Hibernate

1) What is hibernate?

Ans:- Hibernate is an open source and light weight **ORM tool** that is used to store, manipulate and retrieve data from data base.

2) What is ORM?

Ans: - RM is an acronym for Object/Relational mapping. It is a programming strategy to map object with the data stored in the database. It simplifies data creation, data manipulation and data access.

- Hibernate is an ORM (Object Relational Mapping) and persistent framework.
- The framework helps to map plain java object to relational database table using xml configuration file.
- The framework helps to perform following things.

a. Perform basic CURD operations.

b. Write queries referring to java classes (HQL queries).

c. Facilities to specify metadata.

d. Dirty checking, lazy association fetching.

3. Why hibernate and how does it help in the programming?

The main advantage of Hibernate (ORM) framework is that it shields developer to write a messy SQL. Apart from that ORM provides following benefits.

1. Improve Productivity of the developer by providing high level object oriented API (e.g. API for easily maintaining the connection to data base, mapping java classes to relational database tables), less java code to write, helps to avoid writing SQL query.
2. Improved performance by providing sophisticated caching, lazy loading and eager loading features.
3. Provide portability, the framework helps to generate database specific SQL for you.

4. How will you configure Hibernate?

- To configure hibernate, you need hibernate.cfg.xml or hibernate.properties file and *.hbm.xml files, all these files are used by configuration class to create sessionFactory, which in turn creates the session instances.
- Session instances are the primary interface for persistence services.
- The hibernate.cfg.xml or hibernate.properties files are used to configure the hibernate service (database connection driver class, database connection URL, connection username, connection password, dialect, mapping resources etc.).
- The *hbm.xml files are used for mapping persistent objects to relational database.
- From Java 5 onwards you can configure and map persistent objects through annotations.

5. Which settings will be loaded if both hibernate.Properties and hibernat.cfg.xml files are present in the classpath?

If both hibernate.properties and hibernate.cfg.xml files are present in the classpath then hibernate.cfg.xml file will override the settings found in hibernate.properties. So please make sure that your project should include either hibernate.properties or hibernate.cfg.xml file.

6. What are the Core interfaces of Hibernate framework?

There are **"5 core interfaces"** being used extensively in every Hibernate application. Using these interfaces you can store or retrieve any persistent objects and also control transactions.

1. Session interface: - It maintains a **connection** between **hibernate application** and **database**.

It provides methods to store, update, delete or fetch data from the database such as persist (), update (), delete (), load (), get () etc. It is a factory of Query, Criteria and Transaction i.e. it provides factory methods to return these instances.

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

2. SessionFactory interface: - SessionFactory provides the instance of Session. It is a **factory of Session**. It holds the data of **second level cache** that is not enabled by default.
You can get the instance of SessionFactory by the configuration object as below:

SessionFactory sessionFactory = configuration.buildSessionFactory ();

(Yes, **SessionFactory** is a **thread-safe object**, many threads cannot access it simultaneously)

3. Configuration interface:-

4. Transaction interface:-
5. Query and Criteria interfaces:-

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

There are **2 type** of cache in java: - 1. **1st level cache** 2. **2nd level cache**

Hibernate uses two different type of caches for objects: first-level cache and second-level cache. First level of cache is associated with Session object, while second-level of cache is associated with the SessionFactory object. By default, Hibernate uses first-level of cache on a per-transaction basis. Hibernate mainly use this cache to reduce the number of SQL queries it needs to generate within a given transaction.

No.	First Level Cache	Second Level Cache
1)	First Level Cache is **associated with Session**.	Second Level Cache is associated with **SessionFactory**.
2)	It is **enabled** by default.	It is **not enabled** by default.

8. What are the states of object in hibernate?

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

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

2. **Persistent**: The object is in persistent state if session is open, and you just saved the instance in the database or retrieved the instance from the database.
3. **Detached**: The object is in detached state if session is closed. After detached state, object comes to persistent state if you call lock() or update() method.

9. What are the Collection types in Hibernate?

1. Bag
2. Set
3. List
4. Array
5. Map

10. Type of hibernate file:- 1.HVM 2.CFG

11. What the difference is between get and load method?

No.	get()	load()
1)	Returns **null** if object is not found.	Throws **ObjectNotFoundException** if object is not found.
2)	get() method always **hit the database**.	load() method **doesn't hit** the database.
3)	It returns real object **not proxy**.	It returns **proxy object.**
4)	It should be used if **you are not sure** about the existence of instance.	It should be used if **you are sure** that instance exists.

12. What is the difference between update and merge method?

No.	update() method	merge() method
1)	Update means to edit something.	Merge means to combine something.
2)	update() should be used if session doesn't contain an already persistent state with same id. It means update should be used inside the session only. After closing the session it will throw error.	merge() should be used if you don't know the state of the session, means you want to make modification at any time.

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

No.	save()	persist()
1)	returns the identifier (Serializable) of the instance.	return nothing because its return type is void.
2)	Syn: public Serializable save(Object o)	Syn: public void persist(Object o)

14. Session and Session factory
15. Is Session factory is thread safe -- YES

16) How to make a immutable class in hibernate?

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

17. How many types of association mapping are possible in hibernate?

There can be 4 types of association mapping in hibernate.

1. One to One
2. One to Many
3. Many to One
4. Many to Many

18. Lazy loading in Hibernate?

Lazy loading in hibernate improves the performance. It loads the child objects on demand.Since Hibernate 3, lazy loading is enabled by default, you don't need to do lazy="true". It means not to load the child objects when parent is loaded.

19. HQL(Hibernate Query Language):-

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

The main advantage of HQL over SQL is:

1. You don't need to learn SQL
2. Database independent
3. Simple to write query

20. Difference between HQL and SQL

SQL	HQL
1) It is based on relational database model.	1) It is combination of OOPs with Relational database concept
2) It manipulates data stored in tables and modifies its rows and columns.	2) It is concerned about objects and its properties
3) It is concerned about the relationship that exists between table	3) It is concerned about relation between objects.

21. How can we make class as immutable in hibernate?
Ans- we can make immutable class in java using mutable = false

22.How does Hibernate useful in mapping object and its relations?
- Hibernate is a framework for Java that is used to map the object and its relations.
- It is used to build persistent objects using the terminologies of JAVA.
- It uses the following properties to perform the function of mapping:

1. Association : Where the one object can be associated with another object of the same or different class.
2. Inheritance : It is a property of inheriting the properties and features of the parent class.
3. Polymorphism : It is a feature that allows more than one data type to be handled by a

single function.
4. Composition : It is used to combine the simple data types into complex data types.

23. What is Hibernate Template?

The spring framework provides HibernateTemplate (org.springframework.orm.hibernate.HibernateTemplate) which is kind of helper class and provides following benefits.

- HibernateTemplate class simplifies interaction with Hibernate session.
- Common functions are simplified to single method calls.
- Sessions are automatically closed.
- Exception are automatically caught and converted to runtime exceptions.

24) What are the inheritance mapping strategies?

There are 3 ways of inheritance mapping in hibernate.

1. Table per hierarchy
2. Table per concrete class
25. Table per subclass
26.

25. Why are callback interfaces useful in Hibernate?

Callback interfaces are useful as it allows the application to receive important notification about the objects that are in execution phase. It includes the objects that are loaded, saved or deleted from the Hibernate. Callback interfaces are very important to implement the general functionality like audit records, etc. It sends a notification when any object even occurs. It allows the programmer to get the error information or exception handling can be done in a better way to notify the user on run time in case of any problem in the programming code.

26.What is the importance of Lifecycle and validatable interface in Hibernate?

Lifecycle and validatable interface allows the object that is persistent to react to certain events that occurs during their own persistence lifecycle. The lifecycle is managed by the CRUD operations which handles the object lifecycle as well. Hibernate uses ORM solution that are similar in functionality of the callback interface. It allows easy portable code for the transaction to be saved and persistent classes and objects can be validated and sent over for implementation and use.

27.What are the different types of extension interface that is present in Hibernate?

Extension interface allows easy implementation of other interfaces according to the user's requirements. It allows different functionality to be provided when the built in functionality is not sufficient for the users. The extensions that are provided for the use is as follows :

ProxyFactory interface : It is used to create proxies to help in maintaining the mapping of the objects.
ConnectionProvider interface : It is used to provide the JDBC connection management for easy mapping of the object with the database.
TransactionFactory interface : It is used to provide the transaction management.
TransactionManagementLookup interface : It is used to provide the way to look up the transactions that is being managed by transaction management interface.
Cahce interface : It is used to provide the caching techniques and strategies for the objects to speed up the process.

IdentifierGenerator interface : It is used to provide a way to generate the keys like primary key to uniquely identify the objects.

** Is it possible to perform collection mapping with "one-to-one" and "many-to-one" ?

→ No, collection mapping can only be performed with "one-to-many" and "many-to-many".

** what