# Rapport d'analyse de la SAE de BDD

Mon groupe constitué de **moi, Gabriel NAE et Nicolas MARSZALEK**, avons conçu dans la **partie 1** de la SAE de BDD, la base de données permettant de faire un **suivi pédagogique des étudiants durant plusieurs années universitaires**. Il s'agissait de réaliser le schéma conceptuel de données en Entité-Association, le schéma relationnel équivalent obtenu par transformation normalisé si nécessaire ainsi que le rapport d'analyse sur les explications de nos choix et de nos hypothèses.

Dans cette **partie 2**, il nous a été demandé de prendre **un des quatre projets** proposés sur la base de données et de programmer individuellement :

- Le script nommé **script.txt** qui contient la création et la suppression de la base données dont les tables, les procédures, les fonctions.

- Le script nommé **tests.txt** qui contient les tests avec le jeu de données.

La programmation du code a été effectué sur **MySQL** et le **projet 3** a été choisi.

Le **projet 3** a pour but de faire la gestion d'un ou des groupe(s) de formation ainsi que leur(s) réservation(s). Ce qu'on nous demande c'est de :

- Créer une procédure **MajGroupe qui ajoute, met à jour et supprime le(s) groupe(s) de formation.**

- Créer une procédure **ReservationsGroupe qui réserve des salles au créneau indiqué pour le(s) groupe(s) de formation.**

- Créer une fonction **EstLibre qui retourne vrai ou faux si le groupe d'une formation est libre au créneau indiqué.**

***VARCHAR*** : chaîne de caractères

***DECIMAL*** : nombre décimal

***INT*** : nombre entier

***DATETIME*** : date et temps

***TIME*** : temps

## ■ Procédure MajGroupe :

La procédure MajGroupe possède comme paramètres : **Gpe *en VARCHAR,* Forma *en VARCHAR et* Eff *en DECIMAL.***

Si soit **Gpe** est **vide** soit **Forma** est **vide** ou soit **Eff** est **vide**, on **affiche** un **message d'erreur** (« Tous les paramètres sont obligatoires ») sinon, on **continue**.

Ensuite, on **sélectionne** un **compteur qu'on enregistre** dans **@count** qui va compter le nombre de fois où il voit apparaître (**dans WHERE**) **Gpe** et **Forma** dans la table **Formation.**

Si **@count est égal à 0 et** que **Eff est positif**, on **insére les données** (**Gpe, Forma, Effectif**) dans la table **Formation.**

Si **@count est supérieur à 0 et** que **Eff est positif**, on **met à jour les données** (**Gpe, Forma, Effectif**) dans la table **Formation.**

Si **@count est supérieur à 0 et** que **Eff est négatif**, on **affiche un message d'erreur** (« L'effectif est négatif ») sinon (**@count égal à 0**), on **supprime** le groupe de formation (**Gpe, Forma, Effectif**) de la table **Formation** ainsi que tous ses réservations dans la table **Réservations**.

## ■ Procédure ReservationsGroupe :

La procédure ReservationsGroupe possède comme paramètres : **Gpe *en VARCHAR,* Forma *en VARCHAR.***

Au début, on **déclare 3 variables *INT*** explicites (**Groupe_p, Forma_p, Liste_p**) qui seront des **compteurs dans un affichage**.

Ensuite, si **Gpe** est **vide** (**omis**), on **affiche** la liste de réservation de tous les groupes d'une formation (on fait l'affichage avec la table **Réservations où dans *WHERE* on a Forma**) sinon, on **continue**.

On met dans l'**affichage** de la liste une **jointure** de la table **ELP** pour fusionner **CodeELP** de la table **Réservations** avec celle de la table **ELP**. La **jointure** s'explique par le fait qu'on utilise ***GROUP BY*** qui groupe (en fonction de la **clé primaire NoReservation**) tous les éléments de la table **Réservations** et non de la table **ELP**.

Les informations dans la liste sont présentés comme ceci : **Début, Fin (Début + Durée), CodeELP, NomELP, Nature, NoSalle, Gpe.**

Dans **Fin** puisque **Durée** est en **minutes**, on **additionne Début** avec **l'ensemble convertie en *DATETIME*** de la **concaténation** de **Durée divisé par 60** (***heure***) additionné au **caractère « : »** additionné à la **Durée multipliée par 60** (***minutes***) **convertie en *TIME***.

De plus, on **sélectionne** un **compteur qu'on enregistre** dans **Groupe_p** qui compte le nombre de fois où il voit apparaître (**dans *WHERE***) **Gpe** dans la table **Groupes**.

Ensuite, on **sélectionne** un **compteur qu'on enregistre** dans **Forma_p** qui compte le nombre de fois où il voit apparaître (**dans WHERE**) **Forma** dans la table **Groupes**.

Si **Groupe_p est à 0 ou** que **Forma_p est à 0**, on **affiche** un **message d'erreur** (« Groupe ou formation inexistant(e) ») sinon, on **continue** et on **sélectionne** un **compteur qu'on enregistre** dans **Liste_p** de la table **Réservations**.

Si **List_p est à 0**, on **affiche** un **message d'erreur** (« Pas de réservation pour ce groupe ou cette formation ») sinon, on **affiche** la liste de réservation d'un groupe d'une formation (**Gpe, Forma**) de la table **Réservations**.

## ■ Fonction EstLibre :

La fonction EstLibre possède comme paramètres : **Gpe *en VARCHAR,* Forma *en VARCHAR,* Début *en DATETIME,* Duree *en DECIMAL.*** Elle **retourne** une valeur **booléien. Durée** a été ***remplacé* par** un ***INT*** cela ne fonctionne pas en ***DECIMAL***.

Au début, on **déclare 2 variables *INT*** explicites (**Groupe_p, Liste_p**) où **Groupe_p** et **Liste_p** seront des **compteurs dans un affichage**.

On **sélectionne** un **compteur qu'on enregistre** dans **Groupe_p** qui compte le nombre de fois où il voit apparaître (**dans *WHERE***) **Gpe** dans la table **Groupes**.

Si **Groupe_p est égal à 0**, on **arrête** la fonction **et** on **affiche** un **message d'erreur** (« Groupe inexistant ») sinon, on **continue**.

Ensuite, on **sélectionne** un **compteur qu'on enregistre** dans **Liste_p** qui compte le nombre de fois où il voit apparaître (**dans WHERE**) **Gpe, Forma, Début et Durée** dans la table **Réservations**.

Si **Liste_p est égal à 0**, la fonction **retourne vrai** sinon, elle **retourne faux**.