# Task C9.2.1

## Scenario 1:

Run first statement of T2:

Graphical user interface, text, application

Description automatically generated

This makes changes to the Quantity on hand and reduce it by 2 for products table where ProductNumber is 1. But, as we are not executing commit command, changes made by this statement will only affect T2 and not T1 as seen in pictures bellow.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

And currently we only have executed first command for T2 there are no records for OrderNumber 945 so table becomes empty.

## Scenario 2:

After executing remaining commands in T2 and checking results in T1, results are still same as there are no change in tables as we haven’t committed those changes in T2. Shown in picture bellow.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

## Scenario 3:

Copying all the statements of T1 into T2:

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

From these pictures we can see that changes made by previous UPDATE and INSERT statements are clearly visible in SELECT statements as changes are visible locally in that particular transection T2 even if not committed to database.

## Scenario 4:

After committing T2 if we run commands in T1 again we cannot see any changes as we are currently in repeated-read isolation level.

## Scenario 5:

After executing COMMIT statement into T1 we are now able to see changes made by T2 into T1.

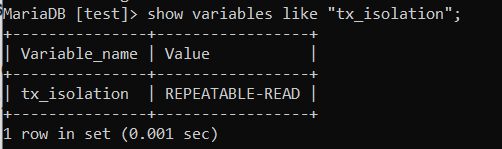
Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

## What isolation level are you working at?



## When does a transaction see the changes made?

Transection only see changes when COMMIT command gets executed in the transection making changes. Otherwise changes can only be visible for that single particular transection not for all other transections.

## Why can’t T1 see the changes of T2 when T2 commits?

In Repeatable-read isolation level changes made in one instance of database or transection will only be visible in other transection if commit statement gets executed in Transaction making changes.

## What do we mean by ‘repeatable read’ and do we have phantoms here in MySQL?

Repeatable read is type of isolation level that locks rows for specific transection until transection gets finished and no other than current transection can make changes to those rows.

## What does the SQL standard say about phantoms and Repeatable Read Isolation level?

Phantoms mans when two identical queries to same database gives different set of rows in result this results in dirty reads to database and data inconsistency. Repeatable read can also have phantoms just like other isolation levels such as Read Committed, Read Uncommitted and non-repeatable read. Only Serializable isolation avoids Phantoms.

