CSCI3170 (2024 Spring)
Database System Project –
Book Ordering System

Group Registration Deadline: 23:59 12th Feb 2024

Phase 1 Deadline: 23:59 25th Feb 2024

Phase 2 Deadline: 23:59 7th April 2024

1. Introduction

You are asked to implement an ordering system for a bookstore so that all information on books, customers, and purchase information are stored and accessed online. The system has to support interactive inquiries from the customers and the bookstore staff.

This project is divided into two phases:
In phase 1, you have to design the database (including an ER-diagram and a relational schema). A suggested solution will be provided after phase 1. You are required to use the suggested solution to complete phase 2.

In phase 2, you are required to implement the system as a Java command-line program. Our tutors will give tutorials about connecting your program to an Oracle database system with JDBC API and deploying the required platform.

This is a **group project**, and each group should have **three** members. ONLY one copy of the solution is required for each group. Please fill out the group registration form in the **Blackboard** system before the group registration deadline. You will be **randomly assigned** to a group if you are not in a group after the registration period. If you want to form a one-person group or a two-person group, please send an email to tutor Zhiyuan He (zyhe@cse.cuhk.edu.hk).

2. Milestones

Preparation

- Read the document thoroughly and make sure you understand all the assumptions and regulations stated in Section 4.

Phase 1 (20 %)

- According to the data specifications in Section 3, design an ER-diagram and transform it into a relational schema without any redundant fields and tables.

Phase 2 (80 %)

- According to the suggested solution of phase 1, implement a Java application that fulfills all requirements stated in Section 5.
- Debug your system with different datasets and user inputs.
- Write a README file to describe the compilation and deployment of your system.

3. Data Specifications

Our tutors will provide the testing data. You need to insert these data into the Oracle database by yourself. Moreover, you need to implement the creation of tables, the deletion of tables and the insertion of data in the system interface in Java.

We divide the data into three main categories (Book, Customer and Order). These categories are used for easier explanation only. They are not meant to be the tables in the final relational design. You should design your own table schemas appropriately as stated in Section 4. In the following sections, *we use "C" and "X" to represent capital letter and digit respectively.*

3.1 Book

The information of a book:

1. **ISBN**: 13 chars with format "X-XXXX-XXXX-X"
2. **Book Title**: non-empty string of maximum 100 chars
3. **Authors**: a list of authors where each author is a non-empty string of maximum 50 characters (NOTE: There is no comma (",") in an author name.)
4. **Unit Price**: non-negative integer
5. **No of Copies Available**: non-negative integer

3.2 Customer

The information of a customer:

1. **Customer ID**: non-empty string of maximum 10 chars
2. **Customer Name**: non-empty string of maximum 50 characters
3. **Shipping Address**: non-empty string of maximum 200 characters. The components of the address are delimited by ',’.
4. **Credit Card No**: 19 chars with format "XXXX-XXXX-XXXX-XXXX".

3.3 Order

The information of an order:

1. **Order ID**: 8 chars with format "XXXXXXXX"
2. **Customer ID**: non-empty string of maximum 10 chars
3. **Order Date**: Date – "YYYY-MM-DD". E.g. 2005-01-15
4. **Books Ordered**: a list of book order records, where each book order record contains the ISBN of the book (which has the 13 chars with format "X-XXXX-XXXX-X") and the quantity of the book ordered (which is a non-negative integer). (NOTE: This is a description of the books ordered. Before Phase 2, you do not need to

consider the format of “Books Ordered”, which will be described according to the suggested solution of Phase 1.) In the remaining of the specification, we adopt the term "Quantity" (in short) to describe the quantity of the book ordered in a book order record.

5. **Charge**: non-negative integer
6. **Shipping Status**: a char with format “C”, which has two possible values “Y” and “N”, where “Y” means it is shipped and “N” means it is not shipped.

4. Assumptions and Regulations

4.1 System

1. Numerical values will not be larger than the maximum integer value handled by Java.
2. The system is case-sensitive.

4.2 Book

1. “ISBN” is unique for each book. (NOTE: Two copies of the same book have the same ISBN.)
2. Different books can have the same book title, the same authors, the same unit price or the same no of copies available. .
3. “Book Title”, “Authors”, “Unit Price” and “No of Copies Available” with the same “ISBN” are the same.
4. There are no “%” and “_” characters in “Book Title” and “Authors”.
5. Each book should have at least one author.
6. Some books may not be ordered by any customers.
7. Each book can be involved in different book orders.
8. Any author in the database should write at least one book (with other authors or independently).
9. Each book is written by authors with different author names (if there is more than one author).

4.3 Customer

1. “Customer ID” is unique for each customer.
2. Different customers can have the same name, the same shipping address or the same credit card no.
3. “Customer Name”, “Shipping Address” and “Credit Card No” with the same “Customer ID” are the same.
4. There are no “%” and “_” characters in “Customer Name” and “Shipping Address”.
5. Some customers may not make any book order.
6. Each customer can make many book orders.

4.4 Order

1. "Order ID" is unique for each order.
2. "Customer ID", "Order Date", "Books Ordered", "Charge" and "Shipping Status" with the same "Order ID" are the same.
3. Each order is ordered by exactly one customer.
4. Each order should contain at least one book ordered (even though the quantity ordered is 0.)
5. "Order ID" and "ISBN" (in "Books Ordered") can uniquely determine "Quantity".
6. "Order ID" starts from "00000000" and is assigned in increasing numerical order (incremented by 1 each time). For example, the "Order ID", which follows "00000000", is "00000001". No "Order ID" can be reused.
7. If an order is altered at a later date, the date of the order is updated to the later date. Otherwise, the original date of the order should be kept.
8. "Customer ID" in the order should be the customer ID of an existing customer in the database.
9. "ISBN" in "Books Ordered" of an order should be the ISBN of an existing book recorded in the database.
10. If the order contains at least one book ordered with quantity greater than or equal to 1, then "Charge" is calculated by the following formula.

    Charge = Total Book Price + Shipping Price

    where

    Total Book Price is the sum of the product of the unit price multiplied by the no of copies (ordered) for all books ordered in this order.

    Shipping Price is calculated by the following formula.

    Shipping Price =

    No of Copies of All Books Ordered x Unit Shipping Charge

    + Handling Charge

    where the unit shipping charge is $10 and the handling charge is $10.

    If the order does not contain books ordered with quantity greater than or equal to 1 (i.e. all books ordered with quantity 0), then "Charge" is equal to 0.

4.5 Data Set

The data set will be given after the deadline of Phase 1. The details will be posted in the course homepage at that time.

5. System Interface Requirements

The system interface should be implemented in text (console) mode, ~~or using Java GUI tools as stated in the bonus part~~. Java Applet will NOT be accepted in this project. The following sections described the functionalities of the system.

The date to be displayed should be in the format YYYY-MM-DD.
The month to be displayed should be in the format YYYY-MM.

In the system, you have to let the user *choose* and *change* one of three interfaces – System Interface, Customer Interface and Bookstore Interface.

```
The System Date is now: 0000-00-00
<This is the Book Ordering System.>
-------------------------------------------
1. System interface.
2. Customer interface.
3. Bookstore interface.
4. Show System Date.
5. Quit the system......

Please enter your choice??..1
```

Now, the functions of each interface are described.

5.1 System Interface

In the System Interface, there are the following functions.

```
<This is the system interface.>
-------------------------------
1. Create Table.
2. Delete Table.
3. Insert Data.
4. Set System Date.
5. Back to main menu.

Pleae enter your choice??..2
```

1. **Create table schemas in the database**
   Allow users to create all table schemas in the database based on the given relational schema.

2. **Delete table schemas in the database**
   Allow users to delete all table schemas in the database based on the given relational schema.

3. **Insert data to the database**
   Allow users to load the system with data. The system should let the user enter the path of the folder containing the data files, then read each data file and load it into a table in the database.

```
<This is the system interface.>
------------------------------
1. Create Table.
2. Delete Table.
3. Insert Data.
4. Set System Date.
5. Back to main menu.

Pleae enter your choice??..3
Please enter the folder path
/uac/gds/xbzheng/BOS/test_data
Processing...Data is loaded!
```

4. **System Date Setting**

   You are required to provide an interface for setting the system date of this application. This system date should be set a date which is the latest "Order Date" in all orders or a later date. It is noted that, during demonstration, the system date is set forward. In other words, the system date will not be set backward. For example, if we set the system date "2020-09-11" previously, we may set a date "2021-09-11" or later.

   For sake of the demonstration, the system will have its own date to indicate a particular instance. However, please remember that a real system date (e.g. system date in Oracle) should be used in real life.

```
<This is the system interface.>
------------------------------
1. Create Table.
2. Delete Table.
3. Insert Data.
4. Set System Date.
5. Back to main menu.

Pleae enter your choice??..4
Please Input the date (YYYYMMDD): 20210101
Latest date in orders: 1899-12-31
Today is 2021-01-01
```

5.2 Customer Interface

In the Customer Interface, there are the following functions.

```
<This is the customer interface.>
---------------------------------
1. Book Search.
2. Order Creation.
3. Order Altering.
4. Order Query.
5. Back to main menu.

What is your choice??..
```

1. **Book Search**

   to query a book by ISBN, Book Title and Author Name

```
What do u want to search??
1 ISBN
2 Book Title
3 Author Name
4 Exit
Your choice?...
```

a. **Query by ISBN**
Input: ISBN

b. **Query by Book Title**
Input: Book Title
(Partial book title is supported. Both the wild cards "%" and "_" should be supported. In the results, exact matches (if any) should be displayed first. E.g. if the input is "ABC%", then "ABC Book" may be the output. The meaning of exact matches is illustrated as follows. If the input is "%ABC%", "%ABC" or "ABC%", then the exact match of this input is "ABC". For simplicity, we assume that there is no exact match if the wild card "%" occurs in the middle of the input (e.g. "A%BC", "%A%BC" and "%A%BC%").)

c. Query by Author Name
Input: Author Name
(Partial author name is supported. The description is similar as above. Note that the above input refers to a *single* author. This author is not restricted to the first author of the book. For example, this author may be the first author. He/she may be the second author.)

For each of the above query, the output and the order are described as follows.
Output: "Book Title", "ISBN", "Unit Price", "No of Copies Available" and "A List of Authors"
Order: Results should be sorted in ascending order by "Book Title" and then in ascending order by "ISBN". In each book, the author should be sorted in ascending order by "Author Name".

```
What do u want to search??
1 ISBN
2 Book Title
3 Author Name
4 Exit
Your choice?...1
Input the ISBN: 1-1234-1234-1

Record 1
ISBN: 1-1234-1234-1
Book Title:Database I
Unit Price:100
No Of Available:50
Authors:
1 :Ada
2 :Raymond
3 :Willy
Operation not allowed after ResultSet closed
cannot query the book
```

2. **Order Creation**
to create a new order
Input: Customer ID
Books to be Ordered (i.e. a list of books with quantities to be ordered)
Action: check if the book copies are available ("No of Copies Available" greater than or equal to the requested number). Then create an order and insert it into the database. "Order ID" to be assigned is equal to one plus the current greatest Order ID stored in the database. "Order Date" to be assigned is equal to the system date. "Shipping Status" is initialized to "N".

(NOTE: “Order ID”, “Order Date”, “Charge” and “Shipping Status” are assigned automatically. This action should create an order with at least one book to be ordered with quantity at least 1.)

```
<This is the customer interface.>
--------------------------------
1. Book Search.
2. Order Creation.
3. Order Altering.
4. Order Query.
5. Back to main menu.

What is your choice??..2
Please enter your customerID??cwwong
>> What books do you want to order??
>> Input ISBN and then the quantity.
>> You can press "L" to see ordered list, or "F" to finish ordering.
Please enter the book's ISBN: L
ISBN          Number:
Please enter the book's ISBN: 4-4567-4567-4
Please enter the quantity of the order: 2
Please enter the book's ISBN: L
ISBN          Number:
4-4567-4567-4   2
```

3. **Order Altering**

to add or drop copies of books to or from an order

Input: Order ID

Action: After Order ID is inputted, a list of books ordered in the order will be displayed. Allow the user to select one of them. Also, allow the user to choose one of the following actions.

```
<This is the customer interface.>
--------------------------------
1. Book Search.
2. Order Creation.
3. Order Altering.
4. Order Query.
5. Back to main menu.

What is your choice??..3
Please enter the OrderID that you want to change: 00000002
order_id:00000002  shipping:N  charge=120  customerId=hyyue
book no: 1 ISBN = 1-1234-1234-1 quantity = 1
Which book you want to alter (input book no.):
1
```

a. **Add Copies of Book**

Input: No of Copies to be Added

Action: to add one or more copied (specified by the user) of a book in the current order. If “Shipping Status” is “N” and the no of copies to be added is smaller than or equal to the “No of Copies Available” of the book, then this operation is successful, the “Quantity” in the order is incremented by the requested amount and the “No of Copies Available” is decremented by the requested amount. Otherwise, return the correspondence error message such as “The books in the order are shipped” or other relevant messages.

```
<This is the customer interface.>
--------------------------------
1. Book Search.
2. Order Creation.
3. Order Altering.
4. Order Query.
5. Back to main menu.

What is your choice??..3
Please enter the OrderID that you want to change: 00000001
```

b. **Remove Copies of Book**

Input: No of Copies to be Removed

Action: to remove one or more copies (specified by the user) of a book in the current order. If "Shipping Status" is "N", then this operation is successful, the "Quantity" is decremented by the request amount and the "No of Copies Available" is incremented by the request amount. Otherwise, return the correspondence error message such as "The books in the order are shipped" or other relevant messages. (NOTE: Even though "Quantity" is decremented to 0, we do not need to remove the record of book ordered, which can allow the customer to add copies of the book later.)

```
<This is the customer interface.>
--------------------------------
1. Book Search.
2. Order Creation.
3. Order Altering.
4. Order Query.
5. Back to main menu.

What is your choice??..3
Please enter the OrderID that you want to change: 00000002
order_id:00000002  shipping:N  charge=120  customerId=hyyue
book no: 1 ISBN = 1-1234-1234-1 quantity = 1
Which book you want to alter (input book no.):
1
input add or remove
add
Input the number: 2
Update is ok!
update done!!
updated charge
order_id:00000002  shipping:N  charge=340  customerId=hyyue
book no: 1 ISBN = 1-1234-1234-1 quantity = 3
```

Each of the above operations should update "Order Date" to be the current system date. (NOTE: Please remember to update "Charge" in the order. In this project, we assume the user will not add or delete any book in the order.)

4. **Order Query**

to query a list of orders made by a customer in a particular year.

Input: Customer ID
Year

Output: "Order ID", "Order Date", "Books Ordered", "Charge" and "Shipping Status"

Order: Results should be sorted in ascending order by "Order ID".

```
<This is the customer interface.>
---------------------------------
1. Book Search.
2. Order Creation.
3. Order Altering.
4. Order Query.
5. Back to main menu.

What is your choice??..4
Please Input Customer ID: hndai
Please Input the Year: 2005

Record : 1
OrderID : 00000004
OrderDate : 2005-09-12
charge : 120
shipping status : N

Record : 2
OrderID : 00000005
OrderDate : 2005-09-20
charge : 120
shipping status : N
```

5.3 Bookstore Interface

In the Bookstore Interface, there are the following functions.

1. **Order Update**

   to update the shipping status of an order

   Input: "Order ID"

   Action: to update the shipping status of an order. If the original shipping status is "N", then check whether the order contains at least one book with quantity greater than or equal to 1. If no, prompt the user. If yes, we should ask the users whether the shipping status will be updated to "Y". If the user wants to update the status, then do it accordingly. Otherwise, quit the update and return to the Bookstore Interface. If the original shipping status is "Y", no update is allowed.

```
<This is the bookstore interface.>
---------------------------------
1. Order Update.
2. Order Query.
3. N most Popular Book Query.
4. Back to main menu.

What is your choice??..1
Please input the order ID: 00000003
the Shipping status of 00000003 is N and 1 books ordered
Are you sure to update the shipping status? (Yes=Y) Y
Updated shiping status
```

2. **Order Query**

   to query the total charges of all orders with shipping status "Y" in a certain month

   Input: Month (Note that the format is YYYY-MM)

   Output: Total Charge in this month (i.e. the summation of the charges of all orders with shipping status "Y" in this month) and a list of the orders with shipping status "Y" in this month (NOTE: The order should show "Order ID", "Customer ID", "Order Date" and "Charge")

Order: The list of orders should be sorted in ascending order by "Order ID".

```
<This is the bookstore interface.>
----------------------------------
1. Order Update.
2. Order Query.
3. N most Popular Book Query.
4. Back to main menu.

What is your choice??..2
Please input the Month for Order Query (e.g.2005-09): 2005-09


Record : 1
order_id : 00000001
customer_id : cwwong
date : 2005-09-02
chage : 120


Record : 2
order_id : 00000003
customer_id : hyyue
date : 2005-09-10
chage : 120


Total charges of the month is 240
```

3. ***N* Most Popular Book Query**

to find the *N* most popular books with the maximum total number of ordered copies (regardless of the shipping status and the order date in the orders). Formally, we should return a list of books with the total number of ordered copies greater than or equal to no_N, where no_N is the *N*-th greatest total number of ordered copies. For instance, suppose *N*=2 and there are four books, namely book1, book2, book3 and book4. If the total number of ordered copies of book1, book2, book3 and book4 are 5, 4, 2 and 1, respectively, then the top *N* results contain book1 and book2. If the total number of ordered copies of book1, book2, book3 and book4 are 5, 4, 4 and 1, respectively, then the top *N* results contain book1, book2 and book3. If the total number of ordered copies of book1, book2, book3 and book4 are 5, 5, 2 and 1, respectively, then the top *N* results contain book1 and book2. If *N* is greater than the total number of books (or the total number of books ordered with quantity greater than or equal to 1), the result should contain all of the books ordered with quantity greater than or equal to 1 (regardless of the shipping status in the orders).

Input: *N*

Output: "Book Title", "ISBN" and the total number of ordered copies

Orders: The results should be sorted in descending order by the total number of ordered copies and then in ascending order by the "Book Title" and "ISBN".

```
<This is the bookstore interface.>
---------------------------------
1. Order Update.
2. Order Query.
3. N most Popular Book Query.
4. Back to main menu.

What is your choice??..3
Please input the N popular books number: 3
ISBN            Title              copies
1-1234-1234-1   Database I  8
2-2345-2345-2   Database II  3
4-4567-4567-4   Programming in C language  1
5-5678-5678-5   Programming in Java language  1
```

5.4. Error Handling

The system should be able to handle errors such as runtime error, data not found, incorrect format, incorrect content, etc. and display error messages.

6. Grading Policy

The mark distribution is as follows:

Phase	Content	Mark Distribution
1	ER-diagram	10%
	Relational schema (based on your ER-diagram)	10%
2	Java application	80%

- There will be a mark deduction if your application is terminated unexpectedly during the demonstration.
- You are not allowed to modify any source code during the demonstration.
- Every member of the same group will receive the same marks for the project. To encourage every student to participate in the project, a question about this project may be asked in the final examination.

7. Demonstration

- We will hold a face-to-face demonstration or online demonstration.
- Every group member should attend the demonstration.
- Each demonstration will last for about **15** minutes.
- We will **compile** and **test** your system on a Linux 64-bit machine of the CSE department.
- The data files used in the demonstration may be different from the data files provided for testing.

8. Submission Methods

8.1. Phase 1

- Submit a PDF file (one copy for each group) to the Blackboard system.
- The PDF file should include an ER diagram, a relational schema, the group number, the names, and the student IDs of all group members of your group.

8.2. Phase 2

- Submit a ZIP file (one copy for each group) to the Blackboard system. The ZIP file should include all your source codes and a README file, which includes:
  - Your group number
  - The name and the student ID of each group member
  - Instructions on how to compile and run your system