


Home

 Home

Learning Modules

 Basic Queries

 DDL Commands

 DML Operations

 DCL Controls


 **TCL Management**


 Aggregate Functions


Tools

 SQL Query Editor


 Quick Links

 Documentation

 Tips & Tricks

 Learning Path

MySQL Handbook v2.0

Built with  using Streamlit

 Transaction Key Points

Transaction States

Best Practices



# MySQL Handbook

*Complete Guide to MySQL Database*



## Transaction Control Language (TCL)

### Master MySQL Transaction Management

 **Fundamentals**

 Savepoints

 Isolation Levels

 Practice Lab



## Transaction Fundamentals



### ACID Properties

ACID adalah fondasi dari sistem database yang reliable:



- **Atomicity (Atomisitas)** - All or nothing principle
- **Consistency (Konsistensi)** - Data integrity maintained
- **Isolation (Isolasi)** - Concurrent transactions don't interfere
- **Durability (Durabilitas)** - Changes survive system failures

#### ACID Properties



 Contoh ACID dalam kehidupan nyata:

#### Transfer Bank (Atomicity):

- Debit dari rekening A: 
- Credit ke rekening B: 
- Jika salah satu gagal → ROLLBACK semua

#### Consistency:

- Total uang sebelum = Total uang sesudah

#### Isolation:

- Transfer A tidak melihat transfer B yang belum selesai

#### Durability:

- Setelah COMMIT, data tersimpan permanen

## Basic Transaction Syntax

### Basic Transaction Structure:

```
-- Start transaction
START TRANSACTION;

-- atau
BEGIN;

-- Your SQL operations
UPDATE accounts SET balance = balance - 1000
WHERE account_id = 1;

UPDATE accounts SET balance = balance + 1000
WHERE account_id = 2;

-- Commit changes
COMMIT;

-- Or rollback if error
-- ROLLBACK;
```

### Transaction with Error Handling:

```
-- Start transaction with error handling
START TRANSACTION;

-- Check if sufficient balance
SELECT balance FROM accounts
WHERE account_id = 1 AND balance >= 1000;

-- If balance sufficient, proceed
UPDATE accounts SET balance = balance - 1000
WHERE account_id = 1;

UPDATE accounts SET balance = balance + 1000
WHERE account_id = 2;

-- Commit if all operations successful
COMMIT;

-- Rollback on any error
-- ROLLBACK;
```

## Interactive Transaction Demo

### Bank Transfer Simulator

#### Transfer Configuration:

From Account:

ACC1001



To Account:

ACC1004



Transfer Amount:

1000.00

- +

☐ Simulate Error Condition

☐ Use Savepoint

#### Account Balances (Before):

	Account	Balance
0	ACC1001	5000
1	ACC1002	7500
2	ACC1003	3200
3	ACC1004	8900
4	ACC1005	4500
5	ACC1006	6700

 Execute Transfer