Engineer

Couche PostgreSQL fondée sur les fonctions

Conçu une couche de données fondée sur les fonctions PostgreSQL — 1 275 fonctions stockées réparties dans 34 schémas — pour que chaque lecture et écriture passe par une fonction dont les droits sont contrôlables, plutôt que par une table.

2025

1,275

fonctions stockées protégeant chaque lecture et écriture

Situation. La logique métier a tendance à déborder. Un morceau finit dans l'API, un autre dans du SQL exécuté directement par un gestionnaire, et bientôt la même règle existe sous deux ou trois variantes légèrement différentes, sans qu'aucun endroit permette de dire « voici ce que le système fait de ses données ». C'est ainsi que s'introduisent les défauts et les failles de sécurité.

Tâche. L'objectif était de tout regrouper en un seul endroit : chaque lecture et écriture passant par la base de données, l'API restant un adaptateur mince qui ignore les règles métier, et l'ensemble pouvant être strictement verrouillé.

Action. La couche de données est organisée autour des fonctions. Le schéma est réparti par domaine — identity, organization, review, message, notification et 29 autres, soit 34 au total — et chaque opération possible de l'application correspond à l'une des 1 275 fonctions PostgreSQL qu'elle appelle ; Go n'accède jamais directement aux tables. La base fait ensuite respecter cette règle. Le rôle utilisé par l'API pour se connecter, mariner, possède EXECUTE sur les fonctions applicatives et USAGE sur les schémas, et rien d'autre — aucun SELECT, aucun INSERT, aucun moyen de toucher directement une table — ce qui représente environ 4 000 autorisations explicites plutôt qu'une autorisation globale. Les fonctions s'exécutent en SECURITY DEFINER, appartiennent à un

rôle distinct function_owner sans connexion, avec un search_path fixé, et le compte superutilisateur reste réservé aux migrations et à cron, bien à l'écart de l'application en fonctionnement.

Résultat. La logique réside en un seul endroit réellement auditable, l'API reste mince et banale au bon sens du terme, et la frontière d'accès est imposée par Postgres lui-même plutôt que par la mémoire des règles de chacun. Même si l'API était compromise, elle ne pourrait toujours rien faire que les fonctions n'autorisent pas. Avec 1 275 fonctions, cette discipline a un coût réel — un nouveau champ exige une migration et une modification de fonction, pas une ligne dans une requête — et cette friction est le prix d'une frontière qui tient.

Mettez cette expérience au service de votre projet

Prenez « Couche PostgreSQL fondée sur les fonctions » comme point de départ pour un défi similaire dans votre organisation. Les prestations mobilisées pour cette réalisation sont disponibles pour votre projet. Définir d'abord le système sur le papier pour que ses éléments s'articulent et que son coût soit connu avant d'écrire du code.

Découvrir nos prestations Contactez-nous →

Examinez ce qui a conduit au résultat

Découvrez la situation, les décisions et le résultat de cet exemple. Nous pouvons discuter des éléments transposables à vos contraintes et de ceux qui nécessitent une autre démarche. Couche PostgreSQL fondée sur les fonctions — 1,275 fonctions stockées protégeant chaque lecture et écriture

Découvrir nos prestations Contactez-nous →

Envie d'une logique métier regroupée et vérifiable ?

Discutez de votre projet avec l'ingénieure responsable de sa réalisation. Nous convenons par écrit du périmètre, de la base tarifaire et de la transmission avant le début des travaux.

Rôle exercé dans ce contexte : Ingénieure bases de données. Cette réalisation fait appel à Architecture de plateformes, Ingénierie backend, Bases de données, PostgreSQL, SQL, Gouvernance des données et Sécurité. Et puisque vous êtes ici, nous sommes aussi plutôt doués en Architecture de plateformes et de solutions, Développement backend et API, Conception et modélisation de bases de données et Gouvernance et qualité des données.

Contactez-nous Découvrir nos prestations

Un élément **d'un parcours plus long** —
entièrement présenté sur ce site.

Chaque entrée suit la même structure : la situation, la tâche, ce que nous avons fait et ce qui a changé.

Voir l'ensemble du parcours

Définissez l'amélioration nécessaire

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Envie d'une logique métier regroupée et vérifiable ?

Interrogez-nous sur les décisions qui sous-tendent « Couche PostgreSQL fondée sur les fonctions ». Vous parlerez aux personnes capables d'expliquer le travail et d'évaluer votre propre projet.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/portfolio/designed-a-function-first-postgresql-data-layer/