Lab: Hibernate – JPA

Preamble

To do the lab, please open the Hibernate project that you created with IntelliJ IDEA before coming to class.

1 First Steps

In the `Main` class of the project, remove all the code from the `main` method, as well as the `getSession()` method. Only keep the `static` section, which creates a configuration object and a session factory when the `Main` class is loaded.

1.1 Connecting to Your Local Database

Complete the configuration of the project in order to connect to your local MySQL instance:

- In the `hibernate.cfg.xml` file, set the `connection.url`, `connection.user`, `connection.password`, `connection.driver_class` properties to the appropriate value.
- From Campus, download MySQL's JDBC driver into the `lib` directory of the project, and add the driver to the classpath of the project: `File/Project Structure/Libraries/+`.

Check that the configuration is correct by running the `Main` class.

1.2 Class Product

Using annotations, create the entity class `Product`, with the following attributes: id, name and price. Create the corresponding table in the database.

Notes:

- The id attribute must be automatically managed by both Hibernate and MySQL.
- The `Product` class must be added to the configuration object using the `addAnnotatedClass()` method.

1.3 CRUD Operations

In the `Main` class, develop four static methods that implement the CRUD operations for Product entities. Note: all operations must be executed in a single managed transaction (i.e. no auto-commit).

2 Query Language

The Hibernate Query Language enables you to write database queries that go beyond the simple CRUD operations. See the tutorial about Query Language.

2.1 Select All and Delete All

In the `Main` class, develop a static method that retrieves all the `Product` entities persisted in the database. Then develop a static method that deletes all the `Product` entities from the database.

2.2 Parameterized Queries

Using static parameterized queries, develop two static methods: (i) the first method retrieves all the product whose price is less than a specified value, (ii) the second method raises the price of all the products by a specified amount.

3 Associations

Next, develop the entity class `Order`, with the following attributes: id, date, and ordered products. Develop the CRUD operations for Order entities.