Open Geospatial Consortium Inc.

Date:   2012-11-15

Reference number of this OGC® project document:    r1

Version: 0.2.0

Category: OGC® Implementation Specification

Editor:   Paul Daisey

OpenGIS® mentation Specification

Copyright notice

Copyright © 2012.Envitia  
 Copyright © 2012.Luciad  
 Copyright © 2012 Sigma Bravo.  
 Copyright © 2012 The Carbon Project.  
 Copyright © 2012 U.S. Army Geospatial Center.

Copyright © 2012 U.S. National Geospatial Intelligence Agency.

The companies listed above have granted the Open Geospatial Consortium, Inc. (OGC) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Document type:    OGC® Publicly Available Standard

Document subtype:    Implementation Specification

Document stage:    Draft

Document language:  English

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Contents

[Preface iv](#_Toc338759037)

[Submitting organizations v](#_Toc338759038)

[Submission contact points v](#_Toc338759039)

[Revision history v](#_Toc338759040)

[Changes to the OGC® Abstract Specification vi](#_Toc338759041)

[Changes to OpenGIS® Implementation Standards vi](#_Toc338759042)

[Future Work vi](#_Toc338759043)

[Foreword vi](#_Toc338759044)

[Introduction vii](#_Toc338759045)

[1 Scope 1](#_Toc338759046)

[2 Conformance 1](#_Toc338759047)

[3 Normative references 2](#_Toc338759048)

[4 Terms and definitions 5](#_Toc338759049)

[5 Conventions 5](#_Toc338759050)

[5.1 Symbols (and abbreviated terms) 5](#_Toc338759051)

[5.2 UML Notation 7](#_Toc338759052)

[6 Container 11](#_Toc338759053)

[6.1 Capabilities 11](#_Toc338759054)

[6.2 Reference Implementation 12](#_Toc338759055)

[7 Table Diagram 14](#_Toc338759056)

[8 Table of Contents 15](#_Toc338759057)

[8.1 Introduction 15](#_Toc338759058)

[8.2 Geopackage Contents Table 15](#_Toc338759059)

[9 Vector Feature Store 18](#_Toc338759060)

[9.1 Introduction 18](#_Toc338759061)

[9.2 Spatial Reference Systems Table 18](#_Toc338759062)

[9.3 Geometry Columns Table 19](#_Toc338759063)

[9.4 Feature Tables 23](#_Toc338759064)

[9.5 Geometry Routines SQL API 26](#_Toc338759065)

[9.5.1 Geometry Routines 26](#_Toc338759066)

[9.5.2 Point Routines 28](#_Toc338759067)

[9.5.3 Curve Routines 29](#_Toc338759068)

[9.5.4 LineString Routines 30](#_Toc338759069)

[9.5.5 Surface Routines 30](#_Toc338759070)

[9.5.6 Polygon Routines 31](#_Toc338759071)

[9.5.7 MultiGeometry Routines 31](#_Toc338759072)

[9.5.8 MultiPoint Routines 32](#_Toc338759073)

[9.5.9 MultiCurve Routines 33](#_Toc338759074)

[9.5.10 MultiLineString Routines 33](#_Toc338759075)

[9.5.11 MultiSurface Routines 34](#_Toc338759076)

[9.5.12 MultiPolygon Routines 34](#_Toc338759077)

[9.5.13 Spatial Predicates 35](#_Toc338759078)

[9.6 Reference Implementation 36](#_Toc338759079)

[10 Raster Tile Store 38](#_Toc338759080)

[10.1 Raster Tile Introduction 38](#_Toc338759081)

[10.2 Raster Columns 39](#_Toc338759082)

[10.3 Tile Table Metadata 43](#_Toc338759083)

[10.4 Tile Matrix Metadata 45](#_Toc338759084)

[10.5 Tiles Table 49](#_Toc338759085)

[10.6 Raster Tables 53](#_Toc338759086)

[10.7 Rasters or Tiles Table Metadata 55](#_Toc338759087)

[10.8 Image Routines SQL API 59](#_Toc338759088)

[10.8.1 gpkgAddTileTriggers SQL Function 60](#_Toc338759089)

[10.8.2 gpkgAddRtMetadataTriggers SQL Function 60](#_Toc338759090)

[10.8.3 gpkgGetNormalZoom SQL Function 60](#_Toc338759091)

[10.8.4 gpkgGetNormalRow SQL Function 61](#_Toc338759092)

[10.8.5 gpkgGetImageType SQL Function 61](#_Toc338759093)

[10.8.6 isJpegBlob SQL Function 61](#_Toc338759094)

[10.8.7 isPngBlob SQL Function 61](#_Toc338759095)

[10.8.8 isTiffBlob SQL Function 61](#_Toc338759096)

[10.8.9 isWebpBlob SQL Function 62](#_Toc338759097)

[10.8.10 gpkgBboxToTiles SQL Function 62](#_Toc338759098)

[10.8.11 gpkgPointToTile SQL Function 62](#_Toc338759099)

[11 Metadata 63](#_Toc338759100)

[11.1 Introduction 63](#_Toc338759101)

[11.2 XML Metadata Table 63](#_Toc338759102)

[11.3 Metadata Reference Table 67](#_Toc338759103)

[12 Other Data Tables 72](#_Toc338759104)

[13 Manifest 72](#_Toc338759105)

[13.1 Introduction 72](#_Toc338759106)

[13.2 Manifest Content Model and Table Definition 73](#_Toc338759107)

[13.3 Manifest XML Schema 73](#_Toc338759108)

[13.4 Manifest Schematron Schema 77](#_Toc338759109)

[13.5 Sample Manifest XML Document 77](#_Toc338759110)

[Annex A – Conformance – Abstract Test Suite (normative) 89](#_Toc338759111)

[Annex B – GeoPackage Trigger Definition SQL (informative) 94](#_Toc338759112)

[Annex C – Hierarchical Metadata Examples (informative) 105](#_Toc338759113)

[Bibliography (informative) 113](#_Toc338759114)

# Preface

An open standard non-proprietary platform-independent GeoPackage container for distribution and direct use of all kinds of geospatial data will increase the cross-platform interoperability of geospatial applications and web services. Standard APIs for access and management of GeoPackage data will provide consistent query and update results across such applications and services. Increased interoperability and result consistency will enlarge the potential market for such applications and services, particularly in resource-constrained mobile computing environments like cell phones and tablets. GeoPackages will become the standard containers for “MyGeoData” that are used as a transfer format by users and Geospatial Web Services and a storage format on personal and enterprise devices.

This OpenGIS**®** GeoPackage Implementation Specification defines a GeoPackage as a self-contained, single-file, cross-platform, serverless, transactional, open source RDBMS data container with table definitions, relational integrity constraints, an SQL API exposed via a “C” CLI and JDBC, and an XML manifest that together act as an exchange and direct-use format for multiple types of geospatial data including vector features, individual rasters and tile matrix pyramids, especially on mobile / hand held devices in disconnected or limited network connectivity environments.

Direct use requirements include the ability to access and update data in a “native” format without intermediate format translations in an environment (e.g. through an API) that guarantees data model and data set integrity and identical access and update results in response to identical requests from different client applications.

Table formats, definitions of geometry types and metadata tables, relational integrity constraints, and SQL API are interdependent specification facets of the SF-SQL [13][11][12] and SQL-MM (Spatial) [14] standards that serve as normative references for the vector feature portion of this specification. While they serve as a “data access layer”, another essential role they perform is that of the guarantor of geometry model and data integrity These standards have been a guiding force in the standardization of spatial RDMBS system capabilities across multiple commercial and open source spatial RDMBS and GIS implementations. Before these standards were adopted and widely implemented, identical spatial queries with spatial predicates (see clause 9.5.13) on the same data sets returned different results from almost every available spatial database and GIS implementation. These standards are cited here to avoid repetition of that experience on mobile / hand held platforms.

This specification attempts to adopt and adapt relevant raster types, storage table definitions, and metadata from widely adopted implementations such as MBTiles [B12] and existing standards such as WMTS [22] and ISO metadata [43], to integrate use of rasters as attributes of geospatial features, and to define relational integrity constraints and an SQL API thereon to provide a raster analogy to the SF-SQL and SF-MM data access and data quality assurance capabilities.

Conformance classes for this specification are classified as core (mandatory) and extension (optional). Data content types include vector features, raster features, tile matrix set pyramids, and an XML manifest. Core conformance classes address data storage formats – SQL table definitions for all data content types and associated metadata tables, procedures to create and index vector featue tables,and raster image formats that shall be supported. Extension conformance classes address SQL API routines, relational integrity constraints implemented via SQL triggers, additional raster image formats that may be supported, and an XML/Schema for the manifest.. Annex A contains the definitions of these conformance classes, which group and include all of the requirements specified in this standard.

Future versions of this specification will include requirements for elevation data and routes. Future enhancements to this specification, a future GeoPackage Web Service specification, and modifications to existing OGC Web Service (OWS) specifications to use GeoPackages as exchange formats will allow OWS to support provisioning of GeoPackages throughout an enterprise or information community.

# Submitting organizations

The following organizations submitted this Implementation Specification to the Open Geospatial Consortium Inc. as a Request For Comment (RFC).

1. Envitia
2. Luciad
3. Sigma Bravo
4. The Carbon Project
5. U.S. Army Geospatial Center
6. U.S. National Geospatial Intelligence Agency

# Submission contact points

All questions regarding this submission should be directed to the editor or the submitters:

|  |  |  |
| --- | --- | --- |
| **Contact** | **Email** | **Company** |
| Roger Brackin | roger.brackin@envitia.com | Envitia |
| Pepijn Van Eeckhoudt | pepijn.vaneeckhoudt@luciad.com | Luciad |
| Brad Hards | bhards@sigmabravo.com | Sigma Bravo |
| Jeff Harrison | jharrison@thecarbonproject.com | The Carbon Project |
| Kevin Backe | Kevin.Backe@usace.army.mil | U.S. Army Geospatial Center |
| David Wesloh | David.G.Wesloh@nga.mil | U.S. National Geospatial Intelligence Agency |
| Paul Daisey | pauld@imagemattersllc.com | Image Matters LLC |

# Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Release** | **Author** | **Paragraph modified** | **Description** |
| 2012-11-15 | r1 | Paul Daisey | 10.3 | Remove min/max x/y fields from all tables and text in clause 10.3 Tile Table Metadata  per change request 250. |
| 2012-11-15 | r1 | Paul Daisey | 10.2, Annex B | add compr\_qual\_factor and georectification columns to raster\_columns table create statement and sample insert statement; add triggers for those columns matching those for \_rt\_metadata per change request 251 |
|  |  |  |  |  |
|  |  |  |  |  |

# Changes to the OGC® Abstract Specification

The OGC**®** Abstract Specification requires/does not require changes to accommodate this OGC**®** standard. The needed changes are proposed in OGC documents XX-XXX, XX-XXX, and XX-XXX. / The following is a list of the required changes:

# Changes to OpenGIS® Implementation Standards

None at present.

# Future Work

* Future versions of this specification may include requirements for elevation data and routes.
* Future enhancements to this specification, a future GeoPackage Web Service specification and modifications to existing OGC Web Service (OWS) specifications to use GeoPackages as exchange formats will allow OWS to support provisioning of GeoPackages throughout an enterprise.
* Future versions of this specification may include additional raster / image formats, including fewer restrictions on the image/tiff format.
* Future versions of this specification may include additional SQL API routines for interrogation and conversion of raster / image BLOBs.
* Future versions of this specification and/or one for a GeoPackage Web Service may address utilities for importing and exporting vector, raster and tile data in various formats.
* Future versions of this specification and/or one for a GeoPackage Web Service may address encryption of GeoPackages and/or individual tables or column values.

# Foreword

The following organizations and individuals have contributed to the preparation of this standard:

* Alessandro Furieri
* Compusult Limited
* Development Seed
* Envitia
* Feng China University
* George Mason University
* Image Matters LLC
* International Geospatial Services Institute (iGSI) GmbH
* LMN Solutions
* Luciad
* MapBox
* OpenGeo
* Sigma Bravo
* The Carbon Project
* Universitat Autònoma de Barcelona (CREAF)
* U.S. Army Geospatial Center (AGC)
* U.S. National Aeronautics and Space Administration (NASA)
* U.S. National Geospatial Intelligence Agency (NGA)

This document includes 2 annexes; Annex A is normative, and Annex B and the Bibliography are informative.

*Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.*

*Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.*

# Introduction

Mobile device users who require map/geospatial application services and operate in disconnected or limited network connectivity environments are challenged by limited storage capacity and the lack of open format geospatial data to support these applications. The current situation is that each map/geospatial application requires its own potentially proprietary geospatial data store. These separate application-specific data stores may contain the same geospatial data, wasting the limited storage available, and requiring custom applications for data translation, replication, and synchronization to enable different map/geospatial applications to share the same world view. In addition, many existing geospatial data stores are platform-specific, which means that users with different platforms must translate data to share it.

An open, standards-based, application-independent, platform-independent, portable, interoperable, self-describing, GeoPackage (GPKG) data container, API and manifest are needed to overcome these challenges and to effectively support multiple map/geospatial applications such as fixed product distribution, local data collection, and geospatially enabled analytics. This standard is intended to facilitate widespread adoption and use of GeoPackages by both COTS and open-source software applications on enterprise production platforms as well as mobile hand-held devices, given that mobile hand held devices do not yet have the processing power or battery life to effectively tackle difficult geospatial product production and analysis tasks. An application that accesses a GPKG will make use of the GPKG capabilities it requires; few if any such applications will make use of all GPKG capabilities.

OpenGIS® GeoPackage Implementation Specification

# 1 Scope

This OpenGIS**®** GeoPackage Implementation Specification defines a GeoPackage as a self-contained, single-file, cross-platform, serverless, transactional, open source RDBMS data container with table definitions, relational integrity constraints, an SQL API exposed via a “C” CLI and JDBC, and an XML manifest that together act as an exchange and direct-use format for multiple types of geospatial data, especially on mobile / hand held devices in disconnected or limited network connected environments. Direct use requirements include the ability to access and update data in a “native” format without intermediate format translations in an environment (e.g. through an API) that guarantees data model and data set integrity and identical access and update results in response to identical requests from different client applications. Specifically, a GeoPackage can contain multiple vector feature types, rasters from various sources, and multiple tile matrix pyramids. A GeoPackage supports storage of rasters and tiles in multiple specified image file formats. Tiles are expected to be georectified or orthorectified view-space images, while rasters could also be raw “as collected” images. An individual GPKG may contain one, some or all of these types of geospatial data. The GeoPackage API provides Simple Features SQL access to vector features and geometries, and additional SQL functions on Rasters, Tiles, and their descriptive metadata. The GeoPackage API supports implementation of data content management and integrity constraints via SQL triggers. The GeoPackage Manifest serves as a table of contents and data source access metadata store for the contents of the GeoPackage data container.

The following items are outside the scope of this GeoPackage specification version, but may be in scope for future versions:

* Elevation data and routes.
* Additional SQL API routines to support 3D data and linear referencing measures.
* Additional raster / image formats
* Additional metadata required to georeference raw “as-collected” images.
* Additional SQL API routines for interrogation and conversion of raster / image BLOBs.
* Utilities for importing and exporting vector, raster and tile data in various formats.
* Encryption of GeoPackages and/or individual tables or column values.

# 2 Conformance

Implementations claiming conformance with this specification shall conform to the rules specified in Clauses 6 to 10 be checked using and pass all relevant test cases of any one of the conformance levels of the abstract test suite specified in Annex A (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing.

# 3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of OGC 12-128 For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of OGC 12-128 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

[1] ISO 19105: Geographic information — Conformance and Testing

[2] ISO/IEC 9075:1992 Information Technology - Database Language SQL (SQL92)

[3] ISO/IEC 9075-1:2011 Information Technology - Database Language SQL - Part 1: Framework

[4] ISO/IEC 9075-2:2011 Information Technology - Database Language SQL - Part 2: Foundation

[5] ISO/IEC 9075-3:2008 Information Technology - Database Language SQL - Part 3: Call-Level Interface (SQL/CLI)

[6] ISO/IEC 9075-4:2011 Information Technology - Database Language SQL - Part 4: Persistent Stored Modules (SQL/PSM)

[7] ISO/IEC 9075-10:2008 Information Technology - Database Language SQL – Part 10: Object Language Bindings (SQL/OLB)

[8] JDBC™ 3.0 Specification, Final Release, John Ellis & Linda Ho with Maydene Fisher, Sun Microsystems, Inc., October, 2001.

[9] SQLite (all parts) <http://www.sqlite.org/> (online) <http://www.sqlite.org/sqlite-doc-3071300.zip> (offline)

[10] <http://sqlite.org/fileformat2.html>

[11] OGC 06-103r4 OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture Version: 1.2.1 2011-05-28 <http://portal.opengeospatial.org/files/?artifact_id=25355> (also ISO/TC211 19125 Part 1)

[12] OGC 06-104r4 OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option Version: 1.2.1 2010-08-04 <http://portal.opengeospatial.org/files/?artifact_id=25354> (also ISO/TC211 19125 Part 2)

[13] OGC 99-049 OpenGIS® Simple Features Specification for SQL Revision 1.1 May 5, 1999, Clause 2.3.8 <http://portal.opengeospatial.org/files/?artifact_id=829>

[14] ISO/IEC 13249-3:2011 Information technology — SQL Multimedia and Application Packages - Part 3: Spatial (SQL/MM)

[15] <http://www.epsg.org/Geodetic.html>

[16] <http://www.epsg-registry.org/>

[17] MIL\_STD\_2401 DoD World Geodetic System 84 (WGS84), 11 January 1994

[18] <https://www.gaia-gis.it/fossil/libspatialite/index>

[19] <http://www.gaia-gis.it/gaia-sins/BLOB-Geometry.html>

[20] <http://trac.osgeo.org/geos/>

[21] <http://trac.osgeo.org/proj/>

[22] OGC 07-057r7 OpenGIS® Web Map Tile Service Implementation StandardVersion 1.0.02010-04-06 (WMTS) <http://portal.opengeospatial.org/files/?artifact_id=35326>

[23] ITU-T Recommendation T.81 (09/92) with Corrigendum (JPEG)

[24] JPEG File Interchange Format Version 1.02, September 1, 1992 <http://www.jpeg.org/public/jfif.pdf>

[25] IETF RFC 2046 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types <http://www.ietf.org/rfc/rfc2046.txt>

[26] Portable Network Graphics <http://libpng.org/pub/png/>

[27] MIME Media Types <http://www.iana.org/assignments/media-types/index.html>

[28] WebP <https://developers.google.com/speed/webp/>

[29] TIFF – Tagged Image File Format, Revision 6.0, Adobe Systems Inc., June 1992 <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>

[30] GeoTIFF Format Specification, Revision 1.0, 10 November 1995; version 1.8.2 <http://www.remotesensing.org/geotiff/spec/geotiffhome.html>

[31] NGA Standardization Document: Implementation Profile for Tagged Image File Format (TIFF) and Geographic Tagged Image File Format (GeoTIFF), Version 2.0, 2001-10-26 <https://nsgreg.nga.mil/doc/view?i=2224>

[32] IETF RFC 3986 Uniform Resource Identifier (URI): Generic Syntax <http://www.ietf.org/rfc/rfc3986.txt>

[33] OGC08-131r3 The Specification Model — A Standard for Modular specifications <https://portal.opengeospatial.org/files/?artifact_id=34762>

[34] OGC10-103 Name type specification - specification elements <http://portal.opengeospatial.org/files/?artifact_id=39194>

[35] W3C Recommendation 26 November 2008 Extensible Markup Language (XML) 1.0 (Fifth Edition) <http://www.w3.org/TR/xml/>

[36] W3C Recommendation 8 December 2009 Namespaces in XML 1.0 (Third Edition) <http://www.w3.org/TR/REC-xml-names/>

[37] W3C Recommendation 28 January 2009 XML Base (Second Edition) <http://www.w3.org/TR/xmlbase/>

**[**38**]** W3C Recommendation 06 May 2010 XML Linking Language (XLink) Version 1.1 <http://www.w3.org/TR/xlink11/>

[39] W3C Recommendation 28 October 2004 XML Schema Part 0: Primer Second Edition <http://www.w3.org/TR/xmlschema-0/>

[40] W3C Recommendation 28 October 2004 XML Schema Part 1: Structures Second Edition <http://www.w3.org/TR/xmlschema-1/>

[41] W3C Recommendation 28 October 2004 XML Schema Part 2: Datatypes Second Edition <http://www.w3.org/TR/xmlschema-2/>

[42] ISO 19115 Geographic information -- Metadata, 8 May 2003, with Technical Corrigendum 1, 5 July 2006

[43] ISO 8601 Representation of dates and times <http://www.iso.org/iso/catalogue_detail?csnumber=40874>

[44] OGC® 10-100r3 Geography Markup Language (GML) simple features profile (with technical note) <http://portal.opengeospatial.org/files/?artifact_id=42729>

[45] Atom Syndication Format - IETF RFC 4287 <http://tools.ietf.org/html/rfc4287>

[46] OGC® OWS Context Documen Conceptual Model

12-080 OWS Context Conceptual Model <https://portal.opengeospatial.org/wiki/OWSContextswg/ConceptualModelHome>

[47] OGC® OWS Context Atom Encoding

12-084 OWS Context Atom Encoding <https://portal.opengeospatial.org/wiki/OWS9/GeoPackageOWSContext>

# 4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

geolocate

identify a real-world geographic location

georectified

raster whose pixels have been regularly spaced in a geographic (i.e., latitude / longitude) or projected map coordinate system using ground control points so that any pixel can be geolocated given its grid coordinate and the grid origin, cell spacing, and orientation.

orthorectified

georectified raster that has also been corrected to remove image perspective (camera angle tilt), camera and lens induced distortions, and terrain induced distortions using camera calibration parameters and DEM elevation data to accurately align with real world coordinates, have constant scale, and support direct measurement of distances, angles, and areas.

# 5 Conventions

## 5.1 Symbols (and abbreviated terms)

Some frequently used abbreviated terms:

ACID Atomic, Consistent, Isolated, and Durable

ASCII American Standard Code for Information Interchange

API Application Program Interface

ATOM Atom Syndication Format

BLOB Binary Large OBject

CLI Call-Level Interface

COTS Commercial Off The Shelf

DEM Digital Elevation Model

DIGEST Digital Geographic Information Exchange Standard

GeoTIFF Geographic Tagged Image File Format

GPKG GeoPackage

GRD Ground Resolved Distance

EPSG European Petroleum Survey Group

FK Foreign Key

IETF Internet Engineering Task Force

IIRS Image Interpretability Rating Scale

IRARS Imagery Resolution Assessments and Reporting Standards (Committee)

ISO International Organization for Standardization

JDBC Java Data Base Connectivity

JPEG Joint Photographics Expert Group (image format)

MIME Multipurpose Internet Mail Extensions

NATO North Atlantic Treaty Organization

NITF National Imagery Transmission Format

OGC Open Geospatial Consortium

PK Primary Key

PNG Portable Network Graphics (image format)

RDBMS Relational Data Base Management System

RFC Request For Comments

SQL Structured Query Language

SRID Spatial Reference (System) Identifier

TIFF Tagged Image File Format

TIN Triangulated Irregular Network

UML Unified Modeling Language

UTC Coordinated Universal Time

XML eXtensible Markup Language

1D One Dimensional

2D Two Dimensional

3D Three Dimensional

## 5.2 UML Notation

The diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagrams. The UML notations used in this standard for RDBMS tables in the GeoPackage container are described in Figure 1 below.

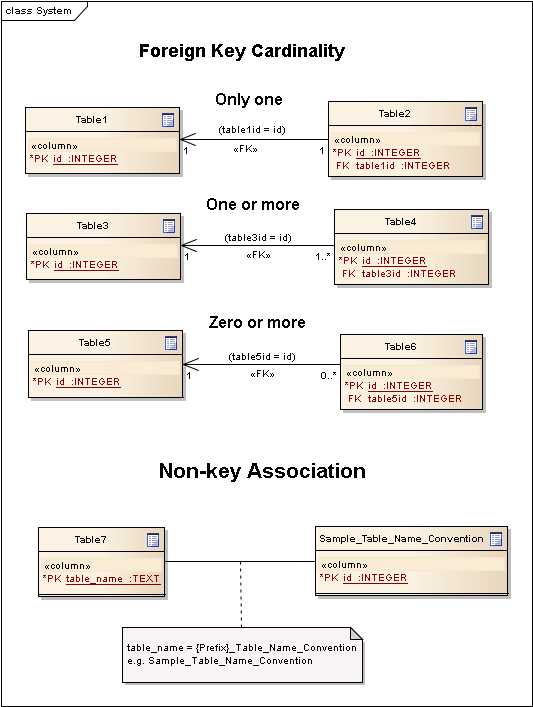


Figure 1 - UML Notation for RDBMS Tables

In this standard, the following two stereotypes of UML classes are used to represent RDBMS tables:

1. <<table>> An instantiation of a UML class as an RDMBS table.
2. <<column>> An instantiation of a UML attribute as an RDBMS table column.

In this standard, the following standard data types are used for RDBMS columns:

1. NULL – The value is a NULL value.
2. INTEGER – A signed integer, stored in 1, 2, 3, 4, 6, or 8 bytes depending on the magnitude of the value
3. REAL – The value is a floating point value, stored as an 8-byte IEEE floating point number.
4. TEXT – A sequence of characters, stored using the database encoding

(UTF-8, UTF-16BE or UTF-16LE).

1. BLOB – The value is a blob of data, stored exactly as it was input.
2. NONE – The value is a Date / Time Timestamp

The UML notations used in this standard for the eXtensible Markup Language (XML) schema for the GeoPackage manifest are described in Figure 2 below.

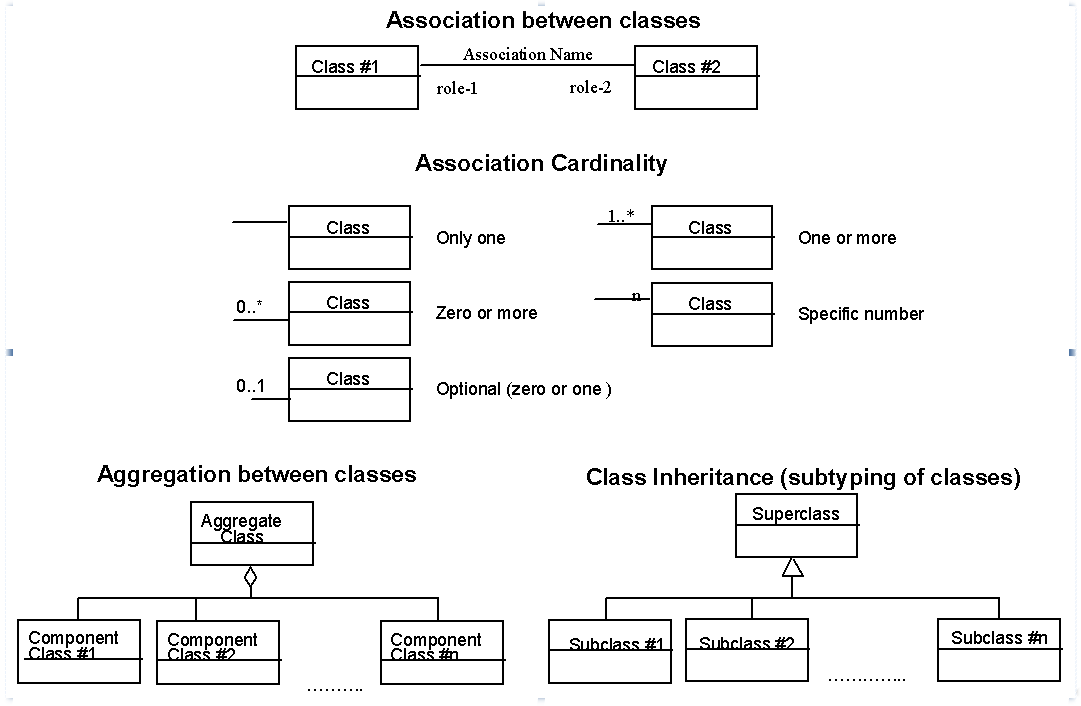


Figure 2 — UML notation for XML Schema

In this standard, the following stereotypes of UML classes are used to describe XML schemas:

1. <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.
2. <<Enumeration>> is a fixed enumeration that uses string values for expressing a list of potential values.
3. <<CodeList>> is an extensible enumeration that uses string values for expressing a list of potential values.

In this standard, the standard data types defined in XML Schema Part 2: Datatypes[42]are used to describe XML Schemas.

# 6 Container

## 6.1 Capabilities

A self-contained, single-file, cross-platform, serverless, transactional, open source RDBMS container is desired to simplify production, distribution and use of GeoPackages and guarantee the integrity of the data they contain. Self-contained means that container software requires very minimal support from external libraries or from the operating system. Single-file means that a container not currently opened by any software application consists of a single file in a file system supported by a computing platform operating system. Cross-platform means that a container file may be created and loaded with data on one computing platform, and used and updated on another, even if they use different operating systems, file systems, and byte order (endian) conventions. Serverless means that the RDBMS container is implemented without any intermediary server process, and accessed directly by application software. Transactional means that RDBMS transactions guarantee that all changes to data in the container are Atomic, Consistent, Isolated, and Durable (ACID) despite program crashes, operating system crashes, and power failures.

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/single-file> | |
| REQ 1. | A closed GeoPackage container shall consist of a single file. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/cross-platform> | |
| REQ 2. | A GeoPackage container shall be interchangeable and interoperable across platforms with different operating systems, file systems, and byte order (endian) conventions. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/transactional> | |
| REQ 3. | A GeoPackage container shall guarantee that all changes to data in the container are ACID. |

GeoPackage container conformance with current ISO/IEC 9075 (SQL) standards [3][4][5][6][7] would be optimal, but at a minimum the GeoPackage container shall support SQL-92 [2], including a “C” language CLI, BLOB data types, user-defined SQL functions, and FOR EACH ROW triggers, and shall support access via JDBC [8].

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/cli> | |
| REQ 4. | The GeoPackage container shall provide a “C” language CLI. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/blob> | |
| REQ 5. | The GeoPackage container shall support BLOB data types. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/functions> | |
| REQ 6. | The GeoPackage container shall support user-defined SQL functions. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/triggers> | |
| REQ 7. | The GeoPackage container shall support FOR EACH ROW triggers. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/jdbc> | |
| REQ 8. | The GeoPackage container shall support JDBC access. |

## 6.2 Reference Implementation

SQLite [9][10] shall be the initial reference implementation of the GeoPackage container.

NOTE: SQLite does not support updateable views, which must be simulated using FOR EACH ROW triggers.

Where a GeoPackage is provided in an SQLite database as a file on a file system, the file shall be named with a ".geopackage" extension, to enable handlers to determine that it is a GeoPackage without opening it and reading the manifest.

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/container/geopackage> | |
| REQ 9. | A GeoPackage container file with a .geopackage extension shall be a SQLite database of version 3.7 or later as determined by the SQL function sqlite\_version(). |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/container/file\_format | |
| REQ 10. | A GeoPackage container file shall have the file format specified by [10], with the first 16 bytes containing “SQLite format 3” in ASCII. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/container/file\_name\_extension | |
| REQ 11. | A GeoPackage container file shall have the file name extension “.geopackage”. |

CHECK constraints shall not used in SQL table definitions in GeoPackages because in SQLite they cannot RAISE exceptions and consequently return completely uninformative SQL errors. GeoPackages shall use before insert and update triggers to enforce the data constraints specified in table definitions in the following clauses, and RAISE exceptions to provide informative error messages.

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/container/constraints | |
| REQ 12. | Table definitions in an SQLite GeoPackage shall not include CHECK constraints. |

NOTE: .SQLite has been used as the base for a number of vector, raster and tile storage specifications, and commercial and open-source implementations. It is deployed and supported by Google on Android [B1] and Apple on IOS [B2] mobile devices. Testing on a laptop indicates that its performance scales well for databases in excess of 200GB containing vector and raster tables of more than 4 million rows. Some evolution in Mobile / Handheld Computing Environment file system capabilities will be necessary to allow such large files, e.g. in EXT file systems instead of FAT32 ones that limit file sizes to 4 GB.

# 7 Table Diagram

The SQL tables shown in this overview GeoPackage table diagram are defined and discussed in the sections that follow.

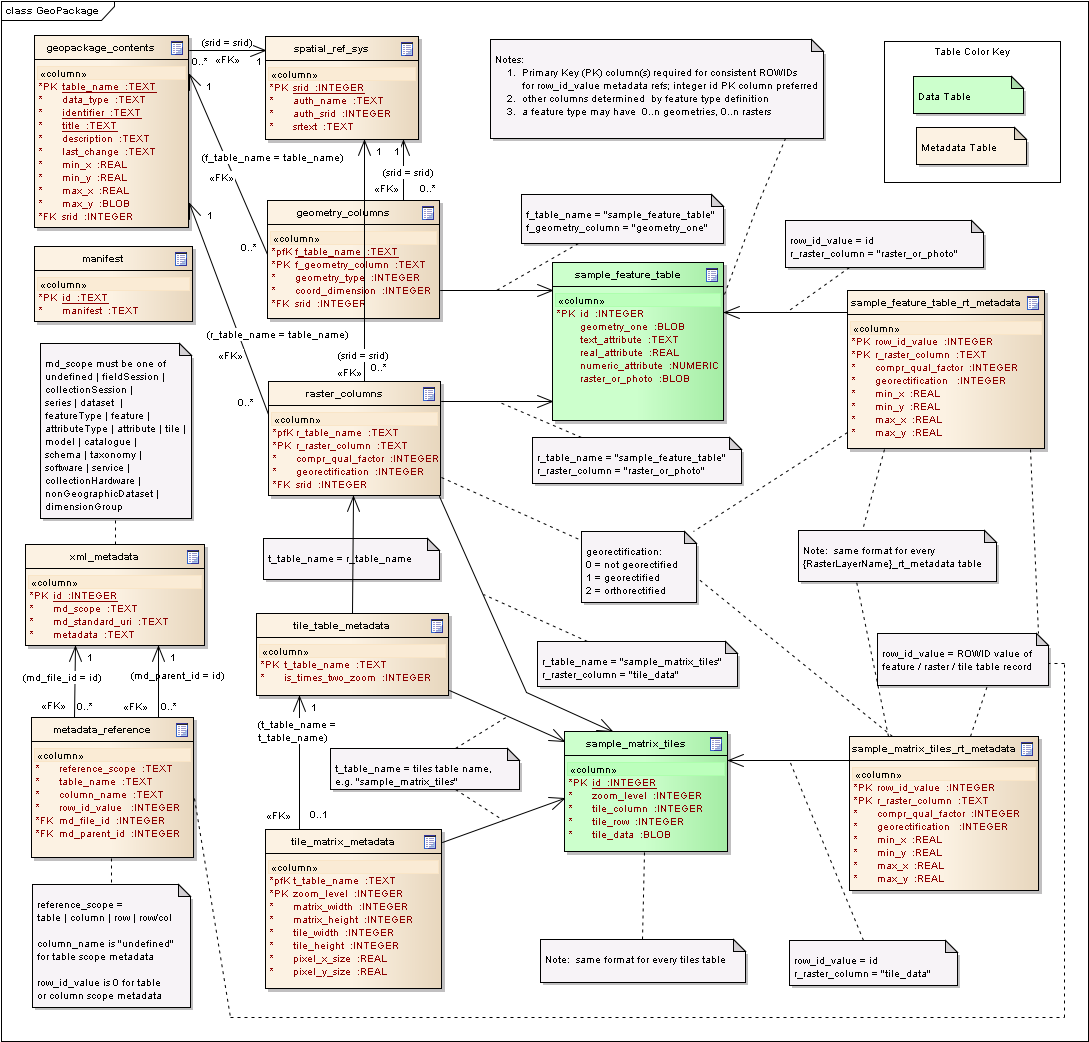


Figure 2 -- GeoPackage Tables

# 8 Table of Contents

## 8.1 Introduction

The purpose of the GeoPackage geopackage\_contents table is to provide identifying and descriptive information that an application can display to a user in a menu of geospatial data that is available for access and/or update.

## 8.2 Geopackage Contents Table

A GeoPackage shall contain a geopackage\_contents table or view as defined in this clause. The geopackage\_contents table or view shall contain one row record for each tile table, raster table and vector features table in the GeoPackage. The geopackage\_contents table or view shall not contain row records for any other type of table in a GeoPackage (see section 13 below).

Table – Geopackage Contents Table or View Definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: geopackage\_contents** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| table\_name | text | The name of the tiles, raster or feature table | no |  | PK |
| data\_type | text | Type of data stored in the table. Must be one of features | featuresWithRasters | rasters | tiles | no |  |  |
| identifier | text | A human-readable identifier (e.g. short name) for the table\_name content | no | “” |  |
| description | text | A human-readable description for the table\_name content | no | “” |  |
| last\_change | text | timestamp value in ISO 8601format as defined by the strftime function '%Y-%m-%dT%H:%M:%fZ' format string applied to the current time | no | strftime('%Y-%m-%dT%H:%M:%fZ', CURRENT\_TIMESTAMP) |  |
| min\_x | double | Bounding box for all content in table\_name | no | -180.0 |  |
| min\_y | double | Bounding box for all content in table\_name | no | -90.0 |  |
| max\_x | double | Bounding box for all content in table\_name | no | 180.0 |  |
| max\_y | double | Bounding box for all content in table\_name | no | 90.0 |  |
| srid | integer | Spatial Reference System ID: spatial\_ref\_sys.srid | no | 0 | FK |

The geopackage\_contents table is intended to provide a list of all geospatial contents in the GeoPackage. The data\_type specifies the type of content. The bounding box (min\_x, min\_y, max\_x, max\_y) provides an informative bounding box (not necessarily minimum bounding box) of the content.

Values of the geopackage\_contents table last\_change column shall be in ISO 8601 format containing a complete date plus UTC hours, minutes, seconds and a decimal fraction of a second, with a ‘Z’ (‘zulu’) suffix indicating UTC.

NOTE: The following statement selects such a timestamp value:

SELECT (strftime('%Y-%m-%dT%H:%M:%fZ','now')).

Table 2 – geopackage\_contents Table Definition SQL

|  |
| --- |
| CREATE TABLE geopackage\_contents (  table\_name TEXT NOT NULL PRIMARY KEY,  data\_type TEXT NOT NULL,  identifier TEXT NOT NULL UNIQUE,  description TEXT NOT NULL DEFAULT 'none',  last\_change TEXT NOT NULL DEFAULT  (strftime('%Y-%m-%dT%H:%M:%fZ',CURRENT\_TIMESTAMP)),  min\_x DOUBLE NOT NULL DEFAULT -180.0,  min\_y DOUBLE NOT NULL DEFAULT -90.0,  max\_x DOUBLE NOT NULL DEFAULT 180.0,  max\_y DOUBLE NOT NULL DEFAULT 90.0,  srid INTEGER NOT NULL DEFAULT 0,  CONSTRAINT fk\_gc\_r\_srid FOREIGN KEY (srid) REFERENCES spatial\_ref\_sys(srid)) |

The geopackage geopackage\_contents table shall have triggers defined to enforce column values as defined in table 3.

Table 3 – geopackage\_contents Trigger Definition SQL

|  |
| --- |
| CREATE TRIGGER 'geopackage\_contents\_table\_name\_insert'  BEFORE INSERT ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'insert on geopackage\_contents violates constraint: table\_name value must not contain a single quote')  WHERE NEW.table\_name LIKE ('%''%');  SELECT RAISE(ROLLBACK,'insert on geopackage\_contents violates constraint: table\_name value must not contain a double quote')  WHERE NEW.table\_name LIKE ('%"%');  SELECT RAISE(ROLLBACK,'insert on geopackage\_contents violates constraint: table\_name value must be lower case')  WHERE NEW.table\_name <> lower(NEW.table\_name);  END  CREATE TRIGGER 'geopackage\_contents\_table\_name\_update'  BEFORE UPDATE OF 'table\_name' ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'update of geopackage\_contents violates constraint: table\_name value must not contain a single quote')  WHERE NEW.table\_name LIKE ('%''%');  SELECT RAISE(ROLLBACK,'update of geopackage\_contents violates constraint: table\_name value must not contain a double quote')  WHERE NEW.table\_name LIKE ('%"%');  SELECT RAISE(ROLLBACK,'update of geopackage\_contents violates constraint: table\_name value must be lower case')  WHERE NEW.table\_name <> lower(NEW.table\_name);  END  CREATE TRIGGER 'geopackage\_contents\_feature\_type\_insert'  BEFORE INSERT ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table geopackage\_contents violates constraint: data\_type must be one of features | featuresWithRasters | rasters | tiles')  WHERE NOT(NEW.data\_type IN ('features','featuresWithRasters','rasters','tiles'));  END  CREATE TRIGGER 'geopackage\_contents\_feature\_type\_update'  BEFORE UPDATE OF 'data\_type' ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table geopackage\_contents violates constraint: data\_type must be one of features | featuresWithRasters | rasters | tiles')  WHERE NOT(NEW.data\_type IN ('features','featuresWithRasters','rasters','tiles'));  END |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/contents/geopackage_contents_table> | |
| REQ 13. | A GeoPackage shall include a geopackage\_contents table or updateable view that includes the columns and constraints defined in Tables 1 and 2 and clause 8.2, and contains the records specified in clause 8.2. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| [http://www.opengis.net/spec/GPKG/1.0/req/contents/geopackage\_contents\_table /triggers](http://www.opengis.net/spec/GPKG/1.0/req/contents/geopackage_contents_table%20/triggers) | |
| REQ 14. | A GeoPackage shall include SQL triggers on the geopackage\_contents table or updateable view as defined in Table 3 and clause 8.2. |

# 9 Vector Feature Store

## 9.1 Introduction

An RDBMS container store with SQL access for simple features with geometry is desired to manage (create, update, delete as well as search and retrieve) both geospatial foundation data for multiple types of features, and newly collected feature observation data. Initial support is required for the basic simple feature geometry types – Geometry, Point, LineString, LinearRing, Polygon, GeometryCollection, MultiPoint, MultiLineString, and MultiPolygon. Subsequent GPKG specification versions may require support for Curves with non-linear interpolation, Surfaces, MultiCurves, MultiSurfaces, Polyhedral Surfaces, TINs, and Full 3D. In both cases spatial indexing and SQL spatial routines are required for access, transform, and relational functions and predicates on geometry types to support direct use by geospatial applications.

Fortunately there are applicable international specifications that have standardized practices for the storage, access and use of vector geospatial features and geometries via SQL in relational databases. The original Simple Features for SQL specification OGC 99-049 [13] and its successors OGC 06-103r4 [11] and 06-104r4 [12] (ISO 19125) describe the common architecture for simple feature geometry and define a standard Structured Query Language (SQL) schema that supports storage, retrieval, query and update of feature collections via the SQL Call-Level Interface (SQL/CLI) [5]. They standardize:

a) Names and geometric definitions of the SQL Types for Geometry;

b) Names, signatures and geometric definitions of the SQL Routines for Geometry.

NOTE: The SF-SQL[13][11][12] and SQL-MM for Geospatial [14] set of standards has been a guiding force in the standardization of spatial RDMBS system capabilities across multiple commercial and open source spatial RDMBS and GIS implementations. Before these standards were adopted and widely implemented, identical spatial queries with spatial predicates (see clause 9.5.13) on the same data sets returned different results from almost every available spatial database and GIS implementation. These standards are cited here to avoid repetition of that experience on mobile / hand held platforms.

## 9.2 Spatial Reference Systems Table

The first component of the standard SQL schema for simple features is a table or updateable view with at a minimum the columns specified in OGC 06-104r4 [11] section 7.1.2.2 containing data that defines spatial reference systems. This table or view may include additional columns to meet the requirements of implementation software or other specifications.

At a minimum this table shall contain a record for auth\_name EPSG and auth\_srid 4326 [15][16] for WGS-84 [17], a record with an SRID of -1, an auth\_name of “NONE”, an auth\_srid of -1, and srtext “undefined” for undefined Cartesian coordinate reference systems, and a record with an SRID of 0, an auth\_name of “NONE”, an auth\_srid of 0, and srtext “undefined” for undefined geographic coordinate reference systems. It shall contain records to define all other spatial reference systems used by features, rasters and tiles in a GeoPackage. It is recommended to contain records for all coordinate reference systems in the latest EPSG database [B3] when storage space permits.

In OGC 06-104r4 only the primary key column of this table is defined to be NOT NULL. In a GeoPackage, all columns in this table or updateable view shall be defined to be NOT NULL.

NOTE1: See [B19] chapter 6 for a discussion of NULL column values.

NOTE2: See section 8.6 for implementation note on undefined SRID values.

Table 4 -- Spatial Reference Systems Table or View Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Column Type** | **Column Description** | **Key** |
| srid | integer | Unique identifier for each Spatial Reference System within a database | PK |
| auth\_name | text | the name of the standard or standards body that is being cited for this reference system, e.g. EPSG |  |
| auth\_srid | integer | the ID of the Spatial Reference System as defined by the Authority cited in AUTH\_NAME |  |
| srtext | text | Well-known Text Representation of the Spatial Reference System as specified in OGC 06-103r4 section 9 |  |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/vector_features/spatial_reference_systems_table> | |
| REQ 15. | A GeoPackage shall include a spatial\_reference\_systems table or updateable view with at a minimum the columns defined in Table 4 and clause 9.2 and containing data described in clause 9.2. |

## 9.3 Geometry Columns Table

The second component of the standard SQL schema is a table or updateable view specified in OGC 06-104r4 [12] section 7.1.3.1 that identifies the geometry columns in tables that contain data representing simple features. This table or updateable view shall contain one record for each geometry column in each vector feature data table in a GeoPackage.

Table 5 -- Geometry Columns Table or View Definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Column Type** | **Column Description** | **Key** |
| **f\_table\_catalog** | text | Catalog containing the table containing the geometry column | PK |
| **f\_table\_schema** | text | Schema containing the table containing the geometry column | PK |
| f\_table\_name | text | Name of the table containing the geometry column | PK,FK |
| f\_geometry\_column | text | Name of a column in the feature table that is a Geometry Column | PK |
| **g\_table\_catalog** | text | Catalog containing separate geometry table, or f\_table\_catalog if no separate geometry table |  |
| **g\_table\_schema** | text | Schema containing separate geometry table, or f\_table\_schema if no separate geometry table. |  |
| **g\_table\_name** | text | Name of separate geometry table, or f\_table\_name if no separate geometry table |  |
| **storage\_type** | integer | Storage type; 0=normalized (separate geometry table),  1=binary “gB”, 2=geometry types “gS” |  |
| geometry\_type | integer | Code from OGC 06-104r4 Table 4, except those for polyhedral surfaces:  0= geometry (2D), 1=point (2D)… |  |
| coord\_dimension | integer | Number of ordinates, +1 if geometry\_type includes an “M” Measure dimension. |  |
| **max\_ppr** | integer | Points per row for storage\_type=0, otherwise NULL |  |
| srid | integer | Spatial Reference System ID: spatial\_ref\_sys.id | FK |

SQLite does not have separate catalogs or schemas, so the **f\_table\_catalog** and **f\_table\_schema** columns are meaningless in an SQLite GPKG container. The foreign key (FK) on f\_table\_name references the primary key (PK) on geopackage\_contents to ensure that geometry columns are only defined for identified feature tables. As described below, the GPKG container does not use a separate geometry table, so the **g\_table\_catalog, g\_table\_schema, g\_table\_name**, and **max\_ppr** columns are also meaningless in a GPKG container. As described below, the initial GPKG container will only support binary “gB” geometry representations, so the **storage\_type** column is also meaningless until GPKG containers can also support “gS” geometry representations. Initial SQLite GPKG implementations may omit **these columns** with names in **blue** in the table above, and shall omit the **max\_ppr** column since the **storage\_type** 0 is not supported in a GPKG. SQLite GPKG implementations may include additional columns in this table or view (see section 9.6 below). The FK on geometry\_columns.srid references the PK on spatial\_ref\_sys.srid to ensure that geometry columns are only defined in feature tables for defined spatial reference systems. This table or view may include additional columns to meet the requirements of implementation software or other specifications.

In OGC 06-104r4, all columns in this table except storage\_type, geometry\_type, coord\_dimension, and max\_ppr are defined to be NOT NULL. The column definitions for the storage\_type (if included), geometry\_type and coord\_dimension columns for this table or updateable view in a GeoPackage shall also be defined to be NOT NULL.

Table 6 -- geometry\_columns Trigger Definitions

|  |
| --- |
| CREATE TRIGGER 'geometry\_columns\_f\_geometry\_column\_insert'  BEFORE INSERT ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'insert on geometry\_columns violates constraint: f\_geometry\_column value must not contain a single quote')  WHERE NEW.f\_geometry\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'insert on geometry\_columns violates constraint: f\_geometry\_column value must not contain a double quote')  WHERE NEW.f\_geometry\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'insert on geometry\_columns violates constraint: f\_geometry\_column value must be lower case')  WHERE NEW.f\_geometry\_column <> lower(NEW.f\_geometry\_column);  END  CREATE TRIGGER 'geometry\_columns\_f\_geometry\_column\_update'  BEFORE UPDATE OF 'f\_geometry\_column' ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'update on geometry\_columns violates constraint: f\_geometry\_column value must not contain a single quote')  WHERE NEW.f\_geometry\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'update on geometry\_columns violates constraint: f\_geometry\_column value must not contain a double quote')  WHERE NEW.f\_geometry\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'update on geometry\_columns violates constraint: f\_geometry\_column value must be lower case')  WHERE NEW.f\_geometry\_column <> lower(NEW.f\_geometry\_column);  END  CREATE TRIGGER 'geometry\_columns\_geometry\_type\_insert'  BEFORE INSERT ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'geometry\_type must be one of 0,1,2,3,4,5,6,7,13,14,1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007')  WHERE NOT(NEW.geometry\_type IN (0,1,2,3,4,5,6,7,13,14,  1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007));  END  CREATE TRIGGER 'geometry\_columns\_geometry\_type\_update'  BEFORE UPDATE OF 'geometry\_type' ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'geometry\_type must be one of 0,1,2,3,4,5,6,7,13,14,1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007)  WHERE NOT(NEW.geometry\_type IN (0,1,2,3,4,5,6,7,13,14,  1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007));  END  CREATE TRIGGER 'geometry\_columns\_coord\_dimension\_insert'  BEFORE INSERT ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'coord\_dimension must be one of 2,3,4')  WHERE NOT(NEW.geometry\_type IN (2,3,4));  END  CREATE TRIGGER 'geometry\_columns\_coord\_dimension\_update'  BEFORE UPDATE OF 'coord\_dimension' ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'coord\_dimension must be one of 2,3,4')  WHERE NOT(NEW.geometry\_type IN (2,3,4));  END |

|  |  |
| --- | --- |
| Requirement: Core | |
| [http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_columns\_table](http://www.opengis.net/spec/GPKG/1.0/req/vector_features/spatial_reference_systems_table) | |
| REQ 16. | A GeoPackage shall include a table or updateable view named “geometry\_columns” that at a minimum includes the columns and foreign key constraints defined in Table 5 and clause 9.3, and containing data described in clause 9.3. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| <http://www.opengis.net/spec/GPKG/1.0/req/vector_features/geometry_columns_table/triggers> | |
| REQ 17. | A GeoPackage shall include the triggers on a table or updateable view named “geometry\_columns” as defined in Table 6. |

## 9.4 Feature Tables

The third component of the standard SQL schema for simple features includes two different storage architectures for tables that contain data representing simple features and geometries. The first architecture represents geometries using predefined data types in a separate geometry table. The second architecture uses SQL geometry types for geometry columns in feature tables.

There are two variants of the “SQL with geometry types” architecture. The first, called “gB”, uses binary types (SQL BLOBs) to contain geometries, and is supported by SQL-92. These BLOBs are required to contain values of the corresponding geometry\_columns.geometry\_type that include “Well Known Binary (WKB)” format data specified in OGC 06-103r4 [11]. The second, called “gS”, uses SQL user-defined types as specified by SQL/MM in ISO/IEC 13249-3 [14], which defines spatial user-defined types (for the corresponding geometry\_columns.geometry\_type) and their associated routines. User defined types were introduced in SQL 3, and so the “gS” geometry types are not supported by SQL-92.

GeoPackages shall initially use the “SQL with geometry types” “gB” architecture with BLOB geometry columns in feature tables supported by SQL-92. GPKG “gB” feature tables shall be defined as specified in OGC 99-049 [13] section 2.3.8, including the use of stored procedures or functions to add and remove geometry columns.

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/add\_feature\_table\_geometry | |
| REQ 18. | A GeoPackage shall provide an **AddGeometryColumn** stored procedure or function for the “SQL with geometry types” “gB” architecture with BLOB geometry columns as specified in OGC 99-049 [13] section 2.3.8. and sections 9.4 and 9.5.1 of this standard. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/remove\_feature\_table\_geometry | |
| REQ 19. | A GeoPackage shall provide a **DropGeometryColumn** stored procedure or function for the “SQL with geometry types” “gB” architecture with BLOB geometry columns as specified in OGC 99-049 [13] section 2.3.8. and section 9.4 of this standard. |

NOTE1: When GeoPackage containers that support current ISO/IEC 9075 standards become available, GeoPackages may also use the “SQL with geometry types” “gS” architecture with the user-defined geometry types from ISO/IEC 13249-3 [14]. GPKG “gS” feature tables shall then be defined as specified in OGC 06-104r4 [12] section 7.2.3.

All geometry types and relational operators implemented by a GeoPackage for either the “gB” or “gS” architecture shall conform to the geometry architecture model specified in section 6.1 and Figure 1 “Geometry Class Hierarchy” of OGC 06-103r4 [11]. At a minimum a GeoPackage shall support the GEOMETRY, POINT, LINESTRING, POLYGON, GEOMETRYCOLLECTION, MULTIPOINT, MULTILINESTRING and MULTIPOLYGON geometry types.

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_types | |
| REQ 20. | A GeoPackage shall support the use of GEOMETRY, POINT, LINESTRING, POLYGON, GEOMETRYCOLLECTION, MULTIPOINT, MULTILINESTRING and MULTIPOLYGON geometry types in feature tables as specified in OGC 06-103r4 and section 9.4 of this standard. |

GeoPackage implementations of the “gB” and “gS” architectures shall assure that the geometry values stored in a geometry column of a feature table are of a geometry type that is compatible with (the same as or a subclass of) the geometry type specified for the column in the geometry\_columns.geometry\_type value for the geometry column, and are of the SRID specified for the geometry column in the geometry\_columns.srid for the column by implementation of appropriate SQL constraints and triggers.

NOTE2: A feature type may be defined to have 0..n geometry attributes, so the corresponding feature table may contain from 0..n geometry columns.

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/reject\_insert\_invalid\_geometry\_type | |
| REQ 21. | A GeoPackage shall reject (not perform) and return an error in response to an SQL statement that attempts to insert a feature table row with a geometry value of a geometry type than is incompatible with that specified for that feature table geometry column in the geometry\_columns table as specified in clauses 9.3 and 9.4. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/reject\_update\_invalid\_geometry\_type | |
| REQ 22. | A GeoPackage shall reject (not perform) and return an error in response to an SQL statement that attempts to update a feature table row with a geometry value of a geometry type than is incompatible with that specified for that feature table geometry column in the geometry\_columns table as specified in clauses 9.3 and 9.4. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/reject\_insert\_invalid\_geometry\_srid | |
| REQ 23. | A GeoPackage shall reject (not perform) and return an error in response to an SQL statement that attempts to insert a feature table row with a geometry with a different SRID than is specified for that feature table geometry column in the geometry\_columns table as specified in clauses 9.3 and 9.4. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/reject\_update\_invalid\_geometry\_srid | |
| REQ 24. | A GeoPackage shall reject (not perform) and return an error in response to an SQL statement that attempts to update a feature table row with a geometry with a different SRID than is specified for that feature table geometry column in the geometry\_columns table as specified in clauses 9.3 and 9.4. |

GeoPackage implementations of the “gB” and “gS” architectures shall have the capability to create, delete and use a spatial index on any geometry column of a feature table to improve the performance of spatial queries that use the spatial methods listed below to select features based on spatial relationships. These indexes may be constructed using any applicable geometry routines listed below, e.g. Envelope, and any appropriate indexing mechanisms, such as quad trees or R trees.

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/create\_spatial\_index | |
| REQ 25. | A GeoPackage shall provide an SQL function or procedure to add a spatial index to a geometry column as specified in clauses 9.3 and 9.4. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/delete\_spatial\_index | |
| REQ 26 | A GeoPackage shall provide an SQL function or procedure to remove a spatial index from a geometry column as specified in clauses 9.3 and 9.4. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/spatial\_index\_results | |
| REQ 27 | A GeoPackage shall provide the same results with and without a spatial index as specified in clauses 9.3 and 9.4.from the ST\_Within query specified in section 9.5.13 using a 1 degree bounding box that selects 1000 or more features from 100,000 or more features spread across a 10 degree square region of interest that are stored in a table with an indexed or un-indexed geometry column. |

|  |  |
| --- | --- |
| **Requirement: Core** | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/spatial\_index\_performance | |
| REQ 28 | A GeoPackage shall provide results more rapidly with a spatial index as specified in clauses 9.3 and 9.4 than without the spatial index from the ST\_Within query specified in section 9.5.13 using a 1 degree bounding box that selects 1000 or more features from 100,000 or more features spread across a 10 degree square region of interest that are stored in a table with an indexed or un-indexed geometry column. |

NOTE 3: A feature type may be defined to have 0..n raster attributes, so the corresponding feature table may also be a raster table that contains from 0..n raster columns. See clause 10 below.

Every feature table in a GeoPackage shall have a primary key defined on one or more columns so that row level metadata records may be linked to the feature records in it by rowed as described in clause 12.3 below.

NOTE4: An integer column primary key is recommended for best performance.

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/integer\_primary\_key | |
| REQ 29 | Every feature table in a GeoPackage shall have a primary key defined on one or more columns as specified in clause 9.4. |

## 9.5 Geometry Routines SQL API

In both the “gB” and “gS” architecture implementations, GeoPackages shall provide the geometry-type routines specified by ISO/IEC 13249-3 [13] and the OGC Simple Features for SQL specifications [11][12][13] listed in the following tables. The ST\_AsGML and ST\_GeomFromGML routines shall operate on the GML-SF geometry types specified in [44].

These routines may be provided by alias of GeoPackage implementation routines that have the same parameters and produce the specified result but have different, non-standard names.

Note: In the following table, for brevity the “ST\_” prefix is removed from the names of Geometry types.

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api | |
| REQ 30 | A GeoPackage shall provide all the geometry routines listed in clause 9.5 as named or by alias as described in clause 9.5. |

### 9.5.1 Geometry Routines

Table 8 - Geometry Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| AddGeometryColumn | adds a geometry column to a feature table |  | 2.3.8 |  |  |
| DropGeometryColumn | drops a geometry column from a feature table |  | 2.3.8 |  |  |
| ST\_Dimension | returns the dimension of a Geometry value | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_GeometryType | returns the type of the Geometry value as a String | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_SRID | returns the spatial reference system identifier of an Geometry value | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_Transform | returns the Geometry value in the specified spatial reference system | 4.1.1.1 |  |  |  |
| ST\_IsEmpty | tests if a Geometry value corresponds to the empty set | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_IsSimple | tests if a Geometry value has no anomalous geometric point, such as self intersection or self tangency | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_IsValid | tests if a Geometry value is well formed | 4.1.1.1 |  |  |  |
| ST\_Is3D | tests whether a Geometry value has z coordinates | 4.1.1.1 |  | 6.1.2.2 |  |
| ST\_IsMeasured | tests whether a Geometry value has m coordinate values | 4.1.1.1 |  | 6.1.2.2 |  |
| ST\_LocateAlong | returns a derived geometry collection value that matches the specified m coordinate value | 4.1.1.1 |  | 6.1.2.6 |  |
| ST\_LocateBetween | returns a derived geometry collection value that matches the specified range of m coordinate values inclusively | 4.1.1.1 |  | 6.1.2.6 |  |
| ST\_Boundary | returns the boundary of a Geometry value | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_Envelope | returns the bounding rectangle of a Geometry value | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_ConvexHull | returns the convex hull of a Geometry value | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_Buffer | returns the Geometry value that represents all points whose distance from any point of a Geometry value is less than or equal to a specified distance | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_Intersection | returns the Geometry value that represents the point set intersection of two Geometry values | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_Union | returns the Geometry value that represents the point set union of two ST\_Geometry values | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_Difference | returns the Geometry value that represents the point set difference of two Geometry values | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_SymDifference | returns the Geometry value that represents the point set symmetric difference of two Geometry values | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_Distance | returns the distance between two geometries | 4.1.1.1 | 2.1.1.3 | 6.1.2.4 | 7.2.8.1 |
| ST\_AsText | returns the well-known text representation for the specified Geometry value | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 | 7.2.8.1 |
| ST\_AsBinary | returns the well-known binary representation for the specified Geometry value | 4.1.1.1 | 2.1.1.1 | 6.1.2.2 |  |
| ST\_AsGML | returns the GML representation for the specified Geometry value | 4.1.1.1 |  |  |  |
| ST\_GeomFromText | returns a Geometry value from its well-known text representation | 4.1.1.2 | 3.2.6.2 |  |  |
| ST\_GeomFromWKB | returns a Geometry value from its well-known binary representation | 4.1.1.2 | 3.2.7.2 |  |  |
| ST\_GeomFromGML | returns a Geometry value from its GML representation | 4.1.1.2 |  |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/geometry\_routines | |
| REQ 31 | Each geometry routine described in clause 9.5.1 of this specification provided by a GeoPackage shall operate as described in clause 9.5 and the cited clauses from [10][11][12][13]. |

### 9.5.2 Point Routines

Table 9 -- Point Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_X | returns the x coordinate value of a Point value | 4.1.3.1 | 3.2.11.2 | 6.1.4.2 | 7.2.9.1 |
| ST\_Y | returns the y coordinate value of a Point value | 4.1.3.1 | 3.2.11.2 | 6.1.4.2 | 7.2.9.1 |
| ST\_Z | returns the z coordinate value of a Point value | 4.1.3.1 |  | 6.1.4.2 | 7.2.9.1 |
| ST\_M | returns the m coordinate value of a Point value | 4.1.3.1 |  | 6.1.4.2 | 7.2.9.1 |
| ST\_PointFromText | returns a Point value from the well-known text representation of a Point | 4.1.3.2 | 3.2.6.2 |  |  |
| ST\_PointFromWKB | returns a Point value from the well-known binary representation of a Point. | 4.1.3.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/point\_routines | |
| REQ 32 | Each geometry routine described in section 8.5.2 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.3 Curve Routines

Table 10 -- Curve Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_Length | returns the length of a Curve value | 4.1.4.1 | 2.1.5.1 | 6.1.6.2 | 7.2.10.1 |
| ST\_StartPoint | returns the Point value that is the start point of a Curve value | 4.1.4.1 | 2.1.5.1 | 6.1.6.2 | 7.2.10.1 |
| ST\_EndPoint | returns the Point value that is the end point of a Curve value | 4.1.4.1 | 2.1.5.1 | 6.1.6.2 | 7.2.10.1 |
| ST\_IsClosed | tests if a Curve value is closed | 4.1.4.1 | 2.1.5.1 | 6.1.6.2 | 7.2.10.1 |
| ST\_IsRing | tests if an Curve value is a ring | 4.1.4.1 | 2.1.5.1 | 6.1.6.2 | 7.2.10.1 |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/curve\_routines | |
| REQ 33 | Each geometry routine described in section 9.5.3 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.4 LineString Routines

Table 11 -- LineString Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_NumPoints | returns the cardinality of the Point collection in the LineString value | 4.1.5.1 | 2.1.6.1 | 6.1.7.2 | 7.2.11.1 |
| ST\_PointN | returns the specified element in the Point collection in the LineString value | 4.1.5.1 | 2.1.6.1 | 6.1.7.2 | 7.2.11.1 |
| ST\_LineFromText | returns a LineString value from the well-known text representation of a LineString | 4.1.5.2 | 3.2.6.2 |  |  |
| ST\_LineFromWKB | returns a LineString value from the well-known binary representation of a LineString | 4.1.5.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/line\_string\_routines | |
| REQ 34 | Each geometry routine described in section 9.5.4 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.5 Surface Routines

Table 12 -- Surface Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_Area | returns the area of a Surface value | 4.1.8.1 | 2.1.9.1 | 6.1.10.2 | 7.2.12.1 |
| ST\_Centroid | returns the Point value that is the mathematical centroid of the Surface value | 4.1.8.1 | 2.1.9.1 | 6.1.10.2 | 7.2.12.1 |
| ST\_PointOnSurface | returns a Point value that is guaranteed to be on the Surface value | 4.1.8.1 | 2.1.9.1 | 6.1.10.2 | 7.2.12.1 |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/surface\_routines | |
| REQ 35 | Each geometry routine described in section 9.5.5 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.6 Polygon Routines

Table 13 -- Polygon Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_ExteriorRing | returns the exterior ring of a Polygon value | 4.1.9.1 | 2.1.10.1 | 6.1.11.2 | 7.2.13.1 |
| ST\_NumInteriorRing | returns the cardinality of the collection of interior rings of a Polygon value | 4.1.9.1 | 2.1.10.1 | 6.1.11.2 | 7.2.13.1 |
| ST\_InteriorRingN | returns the specified element in the collection of interior rings of a Polygon value | 4.1.9.1 | 2.1.10.1 | 6.1.11.2 | 7.2.13.1 |
| ST\_PolyFromText | returns a Polygon value from the well-known text representation of a Polygon | 4.1.10.2 | 3.2.6.2 |  |  |
| ST\_PolyFromWKB | returns a Polygon value from the well-known binary representation of a Polygon | 4.1.10.2 | 3.2.7.2 |  |  |
| ST\_BdPolyFromText | returns a Polygon value from a well-known text  representation of a MultiLineString | 4.1.10.2 | 3.2.6.2 |  |  |
| ST\_BdPolyFromWKB | returns a Polygon value from a well-known binary  representation of a MultiLineString | 4.1.10.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/polygon\_routines | |
| REQ 36 | Each geometry routine described in section 9.5.6 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.7 MultiGeometry Routines

Table 14 -- MultiGeometry Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_NumGeometries | returns the cardinality of the Geometry collection | 4.1.11.1 | 3.2.16.2 | 6.1.3.2 | 7.2.15 |
| ST\_GeometryN | returns the specified element in the Geometry collection | 4.1.11.1 | 3.2.16.2 | 6.1.3.2 | 7.2.15 |
| ST\_GeomCollFromTxt | returns a GeomCollection value from the well-known text representation of a GeomCollection | 4.1.11.2 | 3.2.6.2 |  |  |
| ST\_GeomCollFromWKB | returns a GeomCollection value from the well-known binary representation of GeomCollection | 4.1.11.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/multigeometry\_routines | |
| REQ 37 | Each geometry routine described in section 9.5.7 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.8 MultiPoint Routines

Table 15 -- MultiPoint Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_MPointFromText | returns a MultiPoint value from the well-known text representation of a MultiPoint | 4.1.12.2 | 3.2.6.2 |  |  |
| ST\_MPointFromWKB | returns a MultiPoint value from the well-known binary representation of a MultiPoint | 4.1.12.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/multipoint\_routines | |
| REQ 38 | Each geometry routine described in section 9.5.8 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.9 MultiCurve Routines

Table 16 -- MultiCurve Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_IsClosed | tests if a MultiCurve value is closed | 4.1.13.1 | 3.2.17.2 | 6.1.8.2 | 7.2.17.1 |
| ST\_Length | returns the length of a MultiCurve value | 4.1.13.1 | 3.2.17.2 | 6.1.8.2 | 7.2.17.1 |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/multicurve\_routines | |
| REQ 39 | Each geometry routine described in section 9.5.9 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.10 MultiLineString Routines

Table 17 -- MultiLineString Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_MLineFromText | returns a MultiLineString value from the well-known text representation of a MultiLineString | 4.1.14.2 | 3.2.6.2 |  |  |
| ST\_MLineFromWKB | returns a MultiLineString value from the well-known binary representation of a MultiLineString | 4.1.14.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/multilinestring\_routines | |
| REQ 40 | Each geometry routine described in section 9.5.10 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.11 MultiSurface Routines

Table 18 -- MultiSurface Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_Area | returns the area of a MultiSurface value | 4.1.15.1 | 3.2.18.2 | 6.1.13.2 | 7.2.19.1 |
| ST\_Centroid | returns the Point value that is the mathematical centroid of the MultiSurface value | 4.1.15.1 | 3.2.18.2 | 6.1.13.2 | 7.2.19.1 |
| ST\_PointOnSurface | returns a Point value that is guaranteed to be on the MultiSurface value | 4.1.15.1 | 3.2.18.2 | 6.1.13.2 | 7.2.19.1 |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/multisurface\_routines | |
| REQ 41 | Each geometry routine described in section 9.5.11 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.12 MultiPolygon Routines

Table 19 -- MultiPolygon Routines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_MPolyFromText | returns a MultiPolygon value from the well-known text representation of a MultiPolygon | 4.1.16.2 | 3.2.6.2 |  |  |
| ST\_MPolyFromWKB | returns a MultiPolygon value from the well-known binary representation of a MultiPolygon | 4.1.16.2 | 3.2.7.2 |  |  |
| ST\_BdMPolyFromText | returns a MultiPolygon value from a well-known text  representation of a MultiLineString | 4.1.16.2 | 3.2.6.2 |  |  |
| ST\_BdMPolyFromWKB | returns a MultiPolygon value from a well-known binary representation of a MultiLineString | 4.1.16.2 | 3.2.7.2 |  |  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/multipolygon\_routines | |
| REQ 42 | Each geometry routine described in section 9.5.12 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [10][11][12][13]. |

### 9.5.13 Spatial Predicates

Table 20 -- Spatial Predicates

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Routine Name** | **Description** | **13249-3 Clause** | **99-049 Clause** | **06-103r4 Clause** | **06-104r4 Clause** |
| ST\_Equals | tests if a Geometry value is spatially equal to another Geometry value. | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Relate | tests if a Geometry value is spatially related to another Geometry value by testing for intersections between the interior, boundary and exterior of the two Geometry values as specified by the intersection matrix. | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Disjoint | tests if a Geometry value is spatially disjoint from another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Intersects | tests if a Geometry value spatially intersects another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Touches | tests if a Geometry value spatially touches another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Crosses | tests if a Geometry value spatially crosses another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Within | tests if a Geometry value is spatially within another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Contains | tests if a Geometry value spatially contains another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |
| ST\_Overlaps | tests if a Geometry value spatially overlaps another Geometry value | 4.1.2.3 | 3.2.19.2 | 6.1.15.3 | 7.2.8.1 |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_routines\_sql\_api/spatial\_predicates | |
| REQ 43 | Each spatial predicate described in section 9.5.13 of this specification provided by a GeoPackage shall operate as described in the cited clauses from [11][12][13][14]. |

## 9.6 Reference Implementation

SpatiaLite [18] version 4 [B29] with additional geometry routines and spatial predicates provided by the Geometry Engine Open Source (GEOS) library [20] and PROJ.4 library [21] shall be the initial reference implementation of the GeoPackage Vector Feature Store.

NOTE1: SpatiaLite [18] version 4 [B29] defines two values for undefined Spatial Reference Systems (SRS):

SRID = -1, auth\_name "NONE", srtext="undefined" for Undefined Cartesian (planar, engineering) SRS

SRID = 0, auth\_name "NONE", srtext="undefined" for Undefined Geographic Longitude/Latitute SRS

NOTE2: SpatiaLite [18] adds a proj4text column to the spatial\_ref\_sys table to support the use of the PROJ.4 library.

|  |  |
| --- | --- |
| Requirement: Extension | |
| <http://www.opengis.net/spec/GPKG/1.0/req/vector_features/geopackage/spatialite> | |
| REQ 44. | A GeoPackage container file with a .geopackage extension shall be a SQLite database with libspatialite loaded to provide SpatiaLite extensions of version 3.0.1 or later as determined by the SQL function spatialite\_version(). |

|  |  |
| --- | --- |
| Requirement: Extension | |
| <http://www.opengis.net/spec/GPKG/1.0/req/vector_features/geopackage/geos> | |
| REQ 45. | A GeoPackage container file with a .geopackage extension shall be a SQLite database with SpatiaLite, with libgeos loaded to provide GEOS library extensions of version 3.3.1-CAPI-1.7.1 or later loaded as determined by the SQL function geos\_version. |

.

|  |  |
| --- | --- |
| Requirement: Extension | |
| <http://www.opengis.net/spec/GPKG/1.0/req/vector_features/geopackage/proj4> | |
| REQ 46. | A GeoPackage container file with a .geopackage extension shall be a SQLite database with SpatiaLite, with the PROJ.4 library extensions of version Rel. 4.7.1 or later loaded as determined by the SQL function proj4\_version() . |

In the SpatiaLite version 3 implementation of the spatial\_ref\_sys table, the column containing the WKT spatial reference system definitions is named “srs\_wkt”, whereas in the OGC specifications this column is named “srtext”. GeoPackages shall use the following transaction to add a copy of the “srs\_wkt” column as the "srtext" column to SpatiaLite version 3 implementations to match OGC 06-104r4 [12] clause 7.1.2.2. This addition is not required in SpatiaLite version 4, where the srs\_wkt column has been renamed srtext to match OGC 06-104r4 [12] clause 7.1.2.2 [B29].

Table 21 -- srstext column addition SQL

|  |
| --- |
| ALTER TABLE spatial\_ref\_sys ADD COLUMN srtext TEXT;  COMMIT TRANSACTION;  UPDATE spatial\_ref\_sys SET srtext = srs\_wkt;  COMMIT TRANSACTION; |

The SpatiaLite version 3 implementation of the geometry\_columns table contains a type column instead of a geometry\_type column. GeoPackages shall use the following statement to add a geometry\_type column to SpatiaLite version 3 implementations to match OGC 06-104r4 [12] clause 7.1.3.3. This addition is not required in SpatiaLite version 4 [B29].

Table 22 -- geometry\_type column addition SQL

|  |
| --- |
| ALTER TABLE geometry\_columns ADD COLUMN geometry\_type INTEGER; |

.

The SpatiaLite version 3 implementation of the AddGeometryColumns() function does not set the geometry\_type value. It must be set to the appropriate value shown in table 3 above. This step is not required in SpatiaLite version 4 [B29].

The SpatiaLite storage of SQL Geometry types prefixes WKB Geometry with endian, srid, and bounding box data [19]. This extension is allowed by OGC 99-049 [13] and its successors OGC 06-103r4[11]and 06-104r4 [12] (ISO 19125) which do “*not* attempt to standardize any part of the mechanism by which the Geometry Types are added to and maintained in the SQL environment” and do “not depend upon any part of [that] mechanism” including “The physical storage of type instances in the database”. However, to assure interoperability, GeoPackage Geometry BLOBS shall be stored with the binary structure specified by [19].

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/vector\_features/geometry\_blob | |
| REQ 47 | A GeoPackage container file shall store all geometry BLOBs using the binary structure specified by [19]. |

NOTE3: SpatiaLite is currently the only known vector feature store based on SQLite that meets the specifications documented above for vector features and SQL geometry routines. It is a supported format for the OGR Simple Features Library [B4][B5]. GEOS is widely used by both free and commercial software packages [B6]. Quantum GIS (QGIS) [B7] is an example of an open source GIS that can manage data in SpatiaLite and other spatial databases using GEOS. QGIS has been ported to Android-based tablets [B8]. Luciad Mobile [B9] is a situational awareness application that uses SpatiaLite for vector feature storage. Use of SpatiaLite for point feature storage is planned for FalconView version 5.1 [B10].

# 10 Raster Tile Store

## 10.1 Raster Tile Introduction

There are a wide variety of commercial and open source conventions for storing, indexing, accessing and describing individual rasters and tiles in tile matrix pyramids. Unfortunately, no applicable existing consensus, national or international specifications have standardized practices in this domain. In addition, various image file formats have different representational capabilities, and include different self-descriptive metadata.

The Raster / Tile Store data / metadata model, conventions and SQL functions described below support direct use of rasters and tiles in a GeoPackage in two ways. First, they specify how existing applications may create SQL Views of the data /metadata model on top of existing application tables that that follow different interface conventions. Second, they include and expose enough metadata information at both the dataset and record levels to allow applications that use GeoPackage data to discover its characteristics without having to parse all of the stored images. Applications that store GeoPackage raster and tile data, which are presumed to have this information available, shall store sufficient metadata to enable its intended use.

Following a convention used by MBTiles [B12], the Raster / Tile Store data model may be implemented directly as SQL tables in a SQLite database for maximum performance, or as SQL views on top of tables in an existing SQLite Raster / Tile store for maximum adaptability and loose coupling to enable widespread implementation. A GeoPackage can store multiple raster and tile pyramid data sets in different tables or views in the same container. Following a convention used by RasterLite [B13], tables or views containing record-level metadata are named with a raster or tile table name prefix and a “\_rt\_metadata” suffix, e.g. {RasterTableName}{\_rt\_metadata.

The tables or views that implement the GeoPackage Raster / Tile Store data / metadata model are described and discussed individually in the following subsections.

NOTE: Images of multiple MIME types may be stored in given table. For example, in a tiles table, image/png format tiles without compression could be used for transparency where there is no data on the tile edges, and image/jpeg format tiles with compression could be used for storage efficiency where there is image data for all pixels. Images of multiple bit depths of the same MIME type may also be stored in a given table, for example image/png tiles in both 8 and 24 bit depths.

## 10.2 Raster Columns

A GeoPackage shall contain a raster\_columns table or view as defined in this clause. The raster\_columns table or view shall contain one row record describing each raster or tile column in any table in a GeoPackage. The r\_raster\_column in r\_table\_name shall be defined as a BLOB data type.

The compr\_qual\_ factor column value indicates the lowest image quality of any raster or tile in the associated column on a scale from 1 (lowest) to 100 (highest) for rasters compressed with a lossy compression algorithm. It is always 100 if all rasters or tiles are compressed with a lossless compression algorithm, or are not compressed.

The georectification column value indicates the minimum level of georectification to areas on the earth for all rasters or tiles in the associated column are georectified. A value of 0 indicates that no rasters or tiles are georectified. A value of 1 indicates that all rasters or tiles are georectified (but not necessarily orthorectified). A value of 2 indicates that all rasters or tiles are orthorectified (which implies georectified) to accurately align with real world coordinates, have constant scale, and support direct measurement of distances, angles, and areas.

The srid shall have a value contained in the spatial\_ref\_sys table defined in clause 9.2 above.

All GeoPackages shall support image/png and image/jpeg formats for rasters and tiles. GeoPackages may support image/x-webp and image/tiff formats for rasters and tiles. GeoPackage support for the image/tiff format [29] is limted to GeoTIFF [30] images that meet the requirements of the NGA Implementation Profile [31] for coordinate transformation case 3 where the position and scale of thedata is known exactly, and no rotation of the image is required.

NOTE 1: A feature type may be defined to have 0..n raster attributes, so the corresponding feature table may contain from 0..n raster columns.

NOTE 2: A raster tile layer table has only one raster column named “tile\_data”.

Table 23 -- raster\_columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: raster\_columns** | | | | | |
| **Column Name** | **ColumnType** | **Column Description** | **Null** | **Default** | **Key** |
| r\_table\_name | text | Name of the table containing the raster column, e.g. {FeatureTableName} | {RasterLayerName}\_tiles | no |  | PK FK |
| r\_raster\_column | text | Name of a column in a table that is a raster column with a BLOB data type | no |  | PK |
| compr\_qual\_factor | integer | Compression quality factor: 1 (lowest) to 100 (highest) for lossy compression; always 100 for lossless or no compression. | no | 100 |  |
| georectification | integer | Is the raster georectified; 0=not georectified, 1=georectified, 2=orthorectified | no | 0 |  |
| srid | integer | Spatial Reference System ID: spatial\_ref\_sys.srid | no |  | FK |

Table 24 -- raster\_columns Table Definition SQL

|  |
| --- |
| CREATE TABLE raster\_columns (  r\_table\_name TEXT NOT NULL,  r\_raster\_column TEXT NOT NULL,  compr\_qual\_factor INTEGER NOT NULL DEFAULT 100,  georectification INTEGER NOT NULL DEFAULT 0,  srid INTEGER NOT NULL DEFAULT 0,  CONSTRAINT pk\_rc PRIMARY KEY (r\_table\_name, r\_raster\_column)  ON CONFLICT ROLLBACK,  CONSTRAINT fk\_rc\_r\_srid FOREIGN KEY (srid)  REFERENCES spatial\_ref\_sys(srid),  CONSTRAINT fk\_rc\_r\_gc FOREIGN KEY (r\_table\_name) REFERENCES geopackage\_contents(table\_name)) |

The raster\_columns table or view in a GeoPackage shall have the triggers defined in Table 21 below.

Table 25 -- raster\_columns Trigger Definition SQL

|  |
| --- |
| CREATE TRIGGER 'raster\_columns\_r\_raster\_column\_insert'  BEFORE INSERT ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'insert on raster\_columns violates constraint: r\_raster\_column value must not contain a single quote')  WHERE NEW.r\_raster\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'insert on raster\_columns violates constraint: r\_raster\_column value must not contain a double quote')  WHERE NEW.r\_raster\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'insert on raster\_columns violates constraint: r\_raster\_column value must be lower case')  WHERE NEW.r\_raster\_column <> lower(NEW.r\_raster\_column);  END  CREATE TRIGGER 'raster\_columns\_r\_raster\_column\_update'  BEFORE UPDATE OF r\_raster\_column ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'update on raster\_columns violates constraint: r\_raster\_column value must not contain a single quote')  WHERE NEW.r\_raster\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'update on raster\_columns violates constraint: r\_raster\_column value must not contain a double quote')  WHERE NEW.r\_raster\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'update on raster\_columns violates constraint: r\_raster\_column value must be lower case')  WHERE NEW.r\_raster\_column <> lower(NEW.r\_raster\_column);  END  CREATE TRIGGER 'raster\_columns\_georectification\_insert'  BEFORE INSERT ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''raster\_columns'' violates constraint: georectification must be 0, 1 or 2')  WHERE (NOT (NEW.georectification IN (0, 1, 2)));  END  CREATE TRIGGER 'raster\_columns\_georectification\_update'  BEFORE UPDATE OF georectification ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''raster\_columns'' violates constraint: georectification must be 0, 1 or 2')  WHERE (NOT (NEW.georectification IN (0, 1, 2)));  END  CREATE TRIGGER 'raster\_columns\_compr\_qual\_factor\_insert'  BEFORE INSERT ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''raster\_columns'' violates constraint: compr\_qual\_factor < 1, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor < 1;  SELECT RAISE(ROLLBACK, 'insert on table ''raster\_columns'' violates constraint: compr\_qual\_factor > 100, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor > 100;  END  CREATE TRIGGER 'raster\_columns\_compr\_qual\_factor\_update'  BEFORE UPDATE OF compr\_qual\_factor ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''raster\_columns'' violates constraint: compr\_qual\_factor < 1, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor < 1;  SELECT RAISE(ROLLBACK, 'update on table ''raster\_columns'' violates constraint: compr\_qual\_factor > 100, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor > 100;  END |

Table 26 - EXAMPLE: raster\_columns INSERT Statement

|  |
| --- |
| INSERT INTO raster\_columns VALUES (  "sample\_matrix\_tiles",  "tile\_data",  90,  2,  4326) |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/rasters_tiles/raster_columns_table> | |
| REQ 48. | A GeoPackage shall include a raster\_columns table or updateable view that includes the columns and foreign key constraint defined in Table 23 and clause 10.2, and containing data described in clause 10.2. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| <http://www.opengis.net/spec/GPKG/1.0/req/rasters_tiles/raster_columns_table/triggers> | |
| REQ 49. | A GeoPackage shall include SQL triggers on the raster\_columns table or updateable view as defined in Table 24 and clause 10.2. |

|  |  |
| --- | --- |
| Requirement: Core | |
| <http://www.opengis.net/spec/GPKG/1.0/req/rasters_tiles/mime_types/core> | |
| REQ 50. | A GeoPackage shall support storage and use of MIME types image/jpeg [22][23][24]and image/png [25][26] as defined in clause 10.2. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/mime\_types/extension/ webp | |
| REQ 51. | A GeoPackage shall support storage and use of MIME type image/x-webp [27] as defined in clause 10.2 |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/mime\_types/extension/ geotiff | |
| REQ 52. | A GeoPackage shall support storage and use of MIME type image/tiff [28] for GeoTIFF images [30][31] as defined in clause 10.2 |

## 10.3 Tile Table Metadata

A GeoPackage shall contain a tile\_table\_metadata table or view as defined in this clause. The tile\_table\_metadata table or view shall contain one row record describing each tile table in a GeoPackage. The t\_table\_name column value shall be a row value of r\_table\_name in the raster\_columns table, enforced by a trigger. The is\_times\_two\_zoom column value shall be 1 if zoom level pixel sizes vary by powers of 2 between adjacent zoom levels in the corresponding tile table, or 0 if not.

NOTE1: A row record for a tile table must be inserted into this table before row records can be inserted into the tile\_matrix\_metadata table described in clause 10.4 due to the presence of foreign key and other integrity constraints on that table.

NOTE 2: GeoPackage applications that insert, update, or delete tiles (matrix set) table tiles row records are responsible for maintaining the corresponding descriptive contents of the tile\_table\_metadata table.

Table 27 -- tile\_table\_metadata

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: tile\_table\_metadata** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| t\_table\_name | text | {RasterLayerName}{\_tiles} | no |  | PK |
| is\_times\_two\_zoom | integer | Zoom level pixel sizes vary by powers of 2 (0=false,1=true) | no | 1 |  |

Table 28 -- tile\_table\_metadata Table Definition SQL

|  |
| --- |
| CREATE TABLE tile\_table\_metadata (  t\_table\_name TEXT NOT NULL PRIMARY KEY,  is\_times\_two\_zoom INTEGER NOT NULL DEFAULT 1  ) |

Table 29 -- tile\_table\_metadata Trigger Definition SQL

|  |
| --- |
| CREATE TRIGGER 'tile\_table\_metadata\_t\_table\_name\_insert'  BEFORE INSERT ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_table\_metadata'' violates constraint:t\_table\_name not in raster\_columns.r\_table\_name values')  WHERE NOT (NEW.t\_table\_name IN  (SELECT DISTINCT r\_table\_name FROM raster\_columns));  END  CREATE TRIGGER 'tile\_table\_metadata\_t\_table\_name\_update'  BEFORE UPDATE OF t\_table\_name ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_table\_metadata'' violates constraint:t\_table\_name not in raster\_columns.r\_table\_name values')  WHERE NOT (NEW.t\_table\_name IN  (SELECT DISTINCT r\_table\_name FROM raster\_columns));  END  CREATE TRIGGER 'tile\_table\_metadata\_is\_times\_two\_zoom\_insert'  BEFORE INSERT ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on tile\_table\_metadata violates constraint: is\_time\_two\_zoom must be one of 0|1')  WHERE NOT(NEW.is\_times\_two\_zoom IN (0,1));  END    CREATE TRIGGER 'tile\_table\_metadata\_is\_times\_two\_zoom\_update'  BEFORE UPDATE OF is\_times\_two\_zoom ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update of tile\_table\_metadata violates constraint: is\_time\_two\_zoom must be one of 0|1')  WHERE NOT(NEW.is\_times\_two\_zoom IN (0,1));  END |

Table 30 -- EXAMPLE: tile\_table\_metadata Insert Statement

|  |
| --- |
| INSERT INTO tile\_table\_metadata VALUES (  "sample\_matrix\_tiles",  1); |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_table\_metadata/table | |
| REQ 53. | A GeoPackage shall include a tile\_table\_metadata table as defined in clause 10.3 and table 28. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_table\_metadata/triggers | |
| REQ 54. | A GeoPackage shall have tile\_table\_metadata table triggers as defined in clause 10.3 and table 29. |

|  |  |
| --- | --- |
| Requirement:Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_table\_metadata/data | |
| REQ 55. | A GeoPackage tile\_table\_metadata table shall contain a row record for each tile table in the GPKG as specified in clause 10.3. |

## 10.4 Tile Matrix Metadata

A GeoPackage shall contain a tile\_matrix\_metadata table or view as defined in this clause. The tile\_matrix\_metadata table or view shall contain one row record for each zoom level that contains one or more tiles in each tiles table. It may contain row records for zoom levels in a tiles table that do not contain tiles. A tile\_matrix\_metadata row record shall be inserted for a zoom level for t\_table\_name before any tiles are inserted into the corresponding tiles table, so that triggers on that table specified in clause 10.5 below may reference tile\_matrix\_metadata column values for that zoom level to reject invalid data.

The tile\_matrix\_metadata table documents the structure of the tile matrix at each zoom level in each tiles table. It allows GeoPackages to contain rectangular as well as square tiles (e.g. for better representation of polar regions). It allows tile pyramids with zoom levels that differ in resolution by powers of 2, irregular intervals, or regular intervals other than powers of 2. When the value of the is\_times\_two\_zoom column in the tile\_table\_metadata record for a tiles table is 1 (true) then the pixel sizes for adjacent zoom levels in the tile\_matrix\_metadata table for that table shall only vary by powers of 2.

NOTE 1: Most tile pyramids have an origin at the upper left, a convention adopted by the OGC Web Map Tile Service (WMTS) [23], but some such as TMS [B11] used by MB-Tiles [B12] have an origin at the lower left. Most tile pyramids, such as Open Street Map [B14], OSMDroidAtlas [B15], and FalconView [B16] use a zoom\_out\_level of 0 for the smallest map scale “whole world” zoom level view, another convention adopted by WMTS, but some such as Big Planet Tracks [B17] invert this convention and use 0 or 1 for the largest map scale “local detail” zoom level view.

GeoPackages shall follow the most frequently used conventions of a tile origin at the upper left and a zoom-out-level of 0 for the smallest map scale “whole world” zoom level view, as specified by WMTS [23]. The tile coordinate (0,0) shall always refer to the tile in the upper left corner of the tile matrix at any zoom level, regardless of the actual availability of that tile. Pixel sizes for zoom levels sorted in ascending order shall be sorted in descending order.

GeoPackages shall not require that tiles be provided for level 0 or any other particular zoom level. This means that a tile matrix set can be sparse, i.e. not contain a tile for any particular position at a certain tile zoom level.This does not affect the spatial extent stated by the min/max x/y columns values in the geopackage\_contents record for the same t\_table\_name, or the tile matrix width and height at that level.

NOTE 2: GeoPackage applications may query the tile\_matrix\_metadata table or the tiles (matrix set) table specified in clause 10.7 below to determine the minimum and maximum zoom levels for a given tile matrix table.

NOTE 3: GeoPackage applications may query the tiles (matrix set) table to determine which tiles are available at each zoom level.

NOTE 4: GeoPackage applications that insert, update, or delete tiles (matrix set) table tiles row records are responsible for maintaining the corresponding descriptive contents of the tile\_matrix\_metadata table.

NOTE 5: The geopackage\_contents table (see clause 8.2 above) contains coordinates that define a bounding box as the stated spatial extent for all tiles in a tile (matrix set) table. If the geographic extent of the image data contained in these tiles is within but not equal to this bounding box, then the non-image area of matrix edge tiles must be padded with no-data values, preferably transparent ones.

Table 31 -- tile\_matrix\_metadata

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: tile\_matrix\_metadata** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| t\_table\_name | text | {RasterLayerName}\_tiles | no |  | PK, FK |
| zoom\_level | integer | 0 <= zoom\_level <= max\_level for t\_table\_name | no | 0 | PK |
| matrix\_width | integer | Number of columns (>= 1) in tile matrix at this zoom level | no | 1 |  |
| matrix\_height | integer | Number of rows (>= 1) in tile matrix at this zoom level | no | 1 |  |
| tile\_width | integer | Tile width in pixels (>= 1)for this zoom level | no | 256 |  |
| tile\_height | integer | Tile height in pixels (>= 1) for this zoom level | no | 256 |  |
| pixel\_x\_size | double | In t\_table\_name srid units or default meters for srid 0 (>0) | no | 1 |  |
| pixel\_y\_size | double | In t\_table\_name srid units or default meters for srid 0 (>0) | no | 1 |  |

Table 32 -- tile\_matrix\_metadata Table Creation SQL

|  |
| --- |
| CREATE TABLE tile\_matrix\_metadata (  t\_table\_name TEXT NOT NULL,  zoom\_level INTEGER NOT NULL,  matrix\_width INTEGER NOT NULL,  matrix\_height INTEGER NOT NULL,  tile\_width INTEGER NOT NULL,  tile\_height INTEGER NOT NULL,  pixel\_x\_size DOUBLE NOT NULL,  pixel\_y\_size DOUBLE NOT NULL,  CONSTRAINT pk\_ttm PRIMARY KEY (t\_table\_name, zoom\_level) ON CONFLICT ROLLBACK,  CONSTRAINT fk\_ttm\_t\_table\_name FOREIGN KEY (t\_table\_name) REFERENCES tile\_table\_metadata(t\_table\_name)) |

Table 33 -- tile\_matrix\_metadata Trigger Definition SQL

|  |
| --- |
| CREATE TRIGGER 'tile\_matrix\_metadata\_zoom\_level\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:zoom\_level cannot be less than 0')  WHERE (NEW.zoom\_level < 0);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_zoom\_level\_update'  BEFORE UPDATE of zoom\_level ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:zoom\_level cannot be less than 0')  WHERE (NEW.zoom\_level < 0);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_width\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:matrix\_width cannot be less than 1')  WHERE (NEW.matrix\_width < 1);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_width\_update'  BEFORE UPDATE OF matrix\_width ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:matrix\_width cannot be less than 1')  WHERE (NEW.matrix\_width < 1);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_height\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:matrix\_height cannot be less than 1')  WHERE (NEW.matrix\_height < 1);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_height\_update'  BEFORE UPDATE OF matrix\_height ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:matrix\_height cannot be less than 1')  WHERE (NEW.matrix\_height < 1);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_x\_size\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:pixel\_x\_size must be greater than 0')  WHERE NOT (NEW.pixel\_x\_size > 0);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_x\_size\_update'  BEFORE UPDATE OF pixel\_x\_size ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:pixel\_x\_size must be greater than 0')  WHERE NOT (NEW.pixel\_x\_size > 0);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_y\_size\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:pixel\_y\_size must be greater than 0')  WHERE NOT (NEW.pixel\_y\_size > 0);  END  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_y\_size\_update'  BEFORE UPDATE OF pixel\_y\_size ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:pixel\_y\_size must be greater than 0')  WHERE NOT (NEW.pixel\_y\_size > 0);  END |

Table 34 -- EXAMPLE: tile\_matrix\_metadata Insert Statement

|  |
| --- |
| INSERT INTO tile\_matrix\_metadata VALUES (  "sample\_matrix\_tiles",  0,  1,  1,  512,  512,  2.0,  2.0) |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_metadata/table | |
| REQ 56. | A GeoPackage shall include a tile\_matrix\_metadata table as defined in clause 10.4 and table 32. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_metadata/triggers | |
| REQ 57. | A GeoPackage shall have triggers on the tile\_matrix\_metadata table as defined in clause 10.4 and table 33. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_metadata/data | |
| REQ 58. | A GeoPackage tile\_matrix\_metadata table shall contain one row record for each zoom level that contains one or more tiles in each tiles table. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_zoom\_levels/powers\_of\_two | |
| REQ 59. | A GeoPackage shall support tile matrix set zoom levels for pixel sizes that differ by powers of two between adjacent zoom levels. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_zoom\_levels/other\_intervals | |
| REQ 60. | A GeoPackage shall support tile matrix set zoom levels for pixel sizes that differ by irregular intervals or by regular intervals other than powers of two between adjacent zoom levels. |

## 10.5 Tiles Table

Tiles in a tile matrix set with one or more zoom levels shall be stored in a GeoPackage in a tiles table or view with a unique name for every different tile matrix set in the GeoPackage. Each tiles table or view shall be defined with the columns described in table 29 below. Each tiles table or view shall contain tile matrices at one or more zoom levels of different spatial resolution (map scale). All tiles at a particular zoom level must have the same pixel\_x\_size and pixel\_y\_size values specified in the tile\_matrix\_metadata row record for that tiles table and zoom level.

When the value of the is\_times\_two\_zoom column in the tile\_table\_metadata record for a tiles table row is 1 (true) then the pixel sizes for adjacent zoom levels in the tiles table shall only vary by powers of 2.

NOTE 1: The id primary key allows tiles table views to be created on RasterLite [B13] version 1 raster table implementations, where the tiles are selected based on a spatially indexed bounding box in a separate metadata table.

NOTE 2: The zoom\_level / tile\_column / tile\_row unique key allows tiles to be selected and accessed by “z, x, y”, a common convention used by MB-Tiles [B12], Big Planet [B17], and other implementations. In a SQLite implementation this unique key is automatically indexed. This table / view definition may also follow RasterLite [B13] version 1 conventions, where the tiles are selected based on a spatially indexed bounding box in a separate metadata table.

GeoPackages shall implement appropriate SQL triggers on each tiles table by executing the add\_tile\_triggers() routine specified in clause 9.8 below with the tiles table as a parameter value to ensure that

1. The zoom\_level value is specified for the tiles table in the tile\_matrix\_metadata table
2. The tile column value is between 0 and the matrix\_height specified for the zoom\_level in the tile\_matrix\_metadata table
3. The tile \_row value is between 0 and the matrix\_width specified for the zoom\_level in the tile\_matrix\_metadata table

Table 35 – tiles table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: {TilesTableName} tiles table** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| id | integer | Autoincrement primary key | no |  | PK |
| zoom\_level | integer | min(zoom\_level) <= zoom\_level <= max(zoom\_level) for t\_table\_name | no | 0 | UK |
| tile\_column | integer | 0 to tile\_matrix\_metadata matrix\_width – 1 | no | 0 | UK |
| tile\_row | integer | 0 to tile\_matrix\_metadata matrix\_height - 1 | no | 0 | UK |
| tile\_data | BLOB | Of type raster\_table\_metadata format | no |  |  |

Table 36 -- EXAMPLE: tiles table Create Table SQL

|  |
| --- |
| CREATE TABLE sample\_matrix\_tiles (  id INTEGER PRIMARY KEY AUTOINCREMENT,  zoom\_level INTEGER NOT NULL DEFAULT 0,  tile\_column INTEGER NOT NULL DEFAULT 0,  tile\_row INTEGER NOT NULL DEFAULT 0,  tile\_data BLOB NOT NULL DEFAULT (zeroblob(4)),  UNIQUE (zoom\_level, tile\_column, tile\_row)) |

NOTE 3: zeroblob(n) is an SQLite function.

Table 37 – EXAMPLE: tiles table Trigger Definition SQL

|  |
| --- |
| SELECT add\_tile\_triggers(‘sample\_matrix\_tiles’)  /\* creates the following triggers \*/  CREATE TRIGGER "sample\_matrix\_tiles\_zoom\_insert"  BEFORE INSERT ON "sample\_matrix\_tiles"  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_matrix\_tiles'' violates constraint: zoom\_level not specified for table in tile\_matrix\_metadata')  WHERE NOT (NEW.zoom\_level IN (SELECT zoom\_level FROM tile\_matrix\_metadata WHERE t\_table\_name = 'sample\_matrix\_tiles')) ;  END  CREATE TRIGGER "sample\_matrix\_tiles\_zoom\_update"  BEFORE UPDATE OF zoom\_level ON "sample\_matrix\_tiles"  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''sample\_matrix\_tiles'' violates constraint: zoom\_level not specified for table in tile\_matrix\_metadata')  WHERE NOT (NEW.zoom\_level IN (SELECT zoom\_level FROM tile\_matrix\_metadata WHERE t\_table\_name = 'sample\_matrix\_tiles')) ;  END  CREATE TRIGGER "sample\_matrix\_tiles\_tile\_column\_insert"  BEFORE INSERT ON "sample\_matrix\_tiles"  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_matrix\_tiles'' violates constraint: tile\_column cannot be < 0')  WHERE (NEW.tile\_column < 0) ;  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_matrix\_tiles'' violates constraint: tile\_column must by < matrix\_width specified for table and zoom level in tile\_matrix\_metadata')  WHERE NOT (NEW.tile\_column < (SELECT matrix\_width FROM tile\_matrix\_metadata WHERE t\_table\_name = 'sample\_matrix\_tiles' AND zoom\_level = NEW.zoom\_level));  END  CREATE TRIGGER "sample\_matrix\_tiles\_tile\_column\_update"  BEFORE UPDATE OF tile\_column ON "sample\_matrix\_tiles"  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''sample\_matrix\_tiles'' violates constraint: tile\_column cannot be < 0')  WHERE (NEW.tile\_column < 0) ;  SELECT RAISE(ROLLBACK, 'update on table ''sample\_matrix\_tiles'' violates constraint: tile\_column must by < matrix\_width specified for table and zoom level in tile\_matrix\_metadata')  WHERE NOT (NEW.tile\_column < (SELECT matrix\_width FROM tile\_matrix\_metadata WHERE t\_table\_name = 'sample\_matrix\_tiles' AND zoom\_level = NEW.zoom\_level));  END  CREATE TRIGGER "sample\_matrix\_tiles\_tile\_row\_insert"  BEFORE INSERT ON "sample\_matrix\_tiles"  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_matrix\_tiles'' violates constraint: tile\_row cannot be < 0')  WHERE (NEW.tile\_row < 0) ;  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_matrix\_tiles'' violates constraint: tile\_row must by < matrix\_height specified for table and zoom level in tile\_matrix\_metadata')  WHERE NOT (NEW.tile\_row < (SELECT matrix\_height FROM tile\_matrix\_metadata WHERE t\_table\_name = 'sample\_matrix\_tiles' AND zoom\_level = NEW.zoom\_level));  END  CREATE TRIGGER "sample\_matrix\_tiles\_tile\_row\_update"  BEFORE UPDATE OF tile\_row ON "sample\_matrix\_tiles"  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''sample\_matrix\_tiles'' violates constraint: tile\_row cannot be < 0')  WHERE (NEW.tile\_row < 0) ;  SELECT RAISE(ROLLBACK, 'update on table ''sample\_matrix\_tiles'' violates constraint: tile\_row must by < matrix\_height specified for table and zoom level in tile\_matrix\_metadata')  WHERE NOT (NEW.tile\_row < (SELECT matrix\_height FROM tile\_matrix\_metadata WHERE t\_table\_name = 'sample\_matrix\_tiles' AND zoom\_level = NEW.zoom\_level));  END |

Table 38 -- EXAMPLE: tiles table Insert Statement

|  |
| --- |
| INSERT INTO sample\_matrix\_tiles VALUES (  1,  1,  1,  1,  "BLOB VALUE") |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tiles\_table/table | |
| REQ 61. | All tile matrix sets in a GeoPackage shall be contained in tiles tables defined as specified in clause 10.5 and table 33, and exemplified by table 36. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tiles\_table/raster\_column | |
| REQ 62. | All tile table tile\_data raster columns in a GeoPackage shall be defined with a BLOB data type that is an image mime type as specified in clause 10.2. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tiles\_table/triggers | |
| REQ 63. | All tile matrix set tables in a GeoPackage shall have triggers defined by executing the add\_tile\_triggers() routine specified in clause 10.10, as exemplified by table 37. |

## 10.6 Raster Tables

Raster tables have raster columns defined as BLOB data types that contain rasters that are not part of tile matrix sets. Every table in a GeoPackage that is not a tiles table as described in clause 9.5 and that includes one or more raster columns is a raster table. Raster tables are also feature tables as specified in clause 8.4 above that may or may not have geometry columns in addition to raster columns.

Every raster table in a GeoPackage shall have a primary key defined on one or more columns so that row level metadata records may be linked to the rasters in it by rowid as described in clauses 10.7 and 11.3 below.

NOTE1: An integer column primary key is recommended for best performance.

Table 39 -- EXAMPLE: sample\_rasters Table or View

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: sample\_rasters** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| id | integer | Autoincrement primary key | no |  | PK |
| elevation | BLOB | Elevation coverage;  of type raster\_format\_metadata.mime\_type | no |  |  |
| description | text | Description of the area | no | ‘no desc’ |  |
| photo | BLOB | Photograph of the area;  of type raster\_format\_metadata.mime\_type | no |  |  |

Table 40 -- EXAMPLE: sample\_rasters Table Definition SQL

|  |
| --- |
| CREATE TABLE sample\_rasters (  id INTEGER PRIMARY KEY AUTOINCREMENT,  elevation BLOB NOT NULL,  description TEXT NOT NULL DEFAULT 'no\_desc',  photo BLOB NOT NULL) |

NOTE2: The sample\_rasters table created in Table 40 above could be extended with one or more geometry columns by calls to the addGeometryColumn() rountine specified in clause 9.4 to have both raster and geometry columns like the sample\_feature\_table shown in Figure3: GeoPackageTables above in clause 7.

Table 41 -- EXAMPLE: sample\_rasters Insert Statement

|  |
| --- |
| INSERT INTO sample\_rasters VALUES (  1,  {Elevation Raster},  'rough terrain',  {Area Photo} ) |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/raster\_table/table | |
| REQ 64. | All raster images in a GeoPackage that are not tiles in a tiles table shall be contained in rasters tables that are defined as specified by clause 10.6 and exemplified by tables 39 and 40. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/raster\_table/primary\_key | |
| REQ 65. | Every raster table in a GeoPackage shall have a primary key defined on one or more columns as specified by clause 10.6. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/raster\_table/raster\_column | |
| REQ 66. | All raster table raster columns in a GeoPackage shall be defined with a BLOB data type that is an image mime type as specified in clause 10.2. |

## 10.7 Rasters or Tiles Table Metadata

There shall be a {Raster|Tile TableName}\_rt\_metadata table or view for each rasters or tiles table in a GeoPackage defined with the columns described in table 42 below.

NOTE 1: This table naming convention is adopted from RasterLite [B13].

The data in a row record in this table refers to the raster in the r\_raster\_column column in the {Raster|Tile TableName}table for the record with a rowed equal to the row\_id\_value primary key column value.

NOTE2: In an SQLite implementation, the rowid value is always equal to the value of a single-column primary key on an integer column [B30]. Althought not stated in the SQLite documentation, testing has not revealed a case where rowed values on a table with any primry key column(s) defined are changed by a database reorganization performed by the VACUUM SQL command.

The compr\_qual\_ factor column value indicates the image quality of that raster on a scale from 1 (lowest) to 100 (highest) for rasters compressed with a lossy compression algorithm. It is always 100 for rasters compressed with a lossless compression algorithm, or with no compression.

The georectification column value indicates whether or not that raster is georectified to an area on the earth. A value of 0 indicates that the raster is not georectified. A value of 1 indicates that the raster is georectified (but not necessarily orthorectified). A value of 2 indicates that the raster is orthorectified (which implies georectified) to accurately align with real world coordinates, have constant scale, and support direct measurement of distances, angles, and areas

For a georectified raster (i.e. georectification is 1 or 2), the min\_x, min\_y, max\_x and max\_y column values define a bounding box that shall be the spatial extent of the area on the earth represented by the raster.

NOTE 3: This data structure can be implemented as a table in the absence of geometry data types or spatial indexes. When implemented as a view, the min/max x/y columns could reference ordinates of a bounding box geometry in an underlying table when geometry data types are available, e.g. in RasterLite [B13].

Table 42 -- {RasterLayerName}\_rt\_metadata

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: {RasterLayerName}\_rt\_metadata** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| row\_id\_value | integer | rowid in rasters or tiles table | no |  | PK, |
| r\_raster\_column | text | “tile\_data” for a tiles table, or the name of a raster column for a rasters table | no | raster\_column\_name | PK |
| compr\_qual\_factor | integer | Compression quality factor: 1 (lowest) to 100 (highest) for lossy compression; always 100 for lossless or no compression. | no | 100 |  |
| georectification | integer | Is the raster georectified; 0=not georectified, 1=georectified, 2=orthorectified | no | 0 |  |
| min\_x | double | In raster\_columns.srid | no | -180.0 |  |
| min\_y | double | In raster\_columns.srid | no | -90.0 |  |
| max\_x | double | In raster\_columns.srid | no | 180.0 |  |
| max\_y | double | In raster\_columns.srid | no | 90.0 |  |

Table 43 -- EXAMPLE: {RasterLayerName}\_rt\_metadata Table Definition SQL

|  |
| --- |
| CREATE TABLE sample\_matrix\_tiles\_rt\_metadata (  row\_id\_value INTEGER NOT NULL,  r\_raster\_column TEXT NOT NULL DEFAULT 'tile\_data',  compr\_qual\_factor INTEGER NOT NULL DEFAULT 100,  georectification INTEGER NOT NULL DEFAULT 0,  min\_x DOUBLE NOT NULL DEFAULT -180.0,  min\_y DOUBLE NOT NULL DEFAULT -90.0,  max\_x DOUBLE NOT NULL DEFAULT 180.0,  max\_y DOUBLE NOT NULL DEFAULT 90.0,  CONSTRAINT pk\_smt\_rm PRIMARY KEY (row\_id\_value, r\_raster\_column)  ON CONFLICT ROLLBACK  ) |

Table 44 -- EXAMPLE: {RasterLayerName}\_rt\_metadata Trigger Definition SQL

|  |
| --- |
| SELECT add\_rt\_metadata\_triggers('sample\_rasters')  /\* creates the following triggers \*/  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_r\_raster\_column\_insert'  BEFORE INSERT ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_rasters\_rt\_metadata'' violates constraint: r\_raster\_column must be specified for table sample\_rasters in table raster\_columns')  WHERE (NOT (NEW.r\_raster\_column IN (SELECT DISTINCT r\_raster\_column FROM raster\_columns WHERE r\_table\_name = 'sample\_rasters')));  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_r\_raster\_column\_update'  BEFORE UPDATE OF r\_raster\_column ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_rasters\_rt\_metadata'' violates constraint: r\_raster\_column must be specified for table sample\_rasters in table raster\_columns')  WHERE (NOT (NEW.r\_raster\_column IN (SELECT DISTINCT r\_raster\_column FROM raster\_columns WHERE r\_table\_name = 'sample\_rasters')));  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_georectification\_insert'  BEFORE INSERT ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_rasters\_rt\_metadata'' violates constraint: georectification must be 0, 1 or 2')  WHERE (NOT (NEW.georectification IN (0, 1, 2)));  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_georectification\_update'  BEFORE UPDATE OF georectification ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''sample\_rasters\_rt\_metadata'' violates constraint: georectification must be 0, 1 or 2')  WHERE (NOT (NEW.georectification IN (0, 1, 2)));  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_compr\_qual\_factor\_insert'  BEFORE INSERT ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_rasters\_rt\_metadata'' violates constraint: compr\_qual\_factor < 1, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor < 1;  SELECT RAISE(ROLLBACK, 'insert on table ''sample\_rasters\_rt\_metadata'' violates constraint: compr\_qual\_factor > 100, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor > 100;  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_compr\_qual\_factor\_update'  BEFORE UPDATE OF compr\_qual\_factor ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''sample\_rasters\_rt\_metadata'' violates constraint: compr\_qual\_factor < 1, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor < 1;  SELECT RAISE(ROLLBACK, 'update on table ''sample\_rasters\_rt\_metadata'' violates constraint: compr\_qual\_factor > 100, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor > 100;  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_row\_id\_value\_insert'  BEFORE INSERT ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table sample\_rasters\_rt\_metadata violates constraint: row\_id\_value must exist in sample\_rasters table')  WHERE NOT EXISTS (SELECT rowid  FROM 'sample\_rasters' WHERE rowid = NEW.row\_id\_value);  END  CREATE TRIGGER 'sample\_rasters\_rt\_metadata\_row\_id\_value\_update'  BEFORE UPDATE OF 'row\_id\_value' ON 'sample\_rasters\_rt\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table sample\_rasters\_rt\_metadata violates constraint: row\_id\_value must exist in sample\_rasters table')  WHERE NOT EXISTS (SELECT rowid  FROM 'sample\_rasters' WHERE rowid = NEW.row\_id\_value);  END |

Table 45 -- EXAMPLE: {RasterLayerName}\_rt\_metadata Insert Statement

|  |
| --- |
| INSERT INTO sample\_matrix\_tiles\_rt\_metadata VALUES (  1,  "tile\_data",  1,  -77.0,  38.0,  -75.0,  40.0,  100,  ) |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/rt\_metadata\_table | |
| REQ 67. | There shall be a {Raster|Tile TableName}{\_rt\_metadata} table as specified in clause 10.7 with the columns described in table 41 and exemplified by table 42 for every tile and raster table in a GeoPackage. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/rt\_metadata\_table/data | |
| REQ 68. | Each {Raster|Tile TableName}{\_rt\_metadata} table specified in clause 10.7 for every tile and raster table in a GeoPackage shall have a row record describing each raster and tile in a GeoPackage. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/rt\_metadata\_table/triggers | |
| REQ 69. | All raster table raster columns in a GeoPackage shall have triggers defined by executing the add\_rt\_metadata\_triggers() routine specified in clause 10.8, as exemplified by table 44. |

## 10.8 Image Routines SQL API

GeoPackages shall provide the following image routine support for rasters and tiles.

Table 46 -- Image Routines

|  |  |  |
| --- | --- | --- |
| **Routine** | **Description** | **Clause** |
| gpkgAddTileTriggers(  tileTableName String) : | Adds triggers to a tiles table. See example in table 37 above. | 10.8.1 |
| gpkgAddRtMetadataTriggers (  rasterOrTileMetadataTableName String) : | Adds triggers to a {Raster|Tile TableName}{\_rt\_metadata} table.  See example in table 44 above. | 10.8.2 |
| gpkgGetNormalZoom(  tileTableName String,  invertedZoomLevel Integer) : Integer | Returns normal zoom level for specified table, converting between inverted convention (0 = most detailed) and normal convention (0 = whole world) | 10.8.3 |
| gpkgGetNormalRow(  tileTableName String,  normalZoomLevel Integer,  invertedRowNumber Integer) : Integer | Returns normal row for specified table, converting between inverted origin convention (lower left) to normal origin convention (upper left) | 10.8.4 |
| gpkgGetImageType(image BLOB) : String | Returns the name of the image type, or “unknown” | 10.8.5 |
| isJpegBlob (image BLOB) : Integer | Returns 1 (true) if the BLOB contains a JPEG image | 10.8.6 |
| isPngBlob (image BLOB) : Integer | Returns 1 (true) if the BLOB contains a PNG image | 10.8.7 |
| isTiffBlob (image BLOB) : Integer | Returns 1 (true) if the BLOB contains a TIFF image | 10.8.8 |
| isWebpBlob (image BLOB) : Integer | Returns 1 (true) if the BLOB contains a WebP image | 10.8.9 |
| gpkgBboxToTiles (  tileTableName String,  srid Integer,  minX Real,  maxX Real,  minY Real,  maxY Real,  zoomLevel Integer) : set{Integer} | Returns tiles from tile matrix for the specified srid bounding box and zoom level | 10.8.10 |
| gpkgPointToTile (  tileTableName String,  srid Integer,  x Real,  y Real,  zoomLevel Integer) : BLOB | Returns tile from tile matrix for specified srid, point and zoom level | 10.8.11 |

### 10.8.1 gpkgAddTileTriggers SQL Function

The SQL function gpkgAddTileTriggers shall take one argument, a string that specifies the tiles table name to which triggers are to be added.

The triggers added to the specified tiles table shall be as exemplified in table 37.

The gpkgAddTileTriggers SQL function shall return nothing if triggers are successfully added, or gpkgAddTileTriggers shall raise a SQL exception for any error condition, including an incorrect argument type or incorrect argument value (e.g. tiles table not being present).

### 10.8.2 gpkgAddRtMetadataTriggers SQL Function

The SQL function gpkgAddRtMetadataTriggers shall take one argument, a string that specifies the name of a rasters or tiles metadata table (e.g. {Raster|Tile TableName}{\_rt\_metadata}), to which triggers are to be added.

The triggers added to the specified raster or tile metadata table shall be as exemplified in table 44.

The gpkgAddRtMetadataTriggers SQL function shall return nothing if triggers are successfully added, or gpkgAddRtMetadataTriggers shall raise a SQL exception for any error condition, including an incorrect argument type or incorrect argument value (e.g. raster or tile metadata table not being present).

### 10.8.3 gpkgGetNormalZoom SQL Function

The gpkgGetNormalZoom SQL function shall take two arguments. The first argument shall be a string that specifies the tiles table to operate on. The second argument shall be an integer that specifies a zoom level (nominally in inverted convention where level 0 zoom is the most detailed).

The gpkgGetNormalZoom SQL function shall return a zoom level (nominally in normal convention, where level 0 zoom is the “world” level), or raise a SQL exception for any error conditions, including a incorrect argument type or incorrect argument value (e.g. tiles table not being found, or the zoom argument is outside of the zoom levels in the tile\_matrix\_metadata table for the tiles table).

Note: this function is symmetric, so it can also be used to convert from normal zoom level to inverted zoom convention.

### 10.8.4 gpkgGetNormalRow SQL Function

The gpkgGetNormalRow SQL function shall take three arguments. The first argument shall be a string that specifies the tiles table to operate on. The second argument shall be an integer that specifies the zoom level to operate on (in normal convention, where level 0 zoom is the “world” level). The third argument shall be an integer that specifies a row number (nominally in inverted convention where the origin is at the lower left corner).

The gpkgGetNormalRow SQL function shall return a row number (nominally in normal convention, where the origin is in the top left corner), or raise a SQL exception for any error conditions, including an incorrect argument type or incorrect argument value (e.g. tiles table not being found, zoom level outside of the zoom levels levels in the tile\_matrix\_metadata table for the tiles table, or the row number outside of the rows for the zoom level).

Note: this function is symmetric, so it can also be used to convert from normal row numbers to inverted row number convention.

### 10.8.5 gpkgGetImageType SQL Function

The gpkgGetImageType SQL function shall take one argument, a BLOB that may contain an image.

The gpkgGetImageType SQL function shall return the name of the image type, which shall be the suffix of the MIME[27] type of the image following its “image/” prefix, or “unknown” if the image type cannot be determined, or is not one of the “jpeg”, “png”, “tiff”, or “x-webp” types supported by the GeoPackage implementation. It shall raise a SQL exception if the input is not of the correct type.

Note: the following four isTypeBlob SQL functions provide additional interfaces to the results of this SQL function.

### 10.8.6 isJpegBlob SQL Function

The isJpegBlob SQL function shall take one argument, a BLOB that may contain an image.

The isJpegBlob SQL function shall return 1 if the BLOB contains a JPEG image, or 0 if the BLOB is not a valid JPEG image. This function does not need to fully evaluate the BLOB – it is sufficient to do a simple “magic number” check of the header. It shall raise a SQL exception if the input is not of the correct type.

### 10.8.7 isPngBlob SQL Function

The isPngBlob SQL function shall take one argument, a BLOB that may contain an image.

The isPngBlob SQL function shall return 1 if the BLOB contains a PNG image, or 0 if the BLOB is not a valid PNG image. This function does not need to fully evaluate the BLOB – it is sufficient to do a simple “magic number” check of the header. It shall raise a SQL exception if the input is not of the correct type.

### 10.8.8 isTiffBlob SQL Function

The isTiffBlob SQL function shall take one argument, a BLOB that may contain an image.

The isTiffBlob SQL function shall return 1 if the BLOB contains a TIFF image, or 0 if the BLOB is not a valid TIFF image. This function does not need to fully evaluate the BLOB – it is sufficient to do a simple “magic number” check of the header. It shall raise a SQL exception if the input is not of the correct type.

### 10.8.9 isWebpBlob SQL Function

The isWebpBlob SQL function shall take one argument, a BLOB that may contain an image.

The isWebpBlob SQL function shall return 1 if the BLOB contains a WebP image, or 0 if the BLOB is not a valid WebP image. This function does not need to fully evaluate the BLOB – it is sufficient to do a simple “magic number” check of the header. It shall raise a SQL exception if the input is not of the correct type.

### 10.8.10 gpkgBboxToTiles SQL Function

The gpkgBboxToTiles SQL function shall take seven arguments. The first argument shall be a string specifying the tiles table to work on. The second argument shall be an integer identifier that specifies a spatial reference system. The third argument shall be a floating point number that specifies the minimum X extent (longitude or easting) relative to the specified spatial reference system. The fourth argument shall be a floating point number that specifies the minimum Y extent (latitude or northing) relative to the specified spatial reference system. The fifth argument shall be a floating point number that specifies the maximum X extent (longitude or easting) relative to the specified spatial reference system. The sixth argument shall be a floating point number that specifies the maximum Y extent (latitude or northing) relative to the specified spatial reference system. The seventh argument shall be an integer that specifies the required zoom level.

The gpkgBboxToTiles SQL function shall return a set of integers, being the primary key values for the tiles that match the input arguments. gpkgBboxToTiles shall raise a SQL exception for incorrect input types or if the tiles table is not found. It shall not raise a SQL exception for out-of-range zoom levels, out-of-range extents, or if no tiles are found; these conditions shall result in return of an empty set.

### 10.8.11 gpkgPointToTile SQL Function

The gpkgPointToTile SQL function shall take five arguments. The first argument shall be a string specifying the tiles table to work on. The second argument shall be an integer identifier that specifies a spatial reference system. The third argument shall be a floating point number that specifies the X coordinate (longitude or easting) relative to the specified spatial reference system. The fourth argument shall be a floating point number that specifies the Y coordinate (latitude or northing) relative to the specified spatial reference system. The fifth argument shall be an integer that specifies the required zoom level.

The gpkgPointToTile SQL function shall return a single tile (image BLOB) that matches the input parameters. Where more than one tile matches (e.g. overlapping tiles, or an edge), any matching tile may be returned. . The gpkgPointToTile SQL function shall raise a SQL exception for incorrect input types or if the tiles table is not found. It shall not raise a SQL exception for out-of-range zoom levels, out-of-range coordinates, or if no tiles are found; these conditions shall result in return of an empty set.

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/ rasters\_tiles /image\_routines\_sql\_api | |
| REQ 70 | Each GeoPackage shall provide the image routines defined in clause 10.8 and table 46. |

# 11 Metadata

## 11.1 Introduction

Two tables in a GeoPackage provide a means of storing metadata in the form of XML documents that are defined in accordance with any authoritative metadata specifications, and relating it to the features, rasters, and tiles data in a GeoPackage. These tables are intended to provide the support necessary to implement the hierarchical metadata model defined in ISO 19115 [42], Annex B B.5.25 MD\_ScopeCode, Annex G and Annex H, so that as GeoPackage data is captured and updated, the most local and specific detailed metadata changes associated with the new or modified data may be captured separately, and referenced to existing global and general metadata.

The xml\_metadata table that contains metadata is described in clause 11.2, and the metadata\_reference table that relates metadata to GeoPackage data is described in clause 11.3. These tables shall be defined in every GeoPackage and shall be used to store metadata defined in accordance with authoritative metadata specifications external to this GeoPackage specification.

There is no GeoPackage requirement that such metadata be provided or that defined metadata be structured in a hierarchial fashion with more than one level, only that if it is, these tables are to be used.

NOTE: Informative examples of hierarchical metadata are provided in Annex C.

## 11.2 XML Metadata Table

The first component of GeoPackage metadata is the xml\_metadata table that contains metadata in XML documents or document fragments. This table may contain metadata structured in accordance with any authoritative metadata specification, such as ISO 19115 [42], ISO 19115-2 [B31], ISO 19139 [B32], Dublin Core [B33], CSDGM [B34], DDMS B35], NMF/NMIS [B36], etc. The GeoPackage interpretation of what constitutes “metadata” is a broad one that includes UML models [B37] encoded in XMI [B38], GML Application Schemas [B39], ISO 19110 feature catalogues [B40], OWL [B41] and SKOS [B42]taxonomies, etc.

Table 47 - xml\_metadata

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: xml\_metadata** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| id | integer | Metadata primary key | no |  | PK |
| md\_scope | text | Name of the data scope to which this metadata applies; see table 42 below | no | ‘undefined’ |  |
| md\_standard\_uri | text | URI reference to the metadata structure definition authority | no | <http://schemas.opengis.net/iso/19139/> |  |
| metadata | text | XML metadata document or fragment | no | ’’ |  |

The md\_scope column in the xml\_metadata table is the name of the applicable scope for the contents of the metadata column for a given row. The list of valid scope names and their definitions is provided in table 43 below. The intial contents of this table were obtained from the ISO 19115 [42], Annex B B.5.25 MD\_ScopeCode code list, which was extended for use in the GeoPackage specification by addition of entries with “NA” as the scope code column in table 47.

Table 48 Metadata Scopes

|  |  |  |
| --- | --- | --- |
| **Name (md\_scope)** | **Scope Code** | **Definition** |
| undefined | NA | Metadata information scope is undefined |
| fieldSession | 012 | Information applies to the field session |
| collectionSession | 004 | Information applies to the collection session |
| series | 006 | Information applies to the (dataset) series |
| dataset | 005 | Information applies to the (geographic feature) dataset |
| featureType | 010 | Information applies to a feature type (class) |
| feature | 009 | Information applies to a feature (instance) |
| attributeType | 002 | Information applies to the attribute class |
| attribute | 001 | Information applies to the characteristic of a feature (instance) |
| tile | 016 | Information applies to a tile, a spatial subset of geographic data |
| model | 015 | Information applies to a copy or imitation of an existing or hypothetical object |
| catalog | NA | Metadata applies to a feature catalog |
| schema | NA | Metadata applies to an application schema |
| taxonomy | NA | Metadata applies to a taxonomy or knowledge system |
| software | 013 | Information applies to a computer program or routine |
| service | 014 | Information applies to a capabilitiy which a service provider entity makes available to a service user entity through a set of interfaces that define a behaviour, such as a use case |
| collectionHardware | 003 | Information applies to the collection hardware class |
| nonGeographicDataset | 007 | Information applies to non-geographic data |
| dimensionGroup | 008 | Information applies to a dimension group |

NOTE1: The scope codes in Table 48 include a very wide set of descriptive information types as “metadata” to describe data.

NOTE2: The “catalog” md\_scope may be used for Feature Catalog [B40] information stored as XML metadata that is linked to features stored in a GeoPackage.

NOTE3: The “schema” md\_scope may be used for Application Schema [B37][B38][B39][B45] information stored as XML metadata that is linked to features stored in a GeoPackage.

NOTE4: The “taxonomy” md\_scope may be used for taxonomy or knowledge system [B41][B42] “linked data” information stored as XML metadata that is linked to features stored in a GeoPackage.

The xml\_metadata table shall be created using the SQL shown in Table 49.

Table 49 - xml\_metadata Table Definition SQL

|  |
| --- |
| CREATE TABLE xml\_metadata (  id INTEGER CONSTRAINT xm\_pk PRIMARY KEY ASC  ON CONFLICT ROLLBACK AUTOINCREMENT NOT NULL UNIQUE,  md\_scope TEXT NOT NULL DEFAULT 'dataset',  metadata\_standard\_URI TEXT NOT NULL DEFAULT 'http://schemas.opengis.net/iso/19139/',  metadata BLOB NOT NULL DEFAULT (zeroblob(4))  ) |

The xml\_metadata table md\_scope column values shall be enforced using triggers defined using the SQL shown in Table 50.

Table 50 xml\_metadata Trigger Definition SQL

|  |
| --- |
| CREATE TRIGGER 'xml\_metadata\_md\_scope\_insert'  BEFORE INSERT ON 'xml\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table xml\_metadata violates constraint: md\_scope must be one of undefined | fieldSession | collectionSession | series | dataset | featureType | feature | attributeType | attribute | tile | model | catalogue | schema | taxonomy software | service | collectionHardware | nonGeographicDataset | dimensionGroup')  WHERE NOT(NEW.md\_scope IN ('undefined','fieldSession','collectionSession','series','dataset',  'featureType','feature','attributeType','attribute','tile','model',  'catalogue','schema','taxonomy','software','service',  'collectionHardware','nonGeographicDataset','dimensionGroup'));  END  CREATE TRIGGER 'xml\_metadata\_md\_scope\_update'  BEFORE UPDATE OF 'md\_scope' ON 'xml\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table xml\_metadata violates constraint: md\_scope must be one of undefined | fieldSession | collectionSession | series | dataset | featureType | feature | attributeType | attribute | tile | model | catalogue | schema | taxonomy software | service | collectionHardware | nonGeographicDataset | dimensionGroup')  WHERE NOT(NEW.md\_scope IN ('undefined','fieldSession','collectionSession','series','dataset',  'featureType','feature','attributeType','attribute','tile','model',  'catalogue','schema','taxonomy','software','service',  'collectionHardware','nonGeographicDataset','dimensionGroup'));  END |

The xml\_metadata table shall contain at least the row defined by the SQL insert statement show in Table 51.

Table 51 -- Required xml\_metadata SQL Insert statement

|  |
| --- |
| INSERT INTO xml\_metadata VALUES (  0, 'undefined', 'http://schemas.opengis.net/iso/19139/', (zeroblob(4))) |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/metadata/xml\_metadata\_table | |
| REQ 71. | There shall be an xml\_metadata table in a GeoPackage as specified in clause 11.1 with the columns described in table 47 and defined by SQL in table 49. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/metadata/xml\_metadata\_table/triggers | |
| REQ 72. | The xml\_metadata table in a GeoPackage shall have the triggers specified in clause 11.1, as defined by the SQL in table 50 to enforce the use of the md\_scope names in table 48. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/metadata/xml\_metadata\_table/undefined | |
| REQ 73. | The xml\_metadata table in a GeoPackage shall contain one row record for the ‘undefined’ scope as defined by the SQL in table 51. |

## 11.3 Metadata Reference Table

The second component of GeoPackage metadata is the metadata\_reference table that links metadata in the xml\_metadata table to data in the feature, raster, and tiles tables defined in clauses 9.4, 10.5, and 10.6.

Table 52 -- Table metadata\_reference

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: metadata\_reference** | | | | | |
| **Column Name** | **Col Type** | **Column Description** | **Null** | **Default** | **Key** |
| reference\_scope | text | Metadata reference scope; must be one of ‘table’, ‘column’,’row’,’row/col’ | no | ‘table’ |  |
| table\_name | text | Name of the table to which this metadata reference applies; must be the name of a table in geometry\_columns or raster\_columns table | no | ‘undefined’ |  |
| column\_name | text | Name of the column to which this metadata reference applies; ‘undefined’ for reference\_scope of ‘table’ or ‘row’, or the name of a column in the table\_name table for reference\_scope of ‘column’ or ‘row/col’ | no | ‘undefined’ |  |
| row\_id\_value | int | 0 for reference\_scope of ‘table’ or ‘column’, or the rowed of a row record in the table\_name table for reference\_scope of ‘row’ or ‘row/col’ | no | 0 |  |
| timestamp | text | timestamp value in ISO 8601format as defined by the strftime function '%Y-%m-%dT%H:%M:%fZ' format string applied to the current time | no | strftime('%Y-%m-%dT%H:%M:%fZ', CURRENT\_TIMESTAMP) |  |
| md\_file\_id | int | xml\_metadata table id column value for the metadata to which this metadata\_reference applies | no | 0 | FK |
| md\_parent\_id | int | xml\_metadata table id column value for the hierarchical parent metadata for the metadata to which this metadata\_reference applies | no | 0 | FK |

Every GeoPackage shall contain a metadata\_reference table with columns as described in table 52 as defined by the SQL in table 53. Every GeoPackage metadata\_reference table that contains any rows shall contain at least one row record with an md\_parent\_id value of 0 that references the ‘undefined’ xml\_metadata row record as defined by the SQL in table 51. Such record(s) establish the metadata reference to the “root” of a metadata hierarchy.

NOTE1: Such a metadata hierarchy may have only one level of defined metadata.

NOTE2: A 0 row\_id\_value in the metadata\_reference tables indicates table or column scope and no corresponding row for the metadata reference. So a data row with a rowed value of 0 cannot be the subject of a metadata reference. GeoPackage applications should therefore avoid directly setting rowed values in feature, raster, or tile data tables to any values less than 1 to avoid assignment of 0 rowid values to data rows that could not be linked to metadata.

NOTE3: In an SQLite implementation, the rowid value is always equal to the value of a single-column primary key on an integer column [B30]. Althought not stated in the SQLite documentation, testing has not revealed a case where rowed values on a table with any primry key column(s) defined are changed by a database reorganization performed by the VACUUM SQL command.

Values of the metadata\_reference table timestamp column shall be in ISO 8601 format containing a complete date plus UTC hours, minutes, seconds and a decimal fraction of a second, with a ‘Z’ (‘zulu’) suffix indicating UTC.

NOTE4: The following statement selects such a timestamp value:

“SELECT **(**strftime('%Y-%m-%dT%H:%M:%fZ','now')**)”.**

Table 53 -- metadata\_reference Table Definition SQL

|  |
| --- |
| CREATE TABLE metadata\_reference (  reference\_scope TEXT NOT NULL DEFAULT "table",  table\_name TEXT NOT NULL DEFAULT "undefined",  column\_name TEXT NOT NULL DEFAULT "undefined",  row\_id\_value INTEGER NOT NULL DEFAULT 0,  timestamp TEXT NOT NULL DEFAULT (strftime('%Y-%m-%dT%H:%M:%fZ',CURRENT\_TIMESTAMP)),  md\_file\_id INTEGER NOT NULL DEFAULT 0,  md\_parent\_id INTEGER NOT NULL DEFAULT 0,  CONSTRAINT crmr\_mfi\_fk FOREIGN KEY (md\_file\_id) REFERENCES xml\_metadata(id),  CONSTRAINT crmr\_mpi\_fk FOREIGN KEY (md\_parent\_id) REFERENCES xml\_metadata(id)  ) |

Every GeoPackage metadata\_reference table shall have triggers defined to enforce column values as defined in table 54.

Table 54 -- metadata\_reference Trigger Definition SQL

|  |
| --- |
| CREATE TRIGGER 'metadata\_reference\_reference\_scope\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: reference\_scope must be one of "table", "column", "row", "row/col"')  WHERE NOT NEW.reference\_scope IN ('table','column','row','row/col');  END  CREATE TRIGGER 'metadata\_reference\_reference\_scope\_update'  BEFORE UPDATE OF 'reference\_scope' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: referrence\_scope must be one of "table", "column", "row", "row/col"')  WHERE NOT NEW.reference\_scope IN ('table','column','row','row/col');  END  CREATE TRIGGER 'metadata\_reference\_table\_name\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: table\_name must be the name of a table in geometry\_columns or raster\_columns')  WHERE NOT NEW.table\_name IN (  SELECT f\_table\_name AS table\_name FROM geometry\_columns  UNION ALL  SELECT r\_table\_name AS table\_name FROM raster\_columns);  END  CREATE TRIGGER 'metadata\_reference\_table\_name\_update'  BEFORE UPDATE OF 'table\_name' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: table\_name must be the name of a table in geometry\_columns or raster\_columns')  WHERE NOT NEW.table\_name IN (  SELECT f\_table\_name AS table\_name FROM geometry\_columns  UNION ALL  SELECT r\_table\_name AS table\_name FROM raster\_columns);  END  CREATE TRIGGER 'metadata\_reference\_column\_name\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: column name must be "undefined" when reference\_scope is "table" or "row"')  WHERE (NEW.reference\_scope IN ('table','row')  AND NEW.column\_name <> 'undefined');  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: column name must be defined for the specified table when reference\_scope is "column" or "row/col"')  WHERE (NEW.reference\_scope IN ('column','row/col')  AND NOT NEW.table\_name IN (  SELECT name FROM SQLITE\_MASTER WHERE type = 'table'  AND name = NEW.table\_name  AND sql LIKE ('%' || NEW.column\_name || '%')));  END  CREATE TRIGGER 'metadata\_reference\_column\_name\_update'  BEFORE UPDATE OF column\_name ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: column name must be "undefined" when reference\_scope is "table" or "row"')  WHERE (NEW.reference\_scope IN ('table','row')  AND NEW.column\_name <> 'undefined');  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: column name must be defined for the specified table when reference\_scope is "column" or "row/col"')  WHERE (NEW.reference\_scope IN ('column','row/col')  AND NOT NEW.table\_name IN (  SELECT name FROM SQLITE\_MASTER WHERE type = 'table'  AND name = NEW.table\_name  AND sql LIKE ('%' || NEW.column\_name || '%')));  END  CREATE TRIGGER 'metadata\_reference\_row\_id\_value\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: row\_id\_value must be 0 when reference\_scope is "table" or "column"')  WHERE NEW.reference\_scope IN ('table','column')  AND NEW.row\_id\_value <> 0;  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: row\_id\_value must exist in specified table when reference\_scope is "row" or "row/col"')  WHERE NEW.reference\_scope IN ('row','row/col')  AND NOT EXISTS (SELECT rowid  FROM (SELECT NEW.table\_name AS table\_name) WHERE rowid = NEW.row\_id\_value);  END  CREATE TRIGGER 'metadata\_reference\_row\_id\_value\_update'  BEFORE UPDATE OF 'row\_id\_value' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: row\_id\_value must be 0 when reference\_scope is "table" or "column"')  WHERE NEW.reference\_scope IN ('table','column')  AND NEW.row\_id\_value <> 0;  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: row\_id\_value must exist in specified table when reference\_scope is "row" or "row/col"')  WHERE NEW.reference\_scope IN ('row','row/col')  AND NOT EXISTS (SELECT rowid  FROM (SELECT NEW.table\_name AS table\_name) WHERE rowid = NEW.row\_id\_value);  END  CREATE TRIGGER 'metadata\_reference\_timestamp\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: timestamp must be a valid time in ISO 8601 "yyyy-mm-ddThh-mm-ss.cccZ" form')  WHERE NOT (NEW.timestamp GLOB  '[1-2][0-9][0-9][0-9]-[0-1][0-9]-[1-3][0-9]T[0-2][0-9]:[0-5][0-9]:[0-5][0-9].[0-9][0-9][0-9]Z'  AND strftime('%s',NEW.timestamp) NOT NULL);  END  CREATE TRIGGER 'metadata\_reference\_timestamp\_update'  BEFORE UPDATE OF 'timestamp' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: timestamp must be a valid time in ISO 8601 "yyyy-mm-ddThh-mm-ss.cccZ" form')  WHERE NOT (NEW.timestamp GLOB  '[1-2][0-9][0-9][0-9]-[0-1][0-9]-[1-3][0-9]T[0-2][0-9]:[0-5][0-9]:[0-5][0-9].[0-9][0-9][0-9]Z'  AND strftime('%s',NEW.timestamp) NOT NULL);  END |

Table 55 -- Example: metadata\_reference SQL insert statement

|  |
| --- |
| INSERT INTO metadata\_reference VALUES (  'table','sample\_rasters','undefined', 0, '2012-08-17T14:49:32.932Z', 98, 99) |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/metadata/metadata\_reference\_table | |
| REQ 74. | There shall be a metadata\_reference table in a GeoPackage as specified in clause 11.2 with the columns described in table 52 and defined by SQL in table 53, and values as specified in clause 11.2. |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/metadata/metadata\_reference\_table/triggers | |
| REQ 75. | The metadata\_reference table in a GeoPackage shall have the triggers specified in clause 11.2, as defined by the SQL in table 54. |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/metadata/metadata\_reference\_table/undefined\_parent | |
| REQ 76. | A metadata\_reference table in a GeoPackage with any rows shall contain at least one row record with an md\_parent\_id value of 0 that references the ‘undefined’ xml\_metadata row record as defined by the SQL in table 51 (see REQ 73). |

# 12 Other Data Tables

Application-specific or vendor-specific data tables that do not conform to the table definitions or other requirements for storing feature data, raster or tile data, metadata, or manifests as defined in other sections of this document may be defined and used to store supplemental non-geospatial information in a GeoPackage database. However, such other data tables shall not be used instead of the tables defined in other sections of this document to store feature data, raster or tile data, metadata, or manifests. There shall not be any records in the geopackage\_contents table referencing such other data tables. Any such other data tables shall be represented as such in entries in the manifest as described in section 13 below.

# 13 Manifest

## 13.1 Introduction

The GeoPackage manifest serves as an extended table of contents and a data source access metadata store for the GeoPackage data container. It is intended to provide a standard interface between GeoPackage contents and OGC and otherWeb Services that have provided the content in a GeoPackage, or that in the future may use GeoPackages as an input or output format. The GeoPackage manifest will facilitate two-way geosynchronization of GeoPackage contents with other data stores via web services. GeoPackage manifest metadata may also be used to provide identifying and coverage area information for a GeoPackage and for the GeoPackages with data for adjacent areas.

## 13.2 Manifest Content Model and Table Definition

A GeoPackage manifest is represented by a single XML document. The canonical GeoPackage manifest content model is an extension of the ATOM [45] encoding of the OGC® OWS Context (OWC) Document Conceptual Model [46]. Other XML document content models may also be used, but only one manifest XML document may be provided per content model.

A GeoPackage shall have a manifest table as desecribed in table 56 and defined in table 57.

Table 56 -- ows\_manifest table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table or View Name: manifest** | | | | | |
| **Column Name** | **Column Type** | **Column Description** | **Null** | **Default** | **Key** |
| id | text | Identifier name or abbreviation of the manifest document content model. | no | ‘OWC’ | PK |
| manifest | text | XML manifest document fragment | no | ‘’ |  |

Table 57 -- ows\_manifest Table Definition SQL

|  |
| --- |
| CREATE TABLE manifest (  id TEXT NOT NULL DEFAULT 'OWC',  manifest TEXT NOT NULL DEFAULT '',  CONSTRAINT pk\_m PRIMARY KEY (id) ON CONFLICT ROLLBACK  ); |

|  |  |
| --- | --- |
| Requirement: Core | |
| http://www.opengis.net/spec/GPKG/1.0/req/manifest/manifest\_table | |
| REQ 77. | There shall be a manifest table in a GeoPackage as specified in clause 13.2 with the columns described in table 56 and defined by the SQL in table 57. |

## 13.3 Manifest XML Schema

The GeoPackage manifest is defined in the geoPackageContext.xsd XML schema document shown in table 58 below as an extension of the OGC Context document Atom Encoding defined in [http://schemas.opengis.net/ owc/1.0/OWSContextCore.xsd](http://schemas.opengis.net/ows/2.0/owsManifest.xsd) by 12-084 OWS Context Atom Encoding [47]. The Manifest for a GeoPackage is encoded as an atom:feed element that contains elements describing the GeoPackage itself and GeoPackages for adjacent areas, and atom:entry elements which describe the contents and optionally the source of data in GeoPackage container tables. The geoPackageContext.xsd XML schema defines elements and attributes that are used to extend those of Atom and OWS Context. See the documentation elements in the schema in table 58 for descriptions of these elements and attributes. Their use in a manifest document is discussed in the next section.

The <http://www.opengis.net/gpkg/1.0> namespace will be proposed to the OGC Naming Authority.

Table 58 -- GeoPackage Manifest XML Schema

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <schema xmlns:gpkg="http://www.opengis.net/gpkg/1.0" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gmd="http://www.isotc211.org/2005/gmd" targetNamespace="http://www.opengis.net/gpkg/1.0" elementFormDefault="qualified" version="1.0.0" xml:lang="en">  <!-- 2012-09-04 -->  <annotation>  <documentation>Manifest schema for OGC GeoPackage as OWS context / Atom extension</documentation>  </annotation>  <!-- ========================================================== -->  <import namespace="http://www.isotc211.org/2005/gmd" schemaLocation="../../iso/19139/20070417/gmd/gmd.xsd"/>  <!-- ==========================================================  Types, elements, and attributes  ========================================================== -->  <attribute name="area" type="gpkg:GeoPackageAreaNameType">  <annotation>  <documentation>Applies to Atom link elements in Atom feed element but not in Atom entry elements. </documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <simpleType name="GeoPackageAreaNameType">  <annotation>  <documentation>Names of relative GeoPackage adjacent areas of data coverage.</documentation>  </annotation>  <restriction base="string">  <enumeration value="Here"/>  <enumeration value="North"/>  <enumeration value="NorthEast"/>  <enumeration value="East"/>  <enumeration value="SouthEast"/>  <enumeration value="South"/>  <enumeration value="SouthWest"/>  <enumeration value="West"/>  <enumeration value="NorthWest"/>  </restriction>  </simpleType>  <!-- =========================================================== -->  <attribute name="tableType" type="gpkg:TableTypeNameType">  <annotation>  <documentation>Applies to Atom link elements with rel="self" in Atom entry elements.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <simpleType name="TableTypeNameType">  <annotation>  <documentation>Names of the types GeoPackage tables that contain geospatial data content.</documentation>  </annotation>  <restriction base="string">  <enumeration value="features">  <annotation>  <documentation>Vector features with or without geometry attribute(s), but no raster attribute(s).</documentation>  </annotation>  </enumeration>  <enumeration value="rasters">  <annotation>  <documentation>Rasters table with id primary key and raster attribute(s), but no geometry attribute(s).</documentation>  </annotation>  </enumeration>  <enumeration value="featuresWithRasters">  <annotation>  <documentation>Vector features with id primary key, geometry attribute(s) and raster attribute(s). </documentation>  </annotation>  </enumeration>  <enumeration value="tiles">  <annotation>  <documentation>Tile matrix set of tile matrices with rasters at different zoom levels.</documentation>  </annotation>  </enumeration>  <enumeration value="other">  <annotation>  <documentation>Other data, not features, rasters, featuresWithRasters, or tiles, that is vendor or application specific.</documentation>  </annotation>  </enumeration>  </restriction>  </simpleType>  <!-- =========================================================== -->  <!-- <attribute name="dataVersion" type="gpkg:VersionType"/> replaced by atom:updated -->  <annotation>  <documentation>The following five attributes apply to Atom link elements with rel="self" in the Atom feed element.</documentation>  </annotation>  <attribute name="geopackageVersion" type="gpkg:GeoPackageVersionType"/>  <attribute name="sqliteVersion" type="gpkg:GeoPackageVersionType"/>  <attribute name="spatialiteVersion" type="gpkg:GeoPackageVersionType"/>  <attribute name="projVersion" type="gpkg:GeoPackageVersionType"/>  <attribute name="geosVersion" type="gpkg:VersionType"/>  <!-- =========================================================== -->  <simpleType name="GeoPackageVersionType">  <annotation>  <documentation>Version type restricted to a string of the form 1.2.3, with an optional suffix, such as "a", "b", "-RC1"</documentation>  </annotation>  <restriction base="string">  <pattern value="[1-9][0-9]\*\.[0-9]+\.[0-9]+[a-z,A-Z,0-9,\-]\*"/>  </restriction>  </simpleType>  <!-- =========================================================== -->  <simpleType name="VersionType">  <annotation>  <documentation>Version type specified by any string, to allow convention used by any data provider.</documentation>  </annotation>  <restriction base="string"/>  </simpleType>  <!-- =========================================================== -->  <attribute name="mdScope" type="gpkg:MetadataScopeType">  <annotation>  <documentation>Applies to gpkg:Metadata element.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <simpleType name="MetadataScopeType">  <annotation>  <documentation>Names of metadata scopes from ISO 19115 B.5.25 MD\_ScopeCode code list, with extensions</documentation>  </annotation>  <restriction base="string">  <enumeration value="undefined"/>  <enumeration value="fieldSession"/>  <enumeration value="collectionSession"/>  <enumeration value="series"/>  <enumeration value="dataset"/>  <enumeration value="featureType"/>  <enumeration value="feature"/>  <enumeration value="attributeType"/>  <enumeration value="attribute"/>  <enumeration value="tile"/>  <enumeration value="model"/>  <enumeration value="catalogue"/>  <enumeration value="schema"/>  <enumeration value="taxonomy"/>  <enumeration value="software"/>  <enumeration value="service"/>  <enumeration value="collectionHardware"/>  <enumeration value="nonGeographicDataset"/>  <enumeration value="dimensionGroup"/>  </restriction>  </simpleType>  <!-- =========================================================== -->  <attribute name="authority" type="anyURI">  <annotation>  <documentation>Authority for metadata model structure, e.g. ISO, DC, etc. Applies to gpkg:Metadata element.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <attribute name="columnName" type="string">  <annotation>  <documentation>Column name to identify the attribute or tile type described by attributeType metadata, or combined with rowid, the attribute or tile instance described by metadata. Applies to gpkg:Metadata element.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <attribute name="rowid" type="integer">  <annotation>  <documentation>ROWID to identify the row described by feature instance metadata, or combined with columnName, the attribute or tile instance described by metadata. Applies to gpkg:Metadata element.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <attribute name="boundsRef" type="anyURI">  <annotation>  <documentation>Reference to current 2-D bounding polygons as well as future 3-D bounding geometric solids. Although OWS:Context allows gml:Envelope here, GeoPackage requires gml:Polygon to carry gml:id attribute to support gpkg:boundsRef attribute reference from GeoPackage extended Atom link element with rel="self" attribute.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <attribute name="href" type="anyURI">  <annotation>  <documentation>Online source of metadata. Applies to gpkg:Metadata element.</documentation>  </annotation>  </attribute>  <!-- =========================================================== -->  <element name="Metadata" type="gpkg:MetadataWithAuthorityReferenceType">  <annotation>  <documentation>May be child of Atom feed element describing a GeoPackage, or Atom entry element describing a GeoPackage data table. When it appears as child of the Atom feed element, it implicitly applies to all GeoPackage data tables. </documentation>  </annotation>  </element>  <!-- =========================================================== -->  <complexType name="MetadataWithAuthorityReferenceType" mixed="true">  <sequence>  <any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>  </sequence>  <attribute ref="gpkg:authority" use="required"/>  <attribute ref="gpkg:mdScope" use="required"/>  <attribute ref="gpkg:columnName"/>  <attribute ref="gpkg:rowid"/>  <attribute ref="gpkg:href"/>  </complexType>  <!-- =========================================================== -->  </schema> |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/manifest/manifest\_table/owc\_content | |
| REQ 78. | The manifest table in a GeoPackage specified in clause 13.2 shall contain a record with an “OWC” id value and XML-encoded manifest data that is schema-valid in accordance with the OWC GeoPackage Manifest content model as specified by atom.xsd, OWSContextCore.xsd and geoPackageContext.xsd in clause 13.3. |

## 13.4 Manifest Schematron Schema

TODO

Table 59 -- GeoPackage Manifest Schematron Schema

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Requirement: Extension | |
| http://www.opengis.net/spec/GPKG/1.0/req/manifest/manifest\_table/owc\_content | |
| REQ 79. | The manifest table in a GeoPackage specified in clause 13.2 shall contain a record with an “OWC” id value and XML-encoded manifest data that is schema-valid in accordance with the geoPackageContext.sch Schematron schema specified in clause 13.4. |

## 13.5 Sample Manifest XML Document

A sample GeoPackage Manifest XML document is show in table 60 below. Descriptions of its contents are provided in table 61 below.

Table 60 -- EXAMPLE: Sample GeoPackage Manifest XML Document

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?valbuddy\_schematron ./owc\_schemas\_v3/gpkg/1.0/GeoPackageContext.sch?>  <?xml-model href="../atom/2005/atom.rng" type="application/relax-ng-regular-syntax"?>  <?xml-stylesheet type="text/xsl" href="C:\sw\apache-tomcat-6.0.35\webapps\examples\owscontext\GeoPackageManifestOWC2HTML.xslt"?>  <feed xmlns="http://www.w3.org/2005/Atom"  xmlns:dc="http://purl.org/dc/elements/1.1/"  xmlns:gmd="http://www.isotc211.org/2005/gmd"  xmlns:georss="http://www.georss.org/georss"  xmlns:gml="http://www.opengis.net/gml"  xmlns:owc="http://www.opengis.net/owc/1.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:gpkg="http://www.opengis.net/gpkg/1.0"  xsi:schemaLocation="http://www.w3.org/2005/Atom ../../../owc/1.0/atom/2005/atom.xsd  http://purl.org/dc/elements/1.1/ ../../../csw/2.0.2/rec-dcmes.xsd  http://www.georss.org/georss ../../../owc/1.0/georss/1.1/georss.xsd  http://www.opengis.net/gml ../../../owc/1.0/georss/1.1/gmlgeorss311.xsd  http://www.opengis.net/owc/1.0 ../../../owc/1.0/OWSContextCore.xsd  http://www.opengis.net/gpkg/1.0 ../GeoPackageContext.xsd" xml:lang="en">  <!-- 2012-10-01 -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.manifest</id>  <!-- id: (mandatory) Unique IRI Identifier assigned to the OWS Context Document -->  <category scheme="http://www.opengis.net/gpkg" term="http://www.opengis.net/gpkg/1.0" label="GeoPackage Manifest compliant with v 1.0 of OGC GeoPackage"/>  <category scheme="http://www.opengis.net/owc/specReference" term="http://www.opengis.net/spec/owc/1.0/conf/atom" label="This file is compliant with version 1.0 of OGC Context"/>  <category scheme="http://www.eionet.europa.eu/gemet" term="surface transportation"/>  <!-- Keyword related to this context document. Assimilated to atom:feed/atom:category/@term-->  <title>GeoPackage Manifest in OWS Context (SampleGeoPackageManifestOWCv5.xml)</title>  <subtitle type="html">  Manifest containing references to GeoPackages for &lt;b&gt;this&lt;/b&gt; and adjacent areas, and references to data tables in this GeoPackage.  </subtitle>  <author>  <!-- author Identifier for the author of the document -->  <name>John Doe</name>  <email>JohnDoe@example.com</email>  <uri>http://example.com/~johndoe</uri>  </author>  <dc:publisher>OGC OWS-9 Project</dc:publisher>  <!-- publisher Identifier for the publisher of the document -->  <dc:date>2009-01-23T09:08:56.000Z</dc:date>  <updated>2012-02-21T11:58:23Z</updated>  <!-- updateDate Date when the Context Document was updated -->  <generator uri="http://www.terradue.com" version="1.0">CAS Context Aggregator</generator>  <!-- creatorApplication -->  <rights>Copyright (c) 2012. Some rights reserved. This feed licensed under a Creative Commons Attribution 3.0 License.</rights>  <!-- Rights which apply to the context document -->  <georss:where>  <!-- areaOfInterest Geographic Area of interest of the users of the content pointed or embedded in context document -->  <!-- Although OWS:Context allows gml:Envelope here, GeoPackage requires gml:Polygon to carry gml:id attribute to support gpkg:boundsRef attribute reference from link rel="self" -->  <gml:Polygon gml:id="here">  <gml:exterior>  <gml:LinearRing>  <gml:posList srsDimension="2">38.4921 44.2699 38.6058 43.4414 37.5318 43.2089 37.4215 44.0128 38.4921 44.2699</gml:posList>  </gml:LinearRing>  </gml:exterior>  </gml:Polygon>  </georss:where>  <link rel="self" type="application/x-gpkg" href="sample.geopackage" gpkg:area="#Here" gpkg:boundsRef="#here" gpkg:sqliteVersion="3.7.9" gpkg:spatialiteVersion="3.0.1" gpkg:geosVersion="3.3.1-CAPI-1.7.1" gpkg:projVersion="4.7.1" gpkg:geopackageVersion="1.0.0"/>  <!-- this gpkg -->  <link rel="alternate" type="application/x-gpkg" href="http://tec.army.mil/gpkg?service=GPKG&amp;request=GetGPKG&amp;id=sample.geopackage" gpkg:area="#Here" gpkg:boundsRef="here"/>  <!-- source of this gpkg -->  <link rel="related" type="application/x-gpkg" href="north.geopackage" gpkg:area="North"/>  <!-- GeoPackages for adjoining AOIs -->  <link rel="related" type="application/x-gpkg" href="northeast.geopackage" gpkg:area="NorthEast"/>  <link rel="related" type="application/x-gpkg" href="east.geopackage" gpkg:area="East"/>  <link rel="related" type="application/x-gpkg" href="southeast.geopackage" gpkg:area="SouthEast"/>  <link rel="related" type="application/x-gpkg" href="south.geopackage" gpkg:area="South"/>  <link rel="related" type="application/x-gpkg" href="southwest.geopackage" gpkg:area="SouthWest"/>  <link rel="related" type="application/x-gpkg" href="west.geopackage" gpkg:area="West"/>  <link rel="related" type="application/x-gpkg" href="northwest.geopackage" gpkg:area="NorthWest"/>  <gpkg:Metadata gpkg:authority="http://schemas.opengis.net/iso/19139/" gpkg:mdScope="series">  <!-- metadata that applies to all data tables in this GeoPackage -->  <gmd:MD\_Metadata>  <gmd:contact/>  <gmd:dateStamp/>  <gmd:identificationInfo/>  <!-- incomplete example -->  </gmd:MD\_Metadata>  </gpkg:Metadata>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.roads</id>  <category scheme="http://www.acme.com/category" term="vector features" label="vector feature data"/>  <!-- Keyword related to this resource definition. Assimilated to feed/entry/category/@term -->  <title>Primary and secondary roads</title>  <!-- <summary type="text">Primary and secondary roads, no state or interstate highways.</summary> -->  <!-- abstract -->  <author>  <!-- author Identifier for the author of the resource definition. -->  <name>John Doe</name>  <email>JohnDoe@example.com</email>  <uri>http://example.com/~johndoe</uri>  </author>  <dc:publisher>U.S. Census Bureau</dc:publisher>  <!-- publisher An entity responsible for making the resource available -->  <updated>2012-01-31T12:00:00Z</updated>  <!-- updateDate Date when the resource definition was updated. Assimilated to feed/entry/update -->  <dc:creator>ACME software</dc:creator>  <!-- creatorApplication An entity primarily responsible for making the resource -->  <rights>Copyright (c) 2012. Some rights reserved. This feed licensed under a Creative Commons Attribution 3.0 License. </rights>  <!-- Rights which apply to the resource definition -->  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml" xsi:schemaLocation="http://www.w3.org/1999/xhtml ../../../xhtml/xhtml1-strict.xsd">  Primary and secondary roads, no state or interstate highways.  </div>  </content>  <georss:where>  <!-- geospatialExtent Geographic extent of the data to be retrieved and used. Assimilated to geoss:where -->  <!-- Although OWS:Context allows gml:Envelope here, GeoPackage requires gml:Polygon to carry gml:id attribute to support gpkg:boundsRef attribute reference from link rel="self" -->  <gml:Polygon gml:id="conus">  <gml:exterior>  <gml:LinearRing>  <gml:posList srsDimension="2">38.4921 44.2699 38.6058 43.4414 37.5318 43.2089 37.4215 44.0128 38.4921 44.2699</gml:posList>  </gml:LinearRing>  </gml:exterior>  </gml:Polygon>  </georss:where>  <dc:date>2009-01-23T09:08:56.000Z</dc:date>  <!-- temporalExtent The temporal extent of the content of the resource -->  <link rel="self" type="application/x-gpkg" href="roads" gpkg:boundsRef="#conus" gpkg:tableType="features"/>  <!-- gpkg:dataVersion="2011.03.01" replaced by atom:updated -->  <!-- OWC serviceOffering only -->  <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/wfs">  <owc:operation code="GetCapabilities" method="GET" type="application/xml" href="http://geo.census.gov/arcgis\_server?service=wfs&amp;version=1.1.0&amp;request=GetCapabilities"/>  <owc:operation code="GetFeature" method="GET" type="application/gml+xml" href="http://geo.census.gov/arcgis\_server?service=wfs&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=roads"/>  </owc:offering>  <gpkg:Metadata gpkg:authority="http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index\_html" gpkg:href="http://www.census.gov/geo/www/tlmetadata/tl2005femeta.txt" gpkg:mdScope="series"/>  <!-- online metadata example -->  <gpkg:Metadata gpkg:authority="http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index\_html" gpkg:mdScope="dataset">  <csdgm:CSDGM xmlns:csdgm="http://www.fgdc.gov/csdgm"/>  </gpkg:Metadata>  <gpkg:Metadata gpkg:authority="http://www.loc.gov/catdir/cpso/lcco/" gpkg:mdScope="dataset">  <!-- "mixed" content example for authorities that do not encode metadata in XML -->  Main Class: G -- GEOGRAPHY. ANTHROPOLOGY. RECREATION  Subject: United States--Geography--Databases.  Authority: Census--Data processing.  </gpkg:Metadata>  <gpkg:Metadata gpkg:authority="http://dublincore.org" gpkg:mdScope="featureType">  <dc:publisher>U.S. Census Bureau</dc:publisher>  <dc:coverage>conus</dc:coverage>  </gpkg:Metadata>  <gpkg:Metadata gpkg:authority="http://dublincore.org" gpkg:mdScope="feature" gpkg:rowid="9876">  <dc:source>Maryland DOT</dc:source>  </gpkg:Metadata>  <gpkg:Metadata gpkg:authority="http://schemas.opengis.net/iso/19139/" gpkg:mdScope="attributeType" gpkg:columnName="surfaceType">  <gmd:MD\_DatatypeCode codeListValue="1234" codeList="attributeTypes"/>  </gpkg:Metadata>  <gpkg:Metadata gpkg:authority="http://schemas.opengis.net/iso/19139/" gpkg:mdScope="attribute" gpkg:columnName="geom" gpkg:rowid="9876">  <gmd:DQ\_DataQuality>  <gmd:scope/>  <gmd:report>  <gmd:DQ\_AbsoluteExternalPositionalAccuracy>  <gmd:result>  <gmd:DQ\_QuantitativeResult>  <gmd:valueUnit/>  <gmd:value/>  </gmd:DQ\_QuantitativeResult>  </gmd:result>  </gmd:DQ\_AbsoluteExternalPositionalAccuracy>  </gmd:report>  </gmd:DQ\_DataQuality>  </gpkg:Metadata>  <owc:minScaleDenominator>100</owc:minScaleDenominator>  <!-- minScaleDenominator Minimum scale for the display of the resource -->  <owc:maxScaleDenominator>100000</owc:maxScaleDenominator>  <!-- maxScaleDenominator Maximum scale for the display of the resource -->  </entry>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.streams</id>  <title>Streams in Massachusetts</title>  <!-- <summary type="text">This layer contains feature data on streams in Massachusetts, USA.</summary> -->  <!-- abstract Description of the Context Document Purpose/Content. Assimilated to feed/entry/summary -->  <updated>2012-01-31T12:00:00Z</updated>  <!-- updateDate Date when the resource definition was updated. Assimilated to feed/entry/update -->  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml" xsi:schemaLocation="http://www.w3.org/1999/xhtml ../../../xhtml/xhtml1-strict.xsd">  This layer contains feature data on streams in Massachusetts, USA  </div>  </content>  <georss:where>  <!-- geospatialExtent Geographic extent of the data to be retrieved and used. Assimilated to geoss:where -->  <!-- Although OWS:Context allows gml:Envelope here, GeoPackage requires gml:Polygon to carry gml:id attribute to support gpkg:boundsRef attribute reference from link rel="self" -->  <gml:Polygon gml:id="mass">  <gml:exterior>  <gml:LinearRing>  <gml:posList srsDimension="2">-73.5 42.1 -73.3 42.6 -70.5 42.5 -70.6 41.7 -73.5 42.1</gml:posList>  </gml:LinearRing>  </gml:exterior>  </gml:Polygon>  </georss:where>  <dc:date>2009-01-23T09:08:56.000Z</dc:date>  <!-- temporalExtent The temporal extent of the content of the resource -->  <link rel="self" type="application/x-gpkg" href="streams" gpkg:boundsRef="#mass" gpkg:tableType="features"/>  <!-- gpkg:dataVersion="2011.03.01" replaced by atom:updated -->  <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/wfs">  <owc:operation code="GetCapabilities" method="GET" type="application/xml" href="http://giswebservices.massgis.state.ma.us/geoserver/wfs?service=wfs&amp;version=1.1.0&amp;request=GetCapabilities"/>  <owc:operation code="GetFeature" method="GET" type="application/gml+xml" href="http://giswebservices.massgis.state.ma.us/geoserver/wfs?service=wfs&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=massgis:GISDATA.NWI\_ARC"/>  </owc:offering>  <gpkg:Metadata gpkg:authority="http://dublincore.org" gpkg:mdScope="dataset">  <dc:publisher>MassGIS</dc:publisher>  <dc:coverage>Massachusetts</dc:coverage>  </gpkg:Metadata>  </entry>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.wfstest</id>  <title>WFS Test Suite AggregateGeoFeature</title>  <!-- <summary type="text">Features from WFS 1.1.0 Test Suite.</summary> -->  <updated>2012-01-31T12:00:00Z</updated>  <!-- updateDate Date when the resource definition was updated. Assimilated to feed/entry/update -->  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml" xsi:schemaLocation="http://www.w3.org/1999/xhtml ../../../xhtml/xhtml1-strict.xsd">  Features from WFS 1.1.0 Test Suite.  </div>  </content>  <link rel="self" type="application/x-gpkg" href="wfstest" gpkg:tableType="features"/>  <!-- gpkg:dataVersion="2011.03.01" replaced by atom:updated -->  <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/wfs">  <owc:operation code="GetCapabilities" method="GET" type="application/xml" href="http://www.deegree.org/services/wfs?service=wfs&amp;version=1.1.0&amp;request=GetCapabilities"/>  <owc:operation code="GetFeature" method="POST" type="application/gml+xml" href="http://www.deegree.org/services/wfs">  <owc:payload type="application/xml">  <wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc" version="1.1.0" service="WFS">  <wfs:Query xmlns:sf="http://cite.opengeospatial.org/gmlsf" typeName="sf:AggregateGeoFeature" srsName="urn:ogc:def:crs:EPSG::4326">  <ogc:Filter>  <ogc:PropertyIsEqualTo>  <ogc:PropertyName>sf:strProperty</ogc:PropertyName>  <ogc:Literal>Ma quande lingues coalesce, li grammatica del resultant.</ogc:Literal>  </ogc:PropertyIsEqualTo>  </ogc:Filter>  </wfs:Query>  </wfs:GetFeature>  </owc:payload>  </owc:operation>  </owc:offering>  <gpkg:Metadata gpkg:authority="http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index\_html" gpkg:mdScope="dataset">  <csdgm:CSDGM xmlns:csdgm="http://www.fgdc.gov/csdgm"/>  </gpkg:Metadata>  </entry>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.maps\_rasters</id>  <title>Sample Raster Maps</title>  <!-- <summary type="text">Sample raster maps from various sources</summary> -->  <!-- abstract Description of the Context Document Purpose/Content. Assimilated to feed/entry/summary -->  <updated>2012-01-31T12:00:00Z</updated>  <!-- updateDate Date when the resource definition was updated. Assimilated to feed/entry/update -->  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml" xsi:schemaLocation="http://www.w3.org/1999/xhtml ../../../xhtml/xhtml1-strict.xsd">  Sample raster maps from various sources  </div>  </content>  <link rel="self" type="application/x-gpkg" href="maps\_rasters" gpkg:tableType="rasters"/>  <!-- gpkg:dataVersion="2011.03.01" replaced by atom:updated -->  <link rel="enclosure" gpkg:rowid="1" type="image/png" href="http://www.openstreetmap.org/?lat=39.094&amp;lon=-76.776&amp;zoom=10&amp;layers=M"/> <!-- not an OWS, so link rel="enclosure" instead of an owc:offering -->  <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/wms">  <owc:operation code="GetCapabilities" method="GET" type="application/xml" href="http://apps1.gdr.nrcan.gc.ca/cgi-bin/worldmin\_en-ca\_ows?service=wms&amp;version=1.1.1&amp;request=GetCapabilities"/>  <owc:operation code="GetMap" method="GET" gpkg:rowid="2" type="image/jpeg" href="http://apps1.gdr.nrcan.gc.ca/cgi-bin/worldmin\_en-ca\_ows?service=wms&amp;version=1.1.1&amp;request=GetMap&amp;layers=GSC:WORLD\_MineralDeposits"/>  </owc:offering>  <gpkg:Metadata gpkg:authority="http://dublincore.org" gpkg:mdScope="tile" gpkg:columnName="raster" gpkg:rowid="1">  <dc:publisher>Open Street Map</dc:publisher>  <dc:coverage>Baltimore / Washington D.C.</dc:coverage>  </gpkg:Metadata>  <gpkg:Metadata gpkg:authority="http://dublincore.org" gpkg:mdScope="tile" gpkg:columnName="raster" gpkg:rowid="2">  <dc:description>World Mineral Deposits</dc:description>  <dc:publisher>Natural Resources Canada</dc:publisher>  <dc:coverage>world</dc:coverage>  </gpkg:Metadata>  </entry>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.coverages\_rasters</id>  <title>Ocean Currents Coverage</title>  <updated>2012-01-31T12:00:00Z</updated>  <!-- updateDate Date when the resource definition was updated. Assimilated to feed/entry/update -->  <link rel="self" type="application/x-gpkg" href="coverages\_rasters" gpkg:tableType="rasters"/>  <!-- gpkg:dataVersion="2011.03.01" replaced by atom:updated -->  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml" xsi:schemaLocation="http://www.w3.org/1999/xhtml ../../../xhtml/xhtml1-strict.xsd">  High Frequency Radar Ocean Currents Coverage  </div>  </content>  <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/wcs">  <owc:operation code="GetCapabilities" method="GET" type="application/xml" href="http://sdf.ndbc.noaa.gov/thredds/wcs/hfradar\_usegc\_1km?service=WCS&amp;request=GetCapabilities"/>  <owc:operation code="GetCoverage" method="GET" gpkg:rowid="1" type="image/tiff" href="http://sdf.ndbc.noaa.gov/thredds/wcs/hfradar\_usegc\_1km?service=WCS&amp;request=GetCoverage&amp;coverage=u&amp;bbox=-98,21,-57,47&amp;time=2012-04-25T00:00:00Z&amp;format=GeoTIFF"/>  </owc:offering>  <gpkg:Metadata gpkg:authority="http://dublincore.org" gpkg:mdScope="tile" gpkg:columnName="raster" gpkg:rowid="1">  <dc:description>High Frequency Radar Ocean Currents Coverage</dc:description>  <dc:identifier>surface\_eastward\_sea\_water\_velocity</dc:identifier>  </gpkg:Metadata>  </entry>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.denmark\_tiles</id>  <title>Denmark Orthophotos</title>  <updated>2012-01-31T12:00:00Z</updated>  <!-- updateDate Date when the resource definition was updated. Assimilated to feed/entry/update -->  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml" xsi:schemaLocation="http://www.w3.org/1999/xhtml ../../../xhtml/xhtml1-strict.xsd">  Tile Matrix Set of Denmark Orthophotos  </div>  </content>  <georss:where>  <!-- geospatialExtent Geographic extent of the data to be retrieved and used. Assimilated to geoss:where -->  <!-- Although OWS:Context allows gml:Envelope here, GeoPackage requires gml:Polygon to carry gml:id attribute to support gpkg:boundsRef attribute reference from link rel="self" -->  <gml:Polygon gml:id="denmark">  <gml:exterior>  <gml:LinearRing>  <gml:posList srsDimension="2">7.9 54.3 7.9 57.8 13.2 57.8 7.9 57.8 7.9 54.3</gml:posList>  </gml:LinearRing>  </gml:exterior>  </gml:Polygon>  </georss:where>  <dc:date>2010-09-23T02:08:36.000Z</dc:date>  <!-- temporalExtent The temporal extent of the content of the resource -->  <link rel="self" type="application/x-gpkg" href="denmark\_tiles" gpkg:boundsRef="#denmark" gpkg:tableType="tiles"/>  <!-- gpkg:dataVersion="2011.03.01" replaced by atom:updated -->  <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/wmts">  <owc:operation code="GetCapabilities" method="GET" type="application/xml" href="http://kortforsyningen.kms.dk/topo\_skaermkort?client=Gaia&amp;service=WMTS&amp;request=GetCapabilities&amp;version=1.0.0"/>  <owc:operation code="GetTile" method="GET" gpkg:rowid="1" type="image/jpeg" href="http://kortforsyningen.kms.dk/topo\_skaermkort?client=Gaia&amp;service=WMTS&amp;request=GetTile&amp;version=1.0.0&amp;style=default&amp;format=image/jpeg&amp;TileMatrixSet=View1&amp;TileMatrix=L05&amp;TileRow=1&amp;TileCol=1"/>  <owc:operation code="GetTile" method="GET" gpkg:rowid="2" type="image/jpeg" href="http://kortforsyningen.kms.dk/topo\_skaermkort?client=Gaia&amp;service=WMTS&amp;request=GetTile&amp;version=1.0.0&amp;style=default&amp;format=image/jpeg&amp;TileMatrixSet=View1&amp;TileMatrix=L05&amp;TileRow=1&amp;TileCol=2"/>  <!-- more data\_access operations until Joan gives WMTS a GetMatrix operation -->  </owc:offering>  </entry>  <entry>  <!-- OWC Resource is a GeoPackage Table Reference -->  <id>http://www.opengis.net/gpkg/uuid/sample.geopackage.non.geopackage.other.data</id>  <title>Non GeoPackage Other Data</title>  <updated>2012-10-15T12:00:00Z</updated>  <content type="xhtml">  <div xmlns="http://www.w3.org/1999/xhtml">  Description of vendor or application specific other data that is not in a GeoPackage features, rasters, features with rasters, or tiles table.  </div>  </content>  <link rel="self" href="other\_non\_geopackage\_data" gpkg:tableType="other"/>  </entry>  </feed> |

In the following table, elements and attributes that do not have namespace prefixes are defined by the Atom Syndication Format [45] atom.xsd schema. Elements and attributes with an “owc” namespace prefix are defined in OWSContextCore.xsd. Elements and attributes with a “gml” namespace prefix are defined in [44]. Elements and attributes with a “gpkg” namespace prefix are defined by the GeoPackage manifest schema shown in table 58 above.

Table 61 -- GeoPackage Manifest XML Document Content Descriptions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GeoPackage Role** | **Element or Attribute XPath** | **Description** | **Obl.** | **Occ.** |
| Manifest | /feed | OGC Web Context Manifest container | M | 1 |
| Identifier | /feed/id | Identifier | M | 1 |
| Category | /feed/category | Category | O | 0..n |
| scheme | /feed/category/@scheme | Category scheme | M | 1 |
| term | /feed/category/@term | Category term keyword | M | 1 |
| Title | /feed/title | Title | M | 1 |
| Abstract | /feed/subtitle | Abstract or Description | M | 1 |
| Author | /feed/author | GeoPackage Author(s) | C[[1]](#footnote-1) | 0..n |
| Publisher | /feed/dc:publisher | GeoPackage Publisher | O | 0..1 |
| Pub Date | /feed/dc:date | GeoPackage publication date / time | M | 1 |
| Updated | /feed/updated | GeoPackage last update date/time | M | 1 |
| Source App | /feed/generator | Application that created the GeoPackage | O | 0..1 |
| Copyright | /feed/rights | Rights that apply to the GeoPackage contents | O | 0..1 |
| AOI | /feed/georss:where | Area of Interest for GeoPackage data contents | M | 1 |
| ..AOI BBox | /feed/georss:where/gml:Polygon | Defines AOI BBox Geometry | M | 1 |
| ..AOI Id | /feed/georss:where/gml:Polygon/@gml:id | Identifies AOI “here” | M | 1 |
| ..Container Identifier | /feed/link[@ rel=”self”] | GeoPackage Container Identifier | M | 1 |
| MIME | /feed/link[@ rel=”self”]/@type | MIME type “application/x-gpkg” | M | 1 |
| File | /feed/link[@ rel=”self”]/@href | Container reference | M | 1 |
| AOI Name | /feed/link[@ rel=”self”]/@gpkg:area | GeoPackageAreaNameType “Here” | M | 1 |
| AOI Ref | /feed/link[@ rel=”self”] /@gpkg:bounds\_ref | AOI reference: “here” | M | 1 |
| SQLite | /feed/link[@ rel=”self”]/ @gpkg:sqliteVersion | SQLite container version | M | 1 |
| SpatiaLite | /feed/link[@ rel=”self”]/ @gpkg:spatialiteVersion | Reference implementation version | O | 1 |
| GEOS | /feed/link[@ rel=”self”]/ @gpkg:geosVersion | Reference implementation version | O | 1 |
| PROJ | /feed/link[@ rel=”self”]/ @gpkg:projVersion | Reference implementation version | O | 1 |
| GPKG | /feed/link[@ rel=”self”]/ @gpkg:geopackageVersion | GeoPackage Specification version | M | 1 |
| GPKG Online Source | /feed/link[@ rel=”alternate”] | GeoPackage Online Web Service Source | O | 0..n |
| MIME | /feed/link[@ rel=”alternate”]/@type | MIME type “application/x-gpkg” | M | 1 |
| URL | /feed/link[@ rel=”alternate”]/@href | Web Service request URL | M | 1 |
| AOI Ref | /feed/link[@ rel=”alternate”] /@gpkg:bounds\_ref | AOI reference: “here” | M | 1 |
| Files for surrounding AOIs | /feed/link[@rel=”related”] | Container references for GeoPackages that cover surrounding AOIs | O | 0..8 |
| MIME | /feed/link[@rel=”related”]/@type | MIME type “application/x-gpkg” | M | 1 |
| File | /feed/link[@rel=”related”]/@href | Container reference | M | 1 |
| AOI Name | /feed/link[@ rel=”related”]/@gpkg:area | GeoPackageAreaNameType not “Here”; e.g. “North” … | M | 1 |
| Metadata | /feed/gpkg:Metadata | Metadata that applies to all data tables in this GeoPackage | O | 0..n |
| Authority | /feed/gpkg:Metadata/@gpkg:authority | URI identifying metadata specification authority | M | 1 |
| Scope | /feed/gpkg:Metadata/@gpkg:mdScope | MetadataScopeType e.g. “series”, “dataset”, … | M | 1 |
| Data Table Description | /feed/entry | OWC Resource: Data Table description container, one for each data table in GeoPackage | M | 0..n |
| Identifier | /feed/entry/id | Identifier | M | 1 |
| Category | /feed/entry/category | Category | O | 0..n |
| scheme | /feed/entry/category/@scheme | Category scheme | M | 1 |
| term | /feed/entry/category/@term | Category term keyword | M | 1 |
| Title | /feed/entry/title | Title | M | 1 |
| Abstract | /feed/entry/content | Abstract or Description | M | 1 |
| Author | /feed/entry/author | Table Data Author(s) | C[[2]](#footnote-2) | 0..n |
| Publisher | /feed/entry/dc:publisher | Table Data Publisher | O | 0..1 |
| Updated | /feed/entry/updated | Table Data last update date/time | M | 1 |
| Source App | /feed/entry/dc:creator | Application that created the Table Data | O | 0..1 |
| Copyright | /feed/entry/rights | Rights that apply to the Table Data | O | 0..1 |
| AOI | /feed/entry/georss:where | Area of Interest for Table Data | C[[3]](#footnote-3) | 1 |
| ..AOI BBox | /feed/entry/georss:where/gml:Polygon | Defines AOI BBox Geometry | M | 1 |
| ..AOI Id | /feed/entry/georss:where/gml:Polygon/@gml:id | Identifies AOI e.g. “conus” | M | 1 |
| ..Temporal extent | /feed/entry/dc:date | The temporal extent of the Table Data | C[[4]](#footnote-4) | 1 |
| ..Data Table Container Identifier | /feed/entry/link[@ rel=”self”] | Data Table Container Identifier | M | 1 |
| MIME | /feed/entry/link[@ rel=”self”]/@type | MIME type “application/x-gpkg” | C[[5]](#footnote-5) | 1 |
| Data Table Name | /feed/entry/link[@ rel=”self”]/@href | Data Table Container reference | M | 1 |
| AOI Ref | /feed/entry/link[@ rel=”self”] /@gpkg:bounds\_ref | AOI reference: e.g. “conus” | C[[6]](#footnote-6) | 1 |
| TableType | /feed/entry/@gpkg:tableType | TableTypeNameType: “features”, “rasters”, “featuresWithRasters”, “tiles”, or “other” | M | 1 |
| Table Data Source (OWS) | /feed/entry/owc:offering | Table Data Source from an OGC Web Service | O | 0..n |
| ....Service | /feed/entry/owc:offering/@code | URI identifying the type of service offering | M | 1 |
| Operation | /feed/entry/owc:offering/owc:operation | Service Operation | M | 1..n |
| ......Identifier | /feed/entry/owc:offering/owc:operation/@code | Code identifying the operation | M | 1 |
| ......Method | /feed/entry/owc:offering/owc:operation/@method | HTTP Operation Method | M | 1 |
| ......Data Row Scope | /feed/entry/owc:offering/owc:operation/@gpkg:rowid | Table Data Row ID of data from this operation (All Table Data Rows if unspecified unless @code is “GetCapabilities”) | O | 0..1 |
| MIME | /feed/entry/owc:offering/owc:operation/@type | MIME type of data returned from this operation | M | 1 |
| ......Request URI | /feed/entry/owc:offering/owc:operation/@href | Request URI to invoke the operation | M | 1 |
| ......Request Payload | /feed/entry/owc:offering/owc:operation/owc:payload | Request data if @method is “POST” | C | 1 |
| MIME | feed/entry/owc:offering/owc:operation/owc:payload/@type | MIME type of request payload | M | 1 |
| Table Data Source (non-OWS) | /feed/entry/link[@rel=”enclosure”] | Table Data Source from an non-OGC Web Service | O | 0..n |
| ....Data Row Scope | /feed/entry/link[@rel=”enclosure”]/  @gpkg:rowid | Table Data Row ID of data from this operation (All Table Data Rows if unspecified) | O | 0..1 |
| MIME | /feed/entry/link[@rel=”enclosure”]/  @type | MIME type of data returned from this operation | M | 1 |
| ....Request URI | /feed/entry/link[@rel=”enclosure”]/  @href | Request URI to invoke the operation | M | 1 |
| Metadata | /feed/entry/gpkg:Metadata | Metadata that applies to this Data Table | O | 0..n |
| Authority | /feed/entry/gpkg:Metadata/ @gpkg:authority | URI identifying metadata specification authority | M | 1 |
| Scope | /feed/entry/gpkg:Metadata/ @gpkg:mdScope | MetadataScopeType e.g. “dataset”, “feature”, “attribute”, “tile” … | M | 1 |
| ....Column | /feed/entry/gpkg:Metadata/ @gpkg:columnName | Column name when @gpkg:mdScope is “attributeType” or “attribute” or “tile” | O | 0..1 |
| ....Row | /feed/entry/gpkg:Metadata/@gpkg:rowid | Table Data Row ID when @gpkg:mdScope is “feature” | O | 0..1 |

Annex A – Conformance – Abstract Test Suite (normative)

A.1 General

This Annex contains conformance classes grouping and including all of the requirements specified in this standard as described in clause A.2. It also contains conformance test specifications in clause A.3 that comprise an Abstract Test Suite for those conformance classes, in accordance with the provisions of OGC08-131r3 The Specification Model — A Standard for Modular specifications [33]. Both requirements and conformance classes are named in conformance with OGC10-103 Name type specification - specification elements [34].

A.2 Conformance Classes

Requirement specified in this standard are grouped into the following conformance classes as shown in figure 3 below.

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/full\_geopackage |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/webp\_image\_type |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/geotiff\_image\_type |
| Dependency |  |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_zoom\_levels/other\_intervals |
| Import | http://www.opengis.net/spec/GPKG/1.0/conf/safe\_geopackage |

|  |  |  |
| --- | --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/webp\_image\_type | |
| Requirements | 51 | |
| Import | http://www.opengis.net/spec/GPKG/1.0/conf/base\_geopackage | |
| http://www.opengis.net/spec/GPKG/1.0/conf/owc\_manifest | | |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/geotiff\_image\_type |
| Requirements | 52 |
| Import | http://www.opengis.net/spec/GPKG/1.0/conf/base\_geopackage |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/owc\_manifest |
| Requirements | 78 |
|  |  |

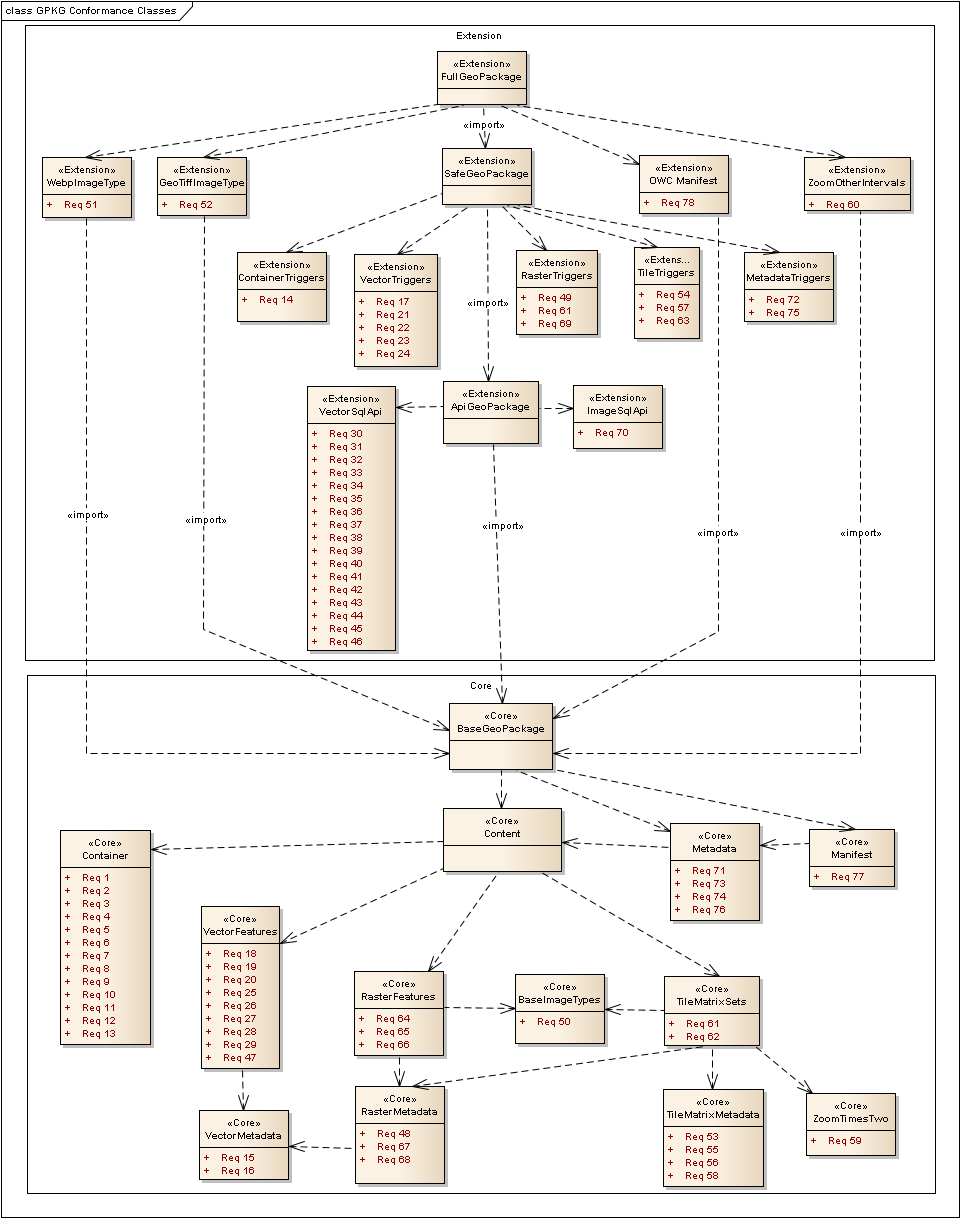


Figure 3 GeoPackage Conformance Classes

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_zoom\_levels/other\_intervals |
| Requirements | 58 |
| Import | http://www.opengis.net/spec/GPKG/1.0/conf/base\_geopackage |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/safe\_geopackage |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_triggers |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_triggers |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/tile\_triggers |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/metadata\_triggers |
| Import | http://www.opengis.net/spec/GPKG/1.0/conf/api\_geopackage |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_triggers |
| Requirements | 17, 21, 22, 23, 24 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_triggers |
| Requirements | 49,61,69 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/tile\_triggers |
| Requirements | 54,57,63 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/metadata\_triggers |
| Requirements | 72,75 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/api\_geopackage |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_sql\_api |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/image\_sql\_api |
| Import | http://www.opengis.net/spec/GPKG/1.0/conf/base\_geopackage |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_sql\_api |
| Requirements | 30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/image\_sql\_api |
| Requirements | 70 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/base\_geopackage |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/content |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/metadata |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/manifest |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/content |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/container |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_features |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_features |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/tile\_matrix\_sets |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/container |
| Requirements | 1,2,3,4,5,6,7,8,9,10,11,12,13 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/metaedata |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/content |
| Requirements | 71,73,74,75 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/manifest |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/metadata |
| Requirements | 77 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_features |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_metadata |
| Requirements | 18,19,20,25,26,27,28,29,47 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_features |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_metadata |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/base\_image\_types |
| Requirements | 64,65,66 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/tile\_matrix\_sets |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_metadata |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/tile\_matrix\_metadata |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/conf/base\_image\_types |
| Dependency | http://www.opengis.net/spec/GPKG/1.0/req/rasters\_tiles/tile\_matrix\_zoom\_levels/powers\_of\_two |
| Requirements | 61,62 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/base\_image\_types |
| Requirements | 50 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/vector\_metadata |
| Requirements | 15,16 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/raster\_metadata |
| Requirements | 48,67,68 |

|  |  |
| --- | --- |
| Conformance Class | http://www.opengis.net/spec/GPKG/1.0/conf/tile\_matrix\_metadata |
| Requirements | 53,55,56,58 |

A.3 Abstract Test Suite

TODO

Annex B – GeoPackage Trigger Definition SQL (informative)

|  |
| --- |
| DROP DROP TRIGGER IF EXISTS 'geopackage\_contents\_table\_name\_insert';  CREATE TRIGGER 'geopackage\_contents\_table\_name\_insert'  BEFORE INSERT ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'insert on geopackage\_contents violates constraint: table\_name value must not contain a single quote')  WHERE NEW.table\_name LIKE ('%''%');  SELECT RAISE(ROLLBACK,'insert on geopackage\_contents violates constraint: table\_name value must not contain a double quote')  WHERE NEW.table\_name LIKE ('%"%');  SELECT RAISE(ROLLBACK,'insert on geopackage\_contents violates constraint: table\_name value must be lower case')  WHERE NEW.table\_name <> lower(NEW.table\_name);  END;  DROP TRIGGER IF EXISTS 'geopackage\_contents\_table\_name\_update';  CREATE TRIGGER 'geopackage\_contents\_table\_name\_update'  BEFORE UPDATE OF 'table\_name' ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'update of geopackage\_contents violates constraint: table\_name value must not contain a single quote')  WHERE NEW.table\_name LIKE ('%''%');  SELECT RAISE(ROLLBACK,'update of geopackage\_contents violates constraint: table\_name value must not contain a double quote')  WHERE NEW.table\_name LIKE ('%"%');  SELECT RAISE(ROLLBACK,'update of geopackage\_contents violates constraint: table\_name value must be lower case')  WHERE NEW.table\_name <> lower(NEW.table\_name);  END;  DROP DROP TRIGGER IF EXISTS 'geopackage\_contents\_feature\_type\_insert;'  CREATE TRIGGER 'geopackage\_contents\_feature\_type\_insert'  BEFORE INSERT ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table geopackage\_contents violates constraint: data\_type must be one of features | featuresWithRasters | rasters | tiles')  WHERE NOT(NEW.data\_type IN ('features','featuresWithRasters','rasters','tiles'));  END;  DROP DROP TRIGGER IF EXISTS 'geopackage\_contents\_feature\_type\_update';  CREATE TRIGGER 'geopackage\_contents\_feature\_type\_update'  BEFORE UPDATE OF 'data\_type' ON 'geopackage\_contents'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table geopackage\_contents violates constraint: data\_type must be one of features | featuresWithRasters | rasters | tiles')  WHERE NOT(NEW.data\_type IN ('features','featuresWithRasters','rasters','tiles'));  END;  DROP DROP TRIGGER IF EXISTS 'geometry\_columns\_f\_geometry\_column\_insert';  CREATE TRIGGER 'geometry\_columns\_f\_geometry\_column\_insert'  BEFORE INSERT ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'insert on geometry\_columns violates constraint: f\_geometry\_column value must not contain a single quote')  WHERE NEW.f\_geometry\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'insert on geometry\_columns violates constraint: f\_geometry\_column value must not contain a double quote')  WHERE NEW.f\_geometry\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'insert on geometry\_columns violates constraint: f\_geometry\_column value must be lower case')  WHERE NEW.f\_geometry\_column <> lower(NEW.f\_geometry\_column);  END;  DROP DROP TRIGGER IF EXISTS 'geometry\_columns\_f\_geometry\_column\_update';  CREATE TRIGGER 'geometry\_columns\_f\_geometry\_column\_update'  BEFORE UPDATE OF 'f\_geometry\_column' ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'update on geometry\_columns violates constraint: f\_geometry\_column value must not contain a single quote')  WHERE NEW.f\_geometry\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'update on geometry\_columns violates constraint: f\_geometry\_column value must not contain a double quote')  WHERE NEW.f\_geometry\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'update on geometry\_columns violates constraint: f\_geometry\_column value must be lower case')  WHERE NEW.f\_geometry\_column <> lower(NEW.f\_geometry\_column);  END;  DROP DROP TRIGGER IF EXISTS 'geometry\_columns\_geometry\_type\_insert';  CREATE TRIGGER 'geometry\_columns\_geometry\_type\_insert'  BEFORE INSERT ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'geometry\_type must be one of 0,1,2,3,4,5,6,7,13,14,1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007')  WHERE NOT(NEW.geometry\_type IN (0,1,2,3,4,5,6,7,13,14,  1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007));  END;  DROP DROP TRIGGER IF EXISTS 'geometry\_columns\_geometry\_type\_update';  CREATE TRIGGER 'geometry\_columns\_geometry\_type\_update'  BEFORE UPDATE OF 'geometry\_type' ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'geometry\_type must be one of 0,1,2,3,4,5,6,7,13,14,1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007)  WHERE NOT(NEW.geometry\_type IN (0,1,2,3,4,5,6,7,13,14,  1000,1001,1002,1003,1004,1005,1006,1007,  2000,2001,2002,2003,2004,2005,2006,2007,  3000,3001,3002,3003,3004,3005,3006,3007));  END  DROP DROP TRIGGER IF EXISTS \_columns\_coord\_dimension\_insert';  CREATE TRIGGER 'geometry\_columns\_coord\_dimension\_insert'  BEFORE INSERT ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'coord\_dimension must be one of 2,3,4')  WHERE NOT(NEW.geometry\_type IN (2,3,4));  END;  DROP DROP TRIGGER IF EXISTS 'geometry\_columns\_coord\_dimension\_update';  CREATE TRIGGER 'geometry\_columns\_coord\_dimension\_update'  BEFORE UPDATE OF 'coord\_dimension' ON 'geometry\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'coord\_dimension must be one of 2,3,4')  WHERE NOT(NEW.geometry\_type IN (2,3,4));  END;  DROP DROP TRIGGER IF EXISTS 'raster\_columns\_r\_raster\_column\_insert';  CREATE TRIGGER 'raster\_columns\_r\_raster\_column\_insert'  BEFORE INSERT ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'insert on raster\_columns violates constraint: r\_raster\_column value must not contain a single quote')  WHERE NEW.r\_raster\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'insert on raster\_columns violates constraint: r\_raster\_column value must not contain a double quote')  WHERE NEW.r\_raster\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'insert on raster\_columns violates constraint: r\_raster\_column value must be lower case')  WHERE NEW.r\_raster\_column <> lower(NEW.r\_raster\_column);  END;  DROP DROP TRIGGER IF EXISTS 'raster\_columns\_r\_raster\_column\_update';  CREATE TRIGGER 'raster\_columns\_r\_raster\_column\_update'  BEFORE UPDATE OF r\_raster\_column ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK,'update on raster\_columns violates constraint: r\_raster\_column value must not contain a single quote')  WHERE NEW.r\_raster\_column LIKE ('%''%');  SELECT RAISE(ROLLBACK,'update on raster\_columns violates constraint: r\_raster\_column value must not contain a double quote')  WHERE NEW.r\_raster\_column LIKE ('%"%');  SELECT RAISE(ROLLBACK,'update on raster\_columns violates constraint: r\_raster\_column value must be lower case')  WHERE NEW.r\_raster\_column <> lower(NEW.r\_raster\_column);  CREATE TRIGGER 'raster\_columns\_georectification\_insert'  BEFORE INSERT ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''raster\_columns'' violates constraint: georectification must be 0, 1 or 2')  WHERE (NOT (NEW.georectification IN (0, 1, 2)));  END  CREATE TRIGGER 'raster\_columns\_georectification\_update'  BEFORE UPDATE OF georectification ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''raster\_columns'' violates constraint: georectification must be 0, 1 or 2')  WHERE (NOT (NEW.georectification IN (0, 1, 2)));  END  CREATE TRIGGER 'raster\_columns\_compr\_qual\_factor\_insert'  BEFORE INSERT ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''raster\_columns'' violates constraint: compr\_qual\_factor < 1, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor < 1;  SELECT RAISE(ROLLBACK, 'insert on table ''raster\_columns'' violates constraint: compr\_qual\_factor > 100, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor > 100;  END  CREATE TRIGGER 'raster\_columns\_compr\_qual\_factor\_update'  BEFORE UPDATE OF compr\_qual\_factor ON 'raster\_columns'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''raster\_columns'' violates constraint: compr\_qual\_factor < 1, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor < 1;  SELECT RAISE(ROLLBACK, 'update on table ''raster\_columns'' violates constraint: compr\_qual\_factor > 100, must be between 1 and 100')  WHERE NEW.compr\_qual\_factor > 100;  END  END;  DROP DROP TRIGGER IF EXISTS 'tile\_table\_metadata\_t\_table\_name\_insert';  CREATE TRIGGER 'tile\_table\_metadata\_t\_table\_name\_insert'  BEFORE INSERT ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_table\_metadata'' violates constraint:t\_table\_name not in raster\_columns.r\_table\_name values')  WHERE NOT (NEW.t\_table\_name IN  (SELECT DISTINCT r\_table\_name FROM raster\_columns));  END;  DROP DROP TRIGGER IF EXISTS 'tile\_table\_metadata\_t\_table\_name\_update';  CREATE TRIGGER 'tile\_table\_metadata\_t\_table\_name\_update'  BEFORE UPDATE OF t\_table\_name ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_table\_metadata'' violates constraint:t\_table\_name not in raster\_columns.r\_table\_name values')  WHERE NOT (NEW.t\_table\_name IN  (SELECT DISTINCT r\_table\_name FROM raster\_columns));  END;  DROP DROP TRIGGER IF EXISTS 'tile\_table\_metadata\_is\_times\_two\_zoom\_insert';  CREATE TRIGGER 'tile\_table\_metadata\_is\_times\_two\_zoom\_insert'  BEFORE INSERT ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on tile\_table\_metadata violates constraint: is\_time\_two\_zoom must be one of 0|1')  WHERE NOT(NEW.is\_times\_two\_zoom IN (0,1));  END;  DROP DROP TRIGGER IF EXISTS 'tile\_table\_metadata\_is\_times\_two\_zoom\_update';  CREATE TRIGGER 'tile\_table\_metadata\_is\_times\_two\_zoom\_update'  BEFORE UPDATE OF is\_times\_two\_zoom ON 'tile\_table\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update of tile\_table\_metadata violates constraint: is\_time\_two\_zoom must be one of 0|1')  WHERE NOT(NEW.is\_times\_two\_zoom IN (0,1));  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_zoom\_level\_insert';  CREATE TRIGGER 'tile\_matrix\_metadata\_zoom\_level\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:zoom\_level cannot be less than 0')  WHERE (NEW.zoom\_level < 0);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_zoom\_level\_update';  CREATE TRIGGER 'tile\_matrix\_metadata\_zoom\_level\_update'  BEFORE UPDATE of zoom\_level ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:zoom\_level cannot be less than 0')  WHERE (NEW.zoom\_level < 0);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_matrix\_width\_insert';  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_width\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:matrix\_width cannot be less than 1')  WHERE (NEW.matrix\_width < 1);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_matrix\_width\_update';  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_width\_update'  BEFORE UPDATE OF matrix\_width ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:matrix\_width cannot be less than 1')  WHERE (NEW.matrix\_width < 1);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_matrix\_height\_insert';  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_height\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:matrix\_height cannot be less than 1')  WHERE (NEW.matrix\_height < 1);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_matrix\_height\_update';  CREATE TRIGGER 'tile\_matrix\_metadata\_matrix\_height\_update'  BEFORE UPDATE OF matrix\_height ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:matrix\_height cannot be less than 1')  WHERE (NEW.matrix\_height < 1);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_pixel\_x\_size\_insert';  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_x\_size\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:pixel\_x\_size must be greater than 0')  WHERE NOT (NEW.pixel\_x\_size > 0);  END  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_pixel\_x\_size\_update';  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_x\_size\_update'  BEFORE UPDATE OF pixel\_x\_size ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:pixel\_x\_size must be greater than 0')  WHERE NOT (NEW.pixel\_x\_size > 0);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_pixel\_y\_size\_insert';  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_y\_size\_insert'  BEFORE INSERT ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table ''tile\_matrix\_metadata'' violates constraint:pixel\_y\_size must be greater than 0')  WHERE NOT (NEW.pixel\_y\_size > 0);  END;  DROP DROP TRIGGER IF EXISTS 'tile\_matrix\_metadata\_pixel\_y\_size\_update';  CREATE TRIGGER 'tile\_matrix\_metadata\_pixel\_y\_size\_update'  BEFORE UPDATE OF pixel\_y\_size ON 'tile\_matrix\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table ''tile\_matrix\_metadata'' violates constraint:pixel\_y\_size must be greater than 0')  WHERE NOT (NEW.pixel\_y\_size > 0);  EN  DROP DROP TRIGGER IF EXISTS 'xml\_metadata\_md\_scope\_insert';  CREATE TRIGGER 'xml\_metadata\_md\_scope\_insert'  BEFORE INSERT ON 'xml\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table xml\_metadata violates constraint: md\_scope must be one of undefined | fieldSession | collectionSession | series | dataset | featureType | feature | attributeType | attribute | tile | model | catalogue | schema | taxonomy software | service | collectionHardware | nonGeographicDataset | dimensionGroup')  WHERE NOT(NEW.md\_scope IN ('undefined','fieldSession','collectionSession','series','dataset',  'featureType','feature','attributeType','attribute','tile','model',  'catalogue','schema','taxonomy','software','service',  'collectionHardware','nonGeographicDataset','dimensionGroup'));  END ;  DROP DROP TRIGGER IF EXISTS 'xml\_metadata\_md\_scope\_update';  CREATE TRIGGER 'xml\_metadata\_md\_scope\_update'  BEFORE UPDATE OF 'md\_scope' ON 'xml\_metadata'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table xml\_metadata violates constraint: md\_scope must be one of undefined | fieldSession | collectionSession | series | dataset | featureType | feature | attributeType | attribute | tile | model | catalogue | schema | taxonomy software | service | collectionHardware | nonGeographicDataset | dimensionGroup')  WHERE NOT(NEW.md\_scope IN ('undefined','fieldSession','collectionSession','series','dataset',  'featureType','feature','attributeType','attribute','tile','model',  'catalogue','schema','taxonomy','software','service',  'collectionHardware','nonGeographicDataset','dimensionGroup'));  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_reference\_scope\_insert';  CREATE TRIGGER 'metadata\_reference\_reference\_scope\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: reference\_scope must be one of "table", "column", "row", "row/col"')  WHERE NOT NEW.reference\_scope IN ('table','column','row','row/col');  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_reference\_scope\_update';  CREATE TRIGGER 'metadata\_reference\_reference\_scope\_update'  BEFORE UPDATE OF 'reference\_scope' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: referrence\_scope must be one of "table", "column", "row", "row/col"')  WHERE NOT NEW.reference\_scope IN ('table','column','row','row/col');  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_table\_name\_insert';  CREATE TRIGGER 'metadata\_reference\_table\_name\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: table\_name must be the name of a table in geometry\_columns or raster\_columns')  WHERE NOT NEW.table\_name IN (  SELECT f\_table\_name AS table\_name FROM geometry\_columns  UNION ALL  SELECT r\_table\_name AS table\_name FROM raster\_columns);  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_table\_name\_update';  CREATE TRIGGER 'metadata\_reference\_table\_name\_update'  BEFORE UPDATE OF 'table\_name' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: table\_name must be the name of a table in geometry\_columns or raster\_columns')  WHERE NOT NEW.table\_name IN (  SELECT f\_table\_name AS table\_name FROM geometry\_columns  UNION ALL  SELECT r\_table\_name AS table\_name FROM raster\_columns);  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_column\_name\_insert';  CREATE TRIGGER 'metadata\_reference\_column\_name\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: column name must be "undefined" when reference\_scope is "table" or "row"')  WHERE (NEW.reference\_scope IN ('table','row')  AND NEW.column\_name <> 'undefined');  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: column name must be defined for the specified table when reference\_scope is "column" or "row/col"')  WHERE (NEW.reference\_scope IN ('column','row/col')  AND NOT NEW.table\_name IN (  SELECT name FROM SQLITE\_MASTER WHERE type = 'table'  AND name = NEW.table\_name  AND sql LIKE ('%' || NEW.column\_name || '%')));  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_column\_name\_update';  CREATE TRIGGER 'metadata\_reference\_column\_name\_update'  BEFORE UPDATE OF column\_name ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: column name must be "undefined" when reference\_scope is "table" or "row"')  WHERE (NEW.reference\_scope IN ('table','row')  AND NEW.column\_name <> 'undefined');  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: column name must be defined for the specified table when reference\_scope is "column" or "row/col"')  WHERE (NEW.reference\_scope IN ('column','row/col')  AND NOT NEW.table\_name IN (  SELECT name FROM SQLITE\_MASTER WHERE type = 'table'  AND name = NEW.table\_name  AND sql LIKE ('%' || NEW.column\_name || '%')));  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_row\_id\_value\_insert;  CREATE TRIGGER 'metadata\_reference\_row\_id\_value\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: row\_id\_value must be 0 when reference\_scope is "table" or "column"')  WHERE NEW.reference\_scope IN ('table','column')  AND NEW.row\_id\_value <> 0;  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: row\_id\_value must exist in specified table when reference\_scope is "row" or "row/col"')  WHERE NEW.reference\_scope IN ('row','row/col')  AND NOT EXISTS (SELECT rowid  FROM (SELECT NEW.table\_name AS table\_name) WHERE rowid = NEW.row\_id\_value);  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_row\_id\_value\_update';  CREATE TRIGGER 'metadata\_reference\_row\_id\_value\_update'  BEFORE UPDATE OF 'row\_id\_value' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: row\_id\_value must be 0 when reference\_scope is "table" or "column"')  WHERE NEW.reference\_scope IN ('table','column')  AND NEW.row\_id\_value <> 0;  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: row\_id\_value must exist in specified table when reference\_scope is "row" or "row/col"')  WHERE NEW.reference\_scope IN ('row','row/col')  AND NOT EXISTS (SELECT rowid  FROM (SELECT NEW.table\_name AS table\_name) WHERE rowid = NEW.row\_id\_value);  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_timestamp\_insert';  CREATE TRIGGER 'metadata\_reference\_timestamp\_insert'  BEFORE INSERT ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'insert on table metadata\_reference violates constraint: timestamp must be a valid time in ISO 8601 "yyyy-mm-ddThh-mm-ss.cccZ" form')  WHERE NOT (NEW.timestamp GLOB '[1-2][0-9][0-9][0-9]-[0-1][0-9]-[1-3][0-9]T[0-2][0-9]:[0-5][0-9]:[0-5][0-9].[0-9][0-9][0-9]Z'  AND strftime('%s',NEW.timestamp) NOT NULL);  END;  DROP DROP TRIGGER IF EXISTS 'metadata\_reference\_timestamp\_update';  CREATE TRIGGER 'metadata\_reference\_timestamp\_update'  BEFORE UPDATE OF 'timestamp' ON 'metadata\_reference'  FOR EACH ROW BEGIN  SELECT RAISE(ROLLBACK, 'update on table metadata\_reference violates constraint: timestamp must be a valid time in ISO 8601 "yyyy-mm-ddThh-mm-ss.cccZ" form')  WHERE NOT (NEW.timestamp GLOB '[1-2][0-9][0-9][0-9]-[0-1][0-9]-[1-3][0-9]T[0-2][0-9]:[0-5][0-9]:[0-5][0-9].[0-9][0-9][0-9]Z'  AND strftime('%s',NEW.timestamp) NOT NULL);  END; |

Annex C – Hierarchical Metadata Examples (informative)

The first example use case is from ISO19115 H.2.

Suppose we have this metadata:

CREATE TABLE xml\_metadata2 (  
  id INTEGER NOT NULL PRIMARY KEY,  
  md\_scope TEXT NOT NULL DEFAULT 'undefined',

  metadata\_standard\_URI TEXT NOT NULL DEFAULT '<http://schemas.opengis.net/iso/19139/>',  
  metadata BLOB NOT NULL DEFAULT (zeroblob(4))  
)

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **md\_scope** | **metadata\_standard\_uri** | **metadata** |
| 0 | undefined | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 3 | series | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 4 | dataset | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 5 | featureType | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 6 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 7 | attributeType | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 8 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |

and this reference table definition:  
  
CREATE TABLE metadata\_reference (

reference\_scope TEXT NOT NULL DEFAULT "table",

table\_name TEXT NOT NULL DEFAULT "undefined",

column\_name TEXT NOT NULL DEFAULT "undefined",

row\_id\_value INTEGER NOT NULL DEFAULT 0,

timestamp TEXT NOT NULL DEFAULT (strftime('%Y-%m-%dT%H:%M:%fZ',CURRENT\_TIMESTAMP)),

md\_file\_id INTEGER NOT NULL DEFAULT 0,

md\_parent\_id INTEGER NOT NULL DEFAULT 0,

CONSTRAINT crmr\_mfi\_fk FOREIGN KEY (md\_file\_id) REFERENCES xml\_metadata(id),

CONSTRAINT crmr\_mpi\_fk FOREIGN KEY (md\_parent\_id) REFERENCES xml\_metadata(id)

)  
  
H.2   1) Consider a geographic data provider generating vector mapping data for three Administrative areas(A, B and C).  ... the metadata could be carried exclusively at Dataset Series level.  
  
Then we need a record for each layer table for the three admin areas, like this:  
  
INSERT INTO metadata\_reference VALUES (  
'table', /\* reference type \*/  
'roads', /\* table name \*/  
'undefined', /\* column\_name \*/  
-1, /\* row\_id\_value  \*/  
(datetime('now')),  
3,  /\* md\_file\_id  \*/  
0  /\* md\_parent\_id  \*/  
)  
  
H.2 2) After some time alternate vector mapping of Administrative area A becomes available. The metadata would then be extended for Administrative area A, to describe the new quality date values. These values would supersede those given for the Dataset series, but only for Administrative area A. The metadata for B and C would remain unchanged. This new metadata would be recorded at Dataset  
level.  
  
Then we need a record for each layer table in "A" like this:  
  
INSERT INTO metadata\_reference VALUES (  
'table', /\* reference type \*/  
'roads', /\* table name \*/  
'undefined', /\* column\_name \*/  
-1, /\* row\_id\_value  \*/  
(datetime('now')),  
 4,  /\* md\_file\_id  \*/  
3  /\* md\_parent\_id  \*/  
)  
  
H.2 3) Eventually further data becomes available for Administrative area A, with a complete re-survey of the road network. Again this implies new metadata for the affected feature types. This metadata would be carried at Feature type level for Administrative area A. All other metadata relating to other feature types remains unaffected. Only the metadata for roads in Administrative area A is modified. This road metadata is recorded at Feature type level.  
  
Then we need a record for each layer table for the roads network, like this:  
  
INSERT INTO metadata\_reference VALUES (  
'table', /\* reference type \*/  
'roads', /\* table name \*/  
'undefined', /\* column\_name \*/  
-1, /\* row\_id\_value  \*/  
(datetime('now')),  
5,  /\* md\_file\_id  \*/  
4  /\* md\_parent\_id  \*/  
)  
  
H.2 4) An anomaly in the road survey is identified, in that all Overhead clearances for the Administrative area A have been surveyed to the nearest metre. These are re-surveyed to the nearest decimetre. This re-survey implies new metadata for the affected attribute type ‘Overhead Clearance’. All other metadata for Administrative area A remains unaffected. This ‘Overhead Clearance’ metadata is recorded at Attribute Type level  
  
Then we need a record for each layer table in the roads network with attribute type 'Overhead Clearance', like this;  
  
INSERT INTO metadata\_reference VALUES (  
'column', /\* reference type \*/  
'roads', /\* table name \*/  
'overhead\_clearance', /\* column\_name \*/  
-1, /\* row\_id\_value  \*/  
(datetime('now')),  
7,  /\* md\_file\_id  \*/  
4  /\* md\_parent\_id  \*/  
)  
  
H.2 5) A new bridge is constructed in Administrative area A. This new data is reflected in the geographic data for Administrative area A, and new metadata is required to record this new feature. All other metadata for Administrative area A remains unaffected. This new feature metadata is recorded at Feature instance level.  
  
Then we need a record for the bridge layer table row for the new bridge, like this:  
  
INSERT INTO metadata\_reference VALUES (  
'row', /\* reference type \*/  
'bridge', /\* table name \*/  
'undefined', /\* column\_name \*/  
987, /\* row\_id\_value  \*/  
(datetime('now')),  
6,  /\* md\_file\_id  \*/  
4  /\* md\_parent\_id  \*/  
)  
  
H.2  6) The overhead clearance attribute of the new bridge was wrongly recorded, and is modified. Again this new attribute requires new metadata to describe the modification. All other metadata for  
Administrative area A remains unaffected. This new attribute metadata is recorded at Attribute  
instance level.  
  
Then we need a record for the clearance attribute value, like this:  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'bridge', /\* table name \*/  
'overhead\_clearance', /\* column\_name \*/  
987, /\* row\_id\_value  \*/  
(datetime('now')),  
8,  /\* md\_file\_id  \*/  
4  /\* md\_parent\_id  \*/  
)  
  
The second example use case is for a field data collection session. This use case demonstrates a mechanism to indicate which data in a GeoPackage that was originally loaded with data from one or more services has been collected or updated since the initial load, and therefore may need to be uploaded to update the original services (e.g. WFS, WCS, WMTS).   
  
Suppose a user with a mobile handheld device goes out in the field and collects observations of a new "Point of Interest" (POI) feature type, and associated metadata  about the field session, the new feature type, some POI instances and some of their attributes (e.g. spatial accuracy, attribute accuracy) that results in the following additional metadata:

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **md\_scope** | **metadata\_standard\_uri** | **metadata** |
| 0 | undefined | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 1 | fieldSession | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 10 | featureType | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 11 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 12 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 13 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 14 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 15 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 16 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 17 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 18 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 19 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |

(This example assumes that the field session data is still considered "raw" and won't be considered a data set or part of a data series until it has been verified and cleaned, but if that is wrong then additional series and data set metadata could be added.)  
  
Then we need a metadata\_reference record for the field session for the new POI table, whose md\_parent\_id is undefined:  
  
INSERT INTO metadata\_reference VALUES (  
'table', /\* reference type \*/  
'poi', /\* table name \*/  
'undefined', /\* column\_name \*/  
-1, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
1,  /\* md\_file\_id  \*/  
0  /\* md\_parent\_id  \*/  
)  
  
Then we need a metadata\_reference record for the feature type for the new POI table, whose md\_parent\_id is that of the field session:  
  
INSERT INTO metadata\_reference VALUES (  
'table', /\* reference type \*/  
'poi', /\* table name \*/  
'undefined', /\* column\_name \*/  
-1, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
10,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
Then we need metadata\_reference records for the poi feature instance rows, whose md\_parent\_id is that of the field session:  
  
INSERT INTO metadata\_reference VALUES (  
'row', /\* reference type \*/  
'poi', /\* table name \*/  
'undefined', /\* column\_name \*/  
1, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
11,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row', /\* reference type \*/  
'poi', /\* table name \*/  
'undefined', /\* column\_name \*/  
2, /\* row\_id\_value  \*/  
14,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row', /\* reference type \*/  
'poi', /\* table name \*/  
'undefined', /\* column\_name \*/  
3, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
17,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
And finally we need metadata\_reference records for the poi attribute instance metadata , whose md\_parent\_id is that of the field session:  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'poi', /\* table name \*/  
'point', /\* column\_name \*/  
1, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
12,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'poi', /\* table name \*/  
'point', /\* column\_name \*/  
2, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
15,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'poi', /\* table name \*/  
'point', /\* column\_name \*/  
3, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
18,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'poi', /\* table name \*/  
'category', /\* column\_name \*/  
1, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
13,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'poi', /\* table name \*/  
'category', /\* column\_name \*/  
2, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
16,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
INSERT INTO metadata\_reference VALUES (  
'row/col', /\* reference type \*/  
'poi', /\* table name \*/  
'category', /\* column\_name \*/  
3, /\* row\_id\_value  \*/  
(strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)),  
19,  /\* md\_file\_id  \*/  
1  /\* md\_parent\_id  \*/  
)  
  
    As long as all metadata collected in the field session either directly (as above) or indirectly (suppose there were a data set level metadata\_reference record intermediary) refers to the field session metadata via md\_parent\_id values, then this chain of metadata references identifies the newly collected information, as Joan requested, in addition to the metadata.

So here is the data after both examples:  
  
xml\_metadata

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **md\_scope** | **metadata\_standard\_uri** | **metadata** |
| 0 | undefined | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 1 | fieldSession | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 2 | collectionSession | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 3 | series | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 4 | dataset | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 5 | featureType | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 6 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 7 | attributeType | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 8 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 10 | featureType | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 11 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 12 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 13 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 14 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 15 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 16 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 17 | feature | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 18 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |
| 19 | attribute | <http://schemas.opengis.net/iso/19139/> | BLOB |

metadata\_reference

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **reference\_type** | **table\_name** | **column\_name** | **row\_id\_value** | **timestamp** | **md\_file\_id** | **md\_parent\_id** |
| table | roads | undefined | 0 | ts | 3 | 0 |
| table | roads | undefined | 0 | ts | 4 | 3 |
| table | roads | undefined | 0 | ts | 5 | 4 |
| column | roads | overhead\_clearance | 0 | ts | 7 | 4 |
| row | bridge | undefined | 987 | ts | 6 | 4 |
| row/col | bridge | overhead\_clearance | 987 | ts | 8 | 4 |
| table | poi | undefined | 0 | ts | 1 | 0 |
| row | poi | undefined | 0 | ts | 10 | 1 |
| row | poi | undefined | 1 | ts | 11 | 1 |
| row | poi | undefined | 2 | ts | 14 | 1 |
| row/col | poi | undefined | 3 | ts | 17 | 1 |
| row/col | poi | point | 1 | ts | 12 | 1 |
| row/col | poi | point | 2 | ts | 15 | 1 |
| row/col | poi | point | 3 | ts | 18 | 1 |
| row/col | poi | category | 1 | ts | 13 | 1 |
| row/col | poi | category | 2 | ts | 16 | 1 |
| row/col | poi | category | 3 | ts | 19 | 1 |

Bibliography (informative)

B.1 General

[B1] [http://developer.android.com/guide/topics/data/data-storage.html#db](http://developer.android.com/guide/topics/data/data-storage.html" \l "db)

[B2] <https://developer.apple.com/technologies/ios/data-management.html>

[B3] <http://www.epsg.org/guides/docs/G7-1.pdf>

[B4] <http://www.gdal.org/ogr/>

[B5] <http://www.gdal.org/ogr/drv_sqlite.html>

[B6] <http://trac.osgeo.org/geos/wiki/Applications>

[B7] <http://www.qgis.org/>

[B8] <http://hub.qgis.org/projects/android-qgis>

[B9] <http://www.luciad.com/products/LuciadMobile>

[B10] <http://www.falconview.org/trac/FalconView/downloads/26>

[B11] <http://wiki.openstreetmap.org/wiki/TMS>

[B12] <https://github.com/mapbox/mbtiles-spec>

[B13] <https://www.gaia-gis.it/fossil/librasterlite/index>

[B14] <http://wiki.openstreetmap.org/wiki/Main_Page>

[B15] <http://code.google.com/p/osmdroid/>

[B16] <http://www.falconview.org/trac/FalconView>

[B17] <http://code.google.com/p/big-planet-tracks/>

[B18] <http://en.wikipedia.org/wiki/ASCII>

[B19] “SQL for Smarties: Advanced SQL Programming” Joe Selko, Morgan Kaufmann, 1995, ISBN 1-55860-323-9

[B20] NATO IIRS STANAG, NATO Imagery Interpretability Rating Scale (NIIRS)  STANAG 7194 Edition 1 2009

<https://nsgreg.nga.mil/doc/view?i=2129>

[B21] U.S. National Image Interpretability Rating Scales<http://www.fas.org/irp/imint/niirs.htm>

[B22] Civil NIIRS <http://www.fas.org/irp/imint/niirs_c/index.html>

[B23] Civil NIIRS Reference Guide <http://www.fas.org/irp/imint/niirs_c/guide.htm>

[B24] Additional Civil NIIRS Criteria <http://www.fas.org/irp/imint/niirs_c/app2.htm>

[B25] Sample Civil NIIRS Images <http://www.fas.org/irp/imint/niirs_c/append.htm>

[B26] History of NIIRS <http://www.fas.org/irp/imint/niirs_c/app3.htm>

[B27] <http://www.ucgis.org/priorities/research/research_white/1998%20Papers/data.html>

[B28] <http://www.gdal.org/frmt_rasterlite.html>

[B29] <https://www.gaia-gis.it/fossil/libspatialite/wiki?name=switching-to-4.0>

[B30] <http://www.sqlite.org/lang_createtable.html#rowid>

[B31] ISO 19115-2 Geographic information - - Metadata - Part 2: Metadata for imagery and gridded data

[B32] ISO 19139: Geographic information -- Metadata -- XML schema implementation

[B33] Dublin Core Metadata Initiative <http://dublincore.org/> [IETF RFC 5013](http://www.ietf.org/rfc/rfc5013.txt)

ISO 15836:2009 <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=52142>

[B34] Content Standard for Digital Geospatial Metadata (CSDGM)

<http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index_html>

[B35] Department of Defense Discovery Metadata Specification (DDMS) <http://metadata.ces.mil/mdr/irs/DDMS/>

[B36] NMF [NGA.STND.0012\_2.0](https://nsgreg.nga.mil/doc/view?i=2142) / NMIS [NGA.STND.0018\_1.0](https://nsgreg.nga.mil/doc/view?i=1243)

[B37] Unified Modeling Language (UML) <http://www.uml.org/>

[B38] XML for Metadata Interchange (XMI) <http://www.omg.org/spec/XMI/>

[B39] Geography Markup Language (GML) [ISO 19136:2007](https://nsgreg.nga.mil/doc/view?i=1040)

[B40] ISO 19110 Geographicinformation – Methodology for feature cataloguing

[B41] Web Ontology Language (OWL) <http://www.w3.org/TR/2009/REC-owl2-xml-serialization-20091027/>

[B42] Simple Knowledge Organization System (SKOS) <http://www.w3.org/TR/skos-reference/>

[B43] MIL-STD-2500C DoD Interface Standard: National Imagery Transmission Format (NITF)

<https://nsgreg.nga.mil/NSGDOC/files/doc/Document/MIL-STD-2500C.pdf>

[B44] STANAG 7074 Digital Geographic Information Exchange Standard (DIGEST) - AGeoP-3A, edition 1, 19 October 1994 <http://www.dgiwg.org/dgiwg/htm/documents/historical_documents.htm>

[B45] ISO 19109 Geographic information - Rules for application schema

1. Mandatory unless there is an Author element for every Entry. [↑](#footnote-ref-1)
2. Mandatory if the Feed element does not contain an Author child element [↑](#footnote-ref-2)
3. Mandatory unless TableType is “other” [↑](#footnote-ref-3)
4. Mandatory unless TableType is “other” [↑](#footnote-ref-4)
5. Mandatory unless TableType is “other” [↑](#footnote-ref-5)
6. Mandatory unless TableType is “other” [↑](#footnote-ref-6)