



FACULTÉ DES SCIENCES APPLIQUÉES

INFO0009-A-B BASES DE DONNÉES

---

# Implémentation d'un site de gestion de base de données en PHP et MySQL

---

*Professeur :*  
Christophe DEBRUYNE

*Groupe 25 :*  
Guillaume DELPORTE  
Maxime FIRRINCIELI  
Romain LAMBERMONT  
Arthur LOUIS

6 mai 2022

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Architecture du site web</b>	<b>1</b>
<b>3</b>	<b>Initialisation de la base de données</b>	<b>2</b>
<b>4</b>	<b>Requêtes</b>	<b>2</b>
<b>5</b>	<b>Version définitive du site</b>	<b>5</b>

# 1 Introduction

Pour ce projet il nous a été demandé de créer un site web de gestion de base de données utilisant PHP et MySQL grâce à l'API PDO. Le modèle relationnel qui nous été demandé d'implémenter est le suivant :

- DEPARTEMENT(NOM,BUDGET)
- EMPLOYE(NO,NOM,NOM\_DEPARTEMENT,NOM\_FONCTION)
  - NOM\_DEPARTEMENT → NOM de DEPARTEMENT, NOM\_FONCTION → NOM de FONCTION
- EVALUATION(PROJET,EXPERT,COMMENTAIRES,AVIS)
  - PROJET → NOM de PROJET, EXPERT → NO de EMPLOYE
- FONCTION(NOM,TAUX\_HORAIRE)
- MOTS\_CLES(RAPPORT,MOT\_CLE)
  - RAPPORT → TITRE de RAPPORT
- PROJET(NOM,DEPARTEMENT,DATE\_DEBUT,CHEF,BUDGET,COUT,DATE\_FIN)
  - DEPARTEMENT → NOM de DEPARTEMENT, CHEF → NO de EMPLOYE
- RAPPORT(EMPLOYE,PROJET,TITRE)
  - EMPLOYE/PROJET → EMPLOYE/PROJET de TACHE
- TACHE(EMPLOYE,PROJET,NOMBRE\_HEURES)
  - EMPLOYE → NO de EMPLOYE, PROJET → NOM de PROJET

Dans ce rapport nous allons décrire notre site web et les requêtes SQL que nous avons du réaliser pour accéder aux bons éléments de notre base de données.

# 2 Architecture du site web

Pour accéder à notre site web, il faut d'abord se connecter via la page de login vers laquelle vous êtes redirigé directement si l'utilisateur n'est pas connecté et ce grâce aux variables \$\_SESSION de PHP.

Une fois connecté, vous accédez à la page d'accueil qui décrit simplement les différents onglets du site web et leurs fonctions. Ces différents onglets sont accessibles via la barre de navigation supérieure.

Voici une courte description des différents onglets :

- Tri : Cet onglet permet de naviguer dans les différentes tables de la base de données et d'effectuer des tris dans celles-ci afin d'effectuer des recherches précises.
- Tâches/Projet : Cet onglet permet d'afficher les tâches relatives aux différents projets.
- Ajout : Cet onglet permet d'ajouter des éléments dans les tables Employe,Projet et Fonction. Vous pourrez également modifier les attributs des employés et les budgets des projets non terminés.
- Tâches : Dans cet onglet vous pouvez afficher les détails de chaque projet et les tâches relatives à ceux-ci. Si le projet est terminé vous pouvez modifier l'avis de l'évaluation de l'expert si il y en a une. Si le projet n'est pas terminé, vous pouvez décider de créer des nouvelles tâches à des employés disposant de fonction et d'un département, d'ajouter des heures à des tâches non terminées (sans rapport) et de mettre fin au projet (calcul du coût, création de la date de fin, évaluation éventuelle d'un expert) si le budget de celui-ci a été enregistré (modification possible dans l'onglet ajout).
- Employé/Projet : Cet onglet permet d'afficher les employés qui ont participé à tous les projets, que ce soit en tant que chef, employé ou expert. Vous pouvez également passer au mode complet qui affiche les responsabilités de tous les employés dans tous les projets.
- Projets : Tableau de bord des différents projets affichant leur statut (cfr. partie 1 du projet) et leurs différents attributs.
- Employés : Tableau des bord des employés permettant de montrer le nombre de projets réalisés par chaque employé et le nombre d'heures des tâches qu'ils ont effectué. Un tableau montre également des statistiques relatives aux données citées plus haut.

### 3 Initialisation de la base de données

Afin d'initialiser correctement la base de données il suffit de se connecter en SSH aux machines de l'université et de lancer MySQL depuis le dossier WWW de notre compte sur ces machines :

```
$ ssh s191230@ms801.montefiore.ulg.ac.be
$ cd WWW
$ mysql -h ms8db -u group25 -p
Enter password: 3uCTA8L2ID
mysql> source ./script.sql
```

Ce script se charge de charger toutes les tables de notre base de données.

### 4 Requêtes

Nous allons maintenant expliciter les requêtes nécessaires au fonctionnement des différents onglets du site web qui permettent de parcourir et d'effectuer des actions sur la base de données :

Pour l'onglet Tri une simple requête SELECT est nécessaire pour afficher les éléments de la table. Si on veut par la suite restreindre des attributs de la table à certaines valeurs il suffisait de rajouter un WHERE.

```
> SELECT * FROM Employe
// Montre toutes les données de tous les employés
> SELECT * FROM Projet WHERE BUDGET=2000
// Montre toutes les données des projets où le budget vaut 2000
> SELECT * FROM Fonction WHERE NOM LIKE "%cheur%"
// Montre toutes les fonctions dont le nom contient "cheur"
```

Pour l'onglet Tâche/Projet il existe seulement deux types de requêtes, une générale affichant toutes les tâches et une autre spécifiant les tâches d'un projet spécifique à afficher. Dans les deux cas on se servait d'INNER JOIN afin d'afficher également la fonction de l'employé réalisant la tâche et le coût de cette tâche.

```
> SELECT T.PROJET, T.EMPLOYE, E.NOM, T.NB_HEURES,
--> E.NOM_FONCTION, F.TAUX_HORAIRE, F.TAUX_HORAIRE*T.NB_HEURES
--> FROM Employe E
--> INNER JOIN Tache T ON T.EMPLOYE = E.NO
--> INNER JOIN Fonction F ON E.NOM_FONCTION = F.NOM
// Version générale

> SELECT T.PROJET, T.EMPLOYE, E.NOM, T.NB_HEURES,
--> E.NOM_FONCTION, F.TAUX_HORAIRE, F.TAUX_HORAIRE*T.NB_HEURES
--> FROM Employe E
--> INNER JOIN Tache T ON T.EMPLOYE = E.NO
--> INNER JOIN Fonction F ON E.NOM_FONCTION = F.NOM
--> WHERE T.PROJET = "STOCH"
// Version précise sur le projet "STOCH"
```

Pour l'onglet Ajout on se sert simplement des commandes INSERT INTO et UPDATE pour ajouter ou mettre à jour des éléments dans les tables de la base de données. Les réglages de la base de données nous permettent de ne pas devoir nous soucier si certains champs peuvent être nuls ou non.

```
> INSERT INTO Fonction (NOM,TAUX_HORAIRE) VALUES ("Assistant developpeur","40")
// Ajout d'un élément dans la table Fonction

> UPDATE Employe SET NOM_DEPARTEMENT="INF", NOM_FONCTION="Chercheur" WHERE NO=8845
// Met à jour les attributs de l'employé possédant le numéro d'employé 8845
```

Pour l'onglet Tâches on affiche simplement les détails du projet (en utilisant une fonction permettant de mettre un code couleur au coût) et on affiche les tâches relatives au projet. Pour ce qui est de la soumission d'heures, la création de tâches, l'écriture de rapport et la mise à terme du projet on doit d'abord vérifier des valeurs dans notre base de données :

- Modification d'une évaluation (projet terminé) : il faut d'abord vérifier que le projet choisit est terminé et qu'il dispose d'une évaluation.
- Soumission d'heures : il faut d'abord vérifier qu'il existe des tâches sans rapport (non terminées) pour le projet choisit.
- Création de tâches : il faut d'abord vérifier qu'il existe des employés ayant une fonction et un département qui ne travaillent pas sur le projet actuel.
- L'écriture de rapport : il faut d'abord vérifier qu'il existe des tâches sans rapport (non terminées) où les heures de travail sont déterminées pour le projet choisit.
- Fin du projet et création d'évaluation : il faut d'abord vérifier que le projet ne dispose pas de date de fin.

```
> SELECT * FROM Projet P WHERE P.NOM = "STOCH"
// Affichage des détails du projet

> SELECT T.PROJET, T.EMPLOYE, E.NOM, T.NB_HEURES,
--> E.NOM_FONCTION, F.TAUX_HORAIRE, F.TAUX_HORAIRE*T.NB_HEURES
--> FROM Employe E
--> INNER JOIN Tache T ON T.EMPLOYE = E.NO
--> INNER JOIN Fonction F ON E.NOM_FONCTION = F.NOM
--> WHERE T.PROJET = "STOCH"
// Affichage des détails des tâches du "STOCH"

// Modification de l'évaluation d'un expert :
> SELECT EXISTS(SELECT * FROM Evaluation E WHERE E.PROJET="DIDACT")
// Recherche une évaluation pour le projet "DIDACT"

> UPDATE Evaluation SET AVIS="ECHEC",COMMENTAIRES="NEW_COMMENT" WHERE PROJET="DIDACT"
// Met à jour l'évaluation du projet "DIDACT"

// Soumission des heures :
> SELECT T.EMPLOYE
--> FROM Tache T
--> WHERE T.EMPLOYE NOT IN
--> (SELECT A.EMPLOYE
--> FROM Tache A INNER JOIN Rapport R ON (A.EMPLOYE = R.EMPLOYE)
--> WHERE A.PROJET="STOCH" AND R.PROJET="STOCH")
--> AND T.PROJET="STOCH"
// Recherche des employés de tâches non terminées (sans rapport) du projet "STOCH"

> UPDATE Tache SET NB_HEURES=10 WHERE EMPLOYE=8829 AND PROJET="STOCH"
// Ajout de 10 heures à la tâche de l'employé 8829 du projet "STOCH" (heures précédentes = 0)

// Création de tâches :
> SELECT NO
--> FROM Employe E
--> WHERE E.NO NOT IN
--> (SELECT EMPLOYE FROM Tache T WHERE T.PROJET="STOCH")
--> AND (E.NOM_DEPARTEMENT IS NOT NULL OR E.NOM_FONCTION IS NOT NULL)
// Employés sans tâche dans le projet "STOCH" ayant une fonction et un département
```

```

> INSERT INTO Tache (EMPLOYE, PROJET, NB_HEURES) VALUES (7777,"STOCH",10)
    // Création d'une tâche de 10 heures pour l'employé 7777 dans le projet "STOCH"

// Écriture d'un rapport :
> SELECT T.EMPLOYE
--> FROM Tache T
--> WHERE T.EMPLOYE NOT IN
-->   (SELECT A.EMPLOYE FROM Tache A
-->     INNER JOIN Rapport R ON (A.EMPLOYE = R.EMPLOYE)
-->     WHERE A.PROJET="STOCH" AND R.PROJET="STOCH")
--> AND T.PROJET="STOCH" AND T.NB_HEURES IS NOT NULL
    // Recherche des employés des tâches du projet "STOCH" sans rapport et avec heures fixées

> INSERT INTO Rapport (EMPLOYE,PROJET,TITRE) VALUES (8829,"STOCH","TITRE_RAPPORT")
    // Ajout d'un rapport pour la tâche de l'employé 8829 du projet "STOCH"

> INSERT INTO Mots_Cles (RAPPORT,MOT_CLE) VALUES ("TITRE_RAPPORT","KEYWORD_1")
> INSERT INTO Mots_Cles (RAPPORT,MOT_CLE) VALUES ("TITRE_RAPPORT","KEYWORD_2")
    // Ajout des mots clés du rapport créé ci-dessus (2 mots clés)

// Fin de projet et évaluation éventuelle :
> SELECT ISNULL(BUDGET) FROM (SELECT BUDGET FROM Projet WHERE NOM="STOCH") temp
    // Recherche si le budget du projet "STOCH" est NULL

> SELECT SUM(A) FROM
--> (SELECT F.TAUX_HORAIRE*T.NB_HEURES as A
-->   FROM Employe E
-->     INNER JOIN Tache T ON T.EMPLOYE = E.NO
-->     INNER JOIN Fonction F ON E.NOM_FONCTION = F.NOM
-->   WHERE T.PROJET = "STOCH") temp
    // Calcul du coût total des tâches du projet "STOCH"

> SELECT DATE(NOW())
    // Récupération de la date de fin de projet

> UPDATE Projet SET COUT=780, DATE_FIN="2022-05-06" WHERE NOM="STOCH"
    // Mise à jour du coût et de la date de fin du projet "STOCH"

> INSERT INTO Evaluation (PROJET,EXPERT,COMMENTAIRES,AVIS)
--> VALUES ("STOCH","8845","Retravailler le code","SUCCES")
    // Création d'une évaluation de l'expert 8845 pour le projet "STOCH"
    // ayant l'avis "SUCCES" et les commentaires "Retravailler le code"

```

Pour l'onglet Employé/Projet on effectue une simple requête SQL permettant de rechercher les employés ayant des rôles dans tous les projets.

```

> SELECT *
--> FROM (SELECT E.NO, E.NOM, E.NOM_FONCTION, CONCAT(
-->   IF(E.NO IN (SELECT P.CHEF FROM Projet P WHERE P.NOM = "MED-STAT"), "CHEF ", ""),
-->   IF(E.NO IN (SELECT E.EXPERT FROM Evaluation E WHERE E.PROJET = "MED-STAT"), "EXPERT ", ""),
-->   IF(E.NO IN (SELECT R.EMPLOYE FROM Rapport R WHERE R.PROJET = "MED-STAT"), "EMPLOYE_RAPPORT ", ""),
-->   IF(E.NO IN (SELECT T.EMPLOYE FROM Tache T WHERE T.PROJET = "MED-STAT"), "EMPLOYE_TACHE ", ""))
--> as "MED-STAT", ...)
--> temp WHERE "MED-STAT" != " " AND ...
    // Les ... indiquent une répétition de la requête pour les autres projets

```

Pour l'onglet Projets on effectue une simple requête SQL permettant d'afficher les détails d'un projet, son statut et son coût actuel si le projet n'est pas terminé.

```
> SELECT P.NOM,P.DEPARTEMENT,P.CHEF,
--> IF(P.DATE_FIN IS NOT NULL,"TERMINÉ",
--> IF(P.BUDGET IS NULL,"EN ATTENTE","EN COURS DE ROUTE")) as STATUT,
--> P.BUDGET,C.COUT,H.NB_HEURES,P.DATE_DEBUT,P.DATE_FIN
--> FROM Projet P,
--> (SELECT P.NOM as NOM, SUM(T.NB_HEURES) as NB_HEURES
--> FROM Tache T
--> INNER JOIN Projet P ON P.NOM = T.PROJET GROUP BY P.NOM) H,
--> (SELECT temp.P_C as PC,SUM(temp.COUT_C) as COUT
--> FROM
--> (SELECT Tache.PROJET as P_C, Tache.NB_HEURES*Fonction.TAUX_HORAIRE as COUT_C
--> FROM Employe
--> INNER JOIN Tache ON Tache.EMPLOYE = Employe.NO
--> INNER JOIN Fonction ON Fonction.NOM = Employe.NOM_FONCTION) temp
--> GROUP BY temp.P_C) C
--> WHERE H.NOM = P.NOM AND C.PC = P.NOM ORDER BY DATE_DEBUT ASC, NOM ASC
```

Pour l'onglet Employés il suffit pour le tableau de bord des employés de réaliser une requête en fonction de la colonne et de la direction de tri. Pour ce qui est du data il faut réaliser une requête plus longue utilisant des fonctions comme SUM, AVERAGE, ... et des GROUP BY.

```
> SELECT temp.NO as NO, temp.NOM as NOM,
--> COALESCE(temp.NB_PROJETS,0) as NB_PROJETS,
--> COALESCE(temp.NB_HEURES,0) as NB_HEURES
--> FROM (SELECT E.NO AS NO, E.NOM AS NOM,
--> COUNT(T.PROJET) AS NB_PROJETS, SUM(T.NB_HEURES) AS NB_HEURES
--> FROM Employe E LEFT JOIN Tache T ON E.NO = T.EMPLOYE
--> GROUP BY E.NO ORDER BY NB_PROJETS DESC) temp
// Affichage des détails pour chaque employé pour les projets et heures prestées

> SELECT "NB_PROJETS" as "DATA", SUM(temp.NB_PROJETS) as "SOMME", ...
--> FROM TABLE_temp_CI_DESSUS
--> UNION
--> SELECT "NB_HEURES" as "DATA", SUM(temp.NB_HEURES) as "SOMME", ...
--> FROM TABLE_temp_CI_DESSUS
// Les ... indiquent qu'il faut réaliser cela pour les
// différentes fonctions AVERAGE, MINIMUM, ...
```

## 5 Version définitive du site

La version finale du site web peut être accédée via l'URL suivant : <http://ms8db.montefiore.ulg.ac.be/s191230>  
 Vous pourrez accéder au site web grâce aux identifiants suivants :

- Identifiant : group25
- Mot de passe : 3uCTA8L2ID