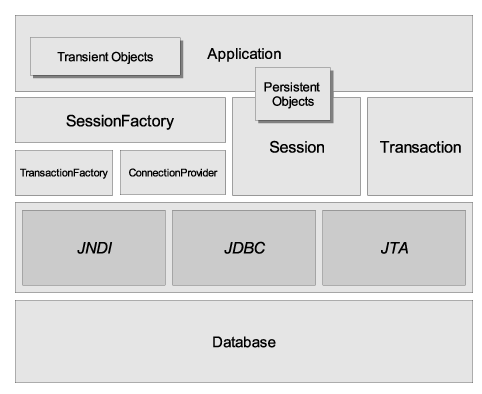
**Hibernate**

1. Hibernate is a ORM (Object Relational Mapping) tool.
2. Hibernate Used in the persistence Layer.
3. Hibernate Framework is the implementation of JPA standard.
4. Hibernate Used for a DB connection.
5. Hibernate handles all the Database check exception internally and convert it into unchecked exception.
6. Hibernate can generate DDL, DML queries internally and also executes the queries.
7. Hibernate Support caching techniques which help to make DB interaction faster. Hibernate internally support Primary Cache.
8. You can integrate Secondary cache inside Hibernate.
9. Hibernate supports collection mapping.

**Hibernate Architecture**



**SessionFactory :**

1. SessionFactory produces an Object of Sessions.
2. Once sessionFactory Object is linked with a one database at a time, If you are using multiple database in the application then you needs to create multiple SessionFacotry

**Session:**

1. Session Objects are consider as a Connection Object.
2. One session multiple transactions.
3. Session Objects are used to execute the queries and get result back into a java application.

**Transaction**

1. One transaction can have a multiple queries.
2. You can group a query into a single transaction so that you can either commit the transaction or rollback the truncation.

**Download Hibernate Jar File (5.5.7)**

[**https://hibernate.org/orm/releases/5.5/**](https://hibernate.org/orm/releases/5.5/)

**Steps To Add Jar file into Core java application**

1. Right click on Core Java Application
2. Go To “Build Path” -> Select “Configure Build Path…”
3. Select “Libraries” tab.
4. Select “Classpath” option
5. Click on “Add External Jars..” button.
6. Search for hibernate jar file
7. Select All the jar file from the path “<YourFolderPath>/hibernate-release-5.5.7.Final\lib\required”
8. Click On “Apply and Close” button

**Steps to Create hibernate configuration XML file**

1. Right Click on src
2. Go To “New” -> Select “Other…”
3. Search for “XML” option in Wizard
4. Select “XML File” option from the list
5. Provide any file name mostly prefer “hibernate.cfg.xml”
6. Click on “Next” -> “Finish” button

***Steps To create Hibernate Project***

1. *Create a Core Java Project*
2. Add Hibernate Jar files
3. *Create an Entity class*
4. *Create xml file to provide hibernate configurations (hibernate.cfg.xml)*
5. *Provide an Hibernate configuration inside xml file.*
6. *Use a main Method to test the hibernate operation using Hibernate APIs*

Entity Class Annotation

1. @Entity : To Define class as a Entity Class
2. @Id : To make column Primary Key
3. @Table : To set the Name of the table
4. @Column : To set name, length, Constraints to a column
5. @Temporal : for Date type of column.
6. @GeneratedValue : Is use to generate the primary key internally/automatically
7. @CreationTimstamp : to insert the System Date time internally by hibernate at the time of inserting data
8. @UpdationTimstamp : to insert the System Date time internally by hibernate at the time of inserting and Updating data
9. @Transient : To ignore the fields from the hibernate operations.

**Query Execution in Hibernate**

1. In the Hibernate you can execute you queries manually.
2. Hibernate has provided 2 approaches to write a query
   1. HQL
      1. Hibernate Query Language
      2. You have to write a query using Entity name and field name instead of table and column name.
   2. SQL
      1. Structure Query language
      2. You used a table name and column name in this queries.
      3. In Hibernate SQL queries are known as Native queries.
3. To Execute query in the Hibernate you can use **Query interface.**

**Collection Mapping in Hibernate (HAS-A)**

* + - 1. Can cerate a tables which has a relation in the form of PK and FK.
      2. To Achieve this you have to create the reference of one class into another class.
      3. This reference has to annotate with one of the following annotation.
         1. @OneToOne
         2. @ManyToOne
         3. @OneToMany
         4. @ManyToMany

1. To create a @OneToOne and @ManyToOne you have to user single object
2. To create a @OneToMany and @ManyToMay you have to use collection of object.



Fetch Type:

1. This property is use to manage the joins applied on multiple table by hibernate.
2. Fetch Type is of 2 types
   1. EAGER: This will pull the information of another table at the time of getting records form original table. This is the default fetch type for OneToOne and ManyToOne
   2. LAZY: This will not pull the information from the another table at time of getting the records from original table but if user is trying to get the info from the another table then separate query will be executed for this operation. This is the default fetch type for OneToMany and ManyToMany