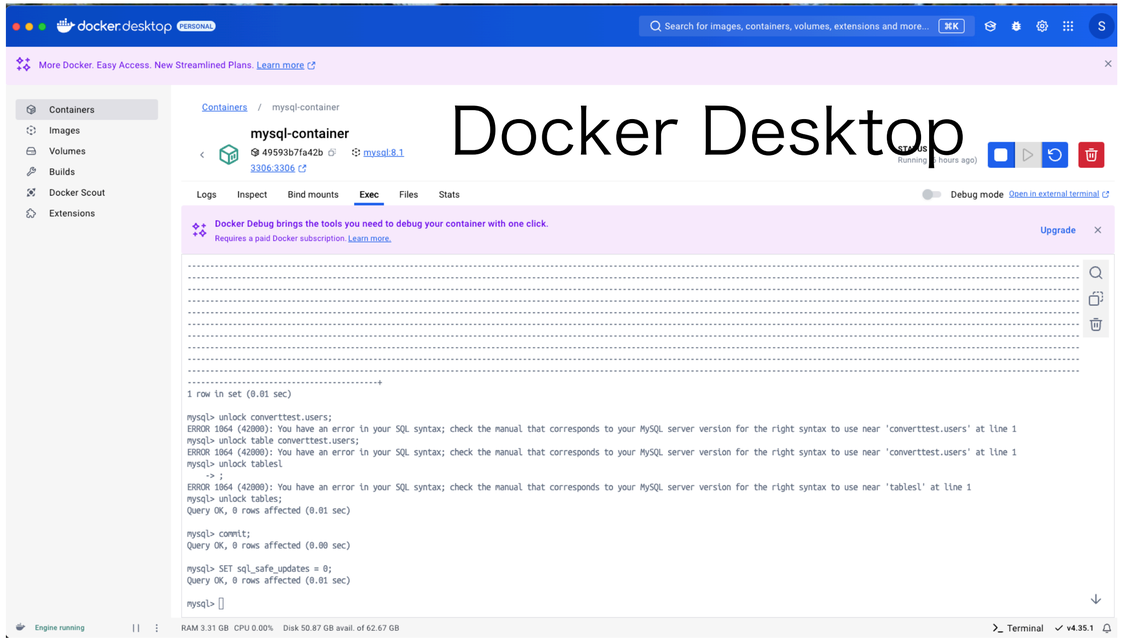
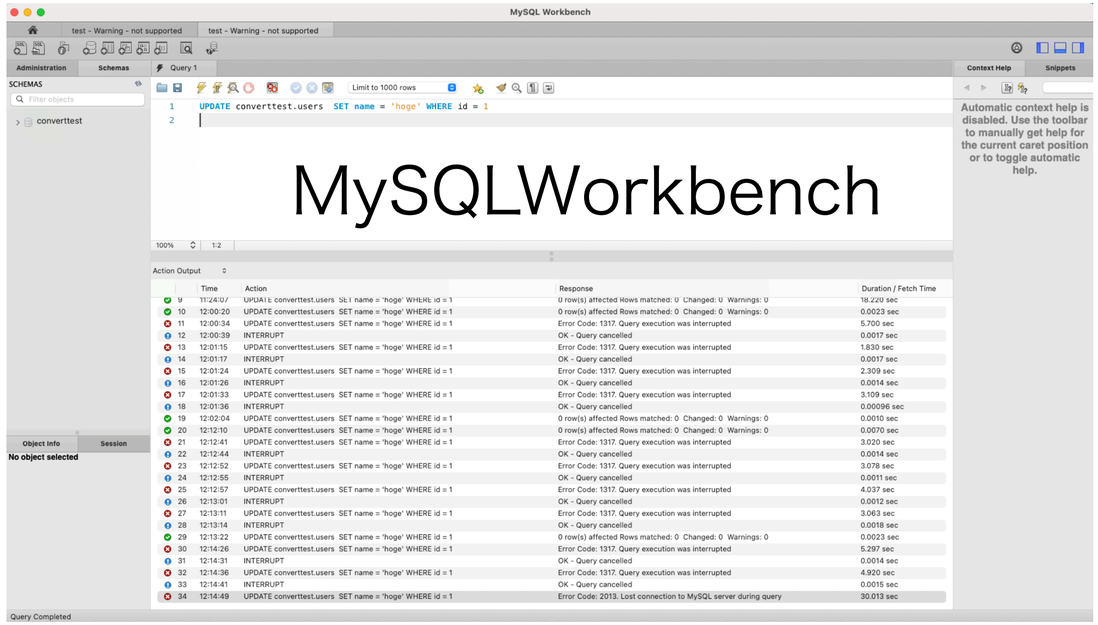
週末の確認結果

おうちの環境でテストしたので、結果に100%の断言はできんので

あしからず

2024.11.16 itohiro



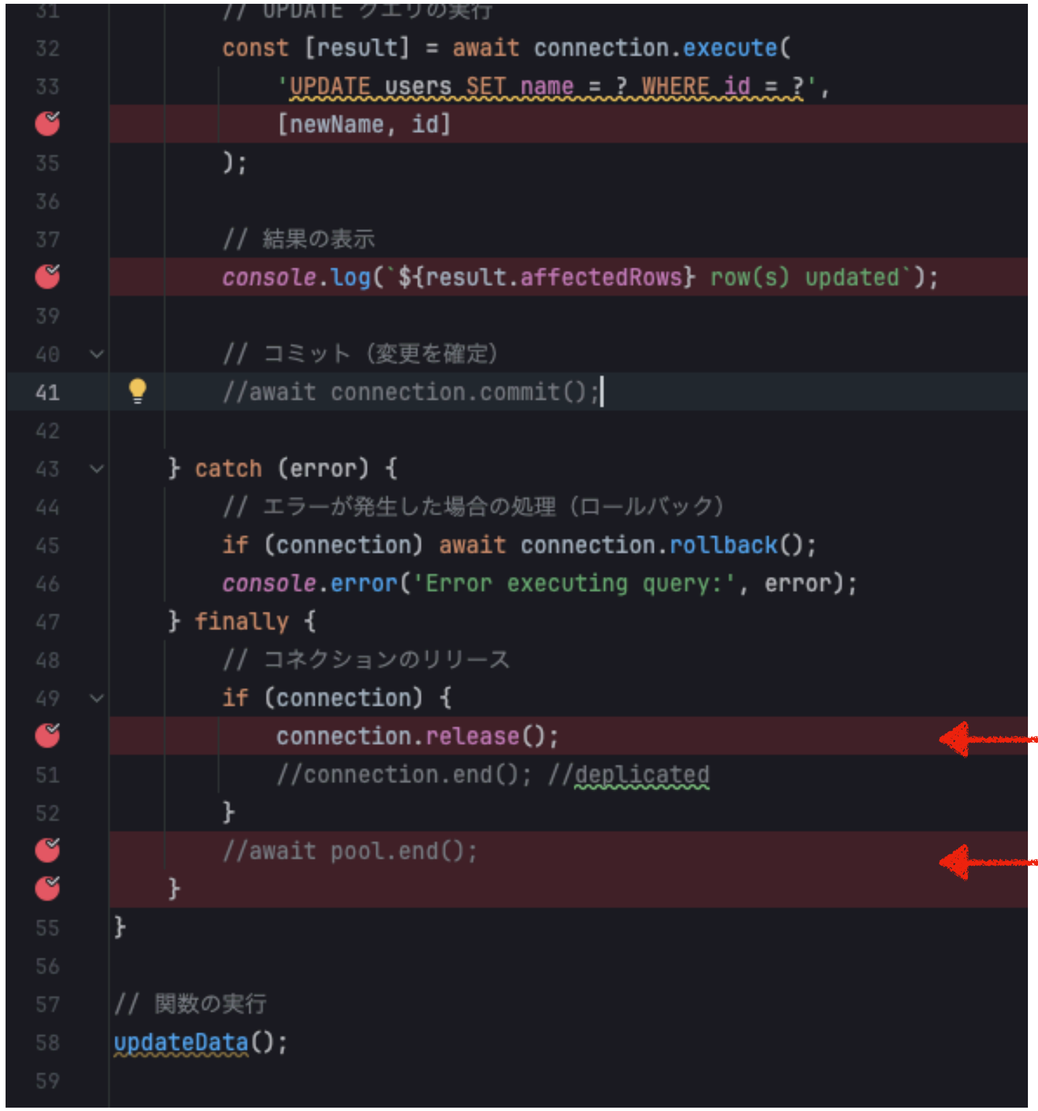


Intellij

現行バッチと同じ作り方で作成。

nodeJsで
ConnectionPool
Transaction
を使ってDBを更新するプログラム
書いてとChatGPTにお願いして
作成したファイル。

現行のロジックと大差はないと思う。



```
31        // UPDATE クエリの実行
32        const [result] = await connection.execute(
33            'UPDATE users SET name = ? WHERE id = ?',
34            [newName, id]
35        );
36
37        // 結果の表示
38        console.log(`${result.affectedRows} row(s) updated`);
39
40        // コミット（変更を確定）
41        //await connection.commit();
42
43    } catch (error) {
44        // エラーが発生した場合の処理（ロールバック）
45        if (connection) await connection.rollback();
46        console.error('Error executing query:', error);
47    } finally {
48        // コネクションのリリース
49        if (connection) {
50            connection.release();
51            //connection.end(); //deplicated
52        }
53        //await pool.end();
54    }
55 }
56
57 // 関数の実行
58 updateData();
59
```

（方法）
行ロックを行うため、MySQLworkbenchに
UPDATE converttest.users SET name = 'hoge' WHERE id = 1
用意し、node.jsのプログラムを実行させながら、SQLを実行して、様子を確かめる。

プログラムはConnection poolとtransactionを使ったnodejsのプログラムをGPTに書いてもらって、稼働中のプログラムと呼び出し方法が合ってることを確認。

参考情報

Node + TypeScript で MySQL に接続して Read, Write してみる

https://qiita.com/tkm_kj/items/40d12693e601b298a0f9

Node.jsのMySQLでAsync/Awaitしたらプロセスが終わらない

https://qiita.com/saoshi/items/728c11a55e99dfb8ad8e

Connection release 時点ではUpdate不可。

pool.end ()を明示的実行すると、開放され、Update可能となった。

プログラムの終了以降も更新テーブルを行ロックしている可能性あり。
connection.release以外のコマンドも確認したが、ロックと直結しなかったので、放置。

pool.end()のみ有効であるが、他のプログラムのpoolに影響しないかの確認は必要。

```
// 必要なモジュールのインポート
const mysql = require('mysql2/promise');

// コネクションプールの作成
const pool = mysql.createPool({
    host: 'localhost',
    user: 'dbuser',  // MySQLのユーザー名
    password: 'P@ssw0rd',  // MySQLのパスワード
    database: 'converttest',  // 対象のデータベース名
    waitForConnections: true,
    connectionLimit: 10,  // プール内の最大コネクション数
    queueLimit: 0
});

// 非同期関数でMySQLの更新処理を行う
async function updateData() {
    let connection;
    try {
        // コネクションプールから接続を取得
        connection = await pool.getConnection();

        console.log('Connected to the database');

        // 更新したいデータ
        const id = 1;  // 更新対象のID
        const newName = 'Updated Name';  // 更新後の名前

        // トランザクションの開始（必要に応じて）
        await connection.beginTransaction();

        // UPDATE クエリの実行
        const [result] = await connection.execute(
            'UPDATE users SET name = ? WHERE id = ?',
            [newName, id]
        );

        // 結果の表示
        console.log(`${result.affectedRows} row(s) updated`);

        // コミット（変更を確定）
        //await connection.commit();

    } catch (error) {
        // エラーが発生した場合の処理（ロールバック）
        if (connection) await connection.rollback();
        console.error('Error executing query:', error);
    } finally {
        // コネクションのリリース
        if (connection) {
            connection.release();
            //connection.end(); //deplicated
        }
        //await pool.end();
    }
}

// 関数の実行
updateData();
```