**Tables**

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

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

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

* **property\_id** (Primary Key)
* **farmer\_id** (Foreign Key → User(id))
* village
* taluka
* District
* State
* type\_of\_land (enum - **Irrigated Land**, Rainfed Land)
* land\_image
* document\_image
* area\_acre
* lease\_price (Rental price if land is leased)
* area\_guntha
* status (Available , notAvailable)
* Create\_date

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

* **id** (Primary Key)
* service\_id ( foreign key -> (services)service\_Id)
* **user\_id** (Foreign Key -> User(id))
* price
* area (per acre)
* available
* village
* taluka
* District
* State
* description
* service\_type (enum - 'Equipment, 'Testing', 'labour',development)
* Create\_date

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

* **agreement\_id** (Primary Key)
* **user\_id** (Foreign Key → User(id))
* **service\_id** (Foreign Key → Service(service\_id) )
* **property\_id** (Foreign Key → Property(property\_id))
* status (enum - 'accepted', 'pending', 'rejected', 'completed')
* start\_date
* end\_date
* time\_duration (If service → work duration, If property → lease duration)(get only days)
* cost (New: Stores final calculated cost)
* create\_date

**5. Payment Table**

* **payment\_id** (Primary Key)
* **agreement\_id** (Foreign Key → Agreement(agreement\_id))
* **total\_amount (** Total amount paid)
* payment\_method ('UPI', 'Credit Card', 'Debit Card', 'Net Banking')(enum)
* payment\_mode ('Online', 'Offline')(enum)
* payment\_status ('Paid', 'Pending', 'Failed')
* **transaction\_id**
* Recived\_date
* payment\_due\_date ( Helps in overdue tracking)

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

* **scheme\_id** (Primary Key)
* title
* benefit\_amount
* 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 (enum - Pending,Accepted,Rejected)

8. Services (stores all the services)

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

**9. property agreement**

**property\_agreement\_id primary key (auto increment)**

**property\_id foreign key(property table)**

**farmer\_id foreignkey (users table)**

**status**

**start\_date**

**end\_date**

**duration**

**cost**

**created\_date**

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 services service

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

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

WHERE agreement.type = 'service'

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

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 services.service\_id AS serviceId, services.service\_name AS serviceName,

COUNT(agreement.agreement\_id) AS totalServices

FROM agreement agreement

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

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

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

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 serviceprovider.id AS serviceProviderId, users.name AS providerName, services.service\_id as serviceId,

services.service\_name as serviceName, serviceprovider.price, serviceprovider.duration, serviceprovider.description

FROM service\_provider serviceprovider

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

JOIN services services ON serviceprovider.service\_id = services.service\_id

ORDER BY providerName;

**Output :**

[

{

" serviceProviderId ": 4,

" providerName ": "GreenGrow Services",

" serviceId ": 1,

" serviceName ": "Tractor Ploughing",

"price": 2500.00,

" costFor ": "Per Acre",

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

},

{

" serviceProviderId ": 7,

" providerName ": "FarmCare Solutions",

" serviceId ": 2,

" serviceName ": "Irrigation Setup",

"price": 7000.00,

" costFor ": "Per Acre",

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

},

{

" serviceProviderId ": 4,

" providerName ": "GreenGrow Services",

" serviceId ": 3,

" serviceName ": "Soil Testing",

"price": 1500.00,

" costFor ": "Per Sample",

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

},

{

" serviceProviderId ": 7,

" providerName ": "FarmCare Solutions",

" serviceId ": 4,

" serviceName ": "Pest Control",

"price": 3000.00,

" costFor ": "Per Acre",

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

},

{

" serviceProviderId ": 4,

" providerName ": "GreenGrow Services",

" serviceId ": 5,

" serviceName ": "Fertilizer Application",

"price": 2000.00,

" costFor ": "Per Acre",

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

},

{

" serviceProviderId ": 7,

" providerName ": "FarmCare Solutions",

" serviceId ": 6,

" serviceName ": "Harvesting Assistance",

"price": 5000.00,

" costFor ": "Per Acre",

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

},

{

" serviceProviderId ": 4,

" providerName ": "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 AS userName, users.email AS userEmail, users.contact AS contactNumber,

ARRAY\_AGG(DISTINCT services.service\_name) AS service

FROM users users

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

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

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

WHERE property.property\_id IS NULL

GROUP BY users.id, users.name, users.email, users.contact;

**Output :**

[

{

" userId ": 10,

"userName": "Vijay Sharma",

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

"contactNumber": "9876540010",

“service”:["Pest Control"]

},

{

" userId ": 1,

" userName ": "Rohit Pawar",

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

" contactNumber ": "9876512310",

“service”:["Greenhouse Construction"]

},

{

" userId ": 2,

" userName ": "Nihal Singh",

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

" contactNumber ": "9876523410",

“service”:["Fertilizer"]

},

{

" userId ": 3,

" userName ": "Manjeet Chiller",

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

" contactNumber ": "9876534510"

“service”:["Harvestor"]

},

{

" userId ": 4,

" userName ": "Vijay Sharma",

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

" contactNumber ": "9876545610",

“service”:["Cultivator"]

},

]

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

SELECT property.type\_of\_land AS typeOfLand,

COUNT(agreement.agreement\_id) AS totalServices,

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

FROM property property

JOIN agreement agreement ON property.property\_id = agreement.property\_id

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

WHERE agreement.type = 'service'

GROUP BY property.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

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",

" createdDate ": "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"

}

]

1. Fetch the farmers Whos Government Schemes are accepted or pending or rejected

select users.id as userId , users.name as userName, ARRAY\_AGG(government.title) as sevices

from users users

join scheme\_applications schemesappln on users.id= schemesappln.farmer\_id

JOIN government\_schemes government ON schemesappln.scheme\_id = government.scheme\_id

where schemesappln.status ='Pending'

group by users.id, users.name;

**Output :**

**[**

{

“userId”:1,

“userName”:”Rahul Sharma”,

“services”:{"Agri Infra Fund"}

},

{

“userId”:3,

“userName”:” Rajesh Kulkarni”,

“services”: {"Fasal Bima Yojana"}

},

{

“userId”:5,

“userName”:” Sneha Patil”,

“services”: {"Organic Farming Promotion"}

}

**]**