BA TOOLKIT · FREE TEMPLATE

AI data dictionary, evidence first

Template, profiling query, prompt and blind test: every definition points at evidence from the data and ends with a named owner signing it.

By **Prachit Parikh**, business analyst · github.com/prachitparikh/ba-toolkit

An AI can draft a data dictionary in seconds. The catch is that it reads column names, and column names drift away from the data. This kit makes every definition point at evidence from the data itself, turns each definition into a check you can run, and ends with a human owner signing the row.

Example data below is fictional (client: Northstar Systems).

1. The template

One row per column. The Evidence and Test columns are the point: a definition with neither is still a guess.

Column	Definition	Source of definition	Evidence (from the profile)	Test (one line of SQL)	Test result	Open question	Owner	Signed (name, date)
amount	Order total in cents (divide by 100 for dollars)	Profile + Finance	max 4,850,000; never negative; whole numbers only	`SELECT MAX(amount) FROM orders;` expect under 5,000,000 (cents)	pass	none	Finance data owner	(name), (date)
status	Order status code: 1, 2, 3 or 9	Profile	4 distinct values	`SELECT COUNT(DISTINCT status) FROM orders;` expect 4	pass	What does 9 mean?	Order ops owner	
created_at	Order timestamp, stored in UTC	Profile	hourly order count peaks at 17:00	hour histogram, see section 4	pass	Is any source system writing local time?	Data engineering	

Column	Definition	Source of definition	Evidence (from the profile)	Test (one line of SQL)	Test result	Open question	Owner	Signed (name, date)
region	Sales region code; 'ZZ' = missing	Profile	12% of rows = 'ZZ'	`SELECT AVG(CASE WHEN region = 'ZZ' THEN 1.0 ELSE 0 END) FROM orders;`	0.12	Should 'ZZ' be NULL instead?	Sales ops	

"Source of definition" takes one of three values: **Name only** (a guess), **Profile** (the data supports it), **Owner confirmed** (a person said yes). Only "Owner confirmed" rows with a signature go in the published dictionary.

---

2. The profiling query (run it before the AI sees anything)

Standard SQL. Swap `orders` and `amount` for your table and column, and run it once per column.

```
SELECT
  COUNT(*)                                          AS row_count,
  COUNT(DISTINCT amount)                            AS distinct_values,
  MIN(amount)                                       AS min_value,
  MAX(amount)                                       AS max_value,
  AVG(CASE WHEN amount IS NULL THEN 1.0 ELSE 0 END)   AS pct_null
FROM orders;

-- top values (useful for codes like status or region)
SELECT amount, COUNT(*) AS n
FROM orders
GROUP BY amount
ORDER BY n DESC
LIMIT 10;

-- five sample rows
SELECT * FROM orders LIMIT 5;
```

Paste the results into the prompt next to the schema.

3. The prompt

```
You are drafting a data dictionary. Below are the schema and a profile of each column
(distinct count, min, max, percent null, top values, five sample rows).

Rules:
1. Describe each column using only the profile.
2. For every claim, cite the profile number that supports it.
3. If the profile can't tell you something (meaning of a code, unit, time zone),
   write UNKNOWN and give one question for the data owner.
4. For each definition, write one SQL check that would fail if the definition were wrong.

Output a table: column | definition | evidence | SQL check | question for owner
```

4. Turn each definition into a test

Definition says	Check	Fails when
amount is in dollars	`MAX(amount)` against the largest order the business allows	max is about 100x too big (cents)
status has 4 codes	`COUNT(DISTINCT status)`	a fifth code shows up
created_at is Eastern time	orders by hour (below)	the lunch rush lands at 17:00, which points to UTC
region is always a real region	share of placeholder values	'ZZ', 'N/A' or blanks above 0%

```
-- orders by hour of day (syntax varies: EXTRACT(HOUR FROM ...), DATEPART(hour, ...),
strftime('%H', ...))
SELECT EXTRACT(HOUR FROM created_at) AS hr, COUNT(*) AS orders
FROM orders
GROUP BY 1
ORDER BY 1;
```

A definition you can't test is an opinion. Mark it as such in the template.

5. The blind test

1. Copy the profile and replace every column name with col_1, col_2, col_3...
2. Ask the AI to describe each column from the profile alone.
3. Put the blind description next to the column name.
4. Wherever they disagree, the name is the suspect. That disagreement becomes a question for the owner.

Example: col_1 came back as "whole numbers, never negative, up to almost five million, looks like money stored in cents". The column is named `amount` and the draft said dollars. One question to Finance settled it before any dashboard used it.

6. Sign-off

- Every published row has an owner, a name and a date.
- Sensitive data (PII) classification stays a human decision. Databricks' own documentation on AI-generated comments says the model "should not be relied on for data classification tasks such as detecting columns with PII."

Sources

- Databricks, "Add AI-generated comments to Unity Catalog objects" (docs, updated 11 Sep 2026): https://docs.databricks.com/aws/en/comments/ai-comments
- Google Cloud blog, "No metadata? No problem, with AI and Dataplex Universal Catalog" (4 Dec 2025): https://cloud.google.com/blog/products/data-analytics/generate-metadata-automatically-in-google-data-cloud
- Singh, Kumar, Donaparthi, Karambelkar, "Leveraging Retrieval Augmented Generative LLMs For Automated Metadata Description Generation to Enhance Data Catalogs" (arXiv 2503.09003, March 2025): https://arxiv.org/abs/2503.09003

ABOUT THE AUTHOR

Prachit Parikh is a business analyst in Apex, NC. He turns what stakeholders ask for into requirements a team can test, checks the data with SQL, and sees changes through UAT. More free templates: github.com/prachitparikh/ba-toolkit. Videos: youtube.com/@PrachitParikh1. Website and portfolio: prachitparikh.github.io.