

Instituto Politecnico Nacional

Escuela Superior de Computo

Class:Distributed Data Base

Group: 3cm4

Professor: Hernandez Contreras Euler

Alumno: Urban Reyes Adan

Boleta: 2012630450

Email: [ayanializita@gmail.com](mailto:ayanializita@gmail.com)

Team: Transactions

Indice

[Introduccion 3](#__RefHeading___Toc502_1246705353)

[Instructions and Screen Shots 4](#__RefHeading___Toc820_1589454436)

[References 9](#__RefHeading___Toc498_1589454436)

Introduccion

A transaction is the execution of a program that update the data base. A transaction have four main properties that are:

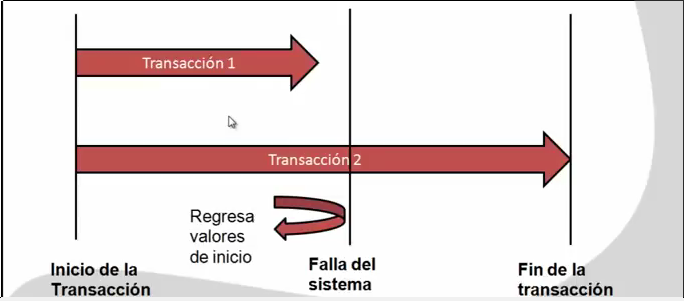
Atomicity: Is that a transaction is a unit indivisible so this execute complete or not will execute

Consistency.- This property it indicates that after executing a transaction, the database must be in the correct state (if all SQL statements properly then how the database is made all stay updated databases that are within of the transaction). If an error occurs before completing the transaction, you must return the system to the same state it was in before the transaction start.

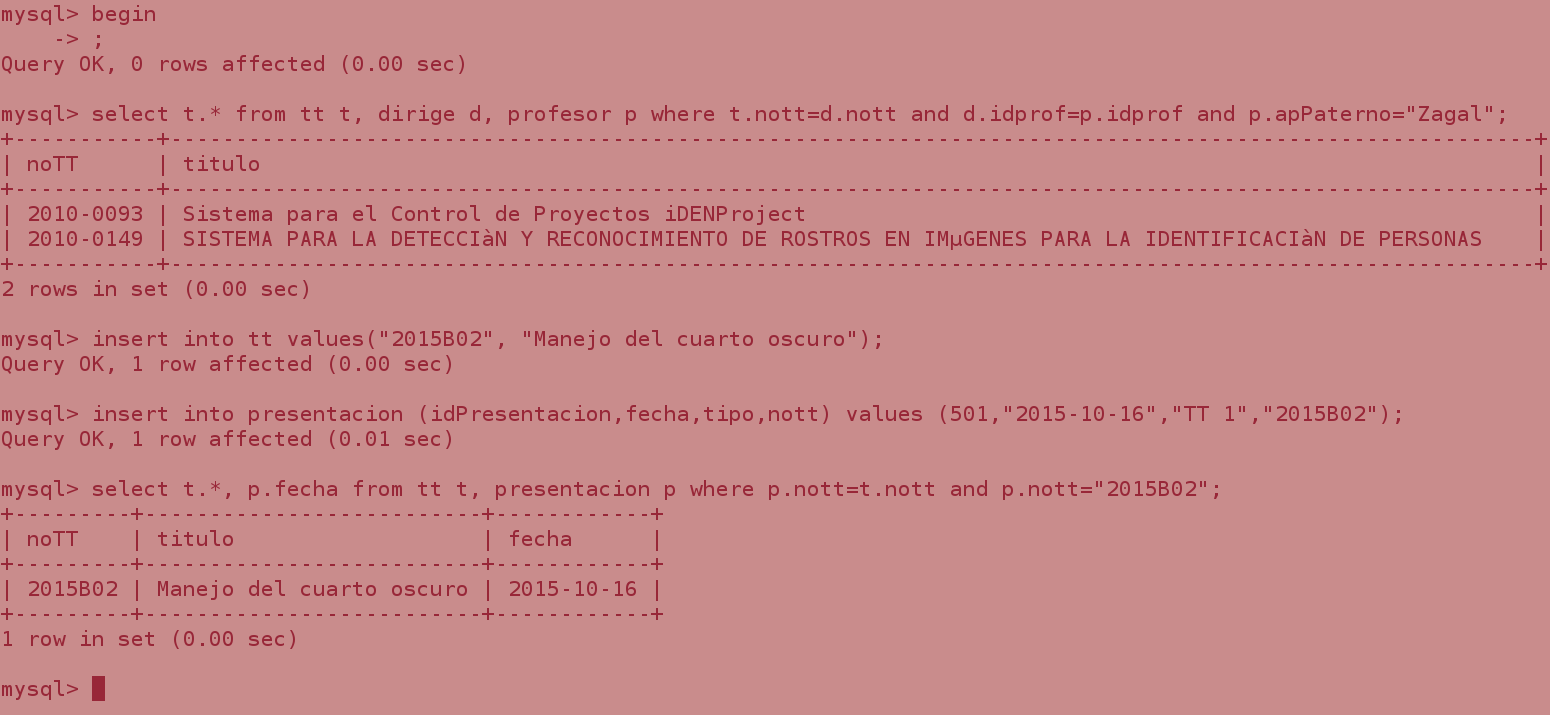
Isolation.- This property indicates that the behavior of a transaction is not affected by the fact that other transactions are executed at the same time .is a feature that allows me betwen of the table locks to update all the tables that are related wing transaction spite of having other executing instructions in other transactions.

Durability.- The effects of a transaction are permanent after recording. A look that is executing the transaction ends all changes are impermanent.

A simple transaction is one that contains instructions beginning, recording and abortion of the transaction. If there is a problem in the middle of the transaction is irretrievably erased, as an intermediate exist not recorded.



Instructions and Screen Shots



**begin;** --start the transaction

**select t.\* from tt t, dirige d, profesor p where t.nott=d.nott and d.idprof=p.idprof and p.apPaterno="Zagal";** --operation of read for the tables tt, dirige and profesor

**insert into tt values("2015B02", "Manejo del cuarto oscuro");** --operation of write for table tt

**insert into presentacion (idPresentacion,fecha,tipo,nott) values (501,"2015-10-16","TT 1","2015B02");** --operation of write for table presentacion

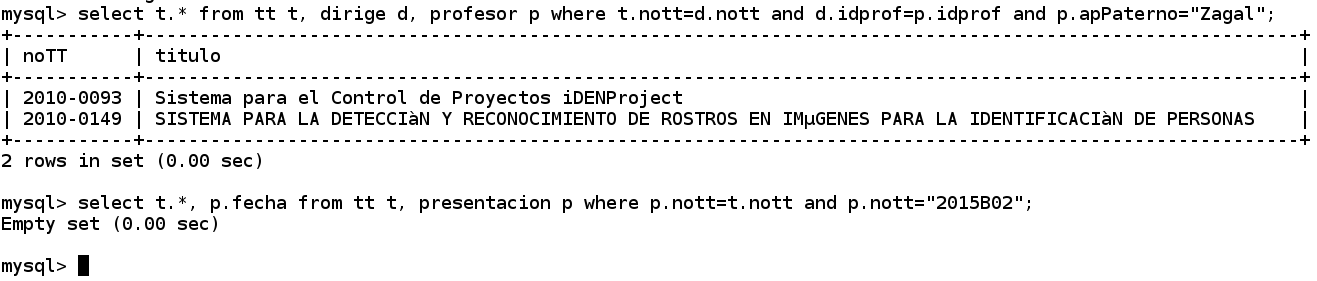
**select t.\*, p.fecha from tt t, presentacion p where p.nott=t.nott and p.nott="2015B02";** --operation of read for tables tt, presentacion

Then we conclude that

operation **select** is for read.

operation **insert, update** is for write.

If we open other terminal and make the same operations of read…

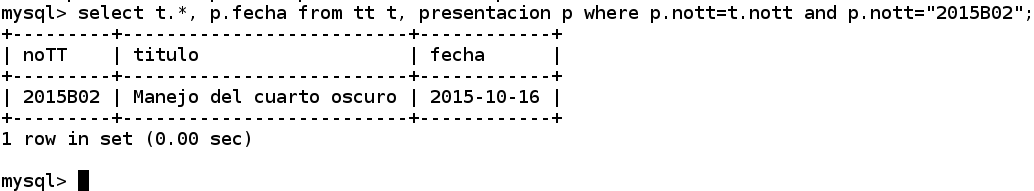
****

We can observe that second select no return anything, is because the first terminal (terminal color red) not save the changes. But if now make it.

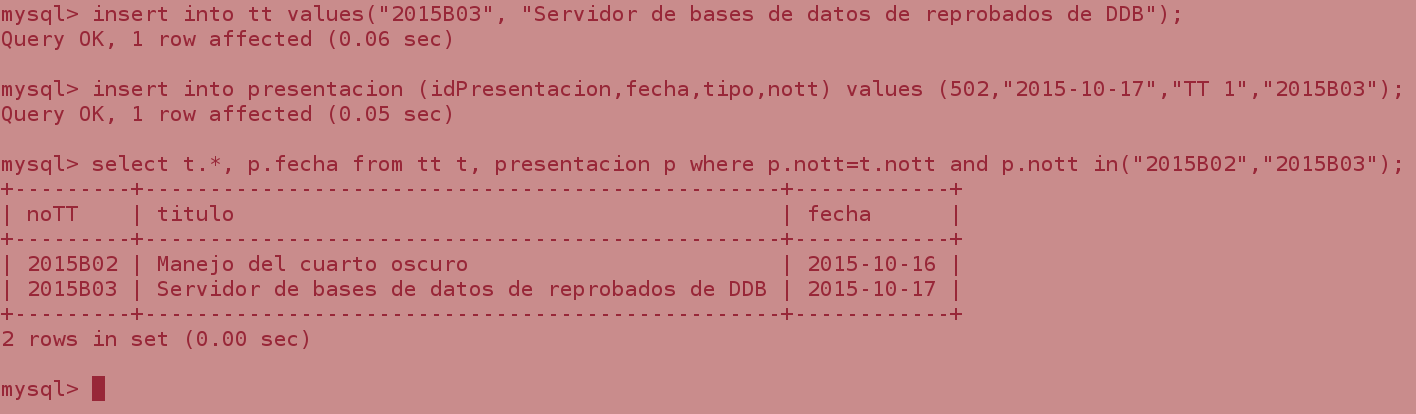


**commit;** --for save the changes make it on this transaction and finish the transaction (property of Consistence)

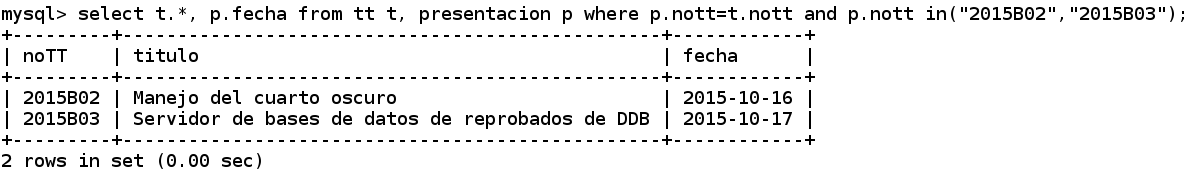
And the again make the read (property of Durability)



if we make this operations:

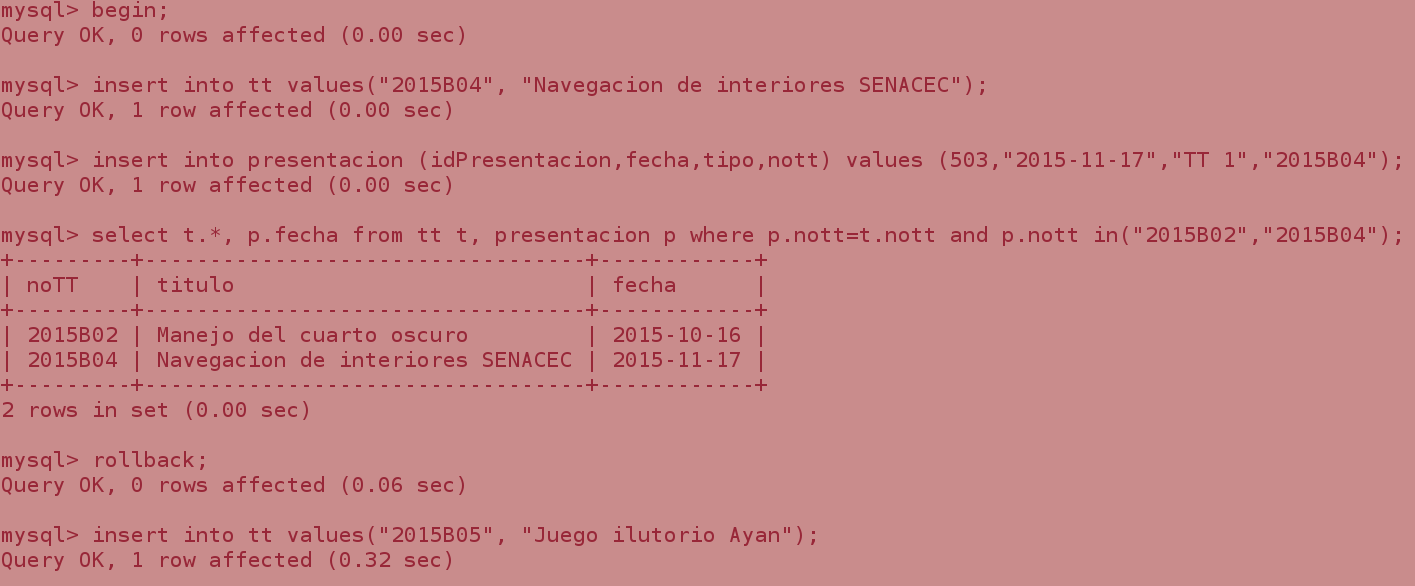


and make the operation of selection on second terminal (white)



the select return same result of first terminal (red).

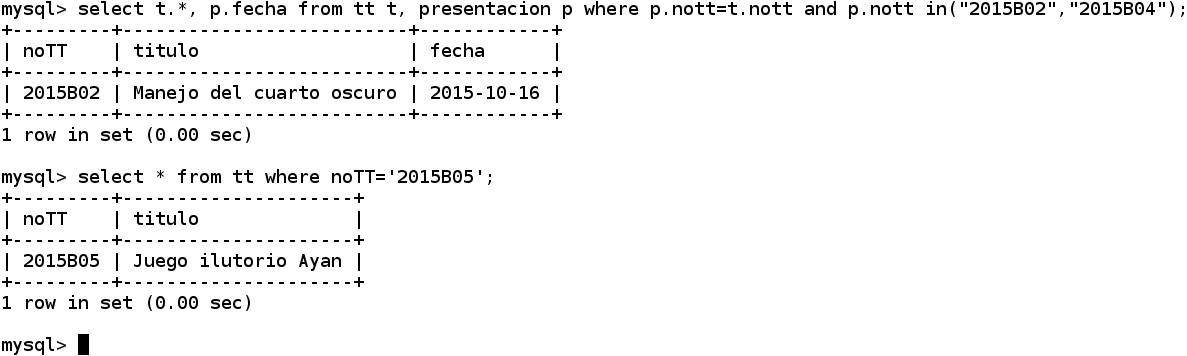
This is because when use the commit the transaction end. Then we never start again the transaction.



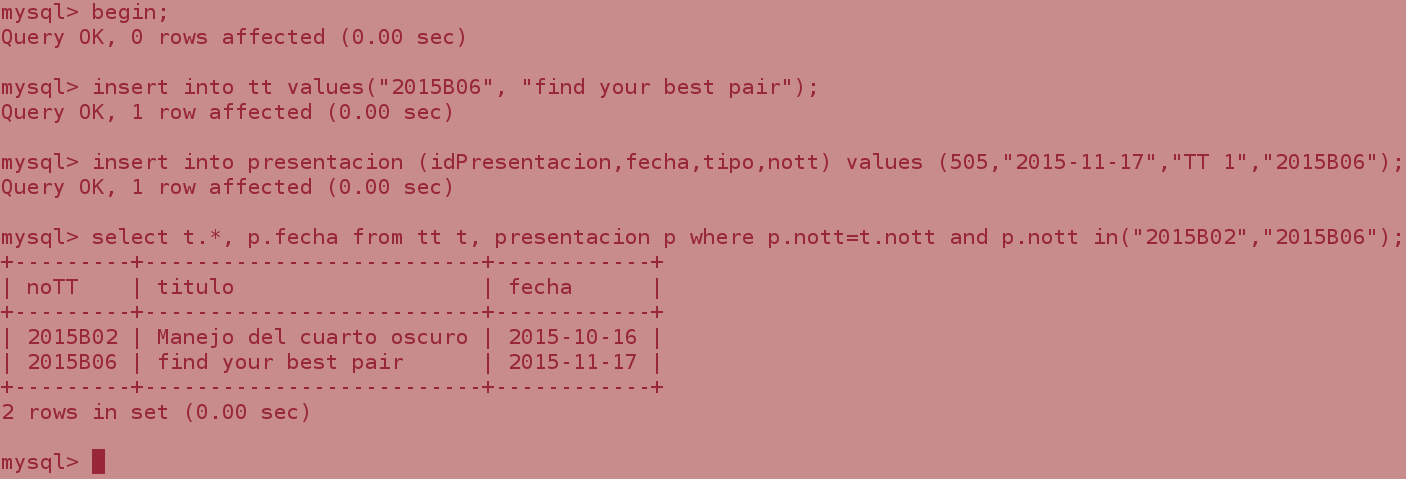
Then start new transaction. And make someone inserts but now watch the command **rollback;**

**rollback;** --remove all operations of write and finish transaction.

Then if we make the operations of read on second terminal (white)

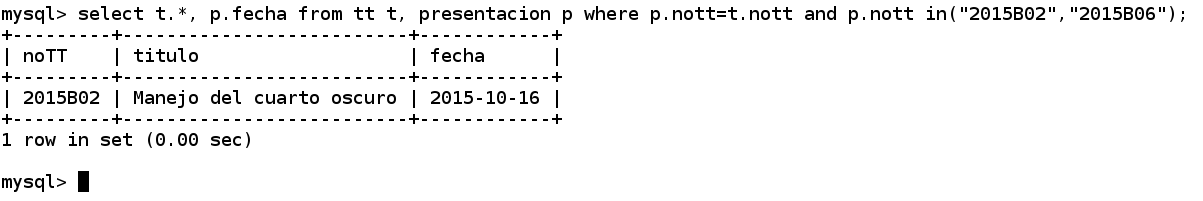


OK Now we simulate system failed. For this make the next insertions.

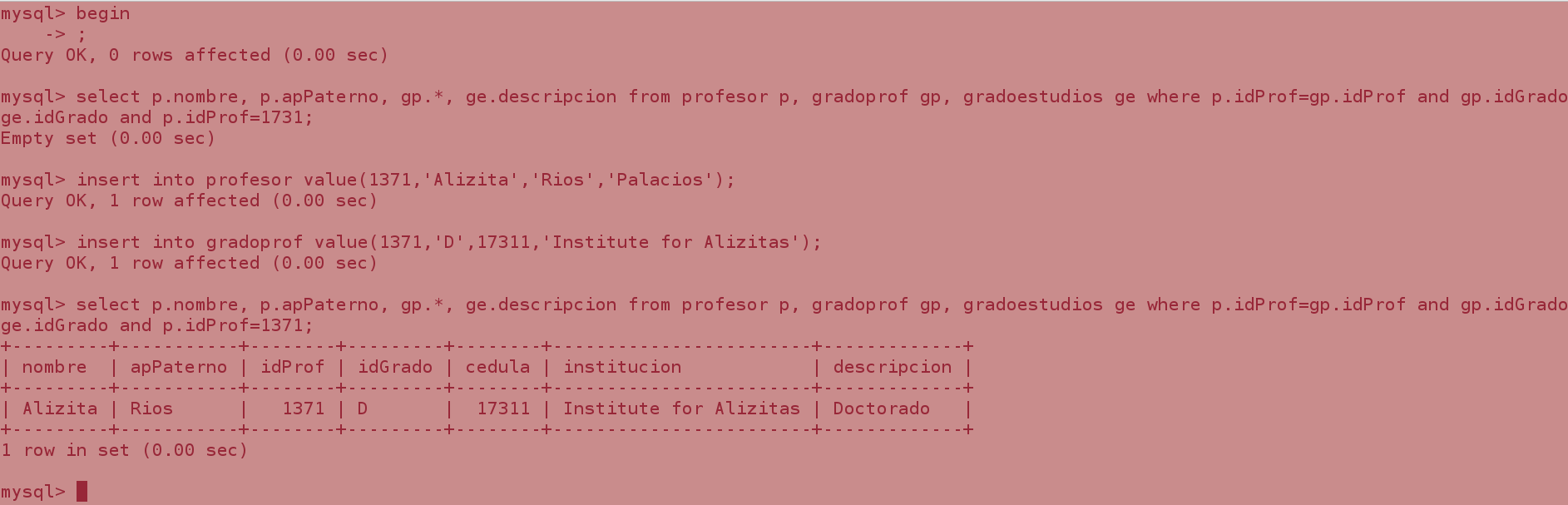


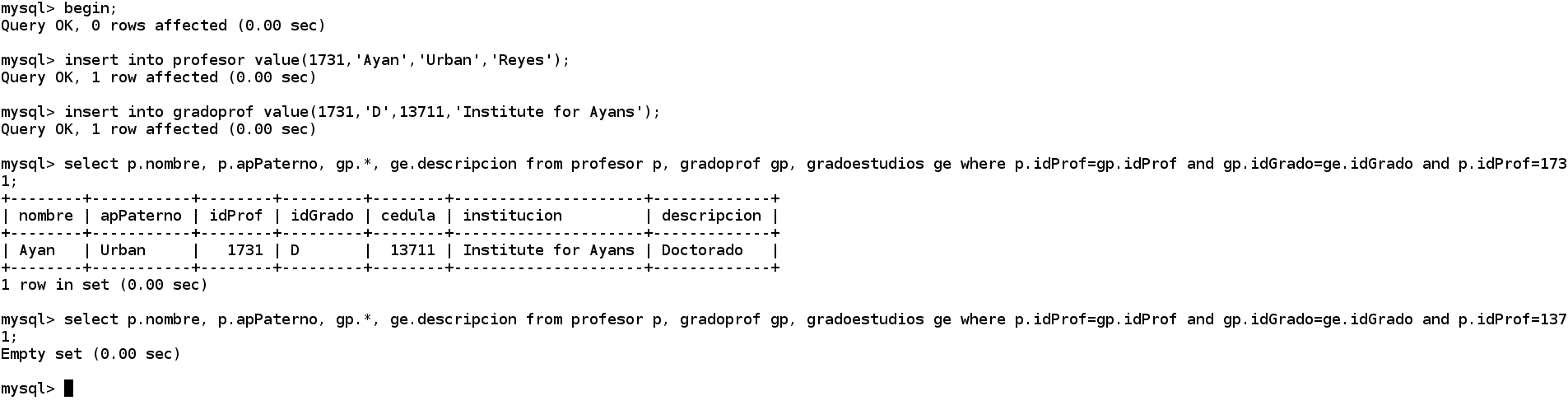
Now we close the terminal any make commit or rollback or other thing

and make read on second terminal (white).

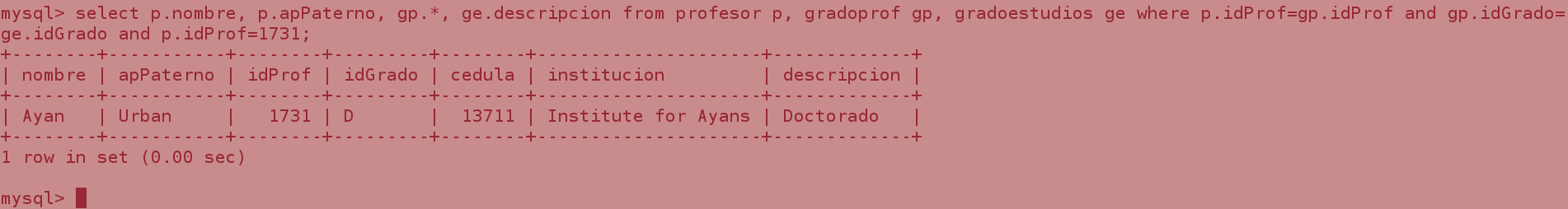


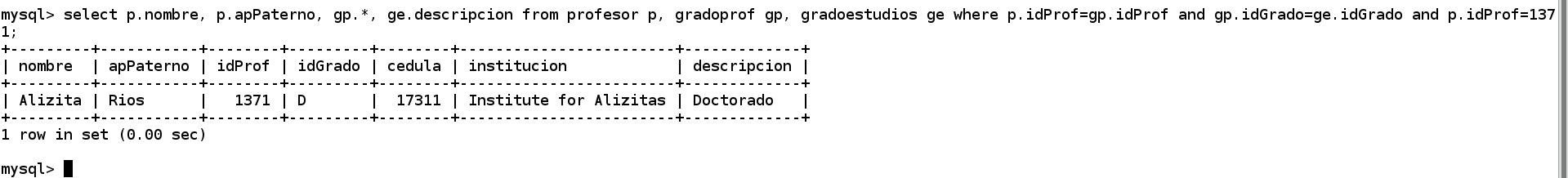
How can observe no return the inserts make on first terminal (red) (property atomicity)



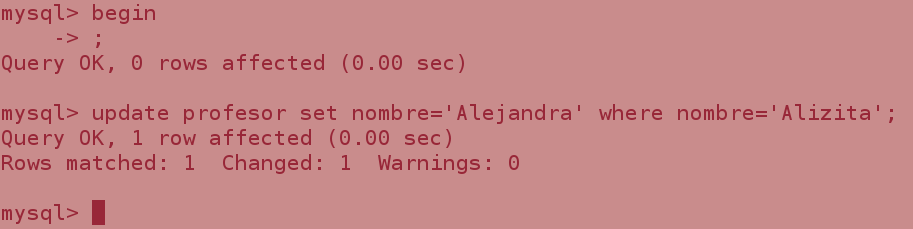


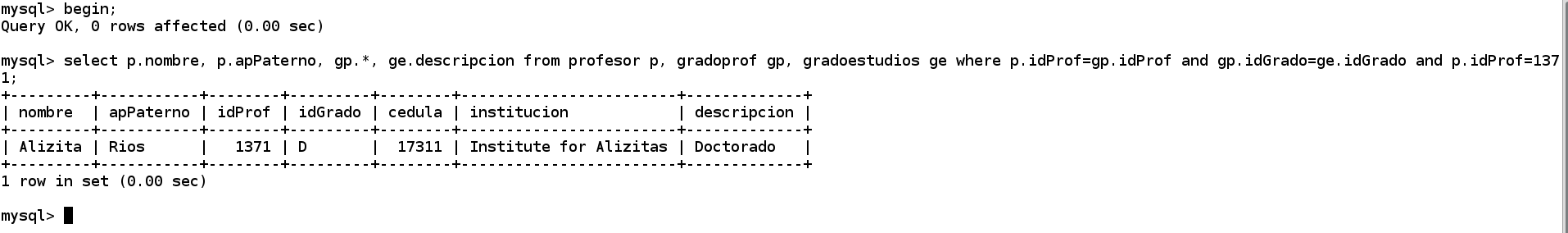
How can observe the someone select not return any because have make **commit;** if we make the commit and again the select on terminals;





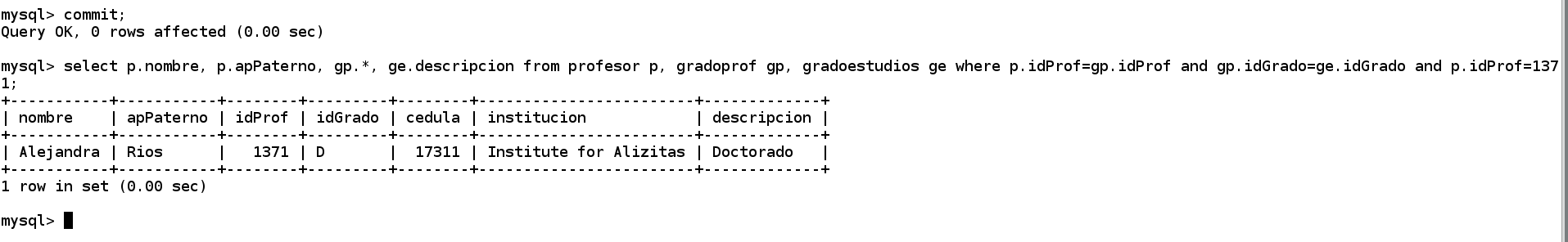
Ok now we can the read on any terminal.





We can observe that changes on first terminal (red), (update) no is watch on second terminal (white), this is by property (isolation).

If we make the changes then we need the commit on the two terminals.



References

<http://dev.mysql.com/doc/refman/5.7/en/commit.html>

http://zetcode.com/databases/mysqltutorial/transactions/