Exercises: Introduction to Hibernate

This document defines the exercise assignments for the "Spring Data" course @ SoftUni.

1. Setup

Use the **provided skeleton** to create **soft_uni** database.

1. Change the **port**, **username** and **password** accordingly to your settings.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
             http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
             version="2.0">

    <persistence-unit name="soft_uni">
        <properties>
            <property name="hibernate.connection.url"
    value="jdbc:mysql://localhost:3306/soft_uni?createDatabaseIfNotExist=true&amp;useSSL=false"/>
            <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver" />
            <property name="hibernate.hbm2ddl.auto" value="update"/>
            <property name="hibernate.connection.username" value=""/>
            <property name="hibernate.connection.password" value=""/>
        </properties>
    </persistence-unit>
</persistence>
```

2. **Create** EntityManagerFactory and **run** your program.

```
EntityManagerFactory factory =
        Persistence.createEntityManagerFactory( persistenceUnitName: "soft_uni");
EntityManager em = factory.createEntityManager();
```

3. Fill the database into **Workbench** or **IntelliJ** by **executing** the provided **.sql** script.

2. Change casing

Use the **soft_uni** database. Persist **all towns** from the database. Detach those whose name length is **more than 5 symbols**. Then transform the **names** of all attached towns **to uppercase** and **save them to the database**.

3. Contains Employee

Use the **soft_uni** database. Write a program that checks if a given employee name **is contained in the database.**

Example

Input	Output
Svetlin Nakov	Yes
John Doe	No

4. Employees with Salary Over 50 000

Write a program that gets the first name of all employees who have salary over **50 000**.

Example:

Output
Terri Jean Ken …

5. Employees from Department

Extract all employees from the **Research and Development** department. Order them by **salary** (in ascending order), then by **id** (in **ascending** order). Print only their **first name**, **last name**, **department name** and **salary**.

Example:

Output
Diane Margheim from Research and Development - $40900.00 Gigi Matthew from Research and Development - $40900.00 Michael Raheem from Research and Development - $42500.00 Svetlin Nakov from Research and Development - $48000.00 Martin Kulov from Research and Development - $48000.00 George Denchev from Research and Development - $48000.00 Dylan Miller from Research and Development - $50500.00

6. Adding a New Address and Updating Employee

Create a new address with **text** "**Vitoshka 15**". Set that address to an **employee** with a **last name**, given as an input.

7. Addresses with Employee Count

Find all addresses, **ordered** by the **number of employees** who live there (**descending**).

Take only the **first 10 addresses** and print their **address text**, **town name** and **employee count**.

Example

Output
163 Nishava Str, ent A, apt. 1, Sofia - 3 employees 7726 Driftwood Drive, Monroe - 2 employees ...

8. Get Employee with Project

Get an **employee by his/her id**. Print only his/her **first name**, **last name**, **job title** and **projects** (only their names). The projects should be **ordered by name** (ascending). The output should be printed in the format given in the example.

Example

Input	Output
147	Linda Randall - Production Technician HL Touring Handlebars ML Road Rear Wheel

	Patch kit Touring-1000
83	John Evans - Production Technician Half-Finger Gloves LL Mountain Handlebars Racing Socks Women's Tights

9. Find Latest 10 Projects

Write a program that prints the **last 10 started projects**. Print **their name, description, start and end date** and **sort them by name** lexicographically. For the output, check the format from the example.

Example

Output
Project name: All-Purpose Bike Stand Project Description: Research, design and development of … Project Start Date:2005-09-01 00:00:00.0 Project End Date: null Project name: Bike Wash Project Description: Research, design and development of … Project Start Date:2005-08-01 00:00:00.0 Project End Date: null Project name: HL Touring Frame Project Description: Research, design and development of … Project Start Date:2005-05-16 16:34:00.0 Project End Date: null …

10. Increase Salaries

Write a program that increases the salaries of all employees, who are in the **Engineering**, **Tool Design**, **Marketing** or **Information Services** departments by **12%**. Then **print the first name, the last name and the salary** for the employees, whose salary was increased.

Example

Output
Roberto Tamburello ($48496.00) Gail Erickson ($36624.00) Jossef Goldberg ($36624.00) Terri Duffy ($71120.00) …

11. Find Employees by First Name

Write a program that finds **all employees**, whose **first name starts with a pattern** given as an input from the console. Print their **first and last names**, their **job title** and **salary** in the format given in the example below.

- **Hint: The expected results of next exercises are with update of salaries in ex 10.**

Example

Input	Output
SA	Sariya Harnpadoungsataya - Marketing Specialist - ($16128.00) Sandra Reategui Alayo - Production Technician - ($9500.00) Sairaj Uddin - Scheduling Assistant - ($16000.00)

	Samantha Smith - Production Technician - ($14000.00) Sameer Tejani - Production Technician - ($11000.00) Sandeep Kaliyath - Production Technician - ($15000.00)

12. Employees Maximum Salaries

Write a program that finds the **max salary** for each **department**. Filter the departments, which max salaries **are not in the range** between 30000 and 70000.

Example

Output
Engineering 71120.00 Sales 72100.00 Marketing 16128.00 Production 84100.00 …

13. Remove Towns

Write a program that **deletes a town**, which name is given as an input. The program should **delete all addresses** that are in the given town. Print on the console the **number of addresses** that were **deleted**. Check the example for the output format.

Example

Input	Output
Sofia	1 address in Sofia deleted
Seattle	44 addresses in Seattle deleted