Engineer Mjukvara

Utformade ett funktionsbaserat skikt i PostgreSQL

Utformade ett funktionsbaserat dataskikt i PostgreSQL — 1 275 lagrade funktioner i 34 scheman — så att varje läsning och skrivning går via en funktion som databasen kan ge behörighet till, i stället för via en tabell.

2025

1,275

lagrade funktioner som vaktar
varje läsning och skrivning

Situation. Affärslogik har en tendens att läcka. Lite hamnar i API:et, lite i någon SQL som en hanterare kör direkt i koden, och snart är samma regel skriven på två eller tre lite olika sätt och det finns ingenstans att peka och säga ”det här är vad systemet gör med sina data”. Det är så buggar och säkerhetshål kommer in.

Uppgift. Målet var ett enda hem för alltihop: varje läsning och skrivning skulle gå genom databasen, API:et skulle vara en tunn adapter som inte känner till affärsreglerna, och det hela skulle gå att låsa ordentligt.

Handling. Datalagret är byggt med funktionerna först. Schemat är uppdelat per domän — identity, organization, review, message, notification och 29 till, 34 totalt — och varje operation som appen kan utföra är en av 1 275 PostgreSQL-funktioner som den anropar; det finns ingen direkt tabellåtkomst från Go över huvud taget. Sedan ser databasen till att det efterlevs. Rollen som API:et loggar in som, mariner, har EXECUTE på appens funktioner och USAGE på schemana och inget annat — ingen SELECT, ingen INSERT, inget sätt att röra en tabell direkt — vilket blir ungefär 4 000 explicita behörigheter i stället för en generell. Funktionerna körs som SECURITY DEFINER och ägs av en separat roll utan inloggning, function_owner, med en fast search_path, och superanvändarkontot är reserverat för migreringar och cron, långt från den körande appen.

Resultat. Logiken bor på ett ställe som faktiskt går att granska, API:et förblir tunt och tråkigt på det bra sättet, och åtkomstgränsen upprätthålls av själva Postgres i stället för av att alla kommer ihåg reglerna. Om API:et på något sätt kompromitterades skulle det ändå inte kunna göra något som funktionerna inte tillåter. Med 1 275 funktioner kostar disciplinen på riktigt — ett nytt fält är en migrering och en funktionsändring, inte en rad i en SQL-fråga — och den friktionen är priset för att gränsen håller.

Använd erfarenheten i ert projekt

Använd ”Utformade ett funktionsbaserat skikt i PostgreSQL” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. Systemets form utarbetad på papper först, så att delarna passar ihop och kostnaden är känd innan någon skriver kod.

Se våra tjänster Hör av er →

Se bakom resultatet

Läs om situationen, besluten och resultatet i exemplet. Vi kan diskutera vad som passar era förutsättningar och vad som kräver ett annat angreppssätt. Utformade ett funktionsbaserat skikt i PostgreSQL — 1,275 lagrade funktioner som vaktar varje läsning och skrivning

Se våra tjänster Hör av er →

Vill du ha din affärslogik på ett ställe som går att granska?

Prata om ert projekt med ingenjören som ansvarar för leveransen. Vi avtalar omfattning, hur priset beräknas och överlämning skriftligt innan arbetet börjar.

Roll i uppdraget: Databasingenjör. Det bygger på Plattformsarkitektur, Backendutveckling, Databaser, PostgreSQL, SQL, Datastyrning och Säkerhet. Och nu när ni är här: vi är ganska bra på Plattforms- och lösningsarkitektur, Backend- och API-utveckling, Databasdesign och datamodellering och Datastyrning och datakvalitet.

Hör av er Se våra tjänster

En del av **en längre historik** — och den finns i sin helhet här.

Varje post är skriven på samma sätt: situationen, uppgiften, vad vi gjorde och vad som förändrades.

Se hela historiken

Definiera förbättringen ni behöver

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Vill du ha din affärslogik på ett ställe som går att granska?

Fråga om besluten bakom ”Utformade ett funktionsbaserat skikt i PostgreSQL”. Ni talar med personer som kan förklara arbetet och bedöma ert eget projekt.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://software.engineer.company/sv/portfolio/designed-a-function-first-postgresql-data-layer/