

IDUG

2026

São Paulo | August 17 - 19

BRAZIL Db2 TECH CONFERENCE

Db2 is Your AI Platform:
Proven AI Use Cases to Build Right Now
Shaikh Quader, *IBM*

Session Code: B11



Platform:

Db2 for LUW

From what a vector is, to code you can clone

No AI background assumed. Everything you see runs inside Db2, called from SQL.

1

The basics

vectors, and the two kinds of search

2

What Db2 gives you

vectors, indexes, models, text search

3

The demo

one search fails, then one works

4

Your turn

the same recipe, on your own data

WHAT YOU LEAVE WITH

✓ the vocabulary

vector, embedding, BM25 — all defined

✓ the judgement

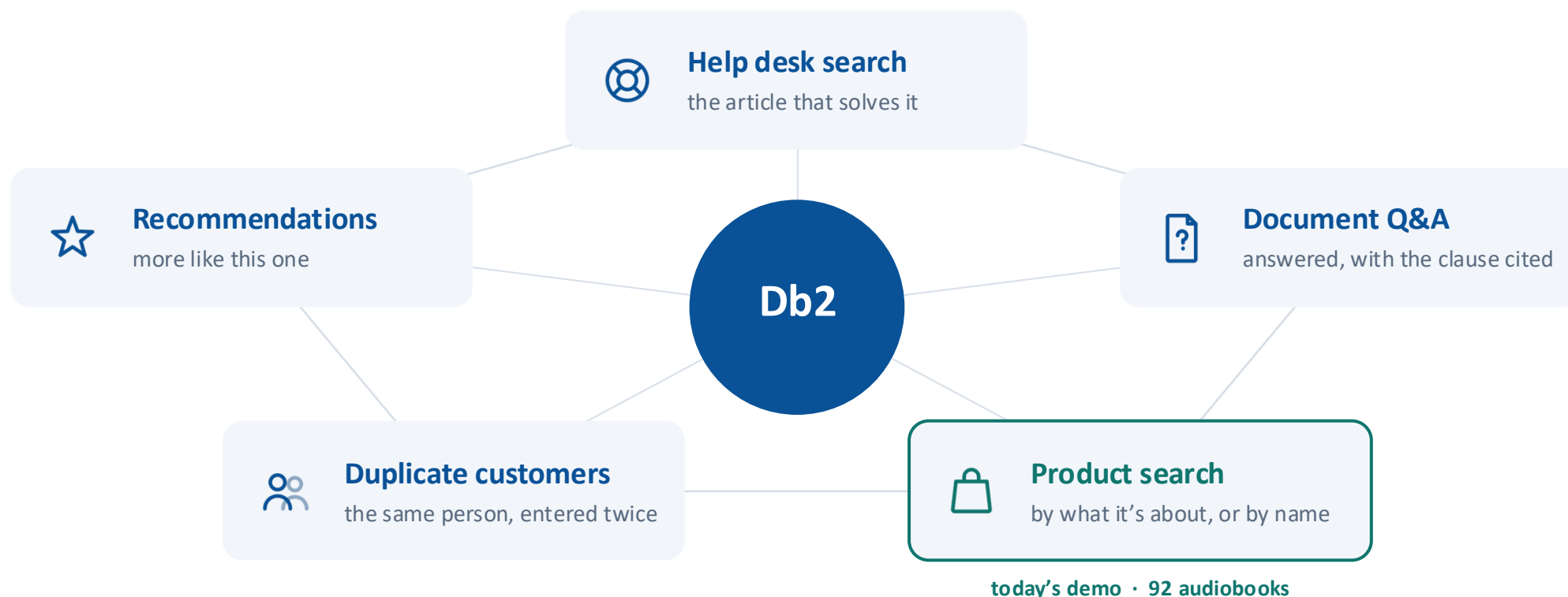
which technique fits which question

✓ the code

a repo that runs offline, on a bare box

Five things you can build on data you already have in Db2

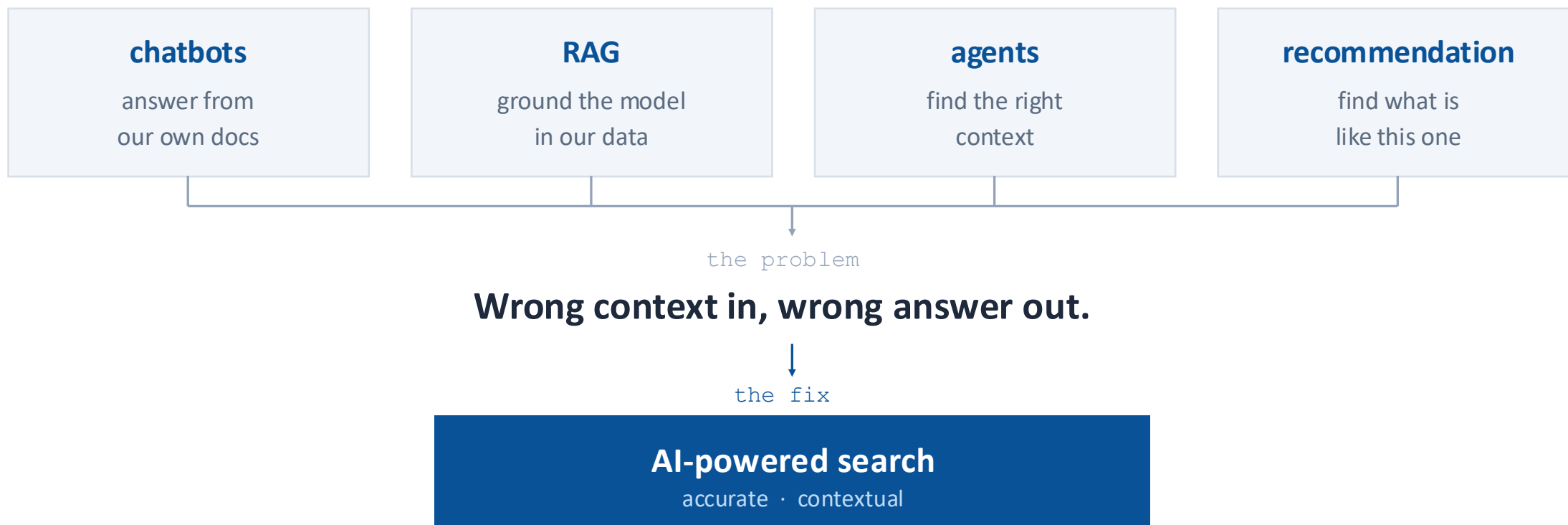
One session isn't enough for five. We'll go deep on one.



Every one of these is **retrieval** with a different job on the end.

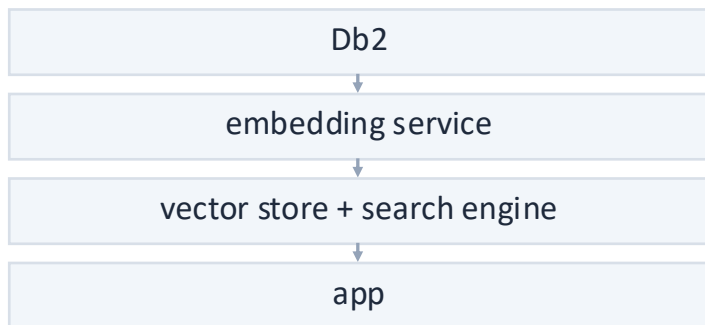
Retrieval is hard

Different apps. Same failure mode.



Retrieval infrastructure is where the cost shows up

Moving parts: complex infrastructure



cost	another system to license, run and staff
latency	every hop is a network hop
complexity	two security models, two backups, data drifts

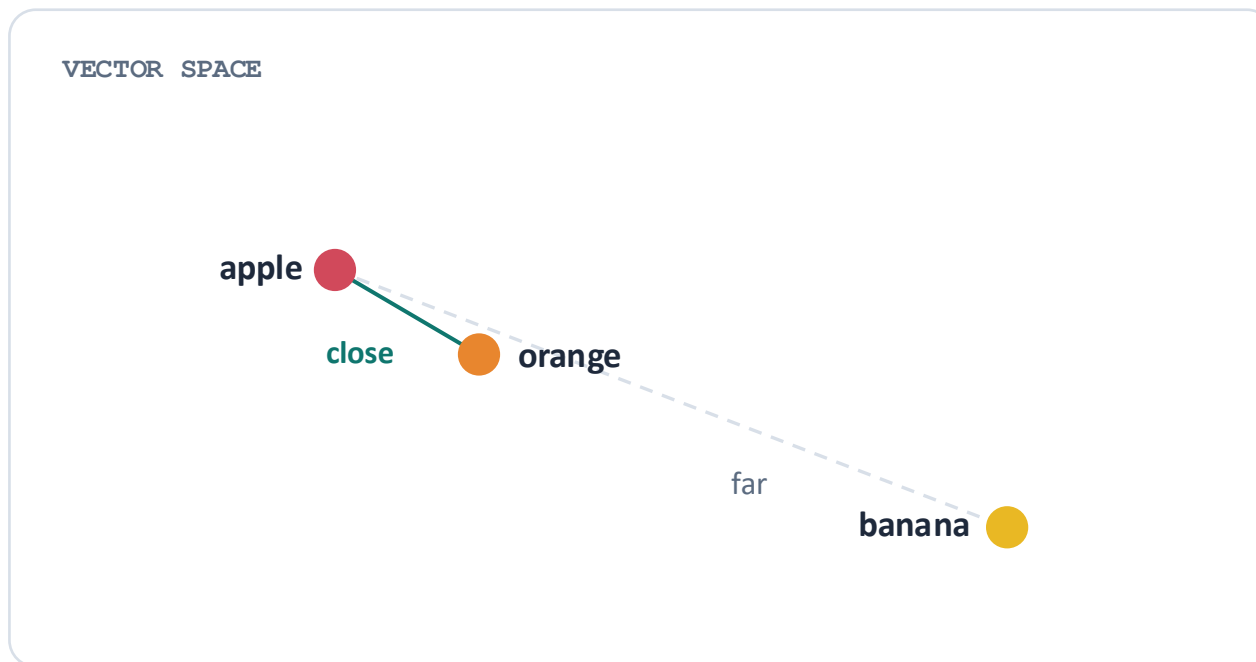
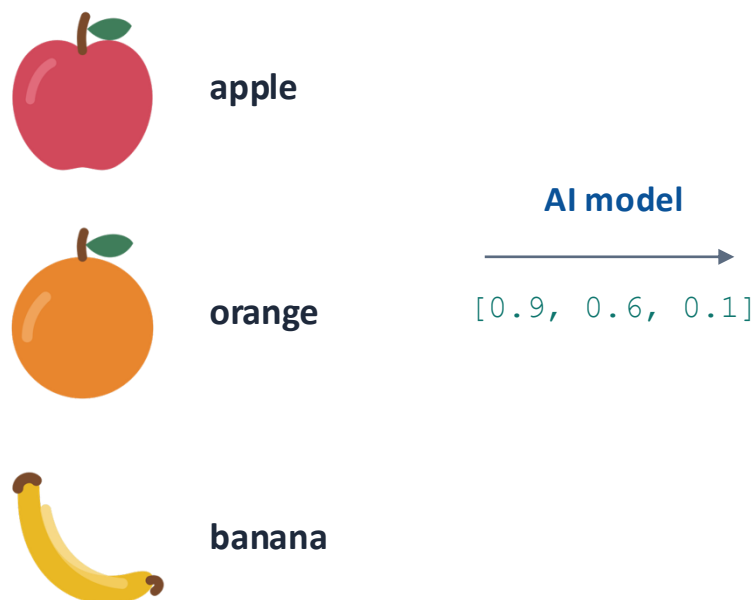
Db2: simpler infrastructure



cost	reduced — fewer systems to license and run
latency	cut down — data and AI on one platform
complexity	unified — one enterprise security and backup model

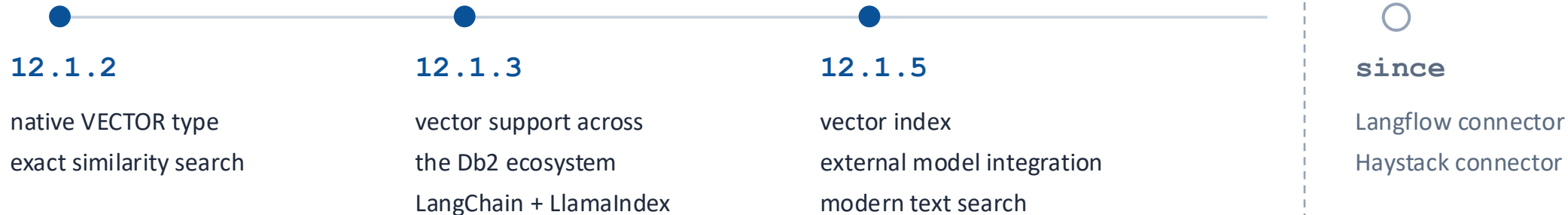
What is a vector?

Vectors are numerical representations of real-world entities and concepts.



AI models learn to project entities so that similar ones land close together.

Db2's AI foundation arrived in stages



Vectors became a column type, not a separate store

chunk_id INTEGER	title VARCHAR	narrators VARCHAR	embedding VECTOR(384)
66	Atomic Ha...	Walter Di...	[0.021, ...]
91	Slow Produ...	Cal Newp...	[-0.118, ...]
36	Thinking, ...	Patrick E...	[0.443, ...]

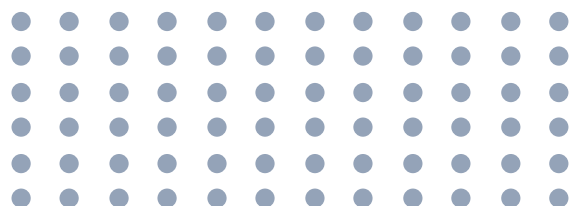
your business columns
still filter, join, index

the same row's
vector, in place

Filter on ordinary columns and rank by similarity — in the same statement.

Exact search reads every row. An index walks a graph.

EXACT

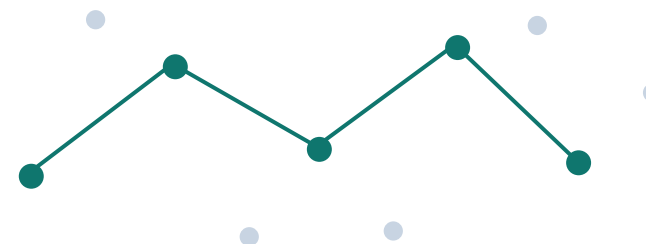


every vector scanned

perfect recall

cost grows with the table

APPROXIMATE

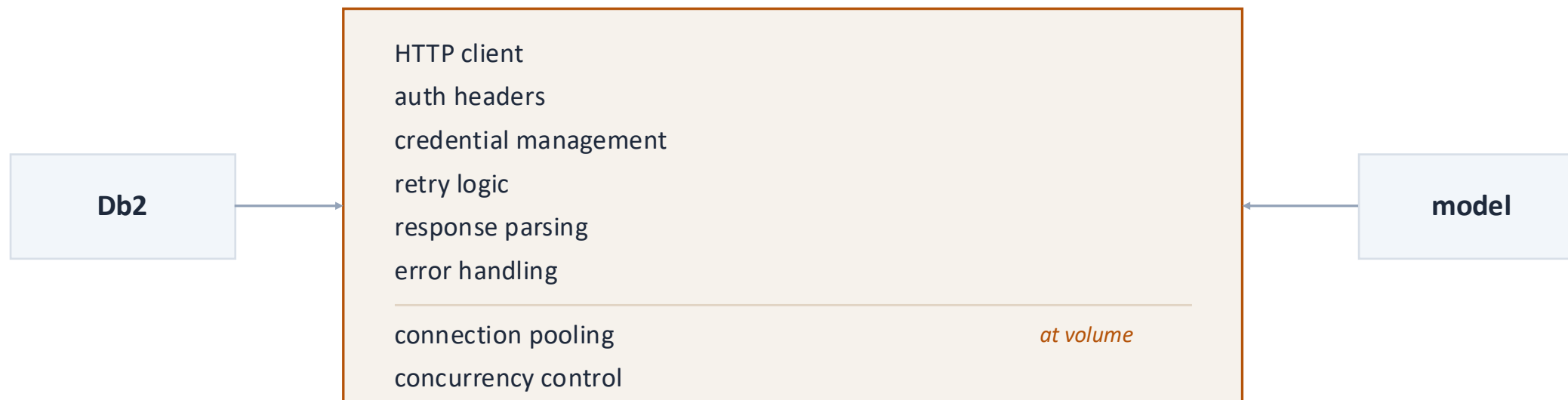


a walk through a graph

trades a little recall

for speed that doesn't

Between your database and your model is code you never wanted to write

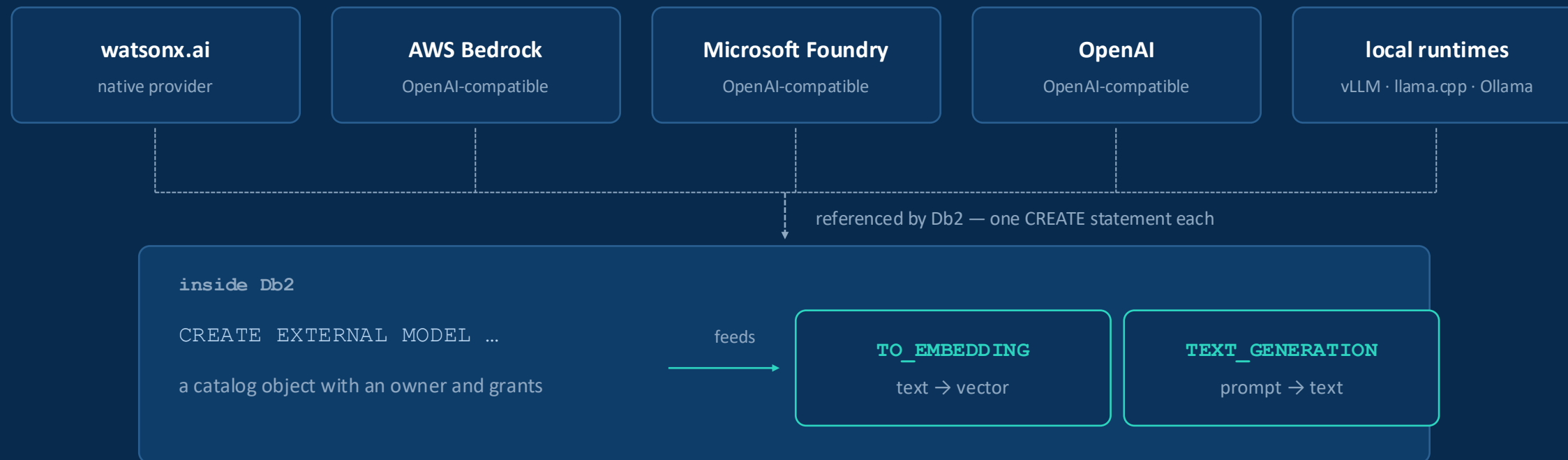


**None of it advances your application.
All of it is yours to maintain.**

Models became database objects

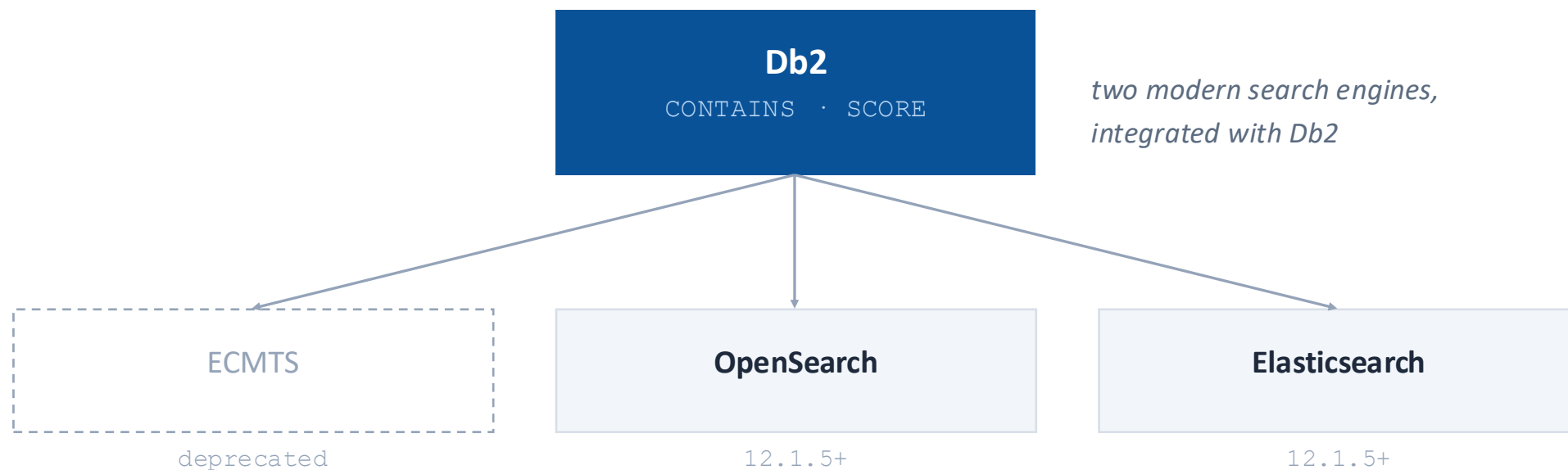
Register once. Call it like any other function.

WHERE THE MODELS LIVE



One statement, and the model behaves like any other function.

Text search moved to the engines your teams already run



Where this is going

`shipped`

vector type
similarity search
vector index
external models
modern text search
framework connectors

`under consideration`

document preprocessing capabilities
further gains in search relevancy
further support for the Python ecosystem

`Directional. Not a commitment to deliver.`

SECTION

Demo

Hybrid search over an audiobook library

**I own 92 audiobooks.
I can't find any of them.**

book about handling hostile Q&A

No results found

Fifteen years of listening. The search box matches titles.
My questions are about meaning.

This is one row in my library

How to Stop Worrying and Start Living

Dale Carnegie · narrated by Andrew MacMillan · 2007

Through Dale Carnegie's seven-million-copy best seller millions of people have been helped to overcome the worry habit. Carnegie offers a set of practical formulas you can put to work today. ... Discover how to eliminate business worries, reduce financial worries, turn criticism to your advantage, avoid fatigue, and find yourself and be yourself — there is no one on earth like you.

Everything a search engine knows about this book is on this slide.

Lexical search finds it — if you remember a word from it

query	result	why
stop worrying	found	exact term in the title
andrew macmillan	found	exact term in the narrator field
criticism	found	exact term, buried in the description
overthinking	nothing	the book is entirely about it — the word never appears
worry	only this one	misses every other book in my library on the same theme

Stemming and synonyms widen the net: worry → worries, worrying, worried, fret.
They reach variants of a word. They never reach a topic.

It matches words. Not ideas. Not topics.

Semantic search matches the topic — and everything else on it

same book — different words

how do I stop replaying
conversations at 2am

found

something for when I can't
switch my brain off

found

neither phrase appears in the description

andrew macmillan

buried

same topic — other books

"handling stress and staying level-headed"

1 **How to Stop Worrying and Start Living**

Dale Carnegie

2 **Working with Emotional Intelligence**

Daniel Goleman

3 **Your Brain at Work**

David Rock

4 **The Willpower Instinct**

Kelly McGonigal

no shared keyword — with the query, or each other

a bare proper name carries almost no meaning to embed

It matches meaning. It's blind to a token that carries none.

Two ways to search. Each one blind to what the other catches.

"walter dixon"

"why we misjudge people
we've just met"

LEXICAL

matches the words

✓ rank 1

✗ no match

SEMANTIC

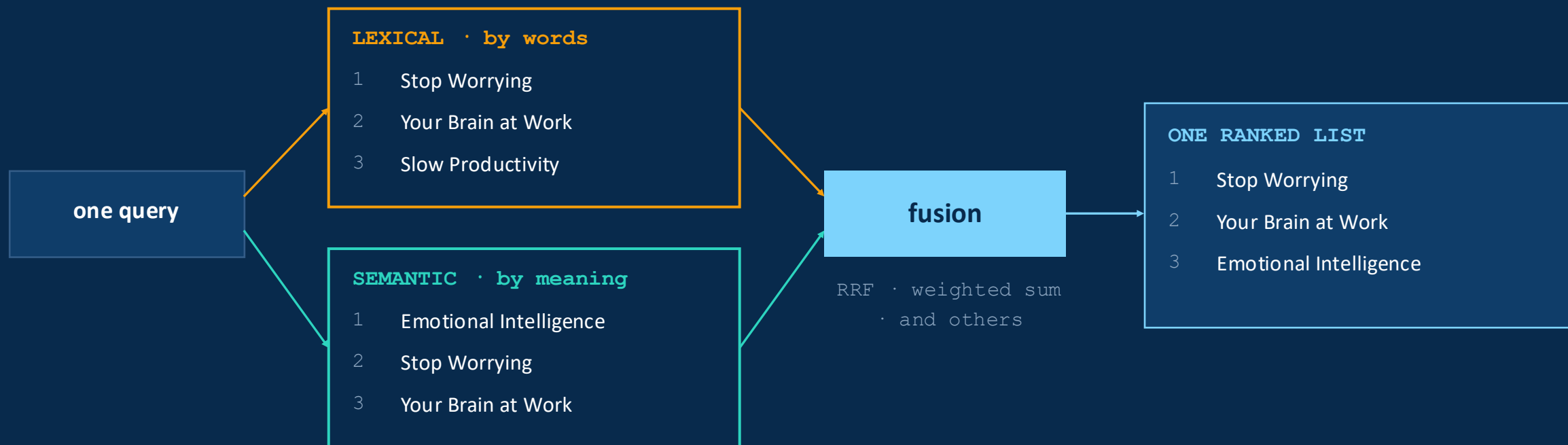
matches the meaning

✗ rank 36

✓ rank 1

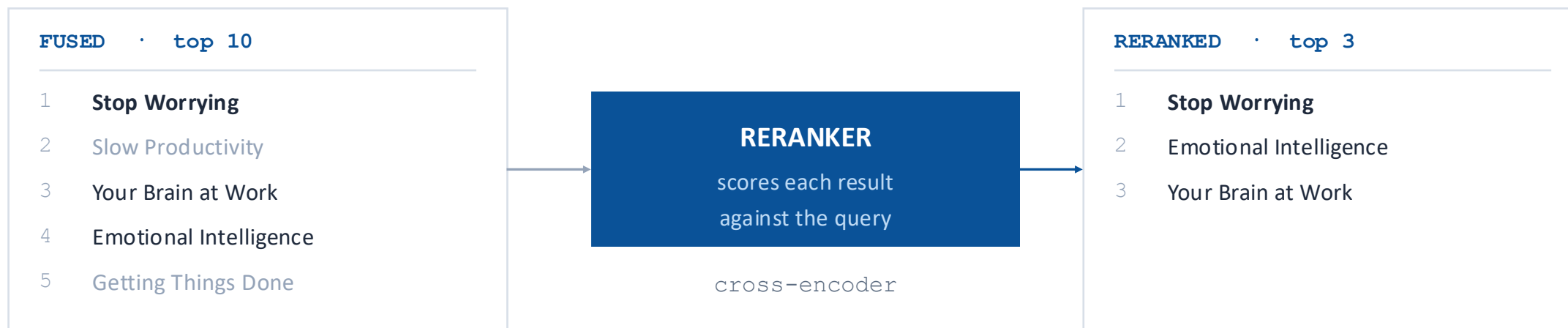
A bare name embeds to noise. · A paraphrase shares none of the book's own words.

Hybrid search combines both search results



Not a third way to search — a decision about what to do with two answers.

One more pass: a reranker reads the query and each result together

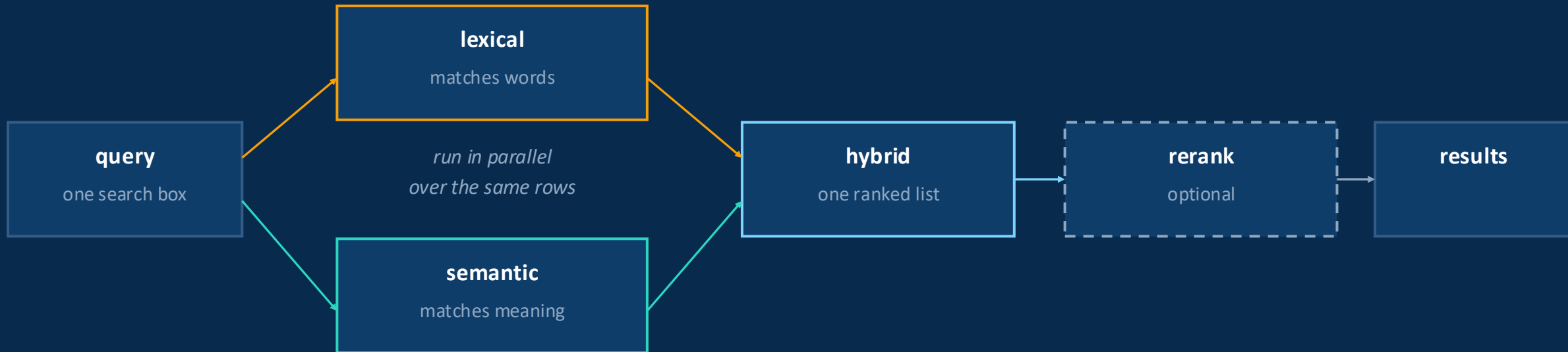


Retrieval encodes query and text separately,
then compares two vectors.

A reranker reads both texts together —
slower, and much better at judging relevance.

Fast and approximate first. Slow and precise on the short list.

What actually runs when you search



One query in. Two retrievers. One list out.

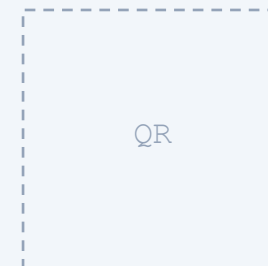
Everything you just saw is in one folder

github.com/IBM/db2-ai-cookbook

03-hybrid-search

bare RHEL box → running app in 30–45 minutes

CPU only · no GPU · no API keys · runs offline



01 tabular

02 multimodal

03 hybrid search

04 RAG

05 agentic RAG

06 recommendation

Six recipes. This is the one that ships a number.

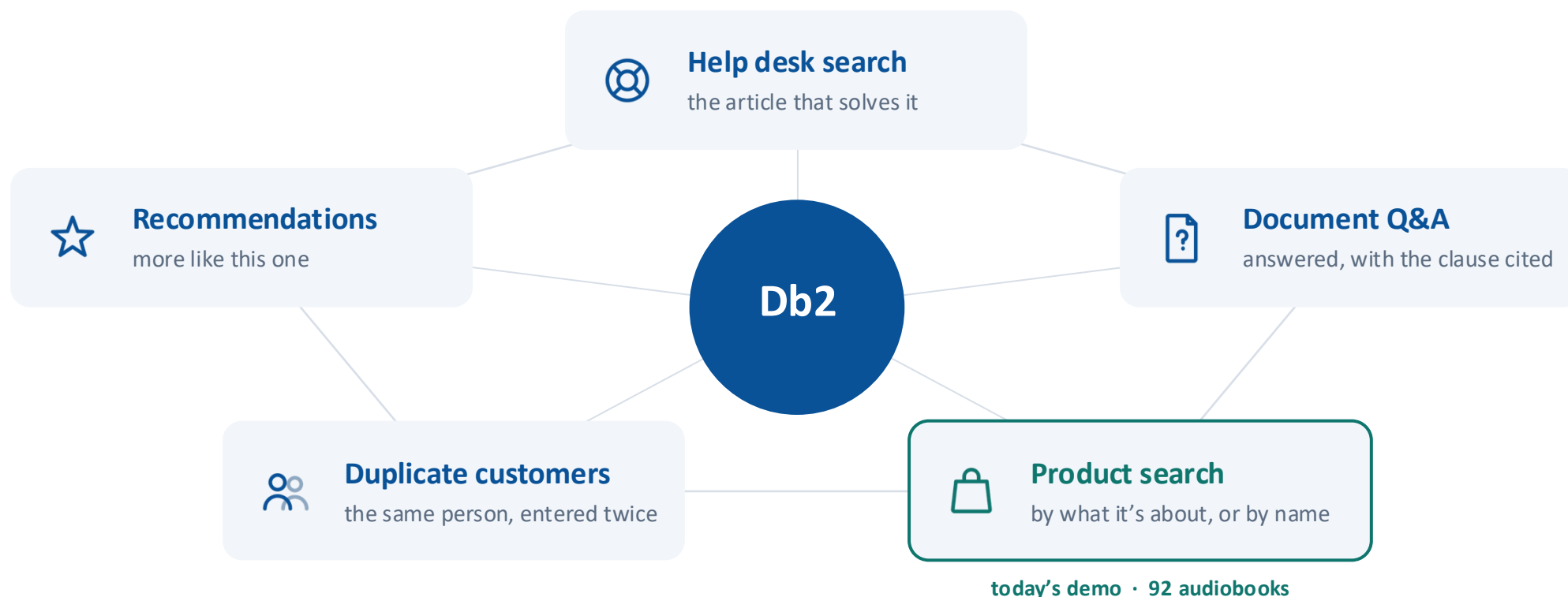
Your corpus has both kinds of query too

	exact token LEXICAL	description SEMANTIC
support KB	<code>SQL0803N</code>	my import rejects every row
product catalog	<code>WH-1000XM5</code>	noise cancelling for long flights
contracts	<code>Section 12.3 (b)</code>	can they exit early without penalty
claims / records	<code>ICD-10 E11.9</code>	adult-onset diabetes

Same box. Same day. Two vocabularies.

Five things you can build on data you already have in Db2

One session isn't enough for five. We'll go deep on one.



Every one of these is retrieval with a different job on the end.



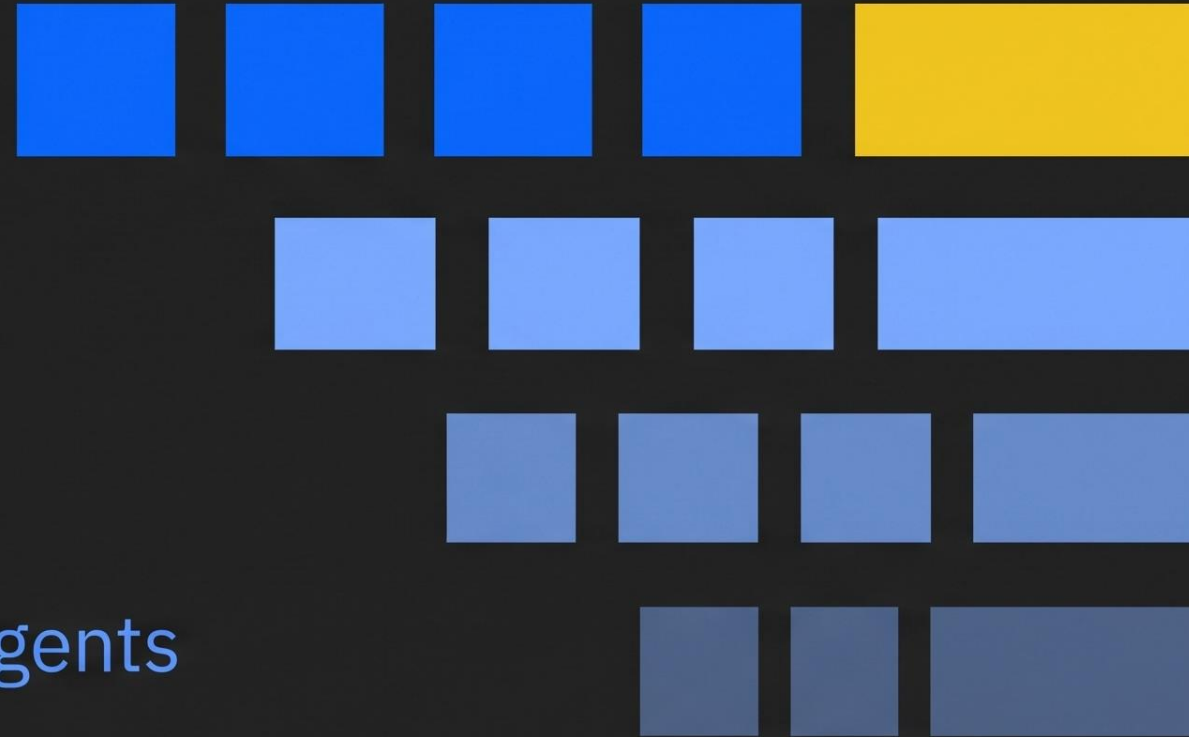
Db2 AI Cookbook

recipes for building AI apps and agents

IBM Db2 12.1+ · from SQL or Python



github.com/IBM/db2-ai-cookbook



I Write Online

LinkedIn



Shaikh Quader

AI Architect, IBM Db2

Newsletter



aiarchitectplaybook.com

How production AI gets built,
and how engineers grow into leading it

An aerial photograph of a city at sunset. A prominent cable-stayed bridge with two tall, angled pylons spans a body of water. Below the bridge, a multi-lane highway with several lanes of traffic curves through the scene. The city skyline is visible in the background, with numerous high-rise buildings. The sky is a mix of orange, pink, and blue. The overall image has a blue tint, especially in the lower half.

IDUG

2026 Brazil **Db2** Tech Conference