### Questions on Part 2: Turn in only this sheet with your answers

1. What does the CREATE USER statement do? Hint: It’s not creating a user….

We know that the command CREATE USER does not actually create a user because we never set a password for it. Knowing this we can assume that the CREATE USER command applies to the database being manipulated, it connects the user account that has been created to the database accounts.

1. Which types of objects are securable with GRANT and DENY statements? Research required. Don’t ask before doing so.

From the lab we know we can use GRANT and DENY for allowing access to commands like SELECT or ALTER, and custom views like v\_fudgemart\_managers. These views are user-defined types. We can also allow or deny access to system-defined types like SELECT or INSERT

1. Describe a scenario by which you would not want a user reading the table directly but rather using a view instead.

Perhaps we would want to censor certain information like social security numbers we may want to create a view of the employees table where the SSN column is stripped from the view. Then we only grant employees access to the view instead of the table.

1. Describe a scenario by which you would not want a user updating data in the table directly but rather using a stored procedure instead?

A stored procedure would allow a user to enter data into a table while maintaining the ability to commit only if the entire procedure is completed successfully, otherwise a procedure allows us to rollback any changes by the user which do not fit our standards or are partially completed.

1. Write SQL to deny the guest user permissions to delete from the colors table you created in part two. Place your sql here and the message returned when executed.

SQL: REVOKE DELETE ON colors TO guestuser

RETURN:

Command(s) completed successfully.

1. Write SQL to allow the guest user the ability to run one of your SQL Views. Place your sql here and the message returned when executed.

GRANT SELECT ON v\_fudgemart\_vendors TO guestuser

Command(s) completed successfully.

1. Write SQL to give the guestuser permissions to insert and read data from the COLORS table you created in part two. Place your sql here and the message returned when executed.

GRANT SELECT, INSERT ON colors TO guestuser

Command(s) completed successfully.

1. Write SQL to give the guestuser permissions to execute the p\_add\_two\_colors stored procedure you created in part one of this lab. Place your sql here and the message returned when executed.

GRANT EXECUTE ON p\_add\_two\_colors TO guestuser

Command(s) completed successfully.