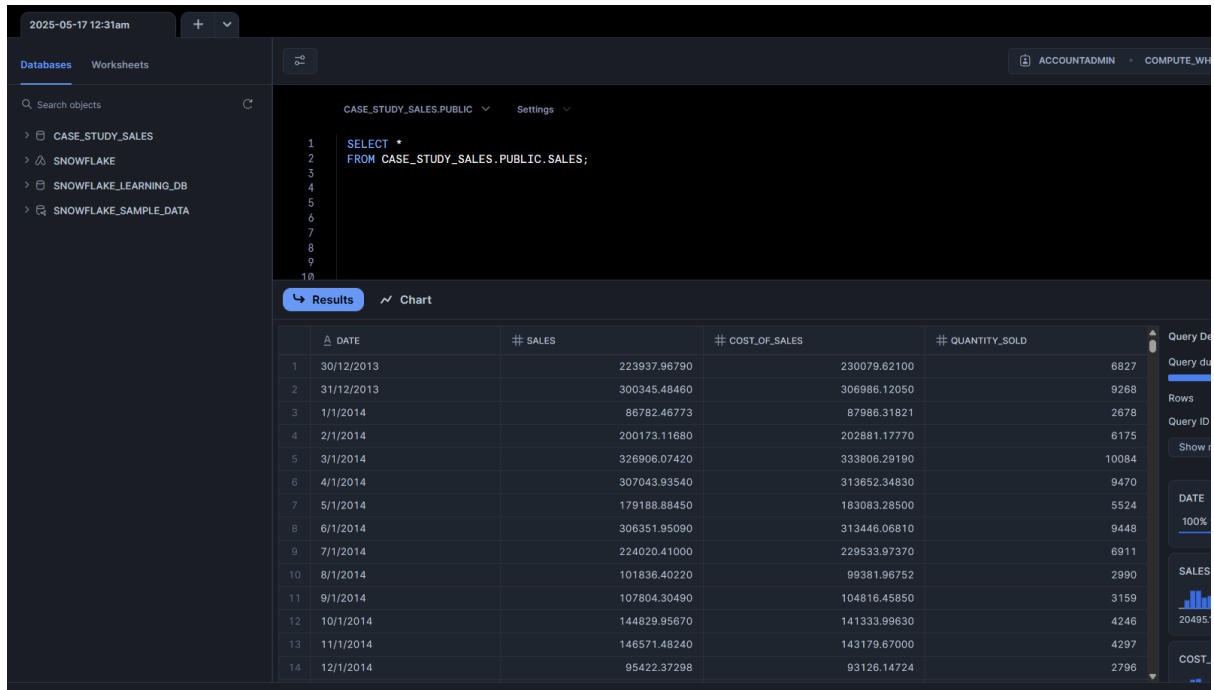


METHODOLOGY

1. Verifying the data.



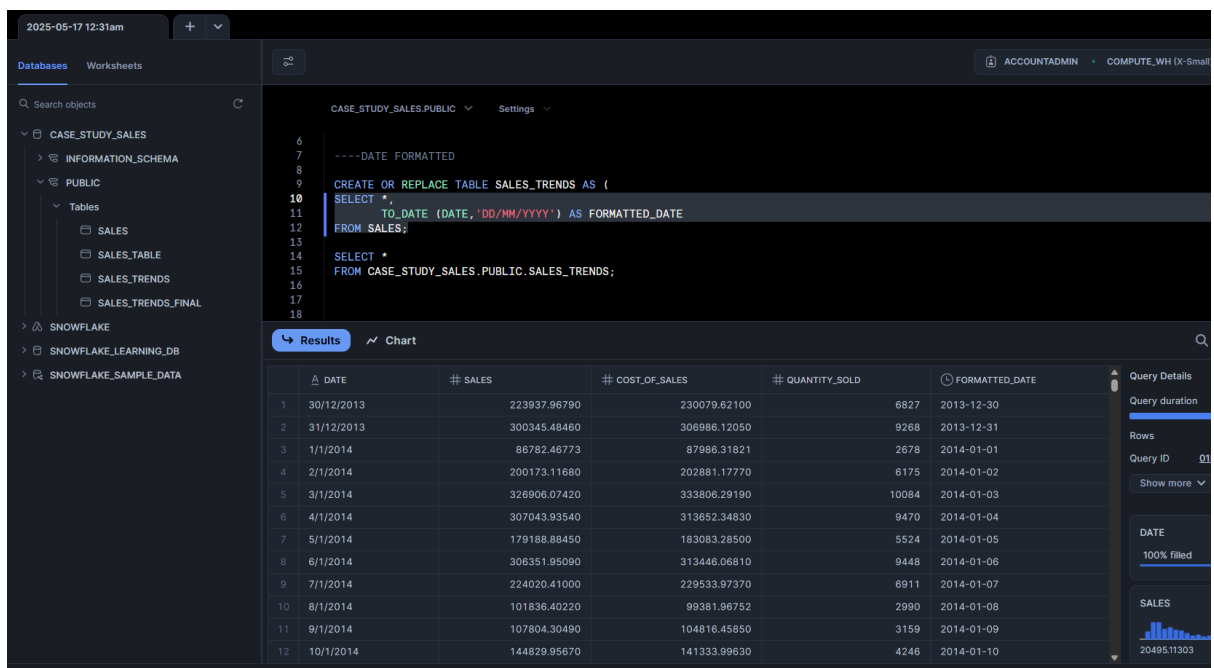
The screenshot shows a database interface with a query editor and a results table. The query is:

```
SELECT *  
FROM CASE_STUDY_SALES.PUBLIC.SALES;
```

The results table has the following columns: DATE, # SALES, # COST_OF_SALES, # QUANTITY_SOLD, and Query Duration. The data is as follows:

| | DATE | # SALES | # COST_OF_SALES | # QUANTITY_SOLD | Query Duration |
|----|------------|--------------|-----------------|-----------------|----------------|
| 1 | 30/12/2013 | 223937.96790 | 230079.62100 | 6827 | Query duration |
| 2 | 31/12/2013 | 300345.48460 | 306986.12050 | 9268 | Rows |
| 3 | 1/1/2014 | 86782.46773 | 87986.31821 | 2678 | Query ID |
| 4 | 2/1/2014 | 200173.11680 | 202881.17770 | 6175 | Show more |
| 5 | 3/1/2014 | 326906.07420 | 333806.29190 | 10084 | DATE |
| 6 | 4/1/2014 | 307043.93540 | 313652.34830 | 9470 | 100% |
| 7 | 5/1/2014 | 179188.88450 | 183083.28500 | 5524 | SALES |
| 8 | 6/1/2014 | 306351.95090 | 313446.06810 | 9448 | 20495 |
| 9 | 7/1/2014 | 224020.41000 | 229533.97370 | 6911 | COST_ |
| 10 | 8/1/2014 | 101836.40220 | 99381.96752 | 2990 | |
| 11 | 9/1/2014 | 107804.30490 | 104816.45850 | 3159 | |
| 12 | 10/1/2014 | 144829.95670 | 141333.99630 | 4246 | |
| 13 | 11/1/2014 | 146571.48240 | 143179.67000 | 4297 | |
| 14 | 12/1/2014 | 95422.37298 | 93126.14724 | 2796 | |

2. Formatting date and creating a new table .



The screenshot shows a database interface with a query editor and a results table. The query is:

```
CREATE OR REPLACE TABLE SALES_TRENDS AS (  
SELECT *,  
       TO_DATE (DATE,'DD/MM/YYYY') AS FORMATTED_DATE  
FROM SALES;  
)  
SELECT *  
FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS;
```

The results table has the following columns: DATE, # SALES, # COST_OF_SALES, # QUANTITY_SOLD, and FORMATTED_DATE. The data is as follows:

| | DATE | # SALES | # COST_OF_SALES | # QUANTITY_SOLD | FORMATTED_DATE |
|----|------------|--------------|-----------------|-----------------|----------------|
| 1 | 30/12/2013 | 223937.96790 | 230079.62100 | 6827 | 2013-12-30 |
| 2 | 31/12/2013 | 300345.48460 | 306986.12050 | 9268 | 2013-12-31 |
| 3 | 1/1/2014 | 86782.46773 | 87986.31821 | 2678 | 2014-01-01 |
| 4 | 2/1/2014 | 200173.11680 | 202881.17770 | 6175 | 2014-01-02 |
| 5 | 3/1/2014 | 326906.07420 | 333806.29190 | 10084 | 2014-01-03 |
| 6 | 4/1/2014 | 307043.93540 | 313652.34830 | 9470 | 2014-01-04 |
| 7 | 5/1/2014 | 179188.88450 | 183083.28500 | 5524 | 2014-01-05 |
| 8 | 6/1/2014 | 306351.95090 | 313446.06810 | 9448 | 2014-01-06 |
| 9 | 7/1/2014 | 224020.41000 | 229533.97370 | 6911 | 2014-01-07 |
| 10 | 8/1/2014 | 101836.40220 | 99381.96752 | 2990 | 2014-01-08 |
| 11 | 9/1/2014 | 107804.30490 | 104816.45850 | 3159 | 2014-01-09 |
| 12 | 10/1/2014 | 144829.95670 | 141333.99630 | 4246 | 2014-01-10 |

3. Min and Max dates to understand the time frame.

The screenshot shows a SQL IDE interface. On the left, a sidebar displays a database schema with the following structure:

- CASE_STUDY_SALES
 - INFORMATION_SCHEMA
 - PUBLIC
 - Tables
 - SALES
 - SALES_TABLE
 - SALES_TRENDS
 - SALES_TRENDS_FINAL
- SNOWFLAKE
- SNOWFLAKE_LEARNING_DB
- SNOWFLAKE_SAMPLE_DATA

The main editor displays the following SQL query:

```
SELECT MIN(FORMATTED_DATE) AS FIRST_TRANSACTION,  
       MAX (FORMATTED_DATE) AS FINAL_TRANSACTION  
FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS;
```

The 'Results' tab is active, showing a table with two columns: FIRST_TRANSACTION and FINAL_TRANSACTION. The data is as follows:

| | FIRST_TRANSACTION | FINAL_TRANSACTION |
|---|-------------------|-------------------|
| 1 | 2013-12-30 | 2016-11-16 |

On the right, the 'Query Details' panel shows a query duration of 1674ms and a query ID of 01bc8f40-0001-05f4-0...

4. Dropping the old date and leaving the new, formatted date.

The screenshot shows the same SQL IDE interface. The main editor displays the following SQL query:

```
ALTER TABLE SALES_TRENDS DROP DATE;  
SELECT *  
FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS;
```

The 'Results' tab is active, but it displays an error message: "column 'DATE' does not exist".

5. Moving to two decimals for monetary values and were applicable to.

The screenshot shows a SQL IDE interface. On the left, a sidebar displays a database schema with 'CASE_STUDY_SALES' as the selected database. The main editor contains a SQL query that selects various fields from 'SALES_TRENDS' and applies truncation to monetary values. The 'Results' tab is active, showing a table with 12 rows of data. The columns are: FORMATTED_DATE, DAY_OF_THE_WEEK, # SALES, # COST_OF_SALES, # QUANTITY_SOLD, and # PRICE_PER_UNIT. The data spans from 2013-12-30 to 2014-01-10. A 'Query Details' panel on the right shows the query ID and duration.

```
SELECT FORMATTED_DATE,
       TO_CHAR(FORMATTED_DATE, 'DY') AS DAY_OF_THE_WEEK,
       TRUNCATE(SALES, 2) AS SALES,
       TRUNCATE(COST_OF_SALES, 2) AS COST_OF_SALES,
       QUANTITY_SOLD,
       TRUNCATE (SALES/QUANTITY_SOLD, 2) AS PRICE_PER_UNIT,
FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS;
```

| | FORMATTED_DATE | DAY_OF_THE_WEEK | # SALES | # COST_OF_SALES | # QUANTITY_SOLD | # PRICE_PER_UNIT |
|----|----------------|-----------------|-----------|-----------------|-----------------|------------------|
| 1 | 2013-12-30 | Mon | 223937.96 | 230079.62 | 6827 | 32.80 |
| 2 | 2013-12-31 | Tue | 300345.48 | 306986.12 | 9268 | 32.40 |
| 3 | 2014-01-01 | Wed | 86782.46 | 87986.31 | 2678 | 32.40 |
| 4 | 2014-01-02 | Thu | 200173.11 | 202881.17 | 6175 | 32.41 |
| 5 | 2014-01-03 | Fri | 326906.07 | 333806.29 | 10084 | 32.41 |
| 6 | 2014-01-04 | Sat | 307043.93 | 313652.34 | 9470 | 32.42 |
| 7 | 2014-01-05 | Sun | 179188.88 | 183083.28 | 5524 | 32.43 |
| 8 | 2014-01-06 | Mon | 306351.95 | 313446.06 | 9448 | 32.42 |
| 9 | 2014-01-07 | Tue | 224020.41 | 229533.97 | 6911 | 32.41 |
| 10 | 2014-01-08 | Wed | 101836.40 | 99381.96 | 2990 | 34.05 |
| 11 | 2014-01-09 | Thu | 107804.30 | 104816.45 | 3159 | 34.12 |
| 12 | 2014-01-10 | Fri | 144829.95 | 141333.99 | 4246 | 34.10 |

6. Creating a new table to include all the changes

The screenshot shows the same SQL IDE interface. The main editor now contains a SQL query that creates or replaces a table named 'SALES_TRENDS_FINAL' with the same structure and data as the previous query. The 'Results' tab is active, showing a single row with the status 'Table SALES_TRENDS_FINAL successfully created.' The 'Query Details' panel on the right shows the query ID and duration.

```
---CREATING ANOTHER TABLE
CREATE OR REPLACE TABLE SALES_TRENDS_FINAL AS (
SELECT *,
       TO_CHAR(FORMATTED_DATE, 'DY') AS DAY_OF_THE_WEEK,
       TRUNCATE(SALES, 2) AS SALES_FINAL,
       TRUNCATE(COST_OF_SALES, 2) AS COST_OF_SALES_FINAL,
       TRUNCATE (SALES/QUANTITY_SOLD, 2) AS PRICE_PER_UNIT
FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS);
```

| status |
|--|
| 1 Table SALES_TRENDS_FINAL successfully created. |

7. Dropping columns to leave on the final column of COS and SALES.

2025-05-17 12:31am

+

▼

Databases

Worksheets

ACCOUNTADMIN

COMPUTE_WH (X-Small)

Share

Search objects

↺

CASE_STUDY_SALES

▼

INFORMATION_SCHEMA

▼

PUBLIC

▼

Tables

SALES

SALES_TABLE

SALES_TRENDS

SALES_TRENDS_FINAL

SNOWFLAKE

SNOWFLAKE_LEARNING_DB

SNOWFLAKE_SAMPLE_DATA

CASE_STUDY_SALES.PUBLIC

Settings

Code View

61

62

63

64

65

66

67

68

69

70

71

72

73

--DROPPING COLUMNS

ALTER TABLE SALES_TRENDS_FINAL DROP SALES, COST_OF_SALES;

SELECT *

FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS_FINAL;

Results

Chart

| | # QUANTITY_SOLD | FORMATTED_DATE | DAY_OF_THE_WEEK | # SALES_FINAL | # COST_OF_SALES_FINAL | # PRICE_PER_UNIT |
|----|-----------------|----------------|-----------------|---------------|-----------------------|------------------|
| 1 | 6827 | 2013-12-30 | Mon | 223937.96 | 230079.62 | 32.80 |
| 2 | 9268 | 2013-12-31 | Tue | 300345.48 | 306986.12 | 32.40 |
| 3 | 2678 | 2014-01-01 | Wed | 86782.46 | 87986.31 | 32.40 |
| 4 | 6175 | 2014-01-02 | Thu | 200173.11 | 202881.17 | 32.41 |
| 5 | 10084 | 2014-01-03 | Fri | 326906.07 | 333806.29 | 32.41 |
| 6 | 9470 | 2014-01-04 | Sat | 307043.93 | 313652.34 | 32.42 |
| 7 | 5524 | 2014-01-05 | Sun | 179188.88 | 183083.28 | 32.43 |
| 8 | 9448 | 2014-01-06 | Mon | 306351.95 | 313446.06 | 32.42 |
| 9 | 6911 | 2014-01-07 | Tue | 224020.41 | 229533.97 | 32.41 |
| 10 | 2990 | 2014-01-08 | Wed | 101836.40 | 99381.96 | 34.05 |
| 11 | 3159 | 2014-01-09 | Thu | 107804.30 | 104816.45 | 34.12 |
| 12 | 4246 | 2014-01-10 | Fri | 144829.95 | 141333.99 | 34.10 |

Query Details

Query duration

Rows

Query ID 01bc6f50-00

Show more

QUANTITY_SOLD

508

FORMATTED_DATE

8. Counting the number of days.

2025-05-17 12:31am

+

▼

Databases

Worksheets

ACCOUNTADMIN

COMPUTE_WH (X-Small)

Share

Search objects

↺

CASE_STUDY_SALES

▼

INFORMATION_SCHEMA

▼

PUBLIC

▼

Tables

SALES

SALES_TABLE

SALES_TRENDS

SALES_TRENDS_FINAL

SNOWFLAKE

SNOWFLAKE_LEARNING_DB

SNOWFLAKE_SAMPLE_DATA

CASE_STUDY_SALES.PUBLIC

Settings

Code View

73

74

75

76

77

78

79

80

81

82

83

84

85

--AVERAGE UNIT SALES

SELECT COUNT (FORMATTED_DATE) AS NUMBER_OF_UNITS

FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS_FINAL;

Results

Chart

| # NUMBER_OF_UNITS |
|-------------------|
| 1053 |

Query Details

Query duration

Rows

Query ID 01bc6f53-00

Show more

NUMBER_OF_UNITS

100% filled

9. Answering question 2 from the case study.

The screenshot shows the Snowflake SQL interface. The left sidebar displays the database schema: CASE_STUDY_SALES, INFORMATION_SCHEMA, PUBLIC, and SNOWFLAKE. The main editor shows a SQL query for question 2:

```
--QUESTION 2
87 SELECT SUM(PRICE_PER_UNIT)/1053 AS AVERAGE_UNIT_SALES_PRICE
88 FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS_FINAL;
```

The query results are displayed in a table with two columns: AVERAGE_UNIT_SALES_PRICE. The result is 37.06782526. The query details panel on the right shows a query duration of 68ms and 1 row.

| AVERAGE_UNIT_SALES_PRICE |
|--------------------------|
| 37.06782526 |

10. Answering question 3 from the case study.

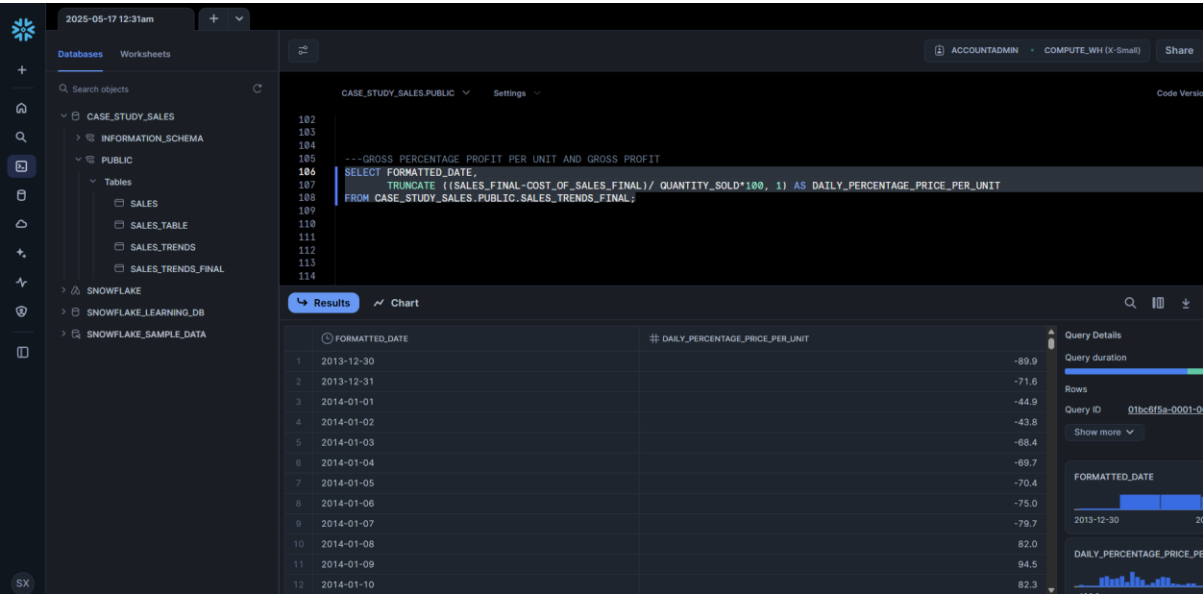
The screenshot shows the Snowflake SQL interface. The left sidebar displays the database schema: CASE_STUDY_SALES, INFORMATION_SCHEMA, PUBLIC, and SNOWFLAKE. The main editor shows a SQL query for question 3:

```
-- GROSS PERCENTAGE PROFIT AND GROSS PROFIT
98 SELECT FORMATTED_DATE,
99 TRUNCATE((SALES_FINAL-COST_OF_SALES_FINAL) /COST_OF_SALES_FINAL*100, 1) AS DAILY_PERCENTAGE_GROSS_PROFIT
100 FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS_FINAL;
```

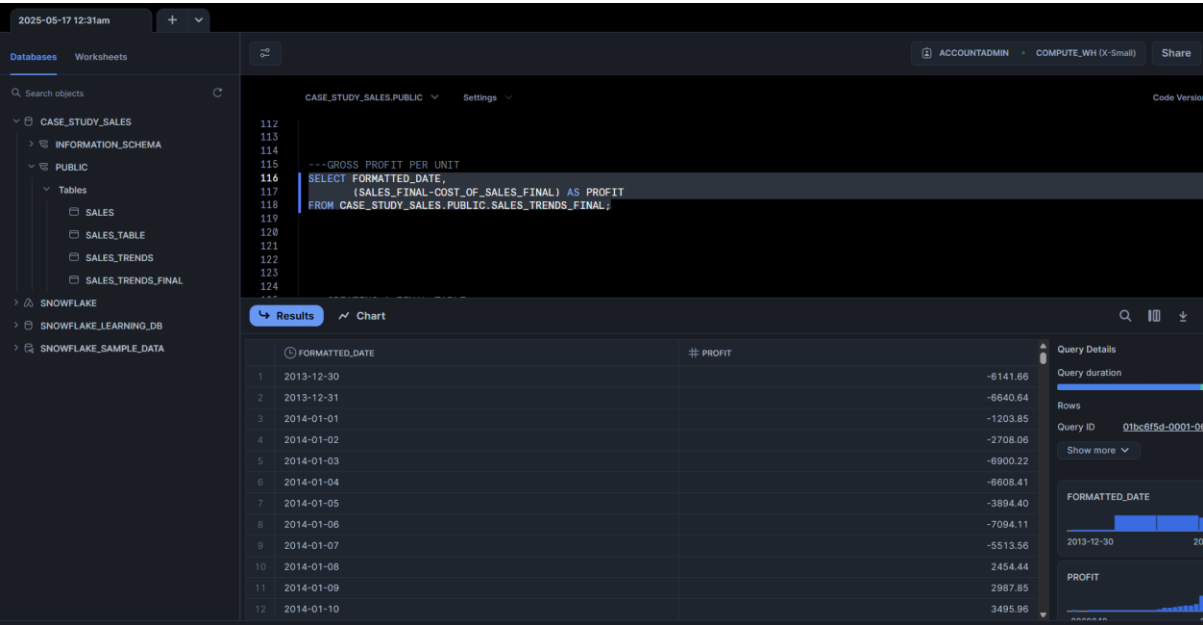
The query results are displayed in a table with two columns: FORMATTED_DATE and DAILY_PERCENTAGE_GROSS_PROFIT. The results show data for the period from 2013-12-30 to 2014-01-09.

| FORMATTED_DATE | DAILY_PERCENTAGE_GROSS_PROFIT |
|----------------|-------------------------------|
| 2013-12-30 | -2.6 |
| 2013-12-31 | -2.1 |
| 2014-01-01 | -1.3 |
| 2014-01-02 | -1.3 |
| 2014-01-03 | -2.0 |
| 2014-01-04 | -2.1 |
| 2014-01-05 | -2.1 |
| 2014-01-06 | -2.2 |
| 2014-01-07 | -2.4 |
| 2014-01-08 | 2.4 |
| 2014-01-09 | 2.8 |

11. Answering question 4 from the case study.



12. Calculating profit for the final output table.



13. This is the final output table that will be used in Excel for pivot tables and graphs.

The screenshot shows the Snowflake SQL Editor interface. On the left, the 'Databases' sidebar is expanded to show the 'CASE_STUDY_SALES' database, with the 'PUBLIC' schema selected. The 'Tables' section lists 'SALES', 'SALES_TABLE', 'SALES_TRENDS', and 'SALES_TRENDS_FINAL'. The main editor displays a SQL script to create a final table named 'SALES_TABLE'. The script includes a comment '---CREATING A FINAL TABLE', followed by a 'CREATE OR REPLACE TABLE' statement. The table is populated with data from 'SALES_TRENDS_FINAL', including columns for formatted date, day of the week, sales, cost of sales, price per unit, quantity sold, and daily percentage gross profit. The 'Results' tab shows a single status message: 'Table SALES_TABLE successfully created.' The 'Query Details' panel on the right indicates a query duration of 50ms and a 100% fill status.

```
124 ---CREATING A FINAL TABLE
125
126 CREATE OR REPLACE TABLE SALES_TABLE AS (
127   SELECT FORMATTED_DATE,
128     DAY_OF_THE_WEEK,
129     TRUNCATE (SALES_FINAL) AS SALES,
130     TRUNCATE (COST_OF_SALES_FINAL) AS COST_OF_SALES,
131     TRUNCATE (SALES/QUANTITY_SOLD) AS PRICE_PER_UNIT,
132     QUANTITY_SOLD,
133     TRUNCATE ((SALES_FINAL-COST_OF_SALES_FINAL) /COST_OF_SALES_FINAL*100) AS DAILY_PERCENTAGE_GROSS_PROFIT,
134     TRUNCATE ((SALES_FINAL-COST_OF_SALES_FINAL) / QUANTITY_SOLD*100) AS DAILY_PERCENTAGE_PRICE_PER_UNIT,
135     TRUNCATE (SALES_FINAL-COST_OF_SALES_FINAL) AS PROFIT
136   FROM CASE_STUDY_SALES.PUBLIC.SALES_TRENDS_FINAL );
137
```

| status |
|---|
| 1 Table SALES_TABLE successfully created. |

Query Details
Query duration 50ms
Rows
Query ID 01bc8f62-0001-0773-0...
Show more
status
100% filled

14. Verifying the final output table.

The screenshot shows the Snowflake SQL Editor interface with the 'Databases' sidebar expanded to the 'PUBLIC' schema. The 'Tables' section lists 'SALES', 'SALES_TABLE', 'SALES_TRENDS', and 'SALES_TRENDS_FINAL'. The main editor displays a SQL query to select all data from the 'SALES_TABLE'. The 'Results' tab shows a table with 11 rows of data. The 'Query Details' panel on the right indicates a query duration of 16ms and a 100% fill status.

```
141 SELECT *
142 FROM SALES_TABLE;
143
144
145
146
147
148
149
150
151
152
153
```

| | FORMATTED_DATE | DAY_OF_THE_WEEK | # SALES | # COST_OF_SALES | # PRICE_PER_UNIT | # QUANTITY_SOLD | # DAILY_PERCENTAGE_GROSS |
|----|----------------|-----------------|---------|-----------------|------------------|-----------------|--------------------------|
| 1 | 2013-12-30 | Mon | 223937 | 230079 | 32 | 6827 | |
| 2 | 2013-12-31 | Tue | 300345 | 306986 | 32 | 9268 | |
| 3 | 2014-01-01 | Wed | 86782 | 87986 | 32 | 2678 | |
| 4 | 2014-01-02 | Thu | 200173 | 202881 | 32 | 6175 | |
| 5 | 2014-01-03 | Fri | 326906 | 333806 | 32 | 10084 | |
| 6 | 2014-01-04 | Sat | 307043 | 313652 | 32 | 9470 | |
| 7 | 2014-01-05 | Sun | 179188 | 183083 | 32 | 5524 | |
| 8 | 2014-01-06 | Mon | 306351 | 313446 | 32 | 9448 | |
| 9 | 2014-01-07 | Tue | 224020 | 229533 | 32 | 6911 | |
| 10 | 2014-01-08 | Wed | 101836 | 99381 | 34 | 2990 | |
| 11 | 2014-01-09 | Thu | 107804 | 104816 | 34 | 3159 | |
| 12 | 2014-01-10 | Fri | 144850 | 141533 | 34 | 4248 | |

Query Details
Query duration 16ms
Rows
Query ID 01bc8f65-0001-0658...
Show more
FORMATTED_DATE
2013-12-30 2014-01-10
DAY_OF_THE_WEEK
Mon

15. Answering question 5 from the case study.

2025-05-17 12:31am

Databases

Worksheets

Search objects

CASE_STUDY_SALES

INFORMATION_SCHEMA

PUBLIC

Tables

SALES

SALES_TABLE

SALES_TRENDS

SALES_TRENDS_FINAL

SNOWFLAKE

SNOWFLAKE_LEARNING_DB

SNOWFLAKE_SAMPLE_DATA

CASE_STUDY_SALES PUBLIC

Settings

Code Versions

152

153

154

155

156

157

158

159

160

161

162

163

164

QUESTION 5

SELECT FORMATTED_DATE,

PRICE_PER_UNIT

FROM SALES_TABLE

ORDER BY PRICE_PER_UNIT ASC

LIMIT 3;

Results

Chart

| | FORMATTED_DATE | PRICE_PER_UNIT | |
|---|----------------|----------------|----|
| 1 | 2014-08-27 | | 30 |
| 2 | 2014-08-28 | | 30 |
| 3 | 2014-08-29 | | 30 |

Query Details

...

Query duration 237ms

Rows 3

Query ID 01bc8f68-0001-0770-0-

Show more

FORMATTED_DATE

2014-08-27 2014-08-29

PRICE_PER_UNIT

#