Database Technology Investigation
Tracking Geolocation with PostGIS and PostgreSQL:
Native Geographic Information System (GIS)

Alexandra Miller

School of Computing and Design

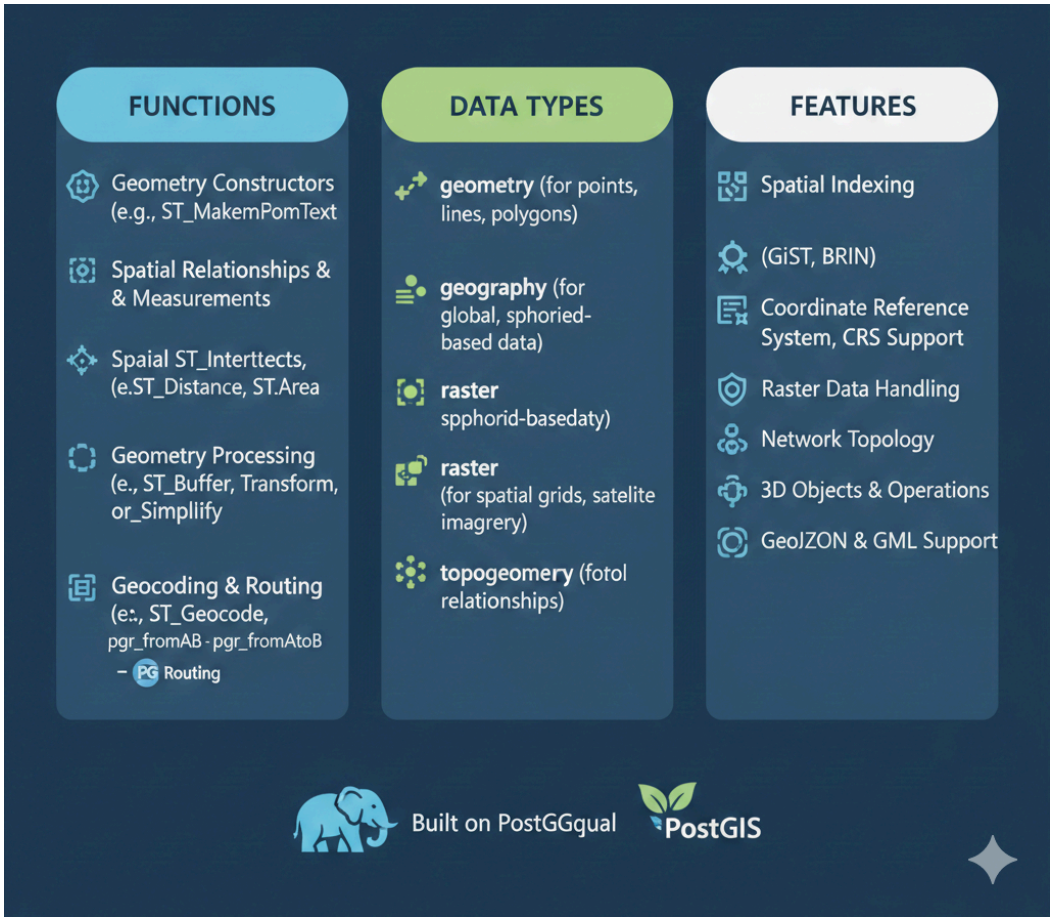
CST 363 Introduction to Databases - Spring 2026

February 23, 2026

PostGIS: Technology Description

PostGIS is a database extension that allows processing additional data types in PostgreSQL. It works by adding new data types (i.e. geometry, geography, and raster) and special functions to the standard PostgreSQL database, allowing users to perform complex spatial analysis using SQL queries. It is open source and maintained by the OSGeo Foundation. It is used by such U.S. organizations as USDA, FEMA and FCC.

By storing spatial data within a robust, transactional database system like PostgreSQL, PostGIS offers improved data consistency and provides better performance for large datasets compared to file-based workflows. Below are some of the features that PostGIS provides for PostgreSQL.



PostGIS: Purpose and Application

PostGIS is a powerful tool for handling ADS-B (Automatic Dependent Surveillance-Broadcast) data. Automatic Dependent Surveillance-Broadcast is an advanced, satellite-based surveillance technology that allows aircraft to determine their precise location via GPS and automatically broadcast it, along with altitude, velocity, and identification, to ground stations and other aircraft.

By utilizing the geometry or geography data types, PostGIS allows to store flight data, where the Z coordinate represents barometric altitude and the M (measure) value stores the Unix timestamp of each broadcast. This approach enables complex SQL queries that can calculate the distance between two aircraft, detect a plane entering restricted airspace, and identify the exact altitude of an aircraft from a spatial snapshot of ADS-B location data.

From a practical database implementation standpoint, PostGIS leverages spatial indexing (such as GiST indexes) to make queries more efficient for high-frequency ADS-B data processing. Determining the "Closest Point of Approach" (CPA) between two flights is possible by using special functions like ST_Distance and ST_ClosestPoint, allowing high precision analysis. Because ADS-B data is often received from multiple ground stations with varying accuracy, PostGIS functions like ST_Simplify can be used to clean up "overloaded" GPS tracks, while ST_Transform ensures that coordinates are projected correctly for accurate distance measurements in meters rather than degrees. This turns a standard PostgreSQL database into a sophisticated flight-tracking engine capable of monitoring and historical analysis.

PostGIS: Project Implementation

For this database technology investigation project, PostGIS was used to process the ADS-B (Automatic Dependent Surveillance-Broadcast) data and enable querying of geospatial data in order to determine a location of an aircraft using a callsign. The table “flight_tracking” is implemented using a special PosGIS-enabled data type “geomerty” to store location data. This allows utilizing special functions to obtain an aircraft location.

```
(base) skynet@skynetnode ~ % docker exec -it postgis_dev psql -U postgres -c "
SELECT
    flight_callsign,
    ST_Y(location_geom::geometry) AS latitude,
    ST_X(location_geom::geometry) AS longitude,
    altitude_ft
FROM flight_tracking
WHERE flight_callsign LIKE '%DAL197%';
"
 flight_callsign | latitude  |  longitude  | altitude_ft
-----------------+-----------+-------------+-------------
 DAL197          | 58.772552 | -170.540594 |       36350
(1 row)
```

Image: PostGIS geolocation query using aircraft callsign “DAL197”

How The Technology Differs from RDMBS

PostGIS is a spatial extension for the PostgreSQL, transforming it into a powerful spatial database by adding support for geographic objects, spatial data types (geometry/geography), and specialized indexing (GiST-based R-Tree indexing). While a standard RDBMS handles tabular data, PostGIS enables SQL-based, high-performance spatial analysis such as location proximity, area and intersection. Essentially, PostGIS transforms a traditional RDBMS into a specialized spatial database engine, allowing it to act as a backend for Geographic Information Systems.

Limitations of the Technology

While PostGIS is a premier tool for spatial analysis, it has some structural and performance limitations when used for high-frequency tracking and monitoring.

Standard PostgreSQL is a row-based relational database designed for transactional integrity (ACID). The nature of live stream data from ADS-B systems can cause issues.

- **Write Throughput:** Continuous streams can lead to index bloat and CPU saturation. With each new position requiring an update of the GIST index, computationally it can be expensive compared to the B-tree index.
- **Computational Expense of "geography" vs. "geometry":** To track aircraft, “geography” is used to account for the Earth's curvature. In order to perform a simple distance check on a “geometry” plane a sqrt() call is used, while a similar check on a “geography” sphere involves multiple sin(), cos(), and arctan() calls.
- **Function Limits:** Not all PostGIS functions support the geography type. Some cases require casting data to geometry (using a local projection like UTM) to perform complex operations, which subsequently adds processing overhead.

Appendix

Building a Geolocation Tracker

In order to build a Geolocation Tracker for aircraft, a location data source is needed, such as an ADS-B collecting server.

- This project uses“ADS-B Exchange” (adsbexchange.com) as a source of location data.
- The data for aircraft geolocation snapshots used for import to “flight_tracking” table is located at this link: adsbexchange.com)
- Utilizing **Docker** to build an instance of PostGIS using “Dockerfile” allows to customize the version of PostgreSQL (18.1) and PostGIS (3.6.2)

  Dockerfile:

  ```
  docker build -t my-custom-postgis .
  docker run --name postgis_dev -e POSTGRES_PASSWORD=geowiz -p 5433:5432 -d custom-postgis
  ```

  Docker build command:

  ```
  docker build -t custom-postgis
  ```

- The below command can be used to ensure current version is correct:
- ```
  docker exec -it postgis_dev psql -U postgres -c "SELECT version();"
  ```

- To activate the PostGIS extension:

  ```
  docker exec -it postgis_dev psql -U postgres -c "CREATE EXTENSION IF NOT EXISTS postgis;"
  ```

- Creating the “flight_tracking” table:

  ```
  docker exec -it postgis_dev psql -U postgres -c "
  CREATE TABLE flight_tracking (
  id serial PRIMARY KEY,
  time_stamp TIMESTAMP,
  icao_hex VARCHAR(10),
  flight_callsign VARCHAR(20),
  altitude_ft INTEGER,
  ground_speed_kts NUMERIC,
  location_geom GEOMETRY(PointZ, 4326)
  );"
  ```
  ```

Appendix (Continued)

- The data for aircraft geolocation snapshots used for import to “flight_tracking” table is located at this link: (adsbexchange.com)

- After importing the data to the flight_tracking table (see data import script here), the querying for geolocation of airplanes can be done as following:

```
docker exec -it postgis_dev psql -U postgres -c “
SELECT
    flight_callsign,
    ST_Y(location_geom::geometry) AS latitude,
    ST_X(location_geom::geometry) AS longitude,
    altitude_ft
FROM flight_tracking
WHERE flight_callsign LIKE '%DAL197%';
“
```