TCL Statements with examples

TCL statements defined…

- TCL is short name of Transactions Control Language.
- These commands are used to manage the transactions in the database.
- Used to manage the changes made by DML statements.
- It also allows the statements to be grouped together into logical transactions.

COMMIT statements…

- Commit command is used to permanently save any transaction into the database.
- When we use Commit in any query then the change made by that query will be permanent and visible.

```
START TRANSACTION;
SELECT
   @orderNumber:=MAX(orderNUmber)+1
FROM
   orders;
INSERT INTO orders(orderNumber,orderDate,requiredDate,shippedDate,status,customerNumber)
VALUES(@orderNumber,'2005-05-31', '2005-06-10', '2005-06-11','In Process',145);
COMMIT;
```

ROLLBACK statements…

- Rollback is used to undo the changes made by any command but only before a commit is done.
- We can't Rollback data which has been committed in the database with the help of the commit keyword.
- Some statements cannot be rolled back. DDL statements such as CREATE or DROP databases, CREATE, ALTER or DROP tables or stored routines.
- InnoDB supports the SQL statements SAVEPOINT, ROLLBACK TO SAVEPOINT, RELEASE SAVEPOINT and the optional WORK keyword for ROLLBACK.

SAVEPOINT statements…

- SAVEPOINT command is used to temporarily save a transaction so that you can roll back to that point whenever necessary.
- Savepoint names must be distinct within a given transaction.
- After a savepoint has been created, you can either continue processing, commit your work, roll back the entire transaction, or roll back to the savepoint.

```
SAVEPOINT identifier  ;
ROLLBACK [WORK] TO [SAVEPOINT] identifier  ;
RELEASE SAVEPOINT identifier;
```

LOCK AND UNLOCK Table statements…

- MySQL enables client sessions to acquire table locks explicitly for the purpose of cooperating with other sessions for access to tables
- Or to prevent other sessions from modifying tables during periods when a session requires exclusive access to them.
- A session can acquire or release locks only for itself.
- LOCK TABLES explicitly acquires table locks for the current client session.
- You must have the LOCK TABLES privilege, and the SELECT privilege for each object to be locked.
- UNLOCK TABLES explicitly releases any table locks held by the current session.
- LOCK TABLE messages READ;
- UNLOCK TABLES;