ORACLE

GeoRaster in Oracle Database

March, 2021

Public

ORACLE

Purpose statement

This document provides an overview of GeoRaster feature included with Oracle Database. It is intended solely to help you assess the business benefits of using this feature and to plan your I.T. projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

Introduction

GeoRaster is a feature of Oracle Database that lets you store, index, query, process, analyze, and serve georeferenced raster image and gridded data and its associated metadata. It provides native data types and an object-relational schema to store and manage multidimensional array, grid layers and digital images that can be referenced to positions on the Earth's surface or in a local coordinate system.

What differentiates GeoRaster is the ability to perform raster analysis on extremely large images and data sets, provide in-place image processing and analysis with no development required, and provide parallelized image processing with simple invocation of PL/SQL procedures.

GeoRaster is used with data from any technology that captures or generates raster data and images, such as remote sensing, photogrammetry, and geospatial thematic mapping. It is used in a wide variety of application areas, including location-based services, geospatial image archiving and serving, environmental monitoring and assessment, geological engineering and exploration, natural resource management, climate modeling, ocean floor surveying and mapping, defense, emergency response, telecommunications, transportation, urban planning, and homeland security.

GeoRaster is designed to deliver enterprise-class data management capability to large image processing and GIS solutions and business applications. Developers can integrate this powerful data management technology with the leading image processing and raster/grid analysis tools and various business applications.

It meets the data management needs of broad application groups including:

Remote sensing, photogrammetry, GIS and geosciences applications – users manage and process their geospatial raster and gridded data assets using a scalable, secure, and robust RDBMS for defense, intelligence, agriculture, environment and natural resource management.

Business applications – leverage raster-based data in conjunction with other basic location data (address, etc.) to evaluate site locations and track fixed and/or continuous assets. These include Asset Management and Facilities Management, change analysis, site selection, suitability analysis and insurance risk assessment.

Image and Gridded Raster Data Repositories/Clearinghouses – support for clearinghouse servers that need to ingest, store, process, analyze, and disseminate very large volumes of geoimagery and raster grids, through intranet or internet.

Architecture

GeoRaster architecture provides the functionality needed to support the storage and use of image or grid-based raster data in Oracle Database. At a very high level of abstraction, GeoRaster architecture includes five basic components:

GeoRaster Engine – This is the core, which provides the native GeoRaster object type and functionality including raster data and metadata indexing, update, query and manipulations.

PL/SQL API – Standard SQL access to the raster and grid-based data in GeoRaster databases.

Java API – Pure Java access to GeoRaster objects in the databases. Users can also use JDBC, OCI, and OCCI to access GeoRaster objects without calling GeoRaster PL/SQL or Java APIs.

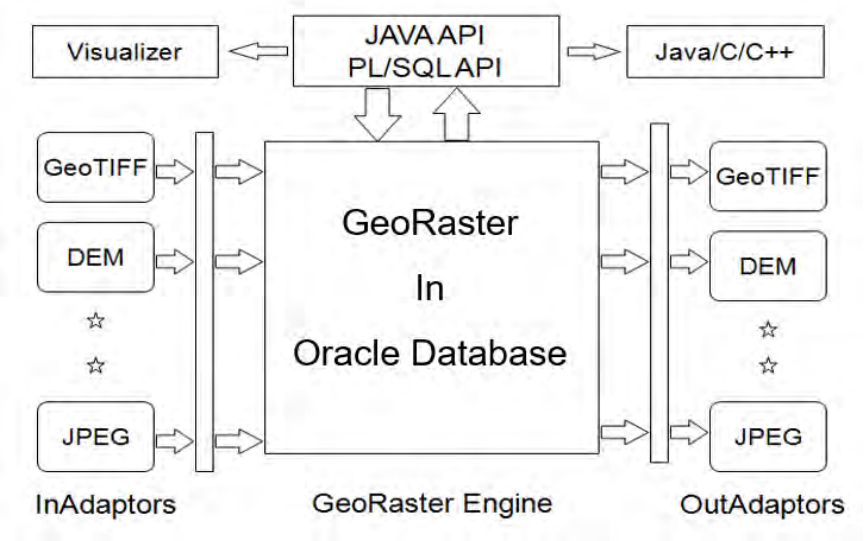

Figure 1. Simplified Architecture of GeoRaster

Viewing Tools and Publishing: Oracle Database map visualization component and Spatial Studio support GeoRaster. OGC WMS and WCS web services are provided for publishing GeoRaster data. A standalone viewer comes with the GeoRaster installation and can be used as a development or DBA tool. Finally, a variety of third party visualization and analysis tools support GeoRaster.

Input and Output [data] Adapters –GeoRaster includes an ETL tool based on the popular Geospatial Data Abstract Layer (GDAL) for concurrent batch loading and exporting of large numbers of raster files. This facilitates loading and unloading raster data between well-known image file formats and GeoRaster. The GDAL tool supports all raster formats supported by GDAL. GeoRaster also provides limited importing and exporting capability on six standard image file formats through both the server-side PL/SQL API and the client-side Java tool.

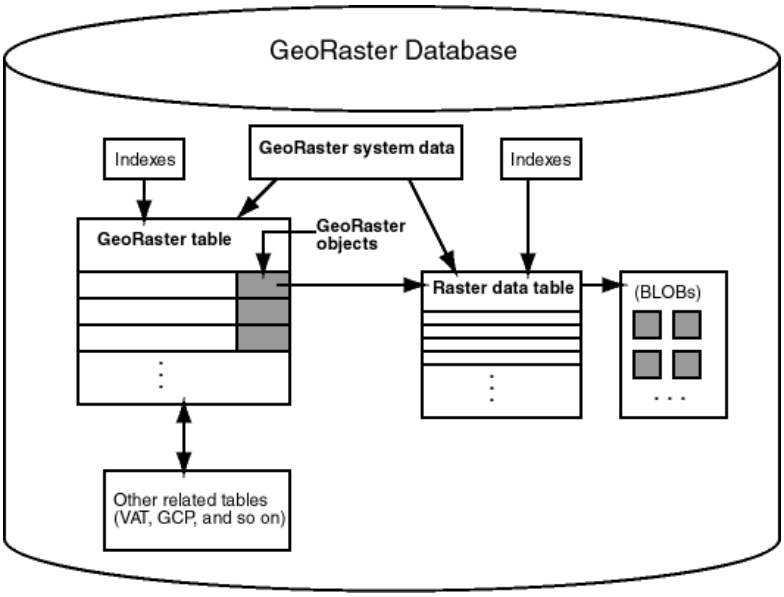

Figure 2. Physical Storage Schema of GeoRaster

The core of GeoRaster is the physical schema designed to facilitate storing and managing raster or grid-based data inside the database. In the GeoRaster engine, the native data type called SDO_GEORASTER is defined and each image or raster grid is stored as a single object of this native type. A GeoRaster table is any user-defined table, which has at least one data column of type SDO_GEORASTER. SDO_GEORASTER objects include metadata and information about how to retrieve GeoRaster cell data that is stored in another user-defined table called a Raster Data Table (RDT), which is an object table of type SDO_RASTER or a relational table with the same columns as the attributes of type SDO_RASTER. The raster data table includes a BLOB column called RASTERBLOCK, which stores the real raster blocks. Other information associated with the GeoRaster objects can be stored in separate columns or tables, such as a Value Attribute Table (VAT). The relationship between a GeoRaster object and its raster data table is automatically managed by GeoRaster using a database dictionary. While users need to create and drop the raster data tables, they are internal to GeoRaster and the raster data in them are automatically updated and processed by the GeoRaster APIs.

A GeoRaster database basically consists of one or a list of GeoRaster tables, in which each image or raster grid is stored as one GeoRaster object in one row. It can contain an unlimited number of GeoRaster objects and each object can be terabytes in size. GeoRaster tables can be in different database schemas and GeoRaster objects can be accessed across schemas.

The specifics of the GeoRaster data model and how this architecture is implemented in Oracle Database are provided in the sections below.

Data Model

There are two basic raster data types supported in GeoRaster: grid-based data and image data.

Grid-based, or gridded, data is a general term used for raster data such as digital terrain elevation, land use and land cover information, pollution concentration, geological information, and rainfall information. It is a rectangular grid of cells that are aligned to the X and Y-axes overlying an area. Each cell in the grid has the same size; this size is the resolution of the grid. Grid data stores the attribute values or attribute index values for each cell in the grid. If the index values are stored, different attribute values can be associated with the index values and are typically stored in a relational table, called Value Attribute Table or Raster Attribute Table.

Digital imagery is a specialized type of raster data. It is a two dimensional array (a matrix or grid) of regularly spaced picture elements (pixels). An image is created from optical or other spectral sensors, and is collected using a variety of technologies including satellite remote sensing and airborne photogrammetry. The size of the pixel is referred to as the resolution of the image. Digital images can be composed of many bands, referred to as multispectral or hyperspectral. Raster Attribute Tables can be associated with the image or its bands.

GeoRaster defines an integrated raster data model for these data types. Conceptually, it is component-based, logically layered, and multidimensional. The core data in a raster is a multidimensional array or matrix of raster cells. Each cell is one element of the matrix, and its value is called the cell value. The matrix has a number of dimensions, a cell depth, and a size for each dimension. The cell depth is the data size of the value of each cell and it applies to all cells. This core raster data set can be blocked for optimal storage, retrieval and processing. Pyramids (generalized, lower-resolution versions of the image – useful for fast retrieval in web applications) of the core raster data can be generated, stored and processed the same way.

Raster data is logically layered. The whole raster data is called the object layer or layer 0, and consists of one or more logical layers (or sublayers). For example, for multi-channel remote sensing imagery, the sublayers are used to model the channels or bands of the imagery. In GeoRaster, each sublayer is a two-dimensional matrix of cells that consists of the row dimension and the column dimension.

Besides the core matrix of cells, a raster data object has specific metadata associated with it. In GeoRaster, the metadata are organized into components that contain the following information:

- Object information
- Raster information
- Spatial reference system information
- Date and time (temporal reference system) information
- Spectral (band reference system) information
- Layer information for each layer

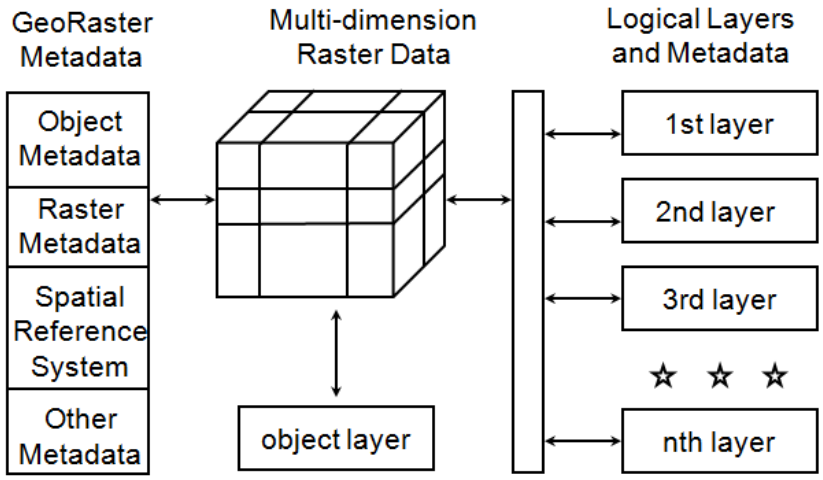

Figure 3. Conceptual Data Model of GeoRaster Objects

Object Information includes metadata such as user-defined ID, description and version information. Raster Information includes metadata such as cell depth (e.g., 1BIT, 32BIT_S, or 64BIT_REAL), dimensionality, blocking sizes, interleaving types, compression and information about pyramids. Spatial Reference System metadata contains information required for georeferencing, in which a generic polynomial model is defined and ground control points (GCPs) can be stored.

Layer Information contains metadata pertaining to each logical layer in a GeoRaster object and it consists of many subcomponents. Major subcomponents of Layer Information metadata include user-defined layer ID, scaling function, bin function, colormap, grayscale lookup table, statistics and histogram, NODATA values and value ranges, and bitmap masks. A Value Attribute Table can be used to maintain information about the values stored in each layer and the table name can be registered in the Layer Information metadata. Each layer, including object layer, can have one bitmap mask associated with it, which is registered in the Layer Information metadata. As special metadata, the bitmap masks are special raster grids with a cell depth of 1. Physical bitmap masks are stored together with the core raster data in its RDT.

Except for Object Information and Raster Information, all other metadata components are optional. This enables GeoRaster to store, manage and manipulate generic multidimensional array data types as well.

GeoRaster Object

Physically, the GeoRaster data model is embodied as two native RDBMS data types: SDO_GEORASTER and SDO_RASTER. SDO_GEORASTER data type encapsulates and stores all information about a raster data and is the external interface for users to create and manage a GeoRaster database. The SDO_RASTER data type is internal and optional to the SDO_GEORASTER data type and users don't need to directly manipulate it.

One raster data (an image or a grid) is stored in Oracle as an object of SDO_GEORASTER data type, which is defined as:

```
CREATE TYPE sdo_georaster AS OBJECT (
    rasterType NUMBER,
    spatialExtent SDO_GEOMETRY,
    rasterDataTable VARCHAR2(32),
    rasterID NUMBER,
    metadata XMLType);
```

GeoRaster metadata is stored as the metadata attribute of the SDO_GEORASTER type. It is an XML document using the Oracle XMLType data type. Metadata is stored according to the GeoRaster metadata XML schema defined by GeoRaster. The spatial extent (footprint) of a GeoRaster object is part of the metadata, but it is stored separately as an attribute of the GeoRaster object. This approach allows GeoRaster to take advantage of the spatial geometry type and related capabilities, such as using spatial R-tree indexing on GeoRaster objects and building huge global imagery databases. Another attribute of the SDO_GEORASTER type is the rasterType, which contains dimensionality information and the data type that can be extended.

Actual raster cell data is internally blocked into small subsets for large-scale GeoRaster object storage and optimal retrieval and processing. All blocks are stored in a relational table or an object table of type SDO_RASTER, which is defined as follows:

```
CREATE TYPE sdo_raster AS OBJECT (
    rasterID NUMBER,
    pyramidLevel NUMBER,
    bandBlockNumber NUMBER,
    rowBlockNumber NUMBER,
    columnBlockNumber NUMBER,
    blockMBR SDO_GEOMETRY,
    rasterBlock BLOB);
```

This relational or object table is called raster data table, or simply RDT table. If the RDT is created as a relational table, it must have the same columns as the attributes of type SDO_RASTER. Each block is stored in the RDT table as a binary large object (BLOB). Each row of the table stores only one block and the blocking information related to that block.

Both pyramid and bitmap masks of the GeoRaster object are stored in the same raster data table of the GeoRaster object, using the same blocking scheme. Pyramids can also be generated on the bitmap masks and stored in the same way as the pyramid of the GeoRaster object.

A blank GeoRaster object is a special type of GeoRaster object in which all cells have the same value. There is no need to store its cells in any SDO_RASTER block; instead, the cell value is registered in the metadata in the blankCellValue element. GeoRaster also supports empty raster blocks to save storage space with

large mosaic objects and to improve raster processing speed. It is used when there is no raster data available for a specific raster block of a large GeoRaster object. Such GeoRaster data can be considered as a special sparse data type. There is still an entry in the raster data table for each empty raster block, but the length of the BLOB is zero, i.e., empty. Pyramids and bitmap masks can have empty blocks too.

The pair of rasterDataTable and rasterID attributes of the SDO_GEORASTER object type uniquely identify a GeoRaster object in the whole database. Together, they provide the information internally required to store and retrieve the raster cell data in its raster data table. This design also allows many smaller GeoRaster objects share one RDT table while one huge GeoRaster object can have one devoted RDT table to improve performance and scalability. Users can have an unlimited number of GeoRaster objects stored in a GeoRaster table. Internally, GeoRaster uses a system dictionary (called the GeoRaster sysdata table) to maintain the relationship between GeoRaster objects and their related raster data tables. Even though stored in a separate RDT table, the raster cell data of a GeoRaster object is handled automatically by the GeoRaster functions, which are described in the next section.

Features of GeoRaster

In addition to providing both a logical model and a physical model, GeoRaster provides the PL/SQL API and Java API with a rich set of foundation functions that facilitate data management for raster information in Oracle Database. This section provides a general overview of the basic functional infrastructure available in GeoRaster.

GeoRaster operations can be grouped into the following categories:

- Database Creation
- Database Administration
- Basic Data Manipulation
- Raster Algebra and Analytics
- Image Processing and Serving

Database Creation

From a user perspective, a GeoRaster database is basically an Oracle database containing one or a list of GeoRaster tables, in which each image or raster grid is stored as a GeoRaster object in a row of a GeoRaster column. It can contain a virtually unlimited number of GeoRaster objects in one or more database schemas. One GeoRaster table can contain an unlimited number of GeoRaster objects and a single GeoRaster object can have only one pixel or be terabytes in size.

To build a GeoRaster database, users use standard SQL statements or PL/SQL language and leverage the GeoRaster API and/or third party tools and solutions. The key steps include:

- Create GeoRaster tables and RDT tables using the standard DDL.

- Load raster data and metadata from files in various raster formats into the GeoRaster tables using third-party ETL tools, GDAL-based ETL tool, SDO_GEOR.importFrom procedure, or client-side Java loader. The SDO_GEOR.importFrom procedure and client-side Java loader are lightweight tools for conveniently loading and exporting a limited number of image and raster files one at a time. GeoRaster GDAL-based ETL tool is a powerful tool to load and export large numbers of raster and image files concurrently in batches, and it supports all raster formats supported by GDAL. GDAL-based ETL and Java loader are also integrated into the GeoRaster Viewer.
- Generate and update the spatial extent on each GeoRaster object if necessary. Spatial extent is of SDO_GEOMETRY type.
- To validate GeoRaster objects, call SDO_GEOR.validateGeoRaster. To validate the metadata, call SDO_GEOR.schemaValidate.
- Build appropriate indexes on various columns of GeoRaster tables e.g., a spatial R-tree index on the spatial extent attribute of GeoRaster column and B-tree indexes on other columns. Users can also build function-based indexes on the metadata attribute of GeoRaster objects.
- GeoRaster objects can be updated and deleted using standard SQL statements. They can be copied from one place to another using the SDO_GEOR.copy procedure.
- To export GeoRaster objects into files in various raster formats, use the GDAL-based ETL tool, SDO_GEOR.exportTo procedure, client-side Java exporter, or third-party ETL tools. Similar to loading, these tools are used to massively export large volume of raster data very quickly.

Once the database is created and the data are loaded into GeoRaster, users can manage it, tune it, perform spatial queries and various advanced processing, as described in the next sections.

Database Administration

GeoRaster object type is a native Oracle data type. This approach allows users to manage GeoRaster database using most standard RDBMS features, such as backup and recovery, partitioning, table security, replication, standby, and multitenency. GeoRaster supports Oracle Workspace Manager for raster block versioning and Oracle Label Security for row-level (raster block) security. In addition, GeoRaster provides the SDO_GEOR_ADMIN package, which includes more than 10 functions, to help manage and maintain GeoRaster databases. For example,

- listGeoRasterObjects, listGeoRasterColumns, listGeoRasterTables, listRDT subprograms in the SDO_GEOR_ADMIN package help check the status of existing GeoRaster objects and related objects in the current schema or the database.
- SDO_GEOR_UTL.renameRDT renames the RDT in the database to solve conflicts, which might happen during data migration.

- SDO_GEOR_ADMIN.maintainSysdataEntries automatically maintains the SYSDATA entries in the current schema or the database.
- SDO_GEOR_ADMIN.upgradeGeoRaster checks for and corrects errors after database upgrading.
- The standalone GeoRaster Viewer is a DBA and development tool. It can connect to multiple databases or database schemas and display GeoRaster objects. It allows zoom in and zoom out, scrolling through the raster at any pyramid level and can query cell values and coordinates. It provides basic image processing operators and can view bitmap masks.

Basic Data Manipulation

Besides leveraging the standard enterprise database features, GeoRaster provides over 200 raster data and metadata operations through the PL/SQL API to optimally manage and manipulate GeoRaster objects in support of various application requirements. Users can achieve many data management and manipulation goals using the following list of basic operations provided by GeoRaster:

- Adjust the internal raster blocking size of any GeoRaster object to improve scalability, optimize space usage and speed up raster processing and query. While each block must have the same size, the raster blocking sizes along different dimensions can be random numbers, not necessarily a power of 2.
- Change the band interleaving types among BSQ, BIL and BIP to best fit into different applications.
- Change the cell depth. Cell depths of 1-bit to 32-bit integers and 32-bit and 64-bit real numbers are supported.
- Generate pyramids using different resampling approaches and remove pyramids. GeoRaster supports resampling methods of nearest neighbor, bilinear interpolation, biquadratic interpolation, cubic convolution, and average using four or six neighboring cells. The pyramiding operation supports parallelism to improve performance dramatically.
- Compress or decompress GeoRaster objects natively in lossless DEFLATE, lossy JPEG, lossy or lossless JPEG 2000 compression types. Oracle Database Advanced LOB Compression can be used as well. Most GeoRaster functions that can be performed on uncompressed (decompressed) GeoRaster objects can be performed directly on compressed objects.
- Georeference GeoRaster objects using PL/SQL functions or load georeferencing information from files. A generic polynomial georeferencing model is supported for georeferencing rectified and unrectified airborne photos and satellite images. It supports up to a power of 5 and 3-D model coordinates. The affine transformation, DLT, RPC, and other models are special types of this generic model. GCPs can be stored and used to georeference GeoRaster objects. The following commonly used geometric models are supported with GCP georeferencing: Affine, Quadratic Polynomial, Cubic Polynomial, DLT, Quadratic Rational, and RPC.

- Add or remove bitmap masks for GeoRaster objects and individual bands or layers. These bitmap masks are stored inside the GeoRaster objects. Pyramids of masks can be created and stored inside the GeoRaster objects as well. Masks are compressed when the GeoRaster object is compressed.
- Add or delete NODATA. Multiple NODATA values and multiple NODATA value ranges are supported for GeoRaster objects and their individual bands or layers.
- Add or delete scaling function, bin function, colormap and grayscale information.
- Spatially search GeoRaster objects using spatial index and operators, such as area-of-interest queries, spatial join queries, and other topology-based spatial operations.
- Crop images and perform subsetting to create new GeoRaster objects for persistent storage in the database or as an answer to a specific query for web distribution and display. Clipping the query or subset result along the polygon (irregular) boundary is supported. Bitmap masks can be used as a cropping windows as well.
- Query cell coordinates and do coordinate transformation between GeoRaster cell space and model space. GeoRaster supports sub-cell or sub-pixel addressing (floating row and column numbers) in the GeoRaster cell spaces.
- Query cell values or evaluate a direct location based on neighboring cell values by using a specified interpolation method, and returns the raster values for the specified bands or layers for that location.
- Partially edit and update a window of raster data and its pyramids using another image or gridded data. Change the value of a single cell. The upper pyramid levels in the updated area can be regenerated partially to reflect the updated areas only.
- Analyze statistics and generate histograms on the whole object or individual layers, both persistently or on-the-fly. Both polygon and bitmap based windows can be used.
- Query, delete and update most other items of the metadata through tens of functions or procedures. GeoRaster cell data and metadata update and query are crucial to successful use of GeoRaster in Oracle Database. Some key subprograms include updating version information, querying the user-defined ID, dimension sizes and blocking sizes, checking the coordinate system and georeferencing information, querying and updating time information, to name just a few.

Raster Algebra and Analytics

GeoRaster provides a PL/SQL-based raster algebra engine enabling fast cell data searching, raster data analysis, and cartographic modeling. GeoRaster raster algebra language is an extension to the PL/SQL language, which includes specific algebraic expressions and functions. Algebraic expressions support general arithmetic, logical, conditional, relational, and statistical operations. Raster algebra functions are mostly of local function type and include four major

procedures to support arithmetic operation, conditional query, classify and cell value-based update. With this in-database raster algebra engine, user can do the following major operations.

- Perform complex arithmetic operations.
- Search cells based on logical and relational conditions.
- Apply arithmetic operations to cells and then segment the resulting raster.
- Edit or update cells of a raster based on conditions.
- Generate statistics, including min, max, mean, median, mode, sum, and standard deviation, on a layer or stack basis.
- Perform complex logical operations and analysis.
- Scale, offset and cast raster data.
- Run cartographic models or other analytics.

These functions take one or many layers from one or many GeoRaster objects, apply raster algebra expressions over those layers, do the specific algebraic computation or modeling, and output a new GeoRaster object or statistic results. Expressions can be defined in any way based on the syntax of the expression language. PL/SQL language is used to declare variables and constants, organize statements, and program raster algebra processes and models. AOI and cropping are supported.

In addition, all raster algebra functions support parallelism to improve performance further.

Image Processing and Serving

GeoRaster provides some advanced image processing and serving capabilities. These include GCP georeferencing, reprojection, rectification, orthorectification, warping, affine transformation, image scaling, stretching, masking, filtering, image segmentation, NDVI computation, Tasseled Cap Transformation, image appending, bands merging, large-scale advanced image mosaicking, and virtual mosaic. Operations described here are most commonly used to process and serve geospatial images, particularly raw satellite imagery and airborne photographs. However, these operations, just like the GeoRaster raster algebra, apply to all raster data types.

- Enlarge or shrink images using nearest neighbor, bilinear interpolation, biquadratic interpolation, cubic convolution, or average resampling.
- Reproject (transform) a GeoRaster object to a different projection or coordinate system, either on-the-fly for serving or persistently for storage.
- Rectify 2-D georeferenced raw images.
- Orthorectify 3-D georeferenced raw images with either an average height or a DEM.
- Warp or affine transform images
- Mask images physically using bitmap masks, in addition to storing masks as pure metadata.

- Enhance image colors by image stretching, normalization, equalization, histogram matching, and dodging.
- Filter images using standard or customized filters.
- Segment images using raster algebra.
- Compute NDVI (Normalized Difference Vegetation Index) and other VI's from images.
- Apply TCT (Tasseled Cap Transformation) on images to analyze physical ground features of soil brightness, vegetation greenness, soil and canopy wetness, etc.
- Merge or union layers from GeoRaster objects. Extract a single layer or a subset of layers from a GeoRaster object.
- Append smaller images to a large image without spatial boundary limitation. Smaller images can be rectified, un-rectified, or have different resolutions. They can be located on any side of the target image.
- Mosaick large raster datasets inside the database. GeoRaster mosaic function allows gaps, overlaps, and missing source GeoRaster objects. It supports both rectified and un-rectified images. It supports internal reprojection/rectification and common point rules. It supports color balancing, including histogram matching, by automatically identifying and using overlapping areas. It is also implemented with parallelism. This mosaicking process results in a single GeoRaster object, which is also called a physical mosaic.
- Support Virtual Mosaic, which is any large collection of georeferenced GeoRaster objects, rectified or un-rectified, from one or more GeoRaster tables or views that is treated as if it is a single GeoRaster object. A virtual mosaic is defined as one or a list of GeoRaster tables, a database view with a GeoRaster column, or any SQL query statement that results in a collection of GeoRaster objects. Pyramids of virtual mosaic are supported. Users issue a single call to query the virtual mosaic based on area-of-interest (i.e., subsetting or cropping) and request the cropped images to be in different coordinate system with different resolutions. On-the-fly transformation with resampling and mosaicking with common point rules and color balancing are done internally and automatically.
- Image and raster data serving to clients or applications are supported through many functions in the PL/SQL API and Java API described in this paper. These include direct exporting to files, searching and then subsetting or cropping rasters, and applying reprojection and rectification on-the-fly. Particularly, when the user doesn't want to create large physical mosaics, virtual mosaic is best used in serving image database out to various applications. It doesn't require the source images to be preprocessed or mosaicked beforehand. Instead, all images are stored as is, and the whole image data set can be served based on small areas-of-interest using single calls to the server.

- Image and raster data can also be served over the web using standard OGC web services, i.e., WMS and WCS.

PL/SQL API and Java API are provided for database creation, database administration, data manipulation, raster analysis, image processing, and raster data serving and delivery. They also aid in the development of and integration with applications on top of GeoRaster. Users can use Java, C or C++ to leverage PL/SQL API and Java API, or directly access binary data of the open GeoRaster data model. In addition, WMS and WCS are provided to serve image and raster data over the web. Further details can be found in the Oracle Spatial GeoRaster Developer's Guide.

Benefits of Managing Raster Data in Oracle Database

This section describes some of the major benefits of managing geospatial imagery and raster data using GeoRaster.

- GeoRaster data model is an open and integrated raster data model. It supports various image and raster data types and removes the need of dealing with many data models and significantly simplifies the data management tasks.
- GeoRaster data type is designed for raster data storage inside the database and is independent from any file formats. Raster files in different file formats are loaded into GeoRaster for efficient storage and management. All metadata and cell data of GeoRaster objects are open to users and can be accessed at bit level as long as the users are granted access privilege. This enables shared third party application developments and avoids vendor lock-ins.
- It breaks the size barriers. A virtually unlimited number of GeoRaster objects can be stored and a single GeoRaster object can be multi-terabytes in size. Internal tuning tools and compressions can be used to optimize the raster blocking and other storage options to best fit into the needs and performance requirements of different applications.
- Native data type approach provides flexibility to store raster data in regular relational tables. Images or rasters in different projections and locations can be stored in the same table and users can easily build global databases with whole-earth spatial index and other indices enabling fast queries and manipulations of rasters anywhere on the Globe.
- SQL access provides better management of raster data. The standard SQL or PL/SQL API is flexible, powerful, and easy-to-use. It helps speed up enterprise integrations and broadens geospatial application areas.
- Database-centric approach and balanced memory management allows thousands of concurrent queries and users. Concurrency can scale-up drastically by leveraging Oracle Enterprise GRID computing technologies or using computer clusters.
- High performance computing is made possible for raster data management and processing. GeoRaster features in-database processing, optimized multi-tasking, and parallel processing. In-database data manipulations,

raster algebra and image processing capabilities allow data to be processed where it is stored. Coupled with multi-tasking and parallel processing, this provides great performance and true security. Traditional image processing and GIS systems can leverage it to pre-process and filter data inside the database, thus improving performance too.

- Data security, replication, partitioning, and bulk load utilities are readily available.
- Raster, vector, XML and various types of attribute data can be stored on a single server. By effectively managing raster, vector and attribute data in a single data management environment – with common storage, indexing, spatial referencing, query optimization, security, and user management – Oracle reduces the processing overhead and eliminates the complexity of coordinating and synchronizing disparate sets of spatial data.
- It provides the best platform for consolidation of disparate data management environments, such as GIS, remote sensing, and business data. Complete and customized data management solutions can be built for both business and traditional geospatial applications.
- Oracle Database map visualization component and Spatial Studio support GeoRaster. Internet deployment enables a large number of concurrent users to access an application at virtually no additional cost with 24 x 365 uptime.
- It is supported by and integrated with leading third party and open source image processing, GIS, web services and visualization tools.
- Reduced training, software, support and application integration costs resulting from consolidating raster, vector and attribute storage.
- Risk reduction – imagery and raster information are integrated into Oracle Database resulting in scalability, reliability, and fast performance.

Conclusion

GeoRaster, a feature of Oracle Database, creates significant capacity for managing, processing, analyzing, and serving large volumes of raster data. Oracle is the only provider of commercial database management software that can store geospatial imagery and grid-based raster data in the database as a named native data type. It enables full integration of raster datasets with other enterprise datasets and supports better business applications.

The open and integrated raster data model supports various image and raster data types and simplifies data management tasks. It allows flexible blocking of raster data so that large-scale rasters can be stored and easily managed. Spatial and other indices provide fast searching and retrieval of raster metadata and cell data.

GeoRaster PL/SQL and Java API help users optimize the internal storage of any existing GeoRaster object, manage the database, search rasters based on area-of-interest, create large mosaics, process massive images, and conduct cartographic modeling with parallelism and on a global basis.

Selective raster query tools and other operations can be easily built upon GeoRaster to quickly serve imagery and rasters to middleware and applications. They reduce the cost of building and fielding of applications.

Users of file-based image processing systems and raster data applications benefit from the scalability, security, reliability and performance of Oracle Database and take advantage of enterprise grid computing technologies to support mission critical applications and integration.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle