

Stock Portfolio Performance Tracker (SQL Project)

<https://github.com/Rohitk45k>

By Rohit Kamble

Tools Used: MySQL, GitHub

Problem Statement

- Analyze stock transactions to understand:
 - - User investment patterns
 - - Stock performance
 - - Data inconsistencies
 - - Buy vs Sell behavior

Dataset Overview

- - Rows: 4,000+
- - Fields:
 - transaction_id
 - user_id
 - stock_symbol
 - transaction_type (Buy/Sell)
 - quantity
 - price_per_share
 - transaction_date

Project Objectives

- 1. Total investment by user
- 2. Most traded stock
- 3. Highest average sell price
- 4. Users who only buy
- 5. Monthly transaction trends
- 6. Users trading in >5 stocks
- 7. Stocks only sold (data issue)

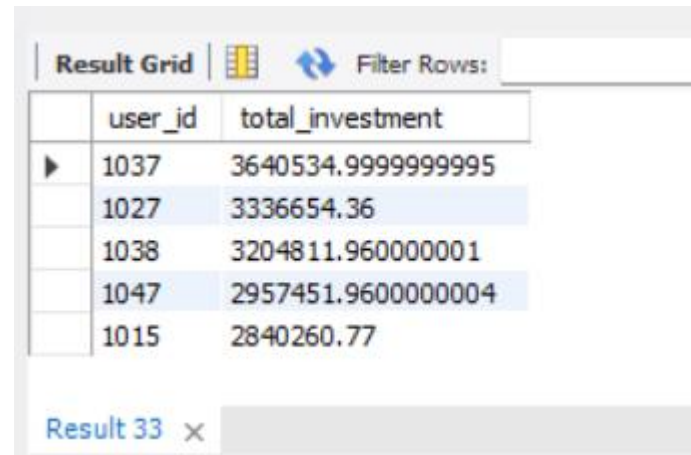
Total number of Buy and Sell transactions per use

```
SELECT  
    transaction_type, COUNT(user_id)  
FROM  
    stock  
GROUP BY transaction_type;
```

Result Grid			Filter Rows:
	transaction_type	COUNT(user_id)	
▶	Buy	2032	
	Sell	1968	

What is the total investment amount (Buy) for each user?

```
SELECT
    user_id, SUM(quantity * price_per_share) AS total_investment
FROM
    stock
WHERE
    transaction_type = 'Buy'
GROUP BY user_id
ORDER BY total_investment DESC
limit 5;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of the SQL query, sorted by total investment in descending order. The first row is highlighted with a mouse cursor. At the bottom, there is a tab labeled 'Result 33' with a close button.

	user_id	total_investment
▶	1037	3640534.9999999995
	1027	3336654.36
	1038	3204811.9600000001
	1047	2957451.9600000004
	1015	2840260.77

Result 33 ×

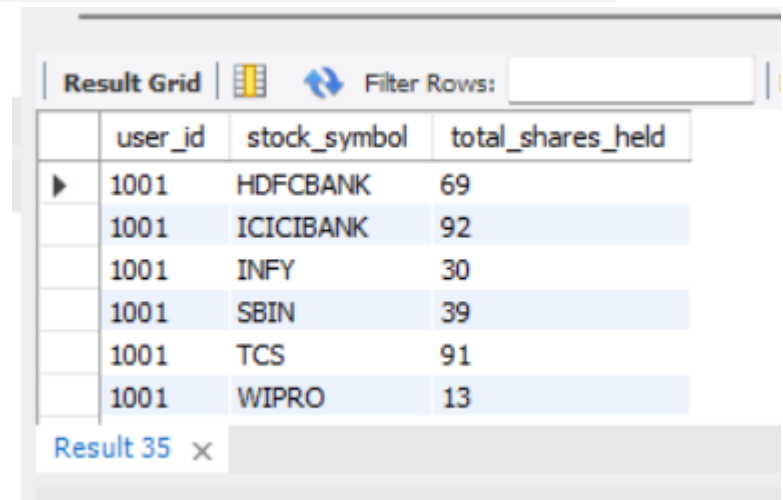
What is the average price per share for each stock when bought?

```
28 • SELECT
29     stock_symbol, ROUND(AVG(price_per_share), 2) AS avg_price
30 FROM
31     stock
32 WHERE
33     transaction_type = 'Sell'
34 GROUP BY stock_symbol;
35
```

Result Grid			Filter Rows:
	stock_symbol	avg_price	
▶	ICICIBANK	2322.22	
	SBIN	2093.15	
	HDFCBANK	2184.6	
	INFY	2090.32	
	HCLTECH	2321.71	
	TCS	2270.36	
Result 34			×

What is the total quantity of shares held by each user for each stock (Buys - Sells)?

- SELECT
 user_id,
 stock_symbol,
 SUM(CASE WHEN transaction_type = 'Buy' THEN quantity ELSE 0 END) -
 SUM(CASE WHEN transaction_type = 'Sell' THEN quantity ELSE 0 END) AS total_shares_held
FROM
 stock
GROUP BY
 user_id, stock_symbol
HAVING
 total_shares_held > 0
ORDER BY
 user_id, stock_symbol;



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with three columns: 'user_id', 'stock_symbol', and 'total_shares_held'. There are six rows of data, all for user_id 1001, representing different stocks and their respective share counts. The interface includes a 'Filter Rows' input field and a 'Result 35' indicator at the bottom.

	user_id	stock_symbol	total_shares_held
▶	1001	HDFCBANK	69
	1001	ICICIBANK	92
	1001	INFY	30
	1001	SBIN	39
	1001	TCS	91
	1001	WIPRO	13

Result 35 ×

List all users who have only Buy transactions and no Sell transactions

```
select * from stock  
where transaction_type= "Buy";
```

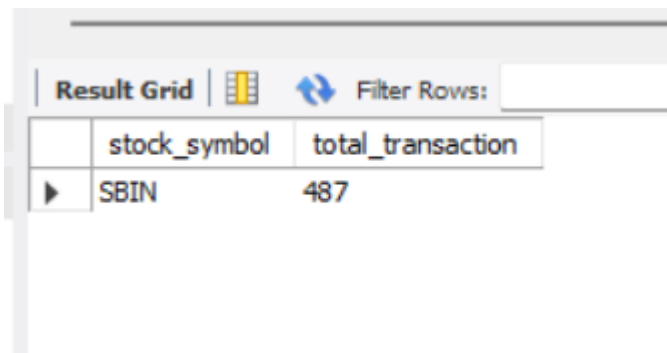
Result Grid							
		Filter Rows:		Export:	Wrap Cell Content:		Fetch rows:
	transaction_id	user_id	stock_symbol	transaction_type	quantity	price_per_share	transaction_date
▶	1	1025	ICICIBANK	Buy	17	3879.13	2024-10-17
	3	1014	HCLTECH	Buy	19	989.11	2024-08-11
	4	1007	WIPRO	Buy	20	845.67	2024-11-22
	6	1028	SBIN	Buy	36	2169.53	2023-02-01
	7	1034	WIPRO	Buy	36	3705.98	2023-07-21
	8	1006	ICICIBANK	Buy	40	2227.52	2024-02-02

stock 36 ×

Output

What is the most traded stock by number of transactions?

- ```
SELECT
 stock_symbol, COUNT(*) AS total_transaction
FROM
 stock
GROUP BY stock_symbol
ORDER BY total_transaction DESC
LIMIT 1;
```

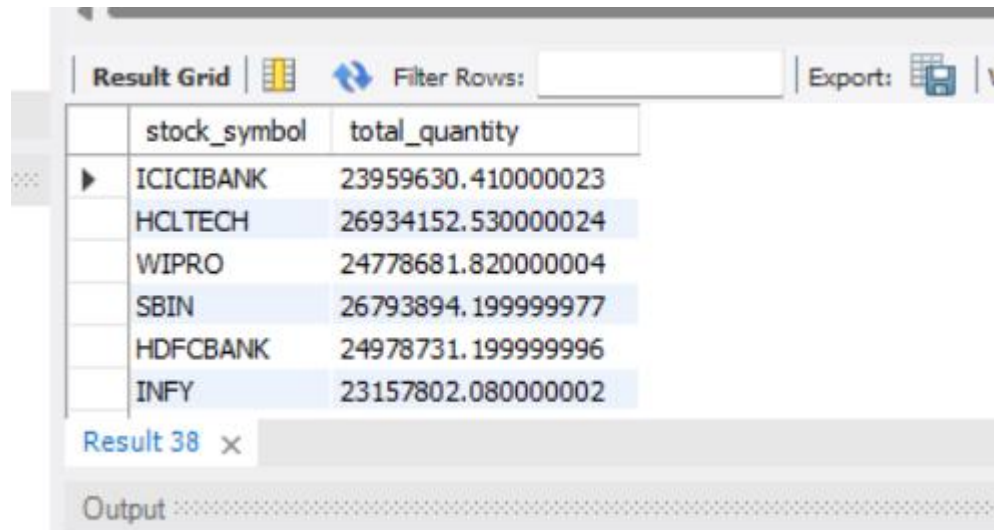


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'stock\_symbol' and 'total\_transaction'. The first row of data shows 'SBIN' with a value of 487. Above the table, there is a 'Filter Rows:' input field and a double-headed arrow icon.

|   | stock_symbol | total_transaction |
|---|--------------|-------------------|
| ▶ | SBIN         | 487               |

# What is the total quantity sold for each stock?

```
SELECT
 stock_symbol,
 SUM(quantity * price_per_share) AS total_quantity
FROM
 stock
GROUP BY stock_symbol;
```



The screenshot shows a database query result grid with two columns: 'stock\_symbol' and 'total\_quantity'. The results are listed for six different stock symbols. The interface includes a 'Result Grid' tab, a 'Filter Rows' input field, and an 'Export' button. The results are displayed in a table with alternating row colors (white and light blue). Below the table, there is a 'Result 38' label and an 'Output' section.

| stock_symbol | total_quantity     |
|--------------|--------------------|
| ICICIBANK    | 23959630.410000023 |
| HCLTECH      | 26934152.530000024 |
| WIPRO        | 24778681.820000004 |
| SBIN         | 26793894.199999977 |
| HDFCBANK     | 24978731.199999996 |
| INFY         | 23157802.080000002 |

# Which stock has the highest average sell price?

```
80
81 • SELECT
82 stock_symbol, ROUND(AVG(price_per_share), 2) AS avg_price
83 FROM
84 stock
85 GROUP BY stock_symbol
86 ORDER BY avg_price DESC
87 LIMIT 1;
88
```

91

Result Grid  Filter Rows:

|   | stock_symbol | avg_price |
|---|--------------|-----------|
| ▶ | HCLTECH      | 2320.53   |

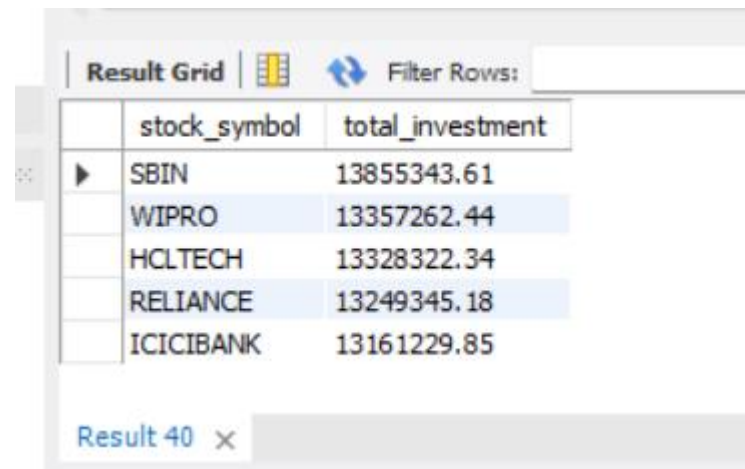
# Which user has made the highest number of transactions?

```
126 • SELECT
127 user_id, COUNT(*) AS total_transaction
128 FROM
129 stock
130 GROUP BY user_id
131 ORDER BY total_transaction DESC
132 LIMIT 1;
133
134
```

| Result Grid |         |                   | Filter Rows: |
|-------------|---------|-------------------|--------------|
|             | user_id | total_transaction |              |
| ▶           | 1007    | 96                |              |

# Find the top 5 stocks with the highest total investment value.4

```
SELECT
 stock_symbol,
 ROUND(SUM(quantity * price_per_share), 2) AS total_investment
FROM
 stock
WHERE
 transaction_type = 'Buy'
GROUP BY
 stock_symbol
ORDER BY
 total_investment DESC
LIMIT 5;
```



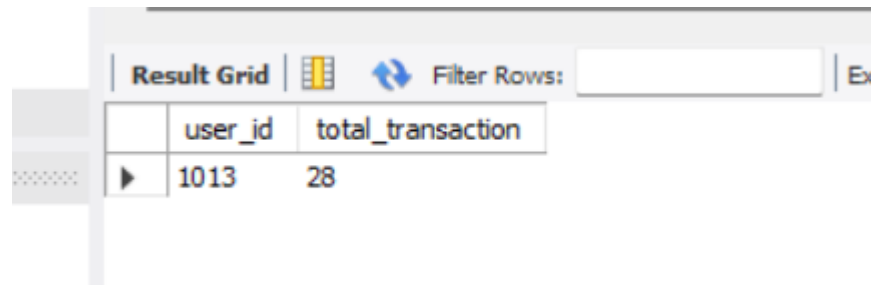
The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of the SQL query, showing the top 5 stocks by total investment. The columns are 'stock\_symbol' and 'total\_investment'. The results are as follows:

|   | stock_symbol | total_investment |
|---|--------------|------------------|
| ▶ | SBIN         | 13855343.61      |
|   | WIPRO        | 13357262.44      |
|   | HCLTECH      | 13328322.34      |
|   | RELIANCE     | 13249345.18      |
|   | ICICIBANK    | 13161229.85      |

At the bottom of the window, it says 'Result 40' with a close button (X).

# Which user has invested the most overall (in Buy transactions)?

- ```
SELECT
    user_id, COUNT(*) AS total_transaction
FROM
    stock
WHERE
    transaction_type = 'Buy'
GROUP BY user_id
ORDER BY total_transaction
LIMIT 1;
```

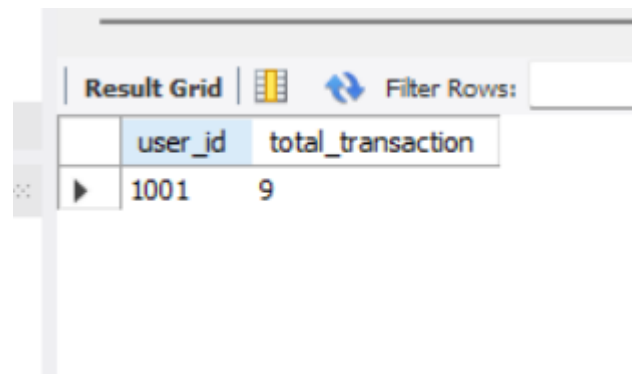


The screenshot shows a database interface with a 'Result Grid' tab. It contains a single row of data representing the user with the highest total transaction count.

	user_id	total_transaction
▶	1013	28

Which users have traded in more than 5 different stocks?

```
152 • SELECT
153     user_id, COUNT(DISTINCT stock_symbol) AS total_transaction
154 FROM
155     stock
156 GROUP BY user_id
157 HAVING COUNT(DISTINCT stock_symbol) > 5
158 ORDER BY total_transaction
159 LIMIT 1;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid has two columns: 'user_id' and 'total_transaction'. The first row of data shows 'user_id' as 1001 and 'total_transaction' as 9. There is a 'Filter Rows' button and a search input field at the top right of the grid.

	user_id	total_transaction
▶	1001	9

How many users started trading in 2023 and continued in 2024?

```
189
190 • SELECT COUNT(DISTINCT user_id) AS users_traded_both_years
191 FROM stock
192 WHERE user_id IN (
193     SELECT user_id
194     FROM stock
195     GROUP BY user_id
196     HAVING
197         MIN(YEAR(transaction_date)) = 2023
198         AND SUM(CASE WHEN YEAR(transaction_date) = 2024 THEN 1 ELSE 0 END) > 0
199 );
200
201
```

Result Grid	
Filter Rows:	
users_traded_both_years	
▶	50

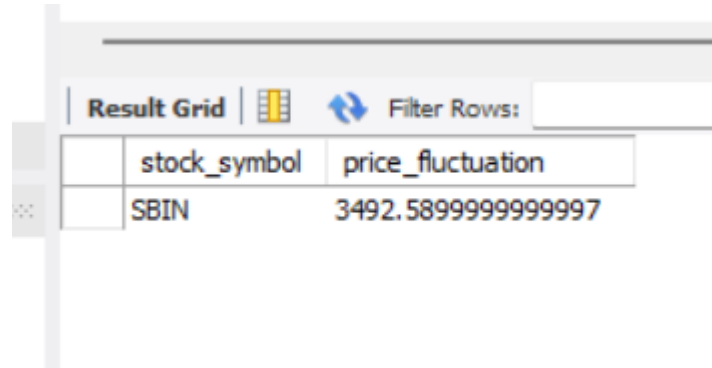
What is the **peak trading month** with the highest volume?

```
202 • SELECT
203     DATE_FORMAT(transaction_date, '%Y-%m') AS trading_month,
204     SUM(quantity) AS total_traded_quantity
205 FROM
206     stock
207 GROUP BY
208     trading_month
209 ORDER BY
210     total_traded_quantity DESC
211 LIMIT 1;
```

Result Grid			Filter Rows:	
	trading_month	total_traded_quantity		
▶	2023-01	4810	4810	

Which stock had the largest price fluctuation (max - min price) in the dataset?

```
214
215 • SELECT
216     stock_symbol,
217     MAX(price_per_share) - MIN(price_per_share) AS price_fluctuation
218 FROM
219     stock
220 GROUP BY
221     stock_symbol
222 ORDER BY
223     price_fluctuation DESC
224 LIMIT 1;
225
```





The screenshot shows a database interface with a 'Result Grid' tab. It contains a single row of data for the stock 'SBIN' with a price fluctuation of 3492.5899999999997. Above the grid, there is a 'Filter Rows' section with a search icon and an empty input field.


stock_symbol	price_fluctuation
SBIN	3492.5899999999997

List all users who traded stocks with a price above ₹3500 more than twice.

```
226 • SELECT
227     user_id,
228     COUNT(*) AS high_price_trades
229 FROM
230     stock
231 WHERE
232     price_per_share > 3500
233 GROUP BY
234     user_id
235 HAVING
236     COUNT(*) > 2;
237
```

Result Grid |   Filter Rows:

	user_id	high_price_trades
▶	1025	13
	1034	10
	1022	19
	1029	11
	1008	15
	1018	10

Result 48 × 

Output