Transactions in Practice

Start a new transaction. Then update salaries table which have emp_no = 201774 and rollback the changes in the company database.

Output: update operation would not impact the table and the select statement would produce the result.

```
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM SALARIES
    -> WHERE EMP_NO = 201774;
+--------+--------+------------+------------+
| emp_no | salary | from_date  | to_date    |
+--------+--------+------------+------------+
| 201774 |  44000 | 1994-12-17 | 1995-12-17 |
| 201774 |  46263 | 1995-12-17 | 1996-12-16 |
| 201774 |  46859 | 1996-12-16 | 1997-03-04 |
+--------+--------+------------+------------+
3 rows in set (0.00 sec)

mysql> mysql> Ctrl-C -- exit!

mysql> mysql> Ctrl-C -- exit!

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE SALARIES SET SALARY = SALARY*(1.1) WHERE EMP_NO=201774;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM SALARIES
    -> WHERE EMP_NO = 201774;
+--------+--------+------------+------------+
| emp_no | salary | from_date  | to_date    |
+--------+--------+------------+------------+
| 201774 |  44000 | 1994-12-17 | 1995-12-17 |
| 201774 |  46263 | 1995-12-17 | 1996-12-16 |
| 201774 |  46859 | 1996-12-16 | 1997-03-04 |
+--------+--------+------------+------------+
3 rows in set (0.00 sec)
```

Figure 01

Following activity which would update those records from the salaries table which have emp_no=201774 and then commit the changes in the database.

Output : table that have emp_no=201774 would be updated and select statement would produce the result.

```
mysql> select * from salaries
    -> where emp_no=201774;
+--------+--------+------------+------------+
| emp_no | salary | from_date  | to_date    |
+--------+--------+------------+------------+
| 201774 |  40000 | 1994-12-17 | 1995-12-17 |
| 201774 |  42057 | 1995-12-17 | 1996-12-16 |
| 201774 |  42599 | 1996-12-16 | 1997-03-04 |
+--------+--------+------------+------------+
3 rows in set (0.00 sec)

mysql> SET AUTOCOMMIT = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE SALARIES SET SALARY = SALARY*(1.1) WHERE EMP_NO=201774;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Figure 02

Concurrent Accesses

In company database select departments table.

```
ERROR 1146 (42S02): Table 'company.departmants' doesn't exist
mysql> SELECT * FROM DEPARTMENTS;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.00 sec)

mysql>
```

```
mysql> USE COMPANY;
Database changed
mysql> SELECT * FROM DEPARTMENTS;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.01 sec)

mysql>
```

Figure 03

Step 01 : Start a new transaction in second window and insert a new value in the department table d010.

Check the insert value in both window

Output: second window shows the inserted value and first window would not impact.

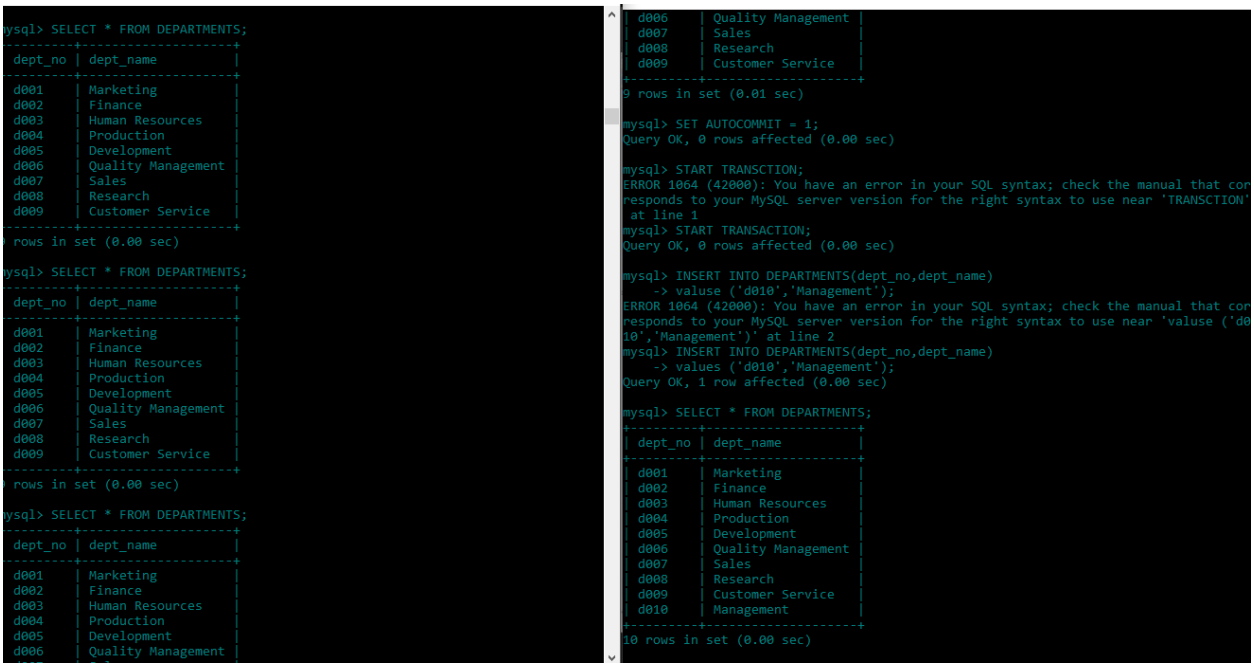

Figure 04

Step 02: commit the second window and check the Output

Output: In first window also impact, commit save all the transactions to the database.

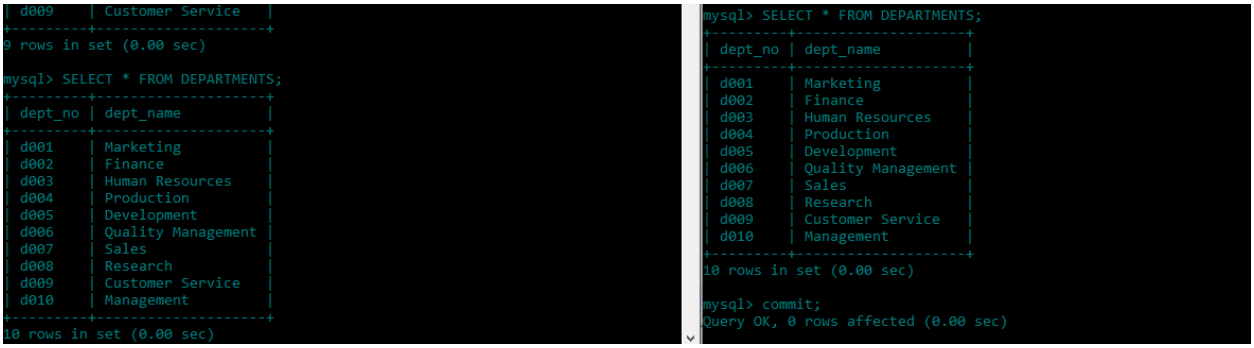

Figure 05

d010 insert to the table after commit the transaction in second window

Step 03: Start a new transaction in first window, Update the departments table using first window check the update saved in the table using first window.

Check the update using both windows before commit.

Output: first window updated and second window not update

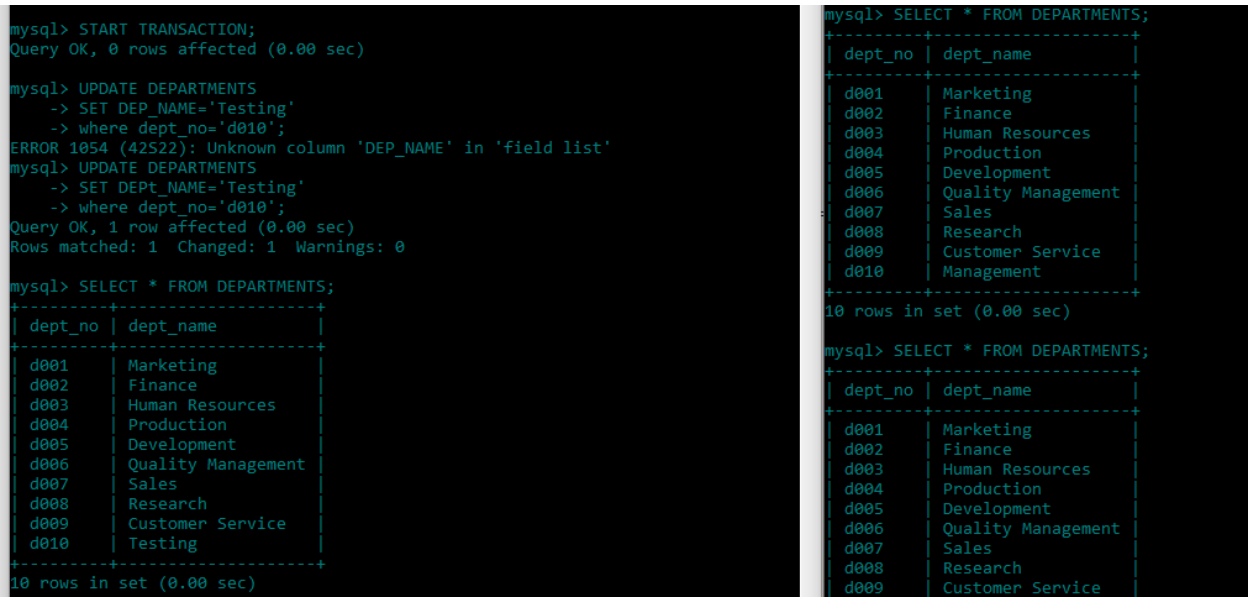

Figure 06

D010 in departments table in first window impact the results and second window would not impact

Step 04: commit the first window and check the update

Output : both windows updated

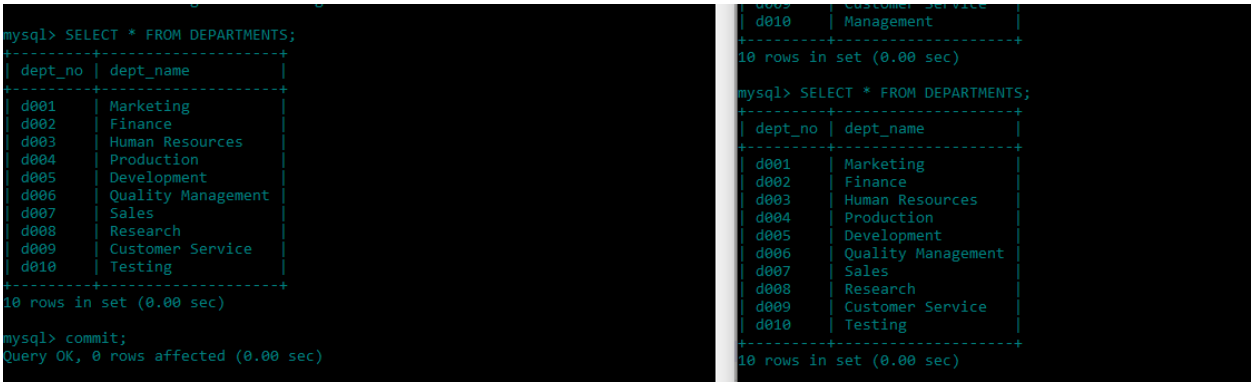

Figure 07

Use your imagination and words to write a scenario where using transactions is essential and then create the required tables and test how the transaction will effect your tables,

1. during the transaction execution.

2. after rollback statement.

3. after the commit statement.

4. during 2 concurrent transactions, both of them update a record and both of them commit it.

- *Create a sample database and create a table Persons.*

```
mysql> use sample;
Database changed
mysql> CREATE TABLE Persons (
    ->     PersonID int,
    ->     LastName varchar(255),
    ->     FirstName varchar(255),
    ->     Address varchar(255),
    ->     City varchar(255)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> insert into Persons(PersonID,lastname,firstname,address,city)
    -> values (0001,'Sathakaran','Thinesh','Akber','kandy');
Query OK, 1 row affected (0.00 sec)

mysql> select * from Persons;
+----------+------------+-----------+---------+-------+
| PersonID | LastName   | FirstName | Address | City  |
+----------+------------+-----------+---------+-------+
|        1 | Sathakaran | Thinesh   | Akber   | kandy |
+----------+------------+-----------+---------+-------+
1 row in set (0.00 sec)

mysql> insert into Persons(PersonID,lastname,firstname,address,city)
    -> values (0002,'Wilman','Tom','Colombo','kandy');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Persons(PersonID,lastname,firstname,address,city)
    -> values (0003,'Cardinal','Zbyszek','Colombo','Colombo');
Query OK, 1 row affected (0.00 sec)

mysql> select * from Persons;
+----------+------------+-----------+---------+---------+
| PersonID | LastName   | FirstName | Address | City    |
+----------+------------+-----------+---------+---------+
|        1 | Sathakaran | Thinesh   | Akber   | kandy   |
|        2 | Wilman     | Tom       | Colombo | kandy   |
|        3 | Cardinal   | Zbyszek   | Colombo | Colombo |
+----------+------------+-----------+---------+---------+
3 rows in set (0.00 sec)
```

Figure 08

1,2

start transaction can be used to initiate the transaction.

Following sinario which would delete those records from Persons table which have PersonID=1 and rollback the changes in the sample database.

Output: Delete operation would not impact the table and the select statement would produce the result.

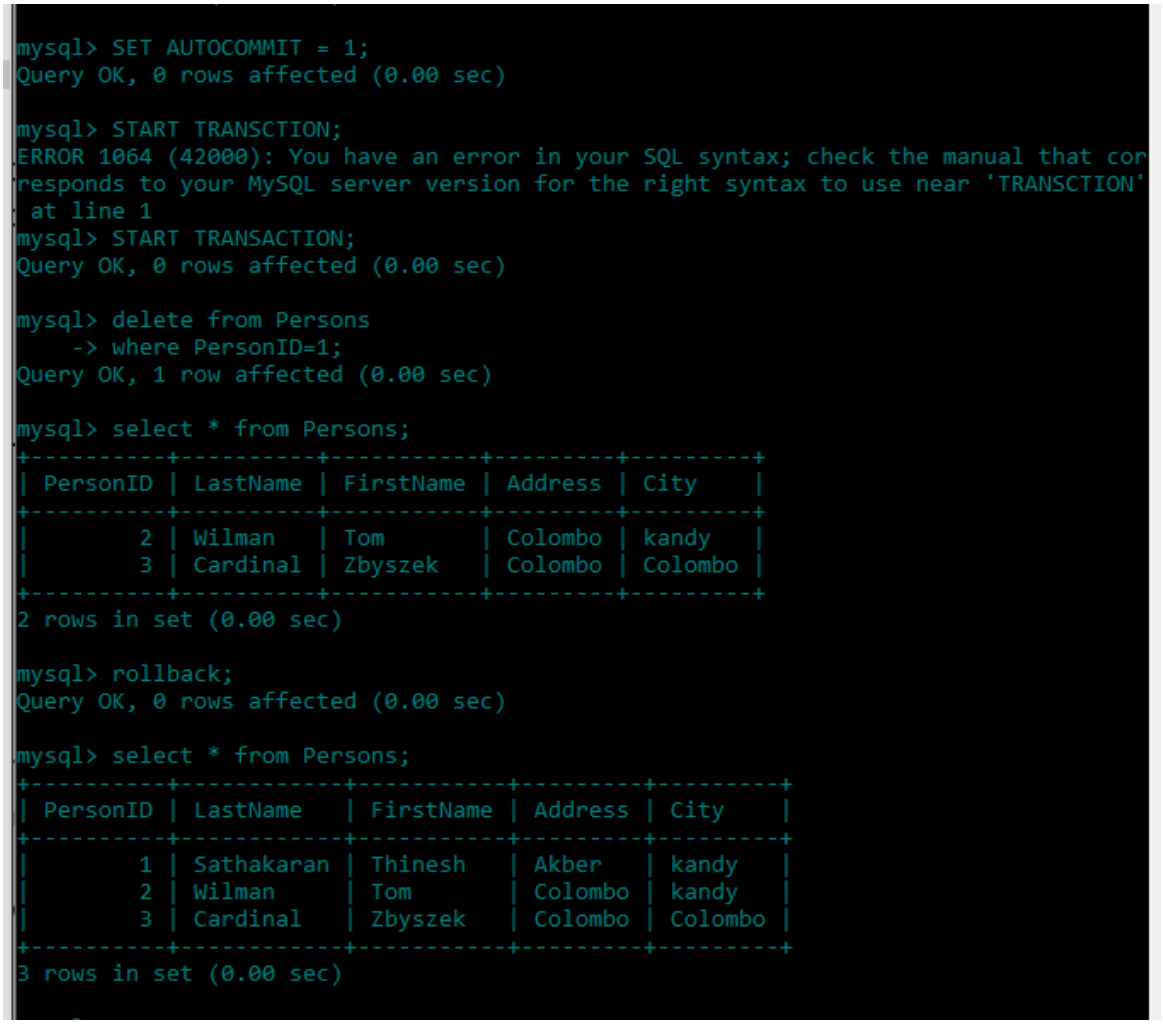

Figure 09

3. following sinario which would delete those records from the Persons table which have PersonID=1 and then commit the changes in the sample database.

Output : table that have PersonID=1 would be deleted and select statement would produce the result.

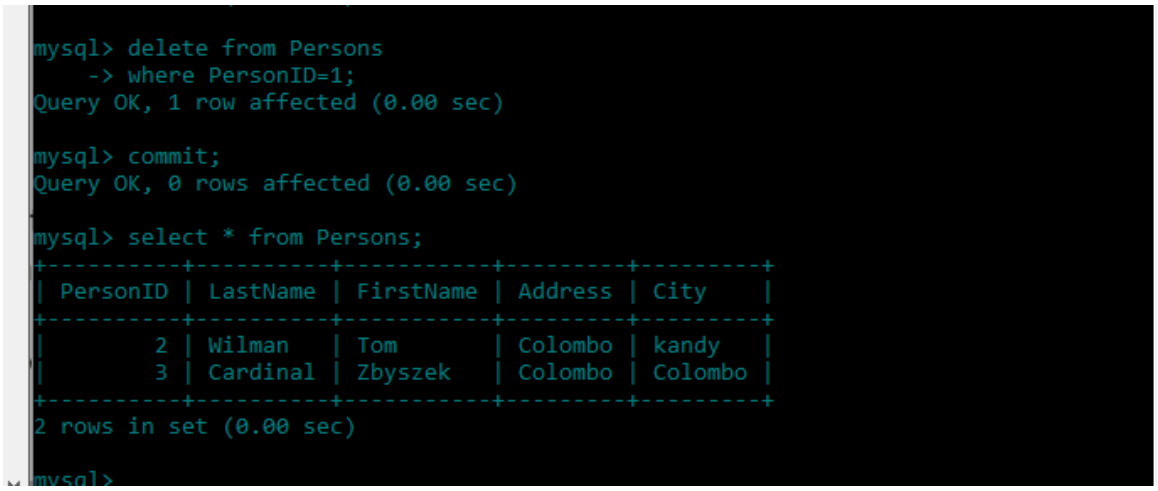

Figure 10

After commit the transaction do the rollback , that delete operation impact the table , select statement would produce the results.

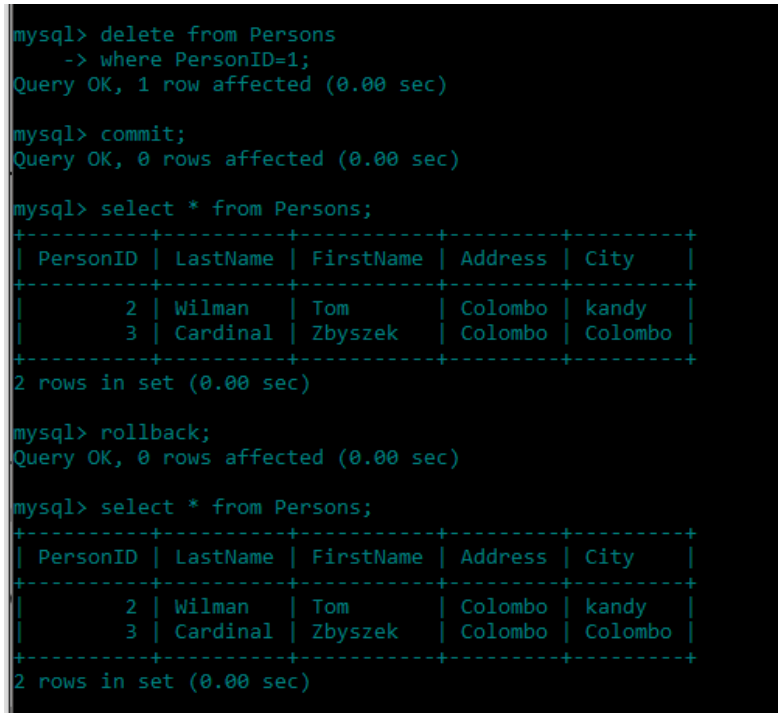

Figure 11

Connect the same database using two command line windows. After commit the transaction that would impact the second command line.

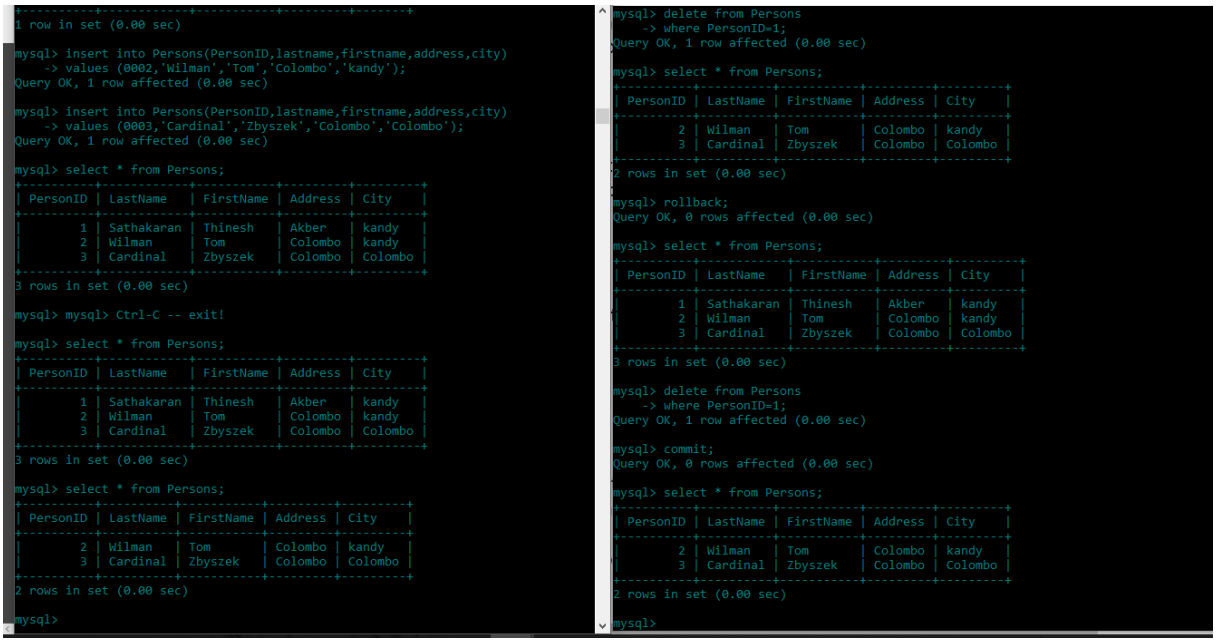

Figure 12

4. start a new transaction , update the persons table from second window and check the updated results from two terminals .

Output : second window shown the updated results

First window not changed

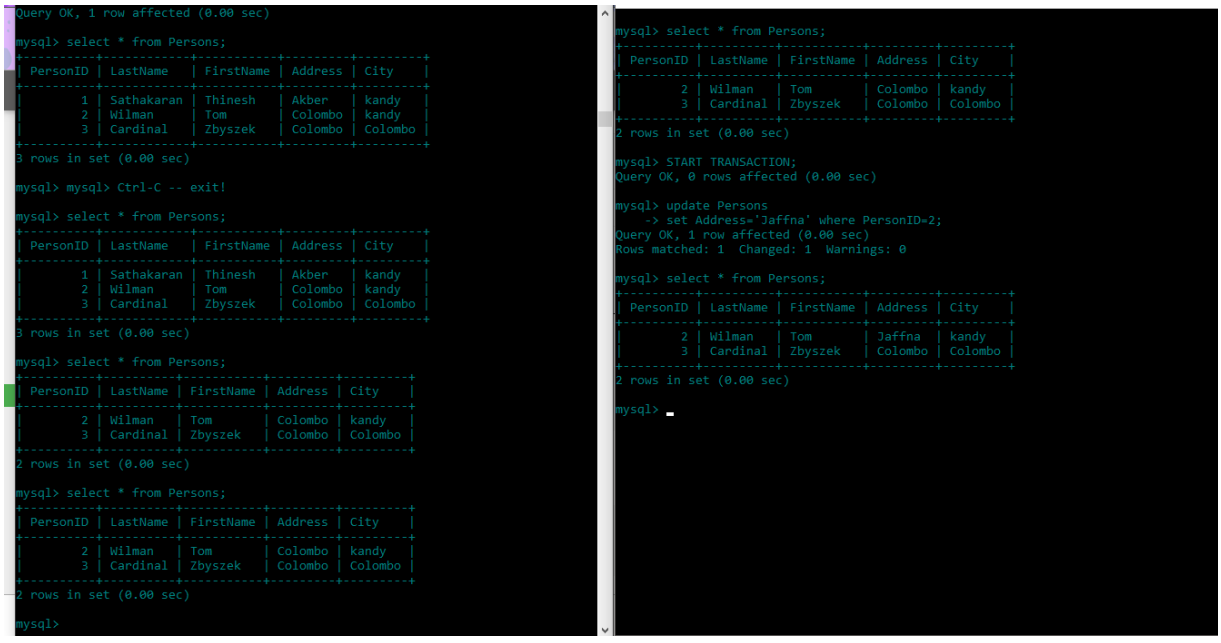

Figure 13

After update the person table from second window and the same time try to update the table from first window before commit the update in second window

Output : I gives the error in first window . before commit the second window transaction

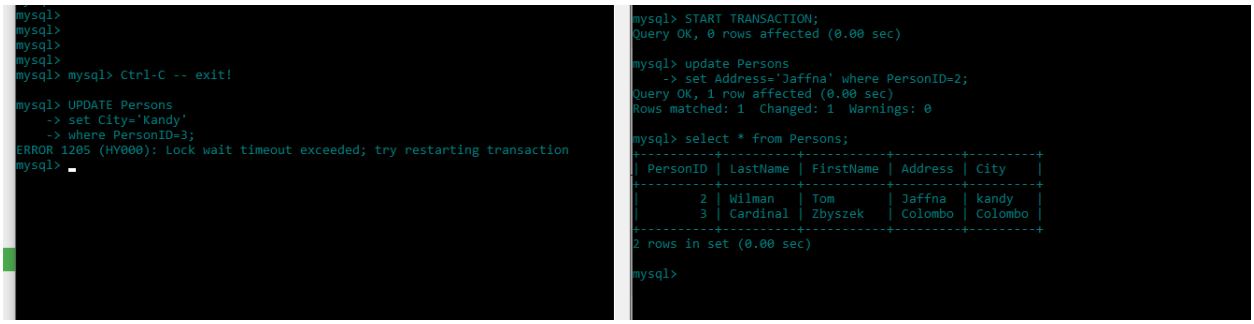

Figure 14

After commit the second transaction, update the first transaction. That impact the database and save the changes.

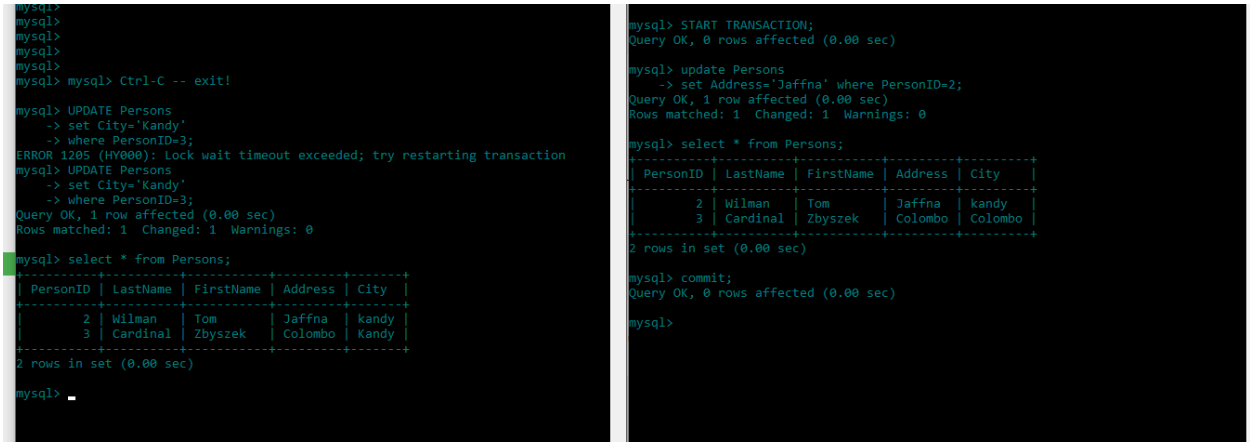

figure 15