Hibernate

What is persistency?

To the store the data permanent storage area where it will be store permanently

Making the state of a data writing a state of the data in to a permanently storage unit is called persistency.

Why we should go for persistent data?

To avoid the application program to re-enter the same data in to the processing data we may wanted to store the data permanently and we want our program to access the data from permanent storage location to compute that output in such case we may have to keep track of that information or may be the outcome computed by your application may wanted to use at later point of time as well in such case also, To avoid re-computation of data we wanted to store the data permanently at some permanent storage the necessity of permanent and permanent storage are persistency highly required .

But initial generation of the system the permanent storage of information or the concept of persistency is not there.

(Before persistency came) Never a computer system has been added with capability of persistency storages.

Reading from the data from the permanent storage and writing the data to the permanent storage, there no concept is called storing a data within a computer system is there.

Every programming language started adding persistency support for working with storing the data and accessing the data for persistency storage device(like an Hard Disk or any other device).

How can we store the file in the permanently area?

By using the file system storage, so the DOS operating system is a FileSystemStorage, NTFS (New Technology File), FAT (File Allocation Table)

Till then OS don't know how to work with storing and retrieving data from an files

They used to add of the data by using "bunch card" and the output is being written on printer ,monitor are not there ,papers of paper written as the part of the output.

None of the programming language has support for working with storing the data within the file

New programing language has been introduce (Like Cobol) Cobol is very popular in the market because none of the programming

language has support for storing the data on to the file but it can .Cobol is being called as business language. Building business application COBOL is highly use because it can easily process the file .One can store the data and one can access the data from the file.

How to store the data within the file?

We can store the data within file in text of character. A text of file contain sequence of character of data any amount of data any character spanning across the any paragraph of the data can store

Why we store the data?

Whenever we are storing the data on a permanent storage, to use the same data for future point of time, that's why we store the data.

Human beings are only can use the data. The format of the data that we are trying store in the text file is **Human Understandable Format only**

E.g

I write a data in to the file

1Ravi25/10/201610100

Can you please WAP the will takes text file as input and will able to compute the total amount of sale?

Here we are unable to distinguish the information which is stored in the file. Because everything simple set of character initially people are worried about store the data permanently.

Every one use character data to store the data in text file

Storing the data in a file, is of two type

1) Human understandable format

2) Programmable format /Machine Understandable.

Human understandable format like MS-office, MS-excel

How do I need to store the data in the file system so that the data can access by the program and can compute the operation so that it is going to save lots of time

Human interdiction can be avoided reading repeatedly

the same data to perform recalculation on same data can be avoided and the computed data that have been computed by my program can we make permanent so kept permanent so that I can kept later point on time so that I can avoid recomputed the same data to build the outcome and save lots of time

How to make the data persistent so that not only human being but also program can understand that's why a research has been made in the field of Engineering in terms of persistence

the data and representing in a file and several file format has introduced

People analyze, that how can we optimize, how can we better store file based persistency that’s why Fix width field size format came into picture

How to store the data in Programmable untestable

First and fore most programing

1) Fix width field size format

1Ravi25/10/201610100

There is now way of distinguish different parts of the data because everything is being store in sequentially the data is linear or there is multiple parts of the data is there

Usually a character base representation is sequentially, but when it comes to business representation of data every character will not same. One character may be different from another, every charter or every piece of information will have an meaningful information and acquits

We need to have mechanism for distinguishing one part of the data from another part of data.

1Ravi25/10/201610100

The business data will be linear or will scalar (means dimension will be there i.e separator “,”)

What is linear?

It stores the data in sequentially, it doesn’t differentiate part of the data, and everything is same.

What is Scalar?

It stores the data in different, different part, each part has different meaning from one and another.

Here every field of data is being reserved with certain field size in Fix with field size format

It easily store the data, and process the data within the fix field size format

Every field has resolve size with which you are going to store the data it’s that the program can read that’s the human can also read data.

Drawbacks of Fix with field size format

To overcome these problem people are searching for alternate thing for storing the data so that, Human being and programs can read and can perform the calculation.

Advantages

Different places object can be easily distinguishable or can be identified easily so that human can read it or program can read it and can automated the business operation rather than human interdiction reading the data

As the data is nicely being formatted by reserving the spaces we can easily read the information as well.

Disadvantage

This format representation of the data are usually consume more amount memory because

These are memory consumable this consume more amount of the data when we are trying store less amount of data

If there will change in field size the whole data that has been stored till now has to be modified and as the program has been written to read the data based on the boundaries and condition of the field .if the boundary of the field are changing again my application logic has to modified to consider new boundary and field, So

there are lot of maintenance cost in maintaining and accessing the programing data is there so maintainability of the data is not easy

(when small scale business will grow to large scale business then we have to change the field size, then we will face above problem).

Comma Separated value (CSV) format /Delimited format

1, Ravi, 25/10/2016, 10,100

How can I store the data in csv format?

Each field of information what we have within our business has to separated by comma So we can distinguish different field using comma, To read one line after by another , tokenize by coma and every token that you extracted present one information representing that business.

Every piece of information is being differentiate with comma before 1st comma one information ,after 1 comma another information and so on ,that's why programs can distinguish the data by using comma separator.

Advantage

The amount of memory that is being consumed by the data will be less.(we are not assuming any boundary through which we are able to resolving problem with **fixed width field size format**)

This is more viable and flexible way of storing the data when compared with (FWFSF) by using comma as separator. The boundaries are derived by comma

Maintenance cost will be less

We can store the any amount of data for a field.

If later on business requirement will be increase there is no need to shifting the space and adding more spaces creating a root of the data is not required

(when small scale business will grow to large scale business then we can easily change the field size).

Drawbacks Comma Separated value (CSV) format /Delimited format

The data that has been representing is not semantic

Semantic means Information about the data that is not representing as the part of the data

The following are drawback

1. Semantic of the data is not represented.
2. Structure of the data is not there.
3. The data are not easily sharable.

Semantic of the data

- Semantic means representation of the data, meaning about the actual data .
- The meaning of the data that we had stored were not represented as the part of the data

Structure of the data is not there

- The information that we stored that containing comma .Comma is use as a delimiter
- The boundary of the data is not there, that where my actual data is starting from and where it is ending with
- We will easily get out of the boundary
- The structure of the data is not preserved easily because the actual information that is containing comma will also take as delimiter, the whole format will get corrupted.
- Where does the field start with and where does end with

How to read the data from csv file? With drawbacks

Read one line after another line which is called record

Read the data, record by record, every record will be delaminated by comma

The data that we are trying to read within the csv format should be at linearly sequentially from top to the bottom (line by line we have read to read the data)

Maintaining and modifying the data within the file is not easy (we have to write more amount of logic managing and adding the character to the existing record of information complex programing we need to do)

How to delete the data?

There are 1000s of record in csv file out of which 10^{th} record wanted to deleted

If wanted to remove the data completely within the thousand records within the csv file what I need to do?

There 1000s records are there the 10^{th} record if I wanted to remove the 990 record has to moved up

It is not only modifying the data even deleting the data also difficult job.

We need to copy the data previous and the last record that is there will become duplicate when we shift ,the last record information has to be fill with spaces

Or should be clear

Adding of the data in the middle is not easy.

That means while performing file based persistency operation are going to more complicated managing and updating of existing data, adding a new data or inserting a new data into existing data or deleting is not so easy.

So these operation is quite complicated it take more complexity and more maintenance cost .While writing the complicated logic for storing and managing the data.

It does not permit you to store relation data as part of file

You can’t establish different type’s relation between the data

The data which we store in the csv file format is flat representation of data.

Flat representation of data means a sequential data but always business will contain relational data ,All the data which is store in csv file has relation with each other but in csv format we can’t represent the relationship between the data easily every business has data and data contain relationship between them

It is not good enough to modeling the real world data many time to represent the real-world data we need to duplicate the data in several places within the file

(the same address of the customer, the mobile number of customer, email address of the customer has to be duplicated)

Lets say mobile number has change to some other number we need to change in all the parts of the data where mobile

number is there so that's why maintaining of the data when we duplicate such data.

//These are all the order that are being placed by an

There are bunch of problem with xml then the people start searching/wondering for alternate. The alternate way of storing data into a file came into a picture

XML format.

XML format (Extensible markup language)

XML stands for (Extensible markup language)

It is not a programming language rather it is a language which is used for storing the data in to a file so xml even also called as markup language

What do you mean by markup language?

Markup languages are language that are being build out of tags anything that you surround tags are called markups

<person-name>Dhananjaya</person-name>

That means any piece of data/information that you try to storing in a xml will be stored in mark up ,mark up means tag as this language that is being constructed in tag that's even reason it is called markup language as well

It is used to storing the data.

How do we store the data in xml in which format do we store the data in an xml?

A field will be allocated with fixed size which is called fixed width field size format

A field will be delaminated with comma which is called comma separator value format

Xml has it own notation/representation/rule based on which we have to store the data in xml which is called xml format and xml also has it's own format which we need to store the data in xml

Who has introduce XML and who has brought the XML into picture?

XML has brought by W3C organization where you can store the data easily in a xml format

What is current version of xml ?

The xml only version in market is 1.0 version.

How does xml store the data what is the format in which we need to store the data in xml? When compared to csv and fixed width file size format what are the benefit to go with xml

Xml represent in well defined semantics being attach the information and in structured way

Why I should write the servlet information web.xml and why we should not write web.xls as it store the data in semantic attachment and structured representation.

The data which we store in the" xls file" is not portable, what we configure the data with xls file it can only use in windows plat form not any other plat form But java one time compile and use anywhere at any time it is platform independent but if you are writing information about the servlet in an xls file format which is non-platform portable which makes java as platform dependent for different, different platform we have to write different, different file for Linux something Solaris something etc.

Xml is piece of sequence of character or character represented text file

Every platform can understand xml (Linux, windows, Solaris)

The data that you have stored in xml is structure representation of data that means one can easily understand the boundary of the data starting from and ending from can be easily derivable and well defined semantics being attach that means the meaning about the data is being stored along with the data/information so any one can understand which

part of the data is being stored with the xml representing what kind of information ,So that anyone can understand the data that is being stored in the xml every one able to see the same face of the xml that is a ‘zero’ chance of someone might misunderstand the information that is being stored within the xml. **The share ability of the data is being easy.**

Which programing language can read the xml file?

Whatever programming language which support for reading the file can read the xml file because it just a piece of file that store in a file system

Irrespective of language and irrespective of platform any one could be use xml file to read the information in xml

The data that you have stored in xml not only structured not only has well defined semantics being attach ,the information that you have written as the part of xml will be portable across the platform and languages which is called interoperable that is the benefit of storing the data in xml file.

What is interoperable?

The information/data that we have written as the part of xml will be portable across the platform and languages which is called interoperable

How store the data in XMl?

There are zero keywords for xml there are no predefine tag for xml that are being incorporated or being reserved as part of the language.

In xml we can define our own tag, there are no predefine tag .Xml tag has given greater amount of flexibility to programmer in choosing and creating the names of the tags with your name so that information that your represented seems to be more clear

One has to follow the syntactic rule while storing the

XML tag starts with prolog

Prolog stands for Processing instruction, it is giving instruction to parser which are going to read the content of the xml ,Those parser while parsing this is the xml that has been written based on the version 1.0 syntactic rule and the character that are being written as part of the xml are being represented in UTL8 encoding character set So parser can easily understand what content has been written how to read based on which syntactic rule I need to read it and how to make it decode the character that being stored within that xml can be understood by the parser .

Every xml must should have only one root tag

Why should xml has one and only root element what is the reason behind it ?

According to xml 1.0 version rule every xml should have one and only root element

Allowing more than 1 rot element makes your xml non-structured because we don’t know where the start of document is from where can I start understand the data where is end of the document I have to stop or where I have reach to read the information finally

We will not be able to deny the start of the document and end of the document if there are multiple entry point multiple exit point, to ensure one can understand easily where I need to start from where is going to end with your must and should have one and only root tag element

Xml meant for store the data

We should not write the data in an orbitary format, it must and should be surrounded with tags

How many types of the tags are there?

There are two types of tags are there

1) Starting tag

2) Ending tag

Why I need to store the data within the tag only?

Because one cannot understand the sematic of the data so every piece of information that you have store as the part of xml must and should be surrounded with tags for every starting tag there must be an ending tag ,the level at which you open starting tag at same level close the ending tag

If we follow the all these rules in writing with xml file then it said to be well-formed

Why my xml should be well-formed?

Because everyone can read the data, the readability will be decided by well-formed-ness of the xml file

If the xml is not well-formed then it is non-readable because the tags are not being closed properly

Because we don’t know where the start of document is from where can I start understand the data where is end of the document

what ever we write the data in the xml file is well-formed but not

well-formed always talks about read the data properly

To validate the data which written as the part of the data what should we do?

How to add the data, delete the data

One can easily add the data and delete the data and modify data because there are relevant api that are being provide programming language like java or .net

Every programing languages has provided result set of library for managing the data within the xml retrieving the data is seems to more easy when compared with csv

In case if we store the data in csv I need to access the data sequentially one over the another one there is no way to search for the data there is no way to query the data

There is no way to access the data randomly

When it comes to xml there are techniques called expan query using which any one can traverse within the xml randomly by using the x-query functionality we can search for specific cord through piece of the data within that xml rather than going through each every record of the data that is being stored within the xml

How to validate the data in xml file?

The person who is using the data of the xml will provide that functionality will be able to provide the validity of an xml

Who decide that this is how xml structure should be look like, this must be my root element and chain element?

Vendor will provide all the things, through the dtd file (the validity of xml is defined in the dtd document)

Dtd stands for **document type declaration**

Dtd used for defining the structure of the xml or validating the data of xml

Dtd will be written by vendor and will be given those who want to send the data to the vendor to him they will provide the dtd asking him follow the structure and write the xml

The data that end-user has written may be valid may not be valid for thar the vendor should validate before using that data

How does vendor knows the xml data what we have supplied is valid or not ?

Vendor has to evaluate using dtd file.

How does he know this xml is following which dtd?

Programmer has to provide xml with dtd

/*<! DOCTYPE

Using “<!” this we specify the language compiler that this is declarative tag rather not an element

Advantage of of xml

Xml represent the in hierarchy format so that you can group the data relationship between the data

<si-no>10</si-no> **<!—More content will be added lalter-->**

<cutumer>

<custumer-name>Jhon</custumer-name>

<custumer-address>Ameepet</custumer-address>

</custmer>***/**

The information/data that we have written as the part of xml is Interoperable.

Drawbacks of xml

- Unlike to csv when it comes to xml one has minimum knowledge in xml for storing the data in xml so every one can’t store the data in xml
- Traversing the data in the xml is lot more complicated It is not easy to accessing the data from xml.
- Xml store the data in hierarchy format so that we have to use complicated logic to read the data from xml because reading the data from xml is not easy even friendly api are provided to program those api it is complicated. There are lots of parser API which is available for reading the data from xml like jaxp api, these API are memory intensive ,the parser will take more amount memory

reading the data from an xml (it will take more amount of memory reading ,loading ,concading identifying extracting and retrieving) due to this it will kill the performance of the application

- .Xml is a technology which is surrounded by lots of technology,it will take more amount of time for one to fill sophisticated in working with xml related technology and one has to go through all the relevant APIs and technology within the xml to make use of working with the accessing the data and storing the data within the xml which seems to be more difficult and challenging aspect.
- When I store the data in the form of csv only **real data** is store when it comes to xml the storage additional data attach to it the **actual data** to represent the semantic.
- By nature xml is too much verbose(it provide greater details about the data due to which the memory consumption will be more ,the amount of memory it will take to represent simple piece of the data is high) So, more amount of memory consumption in xml.
- **Xml is not recommended when you wanted to deal with larger sets of data ,records of information (Huge amount of the data will be there as the part of business which cannot be manage in xml because xml is memory inter-chip ,while storing and while retrieving or accessing as well**

and the performance will be the biggest problem while dealing with lakhs of ,millions of records of data)

How read the content of xml?

It use API to read an xml every language has provide certain API for reading the content of an xml

While reading the xml content it is called parsing, I need to use some api for parsing the content of an xml, but it will take more amount memory while reading the data ,for storing also memory will be more

That means most of the API are memory consumption

e.g

For read an one character it will take 10 kb of memory

Reading will be costly and storing the data will be costly

There are lots of related technology related to technology XMl DTD ,XML XLC ,XML XSLT ,XML XPATH,XML XQUERY, etc

That is not an simple technology rather it is bounded with many other technology and techniques of reading and extracting the data which makes it complicated that one has to spend more amount of time in learning technology to read the data there in the xml file.it is not reachable to most the people for storing the data in xml.

Don’t use fixed with file format, csv format, xml what should one has to use for storing the data?

To overcome the problems of xml Java has invented a Serialization
concepts which seems to be good compared with earlier file systems.
Serializable technique concept came into picture

XML(Extensible markup language)

- It is language used to stored data in tag based format.
- XML is invented by W3C org. and the version is 1.0 in current day also.

- Advantages
  - Data will be stored in tag based format.
  - Data will have high relationship b’w them.
  - it has well structured .
  - It has well semantics.
  - Anyone can understand the data.
  - It provided a validator called dtd (Document type Definition and xsd(xml schema definition).
  - Dtd and xsd have protocols about how to write the xml file.It is operable (means it can run in any platform).It does have any keywords (in the prog. World only XML lang. does have any keywords)

- Disadvantages of XML
  - It is not easy for common people to understand it.
  - Without knowing we can't write the xml file.
  - Data will be stored in hierarchical format.
  - Before writing xml we have to know dtd or xsd file.
  - Xml speak more for storing the data.
  - It consume more memory for little data also.
  - We can stored data but while reading we'll get more problems.
  - IOStreams read data line by line, it couldn't identified actual data in
  - xml file. Parsing technology we have to use.
  - B'z of above reason xml also failed to persistence the data in it.
  - To overcome the problems of xml Java has invented a Serialization
  - concepts which seems to be good compared with earlier file systems.
    - by

Serialization

- Convert the state of an object into an bits bytes is what serialization

- Serialization is never connected to file and it not related to file system technology not related to network technology
- If one has an object he wanted to convert this object into representation of bytes so that he can do anything with this bytes of data if he want to store the data into a file or if he want he can transfer the data over the network that is other part this is not called serialization
- **We can use serialization technique to** convert the state of an object into bits bytes and can write into file ,that's where one can use the serialization technique for persisting the data as well
- Its concept provided by Java Prog. Lang. to persist the data into file.
- Serialization means change the status of an objects in bit or byte.
- While Serialization use public static final serialVersionUID to match the exact sender or receiver.

Advantage

- Usually java class holds the data in an object, same object of the data can be transform into file represent system using serialization technique and can write into a

file and access it back. Additional logic for forwarding the data is not required as it already carry the data in object same object can persistence and accessible . In case of serialization the same bit representation of object can be persistable we don't need to write transforming logic for converting the data .

- The serialization will transfer the data into its native format and will be able store the data within the file and can access it. The rate at which it is going to perform **"io"** operation will be drastically or dramatically will perform when compare with any other storage.it will be more faster
- When it convert into bits and bytes representation and store into file no one can use it(because it program understandable format),it is more secure when compared with character representation of the data and it can't stole by any other.
- Storing the data become easy and modifying the data is become easy because we want to modify a data ,First de-serialize into an object and modify the data in the object and again go for serialization.

- Serialization make an object serializable it will transfer through network.

- Using FileInputStream and FileOutputStream as well as ObjectInputStream and ObjectOutputStream we make an object serializable.
- It contains bits or bytes of information.
- No need to verification while accessing the data.
- Data will be in high security mode while transmission.
- Serializable is the marker interface in Java. It doesn't contains any methods.
- Static and final keywords are not usable in serialization.
- Transient keyword used to hide the data while serialization.
- We can perform operation on the serialized data

Drawbacks of serialization

- The data that we store in the file is program understandable format not human understandable format, if the data is not human understandable format there is threat there we cannot use the data independently without the application Only a java application can read the data, non-java application can't across the plat-form across the language (that means interoperable is gone) no one can read the data that is no point for sharing the data with the other people.

- Every piece object would be store in an object that's why system un-manageable instead of it put all object into collection and serialize the connection directly ,always we need to persist all the whole setup data and wanted to de-serialize the complete data back from the serialize data file. whether it is required or not managing the data will not be easy storing and accessing the data when I have bulk amount of data will not be possible

- Serialization only used in Java it is not universal.
- We can stored multiple object using collection but it is complex to read and write.
- It is used by only programmer.
- It may fail while transmission the data from different platform.

The amount of the logic that we are writing for accesing the logic is more than the actual logic for processing the data your business logic seems to be 2 lines of code but actual logic you're writing for retrieving the data from the file seems to be more than the actual data that means seems like file based storage is good but to work with file we need do

more work to make the work to make the data persistency as part of the file

In every application we need to code huge and huge amount of programmatic logic for storage retrieval manipulating and deleting, migrating this aspect has to be applied by every application as the part of the program

Seems like more amount of persistency management logic we are writing rather than actual logic using the data

The cost of the developing application will be high, the complexity will be more (If the project will take 1 yr. Time then only 6 month will be for persistency logic and managing the persistency)

Whatever we store the data in file system that can access by any one, so it is not secure any one can steal the data in the.

We should not go for all this things, go for data-base, data-base will take care of this. Database concept came into picture

DATA-Base/ (DBMS (Database management System)

Database will manage all the thing we should note write any persistency logic if you give the data ,it will know how to store the data and which format it store the data in which location it store the data.

If you ask for retrieving the data it takes care of traversing in records data between the file if you update you just pass the data that you wanted it will take care of updating.

It will take all the data manipulation, file management, file retrieval has to done by Database

It is the responsibility of the data base to secure the data. Only programmer has to know how talk with data base other than doing the persistency .That's why Data-base came into picture

To overcome the problems DBMS has been invented.

- DBMS is the process of storing, managing and retrieving related data from the DBMS.
- DBMS contains four parts

    1.-Hierarchical Database Management System

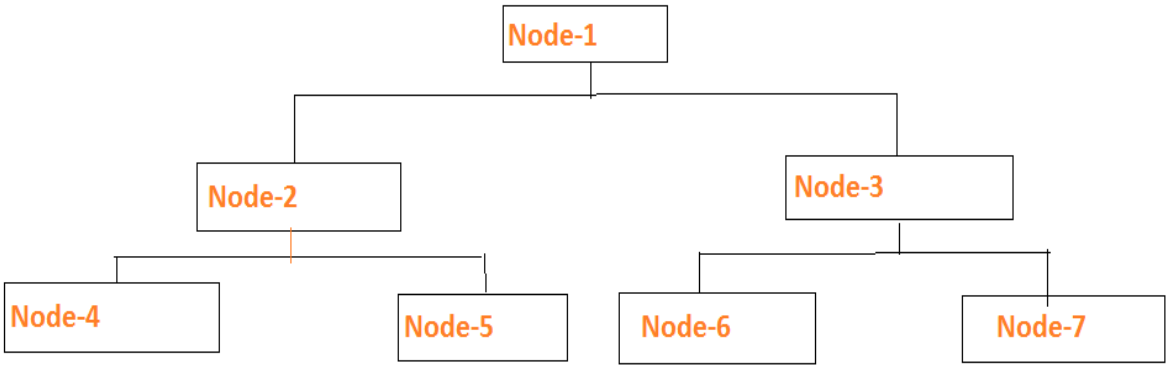
    2.-Network Database Management System

    3.-Relational Database management System

    4.-Object oriented Database Management System

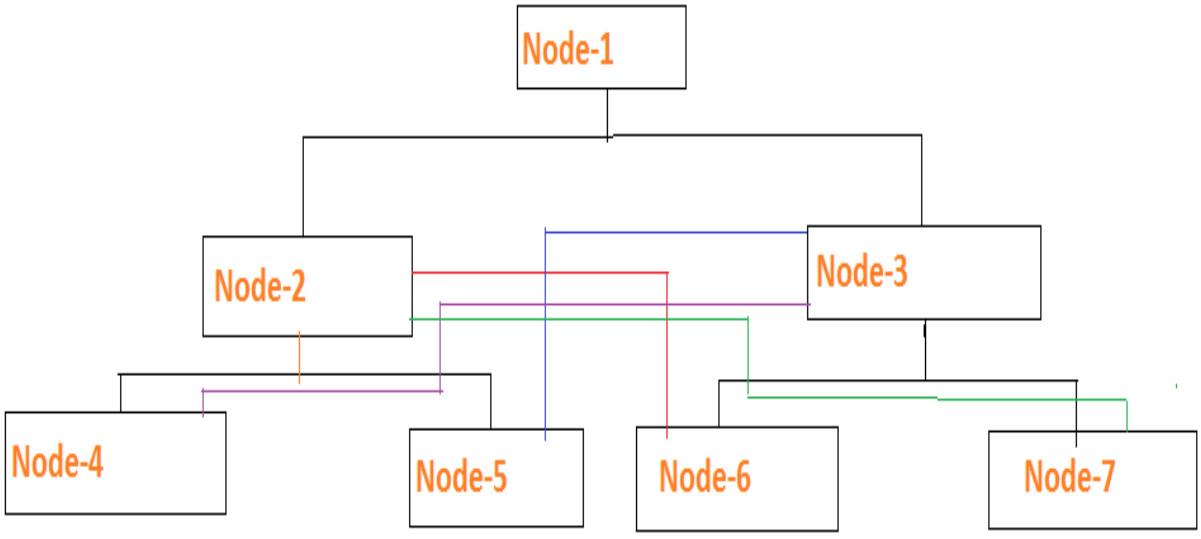
Hierarchical Database:

Hierarchical database comes into picture as the first database management system came into the world.in the year of 1970 being brought of to the market and the key contributed by the hierarchical database management is IBM



Hierarchical DataBaseManagementSystem

- In hierarchical database data will stored in parent and child format which lead to duplication of data.
- We cannot relate one data to another data object B'z of hierarchical concept.
- There is only one bottle neck to access the data it lead to low performance.
- There are many chances to loss of data B'z subsequent data depends on the parent only. Once parent deleted means all the data will lost.
- Data may be in structured format but not in semantics and accessing will be complex.
- -Network Database Management System



Network Database Management System

- In Network database all data are related to each other by address.
- It is critical to keep all the addresses in memory.
- While manipulation it lead to the problems.
- We cannot delete the data without knowing all info. About it.

Relational Database

- It is the world wide used database concept.
- Data stored in the form of tables.
- Relationship between data depend on logical addresses.
- Different types of databases supported by Relational database concept(oracle, DB2, mysql,mysqlServer etc.).
- We can stored related data in it .

- To make data clear and consistence we can use Normalization form.
- It also provide different constraints which helps developer to stored data in persistence manner.
- It uses different model to store the data.
- A table also called as an entity.
- A record also called as a tuple.
- Domain means datatype which allow to store accurate data in it.
- It is easy to persist the data in RDBMS.
- Java provided a package called JDBC API which help you to connect with the data bases.

The data those are coming from relational data-base in terms of records of data

Name of the customer	No.of items purchased	Amount of money
John	2	500
Smith	3	9500
Suzan	3	300

class Salemanagerc {

```
public void salereturning()
    {
    Connection con=null;
    Statement salestsmt=null;
    Satement cutumersalstatement=null;
    ResultSet rs=null;
    Custumer visits;

try{
    Class.forName(“oracle.jdbc.driver.OracleDriver”);
    Con=DriverManager.getConnection(“url”,”uname”,” upass”);
    salestmt=con.createSatement();
    rs=salestmt.executeQuery(“select (*)count
cutumername,count(1),sum of amount from sale(),group by
custmername”);
while(rs.next())
{
 Visit=rs.getInt();

Double amount=rs.getDouble();
If(visit>=&&amount >=12000){
Cutumersalestmt=con.createStatement();
Cutumersalestmt.executeUpdate(“insert into
returningvalues(“custumer_name”,”visit”,amount)”);
```

```
            }
    }
}



}
catch (ClassNotFoundException |SQlException e)
{
e.printStackTrace();

}

    }

}
```

Why one should not intermingle business logic with data access logic?

➔The business logic is exposed where the source data is coming from source system underlying data base (my program is tightly coupled where my data is coming from) Always my business logic is being tightly coupled with underlying source system from where data is coming from

If data is coming from multiple sources system some part of the data is coming from file and some part of the data is coming from database. If programmer want to implement same logic for processing the data i.e. coming file system also and i.e. coming from database , programmer cannot make my business logic talk with multiple source system.

➔If you want to deal with using the business logic across the multiple different source system we need to end of duplicating business logic across the multiple source system where we wanted to work with that

If the business logic is duplicated, the maintenance of the logic become difficult, a change made in business logic has to been across all the places whichever the sources my logic is working

➔The business logic requires data, from underlying source system

If my business logic is getting the data from underlying source system thus my business logic is exposed to the representation of the data from the source system (if table is sales table and tomorrow the table name is changed then my business logic will be impacted, if the column name is amount and it is modify to some other name then business logic will be impacted, if it is storing the data in csv format and

tomorrow it will modified to xml then business logic will get impacted

If your business logic is directly talking with source system you will be exposed to the business representation specific or format specific information about how does the data has been stored within that source system

➔It is not recommended to intermingle business logic with data access logic where you are accessing from

```
class sellmanager{
public void sellRetruningInfor(){
        QualifyCustomer =new ArrayList;
        ReturningCustumers customer=new ReturningCustumers()
        If(customer.getVisits()>=&&customer.getAmount>==12000
    ){
        //logic
        QualifyCustomer.addCustumer();
        }    }
ReturningCustumers {
Private String cutomername;
Private int visits;
Double amount;
//setters and getters
```

```
    }
class SaleAccessor{
public List getReturningCustomer()
    {
List list=new ArrayList();
ReturningCustomer custumer=null;
Resultset rs=stmt.executeStatement(““select (*)count
cutumername,count(1),sum of amount from sale(),group by
custmername”);
While(rs.next())
{
custumer=new ReturningCustomer();
Custumer.setName(rs.getString());
Custumer.setVisits(rs.getInt(2));
Custumer.setAmount(rs.getDouble(3));
l.add(custumer);}
    return list;}
    }
class QualifieldCutomer{
public void addCustumer(){
//logic}}
```

What are the problem when programmer has to write logic for converting Relational model to Object model or vice versa?

We don’t have 1 piece of the data in the table rather we have several piece of columns of data in the table ,we need to write huge amount of logic for dumping the data from relational table to an object, the amount of lines of code that I have to write be more

➔I could be complicated the data to map Relational model to object model ,object model to relational model

For whom we might fetch the data from data base will not be small there could be multiple columns of data might retrieve in every piece of data began from the relational model has to be corresponding attribute of class which seems to be considerable amount of logic it has to be written

➔it seems to be like more amount of logic has to be written

➔Majority of the development effort will be contributed toward managing and accessing the data

➔A simple of the logic querying the data and trying to store an object has to written across multiple places the application

Dt:Nov/12/2016

If business logic called result-set of the data from database ,the business logic will tightly coupled with dataacces technology i.e called jdbc api

My business logic should read or receive the data as an input return the data agnostic to the technology agnostic to the

source system that’s why always pass the data as objects of the and return the data in terms of object of data
That’s why majority of time java application pass the data in terms of object and the returns the data in terms of object
➔We want the data across several places and we want the data that is being quarried from the database will be presented at various different object types the logic for querying and the logic or mapping the data in to an object get duplicated across the all the parts of my application.
➔This encourages duplication of logic in mapping that relational la data the data into object.
➔Many times the data that has been retrieved from the database will be multiple records of data information which has to iterated which has to be grab into an object and has to be pulled into collection and should be return such type of rapidment accumulating into an collection and should return additional logic.
We have to place in several place in our application.
➔Always the data
➔Sometimes we may have to roll up (means multiple record has to one object with list of child)
➔Many time the data query from database will not be map to one object sometimes when we are working on complex query by joining the tables they might yield into multiple record

which has to be rolled up into 1 object such kind of complex logic for iterating and identifying and rolling of the data one object has to be written by the programmer .this is called granular mismatch.

➔Every record of data will not be map into one object sometimes comes at record we wanted to combine them and rolled into one object. The no.of record and the no of object will be same or may not be same .

➔Working with mapping the relational model into object model is to be quite complicated. May problems are there

Conclusion

We cannot go for object oriented databases interms of object The only best bean of storing the data is in

RelationalDataBaseManagementSystem

I can't invest more amount of time ,more development cost and keep on maintain and may have to spent due to the complexcity accesing mapping the data in to object,to avoid these many problem ORM technology come for helping and enabling to works with mapping easily the data from the relational database model to object model that's why ORM technology comes into picture,

➔To avoid above problem researcher owing their thought, investing there time on thinking about how can we make this

data i.e. there in terms of relation model to accessible directly in terms of Object model

➔They try coming with set of rules, principles, mechanism for how map a relational model into an object model or how to map the object model into relational model. Such type of standard and guidelines are provided by those people those name this

Technology as ORM technology.

➔ORM (Object Relational Mapping Technology) is guidelines and rules that helps you enable in mapping the data easily from relational model to object model.

Like ORM ,OOP is also is methodology and it provides guidelines

or rules or principle one has to follow

Don’t write your logic on function if you start writing logic inside function the point of reusability in such kind of procedure oriental programming language is function as 1 function will deals with any no of function tracking the flow of application will become conversion because the only point of reusability is only through function

When you go for procedural oriental programming language you never type data with the logic that is executing that data as the data is being degrade or separated from the logic i.e. processing the data you never have a better control over who

can access which data and who is manipulating which data where you quickly run out of control over manipulating managing the data .this is limitation of procedural programming language

That's why ***Object oriented programing*** principle c

Every piece of logic can written the class, a class will contain data and logic that's where we really together data with the logic so that we have a better control over which logic is using data

Advantage of OOP over the procedural programming language

➔The only way of reusability in procedural programming language is through the function

➔Calling one function inside another function that makes us conversion to understand the flow instead of that my class will contain the data and if any other class want to reuse the functionality of another class you can go through either inheritance or composition

➔There are classes those are executing and expressing in terms of object carrying the data to work on the functionality and you have ***inheritance ,abstraction, encapsulation ,polymorphism,*** all this principle that are provided by OOP principle and methodology

That is being driven by group of experts.

Q➔Using a C programing language can I write my program by following the rules OOP?

Ans. yes I can, but makes me more difficult to apply those mythology and rules to implementing my application when the underlying data is support.

DT:14/11/16

➔Every language has provided language specific API for working ORM technology

➔Initially java has not provide API for working with ORM technology because it wanted to backed up entity beans API which is provided by Sun Microsystem (it helps people working towards accessing the data from relational database model in terms of object so that this gap can be filled)

➔Sun Microsystem has ignored those principle and technology and it started badly promoting the entity bean.

➔Most of java programmer are not interested or against using entity bean

➔As we cannot go for JDBC api it has so many problems and existing entity bean api is useless, their is gap for java programmer to use the relational database model in terms of object model .which was understood by third party vendor and provide third party library which contains programmatic API classes by adopting those principle which is provide by ORM

technology. Hats why these third-party library started ruling in the market that's why Hibernate comes into picture.

JBOSS has provide Hibernate Framework

There are multiple third-party libraries are there within the ORM technology in java

➔ Sun Microsystem has analyses the popularity of ORM technology those are ruling on the market and understand no one is using entity bean that time most popular third party library in the across the world within space of java in ORM is Hibernate provided by the JBOSS the Sun Microsystem request JBOSS to join as official partner building ORM api in java that's why initiative

JPA api started within the java to support ORM technology.

DT:16/11/16

Problem we are going to face while using java JDBC and how can those overcome by hibernate?

➔

1)

➔ Being JDBC is an API ,as it is an API ,it is always partial and it is not completed ,as it is partial it always contain interface and abstract classes, one cannot directly work with API ,we need use implementation to work with API, API are going to provide lots of classes and these classes are interdepend o each other it will more time to learn and

it is highly complicated to understand and program the application using that API.

➔it will not provide boilerplate logic programmer itself have to write boiler plate logic due to which high development cost, high maintenance cost ,more amount of effort will be required and time will be wasted develop the application using java JDBC API and more bugs will be raise.

➔Irrespective to every query that we wanted to execute same piece code we have to write to accomplish the outcome. Which is called ***as boilerplate logic.***

➔Instead of using java JDBC API go for Hibernate, as it is a framework provide prebuild functionality as part of the classes that are zipped as part of framework So the no.of lines of code to developing application will come down ,that time we need to spend is down and the no.of developer we for application will be down, the cost for developing will come down and chance for bugs will come down and the maintenance cost will be less as we didn't duplicate , that one can quickly build the application without writing the boiler plate logic.

2)

➔In case of java JDBC API it provide bunch of classes ,all the classes that are being provided by JDBC API are designed

to through checked exception ,so all the code that you ride as the of java JDBC are surrounded with the ***"try and catch"*** block, as it is biggest dis-advantage while we are working with java JDBC.

➔ Irrespective to checked-exception and unchecked exception all will be reported to my program in when the program is executing or running during the compilation time it will not reported exception will not be arises .Whether its is checked-exception and unchecked exception both the exception will be raised/encountered at Runtime Exception

➔Checked-exception are the exception that are alternate path of execution when my application will be executed it may run in to problem and I can handle them continue the execution of program such problem of that we have alternate path execution which can be handle and recover out of the problem and continue to execute such type of exception are called checked exception.

➔ Unchecked exception are the exception these also will be encountered during execution of the program only when my program has encountered the ***Runtime Exception*** ,encountering terminate even you catch there is no use .ultimately our program will get terminated

So we can broughtly classify Exception into two part

1)Recoverable

2)Non-recoverable

We don’t know whether it is checked or unchecked ,one has to help whether the program is going to run in to problem that can be recoverable problem or not.

If we don’t know it is recoverable problem we never handle it due to which unnecessarily your application terminated rather than executing properly .when we the differentiate it. This understood by sun microsystem then it provide all classes design as throwing checked exception and programmer may not loss any alternate part of execution.

```
Public void readFile(){
    FileInputStream fis=new FileInputStream(“abc.txt”);
}
```

It will through checked exception “**FileNotFoundException**”

If the file is not there, we cannot do anything by handling exception

If it is an unchecked exception compiler will not ask you for try {}and catch block{} after handling it the code will terminated why do i enforce you to write try and catch block.

```
class Salemanager {
public List<String> saleReturning(){
    Connection con=null;
    Statement stmt=null;
```

```
    ResultSet rs=null;
try{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Con=DriverManager.getConnection("url","uname"," upass");
    stmt=con.createSatement();
    rs=stmt.executeQuery(query);
    while(rs.next())
        {//logic}
    }
catch(Exception ee)
{
}
```

If we want to execute the above program we have to write ***try and catch*** or declared as to be thrown

➔JDBC classes is always design to through checked exception due to which I always put my code surrounded by try and catch block While creating the connection obj the exception will raise and it goes to catch block what is alternate path execution we should write to continue execute my program when even not get the connection.so there no alternate path is for executing then why to catch that's why it is called annoying try catch block .

This is the dis-advantage of jdbc.

➔All the classes in java jdbc classes are design to through checked exception even though we write our code surrounded by try and catch block most of the time we dont have alternate path of execution apart from terminating the application by encountering that exception

In this we have not alternate path why jdbc is forcing me to write ***try and catch block*** by throwing checked exception it is biggest problem when it comes to java jdbc.

➔***when it comes to unchecked-exception the flexibility given to programmer whether one can catch or it can ignored when it comes to checked-exception whether you want handle or not you have to write try catch block or declare as thrown.***

Unchecked exception can be caught but some times we might be have to terminated sometimes those may have alternate path of execution

➔When it comes Hibernate it says we leave to you which one you wanted to catch ,which one you don't want to catch we don't want to enforce you to write try catch block always so all the classes in hibernate design to through exception.

➔Programmer has to write complicated logic for managing closing resource within the java jdbc which in not essay.

Managing the resources and the resources that has been created as the java JDBC application are being closed properly are being very complicated work.

➔As Hibernate is a framework it will provide boilerplate logic for
Managing & closing the resource managing the resources and the resources that has been created for application are being closed properly or not.

DT:17/11/16

3)

➔While we are working with java jdbc, the jdbc the logic what we have written as part of the jdbc always enforce we deals with underlying data there as the part of database, most of the time we need to write the logic in jdbc obtaining and referring to the following columns and tables from which we wanted to fetch the data from by expressing such information in terms of query.

➔As we are directly referring to the underlying columns and tables from which we wanted to fetch data from by expressing such information in-terms of queries. But when we have written query we have referred to underlying tables and columns as part of my JDBC logic ,the logic we have written as part of our jsbc application will be tightly coupled with underlying representation of data within there as part of database .Jdbc decouple will not decouple my application from underlying data that is within their table as part of database .

➔Reference to such kind of relational table that we wanted to access the data from are being referred at many place within our application could be the coulomb could be the table could be ant thing within the data base will reference at many place within my application due to which a high maintainability issue will grips into our application when there change might happen in the data within the underlying table as part of database it will make application fragile or completely broken. That means where ever we referred the underlying-table we need to modify which incurs severe maintenance cost .

➔When it comes to hibernate, it never let your application classes access the data from underlying relation database tables

Rather hibernate permits me to access the data that is there within the relational tables in terms of object only .

➔Hibernate protect us to accessing the data from the data bases.

➔ Hibernate never let me access the data from the database in an relational model

➔Hibernate allows my application logic to refer the data from a relational model in-terms of Object .you tell me which class object you want the data because I know what is the class what is the table I need the data from

➔Within the application logic it is not referring the underlying database and corresponding table and colums.it always query that data in-terms of object of data.it helps for decoupling from the underlying database. Any change within the underlying database will not modify the logic in all the places we need to tell the Hibernate for this class, this table is there.

➔Whenever we query the data from same table ,the object in to which we wanted to complete data will be same object whether you querying class1 or class2 or any other place entire application where ever you wanted to access the data from the same table data will be populated into same object.

➔By writing a properties file I need to inform hibernate these are my table and these tables data has been populated into these classes.

➔***Conclusion:-***Unlike to jdbc when you go to hibernate never your application logic will refer to underlying table rather your application logic refers to the underlying table rather your application logic in terms of objects ,hibernate never let you see the data in Relational table rather it always access the data referring to those object

As we never refer to the table where you want to get data rather you always deals with the data by using those objects you application logic will completely decoupled from

the data within your database ,zero maintenance cost will be there even the database and underlying table are being changed

4)

➔In case of java JDBC the logic we have written will be portable across the all database but the query what have written will not be portable across all the database due to which there will be a maintenance cost will be involved some amount of logic or at least the query that you have written within your logic will be written for switching from data base to another database.

➔If the query will be change the data that is return will be change

The logic that is reading the data will be modified when we switch the from one database another database.

➔Relational Database Management came with concept called sql (Structured query language)

Sql92 standard language is there, the first sql language there is sql92 standard that comes with query clauses and condition clauses and statement, that one can query the data in any Relational Database Management .Every database support sql92 standard language.

Beyond the sql92 grammar language every database provide their own set of clauses helping the developer easily building the query for access the data in complicated way. All the query will compatible only few query will not compatible those who are refer to database specific clauses.

➔To over come this problem hibernate says don't write sql query write hql query

5) We need not wary about the SQL query language while working with the hibernate. B'z hibernate provided HQL to the programmers, to works with the database.

• Without knowing single query also we can work with hibernate.

6) Hibernate use dilate internally B'z as per programmer requirement he can switch databases. From oracle to mysql, mysql to myseversql ….etc.

• Hibernate has provided one criteria class with take care of managing the all HQL queries.

7) Most powerful advantage is hibernate has a caching memory, it improve the performance of the application .

• It keep your data in caching and as per request he give the data.

8) Hibernate has version feature which is take of maintaining the version of the modified record.

• If we want know how much time a data has modified or inserted in the database so there is no need to write extra logic, Hibernate has
Provided VERSION feature which take care of maintaining record.

9). Time Stamp it is also one of the feature of hibernate, if we want to know when it is modified or when it is inserted in the database then Time Stamp will going to keep the records.

10). Hibernate manages the Transactions internally we need not be wary either it is local transaction or Global Transaction.

Q)How can 1 class can related with another class ?

There are two ways that one class can related with another class in two ways

Either via Inheritance or via composition.

Inheritance

A class can reuse the functionality of data that contains in another class by extending from one class of another class, by using the extend key word. So that all the data in parent class is part of child class, inheritance always establish **IS-A** relationship.

Composition

One class have other class as an object contain within my class which is establish **Has-A relationship. So that using that reference of the object contain in my class I can access the other class data and functionality.**

From an UML prospective the inheritance can be of multiple forms inheritance can have following types

1)Generalization

2)Specialization

3)Realization

There is nothing being called composition rather it is called association, It is upto three types

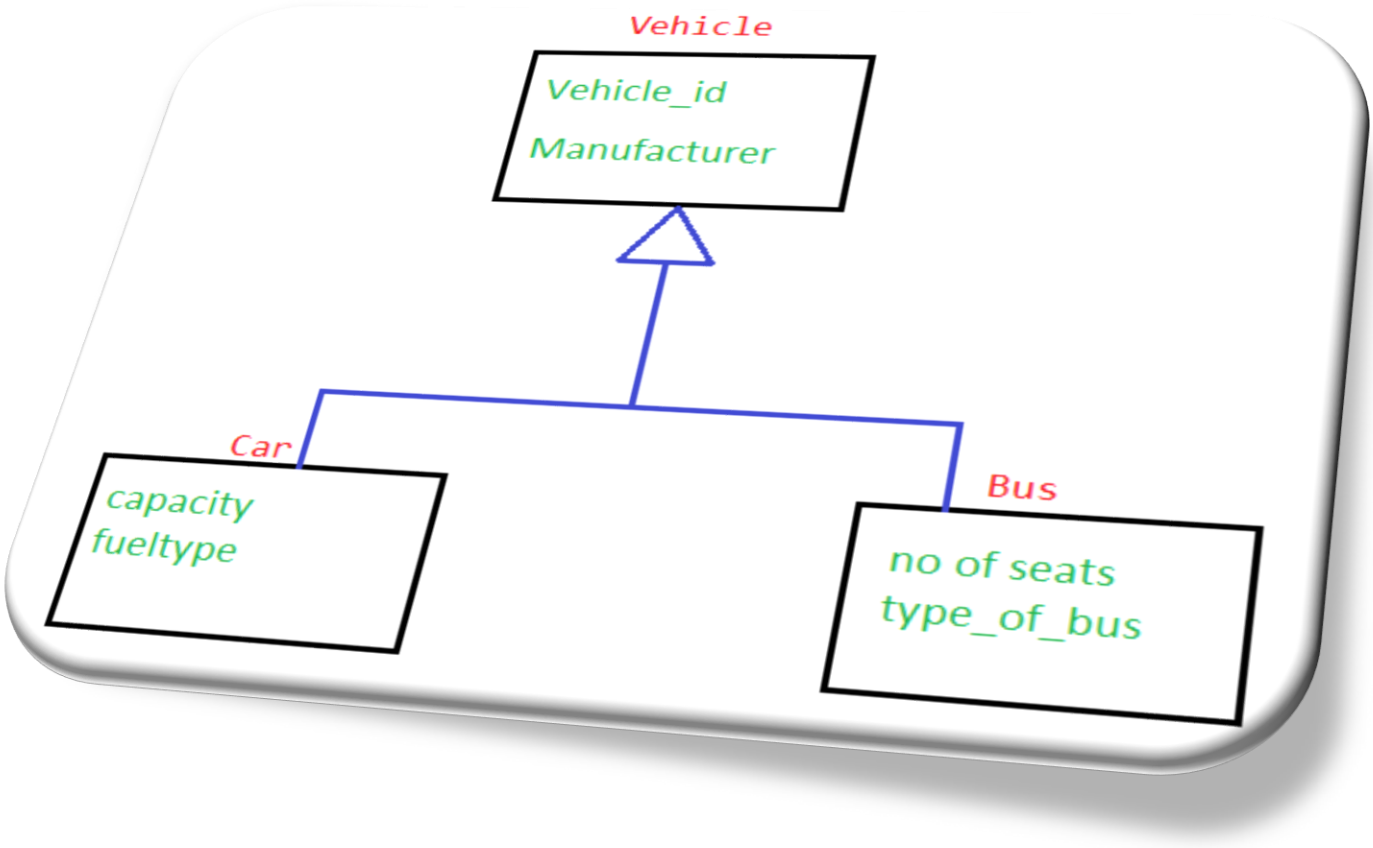
1)Association

2)Aggregation

3)Composition

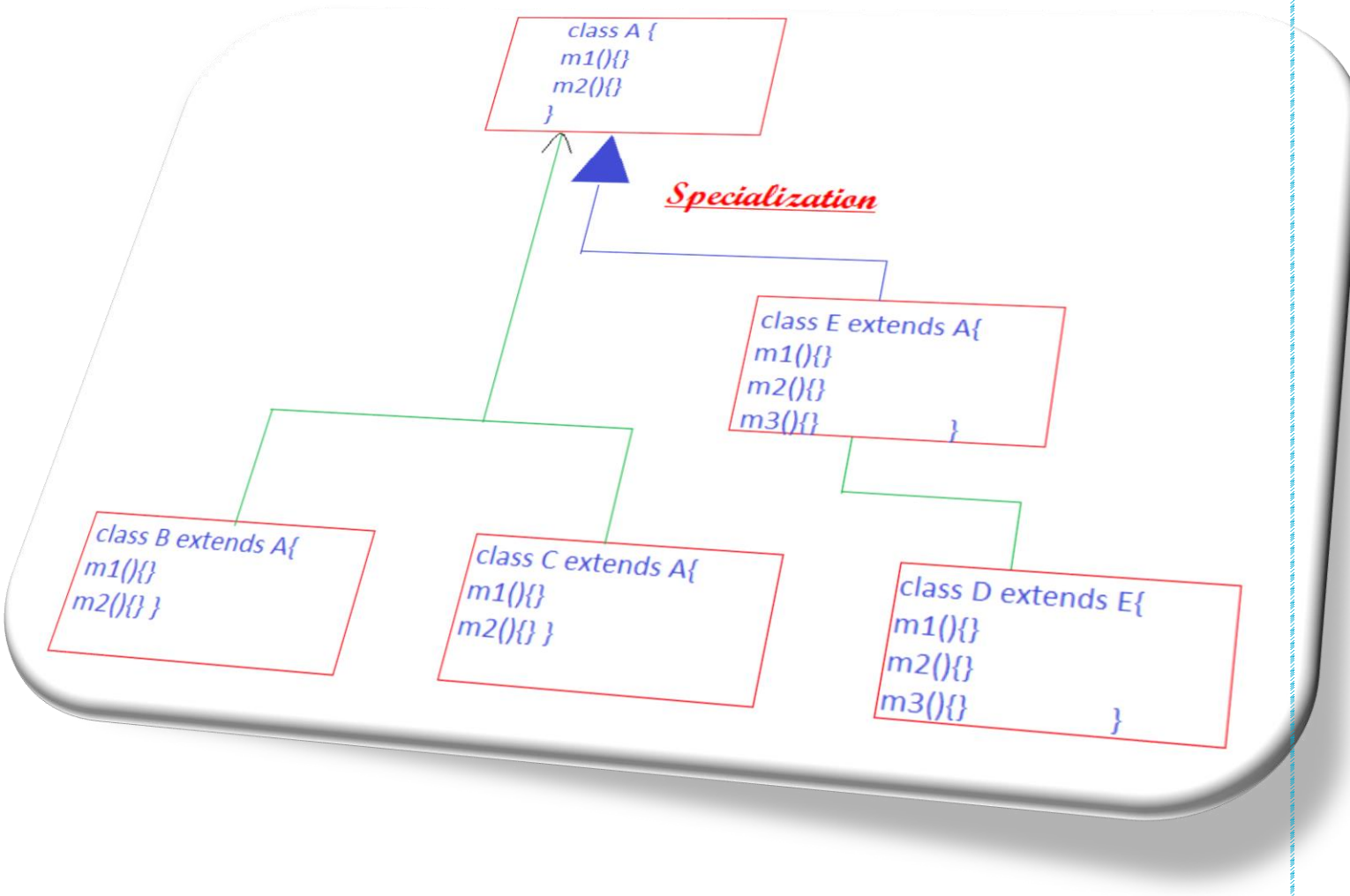
Generalization

Identify the common functionality that exist across multiple classes within your application, we write such functionality and placed it in super class where generalized of all classes being push into the parent class which is called generalization.



Specialization

It is also another form of inheritance only ,here also classes are being inherited from each other ,from the existing set of classes we identify something seems to be common ,will create common class representing the data that have to generalized and shared across all the classes ,this process is from bottom to top .

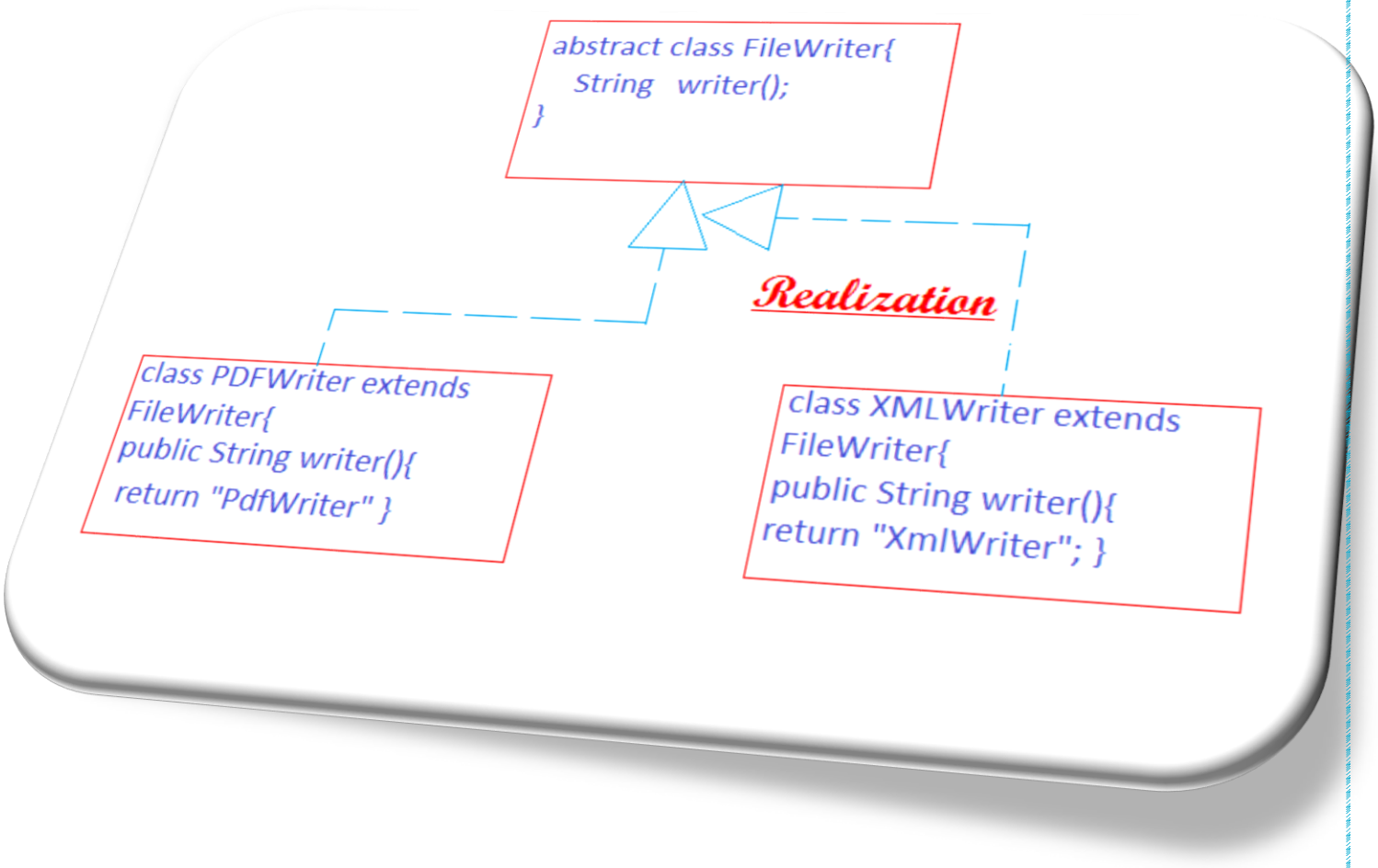


Realization

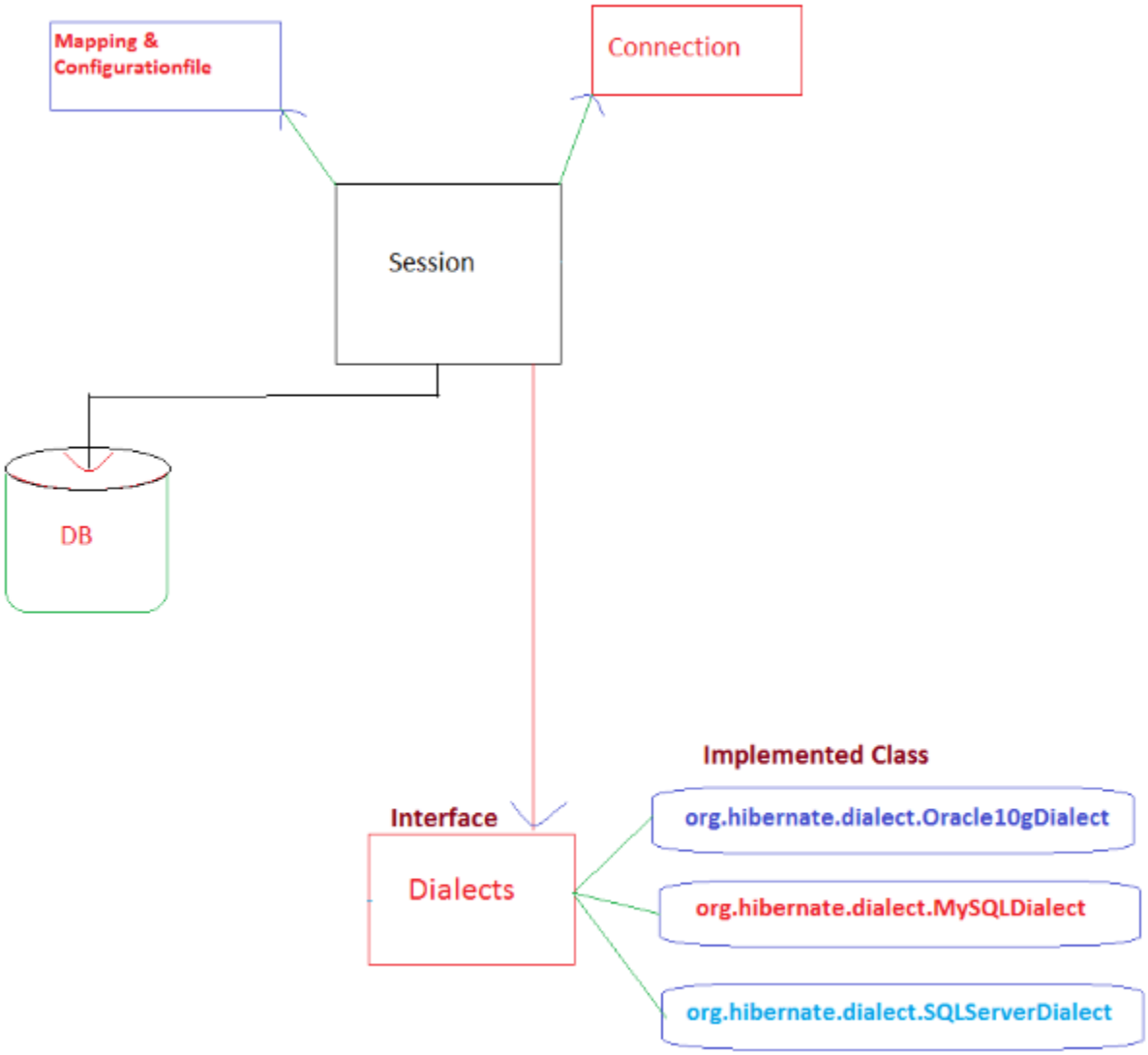
Sometimes we don’t want to re-use functionality of a class sometimes reuse only the proto-type(representation of class) of class another class ,the declaration of method only we want but implementation of those method will differ from class t the class which is called realization .

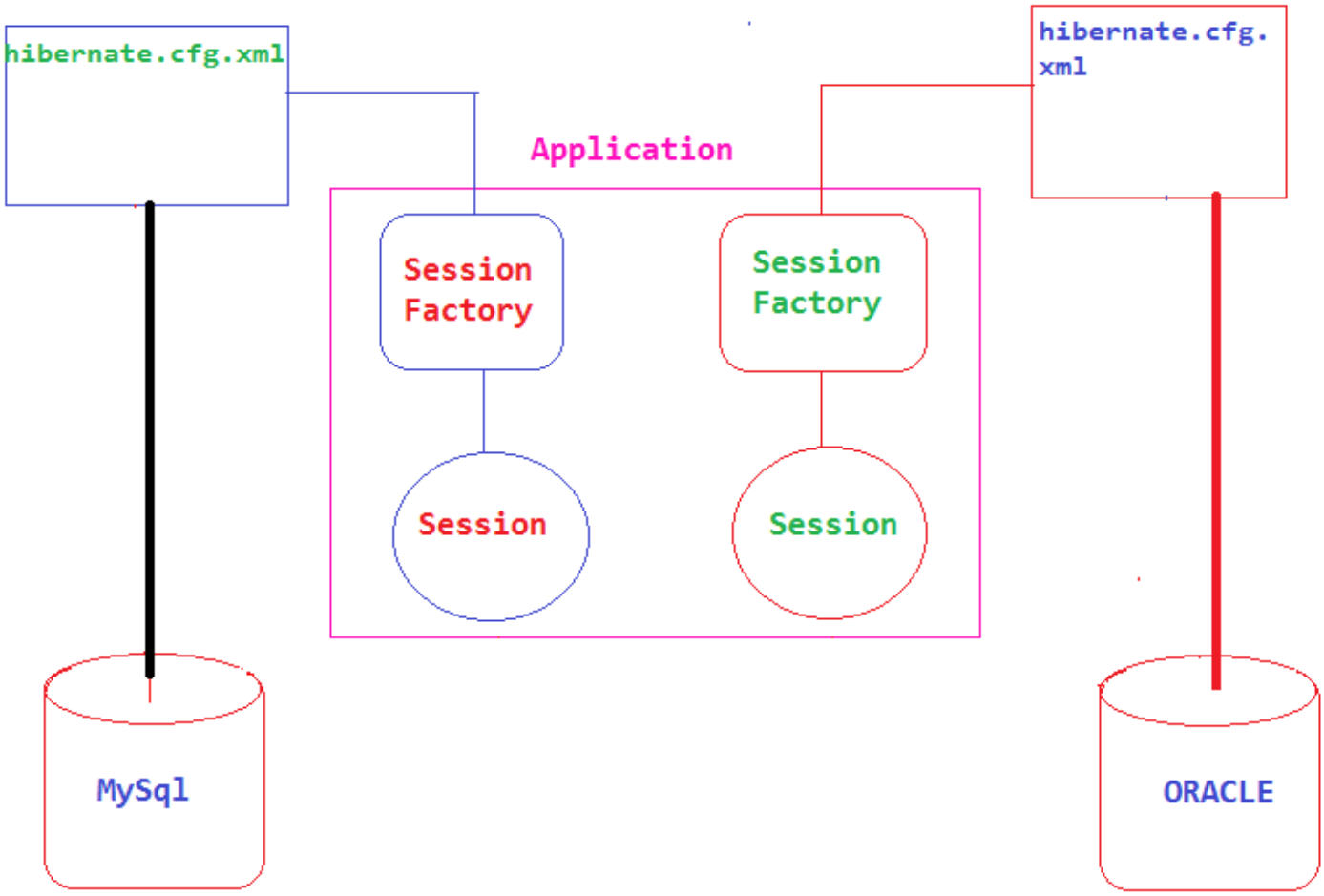
A common prototype for which different, different implementation has to be provide across multiple classes within my application.

In such case if you derive an implementation from a prototype of an existing class then it is call realization which is an interface and it’s implementation



How Hibernate works


Mapping &
Configurationfile
Connection
Session
DB
Implemented Class
Interface
Dialects
org.hibernate.dialect.Oracle10gDialect
org.hibernate.dialect.MySQLDialect
org.hibernate.dialect.SQLServerDialect



➔For a data-base only one Hibernate configuration file can be added within an application and configuration file must be named with hibernate.cfg.xml and it will be place in the class path(src folder).

➔For a Mapping file class we can add multiple entity class for same database. The file name you can give any name but it is recommended to give same name as entity class-name and place it inside same package as entity class is.

➔It is not mandatory to configure Dialect inside mapping file.

➔For every property column name is not mandatory

If we not specify the column-name, the same name as we define as the part of entity-class member .follow convention over configuration.

Q->Why does the hibernate doesn't perform validation? Does hibernate will perform such validation performing such persistency operation underlying database?

Ans:- Whenever we call *session.Save(object o)* it goes to entity class object finds the corresponding class type

For that class type it search for metadata which we will configure already as the part of mapping file

By looking at which it can derive what constraints are there for the table and column each and every thing ,it can go to the data base and it query the data base and it can validate against data supplied in mapping information and against the data-base table and it can validate whether mapping information is correct or not as your provided table. After that it can store the data into database.

➔ For 2nd time when we call session.save() the above operation will be perform for that performance will be badly impacted it is a performance issue that's that's why during the Run-time or execution of your application time hibernate never perform validation.

Q-Why we are configuring database column information as the hibernate is not performing any validation check?

Ans-:Hibernate helps us into creating the table for the corresponding mapping file for the automation of creating the table can be done from the hibernate rather programmer m anually creating those tables and working with that application .

Q->The type what we are mentioning is java type?

Ans-The language what we specifying in mapping file hibernate type not a java type.Using this hibernate can understand that what is corresponding table I have to create within the database table and query from column how do I need to convert into which type so which can be populated as the part of entity class attribute would be determine by hibernate using that type.

Q->is the type is mandatory to specify?

By using introspection it goes to the class and find the corresponding attribute and identify the java type

➔Hibernate provide its inbuilt connection-pool maintain (Every java EE application server will have connection pool using data source find it to JNDI registry i.e. called J2EE connection pooling)

Q->What is factory design pattern?

Ans-:Factories are meant for creating the object of another class .when we cannot create the object of class easily or we have to write complex logic for creating object when such

complex process is involved don’t write such complex logic for creating all the classes because the code will get duplicated ,to abstract form complexity creating thes object Factory methods are Provided or developed.

Q->What is builder designpattern?

Always a FactoryClass will created object of another class by initializing value of the object to default values.but may not we want object to create with empty value we want to create the object with some user supplied value as an input even though we can pass the parameter to the factory Method so initialize the state of an object some times we want many object of the same class.Don’t go for factory rather go to builders.

Builder are also factory that helps creating the object of another class but first you create the object of builder class and populate the data into the builder instance with which you want to create the object original class never builder will create empty object always the objet will create the object of actual class with prepopulate data.

e.g

You give your data to configuration and ask the configuration to create the data object of your SessionFactory

Here the configure() is a method which you call for populate the data into configuration object .So configuration object is for creating the session factory.

When we call session.buildSessionFactory() it will create session with the data that is in Configuration. Thats why configuration always act as builder for session factory.

```
package com.fhp.test;

import org.hibernate.Session;

public class FHTest {
    public static void main(String[] args) {
        Configuration configuration=new Configuration().configure();
        SessionFactory factory=configuration.buildSessionFactory();
        Session session=factory.openSession();
        Student student=(Student)session.get(Student.class, 1);
        System.out.println(student);


    }

}
```

builder

Q->What is the designpattern that the configuration class follows?

Ans-:Configuration class follows builder DesignPattern.

Q->Session Factory is single-tone or non-single-tone?we are creating only one session factory object within our application in such we can call session factory as single-tone?

Ans-:we are only creating only one SessionFactory object within an application, the class SessionFactory is not single tone ,but SessionFactory has been managed or has been created as singleton within our application never the hibernate has provided SessionFactory class as singleton but in our application we are ensuring the SessionFactory created as singleton object

Q-:Why does the hibernate people has created SessionFactory as non-Singleton when we are not going to create more than one object of SessionFactory within our application?

Ans:- SessionFactory cannot created as Singleton class. Because while working with multiple databases we need one SessionFactory. SessionFactory is not singleton as in application level .SessionFactory is singleton at database level. So if our application working on two database that we need to end up creating with two SessionFactory object. That means we should make the SessionFactory singleton per database.

he

```java
package com.fsm.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateFactoryMethod {
    private static SessionFactory sessionfactory;
    static{
        sessionfactory=new Configuration().configure().buildSessionFactory();
        }

    public static SessionFactory getSessionfactory() {
        return sessionfactory;
    }
    public static void closeSessionfactory(SessionFactory sessionfactory) {
        sessionfactory.close();
    }


}

```

Transaction

There are two types of transaction

1. **GlobalTransaction**
2. **LocalTransaction**

GlobalTransaction is support by JDBC ,It only support local transaction

To work with GlobalTransaction java has provide two JTA api

To work with local transaction following are the code. Hibernate has provided feature for working with both local and global transaction

How to work with Bootstrapping

What is Bootstrapping Hibernate?

Ans-:how can we instantiate and initialize the core component of hibernate .

There are two ways of working with Boostrapping in hibernate

1. **"Legacy approach" older API**.(Hibernate[3.x](Legacy))
2. **"ServiceRegistry approach"** (Hibernate[4.x]and[5.x])

Till Hiberanate 3.x there is only one ways of boostrapping the hibaernate which is **"Legacy approach" older API**.

There is an Modern approach which is called **"ServiceRegistry approach" which** is there from hibernate 4.x onwards

From hibernate 5.x onward Legacy approach has been deprecated

Most of application are using Hibernate[3.x] and application is build in 3.x and all the existing project are 3.x.

1. **"Legacy approach" older API.**(Hibernate[3.x](Legacy))

In Legacy approach there are three ways of Bootstrapping *the hibernate*

1. ***Property Approach***
2. ***Configuration driven XML based appraoach***
3. ***Programetic Approach***

❖ **Property Approach**

✓ Don't write the such information as part of xml based configuration file whether you can equally write such information in the properties file because all together specified as part of the configuration file, a simple set

of key and values which you can equally configure as part of the properties file also rather than a configuration xml file.

- ✓ A property file should have fixed name should be and fixed location.
- ✓ property file should not be changed (hibernate.properties) the name of the file must and should be “hibernate.properties” only and the location of the properties file should be definitely under the class path(src), even if you put it into sub-directory it will not work.
- ✓ We have to configure information about the database

Or dialect or various information related to database required for my hibernate in terms of keys and values where every key that we have configured as part of properties file has to have a common prefix called “hibernate.connection.driver_classname” to ensure this property file that you have written will not conflict with any other property

- ✓ In configuration driven xml approach there is xml which is validate through dtd linked to it ,that’s why every element comes form dtd which has an namespace which avoid conflict from one element from another element and can validate it but here there no such kind of mechanism is

there they used a common prefix called "hibernate.connection".To identify that this key is related to hibernate it just uses a common prefix called "hibernate." .*It will not identify the property that you are asking hibernate to use these are not element*

##

```
package com.bsp.entity;
public class Project {
	int id;
	String title;
	String description;
	String domain;
	String duration;
	String client;
	String status;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public String getDescription() {
		return description;
	}
	public void setDescription(String description) {
		this.description = description;
	}
	public String getDomain() {
		return domain;
	}
	public void setDomain(String domain) {
		this.domain = domain;
	}
	public String getDuration() {
		return duration;
	}
	public void setDuration(String duration) {
		this.duration = duration;
	}
	public String getClient() {
```

```
            return client;
      }
      public void setClient(String client) {
            this.client = client;
      }
      public String getStatus() {
            return status;
      }
      public void setStatus(String status) {
            this.status = status;
      }
      @Override
      public String toString() {
            return "Project [id=" + id + ", title=" + title + ", description=" +
description + ", domain=" + domain
                        + ", duration=" + duration + ", client=" + client + ", status=" +
status + "]";
      }
}
```

Project-hbm.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="com.bsp.entity.Project" table="project">
      <id name="id" column="id"/>
        <property name="title" column="title" />
        <property name="description" column="description" />
        <property name="domain" column="domain" />
        <property name="duration" column="duration" />
        <property name="client" column="client" />
        <property name="status" column="status" />
      </class>
</hibernate-mapping>
```

```
package com.bsp.util;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class PropHibernateBeanFactory {
private static    Configuration configuration=null;
private static    SessionFactory factory=null;
      static
      {
            try{
            configuration=new Configuration();//To understand hibernate to read Poperties
file
            configuration.addResource("com/bsp/entity/project.hbm.xml");
            factory=configuration.buildSessionFactory();
            }
            catch(Exception e){
                  e.printStackTrace();
```

```
		}
	}
	public static SessionFactory getFactory() {
		return factory;
	}
	public static void close()
	{
		if(factory!=null)
			factory.close();
	}
}
```

Hibernate.properties

```
hibernate.connection.driver_classname=oracle.jdbc.driver.OracleDriver
hibernate.connection.url=jdbc:oracle:thin:@localhost:1521:xe
hibernate.connection.user=sahid
hibernate.connection.password=sahid
```

```
package com.bsp.test;
import org.hibernate.Session;
import com.bsp.entity.Project;
import com.bsp.util.PropHibernateBeanFactory;
public class Test {
	public static void main(String args[]) {
		Session session = PropHibernateBeanFactory.getFactory().openSession();
		Project project = (Project) session.get(Project.class, 1);
		System.out.println(project);
	}

}
```

Configuration driven XML based appraoach

➔Configuration File name can be any name(it is not mandatory to be "hibernate-cfg.xml") and placed in anywhere in the application but it should be under the class path

```
package com.bsp.entity;
public class Project {
	int id;
	String title;
	String description;
	String domain;
	String duration;
	String client;
	String status;
	public int getId() {
		return id;
	}
```

```
	public void setId(int id) {
		this.id = id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public String getDescription() {
		return description;
	}
	public void setDescription(String description) {
		this.description = description;
	}
	public String getDomain() {
		return domain;
	}
	public void setDomain(String domain) {
		this.domain = domain;
	}
	public String getDuration() {
		return duration;
	}
	public void setDuration(String duration) {
		this.duration = duration;
	}
	public String getClient() {
		return client;
	}
	public void setClient(String client) {
		this.client = client;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
	@Override
	public String toString() {
		return "Project [id=" + id + ", title=" + title + ", description=" +
description + ", domain=" + domain
				+ ", duration=" + duration + ", client=" + client + ",
status=" + status + "]";
	}


}
```

project.hbm.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="com.bsp.entity.Project" table="project">
     <id name="id" column="id"/>
        <property name="title" column="title" />
        <property name="description" column="description" />
        <property name="domain" column="domain" />
        <property name="duration" column="duration" />
        <property name="client" column="client" />
        <property name="status" column="status" />
      </class>
</hibernate-mapping>
```

hibernate.cfg.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.driverclass_name">oracle.jdbc.driver.OracleDriver</pro
perty>
           <property
name="hibernate.connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
           <property name="hibernate.connection.username">sahid</property>
           <property name="hibernate.connection.password">sahid</property>
           <property name="show_sql">true</property>
           <mapping resource="com/bsp/entity/project.hbm.xml"/>

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

ConfHibernateBeanFactory.java

```
package com.bsp.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class ConfHibernateBeanFactory {
```

```
	public static SessionFactory sessionfactory=null;
	static{
	Configuration configure=new
Configuration().configure("com/bsp/entity/hibernate.cfg.xml");
		sessionfactory=configure.buildSessionFactory();
	}
	public static SessionFactory getSessionfactory() {
		return sessionfactory;
	}

}
```

Test.java

```
package com.bsp.test;
import org.hibernate.Session;
import com.bsp.entity.Project;
import com.bsp.util.ConfHibernateBeanFactory;

public class Test {
	public static void main(String args[])
	{
		Session
session=ConfHibernateBeanFactory.getSessionfactory().openSession();
		Project project=(Project)session.get(Project.class, 1);
		System.out.println(project);

	}

}
```

ProgrameticApproach

In programmatic approach you can avoid both properties file and xml configuration file to Bootstrap your hibernate rather all such required configuration and the mapping information that is required for your hibernate component can be fed as an input to your classes using Programmatic approach
***you can also can encrypt the password and decrypt the password using programmatic approach**
***If we have proprietary configuration file from which I need to read the configuration to build my configuration object with which I need to create session-factory and session ,then go for programmatic approach(scenario legacy application while adding a new module using hibernate)**

***While working with encrypted password you should write all the information in the xml file except password .encrypt the password and store it into an properties file decrypt at runtime**
“*configuration*.setProperty("hibernate.connection.password",decriptr_password);
”
In this situation we need to use programmatic approach with XMl driven approach.

```
package com.bsp.entity;
public class Project {
	int id;
	String title;
	String description;
	String domain;
	String duration;
	String client;
	String status;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public String getDescription() {
		return description;
	}
	public void setDescription(String description) {
		this.description = description;
	}
	public String getDomain() {
		return domain;
	}
	public void setDomain(String domain) {
		this.domain = domain;
	}
	public String getDuration() {
		return duration;
	}
	public void setDuration(String duration) {
		this.duration = duration;
```

```
	}
	public String getClient() {
		return client;
	}
	public void setClient(String client) {
		this.client = client;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
	@Override
	public String toString() {
		return "Project [id=" + id + ", title=" + title + ", description=" +
description + ", domain=" + domain
				+ ", duration=" + duration + ", client=" + client + ",
status=" + status + "]";
	}


}
```

****

project.hbm.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="com.bsp.entity.Project" table="project">
      <id name="id" column="id"/>
        <property name="title" column="title" />
        <property name="description" column="description" />
        <property name="domain" column="domain" />
        <property name="duration" column="duration" />
        <property name="client" column="client" />
        <property name="status" column="status" />
      </class>
</hibernate-mapping>
```

```
package com.bsp.util;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class ProgHibernateBeanFactory {
	private static  Configuration configuration=null;
	private static  SessionFactory factory=null;
	static
```

```
	{
		try{
		configuration=new Configuration();/* it will go to classpath and
 search for hibernate.properties and it  will not found and create empty object
*/

	configuration.setProperty("hibernate.connection.driver_classname","oracle.jd
bc.driver.OracleDriver");();/*Here key is fixed */

	configuration.setProperty("hibernate.connection.url","jdbc:oracle:thin:@loca
lhost:1521:xe");
		configuration.setProperty("hibernate.connection.user","sahid");
		configuration.setProperty("hibernate.connection.password","sahid");
		configuration.addResource("com/bsp/entity/project.hbm.xml");
		factory=configuration.buildSessionFactory();
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}
	public static SessionFactory getFactory() {
		return factory;
	}
	public static void close()
	{
		if(factory!=null)
			factory.close();
	}
}
```

```
package com.bsp.test;
import org.hibernate.Session;
import com.bsp.entity.Project;
import com.bsp.util.ProgHibernateBeanFactory;
public class Test {
	public static void main(String args[])
	{
		Session session=ProgHibernateBeanFactory.getFactory().openSession();
		Project project=(Project)session.get(Project.class, 1);
		System.out.println(project);
	}
}
```

2.Modern Approach “ServiceRegistry approach” (Hibernate[4.x]and[5.x])

Hibernate Session-Factory will always create a connection with 1 approach only

Rather it will talk with service classes to provide connection

If we want the connection for c3p0pool Hibernate Session-Factory will go to c3p0 Class to create the connection pool

if we want Connection from Proxool connection pool then it will then it will go to Proxool class.

Session factory always talks to Interface called as Service where all these are implementations of the interface.

That’s why Session-Factory will never bother where the connection in coming from Proxool , c3p0,these are implementation of the service

Session-Factory are designed to talk with Interface

*if some other vendor provide Connection pool then write 1 implementation for service interface which plugin the session factory reference variable called service that is also implementation of base class type

To works with various connection strategy, for getting a connection hibernate application Session-Factory not responsible to creating the connection .That’s where Session-Factory should be able to get the connection from whoever give it

In hibernate 4.0 onwards it will come with plugability

Earlier Session-Factory directly talks with specific set of components to acquire the connection.

In modern approach Session-Factory will not create Connection .if it will create the connection it will specific one database then hibernate will tightly couple with specific set of services then extensibility of framework will not be there.

That’s why 3.5 version hibernate start redesign its framework

It introduce service as component’s

Services are the classes and components that provide a specific set of services that are required for hibernate components to work with persisting the data.

If session has store the data then it need connection the it goes to Session-Factory to get the session.

Then Session-Factory goes to service to get the service ,which service jdbc service ,jee service ,c3po service any of the services session factory can talk to . Such kind of object can be acquire by service api that is being newly introduce in as part of your hibernate

1)Connection(pluggable)

2)Transaction

3)ClassLoadingService

4)StartegySerice

In this way lot of component that are required as an helper component for hibernate to work with offering the persistency functionality to work with application.

Service

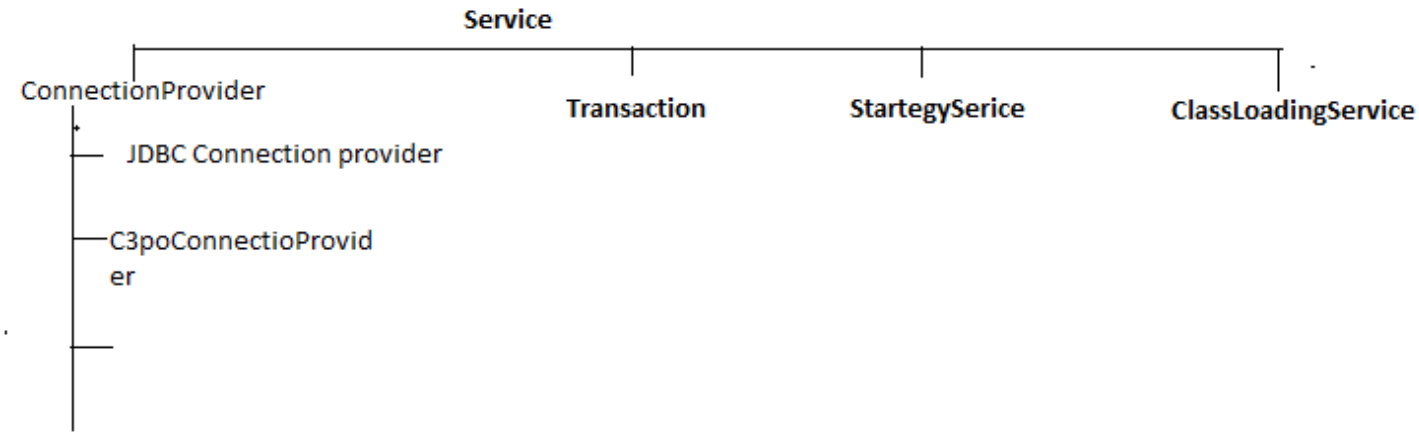
Service is an Marker Interface

Service doesn't have any method to just ensure all the services that are there in hibernate are common type called service type to created service as marker interface.

The service is an marker-interface this is also called as service role class ,that means it plays an role (Indicate the nature of it)

Service is the base interface and for each service 1 sub-inteface is there for one, one service type

Connection Provider



All the services has not same api

(for jdbc connection we getCoonection(),for transaction getTransaction(),for cache we getCahe())

They can't have different service with different classes they want common super class that’s why they created service as a marker interface i.e. all the services are service type.

Service is a components it performs a specific functionality that is required for my hibernate components to work with an external object can be acquired by hibernate components through the help of service it is the pluggable component.

Till 3.5 service API is not swappable, you can provide own service and ask hibernate use this service

Session-Factory will talk with one of the implementation class of the Connection Provider.it will not create the connection or it will not directly talk with concrete class

We can create our own custom connection provider

Q who will manage the dependency? Who will provide one service with other service?

We need some container who will manages the dependency and lifecycle that’s where hibernate comes with concept called ServiceRegistry (related to ioc container)

All the service that are in the hibernate will be kept hold with service registry

ServiceRegistry will contain bunch of services

The scope of service will determine by the registry to which it has been binded, i can't get service unless until i go through the registry

If i have access to registry then only i can see the services that are been binded to the registry

a service will blinded to registry can be access through the registry only. one cannot directly access the service independent of an registry that means service is scope to the registry to access.

Hibernate component should go to registry and should ask the registry for an service then service registry will determine which service will requesting and gives

once you get the service you can call the method on object .

The Process of putting an service into an ServiceRegistry is called ServiceBinding.

This is called binding a service to ServiceRegistry, Instantiating one service to service may differ ,Instantiation of a service is specific to service to the service ,we take care of binding service to service registry

There are two ways of binding a service to service-registry

1) We populate the services by Instantiating (if you have own service, you cannot plugin the service into hibernate you need to plug-in to ServiceRegistry, but a service registry cannot instantiate service you instantiate the service, and populate the ServiceRegistry i.e. manually)

   It eagerly load service which we require later point of time but it use unnecessarily jvm memory. Which waste of jvm memory

2)To instantiate the service we need to write logic in Service-initiator

And gives this to ServiceRegistry (when someone asking about my connection provider ,Neither you can create nor I,I wrote the logic for creating service in service-initiator ,

You go to the service-initiator ask him to create ,he takes care of creating and moving the service to you so that you can host and distribute this is for lazy loading, lazy instantiation)

ServiceRegistry will not know how to create the object ,as it doesn't know how to create the service ,Either developer populate those services into the registry or let us write that logic in instantiating those services in an service-initiator ,plug-in that service-initiator into the ServiceRegistry

Whenever some-one required that service it comes to service-registry

ServiceRegistry will determine this service can be instantiated by which service-initiator.(this is service-initiator this will create this service)

Once some-one asking service it goes ,and see they have service object or not ,no , then is there any service-initiator that is capable of creating the service or not ,yes ,then ask the service initiator to create the object and return So that ServiceRegistry will put inside and distribute it.

These ServiceRegistry are there are acting as container and these ServiceRegistry hierarchical (multiple ServiceRegistry are there ,those are not one ,they can be multiple ServiceRegistry can be there within an hibernate architecture or within an hibernate echo system ,and these ServiceRegistry has relation between them as hierarchical that means one in parent another service-registry are child

The dependency between the service can not only will be manage between the other service that are there in the same registry these dependency can be manage across the hierarchical of ServiceRegistry as well)

You can manage the services across the ServiceRegistry inside the hierarchy ,I should create ServiceRegistry with services or service-initiator.

Q-How to create ServiceRegistry?

Ans:-There something called ServiceRegistryBuilder ,For evevry type of ServiceRegistry there is one ServiceRegistryBuilder.

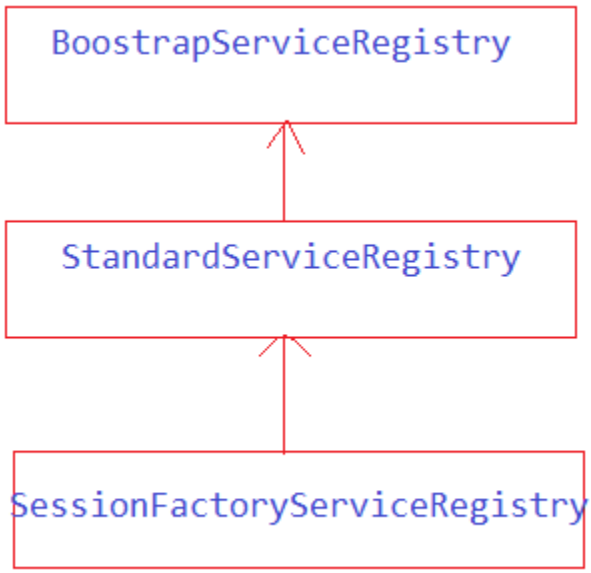
You give your services to ServiceRegistryBuilder ,Configuration to ServiceRegistryBuilder,you give your initiator to ServiceRegistryBuilder ,create the builder with information of those component and ask ServiceRegistryBuilder to build the ServiceRegistry ,My ServiceRegistry will get Instantiated with services which is immutable .(one we create we cannot change Runtime swapping is not possible)

Some one has to create only once ,after creating cannot change if you want to change add it to registry create one more Registry

There are three ServiceRegistry in hibernate

1. BoostrapServiceRegistry
2. StandardServiceRegistry
3. SessionFactoryServiceRegistry

These are hierarchical



BoostrapServiceRegistry

All the mandatory services that are required for your Hibernate to work such kind of mandatory services are managed by BoostrapServiceRegistry that seems to be common for all other services across all the ServiceRegistry

There are four types of ServiceRegistry

1. ClassLoaderService (hibernate will load your entity class using this, it will which class loader will load according to situation

   J2EE classloader service, jdk classloader service, remote classloader service multiple implementation of classloader service containing to the environment)

2. Integrator Service (it use for Integrating Hibernate for other service it not build completely right now no integration is available)
3. SartegySelectorService(local Transaction or global transaction,you need to tell org.Hibernate.tansaction.JDBCTransactionManager or jdbc .it will determine the corresponding class name using short name (SartegySelectorService))
4.

StandardServiceRegistry

ConnectionServiceProvider is one of the service type that is binded with StadardServiceRegistry

JdbcService service type that is binded with StadardServiceRegistry.

Transaction service type that is binded with StadardServiceRegistry.

SessionFactoryServiceRegistry

There are two services that are binded to ServiceRegistry

For every service ServiceRegistry type there is one builder

1. BoostrapServiceRegistryBuilder
2. StandardServiceRegistryBuilder
3. SessionFactoryServiceRegistryBuilder

Through the configuration it will determine which service has to be instantiated

Which service it will create it is default ,Override them with your own services if you don't override it will create with default services if we override then it will create with custom services there with which ServiceRegistry will be created .Use the ServiceRegistry to create SessionFactory ,SessionFactory need what Services ,services are there in ServiceRegistry .

While creating SessionFactory, you create the SessionFactory with configuration, to the configuration give ServiceRegistry, then configuration will populate configuration to the ServiceRegistry, ServiceRegistry have services, with that ServiceRegistry added to SessionFactory, SessionFactory will be created with that configuration

So that my SessionFactory is loaded by ServiceRegistry with services

SessioFactort.openSession();

SessionFactory will goes to the one of these ServiceRegistry and ask the service give me this service ,that gives you the service ,service will be use by SessionFactory to get that connection ,this will be pass to the session and the created the session and return session to you.

```
Configuration configuration=new Configuration().configure();
        StandardServiceRegistryBuilder builder=new
StandardServiceRegistryBuilder();
        builder.applySettings(configuration.getProperties());
/* All configuation that we configure in the xml has been loaded in to
 builder with which builder can parse it and instantiate default
service*/
```

you can also add
builder.addService(servicerole,service);
builder.configure(arg0)
//if you want to add some more configuration you can directly add to the builder also you can custumized the property file names you can complete path of hibernate.properties hibernate-cfg.xml is not required you can give any properties file name this addes in 4.x

```
        StandardServiceRegistry serviceRegistry=builder.build();
```

```
SessionFactory
sessionfactory=configuration.buildSessionFactory(serviceRegistry);
//this registry plugged it into whom SessionFactory that why
SessionFactory act as an Central Component it is centric for whole
hibernate it host an array of services as part of ServiceRegistry
It host the ServiceRegistry from which the hibernate can use services.

//with this service registry create the SessionFactory
         Session session=sessionfactory.openSession();
```

Note-SessionFactory cannot use your Service. SessionFactory goes to service and get service from the ServiceRegistry

The lifecycle of these service is managed by ServiceRegistry dependency of the service is managed by ServiceRegistry either by pulling or pushing

It manages the services if one service depends on another service, injecting a service into another service this will be done by service registry.

if one service required another service ,it is with the ServiceRegistry ,this service goes to the that service give me this service ,pulls and use it.It support either pulling or pushing/injecting.

Such a dependency pull and dependency injection will be supported by ServiceRegistry and managing those services, Starting those service and stopping those services

All those done by ServiceRegistry

How does this services will get populated with ServiceRegistry?

```
package com.fhb.entity;
public class Project {
	int id;
	String title;
	String description;
	String domain;
	String duration;
	String client;
	String status;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public String getDescription() {
		return description;
	}
	public void setDescription(String description) {
		this.description = description;
	}
	public String getDomain() {
		return domain;
```

```
	}
	public void setDomain(String domain) {
		this.domain = domain;
	}
	public String getDuration() {
		return duration;
	}
	public void setDuration(String duration) {
		this.duration = duration;
	}
	public String getClient() {
		return client;
	}
	public void setClient(String client) {
		this.client = client;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
	@Override
	public String toString() {
		return "Project [id=" + id + ", title=" + title + ", description=" +
description + ", domain=" + domain
				+ ", duration=" + duration + ", client=" + client + ",
status=" + status + "]";
	}

}
```

Project-hbm.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="com.fhb.entity.Project" table="project">
     <id name="id" column="id"/>
        <property name="title" column="title" />
        <property name="description" column="description" />
        <property name="domain" column="domain" />
        <property name="duration" column="duration" />
        <property name="client" column="client" />
        <property name="status" column="status" />
     </class>
```

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

Hibernate-cfg-mapping.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">oracle.jdbc.driver.OracleDriv
er</property>
        <property
name="hibernate.connection.password">sahid</property>
        <property
name="hibernate.connection.url">jdbc:oracle:thin:@localhost:1521:xe</p
roperty>
        <property
name="hibernate.connection.username">sahid</property>
        <property
name="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</prope
rty>
        <mapping resource="com/fhb/entity/project.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

```
package com.fhb.test;
import javax.imageio.spi.ServiceRegistry;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import com.fhb.entity.Project;
public class PTest {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Configuration configuration=new Configuration().configure();
```

```
        StandardServiceRegistryBuilder builder=new
StandardServiceRegistryBuilder();
        builder.applySettings(configuration.getProperties());
/* All configuation that we configure in the xml has been loaded in to
 builder with which builder can parse it and instantiate default
service*/
        StandardServiceRegistry serviceRegistry=builder.build();
        SessionFactory
sessionfactory=configuration.buildSessionFactory(serviceRegistry);
        Session session=sessionfactory.openSession();
        Project project=(Project)session.get(Project.class, 1);
        System.out.println(project);
    }
}
```

Custumized

```
package com.fhb.entity;
public class Project {
    int id;
    String title;
    String description;
    String domain;
    String duration;
    String client;
    String status;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public String getDomain() {
        return domain;
```

```
	}
	public void setDomain(String domain) {
		this.domain = domain;
	}
	public String getDuration() {
		return duration;
	}
	public void setDuration(String duration) {
		this.duration = duration;
	}
	public String getClient() {
		return client;
	}
	public void setClient(String client) {
		this.client = client;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
	@Override
	public String toString() {
		return "Project [id=" + id + ", title=" + title + ", description=" +
description + ", domain=" + domain
				+ ", duration=" + duration + ", client=" + client + ",
status=" + status + "]";
	}

}
```

project.hbm.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="com.fhb.entity.Project" table="project">
     <id name="id" column="id"/>
        <property name="title" column="title" />
        <property name="description" column="description" />
        <property name="domain" column="domain" />
        <property name="duration" column="duration" />
        <property name="client" column="client" />
        <property name="status" column="status" />
     </class>
```

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

```
package com.fhp.connectionprovider;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;
import org.hibernate.HibernateException;
import org.hibernate.engine.jdbc.connections.spi.ConnectionProvider;
import org.hibernate.exception.spi.Configurable;
import org.hibernate.service.spi.Startable;
import org.hibernate.service.spi.Stoppable;

public class JdbcConnectionProviderimpl implements
ConnectionProvider,Startable,Stoppable,Configurable {


	@Override
	public boolean isUnwrappableAs(Class arg0) {
		// TODO Auto-generated method stub
		return false;
	}

	@Override
	public <T> T unwrap(Class<T> arg0) {
		// TODO Auto-generated method stub
		return null;
	}

	@Override
	public void configure(Properties arg0) throws HibernateException {
		// TODO Auto-generated method stub

	}

	@Override
	public void stop() {
System.out.println("Stop services");
	}

	@Override
	public void start() {
		System.out.println("Start services");
	}

	@Override
	public void closeConnection(Connection connection) throws SQLException {
		System.out.println("ClosedConnection");
		if(connection!=null)
```

```
		{
			connection.close();
		}

	}

	@Override
	public Connection getConnection() throws SQLException {
		System.out.println("GetConnection");
		try {
			Class.forName("oracle.jdbc.driver.OracleDriver");
		} catch (ClassNotFoundException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sahid",
"sahid");
		return con;

	}

	@Override
	public boolean supportsAggressiveRelease() {
		// TODO Auto-generated method stub
		return false;
	}

}
```

Hibernate-cfg.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">oracle.jdbc.driver.OracleDriver</propert
y>
        <property name="hibernate.connection.password">sahid</property>
        <property
name="hibernate.connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
        <property name="hibernate.connection.username">sahid</property>
        <property
name="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</property>
        <mapping resource="com/fhb/entity/project.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

JdbcConnectionProviderimpl

```
package com.fhb.test;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.engine.jdbc.connections.spi.ConnectionProvider;
import com.fhb.entity.Project;
import com.fhp.connectionprovider.JdbcConnectionProviderimpl;

public class PTest {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Configuration configuration=new Configuration().configure();
		StandardServiceRegistryBuilder builder = new
StandardServiceRegistryBuilder();
		builder.applySettings(configuration.getProperties());
		ConnectionProvider connectionprovider=new
JdbcConnectionProviderimpl();
		builder.addService(ConnectionProvider.class, connectionprovider);
		StandardServiceRegistry serviceRegistry=builder.build();
		SessionFactory
sessionfactory=configuration.buildSessionFactory(serviceRegistry);
		Session session=sessionfactory.openSession();
		Project project=(Project)session.get(Project.class, 1);
		System.out.println(project);


	}

}
```

Annotation

- ✓ From the hibernate 3.5 version hibernate support annotation
- ✓ Annotation are not executable classes rather annotation hold metadata with which hibernate can perform the operation.
- ✓ We used to write the annotation as the part of the source code of our classes that's why it is called source code metadata.

#Q->Why one should use annotation?

Ans-: Annotation are called source code metadata.

- ✓ Earlier we use to configure meta-data (information about something) of our entity class using an external xml file for mapping information which is called mapping file or configuration information we will write in an external xml file.
- ✓

#Q->Why hibernate had added annotation approach rather xml driven configuration approach?

Ans:-There are several limitation or drawbacks are there while working with as the part of xml.

1)

- ✓ XML is verbose(For providing little amount of information classes we wanted to configure metadata write even though it is less the amount of tag we write to surround the metadata will be more huge amount of the data has to be written as the part of the xml file
- ✓ This information has to be written in proper order the names of the tags are case sensitive ,the order of the tag it will show it is mandatory ,this tag are lengthy One can't remember the tag name and the order in which they are ,it is very difficult to working with xml providing metadata.
- ✓ We oftenly forget writing the metadata which we incorrectly write that we will run in to a problem.

2)

- ✓ XML is an different language itself it not related to java language
- ✓ Non-of the java programmer aware about xml ,to provide configuration about entity class in an xml file the java developer should learn xml ,the syntaxes and semantics representation of xml which all together is new for java programmer and they have to re-learn again to work with xml that's why programmer hate to learn xml.

3)

- ✓ XML is file in which we are going to store the data
- ✓ XML is non-compliable ,there is no intermediate compliable process ,when it comes to java there is an intermediary compilation process one has compile the source code to validate whether code is valid or not.

When it comes XML there is no intermediary compilation process so you validate the source code whether it is valid or not.

- ✓ When it is invalid after deployment during execution we will run in to logical error again we need to modify the configuration and we need to re-compile, re-package, and re-deploy restart and access.

An alternate to an XML is brought into picture is called annotation Sun-Microsystem

#Q-.What are the benefit for going annotation instead of XML?

- ✓ Annotation are exactly replacement for xml
- ✓ Annotation are also written in java classes are by themself those are java classes theses exactly reflect the same syntaxes and semantics of java so always the java developer fill comfortable writing in such metadata configuration information in the annotation rather than xml
- ✓ As annotation is part of the source code when code of your application when we compile the source code it compiled and configuration error you have made as the part of annotation will be identified we need not to validate till run time there is an intermediary compilation process possible when it comes to
- ✓ Annotation are not span more than one words it is easy to remember for the programmer to remember and provide information

JPA(java persistence API) : first java does not has support for ORM technology, but
rest of the all programming languages shown there interest to the ORM technology,

so people started losing the interest on java. Because of that java has provided one API
Called JPA (java persistence api). Which has the support for ORM technology.
– JPA –ri(implementation): JPA-RI is the implementation provided by java people to the
Other people to add the support for jpa.

JPA ri,JPA hibernate implementation is there, as the part of JPA api 2.0 ,JPA has provide there annotation

#Q->Why are you using JPA annotation?

✓ Hibernate annotation (core annotation)
✓ JPA annotation

In Hibernate when we are using Hibernate api we can either use JPA annotation or Hibernate annotation

(a) Because hibernate has up taken the JPA annotation In hibernate api itself all the Configuration , SessionFactory, Session to support JPA annotation directly

There are three ways of working with Hibernate

1) Hibernate api hibernate core annotation
2) Hibernate api JPA api annotation
3) JPA api with JPA api annotation

#Q->How completely avoid the XMl file in our application?

Ans write the annotation for the entity classes for the Go Programmatic approach for completely avoid to write XML file as part of application.

When it comes to annotation in Hibernate it groups in to two types

✓ Logical annotation
✓ Physical annotation

Logical annotation

Logical mapping annotation are the annotation that are talks about classes and relation between my entity classes and another entity classes they don't talk about entity classes and table

Ex: @Entity, @ID, an annotation talks about logical thing it doesn't has any physical
Existence called logical annotation.

Physical annotation

These annotation will map the classes with physical schema to which we want to store the data against in.

Ex: @Table(name="Event"), @Column(name="Event_Id"), these two annotation talks
About the physical existence, means table is present into the database and columns also
Present into the table.

Annotation

@Entity-It will make class as a entity class to able persist ate.
@Table-To specify the name of the table

```
@Table(name = "companyDetail")
```

@Id- It is the attribute that is representing primary key column name of my table.

```
@Id
@Column(name = "id")
```

When we are not specifying column name by default it will take attribute name as column name.To Specify the column name

@column(name"PROJECT_NO")

@column(name=" PROJECT_NO", nullable=false, unique=true, length=512)

Various other data you can configured using database metadata.

@Column-To specify the column name of the table

When both table column name entity class attribute name is same no need to specify column name.

- Every attribute has to map using one annotation otherwise it will not where it is persisted attribute or not a persisted attribute.
- When we are not writing it will behave like a transient attribute(it will not get store or will not get accessed)
- By default for every attribute of your class the hibernate will add one default annotation called **@Basic.**
- **@Basic has only one attribute "fetch" mode whether eager or lazy, By default all the attribute is eager Only primitive type attribute we can write @Basic but it cannot written on Object attribute in our class.**
-

String	**columnDefinition** (Optional) The SQL fragment that is used when generating the DDL for the column.
boolean	**insertable** (Optional) Whether the column is included in SQL INSERT statements generated by the persistence provider.
int	**length** (Optional) The column length.
String	**name** (Optional) The name of the column.
boolean	**nullable** (Optional) Whether the database column is nullable.
int	**precision** (Optional) The precision for a decimal (exact numeric) column.
Int	**scale** (Optional) The scale for a decimal (exact numeric) column.

String	**table** (Optional) The name of the table that contains the column.
Boolean	**unique** (Optional) Whether the property is a unique key.
Boolean	**updatable** (Optional) Whether the column is included in SQL UPDATE statements generated by the persistence provider.

Q-What are the basic set of annotation that I need to write for an Entity class for get persisted?

And-Minimum annotation that we need to write to make a entity class persisted is "@Entity @Id".

Some of the Annotations in hibernate:

- @Entity:(To make our class pertistence and pojo)
- @Table(name="tableName"): (map corresponding table to the pojo class)
- @Id: (to specify primary column attribute)
- @Basic (assume attribute name as column name)
- @Column(name="columnName", size=" XXX" sqlType="XXX") : To map with specific column with different column name)
- @Unique : (To apply unique constraints to the column)
- @UniqueConstraint() : (if we want to give unique constraints more than one column we can use this annotation)
- @Basic(fetch=fetchType.Lazy) : (we can make a particular column as lazy loading column by specifying this annotation.
- @Transient : if we don't want to stored column into table so we can use this annotation, actually for reference purpose we can use.

We can write annotation

1. attribute-level
2. method-level

We can write the annotation at attribute-level and method-level at a time but hibernate will choose only one it depends on which level you write @Id and rest of the part completely ignored.
One of the way we can use only , we cant use both the way to configure our class.

• If we use both then it will take one of them, but it will not take both.
• Mostly Attribute level annotations are used.
➔If you write @Id in attribute level hibernate will not use getXxx and setXxx to access the data and set the data it directly access the attribute of the class even those are private using reflection.
➔If we wrote at method level ,it will not directly access attribute It always call setXxx and getXxx method only to store the data .
Depends on where you wrote annotation hibernate will use the data of the entity object either using directly attribute using reflection or using accessor method(setXxx and getXxx).

Q-Why should we go for method level annotation?
And-:Because Inheritance,we can work on oop principle.

```
interface A {
	@Id
	int getA();
}
class AImpl implements A {
	@Override
	public int getA() {
		// TODO Auto-generated method stub
		return 0;
	}
}
class BImpl implements A {
	@Override
	public int getA() {
		// TODO Auto-generated method stub
		return 0;
	}
}
```

In both the class I need not to re-annotate. Interface have only methods so that interface method can be annotated and all the implementation can use same meta-data rather re-writing same annotation all the classes that's benefit writing in method level. Rather than using at attribute level.

Q-I want one of the attribute non-persistable what should I do?
Ans-Mark the attribute as transient @Transient hibernate will not manage accessing and storing the data from the database.

Annotation Approach

```java
package com.hb.entities;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;
@Entity
@Table(name = "TASK", uniqueConstraints = { @UniqueConstraint(columnNames = {
            "taskId", "assignTo" }) })
public class Task {
      @Id
      private int taskId;
      @Column(unique = true)
      private String title;
      private String description;
      private int duration;
      private String assignTo;
      private String status;

      public int getTaskId() {
            return taskId;
      }

      public String getTitle() {
            return title;
      }

      public String getDescription() {
            return description;
      }

      public int getDuration() {
            return duration;
      }

      public String getAssignTo() {
            return assignTo;
      }

      public String getStatus() {
            return status;
      }

      public void setTaskId(int taskId) {
            this.taskId = taskId;
      }
```

```
	public void setTitle(String title) {
		this.title = title;
	}

	public void setDescription(String description) {
		this.description = description;
	}

	public void setDuration(int duration) {
		this.duration = duration;
	}

	public void setAssignTo(String assignTo) {
		this.assignTo = assignTo;
	}

	public void setStatus(String status) {
		this.status = status;
	}

	@Override
	public String toString() {
		return "Task [taskId=" + taskId + ", title=" + title + ",
description="
				+ description + ", duration=" + duration + ", assignTo="
				+ assignTo + ", status=" + status + "]";
	}

}


**********************************************************************
package com.hb.util;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class HibernateSessionFactory {
	private static SessionFactory sessionfactory=null;
	static
	{
		Configuration configutation=new Configuration().configure();
		StandardServiceRegistryBuilder builder=new
StandardServiceRegistryBuilder();
		builder.applySettings(configutation.getProperties());
		StandardServiceRegistry registry=builder.build();
		sessionfactory=configutation.buildSessionFactory(registry);
```

```
	}
	public static SessionFactory getSessionfactory() {
		return sessionfactory;
	}
	public static void closeSessionfactory() {
		if(sessionfactory!=null)
			sessionfactory.close();
	}

}
```

```
package com.hb.test;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.hb.entities.Task;
import com.hb.util.HibernateSessionFactory;
public class Test {
	public static void main(String[] args) {
		Task task=null;
		Session session=null;
		Transaction transaction=null;
		boolean flag=false;
		session=HibernateSessionFactory.getSessionfactory().openSession();
		task=(Task) session.get(Task.class, 1010);
		System.out.println(task);

		HibernateSessionFactory.closeSessionfactory();
	}
}
```

Hibernate-cfg.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">oracle.jdbc.driver.OracleDriver</propert
y>
        <property name="hibernate.connection.password">sahid</property>
        <property
name="hibernate.connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
        <property name="hibernate.connection.username">sahid</property>
        <property
name="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</property>
        <mapping class="com.hb.entities.Task"/>
    </session-factory>
</hibernate-configuration>
```

get() Vs load()

HibernateTools

Hibernate provide rapid application development support in developing the application while working with Hibernate.

It provide bunch of for tools that automate the various aspect of your application development.

These tools are broadly classified into three groups

1. Graphical user interface
2. Command-line tools(Alternate for this is Ant tools)
3. Programmatic or Configuration tools

CommandLineTools

There are 3 tools in Command-line tools

a. SchemaExport tool
b. SchemaUpdate tool
c. SchemaValidator tool

SchemaExport tool

It helps in exporting table scripts for creating the data-base table for mapping what we have written for our application in to the database.

Rather we will create it manually we can take help of SchemaExport tools where it uses the metadata about the database that is provided as the part of the mapping information it quickly go create the table with the database that's why SchemaExport tool will be use.

Q-Where to use SchemaExport tools?

During development environment people will use SchemaExport tool For creating the table rather creating it manually they use SchemaExport tool generating sql script for creating table in database using relevant mapping file as they are written as there part of the application logic to come-up with table.

Q-What is SchemaUpdate tool and what is the purpose of it?

When we have already created data-base as part of our application During the time of development I might do some changes in existing code and additional table ,addition column are being added and that relevant changes has to be made in database.

We need use alter script for these changes

We use SchemaUpdate tool ,it will check the delta and will generate the script for making such changes in the database which can be use for the verifying and capturing the changes in actual sql scripts to load in the production environment that's Schema update Tools will be use.

Q-When does the SchemaValidator tool will be use?

Before deploying the application in production environment I want to check the code which I'm deploying it always in-sink and in-tag with the database table that are being created within the production

The table are not created from mapping information, the tables are being created manually by the developer

The state of the table that I have created and the current code that is being deployed are in sink or not I want always to verify .To ensure that application will work properly that's why SchemaValidator tool will be use .

To deploy application we rely on build tools, which is Ant tool
Ant will configure validator tool.in ant build script.
We will run ant it will takes the code from Repository, it compiles, and packages and it copies into the application server deployment directory, stats the server all the automation done by the ant tool
In this they will add one more step ,that is compile the code, once it compile then validate with schema ,SchemaValidator tool will be configured in ant which will validate the code with the database Schema.

Programmatic or Configuration tools

IdentityGenrator

Id generators are the classes that generates the unique keys
For serogate-key column that are there in the database table
Hibernate has provide IDgenarator for generating these keys

- ID generator / key generator / surrogate key /identity generator these are name for hibernate ID generators.
- Hibernate is the ORM Tech. at each and every place he has provide the flexibility to avoid the boiler plat logic.
- Here also hibernate has provided id generator to avoid the duplicate the logic and it will make our application portable from on DB to another DB.
- But when we deal with RDBMS and we can not migrate from oracle to Mysql, b'z both DB has there own syntaxes.
- Every ID generator has there own implemented classes which is internally called by session object.
- Assigned, identity, native, sequence are the short name of the id generators.
- Even we can create our own id generator by overriding the internal class/interfaces.
- We can call custom id generator.
- Every id generator having there own strategy to generate a primary key value.
- Every id generator different from each other and there implementations also.
- Actually session never generate the primary key for table, it will take the metadata and pass to the corresponding class to generate the primary key value.
- A class going to generate the primary key value and return to
- the session object and session will add to the cache

There are 11 id generators are there

1. Assign
2. Increment
3. Identity
4. Sequence
5. Native

6. Guid
7. Uuid
8. Hilo
9. Sequencehilo
10. select
11. Forien

Assign

- Assign is called default id generator
- It is default id generator
- It will not do any thing rather it himself indicate that the user is going to manually assign the value for primary key.it should generate any primary key.By default if we do not mention any Identity generator by default it will take as Assign Id generator

How does it work?

- Hibernate will not generate any surrogate key for assigned ID generator, programmer has to provide the primary key value to the hibernate.
- Session object will take the provided class name it will go to the sessionFactory , it will get the metadata for corresponding class from mapping information, it look to id generator, if it is ASSIGNED Id generator then it will take programmer provided value and computed into the cache.

Data- type :

int , short , long

Benefit :

It is database independent

Syntax:

```
<ID name="att_name" column="col_name">
<generator class="assigned"/>
```

Increment

- It calculate the max primary key value from the database always increment by with '1' for generating the key.
- It work for all the databases
- It work for following data type int,short,long

- It not recommended to use in all the environment
- It is recommended use in non-cluster environment.
- How does it work? :=> MAX(ID)+1;
- It will going to fire query on the DB table
- Ex: select max(AGENT_ID) from AGENT;
- Session object will take the provided class name, it will go to the sessionFactory and it will get the metadata for corresponding class from mapping information, it look to id generator, if it is INCREMENT Id generator then it will generate one query to get the max value from the database and it will add +1.
- And it will give to the session object, session object going to perform rest of the operation.

• Data-type : int , short , long.

Benefits:

- It is database independent
- It will work with any database.

• Drawback:

- It will not generate unique primary key in multi-threaded environment or cluster environment.
- If two or more classes wanted to persist the data at a time then it may chance to generate duplicate primary key.

• Syntax:

<ID name="att_name" column="col_name">
<generator class="increment"/>

##

```
package com.idg.entities;
import java.util.Date;
public class Appointment {
	public int appointmentno;
	public String dooctorname;
	public Date appointmentdate;
	public String typeofdisease;
	private String patientName;
	private String comment;
	private String status;
	public int getAppointmentno() {
		return appointmentno;
	}
	public void setAppointmentno(int appointmentno) {
		this.appointmentno = appointmentno;
/*Setters and getters .....*/
}
```

****

```
<hibernate-mapping>
    <class name="com.idg.entities.Appointment" table="APPOINTMENT">
        <id name="appointmentno" type="int">
            <column name="APPOINTMENTNO" />
            <generator class="increment" />
        </id>
……</hibernate-mapping>
```

mySqlTable

APPOINTMENTNO	DOOCTORNAME	APPOINTMENTDATE	TYPEOFDISEASE	PATIENTNAME	COMMENT	STATUS
1	Swamy Naidu	2017-02-07 00:04:14	Cold	Sahid	Good	RAC
2	Dr Batra	2017-02-07 00:05:29	Skin	Dhananjaya	Good	CNF
3	Dr Sukla	2017-02-07 00:06:54	Eye	Avinash	Good	WL

Identity

- There are some databases that are support Auto increment as an column type or as primary-key column type for generating primary-key value .Automatically at database level.(postgres sql,mysql,db2,mssqlserver,sap)
- Hibernate will not generate id rather while storing the data ,the database it-self will persist the data by generating the column key you just go and persist the id that has generated by the database,that's why specify 'Identity'.
- After storing the data it goes to the database table queries the primary key vale and return to us

How does it work?

- Identity one of the id generator which relay on the database auto-generated elements.
- Here hibernate will not do any thing, session object will play the role for getting the primary key value from the corresponding database.
- Session will get the metadata from the mapping file and it will go to the specific DB and will run the auto-generate column which is available into DB.
- Here programmer and hibernate will not have any work everything is takes care by session, and it will automatically generate the primary key values.

• Data-type: int , short , long.
• Benefits:
  - It will work with auto- sequence database.
• Drawback:
  - It only work with auto-generated value databases.
• Ex: mysql, ms-sql server.

• What is the difference between assigned and identity?b'z both the places hibernate will not do anything ?

- Assigned : here programmer is the responsible for providing the primary key value to the session to persist into the database.
- Identity: hare identity specific class will automatically execute the corresponding database sequence and it will generate the primary key value.
- Both the places hibernate will not do anything.

• Syntax:

<ID name="att_name" column="col_name">
<generator class="identity"/>

```
package com.idg.entities;
import java.util.Date;
public class Appointment {
	public int appointmentno;
	public String dooctorname;
	public Date appointmentdate;
	public String typeofdisease;
	private String patientName;
	private String comment;
	private String status;

	public int getAppointmentno() {
		return appointmentno;
	}

	public void setAppointmentno(int appointmentno) {
		this.appointmentno = appointmentno;
	}
/*Setters and getters .....*/
```

```
<hibernate-mapping>
    <class name="com.idg.entities.Appointment" table="APPOINTMENT">
        <id name="appointmentno" type="int">
            <column name="APPOINTMENTNO" />
            <generator class="identity" />
        </id>
…….</hibernate-mapping>
```

mySqlTable

APPOINTMENTNO	DOOCTORNAME	APPOINTMENTDATE	TYPEOFDISEASE	PATIENTNAME	COMMENT	STATUS
1	Swamy Naidu	2017-02-07 00:04:14	Cold	Sahid	Good	RAC
2	Dr Batra	2017-02-07 00:05:29	Skin	Dhananjaya	Good	CNF
3	Dr Sukla	2017-02-07 00:06:54	Eye	Avinash	Good	WL
4	Dr Sukla	2017-02-07 00:07:00	Eye	Avinash	Good	WL
5	Dr Sukla	2017-02-07 00:22:27	Eye	Avinash	Good	WL

For Oracle it will show

```
ERROR: HHH000299: Could not complete schema update
  org.hibernate.MappingException:
  org.hibernate.dialect.Oracle10gDialect does not support identity
  key generation
    at
  org.hibernate.dialect.Dialect.getIdentityColumnString(Dialect.java
  :834)
    at
  org.hibernate.dialect.Dialect.getIdentityColumnString(Dialect.java
  :824)

  Hibernate: insert into APPOINTMENT (DOOCTORNAME, APPOINTMENTDATE,
  TYPEOFDISEASE, PATIENTNAME, COMMENT, STATUS) values (?, ?, ?, ?,
  ?, ?)
```

Sequence

- Sequence is something that supported by oracle
- Sequence is a kind of datastucture that generate unique key of value one after the another one when-ever we access the sequence
- In order to generate the key we create one sequence get value from the sequence with which we store the data into the database.
- Whenever hibernate trying to storing the data session will goes to mapping file and check for the generator ,and it found sequence and it immediately ask the sequence generator to go the database query the sequence for the next value and returns to the session .session will update that into the entity object and passes it in to database for storing.
- If we not provide any sequence it will create a sequence HibernateSequnce.it will be generated by the hibernate.
- Why we should not use default sequence?

  Several table will be use by the same sequence ,when we use sequence generator which will not generate sequenctially generate the id of the table.
- How provide our own sequence ?

How does it work?

- Sequence id generator relay on the specific DBs which having the sequences concept.

- ❖ Session will get the metadata from the mapping file and it will go to the specific DB and will run the sequence which is available into DB.

Data-types: int , short, long.

Benefits:

Drawback:

- ✓ It will work with only those DBS which are having the concept called sequence.
- ✓ If we are using sequence and if we are not specifying any name to the sequence the hibernate going to give the default name to the sequence.
- ✓ But if there are more then one entities are there and we haven't specified sequence name then it lead to generate incorrect primary key value .

• Syntax:

- ✓ <generator class="sequence">

<param name="sequence" > agent_ID</param>
</generator>

- ✓ <generator class="sequence">

<param name="sequence" > property_ID</param>
</generator>

• Ex: create sequence hibernate_sequence

• Ex: select hibernate_sequence.nextval from dual

****

```
package com.flc.entities;
public class Product {
	private int productid;
	private String productname;
	private String merchantname;
	private String description;
	private String stock;
	private float price;
	public int getProductid() {
		return productid;
	}
	public void setProductid(int productid) {
		this.productid = productid;
	}
	public String getProductname() {
		return productname;
	}
	/*setter and getters*/
```

```
<hibernate-mapping package="com.flc.entities">
<class name="Product" table="PRODUCT">
```

```
<id name="productid" column="product_id">
<generator class="native">
<param name="sequence">PRODUCTNO</param>
</generator>
  </id>
  ….</class>
  <hibernate-mapping>
```

PRODUCT_ID	PRODUCTNAME	MERCHANTNAME	DESCRIPTION	STOCK	PRICE
9797	Phobia	phillips	32 inch UHD led corved tv	notavailable	134000
9798	Bravia	sony	54 inch 4k led corved tv	available	156000
9799	json	sansui	53 inch 4k led corved tv	available	146000
9800	blizard	samsung	56 inch 4k led corved tv	available	198000

Native

- As per the database
- if our application working with oracle databases the native generator with configure sequence generator.
- If we are working any of the database that working with auto increment column that will work as Identity generator.
- If we configure 'Native' as identity generator it will be portable across all database those who are supporting sequence or Identity as well(auto increment).
- Syntax:

```
<id name="productid" column="product_id">
  <generator class="native">
  <param name="sequence">PRODUCTNO</param>//if it sequence the it will work for the oracle datatbse else for the mysqll//
  </generator>
```

Guid

- It stands for (Global unique id)
- It generate unique set of character which of string or hexadecimal nature
- It will generate by the database not by the hibernate
- It speciality is ,no two times guid will not be generated across the world. Finally we are able generate the which is not human readable.
- No chance the some one guess the key ,you never can access the data using id. Sometime it is more crucial that's why guid will helps

- It work with database which non-integer type which is varchar or String
- It only work with the database that support guid.
- How does it works?
  - ✓ Actually session object will take the class name with that it will go to the sessionFactory and get the metadata for corresponding class with table and it will check which id generator has been used for generating the primary key.
  - ✓ Actually GUID work with JEE environment, and it will uses one algorithm which going to generator primary key value.
  - ✓ GUID will take number of parameter to generator the primary key value. It will generator the value in String format with max length is 256 character.
  - ✓ It will unique across the universe.

Ex: select sys_guid() from dual;

- Benefits:

  We will get unique id in the universe.
- Data-types :

  string
- Drawbacks:
  - ✓ It will work with GUID generator databases only.
  - ✓ It will consume more memory space for storing the primary key value.
  - ✓ It is very hard to remember and query the data.
  - ✓ It is used in special environment only.

UUID

- It Stands for universal unique id
- It is supported by hibernate and it not related to database.
- It Similar guid ,uuid is also unique set of character which of string or hexadecimal nature
- It use several parameter and algorithm to genaraye id may be the system-id,ip-address, location at which we are A lot of thing uses to generate the id.
- It works with all the databases. It has no restriction because it is not relying on database capability.
- E.g whe we don't want to have id to use by enduser ,the people are going use uuid at majorit of the time.

Appointment.hbm.xml

```java
package com.idg.entities;

import java.util.Date;

public class Appointment {
	public String appointmentno;
	public String dooctorname;
	public Date appointmentdate;
	public String typeofdisease;
	private String patientName;
	private String comment;
	private String status;

	public String getAppointmentno() {
		return appointmentno;
	}

	public void setAppointmentno(String appointmentno) {
		this.appointmentno = appointmentno;
	}

	public String getDooctorname() {
		return dooctorname;
	}

	public void setDooctorname(String dooctorname) {
		this.dooctorname = dooctorname;
	}

	/*Setter and getters .......*/
```

```xml
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 2 Feb, 2017 2:46:32 PM by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping>
    <class name="com.idg.entities.Appointment" table="APPOINTMENT">
        <id name="appointmentno" type="java.lang.String">
            <column name="APPOINTMENTNO" />
            <generator class="uuid" />
            </id>
        <property name="dooctorname" type="java.lang.String">
            <column name="DOOCTORNAME" />
        </property>
        <property name="appointmentdate" type="java.util.Date">
            <column name="APPOINTMENTDATE" />
        </property>
        <property name="typeofdisease" type="java.lang.String">
            <column name="TYPEOFDISEASE" />
        </property>
    </class></hibernate-mapping>
```

```
SQL query
Hibernate: insert into APPOINTMENT (DOOCTORNAME, APPOINTMENTDATE, TYPEOFDISEASE,
PATIENTNAME, COMMENT, STATUS, APPOINTMENTNO) values (?, ?, ?, ?, ?, ?, ?)
```

Table

APPOINTMENTNO	DOOCTORNAME	APPOINTMENTDATE	TYPEOFDISEASE	PATIENTNAME	COMMENT	STATUS
297e5c875a148c0b015a148c13110000	sriman	2017-02-06 23:20:08	coug	sahid	Good	waiting
297e5c875a148df1015a148df80f0000	Durga	2017-02-06 23:22:12	Fever	Lala	Good	confirm
297e5c875a14942a015a14942f970000	Swamy Naidu	2017-02-06 23:32:01	Cold	Sahid	Good	RAC

mysqlhibernate.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!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="connection.driver_class">com.mysql.jdbc.Driver</property>
      <property name="connection.url">jdbc:mysql://localhost:3306/sahid</property>
      <property name="connection.username">root</property>
      <property name="connection.password">root</property>
      <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
      <property name="hibernate.hbm2ddl.auto">update</property>
      <property name="hibernate.show_sql">true</property>
      <mapping resource="com/idg/entities/Appointment.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

HibernateSessionFactory

```
package com.idg.util;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class HibernateSessionFactory {
      public static SessionFactory instance;
      static {
            Configuration configuration = new Configuration()
                        .configure("mysqlhibernate.hbm.xml");
            StandardServiceRegistryBuilder builder = new
StandardServiceRegistryBuilder();
            builder.applySettings(configuration.getProperties());
            /*
             * serviceRegistry All configuration that we configure in the xml has
             * been loaded in to builder with which builder can parse it and
             * instantiate default service
             */
            StandardServiceRegistry serviceRegistry = builder.build();
            instance = configuration.buildSessionFactory(serviceRegistry);
```

```
	}

	public static SessionFactory getInstance() {
		return instance;
	}

}
```

Test.java

```
package com.idg.test;
import java.util.Date;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.idg.entities.Appointment;
import com.idg.util.HibernateSessionFactory;

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

		SessionFactory sessionfactory = null;
		Session session = null;
		Transaction transaction = null;
		Appointment appointment = null;
		boolean flag = false;
		sessionfactory = HibernateSessionFactory.getInstance();

		try {
			session = sessionfactory.openSession();
			transaction = session.beginTransaction();
			appointment = new Appointment();
			appointment.setDooctorname("Swamy Naidu");
			appointment.setTypeofdisease("Cold");
			appointment.setAppointmentdate(new Date());
			appointment.setPatientName("Sahid");
			appointment.setStatus("RAC");
			appointment.setComment("Good");
			String uuid = (String) session.save(appointment);
			System.out.println(uuid);

			flag = true;

		} finally {
			if (session != null) {
				if (flag == true) {
					transaction.commit();
				} else {
					transaction.rollback();
				}
			}
		}
	}
}
```

Hilo generator

- Hilo is an generator that will work on based on formula Max_low*next_hi+next_hi
- “Max_low” is configured as a parameter to the hilo generator
- “Next_hi” has to pass an input from database table
- The default table name is HibernateUniqueKey so.It going to create the column name as “next_hi” it reads the next value from the table by default and tries to increment after reading and immediately calculating by substituting that formula.
  Max_low is batch size

What is use of hilo generator?

- If we want store the bunch of data into the database if we use an “increment” or “sequence” or “Identity”
- Every record that you persist into the database it do a round trip
- To the database in computing the id that kill the performance instead of if we ‘hilo’ within the batch it goes once to the database generate the primary key for the batch of the records and avoiding repeated access to the database to compute the primary key.
- The performance will be high when we go with hilo,it works with int, long,short data type and databases all the databases
  - Under any environment we can use ,you must an have to an jta transaction ,without jta transaction it is not recommended to use hilo.

Benefits:

- It will work with any application environment.
- And it will not around trips to the DB maximum time.
- It will use internally locking system while working, b’z in jee application more possibilities are there more then one person can insert the records into the DB, making sure every no one object will get the same hilo value.

Drawbacks:

- It will work with JEE apllication only.

How Hilo internally work

- Hilo work on one of the formula (Max_lo * next_hi + next_hi)
- Hilo is the one to the mathematical algorithm which work with batch of records.

- ✓ Hibernate going to generate one default table with the name(hibernate_unique_key) with one column and column name is next_hi.
- ✓ If we don't want to work with the default generated DB table or column then we can configure our own table name and column.
- ✓ Hilo works with the batch of records.
- ✓ Already other ID generators are there then what is the need for going to word Hilo id generators

❖ Actually every id generator have there own drawbacks while generating the id of the table.

❖ They have to do maximum around trips while generating the primary value.

❖ Hilo will not make maximum around trips with the database for generating the primary key value.

❖ Actually session object will take the class name with that it will go to the sessionFactory and get the metadata for corresponding class with table and it will check which id generator has been used for generating the primary key.

❖ If it is hilo it will check any table and column has been configured or not along with max_lo value.

❖ If it is already configured it will use existing table and column to generate the primary key value, if table and column not configured then hibernate will generate its own table with the name hibernate_unique_key and the column name next_hi.

❖ Max_lo is the value which is talk about how much records inserted at one short. E.g. if we provide 10 means it will insert 10 records.

Internally hilo work as fallows

• Ex <generator class="hilo">
<param name="max_lo">10</param>
[optional]
<param name="table">tbl_name</param>
<param name="column">col_name</param>
</generator>

❖ Now hibernate will generator one table with name hibernate_unique_key with the column name next_hi and it will assign the value as 1 .(next_hi = 1).

❖ We already configured max_lo with the value 10.

❖ Hibernate will compute the formula as below

(max_lo *next_hi * next_hi)
10 * 1 * 1 = 11

- While taking the next_hi from the DB it will increase the next_hi with 1, means next time if other session object want to insert the batch of record, then it will get 10 * 2 * 2 = 22 and they will insert the records.
- No one session object will get duplicate next_hi value.
- It will start inserting the records with id 11, 12 , 13…20. means it will not around trips on the DB it will compute first time and for next 10 records it will get it from hibernate only.
- After inserted 10 records it will go to the DB and get the current next_hi and increments by 1. and perform the tansaction.

Sequence_hilo

- It is exactly work as hilo
- It order to fetch next_hi value the hilo generator goes to the database table but instead of it if you use sequence_hilo the next high value with be fetch from the sequence so that we can avod generating duplicate id in non-jta environment.
- We can use sequence_hilo in any environment and it works only within the specific databases that support sequence
- Seq_hilo same as the hilo id generator, hilo id generator relay on the formula but seq_hilo id generator work with the sequence generator.
- While inserting the data it will generate one of the sequence in to the database.
- It is also used to insert the batch of records, only the difference is, we will get primary value by executing the seq_hilo.
- It will also perform same operation like hilo only.

– How does it works?

- Same as hilo but it will get the value from the sequence.

Benefits:

- It will work with any application environment.

Data-types: int , short , long.

Drawback:

- It will work with oracle database only.
- It will support all the application environment.

Syntax:

```
<generator class=“seq_hilo”>
<param name=“max_lo”>10</param>
<param name=“sequence”>agent_id_seq</param>
```

</generator>
Forien

Select

- It goes to the database select id value return to you
- Select will not generate the primary key ,still the primary will be generated by some logic at the database level ,it goes to the database and fetch the primary key that has been generated.
- While working with legacy application ,in that case the primay key is generated by the procedure while storing the data.
- The key is generated by the database not by the hibernate
- There is some special formula or special logic which is computing the key and storing it into the database.
- In such case when we are working with databse and persisting the data in databse, an object of the data has been persisted by the me.
- Hibernate always after storing the data fetches the primary-key.
- How do I get the primary key that been generated by the legacy procedur written in legacy pl/sql.
- That's why select is id generator it will use. Select.

JPA annotation

API Method

Save()

After persisting the data into the databae within an ID ,the iD which the data has been stored into the data-base it will return the corresponding id
It will return the id at which the corresponding entity Object has been inseted.

- All the other ID generator Except Assign Id generator we can go with save();
- When we use the save() after storing the data into the database with id being generated .the id Immediately will be update the entity object
- After persisting the entity object with id being generated then it immediately update the id in the attribute in the entity object.
- Not only we can use the get as return value but also we can get from entity.

- When you an active transaction to call an save method.
- "We need to call save or persist on the same entity object it has been persisted will results in an exception may not be allow by any of these two method."

Persist()

When it comes to the persist it will not return any Id value that has been inserted into the database to the hibernate because it return type is void.

- While working with assign Id generator as we populate the Id to entity object we don't need the Id as a return value.
- When it comes to persist the entity object which it wlll store in to the database with id it may or may not reflect the id to the entity object.
- Don't rely on persist to reflect the ID (implementation vendor may chose to populate the ID or may not chose to populate the id)
- It may or may not happen when it comes to the persist().
- When it comes to the persist method with transaction without transaction it can persist the data may not save the data into the database unless we call the flush method.

```
package com.dml.test;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.dml.entities.Poilcy;
import com.flc.util.HibernateSessionFactory;
public class Test {

	public static void main(String[] args) {
		SessionFactory sfactory = null;
		Session session = null;
		Poilcy policy = null;
		Transaction transaction = null;
		boolean flag = false;

		try {
			sfactory = HibernateSessionFactory.getInstance();
			session = sfactory.openSession();
			policy = new Poilcy();
			policy.setPolicyHolderName("Amitab Kumar");
			policy.setPolicyname("jeevan Saral");
			policy.setPolicyno(1010101);
			policy.setPremium(99090);
```

```
                policy.setTenure(12);
                session.persist(policy);

                flag = true;
            } finally {

                session.flush();
            }
      }
}//rest of classes and mapping file are common as previous example has explianed
```

SaveOrUpdate()

- It will work based on a attribute when we configured in a mappingfile
- Unsavevalue=
- If the value will match with “Unsavevalue” the attribute then will insert the data into the database
- If it will not-match then it will directly update the data into database.
- It will not query from database for checking that the corresponding Id data there or not it will direct update the data if corresponding id data will not available with database it throws an Exception

Exception in thread "main" org.hibernate.StaleStateException: Batch update returned unexpected row count from update [0]; actual row count: 0; expected: 1

```
package com.dml.test;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.dml.entities.Poilcy;
import com.flc.util.HibernateSessionFactory;
public class Test {
      public static void main(String[] args) {
            SessionFactory sfactory=null;
            Session session=null;
            Poilcy policy=null;
            Transaction transaction=null;
            boolean flag=false;
            try{
                  sfactory=HibernateSessionFactory.getInstance();
                  session=sfactory.openSession();
```

```
                policy =new Poilcy();
                policy.setPolicyHolderName("sachin");
                policy.setPolicyname("freedom");
                policy.setPolicyno(10);
                policy.setPremium(99090);
                policy.setTenure(12);
                transaction=session.beginTransaction();
                session.saveOrUpdate(policy);
          flag=true;
          }finally
          {
                if(session!=null)
                {
                      if(flag==true)
                      {
                            transaction.commit();
                      }
                      else{
                      transaction.rollback();
                      }
                }
          }
     }
}
```

```
<hibernate-mapping>
    <class name="com.dml.entities.Poilcy" table="POILCY">
        <id name="policyno" type="int" unsaved-value="1">
            <column name="POLICYNO" />
            <generator class="assigned" />
        </id>
        .....
</hibernate-mapping>
```

```
package com.dml.entities;

import java.io.Serializable;

public class Poilcy implements Serializable{
private int policyno;
private String policyname;
private String policyHolderName;
private int tenure;
private float premium;
public int getPolicyno() {
     return policyno;
}
public void setPolicyno(int policyno) {
```

```
	this.policyno = policyno;
}
public String getPolicyname() {
	return policyname;
}
public void setPolicyname(String policyname) {
	this.policyname = policyname;
}
//setters and getters…
```

update()

when ever we pass the entity object to the update() method with the modified data ,so that hibernate takes care of updating the record ,that we have populate the data as the part of entity object.
It will execute an update query

It will first query the record from the database and match with current modified data if it matching if there is no change then it will not generate update query
Else if it will not match the it will generate the update query ,and update the corresponding databae.
There are two ways using update()

First fetch entity object and modified the object then call update method;

Note:-
When we call session.save() then it will not go directly to the database the current state of the object will store in the session level cahce

```
package com.dml.test;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.dml.entities.Poilcy;
import com.flc.util.HibernateSessionFactory;
public class Test {
	public static void main(String[] args) {
		SessionFactory sfactory=null;
		Session session=null;
		Poilcy policy=null;
		Transaction transaction=null;
		boolean flag=false;
		try{
			sfactory=HibernateSessionFactory.getInstance();
			session=sfactory.openSession();
			policy =new Poilcy();
```

```
                transaction=session.beginTransaction();
                policy=(Poilcy)session.get(Poilcy.class,10101000);
                System.out.println(policy);
                policy.setPolicyHolderName("sahid");
                policy.setPolicyname("Jeevan Vema");
                policy.setPremium(20000.0f);
                policy.setTenure(10);
                flag=true;
          }finally
          {
                if(session!=null)
                {
                      if(flag==true)
                      {
                            transaction.commit();
                      }
                      else{
                      transaction.rollback();
                      }
                }
          }
    }
}
```

merge()

Note

When we call session.save() then it will not go directly to the database the current state of the object will store in the session level cahce

we take the copy of the entity object which is in the cache level then you update the data then we call session.update()

it goes and reflect these changes in the copy of the session cahe and know the latest copy of the entity object then you will commit then it will gives one time to the databases.

Only one write will be happen such kind of optimization and data has been happen when we use session level cache .

Why id generator generate id before it goes to the databse?

Id generator will not genarte the Id when we go to the database befor itself Id generator will generate the ID in order to store in cache.

If the current state of entity object has been persisted has not been persisted ,it there within the cache

When we call session.update() , it goes to the entity object
it goes to the session , is there any entity object with policy number with 1 ,yes there is with policyno 1 ,is this is persisted ,same reference no(using hascode and equals method) User is not aware about initialization about the entity object he might be overridding the values there in the cache.So it throws an exception Non-unique object I already 1 copy of the object in the cahce which is unsaved and creating 1 more entity object asking me to save .I will not allow throws an exception.

Mappings

Relationship of classes to the relational database is called mapping.
How to desing the database tabe for the RDBMS relationship what way I can store the data of these classes into those related table for my class which is called mapping

There are four types of classes

1. *Pojoclass*
2. *Javabeans*
3. *Component*
4. *Domain/entity*

Pojo(plan old java object) class

When a class directly executable by the undelying jvm then without any third party library reference then the class called as pojo class;

Javabeans

When a class exist with set attributes with accessor method then the class will called as javabeans.

Component

When class contain business functionality that can be carry and used other classes then it is called component class;

Doamin/Entity Object

If it has been created to represent underlying data-model into which we are going to persist .Most of the time this class is going to represent the persisting model.
The domain classes are classes that hold structure of the representing the persistency representation model of the persistancy such classes are called domain classes entity representing the persisting format.

There are two way where classs can related with each other

1. Inheritance
2. Association

Inheritance

One class will extends from another class to reuse the trait of another class a class may extends form another class which is called inheritance.

Association

one class contains another class as an attribute which is called association
A class may use refrence of another class to use the data and funtionalit of another class.
for more clarity go to page no 59;

In how many way table can relatred with each other

1. One to one
2. One to many
3. Many to many.

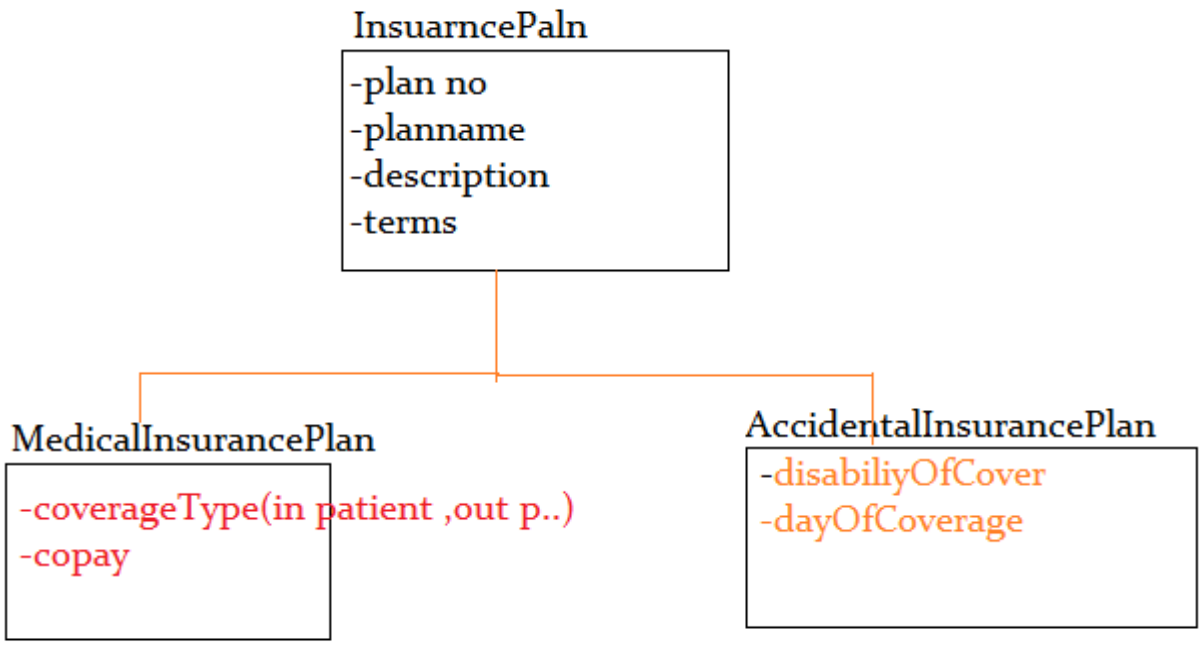
There are two type of mapping
1)Inherintance mapping
2)Assocaition mapping

1)Inherintance mapping
Classes are related with each other via inheritance

There are two aspects of mapping

1. How to create table(design database table representing relationship of classes, based on inheritance or association)
2. How to perform persistency operation on those relation ship

(How to instruct hibernate inquiring and persisting the persisting operation) which is called mapping



1. Table per class hierarchy.(representing the hole inheritance of the classes we create 1 single table)
2. Table per subclass(for a class and the corresponding subclasses come up table only with set of columns corresponding to those classes)
3. Table per concrete classes (For every class create table)
4. Implicit Polymorphism (Which is hibernate specific)

Table per class hierarchy

- The whole hierarchy of classes come up with one table.
- It is not only about storing the data in databse it is about to retrieve the data from the database.(There is no point in storing the data when you cannot acces the data when your store same object which you have store.)

- Inorder to keep track which record of information is related to which class of object that why we maintained 1 extra column discriminator within the table.
- For every entity class we associate one discriminator value.when I give it to hibernate it will store all the data corresponding to the object with discriminator value.
- When we are storing the data into table, In order to distinguish between the records in table we put discriminator in the mapping file.

card.hbm.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 package="com.tbch.entities">
        <class name="Card" table="CARD_INFO" discriminator-value="CARD">
                <id name="cardid" column="CARD_ID">
                        <generator class="increment"></generator>
                </id>
                <discriminator column="CARD_TYPE" type="string" />
                <property name="cardNumber" column="CARD_NO" />
                <property name="validFrom" column="V_FROM" />
                <property name="validUpto" column="V_UPTO" />
                <property name="type" column="TYPE" />
                <property name="ccv" column="CCV" />
        </class>
</hibernate-mapping>
```

creditcard.hbm.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 package="com.tbch.entities" >
<subclass name="CreditCard" extends="Card" discriminator-value="CreditCard">
<property name="transactionLimit" column="TRANSAC_LIMT"/>
<property name="withdrawLimit" column="WITHDRAW_LIMT"/>
</subclass>
</hibernate-mapping>
```

debitcard.hbm.xml

```
<hibernate-mapping package="com.tbch.entities" >
<subclass name="DebiCard" extends="Card" discriminator-value="DEBITCARD">
<property name="debitLimit" column="DEBIT_LIMIT"/>
<property name="debitFreeDays" column="DEBIT_FREE_DAY"/>
</subclass>
</hibernate-mapping>
```

- inorder to inform inform hibernate we write in mapping file that this is not class information this is the subclass which is inherit from parent class we use the

tag <subclass> and which is extending from another class we useWe use one attribute called ‘extends’ in subclass.

- To provide discriminator value as we discuss in above we use one attribute discriminator value we use ‘discriminator-value’.
- While we are using extends we inform to hibernate all the information we have provide as part of the superclass mapping go get the information about what your reuired(for table,id generator)

```
package com.tbch.entities;
import java.util.Date;
public class Card {
	protected int cardid;
	protected String cardNumber;
	protected Date validFrom;
	protected Date validUpto;
	protected String type;
	protected int ccv;
	public int getCardid() {
		return cardid;
	}
	public void setCardid(int cardid) {
		this.cardid = cardid;
	}
//setter and getters
```

```
package com.tbch.entities;
public class CreditCard extends Card{
	private int transactionLimit;
	private float withdrawLimit;
//setter and getters
```

```
package com.tbch.entities;
public class DebitCard extends Card{
	private int debitLimit;
	private int debitFreeDays;
//setter and getters
```

---

card.hbm.xml

```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 package="com.tbch.entities">
        <class name="Card" table="CARD_INFO" discriminator-value="CARD">
                <id name="cardid" column="CARD_ID">
                        <generator class="increment"></generator>
                </id>
                <discriminator column="CARD_TYPE" type="string" />
                <property name="cardNumber" column="CARD_NO" />
                <property name="validFrom" column="V_FROM" />
                <property name="validUpto" column="V_UPTO" />
                <property name="type" column="TYPE" />
                <property name="ccv" column="CCV" />
        </class>
</hibernate-mapping>
```

creditcard.hbm.xml

```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 package="com.tbch.entities" >
<subclass name="CreditCard" extends="Card" discriminator-value="CreditCard">
<property name="transactionLimit" column="TRANSAC_LIMT"/>
<property name="withdrawLimit" column="WITHDRAW_LIMT"/>
</subclass>
</hibernate-mapping>
```

debitcard.hbm.xml

```xml
<hibernate-mapping package="com.tbch.entities" >
<subclass name="DebiCard" extends="Card" discriminator-value="DEBITCARD">
<property name="debitLimit" column="DEBIT_LIMIT"/>
<property name="debitFreeDays" column="DEBIT_FREE_DAY"/>
</subclass>
</hibernate-mapping>
```

```java
package com.tbch.test;
import java.util.Date;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.tbch.entities.Card;
import com.tbch.entities.DebitCard;
import com.tbch.util.HibernateFcatory;
public class Test {
      public static void main(String[] args) {
            boolean flag = true;
            Session session = null;
            Transaction transaction = null;
            session = HibernateFcatory.getSessionfactory().getCurrentSession();
            try {
                  transaction = session.beginTransaction();
                  Card card = new Card();
                  card.setCardNumber("9898-8989-22");
                  card.setValidFrom(new Date());
                  card.setValidUpto(new Date());
                  card.setType("VISA");
```

```
                card.setCcv(117);
                card.setCardid(131);
                session.save(card);
                DebitCard debitcard = new DebitCard();
                debitcard.setCardNumber("9090-8888-45");
                debitcard.setValidFrom(new Date());
                debitcard.setValidUpto(new Date());
                debitcard.setCcv(778);
                debitcard.setType("Rupay");
                debitcard.setDebitFreeDays(45);
                debitcard.setDebitLimit(25000);
                session.save(debitcard);
                flag = true;
          } finally {
                if (transaction != null) {
                      if (flag) {
                            transaction.commit();
                      } else {
                            transaction.rollback();
                      }

                }

          }

    }

}
```

```
package com.tbch.util;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;

public class HibernateFcatory {
      private  static SessionFactory sessionfactory;
      static{
            Configuration configure=new Configuration().configure();
            StandardServiceRegistry sregistry=new
StandardServiceRegistryBuilder().applySettings(configure.getProperties()).build();
            sessionfactory=configure.buildSessionFactory(sregistry);

      }


      public static SessionFactory getSessionfactory() {
            return sessionfactory;
      }
}
```