Open GIS Consortium, Inc.

Conformance Test Guidelines for OpenGIS Simple Features Specification for SQL, Revision 1.1

OpenGIS Project Document 98-046r2

Release Date: October 2, 1998 Version Date: May 5, 1999 Copyright © 1998, Open GIS Consortium, Inc.

NOTICE

The information contained in this document is subject to change without notice.

The material in this document details an Open GIS Consortium testing program in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of this program in any form.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE OPEN GIS CONSORTIUM MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. The Open GIS Consortium shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

The copyright holders listed above acknowledge that the Open GIS Consortium (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks, or other special designations to indicate compliance with these materials.

This document contains information, which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without permission of the copyright owner.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision (c)(1)(ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013

OpenGIS® is a trademark or registered trademark of Open GIS Consortium, Inc. in the United States and in other countries.

Table of Contents

1	OVERV	/IEW	5
2	TEST D	ESCRIPTION AND SCOPE	5
	2.1 No	RMALIZED GEOMETRY	6
		ARY GEOMETRY	
	2.3 TY	PE AND FUNCTIONS	6
3	TEST D	ATA	7
		T DATA SEMANTICS	
		T DATA POINTS AND COORDINATES	
		T DATA SCHEMA AND VALUES	
	3.3.1 3.3.2	Document Type Definition for the Schema and ValuesLakes	
	3.3.2 3.3.3	Road Segments	
	3.3.3 3.3.4	Divided Routes	
	3.3.5	Forests	
	3.3.6	Bridges	
	3.3.7	Streams	
	3.3.8	Buildings	
	3.3.9	Ponds	
	3.3.10	Named Places	
	3.3.11	Map Neatline	
	3.3.12	Routes Error! Bookmark	
	3.3.13	Route Segment Error! Bookmark	v
	3.3.14	MapsError! Bookmark	v
	3.3.15	Map Components	not defined.
4	CONFO	DRMANCE ITEMS	17
	4.1 No	rmalized Geometry	17
		ARY GEOMETRY	
		PES AND FUNCTIONS	
	4.3.1	Queries for Section 3.2.10.2	
	4.3.2	Queries for Section 3.2.11.2	
	4.3.3	Queries for Section 3.2.12.2	
	4.3.4	Queries for Section 3.2.13.2	
	4.3.5	Queries for Section 3.2.14.2	
	4.3.6	Queries for Section 3.2.15.2	
	4.3.7	Queries for Section 3.2.16.2	
	4.3.8	Queries for Section 3.2.17.2	
	4.3.9	Queries for Section 3.2.18.2	
	4.3.10	Queries for Section 3.2.19.2	
	4.3.11	Queries for Section 3.2.20.2	
	4.3.12	Queries for Section 3.2.21.2	
5	ADAPT	ATIONS	23
	5.1 GE	NERAL ADAPTATION DOCUMENTATION GUIDANCE	23
		NERAL ADAPTATION #1: TABLES AND VIEWS	
	5.2.1	Documentation guidance	
	5.2.2	OGC Intentions Regarding This Adaptation	
		NERAL ADAPTATION #2: NAMING OF TABLES/VIEWS, TYPES, AND FUNCTIONS	
	5.3.1		
	5.5.1	Documentation Guidance	25

Conformance Test Guidelines for OpenGIS Simple Features Specifi	

5.4	NORMALIZED GEOMETRY	25
5.5	BINARY GEOMETRY	25
5.6	Types and Functions	25

1 Overview

This document describes the conformance testing guidelines, that the Open GIS Consortium will use in testing of software implementations for conformance to its specification entitled Open GIS Simple Features for SQL, Revision 1.1.

The Open GIS Consortium, Inc. (OGC) maintains a brand (in the form of a certification mark) that cannot be used in connection with a software product by any organization unless they have submitted a software product to OGC's conformance testing, successfully completed this testing, and received a certificate stating such success. Organizations that have earned the certification mark may use it in ways defined within this document. This set of rules ensures that users who buy products that are branded can be sure that the products carrying the certification mark have been submitted to a testing process. The primary purpose of the conformance testing process is to protect the value of the OpenGIS® brand as an element of OGC's program to promote interoperability between diverse geoprocessing systems.

This document defines the following:

- A general description of the conformance tests and the scope of the tests
- A description of the test data
- A description of the test queries
- A description of allowable adaptations and guidelines for documenting these adaptations

2 Test Description and Scope

According to the OpenGIS Simple Features Specification for SQL, Revision 1.1 specification (hereafter referred to as "the specification"), there are two implementation alternatives and the first has two storage alternatives. The specification reads:

In order to be compliant with this OpenGIS ODBC/SQL specification for geospatial feature collections an implementer can choose to implement any one of three alternatives (1a, 1b or 2) described in this specification:

- 1. SQL92 implementation of feature tables
 - a) using numeric SQL types for geometry storage and ODBC access.
 - b) using binary SQL types for geometry storage and ODBC access.
- 2. **SQL92 with Geometry Types** implementation of feature tables supporting both textual and binary ODBC access to geometry.

Thus there are three conformance alternatives. In terms of the architectural description in the specification, these alternatives are:

- 1. SQL 92 Implementation of Feature Tables, Normalized Geometry Schema (alternative 1a. above, hereafter referred to as Normalized Geometry),
- 2. SQL 92 Implementation of Feature Tables, Binary Geometry Schema (alternative 1b. above, hereafter referred to as Binary Geometry), and
- 3. SQL 92 with Geometry Types Implementation of Feature Tables (alternative 2. above, hereafter referred to as Types and Functions).

Test suite software is provided for each of the above alternative. The test suite software is provided in two ways for each test, as a set of SQL scripts and as embedded C programming language source code. OGC recognizes that this test suite software will probably have to be adapted to work with software products seeking conformance. Organizations whose products are being tested may select either form of test suite for adaptation and testing. See section 5 for allowable adaptations (most of these are also documented within the test suite software itself, as well).

In general, the scope of the tests is to exercise each functional aspect of the specification at least once. The test questions and answers are defined to test that the specified functionality exists and is operable. Care has been taken to ensure that the tests are not at the level of rigor that a product quality control process or certification test might be. However, because some of the answers are further examined for reasonableness (for example, the area of a polygon is tested for correctness to two or three significant figures). The following sections further describe each test alternative.

2.1 Normalized Geometry

The Normalized Geometry portion of the specification defines the structure of feature tables (or views), geometry tables (or views), a geometry column metadata view, and a spatial reference system information view. It also defines the relationships (primary key and foreign key) between these tables (views). More specifically, it defines the detail for the way that geometry is stored in normalized tables (or views) with numeric valued columns, and the textual representation of the spatial reference system to be stored in the spatial reference system information view. This is the entirety of the specification.

The test suite for this alternative contains one possible complete implementation of the specification (because of the fact that the specification defines only table (or view) schema and relationships any SQL92 compliant database should pass this test). OGC expects that implementations will actually manage more of this structure for the user and, therefore, liberal adaptations are allowed (see section 5 for allowable adaptations). What this means, in the final analysis, is that the scope of the test is to determine that the database of the test (once inserted) is accessible via the schema defined in the specification.

2.2 Binary Geometry

The Binary Geometry portion of the specification defines the structure of feature tables (or views), geometry tables (or views), a geometry column metadata view, and a spatial reference system information view. It also defines the relationships (primary key and foreign key) between these tables (views). More specifically, it defines the detail for the way that geometry is stored in geometry tables (or views) with a binary valued column, and the textual representation of the spatial reference system to be stored in the spatial reference system information view. This is the entirety of the specification.

The test suite for this alternative contains one possible complete implementation of the specification (because of the fact that the specification defines only table (or view) schema and relationships any SQL92 compliant database should pass this test). OGC expects that implementations will actually manage more of this structure for the user and, therefore, liberal adaptations are allowed (see section 5 for allowable adaptations). What this means, in the final analysis, is that the scope of the test is to determine that the database of the test (once inserted) is accessible via the schema defined in the specification.

2.3 Type and Functions

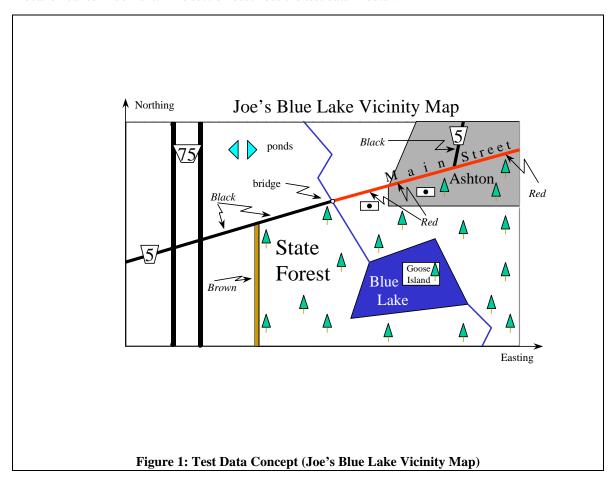
The Types and Functions portion of the specification defines the structure of feature tables (or views), a geometry column metadata view, and a spatial reference system information view. It also defines the relationships (primary key and foreign key) between these tables (views). It does **not** define the detail for the way that geometry is stored in geometry tables (or views), instead it defines a set of abstract data types that constitute the allowable set of values for the geometry-valued columns in feature tables. Furthermore, it defines a set of functions that operate on these types (including textual and binary exportation), the textual (Well-Known Text, or WKT) and a binary (Well-Known Binary, WKB) representation of geometry used (by constructor functions) to construct the types, and the textual representation of the spatial reference system to be stored in the spatial reference system information view.

The test suite for this alternative contains a series of CREATE TABLE statements that create a set of feature tables, a series of INSERT statements that populate these tables (using the geometry type constructor functions). It also includes a series of queries that exercise each function defined in the specification. Thus the scope of this test is to test that the constructor functions create the correct column (and to determine that the database of the test (once inserted) is accessible via the schema defined in the

specification) and that the functions defined in the specification operate properly on these columns. There are a number of adaptations expected (see section 5 for allowable adaptations)

3 Test Data

The data for all of the test alternatives is the same. It is a synthetic data set, developed, by hand, to exercise the functionality of the specification. It is a set of features that make up a map (see Figure 1) of a fictional location called Blue Lake. This section describes the test data in detail.



3.1 Test Data Semantics

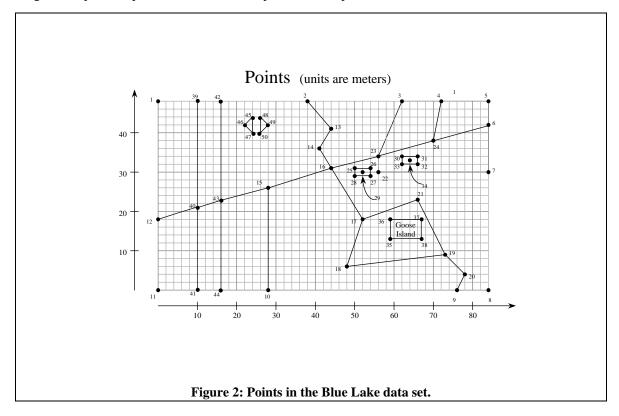
The semantics of this data set are as follows:

- A rectangle of the Earth is shown in UTM coordinates. Horizontal coordinates take meaning from POSC Horizontal Coordinate System #32214. Note 500,000 meters false Easting, and WGS74. Units are meters. (see http://www.petroconsultants.com/epsgweb/epsg.htm)
- Blue Lake (which has an island named Goose Island) is the prominent feature.
- There is a watercourse flowing from North to South. The portion from the top neatline to the lake is called Cam Stream. The portion from the lake to the bottom neatline has no name (Name value is "Null")
- There is an area place named Ashton.
- There is a State Forest whose administrative area includes the lake and a portion of Ashton. Roads form the boundary of the State Forest. The "Green Forest" is the State Forest minus the lake.

- Route 5 extends across the map. It is two lanes where shown in black. It is four lanes where shown in Red.
- There is a major divided highway, Route 75, shown in a double black line, one line for each part of the divided highway. These two lines are seen as a multi-line.
- There is a bridge (Cam Bridge) where the road goes over Cam Stream, a point feature.
- Main Street shares some pavement with Route 5, and is always four lanes wide.
- There are two buildings along Main Street; each can be seen either as a point or as a rectangle footprint.
- There is a one-lane road forming part of the boundary of the State Forest, shown in brown.
- The are two fish ponds, which are seen as a collective, not as individuals; that is, they are a multipolygon.

3.2 Test Data Points and Coordinates

Figure 2 depicts the points that are used to represent the map.



The following table gives these coordinates associated with each point.

Point	Easting	Northing	Point	Easting	Northing
1	0	48	26	52	31
2	38	48	27	52	29
3	62	48	28	50	29
4	72	48	29	52	30
5	84	48	30	62	34
6	84	42	31	66	34

7	84	30	32	66	32
8	84`	0	33	62	32
9	76	0	34	64	33
10	28	0	35	59	13
11	0	0	36	59	18
12	0	18	37	67	18
13	44	41	38	67	13
14	41	36	39	10	48
15	28	26	40	10	21
16	44	31	41	10	0
17	52	18	42	16	48
18	48	6	43	16	23
19	73	9	44	16	0
20	78	4	45	24	44
21	66	23	46	22	42
22	56	30	47	24	40
23	56	34	48	26	44
24	70	38	49	28	42
25	50	31	50	26	40

3.3 Test Data Schema and Values

The database schema and values used are given in this section. Please note that in the test suite the schema is created and populated in different ways, depending on the alternative being tested. These schema and values are provided for the sole purpose of giving the reader of this document a more complete picture of the actual test database. The notation used is XML and the Document Type Definition (DTD) is given here also. The fact that the schema and values are provided in XML is no indication of OGC support for XML, it is just a convenient way to completely document the schema without SQL (which might confuse the issue of testing).

3.3.1 Document Type Definition for the Schema and Values

The following DTD is very simple. It says that XML documents of type "ogc-sfsql-table" have the following structure:

- The table consists of a table definition and one or more row values.
- A table definition consists of a name and one or more column definitions.
- A columns definition consists of a name, a type, and zero or more constraints.
- A row value consists of one or more column values.
- A column value consists of a name and a value.

Here is the DTD:

```
<?XML VERSION="1.0"?>
<!DOCTYPE ogc-sfsql-table[
    <!ELEMENT ogc-sfsql-table (table-definition, row-value+)>
    <!ELEMENT table-definition (name, column-definition+)>
    <!ELEMENT column-definition (name, type, constraint*)>
```

```
<!ELEMENT row-value (column-value+)>
    <!ELEMENT column-value (name, value)>
]>
```

3.3.2 Lakes

```
<ogc-sfsql-table>
   <table-definition>
      <name>lakes</name>
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>shore</name>
          <type>POLYGON</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>101</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>'Blue Lake'</value>
      </column-value>
      <column-value>
          <name>shore</name>
          <value>'POLYGON((52 18,66 23,73 9,48 6,52 18), \
                 (59 18,67 18,67 13,59 13,59 18))'</value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.3 Road Segments

```
<ogc-sfsql-table>
   <table-definition>
      <name>road_segments/name>
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>aliases</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>num_lanes</name>
          <type>INTEGER</type>
      </column-definition>
      <column-definition>
          <name>centerline</name>
          <type>LINESTRING</type>
```

```
</column-definition>
</table-definition>
<row-value>
   <column-value>
      <name>fid</name>
      <value>102</value>
   </column-value>
   <column-value>
      <name>name</name>
      <value>'Route 5'</value>
   </column-value>
   <column-value>
      <name>centerline</name>
      <value>'LINESTRING( 0 18, 10 21, 16 23, 28 26, 44 31 )'
   </column-value>
</re>
<row-value>
   <column-value>
      <name>fid</name>
      <value>103</value>
   </column-value>
   <column-value>
      <name>name</name>
      <value>'Route 5'</value>
   </column-value>
   <column-value>
      <name>centerline</name>
      <value>'LINESTRING( 44 31, 56 34, 70 38 )'</value>
   </column-value>
</row-value>
<row-value>
   <column-value>
      <name>fid</name>
      <value>104</value>
   </column-value>
   <column-value>
      <name>name</name>
      <value>'Route 5'</value>
   </column-value>
   <column-value>
      <name>centerline</name>
       <value>'LINESTRING( 70 38, 72 48 )'</value>
   </column-value>
</row-value>
<row-value>
   <column-value>
      <name>fid</name>
      <value>105</value>
   </column-value>
   <column-value>
      <name>name</name>
       <value>'Main Street'</value>
   </column-value>
   <column-value>
      <name>centerline</name>
      <value>'LINESTRING( 70 38, 84 42 )'</value>
   </column-value>
</re>
<row-value>
   <column-value>
      <name>fid</name>
       <value>106</value>
   </column-value>
   <column-value>
      <name>name</name>
      <value>'Dirt Road by Green Forest'</value>
   </column-value>
```

3.3.4 Divided Routes

```
<ogc-sfsql-table>
   <table-definition>
      <name>divided_routes
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>num_lanes</name>
          <type>INTEGER</type>
      </column-definition>
      <column-definition>
          <name>centerlines</name>
          <type>MULTILINESTRING</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>119</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>'Route 75'</value>
      </column-value>
      <column-value>
          <name>num_lanes</name>
          <value>4</value>
      </column-value>
      <column-value>
          <name>centerlines</name>
          <value>'MULTILINESTRING((10 48,10 21,10 0), \
                 (16 0,16 23,16 48))'</value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.5 Forests

```
<column-definition>
          <name>boundary</name>
          <type>MULTIPOLYGON</type>
       </column-definition>
   </table-definition>
   <row-value>
       <column-value>
          <name>fid</name>
          <value>109</value>
       </column-value>
       <column-value>
          <name>name</name>
          <value>'Green Forest'</value>
       </column-value>
       <column-value>
          <name>boundary</name>
          <value>'MULTIPOLYGON(((28 26,28 0,84 0,84 42,28 26), \
                 (52\ 18,66\ 23,73\ 9,48\ 6,52\ 18)), \setminus
                 ((59 18,67 18,67 13,59 13,59 18)))'</value>
       </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.6 Bridges

```
<ogc-sfsql-table>
   <table-definition>
      <name>bridges</name>
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>position</name>
          <type>POINT</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>110</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>'Cam Bridge'</value>
      </column-value>
      <column-value>
          <name>position</name>
          <value>'POINT( 44 31 )'</value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.7 Streams

```
<constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>centerline</name>
          <type>LINESTRING</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>111</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>'Cam Stream'</value>
      </column-value>
      <column-value>
          <name>centerline</name>
          <value>'LINESTRING( 38 48, 44 41, 41 36, 44 31, 52 18 )'</value>
      </column-value>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>112</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>NULL</value>
      </column-value>
      <column-value>
          <name>centerline</name>
          <value>'LINESTRING( 76 0, 78 4, 73 9 )'</value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.8 Buildings

```
<ogc-sfsql-table>
   <table-definition>
      <name>buildings</name>
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>address</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>position</name>
          <type>POINT</type>
      </column-definition>
      <column-definition>
          <name>footprint</name>
          <type>POLYGON</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
```

```
<name>fid</name>
          <value>113</value>
      </column-value>
      <column-value>
          <name>address</name>
          <value>'123 Main Street'</value>
      </column-value>
      <column-value>
          <name>position</name>
          <value>'POINT( 52 30 )'</value>
      </column-value>
      <column-value>
          <name>footprint</name>
          <value>'POLYGON( ( 50 31, 54 31, 54 29, 50 29, 50 31) )'</value>
      </column-value>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>114</value>
      </column-value>
      <column-value>
          <name>address</name>
          <value>'215 Main Street'</value>
      </column-value>
      <column-value>
          <name>position</name>
          <value>'POINT( 64 33 )'</value>
      </column-value>
      <column-value>
          <name>footprint</name>
          <value>'POLYGON( ( 66 34, 62 34, 62 32, 66 32, 66 34) )'</value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.9 Ponds

```
<ogc-sfsql-table>
   <table-definition>
      <name>ponds</name>
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>type</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>shores</name>
          <type>MULTIPOLYGON</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>120</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>NULL</value>
```

3.3.10 Named Places

```
<oqc-sfsql-table>
   <table-definition>
      <name>named_places
      <column-definition>
          <name>fid</name>
          <type>INTEGER</type>
          <constraint>NOT NULL
          <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>name</name>
          <type>VARCHAR(64)</type>
      </column-definition>
      <column-definition>
          <name>boundary</name>
          <type>POLYGON</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>117</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>'Ashton'</value>
      </column-value>
      <column-value>
          <name>boundary</name>
          <value>'POLYGON( ( 62 48, 84 48, 84 30, 56 30, 56 34, \
                62 48) )'</value>
      </column-value>
   </row-value>
   <row-value>
      <column-value>
          <name>fid</name>
          <value>118</value>
      </column-value>
      <column-value>
          <name>name</name>
          <value>'Goose Island'</value>
      </column-value>
      <column-value>
          <name>boundary</name>
          <value>'POLYGON( ( 67 13, 67 18, 59 18, 59 13, 67 13) )'/value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

3.3.11 Map Neatline

```
<ogc-sfsql-table>
```

```
<table-definition>
      <name>map_neatlines</name>
      <column-definition>
         <name>fid</name>
         <type>INTEGER</type>
         <constraint>NOT NULL
         <constraint>PRIMARY KEY</constraint>
      </column-definition>
      <column-definition>
          <name>neatline</name>
          <type>POLYGON</type>
      </column-definition>
   </table-definition>
   <row-value>
      <column-value>
         <name>fid</name>
          <value>115</value>
      </column-value>
      <column-value>
          <name>neatline</name>
          <value>'POLYGON( ( 0 0, 0 48, 84 48, 84 0, 0 0 ) )'</value>
      </column-value>
   </row-value>
</ogc-sfsql-table>
```

4 Conformance Items

This section details the queries that are to be executed in the three tests. Each query (or each row of the tables in this section) constitutes a Conformance Item in the terminology of the Conformance Testing Program document. The ID in the following tables is used to reference the specific Conformance Item.

4.1 Normalized Geometry

As stated in the description for this alternative, the scope of this test is determine that the database of the test (once inserted) is accessible via the schema defined in the specification. The following queries accomplish this test.

ID	Functionality Tested	Query Description	Answer
N1	GEOMETRY_COLUMNS	For this test we will check to see	lakes, road_segments,
	table/view is created/updated	that all of the feature tables are	divided_routes, buildings,
	properly	represented by entries in the	buildings, forests, bridges,
		GEOMETRY_COLUMNS	named_places, streams, ponds,
		table/view	map_neatlines
N2	GEOMETRY_COLUMNS	For this test we will check to see	lake_geom, road_segment_geom,
	table/view is created/updated	that all of the geometry tables are	divided_route_geom,
	properly	represented by entries in the	forest_geom, bridge_geom,
		GEOMETRY_COLUMNS	stream_geom, building_pt_geom,
		table/view	building_area_geom, pond_geom,
			named_place_geom,
			map_neatline_geom
N3	GEOMETRY_COLUMNS	For this test we will check to see	0
	table/view is created/updated	that the correct storage type for	
	properly	the streams table is represented in	
		the GEOMETRY_COLUMNS	
		table/view	
N4	GEOMETRY_COLUMNS	For this test we will check to see	3 (corresponds to 'LINESTRING')
	table/view is created/updated	that the correct geometry type for	
	properly	the streams table is represented in	
		the GEOMETRY_COLUMNS	
		table/view	
N5	GEOMETRY_COLUMNS	For this test we will check to see	2
	table/view is created/updated	that the correct coordinate	

	1		T
	properly	dimension for the streams table is represented in the GEOMETRY_COLUMNS table/view	
N6	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that the correct value of max_ppr for the streams table is represented in the GEOMETRY_COLUMNS table/view	3
N7	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that the correct value of srid for the streams table is represented in the GEOMETRY_COLUMNS table/view	101
N8	SPATIAL_REF_SYS table/view is created/updated properly	For this test we will check to see that the correct value of srtext is represented in the SPATIAL_REF_SYS table/view	PROJCS["UTM_ZONE_14N", GEOGCS["World Geodetic System 72", DATUM["WGS_72", SPHEROID["NWL_10D", 6378135, 298.26]], PRIMEM["Greenwich", 0], UNIT["Meter", 1.0]], PROJECTION["Transverse_Mercator"], PARAMETER["False_Easting", 500000.0], PARAMETER["False_Northing", 0.0], PARAMETER["Central_Meridian", - 99.0], PARAMETER["Scale_Factor", 0.9996], PARAMETER["Latitude_of_origin", 0.0], UNIT["Meter", 1.0]]'

4.2 Binary Geometry

As stated in the description for this alternative, the scope of this test is determine that the database of the test (once inserted) is accessible via the schema defined in the specification. The following queries accomplish this test.

ID	Functionality Tested	Query Description	Answer
B1	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that all of the feature tables are represented by entries in the GEOMETRY_COLUMNS table/view	lakes, road_segments, divided_routes, buildings, buildings, forests, bridges, named_places, streams, ponds, map_neatlines
B2	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that all of the geometry tables are represented by entries in the GEOMETRY_COLUMNS table/view	lake_geom, road_segment_geom, divided_route_geom, forest_geom, bridge_geom, stream_geom, building_pt_geom, building_area_geom, pond_geom, named_place_geom, map_neatline_geom
В3	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that the correct storage type for the streams table is represented in the GEOMETRY_COLUMNS table/view	1
B4	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that the correct geometry type for the streams table is represented in the GEOMETRY_COLUMNS table/view	3 (corresponds to 'LINESTRING')
В5	GEOMETRY_COLUMNS table/view is created/updated properly	For this test we will check to see that the correct coordinate dimension for the streams table is	2

В6	GEOMETRY_COLUMNS table/view is created/updated properly	represented in the GEOMETRY_COLUMNS table/view For this test we will check to see that the correct value of srid for the streams table is represented in the GEOMETRY_COLUMNS table/view	101
В7	SPATIAL_REF_SYS table/view is created/updated properly	For this test we will check to see that the correct value of srtext is represented in the SPATIAL_REF_SYS table/view	PROJCS["UTM_ZONE_14N", GEOGCS["World Geodetic System 72", DATUM["WGS_72", SPHEROID["NWL_10D", 6378135, 298.26]], PRIMEM["Greenwich", 0], UNIT["Meter", 1.0]], PROJECTION["Transverse_Mercator"], PARAMETER["False_Easting", 500000.0], PARAMETER["False_Northing", 0.0], PARAMETER["Central_Meridian", - 99.0], PARAMETER["Scale_Factor", 0.9996], PARAMETER["Latitude_of_origin", 0.0], UNIT["Meter", 1.0]]

4.3 Types and Functions

As stated in the description for this alternative, the scope of this test is determine that:

- 1. the database of the test (once inserted) is accessible via the schema defined in the specification and
- 2. that the functionality defined in the specification is implemented as described.

The following queries accomplish the first part of this test.

ID	Functionality Tested	Query Description	Answer
T1	GEOMETRY_COLUMNS	For this test we will check to see	lakes, road_segments,
	table/view is created/updated	that all of the feature tables are	divided_routes, buildings, forests,
	properly	represented by entries in the	bridges, named_places, streams,
		GEOMETRY_COLUMNS	ponds, map_neatlines
		table/view	
T2	GEOMETRY_COLUMNS	For this test we will check to see	Centerline
	table/view is created/updated	that the correct geometry column	
	properly	for the streams table is represented	
		in the GEOMETRY_COLUMNS	
		table/view	
Т3	GEOMETRY_COLUMNS	For this test we will check to see	2
	table/view is created/updated	that the correct coordinate	
	properly	dimension for the streams table is	
		represented in the	
		GEOMETRY_COLUMNS	
		table/view	
T4	GEOMETRY_COLUMNS	For this test we will check to see	101
	table/view is created/updated	that the correct value of srid for	
	properly	the streams table is represented in	
		the GEOMETRY_COLUMNS	
		table/view	
T5	SPATIAL_REF_SYS table/view	For this test we will check to see	'PROJCS["UTM_ZONE_14N",
	is created/updated properly	that the correct value of srtext is	GEOGCS["World Geodetic System
		represented in the	72", DATUM["WGS_72", SPHEROID["NWL_10D", 6378135,
		SPATIAL_REF_SYS table/view	298.26]], PRIMEM["Greenwich", 0],
			UNIT["Meter", 1.0]],
			PROJECTION["Transverse_Mercator"
], PARAMETER["False_Easting",

	500000.0],
	PARAMETER["False_Northing", 0.0],
	PARAMETER["Central_Meridian", -
	99.0], PARAMETER["Scale_Factor",
	0.9996],
	PARAMETER["Latitude_of_origin",
	0.0], UNIT["Meter", 1.0]]'

The second part of this test will check the functions described in the specification. The section of the specification that each set of queries will test organizes the remainder of this section.

4.3.1 Queries for Section **3.2.10.2**

ID	Functionality Tested	Query Description	Answer
Т6	Dimension(g Geometry) : Integer	For this test we will determine the dimension of Blue Lake.	2
T7	GeometryType(g Geometry): String	For this test we will determine the type of Route 75.	9 (corresponds to 'MULTILINESTRING')
Т8	AsText(g Geometry) : String	For this test we will determine the WKT representation of Goose Island.	'POLYGON((67 13, 67 18, 59 18, 59 13, 67 13))'
Т9	AsBinary(g Geometry) : Blob	For this test we will determine the WKB representation of Goose Island. We will test by applying AsText to the result of PolygonFromText to the result of AsBinary.	'POLYGON((67 13, 67 18, 59 18, 59 13, 67 13))'
T10	SRID(g Geometry) : Integer	For this test we will determine the SRID of Goose Island.	101
T11	IsEmpty(g Geometry) : Integer	For this test we will determine whether the geometry of a segment of Route 5 is empty.	0 (or whatever indicates the boolean value false)
T12	IsSimple(g Geometry) : Integer	For this test we will determine whether the geometry of a segment of Blue Lake is simple.	1 (or whatever indicates the boolean value true)
T13	Boundary(g Geometry) : Geometry	For this test we will determine the boundary of Goose Island. NOTE: The boundary result is as defined in 3.12.3.2 of 96-015R1.	'LINESTRING(67 13, 67 18, 59 18, 59 13, 67 13)'
T14	Envelope(g Geometry) : Integer	For this test we will determine the envelope of Goose Island.	'POLYGON((59 13, 59 18, 67 18, 67 13, 59 13))'

4.3.2 Queries for Section **3.2.11.2**

ID	Functionality Tested	Query Description	Answer
T15	X(p Point) : Double Precision	For this test we will determine the	44.00
		X coordinate of Cam Bridge.	
T16	Y(p Point) : Double Precision	For this test we will determine the	31.00
		Y coordinate of Cam Bridge.	

4.3.3 Queries for Section 3.2.12.2

ID	Functionality Tested	Query Description	Answer
T17	StartPoint(c Curve) : Point	For this test we will determine the	'POINT(0 18)'
		start point of road segment 102.	
T18	EndPoint(c Curve) : Point	For this test we will determine the	'POINT(44 31)'
		end point of road segment 102.	
T19	IsClosed(c Curve) : Integer	For this test we will determine the	1 (or whatever indicates the
		boundary of Goose Island.	boolean value true)
T20	IsRing(c Curve) : Integer	For this test we will determine the	1 (or whatever indicates the
		boundary of Goose Island.	boolean value true)
T21	Length(c Curve) : Double	For this test we will determine the	26.00 (meters)

Precision	length of road segment 106.	

4.3.4 Queries for Section **3.2.13.2**

ID	Functionality Tested	Query Description	Answer
T22	NumPoints(l LineString) : Integer	For this test we will determine the	5
		number of points in road segment	
		102.	
T23	PointN(l LineString, n Integer):	For this test we will determine the	'POINT(0 18)'
	Point	1st point in road segment 102.	

4.3.5 Queries for Section **3.2.14.2**

ID	Functionality Tested	Query Description	Answer
T24	Centroid(s Surface) : Point	For this test we will determine the	'POINT(53 15.5)'
		centroid of Goose Island.	
T25	PointOnSurface(s Surface) : Point	For this test we will determine a	1 (or whatever indicates the
		point on Goose Island. NOTE: For	boolean value true)
		this test we will have to uses the	
		Contains function (which we don't	
		test until later).	
T26	Area(s Surface) : Double	For this test we will determine the	40.00 (square meters)
	Precision	area of Goose Island.	

4.3.6 Queries for Section **3.2.15.2**

ID	Functionality Tested	Query Description	Answer
T27	ExteriorRing(p Polygon):	For this test we will determine the	'LINESTRING(52 18, 66 23, 73
	LineString	exteroir ring of Blue Lake.	9, 48 6, 52 18)'
T28	NumInteriorRings(p Polygon):	For this test we will determine the	1
	Integer	number of interior rings of Blue	
		Lake.	
T29	InteriorRingN(p Polygon, n	For this test we will determine the	'LINESTRING(59 18, 67 18, 67
	Integer): LineString	first interior ring of Blue Lake.	13, 59 13, 59 18)'

4.3.7 Queries for Section **3.2.16.2**

ID	Functionality Tested	Query Description	Answer
T30	NumGeometries(g	For this test we will determine the	2
	GeometryCollection): Integer	number of geometries in Route 75.	
T31	GeometryN(g	For this test we will determine the	'LINESTRING(16 0, 16 23, 16 48
	GeometryCollection, n Integer):	second geometry in Route 75.)'
	Geometry		

4.3.8 Queries for Section **3.2.17.2**

ID	Functionality Tested	Query Description	Answer
T32	IsClosed(mc MultiCurve) : Integer	For this test we will determine if	0 (or whatever indicates the
		the geometry of Route 75 is	boolean value false)
		closed.	
T33	Length(mc MultiCurve) : Double	For this test we will determine the	96.00 (meters)
	Precision	length of Route 75. NOTE: This	
		makes no semantic sense in our	
		example	

4.3.9 Queries for Section 3.2.18.2

ID	Functionality Tested	Query Description	Answer
T34	Centroid(ms MultiSurface): Point	For this test we will determine the	'POINT(25 42)'
		centroid of the ponds.	
T35	PointOnSurface(ms MultiSurface)	For this test we will determine a	1 (or whatever indicates the
	: Point	point on the ponds. NOTE: For	boolean value true)

		this test we will have to uses the Contains function (which we don't test until later).	
T36	Area(ms MultiSurface) : Double	For this test we will determine the	8.00 (square meters)
	Precision	area of the ponds.	

4.3.10 Queries for Section 3.2.19.2

ID	Functionality Tested	Query Description	Answer
T37	Equals(g1 Geometry, g2	For this test we will determine if	1 (or whatever indicates the
	Geometry): Integer	the geometry of Goose Island is equal to the same geometry as	boolean value true)
		constructed from it's WKT	
		representation.	
T38	Disjoint(g1 Geometry, g2	For this test we will determine if	1 (or whatever indicates the
	Geometry): Integer	the geometry of Route 75 is disjoint from the geometry of	boolean value true)
		Ashton.	
T39	Touch(g1 Geometry, g2	For this test we will determine if	1 (or whatever indicates the
	Geometry): Integer	the geometry of Cam Stream	boolean value true)
		touches the geometry of Blue Lake.	
T40	Within(g1 Geometry, g2	For this test we will determine if	1 (or whatever indicates the
	Geometry): Integer	the geometry of the house at 215	boolean value true)
		Main Street is within Ashton.	
T41	Overlap(g1 Geometry, g2	For this test we will determine if	1 (or whatever indicates the
	Geometry): Integer	the geometry of Green Forest overlaps the geometry of Ashton.	boolean value true)
T42	Cross(g1 Geometry, g2 Geometry)	For this test we will determine if	1 (or whatever indicates the
	: Integer	the geometry of road segment 101	boolean value true)
TF 42	1.4	crosses the geometry of Route 75. For this test we will determine if	1/ 1 / 1
T43	Intersects(g1 Geometry, g2 Geometry): Integer	the geometry of road segment 101	1 (or whatever indicates the boolean value true)
	Geometry): Integer	intersects the geometry of Route	boolean value true)
		<i>7</i> 5.	
T44	Contains(g1 Geometry, g2	For this test we will determine if	0 (or whatever indicates the
	Geometry): Integer	the geometry of Green Forest	boolean value fase)
T45	Relate(g1 Geometry, g2	contains the geometry of Ashton. For this test we will determine if	1 (or whatever indicates the
143	Geometry, PatternMatrix String):	the geometry of Green Forest	boolean value true)
	Integer	relates to the geometry of Ashton	
		using the pattern "TTTTTTT".	

4.3.11 Queries for Section 3.2.20.2

ID	Functionality Tested	Query Description	Answer
T46	Distance(g1 Geometry, g2	For this test we will determine the	12 (meters)
	Geometry): Double Precision	distance between Cam Bridge and	
		Ashton.	

4.3.12 Queries for Section 3.2.21.2

ID	Functionality Tested	Query Description	Answer
T47	Intersection(g1 Geometry, g2	For this test we will determine the	'POINT(52 18)'
	Geometry): Geometry	intersection between Cam Stream	
		and Blue Lake.	
T48	Difference(g1 Geometry, g2	For this test we will determine the	'POLYGON((56 34, 62 48, 84
	Geometry): Geometry	difference between Ashton and	48, 84 42, 56 34))' NOTE: The
		Green Forest.	order of the vertices here is
			arbitrary.
T49	Union(g1 Geometry, g2	For this test we will determine the	'POLYGON((52 18,66 23,73 9,48

	Geometry): Integer	union of Blue Lake and Goose	6,52 18))' NOTE: The outer ring
		Island	of BLue Lake is the answer.
T50	SymmetricDifference(g1	For this test we will determine the	'POLYGON((52 18,66 23,73 9,48
	Geometry, g2 Geometry): Integer	symmetric difference of Blue	6,52 18))' NOTE: The outer ring
		Lake and Goose Island	of BLue Lake is the answer.
T51	Buffer(g Geometry, d Double	For this test we will make a 15	1 (or whatever indicates the
	Precision): Geometry	meter buffer about Cam Bridge.	boolean value true)
		NOTE: This test we count the	
		number of buildings contained in	
		the buffer that is generated. This	
		test only works because we have a	
		single bridge record, two building	
		records, and we selected the buffer	
		size such that only one of the	
		buildings is contained in the	
		buffer.	
T52	ConvexHull(g Geometry):	For this test we will determine the	'POLYGON((52 18,66 23,73 9,48
	Geometry	convex hull of Blue Lake	6,52 18))' NOTE: The outer ring
			of Blue Lake is the answer.

5 Adaptations

This section details the adaptations that are allowable to the test suite code and documentation guidelines that relate to these adaptations. The Conformance Testing Program document states:

"It is likely that organizations will have to adapt test suite software to work with their products. OGC will provide guidance regarding which sections of the test can be adapted and which cannot. In any case, OGC Testing Staff, in consultation with and under the approval authority of the Conformance Testing Coordinator, must then review the adaptation to ensure that the original intent and functionality of the test suite software is maintained. OGC Testing Staff must also document the adaptation in the Conformance Summary Report (See Appendix B). Adaptations must be submitted by organizations that desire to undergo testing at least three (3) weeks before the scheduled test. Failure to do so will result in a test cancellation."

This section constitutes the guidance in the above statement.

5.1 General Adaptation Documentation Guidance

Adaptations to the test suite code must be documented using a standard comment block, liberal text that describes the nature of the adaptation, the original test suite code, and the adapted test suite code. For SQL scripts, the documentation must follow the form of this example:

```
-- !#@ ADAPTATION END ------
```

For Embedded C, the documentation must follow the form of this example:

5.2 General Adaptation #1: Tables and Views

There are two general adaptations that are allowable in all three alternatives. The first general allowable adaptation relates to whether the tables set forth in the specification (the SPATIAL_REF_SYS table/view, the GEOMETRY_COLUMNS table/view, feature tables/views, and geometry tables/views) are implemented as tables or views. Either is allowable. The test suites create these as tables (explicitly with CREATE TABLE statements), but implementations may adapt the test suite to create them (as tables or as views) in any way necessary. See the following sections for further details.

5.2.1 Documentation guidance

Any adaptation made in this area must be documented as if for a user of the software (this requirement should be satisfied by inclusion of the portion of the software documentation that addresses this area of the implementation). For example, if an implementation requires that feature tables and geometry tables be created using a C programming API, then that API documentation must be provided. Another example might be that a series of SQL statements (modeled after a particular template) would need to be developed and executed to ensure proper management of the geometry columns metadata table/view, if this is the case, then documentation that describes this should be included.

5.2.2 OGC Intentions Regarding This Adaptation

Please note that it IS NOT OGC's intention to change this situation, because it would severely limit implementation freedoms. The policy embodied in this statement is that OGC wants to encourage a broad range of implementations and to guide these implementation towards greater levels of interoperability over time.

5.3 General Adaptation #2: Naming of Tables/Views, Types, and Functions

The second general adaptation relates to the names of tables/views, types, and functions in the specification. It is allowable for implementations to use different names in the adapted test suite. There are good reasons for this including, but not limited to:

Limitations of some commercial database systems in the length of names (and even though it was the
intention of the authors to meet an 18 character length maximum, this was not uniformly applied in
this revision).

- Database systems that allow dynamically loadable extensions often require that the names be different so that they do not cause problems with name resolution algorithms within their infrastructure.
- OGC is interested in aligning this specification with the work of the ISO/IEC JTC1/SC32/WG 4 (the committee working on Multimedia extensions to SQL3, especially the Part 3: Spatial). This interest will cause the names to change slightly anyway.
- The specification states, "Views that create the same logical structure are equally compliant." This statement has the implication that we must be lenient when it comes to naming.

5.3.1 Documentation Guidance

If any of these names (tables/views, types, or functions) are changed, the implementer must provide a translation table. This table has two columns. The first column must contain the name as set forth in the test suite, the second the name that the implementer wishes to use. For example, if an implementation calls a view that implements the GEOMETRY_COLUMNS metadata table MY_GEOM_COLS and the SPATIAL_REF_SYS table SRS_TABLE, and the MultiPolygonFromText function MPolyFromText, then they would submit a translation table as follows:

OGC Name	Implementation Name
GEOMETRY_COLUMNS	MY_GEOM_COLS
SPATIAL_REF_SYS	SRS_TABLE
MultiPolygonFromText	MPolyFromText

5.3.2 OGC Intentions Regarding This Adaptation

Please note that it **IS** OGC's intention to stabilize on the naming as soon as and wherever it is **practical** and to encode these stabilized names in a future revisions of the specification. In turn the Conformance Testing Guidelines will become considerably stricter along these lines. Again, the policy is to allow a broad range of implementations and to guide these implementations towards greater levels of interoperability over time.

5.4 Normalized Geometry

For details, see the test suite itself. Allowable adaptations are marked with the text "*** ADAPTATION ALERT ***".

5.5 Binary Geometry

For details, see the test suite itself. Allowable adaptations are marked with the text "*** ADAPTATION ALERT ***".

5.6 Types and Functions

For details, see the test suite itself. Allowable adaptations are marked with the text "*** ADAPTATION ALERT ***".