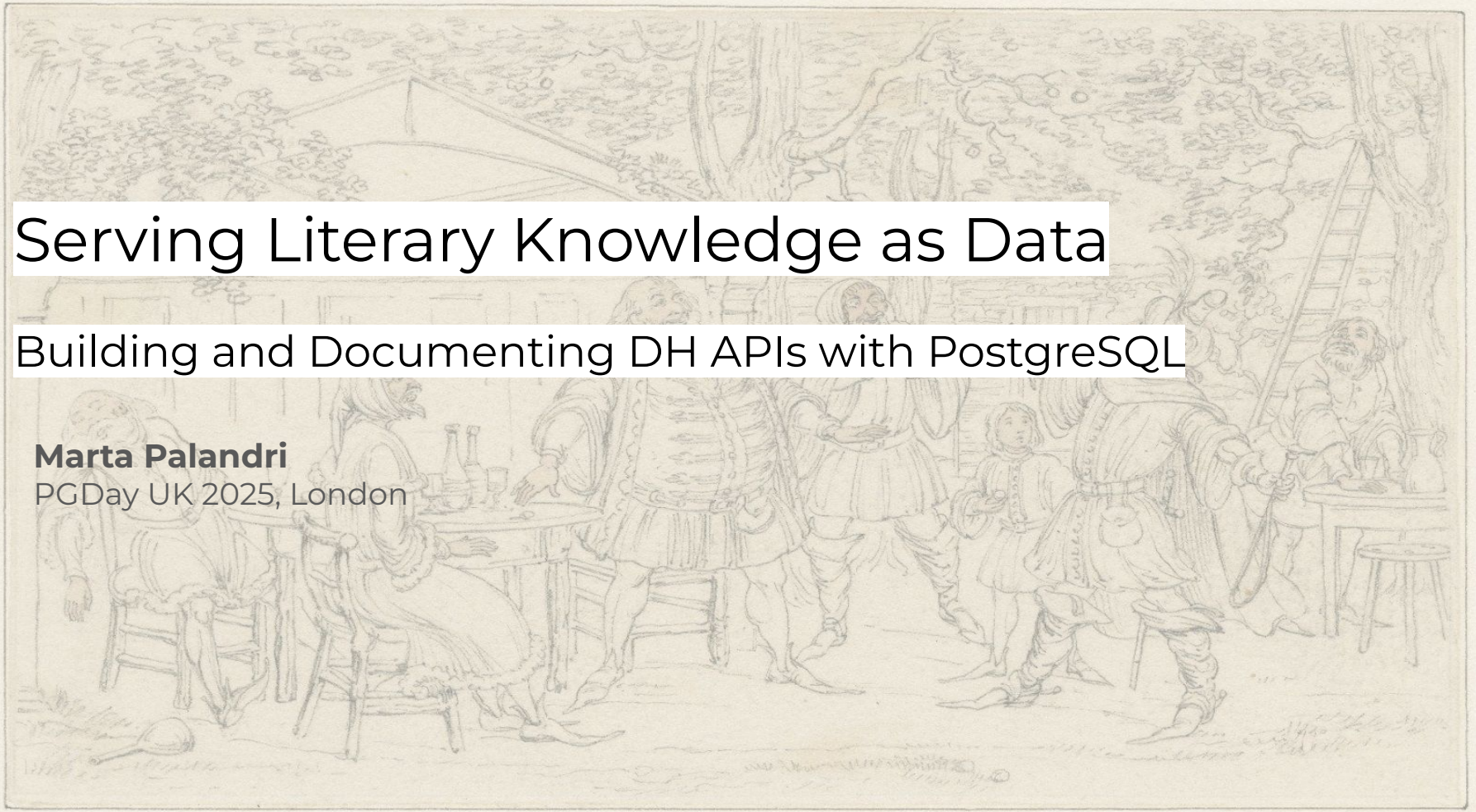


Serving Literary Knowledge as Data

Building and Documenting DH APIs with PostgreSQL

Marta Palandri

PGDay UK 2025, London

Outline

- **Intro to Digital Humanities**
- **Why Digital Humanities APIs?**
- **Why serve literary knowledge as data?**
- **The role of PostgreSQL**
- **Implementation: ShakespeareSearch API**
- **Relational DB as a model of knowledge**
- **Why PostgreSQL does it better**
- **Annotations as a separate layer of knowledge**
- **The API as scholarly lens**

Why am I talking about this?

I am a technical editor, former software developer, and humanist by training. Here are a number of DH projects I have worked on.

More:

- LinkedIn
- tech editorial

What are Digital Humanities?

- **Digital Humanities (DH)** = Interdisciplinary field: computation + humanities

- Methods: Text analysis, data visualization, digital mapping, digital archives, databases, etc.

- **Goal = Enhance research and teaching in humanities disciplines by leveraging computational techniques**

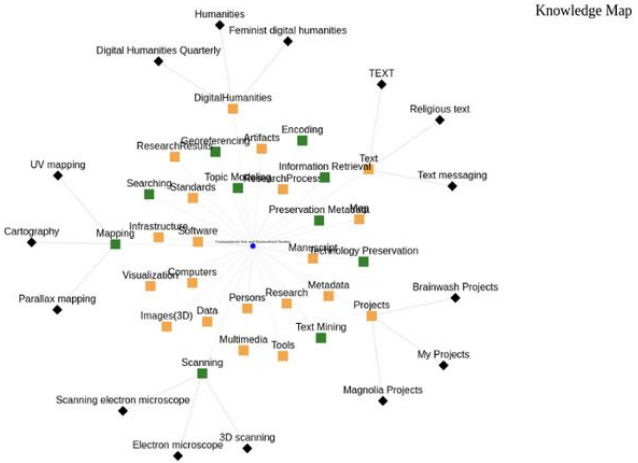
What are Digital Humanities?

In the 1940s, **Father Roberto Busa** collaborated with IBM to create the Index Thomisticus, a **comprehensive concordance of the works of St. Thomas Aquinas** (one of the first DH projects)

What are Digital Humanities?

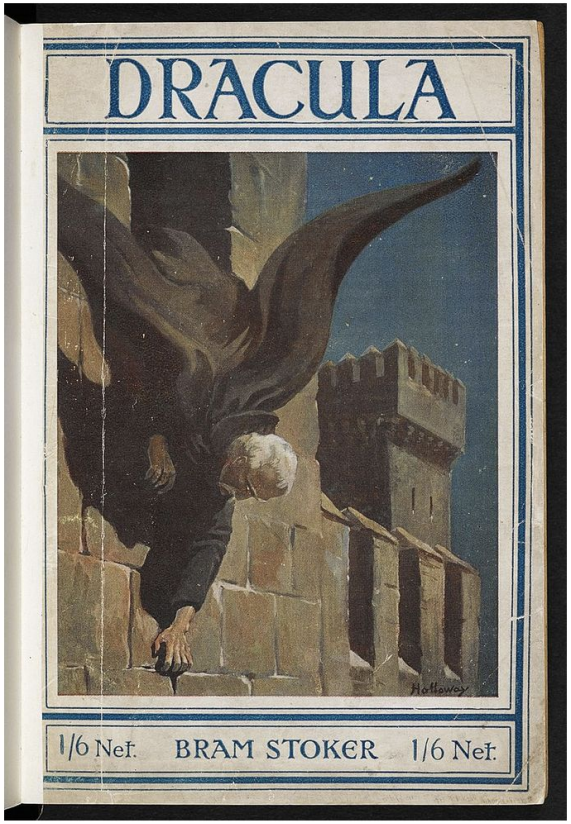
Digital Humanities (DH) = Interdisciplinary field: computation + humanities

Today, the field incorporates technologies like **machine learning and virtual reality** to further expand the possibilities for research and education.



DH Education Knowledge Map: creating knowledge webs via hypertext

By looking at data from multiple perspectives, any dataset is transformed into a sea of possibilities
See the Github repo…



Looking through the windows in Stoker's Dracula

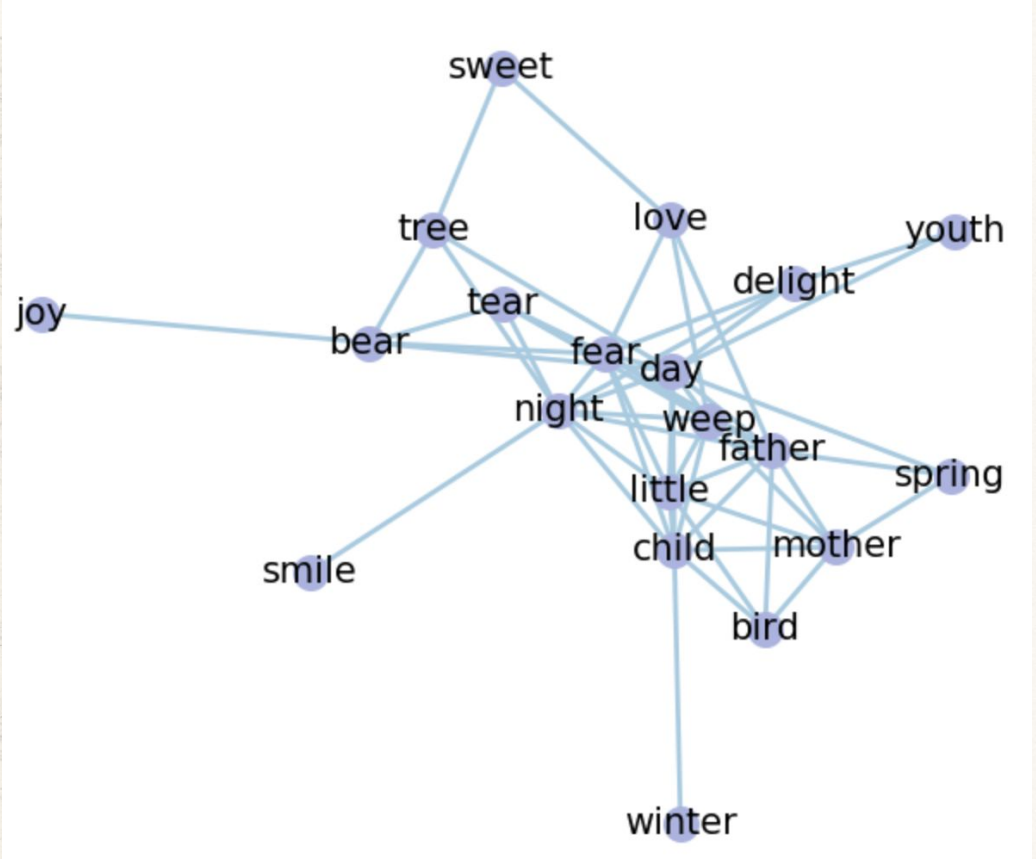
2a3a120

	Left	Node	Right	Book	In bk.
1	room lit by a single lamp, and seemingly without a	window	of any sort. Passing through this, he opened another door	dracula	
2	the stairs, trying every door and peering out of every	window	I could find, but after a little the conviction of	dracula	
3	in every breath I drew. As I leaned from the	window	my eye was caught by something moving a storey below	dracula	
4	I saw was the Count's head coming out from the	window.	I did not see the face, but I knew the	dracula	
5	the rays of the moonlight and pass out through the	window,	for I could see outside the dim, shadowy forms for	dracula	
6	the Szgany came out, and seeing them pointing to my	window,	said something, at which they laughed. ¶ Henceforth no effort of	dracula	
7	hour, when I saw something coming out of the Count's	window.	I drew back and watched carefully, and saw the whole	dracula	
8	the agonised cry of a woman. I rushed to the	window,	and throwing it up, peered between the bars. ¶ There, indeed	dracula	
9	with her head lying up against the side of the	window	sill and her eyes shut. She was fast asleep, and	dracula	
10	to spin round. I kept my eyes fixed on the	window,	but the wolf drew his head back, and a whole	dracula	
11	hear the low howl of the wolf through the broken	window.	¶ The air seems full of specks, floating and circling in	dracula	
12	dreams. There was a dog howling all night under my	window,	which may have had something to do with it; or	dracula	
13	there was no sign. Through these frowning walls and dark	window	openings it was not likely that my voice could penetrate	dracula	

Concordance of *window* in *Dracula*, KWICGrouped for words of vision (11 out of 116 total instances of *window*)

A brief network analysis of symbolism in Blake's poetry with Python

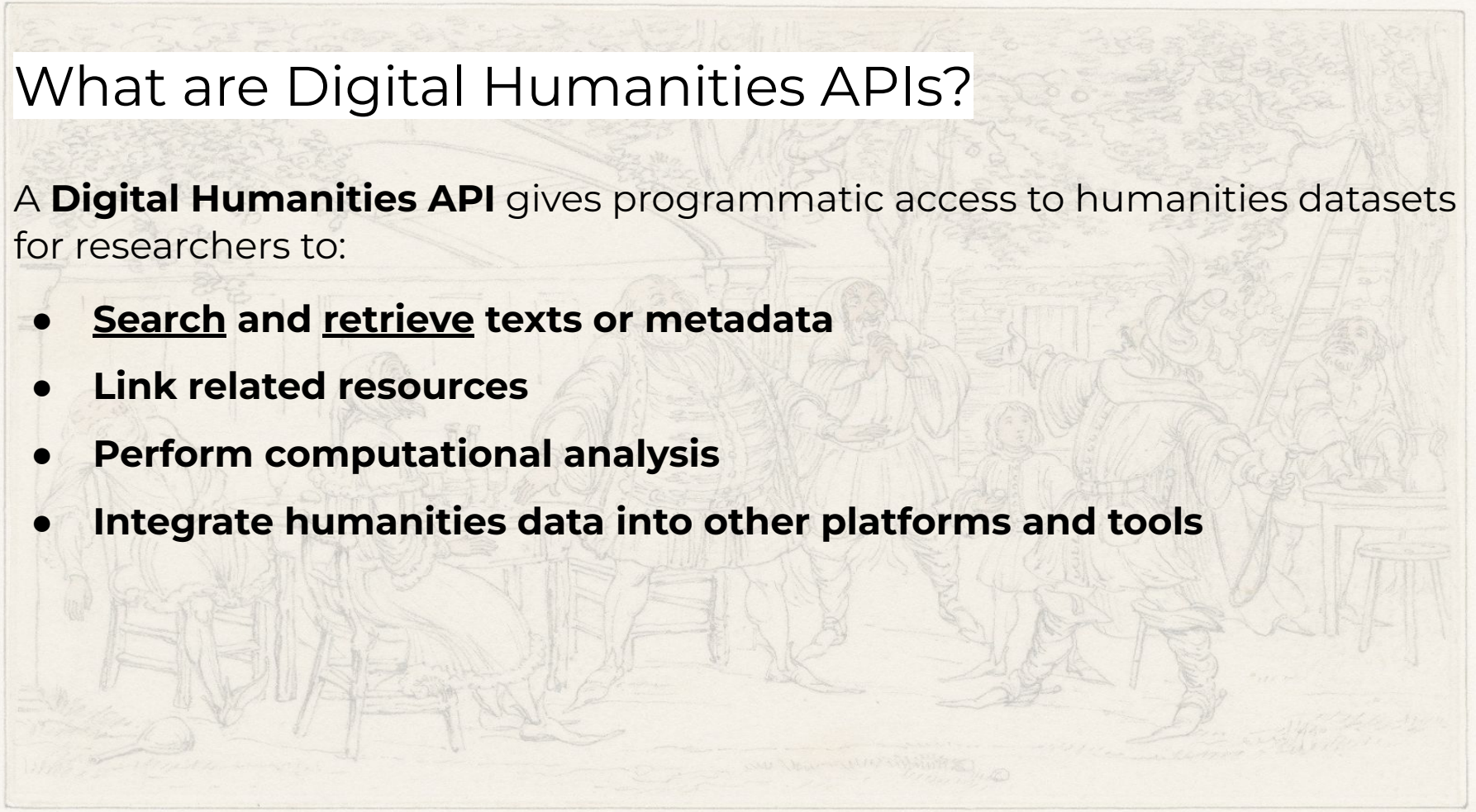
Extracting symbols and imagery from 18th-century Songs of Innocence and of Experience Tiger, tiger, burning bright In the forests of…



What are Digital Humanities APIs?

A **Digital Humanities API** gives programmatic access to humanities datasets for researchers to:

- **Search and retrieve texts or metadata**
- **Link related resources**
- **Perform computational analysis**
- **Integrate humanities data into other platforms and tools**



What are Digital Humanities APIs?

Digital Humanities APIs convert scholarly collections into machine-readable resources, revealing new realms for exploration and reuse.

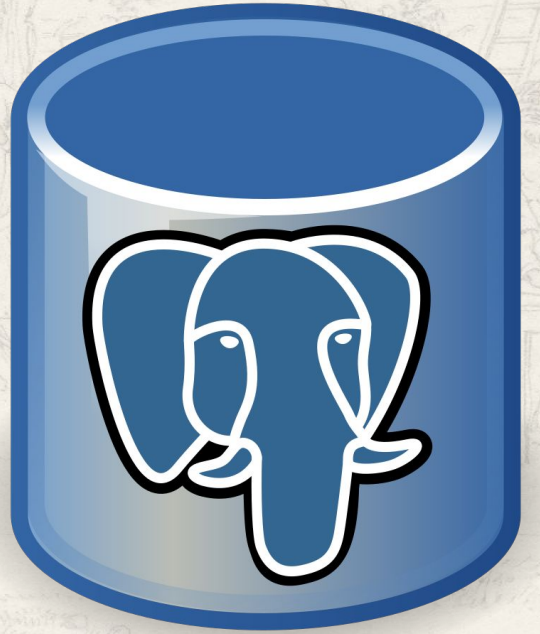
Why serve literary knowledge as data?

- **Accessibility (beyond libraries)**
- **Interoperability (link to biographies, maps)**
- **Reproducibility (shared workflows)**
- **Scalability (millions of words, metadata layers)**

The role of PostgreSQL

PostgreSQL is well-suited for DH APIs due to how it combines:

- ➔ **Relational + flexible (tables + JSONB).**
- ➔ **Full-text search (tsvector).**
- ➔ **Advanced indexing (GIN, trigram).**
- ➔ **Extensions: PostGIS, pg_trgm, ltree.**
- ➔ **Stable, open-source, long-term.**



The role of PostgreSQL

1. **PostgreSQL** becomes the **knowledge engine** for literary data.
2. The **API layer** (e.g., Flask, FastAPI, or Django) translates **scholarly queries into SQL queries** + formats results and serves them
3. **PSQL + API LAYER =** sustainable & reusable interface between *humanistic knowledge* and *computational tools*.

Implementation

A quick API is worth a million reads

- “Distant” reading (Moretti) to look at patterns across large corpora.
- An API extends this: it’s not just distant reading for one researcher, but a shared infrastructure for producing distant readings.

An API is distant reading in motion: it lets you stand back from the text, define how far back you want to stand, and which lens to use.

Implementation: ShakespeareSearch API

The API layer's role is that of a translator between literary scholars and the database by serving queryable data over HTTP.

README MIT license

ShakespeareSearch API

A simple Digital Humanities-friendly REST API that allows users to Search Shakespeare's plays by keyword, filter by play type (comedy, tragedy, history), retrieve individual scenes and characters, investigate metadata, and get annotations tied to specific line ranges.

Implementation

ShakerspeareSearchAPI

1. DB layer (Postgres + SQLAlchemy)

➔ Plays, Scenes, Lines, Characters, Metadata (JSONB), Annotations.

2. API layer (FastAPI)

➔ /plays
➔ /search_tsv
➔ /metadata/
  - /search_lines_by_metadata
➔ lines/<line_id>/annotations

Implementation

ShakerspeareSearchAPI

Image source

3. Knowledge Layer (Features)

- **Metadata = structured context**
- **Annotations = interpretative voices**
- **Full-text search = discovery**

Relational DB as a model of knowledge

Relational DB = beyond storage, **a model of how those entities interact.**

1. **Plays → Scenes → Lines → Annotations**
   ↘ **Characters**

   → Metadata

Connecting these to represent semantic relationships beyond hierarchy.

2. Queries reflect research questions instead of programming questions.

Asking research questions: Search literary lines by keywords (full-text search)

“What lines discuss battles and kings?”

```
Kaggle_Shakespeare_data=# CREATE INDEX IF NOT EXISTS idx_lines_tsv ON lines USING GIN (text_tsv);
CREATE INDEX
Kaggle_Shakespeare_data=# SELECT l.id, l.text
FROM lines l
WHERE l.text_tsv @@ plainto_tsquery('battle & king');
  id   |                      text
-------+-------------------------------------------------
  2270 | What may the king's whole battle reach unto?
  2696 | The king will bid you battle presently.
 36712 | The king himself is rode to view their battle.
(3 rows)
```

Relational DB as a model of knowledge

Asking research questions: Search plays by structured metadata

“What tragedies discuss the battle of Shrewsbury?”

```
Kaggle_Shakespeare_data=# SELECT p.id, p.title
FROM plays p
WHERE p.play_metadata @> '{"historical_event":"Battle of Shrewsbury"}';
 id |      title
----+-----------------
  2 | Henry VI Part 1
(1 row)

Kaggle_Shakespeare_data=#
```

Asking research questions: Metadata-driven queries

Relational DB is therefore

- ➔ PostgreSQL = **strict + flexible** → mirrors interpretative process
- ➔ recognizing patterns rather than focusing on elements
- ➔ allowing methodological self-awareness

Asking research questions: Metadata-driven queries

“If the database allows one to hone in on a fact or relationship quickly, it likewise enables the serendipitous connection to come forth.” (Ramsay)

Why Postgres does it better

- Strict enough to **preserve structure**
- Flexible enough to **admit uncertainty**
- Powerful enough to **surface relationships** you didn't know to look for

(That is what makes it para-interpretative, in Ramsay's sense.)

Why Postgres does it better

Extensibility = DH playground

- **Moderation workflow**: `status TEXT CHECK (status IN ('draft','review','published'))`, only editors can promote to `published` (RLS policy).
- **Conflicting interpretations**: add a `stance` field (`supporting`, `contesting`, `alternative`).
- **Consistency**: validate incoming `play_metadata` with a JSON Schema check in the app.
- **Transparency**: an `/audit` endpoint that lists who changed what and when.

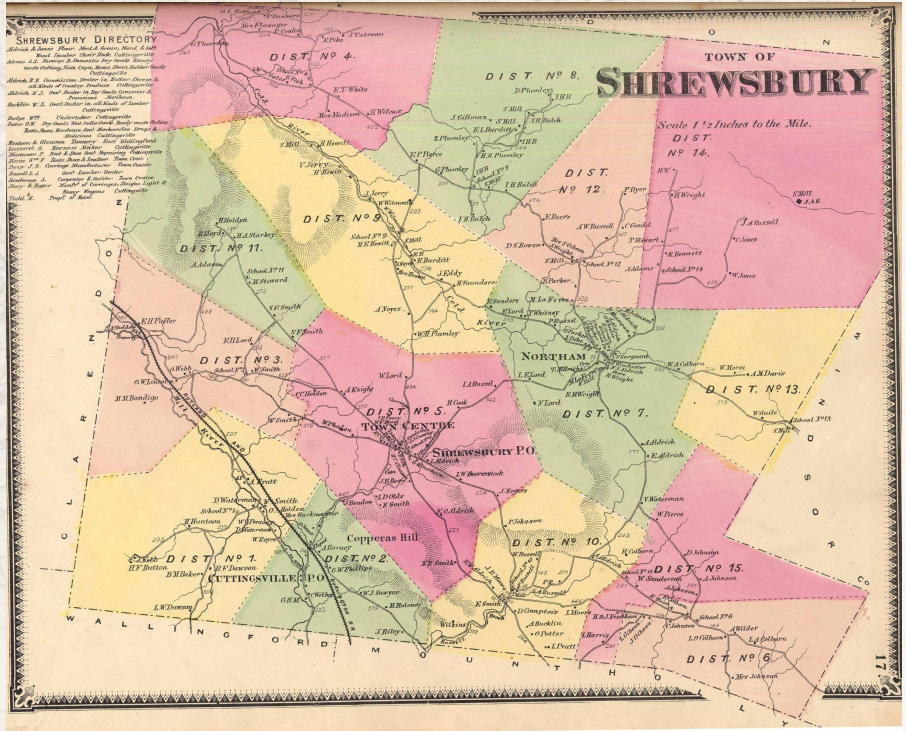
Relational DB as a model of knowledge

Why Postgres does it better

Extensibility = DH playground

- **`PostGIS`:** map performance venues or plot battles on historical maps.
- **`ltree`:** model lineage (royal succession).

→ PostgreSQL DB grows with the interpretative project.



Annotations as a separate layer of knowledge

What is an annotation?

Explanatory notes
"wherefore" → not "where," but "why."

Historical/cultural references
In *Henry IV, Part 1*, the mention of "Hotspur"

Performance/staging notes
"In modern productions, this line is often cut because it slows pacing."

Critical/interpretative glosses
"This pun on 'lie' resonates with earlier deception themes in the play."

Cross-references
e.g. Falstaff's jokes in *Henry IV* → similar humor in *The Merry Wives of Windsor*.

Annotations as a separate layer of knowledge

- Annotations as a **separate table** (=layer)
- Compared to metadata, **text is pure and untethered** (no schema)

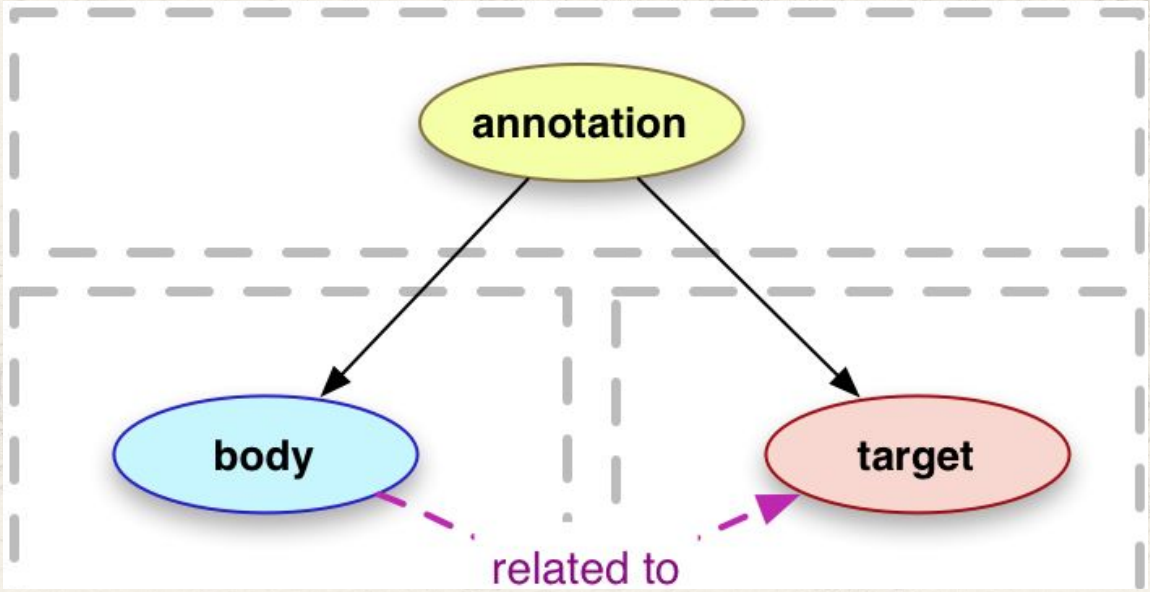
```python
class Annotation(Base):
    __tablename__ = "annotations"
    id = Column(Integer, primary_key=True, index=True)
    line_id = Column(Integer, ForeignKey("lines.id"), nullable=False)
    note = Column(JSONB, nullable=False)
    author = Column(Text)
    created_at = Column(DateTime(timezone=True), server_default=func.now())

    line = relationship("Line", back_populates="annotations")
```

Annotations as a separate layer of knowledge

The W3C Web Annotation Standard

Aim = to provide a standard description model and format to enable annotations to be shared between systems.

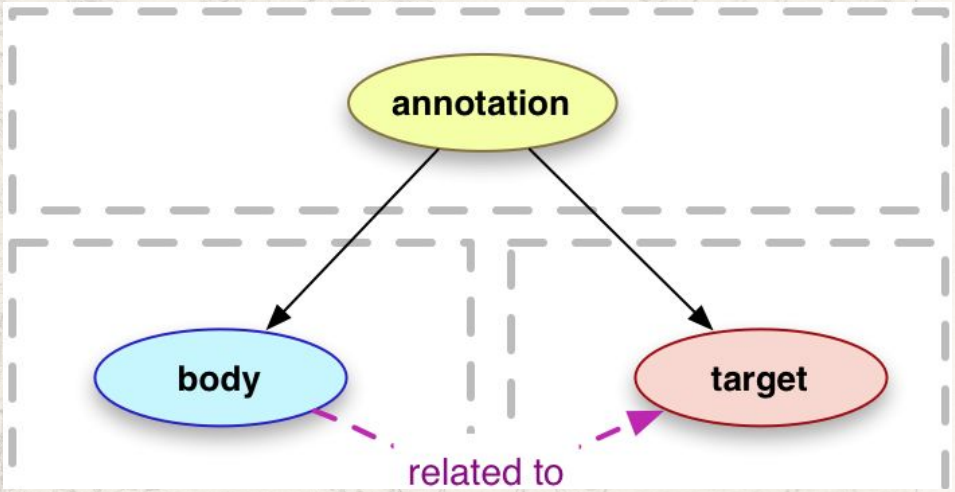


w3.org

Annotations as a separate layer of knowledge

The W3C Web Annotation Standard

- **The W3C Web Annotation Standard** → annotations are portable web resources.
- **PostgreSQL** → annotations are queryable knowledge structures.
- For both, annotations are first-class scholarly objects.



w3.org

Annotations as a separate layer of knowledge

Annotation vs. Metadata

- **Metadata** = **about** an object.
  → “*Hamlet* was first performed in 1601”, “This is a Tragedy”, “Source: EBSCO”.
  → stable, catalog-like, factual

- **Annotation** = **attached** *to* an object.
  → “This line is an example of forbidden love symbolism”, “This speech echoes Senecan tragedy”, “Cross-refers to Marlowe’s *Doctor Faustus*”.
  → scholarly, subjective, layered

The API as a scholarly lens

API + DB + docs ≠ mere delivery layer

- Frames knowledge
- Structures **how** scholars encounter and manipulate literary texts

Endpoints as interpretative guides

- `GET /lines/…` → access textual units
- `GET /lines/<line_id>/annotations` → scholarly commentary layer
- `GET /play/metadata` → contextual, historical frame
- Each design choice reflects interpretative priorities

The API as a scholarly lens

Documentation = scholarly argument

- Explains why annotations are separate, why `JSONB` is used for metadata
- Frames the scholar's **interaction** with the data

API supports collaborative discovery

- `POST …/annotations` and `…/metadata` allow multiple scholars to contribute
- Enhances the "serendipitous apprehension" of relationships in DH databases

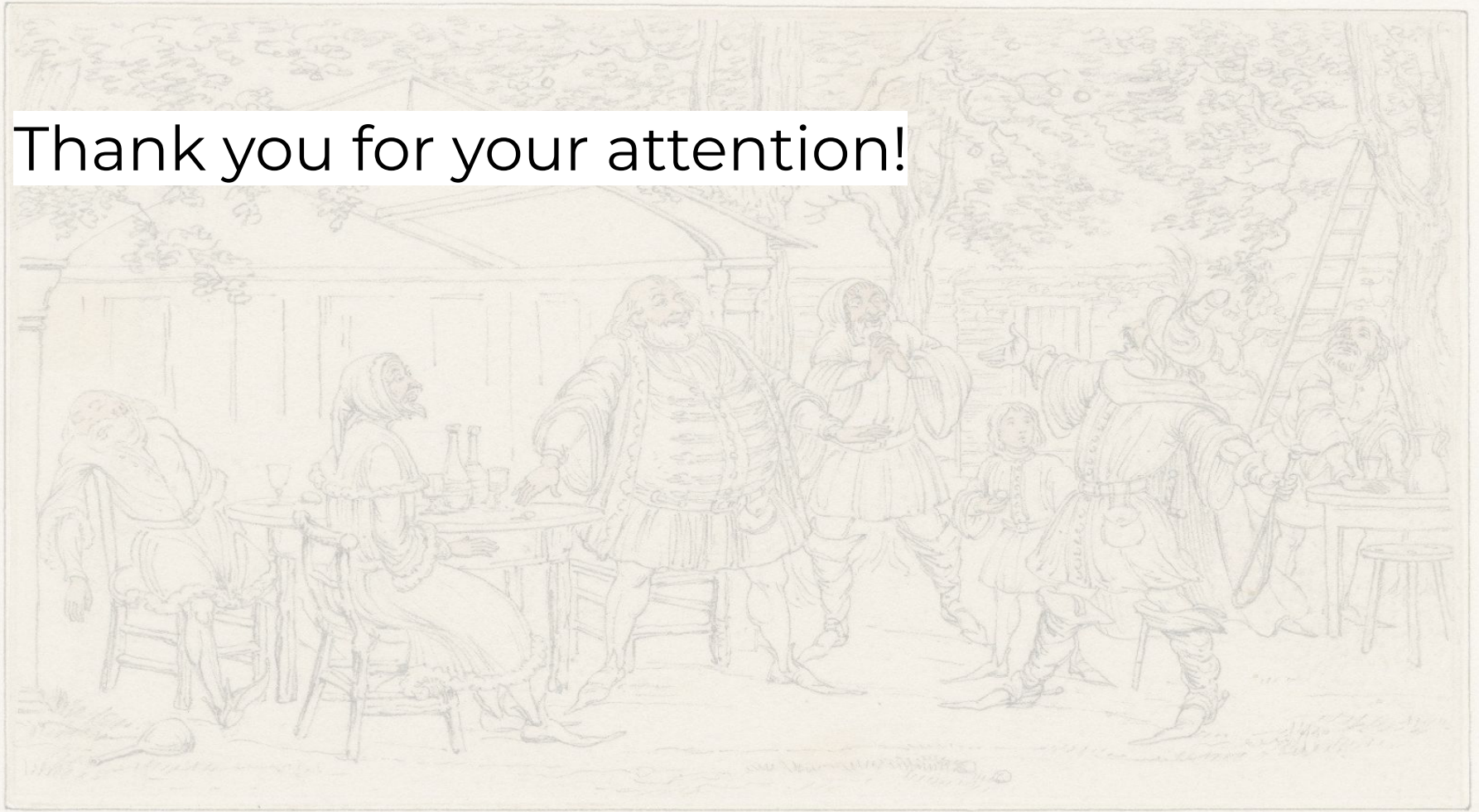
The API as a scholarly lens

Data models as interpretative objects

- Choices in tables, fields, and types reveal methodological decisions
- API responses become **research outputs**

DH API + DB = para-textual artifact

Thank you for your attention!



Sources

Encyclopaedia Britannica. Encyclopaedia Britannica, Inc., www.britannica.com.

Beers, F. W. (Frederick W.). Town of Shrewsbury. New York, NY: F. W. Beers, 1869-01-01. JSTOR, https://jstor.org/stable/community.35428504. Accessed 8 Sept. 2025.

Cruikshank, George, and Robert B. Brough. King Henry IV, Pt. II, V, 3, Pistol Informing Sir John of Henry IV's Death. Folger Shakespeare Library, before 1857. Pencil and watercolor drawing. Artstor, JSTOR, jstor.org/stable/community.25239740.

"Digital Humanities (DH)." EBSCO Research Starters, www.ebsco.com/research-starters/social-sciences-and-humanities/digital-humanities-dh.

Harlow, George Henry, 1787-1819, and Dyer, J., active 19th century. Mrs. Siddons as Queen Katherine [in Shakespeare's] Henry the Eighth [Graphic]. 1 print : engraving, 1828. Folger Shakespeare Library. Open: Folger Shakespeare Library. Artstor, JSTOR, https://jstor.org/stable/community.25247258. Accessed 8 Sept. 2025.

"Henry IV, Part 1." Internet Shakespeare Editions, University of Victoria, internetshakespeare.uvic.ca/doc/1H4_StageHistory/index.html.

"Henry IV, Part 1." Shakespeare Documented, Folger Shakespeare Library, shakespearedocumented.folger.edu/plays-poetry/henry-iv-part-1.

Henry IV, Part 1. Folger Shakespeare Library, 2022, folger-main-site-assets.s3.amazonaws.com/uploads/2022/11/henry-iv-part-1_PDF_FolgerShakespeare.pdf.

Sources

Hunter, L. “Fact – Information – Data – Knowledge: Databases as a Way of Organizing Knowledge.” Literary and Linguistic Computing, vol. 5, no. 1, 1990, pp. 49–57. Oxford UP, doi:10.1093/llc/5.1.49.

“Looking Ahead to 2012 and Looking Back to Father Roberto Busa.” Townsend Center for the Humanities, University of California, Berkeley, townsendcenter.berkeley.edu/blog/looking-ahead-2012-and-looking-back-father-roberto-busa.

“MLA and Digital Humanities.” Fitchburg State University Library Guides, fitchburgstate.libguides.com/c.php?g=1324355&p=9748052.

Schreibman, Susan. “Digital Scholarly Editing.” In A Companion to Digital Humanities, edited by Susan Schreibman, Ray Siemens, and John Unsworth, Blackwell, 2004, companions.digitalhumanities.org/DH/content/9781405103213_chapter_15.html.

“Shakespeare Plays.” Kaggle, www.kaggle.com/datasets/kingburrito666/shakespeare-plays.

Smithies, James. “Foundations of Distant Reading: Historical Roots, Conceptual Development, and Theoretical Assumptions around Computational Approaches to Literary Texts.” DH2020: Digital Humanities Conference, 2020, dh2020.adho.org/wp-content/uploads/2020/07/521_FoundationsofDistantReadingHistoricalRootsConceptualDevelopmentandTheoreticalAssumptionsaroundComputationalApproachestoLiteraryTexts.html.

“Social and Technical Infrastructure for DH.” Digital Humanities Quarterly, vol. 13, no. 3, 2019, dhq.digitalhumanities.org/vol/13/3/000430/000430.pdf.

Sources

TEI: Text Encoding Initiative. TEI Consortium, tei-c.org.

Terras, Melissa. “The Text Encoding Initiative (TEI) and the History of the Digital Humanities.” Digital Humanities Quarterly, vol. 13, no. 2, 2019, dhq-static.digitalhumanities.org/pdf/000655.pdf.

Vesanto, Juuso, et al. “On the Use of Machine Learning for Shakespearean Text Analysis.” arXiv, 2018, arxiv.org/abs/1803.03198.

W3C. Web Annotation Data Model. World Wide Web Consortium, www.w3.org/TR/annotation-model.

Walkowski, Niels-Oliver. “The Landscape of Digital Annotation.” Journal of the Japanese Association for Digital Humanities, 2016, www.sdjt.si/wp/wp-content/uploads/2016/09/JTDH-2016_Walkowski_The-Landscape-of-Digital-Annotation.pdf.

Wellcome Collection. “King Henry IV, Part II, Act V, Scene III.” Wellcome Collection, 29 Mar. 2018, wellcomecollection.org/works/ae3ft2xe. CC-BY-4.0.