Assume that we have list of two tables

1. Companies
2. StockPrices

1. `Companies` table:

sql

CREATE TABLE Companies (

CompanyId INT PRIMARY KEY,

CompanyName VARCHAR(50),

Sector VARCHAR(50)

);

INSERT INTO Companies VALUES

(1, 'ABC Corp', 'Technology'),

(2, 'XYZ Inc', 'Finance'),

(3, 'Sky Corp', 'Technology'),

(4, 'Ray gold', 'Finance'),

(5, 'KK business Corp', 'Technology'),

(6, 'Yummy Inc', 'Finance'),

2. `StockPrices` table:

sql

CREATE TABLE StockPrices (

StockId INT PRIMARY KEY,

CompanyId INT,

StockDate DATE,

StockPrice DECIMAL(10, 2),

TradingVolume INT

);

INSERT INTO StockPrices VALUES

(1, 6, '2023-07-01', 100.50, 100000),

(2, 8, '2023-08-21', 50.75, 75000),

(1, 9, '2023-01-01', 100.50, 10000),

(7, 8, '2023-06-01', 50.75, 5000),

(1, 4, '2023-09-01', 100.50, 10000),

(2, 7, '2023-01-11', 50.75, 76000),

### Data Retrieval Queries:

1. Retrieve all stock prices for a specific company:

sql

SELECT \* FROM StockPrices WHERE CompanyId = 1;

2. Retrieve the highest and lowest stock prices for a given date range:

sql

SELECT MAX(StockPrice) AS HighestPrice, MIN(StockPrice) AS LowestPrice

FROM StockPrices

WHERE StockDate BETWEEN '2023-01-01' AND '2023-01-31';

3. List all companies in a specific sector:

sql

SELECT \* FROM Companies WHERE Sector = 'Technology';

4. Display the closing prices of a specific stock for the last 30 days:

sql

SELECT StockDate, StockPrice

FROM StockPrices

WHERE CompanyId = 1

ORDER BY StockDate DESC

LIMIT 30;

5. Retrieve the top 10 performing stocks based on percentage change:

sql

SELECT CompanyId, MAX(StockPrice) AS MaxPrice, MIN(StockPrice) AS MinPrice,

((MAX(StockPrice) - MIN(StockPrice)) / MIN(StockPrice)) \* 100 AS PercentageChange

FROM StockPrices

GROUP BY CompanyId

ORDER BY PercentageChange DESC

LIMIT 10;

6. Get the total trading volume for a specific stock:

sql

SELECT CompanyId, SUM(TradingVolume) AS TotalVolume

FROM StockPrices

WHERE CompanyId = 1

GROUP BY CompanyId;

7. List the stocks that have experienced a significant price increase in the last week:

sql

SELECT CompanyId, MAX(StockPrice) AS CurrentPrice, MIN(StockPrice) AS LastWeekPrice,

((MAX(StockPrice) - MIN(StockPrice)) / MIN(StockPrice)) \* 100 AS PercentageChange

FROM StockPrices

WHERE StockDate BETWEEN '2023-11-01' AND '2023-11-07'

GROUP BY CompanyId

HAVING PercentageChange > 5;

8. Display the average stock price for each month in the last year:

sql

SELECT MONTH(StockDate) AS Month, AVG(StockPrice) AS AveragePrice

FROM StockPrices

WHERE StockDate BETWEEN '2022-11-01' AND '2023-11-01'

GROUP BY MONTH(StockDate)

ORDER BY Month;

9. Retrieve the stocks with the highest trading volume for a specific date:

sql

SELECT CompanyId, StockPrice, TradingVolume

FROM StockPrices

WHERE StockDate = '2023-01-01'

ORDER BY TradingVolume DESC

LIMIT 5;

10. List the top 5 stocks by market capitalization:

sql

SELECT Companies.CompanyId, CompanyName, MAX(StockPrice) AS MaxPrice, TradingVolume

FROM StockPrices

JOIN Companies ON StockPrices.CompanyId = Companies.CompanyId

GROUP BY Companies.CompanyId

ORDER BY MaxPrice \* TradingVolume DESC

LIMIT 5;

Certainly! Here are more SQL queries for a stock market dataset:

### More Data Retrieval Queries:

11. Retrieve the closing prices and trading volumes for all stocks on a specific date:

sql

SELECT CompanyId, StockPrice, TradingVolume

FROM StockPrices

WHERE StockDate = '2023-01-15';

12. Find the stocks with the highest closing prices in each sector:

sql

SELECT c.Sector, s.CompanyId, s.StockPrice

FROM StockPrices s

JOIN Companies c ON s.CompanyId = c.CompanyId

WHERE s.StockPrice = (SELECT MAX(StockPrice) FROM StockPrices WHERE CompanyId = s.CompanyId)

ORDER BY c.Sector;

13. Retrieve the average trading volume for each sector:

sql

SELECT c.Sector, AVG(s.TradingVolume) AS AvgTradingVolume

FROM StockPrices s

JOIN Companies c ON s.CompanyId = c.CompanyId

GROUP BY c.Sector;

14. List stocks that had a positive percentage change in the last month:

sql

SELECT CompanyId, MAX(StockPrice) AS CurrentPrice, MIN(StockPrice) AS LastMonthPrice,

((MAX(StockPrice) - MIN(StockPrice)) / MIN(StockPrice)) \* 100 AS PercentageChange

FROM StockPrices

WHERE StockDate BETWEEN '2023-10-01' AND '2023-10-31'

GROUP BY CompanyId

HAVING PercentageChange > 0;

15. Retrieve the top 5 stocks with the lowest closing prices:

sql

SELECT CompanyId, MIN(StockPrice) AS MinPrice

FROM StockPrices

GROUP BY CompanyId

ORDER BY MinPrice

LIMIT 5;

### Advanced Data Analysis Queries:

16. Identify stocks that are trading at a new all-time high:

sql

SELECT CompanyId, MAX(StockPrice) AS CurrentPrice

FROM StockPrices

GROUP BY CompanyId

HAVING CurrentPrice = (SELECT MAX(StockPrice) FROM StockPrices WHERE CompanyId = s.CompanyId);

17. Calculate the Moving Average for a specific stock over the last 10 days:

sql

SELECT StockDate, StockPrice,

AVG(StockPrice) OVER (ORDER BY StockDate ROWS BETWEEN 9 PRECEDING AND CURRENT ROW) AS MovingAverage

FROM StockPrices

WHERE CompanyId = 1

ORDER BY StockDate;

18. Identify stocks that have consistently increased in value over the last quarter:

sql

SELECT CompanyId

FROM StockPrices

WHERE StockDate BETWEEN '2023-08-01' AND '2023-10-31'

GROUP BY CompanyId

HAVING COUNT(DISTINCT StockPrice) = DATEDIFF('2023-10-31', '2023-08-01') + 1;

19. Calculate the rate of return for each stock over the last 30 days:

sql

SELECT CompanyId, StockDate, StockPrice,

((StockPrice - LAG(StockPrice) OVER (PARTITION BY CompanyId ORDER BY StockDate)) / LAG(StockPrice) OVER (PARTITION BY CompanyId ORDER BY StockDate)) \* 100 AS RateOfReturn

FROM StockPrices

WHERE StockDate BETWEEN '2023-10-01' AND '2023-10-31'

ORDER BY CompanyId, StockDate;

20. Determine the average daily trading volume for stocks in the Technology sector:

sql

SELECT s.CompanyId, AVG(s.TradingVolume) AS AvgDailyTradingVolume

FROM StockPrices s

JOIN Companies c ON s.CompanyId = c.CompanyId

WHERE c.Sector = 'Technology'

GROUP BY s.CompanyId;

### Data Modification and Maintenance Queries:

21. Update the stock price for a specific company on a given date:

sql

UPDATE StockPrices

SET StockPrice = 105.75

WHERE CompanyId = 1 AND StockDate = '2023-01-05';

22. Insert a new stock price record for a recently listed company:

sql

INSERT INTO StockPrices (CompanyId, StockDate, StockPrice, TradingVolume)

VALUES (3, '2023-01-15', 75.25, 50000);

23. Delete outdated stock price records:

sql

DELETE FROM StockPrices

WHERE StockDate < '2023-01-01';

24. Adjust stock prices for stock splits or dividends:

sql

-- Example: If there's a 2-for-1 stock split for CompanyId = 1 on '2023-01-15'

UPDATE StockPrices

SET StockPrice = StockPrice / 2

WHERE CompanyId = 1 AND StockDate >= '2023-01-15';

25. Archive old stock price records to improve database performance:

sql

-- Move records older than one year to an archive table

INSERT INTO StockPricesArchive

SELECT \* FROM StockPrices WHERE StockDate < DATEADD(YEAR, -1, GETDATE());

-- Delete the archived records from the main table

DELETE FROM StockPrices WHERE StockDate < DATEADD(YEAR, -1, GETDATE());