

Oracle Spatial

Az Oracle adatbázis-kezelő rendszer **Oracle Spatial** (Oracle Locator) nevű kiegészítő modulja támogatja a térbeli adatok kezelését.

Térbeli adatot tartalmazó tábla: Geometry table

Legalább 2 oszlopa van:

- Elsődleges kulcs,
- SDO_GEOMETRY típusú oszlop.

Megjegyzések: SDO =Spatial Data Option

3D és 4D koordináták is megengedettek, de a függvények többsége csak 2D-re működik.

SDO_GEOMETRY definíciója

```
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
);
```

SDO_GEOMETRY (folytatás)

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

CREATE TYPE sdo_elem_info_array AS
  VARRAY (1048576) of NUMBER;

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

SDO_GTYPE

Alakzat típusa: 4 jegy `d1tt` formátumban:

- `d` = dimenzió (2, 3, 4)
- `l` = 0 ha nem LRS alakzat. (LRS = linear referencing system = lineáris címzési rendszer. `l` = 3 vagy 4 adja meg, hogy melyik dimenzió tartalmazza a lineáris mérőszámot.)
- `tt` = alakzattípus (00...07)

SDO_GTYPE (folytatás)

tt = alakzattípus (00...07):

00 = ismeretlen

01 = pont

02 = vonallánc, amely íves szakaszokat is tartalmazhat

03 = poligon, esetleg lyukakkal (először a külső, azután a belső határvonalat kell felvinni.)

04 = kollekció: tetszőleges elemek halmaza.

05 = ponthalmaz.

06 = vonallánc halmaz.

07 = poligonhalmaz (diszjunkt poligonok).

SDO_GTYPE (példa)

Példa: SDO_GTYPE = 2003 = 2D poligon:

- d = dimenzió = 2.
- l = 0 ha nem LRS alakzat.
- tt = 03 = poligon.

Megjegyzések:

- Egy tábla egy oszlopában csak azonos dimenziójú alakzatok lehetnek.
- Az egyes komponeket visszaadó függvények:
Get_Dims, Get_LRS_Dim, Get_Gtype.

SDO_SRID

Koordinátarendszer (Spatial Reference System)
azonosítója: az SDO_COORD_REF_SYS tábla SRID
oszlopára hivatkozik.

Egy tábla egy oszlopában csak azonos SDO_SRID értékek
szerepelhetnek.

SDO_SRID értéke NULL is lehet (nem definiálunk
koordinátarendszert).

SDO_POINT

Pont alakzat megadása, ha az SDO_ELEM_INFO és SDO_ORDINATES attribútumok NULL értékűek.

Egyébként az SDO_POINT értéke közömbös.

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


SDO_ELEM_INFO

A koordináták értelmezésére szolgál. Típusa:

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

Számhármásokból áll, minden számhármás egy alkotóelem leírását tartalmazza:

- ***SDO_STARTING_OFFSET***: az alkotóelemhez tartozó első koordináta sorszáma az SDO_ORDINATE tömbben (1-től számozva).
- ***SDO_ETYPE***: az alkotóelem típusa (lásd alább).
- ***SDO_INTERPRETATION***. Ha SDO_ETYPE összetett elem, akkor a hármások száma, ha nem, akkor a koordináták értelmezése (lásd alább).

<i>ETYPE</i>	<i>INTERP.</i>	<i>Jelentés</i>
1	1	Pont
1	n	Ponthalmaz
2	1	Vonallánc egyenesszakaszokkal
2	2	Vonallánc körívekkel
1003	1	Külső poligon
2003	1	Belső poligon
1003	2	Külső poligon körívekkel
2003	2	Belső poligon körívekkel
1003	3	Külső téglalap
2003	3	Belső téglalap
1003	4	Külső kör
2003	4	Belső kör
4	n	Vegyes vonallánc (egyenesek, körívek)
1005	n	Külső vegyes poligon (egyenesek, körívek)
2005	n	Belső vegyes poligon (egyenesek, körívek)

SDO_ORDINATES

Koordináták felsorolása, nem tartalmazhat NULL értéket.

Típusa:

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

Szögpontok max. száma: 2D alakzatban 1048576 / 2,
3D alakzatban 1048576 / 3, 4D alakzatban 1048576 / 4.

Szabályok

Poligon megadása: első pont = utolsó pont.

Példa: 4 szögpontú poligon:

$(X_1, Y_1, X_2, Y_2, X_3, Y_3, X_4, Y_4, X_1, Y_1)$

Külső poligon: óramutató járásával szemben körüljárva.

Példa: háromszög: $(1,1, 8,1, 1,5, 1,1)$

Belső poligon: óramutató járása szerint körüljárva.

Példa: háromszög: $(1,1, 1,5, 8,1, 1,1)$

Körív megadása: kezdőpont, belső pont, végpont.

Körívlánc megadása: csatlakozó pontot nem ismételjük.

Példa: $(1,1, 1,5, \underline{5,8}, 10,10, 12,10)$

Kör megadása: három kerületi ponttal.

Téglalap megadása: bal alsó és jobb felső sarok koordinátaival. Példa: $(1,2, 10,20)$

1. példa: pont

`SDO_GTYPE = 2001.`

2D, nem LRS, pont

`SDO_SRID = NULL.`

`SDO_POINT = (10,20).`

`SDO_ELEM_INFO = NULL.`

`SDO_ORDINATES = NULL.`

2. példa: pont felvétele SQL-ben

```
CREATE TABLE Város
( kód NUMBER PRIMARY KEY,
  név VARCHAR2(20),
  hely SDO_GEOMETRY );
```

```
INSERT INTO Város VALUES (
152, 'Pécs',
SDO_GEOMETRY(2001, NULL,
  SDO_POINT_TYPE(1235, 4178, NULL),
  NULL, NULL));
```

3. példa: téglalap

`SDO_GTYPE = 2003`

2D, nem LRS, poligon

`SDO_SRID = NULL.`

`SDO_POINT = NULL.`

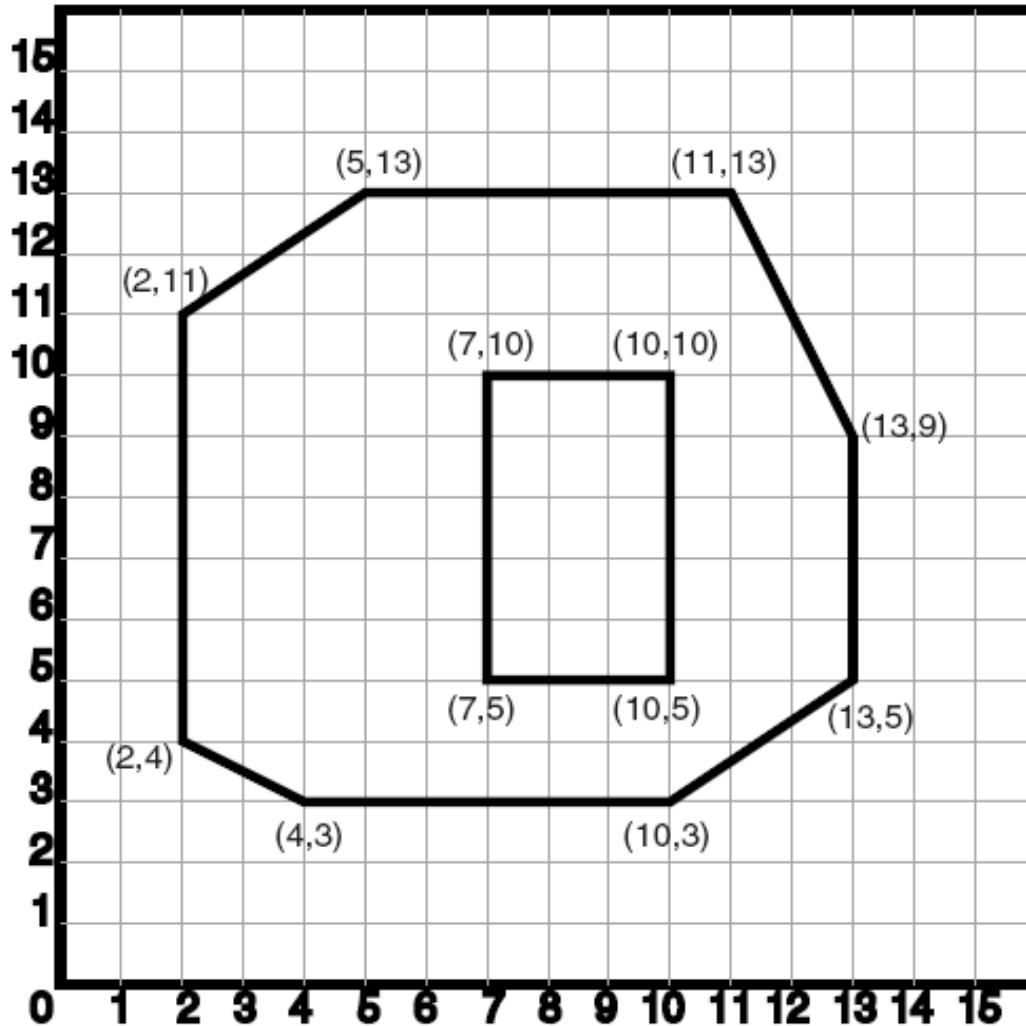
`SDO_ELEM_INFO = (1,1003,3)`

Külső téglalap

`SDO_ORDINATES = (1,1, 5,7)`

Bal alsó, jobb felső

4. példa: poligon lyukkal



4. példa (folytatás)

SDO_GTYPE = 2003

2D, nem LRS, poligon

SDO_SRID = NULL.

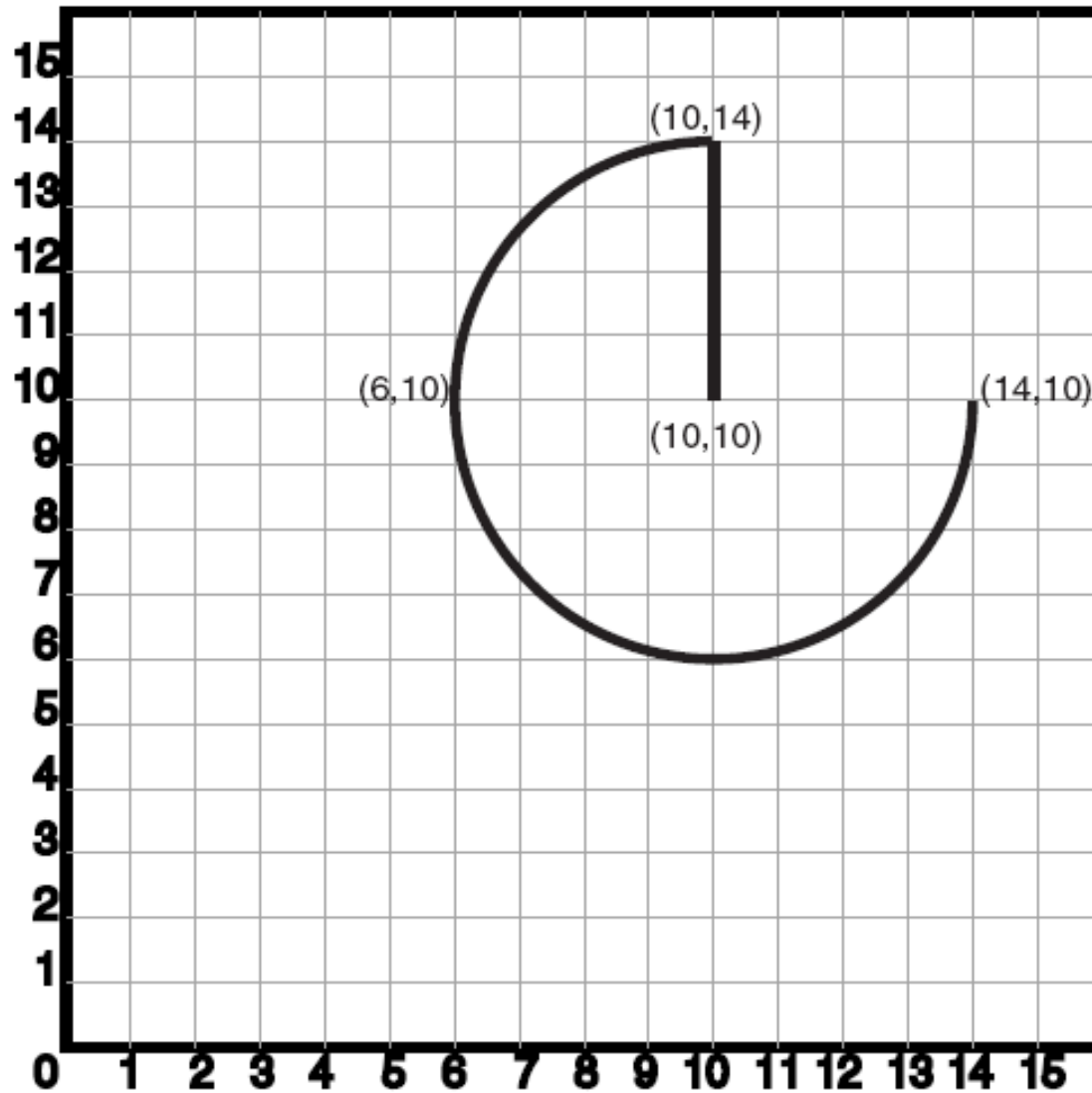
SDO_POINT = NULL.

SDO_ELEM_INFO = (1,1003,1, 19,2003,1)

***Megjegyzés:** 1003: külső poligon, 2003: belső poligon,
ez utóbbi a 19. pozíción kezdődik, első pontja (7,5).*

SDO_ORDINATES = (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) .

5. példa: vegyes vonallánc



5. példa (folytatás)

`SDO_GTYPE = 2002.` 2D vonallánc

`SDO_SRID = NULL.`

`SDO_POINT = NULL.`

`SDO_ELEM_INFO = (1,4,2, 1,2,1, 3,2,2)`

Megjegyzések:

(1,4,2): Vegyes vonallánc 2 komponenssel, amelyek a következők:

(1,2,1): Vonallánc egy vagy több egyenesszakasszal, amely a következő pointerig (3) tart.

(3,2,2): Vonallánc egy vagy több körívvel, amely a több végéig tart.

A csatlakozó pontot (10,14) nem ismételjük.

`SDO_ORDINATES = (10,10, 10,14, 6,10, 14,10).`

6. példa: poligon felvétele SQL-ben

```
CREATE TABLE Telek
( helyrajzszám CHAR(10) PRIMARY KEY,
  terület INTEGER,
  geom SDO_GEOMETRY );

INSERT INTO Telek VALUES
(123, 1200,
 SDO_GEOMETRY(2003, NULL, NULL,
 SDO_ELEM_INFO_ARRAY(1,1003,1),
 SDO_ORDINATE_ARRAY
 (2,4, 10,3, 13,5, 13,9, 5,13, 2,4))
);
```

Megjelenítés

- View → Map View
- Connection beállítása után elérhetők a Display Geometry Shape és az Invoke Map View menüpontok

Index létrehozása

```
CREATE INDEX geomind ON Telek (geom)  
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

SDO_DISTANCE (geom1, geom2, tolerance [, params])

- A geom1 és geom2 alakzatok minimális távolságát (azaz legközelebbi pontjaik távolságát) adja vissza.
- tolerance: kerekítési hibák elkerülésére szolgáló tolerancia paraméter, szokásos értéke 0,5. (Ha például a két alakzat távolsága kisebb, mint tolerance, akkor a távolság nulla lesz.)
- params: a távolság mértékegységének megadására szolgál. Elhagyható, ekkor a koordinátarendszerhez megadott mértékegység érvényes.

RELATE (geom1, mask, geom2, tolerance)

- Két alakzat között fennálló geometriai kapcsolatot határoz meg, ahol mask a meghatározandó kapcsolatot leíró karakterlánc, értékei a következők lehetnek:
 1. DETERMINE: ebben az esetben a függvény határozza meg a kapcsolat típusát, amelyet stringként ad vissza (például 'INSIDE').
 2. A következők egyike: INSIDE, COVEREDBY, COVERS, CONTAINS, EQUAL, OVERLAPBDYDISJOINT, OVERLAPBDYINTERSECT, ON, TOUCH. A függvény értéke a maszkként megadott kapcsolat típusa (ha teljesül), vagy 'FALSE' (ha nem teljesül).
 3. ANYINTERACT: a visszaadott érték 'TRUE', ha bármely a 2) pontban leírt tulajdonság fennáll, 'FALSE' egyébként.
 4. DISJOINT: a visszaadott érték 'TRUE', ha 2) pontban felsorolt tulajdonságok egyike sem áll fenn, 'FALSE' egyébként.

Geometriai kapcsolatok

DISJOINT: The boundaries and interiors do not intersect.

TOUCH: The boundaries intersect but the interiors do not intersect.

OVERLAPBDYDISJOINT: The interior of one object intersects the boundary and interior of the other object, but the two boundaries do not intersect (for example, when a line originates outside a polygon and ends inside that polygon).

OVERLAPBDYINTERSECT: The boundaries and interiors of the two objects intersect.

EQUAL: The two objects have the same boundary and interior.

CONTAINS: The interior and boundary of one object is completely contained in the interior of the other object.

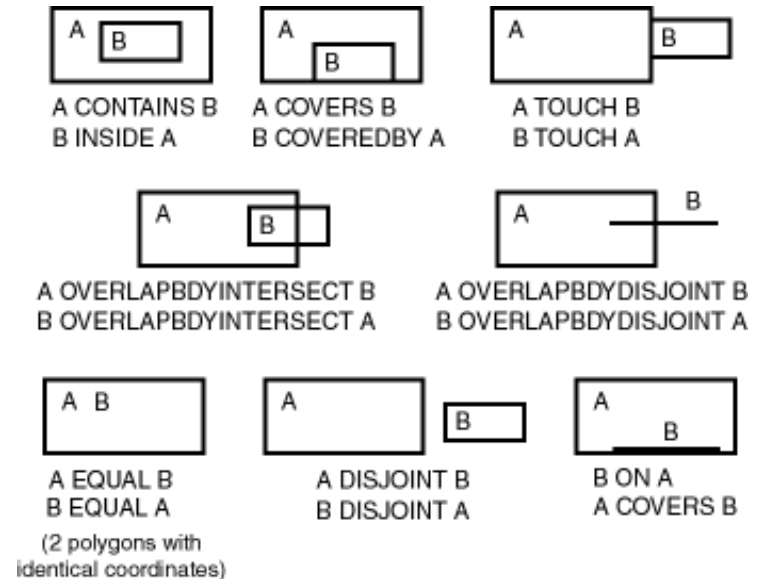
INSIDE: The opposite of CONTAINS. A INSIDE B implies B CONTAINS A.

COVERS: The interior of one object is completely contained in the interior or the boundary of the other object and their boundaries intersect.

COVEREDBY: The opposite of COVERS. A COVEREDBY B implies B COVERS A.

ON: The interior and boundary of one object is on the boundary of the other object (and the second object covers the first object) (for example, when a line is on the boundary of a polygon).

ANYINTERACT: The objects are non-disjoint.



Lekérdezések - 1

Lista távolság szerint növekvő sorrendben azon kutakról, amelyek az 1234 helyrajzi számú telektől legfeljebb 100 koordinátaegységre vannak:

```
SELECT id, típus, SDO_DISTANCE(Telek.geom,  
    Kút.geom, 0.5) AS távolság  
FROM Telek, Kút  
WHERE Telek.helyrajziszám=1234 AND  
SDO_GEOM.SDO_DISTANCE (Telek.geom, Kút.geom,  
    0.5) < 100  
ORDER BY távolság;
```

Lekérdezések - 2

Lista azon telkekről, amelyeken artézi kút van

```
SELECT helyrajzszám FROM Telek, Kút  
WHERE Kút.típus='Artézi' AND  
SDO_GEOM.RELATE (Kút.geom, 'INSIDE', Telek.geom,  
0.5) = 'INSIDE';
```

Oracle Spatial összefoglalás

- Egyetlen univerzális objektumtípus (SDO_GEOMETRY).
- Egy oszlop vegyesen tartalmazhat különféle elemtípusokat.
- Körívek speciális kezelése.
- A nehézkes megadási mód konverziós eszközökkel, PL/SQL programmodulokkal és grafikus szerkesztéssel könnyíthető.