



**FPT POLYTECHNIC**



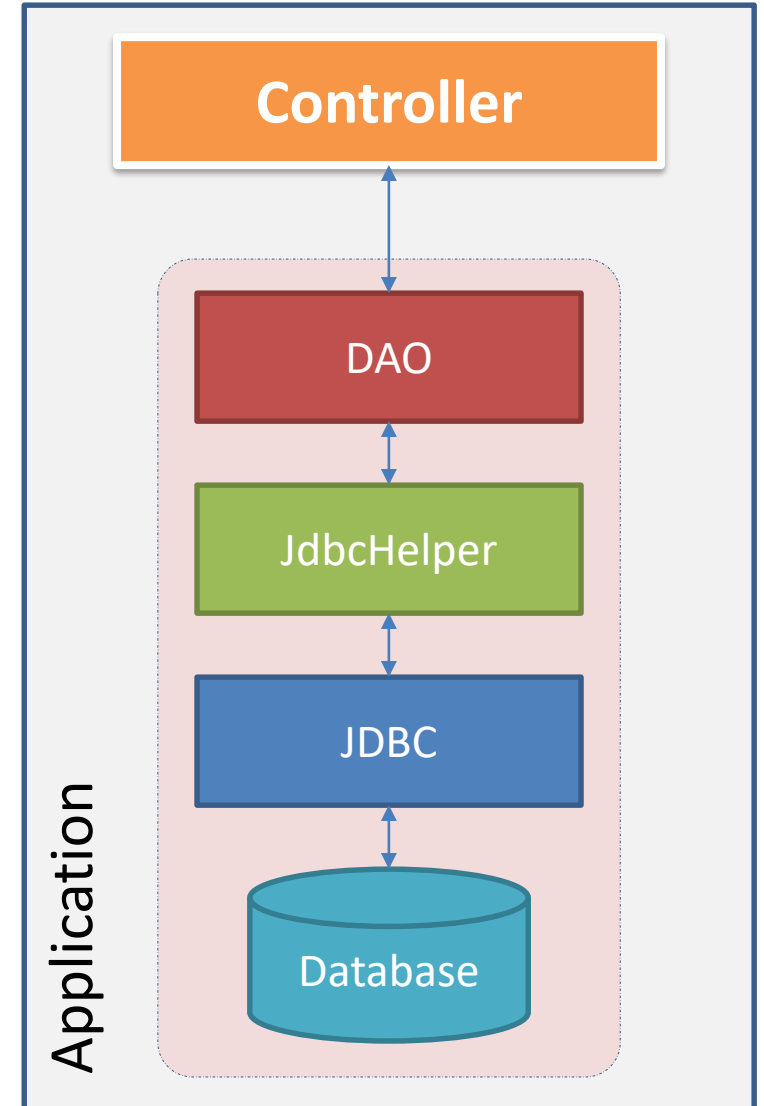
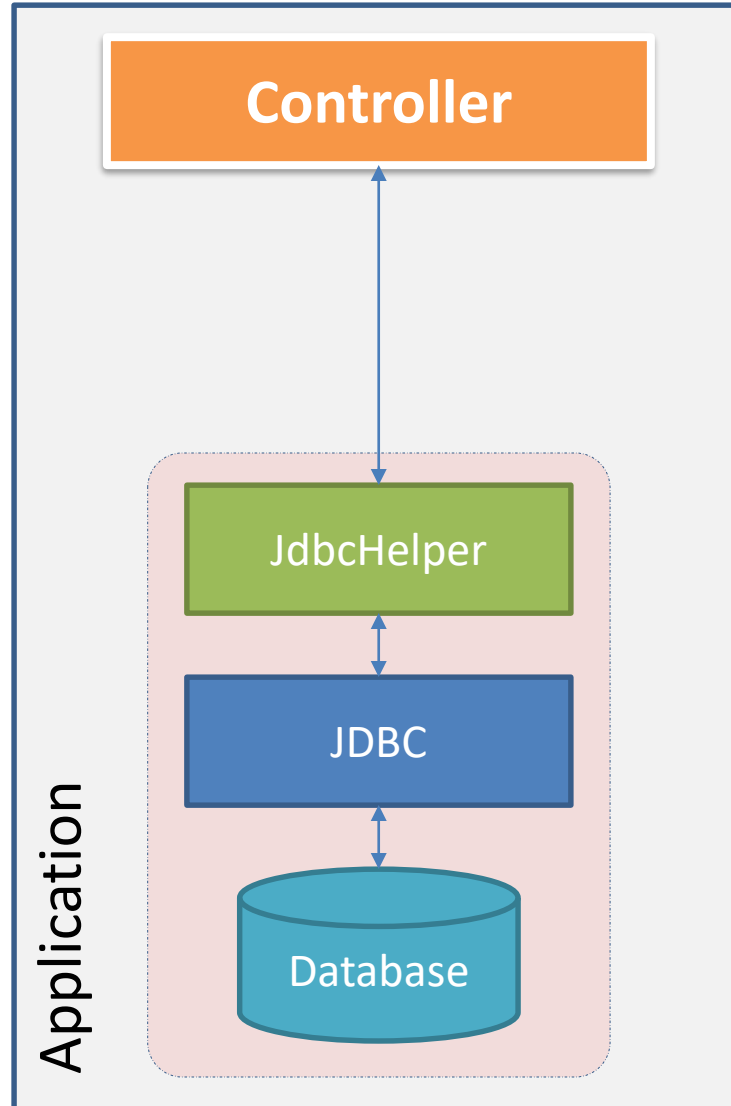
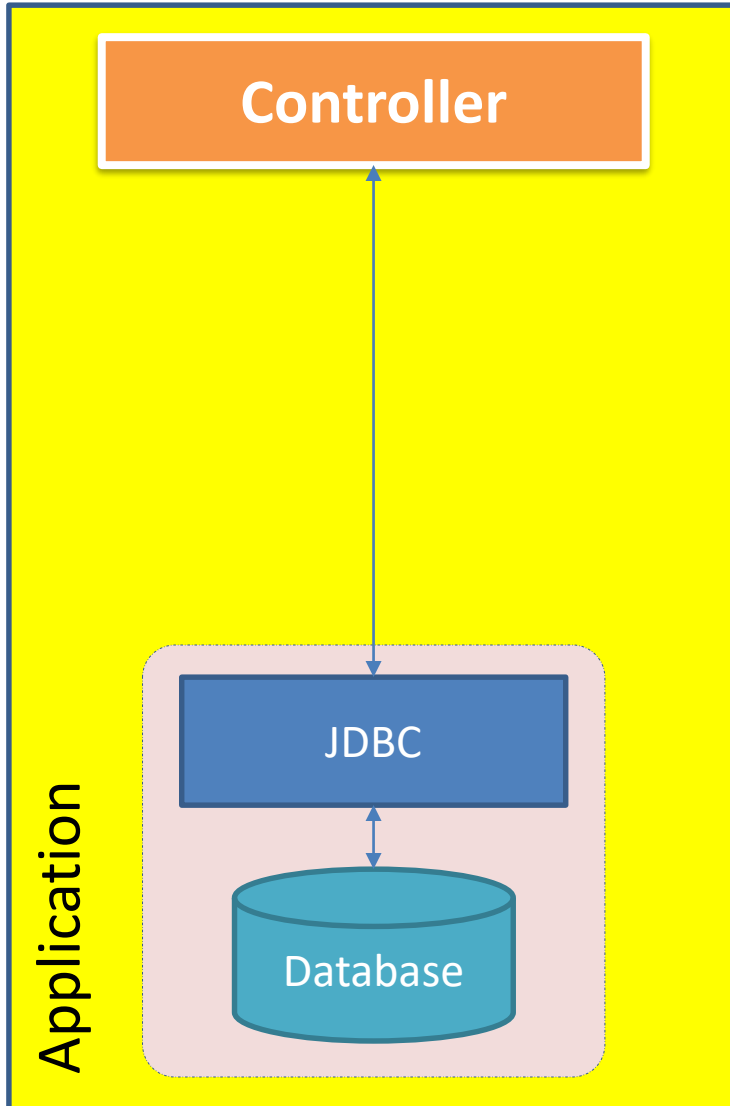
Conceive Design Implement Operate

## LẬP TRÌNH CSDL

GIẢNG VIÊN: NGUYỄN NGHIỆM

[www.poly.edu.vn](http://www.poly.edu.vn)

# MỨC ĐỘ TÁI SỬ DỤNG TRONG LẬP TRÌNH JDBC





**FPT POLYTECHNIC**



Conceive Design Implement Operate

## **LẬP TRÌNH JDBC**

**GIẢNG VIÊN: NGUYỄN NGHIỆM**

[www.poly.edu.vn](http://www.poly.edu.vn)

- ◎ Kết thúc môn học này, bạn có khả năng
  - ◎ Mô hình tổ chức lập trình JDBC
  - ◎ Sử dụng Statement
  - ◎ Sử dụng PreparedStatement
  - ◎ Sử dụng CallableStatement

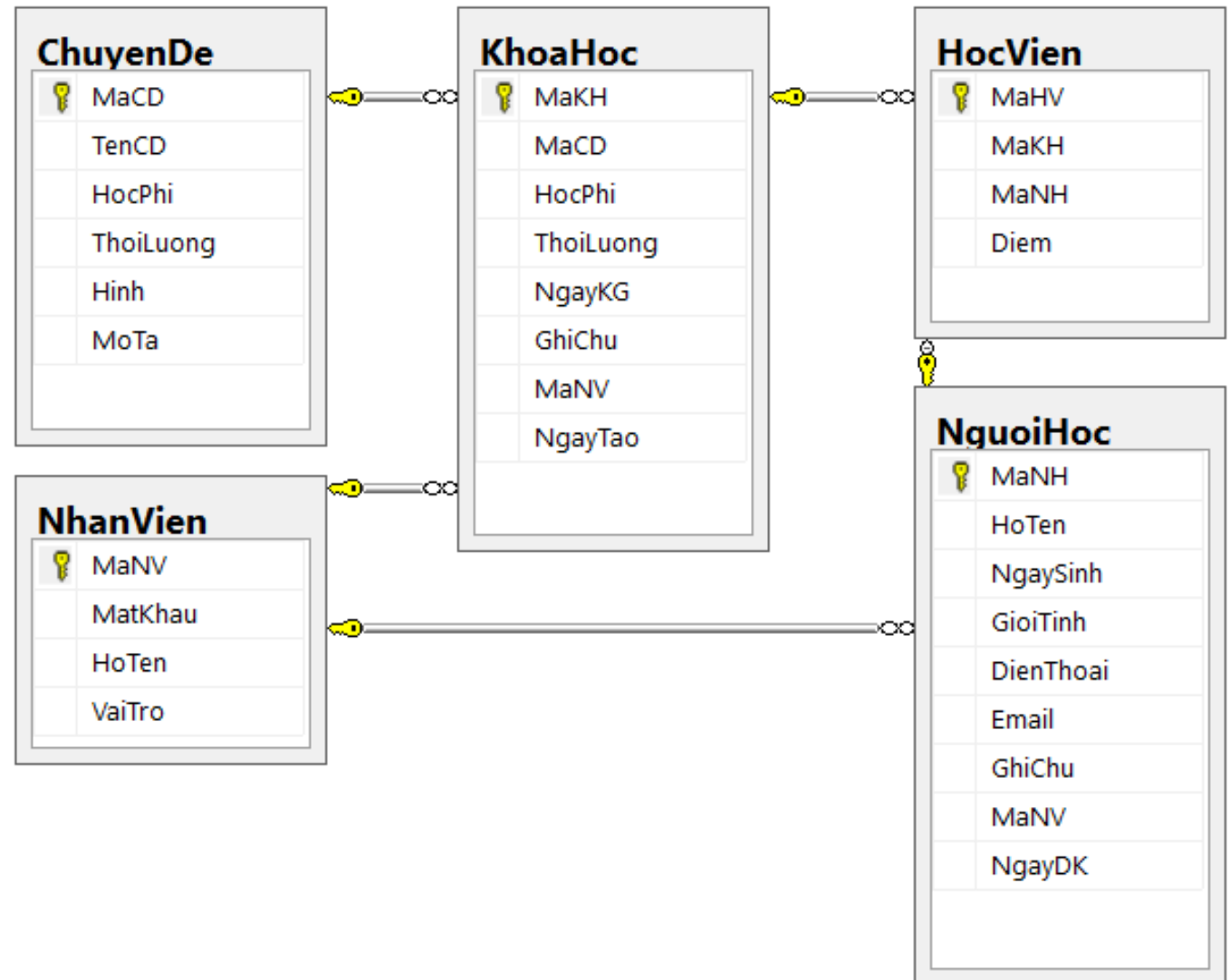


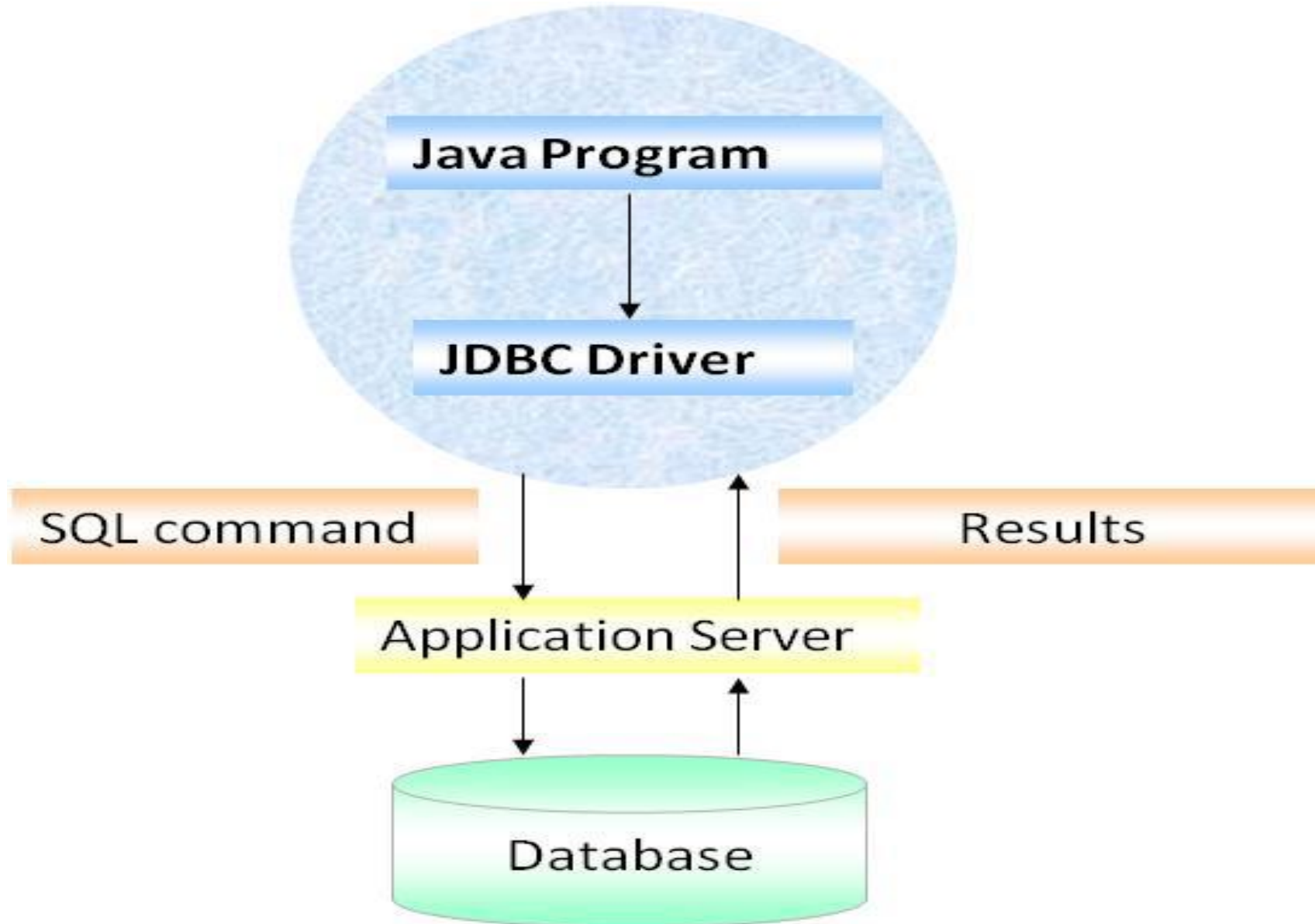
## Table

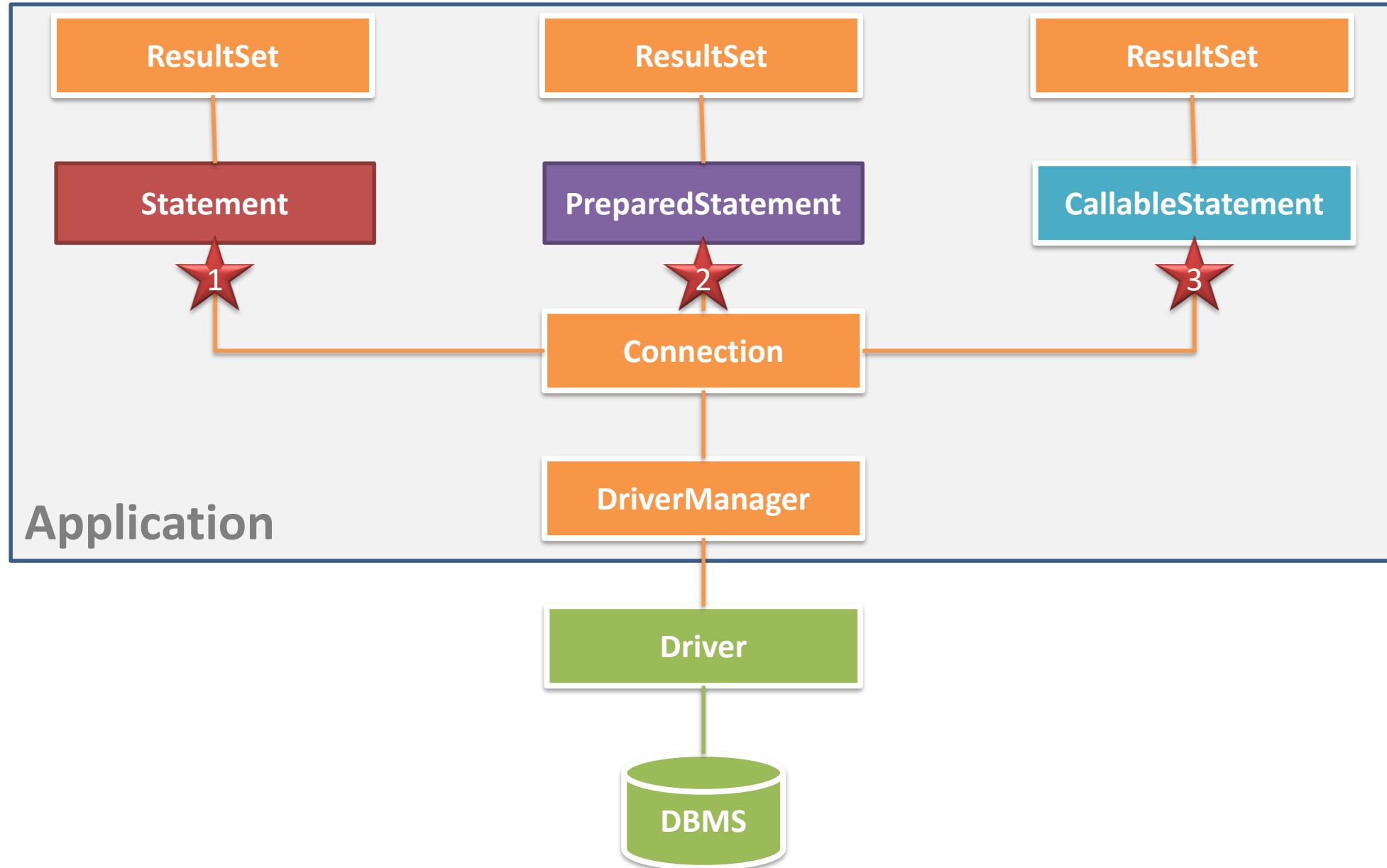
- ❖ Chuyên đề
- ❖ Khóa học
- ❖ Người học
- ❖ Học viên
- ❖ Nhân viên

## Stored Procedure

- ❖ Sp\_BangDiem(INT MaKH)
- ❖ Sp\_DiemChuyenDe()
- ❖ Sp\_DoanhThu(INT year)
- ❖ Sp\_LuongNguoiHoc()





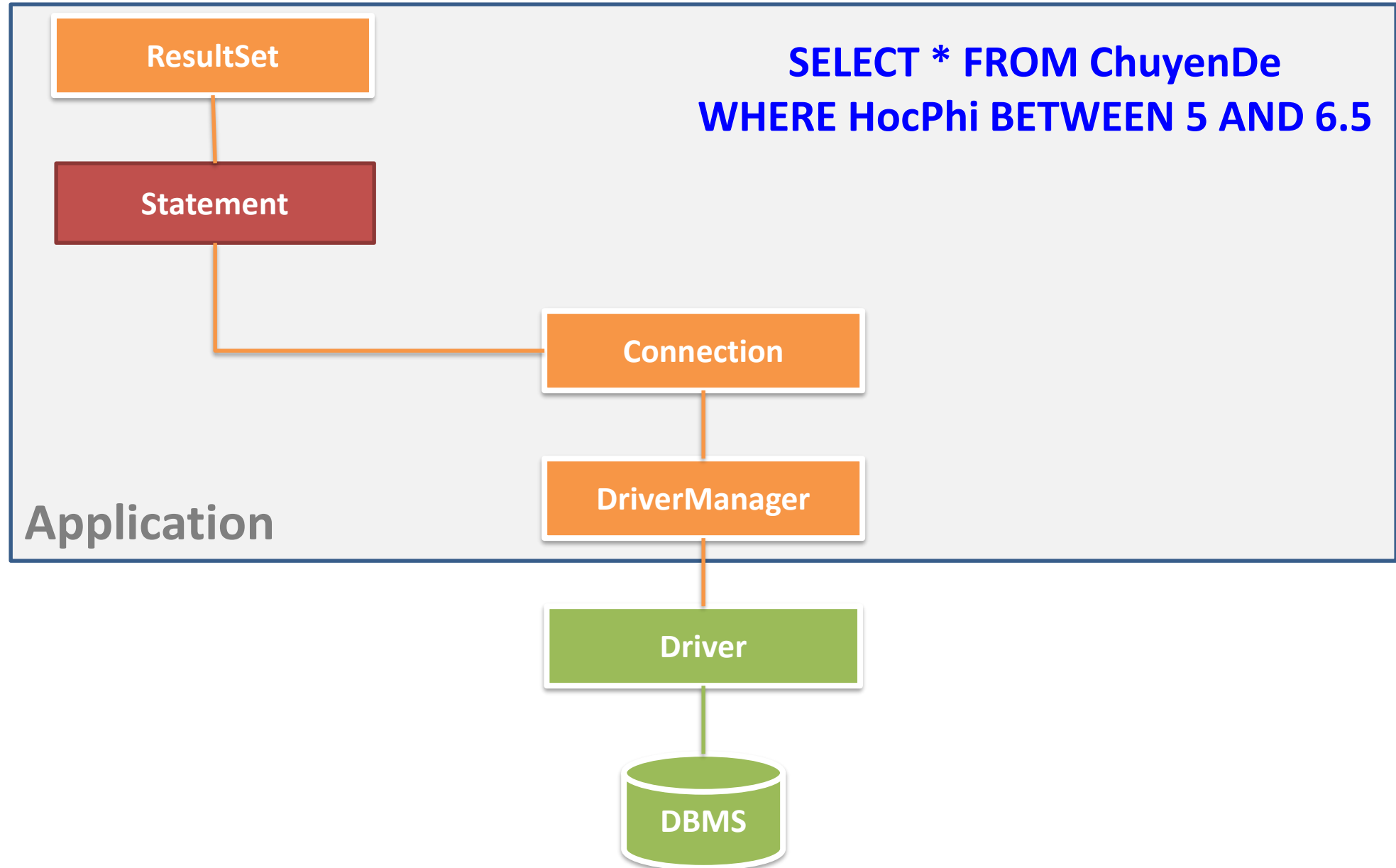




# SỬ DỤNG STATEMENT

---





## NhanVien

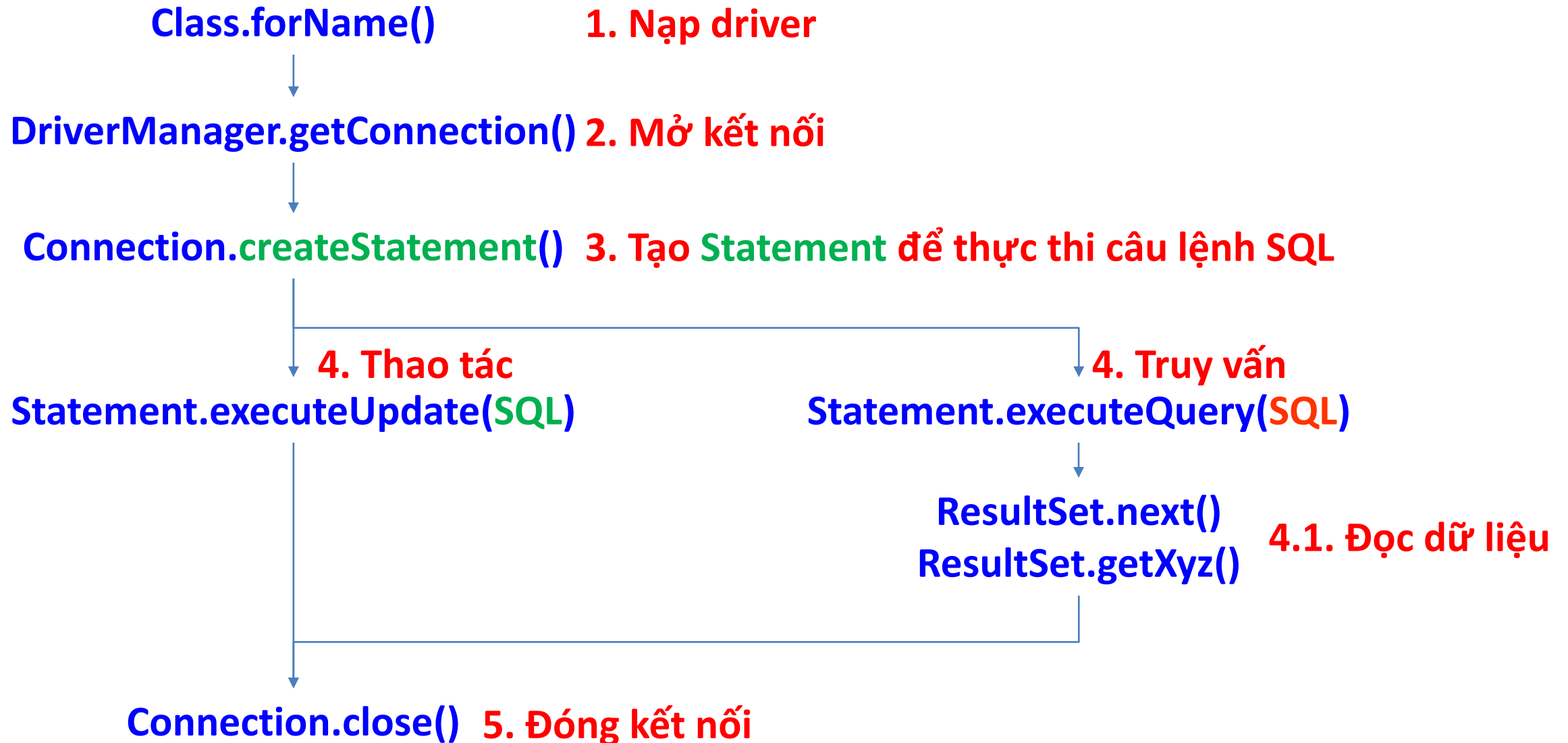
	Column Name	Condensed Type	Nullable	Default Value	Description
	MaNV	nvarchar(50)	No		Mã nhân viên, PK
	MatKhai	nvarchar(50)	No		Mật khẩu
	HoTen	nvarchar(50)	No		Họ và tên
	VaiTro	bit	No	((0))	Vai trò, 1-trưởng ...

INSERT INTO NhanVien(MaNV, MatKhai, HoTen, VaiTro)  
VALUES (N'TeoNV', N'123', N'Nguyễn Văn Tèo', 1)

UPDATE NhanVien  
SET MatKhai = N'123', HoTen = N'Nguyễn Văn Tèo', VaiTro = 1  
WHERE (MaNV = N'TeoNV')

DELETE FROM NhanVien WHERE (MaNV = N'TeoNV')

SELECT \* FROM NhanVien WHERE (MaNV = N'TeoNV')



```
String sql = "INSERT NhanVien(MaNV, HonTen, MatKhau, VaiTro) "  
            + " VALUES(N'TeoNV', N'Nguyễn Văn Tèo', N'123', 1)";
```

```
Class.forName(driver);
```

```
Connection conn = DriverManager.getConnection(dburl, user, pass);
```

```
Statement stmt = conn.createStatement();
```

```
int count = stmt.executeUpdate(sql);
```

```
conn.close();
```

## ❑ Câu lệnh SQL

❖ INSERT

❖ UPDATE

❖ DELETE

```
String sql = "SELECT * FROM ChuyenDe WHERE HocPhi BETWEEN 500.0 AND 900.0";  
Class.forName(driver);  
Connection conn = DriverManager.getConnection(dburl, user, pass);  
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery(sql);  
while (rs.next()) {  
    String name = rs.getString("TenCD");  
    double fee = rs.getDouble("HocPhi");  
}  
conn.close();
```

#### ❑ rs.next()

❖ Đọc 1 bản ghi tiếp theo

#### ❑ rs.getXYZ()

❖ Đọc 1 giá trị của cột có tên tại bản ghi hiện tại

```
public class Jdbc {  
    static String driver = "com.microsoft.sqlserver.SQLServerDriver";  
    static String dburl = "jdbc:sqlserver://localhost;database=EduSys";  
    static String user = "sa";  
    static String pass = "123";  
    static {  
        try {  
            Class.forName(driver);  
        } catch (Exception e) {  
            throw new RuntimeException(e);  
        }  
    }  
    // Truy vấn  
    public static void query() throws SQLException {...}  
    // Thao tác  
    public static void update() throws SQLException {...}  
}
```

Thông số kết nối CSDL

Nạp Driver vào ứng dụng **chỉ một lần duy nhất**

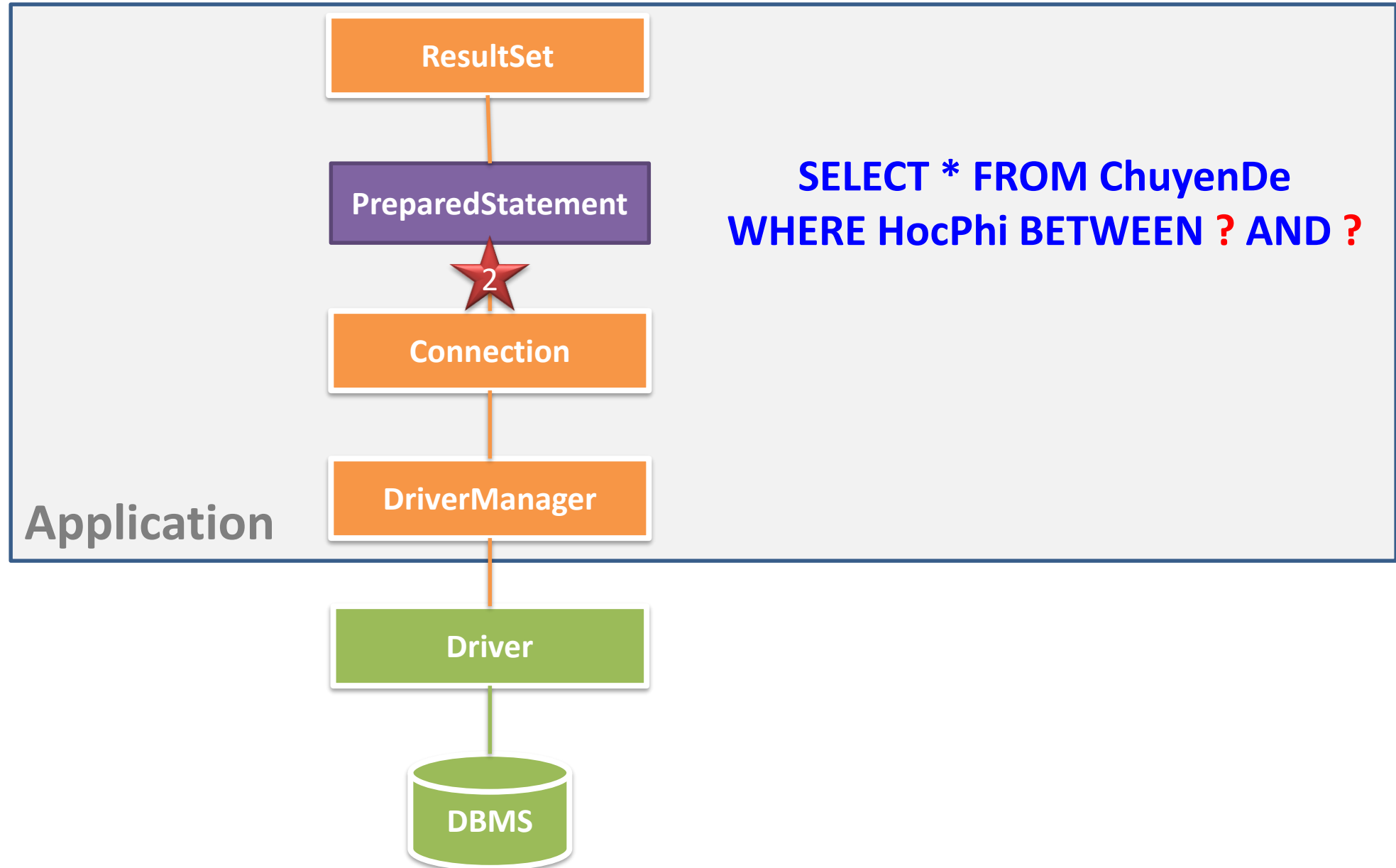
+ SELECT

+ INSERT, UPDATE và DELETE



# SỬ DỤNG PREPARED STATEMENT

---





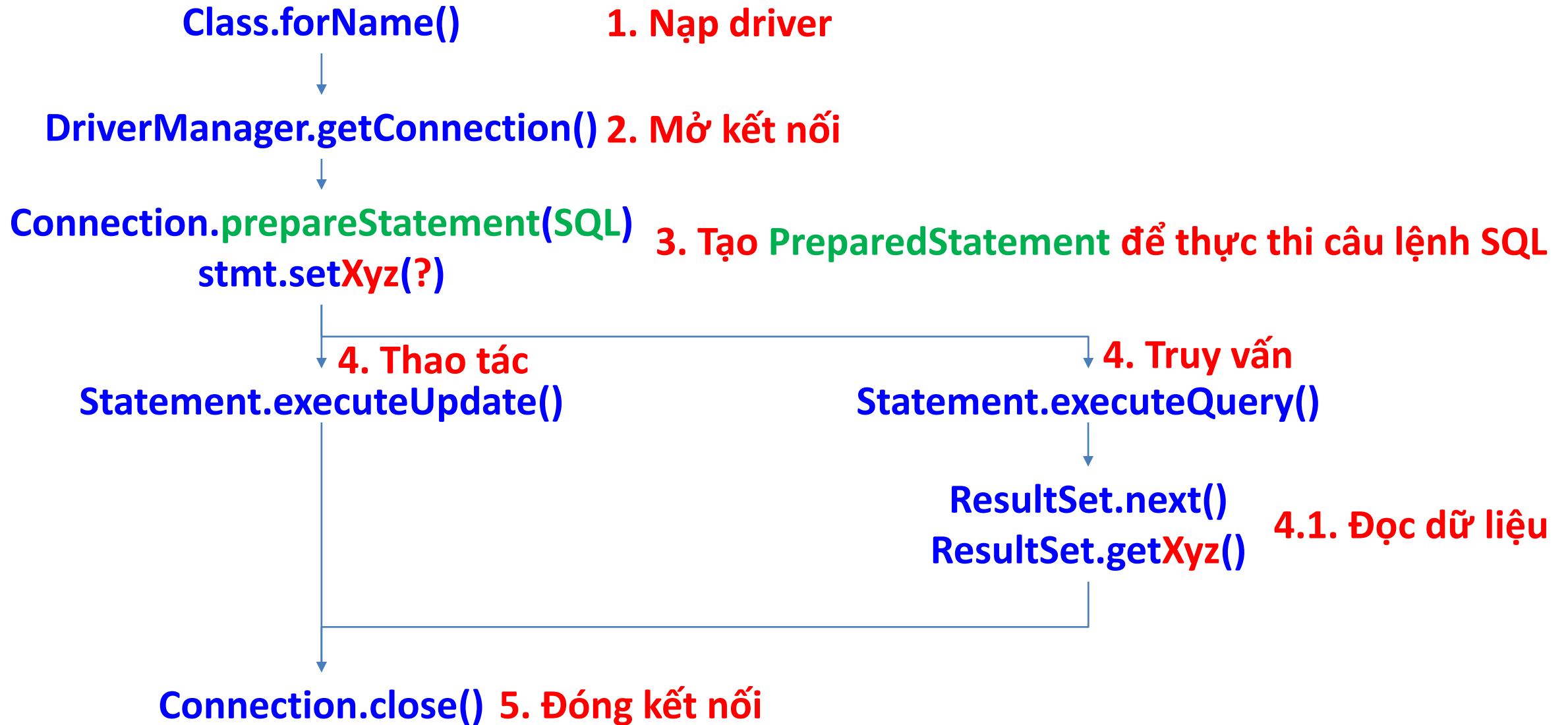
```
INSERT INTO NhanVien(MaNV, MatKhau, HoTen, VaiTro)  
VALUES (?, ?, ?, ?)
```

```
UPDATE NhanVien  
SET MatKhau = ?, HoTen = ?, VaiTro = ?  
WHERE (MaNV = ?)
```

```
DELETE FROM NhanVien WHERE (MaNV = ?)
```

```
SELECT * FROM NhanVien WHERE (MaNV = ?)
```

# CÁC BƯỚC LẬP TRÌNH JDBC - PREPAREDSTATEMENT



```
String sql = "INSERT NhanVien(MaNV, HonTen, MatKhai, VaiTro) VALUES(?, ?, ?, ?)";
```

```
Connection conn = DriverManager.getConnection(dburl, user, pass);
```

```
PreparedStatement stmt = conn.prepareStatement(sql);
```

```
stmt.setString(1, "TeoNV");
```

```
stmt.setString(2, "Nguyễn Văn Tèo");
```

```
stmt.setString(3, "123");
```

```
stmt.setBoolean(4, true);
```

```
int count = stmt.executeUpdate();
```

```
conn.close();
```

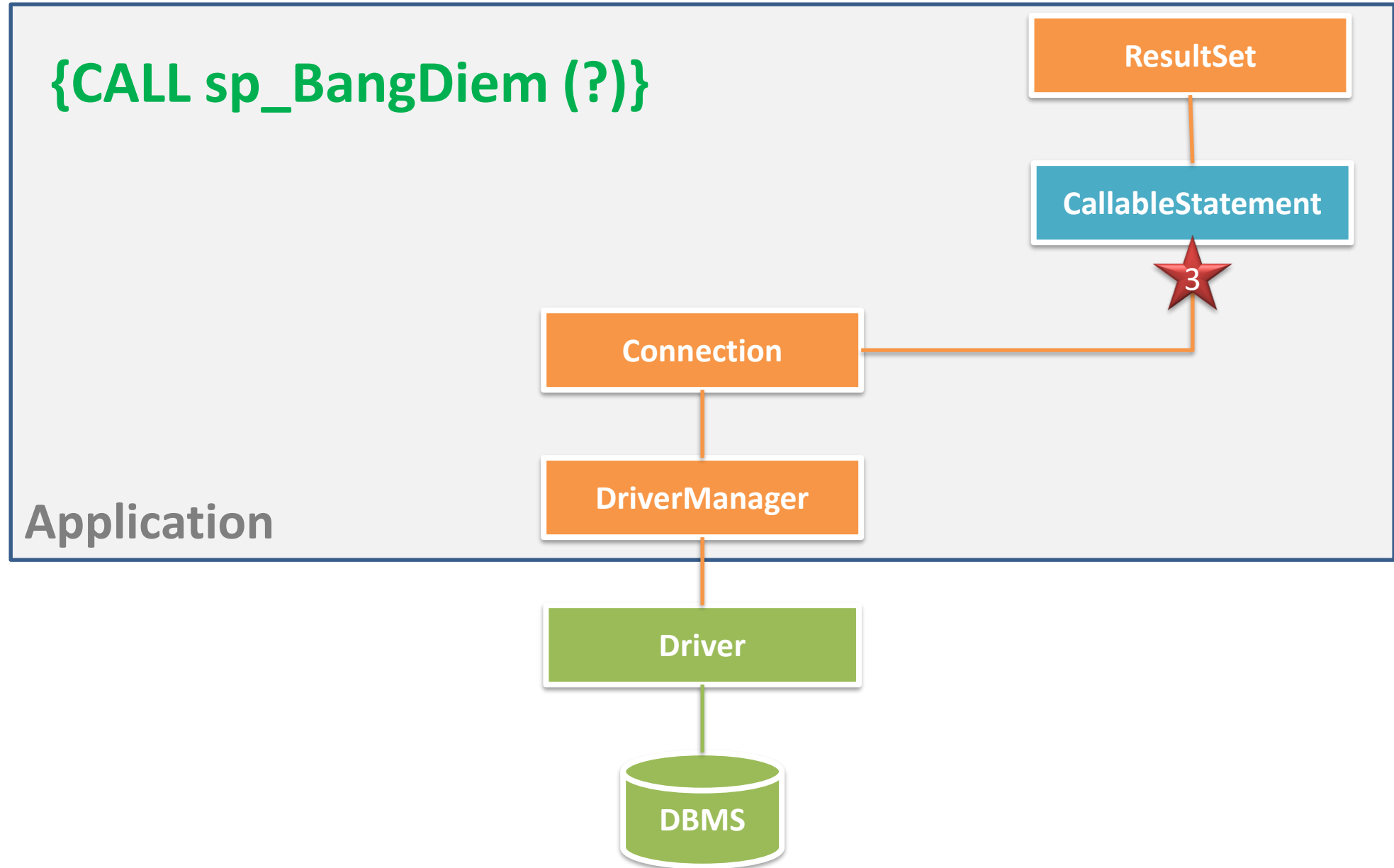
```
stmt.setObject(1, "TeoNV")  
stmt.setObject(2, "Nguyễn Văn Tèo")  
stmt.setObject(3, "123")  
stmt.setObject(4, true)
```

```
String sql = "SELECT * FROM ChuyenDe WHERE HocPhi BETWEEN ? AND ?";  
Connection conn = DriverManager.getConnection(dburl, user, pass);  
PreparedStatement stmt = conn.prepareStatement(sql);  
stmt.setDouble(1, 500.0);  
stmt.setDouble(2, 900.0);  
ResultSet rs = stmt.executeQuery();  
while (rs.next()) {  
    String name = rs.getString("TenCD");  
    double fee = rs.getDouble("HocPhi");  
}  
conn.close();
```



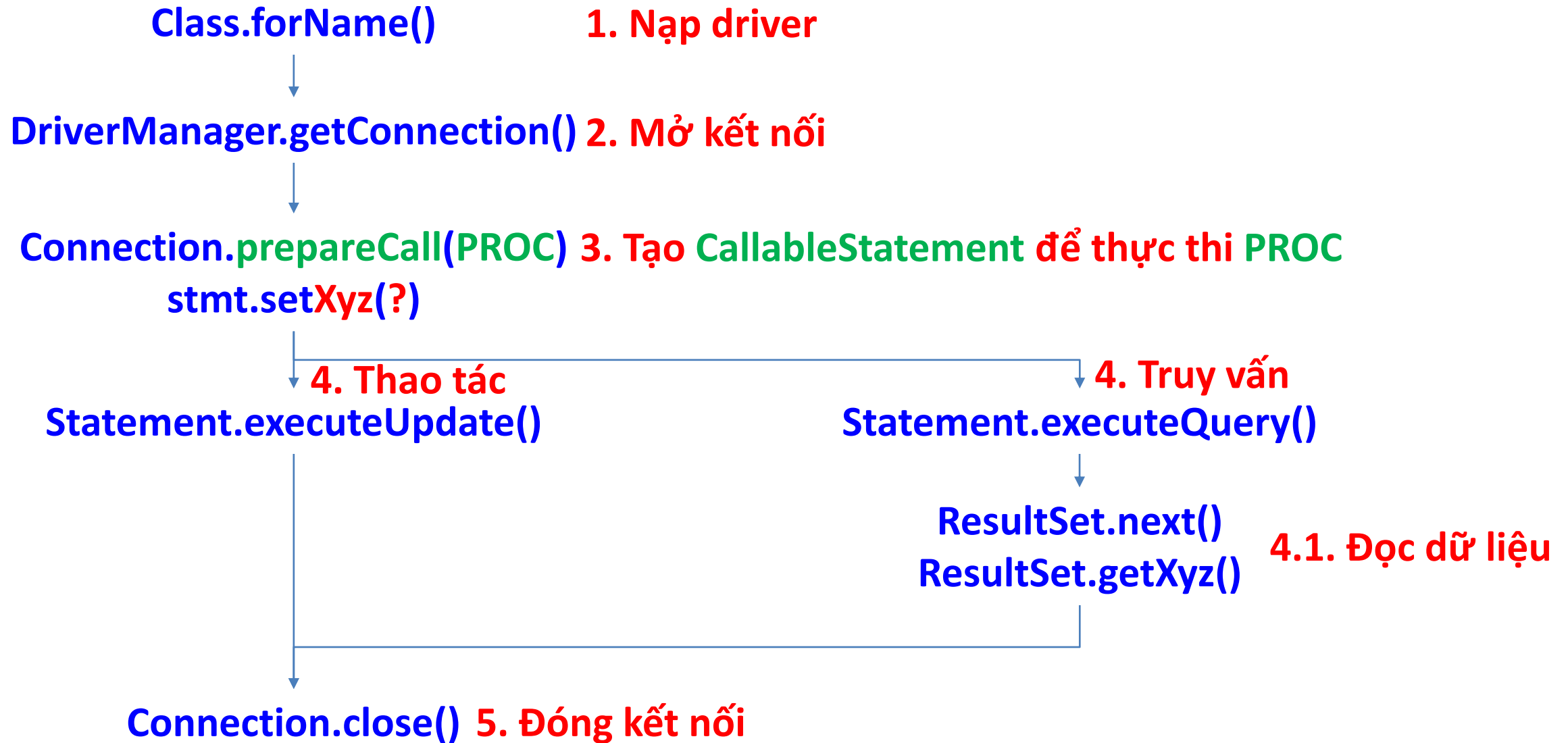
# SỬ DỤNG CALLABLE STATEMENT

---



```
CREATE PROC sp_BangDiem(@MaKH INT)
AS BEGIN
    SELECT
        nh.MaNH,
        nh.HoTen,
        hv.Diem
    FROM HocVien hv
        JOIN NguoiHoc nh ON nh.MaNH=hv.MaNH
    WHERE hv.MaKH = @MaKH
    ORDER BY hv.Diem DESC
END
```

# CÁC BƯỚC LẬP TRÌNH JDBC - CALLABLESTATEMENT





```
String sql = "{CALL sp_BangDiem (?)}";  
Connection conn = DriverManager.getConnection(dburl, user, pass);  
CallableStatement stmt = conn.prepareCall(sql);  
stmt.setObject(1, 1000.0);  
ResultSet rs = stmt.executeQuery();  
while (rs.next()) {  
    String name = rs.getString("HoTen");  
    double mark = rs.getDouble("Diem");  
}  
conn.close();
```

## ❑ Statement

- ❖ Ngắn gọn nhưng không an toàn dễ bị hack bởi SQL Injection

## ❑ PreparedStatement

- ❖ Code dài hơn nhưng có nhiều ưu điểm
  - An toàn, tránh SQL Injection
  - Không phụ thuộc Unicode, dấu nháy đơn (trong suốt DBMS)
  - Cho phép làm việc với dữ liệu nhị phân (byte[])
  - Mã rõ ràng
  - Chạy nhanh hơn nếu câu lệnh được sử dụng nhiều lần (prepared)
- ❖ Có thể thay thế hoàn toàn Statement

## ❑ CallableStatement

- ❖ Ngắn gọn, nhưng phải viết thủ tục lưu
- ❖ Ngoài ưu điểm như PreparedStatement, nó còn chạy nhanh hơn

- ✓ Mô hình ứng dụng JDBC
- ✓ Tìm hiểu JDBC API
- ✓ Sử dụng
  - ✓ Statement
  - ✓ PreparedStatement
  - ✓ CallableStatement





**Cảm ơn**