Title:

Flu Shot Data Analysis Project using PostgreSQL.

Requirements:

You will have a few flat files, load them into PostgreSQL and answer the following important healthcare questions.

Problem Statements:

1. **Annual Flu Shot Statistics by Demographics:**
    - How many patients received a flu shot in the year 2022, broken down by age, race, and county?
    - Answer query:

Query | Query History

```sql
SELECT
    p.race,
    p.county,
    CASE
        WHEN EXTRACT(YEAR FROM AGE('2024-01-01', p.BIRTHDATE)) < 18 THEN '0-17'
        WHEN EXTRACT(YEAR FROM AGE('2024-01-01', p.BIRTHDATE)) BETWEEN 18 AND 35 THEN '18-35'
        WHEN EXTRACT(YEAR FROM AGE('2024-01-01', p.BIRTHDATE)) BETWEEN 36 AND 59 THEN '36-59'
        ELSE '60+'
    END AS AgeGroup,
    COUNT(*) AS GroupCount
FROM patients as p
JOIN immunizations as i ON p.id = i.patient
WHERE EXTRACT(YEAR FROM i.date) = 2022
    AND i.description LIKE '%Flu%'
GROUP BY p.race, p.county, AgeGroup, i.description
```

Data Output | Messages | Notifications

	race character varying (100)	county character varying (100)	agegroup text	groupcount bigint
1	asian	Barnstable County	0-17	1
2	asian	Barnstable County	60+	1
3	asian	Berkshire County	18-35	1
4	asian	Berkshire County	36-59	4
5	asian	Berkshire County	60+	1
6	asian	Bristol County	0-17	5
7	asian	Bristol County	18-35	3
8	asian	Bristol County	36-59	7
9	asian	Bristol County	60+	4
10	asian	Dukes County	36-59	2
11	asian	Essex County	0-17	10
12	asian	Essex County	18-35	12
13	asian	Essex County	36-59	11
14	asian	Essex County	60+	11
15	asian	Hampden County	18-35	2
16	asian	Hampden County	36-59	5

Total rows: 233 of 233 | Query complete 00:00:00.132

2. **Percentage Analysis of Flu Shots:**
    - What is the percentage of patients who received a flu shot in 2022 compared to the total number of patients, stratified by age, race, and county?
    - Answer query:

```sql
------- Below is the answer of question 2 (a)
SELECT
    p.race,
    p.county,
    EXTRACT(YEAR FROM AGE('2024-01-01', p.birthdate)) AS Age
FROM
    patients as p
JOIN
    immunizations as i
ON
    p.Id = i.patient
WHERE
    i.description LIKE '%Flu%'
    AND EXTRACT(YEAR FROM i.date) = 2022
GROUP BY
    p.race,
    p.county,
    Age
```

Data Output | Messages | Notifications

	race character varying (100)	county character varying (100)	age numeric
1	asian	Barnstable County	4
2	asian	Barnstable County	74
3	asian	Berkshire County	20
4	asian	Berkshire County	37
5	asian	Berkshire County	39
6	asian	Berkshire County	59
7	asian	Berkshire County	62
8	asian	Bristol County	3
9	asian	Bristol County	9
10	asian	Bristol County	13
11	asian	Bristol County	14
12	asian	Bristol County	17
13	asian	Bristol County	19
14	asian	Bristol County	23

Total rows: 1000 of 1983 | Query complete 00:00:00.134

- What is the overall percentage of patients who received a flu shot in 2022 across the entire hospital or clinic?
- Answer query:

```sql
------- Below is the answer of question 2 (b)
SELECT
    (SELECT COUNT(i.patient) --specific
    FROM immunizations as i
    WHERE i.description LIKE '%Flu%'
        AND EXTRACT(YEAR FROM i.date) = 2022) * 100 /
    (SELECT COUNT(i.patient) --total
    FROM immunizations as i) AS percentage_of_vaccinated
```

Data Output | Messages | Notifications

	percentage_of_vaccinated bigint
1	5

3. **Monthly Cumulative Flu Shot Data:**

    - How many flu shots were administered each month during the year 2022?
    - Answer query:

```
Query  Query History
1  SELECT COUNT(patient)
2  FROM immunizations as i
3  WHERE i.description LIKE '%Flu%'
4      AND EXTRACT(YEAR FROM i.date) = 2022
5  -- 8437 flu shots were adminstered in the year 2020.
6
7  -- Now lets calculate flu shots per month
8  SELECT
9      SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 1 THEN 1 ELSE 0 END) AS Jan,
10     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 2 THEN 1 ELSE 0 END) AS Feb,
11     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 3 THEN 1 ELSE 0 END) AS Mar,
12     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 4 THEN 1 ELSE 0 END) AS Apr,
13     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 5 THEN 1 ELSE 0 END) AS May,
14     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 6 THEN 1 ELSE 0 END) AS Jun,
15     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 7 THEN 1 ELSE 0 END) AS Jul,
16     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 8 THEN 1 ELSE 0 END) AS Aug,
17     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 9 THEN 1 ELSE 0 END) AS Sep,
18     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 10 THEN 1 ELSE 0 END) AS Oct,
19     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 11 THEN 1 ELSE 0 END) AS Nov,
20     SUM(CASE WHEN EXTRACT(MONTH FROM i.date) = 12 THEN 1 ELSE 0 END) AS Dec
21 FROM
22     immunizations as i
23 WHERE
24     i.description LIKE '%Flu%'
25     AND EXTRACT(YEAR FROM i.date) = 2022
```

Data Output Messages Graph Visualiser Notifications

	jan bigint	feb bigint	mar bigint	apr bigint	may bigint	jun bigint	jul bigint	aug bigint	sep bigint	oct bigint	nov bigint	dec bigint
1	656	616	757	892	723	661	712	711	669	704	667	669

4. **Total Annual Flu Shots:**

    - What is the total number of flu shots given in the year 2022?
    - Answer query:

```
Query  Query History
1  SELECT COUNT(patient)
2  FROM immunizations as i
3  WHERE i.description LIKE '%Flu%'
4      AND EXTRACT(YEAR FROM i.date) = 2022
5  -- 8437 flu shots were adminstered in the year 2020.
```

Data Output Messages Graph Visualiser Notifications

	count bigint
1	8437

5. **Patient Lists for Flu Shot Analysis:**

    - Can you provide a list of patients who received a flu shot in 2022?
    - Answer query:

```sql
SELECT  i.patient, p.first, p.last, p.gender, p.county, i.description
FROM immunizations as i
JOIN patients as p
ON i.patient = p.Id
WHERE i.description LIKE '%Flu%'
    AND EXTRACT(YEAR FROM i.date) = 2022
-- list of patients who received a flu shot in 2022
```

Data Output | Messages | Notifications

	patient character varying (100)	first character varying (100)	last character varying (100)	gender character varying (100)	county character varying (100)	description character varying (500)
1	73d3ebe3-e656-b9de-fd61-88d370a86d...	Hedy	Von	F	Middlesex County	Seasonal Flu Vaccine
2	32a2188a-132e-4fd4-1a0e-3d532f4c4ea8	Lauryn	Wisozk	F	Suffolk County	Seasonal Flu Vaccine
3	4adcad4e-1aa5-5601-4107-55953f4847...	Dorthea	Reichel	F	Norfolk County	Seasonal Flu Vaccine
4	52de8610-de40-203d-4bd6-cf06141fa864	Sherita	Orn	F	Hampden County	Seasonal Flu Vaccine
5	ac63da4a-893a-1d3c-aa93-fb78c8da3d95	Casie	Hilpert	F	Norfolk County	Seasonal Flu Vaccine
6	10d940ae-7114-8bcf-50f9-2bfd53354ff41	Rebeca	Mayer	F	Essex County	Seasonal Flu Vaccine
7	154290c8-6729-fe33-d3b6-a66f04bb939e	Elenora	Raynor	F	Barnstable County	Seasonal Flu Vaccine
8	91f57fef-7f76-6f23-1c4b-eb973e556ce3	Rosario	Ruecker	F	Norfolk County	Seasonal Flu Vaccine

- Can you also provide a list of patients who did not receive a flu shot in 2022?
- Answer query:

Query | Query History

```sql
SELECT  i.patient, p.first, p.last, p.gender, p.county, i.description
FROM immunizations as i
JOIN patients as p
ON i.patient = p.Id
WHERE i.description NOT LIKE '%Flu%'
    AND EXTRACT(YEAR FROM i.date) = 2022
-- list of patients who did not receive a flu shot in 2022
```

Data Output | Messages | Notifications

	patient character varying (100)	first character varying (100)	last character varying (100)	gender character varying (100)	county character varying (100)	description character varying (500)
1	ac63da4a-893a-1d3c-aa93-fb78c8da3d95	Casie	Hilpert	F	Norfolk County	Herpes Zoster Vaccine (Live)
2	ac63da4a-893a-1d3c-aa93-fb78c8da3d95	Casie	Hilpert	F	Norfolk County	Five doses of tetanus toxoid, preservative-free and adsorbed, for adul...
3	6cc89ec7-8e4a-536e-356a-05d89f3fac7a	Dakota	Kihn	F	Worcester County	Novel Coronavirus (COVID-19) mRNA Vaccine 30 mcg/0.3mL Dose
4	6cc89ec7-8e4a-536e-356a-05d89f3fac7a	Dakota	Kihn	F	Worcester County	Human Papillomavirus (HPV) Four-strain Vaccine
5	6cc89ec7-8e4a-536e-356a-05d89f3fac7a	Dakota	Kihn	F	Worcester County	Novel Coronavirus (COVID-19) mRNA Vaccine 30 mcg/0.3mL Dose
6	3e6e47b6-2002-21ee-a8d1-6fa83d15b3e1	Lissa	Luettgen	F	Plymouth County	Novel Coronavirus (COVID-19) mRNA Vaccine 30 mcg/0.3mL Dose
7	3124cf0c-fa3a-480e-bd23-d7c89c099d51	Lorilee	Kemmer	F	Barnstable County	Five doses of tetanus toxoid, preservative-free and adsorbed, for adul...
8	f216356a-fee2-9b37-2b40-f130541e8636	Colene	Bailey	F	Essex County	Five doses of tetanus toxoid, preservative-free and adsorbed, for adul...

More Questions that should be answered:

1. **Patient Demographics Analysis:** How many patients are in each age group (e.g., 0-18, 19-35, 36-60, 60+)?

Query | Query History

```sql
-- 1. Patient Demographics Analysis:
-- How many patients are in each age group (e.g., 0-18, 19-35, 36-60, 60+)?
SELECT
CASE
    WHEN EXTRACT(YEAR FROM AGE('2024-01-01', p.birthdate)) <= 18 THEN '0-18'
    WHEN EXTRACT(YEAR FROM AGE('2024-01-01', p.birthdate)) BETWEEN 19 and 35 THEN '19-35'
    WHEN EXTRACT(YEAR FROM AGE('2024-01-01', p.birthdate)) BETWEEN 36 and 60 THEN '36-60'
    ELSE '60+'
END AS AgeGroup,
COUNT(*) AS GroupCount
FROM patients as p
GROUP BY AgeGroup
```

Data Output | Messages | Notifications

	agegroup text	groupcount bigint
1	19-35	2413
2	60+	3183
3	36-60	3661
4	0-18	2106

2. **Geographical Distribution:** What are the top 5 cities with the highest number of patients?

```sql
-- 2. Geographical Distribution:
-- What are the top 5 cities with the highest number of patients?
SELECT p.city, COUNT(p.id)
FROM patients as p
GROUP BY p.city
ORDER BY COUNT(p.id) DESC
LIMIT 5
```

Data Output | Messages | Notifications

	city character varying (100)	count bigint
1	Boston	1016
2	Worcester	295
3	Springfield	244
4	Lowell	193
5	Cambridge	166

3. **Healthcare Utilization Patterns:** How many encounters does each patient have on average?
    1. **There are 2 meanings of the question:** I've solved both.
        1. This means that how many encounters does each patient have:

```sql
-- 3. Healthcare Utilization Patterns:
-- How many encounters does each patient have on average?
SELECT patient, COUNT(*) AS encounter_count
FROM conditions
GROUP BY patient;
```

Data Output | Messages | Notifications

	patient character varying (1000)	encounter_count bigint
1	519816e7-29d9-dc7e-00ba-cf949569cb...	5
2	f118199c-a128-e36a-dc75-a5699ccf94f7	14
3	1b5d0889-ce72-c592-f47f-fe2596d693...	18
4	009bdb22-c99b-2a3d-a907-3d30d6a7a...	1
5	3e123fca-8168-4a02-b309-e3cf8de0b6...	18

        2. This shows how many encounters does each patient have on average, so the answer will be just one for all patients and not different for each patient:

```sql
SELECT
    AVG(encounter_count) AS AverageEncounters
FROM (
    SELECT
        patient,
        COUNT(*) AS encounter_count
    FROM
        encounters
    GROUP BY
        patient
);
```

Data Output | Messages | Notifications

	averageencounters numeric
1	43.5967680244788679

4. **Condition Prevalence:** What are the top 10 most common conditions among patients?

```sql
-- 4. Condition Prevalence:
-- What are the top 10 most common conditions among patients?
SELECT description, COUNT(*) AS MostCommon
FROM conditions
GROUP BY description
ORDER BY MostCommon desc
LIMIT 10
```

Data Output | Messages | Notifications

	description character varying (200)	mostcommon bigint
1	Other psychological or physical stress, not elsewhere classifi...	58962
2	Pregnant state, incidental	9872
3	Acute bronchitis	5684
4	Body Mass Index 30.0-30.9, adult	5461
5	Unemployment	4717
6	Anemia, unspecified	3704
7	Other specified anemias	3704
8	Unspecified essential hypertension	3682
9	Other chronic pain	2727
10	Tubal ligation status	2537

5. **Healthcare Costs Analysis:** What is the average healthcare expense per patient?

```sql
-- 5. Healthcare Costs Analysis:
-- What is the average healthcare expense per patient?
SELECT avg(healthcare_expenses)
FROM patients
```

Data Output | Messages | Graph Visualiser | Notifications

	avg double precision
1	272690.2093751649

6. **Encounter Analysis:** Which type of encounters (as per **ENCOUNTERCLASS**) is most common?

```sql
-- 6. Encounter Analysis:
-- Which type of encounters (as per ENCOUNTERCLASS) is most common?
SELECT encounterclass, COUNT(encounterclass) AS Count_of_each
FROM encounters
GROUP BY encounterclass
ORDER BY Count_of_each desc
```

Data Output | Messages | Notifications

	encounterclass character varying (100)	count_of_each bigint
1	ambulatory	244148
2	outpatient	85849
3	wellness	77849
4	urgentcare	21479
5	emergency	17903
6	inpatient	3709
7	home	2318
8	snf	1221
9	hospice	750
10	virtual	709

7. **Income vs. Healthcare Coverage:** Is there a correlation between patients' income levels and their healthcare coverage amounts?

```sql
-- 7. Income vs. Healthcare Coverage:
-- Is there a correlation between patients' income levels and their healthcare coverage amounts?
SELECT
    IncomeBracket,
    AVG(healthcare_coverage) AS AverageCoverage
FROM
    (SELECT
        healthcare_coverage,
        CASE
            WHEN income <= 1000 THEN '0-1K'
            WHEN income > 1000 AND income <= 10000 THEN '1K-10K'
            WHEN income > 10000 AND income <= 50000 THEN '10K-50K'
            WHEN income > 50000 AND income <= 100000 THEN '50K-100K'
            WHEN income > 100000 AND income <= 500000 THEN '100K-500K'
            WHEN income > 500000 THEN '500K+'
            ELSE 'Unknown'
        END AS IncomeBracket
    FROM
        patients) AS IncomeBracketed
GROUP BY
    IncomeBracket
ORDER BY
    CASE IncomeBracket
        WHEN '0-1K' THEN 1
        WHEN '1K-10K' THEN 2
        WHEN '10K-50K' THEN 3
        WHEN '50K-100K' THEN 4
        WHEN '100K-500K' THEN 5
        WHEN '500K+' THEN 6
        ELSE 7
    END
```

Data Output | Messages | Notifications

	incomebracket text	averagecoverage double precision
1	0-1K	490905.81836363644
2	1K-10K	824126.4631939803
3	10K-50K	627516.2361551143
4	50K-100K	383668.8178207116
5	100K-500K	573441.8988333847
6	500K+	622949.2446915581

8. **Immunization Records:** How many patients have received each type of immunization?

```sql
-- 8. Immunization Records:
-- How many patients have received each type of immunization?
SELECT description, COUNT(patient) as Number_of_Patients
FROM immunizations
GROUP BY description
ORDER BY COUNT(patient) desc
```

Data Output | Messages | Notifications

	description character varying (500)	number_of_patients bigint
1	Seasonal Flu Vaccine	93219
2	Five doses of tetanus toxoid, preservative-free and adsorbed, for adults.	8434
3	Novel Coronavirus (COVID-19) mRNA Vaccine 30 mcg/0.3mL Dose	7563
4	Diphtheria, Tetanus, and Pertussis Vaccine	6693
5	Novel Coronavirus (COVID-19) mRNA Vaccine 100 mcg/0.5mL Dose	5993

9. **Patient Mortality:** What is the average age at death for patients, and how does it vary by gender and race?

```sql
-- 9. Patient Mortality:
-- What is the average age at death for patients, and how does it vary by gender and race?
SELECT gender, race,
    AVG(EXTRACT(YEAR FROM AGE(patients.deathdate, patients.birthdate)))AS Age_of_patient
FROM patients
WHERE deathdate IS NOT NULL
    AND birthdate IS NOT NULL
GROUP BY race, gender
```

Data Output | Messages | Notifications

	gender character varying (100)	race character varying (100)	age_of_patient numeric
1	F	hawaiian	61.6896551724137931
2	F	asian	55.6172839506172840
3	F	native	40.1250000000000000
4	F	black	56.7402597402597403
5	F	white	53.4388224471021159
6	F	other	43.6000000000000000

10. **Provider Workload:** Which healthcare provider has attended the most encounters?

```sql
-- 10. Provider Workload:
-- Which healthcare provider has attended the most encounters?
SELECT provider, COUNT(provider) AS count_encounter
FROM encounters
GROUP BY provider
ORDER BY count_encounter desc
LIMIT 1
```

Data Output | Messages | Notifications

	provider character varying (100)	count_encounter bigint
1	4e98e792-2919-3258-9159-025edd33f95c	13470

11. **Encounter Costs Analysis:** What are the average total claim costs and payer coverage for each encounter class?

```sql
-- 11. Encounter Costs Analysis:
-- What are the average total claim costs and payer coverage for each encounter class?
SELECT encounterclass, AVG(total_claim_cost) as average_claim, AVG(payer_coverage) as average_coverage
FROM encounters
GROUP BY encounterclass
```

Data Output | Messages | Graph Visualiser | Notifications

	encounterclass character varying (100)	average_claim double precision	average_coverage double precision
1	ambulatory	4910.160586775049	4060.540984484715
2	emergency	3876.8564486398027	3052.0752583365756
3	home	692.5313028472748	566.3527610008498
4	hospice	11124.481733333325	9135.34462666667
5	inpatient	17915.354041520615	14461.583362092226
6	outpatient	1986.2282548432086	1188.7972674113473
7	snf	15394.501990171957	11386.712850122847
8	urgentcare	1034.4758773686017	787.8173932678461
9	virtual	981.3493511988781	868.4101833568405
10	wellness	1124.008335880981	534.0480825700955

12. **Marital Status and Health:** Is there a notable difference in the number of healthcare encounters between married and single patients?

```sql
-- 12. Marital Status and Health:
-- Is there a notable difference in the number of healthcare encounters between married and single patients?
SELECT p.marital, COUNT(e.id) AS NumberOfEncounters
FROM patients AS p
JOIN encounters AS e
ON p.id = e.patient
WHERE p.marital IS NOT NULL
    AND (p.marital = 'S' OR p.marital = 'M')
GROUP BY p.marital
```

Data Output | Messages | Graph Visualiser | Notifications

	marital character varying (100)	numberofencounters bigint
1	M	218310
2	S	75780

13. **Race and Ethnicity in Healthcare:** How do healthcare expenses vary across different races and ethnicities?

```sql
-- 13. Race and Ethnicity in Healthcare:
-- How do healthcare expenses vary across different races and ethnicities?
SELECT p.race, AVG(p.healthcare_expenses) AS avg_healthcare_exp, p.ethnicity
FROM patients AS p
WHERE p.race IS NOT NULL
    AND p.healthcare_expenses IS NOT NULL
    AND p.ethnicity IS NOT NULL
GROUP BY p.race, p.ethnicity
ORDER BY avg_healthcare_exp desc
```

Data Output | Messages | Graph Visualiser | Notifications

	race character varying (100)	avg_healthcare_exp double precision	ethnicity character varying (100)
1	black	305348.6220138888	nonhispanic
2	hawaiian	292749.60713178304	nonhispanic
3	other	288273.9413402062	nonhispanic
4	black	282769.26225806447	hispanic
5	white	274622.1006352431	nonhispanic
6	hawaiian	256504.12083333332	hispanic
7	asian	255636.98189189183	hispanic
8	asian	247047.32128637074	nonhispanic
9	white	246105.73668699188	hispanic
10	native	212815.9646808511	nonhispanic
11	native	164523.97500000003	hispanic
12	other	126811.84500000002	hispanic

14. **Chronic Condition Management:** Which patients with chronic conditions (conditions lasting more than 1 year) have the highest number of encounters?

```sql
-- 14. Chronic Condition Management:
-- Which patients with chronic conditions (conditions lasting more than 1 year) have the highest number of encounters?
SELECT
    EXTRACT(YEAR FROM AGE(c.stop, c.start)) AS Chronic_years,
    c.patient,
    c.encounter
FROM conditions AS c
WHERE c.start IS NOT NULL
    AND c.stop IS NOT NULL
    AND c.patient IS NOT NULL
    AND c.encounter IS NOT NULL
    AND EXTRACT(YEAR FROM AGE(c.stop, c.start)) <> 0
ORDER BY EXTRACT(YEAR FROM AGE(c.stop, c.start)) DESC;
```

Data Output | Messages | Graph Visualiser | Notifications

	chronic_years numeric	patient character varying (1000)	encounter character varying (1000)
1	52	bad6ff27-8229-ca89-fd82-10b8c93cc9bc	6bf53a35-752a-d26b-1a6e-0dcfcd1b01cb
2	51	5e0c39ce-38f7-d66b-aaac-0c61edfe8859	352484c5-1dab-49ab-dbec-5356d45b07...
3	48	13fa35d6-3006-417d-e598-3c03da223d73	48f8bbfc-f52c-6798-8237-2475193682e9

15. **Patient Coverage Gaps:** Identify patients with high healthcare expenses but low healthcare coverage.

```sql
-- 15. Patient Coverage Gaps:
-- Identify patients with high healthcare expenses but low healthcare coverage.
SELECT id,
    healthcare_expenses,
    healthcare_coverage
FROM patients
WHERE
    healthcare_expenses > (SELECT AVG(healthcare_expenses) FROM patients) -- Above average expenses
    AND healthcare_coverage < (SELECT AVG(healthcare_coverage) FROM patients) -- Below average coverage
```

Data Output | Messages | Graph Visualiser | Notifications

	id character varying (100)	healthcare_expenses double precision	healthcare_coverage double precision
1	cc53e99a-6715-603f-b074-eea93fe11e20	824192.36	31979.82
2	52de8610-de40-203d-4bd6-cf06141fa864	556114.94	11443.34
3	10d940ae-7114-8bcf-50f9-2bfd5354ff41	654622.19	270730.9