

Phát triển UD CSDL 2

Chương 3:

Lập trình Java kết nối với CSDL bằng JDBC

GVLT: Nguyễn Trường Sơn





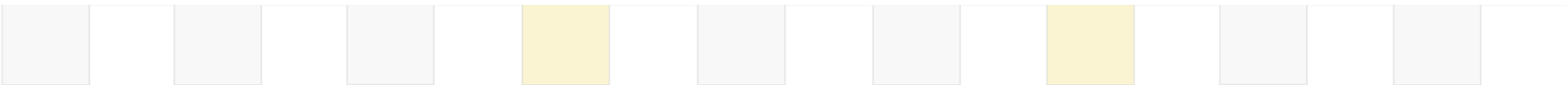
JAVA DATABASE CONNECTIVITY

- Giới thiệu JDBC
- Một số Class và Interface của JDBC API thường dùng
- Các bước thực hiện truy cập CSDL sử dụng JDBC
 - *Bước 1*: Đăng ký driver
 - *Bước 2*: Tạo đối tượng kết nối đến CSDL
 - *Bước 3*: Tạo đối tượng để thực hiện các lệnh thao tác lên CSDL & Xây dựng câu lệnh tương ứng (đọc, thêm, xóa, sửa).
 - *Bước 4*: Thực hiện lệnh
 - *Bước 5*: Xử lý kết quả trả về (optional)
 - *Bước 6*: Đóng kết nối
- ExecuteBatch / Xử lý lỗi và ngoại lệ / Transaction
- Store Procedure



JDBC

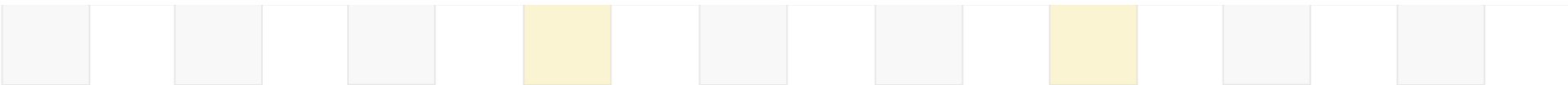
- JDBC (Java Database Connectivity) cung cấp cho java developer tập các interface chuẩn dùng để truy xuất dữ liệu quan hệ.
- JDBC được phát triển bởi JavaSoft.



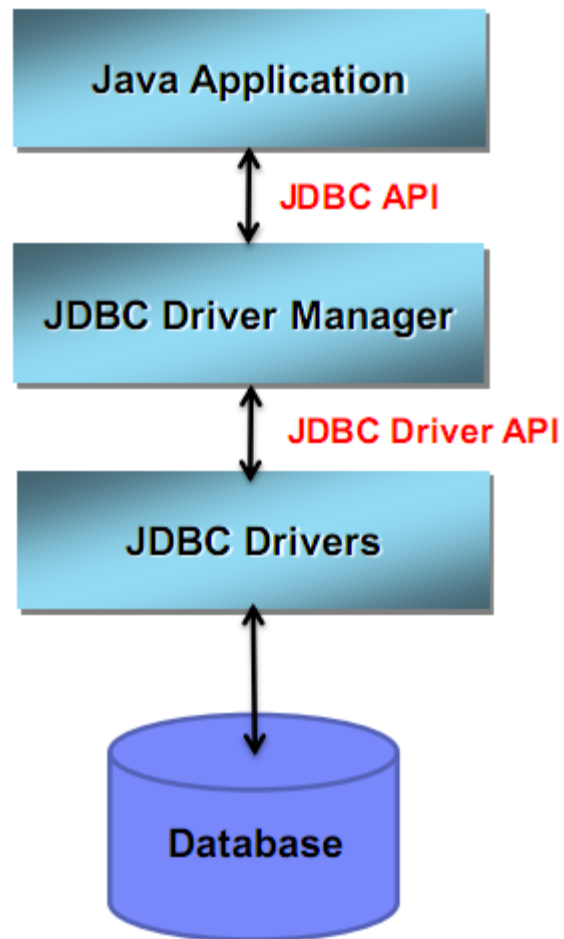


Kiến trúc JDBC

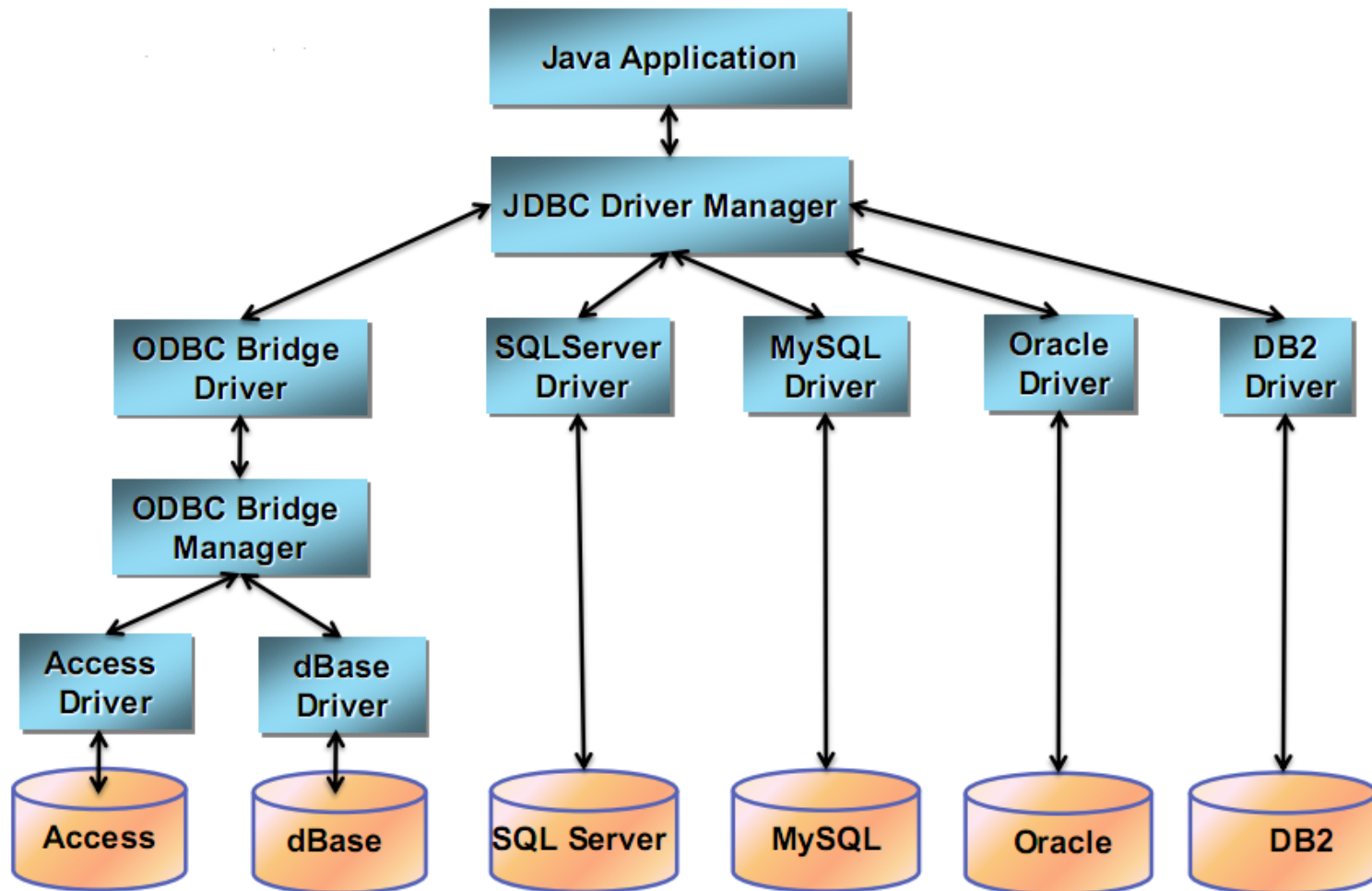
- Khi làm việc với JDBC, java developer sẽ làm việc như nhau đối với các hệ cơ sở dữ liệu khác nhau.
- Java developer không phụ thuộc vào một hệ cơ sở dữ liệu cụ thể nào.
- Java developer không cần phải quan tâm đến sự khác nhau khi giao tiếp với các HQTCSDL khác nhau.



Kiến trúc JDBC

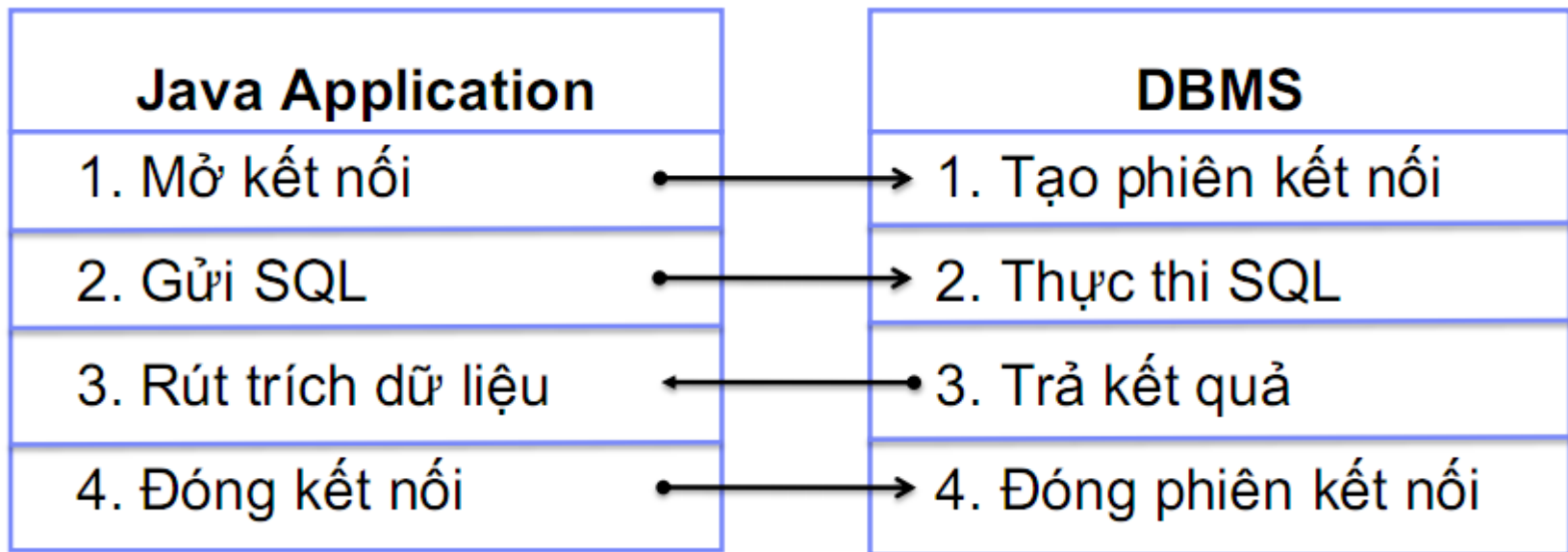


Kiến trúc JDBC





Hoạt động của JDBC





JDBC API

- Class và Interface của JDBC API thuộc gói **java.sql**
- **DriverManager** dùng để nạp các driver và tạo Connection đến cơ sở dữ liệu.
- **Driver**: Driver của cơ sở dữ liệu, mỗi JDBC Driver đều cài đặt lại Interface này.
- **Connection** :Thiết lập một Connection đến cơ sở dữ liệu và cho phép tạo các Statement .
- **Statement**: Gắn kết với một connection đến cơ sở dữ liệu và cho phép thực thi các câu lệnh SQL.
- **CallableStatement** tương tự Statement nhưng áp dụng cho Store procedures.

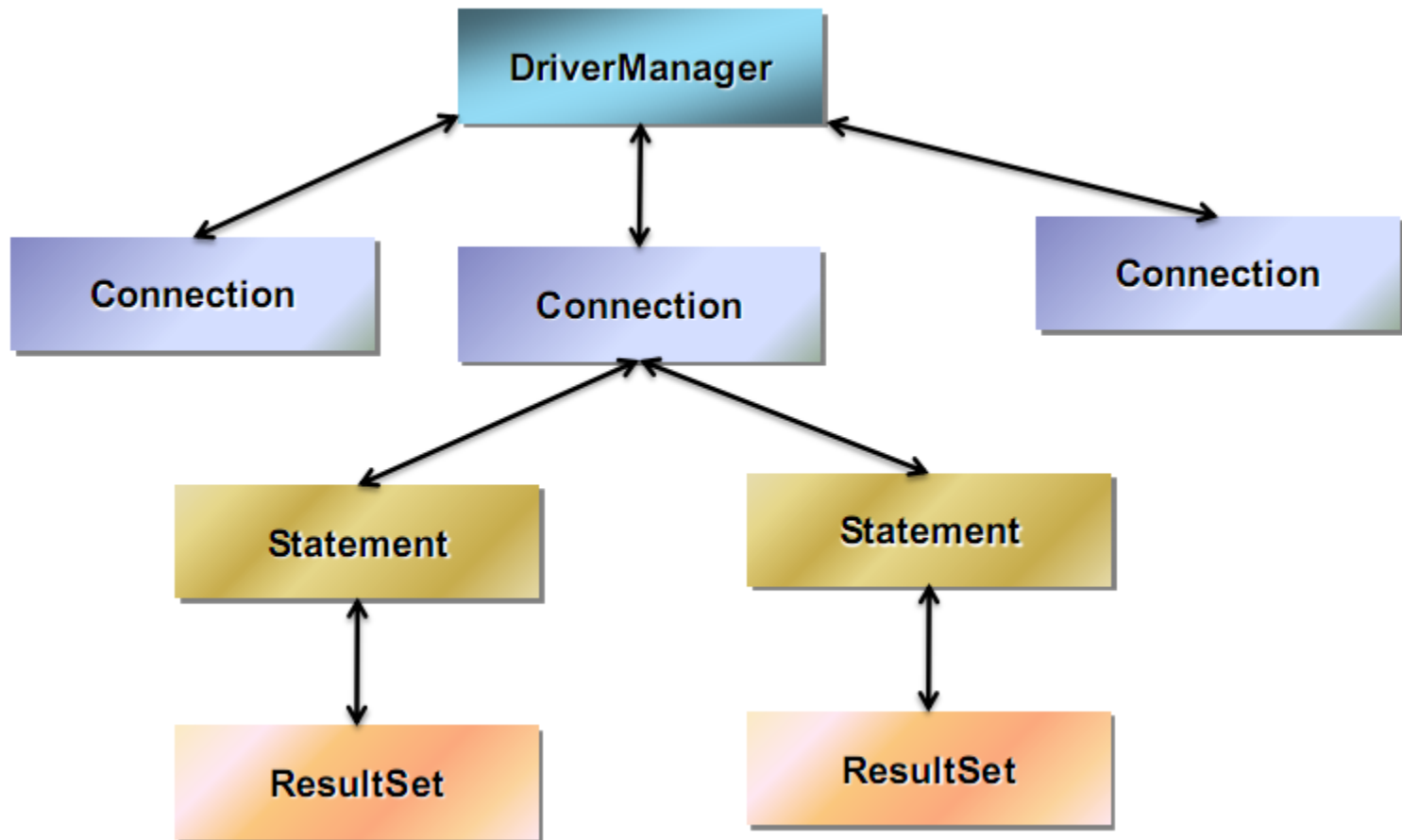


JDBC API

- **PreparedStatement**: Tương tự như Statement nhưng áp dụng cho Precompiled SQL.
- **ResultSet**: Cung cấp thông tin rút trích từ cơ sở dữ liệu , cho phép truy xuất các dòng dữ liệu.
- **ResultSetMetaData**: Cung cấp các thông tin như kiểu dữ liệu và các thuộc tính trong Resultset.
- **DatabaseMetaData**: Cung cấp các thông tin của cơ sở dữ liệu kết nối.
- **SQLException**: Cung cấp thông tin các ngoại lệ xảy ra khi tương tác với cơ sở dữ liệu.



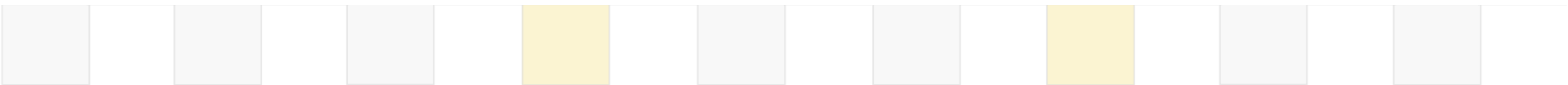
JDBC API





Một số bước kỹ thuật cơ bản sử dụng JDBC

- *Bước 1*: Đăng ký driver
- *Bước 2*: Tạo đối tượng kết nối đến CSDL
- *Bước 3*: Tạo đối tượng để thực hiện các lệnh thao tác lên CSDL & Xây dựng câu lệnh tương ứng (đọc, thêm, xóa, sửa).
- *Bước 4*: Thực hiện lệnh
- *Bước 5*: Xử lý kết quả trả về (optional)
- *Bước 6*: Đóng kết nối



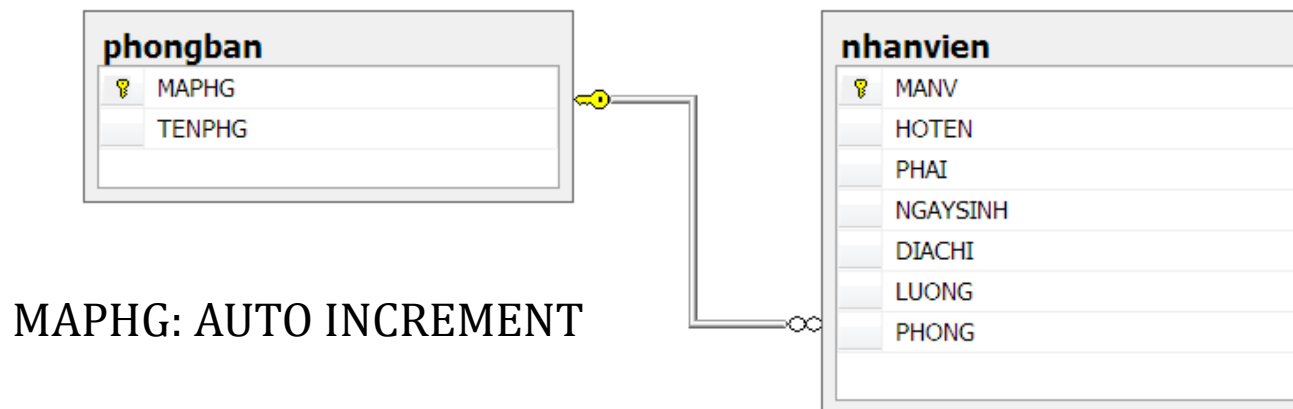


Bước 1: Đăng ký driver JDBC

- Chuẩn bị driver tương ứng với HQTCSĐL & thực hiện đăng ký.
- Đăng ký driver MySQL:
 - Chuẩn bị driver: download mysql-connector-java-5.1.22.zip
 - Đăng ký:
 - Cách 1: Sử dụng Class.forName
`Class.forName("org.gjt.mm.mysql");`
 - Cách 2: DriverManager.registerDriver
`Driver driver = new org.gjt.mm.mysql.Driver();`
`DriverManager.registerDriver(driver);`
 - Các package cần sử dụng:
`import java.sql.Connection;`
`import java.sql.Driver;`
`import java.sql.DriverManager;`

Bước 2: Tạo & mở kết nối đến CSDL

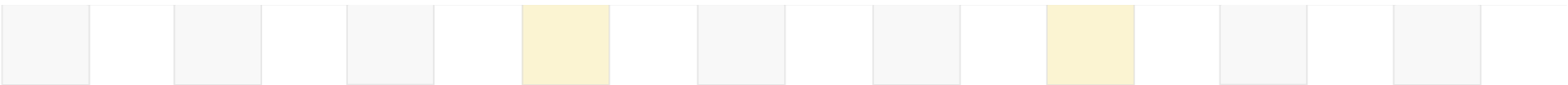
- CSDL minh họa:
 - Name: **NhanVienDB**
 - MySQL DBMS:
 - Username: **root**
 - Password: **root**
 - Port: **3306**
 - Domain: **localhost**



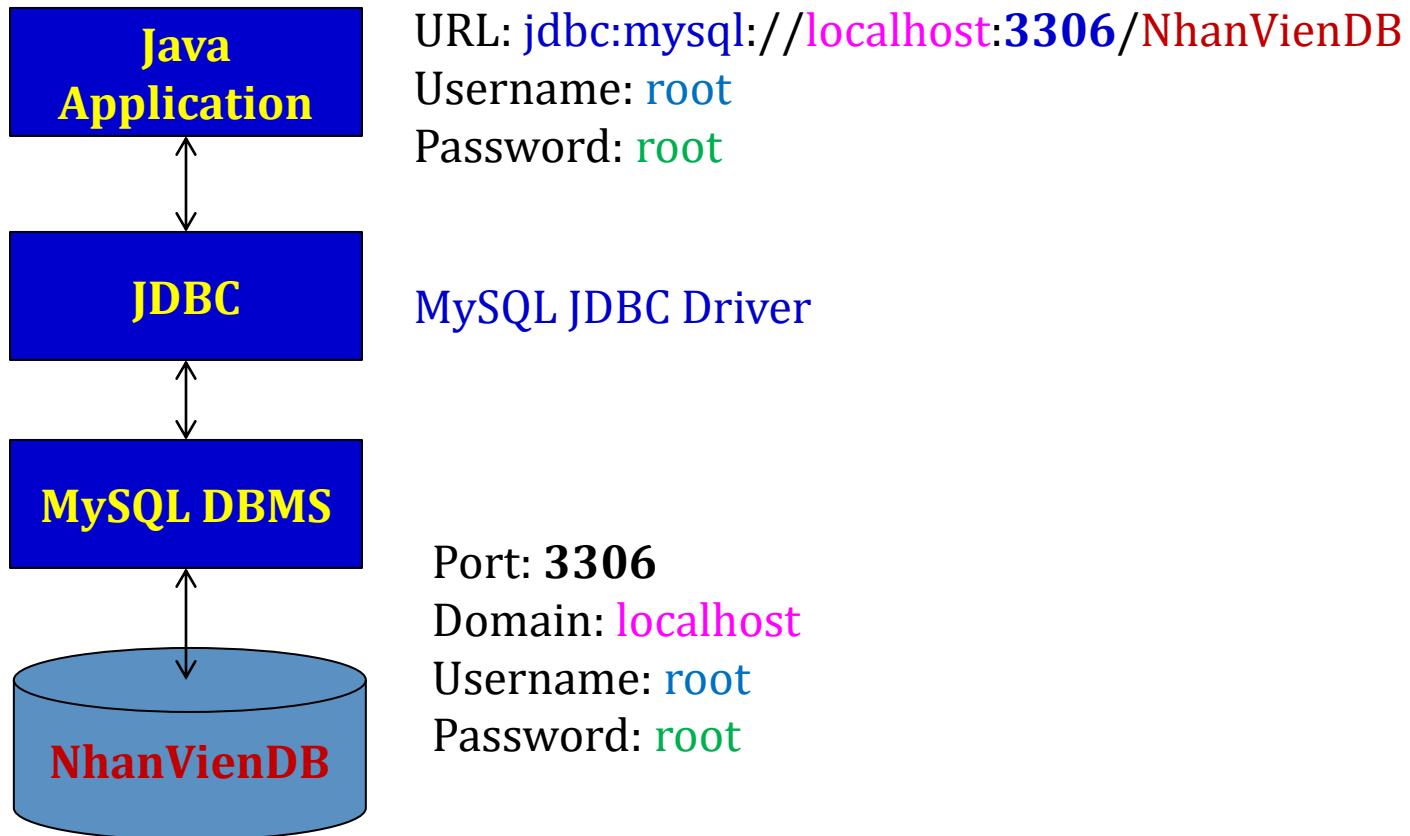


Bước 2: Tạo & mở kết nối đến CSDL

- Sử dụng phương thức **getConnection** của DriverManager
- Một số cú pháp:
 - Connection **getConnection** (String **url**);
 - Connection **getConnection** (String **url**, String **user**, String **password**);
 - Connection **getConnection** (String **url**, java.util.Properties **info**);



Bước 2: Tạo & mở kết nối đến CSDL



Bước 1 & 2

```
1 package demojdbc;
2 import java.sql.*;
3 import java.util.Properties;
4 public class DemoJDBC {
5     public static void main(String[] args){
6         ...
7         //1. Đăng ký driver và tạo kết nối đến CSDL
8         Driver driver = new org.gjt.mm.mysql.Driver();
9         DriverManager.registerDriver(driver);
10        //2. Tạo kết nối đến CSDL
11        String conString = "jdbc:mysql://localhost:3306/NhanVienDB";
12        Properties info = new Properties();
13        info.setProperty("characterEncoding", "utf8");
14        info.setProperty("user", "root");
15        info.setProperty("password", "");
16        Connection connection;
17        connection = DriverManager.getConnection(conString, info);
18        //3. Tạo đối tượng Statement để thực hiện thao tác dữ liệu mong muốn
19        ...
20    }
21 }
22 }
23 }
24 }
```

1

2




Bước 3: Tạo đối tượng để thực thi và chuẩn bị các lệnh SQL

- Sử dụng phương thức **createStatement()** của đối tượng kết nối:
- Chuẩn bị các lệnh SQL:
 - SELECT** : Truy vấn dữ liệu từ bảng
 - DELETE** : Xóa dữ liệu từ bảng
 - UPDATE** : Cập nhật dữ liệu vào bảng
 - INSERT** : Thêm dữ liệu vào bảng



Query (Đọc)



Update (Thay đổi)

- Ví dụ:

```
Statement statement = connection.createStatement();
```

```
String sql = "SELECT * FROM NHANVIEN";
```



Bước 3: Tạo đối tượng để thực thi và chuẩn bị các lệnh SQL

```
1 package demojdbc;
2 import java.sql.*;
3 import java.util.Properties;
4 public class DemoJDBC {
5     public static void main(String[] args){
6         ...
7         //1. Đăng ký driver và tạo kết nối đến CSDL
8         //2. Tạo kết nối đến CSDL
9         //3. Tạo đối tượng Statement để thực hiện thao tác dữ liệu mong muốn
10        Statement statement = connection.createStatement();
11        String sql = "SELECT * FROM NHANVIEN";
12        //4. Thực hiện lệnh SQL
13        //5. Xử lý kết quả trả về
14        ...
15    }
16 }
17 }
18 }
```

3



Bước 4: Thực thi SQL

- Sử dụng một số cú pháp:

//Thực thi câu truy vấn (SELECT). Kết quả trả về là một tập dữ liệu
ResultSet

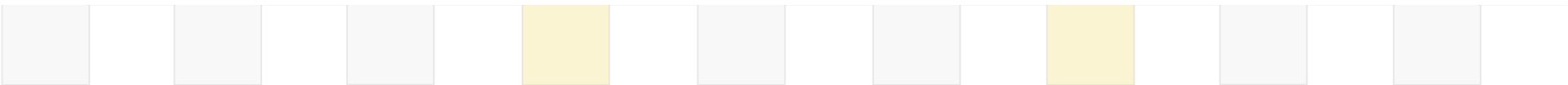
ResultSet statement.*executeQuery* (*String* sql)

//Thực thi câu lệnh cập nhật dữ liệu (Thêm, xóa, sửa)

int statement.*executeUpdate* (*String* sql)

//Thực thi câu lệnh SQL nói chung (thêm, xóa, sửa, truy vấn, ...)

boolean statement.*execute* (*String* sql)





Bước 4: Thực thi SQL

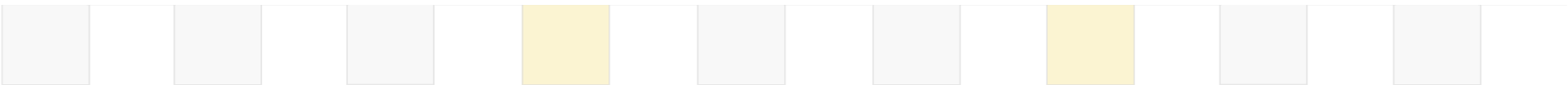
```
1 package demojdbc;
2 import java.sql.*;
3 import java.util.Properties;
4 public class DemoJDBC {
5     public static void main(String[] args){
6         ...
7         //1. Đăng ký driver và tạo kết nối đến CSDL
8         //2. Tạo kết nối đến CSDL
9         //3. Tạo đối tượng Statement để thực hiện thao tác dữ liệu mong muốn
10        Statement statement = connection.createStatement();
11        String sql = "SELECT * FROM NHANVIEN";
12        //4. Thực hiện truy xuất (đọc / ghi)
13        ResultSet rs = statement.executeQuery(sql);
14        //5. Xử lý kết quả trả về
15        ...
16    }
17 }
18 }
```

4



Bước 5: Xử lý kết quả trả về

- Tùy vào kết quả trả về của việc thực thi câu lệnh:
 - *ResultSet* statement.*executeQuery* (*String* sql)
→ Kết quả trả về là một tập dữ liệu của câu truy vấn: Cần duyệt qua tập dữ liệu này để xử lý (hiển thị, ...)
 - *int* statement.*executeUpdate* (*String* sql)
→ Kết quả trả về là số dòng bị ảnh hưởng của việc thêm, xóa, sửa.
 - *boolean* statement.*execute* (*String* sql)
→ Tùy vào câu lệnh SQL sẽ có cách thức xử lý tương ứng



5a. Lấy dữ liệu từ ResultSet

- Lấy kết quả từ câu truy vấn:

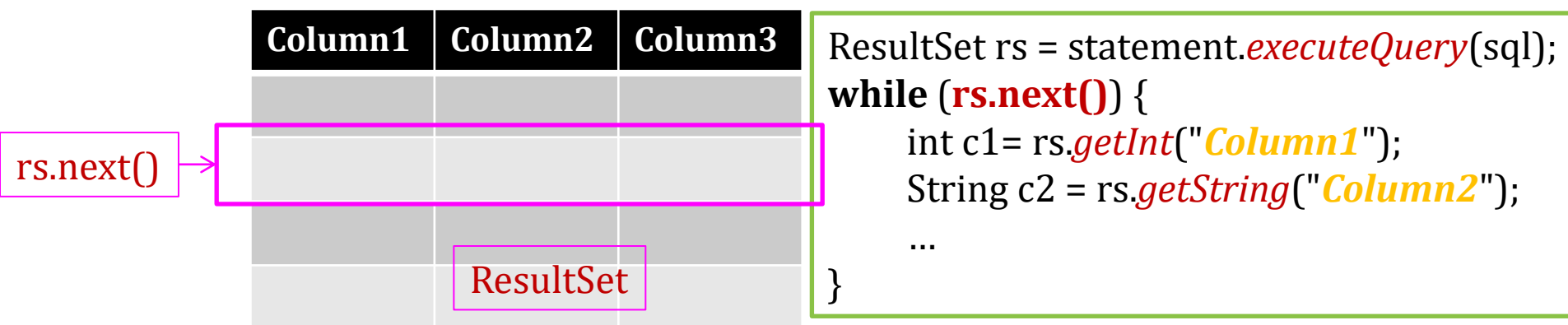
```
ResultSet rs = statement.executeQuery(sql)
```

Hoặc

```
statement.execute (sql)
```

```
ResultSet rs = statement.getResultSet()
```

- Sử dụng *.next()* để đến dòng kế tiếp.





5a. Lấy dữ liệu từ bảng PHONGBAN

```
1  ...
2  Statement statement = connection.createStatement();
3  String sql = "SELECT * FROM PHONGBAN";
4  ResultSet rs = statement.executeQuery(sql);
5
6  while (rs.next()){
7      int manv= rs.getInt("MAPHG");
8      String hoten = rs.getString("TENPHG");
9      System.out.print(manv+" ");
10     System.out.println(hoten);
11 }
12 connection.close();
13
14 ...
```

5



5a. Lấy dữ liệu từ bảng NHANVIEN

```
1  ...
2  Statement statement = connection.createStatement();
3  String sql = "select * from NHANVIEN Where PHONG = 1";
4  statement.execute (sql);
5  ResultSet rs = statement.getResultSet();
6
7  while (rs.next()){
8      int manv = rs.getString("MANV");
9      String hoten = rs.getString("HOTEN");
10     Date ngaysinh = rs.getDate("NGAYSINH");
11     float luong = rs.getFloat("LUONG");
12     System.out.println(...);
13 }
14 connection.close();
15
16 ...
17
```




5b. Xóa dữ liệu bảng PHONGBAN

```
1  ...
2  Statement statement = connection.createStatement();
3  String sql = "delete from PHONGBAN where MAPHG = 1";
4  int n = statement.executeUpdate(sql);
5  if (n == 1){
6      System.out.println("Xóa thành công !!!");
7  } else {
8      System.out.println("Lỗi !!!");
9  }
10 connection.close();
11 ...
```

5



5b. Xóa dữ liệu bảng NHANVIEN

```
1  ...
2  Statement statement = connection.createStatement();
3  String sql = "delete from NHANVIEN where MAPHG = 1";
4  int n = statement.executeUpdate(sql);
5  if (n >= 1){
6      System.out.println("Xóa thành công !!!");
7  } else {
8      System.out.println("Lỗi !!!");
9  }
10 connection.close();
11 ...
```

5



5b. Sửa dữ liệu bảng PHONGBAN

```
1  ...
2  Statement statement = connection.createStatement();
3  String sql = String.format ("update PHONGBAN
4                               set TENPHONG = '%s'
5                               where MAPHG = %d" , tenphong, maphong);
6  int n = statement.executeUpdate(sql);
7  if (n == 1){
8      System.out.println("Xóa thành công !!!");
9  } else {
10     System.out.println("Lỗi !!!");
11 }
12 connection.close();
13 ...
```

5b. Sửa dữ liệu bảng NHANVIEN

```
1  ...
2  Statement statement = connection.createStatement();
3  String sql = String.format("update NHANVIEN
4                               set HOTEN= '%s',
5                               PHAI = '%s',
6                               LUONG= %f
7                               where MANV = %d" , hoten, phai, luong,
8                               manv);
9  int n = statement.executeUpdate(sql);
10 if (n == 1){
11     System.out.println("Xóa thành công !!!");
12 } else {
13     System.out.println("Lỗi !!!");
14 }
15 connection.close();
```



5b. Thêm dữ liệu bảng NHANVIEN

```
1  ...
2  String manv = ...
3  String hoten = ...
4  float luong = ...
5  Statement statement = connection.createStatement();
6  String sql = String.format ("INSERT INTO NHANVIEN (MANV, HOTEN,
7  LUONG, ...) VALUES ('%s', '%s', %f, ...)", manv, hoten, luong, ... )
8  int n = statement.executeUpdate(sql);
9  if (n == 1){
10     System.out.println(" Thêm thành công !!!");
11 } else {
12     System.out.println("Lỗi !!!");
13 }
14 connection.close();
15 ...
```

5b. Thêm dữ liệu bảng PHONGBAN

```
1  ...
2  String tenphong = ...
3  Statement statement = connection.createStatement();
4  String sql = String.format ("INSERT INTO PHONGBAN (TENPHONG) VALUES ('%s'
5  ", tenphong )
6  int n = statement.executeUpdate(sql, Statement.RETURN_GENERATED_KEYS );
7  if (n > 0){
8      System.out.println(" Thêm thành công !!!");
9      ResultSet rs=statement.getGeneratedKeys();
10     if(rs.next()){
11         int maDanhMuc = rs.getInt(1);
12         System.out.println("Mã danh mục vừa thêm: "+maDanhMuc);
13     }
14 } else {
15     System.out.println("Lỗi !!!");
16 }
17 connection.close();
18 ...
```

5

Khóa tự
động tăng

5b. Thêm dữ liệu bảng PHONGBAN

```
1 Statement statement = connection.createStatement();
2 String[] arrPhong = {"PB1", "PB2", "PB3"};
3 String sql = "INSERT INTO PHONGBAN (TENPHG) VALUES ";
4 for (int i=0; i < arrPhong.length; i++){
5     sql = sql + String.format("('%s')", arrPhong[i]);
6     if( i < arrPhong.length-1){ sql = sql + ", "; }
7 }
8     sql: INSERT INTO PHONGBAN (TENPHG)
9         VALUES ('PB1'), ('PB2'), ('PB3')
10
11 int n = statement.executeUpdate(sql, Statement.RETURN_GENERATED_KEYS);
12 if (n > 0) {
13     System.out.println("Thêm thành công !!!");
14     ResultSet rs = statement.getGeneratedKeys();
15     while (rs.next()){
16         int maphong = rs.getInt(1);
17         System.out.println("Mã phòng ban vừa thêm: " + maphong);
18     }
19 }
20 connection.close();
```

5

Khóa tự
động tăng



6. Đóng kết nối

- Sử dụng `.close()` của đối tượng `Connection`

```
1 package demojdbc;
2 import java.sql.*;
3 import java.util.Properties;
4 public class DemoJDBC {
5     public static void main(String[] args){
6         ...
7         //1. Đăng ký driver và tạo kết nối đến CSDL
8         //2. Tạo kết nối đến CSDL
9         ...
10        //6. Đóng kết nối
11        connection.close();
12        ...
13    }
14 }
```




executeBatch

```
1      ...
2      String sql1="INSERT INTO DANHMUC(TENDANHMUC) VALUES('Java')";
3      String sql2="INSERT INTO DANHMUC(TENDANHMUC) VALUES('PHP')";
4      String sql3="UPDATE DANHMUC SET TENDANHMUC='C' WHERE
5      MADANHMUC=43";
6      Statement statement=connection.createStatement();
7      statement.addBatch(sql1);
8      statement.addBatch(sql2);
9      statement.addBatch(sql3);
10     int [] arrResult = statement.executeBatch();
11     ...
```

arrResult [i] : Số dòng bị ảnh hưởng của câu lệnh SQL thứ **i**



Xử lý ngoại lệ

```
1 package demojdbc;
2 import java.sql.*;
3 import java.util.Properties;
4 public class DemoJDBC {
5     public static void main(String[] args){
6         try {
7             //1. Đăng ký driver và tạo kết nối đến CSDL
8             //2. Tạo kết nối đến CSDL
9             ...
10            //6. Đóng kết nối
11        } catch (SQLException ex) {
12            System.out.println(ex);
13        }
14    }
15 }
```



PreparedStatement

```
1 public static void main(String[] args){
2     try {
3         ...
4         String sql = "INSERT INTO T(X, Y) VALUES ( ?, ?)";
5         PreparedStatement statement = connection.prepareStatement(sql,
6 Statement.RETURN_GENERATED_KEYS);
7         statement.setString(1, "Value Of X");
8         statement.setString(2, "Value Of Y");
9         int n = statement.executeUpdate();
10        ...
11        connection.close();
12    } catch (SQLException ex) {
13        System.out.println(ex);
14    }
15 }
16 }
```



Sử dụng Transaction

```
1  try {
2      String sql1= insert/delete/update . . .
3      String sql2= insert/delete/update . . .
4      Connection con =DriverManager.getConnection( cs, info);
5      try{
6          con.setAutoCommit(false);
7          Statement statement1= con.createStatement();
8          Statement statement2= con.createStatement();
9          statement1.executeUpdate(sql1);
10         statement2.executeUpdate(sql2);
11         con.commit();
12     }catch (SQLException ex){
13         con.rollback();
14     }finally{
15         con.close();
16     }
17 } catch (SQLException ex) { //Xử lý lỗi }
```



Sử dụng Transaction

```
1 String sql1= insert/delete/update . . .
2 String sql2= insert/delete/update . . .
3 Connection con =DriverManager.getConnection( cs, info);
4 try{
5     con.setAutoCommit(false);
6     Statement statement1= con.createStatement();
7     Statement statement2= con.createStatement();
8     statement1.executeUpdate(sql1);
9     statement2.executeUpdate(sql2);
10    if(điều kiện commit){
11        con.commit();
12    }else{
13        con.rollback();
14    }
15 }catch (SQLException ex){
16     //Xử lý lỗi
17 }finally{ con.close(); }
```



Sử dụng Transaction

```
1 String sql1= insert/delete/update . . .
2 String sql2= insert/delete/update . . .
3 Connection con =DriverManager.getConnection( cs, info);
4 try{
5     con.setAutoCommit(false);
6     Statement statement1= con.createStatement();
7     Statement statement2= con.createStatement();
8     Savepoint save1 = con.setSavepoint();
9     statement1.executeUpdate(sql1);
10    statement2.executeUpdate(sql2);
11    if(điều kiện commit){
12        con.commit();
13    }else{
14        con.rollBack(save1);
15    }
16 }catch (SQLException ex){
17     //Xử lý lỗi
18 }finally{ con.close();}
```

Sử dụng Transaction

```
1      ...
2      Date ngaysinh = sdf.parse("1999-08-25");
3      Date d = sdf.parse("2000-1-1");
4      String sql = "INSERT INTO NHANVIEN VALUES ...";
5      Connection connection = null;
6      ...
7      connection = DriverManager.getConnection(conString, info);
8      try {
9          connection.setAutoCommit(false);
10         Statement statement = connection.createStatement();
11         statement.executeUpdate(sql);
12         if ( ngaysinh.after(d)) {
13             connection.rollback();
14             System.out.println("NGAYSINH MUST BE BEFORE 1/1/2000");
15         } else {
16             connection.commit();
17         }
18     } catch (SQLException ex) {
19         System.out.println(ex);
20     } finally {connection.close();}
```



CallableStatement - Xử lý Store Procedure

```
1 DELIMITER $$
2 CREATE PROCEDURE spThemDanhMuc (
3     in tenDanhMuc VARCHAR (45))
4 BEGIN
5     INSERT INTO DANHMUC(TENDANHMUC) VALUES (tenDanhMuc);
6     SELECT LAST_INSERT_ID();
7 END
```

```
1 String tenDanhMuc = "C++";
2 CallableStatement statement =
3 connection.prepareCall("{call spThemDanhMuc(?)}");
4 statement.setString(1, tenDanhMuc);
5 statement.execute();
6 ResultSet rs = statement.getResultSet();
7 if (rs.next()) {
8     int maDanhMuc = rs.getInt(1);
9     System.out.println(maDanhMuc);
10 }
11 }
```


CallableStatement - Xử lý Store Procedure

Start phpInfo XCache phpMyAdmin SQLiteManager FAQ Try MAMP PRO

localhost » QLNV

Structure SQL Search Query Export Import Operations Privileges **Routines**

Routines

Name	Action
spThemPhongBan	Edit Execute Export
spThemPhongBan2	Edit Execute Export

New

Add routine

Add routine

Details

Routine name: spName

Type: PROCEDURE

Direction	Name	Type	Length/Values	Options
IN	param1	INT		Drop
IN	param2	VARCHAR	40	utf8 Drop

Add parameter

Definition

```
1 BEGIN
2     SELECT * FROM nhanvien WHERE MaNV= param1 AND HoTen= param2;
3
4 END
```



CallableStatement - Xử lý Store Procedure

```
1 DELIMITER $$
2 CREATE PROCEDURE spThemDanhMuc (
3     in tenDanhMuc VARCHAR (45),
4     out maDanhMuc int )
5 BEGIN
6     INSERT INTO DANHMUC(TENDANHMUC) VALUES (tenDanhMuc);
7     SET maDanhMuc = LAST_INSERT_ID();
8 END
```

```
1 String tenDanhMuc = "C++";
2 CallableStatement statement =
3 connection.prepareCall("{call spThemDanhMuc (?, ?)}");
4 statement.setString(1, tenDanhMuc);
5 statement.registerOutParameter(2, Types.INTEGER);
6 statement.execute();
7 int maDanhMuc = statement.getInt(2);
8 System.out.println(maDanhMuc);
```

MySQL

- Cài đặt MySQL server

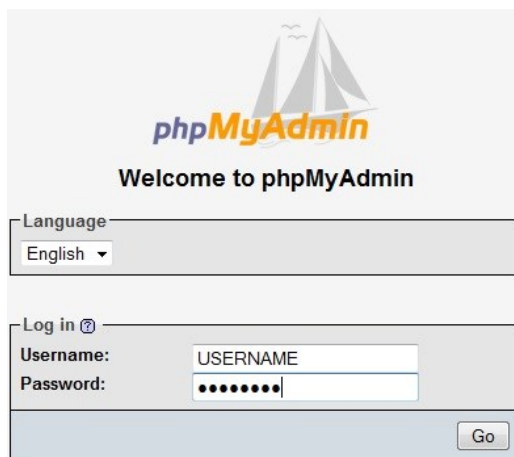
- WAMP



- IDE:

- phpMyAdmin (tích hợp sẵn trong WAMP)

- MySQL Workbench 5.2 CE





Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập slide bài giảng và video môn Lập Trình Java, ĐH KHTN, 2010
- The Java Language Specification Third Edition (2005)

