

COMP 4003A 2024W

Assignment #1

Due: Feb 9 @11:59pm

Instruction

1. You should do the assignment independently. If copying is found, the case will be reported to the office of the Dean of Science immediately.
2. You need to use [Oracle VM](#) to do this assignment and take proper screenshots of execution results for the relevant questions. If there is **no screenshot**, you will get **0** for the question.
3. First replace **Last** below with your last name. If your last name is not showing in the screenshot, you will get a **0** for the assignment. Also, rename this document with your last name+first name.
4. Do the assignment directly on this document by copying your programs and screenshots in this document, and submit it to brightspace. Make sure your uploaded file can be opened and is correct. No submission will be accepted after the deadline no matter what reason.

Part 1 (40)

Use dynamic SQL method 4 to write an Embedded SQL program that allows the user to manipulate the database you have created in Assignment#1 with the following command line menu:

- | | |
|---|--|
| 1. Add a customer | 2. Display a customer |
| 3. Add a product | 4. Display a product |
| 5. Add a line_item | 6. Add an order of line_items |
| 7. Add a customer order | 8. List all orders of a given customer |
| 9. List customers who ordered the given product | 10. Drop all tables |

When the user selects an option, prompts the user to enter additional information if needed.

Use this program to populate the database with five customers: Smith, Jones, Blake, Clark, and **MacDiarmid**; five products: Apple, Banana, Orange, Peach, and Watermelon; Smith orders the *first* product, Jones *first two* products, ..., and **MacDiarmid** orders everything. The program should first find the highest C#, O#, I#, or P# and generate a new number by adding 1 to it, and also check if the customer or product to be added exists; i.e., customer names and product names are unique. Use **MacDiarmid** for option 8 and Apple for option 9.

For the context of the screen shots, whenever there is a customer id that is equal to 5, it is the customer id of MacDiarmid. You can check within the database itself to verify, but just making a statement at the beginning so screen shots are clear.

```
[Enter your choice: 1
[Enter a customer name: MacDiarmid
[Enter a customer address: myaddress
Customer name already exists
[Enter a customer name: nathan
[Enter a customer address: myaddress
Customer added successfully.
```

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

```
[Enter your choice: 11
```

CUSTOMERS

C#	NAME	Address
1	Smith	1121 Colonel By Dr
2	Jones	1122 Colonel By Dr
3	Blake	1123 Colonel By Dr
4	Clark	1124 Colonel By Dr
5	MacDiarmid	1125 Colonel By Dr
6	nathan	myaddress

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

[Enter your choice: 2

[Enter a customer name: MacDiarmid

C#	NAME	Address
5	MacDiarmid	1125 Colonel By Dr

```
[Enter your choice: 3
[Enter a product name: apple
[Enter a product price: 1.25
Product name already exists
[Enter a product name: pineapple
[Enter a product price: 5.13
Product added successfully.
```

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

```
[Enter your choice: 12
```

PRODUCTS

P#	Name	Price
1	apple	1.25
2	banana	0.79
3	orange	1.30
4	peach	1.79
5	watermelon	7.99
6	pineapple	5.13

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

[Enter your choice: 4

[Enter a product name: apple

P#	Product Name	Price
1	apple	1.25


```

[Enter your choice: 5
[Enter a product id: 77
Product id not found
[Enter a product id: 3
[Enter a order id: 66
Order id not found
[Enter a order id: 3
[Enter a quantity: 24
Line Item added successfully.

```

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

```

[Enter your choice: 14

```

LINE ITEMS

I#	Product Id	Order id	Qty
1	1	1	1
2	1	2	1
3	2	2	1
4	1	3	1
5	2	3	1
6	3	3	1
7	1	4	1
8	2	4	1
9	3	4	1
10	4	4	1
11	1	5	1
12	2	5	1
13	3	5	1
14	4	5	1
15	5	5	1
16	1	6	2
17	2	6	1
18	3	6	1
19	4	6	1
20	5	6	1
21	3	3	24
22	3	3	24

```
[Enter your choice: 6
[Enter a customer id: 99
Customer id does not exists
[Enter a customer id: 5
[Enter a order date: Feb
[Enter a product id: 88
Product id not found
[Enter a product id: 3
[Enter a quantity: 24
[More items? 0 - more 1 - done: 0
[Enter a product id: 5
[Enter a quantity: 44
[More items? 0 - more 1 - done: 1
Order added successfully.
```

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

```
[Enter your choice: 13
```

ORDERS

O#	Customer ID	Date	Total Price
1	1	Jan 1 2024	1.25
2	2	Jan 2 2024	2.04
3	3	Jan 3 2024	3.34
4	4	Jan 4 2024	5.13
5	5	Jan 5 2024	13.12
6	5	Jan 6 2024	14.37
7	5	Feb	382.76

```
[Enter your choice: 7
[Enter a customer name: MacDiarmid
[Enter a customer address: myaddress
Customer name already exists
[Enter a customer name: nathan
[Enter a customer address: myaddress
Customer added successfully.
[Enter a order date: Feb20
[Enter a product id: 9
Product id not found
[Enter a product id: 4
[Enter a quantity: 34
[More items? 0 - more 1 - done: 0
[Enter a product id: 3
[Enter a quantity: 24
[More items? 0 - more 1 - done: 1
Order added successfully.
```

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

```
[Enter your choice: 11
```

CUSTOMERS

C#	NAME	Address
1	Smith	1121 Colonel By Dr
2	Jones	1122 Colonel By Dr
3	Blake	1123 Colonel By Dr
4	Clark	1124 Colonel By Dr
5	MacDiarmid	1125 Colonel By Dr
6	nathan	myaddress

[Enter your choice: 11

CUSTOMERS

C#	NAME	Address
1	Smith	1121 Colonel By Dr
2	Jones	1122 Colonel By Dr
3	Blake	1123 Colonel By Dr
4	Clark	1124 Colonel By Dr
5	MacDiarmid	1125 Colonel By Dr
6	nathan	myaddress

Command Line Menu:

1. Add a customer
2. Display a customer
3. Add a product
4. Display a product
5. Add a line item
6. Add an order of line items
7. Add a customer order
8. List all orders of a given customer
9. List customers who ordered the given product
10. Drop all tables
11. Display all customers
12. Display all products
13. Display all orders
14. Display all line_items
0. Exit

[Enter your choice: 13

ORDERS

O#	Customer ID	Date	Total Price
1	1	Jan 1 2024	1.25
2	2	Jan 2 2024	2.04
3	3	Jan 3 2024	3.34
4	4	Jan 4 2024	5.13
5	5	Jan 5 2024	13.12
6	5	Jan 6 2024	14.37
7	5	Feb	382.76
8	6	Feb20	92.06

[Enter your choice: 8
[Enter a customer name: MacDiarmid

O#	Customer name	Order date	Total Price
5	MacDiarmid	Jan 5 2024	13.12
6	MacDiarmid	Jan 6 2024	14.37
7	MacDiarmid	Feb	382.76

[Enter your choice: 9
[Enter a customer name: apple

C#	Customer name	Customer address	Product
2	Jones	1122 Colonel By Dr	apple
5	MacDiarmid	1125 Colonel By Dr	apple
1	Smith	1121 Colonel By Dr	apple
3	Blake	1123 Colonel By Dr	apple
4	Clark	1124 Colonel By Dr	apple

[Enter your choice: 10

Table dropped

Table dropped

Table dropped

Table dropped

Command Line Menu:

1. Add a customer

2. Display a customer

3. Add a product

4. Display a product

5. Add a line item

6. Add an order of line items

7. Add a customer order

8. List all orders of a given customer

9. List customers who ordered the given product

10. Drop all tables

11. Display all customers

12. Display all products

13. Display all orders

14. Display all line_items

0. Exit

[Enter your choice: 11

CUSTOMERS

C#	NAME	Address
----	------	---------

Command Line Menu:

1. Add a customer

2. Display a customer

3. Add a product

4. Display a product

5. Add a line item

6. Add an order of line items

7. Add a customer order

8. List all orders of a given customer

9. List customers who ordered the given product

10. Drop all tables

11. Display all customers

12. Display all products

13. Display all orders

14. Display all line_items

0. Exit

Enter your choice: █

Part 2 (60)

1. Create a PL/SQL package called CPO that contains procedures and/or functions with the corresponding parameters for all the menu options listed in Part 1. It should raise an exception when the value provided are unexpected and display proper error messages. (30)
2. Use dynamic SQL method 4 to rewrite an Embedded PL/SQL program that allows the user to choose a menu option and also provide the values needed, and then call the corresponding PL/SQL procedure or function to do the job and test all options. (30)