Hibernate

2023

TECHPRO EDUCATION

Basic Annotations

Entity, Table, Id, Column etc.

Embedded Annotations

OneToMany - ManyToOne

HQL

Hibernate Query Language.

OneToOne Relation

ManyToMany Relation

Fetch Types

Lazy,Eager

Get-Load Methods

Hibernate Session

Cache Level

First Level -Second Level

Hibernate Lifecycle

States: Transient, Persistent, Detached, Removed

JPA EntityManager

Hibernate

General Questions

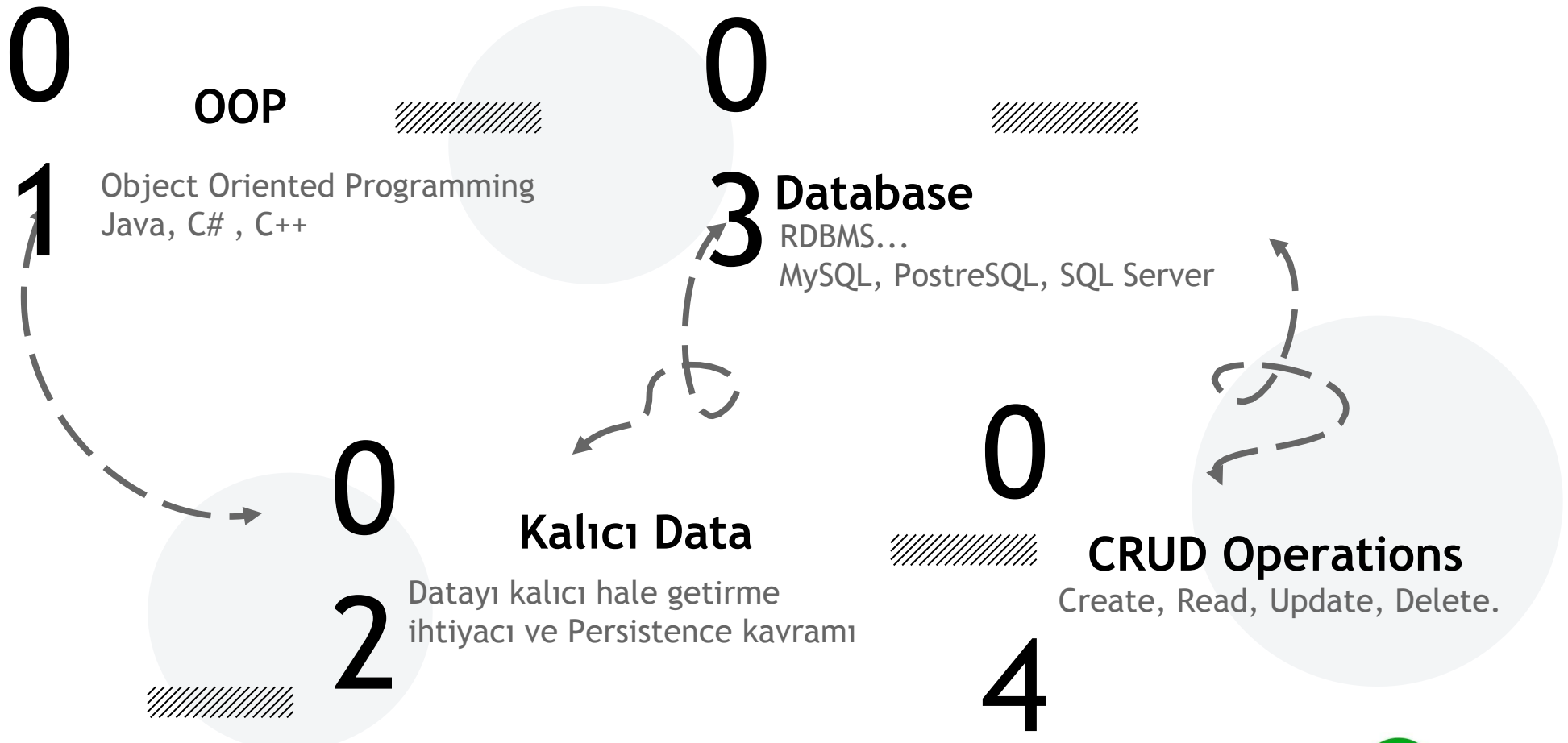
What is Persistence?

What is JDBC?

What is ORM?

What is Hibernate?

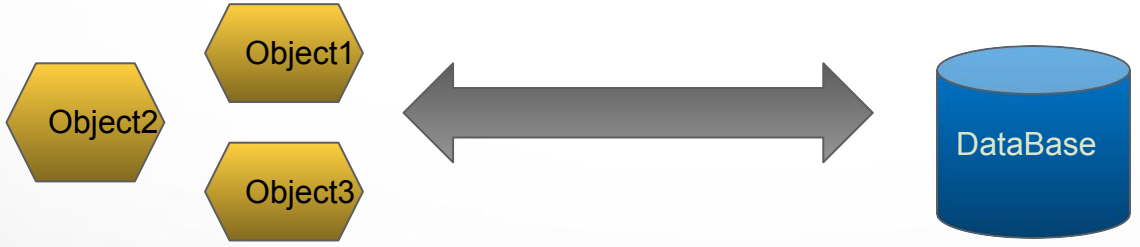
What is JPA?



TECHPRO EDUCATION

HIBERNATE

- JDBC: Java Database Connectivity API
- ORM Object Relational Mapping
  - + Kalıcı Data ile ilgili Sorunlara Çözüm
  - + JDBC BoilerPlate Kodlar
- ORM ve RDBMS , daha az SQL kodu
- ORM framework / tools: Hibernate, EclipseLink, TopLink, IBatis etc.



Classes, Variables, Objects

Tables, Fields, Records (Row)

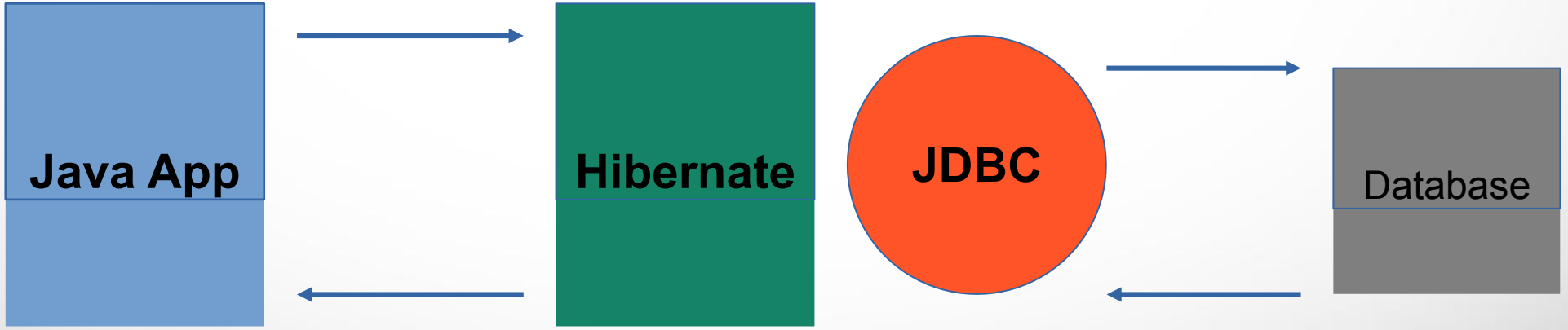
HIBERNATE - JPA

-JPA (Java Persistence API)

- JPA bir spesifikasyondur
  - Kontrolsüz ORM Frameworkleri
- Hibernate, ORM Framework ler içinde JPA yı implement eden en popüler tool dur.
- The Java Persistence API (JPA) ismini 2019 yılında Jakarta Persistence olarak değiştirdi.

HIBERNATE - JDBC

- Hibernate bütün database iletişiminde JDBC'yi kullanır
- Hibernate, JDBC'nin bir üst katmanıdır.
- Uygulamamız datayı almak ve kaydetmek için hibernate API'yi kullanır



HIBERNATE Proje Oluşturma

1. maven project

2. dependencies into pom.xml file
   - Hibernate core
   - PostgreSQL connector

3. Hibernate Configuration File

   url, username, password , etc.

4. Add Java Class and Annotate It
   - @Entity, @Table, @Id, @Column etc.

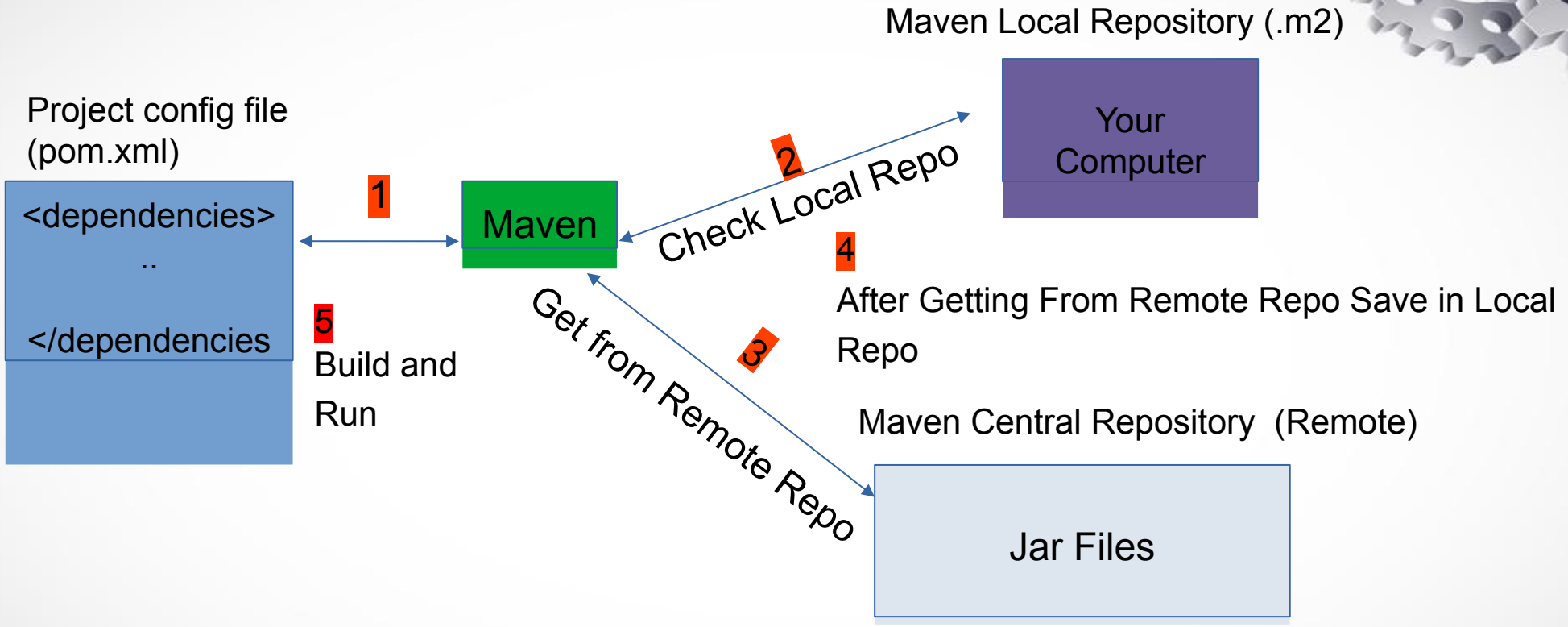
MAVEN DEPENDENCIES

- Maven dependencies

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.6.8.Final</version>
</dependency>
```

```
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.3.6</version>
</dependency>
```

MAVEN nasıl Çalışır?



MAVEN LOCAL REPOSITORY

- Local Repository for Different OS
  - ✓ MS Windows: C:\Users\your user home\.m2\ repository
  - ✓ MAC and Linux: ~/.m2/repository

@Embeddable ve @Embedded

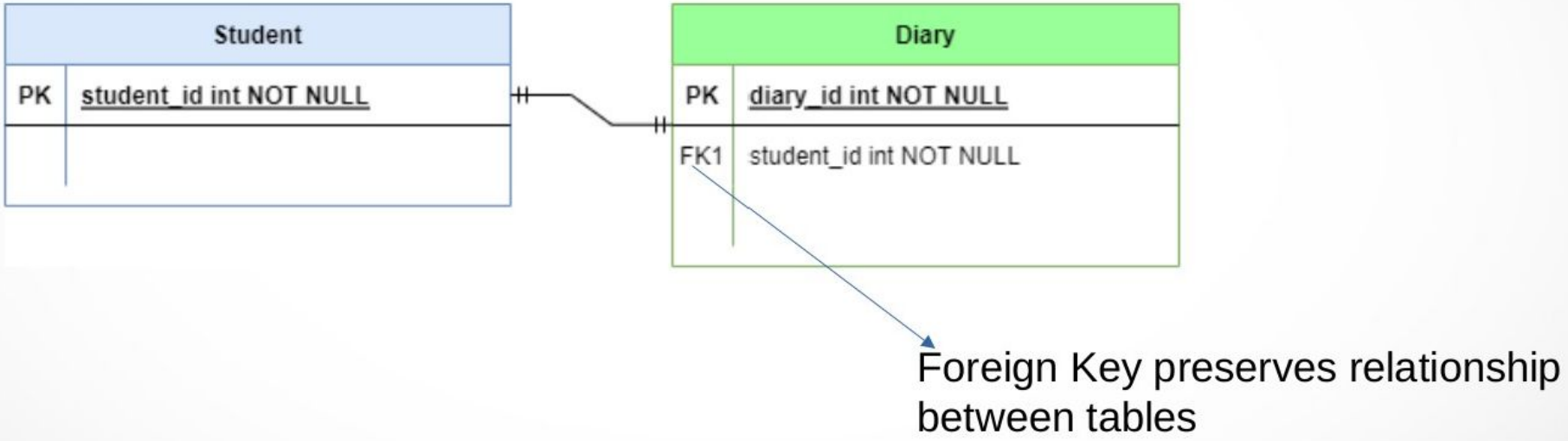
- student table

```
@Embeddable
public class Address {

    private String street;
    private String city;
    private String country;

}

@Entity
public class Student{
   @Id
   private int id
   @Embedded
   private Address address;
}
```

```
Student student=new Student();
student.setId(1);
Address address=new Address();
address.set(street);
.
.
student.setAddress(address);
```

One-to-One

- Bir Student(öğrenci) sadece bir Diary(günlük) ye sahip olabilir : Uni-directional and bi-directional



UNI-DIRECTIONAL ONE-TO-ONE MAPPING

```java
@Entity
public class Student {
@Id
private int id;
private String name;
}
```


```java
@Entity
public class Diary {
@Id
private int id;
private String name;
@OneToOne
@JoinColumn
private Student
student;
}
```

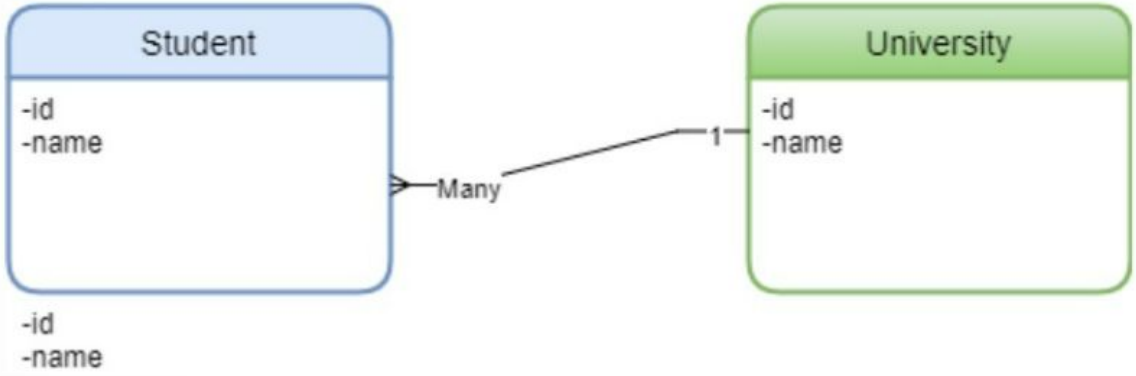
BI-DIRECTIONAL ONE-TO-ONE MAPPING

```java
@Entity
public class Student {
@Id
private int id;
private String name;

@OneToOne(mappedBy = "student")
private Diary diary;
}
```


```java
@Entity
public class Diary {
@Id
private int id;
private String name;
@OneToOne
@JoinColumn
private Student student;
}
```

Uni-Directional ManyToOne

- Bir Üniversite çok Öğrenciye sahip olabilir

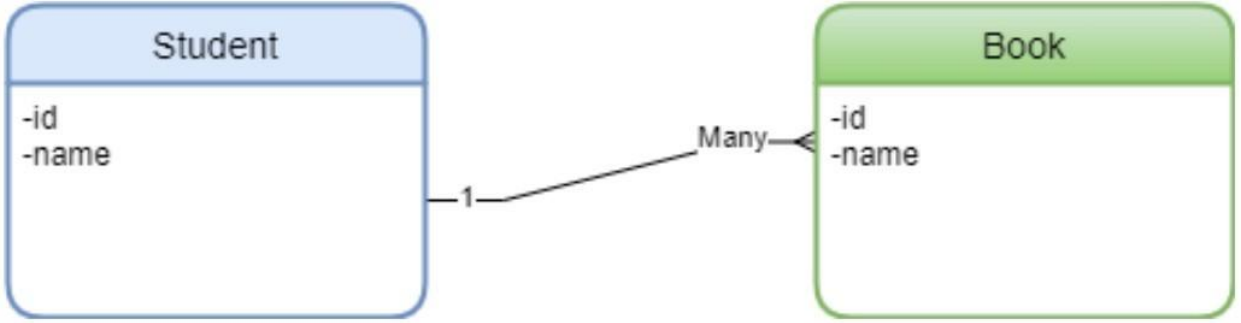


- İlişkinin bir tarafı diğer tarafın referansını tutacak
- Sadece tek tarafdan irtibat kurulabilecek, ulaşılınabilecek.

TECHPROEDUCATION

OneToMany

- Bir Öğrenci sadece 1 veya daha fazla Kitaba sahip olabilir



  - İlişkinin bir tarafı diğer tarafın referansını tutacak
  - Sadece tek tarafdan irtibat kurulabilecek, ulaşılınabilecek.

OneToMany

- Tek Yönlü
  - ✓ @JoinTable
  - ✓ @JoinColumn

- Çift Yönlü
  - ✓ @JoinColumn and mappedBy attribute

Iyi
Calismalar...