



# Information technology — Database languages — SQL Multimedia and Application Packages —

## Part 3: Spatial

### TECHNICAL CORRIGENDUM 1

*Technologies de l'information — Langages de bases de données — Multimédia SQL et paquetages d'application —*

*Partie 3: Spatial*

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 13249-3:1999 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

#### Statement of purpose of rationale:

A statement indicating the rationale for each change to ISO/IEC 13249-3:1999(E) is included. This is to inform the users of that standard as to the reason why it was judged necessary to change the original wording. In many cases the reason is editorial or to clarify the wording; in some cases it is to correct an error or an omission in the original wording.

#### Notes on numbering:

Where this Corrigendum introduces new Definitional Rules and Descriptions, the new rules have been numbered as follows:

Rules inserted between, for example, Rules 7) and 8) are numbered 7.1), 7.2), etc. [or 7) a.1), 7) a.2), etc.]. Those inserted before Rule 1) are numbered 0.1), 0.2), etc.

Where this Corrigendum introduces new subclauses, the new subclauses have been numbered as follows:

Subclauses inserted between, for example, subclause 4.3.2 and 4.3.3 are numbered 4.3.2a, 4.3.2b, etc.

Those inserted before, for example, 4.3.1 are numbered 4.3.0, 4.3.0a, etc.

Contents		Page
3.1.2.3	2-dimensional geometry .....	3
3.1.5.5	meridian .....	3
3.1.2.7	closed curve .....	3
3.1.2.12	linear ring .....	3
3.1.2.14	non-closed curve .....	3
3.1.2.17	ring .....	3
4.1.10	ST_CurvePolygon .....	3
4.1.15	ST_MultiLineString .....	4
4.1.16	ST_MultiSurface .....	4
4.1.17	ST_MultiPolygon .....	4
5.1.13	ST_Buffer Method .....	4
5.1.19	ST_Distance Method .....	4
6.1.1	ST_Point Type .....	5
6.1.2	ST_Point Methods .....	5
7.1.2	ST_Length Method .....	6
7.2.1	ST_LineString Type .....	6
7.2.2	ST_LineString Methods .....	6
7.2.5	ST_PointN Method .....	7
7.3.1	ST_CircularString Type .....	7
7.3.2	ST_CircularString Methods .....	8
7.3.5	ST_PointN Method .....	8
7.4.1	ST_CompoundCurve Type .....	9
7.4.2	ST_CompoundCurve Methods .....	10
7.4.5	ST_CurveN Method .....	10
8.1.2	ST_Area Method .....	11
8.1.3	ST_Perimeter Method .....	11
8.2.1	ST_CurvePolygon Type .....	11
8.2.2	ST_CurvePolygon Methods .....	12
8.2.6	ST_InteriorRingN Method .....	13
8.3.1	ST_Polygon Type .....	14
8.3.2	ST_Polygon Methods .....	15
8.3.9	ST_BdPolyFromWKB Functions .....	16
9.1.1	ST_GeomCollection Type .....	16
9.1.2	ST_GeomCollection Methods .....	17
9.1.5	ST_GeometryN Method .....	18
9.2.1	ST_MultiPoint Type .....	18
9.2.2	ST_MultiPoint Methods .....	19
9.3.1	ST_MultiCurve Type .....	19
9.3.2	ST_MultiCurve Methods .....	20
9.4.1	ST_MultiLineString Type .....	20
9.4.2	ST_MultiLineString Methods .....	21
9.5.1	ST_MultiSurface Type .....	21
9.5.2	ST_MultiSurface Methods .....	22
9.6.1	ST_MultiPolygon Type .....	22
9.6.2	ST_MultiPolygon Methods .....	23
10.1.1	ST_SpatialRefSys Type .....	23
10.1.2	ST_SpatialRefSys Methods .....	23
13	Status Codes .....	24
Annex A (informative) Implementation-defined elements .....		25

**3.1.2.3 2-dimensional geometry**

1. *Rationale: Correct description.*

Replace definition with:

a geometry with a geometric dimension of 2

**3.1.5.5 meridian**

1. *Rationale: Improve and correct description.*

Replace the definition with:

intersection between an ellipsoid and a plane containing the semi-minor axis of the ellipsoid

NOTE This term is often used for the pole-to-pole arc rather than the complete closed figure.

**3.1.2.7 closed curve**

1. *Rationale: Correct description.*

Replace definition with:

a curve such that its start point is equal to its end point

**3.1.2.12 linear ring**

1. *Rationale: Correct description.*

Replace definition with:

a linestring that is closed and simple

**3.1.2.14 non-closed curve**

1. *Rationale: Correct description.*

Replace definition with:

a curve such that its start point is not equal to its end point

**3.1.2.17 ring**

1. *Rationale: Correct description.*

Replace definition with:

a curve that is closed and simple

**4.1.10 ST\_CurvePolygon**

1. *Rationale: Correct description.*

Replace the second paragraph with:

ST\_CurvePolygon values are topologically closed. The boundary of an ST\_CurvePolygon consists of an exterior ring and zero or more interior rings. No two rings in the boundary cross. The rings in the boundary of an ST\_CurvePolygon value may intersect at a point but only as a tangent. An ST\_CurvePolygon shall not have cut lines, spikes or punctures. The interior of every ST\_CurvePolygon is a connected point set. The exterior of an ST\_CurvePolygon with one or more holes is not connected. Each hole defines a disconnected component of the exterior.

#### 4.1.15 ST\_MultiLineString

1. *Rationale: Correct type hierarchy description.*

Replace the paragraph with:

The ST\_MultiLineString type is a subtype of ST\_MultiCurve. The elements of an ST\_MultiLineString are restricted to ST\_LineString values.

#### 4.1.16 ST\_MultiSurface

1. *Rationale: Correct description.*

Replace the paragraph with:

The ST\_MultiSurface type is a subtype of ST\_GeomCollection. The ST\_MultiSurface type may be instantiable. An ST\_MultiSurface is a 2-dimensional geometry collection. The elements of an ST\_MultiSurface value are restricted to ST\_Surface values. The interiors of any two ST\_Surface values in an ST\_MultiSurface shall not intersect. The boundaries of any two elements in an ST\_MultiSurface may intersect at a finite number of ST\_Point values.

#### 4.1.17 ST\_MultiPolygon

1. *Rationale: Correct description.*

Replace the first paragraph with:

The ST\_MultiPolygon type is a subtype of ST\_MultiSurface. The elements of an ST\_MultiPolygon value are restricted to ST\_Polygon values. The interiors of distinct element of an ST\_MultiPolygon do not intersect. The interiors of two ST\_Polygon values that are elements of an ST\_MultiPolygon shall not intersect. The boundaries of any two ST\_Polygon values that are elements of an ST\_MultiPolygon shall not cross and may touch at only a finite number of points. An ST\_MultiPolygon value is defined to be topologically closed.

Replace the second paragraph with:

An ST\_MultiPolygon value shall not have cut lines, spikes or punctures. An ST\_MultiPolygon value is a closed point set. The interior of an ST\_MultiPolygon value with more than one ST\_Polygon value is not a connected point set. The number of connected components of the interior of an ST\_MultiPolygon is equal to the number of ST\_Polygon values in the ST\_MultiPolygon. The boundary of an ST\_MultiPolygon value is a set of linear rings corresponding to the boundaries of the ST\_Polygon elements.

#### 5.1.13 ST\_Buffer Method

1. *Rationale: Correct the parameter value description.*

Replace Description 2) with:

- 2) The parameter *adistance* is measured in an implementation-defined linear unit of measure in the spatial reference system of SELF.

#### 5.1.19 ST\_Distance Method

1. *Rationale: Correct the returned value description.*

Replace Description 3) with:

- 3) Case:
  - a) If the spatial reference system of SELF defines a <linear unit>, then the value returned by *ST\_Distance(ST\_Geometry)* is in the linear unit of measure identified by <linear unit>.
  - b) Otherwise, the value returned by *ST\_Distance(ST\_Geometry)* is in an implementation-defined unit of measure.

### 6.1.1 ST\_Point Type

#### 1. Rationale: Correct constructor method definitions.

Replace the two <original method specification>s that begins with "METHOD ST\_Point" in the Definition with:

```

CONSTRUCTOR METHOD ST_Point
  (xcoord DOUBLE PRECISION, ycoord DOUBLE PRECISION)
  RETURNS ST_Point
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_Point
  (xcoord DOUBLE PRECISION, ycoord DOUBLE PRECISION, asrid INTEGER)
  RETURNS ST_Point
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

```

### 6.1.2 ST\_Point Methods

#### 1. Rationale: Correct constructor method definitions.

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_Point
  (xcoord DOUBLE PRECISION, ycoord DOUBLE PRECISION)
  RETURNS ST_Point
  FOR ST_Point
  RETURN SELF.
  ST_PrivateDimension(0).           -- Return an ST_Point value with
  ST_PrivateCoordinateDimension(2). -- dimension = 0,
  ST_SRID(0).                       -- coordinate dimension = 2,
  ST_X(xcoord).                     -- SRID = asrid,
  ST_Y(ycoord).                     -- ST_X = xcoord,
                                   -- ST_Y = ycoord

CREATE CONSTRUCTOR METHOD ST_Point
  (xcoord DOUBLE PRECISION, ycoord DOUBLE PRECISION, asrid INTEGER)
  RETURNS ST_Point
  FOR ST_Point
  RETURN SELF.
  ST_PrivateDimension(0).           -- Return an ST_Point value with
  ST_PrivateCoordinateDimension(2). -- dimension = 0,
  ST_SRID(asrid).                  -- coordinate dimension = 2,
  ST_X(xcoord).                    -- SRID = asrid,
  ST_Y(ycoord).                    -- ST_X = xcoord,
                                   -- ST_Y = ycoord

```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_Point(DOUBLE PRECISION, DOUBLE PRECISION)* returns an *ST\_Point* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_Point(DOUBLE PRECISION, DOUBLE PRECISION, INTEGER)* returns an *ST\_Point* value with:

### 7.1.2 ST\_Length Method

1. *Rationale: Correct the returned value description.*

Replace Description 3) with:

3) Case:

- a) If the spatial reference system of SELF defines a <linear unit>, then the value returned by *ST\_Length()* is in the linear unit of measure identified by <linear unit>.
- b) Otherwise, the value returned by *ST\_Length()* is in an implementation-defined unit of measure.

### 7.2.1 ST\_LineString Type

1. *Rationale: Correct constructor method definitions.*

Replace the two <original method specification>s that begins with "METHOD ST\_LineString" in the Definition with:

```

CONSTRUCTOR METHOD ST_LineString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_LineString
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_LineString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_LineString
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,
    
```

### 7.2.2 ST\_LineString Methods

1. *Rationale: Correct constructor method definitions.*

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_LineString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_LineString
  FOR ST_LineString
  RETURN SELF.ST_SRID(0).ST_Points(apointarray)

CREATE CONSTRUCTOR METHOD ST_LineString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_LineString
  FOR ST_LineString
  RETURN SELF.ST_SRID(asrid).ST_Points(apointarray)
    
```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_LineString(ST\_Point ARRAY)* returns an *ST\_LineString* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_LineString(ST\_Point ARRAY, INTEGER)* returns an *ST\_LineString* value with:

### 7.2.5 ST\_PointN Method

1. *Rationale: Use a return value for such completion codes.*

Replace the Definition with the following:

```
CREATE METHOD ST_PointN(aosition INTEGER)
  RETURNS ST_Point
  FOR ST_LineString
  BEGIN
    IF SELF.ST_NumPoints = 0 THEN
      BEGIN
        SIGNAL SQLSTATE '01F06'
          SET MESSAGE_TEXT = 'empty geometry';
        RETURN NULL;
      END;
    END IF;
    IF aosition < 1 OR
       aosition > SELF.ST_NumPoints THEN
      BEGIN
        SIGNAL SQLSTATE '01F01'
          SET MESSAGE_TEXT = 'invalid position';
        RETURN NULL;
      END;
    END IF;
    RETURN SELF.ST_PrivatePoints[aosition];
  END
```

In the Description section, replace Description 2) with:

2) For the null-call method *ST\_PointN(INTEGER)*:

Case:

- a) If the cardinality of the *ST\_PrivatePoints* attribute is equal to 0 (zero), then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – empty geometry.*
  - ii) Return the null value.
- b) If *aosition* is less than 1 (one) or greater than the cardinality of the *ST\_PrivatePoints* attribute, then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – invalid position.*
  - ii) Return the null value.
- c) Otherwise, return an *ST\_Point* value at element *aosition* in the *ST\_PrivatePoints* attribute of SELF.

### 7.3.1 ST\_CircularString Type

1. *Rationale: Correct constructor method definitions.*

Replace the two <original method specification>s that begins with "METHOD ST\_CircularString" in the Definition with:

```
CONSTRUCTOR METHOD ST_CircularString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_CircularString
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,
```

```

CONSTRUCTOR METHOD ST_CircularString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
RETURNS ST_CircularString
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

```

### 7.3.2 ST\_CircularString Methods

1. *Rationale: Correct constructor method definitions.*

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_CircularString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements])
RETURNS ST_CircularString
FOR ST_CircularString
RETURN SELF.ST_SRID(0).ST_Points(apointarray)

CREATE CONSTRUCTOR METHOD ST_CircularString
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
RETURNS ST_CircularString
FOR ST_CircularString
RETURN SELF.ST_SRID(asrid).ST_Points(apointarray)

```

Replace the first sentence of Description 2) with:

- 2) The null call type preserving SQL-invoked constructor method *ST\_CircularString(ST\_Point ARRAY)* returns an *ST\_CircularString* value with:

Replace the first sentence of Description 4) with:

- 4) The null call type preserving SQL-invoked constructor method *ST\_CircularString(ST\_Point ARRAY, INTEGER)* returns an *ST\_CircularString* value with:

### 7.3.5 ST\_PointN Method

1. *Rationale: Use a return value for such completion codes.*

Replace the Definition with the following:

```

CREATE METHOD ST_PointN(aosition INTEGER)
RETURNS ST_Point
FOR ST_CircularString
BEGIN
  IF SELF.ST_NumPoints = 0 THEN
    BEGIN
      SIGNAL SQLSTATE '01F06'
      SET MESSAGE_TEXT = 'empty geometry';
      RETURN NULL;
    END;
  END IF;
  IF aosition < 1 OR
  aosition > SELF.ST_NumPoints THEN
    BEGIN
      SIGNAL SQLSTATE '01F01'
      SET MESSAGE_TEXT = 'invalid position';
      RETURN NULL;
    END;
  END IF;
  RETURN SELF.ST_PrivatePoints[aosition];
END

```

In the Description section, replace Description 2) with:

2) For the null-call method *ST\_PointN(INTEGER)*:

Case:

- a) If the cardinality of the attribute *ST\_PrivatePoints* is equal to 0 (zero), then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – empty geometry*.
  - ii) Return the null value.
- b) If *aposition* is less than 1 (one) or greater than the cardinality of the attribute *ST\_PrivatePoints*, then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – invalid position*.
  - ii) Return the null value.
- c) Otherwise, return an *ST\_Point* value at element *aposition* in the attribute *ST\_PrivatePoints* of SELF.

#### 7.4.1 ST\_CompoundCurve Type

1. *Rationale: Correct constructor method definitions.*

Replace the four <original method specification>s that begins with "METHOD *ST\_CompoundCurve*" in the Definition with:

```

CONSTRUCTOR METHOD ST_CompoundCurve (acurve ST_Curve)
  RETURNS ST_CompoundCurve
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_CompoundCurve
  (acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_CompoundCurve
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_CompoundCurve
  (acurve ST_Curve,
   asrid INTEGER)
  RETURNS ST_CompoundCurve
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_CompoundCurve
  (acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_CompoundCurve
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

```

## 7.4.2 ST\_CompoundCurve Methods

### 1. Rationale: Correct constructor method definitions.

Replace the Definition with:

```
CREATE CONSTRUCTOR METHOD ST_CompoundCurve
  (acurve ST_Curve)
  RETURNS ST_CompoundCurve
  FOR ST_CompoundCurve
  RETURN SELF.ST_SRID(0).ST_Curves(ARRAY[acurve])

CREATE CONSTRUCTOR METHOD ST_CompoundCurve
  (acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_CompoundCurve
  FOR ST_CompoundCurve
  RETURN SELF.ST_SRID(0).ST_Curves(acurvearray)

CREATE CONSTRUCTOR METHOD ST_CompoundCurve
  (acurve ST_Curve,
   asrid INTEGER)
  RETURNS ST_CompoundCurve
  FOR ST_CompoundCurve
  RETURN SELF.ST_SRID(asrid).ST_Curves(ARRAY[acurve])

CREATE CONSTRUCTOR METHOD ST_CompoundCurve
  (acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_CompoundCurve
  FOR ST_CompoundCurve
  RETURN SELF.ST_SRID(asrid).ST_Curves(acurvearray)
```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_CompoundCurve(ST\_Curve)* returns an *ST\_CompoundCurve* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_CompoundCurve(ST\_Curve ARRAY)* returns an *ST\_CompoundCurve* value with:

Replace the first sentence of Description 6) with:

- 6) The null-call type preserving SQL-invoked constructor method *ST\_CompoundCurve(ST\_Curve, INTEGER)* returns an *ST\_CompoundCurve* value with:

Replace the first sentence of Description 8) with:

- 8) The null-call type preserving SQL-invoked constructor method *ST\_CompoundCurve(ST\_Curve ARRAY, INTEGER)* returns an *ST\_CompoundCurve* value with:

## 7.4.5 ST\_CurveN Method

### 1. Rationale: Use a return value for such completion codes.

Replace the Definition with the following:

```
CREATE METHOD ST_CurveN(aosition INTEGER)
  RETURNS ST_Curve
  FOR ST_CompoundCurve
  BEGIN
    IF CARDINALITY(SELF.ST_PrivateCurves) = 0 THEN
      BEGIN
        SIGNAL SQLSTATE '01F06'
          SET MESSAGE_TEXT = 'empty geometry';
        RETURN NULL;
      END;
    END IF;
```

```

IF aposition < 1 OR
  aposition > CARDINALITY(SELF.ST_PrivateCurves) THEN
  BEGIN
    SIGNAL SQLSTATE '01F01'
      SET MESSAGE_TEXT = 'invalid position';
    RETURN NULL;
  END;
END IF;
RETURN SELF.ST_PrivateCurves[aposition];
END

```

In the Description section, replace Description 2) with:

2) For the null-call method *ST\_CurveN(INTEGER)*:

Case:

- a) If the cardinality of the *ST\_PrivateCurves* attribute is equal to 0 (zero), then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – geometry*.
  - ii) Return the null value.
- b) If *aposition* is less than 1 (one) or greater than the cardinality of the *ST\_PrivateCurves* attribute, then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – invalid position*.
  - ii) Return the null value.
- c) Otherwise, return an *ST\_Curve* value at element *aposition* in the *ST\_PrivateCurves* attribute of SELF.

### 8.1.2 ST\_Area Method

1. *Rationale: Correct the returned value description.*

Replace Description 3) with:

3) Case:

- a) If the spatial reference system of SELF defines a <linear unit>, then the value returned by *ST\_Area()* is in the linear unit of measure identified by <linear unit> squared.
- b) Otherwise, the value returned by *ST\_Area()* is in an implementation-defined unit of measure.

### 8.1.3 ST\_Perimeter Method

1. *Rationale: Correct the returned value description.*

Replace Description 3) with:

3) Case:

- a) If the spatial reference system of SELF defines a <linear unit>, then the value returned by *ST\_Perimeter()* is in the linear unit of measure identified by <linear unit> squared.
- b) Otherwise, the value returned by *ST\_Perimeter()* is in an implementation-defined unit of measure.

### 8.2.1 ST\_CurvePolygon Type

1. *Rationale: Correct constructor method definitions.*

Replace the four <original method specification>s that begins with "METHOD *ST\_CurvePolygon*" in the Definition with:

```

CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve)
  RETURNS ST_CurvePolygon
  SELF AS RESULT

```

```

LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve,
   acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements])
RETURNS ST_CurvePolygon
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve,
   asrid INTEGER)
RETURNS ST_CurvePolygon
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve,
   acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
RETURNS ST_CurvePolygon
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

```

2. *Rationale: Correct description.*

Replace the Description 11) with:

- 11) An *ST\_CurvePolygon* value shall not have cut lines, spikes or punctures:

$$\forall p \in ST\_CurvePolygon, p = \text{Closure}(\text{Interior}(p))$$

8.2.2 **ST\_CurvePolygon Methods**

1. *Rationale: Correct constructor method definitions.*

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve)
RETURNS ST_CurvePolygon
FOR ST_CurvePolygon
RETURN SELF.ST_SRID(0).ST_ExteriorRing(acurve) .
  ST_InteriorRings(CAST(ARRAY[] AS
  ST_Curve ARRAY[ST_MaxGeometryArrayElements])

CREATE CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve,
   acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements])
RETURNS ST_CurvePolygon
FOR ST_CurvePolygon
RETURN SELF.ST_SRID(0).ST_ExteriorRing(acurve) .
  ST_InteriorRings(acurvearray)

```

```

CREATE CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve,
   asrid INTEGER)
RETURNS ST_CurvePolygon
FOR ST_CurvePolygon
RETURN SELF.ST_SRID(asrid).ST_ExteriorRing(acurve) .
      ST_InteriorRings(CAST(ARRAY[] AS
      ST_Curve ARRAY[ST_MaxGeometryArrayElements])

CREATE CONSTRUCTOR METHOD ST_CurvePolygon
  (acurve ST_Curve,
   acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
RETURNS ST_CurvePolygon
FOR ST_CurvePolygon
RETURN SELF.ST_SRID(asrid).ST_ExteriorRing(acurve) .
      ST_InteriorRings(acurvearray)

```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_CurvePolygon(ST\_Curve)* returns an *ST\_CurvePolygon* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_CurvePolygon(ST\_Curve, ST\_Curve ARRAY)* returns an *ST\_CurvePolygon* value with:

Replace the first sentence of Description 6) with:

- 6) The null-call type preserving SQL-invoked constructor method *ST\_CurvePolygon(ST\_Curve, INTEGER)* returns an *ST\_CurvePolygon* value with:

Replace the first sentence of Description 8) with:

- 8) The null-call type preserving SQL-invoked constructor method *ST\_CurvePolygon(ST\_Curve, ST\_Curve ARRAY, INTEGER)* returns an *ST\_CurvePolygon* value with:

### 8.2.6 ST\_InteriorRingN Method

1. *Rationale: Use a return value for such completion codes.*

Replace the Definition with the following:

```

CREATE METHOD ST_InteriorRingN(aposition INTEGER)
RETURNS ST_Curve
FOR ST_CurvePolygon
BEGIN
  IF CARDINALITY(SELF.ST_PrivateInteriorRings) = 0 THEN
    BEGIN
      SIGNAL SQLSTATE '01F06'
      SET MESSAGE_TEXT = 'empty geometry';
      RETURN NULL;
    END;
  END IF;
  IF aposition < 1 OR
  aposition > CARDINALITY(SELF.ST_PrivateInteriorRings) THEN
    BEGIN
      SIGNAL SQLSTATE '01F01'
      SET MESSAGE_TEXT = 'invalid position';
      RETURN NULL;
    END;
  END IF;
  RETURN SELF.ST_PrivateInteriorRings[aposition];
END

```

In the Description section, replace Description 2) with:

- 2) For the null-call method *ST\_InteriorRingN(INTEGER)*:
 

Case:

  - a) If the cardinality of the *ST\_PrivateInteriorRings* attribute is equal to 0 (zero), then:
    - i) A completion condition is raised: *SQL/MM Spatial warning – geometry*.
    - ii) Return the null value.
  - b) If *aposition* is less than one or greater than the cardinality of the *ST\_PrivateInteriorRings* attribute, then:
    - i) A completion condition is raised: *SQL/MM Spatial warning – invalid position*.
    - ii) Return the null value.
  - c) Otherwise, return an *ST\_Curve* value at element *aposition* in the *ST\_PrivateInteriorRings* attribute of SELF.

### 8.3.1 ST\_Polygon Type

#### 1. Rationale: Correct constructor method definitions.

Replace the four <original method specification>s that begins with "METHOD ST\_Polygon" in the Definition with:

```

CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString)
  RETURNS ST_Polygon
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString,
   alinestringarray ST_LineString
   ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_Polygon
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString,
   asrid INTEGER)
  RETURNS ST_Polygon
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString,
   alinestringarray ST_LineString
   ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_Polygon
  SELF AS RESULT
  
```

```

LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

```

### 8.3.2 ST\_Polygon Methods

#### 1. Rationale: Correct constructor method definitions.

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString)
  RETURNS ST_Polygon
  FOR ST_Polygon
  RETURN SELF.ST_SRID(0).ST_ExteriorRing(alinestring).
    ST_InteriorRings(CAST(ARRAY[] AS
      ST_LineString ARRAY[ST_MaxGeometryArrayElements])

CREATE CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString,
   alinestringarray ST_LineString ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_Polygon
  FOR ST_Polygon
  RETURN SELF.ST_SRID(0).ST_ExteriorRing(alinestring).
    ST_InteriorRings(alinestringarray)

CREATE CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString,
   asrid INTEGER)
  RETURNS ST_Polygon
  FOR ST_Polygon
  RETURN SELF.ST_SRID(asrid).ST_ExteriorRing(alinestring).
    ST_InteriorRings(CAST(ARRAY[] AS
      ST_LineString ARRAY[ST_MaxGeometryArrayElements])

CREATE CONSTRUCTOR METHOD ST_Polygon
  (alinestring ST_LineString,
   alinestringarray ST_LineString ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_Polygon
  FOR ST_Polygon
  RETURN SELF.ST_SRID(asrid).ST_ExteriorRing(alinestring).
    ST_InteriorRings(alinestringarray)

```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_Polygon(ST\_LineString)* returns an *ST\_Polygon* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_Polygon(ST\_LineString, ST\_LineString ARRAY)* returns an *ST\_Polygon* value with:

Replace the first sentence of Description 6) with:

- 6) The null-call type preserving SQL-invoked constructor method *ST\_Polygon(ST\_LineString, INTEGER)* returns an *ST\_Polygon* value with:

Replace the first sentence of Description 8) with:

- 8) The null-call type preserving SQL-invoked constructor method *ST\_Polygon(ST\_LineString, ST\_LineString ARRAY, INTEGER)* returns an *ST\_Polygon* value with:

### 8.3.9 ST\_BdPolyFromWKB Functions

1. *Rationale: Change variable "wkb" to "awkb".*

Replace the Definition with:

```
CREATE FUNCTION ST_BdPolyFromWKB
  (awkb BINARY LARGE OBJECT(ST_MaxGeometryAsBinary))
  RETURNS ST_Polygon
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT
  STATIC DISPATCH
  RETURN ST_BdPolyFromWKB(awkb, 0)
```

```
CREATE FUNCTION ST_BdPolyFromWKB
  (awkb BINARY LARGE OBJECT(ST_MaxGeometryAsBinary),
  asrid INTEGER)
  RETURNS ST_Polygon
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT
  STATIC DISPATCH
  --
  -- See Description
  --
```

### 9.1.1 ST\_GeomCollection Type

1. *Rationale: Correct constructor method definitions.*

Replace the four <original method specification>s that begins with "METHOD ST\_GeomCollection" in the Definition with:

```
CONSTRUCTOR METHOD ST_GeomCollection
  (ageometry ST_Geometry)
  RETURNS ST_GeomCollection
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_GeomCollection
  (ageometryarray ST_Geometry
  ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_GeomCollection
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_GeomCollection
  (ageometry ST_Geometry,
  asrid INTEGER)
  RETURNS ST_GeomCollection
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,
```

```

CONSTRUCTOR METHOD ST_GeomCollection
  (ageometryarray ST_Geometry
   ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
RETURNS ST_GeomCollection
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

```

## 9.1.2 ST\_GeomCollection Methods

### 1. Rationale: Correct constructor method definitions.

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_GeomCollection
  (ageometry ST_Geometry)
RETURNS ST_GeomCollection
FOR ST_GeomCollection
RETURN SELF.ST_SRID(ageometry.ST_SRID).
  ST_Geometries (ARRAY[ageometry])

CREATE CONSTRUCTOR METHOD ST_GeomCollection
  (ageometryarray ST_Geometry ARRAY[ST_MaxGeometryArrayElements])
RETURNS ST_GeomCollection
FOR ST_GeomCollection
RETURN SELF.ST_SRID(ST_CheckSRID(ageometryarray)).
  ST_Geometries (ageometryarray)

CREATE CONSTRUCTOR METHOD ST_GeomCollection
  (ageometry ST_Geometry,
   asrid INTEGER)
RETURNS ST_GeomCollection
FOR ST_GeomCollection
RETURN SELF.ST_SRID(asrid).ST_Geometries (ARRAY[ageometry])

CREATE CONSTRUCTOR METHOD ST_GeomCollection
  (ageometryarray ST_Geometry ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
RETURNS ST_GeomCollection
FOR ST_GeomCollection
RETURN SELF.ST_SRID(asrid).ST_Geometries (ageometryarray)

```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_GeomCollection(ST\_Geometry)* returns an *ST\_GeomCollection* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_GeomCollection(ST\_Geometry ARRAY)* returns an *ST\_GeomCollection* value with:

Replace the first sentence of Description 6) with:

- 6) The null-call type preserving SQL-invoked constructor method *ST\_GeomCollection(ST\_Geometry, INTEGER)* returns an *ST\_GeomCollection* value with:

Replace the first sentence of Description 8) with:

- 8) The null-call type preserving SQL-invoked constructor method *ST\_GeomCollection(ST\_Geometry ARRAY, INTEGER)* returns an *ST\_GeomCollection* value with:

9.1.5 ST\_GeometryN Method

1. Rationale: Use a return value for such completion codes.

Replace the Definition with the following:

```
CREATE METHOD ST_GeometryN(aosition INTEGER)
  RETURNS ST_Geometry
  FOR ST_GeomCollection
  BEGIN
    IF CARDINALITY(SELF.ST_PrivateGeometries) = 0 THEN
      BEGIN
        SIGNAL SQLSTATE '01F06'
          SET MESSAGE_TEXT = 'empty geometry';
        RETURN NULL;
      END;
    END IF;
    IF aosition < 1 OR
       aosition > CARDINALITY(SELF.ST_PrivateGeometries) THEN
      BEGIN
        SIGNAL SQLSTATE '01F01'
          SET MESSAGE_TEXT = 'invalid position';
        RETURN NULL;
      END;
    END IF;
    RETURN SELF.ST_PrivateGeometries[aosition];
  END
```

In the Description section, replace Description 2) with:

2) The null-call method *ST\_GeometryN(INTEGER)*:

Case:

- a) If the cardinality of the *ST\_PrivateGeometries* attribute is equal to 0 (zero), then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – empty geometry*.
  - ii) Return the null value.
- b) If *aosition* is less than one or greater than the cardinality of the *ST\_PrivateGeometries* attribute, then:
  - i) A completion condition is raised: *SQL/MM Spatial warning – invalid position*.
  - ii) Return the null value.
- c) Otherwise, returns the element of the *ST\_PrivateGeometries* attribute at position *aosition*.

9.2.1 ST\_MultiPoint Type

1. Rationale: Correct constructor method definitions.

Replace the two <original method specification>s that begins with "METHOD ST\_MultiPoint" in the Definition with:

```
CONSTRUCTOR METHOD ST_MultiPoint
  (apointarray ST_Point
   ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_MultiPoint
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_MultiPoint
```

```

(apointarray ST_Point
  ARRAY[ST_MaxGeometryArrayElements],
  asrid INTEGER)
RETURNS ST_MultiPoint
SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

```

## 9.2.2 ST\_MultiPoint Methods

### 1. Rationale: Correct constructor method definitions.

Replace the Definition with:

```

CREATE CONSTRUCTOR METHOD ST_MultiPoint
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_MultiPoint
  FOR ST_MultiPoint
  RETURN SELF.ST_SRID(0).ST_Geometries(apointarray)

CREATE CONSTRUCTOR METHOD ST_MultiPoint
  (apointarray ST_Point ARRAY[ST_MaxGeometryArrayElements],
  asrid INTEGER)
  RETURNS ST_MultiPoint
  FOR ST_MultiPoint
  RETURN SELF.ST_SRID(asrid).ST_Geometries(apointarray)

```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_MultiPoint(ST\_Point ARRAY)* returns an *ST\_MultiPoint* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_MultiPoint(ST\_Point ARRAY, INTEGER)* returns an *ST\_MultiPoint* value with:

## 9.3.1 ST\_MultiCurve Type

### 1. Rationale: Correct constructor method definitions.

Replace the two <original method specification>s that begins with "METHOD ST\_MultiCurve" in the Definition with:

```

CONSTRUCTOR METHOD ST_MultiCurve
  (acurvearray ST_Curve
  ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_MultiCurve
  SELF AS RESULT
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_MultiCurve
  (acurvearray ST_Curve
  ARRAY[ST_MaxGeometryArrayElements],
  asrid INTEGER)
  RETURNS ST_MultiCurve
  SELF AS RESULT

```

```
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,
```

### 9.3.2 ST\_MultiCurve Methods

1. *Rationale: Correct constructor method definitions.*

Replace the Definition with:

```
CREATE CONSTRUCTOR METHOD ST_MultiCurve
  (acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_MultiCurve
  FOR ST_MultiCurve
  RETURN SELF.ST_SRID(0).ST_Geometries(acurvearray)

CREATE CONSTRUCTOR METHOD ST_MultiCurve
  (acurvearray ST_Curve ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
  RETURNS ST_MultiCurve
  FOR ST_MultiCurve
  RETURN SELF.ST_SRID(asrid).ST_Geometries(acurvearray)
```

Replace the first sentence of Description 2) with:

- 2) The null-call type preserving SQL-invoked constructor method *ST\_MultiCurve(ST\_Curve ARRAY)* returns an *ST\_MultiCurve* value with:

Replace the first sentence of Description 4) with:

- 4) The null-call type preserving SQL-invoked constructor method *ST\_MultiCurve(ST\_Curve ARRAY, INTEGER)* returns an *ST\_MultiCurve* value with:

### 9.4.1 ST\_MultiLineString Type

1. *Rationale: Correct type hierarchy description.*

Replace the Purpose Section with:

The *ST\_MultiLineString* type is a subtype of the *ST\_MultiCurve* and represents a collection of *ST\_LineString* values.

2. *Rationale: Correct constructor method definitions.*

Replace the two <original method specification>s that begins with "METHOD *ST\_MultiLineString*" in the Definition with:

```
CONSTRUCTOR METHOD ST_MultiLineString
  (alinesringarray ST_LineString
   ARRAY[ST_MaxGeometryArrayElements])
  RETURNS ST_MultiLineString
  SELF AS RESULT
LANGUAGE SQL
DETERMINISTIC
CONTAINS SQL
RETURNS NULL ON NULL INPUT,

CONSTRUCTOR METHOD ST_MultiLineString
  (alinesringarray ST_LineString
   ARRAY[ST_MaxGeometryArrayElements],
   asrid INTEGER)
```