

Modèle de Données de l'Application

1. Introduction

Ce document décrit la structure de la base de données pour l'application.

Le modèle est conçu pour être robuste, évolutif et pour supporter l'ensemble des fonctionnalités requises, de la saisie des besoins à la génération de rapports, en passant par les workflows de validation et l'historisation des modifications.

2. Énumérations (Types de Données Prédéfinis)

Pour garantir la cohérence des données, des types énumérés sont utilisés pour les champs critiques :

- **roleEnum** : Définit les rôles des utilisateurs.
 - user : Personnel de service (saisie).
 - chef_dept : Chef de département (consolidation).
 - direction : Direction (validation finale).
 - comptable : Comptable/Auditeur (consultation, analyse).
- **statusEnum** : Représente le cycle de vie d'une ligne budgétaire.
 - draft : Brouillon, non encore soumis.
 - pending : Soumis, en attente de validation.
 - validated : Approuvé par le niveau supérieur.
 - rejected : Refusé.
 - consolidated : Inclus dans un budget consolidé.
- **budgetTypeEnum** : Spécifie la nature d'une catégorie budgétaire.
 - recette : Entrée de fonds.
 - depense : Sortie de fonds.

3. Description des Tables

3.1. Table users

Stocke les informations relatives aux comptes des utilisateurs de la plateforme.

Champ	Type	Description
id	serial	Identifiant unique de l'utilisateur (Clé primaire).
email	text	Adresse email de l'utilisateur,

		utilisée pour la connexion (unique).
password	text	Mot de passe de l'utilisateur (doit être haché).
name	text	Nom complet de l'utilisateur.
role	roleEnum	Rôle de l'utilisateur, détermine ses permissions.
department	text	Département d'appartenance de l'utilisateur.
isActive	boolean	Indique si le compte est actif ou désactivé.
createdAt	timestamp	Date et heure de création du compte.

3.2. Table budgetCategories

Contient la nomenclature budgétaire officielle.

Champ	Type	Description
id	serial	Identifiant unique de la catégorie (Clé primaire).
code	text	Code officiel de la rubrique budgétaire (ex: "60411").
label	text	Libellé complet de la rubrique (ex: "Consommables informatiques").
type	budgetTypeEnum	Nature de la rubrique (recette ou dépense).
parentCode	text	Code de la catégorie parente, pour une structure hiérarchique.
description	text	Description détaillée de ce que la rubrique inclut.

isActive	boolean	Indique si la rubrique est actuellement utilisée.
----------	---------	---

3.3. Table budgetLines

Table centrale qui enregistre chaque besoin budgétaire exprimé.

Champ	Type	Description
id	serial	Identifiant unique de la ligne budgétaire (Clé primaire).
userId	integer	Référence à l'utilisateur qui a créé la ligne.
categoryId	integer	Référence à la rubrique budgétaire concernée.
year	integer	Année budgétaire pour laquelle le besoin est exprimé.
proposedAmount	numeric	Montant proposé pour cette ligne.
realizedAmount	numeric	Montant réellement dépensé/reçu (pour l'analyse N-1).
description	text	Justification ou détail supplémentaire sur le besoin.
status	statusEnum	Statut actuel de la ligne dans le workflow de validation.
validatedBy	integer	Référence à l'utilisateur qui a validé/rejeté la ligne.
validatedAt	timestamp	Date de la dernière action de validation.
rejectionReason	text	Motif en cas de rejet de la ligne.
createdAt	timestamp	Date de création de la ligne.
updatedAt	timestamp	Date de la dernière

		modification.
--	--	---------------

3.4. Table budgetHistory

Assure la traçabilité en historisant toutes les modifications apportées aux lignes budgétaires.

Champ	Type	Description
id	serial	Identifiant unique de l'enregistrement d'historique.
budgetLineId	integer	Référence à la ligne budgétaire modifiée.
action	text	Type de modification (ex: "CREATE", "UPDATE_AMOUNT", "STATUS_CHANGE").
oldValues	text	Anciennes valeurs (stockées en JSON) avant la modification.
newValues	text	Nouvelles valeurs (stockées en JSON) après la modification.
userId	integer	Référence à l'utilisateur qui a effectué la modification.
createdAt	timestamp	Date de l'enregistrement de la modification.

3.5. Table budgetReports

Stocke les informations sur les rapports budgétaires générés (PDF, Excel).

Champ	Type	Description
id	serial	Identifiant unique du rapport.
userId	integer	Référence à l'utilisateur qui a généré le rapport.

year	integer	Année budgétaire concernée par le rapport.
type	text	Type de rapport (ex: "PROJET_CONSOLIDE", "ANALYSE_ECARTS").
filename	text	Nom du fichier généré.
filePath	text	Chemin de stockage du fichier sur le serveur.
fileSize	integer	Taille du fichier en octets.
createdAt	timestamp	Date de génération du rapport.

4. Relations entre les Tables

Les relations définissent comment les tables sont connectées entre elles :

- Un **Utilisateur** (users) peut créer plusieurs **Lignes Budgétaires** (budgetLines) et plusieurs **Rapports** (budgetReports).
- Une **Ligne Budgétaire** (budgetLines) appartient à un seul **Utilisateur** et à une seule **Catégorie Budgétaire** (budgetCategories).
- Chaque **Ligne Budgétaire** peut avoir un historique complet de ses modifications, stocké dans la table **budgetHistory**.

5. Schémas de Validation (Zod)

Pour garantir l'intégrité des données avant leur insertion en base, des schémas de validation sont définis avec la bibliothèque **Zod**. Ces schémas (insertUserSchema, insertBudgetLineSchema, etc.) assurent que les données envoyées par le client à l'API respectent le format et les contraintes attendus (type de données, champs obligatoires, etc.), renforçant ainsi la robustesse de l'application.