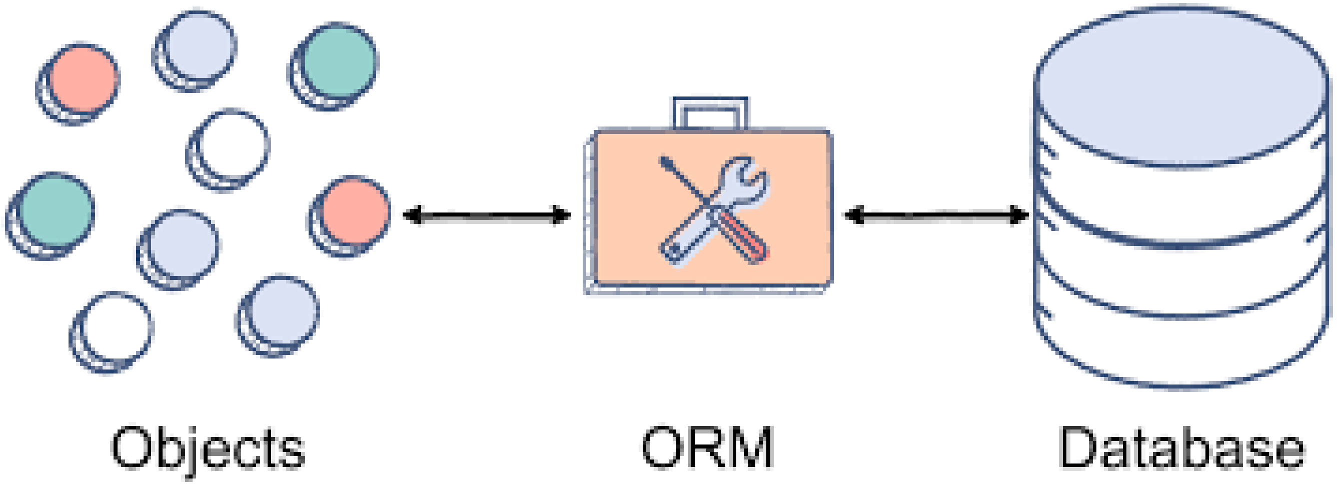
Session 3: Spring Data JPA

Learning goals

- What is Hibernate
- What is JPA
- JPA Architecture
- Spring Data JPA
- Demo

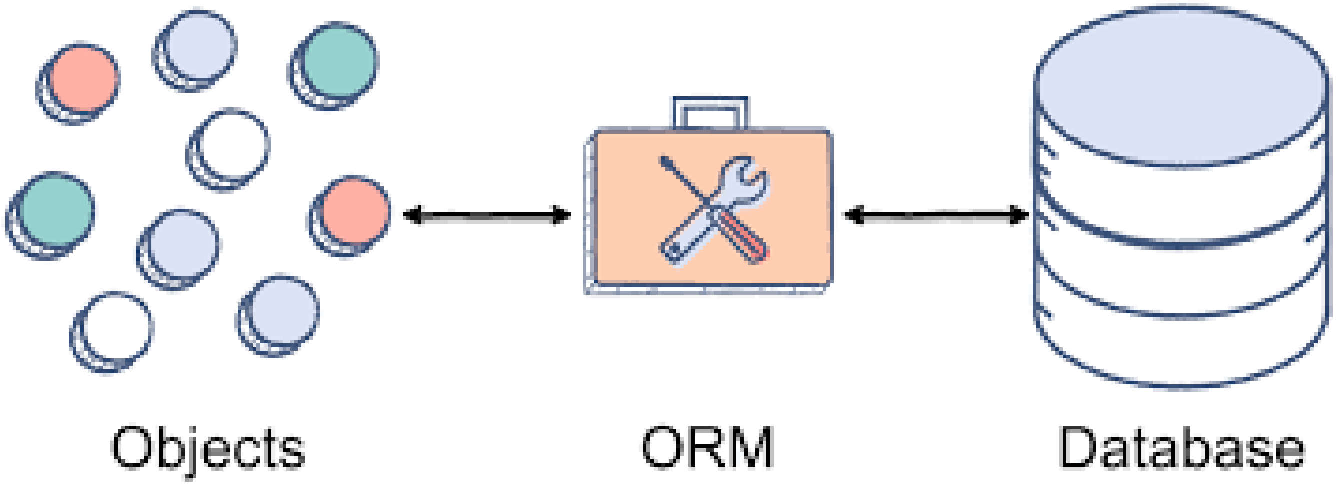
Hibernate – What is Hibernate

Hibernate is an open-source Object-Relational Mapping (ORM) framework for Java applications. It provides a convenient way to map Java classes to database tables, allowing developers to store, retrieve, and manipulate data in an object-oriented way, rather than using low-level SQL statements.

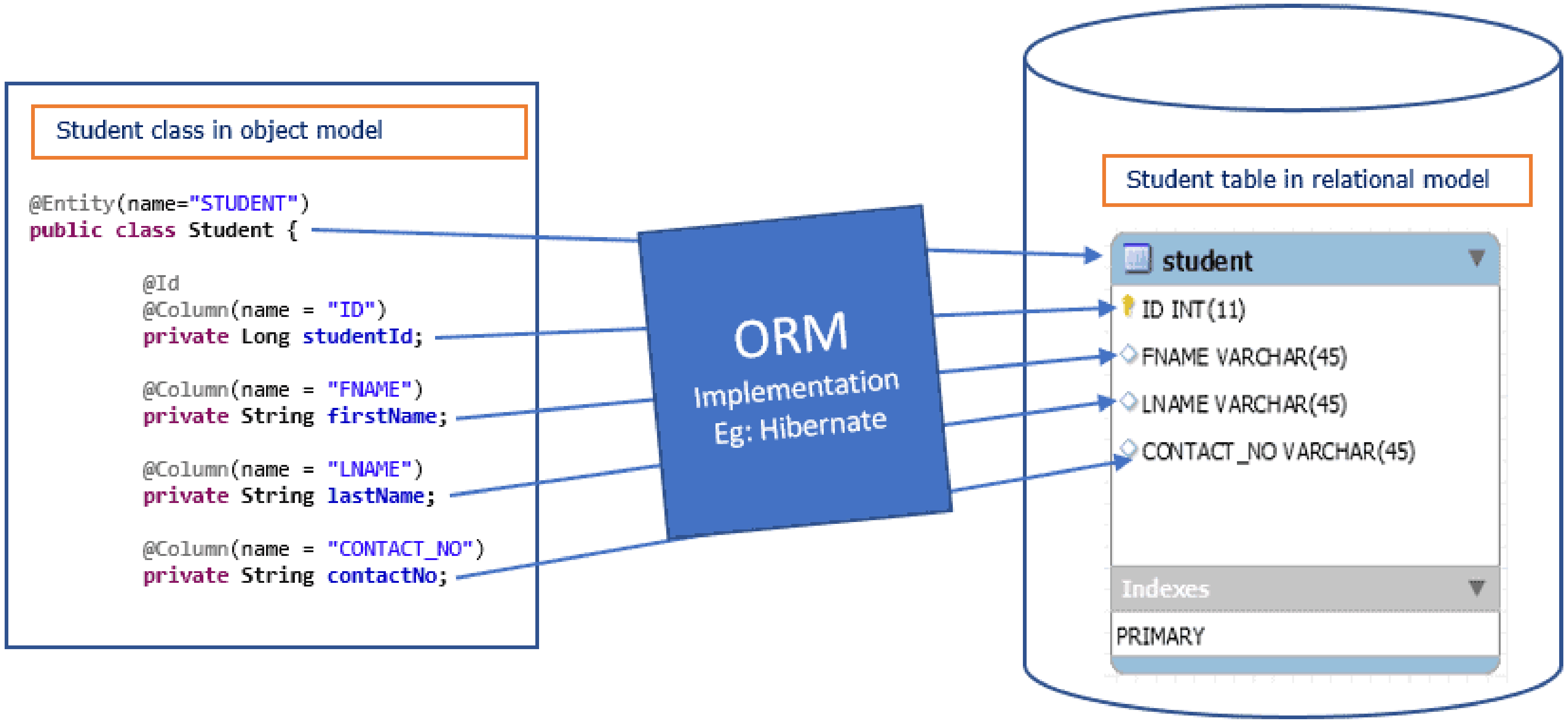


Hibernate – Why using Hibernate

- Handle of the low-level SQL
- Minimize the JDBC code
- Provide the Object-to-Relational Mapping (ORM)



Hibernate – ORM



```
@Entity(name="STUDENT")
public class Student {

        @Id
        @Column(name = "ID")
        private Long studentId;

        @Column(name = "FNAME")
        private String firstName;

        @Column(name = "LNAME")
        private String lastName;

        @Column(name = "CONTACT_NO")
        private String contactNo;
```

ORM implements responsibility of mapping the Object to Relational Model.

JavaByDeveloper

JPA – What is JPA

- JPA stands for **Java Persistence API**, which is a specification for **object-relational mapping** (ORM) in Java. It provides a standard way to map Java classes to relational database tables, allowing developers to write database code in a more object-oriented way. **JPA defines a set of interfaces and annotations that enable developers to perform CRUD** (Create, Read, Update, Delete) operations on databases using Java classes. JPA implementations, such as Hibernate, provide concrete implementations of these interfaces and annotations.

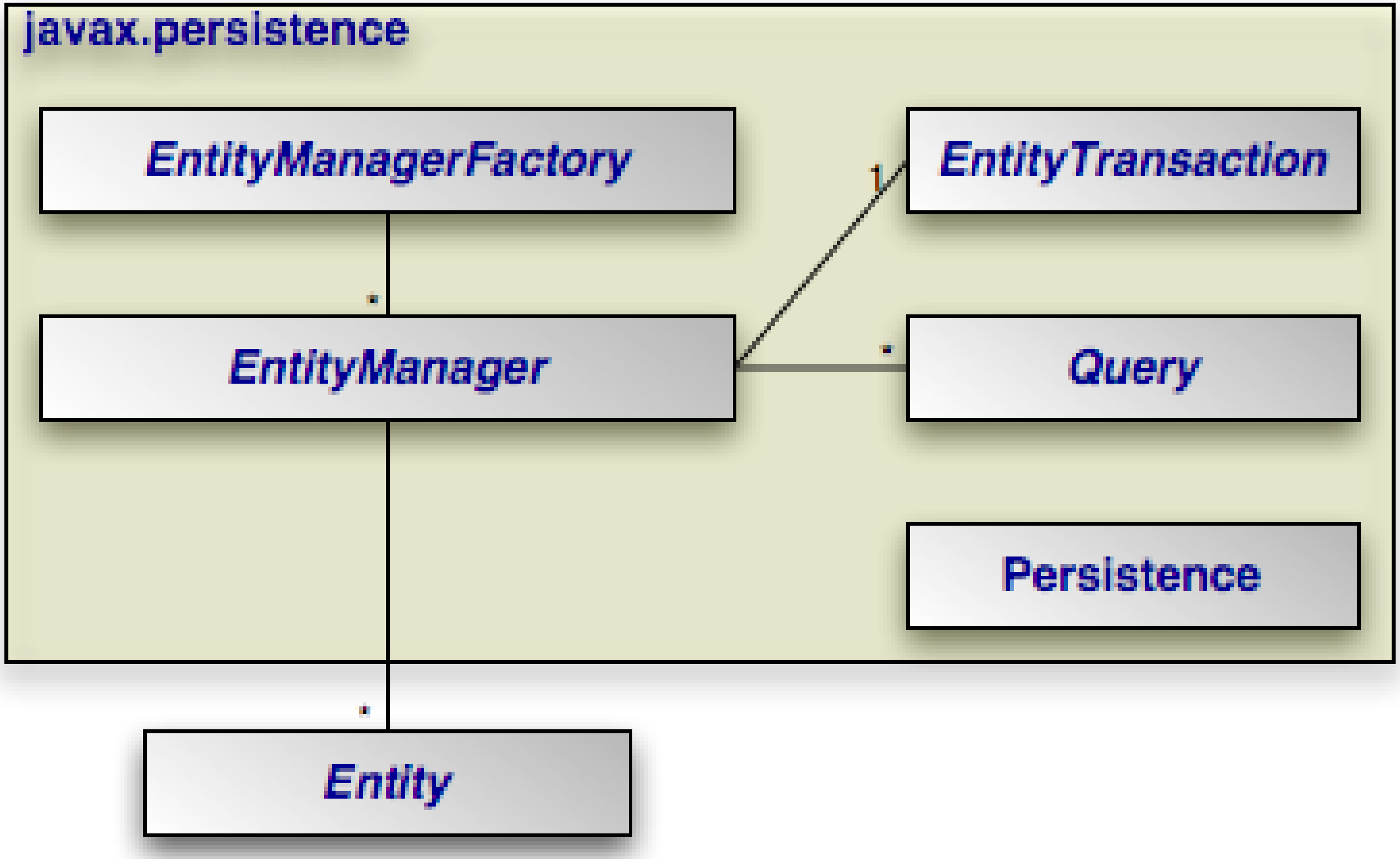
JPA – Why use JPA

- JPA simplifies database access by providing an object-oriented API, supports database portability, improves productivity, enhances performance with caching and lazy loading, and provides a standardized API for consistent database interaction across different applications and frameworks.

JPA – Implementations

- Hibernate: The most advanced and widely used.
- EclipseLink: Is based on TopLink, and is the intended path forward for persistence for Oracle and TopLink

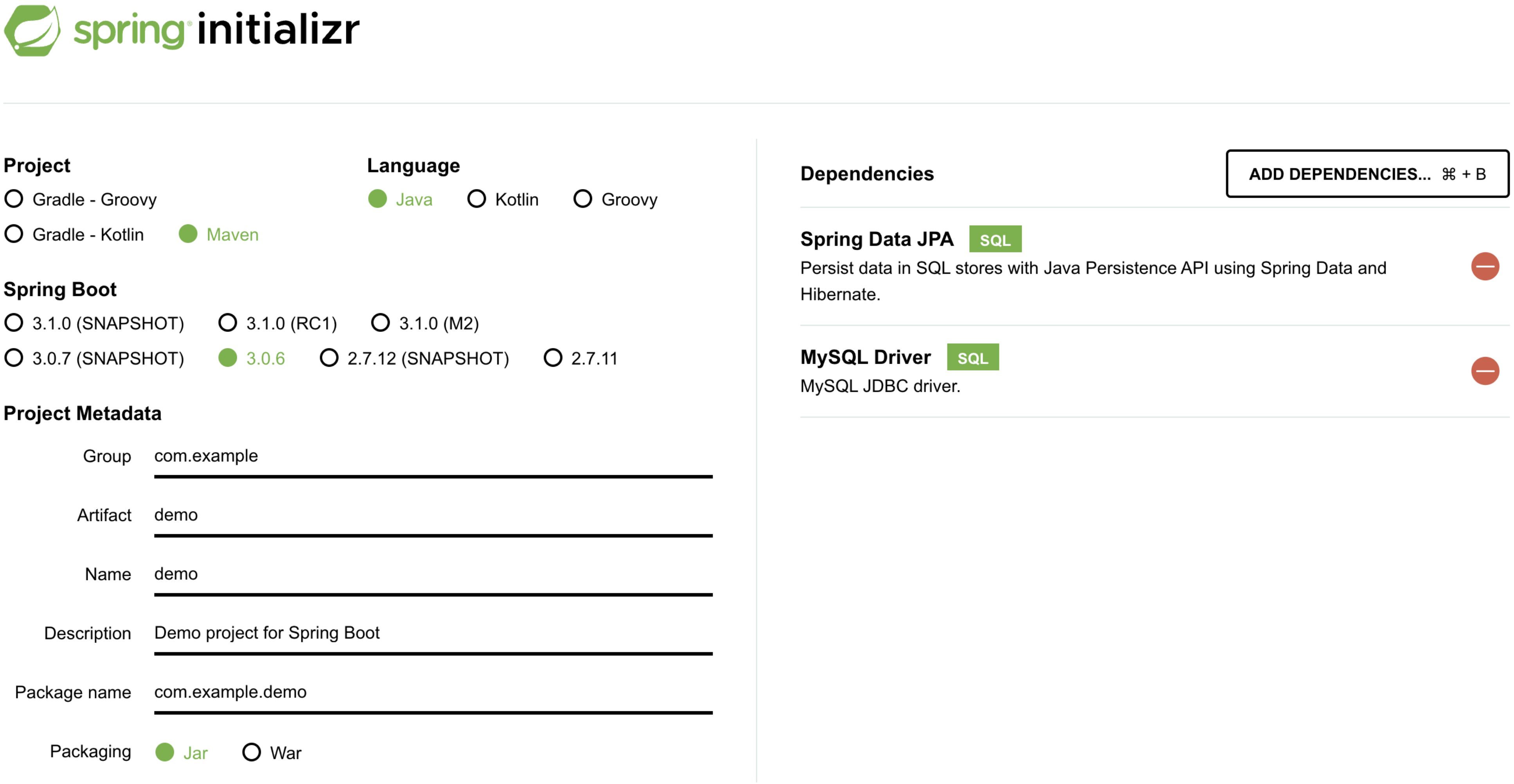
JPA – Architecture



Spring Data JPA

- Persist data in SQL stores with Java Persistence API using Spring data and Hibernate.

Demo – Create new Project



spring initializr

Project

○ Gradle - Groovy
○ Gradle - Kotlin ● Maven

Language

● Java ○ Kotlin ○ Groovy

Spring Boot

○ 3.1.0 (SNAPSHOT) ○ 3.1.0 (RC1) ○ 3.1.0 (M2)
○ 3.0.7 (SNAPSHOT) ● 3.0.6 ○ 2.7.12 (SNAPSHOT) ○ 2.7.11

Project Metadata

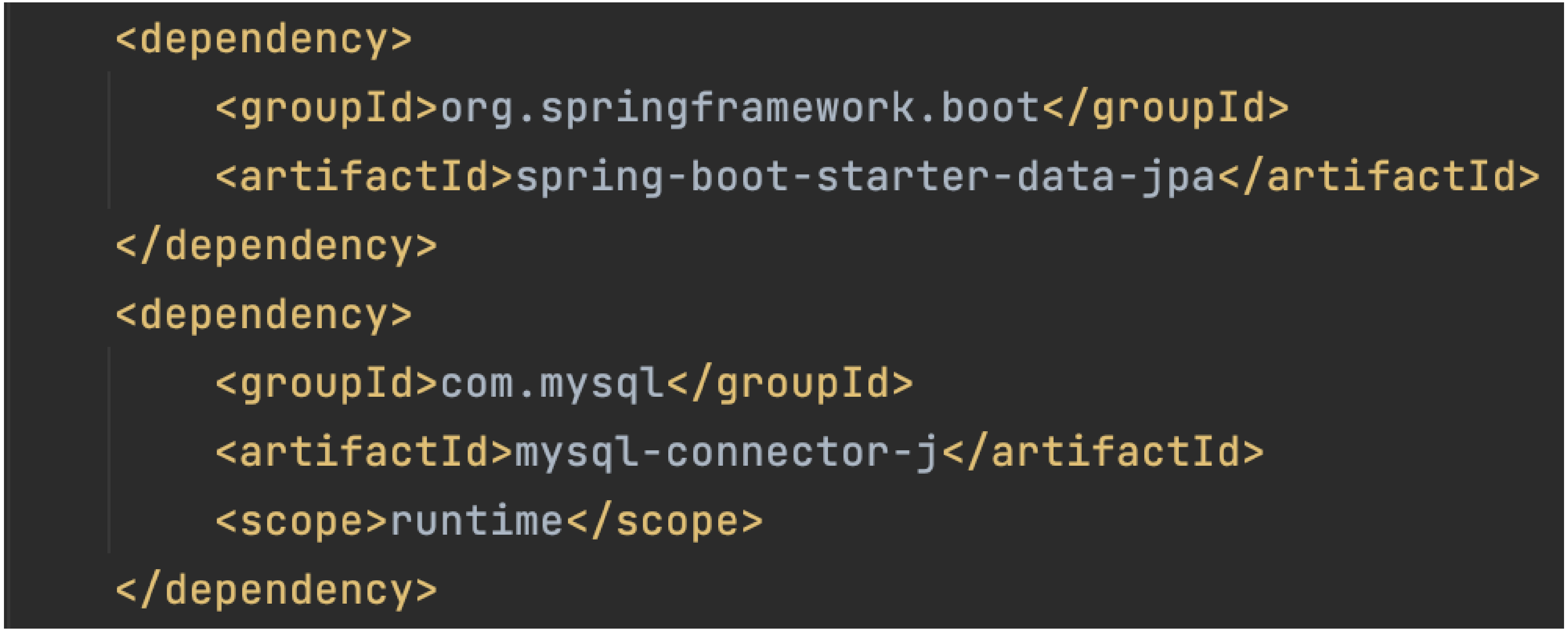
Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	● Jar ○ War

Dependencies

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

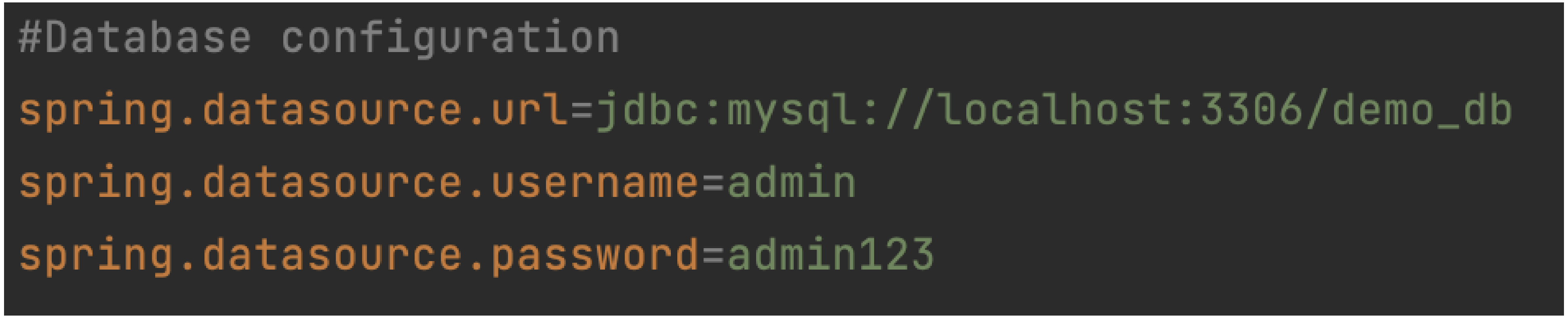
MySQL Driver SQL
MySQL JDBC driver.

Demo – Add dependency



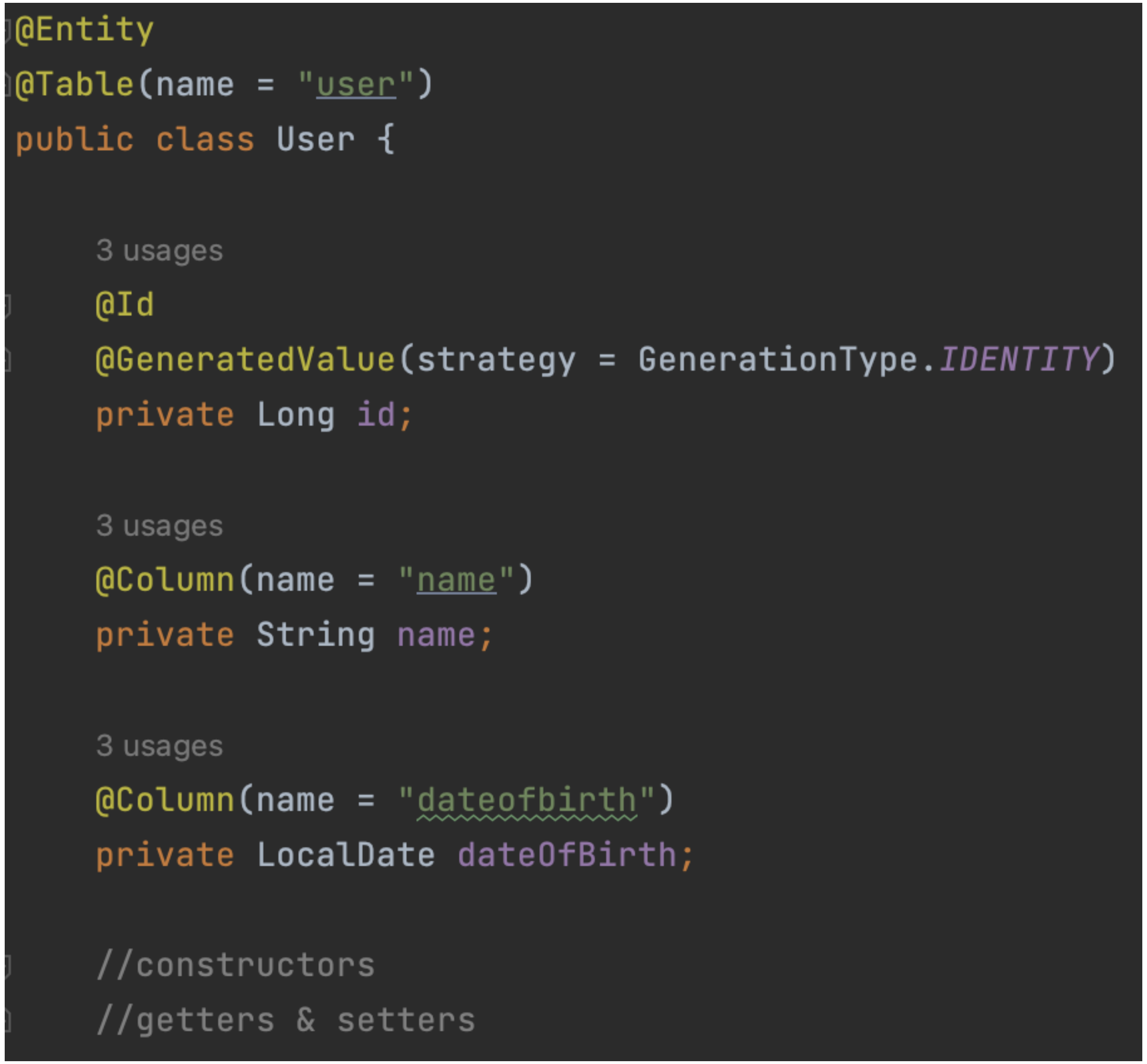
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
</dependency>
```

Demo – Configuration



```
#Database configuration
spring.datasource.url=jdbc:mysql://localhost:3306/demo_db
spring.datasource.username=admin
spring.datasource.password=admin123
```

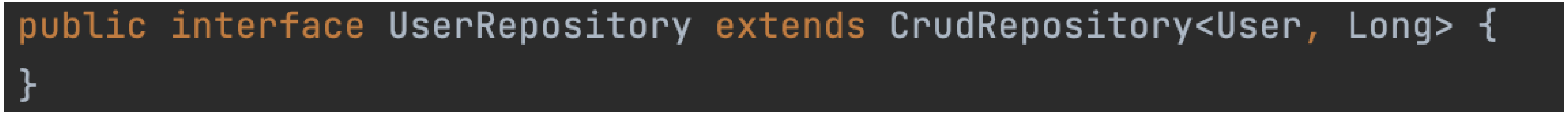
Demo – Class mapping



```
@Entity
@Table(name = "user")
public class User {

    3 usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    3 usages
    @Column(name = "name")
    private String name;

    3 usages
    @Column(name = "dateofbirth")
    private LocalDate dateOfBirth;

    //constructors
    //getters & setters
```

Demo – Repository interfaces

- CrudRepository
- PagingAndSortingRepository
- JpaRepository

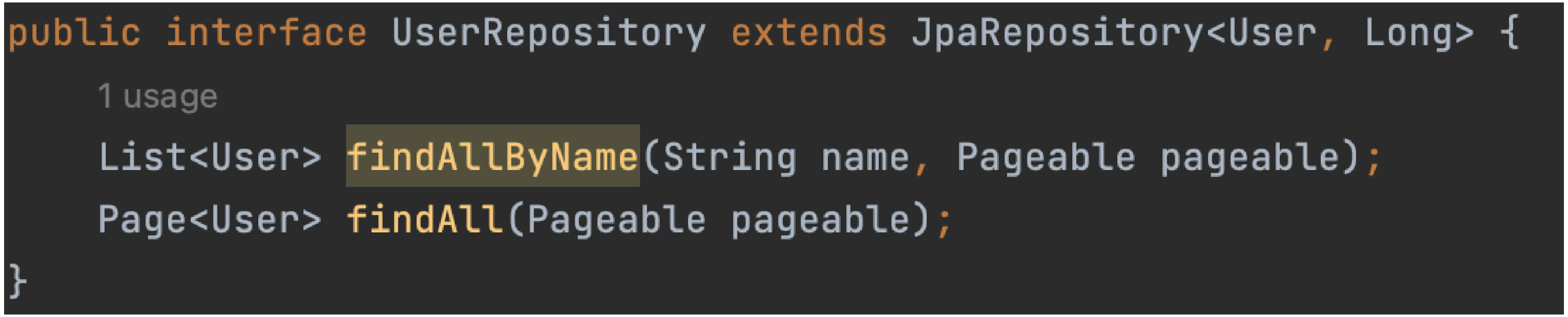


```
public interface UserRepository extends CrudRepository<User, Long> {
}
```

Demo – Create queries

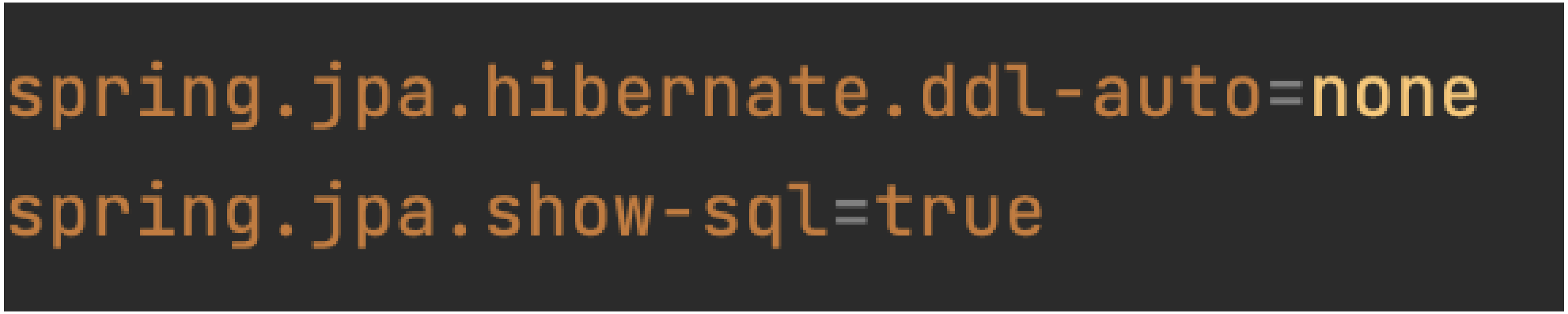
- Method name queries
- @Query annotation (JPQL or native SQL)
- Criteria API

Demo – Pagination and Sorting



```
public interface UserRepository extends JpaRepository<User, Long> {
    1 usage
    List<User> findAllByName(String name, Pageable pageable);
    Page<User> findAll(Pageable pageable);
}
```

Demo – Additional configuration



```
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
```

R2S
Resource Software Solution

*Keeping up those **inspiration** and the **enthusiasm** in the **learning path**.*
*Let confidence to bring it into **your career path** for getting gain the **success** as your expectation.*

Thank you

Contact

- Name: R2S Academy

- Email: daotao@r2s.edu.vn

- Hotline/Zalo: 0919 365 363

- Website: https://r2s.edu.vn

- Fanpage: https://www.facebook.com/r2s.tuyendung

Questions and Answers