

```
-- 1. Average closing price per company
-- Reveals average close prices, helpful for comparing general performance.
SELECT sm.ticker, sm.company, ROUND(AVG(s.closePrice), 2) AS avg_close_price
FROM Stocks s
JOIN StockMetaData sm ON s.ticker = sm.ticker
GROUP BY sm.ticker, sm.company
ORDER BY avg_close_price DESC;
```

```
-- 2. Total volume traded per company
-- Shows investor interest/activity per stock.
SELECT sm.ticker, sm.company, SUM(s.volume) AS total_volume
FROM Stocks s
JOIN StockMetaData sm ON s.ticker = sm.ticker
GROUP BY sm.ticker, sm.company
ORDER BY total_volume DESC;
```

```
-- 3. Days with highest trading volume per stock
-- Useful to investigate what happened on the most active days.
SELECT s1.ticker, s1.date, s1.volume
FROM Stocks s1
WHERE s1.volume = (
    SELECT MAX(s2.volume) FROM Stocks s2 WHERE s2.ticker = s1.ticker
)
ORDER BY s1.ticker;
```

```
-- 4. Average sentiment score per company
-- Indicates how the news sentiment differs per company.
SELECT sm.ticker, sm.company, ROUND(AVG(se.score), 2) AS avg_sentiment
FROM Sentiment se
JOIN StockMetaData sm ON se.ticker = sm.ticker
GROUP BY sm.ticker, sm.company
ORDER BY avg_sentiment DESC;
```

```
-- 5. Count of positive/negative/neutral articles per company
-- Gauges media tone per ticker.
SELECT se.ticker, se.label, COUNT(*) AS count
FROM Sentiment se
GROUP BY se.ticker, se.label
ORDER BY se.ticker, se.label;
```

```

-- 6. Most negative article per company
-- Could explain major stock price drops.
SELECT sm.company, n.datePublished, n.headline, se.score
FROM Sentiment se
JOIN News n ON se.NewsID = n.id
JOIN StockMetaData sm ON se.ticker = sm.ticker
WHERE se.score = (
    SELECT MIN(score) FROM Sentiment se2 WHERE se2.ticker = se.ticker
)
ORDER BY se.ticker;

-- 7. Daily average stock price movement
-- Helps track daily price trends.
SELECT date, ROUND(AVG(closePrice - openPrice), 2) AS avg_price_change
FROM Stocks
GROUP BY date
ORDER BY date;

-- 8. Days with high correlation between sentiment and price drop
-- Potential link between news tone and market behavior.
SELECT s.ticker, s.date, s.openPrice, s.closePrice, se.label
FROM Stocks s
JOIN News n ON s.ticker = n.ticker AND s.date = DATE(n.datePublished)
JOIN Sentiment se ON n.id = se.NewsID
WHERE se.label = 'Negative' AND s.closePrice < s.openPrice;

-- 9. Most mentioned companies in the news
-- Helps identify companies with high media attention.
SELECT ticker, COUNT(*) AS total_articles
FROM News
GROUP BY ticker
ORDER BY total_articles DESC;

-- 10. Days with most news activity
-- May correlate with market movement or events.
SELECT DATE(datePublished) AS date, COUNT(*) AS total_articles
FROM News
GROUP BY date
ORDER BY total_articles DESC;

```

-- 11. Sector-wise average sentiment score

-- Useful to compare investor optimism by industry.

```
SELECT sm.sector, ROUND(AVG(se.score), 2) AS avg_sentiment
FROM Sentiment se
JOIN StockMetaData sm ON se.ticker = sm.ticker
GROUP BY sm.sector;
```

-- 12. Subquery to find top 3 most volatile stocks (based on price range)

```
SELECT ticker, MAX(highPrice) - MIN(lowPrice) AS volatility
FROM Stocks
GROUP BY ticker
ORDER BY volatility DESC
LIMIT 3;
```

-- 13. Price change summary per company

-- How much each stock moved overall.

```
SELECT ticker, ROUND(MAX(closePrice) - MIN(openPrice), 2) AS net_change
FROM Stocks
GROUP BY ticker;
```

-- 14. View: Daily sentiment summary per stock

-- Simplifies reuse when analyzing daily mood per ticker.

```
CREATE OR REPLACE VIEW DailySentimentSummary AS
SELECT s.ticker, DATE(n.datePublished) AS date, ROUND(AVG(se.score), 2) AS avg_score
FROM Sentiment se
JOIN News n ON se.NewsID = n.id
JOIN Stocks s ON s.ticker = se.ticker AND DATE(n.datePublished) = s.date
GROUP BY s.ticker, DATE(n.datePublished);
```

```
SELECT * FROM DailySentimentSummary
WHERE ticker = 'TSLA'
ORDER BY date DESC;
```

-- 15. Using the View: Get dates where sentiment was positive but price dropped

-- Useful for finding sentiment/price mismatches.

```
SELECT dss.ticker, dss.date, dss.avg_score, s.openPrice, s.closePrice
FROM DailySentimentSummary dss
JOIN Stocks s ON dss.ticker = s.ticker AND dss.date = s.date
```

```
WHERE dss.avg_score > 0 AND s.closePrice < s.openPrice;
```