This submission template is a convenient document for you to provide the screenshots and explanations for Lab 1. This submission template is intended to be used in conjunction with the Lab 1 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.

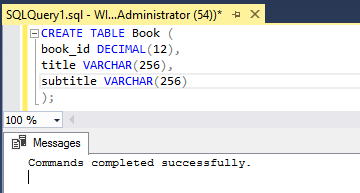
**Name**: Scott Kaeneman

**Date: 5/11/2018**

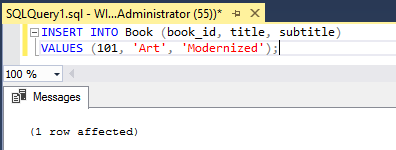
Note: Section One in the lab instructions is for your benefit, to teach you by example how to complete the exercises in Section Two. It is not necessary, however, to submit the screenshots for Section One.

**SECTION TWO**

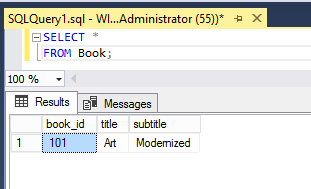
19. Table creation



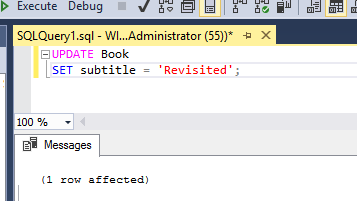
20. Row insertion



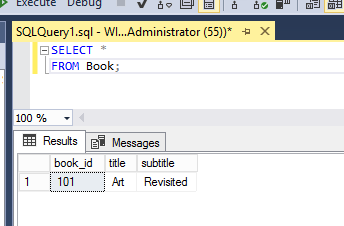
21. Row listing



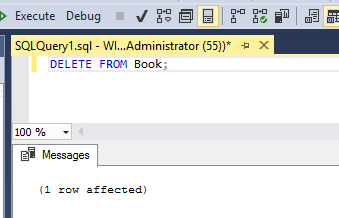
22. Row update



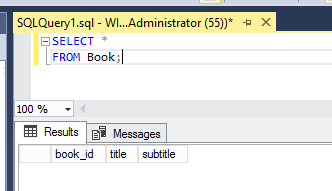
23. Row listing



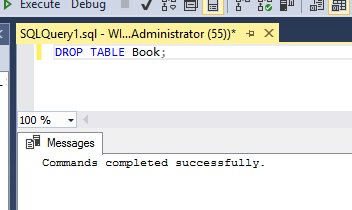
24. Row removal



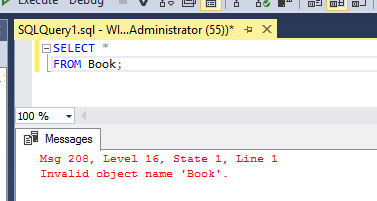
25. Row listing



26. Table drop



27. Row listing attempt



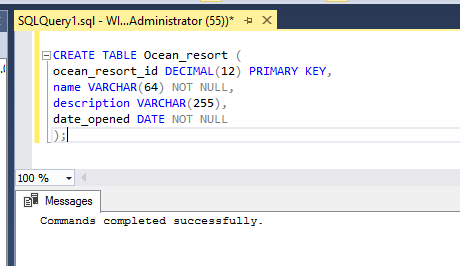
28. Diagnosis explanation

I would use the resulting error message to diagnose the issue by first looking at the line number that is generating the error, in this case it’s “Line 1” which has the “SELECT \*” statement, so we can assume that SQL Server cannot perform a select to return all rows. The error also states that ‘Book’ is an “invalid object name” which means that there is a problem with the Book table and it can’t be found. So at this point we know that there is an issue retrieving rows from the Book table. In Microsoft SQL Server I would then try and take a look at the Book table to see what was going on with it, this can be done through SSMS by clicking to the left of the databases name to expand its properties and then clicking on the “Tables” directory where we would find that the table dbo.Book is missing, alternatively we could also use SQL and do “SELECT \* FROM INFORMATION\_SCHEMA.TABLES” and then we could check under the TABLE\_NAME field where we would also find that the Book table had been dropped.

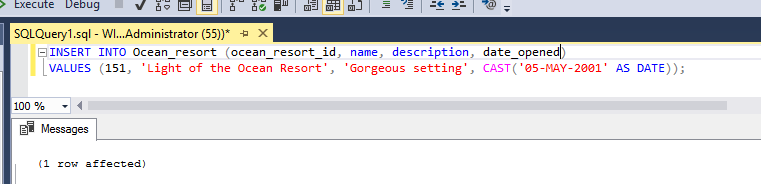
Note: Section Three in the lab instructions is for your benefit, to teach you by example how to complete the exercises in Section Four. It is not necessary, however, to submit the screenshots for Section Three.

**SECTION FOUR**

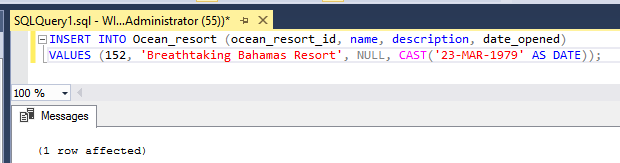
49. Table creation



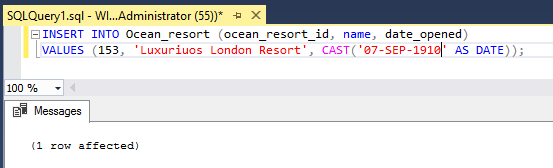
50. Row insertion



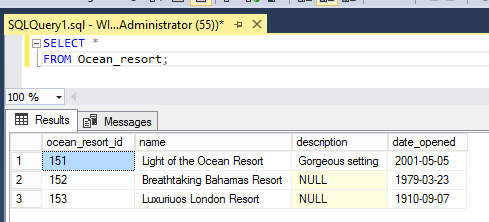
51. Row insertion



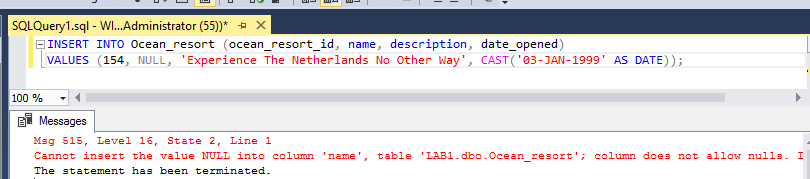
52. Row insertion



53. Row listing



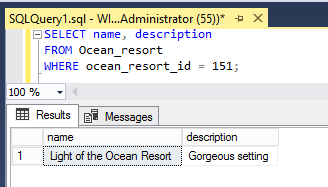
54. Row insertion attempt



55. Diagnosis explanation

After running the above “INSERT INTO” attempt the error states “Cannot insert the value NULL into column ‘name’” and then gives the database name, schema, and table name of where the error is occuring. The error message also states the additional information “Column does not allows nulls. INSERT fails.” A lot of times error messages can be cryptic and need to be looked up but definitely not in this case, SQL Server is kind enough to tell you that you’re trying to insert a null value into a column that does not allow null values and that the insert has failed. The error message is conveying the fact that data must be present in the name column in order for the row to be saved to the table.

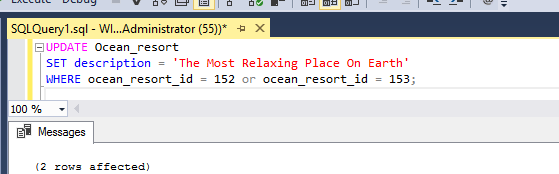
56. Limited row listing



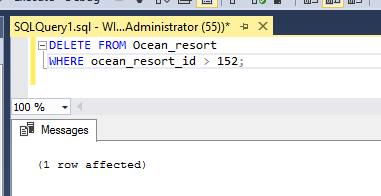
57. Value of limiting row listings explanation

It’s a good idea to limit the rows returned to exactly the data you need instead of simply returning all rows due to the fact that it would be a waste of resources and time otherwise. For example, returning all rows in a table called “Customers” that happens to store millions of customers records could potentially cause a strain on the database application as well as the underlying server platform itself. Also it may take some time to retrieve all of this data when it would have been more efficient to scope the query to exactly the one customer you were looking for. Additionally if the table is large and all results are returned but only a limited subset of the data is needed it’s like trying to find a needle in a haystack when you’re looking through the data.

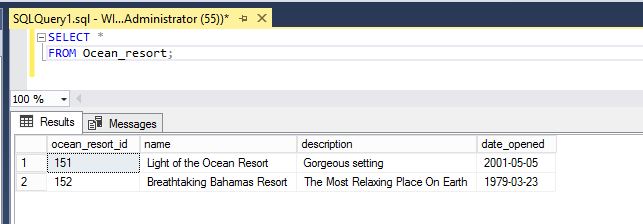
58. Limited row update



59. Limited row deletion



60. Row listing



Your lab submission will be evaluated with the following grading matrix.

|  |  |  |
| --- | --- | --- |
|  | **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. |

