

1) It gives null values for 2nd and 3rd query and the 1st query doesn't change the productonhand number

Limit to 1000 rows

```

1 • Use salesordersexample;
2
3 • SELECT * FROM Products WHERE
4   ProductNumber=1;
5 • SELECT * FROM Orders WHERE

```

Result Grid

	ProductNumber	ProductName	ProductDescription	RetailPrice	QuantityOnHand	CategoryID
▶	1	Trek 9000 Mountain Bike	NULL	1200.00	6	2
*	NULL	NULL	NULL	NULL	NULL	NULL

Query 1 x 00 SalesOrdersStructureMy 01 SalesOrdersDataMY

Limit to 1000 rows

```

1 • Use salesordersexample;
2
3 • SELECT * FROM Products WHERE
4   ProductNumber=1;
5 • SELECT * FROM Orders WHERE

```

Result Grid

	OrderNumber	OrderDate	ShipDate	CustomerID	EmployeeID
*	NULL	NULL	NULL	NULL	NULL

Result Grid

Query 1 x 00 SalesOrdersStructureMy 01 SalesOrdersDataMY

Limit to 1000 rows

```

4 • SELECT * FROM Orders WHERE
5   OrderNumber=945;
6 • SELECT * FROM Order_Details
7   WHERE OrderNumber=945;
8

```

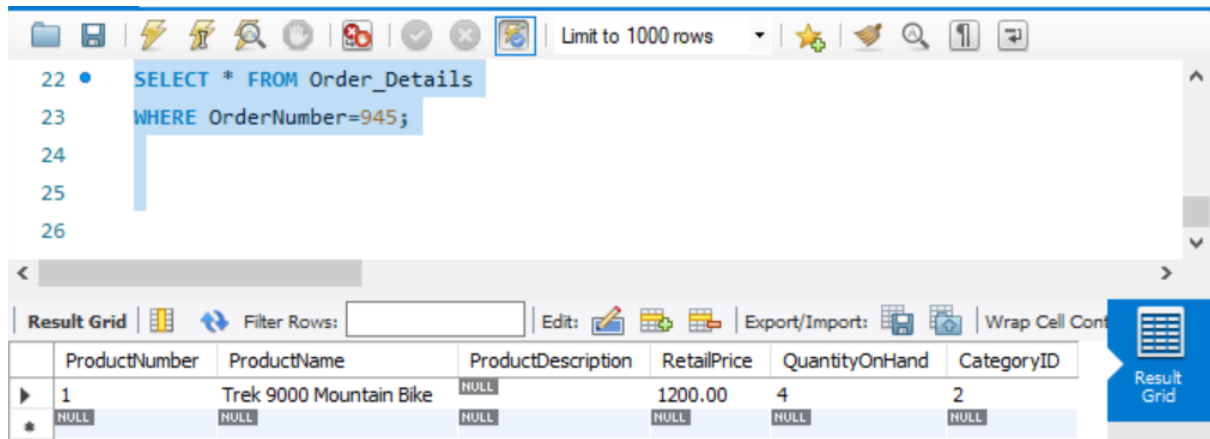
Result Grid

	OrderNumber	ProductNumber	QuotedPrice	QuantityOrdered
*	NULL	NULL	NULL	NULL

Result Grid

2) Still no change in output

3) Now all changes can be seen in their respective tables in right instance

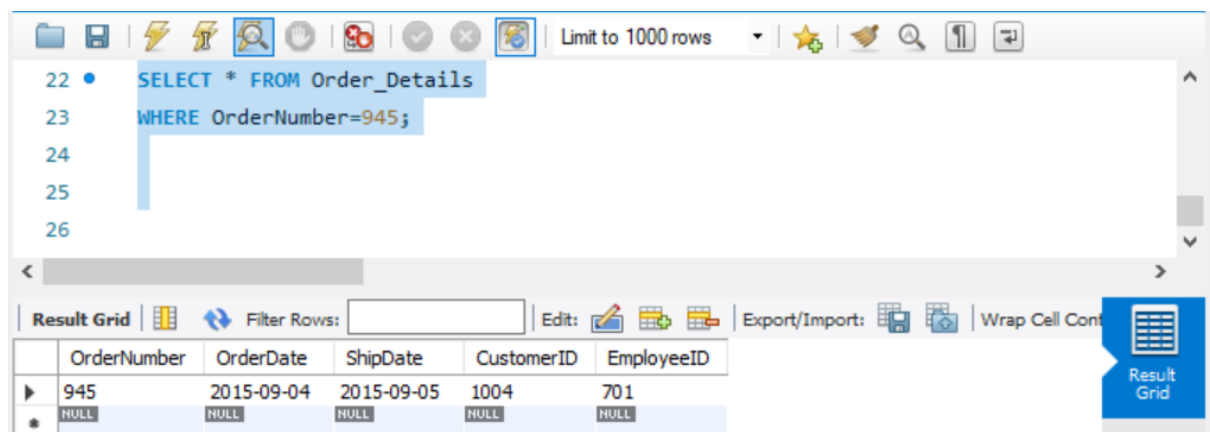


The screenshot shows a SQL query window with the following text:

```
22 • SELECT * FROM Order_Details
23 WHERE OrderNumber=945;
24
25
26
```

Below the query window is the 'Result Grid' tab. It displays a table with the following data:

	ProductNumber	ProductName	ProductDescription	RetailPrice	QuantityOnHand	CategoryID
▶	1	Trek 9000 Mountain Bike	NULL	1200.00	4	2
*	NULL	NULL	NULL	NULL	NULL	NULL

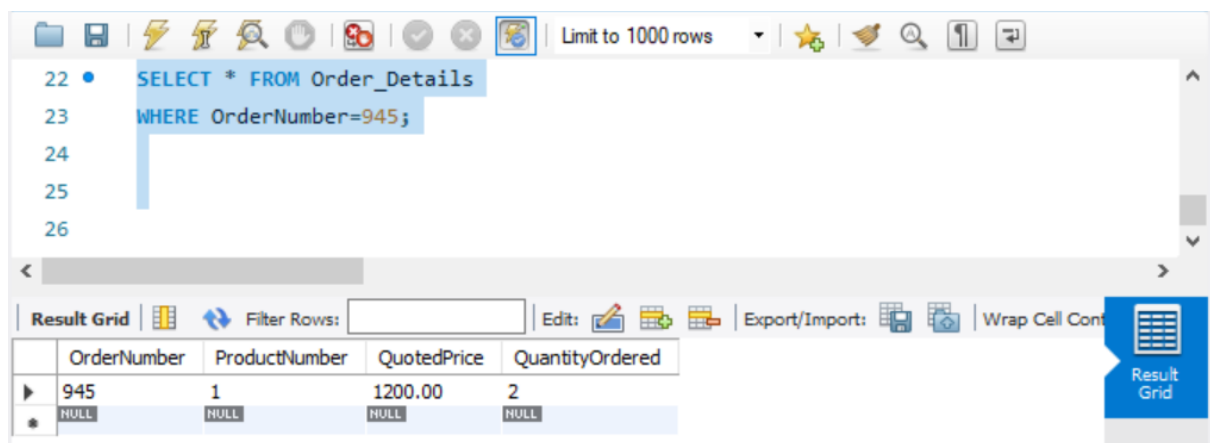


The screenshot shows a SQL query window with the following text:

```
22 • SELECT * FROM Order_Details
23 WHERE OrderNumber=945;
24
25
26
```

Below the query window is the 'Result Grid' tab. It displays a table with the following data:

	OrderNumber	OrderDate	ShipDate	CustomerID	EmployeeID
▶	945	2015-09-04	2015-09-05	1004	701
*	NULL	NULL	NULL	NULL	NULL



The screenshot shows a SQL query window with the following text:

```
22 • SELECT * FROM Order_Details
23 WHERE OrderNumber=945;
24
25
26
```

Below the query window is the 'Result Grid' tab. It displays a table with the following data:

	OrderNumber	ProductNumber	QuotedPrice	QuantityOrdered
▶	945	1	1200.00	2
*	NULL	NULL	NULL	NULL

4)It still shows null

5) Now all of the query results come as expected (giving the exact same results which had been seen in part3)

The isolation used here is REPEATABLE-READ.

For transaction1, it saw changed when both transaction2 and transaction1 had committed. For transaction2, it saw changes before committing but after writing the transaction 1's statements on it.

Because the database is following isolation level of repeatable reads where any data read cannot change, and so no matter how many times it is read, it will find the previously read data in place, unchanged, and available to read.

It is an isolation level where a transaction reading the same item within the same transaction always returns the same result. Here phantom did pop up when we committed the change in transaction1.

According to SQL, phantoms are the new rows that pop up when new rows are added in middle of another transaction reads. It can occur even in the Repeatable read isolation level.