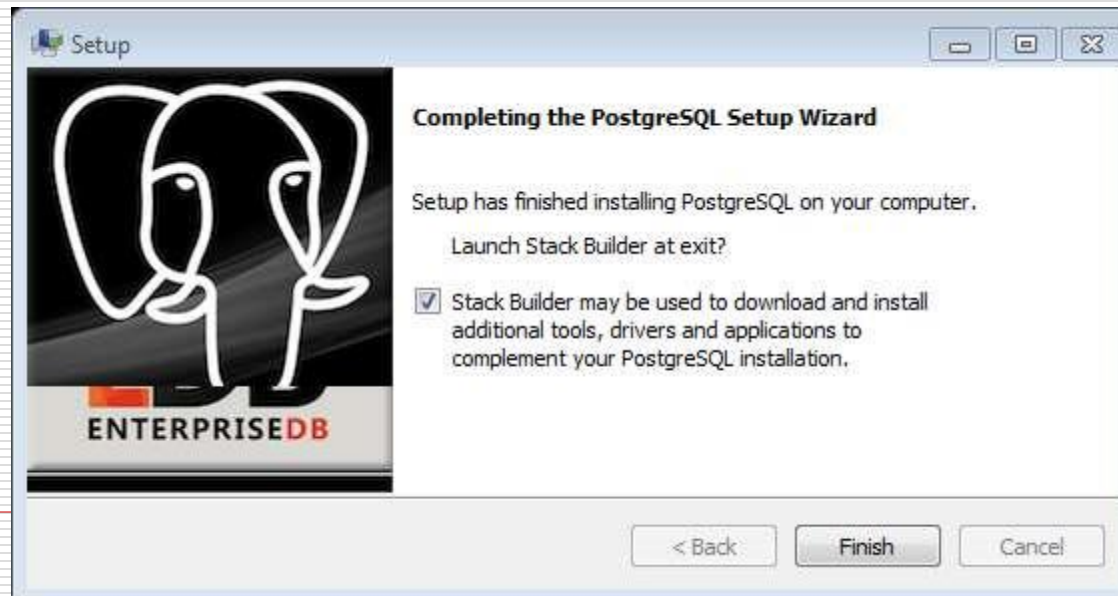
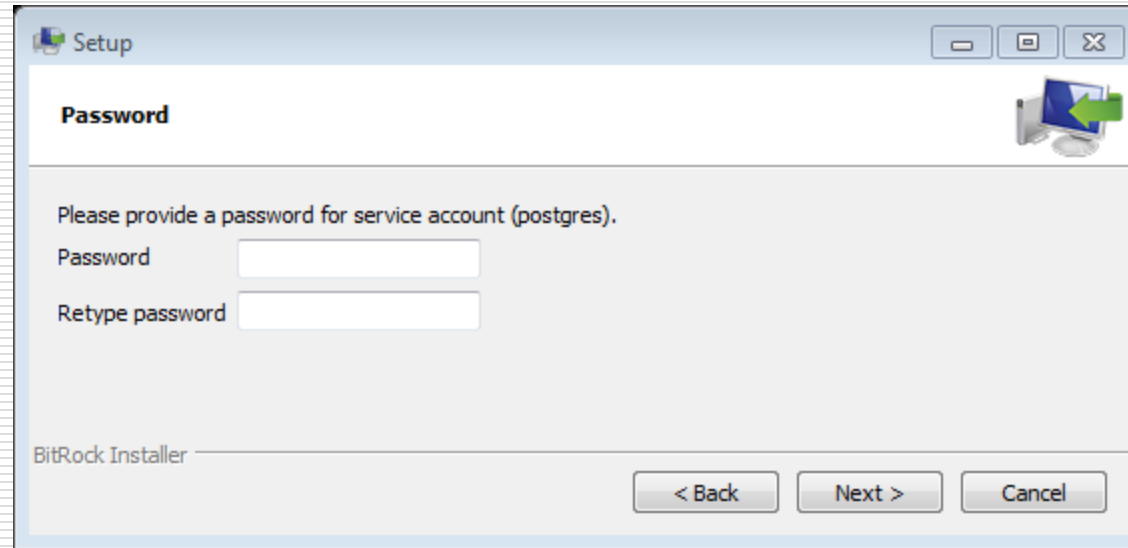


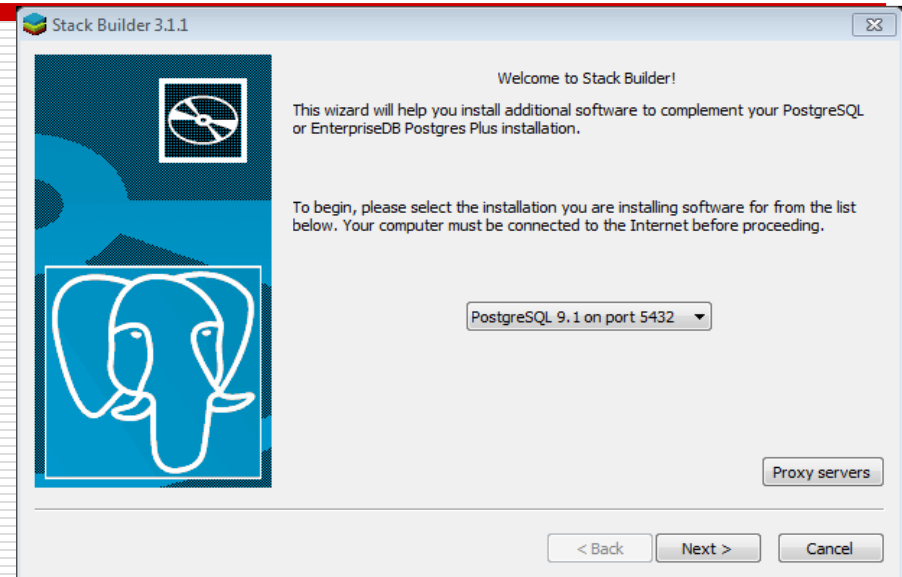
Cài đặt PostGreSQL

- ❖ Tải PostgreSQL: postgresql-9.5.18-3-windows
- ❖ Các bạn tiến hành cài đặt bình thường như các chương trình bình thường
- ❖ Nhập và nhớ password
- ❖ Tick vào tục cài thêm spatial extention: Postgis trong stack builder

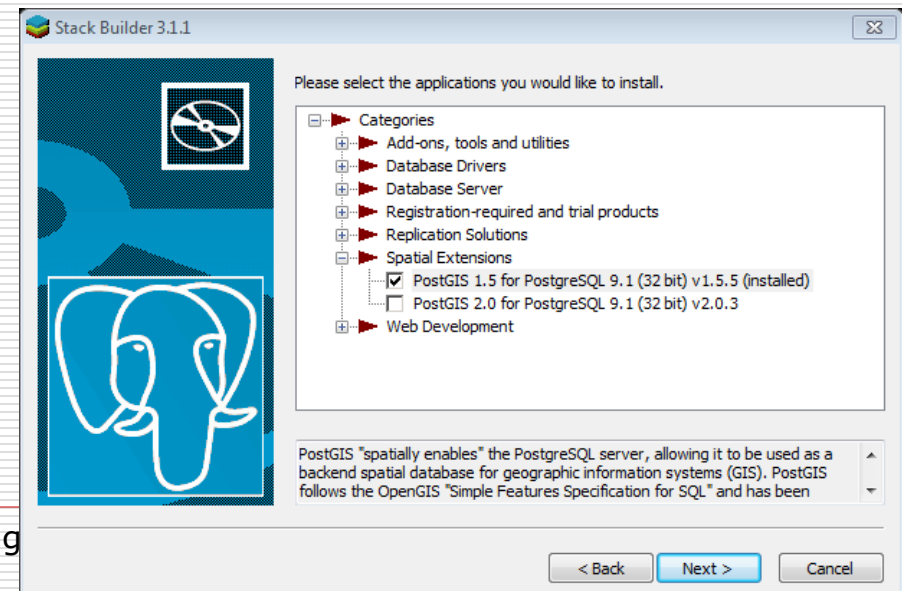


Cài đặt PostGreSQL

- ❖ Chọn PostGreSQL ứng với phiên bản cài đặt: 9.5
- ❖ Cổng mặc định là 5432

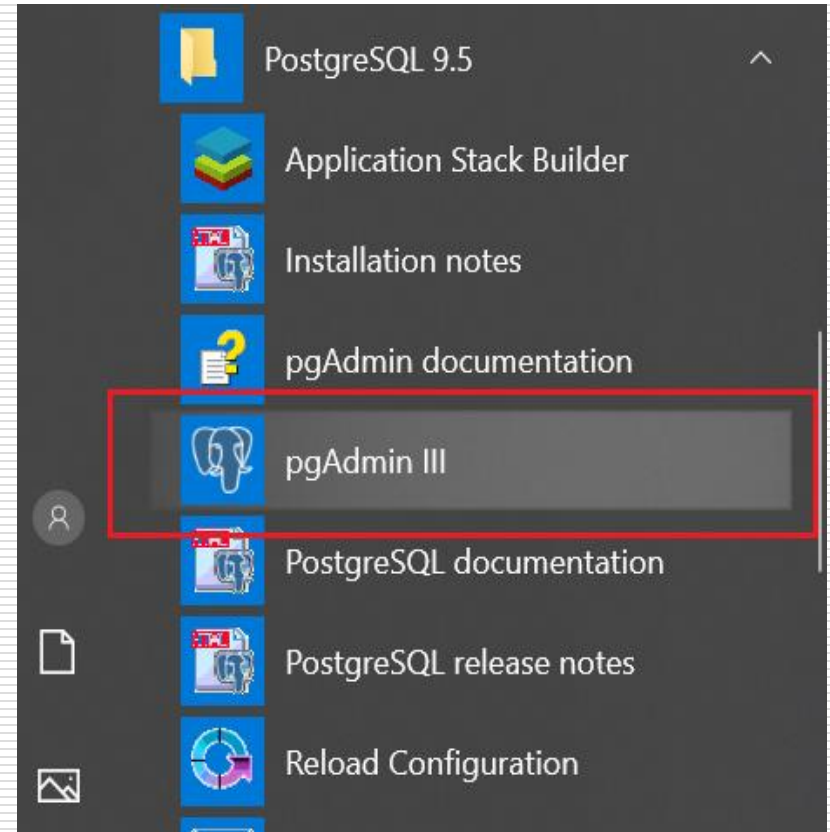


- ❖ Trong spatial extension bạn có thể chọn cài bản PostGIS ứng với PostgresSQL 9.5



Quản trị PostgreSQL với pgAdmin

- ❖ pgAdmin miễn phí và là công cụ quản trị giao diện đồ họa mà nguồn mở cho PostgreSQL, hỗ trợ tất cả các tính năng của PostgreSQL
- ❖ Có thể dùng trên Linux, FreeBSD, Solaris, Mac OSX và Window
- ❖ Khởi động pgAdmin III

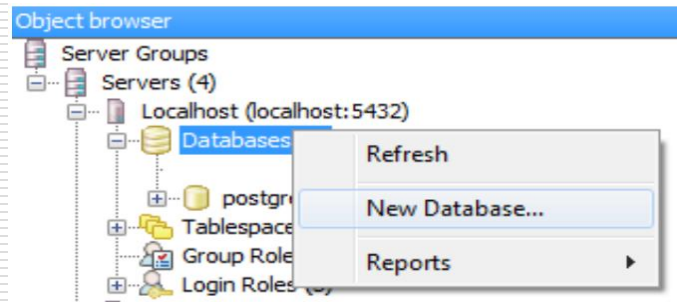


Làm việc với CSDL không gian

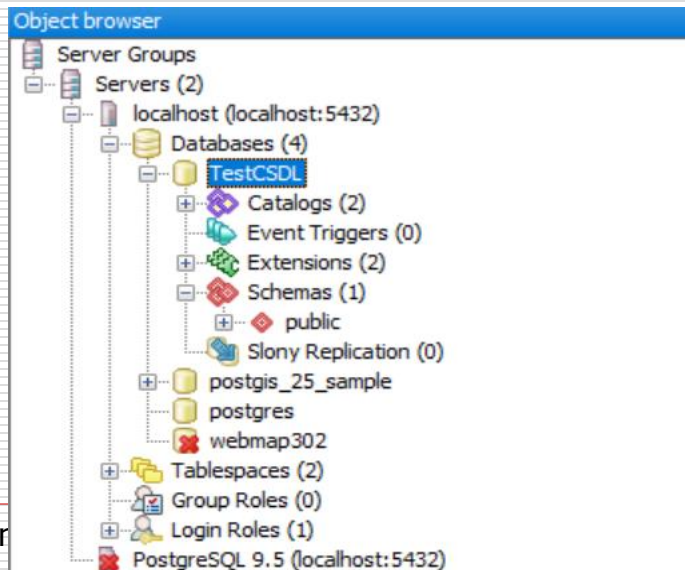
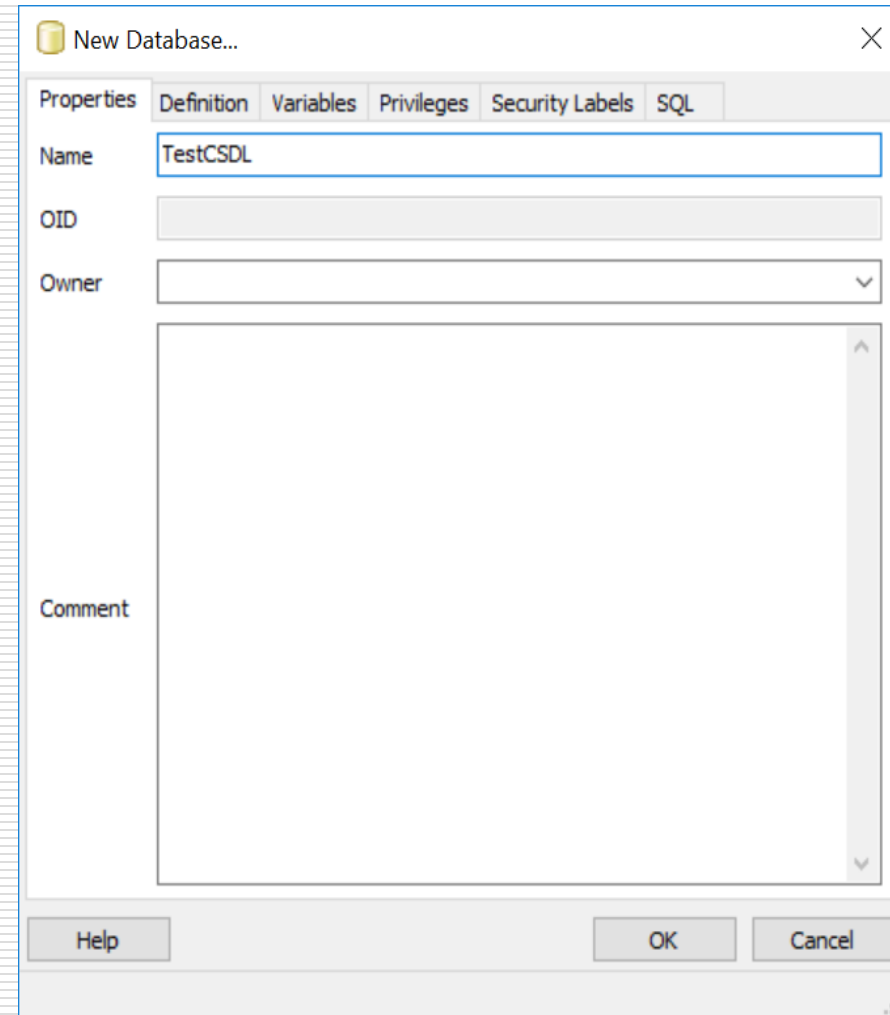
- ❖ Tạo CSDL không gian
- ❖ Thêm dữ liệu shapefile vào CSDL không gian
- ❖ Kiểm tra mã SRID
- ❖ Bảng siêu dữ liệu GEOMETRY_COLUMNS
- ❖ Bảng dữ liệu không gian
- ❖ Một số hàm trong POSTGIS

Tạo CSDL không gian

- ❖ B1: Click chuột phải mục Database, chọn New Database

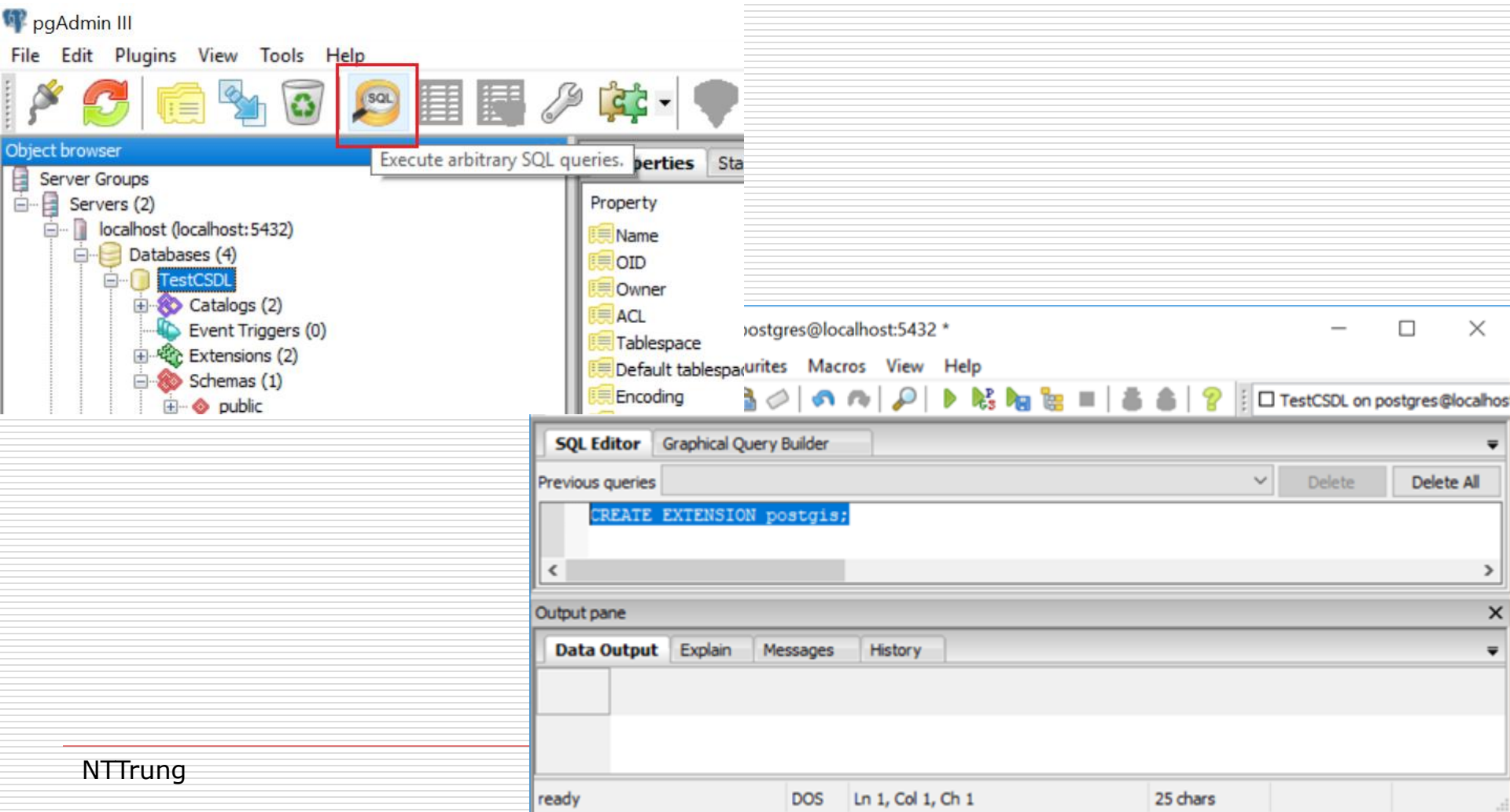


- ❖ B2: Nhập tên Database
- ❖ B3: Click OK



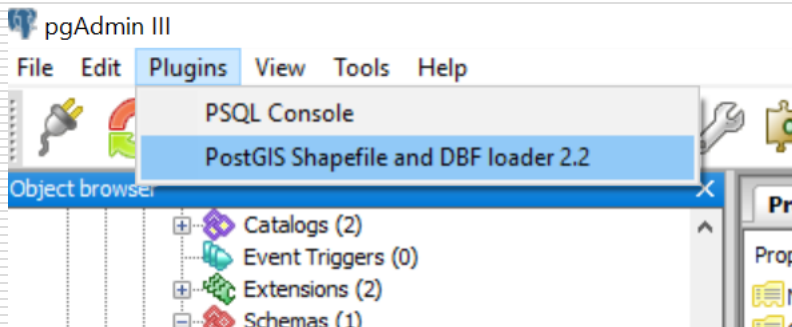
Tạo CSDL không gian

- ❖ B4: Để CSDL có thể lưu trữ dữ liệu không gian, mở cửa sổ SQL và thực thi lệnh: **CREATE EXTENSION postgis;**

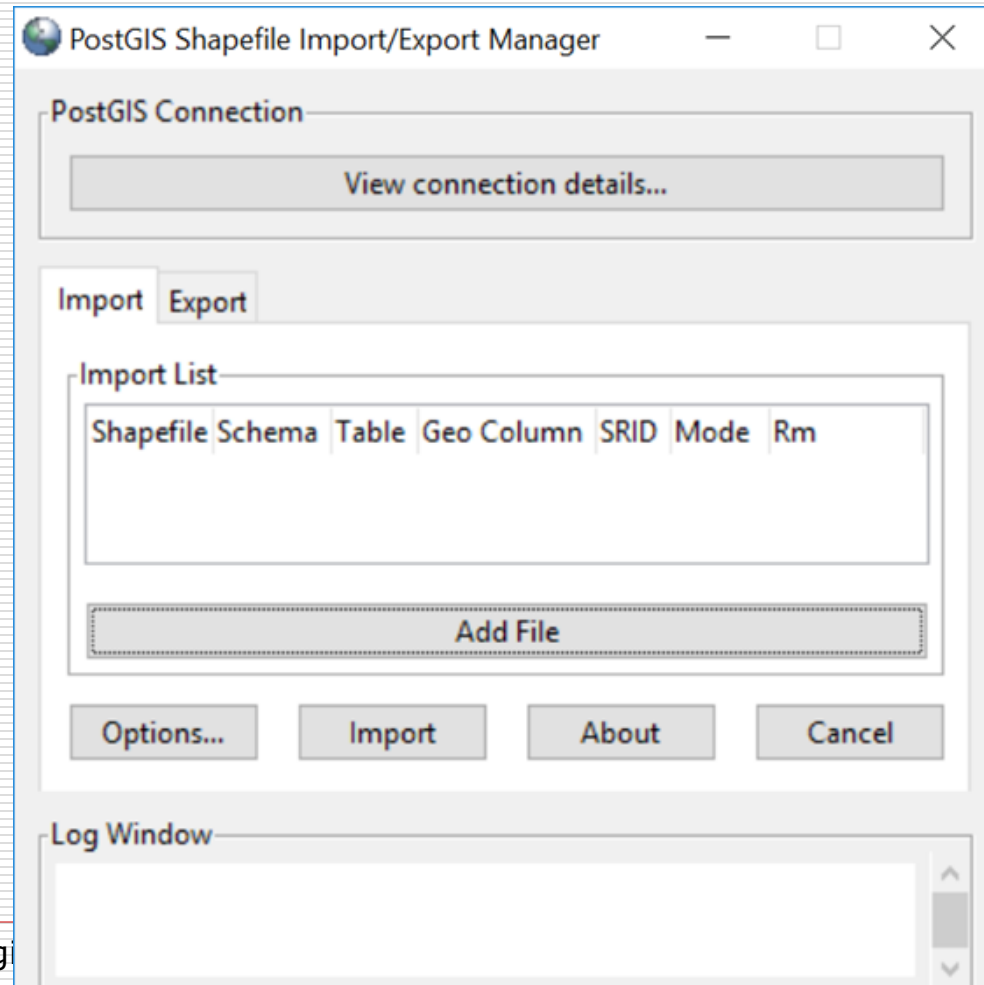


Thêm dữ liệu shapefile vào CSDL không gian

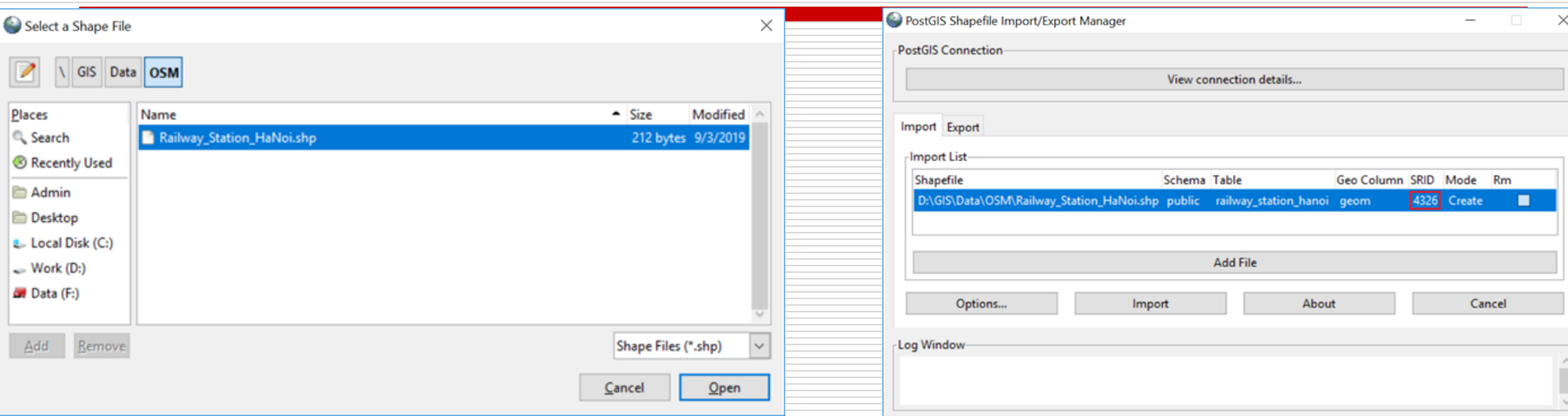
- ❖ B1: Click menu Plugins -> Chọn PostGIS Shapefile and DBF loader 2.2 mở hộp thoại PostGIS Shapefile Import/Export



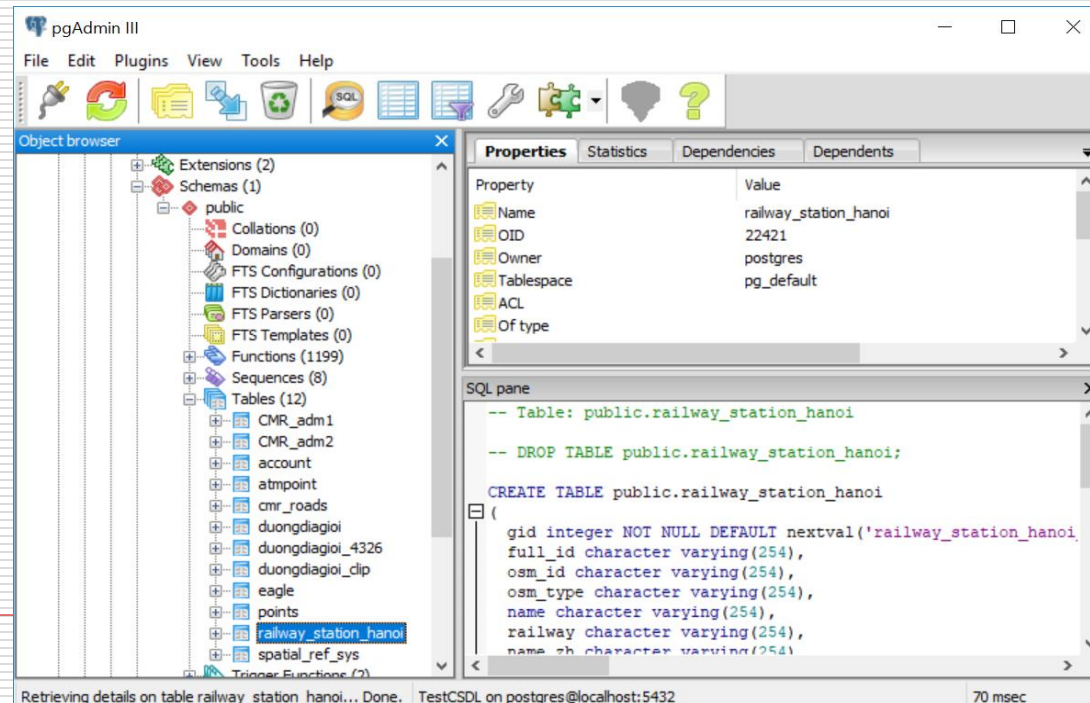
- ❖ B2: Click Add File mở hộp thoại tìm shapfile



Thêm dữ liệu shapefile vào CSDL không gian

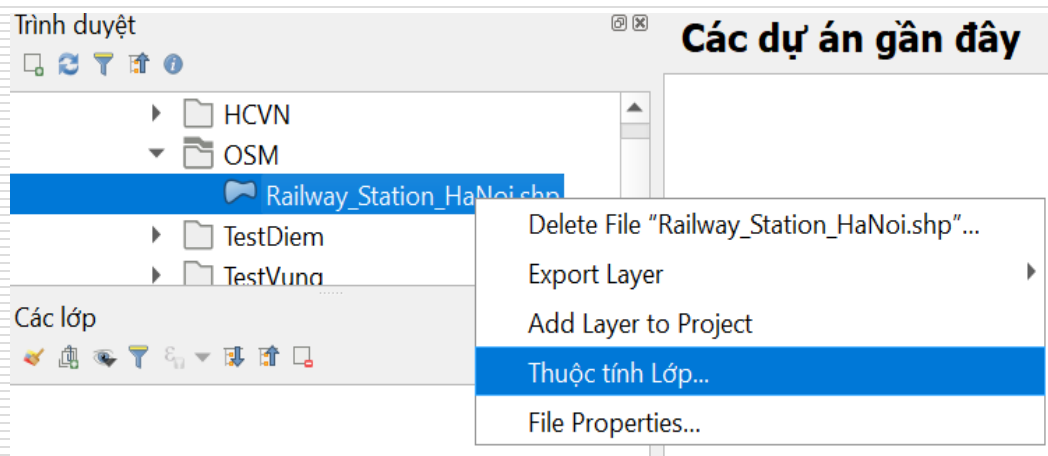


- ❖ B3: Tìm shapefile muốn nạp vào CSDL và click Open
- ❖ B4: Nhập mã SRID
- ❖ B5: Click Import

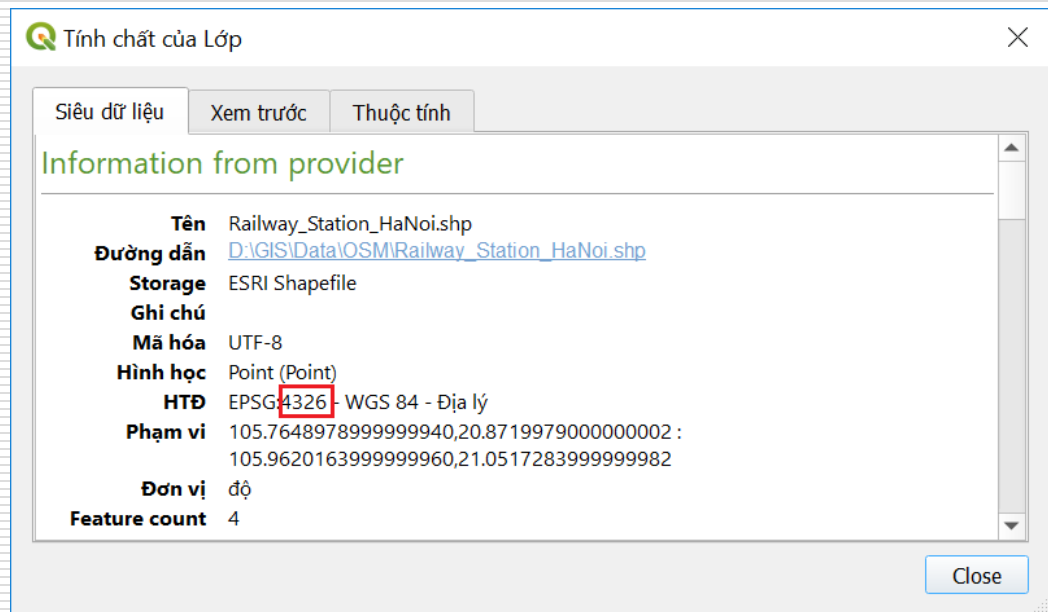


Kiểm tra mã SRID

❖ B1: Click chuột phải vào file dữ liệu -> chọn Thuộc tính lớp để mở hộp thoại Thuộc tính lớp



❖ B2: Xem mã SRID tại dòng HTĐ, sau chữ EPSG

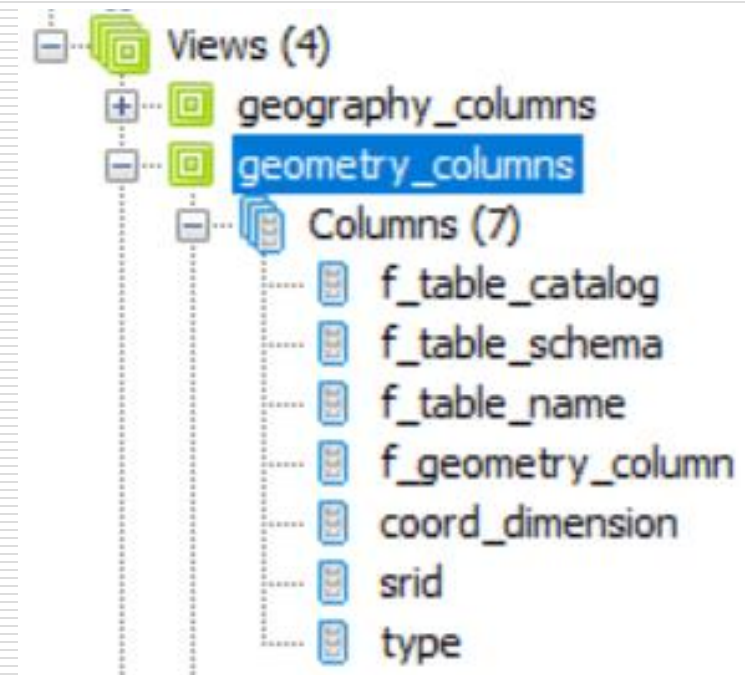


Bảng siêu dữ liệu GEOMETRY_COLUMNS

- ❖ Bảng GEOMETRY_COLUMNS mô tả về bảng đã tồn tại được kích hoạt không gian trong cơ sở dữ liệu
- ❖ Khi thực thi hàm AddGeometryColumn() có thể dùng để thêm đồng thời một cột không gian vào bảng phi không gian và cập nhật bảng GEOMETRY_COLUMNS
- ❖ Cấu trúc bảng GEOMETRY_COLUMNS
 - ❖ f_table_catalog: tên CSDL
 - ❖ f_table_schema: tên schema
 - ❖ f_table_name: tên bảng
 - ❖ f_geometry_column: tên cột không gian
 - ❖ coord_dimension: số chiều không gian của dữ liệu

Bảng siêu dữ liệu GEOMETRY_COLUMNS

- ❖ Cấu trúc bảng GEOMETRY_COLUMNS (tiếp)
 - ❖ srid: mã hệ thống tham chiếu không gian
 - ❖ Type: mô tả kiểu hình học được mô tả trong bảng, sử dụng một trong các kiểu sau: **POINT**, **LINESTRING**, **POLYGON**, **MULTIPOINT**, **MULTILINESTRING**, **MULTIPOLYGON**, **GEOMETRYCOLLECTION**



Bảng dữ liệu không gian

- ❖ Bảng không gian là bảng chứa một hoặc nhiều cột không gian
- ❖ Bao gồm những cột có kiểu dữ liệu thông thường và những cột kiểu dữ liệu không gian
- ❖ Cột không gian chỉ có thể chấp nhận kiểu dữ liệu hình học, bao gồm: **Point, Multipoint, Linestring, MultiLinestring, Polygon, Multipolygon**
- ❖ Tạo bảng không gian:
 - ❖ **Cách thông thường**
 - ❖ **Dùng hàm AddGeometryColumn**

Cách tạo bảng không gian thông thường

- ❖ Tạo bảng với câu lệnh CREATE TABLE và một thuộc tính của bảng sẽ có kiểu dữ liệu dạng “geometry”
 - ❖ Ví dụ: CREATE TABLE points (name varchar, point geometry)
- ❖ Chú ý: khi chèn dữ liệu vào bảng không gian cần chú ý đến trường có kiểu dữ liệu dạng “geometry”, định dạng dữ liệu nhập phải theo chuẩn định dạng WKT, ví dụ:
 - ❖ Để nhập dữ liệu cho đối tượng POINT có tọa độ (0,0), chúng ta dùng dạng POINT(0 0)
 - ❖ Để nhập dữ liệu cho đối tượng LINESTRING nối tọa độ (0,0) và (3, 4), chúng ta dùng dạng LINESTRING (0 0,3 4);
- ❖ Ví dụ thêm điểm vào bảng points:
 - ❖ INSERT INTO points VALUES ('a', 'POINT(0 0)');

Tạo bảng dùng AddGeometryColumn

- ❖ B1: Tạo bảng thông thường (không phải bảng dữ liệu không gian)
- ❖ B2: Thêm cột không gian vào bảng sử dụng hàm “AddGeometryColumn”:
AddGeometryColumn(<schema_name>,<table_name>,<column_name>,<srid>,<type>,<dimension>)
 - ❖ <schema_name>: tên schema của bảng cần thêm cột không gian
 - ❖ <table_name>: tên bảng cần thêm cột không gian
 - ❖ <column_name>: tên cột không gian cần thêm
 - ❖ <srid>: mã duy nhất của hệ thống tham chiếu không gian trong phạm vi của CSDL
 - ❖ <type>: xác định kiểu hình học cho cột
 - ❖ <dimension>: số chiều dữ liệu

Tạo bảng dùng AddGeometryColumn

- ❖ Nếu đang ở sơ đồ hiện tại thì bỏ qua thông số <schema_name>: AddGeometryColumn (<table_name>,<column_name>,<srid>,<type>,<dimension>)
- ❖ Ví dụ: Tạo bảng points_funccol có một cột không gian
 - ❖ Tạo bảng points_funccol: Create table points_funccol (id int, name varchar);
 - ❖ Thêm trường the_geom: Select AddGeometryColumn('public','points_funccol','the_geom',-1,'POINT',2);
 - ❖ Thêm dữ liệu: INSERT INTO points_funccol VALUES (0, 'a', 'POINT(0 0)');

Một số hàm trong POSTGIS

- ❖ Nhóm hàm điều khiển
 - ❖ AddGeometryColumn
 - ❖ DropGeometryColumn
 - ❖ DropGeometryTable
- ❖ Hàm khởi tạo hình học
- ❖ Hàm trả về kiểu hình học ở đầu ra

AddGeometryColumn

❖ Chức năng:

- ❖ Thêm một cột hình học vào bảng đã tồn tại
- ❖ Hàm này rất quan trọng trong việc tạo bảng trong CSDL không gian

❖ Cú pháp:

- ❖ `text AddGeometryColumn(varchar table_name, varchar column_name, integer srid, varchar type, integer dimension);`
- ❖ `text AddGeometryColumn(varchar schema_name, varchar table_name, varchar column_name, integer srid, varchar type, integer dimension);`

DropGeometryColumn

- ❖ Chức năng:
 - ❖ Loại bỏ một cột hình học từ bảng không gian
- ❖ Cú pháp:
 - ❖ text DropGeometryColumn(varchar table_name, varchar column_name);
 - ❖ text DropGeometryColumn(varchar schema_name, varchar table_name, varchar column_name);
- ❖ Ví dụ:
 - ❖ `SELECT DropGeometryColumn('public','points_funccol','the_geom');`

DropGeometryTable

- ❖ Chức năng:
 - ❖ Loại bỏ bảng và tất cả những gì tham chiếu trong cột hình học
- ❖ Cú pháp:
 - ❖ `text DropGeometryTable(varchar table_name);`
 - ❖ `text DropGeometryTable(varchar schema_name, varchar table_name);`
- ❖ Ví dụ:
 - ❖ `SELECT DropGeometryTable('public','points_funccol');`

Hàm khởi tạo hình học

- ❖ Tên hàm:
 - ❖ `ST_GeometryFromText`
- ❖ Chức năng:
 - ❖ Trả về giá trị được chỉ định `ST_Geometry` từ hiển thị WKT
- ❖ Cú pháp:
 - ❖ `geometry ST_GeometryFromText(text WKT);`
- ❖ Ví dụ:
 - ❖ `Select ST_GeomFromText('POINT(1 0)');`

Hàm trả về kiểu hình học ở đầu ra

- ❖ Tên hàm:
 - ❖ ST_AsText
- ❖ Chức năng:
 - ❖ Trả về hiển thị dạng WKT của hình
- ❖ Cú pháp:
 - ❖ text ST_AsText(geometry g);
- ❖ Ví dụ:
 - ❖ **Select**
ST_AsText('010100000000000000000000F03F0000000000000000');
trả về POINT(1 0)
 - ❖ **Select ST_AsText(ST_GeomFromText('POINT(1 0)'));** trả về
POINT(1 0)

