

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra matematiky

## **Diplomová práce**

# **Modelování hierarchie topologicky strukturovaných dat**

Plzeň, 2012

Ing. Kateřina Figallová

## **Prohlášení**

Předkládám tuto diplomovou práci jako součást procesu ukončení studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou diplomovou práci vypracovala samostatně a použila jsem pouze podklady uvedené v příloženém seznamu. Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Českých Budějovicích dne 12. května 2012

.....

## **Poděkování**

Chtěla bych touto cestou poděkovat především Ing. Karlu Janečkovi, Ph.D., vedoucímu diplomové práce, za odborné vedení, rady a věnovaný čas. Dále děkuji Mgr. Miroslavu Matěšovi, aplikačnímu inženýrovi společnosti Bentley Systems ČR za poskytnuté odborné konzultace při práci v prostředí Bentley.

Dále děkuji své rodině za podporu, které se mi dostávalo po celou dobu mého studia.

## **Abstrakt**

Tato práce se zabývá modelováním hierarchie topologicky strukturovaných dat. Vstupními daty jsou data ve formátu SHP – vrstva okresů (okresy.shp), vrstva krajů (kraje.shp) a vrstva s Českou Republikou (cr.shp). Tato data jsou převedena do objektově relačního modelu SDO\_GEOMETRY v prostředí Bentley PowerMap, a následně převedena pomocí funkcí Oracle Spatial do topologického formátu SDO\_TOPO\_GEOMETRY. Cílem práce bylo vybudování tříúrovňové hierarchické struktury, kde nejnižší vrstvu tvoří okresy, vyšší vrstvu kraje, skládající se z okresů na nižší úrovni a nejvýše je republika skládající se z krajů. Z dat s nejvyšší úrovní detailu je možné SQL dotazem vygenerovat data na vyšší hierarchické úrovni.

## **Klíčová slova**

Hierarchie vrstev, topologie, topologická primitiva, okřídlená hrana, Oracle Spatial, SDO\_TOPO\_GEOMETRY.

## **Abstract**

This diploma thesis deals with a modelling of hierarchy in topologically structured data. The input data is the data in the SHP format – layers of districts, counties and the Czech Republic. This data is transformed into the object-relational storage SDO\_GEOMETRY using the Bentley PowerMap interface. Then it is loaded into the SDO\_TOPO\_GEOMETRY topology format, using the Oracle Spatial functions. There is also built topology layer hierarchy with three topology layers. The lowest layer consists of districts, the middle layer consists of counties (consisting of districts), and the highest layer is represented by the Czech Republic (consisting of counties). Using the SQL query it is possible generate data on the higher level, keeping the detail of the lowest level.

## **Keywords**

Layer hierarchy, topology, topology primitives, winged edge, Oracle Spatial, SDO\_TOPO\_GEOMETRY.

# Obsah

<b>Seznam obrázků .....</b>	<b>7</b>
<b>Seznam použitých zkratk .....</b>	<b>10</b>
<b>1 Úvod.....</b>	<b>11</b>
<b>2 Topologicky strukturovaná data.....</b>	<b>12</b>
2.1 ULOŽENÍ TOPOLOGICKY STRUKTUROVANÝCH DAT V ORACLE SPATIAL.....	12
2.1.1 Datový typ <i>SDO_TOPO_GEOMETRY</i> .....	15
2.1.2 <i>Node Table</i> (tabulka uzlů).....	16
2.1.3 <i>Edge Table</i> (tabulka hran).....	17
2.1.4 <i>Face Table</i> (tabulka stěn) .....	18
2.1.5 <i>Relationship Information Table</i> .....	18
2.2 HIERARCHIE VRSTEV GEOMETRICKÝCH POPISŮ TOPOLOGICKÝCH PRVKŮ.....	21
2.3 HIERARCHIE VRSTEV V TOPOLOGII <i>H_TOPO</i> .....	27
<b>3 Zpracování vstupních dat .....</b>	<b>28</b>
3.1 ZALOŽENÍ NOVÉHO PROJEKTU V BENTLEY GEOSPATIAL ADMINISTRATOR .....	28
3.2 NAČTENÍ *.SHP VÝKRESŮ DO BGA .....	30
3.3 VYGENEROVÁNÍ SKRIPTŮ PRO VYTVOŘENÍ TABULEK .....	31
3.4 VYTVOŘENÍ TABULEK V SQL DEVELOPERU .....	34
3.5 VYTVOŘENÍ DATABÁZOVÉHO PŘIPOJENÍ V BGA .....	34
3.6 IMPORT VÝKRESŮ DO POWERMAPU .....	37
3.7 ODESLÁNÍ DAT DO DATABÁZE ORACLE SPATIAL .....	40
<b>4 Topologie vytvořená z prostorových geometrií.....</b>	<b>42</b>
4.1 PŘÍPRAVA DAT .....	42
4.1.1 Úprava prostorových tabulek.....	42
4.1.2 Aktualizace prostorových metadat.....	44
4.1.3 Validace prostorových dat.....	45
4.1.4 Vytvoření prostorových indexů na tabulkách <i>CR, KRAJE, OKRESY</i> .....	46
4.2 VYTVOŘENÍ VLASTNÍ TOPOLOGIE.....	47
4.2.1 Vytvoření topologie s použitím procedury <i>SDO_TOPO.CREATE_TOPOLOGY</i> .....	47
4.2.2 Vytvoření „universe face“ <i>F0</i> .....	48
4.2.3 Vytvoření prvkových tabulek.....	49

4.2.4	<i>Asociace prvkových tabulek s topologií</i> .....	50
4.2.5	<i>Vytvoření TopoMap objektu a načtení celé topologie do mezipaměti</i> .....	51
4.2.6	<i>Naplnění prvkových tabulek</i> .....	52
4.2.7	<i>Inicializace topologických metadat</i> .....	58
<b>5</b>	<b>Nesrovnalosti ve vytvořené topologii H_TOPO</b> .....	<b>59</b>
5.1	VYHLEDÁNÍ CHYB A JEJICH PŮVOD .....	59
5.2	NÁVRH ŘEŠENÍ .....	62
<b>6</b>	<b>Ověření funkčnosti vytvořené hierarchie</b> .....	<b>72</b>
6.1	VYTVOŘENÍ SROVNÁVACÍ TOPOLOGIE A_TOPO.....	72
6.2	POROVNÁNÍ VÝSLEDKŮ NA TOPOLOGIÍCH A_TOPO A F_TOPO.....	81
<b>7</b>	<b>Závěr</b> .....	<b>100</b>
	<b>Seznam použité literatury</b> .....	<b>102</b>
	<b>Struktura příloženého CD</b> .....	<b>104</b>

## Seznam obrázků

OBR. 2.1 PROSTOROVÉ OBJEKTY RŮZNÝCH TYPŮ JSOU ULOŽENY V JEDNÉ NEBO VÍCE TABULKÁCH JAKO SDO_GEOMETRY OBJEKTY [8].....	12
OBR. 2.2 POSUNUTÍ HRANICE [8].....	13
OBR. 2.3 TOPOLOGICKÝ DATOVÝ MODEL. KAŽDÝ TOPOLOGICKÝ PRVEK JE REPREZENTOVÁN POMOCÍ SDO_TOPO_GEOMETRY OBJEKTU A TOPOLOGICKÝCH PRIMITIV: UZLŮ (N1, N2, N3, N4), HRAN (E1, E2, E3, E4) A STĚN (F1, F2)[8] .....	13
OBR. 2.4 KONCEPT OKŘÍDLENÉ HRANY VYUŽÍVANÝ ORACLE SPATIAL [6] .....	14
OBR. 2.5 PROPOJENÍ MEZI PRVKOVOU TABULKOU A TABULKAMI S TOPOLOGICKÝMI PRIMITIVY[5] .....	15
OBR. 2.6 PROPOJENÍ PRVKOVÉ TABULKY A TABULKY <NÁZEV_TOPOLOGIE> RELATIONS\$.....	16
OBR. 2.7 PROPOJENÍ TABULKY TOPOLOGICKÝCH VZTAHŮ S TABULKAMI TOPOLOGICKÝCH PRIMITIV .....	20
OBR. 2.8 ATRIBUT TOPO_TYPE VYSTIHUJE CHILD_LAYER_ID, ČILI TG_ID POTOMKA.....	21
OBR. 2.9 HIERARCHIE VRSTEV ZJEDNODUŠENÉHO ADMINISTRATIVNÍHO ČLENĚNÍ ČR[4] .....	22
OBR. 2.10 TOPOLOGY LAYER HIERARCHY MODEL [3].....	23
OBR. 2.11 PŘEHLED ATRIBUTŮ V HIERARCHII .....	25
OBR. 2.12 HIERARCHIE VRSTEV V TOPOLOGII H_TOPO .....	27
OBR. 3.1 ZALOŽENÍ NOVÉHO PROJEKTU .....	29
OBR. 3.2 VYTVOŘENÍ NOVÉHO PRACOVNÍHO PROSTŘEDÍ (WORKSPACE) .....	29
OBR. 3.3 NOVÉ ZALOŽENÝ PROJEKT V BGA A V NĚM VYTVOŘENÉ PRACOVNÍ PROSTŘEDÍ (WORKSPACE) ...	29
OBR. 3.4 IMPORT SHP VÝKRESŮ DO BGA PRSOTŘEDNICTVÍM TXML.EXE .....	30
OBR. 3.5 VYTVOŘENÉ MAPOVÉ OBJEKTY V PROSTŘEDÍ BGA, VČETNĚ JEJICH ATRIBUTŮ (PROPERTIES) ....	31
OBR. 3.6 EDITACE SOUBORU FROM_XFM_TO_ORACLE_SPATIAL_TABLES.CMD .....	32
OBR. 3.7 SPUŠTĚNÍ SKRIPTU FROM_XFM_TO_ORACLE_SPATIAL_TABLES.CMD.....	33
OBR. 3.8 VYGENEROVANÉ SKRIPTY PRO VYTVOŘENÍ TABULEK A PROSTOROVÝCH INDEXŮ V ADRESÁŘI XFM2ORACLE .....	33
OBR. 3.9 SPUŠTĚNÍ MASTERLOAD.BAT .....	34
OBR. 3.10 VYGENEROVANÉ TABULKYCR, KRAJE, OKRESY .....	34
OBR. 3.11 VYTVOŘENÍ DATABÁZOVÉHO PŘIPOJENÍ V BGA .....	35
OBR. 3.12 REGISTRACE MAPOVÝCH OBJEKTŮ .....	35
OBR. 3.13 VLOŽENÍ METADAT UMÍSTĚNÍ KOLEKCÍ POLYGONU.....	36
OBR. 3.14 PŘIDÁNÍ SEZNAMU DO SPRÁVCE PŘÍKAZŮ.....	36
OBR. 3.15 PŘIPOJENÍ POWERMAPU K ORACLE SPATIAL .....	37
OBR. 3.16 ZOBRAZENÍ ZAREGISTROVANÝCH MAPOVÝCH OBJEKTŮ.....	38
OBR. 3.17 PŘIDRUŽENÍ ATRIBUTŮ K ZAREGISTROVANÝM FEATURES .....	38
OBR. 3.18 IMPORT DAT DO VÝKRESU .....	39
OBR. 3.19 VRSTVY NAČTENÉ DO VÝKRESU .....	39
OBR. 3.20 ANALÝZA PRVKŮ NAČTENÝCH DO POWERMAPU.....	40
OBR. 3.21 ODESLÁNÍ DAT Z POWERMAPU DO ORACLE SPATIAL.....	40



OBR. 3.22 NAPLNĚNÁ TABULKA KRAJE V SQL DEVELOPERU .....	41
OBR. 4.1 PRIMÁRNÍ KLÍČ NA XFM_ID VYTVOŘENÝ PŘI PŘEVODU DO SDO_GEOMETRY .....	43
OBR. 4.2 ZRUŠENÍ PRIMÁRNÍHO KLÍČE NA TABULCE CR .....	43
OBR. 4.3 ZMĚNĚNÝ PRIMÁRNÍ KLÍČ A ZRUŠENÝ SLOUPEC XFM_ID.....	43
OBR. 4.4 AKTUALIZOVANÁ METADATA O PROSTOROVÝCH DATECH V POHLEDU USER_SDO_GEOM_METADATA .....	45
OBR. 4.5 VYTVOŘENÝ PROSTOROVÝ INDEX NA TABULCE OKRESY.....	46
OBR. 4.6 NOVĚ VYTVOŘENÁ TOPOLOGIE H_TOPO V POHLEDU USER_SDO_TOPO_INFO.....	47
OBR. 4.7 NOVĚ VYTVOŘENÉ TABULKY H_TOPO_EDGES\$, H_TOPO_FACES\$, H_TOPO_NODES\$, H_TOPO_HISTORY\$ A INDEXY H_TOPO_PKEIDS\$, H_TOPOHIS_PK.....	48
OBR. 4.8 VLOŽENÁ „FACE ZERO“ F0 V TABULCE H_TOPO_FACES\$ .....	48
OBR. 4.9 VYTVOŘENÉ PRVKOVÉ TABULKY H_CR_TOPO, H_KRAJE_TOPO A H_OKRESY_TOPO.....	49
OBR. 4.10 PRINCIP REALATIONSHIP INFORMATION TABLE(TABULKY TOPOLOGICKÝCH VZTAHŮ) - PROPOJENÍ MEZI PRVKOVOU TABULKOU A TABULKAMI TOPOLOGICKÝCH PRIMITIV .....	51
OBR. 4.11 PŘEHLED ASOCIOVANÝCH TABULEK V POHLEDU USER_SDO_TOPO_METADATA .....	51
OBR. 4.12 NAČTENÁ TOPOLOGIE H_TOPO DO MEZIPAMĚTI TOPOMAP .....	52
OBR. 4.13 NAPLNĚNÁ PRVKOVÁ TABULKA H_OKRESY_TOPO (VÝŘEZ) .....	53
OBR. 4.14 NAPLNĚNÁ PRVKOVÁ TABULKA H_KRAJE_TOPO .....	57
OBR. 4.15 NAPLNĚNÁ PRVKOVÁ TABULKA H_CR_TOPO .....	57
OBR. 4.16 NOVĚ VYTVOŘENÉ INDEXY NA TABULKÁCH H_TOPO_EDGES\$, H_TOPO_FACES\$ A H_TOPO_NODES\$.....	58
OBR. 5.1 OKRES FRÝDEK-MÍSTEK (TG_ID = 7) JE TVOŘEN 19TI STĚNAMI (VÝŘEZ) .....	59
OBR. 5.2 JEDNA STĚNA (JEDNO TOPO_ID) JE SPOLEČNÁ PRO VÍCE OKRESŮ (JEDNO TG_ID) (VÝŘEZ) .....	60
OBR. 5.3 VIZUALIZACE ČR - TOPOLOGIE H_TOPO.....	61
OBR. 5.4 MEZERA MEZI OKRESY .....	61
OBR. 5.5 MEZERA MEZI OKRESY V PŮVODNÍCH DATECH, ZOBRAZENÁ V ARCMAPU .....	62
OBR. 5.6 FACE_ID NEOBSAZENÝCH STĚN.....	67
OBR. 5.7 EDGE_ID OHRANIČUJÍCÍCH HRAN .....	67
OBR. 5.8 VÝČET STĚN, KTERÉ SOUSEDÍ S HRANAMI, JEJICHŽ EDGE_ID = 23, 279, 266 .....	68
OBR. 5.9 TG_ID OKRESŮ, KTERÁ ODPOVÍDAJÍ SOUSEDÍCÍM STĚNÁM .....	68
OBR. 5.10 OKRESY, KTERÉ SOUSEDÍ S HLEDANÝMI OKRESY .....	69
OBR. 6.1 TOPOLOGIE A_TOPO .....	73
OBR. 6.2 ULOŽENÍ ČTVEREČKŮ DVĚMA ROHY, KTERÉ NENÍ PODPOROVÁNO TOPOLOGIÍ .....	75
OBR. 6.3 ULOŽENÍ ČTVEREČKŮ ČTYŘMI ROHY .....	75
OBR. 6.4 PŘEHLED ASOCIOVANÝCH TABULEK V POHLEDU USER_SDO_TOPO_METADATA .....	77
OBR. 6.5 NAPLNĚNÁ TABULKA A_A_TOPO.....	78
OBR. 6.6 NAPLNĚNÁ TABULKA AB_A_TOPO .....	79
OBR. 6.7 NAPLNĚNÁ TABULKA ABCD_A_TOPO .....	79
OBR. 6.8 VYTVOŘENÁ TOPOLOGIE A_TOPO S TOPOLOGICKÝMI PRIMITIVY .....	79
OBR. 6.9 PŘEHLED PROPOJENÍ PRVKOVÝCH TABULEK S TABULKAMI TOPOLOGICKÝCH PRIMITIV.....	80

OBR. 6.10 ČTVEREČKY TVOŘÍCÍ OBDÉLNÍK AB.....	82
OBR. 6.11 VÝČET FACE_ID STĚN TVOŘÍCÍCH OBDÉLNÍK AB.....	82
OBR. 6.12 ČTVEREČKY A, B TVOŘÍCÍ OBDÉLNÍK AB .....	83
OBR. 6.13 VÝČET OKRESŮ JIHOČESKÉHO KRAJE .....	84
OBR. 6.14 STĚNY TVOŘÍCÍ JIHOČESKÝ KRAJ.....	84
OBR. 6.15 SEDM OKRESŮ JIHOČESKÉHO KRAJE .....	85
OBR. 6.16 VÝČET HRAN ČTVEREČKŮ, KTERÉ TVOŘÍ OBDÉLNÍK AB .....	86
OBR. 6.17 HRANY, KTERÉ TVOŘÍ HRANICE ČTVEREČKŮ, KTERÉ TVOŘÍ OBDÉLNÍK AB (ČERVENĚ).....	86
OBR. 6.18 VÝŘEZ VÝČTU HRAN, KTERÉ TVOŘÍ HRANICE OKRESŮ JIHOČESKÉHO KRAJE .....	87
OBR. 6.19 HRANY, KTERÉ TVOŘÍ HRANICE OKRESŮ JIHOČESKÉHO KRAJE (ČERVENĚ).....	87
OBR. 6.20 VÝČET HRAN, KTERÉ TVOŘÍ OBDÉLNÍK AB .....	89
OBR. 6.21 HRANY, KTERÉ TVOŘÍ HRANICE OBDÉLNÍKU AB (ZELENĚ).....	89
OBR. 6.22 VÝŘEZ VÝČTU HRAN, KTERÉ TVOŘÍ HRANICI JIHOČESKÉHO KRAJE.....	90
OBR. 6.23 HRANY, KTERÉ TVOŘÍ HRANICI JIHOČESKÉHO KRAJE (ZELENĚ).....	90
OBR. 6.24 VÝČET OBDÉLNÍKŮ TVOŘÍCÍ ČTVEREC ABCD.....	91
OBR. 6.25 VÝČET STĚN TVOŘÍCÍ ČTVEREC ABCD .....	91
OBR. 6.26 OBDÉLNÍKY TVOŘÍCÍ ČTVEREC ABCD .....	91
OBR. 6.27 VÝČET KRAJŮ TVOŘÍCÍ ČR .....	92
OBR. 6.28 KRAJE ČR.....	92
OBR. 6.29 VÝČET STĚN TVOŘÍCÍ 14 KRAJŮ ČR.....	93
OBR. 6.30 VÝČET ČTVEREČKŮ TVOŘÍCÍCH ČTVEREC ABCD.....	93
OBR. 6.31 ČTVEREC ABCD TVOŘENÝ ČTYŘMI ČTVEREČKY A, B, C, D .....	94
OBR. 6.32 VÝŘEZ VÝČTU OKRESŮ ČR.....	94
OBR. 6.33 77 OKRESŮ TVOŘÍCÍ ČR.....	94
OBR. 6.34 VÝČET HRAN TVOŘÍCÍ HRANICE ČTVEREČKŮ A, B, C, D.....	95
OBR. 6.35 HRANY TVOŘÍCÍ HRANICE ČTVEREČKŮ A, B, C, D.....	96
OBR. 6.36 VÝŘEZ VÝČTU HRAN TVOŘÍCÍCH HRANICE OKRESŮ ČR.....	96
OBR. 6.37 HRANY TVOŘÍCÍ HRANICE OKRESŮ ČR.....	97
OBR. 6.38 VÝČET HRAN TVOŘÍCÍ HRANICE ČTVERCE ABCD .....	98
OBR. 6.39 HRANY TVOŘÍCÍ HRANICI ČTVERCE ABCD (FIALOVĚ) .....	98
OBR. 6.40 VÝŘEZ VÝČTU HRAN TVOŘÍCÍCH HRANICI ČR .....	99
OBR. 6.41 HRANY TVOŘÍCÍ HRANICI ČR (FIALOVĚ) .....	99

## Seznam použitých zkratk

BGA	Bentley Geospatial Administrator
XML	eXtensible Markup Language
XFM	XML Feature Modeling
GIS	Geografický informační systém
SRID	Spatial Reference System Identifier
ČR	Česká Republika

# 1 Úvod

Oracle Spatial je integrovaná množina funkcí a procedur, které umožňují uložení, přístup a analýzu prostorových dat rychlým a efektivním způsobem v databázi Oracle. K uložení prostorových dat používá jednak objektově-relační datový model, jednak topologický datový model. Oba dva modely mají svá pozitiva i negativa.

Objektově-relační datový model využívá objektového datového typu SDO\_GEOMETRY. V jedné tabulce jsou tak pro prvek popisná data uložena společně s prostorovými. Tento model podporuje mnoho geometrických typů, např. oblouky, kružnice, liniové řetězce či polygony. Další výhodou je snadné použití při vytváření a údržbě prostorových indexů a dotazů [16]. Na druhou stranu tento model způsobuje redundantnost, protože identická část geometrického popisu sousedících prostorových objektů je ukládána dvakrát.

Zmiňovaný nedostatek řeší druhý model, který používá Oracle Spatial pro uložení prostorových dat. Jedná se o topologický koncept okřídlené hrany (*winged edge*) s využitím datového typu SDO\_TOPO\_GEOMETRY. Tento model využívá k uložení dat tři topologická primitiva, jimiž jsou uzly, hrany a stěny [9]. Kromě zamezení duplicity dat má tento model i další výhody. Velice dobře zajišťuje konzistenci dat, neboť změny na topologickém primitivu se implicitně promítají do všech prvků, které toto primitivum sdílí. Zabraňuje se tak možným chybám, které mohou být způsobené násobnými změnami ve vrstvách prvků. Další výhodou je jednoduché dotazování z hlediska topologických vztahů; tyto vztahy zůstávají zachovány i pokud dojde ke změnám v souřadnicích, např. k otočení nebo ke změně měřítka [8].

Topologický datový model v Oracle Spatial podporuje hierarchické modelování. Umožňuje tak vytvářet několik hierarchicky závislých vrstev, kdy jedna vrstva se skládá z prvků ve vrstvě o jednu úroveň níž. Tato vrstva na nižší úrovni může nadále sestávat z prvků ve vrstvě, která je v rámci hierarchie ještě o další stupeň níže [4]. A právě tímto modelováním hierarchie topologicky strukturovaných dat se zabývá tato práce.

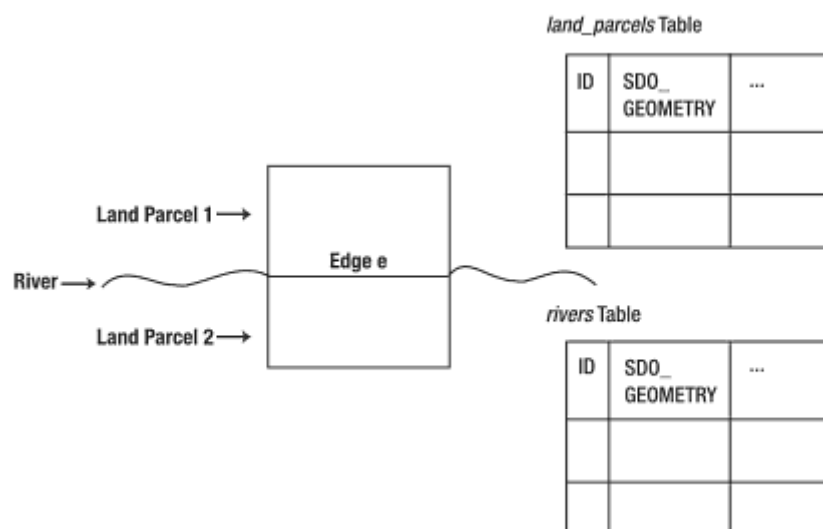
Budovaná hierarchie je tříúrovňová. Na nejvyšší úrovni hierarchie (úroveň 2) je Česká Republika. Ta se skládá ze 14 krajů, které jsou o jednu úroveň níže (úroveň 1). Na nejnižší, nulté úrovni, je 77 okresů tvořící kraje. Podstatou hierarchické struktury je to, že z topologických primitiv (tj. uzlů, hran a stěn), ze kterých se skládá vrstva okresů na nejnižší úrovni, lze poskládat hranice krajů či republiky na vyšších úrovních. Tyto nově „poskládané“ hranice budou mít úroveň detailu dle vrstvy na nejnižší úrovni.

## 2 Topologicky strukturovaná data

Topologie je odvětví matematiky věnující se prostorovým objektům. Topologické vztahy zahrnují vztahy typu např. *uvnitř, obsahuje, pokrývá, je pokryt, dotýká se a překrytí s průsečíkem hranic*. Topologické vztahy zůstávají nezměněny při otočení nebo při změně měřítka.

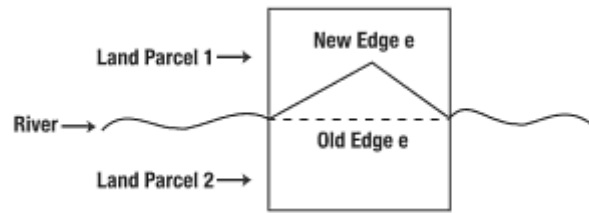
### 2.1 Uložení topologicky strukturovaných dat v Oracle Spatial

Oracle Spatial běžně ukládá geometrii objektů pomocí objektově relačního modelu SDO\_GEOMETRY. Data jsou tak ukládána do mnoha tabulek, např. tabulka řek, tabulka budov, tabulka parcel atd... Znamená to, že pokud prvky sdílí stejnou hranici (např. řeka a hranice parcely), data jsou v databázi uložena dvakrát (např. v tabulce řek a v tabulce parcel, viz obr. 2.1).



**obr. 2.1 Prostorové objekty různých typů jsou uloženy v jedné nebo více tabulkách jako SDO\_GEOMETRY objekty [8]**

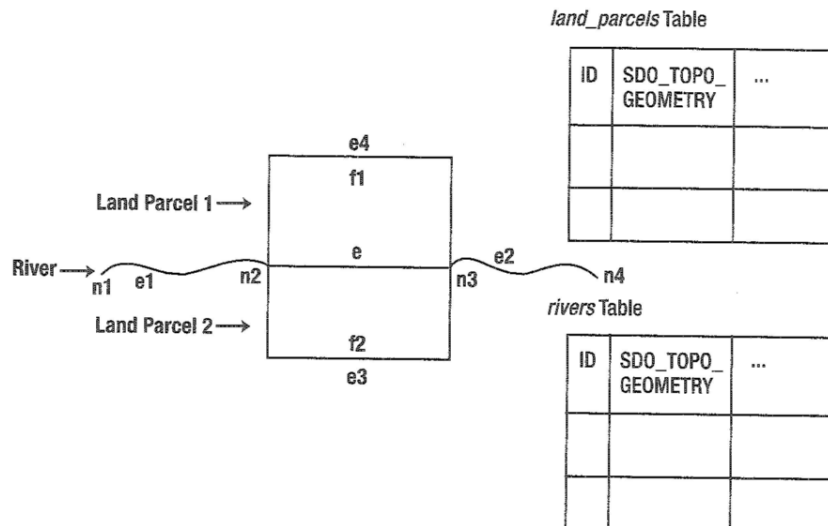
Při změně prostorového objektu, např. posunutí hranice (viz obr. 2.2), je nutné provést změny hned v několika tabulkách. Ve výše uvedeném příkladu by se jednalo o změny u parcely č.1 a č.2 (*Land Parcel 1* a *Land Parcel 2*) v tabulce *land\_parcel* a změnu u řeky v tabulce *Rivers* [8].



obr. 2.2 Posunutí hranice [8]

Pro takovéto případy, kdy je jeden prvek (např. hranice) sdílen více objekty, je vhodnější použít topologický datový model (*Oracle's Topology Data Model*) [8].

Oracle Spatial používá topologického konceptu tzv. „okřídlené hrany“ („*Winged Edge*“). Tento model ukládá jednotlivé objekty pomocí třech topologických primitiv, jimiž jsou *uzly*, *hrany* a *stěny* (viz obr. 2.3).

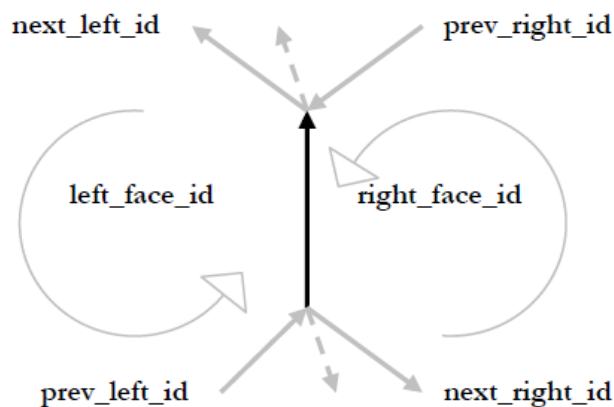


obr. 2.3 Topologický datový model. Každý topologický prvek je reprezentován pomocí SDO\_TOPO\_GEOMETRY objektu a topologických primitiv: uzlů (n1, n2, n3, n4), hran (e1, e2, e3, e4) a stěn (f1, f2)[8]

- **Uzel** (*node*) je reprezentován bodem, může být izolován, nebo může tvořit hraniční body hrany. Prostorová lokace uzlu je ve 2D dána dvojicí souřadnic.
- **Hrana** (*edge*) je ohraničena dvěma uzly, počátečním (*start\_node*) a koncovým (*end\_node*). Počáteční a koncový bod určují orientaci hrany, která je důležitá pro určování topologických vztahů. Hrana může obsahovat několik vrcholů vytvářejících lomenou čáru. Kruhové oblouky nejsou v topologickém konceptu povoleny. Hrana

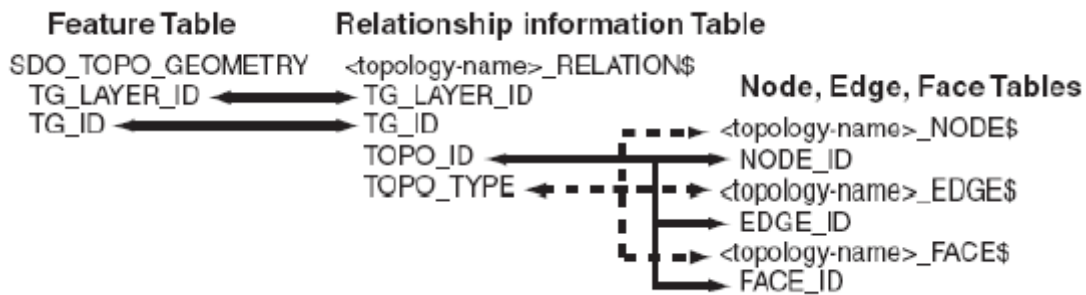
leží mezi dvěma stěnami, na které odkazuje. Každá orientovaná hrana obsahuje také odkazy na předcházející a následující levé a pravé hrany (viz obr. 2.4).

- **Stěna** (*face*) je reprezentována polygonem. Je ohraničená jedním vnějším hranovým řetězcem, může obsahovat libovolný počet „děr“, tj. vnitřních hranic. Obsahuje odkaz na jednu orientovanou hranu, která je součástí její vnější hranice. V případě izolovaných uzlů či hran, které jsou uvnitř této stěny, obsahuje také odkaz na tyto prvky. Každá topologie obsahuje jednu základní stěnu, tzv. „*Universal Face*“, která obsahuje veškeré uzly, hrany a stěny v topologii.



obr. 2.4 Koncept okřídlené hrany využívaný Oracle Spatial [6]

Informace o topologických primitivech jsou uloženy ve speciálních tabulkách uzlů ( $\langle \text{název\_topologie} \rangle\_NODE\$$ ), hran ( $\langle \text{název\_topologie} \rangle\_EDGE\$$ ) a stěn ( $\langle \text{název\_topologie} \rangle\_FACE\$$ ). Tyto tabulky jsou vytvořeny v okamžiku vytvoření topologie. Jednotlivé topologické prvky (*features*) jsou uloženy v prvkové tabulce (*feature table*). Dále je nutná tabulka *Relationship Information Table* ( $\langle \text{název\_topologie} \rangle\_RELATION\$$ ), která umožní propojení prvkové tabulky s tabulkami topologických primitiv, ze kterých sestává geometrický popis prvku. Schéma propojení těchto tabulek je zobrazeno na obr. 2.5.



obr. 2.5 Propojení mezi prvkovou tabulkou a tabulkami s topologickými primitivami[5]

### 2.1.1 Datový typ SDO\_TOPO\_GEOMETRY

Prvkové tabulky ukládají geometrii prvku pomocí datového typu SDO\_TOPO\_GEOMETRY. Obdobně jako datový typ SDO\_GEOMETRY, zachycuje datový typ SDO\_TOPO\_GEOMETRY tvar a umístění prvku. Na rozdíl od SDO\_GEOMETRY, SDO\_TOPO\_GEOMETRY neukládá explicitně vyjádřené souřadnice. Namísto toho ukládá odkazy na topologická primitiva, ze kterých může být tvar odvozen [8]. Tento datový typ má následující strukturu:

- TG\_TYPE                      Typ geometrie topologického prvku:
  - 1 = bod, multibod
  - 2 = liniový (multiliniový) řetězec
  - 3 = polygon, multipolygon
  - 4 = heterogenní kolekce
- TG\_ID                        Unikátní identifikátor topologicky strukturovaného prvku
- TG\_LAYER\_ID                Unikátní identifikátor topologické vrstvy, do které topologicky strukturovaný prvek náleží
- TOPOLOGY\_ID               Unikátní identifikátor topologie

Každá prvková tabulka obsahuje sloupec typu SDO\_TOPO\_GEOMETRY. Tento datový typ obsahuje atribut TG\_LAYER\_ID (unikátní identifikátor topologické vrstvy, který přiřazuje Spatial při jejím vytvoření). Dále obsahuje atribut TG\_ID (unikátní identifikátor prvku ve vrstvě). Hodnoty těchto dvou atributů (v prvkové tabulce) mají odpovídající hodnoty ve sloupcích TG\_LAYER\_ID a TG\_ID v tabulce <název\_topologie> RELATION\$, jak je



zobrazeno na obr. 2.6. Struktura tabulky *<název\_topologie> RELATION\$* bude vysvětlena v kapitole 2.1.5 [4].

The image shows two Oracle SQL Developer windows. The top window displays the 'A\_A\_TOPO' table with the following data:

	NAZEV	AB	FEATURE
1	A	AB	MDSYS.SDO_TOPO_GEOMETRY (3,1,1,22)
2	B	AB	MDSYS.SDO_TOPO_GEOMETRY (3,2,1,22)
3	C	CD	MDSYS.SDO_TOPO_GEOMETRY (3,3,1,22)
4	D	CD	MDSYS.SDO_TOPO_GEOMETRY (3,4,1,22)

The bottom window displays the 'A\_TOPO\_RELATIONS\$' table with the following data:

	TG_LAYER_ID	TG_ID	TOPO_ID	TOPO_TYPE	TOPC
1	1	1	1	3 (null)	
2	1	2	2	3 (null)	
3	1	3	3	3 (null)	
4	1	4	4	3 (null)	

Orange circles highlight the 'TG\_LAYER\_ID' column in the bottom table, which contains the value '1' for all rows. Green circles highlight the 'TG\_ID' column in the bottom table, which contains values 1, 2, 3, and 4 corresponding to the rows in the top table. Orange arrows point from the 'TG\_ID' values to the 'FEATURE' column in the top table, showing the mapping between the two tables.

obr. 2.6 Propojení prvkové tabulky a tabulky *<název\_topologie> RELATIONS*

## 2.1.2 Node Table (tabulka uzlů)

Tabulka uzlů (*<název\_topologie>\_NODE\$*) poskytuje informace o všech uzlech v topologii. Tato tabulka obsahuje následující atributy:

- **NODE\_ID (NUMBER)** Unikátní identifikátor uzlu
- **EDGE\_ID (NUMBER)** Unikátní identifikátor hrany (se znaménkem), pokud je nějaká s uzlem asociována
- **FACE\_ID (NUMBER)** Unikátní identifikátor stěny, pokud je nějaká s uzlem asociována
- **GEOMETRY (SDO-GEOMETRY)** Geometrický popis bodu, reprezentujícího uzel

### 2.1.3 Edge Table (tabulka hran)

Tabulka hran (<název\_topologie>\_EDGES) poskytuje informace o všech hranách v topologii. Tato tabulka obsahuje následující atributy:

- EDGE\_ID (NUMBER) Unikátní identifikátor hrany (se znaménkem)
- START\_NODE\_ID (NUMBER) Unikátní identifikátor počátečního bodu hrany
- END\_NODE\_ID (NUMBER) Unikátní identifikátor koncového bodu hrany
- NEXT\_LEFT\_EDGE\_ID (NUMBER) Unikátní identifikátor next left hrany (se znaménkem)
- PREV\_LEFT\_EDGE\_ID (NUMBER) Unikátní identifikátor previous left hrany (se znaménkem)
- NEXT\_RIGHT\_EDGE\_ID (NUMBER) Unikátní identifikátor next right hrany (se znaménkem)
- PREV\_RIGHT\_EDGE\_ID (NUMBER) Unikátní identifikátor previous right hrany (se znaménkem)
- LEFT\_FACE\_ID (NUMBER) Unikátní identifikátor stěny vlevo od hrany
- RIGHT\_FACE\_ID (NUMBER) Unikátní identifikátor stěny vpravo od hrany
- GEOMETRY (SDO\_GEOMETRY) Geometrický popis hrany

Hodnoty atributů NEXT\_LEFT\_EDGE\_ID a NEXT\_RIGHT\_EDGE\_ID odkazují na hrany, na které narazíme při pohybu v protisměru chodu hodinových ručiček podél obvodu stěny vlevo, respektive vpravo. Hodnoty atributů PREV\_LEFT\_EDGE\_ID a PREV\_RIGHT\_EDGE\_ID narazíme při pohybu po směru chodu hodinových ručiček podél obvodu stěny vlevo, respektive vpravo. Hodnota atributu LEFT\_FACE\_ID odkazuje na stěnu vlevo od orientované hrany, hodnota atributu RIGHT\_FACE\_ID odkazuje na stěnu vpravo od orientované hrany (viz obr. 3.4). Znaménko před hodnotou libovolného z výše uvedených identifikátorů určuje vzájemnou orientaci odkazovaných prvků [4].

### 2.1.4 Face Table (tabulka stěn)

Tabulka stěn (<název\_topologie>\_FACE\$) poskytuje informace o všech stěnách v topologii. Stěny mohou obsahovat izolované uzly a izolované hrany. Tyto izolované uzly a hrany nejsou na hranici stěny, ale uvnitř stěny.

Tato tabulka obsahuje následující atributy:

- FACE\_ID (NUMBER) Unikátní identifikátor stěny
- BOUNDARY\_EDGE\_ID (NUMBER) Unikátní identifikátor hrany (se znaménkem), ke které stěna náleží
- ISLAND\_EDGE\_ID\_LIST (SDO\_LIST\_TYPE) Unikátní identifikátory izolovaných hran, pokud jsou nějaké v dané stěně
- ISLAND\_NODE\_ID\_LIST (SDO\_LIST\_TYPE) Unikátní identifikátory izolovaných bodů, pokud jsou nějaké v dané stěně
- MBR\_GEOMETRY (SDO\_GEOMETRY) Minimální ohraničující obdélník

U atributu BOUNDARY\_EDGE\_ID je zobrazena pouze jedna hrana; všechny ostatní hrany mohou být od této hrany odvozeny (pomocí next a previous ukazatelů pro tuto hrana).

Atribut MBR\_GEOMETRY je povinný (kromě univerzální stěny). Nad tímto sloupcem je vytvářen prostorový index.

### 2.1.5 Relationship Information Table

Tabulka topologických vztahů (<název\_topologie>\_RELATION\$) umožňuje propojení prvkových tabulek s tabulkami topologických primitiv, ze kterých sestává geometrický popis prvku. Tuto tabulku automaticky udržuje Spatial. Každý řádek v tabulce jednoznačně určuje topologicky strukturovaný prvek (nazývaný též *topology geometry*) z prvkové tabulky, která je asociovaná s topologií. Spatial automaticky generuje ID pro každý takový prvek (TG\_ID).

Tato tabulka obsahuje následující atributy:

- **TG\_LAYER\_ID (NUMBER)** Unikátní identifikátor topologické vrstvy, do které topologicky strukturovaný prvek náleží
- **TG\_ID (NUMBER)** Unikátní identifikátor topologicky strukturovaného prvku
- **TOPO\_ID (NUMBER)** Pokud v topologii není hierarchie vrstev, vyjadřuje unikátní identifikátor topologického primitiva asociovaného s topologickým prvkem. Pokud je v topologii hierarchie vrstev, vyjadřuje úroveň v hierarchii. Hodnotu vloží Oracle Spatial.
- **TOPO\_TYPE (NUMBER)** Pokud v topologii není hierarchie vrstev:
  - 1 = uzel
  - 2 = hrana
  - 3 = stěna Pokud je v topologii hierarchie vrstev, vyjadřuje TG\_ID potomka (CHILD\_LAYER\_ID; bude vysvětleno později). Hodnotu vloží Oracle Spatial
- **TOPO\_ATTRIBUTE (VARCHAR2)** Hodnotu vloží Oracle Spatial

První dva atributy, TG\_LAYER\_ID a TG\_ID, byly vysvětleny v kapitole 2.1.1. Třetím atributem je TOPO\_ID. Tento atribut má dva významy, v závislosti na tom, zda je v topologii hierarchie vrstev. Pokud v topologii není hierarchie vrstev, jedná se o unikátní identifikátor topologického primitiva asociovaného s topologickým prvkem. Zde záleží, jaká je hodnota atributu TOPO\_TYPE. Význam i tohoto atributu závisí na tom, zda je v topologii hierarchie vrstev. Uvažujme i zde variantu bez hierarchie. V případě, kdy je TOPO\_TYPE = 1, se jedná o uzel, pokud je TOPO\_TYPE = 2, jedná se o hranu a v případě TOPO\_TYPE = 3 jde o stěnu. Pokud je tedy TOPO\_TYPE = 1, TOPO\_ID je unikátní identifikátor stěny, která tvoří topologický prvek. Obdobně u TOPO\_TYPE = 2 je TOPO\_ID identifikátorem hrany, která tvoří prvek a u TOPO\_TYPE = 3 je TOPO\_ID identifikátorem stěny.

Na obr. 2.7 je znázorněno propojení mezi tabulkou A\_TOPO\_RELATION\$ a tabulkami topologických primitiv (s tabulkou stěn a tabulkou hran). V tabulce A\_TOPO\_RELATION\$ je hodnota atributu TOPO\_TYPE = 3, jedná se tedy o stěnu. TOPO\_ID je zde tudíž identifikátorem stěny, jde o hodnotu FACE\_ID. Na obrázku je vidět (modrou barvou) propojení TOPO\_ID s tabulkou stěn A\_TOPO\_FACE\$ a dále také s tabulkou hran A\_TOPO\_EDGE\$. V tabulce hran jsou tyto hodnoty ve sloupcích LEFT\_FACE\_ID a RIGHT\_FACE\_ID podle toho, s jakou stěnou která hrana sousedí.

The image shows four database tables with their data and relationships highlighted by blue circles and arrows:

- A\_TOPO\_EDGES\$** table:
 

EDGE_ID	S...	E...	NE...	P...	NE...	P...	LEFT_FACE...	RIGHT_FACE...	GEC
1	1	1	2	2	5	-5	-4	1	-1 MDE
2	2	2	3	3	1	4	7	2	2 MDE
- A\_A\_TOPO\$** table:
 

NAZEV	AB	FEATURE
1 A	AB	MDSYS.SDO_TOPO_GEOMETRY (3, 1, 1, 22)
2 B	AB	MDSYS.SDO_TOPO_GEOMETRY (3, 2, 1, 22)
3 C	CD	MDSYS.SDO_TOPO_GEOMETRY (3, 3, 1, 22)
4 D	CD	MDSYS.SDO_TOPO_GEOMETRY (3, 4, 1, 22)
- A\_TOPO\_RELATION\$** table:
 

TG_LAYER_ID	TG_ID	TOPO_ID	TOPO_TYPE	TOPO_ATTRIBUTE
1	1	1	3 (null)	
2	1	2		
3	1	3		
4	1	4		
- A\_TOPO\_FACE\$** table:
 

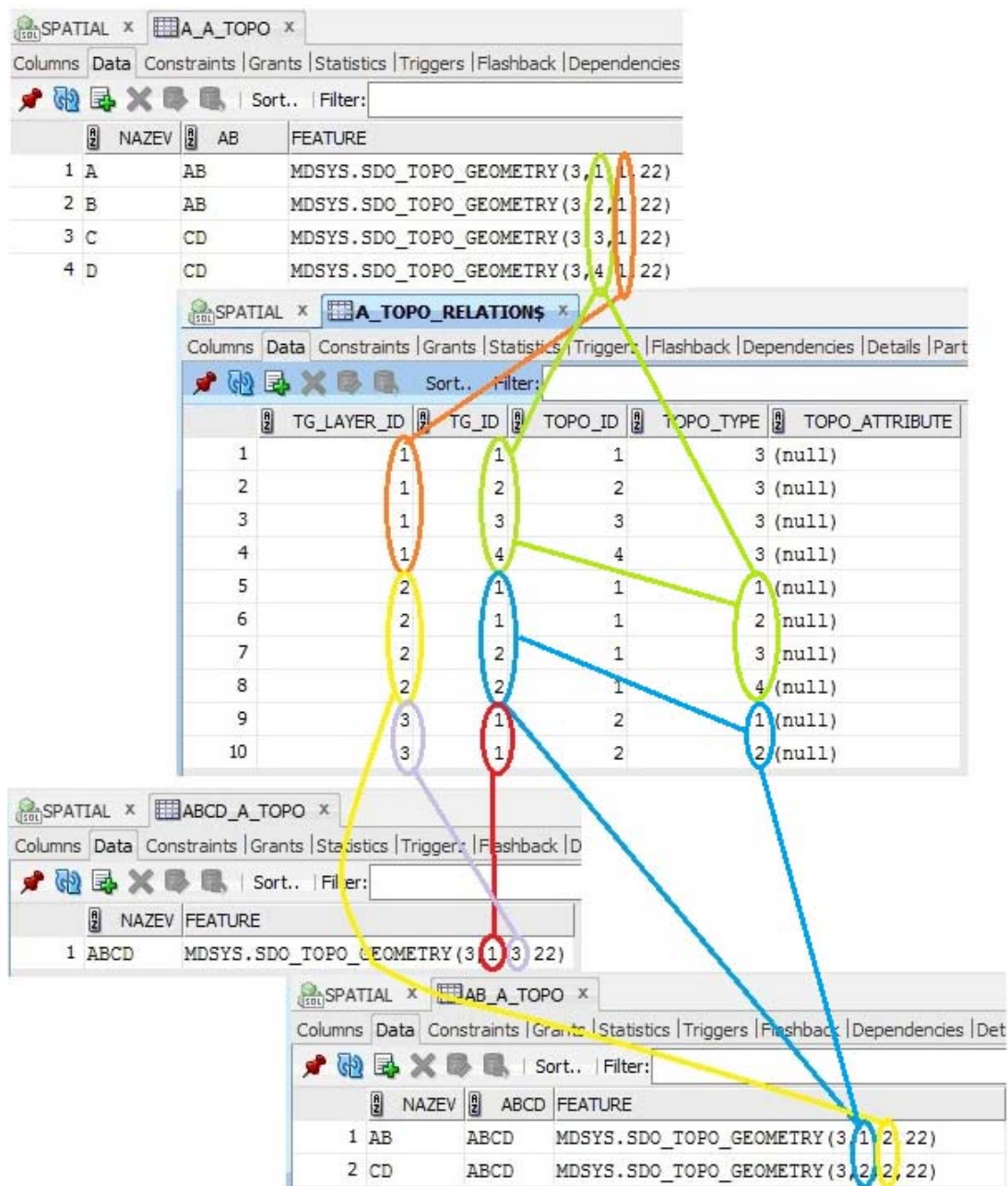
FACE_ID	BOUN...	ISLANI
1	-1	(null) MDSYS
2	1	1 MDSYS
3	2	4 MDSYS
4	3	6 MDSYS
5	4	9 MDSYS

Relationships shown by blue circles and arrows:

- TOPO\_ID = 1 in A\_TOPO\_RELATION\$ points to FACE\_ID = 1 in A\_TOPO\_FACE\$.
- TOPO\_ID = 2 in A\_TOPO\_RELATION\$ points to FACE\_ID = 2 in A\_TOPO\_FACE\$.
- TOPO\_ID = 3 in A\_TOPO\_RELATION\$ points to FACE\_ID = 3 in A\_TOPO\_FACE\$.
- TOPO\_ID = 4 in A\_TOPO\_RELATION\$ points to FACE\_ID = 4 in A\_TOPO\_FACE\$.
- TOPO\_ID = 1 in A\_TOPO\_RELATION\$ points to LEFT\_FACE\_ID = 1 and RIGHT\_FACE\_ID = 2 in A\_TOPO\_EDGES\$.
- TOPO\_ID = 2 in A\_TOPO\_RELATION\$ points to LEFT\_FACE\_ID = 2 and RIGHT\_FACE\_ID = 3 in A\_TOPO\_EDGES\$.
- TOPO\_ID = 3 in A\_TOPO\_RELATION\$ points to LEFT\_FACE\_ID = 3 and RIGHT\_FACE\_ID = 4 in A\_TOPO\_EDGES\$.
- TOPO\_ID = 4 in A\_TOPO\_RELATION\$ points to LEFT\_FACE\_ID = 4 and RIGHT\_FACE\_ID = 5 in A\_TOPO\_EDGES\$.

obr. 2.7 Propojení tabulky topologických vztahů s tabulkami topologických primitiv

Uvažujme nyní variantu, že v topologii máme hierarchii vrstev (hierarchie vrstev bude blíže vysvětlena v kapitole 2.2). Význam atributů TOPO\_ID a TOPO\_TYPE u nejnižší „nulté“ úrovně v hierarchii (kdy TG\_LAYER\_ID = 1) je stejný jako v topologii bez hierarchie. Pokud je hodnota atributu TG\_LAYER > 1, pak hodnota atributu TOPO\_ID vystihuje úroveň v hierarchii. Hodnota atributu TOPO\_TYPE pak vystihuje CHILD\_LAYER\_ID, což je TG\_ID potomka (na obr. 2.8 znázorněno modrou a zelenou barvou).



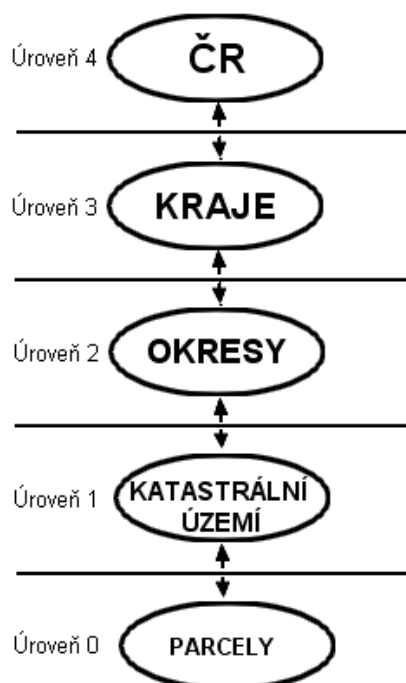
obr. 2.8 Atribut TOPO\_TYPE vystihuje CHILD\_LAYER\_ID, čili TG\_ID potomka

## 2.2 Hierarchie vrstev geometrických popisů topologických prvků

V některých topologiích mají jednotlivé vrstvy mezi sebou hierarchickou závislost. Vrstva na nejvyšší úrovni hierarchie sestává z prvků nacházejících se na nižší úrovni hierarchie. Tato vrstva na nižší úrovni hierarchie může dále sestávat z prvků nacházejících se ve vrstvě, která je v rámci hierarchie ještě o další stupeň níže. Názorně se lze takovou hierarchii vrstev představit na příkladu zjednodušeného administrativního členění státu [4]:

- **Stát** na nejvyšší úrovni hierarchie. Je tvořen kraji.
- **Kraje** na druhé úrovni hierarchie. Jsou tvořeny okresy.
- **Okresy** na další nižší úrovni hierarchie. Skládají se z obcí, případně z katastrálních území.
- **Katastrální území** na předposlední úrovni hierarchie. Jsou tvořeny parcelami.
- **Parcely** na nejnižší (nejpodrobnější) úrovni celé hierarchie.

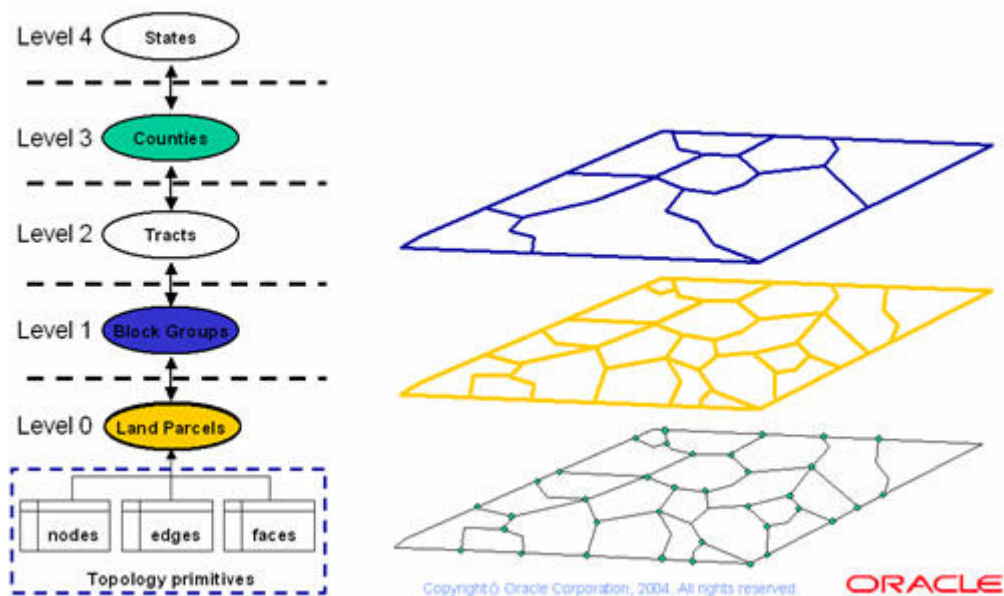
Hierarchie jednotlivých vrstev zjednodušeného administrativního členění ČR je zachycena na obr. 2.9.



obr. 2.9 Hierarchie vrstev zjednodušeného administrativního členění ČR [4]

Pokud mají jednotlivé vrstvy v topologii mezi sebou takovéto hierarchické vazby, je při modelování jednotlivých vrstev efektivnější tuto hierarchickou vazebnost využít, než specifikovat každou vrstvu jednotlivě. Je například efektivnější specifikovat pouze jednotlivé okresy, ze kterých se skládá daný kraj, než určit všechny parcely pro všechna katastrální území ve všech okresech, ze kterých se skládá daný kraj [4].

Nejnižší úroveň v hierarchii vrstev, obsahující největší množství detailu, má označení *Úroveň 0*. Vyšší úrovně v hierarchii jsou pak očíslovány *1, 2, . . .* Vrstvy, které jsou v rámci hierarchie sousední, mají vztah rodič-potomek. Vrstva na vyšší úrovni hierarchie je tzv. rodičovská vrstva pro právě jednu vrstvu na bezprostředně nižší úrovni hierarchie, tzv. vrstva potomka. Vrstva potomka může mít vazbu na více rodičovských vrstev [4].



obr. 2.10 Topology Layer Hierarchy Model [3]

Každá vrstva nad vrstvou na nejnižší úrovni hierarchie může být tvořena jednak prvky nacházející se ve vrstvě na nižší úrovni, nebo také prvky tvořeny topologickými primitivy (stěny, uzly, hrany), případně oběma. Například kraje mohou být tvořeny okresy, nebo mohou být tvořeny stěnami, nebo oběma (např. Jihočeský kraj tvořen okresy, Západočeský kraj tvořen stěnami atd.). Ale jeden konkrétní prvek v rámci vrstvy musí být tvořen buď prvky z další nižší vrstvy, NEBO topologickými primitivy, nikoliv jejich kombinací. Například Jihočeský kraj nemůže být tvořen okresy a stěnami [5].

Topologická hierarchie vrstev je něco podobného jako síťová hierarchie (*network hierarchy*). Jsou zde však dva velké rozdíly, které by neměly být zaměňovány. Například nejnižší úroveň v topologické hierarchii vrstev je *Úroveň 0*, zatímco v síťové hierarchii je nejnižší úrovní *Úroveň 1*. Dále v topologické hierarchii vrstev má každý rodič jednoho potomka a každý potomek může mít mnoho rodičů, zatímco v síťové hierarchii může mít každý rodič mnoho potomků, ale každý potomek pouze jednoho rodiče [5].



K vytváření hierarchické struktury dochází již v průběhu tvorby samotné topologie. V některých krocích jsou odlišnosti od topologií vytvářených bez hierarchické struktury. Prvním bodem, kde dochází k těmto odlišnostem, je *asociace prvkových tabulek s topologií*, kdy se do topologie vkládají geometrické vrstvy topologických prvků. Využívá se k tomu procedury SDO\_TOPO.ADD\_TOPO\_GEOMETRY\_LAYER.

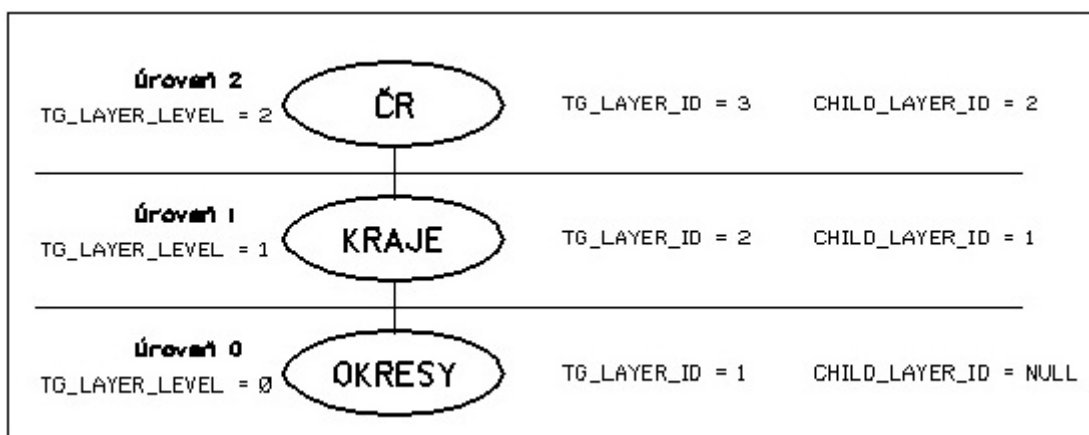
```
SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER(  
    topology          IN VARCHAR2,  
    table_name        IN VARCHAR2,  
    column_name       IN VARCHAR2,  
    topo_geometry_layer_type IN VARCHAR2,  
    relation_table_storage IN VARCHAR2 DEFAULT NULL,  
    child_layer_id    IN NUMBER DEFAULT NULL);
```

Tato procedura má šest parametrů:

- **topology** – název topologie, do které se vkládá geometrická vrstva topologických prvků. Tato topologie musí být vytvořena pomocí procedury SDO\_TOPO.CREATE\_TOPOLOGY.
- **table\_name** – název prvkové tabulky obsahující sloupec specifikovaný v *column\_name*
- **column\_name** – název sloupce (datového typu SDO\_TOPO\_GEOMETRY), který obsahuje topologické geometrie té vrstvy, která má být vložena do topologie
- **topo\_geometry\_layer\_type** – typ topologických prvků vkládané geometrické vrstvy (bod, linie, křivka, polygon nebo kolekce)
- **relation\_table\_storage** – parametry fyzického uložení užívané interně pro vytvoření tabulky *<topology-name>\_RELATION\$ table*. Pokud není tento parametr specifikován, vloží se defaultní hodnota (NULL).
- **child\_layer\_id** – jedná se o *tg\_layer\_id* vrstvy potomka, pokud mají vrstvy v topologii hierarchickou závislost.

Právě poslední parametr této procedury - **child\_layer\_id** – je důležitý při budování hierarchické struktury v topologii. Jako první se vkládá do topologie ta vrstva, která je v hierarchické struktuře na nejnižší úrovni (*level 0*). U této vrstvy (TG\_LAYER\_ID = 1) není

nutné tento parametr specifikovat; bez specifikovaného parametru CHILD\_LAYER\_ID se vrstva ukládá na nejnižší úroveň automaticky. Ve vyšších vrstvách hierarchie je nutné tento parametr specifikovat, jedná se o hodnotu atributu TG\_LAYER\_ID vrstvy potomka v pohledu *USER\_SDO\_TOPO\_METADATA*. Následně se tak vkládá rodičovská vrstva (na úrovni *level 1*, TG\_LAYER\_ID = 2), přičemž se specifikuje CHILD\_LAYER\_ID = 1, tj. TG\_LAYER\_ID jejího potomka. Obdobně se vkládá další vrstva (*level 2*, TG\_LAYER\_ID = 3) se specifikovaným CHILD\_LAYER\_ID = 2, případně další vrstvy na vyšších úrovních. Vždy se postupuje s asociací prvkových tabulek od nejnižší úrovně k nejvyšší [5]. Přehled atributů v rámci hierarchie je zobrazen na obr. 2.11.



obr. 2.11 Přehled atributů v hierarchii

Dalším krokem tvorby topologie, kde je nutné rozlišit, zda se jedná o topologii s hierarchickou strukturou, je *naplnění prvkových tabulek daty z prostorových tabulek*. Využívá se k tomu funkce SDO\_TOPO\_MAP.CREATE\_FEATURE. Tato funkce má dva formáty. První formát (s parametrem *geom* a bez parametru *dml\_condition*) je určen pro naplnění tabulek na nejnižší úrovni hierarchické struktury, nebo pro naplnění tabulek v topologii bez hierarchické struktury:

```
SDO_TOPO_MAP.CREATE_FEATURE(
    topology      IN VARCHAR2,
    table_name    IN VARCHAR2,
    column_name   IN VARCHAR2,
    geom          IN SDO_GEOMETRY
) RETURN SDO_TOPO_GEOMETRY;
```

Tento formát funkce SDO\_TOPO\_MAP.CREATE\_FEATURE má čtyři parametry:

- **topology** – název topologie, která obsahuje specifikovanou tabulku topologických prvků
- **table\_name** – název prvkové tabulky obsahující sloupec specifikovaný v *column\_name*
- **column\_name** – název sloupce datového typu SDO\_TOPO\_GEOMETRY
- **geom** - geometrický objekt (datový typ SDO\_GEOMETRY)

Pro naplnění tabulek na vyšších úrovních hierarchie je nutné použít druhý formát funkce SDO\_TOPO\_MAP.CREATE\_FEATURE:

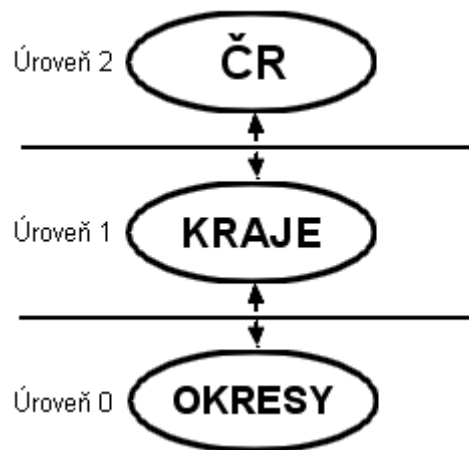
```
SDO_TOPO_MAP.CREATE_FEATURE(  
    topology      IN VARCHAR2,  
    table_name    IN VARCHAR2,  
    column_name   IN VARCHAR2,  
    dml_condition IN VARCHAR2  
) RETURN SDO_TOPO_GEOMETRY;
```

Tento formát neobsahuje parametr *geom*, ale podmínku *dml\_condition*. Pomocí této podmínky se vybírají řádky z vrstvy potomka, které se vkládají do rodičovské vrstvy. Podmínka se specifikuje v apostrofech a bez slova WHERE (např. 'ZKR\_KRAJ=' 'JC' ' ' ) .

Pomocí této funkce se zároveň naplňují tabulky <název\_topologie>\_EDGES\$, <název\_topologie>\_FACES\$, <název\_topologie>\_NODES\$, <název\_topologie>\_HISTORY\$ a <název\_topologie>\_RELATION\$. Také se naplňují pohledy <název\_topologie>\_RELATIONL a <název\_topologie>\_RELATLIDS [5].

## 2.3 Hierarchie vrstev v topologii H\_TOPO

Stěžejním bodem této diplomové práce je vytvoření hierarchické struktury v topologii, jejímiž jednotlivými vrstvami jsou okresy, kraje a Česká Republika. Byla tak vytvořena topologie H\_TOPO, ve které mají jednotlivé vrstvy mezi sebou hierarchickou závislost. Vrstvou na nejvyšší úrovni hierarchie je Česká Republika (*Úroveň 2*), která sestává ze 14 krajů ve vrstvě na nižší úrovni hierarchie (*Úroveň 1*). Tato vrstva dále sestává ze 77 okresů nacházejících se ve vrstvě, která je v rámci hierarchie na nejnižší úrovni (*Úroveň 0*), viz obr. 2.12. Vytvoření této topologie bude popsáno v kapitole 4.2.



obr. 2.12 Hierarchie vrstev v topologii H\_TOPO

### 3 Zpracování vstupních dat

Mým úkolem bylo převést SHP soubory do Oracle Spatial, využívajícího datového typu SDO\_GEOMETRY. Jako prostředí, ve kterém lze takový převod provést, jsem vzhledem k dosavadním zkušenostem s prací v Microstationu zvolila prostředí Bentley, které je mi blízké. Většina úkonů spojených s tímto převodem se provádí v prostředí Bentley Geospatial Administratoru (dále jen BGA), který byl pro mě úplnou novinkou. Jedná se o aplikaci, která poskytuje uživatelům rozhraní pro vývoj vysoce interaktivních aplikací bez nutnosti programování. Vytváří se zde XFM model (XFM = XML Feature Modeling), čili specifické uživatelské rozhraní, ve kterém se následně Bentley PowerMap spustí [15].

Bentley PowerMap (dále jen PowerMap) je software vybudovaný na platformě Microstation. Nabízí 2D nebo 3D mapové produktivní prostředí. Umí pracovat s DGN a DWG výkresy, umí načítat, či jako referenční připojovat SHP soubory. Dále je možné importovat pestrou škálu rastrových i vektorových formátů. Je zde k dispozici několik druhů nástrojů pro topologickou analýzu. Topologický model je založen na topologii typu Oracle Spatial. Je zde k dispozici řada zajímavých nástrojů, např. správce map, správce geoprostorových dat, editování dat uložených v Oracle Spatial, topologické analýzy, prostorové analýzy, mapová zobrazení a transformace souřadných systémů, spolupráce s jinými GIS formáty a další [1].

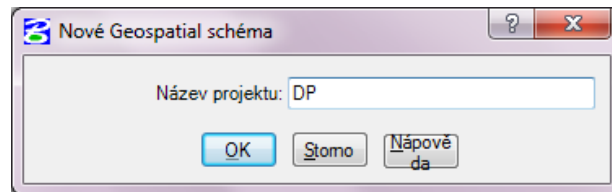
#### 3.1 Založení nového projektu v Bentley Geospatial Administrator

Prvním krokem výše zmíněného převodu bylo založení nového projektu v **BGA** (pojmenovaného „DP“, viz obr. 3.1) a vytvoření nového pracovního prostředí *workspace* (pojmenovaného „vykres“, viz obr. 3.2, obr. 3.3). Takto vytvořené pracovní prostředí jsem vyexportovala, export se defaultně provedl do dané adresářové struktury prostřednictvím výkresu work.dgn:

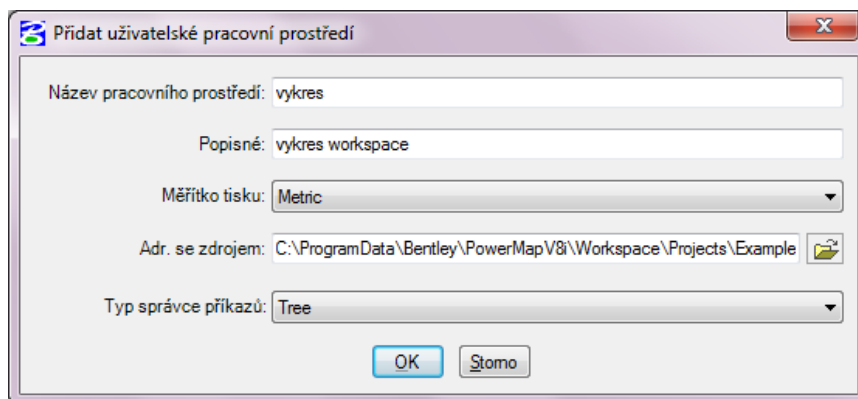
```
C:\Documents AND Settings\All Users\Bentley\PowerMapV8i\WorkSpace\Projects\Examples\Geospatial\DP\vykres\dgn\work.dgn.
```

Tento výkres slouží pouze k uložení nastavení pracovního prostředí.

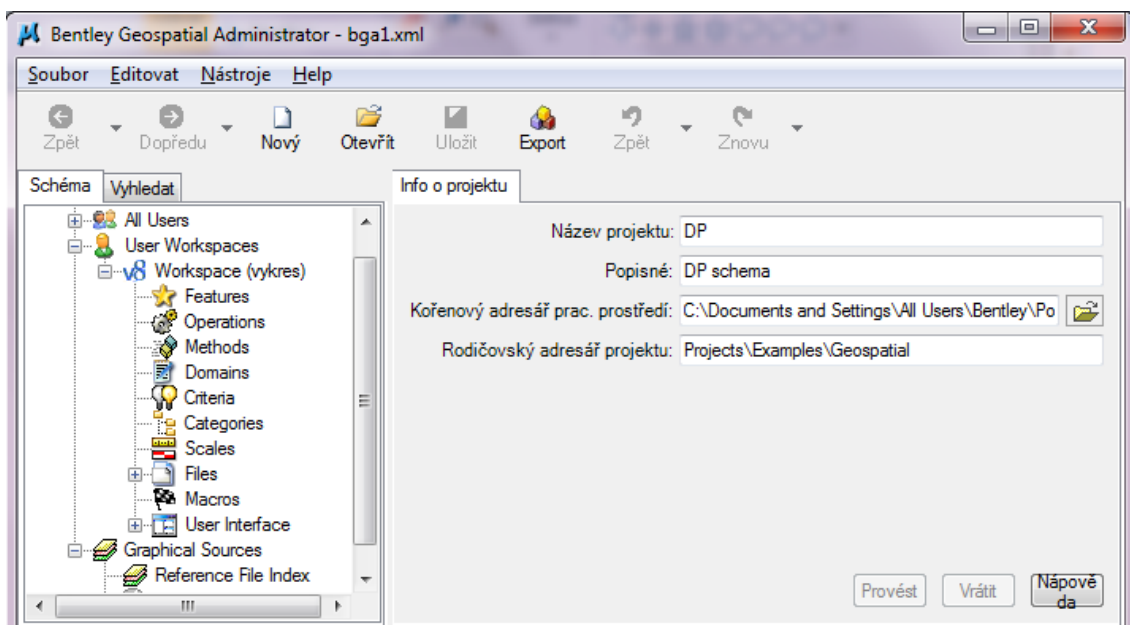
Soubor (v rámci BGA taktéž nazývaný projekt), v rámci kterého jsem projekt „DP“ vytvářela, jsem pojmenovala „bga1.xml“ a uložila.



obr. 3.1 Založení nového projektu



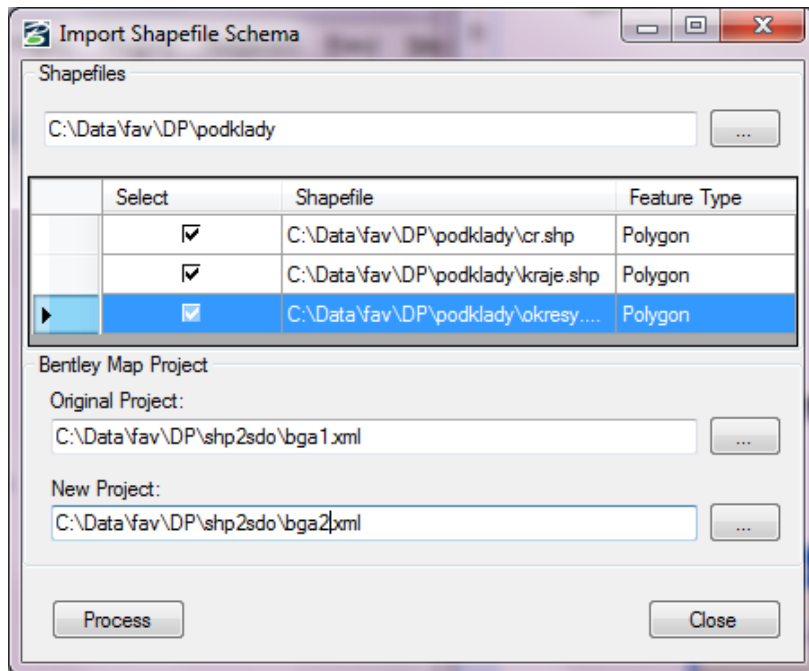
obr. 3.2 Vytvoření nového pracovního prostředí (workspace)



obr. 3.3 Nové založený projekt v BGA a v něm vytvořeného pracovního prostředí (workspace)

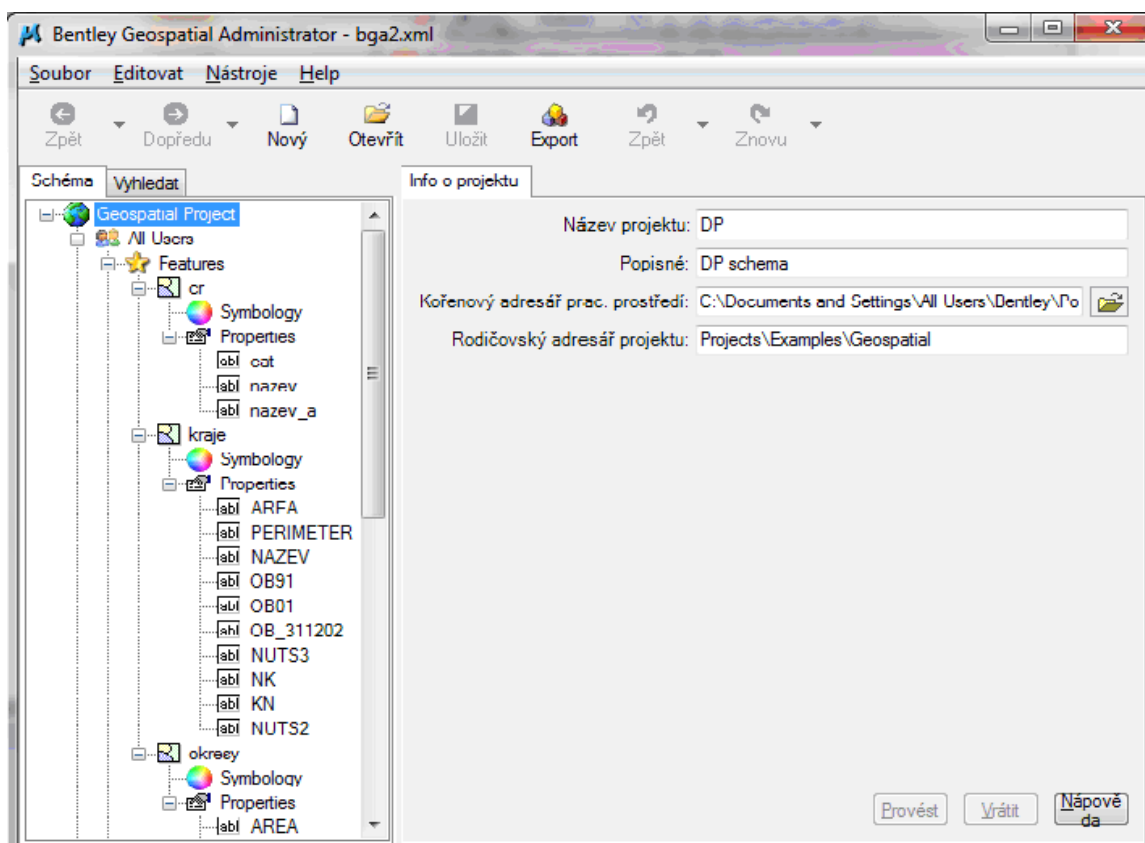
### 3.2 Načtení \*.shp výkresů do BGA

Druhým krokem bylo načtení \*.shp výkresů do BGA prostřednictvím programu **TXml.exe**, poskytnutým firmou Bentley. Tyto soubory (cr.shp, kraje.shp, okresy.shp) se vyexportovaly do nového souboru bga2.xml, který přebírá nastavení z původního souboru bga1.xml a jím vyexportovaného pracovního prostředí (obr. 3.4).



obr. 3.4 Import shp výkresů do BGA prostřednictvím Txml.exe

V bga2.xml se tak vytvořily nezaregistrované mapové objekty - XFM Features (zobrazí se v záložce „Features“) a jejich atributy – Properties (viz obr. 3.5). Soubor bga1.xml zůstal zatím nezměněn.



obr. 3.5 Vytvořené mapové objekty v prostředí BGA, včetně jejich atributů (Properties)

Také toto pracovní prostředí bylo nutné vyexportovat, export se provedl do stejného adresáře jako v prvním kroku, do výkresu work.dgn.

### 3.3 Vygenerování skriptů pro vytvoření tabulek

Třetím krokem bylo vygenerování skriptů potřebných pro vytvoření tabulek. To bylo provedeno prostřednictvím skriptu **FROM\_XFM\_to\_Oracle\_Spatial\_Tables.cmd**, taktéž poskytnutým firmou Bentley. Nejprve bylo nutné tento soubor zeditovat, tzn. nastavit cesty k souborům a adresářům dle konkrétního nastavení cest v uživatelském počítači (viz obr. 3.6):

*SET dgn2sdo* – cesta k souboru Dgn2Sdo.exe<sup>1</sup>

*SET dgnfile* – cesta k souboru work.dgn (soubor s vyexportovaným pracovním prostředím)

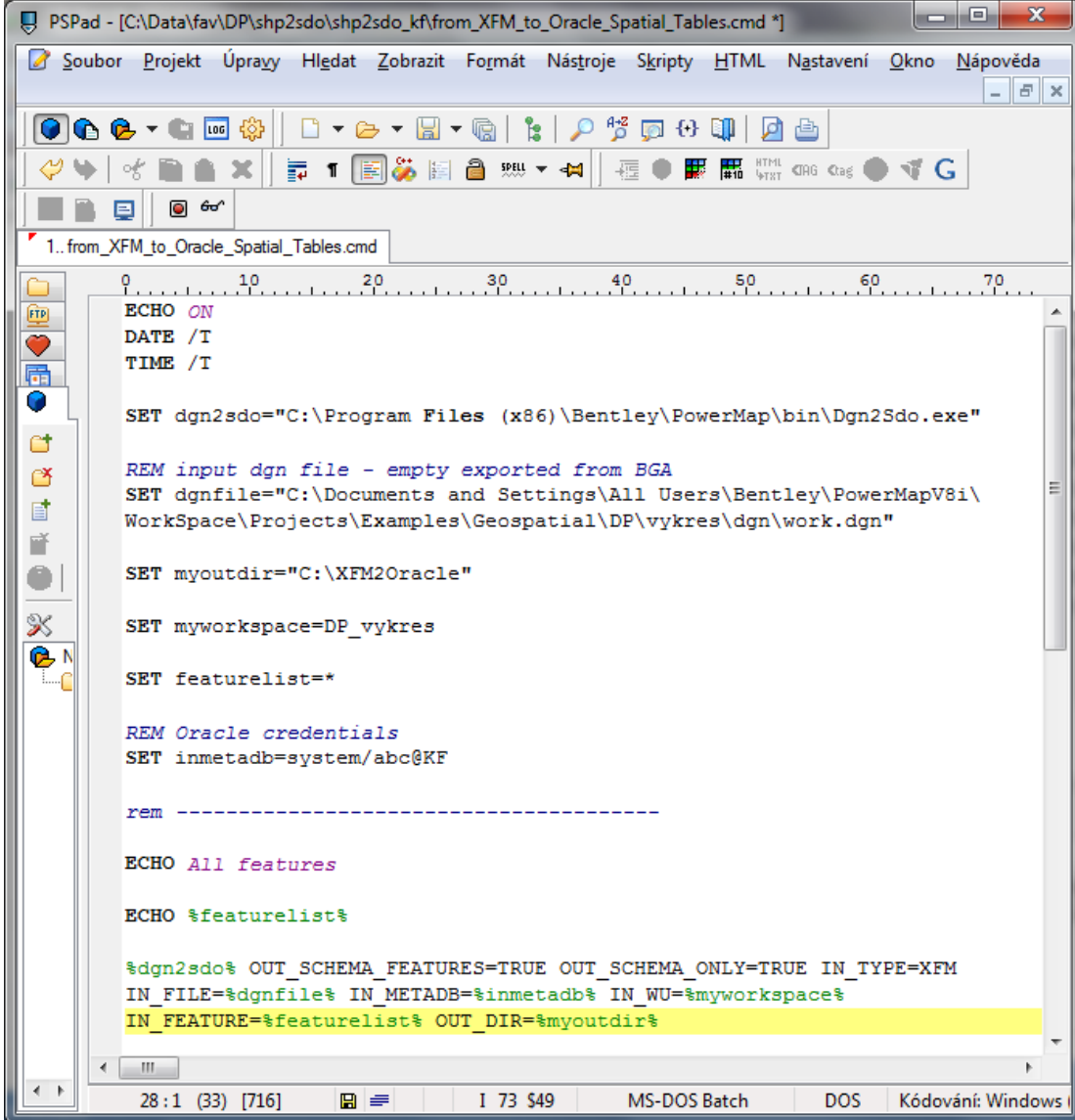
<sup>1</sup> Dgn2Sdo.exe umožňuje načíst soubory Microstationu (geometrická a atributová data) do databáze Oracle Spatial. Více v [2].



*SET myoutdir* – cesta k adresáři, kam se uloží vygenerované skripty

*SET myworkspace* - název projektu a pracovního prostředí

*SET inmetadb* - nastavení připojení k databázi, se kterou se bude pracovat



```
1..from_XFM_to_Oracle_Spatial_Tables.cmd
0      10      20      30      40      50      60      70
ECHO ON
DATE /T
TIME /T

SET dgn2sdo="C:\Program Files (x86)\Bentley\PowerMap\bin\Dgn2Sdo.exe"

REM input dgn file - empty exported from BGA
SET dgnfile="C:\Documents and Settings\All Users\Bentley\PowerMapV8i\
Workspace\Projects\Examples\Geospatial\DP\vykres\dgn\work.dgn"

SET myoutdir="C:\XFM2Oracle"

SET myworkspace=DP_vykres

SET featurelist=*

REM Oracle credentials
SET inmetadb=system/abc@KF

rem -----

ECHO All features

ECHO %featurelist%

%dgn2sdo% OUT_SCHEMA_FEATURES=TRUE OUT_SCHEMA_ONLY=TRUE IN_TYPE=XFM
IN_FILE=%dgnfile% IN_METADB=%inmetadb% IN_WU=%myworkspace%
IN_FEATURE=%featurelist% OUT_DIR=%myoutdir%
```

obr. 3.6 Editace souboru FROM\_XFM\_to\_Oracle\_Spatial\_Tables.cmd

Po spuštění skriptu **FROM\_XFM\_to\_Oracle\_Spatial\_Tables.cmd** se vygenerovala sada souborů - hlavní skript MasterLoad.BAT a další skripty pro vytvoření tabulek a prostorových indexů (viz obr. 3.7). Vše se uloží do adresáře nastaveného v *SET myoutdir* (viz obr. 3.8).

```

C:\Windows\system32\cmd.exe

C:\Data\fav\DP\shp2sdo\shp2sdo_kf>"C:\Program Files (x86)\Bentley\PowerMap\bin\D
gn2Sdo.exe" OUT_SCHEMA_FEATURES=TRUE OUT_SCHEMA_ONLY=TRUE IN_TYPE=XFM IN_FILE="C
:\Documents and Settings\All Users\Bentley\PowerMap08i\WorkSpace\Projects\Exampl
es\Geospatial\DP\vykres\dgn\work.dgn" IN_METADB=system/abcCKF IN_WU=DP_vykres IN
_FEATURE=* OUT_DIR="C:\XFM2Oracle"
DGN2SDO 8.11.5.19 bude pracovat s PowerMapem verze (08.11.05.19).
25.7.2011 14:18:29
Domov efektivního prac. prostředí = (C:\ProgramData\Bentley\PowerMap08i\Workspac
e\Projects\Examples\Geospatial\DP\vykres\dgn\work.dgn)
Ústupní adresář byl nastaven na (C:\XFM2Oracle)
Zavádím Bentley PowerMap
Otevírám výkres "C:\Documents and Settings\All Users\Bentley\PowerMap08i\WorkSpa
ce\Projects\Examples\Geospatial\DP\vykres\dgn\work.dgn"
Generování skriptu pro (KATERINA_KRAJE).
Generování skriptu pro (ReportTable).
Generování skriptu pro (KATERINA_OKRESY).
Generování skriptu pro (KATERINA_CR).
Generování hlavního dávkového souboru (MasterLoad).
PowerMap stroj byl ukončen.
25.7.2011 14:19:12

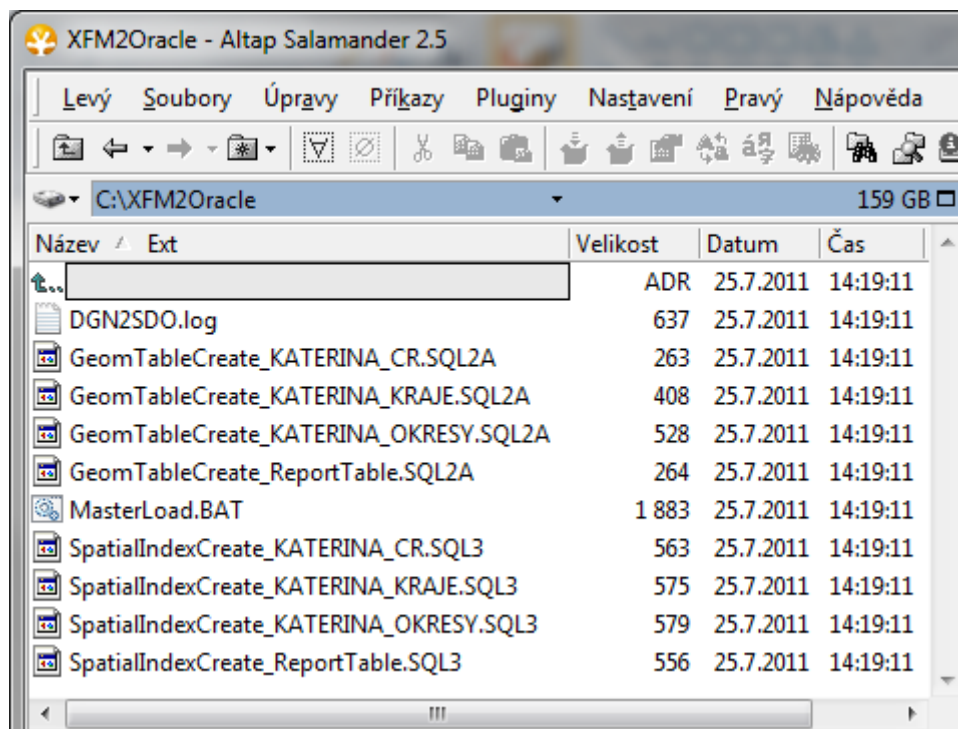
C:\Data\fav\DP\shp2sdo\shp2sdo_kf>DATE /T
po 25.07.2011

C:\Data\fav\DP\shp2sdo\shp2sdo_kf>TIME /T
14:19

C:\Data\fav\DP\shp2sdo\shp2sdo_kf>PAUSE
Pokračujte stisknutím libovolné klávesy...

```

obr. 3.7 Spuštění skriptu FROM\_XFM\_to\_Oracle\_Spatial\_Tables.cmd



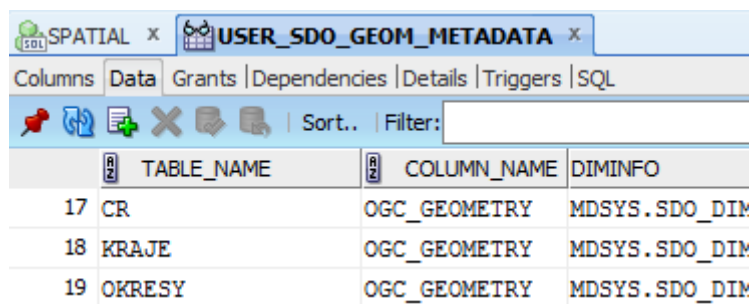
obr. 3.8 Vygenerované skripty pro vytvoření tabulek a prostorových indexů v adresáři XFM2Oracle

### 3.4 Vytvoření tabulek v SQL Developeru

Čtvrtým krokem bylo vytvoření samotných tabulek CR, KRAJE a OKRESY v SQL Developeru. To jsem provedla spuštěním skriptu **MasterLoad.BAT** s upřesněním přístupu k databázi z příkazové řádky (viz obr. 3.9). Soubor MasterLoad.BAT, vygenerovaný v kroku popsaném v odstavci 3.3., spustil skripty vygenerované spolu s ním. Vytvořené Oracle Spatial tabulky jsou téměř identickými kopiemi SHP struktur, mají stejné atributy, jen je zde navíc sloupec datového typu SDO\_GEOMETRY. Zatím jsou tyto tabulky prázdné (viz obr. 3.10).



obr. 3.9 Spuštění MasterLoad.BAT

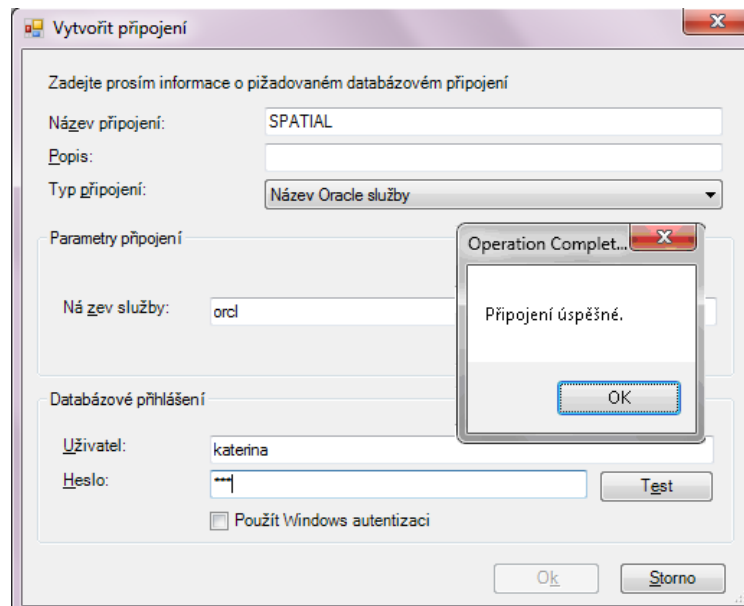
A screenshot of the SQL Developer interface showing the metadata for the 'USER\_SDO\_GEOM\_METADATA' table. The table has three columns: TABLE\_NAME, COLUMN\_NAME, and DIMINFO. The data rows are as follows:

TABLE_NAME	COLUMN_NAME	DIMINFO
CR	OGC_GEOMETRY	MDSYS.SDO_DIM
KRAJE	OGC_GEOMETRY	MDSYS.SDO_DIM
OKRESY	OGC_GEOMETRY	MDSYS.SDO_DIM

obr. 3.10 Vygenerované tabulky CR, KRAJE, OKRESY

### 3.5 Vytvoření databázového připojení v BGA

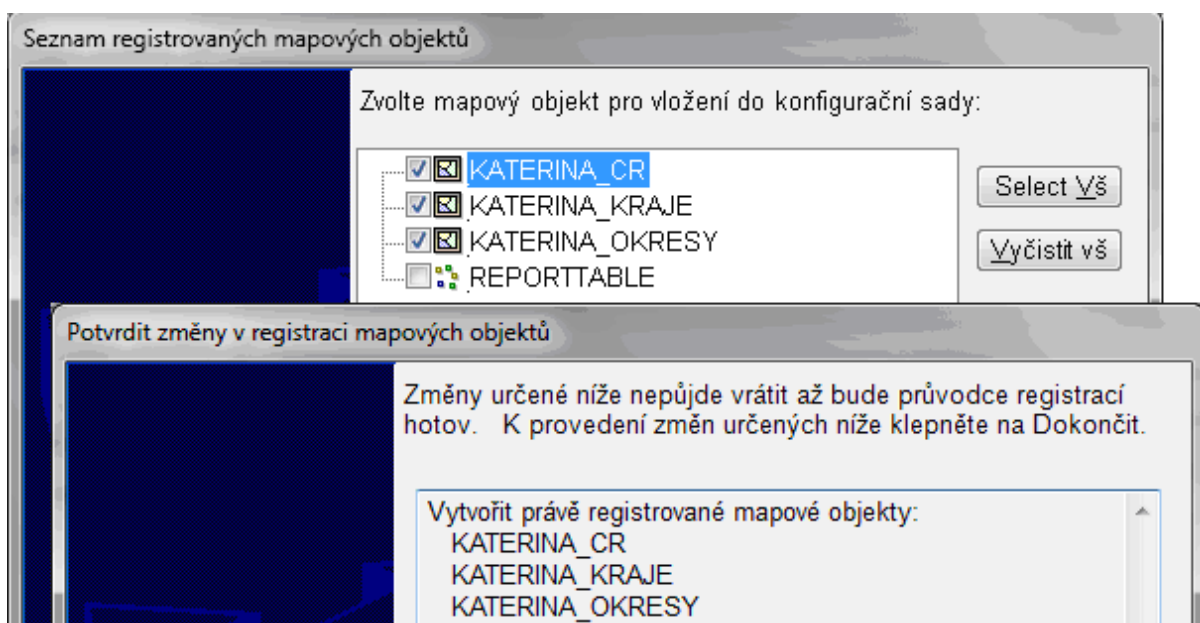
Pátým krokem bylo vytvoření databázového připojení v BGA (viz obr. 3.11), v souboru bga1.xml, v záložce *Graphical Sources – Oracle Spatial Connection – Přidat*.



obr. 3.11 Vytvoření databázového připojení v BGA

Nové mapové objekty, resp. Oracle Spatial tabulky, bylo nutné zaregistrovat (viz obr. 3.12):

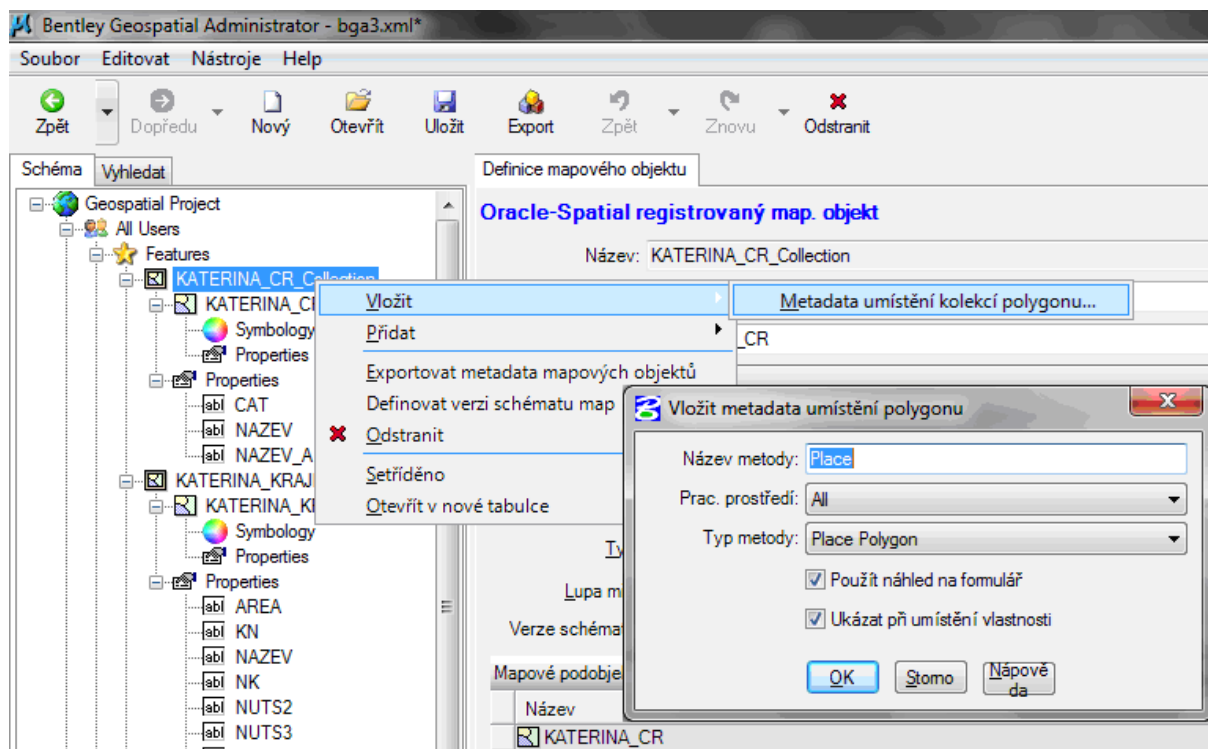
*Graphical Sources – Oracle Spatial Connection – Named Connection (SPATIAL) – Registrovat mapové objekty.*



obr. 3.12 Registrace mapových objektů

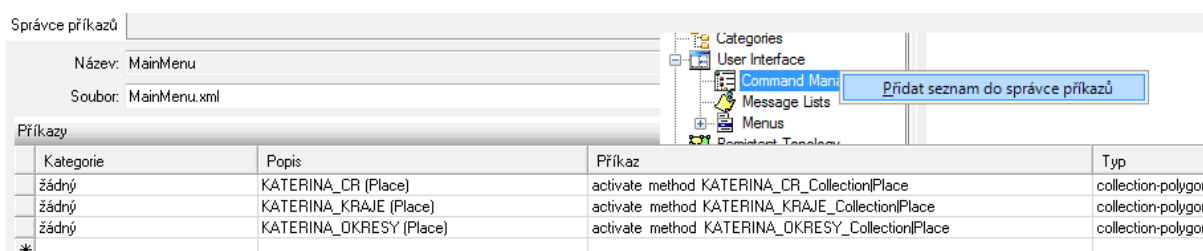
Takto zaregistrované mapové objekty se zobrazí v záložce *Features*. Dále bylo nutné vložit metadata umístění kolekcí polygonů (Insert Polygon Placement Metadata) pro každý

registrovaný objekt v záložce *Features* (viz obr. obr. 3.13). Generují se tak metody a operace, které budou použity při vkládání objektů, jejich editaci a popisování. Zde se mohou navolit různé parametry, např. jakým způsobem a kdy se budou zobrazovat vlastnosti objektů atd.



obr. 3.13 Vložení metadat umístění kolekcí polygonu

V záložce *User Interface – Command Manager – Přidat seznam do správce příkazů* jsem přidala příkazy pro umístění polygonových kolekcí, včetně zahrnutí názvu metody do popisu (viz obr. 3.14).



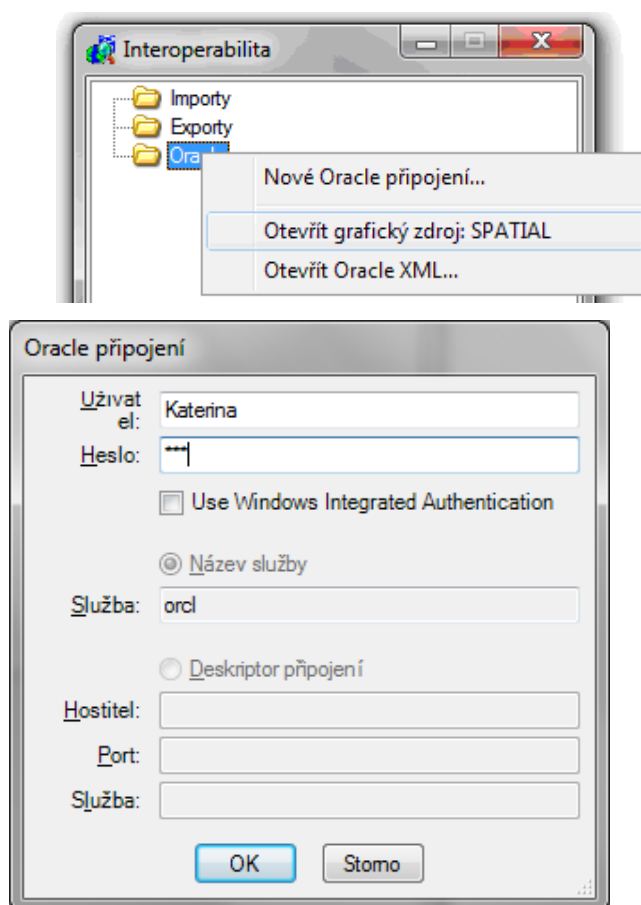
obr. 3.14 Přidání seznamu do správce příkazů

V tuto chvíli již je projekt v BGA dostatečně nadefinován. Je vytvořeno pracovní prostředí (workspace) „vykres“, které se musí opět vyexportovat. Export workspace je opět

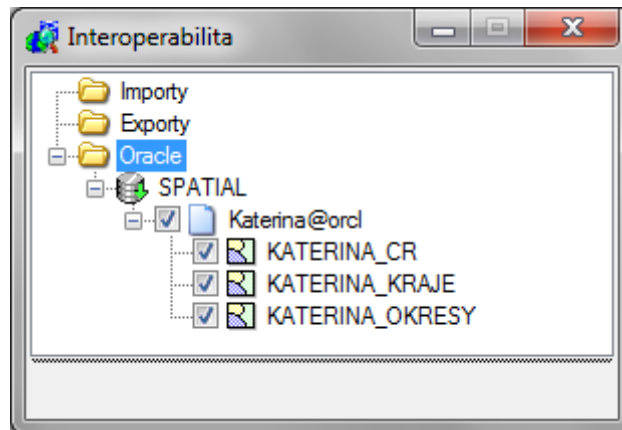
proveden defaultně do adresářové struktury do souboru work.dgn jako v kapitole 3.2. Nyní již je čas spustit Bentley PowerMap z prostředí BGA.

### 3.6 Import výkresů do PowerMapu

Šestý krok probíhal již v prostředí samotného PowerMapu. V nabídce *Soubor – Otevřít Oracle Spatial* jsem spustila dialogové okno *Interoperabilita* (viz obr. 3.15). V tomto dialogu *Oracle – Otevřít grafický zdroj: SPATIAL* jsem zadala parametry připojení přes službu *orcl* a dostala se tak k mapovým objektům (features) zaregistrovaným v BGA (viz obr. 3.16), resp. k Oracle Spatial tabulkám.

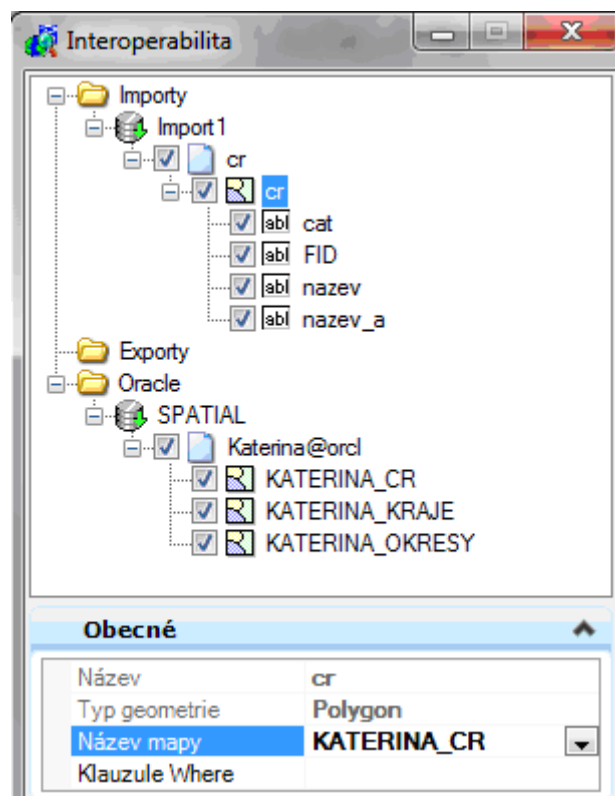


obr. 3.15 Připojení PowerMapu k Oracle Spatial

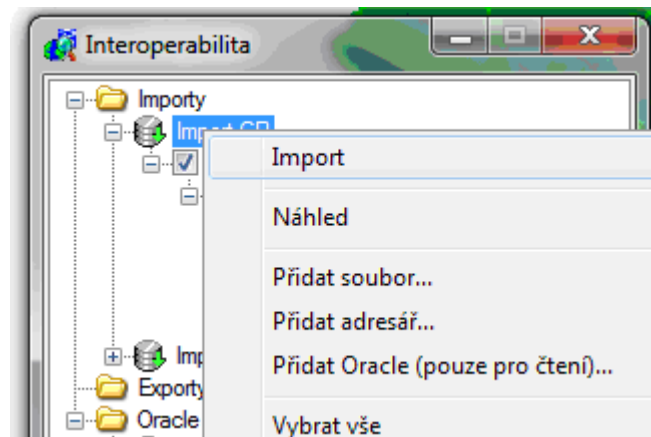


obr. 3.16 Zobrazení zaregistrovaných mapových objektů

Do PowerMapu jsem si naimportovala \*.shp výkresy, taktéž pomocí dialogu *Interoperabilita – Importy – Nový import*. Zde se také zobrazují atributy daného výkresu, které je nutné přidružit k atributům zaregistrovaných mapových objektů v BGA (viz obr. 3.17), včetně jejich přidružení ke konkrétní Oracle Spatial tabulce zaregistrované v BGA. Takto provedený Import je vhodné uložit (*import\_cr.impx*, *import\_kraje.impx*, *import\_okresy.impx*). Při příštím načítání se již načte konkrétní Import a není nutné opětovné přidružování atributů.

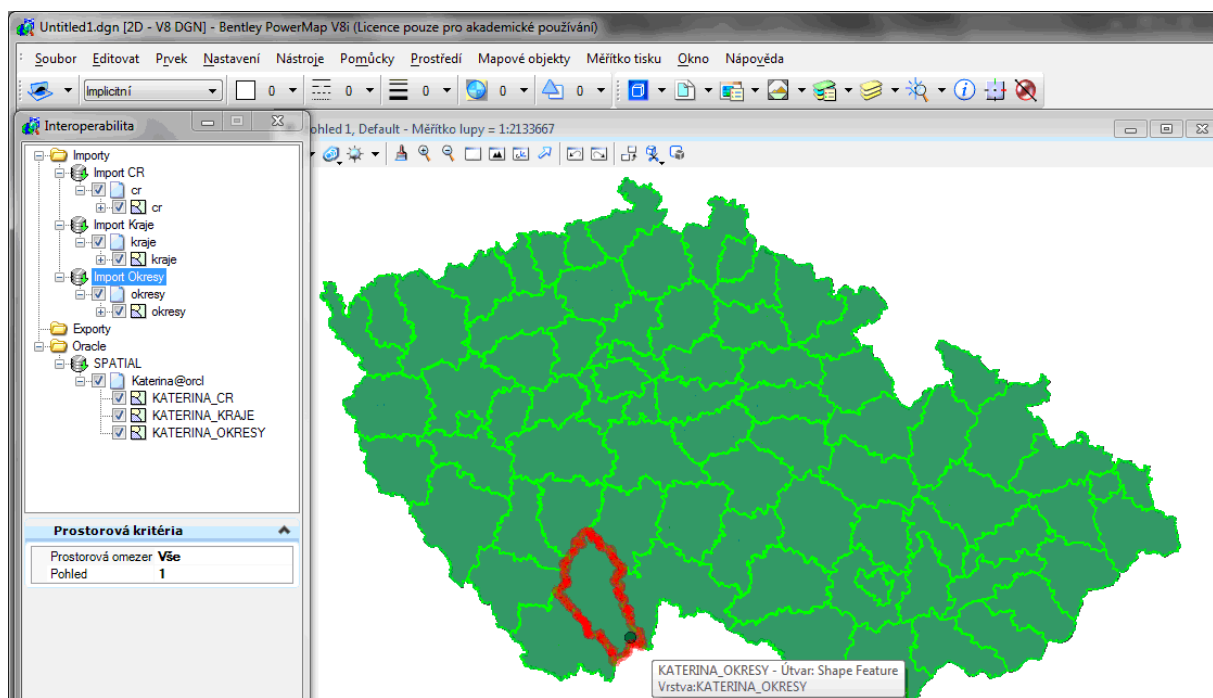


obr. 3.17 Přidružení atributů k zaregistrovaným features




obr. 3.18 Import dat do výkresu

Po Importech jednotlivých vrstev (viz obr. 3.18) jsem načetla již samotnou kresbu do okna v PowerMapu (viz obr. 3.19).



obr. 3.19 Vrstvy načtené do výkresu

Pomocí nástroje *Analyzovat*  je možné jednotlivé prvky mapy analyzovat (viz obr. 3.20). Tato analýza poskytuje informace (hodnoty jednotlivých atributů) o dotazovaném prvku, např. název okresu, počet obyvatel, rozloha, příslušející kraj atd..

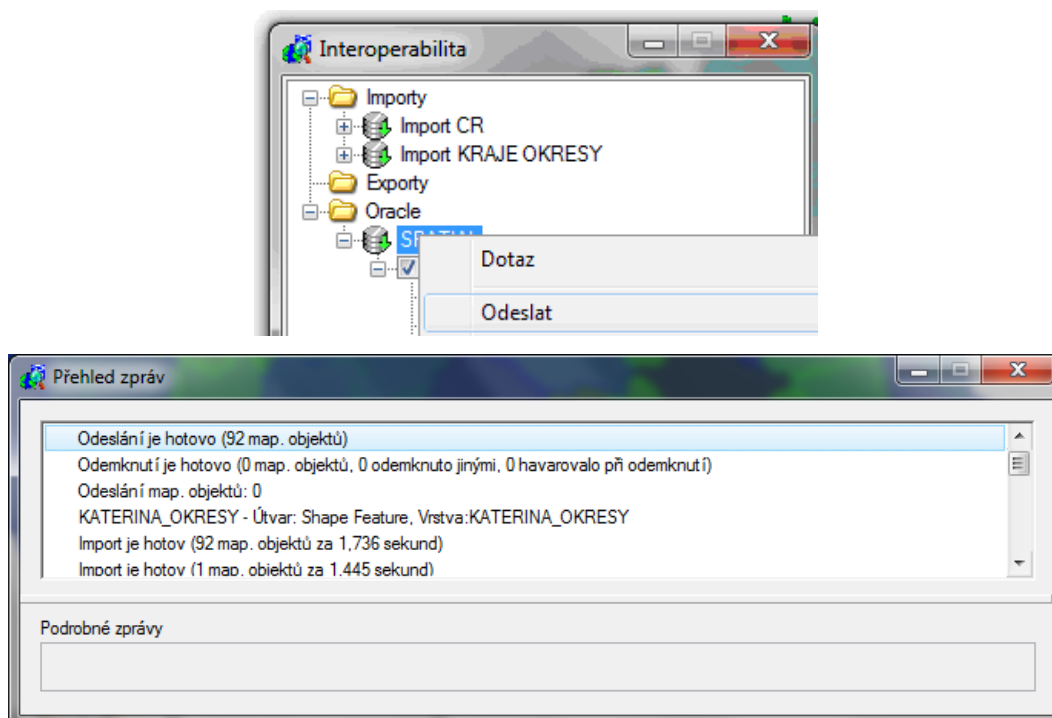


Vlastnost	Hodnota
AREA	1626072704
PERIMETER	305562
NAZEV	České Budějovice
OB91	173386
OB01	180383
OB_311202	178523
OKRES	3301
NUTS4	CZ0311
NUTS3	CZ031
NUTS2	CZ03
KRAJ1960	3300
NK	JC
KN	03
KNOK	0301
NAZKR	Jihočeský

obr. 3.20 Analýza prvků načtených do PowerMapu

### 3.7 Odeslání dat do databáze Oracle Spatial

Posledním krokem bylo již samotné odeslání dat do databáze Oracle Spatial (viz obr. 3.21).



obr. 3.21 Odeslání dat z PowerMapu do Oracle Spatial

Po načtení dat odeslaných z PowerMapu se naplnily tabulky *cr*, *kraje* a *okresy*. V tabulce *cr* se doplnil pouze jeden záznam (hranice republiky), tabulka *kraje* obsahuje 14 nových krajů (viz obr. 3.22) a tabulka *okresy* obsahuje 77 nových okresů.

	AREA	PERI...	NAZEV	OB91	OB01	OB_...	NL
1	11013000000	1019320	Středočeský	1112882	1128766	1128674	CZ02:
2	31629000000	419095	Liberecký	425120	430820	427321	CZ05:
3	55665000000	541629	Moravskoslezský	1283271	1277412	1262660	CZ08:
4	47665000000	554071	Královéhradecký	552809	554594	548437	CZ05:
5	70649000000	736104	Jihomoravský	1136832	1134170	1121792	CZ06:
6	75659000000	582039	Plzeňský	558307	553848	549374	CZ03:
7	51414000000	643412	Olomoucký	642796	642346	636750	CZ07:
8	53411000000	601702	Ústecký	824461	826869	819712	CZ04:
9	69243000000	577442	Vysočina	521068	521308	517630	CZ06:

obr. 3.22 Naplněná tabulka kraje v SQL Developeru

Celý proces převodu SHP souborů do SDO\_GEOMETRY prostřednictvím Bentley Geospatial Administratoru a Bentley PowerMapu je poměrně komplikovaný, což je dáno především tvorbou projektu v BGA. To je však nutné k vytvoření pracovního prostředí (workspace), ve kterém se následně spustí Bentley PowerMap. Přesně nadefinované uživatelské rozhraní Bentley PowerMapu umožňuje propojení s databází Oracle Spatial. Zde je poté možná vizualizace dat uložených v databázi a také jejich analýza.

Za zmínku stojí i nový produkt firmy Bentley - MicroStation V8i SELECTSeries 3, který byl představen v říjnu minulého roku na Bentley Foru 2011. Tento produkt přináší zlepšení práce s ESRI SHP soubory. Myslím si, že produkty firmy Bentley mají velkou budoucnost ve spolupráci s Oracle Spatial [14].

## 4 Topologie vytvořená z prostorových geometrií

Cílem této kapitoly je popsání převodu dat uložených pomocí datového typu SDO\_GEOMETRY na data využívající pro uložení prostorové informace datový typ SDO\_TOPO\_GEOMETRY.

### 4.1 Příprava dat

Pro vytvoření topologie z prostorových geometrií bylo nejdříve nutné provést standardní operace pro přípravu dat použitých v Oracle Spatial, popsané v [5].

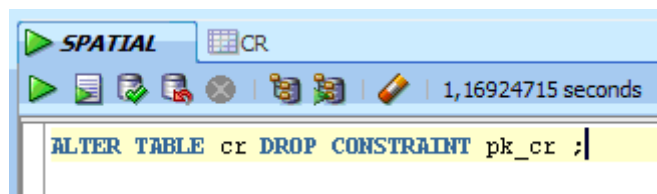
- vytvoření prostorových tabulek – tabulky jsou již vytvořené, provedena byla pouze jejich úprava
- aktualizace prostorových metadat (pohled USER\_SDO\_GEOM\_METADATA)
- validace prostorových dat
- vytvoření prostorových indexů – tyto indexy byly vytvořené při převodu z prostředí MicroStationu

#### 4.1.1 Úprava prostorových tabulek

Vzhledem k tomu, že data byla převáděna do formátu SDO\_GEOMETRY pomocí prostředí MicroStationu – Bentley Geospatial Administrator, byly vytvořeny primární klíče na již nepotřebných atributech XFM\_ID (viz obr. 4.1). Proto jsem se rozhodla tato již nepotřebná data odstranit a změnit primární klíče (viz obr. 4.2, obr. 4.3).

Column Name	Data Type	Nullable	Data Default	COLUMN ID	Primary Key	COMMENTS
XFM_ID	VARCHAR2(36 BYTE)	No	(null)	5	1 (null)	
CAT	NUMBER	Yes	(null)	1	(null) (null)	
NAZEV	VARCHAR2(32 BYTE)	Yes	(null)	2	(null) (null)	
NAZEV_A	VARCHAR2(32 BYTE)	Yes	(null)	3	(null) (null)	
OGC_GEOMETRY	SDO_GEOMETRY	Yes	(null)	4	(null) (null)	

obr. 4.1 Primární klíč na XFM\_ID vytvořený při převodu do SDO\_GEOMETRY



obr. 4.2 Zrušení primárního klíče na tabulce CR

```
ALTER TABLE cr DROP CONSTRAINT pk_cr ;
ALTER TABLE kraje DROP CONSTRAINT pk_kraje ;
ALTER TABLE okresy DROP CONSTRAINT pk_okresy ;
```

```
ALTER TABLE cr ADD CONSTRAINT pk_cr PRIMARY KEY (CAT);
ALTER TABLE kraje ADD CONSTRAINT pk_kraje PRIMARY KEY (NK);
ALTER TABLE okresy ADD CONSTRAINT pk_okresy PRIMARY KEY (OKRES);
```

```
ALTER TABLE cr DROP (xfm_id) ;
ALTER TABLE kraje DROP (xfm_id) ;
ALTER TABLE okresy DROP (xfm_id) ;
```

Column Name	Data Type	Nullable	Data Default	COLUMN ID	Primary Key	COMMENTS
CAT	NUMBER	No	(null)	1	1 (null)	
NAZEV	VARCHAR2(32 BYTE)	Yes	(null)	2	(null) (null)	
NAZEV_A	VARCHAR2(32 BYTE)	Yes	(null)	3	(null) (null)	
OGC_GEOMETRY	SDO_GEOMETRY	Yes	(null)	4	(null) (null)	

obr. 4.3 Změněný primární klíč a zrušený sloupec XFM\_ID

## 4.1.2 Aktualizace prostorových metadat

Dále bylo nutné aktualizovat metadata o prostorových datech v pohledu USER\_SDO\_GEOM\_METADATA. Bez této aktualizace není možné vytvářet prostorové indexy. Takto byla aktualizována metadata pro tabulky CR, KRAJE a OKRESY (viz obr. 4.4).

```
INSERT INTO user_sdo_geom_metadata (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES (
    'CR',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    ),
    NULL -- SRID
);
```

```
INSERT INTO user_sdo_geom_metadata (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES (
    'KRAJE',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    ),
    NULL -- SRID
);
```

```
INSERT INTO user_sdo_geom_metadata (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES (
    'OKRESY',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    ),
    NULL -- SRID
);
```

TABLE_NAME	COLUMN_NAME	DIMINFO	SRID
1 CR	GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)
2 CR	OGC_GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)
3 KRAJE	OGC_GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)
4 OKRESY	OGC_GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)
5 REPORITABLE	OGC_GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)
6 OKRESY	GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)
7 KF_CR_TOPO	FEATURE	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,0,1,0.00005),MDSYS.SDO_DIM_ELEMENT(Y,0,1,0.00005))	(null)
8 KF_KRAJE_TOPO	FEATURE	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,0,1,0.00005),MDSYS.SDO_DIM_ELEMENT(Y,0,1,0.00005))	(null)
9 KF_OKRESY_T...	FEATURE	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,0,1,0.00005),MDSYS.SDO_DIM_ELEMENT(Y,0,1,0.00005))	(null)
10 KF_TOPO_FACE\$	MBR_GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,0,1,0.00005),MDSYS.SDO_DIM_ELEMENT(Y,0,1,0.00005))	(null)
11 KRAJE	GEOMETRY	MDSYS.SDO_DIM_ELEMENT(MDSYS.SDO_DIM_ELEMENT(X,-5000000,5000000,5E-8),MDSYS.SDO_DIM_ELEMENT(Y,-5000000,5000000...	(null)

obr. 4.4 Aktualizovaná metadata o prostorových datech v pohledu USER\_SDO\_GEOM\_METADATA

### 4.1.3 Validace prostorových dat

Dalším krokem byla validace vrstev, která se provádí pomocí funkce SDO\_GEOM.VALIDATE\_LAYER\_WITH\_CONTEXT. Prochází se tak geometrické sloupce (geom\_column) a zjišťuje se, zda uložená geometrie splňuje definovaná pravidla pro geometrické objekty. Validace prováděná touto funkcí zahrnuje validaci prováděnou funkcí SDO\_GEOM.VALIDATE\_GEOMETRY\_WITH\_CONTEXT (tj. kontrolu typové a geometrické konzistence) a dále zda hodnota SRID odpovídá hodnotě ve sloupci DIMINFO v pohledu USER\_SDO\_GEOM\_METADATA. Funkce přidá řádku do předem vytvořené tabulky, pokud nalezne nekonzistentní data. Vypíše číslo řádky, kde se nekonzistentní data nalézají a dále číslo chyby (*Oracle error message*) [10].

```

create table val_results (
    sdo_rowid ROWID,
    result VARCHAR2(2000)
);

call
    SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('CR','OGC_GEOMETRY',
    'VAL_RESULTS');

SELECT * FROM val_results;
drop table val_results;

create table val_results (
    sdo_rowid ROWID,
    result VARCHAR2(2000)
);

call
    SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('KRAJE','OGC_GEOMETRY',
    'VAL_RESULTS');

```

```

SELECT * FROM val_results;
drop table val_results;

create table val_results (
    sdo_rowid ROWID,
    result VARCHAR2(2000)
);

call
    SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('OKRESY', 'OGC_GEOMETRY',
    'VAL_RESULTS');

SELECT * FROM val_results;
drop table val_results;

```

Všechny testované geometrické objekty jsou validní.

#### 4.1.4 Vytvoření prostorových indexů na tabulkách CR, KRAJE, OKRESY

Tyto prostorové indexy již byly vytvořené při převodu dat z prostředí MicroStationu. Vytvořily se tak indexy: CR\_SIDX, KRAJE\_SIDX a OKRESY\_SIDX (viz obr. 4.5). Pokud by se nevytvořily, případně by byl použit jiný způsob převodu dat, je možné je vytvořit pomocí následujících skriptů:

```

CREATE INDEX cr_idx
ON cr(ogc_geometry)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

CREATE INDEX kraje_idx
ON kraje(ogc_geometry)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

CREATE INDEX okresy_idx
ON okresy(ogc_geometry)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```

The screenshot shows the Oracle Spatial interface with the 'OKRESY\_SIDX' index selected. The 'Columns' tab is active, displaying a table with the following data:

Index Owner	Index Name	Table Owner	Table Name	Column Name	Column Posi...	Descend
KATERINA	OKRESY_SIDX	KATERINA	OKRESY	OGC_GEOMETRY	1	ASC

obr. 4.5 Vytvořený prostorový index na tabulce OKRESY

## 4.2 Vytvoření vlastní topologie

Hlavními kroky při vytváření topologie z prostorových geometrií (z SDO\_GEOMETRY) jsou:

- vytvoření topologie s použitím procedury SDO\_TOPO.CREATE\_TOPOLOGY
- vytvoření „universe face“ F0
- vytvoření prvkových tabulek pro každou vrstvu topologických prvků v topologii
- asociace prvkových tabulek s topologií, s použitím procedury SDO\_TOPO.ADD\_TOPO\_GEOMETRY\_LAYER pro každou prvkovou tabulku
- vytvoření TopoMap objektu a načtení celé topologie do mezipaměti
- načtení prvkových tabulek, vložení dat z prostorových tabulek s použitím funkce SDO\_TOPO\_MAP.CREATE\_FEATURE
- inicializace topologických metadat s použitím procedury SDO\_TOPO.INITIALIZE\_METADATA

### 4.2.1 Vytvoření topologie s použitím procedury SDO\_TOPO.CREATE\_TOPOLOGY

Nová topologie H\_TOPO byla vytvořena pomocí procedury SDO\_TOPO.CREATE\_TOPOLOGY:

```
EXECUTE SDO_TOPO.CREATE_TOPOLOGY('H_TOPO', 0.00005);
```

Tato topologie se zobrazila v pohledu USER\_SDO\_TOPO\_INFO (viz obr. 4.6).

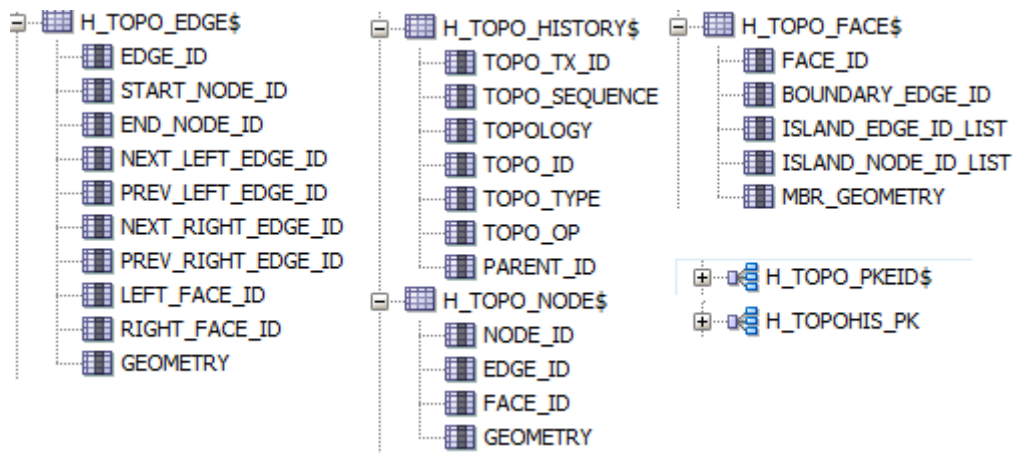
OWNER	TOPOLOGY	TO...	TOLERANCE	SRID	TABLE_SCHEMA	TABLE_NAME	COLUMN...	TG_LAYE...	TG_LAYER_TYPE	TG_LAYER_LE...
1 KATERINA	H_TOPO	10	0,00005 (null)	KATERINA	H_OKRESY_TOPO	FEATURE	1 POLYGON			0
2 KATERINA	H_TOPO	10	0,00005 (null)	KATERINA	H_KRAJE_TOPO	FEATURE	2 POLYGON			1
3 KATERINA	H_TOPO	10	0,00005 (null)	KATERINA	H_CR_TOPO	FEATURE	3 POLYGON			2

obr. 4.6 Nově vytvořená topologie H\_TOPO v pohledu USER\_SDO\_TOPO\_INFO

Vytvořily se zároveň tabulky topologických primitiv: H\_TOPO\_EDGES\$ (tabulka hran), H\_TOPO\_FACES\$ (tabulka stěn), H\_TOPO\_NODES\$ (tabulka uzlů) a



H\_TOPO\_HISTORY\$. Zároveň se vytvořily také indexy H\_TOPO\_PKEID\$ a H\_TOPOHIS\_PK (viz obr. 4.7).



obr. 4.7 Nově vytvořené tabulky H\_TOPO\_EDGES\$, H\_TOPO\_FACE\$, H\_TOPO\_NODES\$, H\_TOPO\_HISTORY\$ a indexy H\_TOPO\_PKEID\$, H\_TOPOHIS\_PK

#### 4.2.2 Vytvoření „universe face“ F0

„Universe face“, neboli také „face zero“ F0, je základní stěna, která se vytváří pro každou topologii. Tato plocha *obsahuje* vše, co se nalézá v topologii, a nejsou s ní asociovány žádné geometrie. Hodnota „face ID“ je rovna -1 (viz obr. 4.8).

```
INSERT INTO H_TOPO_FACE$ values (
    -1,
    NULL,
    SDO_LIST_TYPE(),
    SDO_LIST_TYPE(),
    NULL
);
COMMIT;
```

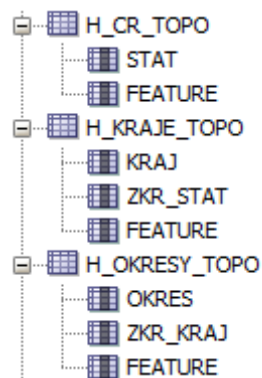
FACE_ID	BOUNDARY_EDGE_ID	ISLAND_EDGE_ID_LIST	ISLAND_NODE_ID_LIST	MBR_GEOMETRY
1	-1	(null)	MDSYS.SDO_LIST_TYPE(-1446)	MDSYS.SDO_LIST_TYPE() (null)

obr. 4.8 Vložená „face zero“ F0 v tabulce H\_TOPO\_FACE\$

### 4.2.3 Vytvoření prvkových tabulek

Pro každou vrstvu topologických prvků je nutné vytvořit prvkovou tabulku (*Feature Table*). Topologickým prvkem (*feature*) je myšlena prostorová reprezentace reálného objektu (např. Jihočeský kraj). Takový prvek je uložen jako sada topologických primitiv a má unikátní ID (přiřazuje Spatial). Vrstva topologických prvků (*topology geometry layer*) sestává z množiny topologických prvků (zde jsou takovými vrstvami ČR, kraje a okresy). Zde se jedná o prvkové tabulky H\_CR\_TOPO, H\_KRAJE\_TOPO a H\_OKRESY\_TOPO, v nichž budou uložena data z tabulek CR, KRAJE a OKRESY v topologickém formátu, tzn. s použitím datového typu SDO\_TOPO\_GEOMETRY (viz obr. 4.9).

```
CREATE TABLE H_OKRESY_TOPO (  
    OKRES VARCHAR2(30) PRIMARY KEY,  
    ZKR_KRAJ VARCHAR2(32),  
    feature SDO_TOPO_GEOMETRY  
);  
  
CREATE TABLE H_KRAJE_TOPO (  
    KRAJ VARCHAR2(30) PRIMARY KEY,  
    ZKR_STAT VARCHAR2(32),  
    feature SDO_TOPO_GEOMETRY  
);  
  
CREATE TABLE H_CR_TOPO (  
    STAT VARCHAR2(32) PRIMARY KEY,  
    feature SDO_TOPO_GEOMETRY  
);
```



obr. 4.9 Vytvořené prvkové tabulky H\_CR\_TOPO, H\_KRAJE\_TOPO a H\_OKRESY\_TOPO

#### 4.2.4 Asociace prvkových tabulek s topologií

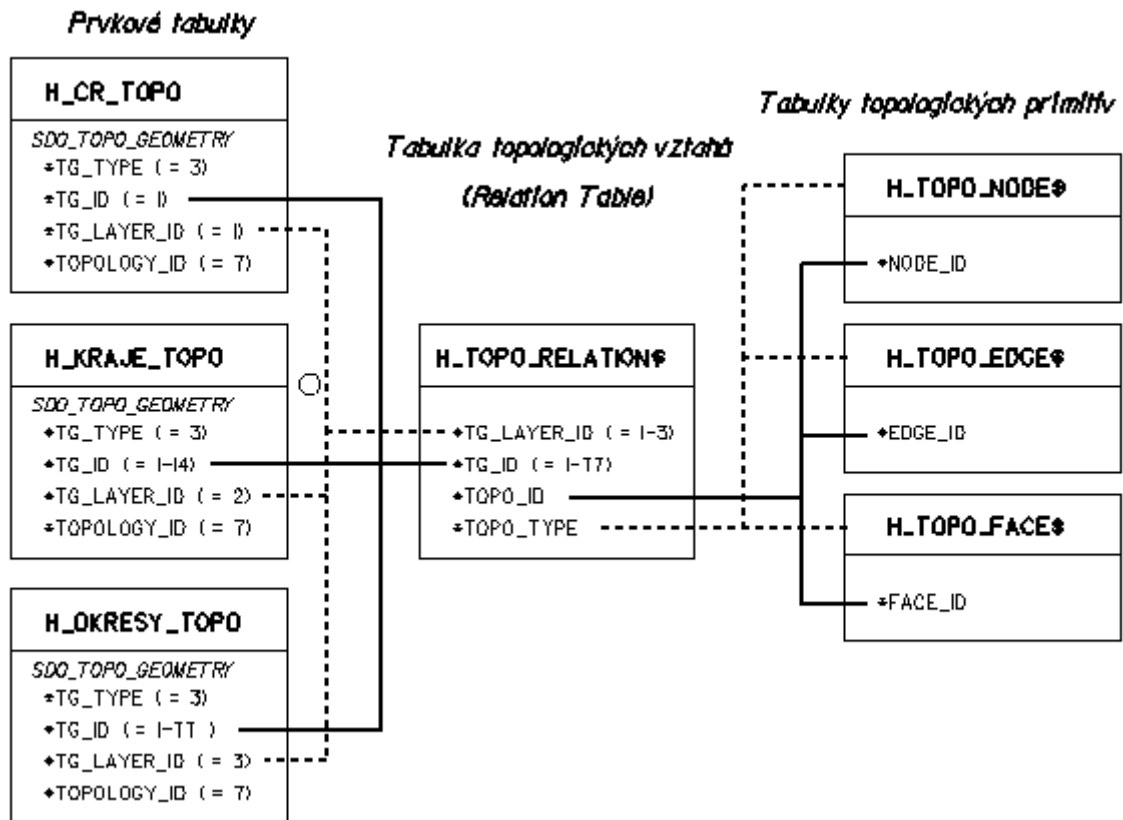
Každou prvkovou tabulku (H\_CR\_TOPO, H\_KRAJE\_TOPO a H\_OKRESY\_TOPO) je nyní nutné asociovat s vytvořenou topologií H\_TOPO. Využívá se k tomu procedury SDO\_TOPO.ADD\_TOPO\_GEOMETRY\_LAYER. Začíná se na nejnižší úrovni v hierarchii - s vrstvou potomka (zde *okresy*). U této vrstvy se neuvádí parametr *child\_layer\_id*, neboť je vrstva uložena na nejnižší (nultou) úroveň v hierarchii. Ve vyšších vrstvách je nutné tento parametr specifikovat. Blíže se asociací prvkových tabulek s topologií zabývá kapitola 2.2.

```
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('H_TOPO', 'H_OKRESY_TOPO',  
    'FEATURE', 'POLYGON');
```

```
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('H_TOPO', 'H_KRAJE_TOPO', 'FEATURE',  
    'POLYGON', NULL, 1);
```

```
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('H_TOPO', 'H_CR_TOPO', 'FEATURE',  
    'POLYGON', NULL, 2);
```

Zároveň se vytvořila tabulka H\_TOPO\_RELATIONS\$. Jedná se o tabulku, která umožní propojení mezi prvkovou tabulkou (H\_CR\_TOPO, H\_KRAJE\_TOPO nebo H\_OKRESY\_TOPO) a tabulkami s uloženými topologickými primitivami (H\_TOPO\_EDGES\$, H\_TOPO\_FACES\$, H\_TOPO\_NODES\$). Princip tohoto propojení je zobrazen na obr. 4.10.



obr. 4.10 Princip Relationship Information table(tabulky topologických vztahů) - propojení mezi prvkovou tabulkou a tabulkami topologických primitiv

Přehled asociovaných prvkových tabulek s topologií je zobrazen v pohledu USER\_SDO\_TOPO\_METADATA (viz obr. 4.11).

OWNER	TOPO...	T...	TOLERANCE	SRID	TABL...	TABLE_NAME	COL...	TG...	TG_LA...	TG_L...	CHIL...	N
KATERINA	H_TOPO	23	0,00005	(null)	KATERINA	H_OKRESY_TOPO	FEATURE	1	POLYGON	0	(null)	H_TC
KATERINA	H_TOPO	23	0,00005	(null)	KATERINA	H_KRAJE_TOPO	FEATURE	2	POLYGON	1		1 H_TC
KATERINA	H_TOPO	23	0,00005	(null)	KATERINA	H_CR_TOPO	FEATURE	3	POLYGON	2		2 H_TC

obr. 4.11 Přehled asociovaných tabulek v pohledu USER\_SDO\_TOPO\_METADATA

## 4.2.5 Vytvoření TopoMap objektu a načtení celé topologie do mezipaměti

V dalším kroku je vytvářen TopoMap objekt H\_TOPO\_TOPOMAP, a to pomocí procedury SDO\_TOPO\_MAP.CREATE\_TOPO\_MAP. Vytvoří se tak mezipaměť pro editaci objektů asociovaných se specifikovanou topologií, tj. topologií H\_TOPO. Následně je celá topologie načítána (případně může být načítán jen pravoúhelníkový výřez) do právě vytvořené

TopoMap mezipaměti. Toto načtení umožňuje editaci objektů. O takto načtené topologii je možné se přesvědčit v tabulce SDO\_TOPO\_MAPS (viz obr. 4.12). Tuto tabulku vytváří a spravuje Oracle Spatial.

```
EXECUTE SDO_TOPO_MAP.CREATE_TOPO_MAP('H_TOPO', 'H_TOPOMAP');
EXECUTE SDO_TOPO_MAP.LOAD_TOPO_MAP('H_TOPOMAP', 'true');
```

TOPOLOGY_ID
1 H_TOPO

obr. 4.12 Načtená topologie H\_TOPO do mezipaměti TopoMap

#### 4.2.6 Naplnění prvkových tabulek

Dalším krokem je naplnění prvkových tabulek daty z prostorových tabulek (CR, KRAJE, OKRESY) s využitím funkce SDO\_TOPO\_MAP.CREATE\_FEATURE. Pomocí této funkce jsem naplnila tabulku H\_OKRESY\_TOPO, tedy tabulku na nejnižší úrovni hierarchie (viz obr. 4.13).

```
BEGIN
  FOR ok_rec IN (SELECT nazev, ogc_geometry, NK FROM OKRESY) LOOP
    INSERT INTO H_OKRESY_TOPO VALUES (
      ok_rec.nazev,
      ok_rec.NK,
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_OKRESY_TOPO',
        'FEATURE', ok_rec.ogc_geometry)
    );
  END LOOP;
END;
/
```

	OKRES	ZKR_KRAJ	FEATURE
1	Plzeň-město	PL	MDSYS.SDO_TOPO_GEOMETRY(3,1,1,23)
2	Žďár nad Sázavou	VY	MDSYS.SDO_TOPO_GEOMETRY(3,2,1,23)
3	Chomutov	US	MDSYS.SDO_TOPO_GEOMETRY(3,3,1,23)
4	Olomouc	OL	MDSYS.SDO_TOPO_GEOMETRY(3,4,1,23)
5	Nymburk	ST	MDSYS.SDO_TOPO_GEOMETRY(3,5,1,23)
6	Semily	LB	MDSYS.SDO_TOPO_GEOMETRY(3,6,1,23)

obr. 4.13 Naplněná prvková tabulka H\_OKRESY\_TOPO (výřez)

Pro naplnění tabulek na vyšších úrovních hierarchie je nutné přidat podmínku *dml\_condition*. Tato podmínka je vytvářena pro kraje na základě sloupce *ZKR\_KRAJ* z tabulky H\_OKRESY\_TOPO (sloupec se zkratkou kraje, ke kterému okres náleží), a pro republiku na základě sloupce *ZKR\_STAT* z tabulky H\_KRAJE\_TOPO (sloupec se zkratkou státu, tedy České Republiky, ke které kraje náleží). Blíže se funkcí SDO\_TOPO\_MAP.CREATE\_FEATURE, používanou v topologii s hierarchickou strukturou, zabývá kapitola 2.2.

Postupně tak byla naplněna tabulka H\_KRAJE\_TOPO všemi čtrnácti kraji (viz obr. 4.14):

#### ***Královéhradecký kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'KH') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'KH'||')
    );
  END LOOP;
END;
/
```

#### ***Jihomoravský kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'JM') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'JM'||')
    );
  END LOOP;
END;
```

/

### ***Olomoucký kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'OL') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'OL'||')
    );
  END LOOP;
END;
/
```

### ***Karlovarský kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'KA') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'KA'||')
    );
  END LOOP;
END;
/
```

### ***Zlínský kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'ZL') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'ZL'||')
    );
  END LOOP;
END;
/
```

### ***Středočeský kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'ST') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'ST'||')
    );
  END LOOP;
END;
/
```

### ***Pardubický kraj:***

```
BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'PA') LOOP
```

```

INSERT INTO H_KRAJE_TOPO VALUES (
    kr_rec.nazev,
    'CR',
    SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||PA||')
);
END LOOP;
END;
/

```

### ***Liberecký kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'LB') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||LB||')
    );
END LOOP;
END;
/

```

### ***Moravskoslezský kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'MS') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||MS||')
    );
END LOOP;
END;
/

```

### ***Kraj Hlavní město Praha:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'HP') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||HP||')
    );
END LOOP;
END;
/

```

### ***Jihočeský kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'JC') LOOP
    INSERT INTO H_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||JC||')
    );

```



```
END LOOP;  
END;  
/
```

### ***Plzeňský kraj:***

```
BEGIN  
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'PL') LOOP  
    INSERT INTO H_KRAJE_TOPO VALUES (  
      kr_rec.nazev,  
      'CR',  
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',  
        'FEATURE', 'ZKR_KRAJ='||'PL'||')  
    );  
  END LOOP;  
END;  
/
```

### ***Ústecký kraj:***

```
BEGIN  
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'US') LOOP  
    INSERT INTO H_KRAJE_TOPO VALUES (  
      kr_rec.nazev,  
      'CR',  
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',  
        'FEATURE', 'ZKR_KRAJ='||'US'||')  
    );  
  END LOOP;  
END;  
/
```

### ***Kraj Vysočina:***

```
BEGIN  
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'VY') LOOP  
    INSERT INTO H_KRAJE_TOPO VALUES (  
      kr_rec.nazev,  
      'CR',  
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_KRAJE_TOPO',  
        'FEATURE', 'ZKR_KRAJ='||'VY'||')  
    );  
  END LOOP;  
END;  
/
```

	KRAJ	ZKR_STAT	FEATURE
1	Královéhradecký	CR	MDSYS.SDO_TOPO_GEOMETRY(3,1,2,23)
2	Jihomoravský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,2,2,23)
3	Olomoucký	CR	MDSYS.SDO_TOPO_GEOMETRY(3,3,2,23)
4	Karlovarský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,4,2,23)
5	Zlínský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,5,2,23)
6	Středočeský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,6,2,23)
7	Pardubický	CR	MDSYS.SDO_TOPO_GEOMETRY(3,7,2,23)
8	Liberecký	CR	MDSYS.SDO_TOPO_GEOMETRY(3,8,2,23)
9	Moravskoslezský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,9,2,23)
10	Hlavní město Praha	CR	MDSYS.SDO_TOPO_GEOMETRY(3,10,2,23)
11	Jihočeský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,11,2,23)
12	Plzeňský	CR	MDSYS.SDO_TOPO_GEOMETRY(3,12,2,23)
13	Ústecký	CR	MDSYS.SDO_TOPO_GEOMETRY(3,13,2,23)
14	Vysočina	CR	MDSYS.SDO_TOPO_GEOMETRY(3,14,2,23)

obr. 4.14 Naplněná prvková tabulka H\_KRAJE\_TOPO

Jako poslední byla naplněna tabulka na nejvyšší úrovni v hierarchii - tabulka H\_CR\_TOPO (viz obr. 4.15):

```

BEGIN
  FOR cr_rec IN (SELECT nazev FROM CR) LOOP
    INSERT INTO H_CR_TOPO VALUES (
      cr_rec.nazev,
      SDO_TOPO_MAP.CREATE_FEATURE('H_TOPO', 'H_CR_TOPO',
        'FEATURE', 'ZKR_STAT=''CR''')
    );
  END LOOP;
END;
/

```

	STAT	FEATURE
1	Česká republika	MDSYS.SDO_TOPO_GEOMETRY(3,1,3,23)

obr. 4.15 Naplněná prvková tabulka H\_CR\_TOPO

V tomto kroku se zároveň naplnily daty tabulky H\_TOPO\_EDGES\$, H\_TOPO\_FACES\$, H\_TOPO\_NODES\$, H\_TOPO\_HISTORY\$ a H\_TOPO\_RELATIONS\$. Také se naplnily pohledy H\_TOPO\_RELATIONL a H\_TOPO\_RELATLIDS.

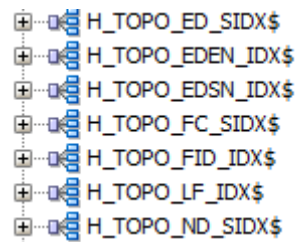
Nakonec byly potvrzeny změny v mezipaměti a smazán TopoMap objekt:

```
CALL SDO_TOPO_MAP.COMMIT_TOPO_MAP();  
CALL SDO_TOPO_MAP.DROP_TOPO_MAP('H_TOPOMAP');
```

#### 4.2.7 Inicializace topologických metadat

Na závěr byla provedena inicializace topologických metadat pomocí procedury SDO\_TOPO.INITIALIZE\_METADATA. Tato procedura zároveň vytváří prostorové indexy na tabulkách H\_TOPO\_EDGE\$ (index H\_TOPO\_ED\_SIDX\$), H\_TOPO\_FACE\$ (index H\_TOPO\_FC\_SIDX\$) a H\_TOPO\_NODE\$ (index H\_TOPO\_ND\_SIDX\$), a také přídatné B-tree indexy na tabulkách H\_TOPO\_EDGE\$ a H\_TOPO\_NODE\$ (viz obr. 4.16).

```
EXECUTE SDO_TOPO.INITIALIZE_METADATA('H_TOPO');
```



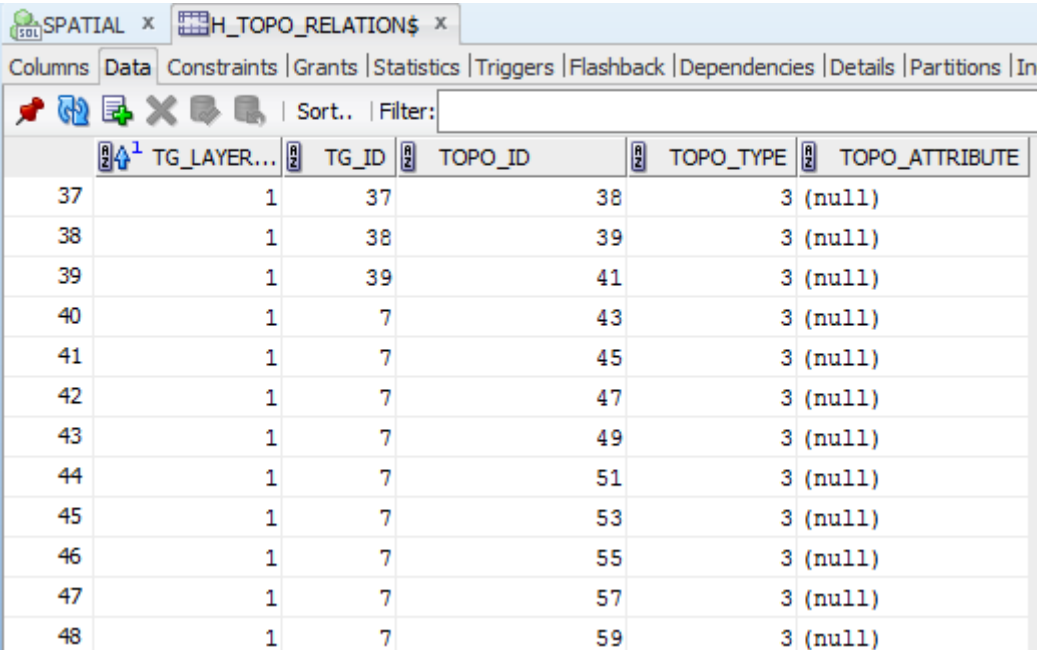
obr. 4.16 Nově vytvořené indexy na tabulkách H\_TOPO\_EDGE\$, H\_TOPO\_FACE\$ a H\_TOPO\_NODE\$

## 5 Nesrovnalosti ve vytvořené topologii H\_TOPO

V tuto chvíli mám vytvořenou topologii H\_TOPO s hierarchickou strukturou. Objevily se však některé nesrovnalosti, tato topologie není úplně čistá. V této kapitole bych se chtěla zaměřit na vyhledání těchto chyb, jejich původ a také možnou nápravu.

### 5.1 Vyhledání chyb a jejich původ

Ve vytvořené topologii H\_TOPO s hierarchickou strukturou jsou nyní vytvořeny tři nové tabulky topologických primitiv – H\_TOPO\_NODES\$, H\_TOPO\_EDGES\$ a H\_TOPO\_FACES\$. Z tabulky H\_TOPO\_RELATIONS\$, která tyto tabulky propojuje, je možné vidět některé nesrovnalosti. Na nejnižší úrovni hierarchie (ve vrstvě, jejíž TG\_LAYER\_ID = 1) by měl být každý prvek (čili okres) tvořen jednou stěnou. Jednomu TG\_ID by tak mělo odpovídat jedno TOPO\_ID. Jak je ale zřejmé z obr. 5.1 Okres Frýdek-Místek (TG\_ID = 7) je tvořen 19ti stěnami (výřez), některé okresy jsou tvořeny více stěnami, např. okres Frýdek-Místek je tvořen devatenácti stěnami. Celkem se tak jedná o 11 problematických okresů: Frýdek-Místek, Karviná, Opava, Nový Jičín, Ostrava – město, Praha – západ, Praha – východ, Hlavní město Praha, Brno – venkov, Brno – město a Blansko.



TG_LAYER_ID	TG_ID	TOPO_ID	TOPO_TYPE	TOPO_ATTRIBUTE
37	1	37	3	(null)
38	1	38	3	(null)
39	1	39	3	(null)
40	1	7	3	(null)
41	1	7	3	(null)
42	1	7	3	(null)
43	1	7	3	(null)
44	1	7	3	(null)
45	1	7	3	(null)
46	1	7	3	(null)
47	1	7	3	(null)
48	1	7	3	(null)

obr. 5.1 Okres Frýdek-Místek (TG\_ID = 7) je tvořen 19ti stěnami (výřez)

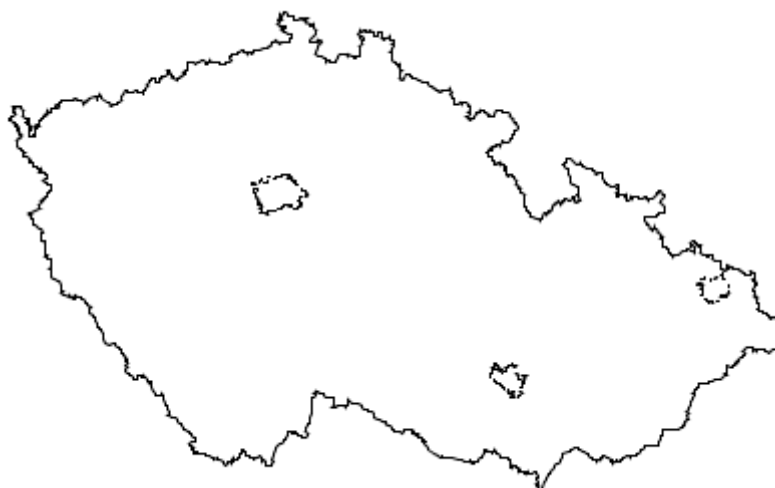
Dále je z tabulky H\_TOPO\_RELATIONS\$ zřejmé, že jedna stěna (jedno TOPO\_ID) je společná pro více prvků (více TG\_ID). Na obr. 5.2 je vidět, že např. okresy Nový Jičín a Ostrava – město jsou tvořeny stejnou stěnou, jejíž TOPO\_ID = 630. Konkrétně tyto dva okresy tak mají 13 společných stěn. Takto problematické jsou opět okresy, které byly vyjmenované v předchozím odstavci. Okres Ostrava – město má společné stěny spolu s okresy Frýdek-Místek, Karviná, Opava a Nový Jičín. Okres Hlavní město Praha tvoří některé stěny, které tvoří zároveň i okresy Praha – západ a Praha – východ. Okres Brno – město tvoří některé stěny, které tvoří zároveň i okresy Brno – venkov a Blansko. Jde tedy vždy o sousední okresy.

TG_LAYER_ID	TG_ID	TOPO_ID	TOPO_TYPE	TOPO_ATTRIBUTE
721	1	630	3	(null)
722	1	630	3	(null)
723	1	632	3	(null)
724	1	632	3	(null)
725	1	634	3	(null)
726	1	634	3	(null)
727	1	636	3	(null)
728	1	636	3	(null)

obr. 5.2 Jedna stěna (jedno TOPO\_ID) je společná pro více okresů (jedno TG\_ID) (výřez)

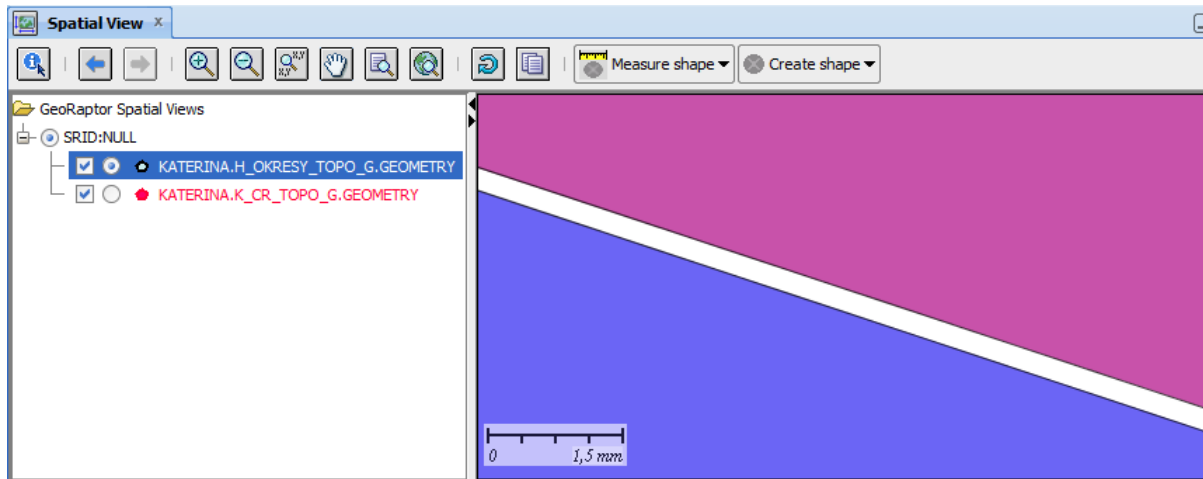
Také množství topologických primitiv v tabulkách H\_TOPO\_FACES\$ a H\_TOPO\_EDGES\$ bylo „podezřelé“. Tabulka H\_TOPO\_FACES\$ obsahuje 655 stěn. Domnívám se, že správný počet by měl být 78 (77 okresů + F0). Tabulka H\_TOPO\_EDGES\$ obsahuje 1447 hran.

Po vizualizaci České republiky (postup bude uveden v kapitole 6.2 v sedmé úloze) se nezobrazila pouze její hranice, ale také množství drobných čar kolem tří okresů – Hlavního města Prahy, Brna – města a Ostravy – města (viz obr. 5.3).



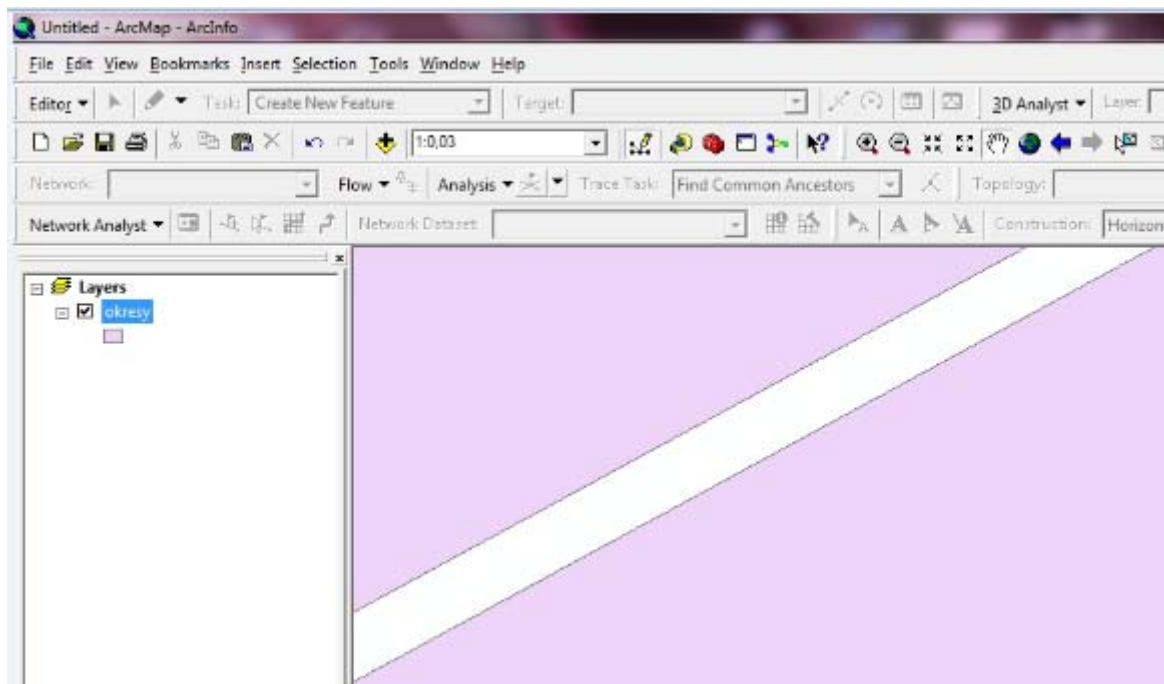
obr. 5.3 Vizualizace ČR - topologie H\_TOPO

Při dostatečném přiblížení těchto drobných čar a při současné vizualizaci okresů a jejich probarvení bylo vidět, že mezi okresy je další stěna, která nepatří k žádnému okresu. Drobné čáry tak nejsou čáry, ale hodně úzké polygony (obr. 5.4). Šířky těchto polygonů jsou v řádech maximálně milimetrů.



obr. 5.4 Mezera mezi okresy

Hranice okresů tudíž na sobě úplně nesedí. Byla zkontrolována vstupní data – soubor okresy.shp. Jak je vidět na obr. 5.5, nepřesnosti jsou již ve vstupních datech, prohlédnutých prostřednictvím ArcMapu. Okresy Hlavní město Praha, Brno – město a Ostrava – město se tak překrývají či nedokrývají s okresy sousedními.



obr. 5.5 Mezera mezi okresy v původních datech, zobrazená v ArcMapu

## 5.2 Návrh řešení

Rozhodla jsem se tento problém vyřešit vytvořením nové topologie F\_TOPO, ve které bude pouze 74 okresů, nebudou tak naimportovány tři výše uvedené problematické okresy. Poté tyto tři okresy vytvořím z hranic okresů, které tyto okresy obklopují.

V prvním kroku byla vytvořena nová tabulka OKRESY\_74, obsahující 74 „bezproblémových“ okresů:

```
CREATE TABLE okresy_74 AS SELECT * FROM OKRESY;
DELETE FROM okresy_74 WHERE nazev = 'Brno-město';
DELETE FROM okresy_74 WHERE nazev = 'Hlavní město Praha';
DELETE FROM okresy_74 WHERE nazev = 'Ostrava-město';
```

Následně byla vytvořena nová topologie F\_TOPO:

```
EXECUTE SDO_TOPO.CREATE_TOPOLOGY('F_TOPO', 0.01);

INSERT INTO F_TOPO_FACE$ values (
    -1,
    NULL,
    SDO_LIST_TYPE(),
```

```

        SDO_LIST_TYPE(),
        NULL
    );
COMMIT;

```

Poté byly vytvořeny prvkové tabulky F\_OKRESY\_TOPO, F\_KRAJE\_TOPO a F\_CR\_TOPO, v nichž budou uloženy data z tabulek OKRESY, KRAJE a CR v topologickém formátu (SDO\_TOPO\_GEOMETRY). Tyto prvkové tabulky byly asociovány s vytvořenou topologií F\_TOPO.

```

CREATE TABLE F_OKRESY_TOPO (
    OKRES VARCHAR2(30) PRIMARY KEY,
    ZKR_KRAJ VARCHAR2(32),
    feature SDO_TOPO_GEOMETRY
);

```

```

CREATE TABLE F_KRAJE_TOPO (
    KRAJ VARCHAR2(30) PRIMARY KEY,
    ZKR_STAT VARCHAR2(32),
    feature SDO_TOPO_GEOMETRY
);

```

```

CREATE TABLE F_CR_TOPO (
    STAT VARCHAR2(32) PRIMARY KEY,
    feature SDO_TOPO_GEOMETRY
);

```

```

EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('F_TOPO', 'F_OKRESY_TOPO',
    'FEATURE', 'POLYGON');

```

```

EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('F_TOPO', 'F_KRAJE_TOPO',
    'FEATURE', 'POLYGON', NULL, 1);

```

```

EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('F_TOPO', 'F_CR_TOPO', 'FEATURE',
    'POLYGON', NULL, 2);

```

V dalším kroku byl vytvořen TopoMap objekt F\_TOPOMAP, a následně byla celá topologie do této mezipaměti načtena. Poté již byla načtena samotná data do topologického formátu do tabulek F\_OKRESY\_TOPO a F\_KRAJE\_TOPO. Nebyly načteny kraje, do kterých patří problémové okresy (Hlavní město Praha, Jihomoravský a Moravskoslezský kraj). Také nebyla načtena Česká Republika, neboť nebyly naplněny všechny kraje, kterými je republika tvořena.

```

EXECUTE SDO_TOPO_MAP.CREATE_TOPO_MAP('F_TOPO', 'F_TOPOMAP');
EXECUTE SDO_TOPO_MAP.LOAD_TOPO_MAP('F_TOPOMAP', 'true');

```

### **Okresy:**

```

BEGIN

```



```

FOR ok_rec IN (SELECT nazev, ogc_geometry, NK FROM OKRESY_74) LOOP
    INSERT INTO F_OKRESY_TOPO VALUES (
        ok_rec.nazev,
        ok_rec.NK,
        SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_OKRESY_TOPO',
            'FEATURE', ok_rec.ogc_geometry)
    );
END LOOP;
END;
/

```

### ***Královéhradecký kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'KH') LOOP
    INSERT INTO F_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ = 'KH''')
    );
END LOOP;
END;
/

```

### ***Olomoucký kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'OL') LOOP
    INSERT INTO F_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='OL''')
    );
END LOOP;
END;
/

```

### ***Karlovarský kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'KA') LOOP
    INSERT INTO F_KRAJE_TOPO VALUES (
        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='KA''')
    );
END LOOP;
END;
/

```

### ***Zlínský kraj:***

```

BEGIN
FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'ZL') LOOP
    INSERT INTO F_KRAJE_TOPO VALUES (

```

```

        kr_rec.nazev,
        'CR',
        SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ=' 'ZL' '')
    );
END LOOP;
END;
/

```

### ***Středočeský kraj:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'ST') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',
            SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ=' 'ST' '')
        );
    END LOOP;
END;
/

```

### ***Pardubický kraj:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'PA') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',
            SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ=' 'PA' '')
        );
    END LOOP;
END;
/

```

### ***Liberecký kraj:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'LB') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',
            SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ=' 'LB' '')
        );
    END LOOP;
END;
/

```

### ***Jihočeský kraj:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'JC') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',

```

```

                SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
                'FEATURE', 'ZKR_KRAJ='||JC||')
            );
        END LOOP;
    END;
/

```

### ***Plzeňský kraj:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'PL') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',
            SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||PL||')
        );
    END LOOP;
END;
/

```

### ***Ústecký kraj:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'US') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',
            SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||US||')
        );
    END LOOP;
END;
/

```

### ***Kraj Vysočina:***

```

BEGIN
    FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'VY') LOOP
        INSERT INTO F_KRAJE_TOPO VALUES (
            kr_rec.nazev,
            'CR',
            SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
            'FEATURE', 'ZKR_KRAJ='||VY||')
        );
    END LOOP;
END;
/

```

V nově vytvořené topologii F\_TOPO je celkem 78 stěn, které odpovídají 74 okresům, třem „dírák“ mezi těmito okresy a základní stěně. Byly vybrány FACE\_ID všech „obsazených“ stěn, čili stěn, které jsou součástí nějakého prvku (feature, čili nějakého okresu). Byly proto vybrány TOPO\_ID prvků na nejnižší úrovni v hierarchii (kde

TG\_LAYER\_ID = 1) z tabulky F\_TOPO\_RELATION\$, které odpovídají FACE\_ID obsazených stěn:

```
SELECT topo_id FROM f_topo_relation$ WHERE tg_layer_id = 1;
```

Následně byly vybrány FACE\_ID všech „neobsazených“ stěn (tj. děr):

```
SELECT f.face_id AS neobsazena_stena FROM f_topo_face$ f
WHERE f.face_id NOT IN (
    SELECT topo_id FROM f_topo_relation$ WHERE tg_layer_id = 1);
```

```
NEOBSAZENA_STENA
-----
-1
59
14
68
```

obr. 5.6 FACE\_ID neobsazených stěn

Tyto neobsazené stěny (viz obr. 5.6), tedy kromě stěny základní, jsou tudíž součástí děr, které suplují nenaimportované okresy. Z těchto neobsazených stěn budou nyní nenaimportované okresy vytvořeny. V tuto chvíli vím, že neobsazené stěny mají FACE\_ID = 14, 59 a 68, nicméně nevím, kterému chybějícímu okresu která z těchto stěn náleží. Bylo nutné těmto FACE\_ID přiřadit konkrétní okres, který není v topologii naimportován. To bylo zjištěno pomocí FACE\_ID stěn, které s těmito „dírami“ sousedí. Nejprve byly vybrány ohraničující hrany (viz obr. 5.7):

```
SELECT boundary_edge_id AS ohranicujici_hrana FROM f_topo_face$ WHERE
face_id in (
    SELECT f.face_id FROM f_topo_face$ f WHERE f.face_id not in (
        SELECT topo_id FROM f_topo_relation$ WHERE tg_layer_id = 1
    )
);
```

```
OHRANICUJICI_HRANA
-----
23
-279
-266
```

obr. 5.7 EDGE\_ID ohraničujících hran

Poté byly vybrány stěny, které sousedí s těmito hranami (viz obr. 5.8), a zjištěno TG\_ID prvků (okresů), které těmto stěnám odpovídají (viz obr. 5.9).

```
SELECT left_face_id, right_face_id FROM f_topo_edge$ WHERE edge_id in (
    SELECT ABS (boundary_edge_id) FROM f_topo_face$ WHERE face_id in (
```

```

SELECT f.face_id FROM f_topo_face$ f WHERE f.face_id not in (
    SELECT topo_id FROM f_topo_relation$ WHERE
        tg_layer_id = 1)
);

```

LEFT_FACE_ID	RIGHT_FACE_ID
14	15
72	68
75	59

**obr. 5.8 Výčet stěn, které sousedí s hranami, jejichž EDGE\_ID = 23, 279, 266**

Se stěnou s FACE\_ID = 14 sousedí stěna s FACE\_ID = 15.

Se stěnou s FACE\_ID = 68 sousedí stěna s FACE\_ID = 72.

Se stěnou s FACE\_ID = 59 sousedí stěna s FACE\_ID = 75.

```

SELECT topo_id AS sousedici_stena, tg_id AS tg_id_okresu
FROM f_topo_relation$ WHERE topo_id in (15,72,75);

```

SOUSEDICI_STENA	TG_ID_OKRESU
15	14
72	65
75	69

**obr. 5.9 TG\_ID okresů, která odpovídají sousedícím stěnám**

Stěně s FACE\_ID = 15, která sousedí se stěnou s FACE\_ID = 14, odpovídá okres s TG\_ID = 14.

Stěně s FACE\_ID = 72, která sousedí se stěnou s FACE\_ID = 68, odpovídá okres s TG\_ID = 65.

Stěně s FACE\_ID = 75, která sousedí se stěnou s FACE\_ID = 59, odpovídá okres s TG\_ID = 69.

Nyní musím zjistit, kterým okresům tato TG\_ID = 14, 65, 69 patří (viz obr. 5.10).

```

SELECT o.feature.tg_id AS tg_id_soused_okresu, okres FROM f_okresy_topo o
WHERE o.feature.tg_id in (
    SELECT tg_id FROM f_topo_relation$ WHERE topo_id in (15,72,75)
);

```

TG_ID_SOUSED_OKRESU	OKRES
14	Brno-venkov
65	Praha-západ
69	Nový Jičín

obr. 5.10 Okresy, které sousedí s hledanými okresy

Se stěnou s FACE\_ID = 14 sousedí okres Brno-venkov, tj. nenaimportovaný okres je Brno-město. Se stěnou s FACE\_ID = 68 sousedí Praha-východ, tj. nenaimportovaný okres je Hlavní město Praha. Se stěnou s FACE\_ID = 59 sousedí Nový Jičín, tj. nenaimportovaný okres je Ostrava město.

Pro vytvoření okresů Hlavní město Praha, Brno – město a Ostrava – město bylo použito konstruktoru pro operace vložení, specifikující topologické prvky. Pomocí tohoto konstruktoru jsou vytvářeny nové topologické geometrické objekty. Existují dva tvary těchto konstruktorů, přičemž já jsem použila následující konstruktor:

```
SDO_TOPO_GEOMETRY (topology VARCHAR2,
                    tg_type NUMBER,
                    tg_layer_id NUMBER,
                    topo_ids SDO_TOPO_OBJECT_ARRAY)
```

Datový typ SDO\_TOPO\_OBJECT\_ARRAY je definován jako řetězec SDO\_TOPO\_OBJECT objektů. Datový typ SDO\_TOPO\_OBJECT má atributy (topo\_id NUMBER, topo\_type NUMBER). Hodnoty těchto atributů musí být v rámci hodnot uvedených v tabulce <topology-name>\_RELATION\$ [5].

Uvedený konstruktor byl použit třikrát, pro vytvoření okresů Hlavní město Praha, Brno – město a Ostrava – město.

```
INSERT INTO F_OKRESY_TOPO VALUES (
    'Brno-město',
    'JM',
    SDO_TOPO_GEOMETRY('F_TOPO', -- název topologie
                      3, -- topologický geometrický typ (polygon)
                      1, -- TG_LAYER_ID
                      SDO_TOPO_OBJECT_ARRAY (
                          SDO_TOPO_OBJECT (14, 3))) - FACE_ID = 14
);

INSERT INTO F_OKRESY_TOPO VALUES (
```

```

'Hlavní město Praha',
'HP',
SDO_TOPO_GEOMETRY('F_TOPO',
                    3,
                    1,
                    SDO_TOPO_OBJECT_ARRAY (
                      SDO_TOPO_OBJECT (68, 3)))
);

INSERT INTO F_OKRESY_TOPO VALUES (
  'Ostrava-město',
  'MS',
  SDO_TOPO_GEOMETRY('F_TOPO',
                    3,
                    1,
                    SDO_TOPO_OBJECT_ARRAY (
                      SDO_TOPO_OBJECT (59, 3)))
);

```

V tuto chvíli již nic nebrání načtení zbylých krajů do topologického formátu do tabulky F\_KRAJE\_TOPO. Následně byla načtena Česká Republika do tabulky F\_CR\_TOPO.

```

EXECUTE SDO_TOPO_MAP.CREATE_TOPO_MAP('F_TOPO', 'F_TOPOMAP');
EXECUTE SDO_TOPO_MAP.LOAD_TOPO_MAP('F_TOPOMAP', 'true');

```

### ***Jihomoravský kraj:***

```

BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'JM') LOOP
    INSERT INTO F_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'JM'||')
    );
  END LOOP;
END;
/

```

### ***Moravskoslezský kraj:***

```

BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'MS') LOOP
    INSERT INTO F_KRAJE_TOPO VALUES (
      kr_rec.nazev,
      'CR',
      SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||'MS'||')
    );
  END LOOP;
END;
/

```

### ***Kraj Hlavní město Praha:***

```

BEGIN
  FOR kr_rec IN (SELECT nazev FROM KRAJE WHERE NK = 'HP') LOOP

```

```

INSERT INTO F_KRAJE_TOPO VALUES (
    kr_rec.nazev,
    'CR',
    SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_KRAJE_TOPO',
        'FEATURE', 'ZKR_KRAJ='||HP||')
);
END LOOP;
END;
/

```

### **Česká Republika:**

```

BEGIN
FOR cr_rec IN (SELECT nazev FROM CR) LOOP
    INSERT INTO F_CR_TOPO VALUES (
        cr_rec.nazev,
        SDO_TOPO_MAP.CREATE_FEATURE('F_TOPO', 'F_CR_TOPO',
            'FEATURE', 'ZKR_STAT='||CR||')
    );
END LOOP;
END;
/

```

V tuto chvíli již jsou načtena do tabulek veškerá data a je vytvořena jejich hierarchická struktura, tj. kraje se skládají ze 77 okresů na nejnižší úrovni a Česká Republika se skládá ze 14 krajů o úroveň níže. Již zde nejsou žádné překryty či nedokryty a data jsou již topologicky čistá.

V závěru byl potvrzen (příkazem *commit*) a smazán TopoMap objekt a provedena inicializace metadat:

```

CALL SDO_TOPO_MAP.COMMIT_TOPO_MAP();
CALL SDO_TOPO_MAP.DROP_TOPO_MAP('F_TOPOMAP');
EXECUTE SDO_TOPO.INITIALIZE_METADATA('F_TOPO');

```

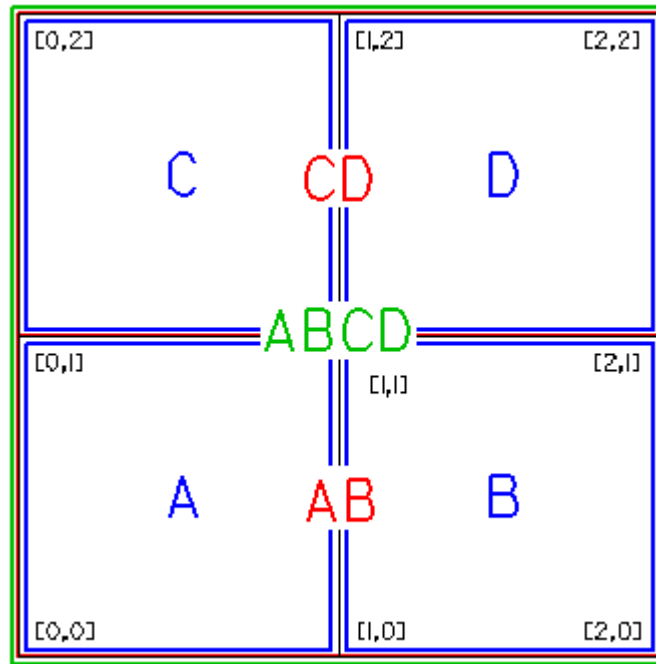


## 6 Ověření funkčnosti vytvořené hierarchie

Pro ověření funkčnosti vytvořené hierarchie v topologii F\_TOPO byla vytvořena jiná testovací topologie. V této topologii byla vytvořena obdobná hierarchie, pouze s jednoduchými tvary (pravoúhelníky – čtverce a obdélníky), na nichž bude možné jednoznačně prokázat požadovaný výsledek. Obě topologie budou testovány současně, aby bylo možné výsledky porovnat.

### 6.1 Vytvoření srovnávací topologie A\_TOPO

Pro ověření funkčnosti vytvořené hierarchie v topologii F\_TOPO byla nejdříve vytvořena srovnávací topologie A\_TOPO, taktéž s hierarchickou závislostí. Byly vytvořeny prostorové tabulky A, AB a ABCD. Tyto tabulky byly naplněny daty; tabulka A čtyřmi čtverečky o velikosti 1x1 (modrou barvou na obr. 6.1), tabulky AB dvěma obdélníčky 2x1 (každý zahrnující dva čtverečky z tabulky A, červenou barvou na obr. 6.1) a tabulka ABCD s jedním čtvercem o velikosti 2x2 (zahrnující všechny čtyři čtverečky z tabulky A, na obr. 6.1 zeleně). V těchto tabulkách je geometrie uložena pomocí datového typu SDO\_GEOMETRY. Tyto tabulky budou použity jako zdrojové tabulky při plnění topologických prvkových tabulek.



obr. 6.1 Topologie A\_TOPO

Vzhledem k tomu, že do topologie nesmí být vkládány tzv. „*optimized shapes*“, čili geometrické popisy zadané dvěma páry souřadnic protilehlých rohů (viz [5]), nemohl být použit pro vkládání těchto čtverečků tvar `SDO_ELEM_INFO_ARRAY(1,1003,3)` s vložením pouze dvou souřadnic pro levý dolní a pravý horní roh `SDO_ORDINATE_ARRAY(0,0,1,1)`, viz obr. 6.2. Byl proto použit tvar: `SDO_GEOMETRY(2003,NULL,NULL, SDO_ELEM_INFO_ARRAY(1,1003,1), SDO_ORDINATE_ARRAY(0,0,1,0,1,1,0,1,0,0))`, kdy byl čtvereček uložen čtyřmi rohy, resp. pěti body (viz obr. 6.3).

### Vytvoření a naplnění tabulky A:

```
CREATE TABLE A (
  NAZEV VARCHAR2(32) PRIMARY KEY,
  AB VARCHAR2(32),
  GEOMETRY SDO_GEOMETRY);

INSERT INTO A VALUES (
  'A',
  'AB',
  SDO_GEOMETRY(2003,NULL,NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1),
    SDO_ORDINATE_ARRAY(0,0,1,0,1,1,0,1,0,0)
  )
);

INSERT INTO A VALUES (
```

```

        'B',
        'AB',
        SDO_GEOMETRY(2003, NULL, NULL,
            SDO_ELEM_INFO_ARRAY(1, 1003, 1),
            SDO_ORDINATE_ARRAY(1, 0, 2, 0, 2, 1, 1, 1, 1, 0)
        )
    );

INSERT INTO A VALUES (
    'C',
    'CD',
    SDO_GEOMETRY(2003, NULL, NULL,
        SDO_ELEM_INFO_ARRAY(1, 1003, 1),
        SDO_ORDINATE_ARRAY(0, 1, 1, 1, 1, 2, 0, 2, 0, 1)
    )
);

INSERT INTO A VALUES (
    'D',
    'CD',
    SDO_GEOMETRY(2003, NULL, NULL,
        SDO_ELEM_INFO_ARRAY(1, 1003, 1),
        SDO_ORDINATE_ARRAY(1, 1, 2, 1, 2, 2, 1, 2, 1, 1)
    )
);

```

### ***Vytvoření a naplnění tabulky AB:***

```

CREATE TABLE AB (
    NAZEV VARCHAR2(32) PRIMARY KEY,
    ABCD VARCHAR2(32),
    GEOMETRY SDO_GEOMETRY);

INSERT INTO AB VALUES (
    'AB',
    'ABCD',
    SDO_GEOMETRY(2003, NULL, NULL,
        SDO_ELEM_INFO_ARRAY(1, 1003, 1),
        SDO_ORDINATE_ARRAY(0, 0, 2, 0, 2, 1, 0, 1, 0, 0)
    )
);

INSERT INTO AB VALUES (
    'CD',
    'ABCD',
    SDO_GEOMETRY(2003, NULL, NULL,
        SDO_ELEM_INFO_ARRAY(1, 1003, 1),
        SDO_ORDINATE_ARRAY(0, 1, 2, 1, 2, 2, 0, 2, 0, 1)
    )
);

```

### ***Vytvoření a naplnění tabulky ABCD:***

```

CREATE TABLE ABCD (
    NAZEV VARCHAR2(32) PRIMARY KEY,
    GEOMETRY SDO_GEOMETRY);

INSERT INTO ABCD VALUES (
    'ABCD',
    SDO_GEOMETRY(2003, NULL, NULL,

```

```

SDO_ELEM_INFO_ARRAY (1,1003,1),
SDO_ORDINATE_ARRAY (0,0,2,0,2,2,0,2,0,0)
)
);

```

	NAZEV	OGC_GEOMETRY
1	A	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),MDSYS.SDO_ORDINATE_ARRAY(0,0,1,1,0))
2	B	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),MDSYS.SDO_ORDINATE_ARRAY(1,0,2,1))
3	C	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),MDSYS.SDO_ORDINATE_ARRAY(0,1,1,2))
4	D	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,3),MDSYS.SDO_ORDINATE_ARRAY(1,1,2,2))

**obr. 6.2 Uložení čtverečků dvěma rohy, které není podporováno topologií**

	NAZEV	AB	GEOMETRY
1	A	AB	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(0,0,1,0,1,1,0,1,0,0))
2	B	AB	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(1,0,2,0,2,1,1,1,1,0))
3	C	CD	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(0,1,1,1,2,0,2,0,1))
4	D	CD	MDSYS.SDO_GEOMETRY(2003,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),MDSYS.SDO_ORDINATE_ARRAY(1,1,2,1,2,1,2,1,1))

**obr. 6.3 Uložení čtverečků čtyřmi rohy**

Dále byla aktualizována metadata o prostorových datech v pohledu USER\_SDO\_GEOM\_METADATA, provedena validace prostorových dat a vytvořeny prostorové indexy:

#### ***Aktualizace metadat:***

```

INSERT INTO user_sdo_geom_metadata (TABLE_NAME,COLUMN_NAME,DIMINFO,SRID)
VALUES (
    'A',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    ),
    NULL
);

```

```

INSERT INTO user_sdo_geom_metadata (TABLE_NAME,COLUMN_NAME,DIMINFO,SRID)
VALUES (
    'AB',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    ),
    NULL
);

```

```

INSERT INTO user_sdo_geom_metadata (TABLE_NAME,COLUMN_NAME,DIMINFO,SRID)
VALUES (
    'ABCD',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    )
);

```

```

),
NULL
);

```

### ***Validace prostorových dat:***

```

create table val_results (
    sdo_rowid ROWID,
    result VARCHAR2(2000)
);

```

```

call SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('A', 'GEOMETRY', 'VAL_RESULTS');
SELECT * FROM val_results;

```

```

DROP table val_results;

```

```

create table val_results (
    sdo_rowid ROWID,
    result VARCHAR2(2000)
);

```

```

call SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('AB', 'GEOMETRY', 'VAL_RESULTS');
SELECT * FROM val_results;

```

```

DROP table val_results;

```

```

create table val_results (
    sdo_rowid ROWID,
    result VARCHAR2(2000)
);

```

```

call SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT('ABCD', 'GEOMETRY', 'VAL_RESULTS');
SELECT * FROM val_results;

```

```

DROP table val_results;

```

### ***Vytvoření prostorových indexů:***

```

CREATE INDEX A_idx ON A (geometry)
    INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```

```

CREATE INDEX AB_idx ON AB (geometry)
    INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```

```

CREATE INDEX ABCD_idx ON ABCD (geometry)
    INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```

Dále byla vytvořena nová topologie A\_TOPO a do ní vložena základní stěna „face zero“ F0:

```

EXECUTE SDO_TOPO.CREATE_TOPOLOGY('A_TOPO', 0.00005);

```

```

INSERT INTO A_TOPO_FACE$ values (
    -1,
    NULL,
    SDO_LIST_TYPE(),
    SDO_LIST_TYPE(),
    NULL);

```

```
COMMIT;
```

Následně byly vytvořeny prvkové tabulky A\_A\_TOPO, AB\_A\_TOPO a ABCD\_A\_TOPO, v nichž budou uloženy data z tabulek A, AB a ABCD v topologickém formátu (SDO\_TOPO\_GEOMETRY).

```
CREATE TABLE A_A_TOPO (  
    NAZEV VARCHAR2(30) PRIMARY KEY,  
    AB VARCHAR2(32),  
    feature SDO_TOPO_GEOMETRY);
```

```
CREATE TABLE AB_A_TOPO (  
    NAZEV VARCHAR2(30) PRIMARY KEY,  
    ABCD VARCHAR2(32),  
    feature SDO_TOPO_GEOMETRY);
```

```
CREATE TABLE ABCD_A_TOPO (  
    NAZEV VARCHAR2(32) PRIMARY KEY,  
    feature SDO_TOPO_GEOMETRY);
```

Tyto prvkové tabulky (A\_A\_TOPO, AB\_A\_TOPO a ABCD\_A\_TOPO) byly asociovány s vytvořenou topologií A\_TOPO. Postupně od nejnižší úrovně v hierarchii (zde tabulka A\_A\_TOPO) byly asociovány vrstvy na vyšší úrovni (tabulka AB\_A\_TOPO), až po nejvyšší (tabulka ABCD\_A\_TOPO). Zároveň se vytvořila tabulka A\_TOPO\_RELATIONS\$, která umožní propojení mezi prvkovými tabulkami (A\_A\_TOPO, AB\_A\_TOPO nebo ABCD\_A\_TOPO) a tabulkami s uloženými topologickými primitivami (A\_TOPO\_EDGES\$, A\_TOPO\_FACES\$, A\_TOPO\_NODES\$).

```
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('A_TOPO', 'A_A_TOPO', 'FEATURE',  
    'POLYGON');
```

```
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('A_TOPO', 'AB_A_TOPO', 'FEATURE',  
    'POLYGON', NULL, 1);
```

```
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('A_TOPO', 'ABCD_A_TOPO', 'FEATURE',  
    'POLYGON', NULL, 2);
```

Přehled asociovaných prvkových tabulek s topologií je zobrazen v pohledu USER\_SDO\_TOPO\_METADATA (viz obr. 6.4).

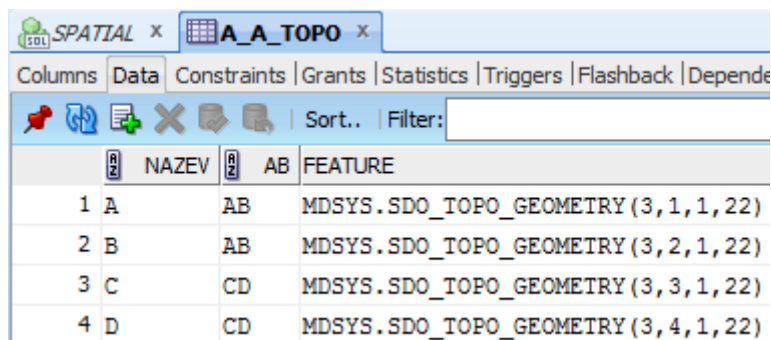
	OWNER	TOPO...	T...	TOLERANCE	SRID	TABL...	TABLE_NAME	COL...	TG...	TG_LA...	TG_L...	CHIL...	NODE_SEQUENC
1	KATERINA	A_TOPO	22	0,00005	(null)	KATERINA	A_A_TOPO	FEATURE	1	POLYGON	0	(null)	A_TOPO_NODE_S
2	KATERINA	A_TOPO	22	0,00005	(null)	KATERINA	AB_A_TOPO	FEATURE	2	POLYGON	1		1 A_TOPO_NODE_S
3	KATERINA	A_TOPO	22	0,00005	(null)	KATERINA	ABCD_A_TOPO	FEATURE	3	POLYGON	2		2 A_TOPO_NODE_S

obr. 6.4 Přehled asociovaných tabulek v pohledu USER\_SDO\_TOPO\_METADATA

V dalším kroku byl vytvořen TopoMap objekt A\_TOPOMAP, a následně byla celá topologie do této mezipaměti načtena. Poté již byla načtena samotná data do topologického formátu do tabulek A\_A\_TOPO, AB\_A\_TOPO a ABCD\_A\_TOPO (viz obr. 6.5, obr. 6.6 a obr. 6.7). V závěru byla provedena inicializace metadat.

```
EXECUTE SDO_TOPO_MAP.CREATE_TOPO_MAP('A_TOPO', 'A_TOPOMAP');
EXECUTE SDO_TOPO_MAP.LOAD_TOPO_MAP('A_TOPOMAP', 'true');

BEGIN
  FOR A_rec IN (SELECT nazev, AB, geometry FROM A) LOOP
    INSERT INTO A_A_TOPO VALUES (
      A_rec.nazev,
      A_rec.AB,
      SDO_TOPO_MAP.CREATE_FEATURE('A_TOPO', 'A_A_TOPO',
        'FEATURE',A_rec.geometry)
    );
  END LOOP;
END;
/
```



	NAZEV	AB	FEATURE
1	A	AB	MDSYS.SDO_TOPO_GEOMETRY(3,1,1,22)
2	B	AB	MDSYS.SDO_TOPO_GEOMETRY(3,2,1,22)
3	C	CD	MDSYS.SDO_TOPO_GEOMETRY(3,3,1,22)
4	D	CD	MDSYS.SDO_TOPO_GEOMETRY(3,4,1,22)

obr. 6.5 Naplněná tabulka A\_A\_TOPO

```
BEGIN
  FOR AB_rec IN (SELECT nazev, ABCD FROM AB WHERE NAZEV = 'AB') LOOP
    INSERT INTO AB_A_TOPO VALUES (
      AB_rec.nazev,
      AB_rec.ABCD,
      SDO_TOPO_MAP.CREATE_FEATURE('A_TOPO', 'AB_A_TOPO',
        'FEATURE', 'AB='||AB_rec.ABCD)
    );
  END LOOP;
END;
/
```

```
BEGIN
  FOR AB_rec IN (SELECT nazev, ABCD FROM AB WHERE NAZEV = 'CD') LOOP
    INSERT INTO AB_A_TOPO VALUES (
      AB_rec.nazev,
      AB_rec.ABCD,
      SDO_TOPO_MAP.CREATE_FEATURE('A_TOPO', 'AB_A_TOPO',
        'FEATURE', 'AB='||AB_rec.ABCD)
    );
  END LOOP;
END;
/
```

	NAZEV	ABCD	FEATURE
1	AB	ABCD	MDSYS.SDO_TOPO_GEOMETRY(3,1,2,22)
2	CD	ABCD	MDSYS.SDO_TOPO_GEOMETRY(3,2,2,22)

obr. 6.6 Naplněná tabulka AB\_A\_TOPO

```

BEGIN
  FOR ABCD_rec IN (SELECT nazev FROM ABCD) LOOP
    INSERT INTO ABCD_A_TOPO VALUES (
      ABCD_rec.nazev,
      SDO_TOPO_MAP.CREATE_FEATURE('A_TOPO', 'ABCD_A_TOPO',
        'FEATURE', 'ABCD='||ABCD_rec.nazev)
    );
  END LOOP;
END;
/

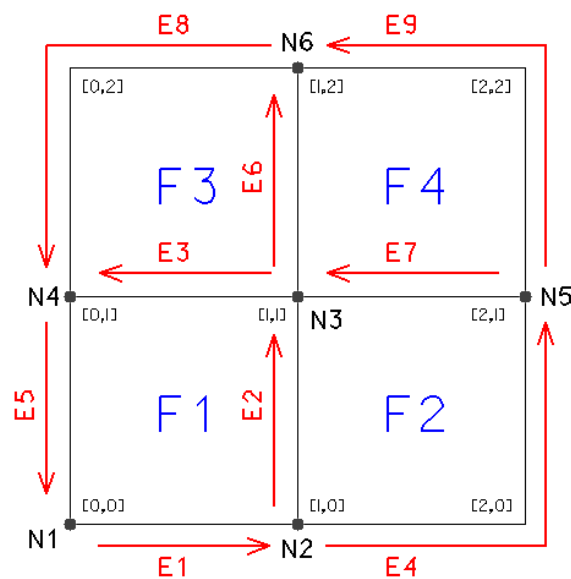
```

	NAZEV	FEATURE
1	ABCD	MDSYS.SDO_TOPO_GEOMETRY(3,1,3,22)

obr. 6.7 Naplněná tabulka ABCD\_A\_TOPO

```
EXECUTE SDO_TOPO.INITIALIZE_METADATA('A_TOPO');
```

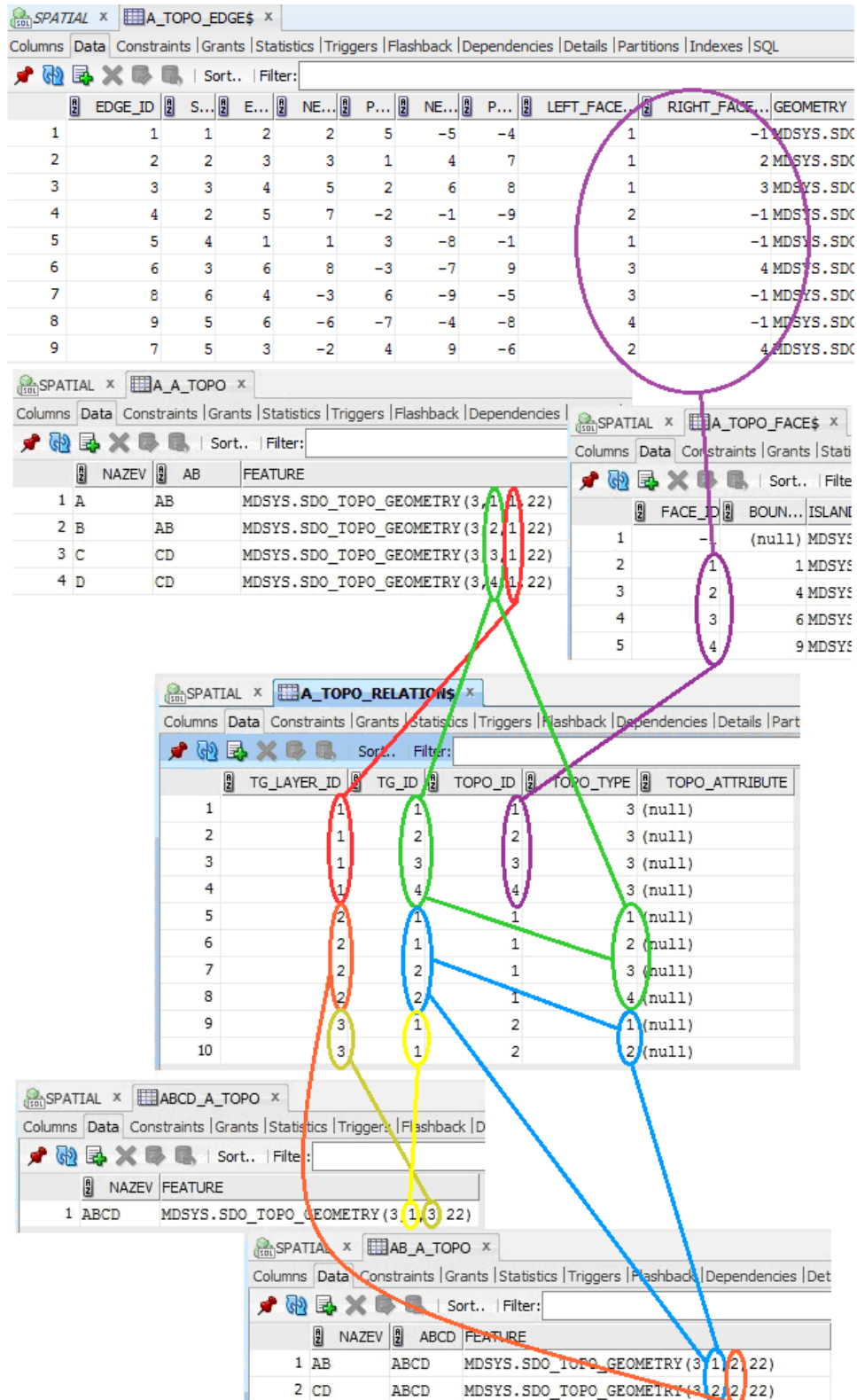
Nově vytvořená topologie A\_TOPO je znázorněná na obr. 6.8. Jsou zde schematicky znázorněná topologická primitiva.



obr. 6.8 Vytvořená topologie A\_TOPO s topologickými primitivy



Na obr. 6.9 je znázorněno propojení prvkových tabulek A\_A\_TOPO, AB\_A\_TOPO a ABCD\_A\_TOPO s tabulkami topologických primitiv A\_TOPO\_FACES\$ a A\_TOPO\_EDGES\$ přes tabulku A\_TOPO\_RELATIONS\$.



obr. 6.9 Přehled propojení prvkových tabulek s tabulkami topologických primitiv

## 6.2 Porovnání výsledků na topologiích A\_TOPO a F\_TOPO

Tato kapitola se zabývá již samotným ověřením funkčnosti vytvořené hierarchie v topologii F\_TOPO. Pro srovnání budou veškeré testy zároveň prováděny i na srovnávací topologii A\_TOPO, na které je výsledek snadno ověřitelný.

Pro vizualizaci dat byl použit nástroj GeoRaptor. Jedná se o open source projekt, který slouží pro manipulaci a vizualizaci prostorových dat, uložených ve formátu SDO\_GEOMETRY. Je možné provádět vizualizaci dat s 2D geometrií a vytvořeným prostorovým indexem. GeoRaptor není samostatná aplikace, funguje jako nadstavba nástroje SQL Developer [12]. Tento přídatný modul je dostupný na stránkách projektu [11], kde lze najít také návod na instalaci a bližší informace. Přehledný návod k používání aplikace lze získat v [13].

Vzhledem k tomu, že GeoRaptor umí vizualizovat data pouze ve formátu SDO\_GEOMETRY, bylo nutné data ve formátu SDO\_TOPO\_GEOMETRY upravit. Na data byla aplikována funkce GET\_GEOMETRY, která vrací geometrický popis prvku vyjádřeného pomocí topologických primitiv [4]. Tento popis se ukládá do sloupce datového typu SDO\_GEOMETRY, o který bylo třeba tabulku rozšířit. GeoRaptor pro vizualizaci vyžadoval, aby v tabulce byl nejdříve sloupec datového typu SDO\_GEOMETRY a až po něm sloupec datového typu SDO\_TOPO\_GEOMETRY. To pravděpodobně souviselo s komplikacemi, kdy SQL Developer s nainstalovaným GeoRaptorem nezobrazoval data ve sloupcích SDO\_TOPO\_GEOMETRY. SQL Developer neumí vkládat nový sloupec mezi stávající sloupečky, vždy vloží sloupec nakonec, proto jsem vytvářela tabulky nové s požadovaným pořadím sloupců. Pro vizualizaci dat bylo ještě nutné aktualizovat metadata v pohledu USER\_SDO\_GEOM\_METADATA. Na závěr byl vždy vytvořen prostorový index nad sloupcem datového typu SDO\_GEOMETRY. Celý postup této vizualizace je popsán v první úloze.

A nyní již několik testovacích úloh...

## 1) Jakými okresy je tvořen Jihočeský kraj? Které stěny jej tvoří?

V první úloze je zadáno, které okresy tvoří Jihočeský kraj (v topologii F\_TOPO). Převáděno na srovnávací topologii A\_TOPO, které ze čtverečků A, B, C, D tvoří obdélník AB.

### a) v topologii A\_TOPO:

```
SELECT a.nazev FROM a_a_topo a WHERE a.feature.tg_id IN (
  SELECT topo_type FROM a_topo_relation$ WHERE tg_id = (
    SELECT ab.feature.tg_id FROM ab_a_topo ab WHERE ab.nazev =
      'AB')
AND tg_layer_id = 2);
```

```
NAZEV
-----
A
B
```

obr. 6.10 Čtverečky tvořící obdélník AB

```
SELECT topo_id AS výčet_stěn_AB FROM a_topo_relation$ WHERE tg_id in (
  SELECT topo_type FROM a_topo_relation$ WHERE tg_id in (
    SELECT ab.feature.tg_id FROM ab_a_topo ab WHERE nazev = 'AB')
  AND tg_layer_id = 2)
AND tg_layer_id = 1;
```

```
VÝČET_STĚN_AB
-----
1
2
```

obr. 6.11 Výčet face\_id stěn tvořících obdélník AB

Obdélník AB tvoří čtverečky A a B (viz obr. 6.10), o čemž se můžeme zpětně přesvědčit na obr. 6.1. Tyto čtverečky jsou tvořeny stěnami, jejichž face\_id = 1 a 2 (viz obr. 6.11), zobrazeny jsou na obr. 6.12.

### Vizualizace:

Pro vizualizaci byla stávající tabulka A\_A\_TOPO rozšířena o sloupec *geometry* (datového typu SDO\_GEOMETRY) a pomocí funkce GET\_GEOMETRY (viz odst. 6.2) byl tento sloupec naplněn. Dále byla vytvořena nová tabulka A\_A\_TOPO\_AB\_G s požadovaným pořadím sloupců a naplněna daty, která vyhovují zadání této úlohy.

```

ALTER TABLE A_A_TOPO ADD (geometrie MDSYS.SDO_GEOMETRY);

UPDATE A_A_TOPO a
SET geometrie = a.FEATURE.get_geometry();

CREATE TABLE A_A_TOPO_AB_G AS (
  SELECT a.nazev, a.geometrie, a.feature
  FROM a_a_topo a WHERE a.nazev in (
    SELECT a.NAZEV FROM a_a_topo a WHERE a.feature.tg_id IN (
      SELECT topo_type FROM a_topo_relation$
      WHERE tg_id = (
        SELECT ab.feature.tg_id FROM ab_a_topo ab
        WHERE ab.nazev = 'AB')
      AND tg_layer_id = 2)
    )
);

```

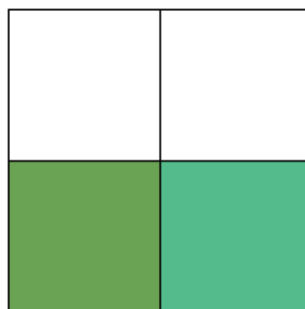
Dále byla aktualizována metadata v pohledu USER\_SDO\_GEOM\_METADATA a vytvořen prostorový index A\_A\_TOPO\_AB\_G nad sloupcem *geometrie*:

```

INSERT INTO user_sdo_geom_metadata (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES (
  'A_A_TOPO_AB_G',
  'GEOMETRIE',
  SDO_DIM_ARRAY(
    SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
    SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
  ),
  NULL
);

CREATE INDEX A_A_TOPO_AB_G_idx
ON A_A_TOPO_AB_G (GEOMETRIE)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```



obr. 6.12 Čtverečky A, B tvořící obdélník AB

**b) v topologii F\_TOPO:**

```

SELECT o.OKRES FROM f_okresy_topo o WHERE o.feature.tg_id IN (
  SELECT topo_type FROM f_topo_relation$ WHERE tg_id = (
    SELECT k.feature.tg_id FROM f_kraje_topo k

```

```

WHERE k.KRAJ = 'Jihočeský')
AND tg_layer_id = 2);
/

```

```

OKRES
-----
Jindřichův Hradec
Tábor
Strakonice
Český Krumlov
Prachatice
České Budějovice
Písek

```

7 rows selected

**obr. 6.13 Výčet okresů Jihočeského kraje**

```

SELECT topo_id AS výčet_stěn_JK FROM f_topo_relation$ WHERE tg_id in (
  SELECT topo_type FROM f_topo_relation$ WHERE tg_id in (
    SELECT k.feature.tg_id FROM f_kraje_topo k
    WHERE kraj = 'Jihočeský')
  AND tg_layer_id = 2)
AND tg_layer_id = 1;

```

Jihočeský kraj tvoří sedm okresů – Jindřichův Hradec, Tábor, Strakonice, Český Krumlov, Prachatice, České Budějovice a Písek (viz obr. 6.15). Jsou tvořeny sedmi stěnami, jejichž face\_id = 8, 18, 33, 48, 57, 64 a 66 (viz obr. 6.14).

```

VÝČET_STĚN_JK
-----
64
8
18
33
48
57
66

```

7 rows selected

**obr. 6.14 Stěny tvořící Jihočeský kraj**

### ***Vizualizace:***

Pro vizualizaci zadané úlohy byla vytvořena nová tabulka JC\_G, rozšířená o sloupec datového typu SDO\_GEOMETRY, která obsahuje ta data z původní tabulky F\_OKRESY\_TOPO, která vyhovují zadané úloze. Nakonec byla aktualizována metadata v pohledu USER\_SDO\_GEOM\_METADATA a vytvořen prostorový index JC\_G\_IDX.

```

CREATE TABLE F_OKRESY_TOPO_TEMP AS (SELECT*FROM f_okresy_topo);

ALTER TABLE f_okresy_topo_temp ADD (geometry MDSYS.SDO_GEOMETRY);

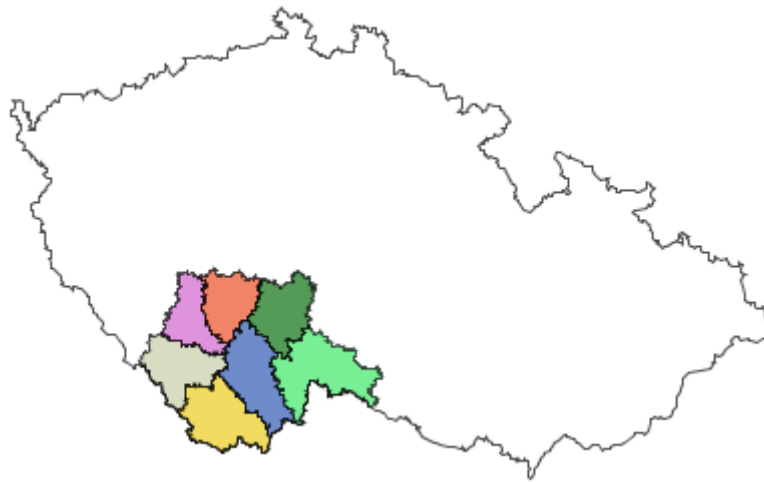
UPDATE f_okresy_topo_temp o
SET geometry = o.FEATURE.get_geometry();

CREATE TABLE JC_G AS (
    SELECT ot.okres, ot.geometry, ot.feature FROM f_okresy_topo_temp ot
    WHERE ot.feature.tg_id IN (
        SELECT topo_type FROM f_topo_relation$ WHERE tg_id = (
            SELECT kt.feature.tg_id FROM f_kraje_topo kt
            WHERE kt.kraj = 'Jihočeský')
        AND tg_layer_id = 2)
);

INSERT INTO user_sdo_geom_metadata (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES (
    'JC_G',
    'GEOMETRY',
    SDO_DIM_ARRAY(
        SDO_DIM_ELEMENT('X', -5000000, 5000000, 5E-8),
        SDO_DIM_ELEMENT('Y', -5000000, 5000000, 5E-8)
    ),
    NULL
);

CREATE INDEX JC_G_idx
ON JC_G (GEOMETRY)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```



**obr. 6.15 Sedm okresů Jihočeského kraje**

## 2) Vyberte hranice okresů v Jihočeském kraji:

Hranice okresů Jihočeského kraje (v topologii F\_TOPO) odpovídají ve srovnávací topologii A\_TOPO hranicím čtverečků, které tvoří obdélník AB.

**a) v topologii A\_TOPO:**

```

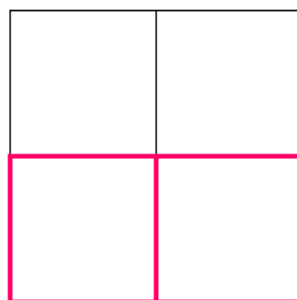
SELECT EDGE_ID, geometry FROM a_topo_edge$ WHERE (
  left_face_id IN (
    SELECT ar.topo_id FROM a_topo_relation$ ar
      WHERE ar.tg_layer_id = 1 AND ar.tg_id IN (
        SELECT ar.topo_type FROM a_topo_relation$ ar
          WHERE ar.tg_id = (
            SELECT ab.feature.tg_id FROM ab_a_topo ab
              WHERE ab.nazev = 'AB')
        )
    )
  AND ar.tg_layer_id = 2)
)
OR
right_face_id IN (
  SELECT ar.topo_id FROM a_topo_relation$ ar
    WHERE ar.tg_layer_id = 1 AND ar.tg_id IN (
      SELECT ar.topo_type FROM a_topo_relation$ ar
        WHERE ar.tg_id = (
          SELECT ab.feature.tg_id FROM ab_a_topo ab
            WHERE ab.nazev = 'AB')
        )
    )
  AND ar.tg_layer_id = 2)
);

```

EDGE_ID	GEOMETRY
1	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(0,0,1,0))
2	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,0,1,1))
3	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,1,0,1))
4	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,0,2,0,2,1))
5	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(0,1,0,0))
6	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(2,1,1,1))

**obr. 6.16 Výčet hran čtverečků, které tvoří obdélník AB**

Hranice čtverečků, které tvoří obdélník AB (viz obr. 6.17), jsou tvořeny šesti hranami, jejichž EDGE\_ID = 1, 2, 3, 4, 5 a 7 (viz obr. 6.16). Můžeme se o tom zpětně přesvědčit na obr. 6.8.



**obr. 6.17 Hrany, které tvoří hranice čtverečků, které tvoří obdélník AB (červeně)**

**b) v topologii F\_TOPO:**

```

SELECT edge_id, geometry FROM f_topo_edge$ WHERE (
  left_face_id IN (

```

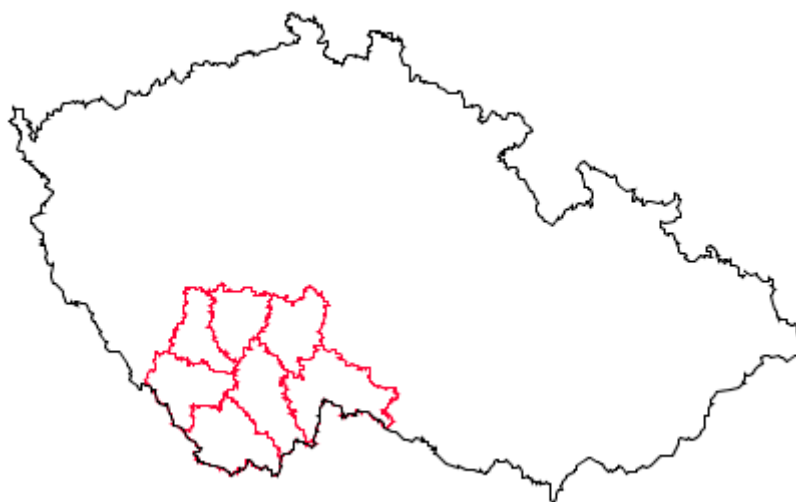
```

SELECT fr.topo_id FROM f_topo_relation$ fr
  WHERE fr.tg_layer_id = 1
  AND fr.tg_id IN (
    SELECT fr.topo_type FROM f_topo_relation$ fr
      WHERE fr.tg_id = (
        SELECT k.feature.tg_id FROM f_kraje_topo k
          WHERE k.kraj = 'Jihočeský')
  )
AND fr.tg_layer_id = 2)
)
OR
right_face_id IN (
  SELECT fr.topo_id FROM f_topo_relation$ fr
    WHERE fr.tg_layer_id = 1
    AND fr.tg_id IN (
      SELECT fr.topo_type FROM f_topo_relation$ fr
        WHERE fr.tg_id = (
          SELECT k.feature.tg_id FROM f_kraje_topo k
            WHERE k.kraj = 'Jihočeský')
    )
  AND fr.tg_layer_id = 2)
)
);

```

EDGE_ID	GEOMETRY
1	8 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-82
2	39 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-77
3	37 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-77
4	38 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-80
5	59 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-78
6	56 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-78
7	95 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-77
8	94 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-76

obr. 6.18 Výřez výčtu hran, které tvoří hranice okresů Jihočeského kraje



obr. 6.19 Hrany, které tvoří hranice okresů Jihočeského kraje (červeně)

Hranice okresů Jihočeského kraje odpovídají hranám zobrazeným na obr. 6.19. Výřez jejich výčtu je na obr. 6.18. Celkem tyto hranice tvoří 28 hran.



### 3) Vyberte hranici Jihočeského kraje:

Hranice Jihočeského kraje v topologii F\_TOPO odpovídá hranici obdélníku AB v topologii A\_TOPO:

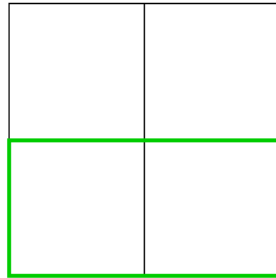
#### a) v topologii A\_TOPO:

```
SELECT EDGE_ID, geometry FROM a_topo_edge$ WHERE (
  left_face_id IN (
    SELECT ar.topo_id FROM a_topo_relation$ ar
    WHERE ar.tg_layer_id = 1
    AND ar.tg_id IN (
      SELECT ar.topo_type FROM a_topo_relation$ ar
      WHERE ar.tg_id = (
        SELECT ab.feature.tg_id
        FROM ab_a_topo ab
        WHERE ab.nazev = 'AB')
      AND ar.tg_layer_id = 2)
    )
  OR
  right_face_id IN (
    SELECT ar.topo_id FROM a_topo_relation$ ar
    WHERE ar.tg_layer_id = 1
    AND ar.tg_id IN (
      SELECT ar.topo_type FROM a_topo_relation$ ar
      WHERE ar.tg_id = (
        SELECT ab.feature.tg_id
        FROM ab_a_topo ab
        WHERE ab.nazev = 'AB')
      AND ar.tg_layer_id = 2)
    )
  )
AND NOT (
  left_face_id IN (
    SELECT ar.topo_id FROM a_topo_relation$ ar
    WHERE ar.tg_layer_id = 1
    AND ar.tg_id IN (
      SELECT ar.topo_type FROM a_topo_relation$ ar
      WHERE ar.tg_id = (
        SELECT ab.feature.tg_id
        FROM ab_a_topo ab
        WHERE ab.nazev = 'AB')
      AND ar.tg_layer_id = 2)
    )
  AND
  right_face_id IN (
    SELECT ar.topo_id FROM a_topo_relation$ ar
    WHERE ar.tg_layer_id = 1
    AND ar.tg_id IN (
      SELECT ar.topo_type FROM a_topo_relation$ ar
      WHERE ar.tg_id = (
        SELECT ab.feature.tg_id
        FROM ab_a_topo ab
        WHERE ab.nazev = 'AB')
      AND ar.tg_layer_id = 2)
    )
  )
```

);

EDGE_ID	GEOMETRY
1	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(0,0,1,0))
2	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,1,0,1))
3	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,0,2,0,2,1))
4	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(0,1,0,0))
5	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(2,1,1,1))

obr. 6.20 Výčet hran, které tvoří obdélník AB



obr. 6.21 Hranice, které tvoří hranice obdélníku AB (zeleně)

Hranice obdélníku AB je tvořena pěti hranami. Jejich výčet je na obr. 6.20. Hranice je znázorněna na obr. 6.21. Můžeme se o tom zpětně přesvědčit na obr. 6.8.

**b) v topologii F\_TOPO:**

```
SELECT edge_id, geometry FROM f_topo_edge$ WHERE (
  left_face_id IN (
    SELECT fr.topo_id FROM f_topo_relation$ fr
    WHERE fr.tg_layer_id = 1
    AND fr.tg_id IN (
      SELECT fr.topo_type FROM f_topo_relation$ fr
      WHERE fr.tg_id = (
        SELECT k.feature.tg_id
        FROM f_kraje_topo k
        WHERE k.kraj = 'Jihočeský')
    AND fr.tg_layer_id = 2)
  )
OR
  right_face_id IN (
    SELECT fr.topo_id FROM f_topo_relation$ fr
    WHERE fr.tg_layer_id = 1
    AND fr.tg_id IN (
      SELECT fr.topo_type FROM f_topo_relation$ fr
      WHERE fr.tg_id = (
        SELECT k.feature.tg_id
        FROM f_kraje_topo k
        WHERE k.kraj = 'Jihočeský')
    AND fr.tg_layer_id = 2)
  )
)
```

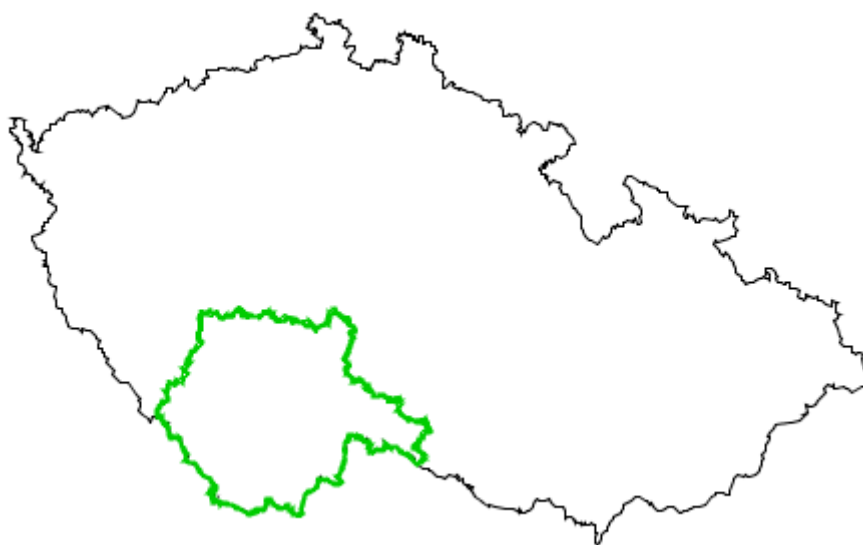
```

AND NOT (
  left_face_id IN (
    SELECT fr.topo_id FROM f_topo_relation$ fr
    WHERE fr.tg_layer_id = 1
    AND fr.tg_id IN (
      SELECT fr.topo_type FROM f_topo_relation$ fr
      WHERE fr.tg_id = (
        SELECT k.feature.tg_id
        FROM f_kraje_topo k
        WHERE k.kraj = 'Jihočeský')
    AND fr.tg_layer_id = 2)
  )
AND
right_face_id IN (
  SELECT fr.topo_id FROM f_topo_relation$ fr
  WHERE fr.tg_layer_id = 1
  AND fr.tg_id IN (
    SELECT fr.topo_type FROM f_topo_relation$ fr
    WHERE fr.tg_id = (
      SELECT k.feature.tg_id
      FROM f_kraje_topo k
      WHERE k.kraj = 'Jihočeský')
    AND fr.tg_layer_id = 2)
  )
);

```

EDGE_ID	GEOMETRY
1	8 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-82...
2	38 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-80...
3	59 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-78...
4	56 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-78...
5	148 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-80...
6	57 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-79...
7	153 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-67...
8	163 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-80...

obr. 6.22 Výřez výčtu hran, které tvoří hranici Jihočeského kraje



obr. 6.23 Hrany, které tvoří hranici Jihočeského kraje (zeleně)

Hranice Jihočeského kraje je tvořena 17 hranami. Výřez jejich výčtu je na obr. 6.22. Hranice je znázorněna na obr. 6.23.

#### 4) Které kraje tvoří ČR? Které stěny je tvoří?

Kraje České republiky v topologii F\_TOPO odpovídají dvěma obdélníkům AB a CD tvořící čtverec ABCD v topologii A\_TOPO.

##### a) v topologii A\_TOPO:

```
SELECT ab.nazev FROM ab_a_topo ab WHERE ab.feature.tg_id IN (
  SELECT topo_type FROM a_topo_relation$ WHERE tg_id=(
    SELECT abcd.feature.tg_id FROM abcd_a_topo abcd
    WHERE nazev = 'ABCD')
  AND tg_layer_id = 3);
```

```
NAZEV
-----
AB
CD
```

obr. 6.24 Výčet obdélníků tvořící čtverec ABCD

```
SELECT topo_id AS výčet_stěn FROM a_topo_relation$ WHERE tg_id in (
  SELECT topo_type FROM a_topo_relation$ WHERE tg_id in (
    SELECT ab.feature.tg_id FROM ab_a_topo ab)
  AND tg_layer_id = 2)
  AND tg_layer_id = 1;
```

```
VÝČET_STĚN
-----
1
2
3
4
```

obr. 6.25 Výčet stěn tvořící čtverec ABCD



obr. 6.26 Obdélníky tvořící čtverec ABCD

Čtverec ABCD tvoří dva obdélníky (viz obr. 6.26), jejich výčet je na obr. 6.24. Tyto dva obdélníky jsou tvořeny čtyřmi stěnami, jejich výčet je na obr. 6.25.

**b) v topologii F\_TOPO:**

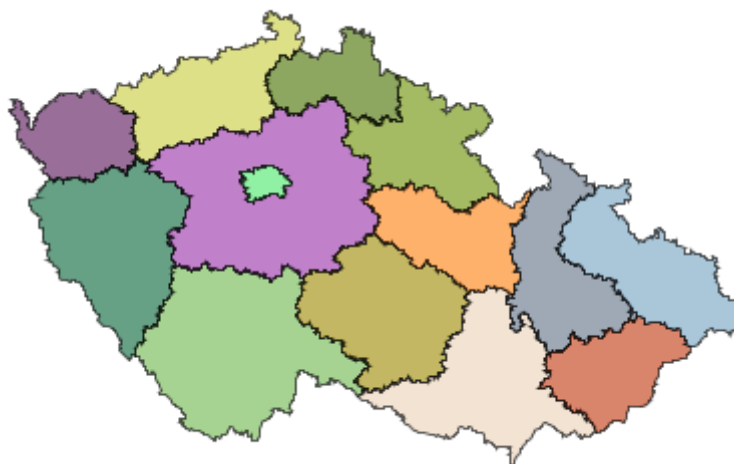
```
SELECT k.kraj FROM f_kraje_topo k WHERE k.feature.tg_id IN (  
  SELECT topo_type FROM f_topo_relation$ WHERE tg_id = (  
    SELECT cr.feature.tg_id FROM f_cr_topo cr  
      WHERE cr.stat = 'Česká republika')  
  AND tg_layer_id = 3);
```

KRAJ

-----  
Jihomoravský  
Plzeňský  
Ústecký  
Královéhradecký  
Liberecký  
Karlovarský  
Jihočeský  
Moravskoslezský  
Pardubický  
Olomoucký  
Zlínský  
Středočeský  
Vysočina  
Hlavní město Praha

14 rows selected

**obr. 6.27** Výčet krajů tvořící ČR



**obr. 6.28** Kraje ČR

Českou Republiku tvoří celkem 14 krajů, jejich výčet je na obr. 6.27. Tyto kraje jsou znázorněny na obr. 6.28. Tyto kraje jsou tvořeny 77 stěnami, výřez výčtu těchto stěn je na obr. 6.29.

```

VÝČET_STĚN_ČR
-----
35
23
45
55
61
57
1
31
46

```

obr. 6.29 Výčet stěn tvořící 14 krajů ČR

### 5) Které okresy tvoří ČR?

Okresy České republiky v topologii F\_TOPO odpovídají čtyřem čtverečkům A, B, C, D tvořící čtverec ABCD v topologii A\_TOPO.

#### a) v topologii A\_TOPO:

```

SELECT a.NAZEV FROM a_a_topo a WHERE a.feature.tg_id IN (
  SELECT topo_type FROM a_topo_relation$ WHERE tg_id IN(
    SELECT topo_type FROM a_topo_relation$ WHERE tg_id IN (
      SELECT abcd.feature.tg_id FROM abcd_a_topo abcd
      WHERE abcd.nazev = 'ABCD'
    )
  )
  AND tg_layer_id = 3)
AND tg_layer_id = 2);

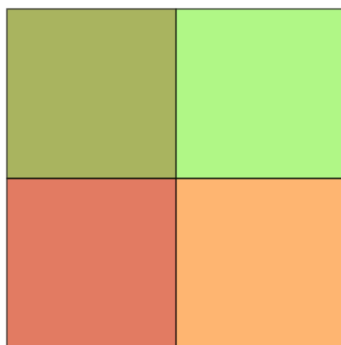
```

```

NAZEV
-----
A
B
C
D

```

obr. 6.30 Výčet čtverečků tvořících čtverec ABCD



obr. 6.31 Čtverec ABCD tvořený čtyřmi čtverečky A, B, C, D

**b) v topologii F\_TOPO:**

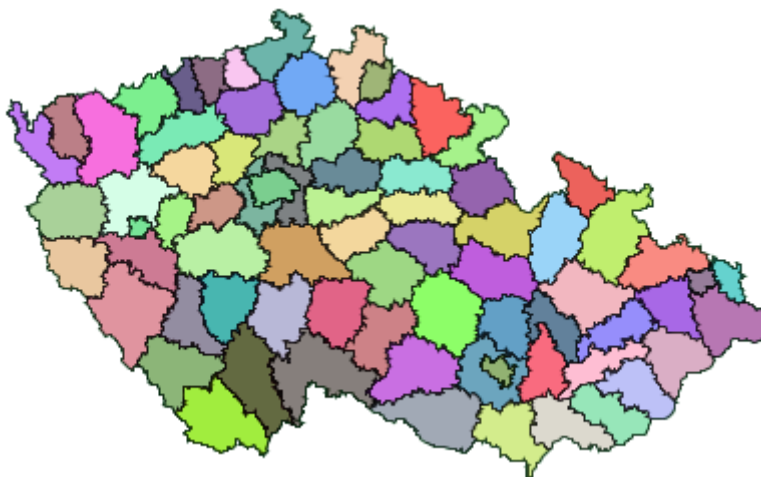
```
SELECT o.okres FROM f_okresy_topo o WHERE o.feature.tg_id IN (
  SELECT topo_type FROM f_topo_relation$ WHERE tg_id IN(
    SELECT topo_type FROM f_topo_relation$ WHERE tg_id IN (
      SELECT cr.feature.tg_id FROM f_cr_topo cr
      WHERE cr.stat = 'Česká republika'
    )
  )
  AND tg_layer_id = 3)
AND tg_layer_id = 2);
```

OKRES

-----

Hradec Králové  
 Náchod  
 Trutnov  
 Rychnov nad Kněžnou  
 Jičín  
 Brno-venkov  
 Hodonín

obr. 6.32 Výřez výčtu okresů ČR



obr. 6.33 77 okresů tvořící ČR

Českou Republiku tvoří 77 okresů, výřez z jejich výčtu je na obr. 6.32, jejich zobrazení na obr. 6.33.

### 6) Vyberte hranice všech okresů ČR:

Hranice všech okresů ČR v topologii F\_TOPO odpovídají hranicím čtyř čtverečků A, B, C, D v topologii A\_TOPO.

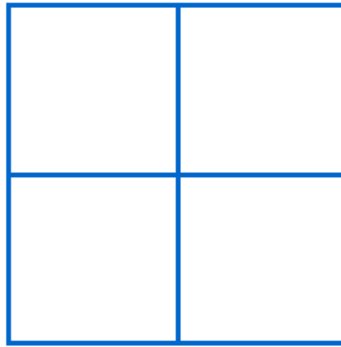
#### a) v topologii A\_TOPO:

```
SELECT EDGE_ID, geometry FROM a_topo_edge$ WHERE (
  left_face_id IN (
    SELECT topo_id FROM a_topo_relation$ WHERE tg_id IN(
      SELECT topo_type FROM a_topo_relation$ WHERE tg_id IN(
        SELECT topo_type FROM a_topo_relation$
          WHERE tg_id IN(
            SELECT abcd.feature.tg_id
              FROM abcd_a_topo abcd
              WHERE abcd.nazev = 'ABCD')
          AND tg_layer_id = 3)
      AND tg_layer_id = 2)
    AND tg_layer_id = 1)
  OR
  right_face_id IN (
    SELECT topo_id FROM a_topo_relation$ WHERE tg_id IN(
      SELECT topo_type FROM a_topo_relation$ WHERE tg_id IN(
        SELECT topo_type FROM a_topo_relation$
          WHERE tg_id IN(
            SELECT abcd.feature.tg_id
              FROM abcd_a_topo abcd
              WHERE abcd.nazev = 'ABCD')
          AND tg_layer_id = 3)
      AND tg_layer_id = 2)
    AND tg_layer_id = 1)
);
```

EDGE_ID	GEOMETRY
1	1 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(0, 0, 1, 0))
2	2 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(1, 0, 1, 1))
3	3 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(1, 1, 0, 1))
4	4 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(1, 0, 2, 0, 2, 1))
5	5 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(0, 1, 0, 0))
6	6 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(1, 1, 1, 2))
7	7 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(1, 2, 0, 2, 0, 1))
8	8 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(2, 1, 2, 2, 1, 2))
9	7 MDSYS.SDO_GEOMETRY(2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY(2, 1, 1, 1))

obr. 6.34 Výčet hran tvořící hranice čtverečků A, B, C, D





obr. 6.35 Hrany tvořící hranice čtverečků A, B, C, D

**b) v topologii F\_TOPO:**

```

SELECT EDGE_ID, geometry FROM f_topo_edge$ WHERE (
  left_face_id IN (
    SELECT topo_id FROM f_topo_relation$ WHERE tg_id IN(
      SELECT topo_type FROM f_topo_relation$ WHERE tg_id IN(
        SELECT topo_type FROM f_topo_relation$
          WHERE tg_id IN(
            SELECT cr.feature.tg_id
              FROM f_cr_topo cr
              WHERE cr.stat = 'Česká republika')
          AND tg_layer_id = 3)
      AND tg_layer_id = 2)
    AND tg_layer_id = 1)
  OR
  right_face_id IN (
    SELECT topo_id FROM f_topo_relation$ WHERE tg_id IN(
      SELECT topo_type FROM f_topo_relation$ WHERE tg_id IN(
        SELECT topo_type FROM f_topo_relation$
          WHERE tg_id IN(
            SELECT cr.feature.tg_id
              FROM f_cr_topo cr
              WHERE cr.stat = 'Česká republika')
          AND tg_layer_id = 3)
      AND tg_layer_id = 2)
    AND tg_layer_id = 1)
);

```

EDGE_ID	GEOMETRY
1	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-82
2	3 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-84
3	4 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-57
4	5 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-71
5	8 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-82
6	10 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-57
7	11 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-54
8	15 MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(-71

obr. 6.36 Výřez výčtu hran tvořících hranice okresů ČR



obr. 6.37 Hraný tvořící hranice okresů ČR

Hranice všech okresů ČR je tvořena 238 hranami, výřez jejich výčtu je na obr. 6.36. Hraný jsou zobrazeny na obr. 6.37.

### 7) Vyberte hranici ČR:

Hranice ČR v topologii F\_TOPO odpovídá hranici čtverce ABCD v topologii A\_TOPO.

#### a) v topologii A\_TOPO:

```

SELECT EDGE_ID, geometry FROM a_topo_edge$ WHERE (
  left_face_id IN (
    SELECT topo_id FROM a_topo_relation$ WHERE tg_id IN(
      SELECT topo_type FROM a_topo_relation$ WHERE tg_id IN(
        SELECT topo_type FROM a_topo_relation$
          WHERE tg_id IN(
            SELECT abcd.feature.tg_id
              FROM abcd_a_topo abcd
              WHERE abcd.nazev = 'ABCD')
          AND tg_layer_id = 3)
        AND tg_layer_id = 2)
      AND tg_layer_id = 1)
    AND right_face_id = -1
  )
OR(
  right_face_id IN (
    SELECT topo_id FROM a_topo_relation$ WHERE tg_id IN(
      SELECT topo_type FROM a_topo_relation$ WHERE tg_id IN(
        SELECT topo_type FROM a_topo_relation$
          WHERE tg_id IN (
            SELECT abcd.feature.tg_id
              FROM abcd_a_topo abcd
              WHERE abcd.nazev = 'ABCD')

```

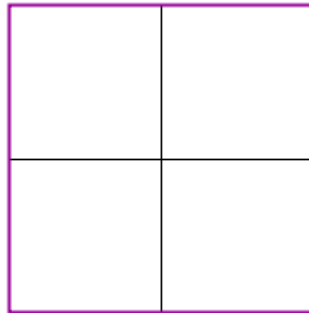
```

        AND tg_layer_id = 3)
        AND tg_layer_id = 2)
        AND tg_layer_id = 1)
AND left_face_id = -1);

```

EDGE_ID	GEOMETRY
1	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(0,0,1,0))
2	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,0,2,0,2,1))
3	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(0,1,0,0))
4	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(1,2,0,2,0,1))
5	MDSYS.SDO_GEOMETRY(2002,NULL,NULL,MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1),MDSYS.SDO_ORDINATE_ARRAY(2,1,2,2,1,2))

obr. 6.38 Výčet hran tvořící hranice čtverce ABCD



obr. 6.39 Hrany tvořící hranici čtverce ABCD (fialově)

**b) v topologii F\_TOPO:**

```

SELECT EDGE_ID, geometry FROM f_topo_edge$ WHERE (
    left_face_id IN (
        SELECT topo_id FROM f_topo_relation$ WHERE tg_id IN(
            SELECT topo_type FROM f_topo_relation$ WHERE tg_id IN(
                SELECT topo_type FROM f_topo_relation$
                WHERE tg_id IN (
                    SELECT cr.feature.tg_id
                    FROM f_cr_topo cr
                    WHERE cr.stat = 'Česká republika')
                AND tg_layer_id = 3)
            AND tg_layer_id = 2)
        AND tg_layer_id = 1)
    AND right_face_id = -1)

OR (
    right_face_id IN (
        SELECT topo_id FROM f_topo_relation$ WHERE tg_id IN(
            SELECT topo_type FROM f_topo_relation$ WHERE tg_id IN(
                SELECT topo_type FROM f_topo_relation$
                WHERE tg_id IN (
                    SELECT cr.feature.tg_id
                    FROM f_cr_topo cr
                    WHERE cr.stat = 'Česká republika')
                AND tg_layer_id = 3)
            AND tg_layer_id = 2)
        AND tg_layer_id = 1)
    AND left_face_id = -1
);

```

EDGE_ID	GEOMETRY
1	3 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-84:
2	31 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-44:
3	41 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-80:
4	60 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-90:
5	70 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-46:
6	77 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-88:
7	98 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-88:
8	107 MDSYS.SDO_GEOMETRY (2002, NULL, NULL, MDSYS.SDO_ELEM_INFO_ARRAY (1, 2, 1), MDSYS.SDO_ORDINATE_ARRAY (-62:

obr. 6.40 Výřez výčtu hran tvořících hranici ČR



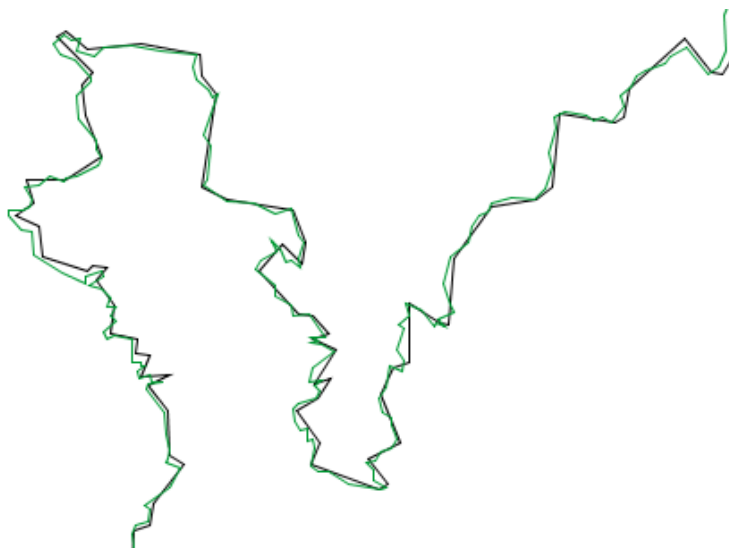
obr. 6.41 Hrany tvořící hranici ČR (fialově)

Hranici ČR tvoří 37 hran, jejich výčet je na obr. 6.40. Hrany jsou zobrazeny na obr. 6.41.

## 7 Závěr

Hlavním cílem této diplomové práce bylo vytvoření hierarchie nad topologicky strukturovanými daty a její modelování. Vstupními daty byly tři vrstvy ve formátu SHP – vrstva okresů (okresy.shp), vrstva krajů (kraje.shp) a vrstva nesoucí hranici České Republiky (cr.shp). Tato data byla převedena do databáze Oracle Spatial, do objektově relačního modelu SDO\_GEOMETRY. Tento převod byl proveden v prostředí Bentley PowerMap. Následně byla tato data převedena z objektově relačního modelu do topologického formátu SDO\_TOPO\_GEOMETRY.

Zároveň byla vybudována tříúrovňová hierarchie nad topologicky strukturovanými daty, kde na nejnižší úrovni hierarchie je vrstva okresů České Republiky. Další vrstva – vrstva krajů – již byla vytvořena z dat na nižší úrovni, čili z okresů příslušejících k danému kraji. Zůstala tak zachována nejnižší úroveň detailu, která příslušela okresům. Poté byla vytvořena třetí vrstva – vrstva s Českou Republikou – na nejvyšší úrovni hierarchie. Tato vrstva byla vytvořena z prvků tvořící druhou vrstvu (vrstvu krajů), nicméně i zde zůstala zachována úroveň detailu nejnižší vrstvy, tj. vrstvy okresů. Na obr. 7.1 Porovnání hranice ČR - hranice ze vstupních dat (černě) a hranice vygenerovaná pomocí hierarchie (zeleně) je vidět porovnání hranice České Republiky v Ašském výběžku. Hranice republiky ze vstupních dat je vykreslena černou čarou, zelenou čarou je vykreslená vygenerovaná hranice republiky pomocí hierarchie. Tato vygenerovaná hranice je stejně podrobná, jako hranice okresů ze vstupních dat.



**obr. 7.1 Porovnání hranice ČR - hranice ze vstupních dat (černě) a hranice vygenerovaná pomocí hierarchie (zeleně)**

Tyto vygenerované hranice nejsou uloženy ve formátu SDO\_GEOMETRY, nicméně jsou uloženy jako topologická primitiva (uzly, hrany, stěny), na která je z jiných tabulek odkazováno. Dochází tak k tomu, že každý prvek je v databázi uložen pouze jednou, data se netuplují a tím se tak zabraňuje nekonzistenci databáze. V práci je toto topologické uložení pomocí datového typu SDO\_TOPO\_GEOMETRY podrobně popsáno, včetně významů jednotlivých atributů na různých úrovních hierarchie.

V průběhu práce s vytvořenou hierarchií jsem se setkala s komplikací, kdy na třech místech v republice na sebe okresy vzájemně „neseděly“. Jednalo se o drobné rozdíly v souřadnicích, které však vedly k vytvoření mnohonásobného množství uzlů, hran a stěn. Vznikly tak stěny, které byly součástí více okresů a zároveň stěny, které žádnému okresu nepříslušely. Jako řešení této situace jsem vytvořila novou topologii, do které jsem nainportovala pouze 74 okresů (bez tří problematických). Poté jsem tyto tři okresy vytvořila z hranic okresů, které tyto okresy obklopují. Následně byla nad všemi okresy opět vytvořena hierarchie s vrstvou krajů a České Republiky.

V závěru práce je uvedeno několik příkladů, pomocí nichž je ověřena funkčnost vytvořené hierarchie. Tato hierarchie umožňuje pomocí SQL dotazů vracet geometrie pro prvky na vyšších úrovních hierarchie. Výsledky byly vizualizovány pomocí nástroje GeoRaptor.

## Seznam použité literatury

- [1] BENTLEY. *Bentley Map V8i*. [online]. Bentley Systems ČR s.r.o. 2011 [cit. 25.2.2011] <http://www.bentley.com/cs-CZ/Products/Bentley+Map/PowerMap-Product.htm>
- [2] Dgn2Sdo Application. Working with Oracle. [online]. [cit. 5.3.2012] <http://docs.bentley.com/en/PowerMap/geooutlkhelpp276.html>
- [3] Corcione, G.; Jayant, S. A Service Oriented Architecture Approach to Semi-automated Updates of Topographic Databases, [online]. [cit. 2.4.2012] 2006. Directions Magazine. <http://www.directionsmag.com/articles/a-service-oriented-architecture-approach-to-semi-automated-updates-of-topog/123066>
- [4] Janečka, K. Modelování konzistentní báze geodat na úrovni datového modelu katastru nemovitostí. Disertační práce, Západočeská Univerzita, Plzeň, 2009.
- [5] Murray, Ch. *Oracle Spatial Topology AND Network Data Model*, 11g Release 1 (11.1). [online]. [cit. 24.10.2010]; Oracle Corp.: USA, 2008. [http://docs.oracle.com/cd/B28359\\_01/appdev.111/b28399.pdf](http://docs.oracle.com/cd/B28359_01/appdev.111/b28399.pdf)
- [6] PENNINGA, F., QUAK, W., TIJSEN, T., VAN OOSTEROM, P. Storage AND Querying of Topological Structures in Oracle Spatial. [online]. [cit. 4.4.2012]. [ftp://ftp.ecn.purdue.edu/jshan/proceedings/topology\\_ucl2005/doc/Storage%20and%20Querying%20of%20Topological%20Structures%20in%20Oracle%20Spatial.pdf](ftp://ftp.ecn.purdue.edu/jshan/proceedings/topology_ucl2005/doc/Storage%20and%20Querying%20of%20Topological%20Structures%20in%20Oracle%20Spatial.pdf)
- [8] KOTHURI, R., GODFRIND, A., BEINAT, E. *Pro Oracle Spatial*. USA: Apress, 2004. 680 p. ISBN 1-59059-383-9.
- [9] JANEČKA, K., ČADA, V. Possibilities of Storage of Spatial Data of Real Estate Registry Information System. [online]. 2008 [cit. 9.4.2012]. [http://www.fig.net/pub/fig2008/papers/ts01e/ts01e\\_02\\_janecka\\_cada\\_2950.pdf](http://www.fig.net/pub/fig2008/papers/ts01e/ts01e_02_janecka_cada_2950.pdf)

- [10] Murray, Ch. *Oracle Spatial Developers Guide*, 11g Release 1 (11.1). [online]. [cit. 25.9.2010] ; Oracle Corp.: USA, 2009.  
[http://docs.oracle.com/cd/B28359\\_01/appdev.111/b28400.pdf](http://docs.oracle.com/cd/B28359_01/appdev.111/b28400.pdf)
- [11] Spatial viewer for Oracle SQL Developer. Sourceforge. [online]. [cit. 12.4.2012]  
<http://sourceforge.net/projects/georaptor/>
- [12] Oracle SQL Developer. [online], [cit. 12.4.2012]  
<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>
- [13] Spatial View. [online], [cit. 28.2.2012]  
<http://dfn.dl.sourceforge.net/project/georaptor/Spatial%20View.pdf>
- [14] Reportáž - Bentley Forum 2011. *GISportal.cz*. [online]. [cit. 19.4.2012]  
<http://www.gisportal.cz/2011/11/reportaz-bentley-forum-2011/>
- [15] Bentley Map, 2011. VARS. [online]. [cit. 19.4.2012]  
<http://web.vars.cz/cs/produkty-a-sluzby/inzenyrske-aplikace-pro-projektovani/microstation-geographics-bentley-geospatial/bentley-map.html>
- [16] Oracle Spatial. Wikipedie. [online]. [cit. 19.4.2012]  
[http://cs.wikipedia.org/wiki/Oracle\\_Spatial](http://cs.wikipedia.org/wiki/Oracle_Spatial)



## Struktura přiloženého CD

- Zdrojova\_data            Zdrojová data
  - CR                        Soubory cr.dbf, cr.shp, cr.shx
  - Kraje                    Soubory kraje.dbf, kraje.prj, kraje.sbn, kraje.sbx, kraje.shp, kraje.shx
  - Okresy                  Soubory okresy.dbf, okresy.prj, okresy.sbn, okresy.sbx, okresy.shp, okresy.shx
  
- DP\_Figallova            Soubor DP\_Figallova.pdf – text této diplomové práce
  
- Hierarchie                Soubor hierarchie.sql - skript pro vytvoření topologie F\_TOPO s hierarchií