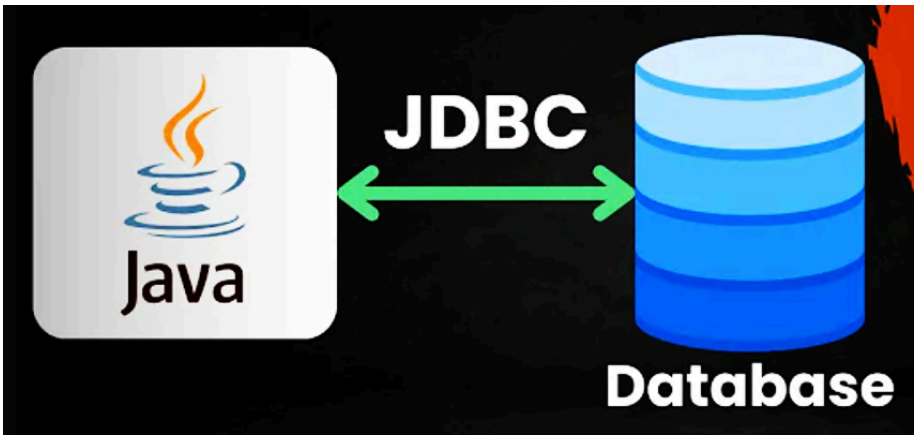
17-Spring Data JPA

When a client sends a request in our application, it is handled through several layers:

1. **Request routing and mapping**:
    - The request is first routed through the front controller.
    - It is then mapped to the appropriate request mappings in the controller layer.
    - Finally, it is directed to the CSR (Controller, Service, Repository) layer.
2. **Service Layer**:
    - The request is sent to the service layer, where the business logic is executed.
3. **Repository Layer**:
    - The request is then passed to the repository layer.
    - Note: In our project, we hardcoded the data instead of using a database, which is not a good practice.

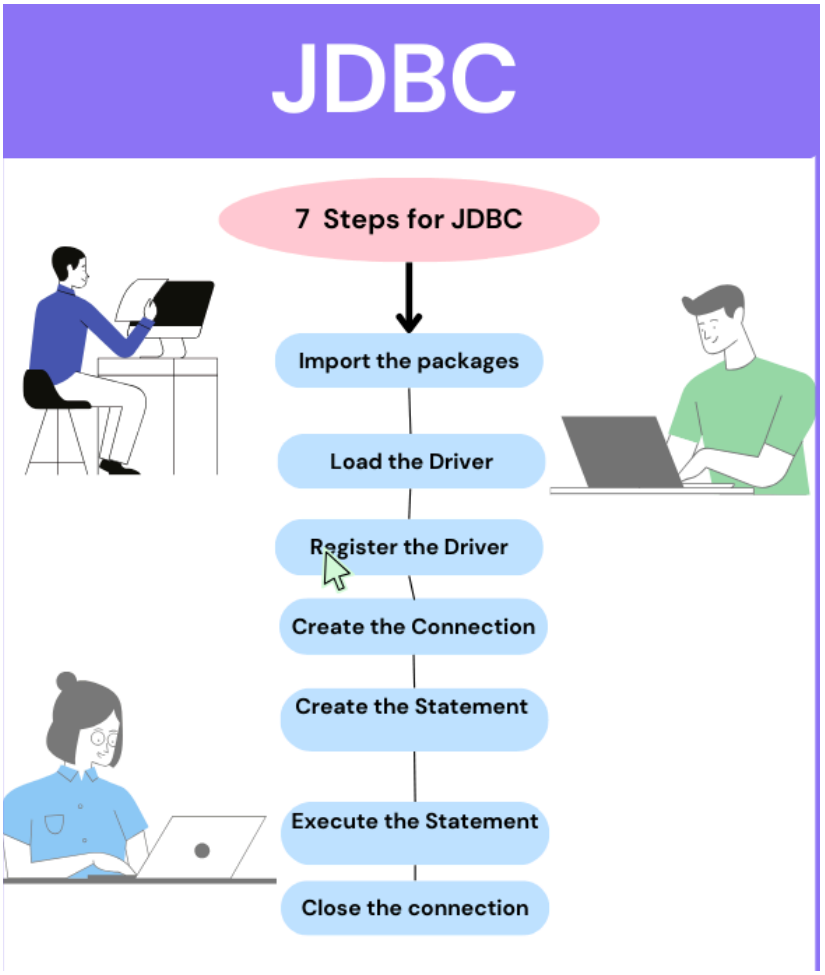
Common Databases and JDBC

- **Common Databases**: Databases like MySQL, PostgreSQL, H2, and others are widely used.
- **Repository Layer**: To interact with these databases, we need to write code in the repository layer.
- **JDBC**:
    - Initially, JDBC (Java Database Connectivity) was used to connect Java to a database.
    - This process involves seven steps, including loading and setting up a database driver.
    - The Spring Framework provides Spring JDBC templates, which are easier to use than regular JDBC.



Object-Relational Mapping (ORM)

- **What is ORM?**
  - ORM stands for "Object-Relational Mapping."
  - It is used for basic CRUD (Create, Read, Update, Delete) operations in applications.
- **Why ORM?**
  - Java follows the OOP (Object-Oriented Programming) theory and treats every entity as an object.
  - A class serves as a blueprint for creating objects, which have methods representing the class's actions or functions.
  - In our project, we have a list of goods, each with attributes like product ID, price, and name.
  - Similarly, an RDBMS (Relational Database Management System) represents this data in tables with rows and columns, where each row corresponds to an object.
- **Connecting Java to Databases**:
  - JDBC is the standard method for connecting Java to a database by following 7 steps.
  - ORM tools map Java objects to the corresponding row data in the database.
  - Each row in the database is called a record.



- **Benefits of ORM**:
    - Writing SQL queries for different databases can be challenging, especially when moving from one database to another.
    - ORM tools simplify this by mapping Java objects to database rows.
    - Classes are linked to table names, and variables are converted into columns.
    - For instance, if there are 10 objects, the ORM tool will create 10 corresponding rows in the database.
- **Popular ORM Tools**:
    - Hibernate
    - EclipseLink
    - MyBatis (formerly iBATIS)
    - Hibernate is the most widely used ORM tool today.

Java Persistence API (JPA)

- **What is JPA?**
    - JPA stands for Java Persistence API.
    - It is a set of rules and guidelines that ORM tools must follow.
    - JPA makes it easy to switch from one ORM tool to another.
- **Analogy**:
    - JPA can be compared to knowing a skill, i.e., driving a vehicle. If you know how to drive, you can drive any car that meets the same rules for how it works.

Next Steps

- In the next part, we will learn about writing code for Data JPA.
- Until then, have fun learning and coding! ☺👾