

# Introduction à Oracle Spatial

L'objectif est de se familiariser avec la gestion de données spatiales dans les Bases de données objet-relationnelles. Pour ce faire, nous allons utiliser le SGBD Oracle qui offre un ensemble de types de données, de fonctions et de procédures pour la gestion (stockage, requête, mise à jour) de données spatiales groupés sous le nom d'Oracle Spatial. Notez que d'autres SGBD (ex., PostgreSQL/ PostGIS, SQL Server, MySQL ...) offrent des fonctionnalités similaires. Le choix pour Oracle s'est fait principalement pour des raisons de simplification logistique. Le contenu de ce TP est basé sur la documentation d'Oracle (Oracle® Spatial, Developer's Guide – On en est à la release 21c).

## 1. Contenu d'Oracle Spatial

Oracle Spatial offre un schéma SQL et des fonctions spécifiques permettant l'intégration de données spatiales avec les données classiques sous Oracle. Oracle Spatial (appelé plus simplement Spatial) comprend :

- ☐ Le schéma MDSYS qui définit la syntaxe et la sémantique des types géométriques pouvant être stockés et interrogés sous Oracle.
- ☐ Un mécanisme d'indexation spatiale
- ☐ Un ensemble d'opérateurs, fonctions et procédures
- ☐ D'autres fonctions utilitaires et procédures, et modèles de données (ex., données topologiques, données réseaux, données GeoRaster ...)

## 2. Types de données spatiales et métadonnées

Spatial permet de gérer des géométries simples (point, ligne, arc et polygone) et des géométries complexes telles les polygones et lignes composés de lignes droites et d'arcs. Tous ces objets spatiaux sont implémentés par un type générique Géométrie (i.e., SDO\_GEOMETRY). Les géométries peuvent être représentées en 2, 3 ou 4 dimensions, soit x, y, z et mesure (référence linéaire). La structure de données géométriques d'Oracle n'est pas topologique. Lorsque deux objets partagent la même limite (exemple lots cadastraux), il y a duplication des coordonnées. Les relations spatiales ne sont pas gérées explicitement dans la base de données mais sont déduites par traitement.

Les types de géométries sont (voir Figure 1) :

- ☐ Point et ensemble de points
- ☐ Polyligne (ou ligne brisée) composée de lignes droites
- ☐ Polygone composé de lignes droites
- ☐ Ligne composée d'arcs
- ☐ Polygone composé d'arcs
- ☐ Polygone composé de lignes droites et arcs
- ☐ Ligne composée de lignes droites et arcs
- ☐ Cercle
- ☐ Rectangle

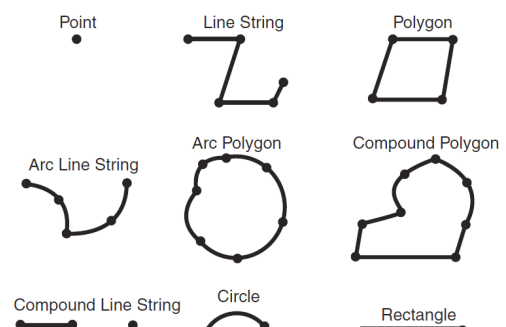


Figure 1

## SDO\_GEOMETRY

Dans Spatial, la description géométrique d'un objet spatial est stockée dans un seul tuple, dans une seule colonne de type d'objet SDO\_GEOMETRY dans une table définie par l'utilisateur. Toute table qui a une colonne de type SDO\_GEOMETRY doit avoir une autre colonne, ou un ensemble de colonnes, qui définit une clé primaire unique pour cette table.

Oracle Spatial définit le type d'objet SDO\_GEOMETRY comme :

```
CREATE TYPE sdo_geometry AS OBJECT (  
    SDO_GTYPE NUMBER,                Dimension du système et type de géométrie de l'objet  
    SDO_SRID NUMBER,                 Identifiant du système de coordonnées  
    SDO_POINT SDO_POINT_TYPE,        Coordonnées des points  
    SDO_ELEM_INFO  
    MDSYS.SDO_ELEM_INFO_ARRAY,       Description des primitives comprenant le rang de la  
                                       première coordonnée de la primitive, la dimension de  
                                       la primitive et son interprétation.  
    SDO_ORDINATES  
    MDSYS.SDO_ORDINATE_ARRAY) ;
```

## SDO\_GTYPE

L'attribut SDO\_GTYPE de SDO\_GEOMETRY indique le type de la géométrie. SDO\_GTYPE est un numéro de 4 chiffres dont le codage est indiqué dans le tableau ci-dessous :

Valeur	Type de géométrie	Description
DL00	GEOMETRIE INCONNUE	Sans Géométrie
DL01	POINT	Géométrie contenant 1 point.
DL02	LIGNE et/ou ARC	Géométrie définie par 1 ligne pouvant contenir des segments droits ou circulaires (arc) ou les deux.
DL03	POLYGONE	Géométrie définie par 1 polygone avec ou sans trou. (1)
DL04	COLLECTION	Géométrie définie par une collection d'éléments hétérogènes. (2) COLLECTION est un agrégat de tous les autres types.
DL05	MULTIPOINT	Géométrie ayant 1 ou plusieurs points. (MULTIPOINT est un agrégat de POINT.)
DL06	MULTILIGNE et/ou MULTIARC	Géométrie ayant 1 ou plusieurs lignes et/ou arcs. Agrégat de LIGNE et/ou ARC.
DL07	MULTIPOLYGONE	Géométrie ayant de multiples polygones disjoints (plus d'une limite extérieure). Agrégat de POLYGONE
DL08 /DL09	SOLID/MULTISOLID	Géométrie ayant de surfaces multiples dans un espace à trois dimensions (ex., un parallélépipède ou un tronc de cône).

- D) Correspond à la dimension de l'univers du système dans lequel est localisé l'objet: 2, 3 ou 4.
- L) Identifie la dimension où sera stockée la mesure si l'on utilise la référence linéaire: 3 ou 4.
- (1) Pour les polygones avec trou, entrez la limite extérieure en 1<sup>er</sup> suivie de la limite intérieure
- (2) Les polygones d'une collection peuvent être disjoints.

## SDO\_SRID

L'attribut SDO\_SRID est utilisé pour identifier un système de coordonnées (système de référence spatial) pouvant être associée à la géométrie. SDO\_SRID peut être NULL et dans ce cas, aucun système de coordonnées n'est associé à la géométrie. Si SDO\_SRID n'est pas nul, il doit contenir une valeur de la colonne SRID de la table SDO\_COORD\_REF\_SYS, et cette valeur doit être insérée dans la colonne SRID de la vue USER\_SDO\_GEOM\_METADATA. Toutes les géométries dans une colonne SDO\_GEOMETRY doivent avoir la même valeur de SDO\_SRID si un index spatial est créé sur cette colonne. L'attribut SDO\_SRID peut être utilisé pour transformer automatiquement des géométries d'un système de coordonnées à un autre (ex., coordonnées latitude/longitude vers coordonnées euclidiennes).

## SDO\_POINT

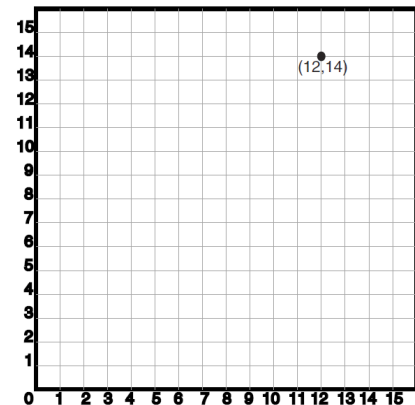
Le champ SDO\_POINT est de format SDO\_POINT\_TYPE :

```
CREATE TYPE sdo_point_type
AS OBJECT (X NUMBER, Y NUMBER, Z NUMBER) ;
```

Il permet le stockage des données ponctuelles. Il est limité à 3 dimensions.

**Exemple : Point.**

```
MDSYS.SDO_GEOMETRY (
2001,          // 2 indique la dimension et 1
indique un seul point
NULL,
MDSYS.SDO_POINT_TYPE (12, 14, NULL) ,
NULL,
NULL) ;
```



**Question 1 :** Créez une table « my\_geometries » contenant trois attributs (un identifiant, une description et une géométrie). Stockez le point défini ci-dessus dans la table.

## SDO\_ELEM\_INFO

Le champ SDO\_ELEM\_INFO est de format SDO\_ELEM\_INFO\_ARRAY :

```
CREATE TYPE SDO_ELEM_INFO_ARRAY AS VARRAY (1048576) OF NUMBER;
```

Il nous renseigne sur comment interpréter les coordonnées stockées dans le champ SDO\_ORDINATES. On y stocke un ou plusieurs triplets comprenant (1<sup>er</sup>) le rang de la coordonnée de départ de la primitive (SDO\_STARTING\_OFFSET), (2<sup>ième</sup>) le type de géométrie de la primitive (SDO\_ETYPE) et (3<sup>ième</sup>) l'interprétation de ce type (SDO\_INTERPRETATION). Les valeurs que les SDO\_ETYPE et SDO\_INTERPRETATION peuvent prendre sont indiquées dans les deux tableaux suivants.

Les types dans le Tableau 1 sont des géométries simples et généralement défini par un seul triplet sauf le cas du polygone avec trou.

**Tableau 1**

SDO_ETYPE	SDO_INTERPRETATION	Meaning
0	(toute valeur)	Utilisé pour modéliser toutes géométries non supportées par Oracle Spatial telles les courbes (autre que les arcs) et splines.
1	1	Point
	n>1	Agrégat de <i>n</i> points
2	1	Ligne dont les sommets sont reliés par des lignes droites.
	2	Ligne composée d'une séquence d'arcs définis par 3 points (début, sur et fin de l'arc). Un point servant de début et de fin d'un arc n'est pas répété.
1003 (polygone extérieur) ou 2003 (polygone intérieur)	1	Simple polygone dont les sommets sont reliés par des lignes droites. Le sommet de fin du polygone doit être le même que le sommet de début.
	2	Polygone composé d'une séquence fermée d'arcs. Le point servant de début et de fin d'un arc n'est pas répété.
	3	Rectangle défini par deux points dont l'un est localisé dans le coin inférieur gauche et l'autre dans le coin supérieur droit. Ce type peut être utilisé uniquement avec un système de coordonnées cartésien, sinon définir le rectangle avec 5 points et une interprétation égale à 1.
	4	Cercle défini par 3 points localisés sur la circonférence.

**Tableau 2**

SDO_ETYPE	SDO_INTERPRETATION	Meaning
4	n>1	Ligne complexe dont certains sommets sont reliés par des lignes droites et d'autres par des arcs. La valeur <b><i>n</i></b> du champ interprétation spécifie le nombre de sous-éléments contigus composant la ligne. Les <b><i>n</i></b> triplets suivants de la matrice SDO_ELEM_INFO décrivent chacun de ces sous-éléments. Les sous-éléments peuvent uniquement être de type SDO_ETYPE 2. Le dernier point d'un sous-élément est le premier du sous-élément suivant et <b>ne doit pas être répéter</b> .

1005 (polygone extérieur) ou 2005 (polygone intérieur)	n>1	Polygone complexe dont certains sommets sont reliés par des lignes droites et d'autres par des arcs. La valeur <b>n</b> du champ interprétation spécifie le nombre de sous-éléments contigus composant la ligne. Les <b>n</b> triplets suivants de la matrice SDO_ELEM_INFO décrivent chacun de ces sous-éléments. Les sous-éléments peuvent uniquement être de type SDO_ETYPE 2. Le dernier point d'un sous-élément est le premier du sous-élément suivant et <b>ne doit pas être répété</b> . Le point de début et de fin d'un polygone doit être le même.
--	-----	--

Les types dans le Tableau 2 sont des géométries complexes dont le premier triplet décrit le type de géométrie complexe et les triplets suivants décrivent la géométrie de chaque primitive simple. Si la ligne ou le polygone est composé de lignes droites et d'arc, on doit alors utiliser le SDO\_ETYPE 4 pour une ligne et 1005 pour un polygone extérieur et 2005 pour un polygone intérieur.

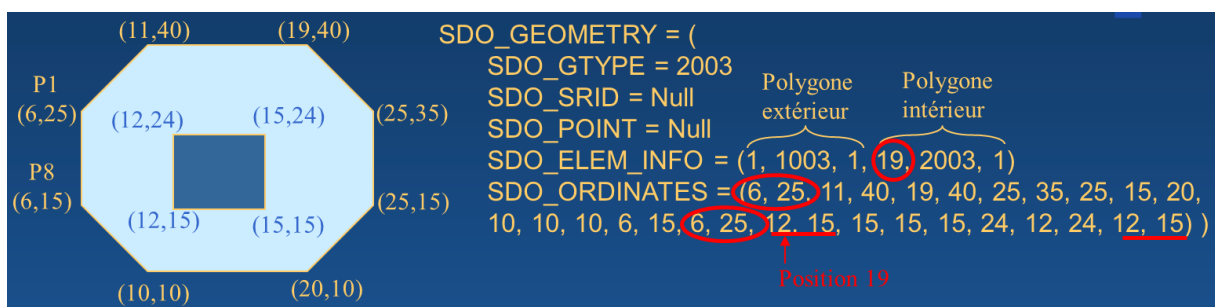
## SDO\_ORDINATES

Le champ SDO\_ORDINATES est de format SDO\_ORDINATE\_ARRAY :

```
CREATE TYPE sdo_ordinate_array AS VARRAY (1048576) OF NUMBER;
```

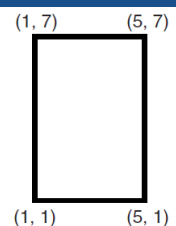
Il comprend la liste des coordonnées séparées par virgule représentant les valeurs qui constituent la frontière d'un objet spatial. Ce tableau doit toujours être utilisé en conjonction avec la valeur du type SDO\_ELEM\_INFO. Les valeurs dans le tableau sont triées par dimension. Il permet de stocker des données de 2, 3 ou 4 dimensions (mesure). SDO\_GEOM.VALIDATE\_GEOMETRY\_WITH\_CONTEXT est une fonction qui permet de valider la géométrie et d'indiquer qu'elles sont les erreurs s'il y en a.

**Exemple : Polygone avec trou.** Notez que les coordonnées du premier point sont répétées à la fin pour les deux polygones (extérieur et intérieur).



**Exemple : Rectangle.**

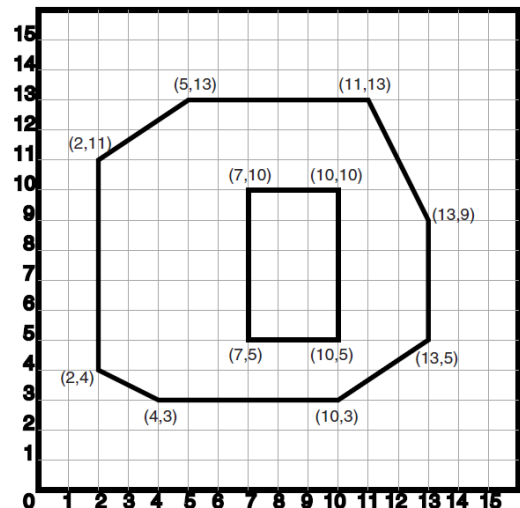
```
SDO_GEOMETRY (
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,3), -- 3 = un rectangle
  SDO_ORDINATE_ARRAY(1,1, 5,7)
  -- seulement les 2 points (bas gauche et haut droit)
```



)

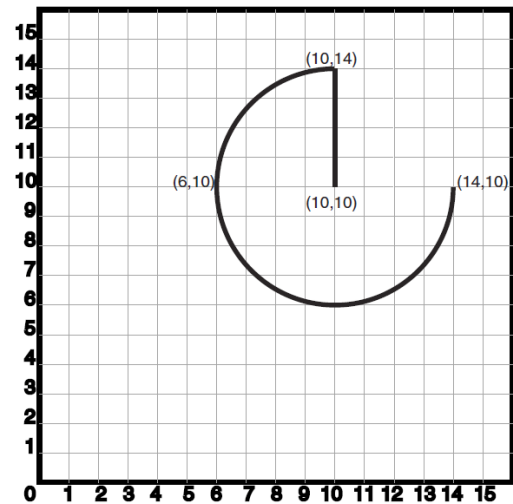
### Exemple : Polygone avec trou.

```
SDO_GEOMETRY (
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1,
19,2003,1),
    SDO_ORDINATE_ARRAY(2,4, 4,3,
10,3, 13,5, 13,9, 11,13, 5,13, 2,11,
2,4, 7,5, 7,10, 10,10, 10,5, 7,5)
)
```



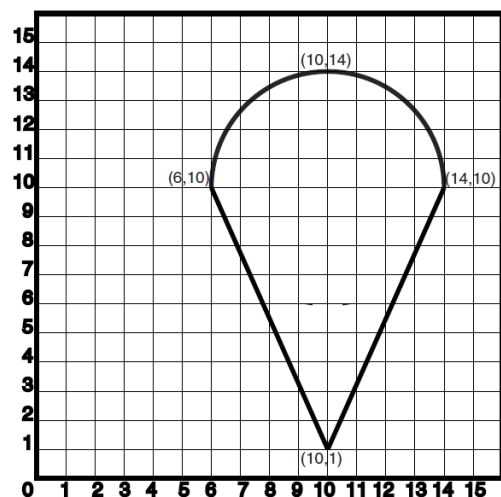
### Exemple : Ligne complexe (composée).

```
SDO_GEOMETRY (
    2002,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,4,2,
1,2,1, 3,2,2),
    SDO_ORDINATE_ARRAY(10,10,
10,14, 6,10, 14,10)
)
```



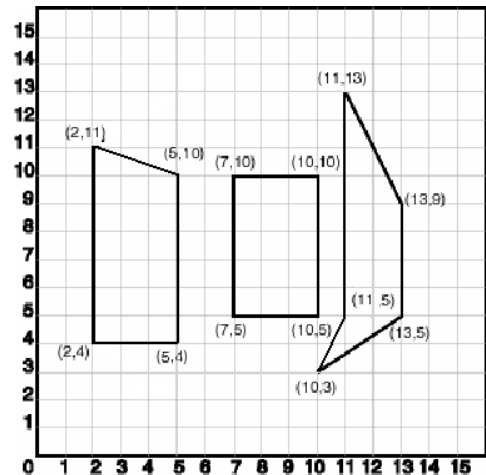
### Exemple : Polygone complexe (composée).

```
SDO_GEOMETRY (
    2003, NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1005,2,
1,2,1, 5,2,2),
    SDO_ORDINATE_ARRAY(6,10, 10,1,
14,10, 10,14, 6,10)
)
```



### Exemple : Multi-polygone.

```
SDO_GEOMETRY (
  2007,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY (1,1003,1,
    11,1003,1, 23,1003,1) ,
  SDO_ORDINATE_ARRAY (2,4, 2,11,
    5,10, 5,4, 2,4, 7,5, 7,10, 10,10,
    10,5, 7,5, 10,3, 11,5, 11,13,
    13,9, 13,5, 10,3)
)
```



**Question 2 :** Insérez toutes ces géométries dans la table « my\_geometries ».

### Table de métadonnées

Les métadonnées des géométries décrivant les dimensions, les limites inférieures et supérieures, et la tolérance dans chaque dimension sont stockées dans une table globale dans le schéma MDSYS nommée USER\_SDO\_GEOM\_METADATA. USER\_SDO\_GEOM\_METADATA contient les métadonnées pour toutes les tables spatiales appartenant au schéma utilisateur. Ceci est la seule table (une vue en réalité) que vous pouvez mettre à jour et la seule dans laquelle les utilisateurs doivent insérer les métadonnées associées aux tables spatiales.

Cette vue comprend les champs suivants:

```
TABLE_NAME VARCHAR2(32),
COLUMN_NAME VARCHAR2(32),
DIMINFO MDSYS.SDO_DIM_ARRAY,
SRID NUMBER
```

- ☐ Table name : Nom de la table ayant un champ Geometry
- ☐ Column name : Nom du champ stockant la géométrie.
- ☐ DimInfo : Type d'objets permettant le stockage des limites supérieures et inférieures de chaque dimension nécessaire au positionnement des objets de la classe.
  - CREATE TYPE SDO\_DIM\_ARRAY AS VARRAY(4) OF SDO\_DIM\_ELEMENT;
  - Le SDO\_DIM\_ELEMENT est défini comme suit :

```
CREATE TYPE SDO_DIM_ELEMENT AS OBJECT (
  SDO_DIMNAME VARCHAR2(64),
  SDO_LB NUMBER,
  SDO_UB NUMBER,
  SDO_TOLERANCE NUMBER);
```
- ☐ SRID : Identifiant du système de coordonnées dans lequel la géométrie de l'ensemble des objets de la classe est stockée.

#### Exemple :

```
INSERT INTO USER_SDO_GEOM_METADATA VALUES (  
    'my_geometries',  
    'geometry',  
    MDSYS.SDO_DIM_ARRAY(  
        MDSYS.SDO_DIM_ELEMENT('X', 0, 2000, 0.005),  
        MDSYS.SDO_DIM_ELEMENT('Y', 0, 2000, 0.005)  
    ),  
    NULL  
);
```

### 3. Indexation spatiale

Une fois les données chargées dans la base de données, un index spatial (i.e., un R-tree) doit être créé afin d'accélérer l'accès aux données (i.e., accélérer l'évaluation des opérateurs spatiaux introduits dans la section suivante). L'index permet de trouver les objets ayant une relation spatiale avec un point ou une zone de recherche ou de trouver la paire d'objets interagissant spatialement (jointure spatiale).

Un index spatial R-tree peut indexer les données spatiales jusqu'à quatre dimensions. Un index R-tree approxime chaque géométrie par un seul rectangle qui entoure la géométrie de façon optimale (appelée Minimum Bounding Rectangle ou MBR), comme le montre la Figure 2.

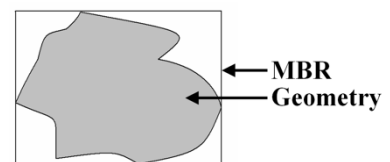


Figure 2

Pour un ensemble d'objets spatiaux donné, le R-tree consiste en un index hiérarchique sur les MBRs (voir Figure 3).

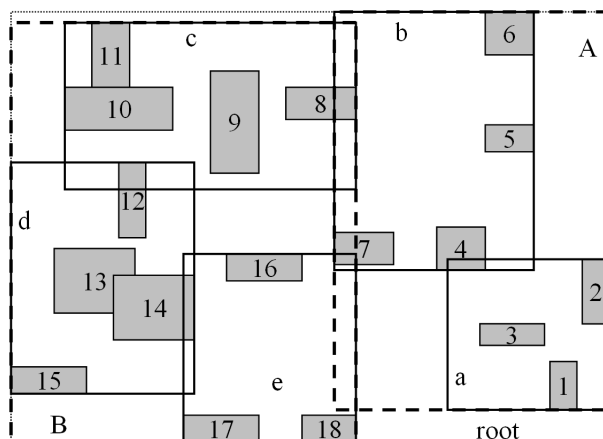
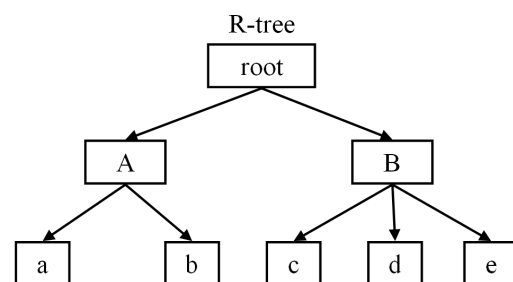


Figure 3



Création d'un index R-tree :

```
CREATE INDEX my_geometries_idx ON my_geometries(geometry)  
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Tables de métadonnées permettant de visualiser les informations des index :



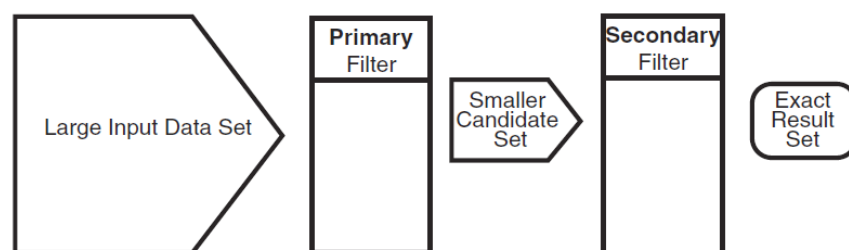
- ❑ **USER\_SDO\_INDEX\_INFO** : contient les informations de base qui ont été fournies lors de la création de l'index (nom de l'index, table contenant l'attribut sur lequel est construit l'index, nom du champ, type d'index, nom de la table stockant les valeurs de l'index, status (deferred ou valid)).
- ❑ **USER\_SDO\_INDEX\_METADATA** : contient des informations détaillées sur l'index telles que le nom de l'utilisateur, le type, le nom, la table, le nom du champ, index primaire ou secondaire, partitionné ou non, nombre de dimensions des objets, nombre de nœuds du R-tree, nombre d'enfants maximum par nœud, SDO\_LEVEL, SDO\_NUMTILES, etc.

Un nombre important d'opérations d'insertions et suppressions dans un R-tree peut dégrader la qualité de la structure du R-tree, ce qui peut affecter les performances des requêtes. Toutefois, l'index spatial sous Oracle est auto-réglable (« self-tuning »).

## 4. Opérateurs spatiaux

Les requêtes et jointures spatiales sont évaluées par un modèle de requêtes en deux phases, composé des filtres primaire et secondaire :

- ❑ Le filtre primaire compare des géométries approximatives (MBR ou tuiles de l'index) afin de réduire le temps de traitement et retourner rapidement un ensemble d'objets susceptibles de satisfaire la requête.
- ❑ Le filtre secondaire compare les géométries exactes afin de valider si elles satisfont ou non la requête. Ce traitement est complexe et coûteux en CPU mais il est appliqué uniquement sur un sous ensemble des objets de la base.



Les opérateurs spatiaux exigent que la colonne de géométrie dans le premier paramètre (geom1) soit indexée avec un index spatial. Les opérateurs spatiaux doivent être utilisés dans la clause WHERE d'une requête. Le premier paramètre de tout opérateur spécifie la colonne de géométrie à rechercher, et le second paramètre spécifie une fenêtre de requête. Si la fenêtre de requête n'a pas le même système de coordonnées que la colonne de géométrie, spatiale effectue une transformation de coordonnées de système implicite.

### SDO\_FILTER

L'opérateur SDO\_FILTER fait usage uniquement du filtre primaire :

```

SDO_FILTER (
    geom1 MDSYS.SDO_GEOMETRY,
    geom2 MDSYS.SDO_GEOMETRY,
    params VARCHAR2
)
  
```

où « geom1 » et « geom2 » sont les géométries des deux tables impliquées dans la relation.

« params » indique le type de filtre :

- ☐ WINDOW : s'il n'y a qu'une seule géométrie dans la table de geom2. L'ensemble des geom1 seront comparées à l'unique geom2.
- ☐ JOIN : si geom1 et geom2 ont toutes les deux plusieurs enregistrements.

## Evaluation exacte des prédicats spatiaux

La plus part des analyses spatiales requièrent l'usage du filtre secondaire afin de sélectionner avec exactitude les objets qui satisfont la requête. Spatial comprend les opérateurs spatiaux suivants :

- ☐ SDO\_RELATE : opérateur topologique.
- ☐ SDO\_WITHIN\_DISTANCE : opérateur métrique qui détermine si deux objets sont situés à l'intérieur d'une distance spécifique l'un de l'autre.
- ☐ SDO\_NN : opérateur métrique qui permet d'identifier le plus proche voisin d'un objet.

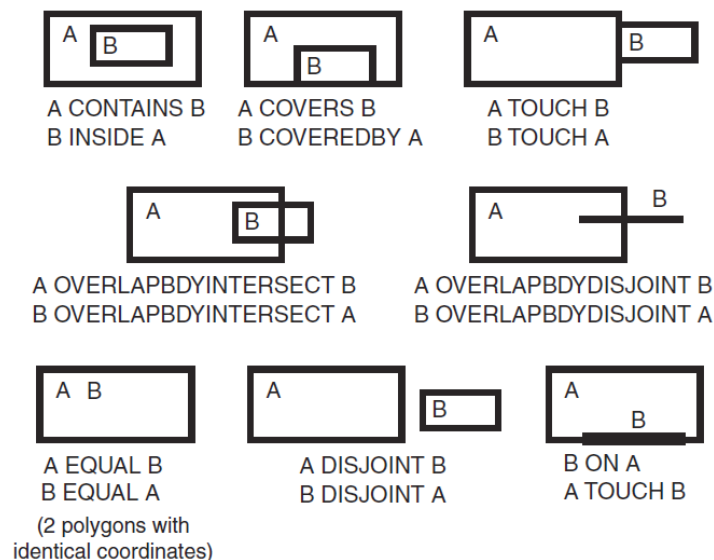


Figure 4 : Relations topologiques

## SDO\_RELATE

La matrice à 9 intersections est utilisée pour catégoriser les relations spatiales entre les points, lignes et polygones (voir Figure 4) :

- ☐ DISJOINT: Les limites et les intérieurs ne s'intersectent pas.
- ☐ TOUCH: Les limites d'intersectent mais pas les intérieurs.
- ☐ OVERLAPBDYDISJOINT: L'intérieur d'un objet intersecte la limite et l'intérieur d'un autre objet.
- ☐ OVERLAPBDYINTERSECT: Les limites et les intérieurs des objets s'intersectent.
- ☐ EQUAL: Les objets ont le même intérieur et la même limite.
- ☐ CONTAINS: L'intérieur et la limite d'un objet sont inclus entièrement dans l'intérieur d'un autre objet.
- ☐ COVERS: L'intérieur d'un objet est complètement inclus dans l'intérieur d'un autre objet et leurs limites s'intersectent

- ☐ **INSIDE:** L'opposé de CONTAINS. A INSIDE B implique que B CONTAINS A.
- ☐ **COVEREDBY:** L'opposé de COVERS. A COVEREDBY B implique que B COVERS A.
- ☐ **ON:** L'intérieur et la limite d'un objet est sur la limite d'un autre objet.
- ☐ **ANYINTERACT:** Les objets sont non disjoints.

`SDO_RELATE(geometry1, geometry2, params);`

**params :** précise le comportement de l'opérateur.

- ☐ **mask :** spécifie la relation topologique a évalué (touch, covers, contains, ..., anyinteract)
- ☐ **querytype:** WINDOW (si la géométrie2 a un seul objet) ou JOIN (si géométrie2 comprend plusieurs objets).

**Exemples d'utilisation de SDO\_RELATE** (notez que les masks peuvent être combinés) :

```
SELECT A.id
FROM my_geometries A, query_geom B
WHERE B.gid = 1
      AND SDO_RELATE(A.geometry, B.geometry,
                     'mask=INSIDE+COVEREDBY querytype=WINDOW') = 'TRUE';
SELECT A.id
FROM my_geometries A, query_geom B
WHERE SDO_RELATE(A.geometry, B.geometry,
                 'mask=ANYINTERACT querytype=JOIN') = 'TRUE';
```

## SDO\_WITHIN\_DISTANCE

`SDO_WITHIN_DISTANCE (geometry1, aGeom, params);`

- ☐ **geometry1:** Géométries à évaluer
- ☐ **aGeom :** Géométrie qui servira à évaluer si des geometry1 se trouvent à une certaine distance de celle-ci.
- ☐ **params :** distance (permet de spécifier la distance), querytype (indique si uniquement le filtre primaire sera lancé ou bien les deux. Par défaut, les deux filtres sont exécutés. Pour exécuter que le filtre primaire, mettre 'querytype = FILTER'.), unit (permet de spécifier l'unité de mesure. Par défaut, c'est le mètre.), max\_resolution et min\_resolution (voir la documentation).

**Exemple :** sélectionner les IDs des géométries situées à l'intérieur d'une distance de 10 unités du rectangle défini.

```
SELECT A.id
FROM my_geometries A
WHERE SDO_WITHIN_DISTANCE(A.geometry,
                          mdsys.sdo_geometry(2003,NULL,NULL,
                                              mdsys.sdo_elem_info_array(1,1003,3),
                                              mdsys.sdo_ordinate_array(x1,y1,x2,y2)),
                          'distance = 10') = 'TRUE';
```

## SDO\_NN

`SDO_NN(geometry1, geometry2, param [, number]);`

Cet opérateur retourne le nombre d'objet (indiqué dans param) de la géométrie1 les plus près de la géométrie2. Voir la documentation pour le détail des paramètres de SDO\_NN.

## 5. Autres fonctions géométriques

### Méthodes de l'objet SDO\_GEOMETRY

- ☐ `get_gtype()` : retourne la dimension de l'objet
- ☐ `get_LRS_dim()` : retourne la valeur de mesure de la géométrie de l'objet

### Fonctions géométriques

- ☐ Relation (vrai/faux) entre deux objets : `Relate`, `Within_distance`, `Validation`, `Validate_geometry`, `Validate_layer`
- ☐ Opération sur deux objets : `Sdo_distance`, `Sdo_difference`, `Sdo_intersection`, `Sdo_union`, `Sdo_xor`
- ☐ Opération sur un seul objet : `Sdo_arc_densify`, `Sdo_area`, `Sdo_buffer`, `Sdo_centroid`, `Sdo_convexhull`, `Sdo_length`, `Sdo_MBR`, `Sdo_pointonsurface`

### Fonctions de transformation de système de coordonnées

- ☐ `SDO_CS.TRANSFORM` : transforme la représentation d'une géométrie d'un système de coordonnées vers un autre.
- ☐ `SDO_CS.TRANSFORM_LAYER` : transforme la représentation géométrique de l'ensemble des objets de la table d'un système de coordonnées vers un autre.

### Fonctions d'agrégation spatiale

- ☐ `SDO_AGGR_CENTROID` : retourne le centroïde (centre de gravité) des géométries spécifiées.
- ☐ `SDO_AGGR_CONVEXHULL` : retourne une géométrie polygonale approximative incluant l'ensemble des géométries spécifiées.
- ☐ `SDO_AGGR_LRS_CONCAT` : agrège les segments pour former un agrégat.
- ☐ `SDO_AGGR_MBR` : retourne le REM des géométries spécifiées.
- ☐ `SDO_AGGR_MBR SDO_AGGR_UNION` : retourne la géométrie résultante de l'union des géométries spécifiées.