

# GIS i dane geograficzne w bazach relacyjnych

Sebastian Ernst


*Zaawansowane Technologie Bazodanowe*

# Przykład na początek

- Baza danych przechowuje informacje o bankomatach:
  - sieć,
  - godziny otwarcia,
  - lokalizacja.







- Baza danych przechowuje informacje o bankomatach:
  - sieć,
  - godziny otwarcia,
  - lokalizacja.

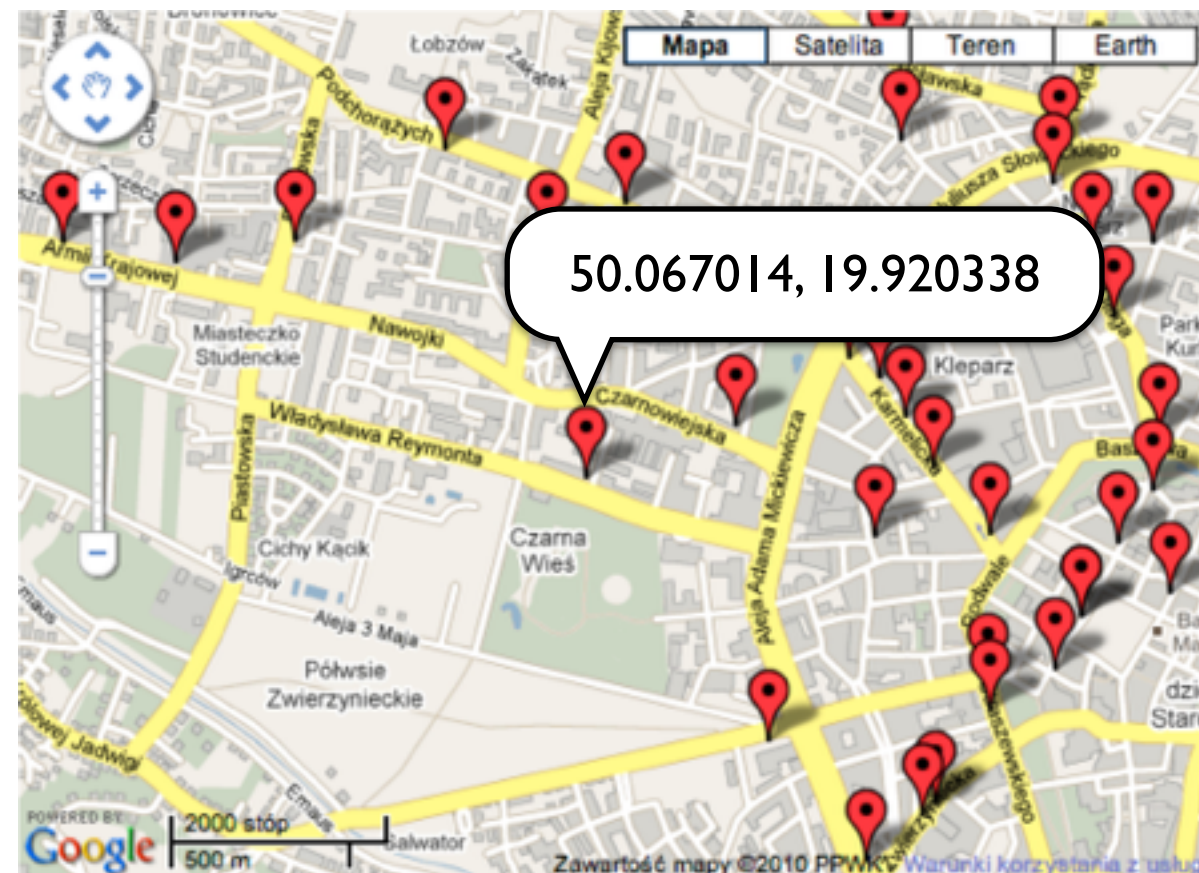
Lokalizacja	Godziny otwarcia	Sieć
BP Stacja Paliw 29 Listopada 39B	24 h	
PSS Podwawelska 29-listopada 57/59	24 h	
BP Stacja Paliw al. 29 Listopada 125	24 h	
OBI al. Bora Komorowskiego 31	24 h	
Comarch al. Jana Pawła II 41e	Pn.-pt. 8:00-18:00	
Alior Bank al. Kijowska/ Kazimierza Wielkiego	24 h	
Skup Clear Channel al. Mickiewicza (Akademia Rolnicza)	24 h	
PSS Kraków al. Pokoju 20	24 h	
Raiffeisen Bank Polska S.A. al. Pokoju 44	24 h	
Plaza Center Kraków al. Pokoju 44	24 h	
Praktiker al. Pokoju 67	24 h	
CH M1 al. Pokoju 67	24 h	
Statoll, stacja paliw Armii Krajowej	24 h	
Raiffeisen Bank Polska S.A. Armii Krajowej 18	24 h	
Market Punkt Balicka 7	24 h	
Firma AKT Bartla 20	24 h	
Delikatesy "Złoty Róg" Bałuckiego 9	24 h	


















Lokalizacja	Godziny otwarcia	Sieć
BP Stacja Paliw 29 Listopada 39B	24 h	
PSS Podwawelska 29-listopada 57/59	24 h	
BP Stacja Paliw al. 29 Listopada 125	24 h	
OBI al. Bora Komorowskiego 31	24 h	
Comarch al. Jana Pawła II 41e	Pn.-pt. 8:00-18:00	
Alior Bank al. Kijowska/ Kazimierza Wielkiego	24 h	
Słup Clear Channel al. Mickiewicza (Akademia Rolnicza)	24 h	
PSS Kraków al. Pokoju 20	24 h	
Raiffeisen Bank Polska S.A. al. Pokoju 44	24 h	
Plaza Center Kraków al. Pokoju 44	24 h	
Praktiker al. Pokoju 67	24 h	
CH M1 al. Pokoju 67	24 h	
Statoil, stacja paliw Armii Krajowej	24 h	
Raiffeisen Bank Polska S.A. Armii Krajowej 18	24 h	
Market Punkt Bałicka 7	24 h	
Firma AKT Bartla 20	24 h	
Delikatesy "Złoty Róg" Bałuckiego 9	24 h	

```
CREATE TABLE `bankomaty` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `location` varchar(30) NOT NULL,
  `working_hours` varchar(10) NOT NULL,
  `network` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

```
CREATE TABLE `bankomaty` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `location` varchar(30) NOT NULL,
  `working_hours` varchar(10) NOT NULL,
  `network` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Lokalizacja	Godziny otwarcia	Sieć
BP Stacja Paliw 29 Listopada 39B	24 h	
PSS Podwawelska 29-listopada 57/59	24 h	
BP Stacja Paliw al. 29 Listopada 125	24 h	
OBI al. Bora Komorowskiego 31	24 h	
Comarch al. Jana Pawła II 41e	Pn.-pt. 8:00-18:00	
Alior Bank al. Kijowska/ Kazimierza Wielkiego	24 h	
Słup Clear Channel al. Mickiewicza (Akademia Rolnicza)	24 h	
PSS Kraków al. Pokoju 20	24 h	
Raiffeisen Bank Polska S.A. al. Pokoju 44	24 h	
Plaza Center Kraków al. Pokoju 44	24 h	
Praktiker al. Pokoju 67	24 h	
CH M1 al. Pokoju 67	24 h	
Statoil, stacja paliw Armii Krajowej	24 h	
Raiffeisen Bank Polska S.A. Armii Krajowej 18	24 h	
Market Punkt Bałicka 7	24 h	
Firma AKT Bartla 20	24 h	
Delikatesy "Złoty Róg" Bałuckiego 9	24 h	




















Lokalizacja	Godziny otwarcia	Sieć
BP Stacja Paliw 29 Listopada 39B	24 h	
PSS Podwawelska 29-listopada 57/59	24 h	
BP Stacja Paliw al. 29 Listopada 125	24 h	
OBI al. Bora Komorowskiego 31	24 h	
Comarch al. Jana Pawła II 41e	Pn.-pt. 8:00-18:00	
Alior Bank al. Kijowska/ Kazimierza Wielkiego	24 h	
Skup Clear Channel al. Mickiewicza (Akademia Rolnicza)	24 h	
PSS Kraków al. Pokoju 20	24 h	
Raiffeisen Bank Polska S.A. al. Pokoju 44	24 h	
Plaza Center Kraków al. Pokoju 44	24 h	
Praktiker al. Pokoju 67	24 h	
CH M1 al. Pokoju 67	24 h	
Statoil, stacja paliw Armii Krajowej	24 h	
Raiffeisen Bank Polska S.A. Armii Krajowej 18	24 h	
Market Punkt Bałicka 7	24 h	
Firma AKT Bartla 20	24 h	
Delikatesy "Złoty Róg" Bałuckiego 9	24 h	


















```
CREATE TABLE `bankomaty` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `lat` float DEFAULT NULL,
  `lon` float DEFAULT NULL,
  `working_hours` varchar(10) NOT NULL,
  `network` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```





Lokalizacja	Godziny otwarcia	Sieć
BP Stacja Paliw 29 Listopada 39B	24 h	
PSS Podwawelska 29-listopada 57/59	24 h	
BP Stacja Paliw al. 29 Listopada 125	24 h	
OBI al. Bora Komorowskiego 31	24 h	
Comarch al. Jana Pawła II 41e	Pn.-pt. 8:00-18:00	
Alior Bank al. Kijowska/ Kazimierza Wielkiego	24 h	
Skup Clear Channel al. Mickiewicza (Akademia Rolnicza)	24 h	
PSS Kraków al. Pokoju 20	24 h	
Raiffeisen Bank Polska S.A. al. Pokoju 44	24 h	
Plaza Center Kraków al. Pokoju 44	24 h	
Praktiker al. Pokoju 67	24 h	
CH M1 al. Pokoju 67	24 h	
Statoil, stacja paliw Armii Krajowej	24 h	
Raiffeisen Bank Polska S.A. Armii Krajowej 18	24 h	
Market Punkt Bałicka 7	24 h	
Firma AKT Bałtka 20	24 h	
Delikatesy "Złoty Róg" Bałuckiego 9	24 h	



Lokalizacja	Godziny otwarcia	Sieć
BP Stacja Paliw 29 Listopada 39B	24 h	
PSS Podwawelska 29-listopada 57/59	24 h	
BP Stacja Paliw al. 29 Listopada 125	24 h	
OBI al. Bora Komorowskiego 31	24 h	
Comarch al. Jana Pawła II 41e	Pn.-pt. 8:00-18:00	
Alior Bank al. Kijowska/ Kazimierza Wielkiego	24 h	
Skup Clear Channel al. Mickiewicza (Akademia Rolnicza)	24 h	
PSS Kraków al. Pokoju 20	24 h	
Raiffeisen Bank Polska S.A. al. Pokoju 44	24 h	
Plaza Center Kraków al. Pokoju 44	24 h	
Praktiker al. Pokoju 67	24 h	
CH M1 al. Pokoju 67	24 h	
Statoil, stacja paliw Armii Krajowej	24 h	
Raiffeisen Bank Polska S.A. Armii Krajowej 18	24 h	
Market Punkt Bałicka 7	24 h	
Firma AKT Bałtka 20	24 h	
Delikatesy "Złoty Róg" Bałuckiego 9	24 h	

**SELECT** id, lat, lon, network **FROM** bankomaty  
**WHERE**

lat > 50.068902 **AND** lat < 50.063255 **AND**  
lon > 19.913750 **AND** lon < 19.923878



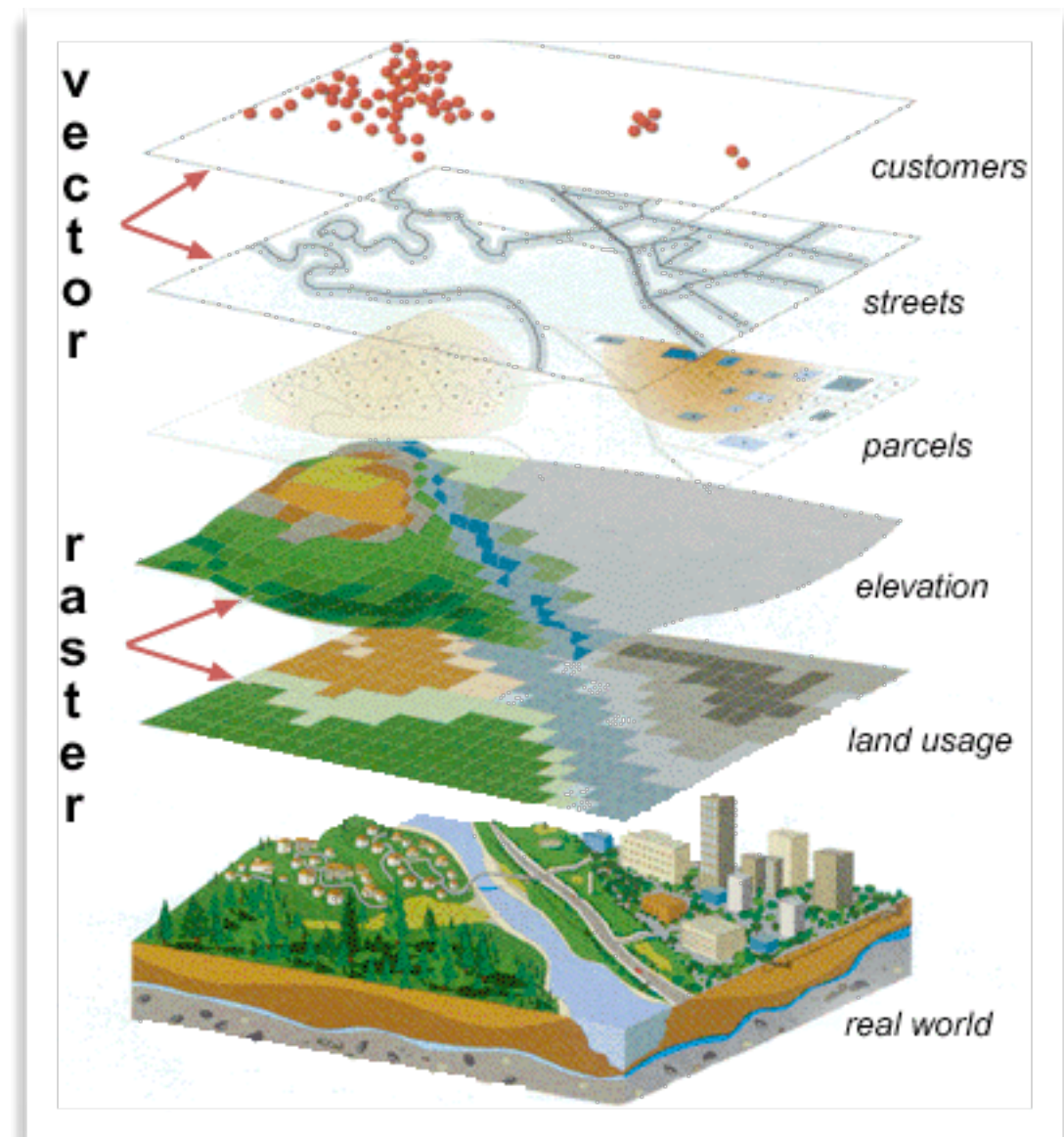




# Dane przestrzenne

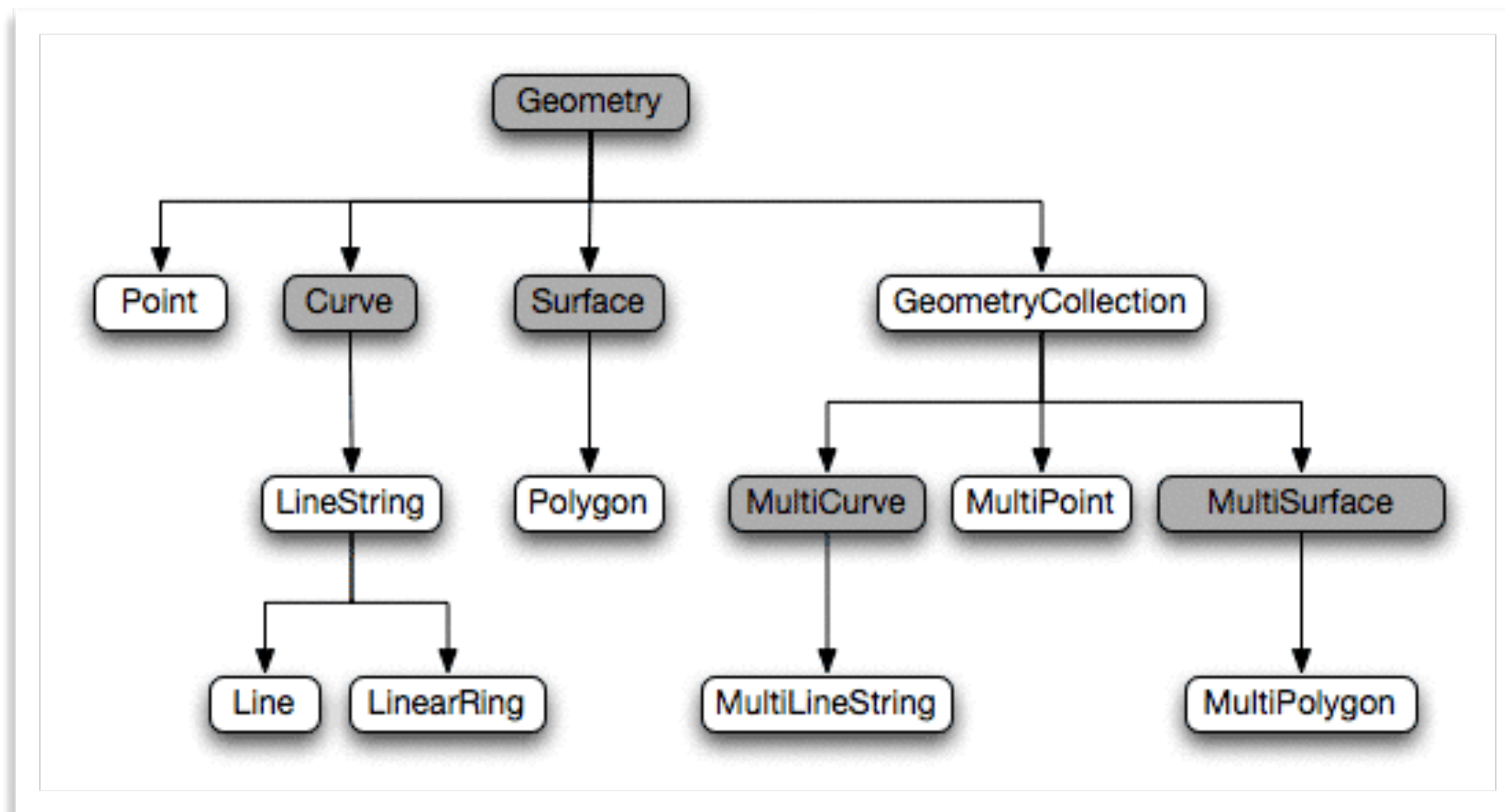
- *Ang.: geospatial data, georeferenced data*
- Zawierają:
  - komponent przestrzenny:  
lokalizacja, kształt
  - komponent atrybutowy: cechy

- **Model rastrowy:**
  - powierzchnia jest dzielona na siatkę równych pól/pikseli,
  - każdemu polu przypisujemy wartość.
- **Model wektorowy:**
  - obiekty opisywane przy pomocy podstawowych kształtów geometrycznych (punkt, linia, wielokąt).



źródło: National Coastal Data Development Center

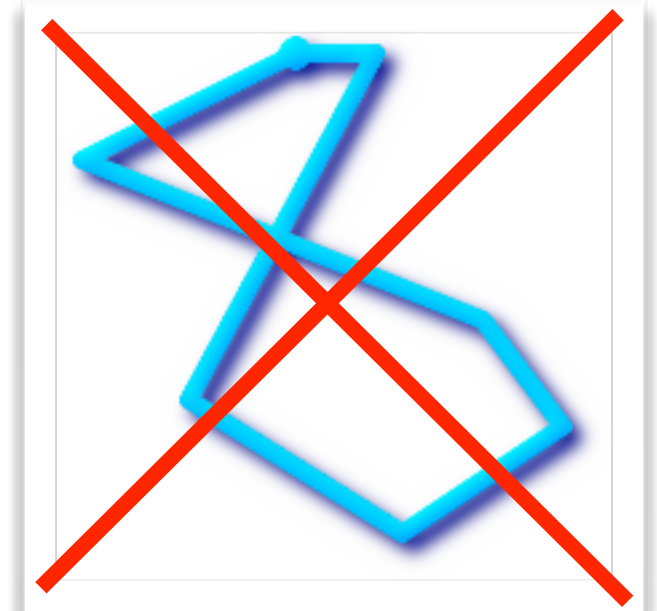
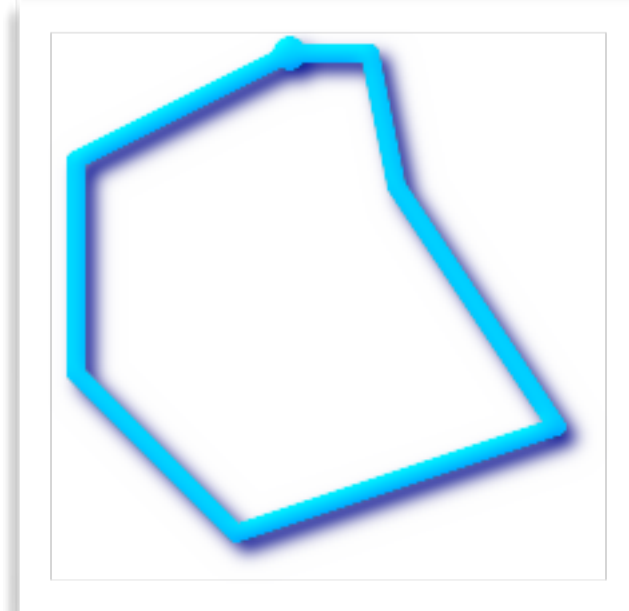
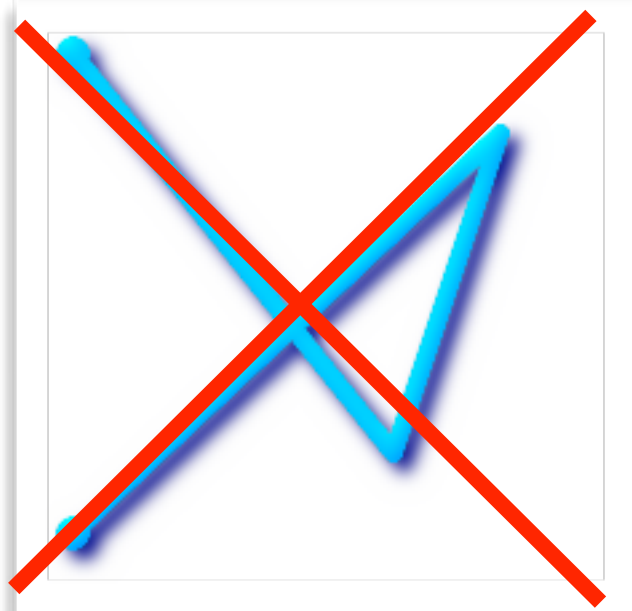
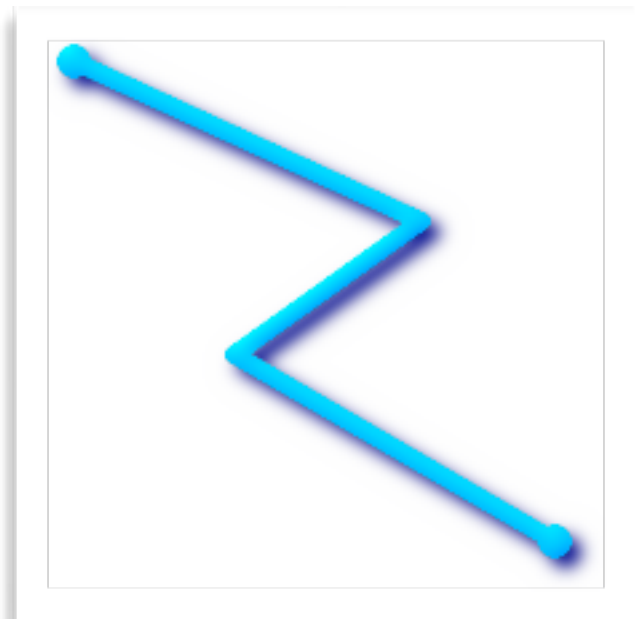
# OGC Geometry Model

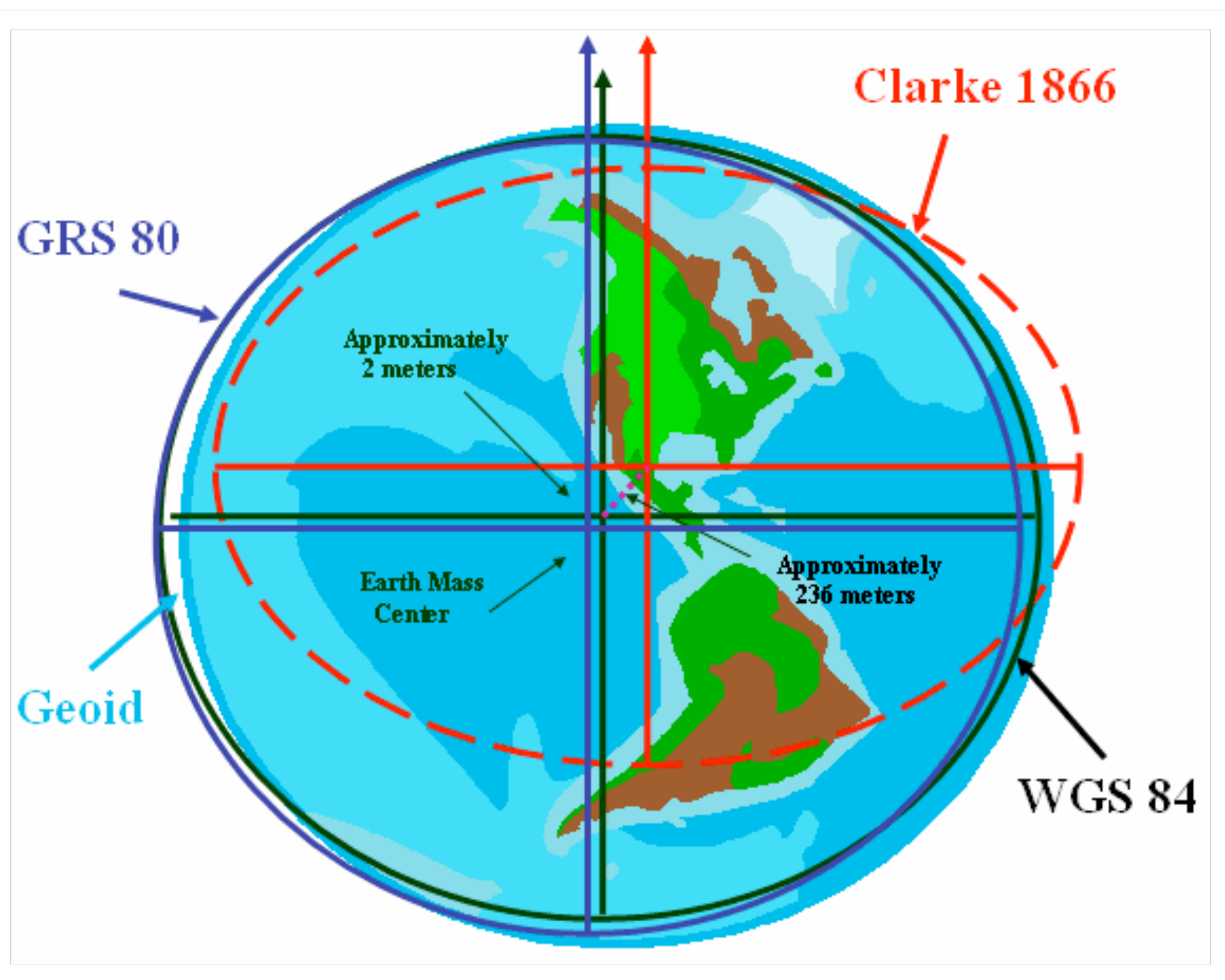


źródło: MySQL DevCenter



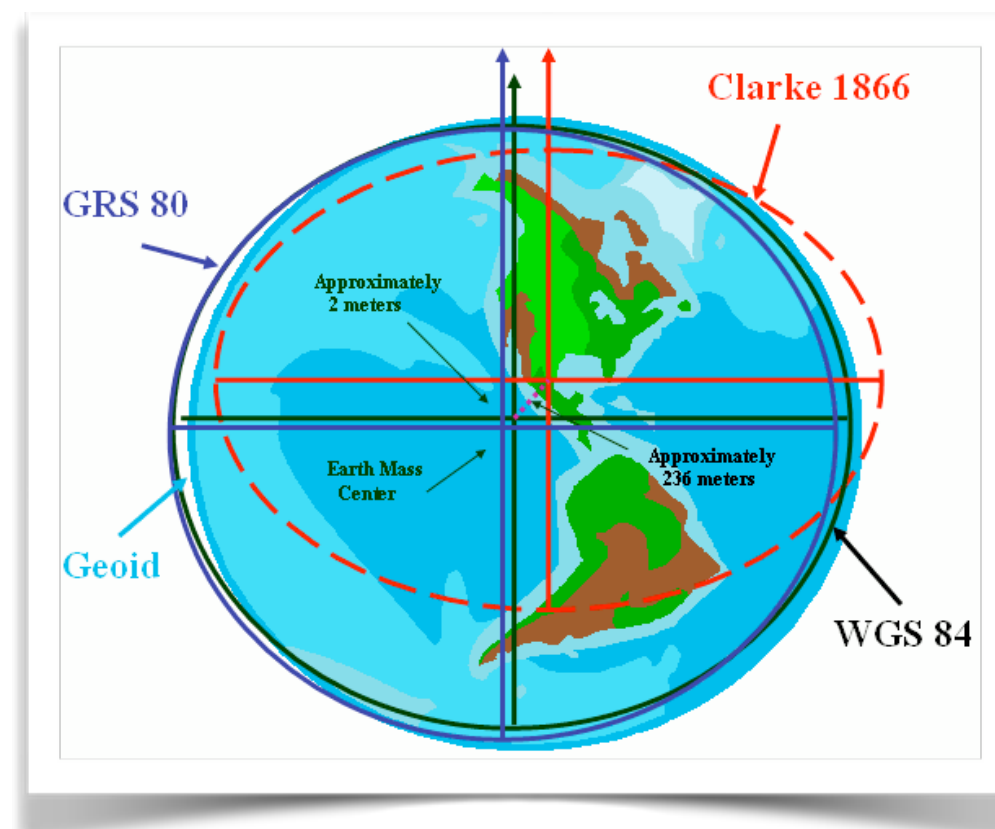
# Typy danych – *linestring*





# Układy odniesienia

- Kształt ziemi opisywany jest przy pomocy różnych układów współrzędnych.
- Klasyfikacją zajmuje się *European Petroleum Survey Group* (kody EPSG).
- WGS-84 (EPSG:4326) – długość/szerokość geograficzna.
- Spherical Mercator (EPSG:3785, chociaż często był oznaczany EPSG:900913) – współrzędne X/Y w metrach, niestety mało dokładny, używany głównie do wizualizacji.



# Formaty wymiany danych

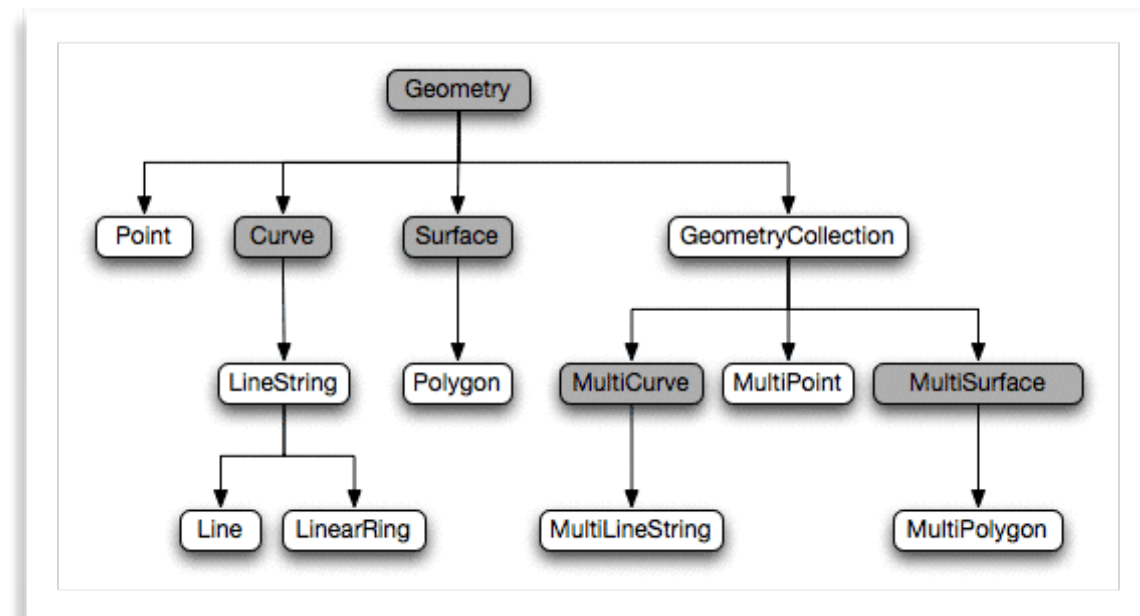
- *Geography Markup Language (GML)* – zdefiniowany przez OGC format XML do danych wektorowych i rastrowych; nieco „ciężki” do większości zastosowań (szczególnie webowych),
- *Well-Known Text (WKT)* i *Well-Known Binary (WKB)* – para prostych formatów zapisu danych o elementach przestrzennych,
- *Polish Map Format (PFM)* – format tekstowy, używany jako źródło map Garmin (IMG), a także w MapCenter,
- *OSM XML* – format używany przez OpenStreetMap,
- *GeoJSON* – rozszerzenie formatu JSON dla danych geograficznych.



# Well-Known Text

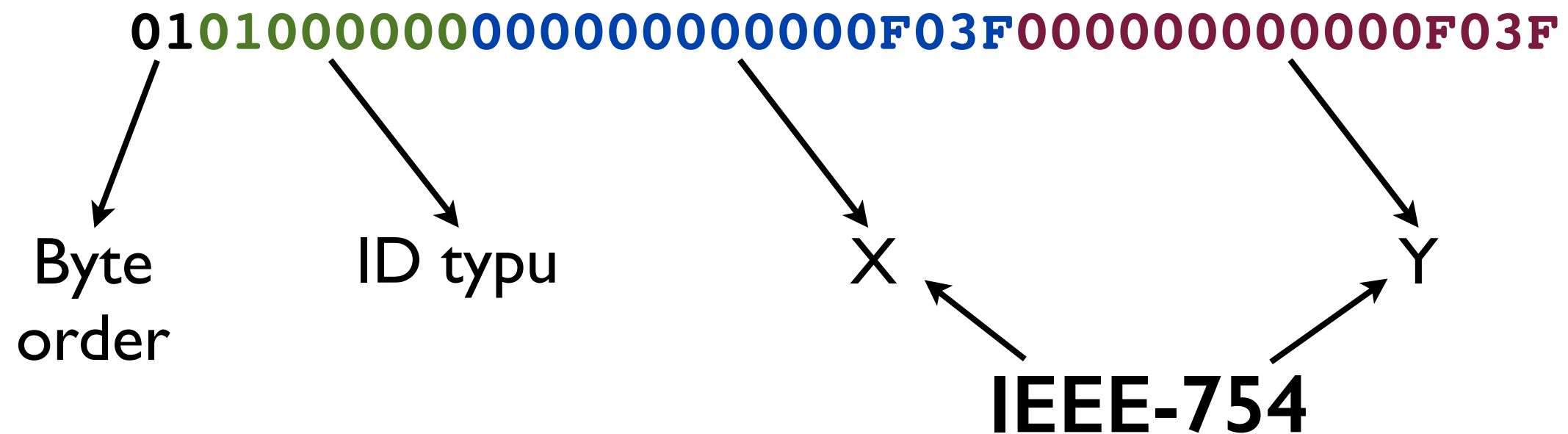
- Format tekstowy, używany często w zapytaniach SQL.

- `POINT(15 20)`
- `LINESTRING(0 0, 10 10, 20 25, 50 60)`
- `POLYGON((0 0,10 0,10 10,0 10,0 0),  
(5 5,7 5,7 7,5 7, 5 5))`
- `MULTIPOINT(0 0, 20 20, 60 60)`
- `MULTILINESTRING((10 10, 20 20), (15  
15, 30 15))`
- `MULTIPOLYGON(((0 0,10 0,10 10,0 10,0  
0)), ((5 5,7 5,7 7,5 7, 5 5)))`
- `GEOMETRYCOLLECTION(POINT(10 10),  
POINT(30 30), LINESTRING(15 15, 20  
20))`



# Well-Known Binary

- Binarny odpowiednik WKT.



# GML

```
<gml:Polygon>
  <gml:outerBoundaryIs>
    <gml:LinearRing>
      <gml:coordinates>0,0 100,0 100,100 0,100 0,0</
gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>
<gml:Point>
  <gml:coordinates>100,200</gml:coordinates>
</gml:Point>
<gml:LineString>
  <gml:coordinates>100,200 150,300</gml:coordinates>
</gml:LineString>
```

# Polish Map Format

```
[ POLYLINE ]  
RoadID=6091  
Type=0x6  
Label=MIECHOWSKA  
EndLevel=1  
Data0=(50.065710,19.913970),(50.066280,19.914050),  
      (50.066950,19.914170),(50.067440,19.914370),  
      (50.067630,19.914450)  
Nod1=0,14093,0  
Nod2=1,14176,0  
Nod3=3,14341,0  
Nod4=4,14372,0  
Routeparam=2,0,0,0,0,0,0,0,0,0,0,0  
[ END ]
```



# OpenStreetMap XML

```
<node id="273470204" lat="50.0679716" lon="19.9312968"
version="1" changeset="302227" user="Rafal Olearski"
uid="12349" visible="true"
timestamp="2008-06-26T19:21:27Z"/>
<way id="25117187" visible="true"
timestamp="2008-06-26T19:21:30Z" version="1"
changeset="302227" user="Rafal Olearski" uid="12349">
  <nd ref="273470204"/>
  <nd ref="273470207"/>
  <nd ref="262201253"/>
  <tag k="name" v="Kremerowska"/>
  <tag k="highway" v="residential"/>
  <tag k="oneway" v="yes"/>
</way>
```

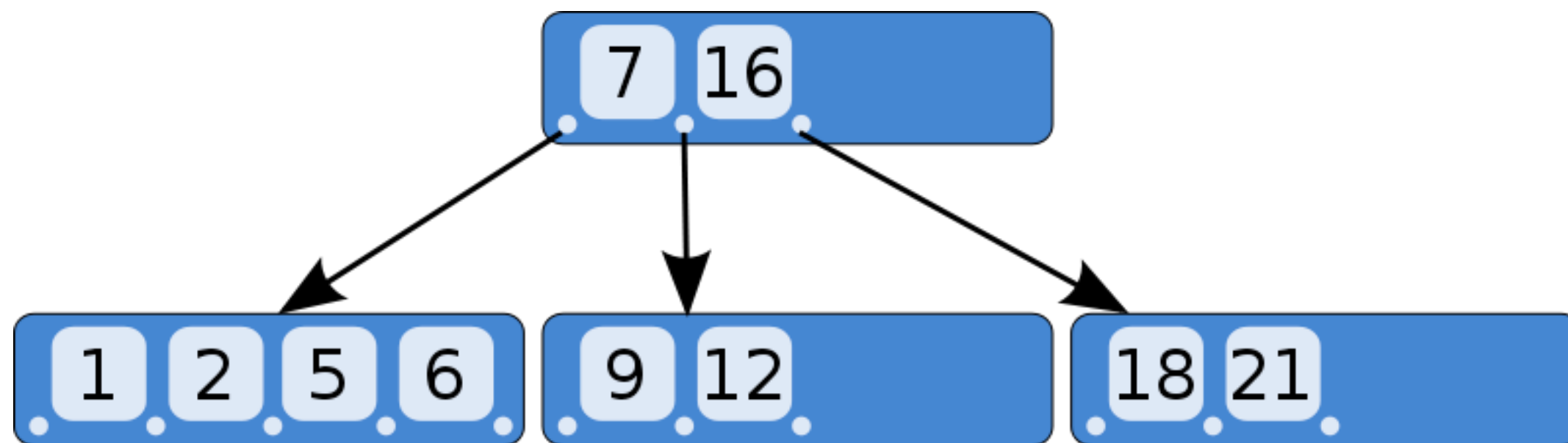
# GeoJSON

```
{ "type": "FeatureCollection",  
  "features": [  
    { "type": "Feature",  
      "geometry": {"type": "Point", "coordinates": [102.0, 0.5]},  
      "properties": {"prop0": "value0"}  
    },  
    { "type": "Feature",  
      "geometry": {  
        "type": "LineString",  
        "coordinates": [  
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]  
        ]  
      },  
      "properties": {  
        "prop0": "value0",  
        "prop1": 0.0  
      }  
    }  
  ]  
}
```

# Indeksy

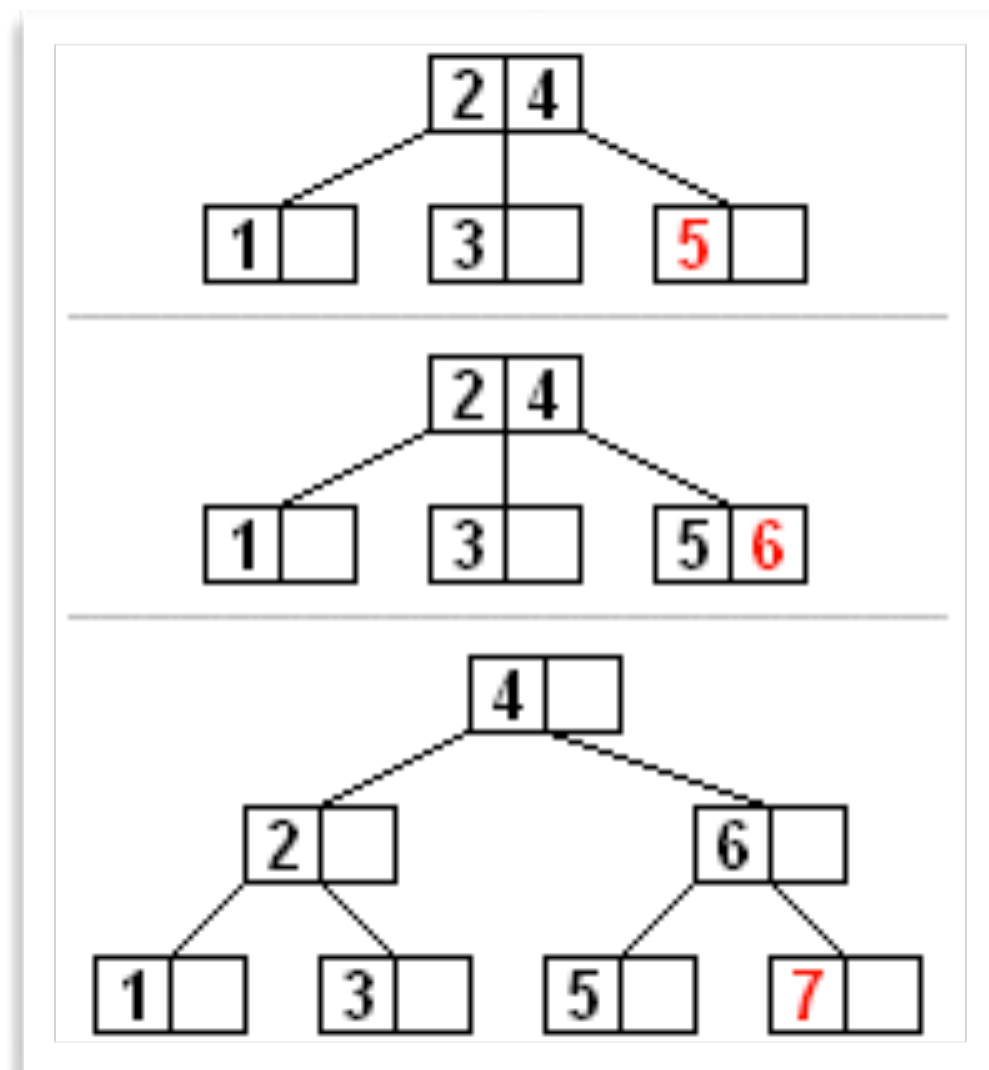
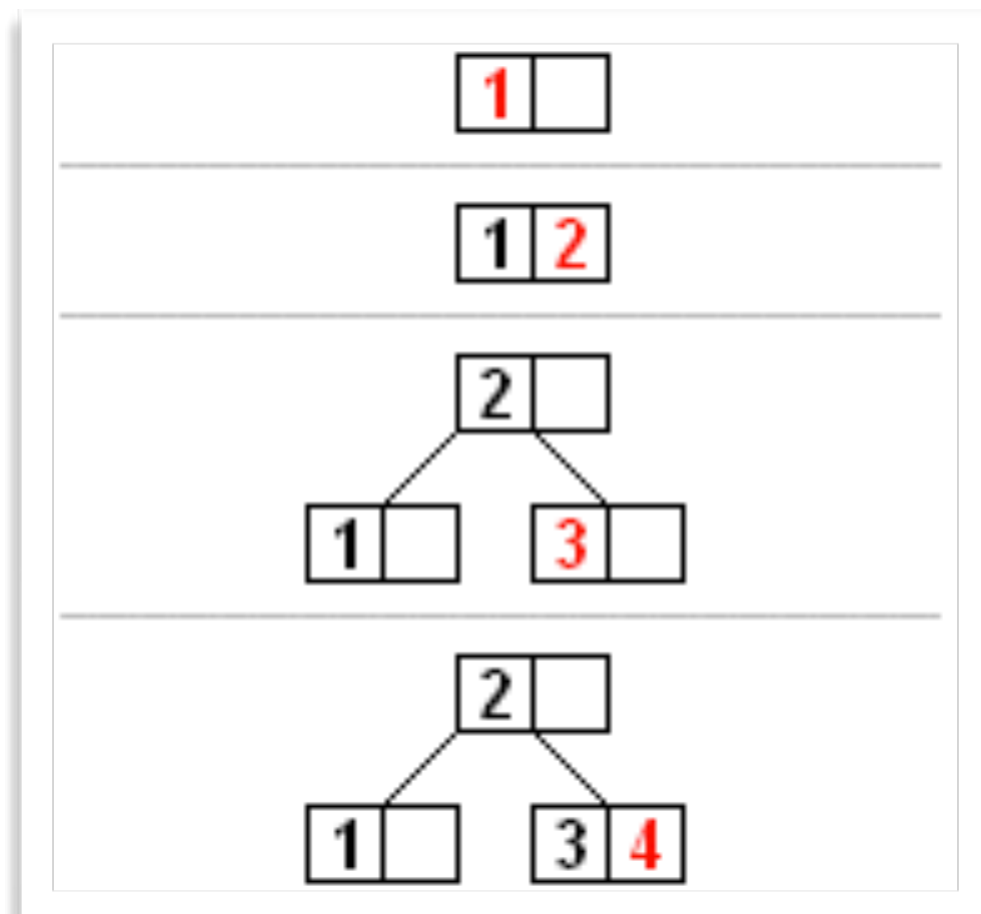
- Klasyczne indeksy: B-drzewa.
- Pozwalają na indeksowanie danych jednowymiarowych – dających się posortować ze względu na jeden atrybut.

# B-drzewa





# B-drzewa: wstawianie



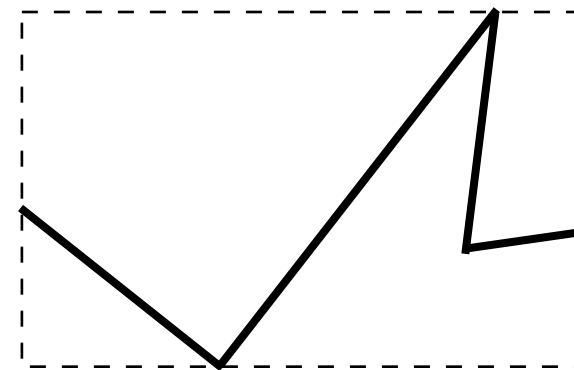
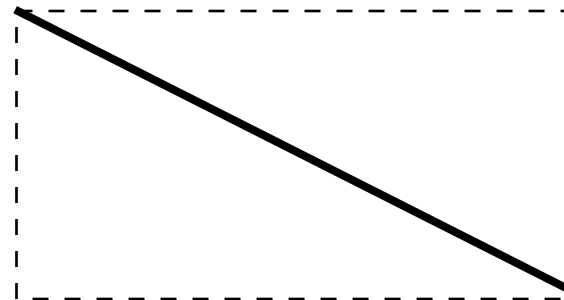
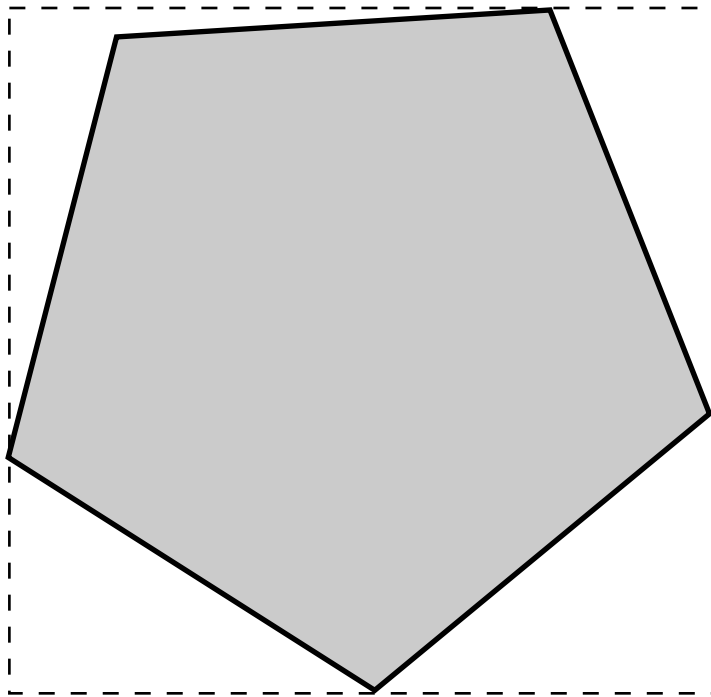
źródło: Wikipedia

# Indeksowanie danych wielowymiarowych

- Dane geograficzne (ale nie tylko) są co najmniej dwuwymiarowe, czasami – 3D.
- Można stworzyć dwa indeksy jednowymiarowe, ale to ogranicza elastyczność zapytań.

# MBR

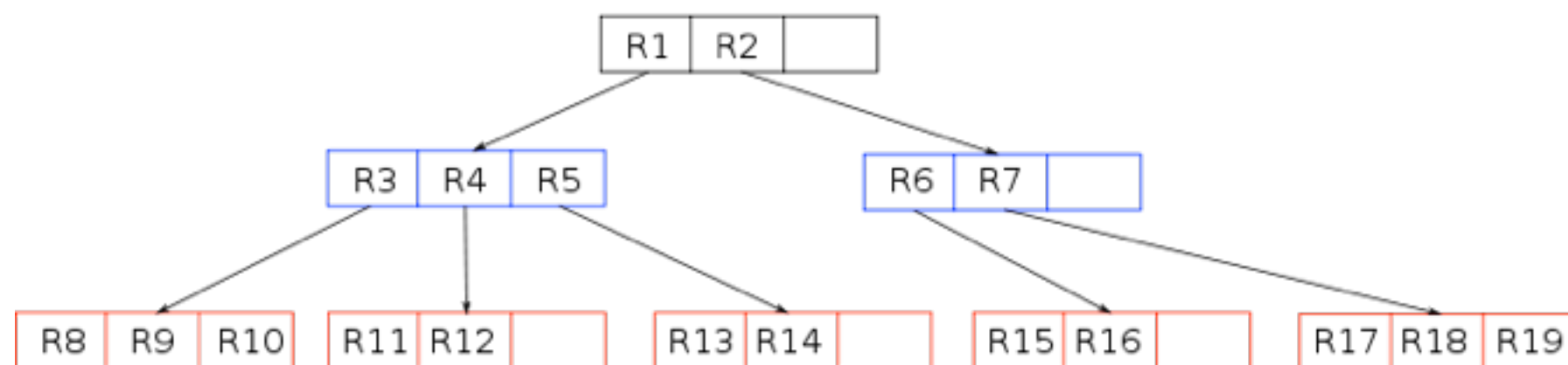
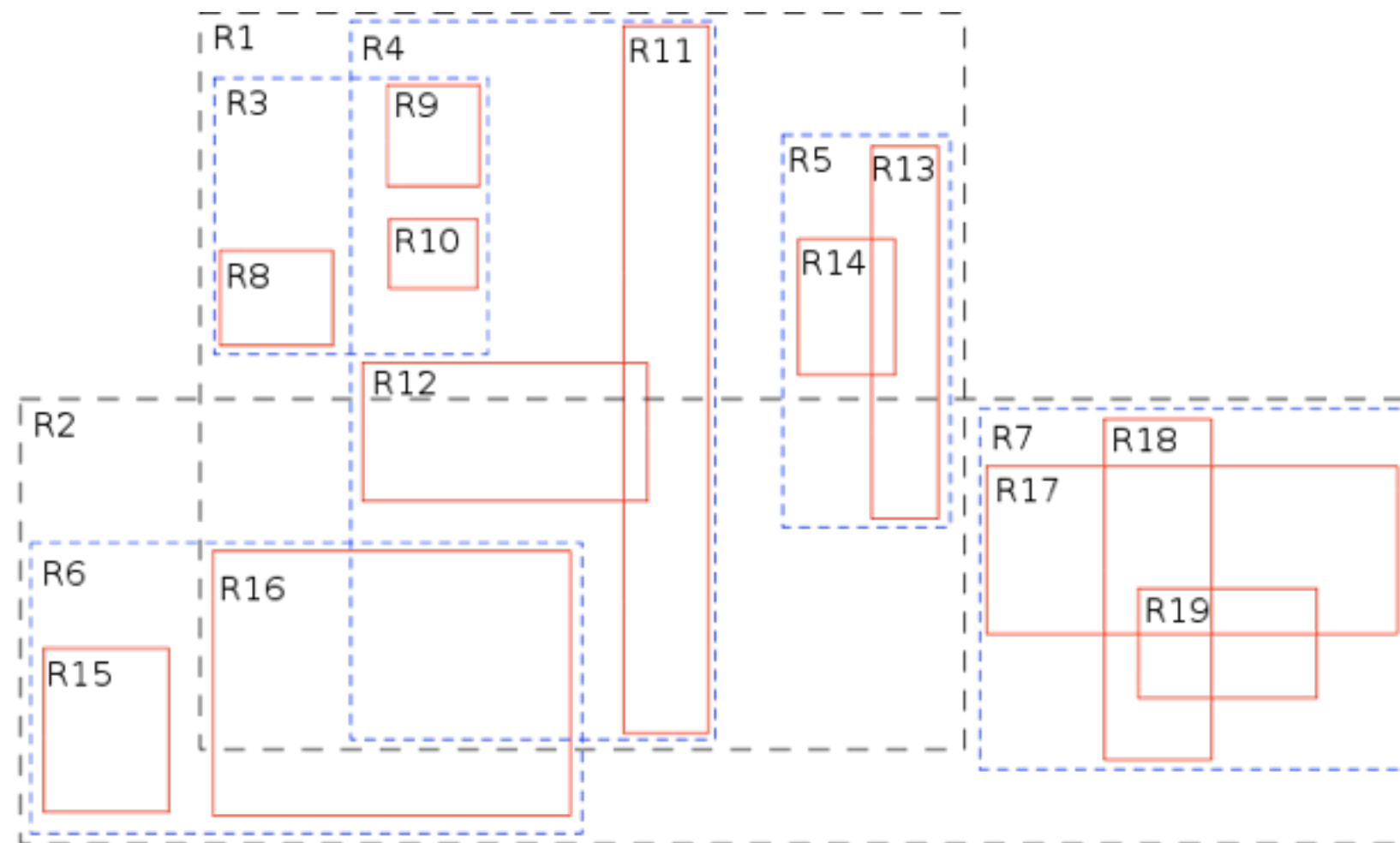
- MBR = minimalny region pokrywający (ang. *minimum bounding rectangle*).



# R-drzewa

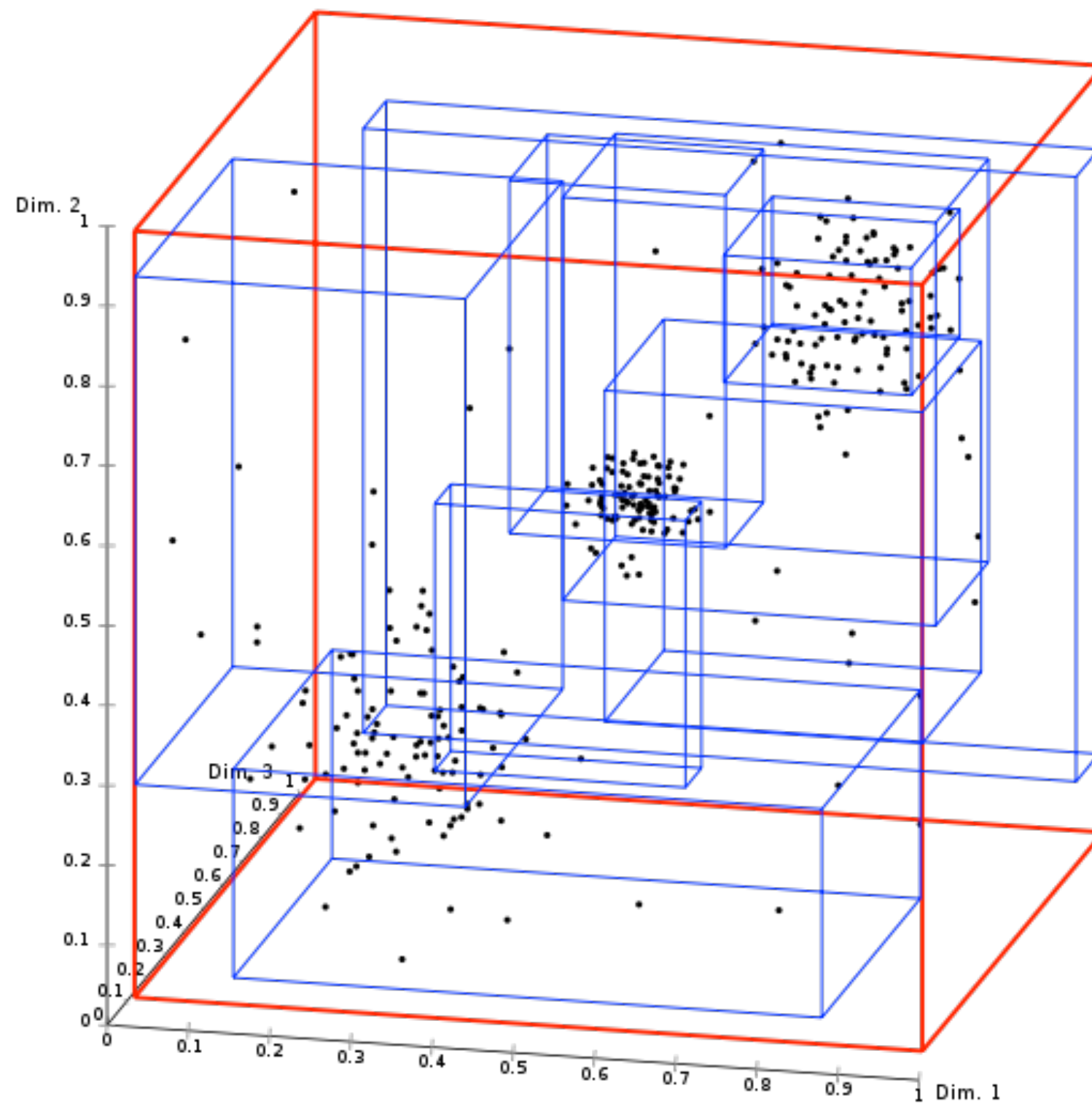
- Struktura podobna do B-drzew, ale służy do indeksowania danych wielowymiarowych.
- Dzielą przestrzeń na zagnieżdżone, nakładające się obszary – MBR.
- „Bliskie” obszary umieszczamy w jednym węźle drzewa (należy określić definicję „bliskości”).

źródło: Wikipedia





źródło: Wikipedia



# Geograficzne DB

- **Typy danych.** Przestrzenny DBMS musi obsługiwać typy danych pozwalające na zapisywanie informacji o elementach map.
- **Operacje.** Muszą istnieć funkcje przetwarzające obiekty przestrzenne (pole powierzchni, odległość, itd.).
- **Wymiana danych przestrzennych.** System musi mieć możliwość wymiany danych przestrzennych z systemami zewnętrznymi.
- **Indeksowanie danych przestrzennych.** Tradycyjne indeksy nie są zoptymalizowane pod kątem zapytań przestrzennych.

# Geograficzne DB

- MySQL od wersji 5.0 wspiera przestrzenne typy danych i funkcje (nie w 100% zgodne z OpenGIS).
- PostGIS – rozszerzenie do PostgreSQL, wprowadza zgodne z OpenGIS typy danych i funkcje analityczne.
- Oracle Spatial – zgodny z OpenGIS dodatek do Oracle.
- IBM DB2 Spatial Extender, Geodetic Extender.
- Dedykowane „bazy danych GIS”, np. GEODAS.

# MySQL – kolumny

- Kolumny dla przestrzennych typów danych tworzymy tak jak zwykle:

```
CREATE TABLE geom (g GEOMETRY);
```

```
ALTER TABLE geom ADD pt POINT;
```

```
ALTER TABLE geom DROP pt;
```

# MySQL – wartości

- Wartości przestrzenne uzyskiwane z WKT lub WKB przy pomocy dedykowanych funkcji, m.in.:

`GeomFromText, GeomCollFromText,  
LineFromText, PolyFromText,  
PointFromText,  
GeomFromWKB, ...`

# MySQL – wartości

- Istnieją też konstruktory bezpośrednie:  
`Point`, `LineString`, `Polygon`,  
`MultiPoint`, ...

```
INSERT INTO t1 (pt_col) VALUES (Point(1,2));
```



# MySQL – wartości

- Istnieją też konstruktory bezpośrednie:  
`Point`, `LineString`, `Polygon`,  
`MultiPoint`, ...

```
INSERT INTO t1 (pt_col) VALUES (GeomFromWKB (Point (1,2)) );
```

(dla wersji wcześniejszych niż 5.4.2)

# MySQL – dodawanie danych

- Przykłady dodawania danych przestrzennych:

```
INSERT INTO geom VALUES (GeomFromText ('POINT(1 1)')) ;
```

```
SET @g = 'POINT(1 1)';
```

```
INSERT INTO geom VALUES (GeomFromText (@g)) ;
```

# MySQL – dodawanie danych

- Przykłady dodawania danych przestrzennych:

```
SET @g = 'LINESTRING(0 0,1 1,2 2)';  
INSERT INTO geom VALUES (GeomFromText(@g));
```

```
SET @g = 'POLYGON((0 0,10 0,10 10,0 10,0 0),  
(5 5,7 5,7 7,5 7, 5 5))';  
INSERT INTO geom VALUES (GeomFromText(@g));
```

```
SET @g =  
'GEOMETRYCOLLECTION(POINT(1 1),  
LINESTRING(0 0,1 1,2 2,3 3,4 4))';  
INSERT INTO geom VALUES (GeomFromText(@g));
```

# MySQL – pobieranie danych

- Dane przestrzenne pobieramy przy pomocy funkcji `AsText` i `AsBinary`, lub w formacie wewnętrznym.

```
SELECT AsText(g) FROM geom;
```

```
SELECT AsBinary(g) FROM geom;
```

```
CREATE TABLE geom2 (g GEOMETRY) SELECT g FROM geom;
```

# MySQL – funkcje

- Konstruktory tworzą nowe geometrie na podstawie istniejących.
- Wynikiem działania operatorów również mogą być geometrie.
- Relacje mogą być analizowane dla MBR lub dla samych geometrii.

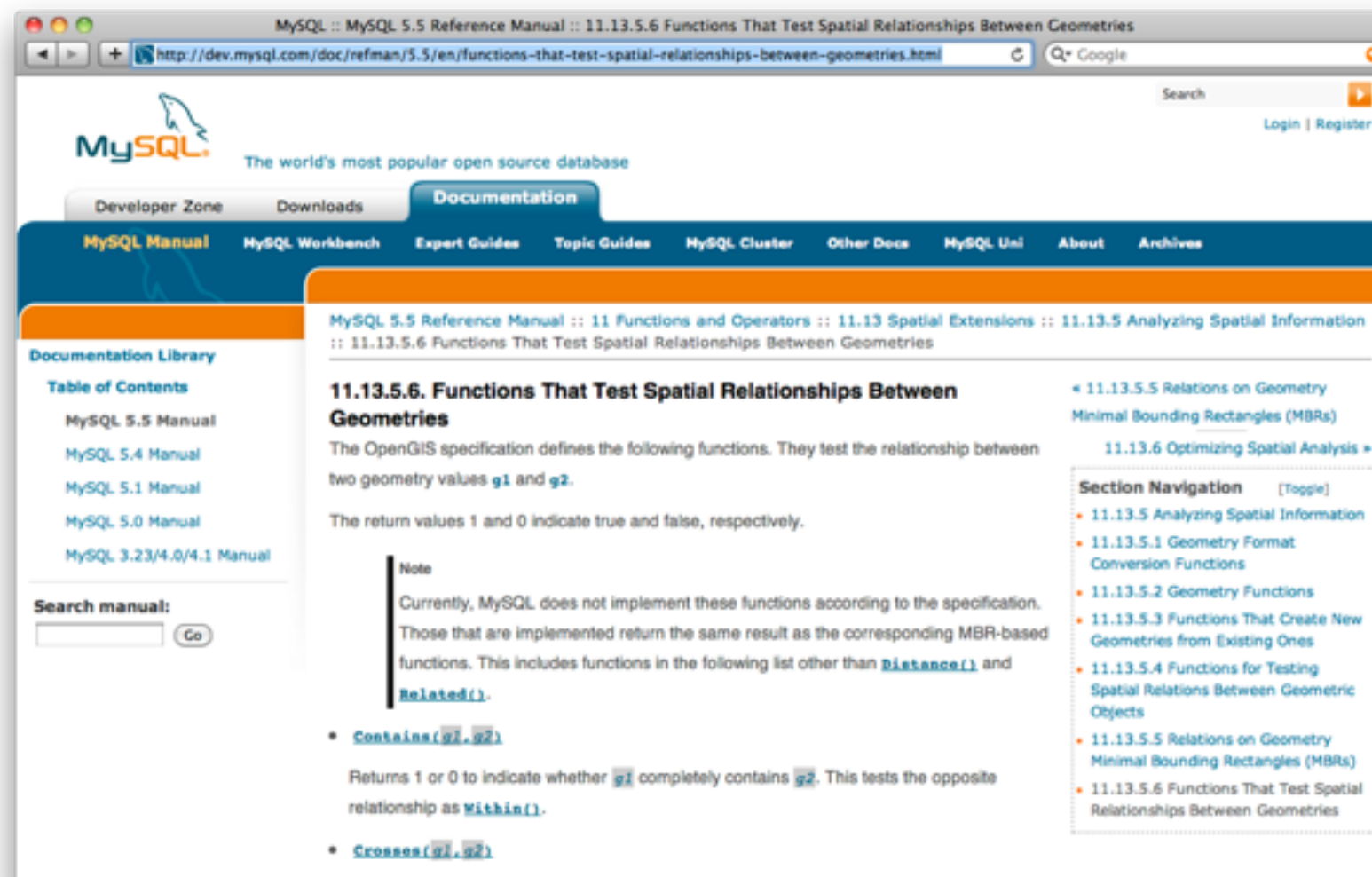
# MySQL – funkcje

- Konstruktory tworzą nowe geometrie na podstawie istniejących.
- Wynikiem działania operatorów również mogą być geometrie.
- Relacje mogą być analizowane dla MBR lub dla samych geometrii.



# MySQL – funkcje

- Relacje mogą być analizowane dla MBR lub dla samych geometrii.



# MySQL – indeksy

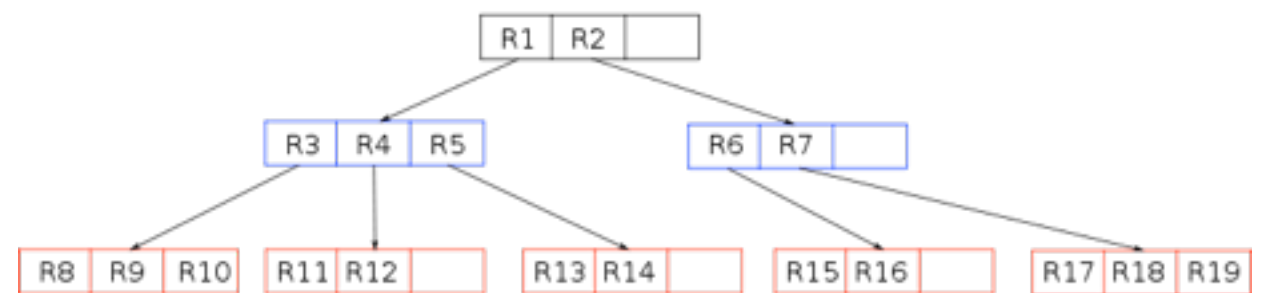
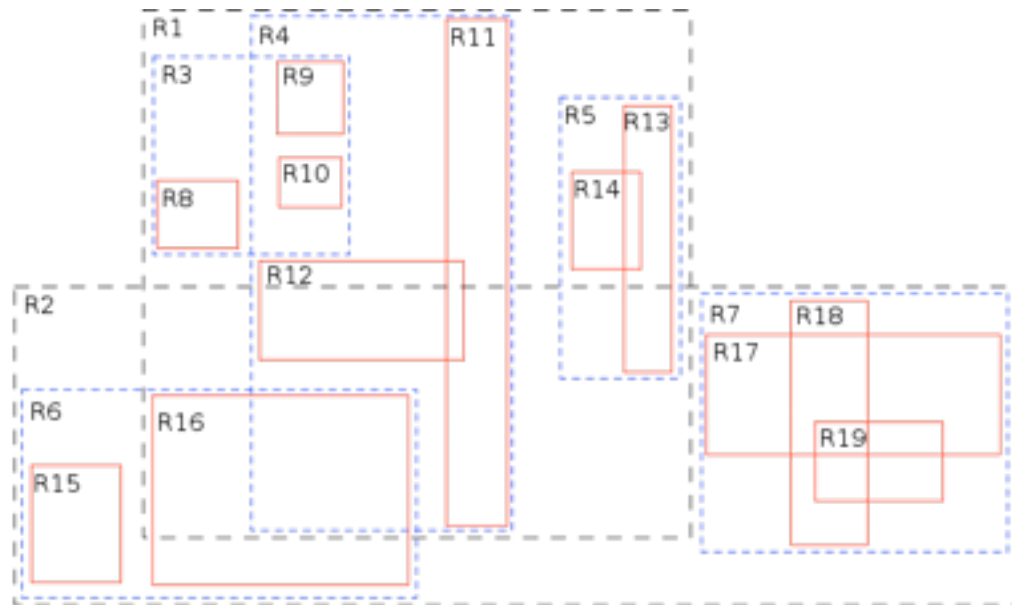
- Klasyczne indeksy słabo się nadają do danych o geometriach.
- MySQL udostępnia indeksy przestrzenne oparte o *r-drzewa* oraz MBR.
- Indeksy przestrzenne tworzy się dodając słowo kluczowe SPATIAL; są dostępne tylko dla silnika *MyISAM*.

# MySQL – indeksy

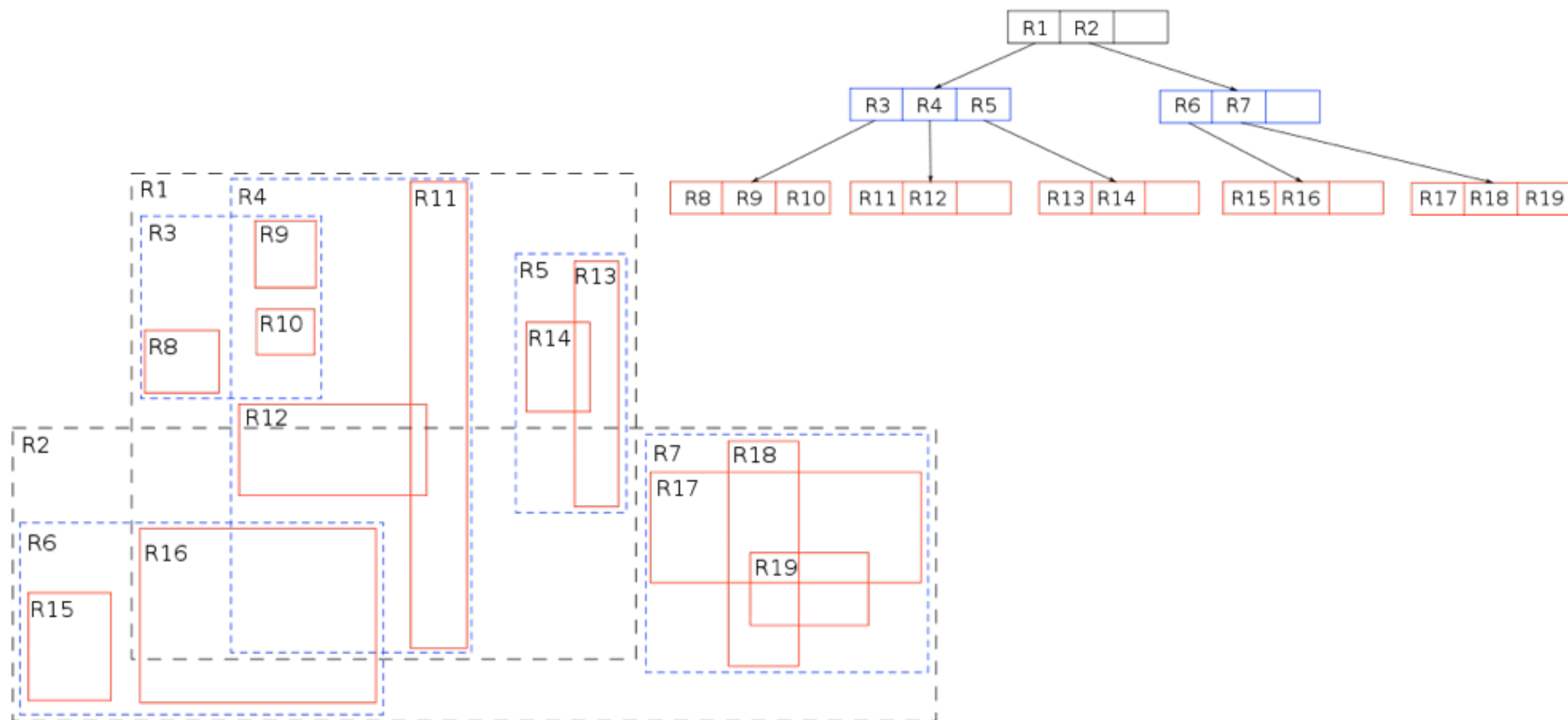
```
CREATE TABLE geom (g GEOMETRY NOT NULL, SPATIAL INDEX (g)) ;
```

```
ALTER TABLE geom ADD SPATIAL INDEX (g) ;
```

```
CREATE SPATIAL INDEX sp_index ON geom (g) ;
```



# MySQL – indeksy

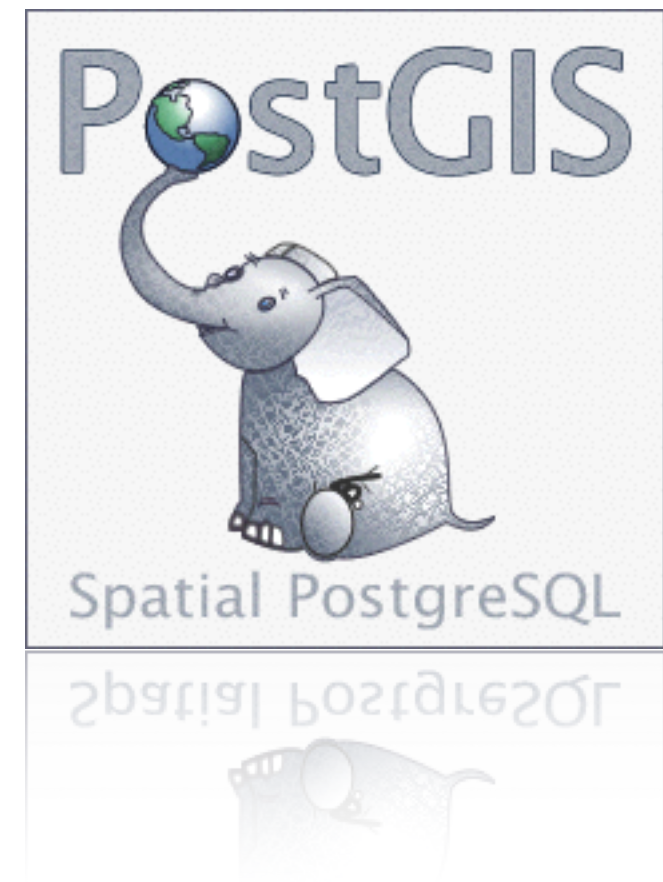


# MySQL – podsumowanie

- Szeroka dostępność, brak konieczności instalowania dodatków.
- W dużej części zgodna ze specyfikacją OpenGIS.
- Dostępność indeksów przestrzennych.
- Analiza ograniczona do MBR.

# PostGIS

- Dodatek do PostgreSQL, zgodny ze specyfikacją OGC.
- Licencja GPL.
- Dostępny na wszystkie platformy, na które jest PostgreSQL.





# PostGIS – formaty

- WKT/WKB.
- EWKT/EWKB – rozszerzają do geometrii 3D, pozwalają na osadzenie SRID.
- SQL-MM Part 3.

# PostGIS – struktura

- Systemowe tabele `spatial_ref_sys` i `geometry_columns` przechowują dane o układach odniesienia i kolumnach przestrzennych.
- Kolumnę przestrzenną dodajemy do tabeli przy pomocy polecenia *AddGeometryColumn*:

```
SELECT AddGeometryColumn( 'parks', 'park_geom',  
                           128, 'GEOMETRY', 2 );
```

# PostGIS – indeksy

- Metody indeksowania wspierane przez PostgreSQL:
  - b-drzewa,
  - r-drzewa,
  - GiST (*Generalized Search Trees*)

# PostGIS – indeksy

```
CREATE INDEX [indexname] ON [tablename]  
USING GIST ( [geometryfield] );
```

```
VACUUM ANALYZE [table_name] [column_name];
```

# PostGIS – indeksy

- GiST, w porównaniu do r-drzew, są „null-safe” – mogą indeksować kolumny z wartościami NULL.
- Pozwalają na indeksowanie obiektów większych niż 8KB (rozmiar strony PostgreSQL).

# PostGIS – funkcje

- Wejście/wyjście,
- Funkcje manipulacji,
- Analiza geometrii,
- Relacje i pomiary przestrzenne.

<http://postgis.refractory.net/documentation/manual-1.5/reference.html>

# PostGIS – przykłady

## **bc\_roads**

Column	Type	Description
gid	integer	Unique ID
name	character varying	Road Name
the_geom	geometry	Location Geometry (Linestring)

## **bc\_municipality**

Column	Type	Description
gid	integer	Unique ID
code	integer	Unique ID
name	character varying	City / Town Name
the_geom	geometry	Location Geometry (Polygon)



# PostGIS – przykłady

- Jaka jest łączna długość wszystkich dróg w kilometrach?

```
SELECT sum(ST_Length(the_geom)) / 1000 AS km_roads FROM bc_roads;
```

# PostGIS – przykłady

- Jaka jest powierzchnia miasta Prince George w hektarach?

```
SELECT
    ST_Area(the_geom)/10000 AS hectares
FROM bc_municipality
WHERE name = 'PRINCE GEORGE';
```

# PostGIS – przykłady

- Które miasto jest największe (wynik w hektarach)?

```
SELECT
    name,
    ST_Area(the_geom) / 10000 AS hectares
FROM
    bc_municipality
ORDER BY hectares DESC
LIMIT 1;
```

# PostGIS – przykłady

- Jaka jest łączna długość dróg wewnętrznych w każdym mieście?

```
SELECT
    m.name,
    sum(ST_Length(r.the_geom)) / 1000 as
roads_km
FROM
    bc_roads AS r,
    bc_municipality AS m
WHERE
    ST_Contains(m.the_geom, r.the_geom)
GROUP BY m.name
ORDER BY roads_km;
```

# PostGIS – przykłady

- Utwórz nową tabelę zawierającą wszystkie drogi w Prince George.

```
CREATE TABLE pg_roads as
SELECT
    ST_Intersection(r.the_geom, m.the_geom) AS intersection_geom,
    ST_Length(r.the_geom) AS rd_orig_length,
    r.*
FROM
    bc_roads AS r,
    bc_municipality AS m
WHERE m.name = 'PRINCE GEORGE' AND
    ST_Intersects(r.the_geom, m.the_geom);
```

# GIS i WWW

- Aplikacje webowe zawierające mapy stają się coraz popularniejsze.
- Funkcjonalność realizowana najczęściej przez API w postaci biblioteki JS.
- Jako *backend* najlepiej sprawdza się baza danych z funkcjonalnością GIS.

# Google Maps API

- Pozwala na umieszczenie mapy Google w wybranym elemencie, najczęściej typu DIV.
- Na mapie można umieszczać markery (najczęściej wykorzystywana funkcjonalność), a także linie, łamane i wielokąty.
- Przykłady.

# Inne API

- Yahoo Maps
- Bing Maps
- Map24
- OpenLayers



# OpenLayers

- API podobne do Google Maps.
- Open-source, nie związane z określonym dostawcą map.
- Zbudowane w oparciu o warstwy, z możliwością dynamicznego przełączania.
- Przykłady.

# Źródła danych geograficznych

- Google Maps jest „standardem” do prezentacji danych geograficznych aplikacji na tle mapy.
- Czy jesteśmy w stanie „wyciągnąć” jakieś dane z Google Maps?

# OpenStreetMap

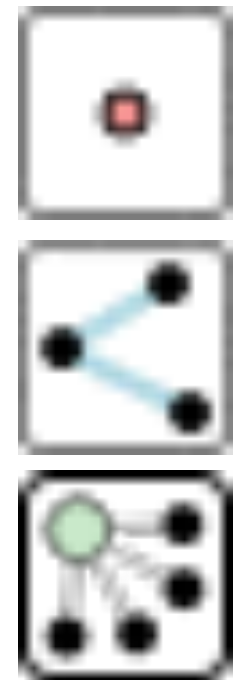
- Mapa Świata, tworzona jak Wikipedia.
- Centralne repozytorium, działa w oparciu o MySQL (!), niezbyt dla nas interesujące.
- Interfejs webowy, pozwala na przeglądanie, edycję, pobieranie.
- Narzędzia dodatkowe: konwertery, edytory (bardziej zaawansowane).

# OSM: WWW

- <http://www.openstreetmap.org>
- Wykorzystuje bibliotekę OpenLayers.
- Edytor oparty o Flash – Potlatch.
- Eksport:
  - HTML do osadzenia na stronie,
  - obrazek z widokiem mapy (skala do wyboru),
  - OSM XML.

# OSM: model danych

- Podstawowe elementy:
  - punkty (node),
  - drogi (way),
  - relacje (relation).
- Parametry każdego elementu (np. kategoria drogi, rodzaj budynku użyteczności publicznej) przypisywane przy pomocy par klucz/wartość.



# OSM: formaty danych

- Zwyczajowo: OSM XML, do pobrania np. przez WWW.
- Nowy format binarny: PBF.

# OSM: elementy mapy

- Klasyfikacja elementów mapy z przykładami przechowywana jest na stronie *Map Features*:
  - [http://wiki.openstreetmap.org/wiki/Map\\_Features](http://wiki.openstreetmap.org/wiki/Map_Features)
  - [http://wiki.openstreetmap.org/wiki/Pl:Map\\_Features](http://wiki.openstreetmap.org/wiki/Pl:Map_Features)

# OSM: changesets

- OSM przechowuje historię zmian.
- Changesety opisują zmiany wprowadzone w danej sesji edycyjnej.
- Operacje: *add, modify, delete*.
- Każdy *changeset* ma swój *bounding box*.



# OSM: pobieranie danych

- Istnieje możliwość pobierania XML dla wybranego obszaru przez interfejs WWW, ale tylko do 50 000 węzłów.
- Co potem?

# OSM: pobieranie danych

- Pobieramy całą kulę ziemską: [planet.osm](http://planet.osm.org/)
- Aktualnie ok. 250 GB (XML), 14 GB (PBF).
- Lepsze rozwiązanie: pobieramy wybrany obszar, np. z [GeoFabrik](http://geofabrik.de/)...
- ...i przycinamy przy pomocy Osmosis.

# OSM: narzędzia

- Edytory – np. JOSM.
- Przetwarzanie danych – Osmium, Osmosis.
- Konwersja – osm2pgsql, osm2pgrouting, ale również Osmosis.

# JOSM

- <http://josm.openstreetmap.de/>
- Java OpenStreetMap Editor
- Zaawansowany, wymaga nauki.

# Osmium

- Biblioteka C++/JavaScript do pracy z danymi OSM.
- Zapis/odczyt różnych formatów.
- Manipulacja danymi możliwa przy pomocy JavaScript (który korzysta z wbudowanych procedur C++).

# Osmosis

- Uniwersalny konwerter/manipulator, napisany w Javie.
- Import/eksport do/z bazy danych.
- Generowanie changesets na podstawie historii zmian w bazie, lub aplikowanie ich na istniejącej bazie.
- Przycinanie danych wg. *bounding box* lub wielokąta.