**Allergo Music Store**

**Introduction**

In this project, you will be designing and implementing a database for an enterprise and some basic operations required by the primary users of the database. The project will be developed in groups of three or four and will be divided into three parts:

* Part I : Functional Specification and Conceptual Design
* Part II : Logical Design (normalization etc.)
* Part III:  Complete Implementation and Demonstration.

Parts I and II each will be worth 25% of the project mark, while part III will be 50% of the total project mark.

**Project Description**

The database system that you are to implement in this project is for a large music retail company with multiple stores and an on-line outlet (like Virgin or Music World), called herein **Allegro Music Store (or AMS)**, or simply **"the company"**. The rest of this section provides a detailed description of the basic operations which the users of the database system expect to perform.

The main function of AMS is to sell music to its customers. AMS has  a number of stores in a number of cities in Canada and also has an online outlet. A store is uniquely identified by a name (like *Vancouver-Main* ), and  has an address, city and phone associated with it. There is a special store called *"Warehouse"* that only services the online orders. Each store keeps two types of music:  audio CD's and DVD's with music videos. The stores have also music books and  sheet music. Each item is uniquely identified by a universal product code (UPC). For each audio CD and  DVD we need to record their title, the leading singer(s) or group, the production company and  year,  and the titles of the songs which are  contained in it.  Each CD and DVD belongs to one of the following music categories: *rock, pop, rap, country, classical, new age* and *instrumental*.  Each book has a title, one or more authors,  a publisher and a year. Finally a music score has a title, one or more composers and a publishing company.

Every item has a price, an indication whether it is taxable, and is supplied by one or more suppliers. For each supplier the company would like to record their name (which is unique), their address, their city, and their status, which can be "active" or "inactive".

To buy an item, an AMS customer can either visit a store and make a regular purchase or can access the on-line AMS site and place an order.  When customers shop in a regular store, they can pick up the items and pay at a check-out counter. A customer may pay for their purchase either by cash, or by one of the acceptable credit cards. The company currently accepts Master Card, Visa and American Express cards. When the customer requests a credit card charge, the clerk requests authorization from the credit card company. If authorization fails the customer is asked to pay cash. If the customer refuses, the purchase is cancelled.  When the payment is completed the system prints a receipt (with a unique receipt number ) for the purchase. The company does not keep records for its customers who shop at the regular stores, but when a customer pays by a credit card, the card number and expiry date are recorded with the purchase.

Alternatively, customers can buy goods from the company's on-line site. When at the site, customers can identify themselves  by typing their customer id and password.  Customers who access the online store for the first time, will be asked to register by providing their personal information, including their name, address, city, phone number, an id and password.   If the id is already in the system, they will be asked to provide another one.  To shop at the site, registered customers will be given a virtual shopping basket and asked to specify the  item they want to buy. The customer will describe the category,  the item title (optional)  and the quantity. If the information is not enough to define a single item, the system will display all the items that match the input and ask the customer to select one.  When an item is selected, it will be added to the customer shopping cart. Each time an item is selected the systems has to make sure that there is enough quantity in the Warehouse to complete the purchase. Otherwise the system will ask the customer to accept the existing quantity.   The customer can repeat the same process for any number of items.   When the customer is ready to check-out, the system will produce a bill with the items and the total amount. The client has to provide a credit card number and expiry date to complete the transaction.  After that,  the system will create an order for the Warehouse and inform the customer about the number of days it will take to receive the goods.  This number is estimated by the number of outstanding orders and  the maximum number of orders that can be delivered in a day (which is fixed).

Customers can return merchandises to any store within 15 days from the purchase date. When an item is returned, the system verifies the receipt and issues a refund in the same  payment form that was used for the original purchase. That is,  if cash was used for the original purchase, the refund is in cash. Otherwise, the credit card  is credited.

In addition to customer orders and deliveries, the system should keep track of the in-store purchases, the returns, and the supply shipments.  A  store may order a number of goods from a supplier. Each order is processed manually and is outside the scope of this system.  When a supply shipment comes from a supplier to a store, the store personnel has to record the shipment and update the inventory.

The system should perform a variety of transactions. The most frequent operations that we need to implement in the first release of the system include

* process a purchase of  items at a conventional store
* register customers for online purchase
* process an online purchase of items
* record the delivery for an online order
* process the return of an item for a refund
* add and remove suppliers,
* process a shipment from a supplier to a store
* print a variety of reports.

At the end of each day, the company wants to create a report with all items sold during the day, grouped by store and item category, and showing the subtotal sales for each category, each store, and the grand total for the day. Every week, the company would like to produce reports showing for each store the items that are low in stock (i.e. have less than 10 pieces on the shelves), or  just showing the low stock items in a specific store.

Finally there are a number of simple queries the system users should be able to ask. They should be able to

* get all the relevant information for a given store and  item (by providing the store's name and the item title), including the stock for that item
* get the suppliers for a given item
* show the top 5 (in terms of quantity)  items and the total quantity that the company sold in a given time period
* get the titles and the stock of the CD's for a given artist (leading singer)
* get the titles of the DVDs that are produced by a given producer.

The system you design will be used by three types of user:

* **customer:** anyone who buys items online
* **clerk**: a typical check-out  counter clerk who performs in-store purchases and returns,
* **manager**: a user  who performs all the rest

**Note 1:**

Normally, when a user logs in, the system detects their privileges and provides them with the appropriate interface. Nevertheless, your system won't deal with clerk and manager information, It will only maintain the customers data.  The data for clerks and managers is maintained by the system administrator module which is out of the scope of this project.  Therefore, in your implementation you will hard code a couple of clerk and manager accounts and when these users log in the system will display the appropriate menu.

**Note 2:**

The requirements stated herein are not expected to be complete. As you start analyzing these requirements you may notice that certain details are missing. In this case, you may make any reasonable assumptions about them, but if there is any uncertainty about some requirements you should post an inquiry on the bulletin board. The TAs and the instructor will try to respond to these requests as soon as possible, so that everyone would follow the same requirements.

**Project Part I**

Once the requirements are realized, write a **short requirements specification document** in some requirements specification language of your choice. In this document you have to identify the different operations the system has to perform and describe their behaviour.  For each operation you should specify:

* the user who initiates this operation
* the inputs for the operation
* the output of the operation
* what the operation does in terms of the entity sets and  the relationship sets that you have identify in the ER diagram
* error and exceptional cases for the operation (the most important ones).

Here is an example of the specification of the customer registration operation:

**Register Customer**

**User:** Customer

**Input:** Customer name, address, phone,  log id and password.

**Output:** "OK" or "Cancelled"

**Basic Case**:

  The system checks the given id and the id does not exist. The system creates a new entity in the customer set and returns "OK".

**Exceptions**:

* If the log id exists, the system keeps asking for a new one, until a new id or a "cancel " is supplied.  If a "cancel" is selected, the  transaction is aborted and "Cancelled" is returned.

Once the requirements have been identified, perform a data analysis and draw an **Entity-Relationship diagram**for the enterprise. Make sure to:

* use meaningful names for the entity and relationship sets
* make the distinction between week and strong entity sets
* underline the primary keys for the entity sets
* specify the participation for the relationships (total or partial)
* specify the multiplicity degrees for the relationships (one-to-one, one to many, many-to-many).
* specify the specialization/generalization type (disjoint, overlapping)  and additional  attributes

 For each entity and relationship set, if you cannot find a meaningful name for it or if there is additional information that is not shown on the ER diagram,  provide a short paragraph with that information. For guidelines,  have a look at  the relevant tutorial which provides a more in-depth information for this first part of the project.