M2TIW-INT SIG tp noté

Antennes au Colorado et leurs couvertures

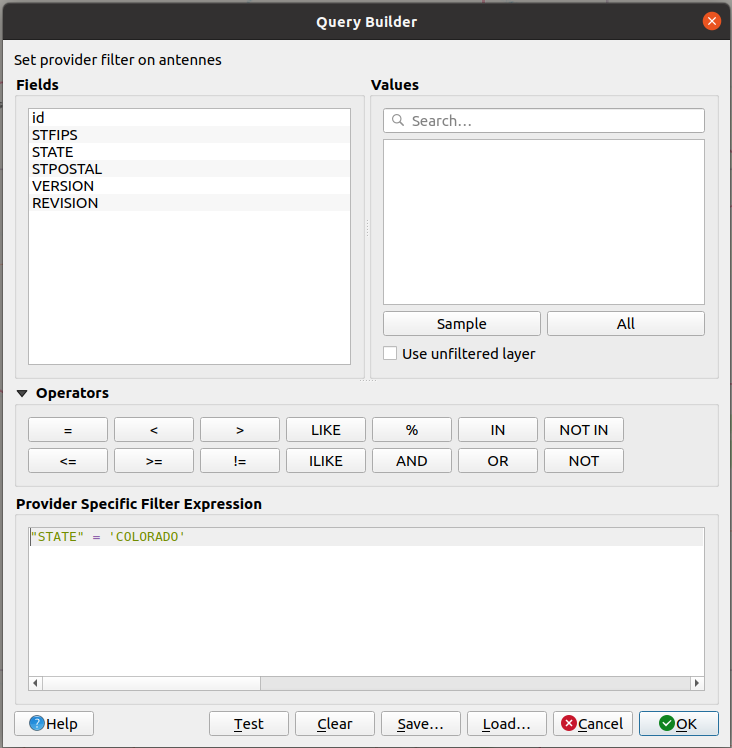
Étudiant1 -> Nom : MOLINARES VALENCIA Prénom : Diogenes Numéro : p2019196

Étudiant2 -> Nom : BUNEL Prénom : Maxime Numéro : p1914012

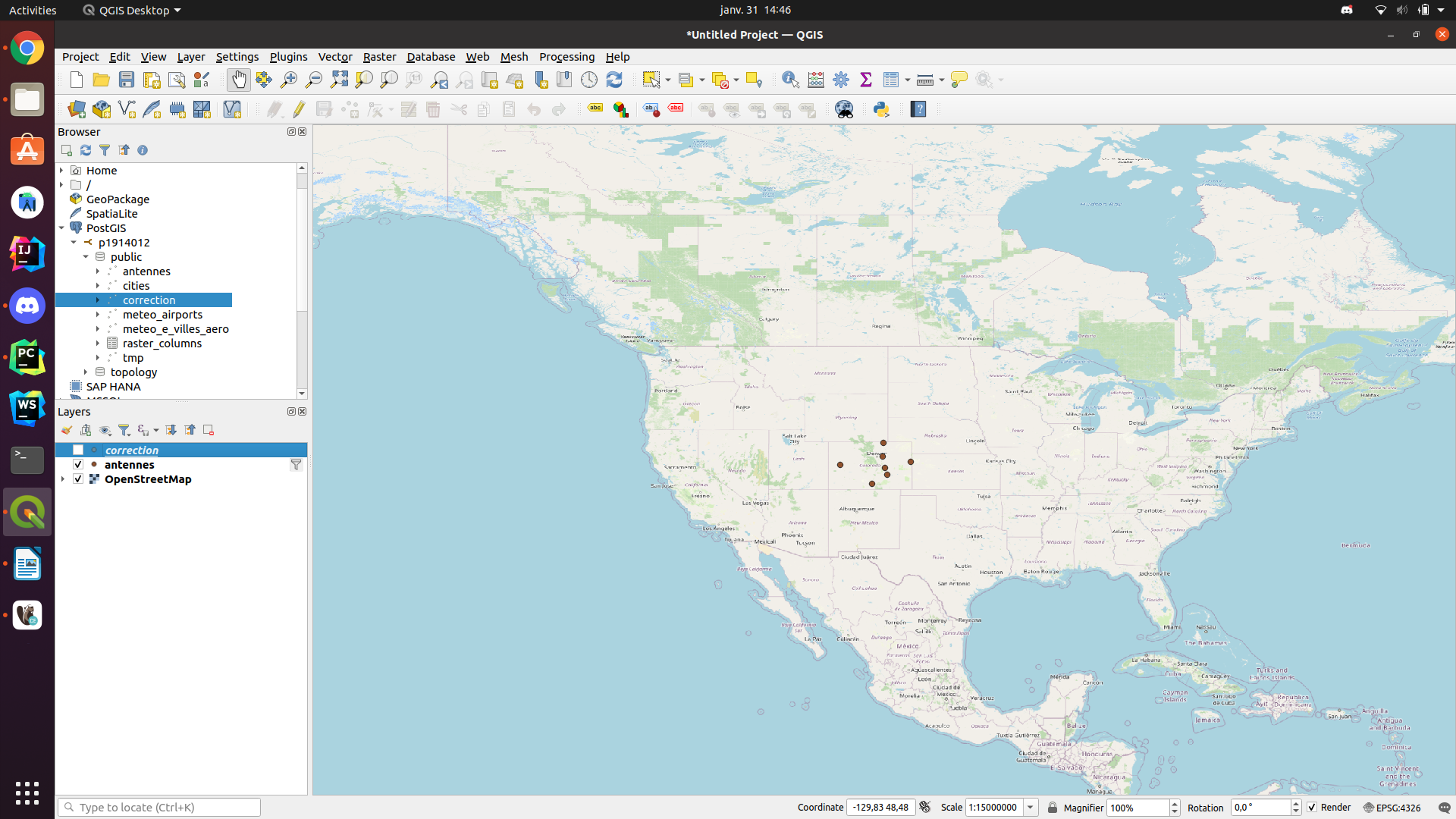
Étudiant3 -> Nom : GIRAUD Prénom : Julien Numéro : p1704709

# a) Filtrage sous QGis

## a.1) Copie d’écran IMAGE "a\_filtrage\_commande" : 1 copie d'écran de QGis montrant la commande QGis qui a permis de filtrer les antennes

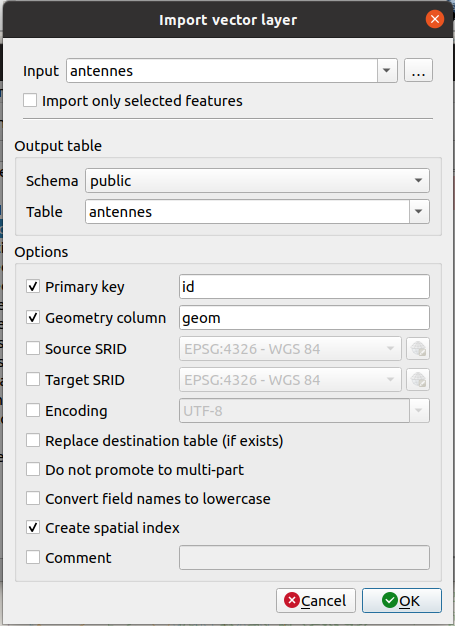


## a.2) Copie d’écran IMAGE "a\_filtrage\_resultat" : 1 copie d'écran de QGis montrant la couche filtrée avec en fond de carte la mappemonde OpenStreetMap centrée sur les USA à l'échelle 1/15 000 000

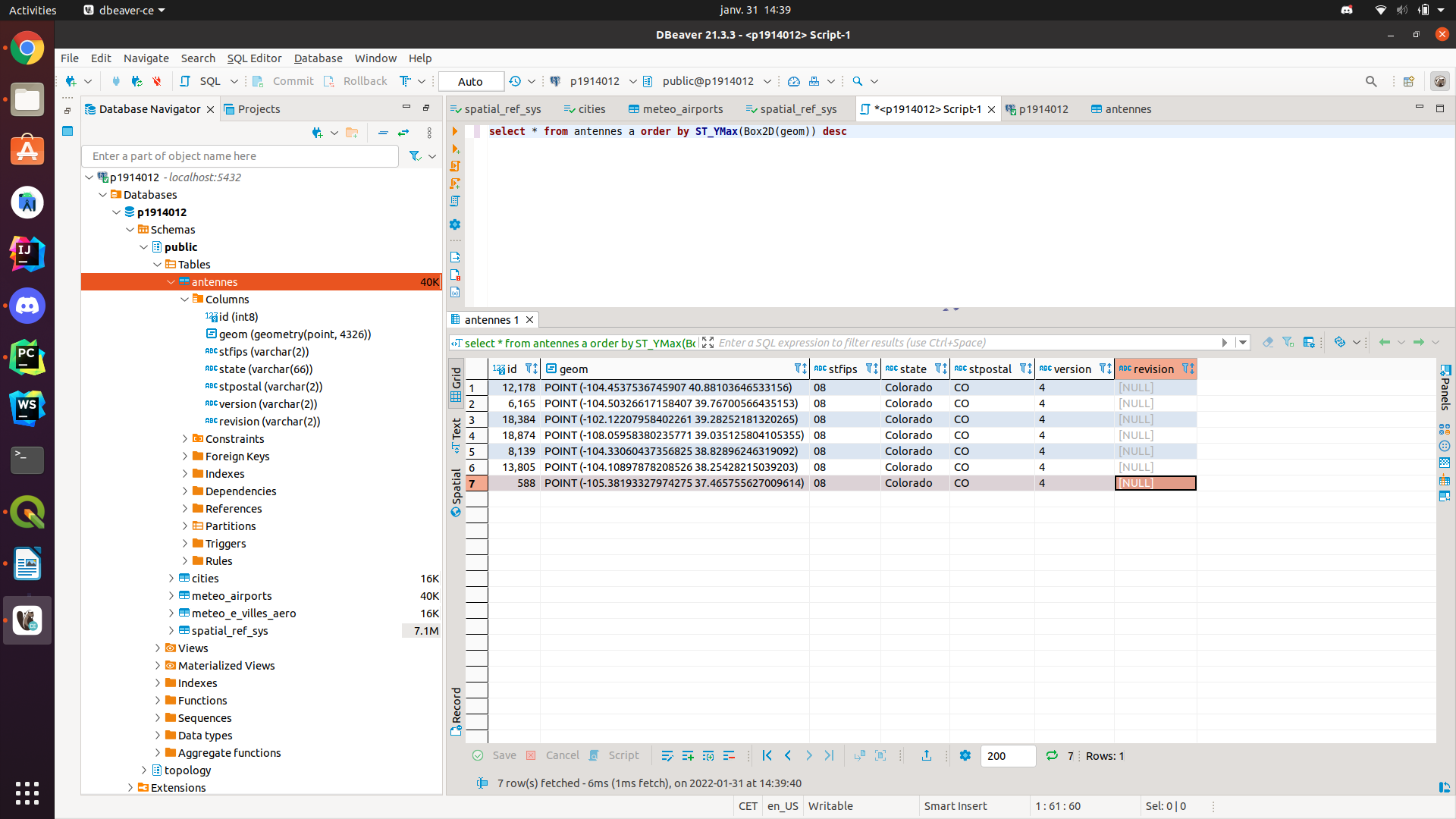


# b) Export QGis vers PostGis

## b.1) Copie d’écran IMAGE "b\_export\_qgis\_commande" : 1 copie d'écran de QGis montrant la commande QGis qui a permis d'exporter la table



## b.2) Copie d’écran IMAGE "b\_export\_postgis\_table" : 1 copie d'écran de PostGis (par exemple pgAdmin ou DBeaver) de la requête qui affiche tous les tuples de la table "antennes" avec son résultat



# c) Correction sous PostGis

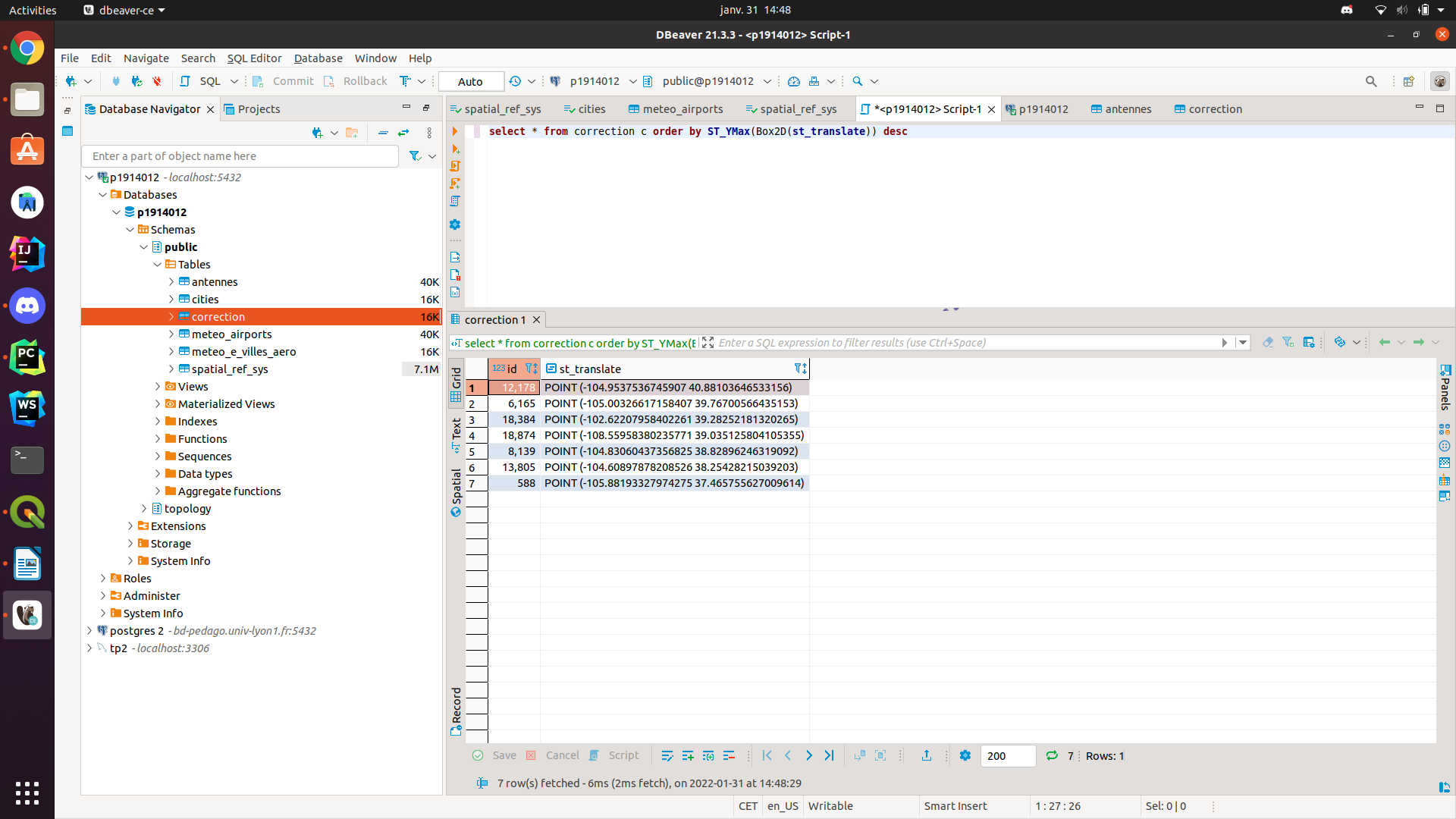
## c.1) TEXTE SQL "c\_correction.sql" : 1 fichier qui contient : la requête SQL qui crée la table "correction" + la requête qui affiche tous les tuples de la table "correction"

-- REQUETES SQL ...

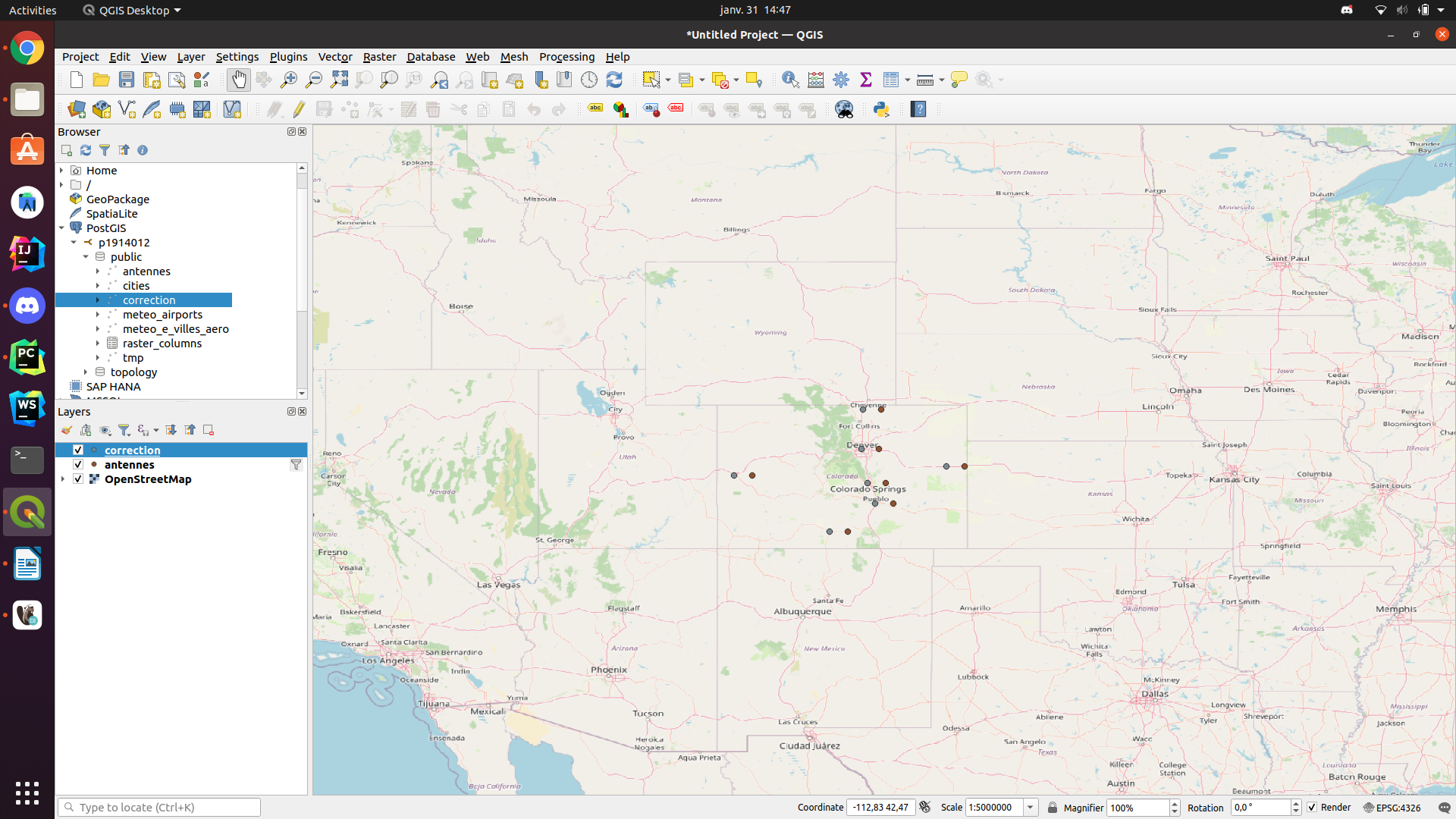
create table correction as (select id, st\_translate(geom,-0.5,0) from antennes a);

select \* from correction c order by ST\_YMax(Box2D(st\_translate)) desc

## c.2) Copie d’écran IMAGE "c\_correction\_postgis\_resultat" : 1 copie d'écran de PostGis de la requête qui affiche tous les tuples de la table "correction" avec son résultat



## c.3) Copie d’écran IMAGE "c\_correction\_qgis\_resultat" : 1 copie d'écran de QGis qui affiche les couches superposées : antennes, antennes corrigées, fond OSM



# d) Couvertures des antennes en Python

## d.1) TEXTE "d\_recuperation\_calcul.txt" : 1 fichier dans lequel est écrite la formule et le résultat du calcul de 50km sur l'équateur en degrés

La formule est ...

L = π \* R \* a / 180

- "L" la distance de l'arc de cercle entre deux points (en km),

- "R" le rayon du cercle (en km),

- "a" l'angle (en dégrées).

Avec R = 6378.1 km

L \* 180 / π \* R = a

6378.1 \* 180 / (π \* 50) = 0.449 °

## d.2) TEXTE CODE "d\_recuperation.py" : 1 fichier nommé "recuperation.py" qui contient le code Python

# Code Python …

import pg8000.native

from shapely import wkb

con = pg8000.native.Connection("p1914012", password="nBdh3CzPeZYp", database="p1914012")

table = con.run("SELECT \* FROM correction")

con.run("ALTER TABLE correction ADD COLUMN IF NOT EXISTS circle geometry")

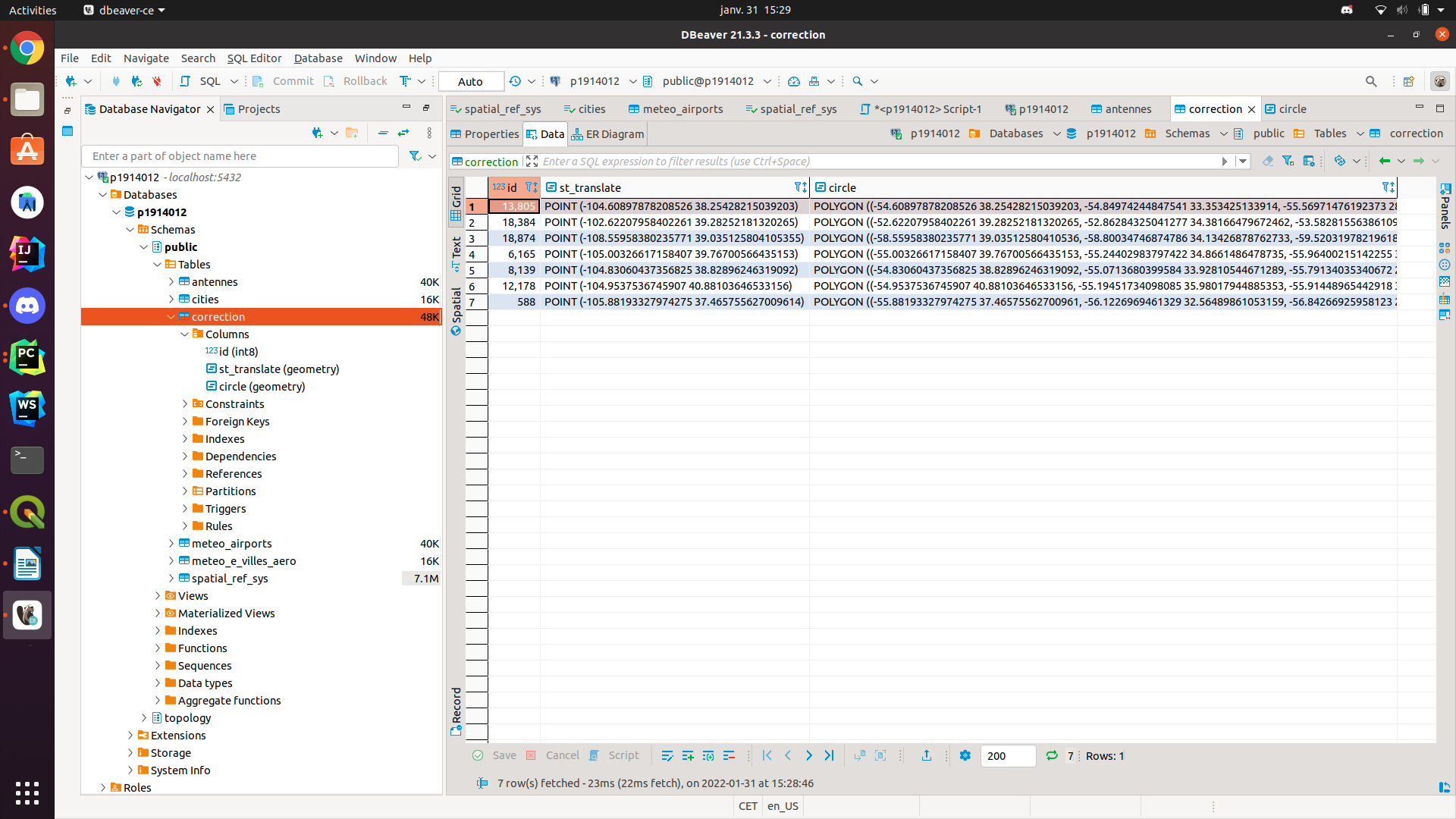
for line in table:

line[1] = wkb.loads(bytes.fromhex(line[1]))

circle = line[1].buffer(50.0)

con.run("UPDATE correction SET circle = :circle WHERE id=:id", circle=circle, id=line[0])

## d.3) Copie d’écran IMAGE "d\_recuperation\_console\_resultat" : 1 copie d'écran de la console où le script liste les infos des antennes



# e) Serveur WFS en Python

## e.1) TEXTE CODE "e\_wfs.py" : 1 fichier qui contient le code Python

# Code Python …

# Ecrivez votre code Python ici ...

# Partie récupération des données

import geojson

import requests

import pg8000.native

from shapely import wkb

import simplekml

import json

antennes = []

con = pg8000.native.Connection("p1914012", password="nBdh3CzPeZYp", database="p1914012")

table = con.run("SELECT \* FROM correction")

for line in table:

# line[5] = wkb.loads(bytes.fromhex(line[5]))

antennes.append([line[0], [wkb.loads(bytes.fromhex(line[1])).y,wkb.loads(bytes.fromhex(line[1])).x], line[2]])

print(antennes)

# Partie serveur Flask

from flask import Flask

from flask\_cors import CORS

app = Flask(name)

CORS(app)

# Sur terminal

# export FLASK\_APP=main\_2\_2

# flask run --port 8010

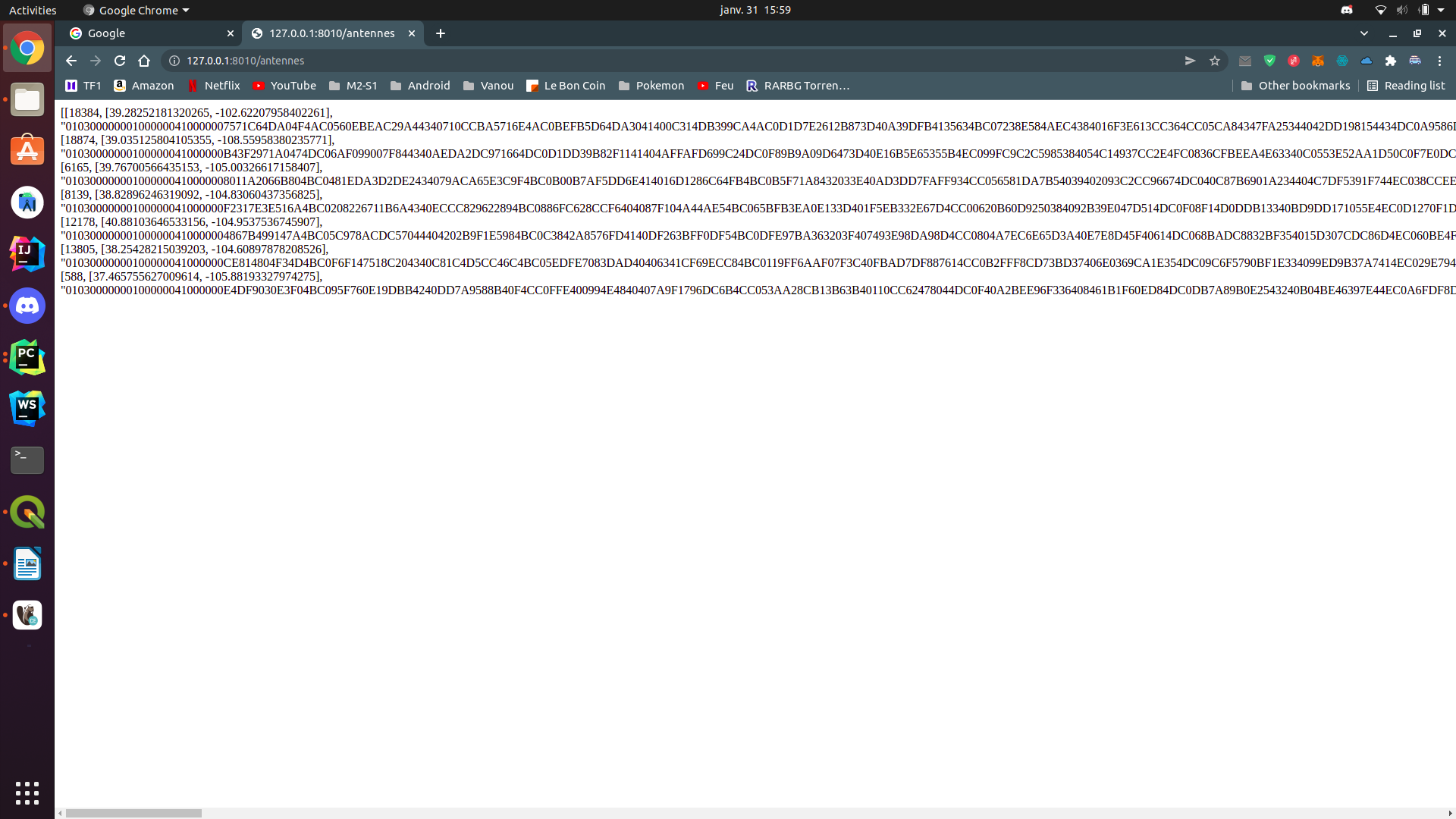
@app.route("/antennes")

def get\_antennes():

return json.dumps(antennes)

## e.2) Copie d’écran IMAGE "e\_wfs\_geojson\_proprietes" : 1 copie d'écran du navigateur affichant les antennes en Geojson filtrées -> développez toutes les propriétés et réduisez toutes les géométries pour qu'on puisse voir toutes les antennes retournées, et qu'on voie l'url

## e.3) Copie d’écran IMAGE "e\_wfs\_geojson\_geometrie" : 1 copie d'écran du navigateur affichant les antennes en Geojson filtrées -> développez la géométrie d'une antenne pour montrer les coordonnées de quelques points



# f) Carte en ligne en JS

## f.1) TEXTE CODE "f\_carteweb.html" : 1 fichier qui contient le code html et javascript

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"

integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A=="

crossorigin=""/>

<!-- Make sure you put this AFTER Leaflet's CSS -->

<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"

integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA=="

crossorigin=""></script>

</head>

<body>

<div id="map" style="height: 100vh"></div>

<script>

(async () => {

var map = L.map('map').fitBounds([[37.0,-105.6], [41.0,-104.7]]);

var tiles = L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access\_token=pk.eyJ1IjoibWFwYm94IiwiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ3N3gifQ.rJcFIG214AriISLbB6B5aw', {

maxZoom: 7,

attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors, ' +

'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',

id: 'mapbox/streets-v11',

tileSize: 512,

zoomOffset: -1

}).addTo(map);

var response = await fetch("http://localhost:8010/antennes");

var contenu = await response.json();

contenu.forEach(c => {

// add polygon

L.marker(c[1]).addTo(map);

// add point

// L.polygon(c[2], {color: 'red'}).addTo(map);

})

}

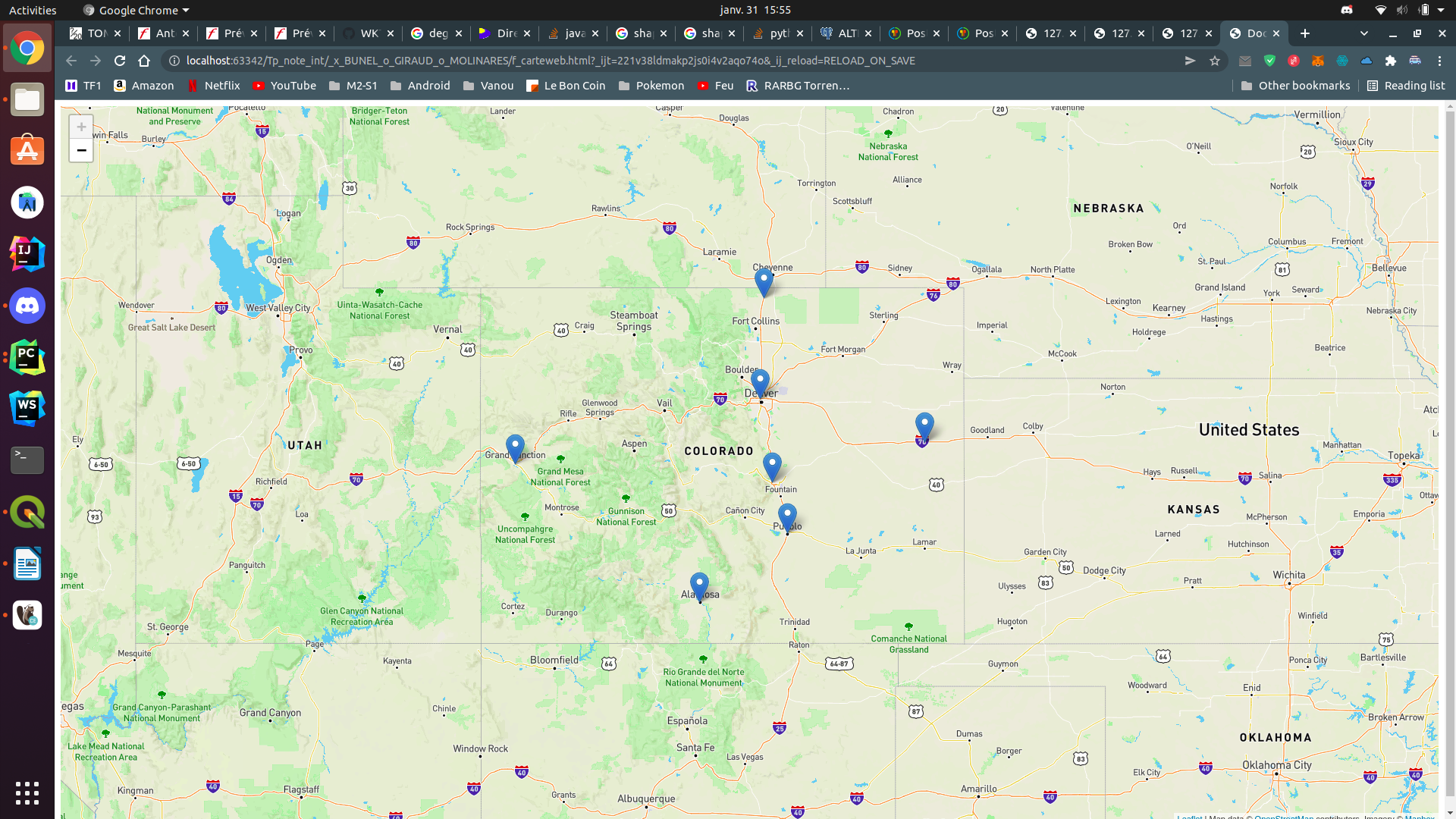
)();

</script>

</body>

</html>

## f.2) Copie d’écran IMAGE "f\_carteweb" : 1 copie d'écran du navigateur affichant les antennes dans le rectangle englobant sur le fond de carte OpenStreetMap



## f.3) TEXTE "f\_carteweb.txt" : 1 fichier texte qui contient les réponses aux questions

Réponses : ...