1. Write a Docker Script to perform the following actions.

* Download the Docker image for SQL Server Developer Edition
* Have a folder called DBScripts in the Docker root container where all the SQL Server Database scripts are present
* The .sql file in the DBScripts folder should run in Ascending order in the Database when the Docker image is built
* The SQL Server Developer edition should be exposed to the Docker host machine so that the Reporting tools / SSMS can connect to it from the host machine

please test with the following SQL scripts in the DBScripts folder.

* 01-create-database.sql

CREATE DATABASE testDB;

* 02-create-table.sql

USE DATABASE testDB;

CREATE TABLE Test

(

Id int,

Data varchar(50)

);

* 03-insert-data.sql

INSERT INTO Test (

Id, Data

)

VALUES (

1, 'A'

);

The expected outcome should be

* Docker container with the SQL Server Developer tools installed
* SQL Port exposed outside to the host image
* The Database and Table created with the Sample

APPROACH –

$ docker build .# Used to build the docker image using the following docker file so the all the .sql files in the sc container is executed when the image is built.

FROM microsoft/mssql-server-windows-developer

COPY sc .

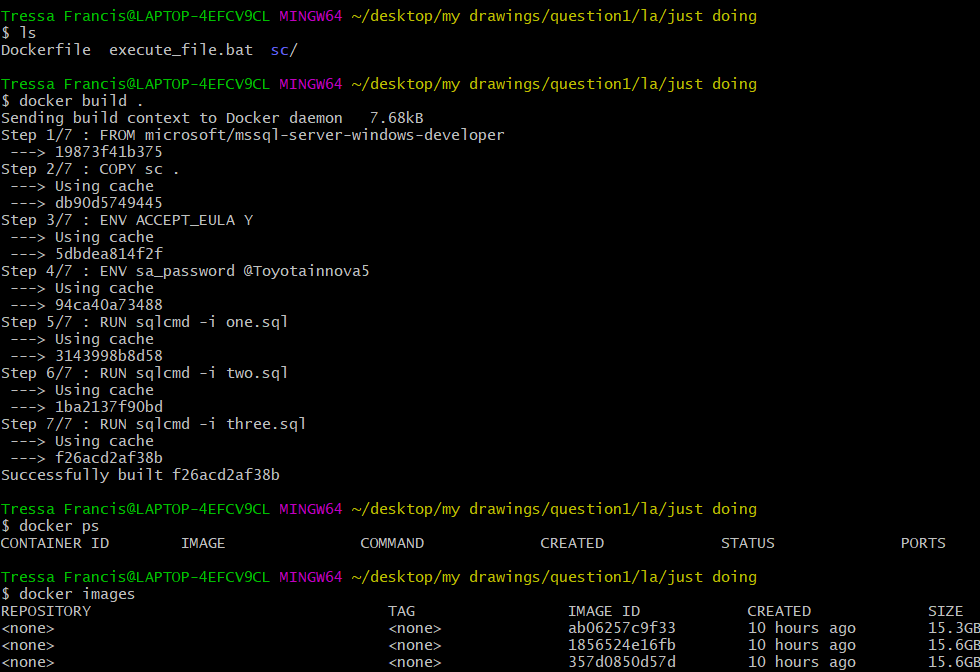
ENV ACCEPT\_EULA Y

ENV sa\_password @Toyotainnova5

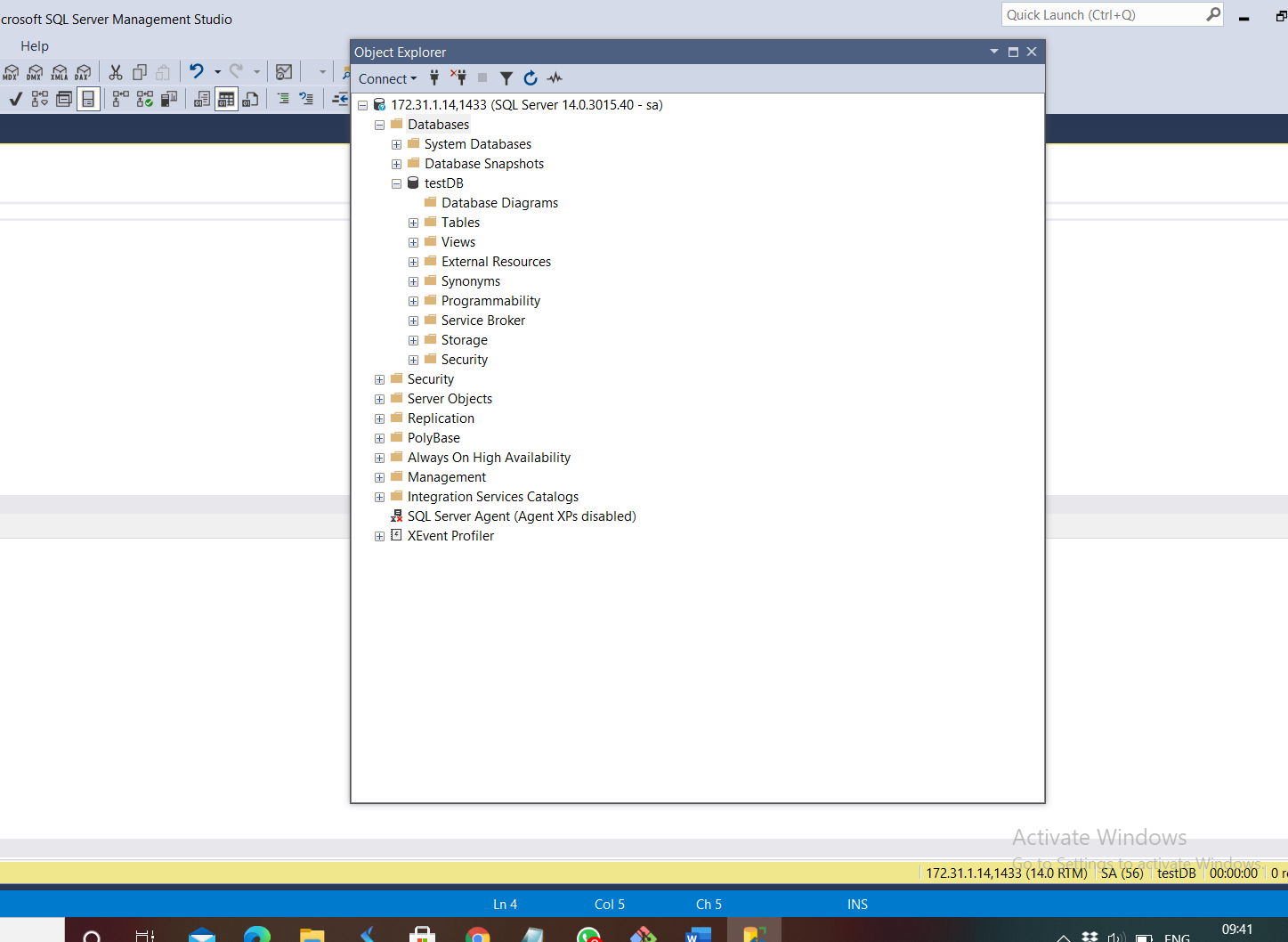
RUN sqlcmd -i one.sql

RUN sqlcmd -i two.sql

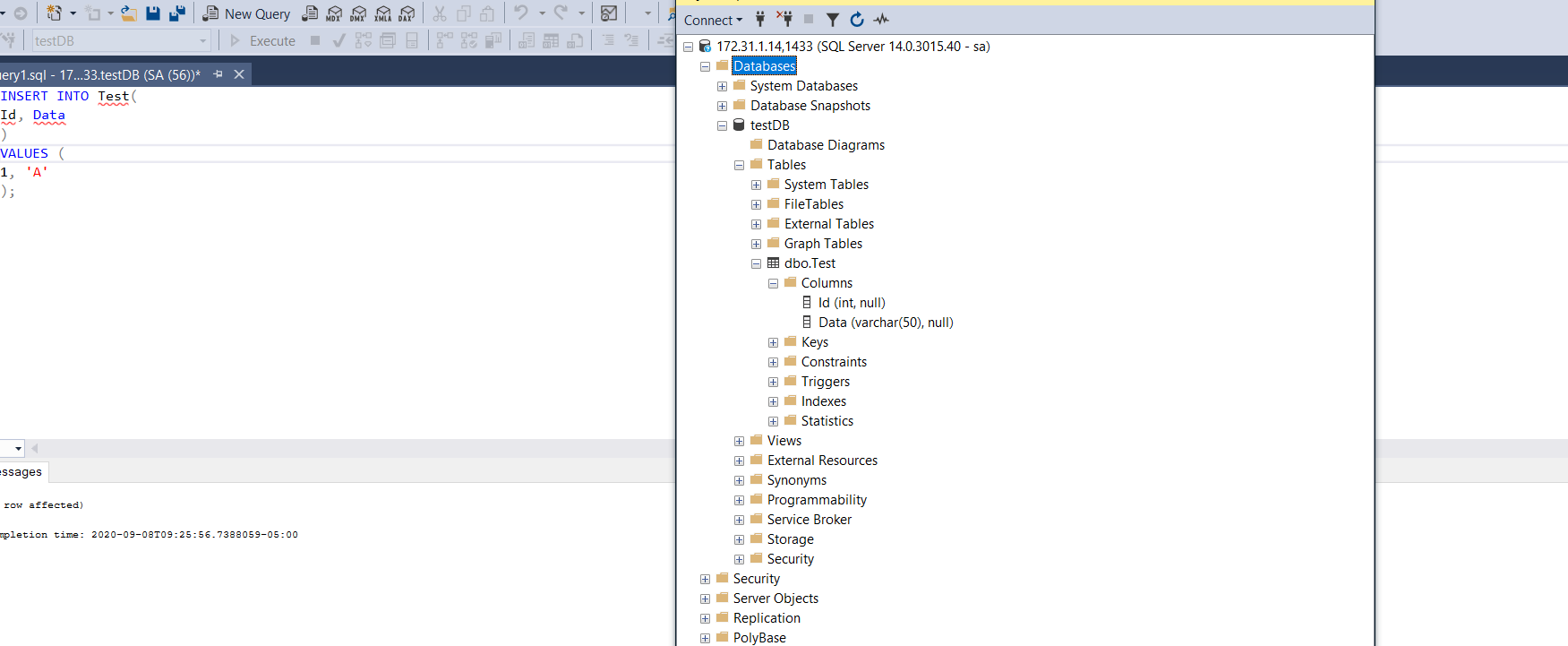
RUN sqlcmd -i three.sql



1. $ docker run -d -p 1433:1433 --name finally1 f26acd2af38b # used to run the container by exposing the SQL server developer edition to the host machine on port 1433
2. $ docker inspect finally1 | grep "IPAddress" # IP address of the container to connect SSMS to it via host



testDB is created. Table was created and values were inserted into it



1. Write a Python program to build an automated Data Pipeline. The program should be a single file python code and should be able to do the following.
   1. Navigate to a particular folder and execute other Python scripts which contains some kind of Data Ingestion / Transformation
   2. Take the following as command line input (bash style preferred)
      1. A folder name – Mandatory - The name of the folder to work upon
      2. Script Order – Optional - Ascending / Descending – The list of files order to work upon. Default Behavior – Ascending
      3. File name – Optional – The exact file name to run. Default Behavior – All the files
      4. Outfile – Optional – Log the output to the mentioned file name. Default behavior – Console
      5. Help – Optional - Display the help and usage information
   3. The following should be displayed as Output (beautification / Table preferred)
      1. The Folder where it is executed
      2. The file/files it is executed (in the order of execution)
      3. The output of the files executed. Just the console output returned should be okay

