

***HIBERNATE***

**Training Assignment**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 20/11/2012 |

**Hanoi, 09/2019**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
| 1 | 25/07/2020 | Create assignments | Create | DieuNT1 | VinhNV |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Hibernate Mapping 4](#_Toc53479928)

[Movie Theater Application 5](#_Toc53479929)

[Objectives: 5](#_Toc53479930)

[Technical Requirements: 5](#_Toc53479931)

[Problem Descriptions: 5](#_Toc53479932)

[Questions to answer: 5](#_Toc53479933)

[Unit Testing Requirements: 6](#_Toc53479934)

[Guidelines: 6](#_Toc53479935)

|  |  |
| --- | --- |
|  | **CODE: ORM.M.A201**  **TYPE: MEDIUM**  **LOC: NA**  **DURATION: 180 Minutes** |

# Hibernate Mapping

This assignment requires trainees having Java, Maven, SQL, File and Hibernate knowledge. Technical details are:

* Java 8
* [Hibernate 4.3](https://hibernate.org/orm/releases/4.3/)
* [Maven 3](https://maven.apache.org/download.cgi)
* MS SQL Server

The important note:

* Should **avoid many-to-many** design for entities and have to know the differences between **create** and **update** for the property **hbm2ddl.auto**.
* Java classes must follow naming convention rules (for examples, class names and attribute names must not have \_)
* Tables must have names as required on the design (for examples, Employee table has first\_name column instead of firstname)
* The DAO class should have at least 5 methods as getXxxByID, getAllXxx, updateXxxByID, deleteXxxById, insertXxx where Xxx is the model class of DAO. You could choose other verbs instead of get, update, delete, and insert. However, they should be consistent among DAO classes and meaningful.

Happy coding!

Movie Theater Application

Objectives:

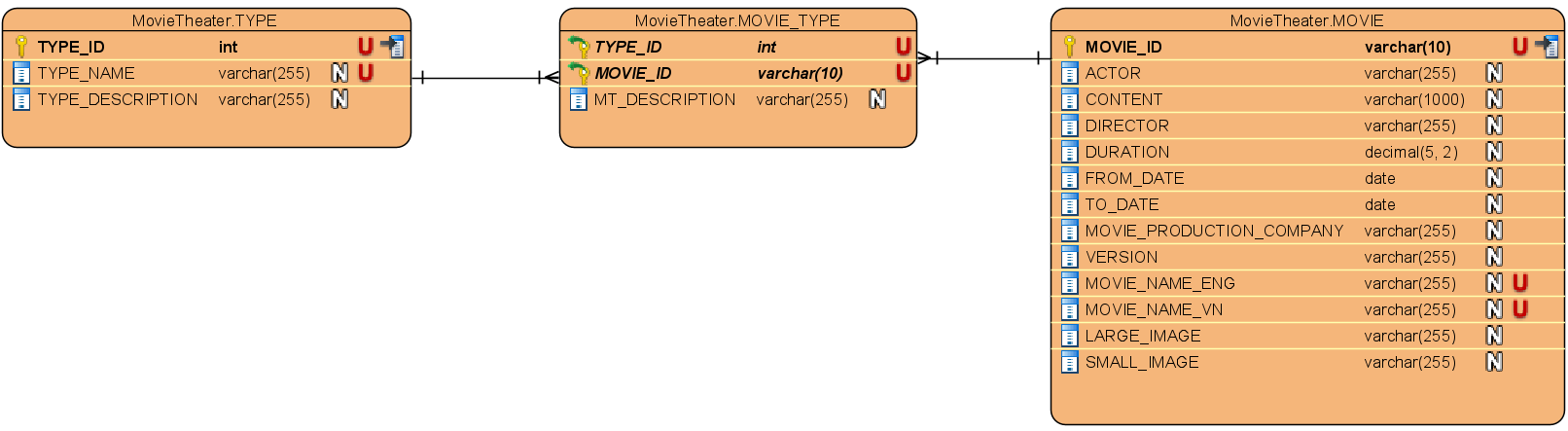
* Understand what is the mapping in Hibernate.
* Know about types of Association mapping.
* Able to use the basic annotations to map Java objects to relational tables.
* Able to implement the association relationship: OneToMany, ManyToMany.
* Know how to implement CRUD operations in Hibernate.
* Know how to implement Lazy Loading and Eager Loading

Technical Requirements:

* Must use Eclipse to build projects.
* Must use tools with versions mentioned above.

Problem Descriptions:

A theater wants to have an application to manage their movie room as well as seat booking. The project team designed the ERD as below after asseted the requirements and seat booking operations.



A movie type includes a id, type name (unique) and description.

A movie is full of information including: id, name, actor, description of content, director, duration, from date, to date, production company, version, movie logo (image).

Each movie type has one or more movies, and each movie also can belong to one or more types.

After developing CRUD operations for this project, you must simulate one senario of all activities. The database name should be **movietheaterdb**.

Questions to answer:

The trainees need to create a new project structure to use Maven managing libraries, project name as **orm.m.a201**.

* Creating a package with name as **fa.training.entities** in this project that contain 3 above entities: Movie, Type, MovieType.
* You must create the appropriate DAO classes for the above models (**TypeDao, MovieDao**, **MovieTypeDao**) to proceed CRUD operations.
* Use the Lazy loading mechanism in the relationships, after that change to Eager loading and make your comparison.

Unit Testing Requirements:

* You must write scripts to test all of the DAO methods that you have developed.

Guidelines:

*The Date or Time columns, it is recommended to use LocalDate or LocalTime to search for high accuracy.*

**-- THE END --**