# MySQL RDBMS

Trainer: Mr. Nilesh Ghule

# Transaction

- Transaction management
    - START TRANSACTION;
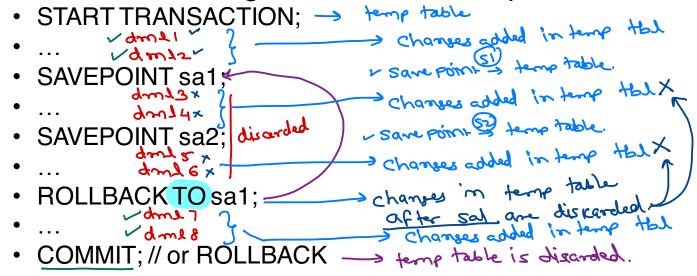    - … dml 1
          dml 2
          dml 3
    - COMMIT WORK; → final save

    - START TRANSACTION;
    - … dml 1;
          dml 2;
          dml 3;
    - ROLLBACK WORK; → discard

- In MySQL autocommit variable is by default 1. So each DML command is auto-committed into database.
    - SELECT @@autocommit;

- Changing autocommit to 0, will create new transaction immediately after current transaction is completed. This setting can be made permanent in config file.
    - SET autocommit=0;

# Transaction

- Save-point is state of database tables (data) at the moment (within a transaction).
- It is advised to create save-points at end of each logical section of work.
- Database user may choose to rollback to any of the save-point.
- Transaction management with Save-points
  - START TRANSACTION; → temp table
  - …    ✓ dml1 ✓
         ✓ dml2 ✓    } → changes added in temp tbl
  - SAVEPOINT sa1;    ↳ save point S1 → temp table.
  - …    dml3 ✗
         dml4 ✗    } → changes added in temp tbl ✗    | discarded
  - SAVEPOINT sa2;    ↳ save point S2 → temp table.
  - …    dml5 ↗
         dml6 ✗    } → changes added in temp tbl ✗
  - ROLLBACK TO sa1;    → changes in temp table after sa1 are discarded.
  - …    ✓ dml7
         ✓ dml8    } → changes added in temp tbl
  - COMMIT; // or ROLLBACK    → temp table is discarded.

- Commit always commit the whole transaction.
- ROLLBACK or COMMIT clears all save-points.

# Transaction

- Transaction is set of DML statements.
- If any DDL statement is executed, current transaction is automatically committed.
- Any power failure, system or network failure automatically rollback current state.
- Transactions are isolated from each other and are consistent.

# Row locking

- When an user update or delete a row (within a transaction), that row is locked and becomes read-only for other users.

- The other users see old row values, until transaction is committed by first user.

- If other users try to modify or delete such locked row, their transaction processing is blocked until row is unlocked.

- Other users can INSERT into that table. Also they can UPDATE or DELETE other rows.

- The locks are automatically released when COMMIT/ROLLBACK is done by the user.

- This whole process is done automatically in MySQL. It is called as "OPTIMISTIC LOCKING".

# Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>