

Rapport d'analyse SAE-3.04 :

Exploration d'une base de données

Le Projet n°2 vise à réaliser la gestion des enseignements au sein du base de données.

Pour ce faire, deux procédures et une fonction nous sont demandés d'être réalisés :

- Premièrement, la procédure 'MajELP' vérifie les ELPs dans la table 'ELP'.
- Deuxièmement, la procédure 'Reservations' vérifie quels ELPs ont été réservés.
- Troisièmement, la fonction 'ReservationComplete' retourne une valeur booléenne selon la condition qu'un ELP à bien été réservé ou non.

▪ **Procédure 'MajELP' :**

La variable : 'elp_existe' est déclarée au début de la fonction en tant qu'integer, qui représente un nombre entier (INT en MySQL).

Une itération est initialisée et elle est stocké au sein de la variable 'elp_existe' qui a été créée juste au préalable.

Si le compteur trouve la valeur = 0 (ce qui se traduit par une non-existence de l'ELP au sein de la table 'ELP'), il insère l'ELP dans la table 'ELP' (avec toutes ses valeurs passées en paramètres de la procédure).

Sinon la procédure vérifie si les heures de cours ou les travaux-dirigés ou les travaux-pratiques sont différents de l'ELP existant, elle utilise comme référence le CodeELP (Code) qui est écrit en paramètre lors de l'appel de la procédure, et si tel est le cas l'ELP est mis à jour avec les valeurs passées en paramètres.

Finalement si aucune de ces conditions n'est satisfaite, l'ELP est supprimé de la table 'ELP'.

▪ **Procédure 'Reservations' :**

Tout d'abord, une variable 'v_existe' est déclarée, comme dans la procédure 'MajELP' cette variable nous servira à sauvegarder des données (l'itération dans ce cas-ci) au sein de la variable.

Dans la table 'ELP' où la paramètre 'Code' correspond à 'CodeELP', la procédure compte à partir de '1' et cherche le 'CodeELP' qui est égal au paramètre 'Code'.

Si elle trouve une valeur qui correspond au paramètre, elle sélectionne la propriété 'Debut' de la table 'Reservation' (elle effectue de surcroît une opération qui calcule un intervalle entre 'Debut' et 'Duree' entre les deux valeurs) en affichant les données trouvées, grâce à une jointure entre la table 'Reservation' et 'ELP', puis elle ordonne les données selon la donnée 'Debut' du résultat.

Puis si elle ne trouve rien lors de l'itération précédente, elle cherche et essaye de trouver des lignes correspondantes au sein de la variable 'v_existe'.

Si elle ne trouve aucune réservation, elle retourne le message = 'Pas de réservation pour cet ELP.'

De plus, si elle ne trouve aucun ELP correspondant, elle retourne le message = 'L'ELP n'existe pas.'

▪ **Fonction 'ReservationComplete' :**

'ReservationComplete' est fonction booléenne et déterministe.

Deux variables sont déclarées au préalable, 'heures_totales' et 'heures_reservees' en tant qu'integer (nombre entier).

Ensuite la fonction fait la somme des heures de cours, travaux-dirigés et travaux-pratiques de l'ELP appelé par la fonction, puis elle insère ces données dans la variable 'heures-totales', les données sont récupérées dans la table 'ELP' ou le 'CodeELP' est égal au paramètre 'Code' de la fonction.

Si la fonction trouve un résultat concordant, elle affiche chaque résultat ligne par ligne (avec 'LIMIT 1').

Si la fonction ne trouve rien en retour, elle stoppe la fonction et affiche un message d'erreur = 'L'ELP n'existe pas !' et retourne une valeur 'NULL'.

Sinon si aucune de ces conditions n'ont pas été remplies, la fonction fait la somme des heures de cour, travaux-dirigés et travaux-pratiques réservées et insère ce résultat au sein de la variable 'heures_reservees' et crée un jointure entre la table 'Reservation' et 'ELP' et compare ces données entre les deux tables grâce au paramètre 'Code' où il est égal à 'CodeELP' (comme la condition précédente elle affiche chaque résultat ligne par ligne avec 'LIMIT 1').

Ensuite elle vérifie si la somme des 'heures_totales' est plus grande ou égale aux 'heures_reservees', si cette condition est validée, elle retourne 'TRUE', sinon elle retourne 'FALSE'.