

Etienne Juan-Gesta  
Sébastien Lecleire

## Base de Données Avancées – Projet

Sujet:

Crowdfunding pour financement de groupe de musique

Tables:

### **Table utilisateurs:**

Tous les utilisateurs ont un nom, un prénom, un age, une adresse, un mail, le niveau global (qui dépend du montant total donné), actif qui est un booléen qui indique si l'utilisateur est actif ou pas, la date d'inscription et date de dernière connexion.

#### Règles de gestion:

-Clé primaire: id\_utilisateur.

-Réglé par check: age > 18 ans

-Réglés par trigger: niveau global, actif, date\_inscription, date\_dernière\_co

-Nullable: rien.

### **Table Projet:**

Il a un id projet, un nom, un montant de base ( montant a atteindre), un montant maximum autorisé, le montant actuel atteint, une description, une date de création, une date limite, la date du dernier don, et un booléen pour savoir si le projet est encore actif.

#### Règles de gestion:

-Clé primaire: id\_projet.

-Clé étrangère: id\_initiateur.

Réglés par check:

Création d'un projet, date limite > date de création, montant\_actuel < montant\_max, montant\_base > 0 et montant\_base < montant\_max

-Réglés par trigger: date\_dernière\_donation, actif

-Nullable: date\_dernier\_don

#### **Table Initiateur:**

Un initiateur a un id initiateur, un id utilisateur et un id projet. Il a également une note de fiabilité et une description.

##### Règles de gestion:

-Clé primaire: id\_initiateur.

-Clé étrangère: id\_utilisateur, id\_projet.

-Réglés par fonction: fiabilité.

-Nullable: description, fiabilité

#### **Table Donateur:**

Un donateur a un id donateur, un id utilisateur et un id projet. On a également un montant et un niveau.

Il s'agit en fait de son statut de donateur pour 1 projet donné.

Il y a donc le montant total donné et son niveau de donateur pour ce projet.

##### Règles de gestion:

-Clé primaire: id\_donateur.

-Clé étrangère: id\_utilisateur, id\_projet.

Réglé par trigger: niveau

-Nullable: rien

### **Table Bénéficiaire:**

Un des bénéficiaires du projet. Il a un id bénéficiaire, un id utilisateur et un idprojet.

On a son rôle dans le groupe, le montant qui est censée lui être reversé en pourcentage ainsi que le montant qui lui a été reversé une fois le projet fini via la fonction redistribute().

#### Règles de gestion:

- Clé primaire: id\_bénéficiaire.
- Clé étrangère: id\_utilisateur, id\_projet.
- Nullable: rôle.

### **Table log\_utilisateur:**

Il y a un id\_log\_u, une old\_value, une new\_value, une date\_action, , une catégorie .

#### Règles de gestion:

- Clé primaire: id\_log\_u
- Réglés par trigger: tout
- Nullable: rien

### **Table log\_projet :**

Il y a un id\_log\_p, une old\_value, une new\_value, une date\_action, , une catégorie .

#### Règles de gestion:

- Clé primaire: id\_log\_p
- Réglés par trigger: tout
- Nullable: rien

**Index :**

Il y a 3 index :

X\_PROJ\_DATE\_CREALIM ON projets (date\_creation, date\_limite);

pour accélérer les recherches sur les projets qui prennent en paramètre des dates et faire des stats

X\_DONA\_MONTNIV ON donateurs (montant, niveau);

pour accélérer les requêtes portant sur le montant donné et le niveau d'un donateur, très pratique pour faire des dons

X\_UTI\_MAIL ON utilisateurs (mail);

pour accélérer les checks des mails

**Triggers :**

Chaque opération d'insertion, de suppression et de mise à jour quelle que soit la table est enregistrée dans les tables de log prévu à cet effet.

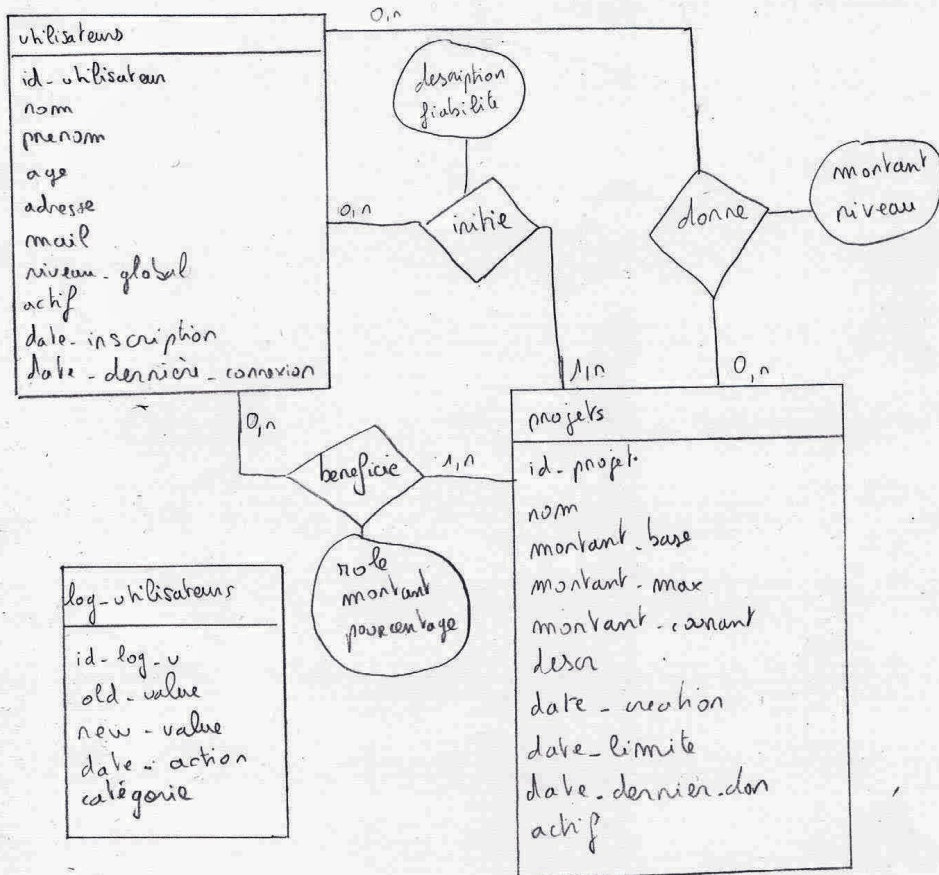
On a des triggers permettant la vérification des paiements fait au bénéficiaire et sur les dons fait par un donateur à un projet, des triggers permettant de vérifier si on ne promet pas plus de 100% du montant aux bénéficiaires... (voir triggers.sql)

**Fonctions :**

- Fonctions permettant de créer un projet (en vérifiant toutes les contraintes d'un projet)
- Fonctions permettant d'arrêter un projet de force ou non si le montant de base est dépassé, ou en cas d'abandon du projet
- Fonctions permettant de créer une ligne dans une table (ex : un utilisateur, un projet, ...)
- Fonctions permettant d'inviter un utilisateur a devenir bénéficiaire d'un projet
- Fonctions permettant a un utilisateur de donner de l'argent a un projet (faire un don)
- Fonction permettant de redistribuer l'argent selon le pourcentage
- Fonction permettant de faire passer de niveau selon le montant donné
- Fonctions permettant de récupérer des données, des logs, adéquates selon le statut de l'utilisateur (ex : moyenne des dons)
- Fonctions permettant d'exécuter les triggers (valable pour tous les triggers car tous les triggers sont liés à une fonction)
- Fonctions de vérification lors de requêtes sensibles (dons, paiement, ajout d'un projet, ...)
- Fonctions permettant de tester le cycle de vie d'un projet
- Fonctions permettant de tester les contraintes d'intégrité du projet (ex : age, montant, ...)
- Fonctions permettant d'éviter des opérations non autorisées (ex : dons avec montant négatif , ... )
- Fonctions permettant la redistribution du montant donné aux bénéficiaires

Tests :

- 1/Un utilisateur trop jeune (-18 ans) tente de s'inscrire.
- 2/Un utilisateur tente de créer un projet avec un montant cible négatif
- 3/Un utilisateur tente de faire un don négatif
- 4/Un utilisateur crée un projet, puis un autre essaye de donner pour plus que le montant maximal autorisé.
- 5/Test de creation d'un projet
- 6/Simulation d'un projet en entier qui démarre, a des dons, dépasse son montant cible, et redistribue aux bénéficiaires.
- 7/Simulation d'un projet qui n'a pas assez de dons et qui essaye de terminer.
- 8/Simulation d'un projet qui n'a pas assez de dons et qui force la fin (echec du projet).
- 9/Essai de création d'un projet avec date cible avant date de création.
- 10/Un utilisateur essaye d'être initiateur de 2 projets en meme temps (interdit)
- 11/Un utilisateur crée un projet, il a quelques dons puis décide de créer un autre projet avant que son premier projet soit fini.  
  
Il constate qu'il ne peut pas et attends que son premier projet termine. Il lance alors son 2<sup>e</sup> projet
- 12/Un initiateur promet plus de 100% des recettes aux bénéficiaires du projet



Entity-relationship model