CHAPTER 10

ADVANCED RDBMS CONCEPTS

Learning Objectives

After studying this lesson the students will be able to:

- Define a Transaction
- Describe reason why all the tasks in a transaction should be executed fully or not at all.
- Perform basic transactions.
- Commit a transaction.
- Add Save Points to a transaction.
- Roll back a Transaction
- Roll back a Transaction to a Savepoint.

Till now we have studied about various SQL statements manipulating data stored in a MySQL database. We executed SQL statements without concern about inconsistencies arising due to group of statements not being executed in entirety. In this lesson, we will study the basic concepts of Transaction processing and how MySQL ensures consistency of data when a group of statements is executed.

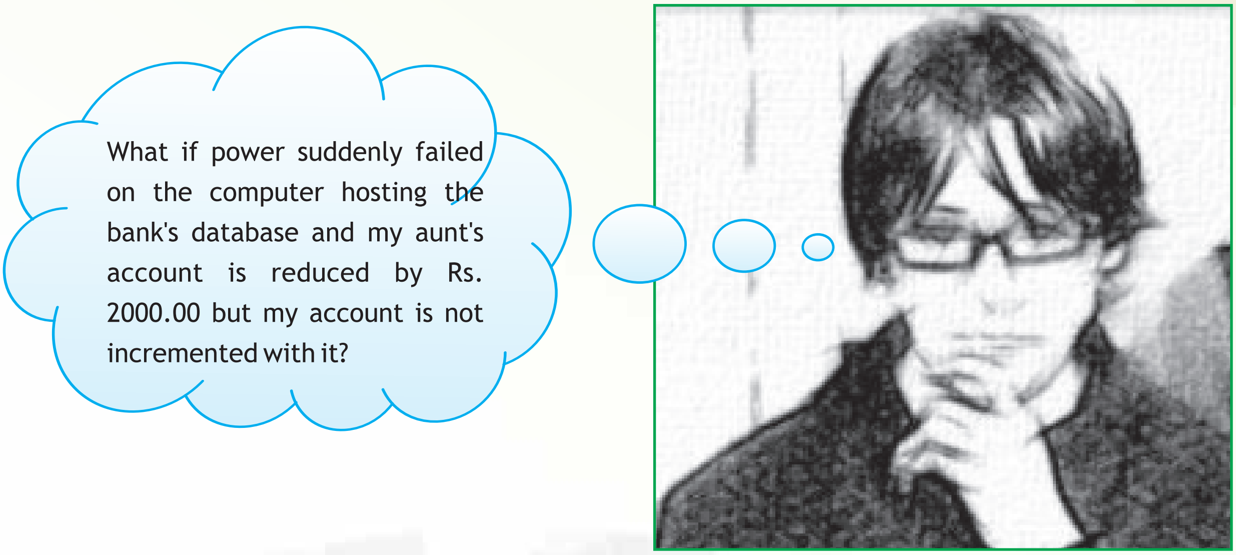
Puzzle[9]

Vijaya has to withdraw ₹ 2500.00 from her account in the bank. She asked for m notes of ₹ 50.00 and n notes of ₹ 100.00. The cashier made a mistake and handed her m notes of ₹ 100.00 and n notes of ₹ 50.00. When she returned back home she realized that she got ₹ 500.00 less. How many notes of ₹ 50.00 and ₹ 100.00 did she ask for?

Introduction

Raunak studies in Class XII. He is very helpful. During summer vacations, he helped his aunt's son in his studies. His aunt was very pleased with him and gave him a cheque of ₹ 2000.00. Raunak knows that after presentation of the cheque to the bank, his aunt's account will be reduced by ₹ 2000.00 and his account will be increased by ₹ 2000.00. Raunak walked up to the bank to present the cheque.

While returning from bank, Raunak is apprehensive about one thing:


What if power suddenly failed on the computer hosting the bank's database and my aunt's account is reduced by Rs. 2000.00 but my account is not incremented with it?

Are Raunak's fears valid? Can you recollect something like this happening with you or with your friends or relatives?

Raunak should not fear at all as the Bank's DBMS looks after these eventualities. DBMSs ensure consistency (correctness) of data by managing Transactions.

DBMS and Transaction Management

Mostly customers view an operation like transfer of funds as a single operation but actually it consists of series of operations.

Suppose Raunak's account number is 3246 and his aunt's account number is 5135. In order to process the cheque presented by Raunak, the following two SQL commands need to be executed on the database maintained by the bank:

```
UPDATE Savings
SET balance = balance - 2000  ←  For Aunt's account
WHERE account_no = 5135;

UPDATE Savings
SET balance = balance + 2000  ←  For Raunak's account
WHERE account_no = 3246;
```

The above two Updates should both take place. If the first Update takes place and there is a system failure, the first updation should be undone. Either both the updations should be done and if it is not possible for both the updations to be done, then no updation should be done.

What is a Transaction?

A Transaction is a unit of work that must be done in logical order and successfully as a group or not done at all. Unit of work means that a Transaction consists of different tasks - but together they are considered as one unit. Each transaction has a beginning and an end. If anything goes wrong in between the execution of transaction, the entire transaction (No matter to what extent has been done) should be cancelled. If it is successful, then the entire transaction should be saved to the database.

> A transaction is a unit of work that must be done in logical order and successfully as a group or not done at all.

In Raunak's case, both the updation statements constitute a transaction. Both are together treated as a single unit.

To understand how transactions are managed, let us study the following 3 statements of SQL:

- START TRANSACTION statement
- COMMIT statement
- ROLLBACK statement

START TRANSACTION Statement :

START TRANSACTION statement commits the current transaction and starts a new transaction. It tells MySQL that the transaction is beginning and the statements that follow should be treated as a unit, until the transaction ends. It is written like this:

```
START TRANSACTION;
```

The **START TRANSACTION** statement has no clauses.

COMMIT Statement :

The COMMIT statement is used to save all changes made to the database during the transaction to the database.Commit statement is issued at a time when the transaction is complete- all the changes have been successful and the changes should be saved to the database. COMMIT ends the current transaction.

COMMIT statement is used like this:

```
COMMIT;
```

Or

```
COMMIT WORK;
```

Here WORK is a keyword and is optional.

In the following example, the table named savings has 2 rows. A transaction is started and balance in Siddharth's account (with account number 1004) is increased by Rs. 2000.00 and the balance in Akriti's account (with account number 1006) is decreased by Rs. 2000.00. COMMIT statement makes the changes made by the transaction permanent.

Example 1:

```
mysql> select * from savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 87000.00 |
|       1006 | Akriti Malik     | 87000.00 |
+------------+------------------+----------+
mysql> START TRANSACTION;
mysql> UPDATE Savings
    -> SET balance = balance + 2000
    -> WHERE account_no = 1004;
mysql> UPDATE Savings
    -> SET balance = balance - 2000
    -> WHERE account_no = 1006;
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 89000.00 |
|       1006 | Akriti Malik     | 85000.00 |
+------------+------------------+----------+
2 rows in set (0.00 sec)
mysql> COMMIT;
```

ROLLBACK Statement :

When a transaction is being executed, some type of error checking is usually performed to check whether it is executing successfully or not. If not, the entire transaction is undone using the ROLLBACK statement. The ROLLBACK statement cancels the entire transaction i.e. It rolls the transaction to the beginning. It aborts any changes made during the transaction and the state of database is returned to what it was before the transaction began to execute and does not save any of the changes made to the database during the transaction.

ROLLBACK statement is used like this:

```
ROLLBACK;
```

Or

```
ROLLBACK WORK;
```

Here WORK is a keyword and is optional.

If in Example 1 shown above ROLLBACK was used instead of COMMIT, the updation of incrementing Siddharth's account by ₹ 2000.00 and decrementing Akriti's account by ₹ 2000 wouldn't have taken place. Let us now initiate a transaction, increase Akriti's account by ₹ 3000.00, then Rollback the transaction and see what happens to the updation done on Akriti's account.

```
mysql> SELECT * FROM Savings;
+-----------+-----------------+---------+
| account_no | name            | balance  |
+-----------+-----------------+---------+
|       1004 | Siddharth Sehgal | 89000.00 |
|       1006 | Akriti Malik     | 85000.00 |
+-----------+-----------------+---------+
mysql> START TRANSACTION;
mysql> UPDATE Savings
```

Before the transaction starts, Siddharth's balance is Rs. 89000 and Akriti's balance is Rs. 85000.00

```
    -> SET balance = balance + 3000
    -> WHERE account_no = 1006;
mysql> ROLLBACK;
mysql> SELECT * FROM Savings;
+------------+------------------+----------+
| account_no | name             | balance  |
+------------+------------------+----------+
|       1004 | Siddharth Sehgal | 89000.00 |
|       1006 | Akriti Malik     | 85000.00 |
+------------+------------------+----------+
```

Akriti's balance is increased by Rs. 3000.00

Because of the Rollback, Akriti's balance is not updated and is displayed as it was before the transaction started.

- After the ROLLBACK command is issued to the database, the database itself starts a new transaction; though no explicit command of starting a transaction like START TRANSACTION is issued.

Example 2:

Let us try out some more SQL statements on Savings table to understand transactions well.

```
mysql> SELECT * FROM savings;
+------------+------------------+----------+
| account_no  | name             | balance   |
+------------+------------------+----------+
|       1004  | Siddharth Sehgal | 84000.00  |
|       1006  | Akriti Malik     | 92000.00  |
|       1008  | Chavi Mehra      | 67000.00  |
|       1009  | Raunak Singh     | 56000.00  |
+------------+------------------+----------+
```

```
mysql> INSERT INTO Savings VALUES
    (1010,'Lakshmi Swamy',34000);

mysql> START TRANSACTION;

mysql> UPDATE Savings SET balance =
balance +2000 WHERE account_no = 1010;

mysql> ROLLBACK;

mysql> SELECT * FROM Savings;

+-----------+-----------------+----------+
| account_no | name           | balance  |
+-----------+-----------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
|       1009 | Raunak Singh     | 56000.00 |
|       1010 | Lakshmi Swamy    | 34000.00 |
+-----------+-----------------+----------+

5 rows in set (0.00 sec)
```

Start transaction statement starts a transaction and commits the previous INSERT INTO statement.

Rollback cancels the effect of Update statement.

SELECT statement displays Lakshmi Swamy's row with balance of 34000.00

Inserting SavePoints :

The SAVEPOINT statement defines a marker in a transaction. These markers are useful in rolling back a transaction till the marker.

We can add a savepoint anywhere in a transaction. When you roll back to that savepoint, any changes made to the database after the savepoint are discarded, and any changes made prior to the savepoint are saved. It is like semicomitting a transaction.

To define a savepoint, we enter the SAVEPOINT statement like this:

```
SAVEPOINT <savepoint-name>;
```

Example :

```
SAVEPOINT Mark1;
```

In the above statement a marker (savepoint) with the name Mark1 is defined. It becomes a bookmark in the transaction. Now we can write the following statement:

```
ROLLBACK TO SAVEPOINT Mark1;
```

to rollback the transaction till the bookmark named Mark1.

Setting Autocommit :

By default, Autocommit mode is on in MySQL. It means that MySQL does a COMMIT after every SQL statement that does not return an error. If it returns an error then either Rollback or Commit happens depending on the type of error. If we do not want individual statements of SQL to be automatically committed, we should set the autocommit mode to off.

When Autocommit is off then we have to issue COMMIT statement explicitly to save changes made to the database.

The following statement sets the autocommit mode to off. It also starts a new transaction

```
SET AUTOCOMMIT=0;
```

The following statement sets the autocommit mode to ON. It also commits and terminates the current transaction.

```
SET AUTOCOMMIT=1;
```

If autocommit is set to ON. we can still perform a multiple-statement transaction by starting it with an explicit START TRANSACTION statement and ending it with COMMIT or ROLLBACK.

Let us look at the following example to understand it:

Example

```
mysql> SET AUTOCOMMIT = 0;
```

Autocommit is disabled.

```
mysql> SELECT * FROM Savings;

+------------+-----------------+---------+
| account_no | name            | balance |
+------------+-----------------+---------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
+------------+-----------------+---------+
```

Table Savings has 3 rows.

```
mysql> INSERT INTO Savings values
(1009,'Raunak Singh', 56000);
```

Another row for Raunak Singh added.

```
mysql> ROLLBACK;
```

Insert statement was not committed so it is undone by Rollback

```
mysql> SELECT * FROM Savings;

+------------+-----------------+---------+
| account_no | name            | balance |
+------------+-----------------+---------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik     | 92000.00 |
|       1008 | Chavi Mehra      | 67000.00 |
+------------+-----------------+---------+
```

Table does not show Raunak Singh's row.

```
mysql> SET AUTOCOMMIT = 1;
```

Autocommit is enabled.

```
mysql> INSERT INTO Savings VALUES
(1009,'Raunak Singh', 56000);
```

Raunak's row is added and is committed too.

```
mysql> ROLLBACK;

mysql> SELECT * FROM Savings;

+------------+-----------------+----------+
| account_no | name            | balance  |
+------------+-----------------+----------+
|       1004 | Siddharth Sehgal | 84000.00 |
|       1006 | Akriti Malik    | 92000.00 |
|       1008 | Chavi Mehra     | 67000.00 |
|       1009 | Raunak Singh    | 56000.00 |
+------------+-----------------+----------+
```

Rollback cannot undo insertion of Raunak's row.

If the autocommit mode has been set to off in a session and you end that session, the autocommit mode is automatically set to on when you start a new session.

Let us try out some more SQL statements :

Example

```
mysql> SET AUTOCOMMIT = 1;

Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM Savings WHERE account_no = 1006;

Query OK, 1 row affected (0.00 sec)

mysql> ROLLBACK WORK;

Query OK, 0 rows affected (0.03 sec)
```

Autocommit is enabled

Start transaction sets autocommit off.

Row with account_no 1006 deleted but is not committed.

Deletion of Row with account_no 1006 is cancelled.

An implicit COMMIT takes place, even if AUTOCOMMIT is set OFF, on the database when the user issues a Data Definition language command like CREATE TABLE, ALTER TABLE etc .

Future Trends

Cloud Computing

A major shift is predicted in the IT industry in the way that software and services are going to be delivered in future. In this regard, cloud computing is an emerging area these days. It is the computing model where the infrastructure and the applications are offered as a service over the Internet. Cloud computing takes place out on someone else's network. Since the details of how it is set up or how it works is hidden from the user, the term cloud is used. Cloud means a large network that is away and is not in our control. Just about any database can be run in a cloud-based infrastructure. Search the web and find out how RDBMSs like MySQL will be significant in such a computing model.

Summary

- Work done during a transaction is a series of operations.
- If one of the operations of a transaction is not executed successfully, then the entire transaction should be cancelled. If all the operations are executed successfully, the transaction should be saved to a database.
- START TRANSACTION statement is used to start a transaction.
- The process of cancelling a transaction is called Rolling back.
- ROLLBACK statement is used to terminate a transaction and roll back the database to its original state before the transaction.
- COMMIT statement is used to save changes to the database.
- When AutoCommit is ON, each SQL statement is a transaction. The changes resulting from each statement are automatically committed.

- When Auto Commit is Off then changes made to database are not committed unless explicitly requested.

EXERCISES

MULTIPLE CHOICE QUESTIONS

1. A ______________________ is a logical unit of work that must succeed or fail in its entirety.

   a) Primary key

   b) Database

   c) Transaction

   d) none of these

2. When AutoCommit is ____________ ,changes made to database are not committed unless explicitly requested.

   a) Equal to "-"

   b) on

   c) off

   d) Equal to "%"

3. When a CREATE TABLE command is issued, a _____________ occurs on the database.

   a) ROLLBACK

   b) COMMIT

   c) SAVEPOINT

   d) ROLLBACK TO SAVEPOINT

4. By default AUTOCOMMIT is ______________.

   a) disabled

   b) enabled

c) inactive

d) none of the above

5. Which of the following statement or command? Completes a transaction?

a) INSERT INTO

b) COMMIT

c) DELETE

d) SELECT

6. If Feroze deposits a cheque of Rs. 1200.00 in his account, which was given to him by Ali, two tasks: decreasing of Rs. 1200.00 from Ali's account and increment of Rs. 1200.00 in Feroze's account are done. ________________ constitute(s) a transaction.

a) First task

b) Both the tasks

c) None of the tasks

d) Second task.

7. START TRANSACTION statement

a) Updates the current database.

b) Rolls back the current transaction.

c) Commits the current transaction and starts a new transaction

d) Starts a new transaction only.

8. ROLLBACK statement

a) Cancels the entire transaction.

b) Deletes the rows added to the table.

c) Commits the current transaction and starts a new transaction

d) Rolls back all the insertions of rows made during the current transaction.

ANSWER THE FOLLOWING QUESTIONS

1. Define a Transaction.
2. Explain with the help of an example that why should a transaction be executed as a whole or it should be not executed at all.
3. Distinguish between COMMIT and ROLLBACK.
4. Write the purpose of inserting Savepoints in a transaction.
5. What happens when Autocommit is set on?
6. Write SQL statement to set Autocommit to off.
7. What does the ROLLBACK statement do?
8. How do you tell the system that a transaction is beginning?
9. Why do we generally execute a COMMIT statement before beginning a transaction?
10. Name two statements that can be used to end a transaction?
11. Does executing the COMMIT or ROLLBACK statement end the current transaction?
12. What happens to the current transaction if a DDL Statement is executed?

LAB EXERCISES

a) **Perform the following tasks:**

Start MySQL session .

Create a table named Student with columns RollNumber, Name and Marks.

Start a transaction and insert two rows to the Student table.

Verify the inserts by SELECT statement.

Commit the changes.

Start another transaction.

Delete a row that was recently inserted.

Verify that the row has been deleted.

Rollback the changes.

Verify that the delete has been cancelled.

b) A table named ITEM has the following contents:

```
+-------+--------------+----------+
| icode | iname        | iprice   |
+-------+--------------+----------+
|   101 | CHAIR        |  1500.00 |
|   102 | DINING TABLE | 24000.00 |
+-------+--------------+----------+
```

Write the output that will be displayed by each SELECT statement as the SQL statements given below are executed:

```
mysql> SELECT * FROM ITEM;
mysql> SET AUTOCOMMIT = 0;
mysql> INSERT INTO ITEM VALUES(103,'COFFEE TABLE',340);
mysql> SELECT * FROM ITEM;
mysql> ROLLBACK;
mysql> SELECT * FROM ITEM;
mysql> START TRANSACTION;
mysql> UPDATE ITEM SET IPRICE = IPRICE +200;
mysql> SAVEPOINT S1;
mysql> UPDATE ITEM SET IPRICE = IPRICE +400;
mysql> SELECT * FROM ITEM;
mysql> ROLLBACK TO  S1;
mysql> SELECT * FROM ITEM;
```

Now verify the output by creating the table using MySQL and executing the above statements.

c) A table named Bill has the following rows:

```
+-----------+-----------+------------+-----------+
| Order_Num | cust_code | bill_date  | Bill_Amt - |
+-----------+-----------+------------+-----------+
|         1 | C101      | 2010-08-02 |    2300   |
|         2 | C105      |  2010-08-02|    5500   |
|         3 | C099      |  2010-08-23|    3000   |
|         4 | C165      |  2010-09-24|     6500  |
|         5 | C105      |  2010-09-24|     1400  |
+-----------+-----------+------------+-----------+
```

Write the output that will be displayed due to last SQL SELECT statement:

```
mysql> START TRANSACTION;
mysql> INSERT INTO BILLS VALUES(7,'C101','2010-09-02',5000);
mysql> UPDATE BILLS SET Bill_Amt = Bill_Amt+500 WHERE
Order_Num = 3;
mysql> SAVEPOINT A;
mysql> INSERT INTO BILLS VALUES(8,'C97','2010-09-03',4500);
mysql> DELETE FROM BILL WHERE cust_code = 'C105';
mysql> ROLLBACK TO A;
mysql> SELECT * FROM bills;
```

Now verify the output by actually executing the statements.

TIME BOUND TEAM BASED EXERCISE

(Team size recommended : 3 students each team)

Suppose the Government has nominated you as Head of Ministry of Information Technology. You have come out with the idea of a National Identity Card that should be