Lab 03. Spatial Data

Due date: Thursday, Feb 13 submitted as Word document to Canvas *Lab03* link. This lab counts 9 % toward your total grade.

Objectives:

1. Understand and apply logical operators to filter datasets.
2. Utilize Data Query Language (DQL) to retrieve and summarize data.
3. Perform Data Manipulation Language (DML) operations to update and modify records.
4. Use Data Definition Language (DDL) to alter table structures.
5. Implement subqueries to perform complex queries.

Format of answer: Submit your answers as a **Word document or pdf** with graphs (screenshot of your result table) and answer in SQL.

Notice: All SQL commands are in blue color

In-Class Exercise (3pts)

Recap:

DATA TYPES

Numeric Types

INTEGER (or INT): Represents whole numbers.

NUMERIC (or DECIMAL): Represents fixed-point numbers with exact precision.

String Types

VARCHAR: Represents variable-length character strings with a maximum length specified.

TEXT: Represents variable-length character strings with a maximum length that can be extremely large

SELECT
stusab,statea,
100.0 * (Sum(associate) + Sum(master)) /Sum(total_pop) AS asso_mastr_pct
FROM ch04.us_attribute
GROUP BY statea,stusab
ORDER BY asso_mastr_pct DESC;
Or
SELECT
stusab,statea,
100 * (Sum(associate) + Sum(master))::NUMERIC /Sum(total_pop) AS asso_mastr_pct
FROM ch04.us_attribute
GROUP BY statea,stusab
ORDER BY asso_mastr_pct DESC;

Import nyc_census_blocks from psql
Psql is the command-line interface (CLI) for PostgreSQL, allowing users to interact with a PostgreSQL database directly from the terminal.
Cmd is the CLI for windows.

1. search cmd and run as administrator, change and paste the following command to check the coordinate system of the data

```
ogrinfo -al -so "C:\path\to\your\shapefile.shp"
```

```
Tips:
```
- Ogrinfo is a command-line tool providing detailed metadata and structure information about **geospatial datasets,** such as Shapefiles, GeoJSON, KML, PostGIS, etc.

Task 1. What is the Projected CRS for this data?

Projected CRS: 26918(NAD83/UTM Zone 18N)

2. In cmd, change and run the following command lines to import .shp file (the password is needed after hitting enter)

First, navigate to PostgreSQL's bin directory:
```
cd C:\Program Files\PostgreSQL\17\bin
```

Second, run the shp2pgsql command from there:
```
shp2pgsql -I -s <SRID> "C:\path\to\your\shapefile.shp"
my_schema.my_table | psql -U your_user -d your_database
```

```
Tips:
```
- `-I` → **Creates a spatial index** for faster spatial queries.
- `-s <SRID>` → Sets the **SRID (e.g., Spatial Reference System - EPSG:4326 for WGS 84)**.
- `"C:\path\to\your\shapefile.shp"` → **Full path** to your shapefile.
- `My_schema.my_table` → The **PostgreSQL schema and table name**.
- `| psql -U your_user -d your_database` → Pipes the SQL output directly into **PostgreSQL**.
- `shp2pgsql` is a command-line tool that converts **ESRI Shapefiles (.shp)** into SQL statements.

Task 2. please paste your command line and make a screenshot to display the nyc_census_blocks table in pgAdmin.

You can now type the following line to connect to PostgreSQL
```
psql -U postgres -d your_database
```

```
Tips:
```
- `\l` → List databases
- `\c db_name` → Connect to a database
- `\dt` → List tables
- `\d table_name` → Describe a table
- `\du` → List users
- `SELECT * FROM table_name;` → Retrieve data

Using the nyc_census_blocks table, answer the following questions.

Here is some helpful information to get started.

blkid	A 15-digit code that uniquely identifies every census **block**. ("360050001009000")
popn_total	Total number of people in the census block
popn_white	Number of people self-identifying as "white" in the block
popn_black	Number of people self-identifying as "black" in the block
popn_nativ	Number of people self-identifying as "native american" in the block
popn_asian	Number of people self-identifying as "asias" in the block
popn_other	Number of people self-identifying with other categories in the block
hous_total	Number of housing units in the block
hous_own	Number of owner-occupied housing units in the block
hous_rent	Number of renter-occupied housing units in the block
boroname	Name of the New York borough. Manhattan, The Bronx, Brooklyn, Staten Island, Queens
geom	Polygon boundary of the block

Task 3. Logical Operators

Operator	Description	Example
AND	Returns TRUE if **both** conditions are met.	WHERE age > 18 AND city = 'New York'
OR	Returns TRUE if **either** condition is met.	WHERE age > 18 OR city = 'Los Angeles'
NOT	Reverses the condition (TRUE → FALSE , FALSE → TRUE).	WHERE NOT city = 'Chicago'
BETWEEN	Checks if a value is within a range.	WHERE price BETWEEN 10 AND 50
IN	Checks if a value matches **any value** in a list.	WHERE city IN ('New York', 'Los Angeles', 'Chicago')
LIKE	Searches for a **pattern** using wildcards (% , _).	WHERE name LIKE 'A%' (Names starting with "A")
IS NULL	Checks if a column contains NULL .	WHERE phone_number IS NULL

Question: Retrieve all census blocks where the total population exceeds 1000 and the number of owner-occupied housing units is greater than renter-occupied ones.

Task 4. Data Query Language (DQL)
AGGREGATE FUNCTION

Aggregate functions perform a calculation on a set of values and return a single, or summary, value.

FUNCTION	DESCRIPTION
AVG	Returns the average of all the values, or only the DISTINCT values, in the expression.
COUNT	Returns the number of non-null values in the expression. When DISTINCT is specified, COUNT finds the number of unique non-null values.
COUNT(*)	Returns the number of rows. COUNT(*) takes no parameters and cannot be used with DISTINCT.
MAX	Returns the maximum value in the expression. MAX can be used with numeric, character and datetime columns, but not with bit columns. With character columns, MAX finds the highest value in the collating sequence. MAX ignores any null values.
MIN	Returns the minimum value in the expression. MIN can be used with numeric, character and datetime columns, but not with bit columns. With character columns, MIN finds the value that is lowest in the sort sequence. MIN ignores any null values.
SUM	Returns the sum of all the values, or only the DISTINCT values, in the expression. SUM can be used with numeric columns only.

Question: List the total population and total housing units for each borough. Aggregate popn_total and house_total to each borough

Task 5. Data Manipulation Language (DML)
Question: Increase the number of renter-occupied housing units by 10% in all blocks where the borough is 'Manhattan'.
UPDATE STATEMENT

- The *UPDATE statement* changes data in existing rows either by adding new data or modifying existing data.

```
UPDATE ch04.us_attribute
SET gisjoin = 'MA_state'
WHERE gisjoin = 'MA';
```

Task 6. Data Definition Language (DDL)
KEY DDL COMMANDS

CREATE – To create databases, tables, or indexes.
ALTER – To modify existing structures.
DROP – To delete objects.
TRUNCATE – To remove all data without deleting the structure.
COMMENT – To add metadata.
RENAME – To rename tables or columns.

Check Column name and Data Type

```
SELECT column_name, data_type
FROM information_schema.columns
WHERE table_schema = 'ch04' AND table_name = 'us_attribute';
```

Question: Add a new column to the table to store the percentage of white population in each block. The output should display the attribute table with new columns.

Own Your Own (6pts)
Task 1. Data Manipulation Language (DML)
Question: Populate the new column 'perc_white' with the percentage of the white population relative to the total population.

Task 2. Advanced Logical Operators
Question: Retrieve blocks where the total population is less than 500 or the percentage of the white population exceeds 70%.

Task 3. Combining DQL and Spatial Queries
Question: Find all census blocks in 'Brooklyn' with a total population density greater than 10,000 people per square kilometer.
Hint: Using ST_Area(geom) to calculate the Area of each census tract

Task 4. Creating a New Table (DDL)
Question: Create a new table 'borough_summary' to store summarized population data by borough.

Column name	Data Type
boroname	VARCHAR(50)

population	INTEGER
housing	INTEGER

Task 5. Inserting Data (DML)
Question: Insert data into the new ' borough_summary ' table.

Task 6. Deleting Records (DML)
Question: Delete records from the 'borough_population_summary' table where the total population is less than 500,000.

Bonus (0.5pts): Using Subqueries
Question: Retrieve blocks where the total population is above the average population of all blocks.
Hint: To solve this, think about how to calculate the average population first using an aggregate function. Then, use a subquery to embed this average into the main query's WHERE clause to compare each block's population against it.
Hint: subquery syntax:

```
SELECT column_name
FROM table_name
WHERE column_name expression operator
   (SELECT column_name FROM table_name WHERE ...);
```