EDB Developer U

Andrew Dunstan
Senior Principal Engineer
Director of Postgres Development
2026-05-21

About this talk

- Program developed at EDB over past year or so
- Why we built it, how it has run, what we've learned
- For anyone interested in growing Postgres talent
- Honest look — wins, stumbles

Why Developer U

It is hard to find contibutors

- PostgreSQL is hard to contribute to
- The barrier is not intelligence — it's unfamiliarity
- C, the codebase, the mailing list, the patch lifecycle
- Each is a hill; together they are a mountain

EDB has untapped talent

- Many engineers work *around* Postgres, not *on* it
- People working on related products
- Engineering and support staff who know the internals from helping customers
- Familiar with using Postgres, less so with contributing to it
- Strong professional skills, just not core hackers
- Curiosity is everywhere — opportunity is not
- Developing existing talent is easier than hiring new contributors

Benefits of developing internal talent

- The more community contributors the healthier the project
- Bigger recruitment pool
- Career growth path for existing employees
- Retention: people stay where they grow
- Enhance job skills for existing positions

Goals for the program

- Take willing engineers and make them credible contributors
- Teach **Code**, the **Architecture**, and the **Community**
- Get participants doing reviews
- Get participants to first patch — and beyond
- Build a repeatable model, not a one-off

Building the Program

Program Outline

- Lasts a year
- Multiple in person sessions, lasting 3 days
- Mixture of knowledge transfer and hands-on activity
- Work in pairs
- Between sessions continue reviewing and contributing

Where we started

- Pitched to EDB management in early 2025 — immediate support
- Middle management buy-in
- No template — community has docs, not a curriculum
- Executive sponsorship was essential

Recruitment

- Open call across engineering
- Self-nomination plus manager endorsement
- Cast a wide net
- No prior C requirement — willingness mattered more
  - In the end everyone did have some C background
- Demand exceeded what we could choose

Selecting the cohort

- Capped at a manageable size for real mentorship
- Mix of experience levels — deliberate, not accidental
- Geographic spread across many time zones
- Commitment from the participant *and* their manager
- Department caps to minimize impact

Protecting the time

- The hardest part of the whole program
- Manager sign-off on dedicated weekly hours
- Treated like any other project commitment
- Without this, the program quietly dies

The Curriculum

Three tracks, woven together

- **Code** — the language and idioms used in the tree
- **Structure** — Postgres internals, layer by layer
- **Community** — how the project actually works

The Code track

- Not "C from scratch" — assumes some programming
- Postgres-flavored C: palloc, ereport, List, hash tables
- Reading code as a skill, not just writing it
- Debugging with gdb against a live backend

The Structure track

- Fundamentals first — e.g. memory contexts, not obvious to outsiders
- Backend startup and the life of a query
- Parser, planner, executor — the big three
- Storage, MVCC, WAL at a working level

The Community track

- We start here — most participants had never posted to -hackers
- Many had never seen the commitfest app, never done a review
- Younger developers used to GitHub-style issues and PRs, not mailing lists
- Reading -hackers, finding threads, following discussions
- Commitfest, reviewing, the patch lifecycle from submission to commit
- Tone, patience, and the long game

Format

- Three-day in-person sessions
- Each day: half knowledge transfer, half hands-on work
- Participants work in pairs — peer support, fewer dead ends
- Material grounded in real code, not slides alone

Starting with reviews

- Before writing patches, participants reviewed them
- Real commitfest entries, not contrived examples
- Forces engagement with -hackers and the patch lifecycle
- Builds judgement before asking them to exercise it

First patches

- All projects drawn from the Retail DDL initiative
- A real, in-flight feature area — not toy exercises
- Submit to -hackers with guidance from the program
  - I did not give enough guidance

The Sessions So Far

Barcelona

The Barcelona cohort

Barcelona — Day 1

- Community first: mailing lists, commitfest, patch lifecycle
- Community culture - tendency for directness
- Played Robert Haas's recording on reviewing patches
- Postgres architecture
  - Parser, Planner, Executor ...
- Afternoon: review real commitfest patches, in pairs
- End of day: each pair presented their review back to the room

Barcelona — Day 2

- Deeper into Postgres architecture
- Memory management fundamentals — unfamiliar coming from other projects
- The system catalog
- Using a debugger against a running backend
- Writing a new builtin SQL function — the simplest task
- Afternoon: picked features from the Retail DDL initiative
- Some of those patches have since shown up on -hackers

Barcelona Day 3

- Writing documentation
- Writing Test cases
  - regression
  - TAP
  - isolation
- Logging and Error Handling
- Coding Standards
- continued work on DDL functions

Dubai

The Dubai cohort

Dubai — February 2026

- Different format: one shared project, instructional not submittable
- "Create EAV object" — a deliberate anti-pattern as a learning vehicle
- Touches grammar, node types, hooking nodes to commands

Dubai Day 1

- Introduction to project
- Creating SQL commands
- Parsers and PostgreSQL
- Keywords
- Nodes

Dubai Day 2

- Creating new system catalogs
- Adding a command implementation
- Hooking up implementation to the traffic cop
- Set returning Functions
- Extending the project (ALTER EAV etc)

Dubai Day 3

- Introduction to claude code
- Installfest
- Experimentation
- What is it (not) good at?
- Reviews
  - what does it miss/catch?
- Code
  - Add a new piece to the EAV project
  - quality?
- Diagnose a bug

Montreal

Upcoming Montreal June 2026

- Focus on deeper internals
- possible projects
  - Complete EXPLAIN memory accounting for HashAgg / HashJoin
  - Parallelize a currently-serial executor node
  - Per-node timing distribution in EXPLAIN ANALYZE
  - …

How It's Gone So Far

What worked

- Cohort camaraderie across time zones
- Real patches on the list, not toy exercises
- Managers who took the time commitment seriously

What was hard

- Time zones — no session time pleases everyone
- Venues - visa difficulties
- "Regular job" pressure crept back in
- Pace: too fast for some, too slow for others

Participant voices

- "If I had to do it all on my own, it was going to be one of those

things which gets put off until I have time."

- "I gained some more solid understanding and new perspectives this

week. That's much harder when it's just me poking around at the code." I finally understand what the planner is doing"

- "It makes sense to support current employees to become more active and more visible."

Lessons

What we'll change

- better preparation of materials
- more time spent checking in regularly with participants

Advice for other organizations

- Start small
- Protected time is essential
- Have measurable goals (e.g. reviews, commits)

Wrapping Up

There is nothing unique to EDB here

- There are possible areas of fruitful cooperation here

Takeaways

- Postgres talent can be grown, not just hired
- Curriculum matters; mentorship and time matter more
- The community wins when more companies do this

Thank you

- Questions, war stories, and skepticism all welcome
- Find me on email, Discord, or in the hallway
- andrew.dunstan@enterprisedb.com

Feedback Link