HIBERNATE 5

Unit IV

Prepared By: Bhavana Hotchandani, DCS, INDUS University

Introduction

- Hibernate is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.
- ORM Tool
  - An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.
  - The ORM tool internally uses the JDBC API to interact with the database.

Advantages of Hibernate Framework

- Open Source and Lightweight
- Fast Performance
  - Two levels of caching
- Database Independent Query
  - HQL
- Automatic Table Creation
- Simplifies Complex Join
- Provides Query Statistics and Database Status

PROS AND CONS OF JDBC

Pros of JDBC	Cons of JDBC
Clean and simple SQL processing	Complex if it is used in large projects
Good performance with large data	Large programming overhead
Very good for small applications	No encapsulation
Simple syntax so easy to learn	Hard to implement MVC concept
	Query is DBMS specific

Why Object Relational Mapping (ORM)?

- When we work with an object-oriented system, there is a mismatch between the object model and the relational database. RDBMSs represent data in a tabular format whereas object-oriented languages, such as Java or C# represent it as an interconnected graph of objects.
- Example
- Problems
  - what if we need to modify the design of our database after having developed a few pages or our application?
  - loading and storing objects in a relational database exposes us to the following five mismatch problems

- **RDBMS** impedance mismatches
  - **Granularity**
  - **Inheritance**
  - **Identity**
  - **Associations**
  - **Navigation**
- The **O**bject-**R**elational **M**apping (ORM) is the solution to handle all the above impedance mismatches.

What is ORM?

- ORM stands for **O**bject-**R**elational **M**apping (ORM) is a programming technique for converting data between relational databases and object oriented programming languages such as Java, C#, etc.
- An ORM system has the following advantages over plain JDBC –

1	Let’s business code access objects rather than DB tables.
2	Hides details of SQL queries from OO logic.
3	Based on JDBC 'under the hood.'
4	No need to deal with the database implementation.
5	Entities based on business concepts rather than database structure.
6	Transaction management and automatic key generation.
7	Fast development of application.

Hibernate Architecture

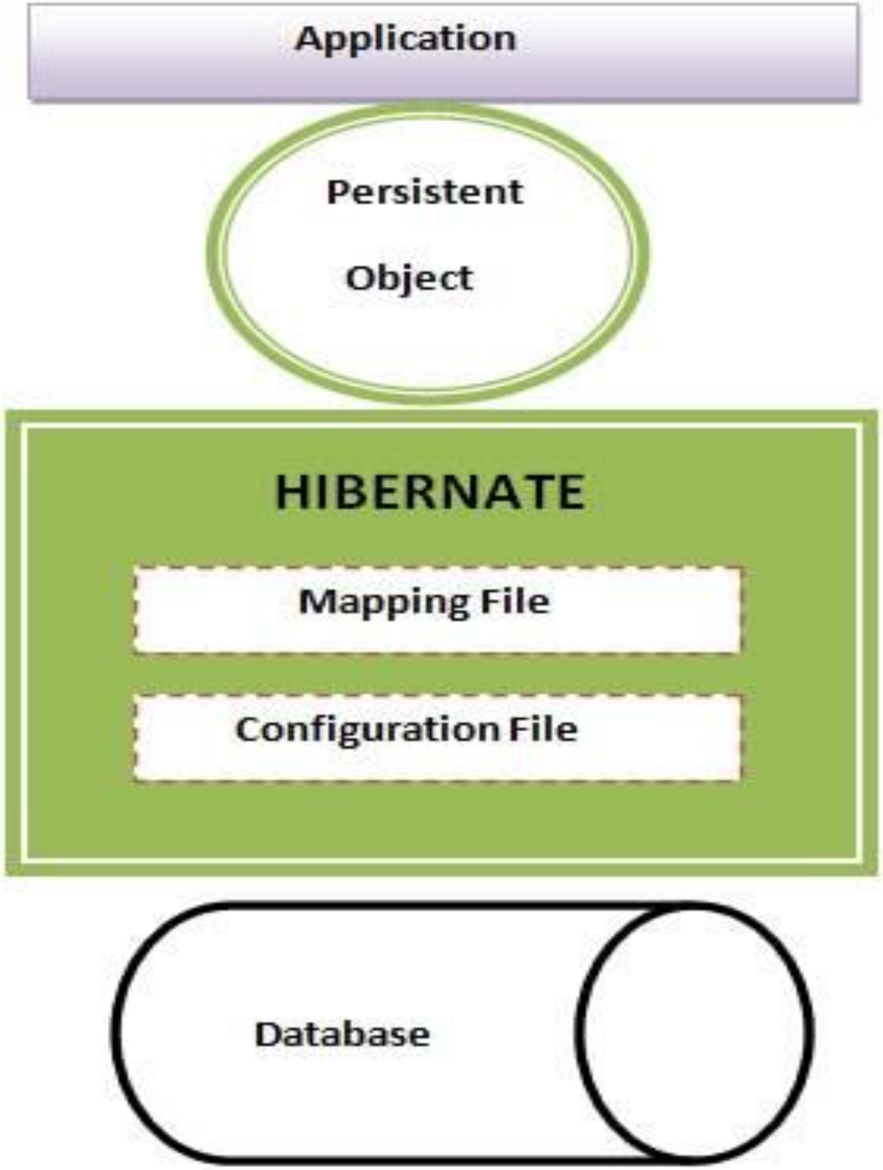
The Hibernate architecture includes many objects such as

- persistent object,
- session factory,
- transaction factory,
- connection factory,
- session,
- transaction etc.

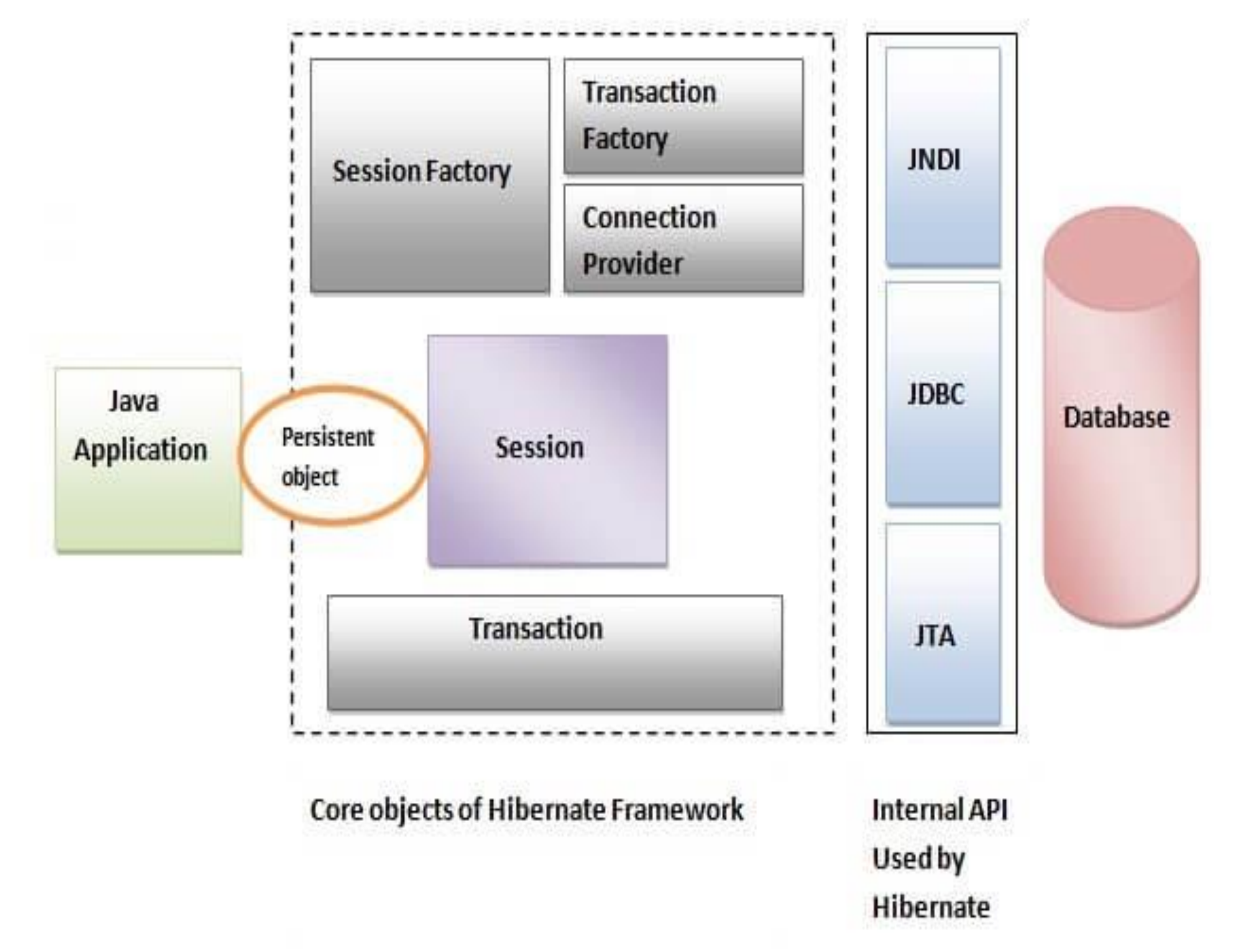
The Hibernate architecture is categorized in four layers.

- Java application layer
- Hibernate framework layer
- Backend api layer
- Database layer

HIGH LEVEL ARCHITECTURE



Prepared By: Bhavana Hotchandani, DCS, INDUS University



Detailed Architecture

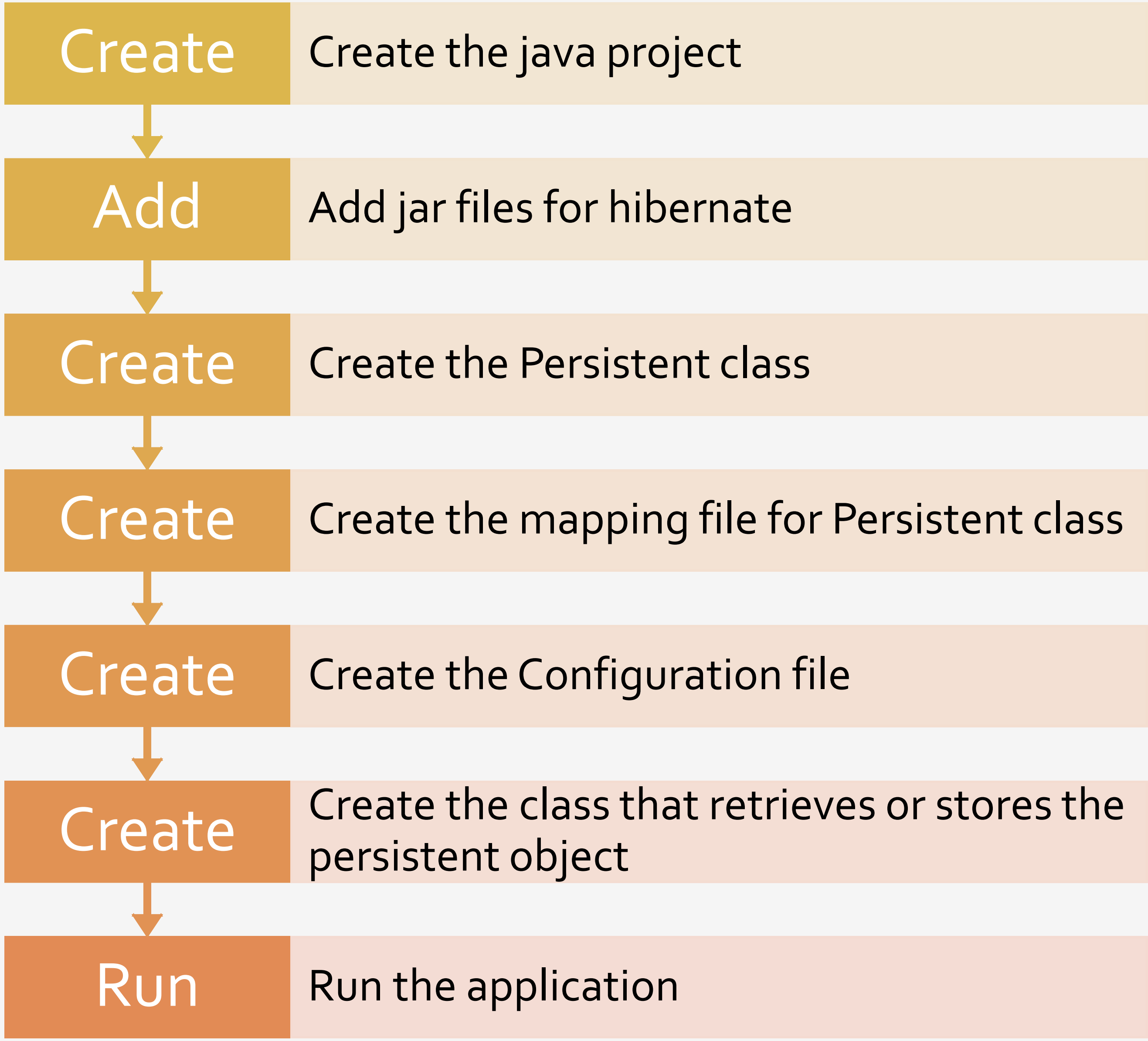
- Hibernate framework uses many objects such as
  - session factory
  - session
  - transaction etc.
- alongwith existing Java API such as
  - JDBC
  - JTA and
  - JNDI

Elements of Hibernate Architecture

- 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.
- 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.
- 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.

Hibernate Example using XML in Eclipse

Prepared By: Bhavana Hotchandani, DCS, INDUS University



Create — Create the java project

Add — Add jar files for hibernate

Create — Create the Persistent class

Create — Create the mapping file for Persistent class

Create — Create the Configuration file

Create — Create the class that retrieves or stores the persistent object

Run — Run the application