

ISO/IEC JTC 1/SC 32 N 0295

Date: 1999-05-21

REPLACES: --

<p style="text-align: center;">ISO/IEC JTC 1/SC 32</p> <p style="text-align: center;">Data Management and Interchange</p> <p style="text-align: center;">Secretariat: United States of America (ANSI) Administered by Pacific Northwest National Laboratory on behalf of ANSI</p>

DOCUMENT TYPE	Other document (Open)
TITLE	SQL MM Full-Text, Spatial, and Still Image Presentation
SOURCE	Peter Pistor SC 32/WG 4
PROJECT NUMBER	
STATUS	Presentation for WG 4 at the Group Tutorial ISO/IEC JTC 1/SC 32 on 1999-05-21
REFERENCES	
ACTION ID.	FYI
REQUESTED ACTION	
DUE DATE	
Number of Pages	11
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32

Pacific Northwest National Laboratory *, 901 D Street, SW., Suite 900, Washington, DC, 20024-2115, United States of America

Telephone: +1 703 575 2114; Facsimile: +1 703 681 9180; E-mail: MannD@battelle.org

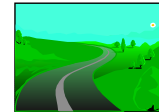
*Pacific Northwest National Laboratory (PNL) administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

SQL/MM Full-Text, Spatial and Still-Image

Peter Pistor (Germany)
Stefan Dessoch(USA)

Presented to SC32 Working Group Tutorial Meeting
Matsue, Japan, 1999-05-21

revised 1999-06-01



▼ SQL/MM Overview

- Multipart standard:
- SQL/MM Framework (Part 1)
 - Overview and conformance
- SQL/MM Full-Text (Part 2)
 - Information about construction of text and search patterns and for searching of text
- SQL/MM Spatial (Part 3)
 - Information about storing, managing, and retrieving information related to spatial data such as geometry and topography
- SQL/MM Still-Image (Part 5)
 - Information about searching large collections of still images

▼ SQL/MM Motivation

- Enabling functionality of SQL3:
 - ▶ definition of user-defined, application specific data types
 - ▶ implementation of user-defined functions to support application specific operations on the data types
 - ▶ storage of large objects (BLOBs and CLOBs)
 - ▶ powerful trigger and constraint mechanisms to maintain the integrity and semantics of the new data types
 - ▶ storage and execution of user-defined stored procedures in the server
- This enables ...
 - ▶ development of application specific collections of user-defined types, user-defined functions, triggers, constraints, and stored procedures (i.e. libraries) "tight" to the DBMS engine

▼ SQL/MM Full-Text

- Why Full-Text standard library?
 - ▶ Built-in search facilities (LIKE, SIMILAR) not powerful enough (text viewed as string of characters).
 - ▶ Need higher level notion of text
- Structural units in Full-Text:
 - ▶ Words
 - ▶ Sentences
 - ▶ Paragraphs
- Operations in Full-Text:
 - ▶ Boolean Search
 - ▶ Ranking
 - ▶ Conceptual Search

SQL/MM Full-Text: Boolean Search

■ Full-Text sample:

Every text value is associated with a specific language.

■ Full-Text items have language attribute

■ Boolean query facilities

- ▶ Single word search
- ▶ Phrase search
- ▶ Context based search
- ▶ Linguistic search
- ▶ Stopword processing
- ▶ Masking facilities
- ▶ Search pattern expansion, e.g.:
 - Sound expansion
 - Broader/narrower term expansion
 - Synonym expansion

SQL/MM Full-Text: Boolean search examples

■ Single word search:

```
SELECT * FROM myDocs
WHERE 1 = CONTAINS(TextBody, "specific")
```

Every text value is associated with a specific language.

■ Phrase search:

```
SELECT * FROM myDocs
WHERE 1 = CONTAINS(TextBody, "specific language")
```

Every text value is associated with a specific language.

■ Context search:

```
SELECT * FROM myDocs WHERE 1 = CONTAINS(TextBody,
"text" IN SAME SENTENCE AS "language")
```

Every text value is associated with a specific language.

■ Stopwords:

```
SELECT * FROM myDocs WHERE 1 = CONTAINS(TextBody,
"value was associated")
```

Every text value is associated with a specific language.

■ Linguistic search:

```
SELECT * FROM myDocs WHERE 1 = CONTAINS(TextBody,
'STEMMED FORM OF "values are associated")
```

Every text value is associated with a specific language.

▼ SQL/MM Full-Text

- Ranking

```
SELECT * FROM myDocs
```

```
WHERE 1.2 < RANK(TextBody, 'specific')
```

- ▶ Ranks according to implementation - defined criteria (e.g. frequency of "specific")

- Conceptual search

```
SELECT * FROM myDocs
```

```
WHERE 1 = CONTAINS(TextBody,  
'IS ABOUT "every text value is associated  
with a specific language"')
```

- ▶ Identifies Full-Text items which are pertinent to rhs of "IS ABOUT" operator

▼ SQL/MM Spatial: goals, motivation

- Goals

- ▶ Support for "flat world" (2-d) geometric objects and operations
- ▶ Coverage of important application areas
- ▶ Simple features

- Motivation

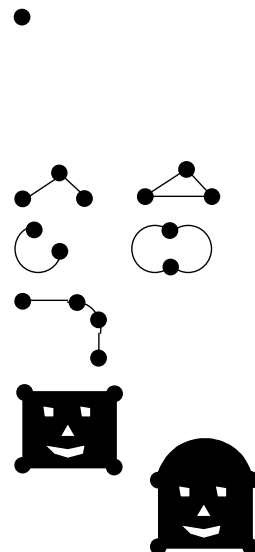
- ▶ Breake ground for **standard** type library
- ▶ Promote efficient access methods on relational platforms

▼ SQL/MM Spatial: Players

- JTC1 SC32 WG4: SQL/MM Spatial
- ISO TC211: Geomatics
- Open GIS Consortium:
 - OpenGIS Simple Feature Specification
 - SQL2 Bindings
 - CORBA Binding
 - OLE Binding
 - SQL3 Binding: SQL/MM Spatial
 - Guarantees implementations
 - Established verification procedures

▼ Spatial objects

- 0-dim. objects: points
- 1-dim. objects: (planar) curves;
sub- types differ w.r.t.
interpolation between points
 - ST_LineString: linear interpolation
 - ST_CircularString (opt): circular arcs
 - ST_CompoundString (opt): mixed
- 2-dim. objects: (planar)
surfaces
 - ST_Polygon: ST_LineString
boundaries
 - ST_CurvePolygon (opt):
ST_CompoundString boundaries



▼ Spatial objects (cont)

- Collection valued objects:

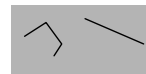
- ST_GeomCollection

- ▶ Reference system: same for all elements
 - ▶ Any geometry type admissible as element type
 - ▶ Subtypes of ST_GeomCollection with restrictions on element types

- ST_MultiPoint



- ST_MultiCurve*, ST_MultiLineString



- ST_MultiSurface*, ST_MultiPolygon

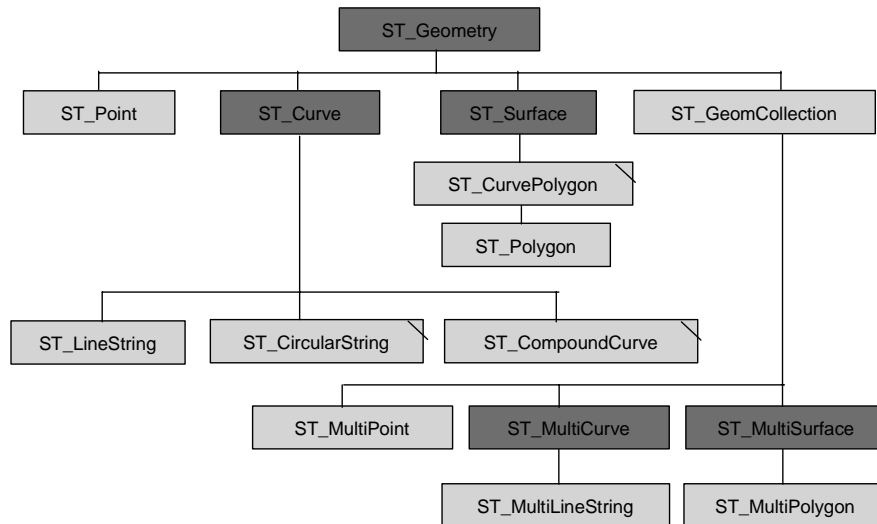


- (* denotes optionally instantiable types)

▼ SQL/MM Spatial: Operations

- Usual observers and mutators
- Transform routines
 - ▶ transform objects into binary or textual representations (and vice versa)
 - ▶ Enables implementation by 3GL functions using minimal SQL3 machinery
- Important topical operations, e.g.
 - ▶ Constructors (controlling wellformedness)
 - ▶ Distance
 - ▶ Tests (contains, overlaps, touches, crosses, ...)
 - ▶ Intersection, difference, union
 - ▶ Find referencing system
 - ▶ Length, area, perimeter

SQL/MM Spatial: Type Hierarchy



Spatial Reference System

- Controls aspects like units, prime meridian, coordinate system etc.
- Relies on reference systems defined by other authorities.
- Defined representation of reference system values
- One common spatial reference system value:
 - For elements of ST_Geometry values
 - Within column of type ST_Geometry

▼ SQL/MM Still-Image: Goals

- Enable screening of large imagebases
- Support for proven set of image features
- Type structure adaptable to evolving image processing technology
- Example: Find all possibly infringed logos by scoring them against a new logo.

```
SELECT * FROM RegLogos  
WHERE 1.2 < SI_findTexture(newLogo).SI_Score(Logo)
```

▼ SQL/MM Still-Image Objects

- SI_StillImage: raster images
- Abstract SI_Feature with subtypes
 - SI_AverageColor
 - SI_ColorHistogram
 - SI_PositionalColor: average colors of $n*m$ image segments
 - SI_Texture: coarseness, contrast, directionality
- SI_FeatureList: weighted list of SI_Feature items

▼ **SI_Still-Image: Operations**

- Constructor function
- Observer Methods for
 - Raw picture data
 - Image format (e.g. JPEG)
 - Pixel properties (bits per color, per pixel)
 - Size ..
 - Generation time, last update time
- Mutator for (raw) image content

▼ **SI_Feature, SI_FeatureList: Operations**

- All: scoring method (SI_Score)
 - Scores image w.r.t. a given feature
- All subtypes of SI_Feature
 - function extracting feature from images
- SI_AverageColor, SI_ColorHistogram
 - function for "manual" feature construction
- SI_FeatureList: feature/weight pairs lists
 - Constructor function for list header
 - Append method to extend feature list by another feature/weight pair

SQL/MM Still-Image: Final example

- Screen all logos in table RegLogos against a given logo (newLogo); use the texture and average colors of a standard grid of image segments ("positional color") for scoring; give these features of newLogo the weights 80% and 20%, resp.

```
SELECT * FROM RegLogos
WHERE 1.2 <
  SI_InitFeatureList
    (SI_findTexture(newLogo),0.8)
  .SI_Append
    (SI_findPositionalColor(newLogo), 0.2)
  .SI_Score(Logo)
```