Stock Market Data Analysis Using SQL

Project Description

- This project focuses on analyzing stock market data using SQL. The dataset contains information about various companies, including their sector, stock prices over time, and dividend records. By leveraging SQL queries, multiple business questions were answered to uncover meaningful insights.

- **Key objectives of this project include:**

•Retrieving company information by sector and date ranges.

•Calculating average, moving average, and standard deviation of stock prices.

•Identifying companies with specific trading behavior, such as:

1. Continuous decline in closing prices.
2. High difference between open and close prices.

•Ranking companies within sectors based on their performance.

•Finding companies that have issued the highest or no dividends.

•Creating views for efficient query operations.

- The project also demonstrates the use of:

1. SQL functions like AVG(), STDDEV(), LAG(), and RANK().
2. Window functions and subqueries.
3. Views for simplified access to combined data.
4. Some of the advanced queries were created with the assistance of AI tools (ChatGPT), and were thoroughly studied and understood before implementation.

Company List

- List all companies along with their sectors.

Input

```sql
SELECT
    name, sector
FROM
    companies;
```

Output

Result Grid | Filter Rows:

name	sector
Reliance Industries	Energy
TCS	IT
Infosys	IT
HDFC Bank	Banking
ICICI Bank	Banking
HUL	FMCG
Bharti Airtel	Telecom
ITC	FMCG
L&T	Infrastructure
Axis Bank	Banking
Wipro	IT
Adani Enterprises	Infrastructure
Bajaj Finance	Finance
Maruti Suzuki	Automobile
SBI	Banking
Kotak Mahindra Bank	Banking
NTPC	Energy

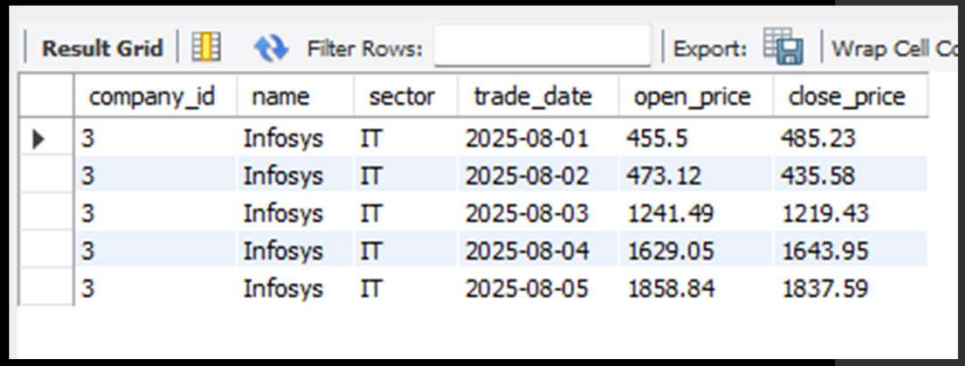
Infosys August 2025 Prices

- Stock prices of Infosys for August 2025.

Input

```sql
SELECT
    c.company_id,
    c.name,
    c.sector,
    p.trade_date,
    p.open_price,
    p.close_price
FROM companies AS c
JOIN prices AS p ON c.company_id = p.company_id
WHERE c.name = 'Infosys'
  AND p.trade_date BETWEEN '2025-08-01' AND '2025-08-31';
```

Output



Result Grid | Filter Rows: | Export: | Wrap Cell Co

company_id	name	sector	trade_date	open_price	close_price
3	Infosys	IT	2025-08-01	455.5	485.23
3	Infosys	IT	2025-08-02	473.12	435.58
3	Infosys	IT	2025-08-03	1241.49	1219.43
3	Infosys	IT	2025-08-04	1629.05	1643.95
3	Infosys	IT	2025-08-05	1858.84	1837.59

Top 10 Companies by Closing Price

- Companies with highest closing price on 2025-08-01.

Input

```sql
SELECT
  c.company_id,
  c.name,
  c.sector,
  p.trade_date,
  p.open_price,
  p.close_price
FROM companies AS c
JOIN prices AS p ON c.company_id = p.company_id
where  p.trade_date = '2025-08-01'
order by p.close_price desc
limit 10;
```

Output

company_id	name	sector	trade_date	open_price	close_price
34	JSW Steel	Infrastructure	2025-08-01	2827.88	2852.76
16	Kotak Mahindra Bank	Banking	2025-08-01	2846.74	2849.35
42	IndusInd Bank	Banking	2025-08-01	2775.54	2818.54
29	Coal India	Energy	2025-08-01	2768.25	2735.2
8	ITC	FMCG	2025-08-01	2643.91	2665.02
20	Asian Paints	FMCG	2025-08-01	2589.31	2551.94
28	Tata Steel	Infrastructure	2025-08-01	2454.12	2463.57
41	BPCL	Energy	2025-08-01	2337.89	2377.55
2	TCS	IT	2025-08-01	2314.95	2267.2
37	Cipla	Pharma	2025-08-01	2255.84	2265.35

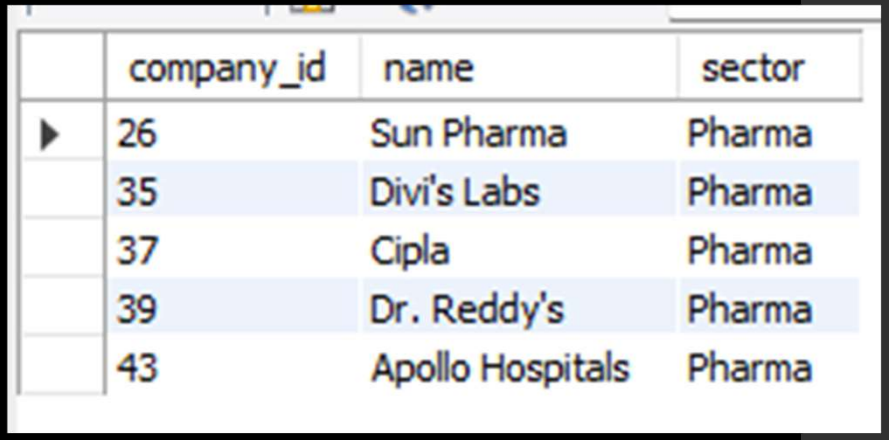
Pharmaceutical Sector

- Companies in the Pharmaceuticals sector.

Input

```
select * from companies
where sector = "Pharma";
```

Output



company_id	name	sector
26	Sun Pharma	Pharma
35	Divi's Labs	Pharma
37	Cipla	Pharma
39	Dr. Reddy's	Pharma
43	Apollo Hospitals	Pharma

Create View

- View created to combine company and price data.

Input

```
CREATE VIEW combine_data AS
    SELECT
        c.company_id,
        c.name,
        c.sector,
        p.trade_date,
        p.open_price,
        p.close_price
    FROM
        companies AS c
            JOIN
        prices AS p ON c.company_id = p.company_id;

        select * from combine_data;
```

Output

company_id	name	sector	trade_date	open_price	close_price
1	Reliance Industries	Energy	2025-08-01	1102.16	1143.77
1	Reliance Industries	Energy	2025-08-02	1387.98	1399.08
1	Reliance Industries	Energy	2025-08-03	1871.33	1827.32
1	Reliance Industries	Energy	2025-08-04	1067.46	1054.66
1	Reliance Industries	Energy	2025-08-05	920.36	965.75
2	TCS	IT	2025-08-01	2314.95	2267.2
2	TCS	IT	2025-08-02	725	763.11
2	TCS	IT	2025-08-03	2249.83	2208.73
2	TCS	IT	2025-08-04	1607.67	1631.15
2	TCS	IT	2025-08-05	2492.91	2521.31
3	Infosys	IT	2025-08-01	455.5	485.23
3	Infosys	IT	2025-08-02	473.12	435.58
3	Infosys	IT	2025-08-03	1241.49	1219.43
3	Infosys	IT	2025-08-04	1629.05	1643.95
3	Infosys	IT	2025-08-05	1858.84	1837.59
4	HDFC Bank	Banking	2025-08-01	957.41	993.42

Average Closing Price

- Average closing price of each company.

Input

```
SELECT
    name,
    sector,
    round(AVG(close_price),2) AS avg_close_price
FROM
    combine_data
GROUP BY
    name, sector;
```

Output

name	sector	avg_close_price
Reliance Industries	Energy	1278.12
TCS	IT	1878.3
Infosys	IT	1124.36
HDFC Bank	Banking	1342.93
ICICI Bank	Banking	1208.6
HUL	FMCG	1405.01
Bharti Airtel	Telecom	1040.46
ITC	FMCG	1612.35
L&T	Infrastructure	1869.42
Axis Bank	Banking	1064.97
Wipro	IT	1534.68
Adani Enterprises	Infrastructure	2015.69
Bajaj Finance	Finance	1851.32
Maruti Suzuki	Automobile	1409.08
SBI	Banking	1037.93
Kotak Mahindra Bank	Banking	1268.55
NTPC	Energy	1554.25

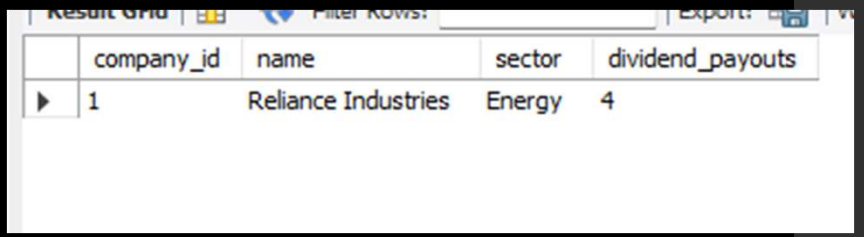
Most Dividend Payouts

- Company with highest number of dividend payouts.

Input

```sql
select company_id, name, sector, count(dividend_per_share) as dividend_payouts
from
(select
c.company_id,
c.name,
c.sector,
d.dividend_per_share
from companies as c join dividends as d
on c.company_id = d.company_id) as cd
group by company_id, name, sector
order by dividend_payouts desc
limit 1;
```

Output



company_id	name	sector	dividend_payouts
1	Reliance Industries	Energy	4

Price Difference > ₹30

- Companies with > ₹30 difference in open and close prices.

Input

```
SELECT
    *,
    abs(ROUND((close_price - open_price), 2)) AS price_change
FROM
    combine_data
WHERE
    (close_price - open_price) >= 30 and trade_date = '2025-08-03'
    order by (close_price - open_price) desc;
```

Output

company_id	name	sector	trade_date	open_price	close_price	price_change
43	Apollo Hospitals	Pharma	2025-08-03	909.12	956.47	47.35
36	Britannia	FMCG	2025-08-03	530.41	576.26	45.85
26	Sun Pharma	Pharma	2025-08-03	960.48	1005.79	45.31
24	ONGC	Energy	2025-08-03	2628.35	2668.29	39.94
34	JSW Steel	Infrastructure	2025-08-03	1905.29	1945.15	39.86
16	Kotak Mahindra Bank	Banking	2025-08-03	578.72	616.91	38.19
48	Havells	Engineering	2025-08-03	2451.35	2489.12	37.77
27	UltraTech Cement	Infrastructure	2025-08-03	1314	1350.95	36.95
29	Coal India	Energy	2025-08-03	794.47	831.31	36.84
39	Dr. Reddy's	Pharma	2025-08-03	1392.64	1429.41	36.77
49	Tata Power	Energy	2025-08-03	1117.33	1154.03	36.7
28	Tata Steel	Infrastructure	2025-08-03	2567.68	2604.27	36.59
41	BPCL	Energy	2025-08-03	1122.94	1155.89	32.95
25	Mahindra & Mahindra	Automobile	2025-08-03	797.05	829.07	32.02
20	Asian Paints	FMCG	2025-08-03	665.75	697.08	31.33

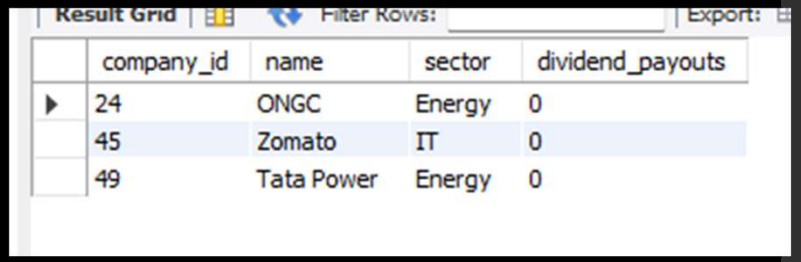
Never Issued Dividend

- Companies that never issued a dividend.

Input

```sql
SELECT *
FROM (
    SELECT
        c.company_id,
        c.name,
        c.sector,
        COUNT(d.dividend_per_share) AS dividend_payouts
    FROM
        companies AS c
    LEFT JOIN
        dividends AS d ON c.company_id = d.company_id
    GROUP BY
        c.company_id, c.name, c.sector
) AS cd
WHERE dividend_payouts = 0;
```

Output



company_id	name	sector	dividend_payouts
24	ONGC	Energy	0
45	Zomato	IT	0
49	Tata Power	Energy	0

3-day Price Decline

- Companies whose stock declined 3 days consecutively.

Input

```sql
SELECT company_id, name,close_price,prev_day_1,prev_day_2
FROM (
    SELECT
        c.company_id,
        c.name,
        p.trade_date,
        p.close_price,
        LAG(p.close_price, 1) OVER (PARTITION BY p.company_id ORDER BY p.trade_date) AS prev_day_1,
        LAG(p.close_price, 2) OVER (PARTITION BY p.company_id ORDER BY p.trade_date) AS prev_day_2,
        LAG(p.close_price, 3) OVER (PARTITION BY p.company_id ORDER BY p.trade_date) AS prev_day_3
    FROM prices p
    JOIN companies c ON c.company_id = p.company_id
) AS sub
WHERE close_price < prev_day_1 AND prev_day_1 < prev_day_2;
```

Output

company_id	name	close_price	prev_day_1	prev_day_2
1	Reliance Industries	965.75	1054.66	1827.32
4	HDFC Bank	691.72	1984.25	2764.78
5	ICICI Bank	347.37	732.03	1748.19
6	HUL	696.11	1305.57	2739.09
8	ITC	461.5	1391.11	1796.25
9	L&T	1280.11	2353.78	2514.2
12	Adani Enterprises	960.89	1498.23	1946.6
15	SBI	910.5	1053.09	1103.66
16	Kotak Mahindra Bank	616.91	1245.37	2849.35
20	Asian Paints	697.08	2483.6	2551.94
21	Nestle India	193.14	2413.34	2606.58
26	Sun Pharma	1005.79	1390.46	1427.92
28	Tata Steel	1421.2	1967.71	2604.27
29	Coal India	831.31	1704.26	2735.2
30	Grasim	752.83	1350.43	1948.49
31	Tech Mahindra	341.48	461.34	1155.9
32	Titan Company	448.43	1672.75	2080.04

Some advanced SQL queries in this project were created with the help of AI tools and used for learning purposes

Rank by Avg Close Price

- Rank companies in each sector by avg close price.

Input

```sql
SELECT
    company_id,
    name,
    sector,
    avg_close_price,
    RANK() OVER (PARTITION BY sector ORDER BY avg_close_price DESC) AS price_rank
FROM (
    SELECT
        company_id,
        name,
        sector,
        AVG(close_price) AS avg_close_price
    FROM combine_data
    GROUP BY company_id, name, sector
) AS ranked_data;
```

Output

company_id	name	sector	avg_close_price	price_rank
25	Mahindra & Mahindra	Automobile	1931.9419999999998	1
38	Eicher Motors	Automobile	1749.714	2
18	Tata Motors	Automobile	1432.6580000000001	3
14	Maruti Suzuki	Automobile	1409.078	4
4	HDFC Bank	Banking	1342.928	1
16	Kotak Mahindra Bank	Banking	1268.546	2
42	IndusInd Bank	Banking	1244.33	3
5	ICICI Bank	Banking	1208.596	4
10	Axis Bank	Banking	1064.974	5
15	SBI	Banking	1037.9279999999999	6
24	ONGC	Energy	2189.936	1
29	Coal India	Energy	1828.6620000000003	2
23	Power Grid	Energy	1674.9919999999997	3
41	BPCL	Energy	1637.198	4
17	NTPC	Energy	1554.2459999999999	5
49	Tata Power	Energy	1467.084	6
1	Reliance Industries	Energy	1278.116	7

Some advanced SQL queries in this project were created with the help of AI tools and used for learning purposes

5-day Moving Average

- 5-day moving average of closing prices.

Input

```sql
SELECT
    company_id,
    name,
    trade_date,
    close_price,
    ROUND(AVG(close_price) OVER (
        PARTITION BY company_id
        ORDER BY trade_date
        ROWS BETWEEN 4 PRECEDING AND CURRENT ROW
    ), 2) AS moving_avg_5_day
FROM combine_data;
```

Output

company_id	name	trade_date	close_price	moving_avg_5_day
1	Reliance Industries	2025-08-01	1143.77	1143.77
1	Reliance Industries	2025-08-02	1399.08	1271.42
1	Reliance Industries	2025-08-03	1827.32	1456.72
1	Reliance Industries	2025-08-04	1054.66	1356.21
1	Reliance Industries	2025-08-05	965.75	1278.12
2	TCS	2025-08-01	2267.2	2267.2
2	TCS	2025-08-02	763.11	1515.16
2	TCS	2025-08-03	2208.73	1746.35
2	TCS	2025-08-04	1631.15	1717.55
2	TCS	2025-08-05	2521.31	1878.3
3	Infosys	2025-08-01	485.23	485.23
3	Infosys	2025-08-02	435.58	460.4
3	Infosys	2025-08-03	1219.43	713.41
3	Infosys	2025-08-04	1643.95	946.05
3	Infosys	2025-08-05	1837.59	1124.36
4	HDFC Bank	2025-08-01	993.42	993.42
4	HDFC Bank	2025-08-02	280.47	636.94

Some advanced SQL queries in this project were created with the help of AI tools and used for learning purposes

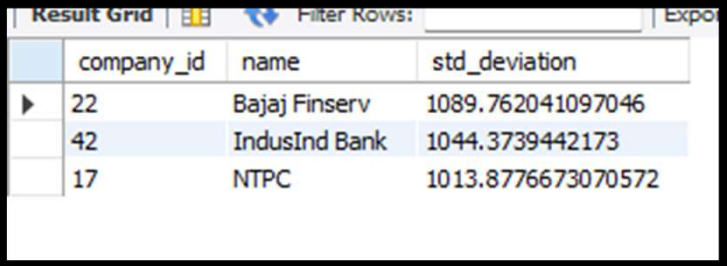
Most Volatile Stocks

- Top 3 most volatile stocks by std deviation.

Input

```sql
SELECT
    company_id,
    name,
    STDDEV(close_price) AS std_deviation
FROM
    combine_data
GROUP BY
    company_id, name
ORDER BY
    std_deviation DESC
LIMIT 3;
```

Output



company_id	name	std_deviation
22	Bajaj Finserv	1089.762041097046
42	IndusInd Bank	1044.3739442173
17	NTPC	1013.8776673070572

Some advanced SQL queries in this project were created with the help of AI tools and used for learning purposes