Hibernate:

Hibernate is a **Java-based persistence framework** and an **object-relational mapping (ORM)** framework that basically allows a developer to map POJO - plain old Java objects - to relational database tables.

1. What is ORM in Hibernate?

Hibernate ORM stands for Object Relational Mapping. This is a mapping tool pattern mainly used for converting data stored in a relational database to an object used in object-oriented programming constructs. This tool also helps greatly in simplifying data retrieval, creation, and manipulation.

1. What are the advantages of Hibernate over JDBC?
2. Clean Readable Code
3. HQL (Hibernate Query Language):
4. Transaction Management: not automatically executing the query but only after the commit from developer

In the same way we can rollback

Commit(), rollback()

1. Exception Handling: Hibernate wraps the JDBC exceptions and throws unchecked exceptions like JDBCException or HibernateException.
2. Special Features: Hibernate supports OOPs features like inheritance, associations and also supports collections. These are not available in JDBC.
3. What are some of the important interfaces of Hibernate framework?

Configuration

SessionFactory

Session

Criteria

Query

Transaction

4) What is a Session in Hibernate?

A **session is an object** that maintains the **connection between Java object application and database**.

Session also has methods for storing, retrieving, modifying or deleting data from database using methods like

persist(), load(), get(), update(), delete(), etc.

Additionally, It has factory methods to return Query, Criteria, and Transaction objects.

5) What is a SessionFactory?

a) SessionFactory provides an instance of **Session**.

b) It is a factory class that gives the Session objects **based on the configuration parameters** in order to establish the connection to the database.

c) As a **good practice**, the application **generally** has a single instance of SessionFactory. - -- **SingleTon**

d) The internal state of a SessionFactory which includes metadata about ORM, and this ORM is **immutable**,

i.e once the instance is created, it cannot be changed.

e) This also provides the facility **to get information like** statistics and **metadata related to a class**, **query executions**, etc. It also **holds second-level cache data** if enabled.

6) What do you think about the statement - “session being a thread-safe object”?

No, Session is not a thread-safe object which means that any number of threads can access data from it simultaneously.

### 7) Can you explain what is lazy loading in hibernate?

### Lazy loading is mainly used for improving the application performance by helping to load the child objects only on demand.

### It is to be noted that, since Hibernate 3 version, this feature has been enabled by default.

### This signifies that child objects are not loaded until the parent gets loaded

### 8) What is the difference between first level cache and second level cache?

### 

### 🡪If an entity or object is loaded by calling the get() method then Hibernate first checked the first level cache,

### 🡪if it doesn’t find the object then it goes to the second level cache if configured.

### 🡪If the object is not found then it finally goes to the database and returns the object, if there is no corresponding row in the table then it returns null

### 9. What can you tell about Hibernate Configuration File?

### Hibernate Configuration File or hibernate.cfg.xml is one of the most required configuration files in Hibernate. By default, this file is placed under the src/main/resource folder.

### The file contains database related configurations and session-related configurations. Hibernate facilitates providing the configuration either in an XML file (like hibernate.cfg.xml) or a properties file (like hibernate.properties).

### This file is used to define the below information:

### Database connection details: Driver class, URL, username, and password.

### There must be one configuration file for each database used in the application, suppose if we want to connect with 2 databases, then we must create 2 configuration files with different names.

### Hibernate properties: Dialect, show\_sql, second\_level\_cache, and mapping file names.

### 10. How do you create an immutable class in hibernate?

### Immutable class in hibernate creation could be in the following way.

### If we are using the XML form of configuration, then a class can be made immutable by marking mutable=false.

### The default value is true there which indicating that the class was not created by default.

### In the case of using annotations, immutable classes in hibernate can also be created by using @Immutable annotation.

### Can you explain the concept behind Hibernate Inheritance Mapping?

### Relational databases do not support inheritance.

### They have a flat structure.

### Hibernate’s Inheritance Mapping strategies deal with solving how to hibernate being an ORM tries to map this problem between the inheritance of Java and flat structure of Databases.

### There are different inheritance mapping strategies available: Single Table Strategy

### Table Per Class Strategy

### Mapped Super Class Strategy

### Joined Table Strategy

### 12) Is hibernate prone to SQL injection attack?

### 🡺SQL injection attack is a serious vulnerability in terms of web security wherein an attacker can interfere with the queries made by an application/website to its database thereby allowing the attacker to view sensitive data which are generally irretrievable.

### 🡺Hibernate does not provide immunity to SQL Injection. However, following good practices avoids SQL injection attacks.

### 🡺It is always advisable to follow any of the below options:

### Incorporate Prepared Statements that use Parameterized Queries.

### Use Stored Procedures.

### Ensure data sanity by doing input validation

### INTERMEDIETE

### 