COMP 4003A 2024W
Assignment #1
Due: Jan 27@11:59pm

Nathan MacDiarmid

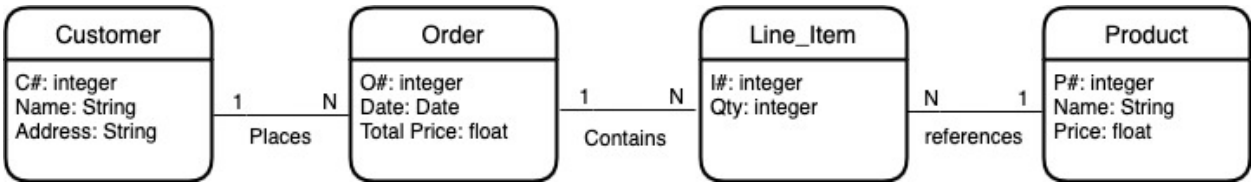
101098993

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 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.

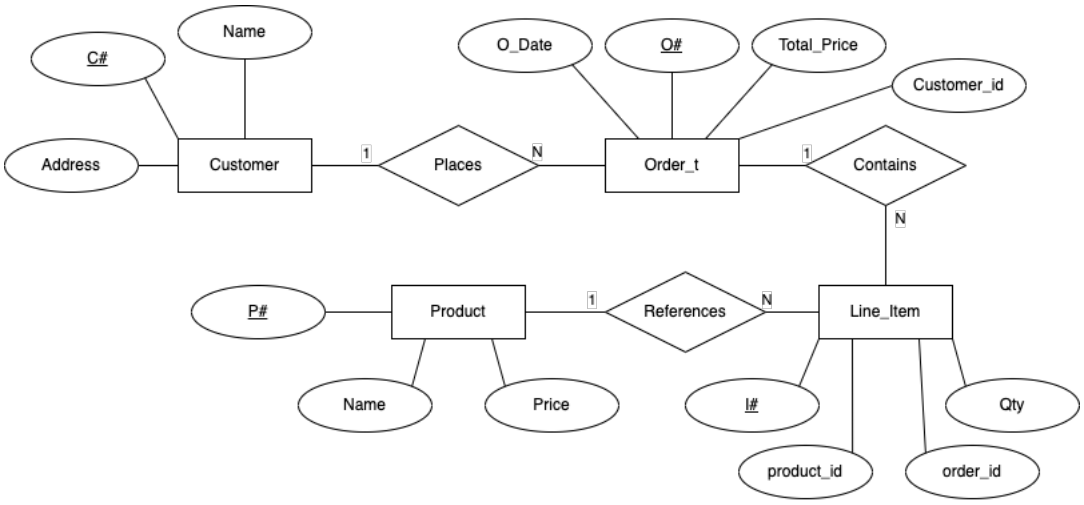
Application Description

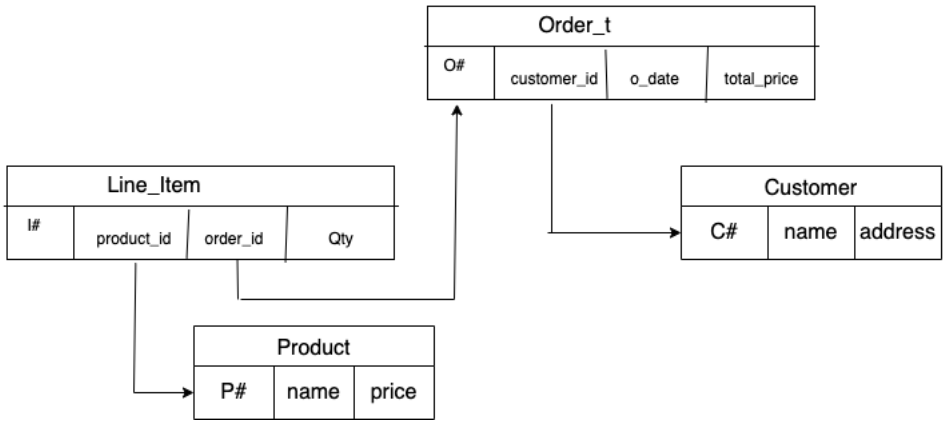
Given the ER model for Customer Purchase Order application as follows.



A Customer has a one-to-many relationship with an Order because a customer can place many orders, but a given purchase order can be placed by only one customer. An Order has a one-to-many relationship with a Line Item because an order can list many line items, but a given line item can be listed by only one purchase order. A line item has a many-to-one relationship with a Product because a line item can refer to only one product, but a given product can be referred to by many line items.

1. Use ER mapping rules learned in COMP3005 to create a relational database for this application by giving the relation names, their attributes and types, primary keys underlined, and foreign keys pointing to the corresponding primary keys. (15)



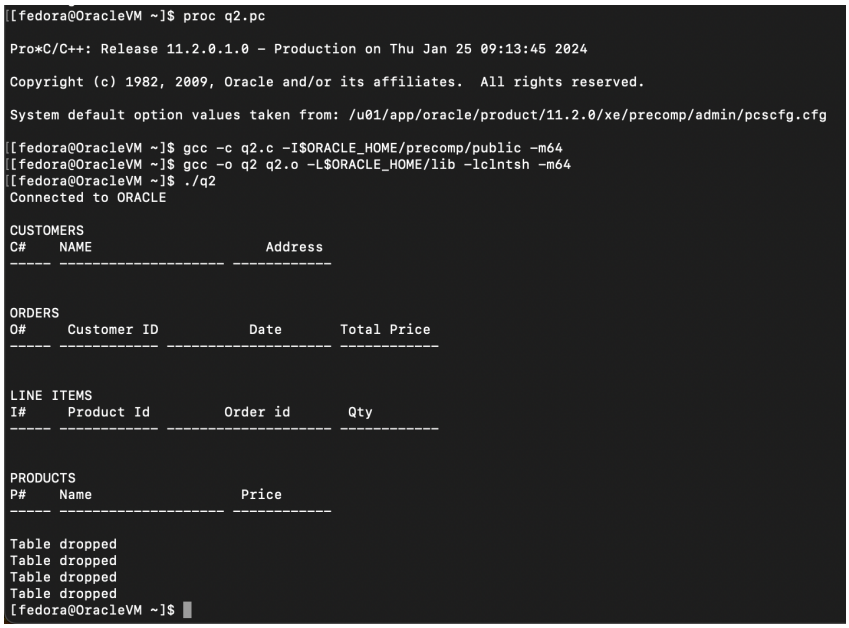


2. Use dynamic SQL method 1 to create the database for this application. You must use execute immediate statement to properly define primary keys and foreign keys of the relations. (15)

This is the code snippet creating the database tables using dynamic SQL method 1. The rest of the code can be found in Q2.pc.

```
strcpy(customer_table, "create table customer (C# int primary key, Name varchar(10), Address varchar(25))");

strcpy(order_table, "create table order_t (O# int primary key, Customer_id int, O_Date varchar(10), Total_Price varchar(10))");

strcpy(line_item_table, "create table line_item (I# int primary key, Product_id int, Order_id int, Qty int)");

strcpy(product_table, "create table product (P# int primary key, Name varchar(10), Price varchar(10))");

exec sql set transaction read write;

exec sql execute immediate :customer_table;
exec sql execute immediate :order_table;
exec sql execute immediate :line_item_table;
exec sql execute immediate :product_table;

exec sql commit release;
```

I have also attached a screenshot of the results when running Q2.pc on the virtual machine.



3. Use dynamic SQL method 2 to populate this database with five customers: Smith, Jones, Blake, Clark, and MacDiarmid; five products: apple, banana, orange, peach, and watermelon; Smith orders 1 product, Blake 2, …, and MacDiarmid orders everything. (20)

This code snippet is an example of how I inserted data into the customer table using dynamic SQL method 2. The rest of the inserts and program can be found in Q3.pc.

```
exec sql set transaction read write;

strcpy(customer, "insert into customer values (:v1, :v2, :v3)");

exec sql prepare s from :customer;

strcpy(no, "5");
strcpy(first, "MacDiarmid");
strcpy(second, "1125 Colonel By Dr");

exec sql execute s using :no, :first, :second;

exec sql commit release;
```

I have also attached a screenshot of the results when running Q3.pc on the virtual machine.

```
[[fedora@OracleVM ~]$ gcc -o q3 q3.o -L$ORACLE_HOME/lib -lclntsh -m64
[[fedora@OracleVM ~]$ ./q3
Connected to ORACLE

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

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

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

PRODUCTS
P#    Name                  Price
----- -------------------- ------------
 1    apple                1.25
 2    banana               0.79
 3    orange               1.30
 4    peach                1.79
 5    watermelon           7.99
Table dropped
Table dropped
Table dropped
Table dropped
[fedora@OracleVM ~]$
```

4. Use dynamic SQL method 3 to prompt the user to enter a customer name and/or a product name. For a given customer name, generate a list of complete sale orders placed by the customer. For a given product name, generate the list of customer names and the date of the purchase. If both are given, then display the date and quantity of the purchase. Test all three cases involving MacDiarmid. (50)

This code snippet is an example of how I queried data given a customer name using dynamic SQL method 3. The rest of the inserts and program can be found in Q4.pc.

```
exec sql set transaction read write;

strcpy(sqlstmt, "select order_t.O#, customer.Name, order_t.o_date, order_t.Total_Price from
order_t inner join customer on order_t.customer_id = customer.C# where customer.name =
:v1");

exec sql prepare t from :sqlstmt;

exec sql declare a_cursor cursor for t;

printf("\nGet a list of all orders placed by customer\n");
printf("Enter a customer name: ");
scanf("%s", &first);

exec sql open a_cursor using :first;

printf("\nO#   Customer name      Order date       Total Price \n");
printf("----- ------------ -------------------- ------------\n");

exec sql fetch a_cursor into :no, :first, :second, :third;
while(sqlca.sqlcode==0) {
  printf("%4s  %10s %10s %10s\n", no, first, second, third);
  exec sql fetch a_cursor into :no, :first, :second, :third;
}

exec sql commit release;
```

This code snippet is an example of how I queried data given a product name using dynamic SQL method 3. The rest of the inserts and program can be found in Q4.pc.

```
exec sql set transaction read write;

strcpy(sqlstmt, "SELECT customer.name, order_t.o_date FROM customer INNER JOIN order_t
ON customer.C# = order_t.customer_id INNER JOIN line_item ON order_t.O# =
line_item.order_id INNER JOIN product ON line_item.product_id = product.P# WHERE
product.name = :v1");

exec sql prepare t from :sqlstmt;

exec sql declare another_cursor cursor for t;

printf("\nGet a list of all orders placed by customer containing this product\n");
printf("Enter a product name: ");
scanf("%s", &first);

exec sql open another_cursor using :first;

printf("\nCustomer name       Date ordered \n");
printf("-------------------- ------------\n");

exec sql fetch another_cursor into :first, :second;
while(sqlca.sqlcode==0) {
  printf("%10s %10s\n", first, second);
  exec sql fetch another_cursor into :first, :second;
}

exec sql commit release;
```

This code snippet is an example of how I queried data given a customer and product name using dynamic SQL method 3. The rest of the inserts and program can be found in Q4.pc.

```
exec sql set transaction read write;

strcpy(sqlstmt, "SELECT order_t.o_date, line_item.qty FROM customer INNER JOIN order_t ON
customer.C# = order_t.customer_id INNER JOIN line_item ON order_t.O# = line_item.order_id
INNER JOIN product ON line_item.product_id = product.p# WHERE customer.Name = :v1 AND
product.name = :v2");

exec sql prepare t from :sqlstmt;

exec sql declare another_more_cursor cursor for t;

printf("\nGet a list of orders containing the quantity of a product\n");
printf("Enter a customer name: ");
scanf("%s", &first);

printf("Enter a product name: ");
scanf("%s", &second);

exec sql open another_more_cursor using :first, :second;

printf("\nDate ordered         Amount of product \n");
printf("-------------------- ------------\n");

exec sql fetch another_more_cursor into :first, :second;
while(sqlca.sqlcode==0) {
  printf("%10s %10s\n", first, second);
  exec sql fetch another_more_cursor into :first, :second;
}

exec sql commit release;
```

I have also attached a screenshot of the results when running Q4.pc on the virtual machine. This is the result from all three queries.

```
[fedora@OracleVM ~]$ proc q4.pc

Pro*C/C++: Release 11.2.0.1.0 - Production on Thu Jan 25 09:38:19 2024

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.

System default option values taken from: /u01/app/oracle/product/11.2.0/xe/precomp/admin/pcscfg.cfg

[fedora@OracleVM ~]$ gcc -c q4.c -I$ORACLE_HOME/precomp/public -m64
[fedora@OracleVM ~]$ gcc -o q4 q4.o -L$ORACLE_HOME/lib -lclntsh -m64
[fedora@OracleVM ~]$ ./q4
Connected to ORACLE

Get a list of all orders placed by customer
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

Get a list of all orders placed by customer containing this product
Enter a product name: apple

Customer name        Date ordered
-------------------- ------------
Smith                Jan 1 2024
Jones                Jan 2 2024
Blake                Jan 3 2024
Clark                Jan 4 2024
MacDiarmid           Jan 6 2024
MacDiarmid           Jan 5 2024

Get a list of orders containing the quantity of a product
Enter a customer name: MacDiarmid
Enter a product name: apple

Date ordered          Amount of product
-------------------- ------------
Jan 5 2024           1
Jan 6 2024           2
Table dropped
Table dropped
Table dropped
Table dropped
[fedora@OracleVM ~]$
```