Getting Started

In this lab you will use Oracle to create and query a simple SQL database.

- When you have successfully connected to the Oracle server, you will see the following sqlplus prompt:
  ```
  SQL>
  ```
- You can now start typing in sqlplus commands or SQL statements.
  (Be sure to terminate every SQL statement with semicolon ";").
- Using your favorite editor, create a file named **`lab1.sql`**; this will be the only file you submit.
- To run all the code in your lab1.sql file, use the following command at the sqlplus prompt:
  ```
  SQL> @lab1
  ```

Special Requirements (yes, these are worth 5 extra points):

- Include your name in a header comment at the top of your source file.
- Comment your code; if nothing else, mark each query with its number.
- Make sure your source file can be executed without errors.

Now you are assigned to help the bookstore to maintain shopping records by database techniques. In this section, you need to create five tables in Oracle, including any appropriate primary/foreign key constraints. The schemas of the tables are as follows.

Customer(C_Id: number, C_Name: string, Credit_Limit:number, Income_Level:char, Gender:char)

C_Id is the customer id, and C_name is the name of the customer. There are three possible income levels recorded as: Low(L), Median(M), High(H). In attribute *Gender*, we use *F* for female, and *M* for male.

Book(Book_Id: number, Book_name: string, Warehouse_Id: number, Quantity_On_Hand: number, Warranty_Period: number, Purchase _Price: numer)

Book_Id identifies the book, and Book_name is the name of the book. Warehouse_Id is the id of the warehouse where this book is stored. Quantity_On_Hand records the quantity of the book in stock. Warranty_Period is represented by the number of days. Here we assume that the warranty is calculated from the date when the order is placed. Purchase_Price is the price that the bookstore paid to buy the book from the factory.

Orders(Order_Id: number, Order_Data: date, C_Id: number, Order_Status: string)

There are two possible statuses of Order_Status: Complete (C) and Processing (P). A “Processing” order means that the order is placed but the book has not yet been shipped to the customer.

OrderItems(Order_Id: number, Book_Id: number, Unit_Price: number, Quantity: number)

Table OrderItems records the books for each order. Here, Unit_Price is the price paid by the customers for one book.

WareHouse(Warehouse_Id:number, Location:string)

Warehouse_Id is the id of the warehouse, and *Location* records the location of the warehouse.

Table1. Customers

C_Id	C_Name	Credit_Limit	Income_Level	Gender
1	Jone	140	L	F
2	Chris	230	M	M
3	Saywer	480	H	F
4	Kropy	500	H	M
5	Lucy	220	M	M
6	Mando	100	L	F
7	Bunny	300	M	F

Table2. Book

Book_Id	Book_name	Warehouse_ Id	Quantity_ On_Hand	Warranty_ Period	Purchased_ Price
1	Life with dog	1	18	90	6
2	Inferno	1	25	180	8
3	Doctor sleep	3	9	365	10
4	Disappear	2	60	30	15
5	Six years	2	50	365	7
6	The lowland	1	5	120	25
7	Wave	3	11	60	20
8	Lost world	2	20	30	15
9	Whiskey beach	3	33	150	10

Table3. Orders

Order_Id	Order_Date	C_Id	Order_Status
1	2016-08-01	1	P
2	2016-08-27	2	C
3	2016-06-20	3	C
4	2016-08-01	4	C
5	2016-08-31	1	P
6	2016-09-01	4	P
7	2016-07-20	6	C
8	2016-08-11	2	C

Table4. Order_Items

Order_Id	Book_Id	Unit_Price	Quantity
1	1	19	2
1	2	20	1
2	1	17	1
3	4	20	2
3	2	25	3
3	8	16	1
4	4	21	10
5	2	10	2
5	8	28	1
6	9	16	10
7	5	12	3
7	7	25	1
8	4	30	2

Table5. WareHouse

Warehouse_Id	Location	Quantity_In_Stock
1	Los Angles	1100
2	Chicago	800
3	New York	700

Query:

1) Count the number of male and female customers.

2) Return the name/names of the customers whose income level are Median (M); and calculate the average credit limit of these median-income customers.

3) Check the complete orders and find the name/names of the customers who have an income level of "H"

4) Return the name of the customer who has placed more orders than any other customer. (Including both processing and complete orders)

5) Return the name/names of the customers who have ever placed an order/orders containing at least *three* items and their income level is *not high*.

6) Calculate the sale revenue of all complete orders; calculate the profit for all complete orders. ** *Sale revenue* is defined by the sum of prices of all items in completed orders; *profit* is defined by the sale revenue minus the sum of purchasing price of all sold items.

7) Return the name of the customer who has processing order valued more than half of his/ her credit limit.

8) Return <order_id, Book_name, order_date, warranty_period> for all the sold books whose warranties are already expired and quantity on hand is greater than 10.

9) Return the name/names of the customers who have purchased at least *two* books in August 2016. (Including all processing and completed orders)

10) Find the names of customer/customers who have placed at least one order including items from at least 2 different warehouses.

11) Return the total number of male customers who did not place any order in August 2016.

12) Which book/books have not been ordered by high level income customers yet?

13) Find the item that has the largest selling volume for customers in different gender

14) Sort the orders by its value and return the sorted Order_Id.

15) Return the name of the book, which is ordered by more customers than any other book.

16) Calculate the average profit of each book; and find the most and least profitable book.

Update:

1. Decrease the credit limit of all female customers by 30%. Then if there exists any "processing" order that exceed the customer's credit limit, delete such orders.
2. After all the "processing" orders are completed; you first update their statuses to "completed" and then update the On_Hand_Quantity in Book table.
3. If customer "Kropy" requested to not list him in our system anymore, delete all the related information of this custormer in all these tables.
4. Increase the warranty period by 30 days for all the products stored in warehouse "Chicago". If total days more than 365, than set it as 365.

Congratulations! You have finished all your work and please include SQL code at the end of your script to ***drop*** all tables and/or views you've created.

Submission

You only need to submit the script 'lab1.sql' from Canvas system.

Before your final submission, please upload your script to our oracle server and test it by running @lab1 at sqlplus prompt. Make sure your script is executable and all your results will be printed out without any error; otherwise, you can only get at most 50% credits if your script is not runnable.