Schema Design

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

You will be able to:

- Explain what an ‘Entity Relationship Diagram’ is (and draw one!)

- Describe what data normalization is, and when we might do it.

The dorky definition of ERD

- An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities

Pre-requisites

- What is a database?
- What are some characteristics of a good database?
- What is a primary key? A secondary key?

Example app - Dog Shelter

- **The app (and the database using it) needs to record…**
  - The dog (name and breed)
  - Which human adopted them, where they live, and if their home is suitable for a dog

What’s wrong with this database?

id	Dog Name	Breed	Fav Toy	Adopter	Adopter Address	Adopter Inspection
1	Pongo	Dalmation	Old Shoe	Cruella DeVil	House DeVil, London	FALSE
2	Perdita	Dalmation	Squeky Pig	Cruella DeVil	House DeVil, London	FALSE
3	Charles	Mutt	Rubber Duck	John	Harlem, NYC	TRUE
4	Murphy	Springer Spaniel	Stick	John	Harlem, NYC	TRUE
5	Patch	Dalmation	Teddy BEar	Cruella DeVil	House DeVil, London	FALSE
6	Mac	West Highland Terrier	Sock	Harry	4 Privet Drive	TRUE

Let’s try again…

dogs

id	Dog Name	Breed	Fav Toy	Adopter
1	Pongo	Dalmation	Old Shoe	2
2	Perdita	Dalmation	Squeky Pig	2
3	Charles	Mutt	Rubber Duck	1
4	Murphy	Springer Spaniel	Stick	1
5	Patch	Dalmation	Teddy Bear	2
6	Mac	Westie	Sock	3

humans

id	name	address	passed inspection
1	John	Harlem, NYC	TRUE
2	Cruella DeVil	House DeVil, London	FALSE
3	Harry	4 Privet Drive	TRUE

Normalization is “organizing your data to reduce redundancy and improve integrity” (aka splitting your DB into separate tables)

Designing a Schema

- **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 a SQL database: What are my tables, attributes, and relationships?
- **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?
- Conceptual Design
  - Conceptual entities and their relationships
- Logical Design
  - In a SQL database: What are my tables, attributes, and relationships?
- Physical Design
  - Javascript code, CREATE TABLE statements

Dog Shelter App

Analysis

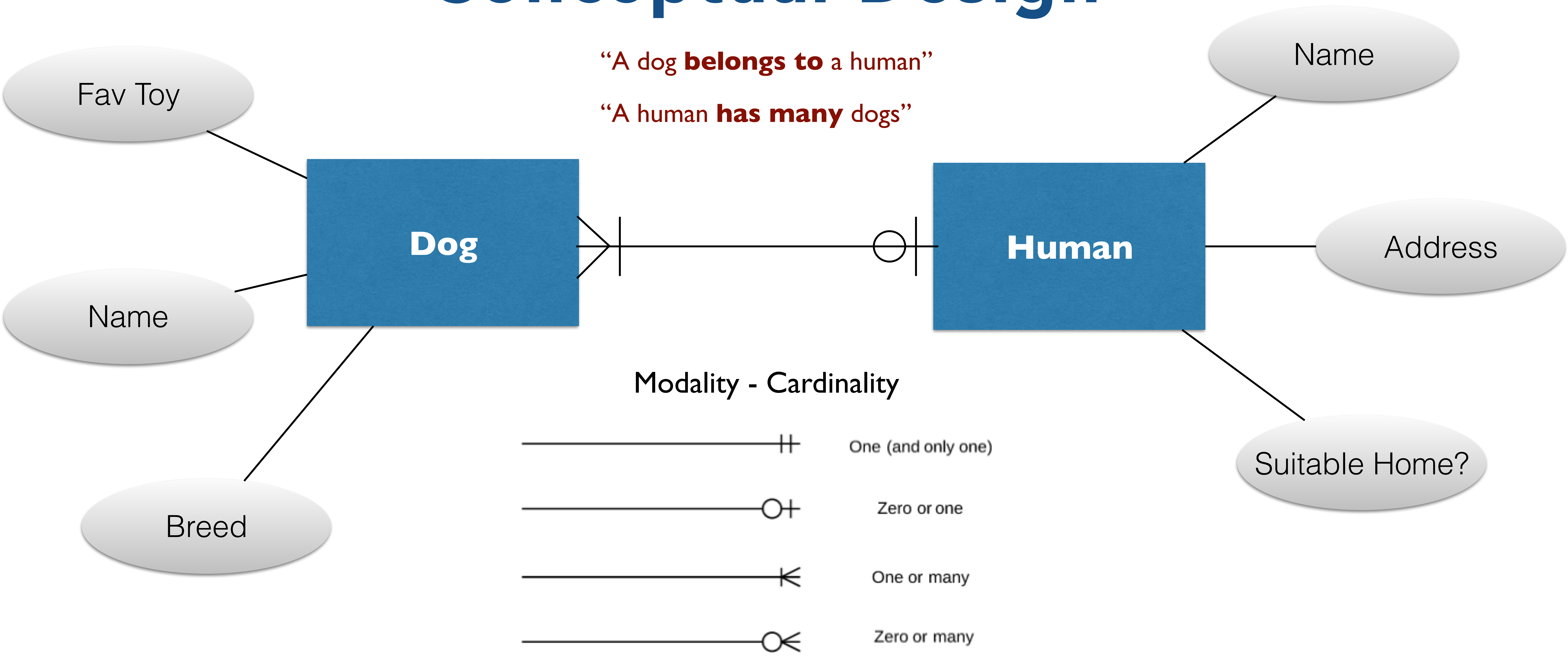
- I want a program to keep track of dogs in the shelter, and who adopts them
- I need to know each dog's name and favorite toy.
- I want to be able to maintain the address of each adopter, and whether their home is suitable for a dog

Designing a Schema

- 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 a SQL database: What are my tables, attributes, and relationships?
- Physical Design
  - Javascript code, CREATE TABLE statements

Dog Shelter App

Conceptual Design



"A dog **belongs to** a human"

"A human **has many** dogs"

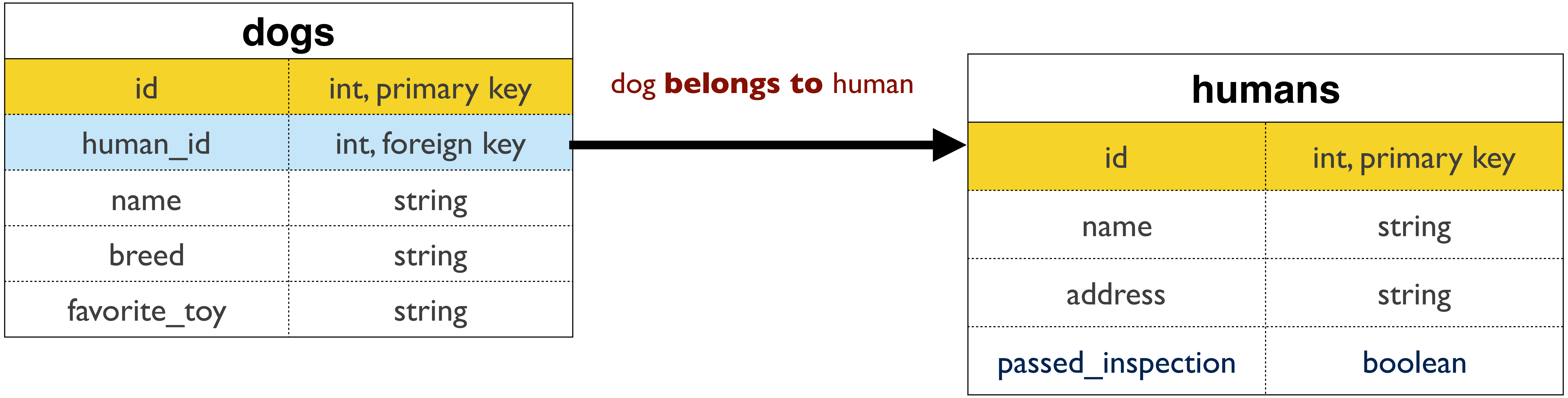
Modality - Cardinality

Designing a Schema

- 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 a SQL database: What are my tables, attributes, and relationships?
- Physical Design
  - Javascript code, CREATE TABLE statements

Dog Shelter App

Logical Design



dogs

id	int, primary key
human_id	int, foreign key
name	string
breed	string
favorite_toy	string

dog **belongs to** human

humans

id	int, primary key
name	string
address	string
passed_inspection	boolean

Designing a Schema

- 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 a SQL database: What are my tables, attributes, and relationships?
- Physical Design
  - Javascript code, CREATE TABLE statements

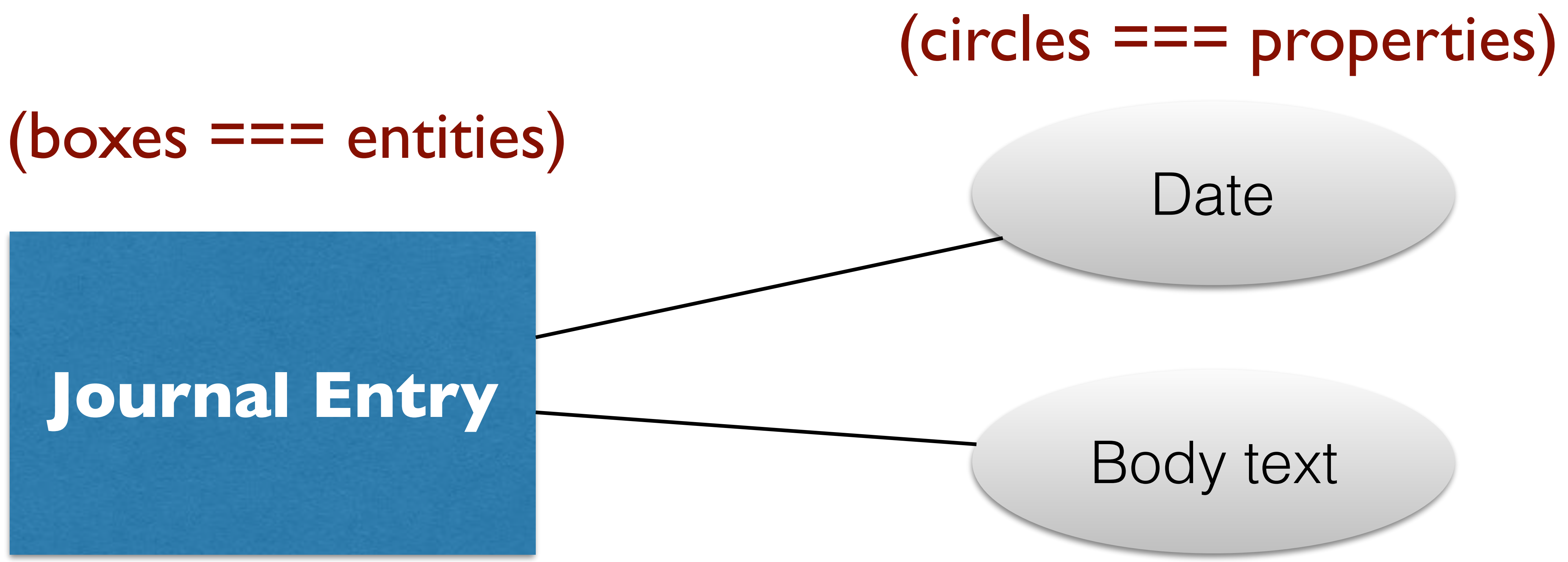
Example: A Journal

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**.

Entity Relationship Diagram (ERD)

Conceptual Design



Entity Relationship Diagram (ERD)

Logical Design

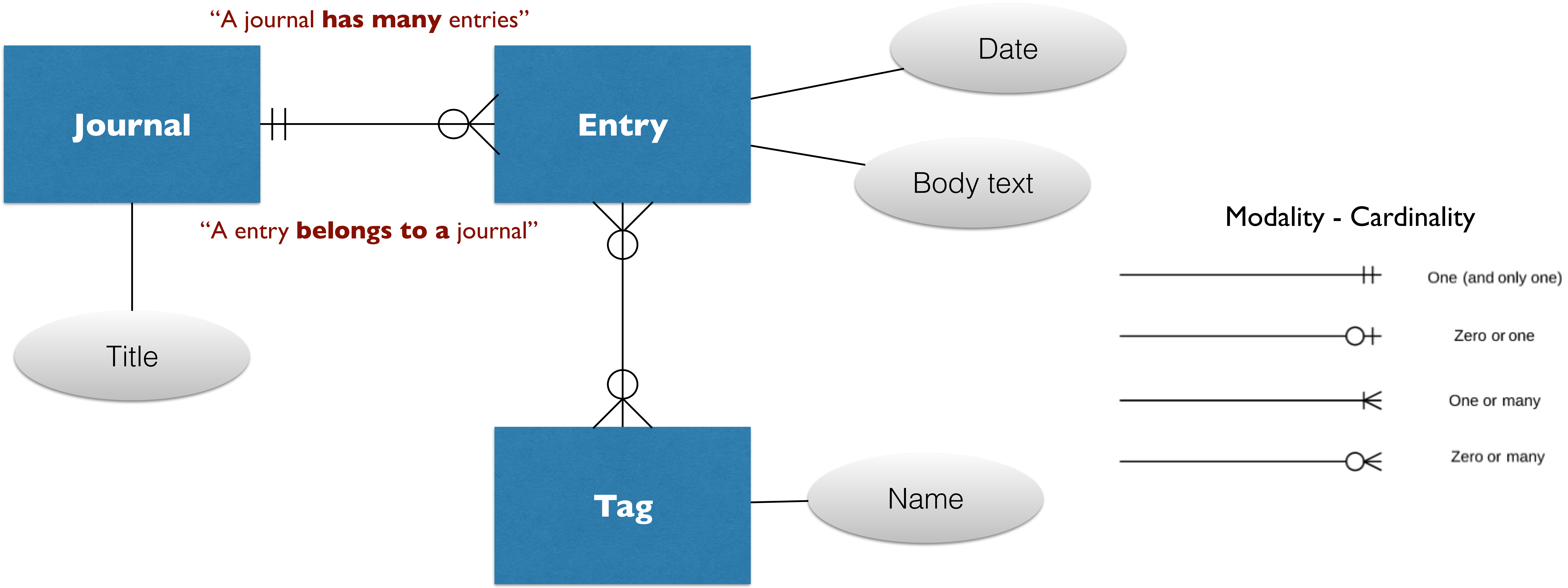
entries	
id	int, primary key
date_created	date
text	text

All done!

- **Oh wait, I forgot a couple of things**
  - 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
- **Take 2…**

Example: A Journal

Conceptual Design, Take 2

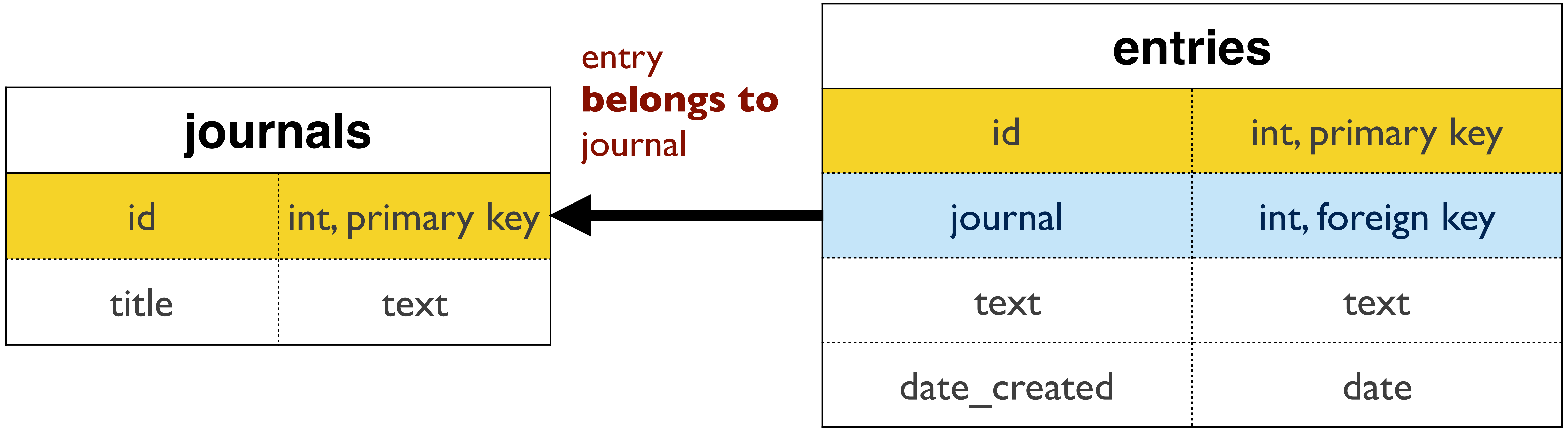


"A journal **has many** entries"

"A entry **belongs to a** journal"

Example: A Journal

Logical Design, Take 2

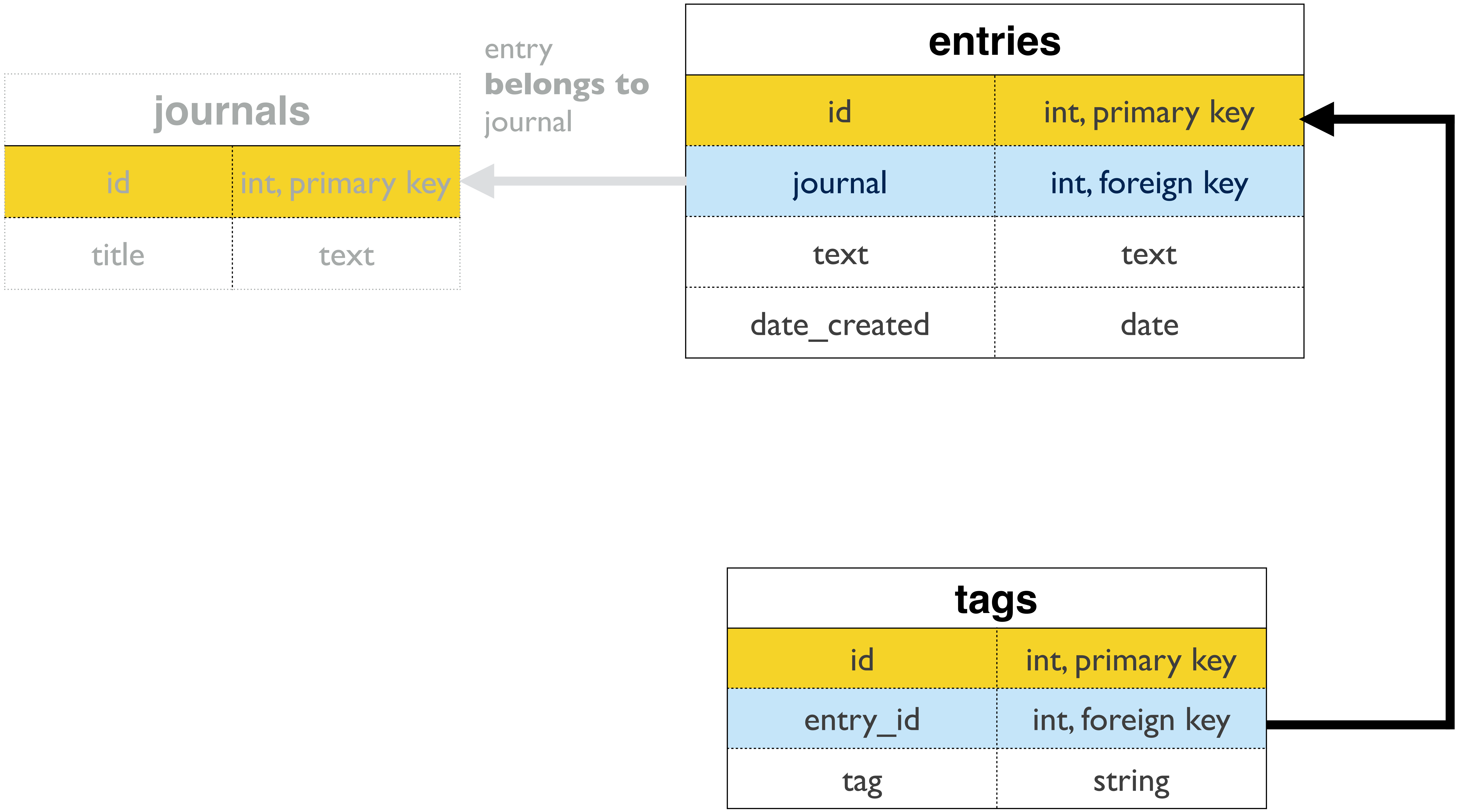


journals	
id	int, primary key
title	text

entry **belongs to** journal

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

Example: A Journal
Logical Design, Take 2



journals

id	int, primary key
title	text

entry **belongs to** journal

entries

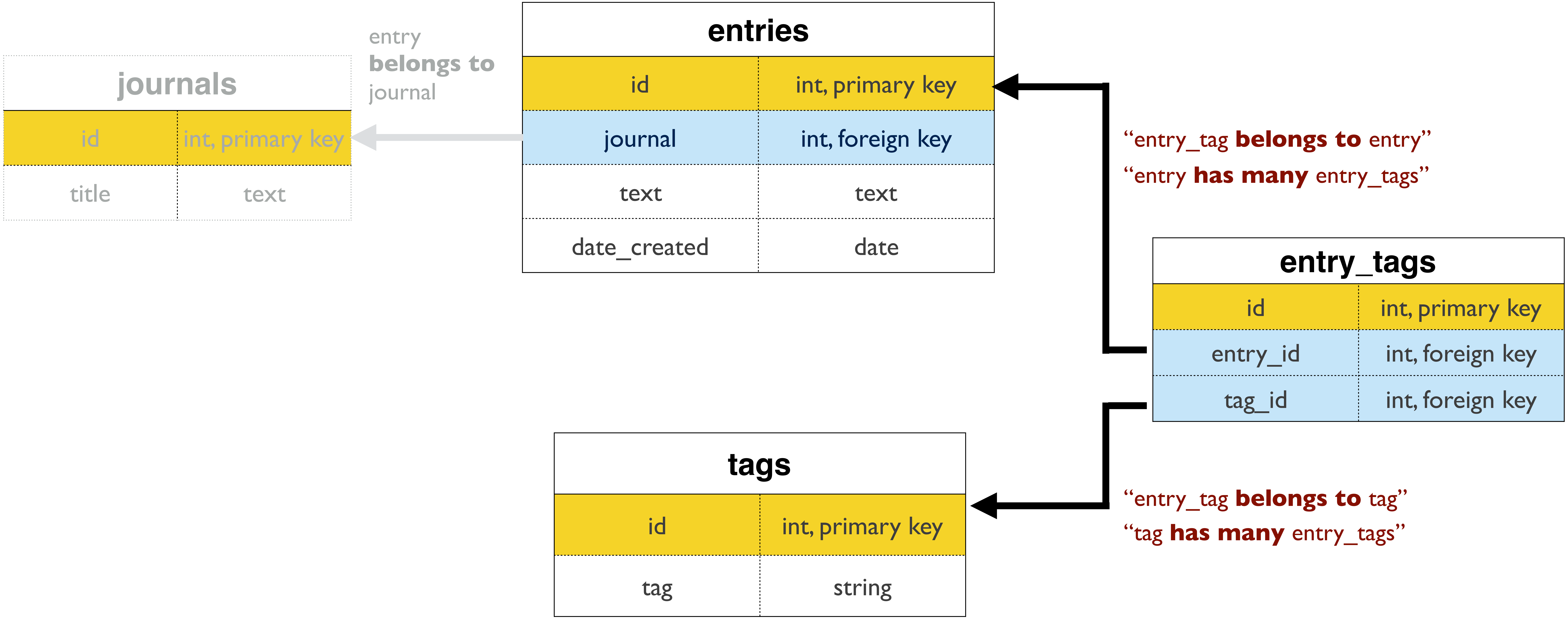
id	int, primary key
journal	int, foreign key
text	text
date_created	date

tags

id	int, primary key
entry_id	int, foreign key
tag	string

Example: A Journal

Logical Design, Take 2



journals

id	int, primary key
title	text

entry **belongs to** journal

entries

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

"entry_tag **belongs to** entry"

"entry **has many** entry_tags"

entry_tags

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

tags

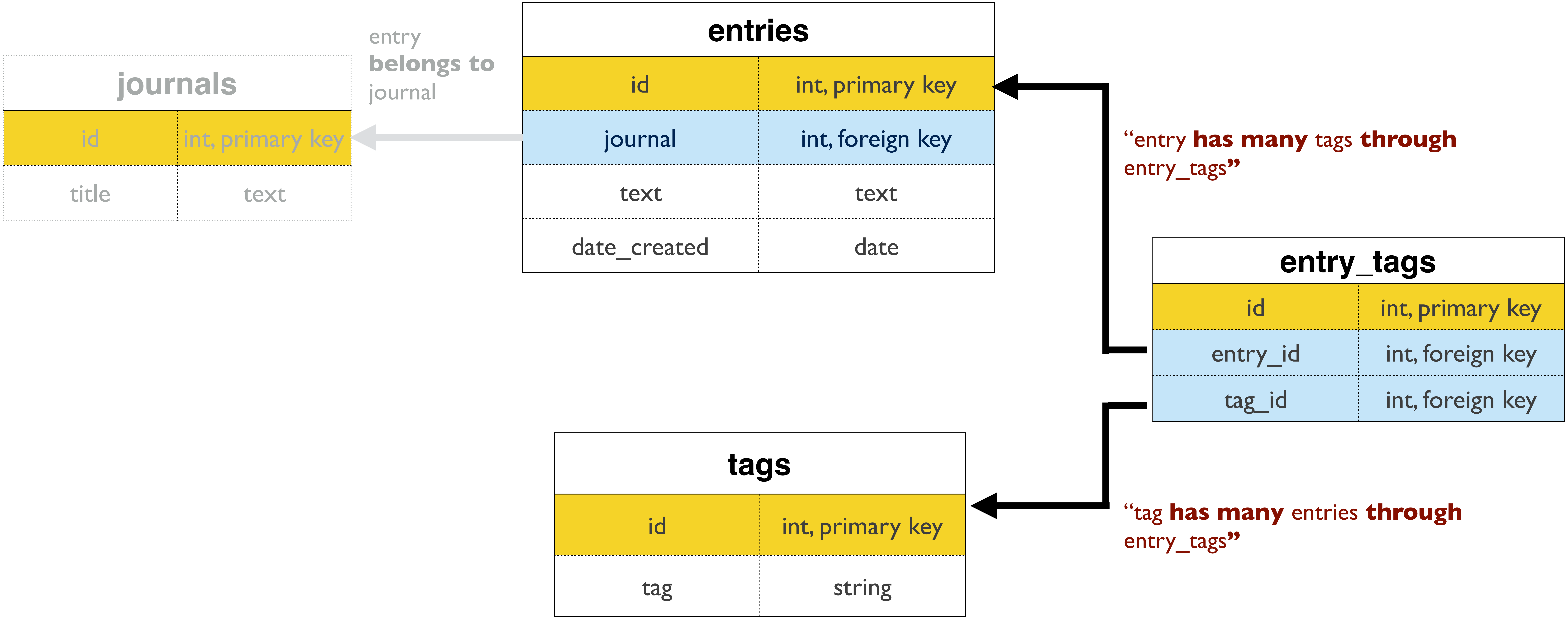
id	int, primary key
tag	string

"entry_tag **belongs to** tag"

"tag **has many** entry_tags"

Example: A Journal

Logical Design, Take 2



journals

id	int, primary key
title	text

entry **belongs to** journal

entries

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

“entry **has many** tags **through** entry_tags”

entry_tags

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

tags

id	int, primary key
tag	string

“tag **has many** entries **through** entry_tags”

Example: A Journal

Logical Design, Take 2

entries

id	journal	text	date_created
1	My Vacation	Italy is amazing! I love…	2017-01-01
2	Fullstack journey	Week one was hard as f…	2016-12-12
3	LOLZ with BFFS	Dear diary, I had a great ti…	2013-06-01

tags

id	tag
1	#amazeballs
2	#nomakeup
3	#nomnom

entries_tags

entry_id	tag_id
1	1
2	3
1	2
3	2

The three main relationships

- has many
- belongs to
- has many through

Normalized Databases

- **Focus on optimal storage - often at odds with retrieval speed due to complex queries using complicated joins**
- **Work best when the application is write-intensive and write-load is more than read-load**
  - Tables are usually smaller as data is divided vertically (fast reads on single tables)
  - Updates and Inserts are fast because there are no duplicates to update
  - Data is not duplicated so there is less of a need for process intensive group by or distinct queries
- **Normalized tables mean join tables, which mean read operations on multiple tables suffer (indexing strategies don't work as well with joins)**

Denormalized

- **Works best when the application is read-intensive**
  - The data is present in the same table (no need for joins)
  - A single table with all required data allows for efficient index usage
- **Data is duplicated which means that updates and inserts 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 of this…

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. Draw fully attributed ERD

Design one!

- Twitter
- Gmail
- Facebook
- Instagram

- Wordpress
- Wikipedia
- AirBnB
- Google (search)

You will be able to:

- Explain what an ‘Entity Relationship Diagram’ is (and draw one!)

- Describe what data normalization is, and when we might do it.