

CSE 594: Spatial Data Science & Engineering

Lecture 4
Spatial SQL Part 2

Spark SQL UDFs

- User Defined Functions helps to define new column-based functions
- Generate new features according to your needs
- Extend the functions available in the framework and reuse in multiple dataframes

Define UDFs in Scala

Approach 1

- Define a function to fulfill your requirement
- Set the function as an input parameter to the *udf* function
- Use udf with dataframes

```
import org.apache.spark.sql.functions.udf
val upper: String => String = _.toUpperCase
val upperUdf = udf(upper)
df.withColumn("upper", upperUdf('text'))
```

Approach 2

- Define a function to fulfill your requirement
- Register it with `SparkSession.udf.register`
- Use udf with SQL queries

```
val squared = (s: Long) => {
    s * s
}

spark.udf.register("square", squared)
spark.sql("select amount, squared(amount) from test")
```

Define UDFs in PySpark

Approach 1

- Define a function to fulfill your requirement
- Set the function as an input parameter to the *udf* function
- Use udf with dataframes

```
def do_square(s):  
    return s*s
```

```
from pyspark.sql.functions import udf  
from pyspark.sql.types import LongType  
squared_udf = udf(do_square, LongType())  
df.withColumn("squared", squared_udf('amount'))
```

Approach 2

- Define a function to fulfill your requirement
- Register it with `SparkSession.udf.register`
- Use udf with SQL queries

```
from pyspark.sql.types import LongType  
spark.udf.register("squared_udf", do_square, LongType())  
spark.sql("select amount, squared_udf(amount) from test")
```

PostGIS Raster

- An extension of PostGIS that supports raster file formats
- Implement raster type similarly to geometry type supported by PostGIS
- Supports raster loading, conversion from raster to geometry and vice versa, raster algebra operations

Raster Data Format in PostGIS

- RASTER is a column type (similar to GEOMETRY type)
- A table with a raster type column means a full raster coverage
- A table row with a raster type column means a raster tile
- Each raster tile has: a pixel size, a width and a height, a georeference, a variable number of band, a pixeltype per band, etc.

Raster Tiles

- Used to divide raster data into small manageable areas
- The basic data I/O unit for transferring raster data over the network
- Efficient for handling large raster datasets with limited memory and space
- Zooming to a small two-tile area means only two rows from the raster data table needs to be fetched
- Tile size controls the number of pixels stored in each row
- Zoomable raster maps consist of many tiles placed next to each other, ordered in a pyramid scheme

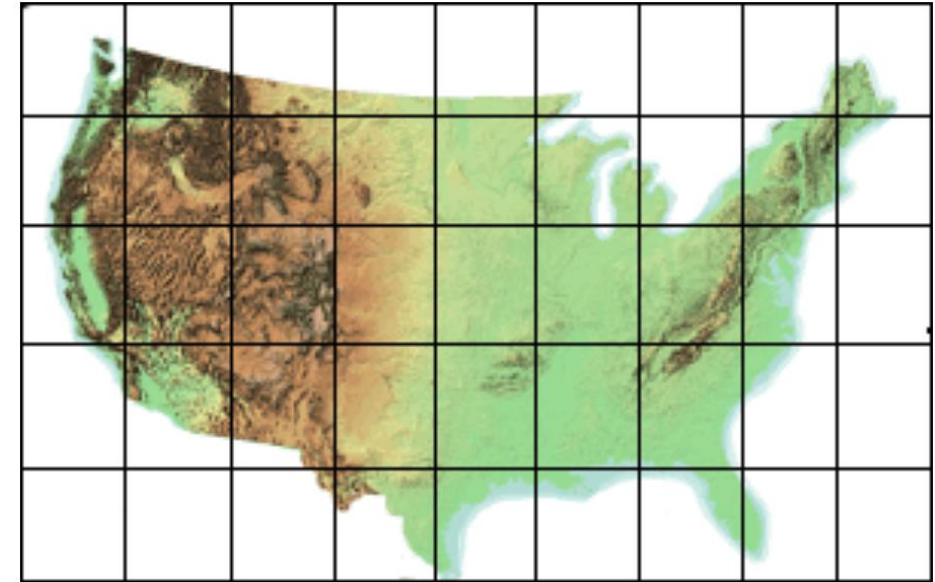


Image Source: ArcGIS Resource Center

PostGIS Raster

Creating Table with Raster Type Column

```
CREATE TABLE myrasters(rid serial primary key, rast raster);
```

Raster Constraints

- blocksize_x – the width of each raster tile
- blocksize_y – the height of each raster tile
- num_bands – number of bands in each tile
- srid – spatial reference identifier
- pixel_types – an array defining pixel type of each band
- spatial_index – true if raster column has a spatial index
- nodata_values – array of double precision numbers denoting nodata value for each band

PostGIS Raster

Raster Band Pixel Types

Describe data type and size of each values stored in each cell of a band

11 Pixel Types

- 1BB – 1 bit boolean
- 2BUI – 2 bit unsigned integer
- 4BUI – 4 bit unsigned integer
- 8BSI – 8 bit signed integer
- 16BSI – 16 bit signed integer
- 16BUI – 16 bit unsigned integer
- 32BSI – 32 bit signed integer
- 32BUI – 32 bit unsigned integer
- 32BF – 32 bit float
- 64BF – 64 bit float

Raster Support Data Types in PostGIS

geomval

- A compound data type consisting of a geometry object referenced by the .geom field and val, a double precision value that represents the pixel value at a particular geometric location in a raster band.
- It is used by the ST_DumpAsPolygon and Raster intersection family of functions as an output type to explode a raster band into geometry polygons.

addbandarg

- A composite type used as input into the ST_AddBand function defining the attributes and initial value of the new band
 - index (integer) - 1-based value indicating the position where the new band will be added amongst the raster's bands. If NULL, the new band will be added at the end of the raster's bands.
 - pixeltype (text) - Pixel type of the new band.
 - initialvalue (double) - Initial value that all pixels of new band will be set to.
 - nodataval (double) - NODATA value of the new band. If NULL, the new band will not have a NODATA value assigned.

Raster Support Data Types in PostGIS

rastbandarg

- A composite type for use when needing to express a raster and a band index of that raster.
 - rast (raster) – the raster to represent
 - nband (integer) - 1-based value indicating the band of raster

raster

- A spatial data type used to represent raster data such as those imported from JPEGs, TIFFs, PNGs, digital elevation models. Each raster has 1 or more bands each having a set of pixel values. Rasters can be georeferenced.

unionarg

- A composite type used as input into ST_Union function defining bands to be processed and behavior of UNION operation.
 - nband (integer) - 1-based value indicating the band of each input raster to be processed
 - uniontype (text) – type of union operations, such as min, max, count, mean, etc.

Raster Support Data Types in PostGIS

summarystats

- A composite type returned by the ST_SummaryStats and ST_SummaryStatsAgg functions.
 - count (integer) – number of pixels counted for the summary statistics.
 - sum (double) – sum of all counted pixels
 - mean (double) – arithmetic mean of all counted pixels
 - stddev (double) – standard deviation of all counted pixels
 - min (double) – minimum value of all counted pixels
 - max (double) – maximum value of all counted pixels

Creating Rasters with PostGIS Raster Functions

Approach 1

- Create empty rasters with `ST_MakeEmptyRaster`
- Add raster elements `ST_AddBand`

Approach 2

- Create raster from geometries
- Use `ST_AsRaster` along with `ST_Union` and other map algebra operations

Approach 2

- Create new raster tables from existing tables
- Change projection of existing raster with `ST_Transform` and create new raster

PostGIS Raster SQL Functions

Raster Constructors

- ST_AddBand - Returns a raster after adding new bands at a given index (at the end if no index)
- ST_AsRaster - Converts a PostGIS geometry to a PostGIS raster.
- ST_Band - Returns one or more bands of an existing raster as a new raster. Useful for building new rasters from existing rasters
- ST_MakeEmptyRaster – Returns an empty raster (having no bands) of given dimensions (width & height), upperleft X and Y, pixel size and rotation (scalex, scaley, skewx & skewy) and reference system (srid)
- ST_Tile – Returns a set of rasters resulting from the split of the input raster based upon the desired dimensions of the output rasters.
- ST_FromGDALRaster - Returns a raster from a supported GDAL raster file

PostGIS Raster SQL Functions

Raster Accessors

- ST_NumBands - Returns the number of bands in the raster object.
- ST_Height - Returns the height of the raster in pixels.
- ST_Width - Returns the width of the raster in pixels.
- ST_PixelHeight - Returns the pixel height in geometric units of the spatial reference system.
- ST_PixelWidth - Returns the pixel width in geometric units of the spatial reference system.
- ST_SRID - Returns the spatial reference identifier of the raster
- ST_Summary - Returns a text summary of the contents of the raster.

PostGIS Raster SQL Functions

Raster Inputs/Outputs

- ST_RastFromWKB - Return a raster value from a Well-Known Binary (WKB) raster.
- ST_AsBinary/ST_AsWKB - Return the Well-Known Binary (WKB) representation of the raster.
- ST_AsGDALRaster – Return the raster tile in the designated GDAL Raster format.
- ST_AsTIFF - Return the raster selected bands as a single TIFF image (byte array). If no band is specified or any of specified bands does not exist in the raster, then will try to use all bands
- ST_AsPNG - Return the raster tile selected bands as a single portable network graphics (PNG) image (byte array)
- ST_AsJPEG - Return the raster tile selected bands as a single Joint Photographic Exports Group (JPEG) image (byte array)



All ST Functions Supported by PostGIS Raster

Visit the link:

https://postgis.net/docs/RT_reference.html

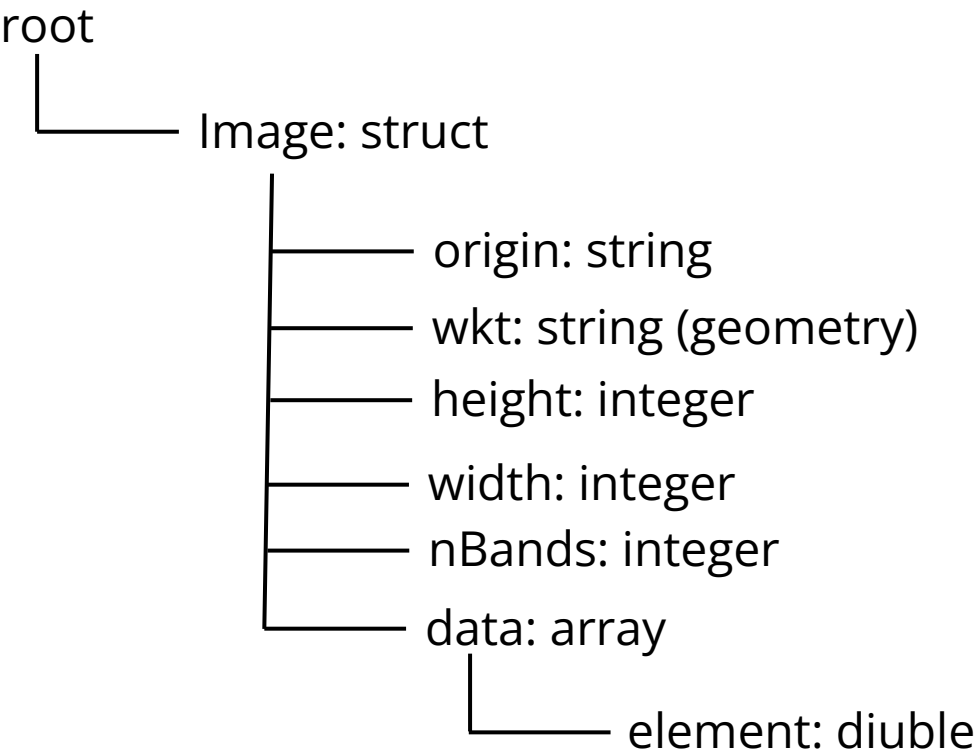


An aerial photograph of a rugged mountainous landscape. A river flows through the center, with a small lake visible in the upper left. The terrain is characterized by steep slopes, ridges, and valleys, with varying shades of green, brown, and blue. The text 'Raster Support in Apache Sedona' is overlaid on the left side of the image.

Raster Support in Apache Sedona

Raster Supports in Apache Sedona

Representation of GeoTIFF Raster Images



Raster Supports in Apache Sedona

Loading GeoTIFF Raster Images

```
geotiffDF = sparkSession.read.format("geotiff").option("dropInvalid", true).load("YOUR_PATH")
```

More Option Parameters During Loading

- readFromCRS - Coordinate reference system of the geometry coordinates representing the location of the Geotiff.
- readToCRS – New CRS if you want to transform the Geotiff location geometry coordinates to a different CRS
- disableErrorInCRS - Indicates whether to ignore errors in CRS transformation (default false).

Raster Supports in Apache Sedona

Writing GeoTIFF Raster Images

```
dfToWrite.coalesce(1).write.mode("overwrite").format("geotiff").save("DESTINATION_PATH")
```

More Option Parameters During Writing

- writeToCRS - Coordinate reference system of the geometry coordinates representing the location of the Geotiff.
- fieldImage - Indicates the image column of GeoTiff DataFrame.
- fieldOrigin - Indicates the origin column of GeoTiff DataFrame.
- fieldNBands - Indicates the nBands column of GeoTiff DataFrame.
- fieldGeometry - Indicates the geometry column of GeoTiff DataFrame.
- fieldData - Indicates the data column of GeoTiff DataFrame.
- fieldWidth/fieldHeight - Indicates the width/height column of GeoTiff DataFrame.

Apache Sedona Raster SQL Functions

- RS_GetBand - Return a particular band from Geotiff Dataframe
- RS_Add- Add two spectral bands in a Geotiff image
- RS_Append - Appends a new band to the end of Geotiff image data and returns the new data
- RS_Normalize – Normalize the value in the array to [0, 255]
- RS_NormalizeDifference – Returns Normalized Difference between two bands(band2 and band1) in a Geotiff image

More details: <https://sedona.apache.org/api/sql/Raster-operators/>

Various Normalized Difference Indices

Normalized Difference Vegetation Index (NDVI)

$$\text{NDVI} = ((\text{NIR} - \text{Red}) / (\text{NIR} + \text{Red}))$$

- NIR = Pixel values from near infrared band
- Red = Pixel values from red band

Normalized Difference Snow Index (NDSI)

$$\text{NDSI} = ((\text{Green} - \text{SWIR}) / (\text{Green} + \text{SWIR}))$$

- Green = Pixel values from green band
- SWIR = Pixel values from shortwave infrared band

Various Normalized Difference Indices

Normalized Burn ratio Index (NBRI)

$$\text{NDVI} = ((\text{NIR} - \text{SWIR}) / (\text{NIR} + \text{SWIR}))$$

- NIR = Pixel values from near infrared band
- SWIR = Pixel values from shortwave infrared band

Normalized Difference Built-up Index (NDBI)

$$\text{NDSI} = ((\text{SWIR} - \text{NIR}) / (\text{SWIR} + \text{NIR}))$$

- SWIR = Pixel values from shortwave infrared band
- NIR = Pixel values from near infrared band