**CCGC 5004 Database Systems**

**Lab Exercise 4 DML Statements and Transactions**

**Overview**

**To receive credit for this lab you must be present in today’s class. Late submissions are deducted 10% per day up to 5 days. Submissions received after 5 days will be given a grade of 0.**

**Make copies of the CATEGORIES, PRODUCTS, and CUSTOMERS tables. APPEND the word \_COPY to each table.**

**Use the following statement to do this:**

**CREATE TABLE categories\_copy AS SELECT \* FROM categories;**

**Do the same for the other tables.**

**Then use the COPY tables for the DML operations. This will not affect the actual tables.**

1. Write an INSERT statement that adds this row to the Categories table:

category\_name: Brass

Code the INSERT statement so MySQL automatically generates the category\_id column. **Screen Capture 1**

Perform a **SELECT** statement to show the row was inserted. **Screen Capture 1A**

1. Write an UPDATE statement that modifies the row you just added to the Categories table. This statement should change the category\_name column to “Woodwinds”, and it should use the category\_id column to identify the row. **Screen Capture** **2**.

Perform a **SELECT** statement to verify the changes were made to the **CATEGORIES** table. Only display the row that was added in the **SELECT**. **Screen Capture 2A.**

1. Write a DELETE statement that deletes the row you added to the Categories table in exercise 1. This statement should use the category\_id column to identify the row. **Screen Capture 3**

Write a **SELECT** statement that will verify the row has been removed from the **CATEGORIES** table**. Screen capture 3A**.

1. Write an INSERT statement that adds this row to the Products table:

product\_id: The next automatically generated ID   
category\_id: 4  
product\_code: dgx\_640  
product\_name: Yamaha DGX 640 88-Key Digital Piano  
description: Long description to come.  
list\_price: 799.99  
discount\_percent: 0  
date\_added: Today’s date/time.

Use a column list for this statement. **Screen capture 4**.

Perform a **SELECT** statement to show the row was inserted. **Screen Capture 4A.**

1. Write an UPDATE statement that modifies the product you added in exercise 4. This statement should change the discount\_percent column from 0% to 35%.  **Screen capture 5.**

Write a **SELECT** to show was completed. **Screen capture 5A**.

1. Write a DELETE statement that deletes the Keyboards category. When you execute this statement, it will produce an error since the category has related rows in the Products table. To fix that, precede the DELETE statement with another DELETE statement that deletes all products in this category. (Remember that to code two or more statements in a script, you must end each statement with a semicolon.)  **Screen capture 6.**

Write a **SELECT** statement that shows the value has been changed. **Screen capture 6A.**

1. Write an INSERT statement that adds this row to the Customers table:

email\_address: rick@raven.com  
password: (empty string)  
first\_name: Rick  
last\_name: Raven

Use a column list for this statement**. Screen Capture 7**

Write a **SELECT** statement that shows the row has been inserted. **Screen capture 7A.**

1. Write an UPDATE statement that modifies the Customers table. Change the password column to “secret” for the customer with an email address of [rick@raven.com](mailto:rick@raven.com). . **Screen capture 7.**

Write a **SELECT** statement that shows the value has been changed. **Screen capture 7A.**

1. Code a transaction starting with **START TRANSACTION** that will use an UPDATE statement that modifies the Customers table. Change the password column to “reset” for every customer in the table. If you get an error due to safe-update mode, you can add a LIMIT clause to update the first 100 rows of the table. (This should update all rows in the table.) **Screen capture 9.**

Write a **SELECT** statement that shows the values have been changed. **Screen capture 9A.**

**ROLLBACK the changes you made using the UPDATE command.**

Write a **SELECT** statement that shows the values have been changed. **Screen capture 9B.**