Exercise: Hibernate Code First + Entity Relations

This document defines the lab assignments for the "Spring Data" course at Software University.

1. Gringotts Database

Your task is to create **table wizard_deposits** using the Code First approach. The table should contain the following fields:

- **id** – Primary Key (number in range [1, 2^{31}-1].
- **first_name** – Text field with max length of 50 symbols.
- **last_name** - Text field with max length of 60 symbols. Required
- **notes** – Text field with max length of 1000 symbols
- **age** – Required
- **magic_wand_creator** – Text field with max length of 100 symbols
- **magic_wand_size** – Number in range [1, 2^{15}-1]
- **deposit_group** - Text field with max length of 20 symbols
- **deposit_start_date** – Date and time field
- **deposit_amount** – Floating point number field
- **deposit_interest** - Floating point number field
- **deposit_charge** - Floating point number field
- **deposit_expiration_date** – Date and time field
- **is_deposit_expired** – Boolean field

2. Sales Database

Create a database for storing data about sales using the Code First approach. The database should have the following tables:

- **product** (id, name, quantity, price)
- **customer** (id, name, email, credit_card_number)
- **store_location** (id, location_name)
- **sale** (id, product_id, customer_id, store_location_id, date)

The relationships between the tables are as follows:

- Sale has one product and a product can be sold in many sales
- Sale has one customer and a customer can participate in many sales
- Sale has one store location and one store location can have many sales

Hint

You can use the following format to design your model classes:

- `Product`
  - `int id`
  - `String name`
  - `Double quantity`
  - `BigDecimal price`
  - `Set<Sale> sales`
- `Customer`
  - `int id`

    - String name
    - String email
    - String creditCardNumber
    - Set<Sale> sales
- **StoreLocation**
    - int id
    - String locationName
    - Set<Sale> sales
- **Sale**
    - int id
    - Product product
    - Customer customer
    - StoreLocation storeLocation
    - Date date

3. University System

Your task is to create a database for a **University System**, using the **Code First** approach. In the database, we should keep information about students, teachers and courses. The database should have the following tables:

- **Student** (id, first name, last name, phone number, average grade, attendance)
- **Teacher** (id, first name, last name, phone number, email, salary per hour)
- **Course** (id, name, description, start date, end date, credits)

The relationships between the tables are as follows:

- Each student can be enrolled in many courses and in each course many students can be enrolled
- A teacher can teach in many courses but one course can be taught only by one teacher

Use class hierarchy to reduce code duplication

4. Hospital Database

Congrats! You were hired as a Junior Database App Developer. But before starting to work you were required to provide some documents such as a fit note from your GP. So, you go to him to get it. When you tell him, what you need and what kind of job you are about to start, he told you that he was just looking for someone to make a software to help him manage and keep data about his patients. He offered to give you the fit note for free if you help him. You thought that's a great opportunity to save 20 leva and go out with friends tonight. Also it would expand your portfolio with 1 project.

Your task is to design a database using the **Code First** approach. The GP needs to keep information about his **patients**. Each patient has **first name**, **last name**, **address**, **email**, **date of birth**, **picture**, **information whether he has medical insurance or not**. The GP should also keep **history about all his visitations**, **diagnoses** and **prescribed medicaments**. Each **visitation** has **date** and **comments**. Each **diagnose** has **name** and **comments** for it. Each **medicament** has **name**. Make sure all data is **validated** before inserting it in the database.

Bonus Task

Make console based user interface, which the doctor can use easily with the database.

5. Bills Payment System

Your task is to create a database for a **Bills Payment System**, using the **Code First** approach. In the database, you should keep information about the **users** who are using that system (**first name, last name, email, password, billing**

details). Every **billing detail** has **number** and **owner**. Also, there are **two types** of billing details - **credit card** and **bank account**. The credit card has **card type, expiration month, expiration year**. And the bank account has **bank name** and **SWIFT code**.

6. *Football Betting Database

Your task is to create a database for the **Football Bookmaker System**, using the **Code First** approach. Model the following tables:

- **Teams** – Id, Name, Logo, 3 letter Initials (JUV, LIV, ARS…), Primary Kit Color, Secondary Kit Color, Town, Budget
- **Colors** – Id, Name
- **Towns** – Id, Name, Country
- **Countries** – Id (3 letters – for example BUL, USA, GER, FRA, ITA…), Name, Continent
- **Continents** – Id, Name
- **Players** – Id, Name, Squad Number, Team, Position, Is Currently Injured
- **Position** – Id (2 letters – GK, DF, MF, FW…), position description (for example – goal keeper, defender…)
- **PlayerStatistics** – Game, Player, Scored Goals, Player Assists, Played Minutes During Game, (PK = Game + Player)
- **Games** – Id, Home Team, Away Team, Home Goals, Away Goals, Date and Time of Game, Home team Win bet rate, Away Team Win Bet Rate, Draw Game Bet Rate, Round, Competition)
- **Rounds** – Id, Name (for example Groups, League, 1/8 Final, 1/4 Final, Semi-Final, Final…)
- **Competitions** – Id, Name, Type (local, national, international)
- **CompetitionTypes** – Id, Name
- **BetGame** – Game, Bet, Result Prediction (PK = Game + Bet)
- **Bets** – Id, Bet Money, Date and Time of Bet, User
- **ResultPrediction** – Id, Prediction (possible values - Home Team Win, Draw Game, Away Team Win)
- **Users** – Id, Username, Password, Email, Full Name, Balance

The relationships between the tables are as follows:

- Team has one primary kit color and one secondary kit color
- Team is resident in one town
- Each town can host several teams
- Town can be placed in one country and a country can have many towns
- Country can be placed in several continents and a continent can have many countries
- Player can play for one team and one team can have many players that play for it
- Player can play at one position and one position can be played by many players
- Player can play in many games and in each game, many players take part
- Additionally, for each player for given game is kept statistics such as scored goals, goal assists and minutes played during given game
- A game can be played in one round and in one round many games can be played
- A game can be played in one competition and in one competition many games can be played
- On a game, many bets can be placed and one bet can be on several games
- Each bet for given game must have prediction result
- A bet can be placed by only one user and one user can place many bets

Hint - Database Schema