Introduction to spring boot data jpa

Spring data module (main module)

it is having multiple sub modules like
spring data jpa (for SQL DB s/ws)
spring data jdbc
spring data mongo db (for NoSQL DB s/ws)
and etc....

nearly 30+ sub modules are there in spring core /spring boot core module

(https://spring.io/projects/spring-data)

Persistence :: Storing and managing data for long time is called persistence
Persistence store :: The place where data will be stored and managed for long time ,, eg: files , DB s/ws and etc..
Persistence Operations:: Insert,update,delete and select operations performed on the DB table are called Persistence operations
(Also called as the CURD operations)
CURD/CRUD operations
C ---> create
U ---> update
R ---> Read
D ---> Delete

Need of persistence technology/ Framework

o-r mapping/ORM :: Object Relational Mapping

java app --IO streams----------> files

java app ---jdbc, o-r mapping (best)---- RDBMS DB s/w (SQL DB s/ws)
(oracle ,mysql,postgresql and etc..)

o-r mapping: Object relational mapping

Persistence technology/framework:: it is the technology/framework using which we develop the Persistence logics to perform the Persistence operations (CURD operations)

JDBC :: Persistence Technology
Hibernate :: Persistence framework
spring data jpa :: Persistence framework

=> Files are not good persistence stores .. so they can not used in medium ,large scale projects (DB s/ws are good as persistence stores)
(skip)
=> if use jdbc as the persistence technology to interact with RDBMS s/w the following limitations are there
a) jdbc logics use sql queries and these sql queries are db s/w dependent queries (Persistence logic is DB s/w dependent)
b) changing the db s/w of the project is very difficult process
c) we need to write lots of boilerplate code (repeating code)
d) throws all its exceptions as checked exceptions .. So exception handling is mandatory (needs to go for exception propagation separately)
e) ResultSet object is not Serializble object by default .. So to send its data over the network we need to convert ResultSEt object data to other collections data. (All collections in java are Serializable by default)
and etc...

Limitations of files as the Persistence stores
a) Can not store huge amount of data
b) No Security
c) No Support for SQL
d) No Support for applying the constraints (PK,FK and etc..)
e) Retrieving data by applying multiple conditions is very complex
f) Performing delete and update operations is quite complex
g) we can not keep files in relationships
h) comparing one file with another file is very complex process
and etc...

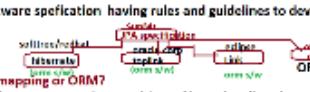
note:: To overcome all these problems take the support of DB s/ws as the persistence stores

JPA :: Java Persistence API

=>To overcome all these problems (above problems) use o-r mapping persistence logics with support of ORM/JPA softwares like hibernate,Ibatis, toplink and etc..

=>Jpa is a software specification having rules and guidelines to develop orm softwares

JPA is theory of o-r mapping
ORM /ORM s/w is pertical of o-r mapping



note: o-r mapping allows to develop the objects based persistence logic with out SQL Queries as the DB s/w indepent persistence logic

ORM :: Object -Relational mapping

what is o-r mapping or ORM?
ans) mapping every record to an object of java class/java bean

Def: The process of mapping/linking java class with Db table , java class properties with db table cols and making the objects of java class representing db table records having synchronization between them is called o-r mapping.
=> Synchronization b/w objects and db table records is nothing but the modifications done in objects will reflect to db table records and vice-versa

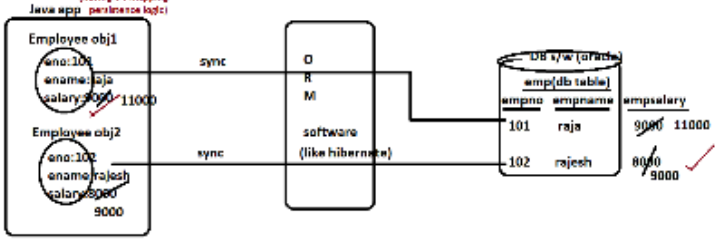
-> JDBC technology is a specification and JDBC drivers/ws is the implementation of that specification
-> servlet technology is a specification and (part of web server s/w) s/w container s/w is the implementation of that specification
=> JPA technology is a specification and ORM s/ws like hibernate,... are the implementation s/ws of that specification

o-r mapping cfgs (xml or annoations)

```
// Entity class/model class/ domain class
class Employee{   (java bean class)
  private int eno;
  private String ename;
  private double salary;
  //setters & getters
  ...
  ...
}
```

properties

Employee <-------> emp (java class) (db table)		
entity class properties	eno <----------> empno	db table cols
	ename <----------> empname	
	salary <----------> empsalary	



=>Here we give all persistence instructions to ORM s/w in the form of objects and the ORM f/w generates the DB s/w specific SQL Queries dynamically .. Since objects are Db s/w independent and we are going deal with only objects while developing persistence logics .. we call o-r mapping persistence logic is DB s/w independent persistence logic

All ORM f/w internally generates the JDBC code + DB s/w specific SQL Queries towards completing Persistence operations on the Db table records

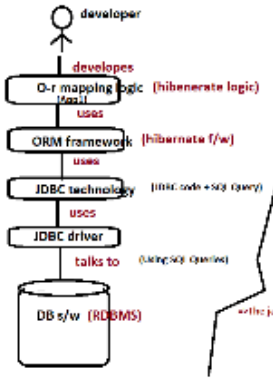
List of ORM sofwares
- hibernate ---------- from softree/RedHat (1)
- IBatis ---------- from apache (3)
- toplink ---------- from oracle corp
- Link ---------- from eclipse (2)
- ojb ---------- from apache
- and etc...

JPA --- specification
ORM is the implementation JPA specification

JPA :: Java Persistence API (old)
(Jakarta Persistence API) (new)
OOB :: Object Java Beans

In O-R mapping , the objects that are representing records can be treated Actors playing the roles of performing one CURD Operation at a time .. once the given CURD operation is over the Object can be used for other CURD operation or can be made ready for garbage collection

=>All the ORM softwares /framework internally use: JDBC to generate the jdbc code and SQL Queries as required for underlying Db s/w based on the objects based persistence instructions recieved from the developers through Java App.



In o-r mapping
- =>saving the object means inserting the data of the object to db table as record (ORM f/w generates jdbc code + INSERT SQL Query)
- =>load the object means selecting a record from db table into the object (ORM f/w generates jdbc code+ SELECT SQL Query)
- => updating the object means updating record represented by the object (ORM f/w generates jdbc code + UPDATE SQL Query)
- => deleting the object means deleting the record represented by object (ORM f/w generates jdbc code + DELETE SQL Query)

note:: we perform all persistence operations on the objects.. but the ORM f/w reflect these operations on db table records by generating JDBC code + DB s/w specific SQL Queries

Note:: Though we got multiple Persistence layers in java for developing persistence logic like hibernate layer/ spring data jpa layer and etc.. but all these layers internally use JDBC code + SQL queries to send and to perform persistence operations on the DB s/w technologies
(After all , the DB s/w understands only SQL Queries)

=>the java bean class that is mapped to db table is called Model class or Entity class or Domain class or persistence class

Advantages of o-r mapping?

a) No need of writing SQL Queries (They will be generated dynamically)
b) allows to develop objects based persistence logic as DB s/w independent persistence logic
c) we can work with objects directly (no need of SQL Queries)

Spring data module (main module

Introduction to spring boot data jpa

it is having multiple sub modules like spring data jpa ✔ spring data jdbc

(for-SQL DB s/ws)

spring data mongo db

(for NoSQL DB s/ws)

and etc....

======

nearly 30+ sub modules are there in spirng data /spring boot data module (https://spring.io/projects/spring-data)

Persistence :: Storing and managing data for long time is called persistence Persistence store :: The place where data will be stored and managed for long time,, eg:: files, DB s/ws and etc.. Persistence Operations:: Insert,update,delete and select operations performed on the

Need of persistence technology/ Framework

o-r mapping/ORM:: Object Relational Mapping

java app

10 streams

-> files

(best)

jdbc, o-r mapping

java app

RDBMS DB s/w (SQL DB s/ws) DB

o-r mapping: Object realational mapping

(oracle,mysql,postgresql and etc..)

be

PB table are called Persistence operations (Also called as the CURD operations) CURD/CRUD operations

C--->create

U---> update

R ---> Read D--->Delete

Persistence technology/framework::

It is the technology/framework using which we develop the Persistence logics to perform the Persistence operations (CURD operations)

=> Files are not good persistence stores.. so they can not used in medium,large scale projects (DB s/ws are good as persistence stores) plain

=> if use jdbc as the persistence technology to interact with RDBMS s/w the following limitations are there

a) jdbc logics use sql queries and these sql queries db s/w dependent queries (Persistence logic is DB s/w dependent) b) changing the db s/w of the project is very difficult process

c) we need to write lots of boilerplate code (repeating code)

d) throws all its exceptions as checed exceptions .. So exception handling is mandatory (needs to go for Exception propagation seperately) e) ResultSet object is not Serializble object by default .. So to send its data over the network

are

we need to convert ResultSEt object data other collections data. (All collections in java serializable by default)

the

JPA :: Java Persistence API

and etc... (above problems) =>To overcome all these problems use o-r mapping persistence logics with support of ORM/JPA softwares like hibernate,ibatis, toplink and etc..

=>Jpa is a software spefication having rules and guidelines to develop orm softwares

softtree/redhat

hibernate

(orm s/w)

what is o-r mapping or ORM?

SunMs

JPA specification

oracle corp toplink

(orm s/w)

eclipse Link

orm s/w

other vendor other ORM s/w

JPA is Theory of o-r mapping

ORM/ORM s/w is partical of o-r mapping note: o-r mapping allows to devleop the objects based persistence logic with out SQL Queries as the DB s/w indepent persistence logic

ORM:: Object -Relational mapping

ans) mapping every record to an object of java class/java bean

Def: The process of mapping/linking java class with Db table, java class properties with db table cols and making the objects of java class representing db table records having synchronization between them is called o-r mapping.

=> Synchronization b/w objects and db table records is nothing but the modifications done in objects will reflect to db table records and vice-versa

// Entity class/model class/ domain class

class Employee{

private int eno;

(java bean class)

private String ename; private double salary;

//setters & getters

...

}

JDBC:: Persistence Technology Hibernate :: Persistence framework

spring data jpa :: Persistence framework

Limitations of files as the Persistence stores

======

========

a) Can not store huge amount of data b) No Security

c) No Support for SQL

===

d) No Support for applying the constraints (PK,FK and etc,,)

e) Retrieving data by applying multiple conditions is very complex

f) Performing delete and update operations is quite complex

d) we can not keep files in relationships

e) comparing one file with another file is very complex process and etc...

note:: To overcome all these problems take the support of DB s/ws as the persistence stores

=> JDBC technology is a specification and JDBC driver s/w is the implementation of that specification

(part of web server s/w)

=> Servlet technology is a specification and ServletContainer s/w is the implementation of that specification

=> JPA technology is a specification and ORM s/ws like hibernate,..., are the implementation s/ws of that specification

o-r mapping cfgs (xml or annoations)

Employee <---------> emp (java class)

(db table)

properties

entity class

eno <--

-> empno

db table

-> empname

cols

(having o-r mapping

Java app persistence logic)

Employee obj1

ename <--

properties salary <----------> empsalary

eno:101

ename:jaja

sync

`salary:9000/11000`

R M

DB s/w (oracle. emp(db table)

empno empname empsalary

101

Employee obj2

software

raja

`9000 11000`

eno:102 enametrajesh salary 8000

sync

(like hibernate)

102

rajesh

8000✓

9000

9000

to

=>Here we give all persistence instructions to ORM s/w in the form of objects and the ORM f/w generates the DB s/w specific SQL Queries dynamically.. Since objects are Db s/w independent and we are going deal with only objects while developing persistence logics .. we canao-r mapping persistence logic is DB s/w independent persistence logic

All ORM f/w intenally generates the JDBC code + DB s/w specific SQL Queries towords completing Persistence operations on the

Db table records

List of ORM sofwares

==========

=====

specification

ORM is the implementation JPA specification

JPA :: Java Persistence API (old) (Jakarta Persistence API) (new) OJB:: Object Java Beans

In O-R mapping, the objects that are representing records can be treated Actors playing the roles of performing one CURD Operation at a time.. once the given CURD operation is over the Object can be used for other CURD operation or can be made ready for garbage collection

JPA

hibernate

iBatis

toplink

from softree/RedHat (1) from apache (3) from oracle corp

Link

from eclipse (2)

ojb

from apache

and etc...

=>All the ORM softwares /framework internally use: JDBC to generate the jdbc code and SQL Queries

as required for underlying Db s/w based on the objects based

developers through Java App.

developer

persistence instructions recieved from the

developes

O-r mapping logic (hibenerate logic)

(App1)

uses

ORM framewo uses

(hibernate f/w)

JDBC technology

(JDBC code + SQL Query)

uses

JDBC driver

talks to

(Using SQL Queries)

DB s/w (RDBMS)

In o-r mapping

==========

=>saving the object means inserting the data of the object to db table as record ORM f/w generates jdbc code + insert SQL Query)

=>Load the object means selecting a record from db table into the object (ORM f/w generates jdbc code+ SELECT SQL Query)

=> updating the object means updating record represented by the object (ORM f/w generates jdbc code + UPDATE SQL Query)

=> deletoing the object means deleting the record represented by object (ORM f/w generates jdbc code + DELETE SQL Query)

note:: we perform all persistence operations on the objects.. but the ORM f/w reflects these operations on db table records by generating JDBC code + DB s/w specific SQL Queries

technologies Note:: Though we got multiple Persistence layers in java for developing persistence logic like hibernate layer/ spring data jpa layer and etc.. but all these layers internally use JDBC code + SQL queries to

send and to perform persistence operations on the DB s/w

(After all, the DB s/w understands only SQL Queries) =>The java bean class that is mapped to db table is called Model class

Advantages of o-r mapping?

a) No need of writing SQL Queries (They will be generated dynamically)

b) allows to develop objects based persistence logic as DB s/w independent persistence logic

c) we can work with objects directly (no need of SQL Queries)

d) change of Db s/w in the project is very easy

as the

or Entity class or Domain class

or persistence class

e) most of the ORM software throw their exceptions unchecked exceptions

them

f) Gives select operation records directly as List of entity objects .. So sending over the network is very easy

g) Supports advanced features like caching, versoning (keeps track of record modifications count),time stamping (keeps track of when the record is modified lastly..)

h) Dynamic schema generation.. (dynamic tables, sequences and etc.. creation)

i) Gives multiple generators to generate id values/pk col values dynamically.. and etc....

PK = primary key

FK = Foreign Key

Limitations of o-r mapping

=> if we select more and more records at time from Db table .. then

more and more objects should be created for entity classes this may lead to

memory issues and application may crash..

with

Best

Solution:: use JDBC.. while dealing huge no.of records processing becoz

the JDBC creates single ResultSet object representing huge no.of records given

by the SELECT SQL Query.

(or)

Enable pagination while selecting huge no.of records,So that we can limit no.of objects creation on per page basis (like 100 to 200 records per page) though we are using ORM dealing

Final conclusion :: JDBC is not outdated, it is stil used in projects directly (while with huge batch of data)

and indirectly (all ORM softwares internally uses JDBC)

=>DB s/w understands only SQL Queries, what ever we use in different layers by getting abstraction like hibernate, spring data jpa, spring ORM and etc .. internally uses the dyanamically JDBC code+ SQL Queries to interact with Db s/w and to manipulate DB data.. (db table records)

generated

do not

=> In spring /spring boot programming, we

use jdbc,hibernate directly.. we use them indirectly through spring ORM, Spring data jpa modules

=> For jdbc style porgamming in in spring /spring boot use spring JDBC (legacy style)

=> For O-r mapping style programming in spring/spring boot use spring ORM (legacy style) or spring data jpa (modern)

(Best) (Latest)

(Developer)

Bit old

developes

Spring /spring boot App

жа

(Developer)

Very Old

developes

Spring /spring boot App

жа

(Developer)

developes

Spring /spring boot App

o- mapping persistence logic

JDBC style persistence logic using

SQL Queries

(o-r mapping persistence logic) having spring data jpa logics (persistence logics)

using

Spring data jpa /spring boot data jpa (Modren)

uses

uses

Spring ORM/spring boot ORM (Legacy model)

Hibernate

uses

JDBC technology

DB s/w

Hibernate

uses

uses

JDBC technology

DB s/w

on

uses

Spring JDBC/Spring Boot JDBC

uses

JDBC technology

DB s w

DB s/w

=> Spring JDBC/spring boot JDBC provides the abstraction JDBC technology and simplifies the app developmet (jdbc style persistence logics) => Spring /spring boot ORM provides the abstraction on Hibernate ORM software to simplify the app developmet

Simplifies o-rm style persistence logics) takes place

=> Spring/spring boot Data JPA provides the more abstraction on Hibernate ORM software to simplify the app developmet more and more

what is the difference b/w Entity class and DAO class?

Ans) Entity class is a java bean class, whose objects represents db table rows.. So these objs

can be used for the persistence operations

to

DAO class is java class that contains persistence logics perform CURD Operations.. thse

Perssistence logics can be written by using the objs of the Entity classes (for ORM style) or using JDBC code

=> Entity class objs are the java class objs who actually represent db table records in ORM mapping persistence logic development. We place these objs based persistence logic in separate java class called DAO class

Conclusion

========

=> Developing IO streams based persistence logic is like washing clothes with hands that to with out detergent powder => Developing JDBC technology based persistence logic is like washing clothes with hands having detergent powder => Developing Hibernate f/w based persistence logic is like washing clothes using semi-automated washing machine => Developing spring ORM/ Spring data jpa based persistence logic is like washing clothes using fully automated washing machine

Spring data/spring boot data module is having 30+ sub modules providing unified model of abstraction on various technologies and frameworks to interact both SQL andNOSQL DB s/ws

ls JDBC Technology outdated or not?

Ans) JDBC Technology is not outdated but its direct utilization is reduced and indirect utilization is increased.. While working ORM frameworks they internally generate JDBC Code +SQL Queries based Persistence logic internally to complete Objects based persistence operations on DB table records