

Institut Universitaire des Sciences (IUS)
FACULTÉ DES SCIENCES ET DES TECHNOLOGIES (FST)

PRESENTATION DU PROJET

Cours : Systèmes d'Exploitation Linux

Gestion d'une Base de Données des Employés pour une Entreprise de Télécom

Étudiant : Wendy COLAS
Niveau : L3

2024

Introduction

Ce script Bash est conçu pour interagir avec une base de données MySQL afin de gérer les informations des employés. Le script permet de :

- Créer une base de données et une table,
- Ajouter, modifier et supprimer des employés,
- Afficher la liste des employés.

L'objectif est de comprendre comment automatiser la gestion des employés en utilisant MySQL via un script shell, et comment créer des interfaces utilisateur simples avec un menu interactif.

Pour réaliser ce travail, je me suis d'abord penché sur le document envoyé par le professeur Saint Amour, puis j'ai fait quelques recherches supplémentaires me permettant d'apporter une touche personnelle au travail. J'ai utilisé Visual Studio Code comme éditeur de texte.

1. Paramètres de Connexion MySQL

Dans cette partie, j'ai inséré les informations que vont lier les actions du script à mon compte utilisateur de MySQL.

DB_NAME : C'est le nom de la base de données où sont stockées les informations des employés. Si la base de données n'existe pas, le script la crée automatiquement.

TABLE_NAME : C'est le nom de la table qui contient les informations sur les employés (nom, prénom, poste, etc.).

DB_USER : Utilisateur MySQL utilisé pour se connecter à la base de données. Ici, j'ai travaillé avec l'utilisateur par défaut qui est root, mais si le vôtre est différent vous allez devoir le changer dans le script en ajoutant votre nom d'utilisateur.

DB_PASS : Le mot de passe associé à mon compte utilisateur il est vide dans le script mais vous devrez l'insérer à chaque fois que vous voulez valider une opération.

2. Le Menu Principal

Le menu principal est une fonction menu() qui présente plusieurs options à l'utilisateur.

- Chaque option correspond à une fonctionnalité spécifique du système de gestion des employés.
- L'utilisateur choisit l'option en entrant un numéro, ce qui déclenche l'exécution de la fonction correspondante.

3. Créer la Base de Données et la Table

Cette fonction permet de créer une base de données telecom_employées si elle n'existe pas déjà et

d'y créer la table « employées ».

CREATE DATABASE IF NOT EXISTS : Crée la base de données si elle n'existe pas.

CREATE TABLE IF NOT EXISTS : Crée la table avec les colonnes spécifiées si elle n'existe pas encore.

id : Identifiant unique pour chaque employé (clé primaire).

prénom et nom : Champs pour stocker les informations personnelles de l'employé.

poste : Le poste occupé par l'employé dans l'entreprise.

département : Le département dans lequel l'employé travaille.

salaire : Le salaire de l'employé (avec une précision de 2 chiffres après la virgule).

date_embauche : La date d'embauche de l'employé.

4. Ajouter un Nouvel Employé

Cette fonction permet à l'utilisateur d'ajouter un nouvel employé à la base de données.

- Elle invite l'utilisateur à entrer les informations de l'employé (prénom, nom, poste, etc.).
- Ensuite, une requête SQL est exécutée pour insérer ces informations dans la table « employées ».

INSERT INTO permet de spécifier les valeurs à insérer dans chaque colonne de la table.

5. Mettre à Jour un Employé

Cette fonction permet de mettre à jour les informations d'un employé existant. L'utilisateur doit fournir l'ID de l'employé et les nouvelles valeurs pour les informations à mettre à jour.

UPDATE : Cette commande permet de modifier les valeurs dans la table en fonction de l'ID de l'employé.

Si un champ est laissé vide, il ne sera pas mis à jour (vérification avec `if [! -z "$prenom"]`).

6. Supprimer un Employé

Cette fonction permet de supprimer un employé de la base de données en utilisant son ID.

DELETE FROM : Cette commande supprime un enregistrement de la table basé sur l'ID fourni.

7. Afficher la Liste des Employés

Cette fonction permet de visualiser tous les employés présents dans la base de données.

SELECT * FROM : Cette requête récupère toutes les lignes et colonnes de la table « employées ».

8. Boucle Principale

Cette boucle permet de garder l'application en fonctionnement jusqu'à ce que l'utilisateur choisisse de quitter (option 6).

case \$choix in : Le choix de l'utilisateur est comparé à différentes options. En fonction de l'option choisie, une fonction correspondante est exécutée.

Conclusion

Ce script Bash offre une interface simple pour gérer une base de données MySQL d'employés. Il couvre les opérations de création, ajout, modification, suppression et affichage des employés. Cependant mon travail a ses propres limites parmi lesquelles on peut citer : Le script ne gère pas les erreurs qui peuvent survenir lors de l'exécution des commandes MySQL ou des entrées de l'utilisateur. De même, Le script n'offre pas beaucoup de fonctionnalités supplémentaires pour l'utilisateur.