PROG3200 – Fall 2018

Assignment 04 – Due December 9, 2018 11:59pm

If there are any parts of the commands or system information asked for you do not know, research how to accomplish the task.

To be completed **individually**. There are more than one way to answer the following questions, so each student will most likely have different command sequences, variable names, etc. **Save the entire sequence of questions in a single .sql script file. Please save the entire output as a single .txt file. Be sure both your script and output are easily viewable.**

Please submit an electronic soft copy (script .sql file and output .txt file) to eConestoga. You do not need to submit a hard copy of this assignment.

Familiarize yourself with the CP/A assignment standards. You should, at a minimum, capitalize SQL language keywords (SELECT, FROM, JOIN, etc.). Identifiers may be in lower case or mixed case – strive for consistency. Be neat.

Question 1 [10 Marks]

Create a table named INVENTORY which has five columns: INVENTORY_ID NUMBER(6) [Primary key], ITEM_NAME VARCHAR2(255), QUANTITY NUMBER, PRICE NUMBER, ITEM_SIZE VARCHAR(255), INVENTORY_VALUE NUMBER.

Create a trigger that only allows an insert to complete if the SIZE value is one of the following (and any case variations): S, M, L, XL, small, medium, large, extra-large. Whichever the user attempts to insert into the table, the value actually inserted should always be upper case S, M, L, or XL. If an invalid input is attempted to be inserted, reject the insert and raise an application error that states 'Please enter for SIZE only: S, M, L, XL, small, medium, large or extra-large'.

Insert into the INVENTORY table (1, 'Web Shooter, 2, 19.00, 's', 38.00)

Insert into the INVENTORY table (2, 'Fantasticar', 4, 3000.00, 'very big', 12000.00)

Insert into the INVENTORY table (3, 'Mjolnir', 1, 100.00, 'medium', 100.00)

Run a SELECT * statement on the INVENTORY table.

Question 2 [10 Marks]

DELETE FROM all rows created in Question 1 from the INVENTORY table.

Create an insert command to insert the following into the table INVENTORY: (42, 'LMD', 331, 199.99, 'M').

Run a SELECT * statement on the INVENTORY table. You will notice the VALUE column for the item Tribble is missing.

Create a trigger which fires on inserts or updates and fills in the INVENTORY_VALUE column as Quantity multiplied by Price.

Create an insert command to insert the following into the table INVENTORY: (43, 'Arc Reactor Cube', 5, 999.99, 'M').

Run a SELECT * statement on the INVENTORY table. You should see that item 43 has a value, while item 42 does not.

Create an update command to update the price of item 42 to 299.99.

Run a SELECT * statement on the INVENTORY table.

Question 3 [10 Marks]

Create a table called INVENTORY_AUDIT which has the following columns: DATE_CHANGED DATE, USER_NAME VARCHAR2(30), INV_ID NUMBER(6) [Foreign Key which references INVENTORY INVENTORY_ID], OLD_ QUANTITY NUMBER, NEW_ QUANTITY NUMBER.

Create a trigger which inserts a row into the INVENTORY_AUDIT table with each field filled whenever an update statement changes the number of items in the INVENTORY table. DATE_CHANGED is the date and time the change was made, and USER_NAME is the name of the user that issued the change.

Run a SELECT * command on the INVENTORY table.

Create an update command to update all items in the inventory table with one less quantity.

Run a SELECT * command on the INVENTORY_AUDIT table.

Question 4 [10 Marks]

Create a table called TABLES_AUDIT which has the following columns: DATE_CREATED DATE, USER_NAME VARCHAR2(30), TABLE_NAME VARCHAR2(30).

Create a trigger that fires whenever a new table is created. This trigger should add a new row into the TABLES_AUDIT table, where DATE_CREATED is the date and time the table was created, USER_NAME is the name of the user that created the table, and TABLE_NAME is the name of the newly created table.

HINT: You do not have permission to create DDL triggers on the entire database, but you do have permission to create DDL triggers on your own schema.

Create a table named CUSTOMER with the following columns: CUSTOMER_ID NUMBER(6) [Primary key], FIRST_NAME VARCHAR2(30), LAST_NAME VARCHAR2(30), PHONE_NUM NUMBER.

Run a SELECT * command on the TABLES_AUDIT table.

Question 5 [10 Marks]

Insert the following rows into the CUSTOMER table:

(1, 'Peter', 'Parker', 5554242)

(2, 'Tony', 'Stark', 5550000)

(3, 'Carol', 'Danvers', 5550504)

Create a RESERVE_ITEM table which allows customers to call the store, and put aside an item until they come in and pick it up. The RESERVE_ITEM table should have the following columns: RESERVE_ID NUMBER(6) [Primary Key], INV_ID NUMBER(6) [Foreign Key which references INVENTORY INVENTORY_ID], CID NUMBER(6) [Foreign Key references CUSTOMER CUSTOMER_ID].

Create a trigger that will ensure that when a new row is inserted into the RESERVE_ ITEM table, the INV_ID exists and that the quantity of the item is greater than 1. The trigger should have the following behaviour:

a) If the INV_ID does not exist in the INVENTORY table, the trigger should raise an application error that states 'Invalid Inventory ID'
b) If the CID does not exist in the CUSTOMER table, the trigger should raise an application error that states 'Invalid Customer ID'
c) If the INV_ID does exist in the INVENTORY table, but the quantity of that item is 0, then the trigger should raise an application error that states '<Item Name> is currently out of stock' where <Item Name> is the proper item name for that item.
d) If the INV_ID does exist in the INVENTORY table and the quantity of that item is greater than 0, then the appropriate row in the INVENTORY table should be updated by decreasing the quantity by 1.

Run a SELECT * statement on the INVENTORY table.

Insert a reservation for INV_ID 101 into the RESERVE_ITEM table for the customer 1.

Insert a reservation for INV_ID 42 into the RESERVE_ITEM table for the customer 4.

Insert a reservation for INV_ID 42 into the RESERVE_ITEM table for the customer 1.

Insert a reservation for INV_ID 42 into the RESERVE_ITEM table for the customer 2.

Insert a reservation for INV_ID 42 into the RESERVE_ITEM table for the customer 3.

Run a SELECT * on the RESERVE_ITEM table.

Run a SELECT * statement on the INVENTORY table.