Enums(AisleShelfPosition / ShelfPosition class)

<https://dzone.com/articles/using-java-enums>

Singleton (StoreMangement class)

<https://stackoverflow.com/questions/70689/what-is-an-efficient-way-to-implement-a-singleton-pattern-in-java>

<https://www.callicoder.com/java-singleton-design-pattern-example/>

Constructor

<https://stackoverflow.com/questions/285177/how-do-i-call-one-constructor-from-another-in-java>

StoreMap class should work with DB to update store list

Serial number

Unique ID in class

Full design (DAO)

https://www.baeldung.com/java-dao-pattern

The Data Access Object (DAO) pattern is a structural pattern that allows us to

isolate the application/business layer from the persistence layer (usually a

relational database, but it could be any other persistence mechanism)

using an abstract API.

The functionality of this API is to hide from the application all the complexities

involved in performing CRUD operations in the underlying storage mechanism.

This permits both layers to evolve separately without knowing anything about each other.

=====================================================================================

Design as persistence level + service level (also mention relational DB (primary key + foreign key))?

Classes (Store, Aisle, Shelf, Merchandise) stored as Entity (DTO) class on Persistence level

Also may need combine with JPA @Entity as hibernate solution to persist, and bootstrapping a

JPA Entity Manager Programmatically

Classes (StoreManagement, just take this as example) stored as DAO class which perform CRUD to

manipulate Store

Note: Class StoreMap is a mock class to initalize a storage of data like DB

Follow up: Use MongoDB ? First create EER diagram (mysql) on relational DB

Question 1: In Store.java, i have private List<Aisle> storeAisles; how to build it on Store table ?

I am using aisleIndex to map Aisle table aisleID

Question 2: Try to add 1 to N relation between Store and Aisle as identifying relationship, since

aisle not able to locate without Store ID, to add 1 to N relation on Diagram need add

one more field as 'aisleStoreID' on Aisle table ? Which means also need java file change ?

Same for relation between Aisle and Shelf ?

**Example of a OneToMany relationship database**[[edit](https://en.wikibooks.org/w/index.php?title=Java_Persistence/OneToMany&action=edit&section=2)]

<https://en.wikibooks.org/wiki/Java_Persistence/OneToMany#Example_of_a_OneToMany_relationship_database>

### JPA/Hibernate One To Many Relationship Mapping Example

<https://hellokoding.com/jpa-one-to-many-relationship-mapping-example-with-spring-boot-maven-and-mysql/>

Basic one to many relation:

The OneToMany relation between Store and Aisle class should following above example pattern

One to many relation based on condition:

I tried two different ways to implement one to many with required condition??? And because of condition I introduce the Enum of type

Aisle.java has two rightshelf / leftshelf two fields

Shelf.java has HashMap<Enum[front, end], merchandise>

Question 3: How to present constrain like Aisle only has left/right Shelf ? Shelf only has front/back position?

Question 4: How to setup Enum type for field like Shelf.java private AisleShelfPosition side; ? Enum('front','end') ?

Question 5: What to do with database foreign keys in an entity class?

<https://softwareengineering.stackexchange.com/questions/313609/what-to-do-with-database-foreign-keys-in-an-entity-class>

Place relationship between two table (set up FK)

An identifying relationship: identified by a solid line between tables

An identifying relationship is one where the child table cannot be uniquely identified without its parent.

Typically this occurs where an intermediary table is created to resolve a many-to-many relationship.

In such cases, the primary key is usually a composite key made up of the primary keys from the two original tables.

A non-identifying relationship: identified by a broken (dashed) line between tables

Mapping Relational Databases and SQL to MongoDB

<https://code.tutsplus.com/articles/mapping-relational-databases-and-sql-to-mongodb--net-35650>

{

\_id:101,

uniqueStoreID:

}

{

\_id:

aisleIndex:

aisleStoreID:101, // Here use reference id point to 'Store' document

rightShelf: {

shelfID:202

side:

frontPosition:{ // Here is the change to make which different than map presentation on Shelf class

merchandiseOnShelf: {

uniqueSkuNumber:

name:

amount:

expirationDate:

merchandiseShelfID:202 // Reference ID not need as embedded document

}

}

backPosition: {

merchandiseOnShelf: {

uniqueSkuNumber:

name:

amount:

expirationDate:

merchandiseShelfID:202

}

}

}

leftShelf: {

shelfID:203

side:

frontPosition:{ // Here is the change to make which different than map presentation on Shelf class

merchandiseOnShelf: {

uniqueSkuNumber:

name:

amount:

expirationDate:

//merchandiseShelfID:203 // Reference ID not need as embedded document

}

}

backPosition: {

merchandiseOnShelf: {

uniqueSkuNumber:

name:

amount:

expirationDate:

//merchandiseShelfID:203

}

}

}

}