==============

Fall 2014

IS 380 Database
Semester Project

Return Authorization

==============

Project Description

Things ‘n Things is an online company that sells hundreds of various products on its e-Commerce store, each product has a product number, description, and unit price. When customers are unsatisfied with their products, they return the products to Things ‘n Things, and Things ‘n Things issues credit to the customer. Each customer has a customer number and profile (such as name, address, etc.). To return, the customer contacts our users at the Operations department and provides the product number, reason of the return, and how many units they are returning. A return authorization is now ‘O’pened for this return with a return authorization (RA) number. Customer then mails the products to our warehouse. Our warehouse notifies our Operations department when it receives the products. Our Operations department then issues credits to the customer, and the return authorization is now ‘C’losed.

In addition to the processing of returns and credits, our management also wants to know why customers return (by reason code), how many returns have not been closed, and so on an so forth.

Thing ‘n things have asked you to design a relational database (Entity Relationship Diagram and normalized tables) for the return process, and to complete the following five programs to process returns and produce reports for the management.

Program 1. OpenRA.sql

To open a return authorization (RA), the user would run the **OpenRA.sql** program. The system first prompts the greeting message and automatically displays today's date. The system then prompts the user to enter the customer number. Customer name (last name first, comma, then first name), address and phone are displayed. The system then prompts for the product number. Product description and unit price is displayed. The system then asks for the return quantity and calculates return amount. Return reason codes are displayed and the user chooses one reason code from the list. At this point, the system updates these data to the database, and assigned a unique RA Number to this return. The return status is 'O'pen. The RA number and status is displayed for the user. A sample screen is as follows. The **bold underlined** text indicates user input; everything else is displayed automatically.

```
SQL> start a:OpenRA.sql

****** OPEN a Return Authorization ******

Today's Date:  22-MAR-2014
Enter Customer Number: 102
   Customer Name: Jones, Elaine
   Shipping Address: 123 Main Street
   City, State Zip: Los Angeles, CA 91001
   Phone: (323) 511-1199
Enter Product Number: P1
   Prod Number: P1
   Prod Description:  Pencil
   Unit Price: $2.00
Enter the return quantity: 20
   Amount returned:   $40.00

Please choose from the following reason code:
     Reason     Reason
      Code      Description
     -------  --------------------------
        1     Unsatisfied quality
        2     Defective product
        3     Cannot assemble
        4     Different from description
        5     Arrive too late

Enter the reason code: 1

***** Return Authorization --> 1001 is now Open
***** Please print the return label. Thank you! *****

Return Authorization Number: 1001
Mail to:  1213 Warehouse Road
          Gainesville, FL 13802
```

Note 1. Since the unit price may change over time, you need to store the unit price and amount credited information under each RA.

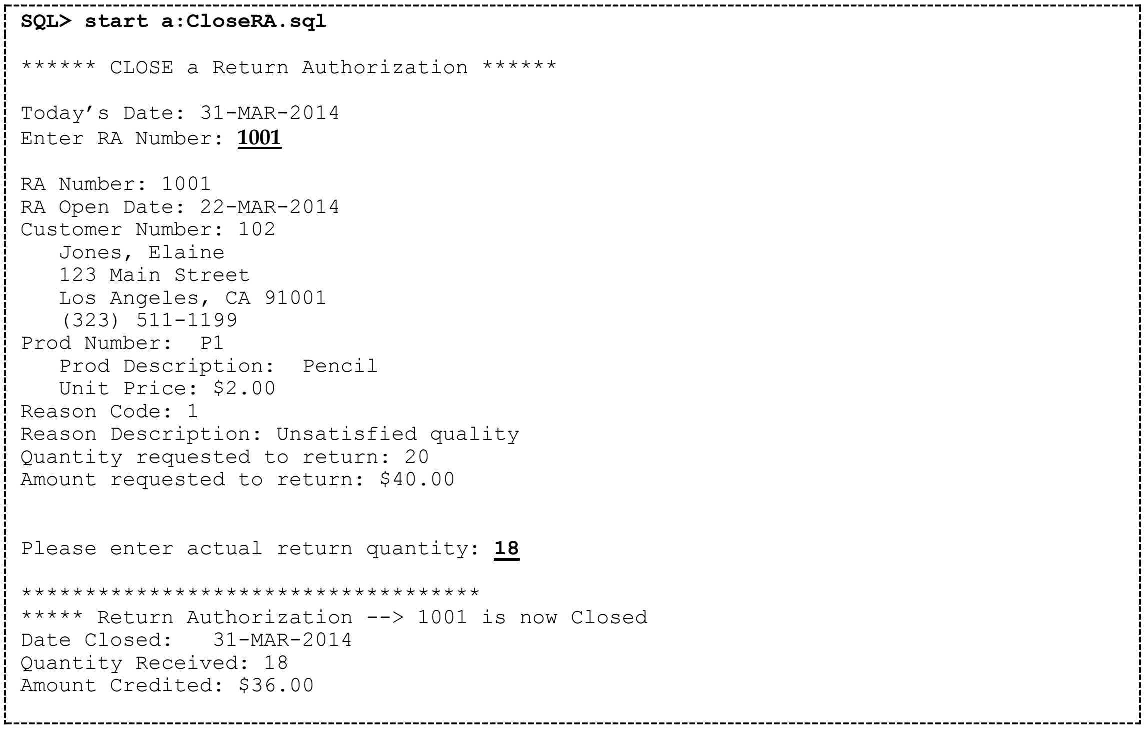
Note 2. At this moment, we trust the users to enter a valid reason code 😁.

Note3. To simplify the matter, the customer can only return one product at a time 😁.

Note 4. The screen shots format are for your reference -- you do not need to count spaces or line spaces to match the screen shots, but you do need to display all information found on the screen shots with a similar appearance.

Program 2. CloseRA.sql

The RA information is transmitted to the warehouse and the warehouse waits for the return to come in. The warehouse notifies us when the actual return is received. The user runs the **CloseRA** program to update this information to the system.
The system would first ask the user to enter the RA number, and displays all details of that return. The system then asks how many units have been received. The system updates the RA close date, quantity received, amount credited, and set the RA status to ‘C’losed.
Here is the sample screen shot:



```
SQL> start a:CloseRA.sql

****** CLOSE a Return Authorization ******

Today’s Date: 31-MAR-2014
Enter RA Number: 1001

RA Number: 1001
RA Open Date: 22-MAR-2014
Customer Number: 102
   Jones, Elaine
   123 Main Street
   Los Angeles, CA 91001
   (323) 511-1199
Prod Number:  P1
   Prod Description:  Pencil
   Unit Price: $2.00
Reason Code: 1
Reason Description: Unsatisfied quality
Quantity requested to return: 20
Amount requested to return: $40.00


Please enter actual return quantity: 18

************************************
***** Return Authorization --> 1001 is now Closed
Date Closed:   31-MAR-2014
Quantity Received: 18
Amount Credited: $36.00
```

Program 3. QueryRA.sql

QueryRA.sql is a program to display details of the RA. In other words, the user enters the RA number, and details of the RA are displayed, like the following:

```
SQL> start a:queryra

****** Return Authorization DETAIL Report ******

Enter RA Number: 1001

RA Number: 1001
RA Status: Closed

Customer Number: 102
   Jones, Elaine
   123 Main Street
   Los Angeles, CA 91001
   (323) 511-1199

Prod Number:  P1
   Prod Description:  Pencil
   Unit Price: $2.00

Reason Code: 1
Reason Description: Unsatisfied quality

RA Open Date: 22-MAR-2014
RA close Date: 31-MAR-2014

Quantity requested to return: 20
Amount requested to return: $40.00
Quantity received: 18
Amount credited: $36.00
```

Program 4. Aging.sql

Many RAs sit there open for a long time and the customers never bother to mail in the products. The manager wants to know, how many RAs have been Opened for *x* number of days and have not been closed (this is typically called the “aging” report).

Here is a sample of the report. Please sort your report by the longest DaysOpen first.

```
SQL> start a:aging


****** OPEN RA AGING Report ******

Today’s Date: 10/28/2014
Please enter number of days to query: 30


RA     RA      Date       Prod   Product        Request   Unit   Request      Days
Number Status  Open       Num    Description    Qty       Price  Amount       Open
------ -----   ---------  ----   -----------    -------   -----  ----------  ------
1008   O       4/11/2014  P1     Pencil         42        2.00    84.00        205
1010   O       7/28/2014  P5     Eraser         15        3.00    45.00         92
...    ...        ...
```

Program 5. ByReason.sql Program

The management wants to know, for each product, what are the reasons for return.
By running the ByReason program, the manager can see for each product, each reason, the number of return authorization, total request quantity and total request amount of returns. Please note that, even the customers never mail in the products, by calling our Operations department, it indicates unsatisfaction of our products. So this query should look at ALL return authorizations regardless of status. Here is a sample report:

```
SQL> start a:byReason


****** Return Product BY REASON Report ******

Prod   Product      Reason  Reason                 No of     Total       Total
Num    Description  Code    Description            Returns   Qty         Amount
-----  -----------  ------  ---------------------  -------   -------   ---------
P1     Pencil       1       Unsatisfied quality        25     1,220    $2,440.00
P1     Pencil       2       Defective product           3        10    $   20.00
...
P2     Pen          1       Unsatisfied quality         1         3    $    4.50
P2     Pen          5       Arrive too late             1       500    $  750.00
P2     Pen          ...
...
```

First Deliverable

Due Date: 11/18/2014

(1) ER Diagram
(2) Third Normal Form tables
(3) Create all tables (including primary keys and foreign keys) and insert test data (5-6 records per table) in Oracle in one program called **setup.sql**. Set echo on and spool setup.sql to **setup.txt**. Turn in a copy of setup.txt.

Second Deliverable

Due Date: 12/2/2014

(1) Revised first delivery (if you receive perfect score on the first delivery, simply turn it in again).
(2) Program 1, OpenRA.sql
(3) Program 3, QueryRA.sql

Final Project Delivery

Due Date: at the Final Exam 12/16/2014

Include the followings in an envelope:

(1) A flash disk or DVD with the following SQL programs. **Please do not put your files under any subdirectory/folders**.

```
Setup.sql
OpenRA.sql
CloseRA.sql
QueryRA.sql
Aging.sql
byReason.sql
```

(2) A stapled document with the following sections:

- Your name.
- The file names of the four programs on your floppy disk.
- (optional) Methods that I can contact you in case of an emergency, such as telephone number or email address, if you check it regularly.
- Table structure (such as `describe products;`).
- Test data (such as `select * from products;`).
- Program code.

Grading: I will personally grade each one of your project on Oracle. I will grade by using your system to create an RA, close an RA, run various reports, and check the correct impact to your database.