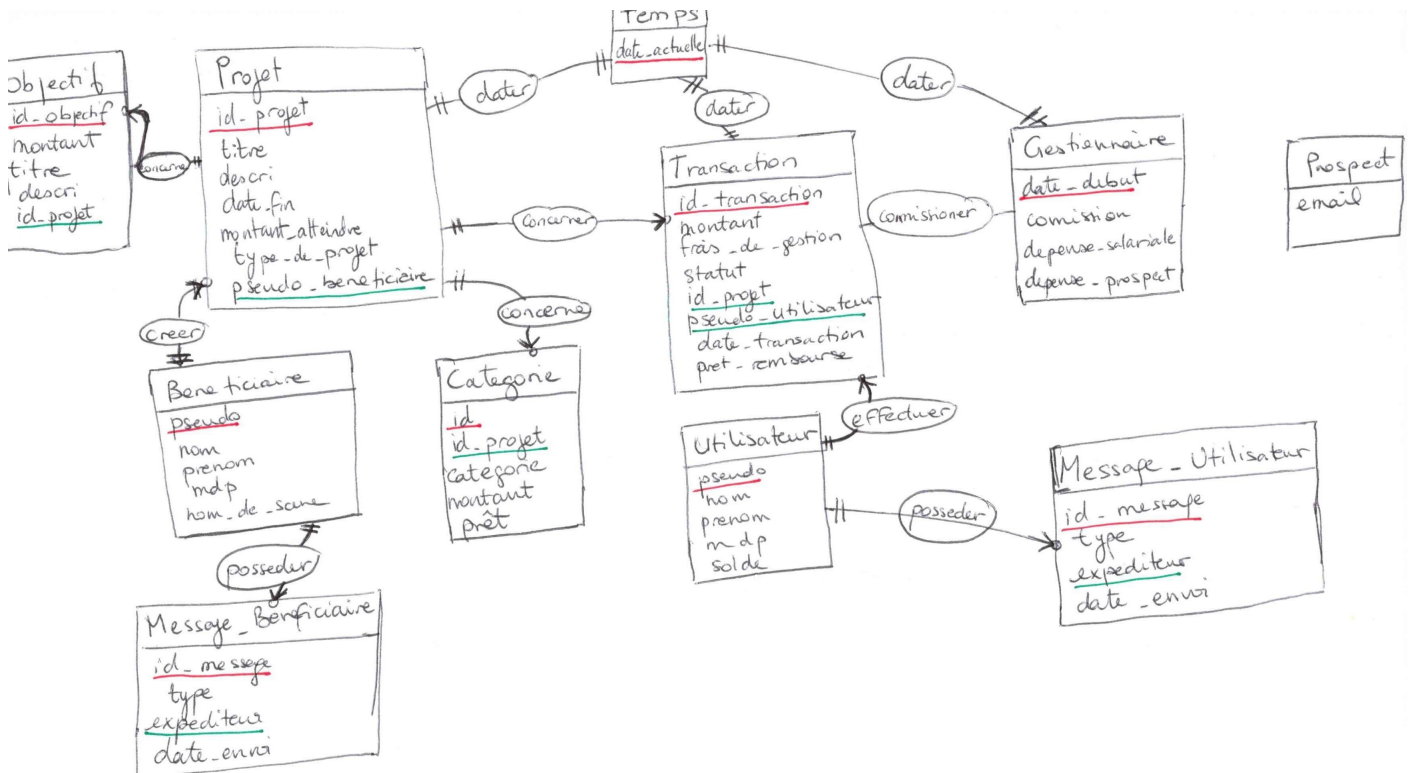


Base de Données Avancées Projet : Financement de Projets

Par Etienne Jouanne et Wilfried Njangu

Schéma relationnel BDA



(Les attributs du schéma ne sont pas exacte, nous avons fait quelques modifications au fur et a mesure. Vous pouvez retrouver ce schéma dans le dossier de rendu ainsi qu'un autre schéma avec les bons attributs)

Contraintes

Utilisateur (pseudo, nom, prenom, mdp, solde) :

pseudo -> nom, prenom, mdp, solde

Bénéficiaire (pseudo, nom, prenom, mdp, nom de scene) :

pseudo -> nom, prenom, mdp, nom de scene

Projet (id_projet, nom, descri, date_fin, pseudo_beneficiaire, montant_atteindre) :

id_projet-> nom, descri, date_fin, pseudo_beneficiare, montant_atteindre

Transaction (id_transaction, montant, frais_de_gestion, statut, id_projet, pseudo_utilisateur, date_transaction, prêt_rembourse) :

id_transaction -> montant, frais_de_gestion, statut, id_projet,
pseudo_utilisateur, date_transaction, prêt_rembourse

Gestionnaire (date_debut, commission, depense_salariale, depense_prospect) :

date_debut -> commission, depense_salariale, depense_prospect

Objectif (id_objectif, montant, titre, descri) :

id_objectif -> montant, titre, descri

Categorie (id, id_projet, categorie, montant, prêt) :

id -> id_projet, categorie, montant, prêt

Message_Utilisateur (id_message, type, expéditeur, date_envoi) :

id_message -> type, expéditeur, date_envoi

Message_Beneficiaire (id_message, type, expéditeur, date_envoi) :

id_message -> type, expéditeur, date_envoi

Temps (date_actuelle) :

date_actuelle

Prospect (email) :

email

Listes fonctions, checks et triggers

- **Check :**

- Utilisateur :
Solde > 0
- Projet :
date_fin >= CURRENT_DATE
- Transaction :
montant > 0
frais de gestion > 0
- Gestionnaire :
commission > 0
dépende salariale > 0
dépendes prospect > 0
- Catégorie :
montant > 0

- **Fonctions :**

Creation d'objet pour toutes les tables (ces fonctions commencent par " creation ").

getTemps qui retourne le temps actuelle dans la base de données.

addOneDay... qui ajout 1 journée a la table Temps.

Rentable qui verifie que le site est rentable.

- **Triggers :**

Transaction :

- Avant création transaction : vérifier que le solde de l'utilisateur moins le don ne soit pas inférieur à 0. Si oui, retourne une exception. Effectue la transaction sinon.
- Avant création transaction : si le montant dépasse une certaine somme de catégorie de donateur que le bénéficiaire a mis, alors le bénéficiaire est obligé de lui rembourser son prêt.
- Après création transaction : Envoie de message à l'utilisateur et au bénéficiaire à propos du don.

Temps :

- Après modification temps : Clôture de projet, si l'objectif non atteint, rembourser (avec frais de gestion). Sinon, prélève les donations qui ont été fait sous forme de prêt.
- Après modification temps : si la date de fin du projet moins la date dans Temps est égale a 2 et que la somme des dons récoltés est inferieur à sa moitié (appelé pour nous en phase critique), alors envoyer un message à tous les utilisateurs qui n'ont pas encore fait de dons au projet.

Projet :

- Avant création d'un nouveau projet : vérifier que les prêts précédents ont été remboursés.

Prospect :

- Après ajout du prospect : lui envoie un mail (fictif).

Tests

- Donations sur un projet :
 - ◆ Création d'une transaction qui est inférieure a une catégorie (de donateur) (pas de prêt a rembourser) :
 - Le créateur du projet peut créer un nouveau projet . (Test1)
 - ◆ Création d'une transaction qui est supérieur a une catégorie (de donateur) :
 - Le créateur du projet ne peut pas créer de nouveau projet.
 - Le donateur a bien reçu son prêt.
 - Le créateur peut de nouveau créer un projet. (Test2)
 - ◆ Création d'une transaction qui fait passer la somme totale récolter a un nouvel objectif et affiche les messages reçu par l'utilisateur et le bénéficiaire(Test3).
- Test4 :
 - ◆ Augmentation du temps qui fait passer un projet en phase critique (J-2 avant la clôture et somme récolte < somme attendu/2) : on affiche les messages reçu par les utilisateurs qui non pas fait de dons.
 - ◆ Rentabilité : on vérifie si les frais de gestion récolte sont supérieur aux dépenses salariale et dépenses prospect.
 - ◆ Ajout d'un prospect.
 - ◆ Invitation d'un utilisateur a un autre utilisateur pour faire un don sur un certain projet