

**Concordia University
Dept. of Computer Science &
Software Engineering**

**Comp 353 - Databases
Main Project**

Title: Book Store

Reports Due: Monday April 8th before noon

Demos Due: April 8th and 9th

A bookstore that provides a wide variety of reader interests would like you and your team to design and implement a database application system to manage the inventory and their day to day business operations. The requirements data to be modelled are as follows:

- Information about the **books** includes their ISBN number, title, author(s), edition, price, quantity-on-hand, and year-to-date-quantity-sold.
- Information about the **employees** include their employee-id, name, SSN, phone number, email, home address (civic number, city, province, and postal code).
- **Customers** are identified by their name, phone number, email, address, and their year-to-date purchase information.
- **Publishers** are identified by their publisher-number, company-name, branches information, phone number, email, address. A publishing company may have multiple branches (e.g., east coast, mid-west, etc.).
- **Branches** are identified by their branch-name, branch manager, phone number, email, and address. Any branch for a given publishing company can supply any book published by that company. Each branch can have only one representative.
- **Orders** are identified by their order-number, order-date, publisher, branch, books ordered, and the quantity for each order.
- The **Bookstore** can order books from several publishers.
- A book is supplied by only one publisher.
- A customer can put an **order** if the book is not available in the inventory. In that case, a log record is created and a unique confirmation number returned to the user. In that case, an employee responsible for managing the orders, groups the orders based on the publishers and **sends them on a weekly basis to the publishers** together with a period specified as well (about two weeks) to receive the order(s). Once received, the books are **shipped** to the customers ordered. The employee will be able to track books that have been ordered but not received within the period set.
- A book may appear on several orders by different customers and an **order may include several books**.
- An employee can make a **sale** to a customer. In addition to the employee and the customer's information, the sale include the date, the quantity and price of each book sold.

- An employee can receive a shipment of books from the publisher. In addition to the employee and the publisher's information, each shipment includes the date of the shipment, the books received and the quantity of each book received.

With this information, do the following initial steps in your database design process:

1. Develop an ER diagram to represent the conceptual database scheme for the above “application”.
2. In the diagram, mark the various constraints (keys, functional dependencies, cardinalities of the relationships, etc.). Identify any constraints that are not captured by the ER diagram.
3. Convert your ER diagram into a relational database schema. Make refinements to the DB schema if necessary. Identify the primary keys and the foreign keys of the relations.

Formulate and evaluate the following SQL queries against an instance of your database in which every relation is populated with “sufficient” representative tuples.

- a. Get detail of all books in the Bookstore.
- b. Get detail of all books that are back order.
- c. Get detail of all the special orders for a given customer.
- d. Get detail of all purchases made by a given customer.
- e. Get detail of all the sales made by a given employee on a specific date.
- f. Get details of all purchases made. For each customer, return the total amount paid for the books ordered since the beginning of the year.
- g. List every book ordered but not received within the period set has passed.

What you should submit:

You should submit a report that includes the E/R diagram, database schemas and functional dependencies, SQL declarations of the relations, the implementation code, relation instances, and the SQL scripts for the queries and transactions, and 5 tuples of each query result. Build a useful web interfaces to facilitate employers/users’ interactions with the application system.