WEEK 04
INTRO TO SQL

Instructor: Yanan Wu
TA: Vanchy Li

Spring 2025

WEEK 04

LECTURE SESSION

Instructor: Yanan Wu
TA: Vanchy Li

Spring 2025

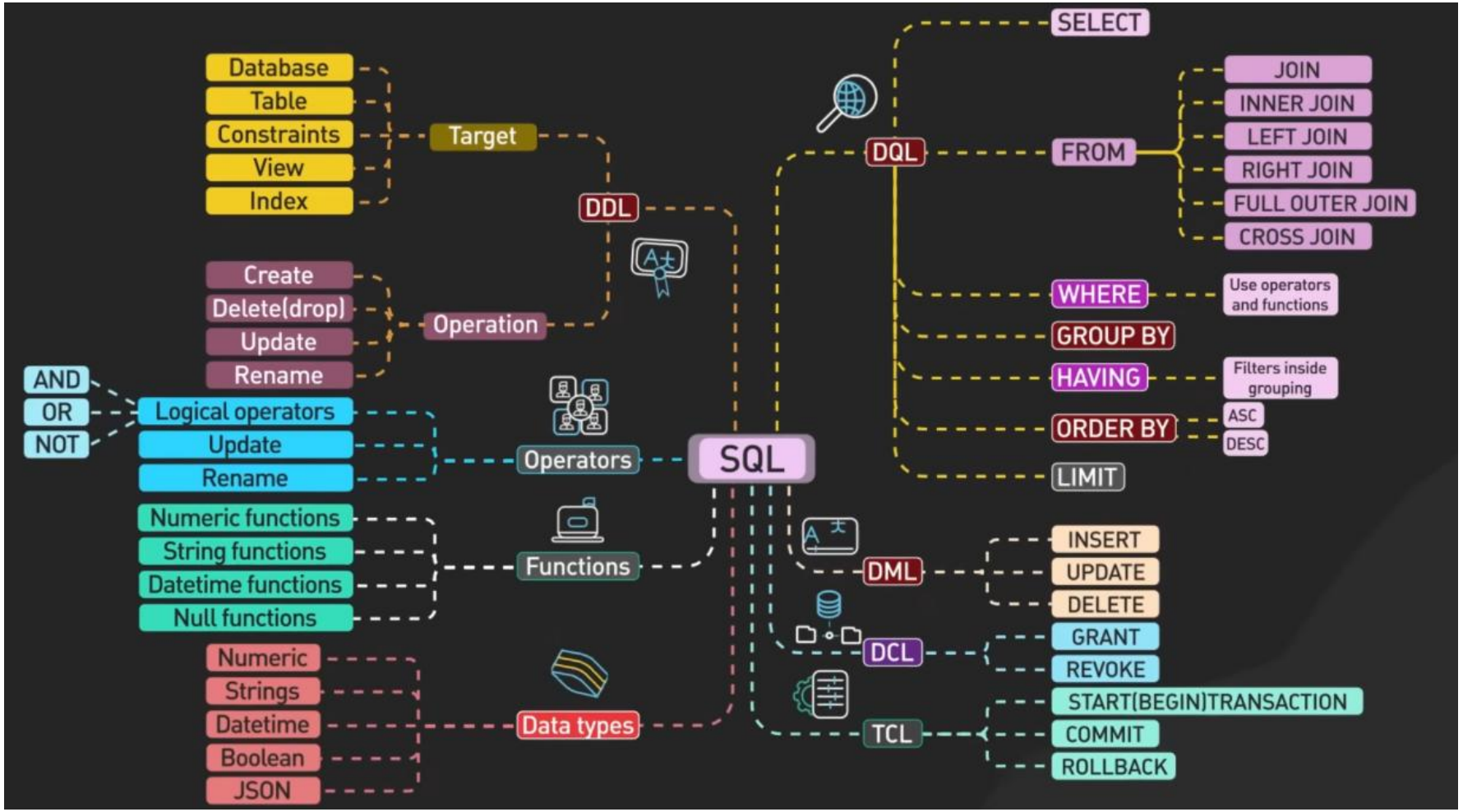
4.1 INTRO

STRUCTURED QUERY LANGUAGE

- SQL (Structured Query Language):

SQL is a standard language for managing relational databases. It provides a set of commands for performing various operations such as querying data, updating data, creating and modifying database schema, and managing access controls.

SQL COMMANDAS



SQL TABLE

- A SQL table is a fundamental component of a relational database, organizing data into rows and columns. Database engineers establish relationships between multiple tables to enhance data efficiency and optimize storage.

- For example, a database engineer might create a SQL table to store product details in a retail store:

Product ID	Product Name	Color ID
0001	Mattress	Color 1
0002	Pillow	Color 2

Color ID	Color Name
Color 1	Blue
Color 2	Red

4.2
SPATIAL DATA

SHAPEFILE

Shapefile: a collection of files with .shp, .shx, .dbf, and other extensions on a common prefix name (e.g., nyc_census_blocks). The actual shapefile relates specifically to files with the .shp extension. However, the .shp file alone is incomplete for distribution without the required supporting files.

Mandatory files:

.shp—shape format; the feature geometry itself

.shx—shape index format; a positional index of the feature geometry

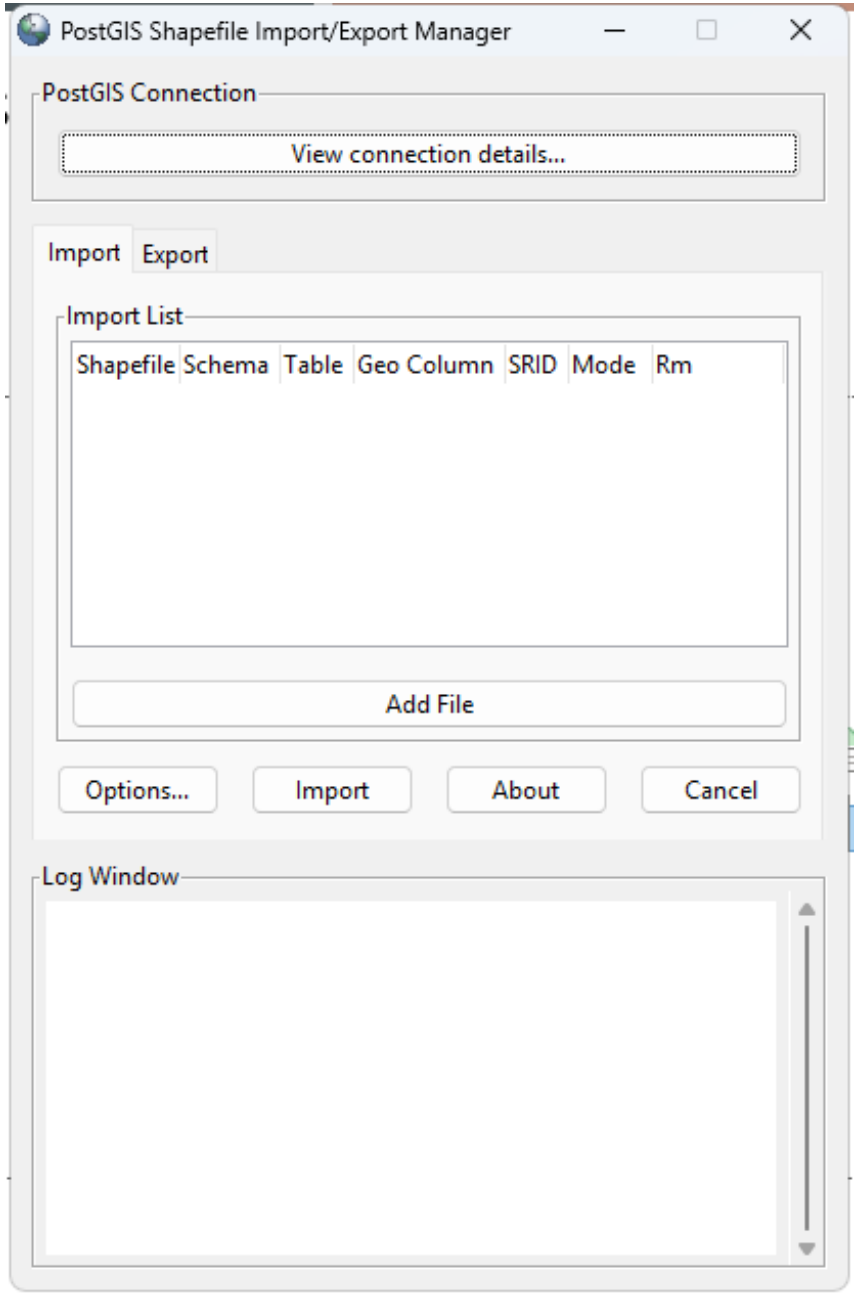
.dbf—attribute format; columnar attributes for each shape, in dBase III

Optional files include:

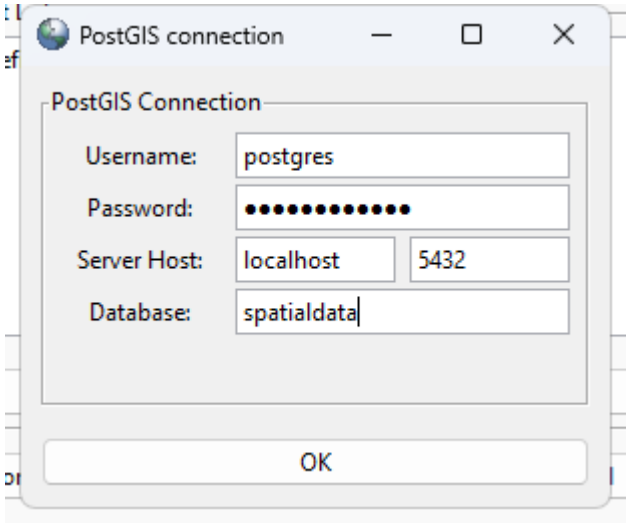
.prj—projection format; the coordinate system and projection information, a plain text file describing the projection using well-known text format

Loading with shp2pgsql -GUI

1) Open the Shapefile Import/Export Manager.
2) In the Shapefile field, browse and select US_tract_2020.shp.
3) Set the SRID to 102003 and Geo Column to geometry (or the SRID and Geo Column match your shapefile).
4) Choose a target schema (e.g., postgis) and set a table name (e.g., us_tracts).
5) Click Import. The tool will: Create a new table. Insert shapefile data into the table. Create a spatial index for faster querying.

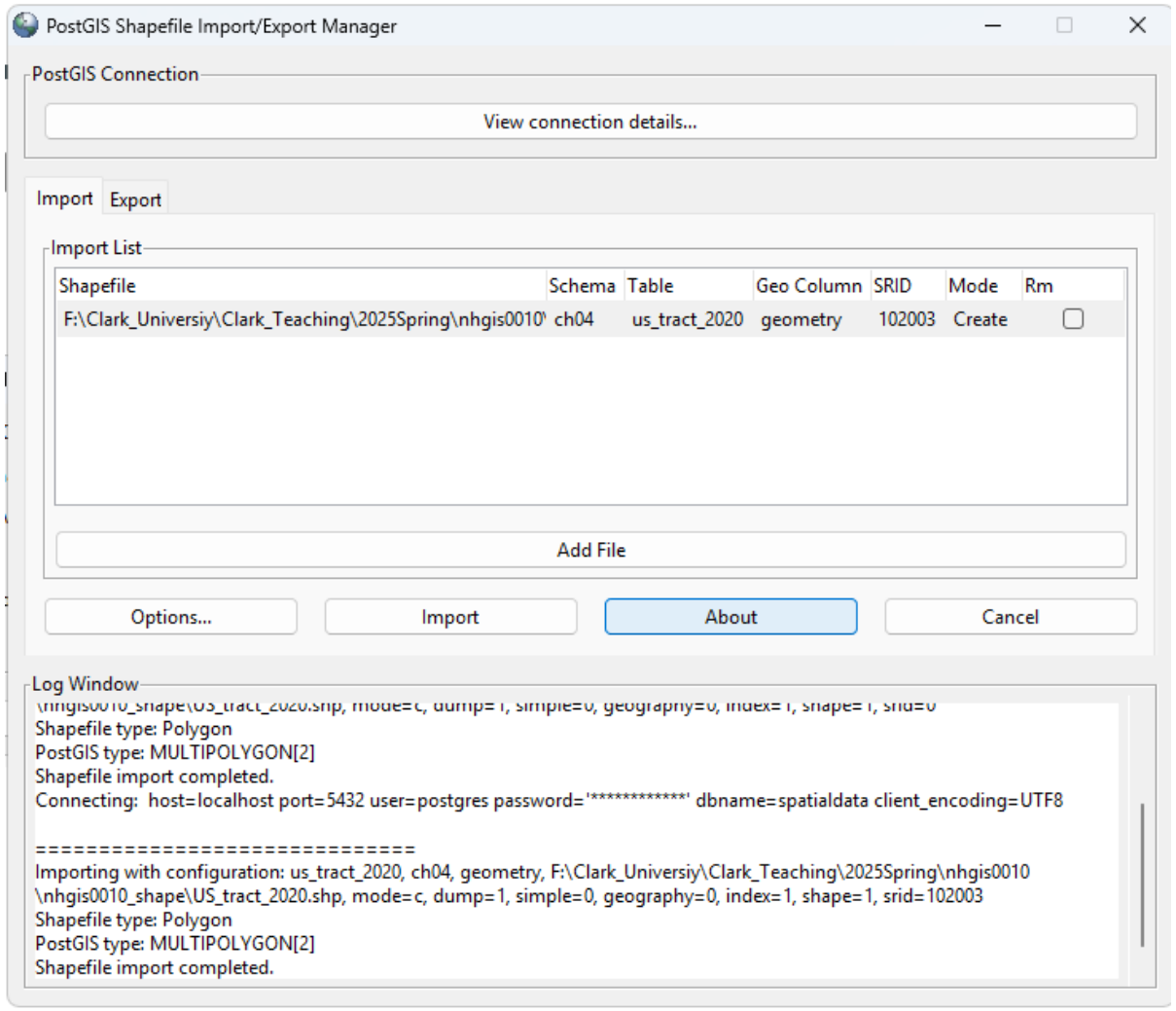


Loading with shp2pgsql -GUI



Geo Column: assign a name to the geometry column in data

SRID: spatial reference system



Loading with shp2pgsql -GUI

Step 3: Verify the Import

Check if the shapefile was imported successfully:

Open the query tool in pgAdmin.

Run the following query:

```
SELECT geometry FROM ch04.us_tract_2020 LIMIT 10;
```

```
CREATE TABLE ch04.us_attribute (
    GEO_ID VARCHAR(20) PRIMARY KEY, -- Census Geographic Area Identifier (Primary Key)
    GISJOIN VARCHAR(20), -- GIS Join Match Code
    STUSAB VARCHAR(20), -- State Abbreviation
    STATE VARCHAR(50), -- State Name
    STATEA VARCHAR(50), -- State Code
    COUNTY VARCHAR(50), -- County Name
    COUNTYA INTEGER, -- County Code
    TRACTA VARCHAR(20), -- Census Tract Code
    TL_GEO_ID VARCHAR(20), -- TIGER/Line Shapefile Geographic Area Identifier
    NAME_E VARCHAR(255), -- Full Geographic Area Name (Estimates)
    -- Population Data
    AQNFE001 INTEGER, -- Total Population
    AQNGE002 INTEGER, -- White Population
    AQNGE003 INTEGER, -- Black or African American Population
    AQNGE004 INTEGER, -- American Indian/Alaska Native Population
    AQNGE005 INTEGER, -- Asian Population
    AQNGE006 INTEGER, -- Native Hawaiian/Other Pacific Islander Population
    AQNGE007 INTEGER, -- Another Race Category
    -- Education Data
    AQPKE017 INTEGER, -- Some Education Field
    AQPKE021 INTEGER, -- Associate's Degree
    AQPKE022 INTEGER, -- Another Degree Field
    AQPKE023 INTEGER, -- Master's Degree
    AQPKE024 INTEGER, -- Professional School Degree
    AQPKE025 INTEGER  -- Doctorate Degree);
```

Import csv file

1) Create a table with column names, the column names should align with the column names in csv file
2) Import Data to table (right click table in server, select Import/Export data)

4.3
DATA TYPES IN SQL

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

DATA TYPES

Date/Time Types

DATE: Represents a date value without time.

TIMESTAMP: Represents date and time values.

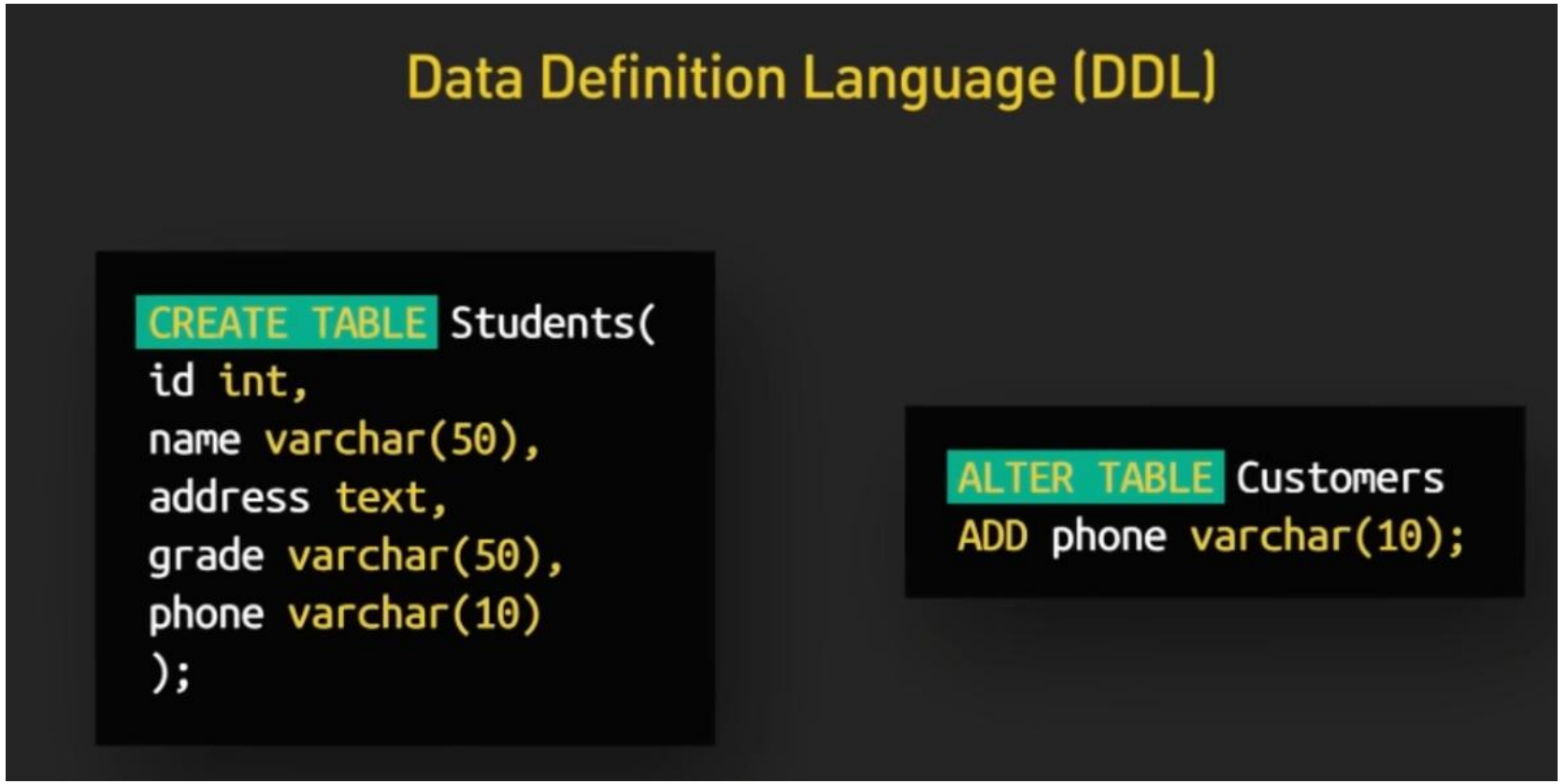
Boolean Type

BOOLEAN: Represents a boolean value (TRUE, FALSE, or NULL)

4.4 DATA DEFINITION LANGUAGES (DDL)

DATA DEFINITION LANGUAGE (DDL)

- DDL is used to define the structure and schema of a database. It includes commands for creating, modifying, and deleting database objects such as tables, indexes, views, and schemas.
- It does **not** manipulate data but manages the structure.



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.

CREATE STATEMENT

TOPICS ON WEEK 03

CREATE GEOMETRY

CREATE DATABASE …
CREATE TABLE…

ALTER STATEMENT

- Used to modify an existing table (add, modify, drop columns).

```
ALTER TABLE ch04. us_attribute ADD COLUMN POP VARCHAR(255);

ALTER TABLE ch04. us_attribute ALTER COLUMN POP TYPE SMALLINT;

ALTER TABLE ch04. us_attribute RENAME COLUMN POP TO POPULATION;

ALTER TABLE ch04. us_attribute DROP COLUMN POPULATION;
```

EXERCISE: RENAME TABLE NAME

```
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNFE001 TO total_pop;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNGE002 TO white_pop;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNGE003 TO black_pop;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNGE004 TO native_pop;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNGE005 TO asian_pop;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNGE006 TO pacific_pop;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQNGE007 TO other_race;
```

Sample But in MySQL

```
ALTER TABLE table_name
    RENAME COLUMN AQNFE001 TO total_pop,
    RENAME COLUMN AQNGE002 TO white_pop,
    RENAME COLUMN AQNGE003 TO black_pop,
    RENAME COLUMN AQNGE004 TO native_pop,
    RENAME COLUMN AQNGE005 TO asian_pop,
    RENAME COLUMN AQNGE006 TO pacific_pop,
    RENAME COLUMN AQNGE007 TO other_race;
```

PostgreSQL does not support renaming multiple columns at once, so if you are using PostgreSQL, you must keep one ALTER TABLE statement per line.

However, MySQL 8.0+ allows multiple RENAME COLUMN statements to be combined.

Check Column name and Data Type

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

4.5 DATA QUERY LANGUAGE

RENAME TABLE

```
ALTER TABLE ch04.us_attribute RENAME COLUMN AQPKE021 TO Associate;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQPKE022 TO Another;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQPKE023 TO Master;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQPKE024 TO Professional;
ALTER TABLE ch04.us_attribute RENAME COLUMN AQPKE025 TO Doctorate;
```

DQL

- DQL (Data Query Language) is used to retrieve records from the database.
- SELECT statement: Used to retrieve data from tables.
- Basic Queries, Filtering, and Aggregation.

THE SELECT STATEMENT

- Basic syntax

```
SELECT column1, column2 FROM table_name;
```

- Selecting all columns

```
SELECT * FROM ch04.us_attribute;
```

- Display unique value for column

```
SELECT DISTINCT * FROM ch04.us_attribute;
```

FILTERING DATA WITH WHERE

- Filtering by a single condition:

```
SELECT stusab, statea, total_pop FROM ch04.us_attribute WHERE stusab = 'New Jersey’;
```

Exercise:

Retrieve the gisjoin, statea, and total_pop columns for records where gisjoin is equal to 'MA'

FILTERING DATA WITH WHERE

- Using comparison operators (=, !=, >, <, >=, <=):

```
SELECT stusab, statea, total_pop FROM ch04.us_attribute WHERE total_pop > 10000;
```

Exercise

Retrieve the gisjoin, statea, and black_pop columns for records where black_pop is less than 10000

FILTERING DATA WITH WHERE

- Using logical operators (AND, OR, NOT):

```
SELECT stusab, statea, total_pop FROM ch04.us_attribute WHERE stusab = 'California' AND total_pop > 10000;
```

Exercise:

Retrieve the gisjoin, statea, and black_pop, master columns for records where gisjoin = ‘MA’, total_pop > 5000, master > 500

ORDER DATA WITH ORDER BY

Display the top 10 most populated counties.

```
SELECT stusab, statea, total_pop
FROM ch04.us_attribute
ORDER BY total_pop DESC
LIMIT 10;
```

Exercise:

Retrieve the stusab, statea, and total_pop columns for records which are 10 least populated region

GROUP DATA BY GROUP BY

The GROUP BY clause is used to create one output row per each group and produces summary values for the selected columns.

Find the number of counties in each state.

```
SELECT stusab, COUNT(COUNTY) AS county_count
FROM ch04.us_attribute
GROUP BY stusab
ORDER BY county_count DESC;
```

Exercise:

Retrieve the top 10 states (stusab) with the highest number of census tracts

4.6
DATA QUERY LANGUAGE: AGGREGATE FUNCTION

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.

TABLE SOURCE: HTTPS://OPENTEXTBC.CA/DBDESIGN01/CHAPTER/CHAPTER-SQL-DML/

CALCULATE THE TOTAL POPULATION BY STATE (SUM)

Find the total population for each state.

```
SELECT stusab, SUM(total_pop ) AS total_population
FROM ch04.us_attribute
GROUP BY stusab
ORDER BY total_population DESC;
```

For each census tract, what percentage of the population is white?

```
SELECT
  stusab,statea,
  100.0 * Sum(white_pop)/Sum(total_pop) AS white_pct
FROM ch04.us_attribute
GROUP BY statea,stusab
ORDER BY white_pct DESC;
```

Exercise:

Calculate the proportion of people with both an associate and a master's degree as a percentage of the total population, and sort the results in descending order.

FIND THE AVERAGE POPULATION OF COUNTIES (AVG)

Calculate the average population of all counties in each state.

```
SELECT stusab, AVG(total_pop) AS avg_population
FROM ch04.us_attribute
GROUP BY stusab
ORDER BY avg_population DESC;
```

PATTERN MATCHING USING LIKE

```
SELECT stusab, statea
FROM ch04.us_attribute
WHERE stusab LIKE 'Mass%’;
```

Exercise:

Calculate the total_pop for stusab WHERE stusab start with letter 'A'

PATTERN MATCHING USING LIKE

LIKE '% Mc'

- LIKE 'Mc%' searches for all records that end with the letters "Mc"

LIKE '%en%'

- LIKE '%en%' searches for all records that have the letters "en"

PATTERN MATCHING USING IN AND BETWEEN

```
SELECT stusab, statea, total_pop
FROM ch04.us_attribute
WHERE stusab IN ('Texas',  'Florida’);
```

```
SELECT stusab, statea, total_pop
FROM ch04.us_attribute
WHERE total_pop BETWEEN 10000 AND 20000;
```

CREATE A TABLE FROM EXISTING TABLE

```
CREATE TABLE ch04.MA AS
SELECT geo_id, stusab,statea, total_pop
FROM ch04.us_attribute
WHERE gisjoin IN ('MA');
```

4.7
DATA MANIPULATE LANGUAGE

INSERT STATEMENT

- The *INSERT statement* adds rows to a table. In addition,
  a) INSERT specifies the table or view that data will be inserted into.
  b) Column_list lists columns that will be affected by the INSERT.
  c) If a column is omitted, each value must be provided.
  d) If you are including columns, they can be listed in any order.
  e) VALUES specifies the data that you want to insert into the table. VALUES is required.

SYNTAX FOR INSERT

INSERT [INTO] Table_name | view name [column_list]
DEFAULT VALUES | values_list |

Exercise:

INSERT value to column geo_id,stusab, master, professional

ADD NEW COLUMN: ALTER

SYNTAX:

```
ALTER TABLE table_name
ADD COLUMN column_name data_type;
```

Exercise:

Insert a new column ‘Area’ to us_attribute table

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';
```