1. What is ORM?
2. what are the main features of ORM?
3. Why Object Relational Mapping?
4. what is hibernate?­­
5. [What is database persistence? How well it is implemented in hibernate](http://www.geekinterview.com/question_details/44834)?
6. Hibernate is ORM tool. ORM stands for Object relational mapping. Object is one instance of Class. Relation is instance of Table. Both object and Relation are ment for Holding data. To convert one format to another format we required mapping. Mapping solves three problems

1. Mismatch of data types. 2. Mismatch of Reaction ships. 3. Storage types (classes and tables).

Any ORM frame work solves these three Mappings issesues. Mapping can be applied

In how many ways. 1. Hbmfiles(XML format) 2. Annotations.

We are using JPA Annotation based mapping. This has flexibility in migrating from one framework to another framework.

What is the Importance Of Wrapper And Primitive Types In Hibernate?

1. Why you are using ORM
2. Benefits of ORM
3. Reduces the development code. No need to register driver, no need to get connection,

Statement , result set. No need to process result set, no need to close resources. No need to implement exception handling. By using hibernate we can have predefined methods like get, save, update we can reduce our work. It also provides HQL , joins to reduce the code with portability feature.

1. Maintainability

Less code improves maintainability.

1. Improves performance.
   1. Caching.

Caching is temp data that reduces network calls. There are three types of cachings

* + 1. First level caching.

First level caching is associated with session. It resides b/w your program and Database. It caches all session operations without interacting database directly. Once Tx has committed then all operations are performed in data base by using batch update by creating query for each operation.

It improves performance by using batch update that follows few network calls.

It reduces the number of query generation. Ex: Multiple update methods on same object have single update query.

Diff b/w flush and commit.

Flush saves the data into database but not commit the data. Tx.commit method save the data and commits the data into database.

What is clear, evict mehod. How can you clear the data from fistlevel caching(session.)

Clear and evict methods are ment for clearing the data from firstlevel(session). Clear method clear all objects.evict method remove specified object only.It is not configurable. Short lived cahing.

* + 1. Second level caching.

Second level caching is associated with SessionFactory. It caches transactional data and query generation. It uses secondary memory(hard disk).It long lived caching. It is configurable.

* + 1. Query level caching.
  1. Lazy/Eager loading.
  2. Optimization techniques.
  3. Batch update

1. Portability.
2. Hibernate entity Object States.
3. What are the dff b/w transient state and persistence state ?
   * 1. No record in the database is available for transient object.
     2. In normal cases Transient state don’t have ID.

In case of persistence object there is record and id.

1. What are the dff b/w detached state and persistence state ?
   * 1. Detached object may not in sync with database record.

In case of persistence object sink is guaranteed

1. What are methods available in session.
2. Diff b/w get and load

Load is lazy and raises Object not found exception when record not found in Database.

Get is eager and return null object once it finds no specified record in Database.

1. What is lazy loading ? What is Proxy design pattern how it works in hibernate?

Lazy loading is one of the fetching strategies that load data only when object is using.

Lazy loading is implemented in Hibernate by using Proxy Design pattern. As per proxy design patterns Hibernate creates subclasses for entity classes and overrides super class methods like getter methods. These methods contain logic to go the database and fetch the information.

When we call load method it gives proxy object (subclass of entity), instead of original entity object. When we call getter method that calls proxy object method that goes to database and fetches the information only once per object. If we call another method on same object that uses same data rather than going to database again.

1. Difference b/w Session Factory and Session.
   * 1. Session Factory is factory for Sessions. Session is factory for Transactions.
     2. Session Factory is Heavy wait , Session is light wait.
     3. Session Factory is thread safe. Session is not thread safe.
     4. Session Factory maintains Connection pooling (Datasource). Session maintains connection.
     5. Session Factory maintains Second level caching. Session maintains first level caching.
     6. Session Factory is One per Database, session is one per request(thread).
2. Can you explain the Hibernate Architecture in detail?
3. Explain about Hibernate SessionFactory ?
4. Explain about inheritance & polymorphism in hibernate?
5. Explain about Hibernate mappings ?
6. What is meant by component mapping? (one table multiple classes)
7. What are the problems you faced while using hibernate in your project?
8. Explain about Lazy Loading?
9. Explain differences between get() and load()?
10. What is difference between First level cache and Second level cache?
11. What is the difference Session and SessionFactory?
12. How can we connect more than one database in hibernate applications?

we have to provide hibernate-config file for each database and create SessionFactory for

each database. Then use specific session factory for storing data in specific database.

1. Difference between Dialect and Driver?

===============================================================

1. Is it possible to write hibernate application without using hibernate mapping file?
2. Is it possible to write hibernate application without using hibernate configuration file ?

By using properties and by using java programming.

1. Explain about Persistance class life cycle states in hibernate?
2. Explain about hibernate major components?
   * 1. What is disadvantage using hibernate?
     2. Hibernate is very usefull only when we are developing read-write centric applications.
     3. Comparatively hibernate is slower than jdbc.
3. [What is hibernate proxy?](http://www.geekinterview.com/question_details/50246)
4. What are the differences between Lazy Loading and Eager Loading? Which one would be suggested?
5. How to define a primary key for a table in hibernate?
6. I want to change my database from Oracle to MySQL in hibernate. Which configuration changes are required and which classes we have to use?
   * 1. Changes happens only in hibernate configuration file to provide new database details.
7. What are the files we have requires doing a hibernate   application?
8. [What is the difference between and merge and update](http://www.geekinterview.com/question_details/51897)?

Merge compares previous data and current data, if difference found then it generates update query. To compare data , merge method generate select queries.

Update method simply generates update query even then no modifications are present.

Update method causes NonuniqueObjectException if same entity object is available in the same session.

Merge method merges both objects if same entity object is available in the same session without casing NonuniqueObjectException.

1. What is nonuniqueObjectException in hibernate?

When two entity objects referring same record in database , with in the same session then

We face nonuniqueObjectException.

1. [What is the use of cascade in hibernate?](http://www.geekinterview.com/question_details/46108)

Cascade is one mechanism that propagates one operation from one table to related tables.

Ex: if we delete one record in student then delete operation works in qualification table to delete related record.

We can apply operations for cascade in different ways.

Ex: cascade=CascadeType.ALL @OneToOne(cascade={CascadeType.REMOVE,CascadeType.PERSIST})

cascade=CascadeType.MERGE || PERSIST || REFRESH || REMOVE

1. [What the version is of hibernate using in current project?](http://www.geekinterview.com/question_details/41866) 3.1
2. [What is the main advantage of using the hibernate Query Language than using the SQL](http://www.geekinterview.com/question_details/36717)?
3. What are the advantages of Hibernate over JDBC?
4. [How to create composite primary key using hibernate?](http://www.geekinterview.com/question_details/36806)
5. How to store Images using Hibernate?

1. CREATE TABLE `slokam`.`avatar` ( `AVATAR\_ID` INT(10) UNSIGNED NOT NULL AUTO\_INCREMENT,

`IMAGE` BLOB NOT NULL,

PRIMARY KEY (`AVATAR\_ID`) USING BTREE) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8;

2. pojo: private byte[] image;

3. File file = new File("C:\\mavan-hibernate-image-mysql.gif");

byte[] bFile = new byte[(int) file.length()];

try { FileInputStreamfileInputStream = new FileInputStream(file);

//convert file into array of bytes

fileInputStream.read(bFile);

fileInputStream.close();

} catch (Exception e) {

e.printStackTrace(); }

4.session.save(pojo);

1. Could you please list out any common hibernate exceptions that you have faced in your project?
   * 1. Nonuniqueobjectexception
     2. SesionException.
     3. ObjectNotfound exception.
     4. MappingException
     5. IdentifierGenerationException
2. What are the Simple Hibernate Application Requirements?
3. Can you list out the steps to implement hibernate in any application?
4. What is versioning in hibernate and what is the advantage?
   * 1. Versioning in hibernate can be implemented in two ways

1.versionId

2. timestamp.

VersionId is better than timestampbcz for all updations comparison requires among versioning. So if it is time stamp it takes much time to process.

1. How to convert an Object from detached to persistant state?
2. **Can you give an example on Composite Primary Keys In Hibernate?**
3. Can you explain generator classes in hibernate?
4. How to pass Runtime values to the database using HQL?

Query qry = new Query (‘From Product where name=:name’)

Qry.setString(“name”,value);

1. **What are the differences between HQL and Criteria?**
   * 1. Hql is faster than Criteria in execution.
     2. When we are working with critical searching operations then criteria is

Very developer friendly .

1. What are the differences between the named queries and Native SQL?
   * 1. Native sql is database specific. i.e if we change the database , we have to change sql also. HQL query is generic for all databases.
     2. Hql first converts to native sql specific to specified database in configuration (dialect). But native sql directly executes in db.
2. How to implement pagination in hibernate?

Query.list()

qry.setFirstResult(30); qry.setMaxResults(10);

1. Can you explain how to write joins using Hibernate?
2. What are the jars required to configure hibernate Annotations?
   * 1. hibernate-commons-annotations-3.0.0.ga.jar
     2. hibernate-entitymanager-3.3.2.GA.jar
     3. persistence-api-1.0.jar
     4. other supporting jar files
3. How to integrate hibernate with spring?
4. How to configured an hibernate annotation configurations in spring xml file?
5. What are the advantages of Hibernate Template?
6. What are the differences between merge and update methods?
7. What differences are between save and persist methods?

Save method returns primarykey. Persist method return null.

1. Can you explain hibernate relations using annotations?
2. Explain about Connection pooling in hibernate?
3. How to get JNDI property configured in web logic into Hibernate applications?

<property name="connection.datasource">java:comp/env/jdbc/stdb</property>

1. How to restrict certain properties of pojo to not to

stored in to the database using hibernate Annototions?

Using @transient we can avoid column in the table for specified property in entity class.

Using updatable = false we can avoid data to be update in the table.

Using insertable = false we can avoid data to be inserted in the table.

1. How to call Stored Procedures from Hibernate?

Stored procedures can be called in two ways in hibernate:

* + 1. Using native sql and anther is using namedqueries.

DELIMITER $$

CREATE PROCEDURE `GetStocks`(int\_stockcode VARCHAR(20))

BEGIN

SELECT \* FROM stock WHERE stock\_code = int\_stockcode;

END $$

DELIMITER ;

CALL GetStocks('7277');

Query query = session.createSQLQuery(

"CALL GetStocks(:stockCode)")

.addEntity(Stock.class)

.setParameter("stockCode", "7277");

List result = query.list();

for(inti=0; i<result.size(); i++){

Stock stock = (Stock)result.get(i);

System.out.println(stock.getStockCode());

}

=========================================

@NamedNativeQueries({

@NamedNativeQuery(

name = "callStockStoreProcedure",

query = "CALL GetStocks(:stockCode)",

resultClass = Stock.class

)

})

Query query = session.getNamedQuery("callStockStoreProcedure")

.setParameter("stockCode", "7277");

List result = query.list();

for(inti=0; i<result.size(); i++){

Stock stock = (Stock)result.get(i);

System.out.println(stock.getStockCode());

}

1)First difference between save and persist is there return type. Similar to save method persist also INSERT records into database but **return type of persist is void** while return type of save is [Serializable](http://javarevisited.blogspot.sg/2012/01/serializable-externalizable-in-java.html) object.

2) persist() method doesn't guarantee that the identifier value will be assigned to the persistent instance immediately, the assignment might happen at flush time.

3) persist() method guarantees that it will not execute an INSERT statement if it is called outside of [transaction boundaries](http://javarevisited.blogspot.sg/2011/11/database-transaction-tutorial-example.html). save() method does not guarantee the same, it returns an identifier, and if an INSERT has to be executed to get the identifier (e.g. "identity" generator), this INSERT happens immediately, no matter if you are inside or outside of a transaction.

4) Fourth difference between save and persist method in Hibernate is related to previous difference on save vs persist. Because of its above behavior of persist method outside transaction boundary, its useful in long-running conversations with an extended Session context. On the other hand save method is not good in a long-running conversation with an extendedSession context.

@column (name=”columnName”) is not required if property name

And column are same.

Even both property and column names are same better providing

@column (name=”columnName”) because To know column name

It uses reflection api, using reflection is always degrades

the performance.