



**TRANSACTION**

# WHAT ARE 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:

# WHAT ARE TRANSACTIONS?

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

In other words, a transaction cannot be successful without completing each operation available in the set. It means if any statement fails, the transaction operation cannot produce results.

# BENEFITS OF TRANSACTIONS

## Data Integrity

- Transactions guarantee that data changes are either fully committed or fully rolled back
- Prevents partial updates that could leave the database in an inconsistent state

# BENEFITS OF TRANSACTIONS

## Concurrency Control

- Transactions manage access to shared data resources
- Prevent conflicts and race conditions between multiple users/processes

# BENEFITS OF TRANSACTIONS

## Reliability

- Transactions provide a way to recover from system failures or errors
- Ensure that the database remains in a valid state

# ACID PROPERTIES OF TRANSACTIONS

## 1. Atomicity

- A transaction must be completed in its entirety or not at all

## 2. Consistency

- Transactions must preserve the integrity of the database

## 3. Isolation

- Transactions must be independent of each other

## 4. Durability

- The effects of a committed transaction must persist, even in the event of a system failure

# TRANSACTION WORKFLOW

## 1. Begin Transaction

- Marks the start of a transaction block

## 2. Perform Operations

- Execute SQL statements within the transaction

## 3. Commit Transaction

- Finalizes the transaction and makes the changes permanent

## 4. Rollback Transaction

- Cancels the transaction and undoes all changes made during the transaction

## EXAMPLE:

- -- 1. Start a new transaction
- `START TRANSACTION;`
- -- 2. Get the highest income
- `SELECT @income:= MAX(income) FROM employees;`
- -- 3. Insert a new record into the employee table
- `INSERT INTO employees(emp_id, emp_name, emp_age, city, income)`
- `VALUES (111, 'Alexander', 45, 'California', 70000);`
- -- 4. Insert a new record into the order table
- `INSERT INTO Orders(order_id, prod_name, order_num, order_date)`
- `VALUES (6, 'Printer', 5654, '2020-01-10');`
- -- 5. Commit changes
- `COMMIT;`

MySQL 8.0 Command Line Client

```
mysql> START TRANSACTION;  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> SELECT @income:= MAX(income) FROM employees;
```

```
+-----+  
| @income:= MAX(income) |  
+-----+  
|           5000000 |  
+-----+
```

```
1 row in set, 1 warning (0.00 sec)
```

```
mysql> INSERT INTO employees(emp_id, emp_name, emp_age, city, income)  
-> VALUES (111, 'Alexander', 45, 'California', 70000);
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO Orders(order_id, prod_name, order_num, order_date)  
-> VALUES (6, 'Printer', 5654, '2020-01-10');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> COMMIT;
```

```
Query OK, 0 rows affected (0.17 sec)
```

## REFERENCE

- <https://www.javatpoint.com/mysql-transaction>
- <https://dev.mysql.com/doc/search/?d=333&p=1&q=transaction>