Journal of Science and Development Economics
Nam Can Tho University

Website: jsde.nctu.edu.vn

Mobile database exercise solving toolkit

Le Duc Thang[1*], Nguyen Van Linh[1]

[1]*Faculty of Information Technology, Nam Can Tho University, Can Tho City, Vietnam*

[*]*Corresponding author: Le Duc Thang (email: ldthang@nctu.edu.vn)*

Received: 10/1/2025
Revised: 3/2/2025
Accepted: 25/2/2025

Keywords: *database, data model, functional dependency, foreign key, primary key, relationship schema*

Từ khóa: *cơ sở dữ liệu, khóa chính, khóa ngoại, lược đồ quan hệ, mô hình dữ liệu, phụ thuộc hàm*

ABSTRACT

This paper presents the process of building and developing a mobile application designed to assist students and users in learning and applying Database management techniques. The goal of the application is to provide a user-friendly tool for solving exercises related to functional dependencies and keys, as well as normalizing relational schemas from basic to advanced levels. The application was developed using general principles of software development, with key features including an intuitive user interface, object classes, and processing procedures. The main research methods used were surveying user needs, designing functional interfaces, and testing the application with the target user group. The results of the research demonstrate that the application not only helps users improve their database management skills, but also reduces learning time through systematic exercises. The conclusions drawn from the study highlight the effectiveness of the application in providing an efficient self-learning environment, and suggest potential areas for further research to enhance the features and user experience.

TÓM TẮT

Bài báo này trình bày quá trình xây dựng phát triển một ứng dụng di động dành cho việc giải quyết bài tập liên quan đến cơ sở dữ liệu nhằm hỗ trợ sinh viên và người dùng trong việc học tập và áp dụng các kỹ thuật quản lý cơ sở dữ liệu. Mục tiêu của ứng dụng là cung cấp một công cụ thân thiện với người dùng để hỗ trợ giải các bài tập về phụ thuộc hàm và khóa, chuẩn hóa lược đồ quan hệ từ cơ bản đến nâng cao. Ứng dụng được xây dựng dựa trên các nguyên lý chung của phát triển phần mềm, với các tính năng chính bao gồm giao diện

người dùng trực quan, các lớp đối tượng và các thủ tục xử lý. Phương pháp nghiên cứu chính bao gồm khảo sát nhu cầu người dùng, thiết kế giao diện chức năng thử nghiệm ứng dụng với nhóm người dùng mục tiêu. Kết quả nghiên cứu cho thấy ứng dụng không chỉ giúp người dùng khả năng cải thiện kỹ năng quản lý cơ sở dữ liệu mà còn giảm thiểu thời gian học tập nhờ vào các bài tập có tính hệ thống. Kết luận rút ra từ nghiên cứu nhấn mạnh sự hiệu quả của ứng dụng trong việc cung cấp một môi trường tự học hiệu quả, đồng thời gợi ý các hướng nghiên cứu tiếp theo để mở rộng tính năng và trải nghiệm người dùng.

1. INTRODUCTION

In the current digital age, databases have become a core foundation for information management and exploitation. Working with databases is not only a basic requirement for IT professionals but also an essential factor in many fields such as business management, scientific research, and data analysis. Therefore, enhancing understanding and practice in databases is necessary for both learners and professionals. However, learning and solving database exercises often present challenges due to their complex and diverse nature. Traditional learning methods often lack interactivity and do not provide timely feedback, limiting the learner's ability to improve their exercise-solving skills. To address this issue, developing mobile-based tools for solving exercises has become increasingly important.

This paper presents the development of a database exercise-solving application aimed at providing an effective learning tool for users. The application is designed to assist learners in solving both basic and comprehensive exercises through detailed usage guides, error notifications, feedback on solution steps, and results. The paper also describes the application's development process, the technical methods used, as well as challenges and solutions encountered during implementation.

This paper not only contributes to enhancing learning efficiency but also opens new directions in the design of learning tools in the database field. By combining interactive learning principles with advanced software technology, this application is expected to significantly improve the learning experience and outcomes for users in the field of databases.

2. RESEARCH METHODS

2.1 Materials

2.1.1 Documentation

The sources of documentation used in the project include books, lecture materials, seminar reports, and online resources (Hung and Minh, 2016; Khoi, et al., 2022) [1],[2].

2.1.2 Python programming language

Python has powerful high-level data structures and a simple yet effective approach to object-oriented programming. Python’s syntax is a major advantage due to its clarity, ease of understanding, and flexibility, making it an ideal language for scripting and application development across various fields and platforms.

2.1.3 Framework Kivy

Kivy is a Python library that supports cross-platform development. This means that a single codebase can be used to create applications for Android, and iOS, as well as Windows, Linux, and macOS.

2.1.4 NoxPlayer Emulator

NoxPlayer is an Android emulator designed to simulate the interface and functionality of the Android operating system on PC and macOS screens. Emerging alongside the rise of many engaging mobile games, NoxPlayer quickly grew in the mobile gaming and app market on computers. With continuous improvements, NoxPlayer has become a widely favored Android emulator among gamers and coders.

2.1.5 Colab Platform

Colab, or Google Colaboratory, is a cloud-based platform that allows users to write and execute Python code, primarily serving machine learning and data analysis. It offers an interactive programming environment with easy sharing capabilities, integrates libraries such as TensorFlow, and supports free GPU usage. Colab helps users learn and develop projects without the need for complex local environment setups.

2.2 Methods

2.2.1 Requirement analysis

Requirement analysis is conducted to determine necessary functionalities using the following methods:

- Survey and interview: Surveys and interviews are conducted with instructors and students to understand the users' needs and desires. Questions focus on necessary features such as exercise types, feedback methods, and learning support features.

- Document Analysis: Study existing teaching materials to determine functional and non-functional requirements, while also surveying the best standards and methods in the field of database teaching and learning.

2.2.2 System design

Based on the gathered requirements, the system design method consists of the following main components:

- System architecture: The application is designed following a three-tier model, consisting of the user interface, logic processing layer, and data storage layer (Loc, 2014; Thang, 2023) [3],[4].

- Data design: The relational schema is defined as s=(Ω,f), where Ω is a set of attributes and f is a set of functional dependencies declared using the data structures String and List, stored as a File data structure in the Python programming language.

- User interface: The user interface is designed to be user-friendly and easy to use, ensuring that users can effectively interact with the application. UX/UI design principles and interface design tools are employed.

2.2.3 Application development

The application development process includes:

- Programming: Python is used for both frontend and backend programming, along with support packages for designing widgets.

- System Integration: The system's components, including the user interface, logic processing, and data structure, are integrated to ensure synchronous and accurate operation.

- Testing: Testing methods such as unit testing, integration testing, and system testing are conducted to ensure the application functions correctly. In the future, automated testing tools

may be utilized.

2.2.4 Application evaluation

The following methods are employed to evaluate the application:

- User testing: A group of students and teachers is invited to test the application in a real learning environment. Feedback on the application's usefulness, ease of use, and effectiveness in solving database exercises is collected.

- Data analysis: Feedback from users is analyzed using both quantitative and qualitative methods, including statistical analysis and content analysis from interviews and surveys.

- Adjustments and improvements: Based on the evaluation results, necessary adjustments and improvements are made to optimize the application, enhancing user experience and learning effectiveness.

3. RESULTS AND DISCUSSION

3.1 Results

3.1.1 Algorithm installation

With the relational schema s=(Ω,f), where Ω is a set of attributes and f represents functional dependencies, the following solving algorithms have been successfully implemented:

– Finding the closure of a set of attributes

– Determining if a functional dependency is derivable from a set of functional dependencies

– Identifying redundant functional dependencies

– Identifying and removing redundant attributes on the left side of a functional dependency

– Finding the minimal cover of a set of functional dependencies

- Identifying a specific key of a relational schema

- Identifying all keys of a relational schema

- Determining if a relational schema satisfies 3NF (Third Normal Form)

- Determining if a relational schema satisfies BCNF (Boyce-Codd Normal Form)

- Decomposing a relational schema into projection schemas satisfying 3NF

- Decomposing a relational schema into projection schemas satisfying BCNF

3.1.2 Key algorithms

Consider the relational schema $s = (\Omega, f)$, where $\Omega = A_1 A_2 \dots A_n$ is the set of attributes, and f is the set of functional dependencies in the form $f = \{L_i \to R_i / L_i, R_i \subseteq \Omega\}$. is a non-empty subset of Ω.

- Algorithm for finding the closure $(X)^+$ of X:

(1) Assign $\Sigma = f$

(2) Assign T = **X**

(3) While there exists a functional dependency $L \to R \in \Sigma \ \wedge \ L \subseteq T \ \wedge \ R \notin T$ do:

a. T = T∪R

b. $\Sigma = \Sigma \backslash \{L \to R\}$

(4) Conclude $(X)^+ = T$

- Algorithm to determine all the keys of a relational schema:

(1) Set:

$L = \bigcup_{L_i \to R_i \in f} L_i$: the union of the left-hand sides of all functional dependencies in f.

$R = \bigcup_{L_i \to R_i \in f} R_i$: the union of the right-hand sides of all functional dependencies in f.

$\theta = \Omega \backslash R$

$\lambda = L \cap R$

(2) If $\lambda = \emptyset$ the key of s is θ; otherwise, move to step 3

(3) Find all subsets X of λ

(4) Find all$X \cup \theta$ such that the closure $(X \cup \theta)^+ = \Omega$, meaning $X \cup \theta$ is a superkey..

(5) The set of minimal superkeys is the set of all keys of s.

- Algorithm for decomposing a relational schema into projection schemas satisfying 3NF:

• Step 1: Find the minimal cover f ' of f

• Step 2: Find the decomposition that splits s into projection schemas satisfying 3NF:

(1) Identify Ω_0 as the set of attributes that do not appear in either side of any functional dependency in the minimal cover f '.

If $\Omega_0 \neq \emptyset$ include Ω_0 in the decomposition.

(2) Identify Ω_i for i > 0:

For each group of functional dependencies in f' with the same left-hand side L_i, for example: $L_i \rightarrow A_{i1}$:

$L_i \rightarrow A_{i1}$

$L_i \rightarrow A_{i2}$

.....

$L_i \rightarrow A_{ik}$

thì định nghĩa $\Omega_i = L_i A_{i1} A_{i2} \dots A_{ik}$

(3) The decomposition $\varphi[\Omega_0, \Omega_i \text{ for } i > 0]$ decomposes s=(Ω, f) into a relational schema satisfying 3NF.

• Step 3: Transform the obtained decomposition into a lossless, dependency-preserving decomposition:

(1) Find a key K of the relational schema s=(Ω, f).

(2) Check key K:

i. If there exists a component Ω_i containing K, this is the required decomposition. End the algorithm.

ii. If no component Ω_i contains K, then K ∪ $\varphi[\Omega_0, \Omega_i \text{ for } i > 0]$ is the required decomposition. End the algorithm.

- Algorithm for decomposing a relational schema into projection schemas satisfying BCNF: This algorithm iteratively decomposes a schema into two schemas:

Set i = 1.

While s=(Ω, f) does not satisfy BCNF, do:

a. Select X→A ∈ f that does not satisfy BCNF, meaning:

X is not a key, and

A ∉ X.

b. Set: Ω_i = X ∪ A $f_i = \Pi_{(\Omega_i)}(f)$

In this case, $s_i=(\Omega_i, f_i)$ has X as the key and satisfies BCNF.

c. Set: Ω = Ω \ A f = Π_Ω(f)

d. Increment i.

Result: The required decomposition is $\varphi[\Omega_i, \Omega]$.

3.1.3 User Interface Implementation

To ensure convenience, the following interfaces of the toolkit have been successfully implemented.

Figure 1. Main interface

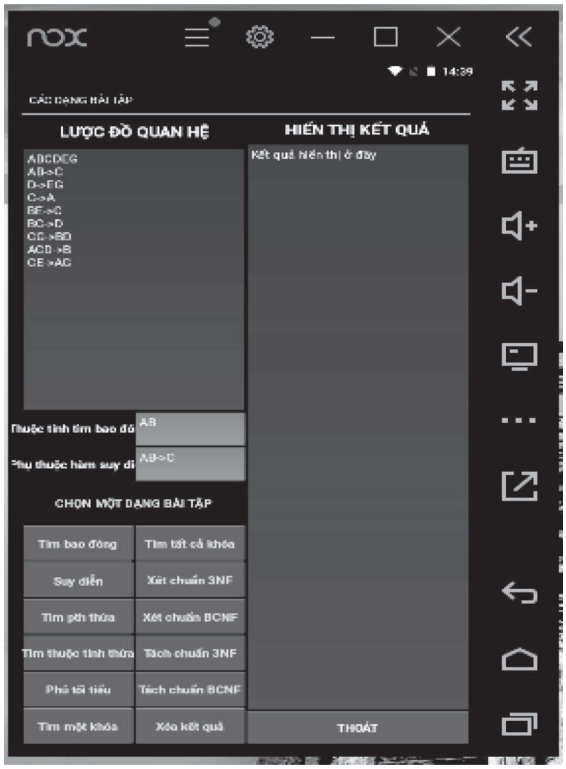

Figure 2. Problem-Solving interface

Figure 3. Relational schema input area

Figure 4. Exercise selection area

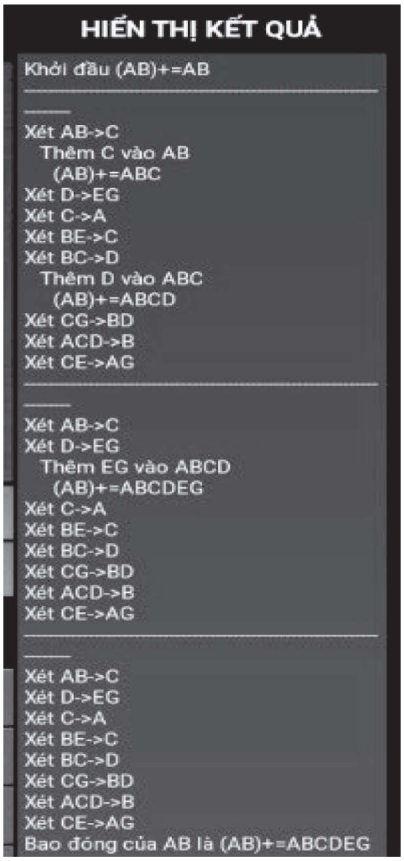

Figure 5. Result display area

3.2 Discussion

3.2.1 Overview of results

This paper focuses on the design and development of a database exercise-solving application aimed at supporting students in learning and mastering database concepts. The results from the deployment and testing of the application indicate that this tool has achieved

significant success in enhancing students' understanding and skills.

3.2.2 Application effectiveness

The application was evaluated based on several criteria, including usability, accuracy in solving exercises, and the level of support it provides to users. Survey results and analysis show that students found the application helpful in making it easier to approach and solve more complex database problems. Improvements in grades and positive feedback from users demonstrate that the application has successfully fulfilled its role in supporting learning.

3.2.3 Limitations of the research

Based on user feedback and suggestions from the project review committee, some areas for improvement were identified. Specifically, several teachers and students recommended the integration of more interactive learning modules or advanced exercises to further strengthen knowledge, as well as a more user-friendly and visually appealing interface. This indicates potential for expanding and updating the application to include additional features, such as explanations of detailed database concepts or automated exercise generation.

Although the application has proven effective in certain areas, it still has some limitations. One of these is the simplicity of the exercise format (it does not scan handwritten exercises), which may affect its generalizability. Additionally, the application has not been extensively tested in various learning environments, so further research is needed to evaluate its feasibility in different scenarios.

To enhance the application's value, future research could focus on integrating Artificial Intelligence and Machine Learning to personalize the learning experience for each user. Incorporating data analysis tools could help identify areas for improvement and provide immediate feedback to students. Furthermore, expanding the application to other fields of study beyond databases could increase its overall usefulness.

Overall, it has been demonstrated that the database exercise-solving application can play an important role in supporting students. Although there are some limitations and areas that need improvement, the current application provides a useful tool for learning and can be developed further to meet the growing demands of teaching and learning in database education.

4. CONCLUSION

This paper has presented the design and development of an application aimed at assisting with the resolution of database exercises. The application not only provides a simple, user-friendly tool but also offers substantial support for students' self-study to enhance their learning process. By applying modern software design principles and advanced technologies, the tool has achieved promising results in effectiveness and interactivity. Testing results indicate that the application allows users to self-assess their exercise-solving abilities by comparing their results and methods with the tool's feedback. Positive user feedback demonstrates that the application effectively meets their learning needs and contributes positively to self-directed learning. However, there are still areas that require improvement, including expanding the range of exercises and enhancing the customization capabilities of the analysis tools. In the future, it may be possible to further expand and update the application to better meet learners'

needs while incorporating the latest research in the field of educational technology to improve the quality and effectiveness of the application. It can be seen that this application is not only a useful tool for learning and teaching databases but also a necessary first step for further research and development in the field of educational technology. It is hoped that future research will continue to expand and optimize the solutions presented in this paper, thereby contributing to the comprehensive development of database learning and teaching methods. The most significant outcome of this project is the successful design and implementation of a toolkit to support solving several basic types of exercises in the database course. The advantage of the product is its lightweight nature, allowing it to be executed on Android mobile devices. Based on the results and analysis from the research on developing the database exercise-solving application, the following content can be further developed in the future: It is recommended to expand the application's features to include a wider variety of exercises, ranging from basic exercises to comprehensive ones. This will help meet the needs of a broad and diverse user base. Integrate analysis tools and direct feedback mechanisms so that users can comment and better understand their mistakes or areas for improvement during the exercise-solving process. Enhance the user interface (UI) and user experience (UX) of the application to ensure ease of use and intuitiveness. The interface should be designed to be simple yet comprehensive enough to include all necessary functions for effective exercise solving. Provide detailed instructions and illustrative examples to help users easily familiarize themselves with how to use the application and its features. Encourage the integration of advanced technologies such as Artificial Intelligence (AI) and Machine Learning to automate the exercise-solving process and provide more accurate feedback. These technologies can help improve the application's performance and support users in understanding and learning from exercises. Propose collaboration with schools and research institutions to gather user feedback data and continuously improve the application based on this feedback. Encourage conducting further studies to evaluate the effectiveness of the application in supporting the learning and teaching process, thereby adjusting and enhancing the application's features.

REFERENCES

[1] Hung, N. B and Minh, D. H. (2016). *Textbook on Programming for Mobile Devices.* Can Tho University Press, 2016.

[2] Khoi, N. H. A., Trang, P. T. X., Trinh, B. T. D., and Khang, T. H. (2022). *Database Textbook for Information Technology Major.* Can Tho University Press, 2022.

[3] Loc, P. T. X. (2014). *Database Lecture. Faculty of Information Technology.* Can Tho University, 2014.

[4] Thang, L. D. (2023). *Database Lecture. Internal circulation.* Nam Can Tho University, 2023.