

Lab 1: ORM Mapping from Java class to Database table

Objective:

- To store a Java object to Relational Database using Hibernate / **JPA** methods
- Learn object querying for using HQL / **JPQL**

Problem Description:

A Trainer has some attributes which needs to be stored in the System.

Steps to be followed:

1. Create a **Java project** by name – **ORM_Lab_01**.
2. Create a class **Trainer** with a package name - **com.pratian.ormlabs.entities** with fields and operations as stated in the class diagram.

com.pratian.ormlabs.entities.Trainer
-trainerId: long -fullName: String -emailId: String -yearlyTarget: float -age: int
+Trainer() +Trainer(String fullName, String emailId, float yearlyTarget, int age) +long getTrainerId() +void setTrainerId(long trainerId) +String getFullName() +void setFullName(String fullName) +String getEmailId() +void setEmailId(String emailId) +float getYearlyTarget() +void setYearlyTarget(float yearlyTarget) +int getAge() +void setAge(int age) +String toString()

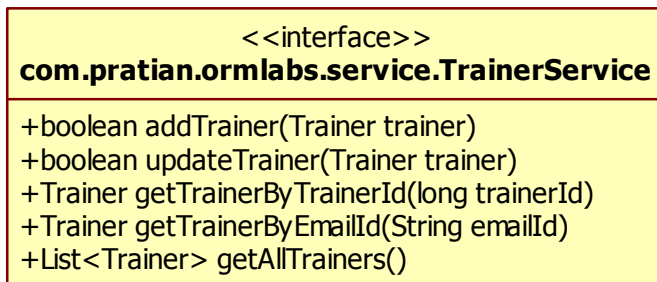
Override “**toString()**” method

Write JPA annotation **@Entity** to persist Trainer class

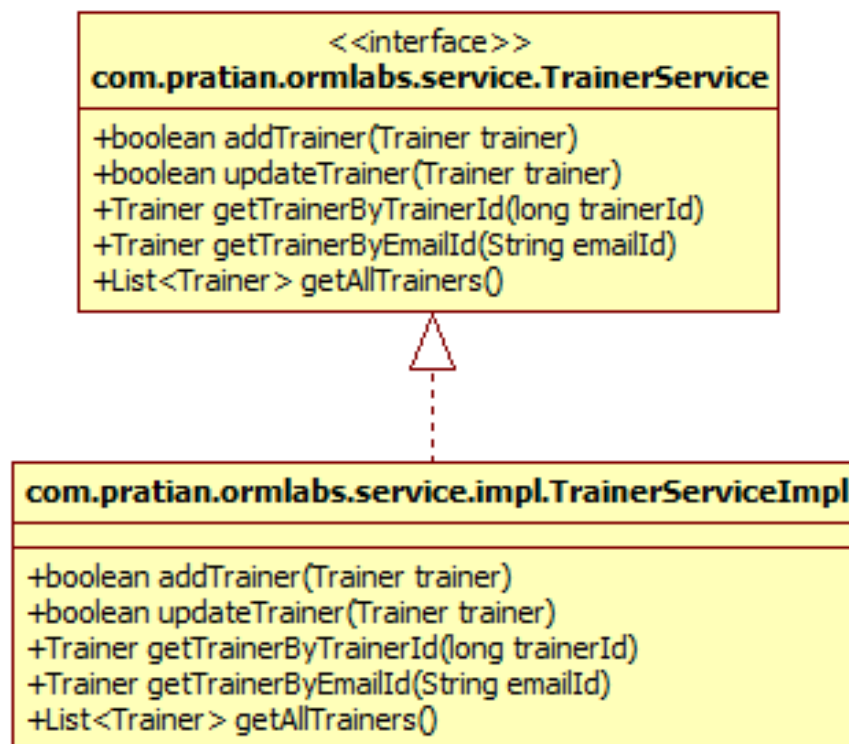
Write JPA annotation **@Id** and **@GeneratedValue(strategy = GenerationType.AUTO)** to identify **primary key**

To perform **business** functionalities, we need to create a new interface that must be realized in the application.

Create an interface **TrainerService** inside a package - **com.pratian.ormlabs.service** with below methods

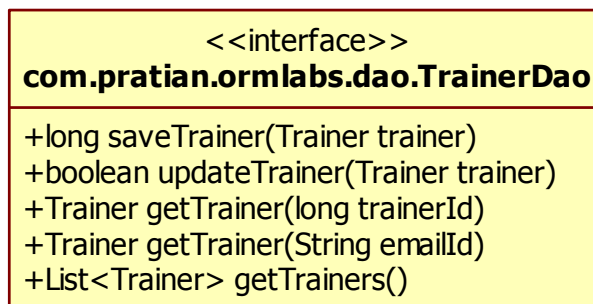


Create a class **TrainerServiceImpl** in a package **com.pratian.ormlabs.service.impl** which implements the above interface



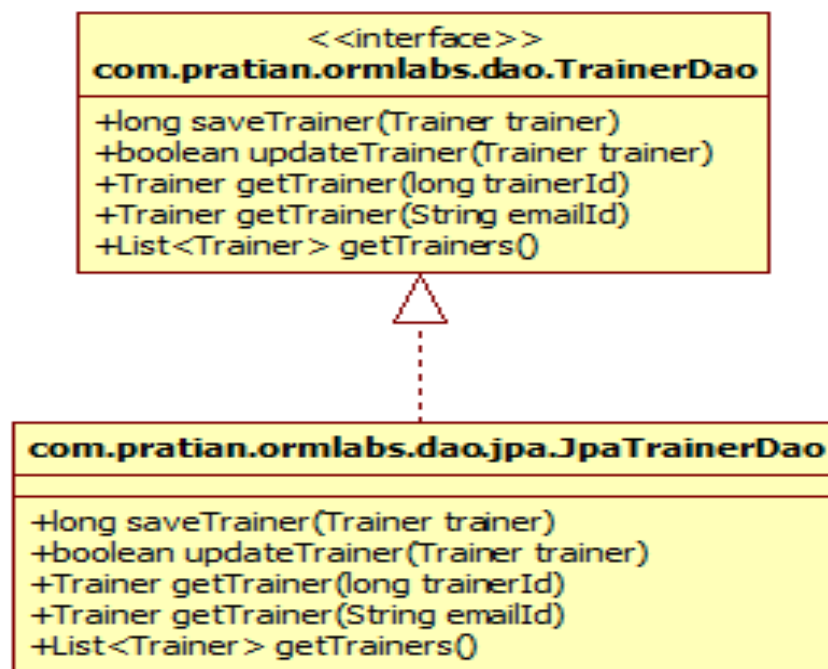
To **store the objects to database**, we will create **Data layer** that manages the domain objects

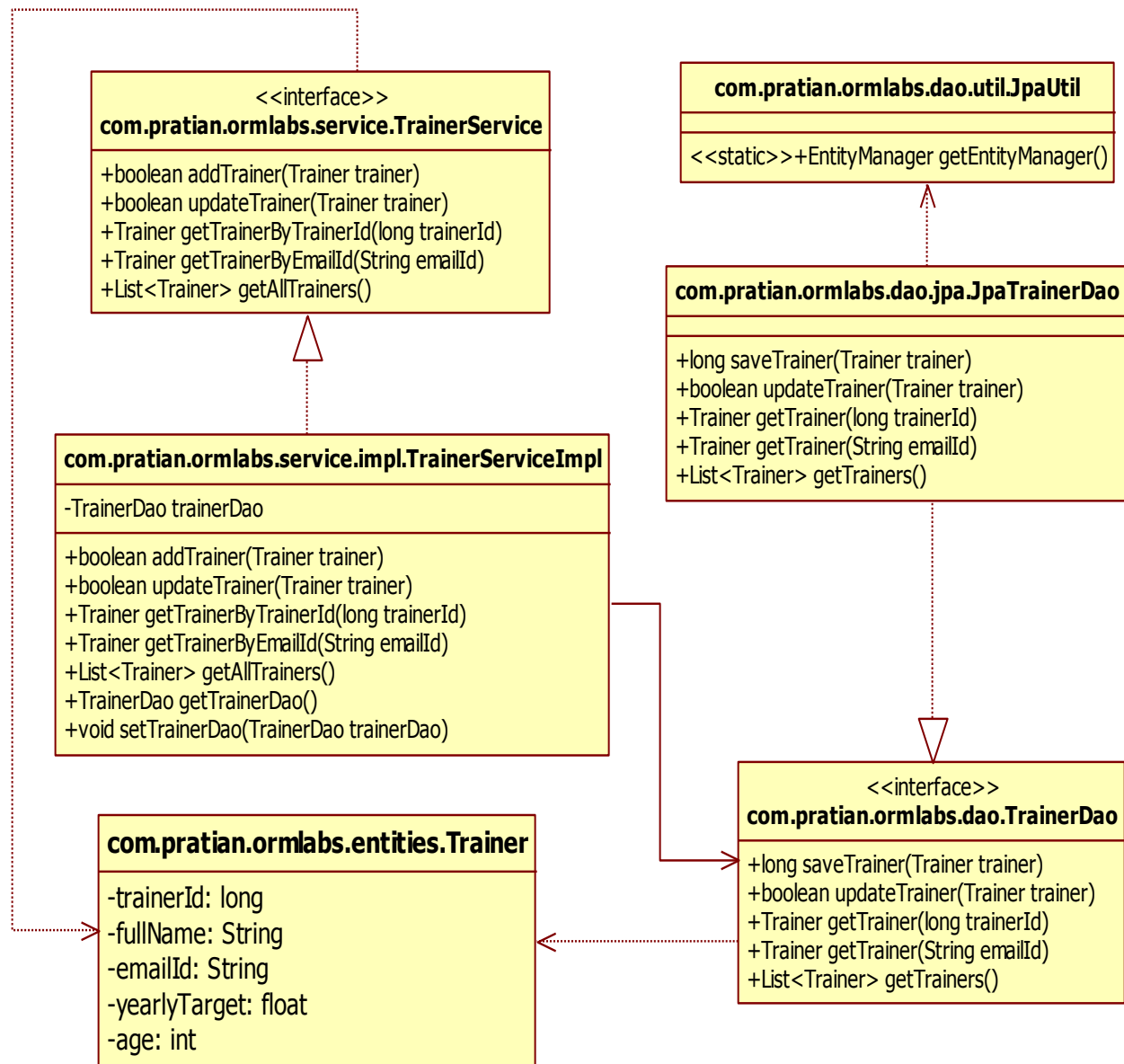
Create an interface **TrainerDao** in a new package **com.pratian.ormlabs.dao** with methods to perform DML / DQL operations based on the service methods identified above in-service package.



Based on the type of DAO implementation, create a class that implements **TrainerDao** interface.

If you are using **Jdbc** / **Hibernate** / **Jpa** than follow naming convention for class as **JdbcTrainerDao** / **HibernateTrainerDao** / **JpaTrainerDao**. The below class is implemented with **JpaTrainerDao** in a package **com.pratian.ormlabs.dao.jpa.JpaTrainerDao**





Write a test class called 'TestMain1 in a package **com.pratian.ormlabs.main** to save a Trainer:

```
public static void main(String[] args) {
    // Identify test data
    String fullName = "Chandra Shekhar";
    String emailId = "chandrashekhar@pratian.com";
    float target = 10_00_000.0f;
    int age = 25;

    // Create a Trainer
    Trainer trainer = new Trainer(fullName, emailId, target, age);

    // Create Trainer service
    TrainerServiceImpl service = new TrainerServiceImpl();

    // Set dao implementation to be used by service
    service.setTrainerDao(new JpaTrainerDao());

    // Save Trainer
    boolean status = service.addTrainer(trainer);

    // Display storing trainer status
    String message1 = "Saved successfully ";
    String message2 = "Something went wrong!!! \n Try Again";
    System.out.println(status ? message1 : message2);
}
```

Expected output of TestMain1

A new table Trainer

```
mysql> desc trainer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| trainerId | bigint(20) | NO | PRI | NULL |  |
| age | int(11) | NO |  | NULL |  |
| emailId | varchar(255) | YES |  | NULL |  |
| fullName | varchar(255) | YES |  | NULL |  |
| yearlyTarget | float | NO |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.32 sec)
```

A record to Trainer table:

```
mysql> select * from trainer;
+-----+-----+-----+-----+-----+
| trainerId | age | emailId | fullName | yearlyTarget |
+-----+-----+-----+-----+-----+
| 1 | 25 | chandrashekhar@pratian.com | Chandra Shekhar | 1000000 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Write a test class called 'TestMain2' in a package **com.pratian.ormlabs.main** to update a Trainer

```
public static void main(String[] args) {
    // Identify test data to update the trainer
    long trainerId = 1;
    String fullName = "Chandra Shekhar H M";
    String emailId = "chandrashekhar@pratian.com";
    float target = 15_00_000.0f;
    int age = 26;

    // Create a Trainer to update the details
    Trainer trainer = new Trainer(fullName, emailId, target, age);
    trainer.setTrainerId(trainerId);

    // Create Trainer service
    TrainerServiceImpl service = new TrainerServiceImpl();

    // Set dao implementation to be used by service
    service.setTrainerDao(new JpaTrainerDao());

    // Update Trainer
    boolean status = service.editTrainer(trainer);

    // Display update trainer status
    String message1 = "Updated successfully ";
    String message2 = "Something went wrong!!! \n Not updated";
    System.out.println(status ? message1 : message2);
}
```

Expected Output is updated trainer details

Updated Age

Updated Name

Updated Target

```
mysql> select * from trainer;
+-----+-----+-----+-----+-----+
| trainerId | age | emailId | fullName | yearlyTarget |
+-----+-----+-----+-----+-----+
| 1 | 26 | chandrashekhar@pratian.com | Chandra Shekhar H M | 1500000 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Write a test class called 'TestMain3' in a package **com.pratian.ormlabs.main** to search a Trainer by trainerId

```
public static void main(String[] args) {
    // Identify test data
    long trainerId = 1;

    // Create a Trainer to be searched
    Trainer trainer = new Trainer();
    trainer.setTrainerId(trainerId);

    // Create Trainer service
    TrainerServiceImpl service = new TrainerServiceImpl();

    // Set dao implementation to be used by service
    service.setTrainerDao(new JpaTrainerDao());

    // Search a Trainer by trainer id
    Trainer resultObj = service.getTrainerByTrainerId(trainerId);

    // Display trainer details
    System.out.println("-----");
    System.out.println("Trainer Id : " + resultObj.getTrainerId());
    System.out.println("Full Name : " + resultObj.getFullName());
    System.out.println("Email Id : " + resultObj.getEmailId());
    System.out.println("Age : " + resultObj.getAge());
    System.out.println("Target : " + resultObj.getYearlyTarget());
    System.out.println("-----");
}
```

Expected Output

```
-----
Trainer Id : 1
Full Name : Chandra Shekhar H M
Email Id : chandrashekhar@pratian.com
Age : 26
Target :1500000.0
-----
```

Write a test class called 'TestMain4' in a package **com.pratian.ormlabs.main** to search a Trainer by email_Id

```
public static void main(String[] args) {
    // Identify test data
    String emailId = "chandrashekhar@pratian.com";

    // Create a Trainer to be searched
    Trainer trainer = new Trainer();
    trainer.setEmailId(emailId);

    // Create Trainer service
    TrainerServiceImpl service = new TrainerServiceImpl();

    // Set dao implementation to be used by service
    service.setTrainerDao(new JpaTrainerDao());

    // Search a Trainer by email id
    Trainer resultObj = service.getTrainerByEmailId(emailId);

    // Display trainer details
    System.out.println("-----");
    System.out.println("Trainer Id : " + resultObj.getTrainerId());
    System.out.println("Full Name : " + resultObj.getFullName());
    System.out.println("Email Id : " + resultObj.getEmailId());
    System.out.println("Age : " + resultObj.getAge());
    System.out.println("Target : " + resultObj.getYearlyTarget());
    System.out.println("-----");
}
```

Expected Output

```
-----
Trainer Id : 1
Full Name : Chandra Shekhar H M
Email Id : chandrashekhar@pratian.com
Age : 26
Target :1500000.0
-----
```


Write a test class called 'TestMain5' in a package **com.pratian.ormlabs.main** to get all Trainers

```
public static void main(String[] args) {  
  
    // Create Trainer service  
    TrainerServiceImpl service = new TrainerServiceImpl();  
  
    // Set dao implementation to be used by service  
    service.setTrainerDao(new JpaTrainerDao());  
  
    // Get all Trainers  
    List<Trainer> trainers = service.getAllTrainers();  
  
    // Display all trainer details  
    System.out.println("-----");  
    System.out.println("TRAINER DETAILS ARE :");  
    for (Trainer trainer : trainers) {  
        System.out.println("-----");  
        System.out.println("Trainer Id : " + trainer.getTrainerId());  
        System.out.println("Full Name : " + trainer.getFullName());  
        System.out.println("Email Id : " + trainer.getEmailId());  
        System.out.println("Age : " + trainer.getAge());  
        System.out.println("Target : " + trainer.getYearlyTarget());  
    }  
}
```

Expected Output

```
-----  
TRAINER DETAILS ARE :  
-----
```

```
Trainer Id : 1  
Full Name : Chandra Shekhar H M  
Email Id : chandrashekhar@pratian.com  
Age : 26  
Target :1500000.0  
-----
```