Kaj je s temi račkami?

Alen Mangafić, alen.mangafic@gis.si

29. Januar 2025

Motivacija

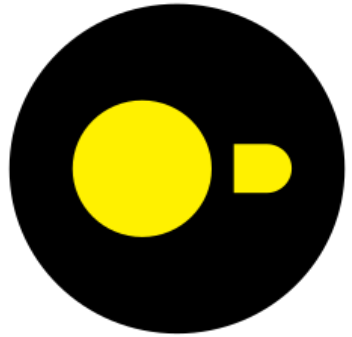
I. Predstavitev novih tehnologij: DuckDB in FireDucks

II. Primerjava s prilubljenimi orodji: prednosti, slabosti

III. Račke

Raca numero uno: DuckDB

- **Odprtokoden projekt.**
- **In-memory OLAP SQL baza podatkov** (optimizirana za analitiko).
- **Podpira standardni SQL** in deluje na različnih platformah.
- **Optimizirana za analizo velikih tabel** na lokalnih napravah.
- **Ponuja API, ki podpira tvoje delo** (Python, R, C++, Rust, …).
- **Odlična dokumentacija**.



DuckDB

Zakaj ni primeren DuckDB?

- **Ni primeren za strežniške zgodbe**, saj nima strežniške arhitekture:
  - Večuporabniške aplikacije in dostopi (npr. sočasni dostop je omejen).
  - Ni replikacije (ne mreš vzpostaviti HA ali distribuirane instance)
  - Omejene transakcije (ACID: izolirane instance, ni centralnega sledenja…).

- **Kaj mi ni všeč**
  - QGIS ne omogoča pregleda, če si povezan z drugim orodjem (buuuuu).
  - Izvoz je počasen.

Zakaj mi je všeč DuckDB?

- **SQL sintaksa kot PostgreSQL/PostGIS**
  - Ni treba učiti vse na novo, prehod je enostaven.
  - Ima svoje dodatne funkcije, ki so veliko "lepše" od dosedanjih (npr. uvoz in izvoz podatkov).

- **Ponuja veliko razširitev. Izbrane:**
  - **spatial** (GIS) – prostorska podpora (tudi GDAL integracija).
  - **postgres** - poizvedbe znotraj obstoječega **PostgreSQL** ekosistema.
  - AWS, Excel, MySQL, h3 (community extension).

Zakaj mi je všeč DuckDB?

- **Osvobojeni smo administracije**
  - Ni treba urejati uporabnikov, pravic, povezav.
  - Ni ločenega strežnika – **deluje kot običajna datoteka**.

- **Minimalna namestitev**
  - Vse odvisnosti so urejene.
  - Brez kompleksne konfiguracije – **preneseš ga in že dela.**

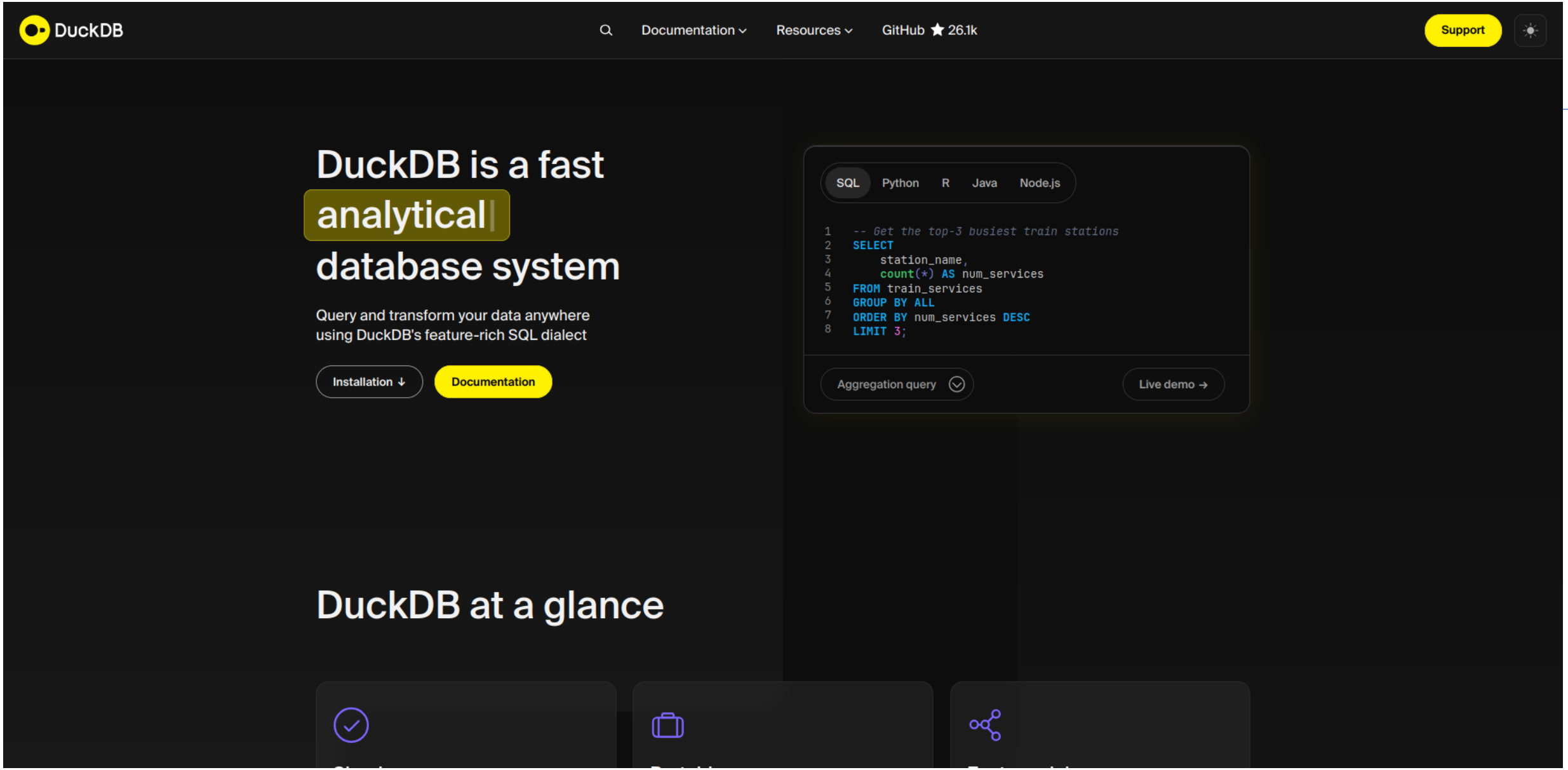
- **QGIS vtičnik**
  - Omogoča neposreden pregled podatkov.

- **Določeni IDE ga podpirajo direktno**
  - Visual Studio Code (open source).
  - Beekeper Studio pravi "coming soon" (open source).
  - DBeaver Community (open source).

Zakaj mi je všeč DuckDB?

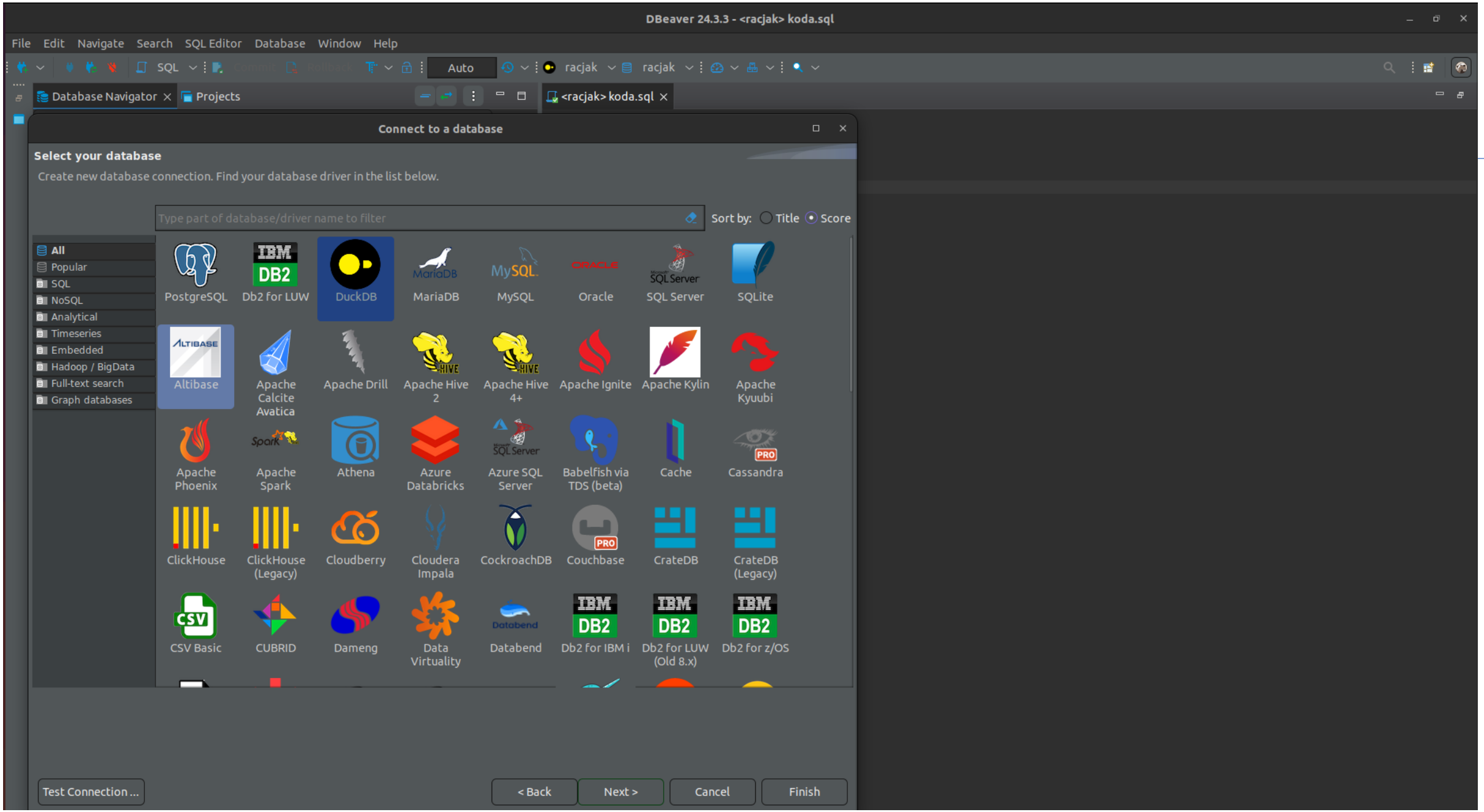
- **Odličen za študente**
  - Ni treba postavljati strežnika.
  - Deluje tudi na šibkejši strojni opremi.
  - SQL znanje je prenosljivo na PostgreSQL/PostGIS.
  - Celotna baza je ena sama datoteka – enostavno posreduješ svoje delo sotrpinom.

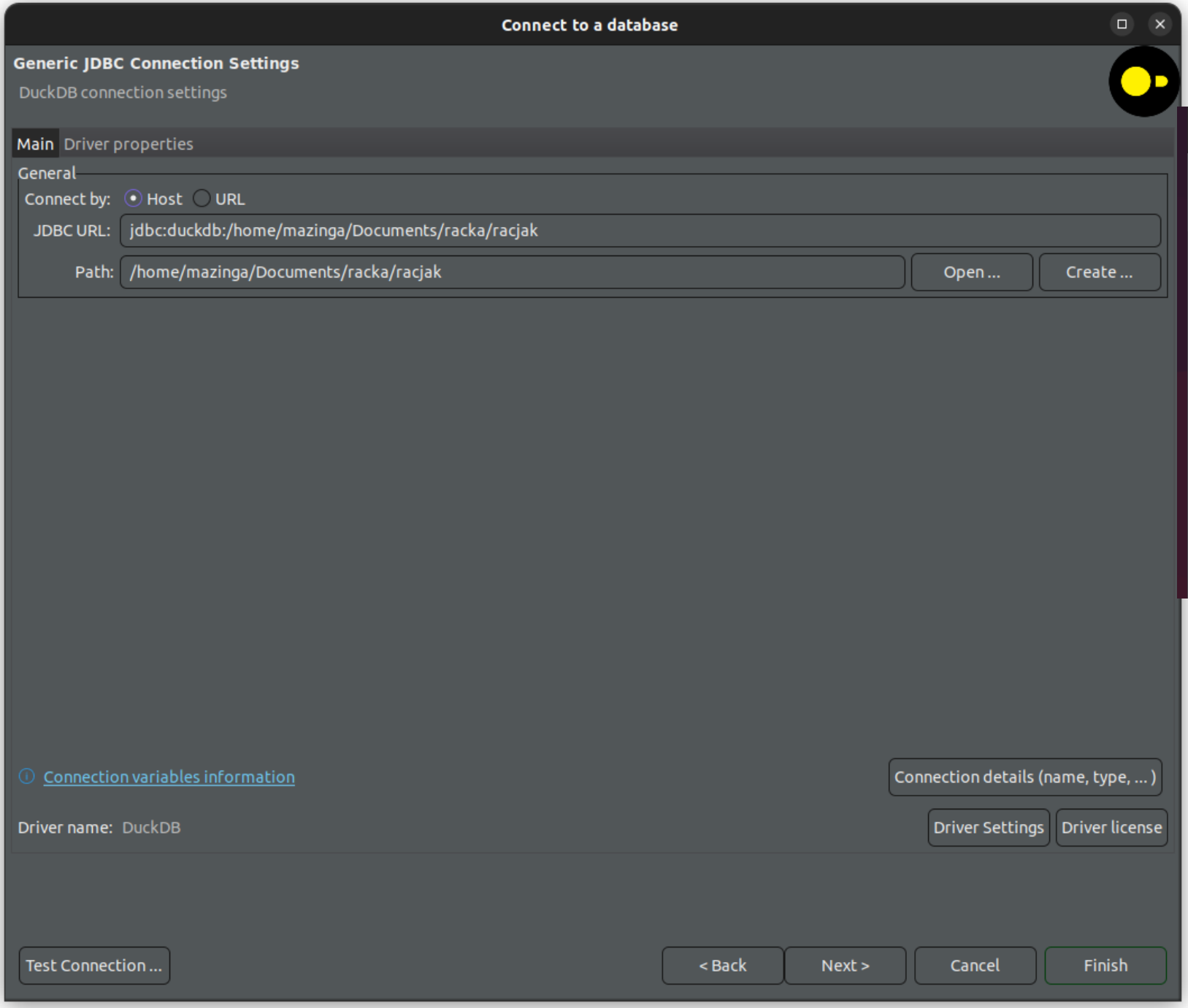
- **Skupnost**
  - Bogata dokumentacija in lastna konferenca
  (Duckcon #6: 31. 1. 2025 @ Amsterdam).
  - DuckDB raste in dobiva vedno več uporabnikov.

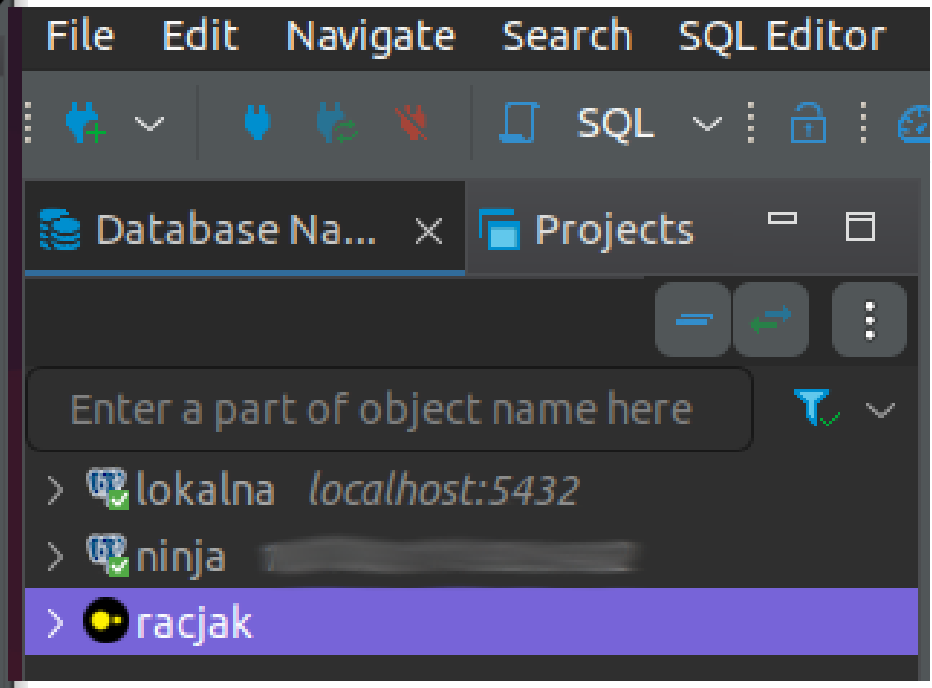

DuckDB
Documentation
Resources
GitHub 26.1k
Support
DuckDB is a fast analytical database system
Query and transform your data anywhere using DuckDB's feature-rich SQL dialect
Installation ↓
Documentation
SQL
Python
R
Java
Node.js
-- Get the top-3 busiest train stations
SELECT
station_name,
count(*) AS num_services
FROM train_services
GROUP BY ALL
ORDER BY num_services DESC
LIMIT 3;
Aggregation query
Live demo →
DuckDB at a glance

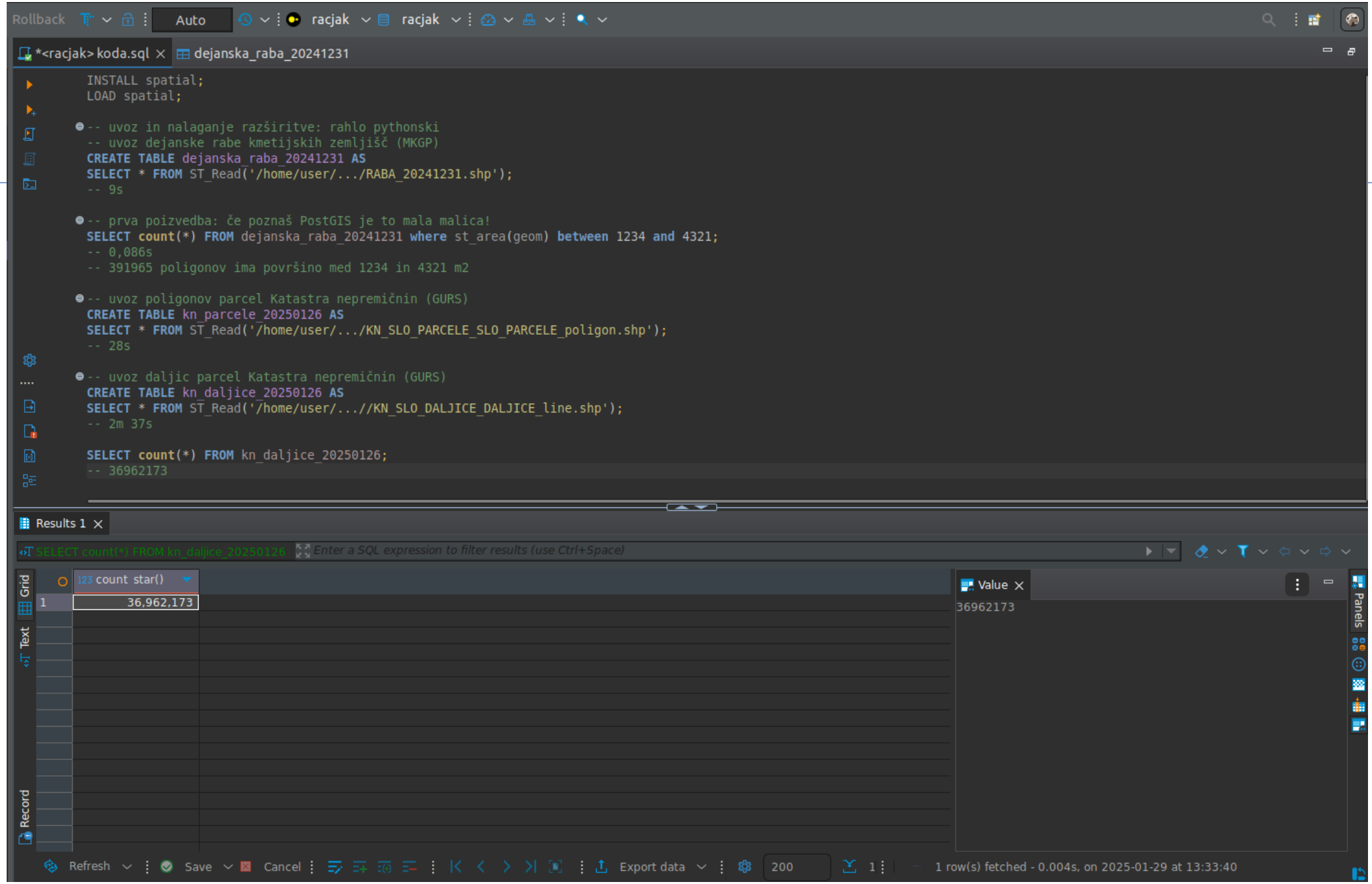
Študijski primer: namestitev, zagon, primerjava.

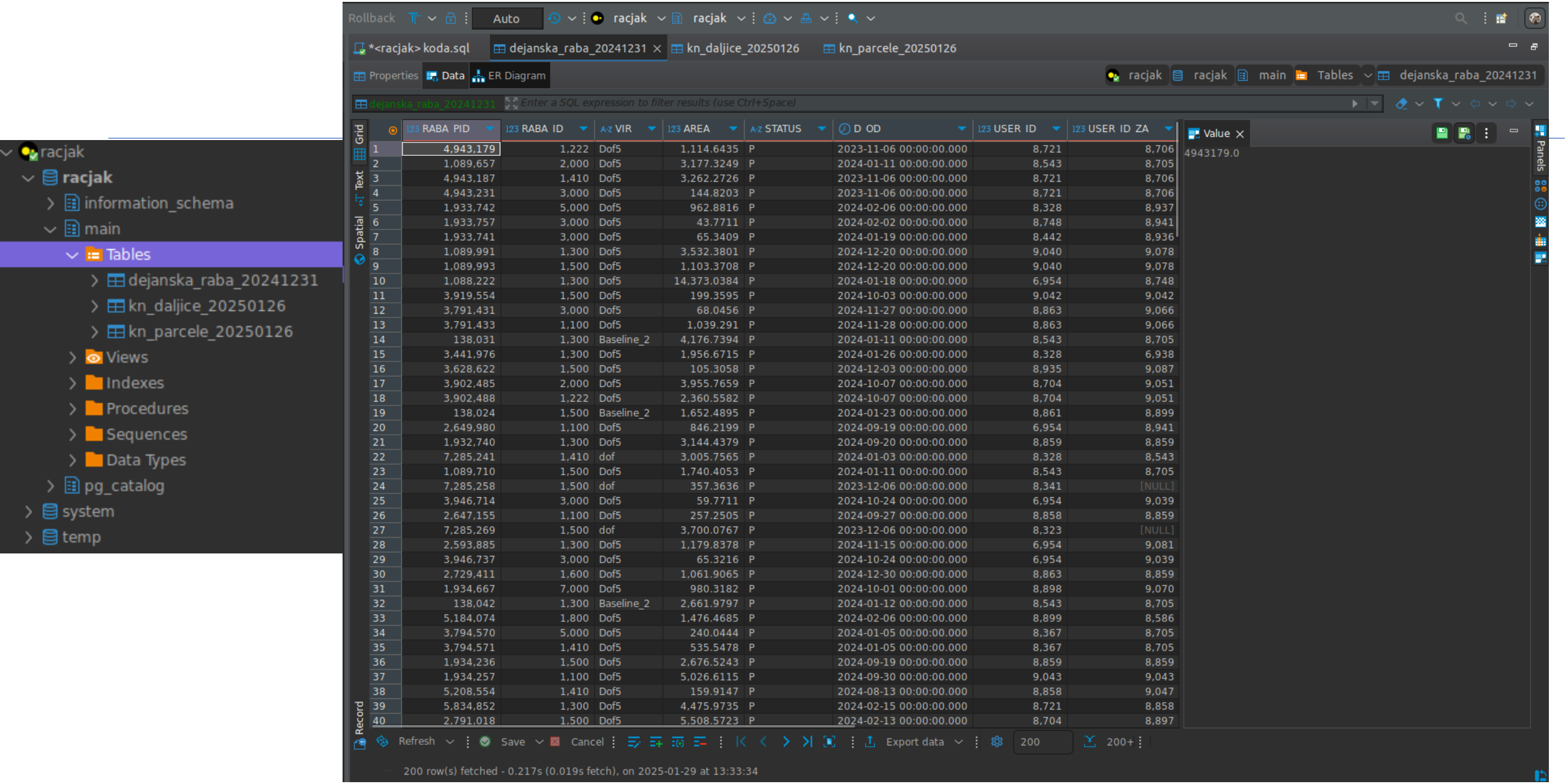
- Namestili bomo DuckDB z DBeaver.
- Namestili bomo vtičnik za prostorsko obdelavo in podporo PostgreSQL.
- Uvozili bomo podatke iz Katastra nepremičnin dejansko rabo.
- Zagnali bomo prostorske poizvedbe in merili čas.
- Namestili bomo QGIS vtičnik in pregledali podatke.
- Namestili ga bomo v Python in uvozili rezultate za naslednjo raco.


DBeaver 24.3.3 - <racjak> koda.sql
File Edit Navigate Search SQL Editor Database Window Help
Database Navigator
Projects
<racjak> koda.sql
Connect to a database
Select your database
Create new database connection. Find your database driver in the list below.
Type part of database/driver name to filter
Sort by: Title Score
All
Popular
SQL
NoSQL
Analytical
Timeseries
Embedded
Hadoop / BigData
Full-text search
Graph databases
PostgreSQL
Db2 for LUW
DuckDB
MariaDB
MySQL
Oracle
SQL Server
SQLite
Altibase
Apache Calcite Avatica
Apache Drill
Apache Hive 2
Apache Hive 4+
Apache Ignite
Apache Kylin
Apache Kyuubi
Apache Phoenix
Apache Spark
Athena
Azure Databricks
Azure SQL Server
Babelfish via TDS (beta)
Cache
Cassandra
ClickHouse
ClickHouse (Legacy)
Cloudberry
Cloudera Impala
CockroachDB
Couchbase
CrateDB
CrateDB (Legacy)
CSV Basic
CUBRID
Dameng
Data Virtuality
Databend
Db2 for IBM i
Db2 for LUW (Old 8.x)
Db2 for z/OS
Test Connection ...
< Back
Next >
Cancel
Finish


Connect to a database
Generic JDBC Connection Settings
DuckDB connection settings
Main
Driver properties
General
Connect by:
Host
URL
JDBC URL:
jdbc:duckdb:/home/mazinga/Documents/racka/racjak
Path:
/home/mazinga/Documents/racka/racjak
Open ...
Create ...
Connection variables information
Connection details (name, type, ...)
Driver name:
DuckDB
Driver Settings
Driver license
Test Connection ...
< Back
Next >
Cancel
Finish


File
Edit
Navigate
Search
SQL Editor
SQL
Database Na...
Projects
Enter a part of object name here
lokalna
localhost:5432
ninja
racjak


Rollback
Auto
racjak
racjak
*<racjak> koda.sql
dejanska_raba_20241231
INSTALL spatial;
LOAD spatial;
-- uvoz in nalaganje razširitve: rahlo pythonski
-- uvoz dejanske rabe kmetijskih zemljišč (MKGP)
CREATE TABLE dejanska_raba_20241231 AS
SELECT * FROM ST_Read('/home/user/.../RABA_20241231.shp');
-- 9s
-- prva poizvedba: če poznaš PostGIS je to mala malica!
SELECT count(*) FROM dejanska_raba_20241231 where st_area(geom) between 1234 and 4321;
-- 0,086s
-- 391965 poligonov ima površino med 1234 in 4321 m2
-- uvoz poligonov parcel Katastra nepremičnin (GURS)
CREATE TABLE kn_parcele_20250126 AS
SELECT * FROM ST_Read('/home/user/.../KN_SLO_PARCELE_SLO_PARCELE_poligon.shp');
-- 28s
-- uvoz daljic parcel Katastra nepremičnin (GURS)
CREATE TABLE kn_daljice_20250126 AS
SELECT * FROM ST_Read('/home/user/...//KN_SLO_DALJICE_DALJICE_line.shp');
-- 2m 37s
SELECT count(*) FROM kn_daljice_20250126;
-- 36962173
Results 1
SELECT count(*) FROM kn_daljice_20250126
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
123 count star()
1
36,962,173
Value
36962173
Panels
Refresh
Save
Cancel
Export data
200
1
1 row(s) fetched - 0.004s, on 2025-01-29 at 13:33:40


racjak
racjak
information_schema
main
Tables
dejanska_raba_20241231
kn_daljice_20250126
kn_parcele_20250126
Views
Indexes
Procedures
Sequences
Data Types
pg_catalog
system
temp
Rollback
Auto
racjak
racjak
*<racjak> koda.sql
dejanska_raba_20241231
kn_daljice_20250126
kn_parcele_20250126
Properties
Data
ER Diagram
racjak
racjak
main
Tables
dejanska_raba_20241231
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Spatial
Record
RABA_PID
RABA_ID
VIR
AREA
STATUS
D_OD
USER_ID
USER_ID_ZA
1 4,943,179 1,222 Dof5 1,114.6435 P 2023-11-06 00:00:00.000 8,721 8,706
2 1,089,657 2,000 Dof5 3,177.3249 P 2024-01-11 00:00:00.000 8,543 8,705
3 4,943,187 1,410 Dof5 3,262.2726 P 2023-11-06 00:00:00.000 8,721 8,706
4 4,943,231 3,000 Dof5 144.8203 P 2023-11-06 00:00:00.000 8,721 8,706
5 1,933,742 5,000 Dof5 962.8816 P 2024-02-06 00:00:00.000 8,328 8,937
6 1,933,757 3,000 Dof5 43.7711 P 2024-02-02 00:00:00.000 8,748 8,941
7 1,933,741 3,000 Dof5 65.3409 P 2024-01-19 00:00:00.000 8,442 8,936
8 1,089,991 1,300 Dof5 3,532.3801 P 2024-12-20 00:00:00.000 9,040 9,078
9 1,089,993 1,500 Dof5 1,103.3708 P 2024-12-20 00:00:00.000 9,040 9,078
10 1,088,222 1,300 Dof5 14,373.0384 P 2024-01-18 00:00:00.000 6,954 8,748
11 3,919,554 1,500 Dof5 199.3595 P 2024-10-03 00:00:00.000 9,042 9,042
12 3,791,431 3,000 Dof5 68.0456 P 2024-11-27 00:00:00.000 8,863 9,066
13 3,791,433 1,100 Dof5 1,039.291 P 2024-11-28 00:00:00.000 8,863 9,066
14 138,031 1,300 Baseline_2 4,176.7394 P 2024-01-11 00:00:00.000 8,543 8,705
15 3,441,976 1,300 Dof5 1,956.6715 P 2024-01-26 00:00:00.000 8,328 6,938
16 3,628,622 1,500 Dof5 105.3058 P 2024-12-03 00:00:00.000 8,935 9,087
17 3,902,485 2,000 Dof5 3,955.7659 P 2024-10-07 00:00:00.000 8,704 9,051
18 3,902,488 1,222 Dof5 2,360.5582 P 2024-10-07 00:00:00.000 8,704 9,051
19 138,024 1,500 Baseline_2 1,652.4895 P 2024-01-23 00:00:00.000 8,861 8,899
20 2,649,980 1,100 Dof5 846.2199 P 2024-09-19 00:00:00.000 6,954 8,941
21 1,932,740 1,300 Dof5 3,144.4379 P 2024-09-20 00:00:00.000 8,859 8,859
22 7,285,241 1,410 dof 3,005.7565 P 2024-01-03 00:00:00.000 8,328 8,543
23 1,089,710 1,500 Dof5 1,740.4053 P 2024-01-11 00:00:00.000 8,543 8,705
24 7,285,258 1,500 dof 357.3636 P 2023-12-06 00:00:00.000 8,341 [NULL]
25 3,946,714 3,000 Dof5 59.7711 P 2024-10-24 00:00:00.000 6,954 9,039
26 2,647,155 1,100 Dof5 257.2505 P 2024-09-27 00:00:00.000 8,858 8,859
27 7,285,269 1,500 dof 3,700.0767 P 2023-12-06 00:00:00.000 8,323 [NULL]
28 2,593,885 1,300 Dof5 1,179.8378 P 2024-11-15 00:00:00.000 6,954 9,081
29 3,946,737 3,000 Dof5 65.3216 P 2024-10-24 00:00:00.000 6,954 9,039
30 2,729,411 1,600 Dof5 1,061.9065 P 2024-12-30 00:00:00.000 8,863 8,859
31 1,934,667 7,000 Dof5 980.3182 P 2024-10-01 00:00:00.000 8,898 9,070
32 138,042 1,300 Baseline_2 2,661.9797 P 2024-01-12 00:00:00.000 8,543 8,705
33 5,184,074 1,800 Dof5 1,476.4685 P 2024-02-06 00:00:00.000 8,899 8,586
34 3,794,570 5,000 Dof5 240.0444 P 2024-01-05 00:00:00.000 8,367 8,705
35 3,794,571 1,410 Dof5 535.5478 P 2024-01-05 00:00:00.000 8,367 8,705
36 1,934,236 1,500 Dof5 2,676.5243 P 2024-09-19 00:00:00.000 8,859 8,859
37 1,934,257 1,100 Dof5 5,026.6115 P 2024-09-30 00:00:00.000 9,043 9,043
38 5,208,554 1,410 Dof5 159.9147 P 2024-08-13 00:00:00.000 8,858 9,047
39 5,834,852 1,300 Dof5 4,475.9735 P 2024-02-15 00:00:00.000 8,721 8,858
40 2,791,018 1,500 Dof5 5,508.5723 P 2024-02-13 00:00:00.000 8,704 8,897
Value
4943179.0
Panels
Refresh
Save
Cancel
Export data
200
200+
200 row(s) fetched - 0.217s (0.019s fetch), on 2025-01-29 at 13:33:34

Primerjava #1: uvoz

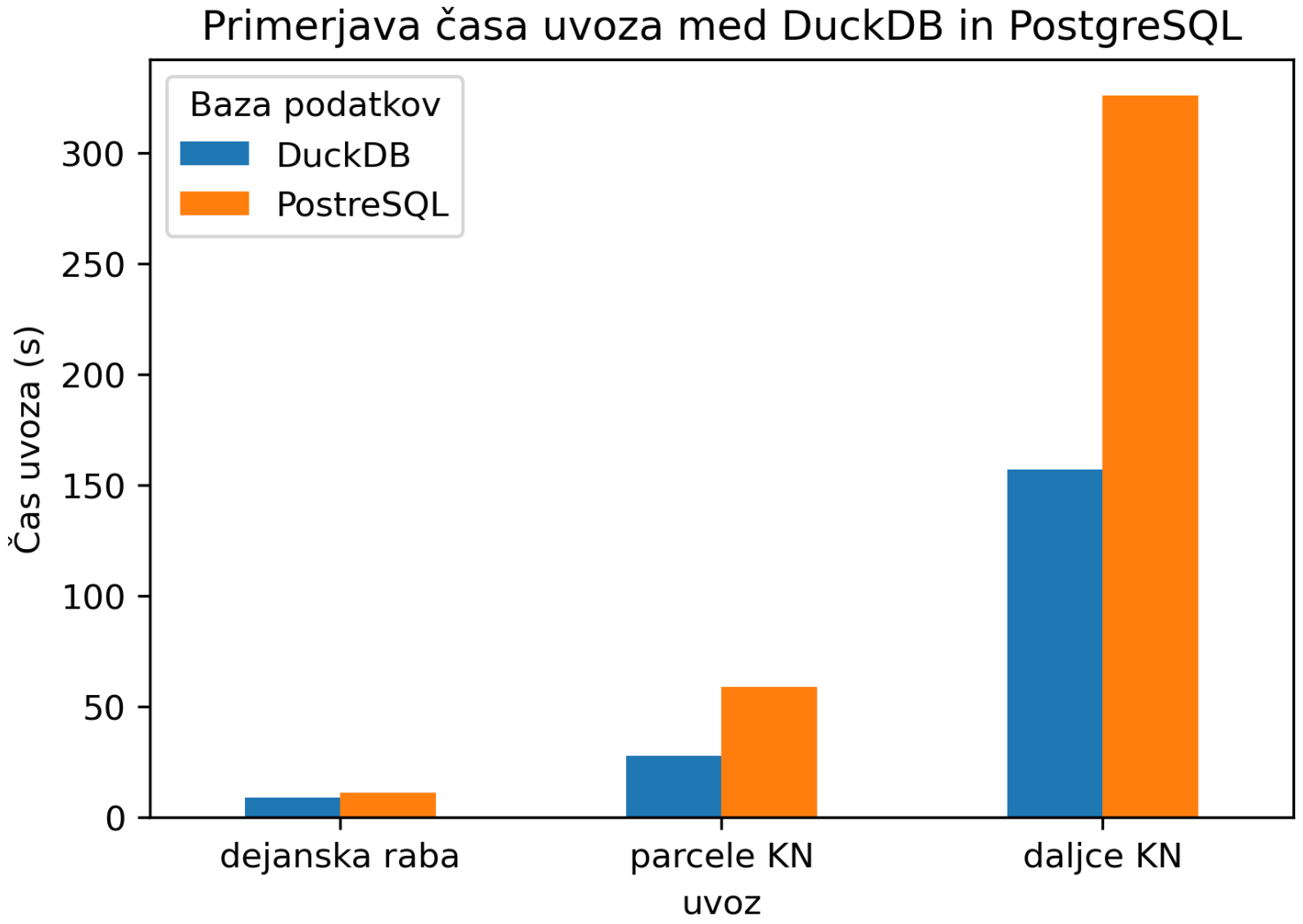
- **DuckDB uvoz:**

```
CREATE TABLE dejanska_raba_20241231 AS
SELECT * FROM ST_Read('/home/user/.../podatek.shp');
```

- **ogr2ogr uvoz (PostGIS) + time:**

```
time ogr2ogr -f "PostgreSQL" PG:"d host=ip port=5432 name=tvoja_baza user=tvoj_uporabnik
password=geslo \
"/home/user/.../podatek.shp" -nln dejanska_raba_20241231 \
 --config PG_USE_COPY YES \
-lco SPATIAL_INDEX=GIST \
-lco PRECISION=NO #za parcele
```

Primerjava #1: uvoz



Primerjava #2: izvoz v GPKG

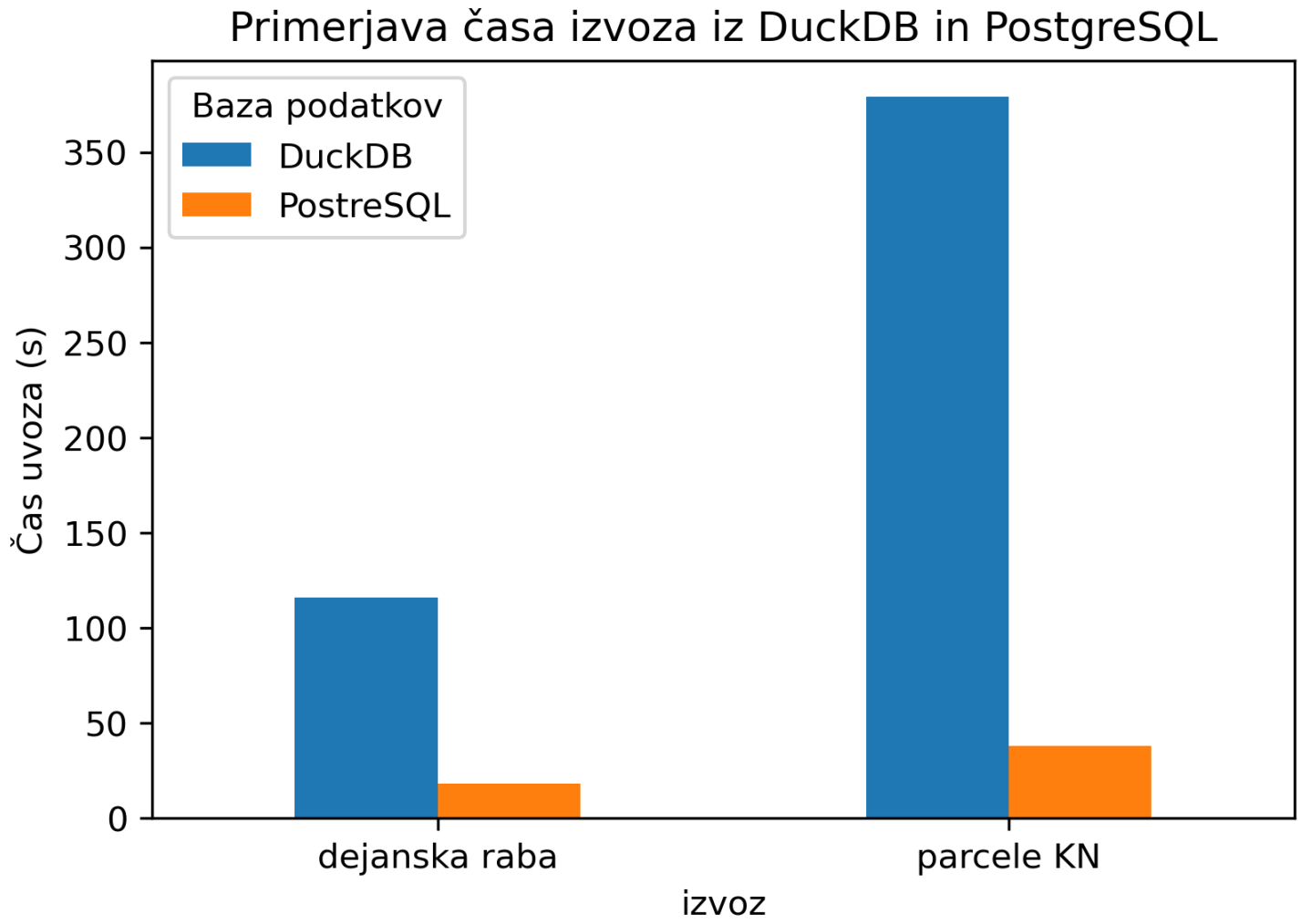
- **DuckDB uvoz**:

```
COPY tabela TO '/home/user/.../podatek.gpkg'
WITH (FORMAT GDAL, DRIVER 'GPKG', LAYER_CREATION_OPTIONS 'WRITE_BBOX=YES');
```

- **ogr2ogr uvoz (PostGIS) + time**:

```
time ogr2ogr -f GPKG /home/user/.../podatek.gpkg \
    PG:"host=x port=x dbname=x user=xpassword=x" \
    -nln tabela \
    -sql "SELECT * FROM tabela \
    --config PG_USE_COPY YES
```

Primerjava #2: izvoz v GPKG



Primerjava #3: poizvedbe

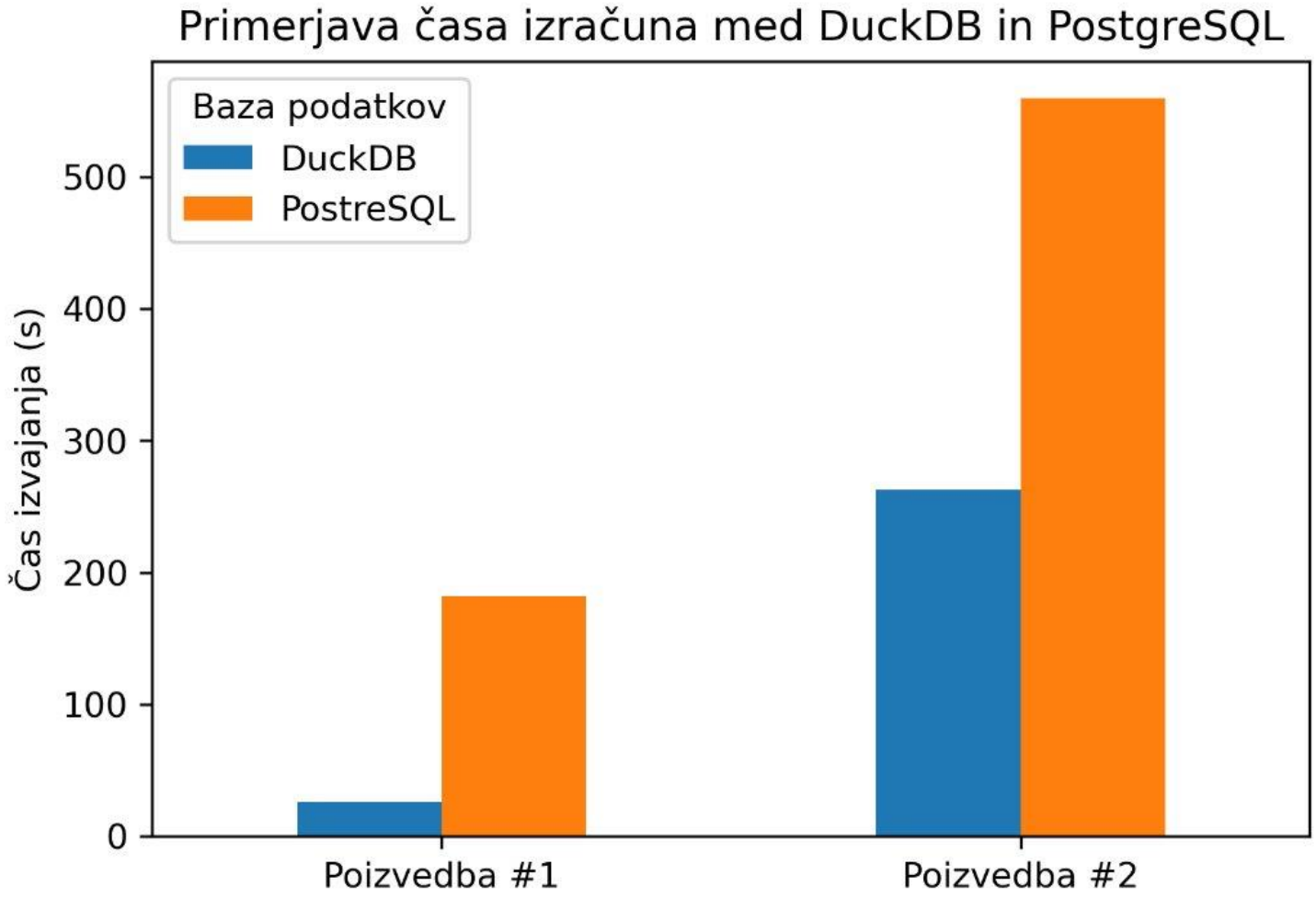
- ## Pozvedba #1: združi vse geometrije z enako rabo (dissolve)

```
CREATE TABLE test_diss AS
SELECT
    "RABA_ID",st_union(geom,geom) AS geom
FROM raba_dejanska_raba_20241231
GROUP BY
    "RABA_ID", geom;
```

Pozvedba #2: razbijanje parcel glede na dejansko rabo

```
CREATE TABLE test_inters AS
SELECT
    a.*, b."RABA_ID" AS raba_id,
    ST_Intersection(a.geom, b.geom) AS geom
FROM
    kn_parcele_20250126 a
JOIN
    dejanska_raba_20241231 b
ON
    ST_Intersects(a.geom, b.geom);
```

Primerjava #3: poizvedbe



Kombiniranje obstoječe PostgreSQL baze

- Namestimo razširitev "postgres" ter je naložimo (INSTALL/LOAD).
- Povežemo se na bazo:

```
ATTACH 'dbname=x user=x host=x' AS moj_postgres(TYPE POSTGRES);
```

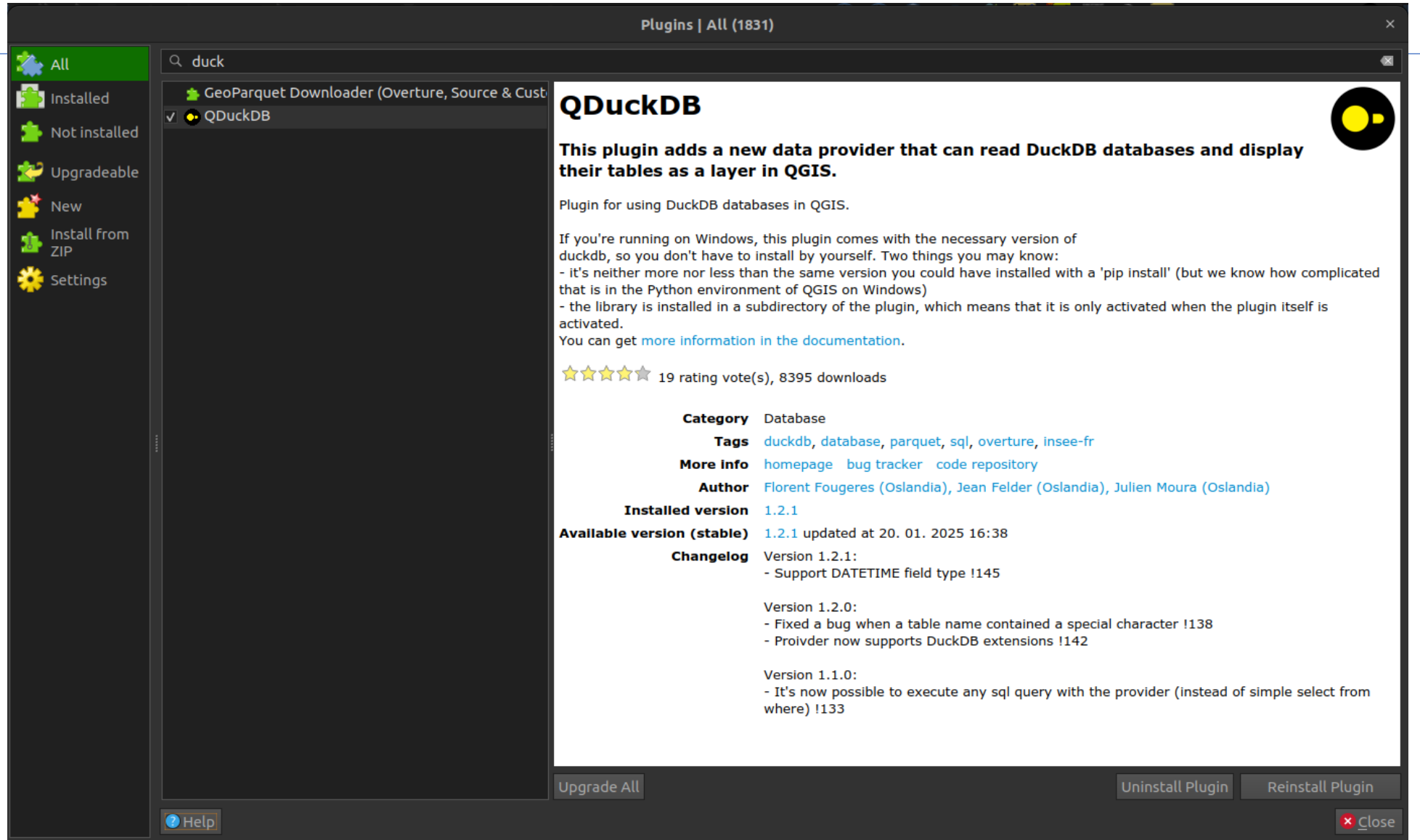
- Lahko se povežemo tudi kot bralci (dodamo READ_ONLY)
- Primer branja:

```
SELECT * FROM moj_postgres.shema.moja_tabela;
```

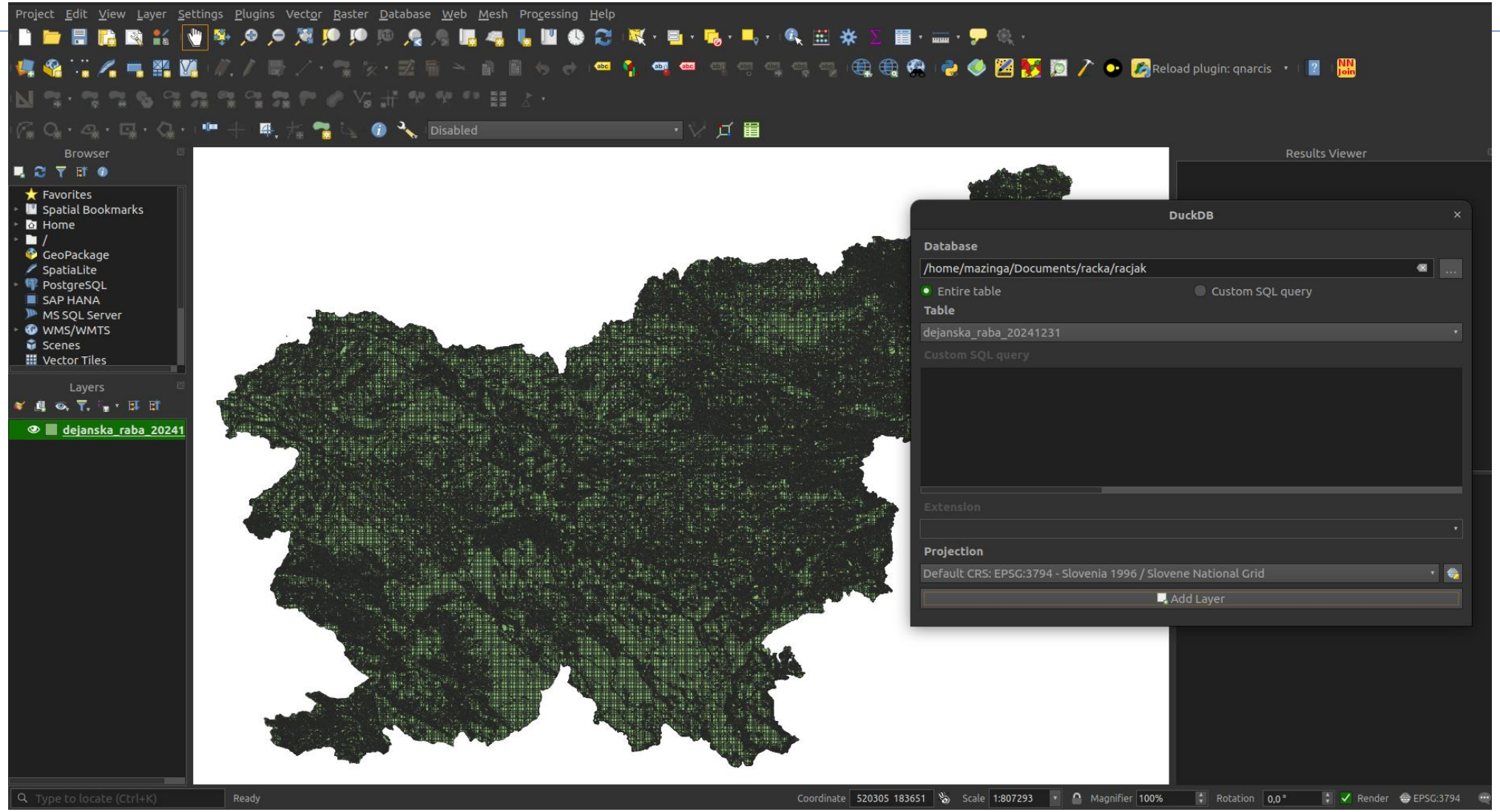
- Primer pisanja:

```
CREATE TABLE moj_postgres.tbl (id INTEGER, name VARCHAR);
INSERT INTO moj_postgres.tbl VALUES (666, 'DuckDB');
```

QGIS vtičnik - pregled podatkov



QGIS vtičnik - pregled podatkov



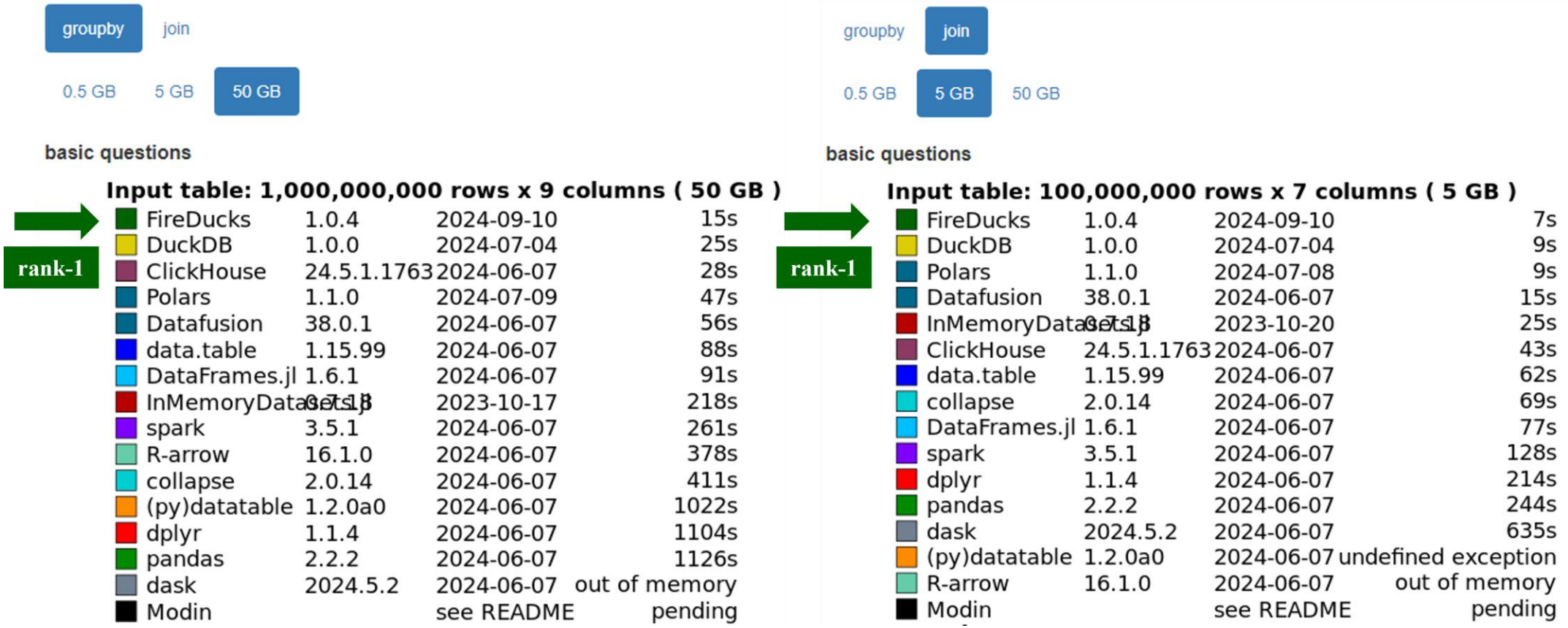
Raca numero due: FireDucks

- Visokozmogljiva knjižnica za obdelavo podatkov v Pythonu, popolnoma združljiva s pandas API-jem.

- Izvaja operacije z uporabo več jeder CPU

- Popolna združljivost s pandas: omogoča uporabo obstoječe pandas kode brez sprememb. This will blow your mind: **import fireducks.pandas as pd**

FireDucks: uradni benchmark je grd in kul



https://fireducks-dev.github.io/

Python in race

- **Namestitev novega python okolja in pip:**

```
micromamba create -n duck_env python=3.10
micromamba activate duck_env
micromamba install -c conda-forge duckdb matplotlib time
pip install fireducks
```

- Uvoz dejanske rabe iz DuckDB v Python
- Poizvedba: združevanje površin trajnih nasadov (0,0166 s).

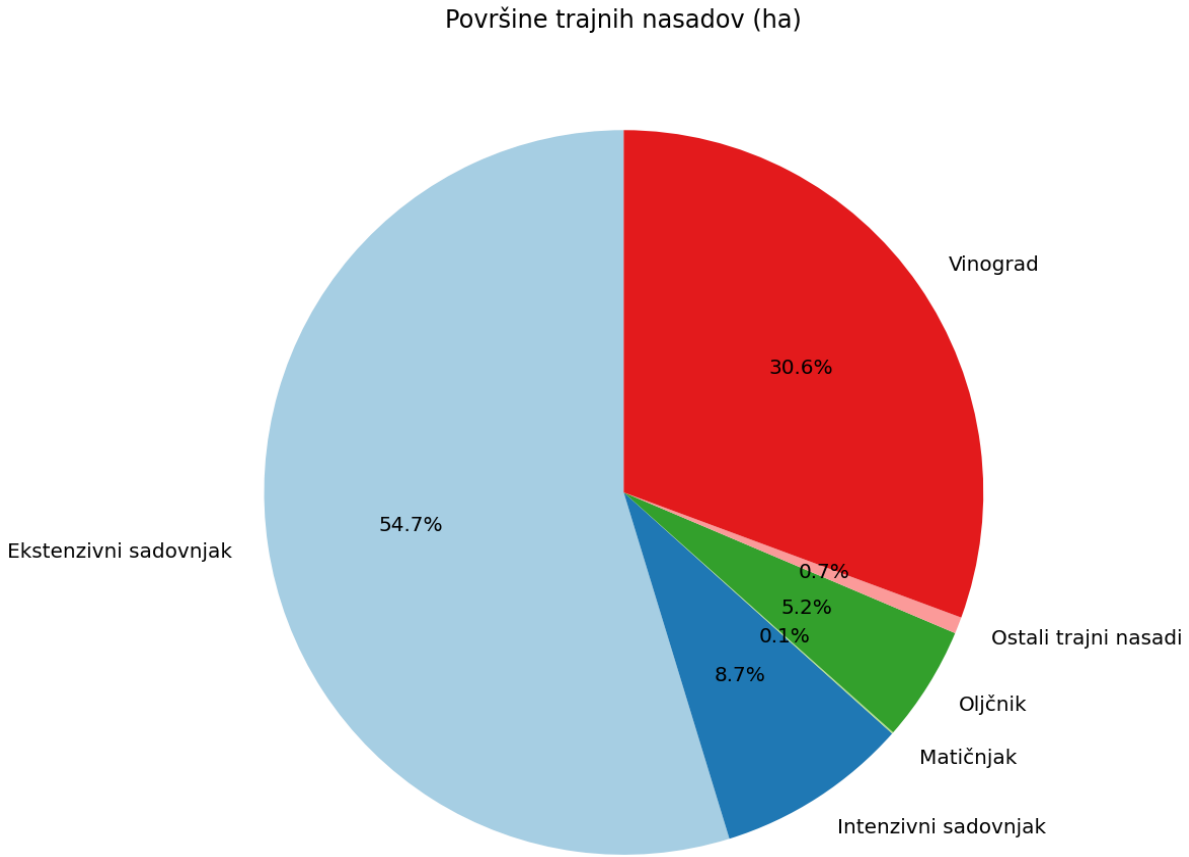
Python in race

```
import fireducks.pandas as pd
import duckdb

# Povezava z DuckDB
conn = duckdb.connect("/home/user/.../racka")

Uvoz podatkov iz DuckDB v FireDucks DataFrame
df = conn.execute("SELECT * FROM dejanska_raba_20241231").df()
```

Python in race



Hvala za pozornost.
Vprašanja?

- Alen Mangafić, alen.mangafic@gis.si