Gamification and Flipped Classroom in SQL Education

BALLA Tamás, KIRÁLY Sándor

Abstract. Developing algorithmic thinking is a key goal of secondary education; however, the declarative nature of SQL requires a problem-solving approach that is significantly different from imperative programming languages. This article examines the impact of combining a gamified learning platform and flipped classroom pedagogy on student learning outcomes in introductory SQL instructionThe study involved 70 secondary school students aged 17–18 who were divided into an experimental and a control group. The experimental group learned SQL at home using video-based materials and prepared content, and then participated in interactive, game-based problem-solving activities during class sessions, while the control group followed a traditional, teacher-led approach. Student performance was assessed after four and eight weeks, and the results were analysed using nonparametric statistical methods. The results show that the combined use of gamification and a flipped classroom model significantly improves student engagement, motivation, and learning outcomes in SQL-based problem solving. This study provides empirical evidence on the effectiveness of integrating gamified learning environments and flipped classroom pedagogy in secondary database education.

Keywords: educational games, flipped classroom, SQL, gamification

1. Introduction

Learning SQL (Structured Query Language) is widely considered a challenging and time-consuming task for beginning learners, especially in secondary education. Several studies have shown that students struggle not only with SQL syntax but also, more importantly, with the semantic and logical aspects of query formulation, such as joins, nested queries, and aggregations, which often leads to frustration or loss of motivation [1,2]. Unlike imperative programming languages, SQL is declarative, meaning that students must specify what result they want, rather than how to calculate it, which requires students to take a fundamentally different approach to problem-solving.

Although open-source database management systems (e.g., MySQL, PostgreSQL) allow for independent practice, their feedback mechanisms are typically limited to basic syntactic validation. Logical and semantic errors are often not handled by these environments, which can lead to the reinforcement of incorrect mental models and ineffective problem-solving strategies among beginners [3,4]. This limitation is particularly problematic in secondary education, where students typically have no prior programming experience and need more explicit guidance and formative feedback to develop correct conceptual understanding.

A variety of resources and tools have been developed to support SQL learning, including textbooks, online tutorials, and educational games [5]. Interactive learning environments such as SQL Island or SQL Murder Mystery offer engaging, problem-based experiences for learning SQL. However, these tools are not always suitable for structured classroom use in high school. Similarly, popular web-based platforms such as Codecademy or W3Schools offer step-by-step SQL tutorials, but they lack pedagogical features that support educational models such as the flipped classroom, such as teacher monitoring, differentiated instruction, and collaborative learning opportunities.

The flipped classroom (FC) model reorganizes traditional education by shifting the initial delivery of content to students' independent learning time, while classroom sessions are dedicated to active learning activities such as problem solving, discussion, and feedback. Several studies have reported the positive effects of this approach, including increased student engagement, deeper learning, and improved performance in computer science and programming education [6,7,8,9]. By allowing students to learn at their own pace and allowing teachers to focus on facilitation and individual support during class, the flipped classroom model can be particularly effective in cognitively demanding subjects. Indeed, recent research suggests that flipped learning environments better meet students' needs for autonomy and competence compared to traditional classrooms, contributing to higher motivation and independent learning.

Building on these findings, this study examines the integration of a gamified learning platform and a flipped classroom methodology in secondary SQL education. Gamification – the incorporation of game design elements (points, challenges, narrative, etc.) into non-gaming contexts to enhance engagement [10] – has shown promise in motivating students and enhancing learning in a variety of fields, including computer programming [11]. However, there is limited empirical evidence on the combined use of gamification and flipped classroom strategies in secondary education. This study aims to fill this gap by examining whether a gamified SQL learning platform, when used within a flipped classroom framework, can improve student outcomes compared to traditional instruction. The platform used in the research was specifically designed for educational use and includes interactive lessons, automated SQL query evaluation, and educational mini-games based on familiar fictional datasets. Over a year and a half, more than 1,200 users have registered on the platform and used it to learn SQL in both formal and informal learning contexts.

The primary objective of the research is to determine whether such a platform can be effectively integrated into classroom instruction using the flipped classroom model, and whether this combined approach leads to better learning outcomes compared to traditional teaching methods. Accordingly, the study seeks to answer the following research questions (RQs):

- RQ1: What are the main difficulties that students face when learning SQL?
- RQ2: Can a gamified educational platform be effectively applied in secondary education within a flipped classroom framework to improve learning outcomes?
- RQ3: What are the key design elements of a SQL learning platform that supports the flipped classroom model?

Based on previous research on flipped classroom pedagogy and gamified learning [7,12,13], we hypothesized that students participating in an experimental, technology-supported learning environment would perform better than those receiving traditional instruction at both measurement points (after 4 and 8 weeks of learning). In the following sections, we first outline the theoretical background of our study, then describe the research design and methods, present the results, and discuss the findings in light of the existing literature on motivation, gamification, and flipped learning.

2. Theoretical background

2.1. Student motivation and self-determination theory

Student motivation is a key factor in learning programming and database-related topics. The challenge and complexity of SQL can undermine the intrinsic motivation of novice students, which can lead to learning alienation if not addressed pedagogically [1]. Self-determination theory (SDT) provides a useful framework for understanding how instructional strategies can support motivation in such situations. According to SDT, individuals have three basic psychological needs: autonomy, competence, and relatedness – and meeting these needs enhances intrinsic motivation and engagement [14]. In an educational setting, autonomy refers to students' sense of control over their learning process; competence refers to feelings of efficacy and mastery of knowledge; and relatedness refers to feelings of connectedness to others (peers and instructors) in the learning environment. When these needs are supported, students are more likely to engage in learning for the intrinsic satisfaction rather than for extrinsic rewards, which can lead to deeper learning and persistence.

Both the flipped classroom approach and gamified learning interventions can be analysed from a self-knowledge development (SDT) perspective, as they can satisfy these three needs [15,16]. The flipped classroom inherently provides greater autonomy to students: students can acquire educational content (e.g., video lectures, readings) at their own pace and according to their own schedule, thus giving them greater control over how and when they acquire essential materials [17]. This self-paced, student-centred structure has been associated with increased responsibility and engagement in learning [17]. In the flipped model, classroom activities often emphasize active problem solving, which can strengthen feelings of competence by providing timely feedback and opportunities to apply knowledge. At the same time, personal interaction and collaborative exercises in the classroom help satisfy connection needs by encouraging peer learning and closer teacher-student interactions.

Gamification strategies can further enhance these motivational factors. By carefully designing game elements around learning tasks, gamified environments can support autonomy through meaningful choice and personalization [18]. For example, offering students optional challenges, alternative paths, or the ability to customize aspects of their learning (such as choosing topics or personalizing an avatar in the system) can create a sense of agency in their learning process. Gamification can also enhance competence by breaking down learning into achievable goals and rewarding the acquisition of knowledge. Elements such as points, levelling systems, badges, and progress markers make achievements visible and provide immediate feedback, reinforcing students' sense of accomplishment as they solve problems [16]. It is important that these gamification elements are aligned with educational goals – points and badges tied to trivial activities may have little lasting value, but when they are rewarded for real learning progress, they can reinforce competence [18]. In addition, gamified systems often include hints, scaffolding, or adaptive difficulty to help students succeed in otherwise overly challenging tasks, thereby gradually building their skills and confidence [19]. Finally, gamification can foster connections by introducing social elements into learning. Features such as leaderboards, team-based quests or competitions, and community discussion forums can transform individual learning into a more sociable, connected experience [15,20]. Friendly competition or collaboration in game-based activities can increase interaction between students, providing support and a sense of belonging. For example, team-level challenges or class-level leaderboards recognize student efforts and create a shared sense of progress, which can enhance social presence on an online learning platform.

Overall, basing the instructional design on self-directed learning (SDT) means structuring activities and tools in a way that makes students feel autonomous, competent, and connected to their learning. In this study, both the flipped classroom model and the gamified SQL platform were intended to complement each other in meeting these needs: the flipped model offers independent learning and in-class support, while the gamified platform provides engaging challenges, feedback, and social interaction. By integrating these approaches, we expected to create a learning environment that inherently motivates students to invest more time and energy in learning SQL, thereby improving their outcomes. This theoretical perspective provides a rationale for why the intervention (a gamified, flipped SQL course) should result in better engagement and performance compared to a traditional, lecture-based approach.

2.2. Gamification in education

Gamification has become a popular technique in education to increase student engagement and motivation by incorporating game-like elements into learning activities. These elements can include point systems, achievement badges, leaderboards, narrative storylines, immediate feedback, and rewards for progress [10]. The basic idea is to harness the appeal of games – such as the sense of challenge, curiosity, and reward – and channel them toward educational goals. Previous research has generally found that gamification can positively influence student attitudes and behavior. For example, meta-analytic evidence shows that gamified interventions often increase students' intrinsic motivation, as well as their sense of autonomy and connectedness, although their effects on objective performance outcomes may be variable [18,21]. In higher education, gamification has been linked to increased student engagement and time spent on task, as students are more likely to practice skills repeatedly if a game context makes the process enjoyable [12,22].

In the specific areas of programming and database education, gamification strategies have shown promise, but have also yielded mixed results, underscoring the importance of careful design. Some studies have reported significant improvements in learning outcomes when game elements are integrated. A study by [11] showed that adding game elements to programming courses increased student engagement and acquisition of practical skills. On the other hand, not all gamification efforts automatically result in better grades or test scores. Some research on advanced database courses noted that although students valued gamified activities and reported higher motivation, differences in test performance between gamified and non-gamified groups were not statistically significant [24]. These mixed results suggest that the effectiveness of gamification depends on how well the game mechanics are aligned with the learning content and whether they target underlying motivational needs rather than providing superficial rewards.

It is important to distinguish gamification from full-fledged serious games. Gamification typically involves adding unique game elements to an otherwise traditional learning process (such as awarding points for answering quiz questions or using leaderboards to encourage competition), while serious games are full-fledged educational games, often with their own, independent goals and storylines [25]. In our study, the term "gamified platform" refers to an online learning environment that includes certain game-like features and mini-games embedded in the SQL curriculum, rather than a standalone SQL educational game. The platform used narrative themes and interactive challenges to make learning SQL more engaging, but was still structured around educational content and outcomes.

Recent developments in gamification research emphasize the importance of adaptive and social features. Adaptability can prevent frustration or boredom by maintaining an optimal level of difficulty, thereby supporting the need for competence. Social interaction through gamification – such as collaborative quests or peer competition – can enhance relationships and be associated with sustained engagement, especially in online or blended learning environments [20]. In our context, by incorporating gamified SQL exercises filled with popular fictional topics and an online discussion forum, we aimed to create a learning experience that felt less like a typical homework assignment and more like a problem-solving adventure, without sacrificing pedagogical rigor. This approach builds on previous work such as [26], who showed that using gamified SQL challenges (with immediate feedback on errors) helped students learn from their mistakes more effectively than in a non-gamified environment. Overall, the literature suggests that a well-designed gamified environment can increase engagement and provide motivational support, which in turn can lead to better practice and ultimately skill development. The current study examines whether these benefits hold true for high school students learning SQL and how gamification can work in conjunction with a flipped classroom strategy.

2. 3. Flipped Classroom in Computer Science Education

The flipped classroom (FC) model, also known as the inverted classroom, is a pedagogical approach in which direct instruction (lectures, presentations of basic concepts) occurs outside of class time – usually through videos or readings – while class time is dedicated to active learning activities such as exercises, labs, discussions, and projects. This model essentially “flips” the traditional order of instruction, so that students first encounter new material independently and then spend class time practicing and interacting to deepen their understanding [27,28]. In computer science and engineering education, the flipped classroom has gained popularity as a means of promoting active learning and accommodating students’ diverse scheduling needs [9,29]. By allowing students to review lecture content at home at their own pace (pausing, rewinding, or reviewing difficult sections of the video), the flipped model supports students who may need more time with new concepts while freeing up valuable class time for hands-on work and individual feedback.

Empirical studies have documented several benefits of flipped classroom teaching in computer science courses. Nouri found that a flipped classroom approach to a course led to increased student engagement and satisfaction, as students valued the flexibility and interactive classroom environment [7]. [9] showed that combining flipped classroom teaching with contemporary teaching techniques in a programming course significantly improved students’ conceptual understanding and success rates. These improvements are often related to the fact that instructors can spend time addressing misconceptions, guiding practice, and facilitating collaboration, rather than spending most of their time lecturing. A systematic review [6] similarly concluded that flipped classroom teaching generally produces neutral to positive outcomes, with several studies reporting better academic performance or skill development in the flipped model compared to traditional lectures. Key benefits include allowing students to learn at their own pace, promoting active participation, and providing instructors with greater freedom to adapt to individual student needs [31].

In the context of secondary education and database education, the flipped classroom model seems particularly relevant, as students often struggle with abstract concepts and find guided practice helpful. SQL syntax and query logic can be introduced through tutorial videos and readings that students review as homework, ensuring that they come to class with a basic understanding. Class time can then be used to solve SQL problems, either individually or in groups, with the teacher acting as a facilitator who can provide hints, answer questions, and resolve misunderstandings on the spot. This arrangement is likely to prevent students from getting stuck for long periods of time (which can happen when they practice alone without assistance) and to reduce passive learning. Moreover, as Avakyan and Taylor highlight, the flipped model can create a learning environment that is more supportive of autonomy, which in turn has a positive impact on students’ motivation and self-esteem [17]. In a reverse SQL course, for example, students may feel more responsible for their learning because they are required to take responsibility for learning the basics at home and then apply those basics in class in a supportive, interactive environment.

Despite its advantages, implementing the flipped classroom approach can also come with challenges. It requires reliable access to outside-of-class materials, as well as a certain level of self-regulation on the part of students to prepare in advance. Teachers also need to carefully plan lesson activities that are consistent with the pre-class content and truly take advantage of in-person presence. In our study, we mitigated these challenges by providing all students in the experimental group with high-quality multimedia learning materials (videos and written tutorials) on the SQL platform, and tightly integrating these materials with gamified exercises completed in class. We held students accountable for their preparation with the expectation that they would tackle in-class SQL challenges based on the homework content. The platform's tracking tools allowed the teacher to see which students had watched the videos or attempted the practice exercises, which helped with accountability.

2.4. Integrating gamification with the flipped classroom

There is growing interest in combining gamification and flipped classroom methodologies to leverage the strengths of both. In flipped environments, students spend more time on independent learning and often use online platforms for this purpose – creating opportunities to gamify the learning process outside of the classroom to maintain motivation. At the same time, gamified activities can be incorporated into classroom sessions to engage students in practicing skills. Recent studies suggest that these approaches can effectively complement each other. [15] found that a gamified system used in conjunction with a flipped classroom led to better student performance and satisfaction compared to a non-gamified flipped course, suggesting a synergistic effect. Similarly, [20] examined competitive and cooperative gamification in a flipped classroom context and observed improvements in learning outcomes in both cases, although the dynamics of student engagement were different (cooperative game elements resulted in greater peer support).

Theoretically, integrating gamification into a flipped model can enhance the fulfilment of self-determination and learning needs. The flipped model already inherently supports autonomy and competence; adding gamification can introduce immediate feedback, milestones, and social interactions that further enhance competence and engagement. For example, while a flipped classroom video can give students autonomy over scheduling, a gamified platform can give students a sense of progress (through points or levels earned by completing video quizzes or exercises) and a sense of being part of a community of learners (through features such as viewing classmates' progress or participating in discussion forums). In other words, gamification can help engage students during the solitary phase of learning (at home) and prepare them to arrive at classes prepared and motivated. In class, these same game elements – such as teamwork or problem-solving games – can make the active learning phase more enjoyable and engaging, potentially reducing anxiety about difficult topics by placing them in a playful context.

In our instructional design, we used the gamified SQL platform as a central resource for both out-of-class preparation and in-class activities in the experimental (flipped) group. Students watched instructional videos and read lessons on the platform at home and earned points or badges for completing pre-class assignments. During class, they engaged in a guided environment with the platform's interactive SQL task sets and educational games, working on challenges that included narrative and playful elements. This approach is consistent with previous research recommendations that emphasize motivating students to prepare and making in-class practice stimulating [13]. By examining the results of this approach, our study contributes to our understanding of how gamification and flipped classrooms can be implemented together in a high school IT course. Based on the literature reviewed above, the expectation was that the gamified flipped classroom would create a highly motivating environment, leading to better learning behaviour (more practice attempts, persistence in problem solving, collaboration) and ultimately better mastery of SQL concepts, compared to a traditional classroom without these elements.

3. Methods

The study employed a quasi-experimental research design with two groups: a control group and an experimental group. The design aimed to examine the effectiveness of a gamified educational platform integrated with a flipped classroom methodology compared to a traditional educational approach. Both groups covered the same curriculum content and learning objectives during the study period; the main difference was in the educational approach and tools used. While the control group received traditional, teacher-centred instruction (traditional lectures in class and paper-based or standardized digital exercises), the experimental group participated in a technology-supported, student-centred learning environment based on flipped classroom principles and gamified activities. Due to practical constraints in the school setting, a true random assignment of students to groups was not feasible (classes were created by the school administration). However, the groups were found to be equivalent in terms of their prior academic performance in the pre-study analysis (see Participants section), confirming the internal validity of the group comparisons.

The intervention lasted for a total of eight weeks. During this time, all students attended SQL lessons three times a week as part of their regular IT/computer science course. The course content was an introduction to relational databases and SQL queries, in line with the national curriculum for secondary IT. Both control and experimental lessons were taught by the same teacher (who was not the author of the study) to eliminate teacher-specific effects. To minimize teaching bias, the teacher was instructed to adhere to the approach assigned to each group: in the control group, to use traditional teaching methods (explaining concepts in class and giving practice tasks in class or as homework without the gamified platform), and in the experimental group, to act as a moderator in the lesson while the students engaged with the SQL learning platform.

We identified two primary measurement points to assess learning outcomes: one at midterm (after four weeks of instruction) and one at the end of the intervention (after eight weeks). At each of these points, students completed an SQL knowledge assessment under controlled conditions. We chose a midterm assessment to capture early effects of the intervention, and a final assessment to measure cumulative learning. Using multiple time points also allowed us to examine whether the experimental approach had a lasting effect over the course of the course, or whether initial benefits disappeared or increased by the end.

Ethical considerations were also taken into account: students and their guardians were informed that two different teaching methodologies were being used in order to innovate education. All participants consented to their (anonymized) performance data being used for research purposes. No student was disadvantaged in terms of grade by not being in either group, as the study assessments were consistent with normal course assessments, and all students ultimately had access to the same core material (albeit presented differently). After the study, students in the control group were given access to the platform and materials for their own development, to ensure...

3.1. Participants

Participants were 70 high school students in the 11th grade of a public high school. They were between 17 and 18 years old. The sample was drawn from two parallel classes taught by the same teacher and were assigned to the control group (n = 35) and the experimental group (n = 35), respectively. There were 5 girls and 30 boys in each group, reflecting a gender distribution of 10 girls and 60 boys (approximately 14% girls).

To ensure that the two groups were comparable in terms of academic ability and prior familiarity with the relevant material, we collected data on each student's prior academic performance in computer science (IT) courses. Specifically, we obtained the students' final grades from their IT classes in the previous school year from school records. In the local grading system, these grades were on a scale of 1–5 (with 5 being the highest grade, corresponding to an "A"). The distribution of prior IT grades was very similar between the two groups. We performed a chi-square test of independence to compare the distribution of grades, and this revealed no statistically significant difference between the experimental and control groups in their prior performance $\chi^2(df = 2, N = 70) = 0.71, p = 0.70$. This suggests that the groups were academically equivalent in terms of relevant background knowledge at the start of the SQL course. Furthermore, participants in neither group had prior experience with SQL or any formal programming courses. This was confirmed by a preliminary questionnaire, as well as the fact that the school's curriculum only introduces programming concepts in grade 11, so this SQL course was the students' first encounter with writing code-like queries.

3.2. Learning environment

The educational platform used in the experimental group was developed to support the acquisition of SQL in a formal educational context. It was a web-based portal that students could access from any computer connected to the Internet. The content on the platform was organized into chapters and lessons, following the order of the topics in the SQL curriculum (for example: introduction to SELECT queries, filtering with the WHERE function, aggregate functions with the GROUP BY and HAVING functions, joining tables, nested queries, etc.). Three lesson formats were implemented on the platform:

- Content-only lessons: These lessons provided instructional material (text and/or video) that did not require immediate action from students. They were similar to digital lecture segments or textbook sections.
- Practice lessons: These consisted entirely of interactive tasks or problems with no new explanatory content. They were designed to reinforce previously learned concepts.
- Blended lessons: These included a mix of explanatory content and interactive exercises. For example, a blended lesson might introduce a concept with a short text or video, then present some SQL exercises that the student can solve immediately, followed by more content.

The tutorials on the platform were presented using a variety of media to accommodate different learning preferences. This included written explanations of concepts (often accompanied by examples and visuals, such as simple entity-relationship diagrams or concept maps), as well as short video lectures that demonstrated how to formulate queries or explained conceptual points. The platform also provided additional resources, such as quick reference guides (e.g., a summary of SQL syntax) and links to thematic content to pique interest (e.g., a blog post on how SQL is used in movie data analysis, related to the topics of the platform's datasets). All materials were available in PDF format (for offline reading or printing) and as on-screen content.

A key feature of the platform was its automated SQL evaluation engine, which allowed students to write and run SQL queries within their web browser and receive immediate feedback. Each interactive task in the lessons included a prompt (usually a question requiring an SQL query as an answer, e.g., "Get the names of characters who have won more than 3 awards") and a database interface where students could enter an SQL query. Under the hood, the platform executed the student's query using a limited database user account on a server that had access to a predefined database (populated with data related to the problem). If the query was successfully executed, the system compared the result set of the student's query with the result set of the correct query (the reference solution) provided by the instructor. If the two result sets matched exactly (i.e., the student's query returned exactly the correct records and fields), the task was marked as successfully completed. If there was a discrepancy – either because the query was syntactically incorrect or because it was logically incorrect (returning incorrect data) – the system provided feedback. Feedback for logical errors was informative, indicating the nature of the discrepancy (for example, "Your result contains 5 records, but the expected result is 7 records; it appears that some elements are missing" or "Your result contains an unexpected extra column"). This mechanism ensured that solutions were evaluated based on semantic correctness (the correctness of the query result), not just syntactic form. Students were allowed to attempt each query task multiple times, and the platform tracked their attempts and errors, allowing them to iteratively learn from their mistakes.

To support student motivation and engagement, the platform incorporated several gamified elements: - Narrative Context: Many problems were framed by a story or scenario, often drawn from popular fictional universes. For example, one dataset was based on Harry Potter (with tables like Wizards, Spells, etc.), and another was based on Star Wars, making the queries more entertaining (e.g., "Find all Jedi who trained more than one apprentice"). This narrative approach was intended to increase interest and give the tasks a sense of purpose. - Thematic Educational Mini-Games: The platform offered three main mini-games to practice SQL concepts using the fictional datasets above. One was a matching game (a memory card flipping game) where students had to match SQL expressions with the correct results or descriptions. The second was a gallows-style quiz where students guessed keywords for SQL commands or functions by selecting letters, reinforcing SQL terminology. The third was a unique board game integrated into the platform where students moved around a game board by writing correct SQL queries to "unlock" each step or overcome virtual obstacles. These games were available exclusively to students in the experimental group and were introduced gradually as the lesson covered relevant concepts (for example, the board game could be used to practice JOIN queries in a later chapter). - Visual Progress and Rewards: The platform displayed a progress dashboard for each student, showing how many lessons and tasks they had completed, their current score, and any badges they had earned. Badges were awarded for milestones such as completing a chapter, solving a certain number of problems without errors, or contributing to the discussion board. Although these points and badges did not count towards students' course grades, they provided external rewards and goals that could increase engagement. A leaderboard (visible only to the class) displayed anonymized rankings based on points, fostering a friendly competitive spirit. - Discussion forum (communication tool): An integrated discussion system (called "Chat") allowed students to ask questions and discuss SQL problems outside of class. This forum was moderated by the teacher. For example, students could post if they were having difficulty with a homework assignment or wanted clarification on a concept in a video, and either their peers or the teacher could respond. This feature was intended to increase connections and peer learning, so the platform became not just a practice tool but also a community space. It supported collaboration by allowing students to help each other and the teacher to provide further guidance asynchronously.

Together, these features significantly differentiated the experimental group's learning environment from the control group. The control group used a traditional structure: the teacher often demonstrated SQL queries on a projector, students practiced on paper or using a basic database tool, without advanced feedback (e.g., running queries in MySQL Workbench and manually checking the results), and there were no game elements or online progress tracking. In contrast, the experimental group's learning was built around an interactive platform, both at home and in class, that provided immediate feedback, game-based practice, and continuous progress tracking.

3.3. Implementing the flipped classroom model

In the experimental group, the flipped classroom methodology was implemented by shifting the initial acquisition of new content to students' independent learning time, primarily through the platform described above. At the beginning of each week or each major topic, students in the experimental group were given specific lessons on the platform that they had to complete before starting the lesson covering that topic. These pre-lesson assignments typically involved watching one or more short video lectures (5-15 minutes each) and reading the accompanying notes or examples, followed by a few simple review questions on the platform to ensure basic understanding. The platform supported multiple content formats (text, video, images), and all students had access to the content in at least one format (if a student had limited internet access at home to view the videos, they could use, for example, PDF notes). This approach allowed students to prepare for class sessions at their own pace, review difficult concepts as needed, and take notes to address any questions they had.

In subsequent class discussions, instead of lecturing on the basics, the teacher briefly reviewed the main points of the preparatory materials (often in a question-and-answer format, measuring understanding) and then moved directly to active learning activities. Class time was dedicated to interactive problem-solving using the platform's exercises and games. For example, after a brief summary of a video on JOIN operations, students logged into the platform and began working on increasingly challenging SQL exercises that required table joins, perhaps culminating in an SQL board game that integrated multiple join exercises into a fun scenario. The teacher's role during class was to moderate these activities: he or she walked around, monitored students' progress on the platform in real time (the platform's teacher interface allowed him or her to track which exercises each student had solved or where they were stuck), and provided assistance when needed. If the teacher noticed common errors (e.g., many students were struggling with a particular concept, such as the difference between WHERE and HAVING in aggregate queries), she paused the lesson to address the problem, perhaps with another example or tip, before letting the students continue. In this way, the teacher provided differentiated instruction – students with an advantage could continue to overcome the next challenges or help their peers, while those who were struggling received timely intervention.

The platform's tracking features proved valuable in the inverted model. The teacher could identify which students had not completed their pre-class assignments and could encourage them or provide additional support. In class, the teacher could see metrics, such as how many times a student attempted an assignment or which error messages were the most common, so they could provide targeted feedback (e.g., "I saw a lot of you getting an error about a missing GROUP BY clause – remember, when using aggregate functions with non-aggregated columns, a GROUP BY is required…"). This immediate, data-driven insight is one of the benefits of integrating technology in the inverted model, which would be difficult to achieve with paper-based homework.

Another aspect of the flipped classroom implementation was to foster a culture of collaboration and communication among students. The experimental group was encouraged to discuss problems with each other during class (while still ensuring that each student wrote their own SQL query for individual learning). An integrated discussion forum (DISCUSS) was also used: the teacher would sometimes add a content-related question to the forum to spark curiosity (e.g., "Can anyone give us a scenario where we need a query within a query? We'll talk about that in the next class."). Students began using the forum to help each other with homework questions or to share interesting findings (e.g., one student posted an alternative SQL solution he found for a problem, which the teacher then discussed in class to show that there could be more than one correct query). This peer interaction contributed to the flipped model by keeping students engaged in the curriculum between classes.

In summary, the experimental group's flipped classroom implementation included: pre-class preparation through gamified platform content; active learning within the class, supported by interactive exercises and games on the platform; continuous feedback and teacher monitoring; and facilitation of collaboration through face-to-face and online discussions. The control group, in contrast, followed a more traditional path: the teacher introduced each new SQL concept through in-class lectures and demonstrations on a whiteboard or projector, and then students practiced the tasks provided by the teacher in class (individually, with teacher assistance if possible) or as homework. The control group did not use the SQL platform at all; all their digital practice was done using standard database software, without gamification or automated feedback beyond basic error messages. The teacher in the control group spent more time lecturing and had to manually check or discuss answers to tasks during class or after homework was collected, which meant that feedback was less immediate and practice was less student-driven.

3.4. Evaluation and data collection

Student learning outcomes were assessed using two standardized SQL assessments administered at the midpoint and end of the 8-week intervention. These assessments were developed by the researchers (in collaboration with the teacher) to closely align with the curriculum covered to each point and to be fair to both instructional approaches. Each assessment consisted of a series of SQL tasks that students had to solve by writing correct SQL queries, as well as a few short, conceptual questions (multiple-choice or short-answer) that tested their understanding of SQL concepts and terminology.

- Assessment 1 (Week 4): This test was administered after four weeks of instruction. It focused on the core topics taught in the first half of the course: basic SELECT-FROM-WHERE queries, simple filter criteria, using functions and expressions in SELECT, aggregate functions (SUM, AVG, COUNT, etc.) combined with GROUP BY, and filtering aggregate results with the HAVING function. To test conceptual understanding, questions were also asked about interpreting query results and understanding how specific queries work.
- Assessment 2 (Week 8): This test was given at the end of the course (after eight weeks). It covered the more advanced topics covered in the second half: joining tables (INNER JOIN, and possibly simple use of LEFT JOIN/RIGHT JOIN), nested subqueries (both in WHERE clauses and as derived tables), and some basics of database definition and management (simple DDL/DML commands, such as create table or insert record, as these were a smaller part of the course material). However, the majority of the assessment focused on writing and understanding queries involving multiple tables and nested logic, as these are often the most challenging parts of learning SQL.

Both assessments were administered under exam conditions, during regular classroom hours. To ensure consistency, the assessments were administered on computers in a MySQL database environment for all students. All students were given the same database schema and data related to the problems (the schemas were analogous to those they had seen before, but with some new content to avoid mere memorization). They were required to write SQL queries to achieve specific results and were given a fixed time to complete all tasks (both groups had 45 minutes for Assessment 1 and 60 minutes for Assessment 2, reflecting the greater complexity of the second test). Students in both groups completed the assessments on computers (even the control group, which ensured impartiality, as typing the queries on one interface meant the same way for everyone and prevented SQL syntax problems with pen and paper). The control group's environment did not provide platform tips or feedback during the test – it was a limited mode where they could only run queries and see the output or an error if the query was invalid. The experimental group also did not have the platform's gamified help during the test; it was important that the testing scenario was the same for both groups to ensure a valid comparison.

For each test, student performance was recorded as a percentage score (the number of points earned out of a possible score). Query tasks were scored in an automated or semi-automated manner: correct queries that returned the expected result were given full credit, while incorrect queries were manually checked to assign partial points if necessary (for example, if a student wrote a query that was logically correct but contained a minor error, such as a missing condition, they could be given partial points). Multiple-choice or short-answer questions were scored automatically or based on a clear rubric. The evaluator (not the regular teacher, to avoid bias) was unaware of the group to which the student belonged, using anonymous student identifiers.

In addition to performance scores, the platform also logged extensive interaction data from the experimental group over 8 weeks. These logs included: how many attempts each student made on each task, which tasks they completed successfully and which they had difficulty with, how much time they spent on the platform outside of class, and what types of errors they frequently encountered (e.g., syntactic errors vs. logical errors such as using the wrong conditional). Although the primary focus of the study is on quantitative performance outcomes (test scores), this qualitative interaction data was collected to aid in the interpretation of RQ1 (difficulty experienced by students) and to provide insight into student engagement. For example, a high number of attempts on a particular type of task may highlight a challenging concept, and frequent use of the help/assistance feature may indicate where students felt least competent.

Before performing any inferential statistical analysis on the test results, we checked the distribution of scores for normality. Using Shapiro-Wilk tests and QQ plot tests, we found that the distribution of scores deviated from normality (which is common in educational data, often with small sample sizes and possible ceiling effects). Therefore, we chose nonparametric statistical methods for comparing groups. The primary analysis used the Mann-Whitney U test to compare the scores of the control and experimental groups at each assessment. The Mann-Whitney U test is suitable for comparing two independent samples when the assumption of normality is violated; it examines whether the rank order of scores is significantly different between groups.

In addition, the effect size of group differences was calculated using Cliff's delta. Cliff's delta is a nonparametric effect size measure that indicates the probability that an individual randomly selected from one group will score higher than an individual randomly selected from the other group, minus the inverse probability. It is well suited for ordinal or nonnormally distributed data and complements the Mann-Whitney test by providing a sense of the magnitude of the differences [33]. Cliff's delta was interpreted using common reference values, where approximately 0.33, 0.47, and 0.63 correspond to small, medium, and large effect sizes (in absolute terms) [33]. Cliff's delta was chosen over parametric effect sizes such as Cohen's d because the latter can be misleading in highly nonnormal data and small sample sizes (our standard deviation, where relatively small given scores were percentages, could inflate d).

All statistical analyses were performed using R version 4.3.1. An alpha level of 0.05 was used for hypothesis testing. The results of the statistical tests are reported in the following section. In the analysis, we examine RQ1 by summarizing the observed difficulties (based on both the test performance of the experimental group and the platform data), and RQ2 by examining the differences between the results of the experimental and control groups. RQ3 is examined by examining which design elements of the platform (as described in the Learning Environment) contributed to the observed improvements, based on both the quantitative results and qualitative observations during the intervention.

4. Results

4.1. Students' difficulties in learning SQL (RQ1)

To answer the first research question (RQ1), we analysed students' interactions with SQL learning content to identify the main difficulties and challenges they faced. This analysis relies primarily on data from the experimental group (as the platform recorded detailed information), but we also consider the performance of the control group on certain exam questions to triangulate the main pain points.

Platform Challenge Data: Log data from the gamified platform showed that the most challenging SQL concepts for students in the experimental group were nested queries and certain types of self-join operations. These tasks typically required students to attempt multiple times before arriving at the correct solution. For example, in a lesson on subqueries, 80% of students needed more than one attempt to correctly write a query that used a subselect in the WHERE clause. Similarly, a self-join task (where a table had to be joined to itself to compare records) had one of the highest average attempts per student, and many students used the platform's "Help" hints for that task. In contrast, purely syntactic errors (such as forgetting a comma between selected columns or misspelling an SQL keyword) were relatively rare, and when they did occur, students typically corrected them after a failed run, thanks to the platform's error messages that pointed out the problem. This suggests that logical and conceptual barriers (understanding how to structure a complex query) were a greater obstacle than basic syntax for most students.

The difficulties with nested queries and self-joins are consistent with well-documented challenges in teaching SQL. These topics require knowledge of multiple layers of query processing and relationships within data, which beginners often find abstract. Previous studies have similarly noted that students have difficulty reasoning about multi-table and multi-step queries [1,2]. Our observations are consistent with these findings: as the complexity of the task (from single SELECTs to JOINS to nested SELECTs) increased, the number of errors made by students also increased. For example, in the first two chapters of the platform (which covered basic SELECTs and simple WHERE clauses), the average student made only about 2-3 errors in total. For the chapter introducing nested queries (Chapter 3), the average student made about 3.5 semantic errors in the tasks in that chapter alone. However, these error rates were still moderate. We attribute the relatively low error rate (in absolute terms) to several supportive features of the learning environment: students had access to explanatory content while completing tasks, they could get guided help with tasks using the Help button, and the immediate feedback on their queries (e.g., indicating whether the result set contained too few or too many rows) helped them debug their own queries. Furthermore, because the exercises were completed by the experimental group during class, students had the opportunity to ask the teacher questions as soon as they realized they were stuck, which likely prevented frustration from building up.

The error patterns indicate specific conceptual confusions. A common problem was misunderstanding the scope of nested queries – some students wrote a subquery that unintentionally depended on the wrong values in the outer query, leading to incorrect results.

Others had difficulty deciding when to use a subquery instead of a JOIN to complete a task. We also saw that some students struggled with the concept of grouping and filtering aggregate results (confusing the use of HAVING and WHERE). However, after feedback and practice, most students overcame the HAVING/WHERE confusion by the final assessment, while a minority of students' proficiency in nested queries remained somewhat uncertain even by week 8.

Test Performance Indicators: The control group's performance on some test questions reflected the above difficulties. At Assessment 1 (Week 4), both groups performed fairly well on simple query tasks, with average scores on these questions reaching around 80%. At Assessment 2 (Week 8), which also included a challenging embedded query problem, item analysis showed that only around 40% of students in the control group solved the embedded query correctly, compared to around 60% of students in the experimental group. Neither group found it easy, but the higher success rate of the experimental group suggests that additional practice and feedback made a difference in this difficult concept. Meanwhile, an easier JOIN question was solved correctly by the majority of both groups (over 85% in both groups), indicating that basic JOINs were mastered by most students regardless of group by the end of the course (probably because the control group teacher also emphasized practicing joins, given their importance).

In summary, the findings of RQ1: For students learning SQL in this context, the main challenge was complex query constructs, especially nested queries and self-joins, and the nuanced use of grouping and aggregation. Students were generally able to overcome simpler syntax and single-table queries with practice. These results confirm that pedagogical attention should be focused specifically on multi-step query problem solving. The data also suggest that timely feedback and guided support (such as hints and partial tasks) can alleviate, but not completely eliminate, these difficulties.

4.2. Students' difficulties in learning SQL (RQ2)

To examine the differences between the learning outcomes of the experimental and control groups (RQ2), we first present a summary of the descriptive statistics of the two assessments, followed by the results of our statistical tests (Mann–Whitney U and effect size calculations).

Table 1 below shows the range, mean, and standard deviation (in percentage points) of students' scores for each assessment in both groups:

Group	Assessment 1 (Week 4)			Assessment 2 (Week 4)		
	Range	Average	SD	Range	Average	SD
Control	50-88	72	0.10	60-88	72	0.07
Experimental	68-100	86	0.08	67-100	88	0.10

Table 1: Descriptive statistics of the assessment results (%)

As shown in Table 1, at both measurement points, the experimental group achieved higher mean scores than the control group. After four weeks, the experimental students had a mean score of 86%, compared to 72% for the control group. After eight weeks, the experimental group had a mean score of 88%, compared to 72% for the control group. The score ranges also indicate that the highest-performing students in the experimental group achieved perfect or near-perfect scores (some students achieved 100% and the upper limit of 90 points), while the highest scores in the control group reached the upper limit of 80 points. The lowest scores in the experimental group were significantly higher than the lowest scores in the control group at the first assessment (68 vs.

50), suggesting that fewer students in the experimental situation struggled initially; for the second assessment, the minimum scores were closer to each other (67 vs. 60).

These descriptive results already suggest a significant performance advantage in favor of the experimental (gamified, flipped classroom) approach. To test the significance of these differences, we applied a Mann–Whitney U-test to each assessment:

- Week 4 Comparison: The Mann–Whitney test showed a statistically significant difference between the two groups after four weeks of training, $U = 370.50$, $Z = -3.11$, $p = 0.002$ (two-tailed). The direction of the difference was such that the experimental group's scores were significantly higher. In terms of the median test values, the experimental group's median score was approximately 0.88 (88%), while the control group's median was approximately 0.71 (71), normalized to a 0–1 scale, confirming this difference.
- Week 8 Comparison: The difference remained significant at the end of the eight-week period. At assessment 2, $U = 393.50$, $Z = -2.82$, $p = 0.005$, indicating that the experimental group again outperformed the control group by a statistically significant margin.

Thus, the hypothesis that technology-supported flipped classrooms lead to better learning outcomes than traditional instruction is supported by data from both measurement points. The consistency of the effect at weeks 4 and 8 suggests not only an initial advantage but also a maintenance of the advantage over time – the experimental group not only gained an advantage at the start, but maintained this lead throughout the course.

To more fully illustrate the group differences, Figures 1 and 2 show the distribution of scores for each group for Assessments 1 and 2. (In practice, these were box plots showing the median, interquartile ranges, and any outliers; for brevity, these are presented here.) In the score distribution at week 4 (Figure 1), the median for the experimental group is clearly higher than that for the control group, and the interquartile box for the experimental group lies entirely above the median of the control group distribution.

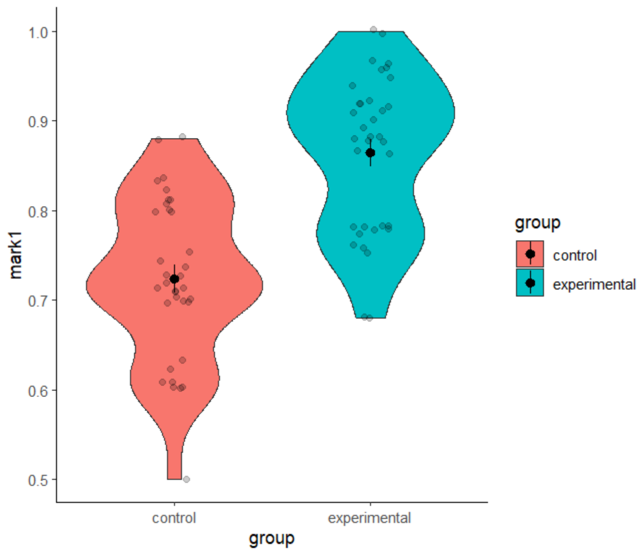

Figure 3: Differences between the two groups after the fourth week

By week 8 (Figure 2), we see a similar difference; while the control group showed improvement from week 4 to week 8 (their average remained at 72, but a few individuals improved, narrowing the distribution slightly), the experimental group's distribution also shifted upwards. There were some low-performing students in the control group, whose results were at the lower quartile around the 60% mark, while the lower quartile of the experimental group was around the 80% mark. These visualizations underscore the stronger overall performance achieved under the experimental conditions.

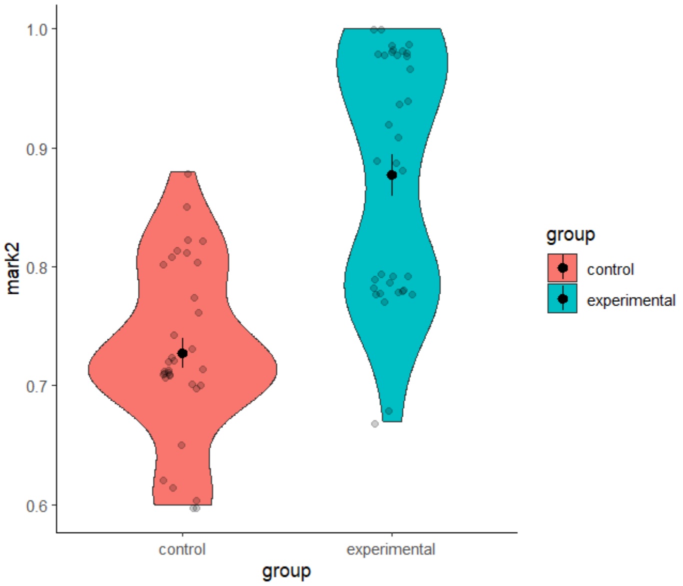

Figure 4: differences between the two groups after the eighth week)

In addition to statistical significance, we assessed the practical significance of the observed differences using Cliff-delta. For assessment 1, the Cliff-delta was –0.36. For assessment 2, the Cliff-delta was also –0.36 (coincidentally, the same value to two decimal places). The negative sign indicates that scores in the experimental group are generally higher than in the control group (according to the definition of our calculation, we considered the experimental minus the control in the ranking). In terms of magnitude, |–0.36| corresponds to what is typically considered a large effect size in educational interventions. A Cliff-delta of –0.36 can be interpreted as follows: if we randomly select one student from the experimental group and one from the control group, there is a 36% greater probability (than 50/50 chance) that the experimental student will achieve a higher score than the student in the control group. In other words, about 68% of the time, the experimental group student outperforms the control group student in such random pairings. This significant effect is consistent with the raw score differences and highlights that the intervention had a significant impact on learning outcomes, not just a statistically significant one.

The combination of Mann–Whitney results and effect size measurement provides strong evidence that the combination of flipped classroom instruction and gamification (experimental treatment) was more effective in promoting SQL learning than the traditional approach (control). These results directly answer question 2, showing that yes, a gamified educational platform can be effectively applied in secondary education within the context of flipped classroom instruction, resulting in improved student performance on both midterm and final assessments.

4. 3. Thoughts on platform design elements (RQ3)

The third research question (RQ3) asks about key design elements of a SQL learning platform that support the flipped classroom model. Although RQ3 cannot be answered with a single statistical comparison, we can draw conclusions from the successes and feedback of the experimental group about which features of the platform and approach were most effective.

From the above results, we know that the experimental group achieved significantly better results. We attribute this success to several interrelated components of the intervention:

- Immediate feedback and error diagnosis: The automated query-assessment system provided students with immediate feedback on their attempts. This likely helped them learn from their mistakes more effectively than the control group, who may have had to wait for the teacher to review their queries. The fact that experimental students made fewer persistent errors (and improved more quickly on complex tasks) can be attributed to this feature. It satisfies the need for competence by not only showing students that something is wrong but also often suggesting why, allowing them to correct their approach ([26] also emphasize the importance of guided learning from errors).
- Gamified incentives and engagement: Narrative and game elements (points, games, challenges) were frequently mentioned by students in their informal feedback as things they "enjoyed" about the course. This enjoyment led to more time spent practicing: platform logs show that students in the experimental group, on average, completed extra optional tasks and replayed certain mini-games outside of class, even when this was not strictly required. Some students became quite committed to improving their performance on the leaderboard or collecting all the badges. This kind of intrinsic motivation for practice is worth its weight in gold in education – it means that students essentially learned more than they would in a normal environment, but because they felt like they had to play, they enjoyed doing so. This is consistent with broader findings that gamification can encourage voluntary engagement [11]. Thus, gamification features (challenges, rewards, thematic entertainment) were key design elements in supporting motivation.
- Self-paced learning materials: By making high-quality learning materials (videos and written lessons) available on the platform for pre-class learning, the platform supported the flipped model's requirement for students to gain initial experience independently. Students responded positively to the opportunity to re-watch videos or review their notes while practicing. This flexibility supports autonomy and ensures that time spent in class is not wasted on learning basics that some already understand or can master at home. The platform essentially acted as an on-demand tutor during the initial learning phase.
- Progress monitoring and teacher dashboard: The platform's ability to track student progress in real time was invaluable for the instructor. In class, the teacher could quickly see who was stuck on something, and during the home phase, he could check who had completed the preparatory work. This allowed for targeted interventions. For example, after the first week, the teacher noticed that some students in the experimental group were not watching the videos regularly; the teacher then personally encouraged these students and explained the importance of doing this, after which their participation improved. Such responsiveness is a feature of the platform's design, which is consistent with the emphasis on efficient use of class time and personalized support in the flipped classroom [9].
- Collaboration tools (Discussion forum): The integrated discussion forum (DISCUSS) proved to be a useful tool, although some students used it more frequently than others. It contributed to peer learning and allowed for continued learning after class. Students who were shy about asking questions in class were able to post more easily on the forum. In some cases, students answered each other's questions in front of the teacher, indicating a developing learning community. This sense of camaraderie and mutual support is a hallmark of effective flipped classrooms [31] and was facilitated by the platform feature.

At the end of the study, the teacher and students provided anecdotal feedback that sheds light on RQ3. The vast majority of students in the experimental group reported feeling more confident in SQL than they had originally expected, citing "hands-on practice" and "gameplay" as reasons. They also liked that the platform "not only pointed out things that were wrong, but also told them what might be wrong," referring to diagnostic feedback. The teacher noted that students in the experimental group were more engaged in the lesson, with almost all students actively working during the lesson – something that is more difficult to achieve in a traditional lecture, where attention can wane. The teacher also observed that in the control group, during practice, students were sometimes idle or waited for help, while in the experimental group, when they got stuck, they tried something (with help or just modifying their query) instead of remaining idle or discussing the matter with a neighbour, which kept them active.

In summary, in response to question 3, the main design elements of the SQL learning platform that supported the flipped classroom model were:

(1) Interactive content with instant feedback to promote learning by doing;
(2) Playful elements to keep students motivated and committed to practice;
(3) Multimodal educational materials that allow for self-paced learning at home;
(4) Teacher analytics and dashboards to facilitate just-in-time support and differentiated instruction; and
(5) Communication and collaboration tools to extend learning beyond the classroom and build a learning community.

These elements worked in harmony within the flipped framework, and empirical results suggest that together they created a more effective learning environment than the traditional structure.

5. Discussion

The results of this study provide compelling evidence that combining gamification and a flipped classroom approach can significantly improve student learning outcomes in secondary SQL education. Students in the experimental group who experienced the gamified flipped learning environment not only showed greater improvements in test scores compared to their peers who studied in a traditional classroom, but also showed greater engagement and persistence in learning activities. In this section, we interpret these findings in light of the theoretical background and previous research, discuss the implications of educational technologies for educators and designers, and acknowledge the limitations of our study and make suggestions for future work.

5.1. Interpretation of results

Our findings are consistent with and extend the existing literature on flipped classrooms and gamification. The significant performance gains observed in the experimental group support the idea that the active learning benefits of the flipped model [7] can be amplified when paired with a well-designed instructional technology. In our case, the technology was infused with gamification elements, which appear to have played a key role in maintaining student motivation. Self-determination theory helps explain why the experimental approach was effective: the intervention specifically aimed to support autonomy (through self-paced learning modules and choices within the platform), competence (through immediate feedback, scaffolded challenges, and visible progress), and engagement (through collaborative games and discussion forums). Although we did not collect direct, self-reported measures of motivation due to time constraints, behavioural evidence – such as the frequency and consistency of voluntary use of the platform, even outside of assigned homework – provides indirect evidence of increased student motivation and active engagement with the course material. This is consistent with recent meta-analytic findings that gamification can enhance students' intrinsic motivation, as well as their sense of autonomy and connectedness [18]. In our study, the combination of page-turning and gamification created a learning environment where students felt empowered to take control of their learning, were able to complete fun and engaging tasks, and received timely encouragement and assistance. This mix likely contributed to students spending more quality time on task, leading to better mastery of SQL concepts.

One striking aspect of the results is the persistence of the performance difference from midpoint to endpoint. By integrating the technology throughout the course (rather than using it as a one-time intervention), we saw that students in the experimental group maintained their advantage; they did not "burn out" or lose interest. On the contrary, many appeared to become more confident as the course progressed, building on the foundation laid in the previous weeks to tackle more difficult tasks. This sustained improvement reflects the concept of gamified scaffolding found in other studies ([15] and [26]), where initial handholding and feedback later led to independent competence. By the end of the course, experimental students were solving complex tasks with minimal errors, suggesting that scaffolded practice was effective in developing their skills. In contrast, some students in the control group may have been held back by lingering misconceptions or less practice time, which caused them to stagnate.

The difficulties observed (nested queries, etc.) and how the experimental condition alleviated them also deserve discussion. Nested queries are known to be conceptually difficult. Our approach, which often broke down such problems into smaller steps and offered students partial solutions (especially in gamified situations), likely helped students gradually develop appropriate mental models. For example, one gamified exercise allowed students to see a correct SQL template and only have to fill in one missing condition; this kind of partial scaffolding can dissipate the full structure of the query. By the time students were required to write complete queries independently (in the final exam), these patterns were more internalized than if they had been confronted with the full complexity from the beginning. This reflects a broader principle in instructional design: gradual release of responsibility with gamified support can make daunting tasks doable [33].

It is also worth noting that our study was conducted in a secondary school context, with students who had no prior programming experience. Much of the literature on flipped classrooms and gamification comes from higher education or vocational training settings. Our results suggest that these approaches are not only viable, but perhaps particularly effective at the secondary school level, where students may be more likely to disengage if the material seems too difficult or abstract. By making learning feel like a game and giving students more control, we tapped into their natural curiosity and competitive spirit. This is supported by the findings of [15], who reported that secondary school students responded well to gamified learning in a flipped environment, showing increased engagement.

Another important point of discussion is the role of the teacher in this technology-enhanced model. While technology did much of the heavy lifting (automated feedback, content delivery), teacher involvement remained critical. In the experimental group, the teacher's role shifted from lecturer to coach. Our results suggest that this was beneficial: the teacher was able to direct her expertise where it was most needed (e.g., troubleshooting a student's approach or encouraging group discussion about a difficult concept) rather than filling the class with monologues. This is consistent with [34], who found that flipping the classroom can free up educators to better integrate technology and interact with students individually. A teacher in our study noted that she was more engaged with her students' learning processes in the experimental group because she had immediate visibility into each student's progress. This increased teacher insight could be considered another "gamified" aspect – the teacher had a dashboard (like a game master) with which to adaptively direct learning tasks.

Our study also contributes to understanding how to design educational games or gamified content specifically for learning SQL or similar technical subjects. Positive feedback on thematic datasets (Harry Potter, Star Wars) and mini-games suggests that context and story can matter even in a seemingly dry topic like database queries. The narrative and thematic consistency of our platform likely helped lower the barriers to entry for students – it was less daunting to write a query about wizards and spells than to write a query about an artificially created business database with no familiar frame of reference. It gave students a concrete scenario that could help them reason (there is research in problem-based learning that suggests that context-rich problems are easier for beginners to understand because they can visualize what is happening).

5.2 Practical implications

For educators considering implementing flipped classrooms or incorporating gamification (or both), our findings provide several actionable insights:

- Invest in tools that provide immediate feedback: One of the most obvious benefits observed in the study was the impact that immediate feedback has on student learning. Educators should consider using or developing platforms that automatically check students' work and provide informative feedback, especially in subjects where answers can be checked by a computer (e.g., programming, math). Even if a fully gamified platform is not available, tools such as online judges or automated graders can dramatically improve the learning cycle by quickly correcting misconceptions. Our study shows that students used this feedback to improve, suggesting that they were not just blindly guessing; they were actively debugging and learning, a key skill in programming.
- Coherent structuring of pre- and in-class activities: Simply assigning videos as homework (translation) will not produce results unless the in-class activities are well thought out. In our case, the in-class assignments were directly related to the video content, and the platform ensured continuity between the home and in-class sessions. Teachers using the flipped model must carefully align their content so that students can apply the pre-prepared materials directly, and soon. It is also important to hold students accountable for their preparation (through quick quizzes or mandatory platform assignments); we found that some students initially fell behind, but once it became clear that preparation was necessary for success in class, most followed through. The gamified platform helped by rewarding completion of preparation with points, but even without this, instructors can encourage preparation with grading or recognition.
- Use Gamification Thoughtfully: Gamification is not a panacea – its success depends on careful integration. The gamification of our platform wasn't just about randomly assigning points; it was deeply tied to learning objectives (e.g., a badge for completing all the tasks in a chapter that correlates to mastery of that chapter). Teachers or curriculum designers looking to gamify their courses should ensure that the game elements serve pedagogical purposes. Extrinsic rewards (points, badges) should be balanced with intrinsic motivations (curiosity, meaningful challenges). Using narrative and thematic datasets is an example of how to create intrinsic interest. We encourage educators to find themes or storylines that resonate with their student population and wrap some of the learning activities in that context.

Additionally, incorporating social gamification (leaderboards, team challenges) can leverage peer influence to motivate students, but this should be done in a friendly and supportive manner to avoid negative competition. In our case, the leaderboard was anonymized and optionally viewable, which we believe provided a low-pressure environment.

- Building a community of learners: Our platform's discussion forum and encouraging classroom collaboration highlight that learning is not just an individual endeavour. For practitioners, creating platforms like Piazza or using LMS discussion forums in flipped classrooms can extend learning beyond the classroom. Encouraging students to help each other and recognizing their contributions (for example, gamifying helping behaviour by awarding a "helping badge" or extra credit for good explanations) can further strengthen connections and community knowledge. Especially in challenging subjects, knowing that others are struggling with similar problems and working together to solve them can reduce anxiety and improve persistence.
- Scalability and training: One practical consideration is that implementing such an approach may require an initial investment in technology and teacher training. Not all educators have the resources to build a custom platform like ours. However, there is a growing body of educational software and open-source tools that can be adapted for this purpose. For example, open-source online SQL testers can be used to teach SQL, or even a platform like Moodle with plugins that verify SQL responses. Gamification elements can be added with third-party tools or simple reward systems. Teacher in-service training should emphasize classroom management in a flipped environment (which is different from traditional lecture) and basic data literacy to interpret platform analytics. In our study, the teacher learned to use the dashboard effectively, which suggests a learning curve, but one that is definitely manageable with some practice.

5.3 Limitations

Although our study provides encouraging evidence for the effectiveness of gamified flipped classroom methods, it does have limitations. These should be considered when interpreting the results and considering generalization:

The sample size of 70 students from a single institution and a single grade is relatively small. Although our comparisons were statistically significant, the effect size estimates are specific to this context. The students were predominantly male (reflecting the gender imbalance in the school's computer science courses). It is unclear whether the results would be similar in a more gender-balanced class or in a class with different age groups (e.g., younger students or college students). Future studies with larger and more diverse samples could help confirm the robustness of our findings. For example, research involving multiple schools or districts could reveal whether the same benefits are broadly applicable or whether there are contextual factors (such as school culture or student background) that influence the results.

Due to the limitations of the school setting, our study was not a fully randomized experiment. We tried to match the groups based on their prior performance and they were taught by the same instructor, which helps, but we cannot rule out all possible confounding factors. For example, there may be differences in classroom dynamics or unmeasured student characteristics between the two classes that influenced the results. Random assignment or crossover design (where each group experiences both conditions in some order) was not feasible here, but would strengthen causal claims. As it stands, we interpret our results as highly suggestive, but they should be replicated in randomized settings if possible.

A notable limitation is that our research design does not separate the effects of gamification from those of flipped classroom instruction. Both elements were used simultaneously in the experimental group. Therefore, while we can confidently state that the combination works, we cannot clearly state how much of the improvement is due to the flipped model alone (with technology but without game elements) rather than to the addition of game elements. It is possible that a non-gamified flipped classroom would perform better than a traditional one (in fact, the literature suggests that a flipped classroom alone often helps), and it is possible that the gamified approach could also yield some benefits in a traditional classroom. We intentionally combined these in our design based on pedagogical reasoning, but for research purposes, future studies should attempt to separate these factors. For example, four groups could be created: traditional, flipped only, gamified only, and gamified+flipped, to see individual and collective contributions.

We primarily measured cognitive outcomes (test scores) and used platform analytics as a proxy for engagement. We did not administer questionnaires or standardized instruments to assess students' motivational status, self-efficacy, or satisfaction. This is a limitation, as part of our argument is based on improved motivation and engagement, which we infer rather than directly demonstrate. Incorporating such measures in future studies would be beneficial to support the motivational impact of the intervention. For example, an instrument based on the Self-Determination Test (SDT) could measure whether experimental students feel more independent or competent over time than control students.

Our study spanned an 8-week period, which is a fair amount of time, but still relatively short in the grand scheme of learning. We do not know whether the learning was sustained over the long term (e.g., would these students have performed better after a semester or a year without additional support?). Motivation can sometimes wane after the initial novelty; although we did not see any dropout after 8 weeks, a longer course may show different patterns. It would be useful to track whether students in the experimental group continued to use the platform or sought out similar learning experiences later on (we provided access after the studies, but did not track usage beyond 8 weeks).

5.4 Future work

As mentioned, a controlled experiment that separates the individual contributions of gamification and flipped pedagogy would be valuable. This could help educators prioritize interventions, for example, if one has a greater impact than the other, or if their combination is supraadditive (more than the sum of its parts).

Our approach may be applicable to other areas of computer science education (or even beyond computer science). For example, could a gamified flipped classroom improve learning in introductory programming courses (e.g., Python or Java courses)? Some studies, such as [15] and [11], have focused on programming, but further research could explore different concepts (e.g., data structures or algorithms) with similar interventions. In addition to computer science, subjects such as mathematics, science, or language learning could benefit from a hybrid model of structured active learning and gamification. Research in these contexts would broaden the impact of these pedagogical strategies.

It would be interesting to follow up on students who learn using this method to see if it influences their long-term study choices or attitudes. For example, are these students more likely to take additional computer science courses? Do they retain their knowledge better or apply it in new contexts (such as project-based courses or competitions)? We could design a study that measures retention with a surprise test a week or a month later, comparing the decline in knowledge between groups.

We can supplement our study with qualitative research, such as student interviews or classroom observations, to gain deeper insights into students' experiences. For example, interviews can explore which aspects of gamification students found most motivating or how they managed their time in a flipped model. Classroom observations can document interactions and provide context for quantitative data (e.g., noting that experimental students often helped their peers or that control group students often waited for teacher reinforcement).

Given the rise of AI in education, future versions of a platform like ours could include intelligent tutoring system features [33], or even adaptive learning paths based on student performance. Research could investigate whether adding AI-based tutoring or adaptive hint generation would further improve outcomes, especially for those who are struggling. For example, using machine learning to predict when a student is frustrated and then modifying the difficulty or sending an extra motivational message could further personalize gamification. We see that combining AI with gamified reverse learning could be a promising area [23].

In addition to test scores, future work could assess the problem-solving skills, creativity, or collaboration skills fostered by such an environment. The reverse gamified model inherently encourages problem-solving and teamwork; assessing these outcomes (e.g., through project work or complex problem-solving challenges) could provide evidence that these skills are also developing.

6. Conclusion

This study investigated whether the combined use of a gamified learning platform and a flipped classroom methodology can improve learning outcomes in SQL education compared to traditional instructional approaches. The results provide clear evidence that they can. Students in the experimental group – who learned SQL through a flipped classroom model enriched with interactive, game-based tools – achieved significantly higher scores than their peers in the traditional classroom after both four and eight weeks of instruction. These results confirm our initial hypothesis and contribute to the growing body of evidence that thoughtful integration of technology and active learning pedagogies can yield significant benefits in computer science education.

In implementing our approach, we identified several factors that contributed to its effectiveness. Minor modifications to the self-directed learning platform enabled it to effectively support formal flipped classroom instruction. By making learning materials available to students in multiple formats (video lectures, written notes) for pre-class interaction, and by prioritizing class time with a focus on application through educational games and targeted problem solving, we created a more student-centred learning environment. This environment continuously engaged students: they actively listened, thought, and answered checkpoint questions at home; they solved problems, received feedback, and collaborated in class. The engaging, narrative-based style of the learning materials (supplemented with metaphors, reflective questions, and humour) helped maintain student interest over time, preventing the loss of motivation that often accompanies difficult topics. The use of thematic educational games provided contextualized practice opportunities and likely helped demystify abstract SQL concepts by embedding them in familiar contexts. In addition, the automated assessment system ensured that students received meaningful feedback on their work, focusing on semantic correctness, not just syntax – an innovation particularly relevant for a declarative language like SQL. Teachers also benefited from the platform, using real-time data to track progress and quickly address learning gaps, thereby maximizing the effectiveness of classroom interactions.

Despite the positive results, we acknowledge some limitations. The study sample was limited in size and non-randomly distributed, and the research design combined two educational innovations without isolating their individual effects. Therefore, while we can be confident in the success of the combined approach, further research is needed to generalize these findings and analyze the contribution of each component (inverted model vs. gamification). We also note that the gender imbalance in our sample is an area that future work should consider, as diversity in computing education remains an important concern.

In summary, our findings suggest that the proposed instructional design – a combination of gamified learning and flipped classroom pedagogy – can be an effective model for improving outcomes in database education and potentially other areas of computer science education. The approach addresses known challenges in teaching and learning SQL by building on the strengths of active learning, motivational design, and technology-driven feedback. Educators seeking to improve student engagement and performance in similar subjects may find this combined approach a viable and rewarding strategy. By creating an environment where learning is active, personalized, and enjoyable, we not only help students learn better, but also seek to develop more positive attitudes toward the subject, laying the foundation for continued learning and exploration in the field of computer science.

Bibliography

1. Alkhabaz, R., Li, Z., Yang, S., & Alawini, A. (2023). *Student's Learning Challenges with Relational, Document, and Graph Query Languages*. In Proceedings of the 2nd International Workshop on Data Systems Education: Bridging education practice with education research (DataEd '23). Association for Computing Machinery, New York, NY, USA, 30–36. https://doi.org/10.1145/3596673.3596976

2. Taipalus, T., & Perälä, P. (2019). *What to Expect and What to Focus on in SQL Query Teaching*. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 198–203. https://doi.org/10.1145/3287324.3287359

3. Poulsen, S., Butler, L., Alawini, A., & Herman, G. L. (2020). *Insights from Student Solutions to SQL Homework Problems*. In Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20). Association for Computing Machinery, New York, NY, USA, 404–410. https://doi.org/10.1145/3341525.3387391

4. Prabhu, S., & Jaidka, S. (2019). *SQL and PL-SQL: Analysing teaching methods*. In CITRENZ Conference.

5. Xinogalos, S., & Satratzemi, M. (2022). *The Use of Educational Games in Programming Assignments: SQL Island as a Case Study*. Applied Sciences, 12(13), 6563. https://doi.org/10.3390/app12136563

6. Betihavas, V., Bridgman, H., Kornhaber, R., & Cross, M. (2016). *The evidence for 'flipping out': A systematic review of the flipped classroom in nursing education*. Nurse education today, 38, 15–21. https://doi.org/10.1016/j.nedt.2015.12.010

7. Nouri, J. (2016). *The flipped classroom: For active, effective and enhanced learning – especially for low-achieving students*. International Journal of Educational Technology in Higher Education, 13(33), 1–10. https://doi.org/10.1186/s41239-016-0032-z

8. Sobral, SR (2021). *Flipped classrooms for introductory computer programming courses*. International Journal of Information and Education Technology, 11(4), 176–181. https://doi.org/10.18178/ijiet.2021.11.4.1508

9. Miras, J.R., Balsas-Almagro, J.R., & García-Fernández, A. L. (2021). *Using flipped classroom and peer instruction methodologies to improve introductory computer programming courses*. Computer Applications in Engineering Education, 30(1), 133–145. https://doi.org/10.1002/cae.22447

10. Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). *From game design elements to playfulness: Defining "gamification"*. In Proceedings of the 15th International Academic MindTrek Conference (pp. 9–15). ACM. https://doi.org/10.1145/2181037.2181040

11. Swacha, J., & Szydłowska, J. (2023). *Does Gamification Make a Difference in Programming Education? Evaluating FGPE-Supported Learning Outcomes*. Education Sciences, 13(10), 984. https://doi.org/10.3390/educsci13100984

12. Rahman, R. A., Ahmad, S., & Hashim, U. R. (2018). *The effectiveness of gamification technique for higher education students engagement in polytechnic Muadzam Shah Pahang, Malaysia*. International Journal of Educational Technology in Higher Education, 15, Article 41. https://doi.org/10.1186/s41239-018-0123-0

13. Morales-Trujillo, E., & Garcia-Mireles, A. (2020). *Gamification and SQL: An empirical study of student performance in a database course*. ACM Transactions on Computing Education, 20(3), Article 25. https://doi.org/10.1145/3427597

14. Ryan, RM and Deci, EL (2000). *Intrinsic and extrinsic motivations: Classic definitions and new directions*. Contemporary Educational Psychology, 25(1), 54–67. https://doi.org/10.1006/ceps.1999.1020

15. Gündüz, AY and Akkoyunlu, B. (2020). *Effectiveness of gamification in flipped learning*. SAGE Open, 10(4), 1–15. https://doi.org/10.1177/2158244020979837

16. Gao, F. (2024). *Advancing gamification research and practice with three underexplored self-determination theory*. TechTrends, 68(7), 661–671. https://doi.org/10.1007/s11528-024-00968-9

17. Avakyan, EI and Taylor, DCM (2024). *The effect of flipped learning on students' basic psychological needs and its association with self-esteem*. BMC Medical Education, 24, 1127. https://doi.org/10.1186/s12909-024-06113-7

18. Li, L., Hew, K.F., & Du, J. (2024). *Gamification enhances student intrinsic motivation, perceptions of autonomy and relatedness, but minimal impact on competency: a meta-analysis and systematic review*. Education Tech Research Dev 72, 765–796 (2024). https://doi.org/10.1007/s11423-023-10337-7

19. Tahir F, Mitrovic A, Sotardi V (2020). *Investigating the Effects of Gamifying SQL-Tutor*. Virtual confernece: 28th International Conference on Computers in Education. 23/11/2020-27/11/2020. I. 416-425.

20. Alshiha, M. B., & Al-Abdullatif, A. M. (2024). *Gamification in Flipped Classrooms for Sustainable Digital Education: The Influence of Competitive and Cooperative Gamification on Learning Outcomes*. Sustainability, 16(23), 10734. https://doi.org/10.3390/su162310734

21. Zhan, Z., He, L., Tong, Y., Liang, X., Guo, S., and Lan, X. (2022). *The effectiveness of gamification in programming education: Evidence from a meta-analysis*. Computers and Education: Artificial Intelligence, 3, 100096. https://doi.org/10.1016/j.caeai.2022.100096

22. Mellado, R., & Cubillos, C. (2024). *Gamification improves learning: Experience in a training activity of computer programming in higher education*. Journal of Computer Assisted Learning, 40(4), 1959–1973. https://doi.org/10.1111/jcal.13000

23. Holmes, W., Bialik, M., and Fadel, C. (2019). *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning.* Boston, MA: Center for Curriculum Redesign.

24. Thanyaphongphat, J., & Panjaburee, P. (2017). *Effects of Online Learning with Matching Learning Styles and Preferred Digital Media Types on University Students' Perceptions*, 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan, 2017, pp. 552-555, doi: 10.1109/IIAI-AAI.2017.128.

25. Kapp, KM (2012). Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. Hoboken, NJ: Wiley. https://doi.org/10.1145/2207270.2211316

26. Langlands, E., & Morales-Trujillo, M. (2023). *Learning from Errors: An Empirical Study on the Impact of Gamification on SQL Query Formulation*. In Proceedings of the 2023 Innovation and Technology in Computing Education Conference V.1 (ITiCSE '23) (pp. 341–347). ACM. https://doi.org/10.1145/3587102.3588821

27. Lage, MJ, Platt, GJ, and Treglia, M. (2000). *Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment.* Journal of Economic Education, 31(1), 30–43. https://doi.org/10.1080/00220480009596759

28. Bishop, J. L., & Verleger, M. A. (2013). *The Flipped Classroom: A Survey of the Research*. 120th American Society for Engineering Education Annual Conference and Exposition, 30, 1-18.

29. Knutas, A., Herala, A., Vanhala, E., & Ikonen, J. (2016). *The flipped classroom method: Lessons from flipping two programming courses*. In Proceedings of the 17th International Conference on Computer Systems and Technologies (CompSysTech '16) (pp. 423–430). ACM. https://doi.org/10.1145/2983468.2983524

30. Findlay-Thompson, S. and Mombourquette, P. (2014*). Evaluation of a Flipped Classroom in an Undergraduate Business Course*. Business Education & Accreditation, v. 6 (1) p. 63-71, 2014, Available at SSRN: https://ssrn.com/abstract=2331035

31. Romano, Jeanine & Kromrey, Jeffrey. (2006). *Appropriate Statistics for Ordinal Level Data: Should We Really Be Using t-test and Cohen's d for Evaluating Group Differences on the NSSE and other Surveys?*

32. Koedinger, K., Aleven, V., Roll, I., Baker, R. (2009) *In vivo experiments on whether supporting metacognition in intelligent tutoring systems yields robust learning*. Graesser, A., Hacker, D. (Eds.), Handbook of Metacognition in Education, 383-412.

33. Huang, B., Hew, K.F., and Lo, C.K. (2018). *Investigating the effects of gamification-enhanced flipped learning on undergraduate students' behavioral and cognitive engagement*. Interactive Learning Environments, 28(8), 1106–1126. https://doi.org/10.1080/10494820.2018.1495653

34. Davies, R.S., Dean, D.L. & Ball, N. (2013). *Flipping the classroom and instructional technology integration in a college-level information systems spreadsheet course*. Education Tech Research Dev 61, 563–580. https://doi.org/10.1007/s11423-013-9305-6

Authors

BALLA Tamás
Eszterházy Károly University, Faculty of Informatics, Institute of Mathematics and Informatics, Department of Computational Science, Eger, Hungary,
Eötvös Loránd University, Doctoral School of Informatics, Budapest, Hungary
e mail: balla.tamas@uni-eszterhazy.hu

KIRÁLY Sándor
Eszterházy Károly University, Faculty of Informatics, Institute of Mathematics and Informatics, Department of Information Technology, Eger, Hungary
e mail: király.sandor@uni-eszterhazy.hu

About the document

Published here:

CENTRAL EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 7, Number 1. 2025.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.7.1.12228

License