

Jarosław GRAMACKI, Artur GRAMACKI

UNIwersytet Zielonogórski, INSTYTUT INFORMATYKI I ELEKTRONIKI

Dane przestrzenne w bazach relacyjnych. Model danych, zapytania przestrzenne

Dr inż. Jarosław GRAMACKI

Pracuje w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego na stanowisku adiunkta. Jego zainteresowania koncentrują się wokół zagadnień związanych z bazami danych, hurtowniami danych oraz kopalniami danych. Pracuje głównie w środowisku bazy Oracle, jednak z dużą sympatią odnosi się do baz danych związanych z ruchem Open Source. Swoje doświadczenie stara się wykorzystywać zarówno w pracy dydaktycznej, jak i aktywnie uczestnicząc w różnych projektach bazodanowych.

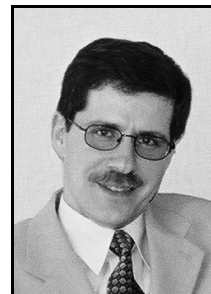
e-mail: j.gramacki@iie.uz.zgora.pl



Dr inż. Artur GRAMACKI

Pracuje w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego na stanowisku adiunkta. Interesuje się zagadnieniami związanymi z bazami danych, w szczególności firmy Oracle ale też rozwiązaniami Open Source. Oprócz prowadzenia zajęć dydaktycznych stara się wykorzystywać swoją wiedzę uczestnicząc w różnych projektach informatycznych z tego zakresu. Brał udział w czterech projektach, których celem było przygotowanie systemów wspomagających działalność UZ.

e-mail: a.gramacki@iie.uz.zgora.pl



Streszczenie

W artykule przedstawiono wybrane zagadnienia dotyczące pracy z tzw. mapami cyfrowymi (ang. Digital Maps) w kontekście wykorzystania do ich obsługi specjalizowanych rozwiązań dostępnych w systemach relacyjnych baz danych. Z продемонstrowano rozwiązanie oferowane przez firmę Oracle w postaci pakietu o nazwie Oracle Spatial. Przykłady użytecznych systemów zarządzających danymi przestrzennymi pokazano w drugiej części pracy „Dane przestrzenne w bazach relacyjnych. Wykorzystanie danych przestrzennych i systemy zarządzania danymi przestrzennymi”.

Słowa kluczowe: mapy cyfrowe, dane przestrzenne w bazach relacyjnych, bazy relacyjne, systemy klasy GIS, Oracle Spatial

Spatial Data in Relational Databases. Data Model and Spatial Queries

Abstract

In the paper some problems related to digital maps in the context of using classical relational databases were presented. A solution called Oracle Spatial was briefly demonstrated. Some more detailed information as well as some useful solutions of systems for spatial data management was presented in the separated paper entitled „Spatial Data in Relational Databases. Using of Spatial Data and Spatial Data Management Systems”.

Keywords: digital maps, spatial data in relational databases, relational databases, GIS systems, Oracle Spatial

1. Wstęp

Systemy informatyczne korzystające z map cyfrowych (mapa cyfrowa może być uważana za odpowiednik tradycyjnej mapy papierowej, jednak o daleko większej funkcjonalności) stosowane są od dość dawna. Systemy takie noszą wspólną nazwę Systemów Informacji Geograficznej (ang. *Geographic Information Systems*, GIS). Głównymi klientami tego rodzaju systemów są przedsiębiorstwa sieciowe, związane z energetyką, ciepłownictwem, telekomunikacją itp. W wielu jednak przypadkach cechą charakterystyczną używanych systemów jest rozdzielenie miejsca przechowywania mapy cyfrowej (grafika wektorowa lub rastrowa) z zaznaczonymi na niej obiektami od danych alfanumerycznych opisujących te obiekty.

W ostatnim czasie pojawiły się komercyjne systemy GIS, bazujące na relacyjnych bazach danych [1]. Bazy te zostały mianowicie uzupełnione o odpowiednie rozwiązania umożliwiające przechowywanie i efektywny dostęp do danych o charakterze przestrzennym. Stało się to możliwe dzięki implementacji specjalizowanych obiektowych typów danych oraz indeksów ukierunkowanych na efektywną obsługę danych przestrzennych [2, 3]. Wspomniane możliwości GIS-owe systemów bazodanowych (ang. *Geodatabases*) są na tyle zaawansowane i atrakcyjne, że warto rozważyć możliwości ich zastosowania w projektach informatycznych.

Ponadto „osadzając” systemy GIS-owe wprost w bazie, otrzymujemy automatycznie możliwość wykorzystywania ogromnego potencjału jaki tkwi we współczesnych bazach relacyjnych. Jako przykład wymienimy tu choćby eksplorację danych przestrzennych (ang. *Spatial Data Mining*) [2, 4] – dziedzinę do niedawna zarezerwowaną wyłącznie do analizy danych tekstowych.

Możliwości przechowywania i zarządzania w bazie relacyjnej informacjami przestrzennymi posiadają również systemy typu *Open Source* [5, 7], co dowodzi, że rozwiązania te nieuchronnie stają się powszechnie dostępnymi implementacjami od dość dawna istniejącego już standardu [6].

Oczywiści w relacyjnej bazie danych gromadzić można wiele informacji o obiektach przestrzennych w postaci alfanumerycznej. Jednak jedynie ich postać przestrzenna umożliwia efektywne i proste zadawanie zapytań takich jak na przykład: „od której stacji, A czy B, bliżej jest do stacji C”, „które powiaty sąsiadują w odległości nie większej niż 100km z powiatem B”, „który z powiatów A czy B jest większy”, „czy powiaty A i B sąsiadują z powiatem C” itd.

2. Oracle Spatial

Moduł *Oracle Spatial* jest zintegrowany z bazą Oracle zbiorom typów obiektowych, specjalizowanych indeksów przestrzennych, metadanych, operatorów, funkcji i procedur umożliwiających efektywne przechowywanie, dostęp oraz analizę danych topograficznych (operujemy na współrzędnych) czyli danych o charakterze przestrzennym. Na danych tych możemy wykonywać takie operacje jak np. „znajdź obiekty oddalone o x km od danego obiektu Y ”

Dodatkowo obiekty te mogą wchodzić między sobą w związki tworząc sieci obiektów przestrzennych, które charakteryzują się istnieniem węzłów, krawędzi pomiędzy węzłami ich skierowaniem oraz ew. przypisaną do każdego elementu nieujemną wartością interpretowaną w analizach jako np. koszt węzła lub czas pokonania krawędzi (*Oracle Spatial Topology and Network Data Models*). Na danych posiadających przypisane informacje o koszcie i czasie możliwe staje się wykonywanie choćby takich operacji jak przykładowo „znajdź wszystkie bezkolizyjne drogi z X do Y , wybierz najszybszą z nich”.

Stosowane jest również tzw. geokodowanie (ang. *Geocoding*), mające na celu określenie współrzędnych obiektu na podstawie np. zarejestrowanych danych adresowych obiektu.

Użytkownik nie jest ograniczony jedynie do danych przestrzennych o charakterze wektorowym. Przykładowo zeskanowane podkłady rastrowe czy zdjęcia lotnicze można również przechowywać w bazie danych wraz z podłączonymi do nich metadanymi „pozycjonującymi” podkłady (oraz obiekty znajdujące się na nich) w danym układzie odniesienia (ang. *Oracle Spatial GeoRaster*).

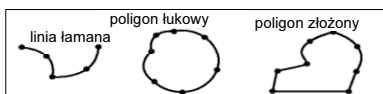
Dostęp do wymienionej wyżej funkcjonalności modułu *Oracle Spatial* zapewnia programistyczny interfejs (API) w językach PL/SQL lub Java. Co ważne, wszystkie dane przestrzenne i towa-

rzyszające im metadane przechowywane są w jednej bazie danych. Jest to ogromna zaleta, gdyż bardzo upraszcza wiele zagadnień (zarządzanie, kopie bezpieczeństwa, synchronizacja danych tekstowych z danymi przestrzennymi i inne).

W dalszej części pracy przedstawione zostaną podstawowe pojęcia związane ze sposobem opisu w relacyjnej bazie danych obiektów przestrzennych oraz podane zostaną przykłady ilustrujące pracę z takimi obiektami w środowisku bazodanowym.

3. Model danych przestrzennych

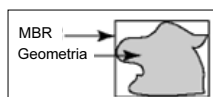
Do opisu obiektu przestrzennego używa się tzw. geometrii (ang. *geometry*) czyli uporządkowanego zbioru punktów połączonych liniami prostymi i/lub łukami. Z geometrii tych tworzone są następnie podstawowe elementy takie jak: koła, prostokąty, linie łamane, poligony proste, poligony łukowe itp. (patrz rysunek 1). Elementy te z kolei umieszczane mogą być na niezależnych warstwach (ang. *layers*). Całość dopełniają dane opisujące wybrany system koordynat przestrzennych typu geodezyjnego (georeferencyjnego), kartezjańskiego (układ osi), własnego wybranego systemu lokalnego (ang. *coordinate system*) oraz przyjęta tolerancja dokładności (ang. *tolerance*). Przez koordynatę rozumiemy możliwość określenia na podstawie przyjętej metody pozycji obiektu w rzeczywistości.



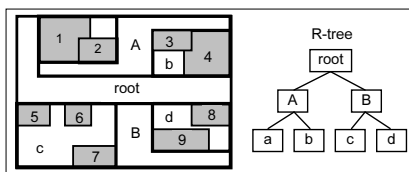
Rys. 1. Przestrzenne typy geometryczne
Fig. 1. Spatial Geometric Types

Indeksowanie danych przestrzennych

Dane o charakterze przestrzennym wymagają specjalizowanych indeksów. W systemie *Spatial* zaimplementowano tzw. indeksy *R-tree*, w którym dokonuje się przybliżenia każdej geometrii przez najmniejszy obejmujący ją prostokąt (ang. MBR – *minimum bounding rectangle*, patrz rysunek 2). Stosowne indeksy przestrzenne obsługiwane są zwykłymi poleceniami DDL (CREATE, ALTER, DROP) oraz DML (INSERT, UPDATE, DELETE). Istotę tych indeksów przybliża rysunek 3.



Rys. 2. Minimalny prostokąt otaczający geometrię
Fig. 2. Minimum bounding rectangle (MBR) enclosing a geometry



Rys. 3. Hierarchiczny indeks obiektów typu MBR
Fig. 3. R-Tree Hierarchical Index on MBRs

Przestrzenny typ danych

W systemie Oracle schematem „sterującym” modulem *Spatial* jest schemat MDSYS. Każda geometria jest przechowywana w bazie jako jeden wiersz w tabeli zawierającej (co najmniej jedną) obiektową kolumnę typu SDO_GEOMETRY. Poniżej pokazano formalne deklaracje odpowiednich typów. Jest to niezbędne do właściwej interpretacji zamieszczanych w pracy przykładów.

```
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);
```

```
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;
```

Przykładowo rozmiar przy SDO_ORDINATE_ARRAY 1.048.576 oznacza maksymalną liczbę możliwych do przechowania współrzędnych wierzchołków danej geometrii co dla obiektu dwuwymiarowego daje „astronomiczną” liczbę 524.288 wierzchołków.

Dla tak zdefiniowanych typów przestrzennych postać tabeli przechowującej elementy przestrzenne z reguły zawiera jedną kolumnę przechowującą dane przestrzenne jednej warstwy tematycznej plus inne wymagane w danym zastosowaniu kolumny:

```
CREATE TABLE MySpatial (
  id      NUMBER PRIMARY KEY,
  name    VARCHAR2(32),
  geom    SDO_GEOMETRY);
```

Poszczególne pola obiektu sdo_geometry oznaczają:

SDO_GTYPE	Typ geometrii, liczba 4. cyfrowa, przykładowa wartość 2003 oznacza: (2) obiekt 2 wymiarowy, (0) nie używane w obiektach 2D, (03) obiekt typu <i>polygon</i> .
SDO_SRID	System koordynat (ang. <i>Coordinate Systems - Spatial Reference Systems</i>). Wszystkie geometrie w tej samej kolumnie muszą posiadać tę samą wartość. Gdy NULL to żaden (zdefiniowany) system koordynat nie jest wykorzystywany. W tabeli MDSYS.CS_SRS zdefiniowano 1018 różnych koordynat (kartezjańskich, geodezyjnych, innych). Jeden z wpisów (Longitude / Latitude (WGS 84) 8307) zawiera opis koordynaty stosowanej m.in. przez system GPS.
SDO_POINT	Trzy wielkości (x, y, z) opisujące współrzędne geometrii typu punkt.
SDO_ELEM_INFO	Gdy geometria nie jest punktem przechowywanym w SDO_POINT, wtedy SDO_ELEM_INFO zawiera opis elementów wchodzących w skład geometrii. Każdy element opisany jest przez trzy parametry: pozycję pierwszej koordynaty, typ elementu, opis elementu.
SDO_ORDINATES	Współrzędne punktów wszystkich elementów tworzących geometrię. Najbardziej „rozbudowany” element opisu.

Metadane

Poszczególne kolumny typu SDO_GEOMETRY (które mogą odpowiadać na przykład kolejnym warstwom tematycznym mapy) mogą (dla niektórych zastosowań muszą) być dodatkowo opisywane w systemie za pomocą tzw. *metadanych*. Utworzenie metadanych jest również konieczne dla poprawnego utworzenia indeksu na danej kolumnie zawierającej geometrie (niektóre zapytania bez indeksów w ogóle nie działają). Metadane mają postać perspektyw systemowych [USER|ALL|DBA]_SDO_GEOM_METADATA i strukturę:

```
( TABLE_NAME  VARCHAR2(32),
  COLUMN_NAME  VARCHAR2(32), -- kolumna SDO_GEOMETRY
  DIMINFO      MDSYS.SDO_DIM_ARRAY, -- info. o wymiarach
  SRID         NUMBER -- informacja o wykorzystywanych
); -- koordynatach, na przykład 8307 dla GPS
```

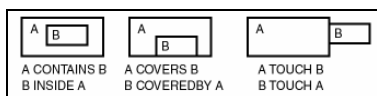
4. Zapytania przestrzenne

Dostęp do zgromadzonych danych przestrzennych umożliwiają zapytania przestrzenne. Składnia zapytań nie odbiega od standardowej składni SQL.

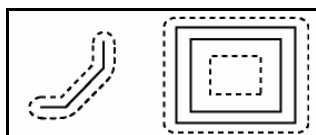
Operatory przestrzenne

W zapytaniach intensywnie korzysta się z wielu dedykowanych operatorów przestrzennych (ang. *spatial operators*) implementujących takie relacje jak na przykład: CONTAINS, INSIDE, COVERS, TOUCH i wiele tym podobnych (rysunek 4). Dodatkowo na potrzeby niektórych z nich wprowadza się pojęcie bufora otaczającego, patrz rysunek 5 (ang. *distance buffer*). Przykładowo operator

`SDO_FILTER` określa na przykład sąsiadów podanego obiektu przestrzennego. Opis przykładowych obiektów przestrzennych oraz więcej przykładów zapytań przestrzennych znajduje się w drugiej części pracy.



Rys. 4. Relacje topologiczne
Fig. 4. Topological Relationships"



Rys. 5. Bufor otaczający dla obiektu liniowego i poligonu
Fig. 5. Distance Buffers for Lines, and Polygons

```
SELECT A.wo_nazwa, B.wo_nazwa
FROM so_województwa A, so_województwa B
WHERE sdo_filter(A.wo_ksztalt, B.wo_ksztalt,
'querytype=join') = 'TRUE' AND B.wo_nazwa = 'lubuskie';
```

```
WO_NAZWA
-----
dolnośląskie
wielkopolskie
zachodniopomorskie
```

Agregacja danych

Podobnie jak w klasycznych zapytaniach typu „SELECT MIN(salary), dept FROM Employees GROUP BY dept;” możliwe jest zapytanie się na przykład o element o najmniejszym MBR lub sumę powierzchni wybranych geometrii. Konieczne są więc przestrzenne odpowiedniki funkcji agregujących (ang. *Spatial Aggregate Functions*). Przykładowo funkcja `SDO_AGGR_UNION` wyznacza topologiczną sumę zbioru obiektów przestrzennych.

Pakiety funkcji i procedur

Wśród wielu dostępnych w systemie *Spatial* pakietów PL/SQL (które decydują *de facto* o sile systemu) podstawowym i najczęściej używanym jest pakiet `SDO_GEOM`. Zawiera on wiele funkcji oraz procedur geometrycznych operujących na obiektach przestrzennych. Przykładowo funkcja `SDO_GEOM.SDO_DISTANCE` oblicza odległość od dwóch dowolnych obiektów przestrzennych (dwóch najbliższych sobie punktów obu obiektów).

Przykłady użycia omówionych wyżej narzędzi pokazane zostaną w drugiej części pracy „Dane przestrzenne w bazach relacyjnych. Wykorzystanie danych przestrzennych i systemy zarządzania danymi przestrzennymi”.

Inne możliwości

Oprócz opisanych wyżej operatorów, funkcji i agregatów operujących na danych przestrzennych, w pakiecie *Spatial* dostępnych jest również wiele nieco bardziej specjalistycznych narzędzi:

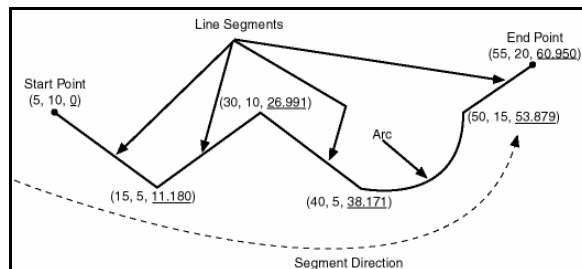
- funkcje służące do transformacji pomiędzy różnymi systemami koordynat zawarte w pakiecie `SDO_CS` (ang. *Coordinate System Transformation*),
- obiekty służące do definiowania różnych systemów koordynat zawarte w pakietach `SDO_COORD`, `SDO_CRS`, (ang. *Coordinate Systems*, *Spatial Reference Systems*),
- liniowe funkcje odniesienia z pakietu `SDO_LRS` (ang. *Linear Referencing System*) służące do łączenia w jeden wspólny obiekt wielu „podobiektów” celem na przykład zamodelowania długiego odcinka gazociągu (patrz rysunek 6).

Interfejs programistyczny Java

W dokumentacji cała funkcjonalność systemu *Spatial* ilustrowana jest z użyciem interfejsu PL/SQL gdyż jest to naturalne

rozszerzenie możliwości języka SQL. Niemniej istnieją również cztery pakiety w ramach *Oracle Spatial Java API*, a mianowicie:

- `oracle.spatial.geometry`
- `oracle.spatial.network`
- `oracle.spatial.topo`
- `oracle.spatial.util`



Rys. 6. Podział obiektu na segmenty w systemie LRS. Zaznaczono punkty początkowy, końcowy oraz przypisane punktom miary
Fig. 6. Geometric Segments in LRS System with start and end points and measures of points defined

Dostarczają one analogicznej funkcjonalności jaką dostarcza interfejs PL/SQL. Możliwość stosowania języka Java do tworzenia aplikacji przestrzennych sprowadza się do użycia stosownej klasy zawartej w pakiecie, która poprzez zdefiniowane w niej pola oraz metody „mapuje” typ `MDSYS.SDO_GEOMETRY`, tym samym udostępniając aplikacji obiekty przestrzenne z bazy danych. Podstawowy schemat użycia Java API wygląda następująco:

```
// oracle.spatial.geometry.JGeometry
// Odczyt geometrii z bazy
ResultSet rs = statement.executeQuery(
"SELECT geom FROM Table WHERE col='abc'");
STRUCT st = (oracle.sql.STRUCT) rs.getObject(1);
//konwersja STRUCT na geometrię
JGeometry j_geom = JGeometry.load(st);

// ... tu manipulacja geometrią

// zapis geometrii do bazy
PreparedStatement ps = connection.prepareStatement(
"UPDATE Table SET geom=? where col='abc'");
// konwersja instancji JGeometry do DB STRUCT
STRUCT obj = JGeometry.store(j_geom, connection);
ps.setObject(1, obj); ps.execute();
```

5. Podsumowanie

W artykule zaprezentowano podstawowe cechy pakietu *Spatial* będącego rozszerzeniem funkcjonalności bazy danych Oracle. Pakiet ten daje możliwość pełnej integracji systemów klasy GIS z klasycznymi systemami zarządzania relacyjnymi bazami danych. Jest bardzo ciekawą alternatywą dla systemów GIS-owych typu *standalone*.

6. Literatura

- [1] ESRI GIS and Mapping Software / ArcSDE (Arc Spatial Data Server), <http://www.esri.com/>
- [2] Oracle Spatial, Oracle Locator, Extensions for Oracle Database, <http://www.oracle.com/technology/products/spatial/>
- [3] DB2 Geodetic Extender, DB2 Spatial Extender, <http://www-306.ibm.com/software/data/spatial/>
<http://www-306.ibm.com/software/data/db2/udb/geodetic.html>
- [4] Oracle Data Mining, Extensions for Oracle Database, <http://www.oracle.com/technology/products/bi/odm/index.html>
- [5] MySQL, MySQL 5.1 Reference Manual, Chapter 18, Spatial Extensions in MySQL, <http://dev.mysql.com/doc/refman/5.1/en/index.html>
- [6] SQL/MM Part 3 Spatial specification, <http://www.iso.org/iso/en/ISOOnline.frontpage>
- [7] Open Source GIS, <http://www.opensourcegis.org/>