

OBJECTTM
TECHNOLOGIES

Introduction to Hibernate

What Will Be Covered

OBJECT

- Drawbacks of JDBC
- What is ORM
- Java ORM Frameworks
- What is Hibernate?
- Advantages of Hibernate Framework
- Features of Hibernate
- Hibernate versions
- Hibernate architecture - high level view
- Hibernate architecture - low level view



Drawbacks of JDBC

OBJECT

- Tied to specific database
- Error-prone try/catch/finally
- Relationships are really hard
 - N+1 selects problem
 - parent/child updates
- Screen-specific DAOs for optimization
- Programmer must hardcode the Transactions and concurrency code in the application.
- Handling the JDBC connections and properly closing the connection is also a big issue. Properly closing the connection is must.
- JDBC is not good for big applications

What is ORM

OBJECT

- ORM stands for **Object-Relational Mapping (ORM)**.
- It is a programming technique for converting data between relational databases and object oriented programming languages such as Java, C# etc.
- ORM solves following problems :
 - if we need to modify the design of our database after having developed few pages or our application.
 - Loading and storing objects in a relational database exposes us to the following mismatched problems like problem of granularity, inheritance and associations etc

Java ORM Frameworks

OBJECT

- An ORM solution consists of the following four entities:

1. An API to perform basic CRUD operations on objects of persistent classes.
 2. A language or API to specify queries that refer to classes and properties of classes.
 3. A configurable facility for specifying mapping metadata.
 4. A technique to interact with transactional objects to perform dirty checking, lazy association fetching, and other optimization functions.
- There are several persistent frameworks-and ORM options in Java.

- ◆ Enterprise JavaBeans Entity Beans
- ◆ Castor
- ◆ TopLink
- ◆ Spring DAO
- ◆ Hibernate

What is Hibernate?

OBJECT

- Java Object/Relational Mapping
 - ◆ Open source – LGPL
 - ◆ <http://www.hibernate.org/> (JBoss Group)
- Lets you avoid SQL
- Works with (almost) all relational DBs o DB2, FrontBase, HSQLDB, Informix, Ingres, Interbase, Mckoi, MySQL, Oracle, Pointbase, PostgreSQL, Progress, SAP DB, SQL Server, Sybase, etc...
- Has a huge feature list
- Maps JavaBeans (POJOs) to tables
 - ◆ XML defines the mappings
 - ◆ Very few bean requirements
- SQL is generated at app startup time
 - ◆ As opposed to bytecode manipulation
 - ◆ New database? No problem! Change a few props

Advantages of Hibernate Framework

OBJECT

- **Open source and Lightweight:** Hibernate framework is open source under the LGPL license and lightweight.
- **Hibernate is a non-invasive framework,** means it won't force the programmers to extend/implement any class/interface, and in hibernate we have all POJO classes so its light weight.
- Hibernate can run **with in or with out server**, i mean it will suitable for all types of java applications (stand alone or desktop or any servlets bla bla.)
- Hibernate is purely for persistence
- **Fast performance:** The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.
- **Database Independent query:** HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries

Advantages of Hibernate Framework

OBJECT

- **Automatic table creation:** Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.
- **Simplifies complex join:** To fetch data from multiple tables is easy in hibernate framework.
- **Provides query statistics and database status:** Hibernate supports Query cache and provide statistics about query and database status.

Features of Hibernate

OBJECT

- **HibernateDaoSupport** – superclass, easy **HibernateTemplate** access
- **Database independence** - sits between the database and your java code, easy database switch without changing any code
- **Object / Relational Mapping (ORM)** - Allows a developer to treat a database like a collection of Java objects
- **Object oriented query language (HQL)** - *Portable* query language, supports polymorphic queries etc.
- **Support for native SQL**, and also queries by “Criteria” (specified using “parse tree” of Java objects)
- **Hibernate Mapping** - Uses HBM XML files to map value objects (POJOs) to database tables
- **Transparent persistence** - Allows easy saves/delete/retrieve for simple value objects
- **Very high performance** “in general” due to intelligent (2-level) caching, although in a few cases hand-written SQL might beat it

Hibernate versions

OBJECT

Hibernate Versions



Hibernate Versions

OBJECT

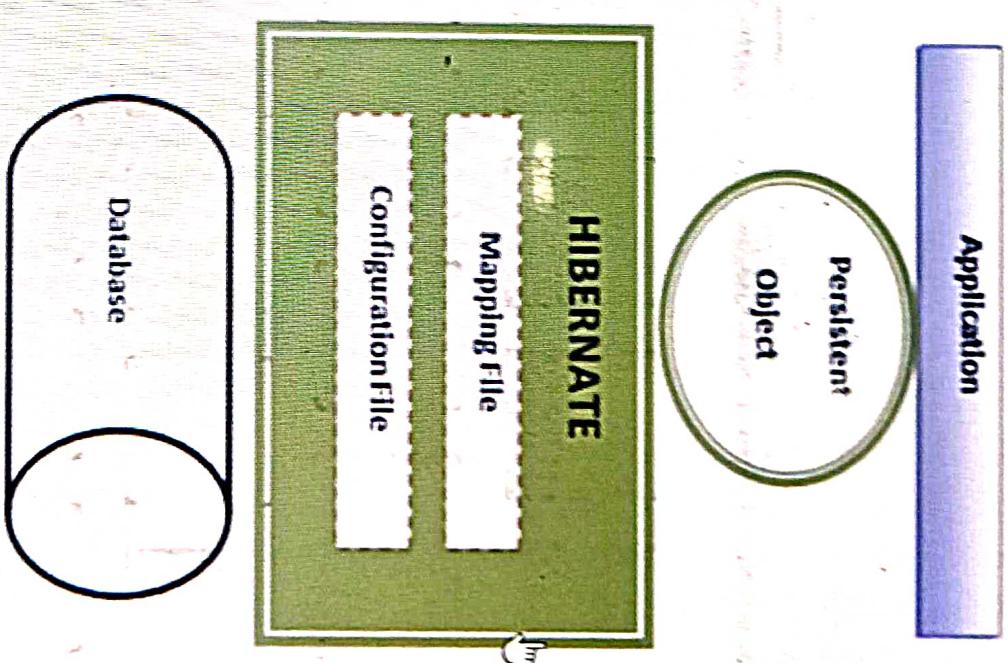
- Current stable version of Hibernate is 6.1.0 with some additional improved features.
- Previous stable version was 5.6

Hibernate ORM	Java	JPA	Jakarta Persistence
6.1	11.17 or 18	N/A	3.1 and 3.0
5.6	8.11, 17 or 18	2.2	3.0

High level view

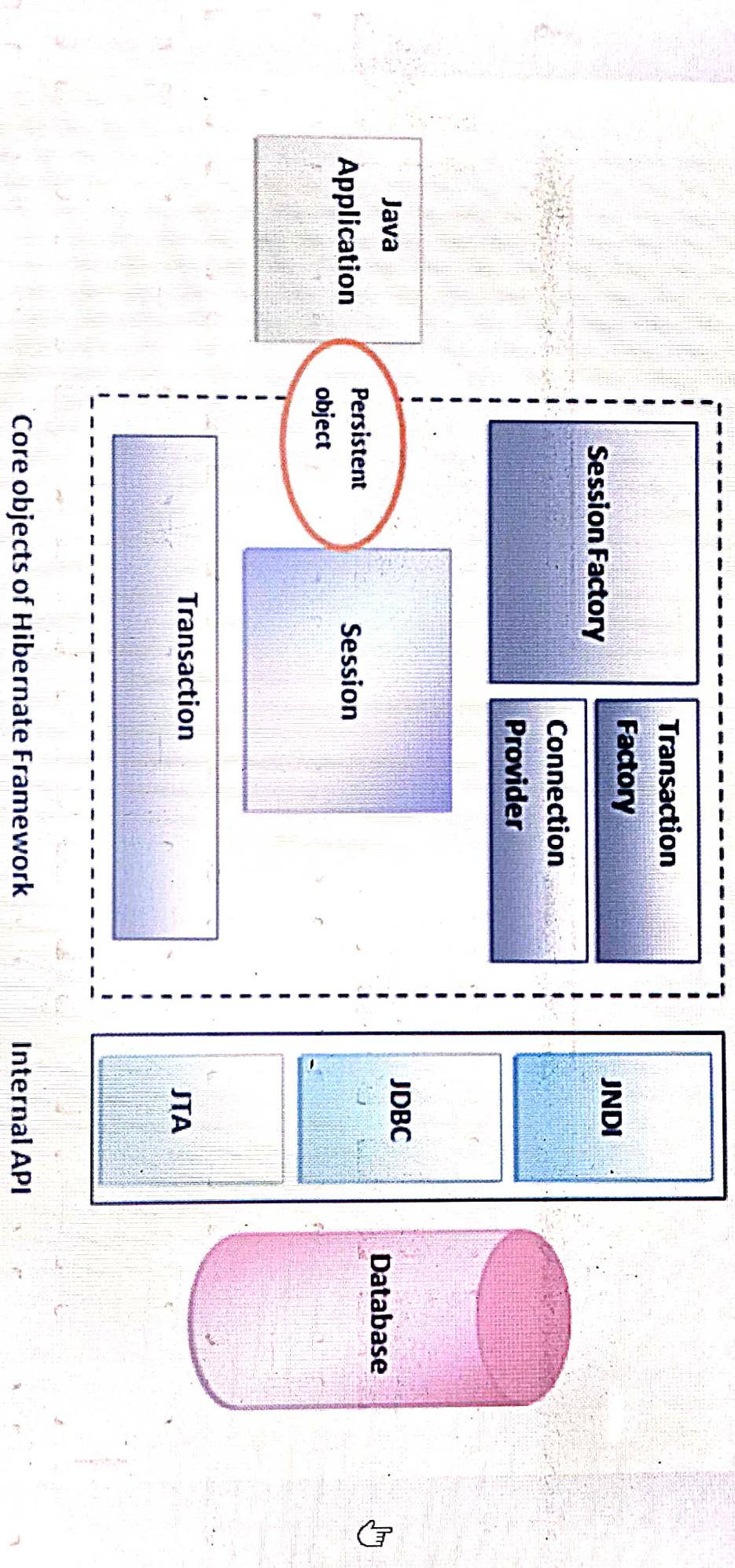
OBJECT

- The Hibernate architecture is layered to keep you isolated from having to know the underlying APIs.
- Hibernate makes use of the database and configuration data to provide persistence services (and persistent objects) to the application.
- High level view of the Hibernate Application Architecture.



Low level view of Hibernate Architecture

OBJECT



Introduction to Hibernate

In this chapter we will learn about the pioneer in ORMapping tools i.e. Hibernate. We will learn about how this tool helps in simplifying JDBC operations and the architecture of this tool.

What is ORM

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. It generates SQL calls and relieves the developer from the manual handling and object conversion of the result set.

What is ORM?

ORM stands for Object-Relational Mapping (ORM) is a programming technique for converting data between relational databases and object oriented programming languages such as Java, C#, etc.

Java ORM Frameworks

There are several persistent frameworks and ORM options in Java. A persistent framework is an ORM service that stores and retrieves objects into a relational database.

1. Enterprise JavaBeans Entity Beans
2. Java Data Objects
3. Castor
4. TopLink
5. Spring DAO
6. Hibernate
7. And many more

Introduction to Hibernate

Hibernate is an Object/Relational Mapping solution for Java environments. The term Object/Relational Mapping refers to the technique of mapping data from an object model representation to a relational data model representation (and vice versa).

Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. It can significantly reduce development time otherwise spent with

Introduction to Hibernate

Hibernate is an Object/Relational Mapping solution for Java environments. The term Object/Relational Mapping refers to the technique of mapping data from an object model representation to a relational data model representation (and vice versa). Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. It can significantly reduce development time otherwise spent with manual data handling in SQL and JDBC. Hibernate's design goal is to relieve the developer from 95% of common data persistence-related programming tasks by eliminating the need for manual, hand-crafted data processing using SQL and JDBC. Hibernate's primary feature is mapping from Java classes to database tables, and mapping from Java data types to SQL data types. Hibernate also provides data query and retrieval facilities.

Advantages and features

Advantage over JDBC

1. Developer has to write code in JDBC to map an object model's data representation to a relational data model and its corresponding database schema.
2. Hibernate itself takes care of this mapping using XML files so developer does not need to write code for this. JDBC supports only native Structured Query Language (SQL).
- Hibernate provides Hibernate Query Language (HQL) which is similar to SQL syntax and supports polymorphic queries too. It also supports native SQL statements.
3. The mapping of Java objects with database tables has to be taken care of in JDBC. Hibernate provides transparent persistence and therefore there is no need to map database tables tuples to application objects during interaction with RDBMS.
4. With JDBC, caching needs to be manually maintained. Hibernate cache is set to application work space. Relational tuples are moved to this cache as a result of query. It improves performance during multiple writes for the same data.
5. In JDBC there is no check that always every user has updated data. Hibernate enables definition of version type field to application, due to which Hibernate updates version field of database table every time relational tuple is updated in form of Java class object to that table.

Hibernate architecture



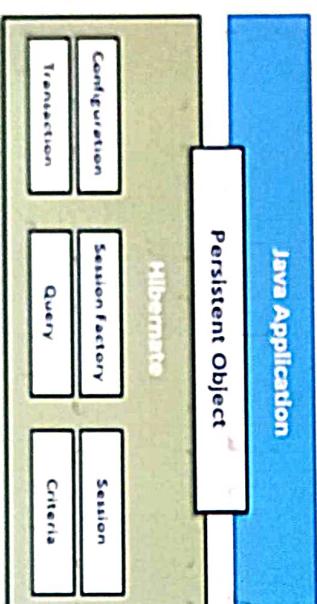
Data Access Layer

Hibernate architecture



Relational Database

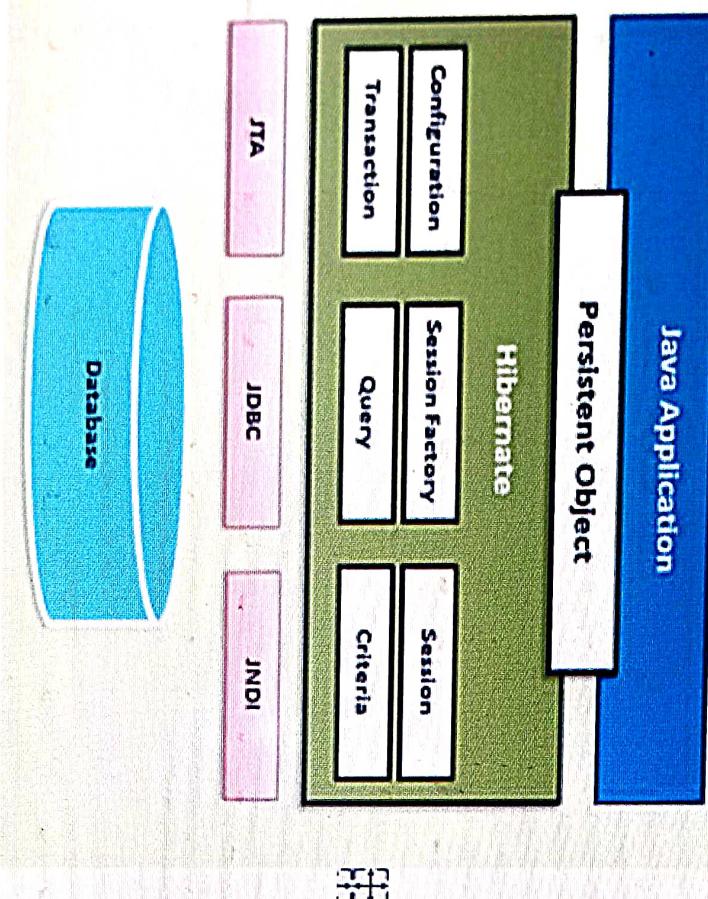
Hibernate, as an ORM solution, effectively "sits between" the Java application data access layer and the Relational Database, as can be seen in the diagram above. The Java application makes use of the Hibernate APIs to load, store, query, etc its domain data.



Assignments

1. Find out other ORMapping frameworks available.

Hibernate, as an ORM solution, effectively "sits between" the Java application data access layer and the Relational Database, as can be seen in the diagram above. The Java application makes use of the Hibernate APIs to load, store, query, etc its domain data.



Assignments

1. Find out other ORMapping frameworks available