```
--Table

SELECT
  *
FROM nifty.nifty_50
LIMIT 10


--7 Days MA & 14 Days MA

SELECT
    date,
    close,
    AVG(close) OVER (ORDER BY date ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS
moving_avg_7d,
    AVG(close) OVER (ORDER BY date ROWS BETWEEN 13 PRECEDING AND CURRENT ROW) AS
moving_avg_14d
FROM nifty.nifty_50
ORDER BY date;


--RSI

WITH price_changes AS
  (SELECT
    date,
    close,
    close - LAG(close) OVER (ORDER BY date) AS price_change
  FROM nifty.nifty_50),

  gains_and_losses AS
  (SELECT
    date,
    close,
    price_change,
    CASE
      WHEN price_change > 0 THEN price_change
      ELSE 0
      END AS gain,
    CASE
      WHEN price_change < 0 THEN ABS(price_change)
      ELSE 0
      END AS loss
  FROM price_changes),

  avg_gain_loss AS
  (SELECT
    date,
    close,
    AVG(gain) OVER (ORDER BY date ROWS BETWEEN 13 PRECEDING AND CURRENT ROW) AS
avg_gain,
    AVG(loss) OVER (ORDER BY date ROWS BETWEEN 13 PRECEDING AND CURRENT ROW) AS
avg_loss
  FROM gains_and_losses),

  rsi_calculation AS
  (SELECT
    date,
    close,
    avg_gain,
    avg_loss,
    CASE
      WHEN avg_loss = 0 THEN 100
      ELSE 100 - (100 / (1 + (avg_gain / avg_loss)))
      END AS rsi
  FROM avg_gain_loss)

SELECT
    date,
    close,
    rsi
FROM rsi_calculation
ORDER BY date;



--MACD

WITH ema_calculations AS
  (SELECT
    date,
    close,
    AVG(close) OVER (ORDER BY date ROWS BETWEEN 25 PRECEDING AND CURRENT ROW) AS slow_ema,
    AVG(close) OVER (ORDER BY date ROWS BETWEEN 11 PRECEDING AND CURRENT ROW) AS fast_ema
  FROM nifty.nifty_50),

  macd_calculations AS
  (SELECT
    date,
    close,
    fast_ema,
    slow_ema,
    fast_ema - slow_ema AS macd
  FROM ema_calculations),

  signal_line_calculation AS
  (SELECT
    date,
    close,
    macd,
    AVG(macd) OVER (ORDER BY date ROWS BETWEEN 8 PRECEDING AND CURRENT ROW) AS signal_line
  FROM macd_calculations),

  macd_final AS
  (SELECT
    date,
    close,
    macd,
    signal_line,
    macd - signal_line AS macd_histogram
  FROM signal_line_calculation)

SELECT
    date,
    close,
    macd,
    signal_line,
    macd_histogram
FROM macd_final
ORDER BY date;


--Bollinger Band

WITH bollinger_data AS
  (SELECT
    date,
    close,
    AVG(close) OVER (ORDER BY date ROWS BETWEEN 19 PRECEDING AND CURRENT ROW) AS
sma_20,
    STDDEV(close) OVER (ORDER BY date ROWS BETWEEN 19 PRECEDING AND CURRENT ROW) AS
stddev_20
  FROM nifty.nifty_50)
SELECT
    date,
    close,
    sma_20 AS middle_band,
    sma_20 + (2 * stddev_20) AS upper_band,
    sma_20 - (2 * stddev_20) AS lower_band
FROM bollinger_data
ORDER BY date;

--Stochastic

WITH stoch_data AS
  (SELECT
    date,
    close,
    MAX(high) OVER (ORDER BY date ROWS BETWEEN 13 PRECEDING AND CURRENT ROW) AS
high_14,
    MIN(low) OVER (ORDER BY date ROWS BETWEEN 13 PRECEDING AND CURRENT ROW) AS
low_14
  FROM nifty.nifty_50)
SELECT
  date,
  close,
  ((close - low_14) / (high_14 - low_14)) * 100 AS percent_k,
  AVG(((close - low_14) / (high_14 - low_14)) * 100) OVER (ORDER BY date ROWS
BETWEEN 2 PRECEDING AND CURRENT ROW) AS percent_d
FROM stoch_data
ORDER BY date;




--Pivot Point

SELECT
  date,
  close,
  (high + low + close) / 3 AS pivot_point,
  (2 * ((high + low + close) / 3)) - low AS resistance_1,
  (high - low) + ((high + low + close) / 3) AS resistance_2,
  (2 * ((high + low + close) / 3)) - high AS support_1,
  ((high + low + close) / 3) - (high - low) AS support_2
FROM nifty.nifty_50
ORDER BY date;


--Rate of Change

SELECT
  date,
  close,
  LAG(close, 14) OVER (ORDER BY date) AS close_14_days_ago,
  ((close - LAG(close, 14) OVER (ORDER BY date)) / LAG(close, 14) OVER (ORDER BY
date)) * 100 AS roc_14
FROM nifty.nifty_50
ORDER BY date;

--Bollinger Bands Breakout Strategy

WITH moving_average AS
  (SELECT
    Date,
    Close,
    AVG(Close) OVER (ORDER BY Date ROWS BETWEEN 19 PRECEDING AND CURRENT ROW) AS
```

```sql
SMA_20
  FROM`nifty.nifty_50`),

  std_deviation AS
  (SELECT
    Date,
    Close,
    SMA_20,
    STDDEV(Close) OVER (ORDER BY Date ROWS BETWEEN 19 PRECEDING AND CURRENT ROW) AS
StdDev_20
  FROM moving_average),

  bollinger_bands AS
  (SELECT
    Date,
    Close,
    SMA_20,
    StdDev_20,
    (SMA_20 + 2 * StdDev_20) AS Upper_Band,
    (SMA_20 - 2 * StdDev_20) AS Lower_Band
  FROM std_deviation),

  breakout_signals AS
  (SELECT
    Date,
    Close,
    Upper_Band,
    Lower_Band,
    CASE
      WHEN Close > Upper_Band THEN 'Buy'
      WHEN Close < Lower_Band THEN 'Sell'
      ELSE 'Hold'
    END AS Breakout_Signal
  FROM bollinger_bands)

SELECT
  Date,
  Close,
  Upper_Band,
  Lower_Band,
  Breakout_Signal
FROM breakout_signals
ORDER BY Date;


--SQL Query for Pivot Point Strategy

WITH pivot_points AS (
  SELECT
    Date,
    High,
    Low,
    Close,
    (High + Low + Close) / 3 AS Pivot_Point
  FROM`nifty.nifty_50`)

SELECT
  Date,
  High,
  Low,
  Close,
  Pivot_Point,
  CASE
    WHEN Close > Pivot_Point THEN 'Buy'
    WHEN Close < Pivot_Point THEN 'Sell'
    ELSE 'Hold'
  END AS Trade_Signal
FROM pivot_points
ORDER BY Date;
```