SoftwareX 22 (2023) 101346

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

ChatbotSQL: Conversational agent to support relational database query language learning

Rubén Pérez-Mercado, Antonio Balderas *, Andrés Muñoz, Juan Francisco Cabrera, Manuel Palomo-Duarte, Juan Manuel Dodero

Computer Science Department, University of Cadiz, Spain

ARTICLE INFO

Article history:
Received 26 November 2022
Received in revised form 24 February 2023
Accepted 24 February 2023

Keywords:
Conversational agent
Databases
SQL
Data scientists

ABSTRACT

SQL is a key programming language for data scientists, but its learning is sometimes challenging, especially if the learner does not have a proper computer science background. This article introduces ChatbotSQL, a conversational agent that aims to support the autonomous learning of the SQL language for database querying. ChatbotSQL proposes exercises and guides about the steps to be taken by the learner in case of doubts, providing customised feedback. ChatbotSQL has been created on top of IBM Watson Assistant and is available in a web platform. Students in a higher education database subject used ChatbotSQL extensively, showing a positive perception of its usefulness in supporting the resolution of complex queries.

Code metadata

Current code version	v2.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00390
Code Ocean compute capsule	
Legal Code License	GNU GPL v3
Code versioning system used	Git
Software code languages, tools, and services used	PHP/Laravel, JavaScript, MySQL, IBM Watson Assistant
Compilation requirements, operating environments & dependencies	Apache web server, PHP, Composer, Node.js, npm
If available Link to developer documentation/manual	https://github.com/rubenperezm/ChatbotSQL#readme
Support email for questions	ruben.perezmercado@alum.uca.es

1. Motivation and significance

Learning programming languages can be particularly complex for students, leading some students to become demotivated and disengaged [1]. The literature review has shown that interactive systems such as chatbots can keep students motivated and engaged in their learning by allowing them to study in an exciting and comfortable environment [2]. Conversational agents, also known as chatbots, are computer programmes that are able to hold a chat conversation with a user about a specific topic as if they were a real person [3,4]. Chatbots have shown to have a significant impact on learning performance and teamwork [5]. Indeed, some studies indicate that the use of chatbots in programming could help to improve the students' autonomous work [6], as well as their cognitive load and learning outcomes [7]. This article presents ChatbotSQL, a conversational agent to assist in learning the SQL database programming language.

This research takes place in the scientific field of Learning Technologies. ChatbotSQL aims to support students' difficulty in learning the relational data model [8] and practising it by solving problems requiring complex SQL queries [9]. ChatbotSQL supports students who need to know SQL as a primary subject in computer science (in particular, data scientists, not computer engineers). Specifically, this research takes place in a context where learners

* Corresponding author.
E-mail address: antonio.balderas@uca.es (Antonio Balderas).

https://doi.org/10.1016/j.softx.2023.101346

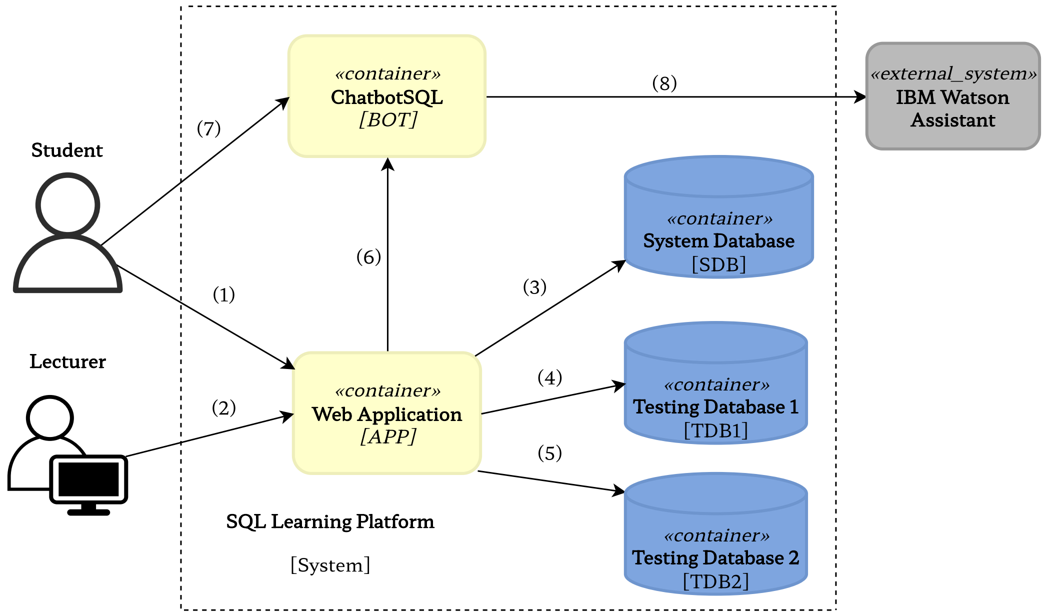

Fig. 1. SQL learning platform architecture.

are driven to do much of their learning autonomously [10]. To be effective, software that supports autonomous learning must be interactive and provide personalised feedback [11].

The first version of ChatbotSQL assisted degree students with single-table queries, obtaining positive evidence of its impact on autonomous learning [12]. However, the students requested that ChatbotSQL also helped them with more complex nested or join-type queries. This article presents the 2.0 version of ChatbotSQL, incorporating support for nested and join queries.

To the best of the authors' knowledge, there is another work using chatbot technology for SQL learning [13]. It combines gamification with a chatbot in which the students have to solve quests in an alternate reality game by means of interactions with the bot using the SQL language. However, the chatbot is limited to 15 SQL queries and the interaction is constrained by the game narrative, whereas our tool allows for more flexible interactions. Other tools that also aim to support SQL language learning such as SQLzoo [14] or SQLeasy [15] do not provide personalised interaction and feedback, being only oriented towards query execution and presentation of results.

The rest of this article is structured as follows. Section 2 describes ChatbotSQL. An example of use is presented in Section 3. The evaluation is shown in Section 4. Section 5 depicts the impact of ChatbotSQL, and conclusions are drawn in Section 6.

2. Software description

The ChatbotSQL client–server web architecture and its main features are introduced in this section.

2.1. Software architecture

ChatbotSQL is developed in Node.js. It connects to a web platform based on the Laravel framework (see *Web Application* in Fig. 1) and to an instance of IBM Watson Assistant to process all the messages received. The numbers in brackets in the following paragraphs refer to the arrows in Fig. 1.

The learning platform uses three MySQL databases: *System Database* (SBD) stores the platform data, i.e., exercises, logs and users. Testing Database 1 (TDB1) and Testing Database 2 (TDB2) are used to run the queries. The platform supports two user roles: *Student* and *Lecturer*. The lecturer can create and modify exercises and, for each exercise, incorporate a solution query and hints for the students (2).

At the beginning of each exercise, ChatbotSQL displays the statement to be solved. If the student attempts to solve the exercise (1), the platform runs both the student's query and the solution query on TDB1 (4). Then, the same process is repeated with TDB2 (5). The reason for checking in two databases is that some incorrect queries can return the same result-set as the solution only for a specific database. As a result of adding a copy of the database with different records, the correctness of the issued query can be double-checked. This way, if the result-set with the issued query and the result-set with the solution query are the same for each database, the student has solved the exercise. Otherwise, the platform gives some feedback to the student.

ChatbotSQL receives the information about the student's interaction (6) and sends it to the IBM Watson Assistant (8). After receiving a response from the Assistant, ChatbotSQL displays it to the user. The student may also directly ask for hints or help (7). The platform stores all the interactions between ChatbotSQL and the student in SDB (3).

It is worth noting that the possible objectives of the user (intents) and some keywords that are useful for the user's purpose (entities) must be defined beforehand in the Assistant. When the Assistant receives a message, it identifies the intent and the entities, and then it uses them to return to the user a meaningful response selected from the Dialog (the set of nodes that model the conversation flow of the Assistant).

2.2. Software functionalities

ChatbotSQL main operation mode (*Exercises mode)* proposes the students exercises defined by the lecturer. But students can also freely practise SQL queries (*Playground mode*).

In the *exercise mode* ChatbotSQL gives customised information to the student at the beginning of an attempt, after proposing a query to solve an exercise and after answering a student's question. In the following paragraphs, the numbers in brackets refer to Fig. 2.

At the beginning of each attempt, the platform retrieves the data about the exercise (1, 2 and 3) and sends it to ChatbotSQL

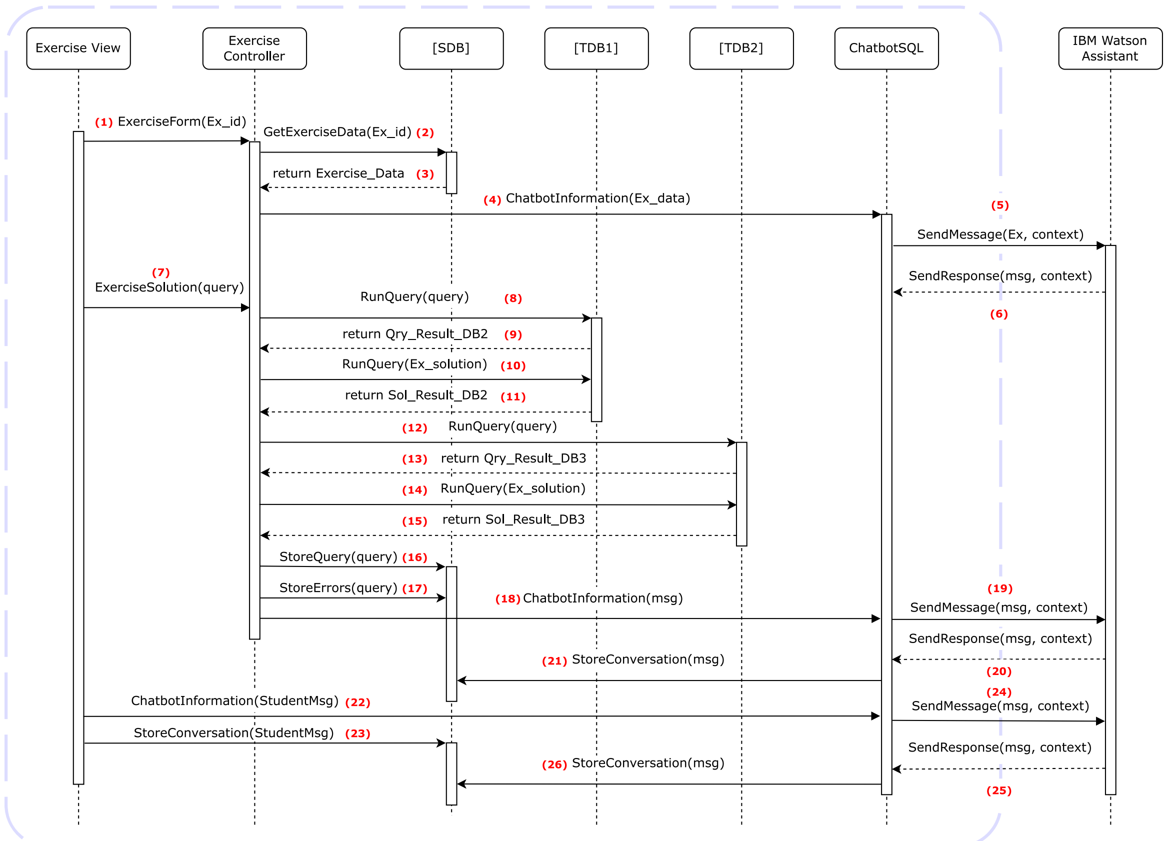

Fig. 2. Sequence diagram of ChatbotSQL.

(4). It sets up the IBM Watson Assistant context with information about the exercise (5) and displays the statement of the exercise.

When the user proposes a query to solve an exercise (7), it is stored (16) and executed (on testing databases), comparing the result-sets to that of the solution (8–15). If the query produces a MySQL error, it is stored (17) and the platform informs ChatbotSQL (18). Similarly, if the query was syntactically correct but it was not a solution for the exercise, the platform identifies differences between them (i.e. compares the fields, tables, clauses, aliases and expressions used in both queries) and proposes a hint for ChatbotSQL to help the student (18).

ChatbotSQL sends the hint message to the Assistant (19), and it decides which message should send back to the user (20) and thus stored (21).

Finally, the student can also directly interact with ChatbotSQL (22). When ChatbotSQL receives a message, it is forwarded to the Assistant (24). It returns the customised answer (a hint for an exercise, information about clauses of functions, etc.), and the student's message and the response are stored (23 and 26).

3. Illustrative examples

The following example shows the interaction of a student solving one of the advanced SQL queries, namely an *outer join*, included in ChatbotSQL 2.0. Specifically, an outer join query is a query that join records from two tables according to a common field in both tables.

3.1. Chatbotsql welcome

When ChatbotSQL indicates the statement to be solved (see Fig. 3, beginning of the conversation in the right column) the student can perform all types of queries. It includes being able to list the available tables using the *"SHOW TABLES"* instruction, as well as using the *"DESCRIBE"* SQL clause to know the structure of the tables.

3.2. Error handling

In Fig. 3, the platform displays a generic error message in the upper area with a red background after the student proposed a wrong solution. The student can see the specific error message returned by the database in the result-set area. Finally, in the conversation with ChatbotSQL, it gives the student a hint: *"Have you used the GROUP BY clause?"*. In fact, the student forgot to perform the GROUP BY aggregation in the proposed query. ChatbotSQL gives such specific feedback by interpreting the error message provided by MySQL.

3.3. Hint management

Now, the student has doubts about how the GROUP BY clause works, types the following message: *"can you give me a hint about the group by clause?"* (Fig. 4, right column). Then, ChatbotSQL answers with the following message: *"You have to group the records, for this you must incorporate the GROUP BY clause, and include all those columns that must have the same value for the records of each group."*.

3.4. Correct completion of the exercise

After receiving the hint, the student refines the query incorporating the GROUP BY clause. Fig. 4 shows how ChatbotSQL informs the student that the query is correct with the message *"You nailed it!"*, whereas the result-set shows the records returned.

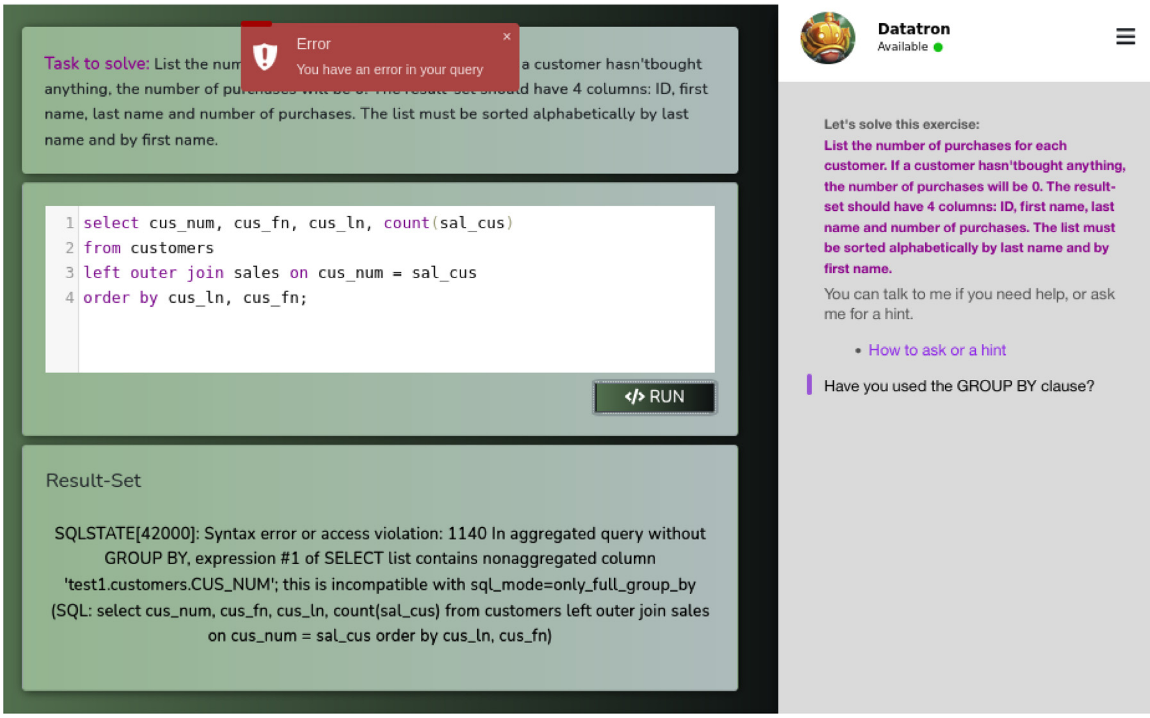

Fig. 3. Response from the platform and chatbot when the student enters a wrong query.

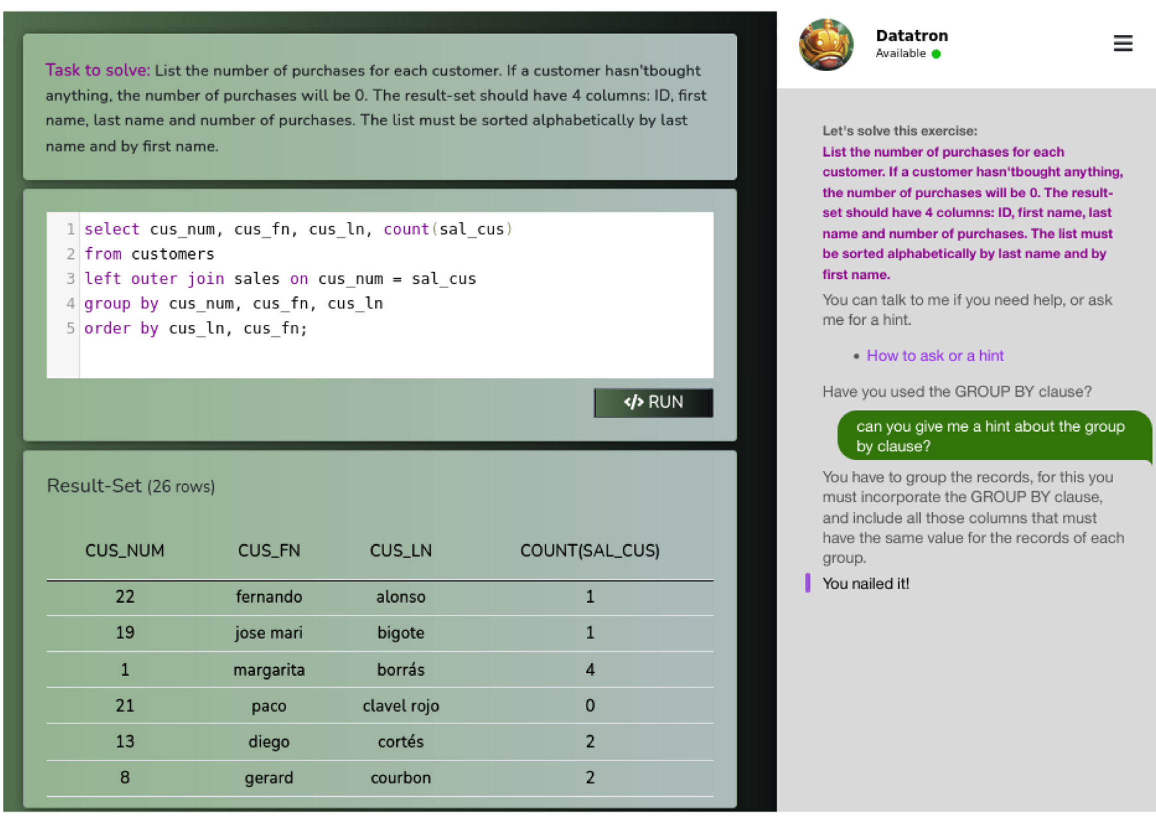

Fig. 4. Screenshot of the platform and chatbot when the student entered the correct query.

4. Evaluation

The evaluation took place at the University of Cadiz (Spain), in a fourth semester compulsory course on Databases for the Computer Engineering degree, during the 2021–22 academic year. The majority of the students were around 20 years, and this was their first database-related course. ChatbotSQL, whose use was completely voluntary, was available from the third week of the course after students were given some basic instructions about its use.

Table 1
Precision, Recall and F1-Scores for the student–chatbot interaction.

	Describe	Show	Select	From	Join	Where/having	Order by	Group by	Other
Precision	100%	100%	98.32%	100%	100%	98.76%	100%	99.01%	83.47%
Recall	50%	93.75%	89.31%	80%	97.06%	98.35%	100%	100%	54.03%
F1-Score	66.67%	96.77%	93.60%	88.89%	98.51%	98.56%	100%	99.50%	65.60%

Table 2
Students' consideration about ChatbotSQL support in solving complex queries.

Query type	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Nested	0.0%	6.9%	27.6%	20.7%	44.8%
Join	0.0%	6.9%	24.1%	31.0%	37.9%

Table 3
Use of ChatbotSQL concerning passing the exam.

Chatbot use	Pass	Fail	Total
Yes	58	6	64
No	28	15	43
Total	86	21	107

The lecturers defined 24 query exercises on the ChatbotSQL platform distributed as follows: 10 beginner, 9 intermediate and 5 advanced level. The following usage data was collected:

- 64 students used ChatbotSQL, resulting in a total of 2,226 attempts to solve query exercises.
- Out of the 2,226 attempts, 1,638 were successful (i.e., the proposed query solved the exercise).
- In trying to solve the query exercises, the 64 students had 9,930 interactions with ChatbotSQL.

4.1. Student–chatbot interaction

An analysis of the keywords contained in the messages showed that in the 77.30% of all messages, the students were asking for a hint. In general, the precision, recall and F1-score for the user intention are higher than 80% for all keywords (see Table 1). The accuracy was 88.10%.

Note that the category "other" in Table 1 comprises the rest of the messages (referring to function names, databases concepts, etc.). If the system responds with a generic message, or the response was not related to the user's concern, it is considered wrong, reducing the recall and F1-score of this category.

4.2. Student perception

At the end of the course, students were asked whether ChatbotSQL had helped them specifically solve nested and join queries (see Table 2). Results were positive in both cases, with 65.5% agreeing and strongly agreeing for nested queries and 68.9% for join queries.

To validate the results of the students' perceptions, we conducted a quasi-experiment based on the SQL exam of the course. Out of the 107 students taking the SQL exams for the course, 86 passed (80.37%) and 21 failed. But if we consider only the results of the students who used ChatbotSQL to prepare for the exam, we get that out of the 64 students who used it, 58 (90.6%) passed while only 6 failed (Table 3).

To correlate these results with the use of ChatbotSQL, let us consider as a null hypothesis (H_0) that the fact that a student has used ChatbotSQL is independent of the fact that he/she has passed the course. On the contrary, the alternative hypothesis (H_1) will be that there is indeed a relation between the student using ChatbotSQL and passing the course.

To validate H_0, the Chi-square test was used, and a p-value of 13.99 was obtained. This value is higher than the significance threshold calculated for 0.001 ($X^2 > 10.82$). Therefore, we cannot accept H_0, and we assume that there is a relation between students having used ChatbotSQL and having passed the course.

Based on these results and the improvement over the previous work [12], we can confirm that the aim of this work has been achieved, i.e. that ChatbotSQL shows evidence of being able to help the student with the most complex queries.

5. Impact

Current research related to Intelligent Tutoring Systems (ITS) points to the use of chatbots as one of the most recommended options, as they offer personalised, human-like tutoring through natural language [5,16]. Other ITS involving observation of students' behaviour (eye-trackers, navigation between applications, etc.) are a more indirect manner of eliciting possible difficulties encountered during a learning process. Therefore, following this current trend in ITS, in this work we decided to approach our tutoring system as a chatbot.

Although chatbots are increasingly popular in education, their applications in supporting programming languages learning is limited. The works found in this area are conversational agents whose knowledge base is limited to a FAQ, which, once it detects a question, returns the associated answer [17]. In our proposal, ChatbotSQL provides students with personalised feedback based on the their proposed solution. ChatbotSQL has shown to be effective in learning SQL language, and invite to further research in this area. On the one hand, ChatbotSQL can be used to improve and expand database learning strategies. SQL is key for data scientists, and learning SQL query techniques requires a notable amount of training and practice, ideally performed through virtual learning platforms [18]. In addition, the learning of SQL needs to consider the ever-expanding database landscape brought about by Big Data, enabling learners to gain a better view of data in a wider range of systems and applications [19]. On the other hand, this research invites to develop and use chatbots for learning other computer programming languages [20,21].

6. Conclusions

This research has developed and evaluated ChatbotSQL, a chatbot that helps students solving complex SQL queries and supports their autonomous learning. ChatbotSQL has been tested with university students, and the analysis of the collected interactions has shown a high accuracy in matching students' hint requests. Additionally, the results of the student evaluation show that they have a positive perception of ChatbotSQL's usefulness, suggesting that it can be a valuable tool for improving SQL learning. Furthermore, the quasi-experiment conducted in this research suggests a possible improvement in student performance, although further research in other contexts is needed to confirm this conclusion.

In future research, we will explore the effectiveness of ChatbotSQL in SQL learning and to determine the extent to which it can contribute to improving student performance in this area. Overall, we consider that ChatbotSQL is a valuable tool to support teaching of other programming languages, being a great potential to further explore the use of chatbots in education and training.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Juan Manuel Dodero reports financial support was provided by Spanish National Research Agency. Andrés Muñoz reports financial support was provided by Regional Ministry of Economy, Knowledge, Business and University of the Andalusian Regional Government.

Data availability

Data will be made available on request.

Acknowledgements

This paper is part of the R&D project CRÊPES (ref. PID2020-115844RB-I00), funded by MCIN/AEI/10.13039/501100011033/. It is also partially granted by the "EMERGIA" programme, funded by the Junta de Andalucía through the grant EMC21_004171.

References

[1] Abdul Talib NH, Ahmad JI. Analysis review on challenges and solutions to computer programming teaching and learning. In: E-proceedings of the 5th International Conference on Computing, Mathematics and Statistics (ICMS2021). Driving Research Towards Excellence. 2021, p. 378–86.

[2] Okonkwo CW, Ade-Ibijola A. Chatbots applications in education: A systematic review. Comput Educ Artif Intell 2021;2:100033, https://doi.org/10.1016/j.caeai.2021.100033.

[3] Huyck CR. Dialogue based interfaces for universal access. Univ Access Inf Soc 2011;10(3):267–74, https://doi.org/10.1007/s10209-010-0209-5.

[4] Abbas T, Khan VJ, Markopoulos P. CoZ: A crowd-powered system for social robotics. SoftwareX 2020;11:100421, https://doi.org/10.1016/j.softx.2020.100421.

[5] Kumar JA. Educational chatbots for project-based learning: investigating learning outcomes for a team-based design course. Int J Educ Technol Higher Educ 2021;18(1):1–28, https://doi.org/10.1186/s41239-021-00302-w.

[6] Verleger M, Pembridge J. A pilot study integrating an AI-driven chatbot in an introductory programming course. In: 2018 IEEE Frontiers in Education Conference. 2018, p. 1–4, https://doi.org/10.1109/FIE.2018.8659282.

[7] Yin J, Goh TT, Yang B, Xiaobin Y. Conversation technology with micro-learning: The impact of chatbot-based learning on students' learning motivation and performance. J Educ Comput Res 2021;59(1):154–77, https://doi.org/10.1177/0735633120952067.

[8] Migler A, Dekhtyar A. Mapping the SQL learning process in introductory database courses. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education. New York, NY, USA: Association for Computing Machinery; 2020, p. 619–25, https://doi.org/10.1145/3328778.3366869.

[9] Ahadi A, Behbood V, Vihavainen A, Prior J, Lister R. Students' syntactic mistakes in writing seven different types of SQL queries and its application to predicting students' success. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. New York, NY, USA: Association for Computing Machinery; 2016, p. 401–6, https://doi.org/10.1145/2839509.2844640.

[10] Sahni J. Does blended learning enhance student engagement? Evidence from higher education. J E-Learn Higher Educ 2019;2019(2019):1–14, https://doi.org/10.5171/2019.121518.

[11] Maliza Muñoz W, Medina León A, Vera Mora G, Castro Molina N. Aprendizaje autónomo en Moodle. J Sci Res 2020;5(CININGEC):632–52.

[12] Balderas A, Baena-Pérez R, Person T, Mota JM, Ruiz-Rube I. Chatbot-based learning platform for SQL training. Int J Interact Multimedia Artif Intell 2022. https://doi.org/10.9781/ijimai.2022.05.003.

[13] Lupano M, Farinetti L, Morreale D. OPUS: an Alternate Reality Game to learn SQL at university. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference. 2021, p. 121–6, https://doi.org/10.1109/COMPSAC51774.2021.00028.

[14] Cumming Andrew. sqlcourse.com Interactive Online SQL Training. 2009, URL https://sqlzoo.net/. [Accessed 26 October 2022].

[15] Easy SQL. Easy interactive online SQL training for beginners. 2009, URL https://www.sql-easy.com/. [Accessed 26 October 2022].

[16] Ji S, Yuan T. Conversational intelligent tutoring systems for online learning: What do students and tutors say? In: 2022 IEEE Global Engineering Education Conference. 2022, p. 292–8, https://doi.org/10.1109/EDUCON52537.2022.9766567.

[17] Sandoval ZV. Design and implementation of a chatbot in online higher education settings.. Issues Inf Syst 2018;19(4):44–52.

[18] Röhm U, Brent L, Dawborn T, Jeffries B. SQL for data scientists: Designing SQL tutorials for scalable online teaching. Proc VLDB Endow 2020;13(12):2989–92, https://doi.org/10.14778/3415478.3415526.

[19] Silva YN, Almeida I, Queiroz M. SQL: From traditional databases to big data. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. 2016, p. 413–8, https://doi.org/10.1145/2839509.2844560.

[20] Yeves-Martínez P, Pérez-Marín D. Prof. Watson: A pedagogic conversational agent to teach programming in primary education. Multidiscipl Digit Publish Inst Proc 2019;31(1). https://doi.org/10.3390/proceedings2019031084.

[21] Daud Siti Hawa Mad, Teo Noor Hasimah Ibrahim, Zain Nurul Hidayah Mat. Ejava chatbot for learning programming language: A post-pandemic alternative virtual tutor. Int J Emerg Trends Eng Res 2020;8(7):3290–8, https://doi.org/10.30534/ijeter/2020/67872020.