Bài 13

JDBC QUERYING & TRANSACTION

Module: JWBD

Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài "JDBC & CRUD"

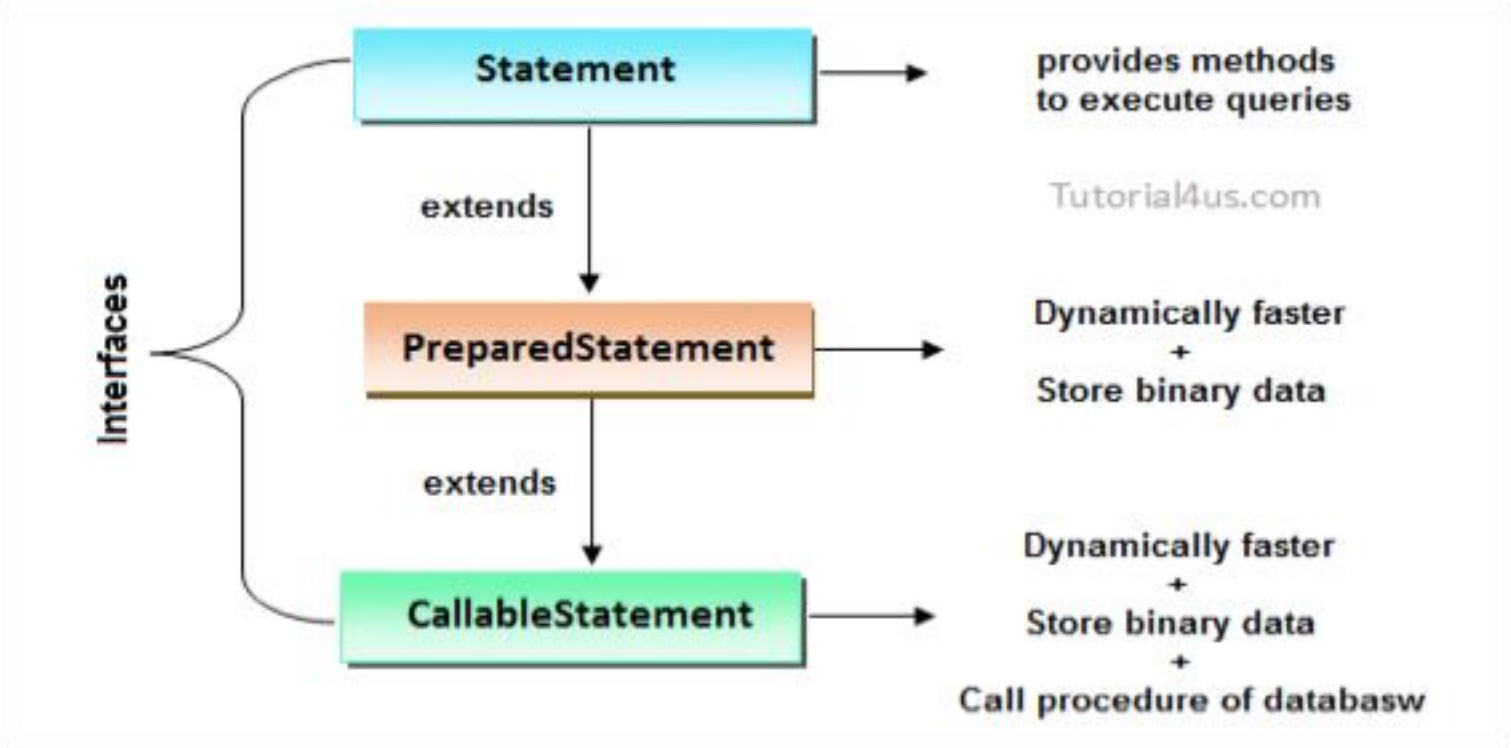
Tóm tắt lại các phần đã học từ bài "JDBC & CRUD"

Mục tiêu

- Sử dụng được JDBC gọi Stored Procedures
- Sử dụng được JDBC Transaction

CallableStatement interface trong JDBC

- CallableStatement Interface được sử dụng để thực thi Stored Procedure



Đối tượng CallableStatement

- CallableStatement là đối tượng kế thừa từ đối tượng PreparedStatement

- CallableStatement có các tính chất của 1 PreparedStatement
- Sử dụng CallableStatement để gọi các stored procedure

Cú pháp tạo CallableStatement

Chúng ta tạo 1 CallableStatement bằng cú pháp sau:

CallableStatement prepareCall(String sql) throws SQLException

Hoặc:

CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency) throws SQLException

Gọi Stored Procedure

Chúng ta triệu gọi các stored procedure bằng các cú pháp sau:

- Đối với procedure không có tham số: {call procedure_name}
- Đối với procedure có tham số: {call procedure_name(?,?,...)}
- Đối với procedure có giá trị trả về (function): {?=call procedure_name(?,?,...)}

Ví dụ:

```
CallableStatement cstmt =con.prepareCall("{call procedureName(?,?)}");
```

Trong đó các dấu ? thay thế cho các tham số kiểu IN, OUT, INOUT

Tham số kiểu IN

- Tham số IN là một tham số mà giá trị của nó chưa được biết khi lệnh SQL được tạo.
- Đối với tham số kiểu IN, ta sử dụng phương thức setXXX của đối tượng PrepareStatement để chuyển tham số vào.

Ví dụ:

```
cstmt.setString(1, "c%");
```

Tham số kiểu OUT

- Tham số OUT là một tham số mà giá trị của nó được cung cấp bởi lệnh SQL nó trả về. Giá trị của chúng được thu nhận thông qua phương thức getXXX().
- Trong đó, kiểu của tất cả tham số OUT phải được đăng ký trước khi thực thi storedprocedure.

Ví dụ:

```
stmt.registerOutParameter(1, java.sql.Types.INTEGER);
```

Tham số kiểu INOUT

Tham số INOUT là một tham số mà cung cấp cả giá trị input và output. Bạn gắn kết các biến với phương thức setXXX() và thu nhận giá trị với phương thức getXXX().

Đóng đối tượng CallableStatement

Để đóng đối tượng CallableStatement, bạn sử dụng phương thức close()

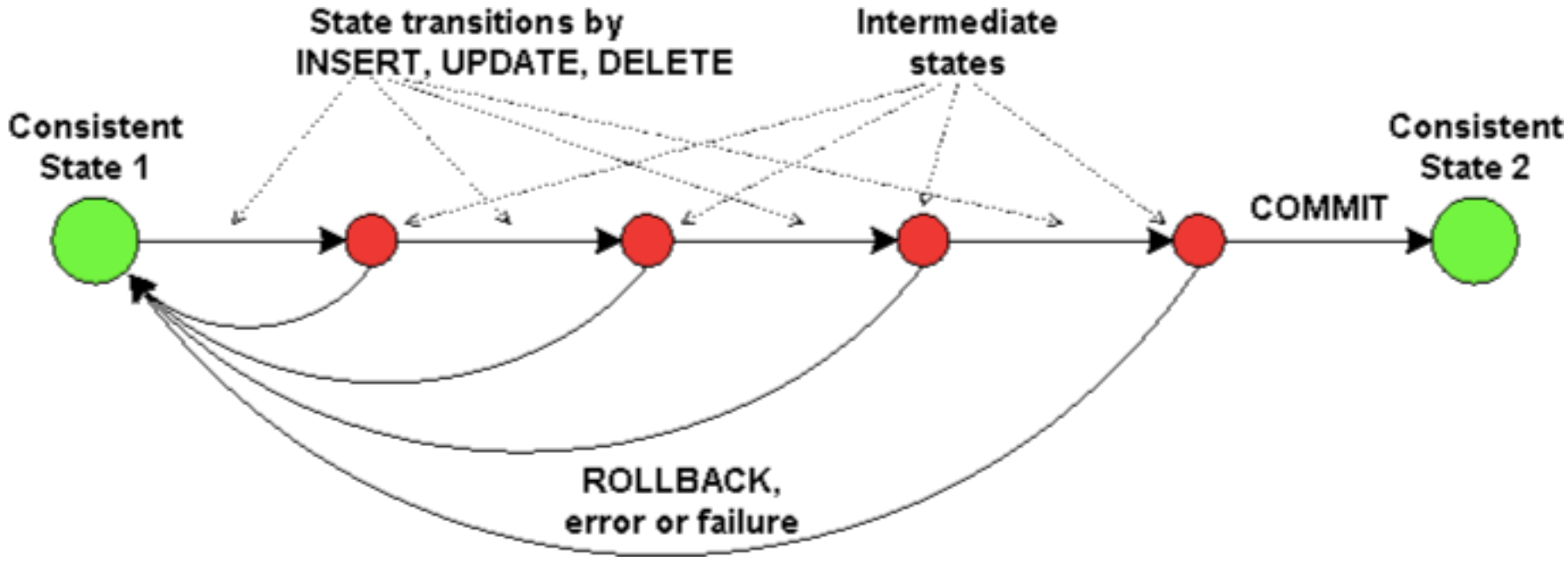
Ví dụ:

```
CallableStatement cstmt =null;
try {
  String SQL ="{call getTenSV (?, ?)}";
  cstmt =conn.prepareCall (SQL);
  . . .
}
catch (SQLException e) {
  . . .
}
finally {
  cstmt.close();
}
```

Transaction là gì ?

- Transaction (giao tác) là một tiến trình xử lý, có điểm bắt đầu và điểm kết thúc
- Transaction gồm nhiều phép thực thi nhỏ, trong đó mỗi phép thực thi được thực thi một cách tuần tự và độc lập theo nguyên tắc là tất cả thành công hoặc một phép thực thi thất bại thì cả tiến trình thất bại.

Transaction (minh hoạ)



Các thuộc tính ACID

Các thuộc tính ACID miêu tả rõ ràng nhất về Transaction. 4 thuộc tính này bao gồm Atomicity, Consistency, Isolation và Durability, trong đó:

- **Atomicity** nghĩa là tất cả thành công hoặc không.
- **Consistency** bảo đảm rằng tính đồng nhất của dữ liệu.
- **Isolation** bảo đảm rằng Transaction này là độc lập với Transaction khác.
- **Durability** nghĩa là khi một Transaction đã được ký thác thì nó sẽ vẫn tồn tại như thế cho dù xảy ra các lỗi, ...

Transaction trong JDBC

Trong JDBC, Connection Interface cung cấp các phương thức sau để quản lý transaction:

- void setAutoCommit(boolean status)
- void commit()
- void rollback()
- setSavepoint(String ten_cua_savepoint)
- releaseSavepoint(Savepoint ten_cua_savepoint)
- rollback (String ten_cua_savepoint)

Phương thức commit()

Phương thức commit() được sử dụng khi bạn đã thực hiện các thay đổi với cơ sở dữ liệu và muốn ký thác các thay đổi đó

```
// Mo mot ket noi
conn = DriverManager.getConnection(DB_URL,USER,PASS);

// Thiet lap auto commit la false.
conn.setAutoCommit(false);

// Chen mot hang vao trong bang sinhvienk60
String SQL = "INSERT INTO sinhvienk60 " +
             "VALUES (5, 'Tran Hung', 'Cuong', 7.5)";
stmt.executeUpdate(SQL);

// Chen them mot hang vao trong bang sinhvienk60
SQL = "INSERT INTO sinhvienk60 " +
             "VALUES (6, 'Vu Ngoc', 'Phan', 6.5)";
stmt.executeUpdate(SQL);

// Ky thac cac thay doi.
conn.commit();
```

Phương thức rollback()

Phương thức rollback() sử dụng để xóa các thay đổi đã được thực hiện trước đó để quay về trạng thái trước khi thực hiện thay đổi khi thấy rằng có lỗi xảy ra

```
// Neu xuat hien loi thi xoa sach cac thay doi
// va tro ve trang thai truoc khi co thay doi
try{
    if(conn!=null)
    conn.rollback();
}catch(SQLException se){
  se.printStackTrace();
}
```

Tổng kết

- CallableStatement Interface được sử dụng để thực thi Stored Procedure
- CallableStatement là đối tượng kế thừa từ đối tượng PreparedStatement
- CallableStatement có các tính chất của 1 PreparedStatement (tức có các tính chất của Statement(public interface CallableStatement extends PreparedStatement)
- Transaction (giao tác) là một tiến trình xử lý, có điểm bắt đầu và điểm kết thúc
- Transaction gồm nhiều phép thực thi nhỏ, trong đó mỗi phép thực thi được thực thi một cách tuần tự và độc lập theo nguyên tắc là tất cả thành công hoặc một phép thực thi thất bại thì cả tiến trình thất bại.

CODEGYM

Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: Packaging Dependences Management

Raising the bar