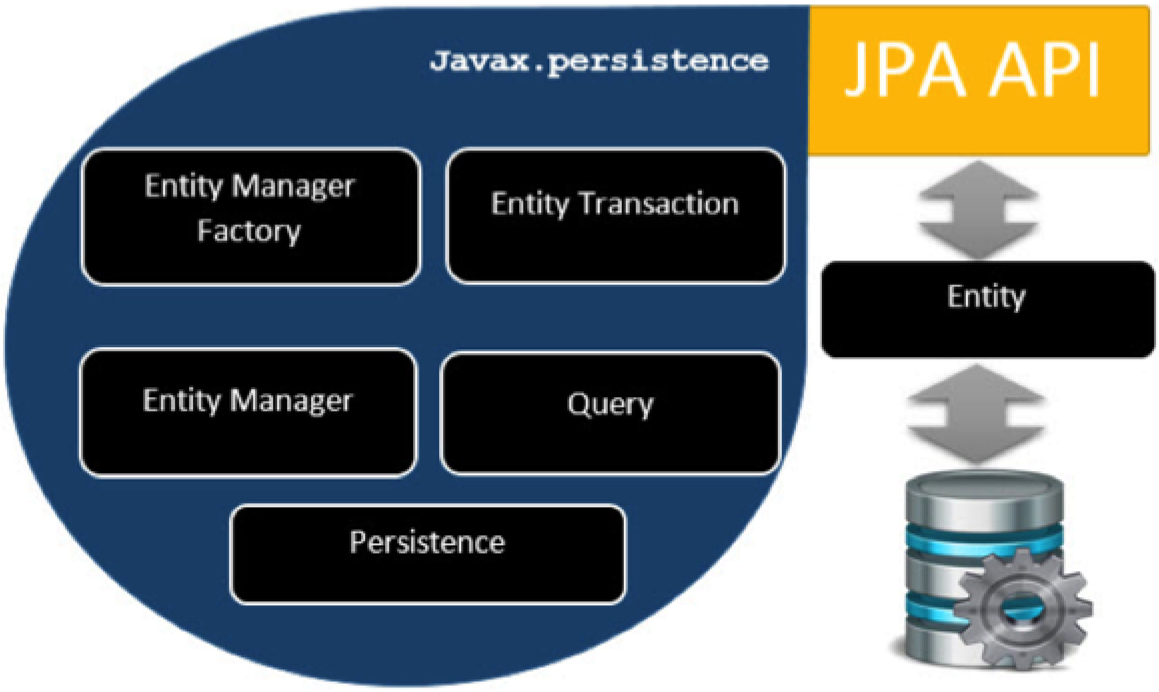
Java Persistence API is a source to store business entities as relational entities. It shows how to define a PLAIN OLD JAVA OBJECT (POJO) as an entity and how to manage entities with relations.

Class Level Architecture

The following image shows the class level architecture of JPA. It shows the core classes and interfaces of JPA.



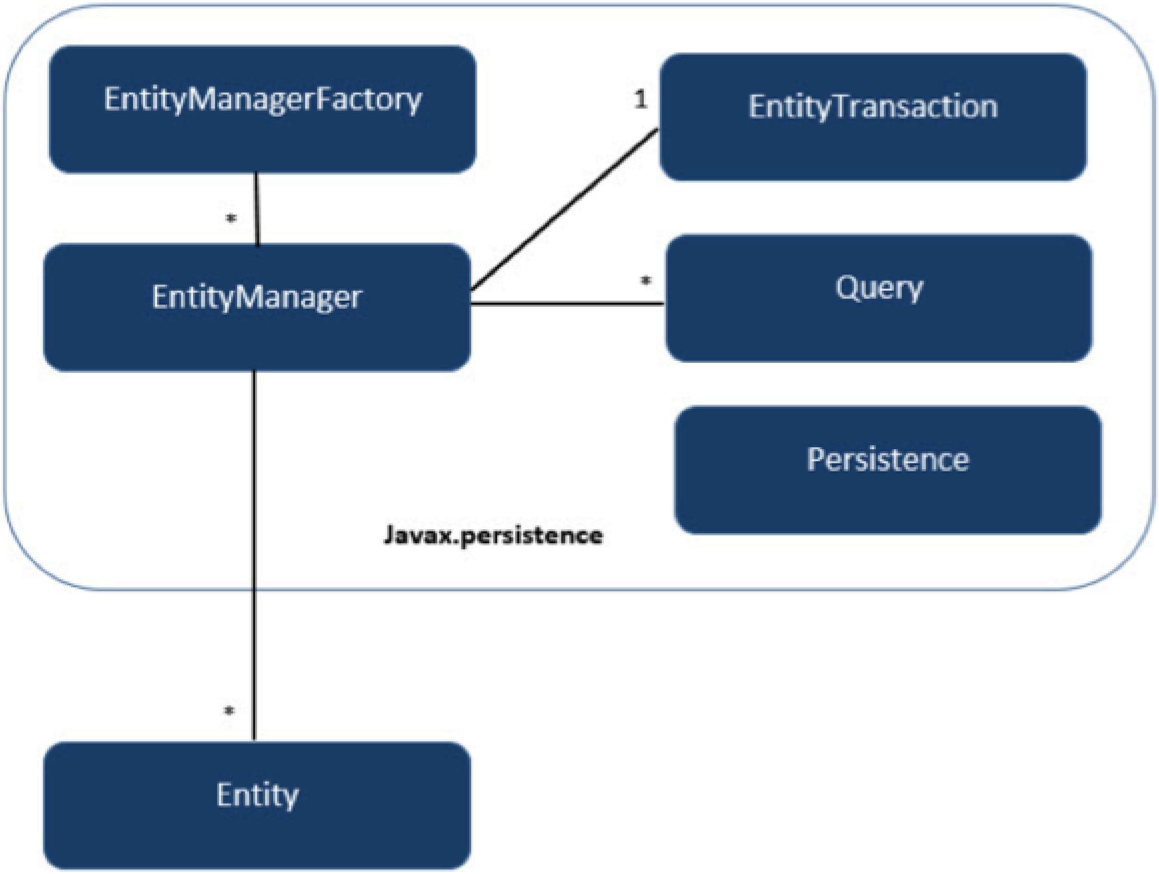
The following table describes each of the units shown in the above architecture.

Units	Description
EntityManagerFactory	This is a factory class of EntityManager. It creates and manages multiple EntityManager instances.
EntityManager	It is an Interface, it manages the persistence operations on objects. It works like factory for Query instance.
Entity	Entities are the persistence objects, stores as records in the database.
EntityTransaction	It has one-to-one relationship with EntityManager. For each EntityManager, operations are maintained by EntityTransaction class.
Persistence	This class contain static methods to obtain EntityManagerFactory instance.
Query	This interface is implemented by each JPA vendor to obtain relational objects that meet the criteria.

The above classes and interfaces are used for storing entities into a database as a record. They help programmers by reducing their efforts to write codes for storing data into a database so that they can concentrate on more important activities such as writing codes for mapping the classes with database tables.

JPA Class Relationships

In the above architecture, the relations between the classes and interfaces belong to the javax.persistence package. The following diagram shows the relationship between them.



- The relationship between EntityManagerFactory and EntityManager is **one-to-many**. It is a factory class to EntityManager instances.
- The relationship between EntityManager and EntityTransaction is **one-to-one**. For each EntityManager operation, there is an EntityTransaction instance.
- The relationship between EntityManager and Query is **one-to-many**. Many number of queries can execute using one EntityManager instance.
- The relationship between EntityManager and Entity is **one-to-many**. One EntityManager instance can manage multiple Entities.