Báo cáo vè các hàm liên kết với cơ sở dữ liệu

Cơ sở dữ liệu là gì:

Cơ sở dữ liệu là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thõa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau.

Hiện nay có rất nhiều **hệ quản trị CSDL** như MYSQL, SQL SERVER, Oracle, MS Access

Nhưng trong netbeans cúng có 1 hệ quản trị CSDL dó là DATABASE DERBY nó cũng giống như các hệ quản trị csdl như trên nhưng nó lại thuận tiên hơn trong việc lấy các giá trị cho app của mình

Để tạo và lấy csdl từ trong derby thì chúng ta cần 1 số câu lênh như trong sql server như là:

Hàm tạo bảng

Create table “table\_name” (values kiểu giá trị)

Hàm insert giá trị vào bảng

Insert into “table\_name values)

Hàm lấy giá trị từ bảng

Select … from “table\_name” where “điều kiện”

Và 1 số hàm khác như

Hàm random:

Hàn random dùng để lấy ngẩu nhiên câu trả lời từ trong bảng

Connection

là phiên làm việc giữa ứng dụng java và cơ sở dữ liệu. Đối tượng Connection được sử dụng để tạo Statement, PreparedStatement, và DatabaseMetaData. Giao diện Connection cung cấp nhiều phương thức quản lý transaction như commit(), rollback(), …

Statement

cung cấp các phương thức để thực thi các câu lệnh truy vấn với cơ sở dữ liệu SQL. Statement interface là một nhà máy của ResultSet, tức là nó cung cấp phương thức để tạo ra đối tượng ResultSet.

**ResultSet**

 duy trì một con trỏ trỏ đến một hàng của một bảng. Ban đầu, con trỏ trỏ đến hàng đầu tiên.

**ResultSet executeQuery(String sql)**

được sử dụng để thực hiện truy vấn SELECT. Nó trả về đối tượng của ResultSet.

createStatement

Tạo một đối tượng Statement được sử dụng để thực thi các câu truy vấn SQL.

Ngoài cơ sở dữ liệu thì còn có 1 cách khác để lưu giá trị đó là dùng file json

Đầu tiên thì chúng ta phải hiểu json là gì:

JSON là chữ viết tắt của Javascript Object Notation, đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được, bạn có thể sử dụng lưu nó vào một file, một record trong CSDL rất dễ dàng. JSON có định dạng đơn giản, dễ dàng sử dụng và truy vấn hơn XML rất nhiều nên tính ứng dụng của nó hiện nay rất là phổ biến.

Trước khi làm việc với file json thì chúng ta cần tải thư viện json.simple sau đó thêm thư vien json.simple vào trong netbeans

Đầu tiên khi làm việc với JSON thì chúng ta cần phải tạo 1 JSONObject

Với câu lệnh: JSONObject obj = new JSONObject();

Sau đó dùng câu lệnh .put để đưa các cặp giá trị vào:

Vidu:

obj.put("age", 19);

để tạo mảng json thì dùng câu lệnh:

JSONArray list = new JSONArray();

Rồi dùng hàm .add để thêm các già trị vào mảng và dung ham .put để đưa vào object của bạn

Ví dụ

list.add("msg 1");

list.add("msg 2");

list.add("msg 3");

obj.put("messages", list);

để đưa các giá trị của json vào file ta dùng câu lệnh như sau:

try (FileWriter file = new FileWriter("đường dẩn tạo file json"))

{

file.write(obj.toJSONString());

file.flush();

} catch (IOException e) {

e.printStackTrace();

}

Hoặc là có thể tự tạo file json trong máy rồi gỏ các giá trị vào cách này có thể nhanh hơn nhương không thể đưa các giá trj có dấu vào file json

Sau khi đã viết được file json thì chúng ta sẽ đọc file json

Đầu tiên gọi lớp JSONParser để phân tích file json

Câu lệnh

JSONParser parser = new JSONParser();

Sau đó gọi lớp object trong java:

(Object là lớp cha của tất cả các lớp trong Java. Nói cách khác, nó là lớp cao nhất của Java. Lớp Object là khá lợi ích nếu bạn muốn tham chiếu bất cứ đối tượng nào có kiểu mà bạn không biết.)

Object obj = parser.parse(new FileReader("đường dẩn đến file json"));

Tiếp theo gọi lớp JSONObject

Và bắt đầu đọc các values từ key có trong file json bang hàm .get

Ví dụ

Object obj = parser.parse(new FileReader("D:\\test.json"));

JSONObject jsonObject = (JSONObject) obj;

String name = (String) jsonObject.get("name"); // lấy values có key bằng name

System.out.println(name); // in ra gia trị có key bằng name