**Draw Backs of JDBC:**

* In JDBC, if we open a database connection we need to write in try, and if any exceptions occurred catch block will takers about it, and finally used to close the connections.
* here as a programmer we must close the connection, or we may get a chance to get our of connections message…!
* Actually if we didn’t close the connection in the finally block, then jdbc doesn’t responsible to close that connection.
* In JDBC we need to write Sql commands in various places, after the program has created if the table structure is modified then the JDBC program doesn’t work, again we need to modify and compile and re-deploy required, which is tedious.
* JDBC used to generate database related error codes if an exception will occurs, but java programmers are unknown about this error codes right.
* 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 Hibernate:

Hibernate is the ORM tool given to transfer the data between a java (object) application and a database (Relational) in the form of the objects.

Hibernate is a non-invasive framework,  means it wont forces the programmers to extend/implement any class/interface, and in hibernate we have all POJO classes so its light weight.’ Hibernate can runs with in or with out server, i mean it will suitable for all types of java applications (stand alone or desktop or any servlets bla bla.)

Hibernate is purely for persistence (to store/retrieve data from Database).

Mapping and Configuration are very familiar keywords we used to here in the hibernate, every hibernate program must need these 2 xml files.

**Mapping:**

* Mapping file is the heart of hibernate application.
* Every ORM tool needs this mapping, mapping is the mechanism of placing an object properties into column’s of a table.
* Mapping can be given to an ORM tool either in the form of an XML or in the form of the annotations.
* The mapping file contains mapping from a pojo class name to a table name and pojo class variable names to table column names.
* While writing an hibernate application, we can construct one or more mapping files, mean a hibernate application can contain any number of  mapping files.

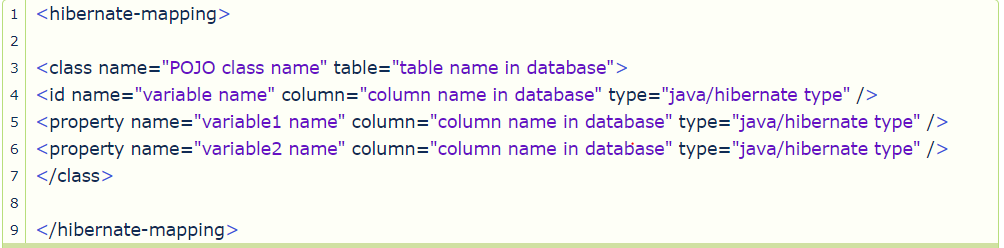
generally an object contains 3 properties like

* Identity (Object Name)
* State (Object values)
* Behavior (Object Methods)

But while storing an object into the database, we need to store only the values(State) right ? but how to avoid identity, behavior.. its not possible. In order to inform what value of an object has to be stored in what column of the table, will be taking care by the mapping,  actually mapping can be done using 2 ways,

* XML
* Annotations.

Actually annotations are introduced into java from JDK 1.5.



**Configuration:**

Configuration is the file loaded into an hibernate application when working with hibernate, this configuration file contains 3 types of information..

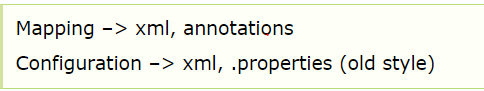
* Connection Properties
* Hibernate Properties
* Mapping file name(s)

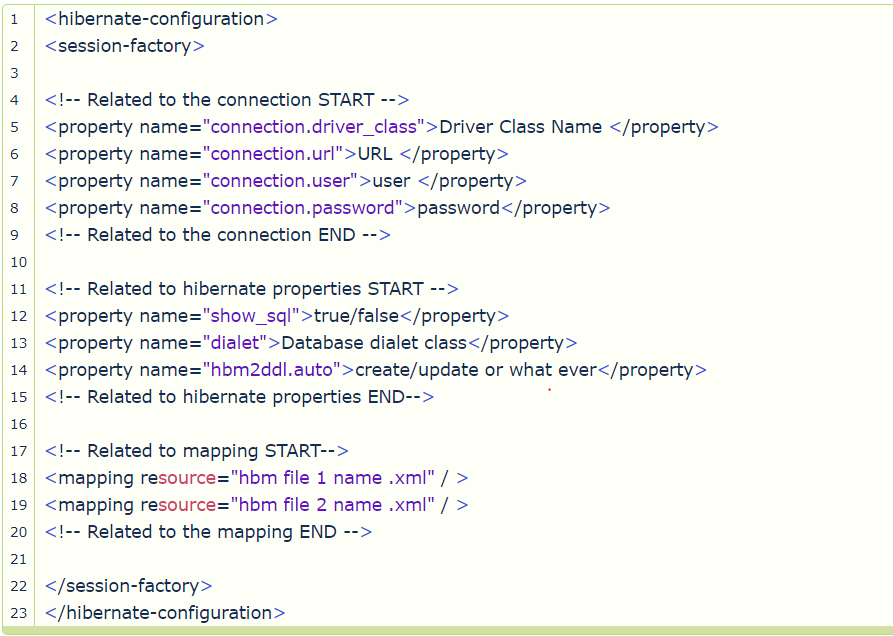
We must create one configuration file for each database we are going to use, suppose if we want to connect with 2 databases, like Oracle, MySql, then we must create 2 configuration files.

No. of databases we are using  = That many number of configuration files

We can write this configuration in 2 ways…

* xml
* By writing Properties file.  We don’t have annotations here, actually in hibernate 1, 2.x we defined this configuration file by writing .properties file, but from 3.x xml came into picture.





**Advantages of hibernates:**

* Hibernate supports Inheritance, Associations, Collections
* In hibernate if we save the derived class object,  then its base class object will also be stored into the database, it means hibernate supporting inheritance
* Hibernate supports relationships like One-To-Many,One-To-One, Many-To-Many-to-Many, Many-To-One
* This will also supports collections like List,Set,Map (Only new collections)
* In jdbc all exceptions are checked exceptions, so we must write code in try, catch and throws, but in hibernate we only have Un-checked exceptions, so no need to write try, catch, or no need to write throws.  Actually in hibernate we have the translator which converts checked to Un-checked
* Hibernate has capability to generate primary keys automatically while we are storing the records into database
* Hibernate has its own query language, i.e hibernate query language which is database independent
* So if we change the database, then also our application will works as HQL is database independent
* HQL contains database independent commands
* While we are inserting any record, if we don’t have any particular table in the database, JDBC will rises an error like “View not exist”, and throws exception, but in case of hibernate, if it not found any table in the database this will create the table for us
* Hibernate supports caching mechanism by this, the number of round trips between an application and the database will be reduced, by using this caching technique an application performance will be increased automatically.
* Hibernate supports annotations, apart from XML
* Hibernate provided Dialect classes, so we no need to write sql queries in hibernate, instead we use the methods provided by that API.
* Getting pagination in hibernate is quite simple.

**Disadvantages of hibernates:**

* I don’t think there are disadvantages in hibernate
* You know some thing.., Its saying hibernate is little slower than pure JDBC, actually the reason being hibernate used to generate many SQL statements in run time, but i guess this is not the disadvantage
* But there is one major disadvantage, which was boilerplate code issue, actually we need to write same code in several files in the same application, but spring eliminated this

Any hibernate application, for example consider even first hello world program must always contains 4 files totally.

* POJO class
* Mapping XML
* Configuration XML
* One java file to write our logic

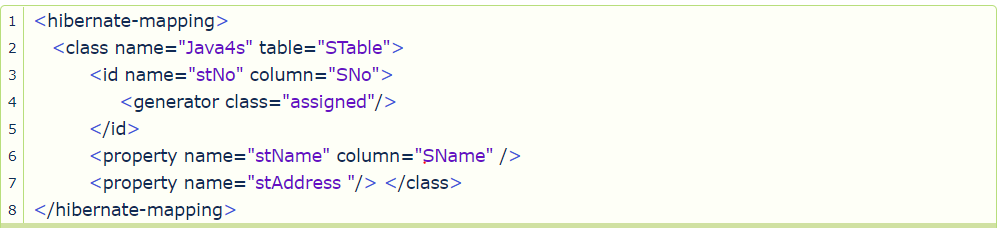
Actually these are the minimum requirement to run any hibernate application, and in fact we may require any number of POJO classes and any number of mapping xml files (**Number of POJO classes = that many number of mapping xmls**), and only one configuration xml and finally one java file to write our logic.

**POJO Class:**

* POJO is a simple java file, no need to extend any class or implement any interface.
* This POJO class contain private properties variables, and for each property a setter and a getter



## ****Mapping xml For POJO****



Yes., see in this above mapping xml, for stAddress property i have not written any column name i just been specified  **<property name=”stAddress “/>,**this means in the database the column name for stAddress property will also be stAddress, in these cases we can ignore the column attribute to write, and i will explain about this <generator /> element later.

## Configuration XML

Refer above

Usually configuration file name will be hibernate.cfg.xml

# How To Install Hibernate

Download hibernate.jar and use

For maven :

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.4.0.Final</version>

</dependency>