IST659 Quiz 6

8 points total, Due 12/9

Your Name: Rayanna Harduarsingh

Your SU Email: rharduar@syr.edu

For the following quiz, use the **cheepwebhosting** database. If you do not have it use the database provisioner on <http://localhost:5000> to create it. Please provide screenshots for each. For credit you must make sure to demonstrate all aspects of your code were executed successfuly and include your name in the screenshot as a comment,  
like this -- name

1. [2 pts] Write a database trigger on the **cheep\_web\_hosting** table to force any new or updated rows to use an **account\_plan** with lower cased characters only.   
   To test, update row ID 7 setting **account\_domains** to tweetybird.com, then select from the table to demonstrate the trigger works – your screenshot should observe that this row’s domain has changed account plan is lower cased.   
   Graphical user interface, application

   Description automatically generated
2. [2 pts] Write a transaction-safe stored procedure called **p\_update\_customer** which takes a **customer\_email** and a **customer\_name** as input, then updates the customer name for all rows matching the email. Rollback when an error occurs or when the number of rows which should be updated does not match the actual number updated.  
   Demonstrate the procedure works by changing **bbird@coldmail.com** to the name **Huge Bird** provide an SQL query which proves the output was changed. Table

   Description automatically generated
3. [2 pts] Write an index to improve the performance of the following query:  
   select hosted\_server\_name, sum(plan\_monthly\_fee) as total\_fee  
   from cheep\_web\_hosting  
   group by hosted\_server\_name

Include the drop/create index script and a screenshot of the query plan which demonstrates the index is being used. Include screen shots of the before and after query plans along with a short justification as to why the has improved performance.

**Before Index:**

Graphical user interface, application

Description automatically generated

**After Index:**

Graphical user interface, text, application, email

Description automatically generated

In the before query plan, we can see the query spent 78% of its time using the table to sort the hosted server name. With a clustered index, SQL is using it to search each row one by one, decreasing performance by slowering down its execution time. With a non-clustered index, although we can’t visually see the process, it aids in the performance of the query much faster by using the index rather than the table to search row by row. In the after query plan, we can see that with the index, it is no longer sorting the query with the elimination of the sort in the select, therefore improving the reading performance significantly.

1. [2 pts] Assuming the **account\_name** column is the business key, what normal form is the **cheep\_web\_hosting** table in? For full credit please justify your answer with a data dependency

Right now, the table Cheep web hosting is in the 2nd Normal Form because I still see many data repeating such as customer name and email. This table has transitive dependencies because there is dependency occurring between two columns. Transitive dependency is when a column’s value relies (or depends) on another column with an intermediate column in the middle. For example, transitive dependency can exist when a column is not a key, but it can act like a key to lookup an atomic value in another column. In this case when we want to normalize the data(improve database design by reducing redundancy and improving integrity), account name is acting as our key as it has data dependencies. It also is a business, or natural, key which is the best for data normalization. When finding the data dependencies, I looked for the non-key columns in the cheep web hosting table. Account name, our business key, can act like a key to look up the columns customer name and customer email as those two columns are dependent on account name. In order for the table to have transitive dependencies, it requires two non-key columns which is customer email and customer name. The account name column acts as a key for values in customer name and customer email, and it does not require account id to exist. Because the transitive dependencies still exist and has not been resolved, the table is still in the second form. Once the transitive dependencies have been resolved by removing the dependent columns and adding them to a new table, where its original table’s column acting as a key will now be the foreign key referencing the new table, it will be in the 3rd normal form where there are zero columns with no key dependency, zero columns with partial key dependency and zero columns with transitive dependency.