

► TD/TP 4: Concevez une application avec une BDD "MySQL"

Objectifs du TD/TP

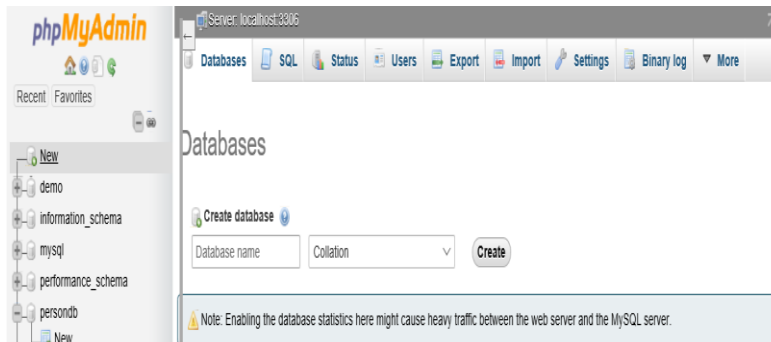
This image shows a full page of a document template. It consists of a white background with evenly spaced, horizontal light blue lines running across the entire width of the page. There are no margins, text, or other graphical elements present.

TD/TP 4: Concevez une application avec une BDD "MySQL"

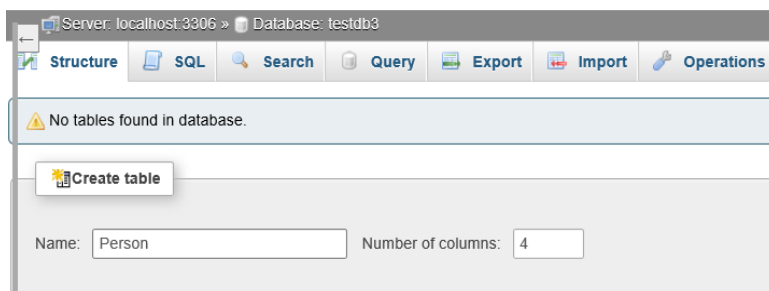
Exercice: Carnet de contacts

Etape 1. Remplissage de la base de Données à partir de phpMyAdmin

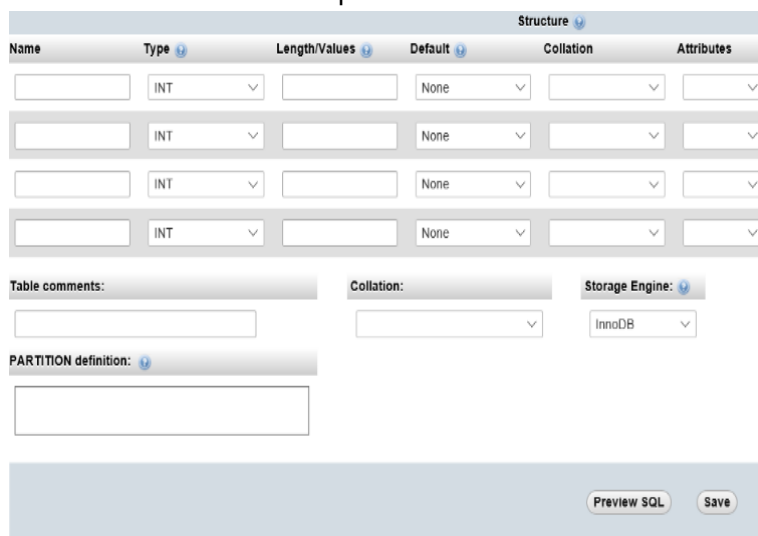
1. New -> Create Databases « persondb »



2. Créer la table Person



3. Créer les attributs de la table Person avec comme clé primaire id

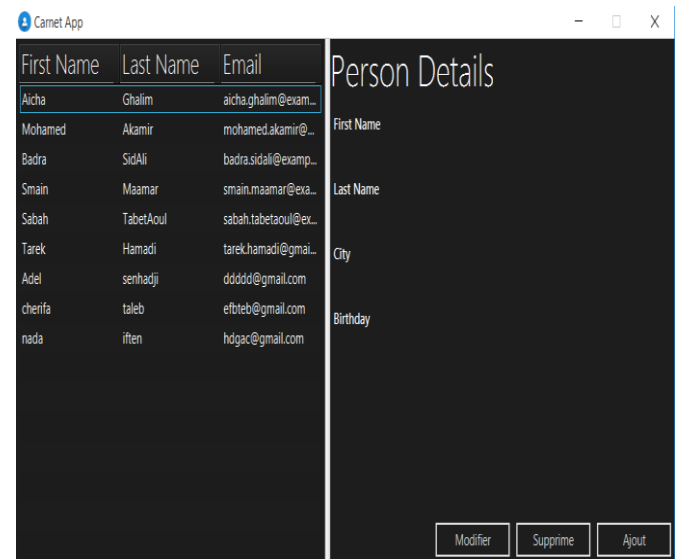


4. Remplir les différents enregistrements

+ Options						id	firstName	lastName	email
<input type="checkbox"/>	Edit	Copy	Delete			1	Aicha	Ghalim	aicha.ghalim@example.com
<input type="checkbox"/>	Edit	Copy	Delete			2	Mohamed	Akamir	mohamed.akamir@example.com
<input type="checkbox"/>	Edit	Copy	Delete			5	Badra	SidAli	badra.sidali@example.com
<input type="checkbox"/>	Edit	Copy	Delete			6	Smain	Maamar	smain.maamar@example.com
<input type="checkbox"/>	Edit	Copy	Delete			7	Sabah	TabetAoul	sabah.tabetoul@example.com
<input type="checkbox"/>	Edit	Copy	Delete			8	Tarek	Hamadi	tarek.hamadi@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete			9	Adel	senhadji	ddddd@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete			10	cherifa	taleb	efteb@gmail.com
<input type="checkbox"/>	Edit	Copy	Delete			11	nada	iften	hdgac@gmail.com

Etape 2 connexion de l'application Carnet de contacts à la BDD

1. Ajouter le driver comme librairie à votre projet déjà créé dans la fiche précédente **mysql-connector-java-5.1.44-bin**



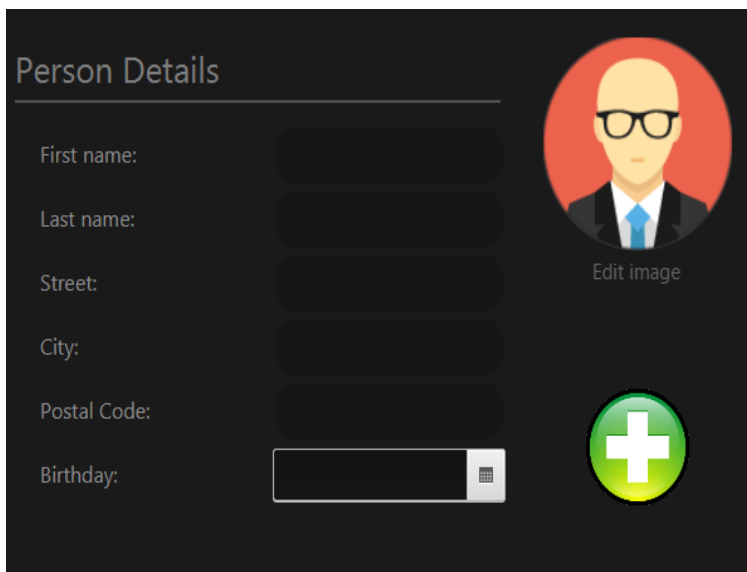
2. Créer une classe « dbConnector » pour tester la connexion à votre base de données


```
public class dbConnection {

    public Connection Connect() {
        try {
            String password = "root";
            String url =
"jdbc:mysql://localhost:3306/persondb";
            String user = "root";
            Class.forName("com.mysql.jdbc.Dr
iver");
            Connection conn = null;
            conn =
DriverManager.getConnection(url, user,
password);
            return conn;
        } catch (ClassNotFoundException |
SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

Etape 3 Ajout d'un nouveau contact

- I. Ajouter une nouvelle vue « Profile.fxml »



2. L'évènement associé au bouton  Permettra d'ajouter un nouveau contact à la base de données à partir du programme Java.

3. Ajouter la class **Person** qui a pour rôle de Model.
4. Définir le modele de **TableView** et **ObservableList** qui définit la liste des contacts pour une personne.
5. Charger la TableView en intégrant ce code

```
public void initialize(URL location,
ResourceBundle resources) {
    try {
        dc = new dbConnection();
        Connection conn = dc.Connect();
        ResultSet rs =
conn.createStatement().executeQuery("S
ELECT * FROM person");

        while (rs.next()){

            data.add(new
Person(rs.getString(2),
rs.getString(3), rs.getString(4)));

        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    firstNameCol.setCellValueFactory(
        new
PropertyValueFactory<>("firstName")
    );
    lastNameCol.setCellValueFactory(
        new
PropertyValueFactory<>("lastName")
    );
    emailCol.setCellValueFactory(
        new
PropertyValueFactory<>("email")
    );
    table.setItems(data);
}
}
```

6. Insérer une nouvelle personne en intégrant cette ligne de code :

```
Controller.data.add(new Person(
    addFirstName.getText(),
    addLastName.getText(),
    addEmail.getText()
));
```