

# Persistence Using Spring Data JDBC 3.0.0 and Spring Framework 6



**Bryan Hansen**

Director of Software Development

@bh5k



# Version Check

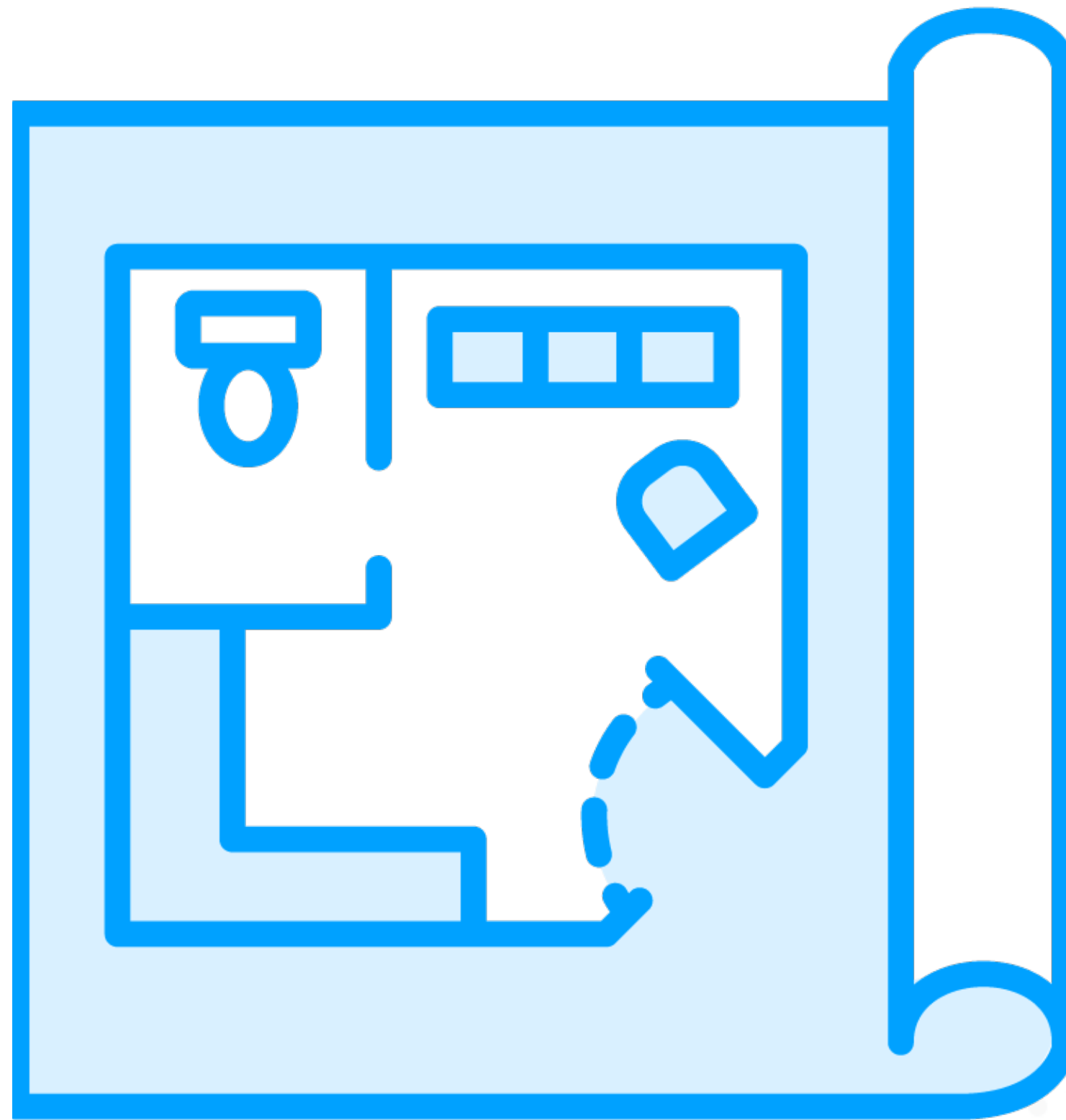


**This course was created by using:**

- Java 21
- Spring Boot 3
- Spring Framework 6
- JDBC 3.0.0



# What Is Spring JDBC?



JDBC framework

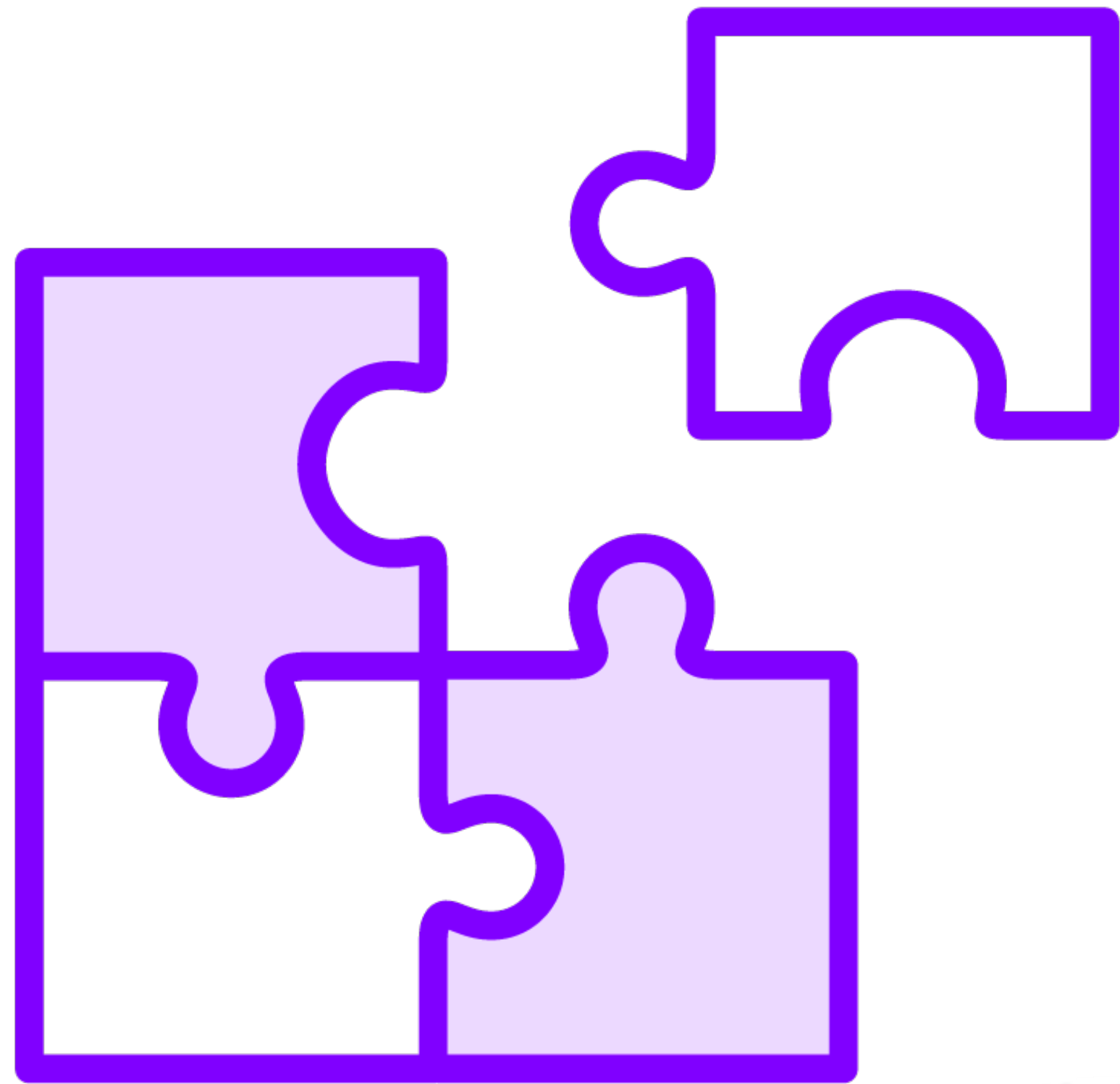
Pattern-based

Injected resources

ORM

Existing databases

# The Problem



**Complexity**

**Design**

**Portability**

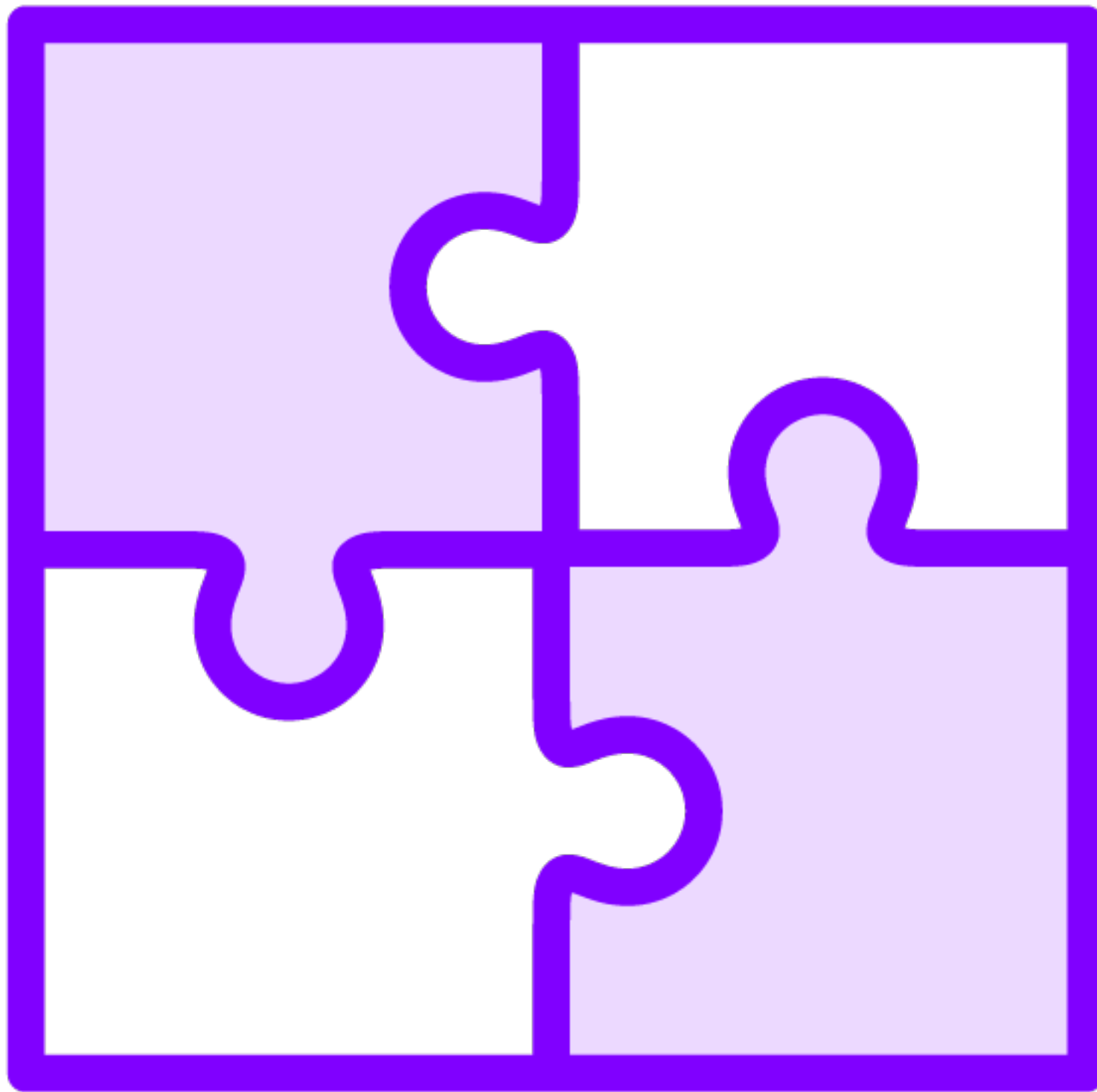
**Business Focus**

# Business Focus

```
public Car getById(String id) {
    Connection con = null;
    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        String sql = "select * from CAR where ID = ?";
        con = DriverManager.getConnection("localhost:3306/cars");
        stmt = con.prepareStatement(sql);
        stmt.setString(1, id);
        rs = stmt.executeQuery();
        if(rs.next()) {
            Car car = new Car();
            car.setMake(rs.getString(1));
            return car;
        }
        else {
            return null;
        }
    } catch (SQLException e) { e.printStackTrace();}
    finally {
        try {
            if(rs != null && !rs.isClosed()) {
                rs.close();
            }
        } catch (Exception e) {}
    }
    return null;
}
```

# The Solution



Configuration

Focus

Testing

Dependency Injection



# Business Focus Revisted

## CarJdbc.java

```
public Car getById(String id) {
    Connection con = null;
    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        String sql = "select * from CAR where ID = ?";
        con = DriverManager.getConnection("localhost:3306/cars");
        stmt = con.prepareStatement(sql);
        stmt.setString(1, id);
        rs = stmt.executeQuery();
        if(rs.next()) {
            Car car = new Car();
            car.setMake(rs.getString(1));
            return car;
        }
        else {
            return null;
        }
    } catch (SQLException e) { e.printStackTrace();}
    finally {
        try {
            if(rs != null && !rs.isClosed()) {
                rs.close();
            }
        } catch (Exception e) {}
    }
    return null;
}
```

## CarDao.java

```
public Car findCar(String id) {
    return getEntityManager().find(Car.class, id);
}
```

# Summary

**What is Spring JDBC?**

**The Problem**

**The Solution**

**Business Focus**