Querying PSO data: A SQL Approach

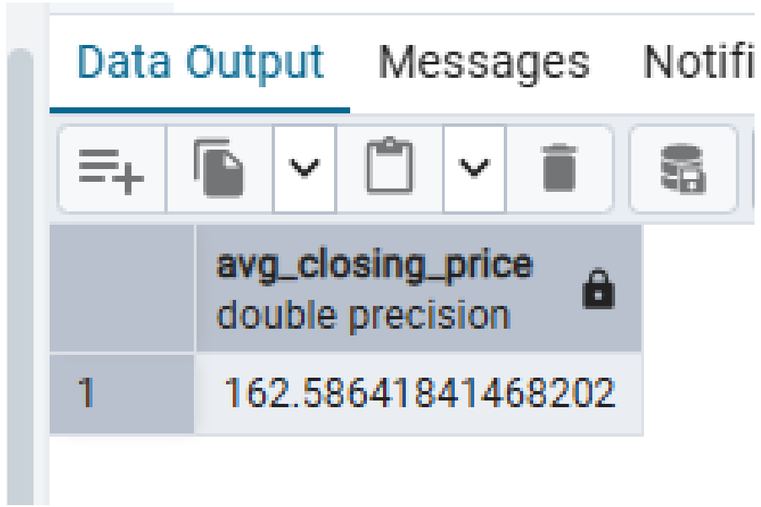
Queries & Outputs

Average Closing Price

Query

```
SELECT avg(close) AS avg_closing_price
  FROM stock_data;
```

Query_Output



📈 Daily Price Change

Query

```
SELECT date,
    open,
    close,
    close - open AS daily_change
  FROM stock_data;
```

Query_Output

Data Output | Messages | Notifications

Showing rows: 1 to 10

	date date	open double precision	close double precision	daily_change double precision
1	2008-01-02	145.370071	141.387833	-3.9822379999999953
2	2008-01-03	145.771957	148.45723	2.6852730000000236
3	2008-01-04	150.338745	152.987488	2.6487430000000245
4	2008-01-07	152.165466	150.539688	-1.6257779999999968
5	2008-01-08	150.539688	152.001053	1.4613650000000007
6	2008-01-09	152.001053	151.982788	-0.018265000000013742
7	2008-01-10	151.982788	152.71347	0.7306820000000016
8	2008-01-11	152.71347	151.800125	-0.9133449999999925

Daily Volatility (High - Low)

Query

```
SELECT date,
    high - low AS daily_volatility
  FROM stock_data;
```

Query_Output

	date date	daily_volatility double precision
1	2008-01-02	6.356979999999993
2	2008-01-03	2.6852730000000236
3	2008-01-04	2.6487430000000245
4	2008-01-07	3.2880849999999953
5	2008-01-08	3.1967620000000068
6	2008-01-09	2.2834020000000237
7	2008-01-10	3.2880859999999927
8	2008-01-11	4.347594999999984
9	2008-01-14	3.6168969999999945
10	2008-01-15	2.557403999999991

Lowest and Highest Trading Days

Query

```
SELECT min(low) AS lowest_trading_day,
    max(high) AS highest_trading_day
   FROM stock_data;
```

Query_Output

Data Output | Messages | Notifications

	lowest_trading_day double precision	highest_trading_day double precision
1	33.648113	285.590271

⏳ Longest Increasing Adj Close Streak (Concept)

Query part:1

```
WITH previous_prices AS (
         SELECT stock_data.date,
            stock_data.adj_close,
            lag(stock_data.adj_close) OVER (ORDER BY stock_data.date) AS prev_adj_close
           FROM stock_data
        ), growth_check AS (
         SELECT previous_prices.date,
            previous_prices.adj_close,
            previous_prices.prev_adj_close,
                CASE
                    WHEN previous_prices.adj_close > previous_prices.prev_adj_close THEN 1
                    ELSE 0
                END AS is_increasing
           FROM previous_prices
        ), streak_groups AS (
         SELECT growth_check.date,
            growth_check.adj_close,
            growth_check.prev_adj_close,
            growth_check.is_increasing,
            sum(
                CASE
                    WHEN growth_check.is_increasing = 0 THEN 1
                    ELSE 0
                END) OVER (ORDER BY growth_check.date) AS group_id
           FROM growth_check
```

Query part:2

```
        ), group_lengths AS (
         SELECT streak_groups.group_id,
            count(*) AS streak_length
           FROM streak_groups
          WHERE streak_groups.is_increasing = 1
          GROUP BY streak_groups.group_id
        )
SELECT max(streak_length) AS longest_increasing_streak
  FROM group_lengths;
```

Query_Output

Data Output | Messages | Explain

	longest_increasing_streak bigint
1	8

Monthly Average Close

Query

```
SELECT EXTRACT(month FROM date) AS month,
   EXTRACT(year FROM date) AS year,
   avg(close) AS avg_monthly_close
  FROM stock_data
 GROUP BY (EXTRACT(year FROM date)), (EXTRACT(month FROM date))
 ORDER BY (EXTRACT(year FROM date)), (EXTRACT(month FROM date));
```

Query_Output

	month numeric	year numeric	avg_monthly_close double precision
1	1	2008	154.083515
2	2	2008	168.040538
3	3	2008	190.43790815000003
4	4	2008	188.10029186363636

📦 5-Day Moving Average of Close Price

Query

```
SELECT date,
   close,
   avg(close) OVER (ORDER BY date ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) AS moving_avg_5_day
  FROM stock_data;
```

Query_Output

	date date	close double precision	moving_avg_5_day double precision
1	2008-01-02	141.387833	141.387833
2	2008-01-03	148.45723	144.9225315
3	2008-01-04	152.987488	147.61085033333333
4	2008-01-07	150.539688	148.34305975
5	2008-01-08	152.001053	149.0746584

Top 5 Highest Volume Days

Query

```
SELECT date,
    volume
  FROM stock_data
 ORDER BY volume DESC
LIMIT 5;
```

Query_Output

Data Output | Messages | Exp

	date date	volume bigint
1	2008-03-13	18996929
2	2008-03-17	18347403
3	2008-01-29	17138403
4	2013-07-10	16993912
5	2013-05-23	16962549