Tabellen:

bln_nodes	point	Kreuzungspunkte im Berliner Straßenenenetz
bln_edges	line	Segmente im Berliner Straßenenenetz
gis_osm_*layer*_free_1	line/point	
gis_osm_*layer*_a_free_1	polygon	

wobei *layer* folgende Werte annimmt:

- places
- pois
- pofw
- natural
- traffic
- transport
- roads
- railways
- waterway
- buildings
- landuse
- water

fclass	code	count
town	1002	2
village	1003	5
hamlet	1004	20
national_capital	1005	1
suburb	1010	97
island	1020	1
locality	1050	50

fclass	name
suburb	Adlershof
suburb	Altglienicke
suburb	Alt-Hohenschönhausen
suburb	Alt-Treptow
suburb	Baumschulenweg
suburb	Biesdorf
suburb	Blankenburg
suburb	Blankenfelde
suburb	Bohnsdorf
suburb	Borsigwalde
suburb	Britz
suburb	Buch
suburb	Buckow
suburb	Charlottenburg
suburb	Charlottenburg-Nord
suburb	Dahlem

fclass	code	count
police	2001	19
fire_station	2002	33
post_box	2004	2436
post_office	2005	210
telephone	2006	824
library	2007	102
town_hall	2008	13
courthouse	2009	4
prison	2010	1
embassy	2011	107
community_centre	2012	282
nursing_home	2013	4
arts_centre	2014	77
graveyard	2015	6
market_place	2016	76
recycling	2030	128
recycling_glass	2031	1174
recycling_paper	2032	21
recycling_clothes	2033	672
university	2081	33
school	2082	137
kindergarten	2083	1229
college	2084	51
pharmacy	2101	720
hospital	2110	3
doctors	2120	1096
dentist	2121	576
veterinary	2129	145
theatre	2201	127
nightclub	2202	135
cinema	2203	72
park	2204	2
playground	2205	1047
dog_park	2206	3
sports_centre	2251	149
pitch	2252	1103
swimming_pool	2253	8
golf_course	2255	1
stadium	2256	1
track	2258	3
restaurant	2301	4204
fast_food	2302	2129
cafe	2303	2276
pub	2304	1009
bar	2305	796
food_court	2306	3
biergarten	2307	58
hotel	2401	468
motel	2402	5
guesthouse	2404	106
hostel	2405	81
chalet	2406	4
shelter	2421	231
camp_site	2422	2
caravan_site	2424	3
supermarket	2501	864
bakery	2502	1307
kiosk	2503	537
mall	2504	6
department_store	2505	58
general	2510	8
convenience	2511	846
clothes	2512	1487
florist	2513	591
chemist	2514	229
bookshop	2515	307
butcher	2516	123
shoe_shop	2517	297
beverages	2518	307
optician	2519	325
jeweller	2520	316
gift_shop	2521	142
sports_shop	2522	100
stationery	2523	164
outdoor_shop	2524	26
mobile_phone_shop	2525	264
toy_shop	2526	109
newsagent	2527	91
greengrocer	2528	82
beauty_shop	2529	697
video_shop	2530	28
car_dealership	2541	159
bicycle_shop	2542	357
doityourself	2543	98
furniture_shop	2544	222
computer_shop	2546	97
garden_centre	2547	22
hairdresser	2561	1721
car_rental	2563	98
car_wash	2564	55
car_sharing	2565	27
bicycle_rental	2566	474
travel_agent	2567	261
laundry	2568	251
vending_machine	2590	119
vending_cigarette	2591	75
vending_parking	2592	620
vending_any	2593	1077
bank	2601	353
atm	2602	936

tourist_info	2701	2374
attraction	2721	63
museum	2722	123
monument	2723	9
memorial	2724	2366
artwork	2725	1354
ruins	2732	6
archaeological	2733	1
wayside_cross	2734	10
wayside_shrine	2735	12
battlefield	2736	1
picnic_site	2741	90
viewpoint	2742	177
zoo	2743	3
theme_park	2744	2
toilet	2901	547
bench	2902	16869
drinking_water	2903	162
fountain	2904	362
hunting_stand	2905	249
waste_basket	2906	7535
camera_surveillance	2907	2349
tower	2950	32
comms_tower	2951	21
water_tower	2952	3
observation_tower	2953	7
water_well	2962	1933
water_mill	2963	1
water_works	2964	38

fclass	code	count
christian	3100	60
christian_catholic	3102	4
christian_evangelical	3103	3
christian_lutheran	3104	3
christian_methodist	3105	2
christian_protestant	3107	25
jewish	3200	2
muslim	3300	40
muslim_sunni	3301	6
muslim_shia	3302	5
buddhist	3400	12
hindu	3500	3

fclass	code	count
spring	4101	8
peak	4111	70
tree	4121	183544
beach	4141	4

Area

fclass	code	count
cliff	4112	1
beach	4141	121

fclass	code	count
traffic_signals	5201	7134
mini_roundabout	5202	9
stop	5203	253
crossing	5204	14600
motorway_junction	5206	105
turning_circle	5207	1532
speed_camera	5208	46
street_lamp	5209	50845
fuel	5250	154
parking	5260	369
parking_multistorey	5262	58
parking_underground	5263	34
parking_bicycle	5270	9385
slipway	5301	65
marina	5302	125
pier	5303	15
waterfall	5321	4
weir	5332	24

```
fclass         |code|count|
---------------+----+-----+
railway_station|5601|  281|
railway_halt   |5602|  110|
tram_stop      |5603|  859|
bus_stop       |5621| 6353|
taxi           |5641|  373|
helipad        |5655|   14|
ferry_terminal |5661|   61|
```

fclass	code	count
motorway	5111	664
trunk	5112	67
primary	5113	3005
secondary	5114	7963
tertiary	5115	4235
unclassified	5121	880
residential	5122	20380
living_street	5123	1773
pedestrian	5124	573
motorway_link	5131	616
trunk_link	5132	16
primary_link	5133	123
secondary_link	5134	215
tertiary_link	5135	41
service	5141	70665
track	5142	986
track_grade1	5143	499
track_grade2	5144	1172
track_grade3	5145	1377
track_grade4	5146	1399
track_grade5	5147	440
bridleway	5151	195
cycleway	5152	4224
footway	5153	90959
path	5154	11435
steps	5155	7990
unknown	5199	1

fclass	code	count
rail	6101	3837
light_rail	6102	2677
subway	6103	1259
tram	6104	1798
narrow_gauge	6106	29
miniature_railway	6107	36

fclass	code	count
river	8101	105
stream	8102	972
canal	8103	157
drain	8104	954

```
fclass  |code|count |
--------+----+------+
building|1500|513119|
```

fclass	code	count
forest	7201	2198
park	7202	2344
residential	7203	8590
industrial	7204	576
cemetery	7206	295
allotments	7207	2734
meadow	7208	655
commercial	7209	1531
nature_reserve	7210	54
recreation_ground	7211	148
retail	7212	994
military	7213	35
orchard	7215	37
vineyard	7216	7
scrub	7217	3829
grass	7218	9323
heath	7219	31
farmyard	7228	59
farmland	7229	206

fclass	code	count
water	8200	1323
reservoir	8201	50
riverbank	8202	119
wetland	8221	433

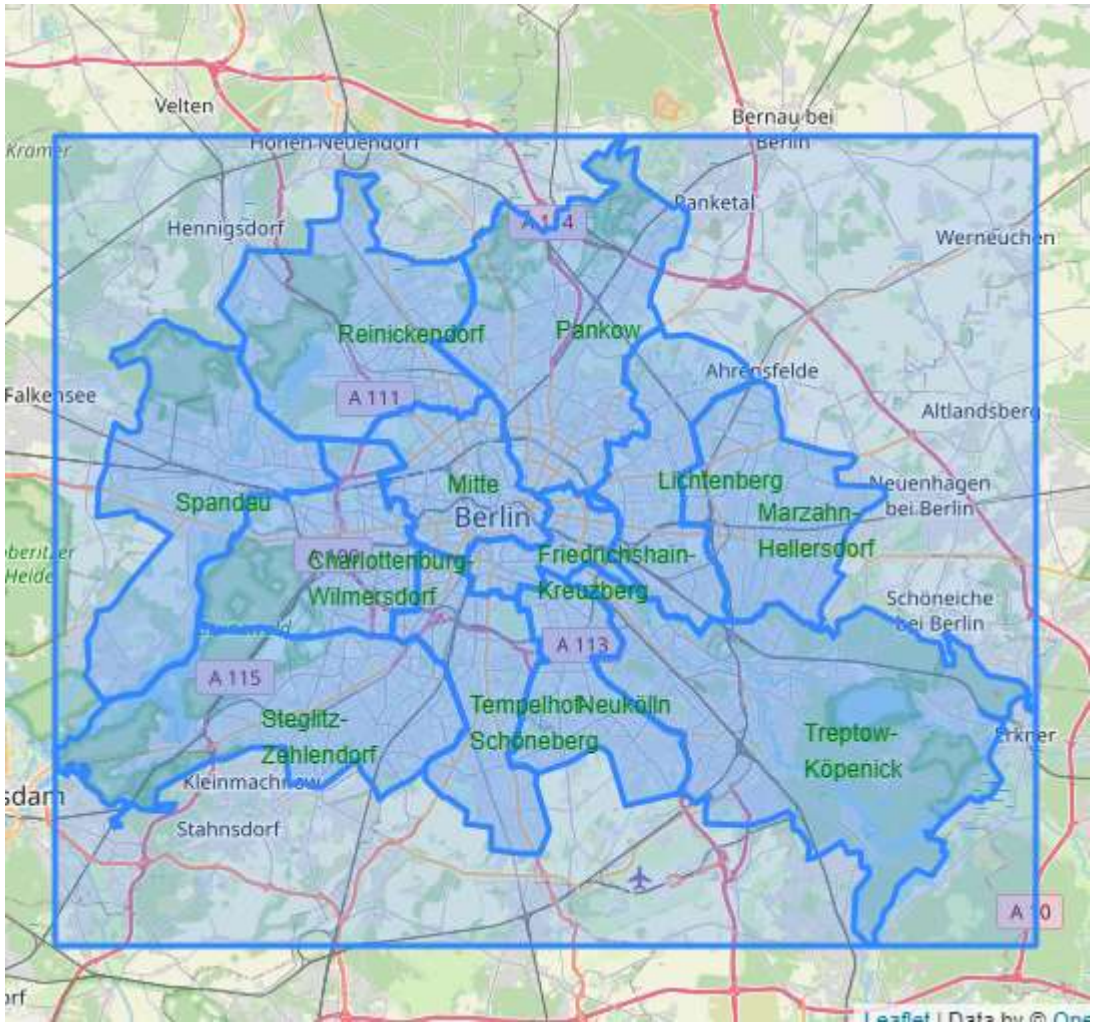
```
import pandas as pd
from sqlalchemy import create_engine
import folium

def lat_lng_from_sql(sql, engine):
    with engine.connect() as con:
        lat_lng_df = pd.read_sql_query(sql, con)
    return lat_lng_df.iat[0,0], lat_lng_df.iat[0,1]
def geojson_from_sql(sql, engine):
    with engine.connect() as con:
        lat_lng_df = pd.read_sql_query(sql, con)
    return lat_lng_df.iat[0,0]

# uuu: userid
# ppp: password
# hhh: host address
url = 'postgresql://uuu:ppp@hhh/adbkt'
engine = create_engine(url, pool_size=1, max_overflow=0, pool_pre_ping=True)
```

```
sql_htw_lat_lng = """
select ST_Y(ST_Centroid(geometry)) as lat, ST_X(ST_Centroid(geometry)) as lng
from uadbkt.gis_osm_pois_a_free_1
where osm_id ='41361350'
order by fclass, name;
"""
htw_lat, htw_lng = lat_lng_from_sql(sql_htw_lat_lng, engine)
htw_lat, htw_lng
map_htw = folium.Map(location=[htw_lat, htw_lng], zoom_start=16)
folium.Marker(
    location=[htw_lat, htw_lng],
    tooltip="HTW"
).add_to(map_htw)
sql_htw_geojson = """
select ST_AsGeoJSON(geometry)
from uadbkt.gis_osm_pois_a_free_1
where osm_id ='41361350'
order by fclass, name;
"""
htw_geojson = geojson_from_sql(sql_htw_geojson, engine)
folium.GeoJson(htw_geojson).add_to(map_htw)
map_htw
```

```
latlng_berlin = [52.5170365, 13.3888599]
map_berlin = folium.Map(location=latlng_berlin, zoom_start=10)
sql_bezirke_geojson = """
select name as bname, ST_AsGeoJSON(shape) as geojson
from uadbkt.bezirk;
"""
with engine.connect() as con:
    bezirke_geojson_df = pd.read_sql_query(
        sql_bezirke_geojson, con)
for b in bezirke_geojson_df.apply(
        lambda r: folium.GeoJson(r.geojson), axis=1):
    b.add_to(map_berlin)
sql_bezirke_centroid_geojson = """
select
  name as bname,
  ST_Y(ST_Centroid(shape)) as clat,
  ST_X(ST_Centroid(shape)) as clng
from uadbkt.bezirk;
"""
with engine.connect() as con:
    bezirke_centroid_geojson_df = pd.read_sql_query(
        sql_bezirke_centroid_geojson, con
    )
def cmarker(r):
    loc = [r.clat, r.clng]
    html = f'<font color="green">{r.bname}</font>'
    divicon = folium.DivIcon(html=html)
    return folium.Marker(location=loc, icon=divicon)
for bc in bezirke_centroid_geojson_df.apply(
        lambda r: cmarker(r), axis=1):
    bc.add_to(map_berlin)
sql_bb_berlin = """
select ST_AsGeoJSON(ST_Extent(shape)) as bbox
from uadbkt.bezirk;
"""
bb_berlin = geojson_from_sql(sql_bb_berlin, engine)
folium.GeoJson(bb_berlin).add_to(map_berlin)
map_berlin
```

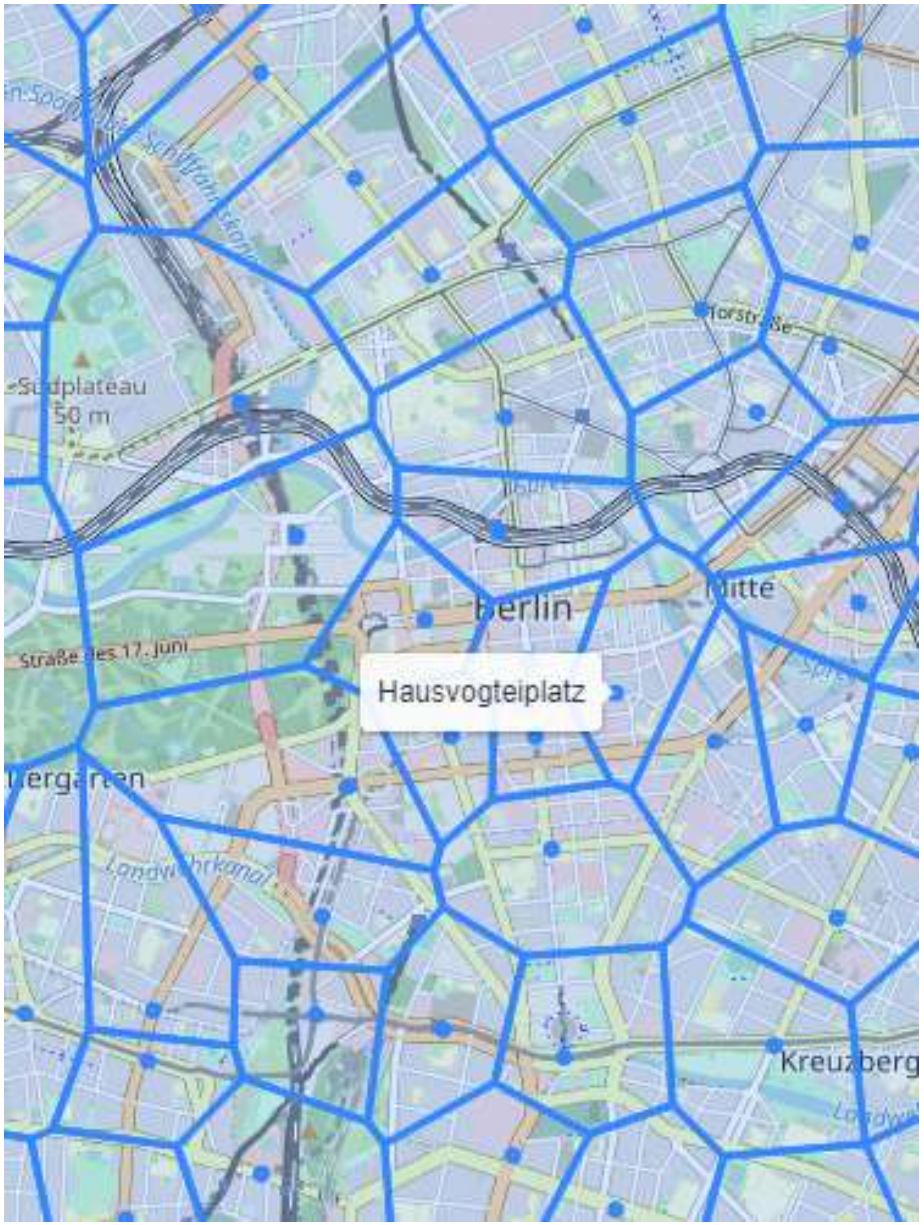


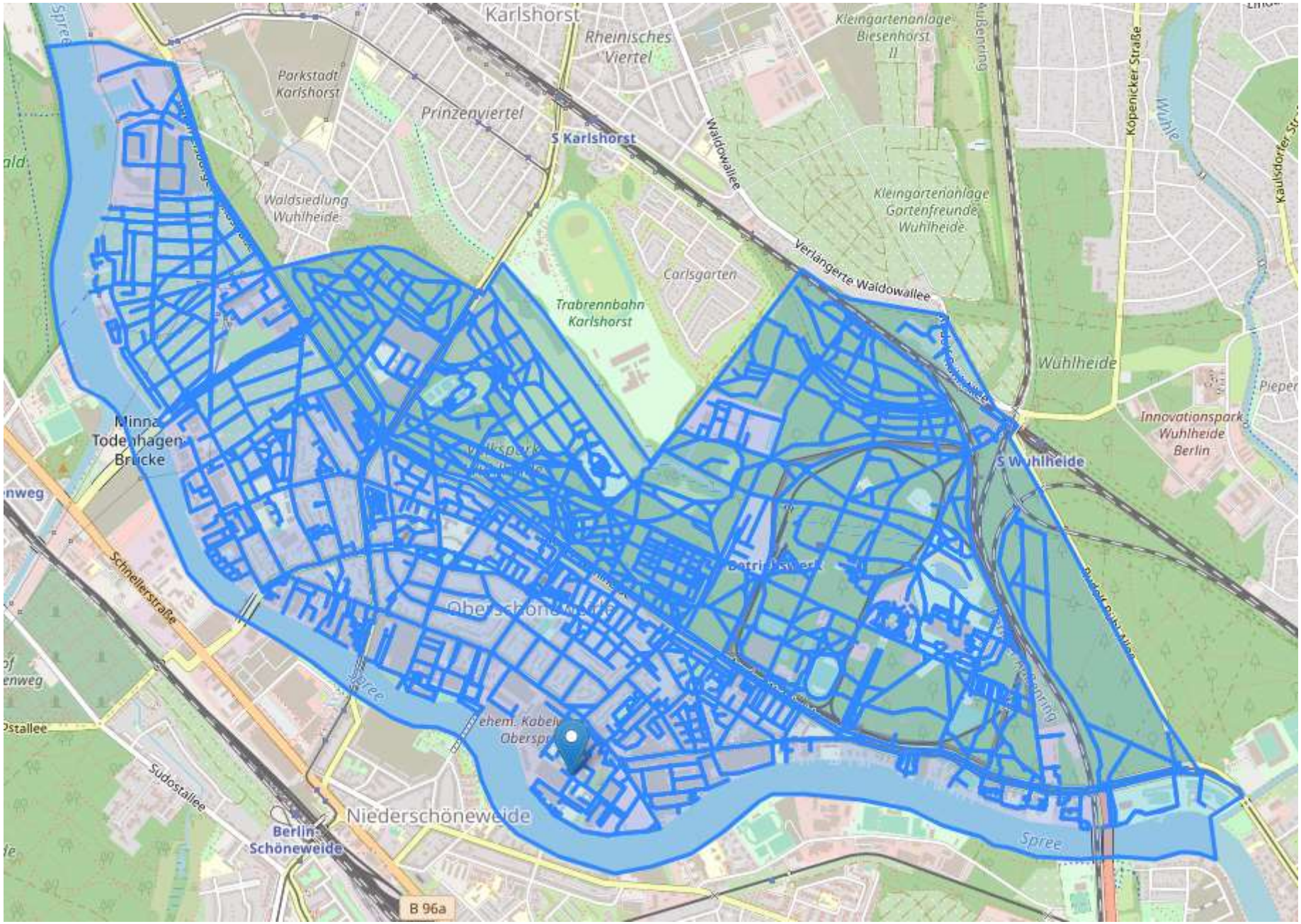
Haltestellen

Prof. Dr. Ingo Claßen

Montag, 21. Februar 2022 14:25

```
latlng_haltestellen = [52.5170365, 13.3888599]
map_haltestellen = folium.Map(location=latlng_haltestellen, zoom_start=13)
sql_voronoi = """
select ST_AsGeoJSON(
    ST_Transform((st_dump(ST_VoronoiPolygons(
        st_collect(h.posp)))).geom, 4326)
    ) as geojson
from uadbkt.haltestelle h
"""
with engine.connect() as con:
    voronoi_df = pd.read_sql_query(sql_voronoi, con)
for v in voronoi_df.apply(
        lambda r: folium.GeoJson(r.geojson), axis=1):
    v.add_to(map_haltestellen)
sql_haltestellen = """
select bez, lat, lng
from uadbkt.haltestelle;
"""
with engine.connect() as con:
    haltestellen_df = pd.read_sql_query(sql_haltestellen, con)
def hcircle(r):
    loc = [r.lat, r.lng]
    return folium.Circle(
        location=loc, radius=20,
        fill=True, fill_color="blue",
        tooltip=r.bez
    )
for h in haltestellen_df.apply(
        lambda r: hcircle(r), axis=1):
    h.add_to(map_haltestellen)
map_haltestellen
```




Karlshorst
Rheinisches Viertel
Parkstadt Karlshorst
Prinzenviertel
S Karlshorst
Waldowallee
Kleingartenanlage Biesenhorst II
Außenring
Köpenicker Straße
Wuhle
Kaulsdorfer Str
Waldsiedlung Wuhlheide
Kleingartenanlage Gartenfreunde Wuhlheide
Carlsgarten
Trabrennbahn Karlshorst
Verlängerte Waldowallee
Wuhlheide
Innovationspark Wuhlheide Berlin
S Wuhlheide
Minna Todenhagen Brücke
Schnellerstraße
Spree
Südostallee
Niederschöneweide
Berlin-Schöneweide
B 96a


ehem. Kabelwerk
Oberspree
Kaisersteg
BAE Batteries
GmbH
Technologie- und Gründerzentrum Spreeknie TGS
Bürogebäude im Block 56
Spreeknie
Spree
Ostendstraße
Hainstraße
Britzer Straße
Motorwassersportclub 1190
Nachwuchsleistungszentrum des 1. FC Union Berlin
Tennisclub Berlin-Oberspree

```
sql_htw_lat_lng = """
select ST_Y(ST_Centroid(geometry)) as lat, ST_X(ST_Centroid(geometry)) as lng
from ugeobln.gis_osm_pois_a_free_1
where osm_id ='41361350';
"""
htw_lat, htw_lng = lat_lng_from_sql(sql_htw_lat_lng, engine)
htw_lat, htw_lng
map_htw_stadtteil = folium.Map(location=[htw_lat, htw_lng], zoom_start=13)
folium.Marker(
    location=[htw_lat, htw_lng],
    tooltip="HTW"
).add_to(map_htw_stadtteil)
sql_htw_stadtteil = """
with
  htw_c as (
    select ST_Centroid(geometry) as c
    from ugeobln.gis_osm_pois_a_free_1
    where osm_id ='41361350'
  )
select ST_AsGeoJSON(geometry)
from ugeobln.gis_osm_places_a_free_1 p, htw_c
where ST_Covers(p.geometry, htw_c.c);
"""
```

```
htw_stadtteil = geojson_from_sql(sql_htw_stadtteil, engine)
folium.GeoJson(htw_stadtteil).add_to(map_htw_stadtteil)
sql_edges = """
with
  htw_c as (
    select ST_Centroid(geometry) as c
    from ugeobln.gis_osm_pois_a_free_1
    where osm_id ='41361350'
  ),
  htw_s as (
    select geometry
    from ugeobln.gis_osm_places_a_free_1 p, htw_c
    where ST_Covers(p.geometry, htw_c.c)
  )
select ST_AsGeoJSON(be.geometry) as geojson
from ugeobln.bln_edges be, htw_s
where ST_Covers(htw_s.geometry, be.geometry);
"""
with engine.connect() as con:
    edges_geojson_df = pd.read_sql_query(sql_edges, con)
for b in edges_geojson_df.apply(
        lambda r: folium.GeoJson(r.geojson), axis=1):
    b.add_to(map_htw_stadtteil)
map_htw_stadtteil
```