

Claude Code for Accounting Research

Module 3 — EDGAR Text: Structured Data, and Your First Skill

John Barrios

Yale School of Management

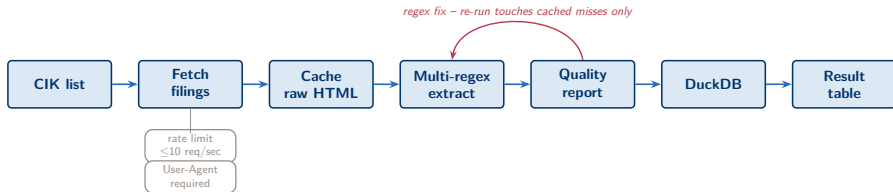
Today's Arc

- **0:00–0:03** Checkpoint follow-up: Duo triage results from Module 2
- **0:03–0:40** Lecture: EDGAR mechanics, text pipeline, measurement validity, skills
- **0:40–1:15** Live demo: two acts — EDGAR pipeline, then SKILL.md live-build
- **1:15–1:25** Break
- **1:25–2:45** Lab: pipeline → result table → your first skill
- **2:45–3:05** Debrief, exit ticket, bridge to Module 4

Two Halves, One Lesson

- Text-as-data: where AI meets accounting most naturally — and most dangerously
- Half one: build the pipeline, CIK → fetch → extract → DuckDB → measure
- Half two: package the SOP you just suffered through as a `SKILL.md`
- Both halves cash in Module 1's mantra — files persist, context doesn't

EDGAR Mechanics



- CIK: stable firm ID, assigned once, never reused
- `efts.sec.gov` full-text search + `data.sec.gov/submissions` per-filer endpoint
- Hard limits: 10 req/sec, mandatory User-Agent (name + email)

Vague vs. Front-Loaded

Vague:

"get me the risk factors from some 10-Ks"

Front-loaded:

"Fetch Item 1A from the 10-Ks filed by the CIKs in pinned_ciks.csv, FY2021--2023, via EDGAR full-text search + submissions endpoints. 10 req/sec, User-Agent set. Cache raw HTML to data/filings/."

Same task. The difference is whether facts you already know are in the prompt, or discovered by trial and error.

10-K Anatomy: Why One Regex Fails

- Item 1A / Item 7 — nominally one heading each; filers do not cooperate
- Four ways to write “Item 1A”: caps, trailing period, split TOC anchor, cross-reference decoy
- Fix: an ordered fallback chain, not a cleverer single pattern
- Log which pattern fired for each filing — “fail silently” is the real danger

Pipeline Robustness

- Cache raw HTML on first download — extraction becomes a pure function over local files
- Fix a regex, re-run: zero new network requests if caching worked
- Quality report every pass: successes, failures, suspiciously-short flags
- Bluntest useful prompt: *ask Claude why N filings are missing*

Common Failure

Treating the quality-report number as the deliverable

“99% extraction” means nothing without the denominator. A pipeline that silently drops the hardest 1% has a selection problem if those filings differ systematically from the ones that extracted cleanly.

DuckDB as the Research Store

- Single file, zero configuration, reads/writes from Python, R, or the CLI
- Three tables: `filings`, `sections`, `keywords`
- Plain English to SQL — but read the query before you trust the numbers
- Replaces a folder of forty CSVs with one file any coauthor can open

Keyword Counts $\neq$ Constructs

- “Tariff” mentioned 14 times, up from 6 — rising exposure, or a boilerplate update?
- A keyword count measures word frequency; “exposure” is a construct about the firm
- The gap between the two is construct validity — the most common way a disclosure measure survives peer review while being wrong

Boilerplate and Year-over-Year Similarity

- Filers reuse risk-factor language year over year, far more than “disclosure” suggests
- Detectable signal: YoY textual similarity (Jaccard / cosine)
- High similarity + rising keyword count $\approx$ boilerplate inflation
- Low similarity + rising keyword count $\approx$ a real, rewritten disclosure
- Diagnostics, not proof — they don't replace the audit sample

The Human Audit Sample

- **Stratified, not convenience-drawn** — high/mid/low measure, size, industry; ~50 filings
- **Rubric decided in advance** — coded before reading, not adjusted filing by filing
- **Inter-rater agreement reported** — or state plainly that only one person coded it

Verify This

The protocol is the deliverable, not the measure

Fifty filings can't prove a measure valid. What you owe a reader is the protocol you'd run before trusting it — which fifty, stratified how, coded against what rubric, by whom.

Skills Preview: Your SOP, Packaged

- Everything painful this morning — front-loading, fallback chain, caching, validity check — is a standard operating procedure
- “Show Other People”: a skill turns something re-explained every session into something a coauthor just runs
- You already wrote today’s skill; you just don’t know it yet

Anatomy of a SKILL.md

.claude/skills/edgar-panel/SKILL.md

frontmatter

name: edgar-panel

description: Use when the user says "scrape 10-Ks", names tickers and a form type, and wants a firm-year panel...

TRIGGER
condition,
not a title

body: steps

1. Resolve CIKs from tickers
2. Fetch & cache filing index (rate-limited)
3. Extract sections (multi-regex fallback)
4. Load into DuckDB; write quality report
5. Emit firm-year result table

resources

reference/cik_list.csv
reference/regex_patterns.md

X wrong: description as a title

description: EDGAR filing tool

never fires – nothing in the user's message matches a "title"

✓ right: description as a trigger

description: Use when the user says "scrape 10-Ks", names tickers and a form type...

fires because it matches how the user actually asks

Frontmatter (name, description), a numbered procedure, pointers to supporting files.

Description as Trigger, Not Title

Wrong — reads as correct, never fires:

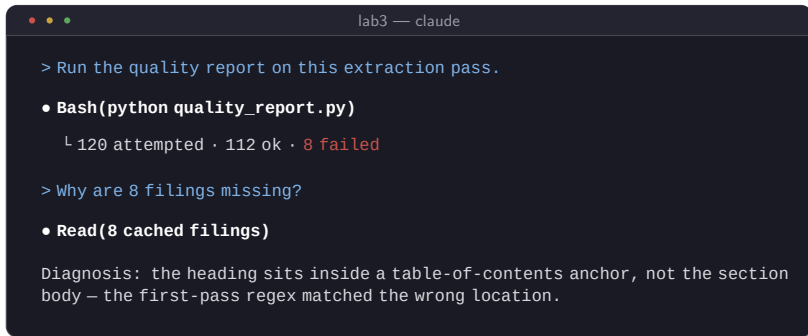
description: EDGAR filing tool

Right — names the situations a real request looks like:

description: Use when the user asks to scrape 10-K filings, build an EDGAR text dataset, or names tickers/CIKs alongside a form type and date range.

Same procedure underneath. A title describes the skill; a trigger is what matching logic can actually recognize.

What You'll See: the Quality Report



```
lab3 — claude

> Run the quality report on this extraction pass.

• Bash(python quality_report.py)

  ↳ 120 attempted · 112 ok · 8 failed

> Why are 8 filings missing?

• Read(8 cached filings)

Diagnosis: the heading sits inside a table-of-contents anchor, not the section
body — the first-pass regex matched the wrong location.
```

Stylized session transcript (not a screenshot).

Read what the quality report flagged — don't just re-run the pipeline.

Successes, failures, and the suspiciously-short flags that turn a silent extraction gap into a diagnosable one.

Next: Demo Act 1 --- EDGAR

0:40–1:00, compressed

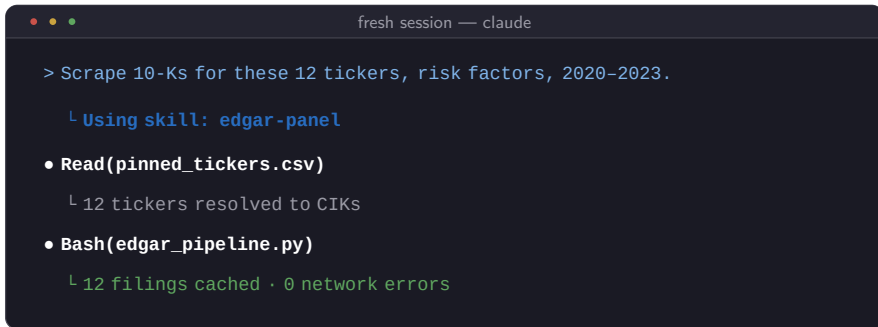
- Contrast opener: vague vs. front-loaded — count the rounds (5 min)
- Plan Mode for real: full plan → `plan.md` → clear context → execute (4 min)
- Phases 1–2 live: CIK resolution + filing index, ~8 firms (5 min)
- Phases 3–4 narrated watch: quality report, regex fix, re-run, 95 → 99 (6 min)

Next: Demo Act 2 --- SKILL.md

1:00–1:15, live-build

- Five-step workflow, run live: notice → name inputs → design outputs → draft SKILL.md → test-fire (10 min)
- Deliberate error: description as a title, shown not to fire; rewritten as a trigger, fires (5 min)

A Skill, Firing



```
fresh session — claude

> Scrape 10-Ks for these 12 tickers, risk factors, 2020-2023.

  ↳ Using skill: edgar-panel

  • Read(pinned_tickers.csv)
    ↳ 12 tickers resolved to CIKs

  • Bash(edgar_pipeline.py)
    ↳ 12 filings cached · 0 network errors
```

Stylized session transcript (not a screenshot).

A fresh session, no re-explaining: the description field fires on the request's own words.

A fresh session recognizes the trigger-condition description and runs the packaged procedure without re-explanation.

Next: The Lab (1:25--2:45)

- **Part A (45 min):** pipeline — ~ 20 firms $\times$ 3 years $\rightarrow$ DuckDB, one keyword topic
- **Part B (15 min):** validity — YoY similarity classification + 3-sentence audit protocol
- **Part C (20 min):** the skill — build `edgar-panel/SKILL.md`, test-fire fresh, install a catalog skill

Lab Verify Checklist

- ☐ Cache works — re-running downloads zero already-fetched filings
- ☐ Quality report: extraction rate stated; every failure has a one-line diagnosis
- ☐ Result table reproduced by named scripts from README
- ☐ Validity paragraph + audit-sample protocol written
- ☐ SKILL.md description is a trigger, and it fires in a fresh session
- ☐ Rate limit respected, User-Agent set — show the code line

Debrief and Bridge to Module 4

- Extraction-rate leaderboard; genuine edge cases vs. fixable regex gaps
- Read 2–3 student skill descriptions aloud — would they trigger?
- Bridge: your text panel has CIKs; fundamentals live in Compustat, keyed by gvkey — in Module 4, the crosswalk, via an MCP, on your own WRDS account
- Exit ticket: standard 3 + “what would your second skill be?”