



Semestrální projekt 153UZPD

SKUPINA D13: ŠTĚPÁN KAISER, DAVID HANOUSEK
12.5.2013

Obsah

1. ÚVOD

1. Zadání projektu
2. Výběr tématu

2. POUŽITÁ DATA A SOFTWARE

1. Open Street Map
2. PostGIS
3. Další data a software

3. TVORBA TEMATICKÝCH VRSTEV

1. Vrstvy z Open Street Map
2. Vrstvy ze schématu GIS1

Obsah

- 4. VALIDACE DAT
- 5. TVORBA TOPOLOGIE
- 6. POSTGIS RASTER
- 7. pgROUTING
- 8. DOTAZY
 - 1. Atributové
 - 2. Prostorové

1.1 ÚVOD

Zadání projektu

- Navrhnout a vytvořit tematické vrstvy na základě dat OpenStreetMap a schématu GIS1
- Aplikace testů datové integrity a odstranění případné nekonzistence vstupních dat
- Tvorba tutoriálu pro výuku PostGIS – sadu atributových a prostorových dotazů

1.2 ÚVOD

Výběr tématu

- Tvorba jednoduché databáze pro lyžaře:
 - Informace o sjezdovkách
 - Informace o ubytovacích zařízeních
 - Informace o možnostech dopravy
 - Další doplňující údaje
- Snaha vytvořit projekt, který doposud nebyl v předmětu UZPD realizován
- Celková prezentace prostředí PostGIS a jeho nadstaveb:
 - PostGIS Raster
 - PostGIS Topology
 - pgRouting

2.1 POUŽITÁ DATA A SOFTWARE

OpenStreetMap



- Projekt zaměřený na tvorbu svobodných geografických dat
- Jeden ze dvou hlavních zdrojů dat pro náš projekt

“Much like Wikipedia challenged Encyclopedia Britannica and Linux took on Microsoft Windows, OpenStreetMap is battling Google Maps, and at least in some cases, it’s winning.”

(Caleb Garling, www.wired.com, 01.09.20012)

2.2 POUŽITÁ DATA A SOFTWARE

PostGIS



- Open source software
- Přidání podpory pro geografické objekty k objektově relačnímu databázovému systému PostgreSQL
- Implementace ,Simple Features for SQL' konsorcia Open Geospatial Consortium
- Nadstavby: PostGIS Raster, PostGIS Topology a pgRouting

2.3 POUŽITÁ DATA A SOFTWARE

Další data a software

- DMT z datasetu FreeGeoDataCZ
 - DMT rastr pro území ČR
 - Velikost pixelů: 60 x 90 metrů
 - Zdroj výškových informací
- Quantum GIS
 - Svobodný geografický informační systém
 - V našem projektu použit pro vizualizaci dat
- pgAdmin
 - Svobodné grafické administrační rozhraní pro PostgreSQL
 - V tomto rozhraní byla tvořena naše databáze
 - Prostorové a atributové dotazy

3.1 TVORBA TEMATICKÝCH VRSTEV

Vrstvy z OpenStreetMap

- Z OSM byly vytvořeny tyto vrstvy:

- Ubytování (Bodová vrstva)
- Vleky (Liniová vrstva)

- Příklad tvorby vrstvy:

```
CREATE TABLE          ubytovani
AS SELECT  osm_id, geom, name AS nazev, tourism AS typ
FROM      czech_point
WHERE     tourism IN ('chalet', 'guest_house', 'hostel', 'hotel');
```

- Nastavení prostorového indexu a primárního klíče (probíhalo u všech vrstev stejně)

```
ALTER TABLE obce
ADD PRIMARY KEY (ogc_fid);
CREATE INDEX obce_geom
ON obce
USING gist (geom);
```

3.1 TVORBA TEMATICKÝCH VRSTEV

Vrstvy z OpenStreetMap

- U vrstvy ubytování bylo třeba převést některé polygony na body

```
CREATE TABLE      ubytovani_pol
AS SELECT          osm_id, geom, name AS nazev, tourism AS typ
FROM              czech_polygon AS cp
WHERE             tourism IN ('chalet', 'guest_house', 'hostel', 'hotel')
AND               osm_id NOT IN (
                                SELECT      cz_p.osm_id
                                FROM          ubytovani AS ub
                                JOIN         czech_polygon AS cz_p
                                ON           ST_contains(cz_p.geom,ub.geom)
                                WHERE        cz_p.tourism IN ('chalet', 'guest_house', 'hostel', 'hotel'));

CREATE TABLE      ubytovani_bod
AS SELECT          osm_id, ST_Centroid (ubytovani_pol.geom) AS geom, nazev, typ
FROM              ubytovani_pol;
INSERT INTO        ubytovani(osm_id, geom, nazev, typ)
SELECT            *
FROM              ubytovani_bod;
DROP TABLE       ubytovani_pol;
DROP TABLE       ubytovani_bod;
```

3.1 TVORBA TEMATICKÝCH VRSTEV

Vrstvy z OpenStreetMap

- U vrstvy vleky bylo třeba vymazat vleky mimo ČR

```
DELETE FROM      vleky
WHERE            vleky.geom NOT IN (
                SELECT      vleky.geom
                FROM        vleky
                JOIN        kraje
                ON          ST_Within(vleky.geom,kraje.geom));
```

- Byla dopočítána délka a odstraněny vleky s moc malou délkou

```
ALTER TABLE vleky
ADD delka float;
UPDATE vleky
SET delka = ST_LENGTH(geom);
DELETE FROM vleky
WHERE delka < 50;
```

- A bylo třeba přeložit typy vleků do češtiny. Příklad:

```
UPDATE vleky SET typ='lanovka' WHERE typ='cable_car';
UPDATE vleky SET typ='sedackova_lanovka' WHERE typ='chair_lift';
```

3.2 TVORBA TEMATICKÝCH VRSTEV

Vrstvy ze schématu GIS1

- Ze schématu GIS1 byly vytvořeny tyto vrstvy:
 - Obce (Bodová vrstva)
 - Vlaky (Bodová vrstva)
 - Železnice (Liniová vrstva)
 - Kraje (Polygonová vrstva)
 - Obce_pol (Polygonová vrstva)
- Tvorba vrstev probíhala téměř totožně jako vrstev z OSM
- Transformace dat ze systému S-JTSK do Google Mercator

```
ALTER TABLE obce
RENAME COLUMN geom
TO geom_beta;
SELECT AddGeometryColumn('obce', 'geom', 900913, 'point', 2);
UPDATE obce
SET geom = ST_Transform(geom_beta, 900913);
SELECT DropGeometryColumn('obce','geom_beta');
```

3.2 TVORBA TEMATICKÝCH VRSTEV

Vrstvy ze schématu GIS1

- Tvorba vrstvy kraje a její převod na multipolygony

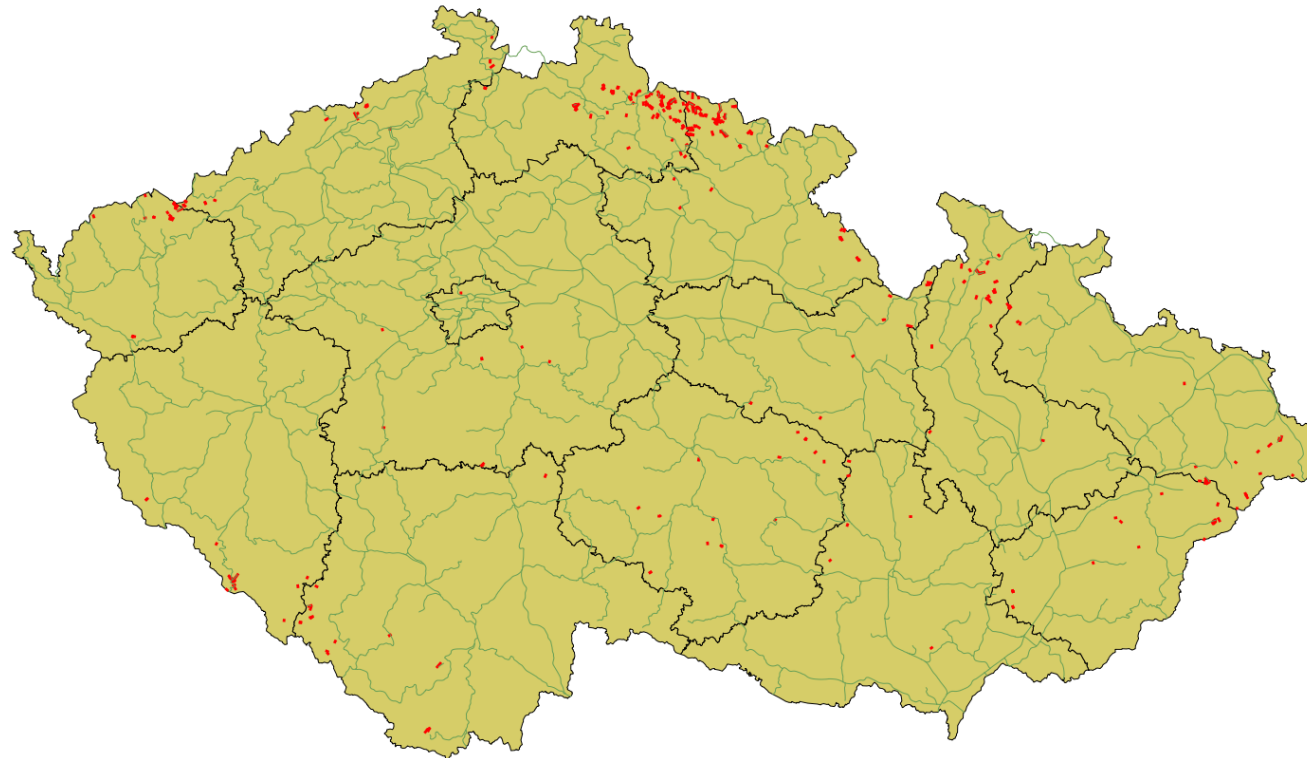
```
CREATE TABLE kraje (ogc_fid SERIAL PRIMARY KEY, nazev varchar(250), geom geometry);
INSERT INTO kraje (geom,nazev)
SELECT
FROM
GROUP BY
CREATE TABLE kraje_beta
AS SELECT
FROM kraje
GROUP BY nazev;
CREATE TABLE kraje_gama
AS SELECT *
FROM kraje;
DELETE FROM kraje_gama
WHERE
NOT IN
(SELECT
SELECT
UPDATE
SET
FROM kraje_beta AS b WHERE a.nazev = b.nazev;
```

```

(SELECT MAX(ctid) FROM kraje GROUP BY nazev);
DropGeometryColumn('kraje_gama', 'geom');
AddGeometryColumn('kraje_gama', 'geom', 2065, 'MultiPolygon', 2);
kraje_gama AS a
geom = ST_Multi(b.geom)
kraje_beta AS b WHERE a.nazev = b.nazev;
```

3 TVORBA TEMATICKÝCH VRSTEV

- ScreenShot vrstev kraje, vleky, železnice

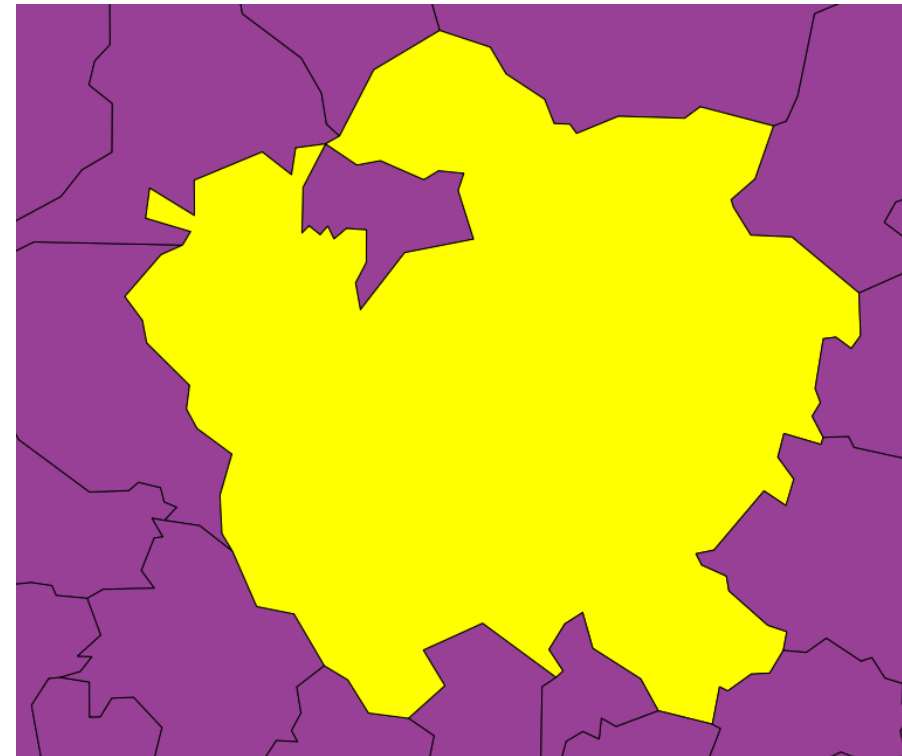


4 VALIDACE DAT

- Pro provádění prostorových dotazů je třeba zajistit, aby byla geometrie všech vrstev validní
- Ke zjištění validity dat slouží tento dotaz:

```
SELECT osm_id, ST_IsValidReason(geom)
FROM vleky
WHERE NOT ST_IsValid(geom);
```
- Funkce ST_IsValid vrací TRUE, pokud je geometrie v pořádku. Pokud je chybná, vypíše důvod.
- Jediný problém – vrstva obce_pol – Jeden *ring self-intersection*
- Tento problém vyřešila aplikace „nulového bufferu“ na tuto vrstvu

```
UPDATE obce_pol
SET geom=ST_Buffer(geom, 0.0);
```
- Kontrola, zda existuje sloupec geometrie a zda má správné prostorové omezení funkcí *populate_geometry_columns*



5 TVORBA TOPOLOGIE

- Vytvoření nové topologie:

```
SELECT CreateTopology ('topo_vleky');
```

- Přidání sloupečku s topologií do vrstvy (v tomto případě vleky)

```
SELECT AddTopoGeometryColumn ('topo_vleky', 'd13', 'vleky', 'topo', 'POLYLINE');
```

- Automatická tvorba topologie funkcí toTopoGeom

```
UPDATE vleky SET topo = toTopoGeom (geom, 'topo_vleky', 1);
```

- Kontrola topologie (u všech vrstev proběhla v pořádku)

```
SELECT TopologySummary ('topo_vleky');
```

```
SELECT ValidateTopology ('topo_vleky');
```


6 PostGIS Raster

- Hlavní využití v našem projektu
 - Přidání výškových informací vlekům
 - Pár prostorových dotazů s využitím PostGIS Raster
- Pomocí funkcí *ST_StartPoint* a *ST_EndPoint* byly přidány počáteční a koncové body všem vlekům
- Výpočet výšek počátečních a koncových bodů funkcí *St_Value*
- Výpočet dalších informací
 - Převýšení (prevyseni = ABS(sp_vyska - ep_vyska);)
 - Sklon (sklon = (prevyseni/delka)*100;)
- Výšky pouze orientační – rozlišení rastru je 60x90 m

7 pgRouting

- S pgRouting se pracovalo na vrstvě železnice
- Nutnost přidání do tabulky železnice počáteční a koncové body linií a jejich indexy
- Po konzultaci na stránce <http://antiagraser.com>, vyřešeno takto:

```
CREATE VIEW zeleznice_ext AS
SELECT *, st_startpoint(geom), st_endpoint(geom)
FROM zeleznice;
CREATE TABLE node
AS SELECT row_number() OVER (ORDER BY foo.p)::integer AS id, foo.p AS geom
FROM (
    SELECT DISTINCT st_startpoint AS p FROM zeleznice_ext
    UNION
    SELECT DISTINCT st_endpoint AS p FROM zeleznice_ext
) foo
GROUP BY foo.p;
CREATE TABLE zeleznice_network
AS SELECT a.*, b.id as start_id, c.id as end_id
FROM zeleznice_ext AS a
JOIN node AS b
ON a.st_startpoint = b.geom
JOIN node AS c
ON a.st_endpoint = c.geom;
```

8.1 DOTAZY

Atributové

- Kolik je na území ČR sjezdovek delších než 2km?

```
SELECT    delka, typ
FROM      vleky
WHERE     delka > 2000;
```

Odpověď: 22

- Kolik ubytovacích zařízení z celkového počtu má v databázi uveden název?

```
SELECT    count(*)
FROM      ubytovani
WHERE     nazev IS NOT NULL;
SELECT    count(*)
FROM      ubytovani;
```

Odpověď: 2297/2477 (93%)

8.2 DOTAZY

Prostorové

- Které ubytovací zařízení leží nejbližně nějaké sjezdovce? A ve které obci leží?

```
SELECT    ROUND(ST_Distance(u.geom, v.geom)) AS vzdalenost, u.nazev, u.typ
FROM      ubytovani AS u
JOIN      vleky AS v
ON        ST_Disjoint(u.geom, v.geom)
ORDER BY  vzdalenost
LIMIT     1;
SELECT    DISTINCT nazev_eng AS obec
FROM      obce_pol
JOIN      ubytovani
ON        ST_Contains(obce_pol.geom, (SELECT ubytovani.geom FROM ubytovani WHERE nazev='Apartmán Novako'));
```

Odpověď: Apartmán Novako, 26m od sjezdovky, obec Boží Dar

8.2 DOTAZY

Prostorové

- Která železniční stanice je nejvhodnější výstupní stanice pro lyžaře? (V jejím 2km okolí leží nejvíce sjezdovek?)

- před optimalizací (ani po 5 minutách dotaz neproběhl):

```
SELECT    nazev, count(nazev) AS pocet_sjezdovek
FROM      vlaky
JOIN      vleky
ON        ST_Intersects(vleky.geom, ST_Buffer(vlaky.geom, 2e3))
GROUP BY  nazev
ORDER BY  pocet_sjezdovek DESC
LIMIT     2;
```

- po optimalizaci (83 ms)

```
SELECT    nazev, count(nazev) AS pocet_sjezdovek
FROM      vlaky
JOIN      vleky
ON        ST_Expand(vlaky.geom, 2e3) && vleky.geom
AND       ST_Intersects(vleky.geom, ST_Buffer(vlaky.geom, 2e3))
GROUP BY  nazev
ORDER BY  pocet_sjezdovek DESC
LIMIT     2;
```

Odpověď: Velké Karlovice (11 sjezdovek do 2 km), Velké Karlovice zast. (10 sjezdovek do 2 km)

8.2 DOTAZY

Prostorové

- Kde se ubytovat pokud chceme lyžovat na nejprudší sjezdovce ve vzdálenosti do 150 km od Prahy, která má alespoň 1500m ?

```
SELECT    osm_id
FROM      vleky
JOIN      obce
ON        ST_Expand(obce.geom, 1.5e5) && vleky.geom
AND       nazob_eng = 'Praha'
AND       ST_Intersects(vleky.geom, ST_Buffer(obce.geom, 1.5e5))
AND       delka > 1500 [22]
ORDER BY  sklon DESC
LIMIT     1; --96934872

SELECT    nazev, o.typ, St_Distance(o.geom, v.geom) AS vzdalenost
FROM      ubytovani AS o
JOIN      vleky AS v
ON        v.osm_id = 96934872 AND nazev LIKE '%'
ORDER BY  vzdalenost
LIMIT     1; --hotel Florian

SELECT    o.nazev_eng
FROM      obce_pol AS o
JOIN      ubytovani AS u
ON        ST_Contains(u.geom, o.geom)
AND       u.nazev = 'Florián';
```

Odpověď: hotel Ještěd, obec Svetla pod Jestedem (ski areál Jested)

8.2 DOTAZY

Prostorové

- Jaká je výměra území, které je ve vzdálenosti 2000m od jakékoliv sjezdovky a nachází se v horách (nad 1000 mnm)?

```
CREATE TABLE vleky_buff
AS SELECT
FROM
CREATE TABLE dmt1000
AS SELECT
FROM
CREATE TABLE dmt_vleky_buff
AS SELECT
FROM
CROSS JOIN
CREATE TABLE dmt_vleky_geom
AS SELECT
FROM
(
    SELECT geomval AS gv
    FROM dmt_vleky_buff
) AS foo;
SELECT
FROM
WHERE SUM(ST_Area(geom))
dmt_vleky_geom
val = 1;
```

Odpověď: 28134 ha

8.2 DOTAZY

Prostorové

- Jakou železniční cestou jet, pokud se chci dostat k nejbližší sjezdovce z Olomouce, která je alespoň 2500m dlouhá?

```
SELECT    osm_id, ST_Distance(v.geom, o.geom) AS vzd
FROM      vleky AS v
JOIN      obce AS o
ON        o.nazob_eng = 'Olomouc'
AND       delka > 2500
ORDER BY  vzd; -- (91641416)

SELECT    nazev, ST_Distance(s.geom, v.geom) AS vzd
FROM      vlaky AS s
JOIN      vleky AS v
ON        v.osm_id = 91641416
ORDER BY  vzd
LIMIT     1; -- (Kouty nad Desnou)

SELECT    z.source, ST_Distance(z.the_geom, v.geom) AS vzd
FROM      zeleznice_network AS z
JOIN      vlaky AS v
ON        v.nazev = 'Kouty nad Desnou'
ORDER BY  vzd
LIMIT     1; -- target id = 594

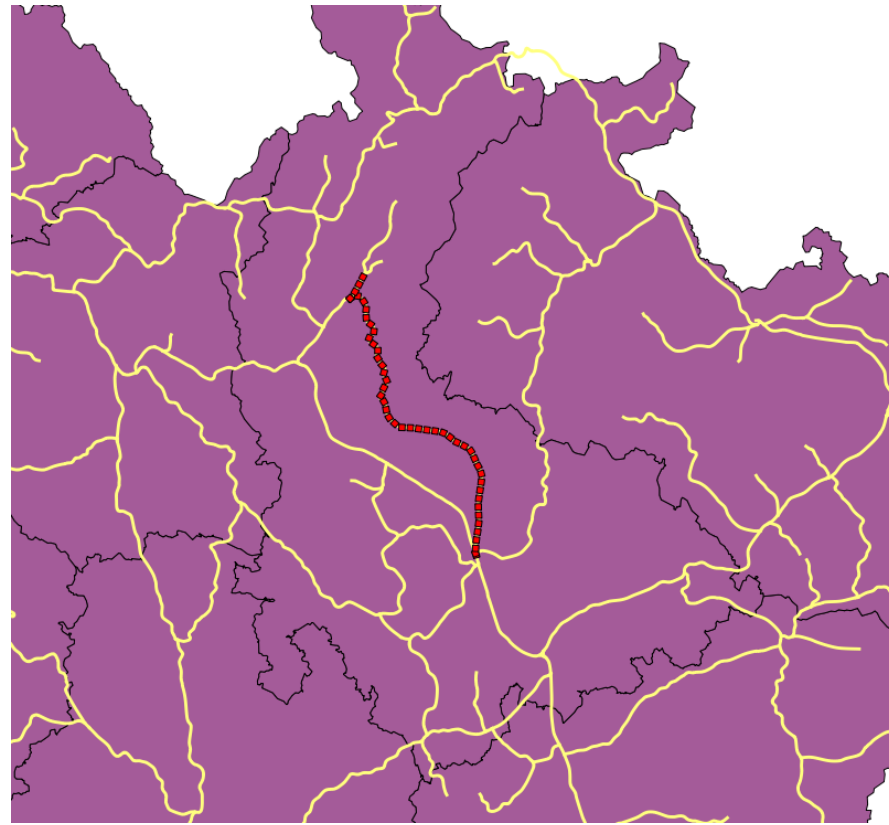
SELECT    z.source, nazev, ST_Distance(z.the_geom, v.geom) AS vzd
FROM      zeleznice_network AS z
JOIN      vlaky AS v
ON        v.nazev LIKE 'Olom%'
ORDER BY  vzd
LIMIT     1; -- source id: 635
```


8.2 DOTAZY

Prostorové

- Pokračování

```
CREATE TABLE olomouc_kouty  
AS SELECT gid, the_geom  
FROM dijkstra_sp('zeleznice_network', 635, 594);
```



8.2 DOTAZY

Prostorové

- Kolik ubytovacích zařízení se nachází do 2 km od železniční trati Praha - Olomouc? A které je nejbližší kolejím?

```
CREATE TABLE olomouc_praha
AS SELECT gid, the_geom
FROM dijkstra_sp('zeleznice_network', 635, 258);
CREATE TABLE trat_buff
AS SELECT ST_Union(ST_Buffer(the_geom, 2000)) AS geom
FROM olomouc_praha;
SELECT count(nazev)
FROM ubytovani as u
JOIN trat_buff as t
ON ST_Intersects(u.geom, t.geom);
SELECT nazev, typ, ST_Distance(u.geom, z.geom) AS vzd
FROM ubytovani as u
JOIN trat_buff as t
ON ST_Intersects(u.geom, t.geom)
JOIN zeleznice as z
ON ST_Distance(z.geom, u.geom) > 0
ORDER BY vzd
LIMIT 1;
SELECT nazev_eng
FROM obce_pol AS o
JOIN ubytovani AS u
ON ST_Within(u.geom, o.geom)
WHERE u.nazev = 'Poprad,
```

Odpověď: 113, nejbližší je hotel Poprad, Ústí nad Orlicí

Děkujeme za pozornost.

TĚŠÍME SE NA VAŠE DOTAZY.