

=> Spring core -> xml + Annotation + Spring Boot ✓

=> Spring Data => JDBC + JPA + Data JPA ✓

Spring Boot profiles ✓

Lombok API ✓

{ logging -> log4j
JDBC
Main
Spring Boot } ✓

=> Spring Data JPA
Spring Boot MongoDB
↓
CRUD Repository
JPA Repository
Custom Query
Pagination And Sorting

MVC
REST
AOP

Persistence => storing data | storage

Persistent store => A place where we store data => DBMS | files - IO

Persistent operations => CRUD => Inserting, updating, deleting data - Retrieving Data.

Persistent logic :- JDBC, Hibernate, Spring Data ✓

JDBC -> Java

ORM => Hibernate -> JPA

Spring JDBC -> Abstraction over JDBC tech.

{ Spring Data JPA }

Spring JDBC ✓

Spring ORM

Spring Framework with ORM ->

↓
Hibernate - JPA

=> DAO Layer -> Repository Layer -> Persistence logic

=> Every table => class => mapping { @Entity

SOO table => SOO class

SRP (Single Responsibility principle).

=> Direct implementation => code to interface

=> create interface -> classes to implement interface

Table => columns length 30 characters

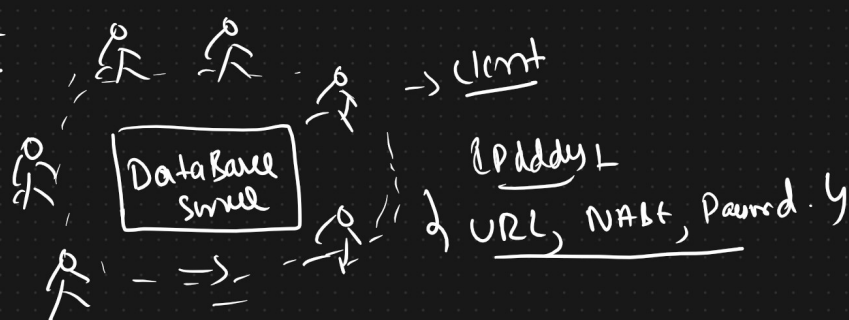
CREDIT_CARD_NUMBER => CREDIT_CARD_NUMBER

action

=> column :-
CREATED_DATE UPDATED_DATE
UPDATED_BY

stored Data
SQL query

=> server / client



CRUD :-

=> { Insert one Data -> Insert multiple
update single -> update multiple Data
Delete single -> Delete multiple Data
Retrieve Data -> Retrieve multiple Data }

=> Every operation we write to methods => custom methods

Interface => 20 Tasks => 20 class
= 4 C
U
D
Boiler plate code.

=> Spring Data JPA

↳ we need not to write single method also to perform any such operation.
↳ CRUD -> implementation will be taken care by this module

=> NoSQL => MongoDB

Spring -> MongoDB + Mongo Driver -> MongoDB (NoSQL)
Spring -> Casandra API + Casandra Driver -> Casandra NoSQL

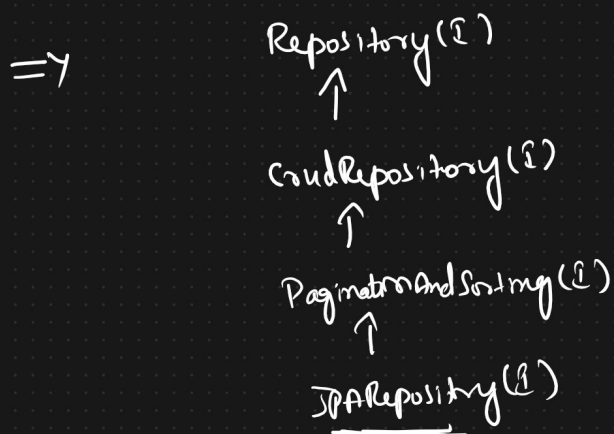
of SpringData JPA \Rightarrow No so effort \Rightarrow Joining
 \hookrightarrow multiple Table

\Rightarrow SpringData JPA \rightarrow provide \rightarrow 2 major interfaces to work with.
 \hookrightarrow CRUDRepository(I)
 \hookrightarrow JpaRepository(I)

\Rightarrow CRUDRepository(I) \Rightarrow CRUD operation

\hookrightarrow JpaRepository(I) \Rightarrow CRUD operation
+
Pagination + Sorting

@Query



\Rightarrow create Interface \Rightarrow must extend

\hookrightarrow CRUDRepository
 \hookrightarrow JpaRepository

methods :- within crud repository :-

\hookrightarrow save(T entity) :- insert record into table.
 \hookrightarrow update same method will perform

Table
primary key

\Rightarrow findById(Long id) \Rightarrow single table

\Rightarrow count()
 \hookrightarrow no. of records in table

\Rightarrow findAllById(Collection<Id> ids)

\Rightarrow findAll() \Rightarrow return all data

\Rightarrow saveAll(Collection<T> entities)

\Rightarrow deleteById()
 \hookrightarrow delete one record

deleteAll() => All records

Soft records => Soft records

deleteAllById(id as a string)

Dependencies for our project:

{ => springData JPA
=> mysql
=> Lombok }

jdk.proxy2.\$Proxy87

↓
Proxy DP

{ implementation added
at time for Repository interface }

App: