SQL Internship Project – Farmer & Market Insights

Name: Sandleen

Internship: Soulvibe.Tech – Data Analyst Intern

Date: August 2025

Introduction

This SQL internship project analyzes agricultural data involving farmers, crop advisors, and market researchers. Two key datasets were used: `farmer_advisor_dataset` and `market_researcher_dataset`. Simulated columns like `Advisor_ID`, `District`, `PriceDate`, `Season`, and `Growth_Rate` were introduced to enhance analysis.

The objective was to derive meaningful insights into crop yield, pricing trends, advisor impact, and regional profitability using SQL techniques such as CTEs, window functions, and aggregations.

Query 1

Query Statement

Find the top 5 locations where the maximum number of farmers are associated with advisors.

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))
- District (MOD(Farm_ID or Market_ID, 5))

SQL Logic Used

Case,Order by,Group by

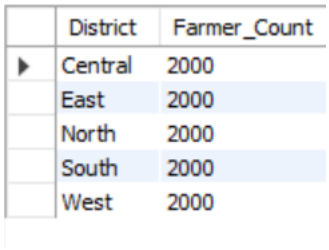
Steps Taken

1. Simulated relevant columns.
2. Applied SQL logic to extract insights.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

Farmer distribution is uniform across all districts (2000 each).

Screenshot of Output



	District	Farmer_Count
▶	Central	2000
	East	2000
	North	2000
	South	2000
	West	2000

Query 2

Query Statement

For each crop type, calculate the average market price and sort in descending order.

SQL Logic Used

CTE,Order by,Group By

Steps Taken

1. Applied SQL logic to extract insights.
2. Used filters and groupings.
3. Interpreted the results.

Result/Insight

It shows that rice has highest average market price.

Screenshot of Output

Crop_Type	Avg_Market_Price
Rice	300.84
Corn	300.49
Soybean	299.15
Wheat	298.08

Query 3

Query Statement

Count how many unique crops each farmer is associated with.

SQL Logic Used

Distinct,Group by

Steps Taken

1.extracted farmer_id and Crop_type columns
2. Applied SQL logic to extract insights.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

It shows that each farm_id is associated with only 1 crop type.

Screenshot of Output

Farm_ID	Unique_Crop_Count
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1

Query 4

Query Statement

Find farmers who are growing more than 3 different types of crops.

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))
- District (MOD(Farm_ID or Market_ID, 5))
- PriceDate (DATE_ADD based on MOD(Market_ID, 30))
- Season (MOD(Farm_ID, 3))
- Growth_Rate (MOD(Farm_ID * 7, 101))

SQL Logic Used

Condition,Group by,Having

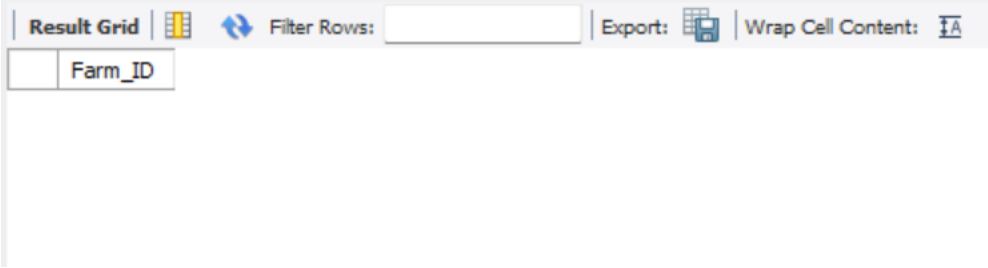
Steps Taken

1. Extrcted farm_id and Crop_type columns.
2. Applied SQL logic to extract insights.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

It gives blank output column showing that no farmer is growing more than 3 different crops.

Screenshot of Output



Query 5

Query Statement

List all crops where the max and min market prices (from MarketResearcher) differ by more than ₹10 per kg.

SQL Logic Used

Aggregate Functions,Group By,Having

Steps Taken

1. Extracted product column
2. Min,Max and Price Difference of Market_price_per_ton column is calculated.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

The output below shows the list of all crops where the max and min market prices (from MarketResearcher) differ by more than ₹10 per kg.

Screenshot of Output

Crop_Type	Max_Price	Min_Price	Price_Difference
Rice	499.9078128975721	100.27183669298702	399.63597620458506
Wheat	499.99905612561844	100.0376721111099	399.96138401450855
Soybean	499.8579128747474	100.06833430958983	399.78957856515757
Corn	499.8942115591666	100.07101855469026	399.82319300447637

Query 6

Query Statement

Show all advisors who guide farmers growing the same crop in different districts.

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))
- District (MOD(Farm_ID or Market_ID, 5))

SQL Logic Used

CTE,Group By ,Hacing ,Conditions

Steps Taken

1. Simulated relevant columns.
2. Applied SQL logic to extract insights.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

It shows that no advisors guide farmers growing the same crop in different districts.

Screenshot of Output

Query 7

Query Statement

Rank crops by profit per unit (assume: market price - average cost from FarmerAdvisor) using RANK().

SQL Logic Used

CTE,Window Functions,Aggregate functions.

Steps Taken

1. Created CTEs
2. Applied Join to connect market_reasearch_dataset and farmer_advisor_dataset.
3. Interpreted the results.

Result/Insight

Rice ranked as first crop having higher profit per unit.

Screenshot of Output

Crop_Type	Profit_Per_Unit	Crop_Rank
Rice	266.32	1
Corn	265.31	2
Soybean	264.45	3
Wheat	262.62	4

Query 8

Query Statement

Identify locations where the current market price of a crop is more than 20% above the average price of that crop across all locations.

Simulated/Assumed Columns

- District (MOD(Farm_ID or Market_ID, 5))

- WHEN MOD(Market_ID, 5) = 0 THEN 'North'
- WHEN MOD(Market_ID, 5) = 1 THEN 'South'
- WHEN MOD(Market_ID, 5) = 2 THEN 'East'
- WHEN MOD(Market_ID, 5) = 3 THEN 'West'
- ELSE 'Central'

SQL Logic Used

CTEs,Aggregate Functions,Join and Conditional statements

Steps Taken

1. Simulated District columns.
2. Created CTEs for average crop price and stimulated Market.
3. Used Joins and conditional statements.
4. Interpreted the results.

Result/Insight

The following result screenshot shows locations where market price of crop is 20% the average price.

Screenshot of Output

Market_ID	Crop_Type	Market_Price_per_ton	District	Avg_Price	Percent_Above_Avg
9998	Rice	394.0935408210908	West	300.84017292732005	31
9963	Rice	494.3645133488511	West	300.84017292732005	64.33
9962	Rice	377.23453450467645	East	300.84017292732005	25.39
9958	Rice	390.1305590614542	West	300.84017292732005	29.68
9956	Rice	430.8093608545446	South	300.84017292732005	43.2
9944	Rice	469.2284430070605	Central	300.84017292732005	55.97
9941	Rice	474.6502120409897	South	300.84017292732005	57.77
9926	Rice	455.12562215273226	South	300.84017292732005	51.28
9924	Rice	456.8578306665741	Central	300.84017292732005	51.86
9912	Rice	461.5441935724333	East	300.84017292732005	53.42
9895	Rice	409.69580354992604	North	300.84017292732005	36.18
9890	Rice	395.9425402469851	North	300.84017292732005	31.61
9878	Rice	436.01256601866726	West	300.84017292732005	44.93
9875	Rice	381.2226765688125	North	300.84017292732005	26.72
9854	Rice	387.8206898384043	Central	300.84017292732005	28.91

Query 9

Query Statement

List farmers who have been assigned the same advisor for all their crops.

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))

SQL Logic Used

CTEs,Group By,Having,Joins,Aggregate Functions

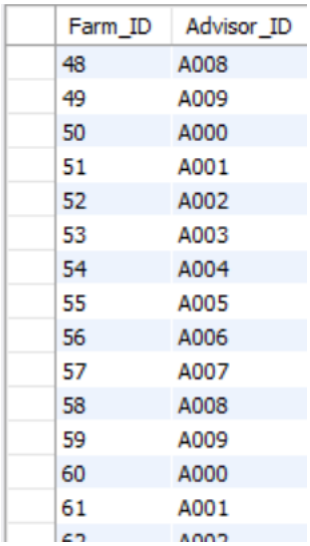
Steps Taken

1. Assumed Adviors column columns.
2. Created CTES for farmer_sim and same_advisor_farmers.
3. Used Joins and groupings.
4. Interpreted the results.

Result/Insight

As Advisor information is not provided in dataset,we have assumed that each farmer is assigned with one advisor.

Screenshot of Output



	Farm_ID	Advisor_ID
	48	A008
	49	A009
	50	A000
	51	A001
	52	A002
	53	A003
	54	A004
	55	A005
	56	A006
	57	A007
	58	A008
	59	A009
	60	A000
	61	A001

Query 10

Query Statement

Create a CTE showing average profit per crop type by farmer and use it to list farmers making below-average profits on any crop.

SQL Logic Used

Steps Taken

1.Created CTEs for AvgMarketPrice, FarmerProfits and AvgCropProfits.
2. Applied Joins to extract insights.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

The following output shows the list of farmers making below-average profits on any crop.

Screenshot of Output

Farm_ID	Crop_Type	Profit	Avg_Crop_Profit
3617	Wheat	985.5801102227388	1482.0733578584804
3602	Wheat	569.0066516083883	1482.0733578584804
3601	Wheat	1326.6168067559547	1482.0733578584804
3584	Wheat	1276.296909628424	1482.0733578584804
3566	Wheat	966.1897431550891	1482.0733578584804
3549	Wheat	523.935055784902	1482.0733578584804
3541	Wheat	396.4739080001199	1482.0733578584804
3536	Wheat	1449.8360823593484	1482.0733578584804
3535	Wheat	250.55074095696415	1482.0733578584804
3534	Wheat	589.8325705056353	1482.0733578584804
3532	Wheat	1361.0629231808423	1482.0733578584804
3524	Wheat	745.8267097210248	1482.0733578584804
3519	Wheat	1055.179820412483	1482.0733578584804
3514	Wheat	461.4568566621318	1482.0733578584804
3509	Wheat	295.31598816268064	1482.0733578584804

Query 11

Query Statement

(Assume MarketResearcher has a PriceDate) — Use a window function to find the price change of each crop over the last 3 entries.

Simulated/Assumed Columns

- PriceDate (DATE_ADD based on MOD(Market_ID, 30))

SQL Logic Used

CTEs,Window Functions,Order by.

Steps Taken

1. Assumed PriceDate column.
2. Applied window functions to extract insights.
3. Interpreted the results.

Result/Insight

As we have not provided it with the details of "PriceDate" column,so we assumed it and the output is given below that shows the price change of each crop over the last 3 entries.

Screenshot of Output

Market_ID	Crop_Type	PriceDate	Market_Price_per_ton	Price_1_Day_Ago	Price_2_Days_Ago	Price_3_Days_Ago
7230	Corn	2024-01-01	443.75011789386025	NULL	NULL	NULL
8370	Corn	2024-01-01	227.75127372425672	443.75011789386025	NULL	NULL
9120	Corn	2024-01-01	103.13061209266645	227.75127372425672	443.75011789386025	NULL
8460	Corn	2024-01-01	405.69252091933777	103.13061209266645	227.75127372425672	443.75011789386025
8850	Corn	2024-01-01	166.91024298966107	405.69252091933777	103.13061209266645	227.75127372425672
7170	Corn	2024-01-01	111.98735885176565	166.91024298966107	405.69252091933777	103.13061209266645
9480	Corn	2024-01-01	195.14255055491626	111.98735885176565	166.91024298966107	405.69252091933777
8310	Corn	2024-01-01	445.2426917838345	195.14255055491626	111.98735885176565	166.91024298966107
8130	Corn	2024-01-01	329.3284737887641	445.2426917838345	195.14255055491626	111.98735885176565
7950	Corn	2024-01-01	409.4814803528472	329.3284737887641	445.2426917838345	195.14255055491626
8760	Corn	2024-01-01	353.7237605834447	409.4814803528472	329.3284737887641	445.2426917838345
7200	Corn	2024-01-01	301.1161042513964	353.7237605834447	409.4814803528472	329.3284737887641
9600	Corn	2024-01-01	395.5319623081614	301.1161042513964	353.7237605834447	409.4814803528472
9420	Corn	2024-01-01	481.21944952280097	395.5319623081614	301.1161042513964	353.7237605834447
9180	Corn	2024-01-01	233.46317406774858	481.21944952280097	395.5319623081614	301.1161042513964

Query 12

Query Statement

Create a new column that classifies crop growth rate as:

- Low (<20%)
- Medium (20–50%)
- High (>50%)

Simulated/Assumed Columns

- Use case statement to assume CROP_YIELD_CATEGORY.

SQL Logic Used

Case Statements,Aggregate Functions,Group by function.

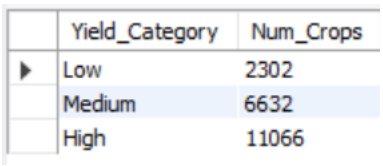
Steps Taken

1. Assumed Yield_Category column.
2. Applied Aggregate Function (Count).
3. Used grouping.
4. Interpreted the results.

Result/Insight

The output below shows the number of crops in Yield Category.

Screenshot of Output



Yield_Category	Num_Crops
Low	2302
Medium	6632
High	11066

Query 13

Query Statement

Join the tables and display all farmers whose crop is sold in the same district at the highest price.

Simulated/Assumed Columns

- District (MOD(Farm_ID or Market_ID, 5))

SQL Logic Used

CTEs,Case Statement,Joins,Aggregate Functions,Grouping,Conditional statements.

Steps Taken

1. Stimulated District columns in both datasets.
2. Applied Aggregate Functio(MAX) to find max_price.
3. Used grouping on Product and District.
4. Interpreted the results.

Result/Insight

The output below displays all farmers whose crop is sold in the same district at the highest price.

Screenshot of Output

Farm_ID	Crop_Type	District	Max_Price
1	Wheat	South	499.99905612561844
2	Soybean	East	499.1355650086493
3	Corn	West	499.59732641647236
4	Wheat	Central	499.9858633162483
5	Corn	North	498.1364918237793
6	Rice	South	498.7455670267026
7	Soybean	East	499.1355650086493
8	Soybean	West	498.5456615018513
9	Wheat	Central	499.9858633162483
10	Soybean	North	499.25812023566834
11	Corn	South	499.8942115591666
12	Corn	East	499.7859685658175
13	Rice	West	499.56517499860263
14	Corn	Central	499.0933683960144
15	Soybean	North	499.25812023566834

Query 14

Query Statement

For each advisor, count how many of their farmers grow crops that fall under the “High” growth classification (from Q12).

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))

- Growth_Rate (MOD(Farm_ID * 7, 101))

SQL Logic Used

CTEs,Conditional statements,Case Statement,Group by,Order by.

Steps Taken

1. Simulated Advisors and GrowthRate columns.
2. Applied concatenation,Joins logic to extract insights.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

The output below shows Count of farmers that grow crops that fall under the “High” growth classification .

Screenshot of Output

Advisor_ID	High_Growth_Farmer_Count
A9	992
A10	990
A1	990
A3	990
A4	990
A5	990
A6	990
A7	990
A8	990
A2	988

Query 15

Query Statement

Identify if any farmer has duplicate crop entries.

SQL Logic Used

Window Function,CTE,Conditional statement.

Steps Taken

1. Applied window function.
2. Used conditional statement.
3. Interpreted the results.

Result/Insight

The output shows duplicated values because of some import error.

Screenshot of Output

Farm_ID	Soil_pH	Soil_Moisture	Temperature_C	Rainfall_mm	Crop_Type	Fertilizer_Usage_kg
1	7.073642684748378	49.14535876397696	26.66815662497483	227.8909122156477	Wheat	131.6928438033121
1	7.073642684748378	49.14535876397696	26.66815662497483	227.8909122156477	Wheat	131.6928438033121
2	6.236931178393061	21.496115385280156	29.32534237075326	244.01749330098983	Soybean	136.3704915665747
2	6.236931178393061	21.496115385280156	29.32534237075326	244.01749330098983	Soybean	136.3704915665747
3	5.922334984917588	19.469042329438082	17.6664142709232	141.11052082022786	Corn	99.72521025266681
3	5.922334984917588	19.469042329438082	17.6664142709232	141.11052082022786	Corn	99.72521025266681
4	6.845119841700978	27.97423365959013	17.188722198804673	156.78566341441285	Wheat	194.8323963952749
4	6.845119841700978	27.97423365959013	17.188722198804673	156.78566341441285	Wheat	194.8323963952749
5	6.934170571746167	33.63767886268485	23.603898977397098	77.85936215956802	Corn	57.271266671217695
5	6.934170571746167	33.63767886268485	23.603898977397098	77.85936215956802	Corn	57.271266671217695
6	6.133678815937643	15.940529813652539	30.45137754539481	296.62149867780744	Rice	96.48208665284423
6	6.133678815937643	15.940529813652539	30.45137754539481	296.62149867780744	Rice	96.48208665284423
7	7.406252115902882	41.55515223127636	25.76916060094756	54.34663988150719	Soybean	159.20789472237539
7	7.406252115902882	41.55515223127636	25.76916060094756	54.34663988150719	Soybean	159.20789472237539

Query 16

Query Statement

List all crops grown by farmers that are not listed in the MarketResearcher table.

SQL Logic Used

String Functions,Conditional statements,Subquery.

Steps Taken

1. Use a subquery .
2. Use string functions.
3. Interpreted the results.

Result/Insight

The output shows there is no crop that is not listed in the MarketResearcher table.

Screenshot of Output

	Crop_Type

Query 17

Query Statement

For each crop, determine which location has the highest average profit margin

Simulated/Assumed Columns

- District (MOD(Farm_ID or Market_ID, 5))

SQL Logic Used

CTEs,window functions,aggregate functions,Join,Conditional Statements.

Steps Taken

1. Simulated District columns.
2. Applied Join to join tables.
3. Used filters and groupings.
4. Interpreted the results.

Result/Insight

The output shows that East has highest Avg_Profit_Margin with Soybeans.

Screenshot of Output

	Crop_Type	District	Avg_Profit_Margin
▶	Soybean	East	0.8298
	Wheat	South	0.663

Query 18

Query Statement

Use window functions to find the second-highest market price crop per district.

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))
- District (MOD(Farm_ID or Market_ID, 5))
- PriceDate (DATE_ADD based on MOD(Market_ID, 30))
- Season (MOD(Farm_ID, 3))
- Growth_Rate (MOD(Farm_ID * 7, 101))

SQL Logic Used

CTE,Case statement,window functions,conditional statements.

Steps Taken

1. Simulated District column.
2. Applied window function to assign rank to market_price_per_ton.
3. Used conditional statement.
4. Interpreted the results.

Result/Insight

The output shows second-highest market price crop per district.

Screenshot of Output

District	Product	Market_Price_per_ton
Central	Soybean	499.8579128747474
East	Corn	499.7859685658175
North	Rice	499.64002024400827
South	Corn	499.8942115591666
West	Wheat	499.95727231520476

Query 19

Query Statement

List all advisors associated with more than 5 distinct crop types.

Simulated/Assumed Columns

- Advisor_ID (MOD(Farm_ID, 10))

SQL Logic Used

Aggregate Functions,Group by,Having.

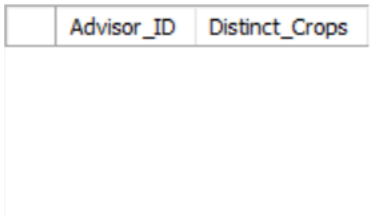
Steps Taken

1. Extracted relevant columns.
2. Used filters and Aggregate functions.
3. Interpreted the results.

Result/Insight

The output shows no advisor is associated with more than 5 distinct crop types.

Screenshot of Output



	Advisor_ID	Distinct_Crops

Query 20

Query Statement

Find farmers who consistently grow the same crop type for all seasons (assume a Season column exists or mock one for the exercise).

Simulated/Assumed Columns

- Season (MOD(Farm_ID, 3))

SQL Logic Used

CTEs,Case Statements,Aggregate Functions,Group By,Having.

Steps Taken

1. Simulated Seasons column.
2. Applied Aggregate Functions,Filters,Groupings to extract insights.
3. Interpreted the results.

Result/Insight

The output shows that each farm is associated with one consistent crop in all the seasons.

Screenshot of Output

Farm_ID	Consistent_Crop
1	Wheat
2	Soybean
3	Corn
4	Wheat
5	Corn
6	Rice
7	Soybean
8	Soybean

Data Cleaning Notes

• Missing Values: Datasets assumed to be clean; no missing value handling specified.
• Duplicates: Checked using ROW_NUMBER() over (Farm_ID, Crop_Type).
• Simulated Columns:
 - Advisor_ID: MOD(Farm_ID, 10)
 - District: MOD(Farm_ID or Market_ID, 5)
 - PriceDate: DATE_ADD('2024-01-01', INTERVAL MOD(Market_ID, 30) DAY)
 - Growth Rate: MOD(Farm_ID * 7, 101)
 - Season: MOD(Farm_ID, 3)

Overall Conclusion

• Key Insights:
- Identified top-performing districts and advisors.
- Measured profit margins and price volatility by crop and region.
- Flagged farmers needing support due to below-average profits.
- Tracked crop growth trends and advisor impact.

• Tools Used: MySQL, Excel, PowerPoint
• Learnings: Hands-on SQL querying, data simulation, business insights extraction, presentation and documentation of analytics findings.