The IJICS (International Journal of Informatics and Computer Science)
**Vol 10 No 1, Maret 2026, Page 64-69**
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i1.9975

# Development and Evaluation of Cloud-Based Virtual Coding Laboratory for SQL Assessment in Database Learning

**Firman Santosa[1,*], Adyanata Lubis[1], Detri Amelia Chandra[2]**
[1]Department of Computer Science, Universitas Rokania, Riau, Indonesia
[2]Department of Education Technology, Universitas Rokania, Riau, Indonesia
Email: [1,*]firman.snts@gmail.com, [2]adyanata@gmail.com, [3]detriamelia@rokania.ac.id
(* : firman.snts@gmail.com)

**Submitted:22/02/2026; Accepted: 26/03/2026; Published: 30/03/2026**

**Abstract-**This study develops and evaluates a cloud-based Virtual Coding Laboratory to support database learning and SQL practicum activities. The system was designed to address common practicum constraints, including dependency on physical computer laboratories, local database installation, heterogeneous device configurations, delayed assessment, and limited learning activity records. A research and development approach was applied using the ADDIE model, consisting of analysis, design, development, implementation, and evaluation. The product integrates authentication, class management, learning materials, SQL exercises, an online query editor, execution workspace, automatic SQL assessment, feedback, scores, reset database functionality, and submission history. The system was developed using PHP with a separated practice database workspace to isolate student queries from the main application database. Evaluation involved black-box functional testing, expert validation, and practicality testing with two lecturers and 30 students. The functional testing result reached 100%, indicating that all main features worked as expected. Expert validation achieved an overall feasibility score of 90.00%, categorized as very feasible, while practicality testing achieved 87.90%, categorized as very practical. These findings indicate that the proposed Virtual Coding Laboratory is suitable as an integrated, flexible, and browser-based practicum environment for database learning.

**Keywords:** Automatic Assessment; Cloud Computing; Database Learning; SQL Practicum; Virtual Coding Laboratory

## 1. INTRODUCTION

Digital transformation in higher education has changed the way universities design, implement, and evaluate learning. Learning activities are no longer limited to physical classrooms and laboratories because web-based applications, cloud computing, and digital learning environments allow students to access learning resources more flexibly. In computer science education, this transformation is particularly important because students need not only conceptual understanding but also repeated practice in writing, executing, debugging, and improving code [1], [2].

Database courses are core subjects in computer science and information technology programs. The course usually covers relational database concepts, table design, normalization, data integrity, transactions, and Structured Query Language (SQL). Mastery of SQL cannot be achieved only through theoretical explanation; students need to read problems, understand table structures, write queries, execute commands, observe outputs, identify errors, and revise their solutions. Previous studies on automatic judging and programming assessment show that automated practice environments can increase practice intensity and provide faster learning feedback [3], [4].

Conventional database practicum activities often rely on local software installation in computer laboratories or on personal student devices. This model creates several technical barriers, such as different operating systems, software versions, limited device specifications, installation failures, and database configuration conflicts. These problems reduce effective practicum time and make it difficult for students to repeat SQL exercises outside scheduled laboratory sessions. Cloud-based laboratories can reduce such barriers by providing a uniform environment accessible through a web browser [5], [6].

A Virtual Coding Laboratory is a specialized virtual laboratory that provides an online environment for writing, executing, and evaluating code. For database learning, this environment should not only provide SQL editors, but also integrate learning materials, exercises, database workspaces, automatic assessment, feedback, scores, and learning history. Automatic SQL assessment must also be designed carefully because different SQL statements may produce the same correct output. Therefore, assessment mechanisms should consider query results or the final database state, not merely textual similarity [7], [8].

Based on these problems, this study develops and evaluates a cloud-based Virtual Coding Laboratory with automatic SQL assessment for database learning. The contribution of this paper is threefold. First, it presents an integrated browser-based SQL practicum environment for database courses. Second, it describes the system design, architecture, and main modules using the ADDIE development model. Third, it reports functional testing, expert validation, and practicality evaluation to determine the feasibility and usability of the developed product.

## 2. RESEARCH METHODOLOGY

### 2.1 Research Design

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 1, Maret 2026, Page 64-69
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i1.9975

This study used a research and development design to produce and evaluate a cloud-based Virtual Coding Laboratory for database learning. The development process followed the ADDIE model, consisting of Analysis, Design, Development, Implementation, and Evaluation. This model was selected because it provides a systematic procedure from needs analysis to product evaluation [9].

The Analysis stage identified practicum problems, user characteristics, SQL materials, infrastructure conditions, and functional and nonfunctional system requirements. The Design stage produced the system architecture, database structure, user workflow, interface design, SQL exercises, assessment rules, and evaluation instruments. The Development stage implemented the product using PHP and a separated database workspace. The Implementation stage involved lecturers and students as product users, while the Evaluation stage measured functionality, feasibility, and practicality.

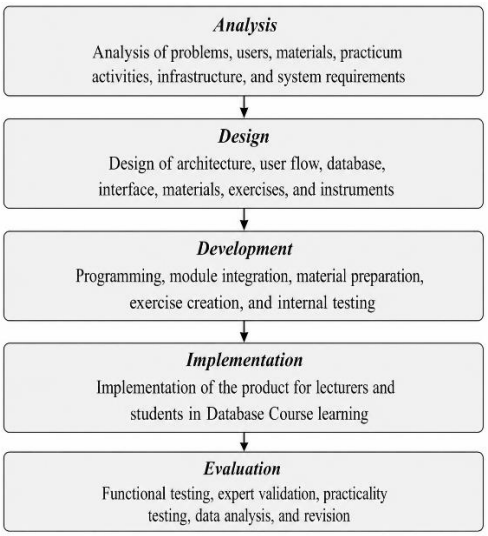


**Fig. 1.** Product development stages using the ADDIE model

### 2.2 Product Architecture and Main Features

The product was designed as a web-based application that can be accessed through a browser. Users are grouped into administrator or lecturer roles and student roles. The administrator or lecturer manages users, classes, materials, exercises, initial datasets, reference queries, assessment rules, and learning reports. Students access classes, read materials, open exercises, write SQL queries, execute commands, submit answers, receive scores and feedback, reset the workspace, and view submission history.

The system architecture separates the main application database from the SQL practice workspace. This separation is important because student queries are executed in an isolated workspace and do not directly modify application data such as users, classes, materials, and scores. The workspace can be reset to the initial dataset, allowing students to repeat exercises safely.

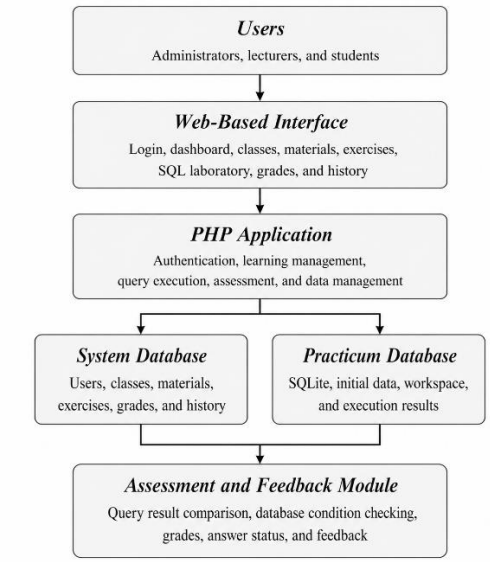


**Fig. 2.** Architecture of the cloud-based Virtual Coding Laboratory

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 1, Maret 2026, Page 64-69
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i1.9975

**Table 1.** Main functions of the developed system

| No | Module | Main Function | User |
|---|---|---|---|
| 1 | Authentication | Verifies accounts and redirects users based on access roles. | All users |
| 2 | Class and material management | Manages database learning materials, classes, and SQL practice content. | Lecturer |
| 3 | SQL workspace | Provides online query editing, execution, output display, and reset database functionality. | Student |
| 4 | Automatic assessment | Compares query outputs or checks final database conditions to produce scores and feedback. | Lecturer and student |
| 5 | History and report | Stores query logs, attempts, status, score, feedback, and submission time. | Lecturer and student |

### 2.3 Participants, Instruments, and Data Analysis

Product evaluation involved three experts and 32 users. The experts consisted of one material expert, one media expert, and one software expert. User practicality testing involved two lecturers and 30 students who used the system in database practicum activities. The participants were selected purposively because they were directly related to database learning or software development evaluation.

The instruments consisted of functional testing sheets, expert validation questionnaires, and practicality questionnaires. Functional testing used black-box testing to evaluate whether each function produced the expected output. Expert validation measured content, media, and software feasibility. Practicality testing measured ease of access, navigation, SQL practice flow, clarity of output, usefulness, and user satisfaction.

Quantitative scores were analyzed using percentage formulas. Functional success was calculated as the number of successful scenarios divided by the total number of testing scenarios. Feasibility and practicality were calculated by dividing the obtained score by the ideal maximum score and multiplying by 100%. The interpretation categories were very feasible or very practical for scores of 86%-100%, feasible or practical for 71%-85%, fairly feasible or fairly practical for 56%-70%, less feasible or less practical for 41%-55%, and not feasible or not practical for scores below 41%.

**Table 2.** Evaluation instruments and respondents

| No | Instrument | Respondent | Main Aspect |
|---|---|---|---|
| 1 | Functional testing sheet | Development team | Login, access rights, class, materials, SQL execution, assessment, reset, history, and logout. |
| 2 | Material validation | Material expert | Content suitability, accuracy, systematics, exercises, feedback, and language. |
| 3 | Media validation | Media expert | Display, navigation, interaction, responsiveness, and usability. |
| 4 | Software validation | Software expert | Functionality, reliability, performance, security, compatibility, and maintainability. |
| 5 | Practicality questionnaire | Lecturers and students | Access, navigation, SQL practicum, output clarity, benefits, and satisfaction. |

## 3. RESULT AND DISCUSSION

### 3.1 Needs Analysis Result

The needs analysis showed that database learning requires a practicum environment that can be accessed through a browser without local database installation. Students need facilities to open classes, read materials, write queries, execute SQL commands, view outputs, submit answers, receive scores, obtain feedback, and review their submission history. Lecturers need facilities to manage classes, learning materials, SQL exercises, initial datasets, reference queries, assessment rules, and student results.

These findings confirm that database practicum should be supported by an integrated system rather than separated learning materials, local software installation, and manual assessment. An integrated platform can reduce configuration problems, improve accessibility, and provide immediate feedback for repeated practice.

### 3.2 Product Implementation

The developed Virtual Coding Laboratory provides a role-based interface. The lecturer side supports class and exercise management, while the student side provides SQL practice activities. The SQL laboratory page contains an editor

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 1, Maret 2026, Page 64-69
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i1.9975

panel, execution output, submission button, score display, feedback message, and history table. This design allows students to test a query before submitting it as an answer and enables them to learn through trial, error, and correction.

Fig. 3 shows an example of the implemented SQL practice interface. The interface demonstrates the integration of instructions, SQL editor, query execution result, feedback, and submission history in one page.

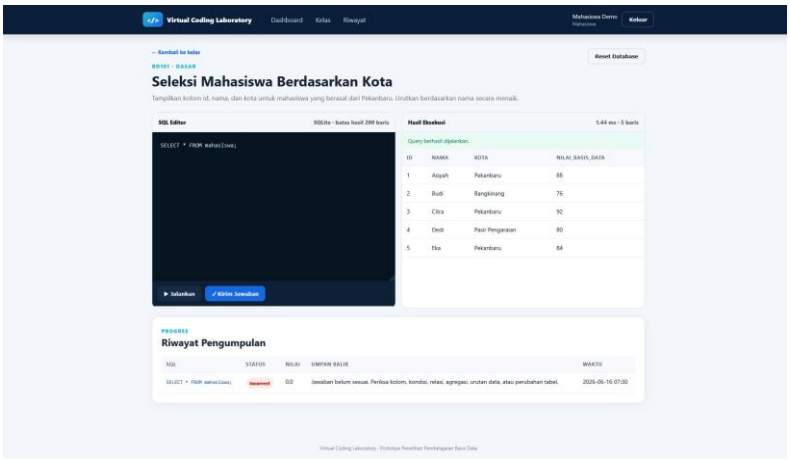


**Fig. 3.** Implementation of the SQL practice interface

### 3.3 Functional Testing Result

Functional testing was conducted using the black-box approach. Ten main scenarios were tested, including login, access rights, dashboards, classes and materials, exercise management, SQL execution, assessment, reset workspace, history, and logout. The result showed that all scenarios were successful, producing a functional success rate of 100%.

**Table 3.** Functional testing summary

| No | Module | Testing Scenario | Status |
|---|---|---|---|
| 1 | Login | Correct and incorrect accounts were tested. | Successful |
| 2 | Access rights | Pages were opened according to user roles. | Successful |
| 3 | Class and materials | Class lists and learning materials were displayed. | Successful |
| 4 | SQL execution | Correct and incorrect queries were executed. | Successful |
| 5 | Assessment | Correct and incorrect answers were submitted. | Successful |
| 6 | Reset workspace | The practice workspace was returned to the initial dataset. | Successful |
| 7 | History and logout | Query history was stored and user sessions ended correctly. | Successful |

### 3.4 Expert Validation Result

Expert validation showed that the product was highly feasible. The material expert score was 91.67%, the media expert score was 86.67%, and the software expert score was 91.67%. The overall feasibility score was 90.00%, which falls into the very feasible category. This result indicates that the product meets content, media, and software quality requirements for database learning.

**Table 4. Expert validation result**

| No | Validator | Score | Maximum | Percentage | Category |
|---|---|---|---|---|---|
| 1 | Material expert | 55 | 60 | 91.67% | Very feasible |
| 2 | Media expert | 52 | 60 | 86.67% | Very feasible |
| 3 | Software expert | 55 | 60 | 91.67% | Very feasible |
| | Total | 162 | 180 | 90.00% | Very feasible |

### 3.5 Practicality Testing Result

Practicality testing was conducted with two lecturers and 30 students. The lecturer practicality score reached 91.43%, while the student practicality score reached 87.67%. The overall practicality score was 87.90%, categorized as very

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 1, Maret 2026, Page 64-69
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i1.9975

practical. These results show that the system is easy to access, supports SQL practice activities, provides useful feedback, and can be used as a supporting facility for database courses.

**Table 5. Practicality testing result**

| No | Respondent Group | Number of Respondents | Percentage | Category |
|---|---|---|---|---|
| 1 | Lecturers | 2 | 91.43% | Very practical |
| 2 | Students | 30 | 87.67% | Very practical |
| | Overall | 32 | 87.90% | Very practical |

### 3.6 Discussion

The findings show that the developed Virtual Coding Laboratory successfully integrates learning materials, SQL exercises, an execution environment, automatic assessment, feedback, scores, and history. This integration is consistent with the characteristics of virtual and remote coding laboratories, which should connect learning objectives, practice activities, execution mechanisms, and evaluation [1], [2].

The separation between the main application database and the practice workspace is an important technical design. It allows students to execute SQL commands safely in a sandboxed environment, while the reset workspace feature returns the dataset to its initial condition. This feature supports repeated practice and reduces the risk of damaging application data.

The automatic assessment mechanism also addresses an essential problem in SQL practicum. Because SQL answers may vary syntactically while producing the same output, the system evaluates query results or final database conditions rather than relying only on query text similarity. This approach is aligned with previous findings that SQL assessment should be designed to avoid false positives and deceptive errors [8].

Although the product achieved high feasibility and practicality scores, several limitations remain. The learning materials and exercises still focus on basic and intermediate SQL topics. The assessment mechanism does not yet evaluate query efficiency, query optimization, or advanced database administration. In addition, practicality testing was conducted in one study program, so broader implementation is required to confirm scalability across different institutions.

## 4. CONCLUSION

This study developed and evaluated a cloud-based Virtual Coding Laboratory with automatic SQL assessment for database learning. The product was developed using the ADDIE model and provides authentication, class and material management, SQL exercises, an online query editor, execution workspace, automatic assessment, feedback, scores, reset database functionality, and submission history. Functional testing showed that all main functions ran successfully with a 100% success rate. Expert validation produced an overall feasibility score of 90.00%, categorized as very feasible, and practicality testing produced an overall score of 87.90%, categorized as very practical. The results indicate that the proposed system can be used as an integrated and flexible practicum facility for database courses. Future work should extend the system to advanced SQL topics, multiple database engines, adaptive feedback, query-efficiency analysis, learning analytics dashboards, and experimental evaluation of student learning outcomes.

## REFERENCES

[1] Y. Aliev, G. Ivanova, and A. Borodzhieva, "Design and research of a virtual laboratory for coding theory," Applied Sciences, vol. 14, no. 20, Art. no. 9546, 2024, doi: 10.3390/app14209546.

[2] M. Garefalakis, Z. Kamarianakis, and S. Panagiotakis, "Towards a supervised remote laboratory platform for teaching microcontroller programming," Information, vol. 15, no. 4, Art. no. 209, 2024, doi: 10.3390/info15040209.

[3] E. Martin-Martin, M. Montenegro, A. Riesco, R. Rubio, and F. Saenz-Perez, "LearnSQL: Impact of an automatic judge in database learning," ACM Transactions on Computing Education, vol. 26, no. 1, Art. no. 3, pp. 1-37, 2026, doi: 10.1145/3769852.

[4] M. Messer, N. C. C. Brown, M. Kolling, and M. Shi, "Automated grading and feedback tools for programming education: A systematic review," ACM Transactions on Computing Education, vol. 24, no. 1, Art. no. 10, pp. 1-43, 2024, doi: 10.1145/3636515.

[5] J. Geetha, D. S. Jayalakshmi, E. Naresh, and N. Sreenivasa, "Lightweight cloud-based solution for digital education and assessment," Science & Technology Libraries, vol. 43, no. 3, pp. 274-286, 2024, doi: 10.1080/0194262X.2023.2247460.

[6] A. Sapriati et al., "The effect of virtual laboratories on improving students' SRL: An umbrella systematic review," Education Sciences, vol. 13, no. 3, Art. no. 222, 2023, doi: 10.3390/educsci13030222.

[7] A. P. Cavalcanti et al., "Automatic feedback in online learning environments: A systematic literature review," Computers and Education: Artificial Intelligence, vol. 2, Art. no. 100027, 2021, doi: 10.1016/j.caeai.2021.100027.

[8] J. Wang, S. Chen, Z. Tang, P. Lin, and Y. Wang, "False positives and deceptive errors in SQL assessment: A large-scale analysis of online judge systems," ACM Transactions on Computing Education, vol. 24, no. 3, Art. no. 30, pp. 1-23, 2024, doi: 10.1145/3654677.

ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i1.9975

[9] N. Zhang and Y. Liu, "Design and implementation of virtual laboratories for higher education sustainability: A case study of Nankai University," Frontiers in Education, vol. 8, Art. no. 1322263, 2024, doi: 10.3389/feduc.2023.1322263.

[10] R. Elmoazen, M. Saqr, M. Khalil, and B. Wasson, "Learning analytics in virtual laboratories: A systematic literature review of empirical research," Smart Learning Environments, vol. 10, Art. no. 23, 2023, doi: 10.1186/s40561-023-00244-y.

[11] I. Georgakopoulos, D. Piromalis, S. Ntanos, V. Zakopoulos, and P. Makrygiannis, "A prediction model for remote lab courses designed upon the principles of education for sustainable development," Sustainability, vol. 15, no. 6, Art. no. 5473, 2023, doi: 10.3390/su15065473.

[12] M. G. Hahn, S. M. B. Navarro, L. de la Fuente Valentin, and D. Burgos, "A systematic review of the effects of automatic scoring and automatic feedback in educational settings," IEEE Access, vol. 9, pp. 108190-108198, 2021, doi: 10.1109/ACCESS.2021.3100890.

[13] Y. Hou et al., "Post-COVID-19 student and faculty perceptions of online computing labs: Better targeted, better perceptions, but still need improvement," Education Sciences, vol. 14, no. 12, Art. no. 1359, 2024, doi: 10.3390/educsci14121359.

[14] A. Sypsas and D. Kalles, "Analysis, evaluation and reusability of virtual laboratory software based on conceptual modeling and conformance checking," Mathematics, vol. 11, no. 9, Art. no. 2153, 2023, doi: 10.3390/math11092153.

[15] B. Fernandez-Gauna, N. Rojo, and M. Grana, "Automatic feedback and assessment of team-coding assignments in a DevOps context," International Journal of Educational Technology in Higher Education, vol. 20, Art. no. 17, 2023, doi: 10.1186/s41239-023-00386-6.