|  |
| --- |
| **Overview of the Lab** |
| The purpose of this lab is to gently get you started using the Structured Query Language (SQL), by teaching you the most fundamental commands and concepts step-by-step. SQL is the de-facto query language for modern relational database management systems (RDBMS). All major, modern RDBMS support SQL. Database developers, administrators, and even software applications access RDBMS through the use of SQL. Familiarity with SQL is essential for working with and understanding modern RDBMS.  In this lab, you will learn how to:   * create and drop a table. * insert, update, delete, and select a row in a table. * use strings, dates, and numbers. * use a WHERE clause to limit the number of rows affected by the SELECT, UPDATE, and DELETE commands. * SELECT only a subset of columns in a result set. * add a NOT NULL constraint to a table column. * add a PRIMARY KEY constraint to a table column. * Insert a NULL value into a table row. |

|  |
| --- |
| **Lab 1 Explanations** |
| It is important to read through the Lab 1 Explanation document to successfully complete this lab. It is available in the assignment inbox alongside this lab. The explanation document illustrates how to correctly execute each SQL construct step-by-step, and explains important theoretical and practical details. |

|  |
| --- |
| **Required Software** |
| The examples in this lab will execute in modern versions of Oracle, Microsoft SQL Server, and PostgreSQL as is. If you have been approved to use a different RDBMS, you may need to modify the SQL for successful execution, though the SQL should execute as is if your RDBMS is ANSI compliant.  The screenshots in this lab display execution of SQL in the default SQL clients supported in the course – Oracle SQL Developer, SQL Server Management Studio, and pgAdmin. You are welcome, however, to use a SQL client other than these defaults if you prefer. |

|  |
| --- |
| **Preparing for the Lab** |
| You will need to install a RDBMS prior to completing this lab. If you are using Oracle, it is highly recommended that you create and login as a non-system user, to avoid damaging the database. You can create a user with the following commands:  CREATE USER *username* IDENTIFIED BY *password* DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;  GRANT connect, resource TO *username*;  You will then be able to login as the new user.    If you are using Microsoft SQL Server, it is highly recommended that you create and use a database other than the Master database. You can do so with the following commands:  CREATE DATABASE *database\_name*;  GO;  USE *database\_name*;  If you are using PostgreSQL you can use the UI wizard to create a database or use the following script from the default databse created at time of installation:  CREATE DATABASE database\_name; |

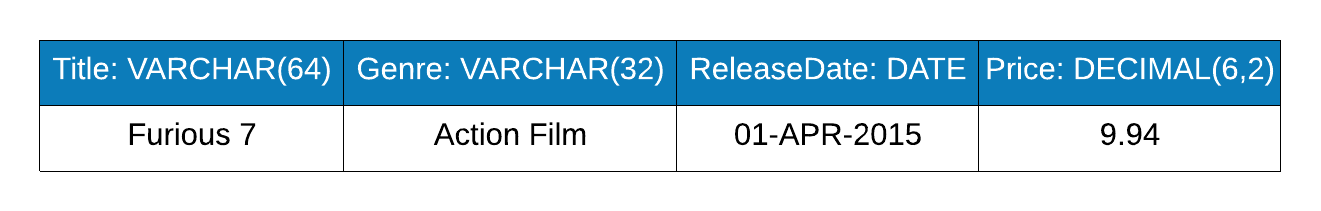
|  |
| --- |
| **Saving Your Data** |
| If you choose to perform portions of the lab in different sittings, it is important to *commit* your data at the end of each session. This way, you will be sure to make permanent any data changes you have made in your current session, so that you can resume working without issue in your next session. To do so, simply issue this command:  COMMIT;  We will learn more about committing data in future weeks. For now, it is sufficient to know that data changes in one session will only be visible only in that session, unless they are committed, at which time the changes are made permanent in the database. |

**Section One – Absolute Fundamentals**

**Section Background**

In this section, you learn the absolute fundamentals of SQL – creating and dropping a table, getting data into the table, listing the data in the table, and deleting and updating the data. You will be working with a Movie table that contains basic information about movies. When you have completed some steps in the section, the Movie table will look as illustrated below.

*Movie Table*

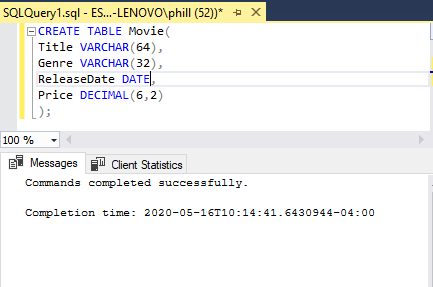
**

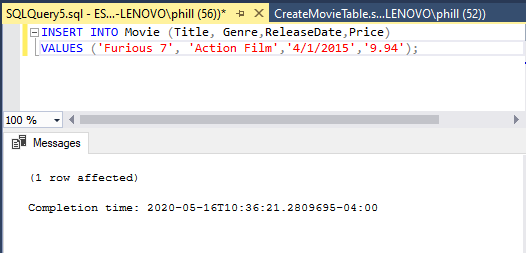
You will create this table and try out SQL commands using the table.

Do not worry if you do not recognize the structure and datatypes in the table above. The Lab 1 Explanation document and supporting lecture and textbook readings give you the information you need. Start reading the explanation document first, then iteratively complete the steps below. Each step below has an accompanying explanation in the explanation document.

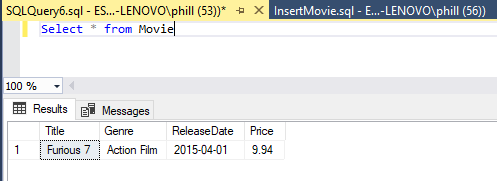
For each step that requires SQL, *make sure to capture a screenshot of the command and the results of its execution.* Submissions that do not contain screenshots will be returned to you. A screenshot is more legible if you use one of the many free tools to capture only the relevant portion of the screen, rather than capturing the entire application window. A few steps ask for explanations rather than SQL; no screenshot is needed for such steps.

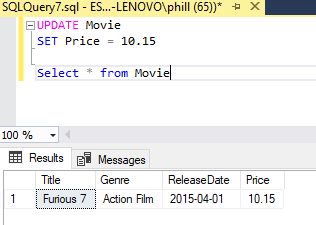
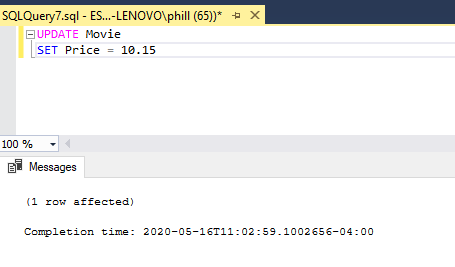
**Section Steps**

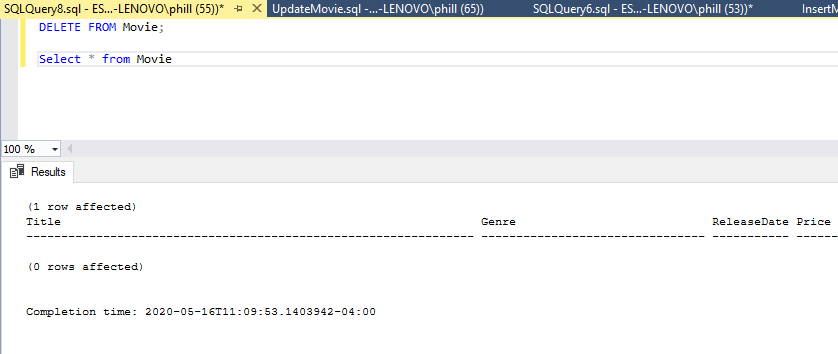
1. Create the Movie table. As a reminder, make sure to follow along in the Lab 1 Explanations document as it shows you how to create tables and complete the other steps.  
   
2. Insert the first row where the title is “Furious 7”, the genre is “Action Film”, the release date is 4/1/2015, and the price is $9.94.



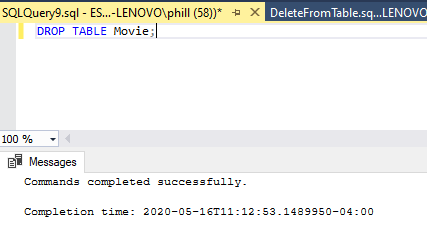
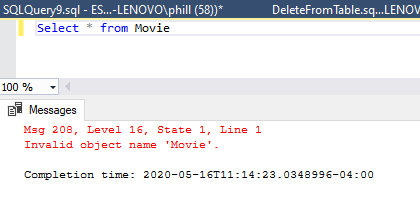
1. Select all rows in the table to view the row you inserted.



1. Update the price of the row in the table to $10.15, then select all rows in the table to view the row you updated.  
   
2. Remove all rows from the table, then select all rows in the table to verify there are no rows.



1. Drop the Movie table, then select all rows in the table to verify the table doesn’t exist.

1. Explain how you would use the error message resulting from step #6, in conjunction with the SELECT command, to diagnose the error.

The error message would point me to a misspelling or that the table object does not exist in my database.

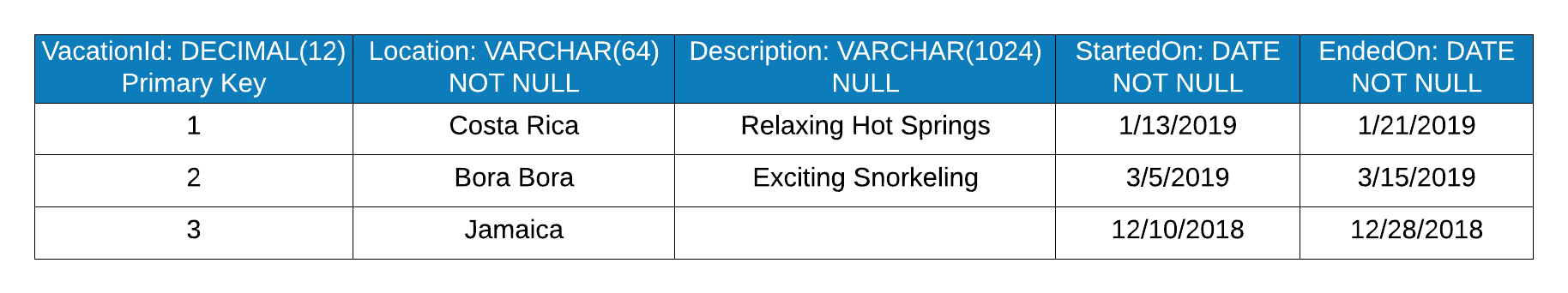
**Section Two – More Precise Data Handling**

**Section Background**

In this section, you enhance your skills by more precisely working with data. In the prior section, you learned to work with all rows in the table. In this section, you add to that by learning to pinpoint specific rows to be retrieved, modified, or deleted. You also learn how to add SQL constraints to your table, and to work with nulls.

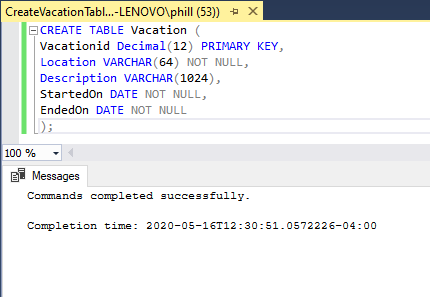
You will work with a vacation table, which will ultimately look like the below when all steps have been completed.

*Vacation Table*

**

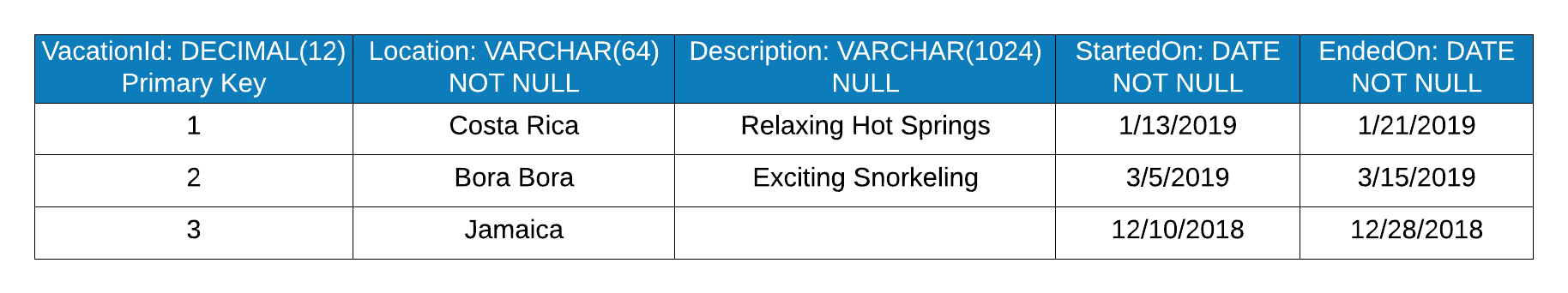
**Section Steps**

1. Create the Vacation table with its columns, datatypes, and constraints. As a reminder, make sure to follow along in the Lab 1 Explanations document as it shows you how to complete the steps.

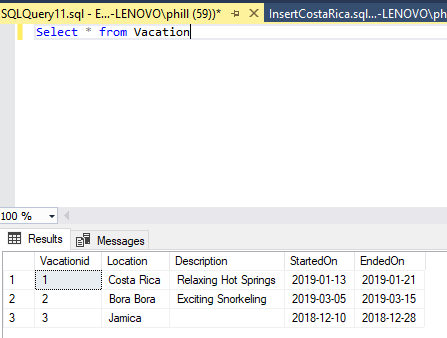


1. Insert the rows illustrated in the figure below.

*Vacation Table*

**

Note that the description for Jamaica is null.

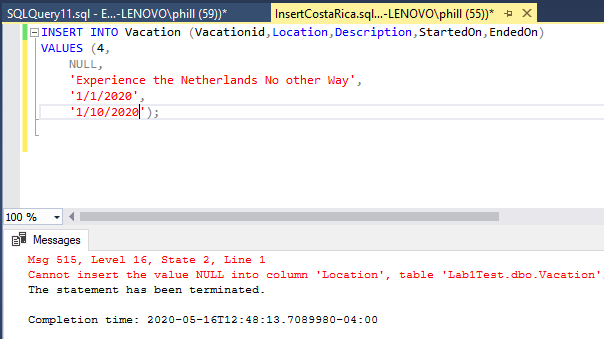
1. Select all rows from the Vacation table to show that the inserts were successful.   
   
2. Attempt to insert a row with the following values. The insert command will fail because the location column must have a value.  
     
   **VacationId** = 4

**Location** = NULL

**Description** = Experience the Netherlands No Other Way

**StartedOn** = 1/1/2020  
**EndedOn** = 1/10/2020

Explain how you would interpret the error message conclude that the location column is missing a required value.



The message indicates that you cannot insert a NULL into Location. I would check the variable type and any restraints on Location. I’d then find that NULL cannot be used.

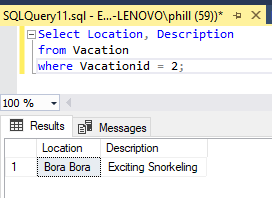
1. Now insert the row with a location intact, with the following values.  
     
   **VacationId** = 4

**Location** = Netherlands

**Description** = Experience the Netherlands No Other Way

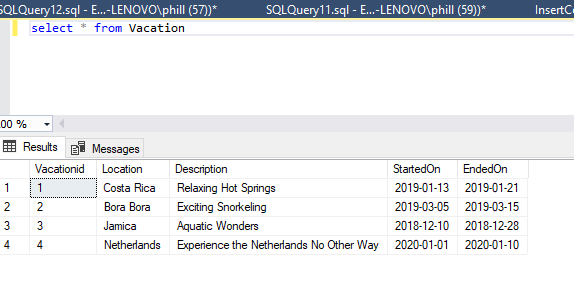
**StartedOn** = 1/1/2020  
**EndedOn** = 1/10/2020

1. Retrieve only the location and description for the Bora Bora vacation, using the primary key as the column that determines which row is retrieved. Explain why it is useful to limit the number of rows and columns returned from a SELECT statement.

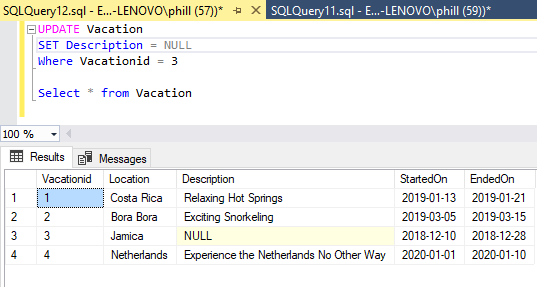


It’s more efficient to only select what is needed – especially if you have a large database.

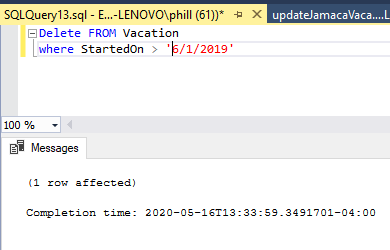
1. The Jamaica vacation has no description. Update the row so that its description is “Aquatic Wonders”. Select all rows in the table to show that the update was successful.

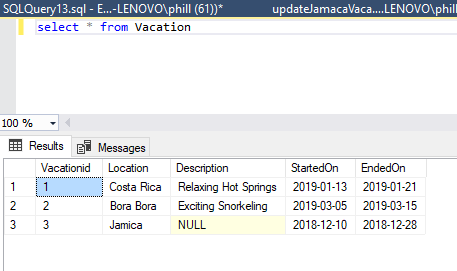


1. Update the Jamaica vacation so that it no longer has the description (i.e. its description is null). Select all rows in the table to show that the update was successful.



1. Delete all rows where the vacation started on a date greater than June 1st, 2019, by using the StartedOn column as the determinant of which rows are deleted. Select all rows in the table to show the delete was successful.





# Evaluation

Your lab will be reviewed by your facilitator or instructor with the following criteria and grade breakdown.



Use the **Ask the Facilitators Discussion Forum** if you have any questions regarding how to approach this lab. Make sure to include your name in the filename and submit it in the *Assignments* section of the course.