



FPT POLYTECHNIC



www.poly.edu.vn



Conceive Design Implement Operate

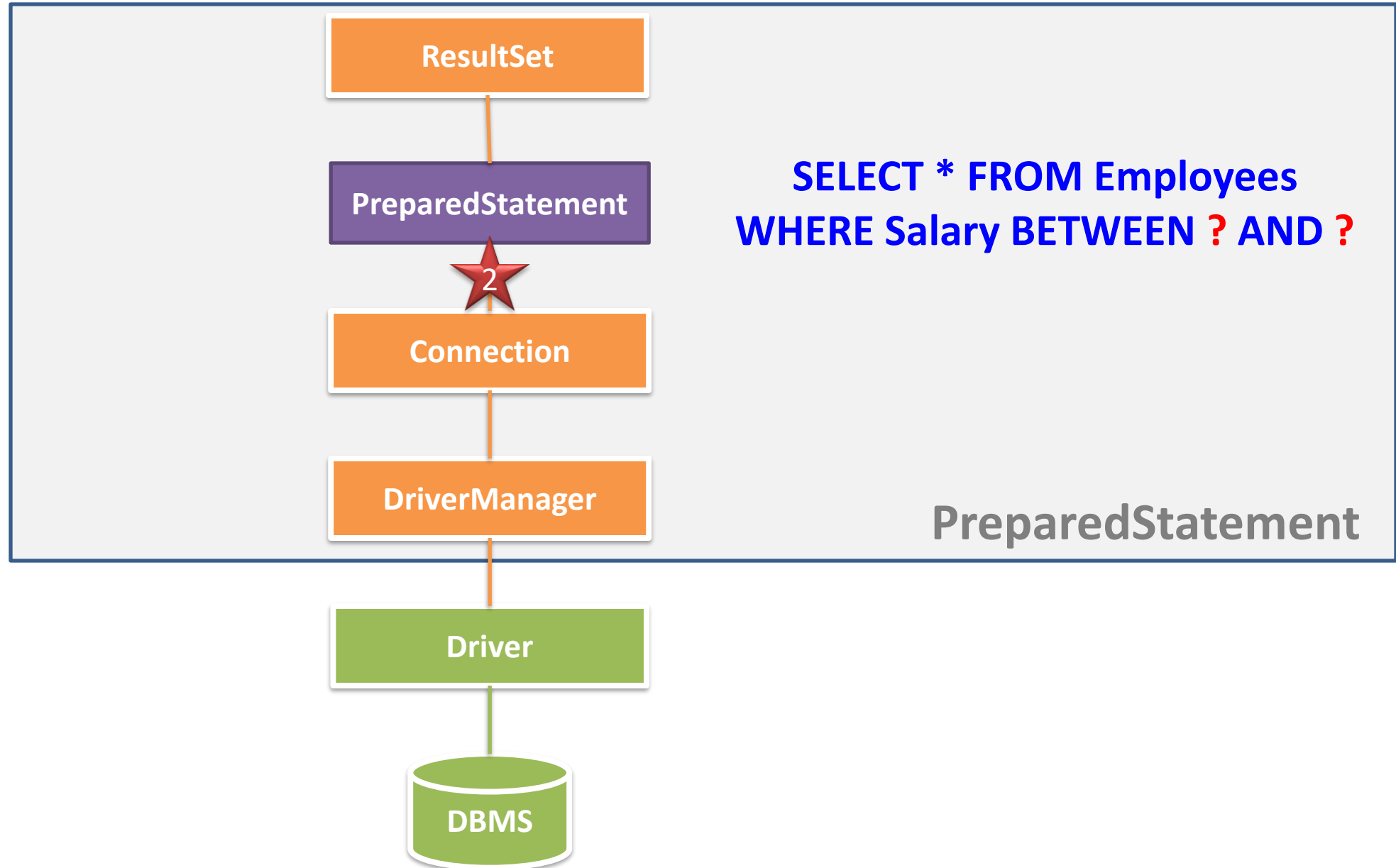
ỨNG DỤNG JDBC KẾT NỐI DATABASE

LẬP TRÌNH JAVA #3 (P6.2)

- ❑ Lập trình JDBC sử dụng PreparedStatement
- ❑ Lập trình JDBC sử dụng CallableStatement
- ❑ Ưu và nhược điểm của các phương pháp lập trình
 - ❖ Statement
 - ❖ PreparedStatement
 - ❖ CallableStatement



LẬP TRÌNH JDBC VỚI PREPAREDSTATEMENT



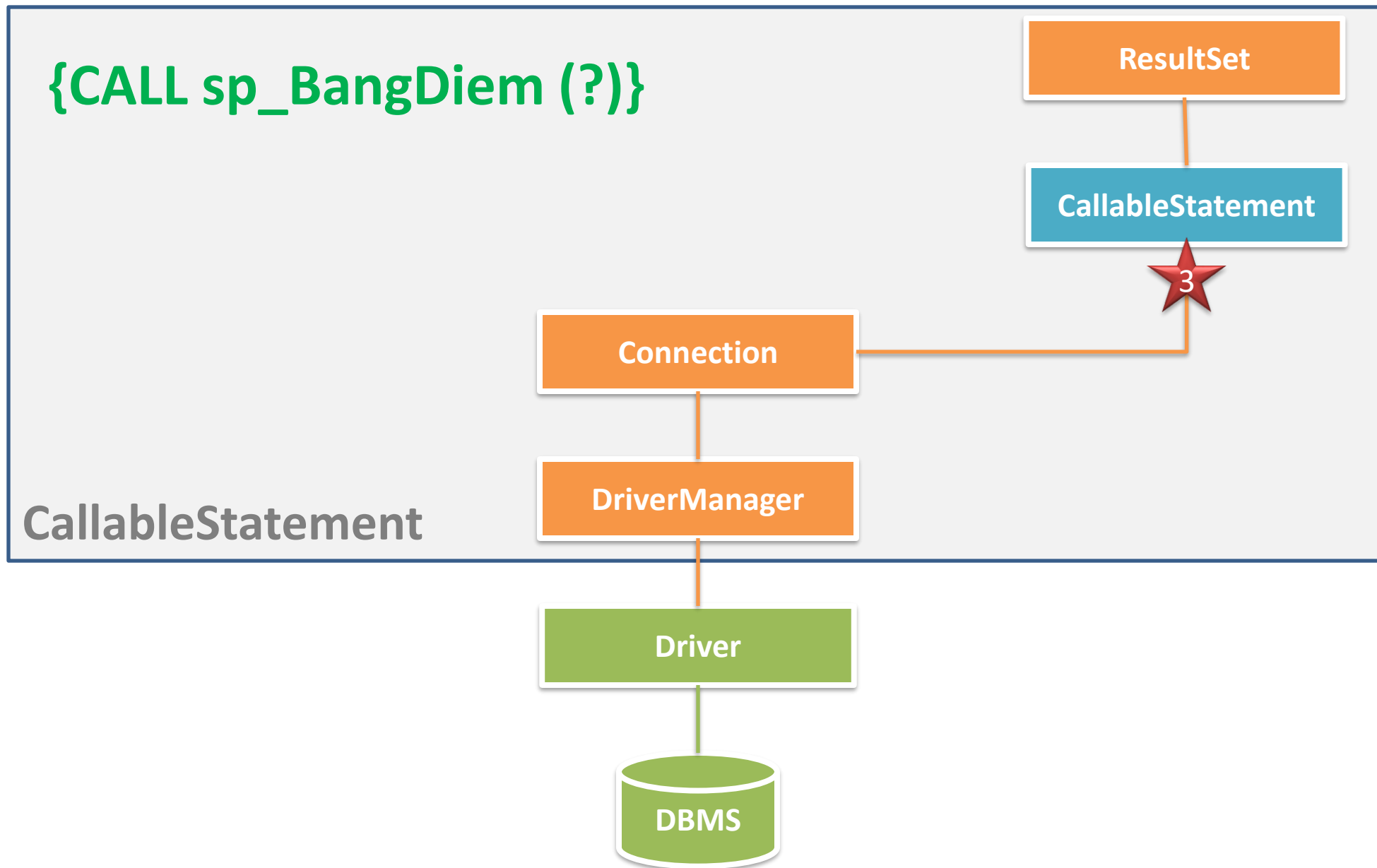
```
Class.forName(driver);
Connection connection =
    DriverManager.getConnection(dburl, username, password);
String psql = "INSERT INTO Departments(Id, Name, Description) VALUES(?, ?, ?)";
PreparedStatement statement = connection.prepareStatement(psql);
statement.setString(1, "id");
statement.setString(2, "name");
statement.setString(3, "description");
int rows = statement.executeUpdate();
connection.close();
```

❑ Chú ý sự khác biệt với Statement

- ❖ Câu lệnh SQL không chứa dữ liệu trực tiếp mà chỉ là các dấu hỏi (?)
- ❖ Sử dụng `prepareStatement(sql)` để tạo `PreparedStatement`
- ❖ Sử dụng các phương thức `set<Type>(index, value)` để cấp dữ liệu vào các vị trí của ? (bắt đầu từ 1). *Có thể sử dụng `setObject()` thay cho `set<Type>()` nhưng value phải đúng kiểu dữ liệu*
- ❖ Sử dụng `executeQuery()` không đối số để thực thi câu lệnh

```
Class.forName(driver);
Connection connection =
    DriverManager.getConnection(dburl, username, password);
String psql = "SELECT * FROM Employees WHERE Salary BETWEEN ? AND ?";
PreparedStatement statement = connection.prepareStatement(psql);
statement.setDouble(1, 500);
statement.setDouble(2, 1000);
ResultSet resultSet = statement.executeQuery();
while(resultSet.next()) {
    String id = resultSet.getString("Id");
    double salary = resultSet.getDouble("Salary");
}
connection.close();
```





--Tạo PROC tổng hợp lương từng phòng

```
CREATE PROCEDURE sp_SalaryReport (@FromYear INT, @ToYear INT)
AS BEGIN
    SELECT
        d.Name AS 'Department',
        count(*) AS 'EmployeeCount',
        avg(e.Salary) AS 'AverageSalary',
        sum(e.Salary) AS 'SalaryTotal'
    FROM Employees e
        JOIN Departments d ON e.DepartmentId=d.Id
    WHERE year(e.Birthday) BETWEEN @FromYear AND @ToYear
    GROUP BY d.Name
END
```



```
Class.forName(driver);
Connection connection =
    DriverManager.getConnection(dburl, username, password);
String csql = "{CALL sp_SalaryReport(?, ?)}";
CallableStatement statement = connection.prepareCall(csql);
statement.setInt(1, 2020);
statement.setInt(2, 2024);
ResultSet resultSet = statement.executeQuery();
while(resultSet.next()) {
    String id = resultSet.getString("Id");
    double salary = resultSet.getDouble("Salary");
}
connection.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

- ☑ Lập trình JDBC sử dụng PreparedStatement
- ☑ Lập trình JDBC sử dụng CallableStatement
- ☑ Ưu và nhược điểm của các phương pháp lập trình
 - ❖ Statement
 - ❖ PreparedStatement
 - ❖ CallableStatement





Cảm ơn