DuckDB : A Critical Evaluation

Table of Contents

This report focuses on the performance comparison of widely used database interaction libraries by measuring the execution time for uploading a dataset into a database and retrieving data from it. To assess upload performance DuckDB, SQLite, pyodbc, SQLAlchemy, polars and pandas were evaluated. For data retrieval, DuckDB and SQLite were evaluated.

Methodology

The dataset consisted of 100,000 customer records across 12 columns, with no missing values (Figure 1).

```
Data columns (total 12 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   Index              100000 non-null  int64
 1   Customer Id        100000 non-null  object
 2   First Name         100000 non-null  object
 3   Last Name          100000 non-null  object
 4   Company            100000 non-null  object
 5   City               100000 non-null  object
 6   Country            100000 non-null  object
 7   Phone 1            100000 non-null  object
 8   Phone 2            100000 non-null  object
 9   Email              100000 non-null  object
 10  Subscription Date  100000 non-null  object
 11  Website            100000 non-null  object
```

Figure 1: Information about the dataset

Two sets of execution times (in seconds) were calculated by running 20 iterations on each library.

- Group1 (DuckDB) : Execution time for DuckDB (n=20)
- Group2 (SQLite) : Execution time for SQLite (n=20)

The performance comparison was conducted on a machine with 16.0 GB of RAM (15.7 GB usable) and a 12th Gen Intel(R) Core (TM) i5-1235U 1.30 GHz processor.

Assumptions of the t -test

- The execution time data are independent.
- The execution time data is normally distributed within each group.
- The variances of the two groups are equal.

An independent two sample one tailed t – test was performed to compare the mean execution times of DuckDB and SQLite, using a significant level 0.05.

Comparison of uploading data set into a database using DuckDB and SQLite

DuckDB, SQLite, pyodbc, SQLAlchemy, polars and pandas – execution times were measured for uploading a dataset into a database using each library. According to the results, pyodbc, SQLAlchemy, polars and pandas exhibited longer execution times, while DuckDB and SQLite had shorter times. Furthermore, DuckDB demonstrated slightly faster performance than SQLite. Therefore, to statistically evaluate whether the execution time of DuckDB was significantly less than that of SQLite, a one-tailed t-test was conducted.

1.1. Hypothesis

- Null Hypothesis (H_0): The mean execution time of DuckDB is greater than or equal to the mean execution time of SQLite
- Alternative Hypothesis (H_1): The mean execution time of DuckDB is lesser than the mean execution time of SQLite

Table 1: Execution Times to upload dataset

DuckDB	SQLite
0.4697234630584717, 0.4788205623626709, 0.536827564239502, 0.4731323719024658, 0.46656346321105957, 0.4908170700073242, 0.5362567901611328, 0.49750423431396484, 0.44797563552856445, 0.45302319526672363, 0.5152430534362793, 0.4853672981262207, 0.4698634147644043,	0.9481751918792725, 0.7332606315612793, 0.6662595272064209, 0.651663064956665, 0.6665325164794922, 0.6825902462005615, 0.6834754943847656, 0.6581552028656006, 0.677525520324707, 0.6639862060546875, 0.682966947555542, 0.6992640495300293, 0.6502251625061035, 0.6663634777069092,

0.49849557876586914, 0.48340272903442383, 0.4624810218811035, 0.4858245849609375, 0.47834157943725586, 0.4999072551727295, 0.47641921043395996	0.6993062496185303, 0.6830713748931885, 0.6887185573577881, 0.6911695003509521, 0.6760156154632568, 0.7001526355743408

Table 2: Descriptive Statistics of uploading data

Statistic	DuckDB	SQLite
Iterations (n)	20	20
Mean(s)	0.48529950380325315	0.6934438586235047
Standard Deviation	0.023867980766678377	0.06304924026700218
Variance	0.000569680505878529	0.0039752066982461694

1.2. Test Results

- t – value : -13.80759626848063
- p – value : 1.0892690690592714e-16

    - Since the p-value is less than the significance level, we reject the null hypothesis. Thus, considering the negative t-value as well, it indicates that the mean execution time of DuckDB is lesser than the mean execution time of SQLite.

1.3. Conclusion of the t-test

- The t-test results show that, to upload the data into a database, DuckDB is significantly faster than SQLite, based on the mean execution times collected during the experiment.

Comparison of retrieving data from a database using DuckDB and SQLite

The experiment involved querying both DuckDB and SQLite to retrieve customer Ids where the subscription date is greater than January 1, 2020. This query was executed on both databases and the execution times were measured. After observing that the execution time of DuckDB is lesser than the execution time of SQLite, a one tailed t-test was conducted to statistically evaluate whether the mean

execution time of DuckDB for retrieving the specified customer Ids was significantly less than that of SQLite.

2.1. Hypothesis

- Null Hypothesis (H_0): The mean execution time of DuckDB is greater than or equal to the mean execution time of SQLite
- Alternative Hypothesis (H_1): The mean execution time of DuckDB is lesser than the mean execution time of SQLite

Table 3: Execution Times to retrieve data

DuckDB	SQLite
0.493854284286499,	1.0699973106384277,
0.4992682933807373,	0.758739709854126,
0.5188136100769043,	0.7583138942718506,
0.4812014102935791,	0.7659657001495361,
0.4714522361755371,	0.7500615119934082,
0.5156733989715576,	0.745466947555542,
0.49033069610595703,	0.7674193382263184,
0.4792938232421875,	0.7851073741912842,
0.4865450859069824,	0.7793111801147461,
0.5025525093078613,	0.7513875961303711,
0.500870943069458,	0.8165216445922852,
0.5076577663421631,	0.7650876045227051,
0.5416500568389893,	0.7829763889312744,
0.512653112411499,	0.7999231815338135,
0.48496460914611816,	0.8332321643829346,
0.5113584995269775,	0.7989082336425781,
0.511265754699707,	0.7677915096282959,
0.4970223903656006,	0.7999632358551025,
0.5268940925598145,	0.7328953742980957,
0.49007391929626465	0.8179125785827637

Table 4: Descriptive Statistics of retrieving data

Statistic	DuckDB	SQLite
Iterations (n)	20	20
Mean(s)	0.5011698246002197	0.792349123954773
Standard Deviation	0.017404786659907915	0.07056799370870627
Variance	0.00030292659867690847	0.004979841736072008

2.2. Test Results

- t – value : -17.91615089428377
- p – value : 1.997742172939328e-20

    - Since the p-value is less than the significance level, we reject the null hypothesis. Thus, considering the negative t-value as well, it indicates that the mean execution time of DuckDB is lesser than the mean execution time of SQLite.

2.3. Conclusion of the t-test

- The t-test results show that, to retrieve the data from a database, DuckDB is significantly faster than SQLite, based on the mean execution times collected during the experiment.

Discussion

- The uploading execution times of SQLAlchemy, pyodbc, polars and pandas were observed that these libraries took significantly longer, with SQLAlchemy taking 2.008030772209l6s, pyodbc taking 263.555660247802s, polars taking 22.4645318984985s and pandas taking 27.6370282173156s. Therefore, Duckdb and SQLite can be identified as the fastest libraries. According to the t-test results, DuckDB is significantly faster than SQLite when uploading data into a database and retrieving data from a database.

Conclusion

- The performance comparison of database interaction libraries revealed that DuckDB is the fastest option for uploading dataset into a database and for retrieving data from a database.