## CS 443 Database Systems

## MyDB 1 (100 points)

## Due Friday June 23rd as a single document submitted to the dropbox.

**MyDB 1: Step 1 of your own Database application (MyDB)**

Note: This assignment is a modification of material developed by Stanford Database Group

As the course progresses you will be building a non-trivial database application for a real-world scenario of your choosing. You will design schemas for the database, then using MySQL: create your database, create synthetic data, load the data, query the system, and write complete interactive packages that use the database.

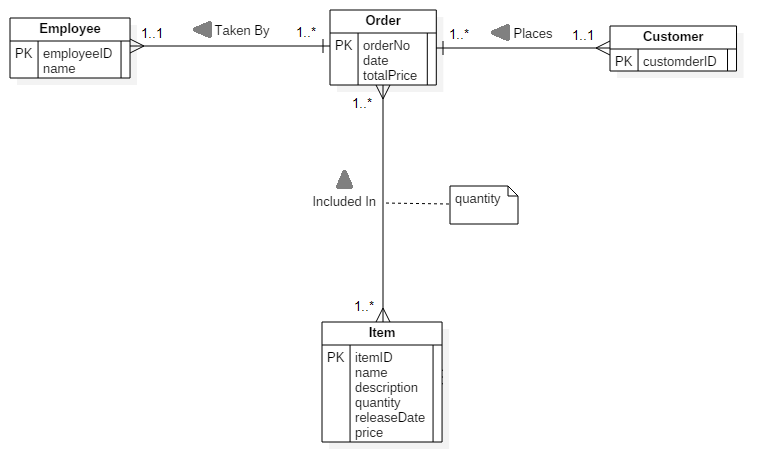
Your first step is to identify the domain you would like to manage with your database, and to construct an Entity-Relationship (ER) schema design for the data. We suggest that you pick an application that you will enjoy working with, since you will be stuck with it for the whole semester! For example, pick something you are interested in—a hobby, material from another course, a research project, etc. Get the most out of this part of CS 443.

Try to pick an application that is relatively substantial, but not too enormous. For example, when expressed in the entity-Relationship (ER) model, you might want your design to have in the range of four or more entity sets, and a similar number of relationships. Note that this is a ballpark figure only! You should certainly include different kinds of relationship (e.g., many-one, many-many) and different kinds of data (strings, integers, etc.), but your application need not necessarily require advanced features such as weak entity sets or roles in ER.

1. (15 pts.) Write a short description of the database application you propose to work with throughout the course. Your description should be brief and relatively informal but should be clear. If there are any unique or particularly difficult aspects of your proposed application, please point them out. Your description will be graded only on suitability and conciseness.

The database application I am proposing to work with is a video game store where customers will be able to buy various items. Each item has a name, description, quantity, release date, price and unique item ID. The employees would take care of any orders placed by customers. Employees have names and unique IDs. Customers have unique customer IDs. Customers do not need to register with the store and will receive a new customer ID for every purchase. The order will have a date, total price of the items and a unique order number.

1. (50 pts.) Specify an Entity-Relationship (ER) diagram for your proposed database. As always, don’t forget to specify primary key attributes and include arrow heads indicating the description of relationships. Export as an image and place in the document.



1. (25 pts.) Consider the ER diagram. Use the method described in the textbook and in class for translating an ER schema to relations to produce a set of relations for your database design. Please specify your relational schema and underline key attributes.

Employee( employeeID, name)

Customer( customerID)

Order( orderNo, date, totalPrice, employeeID, customerID)

IncludedIn( orderNo, itemID, quantity)

Item( itemID, name, description, quantity, releaseDate, price)

1. (10 pts.) Describe two high level interactions (involving 2 or more relations) your database should support. For each describe which relations are involved and how they are involved. (For example, in a Sailor database I may want to have a remove sailor function. The remove sailor function will delete all reservations by that sailor from the Reserve relation, and then remove the user from the Sailor relation. Another example might be to remove boat functions which lists all future reservation information (including sailor name/sid) for that boat. It removes all reservations from the Reserve relation, and then removes the boat from the Boat relation.

Interaction #1: A customer may want to remove items from their orders so it would be good to include a remove item function or a cancel order function if the customer no longer wants to buy anything. The remove item function will delete items a customer placed on their order from the IncludedIn relation and reduce the total price from the Order relation. The cancel order function will instead delete the tuples relating to an order from the Order and IncludedIn relation. The customer ID would also be deleted from the Customer relation if they do not have any other orders on file.

Interaction #2: The introduction of a new item would require an add item function to add the item’s information to the Item relation. Of course, there would also need to be a delete item function to delete the item information from the Item relation in case an item is no longer being sold.

Don’t forget to save a copy of your MyDB 1 for reference as you do future steps of the MyDB assignment. If you are having trouble thinking of an application, or if you are unsure whether your proposed application is appropriate, please feel free to consult with the Professor.