Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 162 (2019) 827–834

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Conference on Information Technology and Quantitative Management (ITQM 2019)

Web platform for learning distributed databases' queries processing

Julio Herce-Zelaya[a], Carlos Porcel[b,1], Juan Bernabé-Moreno[a], Álvaro Tejeda-Lorente[a], Enrique Herrera-Viedma[a]

[a] *University of Granada, Department of Computer Science and Artificial Intelligence, Granada, Spain*
[b] *University of Jaén, Department of Computer Science, Jaén, Spain*

Abstract

A distributed database is a collection of data stored in different locations of a distributed system. The processing of queries in distributed databases is quite complex but of great importance for information management. Students who have to learn that process have serious difficulties for understanding them. On this work we present a web platform for helping the students learning the processing and optimization of queries in distributed databases. The novelty of this platform is that as far as we know, there is no similar graphical tool. It allows to visualize step by step the different phases of distributed query processing, showing how are they forming, making it easier for the students to understand these concepts. Moreover, having this web platform available, always and everywhere, indirectly have an impact on other competences like encouraging students' autonomous work and self-learning, adapting the teaching to its one-time necessities and reinforcing the advantages to apply information techniques in the teaching field. The results of the developed tests to validate the platform's functionalities and student's satisfaction were very positive.

Keywords: Learning platform, Distributed databases, Distributed queries processing

1. Introduction

The ever growing usage of information techniques facilitates the development of new pedagogical models. These models, that complement the on-site education, are known with the term "virtual teaching". The new technologies enrich the education with the possibility, not only to spread the information in a efficient and practical manner, but also to provide the participants (teachers, students, etc.) with the tools for a personal and group communication that reinforces the tutorial action and learning process [1].

One of the main pillars of these extended adoption of information technologies is the use of databases, discipline that have made so much progress and whose usage for complex problem has significantly grown in the last years. This evolution have encouraged the adoption of distributed architectures, in which the data is physically

*Corresponding author. Tel.: +34-953213017.
E-mail address: cporcel@ujaen.es.

1877-0509
10.1016/j.procs.2019.12.056

stored on different locations connected with each other through a network. In such a environment, a set of locations are arranged being able to operate interconnected with each other, but also autonomously. On this kind of environments, the distributed databases are a set of data stored on different locations on a distributed system [2]. Due to the rise of distributed environments, the adoption of this kind of databases is increasingly growing [3].

Because of that, the acquire from this knowledge is essential for future computer engineer graduates, since they definitely will have to face those problems in their future professional careers [4]. In the Degree in Computer Engineering of the University of Jaén (Spain) [1], we offer the subject called "Distributed Databases". The main contents studied in this subject are the following: Distributed architectures, logical and physical design of a distributed database, distributed query processing and administration and management of distributed databases [5]. After having taught for several years this subject and, asked the students, it has been identified that the topics that more effort required from the students were the processing and optimization of distributed queries. The main steps to process a distributed query are as follows (see section 3 for more details):

1. Transformation from a global query (expressed in Structures Query Languaje, i.e. SQL) into a fragmented query on a distributed database.
2. Fragmented query optimization, using the corresponding criteria that could be applied for every case [6].
3. Once the distributed query is obtained, express it again in SQL language.

However, as far as we know, there is no previous proposal for a graphic help tool for the understanding of these concepts. Therefore, in this work we propose an innovative pedagogical method that allows students to practice and improve their understanding of these steps, from anywhere and at any time.

Specifically, it is presented the design, development and implementation of an web application that helps students on the learning on how to process and optimize distributed queries. In particular, it is been developed a platform on which the transformation process from global queries (expressed in SQL) to fragmented queries on a distributed database is shown in detail step by step. Moreover, its open design would make easier the use from other teachers with similar needs, with the possibility to adapt it to another examples. The use of this platform facilitates the promotion of autonomous work and student self-learning. In addition, it represents another example of innovative practices in terms of information and communication technologies applied to teaching.

The rest of the work is organized like following indicated. On section 2 some basic aspects of distributed databases will be analyzed. On section 3 the platform is presented. On section 4 we show the tests performed and the obtained results. Finally, the obtained conclusions are exposed.

2. Distributed databases

In last years there has been an inclination to distributed process of information. The distributed databases adapt better to the decentralized structures, that the mayority of the companies either already integrate or are starting to do it, spreading their usage for the easiness and economical improvements on this adaptation process [2].

A distributed database system is defined as a collection of multiple databases connected in a logical manner, physically distributed on different nodes of the net that connects them [2, 5]. Each of these nodes has the ability of autonomous process. But in order to be considered a real distributed database, it is required that every node also participates on some global application, that is, that it needs to access data stored on other locations. Each of these logical parts of the database that are stored on different locations of this distributed system is what is known as *fragment*.

The design process of a distributed database is split on following steps [7]:

- Conceptual schema design.
- Global logical schema design.
- Local physical schema design.
- Fragmentation design, that is, establishing the logical criteria that motivates the fragmentation.
- Fragment allocation design, that is, deciding the physical location of the data and possible replicas.

[1] `https://www.ujaen.es/en`

A fundamental concept is the *transparency*, that indicates on which degree the users and applications ignore the details of the distribution, fragmentation and replication. All these details are stored in a global data dictionary, that includes the necessary tools to control the database and provide a global vision of itself. In order to access the data is necessary to query this dictionary, access each of the involved fragment and join the queried data.

A *Distributed Database Management System (DDBMS)* [8] manages a distributed database and is responsible for providing to the user the transparency for the distribution, fragmentation and replication. One of its main tasks is to enable the *query processing*. The queries will be expressed in SQL language refering global relations, and the DDBMS handles the transformation of these queries into other queries that match solely the involved fragments [9].

3. Description of the web platform

The main goal of distributed query processing consists of transforming a query on a distributed database into an efficient execution strategy over the local databases [5, 8]. The distributed query processing is something essential to comprehend, but it could turn out very complex [10]. Because of that, being able to visualise graphically and, step by step, the different stages of the distributed query processing and how they are transforming these queries, facilitates the understanding of these concepts for the students. This platform is being design in an open way with the idea that could be utilized as example for other teachers with similar circumstances. As follows it will be analysed the different parts of the system.

3.1. System Architecture

A client/server architecture was chosen and also a web interface was provided for the users [11]. The application is hosted on a server. Thus, it is allowed concurrent access from different clients at the same time, without the need for the users to install locally any additional software. Moreover, given its modular design, this architecture is easily scalable, both on new clients and add-on servers. The software elements that were used for the realisation of this application are the following:

- Web browser. It is the visual interface used for interacting with the system. It could be used any web browser able to execute Java code.
- Web server. In this case we opted for the Apache Tomcat server [2].
- Database managment system. It was used Oracle 11g [12], while the connection between Java and Oracle was made through the JDBC controller.

3.2. Database design

In the current implementation, we have used the database from the example shown in class (in subject "Distributed databases" of the degree of computer engineering of the University of Jaén), taking part of the whole project that is developed through the course [5]. The database itself that is going to be used, must be part of the design of the general database, as well as its fragmentation and localization. Therefore, we need to store information about the tables that our distributed database is composed of, the attributes of each of these tables, information about the design of the fragmentation that was performed, as well as the relation between the tables. With that, we obtain an open design that could be used in any distributed database or fragmentation, just by inserting the corresponding information.

3.3. Steps for the processing of distributed queries

It is being implemented the rules, equivalence transformations, operators and methods that are taught on the Distributed databases course, from the Computer Engineer Degree of the University of Jaén. In particular, the steps that the system performs to process the entered query, are the following [5, 9]:

1. Transformation from a global query into a fragmented query on a distributed database:

[2] `http://tomcat.apache.org`

Fig. 1. Transformation of the query to relational algebra.

(a) Transformation from a global query (expressed through the SQL language) into an equivalent expression of relational algebra. In this respect it was necessary an implementation, through a lexical and syntactic analyser, to check the syntax of the entered query. For that, tools like Jlex and Cup from Java [3] were used.
(b) Obtain the corresponding algebraic tree.
(c) Transformation into its canonical expression, replacing every relation that appears as an operand with the algebraic expression that rebuilds that relation from its fragments.

2. Fragmented query optimization using the corresponding criteria that could be applied for every case [6]:
   (a) Realise the projections of the fragments and using solely the corresponding attributes.
   (b) Realise the selections of the corresponding fragments.
   (c) Search contradictions on the selections and conditions that define each of the fragments, in order to ignore those that are not necessary.
   (d) Obtain contradictions from the conditions of the operands from a distributed natural product.
   (e) Distribution of natural products.
   (f) Distribution of associations and evaluations of set functions.
3. Once the distributed query is obtained, express it again in SQL language in order to be able to access the database and obtain, in this way, the required results.

3.4. System operation

We opted for a simple and intuitive interface. We would like to point out that currently the platform is in Spanish, since it is the course's official language. The interface consists of just a central text input on which the desired query is inserted. Next, if the query is correct, according to SQL syntax, it will be shown step by step how the query is processed. On each step it can be seen how the tree, representing the query, is evolving together with the applied transformation criteria. At the end the resulting tree is shown, as well as the query about the fragments on SQL.

Now we show an *operating example*. Suppose we want to retrieve the code of the suppliers that have made some supply, considering that the tables of suppliers and supplies are fragmented in different locations [5]. We access the platform, and in the text field we insert the following query: *select s.pnum from suministro s,proveedor p where s.pnum = p.pnum*. The system verifies that the query is correct, reports success and displays the query expressed through relational algebra (see Fig. 1).

Since everything has been correct, it also shows us the "Start" button (*"Empezar"* in Spanish), which we have to press to see how the query is being processed. Pressing said button, in a first step, shows the corresponding algebraic tree (see Fig. 2).

[3] `https://openfecks.wordpress.com/jlex-y-cup/`

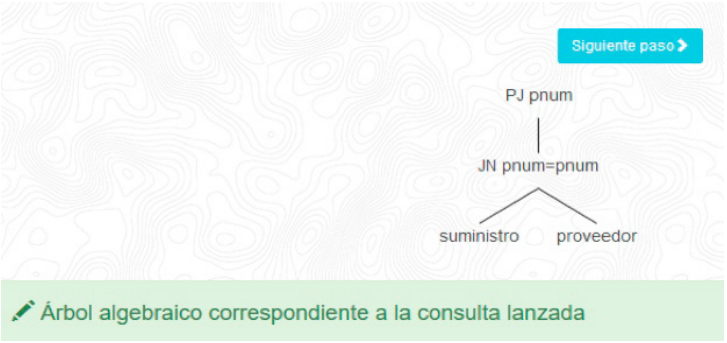

Fig. 2. Show the algebraic tree.

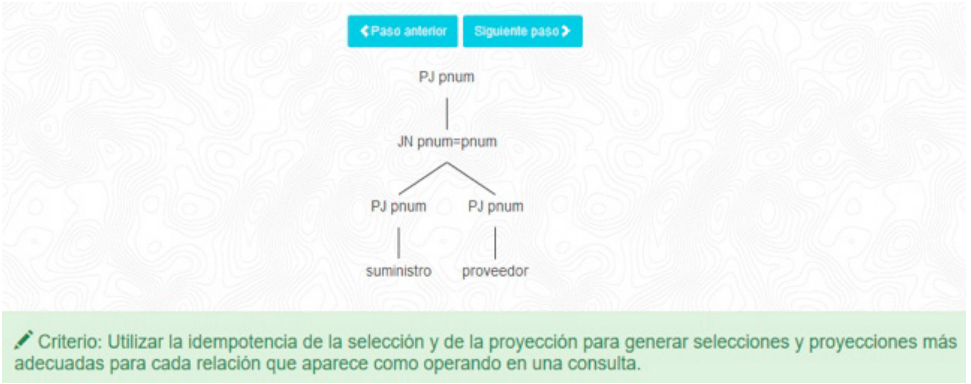

Fig. 3. Shows step by step how the tree is transformed.

In the following steps, this tree will be transformed to optimize the query, indicating the criteria that is being applied (see Fig. 3.

Criteria are still applied until the query is already optimized, at which point the final tree is shown (see Fig. 4). The system allows the user to navigate through the taken steps, going back to the previous step or moving to the next.

The last step is to show how the final query would look, in SQL language, but accessing only the necessary fragments (see Fig. 5).

4. System evaluation

4.1. Test cases

Firstly, were performed a set ot tests orientated to validate the right browse of the web site, by checking up the different links. Next, we proceed to realise functionality tests, that in our case consisted of realising different queries, both correct and incorrect, and to observer the result from the system in each case, comparing the actual result with the expected one. After realising these tests in the final version, the results were totally correct.

4.2. Web platform validation

For a complete validation we considered convenient to realise a satisfaction survey for the students from last academic course (2017-2018). Following the given indications by Nielsen in 2005 [4], the survey was split in two

[4] http://www.useit.com/papers/heuristic/heuristic_list.html

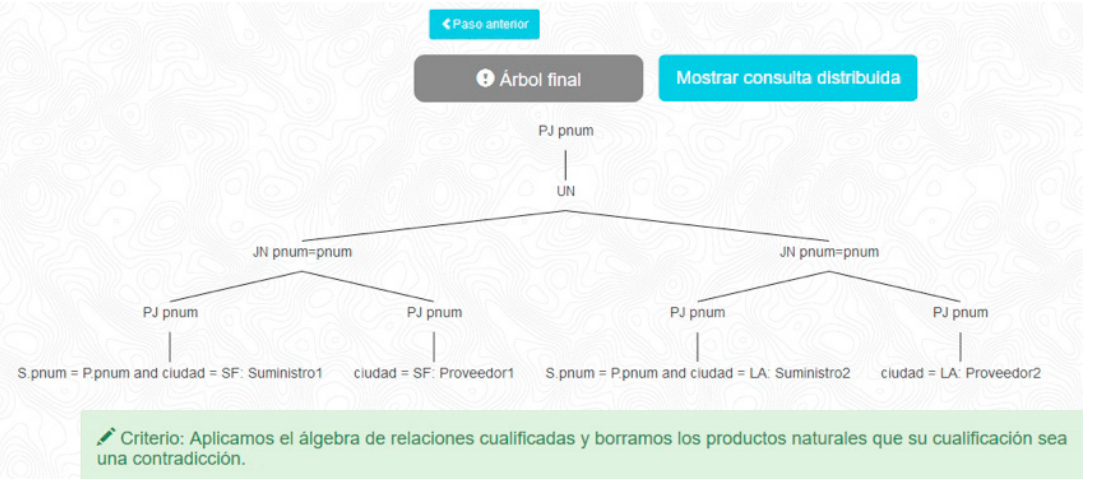

Fig. 4. Final tree of the inserted query.

```
(select pnum from Suministro2, Proveedor2 where Suministro2.pnum = Proveedor2.pnum) union (select pnum from Suministro1, Proveedor1 where Suministro1.pnum = Proveedor1.pnum)
```

Fig. 5. Final query distributed in SQL language.

sections, with the idea of measure on one hand, the users' general satisfaction aspects (4 questions) and on the other hand, analyse the opinions about the usability of the interface, that is, the easiness with which the users interact with the developed system (10 questions). Next is shown the survey, listing the questions from every section:

- User's satisfaction evaluation:
  - Q1. Do you like the information received from the system?
  - Q2. With the help of the system, do you consider that you have learnt the necessary, as well as acquired new ideas?
  - Q3. Do you consider useful the information and ideas that the system provides?
  - Q4. Do you consider that the ideas and the information provided improve the efficiency and results of the learning process?
- System's usability evaluation:
  - Q1. System's status visibility: the system keeps the user informed of what is is doing at every moment.
  - Q2. Relation between the system and the real world: the system communicates with the user using a familiar language.
  - Q3. Control and freedom of the user: the systems includes new ways that allow the user retake the control after accessing undesired functions or options.
  - Q4. Consistency and standards: the system adopts standards that allow the users not to have to check up themselves whether the different words, situations or actions have the same meaning.
  - Q5. Error prevention: the system includes confirmation options in order to prevent errors.
  - Q6. Site visibility instead of remember: the system keeps visible the different options in order to minimize the necessity from user's side to remember them.

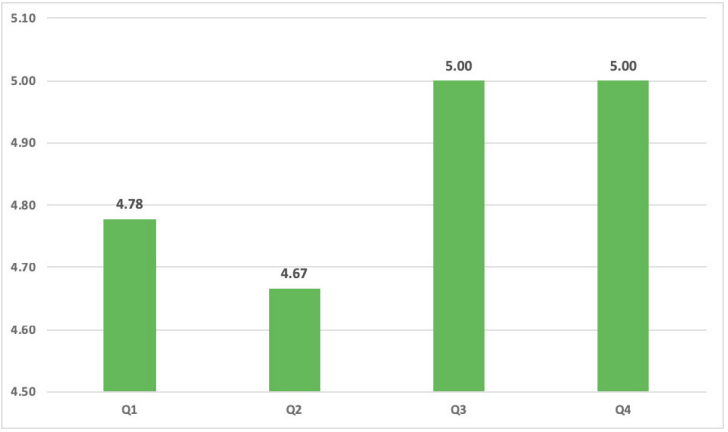

Fig. 6. Results of the user satisfaction survey.

- Q7. Flexibility and efficient usage: the systems allows to accelerate the interaction of expert users for actions frequently realised.
- Q8. Minimalist and cosmetic design: the messages shown contain relevant information and widely used, improving their visibility.
- Q9. Help users identify, diagnose and recover from errors: the error messages shown are expressed in a simple language, indicating the problem and suggesting the solution.
- Q10. Help and documentation: the system provides documentation easy to use and accessible.

To answer every question, a rating between 0 and 5 is assigned, using the 0 to indicate "does not know/ does not answer"; a 1 represents the lowest value and 5 the highest one.

Altogether a total of 35 students participated in this survey. The number of participants was not very high due to the low rate of registered students on the academic course 2017-2018. Therefore, instead of showing the data on a quantitative form, we have opted to show it with global pondered values and percentages, having in this way a more accurate idea in case of extrapolating the study to a larger set of students. That being said, we have realised two graphs that summarize the result of each of the two sections the survey is divided.

On Fig. 6 the obtained results with respect to the users' satisfaction are shown, indicating the average value of all ratings for every question. Observing the results, we can conclude that the satisfaction in each of the questions is very high (maximum in questions 3 and 4). But moreover, this fact is also reinforced if we consider the global average from all valuations for all questions, which is 4,86 (out of 5).

For the usability, the results are shown on the Fig. 7, where the percentage assigned to every of the questions is represented through a circular graph, considering the valuations given by all the students in regard to the total of available ratings. In this case, we can observe that almost the half of the ratings (48%) were a 5 (maximum valuation), revealing that the usability is quite good. And if we also considered that the aggregated data of the two maximum ratings, that is, 4 and 5, we would obtain a total percentage of 69%, confirming the good acceptance regarding system's usability. Especially taking into account that the percentage of bad ratings (1 and 2) is from 0%, that is, nobody provided any bad question in any of the aspects related to system's usability.

5. Conclusions

In this work, we have presented a web platform to help students facing distributed database study. It eases the learning and comprehension from distributed query process and optimization. The fact that students can visualize step by step the different stages on the process of distributed queries, and how they are transforming such a queries, makes the understanding of such as concepts easier without any doubt.

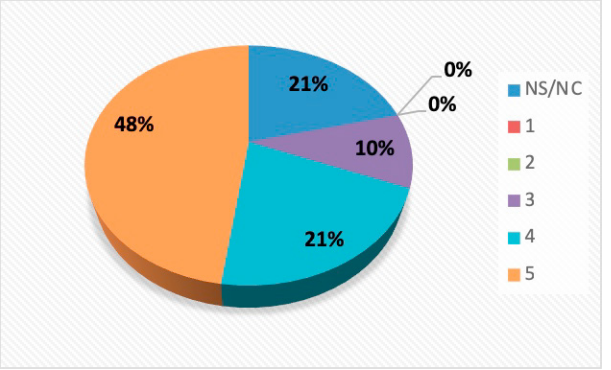

Fig. 7. Results of the survey about system's usability.

Once the application is developed and implemented, a relevant set of tests were performed in order to verify the functionality of itself. Students from the academic 2017-2018 participated on a survey to validate the application, through a survey to collect their opinions about their satisfaction and usability of the system. The obtained results of such tests were very satisfactory in both cases.

Despite the good results, we are aware that the system still has room for improvement, that will be address in next versions. In particular, a first action that we have planned, consists of improve the implemented lexical and syntactic analyser, because in their current state, their use is very restrictive and returns syntax errors that should not return. Another improvement to implement is to add the English language, in a way that the user could choose language and the platform could be used at international level.

Acknowledgements

This work has been developed thanks to the funding of the project PID46-201617 of the Universidad de Jaén.

References

[1] S. Bhattacharya, S. Nath, Intelligent e-learning systems: An educational paradigm shift, International Journal of Interactive Multimedia and Artificial Inteligence 4 (2) (2016) 83–88.
[2] M. Ozsu, Principles of Distributed Database Systems, Prentice-Hall, 2007. doi:https://doi.org/10.1007/978-1-4419-8834-8.
[3] Y. Chen, H. Xie, K. Lv, S. Wei, C. Hu, Deplest: A blockchain-based privacy-preserving distributed database toward user behaviors in social networks, Information Sciences 501 (2019) 100–117.
[4] I. Magdalena, The use of distributed databases in e-learning systems, Procedia - Social and Behavioral Sciences 15 (2011) 2673–2677.
[5] S. Ceri, G. Pelagatti, Distributed Database, Principles and Systems, McGraw-Hill, 1984.
URL https://books.google.es/books/about/Distributed_databases.html?id=WupQAAAAMAAJ&redir_esc=y
[6] V. Mishra, V. Singh, Generating optimal query plans for distributed query processing using teacher-learner based optimization, Procedia Computer Science 54 (2015) 281–290.
[7] U. Tosun, Distributed database design: A case study, Procedia Computer Science 37 (2014) 447–450.
[8] S. Rahimi, F. Haug, Distributed Database Management Systems: A Practical Approach, Prentice-Hall, 2010.
URL https://books.google.es/books/about/Distributed_Database_Management_Systems.html?id=VryuBgAAQBAJ&redir_esc=y
[9] A. Mathew, Data allocation optimization for query processing in graph databases using lucene, Computers & Electrical Engineering 70 (2018) 1019–1033.
[10] I. Magdalena, Distributed queries in the e-learning environment, Procedia - Social and Behavioral Sciences 28 (2011) 241–245.
[11] J. Hennessy, P. D.A., Computer Architecture. A Quantitative Approach, Morgan Kaufman, 1996.
URL https://books.google.es/books?id=HeVQAAAAMAAJ&q=Computer+Architecture.+A+Quantitative+Approach&dq=Computer+Architecture.+A+Quantitative+Approach&hl=es&sa=X&ved=0ahUKEwju3dbHwZbjAhWKxoUKHXaVDOYQ6AEINTAB
[12] K. Loney, Oracle Database 11g The Complete Reference, McGraw Hill Education, 2009.
URL https://books.google.es/books/about/Oracle_Database_11g_The_Complete_Referen.html?id=-4S6xQu-rmYC&redir_esc=y