

i. Find the tickers and all closing prices of all stocks exchanged in 2024.

This query asks me to retrieve all stock tickers along with their closing prices for every day on which they were traded in the year 2024.

The closing price information is stored in the price (ticker, date, close) relation.

Therefore, I first apply a selection (σ) to keep only the tuples where the date falls within the year 2024, and then I project (π) the required attributes: ticker, date and close.

$$\pi_{\text{ticker, date, close}} (\sigma_{2024-01-01 \leq \text{date} \wedge \text{date} \leq 2024-12-31} (\text{price}))$$
$$\pi_{\text{ticker, date, close}} (\sigma_{\text{year}(\text{date})=2024} (\text{price}))$$

Ticker	Date	close
IBM	2024-02-21	90
GOOG	2024-02-21	100
XOM	2024-02-23	125
⋮	⋮	⋮

I filtered the Price relation to include only tuples from the year 2024 and projected the attributes (ticker, date, close) to obtain all closing price records for all traded stocks in that year.

- SQL Query :

```
SELECT DISTINCT ticker, date, close
FROM price
WHERE exchange = 'NYSE'
      AND date = '2024-02-25'
      AND (close < 20 OR close > 100);
```

ii. Find all tickers (i.e. for all dates) whose closing price is both higher than 'IBM' on '2/25/2024' and no higher than 'GOOG' on '2/25/2024'.

This query asked me to compare the closing prices of all stocks on 02/25/2024 with the closing prices of IBM and Goog on the same date.

I first extract IBM's and Goog's closing prices on 02/25/2024 and rename the attributes to avoid name conflicts during joins.

Then, I compare all closing prices with these reference values to retain only those stocks whose closing price is greater than IBM's closing price and less than or equal to Goog's closing price on that date.

Finally, I project the tickers.

$$I \leftarrow \rho_{(i_ticker, i_date, i_close)} (\sigma_{(ticker='IBM' \wedge date='2024-02-25'} (Price))$$
$$G \leftarrow \rho_{(g_ticker, g_date, g_close)} (\sigma_{(ticker='Goog' \wedge date='2024-02-25'} (Price))$$
$$\pi_{ticker} (\sigma_{close > i_close (Price \times I) \cap close \leq g_close (Price \times G))$$

- SQL Query :

```
SELECT ticker, date, close
FROM PRICE
WHERE date = '2024-02-25'
AND close > (
    SELECT close
    FROM PRICE
    WHERE ticker = 'IBM'
    AND date = '2024-02-25'
)
AND close <= (
    SELECT close
    FROM PRICE
    WHERE ticker = 'GOOG'
    AND date = '2024-02-25'
);
```

iii. Find the tickers of all stocks that closed at the highest price on '2/25/2024'.
(we are asking for "all stocks" since there may be more than one with the same "highest price")

This query asks me to find which stocks had the highest closing price on 02/25/2024. Since multiple stocks may share the same highest price, I need to ensure that I return all such tickers.

To do this I first select all price records from the date 02/25/2024.

Next, I create two remove copies of this result to allow comparison of closing prices.

Then, I identify all tickers that have a lower closing price than some other ticker on that date.

Finally, I subtract those from the full set, so the remaining tickers represent those with the maximum closing price.

$$P \leftarrow \sigma_{\text{date} = '2024-02-25'}(P_{\text{price}})$$
$$X \leftarrow \rho_{x_ticker, x_date, x_close}(P)$$
$$Y \leftarrow \rho_{y_ticker, y_date, y_close}(P)$$
$$\text{Lower Tickers} \leftarrow \pi_{x_ticker}(\sigma_{x_close < y_close}(X \times Y))$$
$$\pi_{\text{ticker}}(P) - \text{Lower Tickers}$$

- SQL Query :

```
SELECT ticker
FROM Price
WHERE date = '2024-02-25'
AND close = (
    SELECT MAX(close)
    FROM Price
    WHERE date = '2024-02-25'
);
```

iv. Find the tickers of all stocks in 'NYSE' whose closing price on '2/25/2024' was either strictly below \$20 or strictly above \$100.

This query asks me to retrieve the tickers of all stocks traded in the NYSE whose closing price on 02/25/2024 was either less than \$20 or greater than \$100.

Since the exchange information is stored in the Stock relation, I first form a temporary relation by taking the Cartesian product of price and stock, and then apply a selection to match price records with the correct stock using the ticker attribute.

Then, I filter for records where the exchange is 'NYSE' and the closing price condition is satisfied. Finally, I project the ticker attribute.

$$P_NY \leftarrow \sigma_{\text{price.ticker} = \text{stock.ticker}}(\text{Price} \times \text{Stock})$$
$$\text{Result} = \sigma_{\text{exchange} = 'NYSE' \wedge \text{date} = '2024-02-25' \wedge (\text{close} < 20 \vee \text{close} > 100)}(P_NY)$$
$$\pi_{\text{ticker}}(\text{Result})$$

- SQL Query :

```
SELECT DISTINCT p.ticker
FROM Price p
JOIN Stock s ON p.ticker = s.ticker
WHERE s.exchange = 'NYSE'
AND p.date = '2024-02-25'
AND (p.close < 20 OR p.close > 100);
```

v. Find all tickers in 'NYSE' of the stocks whose closing price showed the highest increase between '2/25/2024' and '2/26/2024' in 'NYSE' and whose closing price was (in 'NYSE') strictly above \$100 for the entire 2024.

(we are asking for "all stocks" since there may be more than one with the same increase. Recall that Relational Algebra does NOT support MAX, MIN, AVG operations.)

This query asks me to find NYSE stocks with the largest 1 day increase from 02/25/2024 to 02/26/2024, while also requiring that their closing price was always >\$100 for the whole year 2024.

I first match price records with the corresponding stock records by taking the Cartesian product of price and then selecting the tuples where the ticker match. Then, I filter to keep only the records where the exchange is 'NYSE'. Then I pair each ticker's 02/25 price with its 02/26 price. I compute the increase as a new attribute using a rename rule, and apply the anti-dominance pattern to keep only the tickers whose increase is not beaten by any other ticker (ties remain). Finally, I intersect this set with the tickers that were strictly above \$100 throughout 2024.

$$P \leftarrow \sigma_{\text{price.ticker} = \text{stock.ticker}} (\text{Price} \times \text{Stock})$$

$$\text{NY_P} \leftarrow \sigma_{\text{exchange} = \text{'NYSE'}} (P)$$

$$P_{25} \leftarrow \rho_{t, c_{25}} (\pi_{\text{ticker, close}} \sigma_{\text{date} = \text{'2024-02-25'}} (P))$$

$$P_{26} \leftarrow \rho_{t, c_{26}} (\pi_{\text{ticker, close}} \sigma_{\text{date} = \text{'2024-02-26'}} (P))$$

$$J \leftarrow \pi_{t, c_{25}, c_{26}} \sigma_{t=t'} (P_{25} \times P_{26})$$

$$\Delta \leftarrow \rho_{c_{26}-c_{25} \rightarrow \text{inc}} (J)$$

$$\Delta_1 \leftarrow \rho_{t_1, \text{inc}_1} (\Delta)$$

$$\Delta_2 \leftarrow \rho_{t_2, \text{inc}_2} (\Delta)$$

$$\text{Dom} \leftarrow \pi_{t_1} \sigma_{\text{inc}_1 < \text{inc}_2} (\Delta_1 \times \Delta_2)$$

$$\text{Max rise} \leftarrow \pi_{t_1} (\Delta) - \text{Dom}$$

$$\text{Bad} \leftarrow \pi_{\text{ticker}} \sigma_{\text{year}(\text{date}) = 2024 \wedge \text{close} \leq 100} (P)$$

$$\text{Always over 100} \leftarrow \pi_{\text{ticker}} \sigma_{\text{year}(\text{date}) = 2024} (P) - \text{Bad}$$

Max rise \cap Always over 100

- **SQL Query :**

```
WITH p AS (  
    SELECT p.ticker, p.date, p.close, s.exchange  
    FROM Price p  
    JOIN Stock s ON p.ticker = s.ticker  
    WHERE s.exchange = 'NYSE'  
),  
  
inc AS (  
    SELECT p26.ticker,  
           p26.close - p25.close AS inc  
    FROM p p25  
    JOIN p p26  
        ON p25.ticker = p26.ticker  
        AND p25.date   = '2024-02-25'  
        AND p26.date   = '2024-02-26'  
),  
  
max_inc AS (  
    SELECT MAX(inc) AS max_inc  
    FROM inc  
),  
  
candidates AS (  
    SELECT i.ticker  
    FROM inc i, max_inc m  
    WHERE i.inc = m.max_inc  
),  
  
always_over_100 AS (  
    SELECT ticker  
    FROM p  
    WHERE date >= '2024-01-01'  
        AND date <= '2024-12-31'  
    GROUP BY ticker  
    HAVING MIN(close) > 100  
)  
  
SELECT c.ticker  
FROM candidates c  
JOIN always_over_100 a ON c.ticker = a.ticker;
```