

## [Thực hành] MySQL JDBC Transaction

### Mục tiêu

Luyện tập sử dụng JDBC transaction

### Mô tả

Cập nhật ứng dụng WEB quản lý User:

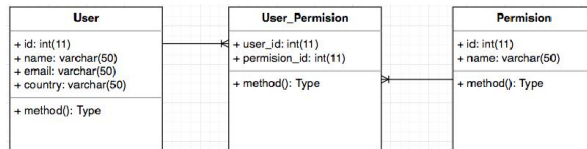
<https://james.codegym.vn/mod/assign/view.php?id=10074>

Cụ thể như sau:

Trong bài này bạn sẽ luyện tập sử dụng phương thức commit() và rollback() của đối tượng Connection để điều khiển transaction. Xây dựng phương thức mới để thực thi transaction.

### Hướng dẫn

Bước 1: Cập nhật cơ sở dữ liệu demo đã tạo ở bài trước. Tạo mới 2 bảng Permission và User\_Permission trong MySQL:



```
create table Permission(
id int(11) primary key,
name varchar(50)
);
```

```
create table User_Permission(
permission_id int(11),
user_id int(11)
);
```

Chúng ta chèn một số bản ghi cho bảng Permission như sau:

```
insert into Permission(id, name) values(1, 'add');
insert into Permission(id, name) values(2, 'edit');
insert into Permission(id, name) values(3, 'delete');
insert into Permission(id, name) values(4, 'view');
```

Trong ví dụ này, chúng ta sẽ chèn một bản ghi mới vào bảng User và cung cấp các quyền cho User mới này. Chúng ta sẽ triển khai việc chèn user và các quyền cho user trong cùng một transaction.

Các bước như sau:

1. Chèn một bản ghi vào bảng User, và lấy về id
2. Chèn một tập các user\_id và permission\_id vào bảng User\_Permission
3. Nếu tất cả các hành động trên thành công, gọi phương thức commit, trái lại, gọi phương thức rollback

Bước 2: Cập nhật interface IUserDAO:

```
void addUserTransaction(User user, int[] permission);
```

Bước 3: Cập nhật lớp UserDAO:

```

@Override
public void addUserTransaction(User user, int[] permissions) {

    Connection conn = null;

    // for insert a new user

    PreparedStatement pstmt = null;

    // for assign permission to user

    PreparedStatement pstmtAssignment = null;

    // for getting user id

    ResultSet rs = null;

    try {

        conn = getConnection();

        // set auto commit to false

        conn.setAutoCommit(false);

        //

        // Insert user

        //

        pstmt = conn.prepareStatement(INSERT_USERS_SQL, Statement.RETURN_GENERATED_KEYS);

        pstmt.setString(1, user.getName());

        pstmt.setString(2, user.getEmail());

        pstmt.setString(3, user.getCountry());

        int rowAffected = pstmt.executeUpdate();

        // get user id

        rs = pstmt.getGeneratedKeys();

        int userId = 0;

        if (rs.next())

            userId = rs.getInt(1);

        //

        // in case the insert operation successes, assign permission to user

        //

        if (rowAffected == 1) {

            // assign permission to user

            String sqlPivot = "INSERT INTO user_permission(user_id,permission_id) "

                + "VALUES(?,?)";

            pstmtAssignment = conn.prepareStatement(sqlPivot);

            for (int permissionId : permissions) {

                pstmtAssignment.setInt(1, userId);

                pstmtAssignment.setInt(2, permissionId);

                pstmtAssignment.executeUpdate();

            }

            conn.commit();

        } else {

            conn.rollback();

        }

    } catch (SQLException ex) {

        // roll back the transaction

        try {

            if (conn != null)

                conn.rollback();

        } catch (SQLException e) {

            System.out.println(e.getMessage());

        }

        System.out.println(ex.getMessage());

    } finally {

        try {

            if (rs != null) rs.close();

            if (pstmt != null) pstmt.close();

            if (pstmtAssignment != null) pstmtAssignment.close();

            if (conn != null) conn.close();

        } catch (SQLException e) {

            System.out.println(e.getMessage());

        }

    }

}

```

Bước 4: Cập nhật lớp UserServicelet:

Trong phương thức doGet():

```

case "permission":
    addUserPermission(request, response);

    break;

```

Xây dựng thêm phương thức addUserPermission():

```

private void addUserPermission(HttpServletRequest request, HttpServletResponse response) {

    User user = new User("quan", "quan.nguyen@codegym.vn", "vn");

    int[] permission = {1, 2, 4};

    userDao.addUserTransaction(user, permission);

}

```

Bước 5: Chạy ứng dụng và truy cập url sau:

<http://localhost:8080/users?action=permission>

Bước 6: Kiểm tra trong MySQL:

Bảng Users:

id	name	email	country
12	Minh	minh@codegym.vn	Viet Nam
13	Nogo Kante	kante@gmail.uk	UK
14	Chanathip Songprasin	cha@gmail.com	Thai Lan
19	kien	kienhoang@gmail.com	vn
20	tunghoang	tunghoang@gmail.com	vn

Bảng User\_Permission:

user_id	permission_id
17	1
17	2
17	3
19	1
19	2
19	4
20	2
20	4

Như các bạn thấy, dữ liệu đã được cập nhật thành công vào bảng Users và bảng User\_Permission trong cùng một transaction.

Checkout sang nhánh transaction để xem mã nguồn tham khảo: <https://github.com/codegym-vn/wbd-user-manager>

## Hướng dẫn nộp bài:

up bài tập lên github

paste link github vào phần nộp bài

### Submission status

Submission status	No attempt
Grading status	Not graded
Last modified	-
Submission comments	<a href="#">Comments (0)</a>

Add submission

You have not made a submission yet.

Previous Activity
Jump to...
Next Activity

#### CHƯƠNG TRÌNH

- Career
- Premium
- Accelerator

#### TÀI NGUYÊN

- Blog
- Tạp chí Lập trình
- AgileBreakfast

#### Follow Us

