

JPA - INTRODUCTION

Any enterprise application performs database operations by storing and retrieving vast amounts of data. Despite all the available technologies for storage management, application developers normally struggle to perform database operations efficiently.

Generally, Java developers use lots of code, or use the proprietary framework to interact with the database, whereas using JPA, the burden of interacting with the database reduces significantly. It forms a bridge between object models *Javaprogram* and relational models *databaseprogram*.

Mismatches between relational and object models

Relational objects are represented in a tabular format, while object models are represented in an interconnected graph of object format. While storing and retrieving an object model from a relational database, some mismatch occurs due to the following reasons:

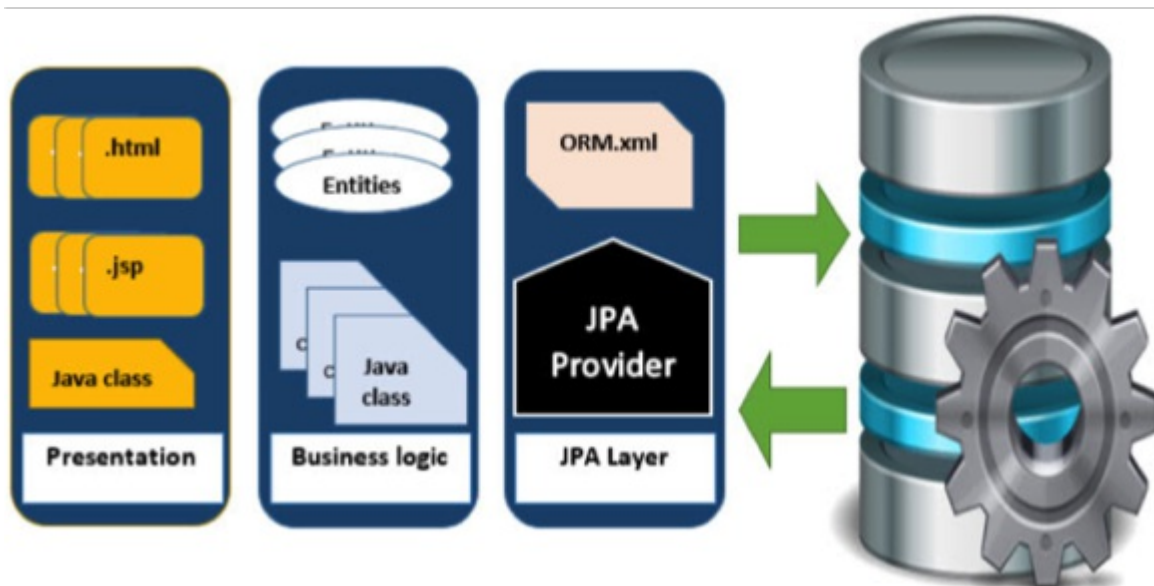
- **Granularity** : Object model has more granularity than relational model.
- **Subtypes** : Subtypes *meansinheritance* are not supported by all types of relational databases.
- **Identity** : Like object model, relational model does not expose identity while writing equality.
- **Associations** : Relational models cannot determine multiple relationships while looking into an object domain model.
- **Data navigation** : Data navigation between objects in an object network is different in both models.

What is JPA?

Java Persistence API is a collection of classes and methods to persistently store the vast amounts of data into a database which is provided by the Oracle Corporation.

Where to use JPA?

To reduce the burden of writing codes for relational object management, a programmer follows the 'JPA Provider' framework, which allows easy interaction with database instance. Here the required framework is taken over by JPA.



JPA History

Earlier versions of EJB, defined persistence layer combined with business logic layer using `javax.ejb.EntityBean` Interface.

- While introducing EJB 3.0, the persistence layer was separated and specified as JPA 1.0 *JavaPersistenceAPI*. The specifications of this API were released along with the specifications of JAVA EE5 on May 11, 2006 using JSR 220.
- JPA 2.0 was released with the specifications of JAVA EE6 on December 10, 2009 as a part of Java Community Process JSR 317.
- JPA 2.1 was released with the specification of JAVA EE7 on April 22, 2013 using JSR 338.

JPA Providers

JPA is an open source API, therefore various enterprise vendors such as Oracle, Redhat, Eclipse, etc. provide new products by adding the JPA persistence flavor in them. Some of these products include:

Hibernate, Eclinselink, Toplink, Spring Data JPA, etc.

Loading [MathJax]/jax/output/HTML-CSS/jax.js