PDP
IT-ACADEMY

Spring boot va Jpa yordamida ma'lumotlar ombori bilan ishlash

PDP
IT-ACADEMY

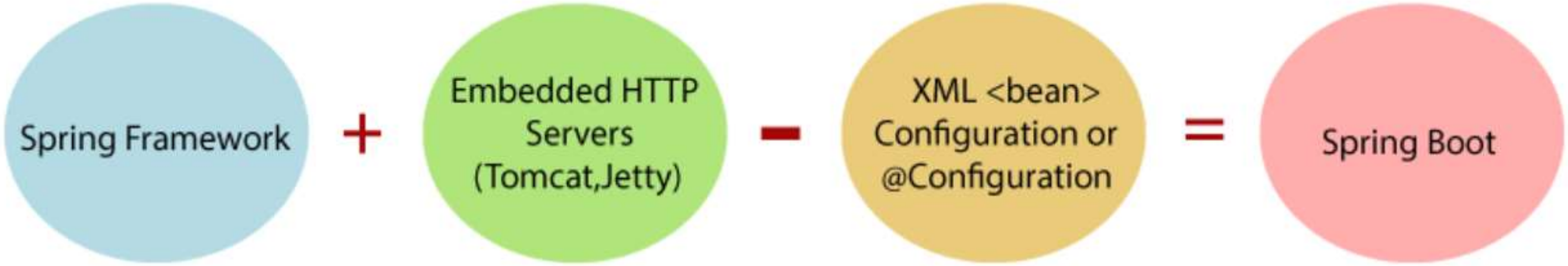
Reja:

1. Spring boot va unda loyiha ochish
2. Spring boot da HTTP so‘rovlar
3. Java Persistence Api (JPA) nima va uning afzalliklari
4. JPA da jadvallar o‘rtasidagi bog‘lanish
5. Ma’lumotlar omboriga sodda so‘rovlarni yuborish

PDP
IT–ACADEMY

Spring boot nima va unda loyiha ochish

SPRING BOOT USHBU QIYMATGA TENG

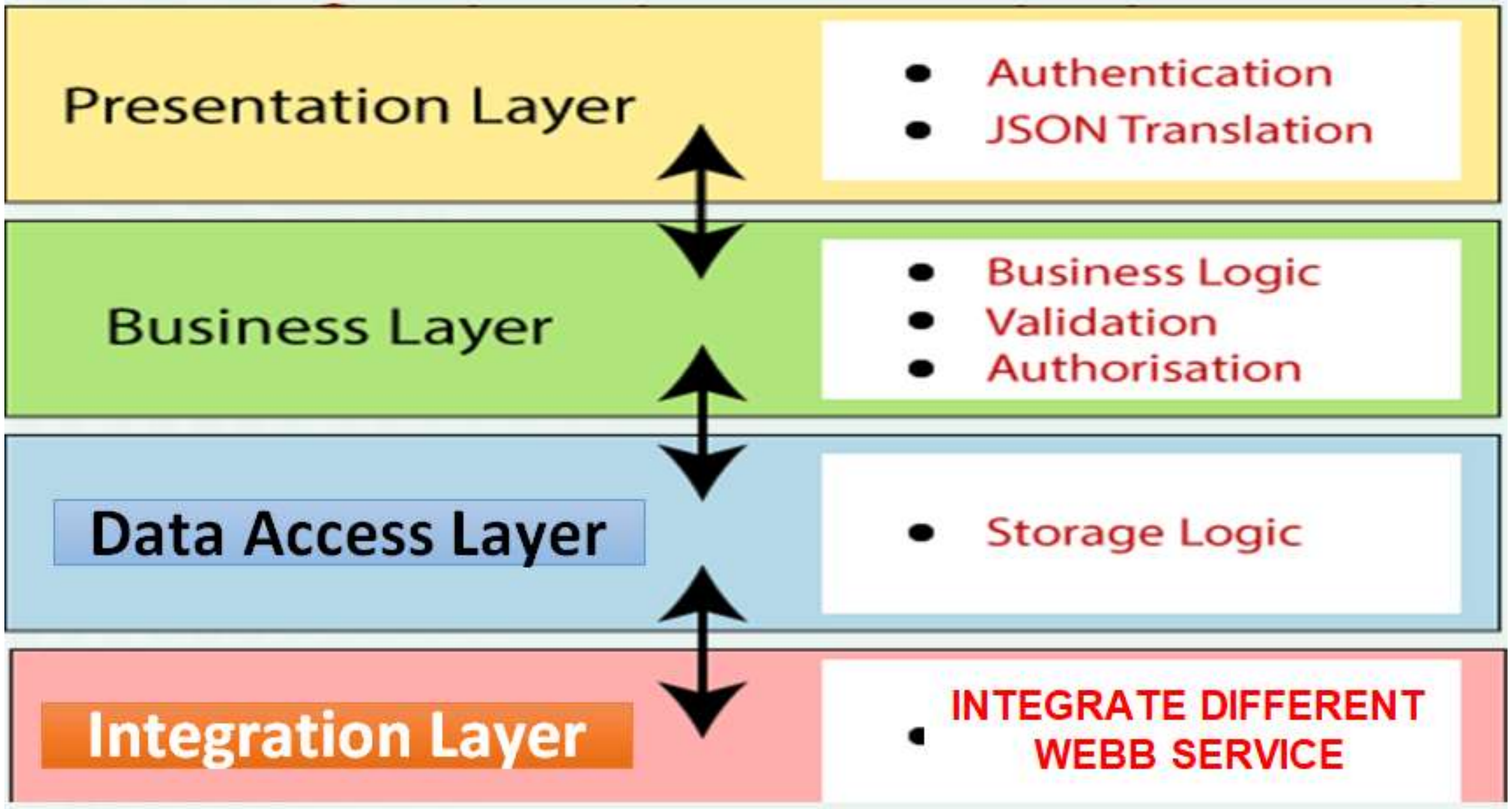


Spring Framework + Embedded HTTP Servers (Tomcat,Jetty) - XML <bean> Configuration or @Configuration = Spring Boot

Spring boot evolyutsiyasi

2014 yil aprel oyida Spring Boot 1.0, undan keyin turli xil versiyalar yaratildi.
2014 yil 1.1,
2015 yil mart oyida 1.2,
2016 yil dekabr oyida 1.3.
2017 yil yanvar oyida 1.4 va
2017 yil fevral 1.5
….
….
2021 yil yanvar 2.4.2 versiyalar

Spring boot arxitekturasi



Presentation Layer

- Authentication
- JSON Translation

Business Layer

- Business Logic
- Validation
- Authorisation

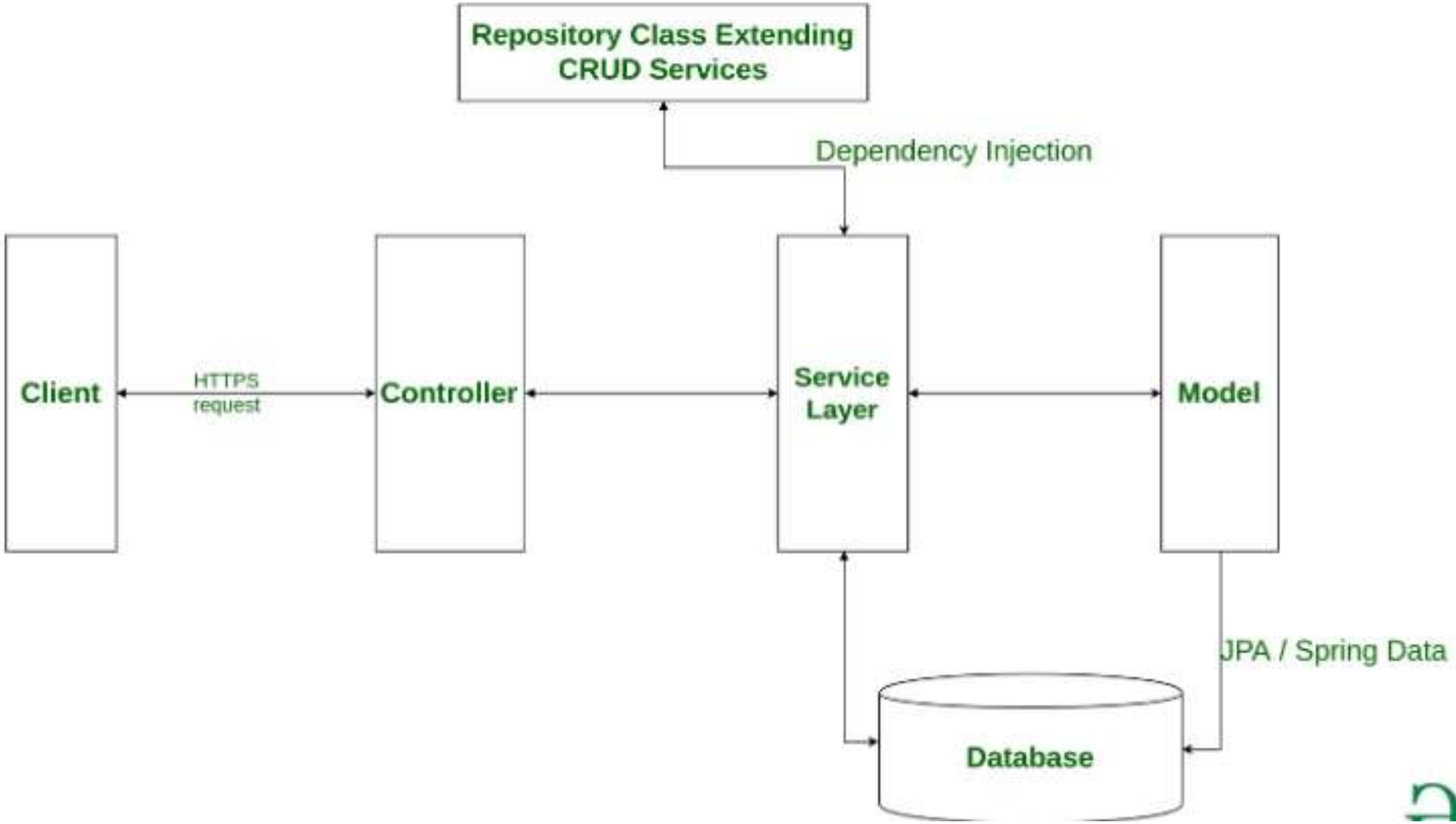
Data Access Layer

- Storage Logic

Integration Layer

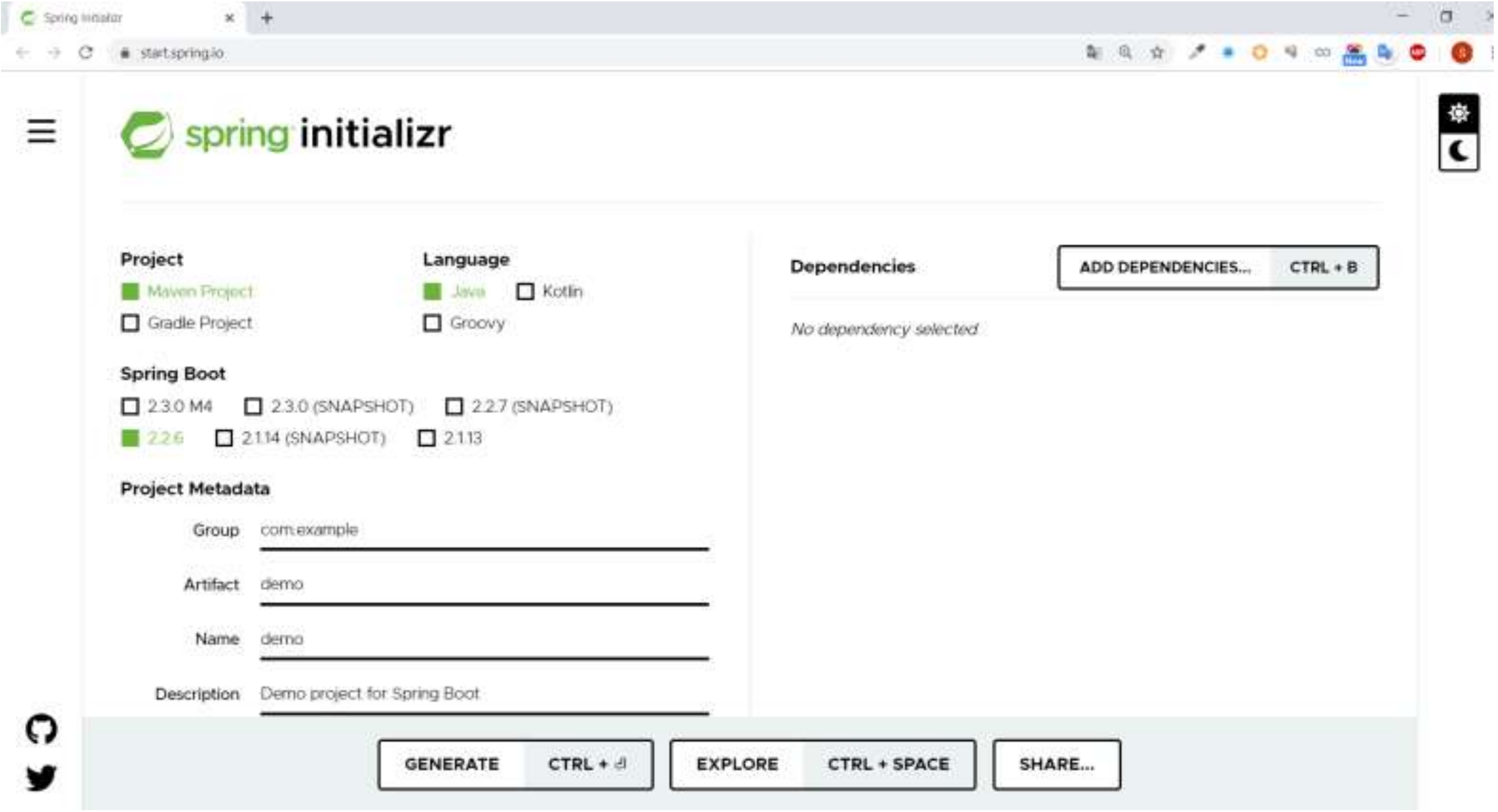
- INTEGRATE DIFFERENT WEBB SERVICE

Spring bootning ishlash



PDP
IT-ACADEMY

Spring bootda proyekt ochish 1-usul



Project

- ■ Maven Project
- □ Gradle Project

Language

- ■ Java
- □ Kotlin
- □ Groovy

Spring Boot

□ 2.3.0 M4 □ 2.3.0 (SNAPSHOT) □ 2.2.7 (SNAPSHOT)
■ 2.2.6 □ 2.1.14 (SNAPSHOT) □ 2.1.13

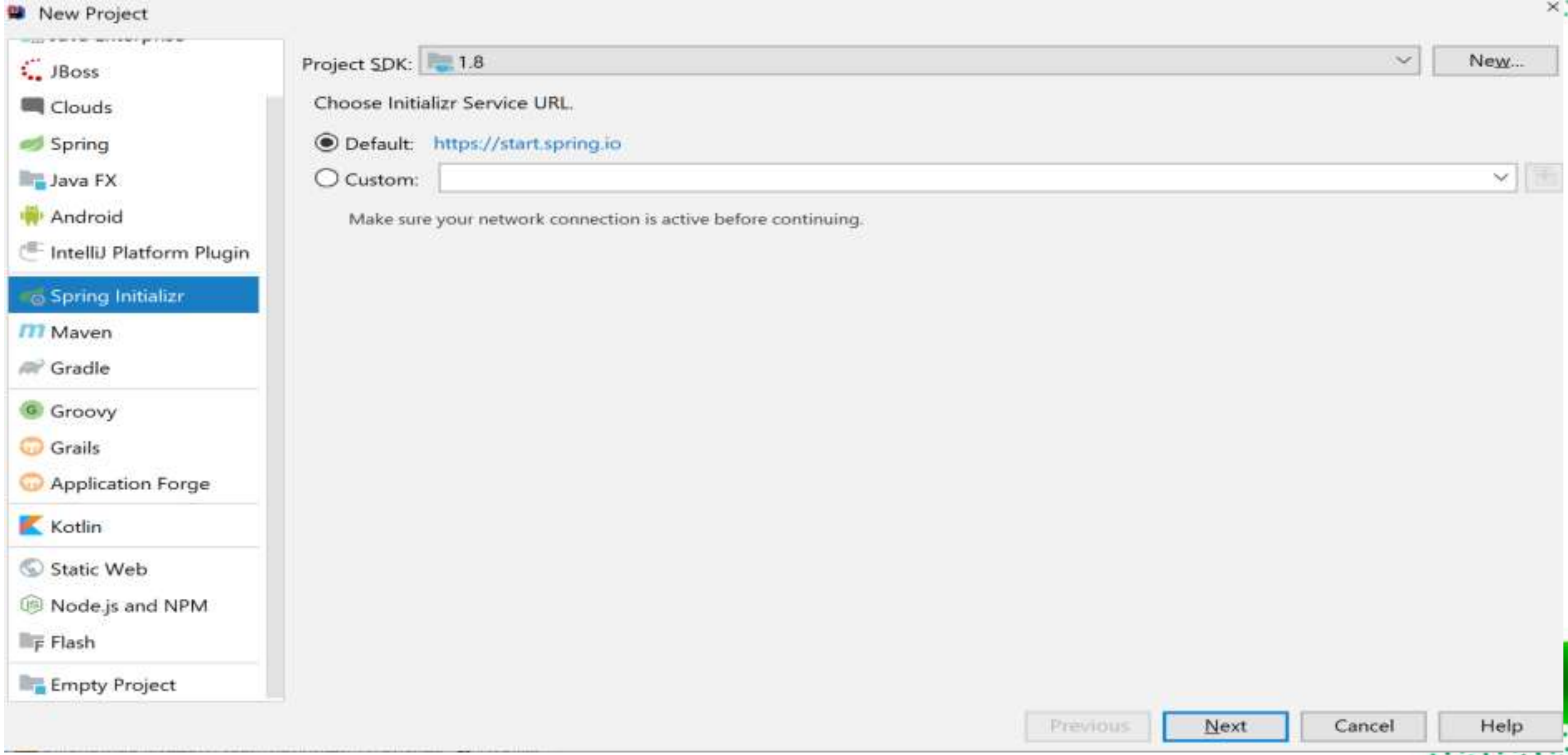
Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot

Dependencies

No dependency selected

Spring bootda proyekt ochish 2-usul (1-qadam)



New Project

Project SDK: 1.8

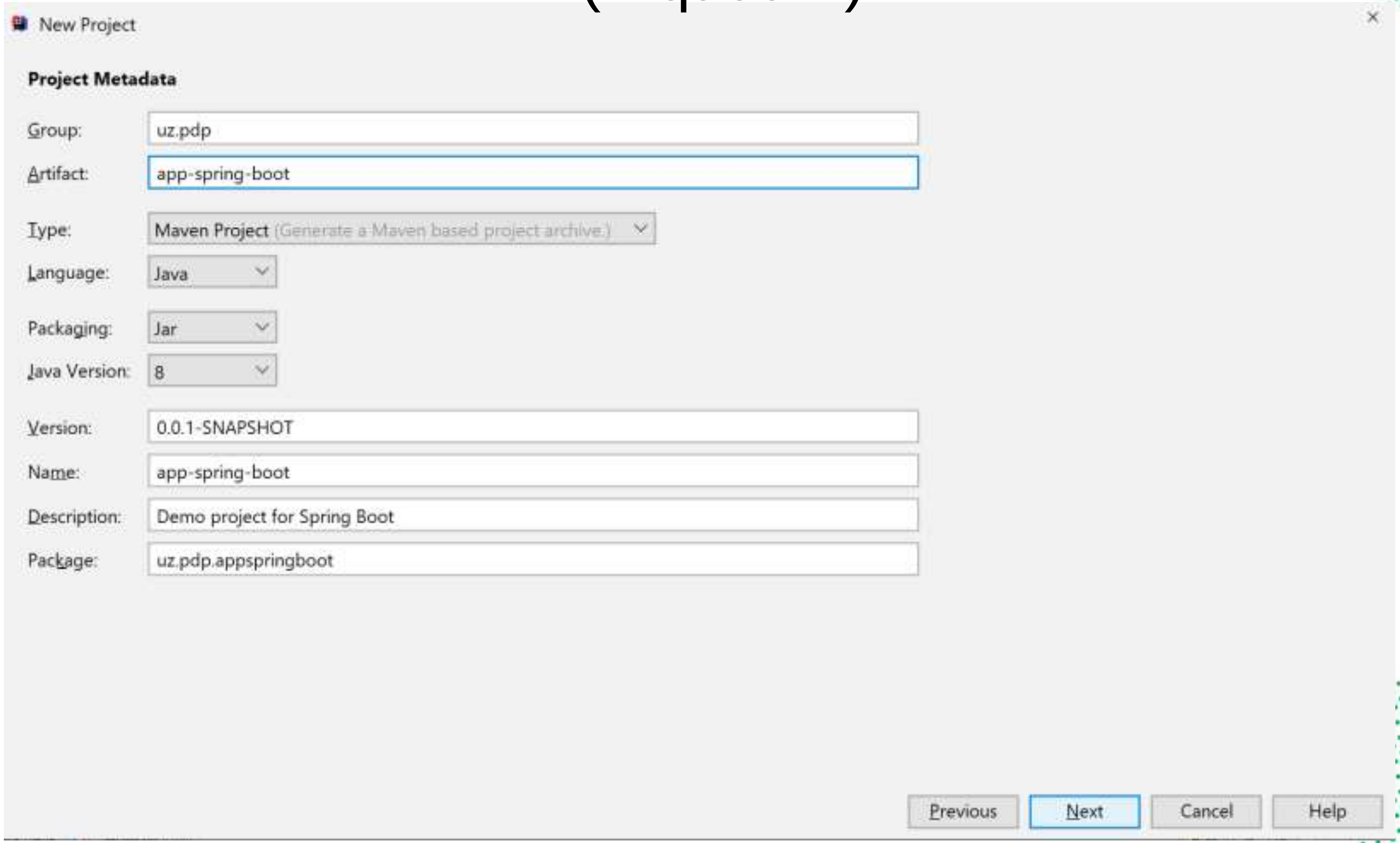
Choose Initializr Service URL.

● Default: https://start.spring.io

○ Custom:

Make sure your network connection is active before continuing.

Spring bootda proyekt ochish 2-usul (2-qadam)



New Project

Project Metadata

Group: uz.pdp

Artifact: app-spring-boot

Type: Maven Project (Generate a Maven based project archive.)

Language: Java

Packaging: Jar

Java Version: 8

Version: 0.0.1-SNAPSHOT

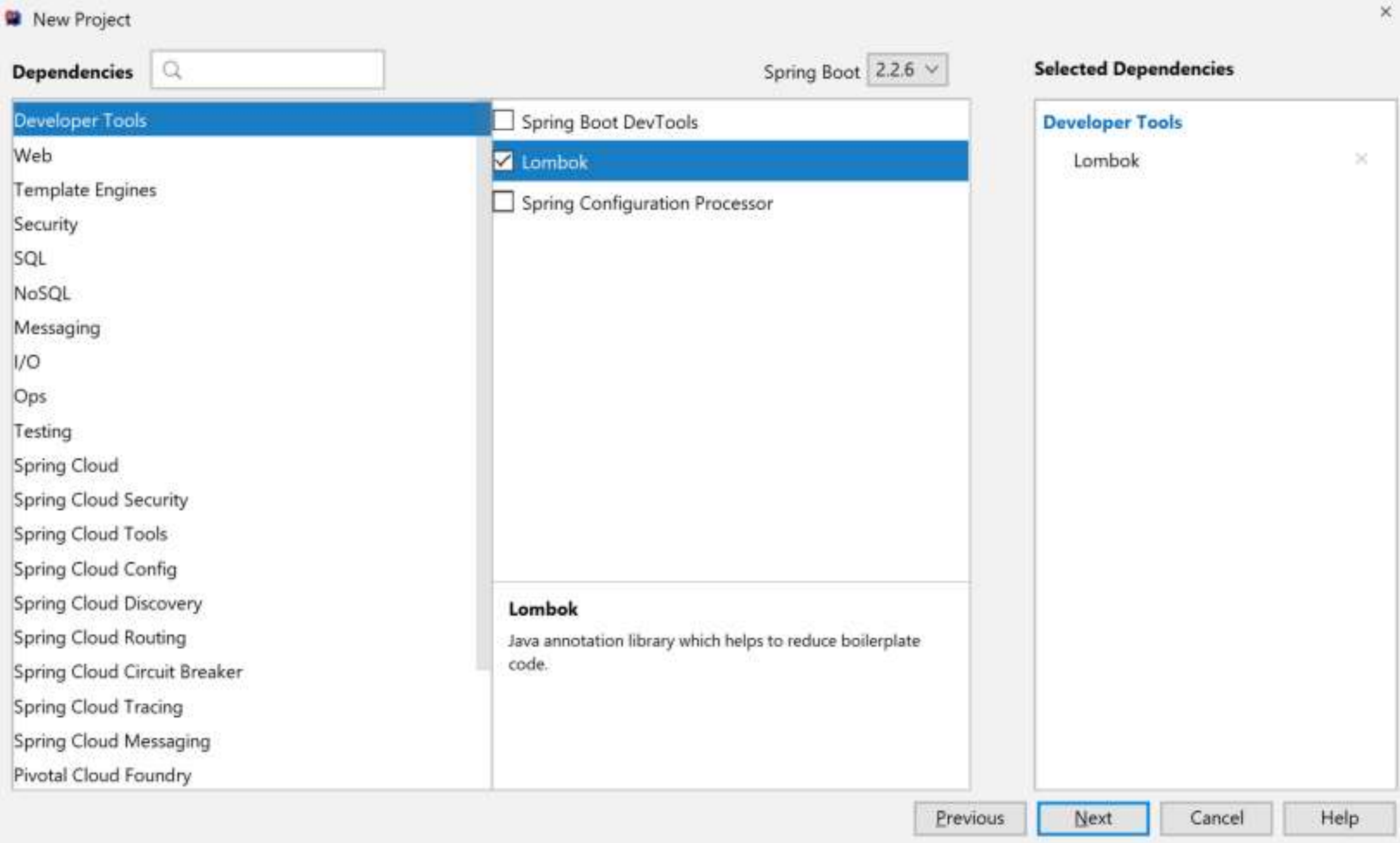
Name: app-spring-boot

Description: Demo project for Spring Boot

Package: uz.pdp.appspringboot

PDP IT-ACADEMY

Spring bootda proyekt ochish 2-usul (3-qadam)



New Project

Dependencies

Spring Boot 2.2.6

Selected Dependencies

- Developer Tools
- Web
- Template Engines
- Security
- SQL
- NoSQL
- Messaging
- I/O
- Ops
- Testing
- Spring Cloud
- Spring Cloud Security
- Spring Cloud Tools
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Routing
- Spring Cloud Circuit Breaker
- Spring Cloud Tracing
- Spring Cloud Messaging
- Pivotal Cloud Foundry

☐ Spring Boot DevTools

☑ Lombok

☐ Spring Configuration Processor

Lombok

Java annotation library which helps to reduce boilerplate code.

Developer Tools

Lombok

PDP
IT-ACADEMY

ANOTATSIYALAR

Bizning doimiy dasturlarimizda:

- so’rovlarni tutuvchi (@Controller, @RestController)
- xizmat ko’rsatuvchi va biznes jarayonlarni boshqaruvchi(@Service),
- dastur sozlamalari (@Configuration)
- ma’lumotlar ombori bilan bog’lanuvchi(@Repository)

PDP
IT-ACADEMY

@Component

@Component - Ushbu annotatsiya classni ustiga qo’yiladi.

@Component annotatsiya Java classni bean yoki komponent sifatida belgilaydi va uni dastur kontekstiga qo'shishi mumkin.

PDP
IT-ACADEMY

@Repository

@Repository ning vazifasi ham aslida @Component kabi bean hosil qilish. Shu bilan birga ma’lumotlar omboriga bog’lanishlar va ularda chiqqan xatoliklarni qaytarish uchun ishatiladi.

@Service

@Service – classni beanni ekanligini e’lon qilish uchun va ushbu class dasturdagi biznes jarayonlarni hal qilishi uchun xizmat qilishini belgilab ketamiz

@Controller & @RestController

@Controller - client tomondan kelgan so’rovlarni tutish uchun xizmat qiladi. Agar biz so’rovga berilayotgan javobning tanasida ma’lumot berib yubormoqchi bo’lsak, @ResponseBody ni qo’yishimiz kerak

@RestController - annotatsiya RESTful veb-xizmatlarini yaratishni soddalashtirish uchun Spring 4.0 da joriy qilingan.
Bu **@Controller** va **@ResponseBody** ni birlashtirilgan holati.

PDP IT-ACADEMY

@Configuration

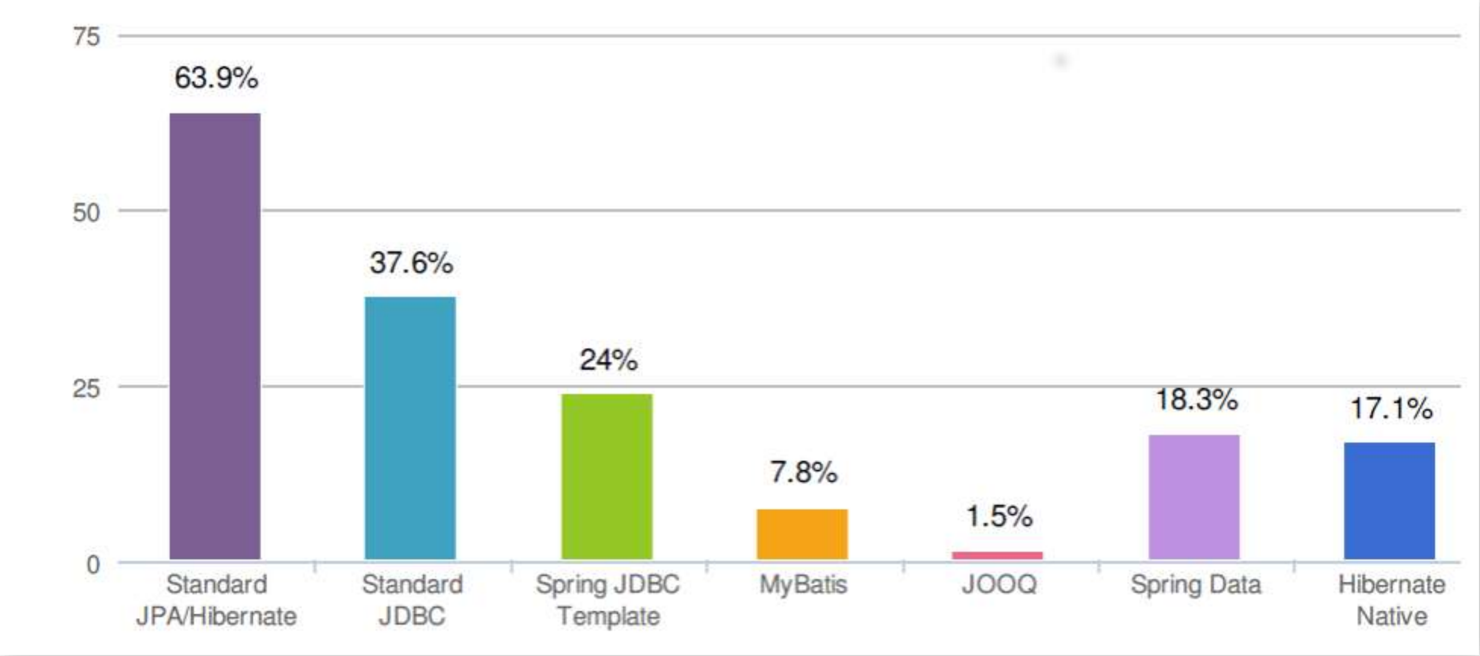
@Configuration - annotatsiya springning asosiy qismidir. Spring Configuration annotatsiya classni @Bean qilish uchun kerak.

Spring konteyner klassni qayta ishlashi va dasturda ishlatilishi uchun spring bean yaratishi mumkin.

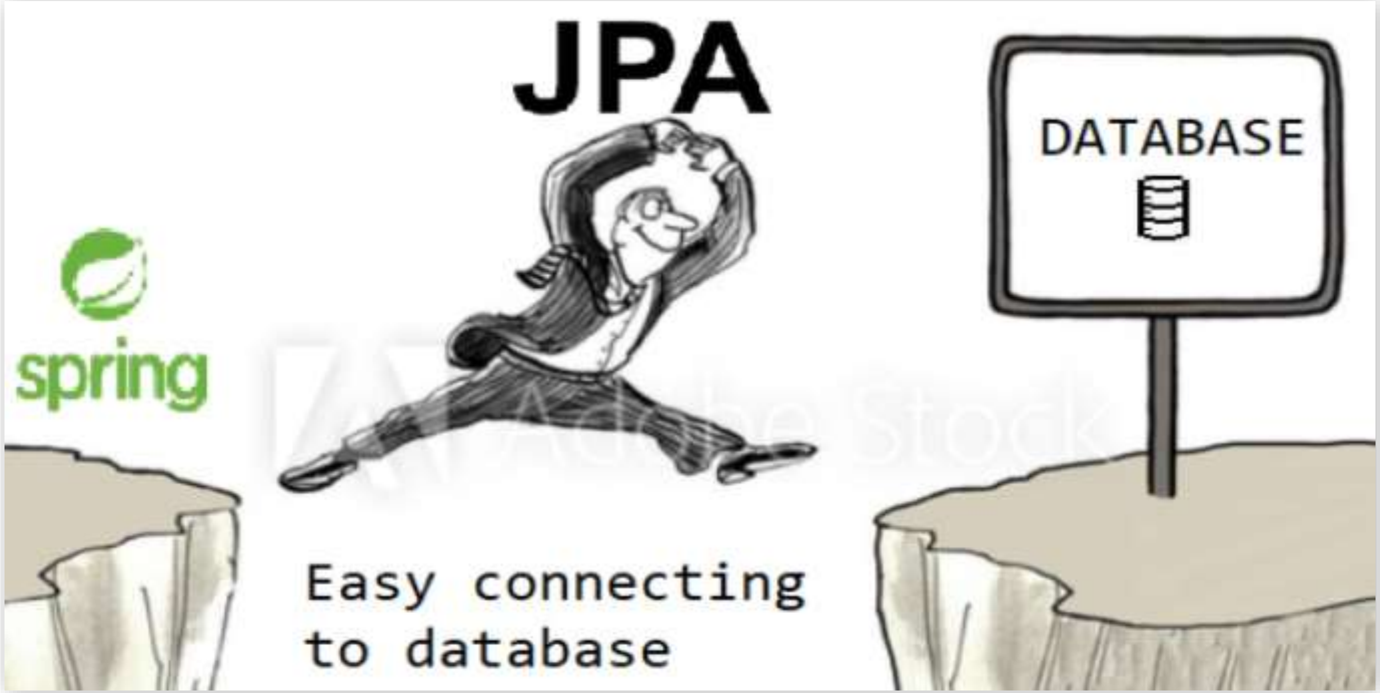
PDP
IT-ACADEMY

Spring JPA

JPA – ma’lumotlar bazasi bilan ishlashni soddalashtiruvchi vosita

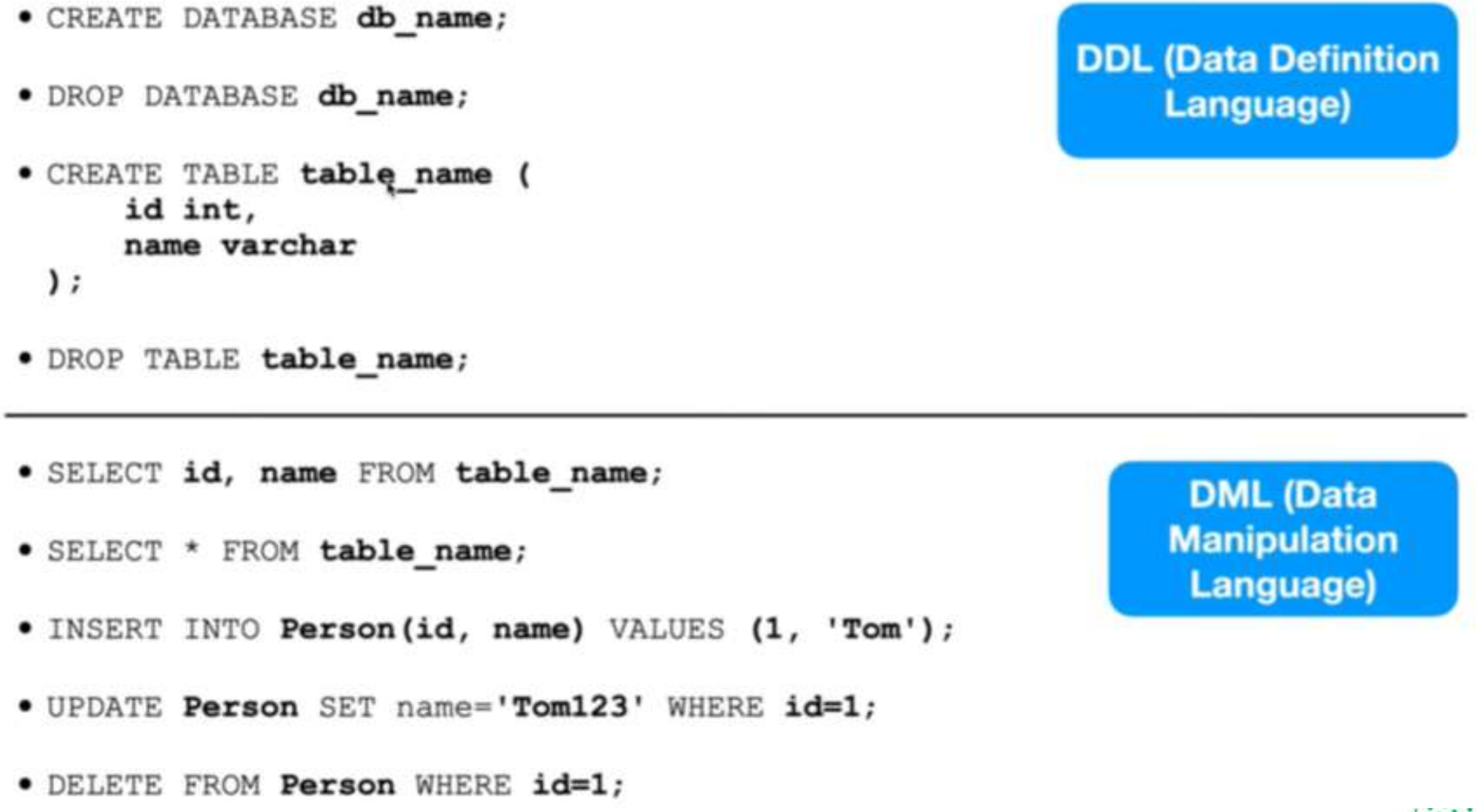


Ma'lumotlar bazasiga oson bog'lanish



SQL (Structured Query Language)

- DDL (Data Definition Language) - **CREATE, ALTER, DROP**
- DML (Data Manipulation Language) - **SELECT, INSERT, UPDATE, DELETE**
- DCL (Data Control Language) - **GRANT, REVOKE**
- TCL (Transaction Control Language) - **COMMIT, ROLLBACK**



- `CREATE DATABASE db_name;`

- `DROP DATABASE db_name;`

-
  ```
  CREATE TABLE table_name (
        id int,
        name varchar
  );
  ```

- `DROP TABLE table_name;`

DDL (Data Definition Language)

- `SELECT id, name FROM table_name;`

- `SELECT * FROM table_name;`

- `INSERT INTO Person(id, name) VALUES (1, 'Tom');`

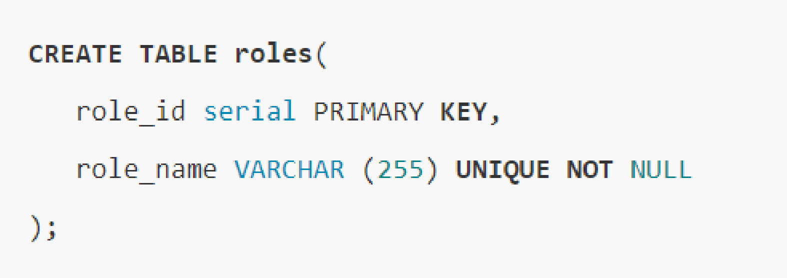
- `UPDATE Person SET name='Tom123' WHERE id=1;`

- `DELETE FROM Person WHERE id=1;`

DML (Data Manipulation Language)

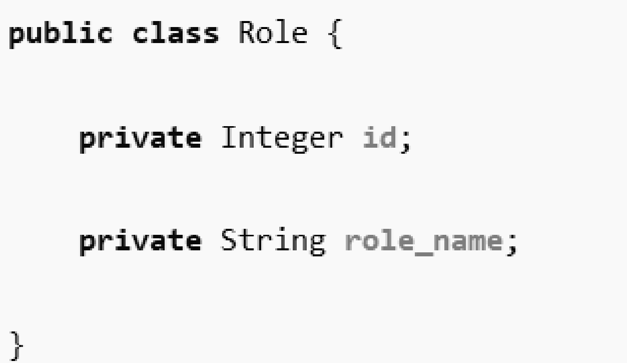
PDP
IT-ACADEMY

Tablelarni Classlar orqali boshqarish



```sql
CREATE TABLE roles(
   role_id serial PRIMARY KEY,
   role_name VARCHAR (255) UNIQUE NOT NULL
);
```

🚫



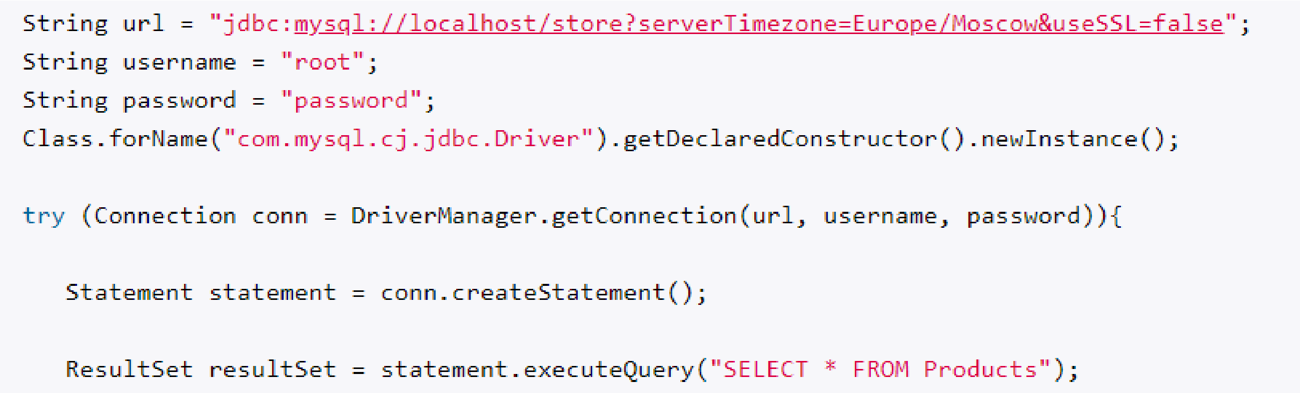
```java
public class Role {

    private Integer id;

    private String role_name;

}
```

✓

Chiroyli query stili

Query in JDBC

Jpa query



```
String url = "jdbc:mysql://localhost/store?serverTimezone=Europe/Moscow&useSSL=false";
String username = "root";
String password = "password";
Class.forName("com.mysql.cj.jdbc.Driver").getDeclaredConstructor().newInstance();

try (Connection conn = DriverManager.getConnection(url, username, password)){

   Statement statement = conn.createStatement();

   ResultSet resultSet = statement.executeQuery("SELECT * FROM Products");
```

🚫

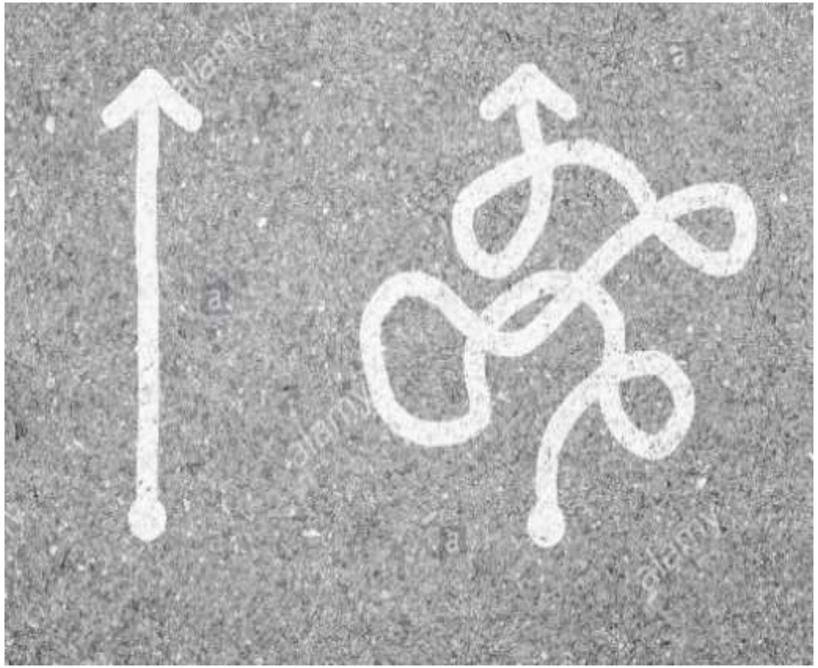
productRepository.findAll()

✓

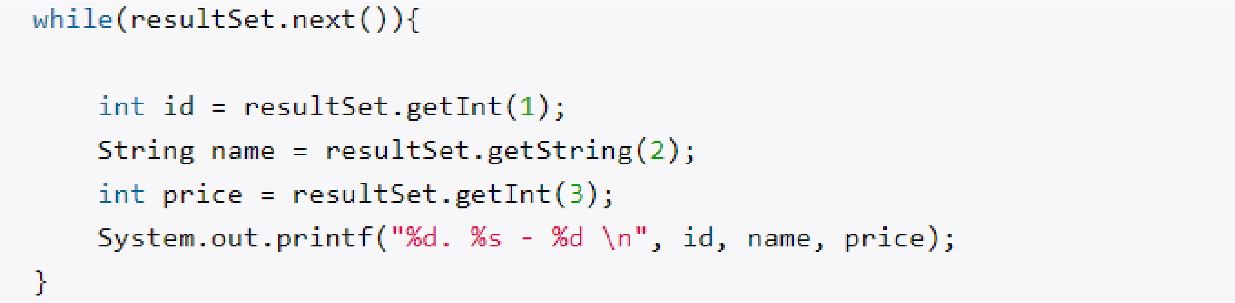
PDP
IT-ACADEMY

Query natijasini osonlik bilan olish

JPAda query natijasini olish



JDBCda query natijasini olish



```
while(resultSet.next()){

    int id = resultSet.getInt(1);
    String name = resultSet.getString(2);
    int price = resultSet.getInt(3);
    System.out.printf("%d. %s - %d \n", id, name, price);
}
```

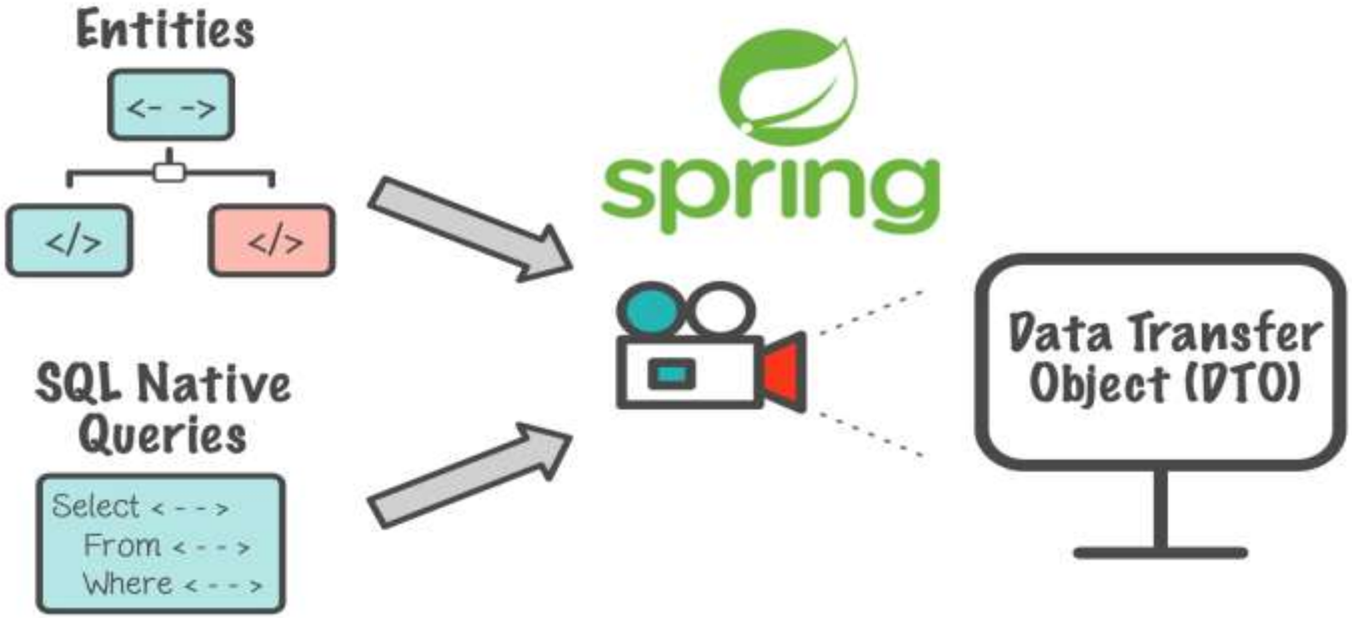
🚫

List<Product> p= productRepository.findAll()

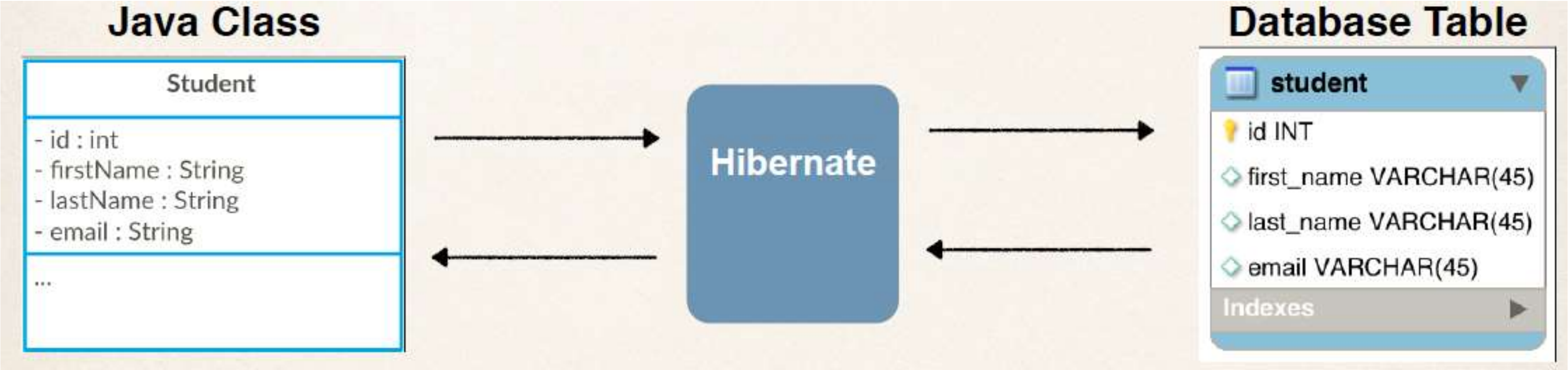
✓

PDP
IT-ACADEMY

Spring JPA bilan ishlashi prinsipi

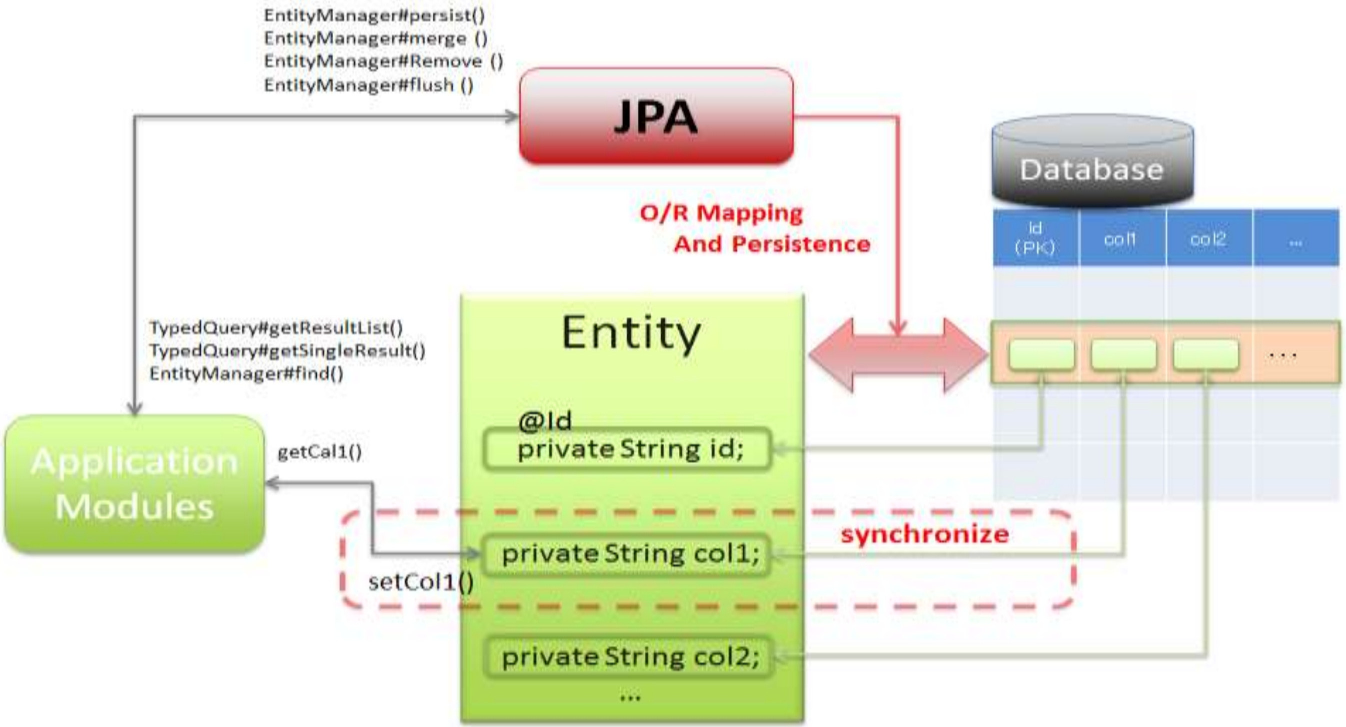


Java class ning table holati



PDP
IT-ACADEMY

Java classning qay tarzda tablega aylanishi



PDP
IT-ACADEMY

JPA da jadvallar o‘rtasida quyidagi bog‘lanish turlari mavjud

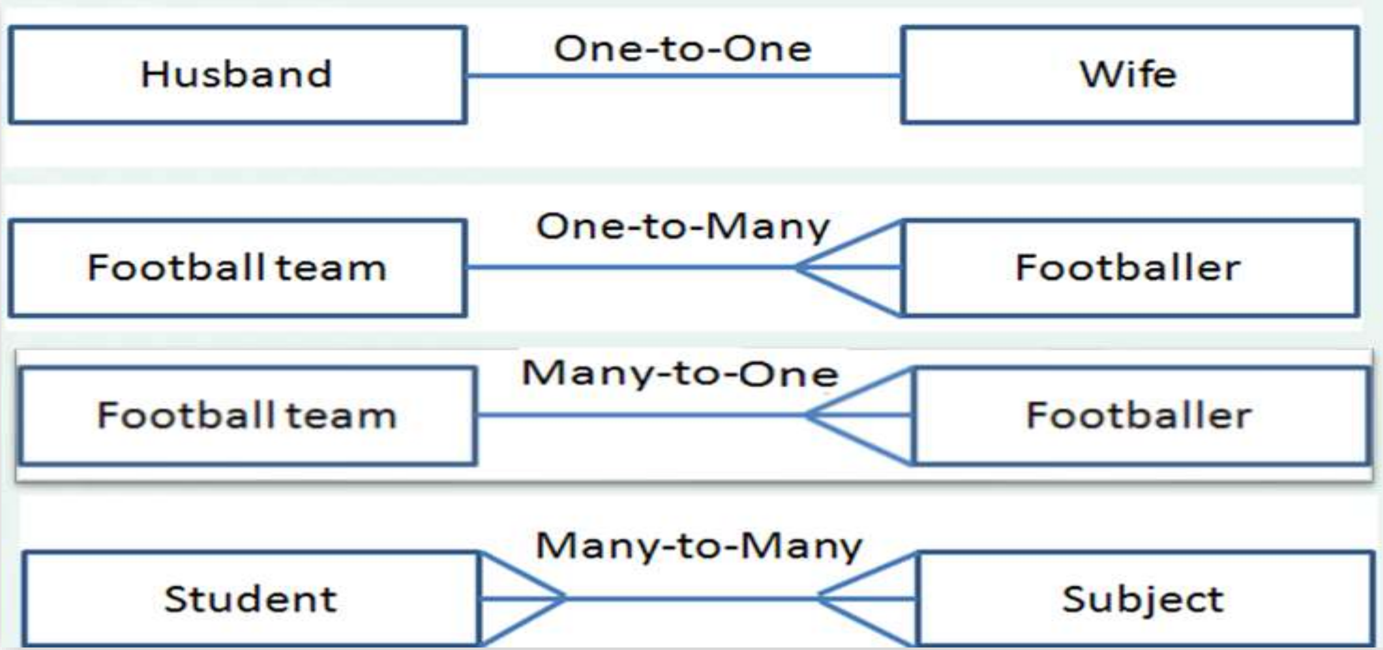
@OneToOne

@ManyToOne

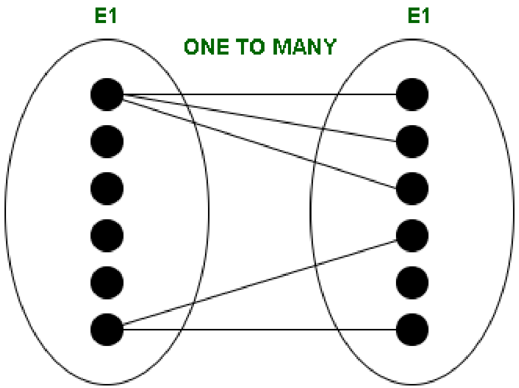
@ManyToMany

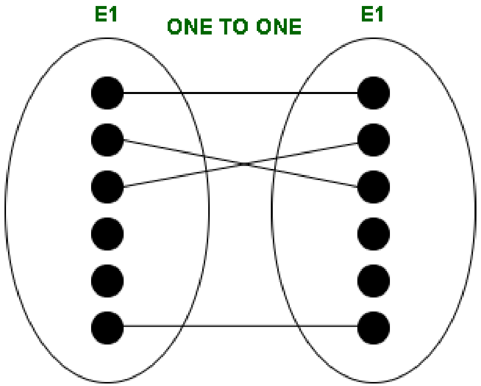
@OneToMany

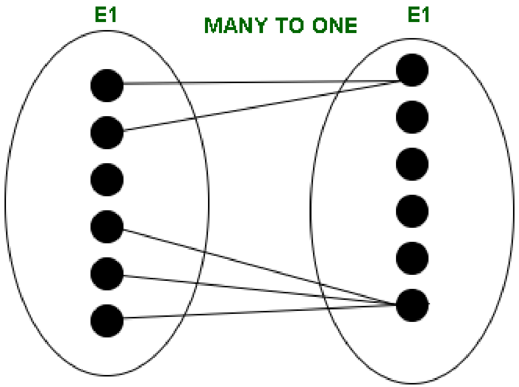
JPA da jadvallar o‘rtasida quyidagi bog‘lanish turlari mavjud

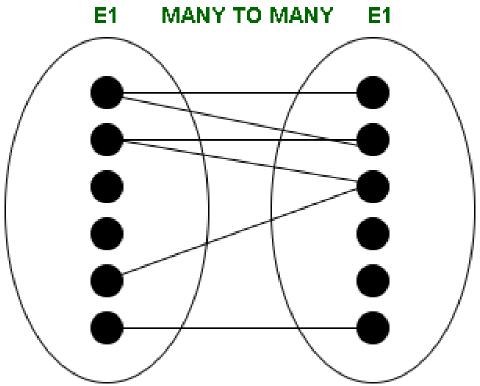


PDP
IT-ACADEMY

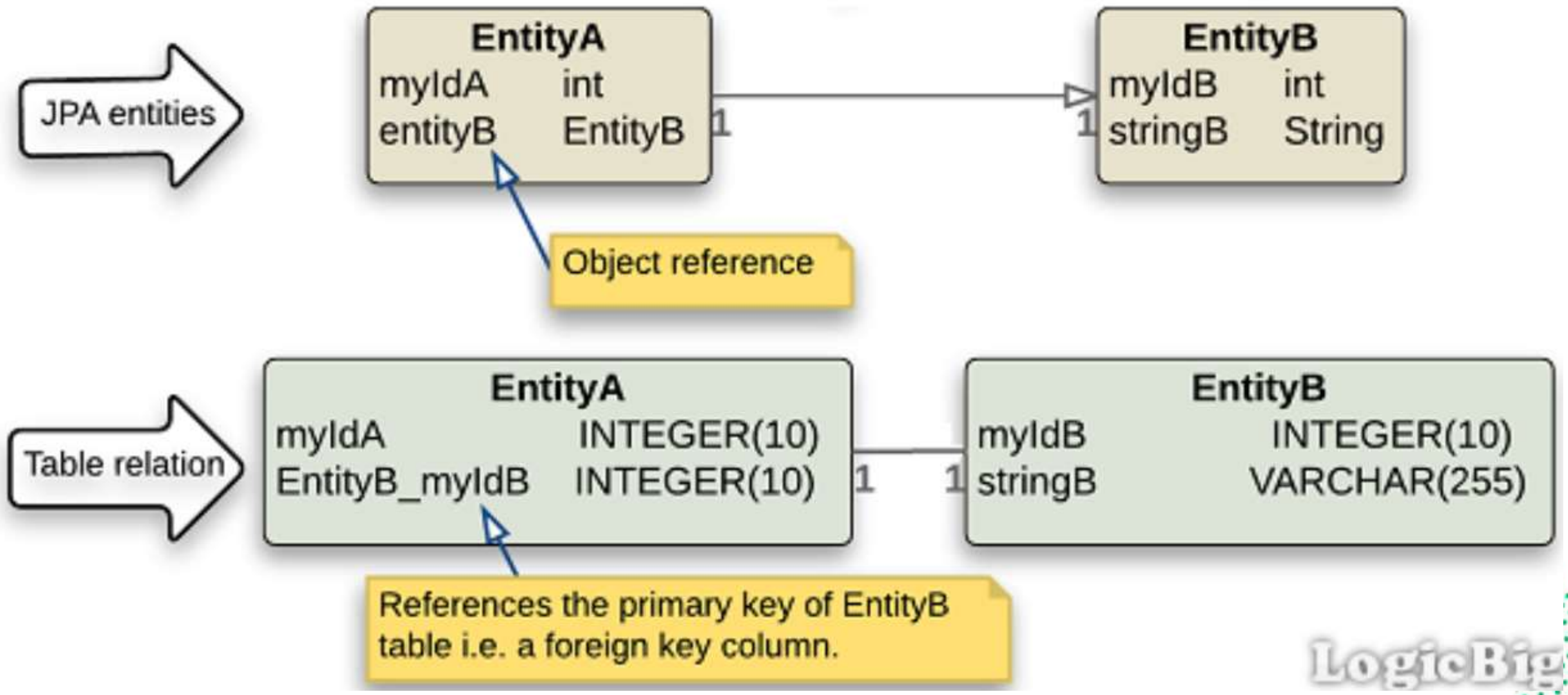








@OneToOne – birga bir bog‘lanish



JPA entities

EntityA

myIdA	int
entityB	EntityB

EntityB

myIdB	int
stringB	String

Object reference

Table relation

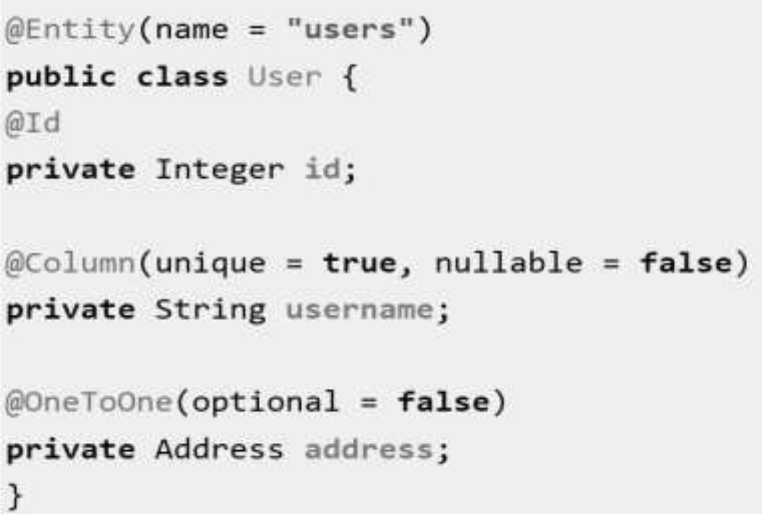
EntityA

myIdA	INTEGER(10)
EntityB_myIdB	INTEGER(10)

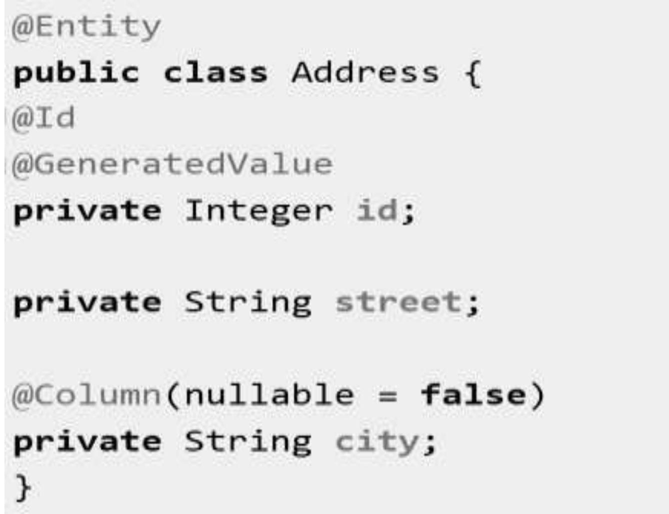
EntityB

myIdB	INTEGER(10)
stringB	VARCHAR(255)

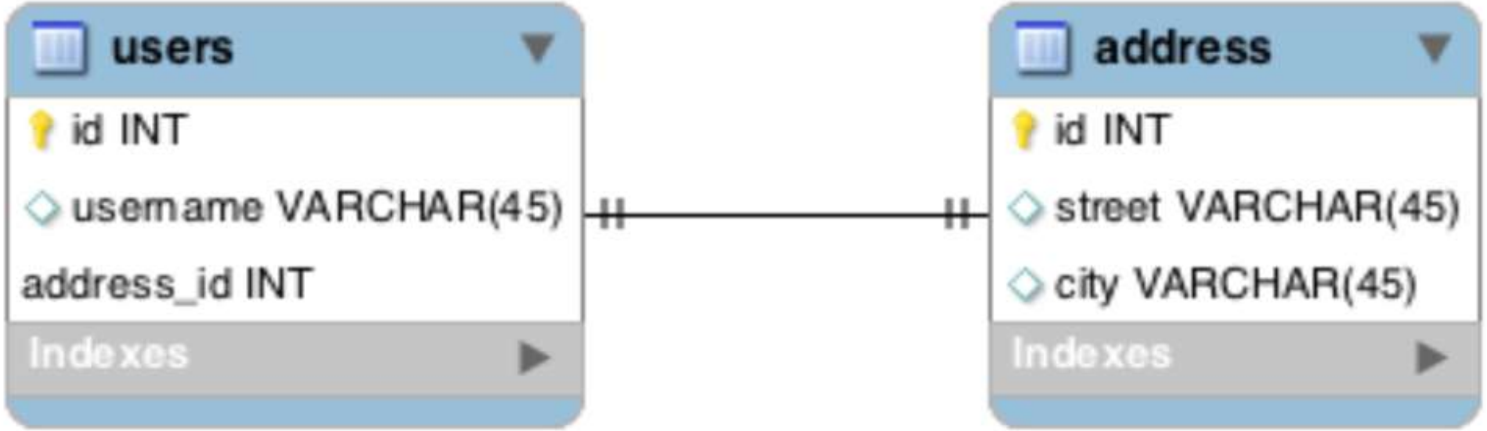
References the primary key of EntityB table i.e. a foreign key column.



```
@Entity(name = "users")
public class User {
@Id
private Integer id;

@Column(unique = true, nullable = false)
private String username;

@OneToOne(optional = false)
private Address address;
}
```



```
@Entity
public class Address {
@Id
@GeneratedValue
private Integer id;

private String street;

@Column(nullable = false)
private String city;
}
```



users

id INT

username VARCHAR(45)

address_id INT

Indexes

address

id INT

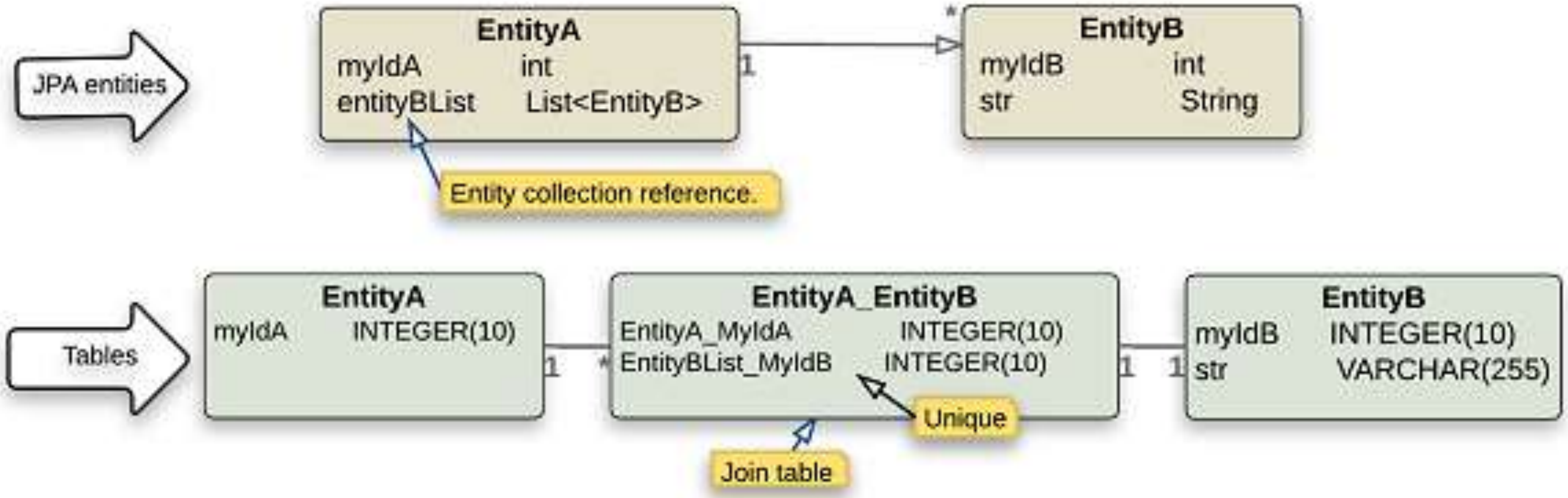
street VARCHAR(45)

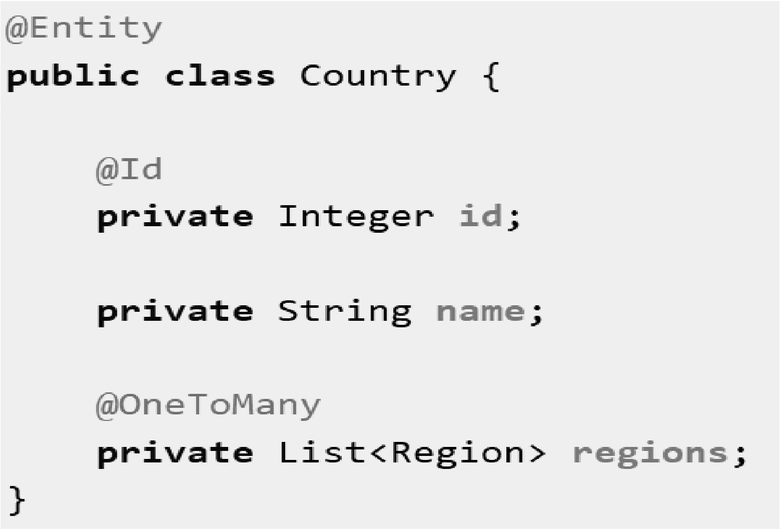
city VARCHAR(45)

Indexes

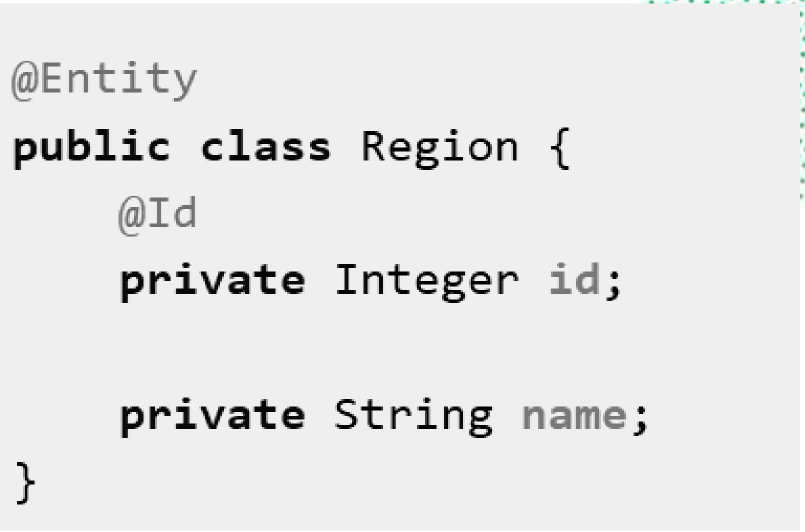
PDP
IT-ACADEMY

@OneToMany – birga ko’p bog’lanish

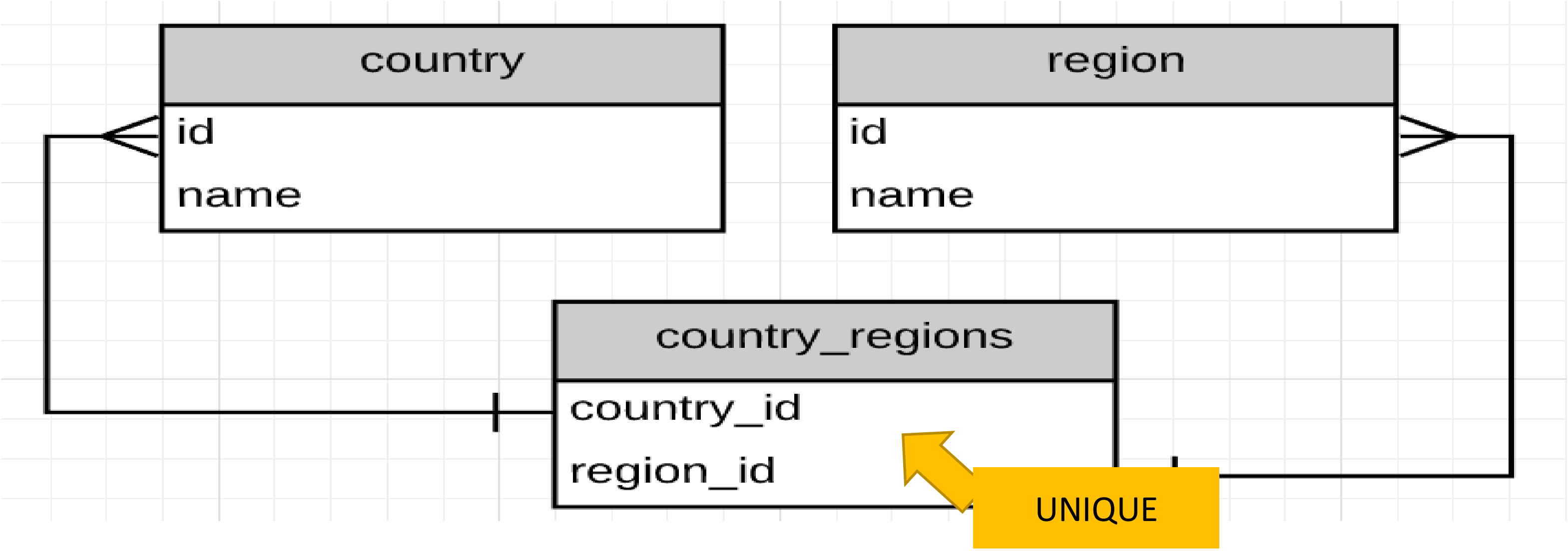




```
@Entity
public class Country {

    @Id
    private Integer id;

    private String name;

    @OneToMany
    private List<Region> regions;
}
```



```
@Entity
public class Region {
    @Id
    private Integer id;

    private String name;
}
```



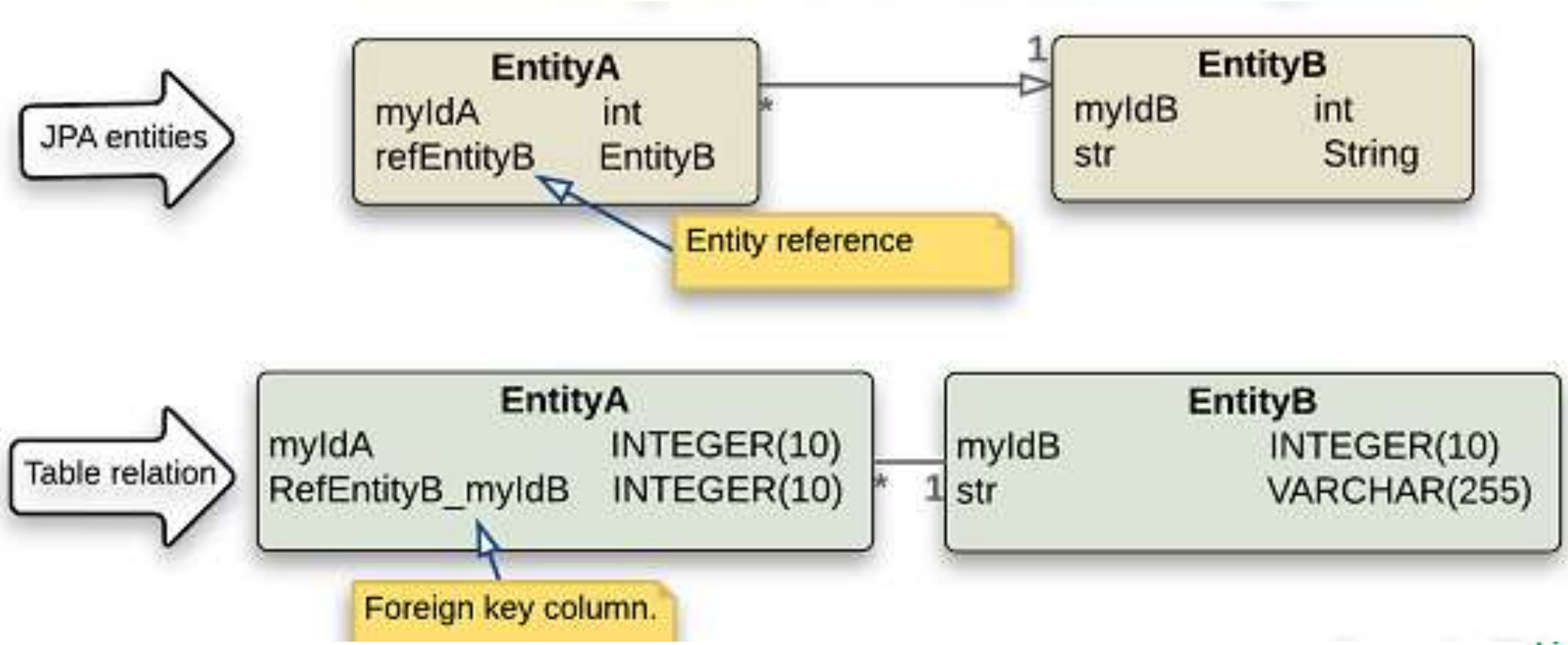
country
id
name

region
id
name

country_regions
country_id
region_id

UNIQUE

@ManyToOne – ko'pga bir bog'lanish



EntityA	
myIdA	int
refEntityB	EntityB

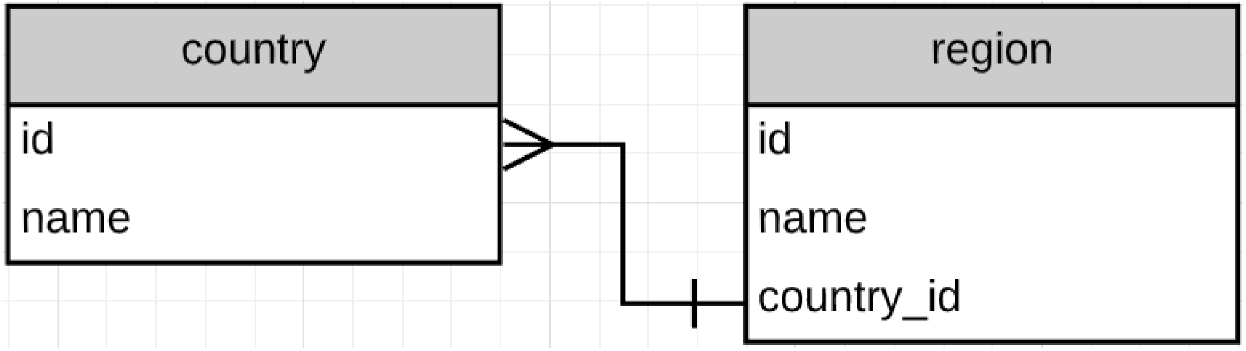
EntityB	
myIdB	int
str	String

EntityA	
myIdA	INTEGER(10)
RefEntityB_myIdB	INTEGER(10)

EntityB	
myIdB	INTEGER(10)
str	VARCHAR(255)

```
@Entity
public class Country {

    @Id
    private Integer id;

    private String name;
}
```

```
@Entity
public class Region {
    @Id
    private Integer id;

    private String name;

    @ManyToOne
    private Country country;
}
```



country

id

name

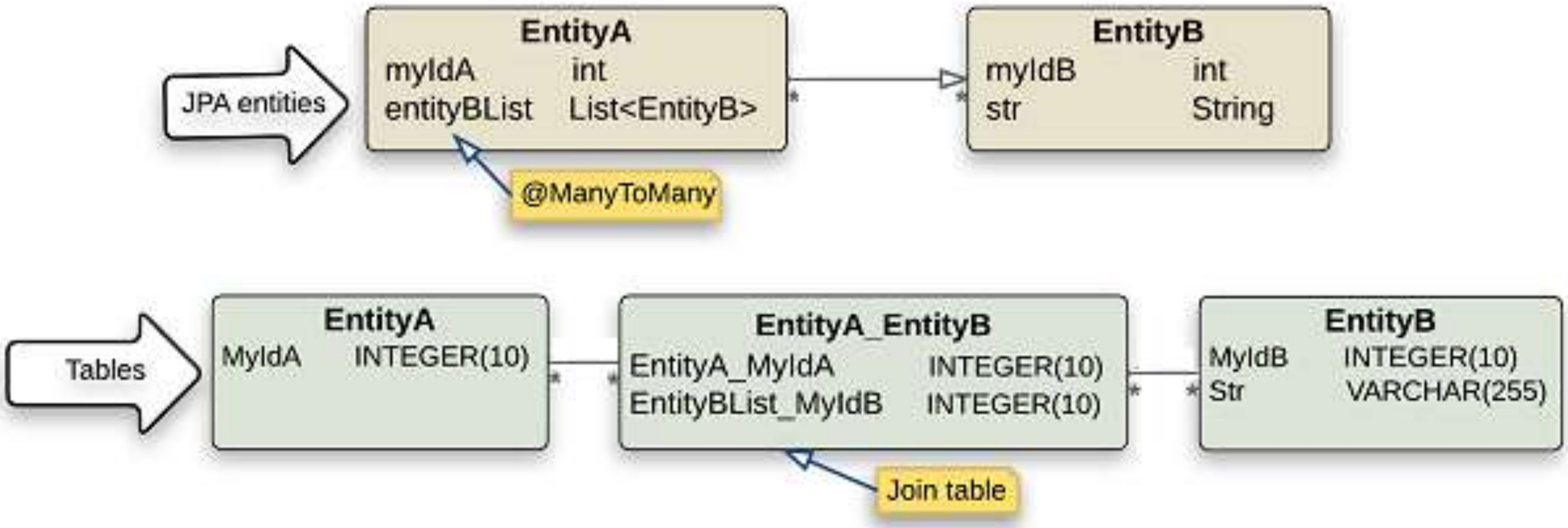
region

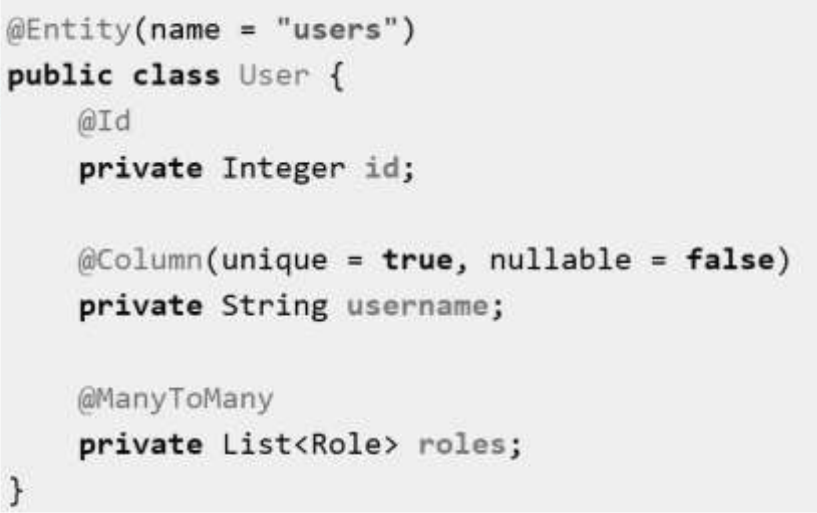
id

name

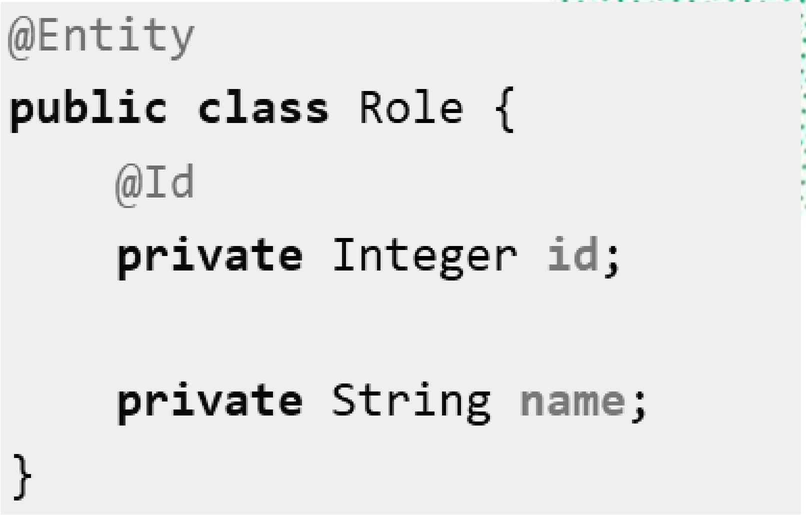
country_id

@ManyToMany – ko'pga ko'p bog'lanish

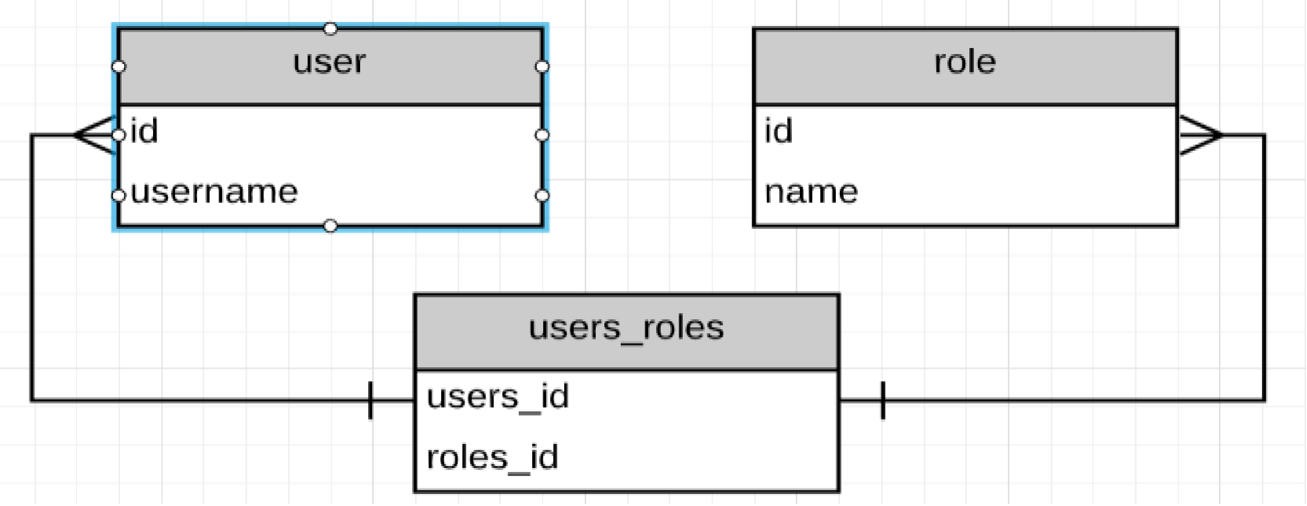




```
@Entity(name = "users")
public class User {
    @Id
    private Integer id;

    @Column(unique = true, nullable = false)
    private String username;

    @ManyToMany
    private List<Role> roles;
}
```



```
@Entity
public class Role {
    @Id
    private Integer id;

    private String name;
}
```



user

id

username

role

id

name

users_roles

users_id

roles_id

Connection configuration

```
server.port=80

spring.datasource.driver-class-name=org.postgresql.Driver

spring.datasource.url=jdbc:postgresql://localhost:5432/dbname

spring.datasource.username=username

spring.datasource.password=password

spring.jpa.hiberante.ddl-auto=create
```

PDP
IT-ACADEMY

Etiboringiz uchun rahmat.