Engineer

Designed a function-first PostgreSQL data layer

Designed a PostgreSQL function-first data layer — 1,275 stored functions across 34 schemas — so every read and write goes through a function the database can grant, rather than through a table.

2025

1,275

stored functions guarding every read and write

Situation. Business logic has a way of leaking. A bit ends up in the API, a bit in some SQL a handler runs inline, and before long the same rule is written two or three slightly different ways and there's nowhere you can point to and say "this is what the system does with its data." That's how bugs and security holes get in.

Task. The goal was one home for all of it: every read and write going through the database, the API a thin adapter that doesn't know the business rules, and the whole thing lockable down tight.

Action. The data layer is function-first. The schema is split by domain — identity, organization, review, message, notification and 29 more, 34 in all — and every operation the app can do is one of 1,275 PostgreSQL functions it calls; there's no direct table access from Go at all. Then the database enforces it. The role the API logs in as, mariner, has EXECUTE on the app functions and USAGE on the schemas and nothing else — no SELECT, no INSERT, no way to touch a table directly — which comes to roughly 4,000 explicit grants rather than a blanket one. The functions run SECURITY DEFINER, owned by a separate non-login function_owner role with a pinned search_path, and the superuser account stays reserved for migrations and cron, well away from the running app.

Result. The logic lives in one place you can actually audit, the API stays thin and boring in the good way, and the access boundary is enforced by Postgres itself rather than by everyone remembering the rules. If the API were somehow compromised, it still couldn't do anything the functions don't allow. At 1,275 functions the discipline costs something real — a new field is a migration and a function change, not a line in a query — and that friction is the price of the boundary holding.

Bring this experience to your project

Use "Designed a function-first PostgreSQL data layer" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. The shape of a system worked out on paper first, so the parts fit and the cost is known before anyone writes code.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Designed a function-first PostgreSQL data layer — 1,275 stored functions guarding every read and write

Explore our services Get in touch →

Want your business logic in one auditable place?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Database Engineer. It draws on Platform Architecture, Backend Engineering, Databases, PostgreSQL, SQL, Data Governance and Security. And while you are here, we are rather good at Platform & Solution Architecture, Backend & API Development, Database Design & Modeling and Data Governance & Quality.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Want your business logic in one auditable place?

Ask about the decisions behind “Designed a function-first PostgreSQL data layer”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/portfolio/designed-a-function-first-postgresql-data-layer/