

1 Explain Hibernate Architecture with diagram.

=> Hibernate is java based ORM tool that can provides framework for mapping application domain objects to the database.

There are Four Layer in Hibernate Architecture.

ci) Java Application Layer

cii) Hibernate Framework Layer

ciii) Backhand Api Layer

civ) Database Layer.

⇒ Hibernate uses various Java API like JDBC, JTA and JNDI.

This are the main element in Hibernate Framework.

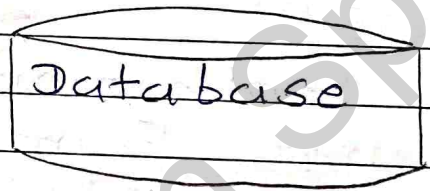
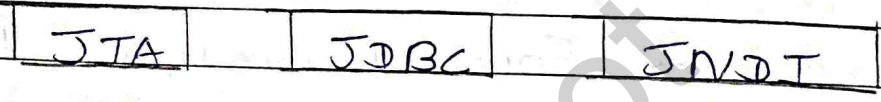
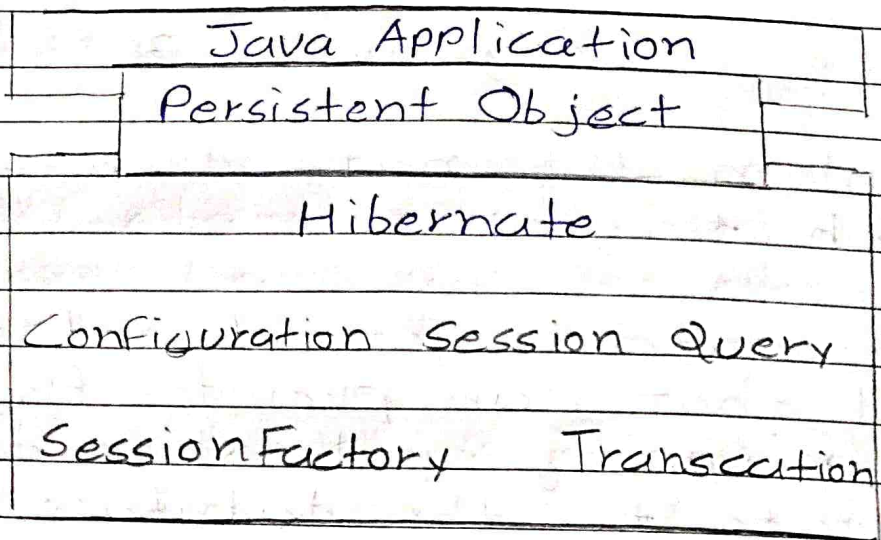
ca) Configuration

cb) Session Factory

cc) Session

cd) Transaction

ce) Query



### a) Configuration :

hibernate.cfg.xml is the configuration file for the Hibernate.

This file contains database connection details, dialect and other details.

### b) Session Factory :

Session Factory is a thread -

safe which contain second level cache of data.

org.hibernate.SessionFactory interface is used in Hibernate application.

(c) Session:

Session is a runtime interface between a Java application and Hibernate.

Session provides methods to perform database operation.

(d) Transaction:

Transaction represents a unit of work with the database.

Hibernate Transaction follows the ACID Properties.

(e) Query:

Hibernate Query Language is used in Hibernate application to perform the database operation.

2. What is HQL explain with example.

⇒ HQL stands for Hibernate Query Language which is similar to the SQL.

HQL is a powerful and flexible query language that is designed to work with Hibernate.

HQL works with Java Objects rather than database table.

HQL does not use SQL keyword like 'SELECT', 'FROM'.

HQL supports parameter binding which allows to create dynamic query.

HQL supports aggregate function like 'SUM', 'AVG', 'MAX' or 'MIN' etc.

HQL allows to use of subquery in Hibernate.

- HQL Query for select all:

```
Query query = session.create
    Query("from Table
        Name");
List list = query.list();
```

- HQL Query for Update:

```
Query query = session.create
    Object query("update
        Table set name =
        name
        :n where id = :1");
```

```
Query
    Object.setParameter("n", "k");
Query
    Object.setParameter("i", 111);
```

```
int Variable = query
    name Object.execute
        Update();
```

HQL Provides different method to perform query.

ca) executeUpdate(): Used to update the data in database

(B) List list() : Returns the result in form of list.

(C) setParameter(int position, Object value) : set the value at specific position.

(d) setMaxResult(int value) : set the Maximum result limit.

(e) setMinResult(int value) : set the minimum result limit.

3 Write Difference between HQL and SQL.

	HQL	SQL
1	Stands For Hibernate Query Language	Stands For Structured Query Language
2	HQL used Hibernate's Object-oriented database.	SQL used relational database.
3	HQL Uses a Dynamic schema.	SQL Uses a Fixed Schema.

- |   |  |   |
|---|--|---|
| 4 | HQL Query are made with object and properties. | SQL Query are made with command like 'select', 'From' or 'Where'. |
| 5 | HQL commands are write in lowercase.           | SQL commands are write in Uppercase.                              |
| 6 | HQL is slower than SQL.                        | SQL is faster than HQL.   |

4 Develop Program to get all students data from databases using Hibernate. Write necessary xml file.

⇒ Java File :

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import java.util.List;

class Main
{
    public static void main(String
        args[])
```

}

```
Session Factory factory =  
new Configuration().  
configure("hibernate.cfg.  
xml").addAnnotatedClass  
(Student.class).build  
SessionFactory();
```

```
Session session = factory.  
getCurrentSession();
```

try

}

```
session.beginTransaction();
```

```
List<Student> stu =  
session.createQuery("From  
Student").getResultList();
```

```
for (Student stu : students)
```

}

```
System.out.println(stu);
```

}

```
session.getTransaction().  
commit();
```

}

Finally }

```
factory.close();
```

}

}

}



⇒ hibernate.cfg.xml File :

```
<!DOCTYPE hibernate-configuration  
PUBLIC "-//Hibernate/Hibernate  
Configuration DTD 3.0//EN"  
"http://hibernate.sourceforge.net/  
hibernate-configuration-3.0.dtd">
```

```
<hibernate-configuration>  
  <session-factory>
```

```
    <property name="hibernate.  
      connection.driver.class">  
      com.mysql.cj.jdbc.Driver  
    </property>
```

```
    <property name="hibernate.  
      connection.url"> jdbc:mysql:  
      //your database URI </property>
```

```
    <property name="hibernate.  
      connection.username">  
      KRG </property>
```

```
    <property name="hibernate.  
      connection.password">  
      KRG </property>
```

```
<property name = "hibernate.  
c3p0.min_size" > 5 </property>
```

```
<property name = "hibernate.  
c3p0.max_size" > 20 </property>
```

```
<property name = "hibernate.  
c3p0.timeout" > 300 </property>
```

```
<property name = "hibernate.  
c3p0.max_statements" >  
50 </property>
```

```
<property name = "hibernate.  
c3p0.idle_test_period" > 3000  
</property>
```

```
<property name = "hibernate.  
dialect" > org.hibernate.  
dialect.MySQLDialect  
</property>
```

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

```
<property name = "hibernate.  
hbm2ddl.auto" > update  
</property>
```

```
<mapping class = "com.your  
package.Student" />
```

```
</Session-Factory>
```

```
</hibernate-configuration>
```

5 What is Hibernate? List the advantages of hibernate over JDBC.

=> Hibernate is an open-source, object-relation mapping framework for java.

It simplifies database programming in java by handling the mapping between Java classes and databases tables.

Hibernate developers can work with Java objects rather than SQL query.

Hibernate can provide object-oriented API to interact with the database.

In Hibernate mapping is defined through XML Files or Annotation in java code.

Hibernate allows developers to write database independent code.

Hibernate can generate database table based on the mapping information provided.

Hibernate Provide Hibernate Query Language which is more powerfull than SQL.

Hibernate provides caching mechanisms at both the session and query level.

Hibernate supports connection pooling which helps to manage and reuse database connection.

→ Advantages of Hibernate:

1 Abstraction of Database Details:

Hides the details of SQL query and do not need to write SQL query for basic CRUD operation which is not provide in JDBC.

## 2 Object-Relational Mapping:

Hibernate Provides a mechanism for mapping Java objects to database table.

## 3 Automatic Table Creation:

Hibernate can automatically generate database table based on Java Object Model.

## 4 Query Language:

Hibernate uses Hibernate Query Language which is similar to SQL but operates on Java Objects.

## 5 Automatic Connection Management

Hibernate can manages database connection automatically and provides connection pooling.

## 6 Cache Management:

Hibernate Provides caching mechanisms both at the session and query level.