D211 Performance Assessment

SLM1– SLM1 Task 1: Data Analysis

Advanced Data Acquisition - D211  
PRFA – SLM1

André Davis  
Student ID: 010630641  
MSDA

Competencies  
4033.2.1 : Storytelling with Data  
The graduate communicates data insights to technical and nontechnical audiences.

4033.2.2 : Data Visualizations and Representations  
The graduate creates data representations to offer insight into an organizational problem.

4033.2.3 : Dashboards  
The graduate designs interactive dashboards to support executive decision-making.

Table of Contents

[A: DATA DASHBOARDS 3](#_Toc138916017)

[A1: BOTH DATA SETS 3](#_Toc138916018)

[A2: DASHBOARD INSTALLATION 3](#_Toc138916019)

[Prepare Performance Assessment Files for WGU Labs on Demand 3](#_Toc138916020)

[WGU Labs on Demand 5](#_Toc138916021)

[A3: DASHBOARD NAVIGATION 9](#_Toc138916045)

[A4: SQL CODE 12](#_Toc138916046)

[B: PANOPTO PRESENTATION 12](#_Toc138916047)

[B1: TECHNICAL ENVIRONMENT 12](#_Toc138916048)

[B2: DEMONSTRATE DASHBOARD FUNCTIONALITY 13](#_Toc138916049)

[B3: SQL SCRIPTS 13](#_Toc138916050)

[B4: DATA STREAMS 13](#_Toc138916051)

[B5: DATA POINTS 14](#_Toc138916052)

[B6: DATABASE CREATION 14](#_Toc138916053)

[B7: REFERENTIAL INTEGRITY 14](#_Toc138916054)

[C: WRITTEN REPORT 14](#_Toc138916055)

[C1: DASHBOARD ALIGNMENT 15](#_Toc138916056)

[C2: BUSINESS INTELLIGENCE TOOL 15](#_Toc138916057)

[C3: DATA CLEANING 15](#_Toc138916058)

[C4: DASHBOARD CREATION 15](#_Toc138916059)

[C5: DATA ANALYSIS RESULTS 15](#_Toc138916060)

[C6: ANALYSIS LIMITATIONS 15](#_Toc138916061)

[D: WEB SOURCES 15](#_Toc138916062)

[E: SOURCES 15](#_Toc138916063)

[F: PROFESSIONAL COMMUNICATION 16](#_Toc138916064)

# A: DATA DASHBOARDS

The data dashboards support executive decision-making.

## A1: BOTH DATA SETS

Both data sets are accurate and complete and support the creation of dashboards for executive decision-making.

The datasets that are being used for to support the Tableau Professional Dashboard are as follows:

1. Within the WGU Labs on Demand has provided two datasets, namely "churn" and "medical\_data," available within PostgreSQL pgAdmin. From these datasets, I have selected the "medical\_data" database to establish a data connection with Tableau Professional for the purpose of creating my dashboards.

2. As part of my ongoing data analysis, I have opted to utilize the same data source utilized in D210. This dataset, known as ['Hospital\_General\_Information\_2016\_2020](https://www.kaggle.com/datasets/abrambeyer/us-hospital-overall-star-ratings-20162020)' (ABeyer, 2021), can be accessed on [kaggle.com](https://www.kaggle.com/). To facilitate the integration of this dataset into my analysis workflow, a straightforward Python script was developed. The script efficiently handles the ingestion of the CSV dataset, enabling the creation of a corresponding database and table within PostgreSQL pgAdmin, with the dataset's information loaded accordingly.

## A2: DASHBOARD INSTALLATION

The step-by-step instructions to guide users through the dashboard installation process is logical and accurate, and the instructions include all steps in the dashboard installation process.

### Prepare Performance Assessment Files for WGU Labs on Demand

As a Performance Assessor at WGU, it is essential to retrieve the submitted files and transfer them to a cloud-based service, such as your WGU Email Google Drive. This process involves organizing the files within a designated folder accessible through the WGU Lab on Demand. Once the files are stored in the appropriate location, they can be easily accessed and downloaded when necessary.  
  
Result should look like:  
A screenshot of a computer

Description automatically generated with medium confidence

### WGU Labs on Demand

### Log-in to the WGU Labs on Demand Environment

### Once logged in, open Microsoft Edge and log-in to your cloud storage such as WGU Email Google Drive.

### Download the folder in which the assessment files were uploaded to. A screenshot of a computer Description automatically generated with medium confidence

### Extract the folder if download in zip format to a location such as the ‘Desktop’ Screenshot showing the accessor that the contents from their cloud has been downloaded to the desktop.

### Copy the folder location to clipboard:

### Double-click and open the desktop folder.

### Use breadcrumb bar to copy the directory location. A screenshot of a computer Description automatically generated

### Open Windows PowerShell ISE as ‘Administrator’: A screenshot of a computer Description automatically generated

### Run the following commands, replacing <clipboard content from step C:ii>:

|  |  |
| --- | --- |
| Command 1 | Set-Location "<paste downloaded directory here>" |
| Command 2 | python.exe .\setup-additional-data-housing\_ratings.py u:postgres p:Passw0rd! |

### Screen shot capturing the information needed to execute the scripts for setting up the PostgreSQL with the additional dataset found from Kaggle.

### Confirm that the PostgreSQL database has been updated with a new database, table, and data.

### Open pgAdmin4

### Navigate to hospital\_ratings -> ratings and show all rows. Screen shot showing how to verify that PostgreSQL has the new database, table, and data via pgAdmin4.

### Confirm data has been imported from csv to PostgreSQL table: Screen shot showing how to confirm the number of rows present in the new table after the Python scripts have run. Number should match 25082.

### Load Performance Assessment Tableau file:

### Navigate back to the File Explorer that contains student files on the desktop. Confirm these two files exist:

### ‘Medical Data and Ratings.hyper’

### ‘D211-PA-SLM1 -TASK 1- DATA ANALYSIS (André Davis).twb’

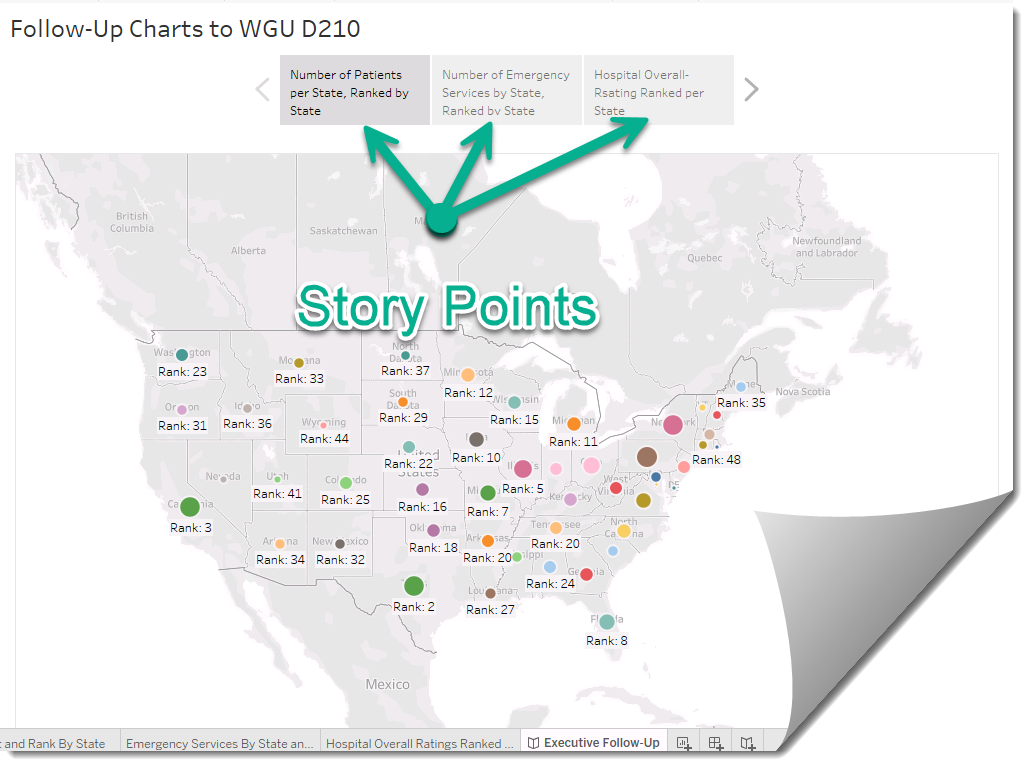
### Double-click ‘D211-PA-SLM1 -TASK 1- DATA ANALYSIS (André Davis).twb’ to launch the Performance Assessment dashboards in the Tableau application.

## A3: DASHBOARD NAVIGATION

The instructions to help users navigate the dashboards are clear and complete.

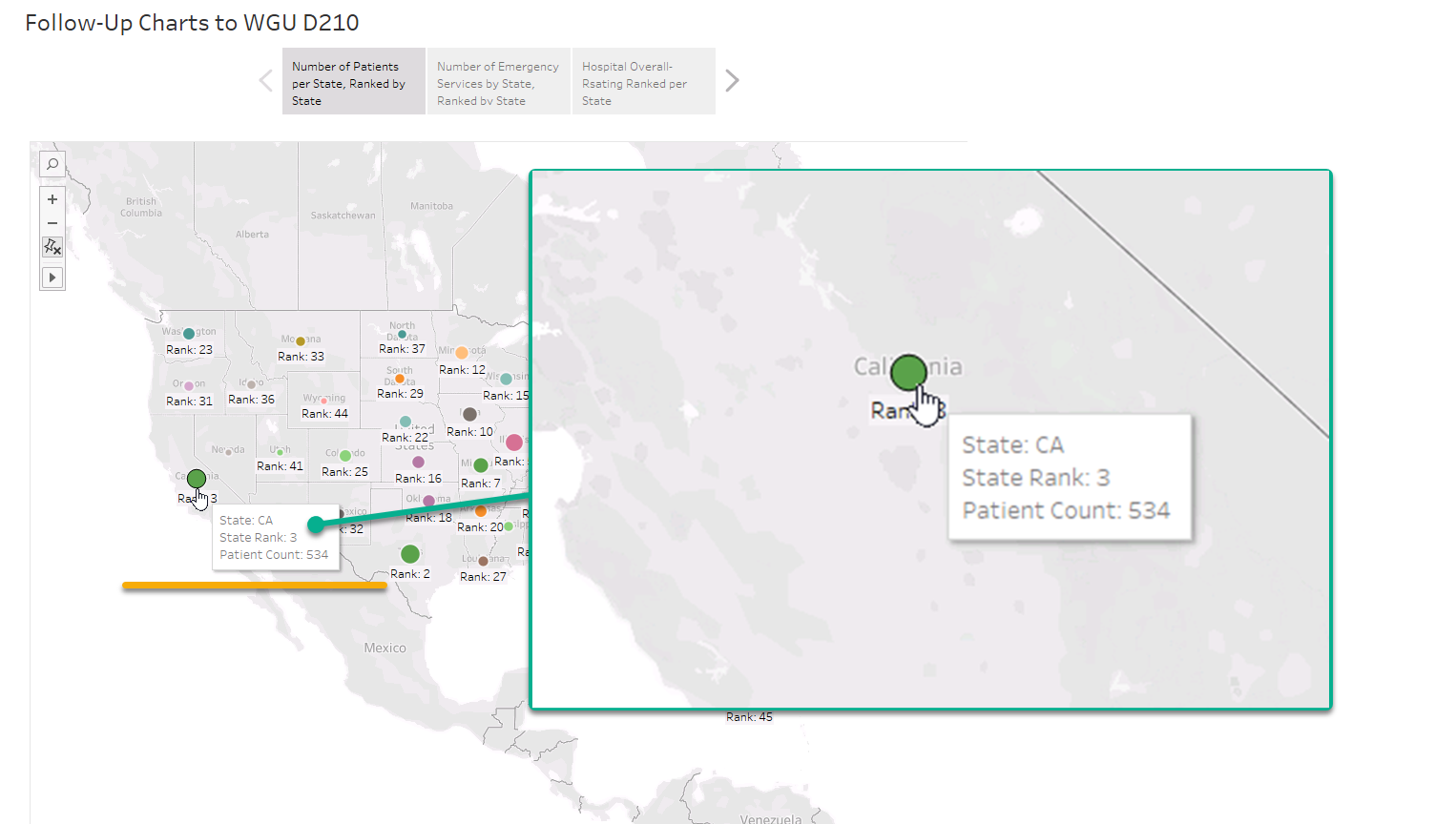
To facilitate navigation through the Dashboards, they have been consolidated into a Tableau Story titled 'Follow-Up Charts to WGU D210'. Within this screen, there are three Tableau Story Points that provide access to the Dashboards. These points are labeled as 'Number of Patients per State, Ranked by State', 'Number of Emergency Services by State, Ranked by State', and 'Hospital Overall-Rating Ranked per State'.

The purpose of this Tableau Story/Dashboard is to follow up with the executive team who attended the previous presentation on re-admissions. During that session, there were questions seeking clarification regarding patient counts, the number of emergency services, and hospital overall-ratings per state. The executive team desired additional details to assist them in further investigations, specifically to determine if areas with higher admission rates exhibited any correlation with these three factors and their geographical distribution.

Tableau Story /Dashboard Navigation  


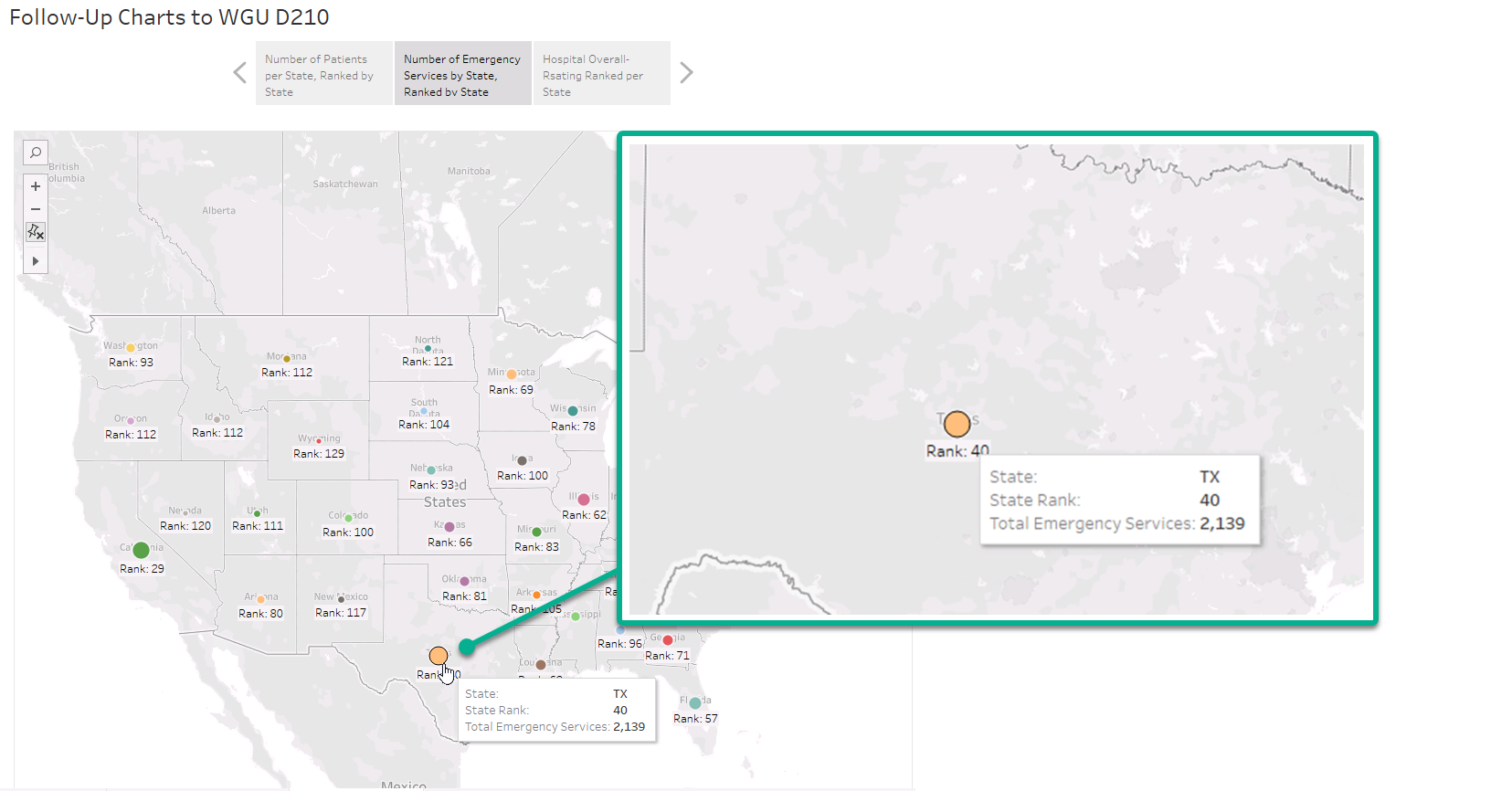
Number of Patients per State, Ranked by State

Interacting with this dashboard is intuitively straightforward. To obtain more information about a specific state, simply hover the mouse over the corresponding data-bubble displaying the state rank. As demonstrated in the example below, hovering the mouse cursor over the data-bubble triggers the appearance of a tooltip. This tooltip provides details such as the state name, state rank, and the corresponding patient count. In the illustration, we observe that California is currently being viewed, ranked 3rd with a patient count of 534. It is to be noted that a higher rank signifies that there is a more patients in that state.



Number of Emergency Services by State, Ranked by State

Interacting with this dashboard is also intuitively straightforward. To obtain more information about a specific state, simply hover the mouse over the corresponding data-bubble displaying the state rank. As demonstrated in the example below, hovering the mouse cursor over the data-bubble triggers the appearance of a tooltip. This tooltip provides details such as the state name, state rank, and the corresponding emergency services count. In the illustration, we observe that Texas is currently being viewed, ranked 40th with an emergency services count of 2,139. It is to be noted that a higher rank signifies that the state has more emergency services available.



Hospital Overall-Rating Ranked per State

This particular dashboard utilizes a more complex SQL query to generate its content. The dashboard offers the ability to apply filters based on Hospital Ratings ranging from 1 to 5, as well as "Not Available". Once a filter has been applied, a histogram chart is displayed, with each state represented by a line. This chart showcases the individual state rankings for Hospital Overall-Ratings. In other words, it reflects how each state's hospitals are rated by individuals, who can assign ratings from 1 to 5 or choose not to provide a rating. Each state then determines its own ranking based on the received ratings. For instance, if a rating of 5 is selected in the filter and we hover over Indiana (IN), we would observe that the Rating of 5 is the third most commonly assigned rating for that state.

A screenshot of a graph

Description automatically generated

## A4: SQL CODE

A copy of all SQL code and other code supporting the dashboards is accurate, complete, and correct.

### SQL Generated by Python Script ‘setup-additional-data-housing\_ratings.py’

NOTE: Because this code generates the insert statement for each row, I will give the first example of it and cutout the rest to not make this paper massive. The file ‘wgu-generated-sql-for-additional-dataset.sql’ is submitted with the Performance Assessment.

\\*

D211 - Generated SQL from script setup-additional-data-housing\_ratings.py

\*/

DROP DATABASE IF EXISTS hospital\_ratings;

CREATE DATABASE hospital\_ratings

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

DROP TABLE IF EXISTS ratings;

CREATE TABLE ratings("FacilityID" VARCHAR(50) NOT NULL,

"FacilityName" VARCHAR(500) NOT NULL,

"Address" VARCHAR(250) NOT NULL,

"City" VARCHAR(100) NOT NULL,

"State" VARCHAR(2) NOT NULL,

"ZipCode" VARCHAR(10) NOT NULL,

"CountyName" VARCHAR(50) NULL,

"Phone" VARCHAR(15) NOT NULL,

"HospitalType" VARCHAR(250) NOT NULL,

"HospitalOwnership" VARCHAR(250) ,

"EmergencyServices" BOOLEAN NOT NULL,

"MeetsCriteriaForInteropEHRs" BOOLEAN NOT NULL,

"HospitalOverallRating" VARCHAR(14) NOT NULL,

"MortalityNationalComparison" VARCHAR(30) NOT NULL,

"SafetyOfCareNationalComparison" VARCHAR(30) NOT NULL,

"ReadmissionNationalComparison" VARCHAR(30) NOT NULL,

"PatientExperienceNationalComparison" VARCHAR(30) NOT NULL,

"EffectivenessOfCareNationalComparison" VARCHAR(30) NOT NULL,

"TimelinessOfCareNationalComparison" VARCHAR(30) NOT NULL,

"EfficientUseOfMedicalImagingNationalComparison" VARCHAR(30) NOT NULL,

"Year" VARCHAR(4));

INSERT INTO ratings("FacilityID", "FacilityName", "Address", "City", "State", "ZipCode", "CountyName", "Phone", "HospitalType", "HospitalOwnership", "EmergencyServices", "MeetsCriteriaForInteropEHRs", "HospitalOverallRating", "MortalityNationalComparison", "SafetyOfCareNationalComparison", "ReadmissionNationalComparison", "PatientExperienceNationalComparison", "EffectivenessOfCareNationalComparison", "TimelinessOfCareNationalComparison", "EfficientUseOfMedicalImagingNationalComparison", "Year")  
VALUES(010001, 'SOUTHEAST ALABAMA MEDICAL CENTER', '1108 ROSS CLARK CIRCLE', 'DOTHAN', 'AL', 36301, 'HOUSTON', '(334) 793-8701', 'Acute Care Hospitals', 'Government - Hospital District or Authority', True, True, '2', 'Below the national average', 'Same as the national average', 'Below the national average', 'Below the national average', 'Same as the national average', 'Same as the national average', 'Same as the national average', 2020);

… rest of the inserts are removed for brevity, see submitted file ‘wgu-generated-sql-for-additional-dataset.sql’ for rest of insert statements.

### SQL Script: ‘patient-count-by-state-with-ranking.sql’

/\*

File: patient-count-by-state-with-ranking.sql

Student: André Davis

Student ID: 010630641

Performance Assesment: SLM1 — TASK 1: DATA ANALYSIS - Advanced Acquisition - D211

Description:

This script will count the number of patients per state and rank them.

\*/

WITH Patient\_Count\_By\_States AS (

SELECT l.state

,COUNT(p.\*) AS PatientCount

FROM patient AS p

INNER JOIN location AS l ON l.location\_id = p.location\_id

GROUP BY l.State

ORDER BY l.State

)

SELECT

pcbs.state

,pcbs.PatientCount

,DENSE\_RANK() OVER (ORDER BY pcbs.PatientCount DESC) AS Ranking

FROM Patient\_Count\_By\_States AS pcbs

ORDER BY state ASC, Ranking DESC;

### SQL Script: ‘emergency-services-count-by-state-ranked.sql’: /\*

File: emergency-services-count-by-state-ranked.sql

Student: André Davis

Student ID: 010630641

Performance Assesment: SLM1 — TASK 1: DATA ANALYSIS - Advanced Acquisition - D211

Description:

This script will count the number of emergencies service count per state and rank them.

NOTE: This will be excluding U.S. territories.

\*/

WITH Emergency\_Counts\_By\_State AS (

SELECT

rating."State" AS State

,COUNT(\*) AS EmergencyServiceCount

FROM public.ratings AS rating

GROUP BY rating."State", rating."EmergencyServices"

ORDER BY rating."State"

)

SELECT ecbs.State

,ecbs.EmergencyServiceCount

,DENSE\_RANK() OVER(ORDER BY ecbs.EmergencyServiceCount DESC) AS Rank

FROM Emergency\_Counts\_By\_State AS ecbs

WHERE ecbs.State NOT IN ('GU', 'PR', 'VI', 'MP', 'AS') --Exclude Territories

ORDER BY ecbs.State, Rank

SQL Script: ‘hopsitaloverallranking-per-state-and-rank.sql’:

/\*

File: hospitaloverallranking-per-state-and-rank.sql

Student: André Davis

Student ID: 010630641

Performance Assesment: SLM1 — TASK 1: DATA ANALYSIS - Advanced Acquisition - D211

Description:

Each state has an Hospital Overall Ranking which is on a scale of 1-5 or "Not Available".

This query groups the Hospital OVer All Rankings into their state buckets. From there with Paritioning

the Rankings are applied to the Hospital Overall Rankings to Rank the scaling.

Example:

Alaska Hospital Overall Rating Scores:

Scale: Rank For State:

------ ---------------

1 1

2 5

3 6

4 4

5 2

"Not Available" 3

In this example a score of 1 took the Rank of 1 meaning that their top score is a Hospital Overall Raking of 1.

This means they worst score is their most selected rating and that should be looked into.

\*/

WITH State\_OverallRanking\_By\_State\_And\_Rating AS (

SELECT rating."State" AS State

,rating."HospitalOverallRating" AS HospitalOverallRating

,COUNT(rating."HospitalOverallRating") AS HospitalOverallRatingCount

FROM public.ratings AS rating

GROUP BY rating."State", rating."HospitalOverallRating"

ORDER BY rating."State"

)

SELECT

sobsar.State

,sobsar.HospitalOverallRating

,RANK() OVER (PARTITION BY sobsar.State ORDER BY sobsar.HospitalOverallRatingCount) AS Ranking

FROM State\_OverallRanking\_By\_State\_And\_Rating AS sobsar

ORDER BY sobsar.State ASC, Ranking DESC

# B: PANOPTO PRESENTATION

The link connects to the Panopto multimedia presentation.

## B1: TECHNICAL ENVIRONMENT

The description of the technical environment used to create the dashboards is both complete and accurate.

The environment being used for D211 is the WGU Labs on Demand. It includes the following technologies to support this performance assessment:

1. Windows 10 Virtual Machine
2. PostgreSQL & pgAdmin4
3. PowerShell
4. Python 3
5. Tableau Professional Edition

## B2: DEMONSTRATE DASHBOARD FUNCTIONALITY

The submission fully demonstrates the functionality of each dashboard.

Dashboards include:

1. Patient Count and Rank By State
2. Emergency Services By State and Rank
3. Hospital Overall Ratings Ranked Per State

Story:

1. Follow-Up Charts to WGU D210
   1. Tabs
      1. Number of Patient per State, Ranked by State
      2. Number of Emergency Services by State, Ranked by State
      3. Hospital Overall Rating Ranked Per State

## B3: SQL SCRIPTS

The explanation of the SQL scripts used to support the creation of the dashboards is accurate and complete.

1. After running the ` *setup-additional-data-housing\_ratings.py*` it will produce a SQL file called ` wgu-generated-sql-for-additional-dataset.sql ` for review of the code used to create the PostgreSQL Database, Table, and the data insertion.
2. There are three SQL files that contain the Custom SQL used in Tableau they are:
   1. patient-count-by-state-with-ranking.sql
   2. emergency-services-count-by-state-ranked.sql
   3. hopsitaloverallranking-per-state-and-rank.sql

## B4: DATA STREAMS

The explanation of how the data streams were prepared to support the analysis is accurate, complete, and logical.

|  |  |
| --- | --- |
| Data Stream | Preparation |
| PostgreSQL – medical\_data | The WGU - Lab On Demand environment is equipped with the pre-installed medical\_data database. This database contains the same data that I have utilized in previous WGU courses. In the previous courses, the data was presented in a CSV file format, whereas now it is organized in a Relational Database, ensuring appropriate referential integrity. |
| Kaggle – Hospital\_General\_Information\_2016\_2020.csv | 1. Download the CSV dataset from kaggle.com [here](https://www.kaggle.com/datasets/abrambeyer/us-hospital-overall-star-ratings-20162020). 2. Created a Python script that cleaned the data in a Pandas DataFrame 3. Used the Python package ‘[psycopg2](https://pypi.org/project/psycopg2/)’ to connect to PostgreSQL and use SQL commands to create the database, table, and insert the data. 4. Saved generated SQL to a file for review within this paper. |

## B5: DATA POINTS

The description of how data were aligned with other data points is accurate, logical, and complete.

## B6: DATABASE CREATION

The demonstration of how the databases were created is both accurate and complete.  
  
Database creation for the WGU D211 Performance Assessment is as follows:

1. WGU Data sources come pre-installed on the WGU Labs on Demand environment.
   1. Database
      1. churn
      2. medical\_data (my selection)
2. Additional External dataset:
   1. The additional dataset that was selected is from [kaggle.com](https://www.kaggle.com/) called ‘[U.S. Hospital Overall Star Ratings 2016 - 2020](https://www.kaggle.com/datasets/abrambeyer/us-hospital-overall-star-ratings-20162020)’
   2. The Kaggle data is in CSV format. I’ve created scripts to convert this into a format that can be imported into PostgreSQL. The file that performs this is written in Python (setup-additional-data-housing\_ratings.py). It creates a new Database, then table and imports the CSV data into the new table.

## B7: REFERENTIAL INTEGRITY

The explanation of how referential integrity was enforced in the database is accurate and complete.

# C: WRITTEN REPORT

The writing accurately outlines the data exploration, the use of advanced SQL operations, and the analysis of the data.

## C1: DASHBOARD ALIGNMENT

The explanation is accurate, logical, and complete.

## C2: BUSINESS INTELLIGENCE TOOL

The justification of the selection of the business intelligence tool used is accurate, logical, and complete.

## C3: DATA CLEANING

The explanation of the steps used to clean and prepare the data for the analysis is accurate, logical, and complete.

## C4: DASHBOARD CREATION

The summary of the steps used to create the dashboards is accurate, logical, and complete.

## C5: DATA ANALYSIS RESULTS

The submission is accurate, logical, and complete, and it discusses how the results of the data analysis support executive decision-making.

## C6: ANALYSIS LIMITATIONS

The discussion of the limitation(s) of the data analysis is accurate and complete.

# D: WEB SOURCES

The record of the web sources used to acquire data or segments of third-party code to support the application is both complete and accurate, and the web sources cited are reliable. Or no web sources are used to acquire data or segments of third-party code, and the submission states this.

# E: [SOURCES](https://lrps.wgu.edu/provision/147882373)

The submission includes in-text citations for sources that are properly quoted, paraphrased, or summarized and a reference list that accurately identifies the author, date, title, and source location as available.

*PSQL*. PostgreSQL Documentation. (2023, May 11). <https://www.postgresql.org/docs/current/app-psql.html>

ABeyer. (2021, May 26). *U.S. hospital overall star ratings 2016-2020*. Kaggle. <https://www.kaggle.com/datasets/abrambeyer/us-hospital-overall-star-ratings-20162020>

*Reference materials*. Tableau. (n.d.).   
<https://www.tableau.com/resources/reference-materials>

# F: [PROFESSIONAL COMMUNICATION](https://lrps.wgu.edu/provision/27641407)

Content reflects attention to detail, is organized, and focuses on the main ideas as prescribed in the task or chosen by the candidate. Terminology is pertinent, is used correctly, and effectively conveys the intended meaning. Mechanics, usage, and grammar promote accurate interpretation and understanding.