

Get started · Learning paths

Path: data engineer

🕒 2 min read · Updated 25 September 2026

For work on the corpus, the database, the gates or a dataset release: from the line record to the pipeline, the editorial ledger and the pin that carries a dataset to production.

<https://docs.gurbanisoul.com/learning-paths/data-engineer/>

You will work in sggs-data, or take its releases here. Two days, in this order. The pipeline itself runs only in the data repository; this platform consumes its output by pin.

Interactive on the web: a checklist that remembers what you have done.

- ☐ Read the **reverence checklist**, then all of **Scripture 101** — the structure page with its poster, and Gurmukhi and Unicode with its quiz, most carefully.
- ☐ **Run it locally** — make dataset and make dataset-check.
- ☐ **Data & pipeline**: the overview, then **anatomy of a line record** with the Ang explorer.
- ☐ **Corpus pipeline and gates** — walk poster 04 step by step; read the fix_text and reconcile.py excerpts.
- ☐ **The editorial ledger**: the two kinds of transform, the eleven applications, how a concern travels.
- ☐ The pinned canonical documents: **scripture integrity**, the **database schema**, the **rebuild runbook**.
- ☐ **The dataset pin** — the lock, the fetcher's guarantees, the lock-bump flow.
- ☐ **Exercise 03 — bump a pin in a scratch branch**.
- ☐ **The Roman fold** — because the index-time fold is yours.
- ☐ **Harnesses and golden vectors** — what a dataset change must not break, and how make contract proves it.
- ☐ **The Answer Protocol for engineers**.
- ☐ **Exercise 04 — write an ADR** for a change to the data model you would propose.
- ☐ **Engineering invariants**, the data bullets in particular.

When you are done you can say what reconcile.py proves and what it does not, why a rebuild never edits the committed corpus until two gates pass, and what a scholar sees before any repair reaches a reader.