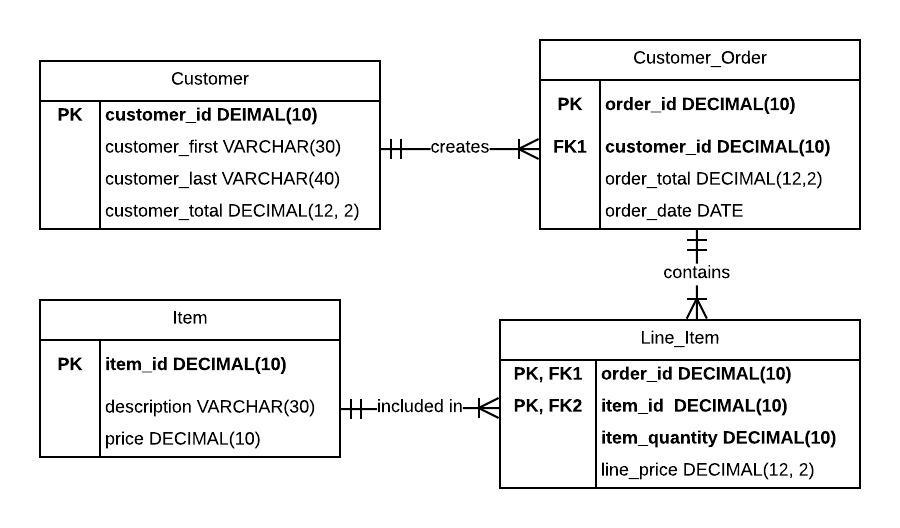
This submission template is a convenient document for you to provide the screenshots and explanations for Lab 4. This submission template is intended to be used in conjunction with the Lab 4 Instructions document. The instructions document illustrates how to correctly execute each SQL construct, explains important theoretical and practical details, and contains the complete set of instructions on how to complete this lab.

**Name**: scott kaeneman

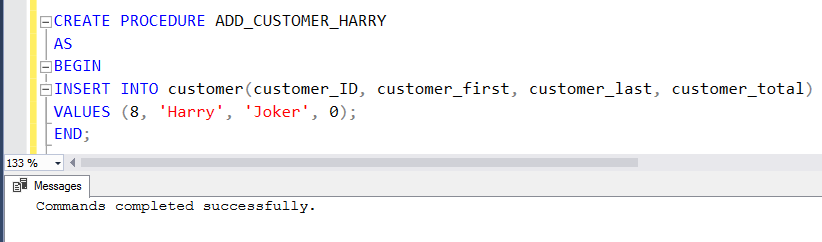
**Date: 6/4/2018**

**Section One – Stored Procedures**

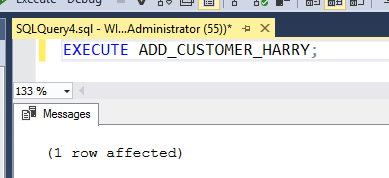
1. ERD creation



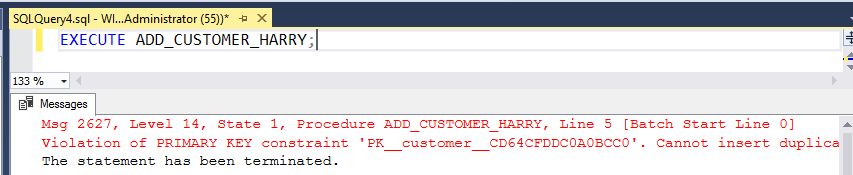
3. Stored procedure creation



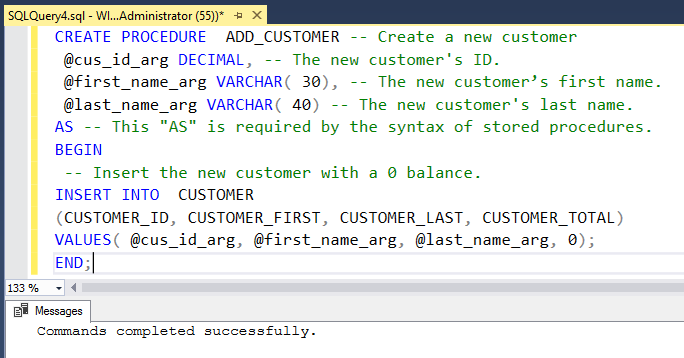
5. Stored procedure execution



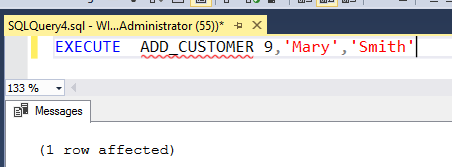
7. Stored procedure execution failure



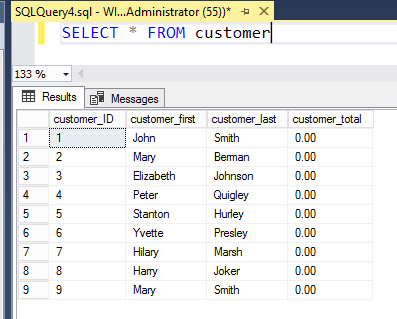
8. Reusable stored procedure creation



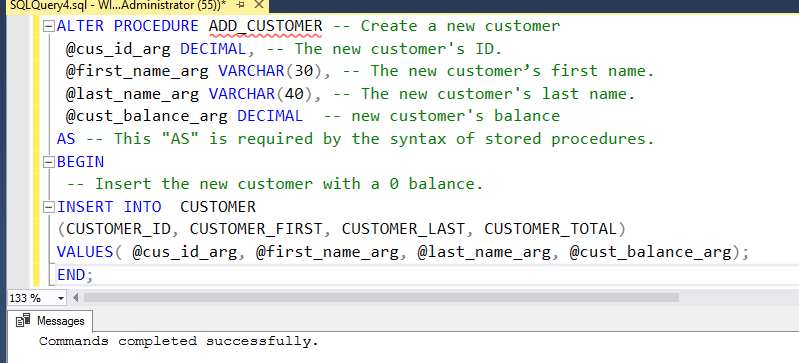
9. Reusable stored procedure execution



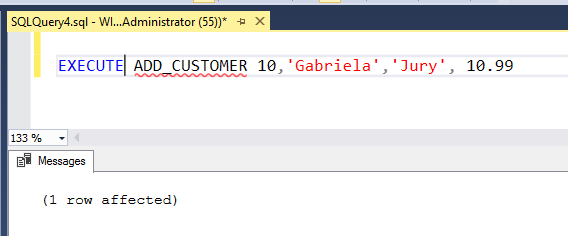
10. Stored procedure execution verification



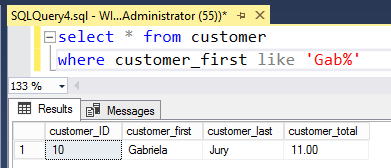
11. Fourth argument stored procedure creation



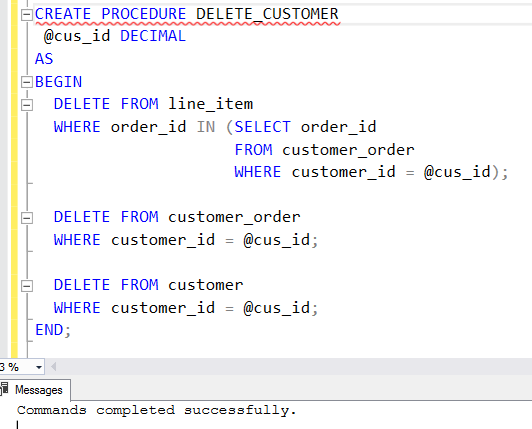
12. Fourth argument stored procedure execution

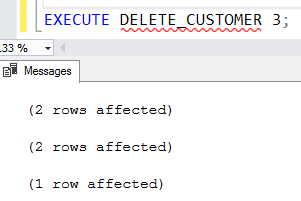


13. Stored procedure execution verification



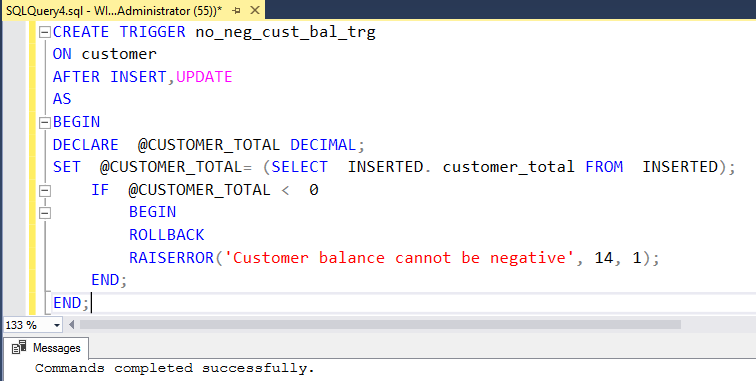
14. Deletion stored procedure creation and execution

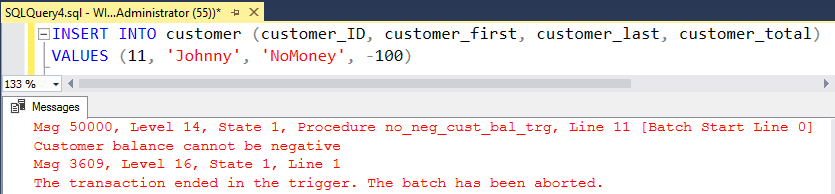




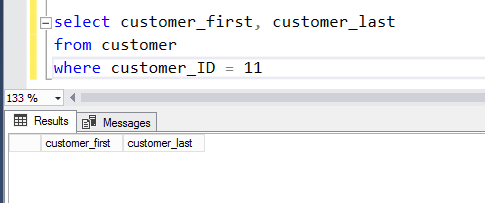
**Section Two – Triggers**

15. Trigger creation

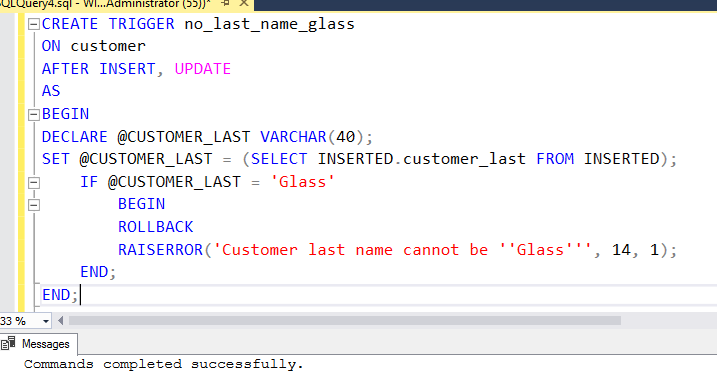


16. Negative balance insertion  


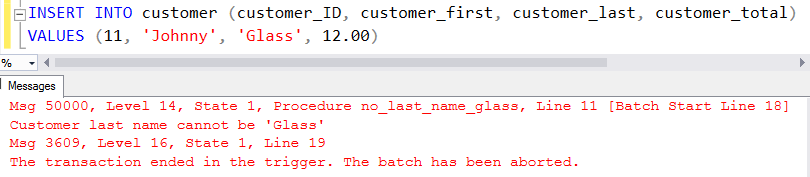
17. Trigger verification



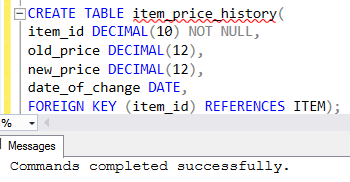
18. Last name trigger creation

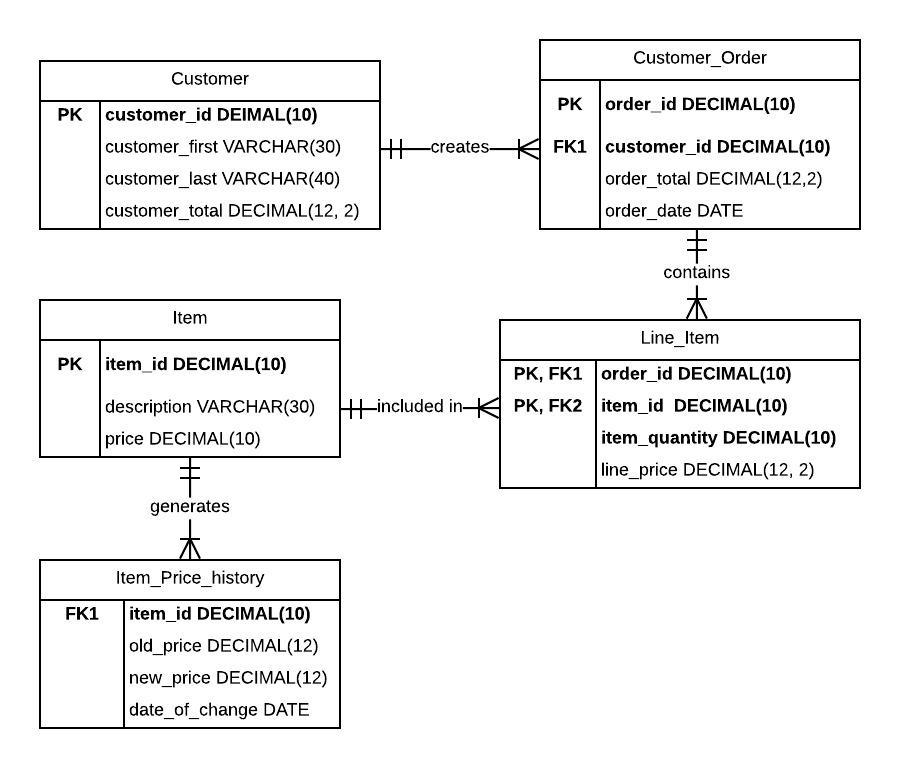


19. Illegal insertion



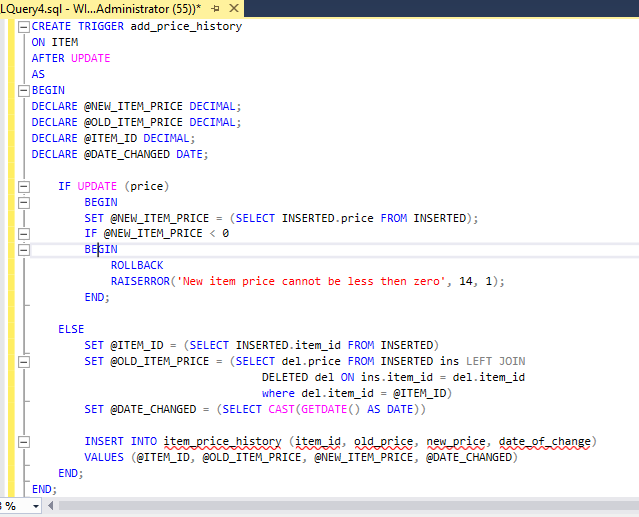
20. ERD creation with Item\_price\_history included





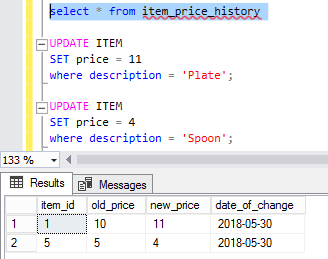
21. Item trigger creation

In the below screenshot a trigger called add\_price\_history is being created. The trigger is activated when a row in the Item table has its price updated. If the price of the new item is less than zero the transaction is rolled back. Otherwise the price of the item is updated in the Item table and a row is created in the item\_price\_history table which lists the item as well as the old and new price for it.



22. Price changes and trigger verification

The contents of the table in the below screenshot consist of an item\_id that is gathered from the Item table when the update occurs. The old\_price consists of an outer left join where the deleted item’s id is equal to the @ITEM\_ID variable. The new\_price data consists of whatever new value is being passed in to update the row. And finally the date\_of\_change simply gets the date that the change was made.





Your lab submission will be evaluated according to the following rubric.

|  |  |  |
| --- | --- | --- |
|  | **Letter Grade** | **Qualities Demonstrated by the Lab Submission** |
| **Correctness, completeness, and constitution**  **Measures the correctness and completeness of the results, and the quality of the constitution of the SQL constructs** | A+ ➔ 100 | The results and explanations are entirely complete and correct for all steps. There are absolutely no technical or other errors present. There is no known way to improve the logic and makeup of any of the SQL constructs. |
| A ➔ 96 | One insignificant technical or other error is present, but otherwise the results and explanations are entirely complete and correct for all steps. Excluding the insignificant error, there is no known way to improve the makeup of any of the SQL constructs. |
| A- ➔ 92 | One or two consequential technical or other errors are present, but otherwise the results and explanations are entirely complete and correct for all steps. Excluding the one or two errors, there is no known way to improve the makeup of any of the SQL constructs. |
| B+ ➔ 88 | A few steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of most SQL constructs are sound. |
| B ➔ 85 | A few steps have significantly incomplete or incorrect results or explanations. The results and explanations are mostly complete and correct for the remainder of the steps, with the exception of a few insignificant technical or other errors. The logic and makeup of most SQL constructs are sound. |
| B- ➔ 82 | About ¼ of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of at least ¾ of the SQL constructs are sound. |
| C+ ➔ 78 | About ¼ of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are mostly complete and correct for the remainder of the steps, with the exception of a few insignificant technical or other errors. The logic and makeup of at least ¾ of the SQL constructs are sound. |
| C ➔ 75 | About half of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of at least half of the SQL constructs are sound. |
| C- ➔ 72 | About half of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are mostly complete and correct for the remainder of the steps, with the exception of a few insignificant technical or other errors. The logic and makeup of at least half of the SQL constructs are sound. |
| D ➔ 67 | About ¾ of the steps have significantly incomplete or incorrect results or explanations. The results and explanations are complete and correct for the remainder of the steps. The logic and makeup of at least ¼ of the SQL constructs are sound |
| F ➔ 0 | All or almost all of the steps have incomplete or incorrect results or explanations. The logic and makeup of all or almost all of the SQL constructs are unsound. |