PostGIS 2.0 Raster Cheatsheet

New in this release ¹  Enhanced in this release ²  Requires GEOS 3.3 or higher ³  2.5/3D support ⁴  SQL-MM ⁵  Supports geography ⁶

<table>
<thead>
<tr>
<th>Raster Support Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>geomval</td>
</tr>
<tr>
<td>histogram</td>
</tr>
<tr>
<td>raster</td>
</tr>
<tr>
<td>reclassarg</td>
</tr>
<tr>
<td>summarystats</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Raster Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRasterConstraints ¹</td>
</tr>
</tbody>
</table>

1. rasttable, rastcolumn, srid, scale_x, scale_y, blocksize_x, blocksize_y, same_alignment, regular_blocking, num_bands=true, pixel_types=true, nodata_values=true, out_db=true, extent=true
2. rasttable, rastcolumn, VARIADIC constraints
3. rastschema, rasttable, rastcolumn, VARIADIC constraints
4. rastschema, rasttable, rastcolumn, srid=true, scale_x=true, scale_y=true, blocksize_x=true, blocksize_y=true, same_alignment=true, regular_blocking=true, num_bands=true, pixel_types=true, nodata_values=true, out_db=true, extent=true

| DropRasterConstraints ¹  | Drops PostGIS raster constraints that refer to a raster table column. Useful if you need to reload data or update your raster column data. |

1. rasttable, rastcolumn, srid, scale_x, scale_y, blocksize_x, blocksize_y, same_alignment, regular_blocking, num_bands=true, pixel_types=true, nodata_values=true, out_db=true, extent=true
2. rastschema, rasttable, rastcolumn, srid=true, scale_x=true, scale_y=true, blocksize_x=true, blocksize_y=true, same_alignment=true, regular_blocking=true, num_bands=true, pixel_types=true, nodata_values=true, out_db=true, extent=true
3. rastschema, rasttable, rastcolumn, constraints

| PostGIS_Raster_Lib_Build_Date () | Reports full raster library build date. |
|---------------------------------|

| PostGIS_Raster_Lib_Version () | Reports full raster version and build configuration infos. |

| ST_GDALDrivers ¹  | Returns a list of raster formats supported by your lib gdal. These are the formats you can output your raster using ST_AsGDALRaster. |

<table>
<thead>
<tr>
<th>Raster Constructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_AddBand</td>
</tr>
</tbody>
</table>

1. rast, pixeltype, initialvalue=0, nodataval=NULL
2. rast, index, pixeltype, initialvalue=0, nodataval=NULL
3. torast, fromrast, fromband=1, torastindex=at_end
4. torast, fromrasts, fromband=1

| ST_AsRaster ¹  | Converts a PostGIS geometry to a PostGIS raster. |

1. geom, ref, pixeltype=ARRAY['SBUF'], value=ARRAY[1], nodataval=ARRAY[0], touched=false
2. geom, ref, pixeltype=ARRAY['SBUF'], value=ARRAY[1], nodataval=ARRAY[0], touched=false
3. geom, scalex, scaley, gridx, gridy, pixeltype, value=1, nodataval=0, skewx=0, skewy=0, touched=false
4. geom, scalex, scaley, gridx=NO, gridy=NO, pixeltype=ARRAY['SBUF'], value=ARRAY[1], nodataval=ARRAY[0], skewx=0, skewy=0, touched=false
5. geom, scalex, scaley, pixeltype, value=1, nodataval=0, upperleftx=NO, upperlefty=NO, skewx=0, skewy=0, touched=false
6. geom, scalex, scaley, pixeltype, value=ARRAY[1], nodataval=ARRAY[0], upperleftx=NO, upperlefty=NO, skewx=0, skewy=0, touched=false
7. geom, width, height, gridx, gridy, pixeltype, value=1, nodataval=0, skewx=0, skewy=0, touched=false
8. geom, width, height, gridx=NO, gridy=NO, pixeltype=ARRAY['SBUF'], value=ARRAY[1], nodataval=ARRAY[0], skewx=0, skewy=0, touched=false
9. geom, width, height, pixeltype, value=1, nodataval=0, upperleftx=NO, upperlefty=NO, skewx=0, skewy=0, touched=false
10. geom, width, height, pixeltype, value=ARRAY[1], nodataval=ARRAY[0], upperleftx=NO, upperlefty=NO, skewx=0, skewy=0, touched=false
ST_Band

Returns one or more bands of an existing raster as a new raster. Useful for building new rasters from existing rasters.

1. rast, nbands = ARRAY[1]
2. rast, nbands, delimiter=,
3. rast, nbands

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). If a raster is passed in, returns a new raster with the same size, alignment and SRID. If srid is left out, the spatial ref is set to unknown (0).

1. rast
2. width, height, upperleftx, upperlefty, scalex, scaley, skewx, skewy, srid=unknown
3. width, height, upperleftx, upperlefty, pixelsize

---

### Raster Accessors

**ST_GeoReference** (rast, format=GDAL)  
Returns the georeference meta data in GDAL or ESRI format as commonly seen in a world file. Default is GDAL.

**ST_Height** (rast)  
Returns the height of the raster in pixels.

**ST_MetaData** (rast)  
Returns basic meta data about a raster object such as pixel size, rotation (skew), upper, lower left, etc.

**ST_NumBands** (rast)  
Returns the number of bands in the raster object.

**ST_PixelHeight** (rast)  
Returns the pixel height in geometric units of the spatial reference system.

**ST_PixelWidth** (rast)  
Returns the pixel width in geometric units of the spatial reference system.

**ST_ScaleX** (rast)  
Returns the X component of the pixel width in units of coordinate reference system.

**ST_ScaleY** (rast)  
Returns the Y component of the pixel height in units of coordinate reference system.

**ST_Raster2WorldCoordX**  
Returns the geometric X coordinate upper left of a raster, column and row. Numbering of columns and rows starts at 1.

1. rast, xcolumn
2. rast, xcolumn, yrow

**ST_Raster2WorldCoordY**  
Returns the geometric Y coordinate upper left corner of a raster, column and row. Numbering of columns and rows starts at 1.

1. rast, yrow
2. rast, xcolumn, yrow

**ST_Rotation** (rast)  
Returns the rotation of the raster in radian.

**ST_SkewX** (rast)  
Returns the georeference X skew (or rotation parameter).

**ST_SkewY** (rast)  
Returns the georeference Y skew (or rotation parameter).

**ST_SRID** (rast)  
Returns the spatial reference identifier of the raster as defined in spatial_ref_sys table.

**ST_UpperLeftX** (rast)  
Returns the upper left X coordinate of raster in projected spatial ref.

**ST_UpperLeftY** (rast)  
Returns the upper left Y coordinate of raster in projected spatial ref.

**ST_Width** (rast)  
Returns the width of the raster in pixels.

**ST_World2RasterCoordX**  
Returns the column in the raster of the point geometry (pt) or a X and Y world coordinate (xw, yw) represented in world spatial reference system of raster.

1. rast, pt
2. rast, xw
3. rast, xw, yw

**ST_World2RasterCoordY**  
Returns the row in the raster of the point geometry (pt) or a X and Y world coordinate (xw, yw) represented in world spatial reference system of raster.

1. rast, pt
2. rast, xw
3. rast, xw, yw

**ST_IsEmpty** (rast)  
Returns true if the raster is empty (width = 0 and height = 0). Otherwise, returns false.

---

### Raster Band Accessors

**ST_BandMetaData** (rast, bandnum=1)  
Returns basic meta data for a specific raster band. band num 1 is assumed if none-specified.

**ST_BandNoDataValue** (rast, bandnum=1)  
Returns the value in a given band that represents no data. If no band num 1 is assumed.

**ST_BandIsNoData**  
Returns true if the band is filled with only nodata values.

1. rast, band, forceChecking=true
2. rast, forceChecking=true
### Raster Pixel Accessors and Setters

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_PixelAsPolygon (rast, columnx, rowy)</td>
<td>Returns the geometry that bounds the pixel for a particular row and column.</td>
</tr>
<tr>
<td>ST_PixelAsPolygons (rast, band=1)</td>
<td>Returns the geometry that bounds every pixel of a raster band along with the value, the X and the Y raster coordinates of each pixel.</td>
</tr>
<tr>
<td>ST_Value (rast, pt, exclude_nodata_value=true)</td>
<td>Returns the value of a given band in a given columnx, rowy pixel or at a particular geometric point. Band numbers start at 1 and assumed to be 1 if not specified. If exclude_nodata_value is set to false, then all pixels include nodata pixels are considered to intersect and return value. If exclude_nodata_value is not passed in then reads it from metadata of raster.</td>
</tr>
<tr>
<td>ST_SetValue (rast, pt, newvalue)</td>
<td>Returns modified raster resulting from setting the value of a given band in a given columnx, rowy pixel or at a pixel that intersects a particular geometric point. Band numbers start at 1 and assumed to be 1 if not specified.</td>
</tr>
</tbody>
</table>

### Raster Editors

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_SetGeoReference (rast, georefcoords, format=GDAL)</td>
<td>Set Georeference 6 georeference parameters in a single call. Numbers should be separated by white space. Accepts inputs in GDAL or ESRI format. Default is GDAL.</td>
</tr>
<tr>
<td>ST_SetRotation (rast, rotation)</td>
<td>Set the rotation of the raster in radian.</td>
</tr>
<tr>
<td>ST_SetScale (rast, xy)</td>
<td>Sets the X and Y size of pixels in units of coordinate reference system. Number units/pixel width/height.</td>
</tr>
<tr>
<td>ST_SetSkew (rast, skewxy)</td>
<td>Sets the georeference X and Y skew (or rotation parameter). If only one is passed in, sets X and Y to the same value.</td>
</tr>
<tr>
<td>ST_SetSRID (rast, srid)</td>
<td>Sets the SRID of a raster to a particular integer srid defined in the spatial_ref_sys table.</td>
</tr>
<tr>
<td>ST_SetUpperLeft (rast, x, y)</td>
<td>Sets the value of the upper left corner of the pixel to projected X and Y coordinates.</td>
</tr>
<tr>
<td>ST_Resample (rast, width, height, srid=same_as_rast, gridx=NULL, gridy=NULL, skewx=0, skewy=0, algorithm=NearestNeighbour, maxerr=0.125)</td>
<td>Resample a raster using a specified resampling algorithm, new dimensions, an arbitrary grid corner and a set of raster georeferencing attributes defined or borrowed from another raster. New pixel values are computed using the NearestNeighbour (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbour.</td>
</tr>
<tr>
<td>ST_Rescale (rast, scalexy, algorithm=NearestNeighbour, maxerr=0.125)</td>
<td>Resample a raster by adjusting only its scale (or pixel size). New pixel values are computed using the NearestNeighbour (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbour.</td>
</tr>
<tr>
<td>ST_Reskew (rast, skewxy, algorithm=NearestNeighbour, maxerr=0.125)</td>
<td>Resample a raster by adjusting only its skew (or rotation parameters). New pixel values are computed using the NearestNeighbour (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbour.</td>
</tr>
<tr>
<td>ST_SnapToGrid (rast, gridx, gridy, scalex=DEFAULT 0, scaley=DEFAULT 0, algorithm=NearestNeighbour, maxerr=0.125)</td>
<td>Resample a raster by snapping it to a grid. New pixel values are computed using the NearestNeighbour (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbour.</td>
</tr>
</tbody>
</table>
### ST_Transform
Reprojects a raster in a known spatial reference system to another known spatial reference system using specified resampling algorithm. Options are NearestNeighbor, Bilinear, Cubic, CubicSpline, Lanczos defaulting to NearestNeighbor.

1. `rast, srid, scalex, scaley, algorithm=NearestNeighbor, maxerr=0.125`
2. `rast, srid, algorithm=NearestNeighbor, maxerr=0.125, scalex, scaley`

### Raster Band Editors

#### ST_SetBandNoDataValue
Sets the value for the given band that represents no data. Band 1 is assumed if no band is specified. To mark a band as having no nodata value, set the nodata value = NULL.

1. `rast, nodatavalue`
2. `rast, band, nodatavalue, forcechecking=false`

#### ST_SetBandIsNoData
Sets the isnodata flag of the band to TRUE.

### Raster Band Statistics and Analytics

#### ST_Count
Returns the number of pixels in a given band of a raster or raster coverage. If no band is specified defaults to band 1. If exclude_nodata_value is set to true, will only count pixels that are not equal to the nodata value.

1. `rast, nbond=1, exclude_nodata_value=true`
2. `rast, exclude_nodata_value`
3. `raster, rastercolumn, nbond=1, exclude_nodata_value=true`
4. `raster, rastercolumn, exclude_nodata_value`

#### ST_Histogram
Returns a set of histogram summarizing a raster or raster coverage data distribution separate bin ranges. Number of bins are autocomputed if not specified.

1. `rast, nbond=1, exclude_nodata_value=true, bins=autocomputed, width=NULL, right=false`
2. `rast, nbond, bins, right`
3. `rast, nbond, exclude_nodata_value, bins, right`
4. `rast, nbond, bins, right`
5. `raster, rastercolumn, nbond, bins, right`
6. `raster, rastercolumn, nbond, exclude_nodata_value, bins, right`
7. `raster, rastercolumn, nbond=1, exclude_nodata_value=true, bins=autocomputed, width=NULL, right=false`
8. `raster, rastercolumn, nbond=1, bins, width=NULL, right=false`

#### ST_Quantile
Compute quantiles for a raster or raster table coverage in the context of the sample or population. Thus, a value could be examined to be at the raster's 25%, 50%, 75% percentile.

1. `rast, nbond=1, exclude_nodata_value=true, quantiles=NULL`
2. `rast, quantiles`
3. `rast, nbond, quantiles`
4. `rast, quantile`
5. `rast, exclude_nodata_value, quantile=NULL`
6. `rast, nbond, quantile`
7. `rast, nbond, exclude_nodata_value, quantile`
8. `rast, nbond, quantile`
9. `raster, rastercolumn, nbond=1, exclude_nodata_value=true, quantiles=NULL`
10. `raster, rastercolumn, nbond, quantiles`

#### ST_SummaryStats
Returns summary stats consisting of count,sum,mean,stddev,min,max for a given raster band of a raster or raster coverage. Band 1 is assumed is no band is specified.

1. `raster, rastercolumn, exclude_nodata_value`
2. `rast, exclude_nodata_value`
3. `raster, rastercolumn, nbond=1, exclude_nodata_value=true`
4. `rast, nbond, exclude_nodata_value`

#### ST_ValueCount
Returns a set of records containing a pixel band value and count of the number of pixels in a given band of a raster (or a raster coverage) that have a given set of values. If no band is specified defaults to band 1. By default nodata value pixels are not counted. and all other values in the pixel are output and pixel band values are rounded to the nearest integer.

1. `rast, nbond=1, exclude_nodata_value=true, searchvalues=NULL, roundto=0`
2. `rast, nbond, searchvalues, roundto=0`
3. `rast, searchvalues, roundto=0`
4. `rast, searchvalue, roundto=0`
5. `rast, nbond, exclude_nodata_value, searchvalue, roundto=0`
6. `rast, nbond, searchvalue, roundto=0`
7. `raster, rastercolumn, nbond=1, exclude_nodata_value=true, searchvalues=NULL, roundto=0`
8. `raster, rastercolumn, searchvalues, roundto=0`
9. `raster, rastercolumn, nbond, searchvalues, roundto=0`
<table>
<thead>
<tr>
<th>Raster Outputs</th>
<th>Raster Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ST_AsBinary</strong> (rast)</td>
<td><strong>Box3D</strong> (rast)</td>
</tr>
<tr>
<td><strong>ST_AsGDALRaster</strong>¹ (rast, format, options=NULL, srid=sameassource)</td>
<td><strong>ST_Clip</strong>¹</td>
</tr>
<tr>
<td><strong>ST_AsJPEG</strong>¹</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ST_AsPNG</strong>¹</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ST_AsTIFF</strong>¹</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

¹ PostGIS 2.0 Raster Cheat Sheet v. 2.0.2 | | PostGIS 2.0 Raster Cheat Sheet v. 2.0.2 | PostGIS 2.0 Raster Cheat Sheet v. 2.0.2
### ST_MapAlgebraExpr
1. raster band version: Creates a new one band raster formed by applying a valid PostgreSQL algebraic operation on the input raster band and of pixeltype provided. Band 1 is assumed if no band is specified.

   1. \( \text{rast, band, pixeltype, expression, nodataval=\text{NULL}} \)
   2. \( \text{rast, pixeltype, expression, nodataval=\text{NULL}} \)

2. raster band version: Creates a new one band raster formed by applying a valid PostgreSQL algebraic operation on the two input raster bands and of pixeltype provided. Band 1 of each raster is assumed if no band numbers are specified. The resulting raster will be aligned (scale, skew and pixel corners) on the grid defined by the first raster and have its extent defined by the "extenttype" parameter. Values for "extenttype" can be: INTERSECTION, UNION, FIRST, SECOND.

   1. \( \text{rast1, rast2, expression, pixeltype=\text{same as rast1 band}, extenttype=\text{INTERSECTION}, nodataval=\text{NULL, rast2expr=\text{NULL, nodatanodataval=\text{NULL}}} \)
   2. \( \text{rast1, band1, rast2, band2, expression, pixeltype=\text{same as rast1 band}, extenttype=\text{INTERSECTION}, nodataval=\text{NULL, rast2expr=\text{NULL, nodatanodataval=\text{NULL}}} \)

### ST_MapAlgebraFct
1. band version - Creates a new one band raster formed by applying a valid PostgreSQL function on the input raster band and of pixeltype provided. Band 1 is assumed if no band is specified.

   1. \( \text{rast, onerasteruserfunc} \)
   2. \( \text{rast, onerasteruserfunc, \text{VARIADIC args}} \)
   3. \( \text{rast, pixeltype, onerasteruserfunc} \)
   4. \( \text{rast, pixeltype, onerasteruserfunc, \text{VARIADIC args}} \)
   5. \( \text{rast, band, onerasteruserfunc} \)
   6. \( \text{rast, band, onerasteruserfunc, \text{VARIADIC args}} \)
   7. \( \text{rast, band, pixeltype, onerasteruserfunc} \)
   8. \( \text{rast, band, pixeltype, onerasteruserfunc, \text{VARIADIC args}} \)

2. band version - Creates a new one band raster formed by applying a valid PostgreSQL function on the 2 input raster bands and of pixeltype provided. Band 1 is assumed if no band is specified. Extent type defaults to INTERSECTION if not specified.

   1. \( \text{rast1, rast2, tworastuserfunc, pixeltype=\text{same as rast1}, extenttype=\text{INTERSECTION, VARIADIC userargs}} \)
   2. \( \text{rast1, band1, rast2, band2, tworastuserfunc, pixeltype=\text{same as rast1 band}, extenttype=\text{INTERSECTION, VARIADIC userargs}} \)

### ST_MapAlgebraFctNgb
(rast, band, pixeltype, ngbwidth, ngbheight, onerastngbuserfunc, nodatamode, \text{VARIADIC args}) 1-band version: Map Algebra Nearest Neighbor using user-defined PostgreSQL function. Return a raster which values are the result of a PLPGSQL user function involving a neighborhood of values from the input raster band.

### ST_Polygon
(rast, \text{band_num=1}) Returns a polygon geometry formed by the union of pixels that have a pixel value that is not no data value. If no band number is specified, band num defaults to 1.

### ST_Reclass
Creates a new raster composed of band types reclassified from original. The nb band is the band to be changed. If nb band is not specified assumed to be 1. All other bands are returned unchanged. Use case: convert a 16BUI band to a 8BUI and so forth for simpler rendering as viewable formats.

   1. \( \text{rast, nb band, reclassexpr, pixeltype, nodataval=\text{NULL}} \)
   2. \( \text{rast, VARIADIC reclassargsset} \)
   3. \( \text{rast, reclassexpr, pixeltype} \)

### ST_Union
Returns the union of a set of raster tiles into a single raster composed of 1 band. If no band is specified for unioning, band num 1 is assumed. The resulting raster's extent is the extent of the whole set. In the case of intersection, the resulting value is defined by \text{p_expression} which is one of the following: LAST - the default when none is specified, MEAN, SUM, FIRST, MAX, MIN.

   1. \( \text{rast} \)
   2. \( \text{rast, band_num} \)
   3. \( \text{rast, p_expression} \)
   4. \( \text{rast, band_num, p_expression} \)

### Raster Processing Builtin Functions

#### ST_Min4ma
(matrix, nodatamode, \text{VARIADIC args}) Raster processing function that calculates the minimum pixel value in a neighborhood.

#### ST_Max4ma
(matrix, nodatamode, \text{VARIADIC args}) Raster processing function that calculates the maximum pixel value in a neighborhood.

#### ST_Sum4ma
(matrix, nodatamode, \text{VARIADIC args}) Raster processing function that calculates the sum of all pixel values in a neighborhood.

#### ST_Mean4ma
(matrix, nodatamode, \text{VARIADIC args}) Raster processing function that calculates the mean pixel value in a neighborhood.

#### ST_Range4ma
(matrix, nodatamode, \text{VARIADIC args}) Raster processing function that calculates the range of pixel values in a neighborhood.

#### ST_Distinct4ma
(matrix, nodatamode, \text{VARIADIC args}) Raster processing function that calculates the number of unique pixel values in a neighborhood.
### Raster Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;&amp; (A, B)</code></td>
<td>Returns TRUE if A's bounding box overlaps B's.</td>
</tr>
<tr>
<td><code>&amp;&lt; (A, B)</code></td>
<td>Returns TRUE if A's bounding box is to the left of B's.</td>
</tr>
<tr>
<td><code>&amp;&gt; (A, B)</code></td>
<td>Returns TRUE if A's bounding box is to the right of B's.</td>
</tr>
</tbody>
</table>

### Raster and Raster Band Spatial Relationships

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ST_Intersects</code></td>
<td>Return true if the raster spatially intersects a separate raster or geometry. If the band number is not provided (or set to NULL), only the convex hull of the raster is considered in the test. If the band number is provided, only those pixels with value (not NODATA) are considered in the test.</td>
</tr>
<tr>
<td><code>ST_SameAlignment</code></td>
<td>Returns true if rasters have same skew, scale, spatial ref and false if they don't with notice detailing issue.</td>
</tr>
</tbody>
</table>

---

This work is licensed under a Creative Commons Attribution .

Feel free to use this material for private or commercial purposes, but we ask that you please retain the http://www.postgis.us PostGIS in Action website link.