In this set of exercises you are going to explore additional ways to group and organize the output of a query when using postgres.

For the first few exercises, we are going to compare the total number of claims from Interventional Pain Management Specialists compared to those from Pain Managment specialists.

1. Write a query which returns the total number of claims for these two groups. Your output should look like this:

specialty\_description |total\_claims|

------------------------------|------------|

Interventional Pain Management| 55906|

Pain Management | 70853|

2. Now, let's say that we want our output to also include the total number of claims between these two groups. Combine two queries with the UNION keyword to accomplish this. Your output should look like this:

From Chris Swann: create a new row

specialty\_description |total\_claims|

------------------------------|------------|

| 126759|

Interventional Pain Management| 55906|

Pain Management | 70853|

3. Now, instead of using UNION, make use of GROUPING SETS (https://www.postgresql.org/docs/10/queries-table-expressions.html#QUERIES-GROUPING-SETS) to achieve the same output.

4. In addition to comparing the total number of prescriptions by specialty, let's also bring in information about the number of opioid vs. non-opioid claims by these two specialties. Modify your query (still making use of GROUPING SETS so that your output also shows the total number of opioid claims vs. non-opioid claims by these two specialites:

specialty\_description |opioid\_drug\_flag|total\_claims|

------------------------------|----------------|------------|

| | 129726|

|Y | 76143|

|N | 53583|

Pain Management | | 72487|

Interventional Pain Management| | 57239|

5. Modify your query by replacing the GROUPING SETS with ROLLUP(opioid\_drug\_flag, specialty\_description). How is the result different from the output from the previous query?

6. Switch the order of the variables inside the ROLLUP. That is, use ROLLUP(specialty\_description, opioid\_drug\_flag). How does this change the result?

7. Finally, change your query to use the CUBE function instead of ROLLUP. How does this impact the output?

8. In this question, your goal is to create a pivot table showing for each of the 4 largest cities in Tennessee (Nashville, Memphis, Knoxville, and Chattanooga), the total claim count for each of six common types of opioids: Hydrocodone, Oxycodone, Oxymorphone, Morphine, Codeine, and Fentanyl. For the purpose of this question, we will put a drug into one of the six listed categories if it has the category name as part of its generic name. For example, we could count both of "ACETAMINOPHEN WITH CODEINE" and "CODEINE SULFATE" as being "CODEINE" for the purposes of this question.

The end result of this question should be a table formatted like this:

city |codeine|fentanyl|hyrdocodone|morphine|oxycodone|oxymorphone|

-----------|-------|--------|-----------|--------|---------|-----------|

CHATTANOOGA| 1323| 3689| 68315| 12126| 49519| 1317|

KNOXVILLE | 2744| 4811| 78529| 20946| 84730| 9186|

MEMPHIS | 4697| 3666| 68036| 4898| 38295| 189|

NASHVILLE | 2043| 6119| 88669| 13572| 62859| 1261|

For this question, you should look into use the crosstab function, which is part of the tablefunc extension (https://www.postgresql.org/docs/9.5/tablefunc.html). In order to use this function, you must (one time per database) run the command

CREATE EXTENSION tablefunc;

Hint #1: First write a query which will label each drug in the drug table using the six categories listed above.

Hint #2: In order to use the crosstab function, you need to first write a query which will produce a table with one row\_name column, one category column, and one value column. So in this case, you need to have a city column, a drug label column, and a total claim count column.

Hint #3: The sql statement that goes inside of crosstab must be surrounded by single quotes. If the query that you are using also uses single quotes, you'll need to escape them by turning them into double-single quotes.