



WEEK 07 PROXIMITY ANALYSIS

Instructor: Yanan Wu
TA: Vanchy Li

Spring 2025



WEEK 07

LECTURE SESSION

Instructor: Yanan Wu
TA: Vanchy Li

Spring 2025

7.1

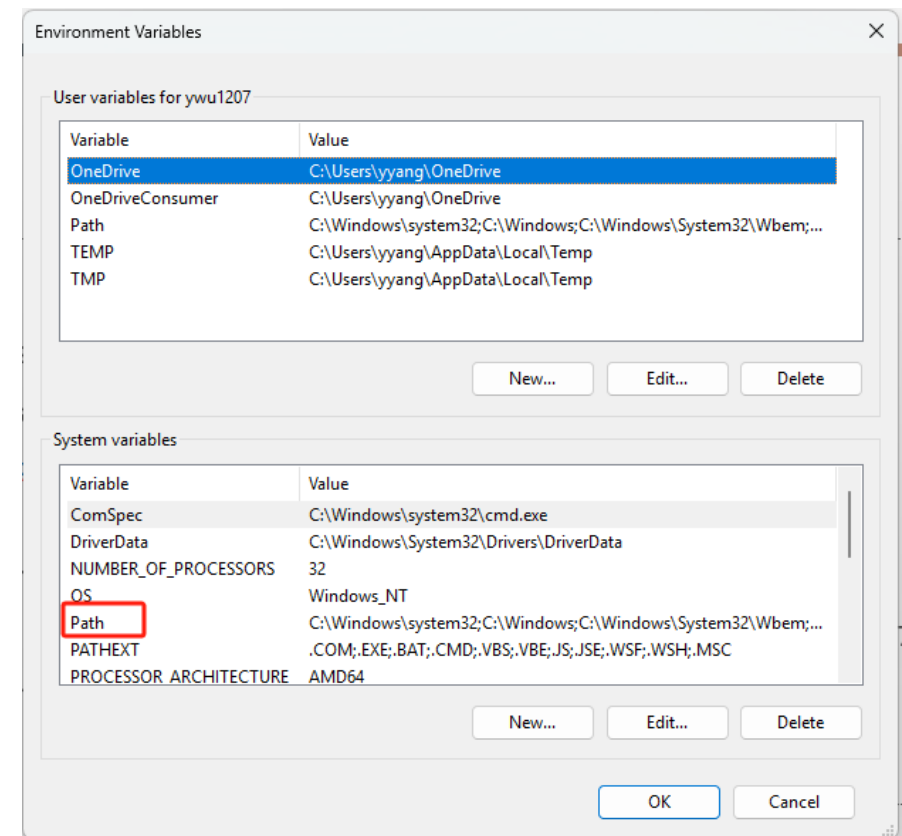
NEAREST NEIGHBOR

INTRODUCTION

- Proximity Analysis Overview:
 - Determines how far something is located from another object.
 - Examples include finding the nearest highway, estimating commute distances, or assigning field reps efficiently.
- Methods Covered:
 - Traditional closest neighbor searches.
 - k-Nearest Neighbor (KNN) indexes for performance optimization.
 - PostGIS clustering window functions for spatial grouping.

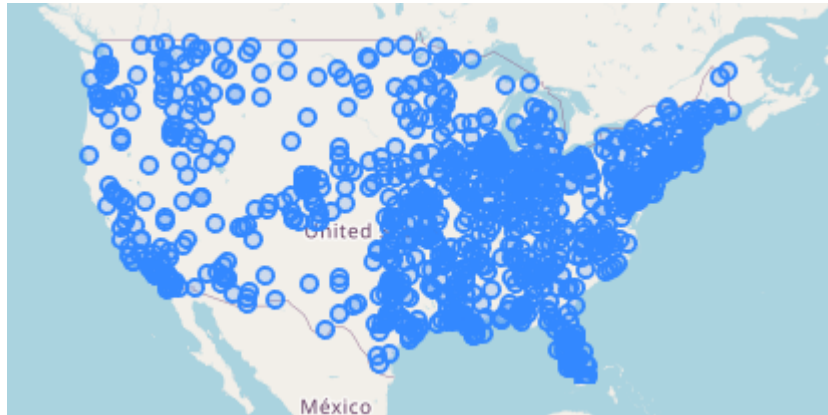
DATA

- ch09_data.sql construct data using SQL
- In terminal:
 - Load data into *spatial* database in pgadmin
`psql -U postgres -d spatial -f E:/Clark/ch09_data.sql`
- If your terminal does not recognize psql
 - Find the PostgreSQL installation folder
(e.g., C:\Program Files\PostgreSQL\17\bin\)
 - Copy the path of postgresql installation to system variable



DATA INTRODUCTION

■ Airports across U.S



id	7267
ident	0KY7
type	small_airport
name	Clinton-Hickman County Airport
lat	36.63610076904297
lon	-88.99859619140625
elevation_ft	360
continent	NA
iso_country	US
iso_region	US-KY
municipality	Clinton

■ Navigational aid across U.S



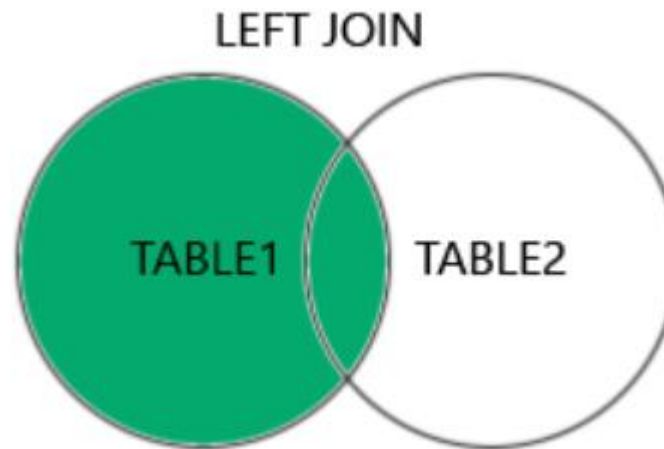
id	85349
filename	Al_Jouf_VORTAC_S
ident	AJF
name	Al Jouf
type	VORTAC
freq_khz	117800
lat	29.78899955749511
lon	40.07419967651367
elevation_ft	2261
iso_country	SA
dme_freq_khz	117800

NEAREST NEIGHBOR SEARCHES

- Which places are within X distance?
 - ST_Dwithin
- N closest results
 - ST_Dwithin
 - ST_Distance
 - LIMIT5

FIND CLOSET LOCATIONS

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.



FIND CLOSET LOCATIONS

- Please finds the closest navaid (navigational aid) to each airport

```
SELECT DISTINCT ON (a.ident)
  a.ident,
  a.name As airport,
  n.name As closest_navaid,
  (ST_Distance(a.geog,n.geog)/1000)::integer As dist_km

FROM   LEFT JOIN

ON ST_DWithin(a.geog, n.geog,100000)
ORDER BY a.ident, dist_km;
```

This query uses a left join instead of an inner join to ensure that even if you find no nav aids, the airport will still be in the results.

FIND CLOSET LOCATIONS – EXERCISE 01

- Please finds the closest navaid (navigational aid) to each airport and order the result from shortest distance to longest

Select * FROM () AS subquery ORDER BY dist_km



FIND CLOSET LOCATIONS – EXERCISE 02

- Please find the closest airport to each navaid (navigational aid)

ITEMS BETWEEN DISTANCES

- Sometimes, you need to find locations that are within a certain distance but also beyond a minimum distance (e.g., finding a place far enough for a workout but not too far).
 - Example: Finding an airport that is close enough to catch a flight but far enough to enjoy a scenic trip.
- This type of query requires checking two distance conditions:
 - The location should be **within the upper distance limit**.
 - The location should be **beyond the lower distance limit**.

```
SELECT name, iso_country, iso_region
FROM ch09.airports
WHERE ST_DWithin(geog,
    ST_Point(-75.8008, 40.2610)::geography, 100000)
AND NOT ;
```

K NEAREST NEIGHBOR

- A classic nearest-neighbor question is finding the N nearest points of interest to a fixed location.
- <->—This is the KNN distance operator for both geometry and geography.
 1. A <-> B returns the distance between two geometries A and B.
- Find 10 nearest large airport (type = 'large_airport') for Boston airport (ident = 'KBOS')

```
SELECT ident, name, geog,  
       geog <->
```

```
FROM ch09.airports  
WHERE ident != 'KBOS' AND type = 'large_airport'  
ORDER BY dist LIMIT 10;
```

K NEAREST NEIGHBOR – EXERCISE 01

- Find 10 nearest navigation aid for navigation aid (filename = 'Bedds_NDB_US')

WINDOW FUNCTION IN K NEAREST NEIGHBOR

- In previous examples, we only output the N closest places but did not number them.
- Sometimes, numbering these places instead of just listing distances is useful.
- ROW_NUMBER, and RANK window functions help in outputting an ordinal ordering of a row.
- A window function allows querying across the entire dataset while providing information about each row's position.
- Window functions work similarly to aggregate functions, but instead of collapsing multiple rows into one, they return all rows with additional computed values.

The difference between RANK and ROW_NUMBER is in how they handle ties.

RANK: 1,2,3,3,3,6,7.

ROW_NUMBER: 1,2,3,4,5,6,7

K NEAREST NEIGHBOR WITH ROW_NUMBER – EXERCISE 01

- Find 10 nearest airports for Boston Airport(`ident = 'KBOS '`)

```
WITH ranked_airports AS (  
  SELECT  
    ident,  
    name,  
    geog,  
    geog <-> (SELECT geog FROM ch09.airports WHERE ident = 'KBOS') AS dist,  
  
  FROM ch09.airports  
    WHERE ident != 'KBOS' AND type = 'large_airport'  
)  
SELECT rank_num, ident, name, geog, dist  
FROM ranked_airports  
WHERE rank_num <= 10;
```