Q1>What is ORM in Hibernate?

Ans == In the context of Hibernate, ORM stands for “Object-Relation Mapping”.it is a programming technique used to bridge the gap between onject-oriented programming (OOP) and relational databases, allowing developers to work with data in their appplications using object-oriented concepts rather than dealing directly with database-specific operation.

Q2>What are the advantages of Hibernate over JDBC?

Ans == Hibernate offers several advantages over using JDBC (java database Connectivity) directly.

1>Object-Relational Mapping(ORM): With Hibernate, you work with java objects directly, and the framework handles the mapping between objects and database tables.

2>Database Independence: Hibernate provides database abstraction,allowing you to switch between different database systems(e.g,MySQL,PostgreSQL,Oracle) without changing the java code.you only need to modify the configuration, and Hibernate takes care of the database-specific differences.

4>Automatic Table Creation and Schema Management: Hibernate can automatically create database tables based on your entity classes,reducing the manual effort required in setting up the database schema.Additionaly,it can handle schema evolution,making it easier to update the database schema as your application evolves.

Q3>What are some of the important interface of Hibernate framework?

Ans == some of the important interface of Hibernate framework are:

1>SessionFactory: This interface is responsible for creating Session objects,which are used to interact with the database.It is a thread-safe object and typically initialized once during application startup.The SessionFactory holds configuration settings and mappings,enabling it to create and manage database connections efficiently.

2>Session: A Session represents a single unit of work with the database.It is a lightweight object used to perform database operations,such as persisting, retrieving,updating,and deleting objects.Sessions are created from the SessionFactory and are not thread-safe, so each thread should have its own Session.

3>Transaction: The Transaction interface is used for managing database transactions.It allows you to begin,commit,rollback,and control the boundries of database transactions.

4>Query: The Query interface is used to execute database queries and retrieve data from the database.It supports both HQL(Hibernate Query Language) and native SQL queries.

Q4>What is a Session in Hibernate?

Ans == In Hibernate, a Session is a crucial component representing a single unit of work with the database.It serves as an intermediary between your java application and the underlying database and is responsible for managing the oersistence of objects (entities) to the database and retrieving data from it.

Q5> What is a SessionFactory?

Ans == In Hibernate, a SessionFactory is a crucial component that represents a factory for creating Session instances.It is a thread-safe and immutable object that is typically created during the application's startup.The SessionFactory is responsible for managing the configuration settings,mapping metadata, and providing a centralized mechanism for creating and caching Session objects.

Q6>What is HQL?

Ans == HQL stands for Hibernate Query Language.It is a powerful and objects-oriented query language provided by Hibernate,which allows developers to write database queries using java-based entity names and properties instead of native SQL queries.HQL is similar to SQL in terms of syntax but is focused on querying and manipulating java objects(entities) rather than database tables.

Q7>What are many to many associations?

Ans == Many-to-many association is a type of relationship between entities in a relational database model,where multiple instance of one entity are associated with multiple instances of another entity.In other words, it is a bi-directional relationship where both entities can have multiple related instances.

Q8>What is Hibernate caching?

Ans == Hibernate caching is a mechanism provided by the Hibernate ORM framework to store frequently accessed data in memory.The main purpose of coaching is to improve the performance and efficiency of database interactions by reducing the number of trips to the database.

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

Ans == First-Level Cache(Session cache):

1> The first-level cache is associated with a single Hibernate Session.It exists within the scope of a single Session and is used to cache entities and their state for the duration of that Session.

2> The first-level cache lives as long as the Hibernate Session is active.When the Session is closed, the first-level cache is cleared,and the cached entities are no longer available for subsequent Sessions.

3> The first-level cache operates at the entity instance level.It caches individual objects with their unique identifiers (primary keys) to ensure each entity is represented only once in the cache.

Second-Level Cache(Session Cache):

1> The Second-level Cache, on the other hand, is shared across multiple Hibernate Sessions.it spans the entire SessionFactory and allows caching of entities or query result sets that can be reused across different Sessions.

2> The second-level cache operates at a coarser level than the first-level cache. It caches collections of entities or query result sets,potentially involving multiple entities.

3> The second-level cache persists across multiple Sessions and is independent of any individual Session's lifecycle.Cached data can be shared among different Sessions until the data is invalidated or evicted from the cache.

Q10>What can you tell about Hibernate Configuration file?

Ans == In Hibernate,the configuration file is usually written in XML formate.However,starting from Hibernate 4.x, you can also use a java-based configuration approach using annotations and the 'Configuration' class.