**1. User Table (Stores all types of users)**

* **id** (Primary Key)
* name / company\_name
* contact
* email
* address
* password
* type\_of\_user ('admin', 'farmer', 'company', 'service\_provider')
* Create\_date

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

* **property\_id** (Primary Key)
* **farmer\_id** (Foreign Key → User(id))
* location
* type\_of\_land (**Irrigated Land**, Rainfed Land)
* land\_image
* document\_image
* area\_acre
* area\_guntha
* Create\_date

**3. Service Table (Stores service provider details)**

* **service\_id** (Primary Key)
* **service\_provider\_id** (Foreign Key → User(id))
* service\_name
* price
* duration (per acre)
* description
* Create\_date

**4. Agreement Table (Tracks agreements for services and properties)**

* **agreement\_id** (Primary Key)
* **user\_id** (Foreign Key → User(id))
* type ('service', 'property')
* **reference\_id** (Foreign Key → Service(service\_id) OR Property(property\_id))
* status ('accepted', 'pending', 'rejected', 'completed')
* start\_date
* end\_date
* time\_duration (If service → work duration, If property → lease duration)
* create\_date

**5. Payment Table**

* **payment\_id** (Primary Key)
* **agreement\_id** (Foreign Key → Agreement(agreement\_id))
* **amount\_paid (** Total amount paid)
* payment\_method ('UPI', 'Credit Card', 'Debit Card', 'Net Banking')
* payment\_mode ('Online', 'Offline')
* payment\_status ('Paid', 'Pending', 'Failed')
* **transaction\_id**
* Recived\_date

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

* **scheme\_id** (Primary Key)
* **farmer\_id** (Foreign Key → User(id))
* title
* price
* start\_date
* last\_date
* description
* Created\_date

7. Government Schemes Application

• application\_id

• scheme\_id (Primary Key)

• farmer\_id (Foreign Key → User(id))

• register\_date

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 (Irrigated Land, Rainfed Land).
* 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.

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

**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).

Got it! You need **at least 5-6 important queries per table**, combining two tables each time. Here's a structured list:

**1. User & Property (Farmers and Their Lands)**

**Farmers Without Registered Land**  
**Farmers Who Have Registered at Least One Property**  
**Farmers and Their Total Land Holdings (Acres & Gunthas)**  
**Farmers Who Own the Largest and Smallest Land Area**  
**Properties Filtered by Location and Land Type**  
**Farmers Who Registered Land in the Last Year**

**2. User & Service (Service Providers and Their Services)**

**Service Providers and Their Offered Services**  
**Services Filtered by Price Range and Duration**  
**Most Frequently Availed Services**  
**Service Providers Who Have Not Provided Any Service Yet**  
 **Service Providers Who Have Set the Highest and Lowest Service Prices**

**3. User & Agreement (Tracking Property and Service Agreements)**

**Active and Pending Agreements for Properties and Services**  
**Farmers Who Have Leased Land But Have Never Taken Any Services**  
**Farmers Who Have Both Leased and Non-Leased Properties**  
**Agreements Sorted by Status (Accepted, Pending, Rejected, Completed)**  
**Users Who Have the Most Number of Agreements (Farmers, Companies, or Providers)**

**4. Agreement & Payment (Tracking Payments for Agreements)**

**Payments Sorted by Status (Paid, Pending, Failed)**  
**Farmers Who Have Made Partial Payments**  
**Service Providers Who Have Not Received Payments Yet**  
**Agreements Without Any Payment Made**  
**Payments Filtered by Payment Method (UPI, Credit Card, Debit Card, etc.)**

**5. User & Government Schemes (Farming-Related Schemes for Farmers)**

**Farmers Who Have Applied for Government Schemes**  
**Farmers Who Have Not Applied for Any Schemes Yet**  
**Schemes Filtered by Active and Expired Status**  
**Schemes with the Highest Number of Beneficiaries**  
**Farmers Who Have Applied for Multiple Schemes**

**6. Property & Agreement (Land Lease and Usage Statistics)**

**Who Have Registered Properties But Have No Agreements on Them**  
**Properties That Have Been Leased for the Longest Duration**  
**Properties That Have Never Been Leased Yet**  
**Properties With Multiple Lease Agreements Over Time**  
**Farmers Who Own Properties Larger Than a Given Size**  
**Farmers Whose Land Size Is Below the Average Farm Size**

**Final Overview**

**Each table has 5-6 key queries**  
**Covers all necessary use cases for your project frontend**  
**Queries include conditions, filters, and aggregations**

Let me know if this works for you or if you need any more adjustments! 🚀

**Tables**

**1. User Table (Stores all types of users)**

* **id** (Primary Key)
* name / company\_name
* contact
* email
* address
* password
* type\_of\_user ('admin', 'farmer', 'company', 'service\_provider')
* Create\_date

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

* **property\_id** (Primary Key)
* **farmer\_id** (Foreign Key → User(id))
* location
* type\_of\_land (**Irrigated Land**, Rainfed Land)
* land\_image
* document\_image
* area\_acre
* area\_guntha
* Create\_date

**3. Service\_provider Table (Stores service provider details)**

* **id** (Primary Key)
* service\_id ( foreign key -> (services)service\_Id)
* **service\_provider\_id** (Foreign Key -> User(id))
* price
* duration (per acre)
* description
* Create\_date

**4. Agreement Table (Tracks agreements for services and properties)**

* **agreement\_id** (Primary Key)
* **user\_id** (Foreign Key → User(id))
* type ('service', 'property')
* **reference\_id** (Foreign Key → Service(service\_id) OR Property(property\_id))
* status ('accepted', 'pending', 'rejected', 'completed')
* start\_date
* end\_date
* time\_duration (If service → work duration, If property → lease duration)
* create\_date

**5. Payment Table**

* **payment\_id** (Primary Key)
* **agreement\_id** (Foreign Key → Agreement(agreement\_id))
* **amount\_paid (** Total amount paid)
* payment\_method ('UPI', 'Credit Card', 'Debit Card', 'Net Banking')
* payment\_mode ('Online', 'Offline')
* payment\_status ('Paid', 'Pending', 'Failed')
* **transaction\_id**
* Recived\_date

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

* **scheme\_id** (Primary Key)
* title
* price
* start\_date
* last\_date
* description
* Created\_date

7. Government Schemes Application

• application\_id

• scheme\_id (Primary Key)

• farmer\_id (Foreign Key → User(id))

• register\_date

* Status (Pending,Accepted,Rejected)

8. Services (stores all the services)

* Service\_id (Primary Key)
* Service\_name

**Fetch Queries**

1. **Fetch farmers who have applied for multiple government schemes**

SELECT users.id as farmarId, users.name as farmerName, COUNT(s.scheme\_id) AS totalSchemes,

ARRAY\_AGG(g.title) AS schemeNames

FROM users users

JOIN scheme\_applications s ON users.id = s.farmer\_id

JOIN government\_schemes g ON s.scheme\_id = g.scheme\_id

GROUP BY users.id, users.name;

**Output :**

[

{

“farmarId”:3,

“farmerName”:”Sneha Patil”,

“totalSchemes” : 2,

“schemeNames”:[ Soil Health Card Scheme", "Fasal Bima Yojana”]

},

{

“farmarId”:1,

“farmerName”: “Rajesh Kulkarni”,

“totalSchemes” : 2,

“schemeNames”:[ “PM Kisan Samman Nidhi", "National Horticulture Mission","Agri Infra Fund”]

},

{

“farmarId”:5,

“farmerName”: “Rahul Sharma”,

“totalSchemes” : 3,

“schemeNames”:[ "Kisan Credit Card (KCC)","Organic Farming Promotion"] },

“farmarId”:7,

“farmerName”: “Rohit Pawar”,

“totalSchemes” : 2,

“schemeNames”:[ "Kisan Credit Card (KCC)","Organic Farming Promotion"

]

},

{

“farmarId”:8,

“farmerName”: “Karan Sharma”,

“totalSchemes” : 3,

“schemeNames”:[ Soil Health Card Scheme","Fasal Bima Yojana”]

}

]

1. **Fetch total amount generated by Each Service**

SELECT service.service\_id as serviceId , service.service\_name as serviceName, SUM(payment.amount\_paid) AS totalMoney

FROM payment payment

JOIN agreement agreement ON payment.agreement\_id = agreement.agreement\_id

JOIN service service ON agreement.reference\_id = service.service\_id

WHERE agreement.type = 'service' AND payment.payment\_status = 'Paid'

GROUP BY service.service\_name, service.service\_id

ORDER BY totalMoney DESC;

**Output :**

[

{

“serviceId”:7,

“serviceName”: “Greenhouse Construction”,

“totalMoney”:7000

},

{

“serviceId”:3,

“serviceName”:” Soil Testing”,

“totalMoney”:3000

},

{

“serviceId”:5,

“serviceName”:” Fertilizer Application”,

“totalMoney”:2000

},

{

“serviceId”:4,

“serviceName”: “Pest Control”,

“totalMoney”:1500

},

]

1. **Fetch how may times each service is accepted**

SELECT service.service\_id as serviceId , service.service\_name as serviceName, COUNT(agreement.agreement\_id) AS totalServices

FROM agreement agreement

JOIN service service ON agreement.reference\_id = service.service\_id

WHERE agreement.type = 'service' AND agreement.status = 'accepted'

GROUP BY service.service\_name,service.service\_id

ORDER BY totalServices DESC;

**Output :**

[

{

“serviceId”=1,

“serviceName”= “Fertilizer Application”,

“totalServices”=1

},

{

“serviceId”=2,

“serviceName”=”Fencing”,

“totalServices”=2

},

{

“serviceId”=3,

“serviceName”= “Harvester”,

“totalServices”=2

},

{

“serviceId”=4,

“serviceName”= “Labour”,

“totalServices”=3

},

{

“serviceId”=5,

“serviceName”= “Cultivevtor”,

“totalServices”=3

},

{

“serviceId”=10,

“serviceName”= “Pest Control”,

“totalServices”=1

},

]

1. **Fetch pending payments of users after complete the agreement**

SELECT payment.payment\_id as paymentId, users.name AS userName, payment.amount\_paid as amountPaid, payment.payment\_status as paymentStatus, agreement.status AS agreementStatus

FROM payment payment

JOIN agreement agreement ON payment.agreement\_id = agreement.agreement\_id

JOIN users users ON agreement.user\_id = users.id

WHERE payment.payment\_status = 'Pending' AND agreement.status= Accepted;

**Output :**

[

{

" paymentId ": 3,

" userName ": "Rajesh Kulkarni",

" amountPaid ": 42000.00,

" paymentStatus ": "Pending",

" agreementStatus ": " Accepted "

}.

{

" paymentId ": 4,

" userName ": "karan Sharma",

" amountPaid ": 22000.00,

" paymentStatus ": "Pending",

" agreementStatus ": " Accepted "

},

{

" paymentId ": 5,

" userName ": "Hardik Joshi",

" amountPaid ": 12000.00,

" paymentStatus ": "Pending",

" agreementStatus ": " Accepted "

},

{

" paymentId ": 6,

" userName ": "Vinod Varma",

" amountPaid ": 15000.00,

" paymentStatus ": "Pending",

" agreementStatus ": " Accepted "

},

{

" paymentId ": 7,

" userName ": "Ram Kumar",

" amountPaid ": 39000.00,

" paymentStatus ": "Pending",

" agreementStatus ": " Accepted "

}

]

1. **List of services provided by service provider**

SELECT users.id as serviceProviderId, users.name AS serviceProviderName, service.service\_id as serviceId, service.service\_name as serviceName, service.price, service.duration as costFor, service.description

FROM users users

JOIN service service ON users.id = service.service\_provider\_id

WHERE users.type\_of\_user = 'service\_provider';

**Output :**

[

{

" serviceProviderId ": 4,

" serviceProviderName ": "GreenGrow Services",

" serviceId ": 1,

" serviceName ": "Tractor Ploughing",

"price": 2500.00,

" costFor ": "Per Acre",

"description": "Tractor ploughing service for land preparation"

},

{

" serviceProviderId ": 7,

"servi serviceProviderName ": "FarmCare Solutions",

" serviceId ": 2,

" serviceName ": "Irrigation Setup",

"price": 7000.00,

" costFor ": "Per Acre",

"description": "Drip irrigation and sprinkler setup for farmlands"

},

{

" serviceProviderId ": 4,

" serviceProviderName ": "GreenGrow Services",

" serviceId ": 3,

" serviceName ": "Soil Testing",

"price": 1500.00,

" costFor ": "Per Sample",

"description": "Detailed soil testing and nutrient analysis"

},

{

" serviceProviderId ": 7,

" serviceProviderName ": "FarmCare Solutions",

" serviceId ": 4,

" serviceName ": "Pest Control",

"price": 3000.00,

" costFor ": "Per Acre",

"description": "Chemical and organic pest control treatments"

},

{

" serviceProviderId ": 4,

" serviceProviderName ": "GreenGrow Services",

" serviceId ": 5,

" serviceName ": "Fertilizer Application",

"price": 2000.00,

" costFor ": "Per Acre",

"description": "Automated and manual fertilizer application service"

},

{

" serviceProviderId ": 7,

"servic serviceProviderName ": "FarmCare Solutions",

" serviceId ": 6,

" serviceName ": "Harvesting Assistance",

"price": 5000.00,

" costFor ": "Per Acre",

"description": "Machinery and labor support for harvesting"

},

{

" serviceProviderId ": 4,

" serviceProviderName ": "GreenGrow Services",

" serviceId ": 7,

" serviceName ": "Greenhouse Construction",

"price": 25000.00,

" costFor ": "Per 1000 sq.ft",

"description": "Greenhouse installation for modern farming"

}

]

1. **Fetch users who not register any property but taken a service**

SELECT DISTINCT users.id as userId, users.name, users.email, users.contact,

ARRAY\_AGG(service.service\_name) as service

FROM users users

JOIN agreement agreement ON users.id = agreement.user\_id

JOIN service service ON service.service\_id= agreement.reference\_id

WHERE users.id NOT IN (SELECT farmer\_id FROM property) AND agreement.type = 'service' and users.type\_of\_user='farmer'

group by users.id, users.name, users.email, users.contact;

Using Join

SELECT DISTINCT users.id as userId, users.name, users.email, users.contact,

ARRAY\_AGG(service.service\_name) as service

FROM users users

JOIN agreement agreement ON users.id = agreement.user\_id AND users.type\_of\_user='farmer' AND agreement.type = 'service'

JOIN service service ON service.service\_id= agreement.reference\_id

LEFT JOIN property property ON property.farmer\_id= agreement.user\_id

where property.farmer\_id is null

group by users.id, users.name, users.email, users.contact;

**Output :**

[

{

" userId ": 10,

"name": "Vijay Sharma",

"email": "vijay@gmail.com",

"contact": "9876540010",

“service”:["Pest Control"]

},

{

" userId ": 1,

"name": "Rohit Pawar",

"email": "rohit@gmail.com",

"contact": "9876512310",

“service”:["Greenhouse Construction"]

},

{

" userId ": 2,

"name": "Nihal Singh",

"email": "nihal@gmail.com",

"contact": "9876523410",

“service”:["Fertilizer"]

},

{

" userId ": 3,

"name": "Manjeet Chiller",

"email": "manjeet@gmail.com",

"contact": "9876534510"

“service”:["Harvestor"]

},

{

" userId ": 4,

"name": "Vijay Sharma",

"email": "vijay@gmail.com",

"contact": "9876545610",

“service”:["Cultivator"]

},

]

1. **Fetch the type of property who have get multiple services**

SELECT p.type\_of\_land as typeOfLand, COUNT(a.agreement\_id) AS totalServices,

STRING\_AGG(s.service\_name,', ') AS serviceNames

FROM property p

JOIN agreement a ON p.property\_id = a.reference\_id

JOIN service s ON s.service\_id=a.reference\_id

WHERE a.type = 'service'

GROUP BY p.type\_of\_land

ORDER BY totalServices DESC;

**Output :**

[

{

" typeOfLand ": "Irrigated Land",

" totalServices ": 3,

“serviceNames”: Soil Testing, Pest Control, Pest Control

},

{

" typeOfLand ": "Rainfed Land",

" totalServices ": 2,

“serviceNames”: Fertilizer Application, Greenhouse Construction

}

]

1. **Fetch list of top 5 user who has most number of agreements**

SELECT users.id as userId, users.name as userName, users.type\_of\_user as typeOfUser, COUNT(agreement.agreement\_id) AS totalAgreements

FROM users users

JOIN agreement agreement ON users.id = agreement.user\_id

GROUP BY users.id, users.name, users.type\_of\_user

ORDER BY totalAgreements DESC

LIMIT 5;

**Output :**

[

{

" userId ": 1,

" userName ": "Rahul Sharma",

" typeOfUser ": "farmer",

" totalAgreements ": 2

},

{

" userId ": 6,

" userName ": "Mahindra Agri Ltd",

" typeOfUser ": "company",

" totalAgreements ": 1

},

{

" userId ": 2,

" userName ": "AgroFarms Pvt Ltd",

" typeOfUser ": "company",

" totalAgreements ": 1

},

{

" userId ": 7,

" userName ": "FarmCare Solutions",

" typeOfUser ": "service\_provider",

" totalAgreements ": 1

},

{

" userId ": 3,

" userName ": "Rajesh Kulkarni",

" typeOfUser ": "farmer",

" totalAgreements ": 1

}

]

1. **Fetch farmers or companies who have leased land but have never taken any services**

SELECT DISTINCT users.id as userId, users.name as userName

FROM users users

JOIN agreement agreement ON users.id = agreement.user\_id AND agreement.type = 'property'

WHERE users.id NOT IN (SELECT DISTINCT user\_id FROM agreement WHERE type = 'service');

Using join:

SELECT DISTINCT users.id as userId, users.name as userName

FROM users users

JOIN agreement agreement1 ON users.id = agreement1.user\_id AND agreement1.type = 'property'

LEFT JOIN agreement agreement2 ON users.id = agreement2.user\_id AND agreement2.type = 'service'

WHERE agreement2.user\_id IS NULL;

**Output :**

[

{

" userId ": 3,

" userName ": "Rajesh Kulkarni"

},

{

" userId ": 6,

" userName ": "Mahindra Agri Ltd"

}

{

" userId ": 10,

" userName ": "Agroson LPP"

},

{

" userId ": 6,

" userName ": "Agrotech LTD"

}

]

1. **Fetch list of properties within specific days**

SELECT users.id as farmerId, users.name as farmerName, property.location, property.create\_date as createdDate

FROM users users

JOIN property property ON users.id = property.farmer\_id

WHERE property.create\_date >= NOW() - INTERVAL '7 days';

**Output :**

[

{

" farmerId ": 1,

" farmerName ": "Rahul Sharma",

"location": "Solapur, Maharashtra",

"create\_date": "2025-03-04 00:23:33.412682"

},

{

" farmerId ": 3,

" farmerName ": "Rajesh Kulkarni",

"location": "Kolhapur, Maharashtra",

" createdDate ": "2025-03-04 00:23:33.412682"

},

{

" farmerId ": 5,

" farmerName ": "Sneha Patil",

"location": "Nagpur, Maharashtra",

" createdDate ": "2025-03-04 00:23:33.412682"

},

{

" farmerId ": 1,

" farmerName ": "Rahul Sharma",

"location": "Pune, Maharashtra",

" createdDate ": "2025-03-04 00:23:33.412682"

},

{

" farmerId ": 3,

" farmerName ": "Rajesh Kulkarni",

"location": "Ahmednagar, Maharashtra",

" createdDate ": "2025-03-04 00:23:33.412682"

},

{

" farmerId ": 5,

" farmerName ": "Sneha Patil",

"location": "Nashik, Maharashtra",

" createdDate ": "2025-03-04 00:23:33.412682"

},

{

" farmerId ": 3,

" farmerName ": "Rajesh Kulkarni",

"location": "Satara, Maharashtra",

" createdDate ": "2025-03-04 00:23:33.412682"

}

]