SQL-MM-Spatial
slajdy k přednášce NDBI001

Jaroslav Pokorný
MFF UK, Praha
pokorny@ksi.mff.cuni.cz

ISO/IEC 13249

- SQL/MM = Multi Media and Application Packages
- Snaha o standardizaci rozšíření pro multimédia v SQL
- Standard byl dokončen v r. 2003 a je rozdělen do šesti částí

ISO/IEC 13249

1. Framework (základy)
2. Fulltext (úplné texty)
3. Spatial (prostorové objekty)
5. Still images (statické obrázky)
6. Data mining (dolování dat)

Drafty:

7. Historie (2011)
8. Registr metadat (2011)

Framework

- Další časti jsou na sobě poměrné nezávislé – kromě této
- společný pro všecky další časti
- popisuje mechanismus definic a konvencí použitých v dalších částech

Fulltext

- úplné texty se od klasických znakových řetězců odlišují např. délkou, způsobem indexace aj.
- funkce a predikáty jako: SOUNDS LIKE, STEMMED, SYNONYM,...
- Počítá se hlavně se západními jazyky, ve kterých se jasně oddělují slova, věty

Still Image

- Obrázková data pomocí strukturovaného typu
  - SI_StillImage uchovává kromě obrázku také informace o jeho rozměrech, typu, formátu,...
- Metody: ořezaní, rotace, vytváření náhledů
  - SI_AverageColor – průměrná barva daného obrázku
  - SI_ColorHistogram – je nalezen výskyt každé barvy
  - SI_PositionalColor - lokace specifické bary

Still Image

- Charakteristiky obrázku mohou být kombinovány do SI_FeatureList

Příklad:

```
CREATE TABLE výrobky(
   Id          INTEGER PRIMARY KEY,
   photo_si    SI_StillImage,
   photo_ac    SI_AverageColor,
   photo_ch    SI_ColorHistogram,
   photo_pc    SI_PositionalColor,
   photo_tx    SI_Texture);
```

Data Mining

- Dolování dat – snaha objevit před tím neznámou, ale důležitou informaci ukrytou v množství dat
- definuje typy a metody pro dolování dat
- 4 techniky – pravidla, shlukování regresní model, klasifikace
- 3 stádia – trénování, testování, aplikace

Spatial

- Definuje jak ukládat a zpracovávat prostorová data pomocí SQL
- Definuje reprezentaci prostorových dat
- Definuje funkce na konverzi, porovnávaní a zpracovávání těchto dát

Obsah

1. Motivace
2. Vývoj prostorových dat a relačních databází
3. Geometrický model
4. Implementace geometrického modelu v prostředí SQL
5. Použití ADT pro prostorové objekty
6. Spatial SQL
7. Další vývoj
8. Literatura

Motivace

- Geografické informační systémy (GIS)
  - Plánování výstavby měst
  - Řízení dopravy
  - Mapovaní nalezišť nerostných surovin
  - Distribuční sítě
  - Bankovnictví
  - Pojišťovnictví
- CAD/CAM (Computer-aided design and manufacturing)
  - Návrh integrovaných obvodů
  - Návrh mostu
  - Návrh motoru

Motivace

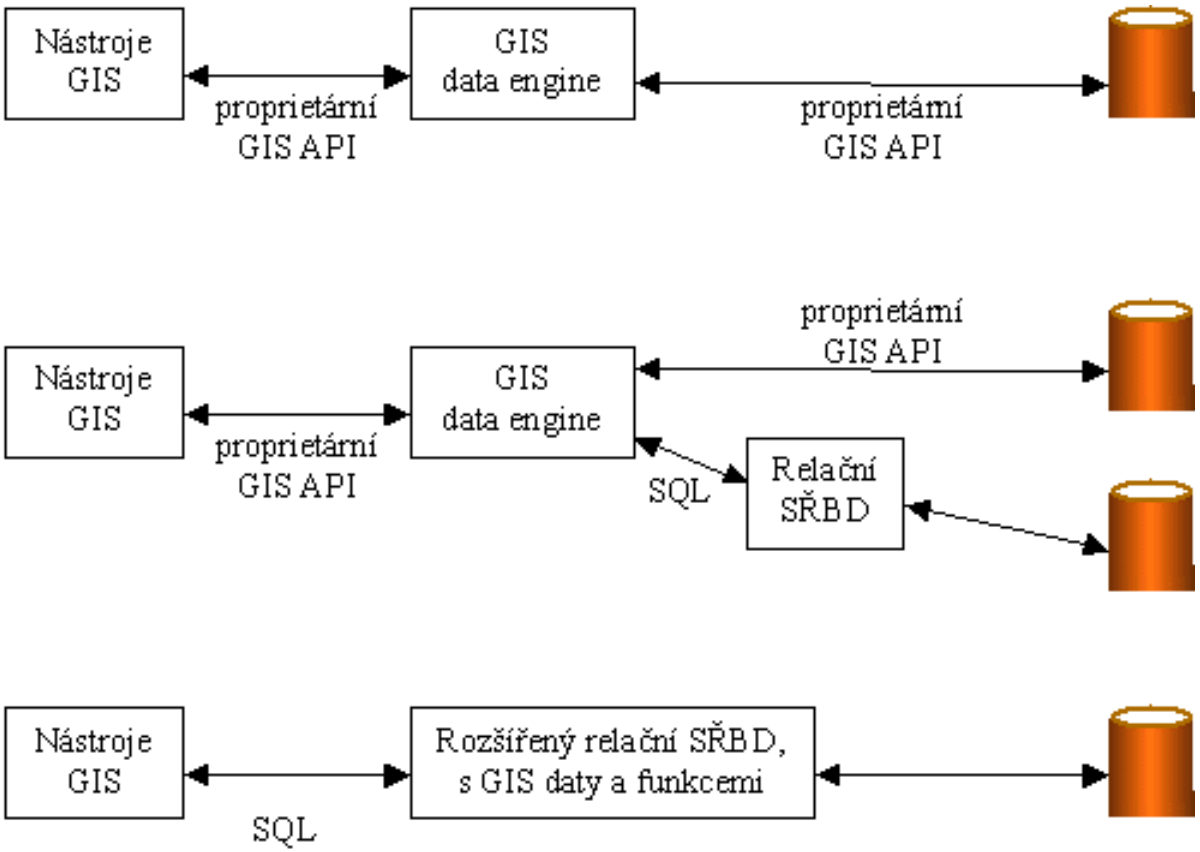
- Požadavky
  - dotazy výhradně na prostorové vlastnosti

    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 Úzké ul.

Historický vývoj GIS databází



Prostorová data v relačních DB

Df.: prostorový atribut, prostorová relace

- Možnosti SQL92:
  - prostorová data jsou uložena v tabulkách v 1NF,
  - prostorová data jsou uložena jako nestrukturované rozsáhlé binární objekty (BLOB) či jako “opaque types”
- Možnosti univerzálních serverů:
  - prostorová data jsou uložena v relační databázi rozšířené o typy prostorových dat
- Možnosti SQL99:
  - prostorová data jsou implementovaná pomocí uživatelsky definovaných typů

Prostorová data v relačních DB

- data jsou rozložena do tabulek

  ČTYŘSTĚNY(t,f), kde *t* je Id čtyřstěnu, *f* je Id jedné z jeho rovin.

  ROVINY(f,e), kde *f* je Id roviny a *e* je jedna z hran v té rovině.

  HRANY(e,p,q), kde *e* je Id hrany, *p* a *q* jsou koncové body hrany.

  BODY(p,x,y,z), kde *p* je Id bodu a *x*, *y*, *z* jeho souřadnice.

  D.: Protíná daná přímka daný čtyřstěn?

  Problémy: mnoho vstupů do mnoha tabulek
- data jako BLOBy nebo „opaque types“
  - Př.: Informix

Prostorová data v relačních DB

- v relační DB rozšířené o nové typy dat (+ speciální indexové struktury)
  - Př.: ORACLE 8
- v objektově-relační DB pomocí uživatelsky definovaných typů a funkcí
  - Př.: cartridges v ORACLE 8i, DataBlades v Informix, extendery v DB2
- v SQL99 a SQL:2003
  - pomocí objektových typů

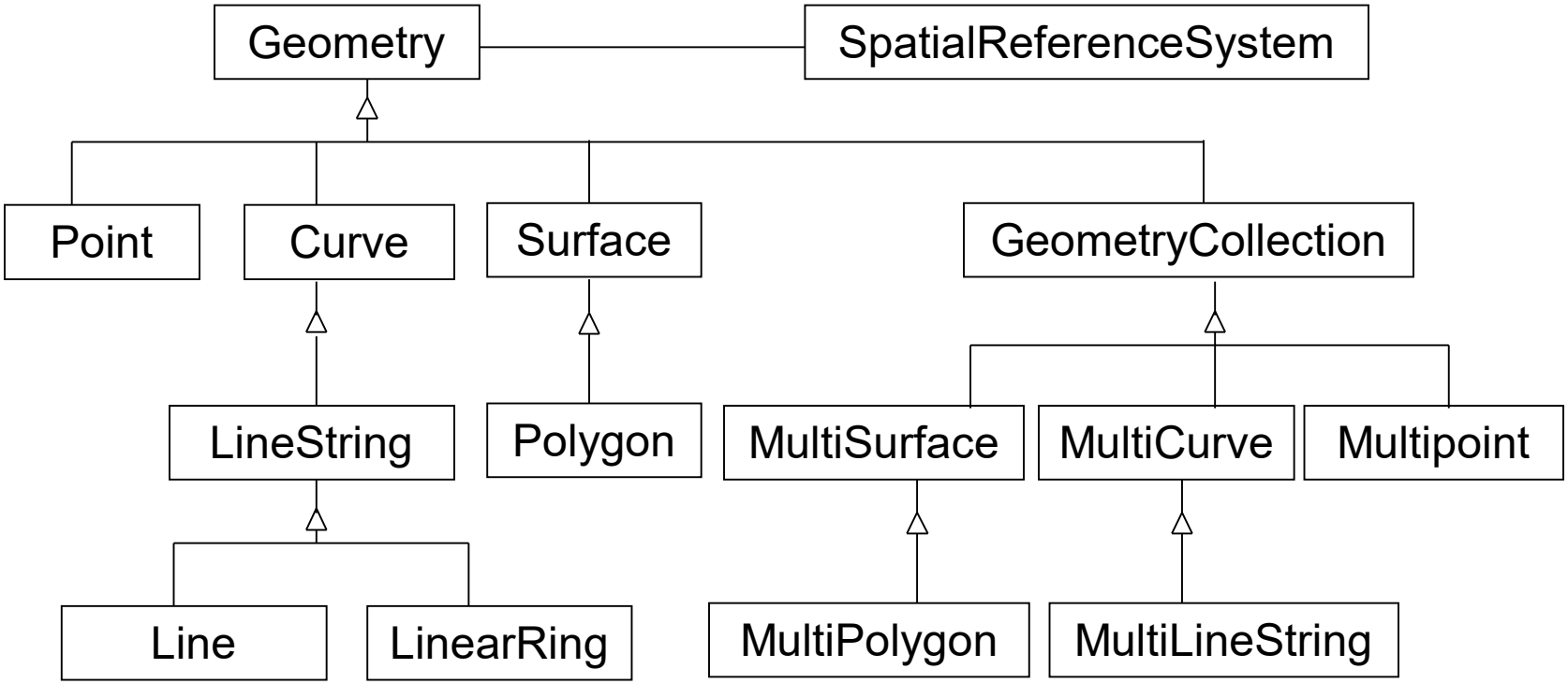
```
CREATE TYPE
lomená_čára AS
(uzly point ARRAY [10])
METHOD délka( );
```

Iniciativy

- Pracovní skupina ISO/IEC/JTC1/SC32WG4
  - SQL99 + další vývoj

    podporuje složitá a netabulková data
  - vznik SQL/MM v r. 2003
- Open Geospatial Consortium (dále OGC), dříve OpenGIS Consorcium, vyvinulo geometrický model pro SQL (Simple Features)
  - standardní SQL schéma pro uložení, výběr, dotazování a aktualizaci jednoduchých geoprostorových kolekcí objektů pomocí API ODBC (Open Database Connectivity)
  - v r. 1998 - kompatibilita s jádrem SQL/MM-Spatial

Geometrický model OGC



Pz.: pouze tzv. jednoduché objekty (2D, lineární interpolace, …)

Geometrický model OGC

- Je to abstraktní model.
- Využívá se pro definovaní vztahů mezi třídami a na zavedení pravidel dědičnosti pro metody pracující s instancemi těchto tříd.

Třídy modelu

- Geometry - abstraktní třída geometrických objektů.
- + systém prostorových referencí (Spatial Reference System).
- Podtřídy:
  - 0D geometrické objekty: body
  - 1D geometrické objekty: křivky.
  - 2D geometrické objekty: plochy
  - geometrická kolekce je multimnožinou,

  Př.: v kolekci bodů (Multipoint) mohou být dva body stejné.

Třídy modelu

- Křivka je dána posloupnost bodů, přičemž její podtřídy určují způsob interpolace.
  - LineString (lomená čára) - lineární interpolace
  - Line (úsečka) - je dána dvěma body.

  Df.: Křivka je jednoduchá, jestliže se sama se sebou nikde neprotíná. Uzavřené křivky mají společný počáteční a koncový bod.

  - LinearRing (kruh) obsahuje pouze jednoduché a uzavřené křivky.

Třídy modelu

- Plocha je rovněž posloupnost bodů. Má množinu hranic.
  - Polygon (mnohoúhelník) mají vnější hranici + několik vnitřních hranic (tvoří “díry”). Hranice (Boundary) jsou dány objekty ze třídy LinearRing.

Metody modelu

Základní operátory na úrovni GEOMETRY

- ENVELOPE – nejmenší opsaný obdélník
- SPATIALREFERENCE – vrací referenční systém geometrie
- EXPORT – změna geometrie na jinou reprezentaci
- BOUNDARY – vrací hranici geometrického objektu
- ISEMPTY – test, zdali je geometrický objekt prázdný, tj. obsahuje prázdnou množinu bodů

Metody modelu

- metody pro porovnání objektů – prostorové predikáty: Equal, Disjoint, Intersects, Cross, Overlaps, Within, Contains, Touch, Relate.
  - Výsledkem metody pro porovnání je 1 v případě, že výsledek porovnání je TRUE.
  - definice prostorových predikátů jsou založeny na modelu DE-9IM (Dimensionally Extended Nine-Intersection Model)
- pro prostorovou analýzu: Distance, Buffer, ConvexHull, Intersection, Union, Difference, SymDifference

Prostorové operátory

- Motivace: poskytnut uživatelům možnost získávat data nezávisle na fyzické organizaci, interoperabilita mezi systémy v distribuovaných sítích...
- Mají zachycovat všecky podstatné geometrické vlastnosti reálných objektů, jejich vztahy a vykonávat prostorovou analýzu

Rozdělení prostorových operátorů

- Topologické operátory
  - spojitost, počet komponent, průnik...
- Projekční operátory
  - konvexnost/konkávnost...
- Metrické operátory
  - kompaktnost, symetrie, vzdálenost, směr...

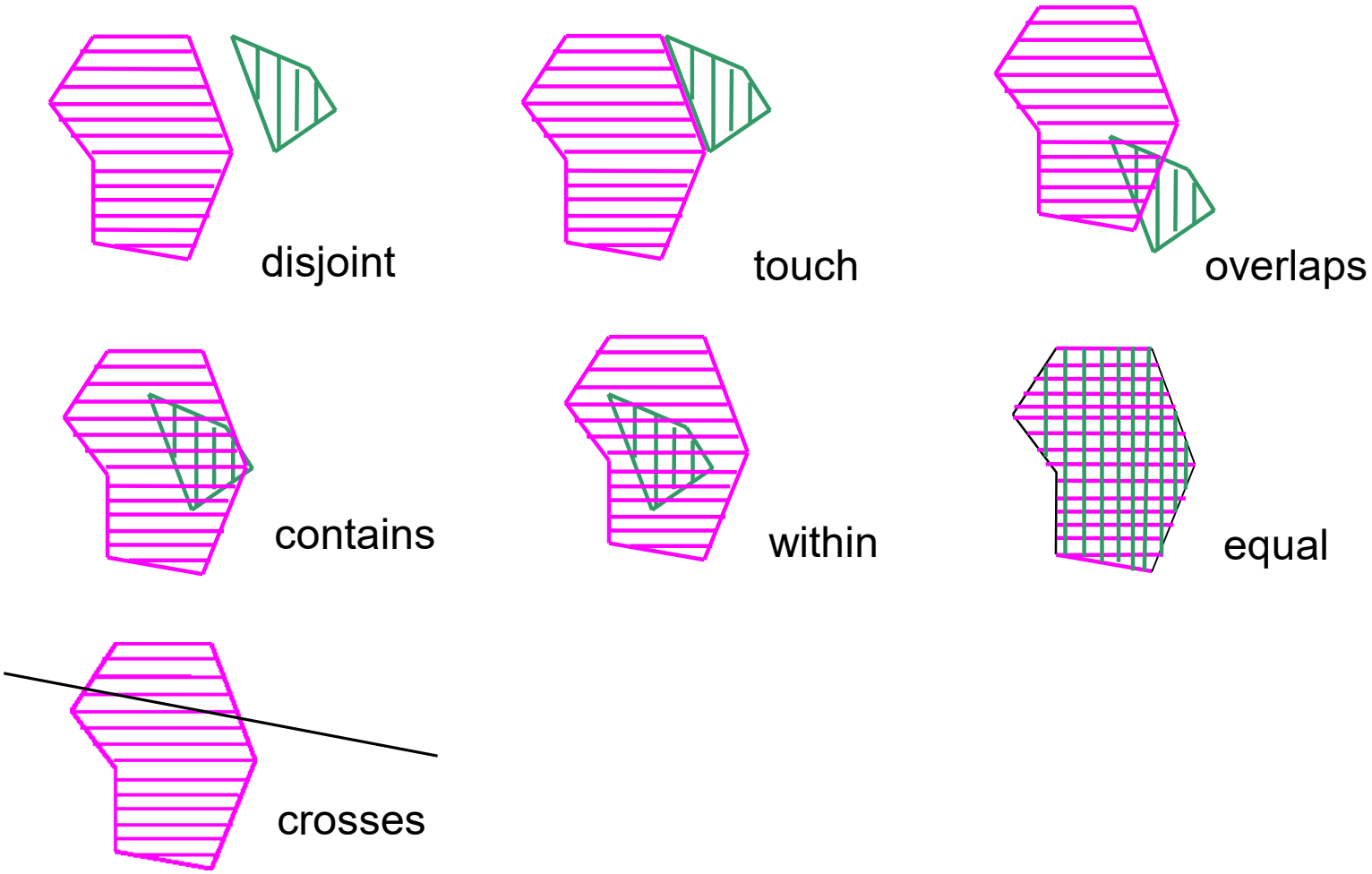
Topologické operátory

- EQUAL – testuje shodnost dvou geometrických objektů
- DISJOINT – testuje disjunktnost dvou geometrických objektů
- INTERSECTS – testuje, zdali se dva geometrické objekty protínají (bez omezení na operandy)
- OVERLAPS – testuje, zdali daný geometrické objekt překrývá jiný daný geometrický objekt; obě geometrie mají stejnou dimenzi
- CROSSES – testuje, zdali daný geometrické objekt protíná jiný daný geometrický objekt, přičemž dimenze průniku je o 1 menší než maximální dimenze operandů
- TOUCH – testuje, zdali se dva geometrické objekty dotýkají
- WITHIN – testuje, zdali je geometrický objekt obsažen v jiném daném geometrickém objektu
- CONTAINS – testuje, zdali daný geometrické objekt obsahuje jiný daný geometrický objekt

Topologické operátory

- EQUAL – testuje shodnost dvou geometrických objektů
- DISJOINT – testuje disjunktnost dvou geometrických objektů
- INTERSECTS – testuje, zdali se dva geometrické objekty protínají (bez omezení na operandy)
- OVERLAPS – testuje, zdali daný geometrické objekt překrývá jiný daný geometrický objekt; obě geometrie mají stejnou dimenzi
- CROSSES – testuje, zdali daný geometrické objekt protíná jiný daný geometrický objekt, přičemž dimenze průniku je o 1 menší než maximální dimenze operandů
- TOUCH – testuje, zdali se dva geometrické objekty dotýkají
- WITHIN – testuje, zdali je geometrický objekt obsažen v jiném daném geometrickém objektu
- CONTAINS – testuje, zdali daný geometrické objekt obsahuje jiný daný geometrický objekt

Topologické operátory



Operátory prostorové analýzy

- DISTANCE – vrací nejkratší vzdálenost mezi dvěma body dvou geometrických objektů
- BUFFER – vrací útvar, který obsahuje všecky body se vzdáleností od geometrického objektu menší nebo rovné zadané vzdálenosti
- INTERSECTION – vrací průnik geometrických objektů
- UNION – vrací sjednocení geometrických objektů
- DIFFERENCE – vrací rozdíl geometrických objektů

Převod z externích formátů

- pro reprezentaci jsou definovány tři externí datové formáty:
  - WKT (Well-known text representation),
  - WKB (Well-known binary representation),
  - GML(Geography Markup Language)
- každý typ implementuje konstruktor na vytvoření z WKT nebo WKB
- Převodní funkce: ST_LineFromGML, ST_AsText, ST_AsBinary, ST_AsGML

Převod z externích formátů

- WKT znamená, že je možné zacházet z prostorovými daty přirozene jako s literály. Např.

  ‘LINESTRING (10 10, 20 20, 30 40)‘

  označuje lomenou čáru danou třemi body.

Implementace geometrického modelu OGC v prostředí SQL

OGC navrhlo tři alternativy:

- pomocí tabulek v SQL92 s použitím numerických typů pro uložení geometrických objektů,
- pomocí tabulek v SQL92 s použitím binárních typů pro uložení geometrických objektů,
- pomocí tabulek v SQL92 s geometrickými typy tak, že ke geometrickým charakteristikám je přístup jak textově, tak binárně

Implementace geometrického modelu OGC v prostředí SQL

- co jsou binární typy pro geometrické objekty?
  - použita tzv. WKB for Geometry (WKBGeometry).
  - jsou kódovány do binárních objektů pevné nebo proměnné délky (podobně jako “opaque types”)
  - ukládají se: geometrické objekty + jejich minimální ohraničující obdélníky.

    $\Rightarrow$ tvorba indexů (např. pomocí R-stromů)
  - použití WKBGeometry je srozumitelné standardu ODBC.
- V SQL pouze 0, 1, 2-dimenzionální objekty

Přístup ORACLE

ORACLE 8 (pomocí SQL92), ORACLE od verze 8g, ORACLE 8i - pomocí OR

Př.: 1 varianta:

primitivní objekty: body, linie, polygony

geometrické objekty: z primitivních objektů + mají GID + event. neprostorové atributy

vrstvy: sdružují geometrické objekty se stejnými atributy

- *heterogenní*, např. mosty (body) a silnice (linie) nebo
- *homogenní*, mosty a silnice jsou ve dvou vrstvách).

Přístup ORACLE - 4 relace

<jm-v>_LAYER(ORDCNT, LEVEL)

<jm-v>_DIM(DIMNUM, LB, UB, TOLERANCE, DIMNAME)

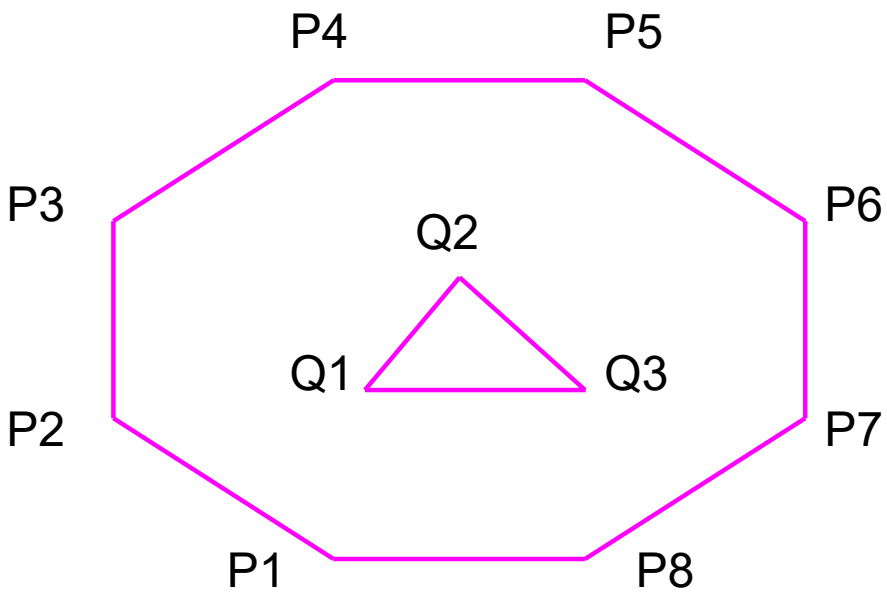
<jm-v>_GEOM(GID, ESEQ, TYPE, SEQ, X1,Y1,…, Xn,Yn)

<jm-v>_INDEX(GID, CODE, MAXCODE)

Indexace: pomocí 4-stromů

- ORDCNT - počet souřadnic v každém řádku tabulky GEOM, LEVEL - počet dělení kvadrantů indexačních čtyřstromů,
- DIMNUM - číslo dimenze, minimální a maximální souřadnice (LB, UB), tolerance, jméno dimenze
- ESEQ - číslo elementu, typ elementu (0, 1, 2), SEQ - pořadí řádku v rámci element, souřadnice

Př.: Polygon s dírou (objekt Region1)



Objekt Region1 (bez indexu)

REGION1_LAYER
ORDCNT
4

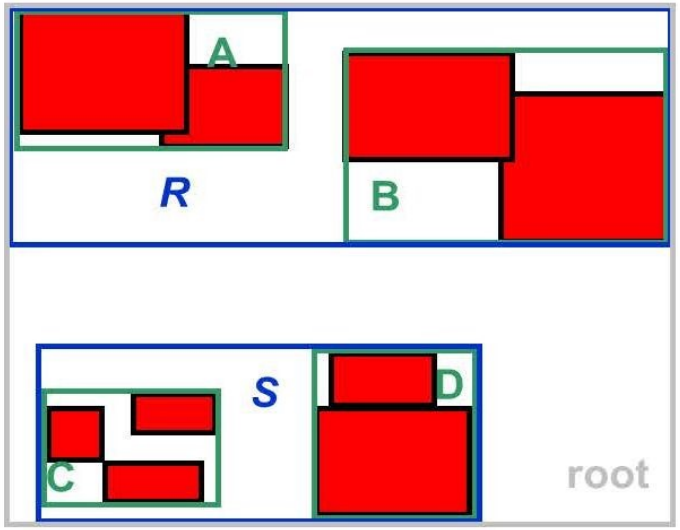
REGION1_DIM				
DIMNUM	LB	UB	TOLERANCE	DIMNAME
1	0	100	0,4	X osa
2	0	100	0,4	Y osa

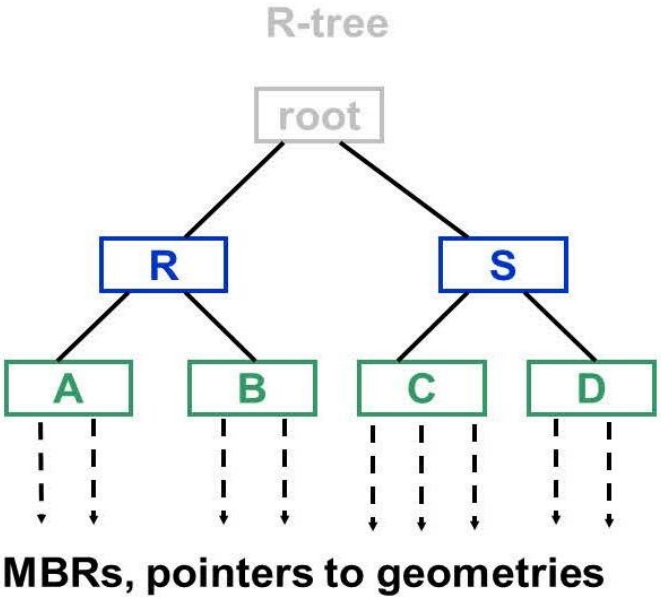
REGION1_GEOM							
GID	ESEQ	ETYPE	SEQ	X1	Y1	X2	Y2
1234	0	3	0	P1(X)	P1(Y)	P2(X)	P2(Y)
1234	0	3	1	P2(X)	P2(Y)	P3(X)	P3(Y)
1234	0	3	2	P3(X)	P3(Y)	P4(X)	P4(Y)
1234	0	3	3	P4(X)	P4(Y)	P5(X)	P5(Y)
1234	0	3	4	P5(X)	P5(Y)	P6(X)	P6(Y)
1234	0	3	5	P6(X)	P6(Y)	P7(X)	P7(Y)
1234	0	3	6	P7(X)	P7(Y)	P8(X)	P8(Y)
1234	0	3	7	P8(X)	P8(Y)	P1(X)	P1(Y)
1234	1	3	0	Q1(X)	Q1(Y)	Q2(X)	Q2(Y)
1234	1	3	1	Q2(X)	Q2(Y)	Q3(X)	Q3(Y)
1234	1	3	2	Q3(X)	Q3(Y)	Q1(X)	Q1(Y)

Další přístupy

- Informix
  - moduly pro 2D (+10 datových typů + 200 funkcí) a 3D (+18 datových typů + 1000 funkcí)
  - využití SQL:1999 + R-stromy
- IBM
  - modul ve spolupráci s ESRI (Environmental Systems Research Institute) – mezinárodním dodavatelem GIS
  - využití OR přístupu
- SQL Server: od r. 2008
- Sybase: od r. 2010

Indexace pomocí R-stromů





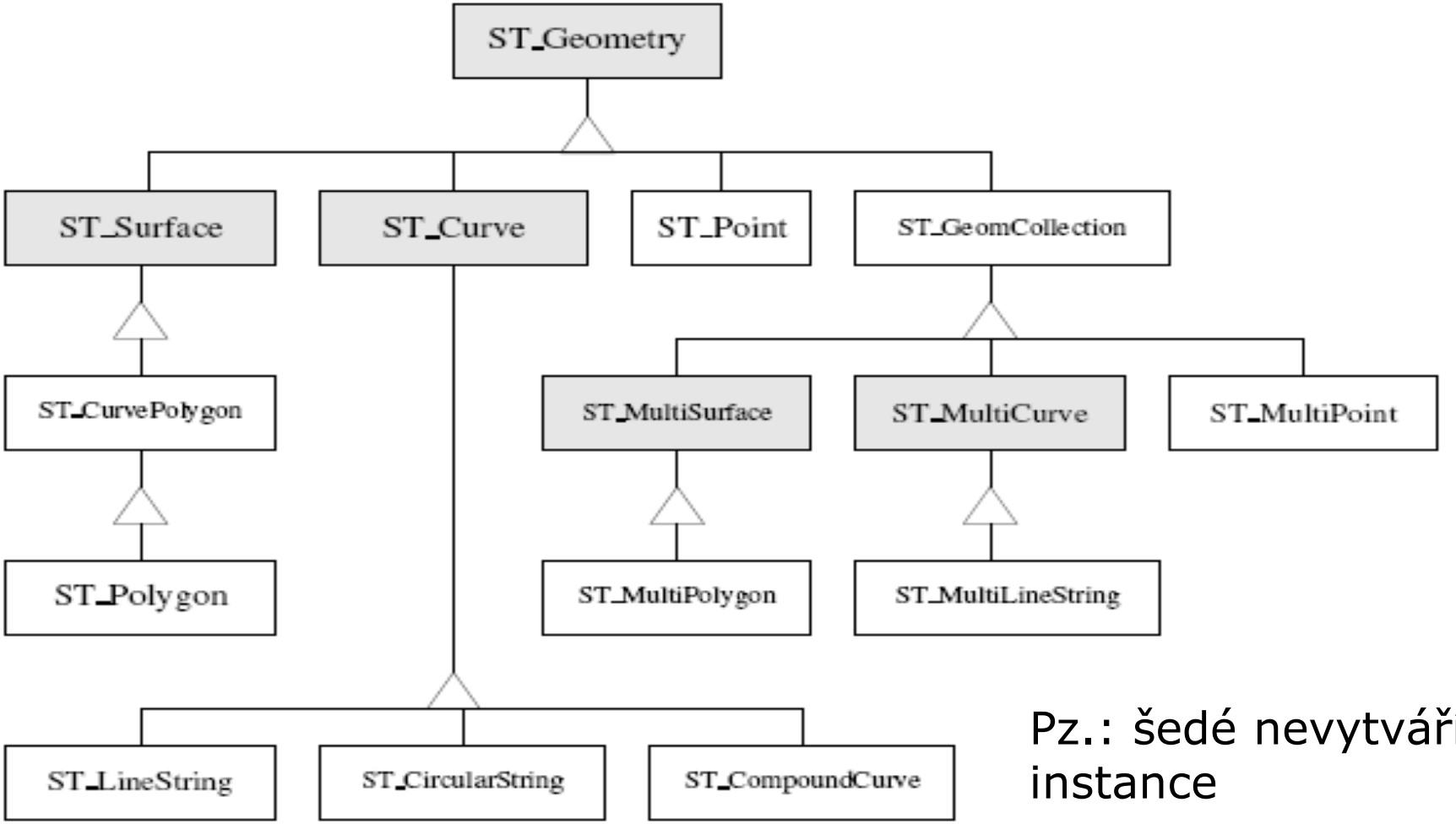
Použity jsou minimální ohraničující obdélníky

Spatial SQL vs. OGC

- Hlavním rozdílem oproti hierarchii geometrických tříd OGC je vynechání odvozených typů Line a LinearRing – SQL na to využívá ST_LineString
- Jsou zavedené nové typy obloukovité křivky a plochy
- Předpona ST_ (Spatial Type)

Původně Spatial and Temporal

Hierarchie typů v Spatial SQL



Pz.: šedé nevytváří instance

Metody na ST_Geometry

Můžeme rozdělit do čtyř kategorií

1. pro převod mezi geometrickými objekty a externími datovými formáty
2. pro testování prostorových vztahů mezi dvěma geometrickými objekty
3. pro generovaní nových objektů z existujících
4. pro práci s vlastnostmi objektů

ST_Point

- 0-dimenzionální geometrický objekt
- definovaný souřadnicemi X a Y
- hranicí je prázdna množina
- objekt tohoto typu (bod) je možné vytvořit

Metody ST_Point

- ST_Point vrací specifikovanou ST_Point hodnotu
- ST_X vrací/modifikuje souřadnici X
- ST_Y vrací/modifikuje souřadnici Y
- ST_ExplicitPoint vrací souřadnice bodů jako DOUBLE PRECISION ARRAY

Praktické využití ST_Point

- malé objekty
- lampa, strom, lavička, odpadkový koš...

ST_Curve

- nedá se vytvořit objekt (křivka) tohoto typu
- 1-dimenzionální geometrický objekt
- Možnost: uzavřená, jednoduchá, kruh jako v OGC

Metody ST_Curve

- ST_Lenght – vrací délku křivky
- ST_StartPoint
- ST_EndPoint
- ST_IsClosed
- ST_IsRing
- ST_CurveToLine

ST_LineString

- objekt tohoto typu je možné vytvořit
- lineární interpolace mezi dvěma ST_Point
- každá dvojice ST_Point definuje úsečku (segment) ST_LineString
- (lineární) kruh je ST_LineString, který je uzavřený a jednoduchý

Metody ST_LineString

- ST_LineString – konstruuje lomenou čáru z textové reprezentace (WKN)
- ST_Point
- ST_NumPoints
- ST_PointN (vrací specifikovaný prvek v množine bodů lomené čáry)

ST_CircularString

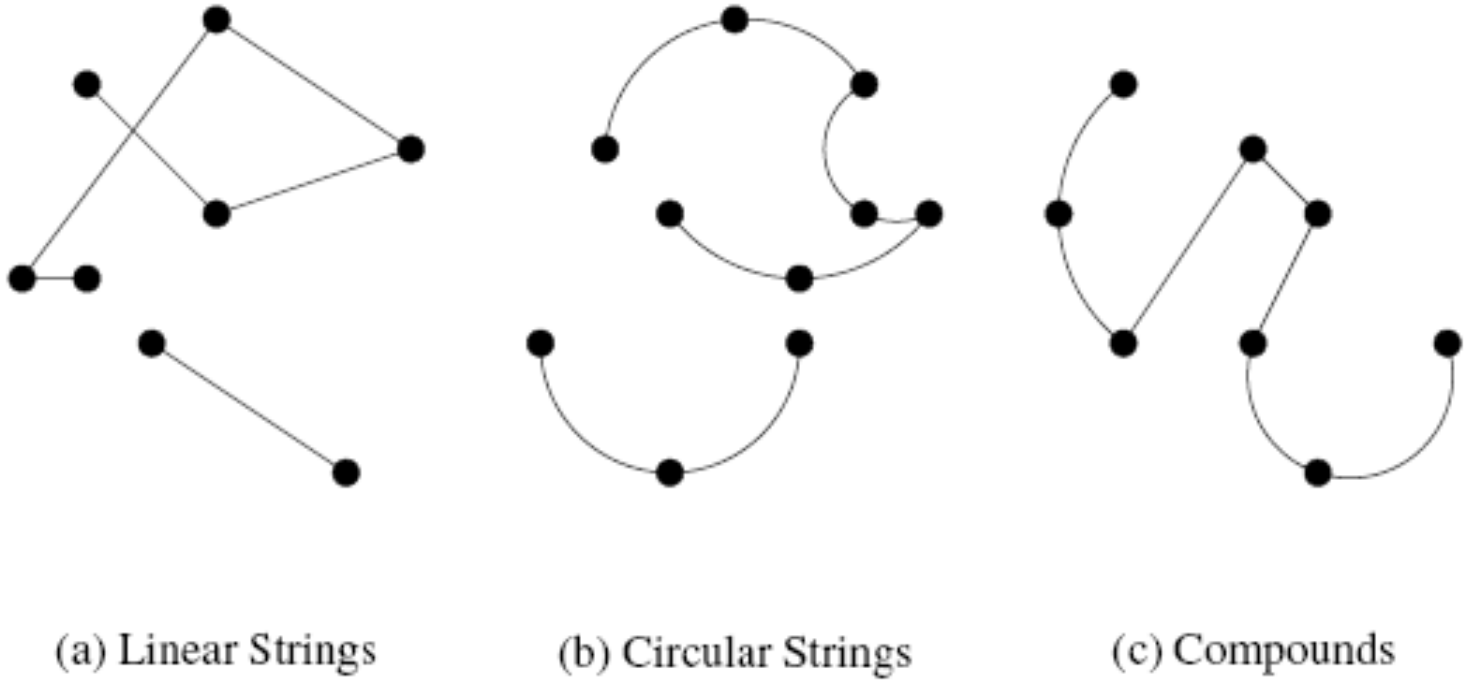
- objekt tohoto typu je možné vytvořit
- jeden nebo více obloukových spojených segmentů
- každý segment sestává ze tří bodů – počáteční, prostřední a koncový bod
- jak se v ST_CircularString nachází více než jeden segment, počáteční bod následujícího segmentu je definován jako koncový bod předcházejícího

ST_CompoundCurve

- objekt tohoto typu je možné vytvořit
- jedna nebo více sousedících křivek typu ST_LineString, ST_CircularString
- koncový bod každé křivky se rovná počátečnímu bodu nesledující

Příklady objektů typu ST_Curve

Příklady: cesta, pobřeží, řeka, vodovod...

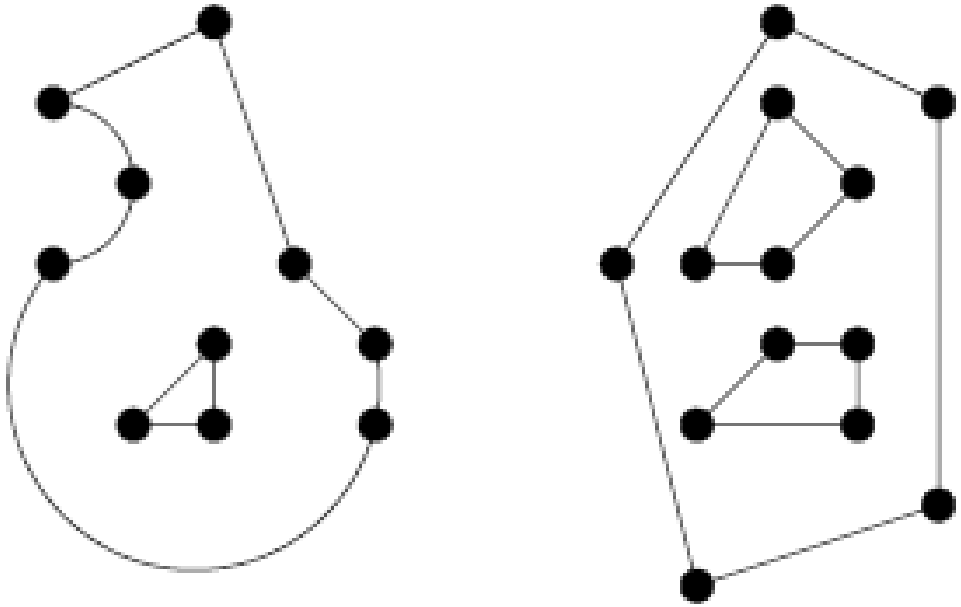


ST_Surface

- není možné vytvořit objekt (plochu) tohoto typu
- 2-dimenzionální geometrický objekt
- definované podobně jako křivky – hranice každé plochy je tvořena z množiny uzavřených jednoduchých křivek

Příklady ST_Surface

- Praktické využití: pozemek, půdorys...



(a) Curve Polygon (b) Polygon

ST_CurvePolygon

- objekt tohoto typu je možné vytvořit
- definován jednou vnější hranicí a 0 nebo více vnitřními
- hranice žádných dvou se neprotínají

Metody ST_CurvePolygon

- ST_CurvePolygon
- ST_ExteriorRing
- ST_InteriorRing
- ST_NumInteriorRing
- ST_InteriorRingN
- ST_CurvePolyNPoly

ST_Polygon

- objekt tohoto typu (mnohoúhelník) je možné vytvořit
- hranice se skládá jen z úseček

ST_Multi*

- ST_MultiPoint
- ST_MultiCurve, ST_MultiLineString
- ST_MultiSurface, ST_MultiPolygon

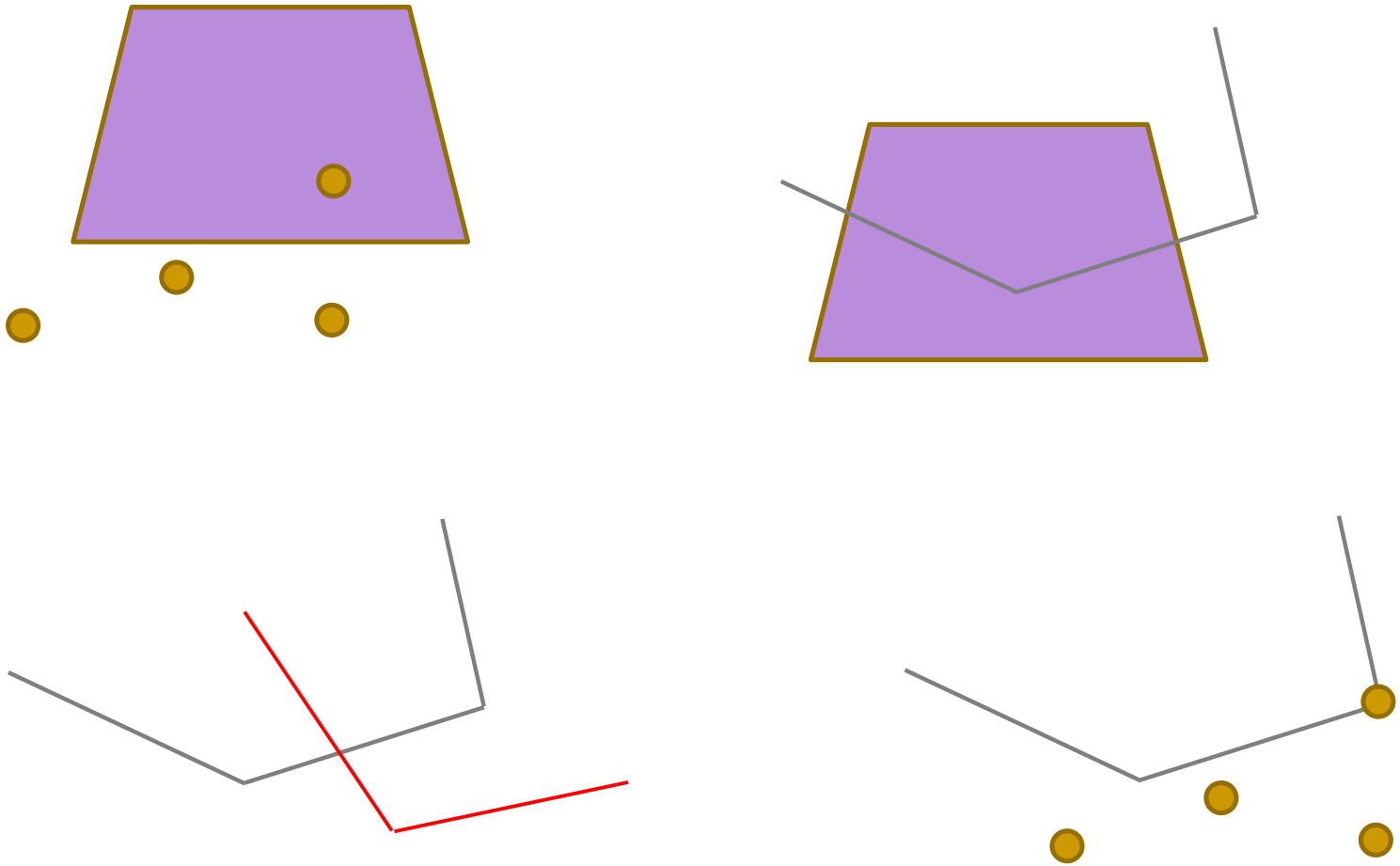
Pz.: objekt typu ST_MultiSurface obsahuje pouze disjunktní plochy

Porovnaní geometrických objektů

- ST_Equals,
- ST_Disjoint
- ST_Intersects (nejobecnější)
- ST_Crosses (viz definice vpředu)
- ST_Overlaps (1 – průnik objektů má stejnou dimenzi jako oba objekty)
- ST_Touches
- ST_Within
- ST_Contains

vracejí 0 nebo 1 (hodnoty jsou typu integer)

Kdy CROSSES vrací 1



Generování nových objektů

tyto metody umožňují vytvořit nové geometrické objekty z existujících

- ST_Buffer
- ST_Envelope (obdélníková hranice)
- ST_ConvexHull (konvexní obal)
- ST_Difference, ST_Intersection, ST_Union

V 2D to je konvexní mnohoúhelník

Práce s vlastnostmi typů

- Příklady: ST_IsValid, ST_IsEmpty, ST_IsSimple
- Každý typ má jiné – viz popisy jednotlivých typů

Další prostorové datové modely

- rozšíření modelu o objekty s neurčitými hranicemi – přidání tříd BBGeometry (podtřída Geometry), BBPolygon, BBMultiPolygon

SQL/MM v databázích

- Oracle 9i, DB2 Spatial extender (UDB), IDS Spatial DataBlade (Informix)
- Podpora prostorových rozšíření v MySQL od verze 4.1
- pro PostgreSQL existuje rozšíření PostGIS

Dva příklady použití

- Pojišťovna

Pojišťovna si chce po záplavách aktualizovat informace o pojištěných budovách a ohrožených oblastech pro správný budoucí odhad rizik z toho pro ni plynoucích závazků.

Příklad 1 - Pojišťovna

Řeky(jméno, průtok, dráha, ohr_obl)

- dráha - křivka typu ST_LineString reprezentující trajektorii toku řeky
- ohrožené_oblasti - množina mnohoúhelníků (typ ST_MultiPolygon) reprezentující oblasti ohrožení řekou při povodních

Budovy(zákazník, ulice, město, PSČ, půdorys)

- půdorys – mnohoúhelník typu ST_Polygon reprezentující tvar základů budovy

Příklad 1 - Pojišťovna

Úloha 1: Po mohutných záplavách je potřebné rozšířit oblasti ohrožené řekou Vltava o 2 km v každém směru.

```
UPDATE Řeky
SET ohr_obl = ohr_obl.ST_Buffer(2, 'KILOMETR')
WHERE jméno = 'Vltava'
```

Příklad 1 - Pojišťovna

Úloha 2: Pojišťovna chce vyhledat zákazníky v těchto nově vytvořených ohrožených oblastech.

```
SELECT zákazník, ulice, město
FROM Budovy AS b, Řeky AS r
WHERE b.půdorys.ST_Within(r.ohr_obl) = 1
```

Příklad 2 - Banka

- Banka si eviduje zákazníky a své pobočky.
- Zákazníci mohou mít více účtů.
- Každý účet je spravován některou pobočkou.
- Banka se snaží zlepšovat kvalitu zvyšováním dostupnosti poboček.

Příklad 2 - Banka

Zákazníci(id_zák, jméno, adresa, typ, místo)

- místo – reprezentace bydliště jako bodu v prostoru, typu ST_Point

Pobočky(id_pob, jméno, manažér, adresa, místo, zóna)

- zóna – oblast činnosti pobočky, typu ST_Polygon

Účty(id_účtu, id_zák, id_pob, typ, stav_účtu)

Příklad 2 - Banka

Úloha 1: Kteří z nejsolventnějších zákazníků (stav_účtu > 500000 Kč) bydlí dále než 20 km od své pobočky?

```
SELECT DISTINCT z.id_zák,z.jméno
FROM Zákazníci AS z JOIN Účty AS u ON (z.id_zák =
   u.id_zák)
WHERE u.stav_účtu > 500000 AND
   z.místo.ST_Distance((SELECT p.místo FROM Pobočky
   AS p WHERE p.id_pob = u.id_pob), 'KILOMETR') > 20
```

Příklad 2 - Banka

Úloha 2: Banka chce nalézt všechny takové dvojice poboček, jejichž průnik jimi pokrývaných oblastí je neprázdný

```
SELECT p1.pob_id, p2.pob_id,
    p1.zóna.ST_Intersection(p2.zóna).ST_AsText()
FROM Pobočky AS p1 JOIN Pobočky AS p2 ON (p1.pob_id <
    p2.pob_id)
WHERE p1.zóna.ST_Intersection(p2.zóna).ST_IsEmpty() = 0
```

Příklad 2 - Banka

Úloha 3: Kvůli zvýšení efektivity se banka rozhodla najít všechny zákazníky, kteří žijí do 10 km od nějaké banky, která nespravuje jejich účet.

```
SELECT z.id_zák, p.id_pob
FROM Pobočky AS p, Zákazníci AS z
WHERE p.místo.ST_Buffer(10,
    'KILOMETR').ST_Contains(z.místo) = 1 AND NOT
    EXISTS (SELECT 1 FROM Účty AS u
                WHERE u.id_zák = z.id_zák AND
                u.id_pob = p.id_pob);
```

Informační schéma

Informační schéma - způsob jak povědět aplikaci používající SQL/MM Spatial, které možnosti SQL/MM Spatial rozšíření může využívat

sestává ze 4 pohledů (views):

- typy sloupců (geometry columns)
- podporované prostorové referenční systémy
- používané jednotky
- odhady velikosti

Informační schéma

- typy sloupců
  - pohled vrací všecky typy sloupců používaných v tabulkách typu ST_Geometry a jeho odvozenin
  - identifikátor sloupce, schéma, tabulka, identifikátor prostorového referenčního systému
- prostorové referenční systémy (též souřadnicové referenční systémy)
  - Co to vlastně je? Používaný souřadnicový systém + transformace mezi různými souřadnicovými referenčními systémy.
  - globální vs. lokální prostorové systémy
  - jméno, identifikátor

Informační schéma

- Používané jednotky
  - je možné využívat různé jednotky pro vzdálenosti, délky křivek, obsahů ploch a pod.
  - identifikující jméno, typ (uhlový, lineární), konverzní faktor na základný typ
- Odhady velikostí
  - specifické metadata např. o maximálních použitelných délkách WKT reprezentací geometrických objektů
- Pohledy jsou uloženy v systémovém katalogu.

Další vývoj

- Poslední verze zahrnuje 3D a 4D objekty
- Geography Markup Language (GML) - ISO 19136
  - zakódování „jednoduchých objektů“ OGC do XML
- Scalable Vector Graphics (SVG)
  - jazyk pro popis 2D grafiky v XML

$\Rightarrow$ požití dotazovacích jazyků nad XML

Další vývoj

Výzvy:

- vágnost v datech i v dotazech (geometrické modely s nepevnými hranicemi),
- vývoj prostorových atributů v čase
  - Př.: STSQL (Spatio-Temporal SQL)

    D.: “Bylo letadlo OK 2903 nuceno se střetnout se sněhovou bouří?”

Doporučená literatura

[1] R. Ďuračiová, D. Cibulka: Databázové systémy v GIS. Návody na cvičenia. STU v Bratislavě, 2015