

Bruce Eckels Mindview Inc Thinking In C 2nd Edition

Thinking in Java Thinking in Java Thinking In C++ (2Nd Edition) Thinking in C++ Refactoring to Patterns On Java 8 Thinking in Java vol. 1 Fondamenti Thinking in Java vol. 2 Tecniche avanzate C/C++ Users Journal Computational Science - ICCS 2002 Learning MariaDB Learn Java in One Week CRYPTOGRAPHY AND IMAGE PROCESSING with Java GUI and SQLite POSTGRESQL FOR JAVA GUI: Database and Image Processing The Best Guide to Database Programming with Java GUI, PostgreSQL, and SQL Server MS Access And SQL Server Crash Course: A Step by Step, Project-Based Introduction to Java GUI Programming BUILDING THREE DESKTOP APPLICATIONS USING JAVA GUI AND MYSQL THE BEST WAY TO LEARN JAVA GUI WITH MYSQL AND SQL SERVER JAVA GUI WITH MYSQL: Database and Image Processing Access Database with JDBC LEARNING SQL SERVER Learn JDBC By Example: A Quick Start Guide to MariaDB and SQL Server Driven Programming LEARN JDBC THE HARD WAY: A Hands-On Reference to MySQL and SQL Server Driven Programming The Best Tutorial to Learn Database Programming with Java GUI, MariaDB, and SQL Server A Walkthrough, Database-Driven Programming with Java GUI for Pragmatic Programmers MariaDB with Java GUI for Cryptography and Image Processing The Quick Way to Learn Java GUI with MariaDB and SQLite SQLite with JDBC for Beginners FROM ZERO TO JDBC HERO POSTGRESQL FOR JAVA GUI: Database, Cryptography, and Image Processing A Comprehensive Guide to Java GUI Programming with SQLite MASTERING SQL SERVER with Java GUI for Pragmatic Programmers Java In Practice: JDBC And Database Applications JAVA GUI WITH MYSQL The Fast Way to Learn Java GUI with PostgreSQL and SQLite Database and Image Processing Using Java GUI and Microsoft Access Learn SQLite with JDBC Java In Action: An Excellent Guide to Explore JDBC And Database Applications Java Programming For Developers: The Definitive Guide to Learn JDBC And Database Applications Thinking in C++

As recognized, adventure as well as experience very nearly lesson, amusement, as without difficulty as harmony can be gotten by just checking out a ebook Bruce Eckels Mindview Inc Thinking In C 2nd Edition along with it is not directly done, you could assume even more something like this life, on the world.

We give you this proper as well as easy mannerism to acquire those all. We allow Bruce Eckels Mindview Inc Thinking In C 2nd Edition and numerous books collections from fictions to scientific research in any way. accompanied by them is this Bruce Eckels Mindview Inc Thinking In C 2nd Edition that can be your partner.

THE BEST WAY TO LEARN JAVA GUI WITH MYSQL AND SQL SERVER Jul 20 2021 *This hands-on tutorial/reference/guide to MySQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from MySQL and SQL Server. As you would expect, this book shows how to build from scratch two different databases: MySQL and SQL Server using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create Crime database and its tables. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MySQL/SQL SERVER programmer.*

Thinking In C++ (2Nd Edition) Nov 04 2022

Thinking in Java Jan 06 2023 *With humor and insight, the author introduces the fundamental concepts of the Java programming language, from object development to design patterns, with the help of straightforward examples. By the author of Thinking in C+++. Original. (Beginner).*

A Walkthrough, Database-Driven Programming with Java GUI for Pragmatic Programmers Dec 13 2020 *This covers how to implement SQLite and SQL Server driven Java GUI programming. The lessons in this book are a highly organized and well-*

indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch a SQLite database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQLite and SQL Server in Java. In chapter one, you will learn: How to create SQLite database and six tables In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six tables. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create SQL Server database and its tables. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

SQLite with JDBC for Beginners Sep 09 2020 In this book, you will learn how to build from scratch a SQLite database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQLite in Java. In the first chapter, you will learn: How to create SQLite database and six tables In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six tables. In the last chapter, you will study how to query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/SQLite programmer.

POSTGRESQL FOR JAVA GUI: Database and Image Processing Nov 23 2021 In this book, you will learn how to build from scratch a criminal records management database system using Java/PostgreSQL. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. There are only three external libraries used in this book: Connector / J to facilitate Java to MySQL connections, JCalendar to display calendar controls, and JFreeChart to display graphics. Digital image techniques to extract image features used in this book are grascaling, sharpening, invertering, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In third chapter, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In the fourth chapter, you will be taught how to create Crime database and its tables. In the fifth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In the sixth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In the seventh chapter, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight

columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In the eighth chapter, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key), `police_station_id` (foreign key), `investigator_id` (foreign key), `victim_id` (foreign key), `status`, and `description`. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful for you.

Thinking in C++ Aug 28 2019 Best selling author Bruce Eckel has joined forces with Chuck Allison to write *Thinking in C++*, Volume 2, the sequel to the highly received and best selling *Thinking in C++*, Volume 1. Eckel is the master of teaching professional programmers how to quickly learn cutting edge topics in C++ that are glossed over in other C++ books. In *Thinking in C++*, Volume 2, the authors cover the finer points of exception handling, defensive programming and string and stream processing that every C++ programmer needs to know. Special attention is given to generic programming where the authors reveal little known techniques for effectively using the Standard Template Library. In addition, Eckel and Allison demonstrate how to apply RTTI, design patterns and concurrent programming techniques to improve the quality of industrial strength C++ applications. This book is targeted at programmers of all levels of experience who want to master C++.

On Java 8 Aug 01 2022

JAVA GUI WITH MYSQL: Database and Image Processing Jun 18 2021 In this book, you will learn how to build from scratch a criminal records management database system using Java / MySQL. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. There are only three external libraries used in this book: `Connector / J` to facilitate Java to MySQL connections, `JCalendar` to display calendar controls, and `JFreeChart` to display graphics. Digital image techniques to extract image features used in this book are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In the first chapter, you will be shown the number of devices needed to be downloaded and installed. You need to know how to add external libraries to the NetBeans environment. These tools are needed so that you can run the Java scripts. In the second chapter, you will be taught how to create Crime database and its tables. In third chapter, you will be taught how to extract image features, utilizing `BufferedImage` class, in Java GUI. In the fourth chapter, you will be taught to create Java GUI to view, edit, insert, and delete `Suspect` table data. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother_name`, `address`, `telephone`, and `photo`. In the fifth chapter, you will be taught to create Java GUI to view, edit, insert, and delete `Feature_Extraction` table data. This table has eight columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In the sixth chapter, you will add two tables: `Police_Station` and `Investigator`. These two tables will later be joined to `Suspect` table through another table, `File_Case`, which will be built in the seventh chapter. The `Police_Station` has six columns: `police_station_id` (primary key), `location`, `city`, `province`, `telephone`, and `photo`. The `Investigator` has eight columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In the seventh chapter, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key), `police_station_id` (foreign key), `investigator_id` (foreign key), `victim_id` (foreign key), `status`, and `description`. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful for you.

The Best Guide to Database Programming with Java GUI, PostgreSQL, and SQL Server Oct 23 2021 This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to PostgreSQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from PostgreSQL and SQL Server. As you would expect, this book shows how to build from scratch two different databases: PostgreSQL and SQL Server using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In chapter two, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In chapter three, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. You will also learn how to create and store salt passwords and verify them. In chapter four, you will create a PostgreSQL database, named Bank, and its tables. In chapter five, you will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter six, you will create an Account table. This account table has the following ten fields: `account_id` (primary key), `client_id` (primarykey), `account_number`, `account_date`, `account_type`, `plain_balance`, `cipher_balance`, `decipher_balance`, `digital_signature`, and `signature_verification`. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter seven, you create a

table named *Client_Data*, which has seven columns: *client_data_id* (primary key), *account_id* (primary key), *birth_date*, *address*, *mother_name*, *telephone*, and *photo_path*. In chapter eight, you will be taught how to create a SQL Server database, named *Crime*, and its tables. In chapter nine, you will be taught how to extract image features, utilizing *BufferedImage* class, in Java GUI. In chapter ten, you will be taught to create Java GUI to view, edit, insert, and delete *Suspect* table data. This table has eleven columns: *suspect_id* (primary key), *suspect_name*, *birth_date*, *case_date*, *report_date*, *suspect_status*, *arrest_date*, *mother_name*, *address*, *telephone*, and *photo*. In chapter eleven, you will be taught to create Java GUI to view, edit, insert, and delete *Feature_Extraction* table data. This table has eight columns: *feature_id* (primary key), *suspect_id* (foreign key), *feature1*, *feature2*, *feature3*, *feature4*, *feature5*, and *feature6*. In chapter twelve, you will add two tables: *Police_Station* and *Investigator*. These two tables will later be joined to *Suspect* table through another table, *File_Case*, which will be built in the seventh chapter. The *Police_Station* has six columns: *police_station_id* (primary key), *location*, *city*, *province*, *telephone*, and *photo*. The *Investigator* has eight columns: *investigator_id* (primary key), *investigator_name*, *rank*, *birth_date*, *gender*, *address*, *telephone*, and *photo*. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter thirteen, you will add two tables: *Victim* and *File_Case*. The *File_Case* table will connect four other tables: *Suspect*, *Police_Station*, *Investigator* and *Victim*. The *Victim* table has nine columns: *victim_id* (primary key), *victim_name*, *crime_type*, *birth_date*, *crime_date*, *gender*, *address*, *telephone*, and *photo*. The *File_Case* has seven columns: *file_case_id* (primary key), *suspect_id* (foreign key), *police_station_id* (foreign key), *investigator_id* (foreign key), *victim_id* (foreign key), *status*, and *description*. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/PostgreSQL/SQL Server programmer.

Java Programming For Developers: The Definitive Guide to Learn JDBC And Database Applications Sep 29 2019 This step-by-step guide to explore database programming using Java is ideal for people with little or no programming experience. The goal of this concise book is not just to teach you Java, but to help you think like a programmer. Each brief chapter covers the material for one week of a college course to help you practice what you've learned. As you would expect, this book shows how to build from scratch two different databases: *MariaDB* and *SQLite* using Java. In designing a GUI and as an IDE, you will make use of the *NetBeans* tool. In the first chapter, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a *KeyStore*, generate *PrivateKey* and *PublicKey*, encrypt / decrypt data, and generate and verify digital prints. In the second chapter, you will learn how to create and store salt passwords and verify them. You will create a *Login* table. In this case, you will see how to create a Java GUI using *NetBeans* to implement it. In addition to the *Login* table, in this chapter you will also create a *Client* table. In the case of the *Client* table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In the third chapter, you will create an *Account* table. This account table has the following ten fields: *account_id* (primary key), *client_id* (primarykey), *account_number*, *account_date*, *account_type*, *plain_balance*, *cipher_balance*, *decipher_balance*, *digital_signature*, and *signature_verification*. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In the fourth chapter, You create a table with the name of the *Account*, which has ten columns: *account_id* (primary key), *client_id* (primarykey), *account_number*, *account_date*, *account_type*, *plain_balance*, *cipher_balance*, *decipher_balance*, *digital_signature*, and *signature_verification*. In the fifth chapter, you will create a *Client_Data* table, which has the following seven fields: *client_data_id* (primary key), *account_id* (primary_key), *birth_date*, *address*, *mother_name*, *telephone*, and *photo_path*. In chapter six, you will be shown how to create *SQLite* database and tables with Java. In chapter seven, you will be taught how to extract image features, utilizing *BufferedImage* class, in Java GUI. Digital image techniques to extract image features used in this chapter are *grayscale*, *sharpening*, *inverting*, *blurring*, *dilation*, *erosion*, *closing*, *opening*, *vertical prewitt*, *horizontal prewitt*, *Laplacian*, *horizontal sobel*, and *vertical sobel*. For readers, you can develop it to store other advanced image features based on descriptors such as *SIFT* and others for developing descriptor based matching. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete *Suspect* table data. This table has eleven columns: *suspect_id* (primary key), *suspect_name*, *birth_date*, *case_date*, *report_date*, *suspect_status*, *arrest_date*, *mother_name*, *address*, *telephone*, and *photo*. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete *Feature_Extraction* table data. This table has eight columns: *feature_id* (primary key), *suspect_id* (foreign key), *feature1*, *feature2*, *feature3*, *feature4*, *feature5*, and *feature6*. All six fields (except keys) will have a *BLOB* data type, so that the image of the feature will be directly saved into this table. In chapter ten, you will add two tables: *Police_Station* and *Investigator*. These two tables will later be joined to *Suspect* table through another table, *File_Case*, which will be built in the seventh chapter. The *Police_Station* has six columns: *police_station_id* (primary key), *location*, *city*, *province*, *telephone*, and *photo*. The *Investigator* has eight columns: *investigator_id* (primary key), *investigator_name*, *rank*, *birth_date*, *gender*, *address*, *telephone*, and *photo*. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: *Victim* and *Case File*. The *File_Case* table will connect four other tables: *Suspect*, *Police_Station*, *Investigator* and *Victim*. The *Victim* table has nine columns: *victim_id* (primary key), *victim_name*, *crime_type*, *birth_date*, *crime_date*, *gender*, *address*, *telephone*, and *photo*. The *Case File* has seven columns: *case_file_id* (primary key), *suspect_id* (foreign key), *police_station_id* (foreign key), *investigator_id* (foreign key), *victim_id* (foreign key), *status*, and *description*. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MariaDB/SQLite programmer.

Refactoring to Patterns Sep 02 2022 In 1994, *Design Patterns* changed the landscape of object-oriented development by introducing classic solutions to recurring design problems. In 1999, *Refactoring* revolutionized design by introducing an effective process for improving code. With the highly anticipated *Refactoring to Patterns*, *Joshua Kerievsky* has changed our approach to design by forever uniting patterns with the evolutionary process of refactoring. This book introduces the theory and practice of pattern-directed refactorings: sequences of low-level refactorings that allow designers to safely move designs to, towards, or away from pattern implementations. Using code from real-world projects, *Kerievsky* documents the thinking and steps underlying over two dozen pattern-based design transformations. Along the way he offers insights into pattern differences and how to implement patterns in the simplest possible ways. Coverage includes: A catalog of twenty-seven pattern-directed refactorings, featuring real-world code examples Descriptions of twelve design

smells that indicate the need for this book's refactorings General information and new insights about patterns and refactoring Detailed implementation mechanics: how low-level refactorings are combined to implement high-level patterns Multiple ways to implement the same pattern—and when to use each Practical ways to get started even if you have little experience with patterns or refactoring Refactoring to Patterns reflects three years of refinement and the insights of more than sixty software engineering thought leaders in the global patterns, refactoring, and agile development communities. Whether you're focused on legacy or "greenfield" development, this book will make you a better software designer by helping you learn how to make important design changes safely and effectively.

Thinking in C++ Oct 03 2022

MASTERING SQL SERVER with Java GUI for Pragmatic Programmers May 06 2020 This is a comprehensive, in-depth introduction to the core Java language book. This book will help you quickly write efficient, high-quality SQL-Server-based code with Java. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch a SQL Server database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQL Server in Java. In chapter one, you will be taught how to create Crime database and its tables. In chapter two, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter three, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter four, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter five, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter six, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. In chapter seven, you will create School database and six tables. In chapter eight, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter nine, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In the last chapter, you will study how to query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/SQL Server programmer.

MS Access And SQL Server Crash Course: A Step by Step, Project-Based Introduction to Java GUI Programming Sep 21 2021 This is a Java GUI crash course. This book will help you quickly write efficient, high-quality access-database-driven code with Java. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch two access database management systems using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will create School database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will create dan configure database. In chapter six, you will some image processing tehniques using Java. In chapter seven, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter nine, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id

(foreign key), *police_id* (foreign key), *investigator_id* (foreign key), *victim_id* (foreign key), *status*, and *description*. You will create GUI to display, edit, insert, and delete for both tables as well.

Thinking in Java Dec 05 2022 An overview of the programming language's fundamentals covers syntax, initialization, implementation, classes, error handling, objects, applets, multiple threads, projects, and network programming.

Learn SQLite with JDBC Dec 01 2019 In this book, you will learn how to build from scratch a criminal records management database system using Java / SQLite. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. In the first chapter, you will be shown how to create SQLite database and tables with Java. In second chapter, you will be taught how to extract image features, utilizing *BufferedImage* class, in Java GUI. Digital image techniques to extract image features used in this chapter are *grayscale*, *sharpening*, *inverting*, *blurring*, *dilation*, *erosion*, *closing*, *opening*, *vertical prewitt*, *horizontal prewitt*, *Laplacian*, *horizontal sobel*, and *vertical sobel*. For readers, you can develop it to store other advanced image features based on descriptors such as *SIFT* and others for developing descriptor based matching. In the third chapter, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: *suspect_id* (primary key), *suspect_name*, *birth_date*, *case_date*, *report_date*, *suspect_status*, *arrest_date*, *mother_name*, *address*, *telephone*, and *photo*. In the fourth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: *feature_id* (primary key), *suspect_id* (foreign key), *feature1*, *feature2*, *feature3*, *feature4*, *feature5*, and *feature6*. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In the fifth chapter, you will add two tables: *Police_Station* and *Investigator*. These two tables will later be joined to Suspect table through another table, *File_Case*, which will be built in the seventh chapter. The *Police_Station* has six columns: *police_station_id* (primary key), *location*, *city*, *province*, *telephone*, and *photo*. The *Investigator* has eight columns: *investigator_id* (primary key), *investigator_name*, *rank*, *birth_date*, *gender*, *address*, *telephone*, and *photo*. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In the sixth chapter, you will add two tables: *Victim* and *Case_File*. The *File_Case* table will connect four other tables: *Suspect*, *Police_Station*, *Investigator* and *Victim*. The *Victim* table has nine columns: *victim_id* (primary key), *victim_name*, *crime_type*, *birth_date*, *crime_date*, *gender*, *address*, *telephone*, and *photo*. The *Case_File* has seven columns: *case_file_id* (primary key), *suspect_id* (foreign key), *police_station_id* (foreign key), *investigator_id* (foreign key), *victim_id* (foreign key), *status*, and *description*. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful for you.

FROM ZERO TO JDBC HERO Aug 09 2020 In this book, you will learn how to build from scratch a criminal records management database system using Java/SQLite. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. In chapter one, you will create Bank database and its four tables. In chapter two, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a *KeyStore*, generate *PrivateKey* and *PublicKey*, encrypt / decrypt data, and generate and verify digital prints. In chapter three, you will learn how to create and store salt passwords and verify them. You will create a *Login* table. In this case, you will see how to create a Java GUI using *NetBeans* to implement it. In addition to the *Login* table, in this chapter you will also create a *Client* table. In the case of the *Client* table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter four, you will create an *Account* table. This account table has the following ten fields: *account_id* (primary key), *client_id* (primary key), *account_number*, *account_date*, *account_type*, *plain_balance*, *cipher_balance*, *decipher_balance*, *digital_signature*, and *signature_verification*. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter five, you will create a *Client_Data* table, which has the following seven fields: *client_data_id* (primary key), *account_id* (primary key), *birth_date*, *address*, *mother_name*, *telephone*, and *photo_path*. In chapter six, you will create *Crime* database and its six tables. In chapter seven, you will be taught how to extract image features, utilizing *BufferedImage* class, in Java GUI. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: *suspect_id* (primary key), *suspect_name*, *birth_date*, *case_date*, *report_date*, *suspect_status*, *arrest_date*, *mother_name*, *address*, *telephone*, and *photo*. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: *feature_id* (primary key), *suspect_id* (foreign key), *feature1*, *feature2*, *feature3*, *feature4*, *feature5*, and *feature6*. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter ten, you will add two tables: *Police_Station* and *Investigator*. These two tables will later be joined to Suspect table through another table, *File_Case*, which will be built in the seventh chapter. The *Police_Station* has six columns: *police_station_id* (primary key), *location*, *city*, *province*, *telephone*, and *photo*. The *Investigator* has eight columns: *investigator_id* (primary key), *investigator_name*, *rank*, *birth_date*, *gender*, *address*, *telephone*, and *photo*. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: *Victim* and *File_Case*. The *File_Case* table will connect four other tables: *Suspect*, *Police_Station*, *Investigator* and *Victim*. The *Victim* table has nine columns: *victim_id* (primary key), *victim_name*, *crime_type*, *birth_date*, *crime_date*, *gender*, *address*, *telephone*, and *photo*. The *File_Case* has seven columns: *file_case_id* (primary key), *suspect_id* (foreign key), *police_station_id* (foreign key), *investigator_id* (foreign key), *victim_id* (foreign key), *status*, and *description*. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

C/C++ Users Journal Apr 28 2022

Learn JDBC By Example: A Quick Start Guide to MariaDB and SQL Server Driven Programming Mar 16 2021 This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of *MariaDB* and *SQL Server* databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to *MariaDB* and *SQL Server* is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from *MariaDB* and *SQL Server*. As you would expect, this book shows how to build from scratch two different databases: *MariaDB* and *SQL Server* using Java. In designing a GUI and

as an IDE, you will make use of the NetBeans tool. In chapter one, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. You will also learn how to create and store salt passwords and verify them. In chapter two, you will create a PostgreSQL database, named Bank, and its tables. In chapter three, you will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter four, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter five, you create a table named Client_Data, which has seven columns: client_data_id (primary key), account_id (primary_key), birth_date, address, mother_name, telephone, and photo_path. In chapter six, you will be taught how to create a SQL Server database, named Crime, and its tables. In chapter seven, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter ten, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MariaDB/SQL Server programmer.

Java In Practice: JDBC And Database Applications Apr 04 2020 This hands-on introduction to database programming using Java is ideal for people with little or no programming experience. The goal of this concise book is not just to teach you Java, but to help you think like a programmer. Each brief chapter covers the material for one week of a college course to help you practice what you've learned. As you would expect, this book shows how to build from scratch two different databases: MySQL and SQLite using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be shown how to create SQLite database and tables with Java. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. Digital image techniques to extract image features used in this chapter are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter nine, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and Case File. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MySQL/SQLite programmer.

LEARNING SQL SERVER Apr 16 2021 The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications).The finished product is the reward, but the readers are fully engaged and

enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch a SQL Server database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQL Server in Java. In chapter one, you will learn: How to create SQL Server database and six tables. In the chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In the last chapter, you will study how to query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/SQL Server programmer.

CRYPTOGRAPHY AND IMAGE PROCESSING with Java GUI and SQLite Dec 25 2021 The book details how programmers and database professionals can develop SQLite-based Java GUI applications that involves cryptography and image processing. In this book, you will learn how to build from scratch a criminal records management database system using Java/SQLite. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. In chapter one, you will create Bank database and its four tables. In chapter two, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. In chapter three, you will learn how to create and store salt passwords and verify them. You will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter four, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter five, you will create a Client_Data table, which has the following seven fields: client_data_id (primary key), account_id (primary_key), birth_date, address, mother_name, telephone, and photo_path. In chapter six, you will create Crime database and its six tables. In chapter seven, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Feature Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter ten, you will add two tables: Police Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

BUILDING THREE DESKTOP APPLICATIONS USING JAVA GUI AND MYSQL Aug 21 2021 In this book, you will learn how to build from scratch a MySQL database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use MySQL in Java. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. In chapter six, you will learn how to create and store salt passwords and verify them. You will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter seven, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter eight, you create a table with the name of the Account, which has ten columns: account_id (primary

key), `client_id` (primarykey), `account_number`, `account_date`, `account_type`, `plain_balance`, `cipher_balance`, `decipher_balance`, `digital_signature`, and `signature_verification`. In chapter nine, you will create a `Client Data` table, which has the following seven fields: `client_data_id` (primary key), `account_id` (primary_key), `birth_date`, `address`, `mother_name`, `telephone`, and `photo_path`. In chapter ten, you will be taught how to extract image features, utilizing `BufferedImage` class, in Java GUI. In chapter eleven, you will be taught how to create Crime database and its tables. In chapter twelve, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother_name`, `address`, `telephone`, and `photo`. In chapter thirteen, you will be taught to create Java GUI to view, edit, insert, and delete `Feature_Extraction` table data. This table has eight columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter fourteen, you will add two tables: `Police_Station` and `Investigator`. These two tables will later be joined to `Suspect` table through another table, `File_Case`, which will be built in the seventh chapter. The `Police_Station` has six columns: `police_station_id` (primary key), `location`, `city`, `province`, `telephone`, and `photo`. The `Investigator` has eight columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter fifteen, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key), `police_station_id` (foreign key), `investigator_id` (foreign key), `victim_id` (foreign key), `status`, and `description`. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

Learning MariaDB Feb 24 2022 In this book, you will learn how to build from scratch a criminal records management database system using MariaDB Connector/J. As you know, MariaDB server is a community developed fork of MySQL server. Started by core members of the original MySQL team, MariaDB actively works with outside developers to deliver the most featureful, stable, and sanely licensed open SQL server in the industry. In the first chapter, you will be taught how to create Crime database and its tables. In the second chapter, you will create `Suspect` table. You will be taught to create Java GUI to view, edit, insert, and delete `Suspect` table data. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother name`, `address`, `telephone`, and `photo`. In the third chapter, you will be taught to create Java GUI to view, edit, insert, and delete `Feature_Extraction` table data. This table has eight columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In the fourth chapter, you will add two tables: `Police_Station` and `Investigator`. These two tables will later be joined to `Suspect` table through another table, `File_Case`, which will be built in the seventh chapter. The `Police_Station` has six columns: `police_station_id` (primary key), `location`, `city`, `province`, `telephone`, and `photo`. The `Investigator` has eight columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In the fifth chapter, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key), `police_station_id` (foreign key), `investigator_id` (foreign key), `victim_id` (foreign key), `status`, and `description`. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

LEARN JDBC THE HARD WAY: A Hands-On Reference to MySQL and SQL Server Driven Programming Feb 12 2021 This hands-on tutorial/reference/guide to MySQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from MySQL and SQL Server. As you would expect, this book shows how to build from scratch two different databases: MySQL and SQL Server using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: `Teacher` table, `TClass` table, and `Subject` table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the `Student` table, the `Parent` table, and `Tuition` table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create Crime database and its tables. In chapter six, you will be taught how to extract image features, utilizing `BufferedImage` class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete `Suspect` table data. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother_name`, `address`, `telephone`, and `photo`. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete `Feature_Extraction` table data. This table has eight columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. In chapter nine, you will add two tables: `Police_Station` and `Investigator`. These two tables will later be joined to `Suspect` table through another table, `File_Case`, which will be built in the seventh chapter. The `Police_Station` has six columns: `police_station_id` (primary key), `location`, `city`, `province`, `telephone`, and `photo`. The `Investigator` has eight columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key),

police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MySQL/SQL SERVER programmer.

POSTGRESQL FOR JAVA GUI: Database, Cryptography, and Image Processing Jul 08 2020 In this book, you will learn how to build from scratch a criminal records management database system using Java/PostgreSQL. All Java code for cryptography and digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. There are only three external libraries used in this book: Connector / J to facilitate Java to PostgreSQL connections, JCalendar to display calendar controls, and JFreeChart to display graphics. Digital image techniques to extract image features used in this book are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In the second chapter, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. In the third chapter, you will learn how to create and store salt passwords and verify them. You will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In the fourth chapter, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In the fifth chapter, you create a table with the name of the Account, which has ten columns: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In the sixth chapter, you will create a Client_Data table, which has the following seven fields: client_data_id (primary key), account_id (primary_key), birth_date, address, mother_name, telephone, and photo_path. In the seventh chapter, you will be taught how to create Crime database and its tables. In eighth chapter, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In the ninth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In the tenth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In the eleventh chapter, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In the twelfth chapter, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful for you.

Database and Image Processing Using Java GUI and Microsoft Access Jan 02 2020 The book details how programmers and database professionals can develop Access-based Java GUI applications that involves database and image processing. This book will help you quickly write efficient, high-quality access-database-driven code with Java. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch two access database management systems using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will create School database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create Crime database and its tables. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key),

suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eighth, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police and Investigator. These two tables will later be joined to Suspect table through another table, Case_File, which will be built in the seventh chapter. The Police has six columns: police_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and Case_File. The Case_File table will connect four other tables: Suspect, Police, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

Java In Action: An Excellent Guide to Explore JDBC And Database Applications Oct 30 2019 This step-by-step guide to explore database programming using Java is ideal for people with little or no programming experience. The goal of this concise book is not just to teach you Java, but to help you think like a programmer. Each brief chapter covers the material for one week of a college course to help you practice what you've learned. As you would expect, this book shows how to build from scratch two different databases: PostgreSQL and SQLite using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In chapter three, you will create a PostgreSQL database, named School, and its tables. In chapter four, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter five, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter six, you will study how to query the six tables. In chapter seven, you will be shown how to create SQLite database and tables with Java. In chapter eight, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. Digital image techniques to extract image features used in this chapter are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter ten, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter eleven, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter twelve, you will add two tables: Victim and Case_File. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/PostgreSQL/SQLite programmer.

MariaDB with Java GUI for Cryptography and Image Processing Nov 11 2020 This book is Java/MariaDB version of our previous books which used Java/MySQL and Java/PostgreSQL. In this book, you will learn how to build from scratch a criminal records management database system and simple bank database system using Java/MariaDB. All Java code for digital image processing in this book is Native Java. Intentionally not to rely on external libraries, so that readers know in detail the process of extracting digital images from scratch in Java. There are only three external libraries used in this book: Connector/J to facilitate Java to MariaDB connections, JCalendar to display calendar controls, and JFreeChart to display graphics. Digital image techniques to extract image features used in this book are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In the first chapter, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys

in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. In the second chapter, you will learn how to create and store salt passwords and verify them. You will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In the third chapter, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In the fourth chapter, You create a table with the name of the Account, which has ten columns: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In the fifth chapter, you will create a Client_Data table, which has the following seven fields: client_data_id (primary key), account_id (primary_key), birth_date, address, mother_name, telephone, and photo path. In the sixth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In the seventh chapter, you will be taught how to create Crime database and its tables. In ninth chapter, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In the eighth chapter, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In the ninth chapter, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In the eleventh chapter, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful for you.

Access Database with JDBC May 18 2021 This is a Java GUI crash course. This book will help you quickly write efficient, high-quality access-database-driven code with Java. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications).The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch two access database management systems using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will create School database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create Crime database and its tables. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eighth, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police and Investigator. These two tables will later be joined to Suspect table through another table, Case_File, which will be built in the seventh chapter. The Police has six columns: police_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and Case_File. The Case_File table will connect four other tables: Suspect, Police, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

The Fast Way to Learn Java GUI with PostgreSQL and SQLite Feb 01 2020 This step-by-step guide to explore database programming using Java is ideal for people with little or no programming experience. The goal of this concise book is not just to teach you Java, but to help you think like a programmer. Each brief chapter covers the material for one week of a college course to help you practice what you've learned. As you would expect, this book shows how to build from scratch two different databases: PostgreSQL and SQLite using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to

create databases, create tables, fill tables, and manipulate table contents is done. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In chapter three, you will create a PostgreSQL database, named School, and its tables. In chapter four, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter five, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter six, you will study how to query the six tables. In chapter seven, you will be shown how to create SQLite database and tables with Java. In chapter eight, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. Digital image techniques to extract image features used in this chapter are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter ten, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter eleven, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter twelve, you will add two tables: Victim and Case_File. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/PostgreSQL/SQLite programmer.

Thinking in Java vol. 2 Tecniche avanzate May 30 2022

JAVA GUI WITH MYSQL Mar 04 2020 In this book, you will learn how to build from scratch a MySQL database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use MySQL in Java. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In the last chapter, you will study how to query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MySQL programmer.

The Quick Way to Learn Java GUI with MariaDB and SQLite Oct 11 2020 This step-by-step guide to explore database programming using Java is ideal for people with little or no programming experience. The goal of this concise book is not just to teach you Java, but to help you think like a programmer. Each brief chapter covers the material for one week of a college course to help you practice what you've learned. As you would expect, this book shows how to build from scratch two different databases: MariaDB and SQLite using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a KeyStore, generate PrivateKey and PublicKey, encrypt / decrypt data, and generate and verify digital prints. In the second chapter, you will learn how to create and store salt passwords and verify them. You will create a Login table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the Login table, in this chapter you will also create a Client table. In the case of the Client table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In the third chapter, you will create an Account table. This account table has the following ten fields: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In the fourth chapter, You create a table with the name of the Account, which has ten columns: account_id (primary key), client_id (primarykey), account_number, account_date, account_type, plain_balance, cipher_balance, decipher_balance, digital_signature, and signature_verification. In the fifth chapter, you will create a

Client_Data table, which has the following seven fields: *client_data_id* (primary key), *account_id* (primary key), *birth_date*, *address*, *mother_name*, *telephone*, and *photo_path*. In chapter six, you will be shown how to create SQLite database and tables with Java. In chapter seven, you will be taught how to extract image features, utilizing *BufferedImage* class, in Java GUI. Digital image techniques to extract image features used in this chapter are grayscale, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based matching. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete *Suspect* table data. This table has eleven columns: *suspect_id* (primary key), *suspect_name*, *birth_date*, *case_date*, *report_date*, *suspect_status*, *arrest_date*, *mother_name*, *address*, *telephone*, and *photo*. In chapter nine, you will be taught to create Java GUI to view, edit, insert, and delete *Feature_Extraction* table data. This table has eight columns: *feature_id* (primary key), *suspect_id* (foreign key), *feature1*, *feature2*, *feature3*, *feature4*, *feature5*, and *feature6*. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter ten, you will add two tables: *Police_Station* and *Investigator*. These two tables will later be joined to *Suspect* table through another table, *File_Case*, which will be built in the seventh chapter. The *Police_Station* has six columns: *police_station_id* (primary key), *location*, *city*, *province*, *telephone*, and *photo*. The *Investigator* has eight columns: *investigator_id* (primary key), *investigator_name*, *rank*, *birth_date*, *gender*, *address*, *telephone*, and *photo*. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: *Victim* and *Case_File*. The *File_Case* table will connect four other tables: *Suspect*, *Police_Station*, *Investigator* and *Victim*. The *Victim* table has nine columns: *victim_id* (primary key), *victim_name*, *crime_type*, *birth_date*, *crime_date*, *gender*, *address*, *telephone*, and *photo*. The *Case_File* has seven columns: *case_file_id* (primary key), *suspect_id* (foreign key), *police_station_id* (foreign key), *investigator_id* (foreign key), *victim_id* (foreign key), *status*, and *description*. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MariaDB/SQLite programmer.

Computational Science - ICCS 2002 Mar 28 2022 Computational Science is the scientific discipline that aims at the development and understanding of new computational methods and techniques to model and simulate complex systems. The area of application includes natural systems - such as biology, environmental and geo-sciences, physics, and chemistry - and synthetic systems such as electronics and financial and economic systems. The discipline is a bridge between 'classical' computer science - logic, complexity, architecture, algorithms - mathematics, and the use of computers in the aforementioned areas. The relevance for society stems from the numerous challenges that exist in the various science and engineering disciplines, which can be tackled by advances made in this field. For instance new models and methods to study environmental issues like the quality of air, water, and soil, and weather and climate predictions through simulations, as well as the simulation-supported development of cars, airplanes, and medical and transport systems etc. Paraphrasing R. Kenway (R.D. Kenway, *Contemporary Physics*. 1994): 'There is an important message to scientists, politicians, and industrialists: in the future science, the best industrial design and manufacture, the greatest medical progress, and the most accurate environmental monitoring and forecasting will be done by countries that most rapidly exploit the full potential of computational science'. Nowadays we have access to high-end computer architectures and a large range of computing environments, mainly as a consequence of the enormous stimulus from the various international programs on advanced computing, e.g.

A Comprehensive Guide to Java GUI Programming with SQLite Jun 06 2020 The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch a SQLite database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQLite in Java. In chapter one, you will learn: How to create SQLite database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: *Teacher* table, *TClass* table, and *Subject* table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the *Student* table, the *Parent* table, and *Tuition* table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six tables. In chapter four, you will study how to query the six tables. In chapter five, you will create *Bank* database and its four tables. In chapter six, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a *KeyStore*, generate *PrivateKey* and *PublicKey*, encrypt / decrypt data, and generate and verify digital prints. In chapter seven, you will learn how to create and store salt passwords and verify them. You will create a *Login* table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the *Login* table, in this chapter you will also create a *Client* table. In the case of the *Client* table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter eight, you will create an *Account* table. This account table has the following ten fields: *account_id* (primary key), *client_id* (primary key), *account_number*, *account_date*, *account_type*, *plain_balance*, *cipher_balance*, *decipher_balance*, *digital_signature*, and *signature_verification*. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter nine, you will create a *Client_Data* table, which has the following seven fields: *client_data_id* (primary key), *account_id* (primary key), *birth_date*, *address*, *mother_name*, *telephone*, and *photo_path*. In chapter ten, you will create *Crime* database and its six tables. In chapter eleven, you will be taught how to extract image features, utilizing *BufferedImage* class, in Java GUI. In chapter twelve, you will be taught to create Java GUI to view, edit, insert, and delete *Suspect* table data. This table has eleven columns: *suspect_id* (primary key), *suspect_name*, *birth_date*, *case_date*, *report_date*, *suspect_status*, *arrest_date*, *mother_name*, *address*, *telephone*, and *photo*. In chapter thirteen, you will be taught to create Java GUI to view, edit, insert, and delete *Feature_Extraction* table data. This table has eight

columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter fourteen, you will add two tables: `Police_Station` and `Investigator`. These two tables will later be joined to `Suspect` table through another table, `File_Case`, which will be built in the seventh chapter. The `Police_Station` has six columns: `police_station_id` (primary key), `location`, `city`, `province`, `telephone`, and `photo`. The `Investigator` has eight columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter fifteen, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key), `police_station_id` (foreign key), `investigator_id` (foreign key), `victim_id` (foreign key), `status`, and `description`. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

Learn Java in One Week Jan 26 2022 This book is the ultimate beginners' crash course to Java GUI programming, as it will help you learn enough about the language in as little as 1 week. Complex concepts in developing MS Access and SQLite driven projects are broken down into easy steps to ensure that you can easily master the Java language even if you have never coded before. The best way to learn Java is by doing it. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch two access database management systems using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will create School database and six tables. In chapter two, you will study: Creating the initial three table projects in the school database: `Teacher` table, `TClass` table, and `Subject` table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the `Student` table, the `Parent` table, and `Tuition` table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will create Crime database and its six tables. In chapter six, you will be taught how to extract image features, utilizing `BufferedImage` class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete `Suspect` table data. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother_name`, `address`, `telephone`, and `photo`. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete `Feature_Extraction` table data. This table has eight columns: `feature_id` (primary key), `suspect_id` (foreign key), `feature1`, `feature2`, `feature3`, `feature4`, `feature5`, and `feature6`. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter nine, you will add two tables: `Police_Station` and `Investigator`. These two tables will later be joined to `Suspect` table through another table, `File_Case`, which will be built in the seventh chapter. The `Police_Station` has six columns: `police_station_id` (primary key), `location`, `city`, `province`, `telephone`, and `photo`. The `Investigator` has eight columns: `investigator_id` (primary key), `investigator_name`, `rank`, `birth_date`, `gender`, `address`, `telephone`, and `photo`. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: `Victim` and `File_Case`. The `File_Case` table will connect four other tables: `Suspect`, `Police_Station`, `Investigator` and `Victim`. The `Victim` table has nine columns: `victim_id` (primary key), `victim_name`, `crime_type`, `birth_date`, `crime_date`, `gender`, `address`, `telephone`, and `photo`. The `File_Case` has seven columns: `file_case_id` (primary key), `suspect_id` (foreign key), `police_station_id` (foreign key), `investigator_id` (foreign key), `victim_id` (foreign key), `status`, and `description`.

Thinking in Java vol. 1 Fondamenti Jun 30 2022

The Best Tutorial to Learn Database Programming with Java GUI, MariaDB, and SQL Server Jan 14 2021 This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MariaDB and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MariaDB and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from MariaDB and SQL Server. As you would expect, this book shows how to build from scratch two different databases: MariaDB and SQL Server using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In chapter one, you will learn the basics of cryptography using Java. Here, you will learn how to write a Java program to count Hash, MAC (Message Authentication Code), store keys in a `KeyStore`, generate `PrivateKey` and `PublicKey`, encrypt / decrypt data, and generate and verify digital prints. You will also learn how to create and store salt passwords and verify them. In chapter two, you will create a PostgreSQL database, named `Bank`, and its tables. In chapter three, you will create a `Login` table. In this case, you will see how to create a Java GUI using NetBeans to implement it. In addition to the `Login` table, in this chapter you will also create a `Client` table. In the case of the `Client` table, you will learn how to generate and save public and private keys into a database. You will also learn how to encrypt / decrypt data and save the results into a database. In chapter four, you will create an `Account` table. This account table has the following ten fields: `account_id` (primary key), `client_id` (primarykey), `account_number`, `account_date`, `account_type`, `plain_balance`, `cipher_balance`, `decipher_balance`, `digital_signature`, and `signature_verification`. In this case, you will learn how to implement generating and verifying digital prints and storing the results into a database. In chapter five, you create a table named `Client Data`, which has seven columns: `client_data_id` (primary key), `account_id` (primary_key), `birth_date`, `address`, `mother_name`, `telephone`, and `photo_path`. In chapter six, you will be taught how to create a SQL Server database, named `Crime`, and its tables. In chapter seven, you will be taught how to extract image features, utilizing `BufferedImage` class, in Java GUI. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete `Suspect` table data. This table has eleven columns: `suspect_id` (primary key), `suspect_name`, `birth_date`, `case_date`, `report_date`, `suspect_status`, `arrest_date`, `mother_name`, `address`, `telephone`, and `photo`. In chapter nine, you will be taught

to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter ten, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter eleven, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MariaDB/SQL Server programmer.