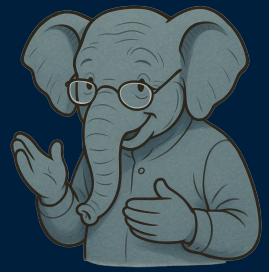
dbtune

Grandpa's guide to technical empathy: Decrypting PostgreSQL for strategic clarity

Ellyne Phneah
Nordic PGDay 2026




So... when do we ship
the new feature?

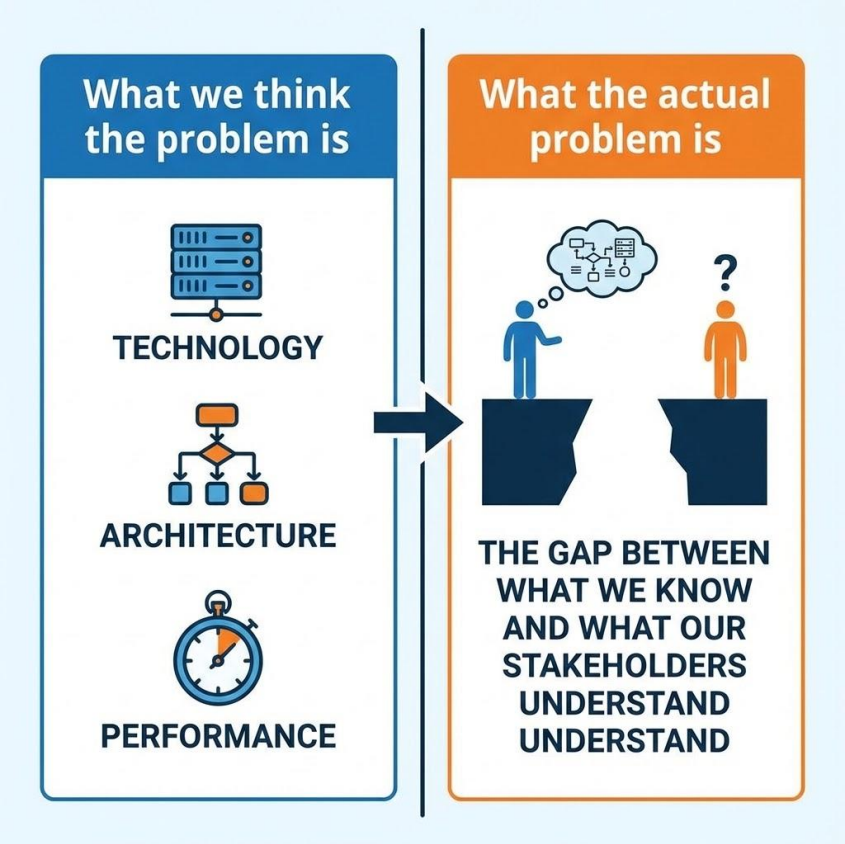
We’ve all been in that room

Have you ever had a technical recommendation ignored, only to have it come back as a 'critical emergency' three months later?

Curse of Knowledge

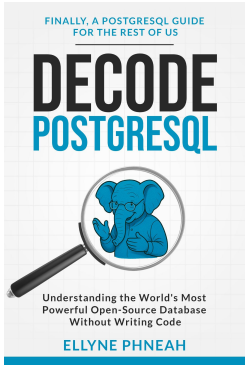
Source

“The bottleneck isn’t your code”



Who am I?

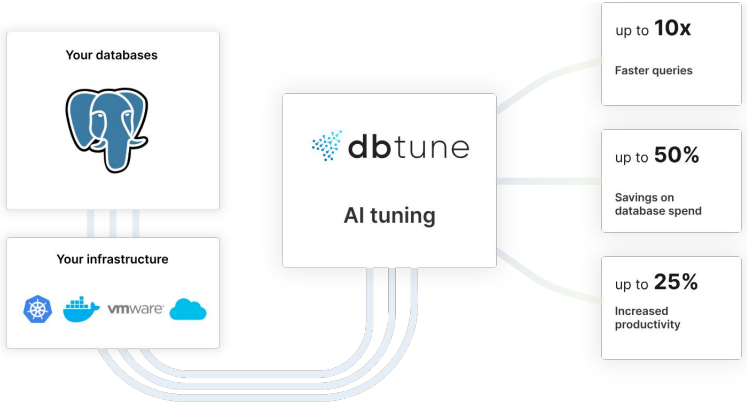
- Ellyne Phneah
- Marketing Manager at DBtune
- Former Tech Journalist
- 14+ years in PR, brand and communications
- BA in English Literature
- MSc in Entrepreneurship and Innovation
- Author of Decode PostgreSQL



Nordic PGDay

About DBtune

- Agentic AI for automated PostgreSQL database tuning
- Observes, iterates and adapts until converging and delivering optimal settings for any individual workload, use case and machine.



Tune your PostgreSQL instance

Nordic PGDay

Malmo PostgreSQL User Group (M-PUG)

M-PUG organizers

Ellyne Phneah
DBtune

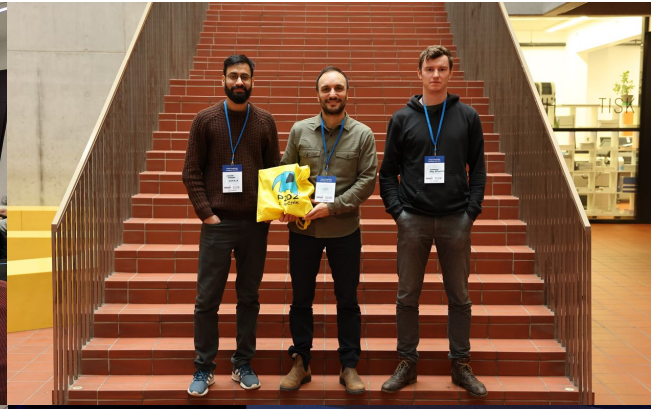
Dr. Luigi Nardi
DBtune

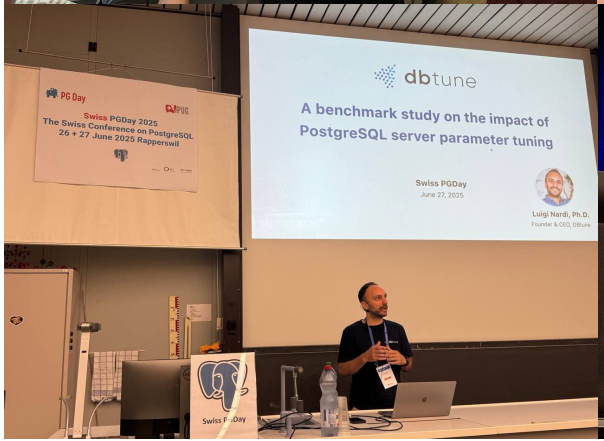
Daniel Gustafsson
Microsoft

Join M-PUG

Nordic PGDay

Community trainings and presentations





Nordic PGDay

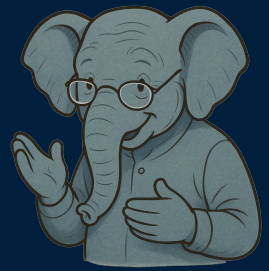
Agenda

- **THE THEORY:** Building Technical Empathy
- **THE FILTER:** Mastering the Grandpa Test
- **THE TOOLKIT:** 3 Repeatable frameworks
- **THE REFRAME:** Translating "Hard Stuff" & debt

dbtune

Technical Empathy

What technical empathy is (and isn't)



Let’s clear something up

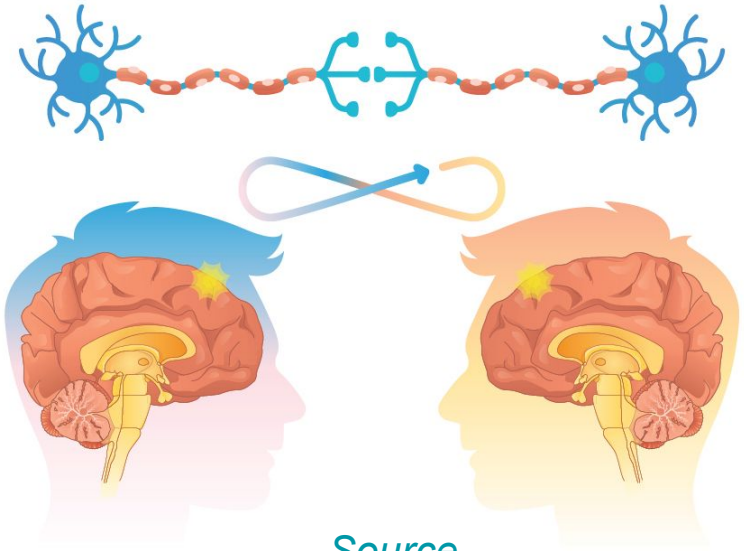
Technical Empathy

Dumbing things down

Technical Empathy *is the ability to understand the knowledge journey of your audience. It's knowing how it feels not to know what you know. It's meeting people where they are — and building the bridge from their world to yours.*

Nordic PGDay

The neuroscience behind why this works

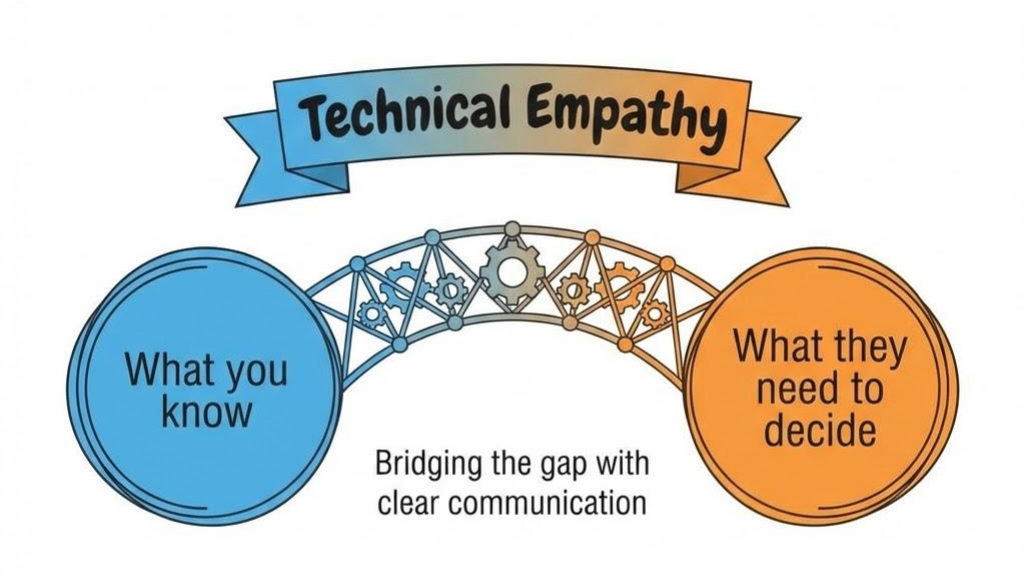


Source

Mirror neurons fire not just when we act — but when we watch someone else act. Empathy isn't metaphorical. It's neurological.

Nordic PGDay

It's not about them. It's about the bridge.



The business case — in one stat.

56%

Projects fail
due to **poor comms**
(PMI Pulse of the Profession)

Not poor technology. Poor communication.

Teams that bridge the gap well, win.

Nordic PGDay

-

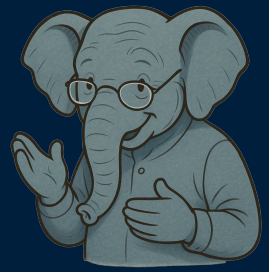
Unlike learning system internals, you can develop this skill in weeks — not years.

Here's how

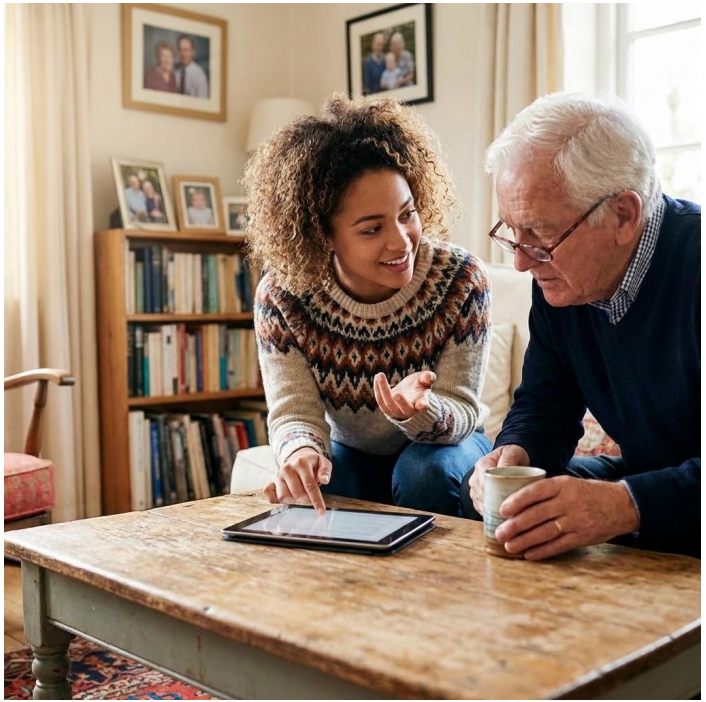
dbtune

The Grandpa Test

Your translation litmus



- What is the most complex PostgreSQL project you are working on right now?
- Explain it to a 10-year-old or your grandpa in exactly two sentences without using the word 'database,' could you do it?



The universal struggle of technical translation

Nordic PGDay

Your grandpa and your CEO want the same things

Grandpa

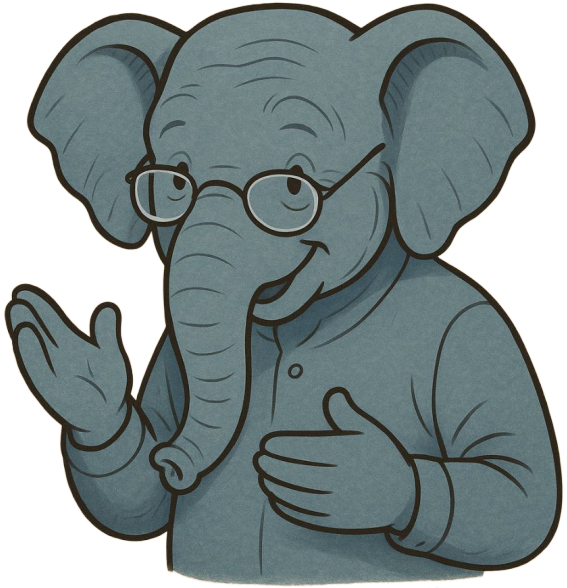
Find things faster

Avoid waste

Don’t pay for problems that could have been prevented

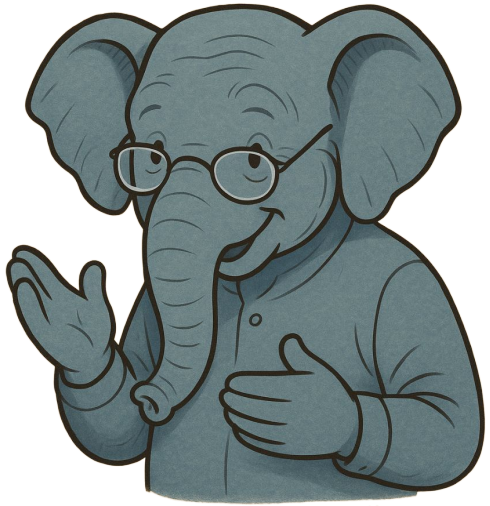
CEO

The Grandpa Test is how you practise Technical Empathy deliberately.



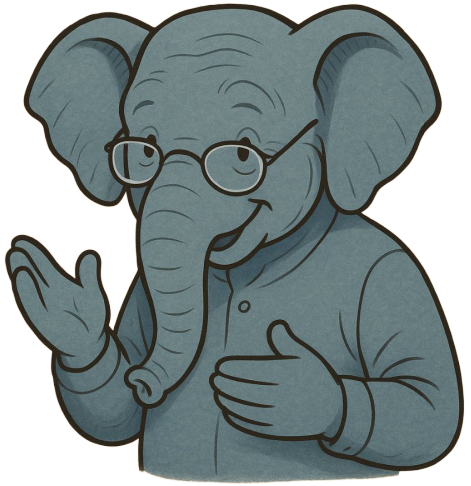
The Grandpa Test: 3 steps

1. State the technical thing in your own words.
2. Ask — "What does this do for the person using it?"
3. Ask — "What does it cost them if it fails or is delayed?"



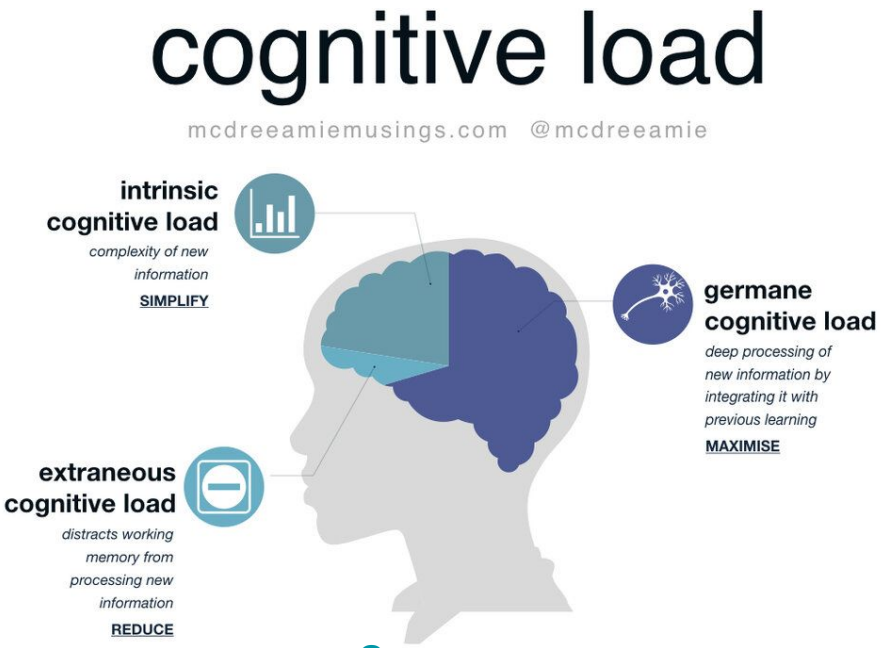
Why these three steps — in this order

Step 1: State it clearly	Step 2: Shift to outcome	Step 3: Name the cost
Cognitive Load Theory	The Elaboration Likelihood Model	Loss Aversion (Kahneman & Tversky)



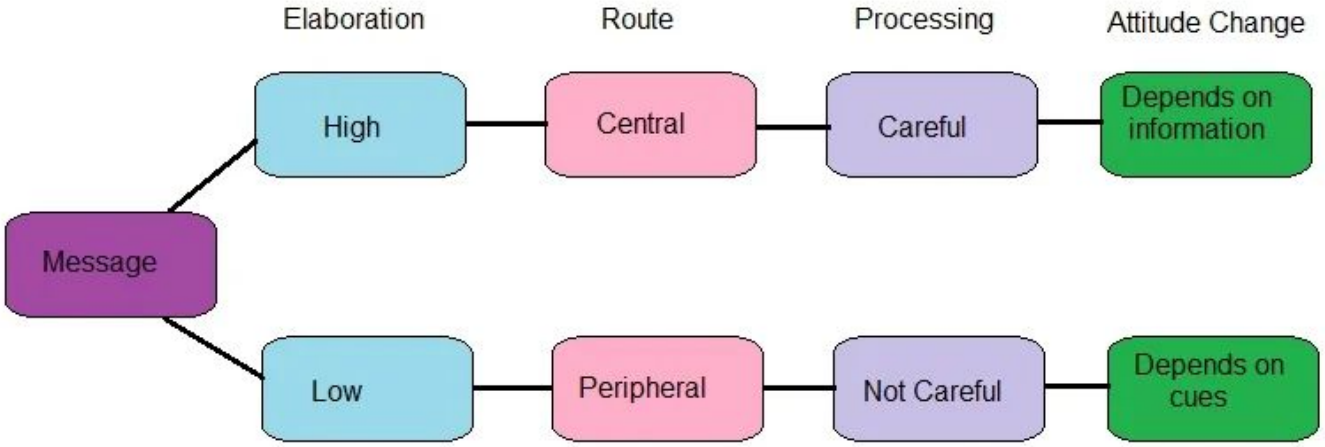
Nordic PGDay

Step 1: State it clearly



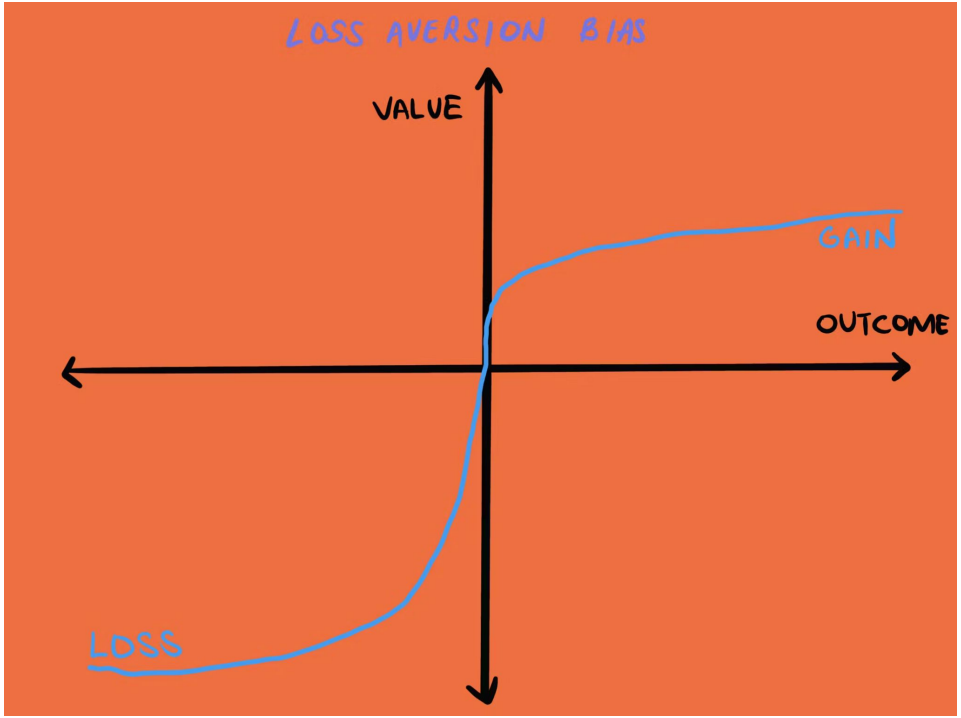
Source

Step 2: Shift to outcome



Source

Step 3: Name the cost



Source

Example 1 — pgvector / Semantic Search [BEFORE]

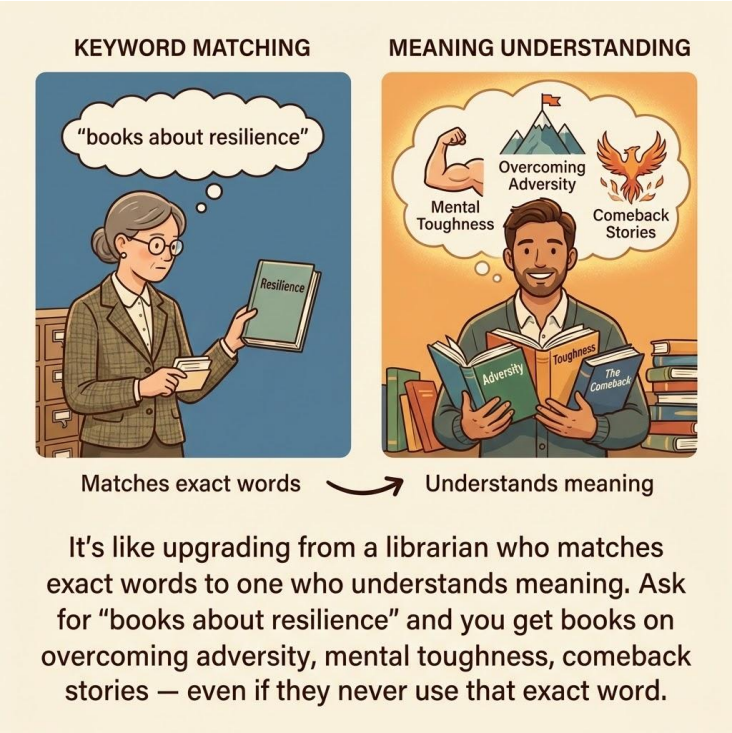
Doc

```
Semantic search uses
vector embeddings
generated by machine
learning models,
models, enabling
similarity matching
beyond
keyword-based
queries.
```

Technically accurate, strategically useless

Nordic PGDay

Example 1 — pgvector / Semantic Search [AFTER]



✅ **What it does:** finds what users mean, not just what they typed

✅ **Cost of delay:** competitors ship smarter search; users migrate to products that feel more intelligent

✅ **Strategic translation:** AI-powered UX advantage, within existing infrastructure — no new vendor

Example 2 — VACUUM [BEFORE]

```
VACUUM reclaims
storage occupied
by dead tuples
and updates the
visibility map to
improve query
planner
accuracy.
```

Technically accurate, strategically useless

Example 2 — VACUUM [AFTER]

Dead tuples are like ex-employees whose files are still taking up space in your filing room. VACUUM clears them out so the system isn't wasting time searching through records that no longer matter.

✅ **What it does:** keeps the database fast and lean as data changes over time

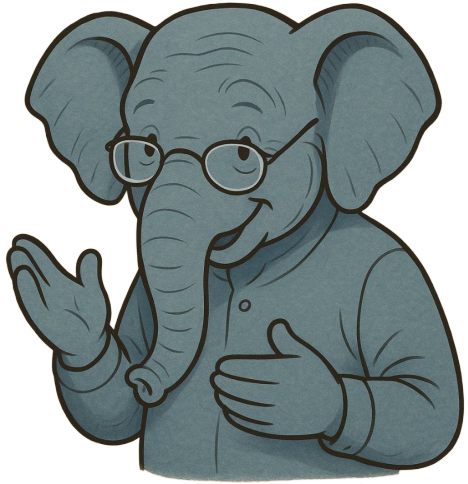
✅ **Cost of delay:** queries slow down; storage costs climb; performance degrades invisibly — until it isn't invisible anymore

✅ **Strategic translation:** scheduled maintenance now vs. emergency intervention later — the price difference is not small

If grandpa gets it — you're ready.

Now it’s your turn!

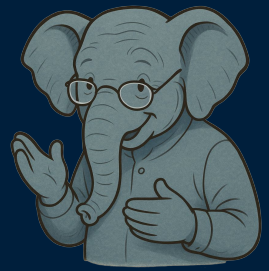
- Find a "painfully technical" PostgreSQL term (e.g., *Write Ahead Log*, *MVCC*, or *B-Tree Indexes*).
- Spend 5 minutes applying the 3-step Grandpa Test to that specific term on the fly.



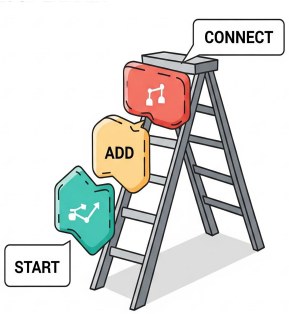
dbtune

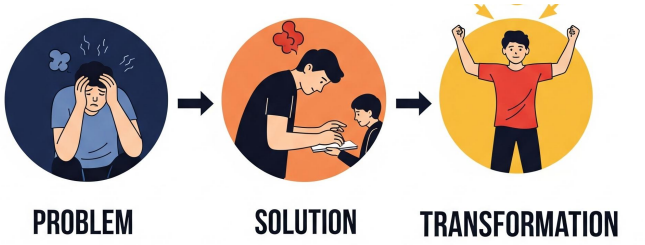
3 translation frameworks

Repeatable blueprints that work with any audience



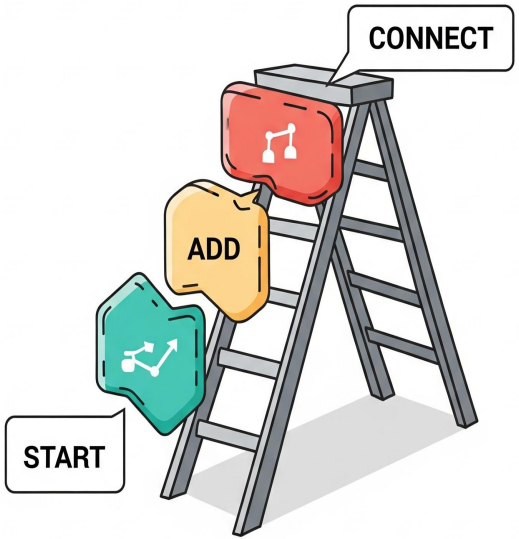
Three frameworks. One goal.



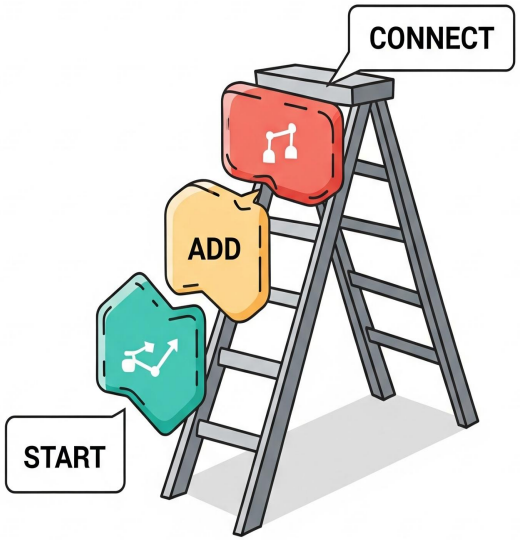


From 'how it works' to 'why it matters' — systematically.

Framework 1: Analogy ladder

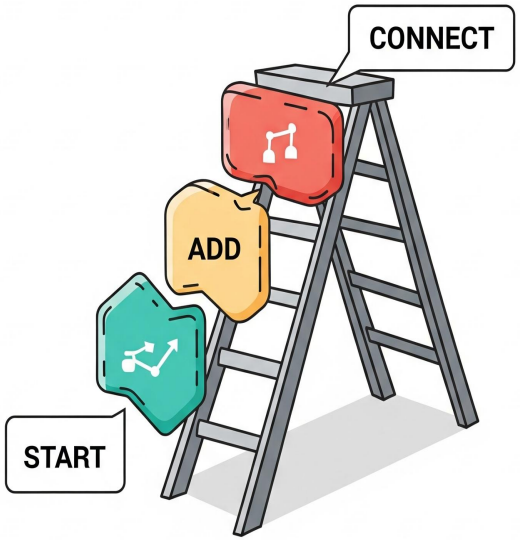


Framework 1: Analogy ladder (Explaining PostgreSQL itself)



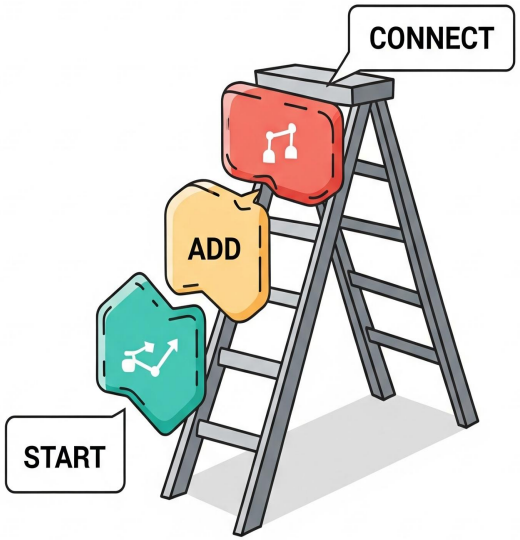
START: You know how Excel organizes data in rows and columns?

Framework 1: Analogy ladder (Explaining PostgreSQL itself)



START: You know how Excel organizes data in rows and columns?
ADD: PostgreSQL is like Excel, but designed for millions of rows and multiple people using it simultaneously

Framework 1: Analogy ladder (Explaining PostgreSQL itself)



START: You know how Excel organizes data in rows and columns?
ADD: PostgreSQL is like Excel, but designed for millions of rows and multiple people using it simultaneously
CONNECT: This means your entire team can access the same data without conflicts or version control nightmares

Framework 1: Analogy ladder (Connection Pooling)

START: You know how a busy restaurant has a reservations system — not every table is set up for every possible guest, they manage the flow?
ADD: Connection pooling does the same thing for your database. Instead of opening a new connection for every single request, it maintains a managed pool that gets reused efficiently.
CONNECT: Which means your application handles ten times the traffic without the database collapsing under the weight of thousands of simultaneous connection requests — which is exactly what happens during a product launch or a flash sale.

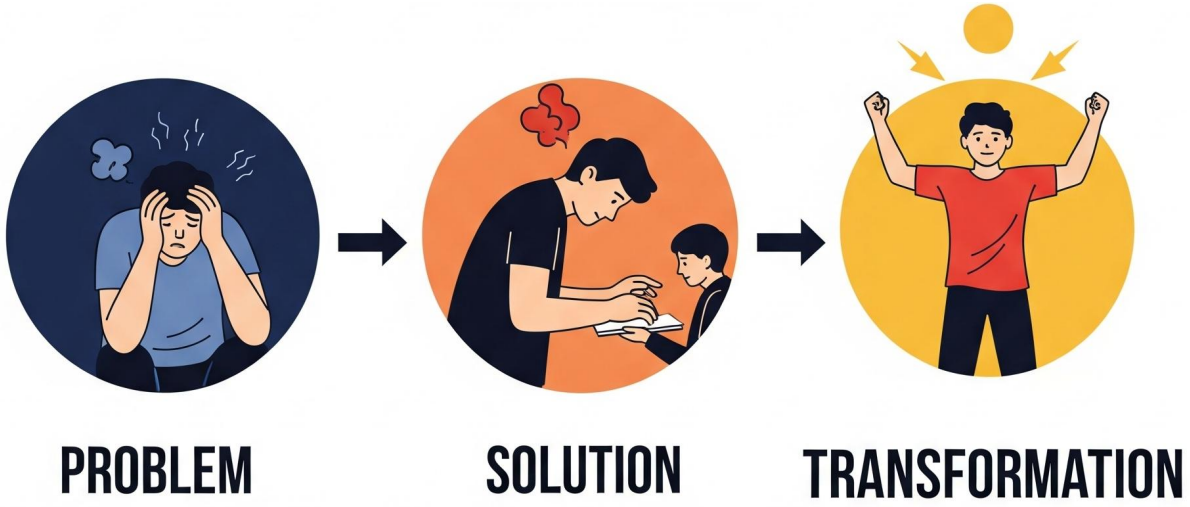
The Analogy Ladder — Key Rule

If your audience has to jump,

you skipped a rung.

Test it: could someone with no technical context follow each step without losing the thread?

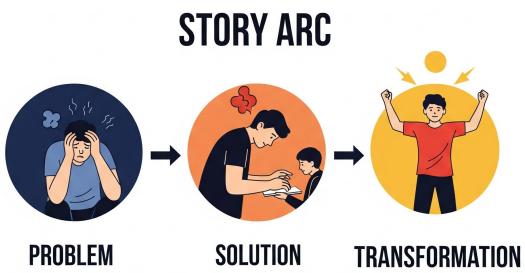
Framework 2: The story arc



Most technical communicators nail problem and solution. Almost nobody delivers transformation.

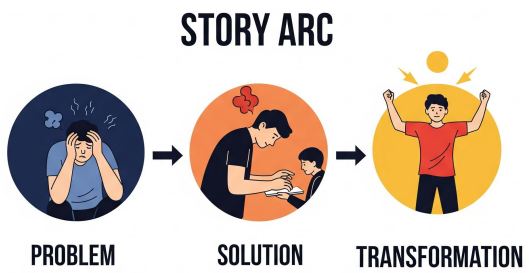
Framework 2: The story arc (Explaining PostgreSQL itself)

PROBLEM: Your company has data scattered across multiple tools—customer info in Salesforce, analytics in Google, orders in Shopify



Nordic PGDay

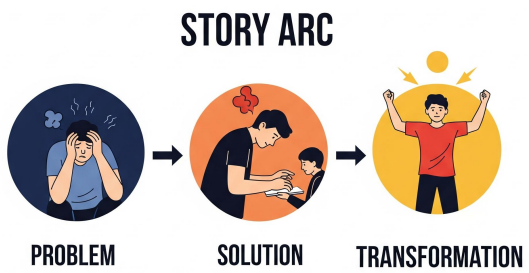
Framework 2: The story arc (Explaining PostgreSQL itself)



PROBLEM: Your company has data scattered across multiple tools—customer info in Salesforce, analytics in Google, orders in Shopify

SOLUTION: PostgreSQL brings it all together in one organized, queryable place

Framework 2: The story arc (Explaining PostgreSQL itself)



PROBLEM: Your company has data scattered across multiple tools—customer info in Salesforce, analytics in Google, orders in Shopify

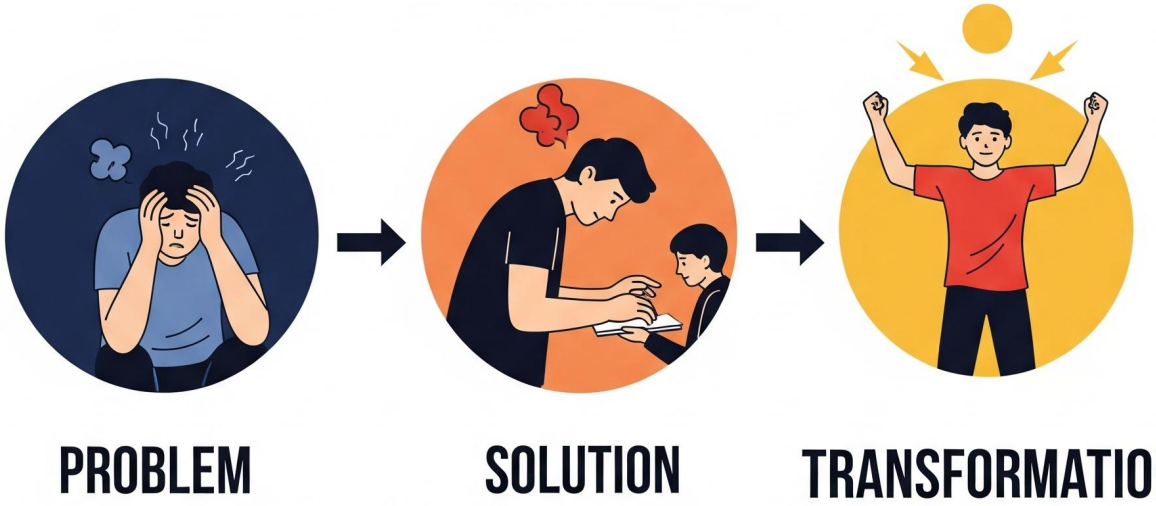
SOLUTION: PostgreSQL brings it all together in one organized, queryable place

TRANSFORMATION: Now you can answer complex questions like 'which marketing campaigns drive the highest-value repeat customers?' in minutes, not days

Framework 2: The story arc (Technical Debt)

PROBLEM: Customer data in Salesforce. Analytics in Google. Orders in Shopify. Three tools, three sources of truth, no single answer to a basic question.
SOLUTION: PostgreSQL brings it together — one organised, queryable, reliable source.
TRANSFORMATION: Which marketing campaigns drive our highest-value repeat customers?' — answered in minutes, not days. Campaigns get smarter. Budget goes where it works.

Transformation is where most people stop too early.



Don't stop at Solution. The Transformation is why anyone should care.

Framework 3: The Value Bridge

This is the framework that turns a feature into a competitive advantage.

Framework 3: The Value Bridge (pgvector)

Tech capability: native vector embedding support.
Accurate.
Unmemorable.

Framework 3: The Value Bridge (pgvector)

Tech capability: native vector embedding support. Accurate. Unmemorable.	**The product outcome:** You can ship AI-powered semantic search without leaving PostgreSQL. No new vendor evaluation. No migration project. No integration overhead eating your engineering sprint capacity. The capability lives where your data already lives.

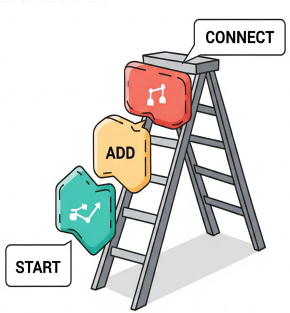
Framework 3: The Value Bridge (pgvector)

Tech capability: native vector embedding support. Accurate. Unmemorable.	**The product outcome:** You can ship AI-powered semantic search without leaving PostgreSQL. No new vendor evaluation. No migration project. No integration overhead eating your engineering sprint capacity. The capability lives where your data already lives.	**The business result:** You ship a competitive AI feature faster than a team that's evaluating external vendors, and you do it without adding a new line to the infrastructure budget. That's a competitive advantage and a cost story in the same sentence.

Framework 3: The Value Bridge (Read Replicas)

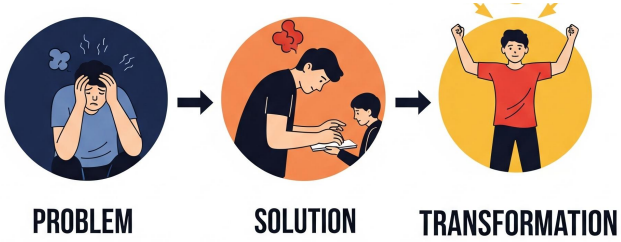
Tech capability: Read replicas distribute query load across multiple database instances	**The product outcome:** Reporting, analytics, and heavy read operations run without slowing down the transactional database your users depend on.	**The business result:** You scale to support a growing user base and data-intensive features without emergency infrastructure spend — and without your product slowing down at the moment it matters most

Three frameworks — When to use which



Analogy Ladder

When your audience needs to *understand* a concept before they can evaluate it



Story Arc

When your audience needs *context and consequence* to make a decision

Value Bridge

Use when your audience needs to see *competitive or business relevance* of a capability

dbtune

Translating the hard stuff

Technical debt as strategy

Now we get to the hard conversations.

Technical debt doesn't lose budget battles because it's unimportant. It loses because nobody has named it correctly.

The vocabulary problem

How we describe it	How they hear it
Table bloat	Housekeeping
Missing indexes	Performance tuning
Deferred VACUUM	Technical backlog

Same problem. Wrong language. Wrong priority.

Table Bloat — Renamed

Technical label	The reframe
❌ We have significant table bloat that needs to be addressed.	✅ "We're paying rent on a warehouse full of furniture we threw away two years ago."

Missing Indexes — Renamed

Technical label	The reframe
❌ We're missing indexes on several high-traffic query paths.	✅ We're searching every filing cabinet by hand when we could have a directory.

VACUUM — Renamed

Technical label	The reframe
❌ We need to ensure VACUUM is running regularly to reclaim dead tuples.	✅ This is a scheduled oil change. Skip enough of them and you're not paying for an oil change anymore — you're paying for an engine replacement.

The vocabulary shift — summarised.

Technical reality	Old language	New language	Business category
Table bloat	Housekeeping	Paying rent on empty space	Cost avoidance
Missing indexes	Performance tuning	Searching without a directory	Operational risk
Deferred VACUUM	Technical backlog	Skipping oil changes	Risk management

Technical debt is not an engineering problem, but a communication one.

The infrastructure was always worth investing in. It just needed to be described in a language that made the investment obvious

dbtune

Audience in the room

Different stakeholders, different value propositions

Same PostgreSQL. Four different conversations

dbtune

Communicating PostgreSQL Value: Audience-Specific Translation

Audience	They optimise for	Your translation lens	PostgreSQL example
Executives	Cost & Risk	Cost avoidance, risk management, control	"Deferred maintenance transfers timing control to the system, not to us"
Sales	Competitive edge	Differentiation, speed to market, customer promises	"PostgreSQL lets us promise real-time analytics and actually deliver it"
Product	Roadmap value	Feature velocity, capability unlocks, user outcomes	"pgvector means AI search ships in this sprint, not next quarter"
Marketing	Campaign performance	Data quality, insight speed, targeting accuracy	"Every personalised campaign starts with clean, queryable data"

Sales

Optimize for competitive edge

⚠️

❌ Don't lead with "Constraints."

✅ Lead with "Performance stability as a selling point.

Executives

Optimize for cost and risk

⚠️

❌ Don't lead with "Efficiency."

✅ Lead with "Insurance against downtime

Product

Optimize for roadmap value

⚠️

❌ Don't lead with "Technical Debt."

✅ Lead with "Clearing the path for faster shipping in Q3

Marketing

Optimize for campaign performance

⚠️

❌ Don't lead with "Features."

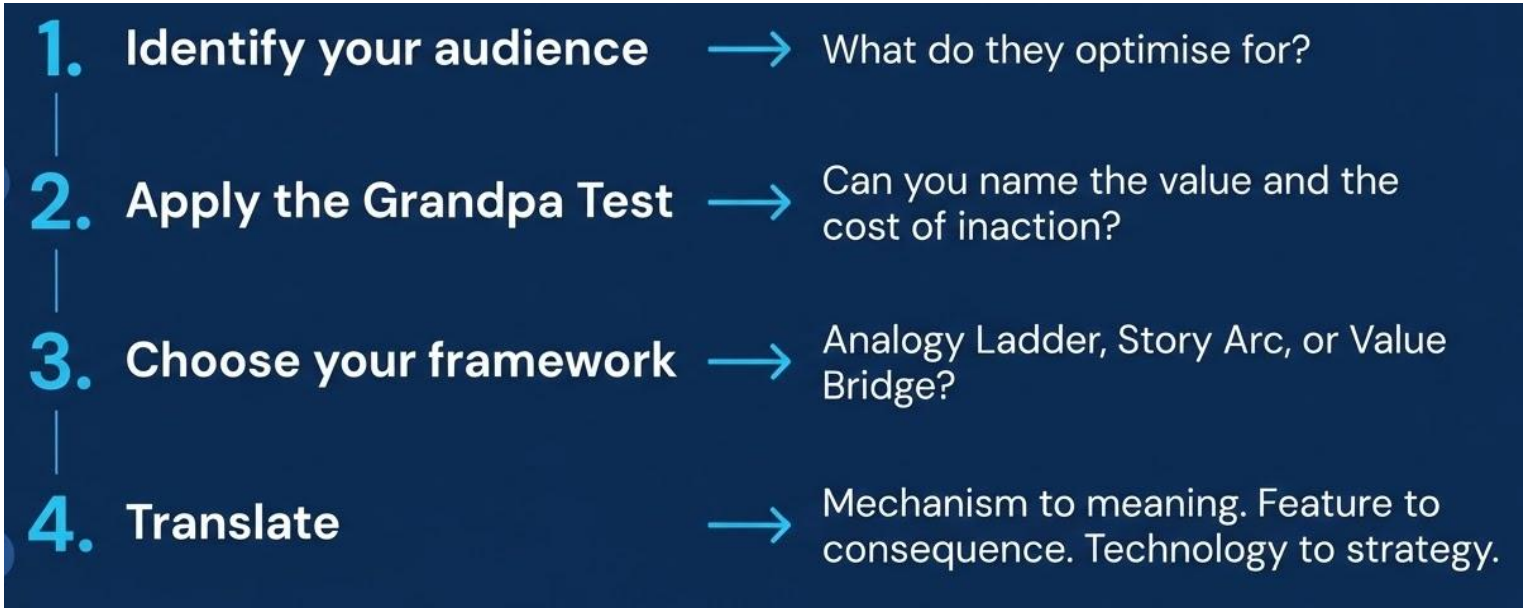
✅ Lead with "The 'Never-Lose-A-Transaction' Promise.

Nordic PGDay

Who is in this room, and what do they optimise for?

Answer that first. Then choose your framework.
Then communicate.

The Technical Empathy workflow



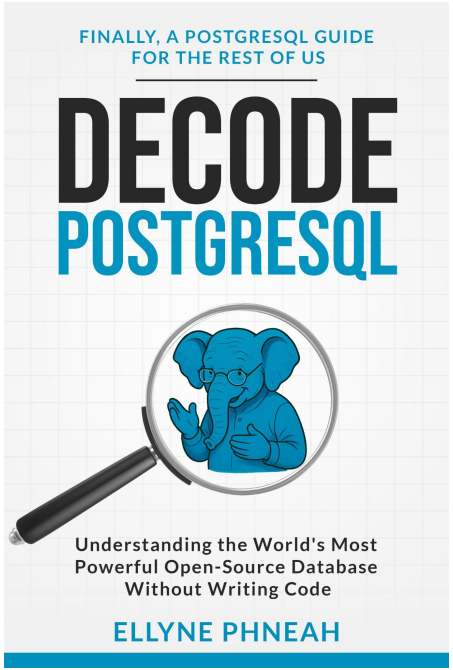
Your Monday Morning Homework

Five minute stakeholder audit

Time	Step	Action
Minute 1	Identify jargon	Pick one technical project (e.g., Partitioning, PG17 Upgrade).
Minute 2	Grandpa test	Explain it in 2 sentences. **Zero** database words allowed.
Minute 3	Map the “boss”	Pick your stakeholder (CEO/Sales/Product). Identify their #1 priority.
Minute 4	Value bridge	Define the **Cost of Inaction**. What do *they* lose if you don't do this?
Minute 5	The hook	Draft your opening line. Lead with the **Stakes**, not the Tech.

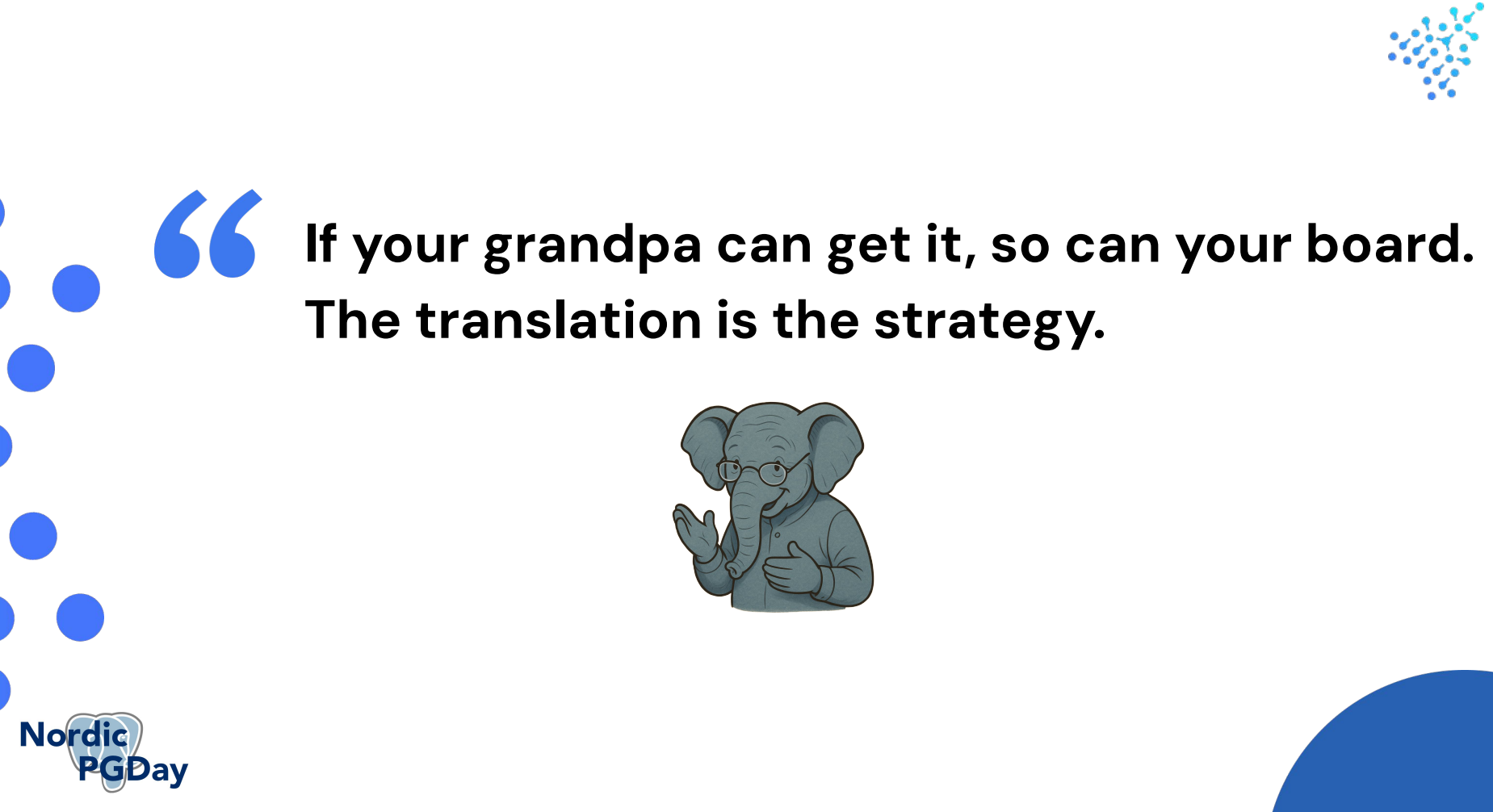
Decode PostgreSQL (ebook)

- **No code, no jargon**: Just clarity
- A metaphor driven framework to explain PostgreSQL
- **14 chapters** of insights
- **Bonus section** including FAQs, common terms and resources
- **Available** across all digital devices



50% discount until **27 Mar**
Code: **NordicPGDAY2026**

Nordic PGDay


If your grandpa can get it, so can your board.
The translation is the strategy.
Nordic
PGDay

Thank you

📩 ellyne@dbtune.com

Let's connect on LinkedIn