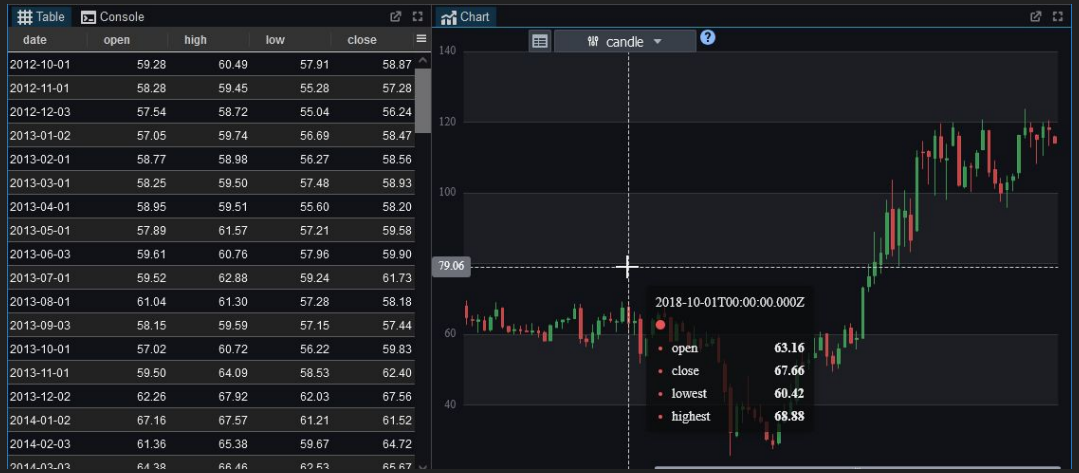
Stock Data Analysis with DuckDB

DuckCon #6 2025-01-31
Ryan Hamilton



TimeStored

Stock Data Analysis with DuckDB

1. Common finance data flows and analys
   a. Trade data
   b. Volume Weighted Average Price
   c. Pivots
   d. OHLC - Candle sticks
   e. Making Money with Window Function

2. QStudio + DuckDB = a powerful tool.

TimeStored

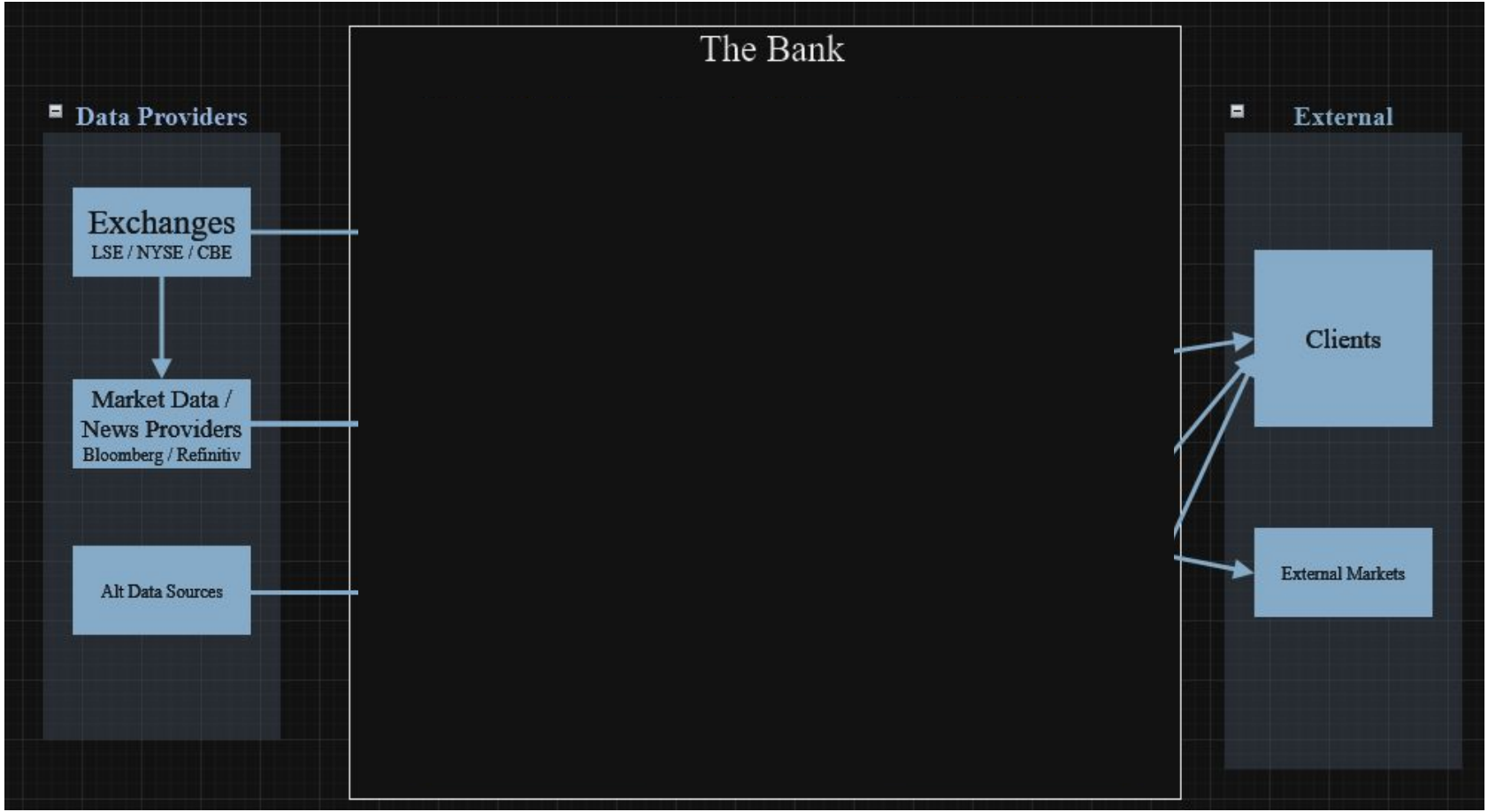
Ryan Hamilton

- 14 Years building large data platforms in banks to allow quants to perform stock data analysis.
  - Morgan Stanley, UBS, Citi.
- TimeStored - 2012
  - Data Analysis Tools
  - QStudio - Free SQL Client
  - Pulse - Data Dashboards

I’m super excited by DuckDB

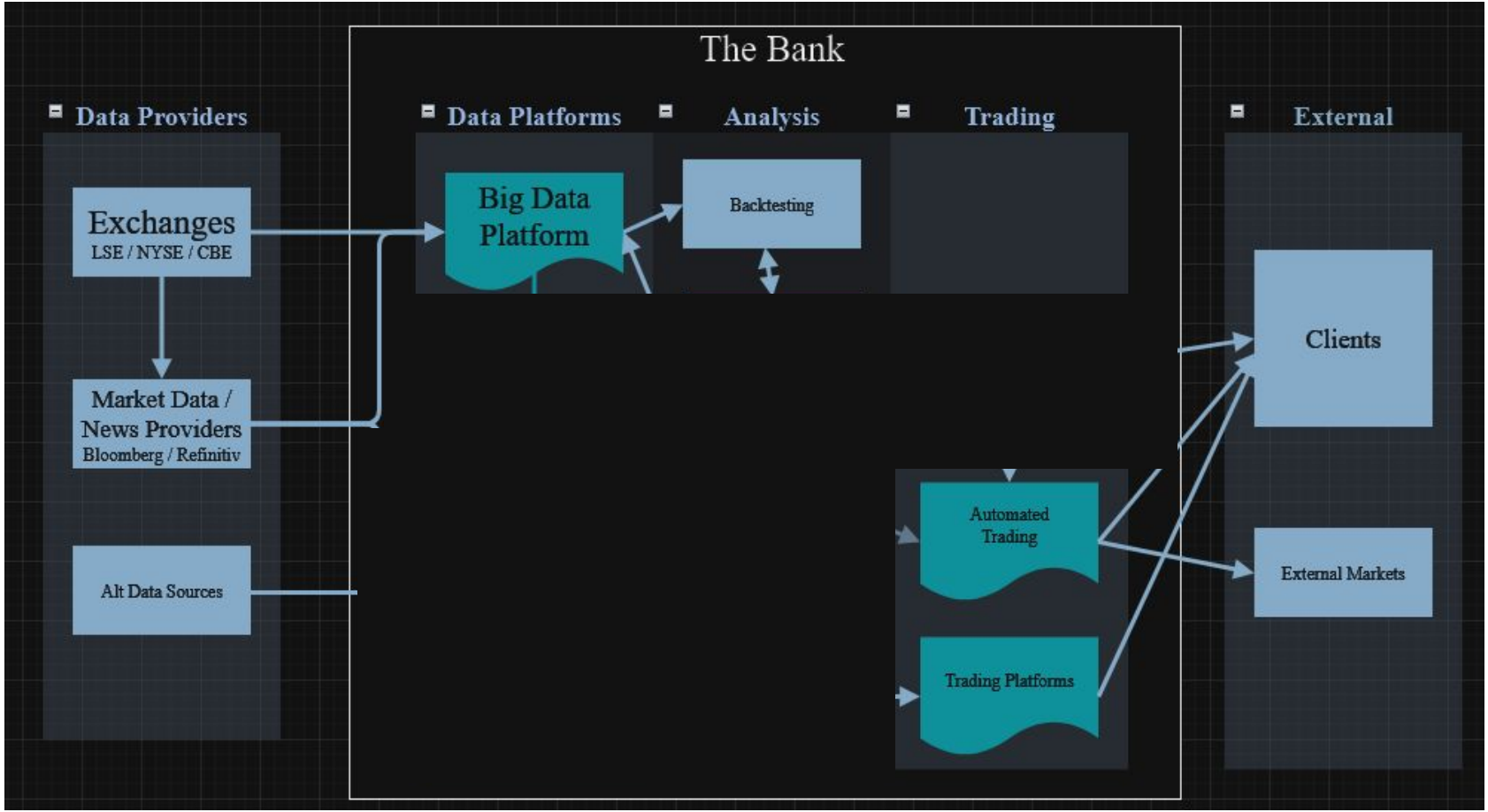
TimeStored

Simplified Firms Data Flow (Entirely Wrong)



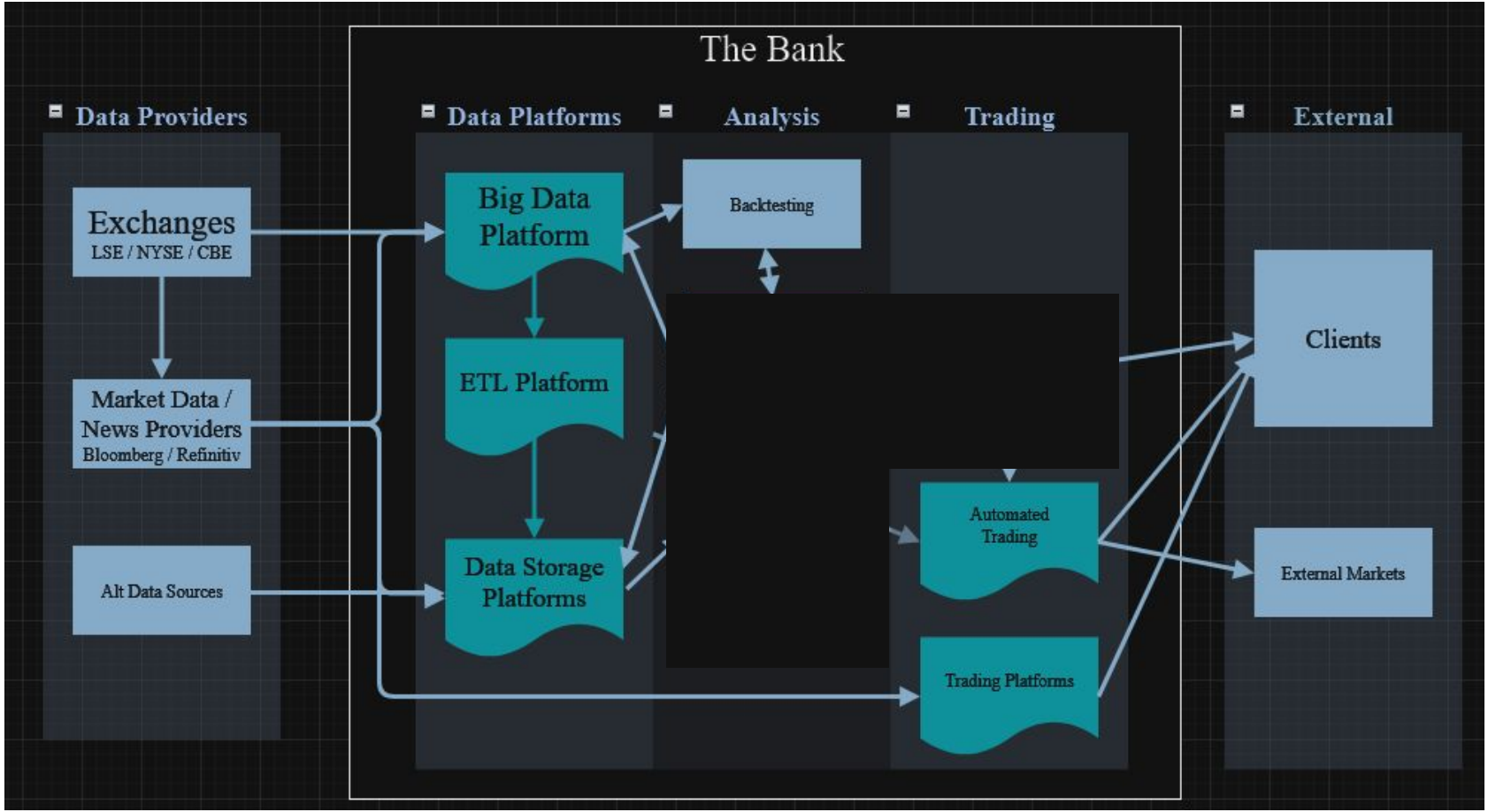
TimeStored

Simplified Firms Data Flow (Entirely Wrong)



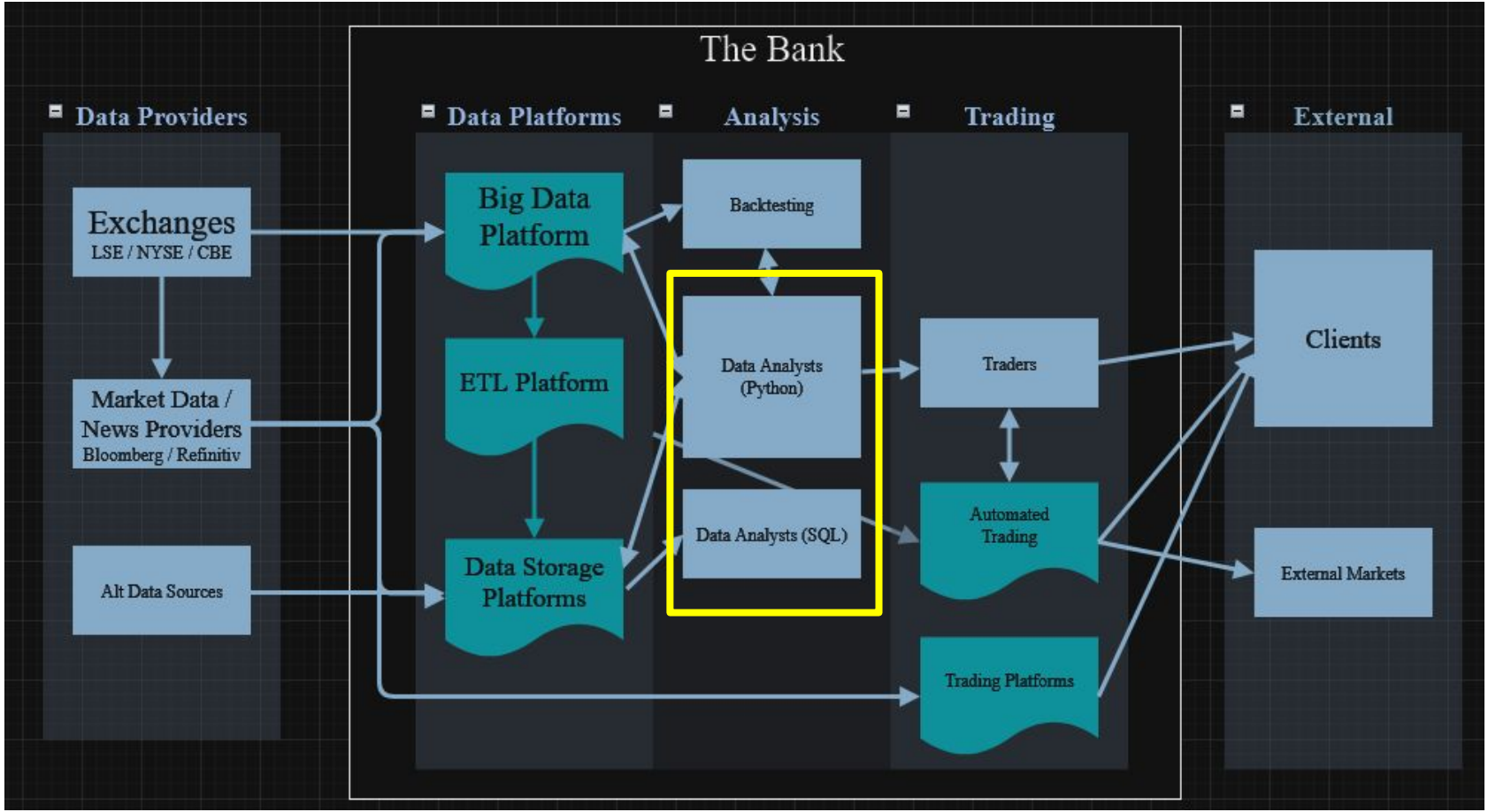
TimeStored

Simplified Firms Data Flow (Entirely Wrong)



TimeStored

Simplified Firms Data Flow (Entirely Wrong)

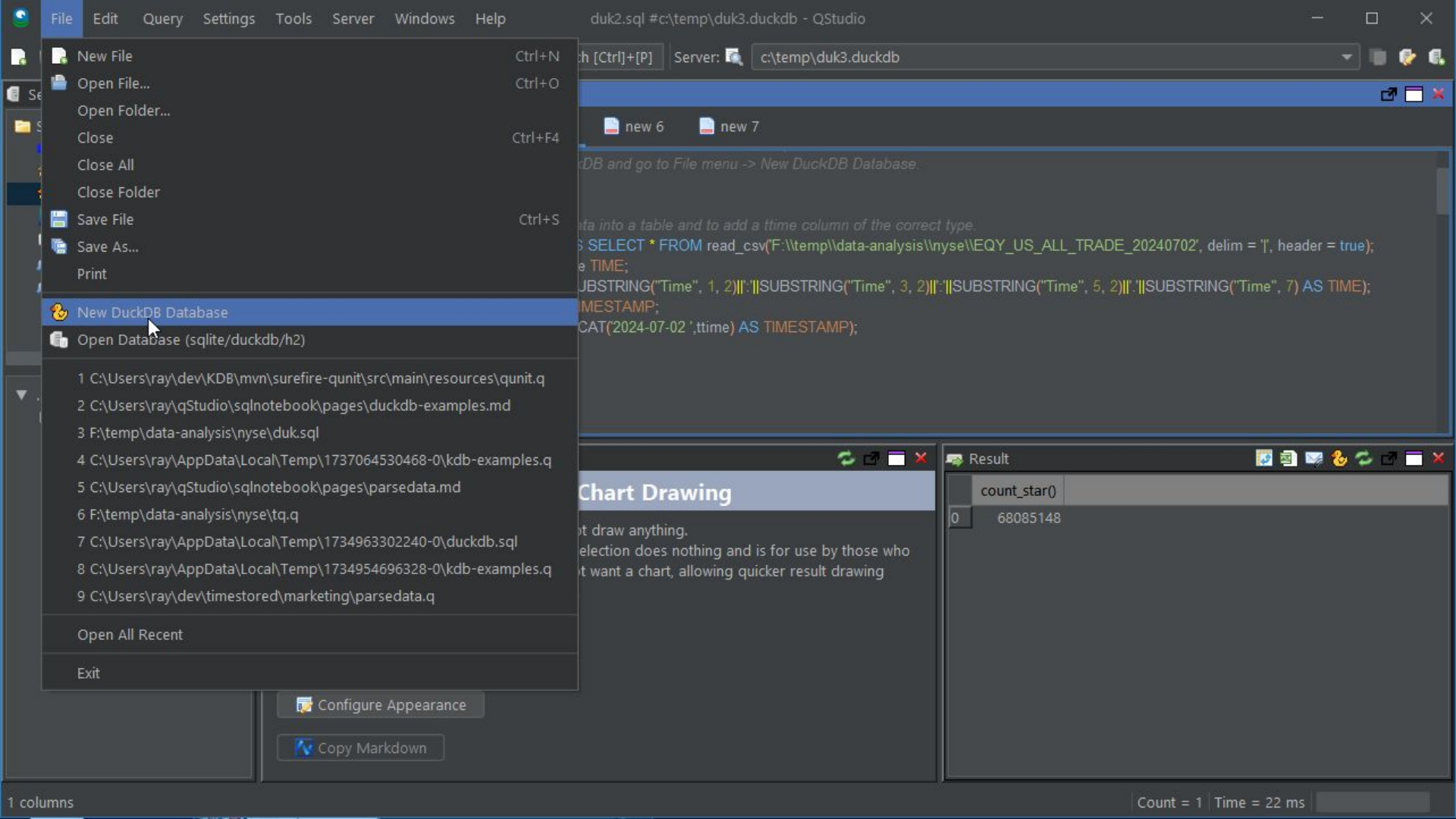


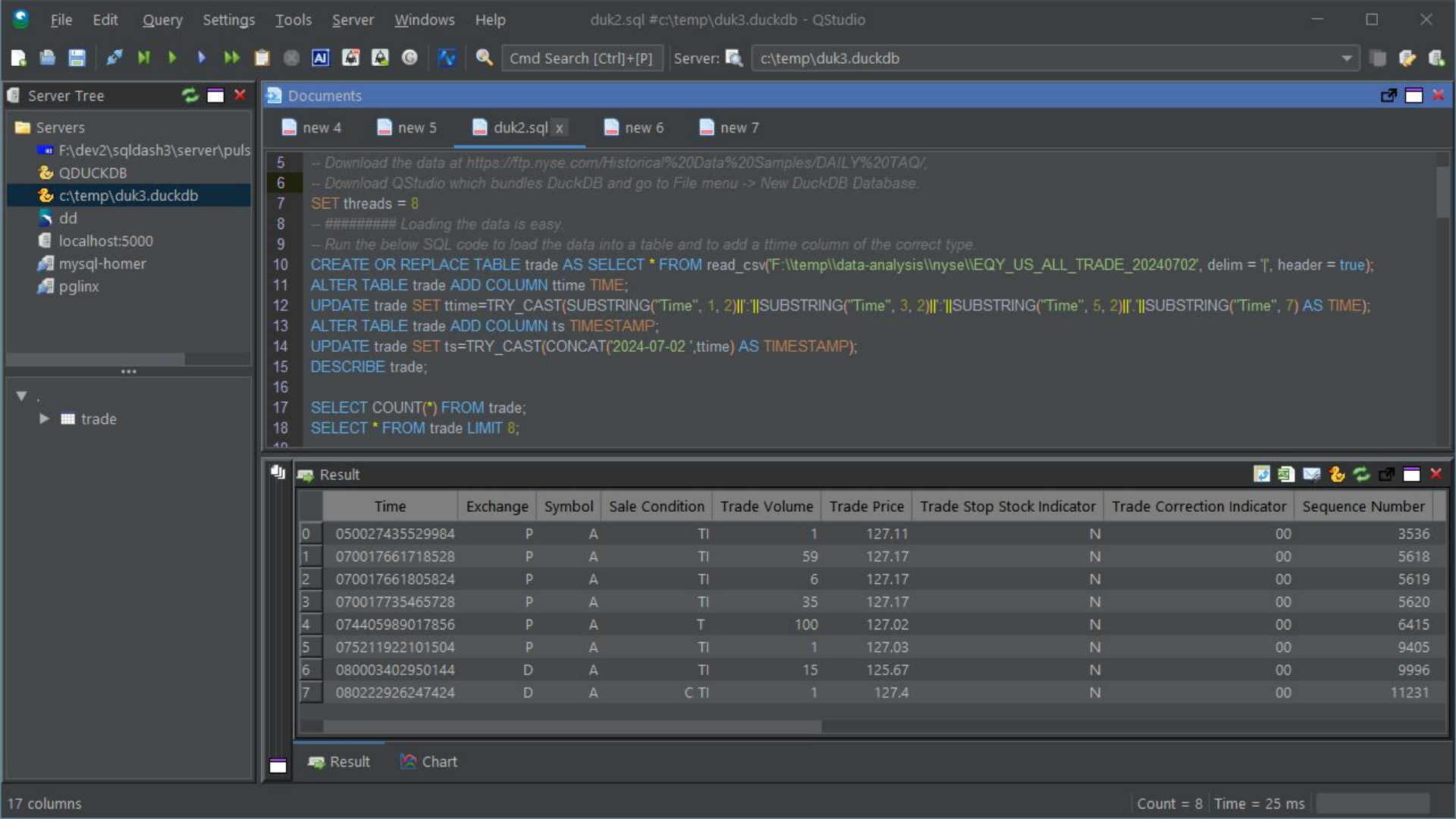
Finance Use Cases

1. H**istorical market data** storage and analysis.

2. **Local Quant analysis**
   e.g. Liquidity analysis, PnL analysis, profitability per client.

3. Real-time Streaming Calculation Engines
   e.g. Streaming VWAP, Streaming TCA


File
Edit
Query
Settings
Tools
Server
Windows
Help
duk2.sql #c:\temp\duk3.duckdb - QStudio
New File Ctrl+N
Open File... Ctrl+O
Open Folder...
Close Ctrl+F4
Close All
Close Folder
Save File Ctrl+S
Save As...
Print
New DuckDB Database
Open Database (sqlite/duckdb/h2)
1 C:\Users\ray\dev\KDB\mvn\surefire-qunit\src\main\resources\qunit.q
2 C:\Users\ray\qStudio\sqlnotebook\pages\duckdb-examples.md
3 F:\temp\data-analysis\nyse\duk.sql
4 C:\Users\ray\AppData\Local\Temp\1737064530468-0\kdb-examples.q
5 C:\Users\ray\qStudio\sqlnotebook\pages\parsedata.md
6 F:\temp\data-analysis\nyse\tq.q
7 C:\Users\ray\AppData\Local\Temp\1734963302240-0\duckdb.sql
8 C:\Users\ray\AppData\Local\Temp\1734954696328-0\kdb-examples.q
9 C:\Users\ray\dev\timestored\marketing\parsedata.q
Open All Recent
Exit
Server: c:\temp\duk3.duckdb
new 6
new 7
SELECT * FROM read_csv('F:\\temp\\data-analysis\\nyse\\EQY_US_ALL_TRADE_20240702', delim = '|', header = true);
Chart Drawing
Result
count_star()
68085148
Configure Appearance
Copy Markdown
1 columns
Count = 1
Time = 22 ms


File Edit Query Settings Tools Server Windows Help
duk2.sql #c:\temp\duk3.duckdb - QStudio
Cmd Search [Ctrl]+[P]
Server: c:\temp\duk3.duckdb
Server Tree
Servers
F:\dev2\sqldash3\server\puls
QDUCKDB
c:\temp\duk3.duckdb
dd
localhost:5000
mysql-homer
pglinx
trade
Documents
new 4
new 5
duk2.sql x
new 6
new 7
5 -- Download the data at https://ftp.nyse.com/Historical%20Data%20Samples/DAILY%20TAQ/,
6 -- Download QStudio which bundles DuckDB and go to File menu -> New DuckDB Database.
7 SET threads = 8
8 -- ######### Loading the data is easy.
9 -- Run the below SQL code to load the data into a table and to add a ttime column of the correct type.
10 CREATE OR REPLACE TABLE trade AS SELECT * FROM read_csv('F:\\temp\\data-analysis\\nyse\\EQY_US_ALL_TRADE_20240702', delim = '|', header = true);
11 ALTER TABLE trade ADD COLUMN ttime TIME;
12 UPDATE trade SET ttime=TRY_CAST(SUBSTRING("Time", 1, 2)||':'||SUBSTRING("Time", 3, 2)||':'||SUBSTRING("Time", 5, 2)||'.'||SUBSTRING("Time", 7) AS TIME);
13 ALTER TABLE trade ADD COLUMN ts TIMESTAMP;
14 UPDATE trade SET ts=TRY_CAST(CONCAT('2024-07-02 ',ttime) AS TIMESTAMP);
15 DESCRIBE trade;
16
17 SELECT COUNT(*) FROM trade;
18 SELECT * FROM trade LIMIT 8;
Result
Time Exchange Symbol Sale Condition Trade Volume Trade Price Trade Stop Stock Indicator Trade Correction Indicator Sequence Number
0 050027435529984 P A TI 1 127.11 N 00 3536
1 070017661718528 P A TI 59 127.17 N 00 5618
2 070017661805824 P A TI 6 127.17 N 00 5619
3 070017735465728 P A TI 35 127.17 N 00 5620
4 074405989017856 P A T 100 127.02 N 00 6415
5 075211922101504 P A TI 1 127.03 N 00 9405
6 080003402950144 D A TI 15 125.67 N 00 9996
7 080222926247424 D A C TI 1 127.4 N 00 11231
Result
Chart
17 columns
Count = 8
Time = 25 ms

File Edit Query Settings Tools Server Windows Help duk2.sql #c:\temp\duk3.duckdb - QStudio

Cmd Search [Ctrl]+[P] Server: c:\temp\duk3.duckdb

Server Tree

- Servers
  - F:\dev2\sqldash3\server\puls
  - QDUCKDB
  - c:\temp\duk3.duckdb
  - dd
  - localhost:5000
  - mysql-homer
  - pglinx
- trade

Documents

new 4 | new 5 | duk2.sql x | new 6 | new 7

```
29 select Symbol,avg("Trade Price") AS avg, sum("Trade Volume" * "Trade Price")/sum("Trade Volume") AS wavg FROM trade GROUP BY ALL ORDER BY Symbol DESC LIM
30 CREATE OR REPLACE function wavg(v,p) AS sum(v * p)/sum(v);
31 select Symbol,avg("Trade Price") AS avg,wavg("Trade Volume", "Trade Price") AS wavg FROM trade GROUP BY ALL ORDER BY Symbol DESC LIMIT 20;
32
```

Result

	Symbol	avg	wavg
0	ZZZ	24.3013412	24.3066933
1	ZYXI	9.0018773	9.0234232
2	ZYME	8.4303509	8.4109458
3	ZXZZ T	15.07717	13.1600393
4	ZXIET	100	100
5	ZWS	29.058342	29.1159304
6	ZVZZ T	27.1280548	26.7750549
7	ZVV	22.5333333	25.0034951
8	ZVSA	3.5965533	3.6101955
9	ZVRA	4.3755242	4.379931
10	ZVIA	0.6904188	0.6796599
11	ZURA W	0.5298787	0.5379482
12	ZURA	3.473992	3.4691229
13	ZUO	9.5514779	9.5808215
14	ZUMZ	18.9454589	18.9081403
15	ZTWO	49.8318824	49.8306898
16	ZTST	40.17	40.17
17	ZTS	174.583678	174.715989
18	ZTRE	49.765475	49.757258
19	ZTR	5.3983141	5.4026712

Chart

Control Panel

Type: Bar Chart Stacked

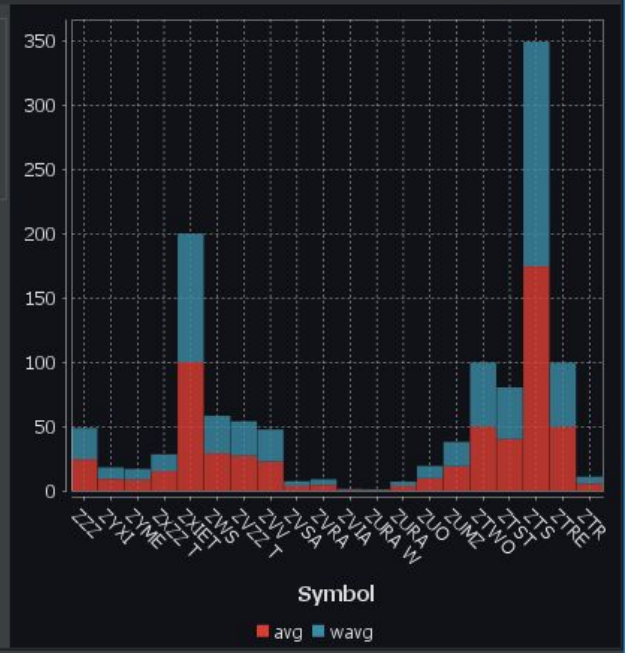
Theme: Dark

Render Very Large Charts

Open in New Window

Configure Appearance

Copy Markdown



3 columns

Count = 20 | Time = 472 ms

File Edit Query Settings Tools Server Windows Help duk2.sql #c:\temp\duk3.duckdb - QStudio

Cmd Search [Ctrl]+[P] Server: c:\temp\duk3.duckdb

Documents

new 4 new 5 duk2.sql x new 6 new 7

```
31 select Symbol,avg("Trade Price") AS avg,wavg("Trade Volume", "Trade Price") AS wavg FROM trade GROUP E
32
33 select Symbol,Exchange,"Sale Condition","Source Of Trade", COUNT(*) as num, 0.1+sum("Trade Volume") AS
34    avg("Trade Price") AS navg, sum("Trade Volume" * "Trade Price")/sum("Trade Volume") AS wavg
35    FROM trade
36    WHERE Symbol IN (SELECT Symbol FROM (SELECT Symbol,sum("Trade Volume") AS v FROM trade  GR
37    GROUP BY ALL ORDER BY Symbol;
38
39 -- ## Demo using PIVOT to drill up/down.    BY symbol -> BY symbol,exchange -> BY symbol -> BY symbol F
40
41
42 -- ######### Candlesticks are easy with window functions.
43
44 (SELECT  min(ts) as ttime,
```

Chart

Control Panel

Type: Bar Chart Stacked

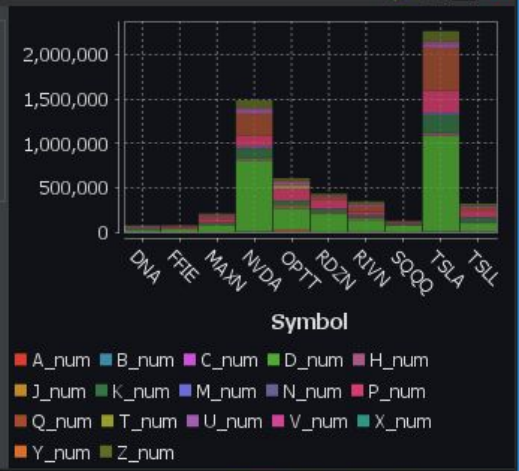
Theme: Dark

Render Very Large Charts

Open in New Window

Configure Appearance

Copy Markdown



Result

Select sum num By: Symbol Exchange Sale Condition Source of Trade Pivot On: Exchange Copy Query

Pulse Pivot

	Symbol	A_num	B_num	C_num	D_num	H_num	J_num	K_num	M_num	N_num	P_num	Q_num	T_num	U_num	V_num	X_num	Y_num	Z_num
0	DNA	140	692	772	23888	2634	972	2751	114	13325	7122		6307	3880	2691	225	1893	5978
1	FFIE	46	171	30	31566	1046	1875	5388	121	442	16279	8114		3997	392	15	2115	2969
2	MAXN	157	356	698	72325	4445	3200	24168	330	1948	39447	31955		11328	4044	42	2660	10431
3	NVDA	2661	4053	1930	796652	16685	6926	111607	1001	22764	119732	252076		46080	12768	2690	6301	81675
4	OPTT	19314	2005	851	239762	15109	21203	54543	365	2214	134566		38673	35027	2945	19	14106	26603
5	RDZN	155	1259	757	198681	9733	4595	42689	36	5420	92206	40613		11296	3330	638	4587	16230
6	RIVN	329	770	2136	128683	12904	7990	29778	280	8967	36598	60570		16345	6709	781	3276	27503
7	SQQQ	259	526	1553	69613	2884	3180	11331	23	3337	7519	11099		3439	1356	840	3507	4246
8	TSLA	2188	5901	1392	1075509	25148	7586	198849	2020	30298	247676	482790		44959	17442	3147	5648	116897
9	TSLL	413	1888	7060	89086	7681	3560	40361	31	9012	64656	37926		12486	2640	5679	4433	22170

18 columns Count = 10 Time = 1384 ms

File Edit Query Settings Tools Server Windows Help duk2.sql #c:\temp\duk3.duckdb - QStudio

Cmd Search [Ctrl]+[P] Server: c:\temp\duk3.duckdb

Documents

new 4 | new 5 | duk2.sql x | new 6 | new 7

```
39  -- ## Demo using PIVOT to drill up/down.    BY symbol -> BY symbol,exchange -> E
40
41
42  -- ######### Candlesticks are easy with window functions.
43
44  (SELECT  min(ts) as ttime,
45        min("Trade Price") as low, first("Trade Price") as open,
46        last("Trade Price") as close, max("Trade Price") as high,
47        sum("Trade Volume") AS volume
48           from trade where Symbol = 'AAPL'
49         GROUP BY time_bucket(INTERVAL '15 minutes', ts) ORDER BY ttime);
50
51  -- ######### How to make money
52
```

Chart

Control Panel

Type: Candlestick

Theme: Dark

Render Very Large Charts

Open in New Window

Configure Appearance

Copy Markdown

Result

	ttime	low	open	close	high	volume
0	2024-07-02T04:00:00.017646	215.89	216.6	216.35	216.6	14140
1	2024-07-02T04:15:10.548782	216.21	216.3	216.4	216.46	8298
2	2024-07-02T04:30:02.247220	216.1	216.39	216.3	216.42	9756
3	2024-07-02T04:45:17.664733	216	216.31	216.07	216.37	11817
4	2024-07-02T05:00:06.074584	216.02	216.07	216.2	216.31	2199
5	2024-07-02T05:15:00.161011	216.06	216.27	216.07	216.31	3117
6	2024-07-02T05:30:28.685559	215.51	216.08	215.69	216.08	11636
7	2024-07-02T05:45:22.793866	215.57	215.65	215.88	215.9	4720
8	2024-07-02T06:00:00.787625	215.84	215.88	215.9	215.92	1443
9	2024-07-02T06:15:33.575345	215.84	215.86	215.92	215.92	3354
10	2024-07-02T06:30:11.229606	215.84	215.9	216.03	216.06	6584

6 columns Count = 64 Time = 62 ms

File Edit Query Settings Tools Server Windows Help duk2.sql #c:\temp\duk3.duckdb - QStudio

Cmd Search [Ctrl]+[P] Server: c:\temp\duk3.duckdb

Documents

new 4 new 5 duk2.sql x new 6 new 7

```
58  (SELECT *,buy2*mv AS profit FROM
59     (SELECT *,lag(buy) OVER (ORDER BY ttime ASC) AS buy2,(lead(price) OVER (ORDER BY tti
60        (SELECT *, CASE WHEN mavg15 > price THEN 215 ELSE 217 END AS buy FROM
61           (SELECT ttime,price, avg(price) OVER(ORDER BY ttime ASC ROWS BETWEEN 15 PRE
62              FROM
63              (SELECT min(ttime) as ttime,
64                    avg("Trade Price") as price
65                 from trade where Symbol = 'AAPL'
66                    GROUP BY xbarTime(5,ttime) ORDER BY ttime)
67           )
68        )
69     )
70  )
71  );
```

Chart

Control Panel

Type: Time Series

Theme: Dark

Render Very Large Charts

Open in New Window

Configure Appearance

Copy Markdown

Result

Select sum price By: buy buy2 Pivot On: --NONE-- Copy Query

	ttime	price	mavg15	buy	buy2	mv	profit	sums
0	04:00:00.017646	216.2358242	216.2358242	217		-0.0758242		
1	04:05:05.834187	216.16	216.1979121	215	217	0.1184314	25.6996078	25.6996078
2	04:10:29.054737	216.2784314	216.2247518	217	215	0.0844258	18.1515406	43.8511485
3	04:15:10.548782	216.3628571	216.2592782	217	217	-0.0233333	-5.0633333	38.7878151
4	04:20:03.305255	216.3395238	216.2753273	217	217	-0.0087395	-1.8964706	36.8913445
5	04:25:02.460690	216.3307843	216.2845701	217	217	0.0306615	6.6535389	43.5448834
6	04:30:02.247220	216.3614458	216.2955524	217	217	-0.0896276	-19.4491895	24.0956939
7	04:35:13.165493	216.2718182	216.2925856	215	217	-0.0044498	-0.9655981	23.1300958
8	04:40:27.602094	216.2673684	216.2897837	215	215	-0.1011047	-21.7375072	1.3925886

8 columns

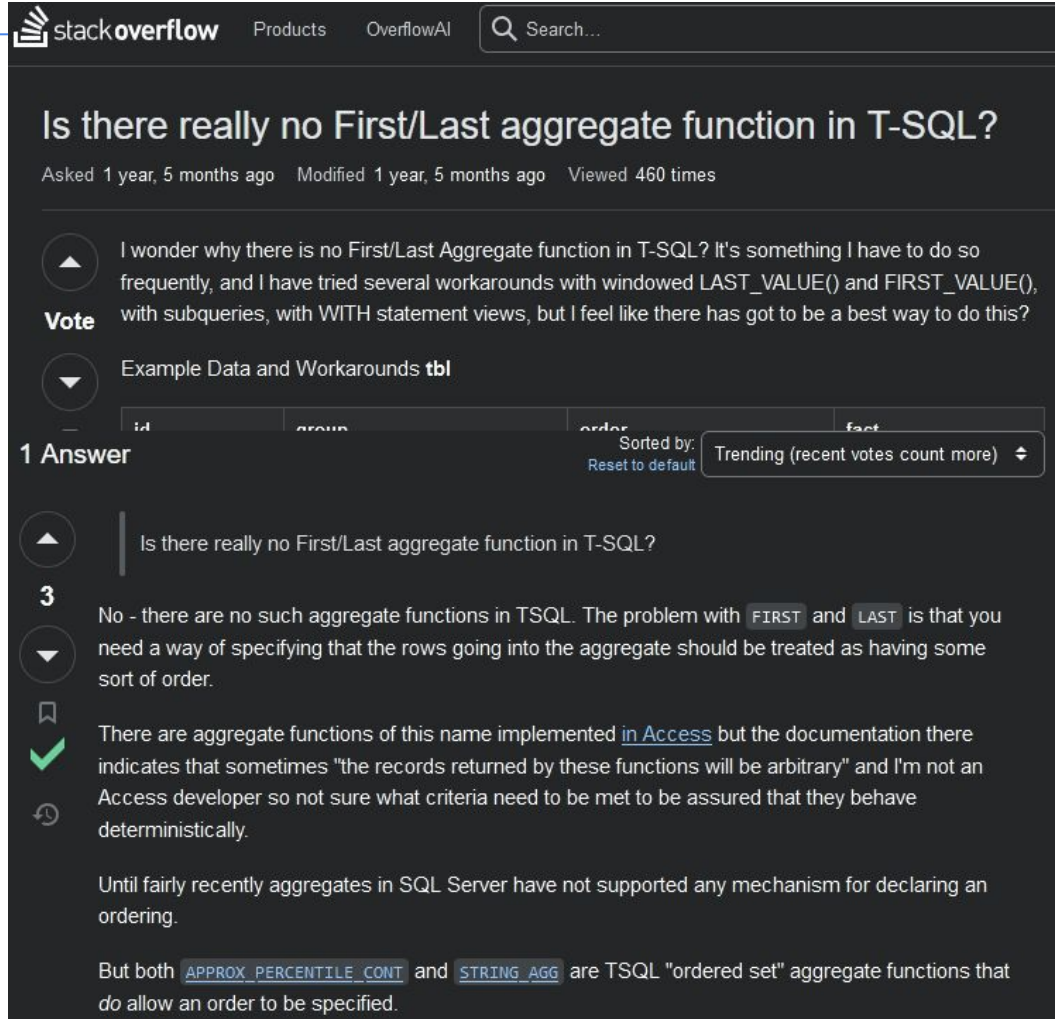
(SELECT *, CASE WHEN mavg15 > price THEN 215 EL...

TimeStored

Stock Data Analysis with DuckDB

1. Common finance data flows and analys
   a. Trade data
   b. Volume Weighted Average Price
   c. Pivots
   d. OHLC - Candle sticks
   e. Making Money with Window Function

2. QStudio + DuckDB = a powerful tool.
3. Hopefully showing those use-cases expla
   Why I'm excited about DuckDB.



stackoverflow Products OverflowAI Search...

Is there really no First/Last aggregate function in T-SQL?

Asked 1 year, 5 months ago Modified 1 year, 5 months ago Viewed 460 times

Vote

I wonder why there is no First/Last Aggregate function in T-SQL? It's something I have to do so frequently, and I have tried several workarounds with windowed LAST_VALUE() and FIRST_VALUE(), with subqueries, with WITH statement views, but I feel like there has got to be a best way to do this?

Example Data and Workarounds **tbl**

id	group	order	fact

1 Answer

Sorted by: Trending (recent votes count more)
Reset to default

3

> Is there really no First/Last aggregate function in T-SQL?

No - there are no such aggregate functions in TSQL. The problem with `FIRST` and `LAST` is that you need a way of specifying that the rows going into the aggregate should be treated as having some sort of order.

There are aggregate functions of this name implemented in Access but the documentation there indicates that sometimes "the records returned by these functions will be arbitrary" and I'm not an Access developer so not sure what criteria need to be met to be assured that they behave deterministically.

Until fairly recently aggregates in SQL Server have not supported any mechanism for declaring an ordering.

But both `APPROX_PERCENTILE_CONT` and `STRING_AGG` are TSQL "ordered set" aggregate functions that *do* allow an order to be specified.

Home
Questions
Staging Ground
Tags
Saves
Users
Companies
LABS

SQL Query to Join Two Tables Based Off Closest Timestamp

Asked 14 years, 3 months ago Modified 3 years, 8 months ago Viewed 39k times

26

I have two tables in SQL and I need to be able to do a join based off of the timestamp in table B that is earlier than or equal to the timestamp in table A.

So, here is some fake data for two tables and the desired output:

Closed Cases (Table A)

```
| id | resolution |         timestamp          |
-----------------------------------------------
|  1 |     solved | 2006-10-05 11:55:44.888153 |
|  2 |     closed | 2007-10-07 12:34:17.033498 |
|  3 |    trashed | 2008-10-09 08:19:36.983747 |
```

2 Answers

Sorted by: Reset to default Trending (recent votes count more)

8

To join each rows in `table1` to the row with with the next higher timestamp. *Exactly* one row in the result per row on `table1`:

```
SELECT *
FROM   table1 t1
LEFT   JOIN LATERAL (
   SELECT *
   FROM   table2 t2
   WHERE  t2.timee >= t1.dateee
   ORDER  BY t2.timee
   LIMIT  1
   ) ON TRUE;
```

An index on `(timee)` is essential for performance.

Share Edit Follow Flag

edited Jun 20, 2020 at 9:12 Community Bot 1 • 1

answered Mar 3, 2015 at 19:21 Erwin Brandstetter 656k • 157 • 1.1k • 1.3k

17

If you can make changes to the table structures, I recommend changing the classification table to include an end date as well as a start date - it will be much easier to join to the table that way.

If not, I suggest the following:

```
SELECT case.id, case.resolution, case.timestamp, class.value
  FROM closed_cases AS case
  LEFT JOIN (select c.*,
                    (select min(timestamp)
                     from classifications c1
                      where c1.timestamp > c.timestamp) timeend
             from classifications c) AS class
  ON case.timestamp >= class.timestamp and
     (case.timestamp < class.timeend or class.timeend IS NULL)
  WHERE case.timestamp BETWEEN $1 AND $2;
```

Why I’m super excited about DuckDB.

1. It’s **Free**.
2. **Faster** than the mainstream alternatives
3. **Easier** to run and cross-platform.
4. **Improving** more quickly.
5. Awesome python integration.
6. ASOF joins, first/last, nested arrays……

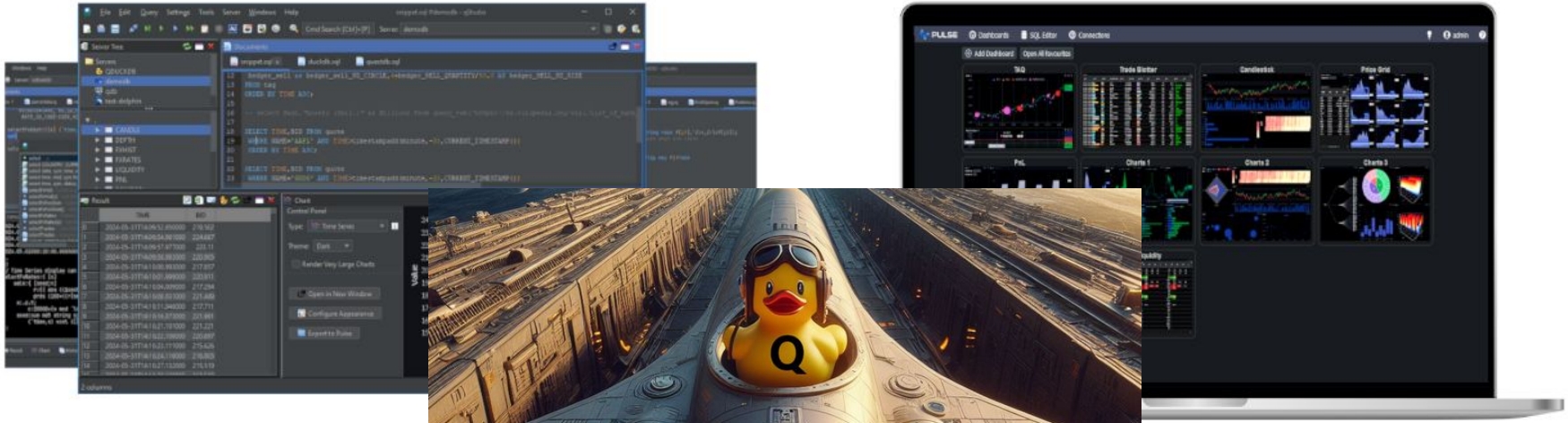
I’ve been waiting over a decade
for a fast column-oriented database that
would solve similar problems in finance.

Essential tools for data analysts

Since 2013 we have provided the tools so that you can you can focus on your business.

qStudio

PULSE



STOP. Thanks. Slido.

Arrow Import

The following demonstrates consuming an Arrow stream from the Java Arrow bindings.

```java
import org.apache.arrow.memory.RootAllocator;
import org.apache.arrow.vector.ipc.ArrowReader;
import org.duckdb.DuckDBConnection;

// Arrow binding
try (var allocator = new RootAllocator();
     ArrowStreamReader reader = null; // should not be null of course
     var arrow_array_stream = ArrowArrayStream.allocateNew(allocator)) {
    Data.exportArrayStream(allocator, reader, arrow_array_stream);

    // DuckDB setup
    try (var conn = (DuckDBConnection) DriverManager.getConnection("jdbc:duckdb:")) {
        conn.registerArrowStream("asdf", arrow_array_stream);

        // run a query
        try (var stmt = conn.createStatement();
             var rs = (DuckDBResultSet) stmt.executeQuery("SELECT count(*) FROM asdf")) {
            while (rs.next()) {
                System.out.println(rs.getInt(1));
            }
        }
    }
}
```