

Data

Anatomy of a line record

🕒 4 min read · Updated 27 September 2026 · Verified against felff7d on 2026-09-25

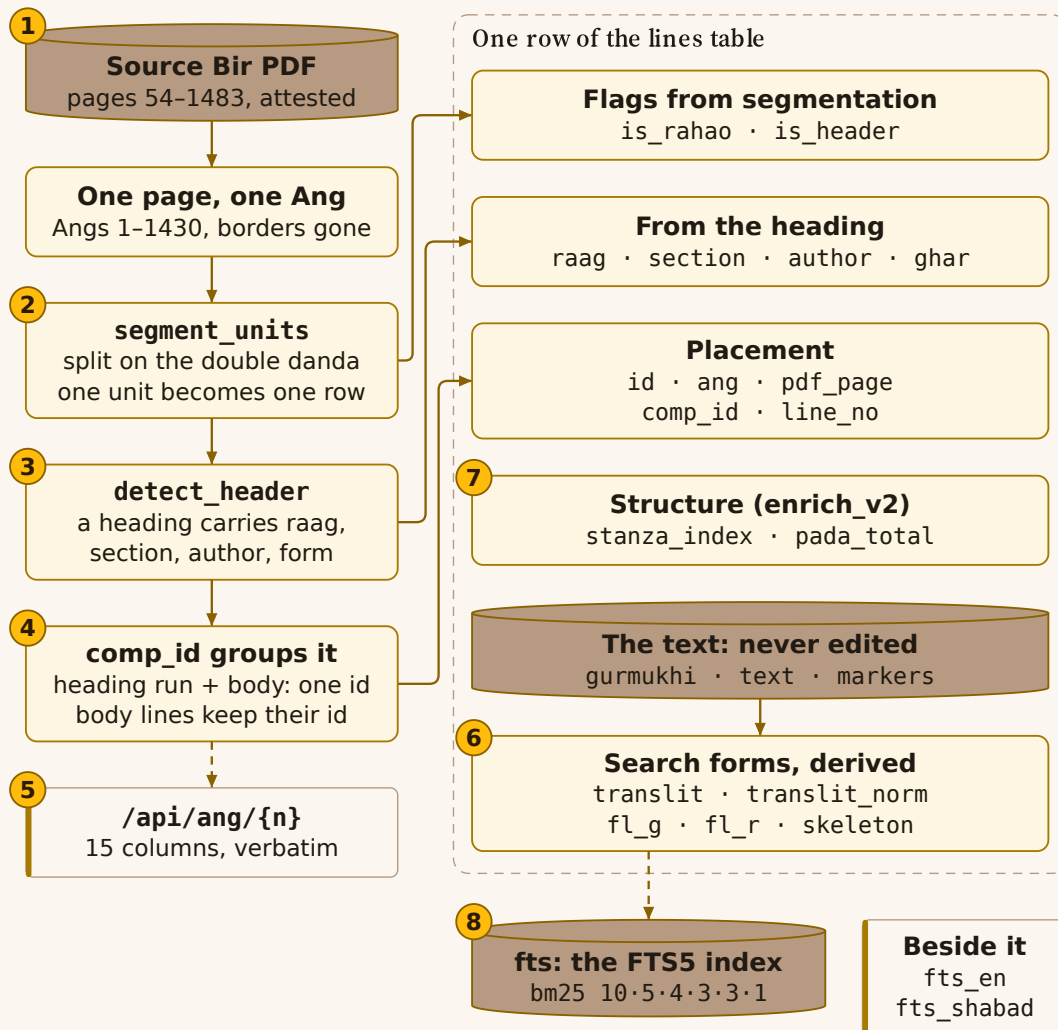
One display line of the Granth is one row of the lines table. What its 23 columns mean, how a printed page becomes rows, which columns search reads — with a live explorer over the real API.

<https://docs.gurbanisoul.com/data/line-record/>

Every unit the source prints between two double dandas becomes **one row** of the lines table. The poster follows one page from the PDF to that row; the explorer below it shows real rows, verbatim, straight from the API.

03 · Anatomy of a line record

How a page of the source Bir becomes rows of the lines table — one display line per row — and which columns search reads



Source of truth: sggs-data/pipeline/build_corpus.py ·
sggs-data/pipeline/sggs_pipeline.py · webapp/sggs/search.py ·
webapp/sggs/core.py

v1.3.10 · verified 2026-09-27 · 017091b

Sri Guru Granth Sahib Ji — Knowledge Base · engineering wiki

Poster 03 — Anatomy of a line record: from a PDF page to units to the 23 columns of the lines table and the FTS index

- 1. A page is an Ang.** The pipeline opens the attested source PDF and reads pages 54 to 1483. page_content strips the page borders and the page number, joins the wrapped lines, and maps each PDF page to its Ang: 1 to 1430. Both numbers are kept on every row (pdf_page, ang).
- 2. A page becomes units.** segment_units splits the page stream on the double danda. Numeral markers stay attached to the unit they close, and a rahao marker sets the rahao flag. One unit becomes one row: is_rahao is 1 on a refrain line, is_header is 1 on a heading.

3. **Headings carry the metadata.** `detect_header` decides whether a unit is a structural heading — a raag title, the invocation, a Mahalla or Bhagat label — and reads the raag, section, author, ghar and composition type (`comp_type`) from it. Those values are inherited by the lines that follow. A verse that merely contains a raag word or a form word is not a heading (the v1.1.4 weak-signal rule). Japji has no Mahalla line in the `print`, so its author is null.
4. **comp_id groups a composition.** A run of consecutive heading lines opens one composition together with the body that follows, so `/api/shabad/{comp_id}` returns the printed title. The run adopts its last heading's id, so no body line ever changes `comp_id` and vacated ids become permanent gaps. `line_no` counts 1..N inside the composition; id is the row's identity for ever.
5. **The text, three ways.** `gurmukhi` is the line as printed, dandas and markers included — verbatim, proven character for character against the PDF, and never edited anywhere in the system. `text` is the same line without markers, for indexing. `markers` is the JSON list of the numerals that closed the unit. `/api/ang/{n}` returns 15 of the columns for every line, `gurmukhi` verbatim, never a rewritten form.
6. **Search forms are derived.** From the text the pipeline derives `translit` (Roman transliteration), `translit_norm` (the spelling-tolerant fold, built in the database), `fl_g` and `fl_r` (first letters in Gurmukhi and Roman, for first-letter search) and `skeleton` (the line with its vowel signs stripped). They are reading aids and search keys, not scripture.
7. **Structure columns.** `enrich_v2` adds `stanza_index`, `pada_total` and `source_category` — structural metadata the reader and the app use for layout — without touching any text column. `comp_type` is known to be mislabelled in places and is suppressed at the display layer; prefer `comp_id` and section.
8. **The FTS index.** `fts` is an FTS5 table with external content over lines: `text`, `translit`, `translit_norm`, `fl_g`, `fl_r` and `skeleton`, tokenised so Gurmukhi signs stay inside tokens. Search ranks with `bm25` and column weights 10, 5, 4, 3, 3, 1 — the verbatim text outranks a transliteration hit, which outranks a first-letter or skeleton hit. Beside it, `fts_en` serves the English tier and `fts_shabad` the passage tier; translations live in their own, labelled table. (The database also carries a trigram index, `fts_tri`, which no route reads.)

Explore real records

Interactive on the web: an Ang, read live from the API.

Try the quick picks. Ang 1 opens with a heading and then Japji, whose author is null because the `print` has no

ਮਹਲਾ

line. Ang 151 shows a raag title and its invocation sharing one heading run. Ang 462 is where Asa Ki Vaar begins, saloks and pauris interleaving. Ang 712's first composition began on Ang 711 (`continued_from`). Ang 1256 prints one refrain twice — a legitimate repeat that the pipeline must never de-duplicate.

The 23 columns

The table is created by `build_db.py` in `sggs-data`; the platform reads it read-only.

Group	Column	Meaning
Placement	<code>id</code>	The row's identity across the corpus (1–60,658); never reused.
	<code>ang</code>	The Ang, 1–1430.

Group	Column	Meaning
	pdf_page	The page of the source PDF (54–1483) the line was read from.
	comp_id	Groups every line of one composition, heading run included; vacated ids are permanent gaps.
	line_no	Position inside the composition, 1..N, contiguous.
From the heading	raag	The musical mode; <code>null</code> after Ang 1353, where the Granth leaves the raag framework.
	section	The bani or section, header-detected.
	author	From the ਮਹਲਾ or Bhagat label; <code>null</code> where the print has none (all of Japji).
	ghar	The musical “house” (ਘਰ) of the composition.
	comp_type	The form word on the heading; known to be mislabelled in places, suppressed at the display layer — prefer <code>comp_id</code> and <code>section</code> .
	is_header	1 on a heading line: a title, the ੴ invocation, a ਮ: label.
Flags	is_rahao	1 on the refrain line, from its ਰਹਾਉ marker.
	markers	The numerals that closed the unit, as a JSON list (for example [" ੴ "]).
The text	gurmukhi	The line as printed, verbatim, dandas and markers included. Never edited anywhere.
	text	The same line without its markers, for indexing.
Search forms	translit	Roman transliteration — a reading aid, not scripture.
	translit_norm	The spelling-tolerant fold of <code>translit (roman_norm)</code> , built in the database, not in the corpus file.
	fl_g / fl_r	First letters of each word, Gurmukhi and Roman, for first-letter search.
	skeleton	The line with its vowel signs stripped, for the typotier.
	stanza_index	Which stanza of its composition the line belongs to.
Structure (v2)	pada_total	How many padas the composition has.

Group	Column	Meaning
	source_category	A coarse category of the source unit (for layout).

The API's `LINE_COLS` returns 15 of them on every line (`/api/ang`, `/api/shabad`, search results), always including gurmukhi:

```
webapp/siggs/core.py · LINE_COLS · lines 57–59

LINE_COLS = ('id, ang, raag, section, author, comp_type, comp_id, line_no, '
             'is_rahao, is_header, gurmukhi, translit, '
             'stanza_index, pada_total, source_category') # v2.0 structural metadata
```

Read from the source at build time — [view these lines on GitHub at c3d9360](#).

Source: [webapp/siggs/core.py](#)³, the `LINE_COLS` tuple.

Headings, compositions and the `comp_id` rule

A heading is detected by `detect_header` from the unit's text — a raag title, the invocation, a

ਮਹਲਾ

or Bhagat label — and the lines that follow inherit its metadata. **A run of consecutive heading lines opens one composition together with the body that follows** (post-pass 1b in `build_corpus.py`), so `/api/shabad/{comp_id}` returns the printed title. The run adopts its *last* heading's id, so no body line ever changes `comp_id` and vacated ids become permanent gaps: 4,527 distinct compositions and `max(comp_id)` 5,376 in the pinned dataset, `comp_id` 1 a gap because the Mool Mantar folds into Japji (`comp_id` 2). The full model, the before/after diagram and the invariants are in the canonical [database schema](#).

Three closing rubrics (`TRAILING_RUBRICS`) belong to the *preceding* unit and stay one-line compositions, flagged for scholarly review; a verse that merely contains a raag or form word is **not** a heading (the v1.1.4 weak-signal rule, which demoted 233 mis-flagged verses).

What search reads

`fts` is an `FTS5` table with external content over lines, covering text, translit, translit_norm, `fl_g`, `fl_r` and skeleton, tokenised so [Gurmukhi](#) signs and

ੴ

stay inside tokens. `do_search` ranks with `bm25` and column weights **10 · 5 · 4 · 3 · 3 · 1** in that order: a hit in the verbatim text outranks a transliteration hit, which outranks a first-letter or skeleton hit. Beside it, `fts_en` serves the English tier and `fts_shabad` cross-line passages; the database also carries a trigram index, `fts_tri`, which no route reads. How the tiers cascade is the [search waterfall](#).

The English translations (Dr. Sant Singh Khalsa via ShabadOS) are a **separate table**, attached to a line as en by the API and labelled as a translation everywhere; they are never blended into the Gurmukhi.

Read next: [Corpus pipeline and gates](#).

Links

1. <https://gurbanisoul.com/reader?ang=1>
2. <https://github.com/Algorythmos-AI/siggs-platform/blob/c3d936031f0c283283c5d1102705affe1f8cf404/webapp/siggs/core.py#L57-L59>
3. <https://github.com/Algorythmos-AI/siggs-platform/blob/integration/webapp/siggs/core.py>

Sri Guru Granth Sahib Ji — Knowledge Base wiki. Generated from commit c3d9360. Updated 27 September 2026. Verified against fe1ff7d on 2026-09-25.
This copy may be out of date; the page is at <https://docs.gurbanisoul.com/data/line-record/>