CSCI 3901 Assignment 5

Winter 2022

Due date: 4:00pm Wednesday, April 6, 2022 in git.cs.dal.ca at

https://git.cs.dal.ca/courses/2022-winter/csci-3901/assignment5/xxxx.git

where xxxx is your CSID (this repository already exists, so clone it and then add your code to it).

Problem 1

Goal
Design a database schema.

Question (5 marks)

From "Modern Database Management" by Hoffer, Prescott, and Topi, 9th edition, 2009, Pearson Prentice Hall.

(5 marks)

An art museum owns a large volume of works of art. Each work of art is described by an item code (identifier), title, type, and size; size is further composed of height, width, and weight. A work of art is developed by an artist, but the artist for some works is unknown. An artist is described by an artist ID (identifier), name, date of birth, and date of death (which is null for still living artists). Only data about artists for works currently owned by the museum are kept in the database. At any point in time, a work of art is either on display at the museum, held in storage, away from the museum as part of a traveling show, or on loan to another gallery. If on display at the museum, a work of art is also described by its location within the museum. A traveling show is described by a show ID (identifier), city in which the show is currently appearing, and the start and end dates of the show. Many of the museum works may be part of a given show, and only active shows with at least one museum work of art need be represented in the database. Finally, another gallery is described by a gallery ID (identifier), name, and city. The museum wants to retain a complete history of loaning a work of art to other galleries, and each time a work is loaned, the museum wants to know the date the work was loaned and the date it was returned.

Problem 2

Goal
Make some meaningful changes to a database.

The Northwind food distribution company for whom you worked in Assignment 5 wants to improve its inventory management system. It has recognized the following deficiencies in its database:
- The company doesn't have a history of the purchases made from suppliers or the price at which it bought its products.
- The company doesn't know when it placed an order to restock its products.
- The company doesn't know how frequently it reorders from its suppliers so it can't know if its current reorder levels are good.
- The process of reordering products is currently done manually.
- There is no clean way for the company to record when products arrive from suppliers.

Your task is to remedy these problems for the company. More specifically, you are responsible for:
1. Modifying the database so that we can track orders from suppliers, know the cost at which we bought the product, know when the products have arrived, know who will be delivering the product to us, and have a reference to track the product while it is in transit.
2. Record when an order for one of our own clients is shipped, update the inventory, and automatically trigger a reorder from the supplier, if necessary, knowing that the number to reorder can vary by product. We only want one reorder to be sent at the end of the day so we'll have two methods:
   a. Ship_order – happens for each order that we send out
   b. Issue_reorders – happens once per day where we place orders to our own suppliers; we only want at most one order to suppliers each day.
3. Record when we receive an order from one of our suppliers.

Database changes
Make a small analysis of what will be needed to fill the business gaps. Create an ER diagram to reflect the changes that you need to make to the database, and create a file with the SQL commands to effect the changes to the database.

Daily transactions
Create a class that implements the following interface:

```
public interface inventoryControl {
        public void Ship_order( int orderNumber ) throws OrderException;
        public int Issue_reorders( int year, int month, int day );
        public void Receive_order( int internal_order_reference ) throws OrderException;
}
```

Two of the methods throw an "OrderException". You need to create that exception. It should allow me to retrieve information on what went wrong with a getMessage() method and to retrieve a reference integer for the order in question with a getReference() method.

The methods have the following functionality:

Ship_order   Given an order placed by one of our customers, do whatever work is needed to indicate that the order has now left our company and the products are out of our inventory. We call this method for each order that we ship.

Issue_reorders   This method is executed once per day at the end of the day. It create all of the information that belongs in a request to one of our suppliers to get products that have fallen at or below the reorder levels, which we'll call a purchase order. Each supplier should get at most one purchase order at the end of the day.

When it comes to establishing the item cost in the purchase order, assume that our company has a standard 15% markup on the price, so use the price in the last sale of the day to estimate the item cost.

Return the number of suppliers from whom we will be placing an order.

Just for testing, we're supplying the date as parameters to this method. Using those parameters, we can pretend that we're running the method at the end of different days.

Receive_order   Given some identifier for an order that we placed to one of our suppliers (that you created as part of the reorder process), do whatever work is needed to know that the order was received and that the inventory is now correct.

Assumptions
- The SQL to modify the database will always be executed on the database.

Constraints
- You may use any data structure from the Java Collection Framework.
- Write your solution in Java and SQL. The solution code must be your own.
- Use the mysql JDBC connection for Java.
- If in doubt for testing, I will be running your program on bluenose.cs.dal.ca. Correct operation of your program shouldn't rely on any packages that aren't available on that system.

Notes
- Use SQL vs Java as you deem best.
- Be sure to document your approach and any resources that you use.
- Look at where the bulk of the marks are in the marking scheme to help focus your efforts.

- You will need to make any database changes to your own copy of the Northwood database. You do not have the access permissions to change the csci3901 database.
- We have other software written to use the current database. Your changes should not impact that software.
- You are expected to submit
  - The ER diagram for the new database
  - The SQL statements to update the database
  - A class that implements the inventoryControl interface
  - External documentation that includes testing information

Marking scheme

- Documentation (internal and external) – 4 marks
- Database design update and ER diagram – 3 marks
- Implementation of the inventoryControl interface – 10 marks
- Argument as to why we should believe that your program operates as required – 3 marks