UNITED
GROUP OF INSTITUTIONS

Web Technology
(KCS-602)
Unit 4
Transaction Processing, Stored Procedures

Prepared By

Abhishek Kesharwani

Assistant Professor,UCER Naini,Allahabad

Transactions

- A transaction in MySQL is a **sequential group of statements**, queries, or operations such as select, insert, update or delete to perform as a one single work unit that can be committed or rolled back.
- If the transaction makes multiple modifications into the database,

two things happen:

- Either all modification is successful when the transaction is committed.
- Or, all modifications are undone when the transaction is rollback.

Properties of Transaction

The transaction contains mainly four properties, which referred to as **ACID** property. The ACID property stands for:

- Atomicity
- Consistency
- Isolation
- Durability

- **Atomicity:** This property ensures that all statements or operations within the transaction unit must be executed successfully. Otherwise, if any operation is failed, the whole transaction will be aborted, and it goes rolled back into their previous state.

- **Consistency:** This property ensures that the database changes state only when a transaction will be committed successfully. It is also responsible for protecting data from crashes.
- **Isolation:** This property guarantees that each operation in the transaction unit operated independently. It also ensures that statements are transparent to each other.

- **Durability:** This property guarantees that the result of committed transactions persists permanently even if the system crashes or failed.

MySQL transaction statements

By default, MySQL automatically commits the changes permanently to the database.

To force MySQL not to commit changes automatically, you use the following statement.

```
SET autocommit = 0;


SET autocommit = OFF
```

COMMIT example

In order to use a transaction, you first have to break the SQL statements into logical portions and determine when data should be committed or rolled back.

- INSERT INTO Orders(order_id, prod_name, order_num, order_date)   VALUES (8, 'Speaker', 6065, '2020-02-18');
- COMMIT;

ROLLBACK example

- START TRANSACTION;
- DELETE FROM orders;
- ROLLBACK;

- START TRANSACTION;
- SELECT * FROM Orders;
- INSERT INTO Orders(order_id, prod_name, order_num, order_date)
- VALUES (6, 'Printer', 5654, '2020-01-10');
- SAVEPOINT my_savepoint;

- INSERT INTO Orders(order_id, prod_name, order_num, order_date)
- VALUES (7, 'Ink', 5894, '2020-03-10');
- ROLLBACK TO SAVEPOINT my_savepoint;

MySQL Stored Procedure

- A procedure (often called a stored procedure) is a collection of pre-compiled SQL statements stored inside the database.
- It is a subroutine or a subprogram in the regular computing language.
- A procedure always contains a name, parameter lists, and SQL statements.
- It was first introduced in MySQL version 5. Presently, it can be supported by almost all relational database systems.

How to create a procedure?

```
DELIMITER &&
CREATE PROCEDURE procedure_name [[IN | OUT | INOUT] parameter_name datatype [, parameter datatype]) ]
BEGIN
   Declaration_section
   Executable_section
END &&
DELIMITER ;
```

DROP PROCEDURE

```
DROP PROCEDURE [IF EXISTS]
stored_procedure_name;
```