**Part 1: CMS Data Exercise**

Here at Clipboard Health, we provide staffing to long-term care facilities. The Centers for Medicaid and Medicare Services publish a quarterly report containing daily staffing data for all registered nursing homes in the U.S. This dataset is called Payroll Based Journal (PBJ) Daily Nurse Staffing and can be found [here](https://data.cms.gov/quality-of-care/payroll-based-journal-daily-nurse-staffing/data). You can review the data dictionary to better understand the data that’s available, but a few notes pertaining to this exercise:

* Nursing homes are staffed by both employees and contractors. Employees work for the nursing home full time, whereas contractors work for the nursing home on a temporary basis. Most nursing homes are staffed using some combination of full time employees and contractors, but the proportions within each nursing home can vary substantially. All workers from Clipboard Health (and our competitors) are classified as contractors.
* Clipboard Health is a nationwide staffing platform, operating in all 50 states.
* There are a variety of other datasets that can be joined to this dataset, all found under the [nursing home data](https://data.cms.gov/provider-data/search?theme=Nursing%20homes%20including%20rehab%20services) section on CMS’ website.
* As mentioned previously, the PBJ data is separated by quarter. For this exercise, please focus on the most recent quarter available (2024Q1).

Can you please use the PBJ data and any other CMS data that you see fit to make a few recommendations to the Clipboard Health sales leadership team? There are no right or wrong answers but a few tips:

* Be sure to include the supporting data (charts, tables, etc.) for any recommendation you make. The purpose of this exercise is largely to see how well you can quickly learn and analyze new datasets, as well as communicate the learnings and recommendations from those analyses.
* Feel free to use any analytical tools that you have at your disposal.
* We do a lot of writing at Clipboard Health (we believe that writing is thinking), so each recommendation should include some writing around how you analyzed the data, what you uncovered, and the specifics of what you’d recommend the sales team do based on your findings.
* There is no minimum number of recommendations we’d like you to make, but please include no more than 5.

**Part 2: SQL Test**

**Instructions:** Please write SQL queries for each of the following questions. You may assume that all tables follow typical database conventions unless otherwise specified.

### **Tables:**

Assume you have the following tables in your database:

1. **Sales**
   * sales\_id (INT)
   * customer\_id (INT)
   * product\_id (INT)
   * sale\_date (DATE)
   * quantity (INT)
   * total\_amount (DECIMAL)
2. **Customers**
   * customer\_id (INT)
   * customer\_name (VARCHAR)
   * sales\_region (VARCHAR)
   * sign\_up\_date (DATE)
3. **Products**
   * product\_id (INT)
   * product\_name (VARCHAR)
   * category (VARCHAR)
   * price (DECIMAL)

### **Questions:**

1. Write a query to return the customer\_name, product\_name, and total\_amount for each sale in the last 30 days.

SELECT customer\_name, product\_name, total\_amount

FROM Sales

JOIN Customers

ON Sales.customer\_id = Customers.customer\_id

JOIN Products

ON Sales.product\_id = Products.product\_id

WHERE Sales.sale\_date >= CURDATE() - INTERVAL 30 DAY;

1. Write a query to find the total revenue generated by each product category in the last year. The output should include the product category and the total revenue for that category.

SELECT category, SUM(total\_amount) AS total\_revenue

FROM Sales

JOIN Products

ON Sales.product\_id = Products.product\_id

WHERE sale\_date >= CURDATE() - INTERVAL 1 YEAR

GROUP BY Category;

1. Write a query to return all customers who made purchases in 2023 and are located in the "West" region.

SELECT distinct customer\_name

FROM Sales

JOIN Customers

ON Sales.customer\_id = Customers.customer\_id

WHERE sales\_region = “West” AND sale\_date between ‘01/01/2023’ AND ‘12/31/2023’;

1. Write a query to display the total number of sales, total quantity sold, and total revenue for each customer. The result should include the customer\_name, total sales, total quantity, and total revenue.

SELECT SUM(quantity), COUNT(sales\_id), SUM(total\_amount) AS total\_revenue, distinct customer\_name

FROM Sales

JOIN Customers

ON Sales.customer\_id = Customers.customer\_id

GROUP BY customer\_name;

1. Write a query to find the top 3 customers (by total revenue) in the year 2023.

SELECT SUM(total\_amount) AS total\_revenue, distinct customer\_name

FROM Sales

JOIN Customers

ON Sales.customer\_id = Customers.customer\_id

WHERE sale\_date between ‘01/01/2023’ AND ‘12/31/2023’

GROUP BY customer\_name

ORDER BY total\_revenue DESC LIMIT 3;

1. Write a query to rank products by their total sales quantity in 2023. The result should include the product\_name, total quantity sold, and rank.

SELECT product\_name, SUM(quantity) AS total\_quantity\_sold, RANK() OVER( ORDER BY(SUM(quantity) DESC) AS rank

FROM Sales

JOIN Products

ON Sales.product\_id = Products.product\_id

WHERE sale\_date between ‘01/01/2023’ AND ‘12/31/2023’

GROUP BY product\_name

ORDER BY total\_quantity\_sold DESC;

1. Write a query that categorizes customers into "New" (if they signed up in the last 6 months) or "Existing" based on their sign\_up\_date. Include the customer\_name, region, and category in the result.

SELECT distinct customer\_name, sales\_region,

CASE WHEN sign\_up\_date >= CURDATE() - INTERVAL 6 MONTHS THEN ‘New’

ELSE ‘Existing’ END AS customer\_category

FROM Customers

GROUP BY customer\_type;

1. Write a query to return the month and year along with the total sales for each month for the last 12 months.

SELECT DATE\_FORMAT(sale\_date, ‘%M- %Y’) AS month\_year, SUM(total\_amount) AS total\_sales

FROM Sales

WHERE sales\_date >= CURDATE() INTERVAL 12 MONTH

GROUP BY MONTH(sale\_date), YEAR(sale\_date)

ORDER BY month\_year;

1. Write a query to return the product categories that generated more than $50,000 in revenue during the last 6 months.

SELECT categories, SUM(total\_amount) AS total\_revenue

FROM Sales

JOIN Products

ON Products.product\_id = Sales.product\_id

WHERE sale\_date >= CURDATE() INTERVAL 6 MONTH

GROUP BY categories

HAVING total\_revenue > 50000;

1. Write a query to check for any sales where the total\_amount doesn’t match the expected value (i.e., quantity \* price).

SELECT total\_amount, price, sales\_id, quantity, product\_name, customer\_name, (quantity \* price) AS expected\_amount

FROM Sales

JOIN Products

ON Sales.product\_id = Products.product\_id

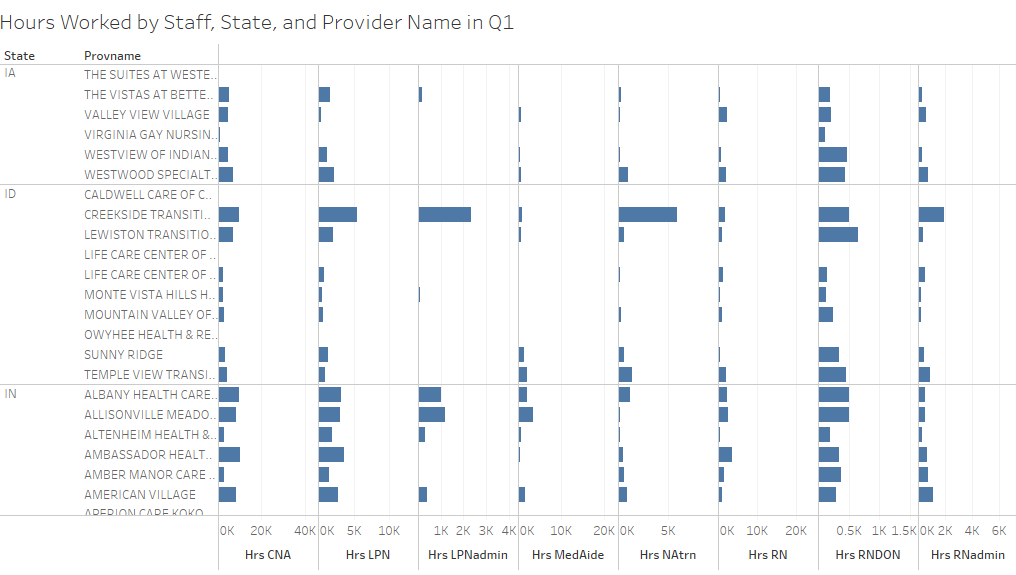
JOIN Customers

ON Sales.customer\_id = Customers.customer\_id

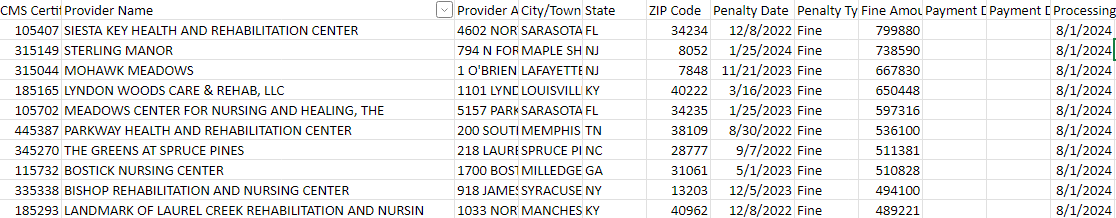
WHERE total\_amount != (quantity\*price)

Part 1 Case Study Analysis:

1. Because of initial file/table size in database making the export file large, I reviewed a few columns on each page and saw there were several empty “hours” cells.
2. I then used the filters in the table to remove any table results where any type of worker must have worked at least 1 hour to be displayed, which dropped the total lines of data from **1,330,966** to **40,436.** 
   1. Improvement suggested: Removing any employee that didn’t work in a quarter will make it easier to see where to redirect resources, who is contributing to revenue, and who is likely actively taking on work. This could be done with an automatic filter set to hide any employee with 0 hours logged in a quarter.
   2. This filter serves two processes, as you could also use the filter in reverse, by setting it to “equal to” 0 for performance reviews, work evaluations, expense reports (if any incurred for employees with 0 hours), and to pinpoint who is not active.
   3. Link to show who was left in excel spreadsheet after filter applied: [PBJ\_Daily\_Nurse\_Staffing\_Q1\_2024.csv](https://1drv.ms/x/c/d9e0d8ba305bf103/ESyUB37GMIVEn0-bXbAdUL8BWqC0Xyc7LvQdN5N7XUrhKA?e=zLAlUm&nav=MTVfezA2REVFREQxLTFGRjAtNEE1NS04NjEzLTlBQ0I3NTdENTEyM30)
   4. Basic graph for easy read example in Tableau:

****

1. I then reviewed the data from the “NH Penalties” dataset, as that would be another type of expense that may be avoidable if analyzed for areas of improvement for staff that were actively working in August.
2. I created filters on both spreadsheets to match on Provider Name to make penalties easy to track on a provider basis. I also removed the “Locations” column, as it is duplicated to “Provider Address” on “NH Penalties”, creating a lot of unnecessary data overlap.
3. I then wanted to see which providers were experiencing the highest penalty amounts over a consistent basis to understand if there is a recurring pattern with staff training or processes. As an example, I took the 10 highest penalties - seen below.

****

1. Improvement Suggested: Because penalties have occurred long-term for several providers in the top 10 list, it may be helpful to cross reference who was on payroll and working hours with these providers in the 1st Qtr of 2024 to evaluate if some of these are penalties occurring in that 1st Qtr and correct actions leading to penalties.

Summary: I feel using these two datasets would greatly contribute to eliminate unnecessary expenses on an easy to implement basis, improving overall profit margins and reputation growth.