tables

**user** = admin, farmer, company, service\_provider {Id Name/company name contact email(optional) address password typeOfUser}

**property** = farmers can add the property details in it {property id , farmerid, location, type of land(bagayti or jirayati), img of land, img of 7/12 , area in acre, area in guntha}

**application** = company can apply for the property {application id, company id, property id, status(accepted or not)}

**service** = service provider can provide the service {service id, serviceprovider id, name , service name , price, description}

**service** \_applications = provide the applications for the service {application id, service id, company id, status(accpetd/reject), iscomplete}

**contact** = all the contact us informations {contact id, name, email(optional), Description}

**payment** = 1000 registration fee up to register the farm with the farmer {payment id, service id/property id, payment method, payment mode, payment Status}

**governmentSchems** = admin adds the farming related schems on the portal {scheme id, title, price, start date, last date, description}

Table Structure:

1. User Table (Stores all types of users)

Columns:

* id
* name
* contact
* email
* address
* password
* type\_of\_user

2. Property Table (Stores farmer's land details)

Columns:

* property\_id
* farmer\_id
* location
* type\_of\_land
* land\_image
* document\_image
* area\_acre
* area\_guntha

3. Application Table (Tracks company applications for properties)

Columns:

* application\_id
* company\_id
* property\_id
* status

4. Service Table (Stores service provider details)

Columns:

* service\_id
* service\_provider\_id
* service\_name
* price
* description

5. Service Applications Table (Tracks company applications for services)

Columns:

* application\_id
* service\_id
* company\_id
* status
* is\_complete

6. Contact Table (Stores "Contact Us" messages)

Columns:

* contact\_id
* name
* email
* description

7. Payment Table (Stores registration fees for farmers)

Columns:

* payment\_id
* service\_id / property\_id
* payment\_method
* payment\_mode
* payment\_status

8. Government Schemes Table (Stores farming-related schemes added by admin)

Columns:

* scheme\_id
* title
* price
* start\_date
* last\_date
* description

**Table Relation:**

1. **User = Property (One-to-Many):** One farmer can add multiple properties.
2. **User = Application (One-to-Many):** One company can apply for multiple properties.
3. **User = Service (One-to-Many):** One service provider can offer multiple services.
4. **User = Service Applications (One-to-Many):** One company can apply for multiple services.
5. **Property = Application (One-to-Many):** One property can receive multiple applications.
6. **Property = Payment (One-to-One):** Each property has one registration payment.
7. **Service = Service Applications (One-to-Many):** One service can have multiple applications.
8. **Service = Payment (One-to-One):** A service payment can be tracked.
9. **Application = (User, Property) (Many-to-One):** A company applies for a property.
10. **Service Applications = (User, Service) (Many-to-One):** A company applies for a service.
11. **Government Schemes:** Independent (Managed by Admin).

**Creating Tables**

**Creating User Table**

CREATE TABLE finaldb.user (

Id INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Contact VARCHAR(20) NOT NULL UNIQUE,

Email VARCHAR(255) UNIQUE,

Address TEXT NOT NULL,

Password VARCHAR(255) NOT NULL,

TypeOfUser ENUM('admin', 'farmer', 'company', 'serviceprovider') NOT NULL

);

**Create the table property:**

CREATE TABLE finaldb.property (

PropertyId INT PRIMARY KEY AUTO\_INCREMENT,

FarmerId INT NOT NULL,

Location TEXT NOT NULL,

TypeOfLand ENUM('bagayti', 'jirayati') NOT NULL,

ImgOfLand VARCHAR(255) NOT NULL,

ImgOf7\_12 VARCHAR(255) NOT NULL,

AreaInAcre FLOAT NOT NULL CHECK (AreaInAcre > 0),

AreaInGuntha FLOAT NOT NULL CHECK (AreaInGuntha > 0),

FOREIGN KEY (FarmerId) REFERENCES finaldb.user(Id) ON DELETE CASCADE

);

**Create table for services**

CREATE TABLE finaldb.service (

ServiceId INT PRIMARY KEY AUTO\_INCREMENT,

ServiceProviderId INT NOT NULL,

Name VARCHAR(255) NOT NULL,

ServiceName VARCHAR(255) NOT NULL,

Price DECIMAL(10,2) NOT NULL CHECK (Price >= 0),

Description TEXT NOT NULL,

FOREIGN KEY (ServiceProviderId) REFERENCES finaldb.user(Id) ON DELETE CASCADE

);

**Create the table for applications of property:**

CREATE TABLE finaldb.application (

ApplicationId INT PRIMARY KEY AUTO\_INCREMENT,

CompanyId INT NOT NULL,

PropertyId INT NOT NULL,

Status ENUM('accepted', 'rejected', 'pending') NOT NULL DEFAULT 'pending',

FOREIGN KEY (CompanyId) REFERENCES finaldb.user(Id) ON DELETE CASCADE,

FOREIGN KEY (PropertyId) REFERENCES finaldb.property(PropertyId) ON DELETE CASCADE

);

**Create the table for application of services:**

CREATE TABLE finaldb.service\_applications (

ApplicationId INT PRIMARY KEY AUTO\_INCREMENT,

ServiceId INT NOT NULL,

CompanyId INT NOT NULL,

Status ENUM('accepted', 'rejected', 'pending') NOT NULL DEFAULT 'pending',

IsComplete BOOLEAN NOT NULL DEFAULT FALSE,

FOREIGN KEY (ServiceId) REFERENCES finaldb.service(ServiceId) ON DELETE CASCADE,

FOREIGN KEY (CompanyId) REFERENCES finaldb.user(Id) ON DELETE CASCADE

);

**Create table for the contact us**

CREATE TABLE finaldb.contact (

ContactId INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Email VARCHAR(255) UNIQUE,

Description TEXT NOT NULL

);

**Create table for payment details:**

CREATE TABLE finaldb.payment (

PaymentId INT PRIMARY KEY AUTO\_INCREMENT,

ServiceOrPropertyId INT NOT NULL,

PaymentMethod VARCHAR(50) NOT NULL,

PaymentMode VARCHAR(50) NOT NULL,

PaymentStatus ENUM('paid', 'pending', 'failed') NOT NULL DEFAULT 'pending'

);

**Create table for the government Schemas:**

CREATE TABLE finaldb.governmentSchemes (

SchemeId INT PRIMARY KEY AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL UNIQUE,

Price DECIMAL(10,2) NOT NULL CHECK (Price >= 0),

StartDate DATE NOT NULL,

LastDate DATE NOT NULL,

Description TEXT NOT NULL

);

**Inserting Sample Data**

**Inserting data in User Table:**

INSERT INTO user (Name, Contact, Email, Address, Password, TypeOfUser) VALUES

('Admin User', '1234567890', 'admin@example.com', 'Admin Address', 'adminpass', 'admin'),

('Farmer1', '9876543210', 'farmer1@example.com', 'Farmer Address', 'farmerpass', 'farmer'),

('Company1', '1122334455', 'company1@example.com', 'Company Address', 'companypass', 'company'),

('ServiceProvider1', '6677889900', 'sp1@example.com', 'SP Address', 'spass', 'serviceprovider');

**Insert data in property table:**

INSERT INTO property (FarmerId, Location, TypeOfLand, ImgOfLand, ImgOf7\_12, AreaInAcre, AreaInGuntha) VALUES

(2, 'Village A', 'bagayti', 'land1.jpg', 'doc1.jpg', 5, 10);

**Insert data in application table:**

INSERT INTO application (CompanyId, PropertyId, Status) VALUES

(3, 1, 'pending');

**Insert data in service table:**

INSERT INTO service (ServiceProviderId, Name, ServiceName, Price, Description) VALUES

(4, 'John Doe', 'Tractor Rental', 500.00, 'Rent a tractor for farming');

**Insert data in service\_application table:**

INSERT INTO service\_applications (ServiceId, CompanyId, Status, IsComplete) VALUES

(1, 3, 'pending', FALSE);

**Insert data in contact table:**

INSERT INTO contact (Name, Email, Description) VALUES

('User1', 'user1@example.com', 'Inquiry about services');

**Insert data in payment table:**

INSERT INTO payment (ServiceOrPropertyId, PaymentMethod, PaymentMode, PaymentStatus) VALUES

(1, 'UPI', 'Online', 'paid');

**Insert data in government Schemas table:**

INSERT INTO governmentSchemes (Title, Price, StartDate, LastDate, Description) VALUES

('Organic Farming Subsidy', 20000.00, '2025-01-01', '2025-12-31', 'Subsidy for organic farming.');

1.User and government

1. Fetching Farmers Who Have Not Applied for Any Scheme
2. Fetching Farmers Who Have Applied for at Least One Scheme
3. Fetching All Schemes and the Farmers Who Applied
4. Fetching Farmers Along with the Total Number of Schemes They Applied For
5. Fetching Farmers Who Applied for the Most Schemes
6. Fetching Farmers Who Applied for the Least Number of Schemes
7. Fetching the Latest Scheme Applied by Each Farmer
8. Fetching the Earliest Scheme Applied by Each Farmer
9. Fetching Farmers Who Applied for Schemes Within a Given Date Range
10. Fetching Farmers Who Applied for a Scheme But Have Not Used Any Services
11. Fetching Farmers Who Applied for a Scheme and Also Used Services
12. Fetching the Total Amount of Money Distributed Through Schemes for Each Farmer
13. Fetching Farmers Who Applied for Expired Schemes
14. Fetching Farmers Who Applied for Active Schemes
15. Fetching Farmers and Their Corresponding Scheme Details in a Single View
16. Fetching Farmers Who Applied for Multiple Schemes of the Same Price
17. Fetching Farmers Who Applied for a Scheme but Never Paid for Any Service
18. Fetching Farmers Who Applied for Government Schemes More Than a Certain Number of Times
19. Fetching Farmers Who Never Applied for Schemes and Never Used Services
20. Fetching Farmers Who Applied for a Scheme in the Last Year but Not in the Current Year

2.User and property

1. Fetching Farmers Who Have Not Registered Any Land
2. Fetching Farmers Who Have Registered at Least One Land Property
3. Fetching All Properties and Their Corresponding Farmers
4. Fetching Farmers Along with the Total Number of Properties They Own
5. Fetching Farmers Who Own the Largest Land Area
6. Fetching Farmers Who Own the Smallest Land Area
7. Fetching Farmers and Their Total Land Area in Acres and Gunthas
8. Fetching Properties Located in a Specific Region
9. Fetching Properties Based on Land Type (Bagayti or Jirayati)
10. Fetching Farmers Who Have Registered More Than a Certain Number of Properties
11. Fetching Farmers Who Have Registered Land But Have Never Leased It
12. Fetching Farmers Who Have Both Leased and Not Leased Land
13. Fetching the Most Common Land Type Among Farmers
14. Fetching Farmers Who Own Multiple Lands of Different Types
15. Fetching Farmers and Their Land Documents
16. Fetching Farmers Who Own Properties Larger Than a Given Size
17. Fetching Farmers Who Registered Land in the Last Year
18. Fetching Farmers Who Have Not Updated Their Land Details for a Long Time
19. Fetching Farmers Whose Land Size Is Below the Average Farm Size
20. Fetching Farmers Who Registered Properties But Have No Agreements on Them

3.User and Service

1. Fetching Service Providers and Their Services
2. Fetching Service Providers Who Have Not Listed Any Services
3. Fetching All Services and Their Corresponding Service Providers
4. Fetching Service Providers Along with the Total Number of Services They Offer
5. Fetching Service Providers Who Offer the Most Services
6. Fetching Service Providers Who Offer Only One Service
7. Fetching the Most Expensive Services and Their Providers
8. Fetching the Cheapest Services and Their Providers
9. Fetching Service Providers and Their Services Sorted by Price
10. Fetching Service Providers Who Have Never Been Hired for a Service
11. Fetching Service Providers Who Have Completed At Least One Service Agreement
12. Fetching Service Providers Who Have the Most Completed Service Agreements
13. Fetching Service Providers Who Have the Most Pending Service Agreements
14. Fetching Service Providers Whose Services Are Frequently Used by Farmers
15. Fetching the Total Revenue Generated by Each Service Provider
16. Fetching the Average Price of Services Offered by Each Provider
17. Fetching Services That Have Never Been Used in an Agreement
18. Fetching Service Providers Who Have Not Updated Their Services for a Long Time
19. Fetching Service Providers and Their Services Along with Descriptions
20. Fetching Farmers Who Have Used Services from More Than One Provider

4.Agreement and user

1. Fetching Users Who Have Never Made an Agreement
2. Fetching Users Who Have Made at Least One Agreement
3. Fetching All Agreements Along with User Details
4. Fetching Users and the Total Number of Agreements They Have
5. Fetching Users Who Have the Most Agreements
6. Fetching Users Who Have Only One Agreement
7. Fetching Users Who Have Pending Agreements
8. Fetching Users Who Have Rejected Agreements
9. Fetching Users Who Have Completed Agreements
10. Fetching Users Who Have Both Accepted and Rejected Agreements
11. Fetching Users and Their Latest Agreement
12. Fetching Users Who Have Service Agreements Only
13. Fetching Users Who Have Property Agreements Only
14. Fetching Users Who Have Both Service and Property Agreements
15. Fetching Users and Their Agreement Status Counts (Pending, Accepted, Rejected, Completed)
16. Fetching Users Who Have Active Agreements (Ongoing Services or Leased Properties)
17. Fetching Users Who Have Agreements That Lasted the Longest
18. Fetching Users and Their Agreements Along with Start and End Dates
19. Fetching Users Who Have Not Had Any Agreements in the Last Year
20. Fetching Users Who Have the Shortest Agreement Duration

**5.Agreement and payment**

1. Fetching Agreements That Have No Payments Made Yet
2. Fetching Agreements That Have at Least One Payment
3. Fetching Agreements and Their Payment Details
4. Fetching Agreements with Pending Payments
5. Fetching Agreements with Completed Payments
6. Fetching Agreements with Failed Payments
7. Fetching Agreements Along with Their Total Payment Amount
8. Fetching Agreements with Multiple Payments
9. Fetching Agreements That Were Paid Offline
10. Fetching Agreements That Were Paid Online
11. Fetching Agreements and Their Payment Methods (UPI, Credit Card, etc.)
12. Fetching Agreements with the Most Payments Made
13. Fetching Agreements Where Partial Payments Have Been Made
14. Fetching Agreements That Were Paid in Full in a Single Transaction
15. Fetching Payments and Their Associated Agreement Status (Pending, Completed, etc.)
16. Fetching the Total Amount Paid for Each Agreement
17. Fetching Agreements Where Payment Was Made But the Agreement Was Rejected
18. Fetching Agreements Where Payments Were Made After the Agreement Ended
19. Fetching Agreements That Had Refunds or Reversed Payments
20. Fetching Agreements Sorted by Payment Date and Status

**6.Agreement and service**

1. Fetching Agreements That Are Related to Services
2. Fetching Services That Have Never Been Involved in an Agreement
3. Fetching Agreements Along with Their Associated Services
4. Fetching Services and Their Total Number of Agreements
5. Fetching Agreements for Services That Are Still Pending
6. Fetching Agreements for Services That Have Been Completed
7. Fetching Agreements for Services That Were Rejected
8. Fetching Services That Have the Most Agreements
9. Fetching Services That Have the Least Agreements
10. Fetching Agreements and Their Associated Service Providers
11. Fetching Agreements for Services Sorted by Start Date
12. Fetching Agreements Where Services Were Provided but No Payment Was Made
13. Fetching Services That Were Provided for the Longest Duration in Agreements
14. Fetching Services That Had the Shortest Agreement Duration
15. Fetching Services That Have Been Frequently Used in Agreements
16. Fetching Agreements Where Multiple Services Were Provided
17. Fetching Agreements for Services That Were Canceled Before Completion
18. Fetching Agreements Where the Service Provider Has the Most Completed Agreements
19. Fetching Agreements That Started But Are Still in Progress
20. Fetching Services That Were Included in Agreements in the Last Month/Year

**7.Agreement and property**

1. Fetching Agreements That Are Related to Properties
2. Fetching Properties That Have Never Been Involved in an Agreement
3. Fetching Agreements Along with Their Associated Properties
4. Fetching Properties and Their Total Number of Agreements
5. Fetching Agreements for Properties That Are Still Pending
6. Fetching Agreements for Properties That Have Been Completed
7. Fetching Agreements for Properties That Were Rejected
8. Fetching Properties That Have the Most Agreements
9. Fetching Properties That Have the Least Agreements
10. Fetching Agreements and Their Associated Farmers (Land Owners)
11. Fetching Agreements for Properties Sorted by Start Date
12. Fetching Agreements Where Properties Were Leased But No Payment Was Made
13. Fetching Properties That Were Leased for the Longest Duration in Agreements
14. Fetching Properties That Had the Shortest Lease Agreement
15. Fetching Properties That Have Been Frequently Used in Agreements
16. Fetching Agreements Where Multiple Properties Were Leased Together
17. Fetching Agreements for Properties That Were Canceled Before Completion
18. Fetching Agreements Where the Farmer Has the Most Completed Agreements
19. Fetching Agreements That Started But Are Still in Progress
20. Fetching Properties That Were Included in Agreements in the Last Month/Year

**1. User-Based Queries**

* Fetch all users and their types (Farmer, Company, Service Provider, Admin).
* Fetch farmers who have registered land properties.
* Fetch service providers who have listed services.
* Fetch users who have made agreements (services or properties).
* Fetch farmers who have applied for government schemes.
* Fetch users who have pending agreements.
* Fetch farmers who have received services.
* Fetch companies that have partnered with farmers.

**2. Property-Based Queries**

* Fetch all registered properties and their farmers.
* Fetch properties based on type (Bagayti, Jirayati).
* Fetch properties that have been leased at least once.
* Fetch properties that have never been leased.
* Fetch properties with the largest and smallest land areas.
* Fetch farmers with multiple properties.
* Fetch properties available for lease.
* Fetch properties with active lease agreements.
* Fetch properties where agreements were canceled.

**3. Service-Based Queries**

* Fetch all services and their providers.
* Fetch services that have been booked through agreements.
* Fetch services that have never been used.
* Fetch most frequently booked services.
* Fetch least booked services.
* Fetch service providers with the highest number of completed agreements.
* Fetch services sorted by price (highest to lowest).
* Fetch services that are pending completion.

**4. Agreement-Based Queries**

* Fetch all agreements and their types (Property or Service).
* Fetch agreements that are pending, accepted, completed, or rejected.
* Fetch agreements that lasted the longest.
* Fetch agreements that had payment failures.
* Fetch agreements where payments were made offline.
* Fetch agreements where payments were made online.
* Fetch agreements where services were completed but not paid for.
* Fetch users who have the most agreements.
* Fetch agreements sorted by start and end dates.

**5. Payment-Based Queries**

* Fetch all payments and their statuses (Paid, Pending, Failed).
* Fetch agreements with successful payments.
* Fetch agreements where payments are still pending.
* Fetch agreements where payments were completed in multiple transactions.
* Fetch total amount collected from agreements.
* Fetch agreements where refunds were issued.
* Fetch most used payment methods (UPI, Credit Card, etc.).
* Fetch payments sorted by transaction date.

**6. Government Schemes-Based Queries**

* Fetch all government schemes available for farmers.
* Fetch schemes applied by farmers.
* Fetch farmers who have never applied for any scheme.
* Fetch schemes that are currently active.
* Fetch expired government schemes.
* Fetch farmers who have benefited from multiple schemes.

**7. Analytical & Business Insight Queries**

* Fetch total revenue generated from services and property leases.
* Fetch the most common land type among farmers.
* Fetch the average price of services.
* Fetch the top-performing service providers.
* Fetch the most active farmers in agreements.
* Fetch the least active farmers.
* Fetch service providers or companies with high engagement.
* Fetch most common reasons for agreement rejection.
* Fetch total number of agreements per month/year.
* Fetch number of pending agreements per category (Property/Service).