20 AI words for data people

Plain English, with a data example for each.

Neurobuttr
by Ishan Yash

BASICS

Large language model (LLM): a model trained on huge amounts of text to predict what comes next, one token at a time. Think of your SQL editor's autocomplete, scaled up from column names to whole paragraphs and queries.

Token: a chunk of text, often part of a word. Pricing and limits are counted in tokens, much as BigQuery bills by bytes scanned, so a pasted 10,000-row CSV can cost more than you expect.

Context window: how much text the model can hold in mind at once, prompt and answer together. Think of it as working memory: past the limit, rows get cut or the call fails.

Prompt: the text you send the model, including any data you paste in. Write it like a good ticket: table names, grain, what 'active customer' means and the output you want.

System prompt: standing instructions the model reads before your message, usually set by whoever built the tool. Like `dbt_project.yml`: you rarely open it, but it sets the defaults for every run.

Temperature: a setting that controls how random the output is. Low gives steadier, more repeatable answers, good for SQL; high gives more variety, good for naming dashboards.

WHEN IT GOES WRONG

Hallucination: when the model states something false with confidence, such as a column or table that does not exist. Check generated SQL against the real schema before you run it.

Guardrails: checks around a model that block unwanted inputs or outputs, such as personal data in a reply. Like dbt tests: they catch the problems you planned for, not every problem.

Eval (evaluation): a repeatable test that scores model output against examples with known answers. Like a regression suite: run the same 50 questions before and after each change.

WORKING WITH YOUR DATA

Embedding: a list of numbers that represents the meaning of a piece of text, so similar ideas sit close together. It can match 'refund not received' to 'still waiting for my money', which `LIKE '%refund%'` would miss.

Vector database: a store that searches embeddings by similarity, not exact match. Where SQL asks `WHERE name = 'X'`, it finds the rows closest in meaning. Postgres can do this with pgvector.

Retrieval-augmented generation (RAG): the system finds relevant documents, such as your data dictionary, and adds them to the prompt. Like a join at query time: fresh facts, no retraining, though it can still misread them.

Fine-tuning: further training a model on your own examples, such as questions paired with your team's SQL. It suits format and style more than facts; for figures that change, use RAG.

Inference: running a trained model to get an answer, as opposed to training it. If training is building the warehouse, inference is every query you run on it, and the source of per-token bills and waiting time.

DOING THINGS

Tool use (function calling): the model asks your code to run a named function, such as `run_query(sql)`, then uses the result. The model only writes the request; your code decides whether it runs and with what permissions.

Agent: a model running in a loop, picking a tool, checking the result and choosing the next step until the job is done. Like an Airflow DAG that writes itself as it runs, so keep its permissions narrow.

Model Context Protocol (MCP): an open standard for connecting AI apps to tools and data through one common interface. Like ODBC for AI: build one MCP server for your warehouse and any compatible assistant can use it.

Multimodal: a model that can take in more than text, such as images, audio or video. Hand it a dashboard screenshot and ask what changed, but check any number it reads off a chart.

Open-weights model: a model whose trained weights are published, so you can run it on your own servers and keep data in-house. Not the same as fully open source: the training data and code are often kept private.

Text-to-SQL: turning a plain-English question, such as 'revenue by region last quarter', into a SQL query. It works far better when the model can see your schema and metric definitions; read the query before trusting the number.