



SQL/MM

SQL Multimedia and Application
Packages



Motivace (1)

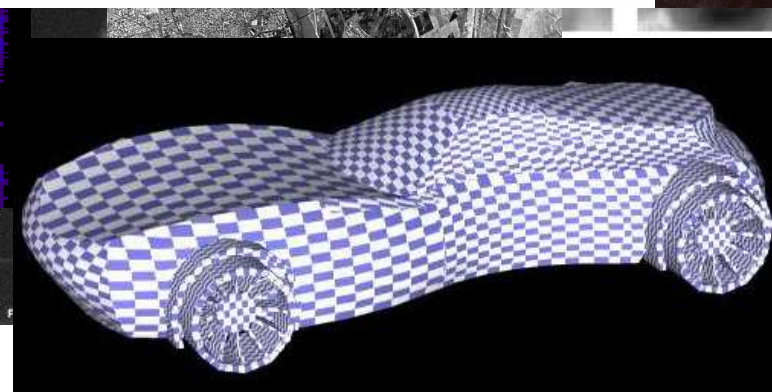
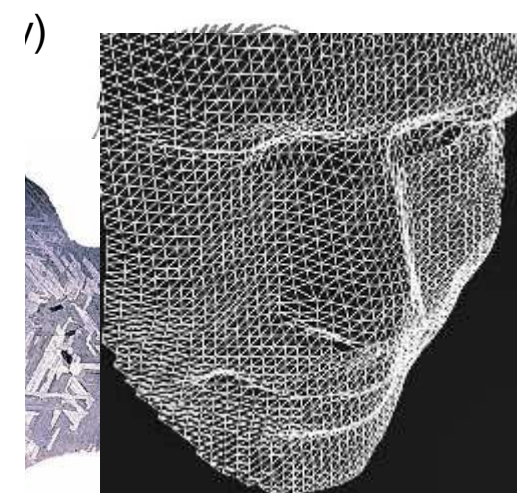
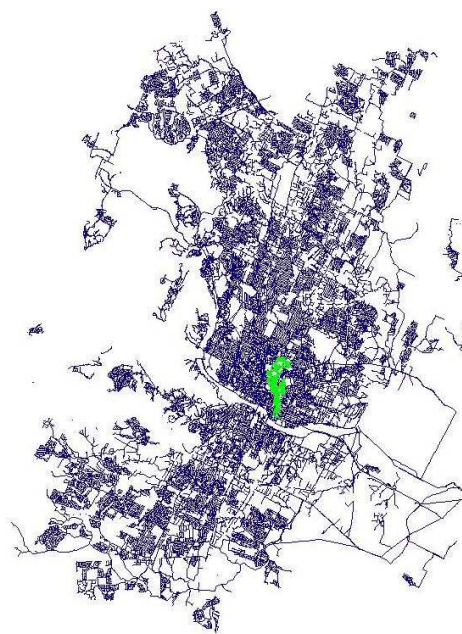
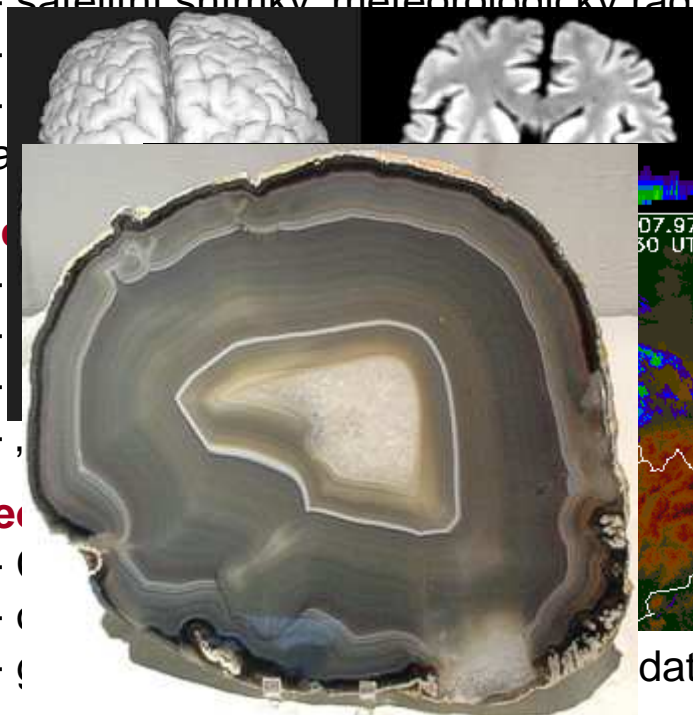
- klasické (relační, objektové) databáze
 - pevně daná struktura i sémantika (schéma databáze, tj. typované atributy, tabulky, integritní omezení, funkční závislosti, dědičnost, atd.)
 - „umělá“ povaha dat (člověkem vytvářené atributy a jednoznačně interpretovatelné atributy)
 - víme co hledáme = stačí dotazy na úplnou shodu
- **multimediální databáze**
 - kolekce obrázků, audia, videa, časových řad, textů, XML, atd.
 - obecně kolekce nestrukturovaných dat (**dokument**)
 - vnitřní struktura i sémantika je skrytá a nejednoznačná
 - závislá na aplikaci, datech, i subjektivitě uživatele
 - „analogová“ povaha dat (digitalizace signálů/senzorových dat)
 - nevíme pořádně co hledáme ani jak se ptát = nestačí dotazy na úplnou shodu

Příklady multimedialních dat

(1)

obrazové databáze

- biometrické databáze (otisky prstů, c
- medicínské snímky (rentgen, tomogr
- satelitní snímky, meteorologický rad

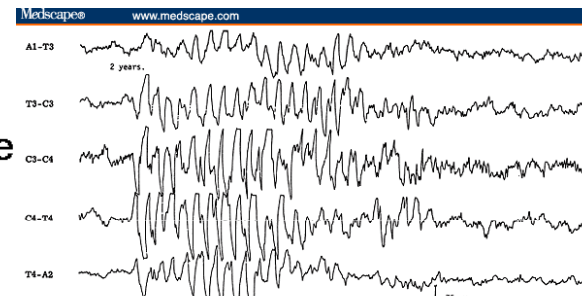


Příklady multimediálních dat

(2)

časové řady, audio, (obecně diskrétní signály)

- vývoj kurzů akcií, měn, atd.
- medicínská data - EEG, EKG, atd.
- řeč (obecně zvuk)
- atd.

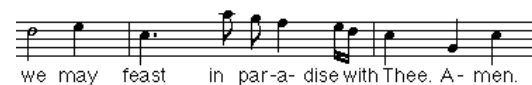
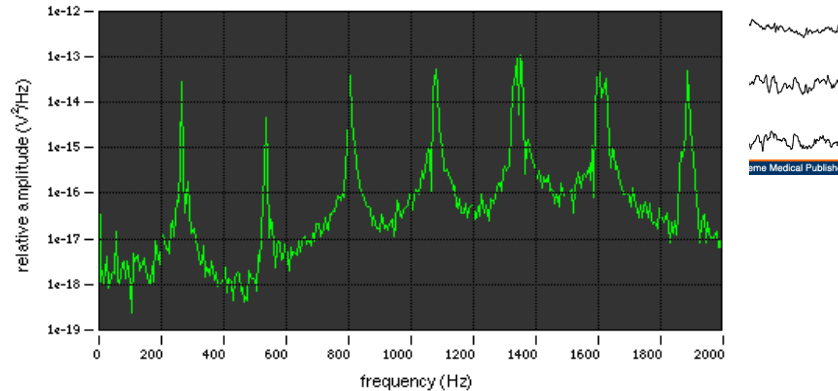
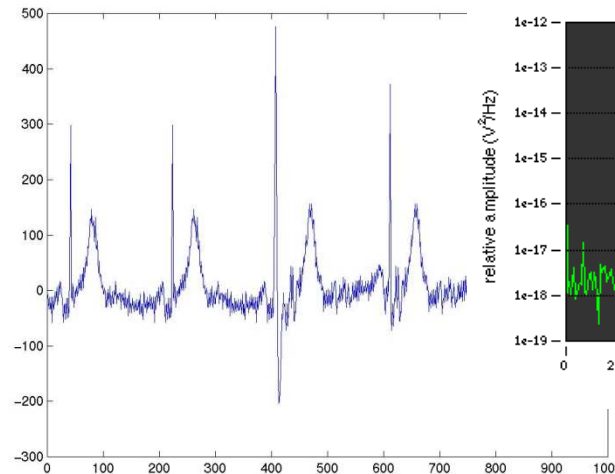


biologick

- chemic
- sekven

melodie

- notové
- MIDI sc



Příklady multimedialních dat

(3)

text, hyper-text

- digitální knihovny, archivy, e-mail
- web
- atd.

„document-centric“ XML data,
semi-strukturovaná data

```
<PERSONA>Young CATO</PERSONA>
<PERSONA>VOLUMNIUS</PERSONA>
<GRPDESCR>friends to Brutus and Cassius.</GRPDESCR>
</PGROUP>
- <PGROUP>
<PERSONA>VARRO</PERSONA>
<PERSONA>CLITUS</PERSONA>
<PERSONA>CLAUDIUS</PERSONA>
<PERSONA>STRATO</PERSONA>
<PERSONA>LUCIUS</PERSONA>
<PERSONA>DARDANIUS</PERSONA>
<GRPDESCR>servants to Brutus.</GRPDESCR>
</PGROUP>
<PERSONA>PINDARUS, servant to Cassius.</PERSONA>
<PERSONA>CALPURNIA, wife to Caesar.</PERSONA>
<PERSONA>PORTIA, wife to Brutus.</PERSONA>
<PERSONA>Senators, Citizens, Guards, Attendants, &c.</PERSONA>
</PERSONAE>
<SCNDESCR>SCENE Rome: the neighbourhood of Sardis: the neighbourhood of Philippi.</SCNDESCR>
<PLAYSUBT>JULIUS CAESAR</PLAYSUBT>
- <ACT>
<TITLE>ACT I</TITLE>
- <SCENE>
<TITLE>SCENE I. Rome. A street.</TITLE>
<STAGEDIR>Enter FLAVIUS, MARULLUS, and certain Commoners</STAGEDIR>
- <SPEECH>
<SPEAKER>FLAVIUS</SPEAKER>
<LINE>Hence! home, you idle creatures get you home:</LINE>
<LINE>Is this a holiday? what! know you not,</LINE>
<LINE>Being mechanical, you ought not walk</LINE>
<LINE>Upon a labouring day without the sign</LINE>
<LINE>Of your profession? Speak, what trade art thou?</LINE>
</SPEECH>
- <SPEECH>
<SPEAKER>First Commoner</SPEAKER>
<LINE>Why, sir, a carpenter.</LINE>
</SPEECH>
```

COMPUTERWORLD

Předplatné | Inzerce | Advertising info | Newsletter | Kontakty

zde zadejte hledané slovo

vynledej rozšířené hledání

Nejčtenější články

Internet Explorer je prý bezpečnější než Firefox (981)
Závady harddisků způsobené nesprávným zacházením (764)
Ikony - nová metoda obrany proti phishingu (703)
Sony chystá zásadní změny ve své činnosti (625)
Širokouhlá řada ThinkPad Z (540)
Katrína ukázala nedostatky u IT (492)

Pioneer vyvinul nové 3D rozhraní

05.10.2005 07:00:00

Firma Pioneer vyvinula nové 3D rozhraní, jehož prostřednictvím lze například psát, kreslit či pohybovat objekty v trojrozměrném prostředí. Nová technologie Floating Interface by měla nabídnout odlišnou technologii, než jakou používají standardní 3D zobrazovací techniky. Jak uvádí výrobce, ke sledování 3D obrazců není potřeba speciálních brýlí, hardwarová konfigurace je vhodná pro hromadnou výrobu a jednodušší je i tvorba aplikací pro zobrazovací systém. Při práci s 3D aplikacemi by měl uživatel získat pocit, jakoby na virtuální obrazovce konečky prstů tlačil objekty nebo vytvářel znaky. To vše se odehrává na 15palcové obrazovce.



A jak celý systém funguje?

Obraz na displeji je zobrazován prostřednictvím pole 3D čoček na přední část virtuální obrazovky. Tato zobrazovací technologie je doplňována psychologickými 3D efekty, jako jsou například různé druhy světelných efektů (stíny objektů, kontrast). Výsledkem sjednocení všech vizuálních

COMPUTERWORLD

Nastal CAS
síti
inteligentních
síti

COMPUTERWORLD 34
vychází 07.10.2005

(Obsah čísla)

Každý nový předplatitel OW získává
15% slevu na kurzy Cleverance
Education Center

Doporučujeme

Business World

Acer si upevnil pozici v oblasti notebooků
Domácí síť ovládne DVD
Monster: Peníze nenahradí
odpočinek od stresu
Proč vládní agenti nefandí IT
SSA nakupuje firmy

Computer world

HP posílá virtualizaci
Intel vyvíjí čipy pro mobilní zařízení
Nizkonapěťové Xeony
OpenView podporí čtyři výrobců
řazení PC



Motivace (2.1)

- **klasické (relační, objektové) databáze**

- dotaz lze jednoduše formulovat, např. pomocí SQL
- dotaz na úplnou shodu přesně určuje jak vypadá plně relevantní a plně nerelevantní možný výstup
- výsledek dotazu není dále strukturován (všechno je stejně relevantní)
- propracované přístupové metody = rychlé vykonávání dotazu

```
SELECT * FROM zamestnanec WHERE vek  
BETWEEN 25 AND 35
```



Motivace (2.2)

- **multimediální databáze**
 - jak vůbec formulovat dotaz?
 - jak dopředu kvantifikovat co pro mně (ještě) je a co (už) není relevantní?
 - co je to vlastně relevance dokumentu k dotazu?
 - jak dotaz provést efektivně (rychle)?



Metody řešení

První část (obecné aspekty, architektury, modelování):

- struktura MDB systémů, modality vyhledávání, dotazy na podobnost
- extrakce vlastností, míry podobnosti, kvalita a rychlost vyhledávání
- mapování a redukce dimenze
- aplikace, ukázky existujících systémů

(důraz na kvalitu vyhledávání)



Metody řešení

Druhá část (implementace, indexování):

- metrické přístupové metody (MAM) vs. prostorové přístupové metody (SAM)
- principy indexování pomocí MAM
- statické MAM, dynamické MAM
- přibližné a pravděpodobnostní vyhledávání
- ostatní...

(důraz na rychlost vyhledávání)

Typy MDB systémů

text-based retrieval systémy

- vyhledávání pouze podle textové anotace (meta-informace)
 - automatické anotování (např. images.google.com využívá textu na stránce, kde je na obrázek odkaz, případně název souboru obrázku)
 - ruční anotace – většinou kvalitnější, anotuje expert, který ví, jak anotovat
- dotazy podobně jako u fulltextových vyhledávačů, tj. množina klíčových slov
- výhoda – využití stávající implementace fulltextových vyhledávačů
- nevýhody
 - nelze aplikovat na neanotované kolekce, ruční anotování je drahé
 - anotace je vždy nějak nepřesná (subjektivní, neúplná, zavádějící, atd.)
 - získané dokumenty mohou být úplně irelevantní
 - nezískali jsme dokumenty, které jsou relevantní - „netrefili“ jsme se do anotace



Typy MDB systémů

- **content-based retrieval systémy**
 - vyhledávání pouze podle obsahu
 - různé metody popisu obsahu
 - výhody
 - vyhledávání podle skutečného obsahu
 - nezávislost na anotaci,
 - nevýhody – mnoho různých metod modelování struktury a sémantiky obsahu, kterou vybrat?
- hybridní systémy
 - kombinují výše zmíněné dva

Modality vyhledávání

■ dotazování (querying)

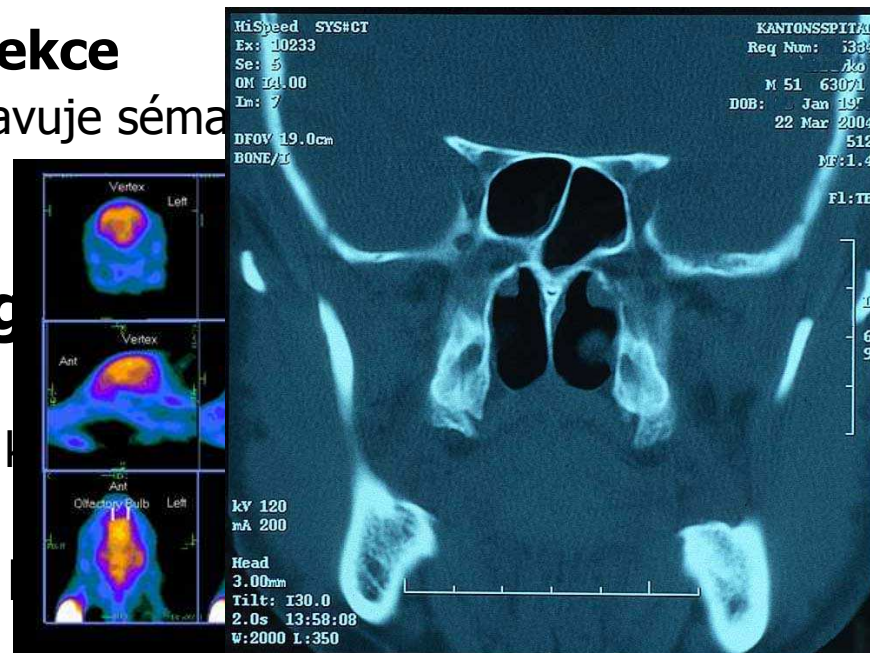
- dotaz v kontextu dokumentu
 - dokument chápán jako databáze, kde hledáme dílčí fragment
 - rozpoznávání/analýza obrazu, vyhledávání v DNA sekvencích, řetězcích, apod.

■ dotaz v kontextu kolekce

- celý dokument představuje séma
- databázový přístup

■ prohlížení (browsing)

- navigace v celé kolekci
 - hierarchická struktura
 - okolí (web, ontologie)
- vhodné pro interaktivní





Dotazování podle podobnosti

- **query-by-example typy dotazů**
 - ptáme se přímo nějakým dokumentem (ať dokumentem z databáze ve které hledáme, nebo z jiným)
 - navíc specifikujeme rozsah dotazu nebo výsledku
 - bodový dotaz
 - **rozsahový dotaz** – práh r
 - **k nejblížších sousedů** - k
 - reverzních k nejblížších sousedů – k
 - a další...

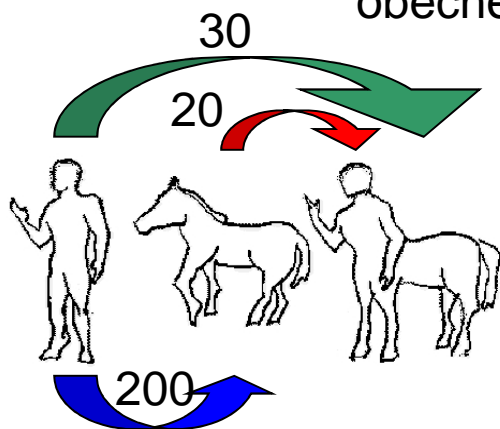
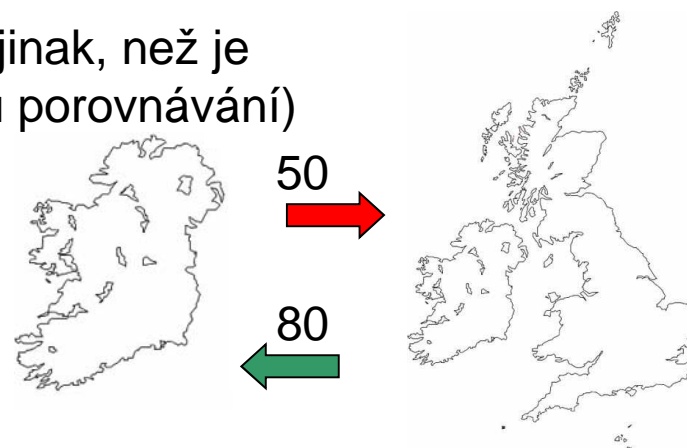
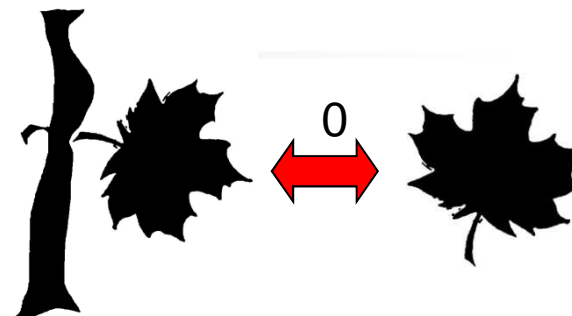
Kritika metrických vlastností

ad **reflexivita**: objekt nemusí být sám sobě podobný

ad **pozitivita**: objekt je maximálně podobný (totožný) jinému objektu

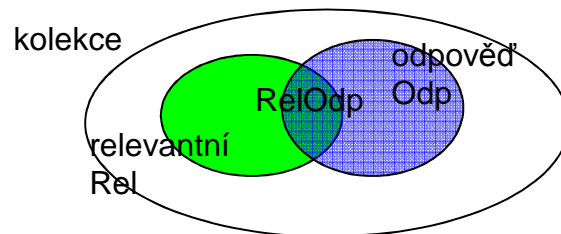
ad **symetrie**: objekt 1 je podobný objektu 2 jinak, než je tomu naopak (záleží na směru porovnávání)

ad **trojúhelníková nerovnost**: obecně neplatí tranzitivita



Kvalita vyhledávání vs. efektivita vyhledávání

- **kvalita vyhledávání** (retrieval effectiveness) je úspěšnost vyhledání dokumentů vzhledem k očekávání uživatele
 - vždy subjektivní, nelze dosáhnout dokonalosti
 - měření na základě subjektivně ohodnocené kolekce
 - nejčastěji *přesnost* $P = |\text{RelOdp}|/|\text{Odp}|$ a *úplnost* $R = |\text{RelOdp}|/|\text{Rel}|$



- **rychlost vyhledávání** (retrieval efficiency) ovlivňuje reálnou použitelnost a škálovatelnost
 - I/O operace, množství výpočtů podobností/vzdáleností, ostatní CPU náklady
 - potřeba speciálních přístupových metod, resp. indexování, sekvenční průchod je u velkých databází nereálný



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

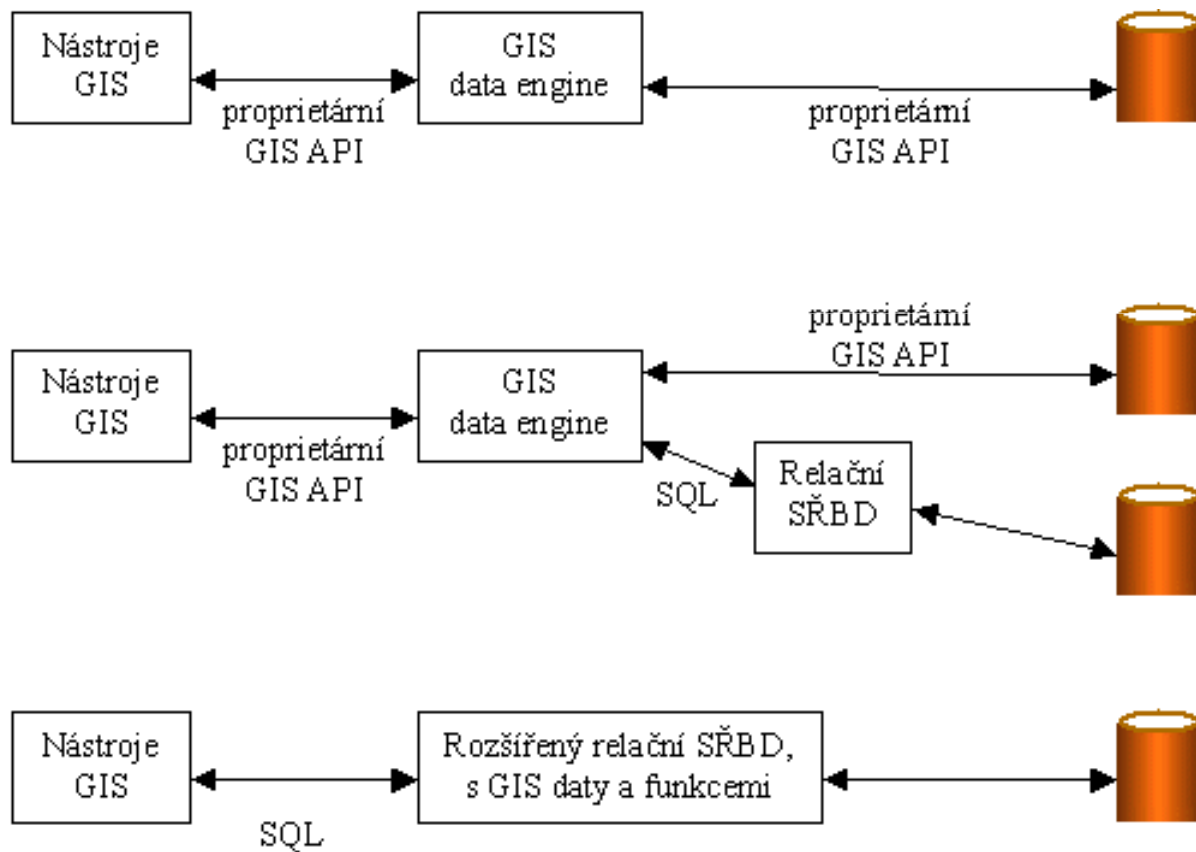
- Nevyvíjet zcela nový dotazovací jazyk, ale využít jazyka SQL
- Proč?
 - SQL je databázový standard
 - V praxi se často vyskytují prostorová data společně s lexikálními daty



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?

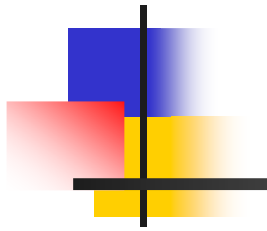
- Ukládání, výběr, dotazování a aktualizaci jednoduchých prostorových objektů
- Reprézentaci prostorových objektů pomocí prostorových datových typů (Spatial types)
- Funkce pro práci s prostorovými objekty



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

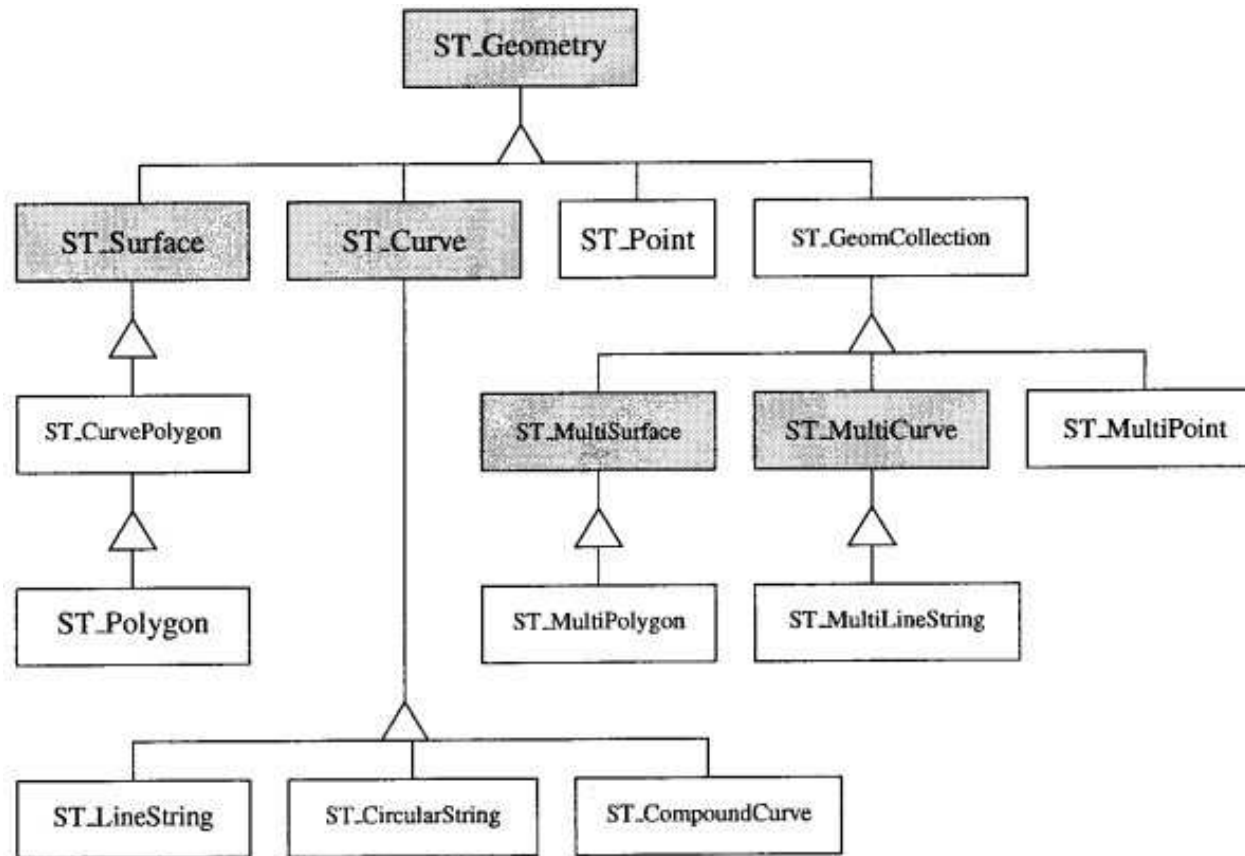




Prostorové objekty

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

SQL geometrický model



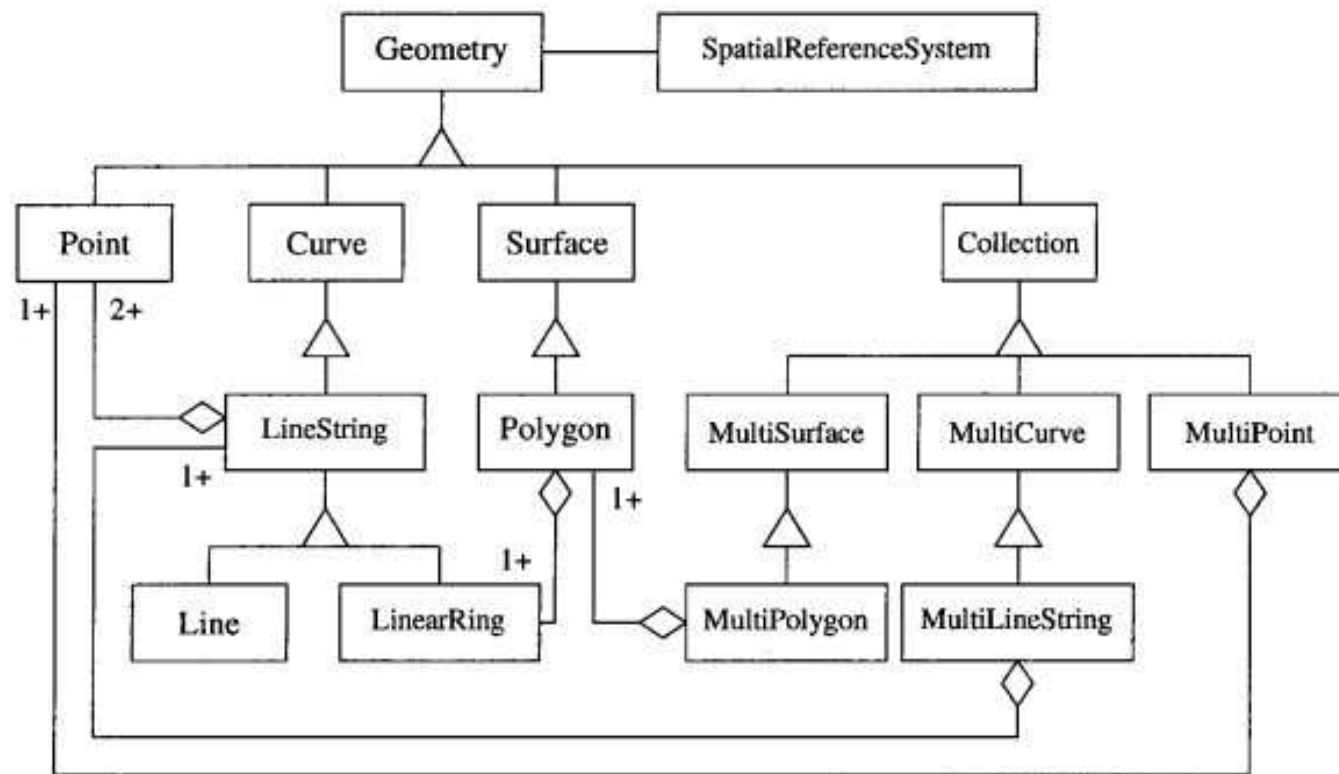


SQL geometrický model

- 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

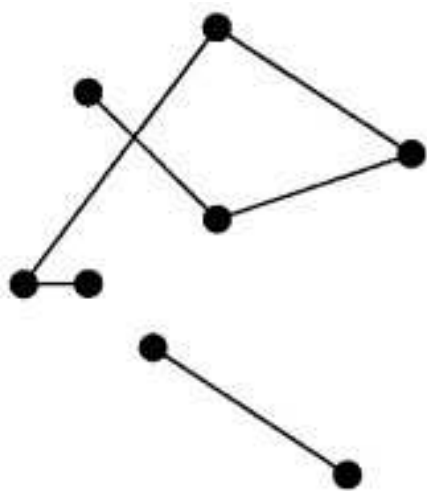
- Bod:
 - ST_Point
 - X a Y souřadnice vzhledem k nějakému systému souřadnic
 - ST_MultiPoint
 - Kolekce bodů



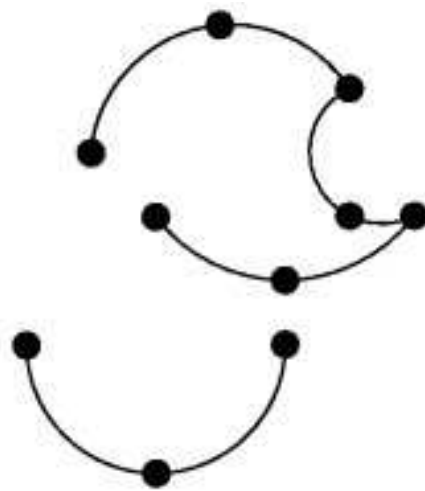
1-dimenzionální objekty

- Křivky:
 - ST_LinearString
 - ST_CircularString
 - ST_CompoundCurve
 - Kombinace ST_LinearString a ST_CircularString
 - ST_MultiLineString
 - Neexistují ST_MultiCircularString a ST_MultiCompoundString

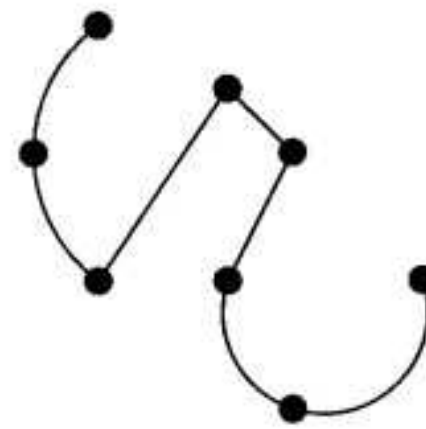
1-dimenzionální objekty



(a) Linear Strings



(b) Circular Strings



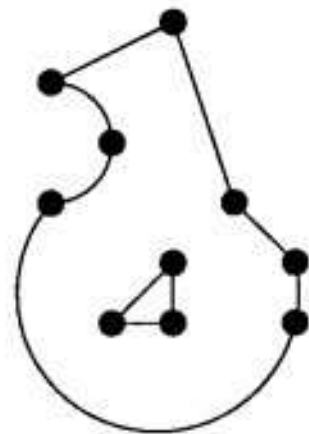
(c) Compounds



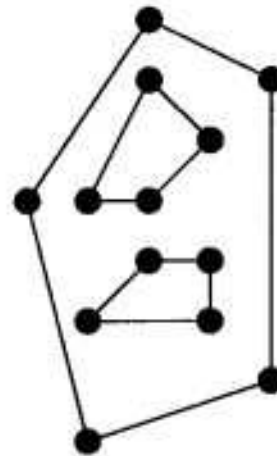
2-dimenzionální objekty

- 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>  
</gml:Polygon>
```



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

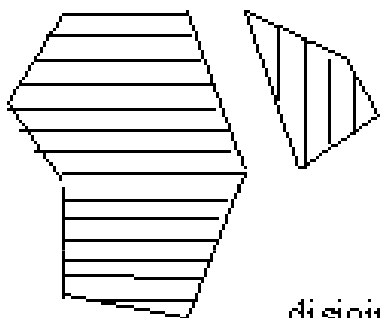
- ST_Equals
 - Test na prostorovou rovnost dvou objektů
- ST_Intersect, ST_Crosses, ST_Overlaps
 - Test na prostorový průnik dvou objektů
- ST_Touches
 - Test, jestli se objekty dotýkají hranicemi, ale neprotínají se
- ST_Within, ST_Contains
 - Test, jestli jeden objekt obsahuje jiný

Porovnávání prostorových objektů

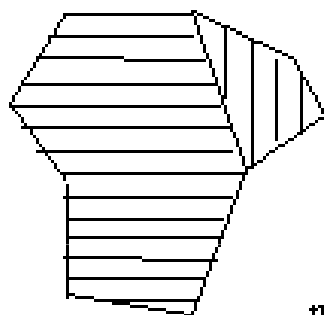


- Metody pro porovnávání objektů jsou založeny na binárních prostorových predikátech
- Všechny výše uvedené metody vracejí 1 pokud je příslušná relace splněna (TRUE), jinak vrací 0
- Další metody
 - ST_Distance

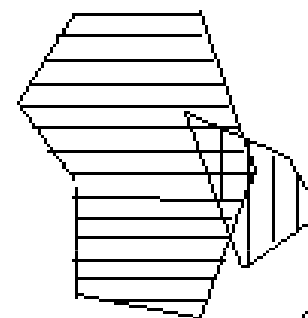
Binární prostorové predikáty



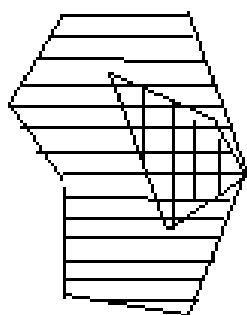
disjoint



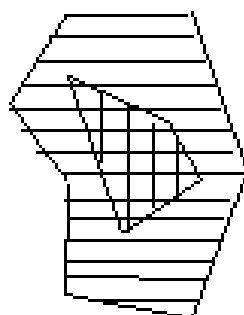
meet



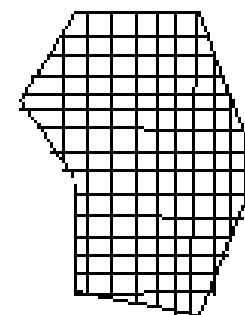
overlap



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

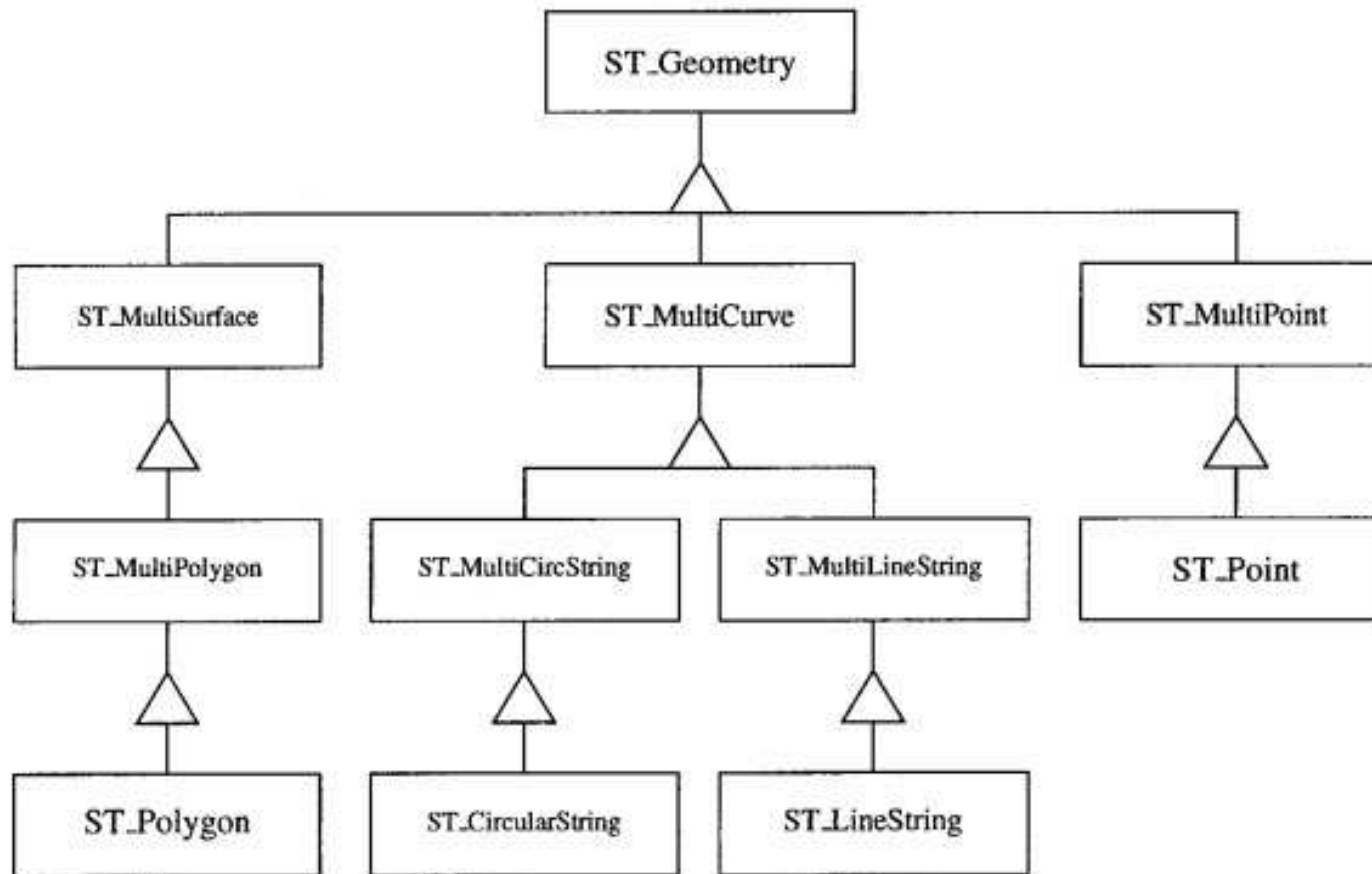
- Výsledek dotazu může být
 - Prázdný objekt
 - Jeden bod (ST_Point)
 - Množina bodů (ST_MultiPoint)
- Jeho získání ale není triviální, neboť ST_Point a ST_MultiPoint se nachází v různých větvích geometrického modelu a tudíž můžeme použít pouze obecných metod definovaných v ST_Geometry



Nedostatky SQL geometrického modelu

- Řešení problému
 - Modifikace geometrického modelu

Modifikovaný geometrický model





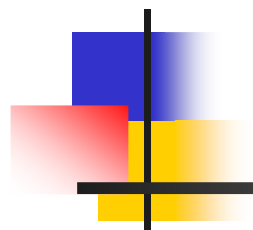
Diskuse k metodám

- 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

The logo consists of a vertical black line intersecting a horizontal black line. To the left of the intersection, there are three overlapping squares: a yellow one on top, a red one on the left, and a blue one on the bottom.

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 pobocky {
    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
}
```


The logo consists of a vertical black line intersecting a horizontal black line. To the left of the intersection, there are three overlapping squares: a yellow one at the top, a red one in the middle, and a blue one at the bottom. The word "Banka" is written in a blue, sans-serif font to the right of the logo.

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 pobočky AS p WHERE p.id=u.id_pobocky),
    'MILES') > 20
```



Banka

- Najdi všechny dvojice poboček, jejichž oblasti působnosti se překrývají

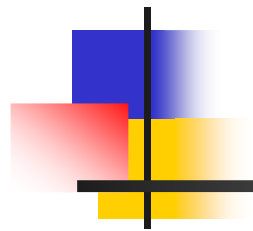
```
SELECT p1.id, p2.id,  
       p1.oblast.ST_Overlaps(p2.oblast).  
       ST_AsText()  
FROM pobočky AS p1 JOIN pobočky AS p2 ON  
     (p1.id < p2.id)  
WHERE p1.oblast.ST_Overlaps(p2.oblast).  
       ST_IsEmpty() = 0
```



Banka

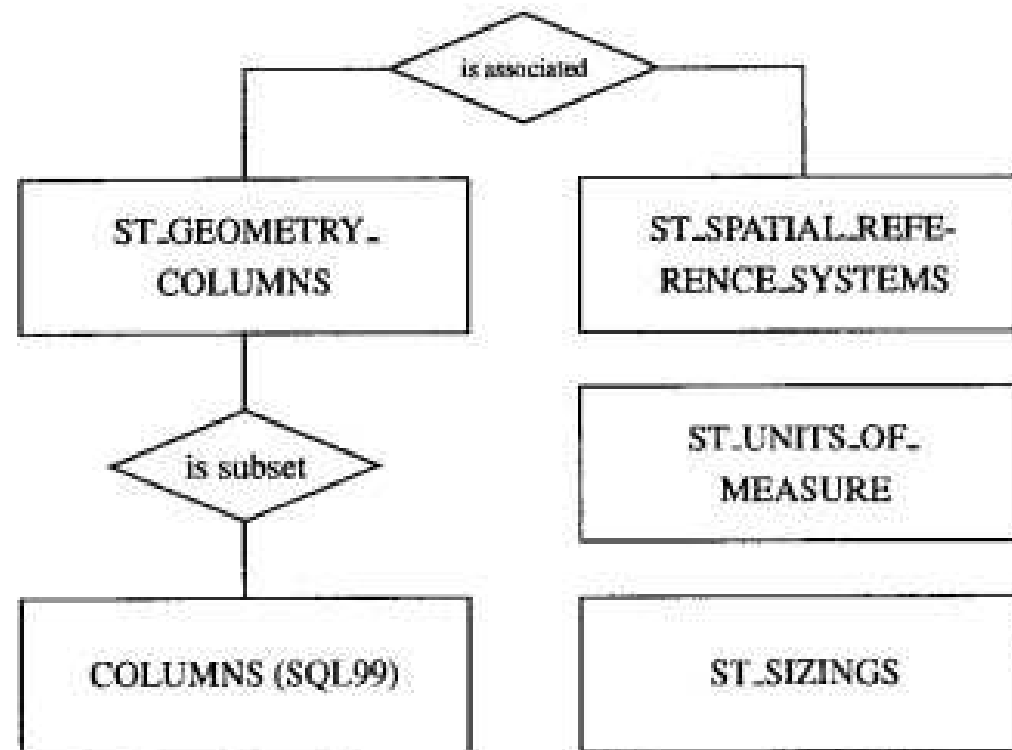
- Najdi všechny zákazníky, kteří bydlí v okruhu 10 mil od nějaké pobočky, která nespravuje jejich účty

```
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
 - Př. ST_MaxGeometryAsText
 - Maximální délka textové reprezentace (WKT) prostorového objektu



Diskuse k datovému katalogu

- **ST_SPATIAL_REFERENCE_SYSTEMS**
 - Celkem primitivní
 - Práce na propracovanější organizaci souřadnicových systému (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



Možnosti implementace

- Jako příklady implementace alternativ (1) a (3) lze uvést prostorové moduly ORACLE 8 (pomocí SQL92) a od verze ORACLE 8i (pomocí objektově relačního přístupu (Pozn.: Další pionýrskou implementací OGC specifikace byl Spatial Database Engine (SDE) vytvořený pracovištěm ESRI (Environmental Systems Research Institute) a využity pro Spatial Extender v SŘBD DB2.).)



Shrnutí

- SQL-MM Spatial
 - Standardizuje ukládání, výběr, dotazování a aktualizaci prostorových objektů
 - Definiuje 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



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

- Aplikační 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



Unstructured Data and Content Management

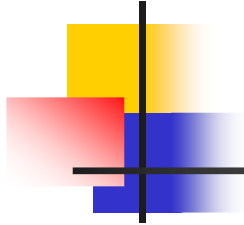
ORACLE 11g Release 2 (11.2)



Oracle Secure Files and Large Objects

- Oracle Secure Files, providing enhanced large object (LOB) storage functionality with file system performance, provides a foundation for a wide range of content management applications.

XML



- Oracle XML DB and XML Developer's Kit enable you to develop performant applications that process XML content and manage XML stored natively in the database.



Oracle Text and Ultra Search

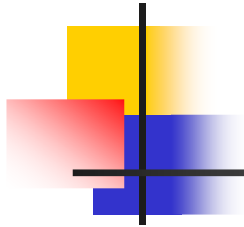
- Oracle Text brings search engine-like full text search capabilities to the Oracle Database. Ultra Search provides a ready-to-use application, while Oracle Text provides a foundation for building your own search applications. Search regular columns, text inside various kinds of binary-format documents, and text, tags, and attributes inside XML documents.



Oracle Spatial and Location Information

- Use features described in these manuals to implement applications that manage data with spatial organization.

Oracle Multimedia



- Oracle Multimedia (formerly known as Oracle interMedia) lets you write applications to manage images, audio, video, and other heterogeneous media data in Oracle databases.



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
- <http://www.dvs.informatik.uni-kl.de/courses/MMDB/WS2003/Vorlesungsunterlagen/SQL-MM.half.pdf> (nemecky)