Engineer Software

Har designet et funktionsbaseret Postgres-datalag

Har designet et funktionsbaseret datalag i PostgreSQL — 1.275 lagrede funktioner fordelt på 34 skemaer — så al læsning og skrivning går gennem en funktion, databasen kan tildele rettigheder til, og ikke direkte gennem en tabel.

2025

1,275

stored functions, der vogter hver læsning og skrivning

Situation. Forretningslogik har det med at lække. Lidt ender i API'et, lidt i noget SQL, som en handler kører inline, og inden længe er den samme regel skrevet på to-tre lidt forskellige måder, og der er intet sted, man kan pege på og sige "det her er, hvad systemet gør med sine data." Det er sådan, fejl og sikkerhedshuller sniger sig ind.

Opgave. Målet var at samle det hele ét sted: al læsning og skrivning skulle gå gennem databasen, API'et skulle være en tynd adapter, der ikke kender forretningsreglerne, og det hele skulle kunne låses stramt af.

Handling. Datalaget er funktionsbaseret. Skemaet er delt op efter domæne — identity, organization, review, message, notification og 29 andre, 34 i alt — og hver operation, appen kan udføre, er én af 1.275 PostgreSQL-funktioner, som den kalder; der er slet ingen direkte tabeladgang fra Go. Og databasen håndhæver det. Den rolle, API'et logger ind som, mariner, har EXECUTE på app-funktionerne og USAGE på skemaerne og intet andet — ingen SELECT, ingen INSERT, ingen mulighed for at røre en tabel direkte — hvilket løber op i cirka 4.000 eksplicitte grants i stedet for én generel. Funktionerne kører som SECURITY DEFINER og ejes af en separat function_owner-rolle uden login og med en fastlåst search_path, og superuser-kontoen er forbeholdt migrationer og cron, langt væk fra den kørende app.

Resultat. Logikken ligger ét sted, som man rent faktisk kan gennemgå, API'et forbliver tyndt og kedeligt i ordets gode forstand, og adgangsgrænsen håndhæves af Postgres selv og ikke af, at alle husker reglerne. Hvis API'et på en eller anden måde blev kompromitteret, kunne det stadig ikke gøre noget, funktionerne ikke tillader. Med 1.275 funktioner har disciplinen en reel pris — et nyt felt er en migration og en funktionsændring, ikke en linje i en databaseforespørgsel — og den friktion er prisen for, at grænsen holder.

Brug erfaringen i jeres projekt

Brug »Har designet et funktionsbaseret Postgres-datalag« som udgangspunkt for en lignende udfordring i jeres organisation. Ydelserne bag arbejdet kan også bruges i jeres projekt. Systemets form lagt fast på papir først, så delene passer sammen, og prisen er kendt, før nogen skriver kode.

Se vores ydelser Kontakt os →

Se bag resultatet

Læs om situationen, beslutningerne og resultatet i eksemplet. Vi kan drøfte, hvad der passer til jeres vilkår, og hvad der kræver en anden tilgang. Har designet et funktionsbaseret Postgres-datalag — 1,275 stored functions, der vogter hver læsning og skrivning

Se vores ydelser Kontakt os →

Vil du have forretningslogikken samlet ét sted, hvor den kan gennemgås?

Tal om jeres projekt med den ingeniør, der har ansvaret for leverancen. Vi aftaler omfang, prisgrundlag og overdragelse skriftligt, før arbejdet begynder.

Rolle i opgaven: Databaseingeniør. Det trækker på Platformarkitektur, Backend-udvikling, Databaser, PostgreSQL, SQL, Data governance og Sikkerhed. Og når I alligevel er her, er vi ret dygtige til Platform- & løsningsarkitektur, Backend- & API-udvikling, Databasedesign & datamodellering og Data governance & datakvalitet.

Kontakt os Se vores ydelser

Ét stykke af **en længere fortegnelse** — og det hele ligger her på sitet.

Hver post er skrevet på samme måde: situationen, opgaven, hvad vi gjorde, og hvad der ændrede sig.

Se hele fortegnelsen

Definér den ønskede forbedring

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Vil du have forretningslogikken samlet ét sted, hvor den kan gennemgås?

Spørg til beslutningerne bag »Har designet et funktionsbaseret Postgres-datalag«. I taler med mennesker, der kan forklare arbejdet og vurdere jeres eget projekt.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://software.engineer.company/da/portfolio/designed-a-function-first-postgresql-data-layer/