--------------------Hibernate----------------

What is hibernate?

Hibernate is an open-source and lightweight ORM tool that is used to store, manipulate, and retrieve data from the database.

1.What is ORM?

----------------

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.

2.What is Session and SessionFactory?

---------------------------------------------

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

Session is only available for particular transaction. Session is short-lived

Session provides a first level cache

SessionFactory provides an instance of Session. It is a factory class that gives the Session objects based on the configuration parameters in order to establish the connection to the database.

As a good practice, the application generally has a single instance of SessionFactory. The internal state of a SessionFactory which includes metadata about ORM is immutable, i.e once the instance is created, it cannot be changed.

It is available for the whole application

SessionFactory provides a second level cache

3.What is First-level cache and Secondlevel cache?

-----------------------------------------------------------

First-level

-----------

.Hibernate caches query data to make our application faster and improves performance

.The idea behind cache is to reduce the number of database queries

.Hibernate first-level cache is associated with the session-object

.first-level cache is enabled by default

.clear the cache completly

.when the session is closed all the cached objects will also be lost

Secondlevel

----------------

.Hibernate second-level cache is disabled by default but we can enabled it through configurations

.This cache is maintained at the SessionFactory level and shared among all sessions in Hibernate.

.The second-level cache is available through the application’s life cycle, it is only destroyed and recreated when an application is restarted.

4.difference between get() and load()?

-----------------------------------------

get()

----------

.Returns null if an object is not found.

.get() method always hit the database.

.It returns the real object, not the proxy.

.It should be used if you are not sure about the existence of instance.

load()

--------

.Throws ObjectNotFoundException if an object is not found.

.load() method doesn't hit the database.

.It returns proxy object.

.It should be used if you are sure that instance exists.

5.What is mean by Lezzy loading and dirty checking?

-------------------------------------------------------

Lazy loading

--------------

.Lazy loading in hibernate improves the performance. It loads the child objects on demand.

.Since Hibernate 3, lazy loading is enabled by default, and you don't need to do lazy="true". It means not to load the child objects when the parent is loaded.

dirty checking

-----------------

Hibernate automatically detects object state changes in order to synchronize the updated state with the database, this is called dirty checking.

6.What is Transaction Management?

------------------------------------------

.A transaction simply represents a unit of work. In such case, if one step fails, the whole transaction fails (which is termed as atomicity). A transaction can be described by ACID properties (Atomicity, Consistency, Isolation and Durability).

.A transaction is associated with Session and instantiated by calling session.beginTransaction()

7.Difference between Merge() and Update()?

--------------------------------------------

.Update means to edit something.

.update() should be used if the session doesn't contain an already persistent state with the same id. It means an update should be used inside the session only. After closing the session, it will throw the error.

.Merge means to combine something.

.merge() should be used if you don't know the state of the session, means you want to make the modification at any time.

8.What about hibernate Mapping configuration file?

--------------------------------------------------------

.The mapping document is an XML document having <hibernate-mapping> as the root element, which contains all the <class> elements.

.The <class> elements are used to define specific mappings from a Java classes to the database tables

.Database Connection

.Class Mapping Setup

9.Can you tell the difference between getCurrentSession and openSession methods?

------------------------------------------------------------------------------------------

getCurrentSession()

-----------------------

.It creates a new Session if not exists, else uses same session which is in current hibernate context.

.You do not need to flush and close session objects, it will be automatically taken care by Hibernate internally.

.In single threaded environment it is faster than openSession().

.You need to configure additional property. "hibernate.current\_session\_context\_class" to call getCurrentSession() method, otherwise it will throw an exception.

openSession()

----------------

.It always creates a new Session object.

.You need to explicitly flush and close session objects.

.In single threaded environment it is slower than getCurrentSession().

.You do not need to configure any property to call this method.

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

-----------------------------------------------------------------------

.Configuration

.SessionFactory

.Session

.Query

.Criteria

.Transaction

11.How do you create an immutable class in hibernate?

--------------------------------------------------------

If you mark a class as mutable="false", the class will be treated as an immutable class. By default, it is mutable="true".

12.Mention some of the advantages of using ORM over JDBC?

-------------------------------------------------------------

ORM has the following advantages over JDBC:

.Application development is fast.

.Management of transaction.

.Generates key automatically.

.Details of SQL queries are hidden.

or

.It allows business code access the objects rather than Database tables.

.It hides the details of SQL queries from OO logic.

.It is based on JDBC “under hood”.

.Dealing with database implementation is not required.

.Entities are based on business concepts rather than database structures.

.It generates the automatic key and Transaction management.

.Application development is faster.

13.Is Session a thread-safe object?

----------------------------------------

No, Session is not a thread-safe object, many threads can access it simultaneously. In other words, you can share it between threads.

SessionFactory

----------------

Yes, SessionFactory is a thread-safe object, many threads cannot access it simultaneously.

14.What are the states of the object (or) persistent entity in hibernate?

--------------------------------------------------------

There are 3 states of the object (instance) in hibernate.

Transient: The object is in a transient state if it is just created but has no primary key (identifier) and not associated with a session.

Persistent: The object is in a persistent state if a session is open, and you just saved the instance in the database or retrieved the instance from the database.

Detached: The object is in a detached state if a session is closed. After detached state, the object comes to persistent state if you call lock() or update() method.

15.What is HQL (Hibernate Query Language)?

--------------------------------------------------

Hibernate Query Language is known as an object-oriented query language. It is like a structured query language (SQL).

The main advantage of HQL over SQL is:

.You don't need to learn SQL

.Database independent

.Simple to write a query

16.Difference between save() and saveOrUpdate() methods in hibernate session.

--------------------------------------------------------------------------------------

save() generates a new identifier and INSERT record into a database

The insertion fails if the primary key already exists in the table.

The return type is Serializable which is the newly generated identifier id value as a Serializable object.

This method is used to bring only a transient object to a persistent state.

Session.saveOrUpdate() can either INSERT or UPDATE based upon existence of a record.

In case the primary key already exists, then the record is updated.

The return type of the saveOrUpdate() method is void.

This method can bring both transient (new) and detached (existing) objects into a persistent state. It is often used to re-attach a detached object into a Session

17.What is criteria API in hibernate?

----------------------------------------------

.Criteria API in Hibernate helps developers to build dynamic criteria queries on the persistence database. Criteria API is a more powerful and flexible alternative to HQL (Hibernate Query Language) queries for creating dynamic queries.

.The hibernate criteria API is very Simplified API for fetching data from Criterion objects. The criteria API is an alternative of HQL (Hibernate Query Language) queries.

-->Define criteria in terms of Hibernate.

.The objects of criteria are used for the creation and execution of the object-oriented criteria queries

19.What is hibernate caching?

--------------------------------------

What is meant by Hibernate caching?

Hibernate caching acts as a layer between the actual database and your application. It reduces the time taken to obtain the required data — as it fetches from memory instead of directly hitting the database.

Hibernate caching improves the performance of the application by pooling the object in the cache. It is useful when we have to fetch the same data multiple times.

.First Level Cache

.Second Level Cache

22.What is the difference between session.save() and session.persist() method?

------------------------------------------------------------------------------------

The persist() method will not execute an insert query if it is called outside of transaction boundaries.

the save() method returns an identifier so that an insert query is executed immediately to get the identifier, no matter if it are inside or outside of a transaction.

24.What are the advantages of Hibernate over JDBC?

--------------------------------------------------------------

The advantages of Hibernate over JDBC are listed below:

Clean Readable Code: Using hibernate, helps in eliminating a lot of JDBC API-based boiler-plate codes, thereby making the code look cleaner and readable.

HQL (Hibernate Query Language): Hibernate provides HQL which is closer to Java and is object-oriented in nature. This helps in reducing the burden on developers for writing database independent queries. In JDBC, this is not the case. A developer has to know the database-specific codes.

Transaction Management: JDBC doesn't support implicit transaction management. It is upon the developer to write transaction management code using commit and rollback methods. Whereas, Hibernate implicity provides this feature.

Exception Handling: Hibernate wraps the JDBC exceptions and throws unchecked exceptions like JDBCException or HibernateException. This along with the built-in transaction management system helps developers to avoid writing multiple try-catch blocks to handle exceptions. In the case of JDBC, it throws a checked exception called SQLException thereby mandating the developer to write try-catch blocks to handle this exception at compile time.

Special Features: Hibernate supports OOPs features like inheritance, associations and also supports collections. These are not available in JDBC.

25.What can you tell about Hibernate Configuration File?

------------------------------------------------------------

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.

26.Can we declare the Entity class final?

-----------------------------------------------------

No, we should not define the entity class final because hibernate uses proxy classes and objects for lazy loading of data and hits the database only when it is absolutely needed. This is achieved by extending the entity bean. If the entity class (or bean) is made final, then it cant be extended and hence lazy loading can not be supported.