

**Vysoká škola báňská - Technická univerzita Ostrava  
Institut geoinformatiky**

**Prostorové databáze  
Zápočtová úloha**  
Užití prostorových operátorů v Oracle

**8.5.2011**

**GN1GIF01**

**Petr Brotánek**

# 1 Zadání

Úkolem je načíst soubory ESRI Shapefile do prostředí databáze Oracle a zde nad nimi vytvořit dotazy s využitím prostorových operátorů. Výsledky mají být vhodným způsobem graficky zobrazeny.

Mezi použité programové prostředky patří SQL Developer a modul GeoRaptor pro zobrazování prostorových dat.

## 2 Příprava vstupních dat

Pro zadanou úlohu jsem použil data územního členění České republiky a volně dostupná data z Digitální báze vodohospodářských dat (DIBAVOD). Z územního členění jsou to vrstvy krajů a okresů, z databáze DIBAVOD vrstva vodních toků (hrubé úseky) a vrstva jezů.

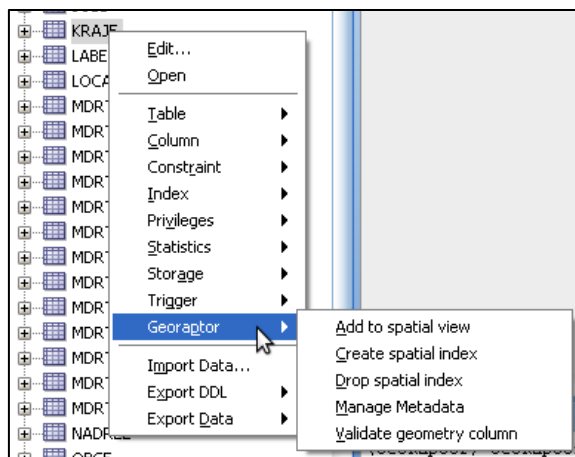
Import vektorových vrstev byl uskutečněn prostřednictvím knihoven *ojdbc14.jar*, *sdoutl.jar* a *sdoapi.jar*. Import proběhl z příkazové řádky nejprve nastavením cesty do adresáře s knihovnamí Oracle, poté zadáním příkazu:

```
java -classpath ojdbc14.jar;sdoutl.jar;sdoapi.jar
oracle.spatial.util.SampleShapefileToJGeomFeature
-h localhost -p 1521 -s xe -u hr -d [heslo]
-t [název tabulky v Oracle] -f C:\Data\[vrstva bez přípony]
```

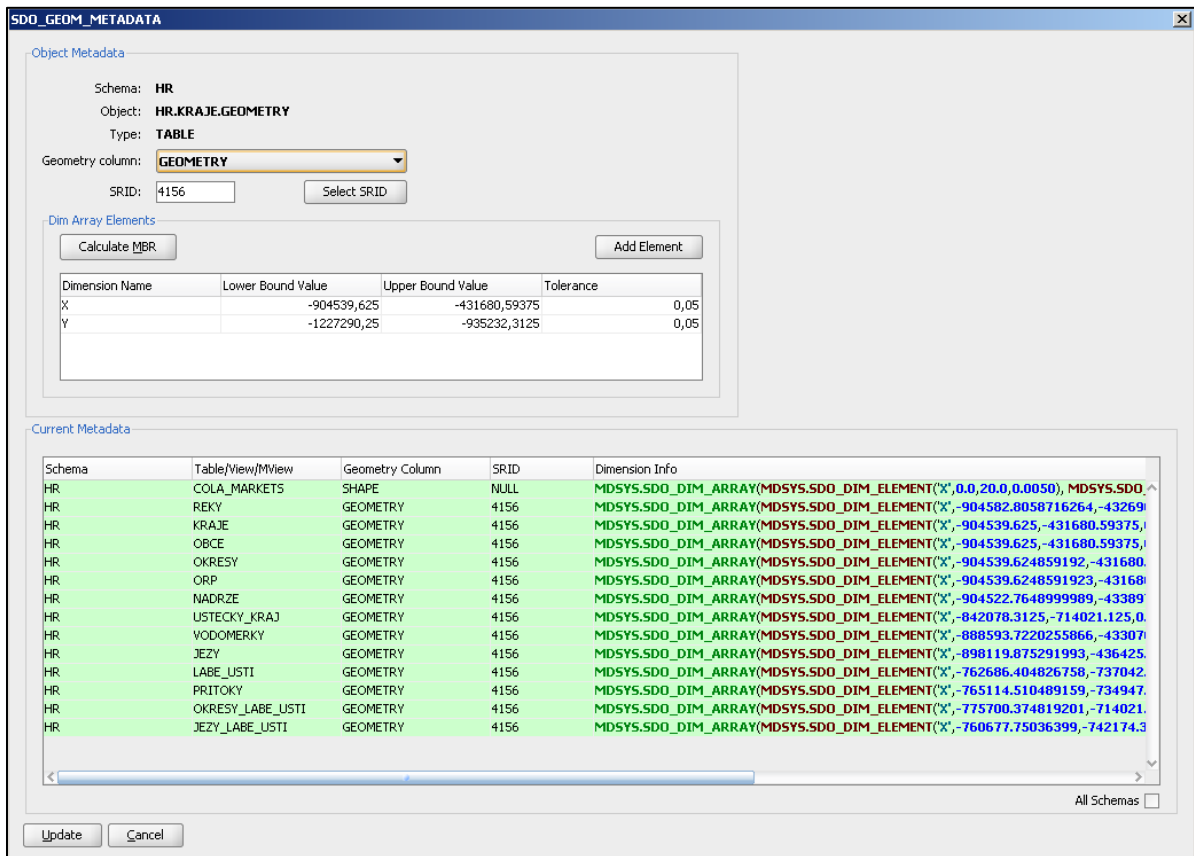
Příkazový řádek poté informuje o průběhu importu a počtu importovaných prvků.

## 3 Práce s prostorovými daty v SQL

Potřeby zpracování prostorových dat jsou velice specifické. Každá tabulka s prostorovými daty musí mít pro správnou funkčnost vytvořena metadata prostorové složky dat a prostorové indexy. To lze zařídit přímo příkazy prostorového rozšíření jazyka SQL, nebo v našem případě pomocí modulu GeoRaptor. Tuto možnost nalezneme po stisknutí pravého tlačítka myši na vybranou tabulku:



Editaci metadat prostorové složky dat zobrazuje následující obrázek:



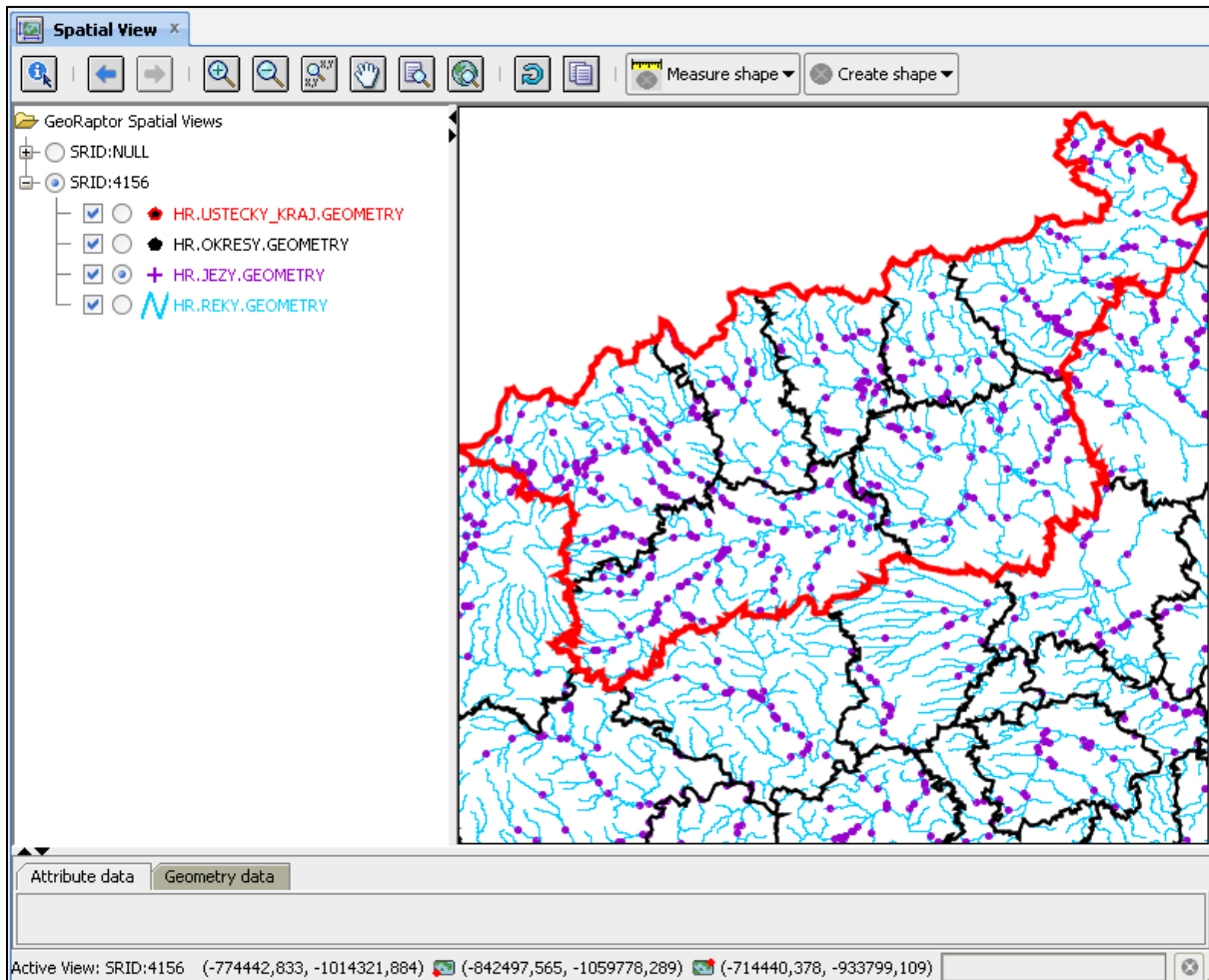
Prostorové indexy jsou pro vyhledávání pomocí prostorových operátorů nutné. Jejich vytvoření pomocí nástroje modulu GeoRaptor je velice jednoduché.

## 4 Zpracování praktické úlohy

Celá úloha se zabývala řekou Labe v úseku, kde protéká Ústeckým krajem. Nejprve zobrazíme všechna data v modulu GeoRaptor. Z vrstvy krajů je zde proveden výběr Ústeckého kraje:

```
--výběr ústeckého kraje
```

```
CREATE TABLE ustecky_kraj AS  
SELECT * FROM kraje WHERE nazev='Ústecký';
```



Dále chceme najít a spočítat všechny přítoky do Labe v Ústeckém kraji. Nejprve vybereme tu část řeky Labe, která se nachází v Ústeckém kraji, poté hledáme její přítoky a nakonec je spočítáme:

```
--vytvoření tabulky s úseky Labe, které se nachází v Ústeckém kraji
```

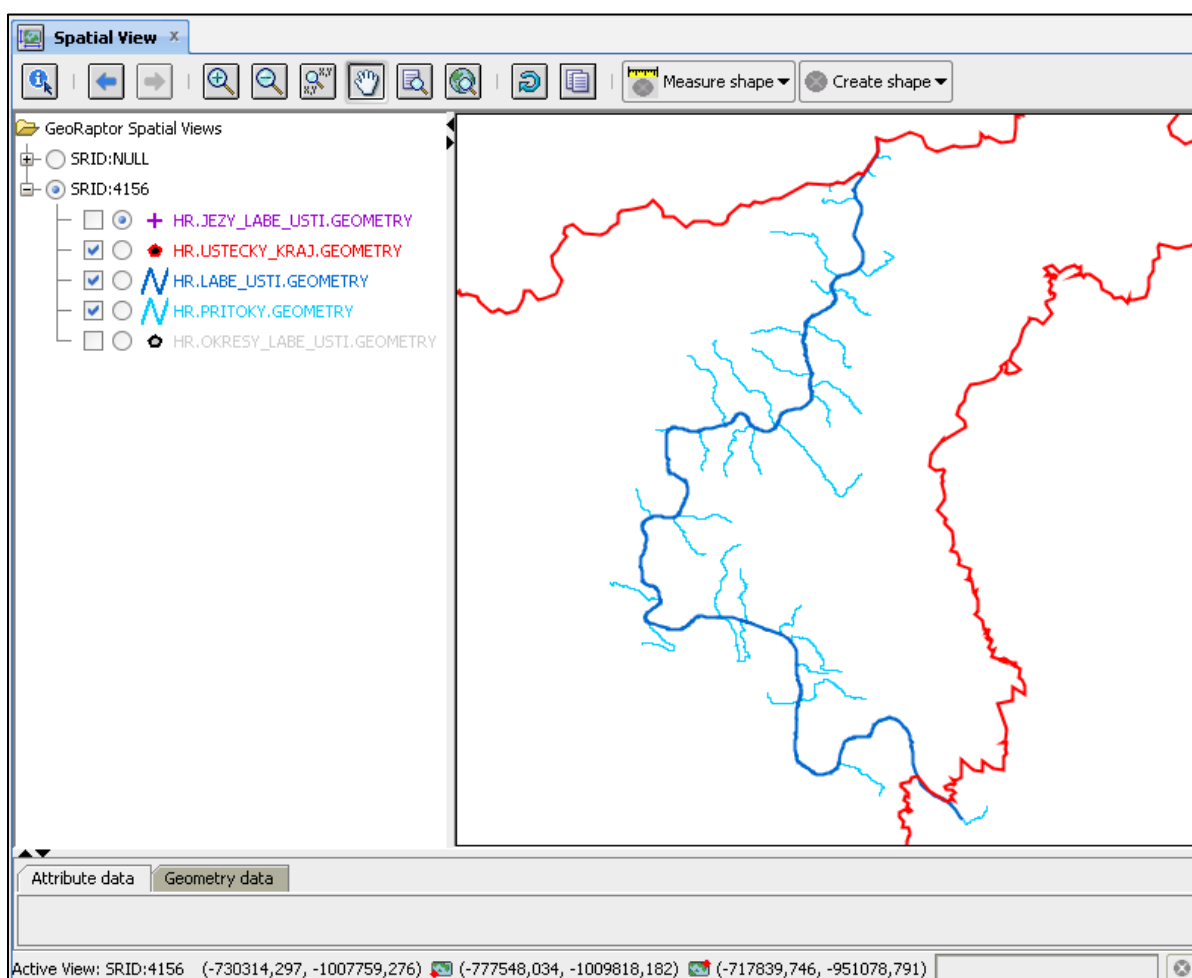
```
CREATE TABLE labe_usti AS  
SELECT r.*  
FROM reky r, ustecky_kraj u  
WHERE  
  (SDO_GEOM.SDO_INTERSECTION(r.geometry,u.geometry,0.05) IS NOT NULL)  
  AND  
  (r.naz_tok='Labe');
```

--nalezení všech přítoků Labe v Ústeckém kraji

```
CREATE TABLE pritoky AS
SELECT r.*
FROM reky r, labe_usti l
WHERE (r.naz_tok != 'Labe')
AND (SDO_ANYINTERACT(r.geometry,l.geometry) = 'TRUE');
```

--počet přítoku Labe je výsledkem následujícího dotazu

```
SELECT COUNT(p.naz_tok) pocet_pritoku_labe
FROM pritoky p;
```



Statement Output x Query Result x

SQL | All Rows Fetched: 1 in 0,881 seconds

POCET_PRITOKU_LABE	
1	71

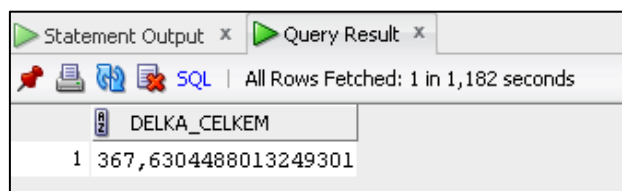
Dále nás zajímá celková délka řeky Labe a délka Labe na území Ústeckého kraje (hodnoty v kilometrech):

--celková délka řeky Labe

```
SELECT SUM(SDO_GEOM.SDO_LENGTH(r.geometry,0.05)/1000) delka_celkem  
FROM reky r  
WHERE naz_tok='Labe';
```

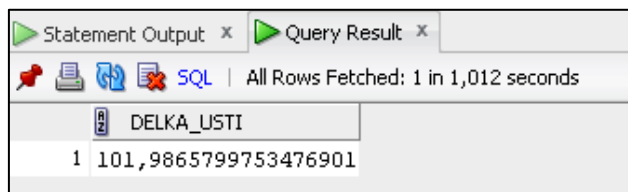
--délka řeky Labe v Ústeckém kraji

```
SELECT SUM(SDO_GEOM.SDO_LENGTH(l.geometry,0.05)/1000) delka_usti  
FROM labe_usti l;
```



The screenshot shows a SQL Developer window with two tabs: 'Statement Output' and 'Query Result'. The 'Query Result' tab is active, displaying the results of the first query. The status bar indicates 'All Rows Fetched: 1 in 1,182 seconds'. The column header is 'DELKA\_CELKEM' and the result is a single row with the value '367,6304488013249301'.

DELKA_CELKEM
367,6304488013249301



The screenshot shows a SQL Developer window with two tabs: 'Statement Output' and 'Query Result'. The 'Query Result' tab is active, displaying the results of the second query. The status bar indicates 'All Rows Fetched: 1 in 1,012 seconds'. The column header is 'DELKA\_USTI' and the result is a single row with the value '101,9865799753476901'.

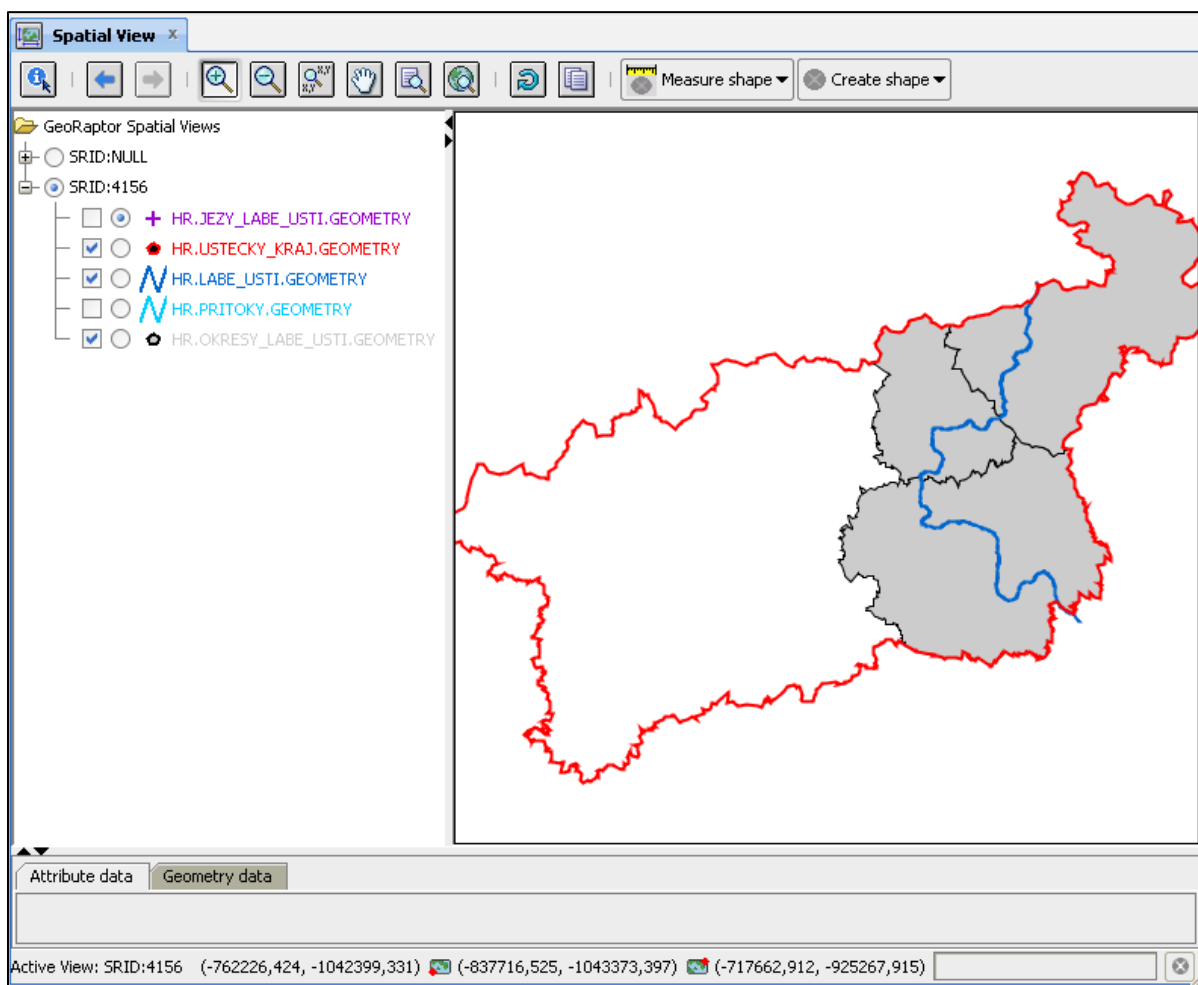
DELKA_USTI
101,9865799753476901

Ted' vybereme okresy Ústeckého kraje, přes které teče řeka Labe. Byl použit složený dotaz, aby se nalezené okresy neopakovaly v případě, kdy jedním okresem protéká více úseku řeky:

--složený dotaz vybere okresy, kterými teče Labe v Ústeckém kraji

```
CREATE TABLE okresy_labe_usti AS
SELECT o.*
FROM okresy o,
(
  SELECT DISTINCT(o.nazev) nazev
  FROM okresy o, labe_usti l
  WHERE SDO_CONTAINS(o.geometry,l.geometry) = 'TRUE'
) x
WHERE o.nazev = x.nazev;
```

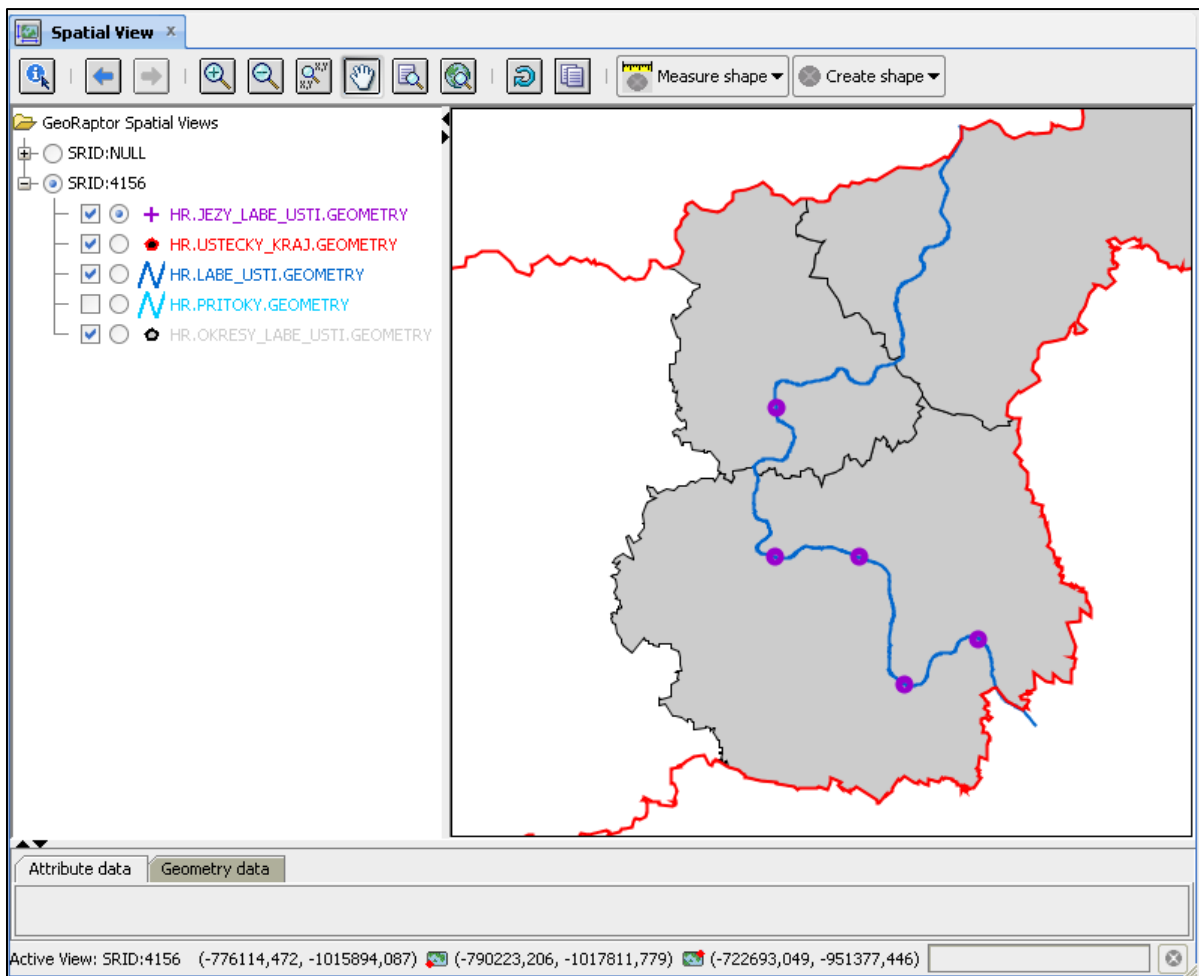
Výsledkem jsou okresy Děčín, Ústí nad Labem a Litoměřice:



Poslední dílčí úkol řeší nalezení jezů, které se nachází na Labi v Ústeckém kraji. Jezy nelze hledat přímo na linii řeky, protože důsledkem generalizace nebo nepřesnosti se jezy nacházejí až několik desítek metrů od linie. Proto byla pro nalezení výsledků využita obalová zóna linie o šířce 100 (m):

```
--nalezení jezů na Labi

CREATE TABLE jezy_labe_usti AS
SELECT j.*
FROM labe_usti l, jezy j
WHERE SDO_ANYINTERACT
(
  j.geometry, SDO_GEOM.SDO_BUFFER(l.geometry, 100, 0.05)
) = 'TRUE';
```



## 5 Závěr

Touto prací jsem si vyzkoušel základní možnosti využití prostorových operátorů v databázi Oracle. Zpracování prostorových dat přímo v databázi přináší řadu výhod, ať už je to zabezpečení dat nebo zmenšení velikosti přenášených dat. Proto se jedná jistě o technologii, která má před sebou zajímavou budoucnost.