

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

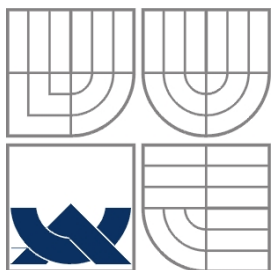
PODPORA PRO PRÁCI S PROSTOROVÝMI DATY
V PROSTŘEDÍ DATABÁZOVÉHO SERVERU ORACLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

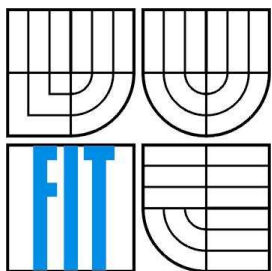
AUTOR PRÁCE
AUTHOR

TOMÁŠ VOLF

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PODPORA PRO PRÁCI S PROSTOROVÝMI DATY V PROSTŘEDÍ DATABÁZOVÉHO SERVERU ORACLE

SUPPORT FOR SPATIAL DATA IN ORACLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ VOLF

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. JAROSLAV ZENDULKA, CSc.

BRNO 2009

Abstrakt

Práce seznamuje s podporou pro práci s prostorovými daty a zmiňuje možnosti využití podpory pro časoprostorová data v prostředí databázového serveru Oracle 11g. Na základě seznámení s podporou pro práci s prostorovými daty byla navržena ukázková aplikace, která využívá této podpory. Ukázková aplikace umožňuje grafické zobrazení prostorových dat, které jsou uloženy v databázi, a informací o daných objektech. Na závěr je uvedeno zhodnocení dosažených výsledků a možnosti pokračování projektu.

Abstract

This work acquaints with support for working with spatial data and mentions the possibility for spatio-temporal data in the Oracle database server 11g. On the basis of acquaintance with support for spatial data was designed sample application that uses this support. The sample application provides to display spatial data, which are stored in the database, graphically and also displays information about spatial objects. Finally, there are reviewed achievements and the possibilities how to continue in the project.

Klíčová slova

Prostorové databáze, prostorové datové typy, prostorové operace, objektově relační model dat, Oracle Database 11g, Oracle shp2sdo Converter, JDBC.

Keywords

Spatial databases, spatial data types, spatial operators, object-relational data model, Oracle Database 11g, Oracle shp2sdo Converter, JDBC.

Citace

Tomáš Volf: Podpora pro práci s prostorovými daty v prostředí databázového serveru Oracle, bakalářská práce, Brno, FIT VUT v Brně, 2009

Podpora pro práci s prostorovými daty v prostředí databázového serveru Oracle

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně
pod vedením doc. Ing. Jaroslava Zendulky, CSc.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Volf
20. května 2009

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce doc. Ing. Jaroslavu Zendulkovi, CSc. za jeho pomoc, rady k práci a čas, který mi věnoval. Dále bych chtěl poděkovat Ing. Jaroslavu Rábovi za seznámení s instalací, správou a administrací databázového serveru Oracle. Další poděkování patří panu doc. Ing. Daliboru Bartoňkovi, CSc. [FAST VUT] za poskytnutí mapových dat.

© Tomáš Volf, 2009

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Podpora prostorových dat v Oracle.....	5
2.1 Model dat.....	5
2.2 Prostorový datový typ SDO_GEOMETRY.....	6
2.3 Souřadné systémy.....	7
2.4 Tolerance.....	8
2.5 Dotazování a filtry.....	8
2.5.1 Primární filtr.....	8
2.5.2 Sekundární filtr.....	9
2.6 Indexování prostorových dat.....	9
2.6.1 Indexování R-stromem.....	9
2.7 Prostorové vztahy a filtrování.....	10
2.8 Topologické vztahy.....	11
2.9 Prostorové procedury a funkce.....	12
2.10 Prostorové agregační funkce.....	14
2.11 Prostorové rozhraní pro Javu.....	15
3 Podpora časoprostorových dat v Oracle.....	16
3.1 Datový typ DATE.....	16
3.2 Datový typ TIMESTAMP.....	16
4 Návrh a příprava aplikace.....	18
4.1 Úvod do problematiky.....	18
4.2 Popis formátu Shapefile.....	18
4.3 Proces konverze dat.....	19
4.4 Diagram případů použití.....	21
4.5 Návrh databáze.....	21
5 Implementace.....	23
5.1 Vytvoření tabulky s polygony.....	23
5.2 Nástroje použité pro realizaci projektu.....	24
5.3 Rozvržení GUI.....	24
5.4 Připojení k databázi.....	25
5.5 Výběr dat z okna dotazu (query window).....	25
5.6 Zobrazení celé mapy.....	28

5.7 Přibližování a oddalování mapy.....	29
5.8 Zobrazení mapy s prostorovými informacemi.....	30
5.9 Zjišťování prostorových informací.....	30
5.10 Zjišťování vzdálenosti.....	32
6 Závěr.....	34
6.1 Zhodnocení výsledků.....	34
6.2 Možnosti pokračování projektu.....	34
Literatura.....	36
Seznam příloh.....	38
Příloha 1.....	39
Příloha 2.....	42
Příloha 3.....	43

1 Úvod

Každý objekt reálného světa se nachází na určité pozici. Kromě své pozice má i další své atributy jako například barvu, vlastníka nebo cenu, atd. Vkládání takových atributů do databáze by nám nemělo činit velký problém, protože pro ně máme elementární datové typy. Kdybychom ale chtěli ukládat do relační databáze informace o prostorech, asi bychom se hodně nadřeli a přehlednost a správa takových tabulek by byla jistě náročná a velmi neefektivní. K tomu nám slouží prostorové rozšíření databází.

Prostorové databáze mají širokou řadu využití napříč obory, například: geografické informační systémy (různé mapové portály), CAD (*computer aided design*) aplikace, aplikace pro návrh budov, 3D modelování, urbanistika, různé dohledové a monitorovací systémy, modelování chemických vazeb, molekulární biologie a další.

Prostorové databáze patří do skupiny takzvaných post-relačních databází. Post-relační databáze jsou databáze pro zpracování a ukládání specializovaných dat s přímou podporou na implementační úrovni. Bez přímé implementační podpory by bylo neefektivní zpracovávat a ukládat specializovaná data. Prostorová rozšíření databázových systémů jsou pro takovéto účely vhodná, protože jsou optimalizovaná pro ukládání, dotazování a zpracování prostorových dat [1].

Podpora prostorových dat je definována ve standardu „*SQL Multimedia and Application Packages (SQL/MM)*“. *SQL/MM* je doplňkem standardu *SQL-99*, který umožňuje ukládat do databáze uživatelsky definované typy a metody. *SQL/MM* podporuje bezrozměrná (bod), jednorozměrná (čára) a dvourozměrná („plochý“ tvar) data. V dalších změnách standardu by mohla být trojrozměrná (objemová tělesa), případně vícerozměrná data [2].

Oracle má pro ukládání prostorových dat vlastní datový typ *SDO_GEOMETRY*, který není s datovým typem *ST_GEOMETRY* (prostorový datový typ dle standardu) vzájemně kompatibilní. Databázový server Oracle má podporu pro ukládání dat dle standardu *SQL/MM* od verze 11g [3].

Prostorové rozšíření databází existuje v celé řadě databázových systémů jako např. Oracle, Microsoft SQL, PostgreSQL, MySQL, a další. Obsahem a cílem této práce je seznámení se s podporou pro práci s prostorovými a časoprostorovými daty v prostředí databázového serveru Oracle, navrhnout a realizovat ukázkovou aplikaci pro využití prostorové podpory.

V práci se vyskytuje pojem geometrie, který znamená rovinný nebo prostorový útvar. Měřítko mapy budeme chápat jako počet metrů na jeden pixel.

Práce je rozdělena do následujících částí: kapitola 2 seznamuje s podporou pro práci s prostorovými daty. Kapitola 3 zmiňuje možnosti využití podpory pro práci s časoprostorovými daty. V kapitole 4 je návrh a příprava ukázkové aplikace pro využití podpory pro práci s prostorovými

daty. V kapitole 5 je popsána vlastní implementace aplikace pojmenované Jasmine. Kapitola 6 obsahuje zhodnocení výsledků a návrhy na pokračování tohoto projektu.

2 Podpora prostorových dat v Oracle

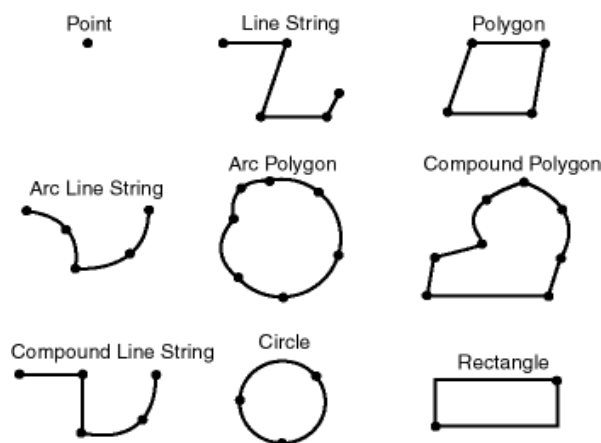
Prostorové rozšíření v Oracle nám nabízí mnoho funkcí a procedur pro zpracování a ukládání prostorových dat. Pro reprezentaci prostorových dat je obvykle používán objektově-relační model dat. Následující podkapitoly jsou zpracovány na základě [5].

2.1 Model dat

Pro ukládání prostorových dat poskytuje databázový server Oracle objektový prostorový datový typ *SDO_GEOMETRY*, který reprezentuje určitý geometrický útvar tvořený bodem, úsečkou, křivkou, kruhem či jejich kombinacemi. Místo několika sloupců, které bychom si v relační databázi s elementárními datovými typy museli vytvořit, budeme mít pouze jeden sloupec *SDO_GEOMETRY* reprezentující danou geometrii. Počet sloupců s prostorovými daty v tabulce není omezen.

Prostorový model dat je hierarchická struktura prvků, geometrií a vrstev. K základním prvkům patří bod, soustava úseček a polygon. Geometrie jsou složeny ze základních prvků, kolekce geometrií tvoří vrstvy.

Prostorový datový typ *SDO_GEOMETRY* nám umožňuje ukládat základní geometrie (body, úsečky, křivky, kruhy a polygony) a geometrie složené ze základních geometrií. Příklady základních a složených geometrií jsou vyobrazeny na obr. 2.1. *SDO_GEOMETRY* podporuje ukládání trojrozměrných a čtyřrozměrných objektů.



Obrázek 2.1: Typy geometrií [5]

K výhodám typu *SDO_GEOMETRY* patří i snadné vytváření a správa indexů a optimální výkon při provádění prostorových dotazů.

2.2 Prostorový datový typ SDO_GEOMETRY

Prostorové rozšíření v Oracle se skládá z objektových datových typů, metod typů a operátorů, funkcí a procedur, které tyto typy využívají.

Prostorový datový typ SDO_GEOMETRY se skládá z atributů:

- *SDO_GTYPE* – určuje typ geometrie
- *SDO_SRID* – určuje souřadný systém
- *SDO_POINT* – určuje polohu bodu v souřadnicovém systému
- *SDO_ELEM_INFO* – pole deskriptorů elementu
- *SDO_ORDINATES* – pole bodů určující polohu v souřadnicovém systému, které tvoří hranici prostorového objektu

Pokud je výslednou geometrií bod, atributy SDO_ELEM_INFO a SDO_ORDINATES obsahují hodnotu NULL, poloha bodu je uložena do atributu SDO_POINT. Pro ostatní geometrie jsou definovány oba atributy SDO_ELEM_INFO a SDO_ORDINATES, atribut SDO_POINT je ignorován.

Atribut *SDO_GTYPE* je čtyřciferné číslo *DLTT*, kde číslice *D* určuje počet rozměrů (dimenzí), *L* určuje lineární referenci (obvykle 0), *TT* určuje typ geometrie. V *TT* může být definován bod (01), čára nebo křivka (02), polygon nebo plocha (03). Další typy geometrií je možno nalézt v [5].

Atribut *SDO_ELEM_INFO* se skládá z tří čísel: *SDO_STARTING_OFFSET*, *SDO_ETYPE* a *SDO_INTERPRETATION*. *SDO_ETYPE* určuje typ elementu. Běžnými hodnotami *SDO_ETYPE* jsou 1 (bod), 2 (soustava úseček, jejichž vrcholy jsou spojeny rovnými čarami), 1003 (vnější polygon, jehož body jsou definovány proti směru hodinových ručiček) a 2003 (vnitřní polygon, jehož body jsou definovány po směru hodinových ručiček). *SDO_INTERPRETATION* určuje, jak máme chápat daný element v *SDO_ETYPE*. Např. polygon můžeme chápat jako obecný polygon (1), obdélník (3, ve speciálním případě také čtverec) nebo kruh (4). Čísla v závorkách označují hodnotu *SDO_INTERPRETATION*. *SDO_STARTING_OFFSET* značí pozici v poli *SDO_ORDINATES* (číslované od 1), kde je uložen první bod pro daný typ elementu. Další hodnoty *SDO_ETYPE* a *SDO_INTERPRETATION* a jejich popis je možno nalézt v [5].

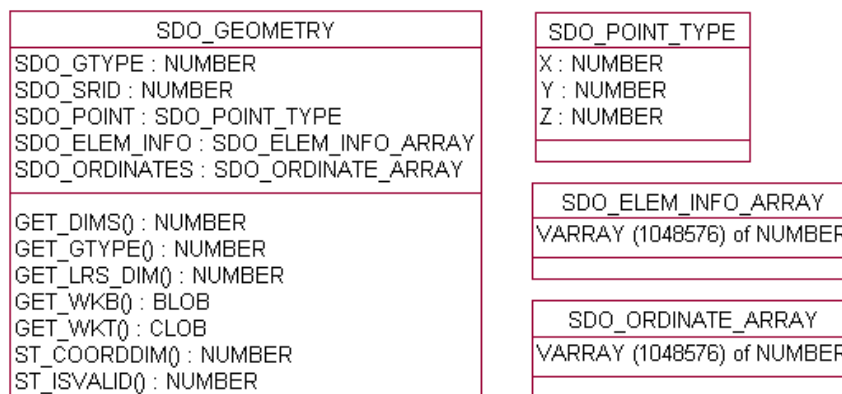
V atributu SDO_ORDINATE_ARRAY může být maximálně 1 048 576 čísel, toto omezení ale závisí na počtu rozměrů (dimenzí):

$$max = \frac{1\,048\,576}{n},$$

kde *max* je maximální počet čísel v SDO_ORDINATE_ARRAY a *n* je počet rozměrů (dimenzí) geometrie.

Struktura prostorového datového typu *SDO_GEOMETRY* a souvisejících typů je vyobrazena na obr. 2.2.

Každý atribut s datovým typem *SDO_GEOMETRY* musí být zaregistrován v tabulce metadat (*user_sdo_geom_metadata*). Ta pro každý atribut *SDO_GEOMETRY* určuje rozmezí hodnot všech dimenzí, které jsou v daném atributu použity.



Obrázek 2.2: SDO_GEOMETRY [4]

2.3 Souřadné systémy

Pomocí souřadného (souřadnicového) systému přiřadíme poloze její souřadnice a zavedeme vztahy mezi množinami souřadnic. Tak můžeme interpretovat množinu souřadnic jako polohu v reálném světě.

Souřadné systémy mohou spadat do skupiny *georeferenčních*, tzn. že se vážou k nějaké konkrétní poloze na Zemi.

Oracle podporuje různé souřadné systémy a poskytuje k nim vzájemné převody. Pokud není žádný souřadný systém explicitně zadán, je implicitně vybrán Kartézský souřadný systém.

Můžeme rozlišovat následující souřadné systémy:

- Kartézský souřadný systém určuje pozici vzhledem k osám. Osy jsou na sebe navzájem kolmé v dvojrozměrném a trojrozměrném prostoru a protínají se ve zvoleném počátečním bodu.
- Geodetické souřadnice (též nazývané jako geografické souřadnice) jsou úhlové souřadnice, které popisují zeměpisnou délku a šířku. Úzce souvisí se sférickými polárními souřadnicemi.
- Projekční souřadnice jsou rovinné Kartézské souřadnice, které jsou výsledkem matematického mapování polohy na zemském povrchu do roviny. Takových mapování existuje mnoho, každé se používá pro konkrétní účel.
- Lokální (místní) souřadnice jsou kartézské souřadnice bez jakéhokoli vztahu k poloze na zemském povrchu. Využívají se například v CAD aplikacích.

2.4 Tolerance

Tolerance je přesnost, s kterou se počítá při prostorových operacích, a udává vzdálenost, při které se ještě mohou považovat dva různé body ležící vedle sebe za stejný bod. Hodnota tolerance je kladné číslo větší než nula a čím menší je hodnota tolerance, tím větší je přesnost. Tolerance se liší v závislosti na použitém souřadnicovém systému.

Pro geodetické souřadnice je tolerance v řádu několika metrů. Hodnota tolerance by neměla být menší než 0,05 (= 5cm). Pokud u geodetických souřadnic zadáme hodnotu tolerance menší, bude se pro prostorové operace uvažovat tolerance 0,05.

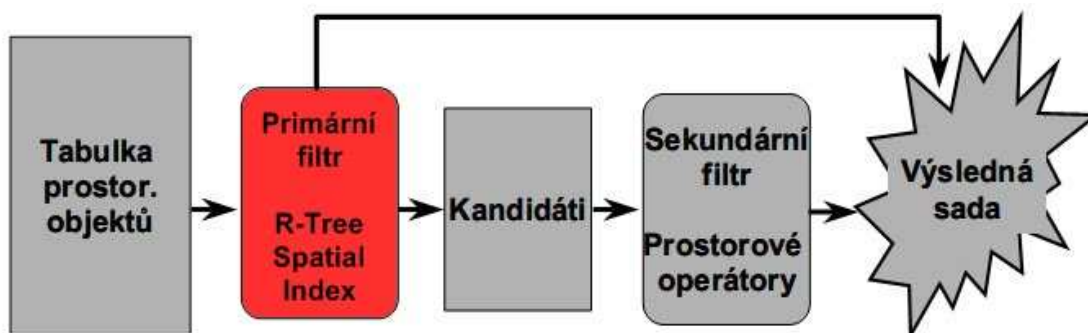
Pro ostatní souřadné systémy je hodnota tolerance v řádu několika jednotek použitých v souřadném systému a datech (např. pokud je měrnou jednotkou míle, pak hodnota tolerance 2 určuje toleranci dvou mil).

Tolerance se zadává:

- při definování metadat pro danou tabulku s prostorovými daty
- jako parametr v mnoha funkcích

2.5 Dotazování a filtry

Prostorové rozšíření v Oracle využívá při vykonávání prostorových dotazů a prostorových spojení dvouvrstvého dotazovacího modelu, tzn. že k dosažení výsledku použije dvě rozdílné operace. Operace jsou označovány jako *primární* a *sekundární filtr*, výsledkem těchto dvou operací je přesná množina výsledků. Obrázek 2.3 zobrazuje aplikaci jednotlivých filtrů a vztahů mezi nimi.



Obrázek 2.3: Postup filtrování [6]

2.5.1 Primární filtr

Úkolem primárního filtru je rychlý výběr množiny přibližných výsledků (přesné výsledky a výsledky jim blízké) pro filtr sekundární, což způsobí snížení režie filtru sekundárního. Primární filtr

porovnává geometrie aproximací pro snížení výpočetní složitosti. Můžeme ho také nazvat jako výpočetně *levný* filtr. Primární filtr vybere z rozsáhlé tabulky prostorových objektů vhodné kandidáty.

2.5.2 Sekundární filtr

Úkolem sekundárního filtru je výběr množiny přesných výsledků na prostorový dotaz. Sekundární filtr již vybere z kandidátů, které získal od filtru primárního, výsledná data, která jsme požadovali. Sekundární filtr je výpočetně náročný, protože potřebujeme získat data již s požadovanou přesností. Pokud bychom neměli filtr primární a měli bychom vybírat přesné výsledky z celé rozsáhlé množiny prostorových dat, bylo by to výpočetně náročné a velmi neefektivní. Můžeme ho nazvat jako výpočetně *drahý* filtr.

2.6 Indexování prostorových dat

Prostorový index, stejně jako index v relačních databázích, má za cíl omezit vyhledávání, ale zde je založeno na prostorových kritériích – průniky a inkluze. U prostorového indexu potřebujeme, aby v indexovaném prostoru našel prostorová data, která mají nějaký vztah s určitým bodem nebo oblastí (např. při dotazování na určitý bod nebo prostor). Dále potřebujeme, aby v indexovaném prostoru našel dvojice objektů, které spolu mají nějaký prostorový vztah (např. pro prostorové spojení).

Prostorový index je považován za index logický. Záznamy v prostorovém indexu se liší v závislosti na umístění geometrie v souřadném systému. Pro indexování prostorových dat se používá index typu R-strom.

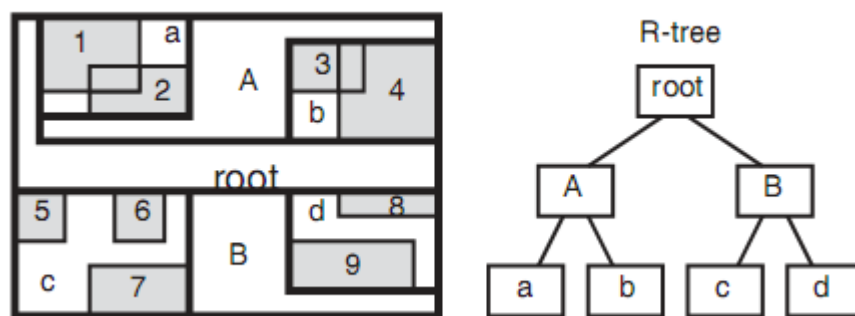
Index se musí založit nad tabulkou s daty, nelze jej vytvořit nad prázdnou tabulkou.

2.6.1 Indexování R-stromem

R-strom může být použit pro indexování až čtyřrozměrných prostorových dat. Index v podobě R-stromu reprezentuje každou geometrii jako její nejmenší ohraničující obdélník, obr. 2.4. R-strom je uspořádaná hierarchická struktura nejmenších ohraničujících obdélníků, obr. 2.5.



Obrázek 2.4: Nejmenší ohraničující obdélník (*MBR*) obklopující danou geometrii (*Geometry*) [5]



Obrázek 2.5: R-strom – hierarchická struktura indexů [5]

2.7 Prostorové vztahy a filtrování

Pro určení prostorových vztahů mezi objekty v databázi se používá již zmíněný sekundární filtr. Prostorové vztahy jsou odvozovány od pozic, na kterých se nacházejí objekty. Nejčastěji jsou prostorové vztahy založeny na vzájemné topologii a vzdálenosti. Např. hranici kraje tvoří množina úseček a křivek, které oddělují města a vesnice v daném kraji od ostatních.

Vnitřní prostor geometrie zahrnuje všechny body tohoto prostoru kromě těch, které leží na hranici geometrie. Vzdálenost mezi geometriemi je nejmenší vzdálenost všech bodů, které leží uvnitř geometrií.

Pro zjištění prostorových vztahů využíváme sekundárních filtračních metod:

- *SDO_FILTER* – určuje, které geometrie mohou mít topologický vztah (všechny topologické vztahy kromě *DISJOINT*) s předanou geometrií
- *SDO_NN* – zjistí nejbližší sousední geometrii pro předanou geometrii
- *SDO_NN_DISTANCE* – zjistí vzdálenost objektu vráceného operátorem *SDO_NN*
- *SDO_RELATE* – zjistí, jestli spolu mají dvě geometrie daný topologický vztah
- *SDO_WITHIN_DISTANCE* – zjistí, jestli jsou od sebe dvě geometrie navzájem vzdáleny do určité vzdálenosti

Operátor *SDO_RELATE* rozlišuje 11 různých vztahů geometrií odpovídajících dále uvedeným topologickým vztahům (viz kapitola 2.8). Kromě toho můžeme použít také samotné topologické vztahy pro filtrování dat. Jejich název je pak vždy uvozen prefixem „*SDO_*“, např. *SDO_EQUAL*.

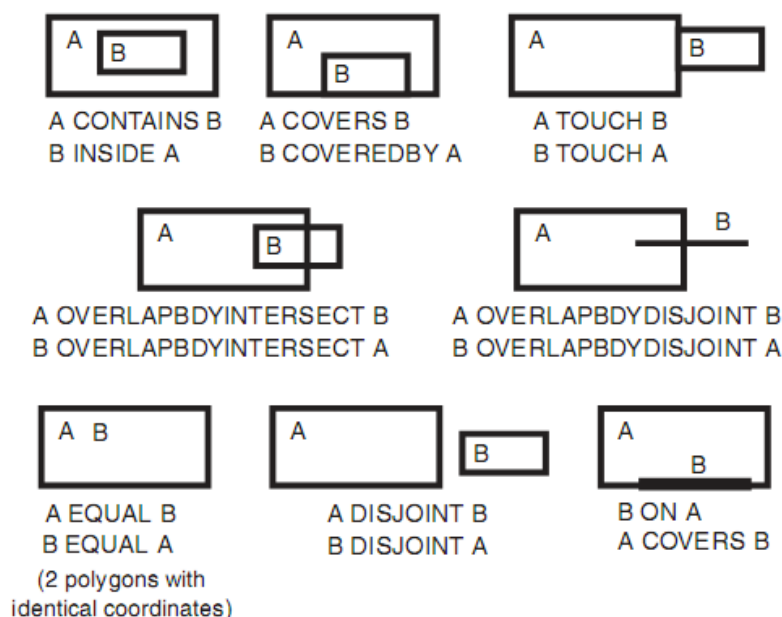
Prostorové operátory, jako např. *SDO_FILTER* a *SDO_RELATE*, poskytují optimální výkon, protože používají prostorový index, proto je nutné mít v tabulce zaveden prostorový index. V SQL dotazu se musí prostorový operátor nacházet v klauzuli *WHERE*. Pokud nejsou dotazované geometrie

ve stejném souřadném systému, v jakém jsou uloženy v databázi, bude na ně použita implicitní transformace.

2.8 Topologické vztahy

U geometrií rozlišujeme následující topologické vztahy, které jsou znázoněny na obr 2.6:

- *DISJOINT* – hranice ani vnitřní prostory geometrií se vzájemně neprotínají
- *TOUCH* – hranice se protínají, ale vnitřní prostory geometrií se neprotínají
- *OVERLAPBYDISJOINT* – hranice jedné geometrie protíná vnitřní prostor geometrie druhé, ale navzájem se hranice geometrií neprotínají, např. úsečka začíná mimo polygon a končí uvnitř polygonu
- *OVERLAPBYINTERSECT* – hranice a vnitřní prostory geometrií se protínají
- *EQUAL* – geometrie jsou totožné – mají stejnou hranici a stejný vnitřní prostor geometrie
- *CONTAINS* – hranice a vnitřní prostor jedné geometrie jsou zcela uvnitř prostoru druhé geometrie
- *COVERS* – vnitřní prostor jedné geometrie je zcela uvnitř prostoru druhé geometrie nebo je uvnitř druhé geometrie pouze část geometrie první a hranice geometrií se protínají
- *INSIDE* – vnitřní prostor jedné geometrie zcela obsahuje hranice a vnitřní prostor geometrie druhé (opak *CONTAINS*)
- *COVEREDBY* – vnitřní prostor jedné geometrie zcela obsahuje geometrii druhou nebo obsahuje část geometrie druhé a hranice geometrií se protínají (opak *COVERS*)
- *ON* – vnitřní prostor a hranice jedné geometrie se nachází na hranici jiného objektu (a druhý objekt obsahuje první) – např. úsečka na hranici polygonu
- *ANYINTERACT* – objekty neleží zcela mimo sebe (všechny topologie kromě *DISJOINT*)



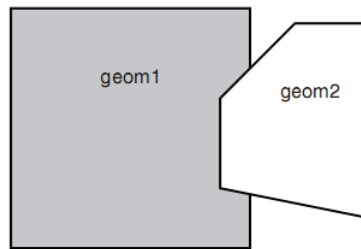
Obrázek 2.6: Topologické vztahy objektů [5]

2.9 Prostorové procedury a funkce

Prostorové procedury a funkce jsou poskytovány jako podprogramy v PL/SQL balíčcích, jako jsou např. *SDO_GEOM*, *SDO_CS* a *SDO_LRS*. Pro tyto podprogramy nemusí být v tabulce zaveden prostorový index. Pokud je v tabulce prostorový index definován, tyto podprogramy ho nevyužívají. V SQL mohou být použity v klauzuli WHERE nebo v poddotazech. Geometrie použité v těchto pod programech musí mít stejný souřadný systém.

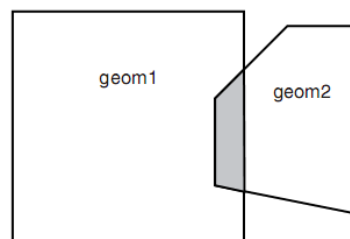
SDO_GEOM je balíček poskytující geometrické operace:

- *SDO_GEOM.RELATE* – zjistí vzájemný vztah dvou objektů
- *SDO_GEOM.SDO_ARC_DENSIFY* – každý kruhový oblouk aproximuje na množinu úseček a každý kruh do polygonu, který se skládá z množiny úseček, které přibližně aproximují tento kruh
- *SDO_GEOM.SDO_AREA* – vypočítá rozlohu dvojrozměrného polygonu
- *SDO_GEOM.SDO_BUFFER* – vytvoří polygon okolo nebo uvnitř dané geometrie
- *SDO_GEOM.SDO_CENTROID* – vrátí těžiště daného polygonu
- *SDO_GEOM.SDO_CLOSEST_POINTS* – vypočítá nejmenší vzdálenost mezi dvěma geometriemi a vrátí body (pro každou geometrii), které vymezují tuto minimální vzdálenost.
- *SDO_GEOM.SDO_CONVEXHULL* – vrátí polygon, který obklopuje danou geometrii
- *SDO_GEOM.SDO_DIFFERENCE* – vrátí objekt, který je topologickým rozdílem dvou geometrií (obr. 2.7)



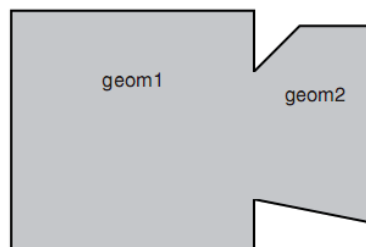
Obrázek 2.7: SDO_GEOM.SDO_DIFFERENCE [5]

- *SDO_GEOM.SDO_DISTANCE* – vrátí vzdálenost mezi dvěma geometriemi
- *SDO_GEOM.SDO_INTERSECTION* – vrátí objekt, který je topologickým průnikem dvou geometrií (obr. 2.8)



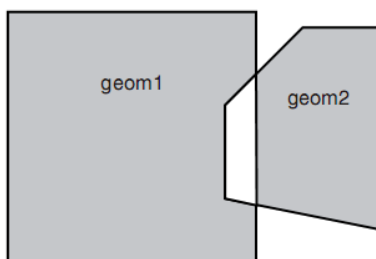
Obrázek 2.8: SDO_GEOM.SDO_INTERSECTION [5]

- *SDO_GEOM.SDO_LENGTH* – spočítá délku (např. bodu nebo úsečky) nebo obvod (např. polygon) geometrie
- *SDO_GEOM.SDO_MAX_MBR_ORDINATE* – vrátí největší hodnotu daného rozměru (souřadnice), který tvoří nejmenší ohraničený obdélník dané geometrie
- *SDO_GEOM.SDO_MBR* – vrátí nejmenší ohraničený obdélník dané geometrie
- *SDO_GEOM.SDO_MIN_MBR_ORDINATE* – vrátí nejmenší hodnotu daného rozměru (souřadnice), který tvoří nejmenší ohraničený obdélník dané geometrie
- *SDO_GEOM.SDO_POINTSURFACE* – vrací bod, který je zaručeně na ploše polygonu
- *SDO_GEOM.SDO_UNION* – vrátí objekt, který je topologickým sjednocením dvou geometrií (obr. 2.9)



Obrázek 2.9: SDO_GEOM.SDO_UNION [5]

- *SDO_GEOM.SDO_VOLUME* – vypočítá objem celistvé trojrozměrné geometrie
- *SDO_GEOM.SDO_XOR* – vrátí objekt, který je doplňkem k topologickému průniku dvou geometrií (obr. 2.10)



Obrázek 2.10: *SDO_GEOM.SDO_XOR* [5]

- *SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT* – zkontroluje, zda je geometrie uložena správně
- *SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT* – zkontroluje, zda jsou všechny geometrie v daném sloupci uloženy správně
- *SDO_GEOM.WITHIN_DISTANCE* – určí, jestli jsou dvě geometrie navzájem vzdáleny do určité vzdálenosti

SDO_CS je balíček poskytující podporu pro práci v různých souřadných systémech a transformací mezi nimi.

SDO_LRS je balíček, který vytváří, upravuje, dotazuje a konvertuje lineární referenční systémy, ale nemění stav databáze.

2.10 Prostorové agregační funkce

Prostorové agregační funkce udělají souhrn výsledků z SQL dotazu, které se týkají geometrických objektů. Prostorové agregační funkce vrací typ *SDO_GEOMETRY*.

Prostorové agregační funkce podporují pouze dvojrozměrná prostorová data, kromě *SDO_AGGR_MBR*, která podporuje dvojrozměrná i trojrozměrná prostorová data.

Příklady agregačních funkcí:

- *SDO_AGGR_CENTROID* – vrátí geometrický objekt, který je těžištěm geometrií v tabulce
- *SDO_AGGR_CONVEXHULL* – vrátí geometrický objekt, který je obálkou geometrií v tabulce
- *SDO_AGGR_MBR* – vrátí minimální ohraničující obdélník, který ohraničuje všechny geometrie v tabulce

2.11 Prostorové rozhraní pro Javu

Oracle nabízí také API rozhraní pro práci s prostorovými daty v programovacím jazyku Java. K tomu slouží knihovna s názvem *sdoapi*, která se skládá z následujících tříd:

- *JGeometry* – třída, která umožňuje ukládat geometrie datového typu *SDO_GEOMETRY* z SQL. Pro načítání prostorových dat do datového typu *JGeometry* slouží metoda *load*, pro kterou musíme data přetypovat na strukturu pomocí (*STRUCT*). Pro uložení do struktury, kterou budeme odesílat SQL příkazem, slouží metoda *store*.
- *JGeometry.Point* – třída, která reprezentuje dvojrozměrný bod typu *double*
- *Jpoint2DD* – třída pro práci s dvojrozměrnými body typu *double*, která umožňuje body porovnávat metodou *equals* a umožňuje nastavovat každou dimenzi bodu zvlášť
- *J3D_Geometry* – třída pro práci s trojrozměrnými geometriemi
- *ElementExtractor* – extraktor prvků z *J3D_Geometry*
- *DataException* – třída pro výjimky při přístupu k mapovým datům

3 Podpora časoprostorových dat v Oracle

Databázový server Oracle neposkytuje přímou podporu pro práci s časoprostorovými daty. Pokud potřebujeme uložit časoprostorová data, můžeme ukládat časová data jako další rozměr dat prostorových - v tomto případě bude čas reprezentován námi vhodně zvoleným časovým razítkem, např. počet sekund od námi definovaného počátku. Další možností, jak zaznamenat časoprostorová data do databáze, je uložit zvlášť prostorová data a zvlášť data časová pomocí datového typu DATE nebo TIMESTAMP.

3.1 Datový typ DATE

Datový typ DATE do tabulky ukládá rok (včetně století), měsíc, den, hodinu, minutu a sekundu časového okamžiku. V Oracle je reprezentován vlastním interním formátem o pevné délce sedmi bytů. Pomocí datového typu DATE můžeme uložit i datum z Juliánského kalendáře.

Datum má na vstupu i výstupu standardně formát DD-MON-YY, tento formát můžeme změnit pomocí parametru NLS_DATE_FORMAT. Pokud zadáváme datum v jiném než standardním formátu, využijeme konverzní funkci TO_DATE, kde v prvním parametru zapíšeme potřebné datum a v druhém parametru uvedeme použitý formát.

Oracle ukládá informaci o čase ve 24 hodinovém formátu HH:MI:SS. Pokud při vkládání data neuvedeme čas, bude implicitně nastaven na čas 00:00:00. Pokud vkládáme pouze čas, datum bude implicitně nastaveno na první den aktuálního měsíce [7].

3.2 Datový typ TIMESTAMP

Datový typ TIMESTAMP je rozšířením datového typu DATE. Ukládá rok, měsíc a den datového typu DATE a z časových informací hodinu, minutu a sekundu. Datový typ TIMESTAMP je přesnější než datový typ DATE, protože můžeme zvolit přesnost sekund až na devět desetinných míst. Přesnost určujeme pomocí TIMESTAMP[přesnost], kde *přesnost* je celé číslo v rozmezí 0 až 9, které udává počet desetinných míst. Implicitní *přesnost* je 6.

Datový typ TIMESTAMP WITH TIME ZONE je variantou datového typu TIMESTAMP, který zahrnuje jméno časového pásma nebo časový rozdíl pásem. Časový rozdíl pásem je rozdíl mezi

místním časem a časem nultého poledníku (Greenwich). Je vhodný pro ukládání a zpracování mezinárodních časů.

Datový typ `TIMESTAMP WITH LOCAL TIME ZONE` je další variantou datového typu `TIMESTAMP`, která zahrnuje časový rozdíl pásem. Na rozdíl od datového typu `TIMESTAMP WITH TIME ZONE` neukládá časový rozdíl pásem jako součást datového sloupce, ale čas přizpůsobí časovému pásmu nastaveném v databázi. Při načítání uživatel dostane data přizpůsobená místnímu časovému pásmu.

Pro načítání dat můžeme využít konverzní funkci `TO_TIMESTAMP` (pro datový typ `TIMESTAMP`) nebo `TO_TIMESTAMP_TZ` (pro datový typ `TIMESTAMP WITH TIME ZONE`) [8].

4 Návrh a příprava aplikace

4.1 Úvod do problematiky

Ukázkovou aplikací demonstrující využití podpory pro prostorová data je jednoduchý geografický informační systém, který zobrazuje mapu a umožňuje získávat informace o objektech jako např. rozlohu vybraného objektu, obvod objektu, střed objektu, sousední objekty vybraného objektu nebo vzdálenost dvou parcel. Jako vstupní vektorová data se podařilo sehnat data katastrálního území obce, která pochází z Fakulty stavební VUT, ve formátu Shapefile (*.shp).

Vstupní data obsahovala body (*bodyKN*) a úsečky (*linieKN*). Pozemky by měly být reprezentovány polygony. Úsečky reprezentují hranice pozemků (a dalších objektů). Body reprezentují identifikaci těchto pozemků (objektů). Úsečky jsou pro vykreslování vhodnější než polygony, protože se hranice neopakují a nekreslí se tedy dvakrát. Ale pro demonstraci v navržené ukázkové aplikaci nedává většina prostorových operací ve spojení s body a s úsečkami rozumný smysl. Pro tyto operace jsou potřebné polygony. Vzhledem k tomu, že nebyla nalezena prostorová operace, která by ze všech úseček v dané vrstvě vytvořila nejmenší polygony, a že předmětem této bakalářské práce nebylo nalezení algoritmu, který by efektivně spojil tyto úsečky v nejmenší polygony, byla zvolena alternativní varianta. Pro ukázkou těchto prostorových operací byla vytvořena demonstrační oblast, kde bylo nalezeno a vytvořeno několik vzorových polygonů manuálně.

4.2 Popis formátu Shapefile

Formát Shapefile je vyvinut společností ESRI. Jedná se o otevřený formát vektorově uložených geografických dat, který je plně dokumentovaný [9].

Formát Shapefile ukládá doplňkové informace a geometrie bez topologií do datových souborů. Geometrie se pro vykreslení ukládají jako tvary zahrnující soubor vektorových souřadnic. Jejich výhodou, oproti jiným zdrojům dat, je rychlejší vykreslování a modifikace, často zabírají méně diskového prostoru a jsou jednodušší pro čtení a zápis, protože nemají režijní náklady se zpracováním topologických datových struktur.

Formát Shapefile ukládá čísla v typech integer a double. Integer má velikost 32 bitů, double má velikost 64 bitů a je s plovoucí desetinnou tečkou, oba datové typy jsou znaménkové. Nejsou povoleny hodnoty jako kladné a záporné nekonečno a NaN (*Not a Number, není číslo*).

Formát Shapefile podporuje geometrie jako jsou bod, úsečky a plochy. Skládá se z několika souborů, z nichž hlavní jsou tři následující:

- *.shp – soubor s přímým přístupem a proměnnou velikostí, kde každý záznam popisuje tvar geometrie se seznamem vrcholů
- *.shx – indexový soubor, kde jsou uloženy indexy geometrií pro rychlé vyhledávání geometrií – každý záznam obsahuje informaci, na které pozici se v souboru *.shp nachází odpovídající geometrie
- *.dbf – databázový soubor formátu dBase, kde je uložena tabulka doplňkových informací (sloupců) ke geometriím – ke každé geometrii je uložen jeden řádek v tabulce doplňkových informací; vztah 1:1 mezi geometrií a řádkem doplňkových informací je zajištěn přes číslo řádku; řádky doplňkových informací musí být uspořádány ve stejném pořadí jako geometrie v souboru *.shp

Bližší informace je možno nalézt v [10].

4.3 Proces konverze dat

Celý proces konverze dat je znázorněn na obr. 4.1. Data, která se podařilo získat, jsou ve formátu Shapefile. Oracle pro tento formát nabízí vlastní konvertor *Oracle Shapefile Converter*. Ten zkonvertuje Shapefile do formátu pro *SQL*Loader*, pomocí kterého pak data načteme do databáze Oracle do tabulek obsahujících prostorová data. Pro konverzi dat potřebujeme dříve zmíněné soubory s geometriemi (*.shp), indexový soubor (*.shx) a soubor s doplňkovými informacemi (*.dbf).

Pro konverzi byl použit konvertor s následujícími parametry:

```
shp2sdo <shapefile> -i <id column> -t <tolerance>
```

- ➔ *shp2sdo* je spustitelný program - název konvertoru
- ➔ *<shapefile>* je jméno zdrojového souboru bez přípony, v našem případě např. *linieKN*. Pokud v parametrech není uveden název tabulky, bude mít implicitně stejné jméno jako zdrojový soubor.
- ➔ *-i <id column>* přidá sloupec se zadaným jménem, který bude primárním klíčem. Pokud není zadáno jméno použije se implicitně „ID“, v našem případě byl použit název ID.
- ➔ *-t <tolerance>* tolerance, která bude v metadatech. V našem případě postačí 0.0005 (= 0,5 mm).

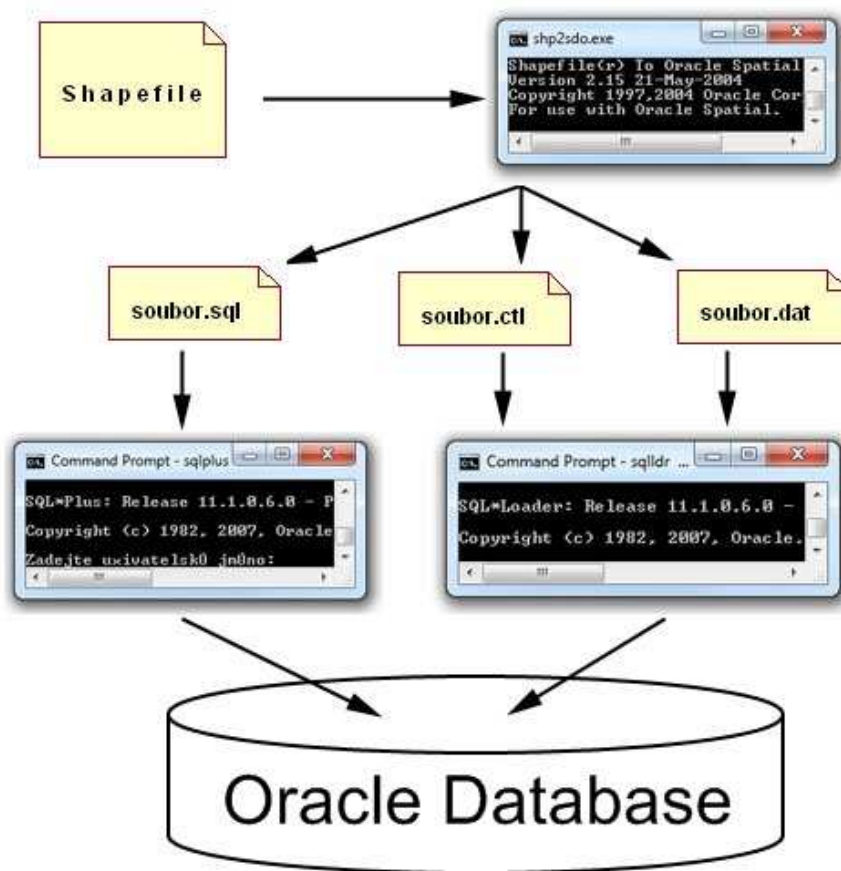
Další parametry konvertoru je možné nalézt v [11].

Konvertor vytvoří tři soubory:

- SQL skript pro vytvoření tabulky, který zároveň naplní tabulku s uživatelskými metadaty (*.sql)
- řídicí soubor pro načtení dat do databáze (*.ctl)
- datový soubor s budoucím obsahem tabulky (*.dat)

Z těchto souborů lze vytvořit odpovídající prostorovou databázi provedením SQL skriptu. Spustíme *SQL*Plus* (z příkazové řádky *sqlplus*), přihlásíme se k databázi a příkazem *@cesta_k_souboru/nazev_souboru.sql* spustíme SQL skript, kterým vytvoříme tabulku s prostorovými daty a zároveň naplníme tabulku s uživatelskými metadaty. V příkazové řádce se přemístíme do složky s vytvořenými soubory, a spustíme *SQL*Loader* příkazem *sqlldr user/pass file.ctl*, kde *user* je uživatelské jméno a *pass* heslo k přístupu k databázi a *nazev_souboru.ctl* je řídicí soubor pro *SQL*Loader*.

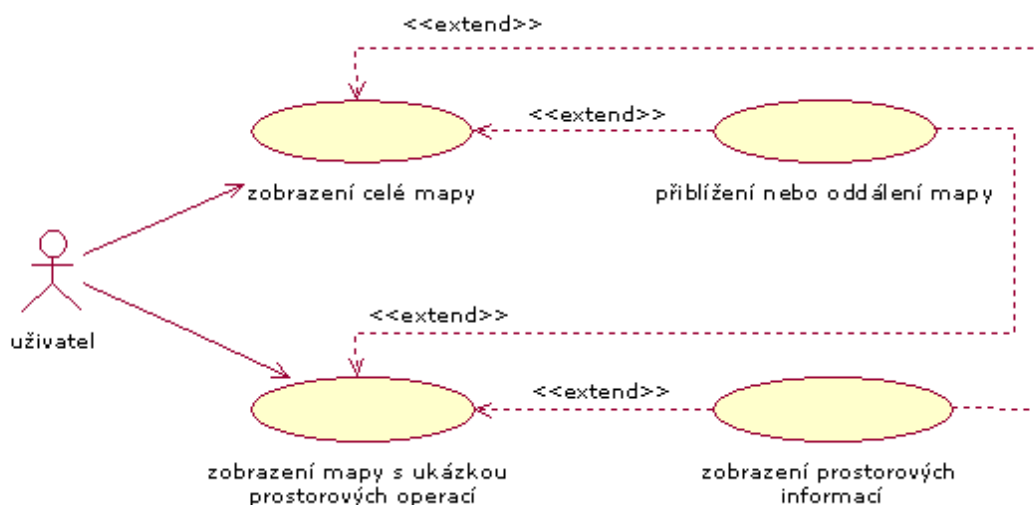
Pokud se nám nepodaří nahrát data pomocí *SQL*Loaderu* a nahrávání skončí s chybou „chyba při konverzi dat, ORA-01722: neplatné číslo“, musíme nahradit desetinné tečky v datovém souboru (*.dat) za desetinné čárky.



Obrázek 4.1: Proces konverze dat

4.4 Diagram případů použití

Po spuštění aplikace se uživateli zobrazí mapa katastrálního území, která ukazuje hranice pozemků. Uživatel si může dle potřeby mapu přibližovat nebo oddalovat, v tomto případě se poloha, na kterou klikne při zvětšování (nebo zmenšování) stane středem nového zobrazení mapy. Dále má uživatel možnost zobrazit si detailněji úsek mapy, na kterém je využita podpora pro práci s protostorovými daty – uživatel dostane informace o pozemku jako např. identifikaci pozemku, celkovou rozlohu pozemku, obvod pozemku (tzn. např. délku plotu), střed pozemku, sousední pozemky nebo vzdálenost dvou parcel. Na obr. 4.2 je znázorněn diagram případů použití ukázkové aplikace Jasmine.



Obrázek 4.2: Diagram případů použití ukázkové aplikace Jasmine

4.5 Návrh databáze

Dvě tabulky již máme vytvořené procesem konverze dat. Jsou to tabulky *bodyKN* (tabulka bodů) a *linieKN* (tabulka úseček).

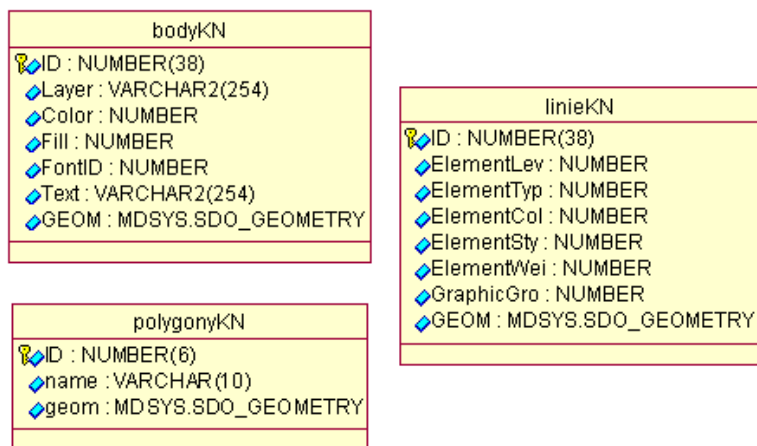
Tabulka *bodyKN* není ve výsledné ukázkové aplikaci Jasmine využívána. Byla využita pouze pro vyhledání identifikace manuálně vytvořených polygonů. Tabulka *bodyKN* obsahovala spoustu zbytečných atributů, které byly ve většině případů shodné, proto byla v rámci udržení přehlednosti a zmenšení velikosti dat zkrácena na sloupce šest (*Layer*, *Color*, *Fill*, *FontID*, *Text* a *GEOM*) a k nim byl přidán sloupec *ID*, který je primárním klíčem. Z těchto sloupců byl využit pouze sloupec *Text* a *GEOM*, kde *Text* reprezentuje číslo parcely a *GEOM* bod na mapě, kde se má identifikace zobrazit, tzn. někde na vnitřní ploše parcely. Ostatní sloupce byly ponechány pro případné budoucí použití.

Tabulka *linieKN* je ve výsledné ukázkové aplikaci Jasmine použita pro vykreslování hranic. Tabulka obsahuje 7 sloupců (*ElementLev*, *ElementTyp*, *ElementCol*, *ElementSty*, *ElementWei*,

GraphicGro a *GEOM*) a k nim byl přidán sloupec *ID*, který je primárním klíčem. Ukázková aplikace Jasmine využívá pouze sloupce *ElementCol* a *GEOM*. Sloupec *ElementCol* reprezentuje barevné odlišení jednotlivých objektů mapy (např. parcely, zastavěné plochy), můžeme o něm tedy říci, že reprezentuje vrstvy mapy. Rozlišuje elektrická vedení (0), parcely (2), hranice katastrálního území (3), zastavěné plochy (4); další (5, 37 a 53) byly vyhodnoceny jako zbytečné, protože vykreslují jednu a tu samou úsečku. Sloupec *GEOM* reprezentuje úsečky vykreslující dané objekty nebo jejich hranice.

Pro účely využití prostorových operací byla vytvořena tabulka *polygonyKN*, kam budou ukládány vytvořené polygony. Tabulka *polygonyKN* má tři sloupce (*ID*, *name* a *geom*), kde *ID* je primární klíč, *name* je unikátní identifikace parcely jejím číslem (10 znaků) nebo hodnotou *NULL* (parcely bez označení) a *geom* vymezuje polygon, který patří dané parcele. Tabulka *polygonyKN* by mohla mít spoustu dalších doplňkových informací o parcelách, ale úkolem bylo zaměřením se na prostorové operace, proto má pouze jeden sloupec s doplňkovými informacemi.

Obr. 4.3 znázorňuje tabulky s prostorovými daty (*bodyKN*, *linieKN* a *polygonyKN*) uložené v databázi.



Obrázek 4.3: Vytvořené tabulky a jejich typy

5 Implementace

5.1 Vytvoření tabulky s polygony

Pro ukázkou dalších prostorových operací je třeba vytvořit tabulku *polygonyKN*, která bude obsahovat polygony. Tu vytvoříme příkazem SQL:

```
CREATE TABLE POLYGONYKN (  
    ID          NUMBER(6)          PRIMARY KEY,  
    name        VARCHAR(10)        UNIQUE,  
    geom        MDSYS.SDO_GEOMETRY  
);
```

Dále je pro tabulku třeba zaregistrovat příslušná metadata v tabulce *user_sdo_geom_metadata* příkazem SQL:

```
INSERT INTO USER_SDO_GEOM_METADATA  
    (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)  
VALUES ('POLYGONYKN', 'GEOM',  
    MDSYS.SDO_DIM_ARRAY(  
        MDSYS.SDO_DIM_ELEMENT('X', -585516, -585310, 0.0005),  
        MDSYS.SDO_DIM_ELEMENT('Y', -1145116, -1144908, 0.0005)  
    ), NULL  
);
```

TABLE_NAME označuje jméno tabulky a *COLUMN_NAME* jméno atributu, pro které se metadata registrují. *DIMINFO* zahrnuje informace o každé dimenzi, kterou v daném atributu využíváme (např. pro dvojrozměrná data musíme mít zaregistrovanou dvě osy, např. *X* a *Y*). Pomocí *SDO_DIM_ELEMENT* vkládáme informace o každé dimenzi, kterou v datech používáme, a ty pak pomocí konstruktoru *SDO_DIM_ARRAY* vložíme do *DIMINFO*. *SDO_DIM_ELEMENT* určuje název dimenze (*SDO_DIMNAME*), pro kterou se data vztahují, minimální (*SDO_LB*) a maximální (*SDO_UB*) hodnotu v dané dimenzi a toleranci (*SDO_TOLERANCE*). *NULL* na konci příkazu SQL reprezentuje souřadný systém použitý pro všechny geometrie daného atributu. Pokud není spojen s geometriemi žádný konkrétní souřadný systém, pak se zadává *NULL*.

Připravenou tabulku naplníme vytvořenými polygony a nakonec pro tabulku vytvoříme prostorový index. Prostorový index se nemůže vytvářet nad prázdnou tabulkou. Ostatní tabulky již

máme v databázi nahrané, ale potřebujeme k nim přidat ještě prostorový index, který budeme vytvářet obdobným způsobem:

```
CREATE INDEX POLYGONYKN_IDX ON POLYGONYKN(GEOM)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Celý skript pro vytvoření tabulky polygonů, zaregistrování metadat a naplnění tabulky daty (`polygonyKN.sql`) a skript pro doplnění prostorových indexů k ostatním tabulkám (`spatial_indexes.sql`) je přiložen na CD.

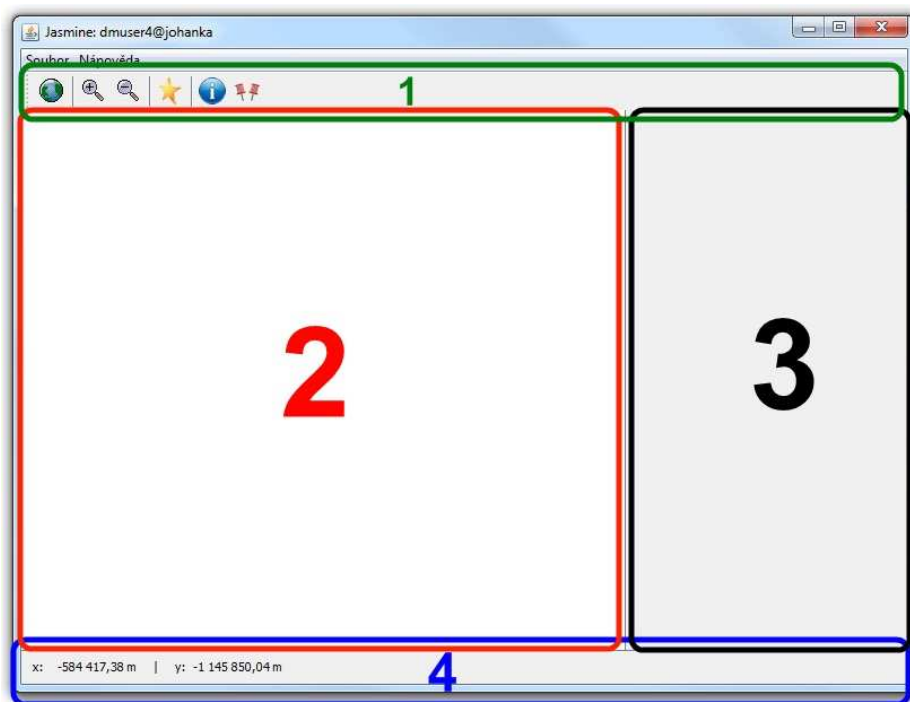
5.2 Nástroje použité pro realizaci projektu

Pro realizaci ukázkové aplikace byl použit programovací jazyk Java verze 6 (update 13). Pro vytváření aplikace bylo použito vývojové prostředí NetBeans IDE verze 6.5.1. Aplikace byla vyvíjena s použitím databázového serveru Oracle verze 11.1.0.6.0, testována byla také na školní verzi databázového serveru Oracle verze 10.2.0.4.0 na serveru *berta.fit.vutbr.cz*. Ukázková aplikace byla vyvíjena a testována v prostředí operačního systému Windows Seven Beta.

5.3 Rozvržení GUI

Rozvržení grafického uživatelského rozhraní znázorňuje obrázek 5.1, kde je oblast s číslem:

1. nástrojová lišta s ikonami, kde si uživatel vybírá funkce aplikace (zobrazení celé mapy, zvětšování, zmenšování, výběr oblasti s vytvořenými vzorovými polygony, zjištění informací, zjištění vzdálenosti)
2. panel, do kterého se vykreslují mapová data
3. panel, do kterého se vypisují informace o vybrané poloze na mapě (v oblasti, kde se nachází vytvořené polygony) jako např. identifikace pozemku, rozloha pozemku, obvod pozemku, střed pozemku, sousedící pozemky nebo informace o vzdálenosti vybraných parcel
4. informační lišta, kde se uživateli po najetí na mapu zobrazí informace o aktuální poloze, nad kterou se nachází kurzor myši



Obrázek 5.1: Ukázková aplikace Jasmine – ukázka rozvržení aplikace

5.4 Připojení k databázi

Aplikace se k databázi připojuje přes rozhraní *JDBC*. K tomu potřebujeme ovladač, což je specifická třída, která je obvykle dodávána výrobcem databáze, v našem případě Oracle.

URI řetězec pro přihlášení k databázi bude ve formátu:

```
jdbc:oracle:thin:login/heslo@adresa_serveru:port:sluzba
```

JDBC označuje JDBC spojení, *oracle* označuje použitý JDBC protokol a *thin* je sub-protokol (nebo také typ spojení).

5.5 Výběr dat z okna dotazu (query window)

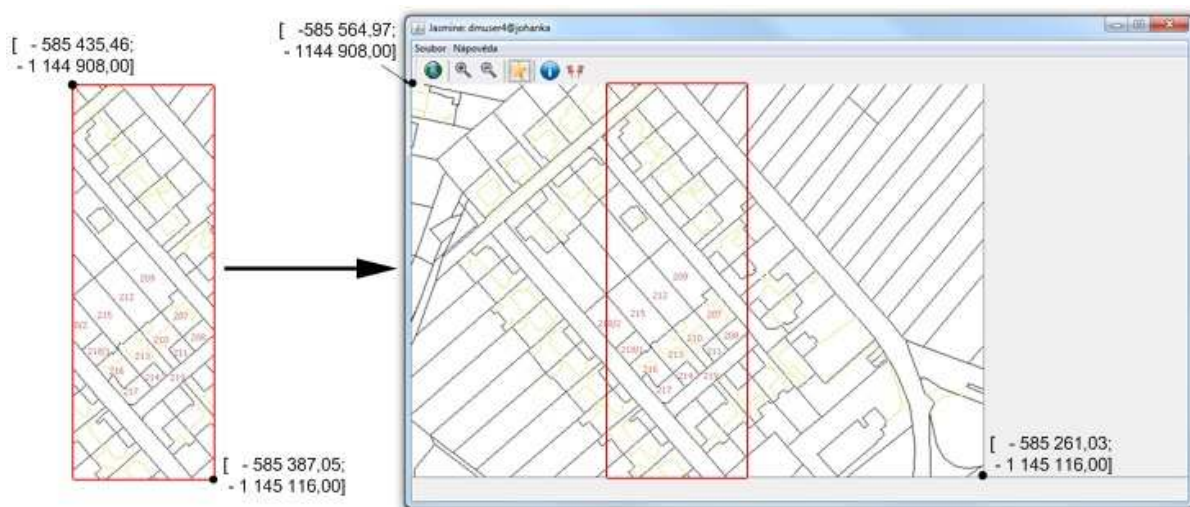
Pro zobrazování mapy bychom potřebovali mít nějakou univerzální funkci, která by nám zařídila výběr prostorových dat z databáze pro aktuální zobrazení a postarala se o jejich zobrazení. Takovou funkčnost nám zajišťuje právě metoda pro výběr dat z okna dotazu.

Metoda potřebuje znát rohové body obdélníkového (ve speciálním případě čtvercového) výřezu mapy, který má být zobrazen. Pro svoje potřeby využívá data uložená v třídních proměnných. Může se stát, že tento obdélník bude nepoměrný vůči oknu, do kterého mapu zobrazujeme. To znamená, že by bylo jiné měřítko pro šířku mapy a jiné pro výšku mapy. V takovém případě by aplikace při zobrazení mapu deformovala. To nemůžeme připustit, proto aplikace kontroluje poměr velikostí

stran obdélníkového výřezu ke stranám panelu, kam se mapa zobrazuje. Pokud není výřez poměrný k oknu, aplikace jej přerozměruje dle vzorce tak, aby byl zobrazen celý obdélníkový výběr. To je znázorněno na obr. 5.2, kde aplikace upravuje obdélníkový výběr (na obr. vlevo) do panelu, kam se mapa zobrazuje (na obr. vpravo), o velikosti 698 x 478 pixelů. Aby byl zobrazen celý obdélníkový výběr musí být upravena šířka obdélníkového výběru dle vzorce:

$$selectedMapWidth = \frac{selectedMapHeight}{(panelHeight - 1)} * (panelWidth - 1) ,$$

kde *selectedMapWidth* je šířka a *selectedMapHeight* výška obdélníkového výběru mapy, *panelWidth* je šířka a *panelHeight* výška panelu, kde se mapa zobrazuje. Původní rozměr obdélníkového výřezu je výchozí pro další přizpůsobování poměru.



Obrázek 5.2: Ukázka přerozměrování obdélníkového výběru do správného poměru

Po kontrole a případném upravení poměru obdélníkového výřezu mapy můžeme databázi dotázat, aby nám vybrala prostorová data, která se nachází uvnitř vybraného obdélníku. Budeme chtít i data, která jsou z části uvnitř tohoto obdélníku a z části vně. Dotazat SQL pro příkaz PREPARE dynamického SQL bude vypadat následovně:

```
SELECT geom, elementcol
FROM linieKN
WHERE SDO_RELATE(geom,
  SDO_GEOMETRY(
    2003, null, null,
    SDO_ELEM_INFO_ARRAY(1, 1003, 3),
    SDO_ORDINATE_ARRAY(?, ?, ?, ?)
  ),
  'mask=anyinteract') = 'TRUE'
```

```
AND SDO_GEOM.SDO_LENGTH(geom, 1) >= ?
ORDER BY elementcol;
```

Dotaz nám vybírá geometrii a informace ze sloupce *elementcol* (reprezentujícího vrstvu), které mají nějaký vztah s obdélníkovým výběrem, jehož souřadnice se doplní za otazníky u *SDO_ORDINATE_ARRAY* v pořadí *x1*, *y1*, *x2*, *y2*, kde *x1*, *y1* je libovolný rohový bod obdélníkového výběru a *x2*, *y2* je protilehlý rohový bod obdélníkového výběru. Konstruktorem *SDO_GEOMETRY* v dotazu vytváříme danou geometrii, v tomto případě obdélník. Protože je zbytečné vybírat všechna data, i ta, která by se neměla zobrazit, tzn. že jsou menší než je měřítko zobrazení, byla do dotazu zapracována jistá optimalizace, kdy vybíráme pouze ty úsečky, které jsou stejně dlouhé nebo delší než je aktuální měřítko zobrazení. Aktuální měřítko se doplní za otazník u *SDO_LENGTH*. To se nám hodí např. při zobrazení celé mapy, kde se při velikosti okna 1070 x 754 pixelů (tj. panel pro zobrazování mapy o rozměrech 804 x 632 pixelů) vybere 16671 úseček z celkového počtu 37758 úseček uložených v databázi, což je méně než polovina.

Pokud měřítko zobrazeného úseku mapy klesne pod hodnotu půl metru na pixel, začnou se uprostřed pozemků, kde máme spojené vzorové polygony, zobrazovat identifikace parcel. Požadavek na identifikaci parcel posíláme parametrickým dotazem:

```
SELECT SDO_GEOM.SDO_CENTROID(p.geom, m.diminfo), p.name
FROM polygonyKN p, user_sdo_geom_metadata m
WHERE SDO_RELATE(geom,
    SDO_GEOMETRY(
        2003, null, null,
        SDO_ELEM_INFO_ARRAY(1, 1003, 3),
        SDO_ORDINATE_ARRAY(?, ?, ?, ?)
    ),
    'mask=anyinteract') = 'TRUE'
AND m.table_name='POLYGONYKN'
AND m.column_name='GEOM';
```

Za otazníky v *SDO_ORDINATE_ARRAY* se doplní příslušné okno dotazu v pořadí *x1*, *y1*, *x2*, *y2*, kde *x1*, *y1* je libovolný rohový bod obdélníkového výběru a *x2*, *y2* je protilehlý rohový bod obdélníkového výběru. Konstruktorem *SDO_GEOMETRY* v dotazu vytváříme danou geometrii, v tomto případě obdélník. Tento dotaz vybere všechny polygony, které se nachází v okně dotazu a vrátí nám pro každý pozemek jeho identifikaci a středový bod. Středový bod využijeme při zobrazování identifikace pozemku v mapě, tak jak je znázorněno na obr. 5.4.

5.6 Zobrazení celé mapy

Po spuštění aplikace bychom potřebovali uživateli zobrazit prostorová data tak, aby viděl celou mapu uloženou v databázi v tabulce *linieKN* tak, jak je to znázorněno na obr. 5.3. Potřebujeme tedy nejmenší a největší bod, který se v dané prostorové tabulce nachází, a to pro každou osu.

Pokud máme správně uložená data v tabulce metadat, měli bychom mít nejmenší a největší body (případně s určitou rezervou) uložené právě v tabulce metadat *user_sdo_geom_metadata*. Aplikace po svém spuštění pošle na databázový server Oracle dotaz na tato metadata. V dotazu zjišťuje pouze tyto dva body pro každou souřadnou osu. Výsledky si uloží do třídních proměnných, takže při příštím zobrazení celé mapy dotaz znovu neposílá a vybere si data pouze z třídních proměnných. Takže například pro tabulku *linieKN* bude dotaz vypadat následovně:

```
SELECT SDO_DIMNAME, SDO_LB, SDO_UB
FROM TABLE (
    SELECT DIMINFO
    FROM user_sdo_geom_metadata
    WHERE TABLE_NAME = 'LINIEKN' AND COLUMN_NAME = 'GEOM'
)
WHERE SDO_DIMNAME = 'X' OR SDO_DIMNAME = 'Y';
```

Dotaz vybere z tabulky metadat záznam, kde jméno tabulky odpovídá tabulce *linieKN* a jejíž prostorový sloupec má název *GEOM*. Atributy *DIMINFO* a *SDO_DIMNAME* byly popsány v kapitole 5.1. Sloupec *DIMINFO*, který jsme z tabulky metadat vybrali, nám tvoří další tabulku, z níž vybereme sloupce *SDO_LB* (*Lower bound*, tzn. nejmenší hranice) a *SDO_UB* (*Upper bound*, tzn. největší hranice) pro osy *X* a *Y*. Nakonec předá tyto body metodě pro výběr z okna dotazu. Celá metoda je spouštěna do nového vlákna.

Pokud bychom neměli pro prostorová data tabulku s metadaty, měli bychom jistou alternativu. Tyto body bychom mohli zjistit dotazem:

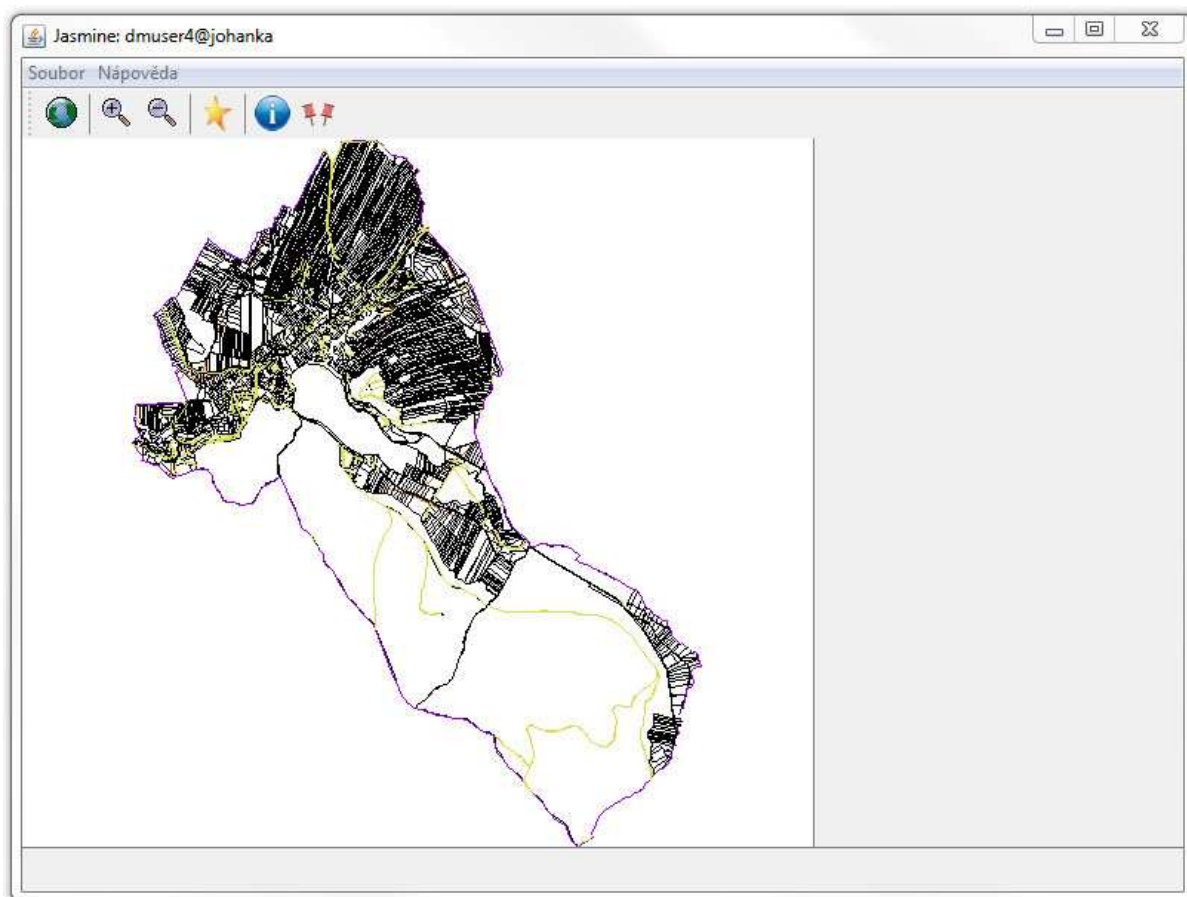
```
SELECT
    MIN(SDO_GEOM.SDO_MIN_MBR_ORDINATE(GEOM, 1)) AS X_MIN,
    MAX(SDO_GEOM.SDO_MAX_MBR_ORDINATE(GEOM, 1)) AS X_MAX,
    MIN(SDO_GEOM.SDO_MIN_MBR_ORDINATE(GEOM, 2)) AS Y_MIN,
    MAX(SDO_GEOM.SDO_MAX_MBR_ORDINATE(GEOM, 2)) AS Y_MAX
FROM linieKN;
```

Zpracování tohoto dotazu je ale časově i výpočetně velmi náročné, protože Oracle ke každé geometrii hledá minimální ohraničující obdélník a z něj vybírá nejmenší a největší bod pro každou souřadnou osu. Agregáčnící funkcí *MIN()* a *MAX()* pak vybíráme nejmenší a největší bod pro každou souřadnou

osu. Odpověď na tento dotaz dostaneme po cca 45 sekundách, což je pro aplikaci nevhodná a nepřipustně dlouhá doba.

Další možností, která však trvá také hodně dlouho (cca 34 sekund), by bylo využít agregační funkci, která vybere nejmenší ohraničující obdélník společný pro všechny geometrie v tabulce:

```
SELECT SDO_AGGR_MBR(GEOM) FROM linieKN;
```



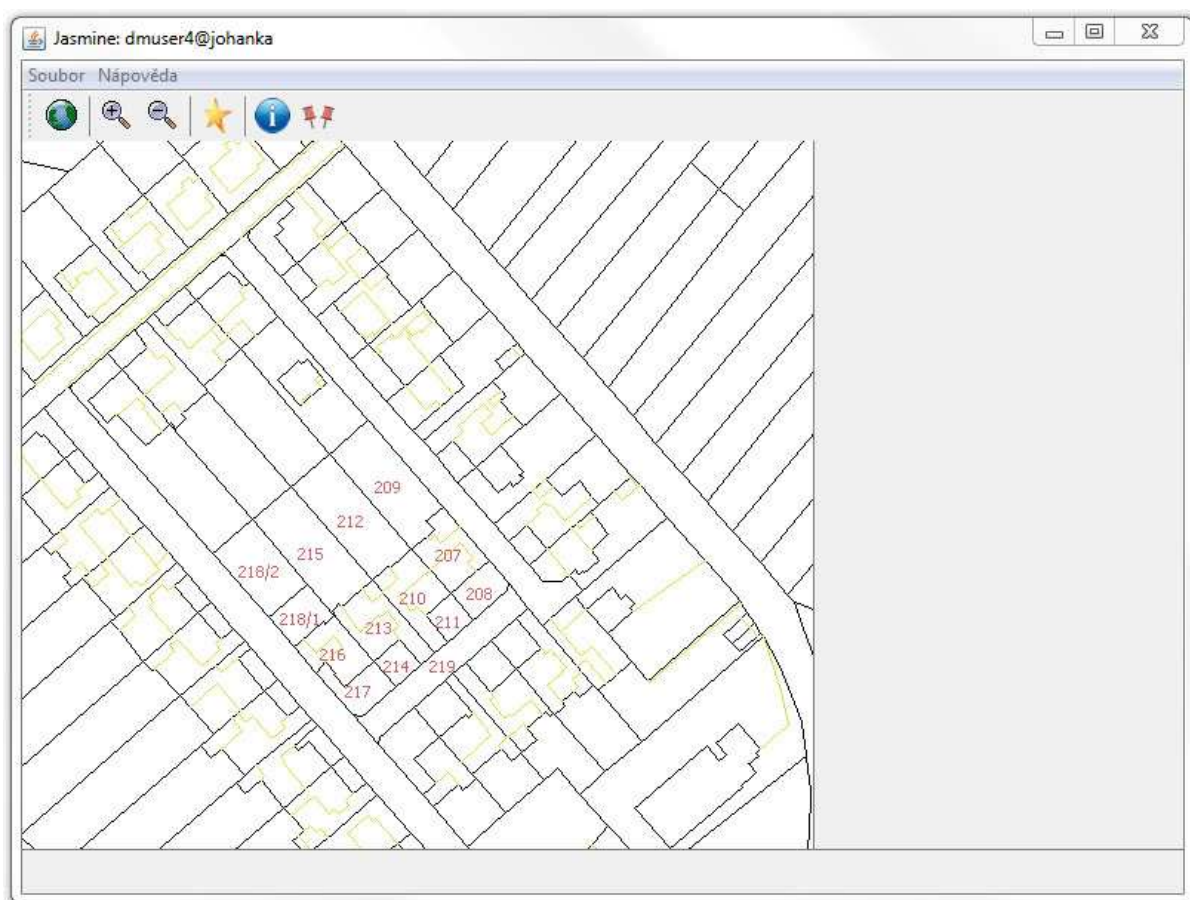
Obrázek 5.3: Ukázková aplikace Jasmine při zobrazení celé mapy katastrálního území

5.7 Přibližování a oddalování mapy

Při přibližování a oddalování je pro aplikaci výchozím bodem pozice na mapě, na kterou uživatel kliknul v rámci přibližování, případně oddalování. Tato pozice bude ve středu zobrazení mapy po přiblížení nebo oddálení. Aplikace zjistí rozsah, v jakém bylo zobrazováno, vypočítá rozsah nový tak, že vezme rohové body zobrazovaného obdélníka, z nich vypočítá polovinu výšky a šířky, které pak podělí konstantou pro zvětšení (případně zmenšení). Konstantu pro zvětšování a zmenšování určuje výběr v menu *Zoom*, implicitně je nastavena na 1,5. Pomocí středu mapy a nového rozsahu vypočítá okraje mapy a ty předá metodě pro výběr z okna dotazu. Celá metoda je spouštěna do nového vlákna. Zobrazení identifikace parcel bylo zmíněno v kapitole 5.5.

5.8 Zobrazení mapy s prostorovými informacemi

Stejně jako při zobrazení celé mapy i zde potřebujeme zjistit pro každou souřadnou osu nejmenší a největší bod, který se nachází v dané prostorové tabulce, v tomto případě v tabulce *polygonyKN*. V které části mapy se nacházejí spojené polygony, aplikace zjistí opět z tabulky metadat se záznamy pro tuto tabulku, tzn. jméno tabulky *polygonyKN* a sloupec s prostorovými daty pojmenovaný *GEOM*. Tyto informace jsou zjišťovány taktéž při startu aplikace a následně uloženy do třídních proměnných, takže při příštím zobrazení mapy s prostorovými informacemi se aplikace znovu nedotazuje na rohové body, ale vezme si je z třídních proměnných. Ty předá metodě pro výběr z okna dotazu. Celá metoda je spouštěna do nového vlákna. Mapa, kde se nacházejí spojené polygony je znázorněna na obr. 5.4.



Obrázek 5.4: Ukázková aplikace Jasmine se zobrazenou oblastí obsahující prostorové informace

5.9 Zjišťování prostorových informací

Při zapnuté funkci zjišťování informací po kliknutí na oblast, kde se nachází spojený polygon (oblast je označena identifikací uprostřed pozemku), dostaneme příslušné informace o dané parcele, jako

např. identifikaci parcely, rozlohu parcely, obvod parcely, střed parcely a sousední parcely, jak je znázorněno na obr. 5.5. Celá metoda je spouštěna do nového vlákna.

Identifikaci, rozlohu, obvod a střed parcely zjistíme následujícím parametrickým dotazem:

```
SELECT p.geom, p.name,
       SDO_GEOM.SDO_LENGTH(p.geom, m.diminfo) AS LENGTH,
       SDO_GEOM.SDO_AREA(p.geom, m.diminfo) AS AREA,
       SDO_GEOM.SDO_CENTROID(p.geom, m.diminfo) AS CENTROID
FROM polygonyKN p, user_sdo_geom_metadata m
WHERE SDO_ANYINTERACT(p.geom,
                       SDO_GEOMETRY(
                           2003, null, null,
                           SDO_ELEM_INFO_ARRAY(1, 1003, 3),
                           SDO_ORDINATE_ARRAY(?, ?, ?, ?)
                       )
) = 'TRUE'
AND m.table_name='POLYGONYKN'
AND m.column_name='GEOM'
ORDER BY p.name;
```

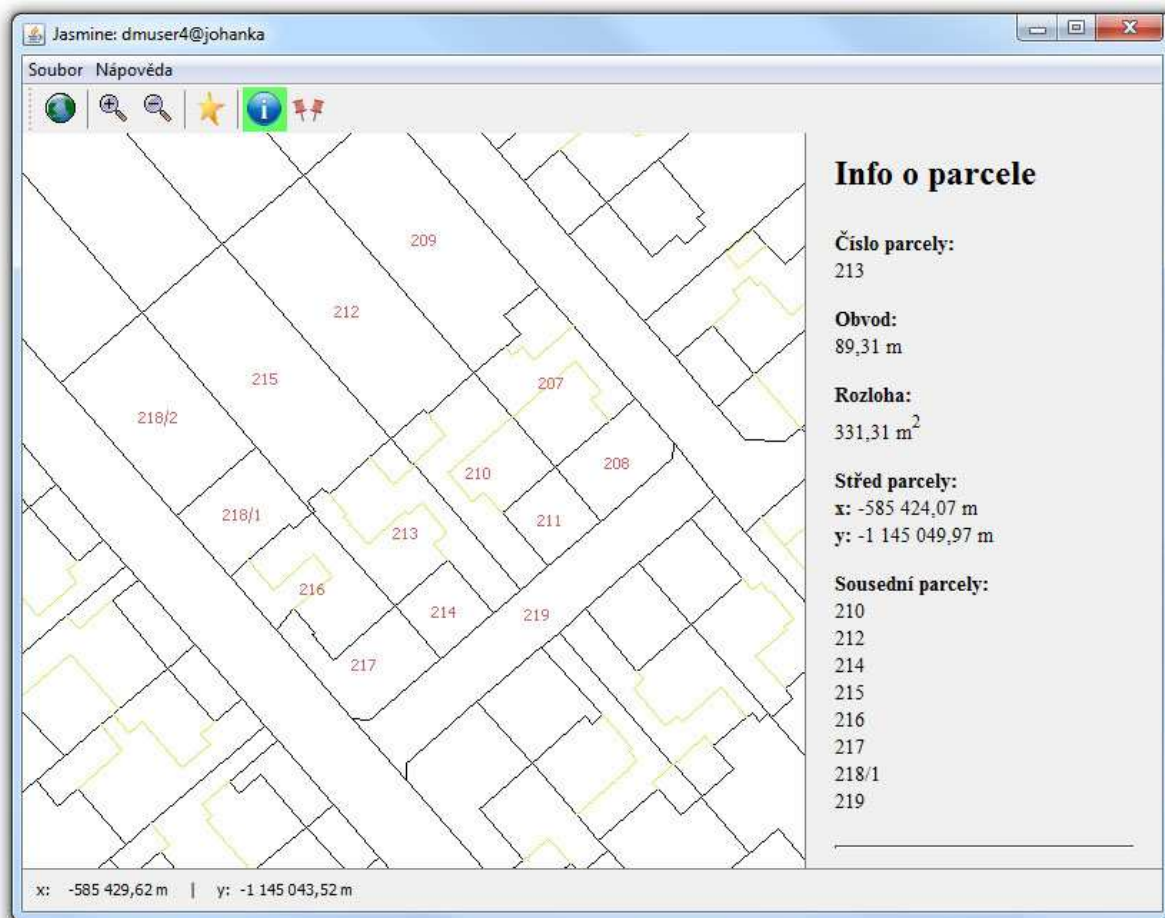
Z databáze vybereme polygon (pro následné zjištění sousedních parcel) a jméno, prostorovou operaci *SDO_LENGTH* zjistíme obvod, pomocí *SDO_AREA* zjistíme rozlohu a pomocí *SDO_CENTROID* zjistíme střed parcely. Výběr učiníme na základě vztahu se čtvercem, jehož středem je bod (pozice) na mapě, kde bylo kliknuto, a hrana čtverce je délky jeden pixel aktuálního měřítka zobrazované mapy. Za otazníky u *SDO_ORDINATE_ARRAY* se doplní *x1*, *y1*, *x2*, *y2*, kde *x1*, *y1* je libovolný rohový bod tohoto čtverce a *x2*, *y2* je protilehlý rohový bod čtverce. Konstruktorem *SDO_GEOMETRY* v dotazu vytváříme danou geometrii, v tomto případě čtverec.

Uživatel může kliknout na společnou hranici zobrazenou na mapě. Pokud bychom dělali výběr na základě bodu na mapě, kam uživatel kliknul, uživatel by musel kliknout přesně do místa hranice, aby měl šanci, že se mu zobrazí informace o parcelách, kterým tato hranice patří. Při větším měřítku (více metrů na pixel) by byla šance na úspěch velmi malá. Proto se informace vybírají na základě čtverce, kde čtverec reprezentuje odchylku všemi směry.

Sousední parcely vybíráme parametrickým dotazem:

```
SELECT name
FROM polygonyKN
WHERE SDO_TOUCH(geom, ?) = 'TRUE'
ORDER BY name
```

Dotaz vybírá jméno parcely, která sdílí hranici s parcelou, jejíž prostorovou reprezentaci v podobě polygonu jsme vybrali v dotazu předchozím a tuto prostorovou reprezentaci dosadíme za otazník v *SDO_TOUCH*. Vybraná jména parcel seřadí dle jména.



Obrázek 5.5: Ukázková aplikace Jasmine se získanými informacemi o parcele

5.10 Zjišťování vzdálenosti

Při zapnuté funkci zjišťování vzdálenosti parcel po kliknutí na oblast, kde se nachází spojený polygon (oblast je označena identifikací uprostřed pozemku), dostaneme po prvním kliknutí příslušné informace o poloze výchozího bodu, po druhém kliknutí dostaneme pozici cílového bodu a informace o vzdálenosti mezi parcelami, jak je znázorněno na obr. 5.6. Celá metoda je spouštěna do nového vlákna. Parametrický SQL dotaz:

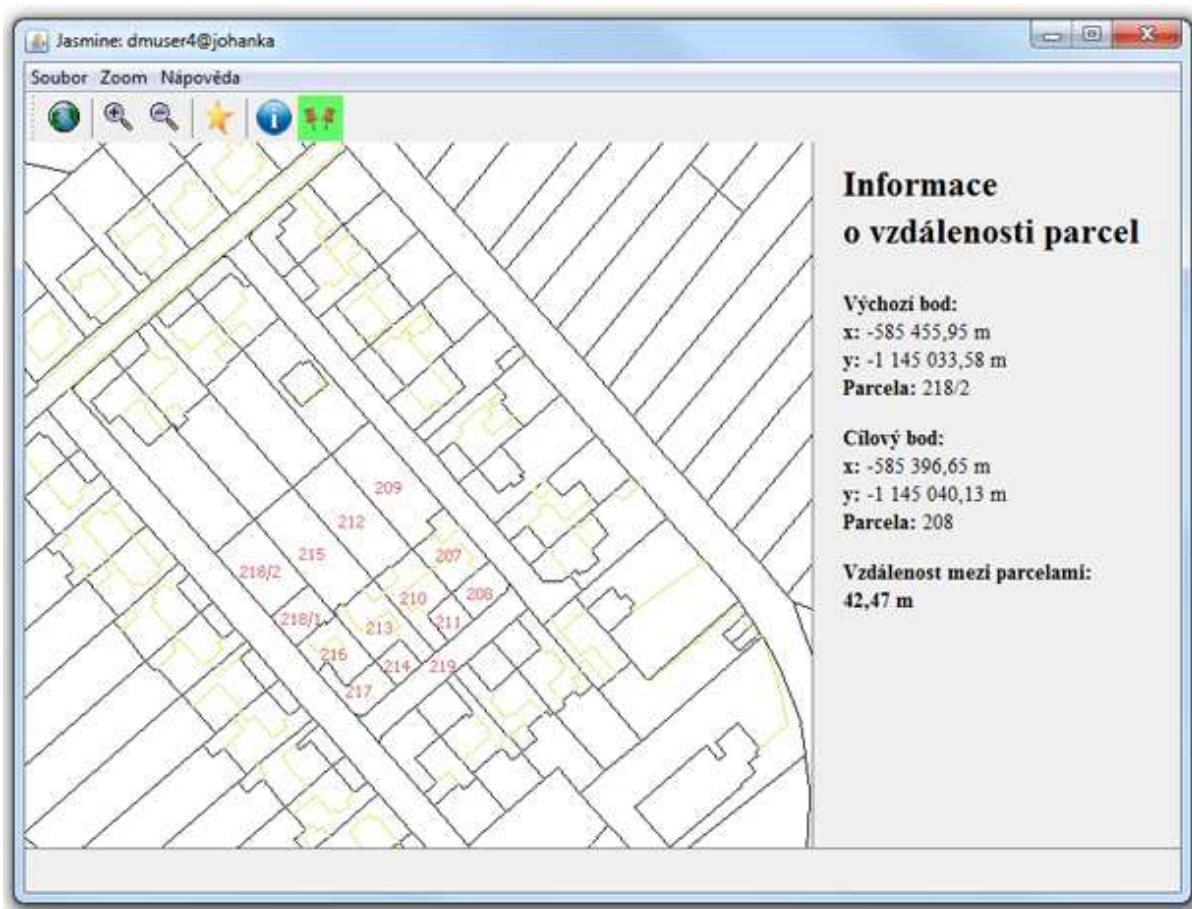
```
SELECT p1.ID, p2.ID, p1.name, p2.name,
       SDO_GEOM.SDO_DISTANCE(p1.geom, p2.geom, 0.0005)
FROM polygonsKN p1, polygonsKN p2
WHERE SDO_ANYINTERACT(p1.geom,
```

```

SDO_GEOMETRY(2001, null,
  SDO_POINT_TYPE(?, ?, null),
  null, null)) = 'TRUE'
AND SDO_ANYINTERACT(p2.geom,
  SDO_GEOMETRY(2001, null,
  SDO_POINT_TYPE(?, ?, null),
  null, null)) = 'TRUE'

```

vybere ID obou parcel, jména obou parcela a vzdálenost mezi těmito parcelami. První parcele náleží výchozí bod (doplní se za otazníky u prvního *SDO_GEOMETRY*), druhé parcele náleží bod cílový (doplní se za otazníky u druhého *SDO_GEOMETRY*). ID obou parcel využije ke zjištění, zda se jedná o stejnou parcelu.



Obrázek 5.6: Zobrazení informací o vzdálenosti parcel

6 Závěr

6.1 Zhodnocení výsledků

Tato bakalářská práce se zabývá podporou pro práci s prostorovými daty v prostředí databázového serveru Oracle. Ve studijní části rovněž zmiňuje možnosti využití podpory pro časoprostorová data v Oracle.

Na základě studijní části byla vytvořena ukázková aplikace nazvaná Jasmine, která demonstruje využití podpory pro práci s prostorovými daty. K aplikaci se podařilo sehnat zdrojová data ve formátu Shapefile, která byla konvertorem Oracle Spatial Converter konvertována do formátu SQL pro načtení do databáze za pomoci *SQL*Loaderu*. Získaná data obsahovala úsečky reprezentující objekty na mapě a body reprezentující identifikaci parcel. Aplikace zobrazuje mapu, uživateli umožní její přibližování a oddalování a uživatel může zobrazovat informace o vybrané parcele jako je její identifikace, rozloh, obvod, střed a parcely s ní sousedící.

Tato bakalářská práce pro mě byla velkým přínosem, dala mi možnost seznámit se se základy tvorby geografických informačních systémů. Hlavní obsah této práce, podpora pro prostorová data v Oracle, mi rozšířil znalosti v oblasti databázových systémů o post-relační databázové systémy. Dále jsem se seznámil s programováním v Javě, s kterou jsem do té doby nikdy nepracoval, protože jsem měl v bakalářském studiu zapsán předmět C++.

6.2 Možnosti pokračování projektu

Už během práce na tomto projektu se objevilo několik možností, jak by se dalo pokračovat v tomto projektu.

Ve vstupních datech, která se podařilo sehnat pro ukázkovou aplikaci k této bakalářské práci, se nenacházely polygony, jak bylo původně předpokládáno, ale pouze čáry reprezentující hranice pozemků (a dalších objektů) a body reprezentující identifikaci parcely.

První možností, jak tento projekt rozšířit, by mohlo být nalezení algoritmu, který by k dané úsečce tvořící hranici pozemku našel další úsečky, které by společně vytvořily vždy nejmenší možný polygon a k tomuto polygonu našel identifikaci, která se nachází topologicky někde uvnitř polygonu.

Další možností by mohlo být implementování interakce do mapy. Pokud by se nepovedlo implementovat dříve uvedený algoritmus, mohli bychom si spojit úsečky do polygonů ručně. Interakce uživatele by byla užitečná i v případě, že by se dříve uvedený algoritmus implementovat podařilo, protože v mapě jsou i další úsečky v jiných vrstvách reprezentující další objekty (např.

zastavěné plochy), které bychom potřebovali spojit. Tyto objekty by pravděpodobně nebylo možné spojit automaticky, protože některé sdílí společnou hranici s pozemky a daná hranice je v databázi pouze jednou, a to pouze u pozemku. Uživatel by si zobrazil mapu, kliknutím na příslušné úsečky v mapě by si vybral hranice příslušného pozemku (nebo jiného objektu), které chce spojit, a k nim příslušnou identifikaci pozemku (nebo objektu). Následně by tuto akci potvrdil, aplikace by zkontrolovala, zda se uživatel nesnaží potvrdit neuzavřenou oblast, tedy zda na sebe hrany navazují a tvoří uzavřený polygon. Dále by zkontrolovala, zda se polygon nebo jeho část již v databázi v dané vrstvě nenachází. V případě, že bychom neumožnili zadání vlastní identifikace, ale omezili bychom se pouze na výběr identifikace z mapy, by aplikace ještě zkontrolovala, zda se bod nachází uvnitř tohoto polygonu. Po těchto kontrolách by aplikace v případě úspěchu polygon uložila do databáze, v případě neúspěchu by uživateli nabídla vhodným způsobem opravu.

Nejdéle na celé aplikaci zdržuje komunikace s databází. Např. při načítání celé mapy je stále poměrně dlouhá prodleva, než získáme mapová data, a to i přes snahu optimalizace. Pro zobrazení dat musíme počkat, až obdržíme všechna data, a teprve potom jsou vykreslena. Další zajímavou možností, jak projekt obohatit, by mohlo být rozdělení SQL dotazu tak, aby nám vracel pouze bloky dat, v kterých by bylo například 100 řádků z tabulky. Takto upravené SQL dotazy by se poslaly buď ve vláknech paralelně a nebo postupně po částech v cyklu. Smyslem tohoto zlepšení je, že by uživatel neměl prázdnou obrazovku s mapou do doby než aplikace obdrží všechna data, ale data by se vykreslovala postupně po částech.

Literatura

- [1] Kolář, D.: *Pokročilé databázové systémy (prostorové databáze)*. Studijní opora, Brno, FIT VUT v Brně, 2006
- [2] Eisenberg, A., Melton, J.: *SQL Multimedia and Application Packages (SQL/MM)*.
Dokument dostupný na URL:
<http://www.sigmod.org/sigmod/record/issues/0112/standards.pdf> (říjen 2001)
- [3] Oracle Corporation: *Oracle Spatial 11g: Advanced Spatial Data Management for the Enterprise*.
Dokument dostupný na URL:
http://www.oracle.com/technology/products/spatial/pdf/11g_collateral/spatial11g_datasheet.pdf
- [4] Vališ, J.: *Databáze pohybujících se objektů*. Diplomová práce, Brno, FIT VUT v Brně, 2008
- [5] Oracle Corporation: *Oracle Spatial Developer's Guide, 11g Release 1 (11.1)*.
Dokument dostupný na URL:
http://download-uk.oracle.com/docs/cd/B28359_01/appdev.111/b28400.pdf (červenec 2007)
- [6] Krch, D.: *Technologie Oracle pro zpracování dat*.
Dokument dostupný na URL:
www.oracle.com/global/cz/events/prezentace/050329-prostorovadata.pdf (březen 2005)
- [7] Oracle Corporation: *Oracle Database Concepts, 11g Release 1 (11.1)*.
Dokument dostupný na URL:
http://download.oracle.com/docs/cd/B28359_01/server.111/b28318.pdf (říjen 2008)
- [8] Oracle Corporation: *Oracle Database SQL Language Reference, 11g Release 1 (11.1)*.
Dokument dostupný na URL:
http://download.oracle.com/docs/cd/B28359_01/server.111/b28286.pdf (září 2008)
- [9] T-Mapy, spol. s r.o.: *ArcČR 500*.
Dokument dostupný na URL:
http://www.tmapy.cz/public/tmapy/cz/_geodata/prodej2/_arccr_500.html

- [10] ESRI: *ESRI Shapefile Technical Description, ESRI White Paper*.

Dokument dostupný na URL:

<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> (červenec 1998)

- [11] Oracle Corporation: *Shapefile To Oracle Spatial Converter*.

Dokument dostupný na URL:

http://www.oracle.com/technology/software/products/spatial/files/text_files/using_shp2sdo.txt

(leden 2004)

Seznam příloh

- Příloha 1: Uživatelská příručka
Příloha 2: CD
Příloha 3: Seznam použitých zkratek

Příloha 1

Uživatelská příručka

Spuštění aplikace

Aplikace Jasmine na přiloženém CD-ROM je připravena pro použití se školním databázovým serverem Oracle (server *berta.fit.vutbr.cz*). Pro aplikaci je možné využít databázový účet, který byl na tomto serveru pro tyto účely vytvořen (uživatelské jméno: „*dmuser7*“, uživatelské heslo: „*jasmine*“). Pod tímto účtem jsou připravena (do doby jeho smazání na konci akademického roku) kompletní data pro okamžité využití aplikace. Spustitelná aplikace ve formátu *.jar se nachází v adresáři /jasmine/build/. Po spuštění se zobrazí celá mapa katastrálního území.

Pro použití s jiným databázovým účtem je třeba provést SQL skripty uložené v adresáři /SQL/ pomocí *SQL*Plus* a *SQL*Loaderu*. Nejprve je třeba vytvořit tabulky pomocí *SQL*Plus*. V příkazovém řádku zadáme:

```
sqlplus "jmeno/heslo@server:port:sluzba" ,
```

kde *jmeno* je uživatelské jméno a *heslo* uživatelské heslo k databázi Oracle, *server* je název serveru s databází Oracle, *port* je číslo portu, na kterém Oracle naslouchá komunikaci, a *sluzba* je název služby. Příkaz:

```
@cesta_k_souboru/nazev_souboru.sql
```

provede SQL skript, který vytvoří tabulku s prostorovými daty a zároveň naplní tabulku s uživatelskými metadaty. Provedeme SQL skripty s názvem *bodyKN*, *linieKN* a *polygonyKN*. Poté musíme naplnit tabulky *bodyKN* a *linieKN* daty pomocí *SQL*Loaderu*. V příkazové řádce se přesuneme do adresáře s SQL skripty a spustíme *SQL*Loader* příkazem:

```
sqlldr "jmeno/heslo@server:port:sluzba" nazev_souboru.ctl ,
```

kde přihlašovací řetězec byl zmíněn u *SQL*Plus* a *nazev_souboru.ctl* je řídicí soubor, pomocí kterého se má databáze naplnit. Jako řídicí soubory použijeme *bodyKN.ctl* a *linieKN.ctl*. Jako poslední provedeme skript *spatial_indexes.sql* pomocí *SQL*Plus*. Dále je třeba nastavit příslušné přístupové údaje k databázovému serveru Oracle v souboru *OraConfig.java*.

Popis ikon v panelu nástrojů

Ikony použité v aplikaci mají následující význam:



celá mapa – po kliknutí na tuto ikonu se zobrazí celá mapa katastrálního území



zvětšování – po kliknutí na tuto ikonu aplikace umožní mapu zvětšovat



zmenšování – po kliknutí na tuto ikonu aplikace umožní mapu zmenšovat



ukázková oblast – po kliknutí na tuto ikonu se zobrazí oblast, kde jsou spojeny vzorové polygony



informace – po kliknutí na tuto ikonu aplikace umožní zobrazovat informace o parcelách



vzdálenost – po kliknutí na tuto ikonu aplikace umožní zobrazovat informace o vzdálenostech parcel

Zvětšování, zmenšování mapy

Pro využití zvětšování nebo zmenšování mapy je třeba aktivovat daný nástroj kliknutím na příslušnou ikonu (*zvětšování*, *zmenšování*) v panelu nástrojů. Je-li aktivní nástroj zvětšování nebo zmenšování, je tato skutečnost indikována podbarvením příslušné ikony zelenou barvou.

Je-li nástroj aktivní, pak kliknutím do prostoru, kde se mapa zobrazuje, mapu zvětšíme nebo zmenšíme (dle vybraného nástroje). Konstanta zvětšování nebo zmenšování se vybírá v menu *Zoom*. Implicitně je konstanta nastavena na hodnotu 1,5. Poloha, na kterou v mapě klikneme se stane středem pro nové zobrazení mapy.

Zjišťování informací o parcelách

Pro využití zjišťování informací o parcelách je třeba aktivovat daný nástroj kliknutím na příslušnou ikonu (*informace*) v panelu nástrojů. Je-li aktivní nástroj pro zjišťování informací o parcelách, je tato skutečnost indikována podbarvením příslušné ikony zelenou barvou.

Je-li nástroj aktivní, pak se po kliknutí do prostoru, kde se nachází spojené vzorové polygony (uprostřed parcely je červeně vypsána jejich identifikace, např. po kliknutí na ikonu *ukázková oblast*), zobrazí v pravém panelu informace o této parcele. Tyto informace zahrnují identifikaci parcely, obvod parcely, rozlohu parcely, střed parcely a sousední parcely k této parcele. Pokud

uživatel klikne na hranici, která patří více parcelám, pak jsou zobrazeny informace ke všem těmto parcelám.

Zjišťování vzdáleností parcel

Pro využití zjišťování vzdáleností parcel je třeba aktivovat daný nástroj kliknutím na příslušnou ikonu (*vzdálenost*) v panelu nástrojů. Je-li aktivní nástroj pro zjišťování vzdáleností parcel, je tato skutečnost indikována podbarvením příslušné ikony zelenou barvou.

Je-li nástroj aktivní, pak se po kliknutí do prostoru, kde se nachází spojené vzorové polygony (uprostřed parcely je červeně vypsána jejich identifikace, např. po kliknutí na ikonu *ukázková oblast*), zobrazí v pravém panelu informace o poloze, která je výchozím bodem. Po druhém kliknutí do této oblasti se zobrazí informace o poloze, která je bodem cílovým. Pokud bylo kliknuto na oblast, kde jsou spojeny vzorové polygony, pak se zjistí vzdálenost parcel obsahujících tyto body. Uživatel je také informován o tom, pokud se jedná o stejnou parcelu nebo parcely sousední. Další vyhledávání může uživatel začít novým kliknutím na panel s mapou.

Příloha 2

Obsah CD

K bakalářské práci je přiložen optický disk CD-ROM s adresářovou strukturou:

- /data/ adresář obsahuje zdrojová mapová data
 - ./originál/ adresář obsahuje originální mapová data z FAST VUT
 - ./upravená/ adresář obsahuje upravená mapová data
- /jasmine/ adresář obsahuje ukázkovou aplikaci Jasmine
 - ./build/ adresář obsahuje spustitelnou ukázkovou aplikaci Jasmine ve formátu *.jar
 - ./lib/ adresář obsahuje knihovny použité v ukázkové aplikaci Jasmine
 - ./src/ adresář obsahuje zdrojové kódy ukázkové aplikace Jasmine
- /jasminedoc/ adresář obsahuje vygenerovanou dokumentaci k ukázkové aplikaci Jasmine
- /shp2sdo/ adresář obsahuje konvertor Oracle shp2sdo
- /sql/ adresář obsahuje skripty pro vytvoření tabulek a naplnění databáze
- /text/ adresář obsahuje text bakalářské práce ve formátu *.odt a *.pdf

Příloha 3

Seznam použitých zkratk

- API Application Programming Interface
- ESRI Environmental System Research Institute, Inc.
- FAST Fakulta stavební na VUT
- GUI Graphical User Interface
- JDBC Java DataBase Connectivity
- Jasmine Java Application: Spatial Map Information Engine
- SQL Structured Query Language
- URI Uniform Resource Identifier
- VUT Vysoké učení technické