文本, 白板

描述已自动生成

* ***University****:* JNU
* ***Department****:* Computer Science and Technology (CST)
* ***Course****:*Database System
* ***Project Title****:* **Database Project One**
* ***Author****:*蒋云翔 2022102330（Yunxiang Jiang）
* ***Instructor****:*吴汉瑞（Hanrui Wu）
* ***Date****:* December 4, 2024

Catalogue

[1. Objective 2](#_Toc184222888)

[2. JDBC Brief Intro 2](#_Toc184222889)

[3. JavaFX Brief Intro 3](#_Toc184222890)

[4. Project Introduction 4](#_Toc184222891)

[4.1 Key Features 5](#_Toc184222892)

[4.2 Technical Overview 5](#_Toc184222893)

[5 Project Development Environment 5](#_Toc184222894)

[6. File structure 6](#_Toc184222895)

[6](#_Toc184222896)

[7. Usage Guide 6](#_Toc184222897)

[7.1 Using MySQL command windows 6](#_Toc184222898)

[6](#_Toc184222899)

[7.2 Using MySQL Workbench 7](#_Toc184222900)

[7.3 Using our Java program 7](#_Toc184222901)

[8. Personal thoughts of this project 11](#_Toc184222902)

# 1. Objective

1. Write a java program to connect, access and update a real DB with JDBC. It should operate both on data and meta-data, work as a simple management tool for DBAs, accept SQL statements and show the results in proper text format.
2. Write a simple project report to describe the file structure of your program and a simple usage guide with snapshots to demonstrate the main functions.
3. Compress all your source code and the project report in pdf to a zip/rar file (named “CST\_DBS\_P1\_YourName”) and send it to the course email: 1920603969@qq.combefore 22pm, Dec. 4, 2024.

# 2. JDBC Brief Intro

JDBC API serves as a Java-based architecture that enables communication with various types of structured data, predominantly in relational databases. It addresses three key programming functions within Java applications. Comprising four essential elements: the JDBC API, which allows Java applications to engage with relational data; the JDBC Driver Manager, that facilitates connections between Java applications and JDBC drivers; the JDBC Test Suite, which aids in confirming the compatibility of JDBC drivers with certain programs; and the JDBC-ODBC Bridge, enabling JDBC connectivity via ODBC drivers. The process of utilizing JDBC in programming encompasses several stages: initiating a connection to a database or other data sources, dispatching queries and updates, and subsequently retrieving and managing the database's reactions to these requests.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Figure 1: Install JDBC

# 3. JavaFX Brief Intro

JavaFX is a Java-based framework designed for creating and delivering dynamic user interfaces, particularly for desktop applications and recently expanded to mobile and embedded systems. It encompasses a comprehensive set of tools for building rich internet applications (RIAs) that can run across a variety of platforms. JavaFX addresses the need for modern UI development with three primary components: the JavaFX SDK, which provides a rich set of UI controls and tools for developers; the JavaFX Runtime, which enables the execution of JavaFX applications; and the JavaFX Scene Graph, which facilitates the rendering of complex UIs with animations and graphics. The process of developing with JavaFX involves several stages: designing the user interface using JavaFX's extensive library of UI components, coding the application logic, and then deploying the application across different platforms.

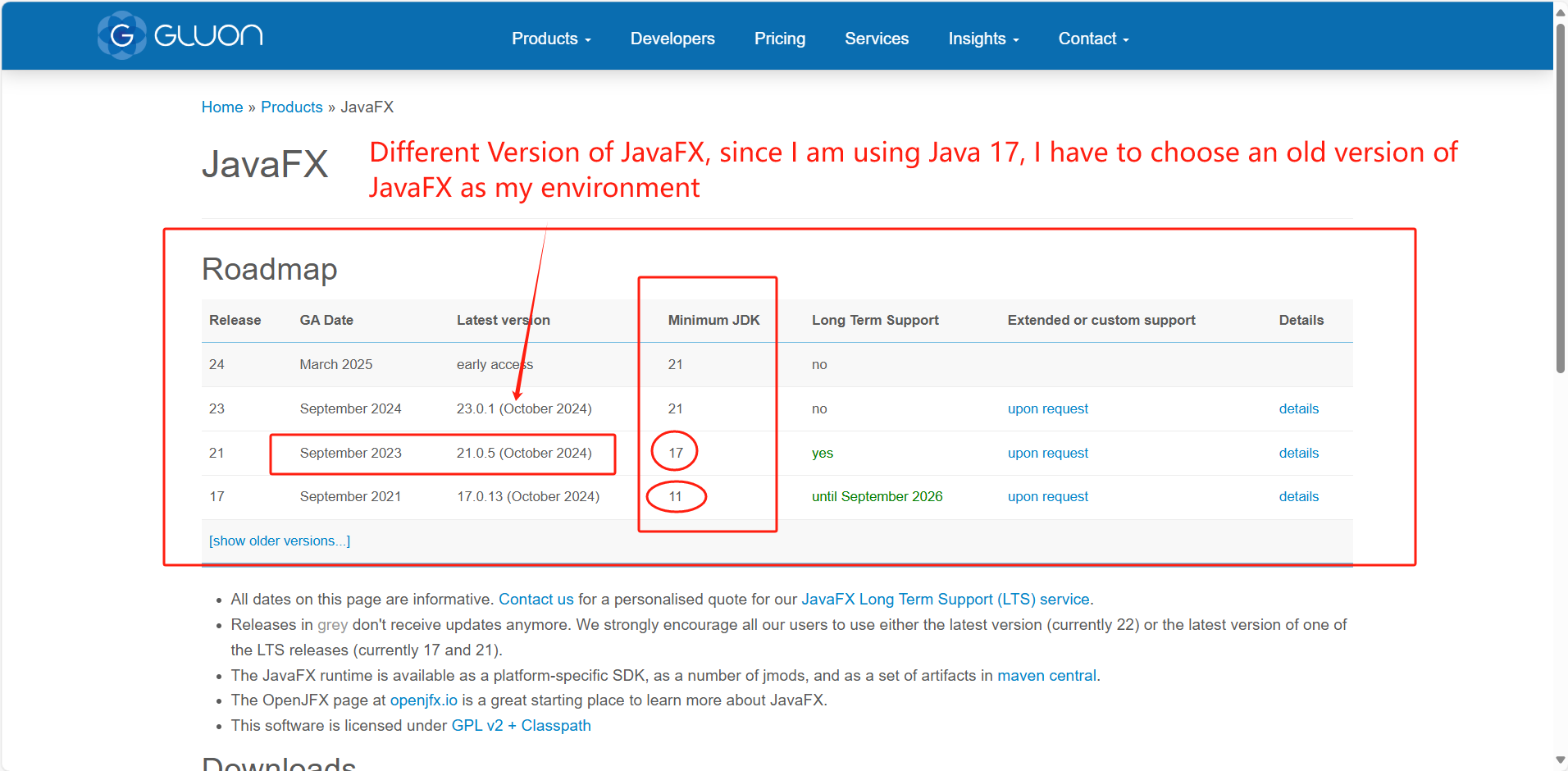


Figure 2: Install JavaFX (choose a suitable version)

# 4. Project Introduction

The **Personal Information Management System** is a simple yet efficient database application developed to manage personal information such as names, ages, genders, and addresses. Built using Java, JDBC, and JavaFX, this system provides a user-friendly interface for adding, viewing, updating, and deleting personal data records in a MySQL database.

图示, 示意图

描述已自动生成

## 4.1 Key Features

1. **User-Friendly Interface:** The system uses JavaFX to create an intuitive and easy-to-navigate graphical user interface (GUI), allowing users to interact with the database through simple form inputs and buttons.
2. **Database Connectivity:** Leveraging JDBC (Java Database Connectivity), the application establishes a reliable connection to a MySQL database, facilitating seamless data transactions and operations.
3. **CRUD Operations:** The system supports full CRUD (Create, Read, Update, Delete) functionalities:
   * **Create:** Users can add new personal information records.
   * **Read:** Users can view existing records in a table format.
   * **Update:** Users can select and modify existing records.
   * **Delete:** Users can remove records from the database.
4. **Error Handling and Data Integrity:** The system ensures robust error handling to manage database exceptions and maintain data integrity during transactions.

## 4.2 Technical Overview

* **Java:** The core programming language used to develop the application.
* **JavaFX:** Utilized for creating a responsive and visually appealing user interface.
* **JDBC:** Provides the bridge between the Java application and the MySQL database, enabling SQL operations.
* **MySQL:** The relational database management system used to store and manage the personal information records.

# 5 Project Development Environment

*JDK: Oracle JDK, Version 17.0.10*

*Database: MySQL, Version 8.0.40*

*JDBC Driver: Version 8.0.25*

*IDE: IntelliJ IDEA Community Edition 2023.3.5*

*OS: Windows 11*

# 6. File structure

# 

Figure 3: Project’s file structure

# 7. Usage Guide

## 7.1 Using MySQL command windows

## 

Figure 4: Using MySQL command windows

## 7.2 Using MySQL Workbench

图形用户界面, 文本, 应用程序, Word

描述已自动生成

Figure 5: Using MySQL Workbench step1

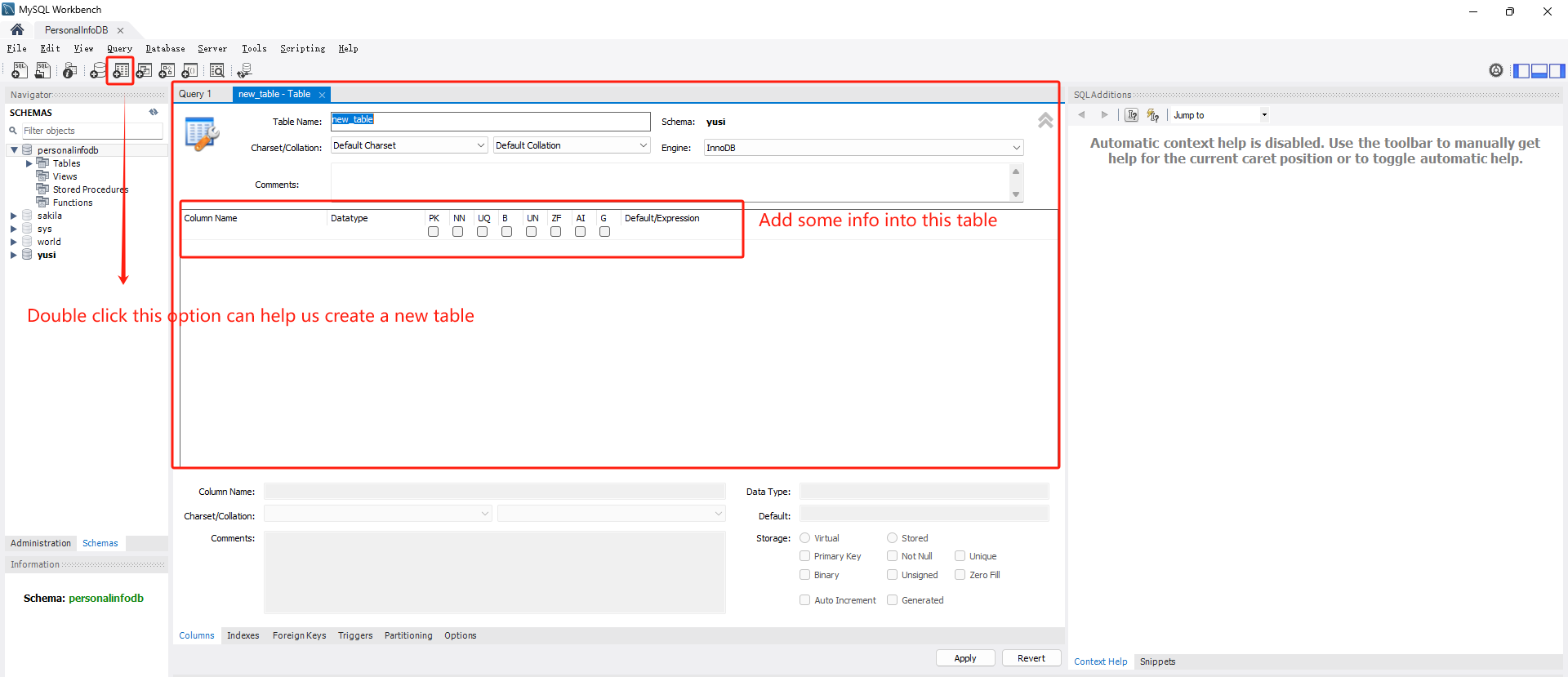


Figure 6: Using MySQL Workbench step2

## 7.3 Using our Java program

PersonalInfoManagementUI.java is the sequential execution entry of the entire program, but other partial files can also be used as separate program entries. When PersonalInfoManagementUI.java is started, the window will open in a magnified form. This is a common human-computer interaction delaying technique and reduces the user's sensitivity to the loading speed of other components.

文本

描述已自动生成

Figure 7: Using our Java program

When the PersonalInfoManagementUI.java is loaded, it will connect directly to our local MySQL database, which can be directly added, deleted, changed and checked (we have pre-placed the username and password in other files to connect to the database).

电脑萤幕的截图

描述已自动生成

Figure 8: Run the specific java file

In this interface, you can add, delete, modify and check a series of operations, the user interface is very simple and clear, in line with the design trend of The Times

图形用户界面, 应用程序, Word

描述已自动生成

Figure 9: User interface

图形用户界面, 应用程序

描述已自动生成

Figure 10: Add operation

图形用户界面, 应用程序, Word

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

Figure 11: Update operation and its result

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

Figure 12: Delete operation and its result

# 8. Personal thoughts of this project

Completing the Personal Information Management System project was both rewarding and educational. I deepened my understanding of Java, JDBC, and JavaFX while gaining practical experience in database operations. Designing a user-friendly interface and implementing CRUD functionalities highlighted the importance of intuitive UI/UX design and robust error handling. The project underscored the integration of various technologies to create a seamless application, enhancing my problem-solving skills and inspiring me to tackle more complex projects in the future. Overall, it was a fulfilling journey that solidified my skills and expanded my horizons in software development.