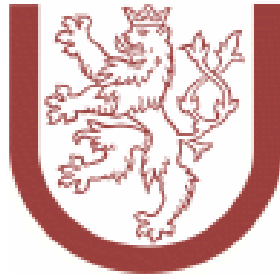


ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd



**Modelování geoprostorové báze
dat na úrovni datového modelu KN**
(Diplomová práce)

Modelling of geospatial features at the data model
of cadastre level
(Diploma work)

**zpracoval: Karel Janečka
ZČU/FAV/GEM/GIS
vedoucí práce: Doc. Ing. Václav Čada, CSc.
dne: 10.03.2005**

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně a s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Bechyni 10.03.2005

.....

Poděkování

Na tomto místě bych chtěl poděkovat všem, kteří přispěli ke vzniku této diplomové práce. Zvláště Doc. Ing. Václavu Čadovi, Csc. za metodické pokyny a věcné připomínky během celého průběhu tvorby této práce a mé rodině za podporu, trpělivost a toleranci.

Anotace

Cílem práce bylo otestovat funkcionalitu databáze Oracle a jejího rozšíření Spatial pro účely generalizace geoprostorových prvků na úrovni datového modelu informačního systému katastru nemovitostí. K využití nabízených funkcí je potřeba zdrojová data vhodným způsobem převést a uložit do navrženého datového modelu. V práci jsem se snažil zachytit a popsat specifika přechodu od uložení zdrojových dat v souboru výměnného formátu ISKN do objektově-relačního uložení v databázi Oracle s využitím objektového typu podporovaného rozšířením Spatial. Nabízené funkce, které se nechají v prostředí Spatial ke generalizaci použít, jsem otestoval nad vybranou množinou dat při generování přehledové mapy katastrálních území.

Klíčová slova:

výměnný formát, katastrální hranice, katastrální území, přehledová mapa, Oracle Spatial, generalizace, geometrický popis prvku

Anotation

The aim of the diploma work was the testing of functions of the Oracle database and Spatial extension for the generalization of geospatial features at the information system of cadastre data model level. It's necessary convert and save source data into the devised data model. There I tried to describe specifications of the conversion from the source data saved in the file of exchange format into the object-relational storage in the Oracle database through the Spatial aided object data type in my diploma work. I tested Spatial generalization functions on selected data set in the course of the generating of the plan position map of cadastral territories.

Keywords:

exchange format, cadastral boundary, cadastral territory, plan position map, Oracle Spatial, generalization, geometry

Typografické konvence

Mnou zavedené typografické konvence pro odlišení různých kategorií textu:

geometrie

feature table

KATASTRALNI_HRANICE

hranice

NUMBER

sdo_gtype

NULL

zvýraznění důležitého pojmu
(spojení)

anglický výraz

název tabulky či pohledu

název proměnné, sloupce v tabulce

či parametru procedury nebo funkce

název datového typu

název atributu (v rámci datového

typu *SDO_GEOMETRY*)

hodnota atributu

Seznam příkladů

Příklad 1	Definice objektového datového typu <i>SDO_GEOMETRY</i>	20
Příklad 2	Definice datových typů <i>SDO_POINT_TYPE</i> , <i>SDO_ELEM_INFO_ARRAY</i> a <i>SDO_ORDINATE_ARRAY</i>	20
Příklad 3	Atributy metadatového pohledu	24
Příklad 4	Definice typů <i>SDO_DIM_ARRAY</i> a <i>SDO_DIM_ELEMENT</i>	25
Příklad 5	Formát funkce <i>SDO_UTIL.REMOVE_DUPLICATE_VERTICES</i>	26
Příklad 6	Formát funkce <i>SDO_UTIL.SIMPLIFY</i>	27
Příklad 7	Formát funkce <i>SDO_SAM.SIMPLIFY_GEOMETRY</i>	29
Příklad 8	Formát procedury <i>SDO_SAM.SIMPLIFY_LAYER</i>	29
Příklad 9	Struktura příkazu INSERT	36
Příklad 10	SQL zápis dotazu na vybrání úseků a bodů katastrální hranice	40
Příklad 11	Definice tabulky <i>KATASTRALNI_HRANICE</i>	41
Příklad 12	Metadata o sloupci hranice	42
Příklad 13	Metadata o sloupci <i>definicni_bod</i>	43
Příklad 14	Definice typu <i>usek</i> a ukazatele na tento typ <i>pusek</i>	46
Příklad 15	Vložení jednoho prvku do tabulky <i>KATASTRALNI_HRANICE</i>	49
Příklad 16	Vložení geometrického popisu prvku o více než 999 vrcholových bodech do tabulky <i>KATASTRALNI_HRANICE</i>	51
Příklad 17	Připojení datového zdroje k aplikaci MapViewer	54
Příklad 18	Zadání dotazu pro vykreslení „přehledové“ mapy katastrálních území	54
Příklad 19	Definice tabulky <i>KATASTRALNI_HRANICE_MAT</i>	57
Příklad 20	Metadata o sloupci hranice v tabulce <i>KATASTRALNI_HRANICE_MAT</i>	60
Příklad 21	Metadata o sloupci <i>definicni_bod</i> v tabulce <i>KATASTRALNI_HRANICE_MAT</i>	61
Příklad 22	Syntaxe funkce <i>SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT</i>	61
Příklad 23	Testování validity dat ve sloupci <i>hranice</i>	62
Příklad 24	Syntaxe operátoru <i>SDO_TOUCH</i>	62
Příklad 25	Vytvoření prostorového indexu nad sloupcem <i>Hranice</i>	63
Příklad 26	Testování topologických vztahů pomocí operátoru <i>TOUCH</i>	63
Příklad 27	Generalizace geometrických popisů uložených ve sloupci <i>hranice</i>	64

Seznam obrázků

Obrázek 1	Dotazovací model	18
Obrázek 2	Ukázka MBR	19
Obrázek 3	Hierarchický R-tree index	19
Obrázek 4	Chování funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES	27
Obrázek 5	Princip Douglas-Peuckerova algoritmu	27
Obrázek 6	Zrekonstruovaný logický datový model pro uložení grafických dat	35
Obrázek 7	Možná poloha bodu B_p vůči přímce p	48
Obrázek 8	Vykreslené hranice katastrálních území pomocí nástroje Simple Spatial Query Visualizator	55
Obrázek 9	Porovnání orientace os	56
Obrázek 10	Vykreslená přehledová mapa z transformovaných dat	58
Obrázek 11	Ukázka vypočtených centroidů	60
Obrázek 12	Hodnota atributu <code>tolerance</code> rovno 1	66
Obrázek 13	Hodnota atributu <code>tolerance</code> rovno 10	67
Obrázek 14	Hodnota atributu <code>tolerance</code> rovno 50	67
Obrázek 15	Hodnota parametru <code>treshold</code> rovna 20	70
Obrázek 16	Hodnota parametru <code>treshold</code> rovna 50	70
Obrázek 17	Hodnota parametru <code>pct_area_change_limit</code> rovna 5	73
Obrázek 18	Hodnota parametru <code>pct_area_change_limit</code> rovna 10	73
Obrázek 19	Hodnota parametru <code>pct_area_change_limit</code> rovna 20	74
Obrázek 20	Schéma postupu od souboru výměnného formátu k testování generalizačních funkcí	77

Seznam tabulek

Tabulka 1	Předdefinované hodnoty atributu <code>sdo_gtype</code>	21
Tabulka 2	Datové bloky ve skupině PKMP	33
Tabulka 3	Vybrané bloky ze skupiny NEMO	33
Tabulka 4	Seznam katastrálních území uložených v databázi	37
Tabulka 5	Nově přidané sloupce do tabulky <code>KATASTRALNI_HRANICE_MAT</code>	65
Tabulka 6	Údaje o počtu vrcholů katastrální hranice při použití funkce <code>SDO_UTIL.REMOVE_DUPLICATE_VERTICES</code>	68
Tabulka 7	Kontrola validity dat po aplikování funkce <code>SDO_UTIL.REMOVE_DUPLICATE_VERTICES</code>	68
Tabulka 8	Počet vrcholů v geometrických popisech hranice kat. území po aplikování funkce <code>SDO_UTIL.SIMPLIFY</code>	69
Tabulka 9	Kontrola validity dat po aplikování funkce <code>SDO_UTIL.SIMPLIFY</code>	69
Tabulka 10	Počet vrcholů v geometrických popisech hranice kat. území po aplikování funkce <code>SDO_SAM.SIMPLIFY_GEOMETRY</code>	71
Tabulka 11	Plochy (v m ²), které uzavírají originální a generalizované geometrické popisy při různých hodnotách parametru <code>pct_area_change_limit</code>	71
Tabulka 12	Procentuální přírůstky u generalizovaných ploch vůči ploše původního popisu	72
Tabulka 13	Kontrola validity dat po aplikování funkce <code>SDO_SAM.SIMPLIFY_GEOMETRY</code>	72
Tabulka 14	Počet vrcholů v geometrických popisech hranice kat. území po aplikování funkce <code>SDO_SAM.SIMPLIFY_LAYER</code>	74

OBSAH

TYPOGRAFICKÉ KONVENCE	1
SEZNAM PŘÍKLADŮ	2
SEZNAM OBRÁZKŮ	3
SEZNAM TABULEK	4
1 ÚVOD	7
2 HLAVNÍ PROJEKTY EU K PROBLEMATICE NAPLNĚNÍ DATABÁZE PROSTOROVÝCH DAT	9
2.1 INSPIRE (INFRASTRUCTURE FOR SPATIAL INFORMATION IN EUROPE).....	9
2.2 GINIE (GEOGRAPHIC INFORMATION NETWORK IN EUROPE).....	9
2.3 OGC (OPEN GIS CONSORCIUM).....	10
2.3.1 <i>Vybrané abstraktní specifikace</i>	11
2.3.1.1 Feature Geometry.....	11
2.3.1.2 OpenGIS(tm) Metadata.....	12
2.3.1.3 Features.....	12
2.3.2 <i>Vybrané specifikace</i>	13
2.3.2.1 OpenGIS Simple Features Specification For SQL.....	13
3 GRAFICKÉ FUNKCE DATABÁZOVÉHO SYSTÉMU ORACLE A JEHO NADSTAVBY SPATIAL	14
3.1 ZÁKLADNÍ FUNKCIONALITA PRO PROSTOROVÁ DATA V DATABÁZI ORACLE.....	14
3.2 POKROČILÁ FUNKCIONALITA ORACLE SPATIAL.....	14
3.3 ÚVOD DO PROSTŘEDÍ ORACLE SPATIAL 10G.....	15
3.3.1 <i>Objektově-relační model uložení dat</i>	15
3.3.2 <i>Podporované geometrické typy</i>	16
3.3.3 <i>Hierarchické uspořádání dat v Oracle Spatial</i>	16
3.3.3.1 Element.....	16
3.3.3.2 Geometrický popis prvku.....	16
3.3.3.3 Vrstva.....	17
3.3.4 <i>Souřadnicový systém</i>	17
3.3.5 <i>Tolerance</i>	17
3.3.6 <i>Dotazovací model</i>	18
3.3.7 <i>Indexování prostorových dat</i>	19
3.3.7.1 R-tree indexování.....	19
3.4 OBJEKTIVÝ DATOVÝ TYP SDO_GEOMETRY.....	20
3.4.1 <i>Atribut sdo_gtype</i>	20
3.4.2 <i>Atribut sdo_srid</i>	22
3.4.3 <i>Atribut sdo_point</i>	22
3.4.4 <i>Atribut sdo_elem_info</i>	22
3.4.5 <i>Atribut sdo_ordinates</i>	23
3.4.6 <i>Metody objektového datového typu SDO_GEOMETRY</i>	23
3.5 GEOMETRY METADATA VIEWS.....	23
3.5.1 <i>Atribut table_name</i>	24
3.5.2 <i>Atribut column_name</i>	24
3.5.3 <i>Atribut diminfo</i>	25
3.5.4 <i>Atribut srid</i>	25
3.6 BALÍK SDO_GEOM.....	25
3.7 BALÍK SDO_UTIL.....	26
3.7.1 <i>SDO_UTIL.REMOVE_DUPLICATE_VERTICES</i>	26
3.7.2 <i>SDO_UTIL.SIMPLIFY</i>	27
3.8 BALÍK SDO_SAM.....	28
3.8.1 <i>SDO_SAM.SIMPLIFY_GEOMETRY</i>	28
3.8.2 <i>SDO_SAM.SIMPLIFY_LAYER</i>	29

4	GENEROVÁNÍ PŘEHLEDOVÉ MAPY Z DAT ISKN	31
4.1	VÝMĚNNÝ FORMÁT ISKN	31
4.2	REKONSTRUKCE LOGICKÉHO DATOVÉHO MODELU Z POPISU VÝMĚNNÉHO FORMÁTU ISKN	32
4.3	IMPORT DAT DO FYZICKÉHO MODELU	36
4.4	GENEROVÁNÍ ÚSEKŮ KATASTRÁLNÍ HRANICE.....	37
4.5	MOŽNÉ ROZŠÍŘENÍ FYZICKÉHO DATOVÉHO MODELU VYTVOŘENÍM TABULKY KATASTRALNI_HRANICE	40
4.5.1	Definice tabulky KATASTRALNI_HRANICE.....	40
4.5.2	Metadata o sloupcích tabulky KATASTRALNI_HRANICE.....	42
4.6	PŘÍPRAVA IMPORTU DAT DO TABULKY KATASTRALNI_HRANICE.....	43
4.6.1	Formát importovaných dat.....	43
4.6.1.1	Atribut sdo_gtype.....	43
4.6.1.2	Atribut sdo_srid.....	44
4.6.1.3	Atribut sdo_point.....	44
4.6.1.4	Atribut sdo_elem_info.....	44
4.6.1.5	Atribut sdo_ordinates.....	45
4.6.2	Rekonstrukce průběhu katastrální hranice	45
4.6.2.1	Program RELtoOO.exe.....	46
4.7	IMPORT DAT DO TABULKY KATASTRALNI_HRANICE	50
4.7.1	Import dat s využitím PL/SQL.....	50
4.7.2	Využití metody „BULK LOADING“	52
5	ZOBRAZENÍ PROSTOROVÝCH DAT	53
5.1	MAPVIEWER.....	53
5.1.1	Využití Simple Spatial Query Visualizer	53
5.2	ORIENTACE OS V MAPVIEWER A S-JTSK	55
5.3	ROZŠÍŘENÍ FYZICKÉHO DATOVÉHO MODELU - VYTVOŘENÍ TABULKY KATASTRALNI_HRANICE_MAT A JEJÍ NAPLNĚNÍ DATY	57
5.3.1	Ztráta přesnosti při uložení v matematickém souřadnicovém systému	58
5.3.2	Naplnění atributu definicni_bod.....	59
5.3.3	Metadata o sloupcích typu SDO_GEOMETRY tabulky KATASTRALNI_HRANICE_MAT.....	60
5.4	KONTROLA VALIDITY DAT	61
5.5	POUŽITÍ OPERÁTORU TOUCH	62
6	TESTOVÁNÍ GENERALIZAČNÍCH FUNKCÍ V ORACLE SPATIAL 10G	64
6.1	TESTOVÁNÍ FUNKCE SDO_UTIL.REMOVE_DUPLICATE_VERTICES	66
6.2	TESTOVÁNÍ FUNKCE SDO_UTIL.SIMPLIFY.....	69
6.3	TESTOVÁNÍ FUNKCE SDO_SAM.SIMPLIFY_GEOMETRY	71
6.4	TESTOVÁNÍ FUNKCE SDO_SAM.SIMPLIFY_LAYER.....	74
7	VYUŽITÍ TOPOLOGICKÉHO DATOVÉHO MODELU.....	76
7.1	ZÁKLADNÍ KONCEPT TOPOLOGICKÉHO MODELU V SPATIAL	76
7.1.1	Uzel (Node).....	76
7.1.2	Hrana (Edge).....	76
7.1.3	Plocha (Face).....	77
8	ZÁVĚR.....	78
	SEZNAM POUŽITÝCH ZKRATEK	80
	LITERATURA.....	81
	PŘÍLOHY	83

1 Úvod

Informační potenciál prostorových¹ dat a jeho využitelnost a přínosy ve všech sférách lidské činnosti jsou dnes všeobecně uznávány. Geografická informace může podpořit organizaci ve stále se rozrůstající Evropské unii a napomoci k řešení problémů. Dále může zlepšit integraci a efektivitu evropských politik, ekonomických a sociálních podmínek [1].

To souvisí s tím, že se stále výrazněji sleduje problematika vytváření tzv. infrastruktury prostorových dat² či prostorové informační infrastruktury (PII), v evropském kontextu známé jako SDI (Spatial Data Infrastructure), GII (Geospatial Information Infrastructure) či GDI (Geographic Data Infrastructure) [1].

Důležitým problémem v oblasti využití prostorových dat, především v geografických informačních systémech (GIS³), je v současné době zajištění interoperability⁴ mezi nejrůznějšími technologickými systémy a aplikačními oblastmi. Rozvoj v oblasti geomatiky⁵ si vyžaduje komplexní přístup ke standardizaci v této oblasti.

Standardy by měly specifikovat metody, nástroje a služby pro správu dat (včetně jejich definování a popisu), jejich sběr, zpracování, analýzu, přístup a přenos takovýchto dat v digitální/elektronické formě mezi různými uživateli, systémy a lokalitami. Standardy by rovněž měly v maximální možné míře navazovat na příslušné standardy v oblasti informačních technologií a dat a vytvořit rámec pro vývoj specifických odvětvových aplikací využívajících geografických dat [5].

Řešení nastíněných problému se neobejde bez přímé podpory a účasti významných firemních společností. Jednou z takových společností je firma Oracle, která se dlouhodobě snaží ve svých produktech budovat funkcionalitu pro manipulaci s prostorovými daty s respektováním vyvíjených mezinárodních standardů.

¹ Prostorová data - data o poloze, tvaru a vztazích mezi jevy reálného světa, pamatovaná zpravidla ve formě souřadnic a topologie [9].

² Infrastruktura prostorových dat - prostředky určené ke shromažďování geografických informací, popisujících uspořádání jevů na Zemi a jejich atributy; infrastruktura zahrnuje materiály, technologii a personál nezbytný pro sběr, zpracování, uchování a distribuci takových informací, realizovaných pro uspokojení širokého sortimentu potřeb [9].

³ Organizovaná kolekce počítačového technického vybavení, programového vybavení, geografických dat a personálu určená k účinnému sběru, pamatování, údržbě, manipulaci, analýze a zobrazování všech forem geograficky vztažené informace [17].

⁴ Interoperabilitou rozumíme schopnost integrace jednotlivých softwarových komponent, které jsou napsané různými výrobci software, a které fungují na různých platformách [16].

⁵ Geomatika - vědecký a technický interdisciplinární obor zabývající se sběrem, distribucí, ukládáním, analýzou, zpracováním a prezentací geografických dat nebo geografických informací [9].

Tato práce se zabývá rozšířením databáze Oracle Spatial 10g⁶, které umožňuje provádění pokročilejších funkcí a procedur nad prostorovými daty. Hlavním cílem by mělo být otestování a popsání možností uložení prostorových dat pomocí objektově – relačního datového modelu Oracle Spatial a funkcionality pro provádění generalizačních postupů. To znamená, otestovat Oracle Spatial jako možnou součást transformačního modulu ve smyslu popsaném v [19]. Tento modul by měl umožňovat přetváření vstupních dat pomocí automatizovaných funkcí kartografické generalizace.

⁶ Konkrétně je v práci testována verze 10.1.0.2

2 Hlavní projekty EU k problematice naplnění databáze prostorových dat

2.1 INSPIRE (Infrastructure For Spatial Information In Europe)

Projekt INSPIRE řeší otázky spojené s identifikací dat, přístupem k nim a následném použití. Na celoevropské úrovni by měl být umožněn otevřený přístup k geodetům a geoinformacím⁷, která budou zdrojem informací pro následné aplikace.

Uživatelé mají své nároky a požadavky na to, jakými (geo)daty by měla být naplněna infrastruktura geodat. Jinou otázkou pak zůstává, jakým způsobem data uživateli poskytnout, ať už na úrovni technické, či právní.

Důležitým cílem zůstává vývoj infrastruktury prostorových dat k její plné interoperabilitě. Struktura vývoje dané infrastruktury je popsána v [2].

Iniciativa INSPIRE je podporována celoevropskou organizací EUROGI (European Umbrella Organisation For Geographic Information) a organizací OGC (Open GIS Consortium). Koordinátorem projektu je University of Sheffield. Expertem České republiky v této věci je doc.RNDr. Milan Konečný, CSc [1].

2.2 GINIE (Geographic Information Network In Europe)

Souběžně s projektem INSPIRE běžel s ním související projekt GINIE. Jednalo se o projekt financovaný z programu Information Society Technology (IST) Evropské Unie. Řešení projektu GINIE probíhalo od 1.11.2001 do 31.1.2004 [1].

Hlavní záměry projektu GINIE podle [3] jsou:

- Vývoj geografické informační strategie na evropské úrovni.
- Vytvoření přehledu o přístupnosti, využití a rozšíření geodat.
- Podílet se ve spolupráci s projektem INSPIRE na vytvoření evropské infrastruktury pro prostorové informace.

⁷ Geoinformace - informace týkající se jevů implicitně nebo explicitně přidružených k místu vztahenému k Zemi; znalost získaná jako výsledek syntézy, analýzy nebo integrace geografických dat [9].

Jedním z dílčích výsledků projektu GINIE je souhrn bodů, jejichž splnění zaručuje úspěšnost infrastruktury prostorových dat [4]. Ta je úspěšná, pokud:

- je vyvinuta, využívána a udržována několika institucemi zodpovědnými za zdroje klíčových dat zahrnujících socioekonomická data, data o životním prostředí, o pozemcích a majetku a referenční data (např. adresy, administrativní hranice, základní prvky území a topografické jevy).
- je připravena poskytnout odpovědi na skutečné požadavky, zejména v době naléhavé potřeby jako např. v době přírodní nebo člověkem zaviněné katastrofy.
- její základní data odpovídají společným specifikacím, data jsou aktualizována a snadno zjištělná a přístupná.
- je víceúrovňová, od místní po regionální a národní úroveň.
- existuje funkční jednotka v právní struktuře, a to na všech úrovních.
- existuje jasná autorita, která řídí a koordinuje celý systém.
- může zdůvodnit svou existenci dostatečnou podporou ekonomiky.

Konečným výsledkem projektu GINIE je plně funkční dozorčí rada ABGI (Advisory Board On Geographic Information), která je tvořena zástupci vládních celků, průmyslu, výzkumu a Evropskou komisí a zabezpečující využívání GI v Evropě pro všechny potenciální uživatele.

2.3 OGC (Open GIS Consortium)

OGC je nezisková mezinárodní organizace mající vůdčí roli při vývoji standardů pro geoprostorové služby. OGC sdružuje nejvýznamnější firmy dodávající hardwarové a softwarové produkty pro geografické informační systémy, stejně tak jako různé vládní organizace či pracoviště vysokých škol. Společně vyvíjí otevřené a rozšiřitelné aplikační programové rozhraní pro GIS a související technologie.

Standard podle OGC musí:

- být vytvořen jako otevřený.
- být volně šířitelný.
- umožňovat volný přístup ke svému popisu.
- nediskriminovat jakoukoli osobu či organizaci.
- mít neutrální technologickou specifikaci.

OGC produkuje širokou škálu dokumentů pro různé účely. Hlavními dokumenty, které OGC vyvíjí, jsou tzv. **specifikace**. Poté, co je specifikace přijata, nese za ní odpovědnost určitá pracovní skupina, která pak musí mimo jiné vytvořit metodu ke správě a případné revizi specifikace. Tento proces je detailněji popsán v [6].

Kromě specifikací jsou vyvíjeny tzv. **abstraktní specifikace**, které jsou méně techničtější. Abstraktní specifikace nabízí konceptuální podklad pro většinu OGC specifikací.

2.3.1 Vybrané abstraktní specifikace

2.3.1.1 Feature Geometry

Tento standard nabízí konceptuální schémata⁸ pro popisování prostorových charakteristik geografických prvků a množinu prostorových operátorů spojených s těmito schématy. Standardizace v této oblasti bude základní pro ostatní geografické informační standardy.

Prvek je zde chápán jako abstrakce fenoménu reálného světa. Prvek nazveme geografickým prvkem, jestliže je polohově spojen se Zemí. Vektorová data sestávají z geometrických a topologických primitiv, použitých k budování geometrických popisů prvků, které vyjadřují prostorové charakteristiky geografických prvků.

Geometrie nabízí prostředky pro kvantitativní popis prostorového objektu za použití souřadnic a matematických funkcí, umožňuje vyjádření prostorových charakteristik prvků, včetně dimenze, pozice, velikosti, tvaru a orientace. Matematické funkce použité pro geometrický popis prvku závisí na typu souřadnicového referenčního systému, použitého k definování pozice v daném prostoru.

Topologie se zabývá charakteristikami geometrických obrazců, které zůstávají neměnné, pokud je prostor deformován elasticky a kontinuálně. Topologie je nejběžněji používána k popisu konektivity n-dimensionálního obrazce, čili vlastnosti, která je invariantní vůči souvislé transformaci obrazce.

Prostorové operátory jsou funkce a procedury, které se používají k dotazování, vytváření, modifikaci nebo odstranění geometrických popisů prostorových objektů. Tento mezinárodní standard definuje výčet těchto operátorů za účelem vytvoření standardu pro jejich definici a implementaci. Cíle jsou:

- Definovat prostorové operátory jednoznačně.
- Použít tyto definice k definování množiny standardních operací, které budou tvořit bázi systémů.
- Definovat „operator algebra“, který bude dovolovat kombinaci základních operátorů k použití při dotazování a manipulaci s geografickými daty.

⁸ Konceptuální schéma má význam zejména v procesu návrhu databázového systému. Nejčastěji užívaná je grafická podoba – ER-diagram.

Standardizovaná konceptuální schémata pro prostorové charakteristiky budou zvyšovat schopnost k sdílení geografických informací mezi aplikacemi. Tato schémata budou použita geografickými informačními systémy a vývojáři softwaru k poskytování konzistentních prostorových datových struktur uživateli geografických informací.

2.3.1.2 OpenGIS(tm) Metadata

Uvědomování si důležitosti geografie a toho, jak věci spolu prostorově souvisí, společně s pokrokem elektronické technologie, zapříčinilo rozvoj v celosvětovém používání digitální geografické informace a geografických informačních systémů. S tím, jak počet, komplexnost a rozmanitost geografických informací roste, nabývá na významu metoda pro zajištění srozumitelnosti všech aspektů těchto informací.

Digitální geografická data se pokouší modelovat a popsat reálný, skutečný svět pro použití v počítačových analýzách a graficky zobrazit tyto informace. Jakýkoli popis reality je vždy abstrakcí, vždy částečný, a vždy právě jeden z mnoha možných „pohledů“. Tento „pohled“, či model reálného světa není přesné zdvojení; některé věci jsou aproximovány, jiné jsou zjednodušeny a některé jsou ignorovány. Málokdy máme perfektní, kompletní a korektní data. K zajištění toho, aby data nebyla chybně použita, musí být plně zdokumentovány předpoklady a omezení ovlivňující tvorbu dat. Metadata dovolují producentům dat popsat tato data komplexním způsobem, uživatelé pak mohou rozumět jednotlivým předpokladům a omezením spojených s tvorbou konkrétních dat a mohou zhodnotit použitelnost dat pro jejich zamýšlené použití.

Typické je, že jsou geografická data využívána jinými lidmi, než samotným výrobcem. Data vyprodukuje jedna společnost a používá je řada jiných organizací. Řádná dokumentace bude umožňovat lepší porozumění datům a umožní jejich využití patřičným způsobem. Tím, jak producenti geografických dat a jejich uživatelé pracují se stále větším množstvím těchto dat, řádná dokumentace jim bude nabízet hlubší znalosti jejich dat a umožní jim lépe řídit výrobu dat, jejich uchování, aktualizaci a opětovné použití.

Cílem tohoto standardu je poskytnout strukturu pro popis digitálních geografických dat. Je určený k použití analytiky informačních systémů, programátory a producenty geografických informačních systémů, ale také ostatními organizacemi k pochopení základních principů a souhrnných požadavků na standardizaci geografické informace. Standard rovněž zavádí příslušnou terminologii týkající se problematiky metadat.

2.3.1.3 Features

Prvkový objekt (v softwarovém pojetí) odpovídá abstraktní entitě či skutečnému objektu (prvku) reálného světa. Atributy tohoto prvkového

objektu popisují měřitelné či popsatelné jevy o této entitě. Užitím konceptu referenčních systémů jsou prvky souřadnicově spjaty s reálným světem.

Prvek je základní jednotkou geoprostorové informace. U prvků může dojít k značné definiční různorodosti, například v závislosti na aplikaci nebo zájmech „shromažďovatele“ informací.

Digitální geoprostorová informace je geoprostorová informace, která je zakódována do digitální formy. Kódování probíhá pomocí počítačových prostředků, které pomáhají zautomatizovat záležitosti zpracování geoprostorové informace: ukládání, přenos, analýzy, a tak dále.

Je zde mnoho odlišných způsobů, jak vytvořit digitální reprezentaci geoprostorové informace. Tato pestrost různých alternativ je spíše nevýhodou. Tento abstrakt se pokouší vnést do této oblasti vnitřní pořádek a uspořádání.

2.3.2 Vybrané specifikace

2.3.2.1 OpenGIS Simple Features Specification For SQL⁹

Cílem této specifikace je definovat standardní SQL schéma, které by podporovalo ukládání, vyhledávání, dotazování a obnovování prvkových množin přes ODBC API.

Geoprostorové prvkové množiny budou konceptuálně uloženy jako tabulky se sloupci obsahujícími geometrické popisy prvků¹⁰ v relační SŘBD (systém řízení báze dat).

⁹ SQL- Structured Query Language; jedná se o neprocedurální jazyk určený pro databázové systémy.

¹⁰ V textu je používán termín **geometrický popis prvku** (*geometry*) tak, jak je chápán v prostředí Spatial, viz kapitola 3.3.3.2. Zde pod pojmem geometrický popis prvku rozumíme geometrickou reprezentaci tvaru prostorového prvku vůči nějakému souřadnicovému systému.

3 Grafické funkce databázového systému Oracle a jeho nadstavby Spatial

3.1 Základní funkcionality pro prostorová data v databázi Oracle

Základní funkcionality pro správu prostorových dat je nabízena ve všech edicích databáze Oracle. Tato funkcionality vychází ze současných požadavků na zpracování prostorových dat, především pak velkého objemu manipulovaných dat.

Mezi základní prvky správy prostorových dat patří možnost vytváření prostorového indexu, dotazování dat a manipulování s nimi pomocí dotazovacího jazyka SQL. Databáze Oracle podporuje též velké množství projekcí a souřadnicových systémů, k dispozici jsou funkce pro vzájemnou transformaci mezi nimi. V celku tak může jít o základní funkcionality geografického informačního systému.

3.2 Pokročilá funkcionality Oracle Spatial

Úvodem této kapitoly bych nejprve rozšířil seznam typografických konvencí:

- při popisech a ukázkách části programového kódu v jazyce Pascal¹¹, SQL, či PL/SQL¹² jsou klíčová slova zapsána velkými písmeny **TUČNĚ**, zbylý text zdrojového kódu fontem Courier New.

V edici Enterprise databáze Oracle je k dispozici nadstavba této databáze umožňující pokročilejší prostorové funkce – Spatial. Testování vybraných funkcí na vybrané množině dat bylo jednou z hlavních náplní této práce. Jednalo se především o funkce, které umožňují vhodnou volbou počátečních parametrů zjednodušení geometrického popisu prostorových prvků.

Procedury a funkce Oracle Spatial jsou nabízeny jako podprogramy v balících¹³ PL/SQL. Spatial má implicitně k dispozici několik balíků procedur a funkcí. Jedná se především o balíky

¹¹ Programovací jazyk Pascal byl navržen začátkem 70. let. Jeho autorem je profesor Niklaus Wirth z Vysoké školy technické v Curychu. Název jazyka byl zvolen na počest francouzského filosofa, matematika a fyzika Blaise Pascala.

¹² PL/SQL je procedurální rozšíření jazyka SQL pocházející od firmy Oracle.

¹³ Procedury a funkce, které spolu souvisí, je možno pro další použití vkládat do tzv. balíků. Zde jsou obsaženy všechny typy, konstanty, proměnné, výjimky, kurzory, procedury a funkce, ke kterým je možno přistupovat z ostatních aplikací (pokud mají k danému balíku patřičná přístupová práva).

SDO_GEOM, SDO_AGGR, SDO_CS, SDO_LRS, SDO_MIGRATE, SDO_TUNE, SDO_UTIL, SDO_SAM. Předmětem mého zájmu bylo pouze několik vybraných funkcí, které budou popsány detailněji v dalším textu. Stručný popis podprogramů základních balíčků lze nalézt v přílohách.

Obecně pro podprogramy z těchto balíčků platí několik společných zásad. Nevyžadují například definování prostorového indexu a ani prostorový index nepoužívají, pokud je definován. Pokud jsou vstupními parametry procedury nebo funkce Oracle Spatial geometrické popisy dvou prvků, musí být oba vztaženy ke shodnému souřadnicovému systému a podobně.

Pro efektivní použití nabízených podprogramů je nutné pochopit základní principy a filosofii prostředí Oracle Spatial.

3.3 Úvod do prostředí Oracle Spatial 10g

Oracle Spatial (dále jen Spatial) je integrovaná množina funkcí a procedur, které umožňují uložení, přístup a analýzu prostorových dat rychlým a efektivním způsobem v databázi Oracle.

Spatial se skládá z následujících komponent:

- Schématu předepisujícího ukládání, syntaxy a sémantiku podporovaných geometrických datových typů.
- Mechanismu prostorového indexování.
- Množiny operátorů a funkcí pro provádění prostorových dotazů a analýz.
- Podpůrných utilit.

3.3.1 Objektově-relační model uložení dat

Spatial podporuje objektově-relační model uložení prostorových dat. Pro uložení prostorových dat využívá objektového datového typu *SDO_GEOMETRY*. V jedné tabulce jsou tak pro prvek popisná data uložena společně s prostorovými.

Výhody nabízené objektově-relačním modelem jsou především:

- Podpora mnoha geometrických typů, např. oblouků, kružnic, liniových řetězců či polygonů.
- Snadné použití při vytváření a údržbě prostorových indexů a dotazů.
- Uložení geometrických popisů prvků v jednom řádku a jednom sloupci tabulky.
- Optimální výkon.

3.3.2 Podporované geometrické typy

Geometrický popis prvku je de facto seřazená sekvence vrcholů spojených příkými liniovými segmenty nebo kruhovými oblouky. Sémantika geometrického popisu prvku je určena jeho typem. Spatial podporuje několik primitivních geometrických typů, a to:

- Bod a shluk bodů.
- Liniový řetězec.
- Polygon o n bodech.
- Liniový řetězec, kde je každý dílčí úsek tvořen kruhovým obloukem.
- Polygon, kde je každý dílčí úsek tvořen kruhovým obloukem.
- Smíšený liniový řetězec – liniový řetězec, kde je každý dílčí úsek tvořen buď příkým liniovým segmentem, nebo kruhovým obloukem.
- Smíšený polygon – polygon, kde je každý dílčí úsek tvořen buď příkým liniovým segmentem, nebo kruhovým obloukem.
- Kružnice.
- Pravoúhelník.

3.3.3 Hierarchické uspořádání dat v Oracle Spatial

Spatial na sloupec typu *SDO_GEOMETRY* nahlíží jako na vrstvu. Vrstva obsahuje jednotlivé prostorové prvky tvořené základními geometrickými elementy.

3.3.3.1 Element

Element je základní stavební blok geometrického popisu prvku. Podporované prostorové typy elementů jsou body, liniové řetězce a polygony. Souřadnice v elementu jsou ukládány jako pár souřadnic X,Y.

Bodové elementy sestávají z jednoho páru X,Y souřadnic. **Liniové elementy**, reprezentující liniový segment elementu, sestávají ze dvou párů souřadnic X,Y. **Polygonové elementy** tvoří dva páry souřadnic X,Y pro každý liniový segment polygonu. Souřadnice jsou definovány v požadovaném pořadí (pro vnější polygon v protisměru chodu hodinových ručiček, pro vnitřní polygon ve směru chodu hodinových ručiček).

3.3.3.2 Geometrický popis prvku

Geometrický popis prvku je reprezentace prostorového prvku, modelovaná jako uspořádaná množina primitivních elementů. Geometrický popis může tvořit samostatný element podporovaného primitivního typu, nebo homogenní či heterogenní soubor elementů.

Multipolygon, který bude reprezentovat množinu několika polygonů, je homogenní kolekce. O heterogenní kolekci se jedná v případě, že jsou jednotlivé elementy rozdílných typů, například bod a polygon.

3.3.3.3 Vrstva

Vrstva je soubor geometrických popisů prvků majících shodnou množinu atributů. Jedna vrstva může obsahovat například topografické prvky, druhá může popisovat hustotu zalidnění a třetí síť silnic a mostů v určité oblasti (linie a body). Všechny geometrické popisy prvků dané vrstvy jsou uloženy v databázi ve standardní tabulce.

3.3.4 Souřadnicový systém

Souřadnicový systém (též prostorový referenční systém) dovoluje interpretovat množinu souřadnic jako reprezentaci polohy v reálném prostoru. Prostorová data jsou spojena s určitým souřadnicovým systémem, který může být souřadnicově spjatý s určitou aproximací Země, nebo jako například kartézský systém souřadnic nemusí být souřadnicově spjatý s žádnou aproximací Země.

Spatial nabízí podporu pro mnoho různých souřadnicových systémů a pro konvertování dat mezi těmito systémy. Bohužel, Spatial v současné době nepodporuje souřadnicový systém S-JTSK.

Prostorová data mohou být v prostředí Spatial spojena s kartézským, geodetickým, projektivním, nebo místním souřadnicovým systémem.

3.3.5 Tolerance

Tolerance je použita ke spojení stupně přesnosti s prostorovými daty. Určuje vzdálenost, ve které od sebe mohou být dva body a stále budou považovány za shodné. Hodnota odchylky musí být kladné číslo různé od nuly. Význam hodnoty tolerance závisí na tom, zda jsou či nejsou prostorová data spojena s geodetickým souřadnicovým systémem.

- Pro data v geodetickém souřadnicovém systému je hodnota tolerance v metrech. Například hodnota 100 znamená toleranci 100 metrů. Hodnota tolerance pro geodetická data by neměla být menší než 0.001 (1 milimetr). Pokud určíme menší hodnotu tolerance než 0.001, Spatial použije hodnotu 0.001.
- Pro data v jiném než geodetickém systému je hodnota tolerance v jednotkách souřadnicového systému, ve kterém jsou data vyjádřena. To znamená, že pokud je zvolenou jednotkou míle, hodnota 0.005 znamená toleranci 1/200 míle.

V obou případech ale platí, že čím menší je hodnota tolerance, tím jsou data přesnější. Hodnota tolerance se specifikuje ve dvou případech:

- V definici metadat pro prvky ve vrstvě.
- Jako vstupní parametr některých funkcí.

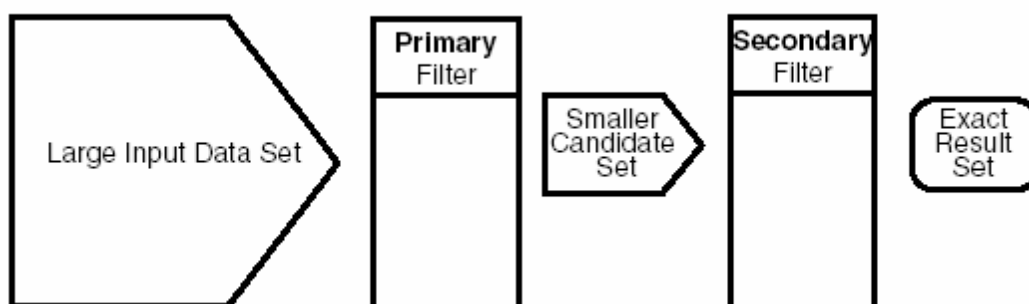
3.3.6 Dotazovací model

Spatial používá k řešení prostorových dotazů a prostorových spojení dvouvrstvý dotazovací model. Během něj jsou vykonány k vyhodnocení dotazů dvě odlišné operace. Výstupem kombinace těchto dvou operací je hledaná množina geometrických popisů prvků.

Vykonané operace jsou označovány jako primární a sekundární operace filtrování (primární a sekundární filtr):

- Primární filtr umožňuje rychlý výběr potenciálních záznamů, které jsou předány sekundárnímu filtru.
- Sekundární filtr aplikuje přesné výpočty na geometrické popisy prvků, které mu předal filtr primární. Sekundární filtr dává přesnou odpověď na prostorový dotaz.

Obrázek 1 ilustruje vztah mezi primárním a sekundárním filtrem.



Obr. 1: Dotazovací model.

Jak ukazuje obrázek 1, operace primárního filtrování aplikovaná na rozsáhlou množinu vstupních dat produkuje množinu menší, která obsahuje hledané geometrické popisy prvků, ale může také obsahovat záznamy navíc, které nemají být do výsledku zahrnuty. K odstranění nežádoucích záznamů dojde aplikováním sekundárního filtru.

Cílem primárního filtru je rychle vytvořit podmnožinu dat a redukovat zatížení zpracování pro sekundární filtr. K tomu Spatial používá prostorový index, jehož charakteristiky rozhodují o úspěšnosti primárního filtru.

3.3.7 Indexování prostorových dat

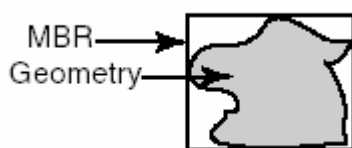
Zavedení možností prostorového indexování do databáze Oracle je klíčové pro produkt Spatial. Prostorový index nabízí mechanismus k vyhledávání založený na prostorových kritériích.

Prostorový index je potřebný k:

- Nalezení geometrického popisu prvku uvnitř indexovaného datového prostoru, které mají souvislost s daným bodem nebo oblastí zájmu (dotazovací okno).
- Nalezení dvou geometrických popisů prvků z různých indexovaných prostorů, které spoluprostorově souvisí (prostorové spojení).

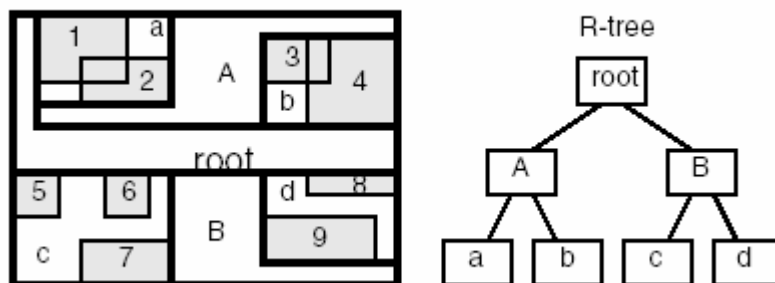
3.3.7.1 R-tree indexování

Prostorový R-tree index může indexovat prostorová data až ve čtyřech dimenzích. R-tree index aproximuje každý geometrický popis prvku samostatným pravoúhlým čtyřúhelníkem (též *minimum bounding rectangle* – MBR), jak ukazuje obrázek 2.



Obr.2.: Ukázka MBR.

R-tree index pro vrstvu geometrických popisů sestává z hierarchického indexu na konvexní obálky prvků ve vrstvě, jak je ukázáno na obrázku 3.



Obr. 3.: Hierarchický R-tree index.

3.4 Objektový datový typ SDO_GEOMETRY

Oracle Spatial ukládá prostorová data společně s daty atributovými podle objektově-relačního datového modelu. Geometrický popis prostorového objektu je uložen pomocí objektového typu *SDO_GEOMETRY*. Jakákoli tabulka, která má sloupec či sloupce typu *SDO_GEOMETRY*, musí mít jiný sloupec, který definuje unikátní primární klíč pro tuto tabulku. Tabulky obsahující alespoň jeden sloupec typu *SDO_GEOMETRY* jsou označovány jako **geometrické tabulky** (*geometry tables*).

Objektový typ *SDO_GEOMETRY* je v prostředí Spatial definován následujícím způsobem:

```
CREATE TYPE sdo_geometry AS OBJECT (  
    sdo_gtype          NUMBER,  
    sdo_srid           NUMBER,  
    sdo_point          SDO_POINT_TYPE,  
    sdo_elem_info      MDSYS.SDO_ELEM_INFO_ARRAY,  
    sdo_ordinates      MDSYS.SDO_ORDINATE_ARRAY);
```

Př.1: Definice objektového datového typu *SDO_GEOMETRY*.

V definici typu *SDO_GEOMETRY* jsou kromě numerického datového typu *NUMBER* použity rovněž typy *SDO_POINT_TYPE*, *SDO_ELEM_INFO_ARRAY* a *SDO_ORDINATE_ARRAY*, které jsou definovány jako:

```
CREATE TYPE sdo_point_type AS OBJECT (  
    x          NUMBER,  
    y          NUMBER,  
    z          NUMBER) ;
```

```
CREATE TYPE sdo_elem_info_array AS VARRAY (1048576) OF  
    NUMBER;
```

```
CREATE TYPE sdo_ordinate_array AS VARRAY (1048576) OF  
    NUMBER;
```

Př.2: Definice datových typů *SDO_POINT_TYPE*, *SDO_ELEM_INFO_ARRAY* a *SDO_ORDINATE_ARRAY*.

3.4.1 Atribut sdo_gtype

Atribut *sdo_gtype* označuje geometrický typ geometrického popisu prvku. Možné geometrické typy korespondují s těmi, které jsou

specifikovány v [7]. V prostředí Spatial je přímá shoda mezi pojmenováním a sémantikou typů s typy, uvedenými v [7].

Kód typu prvku sestává ze čtyř číslic. Lze ho psát ve tvaru dltt, kde:

- d - identifikuje počet dimenzí (2, 3, nebo 4).
- l - tato hodnota je užitečná při využití lineárního odkazování¹⁴, což není případ této práce, proto stačí ponechat výchozí hodnotu **0**.
- tt - identifikuje geometrický typ (hodnoty 00 až 07 jsou předdefinované, 08 až 99 jsou rezervovány pro uživatelem definované typy).

V tabulce 1 je uveden přehled předdefinovaných hodnot atributu `sdo_gtype`.

Hodnota	Geometrický typ	Geometrický popis prvku
dl00	UNKNOWN_GEOMETRY	Spatial ignoruje tento popis prvku.
dl01	POINT	Obsahuje jeden bod
dl02	LINE nebo CURVE	Sestává z jednoho liniového řetězce, který může obsahovat přímé či obloukové segmenty, nebo obojí. (LINE a CURVE jsou v tomto kontextu synonyma.)
dl03	POLYGON	Obsahuje jeden polygon s nebo bez děr. ¹⁵
dl04	COLLECTION	Jedná se o heterogenní soubor prvků. ¹⁶
dl05	MULTIPOINT	Obsahuje jeden nebo více bodů.
dl06	MULTILINE nebo MULTICURVE	Obsahuje jeden nebo více liniových řetězců.
dl07	MULTIPOLYGON	Prvek může obsahovat disjunktní polygony (více než jedna vnější hranice).

Tab. 1: Předdefinované hodnoty atributu `sdo_gtype`.

Všechna geometrická data v rámci jedné vrstvy musí mít stejný počet dimenzí. Nemůžeme například v jedné vrstvě uchovávat dvojdimenzionální a trojdimenzionální data .

¹⁴ Lineární odkazování – vhodný prostředek pro připojení atributů nebo událostí k poloze nebo části rovinného prvku.

¹⁵ Pro polygon s dírami se jako první vloží externí hranice. Za ní pak následuje jakákoli vnitřní hranice.

¹⁶ Heterogenní kolekci tvoří prvky různých geometrických typů.

3.4.2 Atribut `sdo_srid`

Atribut `sdo_srid` slouží k identifikaci souřadnicového systému (prostorového referenčního systému), ve kterém je prostorový prvek geometricky popsán. Jestliže je hodnota atributu `sdo_srid` **NULL**, znamená to, že s geometrickým popisem prvku není asociován žádný souřadnicový systém. V případě, kdy atribut `sdo_srid` nenabývá hodnoty **NULL**, musí obsahovat jednu z hodnot ze sloupce `srid` z tabulky `MDSYS.CS_SRS`¹⁷.

Všechny geometrické popisy prvků v rámci jednoho sloupce typu `SDO_GEOMETRY` musí mít shodnou hodnotu atributu `sdo_srid`.

3.4.3 Atribut `sdo_point`

Atribut `sdo_point` je definován pomocí objektového typu `SDO_POINT_TYPE`. Tento typ má atributy `x`, `y`, `z`, všechny typu `NUMBER`. Jestliže jsou pole `sdo_elem_info` a `sdo_ordinates` obojí prázdná (mají hodnotu **NULL**) a zároveň hodnota atributu `sdo_point` není prázdná, považují se hodnoty atributů `x` a `y` za souřadnice bodu. V jiném případě je atribut `sdo_point` ignorován.

Optimálně by se měly souřadnice bodu uložit do atributu `sdo_point`; a pokud jsou ve vrstvě pouze body, doporučuje se ukládat souřadnice těchto bodů právě do atributu `sdo_point`.

3.4.4 Atribut `sdo_elem_info`

Atribut `sdo_elem_info` popisuje způsob interpretace souřadnic geometrického popisu prvku uložených v atributu `sdo_ordinates`.

Atribut `sdo_elem_info` lze interpretovat pomocí trojice hodnot následujícím způsobem:

- `sdo_starting_offset` – indikuje uložení první souřadnice pro daný element uvnitř pole `sdo_ordinates`. Hodnoty offsetu začínají číslem 1. První souřadnice pro první element tak bude na pozici `SDO_GEOMETRY.sdo_ordinates(1)`.
- `sdo_etype` – indikuje typ elementu. Přípustné hodnoty jsou uvedeny v tabulce 2.
- `sdo_interpretation` – v závislosti na hodnotě `sdo_etype` vyjadřuje jednu ze dvou věcí:

¹⁷ Systémová tabulka obsahující přes 900 nadefinovaných souřadnicových systémů.

- Pokud je hodnota atributu `sdo_etype` rovna **4**, **1005** nebo **2005**, pak hodnota `sdo_interpretation` udává, kolik dalších trojic hodnot je použitých k popisu elementu.
- Pokud je hodnota `sdo_etype` rovna **1**, **2**, **1003** nebo **2003**, pak `sdo_interpretation` vymezuje, jak interpretovat pořadí souřadnic elementu. Například může nést informaci o tom, zda je hranice polygonu tvořena sekvencí spojených přímých linií či kruhových oblouků.

3.4.5 Atribut `sdo_ordinates`

Atribut `sdo_ordinates` je definován pomocí pole čísel proměnné délky typu `NUMBER`. V poli jsou uloženy souřadnice bodů, které vytváří geometrický popis hranice prostorového prvku. Toto pole musí být vždy užito společně s polem `sdo_elem_info`. Hodnoty v poli jsou seřazeny podle dimenze. Například polygon, jehož hranici tvoří čtyři dvojdimenzionální body, je uložen jako množina hodnot $\{X1, Y1, X2, Y2, X3, Y3, X4, Y4, X1, Y1\}$. V třídídimenzionálním prostoru by byl uložen jako množina hodnot $\{X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, X4, Y4, Z4, X1, Y1, Z1\}$. Testovaná verze Oracle Spatial podporuje pouze dvojdimenzionální prostorové objekty. Z tohoto důvodu proces tvorby prostorového indexu, operátory a funkce pracující s objektovým typem `SDO_GEOMETRY`, ignorují hodnotu Z.

Hodnoty v poli `sdo_ordinates` nesmí mít hodnotu null.

3.4.6 Metody objektového datového typu `SDO_GEOMETRY`

Objektový datový typ `SDO_GEOMETRY` má rovněž metody, které poskytují přístup k některým jeho atributům. Celkem se jedná o tři následující metody:

- `GET_DIMS` – vrací počet dimenzí geometrického popisu prvku (hodnota `d` z atributu `sdo_gtype`).
- `GET_GTYPE` – vrací geometrický typ geometrického popisu prvku (hodnota `tt` z atributu `sdo_gtype`).
- `GET_LRS_DIM` – vrací hodnotu `l` z atributu `sdo_gtype` geometrického popisu prvku.

3.5 Geometry Metadata Views

Metadata o prostorovém prvku uchovávají informace o jeho geometrickém popisu týkající se rozsahu možných nabývaných hodnot v každé dimenzi včetně tolerance (odchylny) v dané dimenzi. Jsou

uložena v globální tabulce, jejímž vlastníkem je systém. Každý uživatel Oracle Spatial má k dispozici následující pohledy (views):

- `USER_SDO_GEOM_METADATA` obsahuje informace o všech prostorových tabulkách vlastněných uživatelem. Z toho vyplývá, že uživatel Spatial do něj musí vkládat metadata týkající se sloupců typu `SDO_GEOMETRY`, které si vytvořil. Toto je jediný pohled, na který má uživatel oprávnění provádět změny (příkaz `UPDATE`).
- `ALL_SDO_GEOM_METADATA` obsahuje metadata o všech prostorových tabulkách. Uživatel k nim může přistupovat pomocí příkazu `SELECT`.

Při vložení nového záznamu do pohledu `USER_SDO_GEOM_METADATA` Spatial zajišťuje automatickou aktualizaci pohledu `ALL_SDO_GEOM_METADATA`.

Pohled na metadata má definován následující atributy:

```
(
table_name      VARCHAR2(32),
column_name     VARCHAR2(32),
diminfo         SDO_DIM_ARRAY,
srid            NUMBER
);
```

Př.3: Atributy metadatového pohledu.

Kromě toho má pohled `ALL_SDO_GEOM_METADATA` atribut `owner`, identifikující schéma, které vlastní tabulku uvedenou v atributu `table_name`.

3.5.1 Atribut `table_name`

Atribut `table_name` obsahuje název tabulky obsahující sloupec typu `SDO_GEOMETRY`.

3.5.2 Atribut `column_name`

Atribut `column_name` obsahuje název sloupce typu `SDO_GEOMETRY`.

3.5.3 Atribut `diminfo`

Atribut `diminfo` je typu `SDO_DIM_ARRAY`, který je definován následujícím způsobem:

```
CREATE TYPE SDO_DIM_ARRAY AS VARRAY(4) OF
  SDO_DIM_ELEMENT;
```

Typ `SDO_DIM_ELEMENT` je definován jako:

```
CREATE TYPE SDO_DIM_ELEMENT AS OBJECT (
  sdo_dimname          VARCHAR2(64),
  sdo_lb               NUMBER,
  sdo_ub               NUMBER,
  sdo_tolerance        NUMBER);
```

Př.4: Definice typů `SDO_DIM_ARRAY` a `SDO_DIM_ELEMENT`.

Z předchozí definice je vidět, že můžeme mít prvek popsán až ve čtyřech dimenzích. Podle počtu dimenzí pak atribut `diminfo` obsahuje příslušný počet objektů `SDO_DIM_ELEMENT`, z nichž každý musí mít hodnoty atributů `sdo_lb`¹⁸, `sdo_ub`¹⁹ a `sdo_tolerance`²⁰ různé od hodnoty **NULL**.

Řazení souřadnic bodů v poli atributu `sdo_ordinates` by mělo odpovídat pořadí definic jednotlivých dimenzí v atributu `diminfo`. Pokud například atribut `sdo_ordinates` obsahuje pole `{X1, Y1, ..., Xn, Yn}`, pak jako první musí být definována v poli `diminfo` dimenze X a jako druhá dimenze Y.

3.5.4 Atribut `srid`

Atribut `srid` obsahuje kód souřadnicového systému, který je spojen s geometrickými popisy prvků ve sloupci `SDO_GEOMETRY`, nebo hodnotu **NULL**, jestliže s geometrickým popisem prvku není asociován žádný konkrétní souřadnicový systém.

3.6 Balík `SDO_GEOM`

PL/SQL podprogramy z balíku `SDO_GEOM` umožňují kontrolovat validitu geometrického popisu prvku, vypočítat centroid prvku, délku či plochu objektu, vytvořit buffer nebo konvexní obálku. Rovněž je možno pomocí

¹⁸ Atribut udává počátek rozsahu možných hodnot v dané dimenzi.

¹⁹ Atribut udává konec rozsahu možných hodnot v dané dimenzi.

²⁰ Atribut udává vzdálenost, ve které jsou dva body vnímány jako identické.

podprogramů v tomto balíku vypočítat vzdálenost mezi objekty, vytvářet ze stávajících objektů nové a podobně.

3.7 Balík SDO_UTIL

Tento balík obsahuje velké množství podprogramů, z nichž nejdůležitější z hlediska mé práce jsou funkce **SDO_UTIL.REMOVE_DUPLICATE_VERTICES** a **SDO_UTIL.SIMPLIFY**.

3.7.1 SDO_UTIL.REMOVE_DUPLICATE_VERTICES

Primárním účelem této funkce je eliminovat duplicitní vrcholy z geometrického popisu prostorového prvku, ačkoli tyto duplicity jsem se snažil odstranit před samotným importem dat do databáze. Tato funkce jsem se pomocí vhodné volby jednoho ze vstupních parametrů snažil využít ke generalizačním účelům.

Návratovou hodnotou funkce je objekt typu *SDO_GEOMETRY*, obsahující geometrický popis prvku bez vrcholových duplicit.

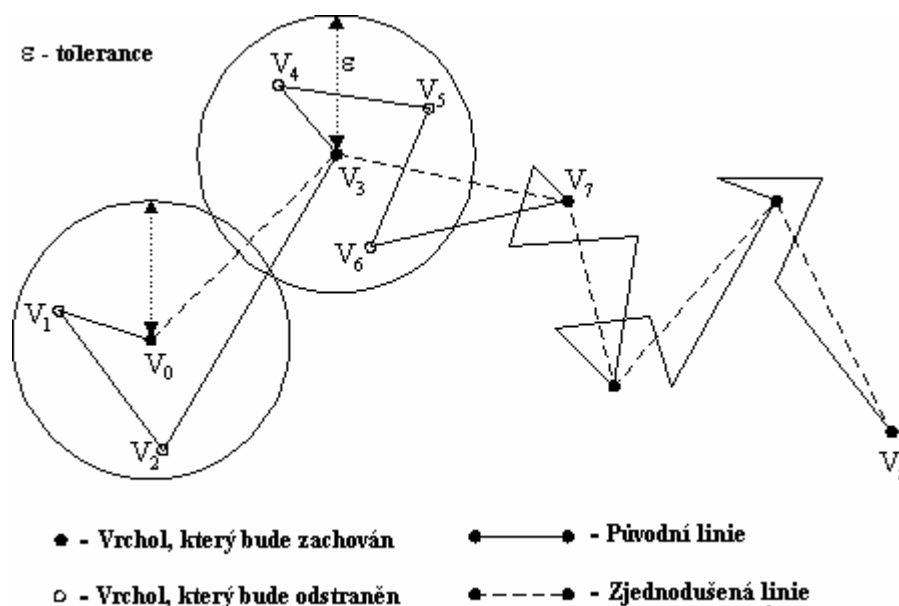
Formát funkce je následující:

```
SDO_UTIL.REMOVE_DUPLICATE_VERTICES (  
  geometry      IN SDO_GEOMETRY,  
  tolerance     IN NUMBER  
) RETURN SDO_GEOMETRY;
```

Př.5: Formát funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES.

Parametr **geometry** obsahuje geometrický popis prvku, u něhož chceme odstranit případné duplicity. Parametr **tolerance** je zde ve smyslu popsáno v kapitole 3.3.5.

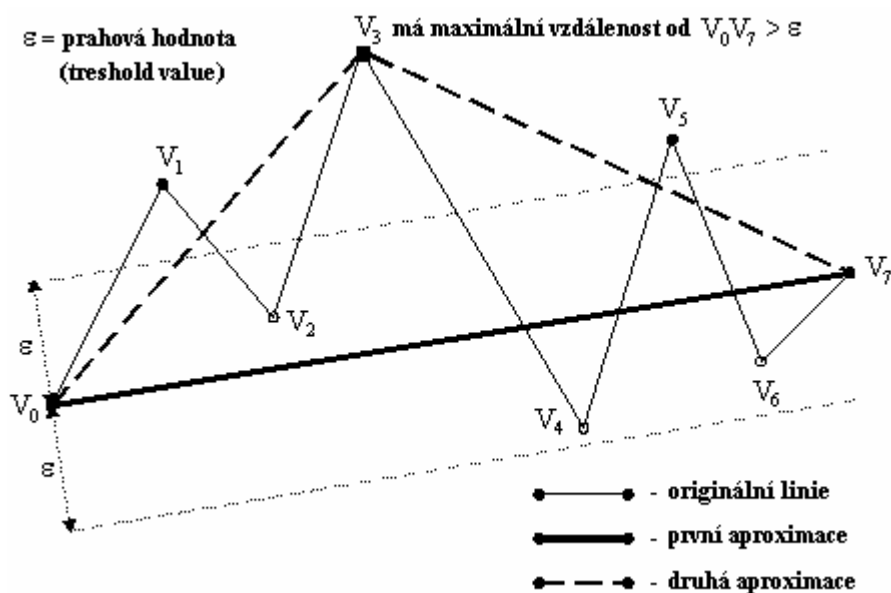
Vhodnou volbou atributu **tolerance** lze dosáhnout toho, že se tato funkce bude chovat de facto jako generalizační. Při provádění funkce dojde k odstranění všech bodů od počátečního bodu, které jsou ve vzdálenosti specifikované atributem **tolerance**. Obrázek 4 graficky znázorňuje chování funkce **SDO_UTIL.REMOVE_DUPLICATE_VERTICES** pro zvolenou hodnotu atributu **tolerance**.



Obr. 4: Chování funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES.

3.7.2 SDO_UTIL.SIMPLIFY

Cílem funkce SDO_UTIL.SIMPLIFY je redukce počtu vrcholů ve vstupním geometrickém popisu prostorového prvku pomocí realizace Douglas-Peuckerova algoritmu. Detailní popis algoritmu lze nalézt v [8]. Základní princip je znázorněn na obrázku 5.



Obr.5: Princip Douglas-Peuckerova algoritmu.

Formát funkce SDO_UTIL.SIMPLIFY je následující:

```
SDO_UTIL.SIMPLIFY(  
  geometry      IN SDO_GEOMETRY,  
  threshold     IN NUMBER  
) RETURN SDO_GEOMETRY;
```

Př.6: Formát funkce SDO_UTIL.SIMPLIFY.

Parametr `geometry` obsahuje geometrický popis prvku, jenž má být zjednodušen. Druhý parametr - `threshold`, je prahová hodnota, ovlivňující průběh realizace generalizačního algoritmu. Tvar výsledného prvku tedy závisí kromě původního geometrického popisu také na počáteční volbě hodnoty atributu `threshold`, což by měla být kladná hodnota. Nulová hodnota by znamenala vrácení původního geometrického popisu prvku. Jednotka atributu `threshold` závisí na souřadnicovém systému, ve kterém je prvek vyjádřen.

Pokud bude nevhodně zvolena prahová hodnota, provedením funkce SDO_UTIL.SIMPLIFY může dojít k redukci polygonu na linii a linie na bod.

Zcela zásadní je však skutečnost, že provedením této funkce se nemusí zachovat topologické charakteristiky, které se vyskytují mezi původními, nezjednodušenými geometrickými popisy prvků.

3.8 Balík SDO_SAM

Balík SDO_SAM obsahuje podprogramy pro provádění prostorových analýz a data mining²¹.

Předmětem mého zájmu byly z tohoto balíku dva podprogramy, které se nechají využít ke generalizaci prostorových prvků - **SDO_SAM.SIMPLIFY_GEOMETRY** a **SDO_SAM.SIMPLIFY_LAYER**.

3.8.1 SDO_SAM.SIMPLIFY_GEOMETRY

Účelem podprogramu SDO_SAM.SIMPLIFY_GEOMETRY je zjednodušit geometrický popis prvku. Přitom se porovnávají plochy původní geometrické reprezentace prvku a plocha nově vzniklého geometrického popisu prvku. Tento poměr by neměl překročit stanovenou hodnotu.

²¹ Data mining - hledání vzájemných vztahů mezi velkým množstvím polí v databázi.

Formát tohoto podprogramu je:

```
SDO_SAM.SIMPLIFY_GEOMETRY(  
  geom      IN SDO_GEOMETRY,  
  tol       IN NUMBER,  
  pct_area_change_limit IN NUMBER DEFAULT 2  
) RETURN SDO_GEOMETRY;
```

Př.7: Formát funkce SDO_SAM.SIMPLIFY_GEOMETRY.

Parametr `geom` obsahuje geometrický popis prvku, jenž má být zjednodušen. Parametr `tol` je zde ve smyslu tolerance popsaného v kapitole 3.3.5. Poslední parametr – `pct_area_change_limit`, se udává v procentech a vyjadřuje, o kolik se mohou od sebe lišit plocha původní geometrické reprezentace prvku a plocha nově vzniklého geometrického popisu v porovnání s plochou, kterou uzavírá původní geometrický popis. Ten představuje návratovou hodnotu podprogramu. Výsledná plocha může být větší či menší, ale vždy do poměru, určeného atributem `pct_area_change_limit`, jehož implicitní hodnota je rovna **2**.

Funkce SDO_SAM.SIMPLIFY_GEOMETRY redukuje počet vrcholů v geometrickém popisu prvku za použití interního volání funkce SDO_UTIL.SIMPLIFY s „přiměřenou“ prahovou hodnotou. Bohužel, z dostupných materiálů není možné získat jakýkoli bližší popis například toho, jak a zda počet vrcholů ve vstupním geometrickém popisu prvku souvisí s vnitřní volbou prahové hodnoty pro provedení funkce SDO_UTIL.SIMPLIFY. Jediné, co lze ovlivnit, je plocha výsledného geometrického popisu prvku určením hodnoty parametru `pct_area_change_limit`.

3.8.2 SDO_SAM.SIMPLIFY_LAYER

Použití této procedury je vhodné v okamžiku, kdy budeme chtít generalizovat celou vrstvu (sloupec typu `SDO_GEOMETRY`) jedním voláním generalizační procedury.

Formát procedury SDO_SAM.SIMPLIFY_LAYER je:

```
SDO_SAM.SIMPLIFY_LAYER(  
  theme_tablename      IN VARCHAR2,  
  theme_colname        IN VARCHAR2,  
  simpl_geom_colname   IN VARCHAR2,  
  commit_interval     IN NUMBER DEFAULT 10,  
  pct_area_change_limit IN NUMBER DEFAULT 2);
```

Př.8: Formát procedury SDO_SAM.SIMPLIFY_LAYER.

Parametr `theme_tablename` obsahuje název tabulky, obsahující sloupec typu `SDO_GEOMETRY`, v němž obsažené prvky (geometrické

popisy prostorových prvků) mají být zjednodušeny. Název tohoto sloupce je uveden v atributu `theme_colname`. Parametr `smpl_geom_colname` obsahuje název sloupce typu `SDO_GEOMETRY`, do kterého jsou umístěny zjednodušené popisy prvků. Parametr `commit_interval` umožňuje nastavit mez pro provedení vnitřní operace `commit`²². Jeho výchozí hodnota je 10. Posledním parametrem této procedury je parametr `pct_area_change_limit` a udává, o kolik se mohou od sebe lišit plocha původní geometrické reprezentace prvku a plocha nově vzniklého geometrického popisu v porovnání s plochou, kterou uzavírá původní geometrický popis. Udává se v procentech a pokud není zadáno jinak, jeho implicitní hodnota je **2**.

Procedura `SDO_SAM.SIMPLIFY_LAYER` zjednoduší všechny geometrické popisy prvků v příslušném sloupci. Při provádění této procedury dochází de facto k aplikování funkce `SDO_SAM.SIMPLIFY_GEOMETRY` na každý takový popis s tím rozdílem, že dojde k uložení generalizovaného popisu přímo do příslušného sloupce zdrojové tabulky.

²² `commit` - potvrzení platnosti transakce

4 Generování přehledové mapy z dat ISKN

Hlavním datovým zdrojem pro mou práci byl Informační systém katastru nemovitostí, spravovaný Českým úřadem zeměměřickým a katastrálním. Data z tohoto systému jsou uživateli poskytována v novém výměnném formátu, který je podrobně popsán v [10].

4.1 Výměnný formát ISKN

Výměnný formát je určen k vzájemnému předávání dat mezi systémem ISKN a jinými systémy zpracování dat. Jedná se o textový soubor s kódováním češtiny dle ČSN ISO 8859-2 (ISO Latin2).

Datový soubor výměnného formátu je textový soubor skládající se z:

- hlavičky,
- datových bloků,
- koncového znaku &K.

Hlavička výměnného formátu (VF) obsahuje následující informace:

- verzi VF,
- datum a čas vytvoření souboru
- původ dat,
- označení kódové stránky,
- seznam skupin datových bloků souboru,
- jméno osoby, která soubor vytvořila,
- časovou podmínku použitou pro vytvoření souboru,
- omezující podmínku - katastrální území,
- omezující podmínku - oprávněné subjekty,
- omezující podmínku - parcely,
- omezující podmínku - polygon.

ádek hlavičky výměnného formátu začíná návěštím &H, následuje označení položky např. SKUPINA (seznam skupin datových bloků souboru) a příslušné údaje oddělené středníkem.

Datové bloky výměnného formátu náleží do některé nadřazené skupiny a obsahují následující řádky:

- právě jeden uvozující řádek obsahující seznam atributů s jejich datovými typy (v požadovaném pořadí),
- řádky obsahující vlastní data (ve stanoveném pořadí).

Uvozující řádek bloku začíná návěštím &B (blok výměnného formátu), následuje zkratka označení bloku, např. PAR (parcela) a jednotlivé atributy oddělené středníkem.

Atribut obsahuje název a typ položky (číselné a textové položky s upřesněním délky). Přípustné typy položek jsou číselná (N), textová (T) a datumová (D).

Řádek obsahující vlastní data začíná návěštím &D (data výměnného formátu), následuje zkratka označení bloku např. PAR (parcela) a vlastní data oddělená středníkem. Pro každý objekt se vytvoří jeden řádek.

4.2 Rekonstrukce logického datového modelu z popisu výměnného formátu ISKN

Využití dat ze souboru výměnného formátu (VF) by nebylo možné bez implementace určitého logického datového modelu a naplnění této struktury daty výměnného formátu.

Bohužel, logický datový model Informačního systému katastru nemovitostí není věc veřejná a i pro účely této práce byl ve své kompletní podobě s návrhem všech relací mezi daty nedostupný.

Bylo proto nutné navrhnout buď vlastní logický datový model uložení dat, nebo se pokusit část logického modelu zrekonstruovat z popisu výměnného formátu. Návrh logického datového modelu není a ani neměl být jedním z bodů této práce, proto jsem se rozhodl jít cestou rekonstrukce modelu z popisu VF.

Jedním z nejdůležitějších kroků rekonstrukce byl výběr potřebných datových bloků. Předmětem mého zájmu z celého informačního systému katastru nemovitostí byly především datové bloky spadající do skupiny nazvané Prvky katastrální mapy (PKMP). Tato skupina obsahuje 11 datových bloků pro uložení prvků katastrální mapy. Seznam bloků je uveden v tabulce 2.

Kód	Název tabulky v popise výměnného formátu	Popis
SOBR	BODY_POLOHOPISU	Souřadnice obrazů bodů polohopisu v mapě.
SBP	SPOJENI_B_POLOH	Spojení bodů polohopisu – definuje polohopisné liniové prvky.
SBM	DPM_SDOGEOM	Spojení bodů mapy – definuje nepolohopisné liniové prvky.
KODCHB	KODY_CHAR_Q_BODU	Číselník kódů charakteristiky kvality bodu.
TYPSOS	T_SOURAD_SYS	Číselník typů souřadnicových systémů.
HP	HRANICE_PARCEL	Hranice parcel.
OP	OBRAZY_PARCEL	Obrazy parcel (parcelní číslo, značka druhu pozemku,...).
OB	OBRAZY_BUDOV	Obrazy budov (obvod budovy, značka druhu budovy).
DPM	DALSI_Prvky_MAPY	Další prvky mapy.
OBBP	OBRAZY_BODU_BP	Obrazy bodů BP.
TYPPPD	T_Prvku_P_DAT	Číselník typů prvků prostorových dat.

Tab.2: Datové bloky ve skupině PKMP.

Při rekonstrukci modelu jsem vycházel z cílů této práce a do jeho návrhu nezahrnul některé tabulky ze skupiny PKMP, které nebyly nezbytně nutné, konkrétně tabulky DPM_SDOGEOM, T_SOURAD_SYS, OBRAZY_BUDOV a OBRAZY_BODU_BP.

Pro generování přehledové mapy bylo naopak nutné při návrhu modelu uvažovat tabulky i z jiných bloků, než PKMP. Jednalo se o vybrané tabulky ze skupiny Nemovitosti (NEMO), uvedené v tabulce 3.

Kód	Název tabulky v popise výměnného formátu	Popis
PAR	PARCELY	Parcely
KATUZE	KATASTR_UZEMI	Číselník katastrálních území
OBCE	OBCE	Číselník obcí
OKRESY	OKRESY	Číselník okresů
KRAJE	KRAJE	Číselník krajů

Tab.3: Vybrané bloky ze skupiny NEMO.

Blok PARCELA spojuje část modelu pro uložení grafických dat s číselníky správních jednotek. Uvažováním a zařazením vybraných bloků ze skupiny NEMO do logického modelu lze generovat přehledovou mapu katastrálních území pro jednotlivou obec, okres či kraj. Popis a účel jednotlivých bloků a příslušných atributů lze nalézt v [10].

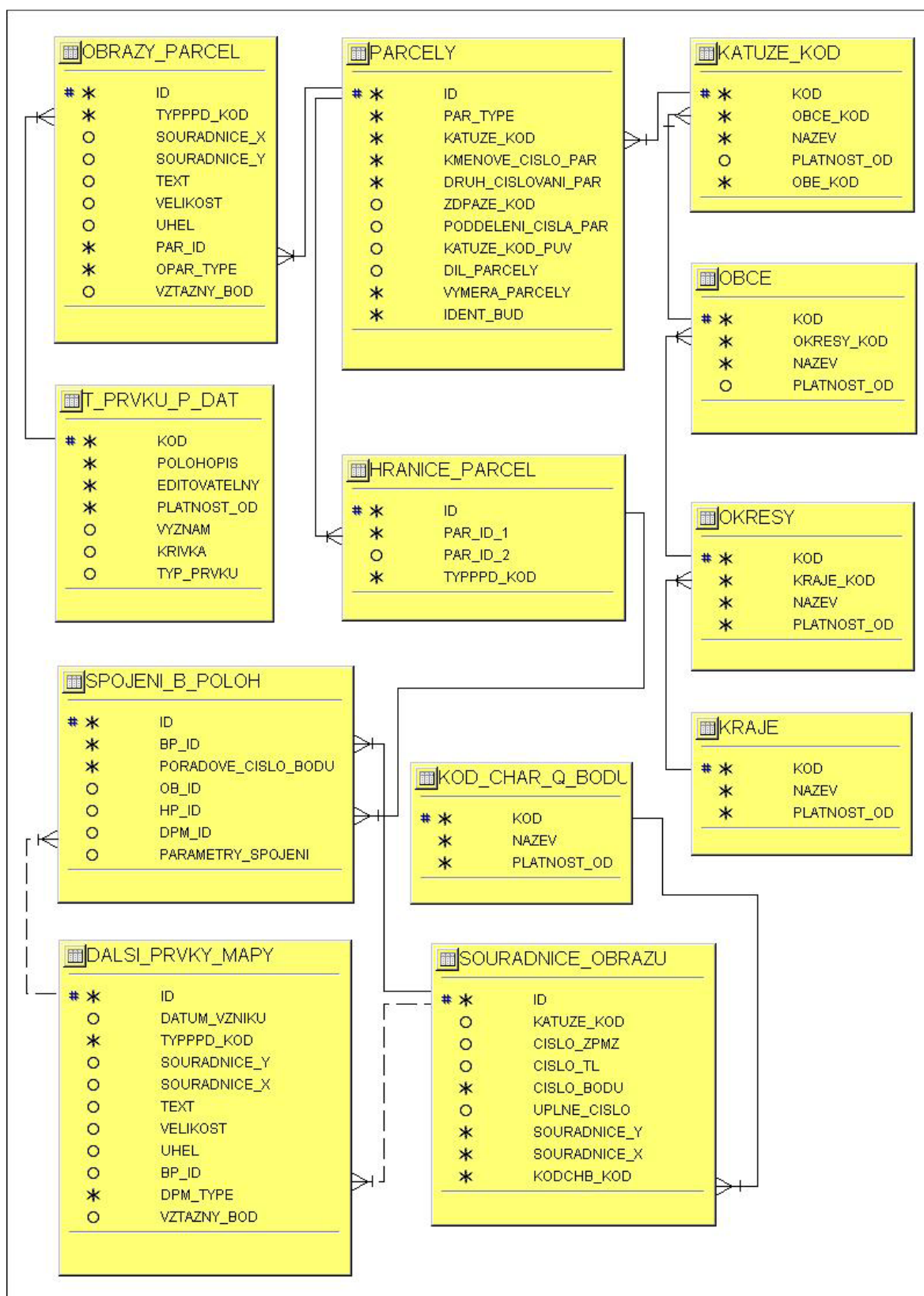
Při navrhování vazeb mezi tabulkami jsem využil popisu výměnného formátu a dle tohoto jednotlivé vazby v modelu realizoval.

Pokud byl v použitém bloku atribut, představující cizí klíč do jiného bloku, který v modelu nebyl uvažován, byl tento atribut sice naplněn daty, ale při fyzické realizaci modelu se s ním operovalo jako s neklíčovým atributem.

Poslední otázkou tvorby zrekonstruovaného modelu byla volba atributů jednotlivých bloků. U některých bloků jsem zachoval všechny atributy tak, jak jsou uvedeny v popisu VF. To bylo v případě, kdy byly všechny atributy povinné a nebylo je podle popisu VF možné vypustit. Samozřejmě, pro naplnění záměru této práce by šly vynechat i tyto, nicméně jsem se snažil v takových případech povinné atributy v zrekonstruovaném logickém modelu zachovat. V ostatních případech jsem do logického modelu vždy zahrnul atribut či atributy daného bloku, které byly podle popisu VF povinné a atributy vhodné pro vyhodnocování specifických dotazů.

Výsledný návrh logického modelu, který odpovídá zároveň i modelu fyzickému, je znázorněn na obrázku 6. V každém bloku je vyznačen primární klíč bloku, seznam všech použitých atributů, včetně grafického znázornění povinnosti jejich naplnění daty. Na obrázku 6 jsou rovněž znázorněny vazby mezi jednotlivými bloky. Grafická podoba jmenovaných prvků modelu odpovídá konvencím prostředí Oracle Designer, ve kterém byl model vytvořen.

Takto vytvořený logický model jsem fyzicky implementoval na platformě Oracle. Při fyzické implementaci jsem ve většině případů použil název bloku a jeho atributů tak, jak je uveden v popisu VF. Pro odstranění problémů s importem dat do databáze byly voleny datové typy a rozsahy atributů ve shodě s popisem VF.



Obr.6: Zrekonstruovaný logický datový model pro uložení grafických dat.

4.3 Import dat do fyzického modelu

Úspěšná fyzická realizace logického modelu znamenala možnost importovat data se souboru výměnného formátu do databázových tabulek.

K dispozici jsem měl soubor výměnného formátu s daty pěti sousedních katastrálních území. Pro import vybraných dat z tohoto souboru do příslušných databázových tabulek jsem v jazyce Pascal realizoval program ImportVF.exe, jehož činnost se dá popsat následujícími body:

- 1) Otevření souboru výměnného formátu (ExportKT.vfk) pro čtení.
- 2) Postupné čtení souboru a v případě, že byl nalezen uvozující řádek bloku, který jsem uvažoval při tvorbě fyzického modelu, uložení vybraných dat z tohoto bloku v následující řádce do pomocného souboru. Pro každý blok, který je reprezentován v databázi tabulkou, byl vytvořen samostatný pomocný textový soubor, otevřený na začátku programu pro zápis.
- 3) Po zpracování vstupního souboru, to znamená přečtení celého souboru výměnného formátu, dojde k otevření souboru ImportVF.sql, do něhož jsou nakopírována data z pomocných souborů.

Při ukládání dat do pomocných souborů, stejně tak jako při jejich kopírování do souboru ImportVF.sql, byla data ukládána ve formě SQL příkazu **INSERT**. Struktura tohoto příkazu je patrná z následující ukázky:

```
INSERT INTO katuze VALUES (  
629511,556394, 'Vacovy', '19.12.1997 00:00:00');
```

Př. 9: Struktura příkazu **INSERT**.

Při kopírování dat z jednotlivých pomocných souborů do výsledného souboru ImportVF.sql bylo nutné dodržet správné pořadí kopírovaných záznamů. Při importu dat do databáze bylo totiž nutné ohlídat, aby se dříve vložil záznam, který neobsahuje vazbu na doposud nevložený údaj, čili dodržení referenční integrity dat. To přesně znamená, že žádný řádek v cizí tabulce nesmí obsahovat takovou hodnotu cizího klíče, která nemá odpovídající záznam v primární tabulce. Z tohoto důvodu jsem si zavedl pomocné soubory, jeden pro každý zájmový blok, jejichž obsah jsem na konci programu ImportVF.exe podle předem stanoveného pořadí zkopíroval do výsledného souboru ImportVF.sql, sestávajícího ze sekvence SQL **INSERT** příkazů.

Data příslušných bloků byla do souboru ImportVF.sql vložena v pořadí bloků – SOBR, KRAJE, OKRESY, OBCE, KATUZE, TYPPPD, PARCELY, OP, HP, DPM, SBP. Tabulka KOD_CHAR_Q_BODU, obsahující

kódy jednotlivých charakteristik kvality bodu, byla naplněna manuálně z důvodu toho, že se jedná o tabulku, jejíž obsah je v různých souborech výměnného formátu konstantní.

Celkem bylo do navržené databáze importováno přes sto tisíc záznamů. Import trval přibližně čtyři hodiny. Zdrojový kód programu ImportVF.exe je uložen na přiloženém CD k této diplomové práci v souboru ImportVF.pas.

4.4 Generování úseků katastrální hranice

Problém generování katastrální hranice určitého katastrálního území byl jedním z klíčových problémů, řešených v této diplomové práci. Bylo nutné se seznámit s jednotlivými bloky a obsaženými daty v nich po importu dat ze souboru výměnného formátu. Import dat proběhl v pořádku, všechny atributy byly naplněny tak, jak bylo očekáváno při vytváření logického datového modelu.

Existuje zřejmě více než jeden způsob řešení otázky generování katastrální hranice ze současného relačního datového modelu. Mělo by být možné jít cestou přes atributy, které by svým primárním účelem měli k řešení této a podobných otázek sloužit. Jinou variantou je využít logiky uložení a popisu dat.

Při výběru jedné z možných variant je nutné si uvědomit pár skutečností, se kterými je potřeba při vytváření výsledného databázového dotazu počítat. Pominutí těchto skutečností by mohlo vést sice k jednodušším, ale nesprávným řešením. V dalším textu se pokusím tyto skutečnosti nastínit, včetně popisu jejich řešení.

Pokud mám v databázi k dispozici data pro více sousedních katastrálních území, musím ohlídat, abych vybral ve svých dotazech směřujících k jednomu konkrétnímu území z tabulek data právě pro toto území a aby ta data jako například při výběru úseků katastrální hranice byla kompletní. Každé katastrální území má svůj jednoznačný identifikátor. V modelu který jsem implementoval, obsahuje tento identifikátor několik tabulek. Zdrojovou tabulkou, obsahující kódy všech katastrálních území, pro něž jsou data v databázi uložena, je tabulka KATUZE. Seznam katastrálních území včetně jejich jednoznačného identifikátoru (*kod*), pro něž jsem měl k dispozici potřebná data, je uveden v tabulce 4.

KÓD	NÁZEV
629481	Dolní Lhota u Klatov
629499	Lomec u Klatov
629502	Novákovice
629511	Vacovy
772313	Týnec u Janovic nad Úhlavou

Tab. 4: Seznam katastrálních území uložených v databázi.

Relační logický model ze své podstaty neobsahuje uložení jednotlivých objektů na jednom místě. Tyto objekty, jakým může být například katastrální hranice, je nutné sestavit využitím vhodných dotazů, aplikovaných na odpovídající tabulky. Z tohoto důvodu je nutné vybrat jednotlivé úseky katastrální hranice konkrétního území ze všech úseků hranic, které jsou v databázi uloženy. Podobně jako katastrální území, i každý hraniční úsek je v databázi uložen pod svým jednoznačným identifikátorem. Zdrojovou tabulkou, obsahující seznam všech identifikátorů úseků hranic je tabulka **HRANICE_PARCEL**. Každý uložený úsek v této tabulce představuje úsek mezi dvěma sousedními parcelami, jejichž identifikátory jsou uloženy spolu s identifikátorem hraničního úseku v jedné tabulce. Přes identifikátor jedné z parcel, konkrétně `par_id_1`, lze spojit tabulku **PARCELY** s tabulkou **HRANICE_PARCEL**. Hraniční úsek je definován právě dvěma body – počátečním a koncovým bodem úseku.

Významným atributem, uloženým v tabulce **HRANICE_PARCEL**, je atribut `typppd_kod`. Tento atribut je cizím klíčem do tabulky **T_PRVKU_P_DAT**, to znamená, podle hodnoty v tomto atributu se nechá identifikovat, o jaký typ prostorového prvku jde. V mém případě mě zajímaly hodnoty, které odpovídají katastrální hranici. Jedná se o hodnoty z intervalu **20600-20655**.

Mohlo by se zdát, že pokud není problém vybrat při znalosti kódu katastrálního území všechny parcely v tomto území a zároveň všechny hraniční úseky, které jsou uloženy s kódem odpovídajícím kódu katastrální hranice, nebude s generováním katastrální hranice nějaký větší problém.

První skutečností, kterou je ale třeba mít na vědomí, je fakt, že pokud mám dvě katastrální území, která mají část hranice společnou, bude (měl by být) tento úsek či množina úseků uložených v databázi pouze jednou, to znamená, každý tento společný úsek bude jednoznačně určitelný podle svého identifikátoru. To ale v relačním způsobu uložení dat tak, jak je nyní navrženo, znamená, že je tento úsek dostupný pouze přes identifikátor jednoho ze dvou sousedních katastrálních území.

Druhým faktem, který je třeba při tvorbě dotazu, který by měl vybrat všechny úseky katastrální hranice, uvažovat, je hierarchický způsob uložení hranic územně správních jednotek. Při označování hraničních úseků kódem typu prostorového prvku platí nadřazenost státní hranice nad hranicí kraje, hranice kraje nad hranicí okresu, hranice okresu nad hranicí obce a hranice obce nad katastrální hranicí. Z toho vyplývá, že pokud budu mít katastrální území, jehož hranice bude například jen z části identická s hranicí obce, pak úseky tvořící tuto část budou uloženy pod odpovídajícím kódem mimo interval 20600 až 20655.

Uvedené skutečnosti mě vedly k tomu, že jsem problém se získáním katastrální hranice z aktuálního způsobu uložení dat řešil postupem, při kterém jsem využil spíše logiky uložení a popisu dat, než

atributů, o kterých si myslím, že by měly primárně otázku dotazování se na katastrální hranici řešit – **katuze_kod**, **typppd_kod**.

Mé řešení spočívá ve sjednocení dvou dotazů do databáze. Možná triviální, ale podstatný předpoklad je, že jsou grafická data vyjádřena ve shodném souřadnicovém systému (v mém případě S-JTSK) a zkonstruovány ve stejném druhu mapy (DKM, KMD). U obou dotazů jsem použil tabulky **HRANICE_PARCEL**, **PARCELY**, **SBP**, **SOBR** a vzájemného spojení přes příslušné klíčové atributy.

V prvním dotazu jsem vybral všechny úseky z **HRANICE_PARCEL**, u kterých parcela zastoupená atributem **par_id_1** náleží mezi parcely s **katuze_kod** rovno kódu příslušného kat. území a zároveň parcela zastoupená atributem **par_id_2** do tohoto území nenáleží, nebo má hodnotu **NULL** (pro případ, že by sousední katastrální území nebylo digitalizováno).

V druhém dotazu jsem vybral úseky hranic z **HRANICE_PARCEL**, u kterých parcela zastoupená atributem **par_id_1** nespadá mezi parcely v daném kat. území a zároveň parcela zastoupená atributem **par_id_2** do tohoto území spadá.

U obou dotazů jsem si jako výsledek vygeneroval identifikátor hraničního úseku a souřadnice odpovídající počáteční a koncového bodu.

Sjednocením dotazů jsem tak dostal identifikátory všech úseků katastrální hranice a seznam souřadnic bodů na hranici. Tato data jsem si uložil do textového souboru **bodyNaHr.txt** a dále programově zpracoval tak, abych dostal úseky seřazené za sebou podle skutečného průběhu katastrální hranice.

SQL zápis celého dotazu vypadá následovně (jsou vybrány a seřazeny podle svého identifikátoru úseky tvořící hranici katastrálního území Lomec u Klatov):

```
SELECT hp.id AS usek_hranice,SOBR.SOURADNICE_X, SOBR.SOURADNICE_Y
FROM HRANICE_PARCEL hp, PARCELY p, SBP, SOBR
WHERE p.id = hp.par_id_1 AND
      hp.id = SBP.HP_ID AND
      SBP.BP_ID = SOBR.ID AND
      p.katuze_kod = 629499 AND
      (hp.par_id_2 NOT IN (SELECT id
                          FROM PARCELY
                          WHERE katuze_kod = 629499) OR
      hp.par_id_2 IS NULL)
UNION
SELECT hp.id AS usek_hranice,SOBR.SOURADNICE_X, SOBR.SOURADNICE_Y
FROM HRANICE_PARCEL hp, PARCELY p, SBP, SOBR
WHERE p.id = hp.par_id_1 AND
      hp.id = SBP.HP_ID AND
      SBP.BP_ID = SOBR.ID AND
      (hp.par_id_1 NOT IN (SELECT id
                          FROM PARCELY
                          WHERE katuze_kod = 629499))AND
      hp.par_id_2 IN      (SELECT id
                          FROM PARCELY
                          WHERE katuze_kod = 629499))
ORDER BY (usek_hranice);
```

Př.10: SQL zápis dotazu na vybrání úseků a bodů katastrální hranice.

4.5 Možné rozšíření fyzického datového modelu vytvořením tabulky KATASTRALNI_HRANICE

Jak již bylo uvedeno, Oracle Spatial podporuje objektově-relační způsob uložení dat. Případ katastrální hranice může být vhodnou ukázkou využití takového přístupu. Uložení hranice jako objektu s sebou přináší nespornou výhodu při generování přehledové mapy katastrálních území, kdy by z jedné tabulky stačilo jednoduchým dotazem vybrat požadované katastrální hranice. Vytvoření, naplnění a připojení takové tabulky do vhodného datového modelu by umožnilo rychle a snadno vytvářet přehledové mapy v různých územně správních úrovních. Ve své práci jsem navrhl tabulku **KATASTRALNI_HRANICE** a její strukturu naplnil daty tak, aby mohla sloužit ke snadnému generování přehledové mapy katastrálních území.

4.5.1 Definice tabulky KATASTRALNI_HRANICE

Hlavním účelem tabulky **KATASTRALNI_HRANICE** je uložení celého průběhu katastrální hranice konkrétního území v jedné tabulce a využít tak možnosti, kterou pro tyto účely nabízí prostředí Spatial pomocí

svého objektového typu *SDO_GEOMETRY*. Pomocí něj je možné katastrální hranici popsat jako polygon a souřadnice vrcholů takového polygonu uložit jako společnou množinu (pole) souřadnic v atributu **sdo_ordinates_array**. Z tohoto důvodu tabulka **KATASTRALNI_HRANICE** obsahuje atribut *hranice*, který je objektového datového typu *SDO_GEOMETRY* a obsahuje vlastní geometrický popis katastrální hranice. Další atributy tabulky **KATASTRALNI_HRANICE** jsou patrné z definice:

```
CREATE TABLE KATASTRALNI_HRANICE (  
id NUMBER PRIMARY KEY,  
katuze_kod NUMBER (6),  
hranice SDO_GEOMETRY,  
definicni_bod SDO_GEOMETRY  
);
```

Př. 11: Definice tabulky **KATASTRALNI_HRANICE**.

Atribut *id* je jednoznačným identifikátorem katastrální hranice. Tím je i atribut *katuze_kod*, obsahující kód katastrálního území, jehož hranice je v daném záznamu uložena. Hlavním důvodem výskytu tohoto atributu je ale možnost rozšíření implementovaného fyzického relačního modelu o tabulku **KATASTRALNI_HRANICE**. Atribut *katuze_kod* je možným cizím klíčem do tabulky **KATUZE**. Rozsah atributu *katuze_kod* byl nastaven jako *NUMBER(6)* z důvodu zachování doménové integrity, čili stejného datového typu a rozsahu atributů, přes které by probíhala fyzická realizace vazby mezi tabulkami (entitami).

Posledním definovaným atributem tabulky **KATASTRALNI_HRANICE** je atribut *definicni_bod* objektového typu *SDO_GEOMETRY*. Význam atributu *definicni_bod* je patrný z jeho pojmenování. Hodnota v něm uložená představuje vypočtený centroid (těžiště) katastrálního území ze souřadnic vrcholů polygonu, reprezentujícího katastrální hranici. Při výpočtu centroidu je katastrální území považováno za homogenní oblast, proto by se mohlo stát, že při jistém „nekonvexním“ tvaru katastrální hranice může dojít k vypočtení centroidu, který bude ležet mimo katastrální území. Pro účely mé práce není výpočet centroidu katastrálního území a jeho poloha vůči němu nijak zásadní, spíše jsem se snažil otestovat funkci *SDO_CENTROID*, která je pro výpočet centroidu plochy v prostředí Spatial standardně k dispozici. Návrátová hodnota funkce *SDO_CENTROID* je objekt typu *SDO_GEOMETRY*. Z tohoto důvodu byl zvolen datový typ atributu *definicni_bod* *SDO_GEOMETRY*, aby mohlo dojít k přímému ukládání vypočtených hodnot do tohoto atributu.

4.5.2 Metadata o sloupcích tabulky KATASTRALNI_HRANICE

Jak vyplývá z kapitoly 3.5, je nutné po vytvoření tabulky obsahující sloupec typu *SDO_GEOMETRY* definovat metadata o tomto sloupci (vrstvě) a vložit je do příslušného pohledu.

Metadata o sloupci *hranice* jsem vložil do pohledu *USER_SDO_GEOM_METADATA* následující SQL sekvencí:

```
INSERT INTO USER_SDO_GEOM_METADATA
VALUES (
    'katastralni_hranice',
    'hranice',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', 0, 1250000, 0.005),
        SDO_DIM_ELEMENT('Y', 0, 950000, 0.005)
    ),
    NULL
);
```

Př. 12: Metadata o sloupci *hranice*.

Z uvedeného kódu je vidět, že pole *sdo_dim_array* obsahuje dvě pole *sdo_dim_element*, což značí, že uvažuji dvojdimenzionální geometrický popis katastrální hranice. Rozsah nabývaných hodnot na ose X byl omezen shora hodnotou **1250000**, na ose Y pak hodnotou **950000**.

Hodnota atributu *srid* je rovna **NULL**, což znamená, že s daty není asociován žádný souřadnicový systém definovaný v tabulce *MDSYS.CS_SRS*. Spatial ve verzi, jakou jsem měl v okamžiku tvorby své diplomové práce k dispozici, implicitně nepodporuje souřadnicový systém S-JTSK, ve kterém jsem měl geometrická data vyjádřena. Z tohoto důvodu jsem pro atribut *srid* volil hodnotu **NULL**. Spatial se pak při manipulaci s takovými daty chová, jako by byla vyjádřena v rovinném pravoúhlém souřadnicovém systému, což je v souvislosti s S-JTSK do jisté míry vyhovující. Jak se později ukáže, jistým problémem je především jiná orientace souřadnicových os rovinného systému, který interně využívá Spatial, od orientace souřadnicových os systému S-JTSK.

Do pohledu *USER_SDO_GEOM_METADATA* jsem vložil rovněž metadata pro sloupec *definicni_bod*, který je také stejně jako sloupec *hranice* objektového typu *SDO_GEOMETRY*.

SQL zápis vložení metadat o sloupci `definicni_bod` se od definice metadat o sloupci `hranice` příliš neliší:

```
INSERT INTO USER_SDO_GEOM_METADATA
VALUES (
    'katastralni_hranice',
    'definicni_bod',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', 0, 1250000, 0.005),
        SDO_DIM_ELEMENT('Y', 0, 950000, 0.005)
    ),
    NULL
);
```

Př. 13: Metadata o sloupci `definicni_bod`.

4.6 Příprava importu dat do tabulky

KATASTRALNI_HRANICE

V podobě tabulky `KATASTRALNI_HRANICE` mám již připravenou strukturu pro uložení dat popisujících katastrální hranici, včetně identifikátoru příslušného katastrálního území a vypočteného centroidu.

4.6.1 Formát importovaných dat

Data do tabulky `KATASTRALNI_HRANICE` jsem vkládal opět pomocí standardního příkazu jazyka SQL `INSERT`. Tomu, abych mohl tento postup využít, předcházelo několik velmi důležitých kroků. Především jde o vkládání údajů do sloupce `hranice` objektové typu `SDO_GEOMETRY`. Je důležité porozumět konceptu ukládání dat do objektu tohoto typu, aby je uměl Spatial zpracovávat korektním způsobem.

Jak bylo uvedeno v kapitole 3.4, objektový typ `SDO_GEOMETRY` slouží k uložení geometrického popisu prostorového prvku. Z definice typu `SDO_GEOMETRY` vyplývá, že k tomuto popisu používá celkem pět atributů. Správné naplnění těchto atributů je základním předpokladem pro další práci s prvkem typu `SDO_GEOMETRY` v Spatial. Konkrétní hodnoty atributů vychází z požadavků a definic prostředí Spatial pro správný popis konkrétních geometrických dat.

4.6.1.1 Atribut `sdo_gtype`

Atribut `sdo_gtype` označuje geometrický typ prvku a lze ho obecně vyjádřit ve tvaru `dltt`. Hodnotu tohoto atributu jsem volil **2003**. Číslice **2** na pozici „d“ indikuje, že je objekt popsán ve dvou dimenzích. Jelikož

jsem při geometrickém popisu katastrální hranice nepoužil lineární odkazování²³, je na pozici „l“ hodnota **0**. Katastrální hranici uvažuji jako polygon, přičemž předpokládám, že v každém záznamu v tabulce **KATASTRALNI_HRANICE** budu mít ve sloupci *hranice* uložen geometrický popis pouze jedné hranice. V takovém případě Spatial vyžaduje, aby hodnota na pozici „tt“ byla **03**.

4.6.1.2 Atribut sdo_srid

Atribut **sdo_srid** slouží k identifikaci souřadnicového systému, ve kterém je prostorový prvek geometricky popsán. Jestliže je hodnota atributu **sdo_srid** **NULL**, znamená to jako například v tomto případě, že s daty není asociován žádný souřadnicový systém, definovaný v tabulce **MDSYS.CS_SRS**.

4.6.1.3 Atribut sdo_point

Atributu **sdo_point** jsem přiřadil hodnotu **NULL**. Do atributu **sdo_point** by se měly ukládat konkrétní hodnoty souřadnic v případě, že by byl prostorový prvek v databázi reprezentován bodovým prvkem.

4.6.1.4 Atribut sdo_elem_info

Atribut **sdo_elem_info** v tomto případě sestává z jedné množiny trojice hodnot, které popisují způsob interpretace souřadnic uložených v atributu **SDO_ORDINATES**.

První hodnota z této množiny značí pozici uložení první souřadnice v poli **SDO_ORDINATES** počátečního elementu geometrického popisu prvku. Druhá hodnota indikuje typ elementu. V mém případě se jedná o polygon, jehož vrcholy jsou spojeny přímými liniovými segmenty, což přímo ovlivňuje třetí hodnotu z interpretační množiny. V takovém případě tato hodnota udává, jak interpretovat pořadí souřadnic elementu v poli **SDO_ORDINATES**.

V závislosti na výše uvedeném jsem atribut **sdo_elem_info** naplnil hodnotou **(1,1003,1)**. Jelikož popis katastrální hranice geometricky reprezentují jedním polygonem, je tento polygon uvažován jako „vnější“, z čehož vyplývá pro korektní geometrický popis nutnost uvedení souřadnic vrcholových bodů polygonu v protisměru chodu hodinových ručiček.

²³ Lineární odkazování (Linear Referencing) je vhodný prostředek k asociování událostí k částem geometrického popisu liniového prvku.

4.6.1.5 Atribut `sdo_ordinates`

Atribut `sdo_ordinates` obsahuje souřadnice bodů použitých ke geometrickému popisu objektu. Pořadí souřadnic musí být shodné s pořadím dimenzí v definici odpovídajících metadat. Pořadí dimenzí jsem zvolil X, Y, proto je nutné ukládat souřadnice do atributu `sdo_ordinates` v pořadí X, Y.

Původní záměr při navrhování tabulky `KATASTRALNI_HRANICE` byl ten, že by v poli `sdo_ordinates` objektu `hranice` byly uloženy odkazy na jednotlivé vrcholové body (do tabulky `SOURADNICE_OBRAZU`). Filosofie a fyzický způsob realizace objektového typu `SDO_GEOMETRY` tuto konstrukci bohužel neumožňuje provést, prvky atributu `sdo_ordinates` musí být konkrétní číselné hodnoty, reprezentující souřadnice vrcholových bodů. Možnost přímého vytváření relací mezi použitými bloky v implementovaném relačním modelu a objekty typu `SDO_GEOMETRY` by jistě našla své uplatnění. Například v případě, kdy mám k dispozici generalizovaný geometrický popis objektu, by bylo možné tento popis uložit jako objekt typu `SDO_GEOMETRY` a do pole `sdo_ordinates` uložit pouze odkazy na příslušnou tabulku se souřadnicemi bodů. V databázi by tak nedocházelo k nadbytečnému ukládání dat, jako v případě, kdy musím pokaždé v poli `sdo_ordinates` uchovávat konkrétní souřadnice.

4.6.2 Rekonstrukce průběhu katastrální hranice

Z návrhu tabulky `KATASTRALNI_HRANICE` bylo zřejmé, že katastrální hranici budu uchovávat jako objekt typu `SDO_GEOMETRY`. Z obsahu předešlé kapitoly vyplývá, že jejím geometrickým popisem bude polygon, jehož spojnice budou tvořeny příkými liniovými elementy a souřadnice počátečního a koncového bodu každého elementu bude popsána ve dvou dimenzích rovinného souřadnicového systému.

V kapitole 4.4 jsem popsal mnou navržený postup, pomocí kterého by mělo být možné z rekonstruovaného relačního modelu vybrat všechny úseky katastrální hranice pro katastrální území, identifikované jeho kódem. Výsledkem postupu a příslušného SQL dotazu jsou společně s identifikátory hraničních úseků také souřadnice počátečního a koncového bodu úseku. Pořadí souřadnic bodů katastrální hranice vrácených dotazem však neodpovídá jejich skutečnému pořadí tak, jak se fyzicky tyto body vyskytují v terénu. Navíc je nutné tyto body uspořádat za sebou tak, aby jejich pořadí v atributu `sdo_ordinates` objektu `hranice` bylo v protisměru chodu hodinových ručiček a odpovídalo skutečnému průběhu hranice v terénu.

4.6.2.1 Program RELtoOO.exe

Účelem programu RELtoOO.exe je načíst soubor bodyNahr.txt s výsledkem dotazu na výběr úseků katastrální hranice popsaného v kapitole 4.4 a tyto úseky za sebou seřadit v protisměru chodu hodinových ručiček v pořadí, jaké odpovídá jejich skutečnému uspořádání v terénu. Z popisu a SQL zápisu dotazu je vidět, že jsou vybrány úseky vždy pro jedno konkrétní katastrální území.

Po spuštění programu je uživatel vyzván k zadání kódu katastrálního území a identifikátoru hranice. Kód katastrálního území je de facto předurčen vybranými daty. V rámci uživatelského komfortu by bylo možné přizpůsobit zdrojový kód výběrového dotazu a programu RELtoOO.exe způsobem, který by nepožadoval explicitní zadání hodnoty kódu katastrálního území. Stejně tak by šlo vhodným způsobem naprogramovat databázový trigger²⁴, který by automaticky generoval a vkládal identifikátory (id) hranice do tabulky **KATASTRALNI_HRANICE**. Výstupem programu je textový soubor RELtoOO.sql, obsahující SQL příkaz **INSERT**, vkládající do tabulky **KATASTRALNI_HRANICE** potřebné údaje.

Princip fungování programu RELtoOO.exe lze shrnout do následujících bodů:

- Načtení vstupního souboru s identifikátory hraničních úseků a souřadnicemi bodů katastrální hranice do vhodné datové struktury. Název vstupního souboru a cesta k němu jsou implicitně dány zdrojovým kódem programu. Proto je třeba při ukládání výsledku dotazu dodržet umístění a název souboru požadovaných programem. Program prochází vstupní soubor, u každého úseku uloží jeho identifikátor společně s dvěma páry souřadnic (souřadnice počátečního a koncového bodu úseku) do proměnné typu `usek`. Typ `usek` a ukazatel `pusek` na tento typ jsem si definoval následujícím způsobem:

TYPE

```
pusek=^usek;           //ukazatel na zaznam typu usek
usek = RECORD          //definice zaznamu typu usek
  id : REAL;           //identifikator useku
  x1,y1, x2,y2 : REAL; //souradnice poc. a kon. bodu
  pred : pusek;        //nasledujici zaznam
  dalsi : pusek;        //predchazejici zaznam
  projito : BOOLEAN;   //identifikator pridani useku
END;
```

Př. 14: Definice typu `usek` a ukazatele na tento typ `pusek`.

²⁴ Triggery jsou programovým kódem, který se vykonává na základě určité operace – nejčastěji nad tabulkou či záznamem.

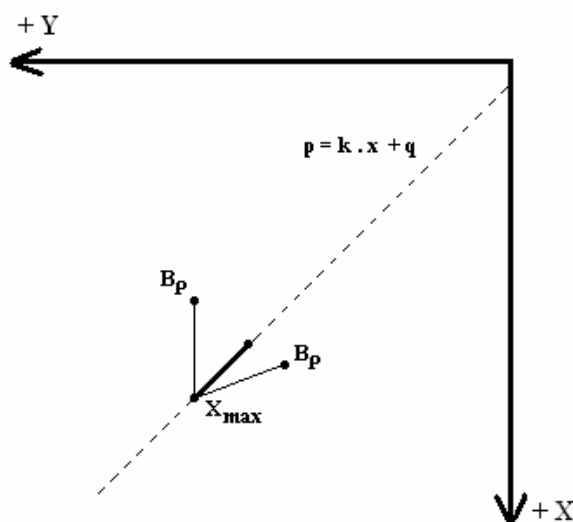
Ze záznamů s uloženými úseky hranic je při načítání dat vytvářen dynamický obousměrně zřetězený lineární seznam²⁵. U každého prvku seznamu je zřejmé, který záznam je jeho předchůdcem a který záznam je jeho následovníkem. Z použití této struktury vyplývá, že po dokončení procházení a čtení vstupního souboru má výsledný seznam právě tolik položek, kolik bylo vybráno úseků katastrální hranice, což přispívá k efektivnímu zacházení s paměťovým prostorem.

- Vyhledání úseku v seznamu vytvořeném při načítání dat (původní seznam) obsahujícího nejjižnější bod (maximální hodnota souřadnice X). Pokud by více úseků obsahovalo maximální hodnotu X-ové souřadnice, uvažoval by se jako počáteční úsek geometrického popisu hranice prvně nalezený úsek s maximální X-ovou souřadnicí. Poté jsem vytvořil **nový** obousměrně zřetězený seznam, jehož položky jsou typu `usek`. Do jeho počátku jsem umístil usek, který obsahuje nejjižnější bod a je považován za počáteční bod geometrického popisu. Zároveň jsem hodnotu proměnné `projito` u tohoto záznamu v původním seznamu nastavil na **TRUE**, což značí, že při hledání navazujícího záznamu (úseku) v původním seznamu vytvořeného při načítání dat nemá být tento úsek již uvažován. Definicí atributu `projito` jsem se snažil zvýšit efektivnost procházení prvotním seznamem a především snížit algoritmickou složitost programu.
- Seřazení úseků tak, jak na sebe ve skutečnosti navazují. Na úsek, který obsahuje nejjižnější bod a představuje první záznam nového seznamu, jsem navázal úsek další, pro který muselo platit, že jeden z jeho bodů je shodný s bodem prvního úseku, který ale zároveň není bodem nejjižnějším. Dále jsem vždy v původním seznamu hledal úsek, který by s posledně přidaným úsekem v novém seznamu měl shodný právě jeden bod. Takto jsem postupně vytvořil zřetězený seznam prvků a platilo, že každé dva sousední prvky nového seznamu (dva úseky) mají právě jeden společný bod. Po každém navázání úseku na předcházející úsek v novém seznamu došlo u odpovídajícího záznamu v původním seznamu k nastavení hodnoty atributu `projito` na hodnotu **TRUE** a úsek tak nebyl při dalším prohledávání původního seznamu dále uvažován. Potvrzením správnosti uvedeného postupu vedoucího k seřazení úseků katastrální hranice podle skutečného stavu byl okamžik, kdy se přidal do nového seznamu poslední prvek. Velice snadno se nechalo otestovat, zda posledně přidaný úsek má odlišný identifikátor od prvního prvku seznamu a zároveň obsahuje bod, jehož souřadnice odpovídají souřadnicím nejjižnějšího bodu prvního úseku. Kontrola této skutečnosti byla úspěšná, čímž se potvrdila správná programová realizace

²⁵ Možnost dynamického uložení množiny údajů do paměti vykazující se provázaností jednotlivých položek.

zamýšleného postupu seřazení hraničních úseků, ale i správnost a úplnost výběrového dotazu z kapitoly 4.4.

- Rozhodnutí, zda se úseky seřadily v novém seznamu od počátečního úseku geometrického popisu v protisměru či po směru chodu hodinových ručiček. Vzhledem k uvedeným skutečnostem v kapitole 4.4 je pro import do tabulky **KATASTRALNI_HRANICE** nutné mít body na katastrální hranici uspořádaný v protisměru chodu hodinových ručiček. K určení toho, v jakém směru se úseky přidávaly, jsem využil následující postup:
 - Pomocí souřadnic počátečního a koncového bodu prvního úseku mohu sestavit směrnici tvaru přímky p . Poslední přidáný úsek v novém spojovém seznamu má jeden bod shodný s počátečním bodem geometrického popisu hranice (jeho souřadnice $X = X_{MAX}$). Druhý bod tohoto úseku, označený jako $B_p [X_p, Y_p]$ zaujímá vůči přímce p polohu buď takovou, že leží nad ní nebo pod ní. Dosazením souřadnice X_p do rovnice přímky p dostanu souřadnici Y_p' . Porovnáním souřadnic Y_p a Y_p' dostanu informaci o tom, v jakém směru jsou za sebou v novém seznamu úseky katastrální hranice uloženy. Pokud $Y_p < Y_p'$, znamená to, že jsou úseky uloženy po směru chodu hodinových ručiček, v obráceném případě jsou úseky uloženy protisměru chodu hodinových ručiček. Možnou polohu bodu B_p vůči přímce p ilustruje obrázek 8.



Obr.7: Možná poloha bodu B_p vůči přímce p .

Ve zdrojovém kódu obsahuje informaci o orientaci uložení úseků hraničních úseků logická proměnná `protismeru`.

Pokud má hodnotu **TRUE**, znamená to orientaci v protisměru chodu hodinových ručiček.

- Vypsání příslušných hodnot do souboru RELtoOO.sql. Název a umístění tohoto souboru je implicitně dáno ve zdrojovém kódu programu *RELtoOO.exe*. Jak již bylo uvedeno, data jsem do tohoto souboru ukládal ve formě SQL příkazu **INSERT**. Bylo nutné v příslušném tvaru do souboru RELtoOO.sql vložit zadaný identifikátor hranice a kód katastrálního území, dále všechny údaje týkající se objektu *SDO_GEOMETRY* jako datového typu atributu *hranice* a nakonec hodnotu **NULL**. Při ukládání jednotlivých informací do příkazu **INSERT** je nutné dodržet pořadí jejich zápisu v souladu s definicí tabulky *KATASTRALNI_HRANICE*. Souřadnice bodů katastrální hranice byly vypsány do atributu *sdo_ordinates* v závislosti na hodnotě atributu *protismeru*. Pokud byla hodnota atributu *protismeru* rovna **TRUE**, vypsaly se body v pořadí, v jakém byly přidávány úseky do nového spojového seznamu. Pokud byla hodnota atributu *protismeru* rovna **FALSE**, vypsaly se body v obráceném pořadí. Aby se dodržela korektnost uložení dat podle kapitoly 4.4, ukládal se společný bod dvou sousedních hraničních úseků pouze jedenkrát s tím, že souřadnice prvního uloženého bodu se na konci výpisu uložily ještě jednou. Obsah souboru RELtoOO.sql tak vypadá následujícím způsobem (ukázka obsahuje data pro katastrální území Lomec u Klatov):

```
INSERT INTO katastralni_hranice VALUES (
    3,
    629499,
    SDO_GEOMETRY (
        2003,
        NULL,
        NULL,
        SDO_ELEM_INFO_ARRAY (1,1003,1),
        SDO_ORDINATE_ARRAY (
            1110904.46,837129.28,1109511.38,836978.48,...,
            1110904.46,837129.28)),
    NULL
);
```

Př. 15: Vložení jednoho prvku do tabulky *KATASTRALNI_HRANICE*.

Hodnota **NULL** zde reprezentuje hodnotu atributu *definicni_bod*. Naplnění atributu *definicni_bod* daty bude provedeno dále. Obsah souboru RELtoOO.sql byl vždy po vložení dat do databáze smazán a připraven pro uložení nových dat.

Závěrem této kapitoly bych rád upozornil ještě na jednu skutečnost, se kterou bylo třeba při realizaci programu RELtoOO.exe počítat. Při testování chování programu RELtoOO.exe program několikrát spadl. Při následném hledání důvodu přerušení běhu

programu jsem narazil na skutečnost, že v datech získaných z výměnného formátu může mít hraniční úsek z tabulky `HRANICE_PARCEL` uložený pod jednoznačným identifikátorem shodný počáteční a koncový bod. Tuto variantu jsem v programu patřičně ošetřil a učinil tak program `RELtoOO.exe` plně funkčním. Zdrojový kód programu `RELtoOO.exe` v jazyce PASCAL lze nalézt na přiloženém CD k této diplomové práci v souboru `RELtoOO.pas`.

4.7 Import dat do tabulky `KATASTRALNI_HRANICE`

K importu dat do tabulky `KATASTRALNI_HRANICE` jsem použil databázového klienta TOAD. Vzhledem k formátu souboru `RELtoOO.sql` bylo možné tento soubor v klientu snadno otevřít a nechat vykonat uložený příkaz `INSERT`. Tímto způsobem jsem se snažil vložit data pro každý nový záznam.

Při importu dat ze souboru `RELtoOO.sql`, který v tu chvíli obsahoval data o průběhu hranice katastrálního území Týnec u Janovic nad Úhlavou, oznámil databázový klient chybu, že vstupní data obsahují příliš mnoho argumentů. Katastrální území Týnec u Janovic nad Úhlavou bylo nejrozsáhlejším územím v datech, která jsem měl pro účely diplomové práce k dispozici. Pokusil jsem se příčinu oznámení chyby nalézt v [11]. Ukázalo se, že Spatial neumí pomocí standardního příkazu `INSERT` vložit do atributu typu `SDO_GEOMETRY` pole `sdo_ordinates`, které obsahuje více jak 999 souřadnic bodů. Tento problém se nechá řešit dvěma způsoby:

- 1) Využití programovacího jazyka PL/SQL.
- 2) Využití metody importu zvané „BULK LOADING“.

4.7.1 Import dat s využitím PL/SQL

Při řešení problému s importem rozsáhlejší množiny dat jsem využil právě možnosti programovacího jazyka PL/SQL. Obecně se nechá řešení s využitím jazyka PL/SQL na vložení takovéto množiny dat popsat následujícím způsobem:

- Definování procedury, která vloží do příslušné tabulky, v mém případě tabulky `KATASTRALNI_HRANICE`, potřebná data včetně proměnné typu `SDO_GEOMETRY` s přiřazeným geometrickým popisem katastrální hranice, přičemž tento popis může obsahovat více jak 999 souřadnic bodů. Horní limit na omezení počtu souřadnic vkládaných v poli `sdo_ordinates` jsem se z žádných dostupných zdrojů nedozvěděl.

- Deklarování proměnné typu `SDO_GEOMETRY`, které se přiřadí geometrický popis hranice katastrálního území, v mém případě ze souboru RELtoOO.sql.
- Volání procedury na vložení dat.

Takto jsem do tabulky `KATASTRALNI_HRANICE` úspěšně importoval záznam s geometrickým popisem hranice katastrálního území Týnec u Janovic nad Úhlavou. Toto území bylo z testovacích území jediné, u kterého se musel využít tento postup, řešící omezení na vkládání rozsáhlejší množiny dat v objektu typu `SDO_GEOMETRY`. Při realizaci výše uvedeného postupu jsem proceduru na vložení dat pojmenoval `InsertGeom`. Popis hranice jsem uložil do proměnné `pom`. Celý postup jasněji z následující ukázky zapsané v jazyce PL/SQL:

```
CREATE OR REPLACE PROCEDURE InsertGeom(
  pom_id      NUMBER,
  pom_kod     NUMBER,
  pom_geom    SDO_GEOMETRY,
  pom_def     SDO_GEOMETRY
)
AS
BEGIN
  INSERT INTO KATASTRALNI_HRANICE VALUES (
    pom_id, pom_kod, pom_geom, pom_def
  );
END;
/

DECLARE
  pom tester1.KATASTRALNI_HRANICE.HRANICE%TYPE := SDO_GEOMETRY (
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1),
    SDO_ORDINATE_ARRAY(
      111371.62,838236.96,111371.44,838239.87,...,
      111371.62,838236.96)
  );
BEGIN
  InsertGeom(7,772313,pom,null);
END;
/
```

Př. 16: Vložení geometrického popisu prvku o více než 999 vrcholových bodech do tabulky `KATASTRALNI_HRANICE`.

Tento postup jsem uskutečnil pomocí databázového klienta Oracle SQL*Plus.

4.7.2 Využití metody „BULK LOADING“

Tuto metodu jsem prakticky nevyzkoušel, proto zde velmi stručně podám pouze základní přehled o účelu a principu této metody, jako jednoho z možných způsobů řešení importu rozsáhlé množiny prostorových dat.

Použití této metody přímo souvisí s obslužným programem (tzv. *utilitou*) SQL*Loader, který je standardně nainstalován při instalaci samotného databázového prostředí Oracle. SQL*Loader umožňuje importovat do databáze velké množství ASCII dat.

SQL*Loader na svém vstupu zpracovává soubor s daty, tzv. *Data File* pomocí několika parametrů, z nichž nejdůležitější je parametr, jenž specifikuje kontrolní soubor, tzv. *Control File*. *Data File* obsahuje data reprezentující geometrické popisy vybraných prostorových prvků. Způsob zpracování těchto dat se řídí instrukcemi obsaženými v kontrolním souboru. Bližší informace o této metodě lze najít v [11].

5 Zobrazení prostorových dat

Společnost Oracle nabízí vlastní produkt pro zobrazování dat uložených v databázi Oracle a využívajících technologii Spatial. Jedná se o aplikaci MapViewer.

5.1 MapViewer

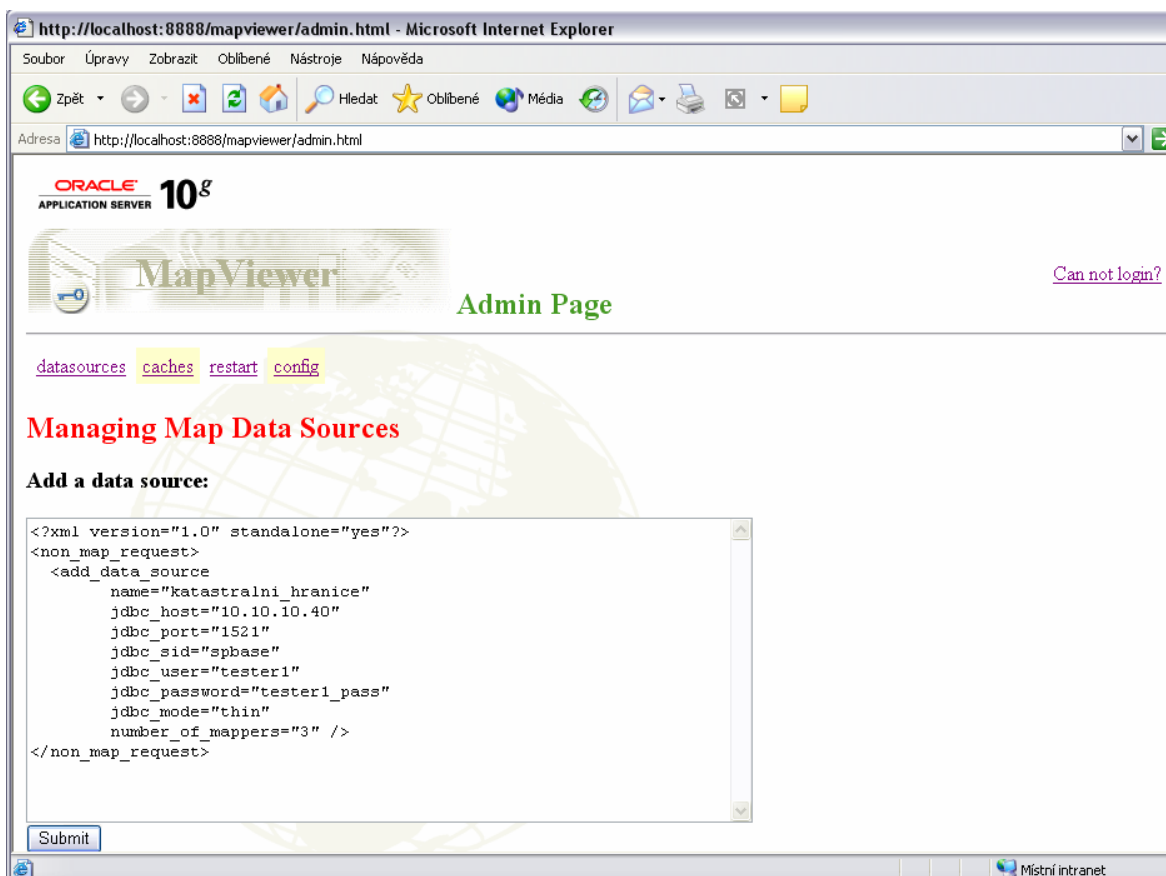
MapViewer je vizualizační nástroj založený na programovacím jazyce Java. Je součástí aplikačního serveru Oracle 10g. Může být použit k

- vytváření uživatelských map zobrazujících geografické prvky jako například silnice, vodní dopravní cesty a jiné transportní sítě.
- zobrazování hranic územně správních jednotek.
- vizualizovat tzv. „business“ data – různé demografické ukazatele populace a podobně.

MapViewer nabízí kromě jiného prostředek **Simple Spatial Query Visualizer** pro vizualizaci prostorových dotazů zapsaných v strukturovaném dotazovacím jazyce SQL. Simple Spatial Query Visualizer jsem využil pro zobrazování hranic katastrálních území. Tento prostředek umožňuje zobrazit data uložená v objektovém datovém typu *SDO_GEOMETRY*.

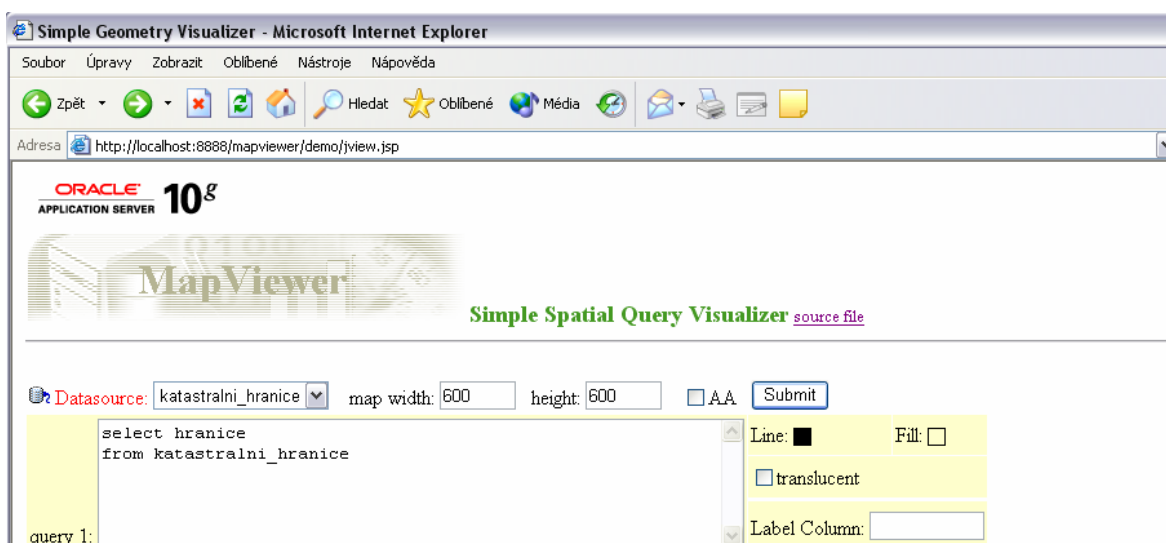
5.1.1 Využití Simple Spatial Query Visualizer

Před zahájením tvorby a vizualizace prostorových dotazů je nutno definovat datový zdroj, to znamená tabulku která obsahuje alespoň jeden sloupec typu *SDO_GEOMETRY*. Připojení k této zdrojové tabulce, uložené v databázi Oracle, probíhá pomocí JDBC. Ukázka definování připojení zdrojové tabulky, konkrétně tabulky *KATASTRALNI_HRANICE*, je popsána v příkladu 16. Uživatel určí pojmenování zdroje (název připojované tabulky), svoji IP adresu, JDBC port (stačí ponechat výchozí hodnotu 1521), globální název databáze obsahující zdrojovou tabulku, jméno platného uživatele včetně jeho hesla.



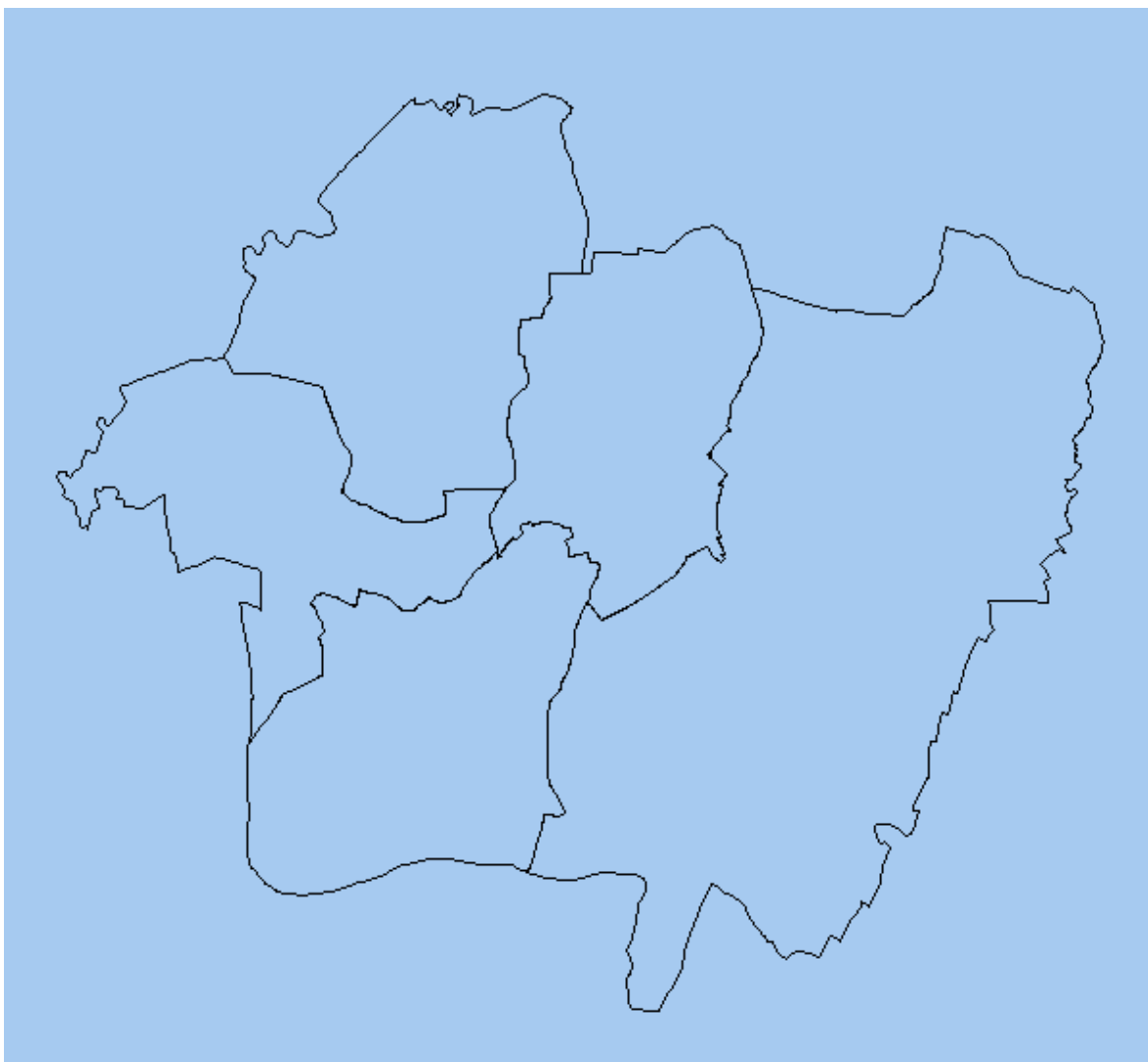
Př. 17: Připojení datového zdroje k aplikaci MapViewer.

Po úspěšném připojení tabulky KATASTRALNI_HRANICE k aplikaci MapViewer je možné spustit nástroj Simple Spatial Query Visualizer a zadat dotaz na vybrání všech geometrických popisů katastrálních hranic uložených ve sloupci hranice, viz příklad 17.



Př. 18: Zadání dotazu pro vykreslení „přehledové“ mapy katastrálních území.

Po stisknutí tlačítka *Submit* dojde k vykreslení hranic katastrálních území, jejichž geometrické popisy jsou uloženy v atributu `hranice`, viz obrázek 8.



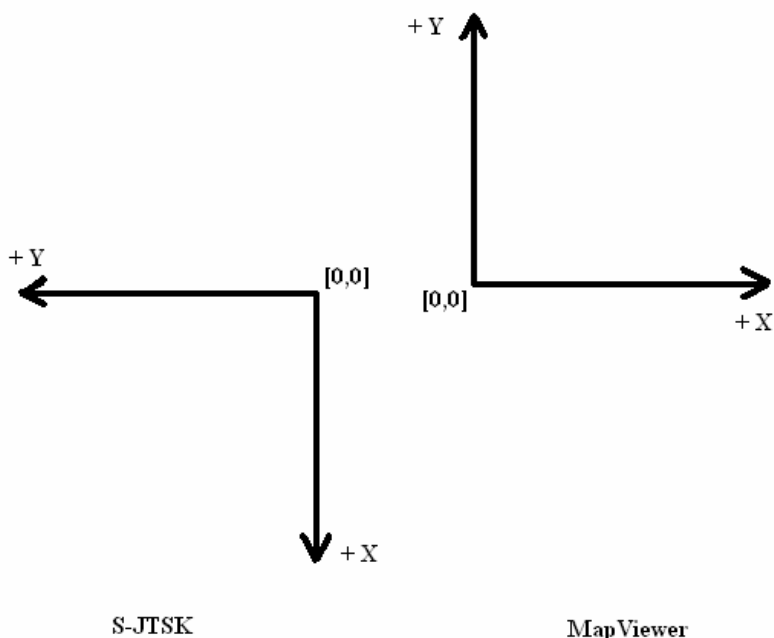
Obr. 8: Vykreslené hranice katastrálních území pomocí nástroje Simple Spatial Query Visualizator.

Na obrázku 8 nejsou uvedeny popisky názvů jednotlivých katastrálních území. Je to z důvodu, že v tabulce `KATASTRALNI_HRANICE` není žádný sloupec, který by popisky názvů obsahoval. Při návrhu tabulky `KATASTRALNI_HRANICE` jsem s tímto sloupcem nepočítal, neboť při jejím navázání do zrekonstruovaného a implementovaného datového modelu by tato informace byla redundantní.

5.2 Orientace os v MapViewer a S-JTSK

Simple Spatial Query Visualizator sice umožňuje rychlé vykreslení dat typu `SDO_GEOMETRY`, je však zapotřebí uvědomit si orientaci os souřadnicového systému, kterou tento prostředek stejně jako celá technologie Oracle Spatial využívá v okamžiku, kdy s daty nemám

asociovaný žádný ze souřadnicových systémů implicitně definovaných prostředím Spatial. V takovém případě spojí Spatial tato data s rovinným pravoúhlým souřadnicovým systémem. Orientace os tohoto systému je odlišná od orientace os souřadnicového systému S-JTSK, jak dokládá obrázek 9.



Obr. 9: Porovnání orientace os.

Z důvodu rozlišné orientace os souřadnicového systému S-JTSK a použitého systému pro zobrazení dat není vyobrazený stav na obrázku 8 geograficky korektní. Proto, abych vykreslil data polohově správným způsobem, bylo nutné uvažovat zmíněnou rozdílnost orientace os použitých souřadnicových systémů. Problém by se nechal nejspíš řešit naprogramováním určitého rozšíření použitého zobrazovacího nástroje. K realizaci takového postupu by bylo nutné důsledné a detailní pochopení celé technologie MapViewer. Pro potřeby své diplomové práce jsem zvolil druhé z možných řešení, a to přechod z uložení dat v souřadnicovém systému S-JTSK do uložení v matematickém souřadnicovém systému. Pro převod jsem použil následující transformační rovnice:

$$\begin{aligned} - X_{\text{MapViewer}} &= Y_{\text{S-JTSK}} \\ - Y_{\text{MapViewer}} &= X_{\text{S-JTSK}} \end{aligned} \quad (1)$$

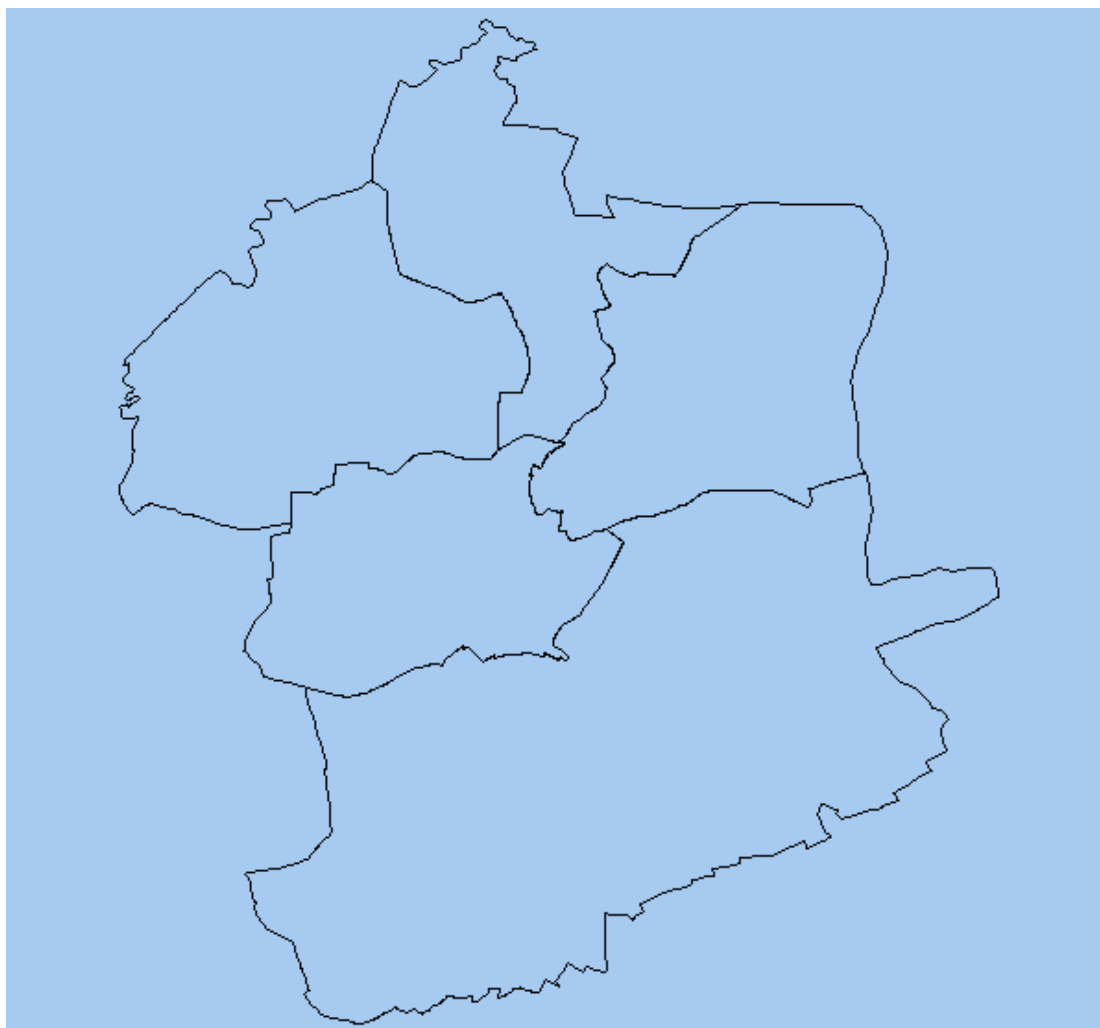
5.3 Rozšíření fyzického datového modelu - vytvoření tabulky *KATASTRALNI_HRANICE_MAT* a její naplnění daty

Vytvořením tabulky *KATASTRALNI_HRANICE_MAT* jsem se snažil vyřešit problém s rozdílnou orientací os souřadnicového systému S-JTSK a použitého souřadnicového systému, ve kterém jsou data vykreslována aplikací MapViewer. Tabulka *KATASTRALNI_HRANICE_MAT* obsahuje shodné atributy jako tabulka *KATASTRALNI_HRANICE* s tím rozdílem, že data geometrického popisu průběhu katastrální hranice jsou uložena v matematickém souřadnicovém systému, čímž dojde ke správnému geografickému vykreslení hranic katastrálních území. Struktura tabulky *KATASTRALNI_HRANICE_MAT* je patrná z její definice zapsané v jazyce SQL:

```
CREATE TABLE KATASTRALNI_HRANICE_MAT (  
    id                NUMBER PRIMARY KEY,  
    katuze_kod        NUMBER(6),  
    hranice           SDO_GEOMETRY,  
    definicni_bod     SDO_GEOMETRY  
);
```

Př. 19: Definice tabulky *KATASTRALNI_HRANICE_MAT*.

Data jsem do této tabulky importoval shodným způsobem, jako tomu bylo v případě tabulky *KATASTRALNI_HRANICE*. Ve zdrojovém kódu programu RELtoOO.exe jsem pouze změnil formát vypisovaných souřadnic podle transformačních rovnic (1), abych docílil uložení souřadnic bodů v matematickém souřadnicovém systému. Po naplnění tabulky *KATASTRALNI_HRANICE_MAT* daty jsem tuto tabulku fyzicky připojil přes atribut *katuze_kod* do zrekonstruovaného datového modelu. Obrázek 10 zobrazuje vykreslené hranice katastrálních území z transformovaných dat. Vyobrazený způsob průběhu hranic již odpovídá skutečnému průběhu v S-JTSK.



Obr. 10: Vykreslená přehledová mapa z transformovaných dat.

5.3.1 Ztráta přesnosti při uložení v matematickém souřadnicovém systému

Při uložení geometrického popisu katastrální hranice v matematickém souřadnicovém systému dojde ke ztrátě centimetrové přesnosti souřadnic v ose $-Y_{\text{MapViewer}}$. Je to dáno datovým typem, ve kterém Spatial uchovává souřadnice geometrického popisu prostorových prvků, viz kapitola 3.4.5. Jednotlivé souřadnice jsou zaznamenávány v datovém typu *NUMBER*, bez bližšího určení možného počtu celých a desetinných míst, což znamená, že pro uložení souřadnice je k dispozici celkem 10 míst [13],[15].

Pokud má tedy souřadnice X v systému S-JTSK vyjádřená s centimetrovou přesností celkem 10 míst (nutno započítat i oddělovač celé a desetinné části), po transformaci podle rovnic (1) bude souřadnice $-Y_{\text{MapViewer}}$ zaujímat při centimetrové přesnosti celkem 11 míst (kvůli znaménku „-“). Z tohoto důvodu dojde při ukládání souřadnic do pole `sdo_ordinates` atributu `hranice` v tabulce `KATASTRALNI_HRANICE_MAT` k uložení souřadnic $-Y_{\text{MapViewer}}$ pouze

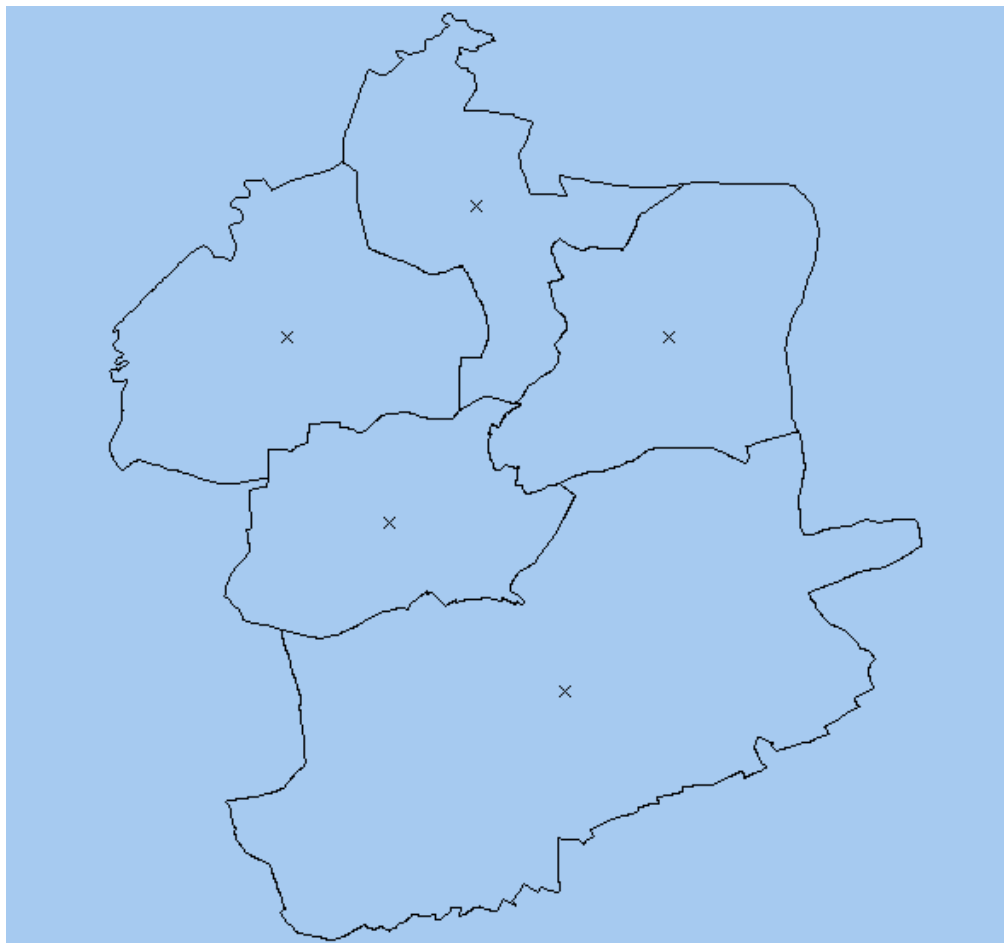
s decimetrovou přesností. Tento fakt není ale pro účely mé diplomové práce nikterak rozhodující a zásadní.

5.3.2 Naplnění atributu `definicni_bod`

Tabulka `KATASTRALNI_HRANICE_MAT` stejně jako tabulka `KATASTRALNI_HRANICE` obsahuje atribut `definicni_bod` objektového datového typu `SDO_GEOMETRY`. Ten má sloužit k uchování souřadnic definičního bodu katastrálního území. Jelikož jsem k zrekonstruovanému datovému modelu fyzicky připojil pouze tabulku `KATASTRALNI_HRANICE_MAT`, naplnil jsem atribut `definicni_bod` pouze u této tabulky.

K naplnění atributu `hranice` jsem využil standardní funkci prostředí `Spatial` `SDO_GEOM.SDO_CENTROID`. Vstupními parametry této funkce jsou geometrický popis katastrální hranice, to znamená atribut typu `SDO_GEOMETRY` a hodnota tolerance ve smyslu popsáném v kapitole 3.3.5. Funkce `SDO_GEOM.SDO_CENTROID` vrací souřadnice bodu, který představuje těžiště plochy dané vstupním geometrickým popisem. Přesný popis algoritmu výpočtu funkce `SDO_GEOM.SDO_CENTROID` není k dispozici. Nicméně se ukázalo, že při „nevhodném“ tvaru speciálně namodelovaného průběhu katastrální hranice může vypočtený bod ležet mimo toto území.

Grafické znázornění vypočtených definičních bodů testovacích katastrálních území je ukázáno na obrázku 11.



Obr. 11: Ukázka vypočtených centroidů.

Podrobné informace o funkci `SDO_GEOM.SDO_CENTROID` lze nalézt v [11].

5.3.3 Metadata o sloupcích typu `SDO_GEOMETRY` tabulky `KATASTRALNI_HRANICE_MAT`

Jelikož tabulka `KATASTRALNI_HRANICE_MAT` obsahuje dva sloupce typu `SDO_GEOMETRY` – `hranice` a `definicni_bod`, je nutné vložit příslušná metadata do pohledu `USER_SDO_GEOM_METADATA`.

Metadata o sloupci `hranice` vypadají následujícím způsobem:

```
INSERT INTO USER_SDO_GEOM_METADATA
VALUES (
    'katastralni_hranice_mat',
    'hranice',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('-X', 0, -950000, 0.005),
        SDO_DIM_ELEMENT('-Y', 0, -1250000, 0.005)
    ),
    NULL
);
```

Př. 20: Metadata o sloupci `hranice` v tabulce `KATASTRALNI_HRANICE_MAT`.

Metadata o sloupci `definicni_bod` vypadají až na název sloupce shodným způsobem:

```
INSERT INTO USER_SDO_GEOM_METADATA
VALUES (
    'katastralni_hranice_mat',
    'definicni_bod',
    SDO_DIM_ARRAY(
    SDO_DIM_ELEMENT('-Y', 0, -1250000, 0.005),
    SDO_DIM_ELEMENT('-X', 0, -950000, 0.005)
    ),
    NULL
);
```

Př. 21: Metadata o sloupci `definicni_bod` v tabulce `KATASTRALNI_HRANICE_MAT`.

5.4 Kontrola validity dat

Spatial umožňuje kontrolovat správnost geometrického popisu prostorového prvku pomocí funkce `SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT`. Syntaxe této funkce je následující:

```
SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT(
    theGeometry      IN SDO_GEOMETRY,
    tolerance        IN NUMBER
) RETURN VARCHAR2;
```

Př. 22: Syntaxe funkce `SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT`.

Parametr `theGeometry` obsahuje geometrický popis prostorového objektu, parametr `tolerance` pak představuje podrobnost dat, viz kapitola 3.3.5. Pokud je geometrický popis validní, vrací tato funkce hodnotu **TRUE**.

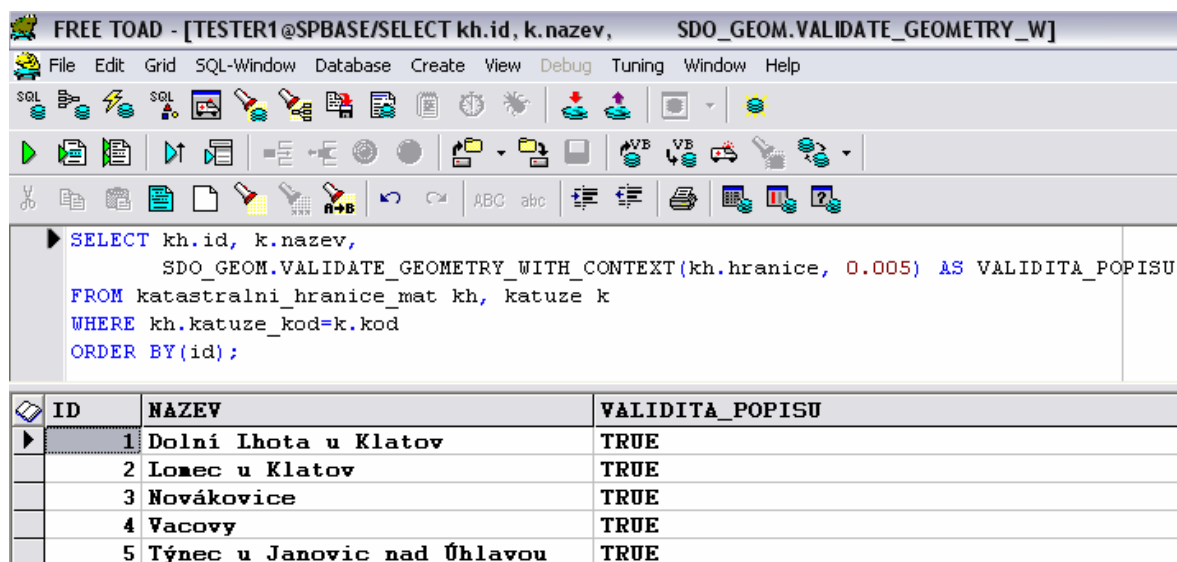
Funkce provádí kontrolu typové konzistence, při které je kontrolována validita atributu `sdo_gtype`, dále zda jsou hodnoty v atributu `sdo_etype` konzistentní s hodnotou `sdo_gtype` (pokud mám hodnotu `sdo_gtype` rovnu **2003** očekává se, že geometrický popis bude popisovat alespoň jeden polygonový objekt) a zda `sdo_elem_info_array` obsahuje validní trojici hodnot.

Kromě kontroly typové konzistence provádí funkce `SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT` kontrolu geometrické konzistence. Aby byl geometrický popis, v mém případě katastrální hranice popsán pomocí polygonu validní, musí být splněny následující požadavky:

- Popis polygonu musí obsahovat nejméně čtyři body, přičemž poslední bod popisu musí být shodný s prvním, aby došlo k uzavření polygonu.
- Žádné dva body popisu polygonu nesmějí být shodné.
- Polygon musí být orientován správným způsobem, v mém případě to znamená orientaci protisměru chodu hodinových ručiček.
- Souřadnice musí být v rozsahu definovaného ve sloupci `diminfo` v příslušných metadatech.

Pokud geometrický popis prvku není validní, vrátí Spatial při kontrole validity pomocí této funkce číslo chybové zprávy (de facto číslo chyby) a příznak chyby.

Otestoval jsem data uložená ve sloupci `hranice` tabulky `KATASTRALNI_HRANICE_MAT` s následujícími výsledky:



```

SELECT kh.id, k.nazev,
       SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT(kh.hranice, 0.005) AS VALIDITA_POPISU
FROM katastralni_hranice_mat kh, katuze k
WHERE kh.katuze_kod=k.kod
ORDER BY(id);
    
```

ID	NÁZEV	VALIDITA_POPISU
1	Dolní Lhota u Klatov	TRUE
2	Lomec u Klatov	TRUE
3	Novákovice	TRUE
4	Vacovy	TRUE
5	Týnec u Janovic nad Úhlavou	TRUE

Př. 23: Testování validity dat ve sloupci `hranice`.

Z uvedené ukázky vyplývá validita všech geometrických popisů katastrálních hranice uložených ve sloupci `hranice`.

5.5 Použití operátoru TOUCH

Společné části katastrální hranice dvou sousedních katastrálních území se podle obrázku 10 jeví jako identické. Abych ale získal exaktní kontrolu identičnosti tohoto společného úseku, využil jsem operátoru `TOUCH`, který je v Spatial standardně k dispozici.

Operát `TOUCH` má následující syntaxi:

```
SDO_TOUCH(geometry1, geometry2);
```

Př. 24: Syntaxe operátoru `SDO_TOUCH`.

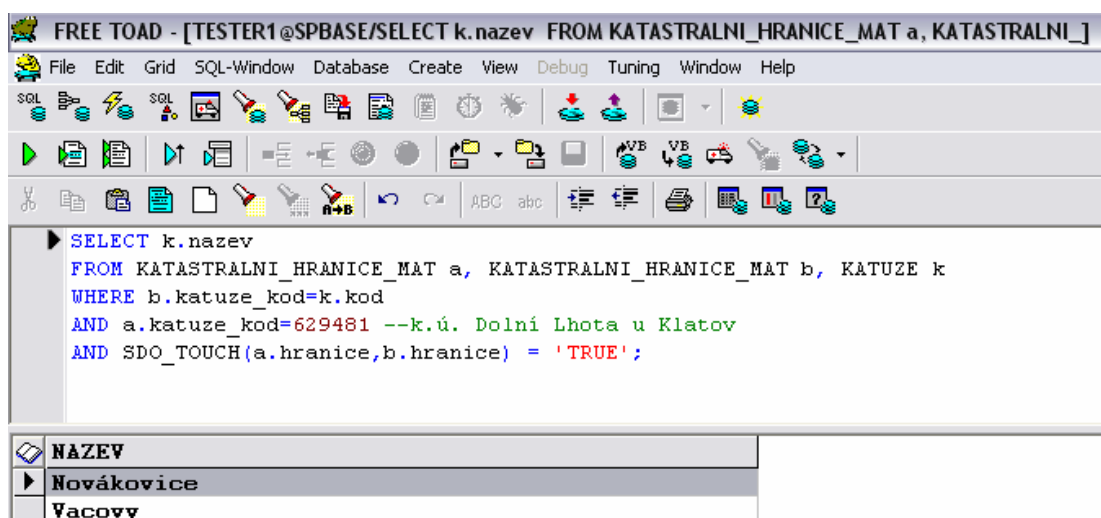
Parametr `geometry1` reprezentuje geometrický popis (popisy) prvku (prvků), vůči němuž zkoumáme topologický vztah geometrického popisu (popisů) reprezentovaného atributem `geometry2`. Pokud mají dva porovnávané geometrické popisy topologický vztah typu TOUCH, to znamená část popisu jednoho z nich je identická s částí popisu druhého z nich, je operátor vyhodnocen na hodnotu **TRUE**. Atribut `geometry1` i `geometry2` jsou datového typu `SDO_GEOMETRY`. Pro použití tohoto operátoru bylo nutné vytvořit prostorový index (R-tree index) nad sloupcem reprezentovaného parametrem `geometry1`:

```
CREATE INDEX kat_hran_mat_idx
ON katastralni_hranice_mat(hranice)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Př. 25: Vytvoření prostorového indexu nad sloupcem `hranice`.

Pro každé katastrální území jsem otestoval, zda je společná část katastrální hranice s jeho sousedními katastrálními územími opravdu identická. Ukázalo se, že ano.

V následující ukázce se mají vypsat všechna katastrální území, která sousedí (mají vztah TOUCH) s katastrálním územím Dolní Lhota u Klatov:



Př. 26: Testování topologických vztahů pomocí operátoru TOUCH.

Výsledek dotazu odpovídá skutečnému stavu, samozřejmě s ohledem na omezenou množinu testovacích dat.

6 Testování generalizačních funkcí v Oracle Spatial 10g

V kapitole 3.7 a 3.8 jsem teoreticky popsal tři funkce a jednu proceduru, které se v prostředí Spatial nechají přímo či vhodnou volbou vstupních parametrů použít ke geometrickým generalizačním účelům. Tyto podprogramy jsem otestoval na datech uložených v tabulce KATASTRALNI_HRANICE_MAT ve sloupci hranice.

Návratovou hodnotou otestovaných funkcí je generalizovaný geometrický popis katastrální hranice uložený v objektovém datovém typu *SDO_GEOMETRY*. Pro uchování generalizovaných dat jsem v tabulce KATASTRALNI_HRANICE_MAT definoval nový sloupec typu *SDO_GEOMETRY*, do kterého jsem uložil výsledky použité funkce. V jazyce PL/SQL jsem naprogramoval postup, během kterého došlo k načtení vstupních hodnot (záznamy ve sloupci hranice), k aplikování generalizační funkce na každý vybraný záznam a uložení zjednodušeného popisu zpět do tabulky KATASTRALNI_HRANICE_MAT do nově definovaného sloupce a odpovídajícího řádku. Nový sloupec byl definován pro každou nově použitou funkci a zároveň pro různé hodnoty parametrů jedné a té samé funkce. Ukázka zmíněného programu v jazyce PL/SQL:

```
DECLARE
  pom_geom HRANICE_VIEW.HRANICE_VIEW%TYPE;
  simpl_geom HRANICE_VIEW.HRANICE_VIEW%TYPE;
  pom_id HRANICE_VIEW.id%TYPE;
  pocet NUMBER;
  pom NUMBER;
CURSOR k1 IS
  SELECT id,HRANICE
  FROM KATASTRALNI_HRANICE_MAT
  WHERE id BETWEEN 1 AND 5;
BEGIN
  pom:=0;
  pocet:=5;
  OPEN k1;
  LOOP
    FETCH k1 INTO pom_id,pom_geom;
    SELECT sdo_util.SIMPLIFY(hranice,50) INTO simpl_geom
    FROM KATASTRALNI_HRANICE_MAT
    WHERE id=pom_id;
    UPDATE KATASTRALNI_HRANICE_MAT
    SET simplify_50=simpl_geom
    WHERE id=pom_id;
    pom:=pom+1;
    IF pom=pocet THEN
      EXIT;
    END IF;
  END LOOP;
  CLOSE k1;
END;
/
```

Př. 27: Generalizace geometrických popisů uložených ve sloupci hranice.

Uvedená ukázka aplikuje na načtené záznamy funkci SDO_UTIL.SIMPLIFY s hodnotou parametru `threshold` rovno 50 a výsledný zjednodušený geometrický popis uloží do sloupce `simplify_50`. Jediným případem, kdy jsem postupoval odlišně od uvedeného způsobu získávání a ukládání generalizovaných dat, bylo použití procedury SDO_SAM.SIMPLIFY_LAYER. Zde stačilo vytvořit nový sloupec v tabulce KATASTRALNI_HRANICE_MAT a jedním voláním procedury SDO_SAM.SIMPLIFY_LAYER naplnit celý sloupec novými záznamy bez nutnosti využití volání procedury v cyklu.

Do tabulky KATASTRALNI_HRANICE_MAT jsem přidal oproti její původní definici celkem devět nových sloupců, do nichž jsem uložil data podle následující tabulky:

Název nově definovaného sloupce	Použitý podprogram
<code>simplify_20</code>	Funkce SDO_UTIL.SIMPLIFY s hodnotou parametru <code>threshold</code> rovno 20.
<code>simplify_50</code>	Funkce SDO_UTIL.SIMPLIFY s hodnotou parametru <code>threshold</code> rovno 50.
<code>simpl_geom_5</code>	Funkce SDO_SAM.SIMPLIFY_GEOMETRY s hodnotou parametru <code>pct_area_change_limit</code> rovno 5 a hodnotou parametru <code>tol</code> rovno 0,005.
<code>simpl_geom_10</code>	Funkce SDO_SAM.SIMPLIFY_GEOMETRY s hodnotou parametru <code>pct_area_change_limit</code> rovno 10 a hodnotou parametru <code>tol</code> rovno 0,005.
<code>simpl_geom_20</code>	Funkce SDO_SAM.SIMPLIFY_GEOMETRY s hodnotou parametru <code>pct_area_change_limit</code> rovno 20 a hodnotou parametru <code>tol</code> rovno 0,005.
<code>simpl_duplic_1</code>	Funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES s hodnotou parametru <code>tolerance</code> rovno 1.
<code>simpl_duplic_10</code>	Funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES s hodnotou parametru <code>tolerance</code> rovno 10.
<code>simpl_duplic_50</code>	Funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES s hodnotou parametru <code>tolerance</code> rovno 50.
<code>simpl_layer_10</code>	Procedura SDO_SAM.SIMPLIFY_LAYER s hodnotou parametru <code>pct_area_change_limit</code> rovno 10 a hodnotou parametru <code>tol</code> rovno 0,005.

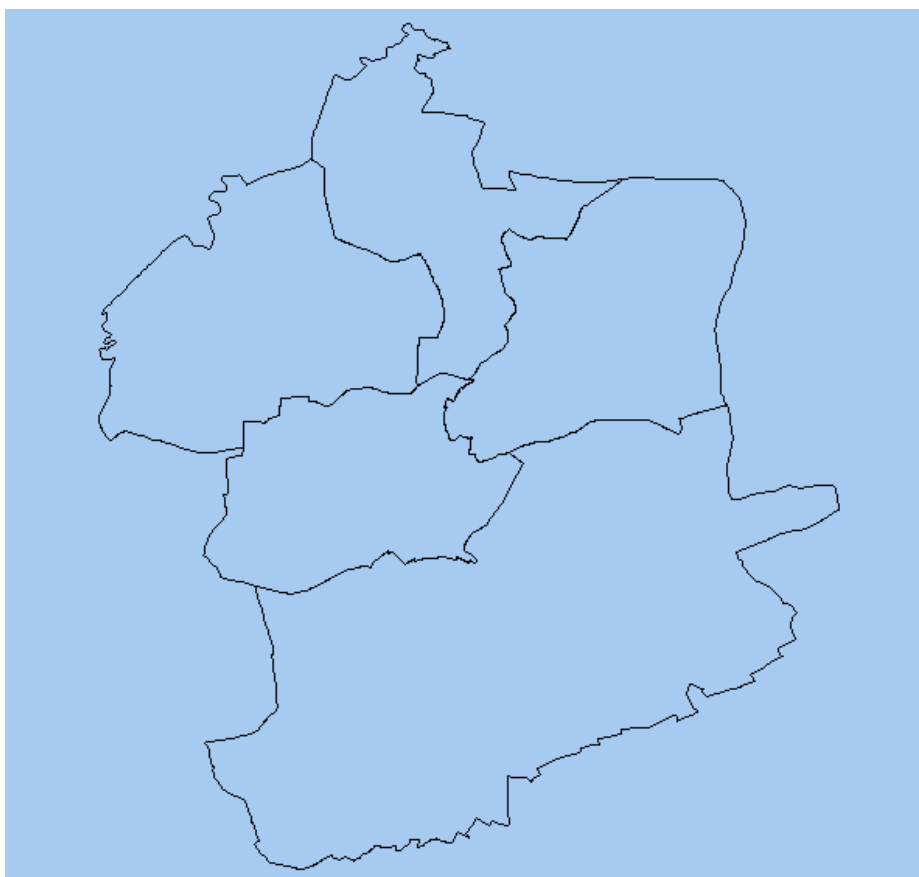
Tab. 5 Nově přidané sloupce do tabulky KATASTRALNI_HRANICE_MAT.

6.1 Testování funkce

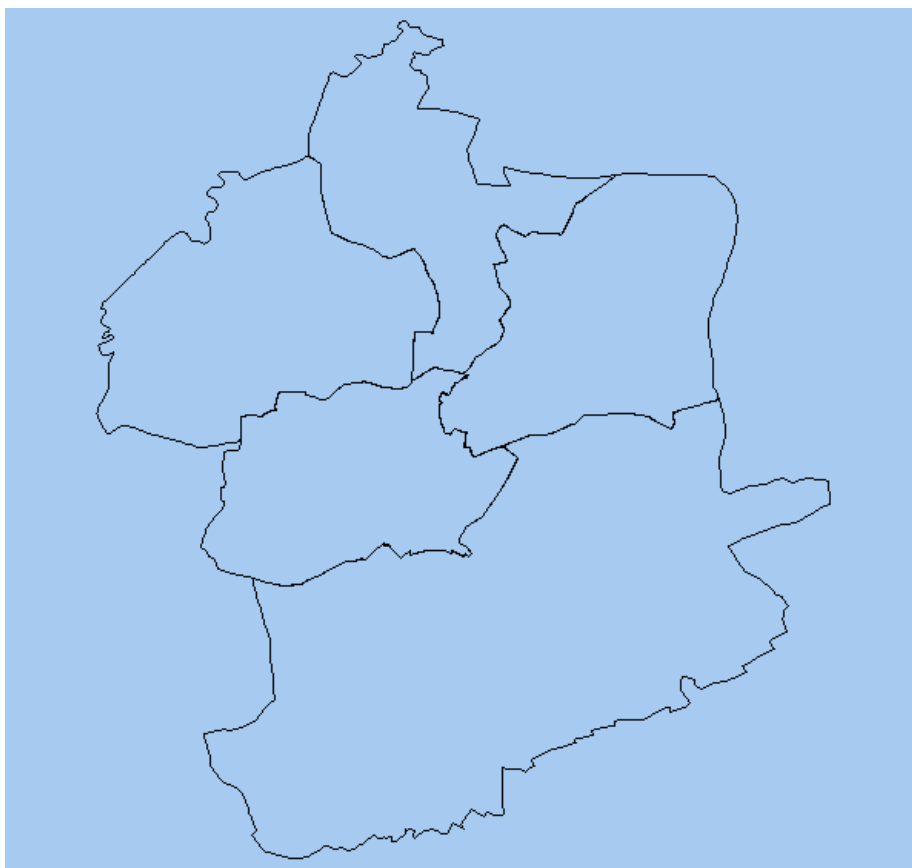
SDO_UTIL.REMOVE_DUPLICATE_VERTICES

Funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES neslouží primárně jako generalizační funkce. Vhodnou volbou vstupního atributu (konkrétně hodnoty atributu tolerance) lze ale tuto funkci ke generalizačním účelům použít.

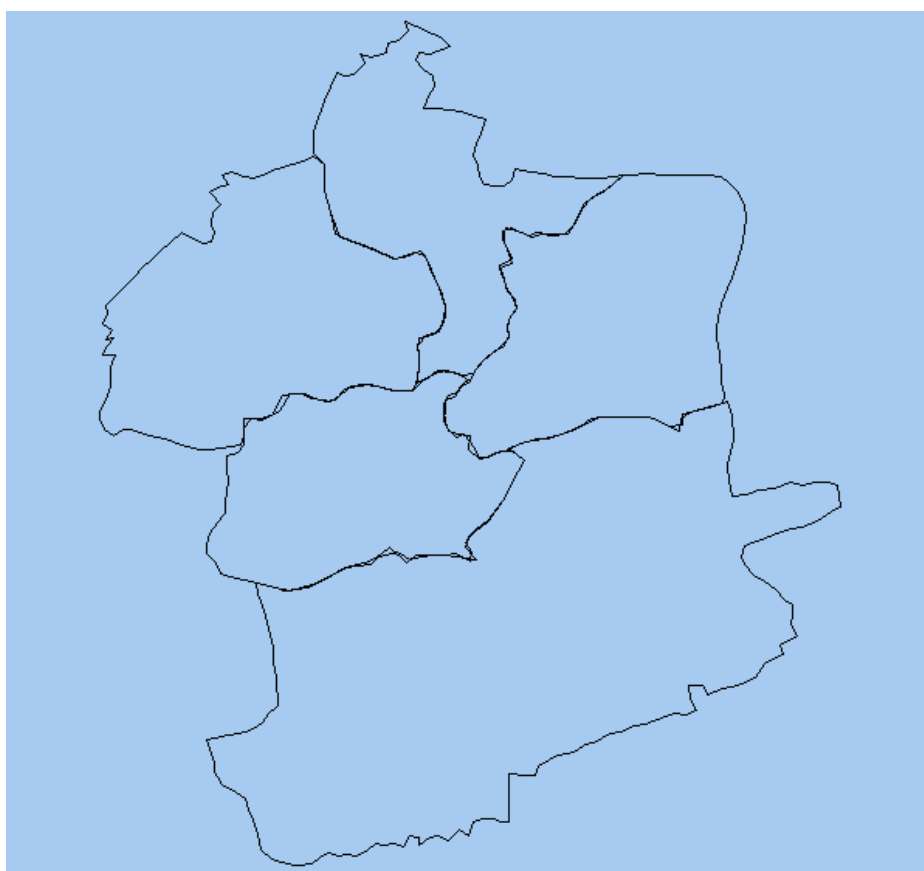
Jak je patrné z tabulky 5, funkci jsem testoval pro tři různé vstupní hodnoty atributu *tolerance*. Obrázky 12, 13 a 14 ilustrují výsledky aplikování funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES s různými hodnotami atributu *tolerance*. Výsledné geometrické popisy katastrální hranice jsem u všech použitých funkcí a procedury vizualizoval pomocí nástroje Simple Spatial Query Visualisator.



Obr. 12: Hodnota atributu *tolerance* rovno **1**.



Obr. 13: Hodnota atributu *tolerance* rovno **10**.



Obr. 14: Hodnota atributu *tolerance* rovno **50**.

Tabulka 6 (jako výsledek SQL dotazu realizovaného v klientu TOAD) obsahuje údaje o počtu navzájem různých bodů katastrální hranice u původního negeneralizovaného geometrického popisu a generalizovaných popisů pro příslušnou hodnotu atributu `tolerance`. K výpočtu počtu vrcholů v geometrickém popisu jsem použil standardní funkci nabízenou v prostředí Spatial SDO_UTIL.GETNUMVERTICES.

NAZEV_KU	PUVODNI_POPIS	SIMPL_DUPLIC_1	SIMPL_DUPLIC_10	SIMPL_DUPLIC_50
Dolní Lhota u Klatov	463	457	296	103
Lomec u Klatov	306	303	215	87
Novákovice	387	382	283	112
Vacovy	328	327	225	81
Týnec u Janovic nad Úhlavou	516	515	375	164

Tab. 6: Údaje o počtu vrcholů katastrální hranice při použití funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES.

Hodnoty v tabulce 6 potvrzují předpokládané chování funkce. S rostoucí hodnotou atributu `tolerance` dochází k redukci počtu vrcholů geometrického popisu katastrální hranice. Při pohledu na obrázek 8 lze rovněž vyzorovat, že se nezachová identický průběh společného úseku katastrální hranice mezi dvěma katastrálními územími. Abych zjistil, zda se tento úsek zachová při nízké hodnotě atributu `tolerance` ($=1$), použil jsem operátor TOUCH, viz kapitola 5.5. Ukázalo se, že ani při takovéto hodnotě atributu `tolerance` se topologické vztahy po generalizaci nezachovávají. Ty by se zachovaly pouze v případě, kdy bych jako hodnotu atributu `tolerance` volil **0.005**, což je hodnota podobnosti dat kterou jsem definoval v příslušných metadatech.

V tabulce 7 jsou uvedeny výsledky kontroly validity generalizovaných geometrických popisů pro různé hodnoty parametru `tolerance`.

NAZEV_KU	SIMPL_DUPLIC_1	SIMPL_DUPLIC_10	SIMPL_DUPLIC_50
Dolní Lhota u Klatov	TRUE	TRUE	TRUE
Lomec u Klatov	TRUE	13348 [Element <1>]	13348 [Element <1>]
Novákovice	TRUE	13348 [Element <1>]	13348 [Element <1>]
Vacovy	TRUE	TRUE	13348 [Element <1>]
Týnec u Janovic nad Úhlavou	TRUE	13348 [Element <1>]	13348 [Element <1>]

Tab. 7: Kontrola validity dat po aplikování funkce SDO_UTIL.REMOVE_DUPLICATE_VERTICES.

Z údajů v tabulce 7 vyplývá, že pro hodnotu parametru `tolerance` rovno **1** nedojde k porušení validity dat. Hodnota 13348 značí číslo chybové zprávy, konkrétně se jedná o problém, kdy v popisu polygonu není uzavřena jeho hranice. Se zvyšující se hodnotou parametru `tolerance` tak dochází ke ztrátě validity geometrického popisu katastrální hranice.

6.2 Testování funkce SDO_UTIL.SIMPLIFY

Funkce SDO_UTIL.SIMPLIFY je v Spatial určena přímo ke generalizačním účelům. Funkce realizuje Douglas-Peuckerův generalizační algoritmus za účelem redukce počtu vrcholů v geometrickém popisu prostorového objektu.

Podle [11] se po aplikování této funkce nemusí zachovat topologické vztahy, jaké byly mezi původními geometrickými popisy. Funkci jsem otestoval pro dvě různé hodnoty parametru `threshold`, jejíž geometrický význam je znázorněn na obrázku 5 v kapitole 3.7.2. Tabulka 8 obsahuje informace o počtu vrcholů v původním a zjednodušených geometrických popisech.

NAZEV_KU	PUVODNI_POPIS	SIMPLIFY_20	SIMPLIFY_50
Dolní Lhota u Klatov	463	30	13
Lomec u Klatov	306	30	18
Novákovice	387	30	14
Vacovy	328	37	20
Týnec u Janovic nad Úhlavou	516	31	6

Tab. 8: Počet vrcholů v geometrických popisech hranice kat. území po aplikování funkce SDO_UTIL.SIMPLIFY.

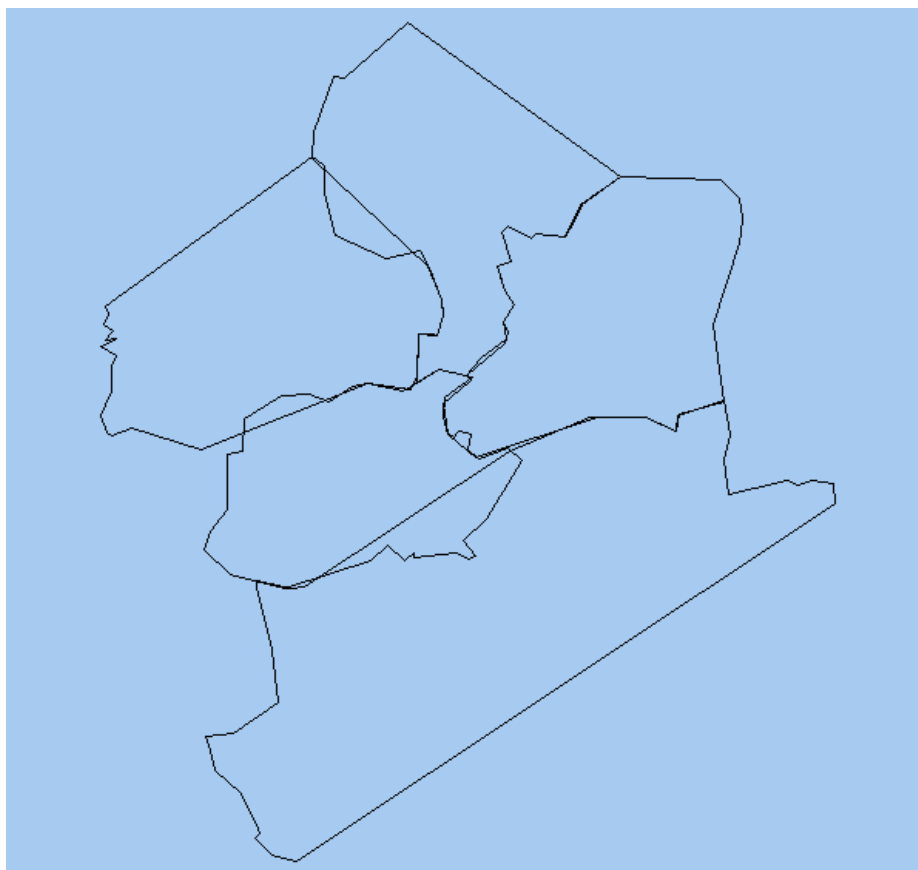
Je patrné, že s rostoucí hodnotou parametru `threshold` dochází k úbytku počtu vrcholů v geometrickém popisu prostorového prvku. Počet vrcholů ve výstupním popisu by se rovnal počtu vrcholů ve vstupním popisu, pokud bych za hodnotu parametru `threshold` volil hodnotu **0.005**.

Tabulka 9 obsahuje informace o validitě jednotlivých popisů pro dvě různé hodnoty vstupního parametru `threshold`.

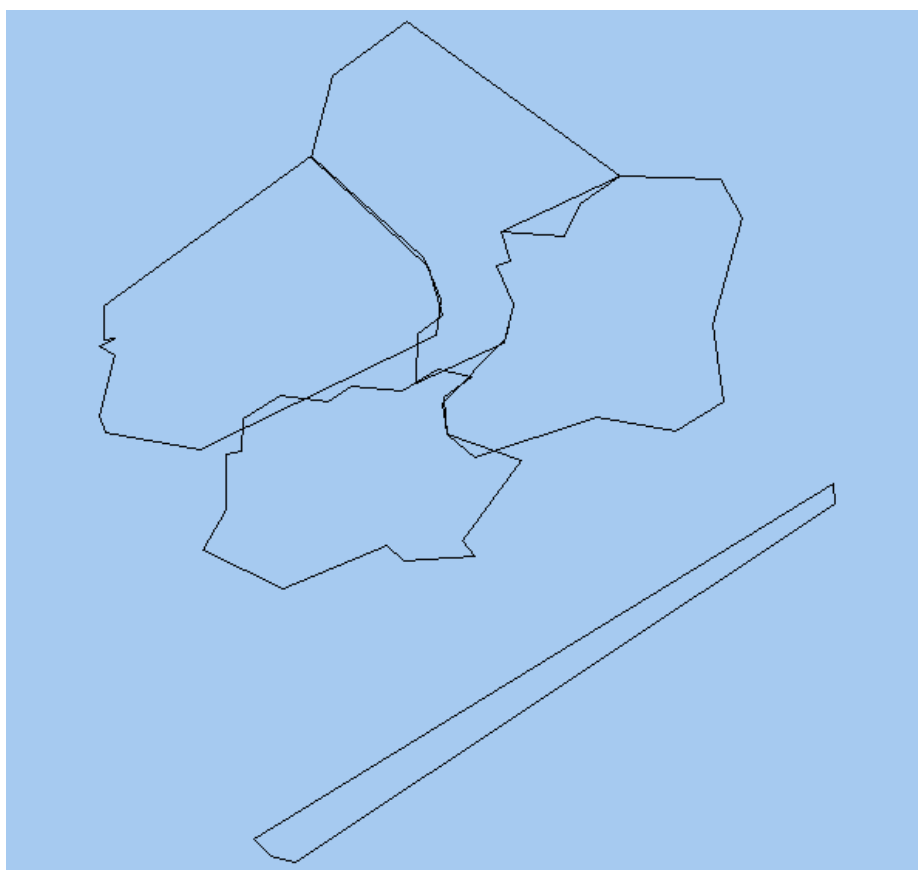
NAZEV_KU	SIMPLIFY_20	SIMPLIFY_50
Dolní Lhota u Klatov	TRUE	TRUE
Lomec u Klatov	TRUE	TRUE
Novákovice	TRUE	TRUE
Vacovy	TRUE	TRUE
Týnec u Janovic nad Úhlavou	TRUE	TRUE

Tab. 9: Kontrola validity dat po aplikování funkce SDO_UTIL.SIMPLIFY.

Obrázky 15 a 16 zobrazují výsledné geometrické popisy katastrálních hranic pro hodnotu parametru `threshold` rovno **20**, respektive **50**. Z obrázků je patrné, že při redukci počtu vrcholů geometrických popisů při použití této funkce dojde k silnému porušení topologických vztahů sousedních katastrálních území.



Obr. 15: Hodnota parametru `treshold` rovno **20**.



Obr. 16: Hodnota parametru `treshold` rovna **50**.

6.3 Testování funkce SDO_SAM.SIMPLIFY_GEOMETRY

Funkce SDO_SAM.SIMPLIFY_GEOMETRY je druhou funkcí v prostředí Spatial, jenž má primárně sloužit k redukci počtu vrcholových bodů v geometrickém popisu prostorového objektu. Při své realizaci volá funkci SDO_UTIL.SIMPLIFY a dbá na to, aby rozdíl ploch, které uzavírají původní geometrický popis a generalizovaný popis, vůči ploše původního popisu, nepřesáhl hodnotu definovanou (v procentech) v parametru `pct_area_change_limit`, viz kapitola 3.8.1.

Parametr `pct_area_change_limit` je klíčový pro možnost ovlivnit chování funkce SDO_SAM.SIMPLIFY_GEOMETRY a ovlivnění výsledného geometrického popisu. Funkci jsem otestoval celkem pro tři hodnoty vstupního parametru `pct_area_change_limit`. Implicitně je hodnota tohoto parametru rovna **2**, já pro otestování funkce volil hodnoty **5**, **10** a **20**. Hodnotu parametru `tol` jsem definoval rovno **0.005**, abych zachoval podrobnost dat určenou při definování příslušných metadat (o sloupci `hranice` v tabulce KATASTRALNI_HRANICE_MAT).

Při testování funkce SDO_SAM.SIMPLIFY_GEOMETRY jsem porovnal počet vrcholů původního a zjednodušeného popisu, splnění podmínky určené parametrem `pct_area_change_limit`, validitu výsledného popisu a zachování identity společného úseku katastrální hranice mezi dvěma sousedními katastrálními územími.

NAZEV_KU	PUVODNI_POPIS	SIMPL_GEOM_5	SIMPL_GEOM_10	SIMPL_GEOM_20
Dolní Lhota u Klatov	463	463	7	7
Lomec u Klatov	306	7	7	7
Novákovice	387	387	387	9
Vacovy	328	8	8	8
Týnec u Janovic nad Úhlavou	516	516	516	516

Tab. 10: Počet vrcholů v geometrických popisech hranice kat. území po aplikování funkce SDO_SAM.SIMPLIFY_GEOMETRY.

Z tabulky 10 lze mimo jiné vyčíst, že i pro hodnotu parametru `pct_area_change_limit` rovno **20** zůstává počet vrcholů geometrického popisu katastrální hranice území Týnec u Janovic nad Úhlavou nezměněn. Z dostupných materiálů se mi nepodařilo zjistit, zda tato skutečnost souvisí například s počtem vrcholových bodů vstupního geometrického popisu.

NAZEV_KU	PUVODNI_PLOCHA	SIMPL_GEOM_5	SIMPL_GEOM_10	SIMPL_GEOM_20
Dolní Lhota u Klatov	2277824.28082275	2277824.28082275	2237390.23724365	2237390.23724365
Lomec u Klatov	2024729.46008301	1958382.62854004	1958382.62854004	1958382.62854004
Novákovice	1579022.86962891	1579022.86962891	1579022.86962891	1821871.57330322
Vacovy	1644822.36480713	1694394.73468018	1694394.73468018	1694394.73468018
Týnec u Janovic nad Úhlavou	5568472.08557129	5568472.08557129	5568472.08557129	5568472.08557129

Tab. 11: Plochy (v m²), které uzavírají originální a generalizované geometrické popisy při různých hodnotách parametru `pct_area_change_limit`.

V tabulce 11 jsou uvedeny vypočtené plochy, které uzavírají jednotlivé geometrické popisy, pomocí funkce SDO_GEOM.SDO_AREA,

kteřá je k výpočtu plochy v Spatial k dispozici. Ukázalo se, že vypočtená plocha u jednotlivých území alespoň částečně reflektuje volbu vstupního parametru `pct_area_change_limit`, viz tabulka 12. Jediným „problémovým“ územím v tomto směru byl opět geometrický popis katastrální hranice území Týnec u Janovic nad Úhlavou, který jako jediný pro testovaná katastrální území obsahoval více než 999 vrcholových bodů.

Katastrální území	SIMPL_GEOM_5	SIMPL_GEOM_10	SIMPL_GEOM_20
Dolní Lhota u Klatov	+0,00%	-1,77%	-1,77%
Lomec u Klatov	-3,27%	-3,27%	-3,27%
Novákovice	+0,00%	+0,00%	+15,37%
Vacovy	+3,01%	+3,01%	+3,01%
Týnec u Janovic nad Úhlavou	+0,00%	+0,00%	+0,00%

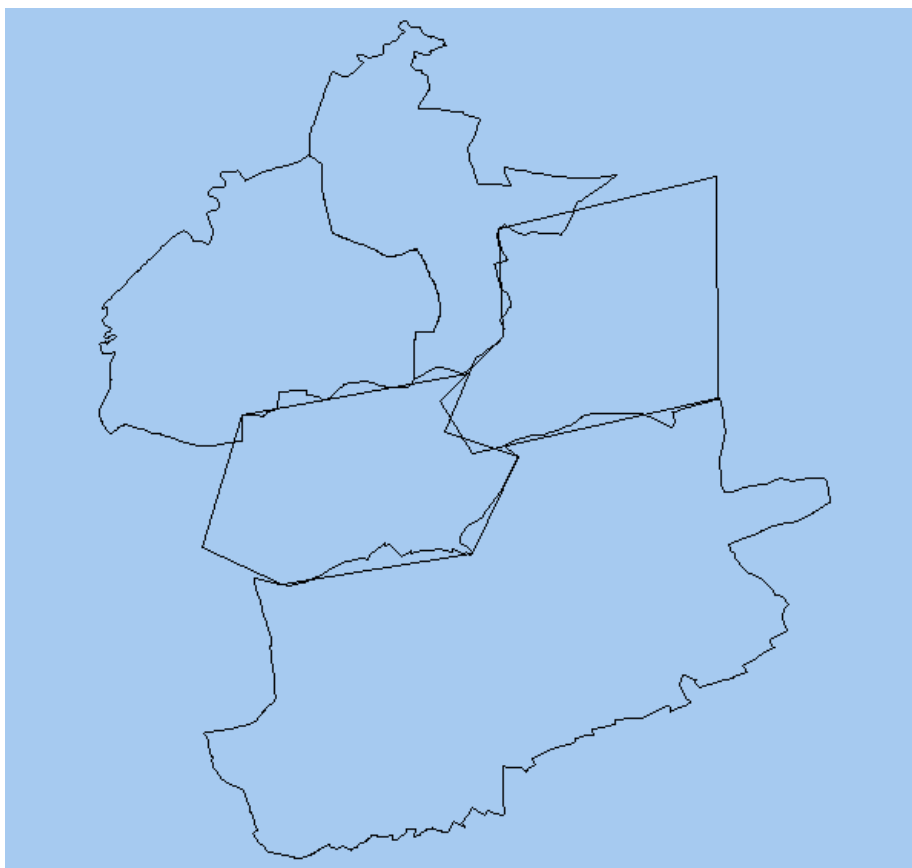
Tab. 12: Procentuální přírůstky u generalizovaných ploch vůči ploše původního popisu

Z údajů v tabulce 13 vyplývá, že všechny geometrické popisy, které jsou výsledkem funkce `SDO_SAM.SIMPLIFY_GEOMETRY`, jsou validní.

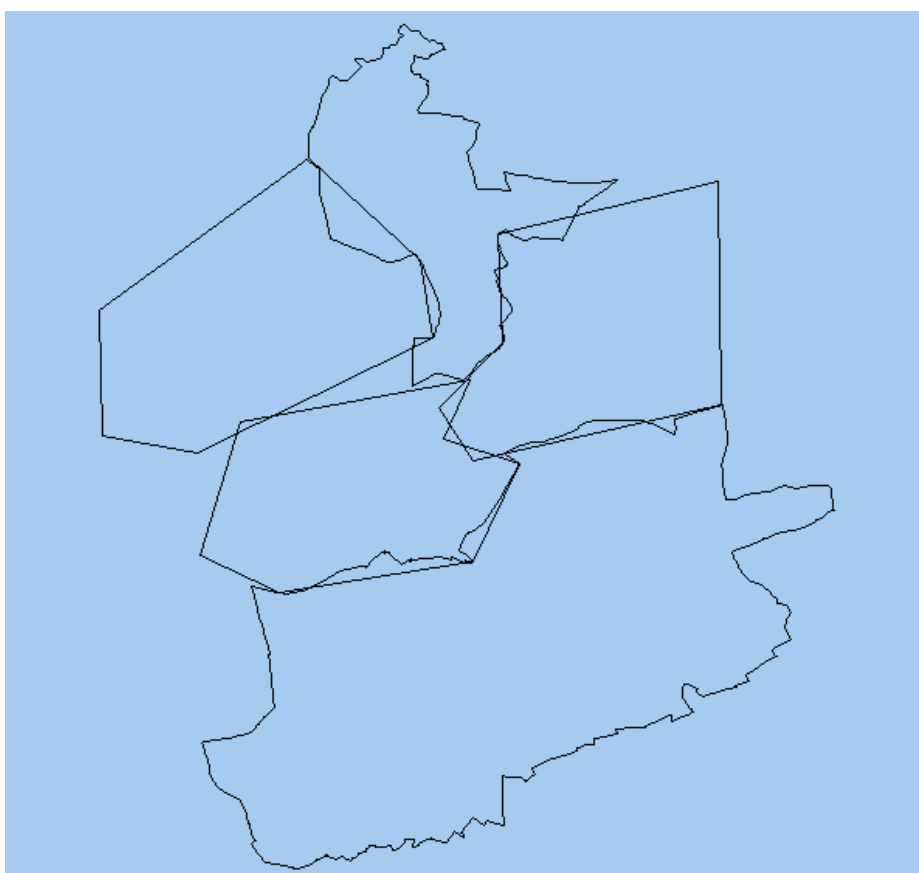
NAZEV_KU	SIMPL_GEOM_5	SIMPL_GEOM_10	SIMPL_GEOM_20
Dolní Lhota u Klatov	TRUE	TRUE	TRUE
Lomec u Klatov	TRUE	TRUE	TRUE
Novákovice	TRUE	TRUE	TRUE
Vacovy	TRUE	TRUE	TRUE
Týnec u Janovic nad Úhlavou	TRUE	TRUE	TRUE

Tab. 13: Kontrola validity dat po aplikování funkce `SDO_SAM.SIMPLIFY_GEOMETRY`.

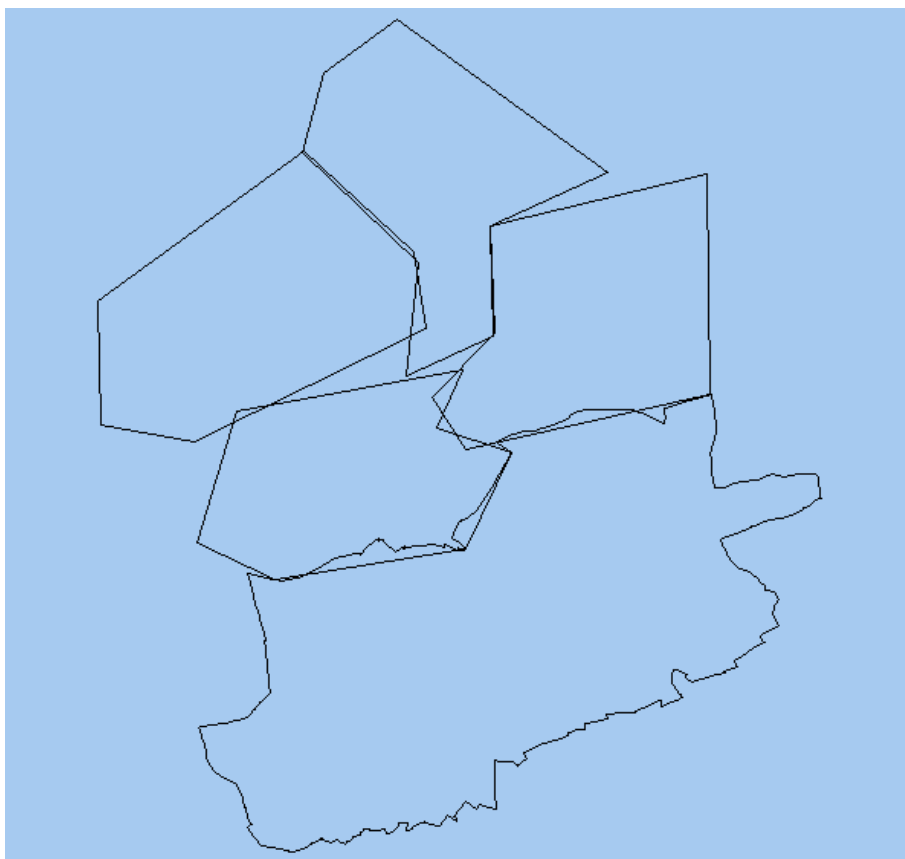
Obrázky 17, 18 a 19 zobrazují výsledné geometrické popisy katastrálních hranic pro hodnotu parametru `pct_area_change_limit` rovno **5**, **10**, respektive **20**. Z obrázků je patrné, že při redukci počtu vrcholů geometrických popisů při použití této funkce dojde k silnému porušení topologických vztahů sousedních katastrálních území.



Obr. 17: Hodnota parametru `pct_area_change_limit` rovna **5**.



Obr. 18: Hodnota parametru `pct_area_change_limit` rovna **10**.




Obr. 19: Hodnota parametru `pct_area_change_limit` rovna **20**.

6.4 Testování funkce `SDO_SAM.SIMPLIFY_LAYER`

Funkce `SDO_SAM.SIMPLIFY_LAYER` při svém provádění volá funkci `SDO_SAM.SIMPLIFY_GEOMETRY` pro každý záznam ve sloupci typu `SDO_GEOMETRY`, který je jedním ze vstupních parametrů funkce, viz kapitola 3.8.2.

Funkci `SDO_SAM.SIMPLIFY_LAYER` jsem otestoval pro hodnotu parametru `pct_area_change_limit` rovno 10 a z popsaného principu chování funkce v [11] jsem očekával, že obdržím stejné výsledky jako u sloupce `simpl_geom_10`, v němž jsou uloženy geometrické popisy vzniklé aplikováním funkce `SDO_SAM.SIMPLIFY_GEOMETRY` s použitou hodnotou parametru `pct_area_change_limit` rovno 10.

 NAZEV_KU	PUVODNI_POPIS	SIMPL_LAYER_10
 Dolní Lhota u Klatov	463	463
Lomec u Klatov	306	306
Novákovice	387	387
Vacovy	328	328
Týnec u Janovic nad Úhlavou	516	516

Tab. 14: Počet vrcholů v geometrických popisech hranice kat. území po aplikování funkce `SDO_SAM.SIMPLIFY_LAYER`.

Z tabulky 14 je patrné, že v generalizovaném popisu u žádné z hranic nedošlo ke změně oproti původnímu geometrickému popisu katastrální hranice, což je v rozporu s dosaženými výsledky popsány v kapitole 6.3. Když jsem na oficiálním internetovém fóru společnosti Oracle [14] zjišťoval možné vysvětlení, dozvěděl jsem se, že může jít o takzvaný „bug²⁶“.

Pokud by mělo dojít k zjednodušení geometrického popisu pomocí požadavku na poměr generalizované a původní plochy, doporučoval bych použití funkce SDO_SAM.SIMPLIFY_GEOMETRY namísto funkce SDO_SAM.SIMPLIFY_LAYER.

²⁶ bug (veš, brouk). Chyba v programu, objevená často až příliš pozdě. Původ slova spočívá v prehistorických dobách computing, kdy byly údajně vady a zkratky prvních sálových počítačů skutečně způsobeny hmyzem, který v jejich útrokách hledal úkryt, který se po spuštění uškvařil a způsobil krátká spojení.

7 Využití topologického datového modelu

V kapitole 3.3.1 byl zmíněn objektově-relační způsob uložení prostorových dat, využívající objektový datový typ `SDO_GEOMETRY`. Uložení dat získaných ze souboru výměnného formátu, jejich konverze do datového typu `SDO_GEOMETRY` a testování vybraných funkcí nad tímto datovým typem bylo hlavním předmětem mé diplomové práce.

Získané a uvedené výsledky v kapitole 6 ukazují, že uložení prostorových dat v typu `SDO_GEOMETRY`, v mém případě katastrální hranice v atributu `hranice`, není pro práci s tímto typem při využití generalizačních funkcí nabízených v Oracle Spatial 10g¹ příliš vhodné. Zachování topologických vztahů mezi generalizovanými geometrickými popisy je hlavním požadavkem na způsob uložení dat a funkcionalitu výpočetního prostředí operujícího s těmito daty.

Možným řešením uvedeného problému se ztrátou topologických vztahů je přechod na topologický datový model. Zmíním zde pouze základní informace o tomto modelu a situace ohledně funkcionality pro provádění generalizačních operací nad tímto způsobem uložení dat.

7.1 Základní koncept topologického modelu v Spatial

Topologický datový model umožňuje popsat prostorové prvky pomocí uzlů, hran a ploch, tzv. topologických elementů. Spatial umožňuje provádět prostorové operace nad jednotlivými elementy. Podrobné informace o topologickém modelu lze nalézt v [12].

7.1.1 Uzel (Node)

Uzel, reprezentovaný bodem, může být izolovaný či neizolovaný. V neizolovaném uzlu se schází dvě a více hran, přičemž prostorová lokalizace uzlu je dána jeho souřadnicemi. Příklady prostorových prvků, které mohou být reprezentovány bodem, jsou například počáteční či koncový bod ulice, historicky zajímavé místo či letiště (samozřejmě v mapě příslušného měřítka).

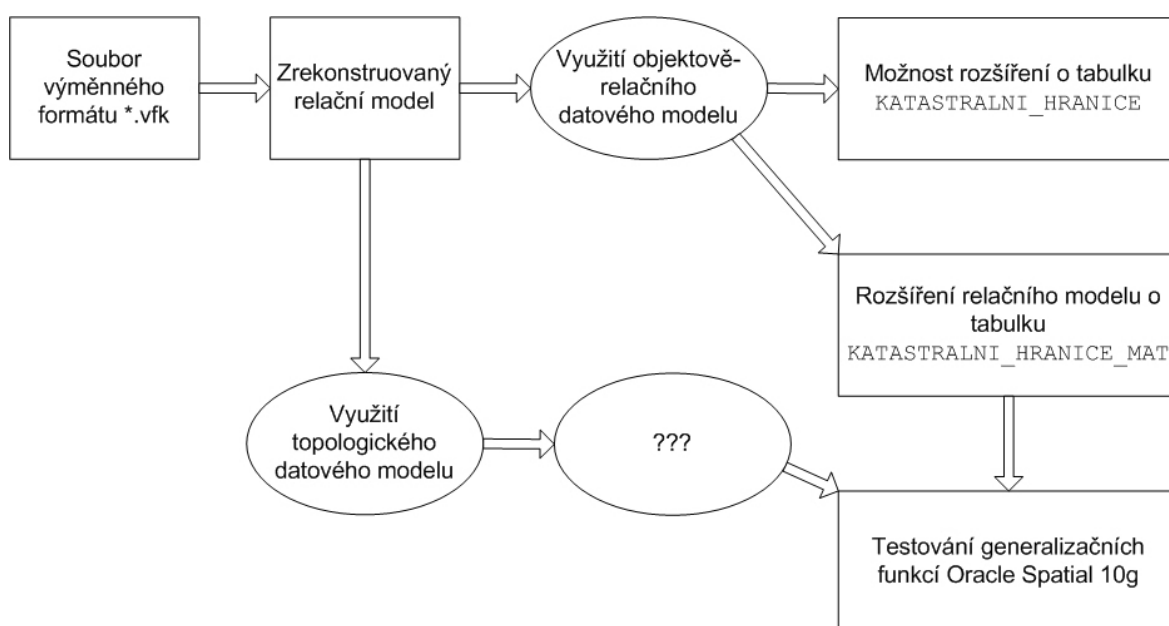
7.1.2 Hrana (Edge)

Hrana je ohraničena dvěma uzly: počátečním a koncovým. S hranou je obvykle spojený řetězec souřadnic, který popisuje prostorovou reprezentaci hrany. Hrana může obsahovat více vrcholů pospojovaných liniovými řetězci, kruhovými oblouky či jejich vzájemnou kombinací. Příkladem prostorových prvků, které mohou být reprezentovány hranou, jsou například řeky či segmenty ulic.

Pořadí souřadnic udává směr (*direction*) hrany. Směr je důležitý při určování topologických vztahů.

7.1.3 Plocha (*Face*)

Plocha, reprezentovaná polygonem, má odkaz na jednu orientovanou hranu na jejím vnějším obvodu. Příkladem prostorových prvků, které mohou být reprezentovány plochou, jsou například parky či státy.



Obr. 20: Schéma postupu od souboru výměnného formátu k testování generalizačních funkcí.

Na obrázku 13 je znázorněno začlenění využití topologického datového modelu k možnosti uložení prostorových dat a případného testování generalizačních funkcí nad tímto způsobem uložení dat.

Problémem v tomto okamžiku zůstává verze databáze Oracle a jejího rozšíření Spatial, kterou mám v době vzniku této práce k dispozici. Společnost Oracle vyvinula topologický model uložení dat pro řešení otázek podobné té, kdy se snažím docílit zachování topologických vztahů generalizovaných geometrických popisů prostorových objektů. Tento vývoj je dynamický, s rostoucí verzí Spatial roste i funkcionality pro operace nad topologickým modelem. Ve verzi 10.1.0.4, která bude uvolněna na trh zatím neznámo kdy, by měla být možnost snadného převodu z uložení dat v objektu *SDO_GEOMETRY* do typu *SDO_TOPO_GEOMETRY*²⁷, ovšem přímá podpora generalizačních funkcí bude zřejmě pro tento způsob uložení dat chybět i nadále. Jistou cestou tak zůstává naprogramování si funkcí vlastních.

²⁷*SDO_TOPO_GEOMETRY* je hlavním datovým typem spojeným s topologickým datovým modelem, slouží k popisu topologie objektu.

8 Závěr

Hlavním cílem práce mělo být otestování a popsání možností uložení prostorových dat pomocí zvolené datové struktury v Oracle Spatial a funkcionality pro provádění generalizačních postupů.

Uložení dat v prostředí Spatial je v současné době možné provést s využitím několika hlavních přístupů. Nejvíce podporovaným přístupem je zřejmě objektově-relační způsob uložení, pro nějž je k dispozici rozsáhlá množina procedur, funkcí a prostorových operátorů. V práci jsem se zabýval právě tímto přístupem, popsal jsem způsob převodu dat ze souboru výměnného formátu do navrženého objektově-relačního modelu s využitím objektového datového typu *SDO_GEOMETRY*. Výhody použití tohoto typu spatřuji v:

- možnosti ukládat prostorová data společně s atributovými v jednom řádku a sloupci tabulky.
- rozsáhlé množině předprogramovaných procedur a funkcí podporujících tento typ v prostředí Spatial.

Nad daty uloženými pomocí typu *SDO_GEOMETRY* lze aplikovat funkce, respektive proceduru, které jsou v prostředí Spatial ke generalizačním účelům primárně určeny, či lze „generalizačního“ průběhu vhodně zvolené funkce dosáhnout přiměřenou volbou vstupního parametru. Toto je ovšem poměrně nešťastný krok, ukázalo se, že takovéto obcházení nemá pro konkrétní účely reálný význam. Ovšem i v případě čistě generalizačních funkcí, pracujících nad typem *SDO_GEOMETRY*, nejsou získané výsledky příliš optimistické. Na první pohled je z vykreslených zjednodušených geometrických popisů katastrálních hranic vidět, že nedojde k zachování topologických vztahů, které se vyskytovaly mezi popisy původními.

Ovšem, i kdyby se tyto otestované funkce chovaly z hlediska požadovaného výstupu korektně, nechalo by se určitě použití tohoto přístupu optimalizovat:

- Vylepšením by jistě bylo nadefinování souřadnicového systému S-JTSK a uložení příslušných hodnot do systémové tabulky *MDSYS.CS_SRS*.
- Rovněž by bylo určitě vhodné mít databázi ve stavu, při kterém by existovala tabulka se souřadnicemi bodů, na kterou by se pouze odkazovalo z tabulek jiných. To znamená, místo přímého ukládání souřadnic geometrického popisu prvku do atributu *sdo_ordinates_array* jak je vyžadováno nyní, by se ukládaly pouze identifikátory alias odkazy na příslušné body.
- Možným vylepšením by jistě bylo i rozšíření funkcionality o další funkce, které by realizovali postupy kartografické generalizace,

neboť jak se ukázalo, současné možnosti Oracle Spatial 10g nejsou v této oblasti příliš široké.

Možným řešením otázky zachování topologických vztahů mezi generalizovanými geometrickými popisy prvků by bylo využití druhého přístupu – přechod na topologický datový model. Ten byl vyvinut právě za účelem požadavků zachování topologických vztahů mezi zjednodušenými geometrickými popisy prvků, ovšem rešerží dostupných informací a zdrojů se dojde k závěru, že ani pro tento model není v Spatial v současné době potřebná funkcionality zakomponována. Výhodou využití topologického datového modelu by bylo zefektivnění uložení dat, souřadnice každého bodu katastrální hranice či obecně jakéhokoli objektu by mohla být uložena pouze jednou, nedocházelo by k výskytu redundantních hodnot, jako v případě využití objektového typu *SDO_GEOMETRY*. Zápor by ale zřejmě byla větší komplikovanost při údržbě a aktualizaci dat.

Oba dva zmíněné přístupy uložení prostorových dat v prostředí Spatial mají tedy své výhody i nevýhody. Otázkou zůstává, zda firma Oracle bude do budoucna rozšiřovat generalizační funkcionality. Jistotu bychom získali v případě, že bychom se rozhodli naprogramovat vlastní funkce realizující postupy kartografické generalizace.

Seznam použitých zkratek

ABGI	Advisory Board on Geographic Information
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ČSN	Česká státní norma
ČÚZK	Český úřad zeměměřický a katastrální
EU	Evropská Unie
EUROGI	European Umbrella Organisation for Geographic Information
GDI	Geographic Data Infrastructure
GI	Geographical Information
GII	Geospatial Information Infrastructure
GINIE	Geographic Information Network in Europe
GIS	Geografický informační systém
GML	Geography Markup Language
INSPIRE	Infrastructure for Spatial Information in Europe
IP	Internet Protocol
ISKN	Informační systém katastru nemovitostí
ISO	International Standards Organization
IST	Information Society Technology
JDBC	Java Database Connectivity
KN	Katastr nemovitostí
MBR	Minimum Bounding Rectangle
ODBC	Open Database Connectivity
OGC	Open GIS Consortium
PII	Prostorová informační infrastruktura
PL/SQL	Procedural Language for Structured Query Language
S-JTSK	Systém Jednotné trigonometrické sítě katastrální
SQL	Structured Query Language
VF	Výměnný formát

Literatura

- [1] MILDORF, T.: *Úvodní studie katalogu geoprvků pro ISVS*. Plzeň 2004.
- [2] INSPIRE Architecture And Standards Position Paper
[ONLINE]. <http://www.ec-gis.org/inspire/>
- [3] PAUKNEROVÁ, E.: *GINIE – Evropský projekt*. Článek z časopisu GEOinformace. 1/2003
[ONLINE]. <http://www.geoinformace.cz/>
- [4] GINIE Spatial Data Infrastructures: Recommendations for Action.
[ONLINE]. <http://www.ec-gis.org/ginie>
- [5] Slaboch, V.: *Standardizace na poli informačních systémů*. Referát ze sborníku konference MGID'99. Brno 1999.
- [6] Open GIS Consortium: *The OGC Technical Committee Policies & Procedures - OpenGIS Project Document 00-023r2*. 15.5.2005.
- [7] Open GIS Consortium, Inc.: *OpenGIS Simple Features Specification For SQL – OpenGIS Project Document 99-049*. 5.5.1999.
- [8] Douglas, D., Peucker, T.: *Algorithms for the reduction of the number of points required to represent a digitised line or its caricature*. The Canadian Cartographer. 1973.
- [9] ŠÍMA, J.: *Geoinformační terminologie pro geodety a kartografy*. Výzkumný ústav geodetický, topografický a kartografický. Zdiaby 2003.
- [10] Český úřad zeměměřický a katastrální: *Struktura výměnného formátu informačního systému katastru nemovitostí České republiky*. Praha 26.11.2004.
- [11] Murray, CH.: *Oracle Spatial User's Guide and Reference, 10g Release*. 1. říjen 2003
[ONLINE].
http://www.oracle.com/technology/products/spatial/spatial_10g_doc_index.html
- [12] Murray, CH.: *Topology and Network Data Models*. Říjen 2003
[ONLINE].
http://www.oracle.com/technology/products/spatial/spatial_10g_doc_index.html
- [13] Lacko, L.: *Oracle - Správa, programování a použití databázového systému*. Computer Press. Praha 2002.
- [14] -----: *Oficiální diskuzní forum o Oracle Spatial*
[ONLINE]. <http://forums.oracle.com/forums/>
- [15] Šimůnek, M.: *SQL – Kompletní kapesní průvodce*. Grada Publishing. Praha 1999.
- [16] Radimský, M.: *Standardy a interoperabilita v GIS*
[ONLINE]. <http://www.fce.vutbr.cz/veda/>
- [17] Tuček, J.: *Geografické informační systémy – Principy a praxe*. Computer Press. Praha 1998.

- [18] Kvoch, M.: *Programování v Turbo Pascalu 7.0*. KOPP. České Budějovice 1993.
- [19] Čada, V., Hojdar, J., Martinek, M.: *Základní datové báze geodat*. In: Sborník 10. konference GIS Ostrava. Ostrava 2003
[ONLINE]. <http://gis.vsb.cz>

Přílohy

Příloha A: Operátory v Oracle Spatial 10g

Příloha B: PL/SQL podprogramy v balíku SDO_GEOM v Oracle Spatial 10g

Příloha C: Agregační funkce v Oracle Spatial 10g

Příloha D: Ladící (tuning) podprogramy v Oracle Spatial 10g

Příloha E: Utility v Oracle Spatial 10g

Příloha F: Analyzující podprogramy v Oracle Spatial 10g

Příloha G: Podprogramy v balíku SDO_TOPO v Oracle Spatial 10g

Příloha H: Podprogramy v balíku SDO_TOPO_MAP v Oracle Spatial 10g

Tablka A-1 obsahuje seznam základních operátorů, které lze v prostředí Spatial využít při práci s objektovým datovým typem *SDO_GEOMETRY*.

Operátor	Popis
SDO_FILTER	Specifikuje, které geometrické popisy prvků mohou mít prostorový vztah (<i>interact</i>) s daným geometrickým popisem prvku.
SDO_JOIN	Provádí prostorové spojení založené na jednom nebo více topologických vztazích.
SDO_NN	Určuje nejbližší geometrický popis prvku k danému geometrickému popisu prvku.
SDO_NN_DISTANCE	Vrací vzdálenost geometrického popisu prvku vrácené operátorem SDO_NN.
SDO_RELATE	Určuje, zda spolu dva geometrické popisy prvků prostorově souvisí požadovaným způsobem.
SDO_WITHIN_DISTANCE	Určuje, zda jsou dva geometrické popisy prvků ve specifikované vzdálenosti jedna od druhé.

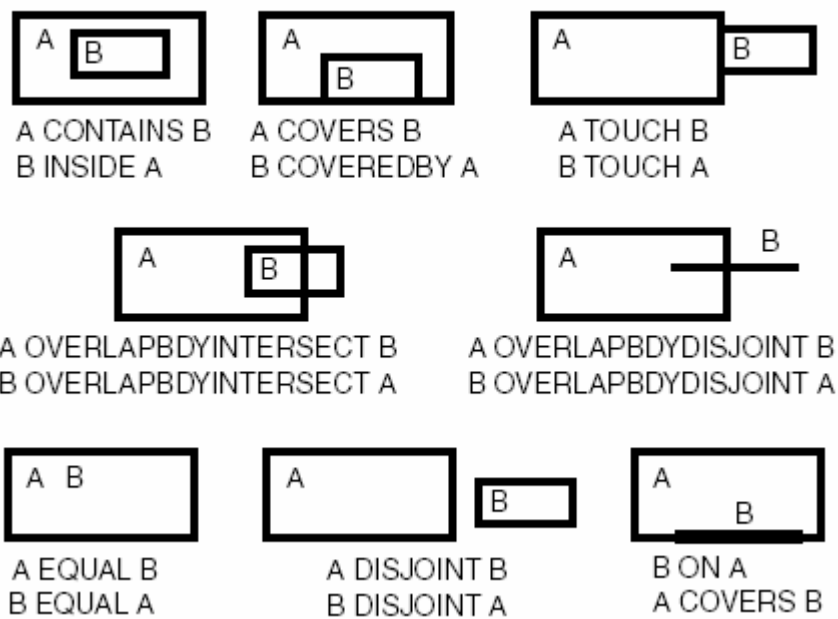
Tab. A-1: Základní prostorové operátory v Oracle Spatial

Tabulka A-2 obsahuje seznam operátorů, které lze použít pro operace typu SDO_RELATE.

Operátor	Popis
SDO_ANYINTERACT	Ověřuje, zda některý z geometrických popisů prvků v tabulce má topologický vztah typu ANYINTERACT se specifikovaným geometrickým popisem prvku.
SDO_CONTAINS	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu CONTAINS se specifikovaným geometrickým popisem prvku.
SDO_COVEREDBY	Ověřuje, zda nějaké geometrické popisy prvků v tabulce mají topologický vztah typu COVEREDBY se specifikovaným geometrickým popisem prvku.

Operátor	Popis
SDO_COVERS	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu COVERS se specifikovaným geometrickým popisem prvku.
SDO_INSIDE	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu INSIDE se specifikovaným geometrickým popisem prvku.
SDO_ON	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu ON se specifikovaným geometrickým popisem prvku.
SDO_OVERLAPBDYDISJOINT	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu OVERLAPBDYDISJOINT se specifikovaným geometrickým popisem prvku.
SDO_OVERLAPBDYINTERSECT	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu OVERLAPBDYINTERSECT se specifikovaným geometrickým popisem prvku.
SDO_OVERLAPS	Ověřuje, zda některé geometrické popisy prvků v tabulce překrývají (to znamená mají topologický vztah typu OVERLAPBDYDISJOINT nebo OVERLAPBDYINTERSECT) se specifikovaným geometrickým popisem prvku.
SDO_TOUCH	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu TOUCH se specifikovaným geometrickým popisem prvku.
SDO_EQUAL	Ověřuje, zda některé geometrické popisy prvků v tabulce mají topologický vztah typu EQUAL se specifikovaným geometrickým popisem prvku.

Tab. A-2: Vhodné operátory pro operace SDO_RELATE



Obr. A-1: Ukázka topologických vztahů mezi dvěmi geometrickými popisy prvků

PL/SQL podprogramy v balíku SDO_GEOM mohou být seskupeny do následujících kategorií:

- Vztah (true/false) mezi dvěma geometrickými popisy prvků:
 - RELATE
 - WITHIN_DISTANCE
- Validace:
 - VALIDATE_GEOMETRY_WITH_CONTEXT
 - VALIDATE_LAYER_WITH_CONTEXT
- Operace s jedním geometrickým popisem prvku:
 - SDO_ARC_DENSIFY
 - SDO_AREA
 - SDO_BUFFER
 - SDO_CENTROID
 - SDO_CONVEXHULL
 - SDO_LENGTH
 - SDO_MAX_MBR_ORDINATE
 - SDO_MBR
 - SDO_POINT_ON_SURFACE
- Operace se dvěma geometrickými popisy prvků:
 - SDO_DISTANCE
 - SDO_DIFFERENCE
 - SDO_INTERSECTION
 - SDO_UNION
 - SDO_XOR

Podprogram

Popis

SDO_GEOM.RELATE

Určí, jak spolu geometrické popisy prvků souvisí.

SDO_GEOM.SDO_ARC_DENSIFY

Změní každý kruhový oblouk v aproximaci sestávající z přímých linií a každou kružnici v polygon sestávající z přímých liniových segmentů.

SDO_GEOM.SDO_AREA

Spočte plochu dvojdimenzionálního polygonu.

SDO_GEOM.SDO_BUFFER

Generuje buffer vně nebo uvnitř geometrického popisu prvku.

Podprogram	Popis
SDO_GEOM.SDO_CENTROID	Vrací centroid polygonu.
SDO_GEOM.SDO_CONVEXHULL	Vrací objekt typu polygon, který reprezentuje konvexní obálku geometrického popisu prvku.
SDO_GEOM.SDO_DIFFERENCE	Vrací geometrický popis prvku, který je topologickým rozdílem (operace MINUS) dvou geometrických popisů prvků.
SDO_GEOM_DISTANCE	Počítá vzdálenost mezi dvěma geometrickými popisy prvků.
SDO_GEOM.SDO_INTERSECTION	Vrací geometrický popis prvku, který je topologickým průnikem (operace AND) mezi dvěma geometrickými popisy prvků.
SDO_GEOM.SDO_LENGTH	Počítá délku či obvod geometrického popisu prvku.
SDO_GEOM.SDO_MAX_MBR_ORDINATE	Vrací maximální hodnotu pro určitou dimenzi MBR geometrického popisu prvku.
SDO_GEOM.SDO_MBR	Vrací MBR geometrického popisu prvku.
SDO_GEOM.SDO_MIN_MBR_ORDINATE	Vrací minimální hodnotu pro určitou dimenzi MBR geometrického popisu prvku.
SDO_GEOM.SDO_POINTONSURFACE	Vrací bod, který náleží polygonu.

Podprogram	Popis
SDO_GEOM.SDO_UNION	Vrací geometrický popis prvku, který je topologickým spojením (operace OR) dvou geometrických popisů prvků.
SDO_GEOM.SDO_XOR	Vrací geometrický popis prvku, který je výsledkem operace XOR mezi dvěma geometrickými popisy prvků.
SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT	Určuje, zda je geometrický popis prvku validní.
SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT	Určuje, zda jsou všechny geometrické popisy prvků uložené ve sloupci <i>SDO_GEOMETRY</i> validní.
SDO_GEOM.WITHIN_DISTANCE	Určuje, zda jsou od sebe dva geometrické popisy prvků do určité vzdálenosti.

Funkce	Popis
SDO_AGGR_CENTROID	Vrací geometrický popis prvku, který je centroidem skupiny geometrických popisů prvků.
SDO_AGGR_CONVEXHULL	Vrací geometrický popis prvku, který je konvexní obálkou určených geometrických popisů prvků.
SDO_AGGR_MBR	Vrací MBR určených geometrických popisů prvků.
SDO_AGGR_UNION	Vrací geometrický popis prvku, který je výsledkem topologického sjednocení (operace OR) určených geometrických popisů prvků.

Podprogram	Popis
SDO_TUNE.AVERAGE_MBR	Počítá průměrný (střední) minimální ohraničující pravoúhelník (MBR) geometrických popisů prvků ve vrstvě.
SDO_TUNE.ESTIMATE_RTREE_INDEX_SIZE	Odhaduje maximální počet paměťového prostoru pro R-tree prostorovou indexovou tabulku.
SDO_TUNE.EXTENT_OF	Vrací MBR dat ve vrstvě.
SDO_TUNE.MIX_INFO	Počítá informaci o geometrickém typu pro prostorovou vrstvu jako procentuální zastoupení každého geometrického typu.
SDO_TUNE.QUALITY_DEGRADATION	Vrací tzv. <i>quality degradation</i> indexu ^{D-1} .

^{D-1} *Quality degradation* je číslo udávající přibližnou dobu, o kterou bude déle trvat provedení I/O operací indexové části jakéhokoli dotazu se současným indexem v porovnání s vykonáním těchto operací toho samého dotazu, pokud bude vytvořen nový index. Například, pokud byl vytvořen nový index a I/O operace indexové části dotazu zaberou přibližně dvakrát více času, hodnota *quality degradation* je 2.

Podprogram	Popis
SDO_UTIL.APPEND	Vytvoří nový geometrický popis prvku přidáním jednoho geometrického popisu prvku k druhému.
SDO_UTIL.CIRCLE_POLYGON	Vrátí geometrický popis prvku typu polygon, který aproximuje a je vepsán určené kružnici.
SDO_UTIL.CONCAT_LINES	Vytvoří nový geometrický popis prvku sloučením dvou dvojdimenzionálních geometrických popisů prvků typu linie či multilinie.
SDO_UTIL.CONVERT_UNIT	Konvertuje hodnoty úhlu, plochy či vzdálenosti z jedné jednotky do druhé.
SDO_UTIL.ELLIPSE_POLYGON	Vrátí geometrický popis prvku typu polygon, který aproximuje a je vepsán určené elipse.
SDO_UTIL.EXTRACT	Vrátí geometrický popis prvku, který reprezentuje určený element vstupního geometrického popisu prvku.
SDO_UTIL.GETNUMELEM	Vrátí počet elementů ve vstupním geometrickém popise prvku.
SDO_UTIL.GETNUMVERTICES	Vrátí počet vrcholů ve vstupním geometrickém popise prvku.
SDO_UTIL.GETVERTICES	Vrátí souřadnice vrcholů vstupního

Podprogram	Popis
	geometrického popisu prvku.
SDO_UTIL.INITIALIZE_INDEXES_FOR_TTS	Inicializuje všechny prostorové indexy v tabulkovém prostoru, který byl transportován do jiné databáze.
SDO_UTIL.POINT_AT_BEARING	Vrátí geometrický popis prvku typu bod, který je v určené vzdálenosti a orientaci od počátečního bodu.
SDO_UTIL.POLYGONTOLINE	Konvertuje všechny polygonové elementy geometrického popisu prvku na liniové elementy, zároveň podle toho nastaví hodnotu sdo_gtype .
SDO_UTIL.PREPARE_FOR_TTS	Připraví tabulkový prostor k transportu do jiné databáze tím, že během operace transportu „konzervuje“ prostorové indexy.
SDO_UTIL.REMOVE_DUPLICATE_VERTICES	Odstraní redundantní vrcholy z geometrického popisu prvku.
SDO_UTIL.REVERSE_LINestring	Vrátí geometrický popis prvku typu linie s vrcholy v opačném pořadí, než má vstupní geometrický popis prvku typu linie.
SDO_UTIL_SIMPLIFY	Zjednoduší vstupní geometrický popis prvku použitím Douglas-Peuckerova

Podprogram

Popis

SDO_UTIL.TO_GMLGEOMETRY

generalizačního
algoritmu.

Konvertuje geometrický
popis prvku z prostředí
Oracle Spatial do GML
2.0 (Geography Markup
Language).

Podprogram	Popis
SDO_SAM.AGGREGATES_FOR_GEOMETRY	Spočítá pro geometrický popis prvku tematický souhrn.
SDO_SAM.AGGREGATES_FOR_LAYER	Spočítá tematicky souhrn pro geometrické popisy prvků ve vrstvě.
SDO_SAM.SIMPLIFY_GEOMETRY	Generalizuje geometrický popis prvku.
SDO_SAM.SIMPLIFY_LAYER	Generalizuje vrstvu geometrických popisů prvků.
SDO_SAM.TILED_BINS	„Dlaždicuje“ dvojdimenzionální prostor a vrací geometrické popisy prvků odpovídající jednotlivým dlaždicím.

Podprogram	Popis
SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER	Přidá geometrickou vrstvu do existující topologie.
SDO_TOPO.CREATE_TOPOLOGY	Vytvoří topologii.
SDO_TOPO.DELETE_TOPO_GEOMETRY_LAYER	Odstraní geometrickou vrstvu z existující topologie.
SDO_TOPO.DROP_TOPOLOGY	Odstraní topologii, která byla vytvořena pomocí procedury SDO_TOPO.CREATE_TOPOLOGY.
SDO_TOPO.GET_FACE_BOUNDARY	Vrací seznam identifikátorů hran pro danou plochu.
SDO_TOPO.GET_TOPO_OBJECTS	Vrací pole objektů SDO_TOPO_OBJECT ^{G-1} , které souvisí s daným objektem.
SDO_TOPO.INITIALIZE_METADATA	Inicializuje metadata o topologii.

^{G-1} Popis objektu SDO_TOPO_OBJECT lze nalézt v [12].

Podprogram	Popis
SDO_TOPO_MAP.ADD_EDGE	Přidá hranu do topologie a vrátí její identifikátor.
SDO_TOPO_MAP.ADD_ISOLATED_NODE	Přidá izolovaný uzel do topologie a vrátí jeho identifikátor.
SDO_TOPO_MAP.ADD_LOOP	Přidá smyčku (linii začínající a končící na shodném bodě) a vrátí její identifikátor.
SDO_TOPO_MAP.ADD_NODE	Přidá bod do existující hrany a vrátí jeho identifikátor.
SDO_TOPO_MAP.CHANGE_EDGE_COORDS	Změní souřadnice a příslušné informace o hraně.
SDO_TOPO_MAP.CLEAR_TOPO_MAP	Smaže všechny objekty a změny ve vyrovnávací paměti spojené s objektem TopoMap ^{H-1} .
SDO_TOPO_MAP.CREATE_EDGE_INDEX	Vytvoří ve vyrovnávací paměti vnitřní R-tree index (nebo obnoví existující index) na hrany spojené s objektem TopoMap.
SDO_TOPO_MAP.CREATE_FACE_INDEX	Vytvoří ve vyrovnávací paměti vnitřní R-tree index (nebo obnoví existující index) na plochy spojené s objektem TopoMap.
SDO_TOPO_MAP.CREATE_TOPO_MAP	Vytvoří vyrovnávací paměť pro objekt asociovaný s topologií.

^{H-1} Objekt TopoMap je spojen s vyrovnávací pamětí asociované s topologií. Více v [12].

Podprogram	Popis
SDO_TOPO_MAP.GET_CONTAINING_FACE	Vrací identifikátor plochy, která obsahuje specifikovaný bod.
SDO_TOPO_MAP.GET_EDGE_ADDITIONS	Vrací pole obsahující identifikátory hran, které byly přidány do objektu TopoMap.
SDO_TOPO_MAP.GET_EDGE_CHANGES	Vrací pole obsahující identifikátory hran, které byly změněny (modifikovány) v objektu TopoMap.
SDO_TOPO_MAP.GET_EDGE_COORDS	Vrací pole obsahující souřadnice počátečního uzlu, mezilehlých bodů a koncového uzlu dané hrany.
SDO_TOPO_MAP.GET_EDGE_DELETIONS	Vrací pole identifikátorů hran, které byly smazány z objektu TopoMap.
SDO_TOPO_MAP.GET_EDGE_NODES	Vrací pole obsahující identifikátory počátečního a koncového uzlu dané hrany.
SDO_TOPO_MAP.GET_FACE_ADDITIONS	Vrací pole obsahující identifikátory ploch, které byly přidány do objektu TopoMap.
SDO_TOPO_MAP.GET_FACE_CHANGES	Vrací pole obsahující identifikátory ploch, které byly změněny (modifikovány) v objektu TopoMap.
SDO_TOPO_MAP.GET_FACE_DELETIONS	Vrací pole identifikátory ploch, které byly smazány z objektu TopoMap.

Podprogram	Popis
SDO_TOPO_MAP.GET_NEAREST_EDGE	Vrací identifikátor hrany, která je nejbližší k danému bodu.
SDO_TOPO_MAP.GET_NEAREST_NODE	Vrací identifikátor uzlu, který je nejbližší k danému bodu.
SDO_TOPO_MAP.GET_NODE_ADDITIONS	Vrací pole identifikátorů uzlů, které byly přidány do objektu TopoMap.
SDO_TOPO_MAP.GET_NODE_CHANGES	Vrací pole obsahující identifikátory uzlů, které byly změněny (modifikovány) v objektu TopoMap.
SDO_TOPO_MAP.GET_NODE_COORD	Vrací objekt typu SDO_POINT_TYPE se souřadnicemi specifikovaného uzlu.
SDO_TOPO_MAP.GET_NODE_DELETIONS	Vrací pole identifikátorů uzlů, které byly smazány z objektu TopoMap.
SDO_TOPO_MAP.GET_NODE_STAR	Vrací objekt SDO_NUMBER_ARRAY s identifikátory hran v pořadí směru chodu hodinových ručiček, které jsou spojeny k daným uzlem.
SDO_TOPO_MAP.MOVE_NODE	Přesune neizolovaný uzel a jeho připojené hrany.
SDO_TOPO_MAP.REMOVE_EDGE	Odstraní hranu z topologie.
SDO_TOPO_MAP.REMOVE_NODE	Odstraní uzel z topologie.

Podprogram

Popis

SDO_TOPO_MAP.UPDATE_TOPO_MAP

Aktualizuje topologii s ohledem na změny provedené v objektu TopoMap.

SDO_TOPO_MAP

Provede validaci objektu TopoMap.