26 novembre 2017

Thibault Cothenet

antoine julien

ERIC CERVEAU

projet tutoré s3

gestion des goûters à l’école de rugby d’Orsay

Introduction

1. Résumé du contexte et objectif de l’application

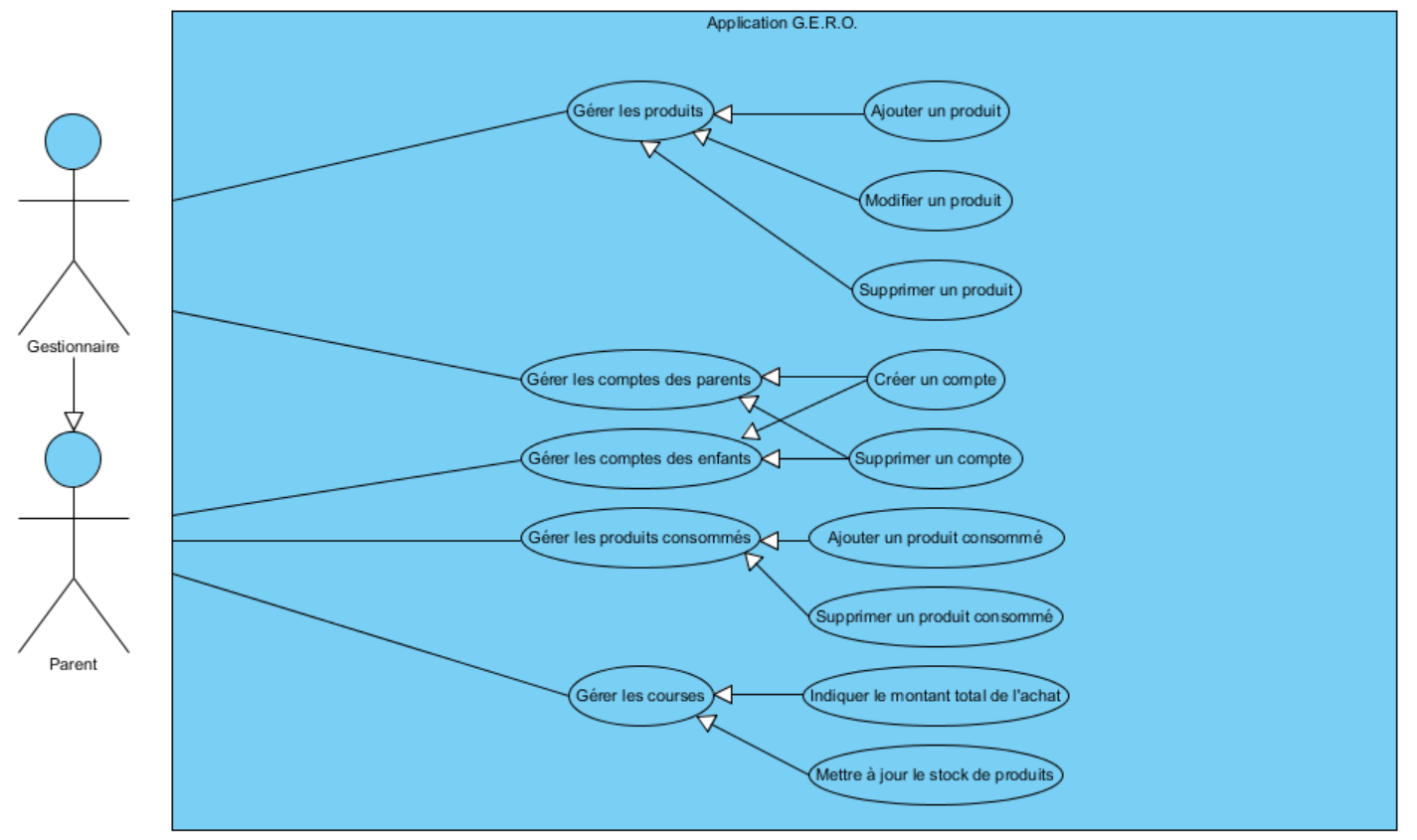
L’association APERO (Association des Parents de l’Ecole de Rugby d’Orsay) distribue des goûters aux élèves de l’école de Rugby d’Orsay deux fois par semaine après leurs entraînements. La gestion de ces goûters se faisant manuellement, cette dernière reste très compliquée. Afin de remplacer l’ancien système manuel nous avons créé l’application GERO (Goûter pour l’Ecole de Rugby d’Orsay).

GERO a pour but de faciliter la distribution et la gestion des gouters de l’école de rugby d’Orsay. Grâce à une implémentation rigoureuse GERO est efficace, simple et rapide à utiliser. Elle est destinée aux parents bénévoles ainsi qu’au président de l’application.

1. Définition des acteurs de l’application

On distingue deux types d’utilisateur :   
- Les parents membres de l’association. Ils ont la possibilité de créer et de supprimer le compte d’un enfant. Ils doivent ajouter ou de supprimer les produits consommés chaque jour pour chaque enfant. Ils ont aussi la possibilité de créditer le compte d’un enfant d’une certaine somme d’argent. Enfin, lorsqu’ils font les courses, ils indiquent le montant total de leur achat, et mettent à jour le stock de chaque produit, en indiquant les quantités achetées de chacun des produits.   
- Le gestionnaire de l’application, c’est-à-dire le président d’APERO. Il a la possibilité de créer et de supprimer le compte d’un enfant. Il peut aussi ajouter ou supprimer des parents de l’association. Enfin, il peut aussi créer des nouveaux produits, fixer et modifier leur prix de vente.

1. Diagramme de cas d’utilisation

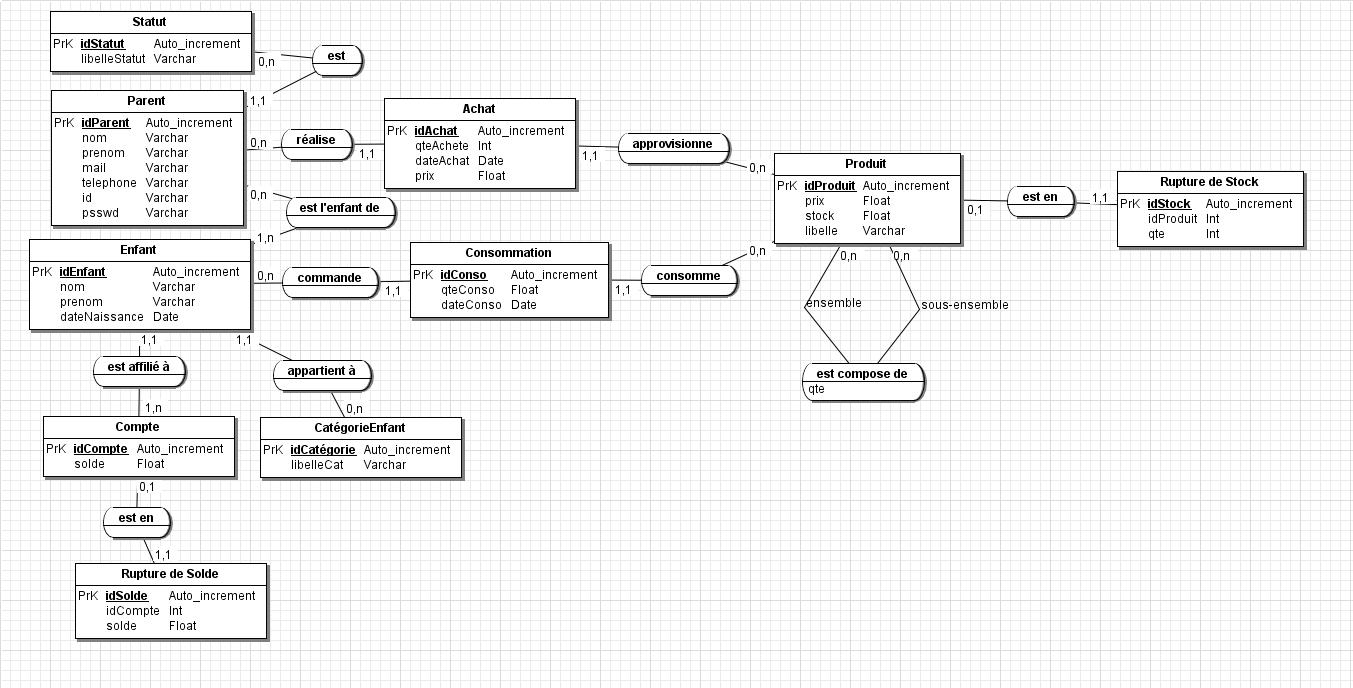


Analyse et Conception

1. Dictionnaire des données

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Code** | **Type** | **Description** |
| Parent | idParent | AUTO\_INCREMENT | Identifiant numérique d’un parent |
|  | nom | VARCHAR(20) | Nom de famille d’un parent |
|  | prenom | VARCHAR(20) | Prénom d’un parent |
|  | mail | VARCHAR(30) | Adresse e-mail d’un parent |
|  | telephone | VARCHAR(10) | Numéro de téléphone d’un parent |
|  | id | VARCHAR(20) | Login d’un parent pour se connecter à l’application |
|  | psswd | VARCHAR(20) | Mot de passe associé au login |
|  | idStatut | INT | Identifiant numérique du statut d’un parent |
|  |  |  |  |
| Enfant | idEnfant | AUTO\_INCREMENT | Identifiant numérique d’un enfant |
|  | nom | VARCHAR(20) | Nom de famille d’un enfant |
|  | prenom | VARCHAR(20) | Prénom d’un enfant |
|  | dateNaissance | DATE | Date de naissance au format JJ/MM/AAAA |
|  | idCategorie | INT | Identifiant numérique de la catégorie d’un enfant |
|  | idCompte | INT | Identifiant numérique du compte d’un enfant |
|  |  |  |  |
| Statut | idStatut | AUTO\_INCREMENT | Identifiant numérique d’un statut |
|  | libelleStatut | VARCHAR(20) | Libellé d’un statut |
|  |  |  |  |
| Compte | idCompte | AUTO\_INCREMENT | Identifiant numérique d’un compte |
|  | solde | FLOAT | Solde restant du compte exprimé en euros |
|  |  |  |  |
| CatégorieEnfant | idCategorie | AUTO\_INCREMENT | Identifiant numérique d’une catégorie |
|  | libelleCat | VARCHAR(15) | Libellé d’une catégorie |
|  |  |  |  |
| Consommation | idConso | AUTO\_INCREMENT | Identifiant numérique d’une consommation |
|  | qteConso | FLOAT | Quantité de la consommation |
|  | dateConso | DATE | Date de la consommation au format JJ/MM/AAAA |
|  | idEnfant | INT | Identifiant numérique d’un enfant |
|  | idProduit | INT | Identifiant numérique d’un produit |
|  |  |  |  |
| Achat | idAchat | AUTO\_INCREMENT | Identifiant numérique d’un achat |
|  | qteAchete | INT | Quantité d’un produit lors d’un achat |
|  | dateAchat | DATE | Date de l’achat au format JJ/MM/AA |
|  | prix | FLOAT | Prix de l’achat exprimé en euros |
|  | idParent | INT | Identifiant numérique d’un parent bénévole |
|  | idProduit | INT | Identifiant numérique d’un produit |
|  |  |  |  |
| Produit | idProduit | AUTO\_INCREMENT | Identifiant numérique d’un produit |
|  | prix | FLOAT | Prix d’un produit exprimé en euros |
|  | stock | FLOAT | Quantité d’un produit |
|  | libelle | VARCHAR(30) | Libellé d’un produit |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Code** | **Type** | **Description** |
| est l’enfant de | idParent | INT | Identifiant numérique d’un parent |
|  | idEnfant | INT | Identifiant numérique d’un enfant |
|  |  |  |  |
| est compose de | idProduit | INT | Identifiant numérique d’un produit |
|  | idProduit\_1 | INT | Identifiant numérique d’un sous-produit |
|  |  |  |  |
| rupture de stock | idStock | AUTO\_INCREMENT | Identifiant numérique d’un produit proche de la rupture de stock |
|  | idProduit | INT | Identifiant numérique d’un compte dont le solde est presque nul |
|  | qte | INT | Quantité restante d’un produit |
|  |  |  |  |
| rupture de solde | idSolde | AUTO\_INCREMENT | Identifiant numérique d’un produit |
|  | idCompte | INT | Identifiant numérique d’un compte |
|  | solde | FLOAT | Solde restant d’un compte |

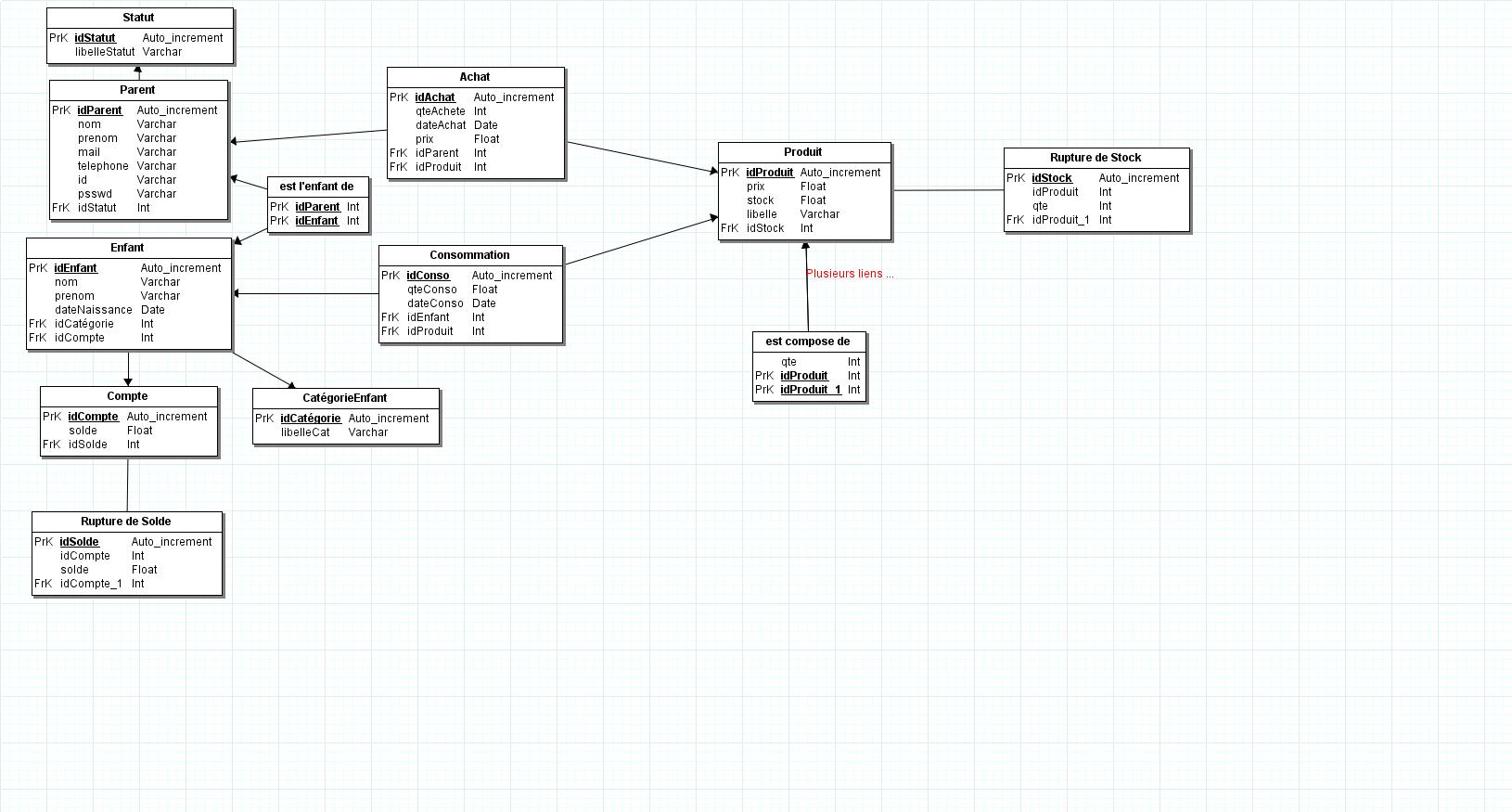
1. Modèle conceptuel de données

Pour réaliser notre application nous avons choisi le découpage de notre base de données en différentes tables :   
- Une table Parent regroupant tous les parents au sein de l’association APERO ainsi que les autres parents de l’école de rugby.  
- Une table Enfant regroupant tous les enfants au sein de l’école de rugby.   
- Une table Statut regroupant les différents statuts au sein de l’application GERO : Président, Parent bénévole ou bien Parent simple.  
- Une table Produit regroupant les différents goûters dont dispose l’association APERO.  
- Une table CatégorieEnfant regroupant les différentes catégories d’âges présentes au sein de l’école de rugby : : -8 ans, -10 ans, -12 ans, -14 ans, etc …  
  
- Une table Consommation regroupant les différentes consommations d’un enfant au cours d’un goûter.  
- Une table Achat regroupant les différentes courses réalisées par les parents bénévoles.  
- Une table Compte regroupant tous les comptes des enfants de l’école de rugby.

S’ajoute à ces tables deux tables nécessaires à nos TRIGGER :   
- Une table Rupture de Solde regroupant tous les comptes des enfants disposant d’un solde de moins de 5€ pour émettre une notification à leurs parents pour qu’ils rechargent le compte de leur enfant.  
- Une table Rupture de Stock regroupant tous les produits disposant d’un stock inférieur à 5 unités pour émettre une notification aux parents bénévoles pour qu’ils achètent ce produit aux prochaines courses.

Nous avons choisi une telle implémentation car elle nous permettra de gérer plus facilement les ruptures de stock et de solde lors de notre implémentation en PHP.

Implémentation

1. Modèle relationnel de données / Schéma relationnel
2. Code SQL de la création de la base de données

Cf. « Code Création BD »

1. Code SQL de l’insertion de données exemples

Cf. « Code Insertion BD »

1. Code des requêtes permettant de réaliser chacune des fonctionnalités de l’application

Cf. « Code Insertion BD »

1. Code des vues

Cf. « Code Vue »

1. Code des triggers

Cf. « Code Trigger »

Documentation technique

Pour installer la base de données : Lancer le logiciel XAMPP/WAMPP/LAMPP et démarrer les modules Apache et MySQL. Connecter vous à « localhost/phpmyadmin », créer une base de données puis exécuter les scripts de création des tables. Ensuite exécuter les scripts des triggers, suivis des scripts d’insertion des données, des scripts des vues et des scripts des fonctions / procédures. Votre base de données est maintenant créée et opérationnelle.

Par la suite, pour réaliser quelconque action sur la base de données vous pouvez utiliser les fonctions et procédures mises à votre disposition (ajout d’argent, gestion de stock, création de comptes, etc …).

Conclusion

Lors de cette partie Base de Données Avancée, nous avons appris à mobiliser nos connaissances du premier et deuxième semestre au sein d’un projet structuré, chose que nous n’avions pas pu faire en profondeur en classe. Parmi les problèmes rencontrés, les principaux étaient des problèmes de syntaxe : En effet, nous avions l’habitude d’Oracle et de non de mySQL. La syntaxe étant quelque peu différente il nous a fallu un temps d’adaptation avant de pouvoir réaliser certaines fonctionnalités. En fin de compte, nous avons réussi à remplir les impératifs qui nous ont été donné en construisant une base de données cohérente, opérationnelle, peuplée et munie de fonctions assurant les fonctionnalités de l’application.