-- 1. a. Which prescriber had the highest total number of claims (totaled over all drugs)? Report the npi and the total number of claims.

SELECT

npi

,SUM(total\_claim\_count) AS total\_claims

FROM prescription

GROUP BY npi

ORDER BY total\_claims DESC;

-- b. Repeat the above, but this time report the nppes\_provider\_first\_name, nppes\_provider\_last\_org\_name, specialty\_description, and the total number of claims.

SELECT

nppes\_provider\_first\_name

,nppes\_provider\_last\_org\_name

,specialty\_description

,SUM(total\_claim\_count) AS total\_claims

FROM prescription

INNER JOIN prescriber

USING(npi)

GROUP BY nppes\_provider\_first\_name

,nppes\_provider\_last\_org\_name

,specialty\_description

ORDER BY total\_claims DESC;

-- 2. a. Which specialty had the most total number of claims (totaled over all drugs)?

SELECT

specialty\_description

,SUM(total\_claim\_count) AS total\_claims

FROM prescriber

INNER JOIN prescription

USING (npi)

GROUP BY specialty\_description

ORDER BY total\_claims DESC;

-- b. Which specialty had the most total number of claims for opioids?

SELECT

specialty\_description

,SUM(total\_claim\_count) AS total\_claims

FROM prescriber

INNER JOIN prescription

USING (npi)

INNER JOIN drug

USING(drug\_name)

WHERE opioid\_drug\_flag = 'Y'

GROUP BY specialty\_description

ORDER BY total\_claims DESC;

-- SELECT \* FROM drug

-- c. Challenge Question: Are there any specialties that appear in the prescriber table that have no associated prescriptions in the prescription table?

SELECT

specialty\_description

,SUM(total\_claim\_count) AS total\_claims

FROM prescriber

LEFT JOIN prescription

USING (npi)

GROUP BY specialty\_description

HAVING SUM(total\_claim\_count) IS NULL;

-- USING EXCEPT

SELECT

specialty\_description

FROM

prescriber

EXCEPT

SELECT

p.specialty\_description

FROM

prescriber AS p

JOIN prescription AS rx USING (npi);

-- d. Difficult Bonus: Do not attempt until you have solved all other problems! For each specialty, report the percentage of total claims by that specialty which are for opioids. Which specialties have a high percentage of opioids?

SELECT

specialty\_description

,ROUND((SUM(CASE WHEN opioid\_drug\_flag = 'Y' THEN total\_claim\_count END)/

SUM(total\_claim\_count)), 2) \* 100 AS percent\_opioid

FROM prescriber

LEFT JOIN prescription

USING(npi)

LEFT JOIN drug

USING (drug\_name)

GROUP BY specialty\_description

ORDER BY percent\_opioid DESC NULLS LAST;

-- USING CTE

WITH claims AS (

SELECT

specialty\_description,

COALESCE(SUM(CASE WHEN opioid\_drug\_flag = 'Y' THEN total\_claim\_count END), 0) AS opioid\_claims,

COALESCE(SUM(total\_claim\_count), 0) AS total\_claims

FROM prescriber

LEFT JOIN prescription

USING(npi)

LEFT JOIN drug

USING(drug\_name)

GROUP BY specialty\_description

)

SELECT

specialty\_description,

CASE

WHEN total\_claims = 0 THEN 0

ELSE ROUND(((opioid\_claims / total\_claims)\*100), 2)

END AS opioid\_percentage

FROM claims

ORDER BY opioid\_percentage DESC;

-- 3.a. Which drug (generic\_name) had the highest total drug cost?

SELECT

generic\_name

, SUM(total\_drug\_cost)::MONEY AS total\_cost

FROM drug

INNER JOIN prescription

USING(drug\_name)

GROUP BY generic\_name

ORDER BY total\_cost DESC;

-- b. Which drug (generic\_name) has the hightest total cost per day? Bonus: Round your cost per day column to 2 decimal places. Google ROUND to see how this works.

SELECT

generic\_name

, (SUM(total\_drug\_cost)/SUM(total\_day\_supply))::MONEY AS total\_cost

FROM drug

INNER JOIN prescription

USING(drug\_name)

GROUP BY generic\_name

ORDER BY total\_cost DESC;

-- 4. a. For each drug in the drug table, return the drug name and then a column named 'drug\_type' which says 'opioid' for drugs which have opioid\_drug\_flag = 'Y', says 'antibiotic' for those drugs which have antibiotic\_drug\_flag = 'Y', and says 'neither' for all other drugs. Hint: You may want to use a CASE expression for this. See https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-case/

SELECT

drug\_name

, CASE

WHEN opioid\_drug\_flag = 'Y' THEN 'opioid'

WHEN antibiotic\_drug\_flag = 'Y' THEN 'antibiotic'

ELSE 'neither'

END drug\_type

FROM drug;

-- b. Building off of the query you wrote for part a, determine whether more was spent (total\_drug\_cost) on opioids or on antibiotics. Hint: Format the total costs as MONEY for easier comparision.

SELECT

CASE

WHEN opioid\_drug\_flag = 'Y' THEN 'opioid'

WHEN antibiotic\_drug\_flag = 'Y' THEN 'antibiotic'

ELSE 'neither'

END drug\_type

, SUM(total\_drug\_cost)::MONEY AS total\_cost

FROM drug

INNER JOIN prescription

USING(drug\_name)

GROUP BY drug\_type

ORDER BY total\_cost DESC;

WITH new\_drug\_table AS (

SELECT drug\_name,

CASE

WHEN opioid\_drug\_flag = 'Y' THEN 'opioid'

WHEN antibiotic\_drug\_flag = 'Y' THEN 'antibiotic'

ELSE 'neither'

END AS drug\_type,

total\_drug\_cost::money

FROM drug

LEFT JOIN prescription

USING(drug\_name)

)

SELECT

SUM(CASE WHEN drug\_type = 'opioid' THEN total\_drug\_cost END) AS opioid\_total,

SUM(CASE WHEN drug\_type = 'antibiotic' THEN total\_drug\_cost END) AS antibiotic\_total

FROM new\_drug\_table;

-- 5. a. How many CBSAs are in Tennessee? Warning: The cbsa table contains information for all states, not just Tennessee.

SELECT

Count(DISTINCT cbsa)

FROM cbsa

INNER JOIN fips\_county

USING(fipscounty)

WHERE state = 'TN';

-- b. Which cbsa has the largest combined population? Which has the smallest? Report the CBSA name and total population.

SELECT

cbsaname

,SUM(population) AS total\_pop

FROM cbsa

INNER JOIN POPULATION

USING(fipscounty)

GROUP BY cbsaname

ORDER BY total\_pop DESC;

-- c. What is the largest (in terms of population) county which is not included in a CBSA? Report the county name and population.

SELECT

county

,population

FROM fips\_county

INNER JOIN population

USING(fipscounty)

WHERE fipscounty NOT IN (SELECT fipscounty FROM cbsa)

ORDER BY population DESC;

-- USING EXCEPT

SELECT county, MAX(population) AS max\_pop

FROM cbsa

RIGHT JOIN fips\_county

USING (fipscounty)

INNER JOIN population

USING (fipscounty)

WHERE cbsa IS NULL

GROUP BY county

ORDER BY max\_pop DESC;

-- 6. a. Find all rows in the prescription table where total\_claims is at least 3000. Report the drug\_name and the total\_claim\_count.

SELECT

drug\_name

,total\_claim\_count

FROM prescription

WHERE total\_claim\_count >= 3000;

-- b. For each instance that you found in part a, add a column that indicates whether the drug is an opioid.

SELECT

drug\_name

,total\_claim\_count

, opioid\_drug\_flag

FROM prescription

INNER JOIN drug

USING(drug\_name)

WHERE total\_claim\_count >= 3000;

-- c. Add another column to you answer from the previous part which gives the prescriber first and last name associated with each row.

SELECT

nppes\_provider\_first\_name

,nppes\_provider\_last\_org\_name

,drug\_name

,total\_claim\_count

, opioid\_drug\_flag

FROM prescription

INNER JOIN drug

USING(drug\_name)

INNER JOIN prescriber

USING(npi)

WHERE total\_claim\_count >= 3000;

-- USING CASE

SELECT

CONCAT(nppes\_provider\_last\_org\_name, ', ', nppes\_provider\_first\_name) AS prescriber\_name,

drug\_name,

total\_claim\_count,

CASE

WHEN opioid\_drug\_flag = 'Y' THEN 'opioid'

WHEN antibiotic\_drug\_flag = 'Y' THEN 'antibiotic'

WHEN opioid\_drug\_flag = 'N' OR antibiotic\_drug\_flag = 'N' THEN 'neither'

END AS drug\_type

FROM prescription

LEFT JOIN prescriber

USING (npi)

INNER JOIN drug

USING (drug\_name)

WHERE total\_claim\_count >= 3000

ORDER BY total\_claim\_count DESC;

SELECT pr.nppes\_provider\_first\_name, pr.nppes\_provider\_last\_org\_name, drug.drug\_name, total\_claim\_count,

CASE

WHEN opioid\_drug\_flag = 'Y' THEN 'opioid'

ELSE 'not\_an\_opioid'

END drug\_opioid

FROM prescription ps

INNER JOIN drug ON drug.drug\_name = ps.drug\_name

INNER JOIN prescriber pr ON pr.npi = ps.npi

WHERE total\_claim\_count >= 3000

GROUP BY drug.drug\_name, total\_claim\_count,opioid\_drug\_flag, pr.nppes\_provider\_first\_name, pr.nppes\_provider\_last\_org\_name

ORDER BY total\_claim\_count DESC;

-- The goal of this exercise is to generate a full list of all pain management specialists in Nashville and the number of claims they had for each opioid. Hint: The results from all 3 parts will have 637 rows.

-- 7. a. First, create a list of all npi/drug\_name combinations for pain management specialists (specialty\_description = 'Pain Management) in the city of Nashville (nppes\_provider\_city = 'NASHVILLE'), where the drug is an opioid (opioid\_drug\_flag = 'Y'). Warning: Double-check your query before running it. You will only need to use the prescriber and drug tables since you don't need the claims numbers yet.

SELECT

\*

FROM prescriber

CROSS JOIN drug

WHERE specialty\_description = 'Pain Management'

AND nppes\_provider\_city = 'NASHVILLE'

AND opioid\_drug\_flag = 'Y';

-- b. Next, report the number of claims per drug per prescriber. Be sure to include all combinations, whether or not the prescriber had any claims. You should report the npi, the drug name, and the number of claims (total\_claim\_count).

SELECT

npi

,drug.drug\_name

,total\_claim\_count

FROM prescriber

CROSS JOIN drug

LEFT JOIN prescription

USING(npi, drug\_name)

WHERE specialty\_description = 'Pain Management'

AND nppes\_provider\_city = 'NASHVILLE'

AND opioid\_drug\_flag = 'Y';

-- c. Finally, if you have not done so already, fill in any missing values for total\_claim\_count with 0. Hint - Google the COALESCE function.

SELECT

npi

,drug.drug\_name

,COALESCE(total\_claim\_count, 0) AS total\_claims

FROM prescriber

CROSS JOIN drug

LEFT JOIN prescription

USING(npi, drug\_name)

WHERE specialty\_description = 'Pain Management'

AND nppes\_provider\_city = 'NASHVILLE'

AND opioid\_drug\_flag = 'Y'

ORDER BY total\_claims DESC;

-- USING CTE

WITH pain\_specialists AS (

SELECT

npi,

drug\_name

FROM

drug

CROSS JOIN

prescriber

WHERE

specialty\_description = 'Pain Management'

AND nppes\_provider\_city = 'NASHVILLE'

AND opioid\_drug\_flag = 'Y'

)

SELECT

npi,

drug\_name,

COALESCE(SUM(total\_claim\_count), 0) AS claim\_count

FROM

pain\_specialists AS p1

LEFT JOIN

prescription AS p2

USING(npi, drug\_name)

GROUP BY npi, drug\_name

ORDER BY claim\_count DESC;