

```

1 SELECT * FROM stocks_tbl;
2
3 -- A.Basic Analysis
4 -- Queries: Perform basic analysis on the data and identify trends.
5 -- What are the distinct stocks in the table?
6 SELECT DISTINCT stock_symbol, stock_name
7 FROM stocks_tbl;
8
9 SELECT stock_symbol, stock_name, count(*)
10 FROM stocks_tbl
11 GROUP BY stock_symbol, stock_name;
12
13 -- Query all data for a single stock. Do you notice any overall trends?
14 SELECT *
15 FROM stocks_tbl
16 WHERE stock_symbol = "AAPL";
17
18 -- Which rows have a price above 100? between 40 to 50, etc?
19 SELECT *
20 FROM stocks_tbl
21 WHERE stock_price > 100;
22
23 SELECT stock_symbol, stock_name, count(*)
24 FROM stocks_tbl
25 WHERE stock_price > 100
26 GROUP BY stock_symbol, stock_name;
27
28 SELECT *
29 FROM stocks_tbl
30 WHERE stock_price BETWEEN 30 AND 50;
31
32 SELECT stock_symbol, stock_name, count(*)
33 FROM stocks_tbl
34 WHERE stock_price BETWEEN 30 AND 50
35 GROUP BY stock_symbol, stock_name;
36
37 -- Sort the table by price. What are the minimum and maximum prices?
38 SELECT *
39 FROM stocks_tbl
40 ORDER BY stock_price;
41
42 -- B.Intermediate Challenge
43 -- 1.Explore using aggregate functions to look at key statistics about the data (e.g., min, max,
44 -- average).
45 SELECT min(stock_price) AS [Lowest Price], round(avg(stock_price),2) AS [Average Price],
46 max(stock_price) AS [Highest Price]
47 FROM stocks_tbl;
48
49 SELECT stock_symbol, stock_name, min(stock_price) AS [Lowest Price], dtm_stamp
50 FROM stocks_tbl;
51
52 SELECT stock_symbol, stock_name, max(stock_price) AS [Highest Price], dtm_stamp
53 FROM stocks_tbl;
54
55 SELECT round(avg(stock_price),2) AS [Average Price]
56 FROM stocks_tbl;
57
58 -- 2.Group the data by stock and repeat. How do the stocks compare to each other?
59 SELECT stock_symbol, stock_name, min(stock_price) AS [Price Floor], round(avg(stock_price),2) AS
60 [Average Price], max(stock_price) AS [Price Ceiling]
61 FROM stocks_tbl
62 GROUP BY stock_symbol, stock_name;
63
64 SELECT stock_symbol, stock_name, min(stock_price) AS [Lowest Price], max(stock_price) AS [Highest
65 Price],
66 round(avg(stock_price),2) AS [Average Price], round((max(stock_price) - min(stock_price)),2) AS
67 [Range]
68 FROM stocks_tbl
69 GROUP BY stock_symbol, stock_name;
70
71 SELECT *
72 FROM stocks_tbl
73 WHERE stock_symbol = "AAPL";

```

```

69
70 SELECT *
71 FROM stocks_tbl
72 WHERE stock_symbol = "TSLA";
73
74 -- 3.Group the data by day or hour of day. Does day of week or time of day impact prices?
75 SELECT DISTINCT stock_symbol, effective_date,
76     (SELECT stock_price FROM stocks_tbl AS [TMP_1]
77      WHERE stocks_tbl.stock_symbol = TMP_1.stock_symbol
78      AND stocks_tbl.effective_date = TMP_1.effective_date
79      AND time(dttm_stamp) = "09:30:00") AS [Opening Balance],
80     (SELECT stock_price FROM stocks_tbl AS [TMP_1]
81      WHERE stocks_tbl.stock_symbol = TMP_1.stock_symbol
82      AND stocks_tbl.effective_date = TMP_1.effective_date
83      AND time(dttm_stamp) = "16:00:00") AS [Closing Balance]
84 --     , closing_balance - opening_balance --> This will be the Change
85 --     , 100 * (closing_balance - opening_balance) / opening_balance --> This will be the Change %
86 FROM stocks_tbl;
87
88 WITH change_comp_query AS (
89     SELECT DISTINCT stock_symbol, effective_date,
90         (SELECT stock_price FROM stocks_tbl AS [TMP_1]
91          WHERE stocks_tbl.stock_symbol = TMP_1.stock_symbol
92          AND stocks_tbl.effective_date = TMP_1.effective_date
93          AND time(dttm_stamp) = "09:30:00") AS opening_balance,
94         (SELECT stock_price FROM stocks_tbl AS [TMP_1]
95          WHERE stocks_tbl.stock_symbol = TMP_1.stock_symbol
96          AND stocks_tbl.effective_date = TMP_1.effective_date
97          AND time(dttm_stamp) = "16:00:00") AS closing_balance
98     FROM stocks_tbl
99 )
100 SELECT *,
101     round((closing_balance - opening_balance), 2) AS [Change],
102     round((100 * (closing_balance - opening_balance) / opening_balance), 2) AS [Change %]
103 FROM change_comp_query;
104
105 -- 4.Which of the rows have a price greater than the average of all prices in the dataset?
106 SELECT *
107 FROM stocks_tbl
108 WHERE stock_price > (SELECT avg(stock_price)
109                     FROM stocks_tbl)
110 ;
111
112
113 SELECT stock_symbol, stock_name, count(*)
114 FROM stocks_tbl
115 WHERE stock_price > (SELECT avg(stock_price)
116                     FROM stocks_tbl)
117 GROUP BY stock_symbol, stock_name;
118

```