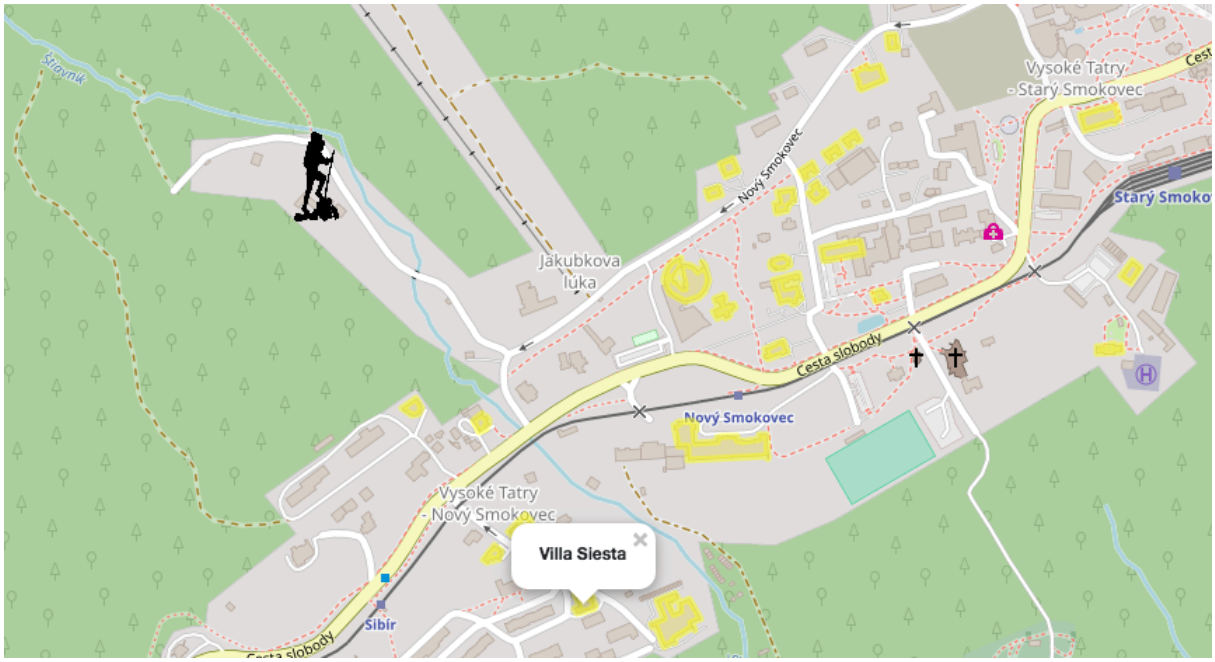
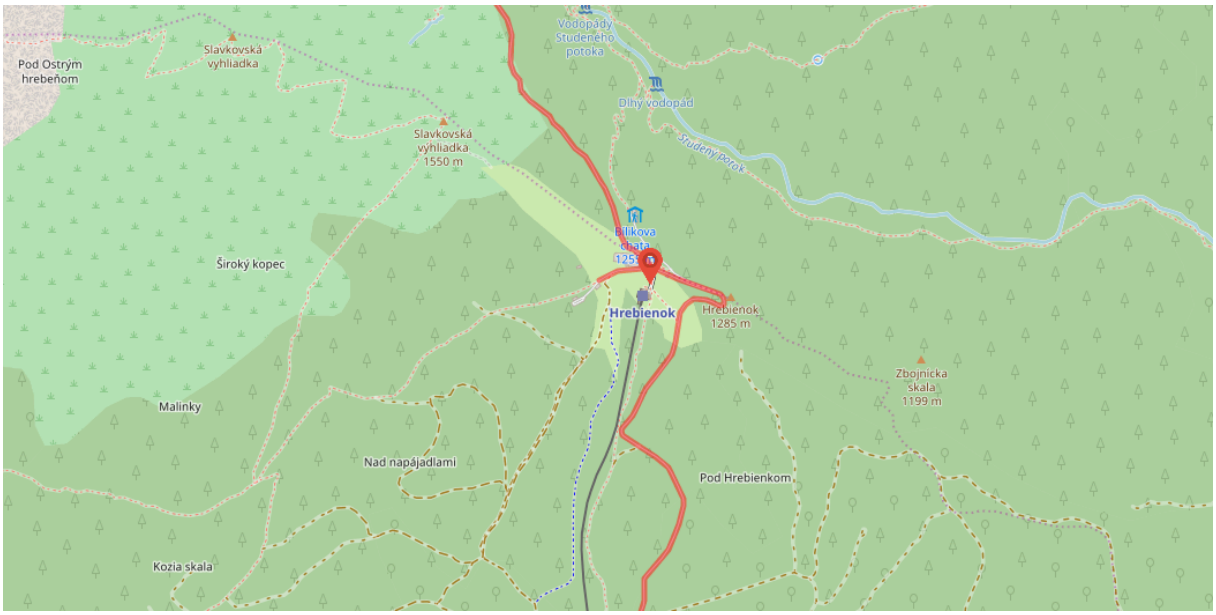
# Scenáre

1. Scenár - Zobrazenie najbližších artefaktov z aktuálnej pozície turistu. Artefakty si turista môže vybrať v selecte a vybrať si môže aj vzdialenosť, do ktorej sa artefakty zobrazia.


2. Scenár - Nájdenie niekoľko najbližších ubytovaní od aktuálnej pozície turistu. Tento počet si zvolí turista sám.




3. Zobrazenie cyklistických trás a zastávok, v ktorých sa pretínajú.



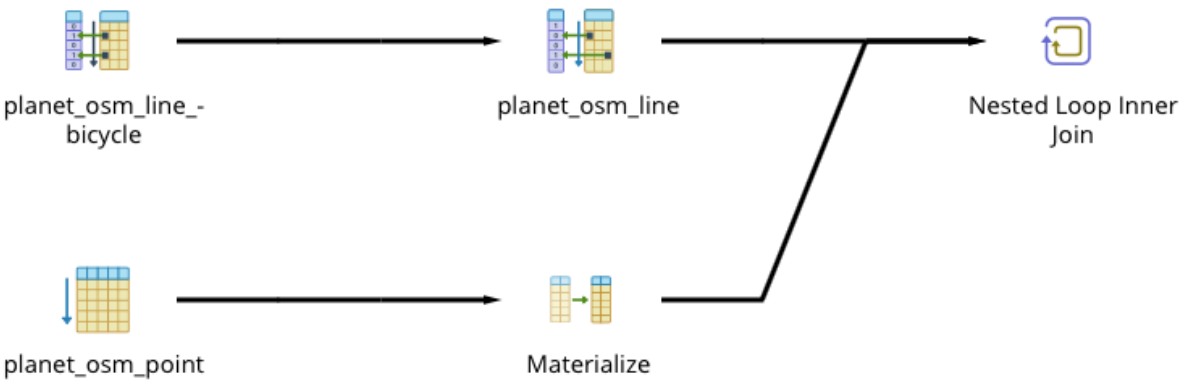
4. Zobrazenie plôch s artefaktami, ktoré sa tam nachádzajú, ak sa tam nejaké nachádzajú


# Query

- vytváranie indexov

```
CREATE INDEX planet_osm_line_way ON planet_osm_line USING gist (way)
CREATE INDEX planet_osm_line_bicycle ON planet_osm_line (bicycle)

CREATE INDEX planet_osm_point_way ON planet_osm_point USING gist (way)
CREATE INDEX planet_osm_point_amenity ON planet_osm_point (amenity)
```



- 1.scenár

```
SELECT body.name, ST_AsText(body.cesta)
FROM(
    SELECT name, ST_Transform (way, 4326) as cesta
    FROM planet_osm_point
    WHERE amenity='parking') as body
WHERE ST_Distance_Sphere(ST_setSRID(ST_MakePoint(20.2096027, 49.1390668), 4326), body.cesta) < 10000
```

- 2.scenár

```
SELECT polygons.coordinaty, polygons.cesta, name FROM (
        SELECT ST_AsText (ST_Transform (way, 4326)) AS coordinaty, name, tourism,
                ST_Distance_Sphere(ST_setSRID(ST_MakePoint(20.2096027, 49.1390668), 4326),
                ST_Centroid(ST_setSRID(way, 4326)))  AS cesta
FROM planet_osm_polygon
WHERE tourism!='null') AS polygons
ORDER BY polygons.cesta
LIMIT 10
```

- 3.scenár

```
SELECT points.name, lines.name, ST_Intersects(ST_Transform(points.way, 4326), ST_Transform(lines.way, 4326)),
       ST_AsText(ST_Transform(points.way, 4326)), ST_AsText(ST_Transform(lines.way, 4326))
FROM planet_osm_line AS lines
INNER JOIN planet_osm_point AS points ON (lines.bicycle = points.bicycle)
WHERE lines.bicycle='yes' and ST_Intersects(ST_Transform(points.way, 4326), ST_Transform(lines.way, 4326))='true'
```

- 4.scenár

```
SELECT polygon.name, ST_AsText (ST_Transform (polygon.way, 4326)), points.name,
        ST_AsText (ST_Transform (points.way, 4326)), ST_Contains(polygon.way, points.way)
FROM planet_osm_polygon AS polygon
INNER JOIN planet_osm_point AS points ON (polygon.amenity = points.amenity)
WHERE ST_Contains(polygon.way, points.way)='true'
```