

---

# PostGIS – Spécifications OGC et ISO

---

**Licence GNU FDL - Version 1.4**

- ❖ Le concept de Feature
- ❖ Base de données spatiales, les standards existants
- ❖ Format de représentation des données (WKT et EWKT)
- ❖ Intégrité référentielle et données spatiales



- ✦ Une 'feature' comme abstraction géographique de la réalité,
  - ✦ Géopositionnée
  - ✦ Peut être dotée d'attributs



- ❖ Permet de définir pour un SGBD existant (SQL92 ou SQL99)
  - ❖ La liste des types géométriques possibles
  - ❖ La manière de représenter les données spatiales
    - ❖ WKT: Textuel
    - ❖ WKB: Binaire
  - ❖ Les méthodes spatiales disponibles
    - ❖ Prototype
    - ❖ Comportement
  - ❖ Les contraintes d'intégrité spatiales
    - ❖ Typage géométrique
    - ❖ Système de projection



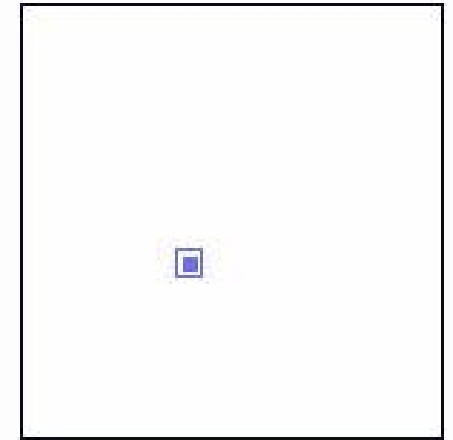
- ❖ Disponible depuis 99
- ❖ <http://www.opengeospatial.org/standards/sfs>
- ❖ Logique 2D (X Y)
- ❖ Types géométrique définis
  - ❖ POINT
  - ❖ POLYGON
  - ❖ LINESTRING
  - ❖ (+ Logique de MULTI et d'agrégat)
- ❖ Environ 80 fonctions d'accès et de manipulation des données spatiales
- ❖ Schéma d'intégrité référentielle spatiale
  - ❖ geometry\_columns
  - ❖ spatial\_ref\_sys

- ❖ Disponible depuis 06
- ❖ <http://www.opengeospatial.org/standards/sfs>
- ❖ Ajoute notamment par rapport à la 1.1:
  - ❖ Dimensions supplémentaires avec Z et M
  - ❖ Géométries 3D
    - ❖ TIN
    - ❖ POLYHEDRALSURFACE
  - ❖ Label
    - ❖ Formatage du texte
    - ❖ Positionnement



- ❖ Standard ISO
- ❖ Logique de stockage X Y [Z] [M]
- ❖ Types spatial additionel:
  - ❖ Curve
  - ❖ Pas de logique de types 3D natif
- ❖ Logique additionnelle
  - ❖ Network et routing
  - ❖ Représentation topologique
  - ❖ Fonctions géographiques
- ❖ Nombreuses fonctionnalités spatiales additionnelles à OGC SFS
- ❖ URL d'un draft ISO:
  - ❖ [http://domino.iec.ch/preview/info\\_isoiec13249-3%7Bed3.0%7Den.pdf](http://domino.iec.ch/preview/info_isoiec13249-3%7Bed3.0%7Den.pdf)

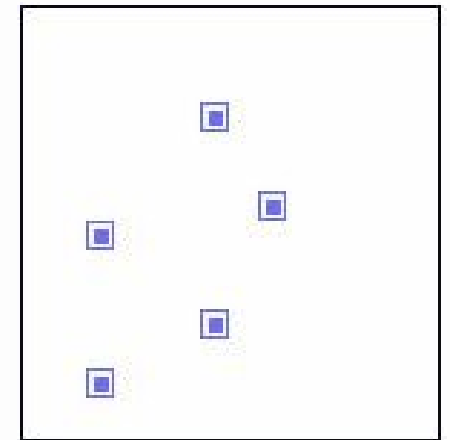
POINT (10 10)





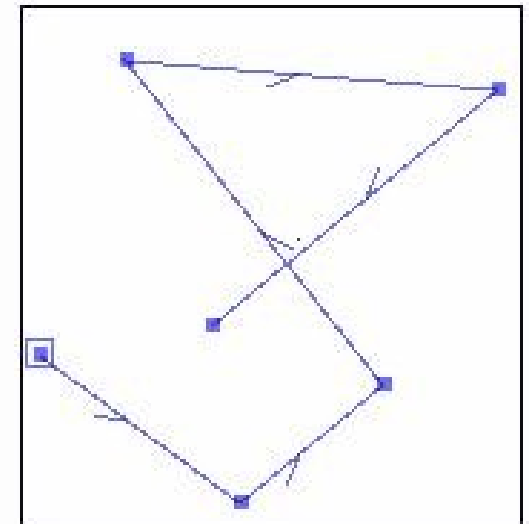
MULTIPOINT

```
(
  5 5, 5 15, 10 7, 10 30, 14 17
)
```



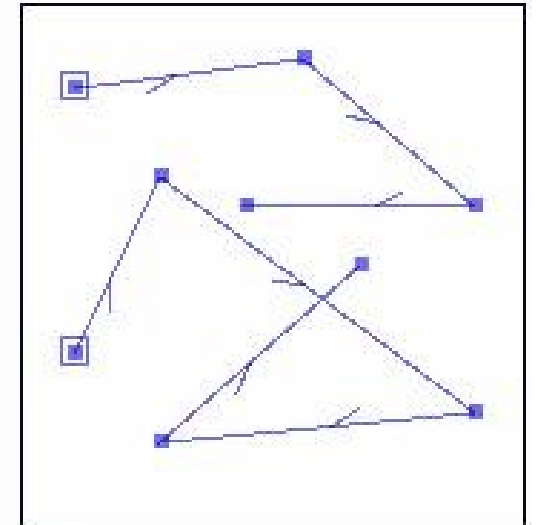
LINESTRING

```
(
  0 5, 5 1, 9 4, 2 14, 14 13, 4 4
)
```



MULTILINESTRING

```
(
  (1 5, 3 9, 14 3, 3 2, 10 8),
  (1 13, 8 13, 14 9, 7 9)
)
```

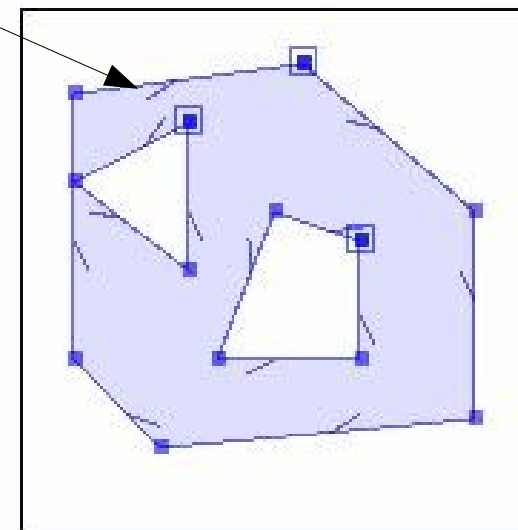


# WKT: POLYGON

POLYGON

```
(
  (9 13,13 9,13 3,4 2,1 4,1 12, 9 13),
  (5 11,5 6,1 9,5 11),
  (10 7, 10 4, 6 4, 8 8, 10 7)
)
```

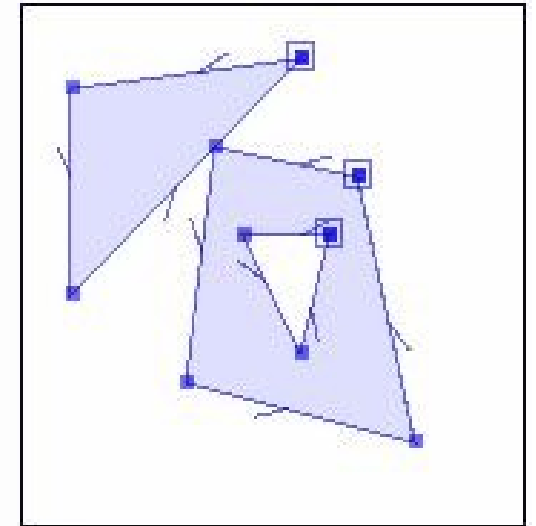
Ring externe



Coordonnées  
fermantes

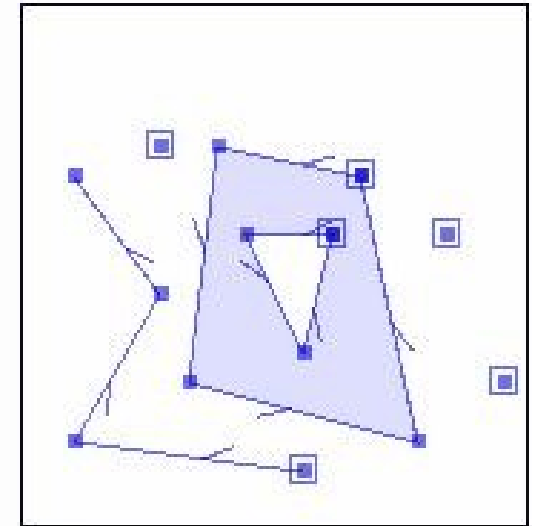
MULTIPOLYGON

```
(
  ((10 14,1 8, 1 13,10 14)),
  (
    (12 9, 13 3, 5 5, 7 10, 12 9),
    (13 7, 12 8, 11 7, 13 7)
  )
)
```



GEOMETRYCOLLECTION

```
(
  MULTIPOINT(4 10, 12 9, 14 4),
  LINESTRING(8 2, 2 3, 4 7, 2 9),
  POLYGON
  (
    (10 9, 11 3, 5 5, 6 10, 10 9),
    (9 7, 8 6, 7 7, 9 7)
  )
)
```



- ❖ Ajout fonctionnel PostGIS par rapport à SFS 1.1 :
  - ❖ Dimensions supplémentaires : 3DZ / 3DM / 4D
  - ❖ Intégration possible du SRID dans le EWKT de la géométrie
- ❖ EWKT :
  - ❖ Point 2D :
    - ❖ `POINT(0 0)`
  - ❖ Point 3DM :
    - ❖ `POINTM(0 0 0)`
  - ❖ Point 3D :
    - ❖ `POINT(0 0 0)`
  - ❖ Point 4D :
    - ❖ `POINT(0 0 0 0)`



- Les types multiples et agrégés doivent respecter:

- Même Dimension
- Même SRID

- Exemple valide:

```
GEOMETRYCOLLECTIONM  
(  
  POINTM(2 3 9),  
  LINESTRINGM((2 3 4,3 4 5, 5 7 9))  
)
```

- Exemple non valide:

```
MULTIPOINT(4 5 2,1 1, 3 6 7)
```



# Format binaire et natif

- Besoin d'un format binaire pour optimiser stockage et accès aux données:
  - Pour WKT: WKB
  - Pour EWKT: EWKB
- En natif PostGIS utilise pour le stockage
  - Un encodage Hédécimal de EWKB: HEWKB
  - Depuis la version 1.0

# Interfaces entre (E)WKT et Geometry

- Les interfaces suivantes permettent de manipuler les données WKT et EWKT :

```
Text WKT = ST_AsText(geometry);
```

```
Text EWKT = ST_AsEwkt(geometry);
```

```
Geometry = ST_GeomFromText(text WKT, SRID);
```

```
Geometry = ST_GeomFromEWKT(text EWKT);
```

# Intégrité référentielle

- Via la table **geometry\_columns**
- Permet d'établir un modèle relationnel cohérent pour les colonnes géométriques
- Définie dans la norme OGC SFS
- Fait le lien pour chaque colonne géométrique de la base:
  - Dimension (2D, 3D, 3DM, 4D)
  - Système de projection (SRID)
  - Type de géométrie (Point, Multipolygon...)

# Schéma de geometry\_columns

```
CREATE TABLE GEOMETRY_COLUMNS (  
    F_TABLE_CATALOG VARCHAR(256) NOT NULL,  
    F_TABLE_SCHEMA VARCHAR(256) NOT NULL,  
    F_TABLE_NAME VARCHAR(256) NOT NULL,  
    F_GEOMETRY_COLUMN VARCHAR(256) NOT NULL,  
    COORD_DIMENSION INTEGER NOT NULL,  
    SRID INTEGER NOT NULL,  
    TYPE VARCHAR(30) NOT NULL  
)
```

Où l'on stocke

Ce que l'on stocke

# Utilisation de geometry\_columns

- Chaque colonne géométrique de la base **devrait** être présente dans cette table
  - Utiliser les accesseurs dédiés
    - AddGeometryColumn
    - DropGeometryColumn
    - DropGeometryTable (version 1.4 ou +)
    - Populate\_Geometry\_Columns
- Possibilité d'avoir plusieurs colonnes géométriques pour la même table spatiale (surface et centroïdes par exemple)

# Intégrité référentielle: spatial\_ref\_sys

- Permet de lister l'ensemble des systèmes de projection utilisables
- Définie également dans la norme OGC SFS
- Utilisation possible de SRID=-1 pour un système de projection indéterminé
  - C'est mal ;)
  - La reprojection à la volée n'est plus possible (ST\_Transform())
- Utilisation possible de UpdateGeometrySRID()

# Schéma de spatial\_ref\_sys



```
CREATE TABLE SPATIAL_REF_SYS (  
  SRID INTEGER NOT NULL PRIMARY KEY,  
  AUTH_NAME VARCHAR(256),  
  AUTH_SRID INTEGER,  
  SRTEXT VARCHAR(2048),  
  PROJ4TEXT VARCHAR(2048)  
)
```

Identifiant du  
système de  
projection

Paramètres du  
système