# **Final Project: CoffeeZone (draft)**

# **Databases2**

## Objectives:

* Analyze and design a relational database for a given problem
* Use SQL queries to create a physical database model
* Create PL/SQL subprograms to answers the questions in the given problem and handle potential errors.
* Develop a Customer facing application that allow an external user to perform:
  + Registration and Login
  + View products
  + Place and modify orders

## Problem:

The Client, a coffee shop chain wants to sell coffee products and cakes online and delivery them to all customers. A customer should login to the shop interface in order to place an order. Visitors can view the products offer but cannot order them unless they are registered.

## Database Requirements:

**Data**:

* **Customers**:  
   We want to store information about the customers who order coffee in the CoffeeZone online store. Each customer is identified by a username and password that they choose upon the registration, as well as a delivery address were the items should be shipped to. They may also have a phone number and email address to receive notifications or specials.   
  A customer should have a referral id that contains the id of the customer that has referred the customer to the CoffeeZone.
* **Products/items**:   
  We want to store information about different kinds of products sold in the coffee shop. There must be a unique identifier associated with each product, a product name, a retail price. For all products, the store should keep track of the product left in stock. No product should be sold if it is out of stock. It’s up to you to add all fields specific to each product.
  + When choosing the different kind of products of the coffee shop think about the relationship between them. For example: an expresso can be bought with/without sugar, with/without milk. Sugar cannot be sold on its own. Also think about those kinds of product that does not belong to a coffee shop.
  + It is up to you to layout additional requirements for your coffee shop
* **Menu**:   
  In CoffeZone, each customer can build its own menu online. A menu is made up of different items of different kind of products. Items are included in the menu, as well as the quantity and the price of each item.  
  Example: A customer menu can be made of these items: 2 small Coffees at 1.50$(CAD) each , with 2 sugar each at 0$ each, one milk in one of the coffees and one cream in the other each one at (0.02$ each) and 2 muffins at 1$ each. A menu is part of a customer order.
* **Shopping cart:** Customers should have their own shopping cart. Menus can be added to the cart to be checked out at the end of the shopping process. Each menu in the cart should have a quantity indicating how many of this menu is being purchase.
* **Orders:** Once a customer has put all the menu(s) inside their cart, they can check out all those menus at once. An order is identified by a unique order number. It should have an address to which the order is being delivered. An order must know which menus are included in the order, and how many of them they are. Each order has the date at which it was placed and the date at which it was completed and sent out for delivery.

## Application requirements

You are responsible of the database design. You will also create an application the customers can use to access the CoffeZone storefront.

Your store front can be text-based interface if you are working alone,

or a graphical user interface if you are working in a group.

* **Registration**: Validate the customer data and insert it inside the database. Ask the customer all necessary data to create a new customer in the database.
* **Login**: A customer should provide a username and a password. The password should be hashed and salted before it is saved to the database. At each customer logging, Log the customer connection time (day and hour) inside a logging table or inside a column of the customers table. Additional information can be added as well.
* **Ordering**: Display the list of products in the front-end. Allow everyone to view the list of available items. Only logged-in customer can select one or more items among the available ones and choose the quantity of each item then add them to their menu.   
  Example: 1coffee, 1cake, 0milk, 2sugar, 1croissants.  
  Upon adding a new item to the menu, validate the selections, add a new menu in the database and calculate the price of the menu. The total price of the menu should be displayed to the customer, as well as an option to add a new menu and to checkout. Verify the quantity in stock before ordering. Raise and catch any exceptions. Avoid customers to order out of the store delivery hours. Delivery are allowed between 6:00 am to 10:00 PM 7days a week.
* **Checkout**:   
  Complete the checkout of the logged-in customer. Display the delivery address. Give the customer the possibility to update the delivery address. Upon completion, add the card data to the database and display a success message to the user. In case of error, also display an error message to the user.
* **Customer referrals** **and additional features**:   
  A register customer can have bonus of 1 free menu of 5$ per customer referral. A customer can refer a maximum of 3 other people.

## Submission:

Submit a zipped file with all your project files containing:

Your ERD, SQL files (create tables, insert, pl/sql subprograms) and your java project.