



Bazy Danych

5. Przestrzenne Bazy Danych

Opracował: Maciej Penar

Spis treści

1. Linki	3
2. Dlaczego ten temat?	3
3. Cóż to są te „dane przestrzenne”?	4
Kontekst.....	4
PRzykład.....	4
Operacje	5
4. (6 pkt) Zadania	6
Treść właściwa zadania z parkingiem.....	7

1. Linki

Przydatne linki:

- Oracle 12c: [link](#)
- Przestrzenne typy danych w Oracle: [link](#)
- Przestrzenne dane w Oracle: [link](#)
- Operatory przestrzenne w Oracle: [link](#)
- Funkcje przestrzenne w Oracle: [link](#)

2. Dlaczego ten temat?

Każdy bazodanowy „duży gracz” ma pewną funkcjonalność której nie można znaleźć u konkurentów. I tak:

- IBM DB2 posiada mechanizm BLU: shadowing + columnstore + in-memory
- Microsoft SQL Server: posiada przetwarzanie pełnotekstowe / semantic search
- Oracle DB: posiada rozszerzenie temporalne

Z tego względu warto zapoznać się z funkcjonalnością która czyni Oracle DB wyjątkową bazą danych.

3. Cóż to są te „dane przestrzenne”?

KONTEKST

Dane przestrzenne (ang. Spatial Data) jest to specjalny typ danych opisujący figury geometryczne np.:

- Punkty
- Linie
- Płaszczyzny (ang. Surfaces)
- Przestrzenie (ang. Planes)
- Hiper-przestrzenie (ang. Hype-planes)

W Oracle istnieje od tego specjalny typ danych: **SDO_GEOMETRY**. Ogólnie rzecz biorąc konstruowanie tego typu obiektów nie należy do najłatwiejszych.

Obiekt przestrzenny jest zdefiniowany jako (nie wykonywać):

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

Gdzie:

- SDO_GTYPE – definiuje typ obiektu przestrzennego: ile ma wymiaru, jaki rodzaj obiektu to jest
- SDO_SRID – definiuje w jakim układzie współrzędnych osadzony jest obiekt
- SDO_POINT – definiuje punkt dla obiektów punktowych
- SDO_ELEM – definiuje ile i jakie punkty wchodzi w skład topologii
- SDO_ORDINATES – definiuje parametry obiektów z SDO_ELEM

PRZYKŁAD

Dana jest tabela:

```
CREATE TABLE SENSORS(  
    VALUE NUMERIC,  
    SHAPE SDO_GEOMETRY  
);
```

Ładowanie **punktu** o wartości 125 i koordynatach (10,5):

```
INSERT INTO SENSORS(VALUE, SHAPE)  
VALUES(125, MDSYS.SDO_GEOMETRY(2001, NULL, MDSYS.SDO_POINT_TYPE(10, 5,  
NULL), NULL, NULL));
```

Spróbuj załadować:

- Punkt o wartości 65 i koordynatach (0, 0)
- Punkt o wartości 43 i koordynatach (2, 2)
- Punkt o wartości 255 i koordynatach (8, 7)

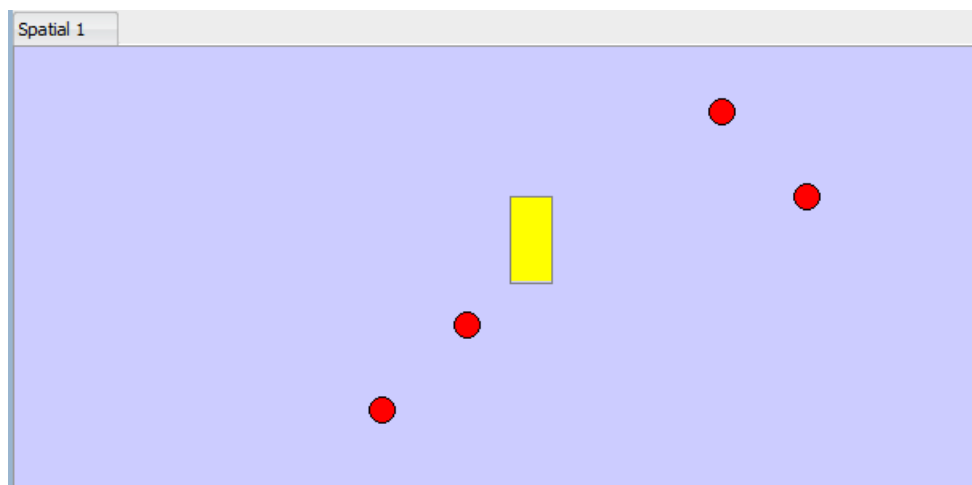
Ładowanie prostokąta o wartości 0 i koordynatach (3,3) do (4,5)

```
INSERT INTO SENSORS(VALUE, SHAPE)
VALUES(0, MDSYS.SDO_GEOMETRY(2003, NULL, NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3), MDSYS.SDO_ORDINATE_ARRAY(3,3, 4,5)));
```

Obiekty przestrzenne mają reprezentację na mapie. Mapę można obejrzeć:

1. View->MapView
2. Klikamy 
3. Wpisujemy nazwę oraz zapytanie: SELECT SHAPE FROM SENSORS

Powinniśmy zobaczyć:



OPERACJE

Na obiektach przestrzennych możemy wykonać wiele operacji dostępnych w formie specjalnych funkcji zwanych operatorami oraz funkcjami przestrzennymi. Do najciekawszych należą:

- SDO_GEOM.SDO_UNION – do liczenia unii dwóch przestrzeni
- SDO_GEOM.SDO_INTERSECTION – do liczenia przecięcia dwóch przestrzeni
- SDO_GEOM.SDO_DIFFERENCE – do liczenia różnicy dwóch przestrzeni
- SDO_GEOM.SDO_DISTANCE – do liczenia odległości pomiędzy dwoma przestrzeniami
- SDO_GEOM.SDO_AREA – do liczenia powierzchni dwu-wymiarowych przestrzeni
- SDO_GEOM.SDO_CENTROID – do wyznaczania centroidu przestrzeni

4. (6 pkt) Zadania

1. (1 pkt) Narysować różnicę prostokątów: (1,1) (7,5) i (2,1)(5,4), próg dokładności przyjąć 0.1
2. (1 pkt) Policzyc pole z 1), próg dokładności przyjąć 0.1
3. (4 pkt) Załóżmy istnienie tabeli PARKING_SPOT takiej jak:

```
CREATE TABLE PARKING_SPOT(  
    NAME VARCHAR(100) NOT NULL UNIQUE,  
    MAX_CAP INT CHECK (MAX_CAP >=0) NOT NULL,  
    CURRENT_CAP INT,  
    PLACE SDO_GEOMETRY NOT NULL,  
    CHECK (CURRENT_CAP <= MAX_CAP)  
)
```

Gdzie:

- NAME to nazwa parkingu
- MAX_CAP to maksymalna liczba samochodów na parkingu
- CURRENT_CAP to aktualna liczba samochodów na parkingu
- PLACE to topologia parkingu

Założmy że istnieją następujące parkingi:

```
INSERT INTO PARKING_SPOT  
VALUES('PARKING A', 100, 98, SDO_GEOM.SDO_DIFFERENCE(  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(8, 8, 12, 14)),  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(8, 8, 10, 10)),  
    0.1  
));
```

```
INSERT INTO PARKING_SPOT  
VALUES('PARKING B', 5, 3,  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(4, 4, 5, 5))  
);
```

```
INSERT INTO PARKING_SPOT  
VALUES('PARKING C', 5, 5,  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(5, 6, 6, 7))  
);
```

```
INSERT INTO PARKING_SPOT  
VALUES('SUNNY PARKING', 75, 73,  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(0, 0, 2, 10))  
);
```

```
INSERT INTO PARKING_SPOT  
VALUES('PARKING DARK', 100, 98, SDO_GEOM.SDO_DIFFERENCE(  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(-2, -2, 16, 16)),  
    MDSYS.SDO_GEOMETRY(2003, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),  
MDSYS.SDO_ORDINATE_ARRAY(-1, -1, 15, 15)),  
    0.1  
));
```

TREŚĆ WŁAŚCIWA ZADANIA Z PARKINGIEM

(3 pkt) Napisać funkcję która:

- **Przyjmuje w parametrze pojazd (pojazd można zamodelować w dowolny sposób)**
- **Rezerwuje miejsce na najbliższym dostępnym parkingu**
- **Zwraca nazwę parkingu**

(1 pkt) Jaki jest wynik dla ciągu testowego:

- Samochód 1 jest w punkcie (8,8)
- Samochód 2 jest w punkcie (5.5, 6.5)
- Samochód 3 jest w punkcie (13, 13.5)
- Samochód 4 jest w punkcie (13, 13.5)
- Samochód 5 jest w punkcie (4.8, 1.2)
- Samochód 6 jest w punkcie (4.75, 1.2)
- Samochód 7 jest w punkcie (4.9, 1.2)
- Samochód 8 jest w punkcie (0,0)
- Samochód 9 jest w punkcie (0,0)