

Ora2Pg

Migration d'une base Oracle Spatial

Type Spatial Oracle

Ce que Ora2Pg connaît du Spatial Oracle :

- Type SDO_GEOMETRY

```
CREATE TABLE cola_markets (  
    mkt_id NUMBER PRIMARY KEY,  
    name VARCHAR2(32),  
    shape SDO_GEOMETRY  
);
```

Détail du type Oracle

Type SDO_GEOMETRY

Nom	Description
-----	-----
SDO_GTYPE	Type de géométrie
SDO_SRID	Système de coordonnées
SDO_POINT	Simple point
SDO_ELEM_INFO	Tableau d'informations
SDO_ORDINATES	Tableau de coordonnées

Type Spatial PostGis

Type équivalent dans PostGis : GEOMETRY

```
CREATE TABLE cola_markets (  
    mkt_id bigint PRIMARY KEY,  
    name varchar(32),  
    shape geometry(GEOMETRY)  
);
```

Contraintes de types

Pour éviter d'insérer n'importe quel objet géométrique il est possible dans PostGis de contraindre le type

```
CREATE TABLE stores (  
    id integer,  
    geom1 geometry(POINT, 4326),  
    geom2 geometry(POLYGONZ, 4326)  
);
```

Géométrie par défaut

Il est possible de mélanger des types (points/lignes/polygones...) dans une même colonne.

Correspond à l'utilisation du type générique GEOMETRY.

C'est le type par défaut utilisé par Ora2Pg.

Détecter la géométrie

Détection automatique du type géométrique d'une colonne :

- une seule solution => le parcours de la table
- AUTODETECT_SPATIAL_TYPE 1

Performances abominables si pas de limite de lecture :

```
SELECT DISTINCT c.SDO_GTYPE  
FROM MYTABLE c WHERE ROWNUM < ?;
```

ROWNUM=50000 par défaut, sauf si AUTODETECT_SPATIAL_TYPE > 1

Le SRID

SRID : système de coordonnées d'une colonne

Oracle "legacy" vs standard de fait "EPSG"

- CONVERT_SRID 1

Fonction de conversion **map_oracle_srid_to_epsg()**, mais retourne souvent NULL

- DEFAULT_SRID 4326

Forcer l'utilisation d'un SRID particulier

- CONVERT_SRID 27572

Lecture du SRID

Le SRID d'une colonne se trouve à la fois :

- dans les métadonnées SRID de ALL_SDO_GEOM_METADATA
- et la colonne SDO_SRID de chaque SDO_GEOMETRY

Requête d'extraction du SRID :

```
SELECT COALESCE(  
    sdo_cs.map_oracle_srid_to_epsg(SRID), 4326 )  
FROM ALL_SDO_GEOM_METADATA  
WHERE TABLE_NAME=? AND COLUMN_NAME=? AND  
OWNER=?;
```

Lecture de la dimension

La colonne DIMINFO de ALL_SDO_GEOM_METADATA est un tableau

- VARRAY(4) OF MDSYS.SDO_DIM_ELEMENT.

Chaque élément correspondant à une dimension, il suffit de regarder la taille effective de ce VARRAY pour avoir la dimension commune des géométries.

```
SELECT SDO_DIMNAME, SDO_LB, SDO_UB FROM TABLE (  
    SELECT DIMINFO a  
    FROM ALL_SDO_GEOM_METADATA  
    WHERE TABLE_NAME=? AND COLUMN_NAME=? AND OWNER=?  
);
```

Types et dimensions

Oracle : SDO_GTYPE => ex : 2003

Le 2 indique la dimension et le 3 indique le type (ici un polygone).

PostGis : En fonction de la dimension ajout de Z (3D) ou ZM (4D) au type géométrique:

GEOMETRY / GEOMETRYZ / GEOMETRYZM

POINT / POINTZ / POINTZM

POLYGON / POLYGONZ / POLYGONZM

...

Les données : Oracle

Insertion d'un simple rectangle dans Oracle :

```
INSERT INTO cola_markets VALUES (  
  302, 'Rectangle',  
  SDO_GEOMETRY(  
    2003, -- polygone a deux dimensions  
    NULL,  
    NULL,  
    SDO_ELEM_INFO_ARRAY(1,1003,3), -- un rectangle  
    SDO_ORDINATE_ARRAY(1,1, 5,7) -- 2 points pour définir le  
rectangle  
  )  
);
```

Les données : PostGis

Insertion du même rectangle dans PostgreSQL :

```
INSERT INTO cola_markets (mkt_id,name,shape) VALUES (  
    302,  
    E'rectangle',  
    ST_GeomFromText(  
        'POLYGON ((1.0 1.0, 5.0 1.0, 5.0 7.0, 1.0 7.0, 1.0 1.0))',  
        4326  
    )  
);
```

Export des données

```
$srid = 'SELECT COALESCE(sdo_cs.map_oracle_srid_to_epsg(SRID),  
$default_srid) FROM ALL_SDO_GEOM_METADATA WHERE TABLE_NAME=?  
AND COLUMN_NAME=? AND OWNER=?'
```

Extraction des données par Ora2Pg en mode INSERT :

```
'ST_GeomFromText('||SDO_UTIL.TO_WKTGEOMETRY(geomcol)||','||  
$srid||)'
```

Extraction des données par Ora2Pg en mode COPY :

```
'SRID=' || $srid || ';' || SDO_UTIL.TO_WKTGEOMETRY(geomcol)
```

Exemple export de données

Mode COPY :

```
COPY cola_markets (mkt_id,name,shape) FROM STDIN;
```

```
301    polygon SRID=4326;POLYGON ((5.0 1.0, 8.0 1.0, 8.0  
6.0, 5.0 7.0, 5.0 1.0))
```

```
\.
```

Mode INSERT :

```
INSERT INTO cola_markets (mkt_id,name,shape) VALUES  
(301,E'polygon',ST_GeomFromText('POLYGON ((5.0 1.0, 8.0  
1.0, 8.0 6.0, 5.0 7.0, 5.0 1.0))',4326));
```

Performances

SDO_UTIL.TO_WKTGEOMETRY est une procédure Java qui explose en vol s'il y a trop de lignes (tout au moins jusqu'en 11g) :

- DBD::Oracle::st fetchall_arrayref failed: ORA-29532: Java call terminated by uncaught Java exception: java.lang.OutOfMemoryError

Ne supporte pas la 3D ? Ne semble plus être le cas en 12c mais reste à vérifier.

SpatialDB Advisor

- Extension Java de Simon Greener (<http://spatialdbadvisor.com/>)

Quelques fonctions spécifiques du package SC40 :

- ST_GeomFromText
- ST_GeomFromEWKT
- ST_AsText
- ST_AsEWKT

Utilisation du package SC40

Forcer l'utilisation du package SC40 :

- USE_SC40_PACKAGE 1
- USE_SC40_PACKAGE SCHEMANAME

Extraction des données par Ora2Pg en mode INSERT :

```
'ST_GeomFromText(''||SC40.ST_AsText(geomcol)||','||$srid||')
```

Extraction des données par Ora2Pg en mode COPY :

```
'SRID=' || $srid || ';' || SC40.ST_AsText(geomcol)
```

Index spatiaux

Index spatial Oracle

```
CREATE INDEX cola_spatial_idx  
  ON cola_markets(shape)  
  INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Index spatial PostgreSQL

```
CREATE INDEX cola_spatial_idx  
  ON cola_markets USING gist(shape);
```

Évolutions

Transformation des fonctions géométriques :

- SDO_UTIL.TO_GML311GEOMETRY -> ST_AsGML
- SDO_RELATE -> ST_Relate
- geomcol.SDO_SRID -> St_Srid(geomcol)
- geomcol.SDO_POINT.X -> St_X(geomcol)
- sdo_contains(COL1, COL2) -> St_Contains(COL1, COL2)

Utilisation du format binaire WKB

Export des types SDO_GEORASTER et SDO_RASTER ?

Autres outils

JTS Topology Suite

- <http://tsusiatsoftware.net/jts/main.html>

ogr2ogr

- <http://www.gdal.org/ogr2ogr.html>

oracle_fdw

- https://github.com/laurenz/oracle_fdw/tree/postgis

Remerciements

Evolution demandée et spécifiée par Vincent Picavet (OSLANDIA).

Travaux réalisés avec l'aide de Dominique Legendre (BRGM).

Travaux financés par Dalibo => 20 % de R&D

Questions ?

Vos retours et suggestions !

Projet:

- <http://ora2pg.darold.net/>
- <http://github.com/darold/ora2pg>
- <http://sourceforge.net/projects/ora2pg/>
- gilles [at] darold [dot] net

Support commercial : Dalibo