Hibernate is an open-source Object-relational mapper solution in Java. It is lightweight and overcomes all the shortcomings that we face while working with JDBC.
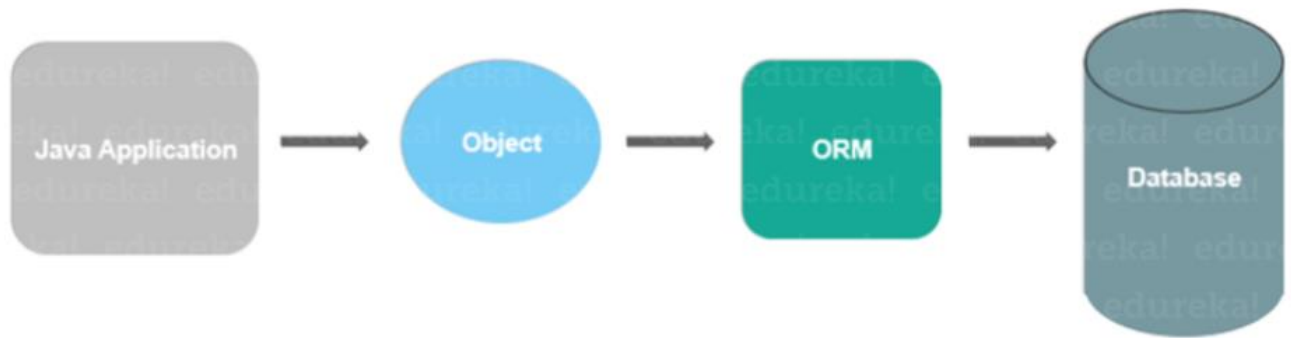
What Is Hibernate In Java?

Hibernate is a framework in Java which comes with an abstraction layer and handles the implementations internally. The implementations include tasks like writing a query for CRUD operations or establishing a connection with the databases, etc.

A framework is basically software that provides abstraction on multiple technologies like JDBC, servlet, etc.

Hibernate develops persistence logic, which stores and processes the data for longer use. It is lightweight and an ORM tool, and most importantly open-source which give it an edge over other frameworks.

What Is An ORM Tool?

It is a technique that maps the object stored in the database. An ORM tool simplifies data creation, manipulation, and access. It internally uses the Java API to interact with the databases.

Let's take a look at the need for using hibernate in Java.

Need For Hibernate Framework

Hibernate eliminates the shortcomings of other technologies like JDBC. Let's take a look at how it optimizes the tasks better than JDBC.

- Hibernate overcomes the database dependency faced in the JDBC.
- Changing of the databases cost a lot working on JDBC, hibernate overcomes this problem with flying colors.
- Code portability is not an option while working on JDBC, which is handled by hibernate easily.
- Hibernate strengthens the object level relationship.
- It overcomes the exception-handling part which is mandatory while working on JDBC.
- Hibernate overcomes the object level relationship.
- It reduces the length of code with increased readability by overcoming the boiler plate problem.

Hibernate provides optimal and efficient solutions for any task by overcoming all the shortcomings of JDBC. Let us take a look at various

operations along with technologies and databases we can work on while using the hibernate framework in Java.

Introduction To Hibernate In Java

Being an open-source framework, it is available for everyone without any cost. The source code can be found on the internet for hibernate which also allows modifications as well.

The advantage of being a lightweight framework can be seen considerably smaller package for installation. The efficiency increases with not using any container for execution.

Even though hibernate can work with multiple technologies at once, but it does not mean that hibernate cannot work alone. We can work on hibernate alone as well i.e. without any technologies.

Hibernate has a peculiar nature, where it does not have to implement hibernate API interfaces or extend from hibernate API classes since classes of hibernate application development are loosely coupled.

Functionalities Supported By Hibernate

- Hibernate uses Hibernate Query Language which makes it database independent.
- It supports auto DDL operations.
- Hibernate has Auto Primary Key Generation support.
- It supports Cache memory.
- Exception handling is not mandatory for hibernate.

- The most important is hibernate is an ORM tool.

Supported Databases In Hibernate

Following are the databases supported by hibernate in Java.

- HSQL Database Engine
- MYSQL
- ORACLE
- FrontBase
- PostgreSQL
- DB2/NT
- Sybase SQL Server
- Informix Dynamic Server
- Microsoft SQL Server Database

Hibernate almost supports all the major RDBMS which makes it efficient and easy to work with.

Let's take a look at a few advantages of hibernate in java.

Advantages Of Hibernate In Java

- Lightweight and open-source – Being lightweight and open-source makes it accessible and efficient.
- Increased performance – Using cache memory helps in fast performance.

- Database Independence – Being database-independent gives it the ability to work with different databases.
- Auto DDL Operations – automatic table creation saves us from manually creating tables.
- It takes care of mapping Java classes databases using XML files without writing any code.
- We can directly store and retrieve data directly from the database using simple APIs.
- It does not require any application server to operate.
- Minimizes database access with smart fetching strategies.
- It provides simple querying of data.

Hibernate is an object-relational mapper that overcomes the shortcomings of JDBC in Java. With optimal solutions and efficiency, it becomes fairly easy to work with databases without any dependencies.

Java programming language is filled with such technologies, with the increased efficiency the demand for java developers has increased significantly during the last decade. With the increasing demand, it is extremely important to be on par with all the technological advancements with the programming language.
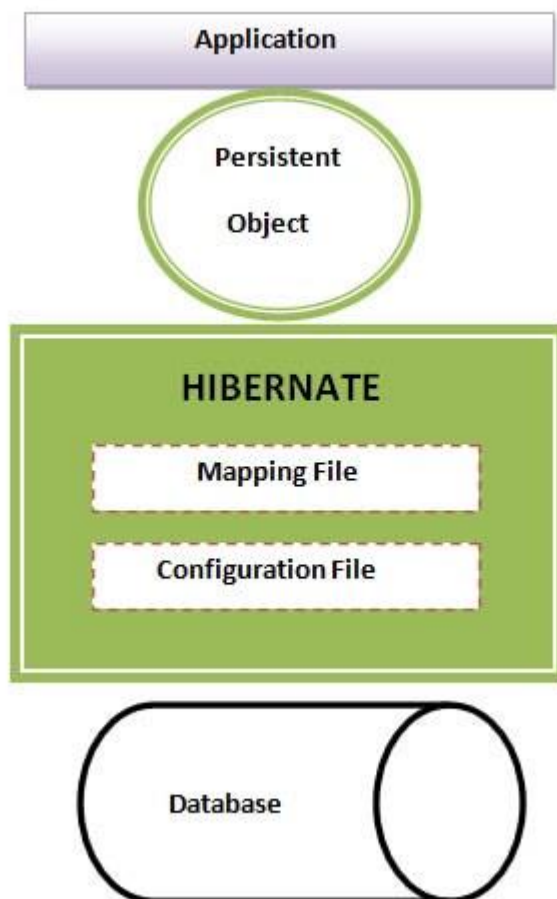
Hibernate Architecture

The Hibernate architecture includes many objects such as persistent object, session factory, transaction factory, connection factory, session, transaction etc.
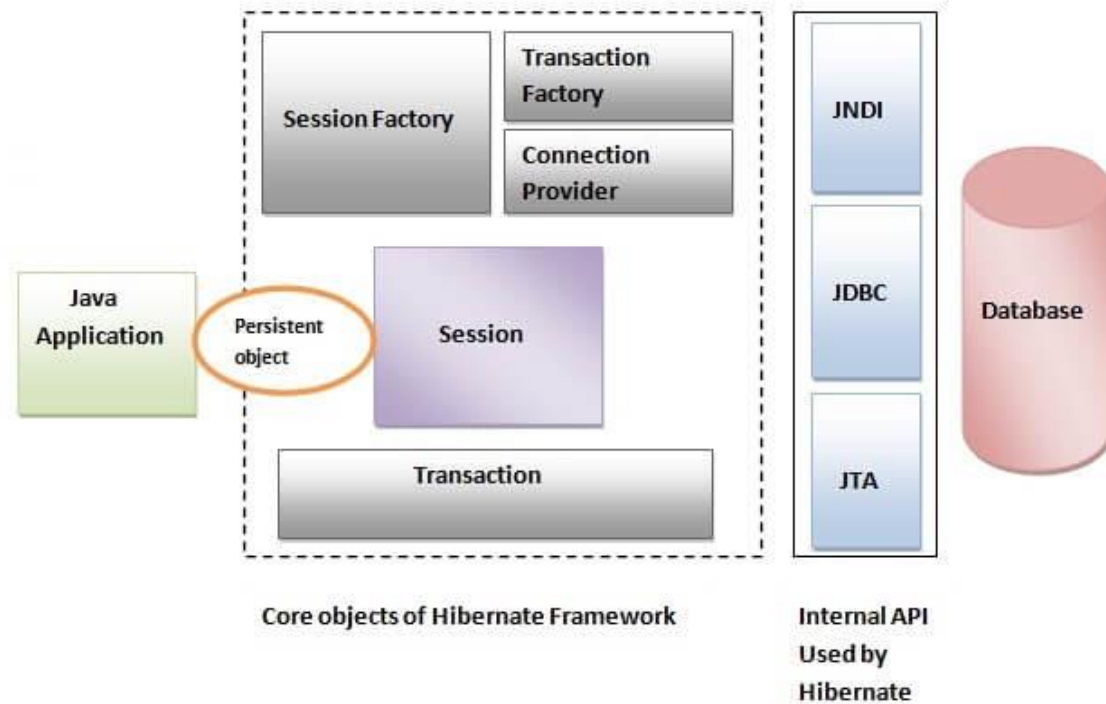
The Hibernate architecture is categorized in four layers.

- o Java application layer
- o Hibernate framework layer
- o Backhand api layer
- o Database layer

Let's see the diagram of hibernate architecture:

This is the high level architecture of Hibernate with mapping file and configuration file.

Core objects of Hibernate Framework   Internal API
Used by
Hibernate

Elements of Hibernate Architecture

For creating the first hibernate application, we must know the elements of Hibernate architecture. They are as follows:

SessionFactory

The SessionFactory is a factory of session and client of ConnectionProvider. It holds second level cache (optional) of data. The org.hibernate.SessionFactory interface provides factory method to get the object of Session.

SessionFactory is an Interface which is present in org. hibernate package and it is used to create Session Object

SessionFactory is a factory class for Session objects. It is available for the whole application

We can create one SessionFactory implementation per database in any application. If your application is referring to multiple databases, then you need to create one SessionFactory per database.

The SessionFactory is a heavyweight object; it is usually created during application start up and kept for later use. The SessionFactory is a thread safe object and used by all the threads of an application.

Session

The session object provides an interface between the application and data stored in the database. It is a short-lived object and wraps the JDBC connection.

It is factory of Transaction, Query and Criteria. It holds a first-level cache (mandatory) of data. The org.hibernate.Session interface provides methods to insert, update and delete the object. It also provides factory methods for Transaction, Query and Criteria.

Session is only available for particular transaction. It is not thread-safe.

sessions will be opened using sessionfactory.openSession() and some database operations will be done finally session will be closed using session.close().

Transaction

The transaction object specifies the atomic unit of work. It is optional. The org.hibernate.Transaction interface provides methods for transaction management.

ConnectionProvider

It is a factory of JDBC connections. It abstracts the application from DriverManager or DataSource. It is optional.

TransactionFactory

It is a factory of Transaction. It is optional.