

---

# **Database Management System**

## **(IT615)**

## **Normalization and Schema Refinement**

**for**

### **Sustainable Agriculture Resource Management**

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

- Documentation of normalization & Schema Refinement Process upto 3NF/BCNF:

- List of redundancies existing for every schema which is part of the database.
- List of update, delete, and insert anomalies for every schema.

Schema	Redundancies	Anomalies		
		Update	Delete	Insert
Farmer	<p>ContactInfo could be redundant if multiple farmers share the same contact details (e.g., family members working on the same farm).</p> <p>If a farmer is involved in multiple activities or crops, repeating Name and FarmSize across different related tables could result in redundancy.</p>	<p>If a farmer's contact information changes, it needs to be updated in every record where the farmer is involved. Failure to update all records leads to inconsistencies.</p>	<p>If a farmer is deleted and this table contains important information like farm size, deleting the farmer might result in the loss of crucial data about their farm.</p>	<p>If a new farmer is added without full details (e.g., missing contact info), the database would reject the insert or leave partial information.</p>
Crop	Type, GrowthPeriod, and Yield might be redundant if the same crop is recorded multiple times (e.g., in crop-rotation cycles). This would lead to repeated data across different records.	If the growth period of a particular crop changes, all records for that crop must be updated. Failing to do so will lead to inconsistencies.	If a crop is deleted, relevant data on its yield and growth characteristics might be lost.	If a new crop is added with missing yield or growth period data, the insert may fail or lead to incomplete records.
Technology	Repetition of Type and Purpose across records of the same	If the purpose or type of a technology is	Deleting a technology record could	If a new technology is inserted

	technology can create redundancy, especially if the same technology is used by multiple farmers or in multiple locations.	updated, all related records must be updated. Otherwise, inconsistent technology information may exist in the database.	result in the loss of valuable information on how and why that technology was used.	without its purpose or type, the entry might be incomplete, leading to a less useful record.
Soil	If multiple records refer to the same Soil Type, the attributes like NutrientContent, pHLevel, and MoistureLevel might be repeated unnecessarily.	If the soil's nutrient content or moisture level changes, it needs to be updated across multiple records. Failure to update all records causes inconsistencies.	If a soil type is deleted, critical data about its composition and suitability for farming might be lost.	Missing values for key attributes like nutrient content could prevent the insertion of a complete soil record.
Fertilizer	Repeating NutrientContent and ApplicationMethod for the same type of fertilizer in multiple records can create redundancy.	If the application method for a specific fertilizer changes, every related record needs to be updated to avoid inconsistencies.	Deleting a fertilizer entry may lead to the loss of important data on its usage and nutrient composition.	Inserting a new fertilizer record without specifying its type or application method could result in an incomplete entry.
TrainingProgram	Repetition of TargetAudience and Content in multiple records of the same program may result in redundancy.	If the duration or content of a program changes, all records for that program need to be updated, which can lead to	Deleting a program could result in the loss of valuable information regarding its content and	If a new training program is inserted without specifying the target audience or content, the

		inconsistencies if not handled properly.	target audience.	record would be incomplete.
Visitor	ContactInfo could be redundant if multiple visitors share the same role or contact information (e.g., a group of people with the same representative contact).	If a visitor's contact information changes, all related records need to be updated. Failure to do so will lead to inconsistencies.	Deleting a visitor might lead to the loss of information about their role or contact info.	Inserting a visitor without contact info may result in incomplete records.
Policy	Repeated Type or ImplementationDate for multiple policies of the same name could result in redundancy.	If the implementation date of a policy is updated, it must be updated in all related records, or inconsistencies may occur.	Deleting a policy could result in losing valuable information about its type and implementation date.	Inserting a policy without providing the implementation date may lead to an incomplete record.
Aid	If the same Type of aid is repeatedly provided, the Amount and Date might be redundant across multiple records.	If the aid amount or date is changed, all related records must be updated to avoid inconsistencies.	Deleting an aid record might lead to the loss of important data regarding the beneficiary and the aid provided.	Inserting a record without specifying the aid amount or beneficiary might lead to incomplete data.
CropDisease	Repetition of Symptoms and Treatment for diseases affecting multiple crops could result in redundancy.	If the symptoms or treatment for a disease are updated, all	Deleting a disease record might result in the loss of critical information	Inserting a new disease without providing its symptoms or

		related records must be updated. Otherwise, inconsistencies will arise.	about its symptoms and treatment.	treatment may result in an incomplete record.
CropRotation	Duration and Benefits might be redundant if the same crop sequence is repeated in multiple records.	If the benefits or duration of a rotation scheme are updated, all related records must be updated to avoid inconsistencies.	Deleting a rotation entry could result in the loss of information about its benefits and duration.	Inserting a new rotation record without specifying the benefits could result in an incomplete entry.
Investment	Repeated Purpose and Amount for the same type of investment might result in redundancy.	If the purpose or amount of an investment is updated, all related records must be updated to avoid inconsistencies.	Deleting an investment record could lead to the loss of important data about the beneficiary and the purpose of the investment.	Inserting a new investment without specifying the beneficiary or amount could lead to an incomplete record.
Weather	Temperature and Precipitation might be redundant if recorded repeatedly for the same Date across different areas.	If the temperature or precipitation for a specific date changes, all related records must be updated.	Deleting a weather record could result in the loss of important weatherrelated data.	Inserting a new weather record without providing key information like temperature or conditions could lead to an incomplete record.
SustainablePractice	Repetition of Benefits and ImplementationLevel	If the benefits or implementation	Deleting a practice could lead to the loss	If a new practice is inserted

	for the same practice might result in redundancy.	level of a sustainable practice changes, all related records need to be updated to maintain consistency.	of valuable information about its benefits and implementation details.	without specifying its benefits, it may lead to an incomplete record.
FarmingTool	FarmerID may appear multiple times if different farmers use the same tool.	Updating a tool's details needs to be done for all records using that tool.	Deleting a tool may inadvertently lose information about its usage by various farmers.	Incomplete data entries, such as missing tool usage or manufacturer details, could occur.
Irrigation	Type may be repeated if multiple farmers use the same irrigation type.	If irrigation type or efficiency changes, all relevant records must be updated.	Deleting an irrigation record may lead to the loss of data relevant to that system's performance.	Missing details about coverage area or efficiency in new records could lead to incomplete datasets.

- Logic of arrived at the 3NF design step by step starting from the original design

### Step 1: First Normal Form (1NF)

All tables contain only atomic values (i.e., each field contains indivisible values) and that each entry is unique.

Schema	Atomic values	Status
1. Farmer(FarmerID, Name, FarmSize, ContactInfo)	TRUE	1NF
2. Crop(CropID, Name, Type, GrowthPeriod, Yield)	TRUE	1NF
3. Technology(TechnologyID, Name, Type, Purpose)	TRUE	1NF
4. Soil(SoilID, Type, NutrientContent, pHLevel, MoistureLevel)	TRUE	1NF
5. Fertilizer(FertilizerID, Name, Type, NutrientContent, ApplicationMethod)	TRUE	1NF
6. TrainingProgram(ProgramID, Title, Duration, TargetAudience, Content)	TRUE	1NF
7. Visitor(VisitorID, Name, Role, ContactInfo)	TRUE	1NF
8. Policy(PolicyID, Name, Type, ImplementationDate)	TRUE	1NF
9. Aid(AidID, Type, Amount, Beneficiary, Date)	TRUE	1NF
10. CropDisease(DiseaseID, Name, Symptoms, Treatment)	TRUE	1NF
11. CropRotation(RotationID, CropSequence, Duration, Benefits)	TRUE	1NF
12. Investment(InvestmentID, Amount, Purpose, Beneficiary, Date)	TRUE	1NF
13. Weather(WeatherID, Date, Temperature, Precipitation, Conditions)	TRUE	1NF
14. SustainablePractice(PracticeID, Name, Description, Benefits, ImplementationLevel)	TRUE	1NF
15. Farmer_Crop(FarmerID, CropID)	TRUE	1NF
16. Farmer_Tech(FarmerID, TechnologyID, TimeofUsage)	TRUE	1NF
17. FarmingTool(ToolID, Name, Type, Usage, Manufacturer, FarmerID)	TRUE	1NF
18. Irrigation(IrrigationID, Type, CoverageArea, Efficiency, InstallationDate, FarmerID)	TRUE	1NF
19. Farmer_Soil(FarmerID, SoilID)	TRUE	1NF
20. Farmer_Fertilizer(FarmerID, FertilizerID, Quantity)	TRUE	1NF
21. Farmer_TrainingProgram(FarmerID, ProgramID)	TRUE	1NF

22.Crop_Fertilizer(CropID, FertilizerID, FertilizerQuantity)	<b>TRUE</b>	<b>1NF</b>
23.Crop_Disease(CropID, DiseaseID)	<b>TRUE</b>	<b>1NF</b>
24.Farmer_Aid(FarmerID, AidID)	<b>TRUE</b>	<b>1NF</b>
25.Farmer_Investment(FarmerID, InvestmentID)	<b>TRUE</b>	<b>1NF</b>
26.Irrigation_Soil(IrrigationID, SoilID)	<b>TRUE</b>	<b>1NF</b>
27.Crop_Soil(CropID, SoilID)	<b>TRUE</b>	<b>1NF</b>
28.Crop_Weather(CropID, WeatherID)	<b>TRUE</b>	<b>1NF</b>
29.Farmer_SustainablePractice(FarmerID, PracticeID)	<b>TRUE</b>	<b>1NF</b>

## Step 2: Second Normal Form (2NF)

First ensure that database is in First Normal Form (1NF) and then remove any partial dependencies.

A partial dependency occurs when a non-key attribute is functionally dependent on part of a composite primary key rather than the whole key.

Now, all the schemas are in 1NF, we will analyze each schema to determine if any partial dependencies exist. If so, we will decompose those schemas.

Schema	Partial Dependencies	Status
1. Farmer(FarmerID, Name, FarmSize, ContactInfo)	<b>None (no composite key)</b>	<b>2NF</b>
2. Crop(CropID, Name, Type, GrowthPeriod, Yield)	<b>None (no composite key)</b>	<b>2NF</b>
3. Technology(TechnologyID, Name, Type, Purpose)	<b>None (no composite key)</b>	<b>2NF</b>
4. Soil(SoilID, Type, NutrientContent, pHLevel, MoistureLevel)	<b>None (no composite key)</b>	<b>2NF</b>
5. Fertilizer(FertilizerID, Name, Type, NutrientContent, ApplicationMethod)	<b>None (no composite key)</b>	<b>2NF</b>
6. TrainingProgram(ProgramID, Title, Duration, TargetAudience, Content)	<b>None (no composite key)</b>	<b>2NF</b>
7. Visitor(VisitorID, Name, Role, ContactInfo)	<b>None (no composite key)</b>	<b>2NF</b>
8. Policy(PolicyID, Name, Type, ImplementationDate)	<b>None (no composite key)</b>	<b>2NF</b>
9. Aid(AidID, Type, Amount, Beneficiary, Date)	<b>None (no composite key)</b>	<b>2NF</b>
10.CropDisease(DiseaseID, Name, Symptoms, Treatment)	<b>None (no composite key)</b>	<b>2NF</b>

11.CropRotation(RotationID, CropSequence, Duration, Benefits)	<b>None (no composite key)</b>	<b>2NF</b>
12.Investment(InvestmentID, Amount, Purpose, Beneficiary, Date)	<b>None (no composite key)</b>	<b>2NF</b>
13.Weather(WeatherID, Date, Temperature, Precipitation, Conditions)	<b>None (no composite key)</b>	<b>2NF</b>
14.SustainablePractice(PracticeID, Name, Description, Benefits, ImplementationLevel)	<b>None (no composite key)</b>	<b>2NF</b>
15.Farmer_Crop(FarmerID, CropID)	<b>None (no composite key)</b>	<b>2NF</b>
16.Farmer_Tech(FarmerID, TechnologyID, TimeofUsage)	<b>None (no composite key)</b>	<b>2NF</b>
17.FarmingTool(ToolID, Name, Type, Usage, Manufacturer, FarmerID)	<b>None (no composite key)</b>	<b>2NF</b>
18.Irrigation(IrrigationID, Type, CoverageArea, Efficiency, InstallationDate, FarmerID)	<b>None (no composite key)</b>	<b>2NF</b>
19.Farmer_Soil(FarmerID, SoilID)	<b>None (no composite key)</b>	<b>2NF</b>
20.Farmer_Fertilizer(FarmerID, FertilizerID, Quantity)	<b>None (no composite key)</b>	<b>2NF</b>
21.Farmer_TrainingProgram(FarmerID, ProgramID)	<b>None (no composite key)</b>	<b>2NF</b>
22.Crop_Fertilizer(CropID, FertilizerID, FertilizerQuantity)	<b>None (no composite key)</b>	<b>2NF</b>
23.Crop_Disease(CropID, DiseaseID)	<b>None (no composite key)</b>	<b>2NF</b>
24.Farmer_Aid(FarmerID, AidID)	<b>None (no composite key)</b>	<b>2NF</b>
25.Farmer_Investment(FarmerID, InvestmentID)	<b>None (no composite key)</b>	<b>2NF</b>
26.Irrigation_Soil(IrrigationID, SoilID)	<b>None (no composite key)</b>	<b>2NF</b>
27.Crop_Soil(CropID, SoilID)	<b>None (no composite key)</b>	<b>2NF</b>
28.Crop_Weather(CropID, WeatherID)	<b>None (no composite key)</b>	<b>2NF</b>
29.Farmer_SustainablePractice(FarmerID, PracticeID)	<b>None (no composite key)</b>	<b>2NF</b>

### Step 3: Third Normal Form (3NF)

To Third Normal Form (3NF), we need to remove any transitive dependencies, where a non-key attribute depends on another non-key attribute rather than directly on the primary key.

Schema	Transitive dependencies	Normalized Schema		Status
		Schema	New Schema	
1. Farmer(FarmerID, Name, FarmSize, ContactInfo)	Not present	-	-	3NF
2. Crop(CropID, Name, Type, GrowthPeriod, Yield)	Present	Crop(CropID, Name, GrowthPeriod, Yield,TypeID)	CropType(TypeID, Type, AdditionalInfo)	3NF
3. Technology(TechnologyID, Name, Type, Purpose)	Present	Technology(TechnologyID, Name, Purpose,TypeID)	TechnologyType(TypeID, Type, AdditionalInfo)	3NF
4. Soil(SoilID, Type, NutrientContent, pHLevel, MoistureLevel)	Present	Soil(SoilID, NutrientContent, pHLevel, MoistureLevel,TypeID)	SoilType(TypeID, Type, AdditionalProperties)	3NF
5. Fertilizer(FertilizerID, Name, Type, NutrientContent, ApplicationMethod)	Present	Fertilizer(FertilizerID, Name, NutrientContent, ApplicationMethod,TypeID)	FertilizerType(TypeID, Type, AdditionalInfo)	3NF
6. TrainingProgram(ProgramID, Title, Duration, TargetAudience, Content)	Present	TrainingProgram(ProgramID, Title, Duration, Content, AudienceID)	TrainingAudience(AudienceID, AudienceType, Description)	3NF
7. Visitor(VisitorID, Name, Role, ContactInfo)	Not Present	-	-	3NF
8. Policy(PolicyID, Name, Type, ImplementationDate)	Present	Policy(PolicyID, Name, ImplementationDate,TypeID)	PolicyType(TypeID, Type, Description)	3NF

9. Aid(AidID, Type, Amount, Beneficiary, Date)	<b>Present</b>	<b>Aid(AidID, Amount, Beneficiary, Date,TypeID)</b>	<b>AidType(TypeID, Type, Description)</b>	<b>3NF</b>
10. CropDisease(DiseaseID, Name, Symptoms, Treatment)	<b>Not present</b>	-	-	<b>3NF</b>
11. CropRotation(RotationID, CropSequence, Duration, Benefits)	<b>Present</b>	<b>CropRotation(RotationID, CropSequence, Duration)</b>	<b>CropBenefits(BenefitID, RotationID, Benefit)</b>	<b>3NF</b>
12. Investment(InvestmentID, Amount, Purpose, Beneficiary, Date)	<b>Present</b>	<b>Investment(InvestmentID, Amount, Beneficiary, Date, PurposeID)</b>	<b>InvestmentPurpose(PurposeID, Purpose, Description)</b>	<b>3NF</b>
13. Weather(WeatherID, Date, Temperature, Precipitation, Conditions)	<b>Not present</b>	-	-	<b>3NF</b>
14. SustainablePractice (PracticeID, Name, Description, Benefits, ImplementationLevel)	<b>Present</b>	<b>SustainablePractice(PracticeID, Name, Description, ImplementationLevel, BenefitID)</b>	<b>PracticeBenefits (BenefitID, PracticeID, Benefit)</b>	<b>3NF</b>
15. Farmer_Crop(FarmerID, CropID)	<b>Not present</b>	-	-	<b>3NF</b>
16. Farmer_Tech(FarmerID, TechnologyID, TimeofUsage)	<b>Not present</b>	-	-	<b>3NF</b>
17. FarmingTool(ToolID, Name, Type, Usage, Manufacturer, FarmerID)	<b>Present</b>	<b>FarmingTool(ToolID, Name, Usage, Manufacturer, FarmerID,TypeID)</b>	<b>ToolType(TypeID, Type, Description)</b>	<b>3NF</b>
18. Irrigation(IrrigationID, Type, CoverageArea, Efficiency,	<b>Present</b>	<b>Irrigation(IrrigationID, CoverageArea, Efficiency,</b>	<b>IrrigationType(TypeID, Type, Description)</b>	<b>3NF</b>

InstallationDate, FarmerID)		<b>InstallationDate, FarmerID, TypeID)</b>		
19.Farmer_Soil(FarmerID, SoilID)	<b>Not present</b>	-	-	<b>3NF</b>
20.Farmer_Fertilizer(FarmerID, FertilizerID, Quantity)	<b>Not present</b>	-	-	<b>3NF</b>
21.Farmer_TrainingProgram(FarmerID, ProgramID)	<b>Not present</b>	-	-	<b>3NF</b>
22.Crop_Fertilizer(CropID, FertilizerID, FertilizerQuantity)	<b>Not present</b>	-	-	<b>3NF</b>
23.Crop_Disease(CropID, DiseaseID)	<b>Not present</b>	-	-	<b>3NF</b>
24.Farmer_Aid(FarmerID, AidID)	<b>Not present</b>	-	-	<b>3NF</b>
25.Farmer_Investment(FarmerID, InvestmentID)	<b>Not present</b>	-	-	<b>3NF</b>
26.Irrigation_Soil(IrrigationID, SoilID)	<b>Not present</b>	-	-	<b>3NF</b>
27.Crop_Soil(CropID, SoilID)	<b>Not present</b>	-	-	<b>3NF</b>
28.Crop_Weather(CropID, WeatherID)	<b>Not present</b>	-	-	<b>3NF</b>
29.Farmer_SustainablePractice(FarmerID, PracticeID)	<b>Not present</b>	-	-	<b>3NF</b>

- **Final Schema**

1. Farmer(FarmerID, Name, FarmSize, ContactInfo)
2. Crop(CropID, Name, GrowthPeriod, Yield,TypeID)
3. CropType(TypeID, Type, AdditionalInfo)
4. Technology(TechnologyID, Name, Purpose,TypeID)
5. TechnologyType(TypeID, Type, AdditionalInfo)
6. Soil(SoilID, NutrientContent, pHLevel, MoistureLevel,TypeID)
7. SoilType(TypeID, Type, AdditionalProperties)
8. Fertilizer(FertilizerID, Name, NutrientContent, ApplicationMethod,TypeID)
9. FertilizerType(TypeID, Type, AdditionalInfo)
10. TrainingProgram(ProgramID, Title, Duration, Content, AudienceID)
11. TrainingAudience(AudienceID, AudienceType, Description)
12. Visitor(VisitorID, Name, Role, ContactInfo)
13. Policy(PolicyID, Name, ImplementationDate,TypeID)
14. PolicyType(TypeID, Type, Description)
15. Aid(AidID, Amount, Beneficiary, Date,TypeID)
16. AidType(TypeID, Type, Description)
17. CropDisease(DiseaseID, Name, Symptoms, Treatment)
18. CropRotation(RotationID, CropSequence, Duration)
19. CropBenefits(BenefitID, RotationID, Benefit)
20. Investment(InvestmentID, Amount, Beneficiary, Date, PurposeID)

21.InvestmentPurpose(PurposeID, Purpose, Description)

22.Weather(WeatherID, Date, Temperature, Precipitation, Conditions)

23.SustainablePractice(PracticeID, Name, Description, ImplementationLevel, BenefitID)

24.PracticeBenefits(BenefitID, PracticeID, Benefit)

25.Farmer\_Crop(FarmerID, CropID)

26.Farmer\_Tech(FarmerID, TechnologyID, TimeofUsage)

27.FarmingTool(ToolID, Name, Usage, Manufacturer, FarmerID, TypeID)

28.ToolType(TypeID, Type, Description)

29.Irrigation(IrrigationID, CoverageArea, Efficiency, InstallationDate, FarmerID, TypeID)

30.IrrigationType(TypeID, Type, Description)

31.Farmer\_Soil(FarmerID, SoilID)

32.Farmer\_Fertilizer(FarmerID, FertilizerID, Quantity)

33.Farmer\_TrainingProgram(FarmerID, ProgramID)

34.Crop\_Fertilizer(CropID, FertilizerID, FertilizerQuantity)

35.Crop\_Disease(CropID, DiseaseID)

36.Farmer\_Aid(FarmerID, AidID)

37.Farmer\_Investment(FarmerID, InvestmentID)

38.Irrigation\_Soil(IrrigationID, SoilID)

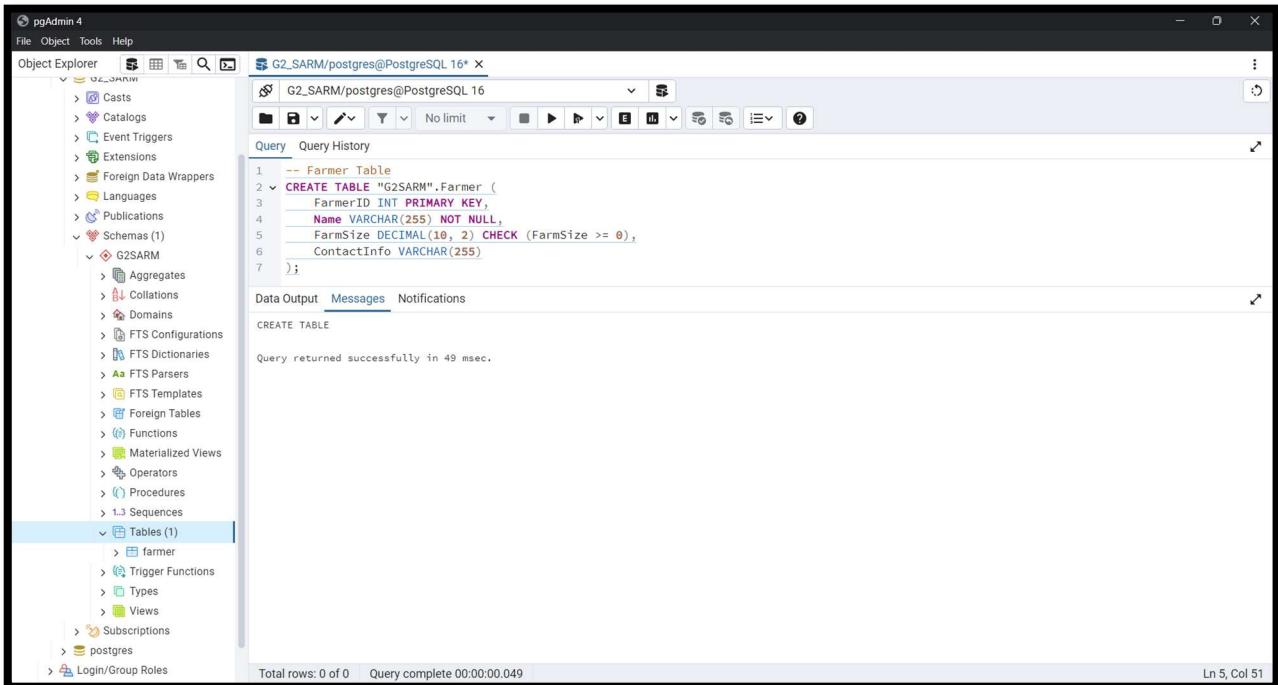
39.Crop\_Soil(CropID, SoilID)

## 40.Crop\_Weather(CropID, WeatherID)

## 41.Farmer\_SustainablePractice(FarmerID, PracticeID)

### • DDL Snapshots of all schemas

#### 1. Farmer(FarmerID, Name, FarmSize, ContactInfo)



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

- Object Explorer:** Shows the database schema with various objects like Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), and Tables (1).
- Query Editor:** The current query is:

```
-- Farmer Table
CREATE TABLE "G2SARM".farmer (
    FarmerID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    FarmSize DECIMAL(10, 2) CHECK (FarmSize >= 0),
    ContactInfo VARCHAR(255)
);
```

- Data Output:** Shows the message: "Query returned successfully in 49 msec."
- Messages:** No messages displayed.
- Notifications:** No notifications displayed.

#### 2. Crop(CropID, Name, GrowthPeriod, Yield,TypeID)



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

- Query Editor:** The current query is:

```
-- Crop Table
CREATE TABLE "G2SARM".Crop (
    CropID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    GrowthPeriod INT CHECK (GrowthPeriod > 0),
    Yield DECIMAL(10, 2) CHECK (Yield >= 0),
    TypeID INT,
    FOREIGN KEY (TypeID) REFERENCES "G2SARM".CropType(TypeID) ON DELETE SET NULL
);
```

- Data Output:** Shows the message: "CREATE TABLE" and "Query returned successfully in 56 msec."
- Messages:** No messages displayed.
- Notifications:** No notifications displayed.

### 3. CropType(TypeID, Type, AdditionalInfo)

The screenshot shows the pgAdmin 4 interface with a query editor window. The connection is set to G2\_SARM/postgres@PostgreSQL 16. The query tab is active, displaying the SQL code for creating the CropType table:

```
1 -- CropType Table
2 CREATE TABLE "G2SARM".CropType (
3    TypeID INT PRIMARY KEY,
4     Type VARCHAR(255) NOT NULL,
5     AdditionalInfo TEXT
6 );
7
```

Below the code, the Data Output tab is selected, showing the executed command and its successful execution message:

```
CREATE TABLE
```

Query returned successfully in 52 msec.

### 4. Technology(TechnologyID, Name, Purpose, TypeID)

The screenshot shows the pgAdmin 4 interface with a query editor window. The connection is set to G2\_SARM/postgres@PostgreSQL 16. The query tab is active, displaying the SQL code for creating the Technology table:

```
1 CREATE TABLE "G2SARM".Technology (
2     TechnologyID INT PRIMARY KEY,
3     Name VARCHAR(255) NOT NULL,
4     Purpose TEXT,
5     TypeID INT,
6     FOREIGN KEY (TypeID) REFERENCES "G2SARM".TechnologyType(TypeID) ON DELETE SET NULL
7 );
8
```

Below the code, the Data Output tab is selected, showing the executed command and its successful execution message:

```
CREATE TABLE
```

Query returned successfully in 56 msec.

### 5. TechnologyType(TypeID, Type, AdditionalInfo)

The screenshot shows the pgAdmin 4 interface with the connection set to G2\_SARM/postgres@PostgreSQL 16. The query editor window is open, displaying the SQL code for creating the TechnologyType table. The code is as follows:

```
1 -- TechnologyType TABLE
2 CREATE TABLE "G2SARM".TechnologyType (
3     TypeID INT PRIMARY KEY,
4     Type VARCHAR(255) NOT NULL,
5     AdditionalInfo TEXT
6 );
7
```

The 'Messages' tab is selected, showing the output of the query:

```
CREATE TABLE
```

Query returned successfully in 50 msec.

## 6. Soil(SoilID, NutrientContent, pHLevel, MoistureLevel,TypeID)

The screenshot shows the pgAdmin 4 interface with the connection set to G2\_SARM/postgres@PostgreSQL 16. The query editor window is open, displaying the SQL code for creating the Soil table. The code is as follows:

```
1 CREATE TABLE "G2SARM".Soil (
2     SoilID INT PRIMARY KEY,
3     NutrientContent TEXT,
4     pHLevel DECIMAL(3, 2),
5     MoistureLevel DECIMAL(5, 2),
6     TypeID INT,
7     FOREIGN KEY (TypeID) REFERENCES "G2SARM".SoilType(TypeID) ON DELETE SET NULL
8 );
```

The 'Messages' tab is selected, showing the output of the query:

```
CREATE TABLE
```

Query returned successfully in 65 msec.

## 7. SoilType(TypeID, Type, AdditionalProperties)

The screenshot shows the pgAdmin 4 interface with a connection to G2\_SARM/postgres@PostgreSQL 16. The query editor contains the following SQL code:

```
1 ✓ CREATE TABLE "G2SARM".SoilType (
2     TypeID INT PRIMARY KEY,
3     Type VARCHAR(255) NOT NULL,
4     AdditionalProperties TEXT
5 );
6
```

The 'Messages' tab is selected, showing the output:

```
CREATE TABLE
```

Query returned successfully in 50 msec.

## 8. Fertilizer(FertilizerID, Name, NutrientContent, ApplicationMethod,TypeID)

The screenshot shows the pgAdmin 4 interface with a connection to G2\_SARM/postgres@PostgreSQL 16. The query editor contains the following SQL code:

```
1 ✓ CREATE TABLE "G2SARM".Fertilizer (
2     FertilizerID INT PRIMARY KEY,
3     Name VARCHAR(255) NOT NULL,
4     NutrientContent TEXT,
5     ApplicationMethod TEXT,
6     TypeID INT,
7     FOREIGN KEY (TypeID) REFERENCES "G2SARM".FertilizerType(TypeID) ON DELETE SET NULL
8 );
```

The 'Messages' tab is selected, showing the output:

```
CREATE TABLE
```

Query returned successfully in 78 msec.

## 9. FertilizerType(TypeID, Type, AdditionalInfo)

G2\_SARM/postgres@PostgreSQL 16

```
1 v CREATE TABLE "G2SARM".FertilizerType (
2     TypeID INT PRIMARY KEY,
3     Type VARCHAR(255) NOT NULL,
4     AdditionalInfo TEXT
5 );
6
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 60 msec.

## 10.TrainingProgram(ProgramID, Title, Duration, Content, AudienceID)

G2\_SARM/postgres@PostgreSQL 16

```
1 v CREATE TABLE "G2SARM".TrainingProgram (
2     ProgramID INT PRIMARY KEY,
3     Title VARCHAR(255) NOT NULL,
4     Duration INT,
5     Content TEXT,
6     AudienceID INT,
7     FOREIGN KEY (AudienceID) REFERENCES "G2SARM".TrainingAudience(AudienceID) ON DELETE SET NULL
8 );
9
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 60 msec.

## 11.TrainingAudience(AudienceID, AudienceType, Description)

The screenshot shows the pgAdmin 4 interface with a connection to G2\_SARM/postgres@PostgreSQL 16. The query editor contains the following SQL code:

```
1 CREATE TABLE "G2SARM".TrainingAudience (
2     AudienceID INT PRIMARY KEY,
3     AudienceType VARCHAR(255) NOT NULL,
4     Description TEXT
5 );
6
```

The 'Messages' tab is selected, showing the output:

```
CREATE TABLE
```

Query returned successfully in 204 msec.

## 12. Visitor(VisitorID, Name, Role, ContactInfo)

The screenshot shows the pgAdmin 4 interface with a connection to G2\_SARM/postgres@PostgreSQL 16. The query editor contains the following SQL code:

```
1 -- Visitor Table
2 CREATE TABLE "G2SARM".Visitor (
3     VisitorID INT PRIMARY KEY,
4     Name VARCHAR(255) NOT NULL,
5     Role VARCHAR(255),
6     ContactInfo VARCHAR(255)
7 );
8
```

The 'Messages' tab is selected, showing the output:

```
CREATE TABLE
```

Query returned successfully in 54 msec.

## 13. Policy(PolicyID, Name, ImplementationDate,TypeID)

Query History

```
1 v CREATE TABLE "G2SARM".Policy (
2     PolicyID INT PRIMARY KEY,
3     Name VARCHAR(255) NOT NULL,
4     ImplementationDate DATE,
5    TypeID INT,
6     FOREIGN KEY (TypeID) REFERENCES "G2SARM".PolicyType(TypeID) ON DELETE SET NULL
7 );
8
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 49 msec.

#### 14.PolicyType(TypeID, Type, Description)

G2\_SARM/postgres@PostgreSQL 16

Query History

```
1 v CREATE TABLE "G2SARM".PolicyType (
2     TypeID INT PRIMARY KEY,
3     Type VARCHAR(255) NOT NULL,
4     Description TEXT
5 );
6
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 53 msec.

#### 15.Aid(AidID, Amount, Beneficiary, Date, TypeID)

G2\_SARM/postgres@PostgreSQL 16

No limit

CREATE TABLE "G2SARM".Aid ( AidID INT PRIMARY KEY, Amount DECIMAL(10, 2), Beneficiary VARCHAR(255), Date DATE,TypeID INT, FOREIGN KEY (TypeID) REFERENCES "G2SARM".AidType(TypeID) ON DELETE SET NULL );

CREATE TABLE

Query returned successfully in 56 msec.

## 16.AidType(TypeID, Type, Description)

G2\_SARM/postgres@PostgreSQL 16

No limit

CREATE TABLE "G2SARM".AidType (TypeID INT PRIMARY KEY, Type VARCHAR(255) NOT NULL, Description TEXT);

CREATE TABLE

Query returned successfully in 55 msec.

## 17.CropDisease(DiseaseID, Name, Symptoms, Treatment)

G2\_SARM/postgres@PostgreSQL 16

Query History

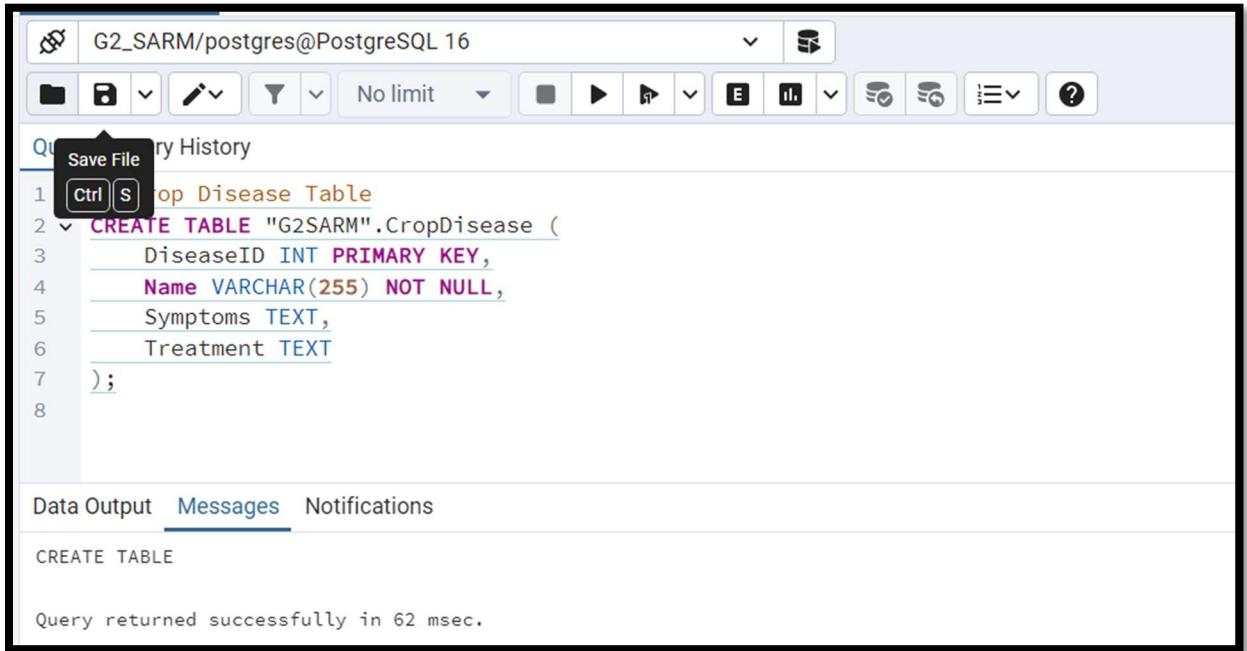
Save File **Ctrl + S**

```
1 op Disease Table
2 CREATE TABLE "G2SARM".CropDisease (
3     DiseaseID INT PRIMARY KEY,
4     Name VARCHAR(255) NOT NULL,
5     Symptoms TEXT,
6     Treatment TEXT
7 );
8
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 62 msec.



### 18.CropRotation(RotationID, CropSequence, Duration)

G2\_SARM/postgres@PostgreSQL 16

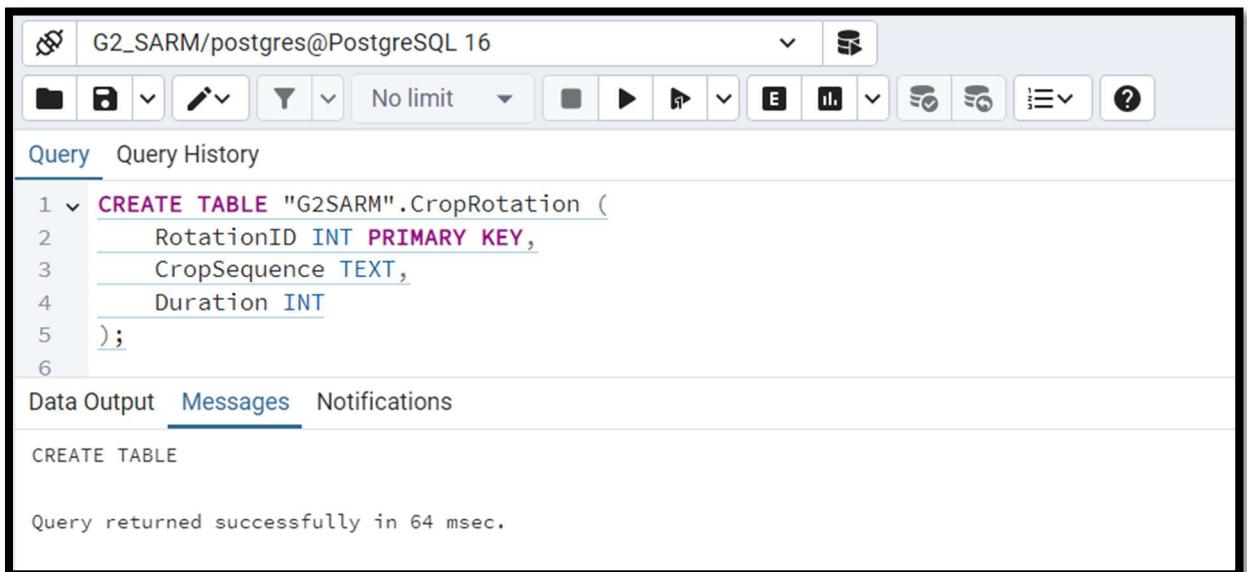
Query History

```
1 CREATE TABLE "G2SARM".CropRotation (
2     RotationID INT PRIMARY KEY,
3     CropSequence TEXT,
4     Duration INT
5 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 64 msec.



### 19.CropBenefits(BenefitID, RotationID, Benefit)

G2\_SARM/postgres@PostgreSQL 16

```
1 v CREATE TABLE "G2SARM".CropBenefits (
2     BenefitID INT PRIMARY KEY,
3     RotationID INT,
4     Benefit TEXT,
5     FOREIGN KEY (RotationID) REFERENCES "G2SARM".CropRotation(RotationID) ON DELETE CASCADE
6 );
7
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 89 msec.

## 20.Investment(InvestmentID, Amount, Beneficiary, Date, PurposeID)

G2\_SARM/postgres@PostgreSQL 16

```
1 v CREATE TABLE "G2SARM".Investment (
2     InvestmentID INT PRIMARY KEY,
3     Amount DECIMAL(10, 2),
4     Beneficiary VARCHAR(255),
5     Date DATE,
6     PurposeID INT,
7     FOREIGN KEY (PurposeID) REFERENCES "G2SARM".InvestmentPurpose(PurposeID) ON DELETE SET NULL
8 );
9
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 86 msec.

## 21.InvestmentPurpose(PurposeID, Purpose, Description)

The screenshot shows the pgAdmin 4 interface with a connection to 'G2\_SARM/postgres@PostgreSQL 16'. The query editor contains the following SQL code:

```
1 v CREATE TABLE "G2SARM".InvestmentPurpose (
2     PurposeID INT PRIMARY KEY,
3     Purpose VARCHAR(255) NOT NULL,
4     Description TEXT
5 );
6
```

The 'Messages' tab is selected, showing the output:

```
CREATE TABLE
```

Query returned successfully in 54 msec.

## 22. Weather(WeatherID, Date, Temperature, Precipitation, Conditions)

The screenshot shows the pgAdmin 4 interface with a connection to 'G2\_SARM/postgres@PostgreSQL 16'. The query editor contains the following SQL code:

```
1 -- Weather Table
2 v CREATE TABLE "G2SARM".Weather (
3     WeatherID INT PRIMARY KEY,
4     Date DATE NOT NULL,
5     Temperature DECIMAL(5, 2),
6     Precipitation DECIMAL(5, 2),
7     Conditions VARCHAR(255)
8 );
9
```

The 'Messages' tab is selected, showing the output:

```
CREATE TABLE
```

Query returned successfully in 53 msec.

## 23. SustainablePractice(PracticeID, Name, Description, ImplementationLevel)

The screenshot shows the pgAdmin 4 interface with a connection to G2\_SARM/postgres@PostgreSQL 16. The query editor contains the following SQL code:

```
1 v CREATE TABLE "G2SARM".SustainablePractice (
2     PracticeID INT PRIMARY KEY,
3     Name VARCHAR(255) NOT NULL,
4     Description TEXT,
5     ImplementationLevel VARCHAR(255)
6 );
7
```

The 'Messages' tab shows the execution results:

```
CREATE TABLE

Query returned successfully in 49 msec.
```

## 24. PracticeBenefits(BenefitID, PracticeID, Benefit)

The screenshot shows the pgAdmin 4 interface with a connection to G2\_SARM/postgres@PostgreSQL 16. The query editor contains the following SQL code:

```
1 v CREATE TABLE "G2SARM".PracticeBenefits (
2     BenefitID INT PRIMARY KEY,
3     PracticeID INT,
4     Benefit TEXT,
5     FOREIGN KEY (PracticeID) REFERENCES "G2SARM".SustainablePractice(PracticeID) ON DELETE CASCADE
6 );
```

The 'Messages' tab shows the execution results:

```
CREATE TABLE

Query returned successfully in 58 msec.
```

## 25. Farmer\_Crop(FarmerID, CropID)

G2\_SARM/postgres@PostgreSQL 16

Query History

```
1 -- Farmer_Crop Table
2 v CREATE TABLE "G2SARM".Farmer_Crop (
3     FarmerID INT,
4     CropID INT,
5     PRIMARY KEY (FarmerID, CropID),
6     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
7     FOREIGN KEY (CropID) REFERENCES "G2SARM".Crop(CropID) ON DELETE CASCADE
8 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 59 msec.

## 26.Farmer\_Tech(FarmerID, TechnologyID, TimeofUsage)

G2\_SARM/postgres@PostgreSQL 16

Query History

```
1 -- Farmer_Tech Table
2 v CREATE TABLE "G2SARM".Farmer_Tech (
3     FarmerID INT,
4     TechnologyID INT,
5     TimeofUsage DATE NOT NULL,
6     PRIMARY KEY (FarmerID, TechnologyID),
7     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
8     FOREIGN KEY (TechnologyID) REFERENCES "G2SARM".Technology(TechnologyID) ON DELETE CASCADE
9 );
10
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 218 msec.

## 27.FarmingTool(ToolID, Name, Usage, Manufacturer, FarmerID,TypeID)

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar says "G2\_SARM/postgres@PostgreSQL 16". The toolbar has various icons for database management. The main area is titled "Query" and contains the following SQL code:

```
1 -- FarmingTool Table
2 v CREATE TABLE "G2SARM".FarmingTool (
3     ToolID INT PRIMARY KEY,
4     Name VARCHAR(255) NOT NULL,
5     Usage VARCHAR(255) NOT NULL,
6     Manufacturer VARCHAR(255),
7     FarmerID INT,
8    TypeID INT,
9     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE SET NULL,
10    FOREIGN KEY (TypeID) REFERENCES "G2SARM".ToolType(TypeID) ON DELETE SET NULL
11 );
```

Below the code, there are tabs for "Data Output", "Messages" (which is selected), and "Notifications". The "Messages" tab shows the output of the query:

```
CREATE TABLE

Query returned successfully in 65 msec.
```

## 28. ToolType(TypeID, Type, Description)

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar says "G2\_SARM/postgres@PostgreSQL 16". The toolbar has various icons for database management. The main area is titled "Query" and contains the following SQL code:

```
1 -- ToolType Table
2 v CREATE TABLE "G2SARM".ToolType (
3     TypeID INT PRIMARY KEY,
4     Type VARCHAR(255) NOT NULL,
5     Description TEXT
6 );
```

Below the code, there are tabs for "Data Output", "Messages" (which is selected), and "Notifications". The "Messages" tab shows the output of the query:

```
CREATE TABLE

Query returned successfully in 65 msec.
```

## 29. Irrigation(IrrigationID, CoverageArea, Efficiency, InstallationDate, FarmerID, TypeID)

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar indicates the connection is to G2\_SARM/postgres@PostgreSQL 16. The toolbar has various icons for database management. The main area is titled 'Query' and contains the following SQL code:

```
1 -- Irrigation Table
2 v CREATE TABLE "G2SARM".Irrigation (
3     IrrigationID INT PRIMARY KEY,
4     CoverageArea DECIMAL(10, 2) CHECK (CoverageArea >= 0),
5     Efficiency DECIMAL(5, 2) CHECK (Efficiency BETWEEN 0 AND 100),
6     InstallationDate DATE NOT NULL,
7     FarmerID INT,
8     TypeID INT,
9     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE SET NULL,
10    FOREIGN KEY (TypeID) REFERENCES "G2SARM".IrrigationType(TypeID) ON DELETE SET NULL
11 );
```

Below the code, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the output of the query:

```
CREATE TABLE
```

Query returned successfully in 54 msec.

### 30.IrrigationType(TypeID, Type, Description)

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar indicates the connection is to G2\_SARM/postgres@PostgreSQL 16. The toolbar has various icons for database management. The main area is titled 'Query' and contains the following SQL code:

```
1 -- IrrigationType Table
2 v CREATE TABLE "G2SARM".IrrigationType (
3     TypeID INT PRIMARY KEY,
4     Type VARCHAR(255) NOT NULL,
5     Description TEXT
6 );
```

Below the code, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the output of the query:

```
CREATE TABLE
```

Query returned successfully in 62 msec.

### 31.Farmer\_Soil(FarmerID, SoilID)

G2\_SARM/postgres@PostgreSQL 16

Query History

```
1 -- Farmer_Soil Table
2 v CREATE TABLE "G2SARM".Farmer_Soil (
3     FarmerID INT,
4     SoilID INT,
5     PRIMARY KEY (FarmerID, SoilID),
6     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
7     FOREIGN KEY (SoilID) REFERENCES "G2SARM".Soil(SoilID) ON DELETE CASCADE
8 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 154 msec.

### 32.Farmer\_Fertilizer(FarmerID, FertilizerID, Quantity)

G2\_SARM/postgres@PostgreSQL 16

Query History

```
1 -- Farmer_Fertilizer Table
2 v CREATE TABLE "G2SARM".Farmer_Fertilizer (
3     FarmerID INT,
4     FertilizerID INT,
5     Quantity DECIMAL(10, 2) CHECK (Quantity >= 0),
6     PRIMARY KEY (FarmerID, FertilizerID),
7     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
8     FOREIGN KEY (FertilizerID) REFERENCES "G2SARM".Fertilizer(FertilizerID) ON DELETE CASCADE
9 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 56 msec.

### 33.Farmer\_TrainingProgram(FarmerID, ProgramID)

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar says "G2\_SARM/postgres@PostgreSQL 16". The toolbar has various icons for database management. The query history tab is selected, showing the creation of the "Farmer\_TrainingProgram" table. The table structure includes columns for FarmerID and ProgramID, with primary key constraints and foreign key references to the "Farmer" and "TrainingProgram" tables respectively, both with cascade delete options. The data output tab shows the successful execution of the CREATE TABLE statement.

```
-- Farmer_TrainingProgram Table
CREATE TABLE "G2SARM".Farmer_TrainingProgram (
    FarmerID INT,
    ProgramID INT,
    PRIMARY KEY (FarmerID, ProgramID),
    FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
    FOREIGN KEY (ProgramID) REFERENCES "G2SARM".TrainingProgram(ProgramID) ON DELETE CASCADE
);
```

CREATE TABLE

Query returned successfully in 56 msec.

### 34.Crop\_Fertilizer(CropID, FertilizerID, FertilizerQuantity)

The screenshot shows the pgAdmin 4 interface with a query editor window. The title bar says "G2\_SARM/postgres@PostgreSQL 16". The toolbar has various icons for database management. The query history tab is selected, showing the creation of the "Crop\_Fertilizer" table. The table structure includes columns for CropID and FertilizerID, with a primary key constraint and foreign key references to the "Crop" and "Fertilizer" tables respectively, both with cascade delete options. A CHECK constraint ensures that FertilizerQuantity is greater than or equal to 0. The data output tab shows the successful execution of the CREATE TABLE statement.

```
-- Crop_Fertilizer Table
CREATE TABLE "G2SARM".Crop_Fertilizer (
    CropID INT,
    FertilizerID INT,
    FertilizerQuantity DECIMAL(10, 2) CHECK (FertilizerQuantity >= 0),
    PRIMARY KEY (CropID, FertilizerID),
    FOREIGN KEY (CropID) REFERENCES "G2SARM".Crop(CropID) ON DELETE CASCADE,
    FOREIGN KEY (FertilizerID) REFERENCES "G2SARM".Fertilizer(FertilizerID) ON DELETE CASCADE
);
```

CREATE TABLE

Query returned successfully in 54 msec.

### 35.Crop\_Disease(CropID, DiseaseID)

G2\_SARM/postgres@PostgreSQL 16

No limit

Query History

```
-- Crop_Disease Table
CREATE TABLE "G2SARM".Crop_Disease (
    CropID INT,
    DiseaseID INT,
    PRIMARY KEY (CropID, DiseaseID),
    FOREIGN KEY (CropID) REFERENCES "G2SARM".Crop(CropID) ON DELETE CASCADE,
    FOREIGN KEY (DiseaseID) REFERENCES "G2SARM".CropDisease(DiseaseID) ON DELETE CASCADE
);
```

Farmer\_Aid Table

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 51 msec.

### 36.Farmer\_Aid(FarmerID, AidID)

G2\_SARM/postgres@PostgreSQL 16

No limit

Query History

```
-- Farmer_Aid Table
CREATE TABLE "G2SARM".Farmer_Aid (
    FarmerID INT,
    AidID INT,
    PRIMARY KEY (FarmerID, AidID),
    FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
    FOREIGN KEY (AidID) REFERENCES "G2SARM".Aid(AidID) ON DELETE CASCADE
);
```

Farmer\_Aid Table

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 57 msec.

### 37.Farmer\_Investment(FarmerID, InvestmentID)

G2\_SARM/postgres@PostgreSQL 16

No limit

Query History

```
1 -- Farmer_Investment Table
2 v CREATE TABLE "G2SARM".Farmer_Investment (
3   FarmerID INT,
4   InvestmentID INT,
5   PRIMARY KEY (FarmerID, InvestmentID),
6   FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
7   FOREIGN KEY (InvestmentID) REFERENCES "G2SARM".Investment(InvestmentID) ON DELETE CASCADE
8 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 54 msec.

### 38.Irrigation\_Soil(IrrigationID, SoilID)

Query History

```
1 -- Irrigation_Soil Table
2 v CREATE TABLE "G2SARM".Irrigation_Soil (
3   IrrigationID INT,
4   SoilID INT,
5   PRIMARY KEY (IrrigationID, SoilID),
6   FOREIGN KEY (IrrigationID) REFERENCES "G2SARM".Irrigation(IrrigationID) ON DELETE CASCADE,
7   FOREIGN KEY (SoilID) REFERENCES "G2SARM".Soil(SoilID) ON DELETE CASCADE
8 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 98 msec.

### 39.Crop\_Soil(CropID, SoilID)

G2\_SARM/postgres@PostgreSQL 16

No limit

CREATE TABLE "G2SARM".Crop\_Soil (

CropID INT,

SoilID INT,

PRIMARY KEY (CropID, SoilID),

FOREIGN KEY (CropID) REFERENCES "G2SARM".Crop(CropID) ON DELETE CASCADE,

FOREIGN KEY (SoilID) REFERENCES "G2SARM".Soil(SoilID) ON DELETE CASCADE

);

CREATE TABLE

Query returned successfully in 50 msec.

#### 40.Crop\_Weather(CropID, WeatherID)

G2\_SARM/postgres@PostgreSQL 16

No limit

CREATE TABLE "G2SARM".Crop\_Weather (

CropID INT,

WeatherID INT,

PRIMARY KEY (CropID, WeatherID),

FOREIGN KEY (CropID) REFERENCES "G2SARM".Crop(CropID) ON DELETE CASCADE,

FOREIGN KEY (WeatherID) REFERENCES "G2SARM".Weather(WeatherID) ON DELETE CASCADE

);

CREATE TABLE

Query returned successfully in 140 msec.

#### 41.Farmer\_SustainablePractice(FarmerID, PracticeID)

G2\_SARM/postgres@PostgreSQL 16

No limit

Query History

```

1 -- Farmer_SustainablePractice Table
2 CREATE TABLE "G2SARM".Farmer_SustainablePractice (
3     FarmerID INT,
4     PracticeID INT,
5     PRIMARY KEY (FarmerID, PracticeID),
6     FOREIGN KEY (FarmerID) REFERENCES "G2SARM".Farmer(FarmerID) ON DELETE CASCADE,
7     FOREIGN KEY (PracticeID) REFERENCES "G2SARM".SustainablePractice(PracticeID) ON DELETE CASCADE
8 );

```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 172 msec.

- **Data Snapshots of all the tables after insertion of data**

1. Farmer(FarmerID, Name, FarmSize, ContactInfo)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and a query result grid on the right. The Object Explorer displays various tables under the 'Tables (41)' section. The query result grid shows the following data:

farmerid	name	farmsize	contactinfo
1	Manav Singh	20.50	manav.singh@example.com
2	Veer Sharma	61.00	veer.sharma@example.com
3	Kumar Gupta	15.50	kumar.gupta@example.com
4	Aamir Patel	30.00	aamir.patel@example.com
5	Karma Verma	74.25	karma.verma@example.com
6	Ronak Joshi	50.00	ronak.joshi@example.com
7	Neel Mehta	30.00	neel.mehta@example.com
8	Raj Rao	11.00	raj.rao@example.com
9	Shiraz Desai	27.00	shiraz.desai@example.com
10	Kaushal Reddy	80.00	kaushal.reddy@example.com
11	Aarav Singh	22.50	aarav.singh@example.com
12	Vikram Sharma	65.00	vikram.sharma@example.com
13	Rahul Gupta	10.50	rahul.gupta@example.com
14	Arjun Patel	33.00	arjun.patel@example.com
15	Karan Verma	77.25	karan.verma@example.com
16	Rohit Joshi	55.00	rohit.joshi@example.com
17	Nikhil Mehta	40.00	nikhil.mehta@example.com
18	Rajesh Rao	12.00	rajesh.rao@example.com
19	Siddharth Desai	27.00	siddharth.desai@example.com

Total rows: 95 of 95    Query complete 00:00:00.076    Ln 1, Col 31

**Number of records: 95**

2. Crop(CropID, Name, GrowthPeriod, Yield,TypeID)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and a query editor on the right. The query editor displays the results of the following SQL statement:

```
1  SELECT * FROM "G2SARM".CROP;
```

The results are presented in a table with the following columns: cropid, name, growthperiod, yield, and typeid. The data consists of 19 rows of crop information.

cropid	name	growthperiod	yield	typeid
1	Wheat	120	3.50	1
2	Rice	150	4.00	1
3	Barley	90	3.00	1
4	Oats	100	2.80	1
5	Corn	90	4.20	1
6	Soybean	100	2.75	2
7	Chickpea	110	1.50	2
8	Lentil	80	1.80	2
9	Pea	70	1.20	2
10	Black Bean	90	1.40	2
11	Carrot	90	1.50	3
12	Tomato	80	2.00	3
13	Lettuce	30	0.80	3
14	Potato	120	4.00	3
15	Onion	120	2.50	3
16	Apple	180	5.00	4
17	Banana	365	8.00	4
18	Orange	365	7.50	4
19	Strawberry	120	2.00	4

Message bar: Successfully run. Total query runtime: 108 msec. 90 rows affected. Ln 1, Col 28

**Number of records: 90**

### 3. CropType(TypeID, Type, AdditionalInfo)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and a query editor on the right. The query editor displays the results of the following SQL statement:

```
1  SELECT * FROM "G2SARM".CROPTYPE;
```

The results are presented in a table with the following columns: typeid, type, and additionalinfo. The data consists of 19 rows of crop type information.

typeid	type	additionalinfo
1	Cereal	Includes grains such as wheat, rice, and corn.
2	Legume	Includes beans, peas, lentils, and peanuts.
3	Vegetable	Includes leafy greens, root vegetables, and tubers.
4	Fruit	Includes apples, bananas, berries, and citrus.
5	Herb	Includes culinary herbs like basil, parsley, and cilantro.
6	Oilseed	Includes crops like canola, sunflower, and soybeans.
7	Tubers	Includes potatoes, sweet potatoes, and yams.
8	Pseudocereal	Includes quinoa, amaranth, and buckwheat.
9	Spice	Includes crops like black pepper, vanilla, and cinnamon.
10	Nut	Includes almonds, walnuts, cashews, and pecans.
11	Berries	Includes strawberries, blueberries, and raspberries.
12	Fungi	Includes mushrooms and truffles.
13	Aquatic Plants	Includes watercress, lotus, and algae.
14	Citrus	Includes oranges, lemons, limes, and grapefruits.
15	Drupes	Includes peaches, cherries, and plums.
16	Melons	Includes watermelons and cantaloupes.
17	Root Crops	Includes carrots, beets, and turnips.
18	Green Manure	Includes cover crops used to improve soil.
19	Fiber	Includes cotton, jute, and hemp.

Message bar: Successfully run. Total query runtime: 119 msec. 90 rows affected. Ln 1, Col 32

**Number of records: 90**

### 4. Technology(TechnologyID, Name, Purpose,TypeID)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query Editor on the right. The Query Editor displays the following SQL query:

```
1 SELECT * FROM "G2SARM".TECHNOLOGY;
```

The results are shown in a table:

technologyid	name	purpose	typeid
1	Drip Irrigation	Efficient water management in agriculture	1
2	Sprinkler System	Uniform distribution of water to crops	1
3	Soil Sensors	Monitoring soil moisture and nutrient levels	2
4	Weather Forecasting Tools	Predicting weather conditions for better planning	3
5	Precision Farming Software	Optimizing field variability	4
6	Drones for Crop Monitoring	Aerial surveillance of crop health	3
7	Automated Irrigation Systems	Automating irrigation based on weather and soil conditions	1
8	Crop Rotation Software	Planning crop rotation to enhance soil health	7
9	Pest Control Drones	Targeted pest management using drones	5
10	Soil Testing Kits	Analyzing soil properties for better crop choices	2
11	Greenhouse Technology	Controlling climate for optimal plant growth	6
12	Hydroponic Systems	Growing plants without soil	6
13	Aquaponics Systems	Combining aquaculture with hydroponics	6
14	Plant Disease Diagnosis Apps	Identifying diseases through image recognition	5
15	Mobile Farming Apps	Managing farming tasks and schedules on mobile devices	4
16	Crop Yield Prediction Models	Forecasting crop yields based on various factors	4
17	Automated Harvesters	Mechanizing the harvesting process	9
18	Robotic Weeders	Weeding crops using robotics	3
19	Nutrient Management Software	Managing nutrient application for crops	

Total rows: 90 of 90    Query complete 00:00:00.070    Successfully run. Total query runtime: 70 msec. 90 rows affected.

**Number of records: 90**

## 5. TechnologyType(TypeID, Type, AdditionalInfo)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query Editor on the right. The Query Editor displays the following SQL query:

```
1 SELECT * FROM "G2SARM".TECHNOLOGYTYPE;
```

The results are shown in a table:

typeid	type	additionalinfo
1	Irrigation	Technologies for efficient water usage in farming
2	Soil Management	Tools for soil health and monitoring
3	Crop Monitoring	Systems for tracking crop growth and health
4	Farm Management Software	Software applications for farm planning and management
5	Pest and Disease Control	Methods for managing pests and diseases in crops
6	Greenhouse and Indoor Farming	Technologies for controlled environment farming
7	Sustainable Practices	Practices and tools for sustainable agriculture
8	Agroecology	Ecological approaches to agriculture
9	Post-Harvest Management	Technologies for handling crops after harvest
10	Livestock Management	Sustainable livestock farming methods
11	Renewable Energy	Alternative energy sources for agriculture
12	Supply Chain and Market Access	Platforms connecting farmers to markets
13	Training and Community Support	Programs and services for farmer education and support
14	Food and Water Safety	Technologies ensuring safety in food and water for farming
15	Genetic Modification	Biotechnology for crop improvement
16	Post-Harvest Technology	Technologies for handling harvested crops
17	Climate Control	Systems to manage climate conditions in agriculture.
18	Farm Management Software	Software for overall farm management
19	Organic Farming Technology	Technologies specific to organic practices.

Total rows: 90 of 90    Query complete 00:00:00.067    Successfully run. Total query runtime: 67 msec. 90 rows affected.

**Number of records: 90**

## 6. Soil(SoilID, NutrientContent, pHLevel, MoistureLevel,TypeID)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and a query editor on the right. The query editor displays the following SQL code:

```
1  SELECT * FROM "G2SARM".SOIL;
```

The results are shown in a table with the following columns: soilid [PK] integer, nutrientcontent text, phlevel numeric (3,2), moisturelevel numeric (5,2), and typeid integer. The data consists of 19 rows of soil types and their properties.

soilid	nutrientcontent	phlevel	moisturelevel	typeid
1	High Nitrogen	6.50	25.00	1
2	Low Phosphorus	7.00	30.00	2
3	Balanced Nutrients	6.80	22.50	3
4	High Potassium	5.80	20.00	4
5	Organic Matter Rich	6.30	45.00	5
6	Sandy Soil	7.20	10.00	6
7	Clayey Soil	5.50	35.00	7
8	Loamy Soil	6.40	28.00	8
9	Peaty Soil	5.90	50.00	9
10	Saline Soil	8.00	5.00	10
11	Acidic Soil	4.80	60.00	11
12	Alkaline Soil	8.20	15.00	12
13	Nutrient Deficient	6.00	32.00	13
14	Fertile Soil	6.70	40.00	14
15	Silty Soil	6.90	35.00	15
16	Heavy Clay	5.70	20.00	7
17	Light Sand	7.50	12.00	6
18	Moderately Acidic	5.60	55.00	11
19	Highly Saline	9.00	2.00	10

Total rows: 90 of 90    Query complete 00:00:00.069    Successfully run. Total query runtime: 69 msec. 90 rows affected.

**Number of records: 90**

## 7. SoilType(TypeID, Type, AdditionalProperties)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and a query editor on the right. The query editor displays the following SQL code:

```
1  SELECT * FROM "G2SARM".SOILTYPE;
```

The results are shown in a table with the following columns: typeid [PK] integer, type character varying (255), and additionalproperties text. The data consists of 19 rows of soil types and their properties.

typeid	type	additionalproperties
1	Nitrogen-Rich Soil	High nitrogen content, suitable for leafy crops
2	Phosphorus-Low Soil	Requires phosphorus supplementation
3	Balanced Nutrients Soil	Suitable for a variety of crops
4	Potassium-Rich Soil	High in potassium, ideal for root vegetables
5	Organic Matter Rich Soil	High organic content for organic farming
6	Sandy Soil	Good drainage, low nutrient retention
7	Clayey Soil	High water retention, suitable for water-demanding crops
8	Loamy Soil	Ideal structure for most crops
9	Peaty Soil	High organic matter; retains moisture well
10	Saline Soil	High salt content, requires salt-tolerant crops
11	Acidic Soil	Low pH, suitable for acid-loving plants
12	Alkaline Soil	High pH, suitable for alkaline-loving plants
13	Nutrient Deficient Soil	Requires additional fertilization
14	Fertile Soil	Rich in nutrients and ideal for most crops
15	Silty Soil	Good nutrient and moisture retention
16	Desert Soil	Very low moisture content, often sandy, supporting minimal vegetation
17	Volcanic Soil	Fertile soil formed from volcanic ash, rich in nutrients
18	Mountain Soil	Thin, acidic soil found in mountainous regions, poor in nutrients
19	Regur Soil	Clayey soil rich in black color, known for its moisture retention

Total rows: 90 of 90    Query complete 00:00:00.133    Successfully run. Total query runtime: 133 msec. 90 rows affected.

**Number of records: 90**

## 8. Fertilizer(FertilizerID, Name, NutrientContent, ApplicationMethod,TypeID)

Object Explorer    DDL file.sql\*    G2\_SARM/postgres@PostgreSQL 16

Tables (41)

**FERTILIZER**

```
1 SELECT * FROM "G2SARM".FERTILIZER;
```

Data Output Messages Notifications

fertilizerid [PK] integer	name character varying (255)	nutrientcontent text	applicationmethod text	typeid integer
1	Nitrogen Fertilizer	Nitrogen 20%	Granular Application	1
2	Phosphate Fertilizer	Phosphorus 18%	Liquid Fertilization	1
3	Potash Fertilizer	Potassium 24%	Broadcasting	1
4	Urea	Nitrogen 46%	Top Dressing	1
5	Organic Compost	Nutrient-Rich	Soil Incorporation	2
6	Fish Emulsion	Nitrogen 5%	Foliar Feeding	2
7	Blood Meal	Nitrogen 12%	Granular Application	2
8	Chicken Manure	Nitrogen 1.5%	Granular Application	2
9	Slow-Release Fertilizer	Balanced Nutrients	Granular Application	3
10	Liquid Seaweed	Trace Minerals	Foliar Spray	4
11	Granular NPK Fertilizer	NPK 10-20-10	Soil Application	7
12	Specialized Fertilizer	Tailored Nutrients	Granular Application	8
13	Organic Liquid Fertilizer	Nutrient-Rich	Foliar Feeding	2
14	Controlled-Release Fertilizer	Balanced Nutrients	Soil Application	3
15	Biofertilizer	Microbial Culture	Soil Application	6
16	Calcium Nitrate	Calcium 15%	Drip Irrigation	1
17	Potassium Nitrate	Potassium 44%	Granular Application	1
18	Superphosphate	Phosphorus 20%	Broadcasting	1
19	Ammonium Sulfate	Nitrogen 21%	Top Dressing	1

Total rows: 80 of 80    Query complete 00:00:00.069    ✓ Successfully run. Total query runtime: 69 msec. 80 rows affected.    Ln 1, Col 34

**Number of records: 80**

## 9. FertilizerType(TypeID, Type, AdditionalInfo)

Object Explorer    DDL file.sql\*    G2\_SARM/postgres@PostgreSQL 16

**FERTILIZERTYPE**

```
1 SELECT * FROM "G2SARM".FERTILIZERTYPE;
```

Data Output Messages Notifications

typeid [PK] integer	type character varying (255)	additionalinfo text
1	Synthetic	Chemically manufactured fertilizers
2	Organic	Derived from natural sources
3	Slow-Release	Nutrients released gradually over time
4	Liquid	Dissolved in water for quick absorption
5	Granular	Solid particles for direct application
6	Biofertilizer	Contains living microorganisms
7	NPK	Contains Nitrogen, Phosphorus, and Potassium
8	Specialty	Tailored for specific crops or conditions
9	Compost	Decomposed organic matter used to enrich soil.
10	Manure	Animal waste used as a fertilizer, rich in nutrients.
11	Fish Emulsion	Liquid fertilizer made from fish byproducts, high in nitrogen.
12	Bone Meal	Ground animal bones, high in phosphorus.
13	Blood Meal	Dried animal blood, high in nitrogen.
14	Kelp Meal	Seaweed-based fertilizer, rich in trace minerals.
15	Green Manure	Crops grown to be plowed back into the soil for nutrients.
16	Mulch	Layer of organic material spread over soil to retain moisture.
17	Superphosphate	Highly soluble phosphorus fertilizer.
18	Triple Superphosphate	Concentrated form of superphosphate with high phosphorus cont...
19	Ammonium Nitrate	Quick-release nitrogen fertilizer.

Total rows: 90 of 90    Query complete 00:00:00.072    ✓ Successfully run. Total query runtime: 72 msec. 90 rows affected.    Ln 1, Col 38

**Number of records: 80**

## 10. TrainingProgram(ProgramID, Title, Duration, Content, AudienceID)

```

Object Explorer   DDL file.sql* 
G2_SARM/postgres@PostgreSQL 16
Tables (41)
aid
aidtype
crop
crop_disease
crop_fertilizer
crop_soil
crop_weather
cropbenefits
cropdisease
croprotection
croptype
farmer
farmer_aid
farmer_crop
farmer_fertilizer
farmer_investment
farmer_soil
farmer_sustainable
farmer_tech
farmer_trainingprogram
farmingtool
fertilizer
fertilizertype
investment
investmentpurpose
irrigation
irrigation_soil
irrigationstype

Query  Query History
1  SELECT * FROM "G2SARM".TRAININGPROGRAM;

Data Output  Messages  Notifications
SQL

programid [PK] integer title character varying(255) duration integer content text audienceid integer
1 1 Sustainable Farming Techniques 30 Learn about sustainable practices and crop rotation. 1
2 2 Soil Health Management 45 Understanding soil composition and health monitoring. 1
3 3 Water Conservation Methods 60 Techniques for efficient water usage in agriculture. 1
4 4 Integrated Pest Management 35 Strategies to manage pests while minimizing chemical... 1
5 5 Organic Farming Practices 50 Principles and practices of organic farming. 1
6 6 Precision Agriculture 40 Utilizing technology for optimized farming practices. 1
7 7 Climate Resilience in Farming 25 Adapting farming practices to climate change. 1
8 8 Farm Business Management 55 Managing farm operations and financial planning. 1
9 9 Crop Diversification Strategies 30 Benefits and methods of diversifying crops. 1
10 10 Sustainable Livestock Management 60 Best practices for raising livestock sustainably. 1
11 11 Composting Techniques 20 How to create and use compost effectively. 1
12 12 Herbicide and Pesticide Alternatives 45 Exploring natural pest management alternatives. 1
13 13 Advanced Irrigation Techniques 50 Innovative methods to improve irrigation efficiency. 1
14 14 Agroforestry Practices 40 Integrating trees into agricultural landscapes. 1
15 15 Farming with Native Plants 35 Utilizing native plants for sustainable farming. 1
16 16 Farm Machinery and Maintenance 50 Proper use and maintenance of farm machinery. 1
17 17 Youth Engagement in Agriculture 30 Programs to attract youth to farming careers. 7
18 18 Sustainable Certification Processes 45 Understanding certification for sustainable practices. 1
19 19 Digital Tools for Farmers 40 Using apps and tools for better farm management

Total rows: 80 of 80  Query complete 00:00:00.073

```

Successfully run. Total query runtime: 73 msec. 80 rows affected. Ln 1, Col 39

**Number of records: 80**

## 11. TrainingAudience(AudienceID, AudienceType, Description)

```

Object Explorer   DDL file.sql* 
G2_SARM/postgres@PostgreSQL 16
Tables (41)
aid
aidtype
crop
crop_disease
crop_fertilizer
crop_soil
crop_weather
cropbenefits
cropdisease
croprotection
croptype
farmer
farmer_aid
farmer_crop
farmer_fertilizer
farmer_investment
farmer_soil
farmer_sustainable
farmer_tech
farmer_trainingprogram
farmingtool
fertilizer
fertilizertype
investment
investmentpurpose
irrigation
irrigation_soil
irrigationstype

Query  Query History
1  SELECT * FROM "G2SARM".TRAININGAUDIENCE;

Data Output  Messages  Notifications
SQL

audienceid [PK] integer audiencetype character varying(255) description text
1 1 Farmers Individuals engaged in farming activities seeking knowledge on sustainable practices...
2 2 Agricultural Students Students pursuing education in agriculture or related fields.
3 3 Extension Workers Professionals providing educational support to farmers.
4 4 Researchers Individuals conducting research in agricultural sciences.
5 5 Policy Makers Government officials involved in agricultural policy and planning.
6 6 Agribusiness Professionals Individuals working in agricultural business sectors.
7 7 Youth Young individuals interested in pursuing careers in agriculture.
8 8 Women in Agriculture Women engaged in agricultural activities and initiatives.
9 9 Community Leaders Leaders working to promote agricultural development in communities.
10 10 Environmentalists Individuals focused on environmental protection related to agriculture.
11 11 Agri-entrepreneurs Individuals starting businesses in the agricultural sector.
12 12 Sustainable Agriculture Advocates Promoters of sustainable farming practices.
13 13 Agrochemical Sales Representatives Sales personnel for agricultural chemicals.
14 14 Irrigation Specialists Experts in irrigation systems and water management.
15 15 Soil Health Advisors Professionals providing soil health recommendations.
16 16 Farm Managers Individuals managing farm operations and labor.
17 17 Veterinarians Animal health professionals in livestock farming.
18 18 Food Safety Inspectors Officials ensuring food safety standards.
19 19 Organic Farmers Farmers focused on organic agricultural practices.

Total rows: 90 of 90  Query complete 00:00:00.068

```

Successfully run. Total query runtime: 68 msec. 90 rows affected. Ln 1, Col 40

**Number of records: 90**

## 12. Visitor(VisitorID, Name, Role, ContactInfo)

```

SELECT * FROM "G2SARM".VISITOR;

```

visitorid	name	role	contactinfo
1	Aarav Patel	Researcher	aarav.patel@example.com
2	Vivaan Sharma	Student	vivaan.sharma@example.com
3	Aditya Mehta	Researcher	aditya.mehta@example.com
4	Vihaan Joshi	Student	vihaan.joshi@example.com
5	Reyansh Singh	Researcher	reyansh.singh@example.com
6	Rohan Kapoor	Student	rohan.kapoor@example.com
7	Krishna Nair	Researcher	krishna.nair@example.com
8	Lakshya Gupta	Student	lakshya.gupta@example.com
9	Kunal Verma	Researcher	kunal.verma@example.com
10	Samarth Iyer	Student	samarth.iyer@example.com
11	Aryan Choudhury	Researcher	aryan.choudhury@example.com
12	Aniket Rao	Student	aniket.rao@example.com
13	Yash Bansal	Researcher	yash.bansal@example.com
14	Advik Jain	Student	advik.jain@example.com
15	Ayaan Reddy	Researcher	ayaan.reddy@example.com
16	Dev Agarwal	Student	dev.agarwal@example.com
17	Siddharth Malhotra	Researcher	siddharth.malhotra@example.com
18	Nishant Ghosh	Student	nishant.ghosh@example.com
19	Harsh Desai	Researcher	harsh.desai@example.com

Total rows: 100 of 100    Query complete 00:00:00.067    Ln 1, Col 31

**Number of records: 100**

### 13. Policy(PolicyID, Name, ImplementationDate,TypeID)

```

SELECT * FROM "G2SARM".POLICY;

```

policyid	name	implementationdate	typeid
1	National Agriculture Policy	2020-01-15	2
2	Soil Health Management Program	2020-02-20	1
3	Integrated Pest Management Policy	2020-03-25	1
4	Water Conservation Initiative	2020-04-10	11
5	Climate Resilient Agriculture Program	2020-05-05	13
6	Organic Farming Promotion Scheme	2020-06-15	14
7	Fertilizer Subsidy Policy	2020-07-01	8
8	Rural Employment Generation Scheme	2020-08-12	3
9	Precision Farming Policy	2020-09-18	4
10	Sustainable Livestock Development Policy	2020-10-30	2
11	National Agroforestry Policy	2020-11-15	1
12	Digital Agriculture Initiative	2020-12-01	4
13	Crop Insurance Scheme	2021-01-10	8
14	Women in Agriculture Program	2021-02-14	3
15	Food Security Policy	2021-03-16	10
16	Agri-Startup Policy	2021-04-20	2
17	National Dairy Development Policy	2021-05-25	2
18	National Fisheries Policy	2021-06-30	2
19	National Seed Policy	2021-07-18	1

Total rows: 85 of 85    Query complete 00:00:00.071    Ln 1, Col 30

**Number of records: 85**

### 14. PolicyType(TypeID, Type, Description)

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows the database schema with tables like aid, aidtype, crop, etc.
- Query Editor:** Contains the SQL command: `SELECT * FROM "G2SARM".POLICYTYPE;`
- Data Output:** Displays the results of the query, which lists 19 rows of policy types with their descriptions.
- Status Bar:** Shows "Successfully run. Total query runtime: 283 msec. 90 rows affected." and "Ln 1, Col 34".

typid	type	description
1	Environmental	Policies aimed at promoting sustainable environmental practices.
2	Economic	Policies that support economic growth and stability in agriculture.
3	Social	Policies focused on social equity and community development in agriculture.
4	Technological	Policies that encourage the adoption of innovative agricultural technologies.
5	Regulatory	Policies designed to regulate agricultural practices for sustainability.
6	Research	Policies that promote agricultural research and development.
7	Education	Policies aimed at improving agricultural education and training.
8	Financial	Policies that provide financial support and incentives for farmers.
9	Market Access	Policies that facilitate access to markets for agricultural products.
10	Food Security	Policies that ensure food security and nutrition for all.
11	Water Management	Policies that promote sustainable water management practices.
12	Land Use	Policies focusing on the sustainable use of land resources.
13	Climate Adaptation	Policies that help farmers adapt to climate change impacts.
14	Organic Farming	Policies that promote organic farming practices.
15	Biodiversity	Policies aimed at conserving agricultural biodiversity.
16	Agroecology Policy	Policies promoting ecological approaches to agriculture.
17	Farming Technology Policy	Support for the adoption of agricultural technologies.
18	Renewable Energy Policy	Regulations encouraging the use of renewable energy in agriculture.
19	Youth Engagement Policy	Initiatives to engage young people in agriculture.

**Number of records: 90**

## 15.Aid(AidID, Amount, Beneficiary, Date,TypeID)

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows the database schema with tables like aid, aidtype, crop, etc.
- Query Editor:** Contains the SQL command: `SELECT * FROM "G2SARM".AID;`
- Data Output:** Displays the results of the query, which lists 90 rows of aid records with columns: aidid, amount, beneficiary, date, and typeid.
- Status Bar:** Shows "Successfully run. Total query runtime: 113 msec. 90 rows affected." and "Ln 1, Col 27".

aidid	amount	beneficiary	date	typeid
1	3000.00	1	2024-01-15	1
2	1500.00	2	2024-01-20	2
3	2500.00	3	2024-02-01	3
4	1800.00	4	2024-02-15	4
5	3200.00	5	2024-02-28	5
6	2100.00	6	2024-03-05	6
7	1900.00	7	2024-03-10	1
8	2000.00	8	2024-03-20	2
9	2300.00	9	2024-03-25	3
10	1600.00	10	2024-04-01	4
11	1750.00	11	2024-04-05	5
12	1800.00	12	2024-04-10	6
13	2600.00	13	2024-04-15	1
14	2700.00	14	2024-04-20	2
15	1500.00	15	2024-05-01	3
16	2900.00	16	2024-05-10	4
17	2300.00	17	2024-05-15	5
18	1600.00	18	2024-05-20	6
19	2400.00	19	2024-06-01	1

**Number of records: 90**

## 16.AidType(TypeID, Type, Description)

Object Explorer    DDL file.sql\*    G2\_SARM/postgres@PostgreSQL 16

```
1 SELECT * FROM "G2SARM".AIDTYPE;
```

Execute script

Query History

typeid	type	description
1	Financial Aid	Monetary support for various needs
2	Food Aid	Provision of food to those in need
3	Educational Aid	Support for educational purposes
4	Medical Aid	Health-related assistance and resources
5	Technical Assistance	Support for technical projects and initiatives
6	Emergency Aid	Immediate help during crises or disasters
7	Crop Insurance	Financial protection against crop losses due to unforeseen events.
8	Market Access Programs	Programs designed to help farmers reach markets.
9	Research Grants	Funding for agricultural research projects.
10	Training Programs	Educational programs for farmers on best practices.
11	Soil Health Initiatives	Programs aimed at improving soil quality and health.
12	Pest Management Support	Assistance with integrated pest management strategies.
13	Water Conservation Programs	Initiatives promoting efficient water use in agriculture.
14	Youth Empowerment Initiatives	Programs aimed at engaging and empowering young farmers.
15	Women in Agriculture Support	Support programs focused on women farmers.
16	Sustainable Farming Grants	Grants for projects promoting sustainable farming practices.
17	Organic Farming Support	Assistance for transitioning to organic farming methods.
18	Community Development Aid	Support for rural community development initiatives.
19	Food Security Programs	Programs aimed at improving food security for vulnerable populations.

Total rows: 80 of 80    Query complete 00:00:00.109    Successfully run. Total query runtime: 109 msec. 80 rows affected. Ln 1, Col 31

**Number of records: 80**

## 17.CropDisease(DiseaseID, Name, Symptoms, Treatment)

Object Explorer    DDL file.sql\*    G2\_SARM/postgres@PostgreSQL 16

```
1 SELECT * FROM "G2SARM".CROPDISEASE;
```

Query History

diseased	name	symptoms	treatment
1	Downy Mildew	Yellowing of leaves, mold on underside.	Fungicides, good air circulation.
2	Powdery Mildew	White powdery spots on leaves.	Fungicides, sulfur-based treatments.
3	Bacterial Blight	Water-soaked spots on leaves.	Copper-based bactericides.
4	Leaf Spot	Brown or black spots on leaves.	Remove infected leaves, fungicides.
5	Rust	Orange or rust-colored spots on leaves.	Fungicides, crop rotation.
6	Blight	Wilting, brown stems, and leaves.	Fungicides, proper sanitation.
7	Crown Rot	Wilting and darkening of plant base.	Improve drainage, fungicides.
8	Fusarium Wilt	Yellowing and wilting of leaves.	Soil sterilization, resistant varieties.
9	Anthracnose	Dark, sunken lesions on fruits.	Fungicides, crop rotation.
10	Sclerotinia	White mold on infected tissues.	Fungicides, good air circulation.
11	Root Rot	Yellowing and wilting of plants.	Improve drainage, remove infected plants.
12	Powdery Scab	Powdery spots on tubers.	Crop rotation, resistant varieties.
13	Leaf Curl	Curling and distortion of leaves.	Pesticides, proper pruning.
14	Viruses	Stunted growth, mottled leaves.	Use virus-free seeds, insect control.
15	Gray Mold	Soft, brown rot on fruits and flowers.	Fungicides, reduce humidity.
16	Black Rot	Dark lesions on stems and leaves.	Remove infected plants, fungicides.
17	White Rust	White pustules on leaves.	Fungicides, resistant varieties.
18	Nematodes	Root galls, stunted growth.	Crop rotation, nematode-resistant varieties.
19	Flea Beetle	Tiny holes in leaves.	Insecticides, row covers.

Total rows: 90 of 90    Query complete 00:00:00.068    Successfully run. Total query runtime: 68 msec. 90 rows affected. Ln 1, Col 34

**Number of records: 90**

## 18.CropRotation(RotationID, CropSequence, Duration)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query Editor on the right. The Object Explorer lists various tables and sequences. The Query Editor displays the following SQL query and its results:

```
1 SELECT * FROM "G2SARM".CROPROTATION;
```

The results show 90 rows of crop rotation sequences, each consisting of a rotation ID, a sequence of crops, and a duration. The data is as follows:

rotationid	crosequence	duration
1	Corn - Soybean - Wheat	3
2	Wheat - Barley - Peas	3
3	Tomato - Cabbage - Lettuce	3
4	Potato - Corn - Beans	3
5	Rice - Soybean - Corn	3
6	Oats - Clover - Barley	3
7	Soybean - Wheat - Canola	3
8	Cotton - Peanuts - Sorghum	3
9	Pumpkin - Squash - Corn	3
10	Peas - Barley - Canola	3
11	Sugar Beet - Wheat - Peas	3
12	Corn - Peas - Oats	3
13	Cabbage - Carrot - Spinach	3
14	Onion - Garlic - Lettuce	3
15	Corn - Sunflower - Soybean	3
16	Carrot - Radish - Beet	3
17	Cucumber - Tomato - Lettuce	3
18	Broccoli - Cauliflower - Spinach	3
19	Millet - Sorghum - Cowpeas	3

Total rows: 90 of 90    Query complete 00:00:00.072    Ln 1, Col 36

**Number of records: 90**

## 19. Crop Benefits (BenefitID, RotationID, Benefit)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query Editor on the right. The Object Explorer lists various tables and sequences. The Query Editor displays the following SQL query and its results:

```
1 SELECT * FROM "G2SARM".CROPBENEFITS;
```

The results show 90 rows of crop benefits, each associated with a benefit ID, a rotation ID, and a detailed benefit description. The data is as follows:

benefitid	rotationid	benefit
1	1	1 Improves soil fertility through nitrogen fixation
2	2	1 Reduces pest populations
3	3	1 Enhances biodiversity
4	4	2 Increases yields by improving soil structure
5	5	2 Prevents disease buildup
6	6	3 Promotes healthy growth and increases yield
7	7	3 Improves soil structure and fertility
8	8	4 Diversifies crop production
9	9	4 Helps in weed management
10	10	5 Reduces dependency on chemical fertilizers
11	11	5 Improves pest management
12	12	6 Enhances soil health
13	13	6 Reduces erosion
14	14	7 Increases nutrient cycling
15	15	7 Improves water retention
16	16	8 Boosts overall farm resilience
17	17	8 Increases profits by diversifying crops
18	18	9 Improves nutrient availability
19	19	9 Reduces environmental impact

Total rows: 90 of 90    Query complete 00:00:00.139    Ln 1, Col 1

**Number of records: 90**

## 20. Investment (InvestmentID, Amount, Beneficiary, Date, PurposeID)

Object Explorer    DDL file.sql\*    G2\_SARM/postgres@PostgreSQL 16

Tables (41)

Query    Query History

```
1 SELECT * FROM "G2SARM".INVESTMENT;
```

Data Output    Messages    Notifications

investmentid [PK] integer	amount numeric (10,2)	beneficiary character varying (255)	date date	purposeid integer
1	1200.00	1	2024-01-20	1
2	2500.75	2	2024-02-22	2
3	1800.50	3	2024-03-15	3
4	3100.00	4	2024-04-12	4
5	450.00	5	2024-05-02	1
6	1350.90	6	2024-06-18	2
7	1900.30	7	2024-07-27	3
8	2500.10	8	2024-08-29	4
9	3200.40	9	2024-09-09	1
10	800.50	10	2024-10-05	2
11	1700.90	11	2024-11-15	3
12	2200.75	12	2024-12-12	4
13	1600.00	13	2024-01-30	1
14	2900.25	14	2024-02-18	2
15	3500.40	15	2024-03-05	3
16	1900.80	16	2024-04-07	4
17	2000.00	17	2024-05-25	1
18	1450.60	18	2024-06-22	2
19	2700.90	19	2024-07-30	3

Successfully run. Total query runtime: 72 msec. 90 rows affected. Ln 1, Col 34

Total rows: 90 of 90    Query complete 00:00:00.072

**Number of records: 90**

## 21. InvestmentPurpose(PurposeID, Purpose, Description)

Object Explorer    DDL file.sql\*    G2\_SARM/postgres@PostgreSQL 16

Tables (41)

Query    Query History

```
1 SELECT * FROM "G2SARM".INVESTMENTPURPOSE;
```

Data Output    Messages    Notifications

purposed [PK] integer	purpose character varying (255)	description text
1	Education	Funding for educational programs and resources.
2	Infrastructure	Investment in agricultural infrastructure improvements.
3	Research	Financial support for agricultural research initiatives.
4	Sustainability	Funding for projects aimed at sustainable farming practices.
5	Equipment	Investment in modern farming equipment and technology.
6	Training	Funding for training programs and workshops for farmers.
7	Community Development	Investing in community-driven agricultural projects.
8	Market Access	Supporting farmers in accessing larger markets.
9	Resource Management	Funding for water and soil conservation practices.
10	Policy Advocacy	Supporting initiatives for better agricultural policies.
11	Organic Farming	Investment in organic farming practices and certifications.
12	Research Facilities	Building and maintaining agricultural research facilities.
13	Extension Services	Funding for agricultural extension services and outreach.
14	Pest Management	Investment in integrated pest management strategies.
15	Seed Development	Funding for the development of improved seed varieties.
16	Technology Integration	Support for integrating new technologies in farming.
17	Biodiversity Projects	Funding initiatives that promote agricultural biodiversity.
18	Water Conservation	Investment in water-saving irrigation technologies.
19	Youth Programs	Funding programs aimed at engaging youth in agriculture.

Successfully run. Total query runtime: 93 msec. 90 rows affected. Ln 1, Col 41

Total rows: 90 of 90    Query complete 00:00:00.093

**Number of records: 90**

## 22. Weather(WeatherID, Date, Temperature, Precipitation, Conditions)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query Editor on the right. The Object Explorer lists various tables under the 'G2\_SARM/postgres@PostgreSQL 16' connection. The Query Editor contains the following SQL code:

```
1  SELECT * FROM "G2SARM".WEATHER;
```

The results pane displays the data from the WEATHER table:

weatherid	[PK] integer	date	temperature	precipitation	conditions
1	1	2024-01-01	25.00	0.00	Sunny
2	2	2024-01-02	22.50	0.00	Clear
3	3	2024-01-03	20.00	5.00	Rainy
4	4	2024-01-04	18.00	10.00	Cloudy
5	5	2024-01-05	21.00	2.00	Partly Cloudy
6	6	2024-01-06	24.00	0.00	Sunny
7	7	2024-01-07	23.50	0.50	Clear
8	8	2024-01-08	19.00	8.00	Rainy
9	9	2024-01-09	20.50	3.00	Cloudy
10	10	2024-01-10	22.00	0.00	Sunny
11	11	2024-01-11	26.00	0.00	Sunny
12	12	2024-01-12	27.50	0.00	Clear
13	13	2024-01-13	29.00	0.00	Clear
14	14	2024-01-14	24.00	1.00	Cloudy
15	15	2024-01-15	22.00	5.00	Rainy
16	16	2024-01-16	23.00	0.00	Sunny
17	17	2024-01-17	25.00	0.00	Sunny
18	18	2024-01-18	26.50	0.00	Clear
19	19	2024-01-19	27.00	0.00	Clear

Below the table, a message indicates: **Successfully run. Total query runtime: 80 msec. 85 rows affected.**

**Number of records: 85**

23. SustainablePractice(PracticeID, Name, Description, ImplementationLevel, BenefitID)

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query Editor on the right. The Object Explorer lists various tables under the 'G2\_SARM/postgres@PostgreSQL 16' connection. The Query Editor contains the following SQL code:

```
1  SELECT * FROM "G2SARM".SUSTAINABLEPRACTICE;
```

The results pane displays the data from the SUSTAINABLEPRACTICE table:

practiceid	[PK] integer	name	description	implementationlevel
1	1	Crop Rotation	The practice of alternating the types of crops grown on a particular piece of land to improve soil health and reduce pest pressure.	High
2	2	Cover Cropping	Planting crops that are not intended for harvest to cover the soil and improve soil quality.	Medium
3	3	Conservation Tillage	Reducing the amount of tillage to preserve soil structure and health.	Medium
4	4	Agroforestry	Integrating trees and shrubs into agricultural landscapes to enhance biodiversity and ecosystem services.	High
5	5	Integrated Pest Management (IPM)	Using a combination of biological, cultural, and chemical practices to manage pests in an environmentally friendly manner.	High
6	6	Organic Farming	A method of farming that avoids synthetic fertilizers and pesticides, focusing instead on natural processes.	High
7	7	Permaculture	Designing agricultural landscapes that mimic natural ecosystems to achieve sustainability.	Medium
8	8	Crop Diversification	Growing a variety of crops to enhance biodiversity and reduce risk.	Medium
9	9	Soil Conservation Techniques	Practices aimed at preventing soil erosion and maintaining soil health.	High
10	10	Water Harvesting	Collecting and storing rainwater for agricultural use.	Medium
11	11	Organic Fertilizers	Using natural materials to enhance soil fertility instead of synthetic fertilizers.	Medium
12	12	Mulching	Covering the soil with organic or inorganic materials to retain moisture and suppress weeds.	High
13	13	Composting	The process of recycling organic matter to create nutrient-rich soil amendments.	High
14	14	Sustainable Irrigation	Using methods that conserve water and reduce runoff while irrigating crops.	High
15	15	Green Manuring	Planting specific crops to be plowed back into the soil to enhance fertility.	Medium
16	16	Biological Control	Using natural predators or parasites to control pest populations.	Medium
17	17	Reduced Chemical Use	Minimizing the application of synthetic chemicals to protect the environment.	High
18	18	Agroecology	Applying ecological principles to agricultural systems to enhance sustainability.	High
19	19	Food Forests	Creating multi-layered forests that produce food while enhancing biodiversity.	Medium

Below the table, a message indicates: **Successfully run. Total query runtime: 68 msec. 90 rows affected.**

**Number of records: 90**

24. PracticeBenefits(BenefitID, PracticeID, Benefit)

```

SELECT * FROM "G2SARM".PRACTICEBENEFITS;

```

benefitid [PK] integer	practiceid [PK] integer	benefit text
1	1	Improves soil fertility and reduces pest infestations.
2	2	Enhances soil quality and prevents erosion.
3	3	Preserves soil structure and minimizes erosion.
4	4	Increases biodiversity and improves ecosystem services.
5	5	Reduces chemical use and promotes ecological balance.
6	6	Enhances soil health and reduces chemical dependency.
7	7	Promotes sustainable land use and biodiversity.
8	8	Increases resilience to pests and market fluctuations.
9	9	Protects soil from erosion and maintains fertility.
10	10	Conserves water and improves crop yield.
11	11	Improves soil structure and nutrient content.
12	12	Helps retain soil moisture and suppresses weeds.
13	13	Recycles nutrients and improves soil health.
14	14	Improves water efficiency and reduces waste.
15	15	Enhances soil fertility and prevents erosion.
16	16	Reduces pest populations naturally and sustainably.
17	17	Minimizes environmental harm and promotes health.
18	18	Enhances sustainability and ecosystem resilience.
19	19	Provides diverse food sources and habitat.

Total rows: 90 of 90    Query complete 00:00:00.146    Ln 1, Col 24

**Number of records: 90**

## 25. Farmer\_Crop(FarmerID, CropID)

```

SELECT * FROM "G2SARM".FARMER_CROP;

```

farmerid [PK] integer	cropid [PK] integer
1	1
2	1
3	2
4	1
5	3
6	2
7	4
8	5
9	6
10	7
11	8
12	9
13	10
14	11
15	12
16	13
17	14
18	15
19	16

Total rows: 90 of 90    Query complete 00:00:00.070    Ln 1, Col 35

**Number of records: 90**

## 26. Farmer\_Tech(FarmerID, TechnologyID, TimeofUsage)

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows the database schema with tables like farmer\_trainingprior, farmingtool, fertilizer, fertillertypetype, investment, investmentpurpose, irrigation, irrigation\_soil, irrigationtype, policy, polictype, practicebenefits, soil, solitype, sustainablepractice, technology, technologype, tooltype, trainingaudience, trainingprogram, visitor, weather, and various triggers, types, views, subscriptions, and roles.
- Query Editor:** Contains the SQL command: `SELECT * FROM "G2SARM".FARMER_TECH;`
- Data Output:** Displays the results of the query, which consists of 90 rows. The columns are `farmerid`, `technologyid`, and `timeofusage`.
- Status Bar:** Shows "Successfully run. Total query runtime: 114 msec. 90 rows affected." and "Ln 1, Col 36".

farmerid	technologyid	timeofusage
1	1	2023-01-15
2	1	2023-02-20
3	1	2023-03-10
4	2	2023-01-25
5	2	2023-04-15
6	2	2023-05-05
7	3	2023-06-30
8	3	2023-07-18
9	4	2023-08-22
10	4	2023-09-15
11	5	2023-10-20
12	5	2023-11-05
13	6	2023-01-30
14	6	2023-02-15
15	7	2023-03-28
16	7	2023-04-12
17	8	2023-05-23
18	8	2023-06-14
19	9	2023-07-30

**Number of records: 90**

## 27. FarmingTool(ToolID, Name, Usage, Manufacturer, FarmerID,TypeID)

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows the database schema with tables like farmer\_trainingprior, farmingtool, fertilizer, fertillertypetype, investment, investmentpurpose, irrigation, irrigation\_soil, irrigationtype, policy, polictype, practicebenefits, soil, solitype, sustainablepractice, technology, technologype, tooltype, trainingaudience, trainingprogram, visitor, weather, and various triggers, types, views, subscriptions, and roles.
- Query Editor:** Contains the SQL command: `SELECT * FROM "G2SARM".FARMINGTOOL;`
- Data Output:** Displays the results of the query, which consists of 90 rows. The columns are `toolid`, `name`, `usage`, `manufacturer`, `farmerid`, and `typeid`.
- Status Bar:** Shows "Successfully run. Total query runtime: 110 msec. 90 rows affected." and "Ln 1, Col 35".

toolid	name	usage	manufacturer	farmerid	typeid
1	Basic Hammer	Used for driving nails into wood	Craftsman	1	1
2	Cordless Drill	Used for drilling holes in various materials	DeWalt	2	2
3	Garden Spade	Used for digging and turning soil	Fiskars	1	3
4	Drip Irrigation Kit	Used for efficient watering of plants	Rain Bird	3	4
5	Soil pH Meter	Measures the acidity or alkalinity of the soil	Hanna Instruments	4	5
6	Seed Drill	Used for planting seeds in rows	John Deere	2	6
7	Fertilizer Spreader	Distributes fertilizers evenly across fields	Agri-Fab	5	7
8	Rotary Tiller	Tills the soil to prepare for planting	Husqvarna	6	8
9	Sickle	Used for harvesting crops	Ego	7	9
10	Weeder	Removes weeds from gardens and fields	Garden Weasel	8	10
11	Hand Cultivator	Breaks up and aerates the soil	Bosch	9	11
12	Backpack Sprayer	For applying pesticides and herbicides	Chapin	10	12
13	Tractor	Heavy machinery for various farming tasks	Case IH	1	13
14	Subsoiler	Used to break up compacted soil layers	Kinze	2	14
15	Mulching Mower	Cuts and chops grass, returning nutrients to the soil	Honda	3	15
16	Portable Greenhouse	Provides a controlled environment for plants	Palram	4	16
17	Irrigation Timer	Automates watering schedules	Orbit	5	17
18	Compost Bin	For composting organic materials	Envirocycle	6	18
19	Drone Sprayer	Drones used for crop spraying	DJI		

**Number of records: 90**

## 28. ToolType(TypeID, Type, Description)

```

SELECT * FROM "G2SARM".TOOLTYPE;

```

typeid	type	description
1	Hand Tools	Basic hand tools such as hammers, screwdrivers, and wrenches.
2	Power Tools	Electric or battery-powered tools for various tasks.
3	Gardening Tools	Tools specifically designed for gardening activities.
4	Irrigation Equipment	Tools used for efficient irrigation practices.
5	Soil Testing Kits	Kits for testing soil pH and nutrient levels.
6	Seeders	Devices used to sow seeds in the soil.
7	Fertilizer Spreaders	Tools for evenly spreading fertilizers on crops.
8	Plows	Equipment used for tilling soil and preparing fields.
9	Harvesting Tools	Tools designed for harvesting crops efficiently.
10	Weeding Tools	Tools for removing weeds from gardens and fields.
11	Cultivators	Tools used to break up soil and prepare seedbeds.
12	Pesticide Sprayers	Equipment for spraying pesticides on crops.
13	Tractors	Heavy machinery used for various farming tasks.
14	Tillage Equipment	Tools for soil preparation before planting.
15	Mulchers	Tools for shredding and spreading mulch.
16	Greenhouse Supplies	Materials and tools for greenhouse management.
17	Irrigation Controllers	Devices for automating irrigation systems.
18	Compost Bins	Containers for composting organic materials.
19	Crop Monitors	Tools for monitoring crop health and growth.

Total rows: 90 of 90    Query complete 00:00:00.093    Ln 1, Col 32

✓ Successfully run. Total query runtime: 93 msec. 90 rows affected. X

**Number of records: 90**

29.Irrigation(IrrigationID, CoverageArea, Efficiency, InstallationDate, FarmerID, TypeID)

```

SELECT * FROM "G2SARM".IRRIGATION;

```

irrigationid	coveragearea	efficiency	installationdate	farmerid	typeid
1	1	1000.00	85.00	2023-01-15	1
2	2	1500.50	90.00	2023-02-20	2
3	3	750.25	78.50	2023-03-05	3
4	4	1200.00	88.00	2023-04-10	4
5	5	900.75	82.00	2023-05-14	5
6	6	1100.00	80.00	2023-06-01	6
7	7	1600.20	95.00	2023-06-15	7
8	8	2000.00	91.50	2023-07-22	8
9	9	500.00	76.00	2023-08-03	9
10	10	800.00	83.50	2023-08-30	10
11	11	1300.50	89.00	2023-09-15	11
12	12	950.00	81.00	2023-10-10	12
13	13	1750.25	92.00	2023-11-01	13
14	14	2200.00	93.50	2023-11-05	14
15	15	1350.50	87.00	2023-12-12	15
16	16	1250.00	90.00	2023-01-11	16
17	17	800.50	79.50	2023-02-25	17
18	18	670.75	82.00	2023-03-15	18
19	19	980.00	84.50	2023-04-20	19

Total rows: 90 of 90    Query complete 00:00:00.070    Ln 1, Col 35

✓ Successfully run. Total query runtime: 70 msec. 90 rows affected. X

**Number of records: 90**

30.IrrigationType(TypeID, Type, Description)

```

SELECT * FROM "G2SARM".IRRIGATIONTYPE;

```

typeid [PK] integer	type character varying (255)	description text
1	1	Drip Irrigation
2	2	Sprinkler Irrigation
3	3	Surface Irrigation
4	4	Subsurface Irrigation
5	5	Flood Irrigation
6	6	Basin Irrigation
7	7	Furrow Irrigation
8	8	Center Pivot Irrigation
9	9	Line Source Irrigation
10	10	Surface Drip Irrigation
11	11	Spray Irrigation
12	12	Micro Irrigation
13	13	Portable Sprinkler System
14	14	Flooding with Water Harvesting
15	15	Overhead Irrigation
16	16	Manual Irrigation
17	17	Soaker Hose Irrigation
18	18	Rainwater Harvesting
19	19	Green Roof Irrigation

Total rows: 90 of 90    Query complete 00:00:00.086    ✓ Successfully run. Total query runtime: 86 msec. 90 rows affected.    Ln 1, Col 38

**Number of records: 90**

### 31. Farmer\_Soil(FarmerID, SoilID)

```

SELECT * FROM "G2SARM".FARMER_SOIL;

```

farmerid [PK] integer	soilid [PK] integer
1	1
2	1
3	2
4	2
5	3
6	3
7	4
8	5
9	5
10	6
11	7
12	8
13	9
14	10
15	11
16	12
17	13
18	14
19	15

Total rows: 90 of 90    Query complete 00:00:00.067    ✓ Successfully run. Total query runtime: 67 msec. 90 rows affected.    Ln 1, Col 35

**Number of records: 90**

### 32. Farmer\_Fertilizer(FarmerID, FertilizerID, Quantity)

```

Object Explorer   DDL file.sql* 
G2_SARM/postgres@PostgreSQL 16
Query History
1  SELECT * FROM "G2SARM".FARMER_FERTILIZER;
Data Output  Messages  Notifications
SQL
farmerid [PK] integer
fertilizerid [PK] integer
quantity numeric (10,2)
1 1 1 100.00
2 1 2 200.00
3 2 3 150.50
4 2 4 300.00
5 3 5 250.00
6 3 6 400.00
7 4 7 500.00
8 5 8 600.00
9 5 9 50.00
10 6 10 75.00
11 7 11 120.00
12 8 12 180.00
13 9 13 90.00
14 10 14 300.50
15 11 15 200.00
16 12 16 250.00
17 13 17 80.00
18 14 18 70.00
19 15 19 95.00

```

Total rows: 80 of 80 Query complete 00:00:00.115 ✓ Successfully run. Total query runtime: 115 msec. 80 rows affected. Ln 1, Col 46

**Number of records: 80**

### 33. Farmer\_TrainingProgram(FarmerID, ProgramID)

```

Object Explorer   DDL file.sql* 
G2_SARM/postgres@PostgreSQL 16
Query History
1  SELECT * FROM "G2SARM".FARMER_TRAININGPROGRAM;
Data Output  Messages  Notifications
SQL
farmerid [PK] integer
programid [PK] integer
1 1 1
2 1 2
3 2 3
4 2 4
5 3 5
6 3 6
7 4 7
8 5 8
9 5 9
10 6 10
11 7 11
12 8 12
13 9 13
14 10 14
15 11 15
16 12 16
17 13 17
18 14 18
19 15 19

```

Total rows: 80 of 80 Query complete 00:00:00.070 ✓ Successfully run. Total query runtime: 70 msec. 80 rows affected. Ln 1, Col 46

**Number of records: 80**

### 34. Crop\_Fertilizer(CropID, FertilizerID, FertilizerQuantity)

Object Explorer    DDL file.sql\* x

G2\_SARM/postgres@PostgreSQL 16

No limit ▾

Query History

```
1 SELECT * FROM "G2SARM".CROP_FERTILIZER;
```

Data Output Messages Notifications

SQL

	cropid	fertilizerid	fertilizerquantity
1	1	1	50.00
2	1	2	30.00
3	2	3	40.00
4	2	4	20.00
5	3	5	60.00
6	3	6	25.00
7	4	7	45.00
8	5	8	55.00
9	6	9	70.00
10	7	10	35.00
11	8	11	50.00
12	9	12	80.00
13	10	13	30.00
14	11	14	90.00
15	12	15	45.00
16	13	16	60.00
17	14	17	75.00
18	15	18	30.00
19	16	19	40.00

Total rows: 83 of 83    Query complete 00:00:00.066 ✓ Ln 1, Col 39

**Number of records: 83**

### 35.Crop\_Disease(CropID, DiseaseID)

Object Explorer    DDL file.sql\* x

G2\_SARM/postgres@PostgreSQL 16

No limit ▾

Query History

```
1 SELECT * FROM "G2SARM".CROP_DISEASE;
```

Data Output Messages Notifications

SQL

	cropid	diseaseid
1	1	1
2	1	2
3	1	3
4	2	2
5	2	4
6	2	5
7	3	1
8	3	6
9	3	7
10	4	2
11	4	8
12	5	3
13	5	9
14	6	10
15	6	11
16	7	4
17	7	12
18	8	5
19	9	1

Total rows: 92 of 92    Query complete 00:00:00.077 ✓ Ln 1, Col 37

**Number of records: 92**

### 36.Farmer\_Aid(FarmerID, AidID)

```

Object Explorer   DDL file.sql*   G2_SARM/postgres@PostgreSQL 16
farmer_trainingpr...
farmingtool
fertilizer
fertilizertype
investment
investmentpurpose
irrigation
irrigation_soil
irrigationtype
policy
policytype
practicebenefits
soil
solitype
sustainablepractices
technology
technologytype
tooltypes
trainingaudience
trainingprogram
visitor
weather
Trigger Functions
Types
Views
Subscriptions
postgres
Login/Group Roles
Tablespaces
pgAgent Jobs

Query  Query History
1  SELECT * FROM "G2SARM".FARMER_AID;

Data Output  Messages  Notifications
farmerid [PK] integer | aidid [PK] integer |
1 | 1 |
2 | 1 |
3 | 2 |
4 | 2 |
5 | 3 |
6 | 3 |
7 | 4 |
8 | 5 |
9 | 5 |
10 | 6 |
11 | 6 |
12 | 7 |
13 | 8 |
14 | 8 |
15 | 9 |
16 | 10 |
17 | 10 |
18 | 11 |
19 | 12 |

Total rows: 90 of 90  Query complete 00:00:00.161  ✓ Successfully run. Total query runtime: 161 msec. 90 rows affected.  Ln 1, Col 34

```

**Number of records: 90**

### 37.Farmer\_Investment(FarmerID, InvestmentID)

```

Object Explorer   DDL file.sql*   G2_SARM/postgres@PostgreSQL 16
farmer_trainingpr...
farmingtool
fertilizer
fertilizertype
investment
investmentpurpose
irrigation
irrigation_soil
irrigationtype
policy
policytype
practicebenefits
soil
solitype
sustainablepractices
technology
technologytype
tooltypes
trainingaudience
trainingprogram
visitor
weather
Trigger Functions
Types
Views
Subscriptions
postgres
Login/Group Roles
Tablespaces
pgAgent Jobs

Query  Query History
1  SELECT * FROM "G2SARM".FARMER_INVESTMENT;

Data Output  Messages  Notifications
farmerid [PK] integer | investmentid [PK] integer |
1 | 1 |
2 | 1 |
3 | 2 |
4 | 2 |
5 | 3 |
6 | 3 |
7 | 4 |
8 | 5 |
9 | 5 |
10 | 6 |
11 | 6 |
12 | 7 |
13 | 8 |
14 | 8 |
15 | 9 |
16 | 10 |
17 | 10 |
18 | 11 |
19 | 12 |

Total rows: 90 of 90  Query complete 00:00:00.065  ✓ Successfully run. Total query runtime: 65 msec. 90 rows affected.  Ln 1, Col 42

```

**Number of records: 90**

### 38.Irrigation\_Soil(IrrigationID, SoilID)

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying various database objects like tables, functions, and roles. The right pane is a query editor window titled 'DDL file.sql' with the connection 'G2\_SARM/postgres@PostgreSQL 16'. A query is running:

```
1  SELECT * FROM "G2SARM".IRRIGATION_SOIL;
```

The results pane shows the data from the 'IRRIGATION\_SOIL' table:

	irrigationid	solidid
1	1	1
2	1	2
3	2	3
4	2	4
5	3	5
6	3	6
7	4	7
8	5	8
9	5	9
10	6	10
11	6	11
12	7	12
13	8	13
14	8	14
15	9	15
16	10	16
17	10	17
18	11	18
19	12	19

At the bottom of the results pane, a green message box says: 'Successfully run. Total query runtime: 68 msec. 90 rows affected.' and 'Ln 1, Col 39'.

**Number of records: 90**

### 39.Crop\_Soil(CropID, SoilID)

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying various database objects. The right pane is a query editor window titled 'DDL file.sql' with the connection 'G2\_SARM/postgres@PostgreSQL 16'. A query is running:

```
1  SELECT * FROM "G2SARM".CROP_SOIL;
```

The results pane shows the data from the 'CROP\_SOIL' table:

	cropid	solidid
1	1	1
2	1	2
3	2	3
4	2	4
5	3	5
6	3	6
7	4	7
8	5	8
9	5	9
10	6	10
11	6	11
12	7	12
13	8	13
14	8	14
15	9	15
16	10	16
17	10	17
18	11	18
19	12	19

At the bottom of the results pane, a green message box says: 'Total rows: 90 of 90' and 'Query complete 00:00:00.089'.

**Number of records: 90**

### 40.Crop\_Weather(CropID, WeatherID)

```

SELECT * FROM "G2SARM".CROP_WEATHER;

```

	cropid	weatherid
1	1	1
2	1	2
3	1	3
4	2	4
5	2	5
6	2	6
7	3	7
8	3	8
9	3	9
10	4	10
11	4	11
12	4	12
13	5	13
14	5	14
15	5	15
16	6	16
17	6	17
18	7	18
19	7	19

Total rows: 90 of 90    Query complete 00:00:00.070    ✓ Successfully run. Total query runtime: 70 msec. 90 rows affected.    Ln 1, Col 37

**Number of records: 90**

#### 41. Farmer\_SustainablePractice(FarmerID, PracticeID)

```

SELECT * FROM "G2SARM".FARMER_SUSTAINABLEPRACTICE;

```

	farmerid	practiceid
1	1	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	6
7	6	7
8	7	8
9	8	9
10	9	10
11	10	11
12	11	12
13	12	13
14	13	14
15	14	15
16	15	16
17	16	17
18	17	18
19	18	19

Total rows: 90 of 90    Query complete 00:00:00.119    ✓ Successfully run. Total query runtime: 119 msec. 90 rows affected.    Ln 1, Col 50

**Number of records: 90**