

Scenarios :

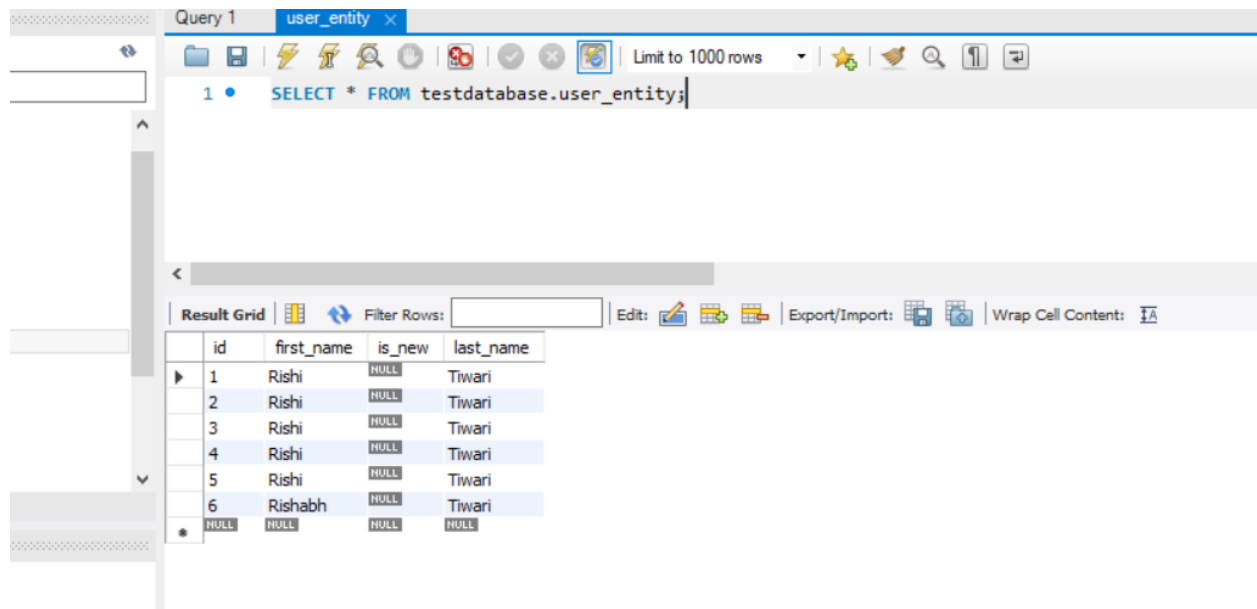
- First Scenario :

/* Test 1 — no exception thrown, everything committed

* Outcome : Happy case

*/

DB output : All items will get working.



The screenshot shows a database query tool interface. At the top, a tab labeled 'Query 1' is active, showing a SQL query: `SELECT * FROM testdatabase.user_entity;`. Below the query editor, a 'Result Grid' displays the query results. The grid has four columns: 'id', 'first_name', 'is_new', and 'last_name'. It contains six rows of data, all with 'is_new' set to 'NULL'. The first five rows have 'first_name' as 'Rishi' and 'last_name' as 'Tiwari', while the sixth row has 'first_name' as 'Rishabh' and 'last_name' as 'Tiwari'. A summary row at the bottom shows counts for each column.

id	first_name	is_new	last_name
1	Rishi	NULL	Tiwari
2	Rishi	NULL	Tiwari
3	Rishi	NULL	Tiwari
4	Rishi	NULL	Tiwari
5	Rishi	NULL	Tiwari
6	Rishabh	NULL	Tiwari
Σ	NULL	NULL	NULL

- Second Scenario :

/*

* Test 2 — exception thrown in an inner transaction,

* no handling in outer transaction

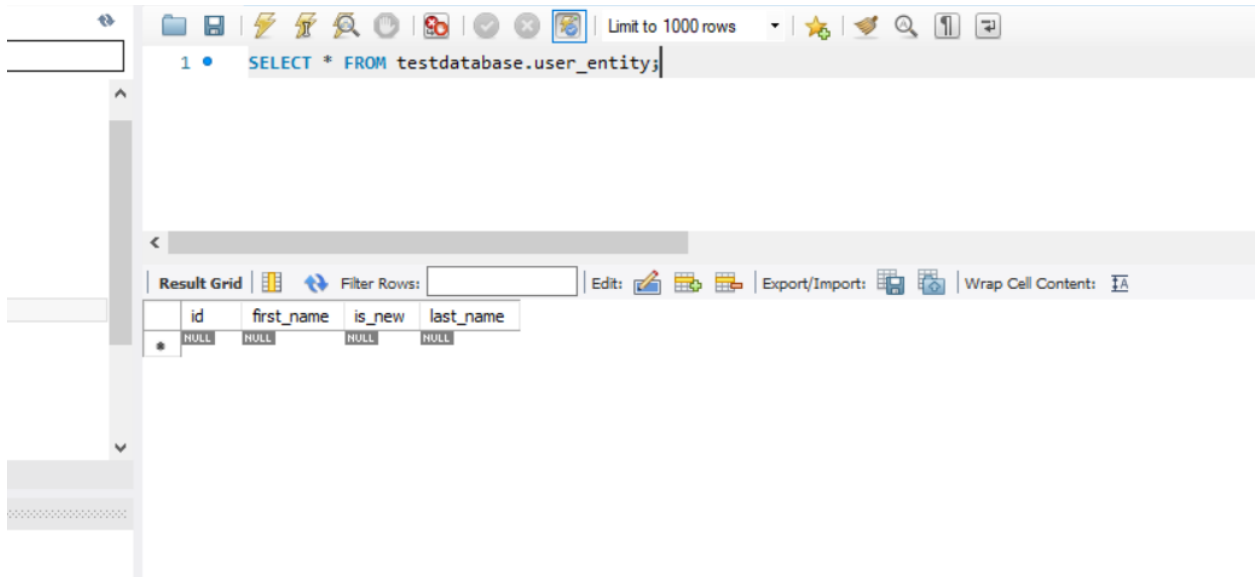
*

* Outcome: Everything got rollback.

*/

*/

DB output : All items will get rollbacked.

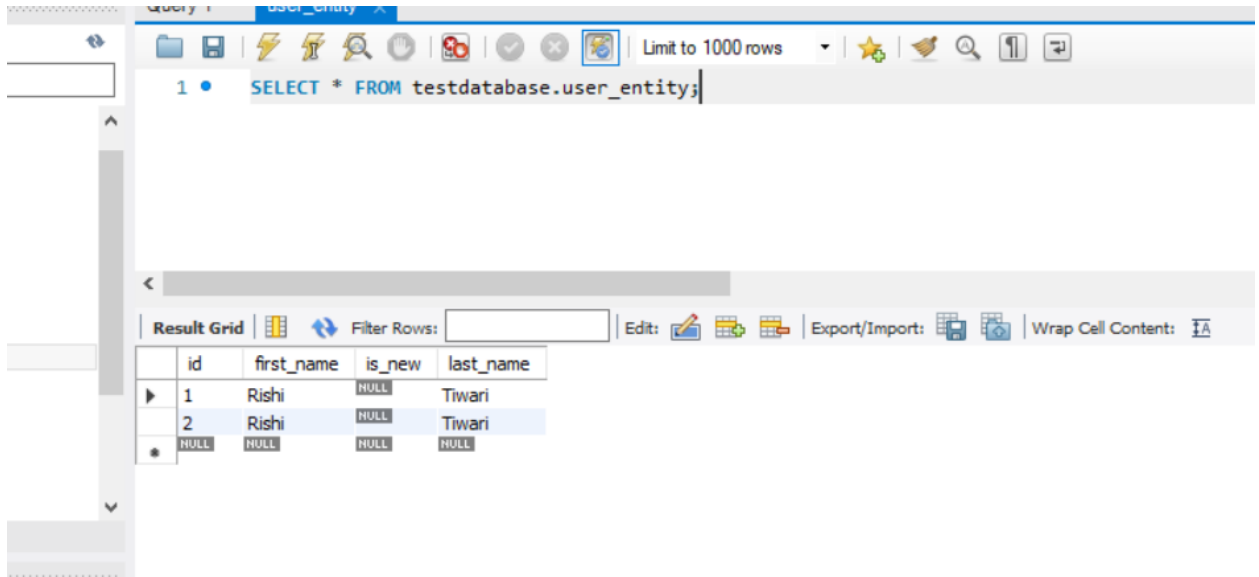


- Three Scenario :

/*

- * Test 3— exception thrown in an inner transaction,
 - * noRollbackFor=TransactionFailureException
 - * in outer transaction
 - * Outcome : exception throw on 3 item,
 - * last 2 will persist in this scenario
- */

DB output :Rollback will happen for current item, last two will not get rollback.



- Fourth Scenario :

/*

- * Test 4— exception thrown in an inner transaction,

- * try/catch block in outer transaction

- *

- */

DB output :Rollback will happen for current item, first two items will not get rollback.

The screenshot shows a database management interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is entered in a text area: `1 • SELECT * FROM testdatabase.user_entity;`. Below the query, there is a 'Result Grid' section. It includes a 'Filter Rows:' input field and an 'Edit:' button. The result grid itself is a table with the following data:

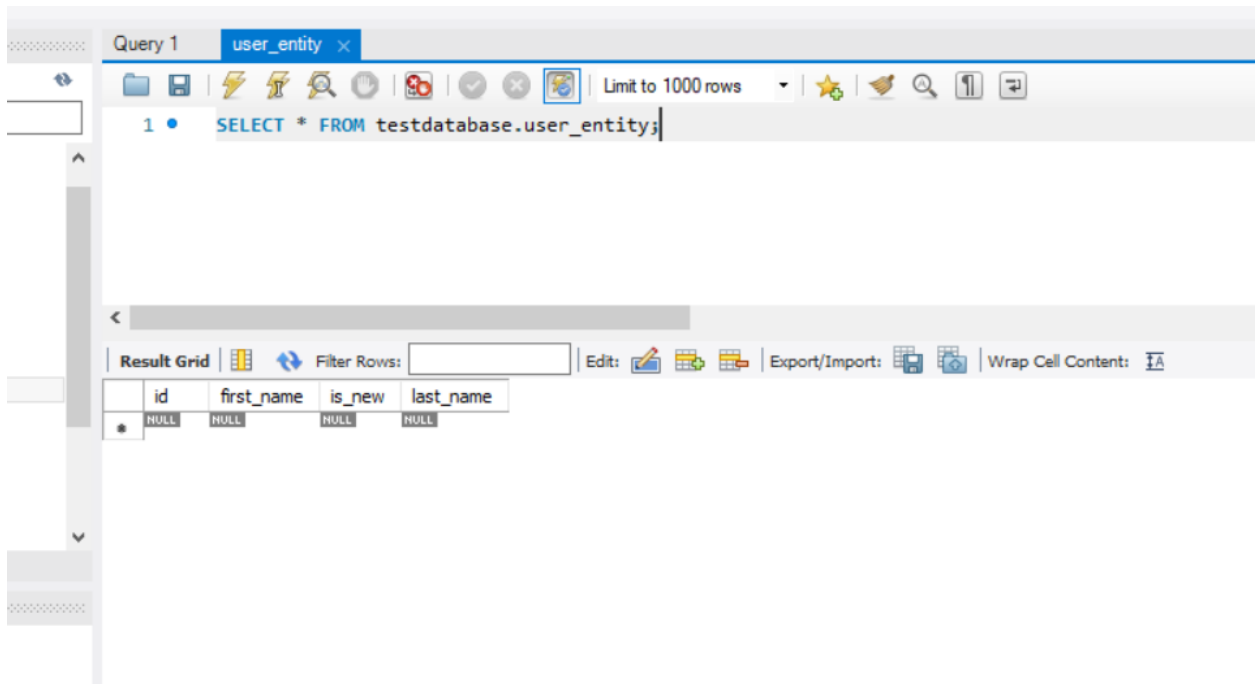
	id	first_name	is_new	last_name
	1	Rishi	NULL	Tiwari
▶	2	Rishi	NULL	Tiwari
*	NULL	NULL	NULL	NULL

- Fifth Scenario :

/*

- * Test 5 — exception (ObjectOptimisticLockingFailureException)
- * thrown in an inner transaction,
- * no handling in outer transaction
- *
- * Outcome: Everything got rollback
- */

DB output :Rollback will happen to all the elements.

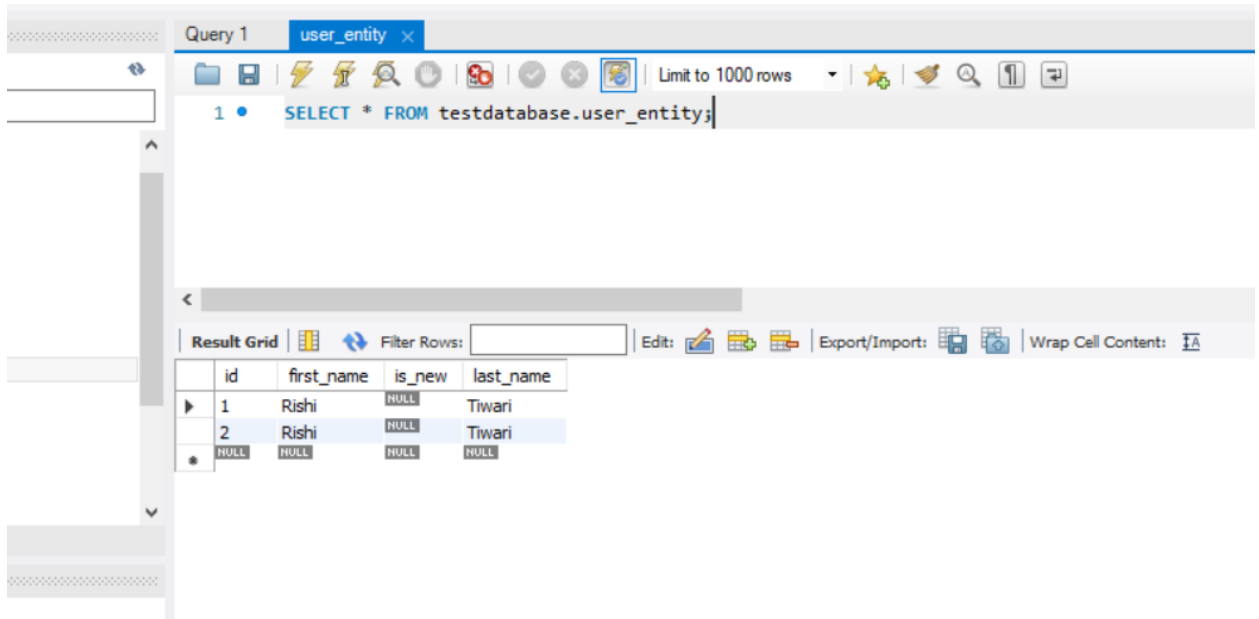


- Sixth Scenario :

/*

- * Test 6— exception thrown in an inner transaction,
 - * noRollbackFor=TransactionFailureException
 - * in outer transaction
 - *
 - * Outcome : exception throw on 3 item,
 - * first 2 will persist in this scenario
- */

DB output : Rollback will happen to current item only..last two items will persist.



- Seventh Scenario :

/*

* Test 7—

*

* Trying to update one item in outer transaction

*

* exception thrown in an inner transaction,

* noRollbackFor=TransactionFailureException

* in outer transaction

*

* Outcome : exception throw on 3 item,

* first 2 will persist in this scenario

*/

DB output : Rollback will happen to current item only..last two items will persist along with first transaction update.

Note : This one will help with the current proposed solution by me.

The screenshot shows the SQL query editor with the query `SELECT * FROM testdatabase.user_entity;` entered. Below the query, the "Result Grid" tab is active, displaying the following data:

id	first_name	is_new	last_name
1	Rishabh	Y	Tiwari
2	Rishi	NULL	Tiwari
3	Rishi	NULL	Tiwari
NULL	NULL	NULL	NULL