**Eduard Cuadros 2017329**

**Edwin Garces 2017201**

**Thiago Almeida 2017201**

**Part A - Database analysis and design**

\_ As **a group**, provide a complete description of the application domain and any assumptions made;

**5%**

We as a team have decided to create a complete database specifically for a library in our school, which allows administrative staff and students to constantly monitor the books. Basically we have created a series of tables that allow us to have total control of each of the books. We have created the table called Books which contains: Id\_book, Title\_book and Type\_book. On the other hand we have the Students table: Id\_student, Name\_student, Gender and Course.

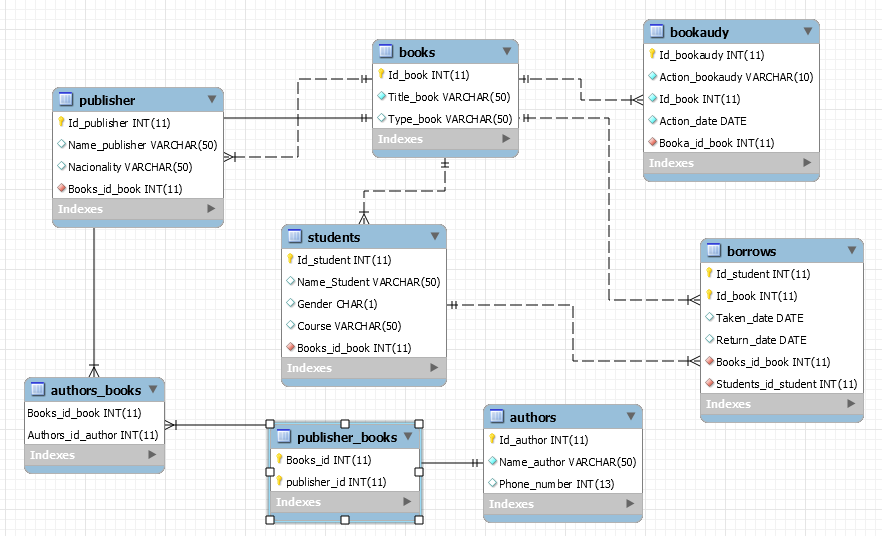
We have also created a table called Authors that contains: Id\_author, Name\_author, and Phone\_number. Without leaving aside Publisher (Id\_publisher, Name\_publisher, Nacionality). The table called Bookaudy (Id\_bookaudy, Action\_bookaudy, Id\_book, Action\_date). And the relations tables Authors\_books and Publisher\_books.

We are developing this database in great detail so that we can get the best out of it.

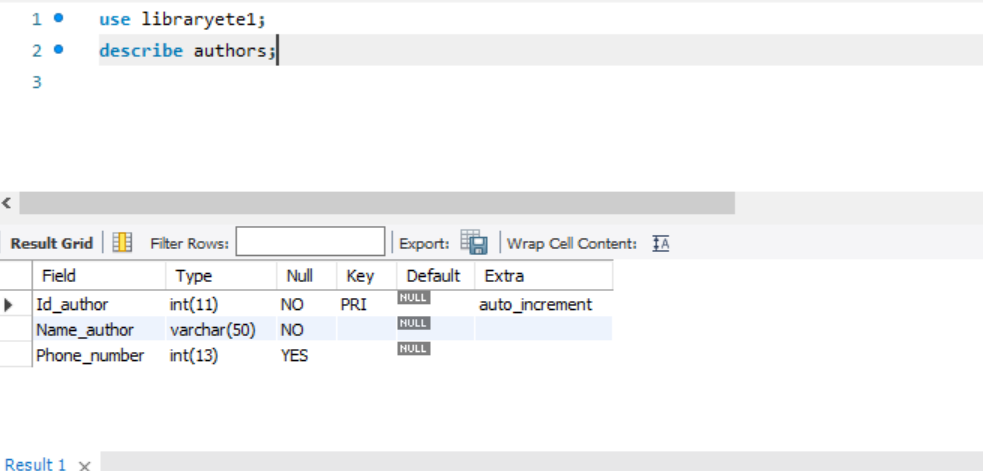
\_ As **a group**, provide complete **design documentation** in the form of complete table descriptions

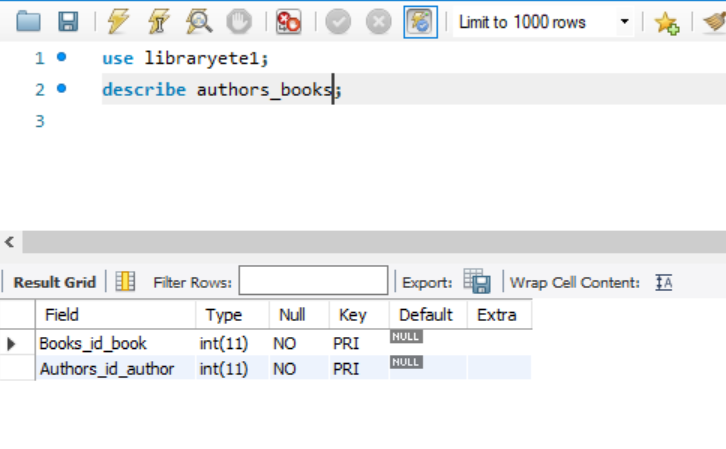
and an **enhanced entity-relationship diagram**;

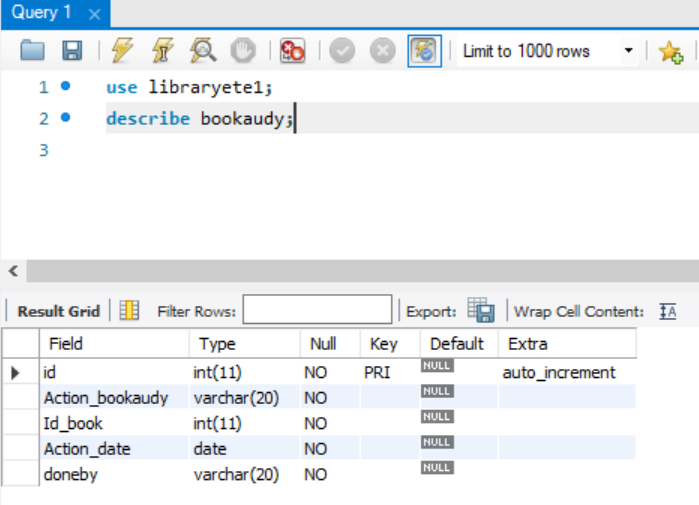
// we need to change this part with the arrows….

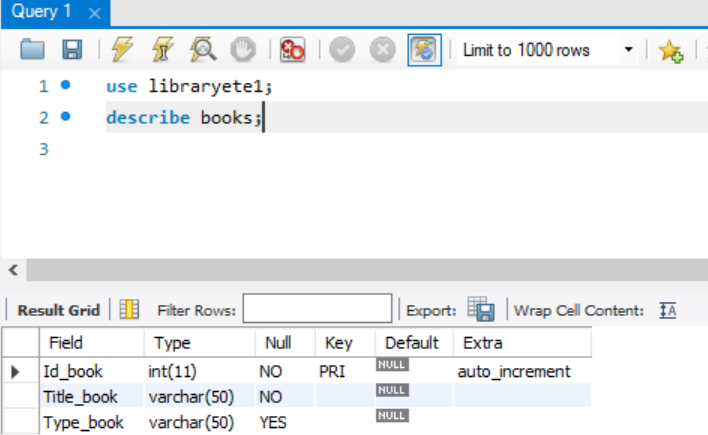


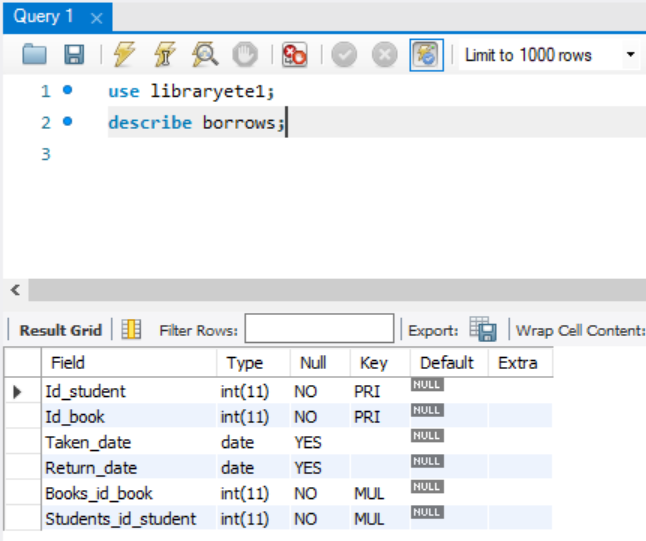
\_

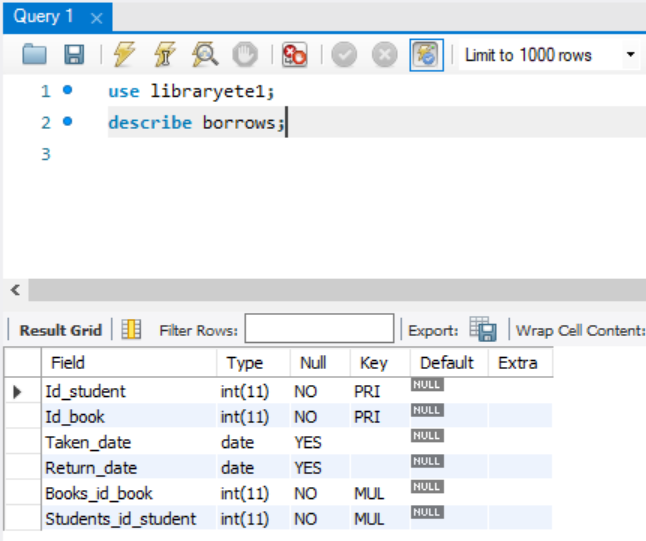


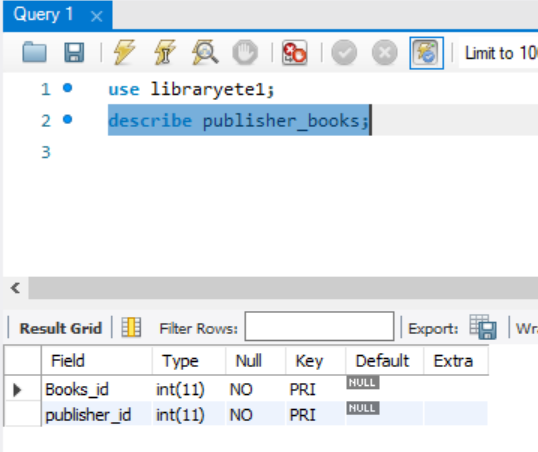


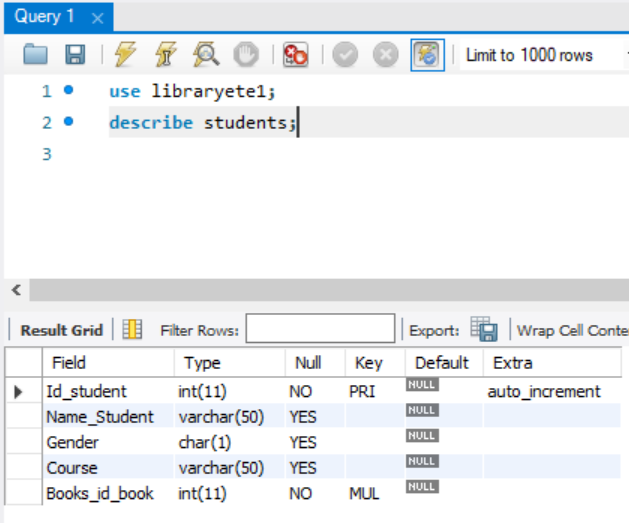








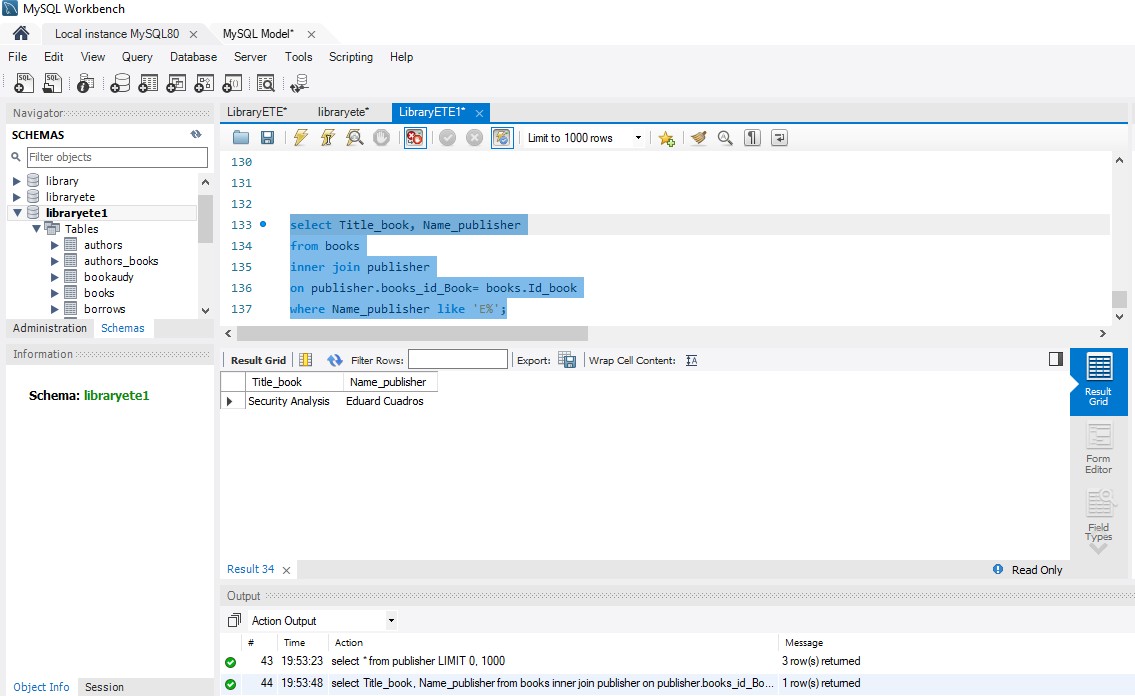




An example of each of the following:

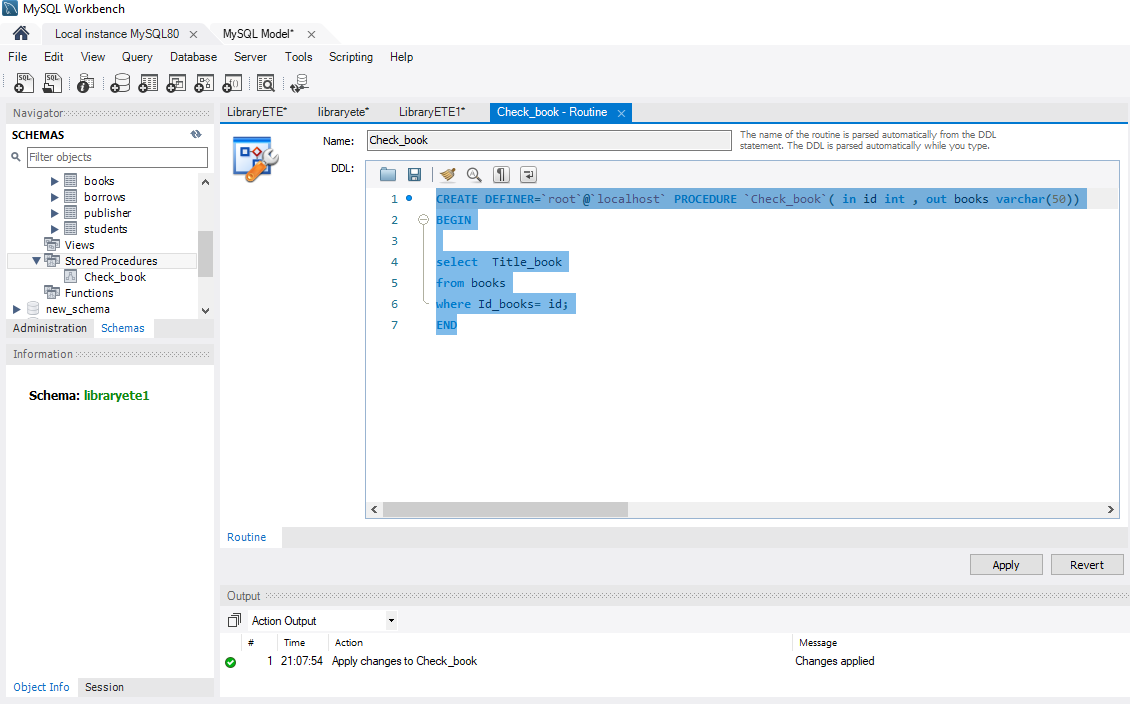
Eduard Cuadros 2017329

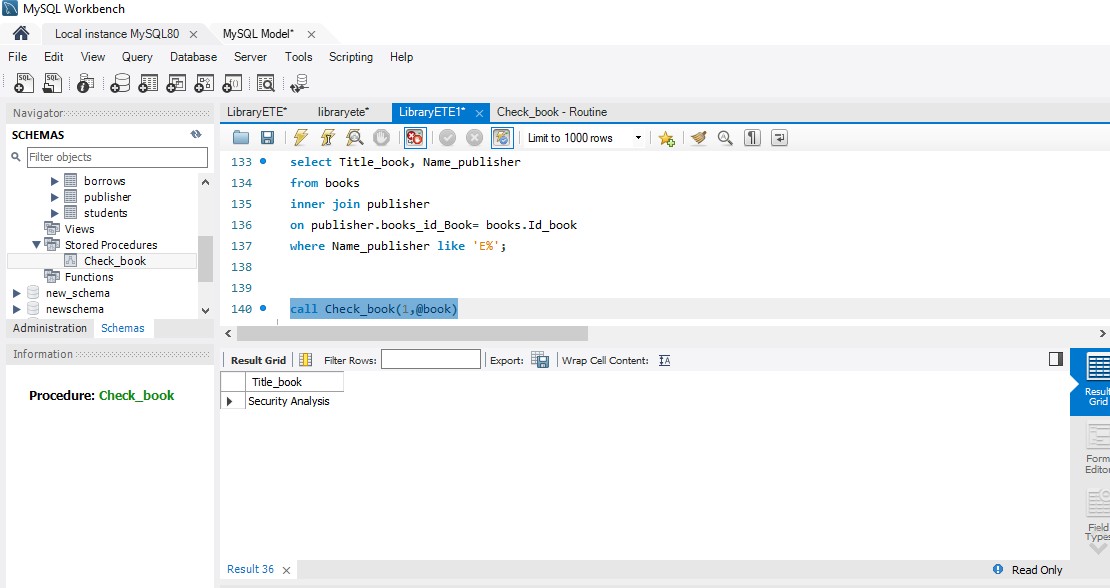
* One view **per student** incorporating at least two tables and a WHERE clause; **5%**



* One stored procedure **per student** that performs a multi-table SELECT query and includes at

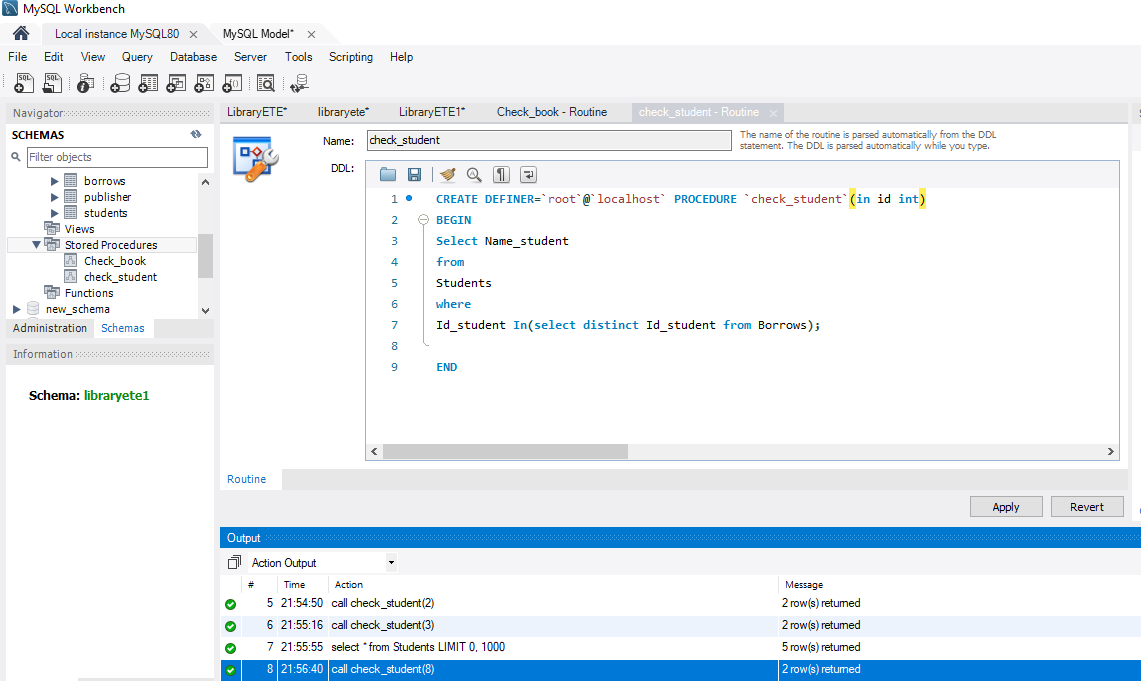
least one IN and one OUT parameter; **5%**





* One stored procedure **per student** that performs a correlated subquery and includes at least

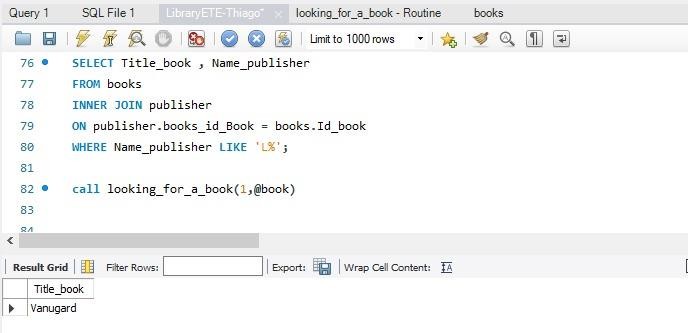
one IN parameter;



\_ An example of each of the following:

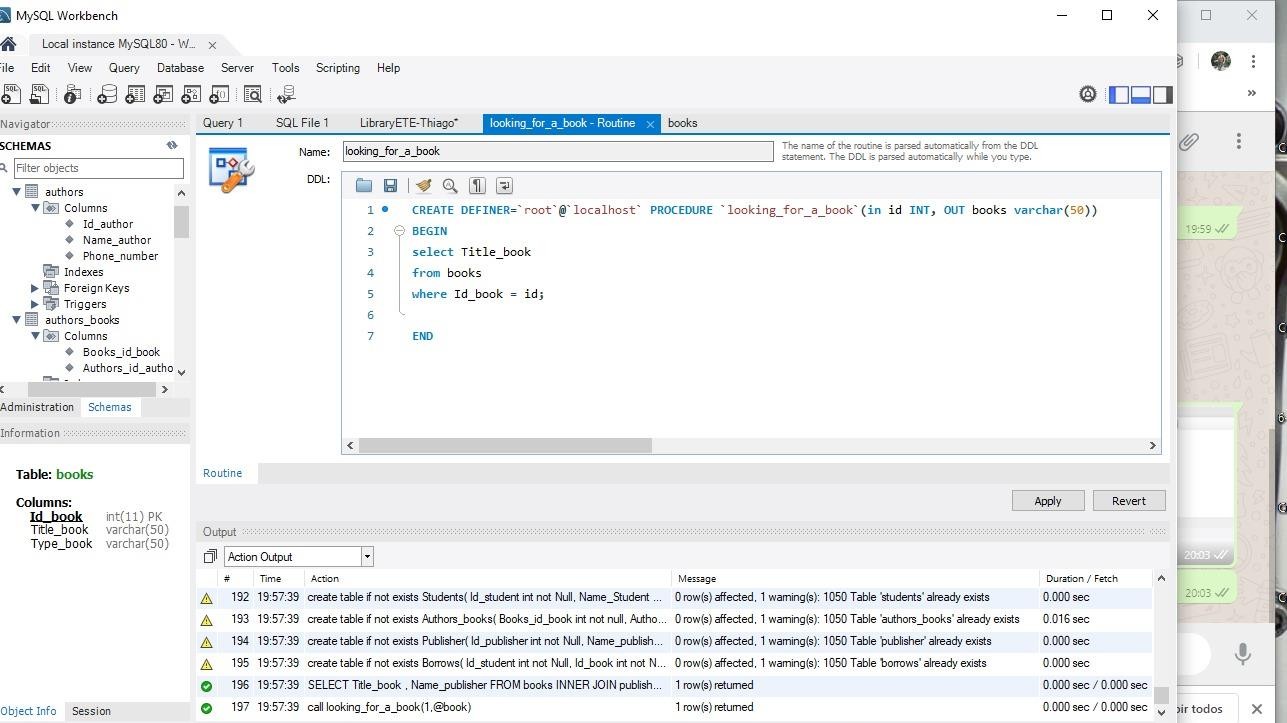
Thiago Almeida

* One view **per student** incorporating at least two tables and a WHERE clause; **5%**



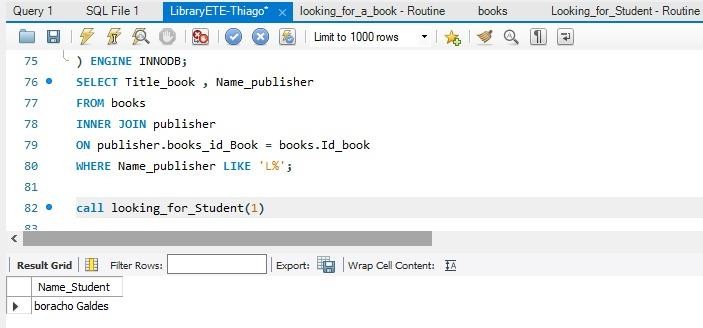
* One stored procedure **per student** that performs a multi-table SELECT query and includes at

least one IN and one OUT parameter; **5%**



* One stored procedure **per student** that performs a correlated subquery and includes at least

one IN parameter; **5%**



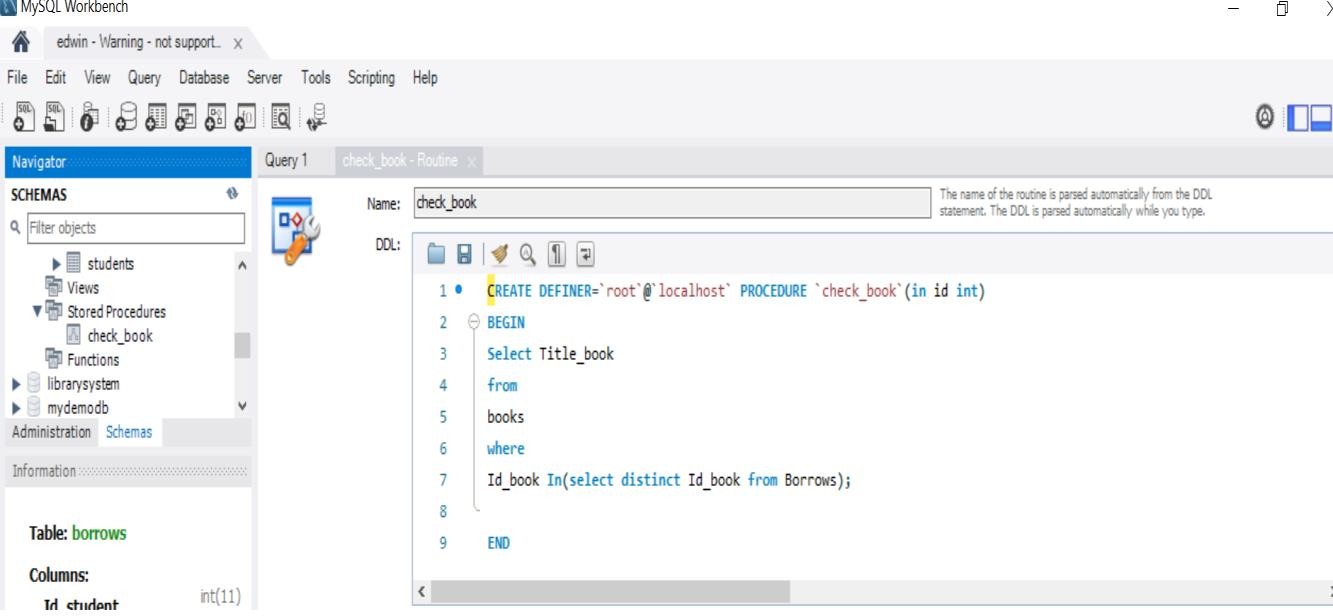
* One triggers **per group**, each of which must be a separate type (e.g. on UPDATE, on DELETE,

on INSERT);

\_ An example of each of the following:

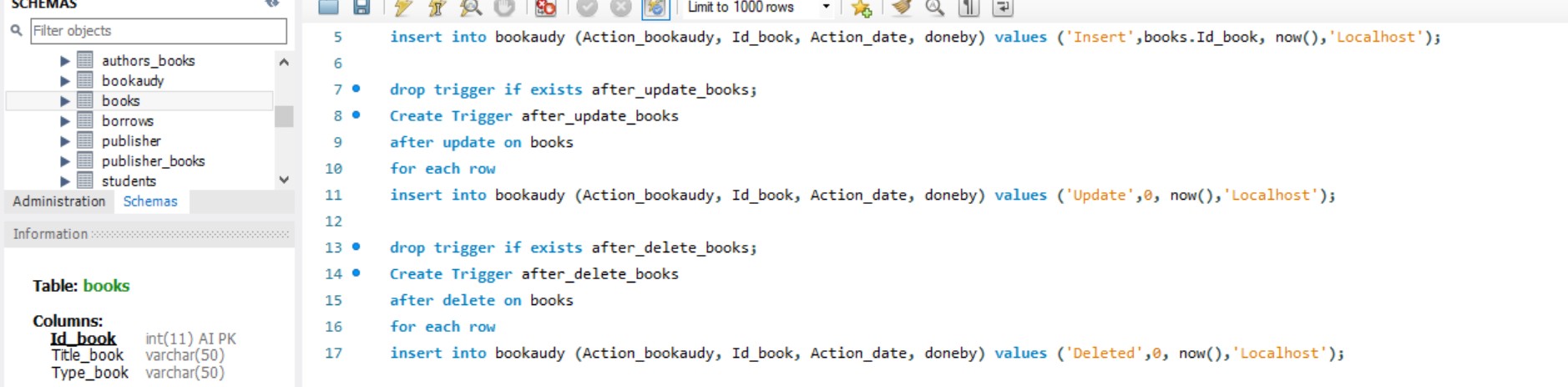
Edwin Garces

* One view **per student** incorporating at least two tables and a WHERE clause; **5%**



* One triggers **per group**, each of which must be a separate type (e.g. on UPDATE, on DELETE,

on INSERT);



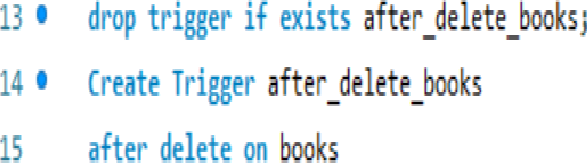
e u r r R u a e e sky •• •= I « 4 ua











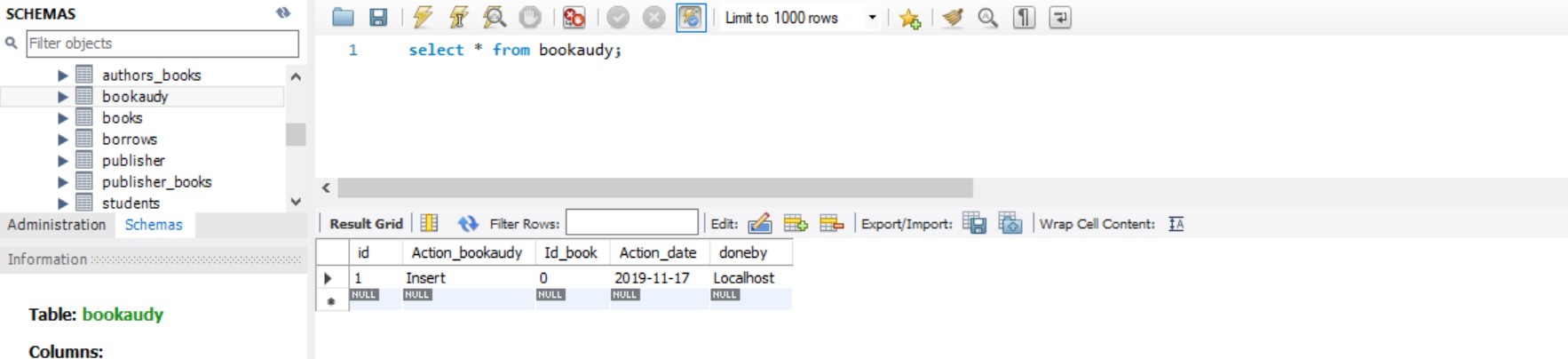




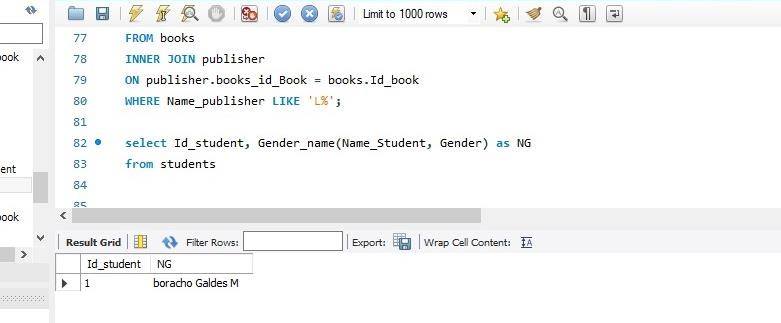


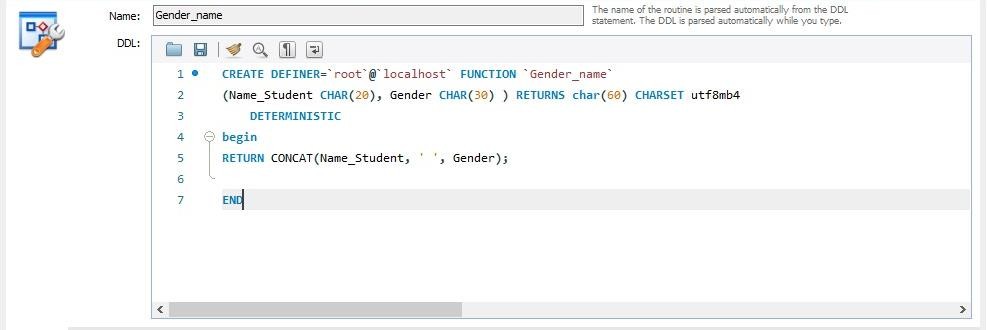


 fDlia



One stored function **per group**;



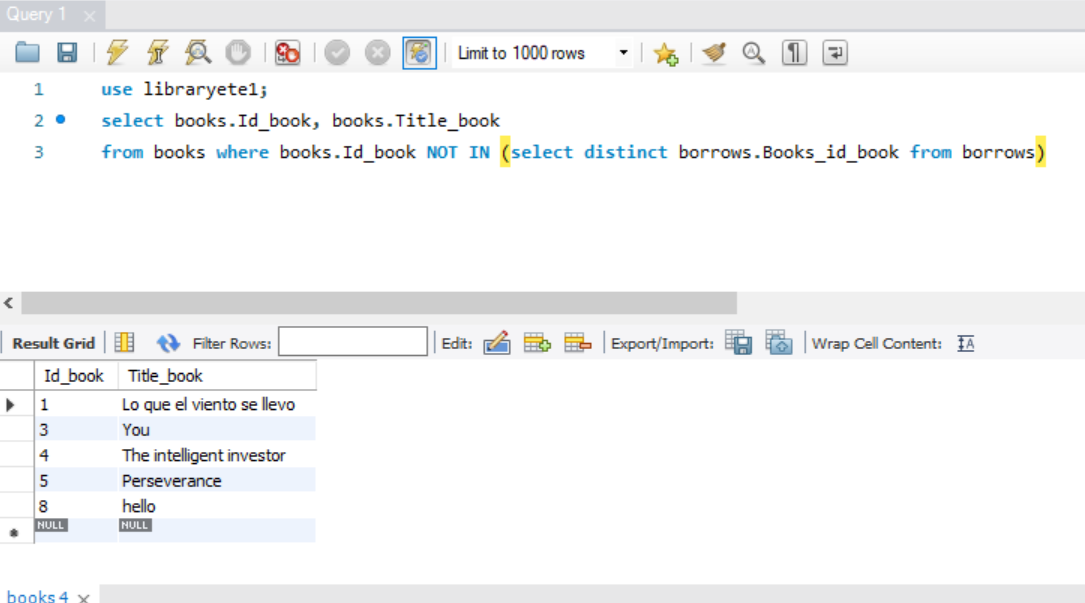


Part B

Not correlated subquery:

We are trying to find books that have not yet been borrowed by anyone.

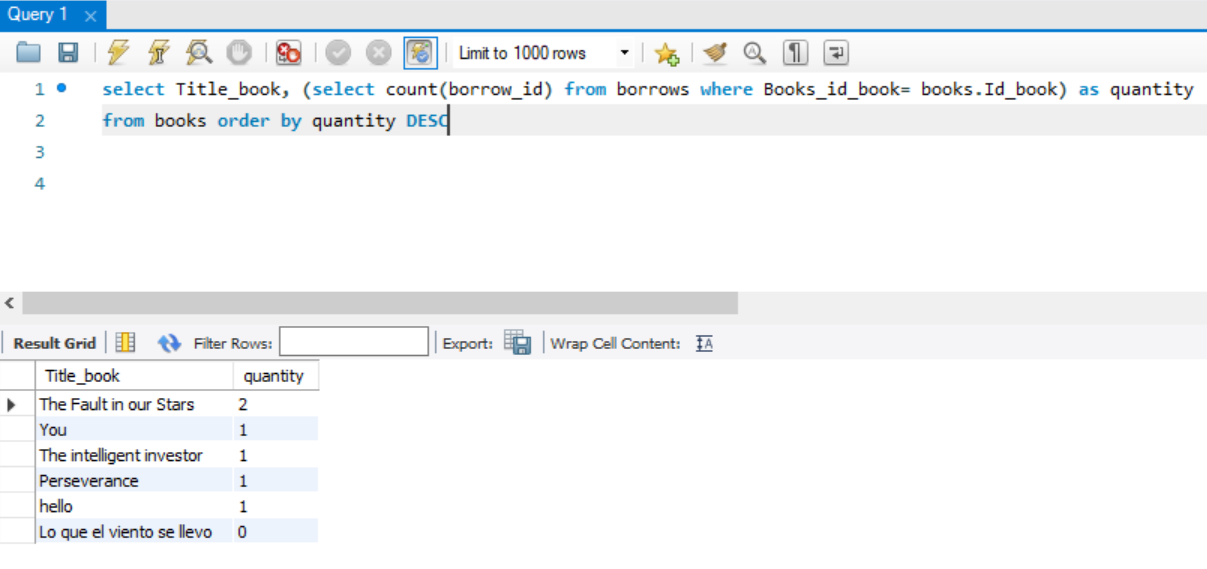
The subquery can still run without the main query which makes it a not correlated subquery



Correlated subquery:

In this query we are looking for the number of times every book has been borrowed. We have entered more data into the borrows table for testing purposes.

In this query, the subquery is correlated to the main query and it can not run on its own

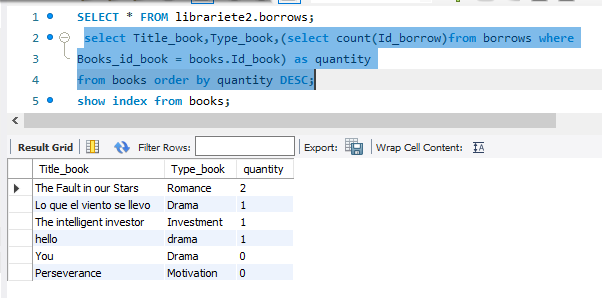


|  |
| --- |
| Transaction analysis form  **Transaction** In this transaction we are trying to get the number of times a title from the books table has been borrowed  **Transaction volume**  **Average**  3 per hour  **Peek**  10 per hour (between 10:00 and 17:00 Monday to Friday ) |
| |  |  |  |  | | --- | --- | --- | --- | | select Title\_book, (select count(borrow\_id) from borrows where Books\_id\_book= books.Id\_book) as quantity  from books order by quantity DESC | |  |  | | --- | --- | | Predicate  Join attribute  Ordering attribute  Grouping attribute  Built-in functions  Attribute updated | Book\_id\_book  Books\_id\_book= books.Id\_book  quantity  none  none  none | | |
| Transaction usage map |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Access | Entity | Type of entity | |  |  |  | | --- | --- | --- | | Per transaction | Per Hour | Per peek |   No of references | | | | 1  2 | Books  Borrows | R  R | 7  6 | 3  10 | 10  30 | | Total references | | | 13 | 13 | 40 |  |  |  | | --- | --- | |  |  | |

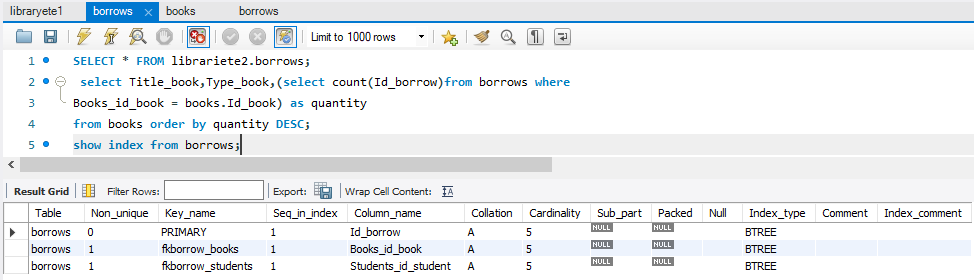
Analyse the query with the EXPLAIN command. Provide a detailed analysis containing:

– the state of the database before the query execution; 5%

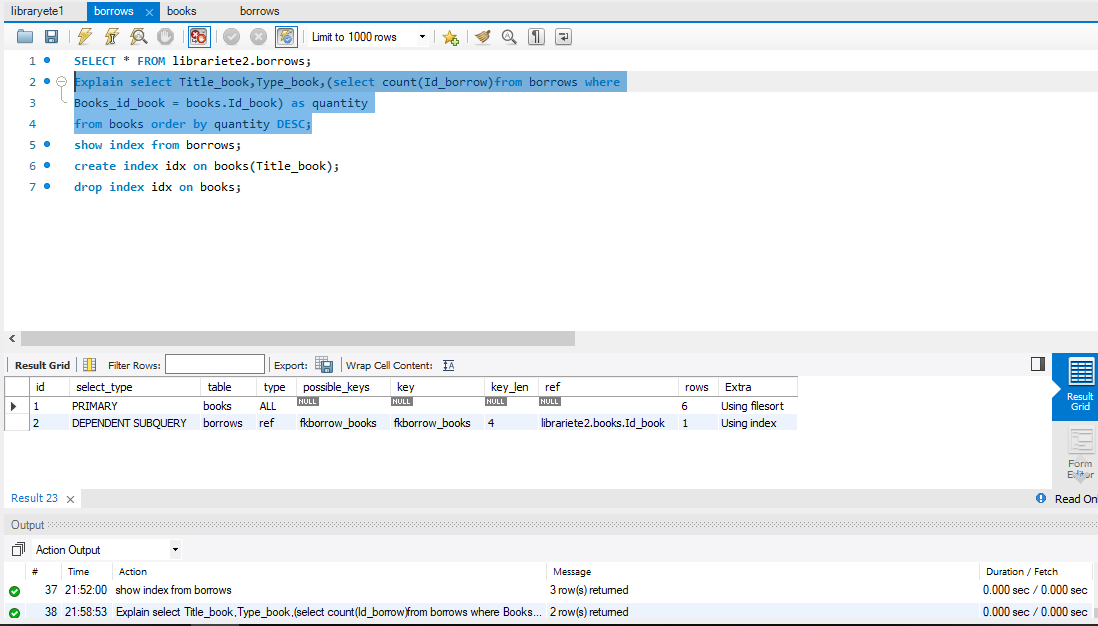
This is the out put for the query;



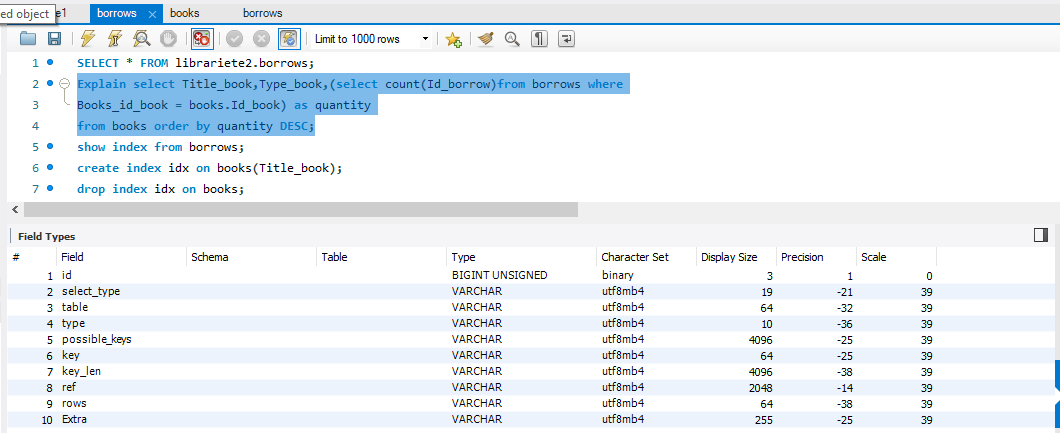
This is the show index command



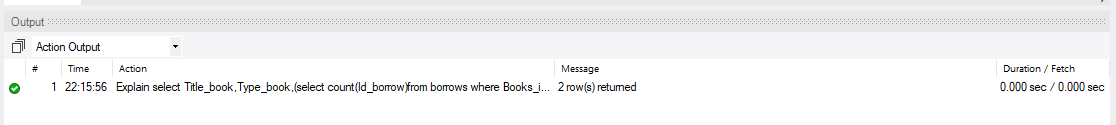
– the output of the EXPLAIN command; 5%



– the state of the database after the query execution; 5%



– the time taken for query execution 5%



– Comment on the actual outcome versus the expected outcome.

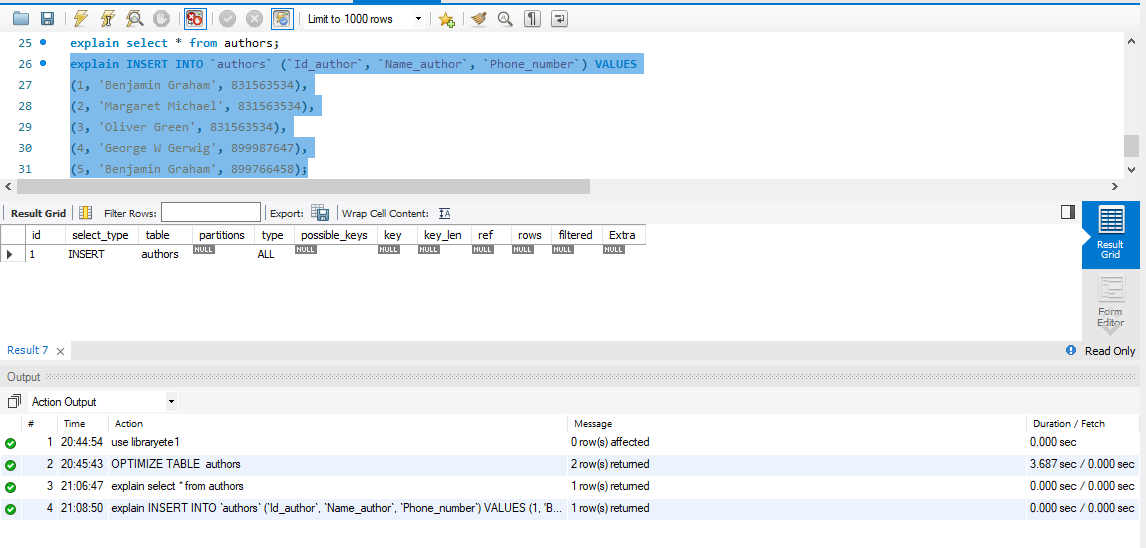
State whether or not there is a difference between what you expected and what you got. 5%

We don’t think there is a difference between what we are expecting and what we got everything occur as we expected.

-Create a query to perform an update, delete or insert operation. Propose an optimisation to the

query and/or database structure to improve the performance. Outline the effect your optimisation

can be expected to have. Test your query before and after the optimisation using the EXPLAIN

command. **10%**

Part C- group Work

**Eduard Cuadros 2017329**

1. What did you learn from working within a team?

We learn many things when we work as team, because if I do not know something my team can help it, or even I know something I can explain to them. We always support between us. For example I starting to make the Database and then one of them improving the data adding more information, and then, the another guy help us with the commands. We never lost the main idea, we are thinking in the same goal.

It is normal for comments and criticisms about our contributions, performance and results. Working as a group, I learned to accept them and see them as a way to improve is essential for our growth.

I have been working where trust is the most appropriate environment for all members to feel good, involved with our database and we could do our best. We learned to be honest and say things in the face avoiding tensions, with the only objective of doing the best teamwork

2. What would you do differently if you had to build it again?

I think I would do the same job, only with few changes that are not relevant I really liked the topic of the library, I would also choose the same topic. If I had to build it again I would further expand the database that could be adapted to any Irish library.

3. What did you find most difficult to implement or understand.

Actually was difficult understands the requirements. So I tried to search a lot commands online on Moodle they help me a lot . After check those presentations I got more knocking about all those queries. Now I feel more confident when handling the database.

Name: Thiago Almeida.

Student N: 2017201.

1. What did you learn from working within a team?

Answer.

I learned that is easier when we have a group where everyone works on the same path, and the job is well distributed.

1. What would you do differently if you had built it again?

Answer,

Asked more the teacher about his requirements, because it wasn’t clear for me.

1. What did you find most difficult to implement or understand?

Understand the requirements for me was a bit confuse to understand and we did not know which question was referred or related to de priviest question or not.

Thanks anyway it was very good to practice the knowledge that we got during the classes.