Rapport Travaux Pratiques

**TP 2: Gestion de Congé**

Réalise par: FEKARI Hafsa

Encadre par: Dr.laila

Année Universitaire : 2024-2025

# I. Structure du Dossier

Le dossier est organis´e en cinq sections principales : Mod`ele, DAO, Controˆleur, Vue et Main.

## 1. Mod`ele

Cette section contient la classe HolidayModel, qui d´efinit la logique m´etier li´ee aux cong´es. Elle fait appel au DAO pour effectuer des op´erations sur les donn´ees des cong´es. Les principales fonctionnalit´es incluent :

**Constructeurs** : Permet de cr´eer des objets HolidayModel avec un DAO sp´ecifique.

**M´ethodes** :

* getEmployeeId(String fullName) : R´ecup`ere l’ID d’un employ´e a` partir de son nom complet.
* getHolidayById(int holidayId) : Renvoie les d´etails d’un cong´e a` partir de son ID.
* getSoldeDisponible(int id) : Renvoie le solde de cong´es disponible pour un employ´e.
* addHoliday(Holiday holiday) : Ajoute un cong´e `a la base de donn´ees.
* updateHoliday(Holiday holiday) : Met a` jour les informations d’un cong´e existant.
* deleteHoliday(int holidayId) : Supprime un cong´e en fonction de son ID.
* updateSoldeDisponible(int empId, double newBalance) : Met a` jour le solde de cong´es d’un employ´e apr`es avoir pris un cong´e.
* getAllHolidays() : R´ecup`ere tous les cong´es existants.

## 2. DAO (Data Access Object)

Cette section g`ere l’interaction avec la base de donn´ees, permettant d’effectuer des op´erations CRUD sur les cong´es. Les ´el´ements cl´es sont :

**2.1 Classe HolidayDAOImpl**

La classe HolidayDAOImpl impl´emente l’interface g´en´erique GenericDAOI pour g´erer les op´erations li´ees aux cong´es dans la base de donn´ees. Les principales fonctionnalit´es incluent :

**M´ethodes** :

* getType() : R´ecup`ere tous les types de cong´es disponibles depuis la table type.
* getEmploye() : Retourne une liste des employ´es sous forme de noms complets.
* updateSoldeDisponible(int empId, double newBalance) : Met a` jour le solde de cong´es d’un employ´e.
* getSoldeDisponible(int empId) : R´ecup`ere le solde de cong´es total d’un employ´e.
* getTakenHolidays(int empId) : Calcule le nombre total de jours de cong´e d´ej`a pris par un employ´e.
* getRemainingHolidayBalance(int empId) : Calcule le solde de cong´es restants pour un employ´e.
* canTakeHoliday(int empId, long duration) : V´erifie si un employ´e peut prendre un cong´e de dur´ee donn´ee.
* add(Holiday holiday) : Ajoute un nouveau cong´e en base tout en mettant `a jour le solde de cong´es de l’employ´e.
* findById(int id) : Recherche un cong´e par son identifiant.
* getHolidayById(int id) : R´ecup`ere un cong´e par son ID en renvoyant tous ses d´etails.
* getAllHolidays() : R´ecup`ere tous les cong´es enregistr´es sous forme de liste.

**2.2 Interface GenericDAOI**

Cette interface g´en´erique d´efinit des m´ethodes CRUD pour manipuler tout type d’entit´e, y compris les cong´es et les employ´es. Elle inclut les m´ethodes

:

T findById(int id) : Recherche une entit´e par son identifiant.

void add(T entity) : Ajoute une nouvelle entit´e dans la base de donn´ees.

List<T> findAll() : R´ecup`ere toutes les entit´es sous forme de liste. void update(T entity, int id) : Met a` jour une entit´e existante. void delete(int id) : Supprime une entit´e par son identifiant.

## 3. Vue

Cette section repr´esente l’interface utilisateur. Elle comprend deux vues principales :

**EmployeeView** : Pour g´erer les employ´es.

**HolidayView** : Pour g´erer les cong´es.

**3.1 Classe MainView**

La classe MainView est la fenˆetre principale de l’application, regroupant deux onglets pour l’affichage des employ´es et des cong´es. Les principales fonctionnalit´es incluent :

**Attributs** :

* tabbedPane : Un panneau d’onglets pour naviguer entre la gestion des employ´es et des cong´es.

**M´ethodes** :

* Initialisation de la fenˆetre principale avec deux onglets : ”Employ´es” et ”Cong´es”.
* Configuration de la taille de la fenˆetre et du comportement a` la fermeture.

## 4. Controˆleur

Le controˆleur joue le rˆole d’interm´ediaire entre la vue et le mod`ele, permettant la gestion des interactions utilisateur. Ses responsabilit´es incluent :

Ajouter des listeners aux ´el´ements de l’interface (boutons, champs de texte, etc.).

R´ecup´erer les donn´ees saisies par l’utilisateur et les transmettre au mod`ele pour manipulation.

Mettre a` jour l’interface apr`es chaque action (ajout, suppression, mise a` jour).

Afficher des messages d’information ou d’erreur selon les r´esultats des actions effectu´ees.

**4.1 HolidayController**

La classe HolidayController g´ere la logique des actions de l’utilisateur pour les cong´es. Elle permet d’ajouter, modifier et supprimer des cong´es en interagissant avec le mod`ele HolidayModel et la vue HolidayView. Les principales fonctionnalit´es incluent :

**M´ethodes** :

* addHoliday() : Permet d’ajouter un cong´e en validant les dates et le solde de cong´es.
* modifyHoliday() : Permet de modifier un cong´e en v´erifiant le solde et en ajustant les dates si n´ecessaire.
* deleteHoliday() : Permet de supprimer un cong´e s´electionn´e.
* convertToLocalDate(Date date) et convertToDate(LocalDate localDate) : Utilitaires pour convertir entre les types Date et LocalDate.

## 5. Main

Le point d’entr´ee de l’application est responsable de :

Initialiser la connexion a` la base de donn´ees.

Cr´eer les instances des classes n´ecessaires (DAO, Vue, Contrˆoleur).

Lancer l’application et afficher l’interface utilisateur.

La classe MAIN g´ere cette initialisation en instanciant les vues, les mod`eles et les controˆleurs pour les employ´es et les cong´es. Elle cr´ee ´egalement la vue principale MainView, qui regroupe les vues.

# Ex´ecution du Code

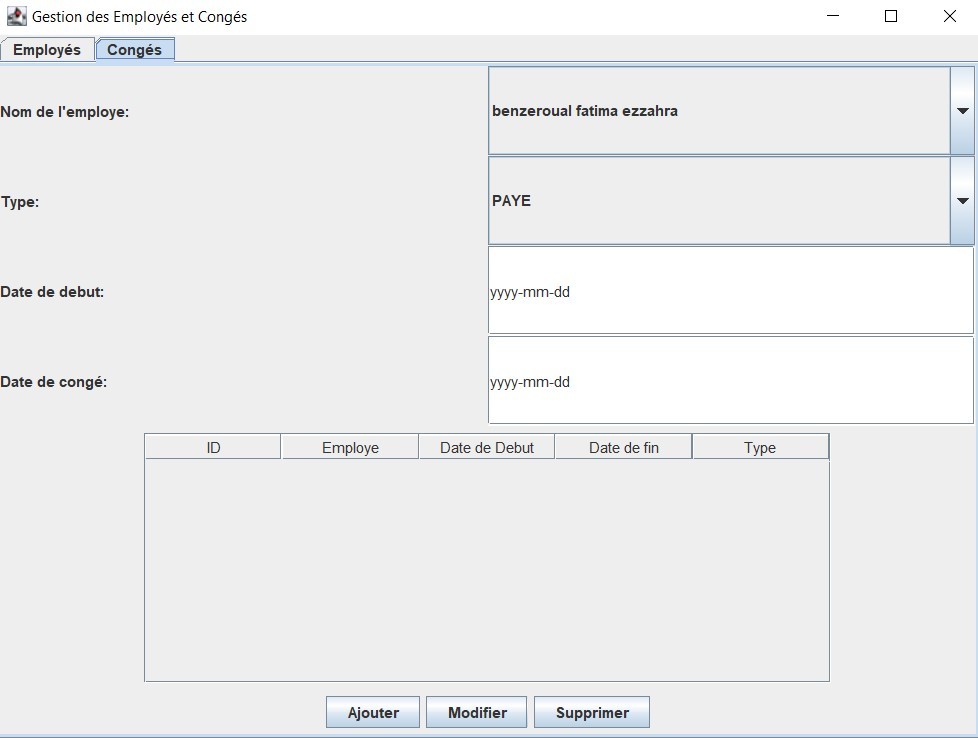


Figure 1: Gestion des cong´es

## Button Ajouter

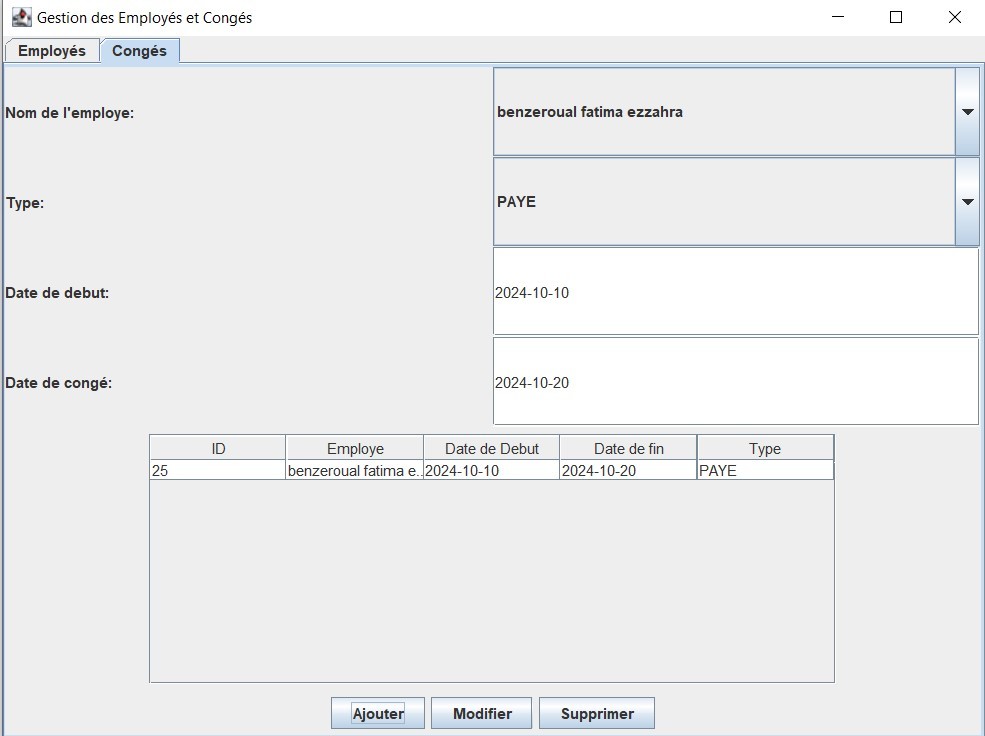


Figure 2: Ajout d’un cong´e

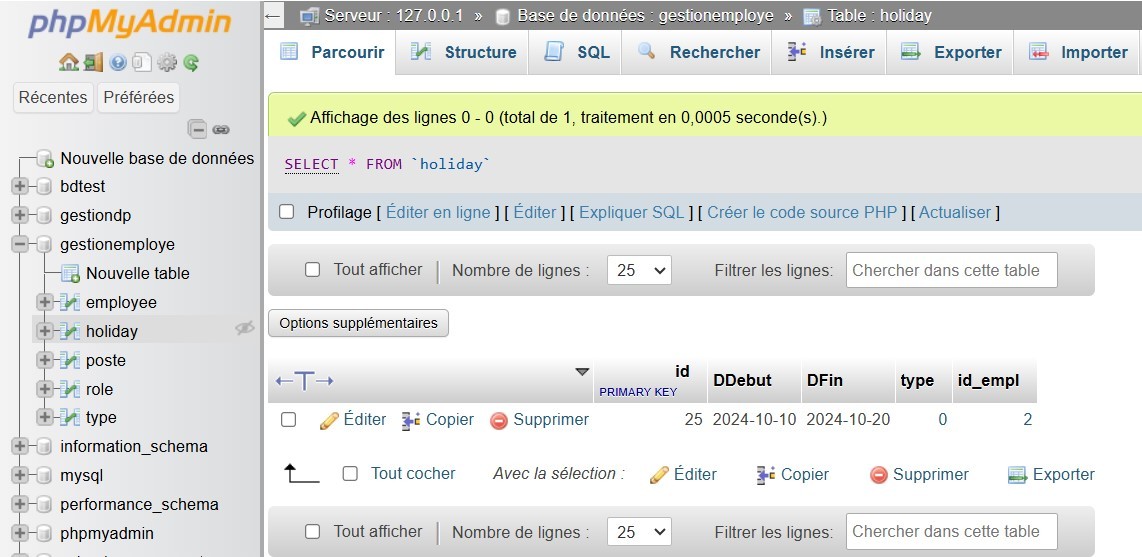


Figure 3: Base de donn´ees apr`es ajout

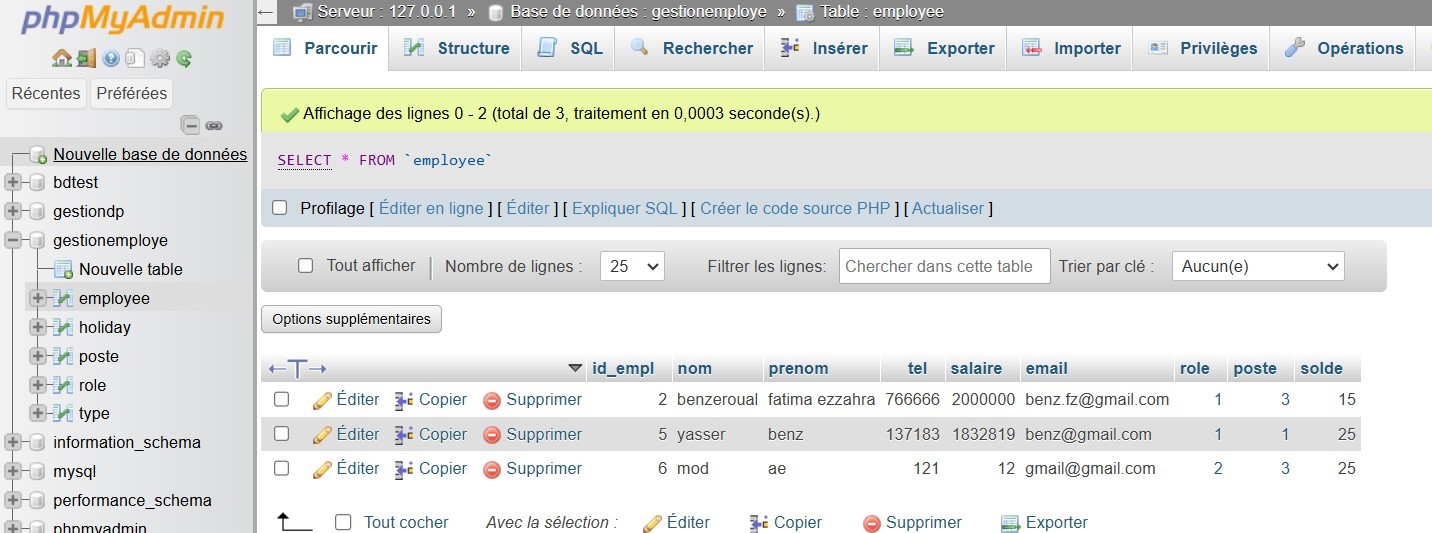


Figure 4: Solde modifi´e apr`es ajout

## Modifier

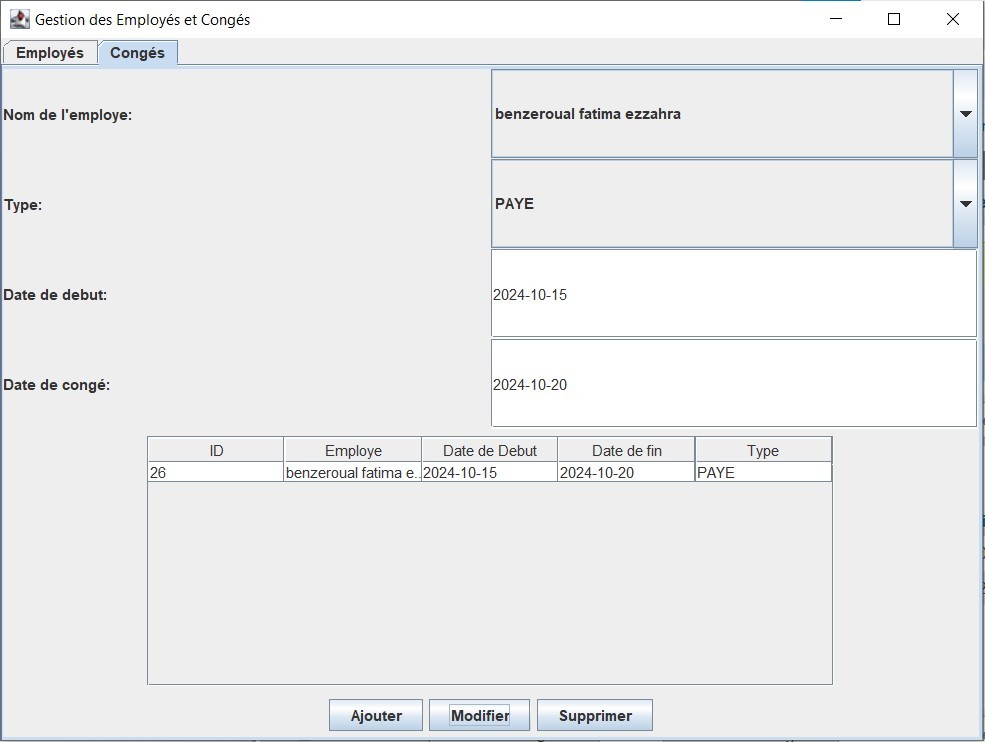


Figure 5: Modification du solde de cong´e

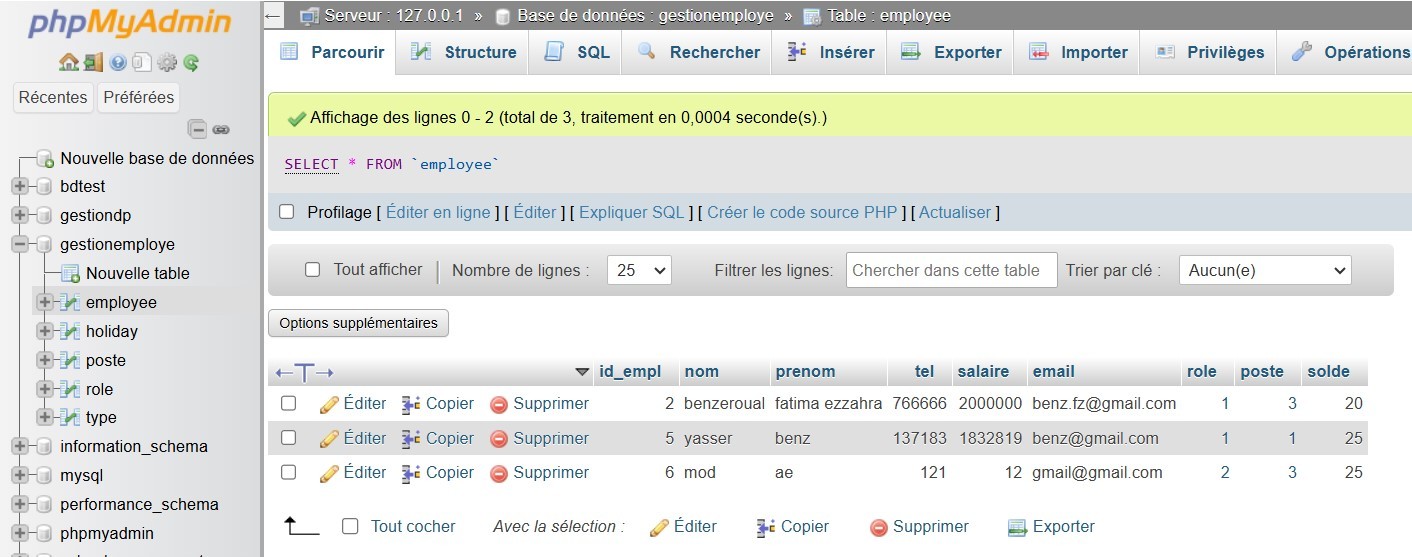


Figure 6: Base de donn´ees apr`es modification

## Button Supprimer

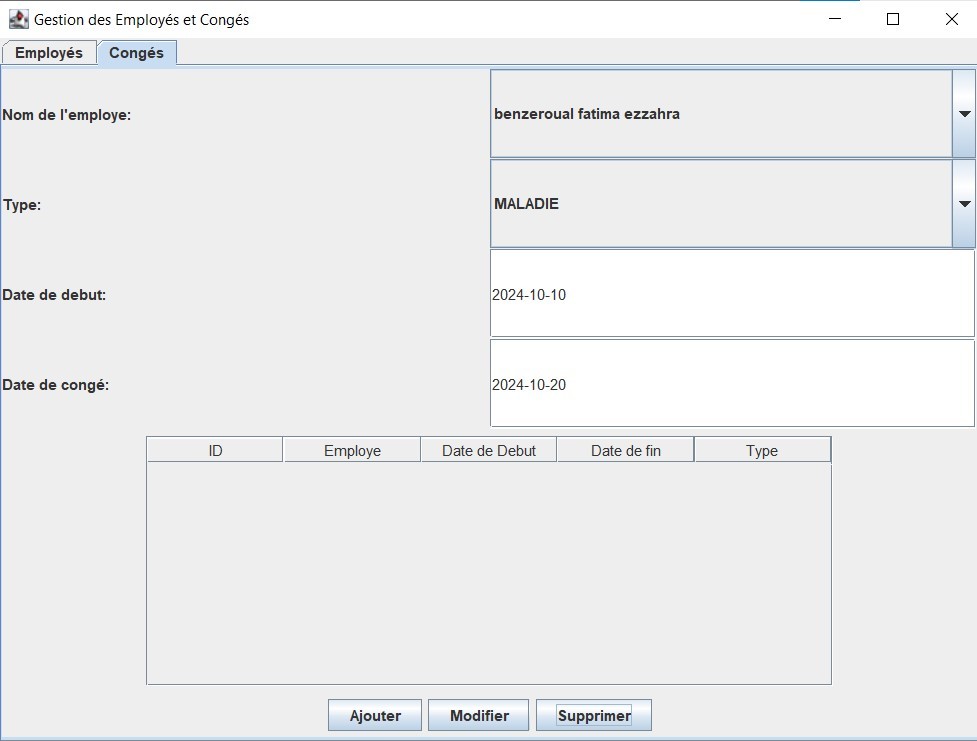


Figure 7: Suppression d’un cong´e

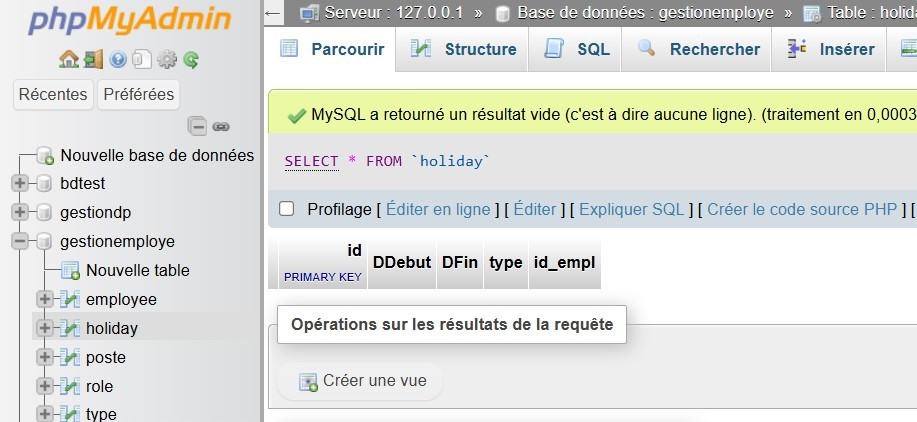


Figure 8: Base de donn´ees apr`es suppression

# Conclusion

Cette structure modulaire offre une s´eparation claire des responsabilit´es, simplifiant la maintenance et l’ajout de nouvelles fonctionnalit´es. Elle garantit une interaction fluide entre l’utilisateur et la base de donn´ees.