***What is dialect in hibernate?***

Hibernate.dialect property tells Hibernate to generate the appropriate SQL statements for the chosen database. Dialect is the SQL dialect that your database uses. Hibernate.dialect property makes Hibernate to generate the appropriate SQL statements for the chosen database.

***What is persistence class in hibernate?***

Java **classes** whose objects or instances will be stored in database tables are called **persistent classes in Hibernate**. **Hibernate** works best if these **classes** follow some simple rules, also known as the Plain Old Java Object (POJO) programming model.

***States in Hibernate: Transient, Persistence, Detached in hibernate?***

A new instance of a a persistent class which is not associated with a Session, has no representation in the database and no identifier value is considered ***transient*** by Hibernate:

Person person = new Person();

person.setName("Foobar");

// person is in a transient state

A ***persistent*** instance has a representation in the database, an identifier value and is associated with a Session. You can make a transient instance ***persistent*** by associating it with a Session:

Long id = (Long) session.save(person);

// person is now in a persistent state

Now, if we close the Hibernate Session, the persistent instance will become a ***detached*** instance: it isn't attached to a Session anymore (but can still be modified and reattached to a new Sessionlater though).

The database and no identifier value is considered **transient** by Hibernate:

***What is Session & use in Hibernate?***

***Hibernate ORM?***

***String, String Buffer, String Builder?***

***Different between session.get() and session.load()***

https://www.mkyong.com/hibernate/different-between-session-get-and-session-load/

***GET and POST, PUT in REST?***

***How to create a Service in REST?***

http://www.vogella.com/tutorials/REST/article.html#restjersey

***Checked and Unchecked Exception?***

***Collections MAP?***

***Difference between for loop and iterator?***

You might need to use iterators if you need to modify collection in your loop. First approach will throw exception.

for (String i : list) {

System.out.println(i);

list.remove(i); // throws exception

}

Iterator it=list.iterator();

while (it.hasNext()){

System.out.println(it.next());

it.remove(); // valid here

}

<http://stackoverflow.com/questions/18508786/for-each-vs-iterator-which-will-be-the-better-option>

***Java Concurrent Modification Exception?***

java.util.ConcurrentModificationException is a very common exception when working with java collection classes. Java Collection classes are fail-fast, which means if the Collection will be changed while some thread is traversing over it using iterator, the *iterator.next()* will throw **ConcurrentModificationException**. Concurrent modification exception can come in case of multithreaded as well as single threaded java programming environment.

http://www.journaldev.com/378/java-util-concurrentmodificationexception

***Interview Questions***

<http://www.journaldev.com/3633/hibernate-interview-questions-and-answers>