

# ADATTÍPUSOK, METAADATOK

## Adatszerkezet

A Spatial tartalmaz egy objektum adattípus-halmazt, ezeket az objektum adattípusokat felhasználó operátorokat, függvényeket és eljárásokat. Egy objektumot egy egyszerű sorban tudunk tárolni, az oszlop típusa pedig *SDO\_GEOMETRY*, melyet az alábbiak szerint definiáltak:

```
CREATE TYPE sdo_geometry AS OBJECT (  
  SDO_GTYPE NUMBER,  
  SDO_SRID NUMBER,  
  SDO_POINT SDO_POINT_TYPE,  
  SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,  
  SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

Az egyes részek jelentése:

*SDO\_GTYPE*: az alakzat típusát adja meg. Egy négyjegyű szám, melynek formátuma *dltt*. Az első számjegy (d) adja meg a dimenziószámot, ez lehet 2, 3 vagy 4. Magasabb dimenziók esetén használhatunk lineáris viszonyítási rendszert (LRS), a második számjegy (l) adja meg, hogy melyik dimenzióhoz rendeltük az ehhez használt mértéket. Ha nem használunk LRS-t vagy alapértelmezetten az utolsó dimenziót akarjuk megadni, akkor ez az érték 0. Az utolsó két számjegy (tt) az alakzatot írja le, ez jelenleg 00-tól 07-ig használt, a többi érték későbbi felhasználásra van.

Példa: *SDO\_GTYPE* = 2003, tehát egy kétdimenziós alakzatról beszélünk (2), ahol nem használhatunk LRS-t (0), a 03-as alakzat pedig egy poligon.

Fontos tudnunk, hogy az itt megadott dimenzióadat meghatározza például azt is, hogy hány adat ad meg egy pontot vagy egy egyenest, illetve hogy egy rétegen belül nem keverhetjük a különböző dimenziószámú elemeket.

*SDO\_SRID*: a koordinátarendszert adja meg. Ha értéke NULL, akkor nem definiálunk koordinátarendszert. Ha nem NULL, akkor egy létező értéknek kell lennie az *SDO\_COORD\_REF\_SYS* táblából, továbbá ennek a *SRID* értéknek szerepelnie kell a metaadatok között is. Egy oszlopban lévő alakzatoknak azonos *SRID* értéket kell adnunk.

*SDO\_POINT*: összetett adatszerkezet, az alábbi típusdefinícióval:

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

Egy darab háromdimenziós pontot tudunk vele definiálni. Ha az utolsó két elem az *SDO\_GEOMETRY*-ben NULL, akkor ez a mező nem NULL, X és Y értéke egy pont koordinátáinak felel meg. Ha az említett mezők értéke nem NULL, akkor a Spatial ezt a mezőt figyelmen kívül hagyja. Pont típusú alakzatot tárolhatnánk a másik két mezőben is, azonban csak pont típusú alakzat esetén ez javallott.

*SDO\_ELEM\_INFO*: összetett adatszerkezet az alábbi típussal:

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

Tehát egy változó hosszúságú tömbről van szó, melyben megtudhatjuk, hogy az *SDO\_ORDINATES* tömbben tárolt adatokat hogyan kell értelmezni. Minden egyes, az objektumunkban tárolt alakzathoz egy számhármas tartozik, amely az alábbi formátumú:

```
(SDO_STARTING_OFFSET, SDO_ETYPE, SDO_INTERPRETATION)
```

Az első mutatja meg, hogy az alakzathoz tartozó koordinátaértékek hol kezdődnek az *SDO\_ORDINATES* tömbben, a számozás 1-gyel indul. *SDO\_ETYPE* az alakzat típusát adja meg. Ez a típus lehet egyszerű vagy összetett, de bizonyos alakzatoknál más információt is hordozhat ez a mező. Például megadható az is egy poligon esetén, hogy az egyes pontokat az óramutató járásával megegyezően vagy azzal ellentétesen sorolunk-e fel. Az utolsó szám két dolgot jelenthet. Összetett alakzat esetén azt mutatja meg, hogy hány darabból épül fel az alakzat, míg egyszerű esetben azt tudhatjuk meg, hogy miként értelmezzük a megadott pontkoordinátákat (például hogy a megadott ponthalmazt, mint törtvonalat, vagy pedig mint körívekből álló alakzatot kell értelmezni).

*SDO\_ORDINATES*: ez a mező tartalmazza az alakzatot felépítő pontok koordinátáit. Attól függően, hogy hány dimenziós alakzatot írunk le, változik az egy ponthoz tartozó elemek száma, tehát például háromdimenziós esetben egy ponthoz három érték tartozik. Az egy ponthoz tartozó koordinátaértékeket rendezetten tároljuk, a koordinátáknak mindig ugyanaz a sorrendje.

Alapértelmezésben a Spatial nem ellenőrzi, hogy a *GTYPE*-ban megadott alakzat típusnak megfelelnek-e az itt megadott koordináták, ezt nekünk kell figyelemmel kísérnünk. Erre a célra létezik egy előre megírt könyvtári függvény is, ez az *SDO\_GEOM\_VALIDATE\_GEOMETRY\_WITH\_CONTEXT* függvénye.

## Metódusok

Az *SDO\_GEOMETRY*-hez definiáltak alapértelmezett metódusokat, melyekkel az adatszerkezet különböző részeit lehet lekérdezni. Létezik ilyen metódus a dimenziószám, a *GTYPE* típus és az LRS adatainak lekérdezésére, továbbá annak eldöntésére, hogy a megadott alakzatunk érvényes-e. Két további metódus segítségével pedig (*GET\_WKB*, *GET\_WKT*) bináris illetve szöveges formátumba alakíthatjuk át az objektumunkat, ezeket az alakokat a konstruktorok tudják felhasználni. Ilyenkor az objektum minden adata átkerül az új formátumba, viszont a koordinátarendszer információi elvesznek.

## Konstruktorok

Konstruktorok segítségével geometriai alakzatot hozhatunk létre az előző részben már említett bináris illetve szöveges formátumból. Mivel ezek a formátumok nem tartalmaznak koordináta-rendszerrel kapcsolatos információkat, ezért ezt nekünk kell beállítanunk, ügyelve arra, hogy ez az adat illeszkedjen a már korábban definiált alakzatok koordináta-rendszeréhez.

## Metaadatok

A geometriai metaadatok a dimenziók jellemzésére használatosak, minden egyes dimenzióhoz tárolhatjuk a magasabb és alacsonyabb szintek felé fennálló kötöttségeket illetve az adott dimenzióhoz tartozó tűréshatárt is. Ezek az adatok egy külön táblában tárolódnak, melynek tulajdonosa *MDSYS*. Minden Spatial felhasználó két nézetet láthat ebből a táblából, ezek: *USER\_SDO\_GEOM\_METADATA* és *ALL\_SDO\_GEOM\_METADATA*. Az előbbi tartalmazza azokat a metaadatokat, melyek a felhasználó saját tábláihoz kapcsolódnak. Ebben a nézetben lehetőségünk van módosítani és beszűrni is. A második nézet minden táblához tárolja a metaadatokat, ez a nézet csak lekérdezésekhez használható.

Egy metaadat az alábbi definícióval rendelkezik:

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

A tábla és az oszlop neve adja meg, hogy hol található a térbeli / geometriai információ. A *DIMINFO* egy változó hosszúságú (max. 4) tömb, ahol minden dimenzióhoz egy adatszerkezetet tárolunk, egy ilyen adatszerkezet a dimenzióra jellemző értékeket tárolja. *SRID* mezőben az adatokhoz rendelt koordináta-rendszer típusát találhatjuk, ha értéke *NULL*, akkor nincs meghatározva koordináta-rendszer. A mindenki által látható metaadatok esetén a definíció egy *OWNER* mezővel egészül ki, itt a táblát birtokló felhasználó található.

A Spatial által használt indexelési eljárásnak is szüksége van metaadatokra, ezeket szintén egy közös táblában tárolja az Oracle. A felhasználók itt is két nézetben láthatják az adatokat, az egyikben a saját tábláikhoz tartozó indexeiket módosíthatják, míg a másikban minden felhasználó adata látható, de ez csak lekérdezhető.

# SPATIAL TOVÁBBI SAJÁTOSSÁGAI

## Geokódolás

Geokódolásnak nevezzük azt az eljárást, amikor megfeleltetést létesítünk földrajzi / térbeli pontok (például hosszúsági és szélességi koordinátaival adott pontok) és postai címek között. Egy postai cím lehet formázott vagy formázatlan. Formázott esetben a cím egy attribútumhalmaz, melyben olyan elemek találhatóak, mint utcanév, házszám, ország, település, irányítószám, stb. Formázatlan esetben a cím egy sztring, amelynek formátuma nem kötött, lehetnek benne olyan elemek is, melyek a geokódolás számára feleslegesek.

Ilyen alkalmazások során gyakran előfordul, hogy nem tudunk pontos címekkel dolgozni. Ezért vezetnek be illeszkedési módokat, amik azt határozzák meg, hogy a megadott (és nem feltétlenül pontos) címnek mennyire kell illeszkednie a geokódolt címmel. Ez meglehetősen széles határok között változhat, kezdve a teljes illeszkedéstől egészen odáig, hogy már csak azt követeljük meg, hogy a két helyszín azonos egy közigazgatási területen belül legyen. Alapértelmezett esetben az utcanévben engedünk csak meg eltéréseket.

Geokódoláshoz egy új adattípust is definiáltak, ez az *SDO\_GEO\_ADDR* típus. Ilyen típusú változóban tudunk formázott címet eltárolni, illetve olyan eljárások, melyek kimenete valamilye címinformáció, szintén ilyen típust használnak. Gyakorlatilag minden lehetséges címattribútumot tartalmaz, ezek közül természetesen nem használjuk mindig az összeset. Kiegészítették még olyan mezőkkel, melyek az illeszkedési módot és az esetleges hibáüzenetet tárolják (eljárások kimenete számára). Az utolsó két mező pedig a geokódolás eredménye, vagyis egy hosszúsági és szélességi koordináta érték.

Geokódolás segítségével többféle konverziót is végezhetünk adatainkon, léteznek függvények melyekkel formázott vagy formázatlan címeket tudunk kódolni, van lehetőség az inverz műveletre, vagyis amikor földrajzi adatokból kapunk címinformációt, továbbá a kimeneti típus is változtatható, ha szükséges, szabványos geometriai típus is lehet egy kódolás eredménye.

A geokódoláshoz szükséges adatokat többféle táblából nyerhetjük ki. Létezik két olyan tábla, amelyet az Oracle biztosít számunkra, ezek címformátumokat tartalmaznak. Vannak továbbá olyan táblák is, melyek országspecifikusak, itt olyan adatok tárolódhatnak, mint irányítószámok, utcanévek, helységnevek, kereszteződések, stb.

## Koordinátarendszerek

Koordinátarendszernek vagy térbeli viszonyítási rendszernek nevezzünk egy olyan szabályrendszert, mely koordináták halmazát rendeli egy térbeli ponthoz. Többféle ilyen rendszer használatos, a legismertebbek:

- Descartes koordinátarendszer: egy meghatározott pontból (az origóból) kiinduló két (vagy három) merőleges egyenes által megadott viszonyítási alap
- geográfiai koordináták: tulajdonképpen a szélességi és hosszúsági adatok, melyek nagyban hasonlítanak a matematikai polárkoordinátákra

- projektív koordináták: síkbeli Descartes-féle koordináta-rendszer, amit úgy kaphatunk, hogy egy földfelszíni pontból a matematikai vetítés műveletével leképezzük a koordináta-rendszert egy síkra
- helyi koordináták: olyan Descartes-féle koordináta-rendszer, amely független a Földtől

Ezek között létezik egy egyértelmű megfeleltetés, tehát egyik koordináta-rendszerből egyértelműen át tudunk térni egy másikra.

A Spatial-ben lehetőségünk van tetszőleges koordináta-rendszerben értelmezni az adatainkat, a különböző megvalósítások közötti átjárás teljes körűen biztosított.

A koordináta-rendszer választás tehát teljesen ránk van bízva, azonban van néhány olyan tényező, mely igen fontos lehet a választás során. Az egyik ilyen a kezelt terület mérete, mivel földrajzi koordinátáink nagy területen is pontos eredményt adnak, ha viszont Descartes koordinátákat használunk a Földön nagy távolságok mérésére, akkor megehetősen nagy hibát is okozhatunk. Másrészt viszont Descartes koordinátákkal sokkal könnyebb, ezáltal gyorsabb is számolni, így ha fontos a gyorsaság és némi pontatlanság megengedett, akkor célszerűbb Descartes koordinátákat használni földrajzi hosszúsági és szélességi adatok helyett.

## ÚJDONSÁGOK

Spatial és Locator újdonságok Oracle 10g-ben

### Locator:

- European Petroleum Survey Group (EPSG) adatmodelljének támogatása
- explicit koordinátatranszformációk
- egyszerűsített operátorok
- párhuzamos lekérdezések

### Spatial:

- topológiai és hálózati adatmodell fejlesztése
- geokódoló motor
- GeoRaser tömörítési algoritmus
- útvonaltervező bővítése Nyugat-Európával
- térinformatikai analitikus függvények

## Spatial és Locator újdonságok Oracle 11g-ben

### Locator:

- puffer függvények
- SQL/MM térbeli függvények és típusok támogatása

### Spatial:

- új 3D Java adatmodell
- GeoSpatial Web Services
- GeoRaster fejlesztése (több fájlformátum, adat- és metaadattípus, megbízhatóság)
- hálózati adatmodell fejlesztése
- útvonaltervező motor fejlesztése
- SQL/MM térbeli függvények és típusok támogatása

### **Forrás:**

Spatial User's Guide

ORACLE SPATIAL 11g – Advanced Spatial Data Management for the Enterprise – Oracle Data Sheet

ORACLE LOCATOR ÉS ORACLE SPATIAL OPCÍÓ – Térinformatika az Oracle Database 10g Release 2-ben – Oracle Adatlap

ORACLE LOCATOR – Location Features in Oracle Database 11g – Oracle Feature Overview