P9: Transaction Control Language

[1]. COMMIT [2]. ROLLBACK [3]. SAVEPOINT [4]. SET TRANSACTION

OUTPUT

Begin the transaction

After executing
this command, the transaction state has been initiated as begin

```
mysql> insert into menu values(6007,'Kunafa',200,'Arab','Veg');
Query OK, 1 row affected (0.01 sec)

mysql> select * from menu;
+---------+-----------------------------------+-------+-----------------+----------+
| Menu_id | Name                              | Price | Type            | Category |
+---------+-----------------------------------+-------+-----------------+----------+
|    6001 | Spaghetti with meatballs          |   250 | Italian-American| Non veg  |
|    6002 | Mutton Biryani                    |   450 | Hyderabadi      | Non veg  |
|    6003 | Veg Biryani                       |   300 | Hyderabadi      | Veg      |
|    6004 | Chicken Biryani                   |   350 | Hyderabadi      | Non veg  |
|    6005 | Sheet Pan Sausage Beef with Broccoli | 500 | Italian      | Non veg  |
|    6006 | Grilled halloumi                  |   400 | Arab            | Non veg  |
|    6007 | Kunafa                            |   200 | Arab            | Veg      |
+---------+-----------------------------------+-------+-----------------+----------+
7 rows in set (0.00 sec)

mysql> SAVEPOINT S2;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from menu where Menu_id=6006;
Query OK, 1 row affected (0.02 sec)

mysql> select * from menu;
+---------+-----------------------------------+-------+-----------------+----------+
| Menu_id | Name                              | Price | Type            | Category |
+---------+-----------------------------------+-------+-----------------+----------+
|    6001 | Spaghetti with meatballs          |   250 | Italian-American| Non veg  |
|    6002 | Mutton Biryani                    |   450 | Hyderabadi      | Non veg  |
|    6003 | Veg Biryani                       |   300 | Hyderabadi      | Veg      |
|    6004 | Chicken Biryani                   |   350 | Hyderabadi      | Non veg  |
|    6005 | Sheet Pan Sausage Beef with Broccoli | 500 | Italian      | Non veg  |
|    6007 | Kunafa                            |   200 | Arab            | Veg      |
+---------+-----------------------------------+-------+-----------------+----------+
6 rows in set (0.04 sec)

mysql> ROLLBACK WORK TO SAVEPOINT S2;
Query OK, 0 rows affected (0.01 sec)
```



```
mysql> ROLLBACK WORK TO SAVEPOINT S2;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from menu;
+---------+-----------------------------------+-------+-----------------+----------+
| Menu_id | Name                              | Price | Type            | Category |
+---------+-----------------------------------+-------+-----------------+----------+
|    6001 | Spaghetti with meatballs          |   250 | Italian-American| Non veg  |
|    6002 | Mutton Biryani                    |   450 | Hyderabadi      | Non veg  |
|    6003 | Veg Biryani                       |   300 | Hyderabadi      | Veg      |
|    6004 | Chicken Biryani                   |   350 | Hyderabadi      | Non veg  |
|    6005 | Sheet Pan Sausage Beef with Broccoli | 500 | Italian      | Non veg  |
|    6006 | Grilled halloumi                  |   400 | Arab            | Non veg  |
|    6007 | Kunafa                            |   200 | Arab            | Veg      |
+---------+-----------------------------------+-------+-----------------+----------+
7 rows in set (0.00 sec)

mysql>
```

After rolling back, we will able to retrieve the data that we have created in backup location before the save point location. It means that whatever queries we have executed before the save point will be created. Using the rollback, we have aborted the state of transaction.


```
mysql> COMMIT;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Here I have applied the **commit** command where it will be used to terminate the state of transaction that I have created in the previous stage. So if I again attempts to rollback to previous save point location, then it throw an error, that this save point doesn't exist. So if we want to create the new transaction, we need to repeat the step that we have done in the previous stage

```
mysql> COMMIT;
Query OK, 0 rows affected (0.02 sec)

mysql> ROLLBACK WORK TO SAVEPOINT S1;
ERROR 1305 (42000): SAVEPOINT S1 does not exist
mysql> ROLLBACK WORK TO SAVEPOINT S2;
ERROR 1305 (42000): SAVEPOINT S2 does not exist
mysql>
```

```
mysql> select * from menu;
+---------+-----------------------------------+-------+-----------------+----------+
| Menu_id | Name                              | Price | Type            | Category |
+---------+-----------------------------------+-------+-----------------+----------+
|    6001 | Spaghetti with meatballs          |   250 | Italian-American | Non veg  |
|    6002 | Mutton Biryani                    |   450 | Hyderabadi      | Non veg  |
|    6003 | Veg Biryani                       |   300 | Hyderabadi      | Veg      |
|    6004 | Chicken Biryani                   |   350 | Hyderabadi      | Non veg  |
|    6005 | Sheet Pan Sausage Beef with Broccoli |  500 | Italian      | Non veg  |
|    6006 | Grilled halloumi                  |   400 | Arab            | Non veg  |
|    6007 | Kunafa                            |   200 | Arab            | Veg      |
+---------+-----------------------------------+-------+-----------------+----------+
7 rows in set (0.01 sec)

mysql> insert into menu values(6008,"Falafel",60,"Arab","Veg");
Query OK, 1 row affected (0.01 sec)

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

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

mysql> select * from menu;
+---------+-----------------------------------+-------+-----------------+----------+
| Menu_id | Name                              | Price | Type            | Category |
+---------+-----------------------------------+-------+-----------------+----------+
|    6001 | Spaghetti with meatballs          |   250 | Italian-American | Non veg  |
|    6002 | Mutton Biryani                    |   450 | Hyderabadi      | Non veg  |
|    6003 | Veg Biryani                       |   300 | Hyderabadi      | Veg      |
|    6004 | Chicken Biryani                   |   350 | Hyderabadi      | Non veg  |
|    6005 | Sheet Pan Sausage Beef with Broccoli |  500 | Italian      | Non veg  |
|    6006 | Grilled halloumi                  |   400 | Arab            | Non veg  |
|    6007 | Kunafa                            |   200 | Arab            | Veg      |
|    6008 | Falafel                           |    60 | Arab            | Veg      |
+---------+-----------------------------------+-------+-----------------+----------+
8 rows in set (0.00 sec)

mysql>
```

After inserting the record with then after rollback it will be undoing the changes to previous commit work.

```
mysql> select * from menu;
+---------+---------------------------------------+-------+-----------------+---------+
| Menu_id | Name                                  | Price | Type            | Category |
+---------+---------------------------------------+-------+-----------------+---------+
|    6001 | Spaghetti with meatballs              |   250 | Italian-American | Non veg |
|    6002 | Mutton Biryani                        |   450 | Hyderabadi      | Non veg |
|    6003 | Veg Biryani                           |   300 | Hyderabadi      | Veg     |
|    6004 | Chicken Biryani                       |   350 | Hyderabadi      | Non veg |
|    6005 | Sheet Pan Sausage Beef with Broccoli  |   500 | Italian         | Non veg |
|    6006 | Grilled halloumi                      |   400 | Arab            | Non veg |
|    6007 | Kunafa                                |   200 | Arab            | Veg     |
|    6008 | Falafel                               |    60 | Arab            | Veg     |
+---------+---------------------------------------+-------+-----------------+---------+
8 rows in set (0.00 sec)

mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.01 sec)

mysql> insert into menu values(6009,'Manakeesh',400,'Arab','Non veg');
Query OK, 1 row affected (0.01 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.49 sec)

mysql> select * from menu;
+---------+---------------------------------------+-------+-----------------+---------+
| Menu_id | Name                                  | Price | Type            | Category |
+---------+---------------------------------------+-------+-----------------+---------+
|    6001 | Spaghetti with meatballs              |   250 | Italian-American | Non veg |
|    6002 | Mutton Biryani                        |   450 | Hyderabadi      | Non veg |
|    6003 | Veg Biryani                           |   300 | Hyderabadi      | Veg     |
|    6004 | Chicken Biryani                       |   350 | Hyderabadi      | Non veg |
|    6005 | Sheet Pan Sausage Beef with Broccoli  |   500 | Italian         | Non veg |
|    6006 | Grilled halloumi                      |   400 | Arab            | Non veg |
|    6007 | Kunafa                                |   200 | Arab            | Veg     |
|    6008 | Falafel                               |    60 | Arab            | Veg     |
+---------+---------------------------------------+-------+-----------------+---------+
8 rows in set (0.00 sec)
```

In the above snap, since we are using rollback without mentioning the commit command, then it will be used for the undoing the query because we haven't committed the state of work

```
mysql> select * from customer;
+-------------+----------+---------+----------------+----------------------+
| Customer_id | Fname    | Lname   | Contact        | Email_id             |
+-------------+----------+---------+----------------+----------------------+
|           1 | ambilyfra| pk      | 987654321      | NULL                 |
|        2001 | Jolly    | Francis | +918605747571  | jolly98@gmail.com    |
|        2002 | C J      | Francis | +918635747571  | francis@gmail.com    |
|        2003 | Sachin   | M       | +918335747571  | sachi@gmail.com      |
|        2004 | Ambily   | Fra     | +918335747311  | amfra@gmail.com      |
|        2005 | Arpitha  | Reddy   | +918765747311  | arpi@gmail.com       |
|        2008 | appu     | cc      | +91865423123   | app@gmail.com        |
+-------------+----------+---------+----------------+----------------------+
7 rows in set (0.01 sec)

mysql> insert into customer values(2009,'John','C','+018542165462','jhonnykuttan@gmail.com');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(2010,'SRK','Khan','+918942165462','srk@gmail.com');
Query OK, 1 row affected (0.01 sec)

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

mysql> select * from customer;
+-------------+----------+---------+----------------+----------------------+
| Customer_id | Fname    | Lname   | Contact        | Email_id             |
+-------------+----------+---------+----------------+----------------------+
|           1 | ambilyfra| pk      | 987654321      | NULL                 |
|        2001 | Jolly    | Francis | +918605747571  | jolly98@gmail.com    |
|        2002 | C J      | Francis | +918635747571  | francis@gmail.com    |
|        2003 | Sachin   | M       | +918335747571  | sachi@gmail.com      |
|        2004 | Ambily   | Fra     | +918335747311  | amfra@gmail.com      |
|        2005 | Arpitha  | Reddy   | +918765747311  | arpi@gmail.com       |
|        2008 | appu     | cc      | +91865423123   | app@gmail.com        |
|        2009 | John     | C       | +018542165462  | jhonnykuttan@gmail.com |
|        2010 | SRK      | Khan    | +918942165462  | srk@gmail.com        |
+-------------+----------+---------+----------------+----------------------+
9 rows in set (0.00 sec)

mysql>
```

Effect of auto commit =0 with rollback action

Enabling auto commit and committing the transaction will insert the record in the table on the contrary, if auto commit is disabled and when the data is inserted, it will rollback or undo the changes to the previous committed state. Which means previous values which are inserted to customer table (Vinod and Vinaya's details) will only be in committed state

Using read only command

Here if we invoke the set the transaction with read only privileges, then we can have the provision to view the data contents in the table. When I attempt to insert the data in the table, you can see the violation of read only is supported where we can use the select command to view the data. Other DML commands like insert, update, delete will not work.

Using read write command

```
Command Prompt - mysql -u root -p                    —    □    ×

mysql> insert into customer values(2012,'vinaya','M','+918972165482','vinaya ^
d@gmail.com');
ERROR 1792 (25006): Cannot execute statement in a READ ONLY transaction.
mysql> set transaction read write;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into customer values(2012,'vinaya','M','+918972165482','vinaya
d@gmail.com');
Query OK, 1 row affected (0.01 sec)

mysql> commit;
Query OK, 0 rows affected (0.01 sec)

mysql> _
```
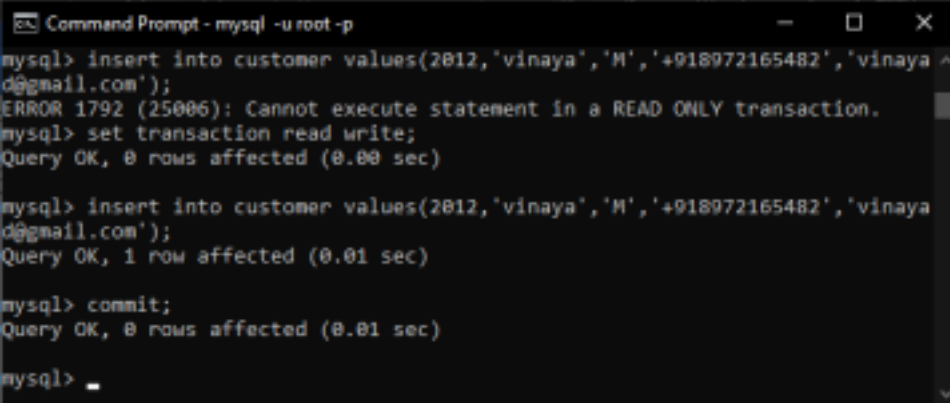
Using the **read write** as you view clearly in the above snap, we need to make use of command known to be set transaction with read and write where after setting the transaction state to read and write, we can go for insertion of data and then I saved the transaction by terminating it using commit command.