ALY 6030 – DATA WAREHOUSING AND SQL

CRN: 70674   
WEEK 3 – ASSIGNMENT 3

SUBMITTED BY: AKHILA SINGARAJU

Singaraju.a@northeastern.edu

**INTRODUCTION**

In this assignment, I aimed at analyzing hospital bed capacity across various healthcare facilities. The objective was to create a structured database that encompasses critical information related to licensed, census, and staffed beds within hospitals. By developing a star schema using a fact table and dimension tables, I aimed to facilitate efficient data querying and analysis. Through this process, my goal was to derive valuable insights into hospital capacity, staffing levels, and overall resource allocation, ultimately contributing to improved decision-making in healthcare management.

**Selected Dataset**

To achieve these objectives, I selected a comprehensive dataset that includes data on hospitals, bed types, and their respective capacities. This dataset provided the foundation for my analysis, allowing me to explore various aspects of hospital bed management. The dataset includes details such as hospital names, total licensed beds, total staffed beds, and bed descriptions, enabling a thorough examination of the relationships between different variables.

**Tools and Languages**

In terms of tools and technologies, I utilized SQL for database creation and data manipulation, leveraging its capabilities to construct tables, execute queries, and perform aggregations. Additionally, I used an Entity-Relationship Diagram (ERD) tool to visualize the relationships between the different tables in my database. This combination of SQL and ERD visualization not only streamlined my workflow but also enhanced my understanding of data relationships within the healthcare context. By the end of this assignment, I expected to gain practical experience in database design and management while also uncovering meaningful insights from the data.

**RESEARCH**

**Task 1 : Identify Dimensions (first three only) from the two-dimension tables.**

From the two-dimension tables, business, and bed\_type, we can identify the first three attributes for each:

**Dimension Table 1: business**

* **Dimension 1:** ims\_org\_id — Unique identifier for each hospital or medical center.
* **Dimension 2**: business\_name — Name of the hospital or medical center.
* **Dimension 3:** ttl\_license\_beds — Total number of licensed beds allowed by the state license.

**Dimension Table 2: bed\_type**

* **Dimension** **1:** bed\_id — Unique identifier for each type of bed.
* **Dimension 2:** bed\_code — Short code representing the type of bed.
* **Dimension 3:** bed\_desc — Description of the type of bed (e.g., ICU, SICU, etc.).

**Task 2 : Identify the Facts (first three only) from the single fact table.**

From the fact table bed\_fact, we can identify the first three attributes:

**Fact Table: bed\_fact**

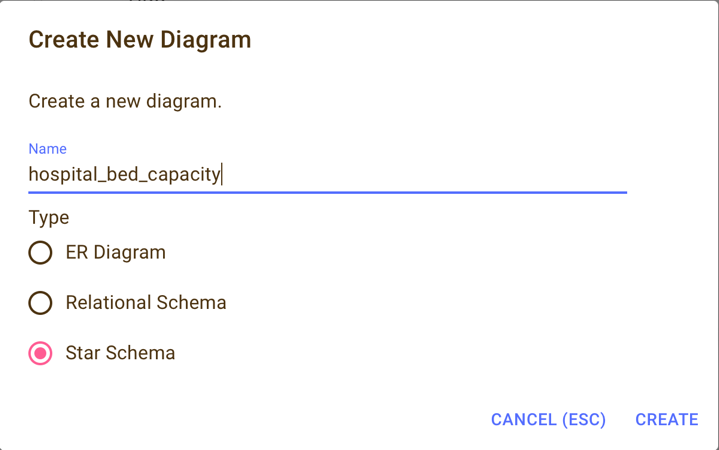
* **Fact 1: license\_beds**: This represents the total number of licensed beds a hospital is allowed to have according to state regulations. It indicates the maximum bed capacity legally permitted.
* **Fact 2: census\_beds**: This reflects the total number of actual beds present in the hospital. These are the beds currently available for patient use.
* **Fact 3: staffed\_beds**: This fact captures the total number of beds for which the hospital has appropriate staffing, such as physicians and nurses, to operate effectively.

**Task 3 : Creating star-schema**

**Creating Fact Table**

**Step 1:**

I initiated the process of creating a star schema by selecting the "Star Schema" option in the [ERDPlus](https://erdplus.com/#/standalone) tool. I named the diagram "hospital\_bed\_capacity" to reflect the focus of the analysis on hospital bed capacity and utilization. This step allowed me to set the foundation for organizing the necessary tables and defining the relationships between them.



**Step 2:**

For the second step in creating my star schema, I clicked on the **"Fact"** option within the ERDPlus tool. This action allowed me to define the fact table that will store the quantitative data relevant to my analysis.

By selecting the **"Fact"** option, I was able to create the bed\_fact table.

A screenshot of a computer

Description automatically generated

**Step 3:**

For the next step, I renamed the table to bed\_fact to accurately reflect its purpose in my star schema. Then, I added the necessary columns by clicking on the "Add" option under the Columns section.

The columns I included are as follows:

* license\_beds (INT): This tracks the number of licensed beds in each facility.
* census\_beds (INT): This keeps track of the total census beds in the hospital.
* staffed\_beds (INT): This indicates the number of staffed beds available for patient care.

Below is a screenshot of the bed\_fact table with the columns I added:

A screenshot of a phone

Description automatically generated

**Step 4:**

Below is the bed\_fact table that is created. It contains of 3 columns: license\_beds, census\_beds, staffed\_beds.

A close-up of a table

Description automatically generated

**Creating bed\_type dimension table**

**Step 1:**

After setting up the fact table, I proceeded to create the dimension table named bed\_type, I clicked on the **"Dimension"** option within the ERDPlus tool. This action allowed me to define the dimension table that will store the quantitative data relevant to my analysis.

By selecting the **"Dimension"** option, I was able to create the bed\_type table.

A screenshot of a computer

Description automatically generated

**Step 2:**

I renamed the table to bed\_type to accurately reflect its purpose. Then, I utilized the **Add** option under the columns section to include the necessary attributes for this table.

The columns I added to the bed\_type table are as follows:

* **bed\_id:** This column serves as the unique identifier for each type of bed.
* **bed\_code:** This column contains a short code representing the type of bed, providing a concise reference.
* **bed\_desc:** This column provides a description of the bed type (e.g., ICU, SICU), which is essential for understanding the context of the data.

A screenshot of a phone

Description automatically generated

**Step 3:**

Next, I selected the **"Primary Key"**option to designate the primary keys for the bed\_type table.

A screenshot of a phone

Description automatically generated

**Step 4:**

Next, I designated bed\_id as the primary key for the bed\_type table. This choice was made because the bed\_id serves as a unique identifier for each type of bed, ensuring that there are no duplicates within the table. By setting bed\_id as the primary key, I can maintain data integrity and establish a reliable link between the bed\_type dimension table and the bed\_fact fact table.

A screenshot of a phone

Description automatically generated

**Step 5:**

Below is the bed\_fact table that is created. It contains of 3 columns: bed\_id, bed\_code, bed\_des, with bed\_id as the primary key.

A close-up of a code

Description automatically generated

**Creating business dimension table**

**Step 1:**

After setting up the bed\_type dimension table, I proceeded to create the business dimension table, I clicked on the **"Dimension"** option within the ERDPlus tool. This action allowed me to define the dimension table.

A screenshot of a computer

Description automatically generated

**Step 2:**

After creating the dimension table, I renamed the table to business and added the following columns using the Add option under the columns section:

* **ims\_org\_id** - This column serves as a unique identifier for each hospital or medical center.
* **business\_name** - This column holds the name of the hospital or medical center.
* **ttl\_license\_beds** - This column represents the total number of licensed beds allowed by the state license.
* **ttl\_census\_beds** - This column indicates the total number of census beds available at the hospital.
* **ttl\_staffed\_beds** - This column reflects the total number of staffed beds for which staffing (e.g., physicians and nurses) exists.
* **bed\_cluster\_id** - This column can be used to categorize hospitals into specific clusters or groups based on various criteria.

A screenshot of a phone

Description automatically generated

**Step 3:**

Next, I selected the **"Primary Key"**option to designate the primary keys for the business table.

A screenshot of a phone

Description automatically generated

**Step 4:**

I selected **ims\_org\_id** as the primary key for the **business** table. This decision was made because **ims\_org\_id** serves as a unique identifier for each hospital or medical center, ensuring that each entry in the table can be distinctly referenced.

A screenshot of a phone

Description automatically generated

**Step 5:**

Below is the business table that is created. It contains of 6 columns: ims\_org\_id, business\_name, ttl\_license\_beds, ttl\_census\_beds, ttl\_staffed\_bed, bed\_cluster\_id.

**A close-up of a list

Description automatically generated**

**STAR SCHEMA**

Next, I used the Connect option to establish relationships between the dimension tables and the fact table, which is crucial for creating the star schema. In ERDPlus, connecting tables is straightforward. I simply clicked on the Connect icon, which allows me to draw lines between tables.

A screenshot of a computer

Description automatically generated

To connect the dimension tables to the fact table, I first clicked on the bed\_type table and then drew a line to the bed\_fact table. I repeated this process for the business table, connecting it to the bed\_fact table as well. Each connection signifies a relationship where the primary keys from the dimension tables become foreign keys in the fact table.

When I made these connections, the system automatically created foreign key (FK) constraints in the bed\_fact table. This means that the ims\_org\_id from the business table and the bed\_id from the bed\_type table are now foreign keys in the bed\_fact table, establishing referential integrity.

A diagram of a bed type

Description automatically generated

The Schema hospital\_bed\_capacity, includes three main entities: the **Business table**, the **Bed Type table**, and the **Bed Fact table.**

### Entities:

1. **Business Table**: I defined the Business table with ims\_org\_id as the primary key. This table includes attributes such as business\_name, which represents the name of the hospital, and ttl\_license\_beds, indicating the total number of licensed beds authorized by the state. Additionally, it contains ttl\_census\_beds for the total number of census beds currently available and ttl\_staffed\_beds for the total number of staffed beds. There’s also a bed\_cluster\_id attribute, which could be used to categorize bed clusters within the facility.
2. **Bed Type Table**: I created the Bed Type table with bed\_id as the primary key. This table captures the different types of beds available in hospitals. It includes attributes like bed\_code, which is a short code representing the bed type, and bed\_desc, providing a description of the bed type, such as "Intensive Care Unit" or "Surgical Intensive Care Unit."
3. **Bed Fact Table**: This table contains critical data about bed capacity and occupancy. The attributes include license\_bed, indicating the number of licensed beds for a specific type, census\_beds, which reflects the number of beds currently occupied or available, and staffed\_beds, showing the number of beds staffed by healthcare professionals and ims\_org\_id and bed\_id as foreign keys.

### Relationships:

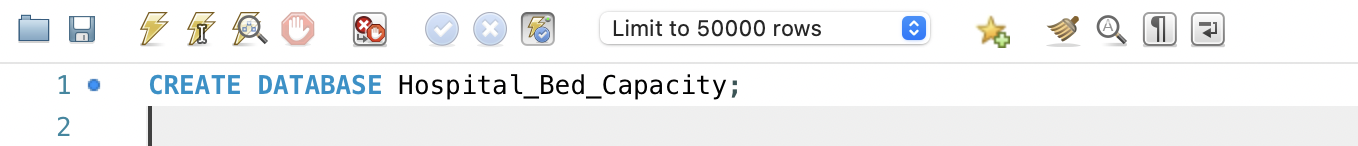
### The Bed Fact table serves as the central fact table, linking the Business and Bed Type dimension tables. This structure creates a star schema, which is effective for data warehousing.

1. I established foreign key relationships to maintain data integrity. The ims\_org\_id in the Bed Fact table references the Business table, allowing the fact data to be associated with the respective hospital. Similarly, the bed\_id in the Bed Fact table references the Bed Type table, enabling categorization of beds according to their types.

**Task 4 : Implementing the schema in SQL**

**Step1 : Creating Database**

The first step in the process was to create the database. I did this by executing the Create command and named it Hospital\_Bed\_Capacity. This initialized the environment where I would store all the necessary tables and data for the project.



**Step 2:**

After creating the database, I ensured that it was selected for use by executing USE Hospital\_Bed\_Capacity; to make sure that any subsequent commands would apply directly to this specific database. This set the foundation for building the entire schema and conducting further analysis.

A close up of a sign

Description automatically generated

**Step 3: Creating business dimension table.**

The next step was to create the business dimension table. I wrote the SQL command:

A computer code with text

Description automatically generated

This table stores essential details about each hospital or medical center, such as the organization's unique identifier (**ims\_org\_id**), the business name, and the total number of licensed, census, and staffed beds. I assigned **ims\_org\_id** as the primary key to ensure each business is uniquely identified, which is crucial for establishing relationships in the schema. This table serves as one of the dimension tables in the star schema model.

**Step 4: Importing data into business table.**

Next, I Imported data into the business dimension table. I right clicked on the business table and selected the Data Import Wizard option.

A screenshot of a phone

Description automatically generated

I clicked Browse and selected the business-1.csv file I downloaded for this assignment, in the select file page.

A screenshot of a computer

Description automatically generated

After selecting the file to import, I chose the use existing table option to select the business table I created in MySQL as destination to import the dataset.

A screenshot of a computer

Description automatically generated

Since I was importing a CSV file, the wizard prompted me to map the CSV columns to the corresponding columns in my **business** table. I reviewed the mappings carefully to ensure that each field in the CSV matched the correct column in the database table.

A screenshot of a computer

Description automatically generated

Once everything was correctly mapped, I clicked **Next** to start the import.

A screenshot of a computer

Description automatically generated

MySQL Workbench processed the file and inserted the rows into the **business** table. After the import was complete, I double-checked the data to make sure it was correctly inserted.

A screenshot of a computer

Description automatically generated

After importing, I see that 23147 records were correctly imported into the business table.

A screenshot of a computer

Description automatically generated

**Step 4: Creating bed\_type dimension table.**

Next, I moved on to creating the **bed\_type** table, which was my second dimension table for the project. To do this, I wrote the following SQL query.

A white background with black and white clouds

Description automatically generated

In this step, I defined **bed\_id** as the primary key, ensuring that each type of bed has a unique identifier. I also added the **bed\_code** (a short code for each bed type) and **bed\_desc** (a more detailed description of the bed, like ICU or SICU).

**Step 5: importing data into bed\_type dimension table.**

I Imported data into the bed\_typedimension table by using the Data Import Wizard option.

A screenshot of a phone

Description automatically generated

In the select file to import page, I selected bed\_type-1.csv file from my downloads to import to the table.

A screenshot of a computer

Description automatically generated

After selecting the file to import, I chose the use existing table option to select the bed\_type table I created in MySQL as destination to import the dataset.

A screenshot of a computer

Description automatically generated

The wizard prompted me to map the CSV columns to the corresponding columns in my **bed\_type**table. I reviewed the mappings carefully to ensure that each field in the CSV matched the correct column in the database table.

A screenshot of a computer

Description automatically generated

Once everything was correctly mapped, I clicked **Next** to start the import.

A screenshot of a computer

Description automatically generated

MySQL Workbench processed the file and inserted the rows into the **bed\_type** table.

A screenshot of a computer

Description automatically generated

A total of 20 records were imported to the bed\_type table after importing.

A screenshot of a computer

Description automatically generated

**Step 6: Creating bed\_fact table.**

The next step was to create the bed\_fact table, which is the central fact table in my star schema. To do this, I used the following SQL query:

**A computer code with text

Description automatically generated**

**Step 7: Importing data into bed\_fact table.**

I Imported data into the bed\_facttable by using the Data Import Wizard option.

A screenshot of a phone

Description automatically generated

In the select file to import page, I selected bed\_fact-1.csv file from my downloads to import to the table.

A screenshot of a computer

Description automatically generated

After selecting the file to import, I chose the use existing table option to select the bed\_fact table I created in MySQL as destination to import the dataset.

A screenshot of a computer

Description automatically generated

The wizard prompted me to map the CSV columns to the corresponding columns in my **bed\_type**table. I reviewed the mappings carefully to ensure that each field in the CSV matched the correct column in the database table.

A screenshot of a computer

Description automatically generated

Once everything was correctly mapped, I clicked **Next** to start the import.

A screenshot of a computer

Description automatically generated

MySQL Workbench processed the file and inserted the rows into the **bed\_type** table.

A screenshot of a computer

Description automatically generated

A total of 53770 records were imported to the bed\_type table after importing.

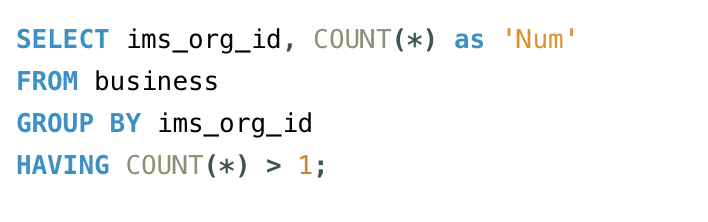
A screenshot of a computer

Description automatically generated

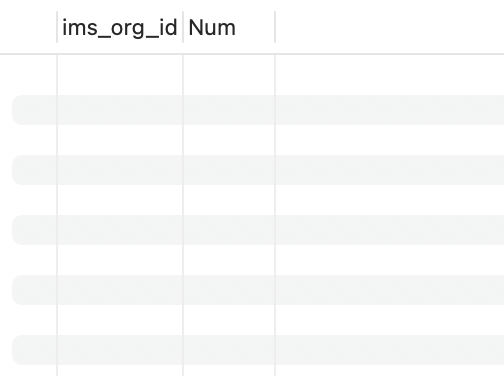
**Step 7: Checking for duplicates.**

For the next step in my assignment, I wrote a simple SQL query to check if there were any duplicate entries in the ims\_org\_id column of the business table. The goal was to identify if any hospital or medical center had more than one entry.

This query groups the data by ims\_org\_id and counts the number of records for each group. The HAVING COUNT(\*) > 1 clause ensures that only those entries with more than one record are displayed.



When I ran the query, the output was empty, meaning there were no duplicate entries in the **ims\_org\_id** field, which confirmed that each hospital had a unique identifier.



**1. Total number of licensed beds (total beds allowed by state license)**

I wrote a SQL query to calculate the total number of licensed, census, and staffed beds for ICU and SICU categories. The idea was to focus on those two bed types and rank the hospitals based on the total number of licensed beds.

This query first joins the bed\_fact, business, and bed\_type tables to extract relevant information. I used CASE statements to calculate the sum of licensed beds, census beds, and staffed beds for ICU and SICU separately. I also applied GROUP BY on the hospital name and ordered the results by the total number of licensed beds in descending order.

A computer code with text

Description automatically generated

The query returned the following output:

A table with numbers and letters

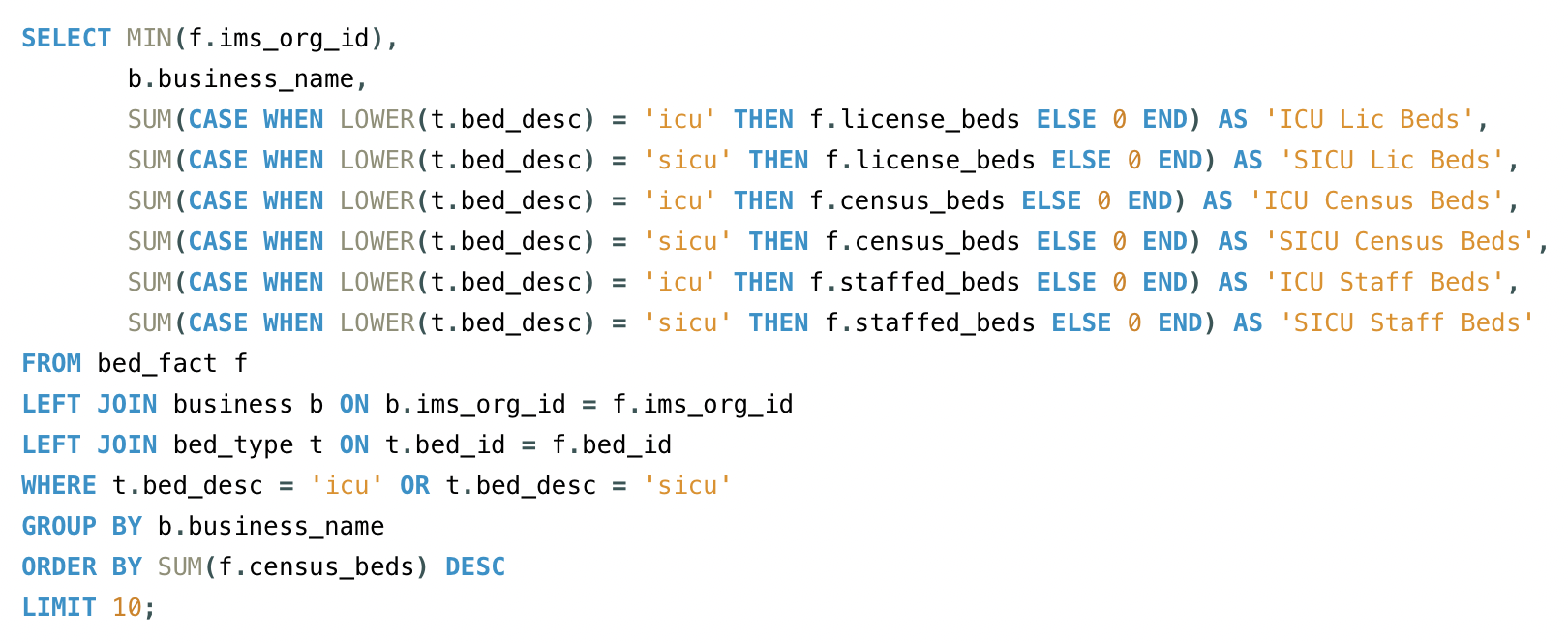
Description automatically generated

Each row represents a hospital, followed by its ICU and SICU bed counts for licensed, census, and staffed beds. For example, Saint Marys Hospital had 360 licensed ICU beds, 19 licensed SICU beds. This helped me address the task of identifying the top 10 hospitals based on licensed ICU and SICU beds.

2. Total number of  census beds (total beds at the hospital)

I wrote the below query to retrieve the top 10 hospitals or medical centers with the highest number of **census beds** for ICU and SICU, which represent the total number of beds available at the hospital for those units.

In my query, I joined the bed\_fact, business, and bed\_type tables to consolidate information on the types of beds (ICU and SICU) and their counts across various hospitals. To calculate the total number of census beds specifically for ICU and SICU, I utilized CASE statements to isolate these beds and then summed them up accordingly. The GROUP BY clause allowed me to organize the data by each hospital's name, while the ORDER BY clause ranked the hospitals based on their census bed counts in descending order.



This output provides the number of licensed, census, and staffed beds for ICU and SICU at each hospital. Saint Marys Hospital and Saint Anthony Hospital are at the top of the list with the highest number of census beds in their ICUs.

A screenshot of a computer

Description automatically generated

3. Total number of staffed beds (total beds for which staffing, e.g., physicians and nurses, exists)

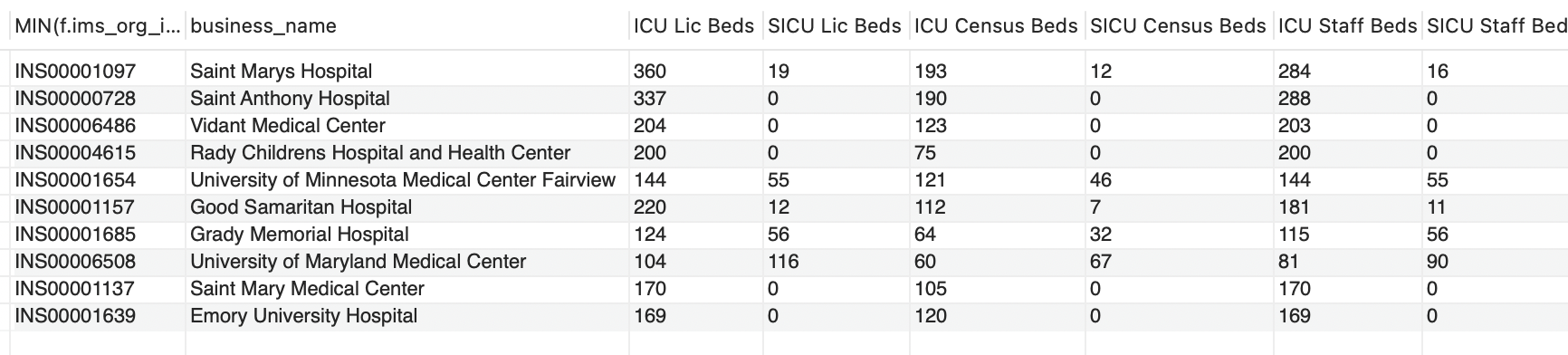
To analyze the total number of staffed beds in hospitals, I executed the below SQL query. This query pulls together data from the bed\_fact, business, and bed\_type tables to provide a comprehensive view of staffed beds, particularly in ICU and SICU units.

In the query, I started by selecting the ims\_org\_id from the bed\_fact table along with the corresponding business\_name from the business table. To isolate the staffed beds, I employed CASE statements to sum up the number of staffed beds separately for both ICU and SICU beds The grouping was done by business\_name to ensure that the results reflect the counts per hospital. I also included an order by clause to rank the hospitals based on their total staffed beds in descending order. This way, I could easily identify the hospitals with the highest number of staffed beds.

A computer screen shot of text

Description automatically generated

The output of my query provided a comprehensive view of the total number of staffed beds across various hospitals, alongside other relevant metrics.



In this output, each row corresponds to a hospital, identified by its unique ims\_org\_id, along with its name and the number of licensed, census, and staffed beds, for ICU and SICU.

For instance, Saint Marys Hospital has a total of 360 licensed beds and 284 staffed beds, indicating a robust capacity to provide care.

### Discussion of Results with Client As part of my analysis for your healthcare network, I’ve compiled a list of the top 10 hospitals that have Surgical Intensive Care Units (SICUs) and Intensive Care Units (ICUs) based on three key bed metrics: licensed beds, census beds, and staffed beds.

#### **1. Total Number of Licensed Beds**

#### Licensed beds refer to the maximum number of beds that a hospital is allowed to operate under state regulations. This is important because it sets the legal capacity for how many patients a hospital can accommodate. Based on the data, **Saint Marys Hospital** ranked the highest with 360 licensed ICU beds and 19 SICU beds. This indicates that Saint Marys has a robust capacity to handle critical patients, which is crucial for emergency preparedness and overall patient management. Other hospitals like Saint Anthony Hospital and Phoenix Children's Hospital also stood out in this category. By knowing which hospitals have the largest state-authorized capacity, your network can strategically plan patient transfers or expansion in times of high demand, ensuring that you are operating within legal limits while maximizing care.

#### **2. Total Number of Census Beds:** Census beds represent the actual number of beds that are occupied or available for patient care at the hospital. This is a good indicator of how effectively the hospitals are utilizing their resources at any given time. In the results, Saint Marys Hospital again ranked at the top, with 193 census ICU beds and 12 SICU beds actively available. This shows that Saint Marys is not just licensed for many beds but is also effectively using those beds for patient care. This metric is particularly useful for understanding real-time capacity and can be critical when your network needs to manage hospital bed availability during surges or normal operations.

#### **3. Total Number of Staffed Beds:** Staffed beds are those that have adequate personnel, such as physicians and nurses, available to care for patients. Having beds is one thing, but having the staff to support those beds is another, and that’s where this metric becomes important. Again, Saint Marys Hospital led the list with 284 ICU staffed beds and 16 SICU staffed beds, demonstrating that they not only have the infrastructure but also the personnel to support patient care. Hospitals with high staffed bed counts are well-positioned to handle patient influxes without stretching their human resources too thin. For your network, this is a vital metric to track, especially if you’re looking to improve staffing efficiencies or allocate resources to hospitals in need.

### Key Takeaways for Your Network

* **Saint Marys Hospital** consistently ranks at the top across all three categories—licensed, census, and staffed beds—making it a key facility for critical care.
* Other hospitals like **Saint Anthony Hospital** and **Phoenix Children's Hospital** also show strong capacity in licensed and census beds, though staffing levels may vary.
* Knowing the breakdown of ICU and SICU beds in terms of licensing, actual usage, and staffing will allow your network to make data-driven decisions about where to allocate resources, transfer patients, or plan for future expansions.

#### **CONCLUSION**

#### **Goals and Expectations**

Reflecting on this assignment, I can confidently say that my goals and expectations were largely met. I successfully created a comprehensive database that captures critical information regarding hospital bed capacity. Using SQL, I was able to construct various tables, execute queries, and extract meaningful insights from the data. The process of designing the database and establishing relationships between the fact and dimension tables provided me with valuable hands-on experience in database management.

#### **Pros and Cons of Tools, Methods, and Techniques**

However, I did encounter some challenges along the way. One of the main drawbacks of using SQL was its steep learning curve, particularly for more complex queries and aggregations. While SQL is powerful and efficient, mastering its syntax can be daunting for beginners. On the other hand, one of the significant advantages of using SQL is its ability to handle large datasets and perform complex operations efficiently, which proved invaluable for this assignment. Additionally, using an ERD tool helped me visualize the relationships between tables, making it easier to conceptualize the database structure and identify potential issues.

#### **Future Improvements**

If I were to approach this assignment again, I would allocate more time to exploring advanced SQL techniques, such as window functions and subqueries, to enhance the depth of my analysis. I also believe that incorporating data visualization tools, such as Tableau or Power BI, could provide even more insightful visual representations of the data, allowing for better interpretation of the results.

#### **Conclusion**

In summary, this assignment not only met my expectations but also enhanced my understanding of database design and management. The experience has encouraged me to continue developing my skills in data analysis and explore additional tools and techniques to further enrich my capabilities in the field.

**REFERENCES**

1. *Understanding Star Schema | DataBricks*. (n.d.). Databricks. <https://www.databricks.com/glossary/star-schema>
2. *W3Schools.com*. (n.d.-k). <https://www.w3schools.com/sql/sql_case.asp>
3. info@erdplus.com. (n.d.). *ERDPlus*. Mosor, Inc. <https://erdplus.com/>