#### **BÀI GIẢNG**





**CHƯƠNG 11:** 

LẬP TRÌNH TRUY XUẤT CƠ SỞ DỮ LIỆU (JDBC)

TS. Nguyễn Sĩ Thìn (nsthin@vku.udn.vn)

# Chương 11. Lập trình truy xuất cơ sở dữ liệu JDBC



#### Mục tiêu

Hiểu mục đích, ý nghĩa JDBC trong lập trình ứng dụng kết nối cơ sở dữ liệu

Hiểu các khái niệm cơ bản: JDBC driver, JDBC URL,...







#### Nội dung



Chương 11. Lập trình truy xuất cơ sở dữ liệu

- 1. Giới thiệu
- 2. Một số khái niệm cơ bản
- 3. Các bước kết nối cơ sở dữ liệu



#### 1. Giới thiệu



■ JDBC: cung cấp các API để kết nối cơ sở dữ liệu trong các chương trình Java.

#### Lưu ý:

- Hỗ trợ kết nối với nhiều Hệ quản trị CSDL khác nhau : SQL Server, MySQL, Oracle,...
- Nằm trong gói java.sql gồm :

#### Interface

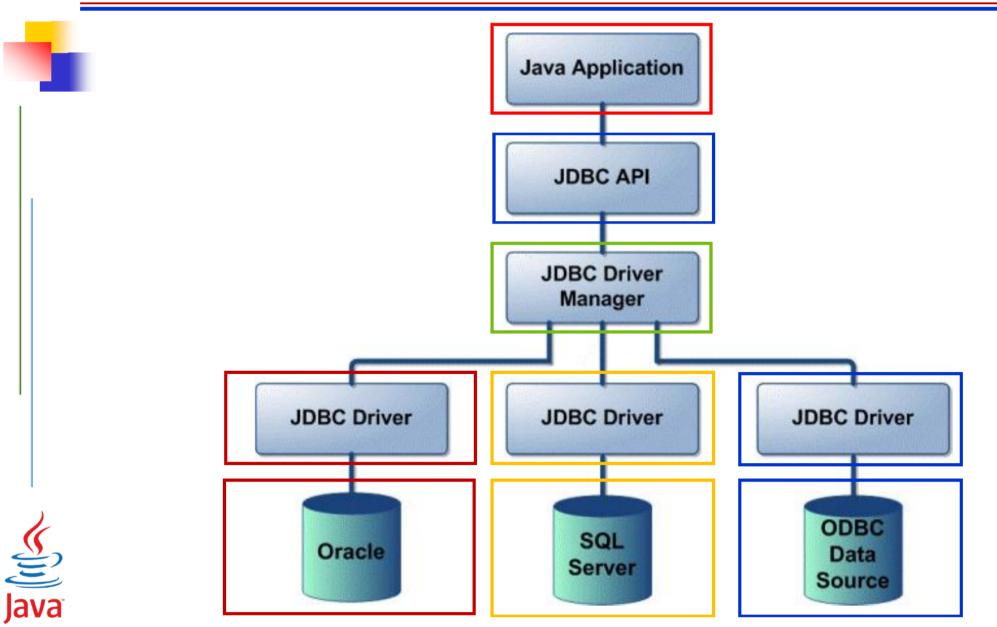
- Driver interface
- Connection interface
- Statement interface
- PreparedStatement interface
- CallableStatement interface
- ResultSet interface
- ResultSetMetaData interface
- DatabaseMetaData interface
- RowSet interface

#### **Class**

- DriverManager class
- Blob class
- Clob class
- Types class



# Kiến trúc JDBC





#### 2. Một số khái niệm cơ bản



- JDBC Driver: xử lý giao tiếp giữa ứng dụng và Cơ sở dữ liệu
- Driver Manager: quản lý driver của Cơ sở dữ liệu
- SQLException:
  - Mỗi SQLException có nhiều SQLException khác được gọi thông qua phương thức "getNextException()"
  - Từ Java SE6, có thể dùng interface "Iterable<Throwable>":

```
for (Throwable t : sqlException)
{
     // do something with t;
}
```



## 3. Các bước kết nối cơ sở dữ liệu







#### Bước 1. Nạp Driver



Database	JDBC Driver Provider	JAR file name
MySQL	Oracle Corporation	mysql-connector-java VERSION.jar
SQL Server	Microsoft Corporation	sqljdbc41.jar, sqljdbc42.jar
Oracle	Oracle Corporation	ojdbc6.jar, ojdbc7.jar, ojdbc8.jar
PostgreSQL	The PostgreSQL Global Development Group	postgresql-VERSION.jar
<b>Apache Derby</b>	Apache Software Foundation	derby.jar, derbyclient.jar
SQLite	Xerial.org	sqlite-jdbc-VERSION.jar
Microsoft Access	UCanAccess.com	ucanaccess-VERSION.jar



 Cài đặt Driver: Class.forName("com.mysql.jdbc.Driver");

# Bước 2. Thiết lập kết nối



#### Đối tượng Connection:

Connection conn = DriverManager.getConnection (url, username, password);

Url : jdbc:mysql://localhost:3306/DB\_Name

Username: root

Password: tùy vào CSDL lúc thiết lập

#### Ví dụ:

Connection conn = DriverManager.getConnection

("jdbc:mysql://localhost:3306/DB\_Name","root","\*\*\*\*\*");



# Bước 3. Thực thi truy vấn



- Statement: cung cấp các phương thức thực thi các câu lệnh SQL
- Khởi tạo:
  - Connection conn = DriverManager.getConnection ( url, username, password );
  - Statement st = conn.createStatement();
- Phương thức:
  - public ResultSet executeQuery (String sql): thực thi câu truy vấn SELECT
  - public int executeUpdate (String sql): thực thi câu truy vấn như create, drop, insert, update, delete ...
  - public boolean execute(String sql): thực thi các câu truy vấn trả về nhiều kết quả
  - public int[] executeBatch(): thực thi nhóm các lệnh



# Bước 3. Thực thi truy vấn (tt)



- PreparedStatement: cung cấp các phương thức thực thi các câu lệnh SQL có tham số
- Khởi tạo:
  - Connection conn = DriverManager.getConnection ( url, username, password );
  - PreparedStatement pst = conn.createPreparedStatement();

Phương thức	Mô tả
public void setInt(int paramIndex, int value)	Thiết lập giá trị integer cho chỉ số tham số cho trước
public void setString(int paramIndex, String value)	Thiết lập giá trị chuỗi cho chỉ số tham số cho trước
public void setFloat(int paramIndex, float value)	Thiết lập giá trị float cho chỉ số tham số cho trước
public void setDouble(int paramIndex, double value)	Thiết lập giá trị double cho chỉ số tham số cho trước
public int executeUpdate()	Thực thi các truy vấn như create, drop, insert, update, delete
public ResultSet executeQuery()	Thực thi truy vấn Select



# Bước 4. Xử lý kết quả trả về



- ResultSet: chứa tất cả các dòng dữ liệu trả về
- Khởi tạo:
  - Statement stmt = con.createStatement();
  - Statement stmt = con.createStatement(int RS.TYPE, int RS.Currency);
  - RS.TYPE: ResultSet. TYPE\_SCROLL\_INSENSITIVE hoặc ResultSet. TYPE\_SCROLL\_SENSITIVE
  - ResultSet rs=stmt.executeQuery();

Phương thức	Mô tả
public boolean next():	Di chuyển con trỏ đến hàng tiếp theo
public boolean previous():	Di chuyển con trỏ về hàng phía trước
public boolean first():	Di chuyển con trỏ về hang đầu tiên
public boolean last():	Di chuyển con trỏ về hàng cuối cùng
public boolean absolute(int row):	Di chuyển con trỏ về hàng cho trước
public boolean relative(int row):	Di chuyển con trỏ về hàng cho trước (tham số row có thể dương hoặc âm)



# Bước 4. Xử lý kết quả trả về (tt)



 ResultSetMetaData: tương tự như ResultSet nhưng có thể lấy thêm nhiều thông tin hơn về kết quả trả về.

#### Khởi tạo:

- Statement stmt = con.createStatement();
- Statement stmt = con.createStatement(int RS.TYPE, int RS.Currency);
  - RS.TYPE: ResultSet. TYPE\_SCROLL\_INSENSITIVE hoặc
     ResultSet. TYPE\_SCROLL\_SENSITIVE
- ResultSet rs=stmt.executeQuery();
- ResultSetMetaData rsmd = rs.getMetaData();

Phương thức	Mô tả
public int getColumnCount()	Trà về tổng số cột của kết quả
public String getColumnName(int index)	Trả về tên cột với chỉ số cho trước
public String getColumnTypeName(int index)	Trả về kiểu giá trị của cột với chỉ số cho trước
public String getTableName(int index)	Trả về tên bảng với chỉ số cột cho trước



## Bài thực hành cuối chương 11



#### ĐÈ BÀI

Tạo cơ sở dữ liệu trên HQTCSDL (MySQL hoặc SQL Server...) chứa bảng "students" gồm các cột: ID, Name, Java, GhiChu và tạo sẵn 3 dòng thông tin. Viết chương trình quản lý sinh viên sử dụng GUI (kết nối CSDL trên) gồm các chức năng cơ bản:

- Xem danh sách sinh viên
- Thêm, sửa, xóa sinh viên khỏi danh sách thông qua các nút: "INSERT", "EDIT", "DELETE" tương ứng.



# THANK YOU!