Knowledge Graphs of Islamic Manuscript Materials

Recipes · Conservation Methods · Degradation — definitions, methodology, use cases, and a defence of the design for publication.

Built from **Ink-database.jsonl**: 3,313 extracted findings drawn from 300 scholarly papers on historical inks, pigments and writing supports, with the Islamic manuscript tradition as the analytical focus (2,474 findings, 74.7%).

Graph	Nodes	Edges	Records	Focus
Recipes	7,391	30,757	2,289	materials, ingredients, colours
Conservation	1,058	4,177	475	protection methods & agents
Degradation	2,963	12,219	1,656	decay phenomena & mitigation

Figures reflect the full graphs (the interactive viewer opens a compact subset and reveals the rest on demand).

Prepared as a companion to the interactive viewers (index.html / conservation.html / degradation.html) and the Neo4j exports.

1. What a knowledge graph is

A **knowledge graph (KG)** represents information as a network: **nodes** (entities or concepts) connected by **edges** (typed relationships). The atomic unit is a **triple** — *subject → relation → object* — for example *Iron-gall ink → uses → Gum arabic*. Many triples sharing nodes form a graph that can be browsed, queried and analysed as a network.

Each node carries **properties** (a label, a type, counts, free-text fields), and each edge carries a **type** and may carry properties such as a weight. This is the **labelled property-graph** model used by graph databases such as Neo4j, and the model these three graphs follow.

Key terms

Term	Meaning in this project
Node	An entity or concept (a pigment, an ingredient, a region, a degradation phenomenon, a record).
Edge / relation	A typed, directed link between two nodes (e.g. USES_INGREDIENT, EXHIBITS, MITIGATED_BY).
Triple	subject–relation–object; the smallest statement of knowledge.
Ontology / schema	The agreed set of node types and relation types — the 'grammar' of the graph.
Class hierarchy	'is-a' structure (Iron-gall ink IS_A Ink IS_A Material).
Weight	On aggregated edges, how many source records support that link (co-occurrence count).

Ontology vs. knowledge graph vs. database

These terms are often blurred; the distinction matters when defending the work:

- **An ontology** is the schema and its logical rules — the classes, properties and constraints. It supports formal reasoning. Ours is a lightweight, domain-specific ontology (the node/relation types in §3).
- **A knowledge graph** is the populated network — the ontology plus all the instance data. It is optimised for browsing, linking and network analysis rather than heavyweight logical proof.
- **A relational database / spreadsheet** stores the same facts in tables. A KG is preferable when the questions are about *relationships and paths* (what co-occurs with what, what mitigates what) rather than row-by-row lookup.

In short: the ontology is the blueprint, the knowledge graph is the building, and a graph database (Neo4j) is the plot of land it stands on.

Why a graph for this material?

Historical recipe knowledge is intrinsically relational: a recipe links ingredients, a colour, a region, a date and a manuscript; a degradation phenomenon links the material that suffers it and the treatment that mitigates it. A graph makes those connections first-class, so questions such as *“which mitigations are reported against iron-gall ink corrosion, and in which records?”* become a short traversal rather than a manual cross-tabulation.

2. The data and its provenance

Every node and edge is traceable to **Ink-database.jsonl**, a structured extraction in which each line is one finding from a scholarly source. The 3,313 findings come from 300 distinct source files and split across three material categories:

Material category	Findings	Share
Ink	1,262	38.1%
Pigment	1,150	34.7%
Paper / support	901	27.2%
Total	3,313	100%

Each finding records (where available): the normalised material/ink/pigment/paper type, colour, ingredients, binder, chemical composition; the manuscript region, century and dynasty; the analytical technique(s) used; a confidence level and whether it was instrumentally verified; and free-text fields — the recipe, an evidence quote, remarks, a conservation/mitigation note and a preservation-state note — plus the citation and DOI.

Evidential status is preserved, not hidden

The graph does not flatten weak and strong evidence together. Each record keeps its confidence so claims can be filtered or qualified:

Instrumentally verified	Count	Certainty level	Count
Yes	830	Confirmed	1,078
No	2,227	Probable	298
Not stated	256	Inferred	1,733
		Disputed / not stated	2 / 202

This is important for publication: roughly a quarter of findings are instrumentally verified and about a third are 'Confirmed'; the rest are 'Probable' or 'Inferred'. The viewer surfaces this on every record so a reader never mistakes an inferred claim for a measured one.

3. The three graphs and their schema

The same material 'spine' (ink / pigment / paper concepts) is shared across three purpose-built graphs that cross-reference one another. A central design choice is the explicit separation of **data** from **metadata**.

Data vs. metadata

Data = the substantive content a researcher reasons about: materials, ingredients, colours, recipes, conservation methods, agents and degradation phenomena, plus the individual source **records**. **Metadata** = the context that situates a finding: region, century, dynasty, source paper and analytical technique. In the viewer, data are filled circles and metadata are dashed squares, and metadata can be hidden in one click so the content stands alone.

Node types

Type	Kind	Role
MaterialClass / MethodClass / PhenomenonClass	data	top of the class hierarchy
InkType, Pigment, PaperType	data	specific material concepts
Ingredient, Color	data	recipe components & appearance
ConservationMethod, ConservationAgent	data	how material is protected / with what
Degradation	data	decay phenomenon (e.g. ink corrosion)
Record	data	one source finding — full readable text
Region, Century, Dynasty	metadata	where / when / under whom
Source	metadata	the paper / citation
Technique	metadata	analytical method (XRF, Raman, FTIR…)

Relation types (edge glossary)

Relation	Reads as
IS_A	subclass → superclass (Iron-gall ink IS_A Ink)
USES_INGREDIENT / USES_BINDER	material/record → ingredient or binder
HAS_COLOR	material/record → colour
ABOUT	record → the material it concerns
FOUND_IN / DATED_TO / ASSOCIATED_WITH	→ region / century / dynasty
ANALYZED_BY	material → analytical technique
DOCUMENTED_IN	→ source paper
PROTECTED_BY / USES_AGENT	material → method; method → agent
EXHIBITS	material/record → degradation phenomenon
MITIGATES / MITIGATED_BY	method ↔ degradation (the cross-link between graphs)
APPLIES	record → the conservation method present in it

*Edges on aggregated concept links carry a **weight** = the number of records supporting them, so node size (degree) and link thickness are evidence-weighted, and clicking a link retrieves exactly the records behind it.*

4. Methodology — how the graphs were built

The pipeline is fully scripted and re-runnable (kg_common.py + build_graphs.py), so any figure here can be regenerated from the source file. Five steps:

4.1 Canonicalisation

The source 'normalised' fields still contained case variants, synonyms and chemistry-vs-common-name duplicates. These were mapped to single canonical concepts so that counts are meaningful — e.g. mercury sulfide → **Vermilion (cinnabar)**, lazurite → **Lapis lazuli (ultramarine)**, arsenic trisulfide → **Orpiment**, lead tetroxide → **Red lead (minium)**, crocin → **Saffron**, acacia gum → **Gum arabic**. Arabic terms are translated where a gloss exists and otherwise bucketed so they never pollute the high-level view.

4.2 Classifying free text into categories

The conservation and degradation fields are prose, not codes. A transparent keyword classifier (auditable in kg_common.py) assigns each note to one or more canonical categories — 15 conservation methods and 21 degradation phenomena. The distributions:

Conservation method	n	Degradation phenomenon	n
Protective coating / sizing	101	Stable / good condition	710
Lining / reinforcement / repair	100	Tearing / loss	201
Controlled storage / repository	86	Ink corrosion	127
Environmental control	59	General deterioration	121
Rebinding	59	Fading	101
Cleaning / purification	53	Discolouration	94
Deacidification / alkaline buffering	40	Abrasion / wear	80
Encapsulation / glazing	33	Fire / burn damage	71
Protective housing	26	Embrittlement	64
Consolidation / fixing	19	Moisture damage	53
Digitisation; Handling; Phytate; Antioxidant; Interleaving	12/9/4/3/2	Staining; Flaking; Acidification; Biodeterioration	48/32/28/25
		Offsetting; Smudging; Fragility; Cracking; Delamination; Abrasion; Foxing	23/21/20/13/12/10/8

Note: 'Stable / good condition' (710) is retained deliberately so well-preserved material is visible alongside damage, rather than silently dropped.

4.3 Islamic-tradition tagging

Each finding is flagged Islamic by keyword matching on region and dynasty (Egypt, Iran, Iraq, Anatolia, al-Andalus, the Maghreb, West Africa, India, Yemen…; Ottoman, Abbasid, Timurid, Safavid, Mamluk, Fatimid, Nasrid…). 2,474 of 3,313 findings (74.7%) qualify. The tag drives the optional 'dim non-Islamic context' emphasis; no data is discarded.

4.4 The record layer

Beyond aggregated concepts, every substantive finding becomes a **Record node** carrying its full text and links to all of its concepts. Records stay hidden in the canvas and are reached through the panel, which keeps the visual graph readable while making the underlying evidence one click away.

4.5 Reproducibility

Inputs (the JSONL), code (two Python files), and outputs (graph JSON, Neo4j CSV + Cypher, and the viewers) are all included. Re-running build_graphs.py regenerates every node, edge and figure deterministically.

5. How the graphs are used

Two complementary access modes:

Interactive viewers (browser)

- **Progressive disclosure:** each graph opens compact (a data-only core) and you double-click a node to reveal its neighbours, which are placed in open space — so the view never becomes a hairball.
- **Full-database search:** ranked search over every concept; selecting a result reveals and centres it.
- **Faceted record drill-down:** click a node to see its records *and* the next layers to narrow by; e.g. Ink corrosion (127 records) → narrow by mitigation → Deacidification → the 9 records that are both. A breadcrumb steps back up. Clicking a link shows the records behind that exact relationship.
- **Read in full:** a record opens its complete entry — ingredients, recipe, condition, conservation note, evidence quote, remarks, citation and DOI.
- **Focus controls:** hide metadata, filter by node type, set a minimum link strength.

Graph database (Neo4j)

Each theme ships nodes.csv, edges.csv and an upload.cypher script. The data/metadata split is stored on every node as a `kind` property, so analytical queries are direct, e.g. count substantive content, or list what mitigates a phenomenon:

```
MATCH (:KGNode {id:'deg:ink_corrosion'})-[:MITIGATED_BY]->(m) RETURN m.label, m.degree ORDER BY m.degree DESC;
```

6. Representative use cases

- **Comparative material history** — which pigments/inks appear in which regions and centuries, and how recipes cluster across the Islamic world.
- **Conservation decision support** — for a given degradation (e.g. iron-gall corrosion), the mitigations reported in the literature and the records that document them.
- **Literature synthesis & gap-finding** — sparsely connected concepts reveal under-studied materials or regions; densely cited ones show consensus.
- **Teaching & exhibition** — an explorable map of historical recipe knowledge with the primary evidence attached.
- **A reusable evidence base** — the Neo4j export supports custom queries beyond the viewer.

7. Anticipated questions (and answers)

Likely questions at review or presentation, with defensible answers.

Q. Is this a knowledge graph or just a network diagram?

A. It is a labelled property graph with a defined ontology — typed nodes in a class hierarchy and typed, weighted relationships — populated with instance data and queryable in Neo4j. The diagram is one view of it, not the thing itself.

Q. Where does the data come from, and can a claim be traced back?

A. Every node and edge derives from Ink-database.jsonl (3,313 findings, 300 papers). Each Record node retains its evidence quote, citation and DOI, so any displayed relationship can be traced to a specific source passage.

Q. How do you handle uncertain or unverified findings?

A. Confidence is preserved, not averaged away: each record keeps its certainty level (Confirmed/Probable/Inferred/Disputed) and an instrumentally-verified flag (830 verified). These are shown on every record and are queryable, so conclusions can be restricted to high-confidence evidence.

Q. The conservation/degradation categories come from keyword rules — how reliable are they?

A. The classifier is fully transparent and auditable (kg_common.py); it is deliberately simple and conservative. It is robust for the common, explicit cases and approximate at the margins. Crucially, the original free text is retained on every record, so a category is a navigational aid, never a replacement for the source wording — a reviewer can always check the note.

Q. Doesn't normalisation risk merging things that differ?

A. Mappings are explicit and documented (e.g. chemistry → named material). They were chosen to match standard conservation-science usage. Because the raw values are preserved on records, any merge can be inspected, and the pipeline can be re-run with revised mappings.

Q. How is 'Islamic' defined — isn't that a judgement call?

A. It is a heuristic on region and dynasty keywords, stated explicitly (74.7% of findings). It drives an optional emphasis only; nothing is excluded, and border cases remain visible and reclassifiable.

Q. Could the graph over-represent well-studied collections?

A. Yes — frequency reflects the literature, not absolute historical prevalence. This is stated as a limitation. Edge weights therefore indicate scholarly attention/co-occurrence, which is itself informative, but should not be read as historical proportion.

Q. What does the graph add over a spreadsheet of the same data?

A. Relationship-first questions become direct traversals: multi-hop paths (material → degradation → mitigation → agent), intersection queries (records that are both X and Y), and network measures (centrality, connectivity) that are awkward in tabular form.

Q. Is it reproducible and auditable?

A. Fully. Inputs, code and outputs are bundled; build_graphs.py regenerates everything deterministically, and the Neo4j CSV/Cypher exports let others load and re-query the graph.

Q. What are the main limitations?

A. It mirrors the source extraction (any upstream error propagates); keyword classification is approximate; coverage is uneven across regions/periods; and frequencies index the literature, not history. None of these undermine the graph's purpose as a traceable, explorable synthesis.

Companion files: index.html / conservation.html / degradation.html (interactive); kg/<theme>/ (graph JSON, Neo4j nodes.csv, edges.csv, upload.cypher); kg_common.py & build_graphs.py (pipeline); README.md.