# Project: Showcasing Critical Hospital Statistics Across California

**Objective**: The purpose of this project is to showcase Hospital statistics from facilities across the state of California. This will be done by creating robust Power BI and Excel dashboards and reports, which provide easy-to-read insight into how hospitals across California are performing. Data will be imported from both the California Department of Health and Human Services (CHHS) and Hospital Compare via API requests in Python and through SQL stored procedures, and the data will be stored in SQL Server databases and tables. All data, documentation, and scripts for the project are available in my Github repository, found at: <https://github.com/smarquee/Repo/>.

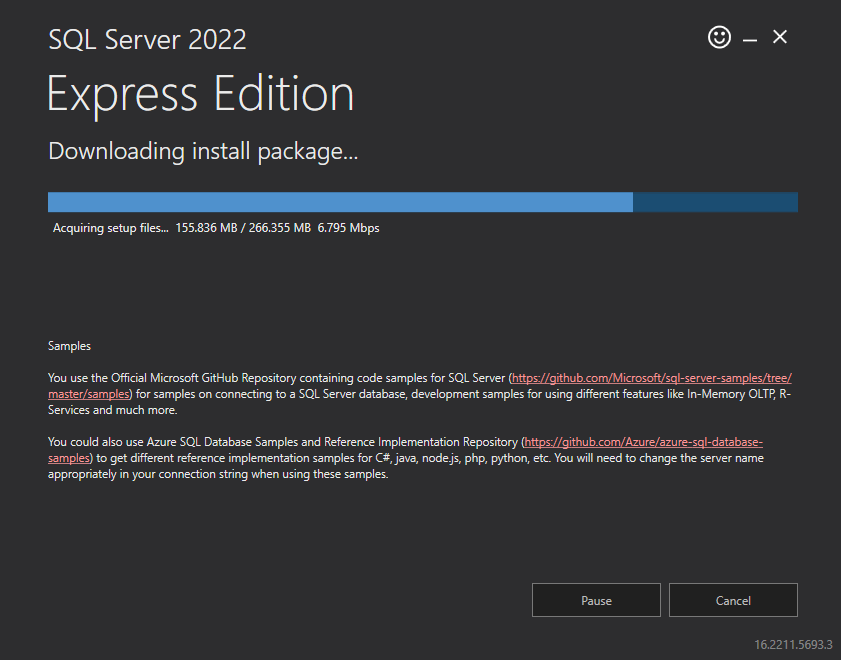
Besides showcasing hospital data, this project aims to demonstrate how to complete several common data engineering tasks, using tools such as Python, SQL, Power BI, and PowerShell. The tasks this project aims to complete are found below.

* Install Microsoft SQL Server
* Create New SQL Instance and Install SSMS
* Create New SQL Database
* Create New SQL Table with Primary Key Constraints
* Import Excel Dataset to SQL Table via Stored Procedure
* Import Dataset to SQL Table Using API Connection
* Create Power BI Data Model using Kimball Star Schema Methodology
* Connect Excel Report to Power BI Data Model for Automated Refreshes
* Create Power BI Dashboard
* Create PowerShell Script to Automate Power BI Refreshes
* Create PowerShell Script to Automatically Refresh Excel Report with Power BI Data
* Create PowerShell Script to Send Email with Attachments
* Create PowerShell Scrip t to Upload to WinSCP SFTP Site
* Create Windows Task Scheduler Job to Schedule PowerShell Script

## Install Microsoft SQL Server

1. To begin working with databases, SQL Server needs to be installed. I installed the Express Edition, since this version is free and it satisfies the needs of this project. The link to the download page can be found here:

<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>



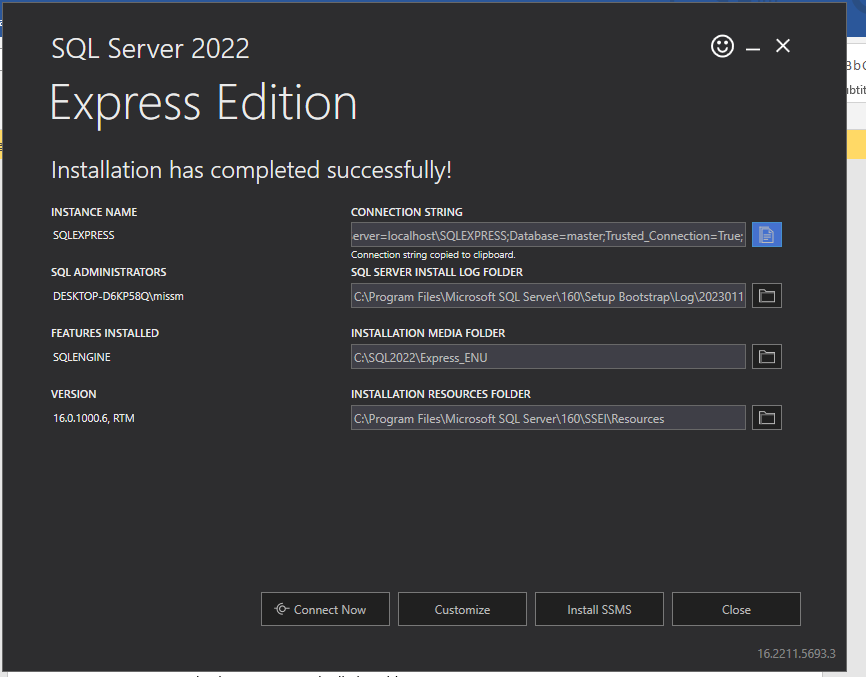
## Create New SQL Instance and Install SSMS

1. The new SQL instance was created from within the installer. Here are some important details about the database instance. See below for the page that displayed at the end of SQL Server installation. As can be seen from the screenshot, a new SQL instance was created, and important data was provided such as the instance name, log folder, administataror information, and version.

**Instance Name**: SQLEXPRESS

SQL Server Install Log Folder: C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap\Log\20230111\_201134

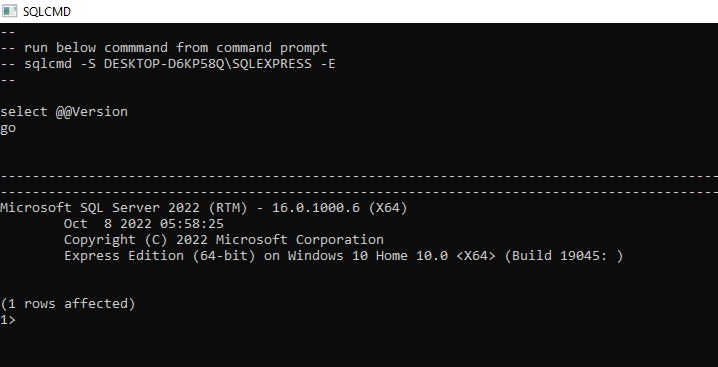
**Connection String**: Server=localhost\SQLEXPRESS;Database=master;Trusted\_Connection=True;



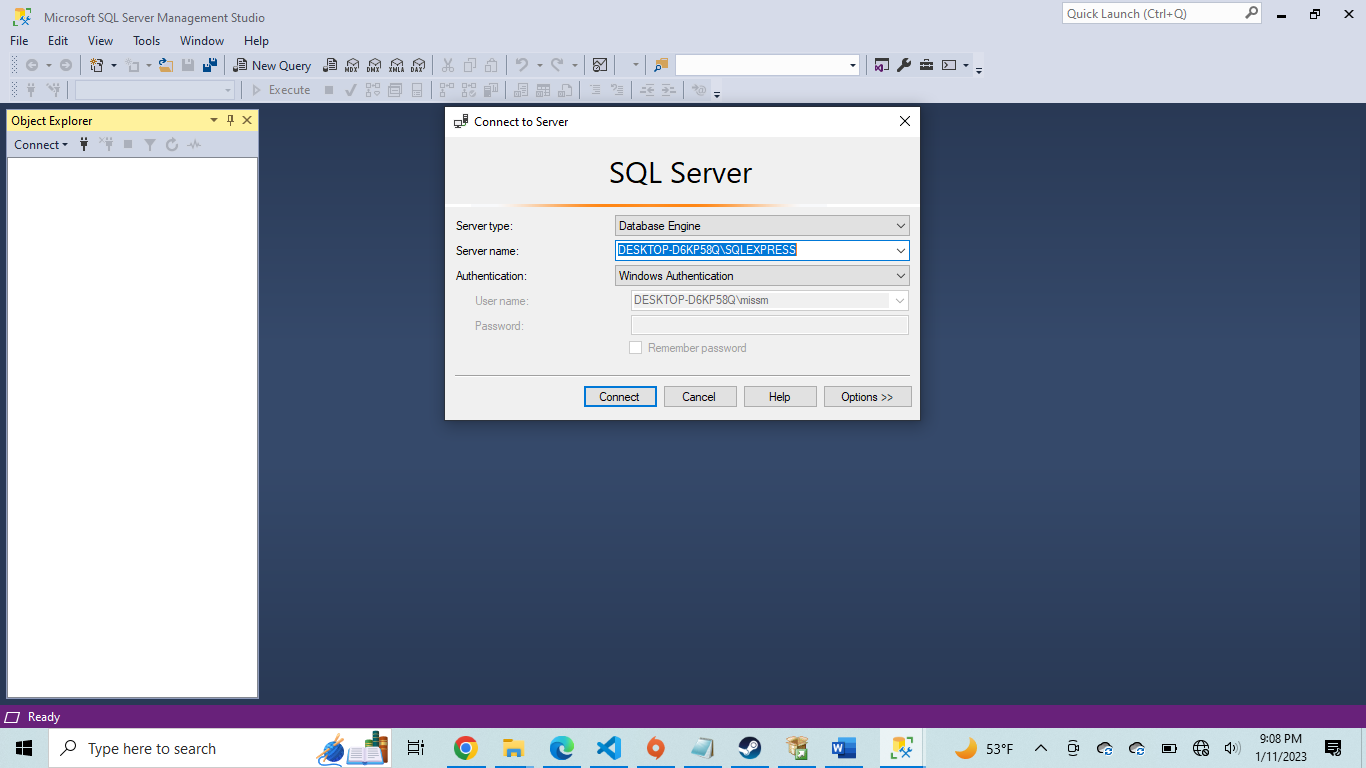
1. It is necessary to have a Relational Database Management System (RDBMS) to be able to connect to the SQL instance. Using this RDBMS, we will be able to create a database and start interacting with it using SQL queries. For SQL Server, the appropriate RDBMS to use is SQL Server Management Studio. To install SSMS, I clicked “Install SSMS” on the SQL Server installation windows (see screenshot above). Otherwise, you can install it from its download page:

[Download SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS) | Microsoft Learn](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?redirectedfrom=MSDN&view=sql-server-ver16)

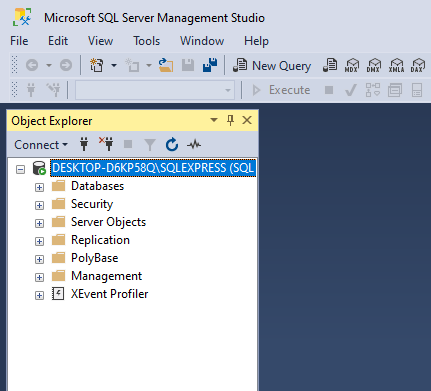
1. While SSMS was installing, I pressed the “Connect Now” button to connect to my new instance, SQLEXPRESS. As can be seen from the screenshot below, connection to the instance was successful, and we were able to return a version for the instance by running a SQLCMD.



1. After successful installation of SSMS, the newly downloaded app was opened, and the instance name “DESKTOP-D6KP58Q\SQLEXPRESS” was entered.

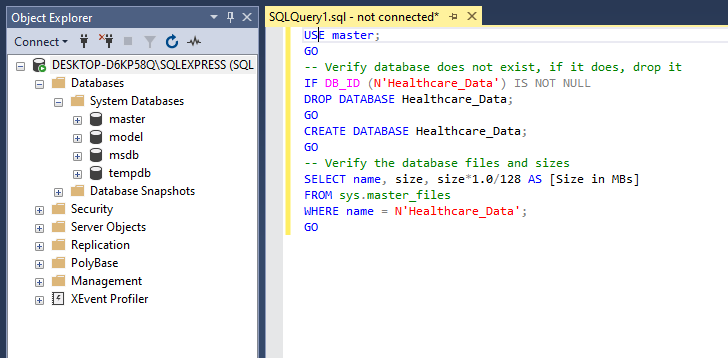


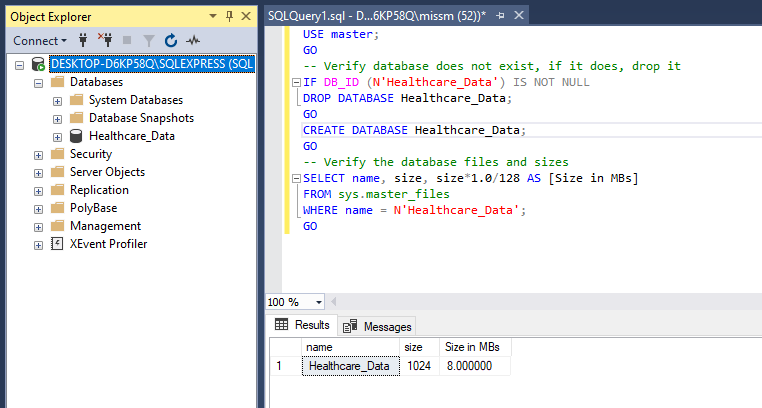
1. The connection was successful.



## Create New SQL Database

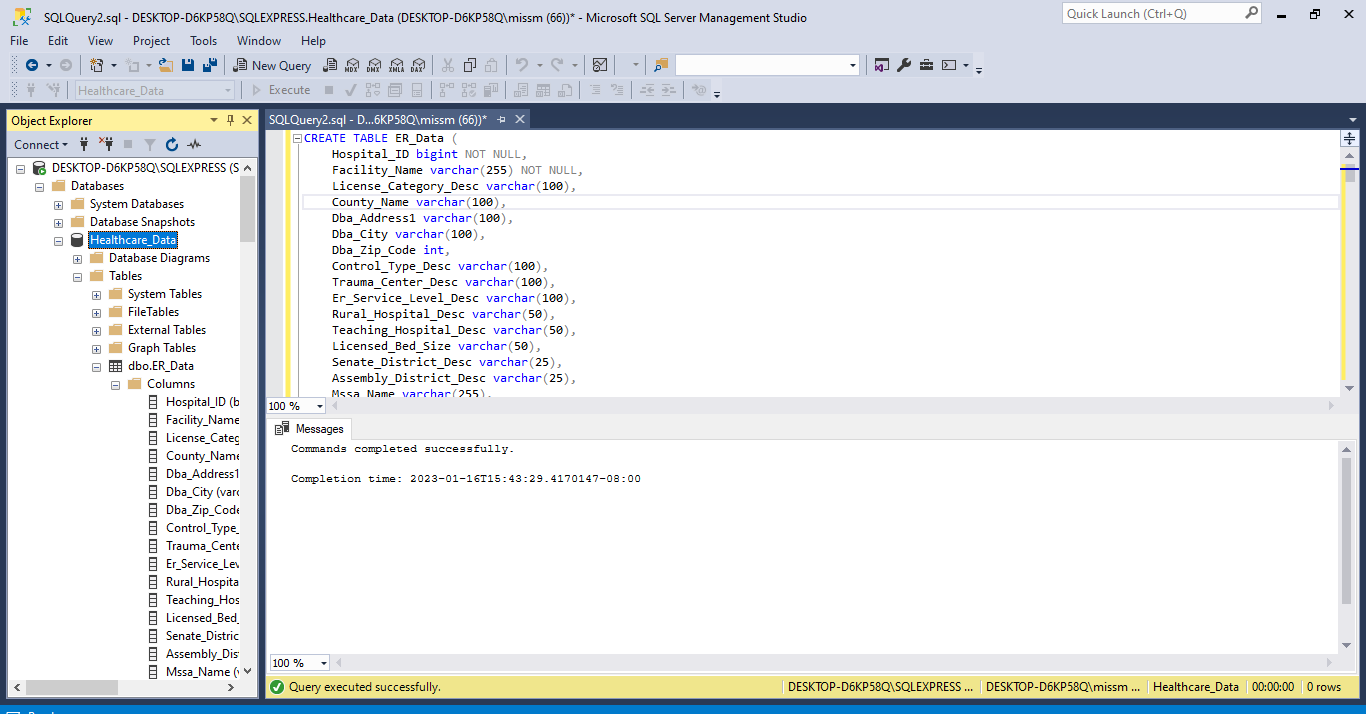
1. It is now time to create a new database. This new database is called Healthcare\_Data. The SQL script to create the database is stored in the Repo.





## Create New SQL Table with Primary Key Constraints

1. A new table was created called ER\_Stats. The SQL script to create the table is included in the Repo. This will be the main table that we will be working from to create our reports and dashboards on ER statistics.



## Link to ER\_Stats Table Dataset

The dataset is available on the CHHS website at: <https://data.chhs.ca.gov/dataset/hospital-emergency-department-characteristics-by-facility-pivot-profile/resource/2ed730f2-0dcd-4d4e-9503-422d704ed969>

## Import Excel Dataset to SQL Table via Stored Procedure

1. One way to import data into the ER\_Stats SQL table from an Excel file is to use a SQL stored procedure and run that.

**Note**: While it is possible to just use the SSMS GUI to simply right-click the table and import an Excel file, this is not how we’re going to do it. Using the shortcuts in SSMS would provide a simple way to perform a *one-time* import of the Excel file, but this would not be helpful if we want to automatically and continuously import data from Excel files to a table. This is because we want to create a script that is reusable, so we can import as many Excel files into our ER\_Stats table as we want on a regular basis. Instead, we will create a stored procedure to import the Excel data, and that stored procedure can be scheduled via a scheduler like SQL Server Agent Jobs.