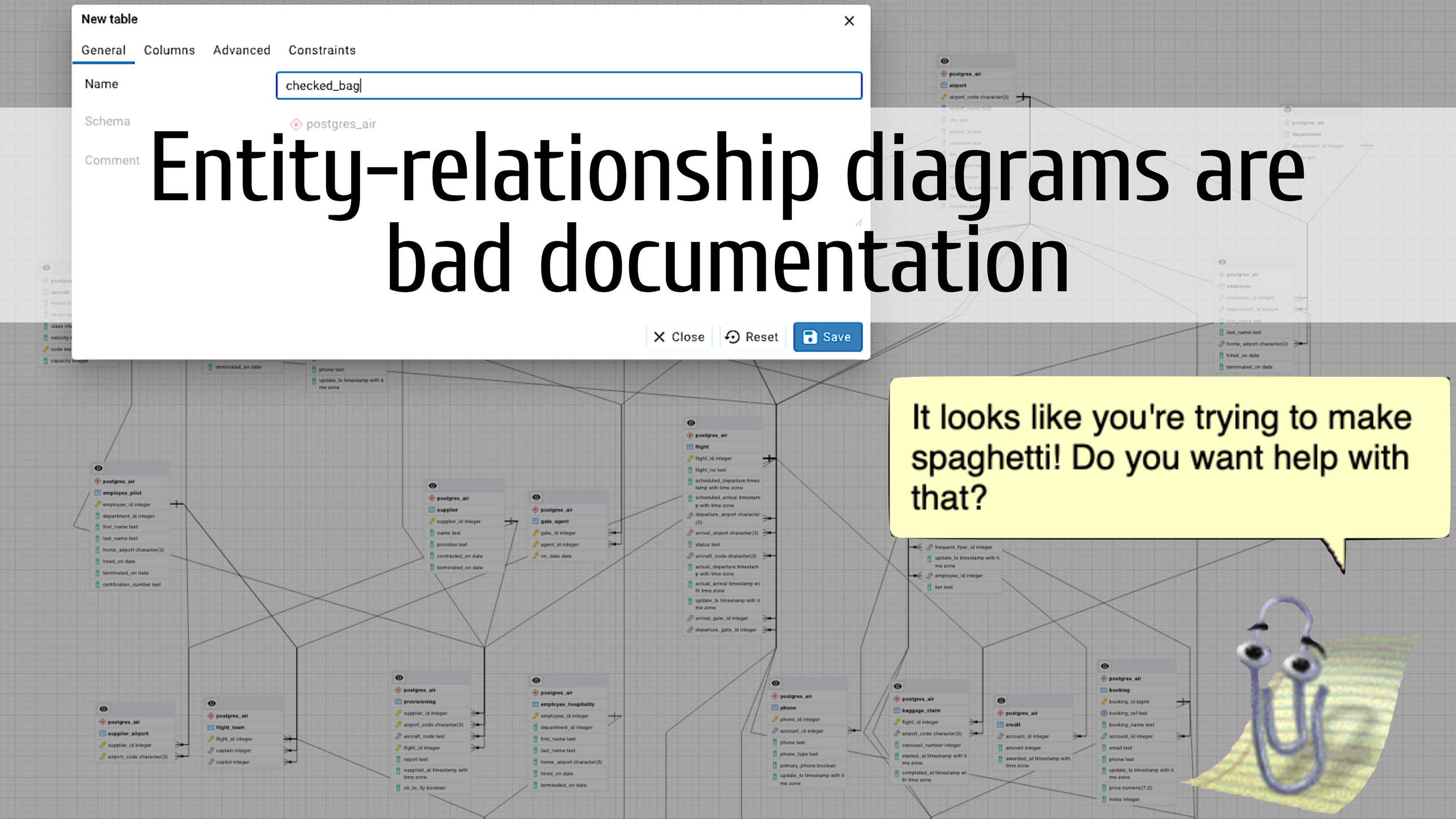
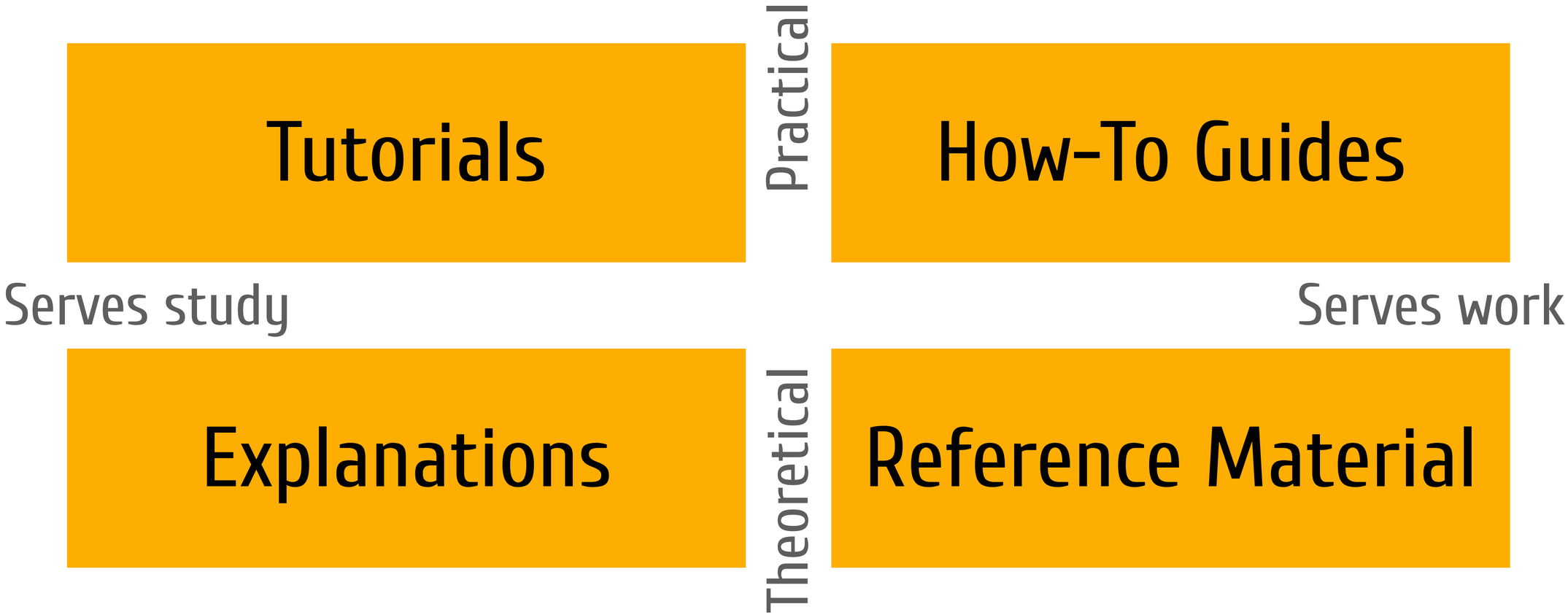
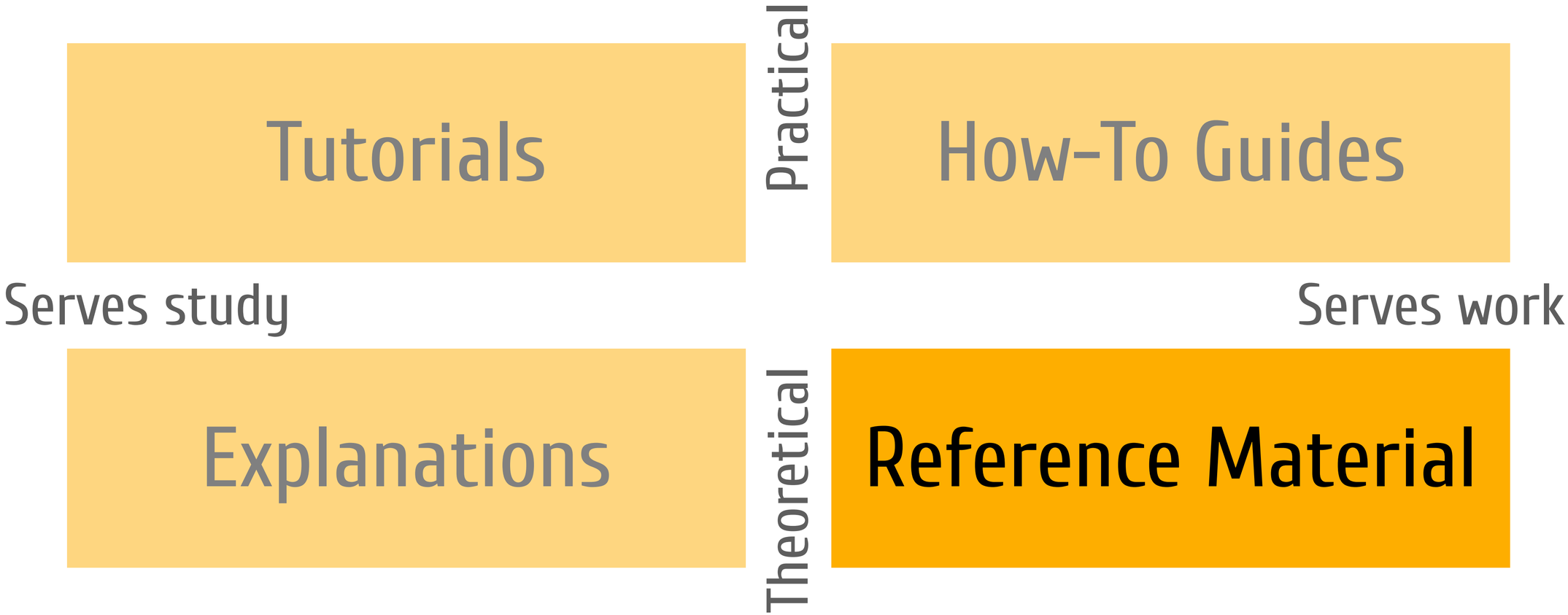
Exploring Postgres Databases with Graphs


New table
General
Columns
Advanced
Constraints
Name
checked_bag
Schema
postgres_air
Comment
Close
Reset
Save
Entity-relationship diagrams are bad documentation
It looks like you're trying to make spaghetti! Do you want help with that?

Diátaxis: documentation plays multiple roles



Whither ERDs?

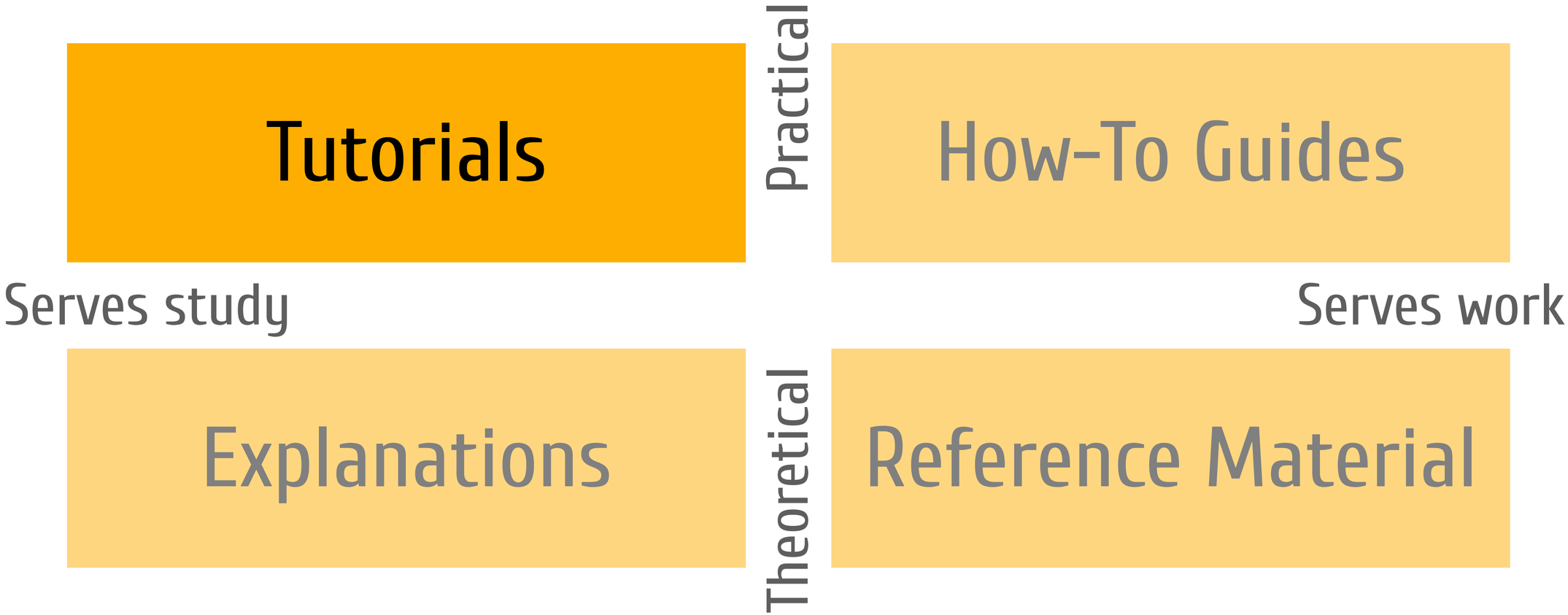


Are ERDs *good* reference material?

- Intended to be legible to original designers, not maintainers or users
- Databases tend to change more quickly than docs
- Outdated database diagrams can be worse than useless
- High effort ceiling to update and untangle
- But worst of all....

People learn the database, not the diagram

ERDs are tutorials for navigating databases



ERDs as tutorial: any better?

- Intended to be legible to original designers, not maintainers or users
- Databases tend to change more quickly than docs
- Outdated database diagrams can be worse than useless
- High effort ceiling to update and untangle

ERDs as tutorial: any better?

- Intended to be legible to original designers, not maintainers or users ✔
- Databases tend to change more quickly than docs ✔
- Outdated database diagrams can be worse than useless ✔
- High effort ceiling to update and untangle ✔
- But worst of all....

ERDs as tutorial: any better?

- Intended to be legible to original designers, not maintainers or users ✔
- Databases tend to change more quickly than docs ✔
- Outdated database diagrams can be worse than useless ✔
- High effort ceiling to update and untangle ✔
- No interactivity with the database itself, no direct route to more detail
- Primitive means for emphasizing situationally relevant processes & flows

A good tutorial for a database:

- Helps users build a “sense of place”
- Focuses on specific areas of interest
- Makes it easy to experiment
- Requires minimal maintenance effort

pdot:

Postgres + GraphViz + your shell / docs

https://gitlab.com/dmfay/pdot

One remarkable feat of the human mind is to conceive of some large spaces as integrated wholes rather than piecemeal as they are experienced.

Barbara Tversky 2003 “Structures of Mental Spaces”

7 ± 2

George Miller 1956 “Some Limits on Our Capacity for Processing Information”

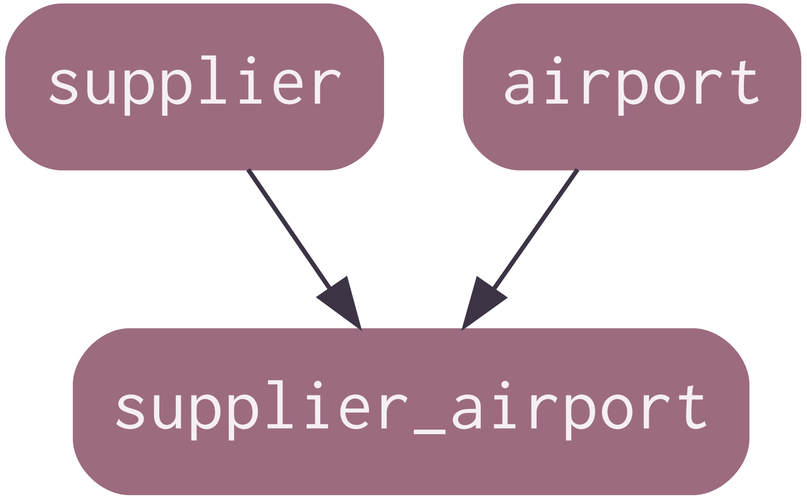
~4

Nelson Cowan 2001 “A Reconsideration of Mental Storage Capacity”

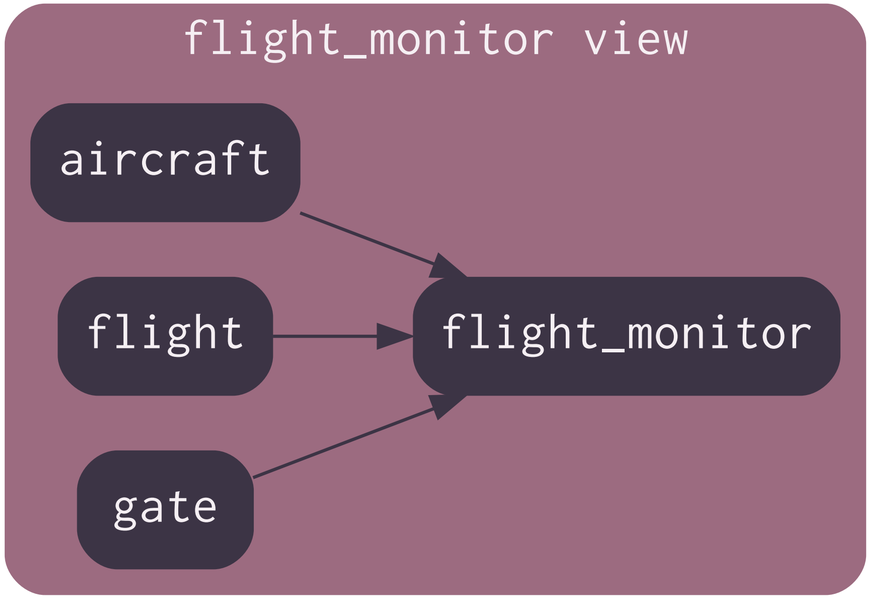
Transcending our mortal limits with chunking

- Chunks group individual information “pieces” in short-term memory
- We associate chunked items more closely with each other than with outside info
- Grouping is arbitrary, subjective, and depends on the nature of the information!
- Relevant long-term memories help increase chunk size

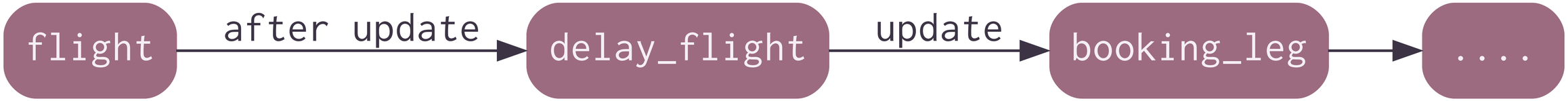
“Simple" database chunks



many:many relationship



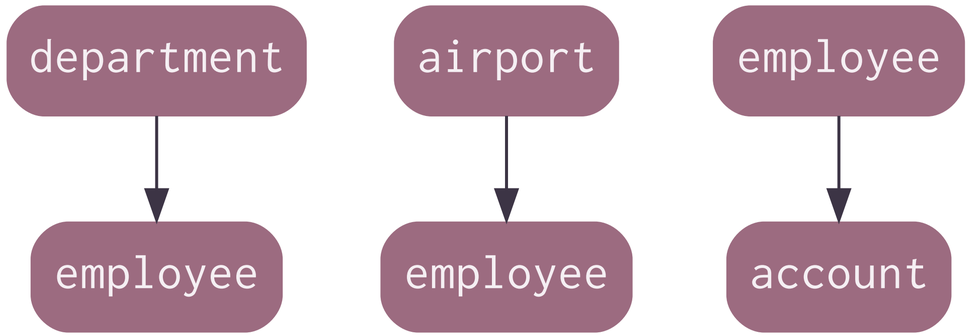
encapsulating related tables

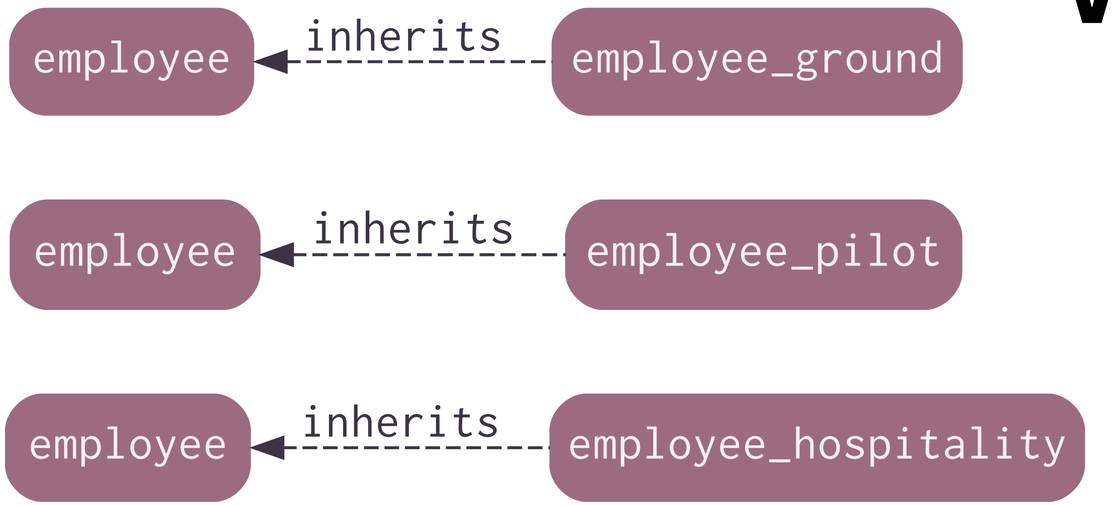


small flows and subprocesses

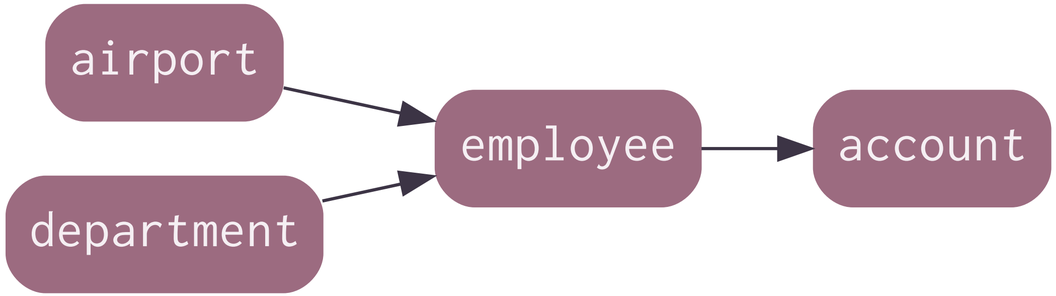
Non-expert and expert chunking

How do employees relate to other tables?

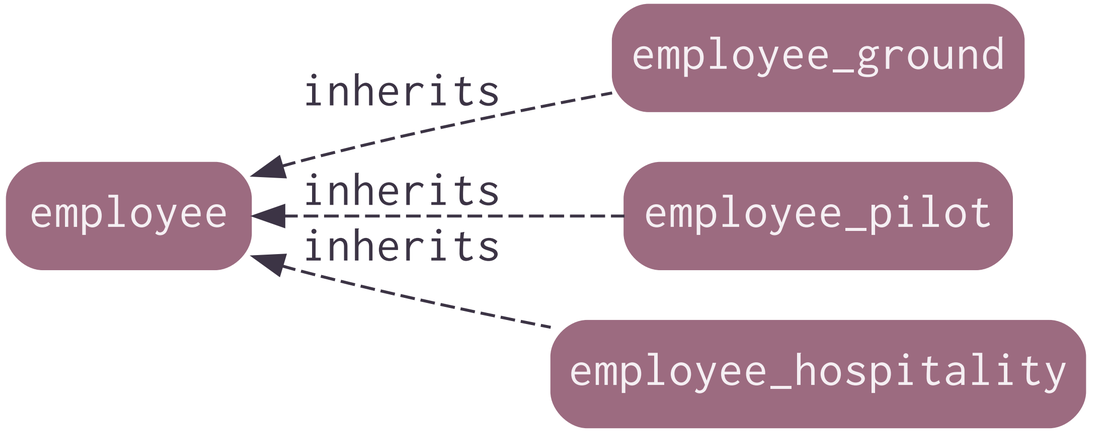




vs



an employee uses their account to work for a department from their home airport



three types of employee

How to improve chunking & storage

- Practice!

How to improve chunking & storage

- Practice!
- Prefilter visible context

How to improve chunking & storage

Filtering visible context

- Criteria need to be simple and applied consistently
- For everything with an origin object, we can rely on relevance
- Relevance alone suffices for most operations, but not for fks

How to improve chunking & storage

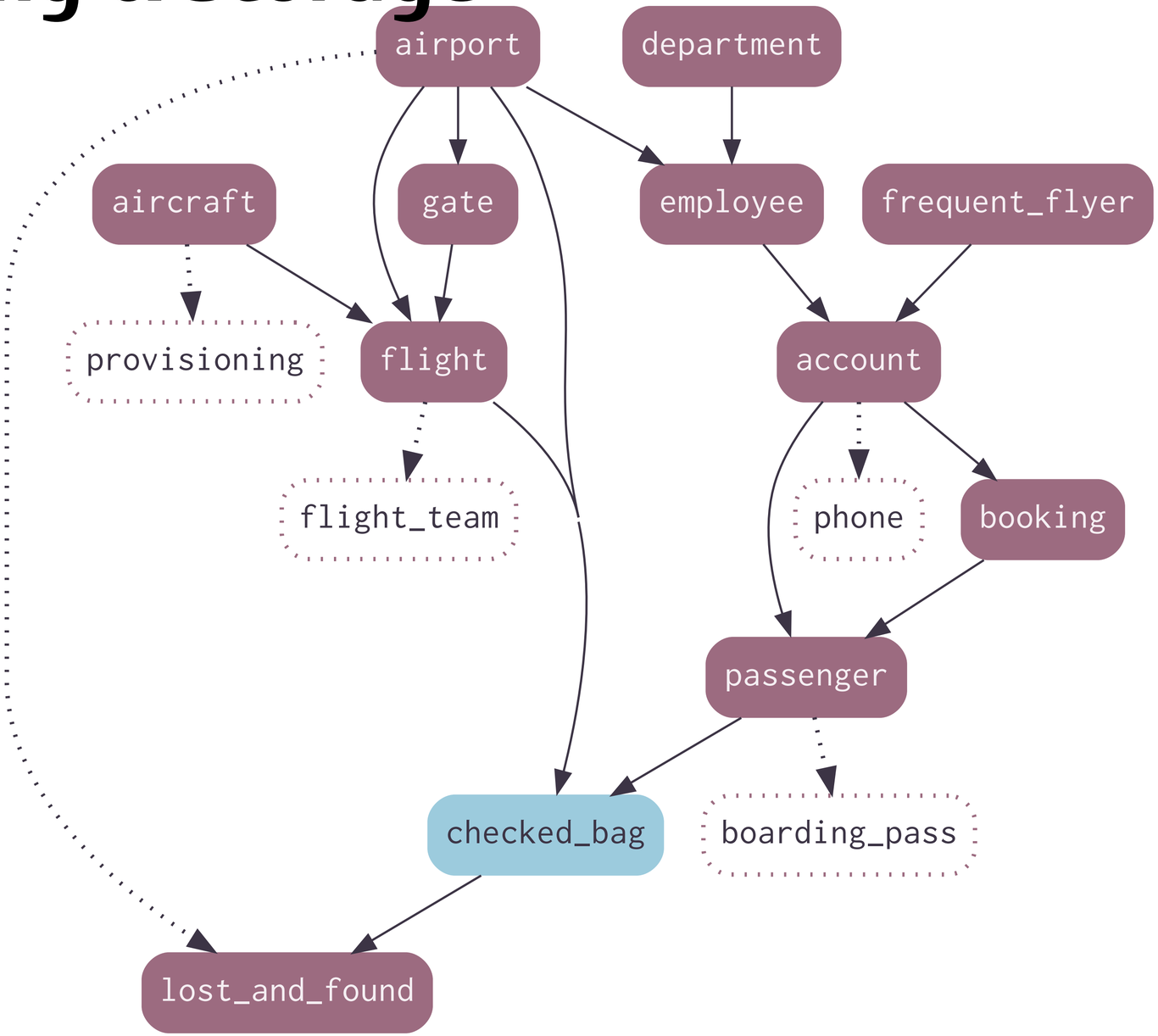
The foreign key supercluster problem

- As the ratio of relationships to tables increases, we see clusters of tables form
- Ratios I've seen range from 0.56 to 1.97; my postgres_air has 1.24 fks/table
- At a ratio of 0.5, each additional relationship likely involves at least one cluster
- Most tables in a database/schema participate in a cluster with *most other tables*

How to improve chunking & storage

pdot's linear impact filter rule

- Only some other edges in the foreign key graph affect our origin:
  - Will a dependent row in origin block a delete from $table?
  - Will a delete from origin cascade to a dependent row in $table?



How to improve chunking & storage

- Practice!
- Prefilter visible context
- Exploit natural faculty for learning through navigation

How to improve chunking & storage

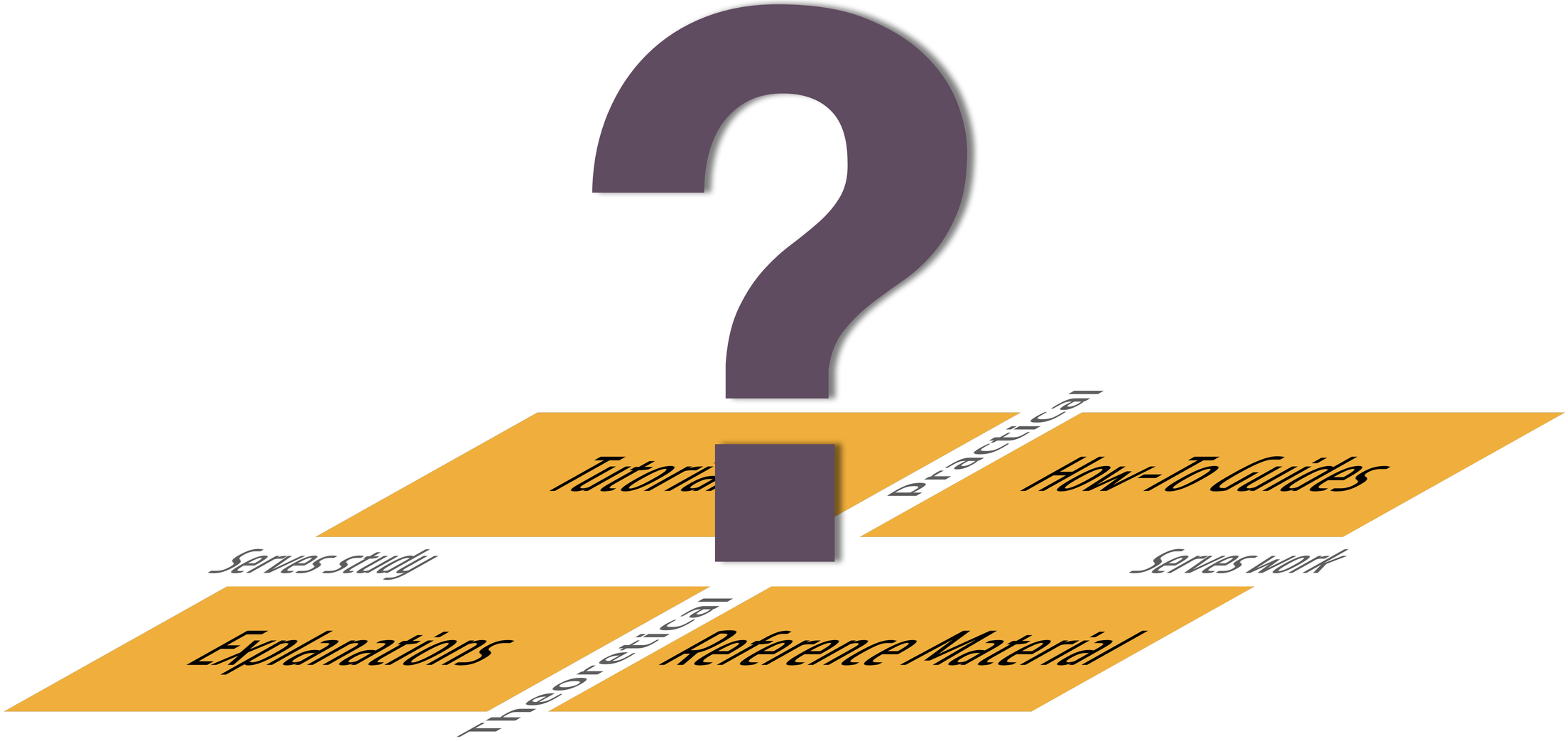
Learning through navigation

- With limited visible context, we have to move to learn more
- Spatial indexing: it isn't just for PostGIS!
  - Visual search of a diagram is nonlinear, compared to reading or searching lists
  - Proximity correlates with relevance; edges and borders help guide attention
  - Complex notation or explanation requires extra viewer effort to integrate

Addressing ERD-tutorial weaknesses

- ~~Intended to be legible to designers, not maintainers or users~~
- ~~Databases tend to change more quickly than docs~~
- ~~Outdated database diagrams can be worse than useless~~
- High effort ceiling to ~~update and~~ untangle
- No interactivity with the database itself, no direct route to more detail *
- ~~Primitive means for emphasizing situationally relevant processes & flows~~

Is this still documentation at all?



Read documentation, and you read alone

- Documentation helps a single person at a time get to grips with a system
- Individual understanding develops through feedback
- Even expert instruction is *interpreted* subjectively by each hearer

Diagrams coordinate attention

- Databases are spaces many times over
- When those spaces are rendered visually, pointing becomes the fastest way to create a “meeting of minds” in a group
- ….once you know where to point, which all-encompassing diagrams tend to be bad at

Explore your database!

https://gitlab.com/dmfay/pdot