

Hibernate Interview Questions - Part-1

1.What is Hibernate ?

Hibernate is an open source and a pure Java object-relational mapping (ORM) and persistence framework that allows you to map plain old Java objects to relational database tables using (XML) configuration files. Hibernate not only takes care of the mapping from Java classes to database tables but also provides data query and retrieval facilities and can significantly reduce development time. Hibernate's goal is to relieve the developer from 95 percent of common data persistence related programming tasks.

2. Why Should You Use Hibernate ?

Reasons:

Performance :

1. Hibernate employs very aggressive, and very intelligent first and second level caching strategy. This is a major factor in achieving the high scalability.

2. Hibernate spares you a unnecessary database calls and in persistence state it is automated persistence then it still increases the performance.

Productivity, Maintainability, Portability

1.Portability : Hibernate's portability across the relational databases is amazing. It is literally one configuration parameter change. You only have to change the database dialect.

2.Productivity – Hibernate reduces the burden of developer by providing much of the functionality and let the developer to concentrate on business logic.

3.Maintainability – As hibernate provides most of the functionality, the LOC for the application will be reduced and it is easy to maintain. By automated object/relational persistence it even reduces the LOC.

Cost Effective : Hibernate is free and open source – Cost Effective

Learning curve is short :Since we all have working experience in using Hibernate, and Hibernate is totally object orientated concept, it will shorted our learning curve.

Learning curve is short :Since we all have working experience in using Hibernate, and Hibernate is totally object orientated concept, it will shorted our learning curve.

3.What are the advantages of Hibernate ?

1.Database Independent: Hibernate is database independent and you can use any database of your choice.

2.Caching Framework: There are many caching framework that works with Hibernate. You can use any one in your application to improve the performance of your application.

3.As Hibernate is set of Objects , you don't need to learn SQL language. You can treat TABLE as a Object . In case of JDBC you need to learn SQL.

4. Don't need Query tuning in case of Hibernate. If you use Criteria Queries in Hibernate then hibernate automatically tuned your query and return best result with performance.

5. Automatic primary key generation and Object/Relational mapping definition and High Performance

6. In the xml file you can see all the relations between tables in case of Hibernate. Easy readability.

7. You can load your objects on start up using lazy=false in case of Hibernate. JDBC Don't have such support.

8. Hibernate Supports automatic versioning of rows but JDBC Not.

4. What are the disadvantages of using hibernate ?

1- For complex data, mapping from Object-to-tables and vice versa reduces performance and increases time of conversion(entity to object conversion).

2- Hibernate does not allow some type of queries which are supported by JDBC. For example It does not allow to insert multiple objects (persistent data) to same table using single query. Developer has to write separate query to insert each object.

3- Debugging: Sometimes debugging and performance tuning becomes difficult.

4- Slower than JDBC: Hibernate is slower than pure JDBC as it is generating lots of SQL statements in runtime.

5- Not suitable for Batch processing: It advisable to use pure JDBC for batch processing.

6-Composite mapping is complex : If you do not understand it , yes it is complex . But this will not be a disadvantage. We have talked about this here mapping composite keys in hibernate.

5.What are the hibernate features ?

- 1.Support POJO/POJI model programming .
- 2.Light weight technology to develop DB s/w independent persistence logic .
- 3.Allows work with any java,j2ee,framework s/w`s based applications to them interaction with DB s/w.
- 4.Gives built in transaction management , connection pooling support.
- 5.Allows to work with 3rd party jdbc connection pool s/w like c3po,proxol and etc....
- 6.Supports two level caching to reduce network round trips of client APP and DB s/wOne internet explorer browser can share buffer of another browser.
- 7.Allows to callpl/sql procedures and functions.
- 8.Gives HQL as database s/w independent query language to perform persistence operations.
- 9.Allows to work with database specific native SQL to perform persistence operations.
- 10.Allows object level relationships in the development of persistence logic when tables are there in 11.relationship like one-one,one-many,many-one,many-many etc.

12. Gives special data structures like bag etc... to support object level relationships.

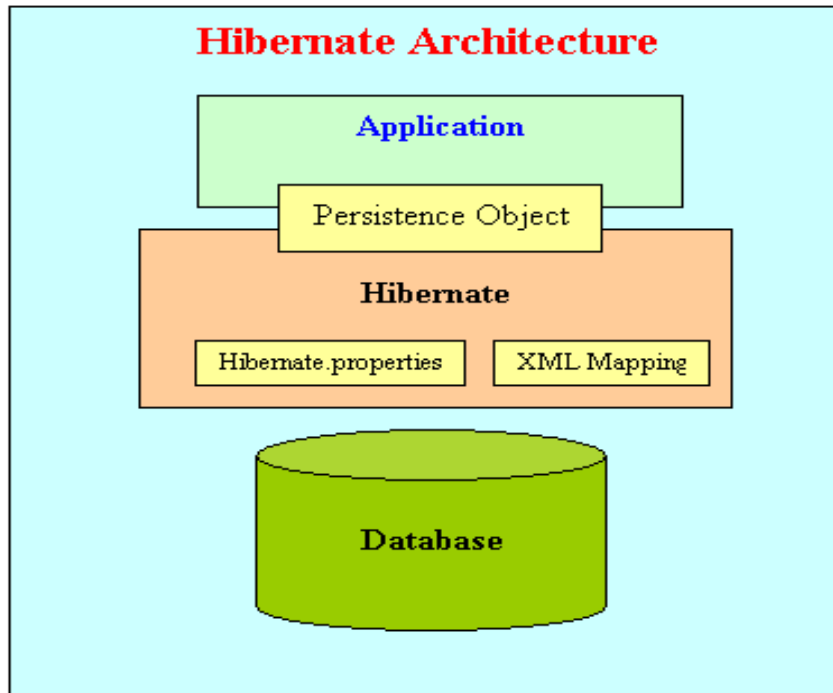
13. Easy to learn and easy to apply.

6. Hibernate Architecture

Hibernate:

1. itself opens connection to database,
2. converts HQL (Hibernate Query Language) statements to database specific statement,
3. receives result set,
4. then performs mapping of these database specific data to Java objects which are directly used by Java application.

Hibernate uses the database specification from Hibernate Properties file. Automatic mapping is performed on the basis of the properties defined in hbm XML file defined for particular Java object.



7. What is the general flow of Hibernate communication with RDBMS ?

General steps:

1 . Load the Hibernate configuration file and create configuration object. It will automatically load all hbm mapping files.

```
Configuration cfg = new  
Configuration().configure(CONFIG_FILE_LOCATION);
```

2 . Create session factory from configuration object

```
SessionFactory sessionFactory = cfg.buildSessionFactory();
```

3 . Get one session from this session factory.

```
Session ses=factory.opensession()
```

4 . Create HQL query.

```
Query query = session.createQuery("from EmployeeBean");
```

5 . Execute query to get list containing Java objects.

```
ListfinalList = query.list();
```

8.what is the difference between jdbc and hibernate ?

1. Hibernate is data base independent, your code will work for all ORACLE,MySQL ,SQLServeretc.In case of JDBC query must be data base specific.
2. As Hibernate is set of Objects , you don?t need to learn SQL language.You can treat TABLE as a Object . Only Java knowledge is need.In case of JDBC you need to learn SQL.
3. Don?t need Query tuning in case of Hibernate. If you use Criteria Quieres in Hibernate then hibernate automatically tuned your query and return best result with performance.In case of JDBC you need to tune your queries.
4. You will get benefit of Cache. Hibernate support two level of cache. First level and 2nd level. So you can store your data into Cache for better performance.In case of JDBC you need to implement your java cache .
5. Hibernate supports Query cache and It will provide the statistics about your query and database status.
6. Hibernate Supports automatic versioning of rows but JDBC Not.
7. Development fast in case of Hibernate because you don?t need to write queries

8. No need to create any connection pool in case of Hibernate. You can use c3p0. In case of JDBC you need to write your own connection pool

9. In the xml file you can see all the relations between tables in case of Hibernate. Easy readability.

9. What is ORMsoftwares ?

1. Object-relational mapping software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.

2 . All ORM softwares internally use Jdbc code to perform persistence operation on database table rows

3. ORM softwares are given only to develop persistence logic and they are not suitable to develop other logics like presentation logic,business Logic...etc.

4. ORM softwares are responsible for the synchronization between the object and table row and to develop objects based "O-R Mapping Persistence Logic".

10. What is ORM ?The process of mapping java class with database table and class member variables with table columns and making that java class objects representing database table rows by having synchronization between them is called O-R Mapping. Synchronization between object and table row is nothing but modification done in java application will reflect in table row and vice versa.

11.What are the Core interfaces are of Hibernate framework?

The five core interfaces are used in just about every Hibernate application. Using these interfaces, you can store and retrieve persistent objects and control transactions.

- 1 . Session interface
- 2 . SessionFactory interface
- 3 . Query .
- 4Transaction and Criteria interfaces

12. Explain the role of Session interface in Hibernate.

- 1 . A Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database.
- 2 . This session object represents persistence context where persistence state hibernate pojo class objects reside representing table rows.
- 3 . Hibernate session object is the base for the programmer to provide instructions to hibernate s/w to perform persistence operations on the table rows based on the operations performed on the pojo class objects.
- 4 . Hibernate session object means it is the object of a class that implements org.hibernate.Session interface.

13. Explain the role of SessionFactory interface in Hibernate ?

The SessionFactory is used to create Sessions. Each application is having usually only one SessionFactory. The requests from the servicing client threads obtain the Sessions from the session factory. SessionFactory is thread safe, so that many threads can access it concurrently and requests the sessions.

14. Explain the role of Configurationclass in Hibernate ?

The org.hibernate.cfg.Configuration is used to build an immutable org.hibernate. SessionFactoryobject . Configuration class object activates hibernate software and configure () is the factory method of hibernate.cfg.Configuration class which reads configuration properties from hibernate.cfg.xml file. buildSessionFactory() method uses hibernate.cfg.xml properties of Configuration object Creates jdbc connection pool.

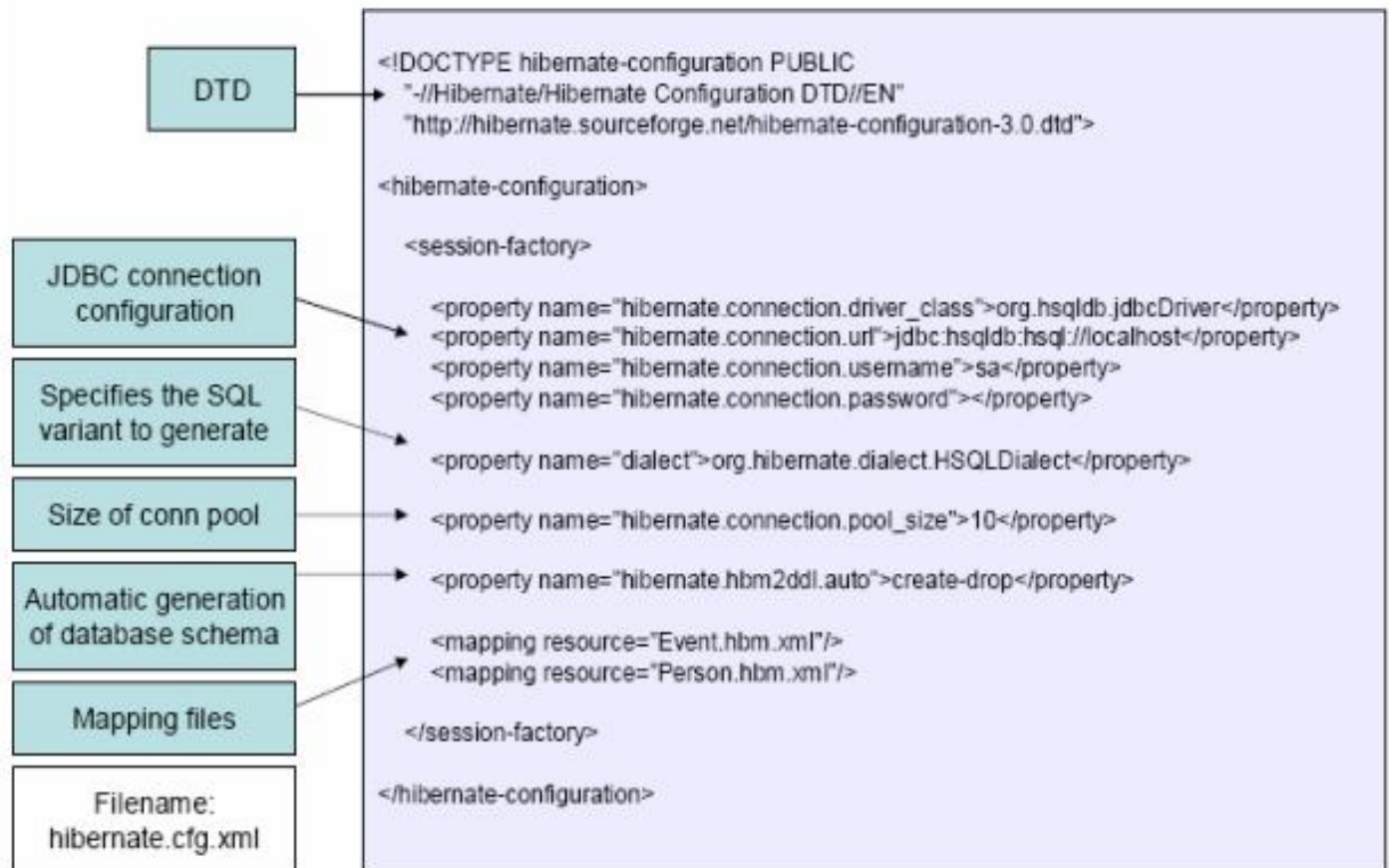
15. Explain the role of Transaction Interface in Hibernate ?

A transaction is associated with a Session and is usually instantiated by a call to session .beginTransaction(). A single session might span multiple transactions since the notion of a session (a conversation between the application and the datastore) is of coarser granularity than the notion of a transaction. However, it is intended that there be at most one uncommitted Transaction associated with a particular Session at any time.

16.What are the important tags of hibernate.cfg.xml?

This configuration file contains details connect to database s/w like driver classes, database url, database username, password and etc..all these details we should pass as the values of hibernate configuration properties . These property names are fixed but values will be passed on the database s/w we use.The following

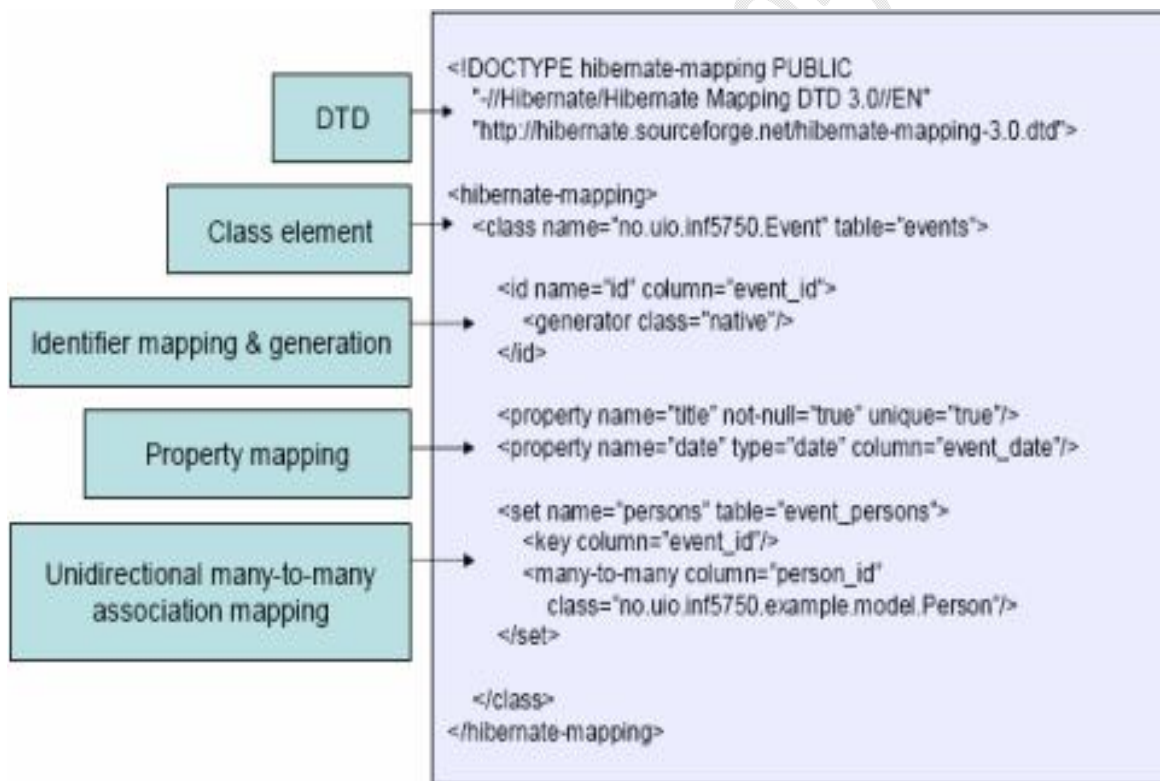
properties are minimum properties of hibernate configuration file for any database.



17. What is the need for Hibernate xml mapping file?

Any .xml can act as hibernate mapping file. This file name should specify in hibernate configuration file. There is no default name for this file and this file contains various types of O-R mapping configurations like O-R mapping, collection mapping, association mapping, inheritance mapping and etc.

Any Basic O-R mapping means mapping class with table and class member variables with db column names. This file is base file for hibernate s/w to understand O-R mapping configuration related to hibernate persistence classes.



18. What is the use of hibernate dialect property?

This property takes hibernate s/w supplied database s/w specific class name as the value. So this class name will change based on the database s/w and its version that we use in hibernate

application. `Org.hibernate.dialect.property` values helps hibernate s/w to generate and assign intelligent and sensible default values for some hibernate configuration properties (when they are not specified in configuration file) based on the database s/w.

Dialect class names :

Oracle (any version): `org.hibernate.dialect.OracleDialect`

Oracle 9i : `org.hibernate.dialect.Oracle9iDialect`

Oracle 10g : `org.hibernate.dialect.Oracle10gDialect`

19. What are the different states of Hibernate object ?

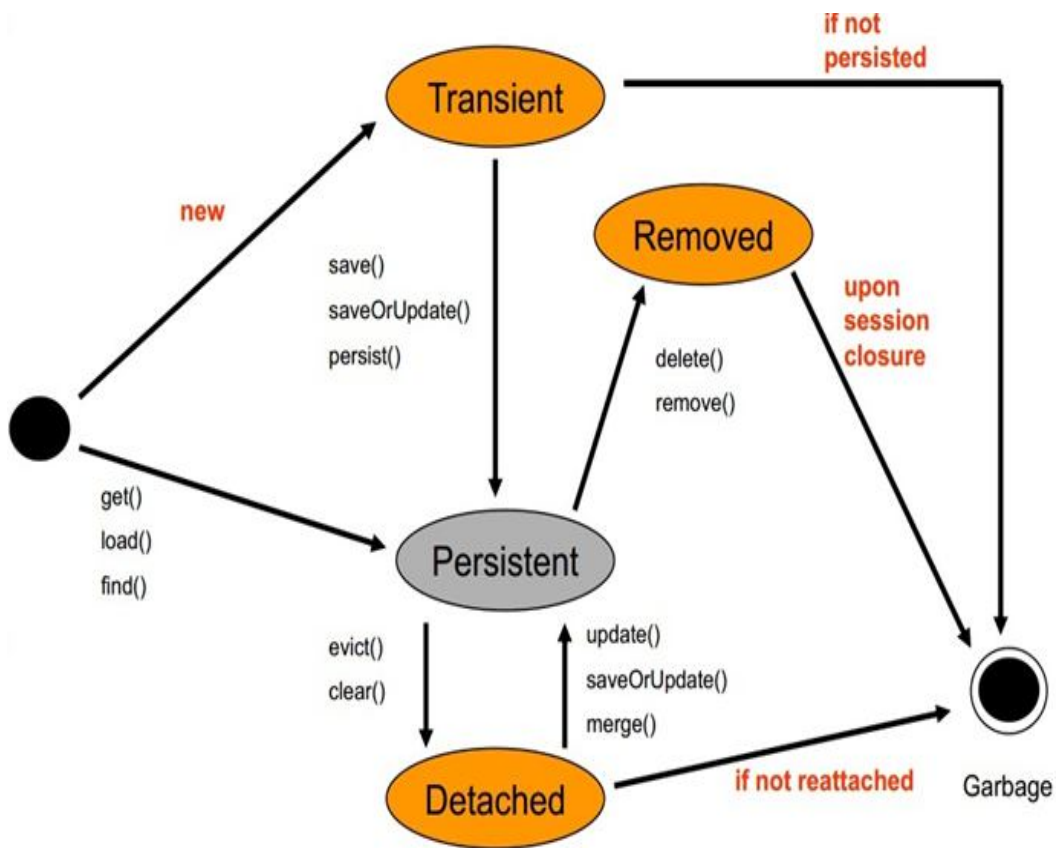
Hibernate supports the following three object states.

Transient: In this state pojo class object doesn't represent record in the table and object doesn't contain identify value. It is just ordinary java class object of pojo class and it is not associated with a Hibernate Session.

Persistent: This state object represent table row and maintains synchronization with table row. This state object contains identify value. Programmer uses this state object to perform persistence operations on table rows. The environment in java app that maintains persistence state objects is called persistence content. Detached state, transient state objects reside outside the persistence content.

Detached : This state object contains identify value but doesn't represent the record because either that associated session is clear or closed. Detached state object doesn't perform synchronization with record detached state object has represent the record earlier

but currently not representing the record that is the reason to have identify value in detached state object .



20. What are the benefits of detached objects?

Detached objects can be passed across layers all the way up to the presentation layer without having to use any DTOs (Data Transfer Objects). You can later on re-attach the detached objects to another session.

21. How does Hibernate distinguish between transient (i.e. newly instantiated) and detached objects?

" Hibernate uses the versionproperty, if there is one. If not uses the identifier value. No identifier value means a new object. This does work only for Hibernate managed surrogate keys. Does not work for natural keys and assigned (i.e. not managed by Hibernate) surrogate keys. Write your own strategy with Interceptor.isUnsaved().

22. What is the difference between session.update() and session.merge()?

The **session.update()** update the record only when given object related row is available in the table otherwise update() fails in record updation(if record is not available).

the **session.merge()** tries to update the record if the given pojo object related record is available in the database table and if it is not available this method inserts the record by using given pojo class object data. The given instance does not become associated with the session, it remains detached.

23. What is the difference between session.saveOrUpdate() and session.merge()?

session.merge(): if there is a persistent instance with the same identifier currently associated with the session, copy the state of the given object onto the persistent instance

if there is no persistent instance currently associated with the session, try to load it from the database, or create a new persistent instance the persistent instance is returned

session.saveOrUpdate(): The session save or update() takes given pojo class object identify field member variable value as criteria value to check wheather value based record is already available in the table or not if available this method perform update operation otherwise this method perform insert operation.

Note: if you are trying to update objects by using merge() then the given instance does not become associated with the session, it remains detached . if you are trying to update objects by using saveOrUpdate() then the given instance become associated with the session

24. Difference between session.load() and session.get()?

session.load() performs lazy loading to select the record from the database. session.load() will always return a "proxy" (Hibernate term) without hitting the database. In Hibernate, proxy

is an object with the given identifier value, its properties are not initialized yet, it just look like a temporary fake object. If no row found , it will throws an `ObjectNotFoundException`.

`session.get()` always hit the database and return the real object, an object that represent the database row, not proxy. If no row found , it return null.

25. What is lazy loading Vs Eager Loading ?

are two types of Loading in application. Eager loading and Lazy loading. In eager loading we will fetch all the values from the Persistent storage and cache it. It will make serious performance issues. There we use lazy loading to avoid that scenario.

Sometimes we don't need to load the child table values, In that case we have to us `lazy = true` in .hbm file. so hibernate will fetch only parent table values. Internally it will load the wrapper class but it does not cache all the values till we perform operation.

26. Explain about the id field?

This id field corresponds to the surrogate key which is generated by the database. These fields are handled by the id field. Name attribute is used to specify the names of the field and it should correspond to the method name of `getId`. This also should correspond to long type and the values should be stored I the database in the long column.

27. Define HibernateTemplate?

org.springframework.orm.hibernate.HibernateTemplate is a helper class which provides different methods for querying/retrieving data from the database. It also converts checked HibernateExceptions into unchecked DataAccessExceptions.

28. What are the benefits does HibernateTemplate provide?

The benefits of HibernateTemplate are :

HibernateTemplate, a Spring Template class simplifies interactions with Hibernate Session.

Common functions are simplified to single method calls.

Sessions are automatically closed.

Exceptions are automatically caught and converted to runtime exceptions.

29 . If you want to see the Hibernate generated SQL statements on console, what should we do?

In Hibernate configuration file set as follows:

```
<property name="show_sql">true</property>
```

.

30. what are the advantages of using HQL ?

1. HQL queries are database independent. Since these queries are based on persistence logic, the logic is database independent.
2. HQL queries are "object level queries" so they return Hibernate POJO class objects as results.
3. HQL queries and keywords are very much similar to SQL queries of Oracle.
4. HQL queries support operators, expressions, conditions, joins, sub queries, aggregate functions and etc.
5. HQL allows representing SQL queries in the form of objects. Hibernate Query Language uses classes and properties instead of tables and columns.

31. what are the disadvantages of using HQL ?

1. HQL queries can't perform "DDL" operations.
2. HQL queries can't be used in PL/SQL program.
3. HQL queries can't be used to insert a single record into a table.
4. An HQL query gives negligible performance degradation because of conversions when compared to SQL.

32. what is the difference between getCurrentSession and openSession in Hibernate ?

A Session is opened when `getCurrentSession()` is called for the first time and closed when the transaction ends. It is also flushed automatically before the transaction commits. You can call `getCurrentSession()` as often and anywhere you want as long as the transaction runs.

If you decide to use manage the Session yourself then go for `sf.openSession()` , you have to `flush()` and `close()` it. It does not flush and `close()` automatically.

33.what is CriteriaAPI in hibernate ?

- 1 . This API is given to develop hibernate persistence logic pure in java statements and without using SQL and HQL queries.
2. Criteria API means working with classes and interfaces of `org.hibernate.criterion` package classes and interfaces to develop persistence logic.
- 3 . We can use criteria-api only select operations based persistence logic development and criteria-api is not design to select specific column values of a table it will select all the column values.
- 4 . While developing the criteria –api based persistence logic pojo classes, member variable names will be utilized.Criteria-api based persistence logic is “database independent”persistence logic.

34.What is Hibernate proxy?

In hibernate proxies are placeholder that are generated at runtime. Whenever hibernate returns an instance of any entity class, it checks weather it returns a proxy object to avoid the database hit.

In hibernate proxies are triggers the loading of real object when it just accessed at very first time.

When we use `session.load(.....)` to get the resultset of any entity with the help of identifier and by calling the property other than identifier, the proxy is initialized.

35. Define cascade and inverse Hibernate ?

Inverse : An inverse keyword has the boolean value "true/false". Default value of this keyword is 'false' it means the parent class is responsible for saving/updating the child and its relationship. And when the keyword inverse is set to the 'true' it means an associated subclass is responsible for saving/updating itself.

Cascade : The "Cascade" keyword is often appear on the collection mapping to manage the state of the collection automatically. Cascade has the various options "all ", " save | update | delete | merge " that may be used with either of the single option or may be used with the multiple options. Cascade decides the same operation done on the parent object is done on the associated object at the same time.

36. how to create primary key using hibernate?

The id element describes the primary key for the persistent class and the generator class automatically generate the primary key for id

```
<id name="id" column="id" type="long">  
<generator class="increment"/>
```

37. What is first level cache in hibernate?

First-level cache always Associates with the Session object. Hibernate uses this cache by default. Here, it processes one transaction after another one, means wont process one transaction many times. Mainly it reduces the number of SQL queries it needs to generate within a given transaction. That is instead of updating

after every modification done in the transaction, it updates the transaction only at the end of the transaction.

38. What is Second-level cache in hibernate?

Second-level cache always associates with the Session Factory object. While running the transactions, in between it loads the objects at the Session Factory level, so that those objects will be available to the entire application, don't bound to single user. Since the objects are already loaded in the cache, whenever an object is returned by the query, at that time no need to go for a database transaction. In this way the second level cache works. Here we can use query level cache also. Later we will discuss about it.