



# SQL/MM

---

SQL Multimedia and Application  
Packages

Tomáš Kuchař & Jiří Krůček



# Obsah

---

- **Úvod**
- Část 1: Framework
- Část 2: Full-Text
- Část 3: Spatial
- Část 4: General Purpose Facilities
- Část 5: Still Image
- Část 6: Data Mining



# Úvod > Motivace

---

- Motivace
  - SFQL (1991-1992)
    - Structured Full-text Query Language
    - Rozšíření SQL pro práci s full-textovými daty
    - Vytvořen „Full-textovou“ komunitou
    - Konflikt klíčových slov (CONTAINS)
      - Full-Text vs. Prostorová data
  - Možný konflikt klíčových slov je třeba řešit



# Úvod > Řešení

---

- SQL/MM (1999-2002)
  - Jde o řadu UDT a UDF dle SQL:1999
  - Okruhy působnosti
    - Full-textová data
    - Prostorová data
    - Obrázky (statické i videa)



# Obsah

---

- Úvod
- **Část 1: Framework**
- Část 2: Full-Text
- Část 3: Spatial
- Část 4: General Purpose Facilities
- Část 5: Still Image
- Část 6: Data Mining



# Framework

---

- Jednotlivé části SQL/MM jsou na sobě poměrně nezávislé
- Framework je část společná a závazná „zbytku světa“
- Poskytuje:
  - definici společného konceptu užitého v dalších částech SQL/MM
  - hlavní rysy přístupu k definici těchto částí



# Obsah

---

- Úvod
- Část 1: Framework
- Část 2: Full-Text
- Část 3: Spatial
- Část 4: General Purpose Facilities
- Část 5: Still Image
- Část 6: Data Mining



# Full-Text > Úvod

---

- „Full-Text“  $\approx$  textová data lišící se od běžných znakových řetězců
  - Obvykle delší záznamy
  - Specifické operace
  - Způsob indexování
    - Index slov v dokumentu
    - Index vzájemných vzdáleností slov a frází
    - ...





# Full-Text > Typ "FullText"

---

- Konstruktory
  - Řetězec znaků
  - Řetězec znaků + zadání jazyka
- Konverzi do běžných SQL znak. řetězců.
  - FullText\_to\_Character
- Vyhledávací metody
  - ano/ne (CONTAINS)
  - Rank (RANK)



# Full-Text > vyhledávání

---

- Vzorek (+ wildcards)
- Odvozená slova (STEMMED)
- Slova s podobným nebo stejným významem (THESAURUS, SYNONYM)
- Stejně znějící slova (SOUNDS LIKE)
- Dle pozice v textu (NEAR, ...)
- Dle konceptu textu (IS ABOUT)



## Full-Text > Podpora jazyků

---

- SQL/MM počítá především s podporou jazyků, kde je snadné výpočetně rozeznat jednotlivé tokeny.



# Full-Text > Příklad > tabulka

---

```
CREATE TABLE informace (  
    číslo_dokumentu    INTEGER,  
    dokument            FULLTEXT  
);
```



# Full-Text > Příklad > dotaz

---

```
SELECT číslo_dokumentu
FROM informace
WHERE dokument.CONTAINS
(
    'STEMMED FROM OF "standard"
IN SAME PARAGRAPH AS
SOUNDS LIKE "sequel"'
) = 1;
```



# Obsah

---

- Úvod
- Část 1: Framework
- Část 2: Full-Text
- **Část 3: Spatial**
- Část 4: General Purpose Facilities
- Část 5: Still Image
- Část 6: Data Mining



# Obsah

---

- Úvod
- Geometrický model
- Příklady
- Datový katalog
- Závěr



# Úvod

---





# Prostorové databáze

---

- Databáze, ve kterých jsou kombinována formátovaná data a prostorová data



# Využití prostorových databází

---

- Geografické informační systémy (GIS)
  - Plánování výstavby měst
  - Řízení dopravy
  - Mapování nalezišť nerostných surovin
  - Distribuční sítě
  - Bankovníctví
  - Pojišťovnictví



# Využití prostorových databází

---

- CAD/CAM (Computer-aided design and manufacturing)
  - Návrh integrovaných obvodů
  - Návrh mostu
  - Návrh motoru



# Co potřebujeme?

---

- Vhodný dotazovací jazyk
- Použití standardních dotazovacích jazyků je nevhodné
  - Neposkytují žádnou podporu pro dotazování nad prostorovými daty



# Řešení

---

- Nevyvíjet zcela nový dotazovací jazyk, ale využít jazyka SQL
- Proč?
  - SQL je databázový standard
  - V praxi se často vyskytují prostorová data společně s lexikálními daty

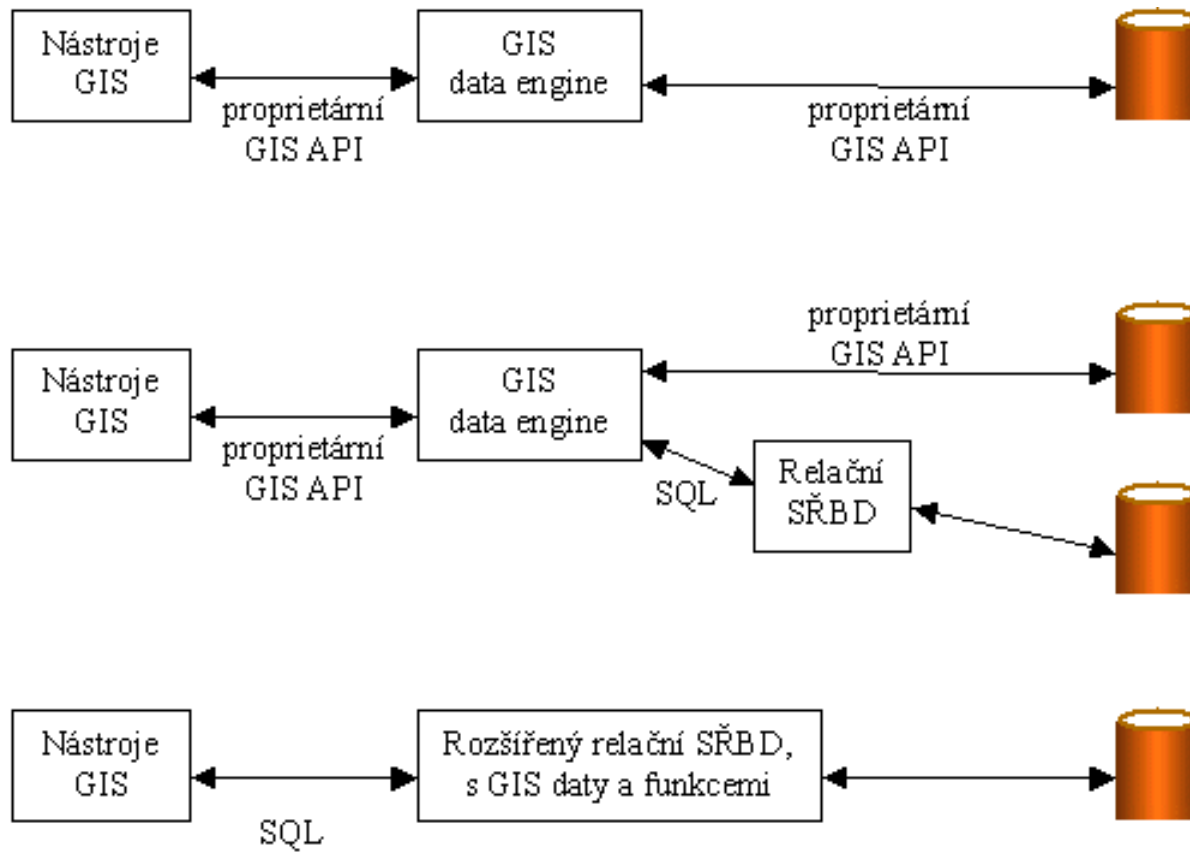


# Co musí jazyk splňovat?

---

- Musí umožnit formulování dotazů následujících typů
  - Dotazy výhradně na prostorové vlastnosti (Najdi všechna města rozdělená řekou)
  - Dotazy pouze na neprostorové vlastnosti (Kolik lidí žije v Ostravě?)
  - Kombinace předchozích typů (Vypiš jména všech sousedů parcely číslo 15 v Soukenické ulici)

# Historický vývoj GIS databází





# Historie ukládání prostorových dat v relačních databázích

---

- Prostorová data jsou uložena v tabulkách v 1NF
- Prostorová data jsou uložena jako nestructurované rozsáhlé binární objekty (BLOB) či jako “opaque types”
- Prostorová data jsou uložena v relační databázi rozšířené o typy prostorových dat
- Prostorová data jsou implementovaná pomocí uživatelsky definovaných typů





# Vznik SQL/MM (1999)

---

- SQL/MM:
  - Part 1: Framework
  - Part 2: Full-text
  - **Part 3: Spatial**
  - Part 5: Still images
  - Part 6: Data mining

# Vlivy na utváření SQL-MM

## Spatial



---

- OpenGIS konsorcium: OpenGIS Simple Features Specification (geometrický model)
- Standard SQL:1999 (objektová orientace)



# Co SQL/MM Spatial definuje?

---

- Ukládání, výběr, dotazování a aktualizaci jednoduchých prostorových objektů
- Re prezentaci prostorových objektů pomocí prostorových datových typů (Spatial types)
- Funkce pro práci s prostorovými objekty



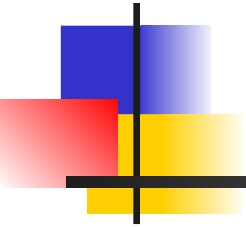
# Dělení SQL/MM Spatial

---

- Standard SQL/MM Spatial je rozdělen do několika klauzulí
- Jednotlivé klauzule se zabývají:
  - Prostorovými datovými typy a jejich metodami (Spatial types)
  - Datovým katalogem (Information Schema)
  - ...

# Datové typy pro reprezentaci prostorových objektů

---



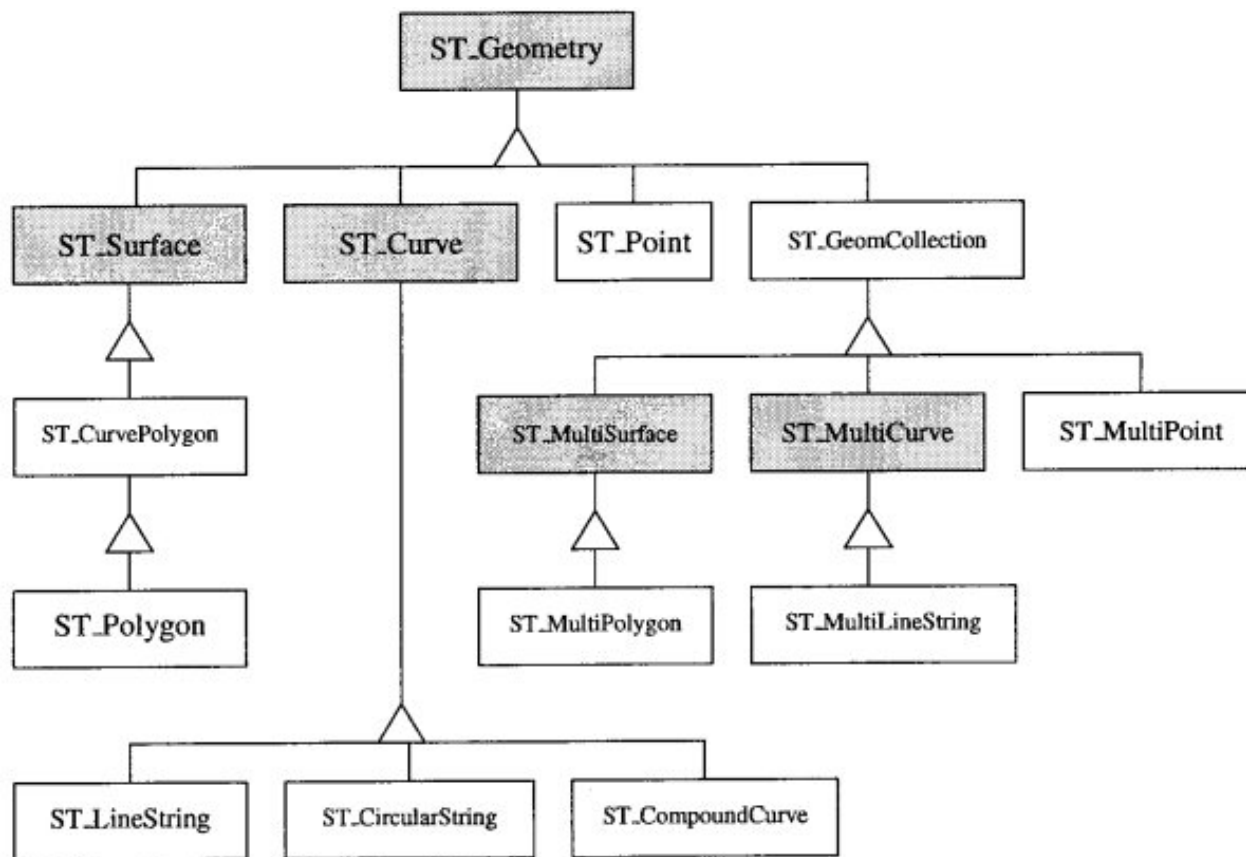


# Prostorové objekty

---

- Body:
  - pouliční lampa, lavička...
- Křivky:
  - koryto řeky, hranice pobřeží...
- Polygony:
  - půdorys budov, území států, pozemky...

# SQL geometrický model





# SQL geometrický model

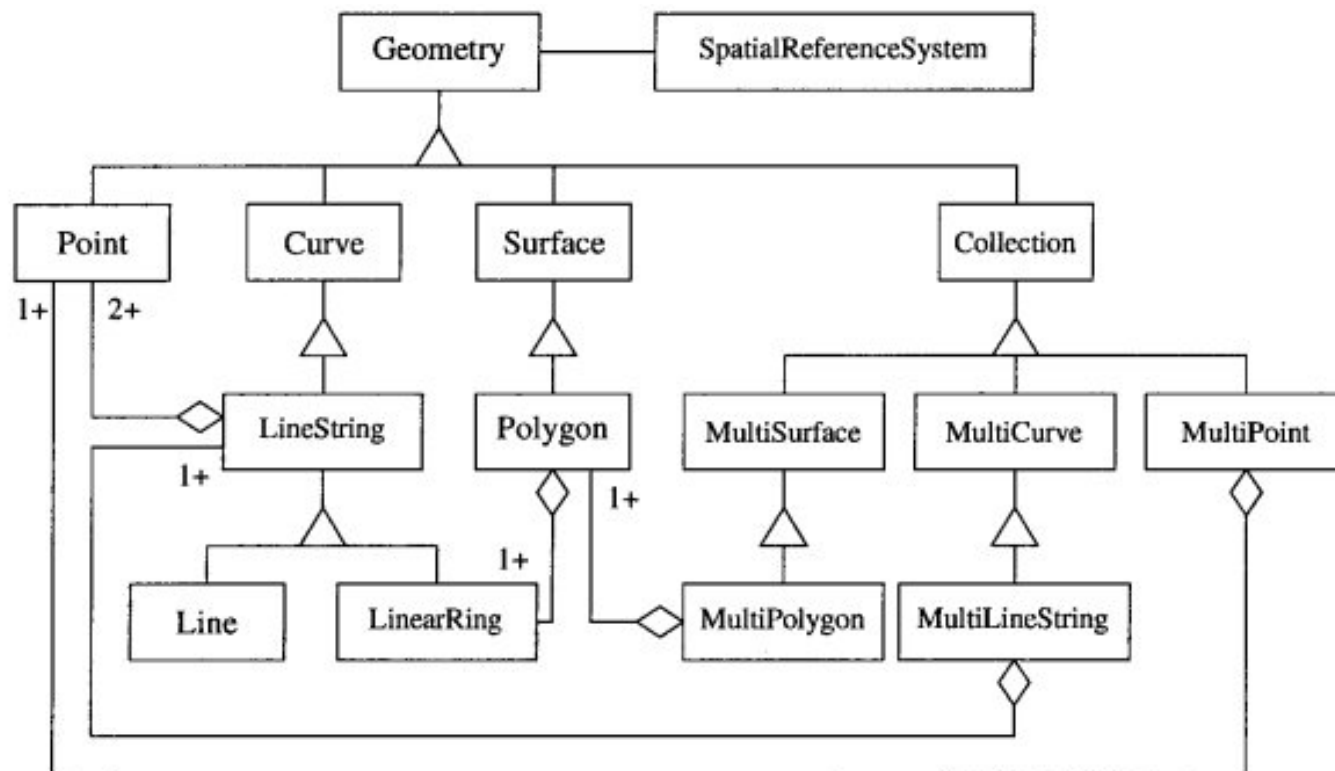
---

- Hierarchický model, zachycuje vztahy mezi prostorovými datovými typy (dědičnost)
- Prostorové datové typy reprezentují 0, 1 a 2 dimenzionální geometrické útvary
  - Prostorové datové typy jsou založeny na 2D geometrii s lineární interpolací mezi vrcholy
- Obsahuje abstraktní typy, od kterých nelze vytvářet instance (ST\_Surface...)



# SQL geometrický model

- Vychází z geometrického modelu OGC





# 0-dimenzionální objekty

---

- Bod:
  - ST\_Point
    - X a Y souřadnice vzhledem k nějakému systému souřadnic
  - ST\_MultiPoint
    - Kolekce bodů

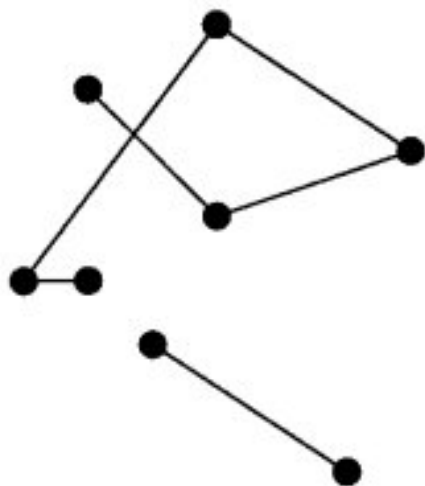


# 1-dimenzionální objekty

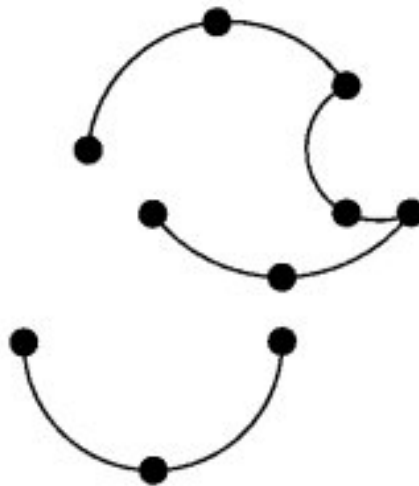
---

- Křivky:
  - ST\_LinearString
  - ST\_CircularString
  - ST\_CompoundCurve
    - Kombinace ST\_LinearString a ST\_CircularString
  - ST\_MultiLineString
    - Neexistují ST\_MultiCircularString a ST\_MultiCompoundString

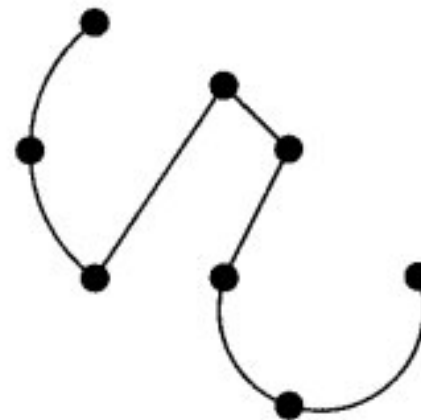
# 1-dimenzionální objekty



(a) Linear Strings



(b) Circular Strings



(c) Compounds

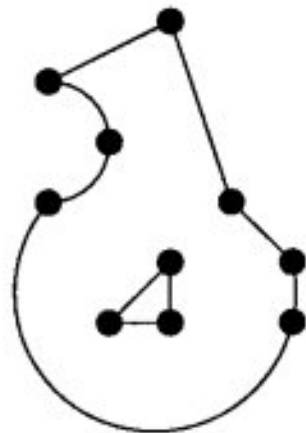


# 2-dimenzionální objekty

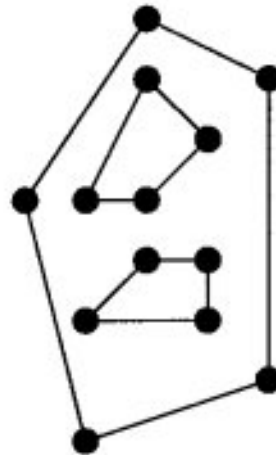
---

- Povrchy
  - ST\_CurvePolygon
    - ST\_Polygon (potomek ST\_CurvePolygon)
  - ST\_MultiPolygon
- Hranice (boundary) je reprezentována uzavřenou křivkou (resp. křivkami, obsahuje-li povrch díry)

# 2-dimenzionální objekty



(a) Curve Polygon



(b) Polygon

# Rozdíly SQL oproti OGC modelu



---

- SQL model vypouští typy Line a LinearRing, místo nich zaveden nový typ ST\_LineString
- Zavedeny nové typy pro reprezentaci křivek a povrchů tvořených kruhovými oblouky (ST\_CircularString...)

# Rozdíly SQL oproti OGC modelu



---

- U agregovaných typů (kolekcí) není z SQL modelu zřejmé, ze kterých „jednoduchých“ typů jsou složeny
  - Není například zřejmé, zda se typ ST\_MultiPoint skládá z objektů typu ST\_Point





# Prostorové systémy souřadnic

---

- Každý prostorový (geometrický) objekt je asociován s nějakým prostorovým systémem souřadnic (Spatial Reference System)
- Mnoho druhů souřadnicových systémů
  - U některých je třeba brát v úvahu zakřivení zemského povrchu



# Prostorové systémy souřadnic

---

- Reprezentují se typem ST\_SpatialRefSys
- Všechny hodnoty sloupců reprezentujících prostorová data (prostorové atributy) v SQL dotazu musí být definovány ve stejném souřadnicovém systému



# Prázdné prostorové objekty

---

- SQL geometrický model nezahrnuje samostatný typ pro reprezentaci prázdného prostorového objektu
- Instance každého typu může obsahovat prázdný prostorový objekt



# Prázdné prostorové objekty

---

- Pokud je návratovou hodnotou nějaké metody prázdný prostorový objekt a návratový typ není přesně definován, je vrácen jako prázdný prostorový objekt bod
- Mezi prázdnými prostorovými objekty lze provádět implicitní i explicitní přetypování

# Operace na prostorových objektech



---

- Průnik
- Sjednocení
- Relace „sousedí“
  - Objekt A sousedí s objektem B
- Relace „obsahovat“
  - Objekt A obsahuje objekt B
- Vzdálenost dvou objektů
- ...



# Typické dotazy

---

- Najdi všechna města vzdálená od Londýna nejvíce 50 km
- Najdi 5 nejbližších měst k Londýnu
- Najdi všechny dvojice měst, která jsou do sebe vzdálena nejvýše 200 km
- Najdi mi všechny budovy, které sousedí s vlakovým nádražím



# Metody definované na prostorových datových typech

---

- Rozděleny do čtyř základních kategorií:
  - Konverze prostorových objektů do/z externích datových formátů
  - Práce s atributy prostorových objektů
  - Porovnávání prostorových objektů
  - Generování nových prostorových objektů



# Konverze do/z externích datových formátů

---

- SQL/MM standard definuje tři implementačně nezávislé externí datové formáty pro reprezentaci prostorových objektů:
  - Textová reprezentace
    - Well-known text representation (WKT)
  - Binární reprezentace
    - Well-known binary representation (WKB)
  - Geography markup language (GML)





# Externí datové formáty

---

- Textová reprezentace
  - point (10 10)
  - multipolygon (((1 1, 2 2, 1 2, 1 1)),((10 10, 10 20, 20 20, 20 10, 10 10)))
- Binární reprezentace
  - 01010000000000001010



# Externí datové formáty

---

- Geography markup language

```
<gml:Polygon srsName="EPSG:4326">  
  <gml:outerBoundaryIs>  
    <gml:LinearRing>  
      <gml:coordinates>  
        <gml:ctuple>0,0</gml:ctuple>  
        <gml:ctuple>50,0</gml:ctuple>  
        <gml:ctuple>50,40</gml:ctuple>  
        <gml:ctuple>0,0</gml:ctuple>  
      </gml:coordinates>  
    </gml:LinearRing>  
  </gml:outerBoundaryIs>
```

# Konverze z externích datových formátů



---

- Pomocí konstruktorů prostorových datových typů
  - Pouze z WKT a WKB reprezentace
- Z GML pomocí speciálních funkcí
  - ST\_LineFromGML, ST\_MPointFromGML...
  - Pro zpětnou kompatibilitu s OGC specifikací
    - ST\_PointFromText...



# Konverze do externích datových formátů

---

- Tři metody
  - ST\_AsText
  - ST\_AsBinary
  - ST\_AsGML

# Práce s atributy prostorových objektů



---

- Všechny prostorové datové typy mají nějaké společné atributy (př. dimenze)
- Každý podtyp v SQL geometrickém modelu si přidává další, jemu specifické, atributy
  - Př. Polygon – plocha, kterou zabírá



# Příklady metod

---

- ST\_Boundary
  - Vrací hranici prostorového objektu
- ST\_IsEmpty
- ST\_X
  - Vrací X-ovou souřadnici bodu
- ST\_Length
  - Vrací délku křivky

# Porovnávání prostorových objektů



---

- ST\_Equals
  - Test na prostorovou rovnost dvou objektů
- ST\_Intersect, ST\_Crosses, ST\_Overlaps
  - Test na prostorový průnik dvou objektů
- ST\_Touches
  - Test, jestli se objekty dotýkají hranicemi, ale neprotínají se
- ST\_Within, ST\_Contains
  - Test, jestli jeden objekt obsahuje jiný

# Porovnávání prostorových objektů

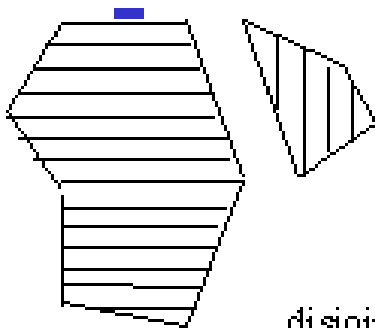


---

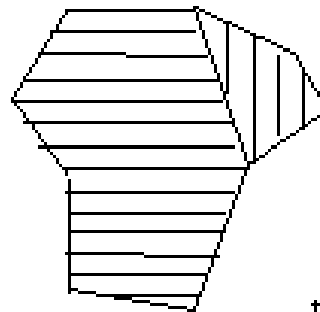
- Metody pro porovnávání objektů jsou založeny na binárních prostorových predikátech
- Všechny výše uvedené metody vracejí 1 pokud je příslušná relace splněna (TRUE), jinak vrací 0
- Další metody
  - ST\_Distance



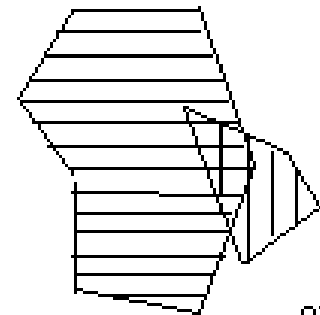
# Binární prostorové predikáty



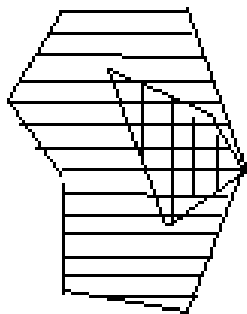
disjoint



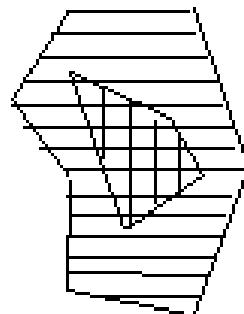
meet



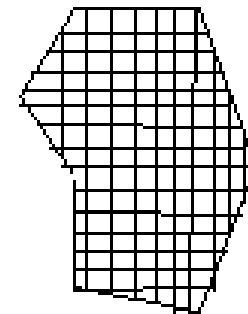
overlap



covers/  
covered\_by



inside/  
contains



equal

# Rozdíl mezi predikáty contains a covers



---

- A covers B nabývá TRUE, když  $\text{interior}(B) \subset \text{interior}(A)$  a současně  $\text{boundary}(A) \cap \text{boundary}(B) \neq \emptyset$
- A contains B nabývá TRUE, když  $\text{interior}(B) \cup \text{boundary}(B) \subset A$



# Generování nových prostorových objektů

---

- Nové objekty mohou být výsledkem provedení nějaké operace na množině stávajících objektů
  - ST\_Intersection
  - ST\_Union
  - ST\_Difference



# Generování nových prostorových objektů

---

- Mohou být získány pomocí nějakého algoritmu provedeného na jeden objekt
  - ST\_Buffer („nárazníková“ zóna)
  - ST\_ConvexHull (konvexní obal)
  - ST\_Envelope
    - Vrací minimální ohraničující obdélník (MOO) k danému objektu



# Přehled metod

---

ST_Geometry	ST_Dimension	ST_Length	ST_Distance
ST_Point	ST_CoordDim	ST_StartPoint	ST_Equals
ST_LineString	ST_GeometryType	ST_EndPoint	ST_Relate
ST_CircularString	ST_SRID	ST_IsClosed	ST_Disjoint
ST_CompoundCurve	ST_IsEmpty	ST_IsRing	ST_Intersects
ST_CurvePolygon	ST_IsSimple	ST_NumPoints	ST_Touches
ST_Polygon	ST_IsValid	ST_PointN	ST_Crosses
ST_GeomCollection	ST_Boundary	ST_NumCurves	ST_Within
ST_MultiPoint	ST_Envelope	ST_CurveN	ST_Contains
ST_MultiLineString	ST_ConvexHull	ST_Area	ST_Overlaps
ST_MultiPolygon	ST_Buffer	ST_Perimeter	
ST_AsText	ST_Intersection	ST_Centroid	
ST_AsBinary	ST_Union	ST_PointOnSurface	
ST_AsGML	ST_Difference	ST_ExteriorRing	
	ST_SymDifference	ST_InteriorRings	
ST_Transform		ST_NumInteriorRing	
	ST_X	ST_InteriorRingN	
	ST_Y		



# Rozšiřování SQL geometrického modelu

---

- Při rozšiřování geometrického modelu o nový datový typ je třeba vytvořit ještě další typ reprezentující množinu objektů takového typu
  - Viz např. `ST_Polygon` a `ST_MultiPolygon`



# Nedostatky SQL geometrického modelu

---

- Mějme následující dotaz:

```
SELECT ST_Intersection(ST_Polygon(  
  'polygon((10 10, 10 20, 20 20, 20  
  10, 10 10))', 1), mnozina_bodu)  
FROM tabulka  
WHERE ...
```

- Sloupeček mnozina\_bodu obsahuje hodnoty typu ST\_MultiPoint



# Nedostatky SQL geometrického modelu

---

- Výsledek dotazu může být
  - Prázdný objekt
  - Jeden bod (ST\_Point)
  - Množina bodů (ST\_MultiPoint)
- Jeho získání ale není triviální, neboť ST\_Point a ST\_MultiPoint se nachází v různých větvích geometrického modelu a tudíž můžeme použít pouze obecných metod definovaných v ST\_Geometry



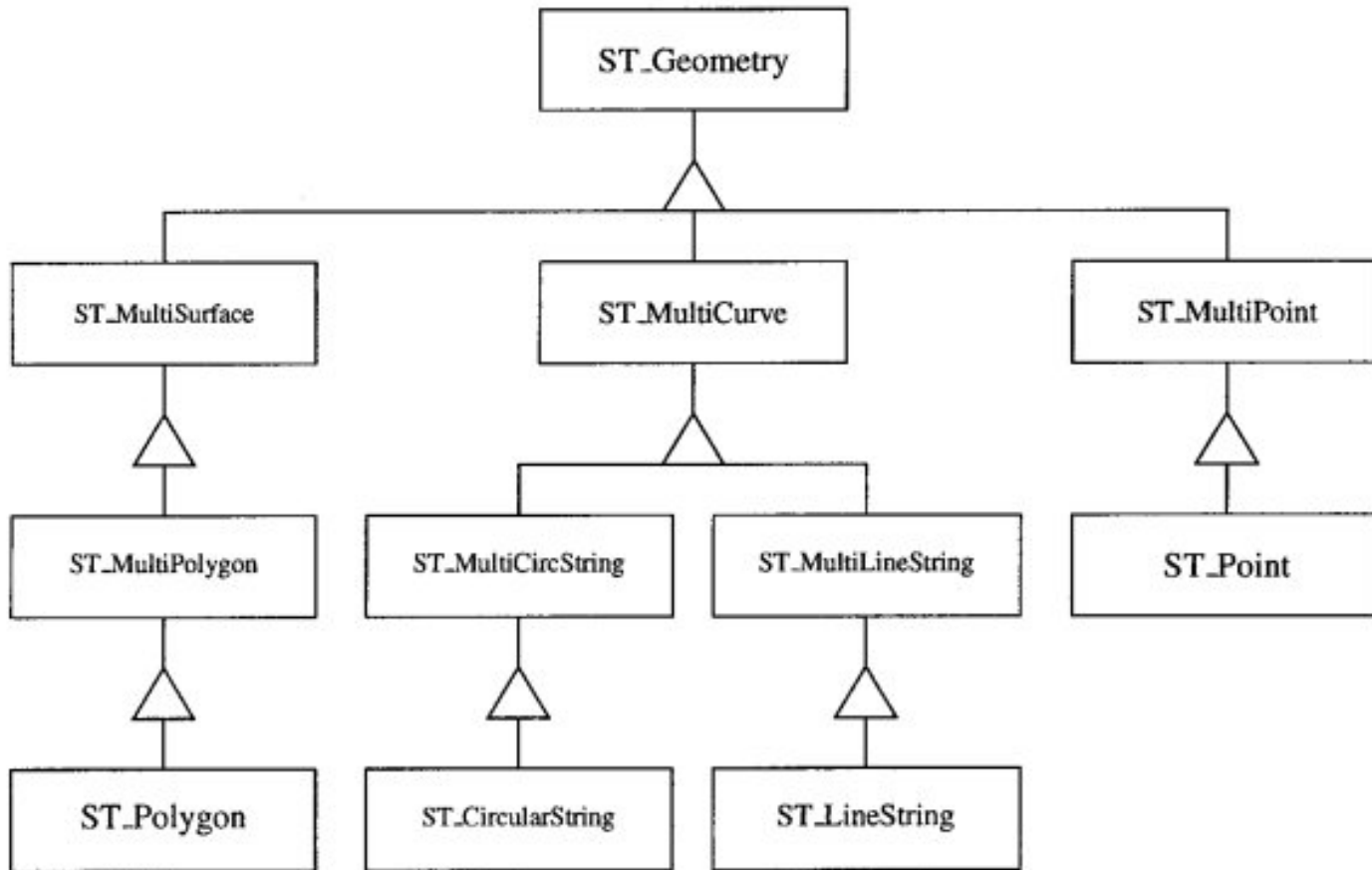


# Nedostatky SQL geometrického modelu

---

- Řešení problému
  - Modifikace geometrického modelu

# Modifikovaný geometrický model





# Diskuse k metodám

---

- Chybí konstruktory využívající GML
  - Řešení: Využití WKT konstruktorů
- Chybí podpora dalších důležitých externích formátů
  - Př. „shape format“



# Diskuse k metodám

---

- Některé metody poskytují téměř stejnou funkcionalitu
  - ST\_Intersects
  - ST\_Crosses
  - ST\_Overlaps



# Příklady

---



# Příklad 1: Pojišťovna

---



# Pojišťovna

---

- Pojišťování budov v záplavových oblastech



# Pojišťovna

---

```
CREATE TABLE reky {  
    nazev      VARCHAR(30) PRIMARY KEY,  
    mnozstvi_vody  DOUBLE PRECISION,  
    koryto     ST_LineString,  
    zaplavova_oblast  ST_MultiPolygon  
}
```

```
CREATE TABLE budovy {  
    zakaznik  VARCHAR(50) PRIMARY KEY,  
    ulice     VARCHAR(50),  
    mesto     VARCHAR(20),  
    zip       VARCHAR(10),  
    pozemek   ST_Polygon  
}
```





# Pojišťovna

---

- Rozšíření záplavové oblasti řeky FLOOD o 2 km ve všech směrech

```
UPDATE reky
SET zaplavova_oblast =
    zaplavova_oblast.ST_Buffer(2,
    'KILOMETER')
WHERE nazev='FLOOD'
```



# Pojišťovna

---

- Najdi všechny zákazníky, jejichž budovy se nachází v záplavové oblasti řeky FLOOD

```
SELECT zakaznik, ulice, mesto,  
zip
```

```
FROM budovy AS b, reky AS r
```

```
WHERE b.pozemek.ST_Within(  
    r.zaplavova_oblast) = 1
```



# Příklad 2: Banka

---



# Banka

---

- Síť poboček
- Každý zákazník může mít založen jeden, nebo více účtů
- Všechny účty jednoho zákazníka jsou spravovány právě jednou pobočkou



# Banka

---

```
CREATE TABLE zakaznici {
    id            INTEGER PRIMARY KEY,
    jmeno        VARCHAR(20),
    ulice        VARCHAR(25),
    město        VARCHAR(10),
    stat         VARCHAR(2),
    zip          VARCHAR(5),
    typ          VARCHAR(10),
    lokace       ST_Point
)
```



# Banka

---

```
CREATE TABLE pobočky {
    id INTEGER PRIMARY KEY,
    nazev    VARCHAR(12),
    manager  VARCHAR(20),
    ulice    VARCHAR(20),
    mesto    VARCHAR(10),
    stat     VARCHAR(2)
    zip VARCHAR(5),
    lokace   ST_Point,
    oblast   ST_Polygon
}
```



# Banka

---

```
CREATE TABLE ucty (  
    id            INTEGER PRIMARY KEY,  
    id_vlastnika  INTEGER NOT NULL,  
    id_pobocky    INTEGER NOT NULL,  
    typ           VARCHAR(10) NOT NULL,  
    zustatek      DECIMAL(14, 2) NOT NULL,  
    CONSTRAINT fk_zakaznici FOREIGN  
    KEY(id_vlastnika) REFERENCES zakaznici(id),  
    CONSTRAINT fk_pobocky FOREIGN KEY(id_pobocky)  
        REFERENCES pobocky(id)  
)
```



# Banka

---

- Najdi všechny zákazníky se zůstatkem na účtu větším než \$10 000, a kteří bydlí dále jak 20 mil od příslušné pobočky, která jim účet spravuje

```
SELECT DISTINCT z.id, z.jmeno
FROM zakaznici AS z JOIN ucty AS u ON
(z.id=u.id_vlastnika)
WHERE u.zustatek > 10000 AND
z.lokace.ST_Distance((SELECT p.lokace
FROM pobocky AS p WHERE p.id=u.id_pobocky),
'MILES') > 20
```





# Banka

---

- Najdi všechny dvojice poboček, jejichž oblasti působnosti se překrývají

```
SELECT p1.id, p2.id,  
p1.oblast.ST_Overlaps(p2.oblast) .  
    ST_AsText()  
FROM pobocky AS p1 JOIN pobocky AS p2 ON  
    (p1.id < p2.id)  
WHERE p1.oblast.ST_Overlaps(p2.oblast) .  
    ST_IsEmpty() = 0
```



# Banka

---

- Najdi všechny zákazníky, kteří bydlí v okruhu 10 mil od nějaké pobočky, která nespravuje jejich účty

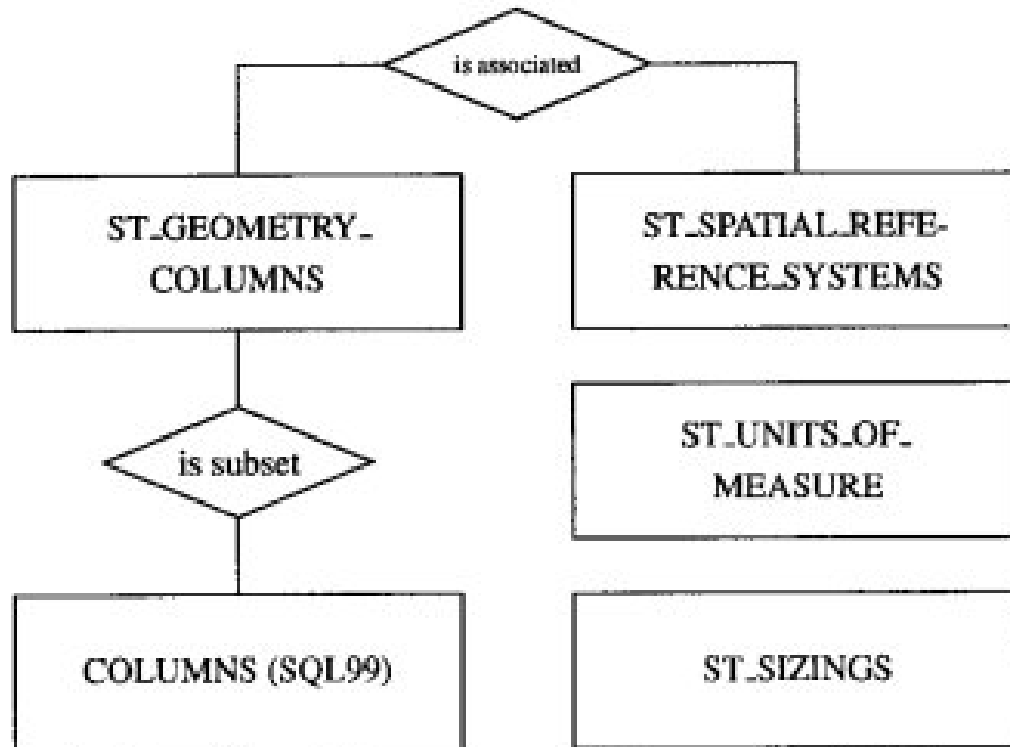
```
SELECT z.jmeno, p.id
FROM pobocky AS p, zakaznici AS z
WHERE p.lokace.ST_Buffer(10,
'MILES').ST_Contains(z.lokace)
= 1 AND NOT EXISTS (SELECT 1 FROM ucty AS u
WHERE u.id_vlastnika = z.id AND
p.id = u.id_pobocky)
```



# Datový katalog

---

# Pohledy (VIEWS)





# ST\_GEOMETRY\_COLUMNS

---

- Záznamy o všech sloupcích (atributech), které jsou deklarovány jako prostorový datový typ
- Ke každému sloupci může být přidružen patřičný souřadnicový systém



# ST\_GEOMETRY\_COLUMNS

---

- Jeden záznam pohledu se skládá z
  - Identifikátoru sloupce (atributu)
    - Katalog
    - Schéma
    - Tabulka
    - Název sloupce
  - Identifikátoru souřadnicového systému
    - Název souřadnicového systému
    - Numerický identifikátor

# ST\_SPATIAL\_REFERENCE\_SYS TEMS



---

- Informace o definovaných souřadnicových systémech



# ST\_UNITS\_OF\_MEASURE

---

- Informace o matematických jednotkách využívaných při výpočtech vzdálenosti dvou objektů, délky křivky...
- Každá jednotka má
  - Název (KILOMETER, RADIAN..)
  - Typ (úhlová, délková)
  - Konverzní faktor vůči základní jednotce daného typu





# ST\_SIZINGS

---

- Různé meta-proměnné a jejich hodnoty
  - Příklad: ST\_MaxGeometryAsText
    - Maximální délka textové reprezentace (WKT) prostorového objektu



# Diskuse k datovému katalogu

---

- **ST\_SPATIAL\_REFERENCE\_SYSTEMS**
  - Celkem primitivní
  - Práce na propracovanější organizaci souřadnicových systémů (EPSG - The European Petrol Survey Group)
- **ST\_SIZINGS**
  - Podobný pohledu SIZING definovanému v SQL99



# Závěr

---



# Produkty

---

- IBM DB2 Spatial Extender
- IDS Spatial DataBlade
- Oracle 9i Spatial



# Shrnutí

---

- SQL-MM Spatial
  - Standardizuje ukládání, výběr, dotazování a aktualizaci prostorových objektů
  - Definuje množinu typů a metod pro reprezentaci 0, 1 a 2 dimenzionálních prostorových objektů
- Připravuje se druhá verze standardu



# Literatura

---

- K. Stolze: SQL/MM Spatial: The Standard to Manage Spatial Data in Relational Database Systems  
<http://www.btw2003.de/proceedings/paper/68.pdf>
- J. Pokorný: Prostorové objekty a SQL  
[http://gis.vsb.cz/Publikace/Sborniky/GIS\\_Ova/gis\\_ova\\_2001/sbornik/Referaty/pokorny.htm](http://gis.vsb.cz/Publikace/Sborniky/GIS_Ova/gis_ova_2001/sbornik/Referaty/pokorny.htm)



# Obsah

---

- Úvod
- Část 1: Framework
- Část 2: Full-Text
- Část 3: Spatial
- **Část 4: General Purpose Facilities**
- Část 5: Still Image
- Část 6: Data Mining



# General Purpose Facilities

---

- Pokus o množinu tříd pro hlavní matematické operace
- Tato část ze standardu prozatím vypadla
  - Vysoké náklady
  - Málo uživatelů





# Obsah

---

- Úvod
- Část 1: Framework
- Část 2: Full-Text
- Část 3: Spatial
- Část 4: General Purpose Facilities
- Část 5: Still Image
- Část 6: Data Mining



# Still Image > Úvod

---

- Obrázky = hodnotná data
- Ukládání obrázků
- Úprava obrázků
- Vyhledávání obrázků (dle vizuálních vlastností)



# Still Image > Typy

---

- SI\_StillImage
  - Pro obrazová data
- SI\_Feature
  - Nadtyp pro různé vlastnosti obrázku (dále)
- SI\_FeatureList
  - Seznam pro (všechny) vlastnosti obrázku



# Still Image > SI\_StillImage (1)

---

- Konstruktory
  - BLOB
  - BLOB + Formát (JPEG, TIFF, GIF, ...)
- Metody pro přeformátování (SI\_changeFormat)
- Vytvoření miniatury („Thumbnails“)
- Změna velikosti
- Ořezávání
- Rotace
- ...



# Still Image > SI\_StillImage (2)

---

```
create type SI_StillImage as (  
  SI_content binary large  
    object(SI_MaxContLength),  
  SI_contentLength integer,  
  SI_format character varying(8),  
  SI_height integer,  
  SI_width integer,  
  ...  
)
```



# Still Image > SI\_Feature (1)

---

- Užitečné pro vyhledávání
- Typ SI\_Feature má tyto podtypy
  - SI\_AvarageColor
  - SI\_ColorHistogram
  - SI\_PositionalColor
  - SI\_Texture
- Všechny vlastnosti mají metodu SI\_Score, která spočítá podobnost vlastností obrázků a vrátí reálnou hodnotu mezi 0 a 1



# Still Image > SI\_Feature (2)

---

```
create type SI_AverageColor under SI_Feature  
  (SI_AverageColorSpec SI_Color)
```

```
method SI_AverageColor (  
  RedValue integer,  
  GreenValue integer,  
  BlueValue integer)  
  returns SI_AverageColor
```

```
create function SI_AverageColor (image  
  SI_StillImage)  
  returns SI_AverageColor
```



# Still Image > Příklad > dotaz

---

```
SELECT *
  FROM Registrovaná_logo
 WHERE
      SI_findTexture (Naše_logo) .
      SI_Score (Logo) > 0.9;
```





# Obsah

---

- Úvod
- Část 1: Framework
- Část 2: Full-Text
- Část 3: Spatial
- Část 4: General Purpose Facilities
- Část 5: Still Image
- Část 6: Data Mining



# Data Mining > Úvod

---

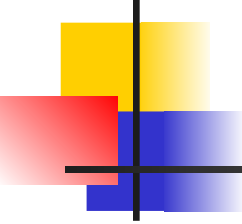
- Aplikací balíček
- Data Mining
  - Analýza (často rozsáhlých) observačních dat s cílem nalézt netušené vztahy a sumarizovat data novými způsoby tak, že jsou srozumitelná a užitečná pro jejich majitele.



# Data Mining > Techniky

---

- **RULE MODEL**
  - Hledá pravidla a vztahy v datech.
- **CLUSTERING MODEL**
  - Slučuje data do skupin s podobnými charakteristikami.
- **REGRESSION MODEL**
  - Předvídá klasifikaci nově pořizovaných dat.
- **CLASSIFICATION MODEL**
  - Předvídá klasifikaci (způsob shlukování), která bude nejlépe odpovídat novým datům.



# Data Mining > Stupně DM

---

- Vytvoření (train) DM modelu
- Testování modelu (u regression a classification technik)
- Aplikace na primární data



# Data Mining > Typy

---

- DM\_\*Model (Definice modelu)
  - \*  $\approx$  'Rule', 'Clus', 'Clas' nebo 'Reg'
- DM\_\*Setting (Nastavení modelu)
- DM\_MiningData (Testování modelu)
- DM\_\*TestResult (Výsledek testování modelu)
- DM\_\*Result (Výsledek aplikace modelu)
- DM\_\*Task (Řízení běžícího modelu)



# Závěr

---

- Budoucnost SQL/MM
  - SQL/MM General Purpose Facilities
  - SQL/MM Moving Images



# Zdroje

---

- SQL/MM Spatial: The Standard to Manage Spatial Data in Relational Database Systems
  - <http://www.btw2003.de/proceedings/paper/68.pdf>
- SQL Multimedia and Application Packages (SQL/MM)
  - <http://dbs.uni-leipzig.de/en/lehre/sql-multimedia-2001-records.pdf>
- Oracle® Spatial User's Guide and Reference: SQL Statements for Indexing Spatial Data
  - [https://cwisdb.cc.kuleuven.ac.be/ora10doc/appdev.101/b10826/sdo\\_objindex.htm](https://cwisdb.cc.kuleuven.ac.be/ora10doc/appdev.101/b10826/sdo_objindex.htm)
- <http://www.dvs.informatik.uni-kl.de/courses/MMDB/WS2003/Vorlesungsunterlagen/SQL-MM.half.pdf> (nemecky)



Otázky?

---