Cập nhật tháng 8 năm 2024

[Bài Đọc] Giới thiệu về PreparedStatement và cách tạo truy vấn

1. **Định nghĩa**

- **PreparedStatement** là một interface phụ của **Statement.** Đối tượng PreparedStatement có một số tính năng bổ sung hữu ích hơn so với đối tượng Statement. Thay vì các truy vấn mã hóa cứng, đối tượng PreparedStatement cung cấp một tính năng để thực hiện truy vấn tham số

2. **Ưu điểm của PreparedStatement**

- Khi PreparedStatement được tạo, truy vấn SQL được truyền dưới dạng tham số. Prepared Statement này chứa một truy vấn SQL được biên dịch trước, do đó khi PreparedStatement được thực thi, DBMS có thể chỉ cần chạy truy vấn thay vì biên dịch trước
- Chúng ta có thể sử dụng cùng một PreparedStatement và cung cấp các tham số khác nhau tại thời điểm thực thi
- Một lợi thế quan trọng của PreparedStatements là chúng ngăn chặn các cuộc tấn công SQL Injection

3. **Các bước sử dụng PreparedStatement**

- **Bước 1:** Tạo kết nối đến database

```
Connection con = ConnectionDB.openConnection();
```

- **Bước 2:** Sử dụng câu lệnh sql động có tham số như sau:

```
PreparedStatement ps = con.prepareStatement( sql: "SELECT * FROM customers");
```

- **Bước 3:** Execute query

```
ResultSet rs = ps.executeQuery();
```

- **Bước 4: Thực hiện in ra danh sách khách hàng**

```
while (rs.next())
{
    System.out.printf(
            "[ ID: %-5d | NAME: %-20s | AGE: %-5d]\n",
            rs.getInt( columnLabel: "id"),
            rs.getString( columnLabel: "name"),
            rs.getInt( columnLabel: "age")
    );
}
```

- **Các phương thức của PreparedStatement**
    - **setInt(int, int)**
        - Phương thức sử dụng để thiết lập giá trị số nguyên tại vị trí tham số đã cho
    - **setString(int, String)**
        - Phương thức sử dụng để thiết lập giá trị chuỗi tại vị trí tham số đã cho
    - **setFloat(int, float)**
        - Phương thức sử dụng để thiết lập giá trị số thực tại vị trí tham số đã cho
    - **setDouble(int, double)**
        - Phương thức sử dụng để thiết lập giá trị số thực tại vị trí tham số đã cho
    - **executeUpdate()**
        - Phương thức sử dụng để tạo, xóa, chèn, cập nhật, xóa, v.v. Nó trả về kiểu int
    - **executeQuery()**
        - Nó trả về một thể hiện của ResultSet khi một truy vấn chọn được thực thi

4. **Ví dụ:**

```java
public class Main  new *
{
    public static void main(String[] args) throws SQLException  new *
    {
        Connection con = ConnectionDB.openConnection();
        try
        {
            PreparedStatement ps = con.prepareStatement( sql: "SELECT * FROM customers");

            ResultSet rs = ps.executeQuery();

            while (rs.next())
            {
                System.out.printf(
                        "[ ID: %-5d | NAME: %-20s | AGE: %-5d]\n",
                        rs.getInt( columnLabel: "id"),
                        rs.getString( columnLabel: "name"),
                        rs.getInt( columnLabel: "age")
                );
            }
        }
        catch (Exception e)
        {
            throw new RuntimeException(e);
        }
        finally
        {
            ConnectionDB.closeConnection(con);
        }
    }
}
```

Link tài nguyên đọc thêm: https://www.geeksforgeeks.org/how-to-use-preparedstatement-in-java/

Danh sách các bài học