
Database Management System (IT615)

DDL & SQL
for
Sustainable Agriculture Resource Management

Group Number: 02
Aarushi Goel (202412002)
Jayesh Chauhan (202412012)

- **SQL Queries (20 Simple SQL Query)**

1.

```
SELECT COUNT(FarmerID) AS NumberOfFarmers
FROM "G2SARM".Farmer_TrainingProgram
WHERE ProgramID BETWEEN 10 AND 50;
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 SELECT COUNT(FarmerID) AS NumberOfFarmers
2 FROM "G2SARM".Farmer_TrainingProgram
3 WHERE ProgramID BETWEEN 10 AND 50;
```

The results pane displays a single row of data:

	numberoffarmers
1	41

This query counts the number of farmers who have participated in training programs with ProgramID between 10 and 50.

2.

```
SELECT COUNT(FertilizerID) AS TotalFertilizers FROM
"G2SARM".Fertilizer;
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

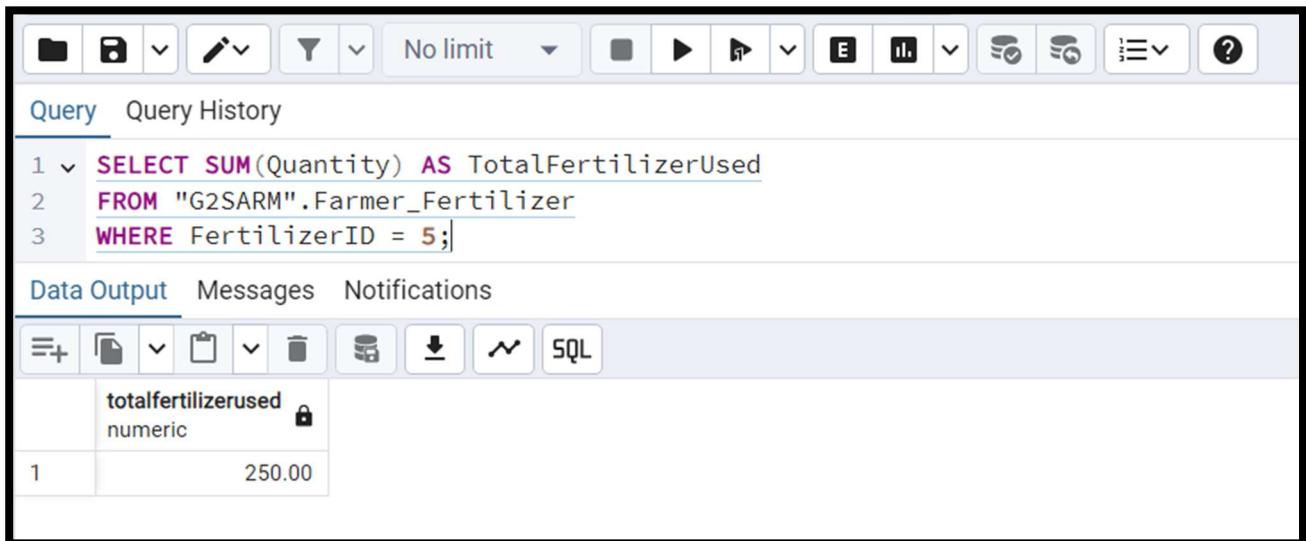
```
1 SELECT COUNT(FertilizerID) AS TotalFertilizers FROM "G2SARM".Fertilizer;
```

The results pane displays a single row of data:

	totalfertilizers
1	80

This query counts the Number of Different Fertilizers.

3. SELECT SUM(Quantity) AS TotalFertilizerUsed
 FROM "G2SARM".Farmer_Fertilizer
 WHERE FertilizerID = 5;



The screenshot shows a SQL query editor interface. The query window contains the following SQL code:

```

1 v  SELECT SUM(Quantity) AS TotalFertilizerUsed
2   FROM "G2SARM".Farmer_Fertilizer
3   WHERE FertilizerID = 5;

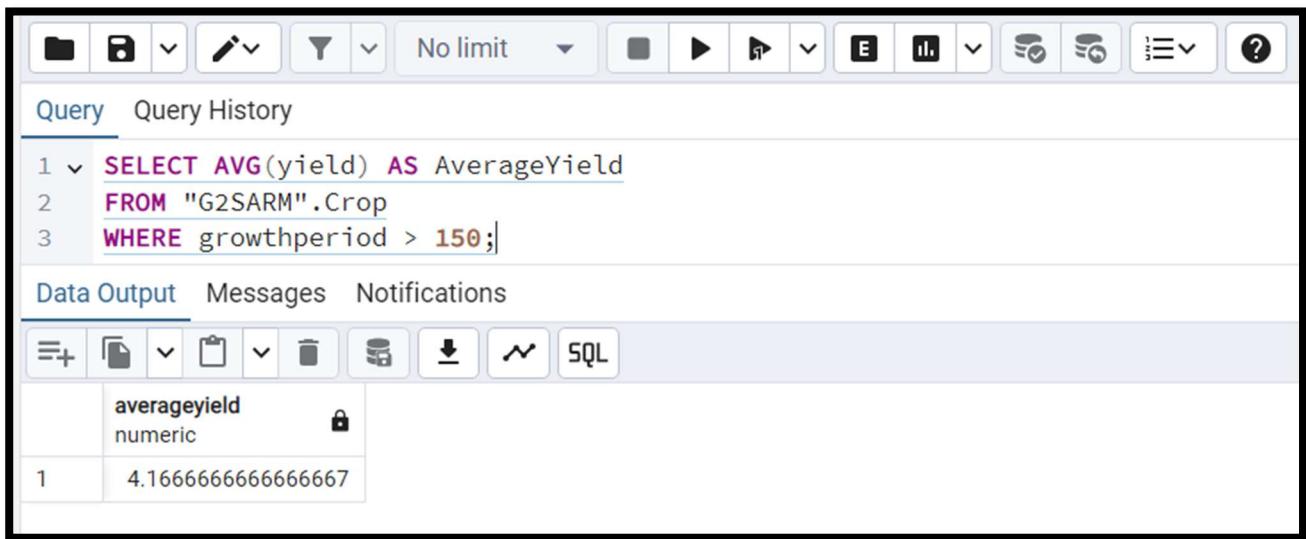
```

The results pane displays a single row of data:

	totalfertilizerused
1	250.00

This query calculates the total quantity of fertilizer (with FertilizerID 5) used by all farmers.

4. SELECT AVG(yield) AS AverageYield
 FROM "G2SARM".Crop
 WHERE growthperiod > 150;



The screenshot shows a SQL query editor interface. The query window contains the following SQL code:

```

1 v  SELECT AVG(yield) AS AverageYield
2   FROM "G2SARM".Crop
3   WHERE growthperiod > 150;

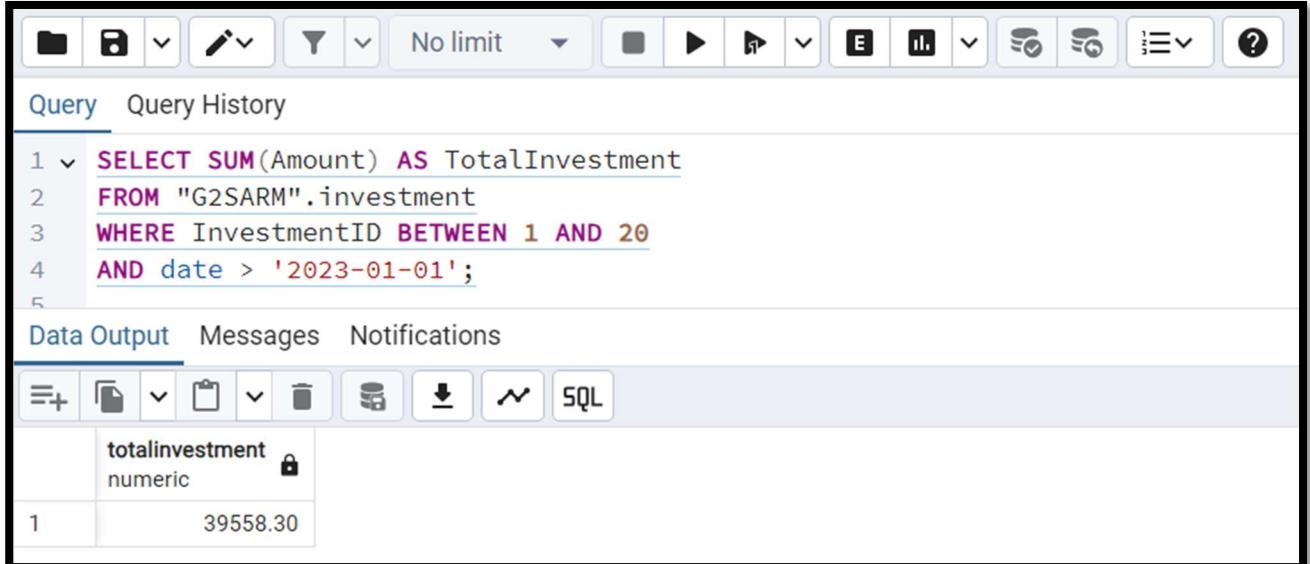
```

The results pane displays a single row of data:

	averageyield
1	4.1666666666666667

This query calculates the average yield for all crops which have growth period is greater than 150;

5. SELECT SUM(Amount) AS TotalInvestment
 FROM "G2SARM".investment
 WHERE InvestmentID BETWEEN 1 AND 20
 AND date > '2023-01-01';



The screenshot shows a MySQL Workbench interface. The query tab contains the following SQL code:

```

1 ✓ SELECT SUM(Amount) AS TotalInvestment
2   FROM "G2SARM".investment
3   WHERE InvestmentID BETWEEN 1 AND 20
4   AND date > '2023-01-01';
5

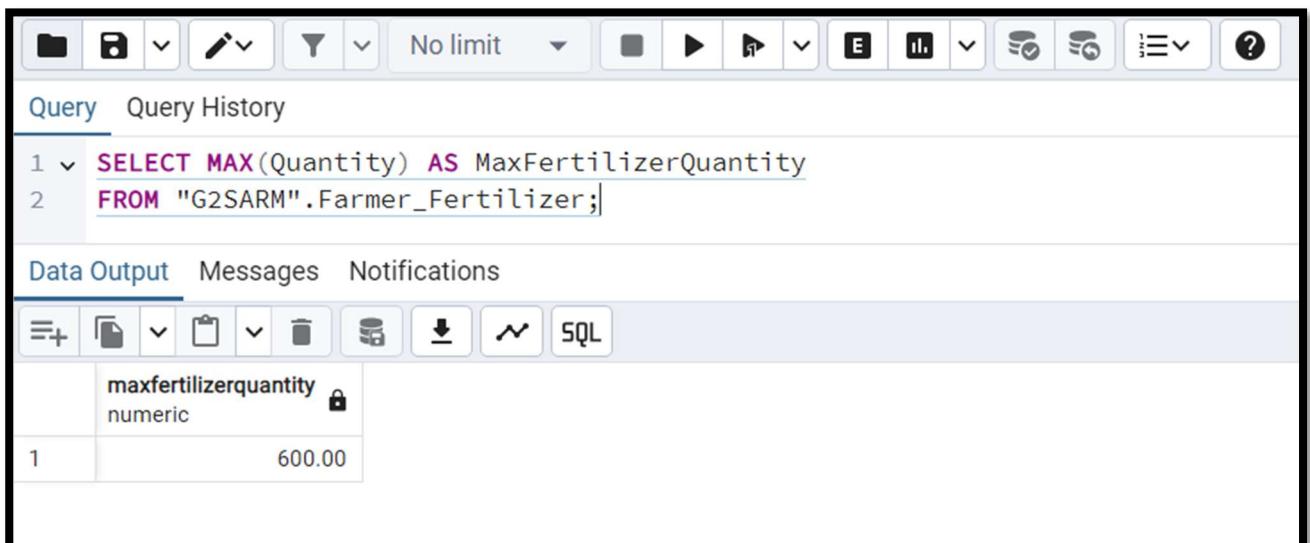
```

The results tab displays the output of the query:

	totalinvestment	numeric
1	39558.30	

This query calculates the total investment amount for investments with IDs between 1 and 20, made after January 1, 2023.

6. SELECT MAX(Quantity) AS MaxFertilizerQuantity
 FROM "G2SARM".Farmer_Fertilizer;



The screenshot shows a MySQL Workbench interface. The query tab contains the following SQL code:

```

1 ✓ SELECT MAX(Quantity) AS MaxFertilizerQuantity
2   FROM "G2SARM".Farmer_Fertilizer;

```

The results tab displays the output of the query:

	maxfertilizerquantity	numeric
1	600.00	

This query finds the highest fertilizer quantity used by any single farmer.

7. `SELECT COUNT(DISTINCT FarmerID) AS FarmersUsingTool
FROM "G2SARM".FarmingTool;`

The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons. Below the toolbar, the title bar says "Query Query History". The main area contains the following SQL code:

```
1 ✓ SELECT COUNT(DISTINCT FarmerID) AS FarmersUsingTool  
2 FROM "G2SARM".FarmingTool;  
3
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". Under "Data Output", there is a table with one row of data:

	farmersusingtool	bigint
1		10

This query counts the number of distinct farmers who are using any farming tool.

8. `SELECT COUNT(DISTINCT FarmerID) AS NumberOfFarmers
FROM "G2SARM".Farmer_Tech
WHERE TimeofUsage BETWEEN '2018-01-01' AND '2022-12-31';`

The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons. Below the toolbar, the title bar says "Query Query History". The main area contains the following SQL code:

```
1 ✓ SELECT COUNT(DISTINCT FarmerID) AS NumberOfFarmers  
2 FROM "G2SARM".Farmer_Tech  
3 WHERE TimeofUsage BETWEEN '2018-01-01' AND '2022-12-31';  
4
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". Under "Data Output", there is a table with one row of data:

	numberoffarmers	bigint
1		0

This query counts the number of unique farmers who have used technology between the years 2018 and 2022.

9. `SELECT FarmerID
FROM "G2SARM".Irrigation
WHERE InstallationDate > '2022-01-01'
ORDER BY InstallationDate DESC`

LIMIT 10;

The screenshot shows a SQL query editor interface. At the top, there's a toolbar with various icons. Below the toolbar, the title bar says "Query Query History". The main area contains the following SQL code:

```
1 ▾ SELECT FarmerID
2   FROM "G2SARM".Irrigation
3   WHERE InstallationDate > '2022-01-01'
4   ORDER BY InstallationDate DESC
5   LIMIT 10;
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications", with "Data Output" being the active tab. Under "Data Output", there's a table with one column labeled "farmerid" and a lock icon. The data is as follows:

	farmerid
1	40
2	68
3	39
4	67
5	66
6	38
7	90
8	65
9	37
10	89

This query retrieves the IDs of farmers who installed irrigation systems after January 1, 2022, ordered by installation date, limited to the top 10 results.

10. SELECT FarmerID, Name
 FROM "G2SARM".Farmer
 WHERE FarmerID NOT IN (
 SELECT DISTINCT FarmerID
 FROM "G2SARM".Farmer_Fertilizer
);

The screenshot shows a SQL query editor with the following details:

- Query History:** Shows the history of queries run.
- Query:** The current query is:


```

1 ✓ SELECT FarmerID, Name
2   FROM "G2SARM".Farmer
3 WHERE FarmerID NOT IN (
4     SELECT DISTINCT FarmerID
5     FROM "G2SARM".Farmer_Fertilizer
6 );

```
- Data Output:** The results of the query are displayed in a table:

	farmerid [PK] integer	name character varying (255)
1	77	Ishaan Mehta
2	78	Vishal Verma
3	79	Siddhant Sharma
4	80	Samir Gupta
5	81	Parth Rao
6	82	Ritesh Reddy
7	83	Shreyas Bhatia
8	84	Arnav Singh
9	85	Nitin Gupta
10	86	Vikrant Reddy
11	87	Suryansh Patel
12	88	Aakash Singh
13	89	Bharat Kumar
14	90	Yogesh Sharma
15	91	Hitesh Reddy
- Messages:** No messages are present.
- Notifications:** No notifications are present.
- Toolbar:** Includes icons for file operations (New, Open, Save, Print), search, filter, refresh, and various database management functions.
- Status Bar:** Shows "Total rows: 19 of 19" and "Query complete 00:00:00.081".

This query retrieves the IDs and name of farmers who have not used any fertilizer, by checking the absence of their FarmerID in the Farmer_Fertilizer table.

11. SELECT AVG(Efficiency) AS AverageEfficiency
 FROM "G2SARM".Irrigation
 WHERE TypeID = 1
 AND InstallationDate BETWEEN '2020-01-01' AND '2023-12-31';

The screenshot shows a SQL query editor interface. At the top, there are various toolbar icons. Below the toolbar, the title bar says "Query Query History". The main area contains the following SQL code:

```
1 ✓ SELECT AVG(Efficiency) AS AverageEfficiency
2 FROM "G2SARM".Irrigation
3 WHERE TypeID = 1
4 AND InstallationDate BETWEEN '2020-01-01' AND '2023-12-31';
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". Under "Data Output", there is a table with one row of data:

	averageefficiency
1	85.00000000000000

This query calculates the average efficiency of irrigation systems of type 1, installed between 2020 and 2023.

12. SELECT CropID
 FROM "G2SARM".Crop
 WHERE Yield > 5
 AND CropID < 50;

The screenshot shows a SQL query editor interface. At the top, there are various toolbar icons. Below the toolbar, the title bar says "Query Query History". The main area contains the following SQL code:

```
1 ✓ SELECT CropID
2 FROM "G2SARM".Crop
3 WHERE Yield > 5
4 AND CropID < 50;
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". Under "Data Output", there is a table with three rows of data:

	cropid
1	17
2	18
3	49

This query retrieves the CropID for crops with a yield greater than 5, specifically for CropID less than 50.

```

13.      SELECT ProgramID
        FROM "G2SARM".TrainingProgram
        WHERE duration = (
            SELECT MAX(duration)
            FROM "G2SARM".TrainingProgram
        );

```

The screenshot shows a SQL query editor with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Query History:** Shows the executed query steps.
- Data Output:** Shows the results of the query execution.
- Results Grid:** Displays the following data:

	programid	[PK] integer
1		3
2		10
3		35
4		47
5		58
6		73

This query retrieves the ProgramID of the training program with the longest duration.

```

14.      SELECT COUNT(DISTINCT FarmerID) AS NumberOfFarmers
        FROM "G2SARM".Farmer_Soil
        WHERE SoilID = 5;

```

The screenshot shows a SQL query editor with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Query History:** Shows the executed query steps.
- Data Output:** Shows the results of the query execution.
- Results Grid:** Displays the following data:

	numberoffarmers	bigint
1		1

This query counts the number of farmers using a specific soil type (SoilID 5).

```
15.    SELECT COUNT(DiseaseID) AS DiseaseCount  
        FROM "G2SARM".Crop_Disease  
        WHERE CropID = 2;
```

The screenshot shows a SQL query interface with the following details:

- Query History:** Shows the executed query:

```
2  FROM "G2SARM".Crop_Disease  
3  WHERE CropID = 2;  
4
```
- Data Output:** Displays the result of the query:

	diseasecount
1	3

This query counts the number of diseases affecting CropID 2.

```
16.    SELECT DISTINCT FarmerID  
        FROM "G2SARM".Farmer_Fertilizer  
        WHERE Quantity > 50;
```

The screenshot shows a SQL query interface with the following details:

- Query History:** Shows the executed query:

```
1  SELECT DISTINCT FarmerID  
2  FROM "G2SARM".Farmer_Fertilizer  
3  WHERE Quantity > 50;
```
- Data Output:** Displays the result of the query:

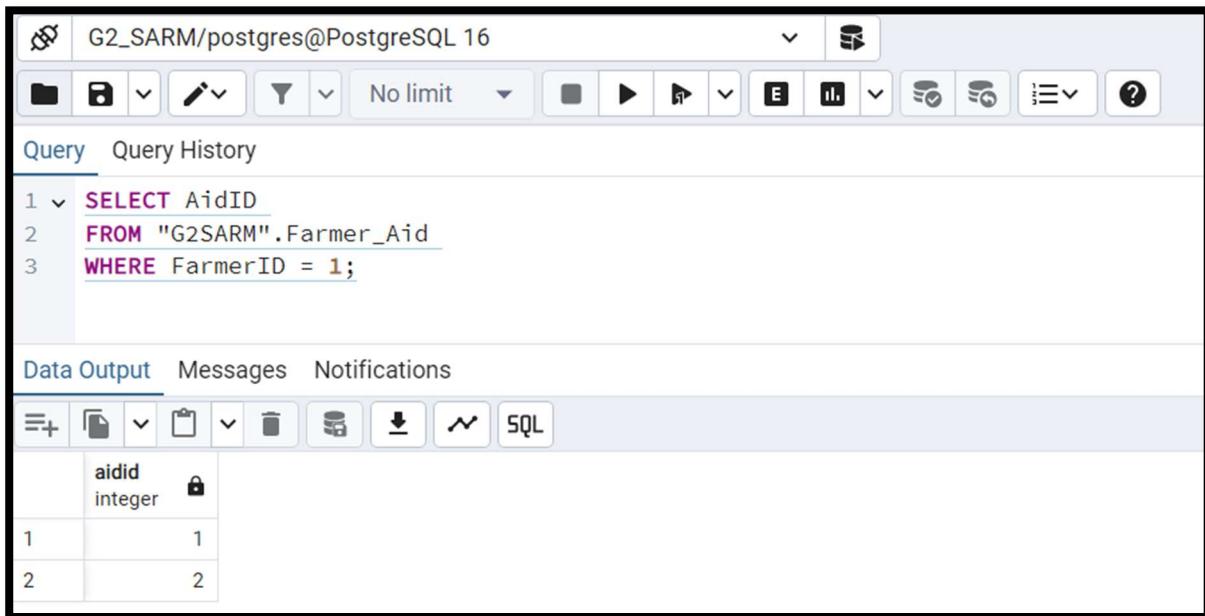
	farmerid
1	74
2	54
3	29
4	71
5	68

This query retrieves the list of farmers who have used more than 50 units of fertilizer.

```

17.    SELECT AidID
        FROM "G2SARM".Farmer_Aid
        WHERE FarmerID = 1;

```



The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes icons for connection, schema browser, table editor, search, and various execution and refresh buttons.
- Query History:** Shows the executed query: `SELECT AidID FROM "G2SARM".Farmer_Aid WHERE FarmerID = 1;`
- Data Output:** Shows the results of the query in a table format:

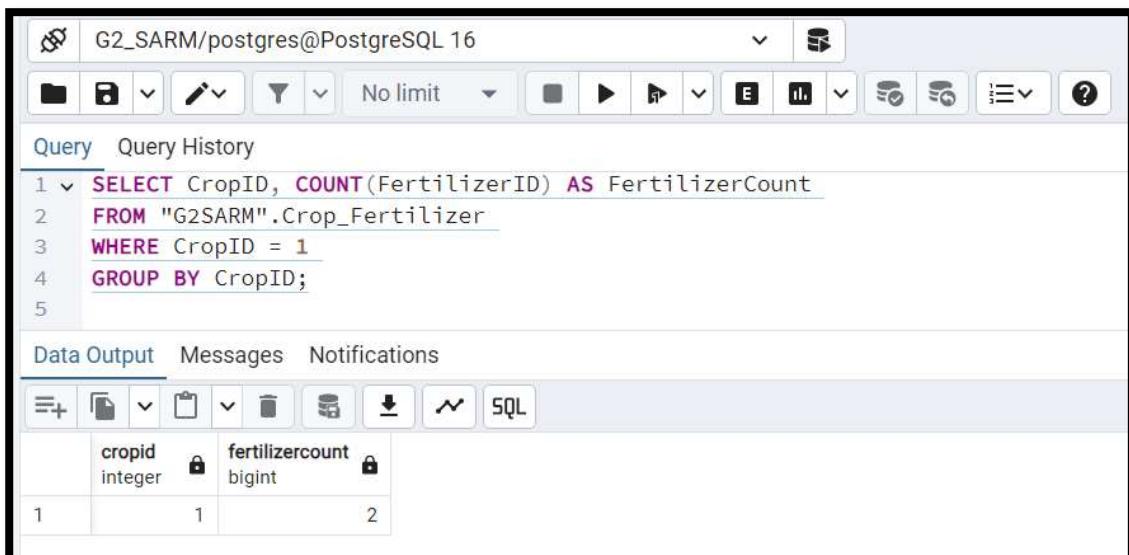
	aidid	integer
1		1
2		2

This query retrieves the AidID(s) associated with FarmerID 1 from the Farmer_Aid table.

```

18.    SELECT CropID, COUNT(FertilizerID) AS FertilizerCount
        FROM "G2SARM".Crop_Fertilizer
        WHERE CropID = 1
        GROUP BY CropID;

```



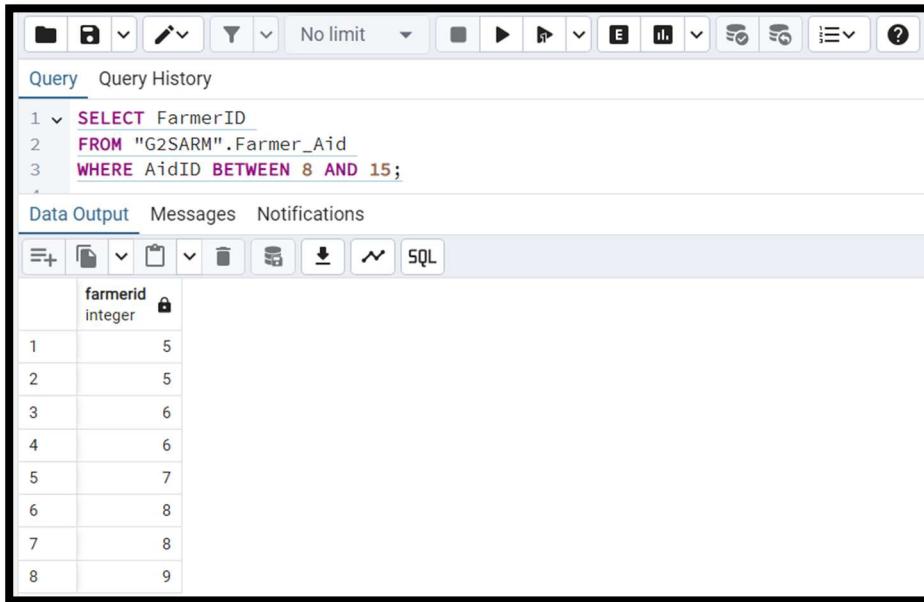
The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes icons for connection, schema browser, table editor, search, and various execution and refresh buttons.
- Query History:** Shows the executed query: `SELECT CropID, COUNT(FertilizerID) AS FertilizerCount FROM "G2SARM".Crop_Fertilizer WHERE CropID = 1 GROUP BY CropID;`
- Data Output:** Shows the results of the query in a table format:

	cropid	integer	fertilizercount	bigint
1		1		2

This query counts the number of fertilizers used for CropID 1 and groups the result by CropID.

19. SELECT FarmerID
 FROM "G2SARM".Farmer_Aid
 WHERE AidID BETWEEN 8 AND 15;



The screenshot shows a SQL query interface with the following details:

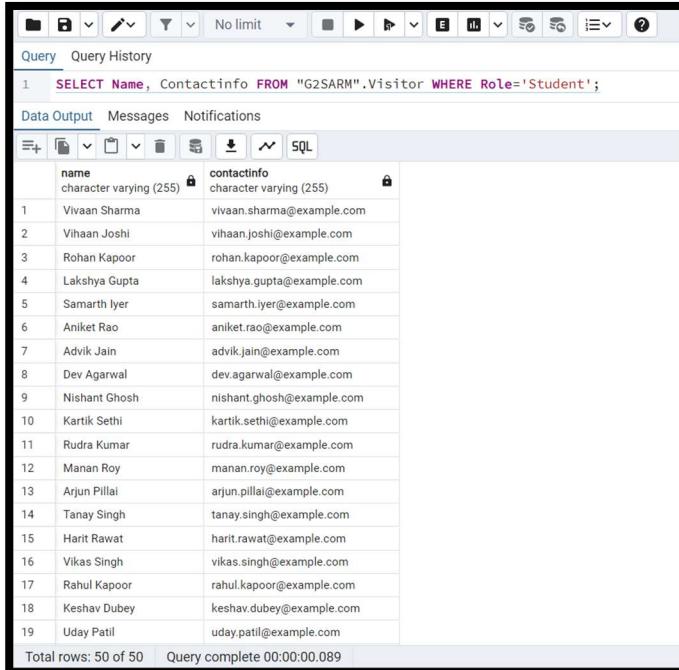
- Query History:** Shows the executed query:


```
1 SELECT FarmerID
2 FROM "G2SARM".Farmer_Aid
3 WHERE AidID BETWEEN 8 AND 15;
```
- Data Output:** Shows the results of the query in a table format.
- Table Structure:** The table has one column named "farmerid" of type integer.
- Table Data:**

	farmerid
1	5
2	5
3	6
4	6
5	7
6	8
7	8
8	9

This query retrieves the FarmerID(s) associated with AidID values between 8 and 15 from the Farmer_Aid table.

20. SELECT Name, Contactinfo FROM "G2SARM".Visitor WHERE Role='Student';



The screenshot shows a SQL query interface with the following details:

- Query History:** Shows the executed query:

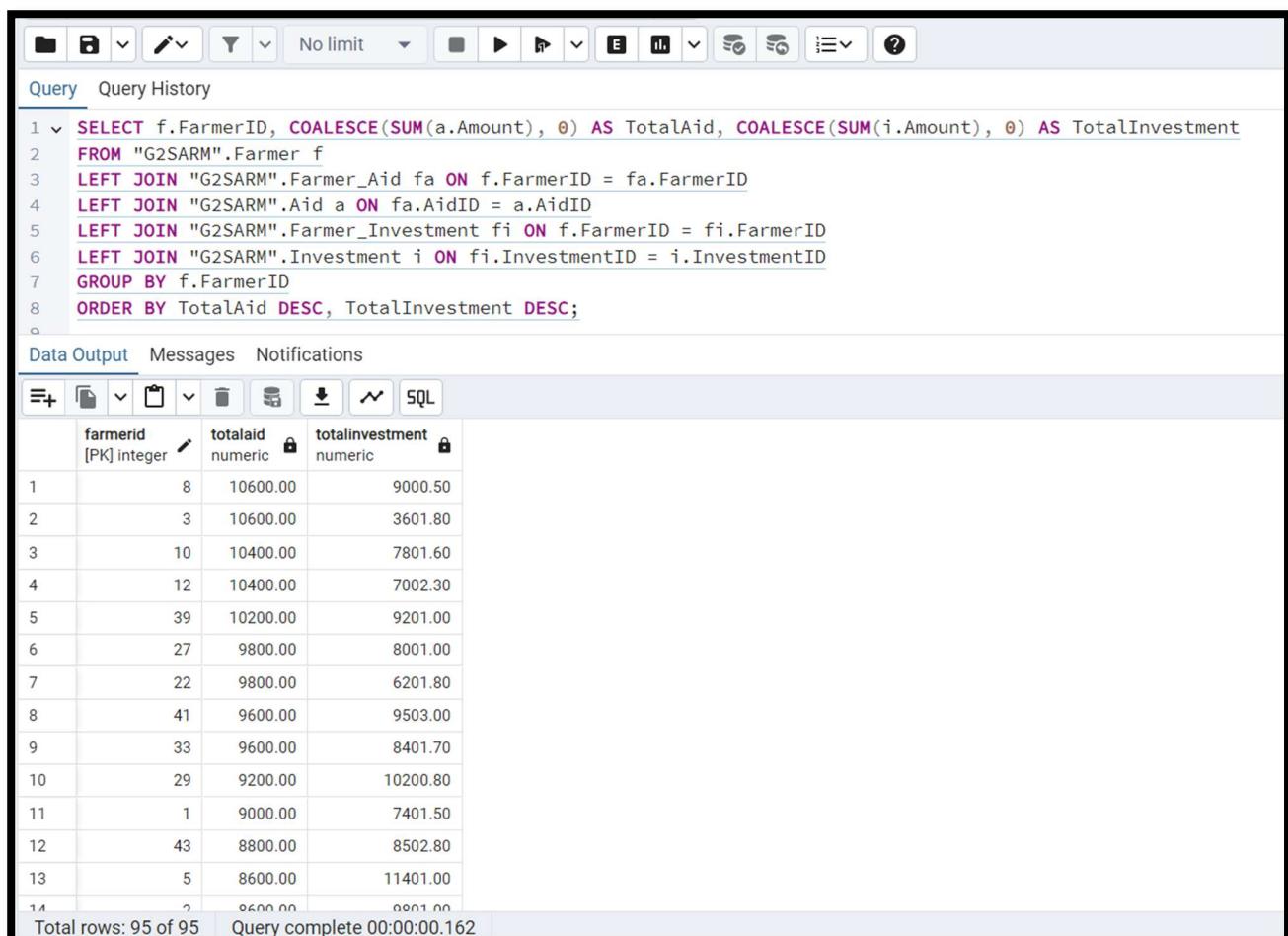

```
1 SELECT Name, Contactinfo FROM "G2SARM".Visitor WHERE Role='Student';
```
- Data Output:** Shows the results of the query in a table format.
- Table Structure:** The table has two columns: "name" and "contactinfo", both of type character varying (255).
- Table Data:**

	name	contactinfo
1	Vivaan Sharma	vivaan.sharma@example.com
2	Vihaan Joshi	vihaan.joshi@example.com
3	Rohan Kapoor	rohan.kapoor@example.com
4	Lakshya Gupta	lakshya.gupta@example.com
5	Samarth Iyer	samarth.iyer@example.com
6	Aniket Rao	aniket.rao@example.com
7	Advik Jain	advik.jain@example.com
8	Dev Agarwal	dev.agarwal@example.com
9	Nishant Ghosh	nishant.ghosh@example.com
10	Kartik Sethi	kartik.sethi@example.com
11	Rudra Kumar	rudra.kumar@example.com
12	Manan Roy	manan.roy@example.com
13	Arjun Pillai	arjun.pillai@example.com
14	Tanay Singh	tanay.singh@example.com
15	Harit Rawat	harit.rawat@example.com
16	Vikas Singh	vikas.singh@example.com
17	Rahul Kapoor	rahul.kapoor@example.com
18	Keshav Dubey	keshav.dubey@example.com
19	Uday Patil	uday.patil@example.com
- Message:** Total rows: 50 of 50 Query complete 00:00:00.089

This query retrieves the Name and Contactinfo of visitors who have the role of 'Student' from the Visitor table.

- **SQL Queries (20 Complex SQL Query)**

1. SELECT f.FarmerID, COALESCE(SUM(a.Amount), 0) AS TotalAid, COALESCE(SUM(i.Amount), 0) AS TotalInvestment
 FROM "G2SARM".Farmer f
 LEFT JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
 LEFT JOIN "G2SARM".Aid a ON fa.AidID = a.AidID
 LEFT JOIN "G2SARM".Farmer_Investment fi ON f.FarmerID = fi.FarmerID
 LEFT JOIN "G2SARM".Investment i ON fi.InvestmentID = i.InvestmentID
 GROUP BY f.FarmerID
 ORDER BY TotalAid DESC, TotalInvestment DESC;



The screenshot shows a database management interface with the following details:

- Query Tab:** Contains the executed SQL code.
- Data Output Tab:** Displays the results of the query in a table format.
- Table Structure:**

	farmerid [PK] integer	totalaid numeric	totalinvestment numeric
1	8	10600.00	9000.50
2	3	10600.00	3601.80
3	10	10400.00	7801.60
4	12	10400.00	7002.30
5	39	10200.00	9201.00
6	27	9800.00	8001.00
7	22	9800.00	6201.80
8	41	9600.00	9503.00
9	33	9600.00	8401.70
10	29	9200.00	10200.80
11	1	9000.00	7401.50
12	43	8800.00	8502.80
13	5	8600.00	11401.00
14	?	8600.00	8801.00
- Total Rows:** 95 of 95
- Query Status:** Query complete 00:00:00.162

This query calculates the total aid and investment amounts for each farmer, sorting the results by aid and investment amounts in descending order.

```

2. SELECT s.SoilId, AVG(c.Yield) AS AverageYield
FROM "G2SARM".Soil s
JOIN "G2SARM".Crop_Soil cs ON s.SoilID = cs.SoilID
JOIN "G2SARM".Crop c ON cs.CropID = c.CropID
GROUP BY s.SoilId
ORDER BY AverageYield DESC;

```

The screenshot shows a SQL query being run in a database client. The query calculates the average yield of crops grown on different soil types, ordered by average yield in descending order.

Query History:

```

1 v  SELECT s.SoilId, AVG(c.Yield) AS AverageYield
2   FROM "G2SARM".Soil s
3   JOIN "G2SARM".Crop_Soil cs ON s.SoilID = cs.SoilID
4   JOIN "G2SARM".Crop c ON cs.CropID = c.CropID
5   GROUP BY s.SoilId
6   ORDER BY AverageYield DESC;

```

Data Output:

	soilid [PK] integer	averageyield numeric
1	27	8.000000000000000
2	26	8.000000000000000
3	28	7.500000000000000
4	69	6.000000000000000
5	67	5.000000000000000
6	53	5.000000000000000
7	25	5.000000000000000
8	31	4.500000000000000
9	68	4.500000000000000
10	56	4.200000000000000
11	57	4.200000000000000
12	9	4.200000000000000
13	8	4.200000000000000
14	22	4.000000000000000
15	3	4.000000000000000

Total rows: 90 of 90 Query complete 00:00:00.085

This query calculates the average yield of crops grown on different soil types.

```

3. SELECT DISTINCT f.FarmerID, f.Name
   FROM "G2SARM".Farmer f
JOIN "G2SARM".Farmer_SustainablePractice fp ON f.FarmerID =
fp.FarmerID
JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
WHERE fp.PracticeID IN (SELECT PracticeID FROM
"G2SARM".SustainablePractice WHERE Description LIKE '%organic%');

```

The screenshot shows a SQL query editor with the following details:

- Toolbar:** Includes standard database navigation icons like file, save, search, and refresh.
- Query Tab:** Displays the SQL query:

```

1  SELECT DISTINCT f.FarmerID, f.Name
2  FROM "G2SARM".Farmer f
3  JOIN "G2SARM".Farmer_SustainablePractice fp ON f.FarmerID = fp.FarmerID
4  JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
5  WHERE fp.PracticeID IN (SELECT PracticeID FROM "G2SARM".SustainablePractice WHERE Description LIKE '%organic%');

```

- Data Output Tab:** Shows the results of the query as a table:

	farmerid	name
1	11	Aarav Singh
2	12	Vikram Sharma
3	53	Anish Verma

This query finds farmers who practice sustainable methods (such as "organic" practices) and have received aid.

```

4. SELECT f.FarmerID, f.Name, c.Name
   FROM "G2SARM".Farmer_Crop fc
  JOIN "G2SARM".Farmer f ON fc.FarmerID = f.FarmerID
  JOIN "G2SARM".Crop c ON fc.CropID = c.CropID
  JOIN "G2SARM".Crop_Weather cw ON c.CropID = cw.CropID
  JOIN "G2SARM".Weather w ON cw.WeatherID = w.WeatherID
 WHERE w.conditions = 'Clear';

```

The screenshot shows a database interface with a toolbar at the top, a 'Query History' tab, and a main area for running queries. The query is displayed in the main area:

```

1 ▾ SELECT f.FarmerID, f.Name, c.Name
2   FROM "G2SARM".Farmer_Crop fc
3   JOIN "G2SARM".Farmer f ON fc.FarmerID = f.FarmerID
4   JOIN "G2SARM".Crop c ON fc.CropID = c.CropID
5   JOIN "G2SARM".Crop_Weather cw ON c.CropID = cw.CropID
6   JOIN "G2SARM".Weather w ON cw.WeatherID = w.WeatherID
7   WHERE w.conditions = 'Clear';

```

The results are shown in a table titled 'Data Output' with columns: farmerid, name, and name. The data consists of 22 rows:

	farmerid	name	name
1	1	Manav Singh	Wheat
2	1	Manav Singh	Barley
3	2	Veer Sharma	Oats
4	2	Veer Sharma	Corn
5	3	Kumar Gupta	Chickpea
6	3	Kumar Gupta	Chickpea
7	5	Karma Verma	Tomato
8	8	Raj Rao	Banana
9	8	Raj Rao	Orange
10	9	Shiraz Desai	Strawberry
11	1	Manav Singh	Parsley
12	2	Veer Sharma	Sunflower
13	10	Kaushal Reddy	Buckwheat
14	2	Veer Sharma	Black Pepper
15	6	Donal Cook	Walnut

Total rows: 22 of 22 | Query complete 00:00:00.065

This query retrieves information about farmers who grow crops suited to clear weather conditions.

5. SELECT CropID, COUNT(DiseaseID) AS DiseaseCount
FROM "G2SARM".Crop_Disease
GROUP BY CropID
HAVING COUNT(DiseaseID) > 1
ORDER BY DiseaseCount DESC;

The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons for file operations, search, and navigation. Below the toolbar, the title bar says "Query Query History". The main area contains the SQL code for the fifth query. The code is as follows:

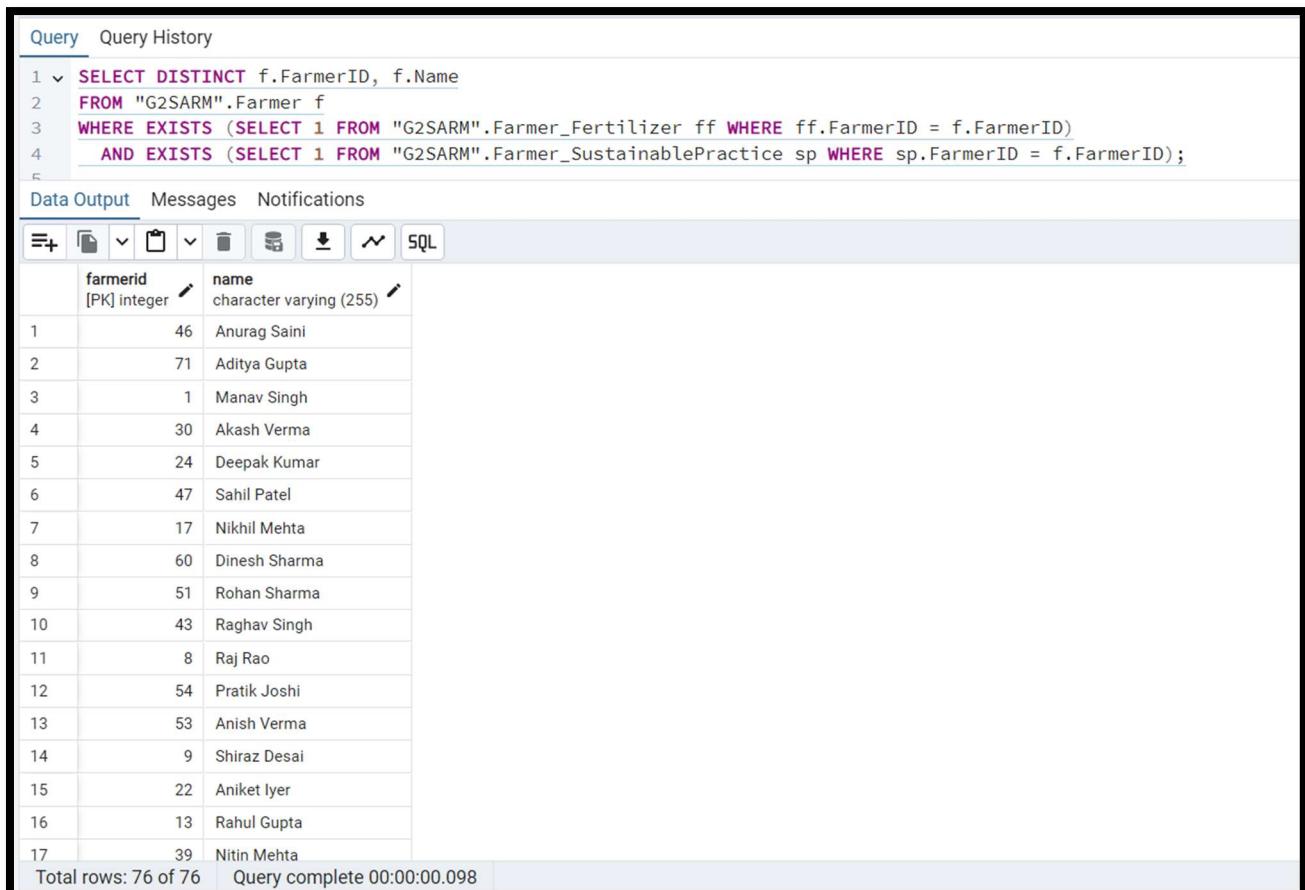
```
1 SELECT CropID, COUNT(DiseaseID) AS DiseaseCount
2 FROM "G2SARM".Crop_Disease
3 GROUP BY CropID
4 HAVING COUNT(DiseaseID) > 1
5 ORDER BY DiseaseCount DESC;
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is selected, showing a table with two columns: "cropid" and "diseasecount". The data is as follows:

	cropid	diseasecount
1	integer	3
2		1
3		2
4		9
5		7
6		4
7		5
8		6
9		11

This query identifies crops affected by more than one disease, sorted by the number of diseases.

6. SELECT DISTINCT f.FarmerID, f.Name
 FROM "G2SARM".Farmer f
 WHERE EXISTS (SELECT 1 FROM "G2SARM".Farmer_Fertilizer ff
 WHERE ff.FarmerID = f.FarmerID)
 AND EXISTS (SELECT 1 FROM
 "G2SARM".Farmer_SustainablePractice sp WHERE sp.FarmerID =
 f.FarmerID);



The screenshot shows a SQL query execution interface with the following details:

- Query Tab:** Contains the SQL code for the query.
- Data Output Tab:** Active tab, showing the results of the query.
- Table Results:**

	farmerid [PK] integer	name character varying (255)
1	46	Anurag Saini
2	71	Aditya Gupta
3	1	Manav Singh
4	30	Akash Verma
5	24	Deepak Kumar
6	47	Sahil Patel
7	17	Nikhil Mehta
8	60	Dinesh Sharma
9	51	Rohan Sharma
10	43	Raghav Singh
11	8	Raj Rao
12	54	Pratik Joshi
13	53	Anish Verma
14	9	Shiraz Desai
15	22	Aniket Iyer
16	13	Rahul Gupta
17	39	Nitin Mehta
- Statistics:**
 - Total rows: 76 of 76
 - Query complete 00:00:00.098

This query finds farmers who are using both fertilizers and sustainable practices.

7. SELECT FarmerID, COUNT(DISTINCT CropID) AS CropVarietyCount
FROM "G2SARM".Farmer_Crop
GROUP BY FarmerID
ORDER BY CropVarietyCount DESC
LIMIT 5;

The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons for file operations, search, and navigation. Below the toolbar, the title bar says "Query Query History". The main area contains the SQL code for the query. The code is as follows:

```
1 v SELECT FarmerID, COUNT(DISTINCT CropID) AS CropVarietyCount
2 FROM "G2SARM".Farmer_Crop
3 GROUP BY FarmerID
4 ORDER BY CropVarietyCount DESC
5 LIMIT 5;
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is selected, showing a table with the results of the query. The table has two columns: "farmerid" and "cropvarietycount". The data is as follows:

	farmerid	cropvarietycount
1	1	10
2	2	10
3	3	9
4	4	9
5	5	9

This query identifies the top 5 farmers who grow the most diverse range of crops.

```

8. SELECT c.Name, AVG(ff.Quantity) AS AvgFertilizerQuantity
FROM "G2SARM".Crop c
JOIN "G2SARM".Farmer_Crop fc ON c.CropID = fc.CropID
JOIN "G2SARM".Farmer_Fertilizer ff ON fc.FarmerID = ff.FarmerID
GROUP BY c.Name
ORDER BY AvgFertilizerQuantity DESC;

```

The screenshot shows a SQL query editor with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste), search, filter, and various database management functions.
- Query History:** Shows the executed query.
- Data Output:** Displays the results of the query in a table.
- Table Headers:** The table has two columns: 'name' (character varying (255)) and 'avgfertilizerquantity' (numeric).
- Table Data:** The table contains 16 rows of data, each representing a crop and its average fertilizer quantity. The data is as follows:

	name	avgfertilizerquantity
	character varying (255)	numeric
1	Cotton	500.0000000000000000
2	Pea	500.0000000000000000
3	Ginger	500.0000000000000000
4	Safflower	500.0000000000000000
5	Lotus	500.0000000000000000
6	Celeriac	500.0000000000000000
7	Black Bean	500.0000000000000000
8	Vanilla	500.0000000000000000
9	Taro	500.0000000000000000
10	Nettle	325.0000000000000000
11	Duckweed	325.0000000000000000
12	Fennel	325.0000000000000000
13	Almond	325.0000000000000000
14	Hemp	325.0000000000000000
15	Palm Oil	325.0000000000000000
16	Mango	325.0000000000000000

- Total rows:** 89 of 89
- Query complete:** 00:00:00.091

This query calculates the average fertilizer quantity used for each crop type.

```
9. SELECT f.FarmerID, f.Name  
FROM "G2SARM".Farmer_Aid fa  
JOIN "G2SARM".Farmer f ON fa.FarmerID = f.FarmerID  
JOIN "G2SARM".Farmer_Fertilizer ff ON f.FarmerID = ff.FarmerID  
WHERE ff.FertilizerID = 2;
```

The screenshot shows a PostgreSQL query editor interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, the title bar says "Query History". The main area contains the SQL query code:

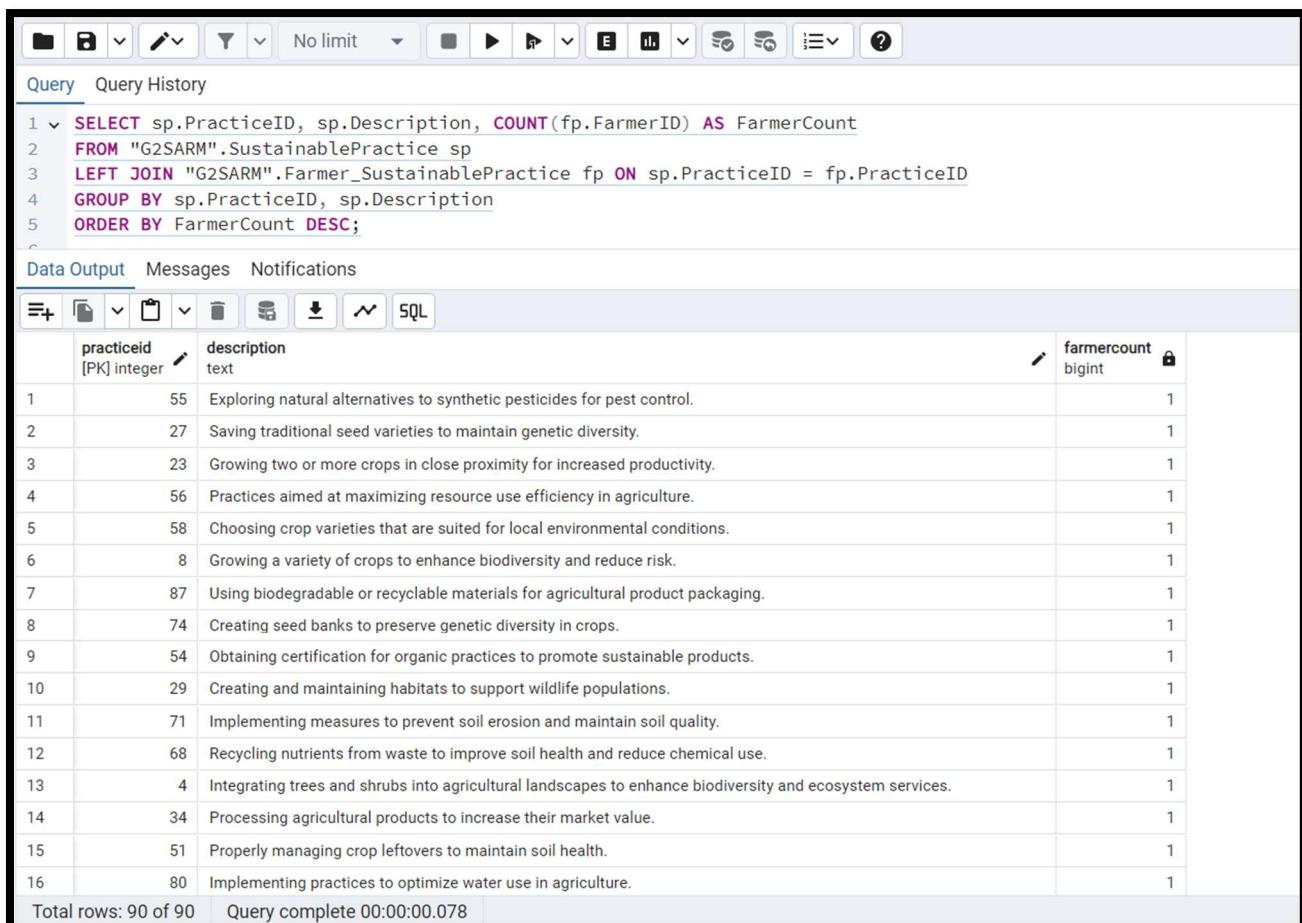
```
1 ✓ SELECT f.FarmerID, f.Name  
2 FROM "G2SARM".Farmer_Aid fa  
3 JOIN "G2SARM".Farmer f ON fa.FarmerID = f.FarmerID  
4 JOIN "G2SARM".Farmer_Fertilizer ff ON f.FarmerID = ff.FarmerID  
5 WHERE ff.FertilizerID = 2;
```

Below the code, there are tabs for "Data Output", "Messages", and "Notifications". Under "Data Output", there's a table preview:

	farmerid [PK] integer	name character varying (255)
1	1	Manav Singh
2	1	Manav Singh

This query retrieves farmers who have received aid and use a specific fertilizer with FertilizerID = 2.

10. SELECT sp.PracticeID, sp.Description, COUNT(fp.FarmerID) AS FarmerCount
 FROM "G2SARM".SustainablePractice sp
 LEFT JOIN "G2SARM".Farmer_SustainablePractice fp ON
 sp.PracticeID = fp.PracticeID
 GROUP BY sp.PracticeID, sp.Description
 ORDER BY FarmerCount DESC;



The screenshot shows a SQL query interface with the following details:

- Query History:** The query is listed under the "Query History" tab.
- SQL Query:**

```

1  SELECT sp.PracticeID, sp.Description, COUNT(fp.FarmerID) AS FarmerCount
2  FROM "G2SARM".SustainablePractice sp
3  LEFT JOIN "G2SARM".Farmer_SustainablePractice fp ON sp.PracticeID = fp.PracticeID
4  GROUP BY sp.PracticeID, sp.Description
5  ORDER BY FarmerCount DESC;

```
- Data Output:** The results are displayed in a table with three columns: practiceid, description, and farmercount.
- Table Data:**

	practiceid [PK] integer	description text	farmercount bigint
1	55	Exploring natural alternatives to synthetic pesticides for pest control.	1
2	27	Saving traditional seed varieties to maintain genetic diversity.	1
3	23	Growing two or more crops in close proximity for increased productivity.	1
4	56	Practices aimed at maximizing resource use efficiency in agriculture.	1
5	58	Choosing crop varieties that are suited for local environmental conditions.	1
6	8	Growing a variety of crops to enhance biodiversity and reduce risk.	1
7	87	Using biodegradable or recyclable materials for agricultural product packaging.	1
8	74	Creating seed banks to preserve genetic diversity in crops.	1
9	54	Obtaining certification for organic practices to promote sustainable products.	1
10	29	Creating and maintaining habitats to support wildlife populations.	1
11	71	Implementing measures to prevent soil erosion and maintain soil quality.	1
12	68	Recycling nutrients from waste to improve soil health and reduce chemical use.	1
13	4	Integrating trees and shrubs into agricultural landscapes to enhance biodiversity and ecosystem services.	1
14	34	Processing agricultural products to increase their market value.	1
15	51	Properly managing crop leftovers to maintain soil health.	1
16	80	Implementing practices to optimize water use in agriculture.	1
- Total rows:** 90 of 90
- Query complete:** 00:00:00.078

This query calculates the number of farmers who are practicing each sustainable method.

```

11.   SELECT DISTINCT f.FarmerID, f.Name
      FROM "G2SARM".Farmer f
      JOIN "G2SARM".Farmer_Crop fc ON f.FarmerID = fc.FarmerID
      JOIN "G2SARM".Crop_Weather cw ON fc.CropID = cw.CropID
      JOIN "G2SARM".Weather w ON cw.WeatherID = w.WeatherID
      WHERE w.conditions IN ('Sunny', 'Rainy')
      GROUP BY f.FarmerID
      HAVING COUNT(DISTINCT w.WeatherId) > 2;

```

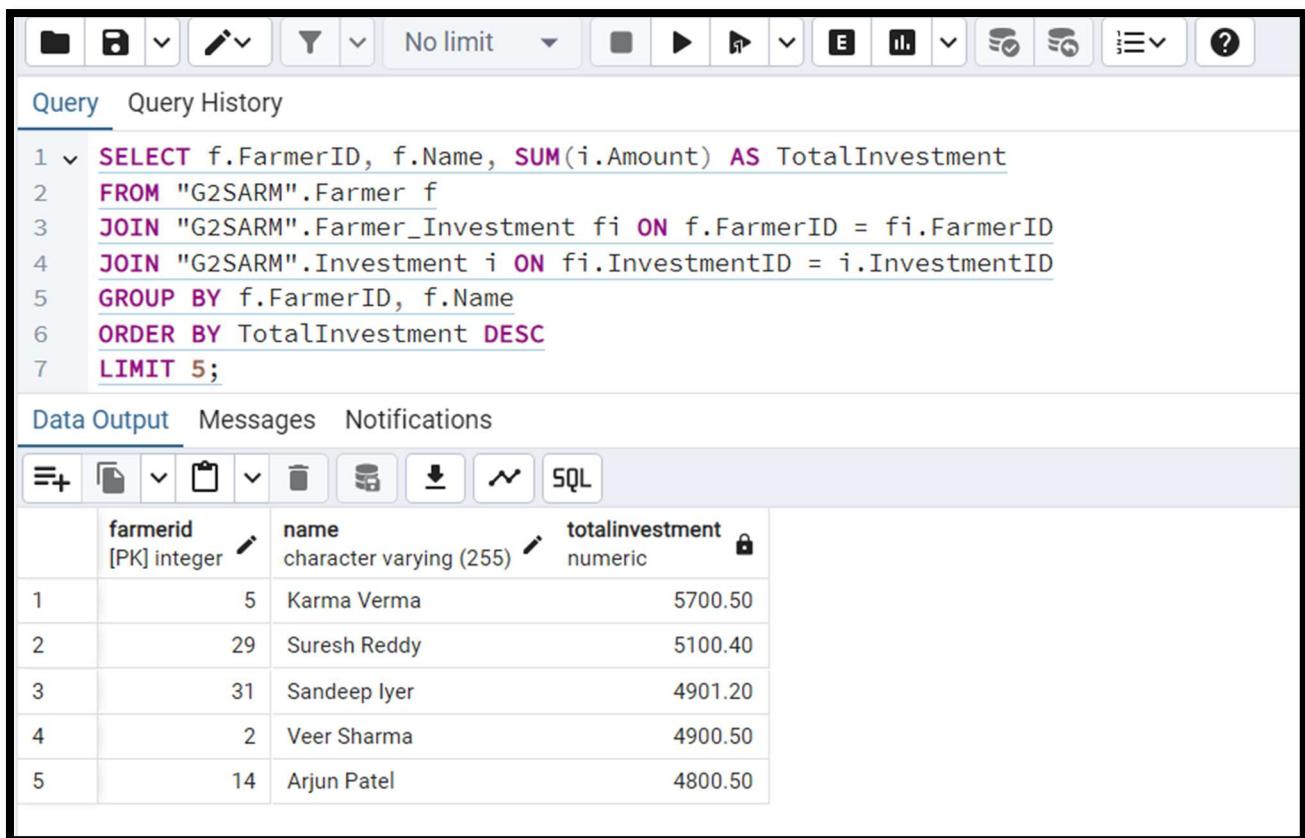
The screenshot shows a PostgreSQL client interface with the following details:

- Query Bar:** Contains various icons for file operations, search, and navigation.
- Query History:** Shows the executed SQL query.
- SQL Editor:** Displays the numbered query code.
- Data Output:** Shows the results of the query as a table.
- Table Data:** The result table has columns `farmerid` and `name`. The data is:

	farmerid	name
1	5	Karma Verma
2	1	Manav Singh
3	10	Kaushal Reddy
4	4	Aamir Patel
5	9	Shiraz Desai
6	6	Ronak Joshi
7	3	Kumar Gupta
8	2	Veer Sharma

This query retrieves farmers growing crops suited to both sunny and rainy rainfall conditions.

12. SELECT f.FarmerID, f.Name, SUM(i.Amount) AS TotalInvestment
 FROM "G2SARM".Farmer f
 JOIN "G2SARM".Farmer_Investment fi ON f.FarmerID = fi.FarmerID
 JOIN "G2SARM".Investment i ON fi.InvestmentID = i.InvestmentID
 GROUP BY f.FarmerID, f.Name
 ORDER BY TotalInvestment DESC
 LIMIT 5;



The screenshot shows a PostgreSQL SQL editor interface. The top bar contains various icons for file operations, search, and navigation. Below the bar, there are tabs for 'Query' (which is selected), 'Query History', and other options. The main area displays the SQL query numbered from 1 to 7. Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing a table with five rows of data. The table has three columns: 'farmerid' (with a primary key icon), 'name' (with a varying character type icon), and 'totalinvestment' (with a numeric type icon). The data is as follows:

	farmerid [PK] integer	name character varying (255)	totalinvestment numeric
1	5	Karma Verma	5700.50
2	29	Suresh Reddy	5100.40
3	31	Sandeep Iyer	4901.20
4	2	Veer Sharma	4900.50
5	14	Arjun Patel	4800.50

This query finds the top 5 farmers with the highest total investment.

```

13.    SELECT f.FarmerID, f.Name
        FROM "G2SARM".Farmer f
        WHERE NOT EXISTS (
            SELECT 1
            FROM "G2SARM".Farmer_SustainablePractice fp
            JOIN "G2SARM".SustainablePractice sp ON fp.PracticeID =
                sp.PracticeID
            WHERE fp.FarmerID = f.FarmerID AND sp.Description NOT LIKE
                '%Organic%'
        );

```

The screenshot shows a database management interface with the following details:

- Toolbar:** Includes various icons for file operations, search, and navigation.
- Query History Tab:** Shows the executed SQL query.
- SQL Query:**

```

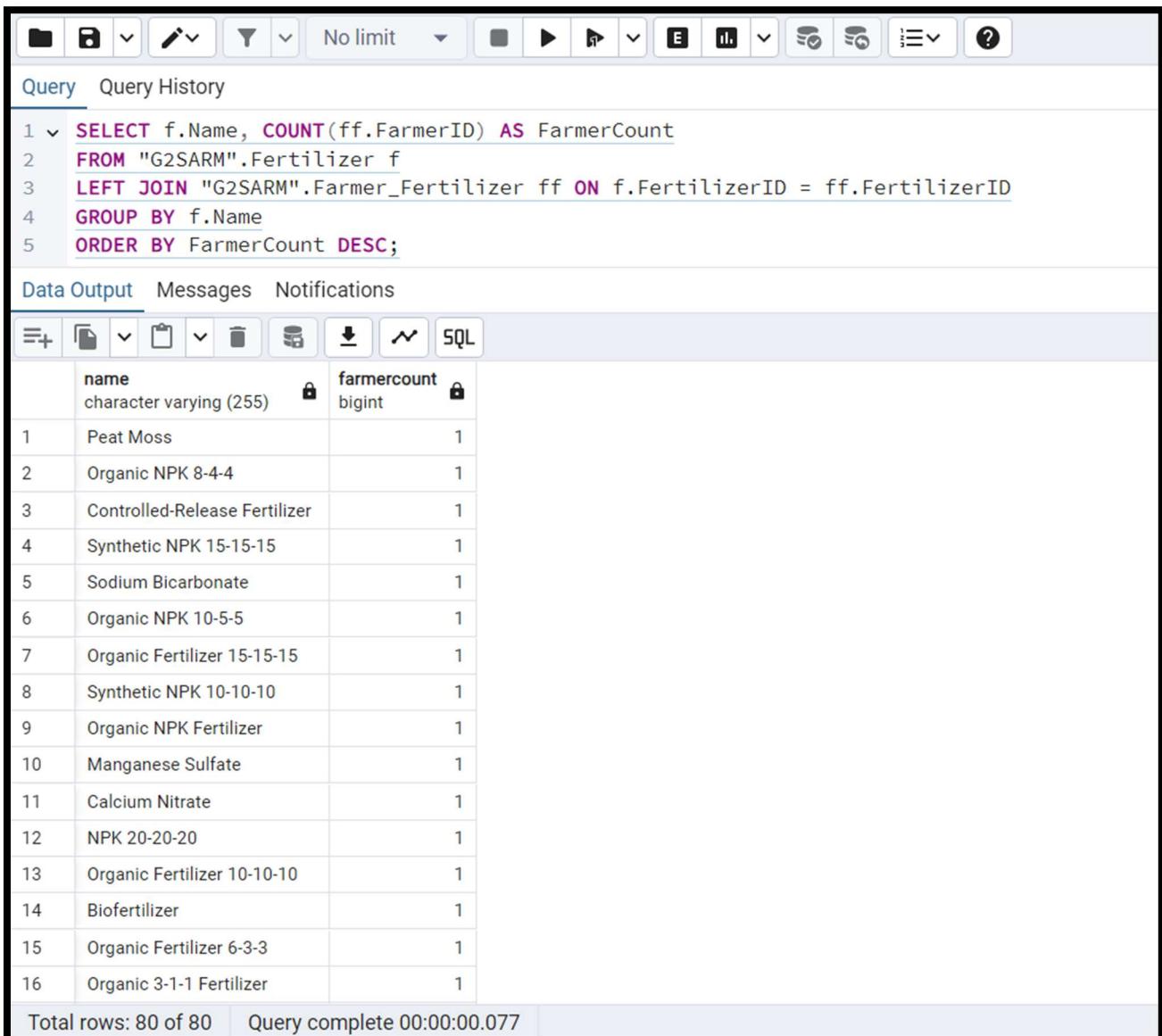
1  SELECT f.FarmerID, f.Name
2  FROM "G2SARM".Farmer f
3  WHERE NOT EXISTS (
4      SELECT 1
5      FROM "G2SARM".Farmer_SustainablePractice fp
6      JOIN "G2SARM".SustainablePractice sp ON fp.PracticeID = sp.PracticeID
7      WHERE fp.FarmerID = f.FarmerID AND sp.Description NOT LIKE '%Organic%'

```
- Data Output Tab:** Displays a table with the results of the query.
- Table Results:**

	farmerid [PK] integer	name character varying (255)
1	90	Yogesh Sharma
2	91	Hitesh Reddy
3	92	Vikram Verma
4	93	Ajay Mehta
5	94	Rajan Singh
6	95	Vinod Patel

This query lists farmers who exclusively use organic sustainable practices.

14. SELECT f.Name, COUNT(ff.FarmerID) AS FarmerCount
 FROM "G2SARM".Fertilizer f
 LEFT JOIN "G2SARM".Farmer_Fertilizer ff ON f.FertilizerID =
 ff.FertilizerID
 GROUP BY f.Name
 ORDER BY FarmerCount DESC;



The screenshot shows a database query interface with the following details:

- Query History:** The tab is labeled "Query History".
- SQL Query:**

```

1 v SELECT f.Name, COUNT(ff.FarmerID) AS FarmerCount
2   FROM "G2SARM".Fertilizer f
3   LEFT JOIN "G2SARM".Farmer_Fertilizer ff ON f.FertilizerID = ff.FertilizerID
4   GROUP BY f.Name
5   ORDER BY FarmerCount DESC;
    
```
- Data Output:** The tab is labeled "Data Output". The results are displayed in a table.
- Table Headers:** The table has two columns: "name" (character varying (255)) and "farmercount" (bigint).
- Table Data:**

	name	farmercount
1	Peat Moss	1
2	Organic NPK 8-4-4	1
3	Controlled-Release Fertilizer	1
4	Synthetic NPK 15-15-15	1
5	Sodium Bicarbonate	1
6	Organic NPK 10-5-5	1
7	Organic Fertilizer 15-15-15	1
8	Synthetic NPK 10-10-10	1
9	Organic NPK Fertilizer	1
10	Manganese Sulfate	1
11	Calcium Nitrate	1
12	NPK 20-20-20	1
13	Organic Fertilizer 10-10-10	1
14	Biofertilizer	1
15	Organic Fertilizer 6-3-3	1
16	Organic 3-1-1 Fertilizer	1
- Total Rows:** Total rows: 80 of 80
- Query Complete:** Query complete 00:00:00.077

This query shows how many farmers are using each type of fertilizer, ordered by the number of users per fertilizer.

```

15.    SELECT AVG(a.Amount) AS AvgAid
        FROM "G2SARM".Farmer f
        JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
        JOIN "G2SARM".Aid a ON fa.AidID = a.AidID
        WHERE EXISTS (
            SELECT 1
            FROM "G2SARM".Farmer_SustainablePractice fp
            WHERE fp.FarmerID = f.FarmerID
        );

```

The screenshot shows a SQL query editor window. At the top, there's a toolbar with various icons. Below it, the 'Query History' tab is selected, displaying the SQL code. The code itself is as follows:

```

1  SELECT AVG(a.Amount) AS AvgAid
2  FROM "G2SARM".Farmer f
3  JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
4  JOIN "G2SARM".Aid a ON fa.AidID = a.AidID
5  WHERE EXISTS (
6      SELECT 1
7      FROM "G2SARM".Farmer_SustainablePractice fp
8      WHERE fp.FarmerID = f.FarmerID
9  );

```

Below the code, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Under 'Data Output', a table is shown with one row of data:

	avgaid
1	2271.6666666666666667

This query calculates the average aid amount received by farmers who are engaged in sustainable practices.

```

16.   SELECT DISTINCT f.FarmerID, f.name, d.name
      FROM "G2SARM".Farmer f
      JOIN "G2SARM".Farmer_Crop fc ON f.FarmerID = fc.FarmerID
      JOIN "G2SARM".Crop_Disease cd ON fc.CropID = cd.CropID
      JOIN "G2SARM".cropdisease d ON cd.DiseaseID = d.DiseaseID
      WHERE d.name = 'Blight';

```

The screenshot shows a database interface with the following details:

- Query History:** The query is listed under the "Query" tab.
- SQL Query:**

```

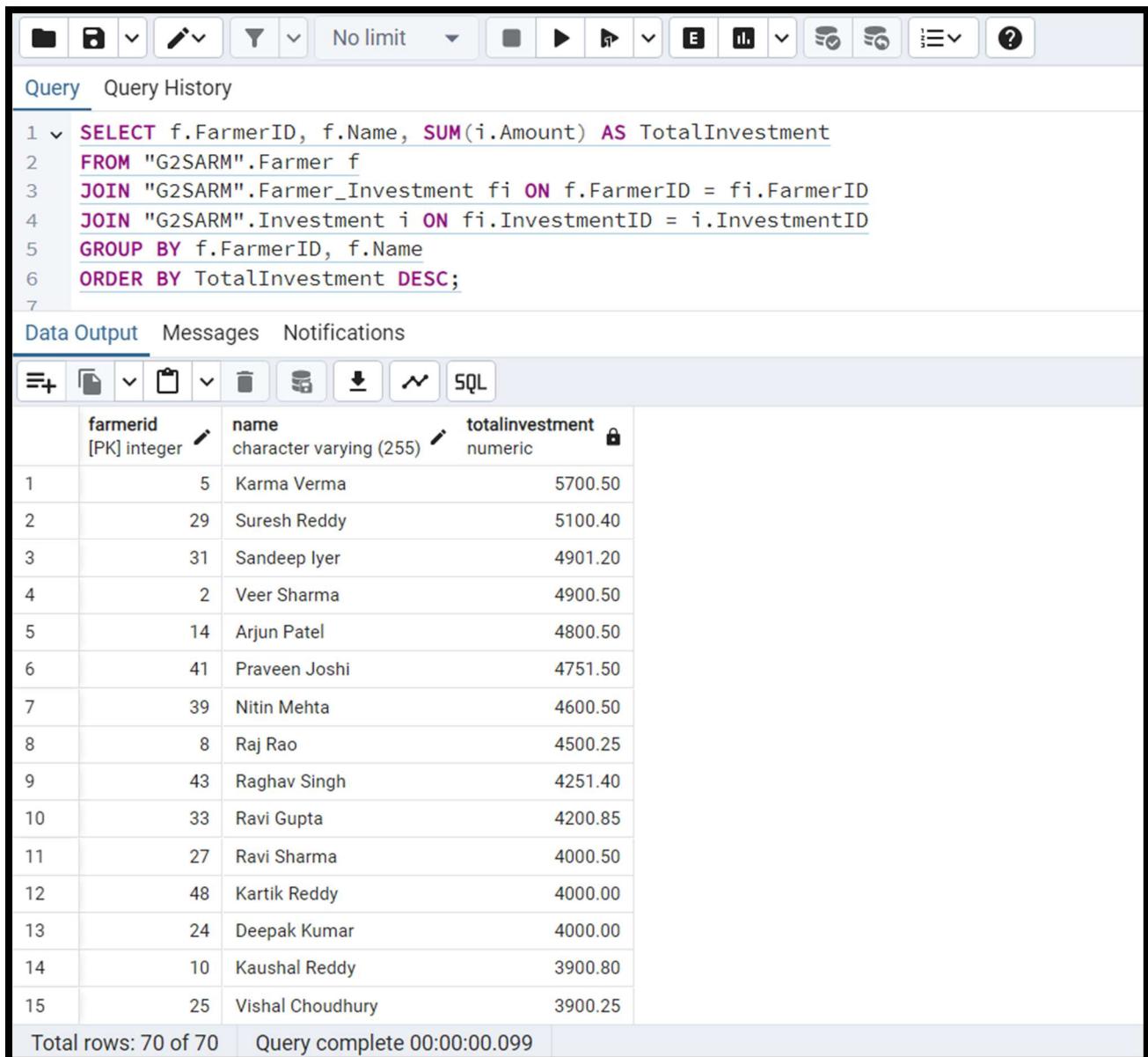
1  SELECT DISTINCT f.FarmerID, f.name, d.name
2  FROM "G2SARM".Farmer f
3  JOIN "G2SARM".Farmer_Crop fc ON f.FarmerID = fc.FarmerID
4  JOIN "G2SARM".Crop_Disease cd ON fc.CropID = cd.CropID
5  JOIN "G2SARM".cropdisease d ON cd.DiseaseID = d.DiseaseID
6  WHERE d.name = 'Blight';

```
- Data Output:** The results are displayed in a table with the following columns and data:

	farmerid	name	name
	integer	character varying (255)	character varying (255)
1	1	Manav Singh	Blight
2	2	Veer Sharma	Blight
3	4	Aamir Patel	Blight
4	9	Shiraz Desai	Blight
5	10	Kaushal Reddy	Blight

This query finds farmers who grow crops resistant to a specific disease, in this case, "Blight."

17. SELECT f.FarmerID, f.Name, SUM(i.Amount) AS TotalInvestment
 FROM "G2SARM".Farmer f
 JOIN "G2SARM".Farmer_Investment fi ON f.FarmerID = fi.FarmerID
 JOIN "G2SARM".Investment i ON fi.InvestmentID = i.InvestmentID
 GROUP BY f.FarmerID, f.Name
 ORDER BY TotalInvestment DESC;



The screenshot shows a database interface with a toolbar at the top containing various icons for file operations, search, and navigation. Below the toolbar, there are two tabs: 'Query' (which is selected) and 'Query History'. The 'Query' tab contains the SQL code for the query. The 'Data Output' tab is selected, displaying a table with 15 rows of data. The table has three columns: 'farmerid' (PK integer), 'name' (character varying (255)), and 'totalinvestment' (numeric). The data is sorted by 'totalinvestment' in descending order.

	farmerid [PK] integer	name character varying (255)	totalinvestment numeric
1	5	Karma Verma	5700.50
2	29	Suresh Reddy	5100.40
3	31	Sandeep Iyer	4901.20
4	2	Veer Sharma	4900.50
5	14	Arjun Patel	4800.50
6	41	Praveen Joshi	4751.50
7	39	Nitin Mehta	4600.50
8	8	Raj Rao	4500.25
9	43	Raghav Singh	4251.40
10	33	Ravi Gupta	4200.85
11	27	Ravi Sharma	4000.50
12	48	Kartik Reddy	4000.00
13	24	Deepak Kumar	4000.00
14	10	Kaushal Reddy	3900.80
15	25	Vishal Choudhury	3900.25

Total rows: 70 of 70 Query complete 00:00:00.099

This query provides the total investment amount received by each farmer, sorted in descending order.

```

18.   SELECT f.FarmerID, f.Name, s.SoilId
      FROM "G2SARM".Farmer f
      JOIN "G2SARM".Farmer_Crop fc ON f.FarmerID = fc.FarmerID
      JOIN "G2SARM".Crop_Soil cs ON fc.CropID = cs.CropID
      JOIN "G2SARM".Soil s ON cs.SoilID = s.SoilID
      WHERE s.phlevel > 7;

```

Query History

```

1 ▾ SELECT f.FarmerID, f.Name, s.SoilId
2 FROM "G2SARM".Farmer f
3 JOIN "G2SARM".Farmer_Crop fc ON f.FarmerID = fc.FarmerID
4 JOIN "G2SARM".Crop_Soil cs ON fc.CropID = cs.CropID
5 JOIN "G2SARM".Soil s ON cs.SoilID = s.SoilID
6 WHERE s.phlevel > 7;

```

Data Output Messages Notifications

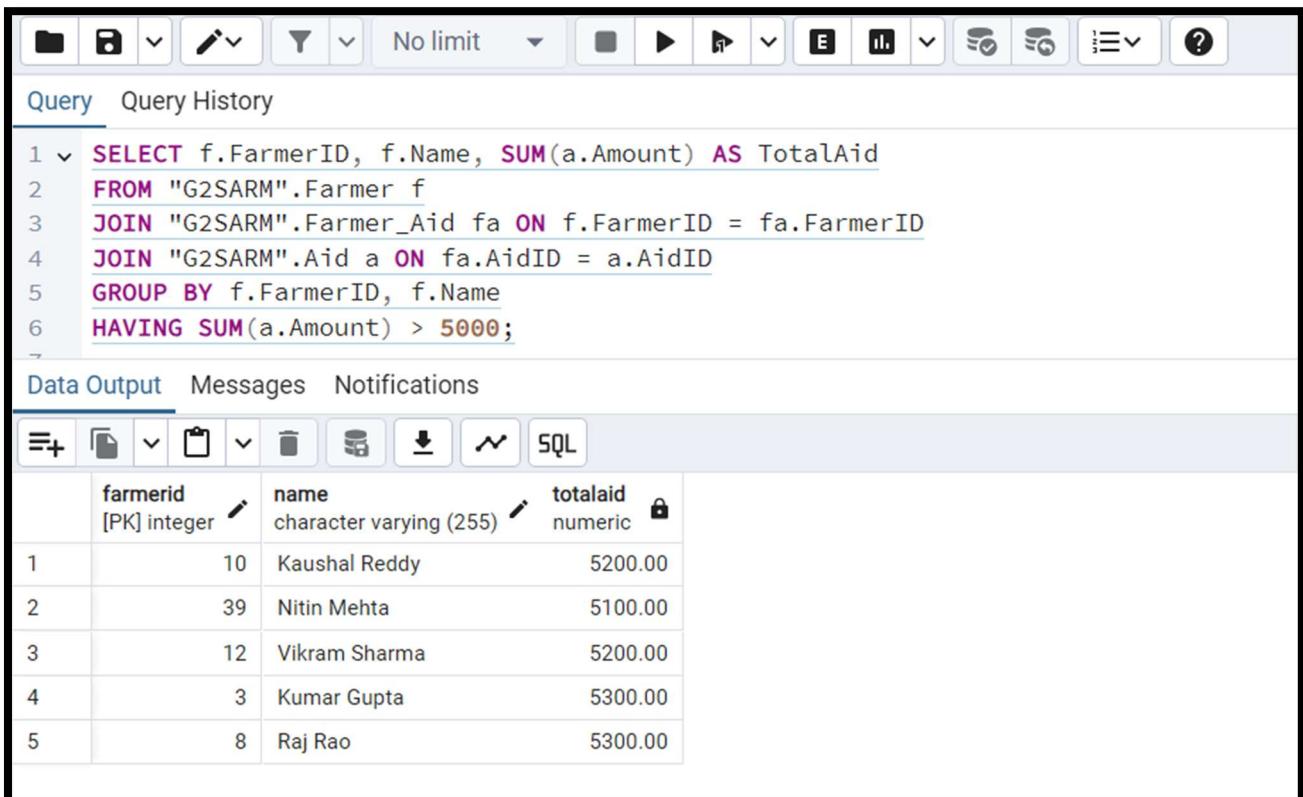
SQL

	farmerid integer	name character varying (255)	soilid integer
1	1	Manav Singh	6
2	1	Manav Singh	83
3	1	Manav Singh	73
4	2	Veer Sharma	64
5	2	Veer Sharma	10
6	3	Kumar Gupta	38
7	3	Kumar Gupta	12
8	3	Kumar Gupta	65
9	4	Aamir Patel	17
10	5	Karma Verma	77
11	5	Karma Verma	19
12	6	Ronak Joshi	88
13	6	Ronak Joshi	21
14	6	Ronak Joshi	55
15	7	Neel Mehta	57

Total rows: 17 of 17 Query complete 00:00:00.091

This query lists farmers who grow crops which is having the phlevel of soil is greater than 7.

19. SELECT f.FarmerID, f.Name, SUM(a.Amount) AS TotalAid
 FROM "G2SARM".Farmer f
 JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
 JOIN "G2SARM".Aid a ON fa.AidID = a.AidID
 GROUP BY f.FarmerID, f.Name
 HAVING SUM(a.Amount) > 5000;



The screenshot shows a database interface with a toolbar at the top and three tabs below it: 'Query', 'Query History', and 'Data Output'. The 'Query' tab is active, displaying the SQL code. The 'Data Output' tab is selected, showing a table with five rows of data.

```

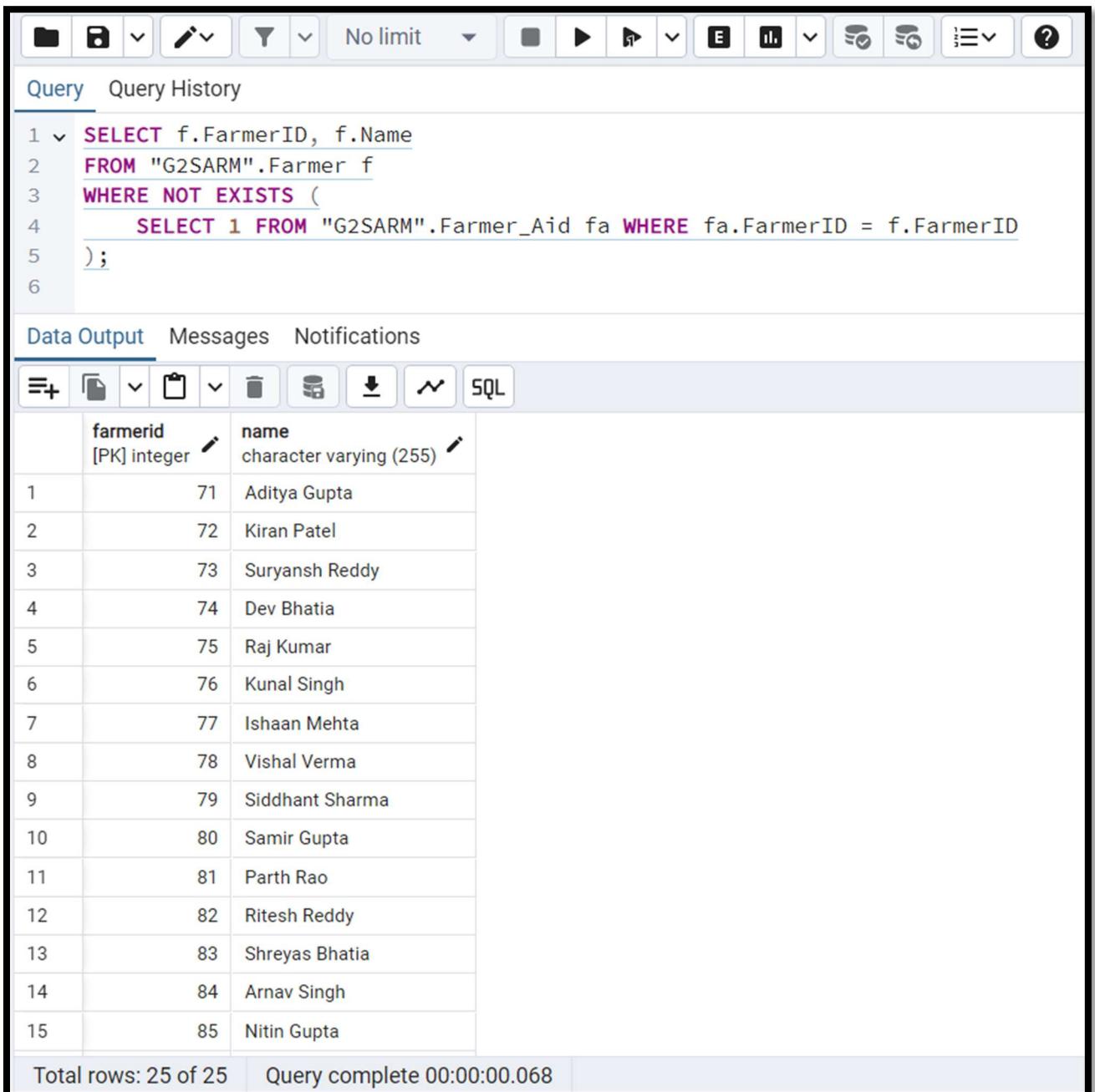
1  SELECT f.FarmerID, f.Name, SUM(a.Amount) AS TotalAid
2  FROM "G2SARM".Farmer f
3  JOIN "G2SARM".Farmer_Aid fa ON f.FarmerID = fa.FarmerID
4  JOIN "G2SARM".Aid a ON fa.AidID = a.AidID
5  GROUP BY f.FarmerID, f.Name
6  HAVING SUM(a.Amount) > 5000;

```

	farmerid [PK] integer	name character varying (255)	totalaid numeric
1	10	Kaushal Reddy	5200.00
2	39	Nitin Mehta	5100.00
3	12	Vikram Sharma	5200.00
4	3	Kumar Gupta	5300.00
5	8	Raj Rao	5300.00

This query finds farmers who have received aid exceeding a specified threshold of more than 5000.

20. SELECT f.FarmerID, f.Name
 FROM "G2SARM".Farmer f
 WHERE NOT EXISTS (
 SELECT 1 FROM "G2SARM".Farmer_Aid fa WHERE fa.FarmerID =
 f.FarmerID
);



The screenshot shows a SQL query editor with the following details:

- Toolbar:** Includes various icons for file operations, search, and database management.
- Query History:** Shows the executed SQL code in a step-by-step manner.
- Data Output:** Displays the results of the query as a table.
- Table Data:** The table has two columns: 'farmerid' and 'name'. The data is as follows:

	farmerid [PK] integer	name character varying (255)
1	71	Aditya Gupta
2	72	Kiran Patel
3	73	Suryansh Reddy
4	74	Dev Bhatia
5	75	Raj Kumar
6	76	Kunal Singh
7	77	Ishaan Mehta
8	78	Vishal Verma
9	79	Siddhant Sharma
10	80	Samir Gupta
11	81	Parth Rao
12	82	Ritesh Reddy
13	83	Shreyas Bhatia
14	84	Arnav Singh
15	85	Nitin Gupta

- Message Bar:** Shows "Total rows: 25 of 25" and "Query complete 00:00:00.068".

This query finds farmers who have not received any aid.