ISO/IEC JTC 1/SC 21 **N 11160**

Date: 1997-11-30

ISO/IEC CD 13249-5:199x (E)

ISO/IEC JTC 1/SC 21/WG 3

Secretariat: Canada

Information Technology — Database Languages —

SQL Multimedia and Application Packages —

Part 5: Still Image

Contents Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 13249 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*.

This document is based on the content of ISO/IEC Committee Draft Database Language (SQL).

This is the first edition of ISO/IEC SQL/MM – Part 5: Still Image.

Introduction

The purpose of this International Standard is to define multimedia and application specific types and their associated routines using the user-defined features in ISO/IEC 9075.

SQL/MM is structured as a multi-part standard. At present it consists of the following parts:

Part 1: Framework

Part 2: Full-Text

Part 3: Spatial

Part 4: General Purpose Facilities

Part 5: Still Image

Information Technology — Database Languages — SQL Multimedia and Application Packages — SQL/MM Part 5: Still Image

1 Scope

This part of ISO SQL/MM:

a) introduces this part (Still Image) of this International Standard,

b) gives the references necessary for this part of this International Standard,

c) defines notations and conventions specific to this part of this International Standard,

d) defines concepts specific to this part of this International Standard,

e) defines the still image data types and their associated routines,

2 Normative references

The following standards and publicly available specifications contain provisions that, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards and publicly available specifications are subject to revision and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of standards and public specifications listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

2.1 International standards

ISO/IEC 9075-1:199x, *Information Technology — Database Languages — SQL — Part 1: Framework (SQL/Framework).*

ISO/IEC 9075-2:199x, *Information Technology — Database Languages — SQL — Part 2: Foundation (SQL/Foundation).*

ISO/IEC 9075-4:1996, *Information Technology — Database Languages — SQL — Part 4: Persistent Stored Modules (SQL/PSM).*

ISO/IEC 9075-5:199x, *Information Technology — Database Languages — SQL — Part 5: Host Language Bindings (SQL/Bindings).*

ISO/IEC 13249-1:199x, *Information Technology — Database Languages — SQL Multimedia and Application Packages — Part 1: Framework.*

3 Definitions, notations, and conventions

3.1 Definitions

For the purpose of this International Standard, the following definitions apply.

3.1.1 Definitions provided in this Standard

To be supplied.

3.2 Notations

The notation used in ISO/IEC 13249-5 is defined in ISO/IEC 9075-1.

3.3 Conventions

3.3.1 Subclause structure

Subclauses for a type definition and its associated routines will be structured as follows:

x.3.5 SI_SampleDataType Type and Routines

x.3.5.1 SI_SampleDataType Type

x.3.5.2 SI_UserDefinedFunction1 Function

x.3.5.3 SI_UserDefinedFunction2 Function

x.3.5.4 SI_UserDefinedFunction3 Function

3.3.2 Structure within a subclause

1) **Purpose**: The Purpose section contains a brief description of the purpose of the type or routine.

2) **Definition**: This section contains the ISO/IEC 9075 syntax used to define the type or routine. In the case of routine specifications, the routine body should be defined using the facilities of ISO/IEC 9075-4 (SQL/PSM) where possible. This clause is in the Courier font. <key word>s, as defined in ISO/IEC 9075, are in uppercase. Parameter and variable identifiers are in lower case or mixed case. Data type, attribute and SQL-invoked routine identifiers are specified as defined in Subclause 3.3.3, “Data type, attribute and SQL-invoked routine identifiers”.

3) **Definitional Rules**: This section contains an enumerated list of rules to be applied when defining the type or routine. If this section is empty, the section heading is not present.

4) **Description**: This section contains an enumerated list of rules describing the type or routine. For a type, the first item contains a statement indicating the attributes and routines that are part of the public specification. For a routine, the first item contains the definition of the routine’s parameters. If this section is empty, the section heading is not present.

3.3.3 Data type, attribute and SQL-invoked routine identifiers

Data type identifiers, attribute identifiers and routine identifiers:

1) shall have a prefix. Still Image uses “SI_”.

2) shall not use the underbar character except in the prefix (i.e. only Alphanumeric characters [a-zA-Z0-9]),

3) shall capitalize each word used in the identifier. For example: SI_StillImage,

4) shall be in Italic when used in the Definitional Rules and the Description sections.

3.3.4 Parameter identifiers

Parameter identifiers are in lowercase. Parameters are in Italic when used in the Definitional Rules and the Description sections. This will distinguish them from other identifiers used in the Definitional Rules and Description sections.

3.3.5 Meta-variables

Meta-variables used to define implementation-dependent or implementation-defined constants such as SI_MaxCommentLength follow the conventions of Subclause 3.3.3, “Data type, attribute and SQL-invoked routine identifiers”.

3.3.6 Definitional variables

Definitional variables used in the Definitional Rules or Description sections are in uppercase Italics. This will distinguish them from other identifiers in the Definitional Rules or Description sections.

3.3.7 Exceptions

Except where otherwise specified, the phrase “an exception condition is raised:”, followed by the name of a condition, is used in the Definitional Rules and Description sections to indicate that:

— The execution of a routine is unsuccessful.

— Application of Definitional Rules or Description items may be terminated.

The effect on any assignment target and SQL descriptor area of an SQL-statement that terminates with an exception condition, unless explicitly defined by ISO/IEC 9075, is implementation-dependent.

The phrase “a completion condition is raised:”, followed by the name of a condition, is used in Definitional Rules and Description sections to indicate that application of Definitional Rules or Description items is not terminated and diagnostic information is to be made available; unless an exception condition is raised, the execution of the SQL-statement is successful.

4 Concepts

4.1 Schemas

ISO/IEC 9075 specifies that an object such as an SQL-invoked routine, a user-defined type, a domain, a table, a view, or a privilege is part of exactly one schema.

This International Standard does not include statements for creating schemas. An implementation-defined set of <schema definition> statements shall be effectively executed such that each <schema definition> statement that contains a <schema element> for a schema object defined in this International Standard shall contain exactly one <schema element> for each object defined by this International Standard. The number of such schemas and their names is implementation-defined.

It is assumed that the default character set of the SQL-schema in which an SQL-invoked routine specified in this International Standard is created includes the characters used in all character string literals contained in the body of that routine and a space character denoted by a blank space in such literals.

4.2 USAGE Privileges on User-defined Types

ISO/IEC 9075 specifies that users must have the USAGE privilege on a domain or a user-defined type before they can use it for defining other objects such as SQL-invoked routines, tables, view, domains or user-defined types.

This International Standard does not include the GRANT USAGE statements for the domains and user-defined types defined in this International Standard. For each object defined by this International Standard, a GRANT statement granting USAGE privilege to an implementation-defined set of grantees shall be effectively executed when these domains and user-defined types are created, except when explicitly noted by the Definitional Rules in this International Standard. It is implementation-defined whether the GRANT statement includes WITH GRANT OPTION.

4.3 UNDER Privileges on User-defined Types

ISO/IEC 9075 specifies users must have the UNDER privilege on a user-defined type *A* before they can use it for defining subtypes of *A*.

This International Standard does not include the GRANT UNDER statements for the user-defined types defined in this International Standard. For each object defined by this International Standard, a GRANT statement granting UNDER privilege to an implementation-defined set of grantees shall be effectively executed when these user-defined types are created, except when explicitly noted by the Definitional Rules in this International Standard. It is implementation-defined whether the GRANT statement includes WITH GRANT OPTION.

4.4 EXECUTE Privileges on Routines

ISO/IEC 9075 specifies that users must have the EXECUTE privilege on a routine before they can execute it.

This International Standard does not include the GRANT EXECUTE statements for the routines defined in this standard. For each routine defined by this International Standard, a GRANT statement granting EXECUTE privilege to an implementation-defined set of grantees shall be effectively executed when the routine is created, except where explicitly noted by the Definitional Rules in this International Standard. It is implementation-defined whether the GRANT statement includes WITH GRANT OPTION.

5 Still Image Data Types

The types in this family provide for the storage and retrieval of still image values.

5.1 SI_StillImage Types and Routines

5.1.1 SI_StillImage Type

Purpose

The SI_StillImage data types provides the definition of the root type of all still image abstract data types.

Definition

```
CREATE TYPE SI_StillImage NOT INSTANTIABLE
   (bitsPerColor     INTEGER,
   bitsPerPixelINTEGER,
   comment           CHARACTER VARYING(SI_MaxCommentLength),
   content           BINARY LARGE OBJECT(SI_MaxContentLength),
   contentLength     INTEGER,
   format      CHARACTER VARYING(8),
   height      INTEGER,
   importTime        TIMESTAMP(6),
   updateTime        TIMESTAMP(6),
   width             INTEGER)
```

****Editor's Note 5-001****
The SI_StillImage ADT defines the attributes bitsPerColor and bitsPerPixel. It might be better if one or both of these attributes was defined only for subtypes for which they can be supported.

****Editor's Note 5-003****
The original SI_StillImage ADT defined in SQL/MM MCI-012 contained a thumbnail attribute which was meant to contain a “smaller” image. While it was agreed to delete this attribute since SQL3 does not yet support “deep references” it was suggested that a thumbnail could be supplied for some or all subtypes as a virtual attribute (computed from *content* attribute).

Definitional Rules

1) *SI_MaxCommentLength* is the implementation-defined maximum length for the character representation of a StillImage *comment*.

2) *SI_MaxContentLength* is the implementation-defined maximum length for the binary representation of a StillImage *content*.

Description

1) The *SI_StillImage* type provides for public use:

   a) a function *commentText(SI_StillImage, CHARACTER VARYING)* .

2) The SI_StillImage mutator functions for the attributes *bitsPerColor*, *bitsPerPixel*, *comment*, *content*, *contentLength*, *format*, *height*, *importTime*, *updateTime* and *width* are private.

5.1.2 commentText Function

Purpose

Update an SI_StillImage instance from a character string comment

Definition

```
CREATE FUNCTION commentText
   (image    SI_StillImage RESULT,
    text     CHARACTER VARYING(SI_MaxCommentLength))
   RETURNS SI_StillImage
   LANGUAGE SQL
   BEGIN
      DECLARE NullInstance EXCEPTION FOR SQLSTATE '2202A';
      IF image IS NULL THEN
         SIGNAL NullInstance;
      END IF;
      SET image>>comment = text;
      SET image>>updateTime = CURRENT_TIMESTAMP;
   RETURN image;
END
```

Definitional Rules

1) *SI_MaxCommentLength* is the implementation-defined maximum length for the character representation of a StillImage *comment*.

Description

1) The function *commentText(SI_StillImage, CHARACTER VARYING)* takes the following input parameters:

   a) an *SI_StillImage* value *image*,

   b) a *CHARACTER VARYING* value *text*.

5.2 SI_JPEG Type and Routines

5.2.1 SI_JPEG Type

Purpose

The *SI_JPEG* type provides the definition of the JPEG still image data type.

Definition

```
CREATE TYPE SI_JPEG
   UNDER SI_StillImage
```

Description

1) The *SI_JPEG* type provides for public use:

   a) a function *SI_JPEG(BINARY LARGE OBJECT, CHARACTER VARYING)*,

   b) a function *SI_JPEG(BINARY LARGE OBJECT),*

   c) a function *setContent(SI_JPEG, BINARY LARGE OBJECT).*

2) The SI_JPEG mutator functions for the attributes *bitsPerColor, bitsPerPixel, comment, content, contentLength, format, height, importTime, updateTime* and *width* are private.

****Editor's Note 5-002****

The current SI_JPEG ADT does not define any new attributes. It is a language opportunity to define attributes that are specific to the JPEG sub-type.

5.2.2 SI_JPEG Functions

Purpose

Construct and initialize an *SI_JPEG* instance.

Definition

```
CREATE FUNCTION SI_JPEG
   (content    BINARY LARGE OBJECT(SI_MaxContentLength),
    comment    CHARACTER VARYING(SI_MaxCommentLength),
   RETURNS SI_JPEG
   LANGUAGE SQL
   BEGIN
      DECLARE InvalidImageFormat EXCEPTION FOR SQLSTATE 'H5F02';

      DECLARE image        SI_JPEG;

      IF isJPEG(content) = FALSE THEN
         SIGNAL InvalidImageFormat;
      END IF;

      SET image = SI_JPEG();
      SET image>>bitsPerColor = bitsPerColor(content);
      SET image>>bitsPerPixel = bitsPerPixel(content);
      SET image>>comment = comment;
      SET image>>content = content;
      SET image>>contentLength = LENGTH(content);
      SET image>>format = 'JPEG';
      SET image>>height = height(content);
      SET image>>importTime = CURRENT_TIMESTAMP;
      SET image>>updateTime = NULL;
      SET image>>width = width(content);

      RETURN image;
   END

CREATE FUNCTION SI_JPEG
   (content    BINARY LARGE OBJECT(SI_MaxContentLength))
   RETURNS SI_JPEG
   LANGUAGE SQL
   BEGIN
      DECLARE InvalidImageFormat EXCEPTION FOR SQLSTATE 'H5F02';

      DECLARE image        SI_JPEG;

      IF isJPEG(content) = FALSE THEN
         SIGNAL InvalidImageFormat;
      END IF;

      SET image = SI_JPEG();
      SET image>>bitsPerColor = bitsPerColor(content);
      SET image>>bitsPerPixel = bitsPerPixel(content);
      SET image>>comment = NULL;
      SET image>>content = content;
      SET image>>contentLength = LENGTH(content);
      SET image>>format = 'JPEG';
      SET image>>height = height(content);
      SET image>>importTime = CURRENT_TIMESTAMP;
      SET image>>updateTime = NULL;
      SET image>>width = width(content);

      RETURN image;
   END
```

Definitional Rules

1) *SI_MaxCommentLength* is the implementation-defined maximum length for the character representation of a StillImage *comment*.

2) *SI_MaxContentLength* is the implementation-defined maximum length for the binary representation of a StillImage *content*.

Description

1) The function *SI_JPEG(BINARY LARGE OBJECT, CHARACTER VARYING)* takes the following input parameters:

   a) a *BINARY LARGE OBJECT* value *content*,

   b) a *CHARACTER VARYING* value *comment*.

2) The function *SI_JPEG(BINARY LARGE OBJECT)* takes the following input parameters:

   a) a *BINARY LARGE OBJECT* value *content*.

5.2.3 setContent Function

Purpose

Update the *SI_JPEG* content.

Definition

```
CREATE FUNCTION setContent
   (imageSI_JPEG RESULT,
    content    BINARY LARGE OBJECT(SI_MaxContentLength))
   RETURNS SI_JPEG
   LANGUAGE SQL
   BEGIN
      DECLARE NullInstance EXCEPTION FOR SQLSTATE '2202A';
      DECLARE InvalidInput EXCEPTION FOR SQLSTATE 'H5F01';
      DECLARE InvalidImageFormat EXCEPTION FOR SQLSTATE 'H5F02';

      IF image IS NULL THEN
         SIGNAL NullInstance;
      END IF;
      IF image>>format <> 'JPEG' THEN
         SIGNAL InvalidInput;
      END IF;

      IF isJPEG(content) = FALSE THEN
         SIGNAL InvalidImageFormat;
      END IF;

      SET image>>bitsPerColor = bitsPerColor(content);
      SET image>>bitsPerPixel = bitsPerPixel(content);
      SET image>>comment = NULL;
      SET image>>content = content;
      SET image>>contentLength = LENGTH(content);
      SET image>>format = 'JPEG';
      SET image>>height = height(content);
      SET image>>updateTime = CURRENT_TIMESTAMP;
      SET image>>width = width(content);

      RETURN image;
   END
```

Definitional Rules

1) *SI_MaxContentLength* is the implementation-defined maximum length for the binary representation of a StillImage *content*.

Description

1) The function setContent(*SI_JPEG, BINARY LARGE OBJECT)* takes the following input parameters:

   a) a *SI_JPEG* value *image*,

b) a *BINARY LARGE OBJECT* value *content*.

5.2.4 Private Functions

Purpose

SI_JPEG private functions.

Definition

```
CREATE FUNCTION bitsPerColor
  (content BINARY LARGE OBJECT(SI_MaxContentLength)) RETURNS INTEGER
      --
      -- !! See Description
      --


CREATE FUNCTION bitsPerPixel
  (content BINARY LARGE OBJECT(SI_MaxContentLength)) RETURNS INTEGER
      --
      -- !! See Description
      --



CREATE FUNCTION height
  (content BINARY LARGE OBJECT(SI_MaxContentLength)) RETURNS INTEGER
      --
      -- !! See Description
      --


CREATE FUNCTION isJPEG
  (content BINARY LARGE OBJECT(SI_MaxContentLength)) RETURNS BOOLEAN
      --
      -- !! See Description
      --


CREATE FUNCTION width
  (content BINARY LARGE OBJECT(SI_MaxContentLength)) RETURNS INTEGER
      --
      -- !! See Description
      --
```

Definitional Rules

1) *SI_MaxContentLength* is the implementation-defined maximum length for the binary representation of a StillImage *content*.

Description

1. The private function *bitsPerColor(BINARY LARGE OBJECT)* determines and returns the number of bits used to represent the true color of a pixel in the *content* of an instance of SI_JPEG.

2. The private function *bitsPerPixel(BINARY LARGE OBJECT)* determines and returns the maximum number of bits used to represent a pixel in the *content* of an instance of SI_JPEG.

3. The private function *height(BINARY LARGE OBJECT)* determines and returns the height in pixels of a JPEG image from the *content* attribute of the SI_JPEG abstract data type.

4. The private function *isJPEG(BINARY LARGE OBJECT)* determines if its argument contains a JPEG image.

5. The private function *width (BINARY LARGE OBJECT)*determines and returns the width in pixels of a JPEG image from the *content* attribute of the SI_JPEG abstract data type.

6 Status Codes

The character string value returned in an SQLSTATE parameter comprises a 2-character class value followed by a 3-character subclass value. The class value for each condition and the subclass value or values for each class value are specified in Table 1 - SQLSTATE class and subclass values.

The "Category" column has the following meanings: "S" means that the class value given corresponds to successful completion and is a completion condition; "W" means that the class value given corresponds to a successful completion but with a warning and is a completion condition; "N" means that the class value corresponds to a no-data situation and is a completion condition; "X" means that the class value given corresponds to an exception condition.

Table 1 – SQLSTATE class and subclass values

Category	Condition	Class	Subcondition	Subclass
X	data exception	22	null instance used in mutator function	02D
X	SQL/MM Still Image	H5	Invalid input	F01
X	SQL/MM Still Image	H5	Invalid image format	F02

7 Conformance

7.1 Introduction

This part of ISO/IEC 13249 specifies conforming SQL/MM Still Image implementations.

A conforming SQL/MM Still Image implementation shall support the public Still Image data types and functions according to the associated Definitions and Description Rules specified in this part of ISO/IEC 13249.

A conforming SQL/MM Still Image implementation shall supply <SQL-invoked function>s whose <routine body> is either a <SQL routine body> or an <external body reference> that specifies PARAMETER STYLE SQL as defined in Subclause 12.5, “<SQL-invoked routine>“ in part 2 of ISO 9075.

A conforming SQL/MM Still Image implementation is not required to perform the exact sequence of actions defined in the Description Rules or in the <SQL routine body>s contained this International Standard, but shall achieve the same effect as that sequence.

7.2 Relationship to other International Standards

****Editor's Note 5-008****

Relationships to other International Standards to be supplied. This section needs to explain the dependency on ISO/IEC 9075.

7.3 Claims of conformance

Claims of conformance to this part of ISO/IEC 13249 shall state:

****Editor's Note 5-009****

Claims of conformance are to be supplied.

1) The definitions for all elements and actions that this part of ISO/IEC 13049 specifies as implementation-defined.

7.4 Extensions and options

A conforming implementation may provide support for additional implementation-defined routines defined using the Still Image data types.

An implementation remains conforming even if it provides user options to process Still Image routines in a nonconforming manner.

Index

24 Index