SQL multimedia and application packages
ISO/IEC 13249 SQL/MM

Marek Rychlý

Brno University of Technology
Fakulty Information Technology
Department of Information Systems

Socrates/Erasmus Lecture
3rd April 2014
Outline

ISO/IEC 13249 SQL/MM Standard
  From SQL to SQL/MM
  SQL/MM Standard
  Documentation

SQL/MM Part 5: Still Image
  Data-types for Still Images
  SQL Statements for Still Images
  Implementations of Still Images DB Management Systems

SQL/MM Part 3: Spatial
  Data-types for Spatial Data
  SQL Statements for Spatial Data
  Implementations of Spatial DB Management Systems
Specialized Relational Databases

- Relational databases enable applications to store their data of particular data-types (data stored as members of defined relations) and to query, to process, and to modify the data. (by application of operations of the relational algebra)

- Many applications require special data-types/operations for vector and bitmap images, video and audio records, entities in geometric and geographical spaces, large text documents with full-text querying, storage of large data-sets and their data mining, etc.

- Modern database systems have to reflect these needs.

- There exists specialized databases by different vendors. (supporting the special data-types and/or operations; e.g. PostGIS)
Why do we need SQL/MM?

- Structured Full-text Query Language (SFQL) in 1991. (proposed by a small group of text search engine vendors in IEEE)
- The CONTAINS keyword proposed in SFQL. (meaning: a given text CONTAINS a particular word or phrase)
- But CONTAINS well-established in spatial databases. (meaning: a given spatial entity CONTAINS another spatial entity)
- It has been necessary to standardize SQL extensions. (the conclusion of a summit meeting in Tokyo in 1992)
- Utilization of object-oriented aspects of SQL:1999. (the concept of structured user-defined types and their methods)
- The extensions standardized in the SQL/MM standard. (developed by ISO subcommittee JTC1/SC32)
ISO/IEC 13249 SQL/MM Standard

- SQL/MM is a multi-part standard.
  (like SQL, but the various parts of SQL/MM are quite independent)

- Part 1 specifies common mechanisms, conventions used, and implementation requirements for the following parts.

- Parts 2–8 describe individual extensions of SQL for particular special types of data/operations.
  (standards ISO/IEC 13249-2 to ISO/IEC 13249-8)

- No implementation details, just requirements specification.
  (the implementation by particular database management systems)
Parts of the SQL/MM Standard

ISO/IEC 13249-...

2 Full-Text
(extended text search capabilities, beyond the native SQL operators)

3 Spatial
(how to store, retrieve, convert, compare, and process spatial data)

5 Still image
(to manage still, i.e. not moving, visual images such as photos)

6 Data mining
(an application package for the discovery of previously unknown, but important, information in large quantities of data, for the data mining)

draft 7 History
(to track and store modification history of data/rows of persistent tables)

draft 8 Metadata registries (MDR)
(for systematic management of metadata describing data)
SQL/MM Standard Documentation

  (document 32N1143, 3rd edition from 2004)
  (document 32N0640, 2nd edition from 2002)
  (document 32N1820, 4th edition from 2009)
- ISO/IEC TS 13249-7:2013 – Part 7: History
- ISO/IEC CD 13249-8:2011 – Part 8: Metadata Registry Access
  (document 32N2074, Committee Draft from 2011)
Still Image Data-types

- image data in structured type called *SI_StillImage* (collections of pixels representing 2-dimensional images; provides information about each image, e.g., its format, dimensions, etc.)

- methods can be applied to *SI_StillImage* to modify them (e.g., to scale, crop, rotate, to create “a thumbnail”, etc.)

- another data types describe various features of images
  - *SI_AverageColor* the “average” color of a given image
  - *SI_ColorHistogram* the occurrence of each color is found
  - *SI_PositionalColor* the location of specific colors
  - *SI_Texture* coarseness, contrast, direction of granularity

- the features can be combined into *SI_FeatureList*
  (and the result can be used to query images with similar features)
CREATE TABLE products (  
id integer primary key,  
photo_si SI_StillImage,  
photo_ac SI_AverageColor,  
photo_ch SI_ColorHistogram,  
photo_pc SI_PositionalColor,  
photo_tx SI_Texture);

Image features should be stored in the table alongside the image for indexing and utilization during querying similar images by features.
To Insert and Modify Data with Still Images

```sql
INSERT INTO products (id, photo_si) VALUES (1, SI_loadImage('/tmp/test_1.jpg'));

UPDATE products SET
    photo_si = SI_rotate(photo_si, 45)
WHERE id = 1;

UPDATE products SET
    photo_ac = SI_AverageColor(photo_si),
    photo_ch = SI_ColorHistogram(photo_si),
    photo_pc = SI_PositionalColor(photo_si),
    photo_tx = SI_Texture(photo_si)
WHERE id = 1;
```

The last `UPDATE` generates features descriptions for the image.
To Search a Still Image with Similar Features

```sql
SELECT src.id as source, dst.id as destination,
       SI_ScoreByFtrList(
           new SI_FeatureList(
               src.photo_ac, 0.3, src.photo_ch, 0.3,
               src.photo_pc, 0.1, src.photo_tx, 0.3
           ),
           dst.photo_si
       ) as similarity
FROM products src, products dst
WHERE src.id <> dst.id AND src.id = 1
ORDER BY similarity ASC;

SI_FeatureList for weights/importance of individual features.
```
Implementations of SQL/MM Still Image

- IBM DB2 Image Extender
  (a part of IBM DB2 Universal Database)
- Oracle Multimedia
- PostgreSQL-IE

Specification and implementation requirements in
Spatial Data in SQL/MM

- SQL/MM Spatial currently supports spatial data in
  - 0-dimension (i.e., points)
  - 1-dimension (i.e., lines)
  - 2-dimensions (i.e., “flat” shapes)
  - 3-dimensional (i.e. volumetric shapes) (and higher dimensions in future revisions)

- Spatial data mgmt systems dealing with geographic entities. (governmental bodies and enterprises dealing with geographic data)

- Various spatial reference systems for coordinates, e.g. (for different description of entities on the surface of the Earth)

- Spatial data-types are subtypes of \textit{ST_Geometry}. (based on OpenGIS Consortium geometry class hierarchy)
(adopted from “Integration of spatial vector data in . . .”; shaded boxes are not-instantiable types)
ST_Geometry Types (1)

- **ST_Point** for points (0 dimension, i.e., a single location) (a location described by an X and a Y coordinate to identify the location)

- **ST_LineString**, **ST_CircularString**, **ST_CompoundCurve** for different types of a curve (1 dimension, a seq of locations)

(adopted from “Integration of spatial vector data in enterprise relational database environments”)
ST_Geometry Types (2)

- **ST_Polygon**, **ST_CurvePolygon** for polygons with various types of boundaries (2 dimension, a sequence of curves) (for a surface having holes in its interior, the first curve describes the exterior/outer boundary and further curves define the holes)

- **ST_Multi...** data-types for multi-sets of other entities

(adopted from “Integration of spatial vector data in enterprise relational database environments”)

ST_LineString  
ST_CircularString
How to Create a Table with Spatial Entities

CREATE TABLE rivers (  
    name VARCHAR(30) PRIMARY KEY,  
    water_amount DOUBLE PRECISION,  
    river_line ST_LineString,  
    flood_zones ST_MultiPolygon);

CREATE TABLE buildings (  
    customer_name VARCHAR(50) PRIMARY KEY,  
    street VARCHAR(50),  
    city VARCHAR(20),  
    zip VARCHAR(10),  
    ground_plot ST_Polygon);

ST_Polygon for both ST_Polygon & ST_CurvePolygon entities.
To Insert and Modify Spatial Entities

\[
\text{INSERT INTO rivers (name, river\_line) VALUES ('MyRiver', ST\_LineString('linestring(750,150,750,750,200,900)'), 0));}
\]

\[
\text{UPDATE rivers SET flood\_zones = river\_line.ST\_Buffer(2, 'KILOMETER');}
\]

\[
\text{ST\_LineString(text) creates a line-string from OGC Well-known text representation. ST\_Buffer represents all points with less than or equal distance from a given entity (i.e., its buffer).}
\]
To Query Spatial Entities

SELECT customer_name, street, city, zip
FROM buildings AS b, rivers AS r
WHERE
  b.ground_plot.ST_Within(r.flood_zones) = 1
  AND
  b.ground_plot.ST_Distance(r.river_line, 'KILOMETER') <= 5
  AND
  b.ground_plot.ST_Area('METER') <= 100
ORDER BY b.ground_plot.ST_Area() DESC;

\[ST\_Within\] returns 1 iff one entity is inside another one. 
\[ST\_Distance\] returns a minimal distance of two entities. 
\[ST\_Area\] returns an area of a surface.
To Index Spatial Columns

```
INSERT INTO ST_Geometry_Columns
  (table_catalog, table_schema, table_name, column_name, coord_dimension, srid, geometry_type)
SELECT '', 'myschema', 'rivers', 'river_line',
    river_line.ST_Dimension(),
    river_line.ST_SRID(),
    river_line.ST_GeometryType()
FROM rivers LIMIT 1;

CREATE INDEX river_line_idx
  ON myschema.rivers(river_line)
  INDEXTYPE IS ST_Spatial_Index;
```

View `ST_Geometry_Columns` for metadata on spatial columns.
Implementations of SQL/MM Spatial

- IBM DB2 Spatial and Geodetic Extenders
  (a part of IBM DB2 Universal Database)
- IBM Informix Spatial DataBlade
  (a plug-in for the IBM Informix Dynamic Server)
- MapInfo SpatialWare
- MySQL Spatial
- Oracle Spatial
- PostgreSQL & PostGIS

Specification and implementation requirements in
(document 32N1820, 4th edition from 2009)
Summary

- SQL/MM defines particular extensions of SQL:1999.
- It defines SQL structured types and methods for DBMS.
- Support of full-text, spatial, still image, data mining, history, and metadata registries.
- The SQL/MM standard is still evolving (new parts 7 and 8).
- Current implementations covers Still Image and Spatial.

For more details, see ISO/IEC 13249 SQL/MM.
Thank you for your attention!

Marek Rychlý
<rychly@fit.vutbr.cz>