SLOVENSKI STANDARD
SIST ISO 19125-2:2005

01-februar-2005

Geografske informacije – Dostop do enostavnih pojavov – 2. del: Možnost SQL

Geographic information -- Simple feature access -- Part 2: SQL option

Information géographique -- Accès aux entités simples -- Partie 2: Option SQL

Ta slovenski standard je istoveten z: ISO 19125-2:2004

ICS:

35.240.70	Uporabniške rešitve IT v znanosti	IT applications in science

SIST ISO 19125-2:2005 **en**

INTERNATIONAL STANDARD

ISO 19125-2

First edition
2004-08-01

Geographic information — Simple feature access —

Part 2: **SQL option**

Information géographique — Accès aux entités simples —
Partie 2: Option SQL

Reference number
ISO 19125-2:2004(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 19125-2 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics* from a base document supplied by the Open GIS Consortium, Inc.

ISO 19125 consists of the following parts, under the general title *Geographic information — Simple feature access*:

— *Part 1: Common architecture*

— *Part 2: SQL option*

Part 3: COM/OLE option is under preparation.

Introduction

The purpose of this part of ISO 19125 is to 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) (ISO/IEC 9075-3:2003). A feature has both spatial and non-spatial attributes. Spatial attributes are geometry valued, and simple features are based on 2D geometry with linear interpolation between vertices. This part of ISO 19125 is dependent on the common architectural components defined in ISO 19125-1.

Feature collections are stored as tables with geometry valued columns in a SQL-implementation; each feature is a row in the table. The non-spatial attributes of features are mapped onto columns whose types are drawn from the set of standard SQL data types. The spatial attributes of features are mapped onto columns whose SQL data types are based on the underlying concept of additional geometric data types for SQL. A table whose rows represent these features is referred to as a feature table. Such a table contains one or more geometry valued columns. Feature-table schemas are described for two SQL-implementations: implementations based on predefined data types and SQL with Geometry Types.

In an implementation based on predefined data types, a geometry-valued column is implemented as a Foreign Key reference into a geometry table. A geometry value is stored using one or more rows in the geometry table. The geometry table may be implemented using either standard SQL numeric types or SQL binary types; schemas for both are described.

The term SQL with Geometry Types is used to refer to a SQL-implementation that has been extended with a set of Geometry Types. In this environment, a geometry-valued column is implemented as a column whose SQL type is drawn from this set of Geometry Types. The mechanism for extending the type system of an SQL-implementation is through the definition of user defined User Defined Types. Commercial SQL-implementations with user defined type support have been available since mid-1997.

Geographic information — Simple feature access —

Part 2:
SQL option

1 Scope

This part of ISO 19125 specifies an SQL schema that supports storage, retrieval, query and update of simple geospatial feature collections via the SQL Call Level Interface (SQL/CLI) (ISO/IEC 9075-3:2003).

This part of ISO 19125 establishes an architecture for the implementation of feature tables.

This part of ISO 19125 defines terms to use within the architecture.

This part of ISO 19125 defines a simple feature profile of ISO 19107.

This part of ISO 19125 describes a set of SQL Geometry Types together with SQL functions on those types. The Geometry Types and Functions described in this part of ISO 19125 represent a profile of ISO 13249-3.

This part of ISO 19125 does not attempt to standardize and does not depend upon any part of the mechanism by which Types are added and maintained in the SQL environment including the following:

a) the syntax and functionality provided for defining types;

b) the syntax and functionality provided for defining SQL functions;

c) the physical storage of type instances in the database;

d) specific terminology used to refer to User Defined Types, for example, UDT.

This part of ISO 19125 does standardize:

— names and geometric definitions of the SQL Types for Geometry;

— names, signatures and geometric definitions of the SQL Functions for Geometry.

This part of ISO 19125 describes a feature access implementation in SQL based on a profile of ISO 19107. ISO 19107 does not place any requirements on how to define the Geometry Types in the internal schema. ISO 19107 does not place any requirements on when or how or who defines the Geometry Types. In particular, a compliant system may be shipped to the database user with the set of Geometry Types and Functions already built into the SQL-implementation, or with the set of Geometry Types and Functions supplied to the database user as a dynamically loaded extension to the SQL-implementation or in any other manner not mentioned in this part of ISO 19125.

2 Conformance

In order to conform to this part of ISO 19125, an implementation shall satisfy the requirements of one of the following three conformance classes, as well as the appropriate components of ISO 19125-1:

a) SQL implementation of feature tables based on predefined data types:

1) using numeric SQL types for geometry storage and SQL/CLI access,

2) using binary SQL types for geometry storage and SQL/CLI access;

b) SQL with Geometry Types implementation of feature tables supporting both textual and binary SQL/CLI access to geometry.

Annex B provides conformance tests for each implementation of this part of ISO 19125.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9075-1:2003, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)*

ISO/IEC 9075-2:2003, *Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)*

ISO/IEC 9075-3:2003, *Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI)*

ISO/IEC 9075-4:2003, *Information technology — Database languages — SQL — Part 4: Persistent Stored Modules (SQL/PSM)*

ISO/IEC 9075-5:1999, *Information technology — Database languages — SQL — Part 5: Host Language Bindings (SQL/Bindings)*

ISO/IEC 13249-3:2003, *Information technology — Database languages — SQL multimedia and application packages — Part 3: Spatial*

ISO 19107:2003, *Geographic information — Spatial schema*

ISO 19109:—[1)], *Geographic information — Rules for application schema*

ISO 19119:2004, *Geographic information — Services*

ISO 19125-1:2004, *Geographic information — Simple feature access — Part 1: Common architecture*

4 Terms and definitions

For the purposes of this part of ISO 19125, the following terms and definitions apply.

4.1
feature table
table where the columns represent feature attributes, and the rows represent features

1) To be published.

4.2
geographic feature
representation of real world phenomenon associated with a location relative to the Earth

5 Symbols and abbreviated terms

FID	Feature ID column in the implementation of feature tables based on predefined data types
GID	Geometry ID column in the implementation of feature tables based on predefined data types
MM	Multimedia
SQL	Structured Query Language
SRID	Spatial Reference System Identifier
SRTEXT	Spatial Reference System Well Known Text
WKB	Well-Known Binary (representation for example, geometry)
WKTR	Well-Known Text Representation
2D	2-Dimensional
$\Re^1$	1-Dimensional space
$\Re^2$	2-Dimensional space
$\emptyset$	empty set
$\cap$	intersection
$\cup$	union
—	difference
$\in$	is a member of
$\notin$	is not a member of
$\subset$	is a proper subset of
$\subseteq$	is a subset of
$\Leftrightarrow$	if and only if
$\Rightarrow$	implies
$\forall$	for all
$\{ X \mid \ldots \}$	set of X such that…
$\wedge$	and
$\vee$	or
$\neg$	not
$=$	equal
$\neq$	not equal
$<$	less than
$>$	greater than

6 Architecture

6.1 Architecture — SQL implementation of feature tables based on predefined data types

6.1.1 Overview

This part of ISO 19125 defines a schema for the management of feature table, Geometry, and Spatial Reference System information in an SQL-implementation based on predefined data types. This part of ISO 19125 does not define SQL functions for access, maintenance, or indexing of Geometry in an SQL-implementation based on predefined data types.

Figure 1 illustrates the schema to support feature tables, Geometry, and Spatial Reference Information in an SQL-implementation based on predefined data types.

a) The GEOMETRY_COLUMNS table describes the available feature tables and their Geometry properties.

b) The SPATIAL_REF_SYS table describes the coordinate system and transformations for Geometry.

c) The feature table stores a collection of features. A feature table's columns represent feature attributes, while rows represent individual features. The Geometry of a feature is one of its feature attributes; while logically a geometric data type, a Geometry Column is implemented as a foreign key to a geometry table.

d) The geometry table stores geometric objects, and may be implemented using either standard SQL numeric types or SQL binary types.

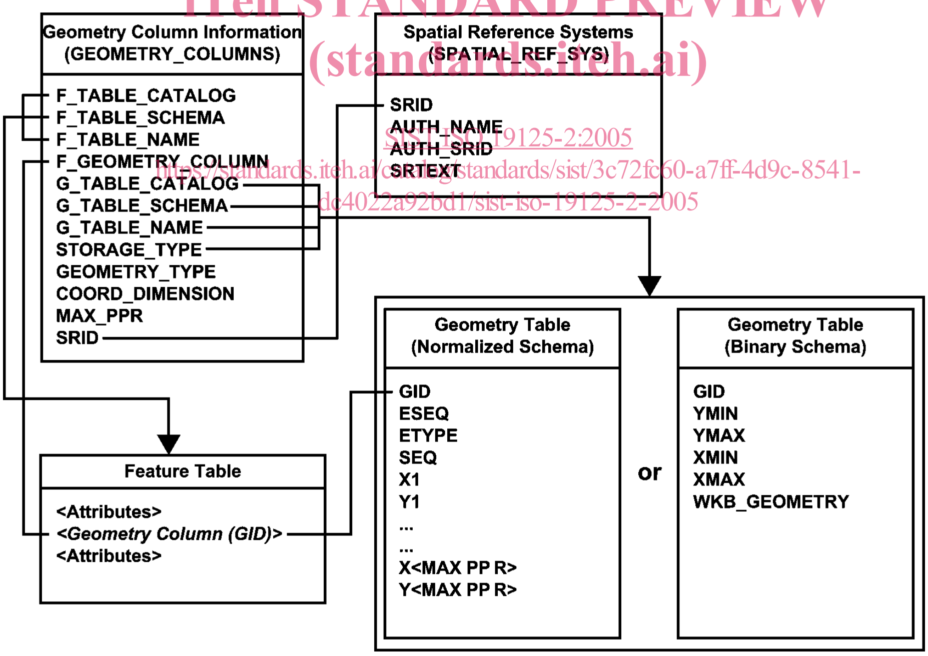


Figure 1 — Schema for feature tables using predefined data types

Depending upon the storage type specified by the GEOMETRY_COLUMNS table, a geometric object is stored either as an array of coordinate values or as a single binary value. In the former case, predefined SQL numeric types are used for the coordinates and these numeric values are obtained from the geometry table until the geometric object has been fully reconstructed. In the latter case, the complete geometric object is obtained in the Well-known Binary Representation as a single value.

6.1.2 Identification of feature tables and geometry columns

Feature tables and Geometry columns are identified through the GEOMETRY_COLUMNS table. Each Geometry Column in the database has an entry in the GEOMETRY_COLUMNS table. The data stored for each geometry column consists of the following:

a) the identity of the feature table of which this Geometry Column is a member;

b) the name of the Geometry Column;

c) the spatial reference system ID (SRID) for the Geometry Column;

d) the type of Geometry for the Geometry column;

e) the coordinate dimension for the Geometry Column;

f) the identity of the geometry table that stores geometric objects for this Geometry Column;

g) the information necessary to navigate the geometry table in the case of normalized geometry storage.

6.1.3 Identification of Spatial Reference Systems

Every Geometry Column is associated with a Spatial Reference System. The Spatial Reference System identifies the coordinate system for all geometric objects stored in the column, and gives meaning to the numeric coordinate values for any geometric object stored in the column. Examples of commonly used Spatial Reference Systems include "Latitude Longitude" and "UTM Zone 10".

The SPATIAL_REF_SYS table stores information on each Spatial Reference System in the database. The columns of this table are the Spatial Reference System Identifier (SRID), the Spatial Reference System Authority Name (AUTH_NAME), the Authority Specific Spatial Reference System Identifier (AUTH_SRID) and the Well-known Text description of the Spatial Reference System (SRTEXT). The Spatial Reference System Identifier (SRID) constitutes a unique integer key for a Spatial Reference System within a database.

Interoperability between clients is achieved via the SRTEXT column which stores the Well-known Text representation for a Spatial Reference System.

6.1.4 Feature tables

A feature is an abstraction of a real-world object. Feature attributes are columns in a feature table. Features are rows in a feature table. The Geometry of a feature is one of its feature attributes; while logically a geometric data type, a geometry column is implemented as a foreign key to a geometry table.

Relationships between features may be defined as foreign key references between feature tables.

6.1.5 Geometry tables

6.1.5.1 Normalized geometry schema

The normalized geometry schema stores the coordinates of geometric objects as predefined SQL numeric types. One or more coordinates (X and Y ordinate values) will be represented by pairs of numeric types in the geometry table, as shown in Figure 2. Each geometric object is identified by a key (GID) and consists of one or more primitive elements ordered by an element sequence (ESEQ). Each primitive element in the geometric object is distributed over one or more rows in the geometry table, identified by a primitive type (ETYPE), and ordered by a sequence number (SEQ).

The rules for geometric object representation in the normalized schema are defined as follows.

a) ETYPE designates the Geometry Type.

b) Geometric objects may have multiple elements. The ESEQ value identifies the individual elements.

c) An element may be built up from multiple parts (rows). The rows and their proper sequence are identified by the SEQ value.

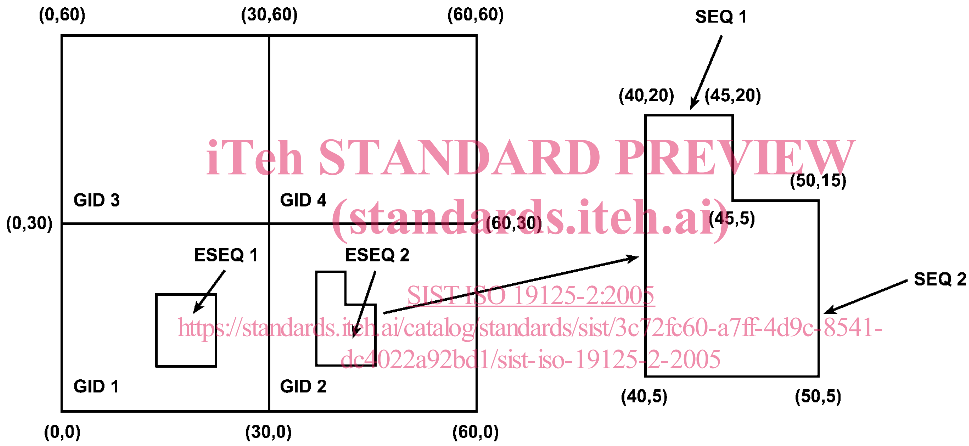
d) Polygons may contain holes, as described in the Geometry object model.

e) PolygonRings shall close when assembled from an ordered list of parts. The SEQ value designates the part order.

f) Coordinate pairs that are not used shall be set to Nil in complete sets (both X and Y). This is the only way to identify the end of the list of coordinates.

g) For geometric objects that continue onto an additional row (as defined by a constant element sequence number or ESEQ), the last Point of one row is equal to the first Point of the next.

h) There is no limit on the number of elements in the geometric object, or the number of rows in an element.



GID 1	ESEQ	ETYPE	SEQ	X0	Y0	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	1	3	1	0	0	0	30	30	30	30	0	0	0
1	2	3	1	10	10	10	20	20	20	20	10	10	10
2	1	3	1	30	0	30	30	60	30	60	0	30	0
2	2	3	1	40	5	40	20	45	20	45	15	50	15
2	2	3	1	50	15	50	5	40	5	Nil	Nil	Nil	Nil
3	1	3	1	0	30	0	60	30	60	30	30	0	30
4	1	3	1	30	30	30	60	60	60	60	30	30	30

Figure 2 — Example of geometry table for Polygon Geometry using SQL

6.1.5.2 Binary geometry schema

The binary Geometry schema is illustrated in Table 1, uses GID as a key and stores the geometric object using the Well-known Binary Representation for Geometry (WKBGeometry). The geometry table includes the minimum bounding rectangle for the geometric object as well as the WKBGeometry for the geometric object. This permits construction of spatial indexes without accessing the actual geometric object structure, if desired.

Table 1 — Example of geometry table for the above Polygon Geometry using the Well-known Binary Representation for Geometry

GID	XMIN	YMIN	XMAX	YMAX	Geometry
1	0	0	30	30	< WKBGeometry >
2	30	0	60	30	< WKBGeometry >
3	0	30	30	60	< WKBGeometry >
4	30	30	60	60	< WKBGeometry >

6.1.6 Use of numeric data types

SQL-implementations usually provide several numeric data types. In this part of ISO 19125, the use of a numeric data type in examples is not meant to be binding. The data type of any particular column can be determined, and casting operators between similar data types are available. Any particular implementation may use alternative data types as long as casting operations shall not lead to difficulties.

6.1.7 Notes on SQL/CLI access to Geometry values stored in binary form

SQL/CLI provides standard mechanisms to bind character, numeric and binary data values.

This subclause describes the process of retrieving geometric object values for the case where the binary storage alternative is chosen.

The WKB_GEOMETRY column in the geometry table is accessed in SQL/CLI as one of the binary SQL data types (SQL_BINARY, SQL_VARBINARY, or SQL_LONGVARBINARY).

EXAMPLE The application would use the SQL_C_BINARY value for the fCType parameter of SQLBindCol (or SQLGetData) in order to describe the application data buffer that shall receive the fetched Geometry data value. Similarly, a dynamic parameter whose value is a Geometry would be described using the SQL_C_BINARY value for the fCType parameter of SQLBindParameter.

This allows binary values to be both retrieved from and inserted into the geometry tables.

6.2 Architecture — SQL with Geometry Types implementation of feature tables

6.2.1 Overview

This part of ISO 19125 defines a schema for the management of feature table, Geometry, and Spatial Reference System information in an SQL-implementation with a Geometry Type extension.

Figure 3 illustrates the schema to support feature tables, Geometry, and Spatial Reference Information in an SQL-implementation with a Geometry Type extension.

a) The GEOMETRY_COLUMNS table describes the available feature tables and their Geometry properties.

b) The SPATIAL_REF_SYS table describes the coordinate system and transformations for Geometry.

c) The feature table stores a collection of features. A feature table's columns represent feature attributes, while rows represent individual features. The Geometry of a feature is one of the feature attributes, and is an SQL Geometry Type.

