What is a framework?

A framework is reusable semi finished application that can be customized to develop a specific application.

What are the different logics available in Enterprise Application?

Presentation Logic= Logic used to present the output/input.
Application/Controlling Logic= Logic used t o control the flow of application.
Business Logic=Programmatical implementation of business rules is nothing but business logic.
Data Access Logic=Logic used to contact the Database.

What is persistence in a java based enterprise application?

The process of storing enterprise data in to relational database is Iknown as persistence

What are the limitations of the traditional approach?

i) Application portability to the Database is lost (Vendor lock: diff SQL statement for the db's)

ii)Mismatches between Object oriented data representation and reational data representation are not properly addressed .

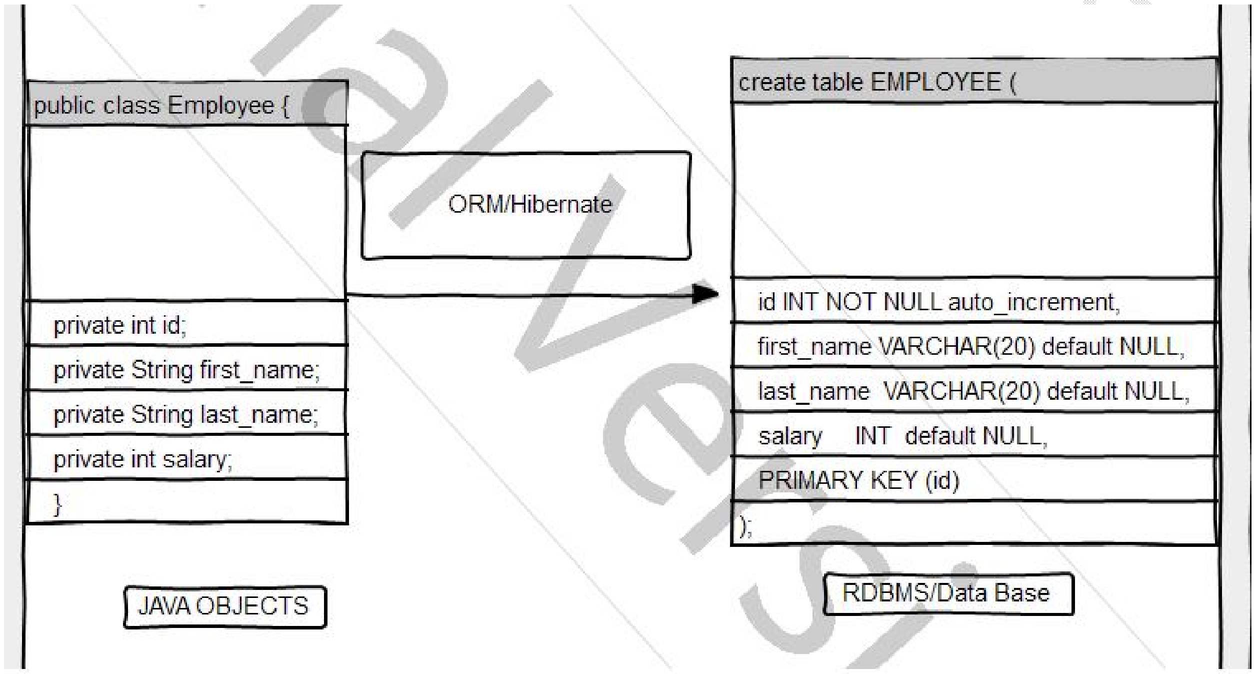
iii) Requires the extensive knowledge of DB
iv) Manual operations on Resultset
v) For every problem while communicating with the database (using JDBC), it throws same exception(java.sql.SQLException). As SQLException is checked exception, so we must write code in try-catch block or throws has to be specified.
vi)Need to implement caching manually .
vii)In the Enterprise applications, the data flow with in an application from class to class will be in the
form of objects, but while storing data finally in a database using JDBC then that object will be converted into text. **Because JDBC doesn't transfer objects directly**.

what is an alternative for traditional approach?

ORM (Object Relational mapping)
It is technique of mapping objected oriented data to that of relational data
Through ORM technique persistence services (database) are provided to business layer in pure object oriented manner by overcoming all limitations of the traditional approach.

What is ORM?

object/relational mapping is the automated (and transparent) persistence
of objects in a Java application to the tables in a relational database, using
metadata that describes the mapping between the objects and the database.
ORM, in essence, works by (reversibly) transforming data from one representation
to another.
Orm is also called as object role modeling/object reatonal mapping.



Java ORM Frameworks:

There are several persistent frameworks and ORM options in Java. A persistent framework is an ORM service that stores and retrieves objects into a relational database.

1. Enterprise JavaBeans Entity Beans
2. Java Data Objects
3. TopLink
4. Spring DAO
5. Hibernate
6. And many more

What is Hibernate?

Hibernate is an ORM implementation
Hibernate is an Open source
Hibernate is a framework
Hibernate invented by Gavin King in 2001. He also invented JBoss server and JPA.
Hibernate can run with or without server, I mean it will suitable for all types of applications (desktop or web applications).

Hibernate Advantages:

Hibernate persists java objects into database (Instead of primitives)
It provides Database services in Database vendor independent Manner, so that java applications become portable across the multiple databases

Hibernate generates efficient queries for java application to communicate with Database i
It provides fine-grained exception handling mechanism. In hibernate we only have Un-checked exceptions, so no need to write try, catch, or no need to write throws (In hibernate we have the translator which converts checked to Un-checked)

Hibernate supports Inheritance, Associations, Collections
Hibernate supports a special query language(HQL) which is Database vendor independent
9 Hibernate has

Supports over 30 dialects
Hibernate provides caching mechanism for efficient data retrieval I - - > Lazy loading concept is also included in hibernate so you can easily load objects on start up time
Hibernate supports annotations, apart from XML.

Q.)What are the disadvantages of hibernate?

Since hibernate generates lots of SQL statements at runtime so it is slower than pure JDBC
Hibernate is not much flexible in case of composite mapping. This is not disadvantage since understanding of conlposite mapping is complex
Hibernate does not support some type of queries which are supported by JDBC
Boilerplate code issue, actually we need to write same code in several files in the same application,but spring eliminated this.

What are the Simple Hibernate Application Requirements?

1 Entity class
2. Mapping file(Required if you are not using annotations)
3. Configuration file
4. DAO class (Where we write our logic to work with database

What are the Steps to develop hibernate applications? -

Step 1: Develop persistent/domain/entity class for each table of the relational model
Step 2: For each entity develop a mapping file
Step 3: Develop the configuration file
Step 4: Add hibernate framework jar files i n the classpath
Step 5: Make use of hibernate API and perform persistent operations

How to Make use of hibernate **API** to perform persistent'operations?

STEP1:
Create Configuration object
Configuration configuration = new Configuration();
STEP2:
Read configuration file along with mapping files using configure() rnethod of Configuration Object
configuration.configure();
STEP3:
Build a SessionFactory from Configuration
SessionFactory factory = **configuration.bhildSessionFactory();**
ST P4:
Get Session from SessionFactory object
Session session = factory.openSession();
STEP 5:
BEGIN Transacton
Transaction transaction=session.beginTransaction();
transaction.begin();

STEP6:
Perform persistence operations
Save/delete/read/update
STEP7: commit transaction and close session.

What is hibernate configuration file?

It is an XML file in which database connection details (username, password, url, driver class name) and ,
Hibernate Properties(dialect, show-sql, second-level-cache ... etc) and Mapping file name(s) are t specified to the hibernate
Hibernate uses this file to establish connection to the particular database server .
Standard for this file is <hibernate.cfg.xml>
We must create one configuration file for each database we are going to use, suppose if we want to connect wi th 2 databases, like Oracle, MySql, then we must create 2 configuration files.
No. of databases **we** are using = That many number of configuration files I
We can write this configuration in 2 ways ...
o XML file
o Properties file(old style)

We don't have annotations to write configuration details. Actually in hibernate 1.x, 2.x we defined this configuration by using .properties file, **but from 3.x XML came into picture.**
XmL files are always recommended to use

Example Of Configuration xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
        <session-factory>

        <property name="hibernate.connection.driver_class">Driver name</property>
        <property name="hibernate.connection.url">jdbc url with database name</property>
        <property name="hibernate.connection.user">data base user name</property>
        <property name="hibernate.connection.password">password</property>

//hibernate support 30 dialect
<property name="dialect">databse specific Dialect</property>

<property name="show_sql">true</property>
<property name="hbm2ddl.auto">update/create</property>

//if you are using hbm file then use below tag
<mapping resource="com\jsp\orm\EmployeeDto.hbm.xml"/>

//if your using Hibernate Annotation use below tag
<mapping class="com.jsp.orm.Employee"/>
        </session-factory>
        </hibernate-configuration>.
```

Dialect means **"the variant of a language"**

There may a chances for some point hibernate has to use database specific SQL. Hibernate uses "dialect" configuration to know which database you are using so that it can switch to the database specific SQL generator code wherever/whenever necessary.

Dialect is a communicator to JDBC from Hibernate. So then Hibernate should have one Dialect to communicate with different Database vendor

RDBMS	Dialect
DB2	org.hibernate.dialect.DB2Dialect
DB2 AS/400	org.hibernate.dialect.DB2400Dialect
DB2 OS390	org.hibernate.dialect.DB2390Dialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect
MySQL	org.hibernate.dialect.MySQLDialect
MySQL with InnoDB	org.hibernate.dialect.MySQLInnoDBDialect
MySQL with MyISAM	org.hibernate.dialect.MySQLMyISAMDialect
Oracle (any version)	org.hibernate.dialect.OracleDialect
Oracle 9i/10g	org.hibernate.dialect.Oracle9Dialect
Sybase	org.hibernate.dialect.SybaseDialect

Sybase Anywhere	org.hibernate.dialect.SybaseAnywhereDialect
Microsoft SQL Server	org.hibernate.dialect.SQLServerDialect
SAP DB	org.hibernate.dialect.SAPDBDialect
Informix	org.hibernate.dialect.InformixDialect
HypersonicSQL	org.hibernate.dialect.HSQLDialect
Ingres	org.hibernate.dialect.IngresDialect
Progress	org.hibernate.dialect.ProgressDialect
Mckoi SQL	org.hibernate.dialect.MckoiDialect
Interbase	org.hibernate.dialect.InterbaseDialect
Pointbase	org.hibernate.dialect.PointbaseDialect
FrontBase	org.hibernate.dialect.FrontbaseDialect
Firebird	org.hibernate.dialect.FirebirdDialect

What is hibernate mapping file?

In this file hibernate application developer specify the mapping from entity class name to table name and entity properties names to table column names. i.e. mapping object oriented data to relational data .
Standard name for this file is **<domain-object-name.hbm.xml>**
In general, for each domain object we create one mapping file.
Number of Entity classes = that many number of mapping xmls.
Mapping can be done using annotations also. If we use annotations for mapping then we no need to write mapping file.
From hibernate 3.x version on wards it provides support for annotation, So mapping can be done in two ways

- o XML
- o Annotations

example **Of** Mapping xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
   "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
   "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>

<class name="FULLY QUALIFIED CLASS NAME" table="DB table name">
<id name="JAVA FILED NAME" column="DB COLUMN NAME">
<generator class="increment"></generator>
</id>

<property name=" JAVA FILED NAME " column=" DB COLUMN NAME "  />
<property name=" JAVA FILED NAME " column=" DB COLUMN NAME "/>

</class>
</hibernate-mapping>
```