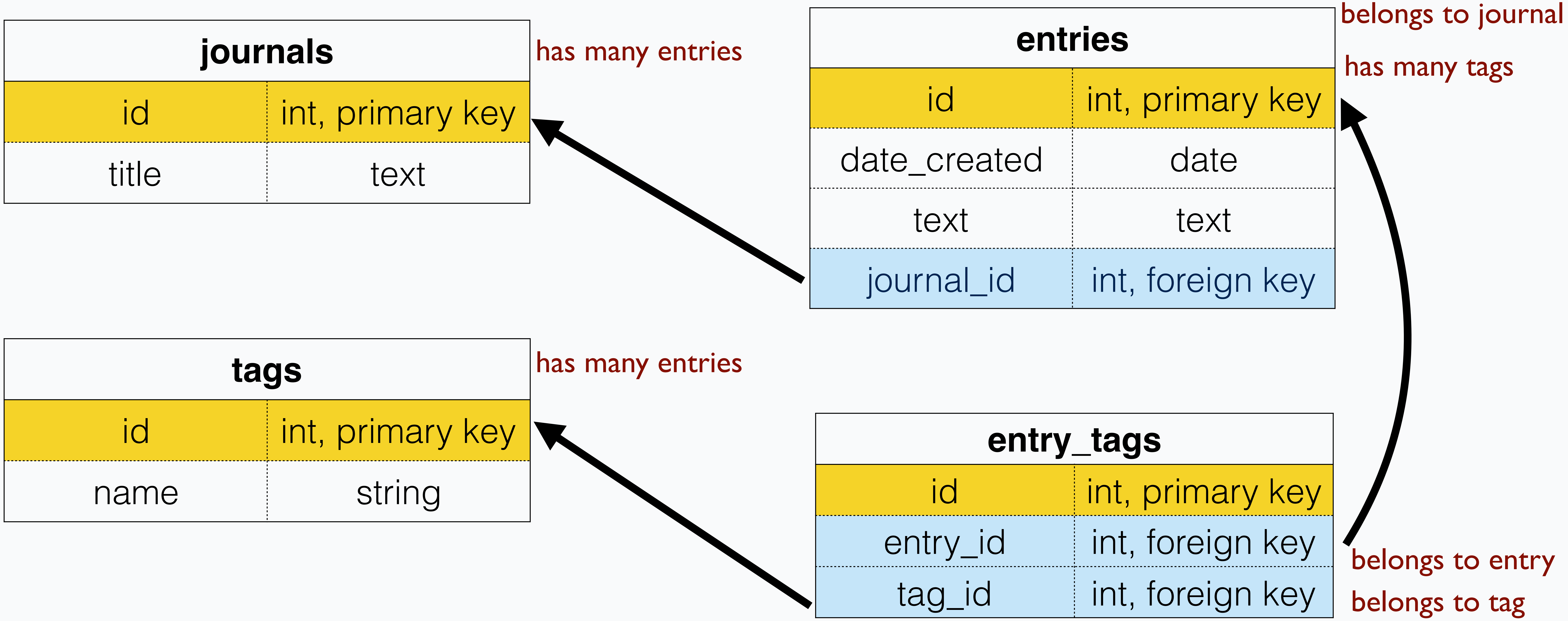
id	int, primary
entry_id	int, foreign
tag	string

belongs to

belongs to many???

Example: A Journal

Normalized!



journals	
id	int, primary key
title	text

has many entries

entries	
id	int, primary key
date_created	date
text	text
journal_id	int, foreign key

belongs to journal

has many tags

tags	
id	int, primary key
name	string

has many entries

entry_tags	
id	int, primary key
entry_id	int, foreign key
tag_id	int, foreign key

belongs to entry

belongs to tag

Normalization

- Best when application is **write-intensive**
- Usually smaller tables
  - Fast reads on single tables
  - Updates, etc. are fast
  - Data is not duplicated
  - Retrieval speed suffers due to potential joins needed

De-Normalization

- Best when application is **read-intensive**
- Usually larger tables
  - Less need for joins, etc
  - Allows for efficient index usage
  - May have redundant data
    - Updates, etc. become complex and costly

What Do I Do?!

- Real world applications will most likely have both read-loads and write-loads
- Utilize both approaches depending on the situation!
- Also, let your DBA handle most this…

Design one!

- Twitter
- Gmail
- Facebook
- Instagram

- Wordpress
- Wikipedia
- AirBnB
- Google (search)

Steps for Developing your ERD

1. Identify Entities
2. Define Relationships
3. Draw Rough-Draft ERD
4. Fill in Cardinality/Modality (arrows with relationship type)
5. Define Primary Keys
6. Draw Key-Based ERD (labeling Primary and Foreign Keys)
7. Identify Attributes
8. Map Attributes
9. Draw fully attributed ERD

Conceptual or
Logical Design
either is fine