The Effects of Adding Non-Compulsory Exercises to an Online Learning Tool on Student Performance and Code Copying

CÉSAR DOMÍNGUEZ, ARTURO JAIME, JÓNATHAN HERAS, and FRANCISCO J. GARCÍA-IZQUIERDO, Universidad de La Rioja

This study analyzes the impact of adding a review exercises module to an online tool used in a software engineering degree program. The objective of the module is to promote students' self-learning effort to improve their performance. We also intend to determine if this new feature has any effect on the amount of code copies detected in lab sessions when using the same online tool. Two groups of students were compared quantitatively: the first group used the tool exclusively during lab sessions, whereas the second group had the option of employing the tool's new module to enhance their study. The tool allows us to collect interesting data related to the focus of this research: supplementary work completed voluntarily by students and the percentage of students copying others' code during compulsory lab sessions. The results show that the students in the second group achieved better academic results and copied less in lab sessions. In the second group, the students who invested more effort in doing revision exercises and copied less in lab sessions obtained better results; and, interestingly, the effort invested in completing review exercises did not seem to compensate for the learning effort avoided by copying others' exercises during lab sessions. The results show the advantages of a tool used with a dual orientation: compulsory and voluntary. Mandatory usage in lab sessions establishes some milestones that, eventually, act as an incentive fostering learning, while voluntary use reinforces students' perception of the tool's usefulness in terms of learning.

CCS Concepts: Software engineering education; Computer-assisted instruction; E-learning

Additional Key Words and Phrases: Online learning tool, non-compulsory exercises, academic performance, code copying

ACM Reference format:
César Domínguez, Arturo Jaime, Jónathan Heras, and Francisco J. García-Izquierdo. 2018. The Effects of Adding Non-Compulsory Exercises to an Online Learning Tool on Student Performance and Code Copying. *ACM Trans. Comput. Educ.* 19, 3, Article 16 (November 2018), 22 pages.
https://doi.org/10.1145/3264507

This work was partially supported by by the Ministerio de Economía, Industria y Competitividad, project EDU2016-79838-P. We would like to thank to Diego Díez Ricondo and Juan-Francisco Diez Léglise for the development of the SQL learning tool Aplicación BD as part of their final degree projects. We also thank the anonymous reviewers whose comments have greatly improved this manuscript.

Authors' address: C. Domínguez, A. Jaime, J. Heras, and F. J. García-Izquierdo, Dpto. de Matemáticas y Computación de la Universidad de La Rioja. Ed. Vives, Luis de Ulloa s/n, 26004 Logroño, Spain; emails: {cesar.dominguez, arturo.jaime, jonathan.heras, francisco.garcia}@unrioja.es.

1946-6226/2018/11-ART16 $15.00
https://doi.org/10.1145/3264507

1 INTRODUCTION

Online tools that include automatic feedback or assessment exemplify how IT tools can be utilized in teaching environments. These tools can be made available to students to practice outside of class and they provide a significant advantage over unassisted studying [46]. Recently, the use of this type of online tool as complementary and non-compulsory material has been analyzed in disciplines such as mathematics [4, 23], language learning [21, 57] and medicine [44]. The learning effort that students make when using these tools is unrelated to the assessment process. The results obtained agree that such effort has clear benefits for academic performance. Another different and common use of online tools in education is to support in-class laboratory sessions. Although in this context students should be focused on learning, the need to develop deliverables during the session can shift their focus. The grade for the deliverables may have some influence on the grade for the course [3], and this factor may encourage some students to improve their deliverables by options other than study and exploration with the tool. There is not a consensus in the literature regarding the effectiveness of the use of these tools in laboratory sessions in terms of academic results, since the risk of students copying exercises also exists [46]. Thus, instead of investing the time reserved for the lab session in learning with the tool, students concentrate on copying answers to exercises in order to achieve better scores [3, 4]. The fact that students are not very clear on what study practices are acceptable [22], combined with the fact that the Internet facilitates this type of copying (since exercise results can be easily shared among classmates [25, 46]), pose a threat to the adoption of e-learning practices in higher education, as indicated in the work of Hart and Friesner [25].

One strategy to remedy inappropriate academic practices proposes following a teaching-and-learning approach [40]. This approach focuses on creating an atmosphere of participation and interest, while downplaying the detection and punishment of students' poor habits. Following this approach, academic performance increases and inappropriate practices seem to decrease [10, 40]. To accomplish this goal, technology can be employed to enhance learning tasks or diversify instruction, make learning more relevant and purposeful, and greatly reduce the need for or temptation of misconduct [5, 29]. In the same line of thought, Anderman and Midgley [2] identify a relationship between classroom environment and cheating. Students' perception of a mastery goal structure (which emphasizes learning, improvement, effort, and understanding as important reasons for engaging in academic work) leads to positive academic outcomes and is related to less cheating [2, 12]. Hence, the teaching-and-learning strategy is clearly supported by the literature [40, 10]. Nevertheless, only a few quantitative studies have been conducted, particularly in the case of engineering students [6], that demonstrate the validity of this teaching-and-learning strategy. The study presented herein aims to address this knowledge gap. Instead of trying to use technology to catch students cheating, in order to penalize them, this study intends to provide the students, by means of technology, with mechanisms that allow them to easily enhance their level of dedication to the subject.

This article presents an online tool for learning SQL developed by the authors. The tool was used during two consecutive time periods in a relational database course during the first year of a Software Engineering degree program. During the first period, the tool provided support for the compulsory lab sessions. In the second period, the use of the tool was expanded to include a sufficiently ample non-compulsory set of exercises of the same level of difficulty as those exercises completed during lab sessions.

The first aim of the present study is to analyze the impact—in terms of academic performance and code copied during labs—of incorporating a new module designed to facilitate self-learning through an online tool. Therefore, we propose the first research question: *RQ1. What is the* *relationship between adding a non-compulsory set of exercises to an online learning tool and both student performance and the rate of code copying in compulsory labs that use the same tool?*

In addition, the literature suggests that female students work harder and more consistently and that they are less likely to cheat than males [2, 25, 36, 42, 48]. The second objective of this study is to examine the effect of the review exercises on men and women. Hence, a second research question is formulated: *RQ2. Did the new set of exercises have a different relationship with student performance and code copying depending on gender?*

In the literature, numerous studies consider either learning effort (see, for instance, [48] or [53]) or code copying (see, for instance, [43] or [46]). All of these studies conclude that students who review more or copy less achieve better academic results. However, to the best of our knowledge, the effect of extra effort combined with code copying on learning results has yet to be considered. The use of a software tool that monitors extra effort and code copying allows us to gather data to analyze such a combination. Thus, we establish our third research question: *RQ3. What is the relationship between the combination of extra effort and code copying and learning results?*

Finally, students' very positive satisfaction with the use of SQL learning tools is thoroughly documented in the literature [9, 13, 38, 47, 51]. However, an increase in learning effort does not always translate into an increase in satisfaction; in fact, it can have the opposite effect [28]. The final objective of this study consists of analyzing the influence of the new module on student satisfaction. Hence, our final research question is: *RQ4. What is the relationship between the new module and student satisfaction with the tool?*

2 APLICACIÓN BD: AN SQL LEARNING TOOL

We have developed Aplicación BD (which stands for Database Application in Spanish), a new online tool that assists students in the process of learning the SQL query language. SQL is the predominant language for defining and manipulating relational databases. SQL has a simple syntax with a limited set of commands that can give the impression of simplicity. However, it is possible to create very complex queries with powerful results [38]. Learning to write SQL queries is an onerous task and students struggle with a number of difficulties [14]. Hence, using a tool to help them in the learning process is a sensible idea [47]. Aplicación BD was designed and tested in collaboration with students. Pilot testing of the tool was conducted during several lab sessions throughout the 2011 academic year.

Before introducing Aplicación BD into the database course, students had access to the CMS Blackboard where a document with the statement of the exercises and the expected results of the queries were available, along with an image with the database schema to employ, and a script to create the database and populate it with data. During lab sessions, students employed a visual database tool, like MySQL Workbench or SQL Server Management Studio, to solve the exercises. With the help of such a tool, students created and populated the database. Subsequently, and based on both the database schema and the expected results of the queries, students could test out their ideas to solve the exercises. Finally, the students generated a text document with their solutions and sent it via Blackboard. In order to assess the progress made during lab sessions, instructors had to download the documents submitted throughout the lab sessions and grade them manually [16]. Aplicación BD integrates the services provided by Blackboard to submit materials and collect exercises completed by students, as well as the services provided by the visual database tool for testing queries. In addition, Aplicación BD facilitates the instructors' work of preparing exercises and organizing lab sessions. Lab sessions are designed to be conducted in a classroom with personal computers and an instructor present. The tool allows the instructor to start the lab session and establish its duration. When the time limit is up, the tool automatically collects the exercises completed by students that have not yet been submitted. Aplicación BD can also be employed by

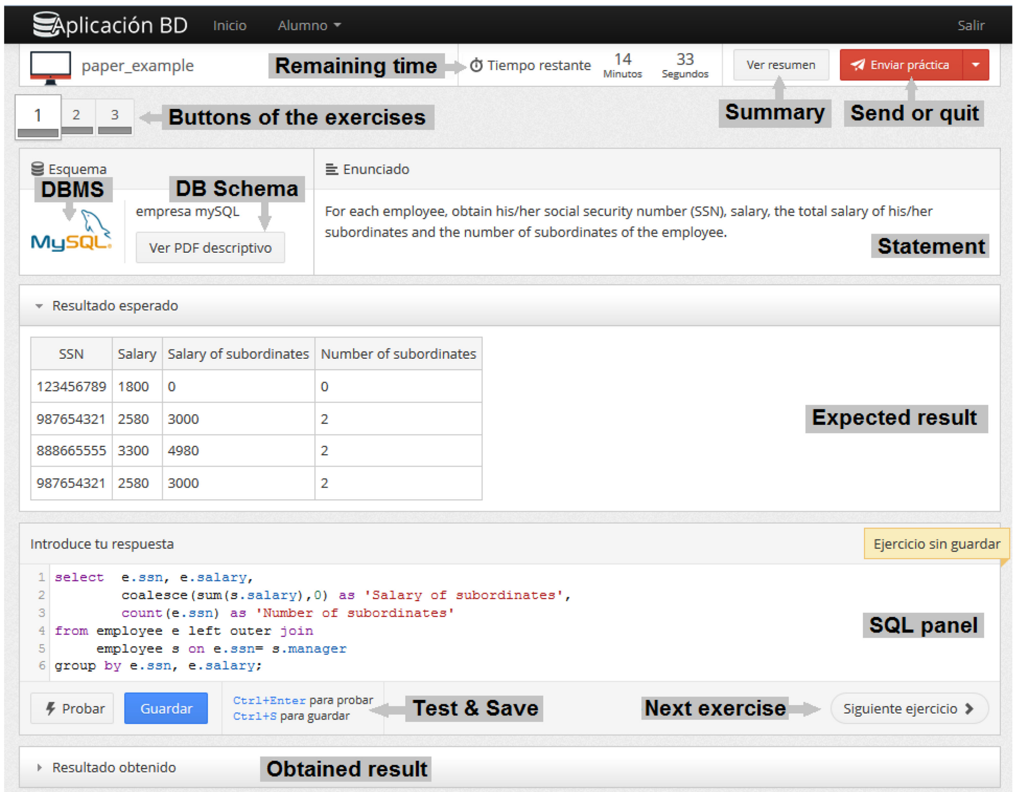

Fig. 1. Annotated student view of the aplicación BD interface for lab sessions.

students to complete review sessions. The review exercises are selected by the tool based on a set of options such as difficulty level or type of query.

2.1 Interface for Students

Currently, students perform the lab-session exercises using the interface presented in Figure 1. The time remaining appears at the top of the interface, next to a button to download a summary of the exercises and solutions completed up to that moment, and a button to submit exercises or quit the lab session. The numbered buttons access different exercises. The interface also shows the database management system (DBMS), e.g., MySQL, SQL Server, or ORACLE, where the database of the exercise resides. The button to the right of the DBMS logo allows the user to download an image with the database schema associated with the exercise. The exercise statement is displayed on the right panel. The rest of the interface consists of three panels. The first panel shows the expected result. The second panel is a text area to write the SQL query, test it, and save it (test and save buttons). The third panel displays the result after executing the query or an error message.

2.2 Application Query Analyzer

Another goal that led to the development of Aplicación BD was to provide students with a tool that facilitates the task of checking if their responses to exercises match the ideal responses provided by the instructor, without revealing the latter. Thus, when the execution of the query is free of syntactical errors, Aplicación BD checks to see if the student's query result and the expected query result match. This process consists of a string comparison of the values of both result sets, taking

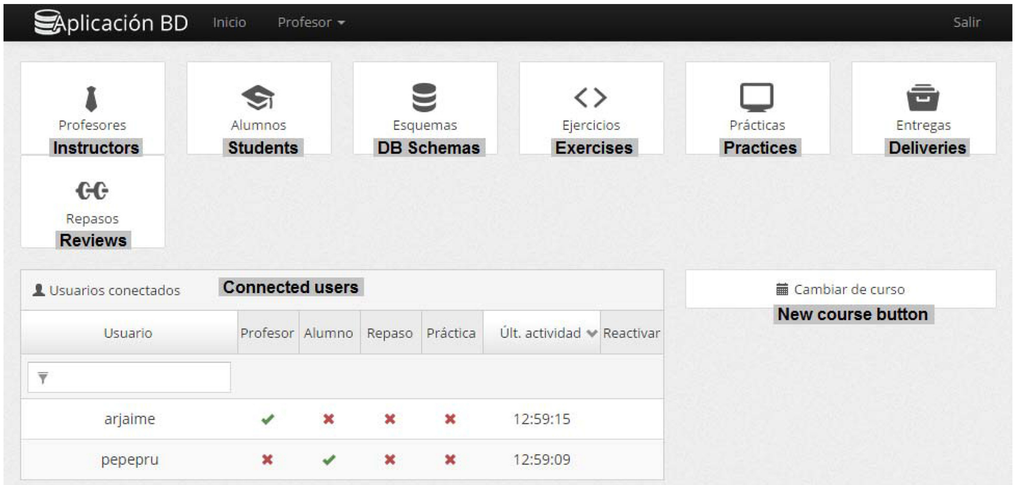

Fig. 2. Annotated interface of Aplicación BD with control panel for instructors.

into account the row order in those exercises that require the use of the ORDER BY clause. A mere comparison could overlook some student misconceptions, such as the incorrect use of GROUP BY or DISTINCT, the use of unnecessary tables, or unnecessary or excessive subqueries. Consequently, a help system has been incorporated into the tool to draw students' attention to this kind of error that cannot be detected by the DBMS. Based on the entire query analysis, the tool marks the responses to the exercises with a "not match/match" label and eventually displays a set of hints to help students correct the aforementioned common errors.

2.3 Interface for Admin-Instructors

The interface for admin-instructors is presented in Figure 2. This interface allows the instructor to create different types of users (e.g., instructor or student). It is possible to create multiple student accounts from information in a text file. In addition, new database schemas can be included with the necessary connection information.

Using the interface depicted in Figure 2, the admin-instructor can also create exercises. The exercises introduced in Aplicación BD must be associated with one of the available schemas in the tool, and they must include both the statement and the SQL solution. The exercise must be classified by means of two values: difficulty (easy, medium, or hard) and the specific lab session wherein the student will learn how to solve it. In addition, the instructor also establishes whether the exercise is visible for review sessions, or if it is only available for lab sessions. Just as several student-users can be created, a feature is also available to load several exercises from a text file.

A lab session consists of a set of exercises selected by the instructor from those available in the system. Each session must have a unique name. The tool allows the instructor to choose the order of the exercises.

The instructor can download students' lab session deliverables. For each lab session, the system generates a PDF document with the students' deliverables completed during the session. Such a document includes the mark (not match or match) assigned to each exercise based on the abovementioned query analyzer. Instructors assess student performance by revising said PDF documents.

As shown in Figure 2, the interface also displays the users connected to the system, and indicates whether they are working on a lab session or a review session.

In addition, by accessing the information stored in the tool's database, the solutions sent by different students can be compared and incidences of copying can be detected (a basic comparison of queries is performed after removing whitespaces and separation characters like tabs or end-of-line characters). With this information and the delivery timestamp, originals and copies can be identified.

2.4 Other SQL Learning Tools

Teaching SQL through software tools is a widely popular idea. Different solutions have been developed to facilitate the learning process. Usually the tools for teaching SQL provide a simple environment to write and test SQL instructions against databases, and give more immediate and informative feedback than that provided by a DBMS [14]. Let us mention the following tools: ActiveSQL [46], SQLator [47], SQL-Tutor [38], SQLify [14], AsseSQL [45], SQL-Trainer [33], XDa-TA [8], ADVICE [13], Acharya [7], LEARN-SQL [9], SQL-ACME [51], SQL-LES [26], SQL-LTM [15] and SQLZOO [52]. Despite the broad range of tools, it seems that none of them has achieved significant visibility. The aforementioned tools all share two common features:

1. *Checking the Correctness of Queries.* Almost all the systems compare the result of executing the student's query against a solution stored in the tool (Aplicación BD, SQLator, AsseSQL, SQL-LES, or SQL-ACME). ActiveSQL, ADVICE, and SQLZOO also mark the rows and columns where the result does not match the expected result. Instead of showing the expected result, SQLify indicates the degree of correctness of the instruction (correct, largely correct, seem largely correct, and so on). LEARN-SQL allows the user to introduce several database states and compares the results for all of them. XDa-TA goes a step further and automatically generates several database states with the same aim. SQL-Tutor verifies that a query without subqueries satisfies a set of constraints before deeming it correct. SQL-Trainer also employs constraints to check the correctness of queries; however, the number of constraints is smaller than in SQL-Tutor. SQLify can confirm if two conjunctive queries are equivalent. Acharya normalizes the 'where' clause of a query without subqueries by means of logically equivalent clauses before comparing instructions. SQL-LTM achieves the same result by means of an XML representation of the query, where several transformation patterns are applied. In the latter system, the comparison of the student's query against the stored query is based on metrics extracted from the aforementioned representation.
2. *Reviewing Exercises.* All the aforementioned systems incorporate a set of exercises that is available to students for voluntary study. In fact, some systems, such as SQL-Trainer or SQLZOO, were developed solely for this purpose.

As shown in Table 1, other features are only available in certain tools:

1. *Data Description and Data Manipulation Languages (DDL and DML).* All the systems were developed to learn SQL queries, but only some of them allow users to practice with instructions to create, delete, and modify tables (DDL) or to insert, delete, and modify rows (DML).
2. *Hints.* Some systems, like Aplicación BD, show hints to students that go beyond the messages generated by the DBMS. ActiveSQL measures the quality of the instruction by means of heuristics such as the length of the instruction, the use of 'distinct' or the use of 'like' comparison instead of '='. SQL-Tutor provides incremental guidance through its system of constraints. This tool has recently enhanced its help system by offering positive feedback instead of error feedback [39]. SQL-Trainer also provides hints, but it is based on a more

Table 1. Comparison of SQL Learning Tools

SQL tool	DDL	DML	Hints	Copies	Lab sessions	Available
ActiveSQL [46]	–	Yes	Yes	Yes	–	Yes
SQLator [47]	–	Yes	–	Yes	–	–
SQL-Tutor [38]	–	Yes	Yes	–	–	–
SQLify [14]	–	Yes	–	–	–	–
AsseSQL [45]	–	Yes	–	–	–	–
SQL-Trainer [33]	–	Yes	Yes	–	–	–
XDa-TA [8]	–	Yes	–	–	Yes[1]	–
ADVICE [13]	Yes	Yes	–	–	Yes	Yes[2]
LEARN-SQL [9]	Yes	Yes	–	–	Yes[1]	–
SQL-ACME [51]	Yes	Yes	–	–	Yes	–
SQL-LES [26]	–	Yes	–	–	Yes	–
SQL-LTM [15]	–	Yes	Yes	–	–	–
SQLZOO [52]	–	Yes	–	–	–	Yes
Aplicación BD	–	Yes	Yes	Yes	Yes	Yes[2]

[1]Through a CMS; [2]Upon request.

limited set of constraints. SQL-LTM uses its transformation patterns to provide this kind of help. Finally, Oscar [34] is an intelligent tutor with a natural language interface that answers questions and gives hints by adapting itself to each student's learning style.

3. *Code copies.* There are some systems concerned with detecting code copies. Aplicación BD and SQLator detect copies by comparing the text of the instruction after some changes such as removing whitespaces, or by being case-insensitive. ActiveSQL also removes unnecessary parenthesis, alias, changes in the case clauses, commutativity in equalities and so on.
4. *Lab Session Management.* ADVICE, SQL-ACME, SQL-LES, and Aplicación BD allow instructors to manage face-to-face sessions for groups of students. Other systems like XDa-TA and LEARN-SQL obtain this functionality through integration with a CMS.

In addition to these functionalities, some tools provide more specific features. For instance, eSQL [32], ADVICE, and SAVI [11] are focused on learning the query execution process. In order to achieve this functionality, these tools visualize how different operators take part in the process to obtain a result. The system SQL-LES incorporates a model that allows the user to identify the complexity of a query [27]. This might be helpful in selecting a set of exercises of a similar level of difficulty. SQLify offers the possibility of conducting peer reviews. SiS [20] allows the user to build a query (without subqueries) in small steps using a graphical user interface that is closely coupled with the textual translation of the query. SQL-ACME facilitates tracking student work in order to detect common mistakes, and includes a communication channel for tool users. Finally, the authors of [50] suggest using adaptive game-based learning.

3 MATERIALS AND METHOD

3.1 The Relational Database Course

The study was conducted in a relational database course during the first year of a Software Engineering degree program. Learning SQL is a key objective of the course: the 60 hours of coursework include 9 two-hour lab sessions and 18 one-hour theoretical lessons devoted to this topic.

The continuous assessment of this course consists of three parts:

1. *Tests.* Three tests that include 6 to 10 questions about theoretical concepts covered during the course. The first test deals with the relational model. Students take this test before starting the SQL lab sessions. The second and third tests deal with other aspects of databases not related to SQL, such as characteristics and advantages of the database approach, user roles, database management system components, basic concepts of relational algebra, etc. These three tests constitute 10% of the final grade.
2. *Lab Sessions.* The set of exercises completed by students individually during lab sessions. Each student completes a total of 70 exercises. Three different sets of exercises are created, one for each subset of approximately 20 students. Aplicación BD is employed in seven lab sessions about queries. The other two lab sessions are devoted to DDL and DML instructions. The latter sessions employ the traditional method, which combines the CMS Blackboard and an SQL console (e.g., MySQL Workbench) to work on a DBMS like MySQL. It is quite easy for students to copy each other's exercises during lab sessions [46]. However, our study followed the same approach as Morris [40] or Carroll [10], and instead of focusing on catching students copying, lab sessions were designed as a way for students to learn and to prepare for SQL exams (which constitute a large portion of the final grade). Lab sessions represent 10% of the final grade, and to a certain extent, this rather small percentage is intended to safeguard against unfair grades that could result from students copying, which still occurred, despite the minor impact of lab sessions on students' final grades [49].
3. *SQL Exams.* Three written exams, given at different points throughout the course; where students solve SQL exercises. This part constitutes 80% of the final grade. Students receive early feedback about their correct answers and detected errors.

3.2 Research Design

In order to analyze the research questions proposed in this study, two groups of students were compared. Group 1 includes students from the 2012 and 2013 academic years, and group 2 includes the students from the 2014 and 2015 academic years. The course instructors were the same for both groups of this quasi-experimental design, and the same difficulty was maintained across all the assessments. The teaching method changed in just one aspect: for the second group, a set of review exercises about SQL queries was included in Aplicación BD, thereby allowing students to use this application to enhance their studying. These exercises were always available for group 2 and completing them was optional. The same tool was employed by both groups to complete the lab sessions. The set of voluntary exercises for group 2 consisted of 123 exercises (23 difficult exercises, 46 of a medium level of difficulty, and 54 easy exercises). In addition to these exercises, 113 exercises were prepared for the lab sessions (29 difficult exercises, 22 of a medium level of difficulty, and 62 easy exercises). This last set of exercises was not available outside of the lab sessions. Each exercise was associated with one of the four database schemas that were deployed both on the MySQL and SQL Server DBMS. The review exercises were designed to demand a certain level of knowledge from students, and were released on Aplicación BD as the instructors introduced the concepts required to solve them in class. These review exercises were intended to supplement the exercises and assignments given by the instructors throughout the course, which were different from those included in the lab sessions or SQL exams, though they had the same level of difficulty.

The variables to be analyzed include the three types of assessment explained in the previous section. The tests grades were employed as a control variable to estimate how similar or different the students were since the tests did not change over the course of this study. The lab-sessions grade and the grades from the three written exams were examined. All grades range from 0 to 10. Students who did not complete all three types of assessment were excluded from the sample.

In order to compute the percentage of copied exercises, only those from the laboratory sessions were considered. A student's exercise was considered copied if it completely matched (except characters like white spaces, tabs, or end-of-line) another exercise previously submitted by another student who attended the same lab session group (different groups were given different exercises). This type of code copy, which is used in other SQL learning studies [47], will be referred to from now on as a *complete copy*. Obviously, there are more advanced mechanisms to detect copies (see, for instance, [37]), but our experience reviewing exercises in previous years indicates that this measure of similarity is sufficient for our purposes. Over the years, it has been observed that students who copy use this type of complete copy in laboratory sessions, without bothering to come up with more sophisticated strategies. We did not seek to punish the students who did such copying. Therefore, this general and operational definition of copying was considered the least restrictive for our research purposes. On the other hand, the fact that code-similarity is a common phenomenon in software development must be taken into account [56]. Copying is the most important cause of code-similarity in programming assignments, but other causes include software reuse, coding style or collation. Therefore, we deemed these copies acceptable and did not try to punish students for them. Indeed, given that SQL is an artificial language, it is possible to see overlap between responses even in cases where copying has not occurred. For this reason, complete copies occurring during the first lab session were discarded, because the exercises were easy and very short, and the possibility of solving them with the same answer was extremely likely. Finally, to assess students' extra effort, the number of self-initiated review sessions completed was utilized.

Furthermore, a survey evaluating student satisfaction with the tool was given to students during the last lab session. This questionnaire was designed based on questionnaire items employed by authors of other satisfaction surveys for similar SQL systems [9, 38, 51]. The wording was reviewed by a group of instructors and experts. The items were assessed with a Likert scale from 1 to 4 (labeled as completely disagree, disagree, agree, and completely agree). There is no middle option, creating a forced choice method. Although the typical five-level Likert scale is a bipolar scaling method including a neutral option "Neither agree nor disagree," this option can be viewed as an easy option to resort to when a respondent is unsure and it is very open to interpretation. Hence, whether it is truly a neutral option is questionable [41].

3.3 Statistical Analyses

A quantitative study is included in accordance with the recommendations of López et al. [35] to answer the four research questions proposed in the introduction section.

The first research question requires analyzing the relationship between incorporating a set of non-compulsory exercises and both student performance and the percentage of exercises copied in laboratory sessions. In this analysis, Student's t-tests were used. The independent variable was the use and non-use of additional exercises and the dependent variables were the grades obtained on the different assessments (tests, exams and labs) and the percentage of copies.

The second research question involves studying the above relationship in terms of gender. A similar study was proposed for each group of students. For the analysis of group 1, Student's t-tests were also used, gender being the independent variable and the dependent variables were the same as in the previous analysis. The analysis of group 2 was equivalent to that of group 1, but the number of revisions was added as a dependent variable.

The third research question requires studying the relationship between combining extra effort and code copying and the academic performance of group 2. First, Pearson correlation coefficients were used to correlate the dependent variables used in the previous analyses (no causality can be inferred from these correlations). Next, three analyses were proposed. For the first one, students were classified beforehand according to whether they had a higher or lower percentage of copies than the median. Student's *t*-tests were proposed with the aforementioned classification as the independent variable and the grades obtained in the different assessments as dependent variables (tests, exams and labs). The second analysis was similar to the previous one, but the independent variable was a classification of the students according to whether they had done more or less review sessions than the median. Finally, a 2×2 student classification [24] was created from the classifications used in the two previous analyses. ANOVA tests were conducted (including Bonferroni corrections to analyze each pair of data sets included in the ANOVA), the last classification was the independent variable and the dependent variables were those of the two previous analyses.

The fourth research question involves studying the relationship between the new set of exercises and student satisfaction. In this analysis, Student's *t*-tests were used. The independent variable was the use or non-use of the additional exercises and the dependent variables were the items on the survey.

Parametric conditions were verified prior to using these tests, and when parametric conditions were not fulfilled the corresponding non-parametric tests (i.e., Mann–Whitney U test, Spearman Rho correlation test, or Kruskal–Wallis test) were also acceptable. The controlled variables included the instructors; the tool; the difficulty of the questions on the tests, exams and labs; and the compulsory use of the tool in labs. The confounding variables included the differences in student capabilities between the groups. In particular, we used the test grades (which were taken prior to the inclusion of SQL, or did not include SQL concepts), grade point average before starting university, and studies completed before starting university. Finally, a chi-square test was utilized to study the distribution independence for categorical data. The effect size was interpreted following Cohen's effect size benchmarks [18], which can help determine whether the observed differences or correlations are meaningful.

3.4 Participants

A total of 236 students participated in this study. Group 1 consisted of 123 students, and group 2 consisted of 113 students. From the total number of students, 101 students from the first group (84% males) completed the assessments, and 99 (77% males) from the second group. This means that the percentage of students that did not complete the assessments (drop-out students) from the first group was slightly higher (18% versus 12%; $\chi^2 = 1.377$, $p = 0.241$). Drop-out students were excluded from this study. Of the students included, there are no significant differences in the distribution of males and females between the two groups ($\chi^2 = 0.741$, $p = 0.389$). The mean (standard deviation) of the students' grade point average before starting university is 6.46 (1.03) for the first group and 6.55 (0.93) for the second. In the first group, 69 (68%) students had finished high school before starting the degree program, 20 (20%) already had other university degrees, and 12 (12%) had some kind of vocational education. In the second group 71 students (72%) had finished high school before starting the degree program, 16 (16%) already had other university degrees and 12 (12%) had some kind of vocational education. There are no significant differences in the students' grade point averages before starting university ($U = -0.946$, $p = 0.344$) with small size effects (Cohen's $d = 0.09$), or in the distribution of type of university access ($\chi^2 = 0.453$, $p = 0.797$); these variables can be considered confounding. With this sample size, and fixing the alpha significance criterion to 0.05 and a medium effect size of 0.5 for a two-tailed *t*-test to obtain the difference between two independent means, a statistical power of 0.94 [19] was obtained, which is considered a large statistical power superior to the 0.80 recommended minimum [18].

Table 2. Mean (Standard Deviation) of the Grades for the Different Assessments, and the Percentage of Complete Copies in Lab Sessions for Both Groups

	N	Tests	Labs	Exam 1	Exam 2	Exam 3	% of copies
Group 1	101	5.34 (2.02)	7.23 (2.24)	4.18 (2.85)	3.69 (2.89)	2.94 (2,98)	27.66 (22.54)
Group 2	99	4.93 (2.03)	7.98 (2.05)	5.16 (2.70)	4.81 (2.69)	5.03 (3.11)	19.90 (18.35)
Statistic		$t = 1.410$ $p = 0.197$	$Z = -2.582^{*}$ $p = 0.01$	$Z = -2.749^{**}$ $p = 0.004$	$t = -2.731^{**}$ $p = 0.007$	$Z = -4.293^{***}$	$Z = -2.950^{**}$ $p = 0.003$

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, Statistics: Student's t-test, Mann–Whitney U test.

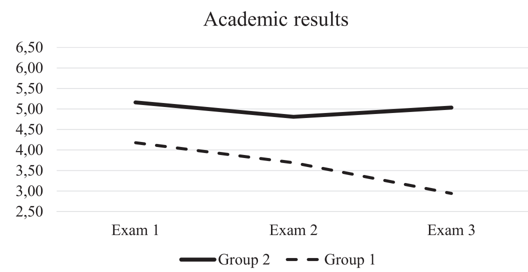

Fig. 3. Comparison of exam grades between the groups studied.

4 FINDINGS AND ANALYSIS

4.1 Academic Performance and Code Copies

This section describes the findings related to the analysis of the relationship between academic performance and code copies. Table 2 contains the results obtained on the assessments along with the percentage of complete copies detected in lab sessions for both groups.

No differences in the test grades related to theoretical concepts are observed. This indicates that the students' level was similar across both groups. However, once review exercises were introduced, the percentage of completely copied exercises decreased significantly and the lab session grades improved. Finally, the grades for the three SQL exams also improved considerably with significant differences across all of them. In addition, it should be noted that the mean grade for these three exams progressively decreased in group 1, while it remained steady in group 2. Both trends are displayed graphically in Figure 3. This occurred in spite of the progressive increment in difficulty of the exercises included in the three exams. These results show that after incorporating the set of review exercises, academic results improved considerably, and a significant reduction in the percentage of complete copies was observed.

4.2 Gender Analysis

This section describes the findings regarding the relationship between academic performance and code copies depending on gender. Table 3 includes, for each group of students separated by gender, the results obtained on the tests, lab sessions, and SQL exams, and the percentage of copies as well. For group 2, the mean number of review sessions completed is included. In both groups, the percentage of copies is higher among men, but without significant differences (Cohen's $d = -0.08$, low size, for group 1; and Cohen's $d = -0.43$, low-medium size, for group 2).

Women completed more reviews than men, but without significant differences (Cohen's $d = -0.46$, low-medium size). However, the introduction of review exercises generated a more pronounced reduction in the percentage of copies among women: 49% in women (Cohen's $d = -0.60$,

Table 3. Mean (Standard Deviation) of the Grades Obtained on the Different Assessments and the Percentage of Copies for the Groups Studied, Broken Down According to Gender

	N	Tests	Labs	Exam 1	Exam 2	Exam 3	% of copies	Reviews
Group 1								
Men	84	5.24 (2.08)	6.93 (2.23)	3.96 (2.76)	3.57 (2.92)	2.88 (3.03)	28 (22.16)	–
Women	17	5.82 (1.68)	8.73 (1.63)	5.23 (3.11)	4.26 (2.74)	3.23 (2.86)	26.12 (24.86)	–
Statistic		$t = -1.092$ $p = 0.278$	$t = -3.162^{**}$ $p = 0.002$	$Z = -1.503$ $p = 0.133$	$t = -0.895$ $p = 0.373$	$t = -0.438$ $p = 0.662$	$Z = 0.610$ $p = 0.542$	–
Group 2								
Men	77	4.78 (2)	7.94 (2.09)	5.16 (2.77)	4.82 (2.59)	4.94 (2.97)	21.98 (23.58)	18.4 (18.65)
Women	22	5.43 (2.11)	8.13 (1.94)	5.17 (2.59)	4.79 (3.09)	5.32 (3.64)	13.35 (16.76)	31.41 (35.38)
Statistic		$t = -1.263$ $p = 0.216$	$Z = -0.301$ $p = 0.764$	$Z = -0.044$ $p = 0.965$	$t = 0.044$ $p = 0.965$	$t = -0.458$ $p = 0.648$	$Z = 1.269$ $p = 0.204$	$Z = -1.012$ $p = 0.312$

$^{**}p < 0.01$, Statistics: Student's t-test, Mann–Whitney U test.

Table 4. Correlations among the Different Graded Assessments, Percentage of Complete Copies, and Number of Review Sessions Completed by Students in Group 2

	Tests		Labs		Exam 1		Exam 2		Exam 3		Reviews	
	Rho	***p***	**Rho**	***p***	**Rho**	***p***	**Rho**	***p***	**Rho**	***p***	**Rho**	***p***
Labs	0.340^{**}	0.001										
Exam 1	0.260^{*}	0.01	0.231^{*}	0.024								
Exam 2	0.409^{***}		0.472^{***}		0.446^{***}							
Exam 3	0.422^{***}		0.307^{**}	0.004	0.449^{***}		0.649^{***}					
Reviews	0.328^{**}	0.001	0.208^{*}	0.041	0.131	0.201	0.321^{**}	0.002	0.244^{*}	0.025		
% of copies	-0.402^{***}		-0.122	0.274	-0.140	0.209	-0.357^{***}		-0.411^{***}		-0.324^{**}	0.003

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, Statistics: Spearman's Rho correlation coefficient.

medium-large size), and 22% in men (Cohen's $d = -0.26$, low size). Although in general women's grades are higher across all the assessments and in both groups, there are no significant differences.

The results of the gender analysis show that women copied less and obtained better results, but without significant differences.

4.3 Influence of Reviewing on Academic Performance and Copying

This section describes the findings related to the analysis of the relationship between the combination of extra effort and code copying and the learning results. This part of the study was conducted exclusively on group 2. Table 4 contains the correlation coefficients among the results obtained on the tests, lab sessions, SQL exams, the percentage of complete copies, and the number of review sessions completed.

It is worth mentioning that all the correlation coefficients with the percentage of complete copies are negative. In addition, these correlations are significant in all cases except for lab session exercises, where copying was not penalized, and in the first SQL exam as well, which was the easiest one. Regarding the SQL exams, the negative correlation coefficient is higher when the difficulty of the exam increased (effect size is medium-large in the most difficult exams). On the contrary, in terms of reviewing, all the coefficients are positive except for the case mentioned regarding the percentage of complete copies. In this case, the correlations are significant except for the first SQL exam. We must bear in mind that the correlations obtained between the studied variables do not imply causation between them.

Table 5. Mean (Standard Deviation) of Student Results Classified by the Following Categories: More Copies (C+) or Less Copies (C−), and More Review Sessions (R+) or Less Review Sessions (R−), and a 2 × 2 Classification Taking into Account Both Factors

Group	N	Tests	Labs	Exam 1	Exam 2	Exam 3
C−	48	5.96 (1.78)	8.50 (1.61)	5.77 (2.69)	5.67 (2.76)	6.01 (2.97)
C+	51	4.57 (1.77)	8.44 (1.54)	4.86 (2.67)	4.08 (2.50)	3.84 (2.92)
Statistic		$t = 3.558^{**}$ $p = 0.001$	$Z = 0.319$ $p = 0.750$	$Z = 1.376$ $p = 0.169$	$t = 2.717^{**}$ $p = 0.008$	$t = 3.253^{**}$ $p = 0.002$
R+	50	5.61 (1.93)	8.11 (2,15)	5.55 (2.91)	5.55 (2.79)	5.88 (2.85)
R−	49	4.23 (1.90)	7.86 (1.96)	4.74 (2.42)	4 (2.37)	4.07 (3.14)
Statistic		$t = 3.560^{**}$ $p = 0.001$	Z = 0.900 $p = 0.368$	Z = 1.158 $p = 0.247$	$t = 2.814^{**}$ $p = 0.006$	$t = 2.783^{**}$ $p = 0.007$
C−R+	33	6.27 (1.68)	8.69 (1.54)	6 (2.89)	5.86 (2.92)	6.2 (2.85)
C−R−	15	5.22 (1.85)	8.05 (1.75)	5.18 (2.15)	5.25 (2.15)	5.54 (3.35)
C+R+	17	5.34 (1.50)	8.59 (1.29)	4.93 (2.70)	4.77 (2.85)	4.71 (2.82)
C+R−	34	4.20 (1.80)	8.37 (1.66)	4.82 (2.71)	3.73 (2.28)	3.37 (2.91)
Statistic		$F = 6.889^{***}$	$\chi^2 = 1.657$ $p = 0.647$	$\chi^2 = 2.359$ $p = 0.501$	$F = 3.082^{*}$ $p = 0.032$	$F = 4.302^{**}$ $p = 0.007$
Bonferroni		C−R+>C+R−	–	–	C−R+>C+R−	C−R+>C+R−

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, Statistics: Student's t-test, Mann–Whitney U test, ANOVA F test, Kruskal–Wallis χ^2 test.

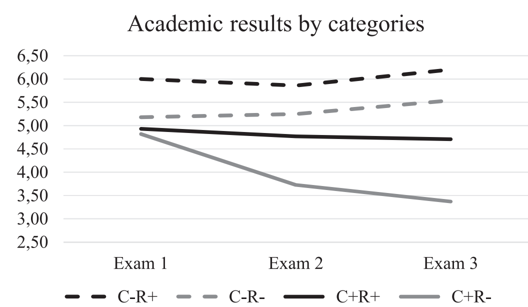

Fig. 4. Comparison of exam grades in 2 × 2 classification.

In order to further examine the relationship between review exercises and academic performance, students were classified in two groups: students that completed more reviews (R+), and students that completed less reviews (R−). To this end, the median of the total number of reviews completed was employed, which is 13 sessions. Students were also classified according to the number of complete copies: students that copied more (C+), and students that copied less (C−). Again, the median was used, which had a value of 13% copied exercises. These two factors allowed us to make a 2 × 2 classification of the students [24].

Table 5 contains the results obtained for each group of students on the different assessments using the aforementioned classifications, and Figure 4 depicts the exam grades in the 2 × 2 classification. Students in group C− obtained significantly better results on tests; hence, they could be considered better students than those in the C+ group. However, lab session grades were quite similar, though the fact that copying was not penalized must be taken into account. Finally, students in group C− obtained better results on all the SQL exams. In addition, all the differences are significant except for the first SQL exam which contained the easiest exercises. Regarding the R+

group, something similar to the C− group occurred: the students in this group obtained significantly better results on the tests and on the second and third SQL exams.

Regarding the 2×2 classification of whether students belong to C+ or C−, and R+ or R−, one can observe that students are not randomly distributed among these groups ($\chi^2 = 11.622$; $p = 0.001$) and that they tended to polarize into opposing groups containing the students with the worst (C+R− with 34% of the students) and the best (C−R+ with 33%) academic results. The group C−R− includes 17% of the students, and C+R+ 15%.

Students in the C−R+ group obtained the best results on all the assessments, with significant differences after the Bonferroni correction as compared to the C+R− group, which is the group of students who obtained the worst results on the tests and SQL exams 2 and 3. There are no significant differences in the case of the first SQL exam, which was notably easier, or in the lab sessions where copied code was not penalized (and the grades are quite similar).

The intermediate strategies (C−R and C+R+) obtained similar grades on tests, and there are not significant differences between them. However, despite the fact that the group C−R− obtained worse results in the lab sessions (where copying might have been influential), they achieved better results on all the SQL exams. This group progressively improved on the SQL exams, whereas the C+R+ group's performance progressively declined on those assessments. Regarding the third SQL exam (the most difficult), it can be observed that the difference between C−R+ (the group with the best results) and C−R− produces a low size effect (Cohen's $d = 0.21$), whereas the difference between C−R+ and C+R+ produces a medium size effect (Cohen's $d = 0.53$).

After dividing students into groups that copied more and less, and groups that reviewed more and less, the above analysis shows that both the students who copied less and the students who reviewed more obtained better academic results. In addition, although significant differences between the two intermediate strategies were not found, copying less seems to lead to better academic results than reviewing more.

4.4 Satisfaction with the Learning Tool

This final subsection describes the findings regarding the relationship between the new module and student satisfaction with the tool. Table 6 contains the results of the satisfaction survey, which was completed by students on an anonymous basis. Group 1 submitted 87 surveys and group 2, 89 surveys.

The survey has a high reliability with a Cronbach's alpha of 0.759 when considering all the items used for the second group (and 0.784 when only considering the items common to both groups). In addition, a positive correlation of Rho = 0.652 ($p < 0.001$) was obtained between the final global measurement item and the score obtained from the sum of rest of the items when considering all the items used for the second group (Rho = 0.735, $p < 0.001$ when considering only the items common to both groups), representing an acceptable criterion-related validity [24].

The level of satisfaction with the tool is quite high for both groups of students. Across all the items, the results are closer to agreement (3.0) than to disagreement. Group 2 was notably more positive regarding all the items as compared to group 1. It is worth mentioning that the last item, "In general, I am satisfied with the tool," shows that 52% of students in group 1 agree or strongly agree, and this percentage increases to 97% for group 2. Another item that should be mentioned is "I prefer to conduct the lab sessions using this tool instead of working with Blackboard and MySQL Workbench." In this case, 66% of the students in group 1 agree or strongly agree, compared to the 79% of the students in group 2. Finally, let us highlight that 93% of the students in group 2 agree or strongly agree with the statement "It facilitates the study of SQL queries," and 97% agree or strongly agree with the statement "We should have similar tools in other courses." These results

Table 6. Mean (Standard Deviation) of Student Satisfaction with the Tool before and after Introducing Review Exercises

Items about the SQL learning tool	Group 1	Group 2	Statistic	
			Z	p
It is simple to use.	3.30 (0.7)	3.49 (0.64)	−1.943	0.052
It has an intuitive interface.	3.28 (0.64)	3.36 (0.64)	−0.907	0.365
It provides useful suggestions/hints about correcting exercises.	2.55 (0.94)	2.81 (0.81)	−1.963	0.050
The summary about the answered exercises is useful.	3.53 (0.71)	3.72 (0.54)	−1.931	0.054
I prefer to conduct the lab sessions using this tool instead of working with Blackboard and MySQL Workbench.	2.81 (0.86)	3.25 (0.99)	−2.707**	0.007
It helps to complete the lab sessions.	3.26 (0.69)	3.51 (0.52)	−2.249*	0.025
Doing review exercises is useful for learning.	–	3.51 (0.62)		–
The number of exercises proposed during lab sessions is enough to be prepared for the course.	–	2.90 (0.74)		–
The number of review exercises is enough to be prepared for the course.	–	2.92 (0.76)		–
It facilitates the study of SQL queries.	–	3.33 (0.67)		–
We should have similar tools in other courses.	–	3.61 (0.56)		–
In general, I am satisfied with the tool.	2.81 (0.82)	3.51 (0.61)	−4.327***	

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, Statistics: Mann–Whitney U test.

show that student opinion is very positive regarding the tool for all the survey items, and their satisfaction increased when review exercises were incorporated.

5 DISCUSSION

5.1 Academic Performance and Code Copies

After incorporating a set of review exercises into an online tool for learning SQL, academic results improved considerably and a significant reduction in the percentage of complete copies was observed. This improvement was unrelated to the students' level, which was determined by their test grades; and indeed, the test grades for group 2 were slightly worse. It is worth recalling that the tests were unrelated to the use of the tool: the first part of the test was done prior to introducing the tool in the subject and the second part was based on theoretical database concepts unrelated to SQL. The percentage of students who copied in group 2 was 20%, which corresponds with the lowest level detected in a previous study about code copying in programming courses [25], where such percentages ranged from 20% to 50%.

The experience presented by Akçapınar [1] uses technology to detect copying between students in the reflections and discussions on concepts learned in a Computer Hardware course. He also employs technology to communicate the detected copies, but does not penalize students for it. With this approach, Akçapınar achieves a significant reduction in copying; however, he does not study its influence on academic results. Kaya and Özel [31] also employed a tool, integrated in Moodle, to monitor code copying in programming assignments. The tool was used during two consecutive years without penalizing copying. In the second year, the students were informed as to the existence of the tool; whereas during the first year, students were not aware of its existence. The results showed a significant reduction in copying and an improvement in academic performance. The authors explain that this system forces students to work harder on assignments, and that this increased effort positively influences their success rate. On the contrary, our strategy does not force students to complete the exercises without copying; and, indeed, copying was not monitored during lab sessions. Nevertheless, some students voluntarily chose not to copy and instead opened the tool for self-learning exercises. In this study, we have followed a strategy similar to the approach suggested by authors like Morris [40] or Carroll [10]. Rather than a catch-and-punish strategy, we aimed to encourage students to do their own work, thus having the secondary effect of discouraging code copying.

Several studies identify a relationship between cheating and goal structures used in the learning environment [2, 12, 30]. Two kinds of goal structures are distinguished. The first one is performance. In this goal structure, students mainly engage in academic work to demonstrate their abilities and to compete with other students [2]. In a performance-oriented classroom, students may apply avoidance strategies to achieve their goals. Research generally indicates that performance perceptions are related to negative academic outcomes [2]. The second kind of goal structure is mastery. Classrooms that convey this type of goal structure emphasize learning, improvement, effort, and understanding as important reasons for engaging in academic work [53]. In this kind of goal structure, research indicates that students are less likely to use maladaptive strategies like avoidance. Mastery perceptions are related to positive outcomes [2]. In our case, before the review exercises were incorporated into Aplicación BD, the learning tool could have represented (for students) a way to solve and submit compulsory exercises during lab sessions. These sessions were conducted under the pressure of a limited, though sufficient, amount of time and in a confined space. However, incorporating review exercises might have shifted the learning environment towards a more mastery-oriented structure. By using our tool for review sessions, students understood that the tool is a way to learn SQL, a nuance that group 1 would not have perceived. Therefore, in labs with the same time constraints, group 2 reduced code copying and increased the use of the tool as a means to achieve learning. This reduction in code copying, along with the learning effort devoted to using the review module, translated into an improvement in academic results.

5.2 Gender Analysis

The results of the gender analysis show that women copied less and obtained better results, but without significant differences. Once the review sessions were introduced, it is observed that women reviewed more but, again, without significant differences. However, a more pronounced decrease of copying among women with a medium-large effect size was found. Several studies highlight the fact that women work harder and copy less than men [2, 25, 36, 42, 48]. Our results about the performance of both genders agree with other studies about teaching SQL [17] or in other DB courses [16] where no significant differences were found for academic performance in terms of gender.

5.3 Influence of Copying and Reviewing on Academic Performance

The results obtained after dividing students into groups that copied more and less, and groups that reviewed more and less, show that both the students that copied less (and therefore, put more effort into the labs), as well as the students that reviewed more, obtained better academic results. These results agree with those obtained in other studies about code copying [3, 31] or learning effort in academic performance [4, 23, 48, 53]. However, we have not been able to find studies that involve both parameters: code copying (derived from a third party's effort) and review exercises (derived from extra effort), and their relationship with academic performance.

From the aforementioned factors (copying and reviewing), four groups of students can be established. The students that copied less and reviewed more obtained the best results and showed a higher level of performance. On the opposite end of the spectrum are the students who copied more and reviewed less and obtained results similar to the former group only in the lab sessions. We can find parallels between the studied groups and the classification of students according to their goal orientation [30, 54]. The students that copied less and reviewed more might fit into a mastery goal orientation. This orientation focuses on learning the material and mastering the task, students may use meaningful cognitive strategies to solve problems and show persistent effort. On the contrary, students that copied more and reviewed less seemed to follow a work-avoidant goal orientation. These students seek to complete the course or curriculum with minimal effort expenditure. It is well documented that the first strategy has a positive impact on learning outcomes, whereas the second is the most detrimental to learning and achievement outcomes [12, 54].

In addition, significant differences between the two intermediate groups were not found (the students that copied less but also reviewed less, and the students that copied more but also reviewed more). However, copying less seems to lead to better academic results than reviewing more. That is to say, the effort invested in doing review exercises does not seem to compensate for the learning effort avoided by copying exercises during lab sessions. If we continue with the parallelisms observed in the classification according to goal orientation, it seems that the students that copied less but also reviewed less might be following a performance goal orientation. This orientation involves a person comparing his or her own competence to others' in order to demonstrate his or her ability. These students view themselves as having a good deal of ability, and they are willing to put forth the effort to ensure that others see that they are intelligent. Due to this characteristic, students tend to submit their own version of the exercises without copying others. However, these students may not be task-involved [54]; therefore, they are not willing to do review exercises that are not compared with others' work. Finally, the students that copied more while also reviewing more seemed to follow a performance-avoidant goal orientation. Students who operate according to this orientation view themselves as lacking ability and, therefore, wish to avoid a public display that would confirm their lack of ability. These students are ego-involved and in order to protect their self-worth, they adopt failure-avoiding strategies [54]. Hence, they try to copy exercises from other students; but the desire to overcome their feelings of insecurity leads them to do review exercises as well.

Adults who display performance orientation have negative learning outcomes compared to mastery orientation [12, 54]. Likewise, performance goal-oriented individuals are more likely to succeed than performance-avoidant individuals [55]. These conclusions concur with the results observed for the groups analyzed in our study (where parallels can be observed). However, it must also be noted that our study's manner of classifying students is different. In the case of goal orientation studies (see. for instance, [30] or [54]), the data employed to classify students is obtained from a self-report completed by the students; in our case, however, we employed data from students' actual work with the learning tool, which stores the data itself.

5.4 Satisfaction with the Learning Tool

The results regarding student satisfaction with the SQL learning tool show that student opinion is very positive regarding the tool for all the items. These results agree with the findings of other studies involving similar tools [9, 13, 38, 47, 51]. In addition, let us underscore the fact that student satisfaction improved when review exercises were incorporated, and significantly for the most important items on the survey. Furthermore, it should be noted that even satisfaction with those items unrelated to the introduction of review exercises also improved. Given the anonymous nature of the survey, it is not possible to determine which type of student (whether the student who copies or the student who reviews) showed increased satisfaction. The set of exercises was continually available through the tool, and the tool provided helpful hints throughout the exercise. This seemed to motivate students more than the traditional lists of exercises. As a result, the time devoted to study increased, and this fact can be appreciated in students' academic results and satisfaction. However, an increase in learning effort does not always translate to an increase in satisfaction. In fact, it can have the opposite effect [28] if, for example, the necessary extra effort was not voluntary. Let us also point out that continuous interaction with the environment positively influences people's preferences and the activities they voluntarily choose to do [30]. This effect is likely further enhanced by introducing voluntary review exercises.

5.5 Threats to Validity

There are several limitations to this research. First, a random allocation of subjects to different experimental groups was not implemented. In our case, the experimental design posed problems because in our application it is not possible to release review exercises to only a certain group of students, while hiding them from another group. Thus, a quasi-experimental design was used, comparing students from two different periods of time. With this method, it is not possible to avoid differences between the two groups of students from those two distinct periods of time. For instance, the overall academic level of students could improve over time. While trying to control such circumstances as much as possible, we studied the confounding variables related to the differences in student capabilities between the groups. In particular, we found that both groups obtained similar grades on the tests (which were taken prior to the inclusion of SQL or did not include SQL concepts). Furthermore, both groups of students had similar grade point averages before starting university and a similar distribution regarding prior studies completed. Although these similarities between the groups could be sufficient for our quasi-experimental study, other techniques to avoid the effect of confounding variables could be taken into account, for instance, normalizing the performance measures using those confounding variables.

In addition, the application context (i.e., language programming learning) and the sample size, while generally considered sufficient for analysis, do require further research in different contexts and with varied and larger samples to verify the results and conclusions before making broad generalizations. The results of this study are consistent with the research on effort and code copying behaviors in other contexts. Finally, the methods used for measuring code copying and extra effort are quite simple, though sufficient for our purposes. Nevertheless, more advanced methods could be implemented to the same end.

5.6 Implications

The significant implications of our study are related to the positive effects of a teaching-and-learning approach on academic performance and code copying. Although the teaching-and-learning strategy is already addressed in the literature [40, 10], there is a lack of quantitative research to support its validity, particularly in regards to engineering students [6]. Instead of using a tool to identify and penalize code copying, this study employed a tool to improve the learning context with the aim of guiding the learning environment towards a mastery-goal structure.

On the other hand, it is our belief that students' results improved thanks to the introduction of a dual orientation, compulsory-voluntary, in the use of the tool. Initially, the tool was only used to set several milestones (lab sessions reinforced by means of written assessments) that had to be fulfilled by students. Then, a new voluntary aspect was incorporated: additional optional exercises to reach those milestones. We have verified that the combination of mandatory and voluntary activities leads to better results than the compulsory part alone. We believe that when the tool is only used for compulsory tasks (solving and submitting exercises in lab sessions), students perceive it as performance-oriented, that is, as merely a means of assessment. This type of orientation may impel them toward avoidance strategies such as code copying [2]. Thus, the role of the tool as a learning aid is overshadowed. The combination of both orientations helps make students aware of how the tool can help them learn, and the effort saved as compared to pen-and-paper exercises or the use of a generalist tool (like a DBMS). This shift in perception encourages students to use the tool as a means to learning in lab sessions as well, not just to submit results, thus reducing code copying. Our results suggest that when a new e-learning tool is designed, this dual orientation should be taken into account.

5.7 Further Work

Although our tool has a specific purpose, SQL learning, it is our belief that the aforementioned dual orientation could be generalized to other tools, used in different subjects, and with other types of copying on reports, assignments, essays, projects, and so on, by using technology to foster a more mastery-oriented learning environment through a teaching-and-learning approach [10, 30, 40]. This possibility needs to be verified through further research.

It is our belief that the voluntary aspect of the tool is not sufficient on its own either. Nowadays, students have access to a plethora of opportunities to learn. In this study, these opportunities have essentially remained the same over both periods studied (except for the review exercises included in Aplicación DB in the second period). One example is instructors' office hours, which in our experience are underutilized by students. Another example is the Internet, which is full of opportunities and examples for students to use to practice. For instance, there are interesting online tools like SQLZOO [52], but we have not received any comments or questions indicating that these resources are employed by students. These are examples of voluntary options of which students fail to take advantage. Future research should investigate what happens if a tool is only employed on a voluntary basis. It is our belief that the compulsory part of the tool acts as an incentive to encourage learning. Without the established lab-session milestones, students would see the tool simply as extra material that supplements their textbooks, lists of exercises, or exercises available on the Internet. Verifying such a hypothesis remains for further research.

Finally, we could analyze the relationship between the 2×2 classification used in our study (based on more or less copying and more or less reviewing) and the classification found in the literature [12, 54] of mastery, performance-goal, performance-avoidant, and work-avoidant oriented.

6 CONCLUSIONS

This study analyzed the impact of a new review exercises module incorporated into an online tool with the aim of promoting self-learning and reducing code copying in a software engineering subject, in accordance with the teaching-and-learning approach. Two groups of students were compared. The first group was required to utilize the learning tool in lab sessions, whereas the second also had the option of using it for self-learning. Better academic results, a decrease in code copying (more pronounced among women), and a higher level of satisfaction were observed among the students in the second group.

In addition, within the second group, the students that copied more or less, and the students who put forth more or less effort were identified and the combination of both factors was analyzed. It was found that students who copied less and invested more effort obtained better results. It was also determined that students who, in spite of making the extra effort of completing voluntary exercises, still copied exercises from classmates, obtained worse results than those students who copied less exercises, even if they completed less review exercises.

This experience has shown the positive effects of using a learning tool with a dual purpose. Its first function is its compulsory use during lab sessions. This requirement establishes some milestones that act as incentives to encourage learning. The second purpose is voluntary and aims to enhance the advantages of employing a tool that helps in the learning process. This two-part orientation should be taken into consideration for new e-learning tools in the future.

REFERENCES

[1] G. Akçapınar. 2015. How automated feedback through text mining changes plagiaristic behavior in online assignments. *Computers & Education* 87 (2015). 123–130.

[2] E. M. Anderman and C. Midgley. 2004. Changes in self-reported academic cheating across the transition from middle school to high school. *Contemporary Educational Psychology* 29, 4 (2004), 499–517.

[3] I. J. M. Arnold. 2016. Cheating at online formative tests: Does it pay off? *The Internet and Higher Education* 29 (2016), 98–106.

[4] D. Bartelet, J. Ghysels, W. Groot, C. Haelermans, and H. M. van den Brink. 2016. The differential effect of basic mathematics skills homework via a web-based intelligent tutoring system across achievement subgroups and mathematics domains: A randomized field experiment. *Journal of Educational Psychology* 108, 1 (2016), 1–20.

[5] Bertram T. Gallant. 2008. Academic integrity in the twenty-first century: A teaching and learning imperative. *ASHE Higher Education Report* 33, 5 (2008), 1–143.

[6] Bertram T. Gallant, L. Van Den Einde, S. Ouellette, and S. Lee. 2014. A systemic analysis of cheating in an undergraduate engineering mechanics course. *Science and Engineering Ethics* 20 (2014), 277–298.

[7] S. Bhagat, L. Bhagat, J. Kavalan, and M. Sasikumar. 2002. Acharya - An intelligent tutoring environment for learning SQL. In *Proceedings of Vidyakash 2002 International Conference on Online Learning*.

[8] A. Bhangdiya, B. Chandra, B. Kar, B. Radhakrishnan, K. V. M. Reddy, and S. Shah et al. 2015. The XDa-TA system for automated grading of SQL query assignments. In *Proceedings of the 31st IEEE International Conference on Data Engineering (ICDE)*. 1468–1471.

[9] X. Burgués, C. Martín, M. C. Quer, A. Abelló, M. J. Casany, and A. Urpí et al. 2010. Adapting LEARN-SQL to database computer supported cooperative learning. In *Proceedings of the 3rd Workshop on Methods and Cases in Computing Education*. 22–29.

[10] J. Carroll. 2009. Plagiarism as a threat to learning: an educational response. In *Assessment, Learning and Judgement in Higher Education*, J. Gordon (Ed.). Springer, 115–131.

[11] M. Cembalo, A. De Santis, and U. Ferraro. 2011. SAVI: A new system for advanced SQL visualization. In *Proceedings of the 2011 Conference on Information Technology Education (SIGITE'11)*. ACM, New York, 165–170.

[12] W. W. Chen and Y. L. Wong. 2015. The Relationship between goal orientation and academic achievement in Hong Kong: The role of context. *The Asia-Pacific Education Researcher* 24, 1 (2015), 169–176.

[13] M. Cvetanovic, Z. Radivojevic, V. Blagojevic, and M. Bojovic. 2011. ADVICE—Educational system for teaching database courses. *IEEE Transactions on Education* 54, 3 (2011), 398–409.

[14] M. de Raadt, S. Dekeyser, and T. Y. Lee. 2007. A system employing peer review and enhanced computer assisted assessment of querying skills. *Informatics in Education* 6, 1 (2007), 163–178.

[15] R. Dollinger and N. A. Melville. 2011. Semantic evaluation of SQL queries. In *Proceedings of the 2011 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. 57–64.

[16] C. Domínguez and A. Jaime. 2010. Database design learning: A project-based approach organized through a course management system. *Computers & Education* 55, 3 (2010), 1312–1320.

[17] A. Efendioğlu and T. Y. Yelken. 2010. Programmed instruction versus meaningful learning theory in teaching basic structured query language (SQL) in computer lesson. *Computers & Education* 55, 3 (2010), 1287–1299.

[18] P. D. Ellis. 2010. *The Essential Guide to Effect Sizes Statistical Power, Meta-Analysis, and the Interpretation of Research Results Textbook*. Cambridge University Press.

[19] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner. 2007. G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods* 39 (2007), 175–191.

[20] P. Garner and J. Mariani. 2015. Learning SQL in steps. *Journal of Systemics, Cybernetics and Informatics* 13, 4 (2015), 19–24.

[21] J. Ghysels and C. Haelermans. 2018. New evidence on the effect of computerized individualized practice and instruction on language skills. *Journal of Computer Assisted Learning* (2018), 1–10.

[22] J. M. Gullifer and G. A. Tyson. 2014. Who has read the policy on plagiarism? *Unpacking Students' Understanding of Plagiarism, Studies in Higher Education* 39, 7 (2014), 1202–1218.

[23] C. Haelermans and J. Ghysels. 2017. The effect of individualized digital practice at home on math skills—Evidence from a two-stage experiment on whether and why it works. *Computers & Education* 113 (2017), 119–134.

[24] J. F. Hair, W. C. Black, B. J. Babin, and R. E. Anderson. 2009. *Multivariate Data Analysis*. Pearson. 7th ed.

[25] M. Hart and T. Friesner. 2004. Plagiarism and poor academic practice—A threat to the extension of e-learning in higher education? *Electronic Journal of eLearning* 2, 1 (2004), 89–96.

[26] A. S. L. Hoque, G. M. Bashir, and R. Uddin. 2014. Equivalence of problems in problem based e-learning of database. In *Proceedings of the 2014 IEEE 6th International Conference on Technology for Education*. 106–109.

[27] A. S. L. Hoque, M. M. Islam, I. Hossain, and F. Ahmed. 2012. Problem-based e-learning and evaluation system for database design and programming in SQL. *International Journal of e-Education, e-Business, e-Management and e-Learning* 2, 6 (2012), 537–542.

[28] A. Jaime, C. Domínguez, A. Sánchez, and J. M. Blanco. 2013. Interuniversity telecollaboration to improve academic results and identify preferred communication tools. *Computers & Education* 64 (2013), 63–69.

[29] L. P. Pate. 2016. Technology implementation: Impact on students' perception and mindset. *International Journal of Information and Learning Technology* 33, 2 (2016), 91–98.

[30] Y. Kauffman and M. F. Young. 2015. Digital plagiarism: An experimental study of the effect of instructional goals and copy-and-paste affordance. *Computers & Education* 83 (2015), 44–56.

[31] M. Kaya and S. A. Özel. 2014. Integrating an online compiler and a plagiarism detection tool into the Moodle distance education system for easy assessment of programming assignments. *Computer Applications in Engineering Education* 23, 3 (2014), 363–373.

[32] R. Kearns, S. Shead, and A. Feteke. 1997. A teaching system for SQL. In *Proceedings of the 2nd Australasian Conference on Computer Science Education (ACSE'97)*. ACM, New York. 224–231.

[33] H. Laine. 2002. SQL-Trainer. In *Proceedings of the 1st Annual Finnish/Baltic Sea Conference on Computer Science Education*. University of Joensuu, Department of Computer Science, Report A-2002-1. 13–17.

[34] A. Latham, K. Crockett, D. McLean, and B. Edmonds. 2012. A conversational intelligent tutoring system to automatically predict learning styles. *Computers & Education* 59, 1 (2012), 95–109.

[35] X. López, J. Valenzuela, M. Nussbaum, and Ch. Tsai. 2015. Some recommendations for the reporting of quantitative studies. *Computers & Education* 91 (2015), 106–110.

[36] Ch. Mensah, E. Maxwell, and M. Enyonam. 2016. Examination cheating attitudes and intentions of students in a Ghanaian polytechnic. *Journal of Teaching in Travel & Tourism* 16, 1 (2016), 1–19.

[37] M. Misic, Z. Siustran, and J. Protic. 2016. A comparison of software tools for plagiarism detection in programming assignments. *International Journal of Engineering Education* 32, 2 (2016), 738–748.

[38] A. Mitrovic and S. Ohlsson. 1999. Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education* 10 (1999), 238–256.

[39] A. Mitrovic, S. Ohlsson, and D. K. Barrow. 2013. The effect of positive feedback in a constraint-based intelligent tutoring system. *Computers & Education* 60, 1 (2013), 264–272.

[40] E. J. Morris. 2016. Academic integrity: A teaching and learning approach. In *Handbook of Academic Integrity*, T. Bretag (Ed.). Springer, 1037–1054.

[41] J. T. Nadler, R. Weston, and E. C. Voyles. 2015. Stuck in the middle: The use and interpretation of mid-points in items on questionnaires. *Journal of General Psychology* 142, 2 (2015), 71–89.

[42] C. G. Northcutt, A. D. Ho, and I. L. Chuang. 2016. Detecting and preventing "multiple-account" cheating in massive open online courses. *Computers & Education* 100 (2016), 71–80.

[43] M. Novak. 2016. Review of source-code plagiarism detection in academia. In *Proceedings of the 39th International Convention on Information and Communication Technology*. Electronics and Microelectronics, MIPRO. 901–906.

[44] V. Perumal, B. Daniel, and R. Butson. 2017. Learning Analytics of Clinical Anatomy e-Cases. In *Big Data and Learning Analytics in Higher Education*. Springer, Cham. 253–263.

[45] J. C. Prior. 2003. Online assessment of SQL query formulation skills. In *Proceedings of the 5th Australasian Conference on Computing Education (ACE '03)*, Vol. 20. Australian Computer Society, Inc., Darlinghurst, Australia. 247–256.

[46] G. Russell and A. Cumming. 2005. On-line assessment and checking of SQL: Detecting and preventing plagiarism. In *Proceedings of the 3rd Higher Education Academy Workshop on Teaching, Learning and Assessment of Databases*. TLAD, Sunderland UK, LTSN-ICS. 46–50.

[47] S. W. Sadiq, M. E. Orlowska, W. Sadiq, and J. Lin. 2004. SQLator: An online SQL learning workbench. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. Leeds, U.K., 223–227.

[48] M. Sheard. 2009. Hardiness commitment, gender, and age differentiate university academic performance. *British Journal of Educational Psychology* 79, 1 (2009), 189–204.

[49] J. Sheard, Simon, M. Butler, K. Falkner, M. Morgan, and A. Weerasinghe. 2017. Strategies for maintaining academic integrity in first-year computing courses. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*. ITiCSE. 244–249.

[50] M. Soflano, T. M. Connolly, and T. Hainey. 2015. An application of adaptive games-based learning based on learning style to teach SQL. *Computers & Education* 86 (2015), 192–211.

[51] J. Soler, F. Prados, I. Boada, and J. Poch. 2006. A Web-based tool for teaching and learning SQL. In *Proceedings of the International Conference on Information Technology Based Higher Education and Training*. ITHET.

[52] SQLZOO. 2017. Online tool http://sqlzoo.net, (2017).

[53] G. S. Stump, J. Husman, and M. Corby. 2014. Engineering students' intelligence beliefs and learning. *Journal of Engineering Education* 103, 3 (2014), 369–387.

[54] Ch. Was. 2006. Academic achievement goal orientation: Taking another look. *Electronic Journal of Research in Educational Psychology, No. 10* 4, 3 (2006), 529–550.

[55] Ch. Was and T. L. R. Beziat. 2015. Exploring the relationships between goal orientations, knowledge monitoring and academic achievement. *Journal of Education and Human Development* 4, 3 (2015), 67–77.

[56] F. P. Yang, H. Ch. Jiau, and K. F. Ssu. 2014. Beyond plagiarism: An active learning method to analyze causes behind code-similarity. *Computers & Education* 70 (2014), 161–172.

[57] K. Zhao and C. Lei. 2017. Technology-enhanced content and language integrated learning in chinese tertiary english classes: Potentials and challenges. In *Innovation in Language Learning and Teaching*, New Language Learning and Teaching Environments. Palgrave Macmillan, London. 89–113.

Received October 2017; revised July 2018; accepted July 2018