Assignment 8

Q1.What is ORM in Hibernate?

ans)ORM stands for Object-Relational Mapping. In the context of Hibernate, ORM refers to the technique of mapping object-oriented entities to relational database tables. Hibernate is an ORM framework for Java that provides a convenient way to interact with databases using object-oriented principles.

Q2.What are the advantages of Hibernate over JDBC?

ans)Advantages of Hibernate over JDBC:

1. Object-Relational Mapping: Hibernate provides a higher level of abstraction by mapping Java objects to database tables, eliminating the need for manual mapping and reducing the amount of boilerplate code.

2. Database Independence: Hibernate allows developers to write database-agnostic code, enabling seamless switching between different database systems without changing the application code.

3. Query Language: Hibernate offers HQL (Hibernate Query Language), an object-oriented query language that simplifies database queries and supports advanced querying capabilities.

4. Caching: Hibernate includes caching mechanisms that improve performance by reducing the need for frequent database access.

6. Transaction Management: Hibernate provides transaction management capabilities, ensuring data consistency and integrity.

7. Easier Maintenance: Hibernate reduces the amount of low-level JDBC code, making the codebase more maintainable and easier to understand.

8. Support for Lazy Loading: Hibernate supports lazy loading, which improves performance by loading associated entities only when needed.

Q3.What are some of the important interfaces of Hibernate framework?

ans)Some of the important interfaces of the Hibernate framework are:

1. SessionFactory: This interface represents a factory for creating Session objects. It is responsible for creating and managing database connections and providing a Session instance.

2. Session: The Session interface represents a single-threaded unit of work in Hibernate. It is used to perform database operations, such as saving, updating, or deleting entities, executing queries, and managing transactions.

3. Transaction: The Transaction interface provides methods to manage database transactions. It allows developers to control the boundaries of transactions, commit or rollback transactions, and control transactional behaviour.

4. Query: The Query interface is used to execute HQL (Hibernate Query Language) or native SQL queries. It provides methods for parameter binding, pagination, and result retrieval.

5. Criteria: The Criteria interface provides a more type-safe and object-oriented way to construct queries. It allows developers to build queries using criteria expressions, restrictions, and projections.

Q4.What is a Session in Hibernate?

ans)In Hibernate, a Session represents a single-threaded unit of work with the database. It serves as a gateway for performing database operations and managing the persistence of entities.

A Session allows you to perform various tasks, including saving, updating, or deleting entities, executing queries, and managing transactions. It provides methods to interact with the underlying database, such as persist(), save(), update(), delete(), and createQuery().

Q5.What is a SessionFactory?

ans)In Hibernate, a SessionFactory is a heavyweight object responsible for creating and managing database connections. It is a thread-safe and immutable representation of the database configuration and mapping metadata.

The SessionFactory is typically created during application startup and shared across the entire application. It is responsible for creating

individual Session objects, which in turn represent the units of work with the database.

The SessionFactory is responsible for managing database connections, caching metadata, generating SQL statements, and providing access to various configuration settings. It acts as a factory for creating Session instances and ensures efficient use of database resources.

Q6.What is HQL?

ans)HQL stands for Hibernate Query Language. It is a powerful object-oriented query language provided by Hibernate, which is used to perform database operations on persistent objects. HQL is similar to SQL (Structured Query Language), but instead of operating on database tables and columns, it operates on persistent entities and their properties.

Q7.What are Many to Many associations?

ans)Many-to-many association is a type of relationship between entities in which multiple instances of one entity can be associated with multiple instances of another entity. It represents a bidirectional relationship where both entities can have multiple references to each other.

Q8.What is hibernate caching?

ans)Hibernate caching is a mechanism used by the Hibernate ORM framework to improve application performance by reducing the number of database queries. It involves storing frequently accessed data

in memory so that subsequent requests for the same data can be served from the cache instead of querying the database.

Q9.What is the difference between first level cache and second level cache?

ans)

First-level cache	Second-level cache
Bound to a specific Hibernate session	Shared among multiple Hibernate Sessions
Enabled by default in Hibernate	Needs to be explicitly configured and enabled
Caches entities within the scope of a single session	Caches entities, queries, and other data structures
Provides automatic caching and transparent behaviour	Requires explicit configuration and management
Scoped to the lifetime of the Hibernate “Session”	Scoped to the lifetime of the Hibernate “SessionFactory”

Q10.What can you tell about Hibernate Configuration File?

ans)The Hibernate configuration file, typically named “hibernate.cfg.xml”, is an XML file used to configure the Hibernate framework. It contains essential information and settings required for Hibernate to

establish a connection with the database and manage the persistence of objects.

The Hibernate configuration file includes the following key elements:

1. Database connection details: It specifies the database driver, URL, username, password, and other necessary connection properties.

2. Mapping metadata: It defines the mapping between the Java classes/entities and the database tables.

3. Session factory configuration: It includes settings for various Hibernate features such as caching, transaction management, and connection pooling.

4. Dialect configuration: It specifies the SQL dialect for the database being used.

5. Additional configuration: It allows customization of various Hibernate settings and behaviours, such as logging, naming strategy, and query cache configuration.