Title :- Moqui MySQL XA Transaction Failure on Windows (MySQL 8, Bitronix)

Impact Level:

- Learning environment only

2. Context / Background

Answer:

- **What system?**
- **What environment?**
- **What task?**

Write facts only.

Example:

- **Framework: Moqui 3.x**
- **OS: Windows 10**
- **Database: MySQL 8.0.x**
- **Transaction Manager: Bitronix (JTA/XA)**
- **Goal: Configure local MySQL datasource for Moqui training assignment**

No opinions here.

3. What We Were Trying to Do

This is intent, not steps.

Example:

Configure Moqui to connect to a locally installed MySQL database using a non-root user and load initial data using gradlew load.

This tells future-you *why* you touched this system.

4. What Happened (Observed Behavior)

Only observable facts:

- **errors**
- **logs**
- **symptoms**

Example:

- **gradlew load failed during datasource initialization**
- **Error: XAER_RMERR: Fatal error occurred in the transaction branch**
- **Repeated retries by Bitronix XA datasource**
- **Application failed to start**

Do not explain yet.

5. What Went Wrong (Initial Hypotheses)

This section shows your thinking evolution.

List suspected causes, even wrong ones.

Example:

- **MySQL JDBC driver mismatch**
- **Incorrect database charset (UTF-8 vs utf8mb4)**
- **Incorrect datasource configuration**
- **Missing JDBC URL**
- **Moqui configuration error**

This is valuable because it shows debugging process.

6. Key Clues That Changed Understanding

This is the most important section.

Write:

- **exact log lines**
- **behaviors that narrowed the problem**

Example:

- **Error consistently referenced XA and XA RECOVER**
- **Bitronix attempted recovery during startup**
- **Same configuration worked when using MySQL root user**
- **Same database worked with non-XA tools**

This is where understanding starts forming.

7. Root Cause Analysis

This is the why, not the what.

Use cause-and-effect reasoning.

Example:

Moqui uses XA-capable datasources by default.
MySQL stores prepared XA transactions persistently.
A previous failed XA transaction left an in-doubt state in MySQL.
The non-root database user did not have sufficient privileges to recover or resolve XA transactions.
During startup, Bitronix attempted XA recovery and failed, causing datasource initialization to abort.

This is the heart of the document.

8. Why It Worked with Root User

This section prevents future confusion.

Example:

The MySQL root user has unrestricted privileges, including internal transaction recovery capabilities.
Using root allowed Bitronix to successfully recover or bypass existing XA states, allowing the application to start normally.

This explains the "magic fix".

9. Resolution / Workaround

What actually solved it.

Example:

- **Switched datasource user to root**
- **Application started and loaded data successfully**

Be honest if it's a workaround, not a perfect fix.

10. Correct Fix (Ideal Solution)

This shows engineering maturity.

Example:

- **Ensure clean XA state (XA RECOVER, XA ROLLBACK)**
- **Properly grant XA-related privileges**
- **Or reset MySQL XA state on Windows**
- **Prefer Linux environment for XA-heavy frameworks**

11. Lessons Learned

Bullet points. No stories.

Example:

- **XA transactions are not limited to multiple databases**
- **Transaction managers may enforce XA even for single resources**
- **Windows + MySQL XA can be fragile**
- **Root masks privilege and recovery issues**
- **Always read errors literally (XA ≠ JDBC)**

12. Preventive Actions (Future)

What you'll do differently next time.

Example:

- **Prefer Linux for enterprise frameworks**
- **Validate XA behavior early**
- **Avoid assuming database users are equivalent to root**
- **Document recovery steps for XA errors**