

ASFAR MOHAMED RAFFIQUE

2017/2018

OTHMANE EL AYADI

YUXUAN WANG

PROJET TUTEORE S3

GESTION DES GOUTERS A L'ECOLE DE RUGBY D'ORSAY



Table des matières

I – INTRODUCTION	3
A – CONTEXTE ET OBJECTIFS	3
B – DIAGRAMME DE CAS D'UTILISATION	4
II - ANALYSE ET CONCEPTION	6
A. DICTIONNAIRE DES DONNEES	6
B. MODELE CONCEPTUEL DES DONNEES	7
1. LE MODELE	7
2. JUSTIFICATION DES CHOIX	7
III. IMPLEMENTATION	9
A. MODELE RELATIONNEL DES DONNEES	9
B. Code SQL de la création de la base de données	9
C. Code SQL de l'insertion de données exemples	9
D. Code des requêtes permettant de réaliser chacune des fonctionnalités de l'application	9
E. Code des vues	9
F. Code des triggers	9
IV- DOCUMENTATION TECHNIQUE	10
A. INSTALLATION DE LA BASE DE DONNEES	10
B. MAINTENANCE DE LA BASE DE DONNEES	13
CONCLUSION	14



I – INTRODUCTION

A – CONTEXTE ET OBJECTIFS

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.

B – DEFINITION 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.

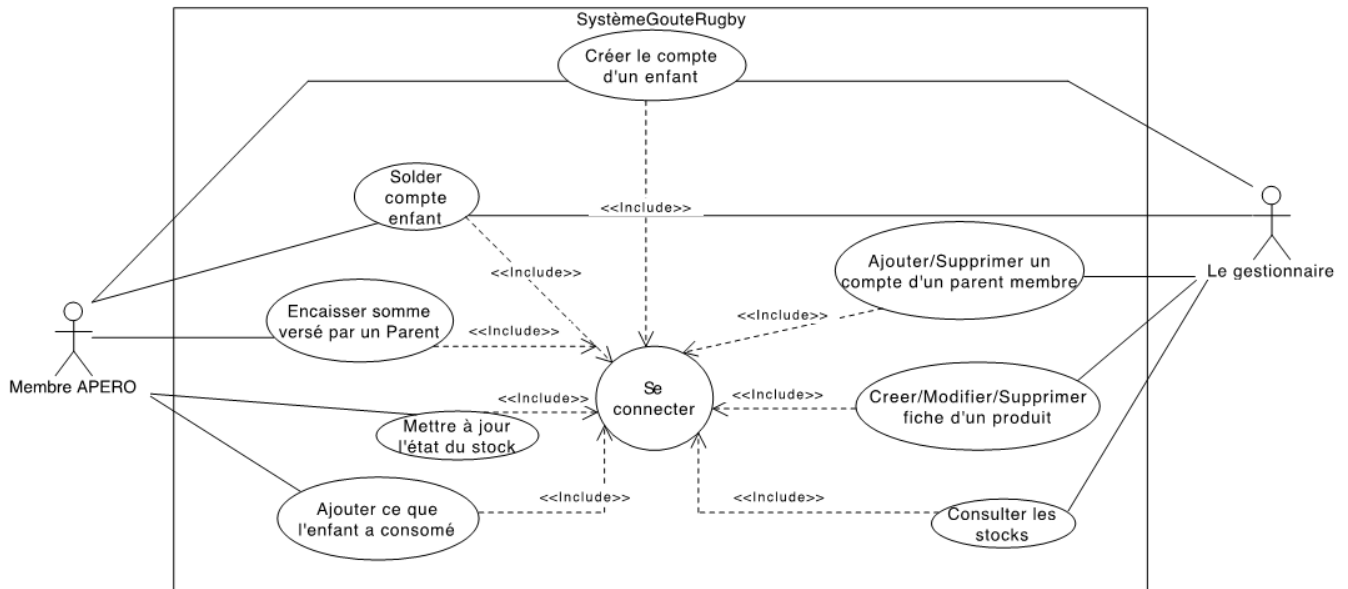
Ils ont aussi la possibilité de créditer le compte d'un enfant d'une certaine somme d'argent. Enfin, lorsqu'ils font les courses, ils indiquent le montant total de leur achat, et mettent à jour le stock de chaque produit, en indiquant les quantités achetées de chacun des produits.

- 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.



B – DIAGRAMME DE CAS D'UTILISATION



- **Se connecter** : Le gestionnaire ainsi que les parents qui sont membres de l'association doivent pour accéder à la fonctionnalité de l'application, se connecter (login + mot de passe validé à la création du compte).
- **Créer le compte d'un enfant** : les utilisateurs de l'application peuvent créer le compte d'un enfant participant au goûter proposé par l'association
- **Solder compte enfant** : afin de fermer et de supprimer le compte d'un enfant, l'argent qui reste sur le solde de l'enfant doit être versé au parent si le solde est positif ou les parents doivent verser le montant du « découvert » si le solde est négatif. Le compte d'un enfant ne peut être supprimé que si le solde est de 0.



- **Encaisser somme versée par un Parent** : Indique qu'un parent d'un enfant a versé une somme à l'association pour son enfant, cette somme doit être ajoutée au solde de l'enfant et mettre à jour le solde de la « caisse » de l'association.
- **Mettre à jour état du stock** : Lors de la réalisation des courses pour remplir le stock, les utilisateurs doivent mettre à jour les nouvelles quantités des produits achetés.
- **Ajouter ce que l'enfant a consommé** : Enregistrer ce que l'enfant a pris au goûter. Cela induit la mise à jour de l'état du stock ainsi que du solde de l'enfant.
- **Ajouter/Supprimer un compte d'un parent membre** : Le gestionnaire peut supprimer ou ajouter un nouveau compte pour un nouveau membre de l'association.
- **Créer/Modifier/Supprimer fiche d'un produit** : Le gestionnaire peut créer un nouveau produit, le supprimer ou alors modifier le prix de vente d'un produit.
- **Consulter les stocks** : Le gestionnaires peut voir les produits qui seront bientôt en pénuries et peut anticiper en relevant ce qu'il faut acheter.



II - ANALYSE ET CONCEPTION

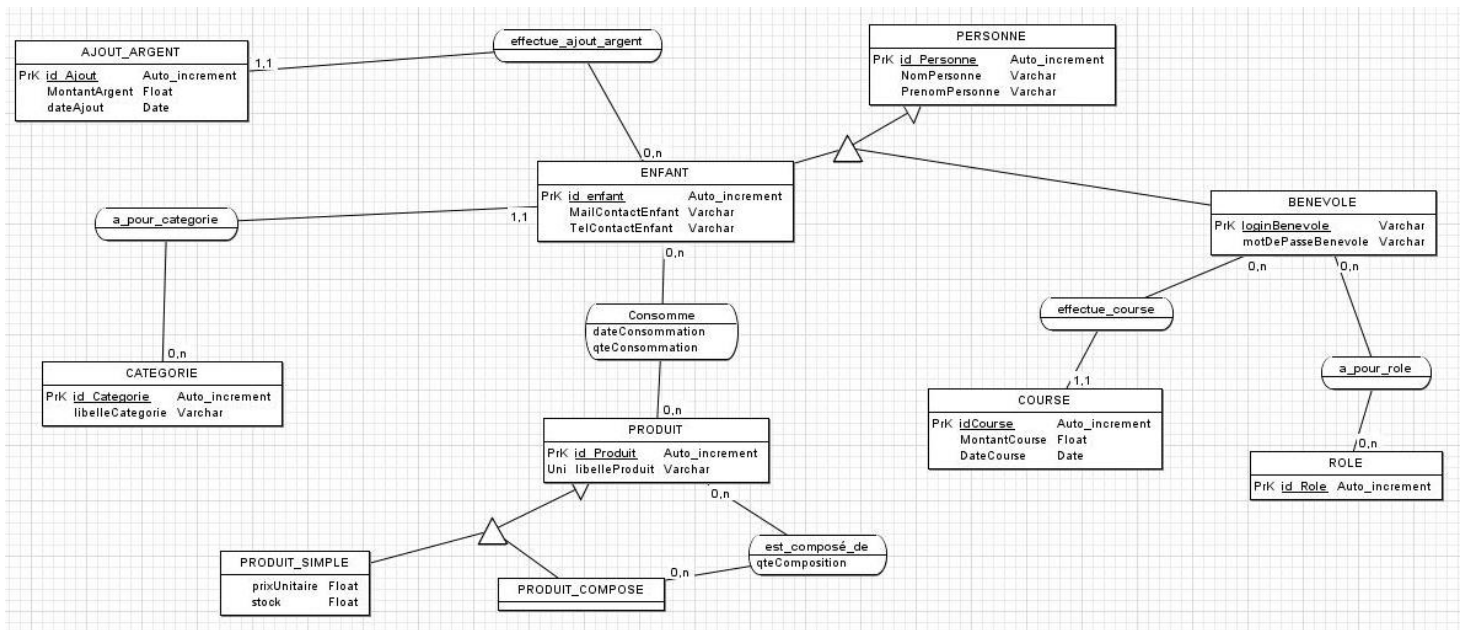
A. DICTIONNAIRE DES DONNEES

Table	NOM	TYPE	DEFINITION
ENFANT	Id_enfant	A	Identifiant de l'enfant
	MailContactEnfant	A	Adresse e-mail de l'un des parents de l'enfant
	TelContactEnfant	A	Numéro de téléphone de l'un des parents de l'enfant
CATEGORIE	Id_categorie	A	Identifiant de la catégorie
	libelleCateéorie	A	Libellé de la catégorie
AJOUT_ARGENT	Id_ajout	A	Identifiant de l'ajout d'argent
	MontantArgent	A	Montant de l'ajout
	DateAjout	A	Date de l'ajout
PRODUIT	Id_produit	A	Identifiant du produit
	LibelleProduit	A	Libellé du produit
PRODUIT_SIMPLE	PrixUnitaire	A	Prix Unitaire du produit
	Stock	A	Quantité en stock du produit
PERSONNE	Id_Personne	A	Identifiant de la personne
	NomPersonne	A	Nom de la personne
	PrenomPersonne	A	Prénom de la personne
COURSE	Id_course	A	Identifiant de course
	MontantCourse	A	Montant de la course
	DateCourse	A	Date de la course
ROLE	Id_role	A	Identifiant du rôle
	LoginBenevole	A	Login du bénévole
	motDePasseBenevole	A	Mot de passe du bénévole
CONSOMME	DateConsommation	A	Date de la consommation de produits
	QteConsommation	A	Quantité de la consommation de produits
PRODUIT_COMPOSE	QteComposition	A	Quantité de la composition d'un produit
	PrixProduitCompose	C	Le prix du produit selon la quantité de chaque produit unitaire qui le compose multiplié par le prix de chacun de ces composants $\sum(\text{Prix Unitaire} * \text{QuantitéComposition})$
	MontantCommande	C	Prix d'une commande = QuantitéConsommation * PrixUnitaire (si la commande concerne un produit unique) sinon Prix commande = QuantitéConsommation * PrixProduitCompose (si la commande concerne un produit composé)
	SoldeEnfant	C	Solde d'un enfant = $\sum \text{MontantArgent} - \sum \text{MontantCommande}$
	SoldeCaisse	C	Solde de la caisse de l'association = $\sum \text{SoldeEnfant} - \sum \text{Montant des courses}$



B. MODELE CONCEPTUEL DES DONNEES

1. LE MODELE



2. JUSTIFICATION DES CHOIX

Les catégories : Chaque enfant dépend d'une catégorie, qui lui est attribué en fonction de son âge. Chaque catégorie est identifiée par son âge de seuil. Par exemple la catégorie 'Moins de 10 ans' est identifié par le numéro 10, ou 'les plus de 15 ans' par le numéro 15.

Les courses : Les course sont effectués par les membres de l'association, ils enregistrent le prix global des courses ainsi que sa date d'achat. Les course sont composé de produit « unique ». Les membres devront également informer le montant de la course et pour chaque produit, mettre à jour la quantité en stock directement dans la table PRODUIT_UNIQUE.

Les produits : Il y a 2 types de produits, ce dit « unique » qui sont directement donnés aux enfants sans préparation préalable (ex : sachet de bonbon, biscuit, barre chocolat, canette etc.). Et enfin ceux dit « composé » qui sont créer à partir de plusieurs produits simples. Pour le « fabriquer », une certaine quantité des composants sont nécessaire à avoir. Par exemple le pain-chocolat c'est 1/8 de baquette et 1/12 de chocolat, ou alors verre de jus c'est 1 verre en plastique et de 1/8 de jus d'orange. Avec ce choix, on ne peut pas directement représenter un produit composé, composé à partir d'autre produit composé.

Les Commandes : Afin de mieux savoir ce qu'un enfant à consommer, chaque commande possède un unique produit, et un unique enfant. Une commande est identifiée par un attribut qui est généré automatiquement.



Les ajouts d'argent : Comme pour les commandes, chaque entrée d'argent est datée. Ainsi on sait quand, combien d'argent est rentré et pour quels enfants.

Les personnes : Même s'ils ont en commun un login pour se connecter et un mot de passe, on souhaite distinguer les « rôles ».

Benevole : une table bénévole qui contient le login et le mot de passe de ce bénévole qui peut effectuer des courses et il a un rôle.

Enfant : Une table Enfant regroupant tous les enfants au sein de l'école de rugby.

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.

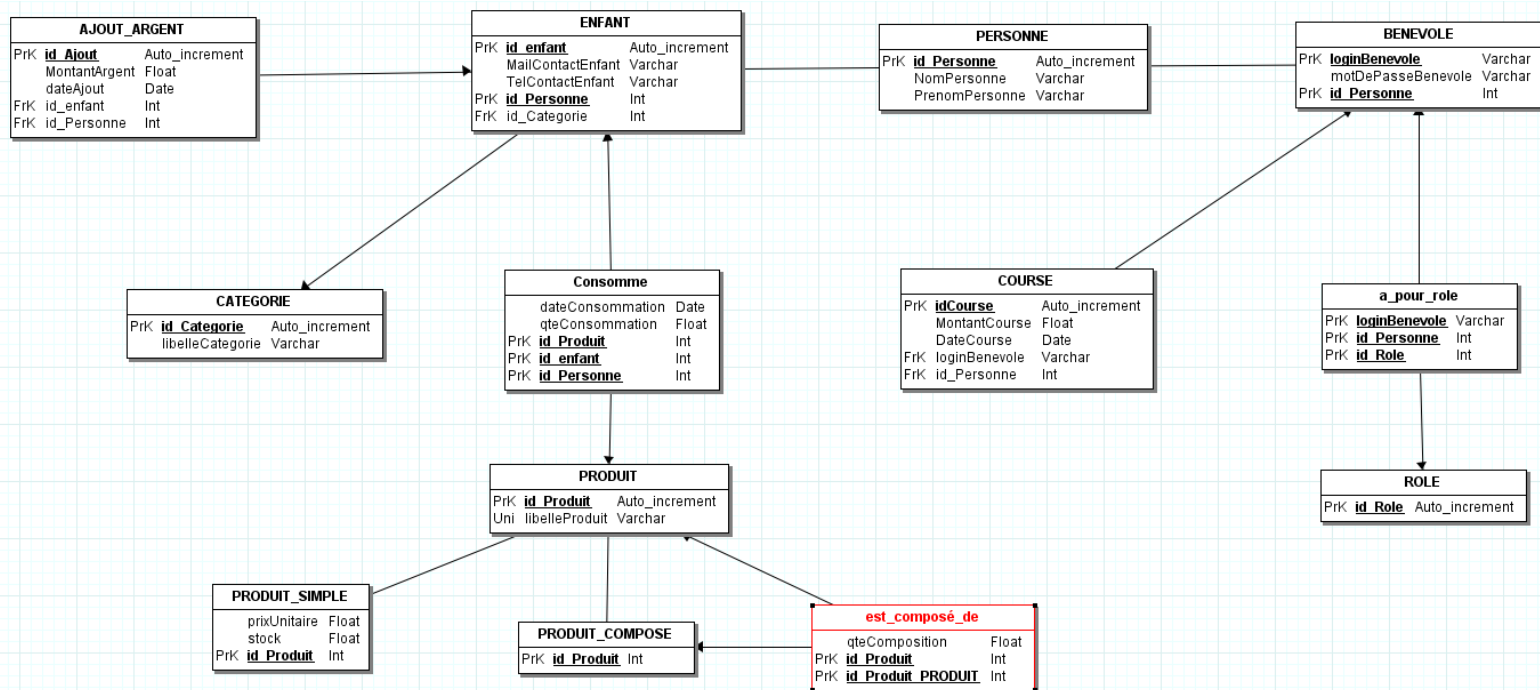
Produit : une table Produit regroupant les différents goûters dont dispose l'association APERO.

Rôle : une table Rôle regroupant les différents rôles au sein de l'application GERO : Président, Parent bénévole ou bien Parent simple



III. IMPLEMENTATION

A. MODELE RELATIONNEL DES DONNEES



Les fichiers « .sql » qui ont permis d'implémenter la base de données sont disponibles sur moodle dans l'espace de rendus du projet tutoré S3.

B. Code SQL de la création de la base de données

Cf. « lesTables.sql », disponible sur moodle.

C. Code SQL de l'insertion de données exemples

Cf. « insertions.sql », disponible sur moodle.

D. Code des requêtes permettant de réaliser chacune des fonctionnalités de l'application

Cf. « fonctionnalité.sql », disponible sur moodle.

E. Code des vues

Cf. « LESVUES.sql », disponible sur moodle.

F. Code des triggers

Cf. « trigger.sql », disponible sur moodle.



IV- DOCUMENTATION TECHNIQUE

A. INSTALLATION DE LA BASE DE DONNEES

Création de votre compte Oracle pour l'IUT

La première chose à faire avant de tenter une connexion, avec l'outil SQLdeveloper, est de créer votre compte Oracle sur la base de données de l'Iut d'Orsay.

Pour cela, vous avez à votre disposition un lien qui vous donnera accès à une interface vous permettant la création ou la modification de votre compte Oracle. Cette interface vous permettra également de récupérer un certain nombre d'information vous concernant. Notamment votre login court, votre adresse mail et votre groupe.

Cette interface est accessible à l'adresse :

<https://oracle.iut-orsay.fr/>

Pour vous connecter, il vous faut utiliser votre compte Adonis (prénom.nom) avec votre mot de passe.

Une fois connecté vous avez la possibilité ou pas de créer un compte Oracle. Pour cela, il vous suffit d'entrer un mot de passe. Dans le cas ou votre compte n'existe pas, il sera créé. Si votre compte existe, il sera modifié.

IMPORTANT : Le mot de passe doit obligatoirement commencer par une lettre, puis lettres ou/et chiffres ou caractères spéciaux (dièse, Underscore et le dollar). Le mot de passe ne doit pas dépasser 30 caractères.

Une fois créé votre compte oracle sera :

Votre login court

Et le mot de passe que vous aurez indiqué sur l'interface



Installer SQLdeveloper

1) Téléchargement

Pour le téléchargement de SQLdeveloper, aller à l'adresse suivante :

<http://www.oracle.com/technetwork/database/database-technologies/expressedition/downloads/index.html>

La création d'un compte vous sera demandé pour effectuer le téléchargement. Si vous n'en avez pas, vous devrez en créer un.

Remarque : Si vous avez récupéré une version de SQLdeveloper sans le jdk. Vous pouvez le récupérer à l'adresse suivante :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2) Installation

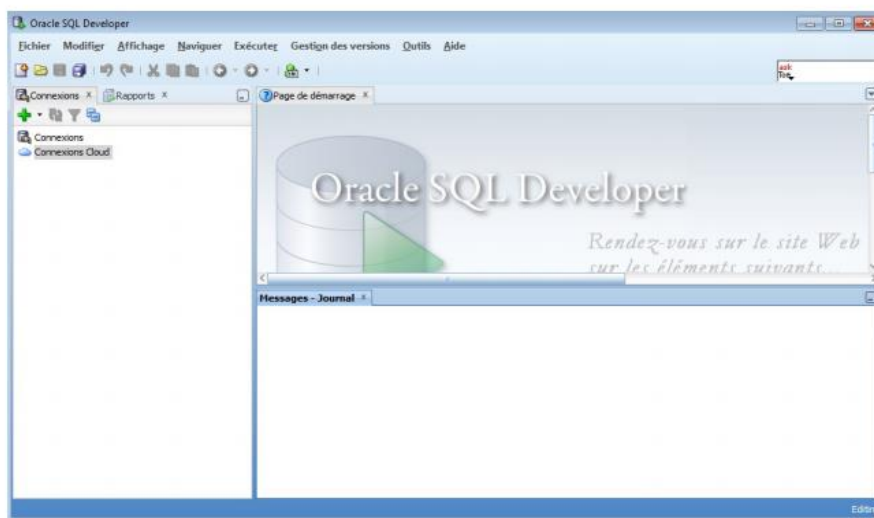
Décompressez et copier le répertoire sqldeveloper à l'endroit que vous souhaitez (pour ma part, [c:\programmes](#)). Je crée ensuite un raccourci sur le bureau.

Lancer SQLdeveloper

Pour exécuter SQLdeveloper, il vous suffit de double cliquer sur l'icône suivant :



La fenêtre suivante apparaîtra :

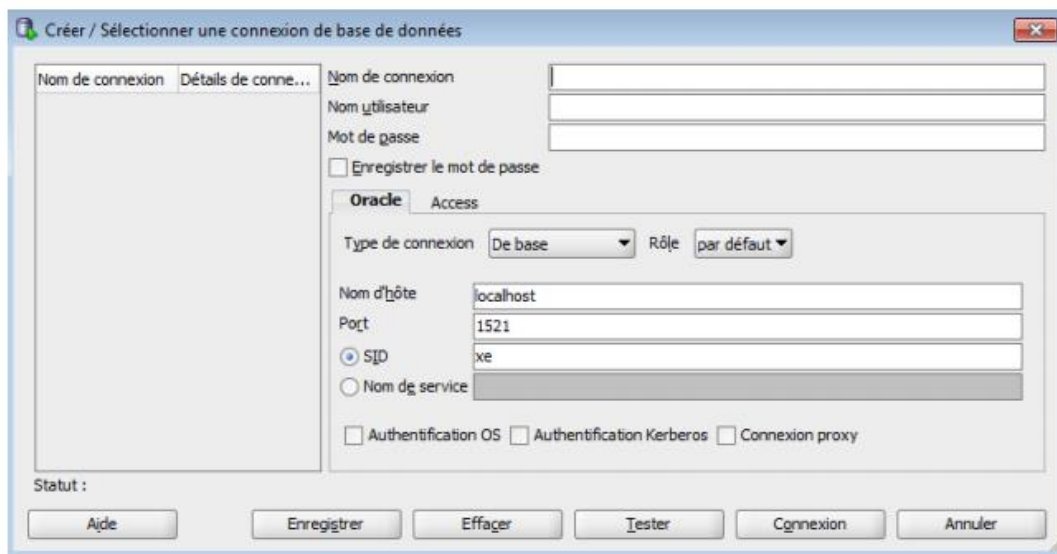


Ajouter une connexion

Pour créer votre connexion, il vous faut effectuer un clic droit sur connexion, puis sélectionner dans le menu qui s'affiche « Nouvelle Connexion ».



La fenêtre suivante apparaîtra :



Tester votre connexion

Maintenant que votre logiciel est configuré, vous allez pouvoir tester le bon fonctionnement de votre connexion.

Pour cela cliquez sur le bouton « Tester ».

Succès


Si votre connexion fonctionne, vous devez voir apparaître le statut « succès ». Votre connexion est opérationnelle, il ne vous reste plus qu'à lancer la connexion via le bouton « Connexion ».


Erreur de connexion

Dans le cas où votre connexion échoue :

- 1) Vérifiez que toutes les informations renseignées sont exactes et que votre compte Oracle a bien été créé (la création du compte se fait via l'interface S.C.O.A à l'adresse : <https://oracle.iut-orsay.fr/>).
- 2) Si malgré cela le problème persiste, veuillez en informer le CCRI (voir la section « joindre le ccri »).

Ouverture et exécution des fichiers .sql :

1^{er} Etape : Ouvrir le fichier lesTables.sql dans sqlDeveloper, puis cliquer sur le bouton « Run Script »  Vérifiez que toutes les tables ainsi que toutes les séquences sont créés, et apparaissent dans votre SGBD (sqlDeveloper).

2^{ème} Etape : Ouvrez le fichier Trigger.sql, puis sélectionnez un trigger et cliquez sur le bouton « Run statement ».  Une fenêtre s'ouvre, cliquez sur « Apply ». Vérifiez que le trigger s'est parfaitement compilé. Recommencez pour les autres triggers du fichier.

3^{ème} Etape : Ouvrez le fichier insertions.sql, puis cliquer sur le bouton « Run script ». Vérifiez que le script s'est parfaitement lancé, et que les données figures bien dans chaque table.

4^{ème} Etape : Ouvrez le fichier LESVUES.sql et fonctionnalités.sql, puis exécutez entièrement le script comme vue précédemment.

Votre base de données est maintenant créée et opérationnelle.

B. MAINTENANCE DE LA BASE DE DONNEES

Après avoir exécuté 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 quelque action sur la base de données veuillez 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 au S3, 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. Voici la liste des problèmes rencontrés et que nous avons résolus :

Lors de la création de la base de données (les tables) : Quelques identifiants de tables ont besoin d'AUTO_INCREMENT par exemple ID_ENFANT, ID_PERSONNE. Mais l'expression AUTO_INCREMENT ne fonctionne pas sur sqlDelovoper.

Solution : en cherchant l'instruction utile mais vraiment compliqué pour réaliser ce fonctionnement.

```
→ GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE  
999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20  
NOORDER NOCYCLE.
```

Lors des insertions des données : A cause d'AUTO_INCREMENT, on ne peut pas insérer des identifiants de quelques tables.

Solution : en utilisant l'expression INSERT INTO Table (colonne1, colonne2...) values (v1, v2...). Les identifiants vont se créer automatiquement.

Pour les vues : cette partie est la plus compliquée. Pour réaliser ce fonctionnement, il faut créer plusieurs vues soient utilisées par la vue demandée. Parce que pour chaque enfant, il peut consommer des produits simples, des produits composés et même les deux types de produits. Et le prix d'un produit composé doit aussi être bien calculé.

Solution : On a créé 4 vues dont 3 vues sont utilisées pour réaliser l’affichage de solde de chaque enfant.

VUECALCULM -> calculer et affiche pour chaque enfant, son montant d'ajout total.

VUEPRODUIT -> calculer et affiche pour chaque produit, son prix.

VUEPRIXTOTAL -> calculer et afficher pour chaque enfant, sa consommation totale.

VUESOLDE -> calculer et afficher pour chaque enfant, son solde. (Ce que l'on veut).

Il faut trouver des bonnes valeurs pour bien tester le fonctionnement de la base des données, ce qui n'est pas compliqué mais nous a pris beaucoup de temps.

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.

