Chapter 10:

Advanced RDBMS Concepts

Informatics Practices

Class XII (CBSE Board)

Revised as per
CBSE
Curriculum
2015

“Open Teaching-Learning Material”

What is Database Transaction?

- In general, an event of access or modify the record from a database is called Database Transaction.
- **A Transaction is a Logical Unit of Work (LUW) on the database that must succeed or fail entirely.**
- A database transaction may contains several statement or commands but works as an atomic operation on the database.

Properties of a Transaction (ACID Properties)

A Transaction possesses the following **ACID** properties.

- **Atomicity: (All-or-None)**

  A transaction is executed entirely or none. No any transaction is allowed to run partially.

- **Consistency:**

  A transaction must leave the database in Consistent state after completion. No any transaction is allowed to leave the database in In-consistent state, because before execution it was in consistent state.

- **Integrity:**

  Transaction is an atomic unit, it must be executed independently, no any other transaction should interfere during the execution of a transaction to avoid conflicts.

- **Durability:**

  The changes made by the transaction are permanent in nature i.e. effect of a transaction is recorded permanently.

Database Transaction – an example

A transaction may contains several commands like SELECT, DELETE and UPDATE etc. to perform an specific action (work) on the database.

Suppose an amount of 1000/- is transferred from Ajay' s account (AccountNo 1005) to Mohan's Account (Account No 1102), in ACCOUNT table, then it can be represented as-

```
mysql> START TRANSACTION;
mysql> UPDATE ACCOUNT SET Balance = Balance-1000
       Where AccountNo=1005;

mysql> UPDATE ACCOUNT SET Balance = Balance+1000
       Where AccountNo=1102;
Mysql> COMMIT;
```

Ajay's Balance is debited

Mohan's Balance is credited

Transaction Control Statements in MySQL

MySQL offers the following Transaction handling statements-

- **START TRANSACTION**

  Represents the start of a transaction.

- **COMMIT [Work]**

  Represents the end of a transaction. It saves all the changes on a database permanently.

- **SAVEPOINT <Savepoint_Name>**

  It creates a flag or mark during the execution of transaction which can be used to cancel the transaction partially, if required.

- **ROLLBACK [To SAVEPOINT <savepoint_Name>]**

  It cancels the effect of a transaction and restores the previous state of the database (works like UNDO operation). A partial rollback can be done using Save Points created in the transaction.

- **Set Autocommit**

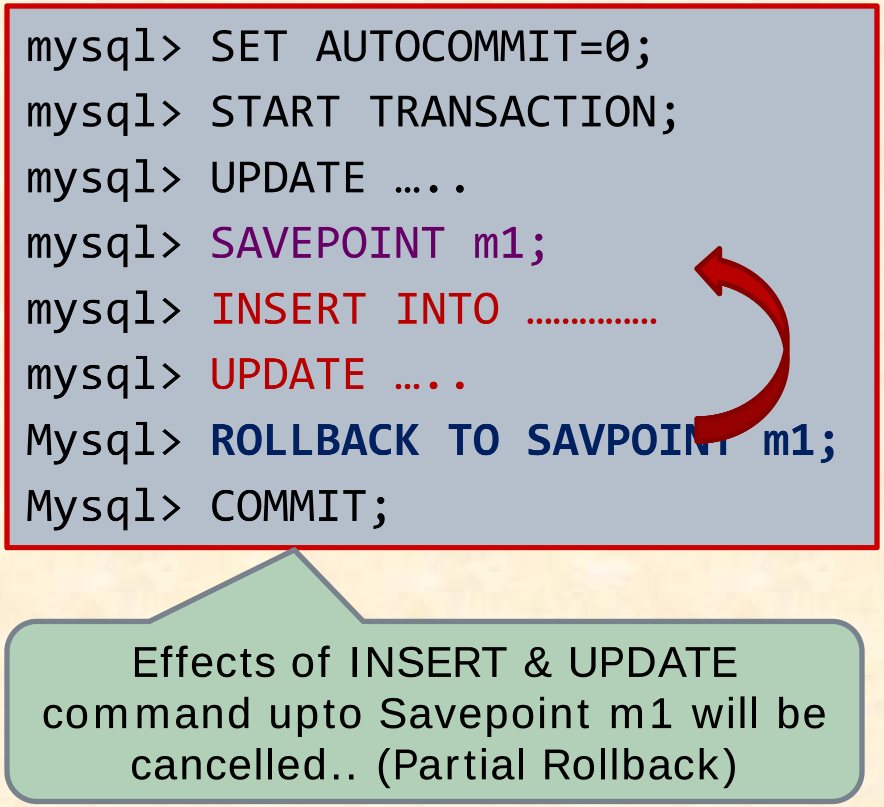
  If Autocommit is Enabled , the changes are immediately saved after completion of the command, without using **Commit** command explicitly.

```
mysql> Set Autocommit=1;    (enables Autocommit feature)
mysql> Set Autocommit=0;    (disables Autocommit feature)
```

Save Points and Rollback of Transactions

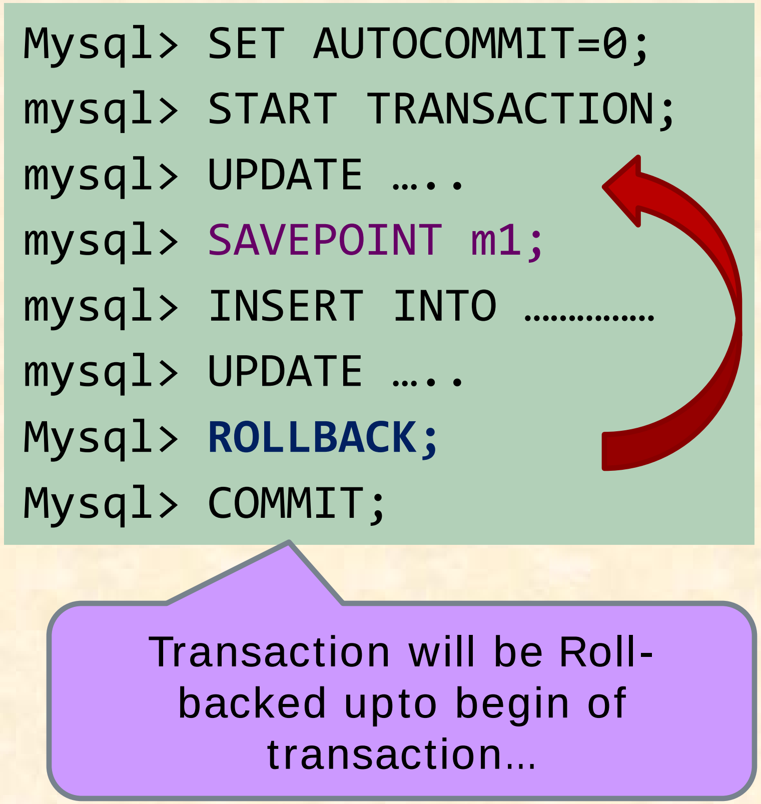
Save Points are flag or marker created during the execution of transaction which can be used to cancel the transaction partially.

Example :



```
mysql> SET AUTOCOMMIT=0;
mysql> START TRANSACTION;
mysql> UPDATE …..
mysql> SAVEPOINT m1;
mysql> INSERT INTO ……………
mysql> UPDATE …..
Mysql> ROLLBACK TO SAVPOINT m1;
Mysql> COMMIT;
```

Effects of INSERT & UPDATE command upto Savepoint m1 will be cancelled.. (Partial Rollback)



```
Mysql> SET AUTOCOMMIT=0;
mysql> START TRANSACTION;
mysql> UPDATE …..
mysql> SAVEPOINT m1;
mysql> INSERT INTO ……………
mysql> UPDATE …..
Mysql> ROLLBACK;
Mysql> COMMIT;
```

Transaction will be Roll-backed upto begin of transaction…

Autocommit and Rollback

MySQL offers AUTOCOMMIT feature, which automatically commits changes immediately after execution of command in the transactions, if it is enabled (i.e. SET AUTOCOMMIT=1)

Example :

```
mysql> SET AUTOCOMMIT=0;
mysql> START TRANSACTION;
mysql> INSERT …..
mysql> ……….
Mysql> ROLLBACK;
```

Record is not inserted, since it is undone by Rollback command.

```
mysql> SET AUTOCOMMIT=1
mysql> START TRANSACTION;
mysql> INSERT INTO……….
Mysql> …..
Mysql> ROLLBACK;
```

Record is added and automatically committed too… (since Autocommit is enabled.)

Once a transaction has been committed(saved) by Commit command Or AUTOCOMMIT, it can’t be rollbacked.