

AGRISOCIETY NETWORK **MANAGEMENT SYSTEM**

TEAM 30

Prepared By:

Bhatt Bhavik(18BIT010)

Aaryan Satpal(18BIT001)

Jaimin Rana(18BIT036)

Manan Bharwad(18BIT061)






INDEX

Topic	Page no.
1. Experiment description	3
2.Dataset query program	8
3.Project Description	11
4.DBMS vs File system	13
5.Relational model	17
6.Relational algebra	31
7.ER Model	32
8. Installation of MYSQL workbench	34
9.Create table Syntax	35
10.Filled tables	42
11.SQL queries	49
12.Normalization of tables	56
13.Assignment topic (Stored procedures)	68
14.Research publication	72
15.Our contribution	73

EXPERIMENT DESCRIPTION

- **Experiment 0**



(For assignment 0, dataset was taken in csv file and query was written in java programming and input dataset, output dataset and java code are uploaded here)

 Name ▾	Modified ▾	Created By ▾
 Assignment0.class	September 19	BhavikBhatt
 Assignment0.java	September 19	BhavikBhatt
 Assignment0_input_dataset.csv	September 19	BhavikBhatt
 Assignment0_output_dataset.csv	September 19	BhavikBhatt

- **Experiment 1**

(DBMS vs File system)





Documents > Assignments > 30 > Team Work > **DBMS vs FileSystem**

 Name ▾	Modified ▾	Created By ▾
 DBMS Vs File system in our project.docx	August 23	BhavikBhatt

- **Experiment 2**


(Relational model)

Documents > Assignments > 30 > Team Work > **Relational_schema_diagram**

 Name ▾	Modified ▾	Created By ▾
 Relational_Schema_Diagram (Updated).mwb	A few seconds ago	AaryanSatpal
 Relational_Schema_Diagram (Updated).png	A few seconds ago	AaryanSatpal
 Relational schema diagram.pdf	September 19	AaryanSatpal



- **Experiment 3**
(Relational algebra)

Documents > Assignments > 30 > Team Work > **Relational Algebra**

 Name ▾	Modified ▾	Created By ▾
 Relational algebra.docx	About a minute ago	BhavikBhatt

- **Experiment4**
(ERDiagram)

Documents > Assignments > 30 > Team Work > **ER diagram**

 Name ▾	Modified ▾	Created By ▾
 ER diagram (updated).png	A few seconds ago	BhavikBhatt

- **Experiment5**
(All the our team member installed MYSQL Workbench and one member installed SQL server management studio)

- **Experiment 6**
(Create table syntax)

Documents > Assignments > 30 > Team Work > **crate table commands**

 Name ▾	Modified ▾	Created By ▾
 Create Table.docx	About a minute ago	BhavikBhatt


- **Experiment 7,8,9,10**
(SQL queries)

Documents > Assignments > 30 > Team Work > SQL queries

 Name ▾	Modified ▾	Created By ▾
 SQL queries.docx	A few seconds ago	BhavikBhatt



- **Experiment 11**
(Database normalization)

Documents > Assignments > 30 > Team Work > FDs and Normalization

 Name ▾	Modified ▾	Created By ▾
 FDs and Normalization.docx	5 minutes ago	BhavikBhatt



- **Experiment 12**
(Given topic Stored procedure)

Documents > Assignments > 30 > Team Work > Extra Topic

 Name ▾	Modified ▾	Created By ▾
 Stored Procedure.docx	About an hour ago	BhavikBhatt


- **Experiment 15**
(Book Chapter)

Documents > Assignments > 30 > Team Work > BookChapter

 Name ▾	Modified ▾	Created By ▾
 BookChapter .pdf	About a minute ago	BhavikBhatt


- **Stage1**

Documents > Assignments > 30 > Team Work > **Stage1**

 Name ▾	Modified ▾	Created By ▾
 Stage 1_Team 30.docx	August 16	AaryanSatpal







- **Stage2 (Presentation given in class)**

Documents > Assignments > 30 > Team Work > **Stage2**

 Name ▾	Modified ▾	Created By ▾
 Presentation_stage2.pdf	October 18	BhavikBhatt













- **Personal and Team work folders**

Documents > Assignments > 30

 Name ▾	Modified ▾	Created By ▾
 18BIT001	August 16	AaryanSatpal
 18BIT010	August 16	AaryanSatpal
 18BIT036	August 16	AaryanSatpal
 18BIT061	August 16	AaryanSatpal
 Team Work	August 16	AaryanSatpal

- **Team work folder**

Documents > Assignments > 30 > Team Work

 Name ▾	Modified ▾	Created By ▾
 Assignment0	September 19	BhavikBhatt
 BookChapter	About a minute ago	BhavikBhatt
 ↗ crate table commands	2 hours ago	BhavikBhatt
 ↗ DBMS vs FileSystem	3 hours ago	BhavikBhatt
 ↗ Extra Topic	About an hour ago	BhavikBhatt
 ↗ Relational Algebra	3 hours ago	BhavikBhatt
 ↗ Relational_schema_diagram	About an hour ago	BhavikBhatt
 ↗ SQL queries	2 hours ago	BhavikBhatt
 ↗ Stage1	About an hour ago	BhavikBhatt
 ↗ Stage2	About an hour ago	BhavikBhatt
 ↗ Team_weekly_diaries	About an hour ago	BhavikBhatt

Dataset query program

Java program for registered murder cases district wise from CSV file of overall registered crime cases' dataset and store that in CSV file..

Program:

```
/* Query: Make a list of number of registered murder cases district wise from CSV file of overall registered crime cases' dataset and store that in CSV file. */
```

```
//package com.java.createcsvfile;
```

```
import java.io.FileWriter;
```

```
import java.io.BufferedReader;
```

```
import java.io.FileReader;
```

```
import java.io.IOException;
```

```
import java.util.*;
```

```
public class Assignment0 {
```

```
    public static FileWriter writer=null;
```

```
    public static String location;
```

```
    public static void generateCsvFile(String fileName)
```

```
    {
```

```
        /*This method will create csv file where output will be stored */
```

```
        location="Assignment0_output_dataset.csv";
```

```
        try{
```

```
            writer = new FileWriter(location);
```

```
        }
```

```
        catch(IOException e){
```

```
            e.printStackTrace(); }
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Scanner sc=new Scanner(System.in);
```

```
        String line="";
```

```
        String splitby=",";
```

```
        int c=0;
```

```
        int i=0;
```



```

int l=0;

String [] district=new String[9040];

try{

    generateCsvFile(location);

    //query to find districts with less than 100 Murders in 2001

    BufferedReader br=new BufferedReader(new FileReader("Assignment0_input_dataset.csv"));

    /* here Assignment0_input_dataset is the source dataset */

    System.out.println("Stated districts below had less than 100 murder cases");

    writer.append("District"+" "+"State"+" "+"Murder Cases");

    while((line=br.readLine())!=null){

        if(c==0){

            c=1;

        }

        else{

            String[] data=line.split(splitby);

            int k= Integer.parseInt(data[3]);

            if((k<100 && data[2].equals("2001"))){

                System.out.println(data[1]+" " + "of " + data[0]);

                writer.append("\n");

                writer.append(data[1]+" "+data[0]+" "+data[3]);

                writer.append(",");

                i++;

                writer.flush();

                // writer.close();

            }

        }

    }

}

catch(IOException e){

    e.printStackTrace();

}

```

```
    /*finally{  
    writer.flush();  
    writer.close();  
    } */  
}  
}
```

PROJECT DESCRIPTION

In this project we are making an e-platform that will manage a network of people in agriculture business. We are going to connect them on the basis of their requirements. In abstract this project will lead us to a platform that will provide more opportunities for agricultural growth by developing a proper business system and avoiding problems like price rise in a society. Need is to control price rises of basic needs of society and it is possible by proper management of base of a system (agribusiness). A major purpose of this project is to control hoarding of essential products by providing a better and reliable business platform with greater opportunities for small business holders.

- How we added IOT devices to our database?

→ In our sensor based irrigation system we embedded moisture sensor, temperature sensor and methane sensor.

→ The readings those will be shown by these sensors will be stored in table named Irrigation_system.

→ And the developers will take the data from tables like farm_area from farm table, moisture, temperature and methane information from irrigation_system table and growing crop info from farm table and the needed levels of element will be compared with the existing and the call like watering crop will be taken.

➔ Here is some information about our IOT devices:

IOT device name	In which table it is?
Moisture sensor	irrigation_system
Temperature sensor	irrigation_system
Methane sensor	irrigation_system

IOT device name	What it does?
Moisture sensor	Shows moisture level
Temperature sensor	Shows temperature
Methane sensor	Methane element level

Outcomes and potential impacts of this project:

- As there will be all the data about past and present of production of crop and products, government can take further decision about what to import and what to export for a country
- People involved in agriculture, will be well aware of market situation and government's new schemes
- Circulation of crop or agricultural products will be known to government and if in any circumstances government find lack of crop than according to circulation and production rate they can find if there is hoarding of things happening or not
- Agriculture business world will be reachable and narrow.
- Government and agribusiness holders will be able to interact with each other
- Price rises of basic needs will be controlled due to transparent platform

- There will be a transparent and efficient platform for agribusiness
- Due to gathering of many different professions on a single platform there will be more options for people to grow their business and to connect with people
- There will be less fraud problem because there will not be any unauthorized account and even if any fraud happen then due to deal record, cheater will be punished
- There will be an online platform after completion of this project that will suggest and offer solutions for almost all agricultural businesses.

Benefits:

- “Hoarding of basic products” will be prohibited
 - Poor business holder will have more opportunities
 - Same types of products can be made available at nearly same price
 - Proper requirement based cycle will be generated among people
 - Efficient guidance will be available using features of platform
 - Efficient use of resources
 - An efficient network can be made
 - Control on price rises of daily essential products
-

DBMS vs Filesystem

- **Data redundancy**

In our project we are having some information related to one user, and to represent that if we use file system we might not have facility to store them properly and data may be stored redundantly.

Examples:

- Our platform is based on requirement of people related to agriculture and we are making record to show user's requirement on platform and certain requirements may not change with time and if we use file system there than there will be several DR issues compare to DBMS.
- In our project there is a department of transportation and while showing live location of transportation vehicle there might be possibility that vehicle may not be driven to other place due to some of reason and in that case live location will not be change for specific time and it may cause DR problem. And so we can't use file system here.
- In our project we are storing info of farmer's previous season's crop and there are some types of farming which can be happened all the seasons and in that case previous season's crop value will be repeated and that may generate DR.
- On our platform we are storing information of some of users whose past data we are storing but that might be possibility that it will not change with time and will be repeated. For example, we farmer may not have leased land for many years and so that field will have same value for long time and it will create DR.

- **Data inconsistency**

Examples:

- If registered organization or shop has been sold by registered owner than all the related information of shops and owner must be changed and if we store such information in file system than there will be difficulties and we may need to make another files to represent new data but while in DBMS data edition will be easier and it will be easy to handle data inconsistency.
- If any user leaves platform than we have to remove all the data related to that user from platform and if that data is stored in different files, it may create problem to remove that data because it is scattered but if we handle that data using DBMS than adding or removing data will be easier and DI will be handled.
- We are registering address and contact number of a user and using them at some specific places but if user changes them than we must have to change them from everywhere they exist else there will be data inconsistency and to handle this DBMS is easier to use compare to file system.
- On our platform we are registering workers of small transportation company and if owner sells his one of vehicle or worker stops working than information related to them must be changed else there will be data inconsistency and it is possible to handle easily in DBMS.

- **Atomicity:**

Examples:

- In our application if transportation provider has taken an order so after order is taken than availability status of vehicle should be changed.
- Fertilizer merchant will show available stock of fertilizer but if he gets an order and if order is accepted than at that point of time stock updates should be changed.

→On our application, crop buyer will add his required crops for buying but until he updates that list and buys crop of interest than certain changes in required crop record should be done.

→Once deal between two parties has been done than deal details should be added in deal record.

- **Difficulty in accessing data:**

Examples:

→On our application there are some sensitive information related to user's business and it must be accessed to user only and to do so we must set some criteria by adding some constraints and if we follow file system than we have to do different coding for different kind of data access and hence DBMS is useful in such condition.

→As our project is related to agricultural business so if government want to analyse specific data related to agribusiness than they have to access or retrieve some data and it is easier to do with DBMS than file system.

→If any farmer wants to search about in what price crop buyers want to buy crop than he will have organised results for his search if database for that subject is created using DBMS as data access will be easier.

→If users on application want to know live location of transportation provider than that real time data can be accessed easily than File system.

- **Security issue**

→We are having deal record between two parties and if any cheating occurs than it can be inspected from record but because of real time addition in record it is quite difficult to do with file system than DBMS.

→On our platform we are asking information from user and in that there are some points to be shown on platform and some of are not to be shown on platform due to security purpose and to do so file system can't provide access or constraints and it is easy to do so with DBMS.

→We are not allowing any random person to register on platform without specific information and it is possible by adding some conditions or qualification criteria to register and it will be very complicated with file system as there we will need to write several codes but it can be done using DBMS.

→As all the data of users and their actions is on server, if data gets deleted somehow than it can be restored using back up facility but in case of file system it is difficult.

- **Integrity issues**

Examples:

→On platform, we are making record for deal happened between two parties and there are no restriction on number of deals to do in a day for a user and to handle such a condition file system is not reliable and DBMS can provide such feature.

→In our database, there are some parameters like contact number, time and date, and rating of service which should contain specific type and range and to do so DBMS is feasible solution.

→On our virtual platform we are seeking information like Adhaar card number from user and Adhaar card number is of 12 digits, if user does any mistake while entering it system must stop it and it can be done in DBMS by adding constraints.

→For farmer on our platform, we are asking him about he has leased land or not and answer should be in 'Y' or 'N' and similar example for transportation driver about his availability at current time and he must add this detail with 'Y' and 'N' and not in another form so this kind of situations are easier to handle in DBMS than file system.

- **Data abstraction**

Examples:

→To register on our platform, we are asking information from farmer about contact number, address, current crop in his farm but apart from that for analysis purpose and to predict some schemes for them we are asking some information about Loan and area of leased land and so we must hide such information and we should show his abstract information form on platform and which is possible by creating back end system by DBMS.

→We are asking yearly income of users to analyse average income of users and to compare current income with past's income and this should not be visible to other users apart from details of visible business information of users and even in this case data abstraction helps to hide some information using DBMS.

→There will be details needed to check proof of approval from government for "soil analysis lab" and "fertilizer factories" and such information should not be visible to others apart from organization's visible business information. And this is the concept of data abstraction.

→Transportation provider will have to give some information like RC number of vehicles, vehicle number, vehicle types but in that data set RC number should not be shown on platform but to show other information data abstraction will be used by DBMS.

- **Concurrent access to data:**

on our virtual system, there are some processes which will be done parallel and if we use file system to do so then concurrent access will not be good option and so we will use DBMS for them and examples are given below.

→If farmer and crop buyer does a deal and they have to add their deal details in farmer-crop buyer business record but there is possibility that more than one pair will do deal and update record at the same time.

→Similar example is of fertilizer dealer and farmer that even they will have business deal records and multiple deals between different pairs may occur and they simultaneously add their deal entry.

→ Transportation provider will take orders for transportation and accordingly availability of transportation provider should be changed and there will be possibility that at same time he might get more than 1 order so data must be handled in such case.

→Fertilizer merchant will get order for fertilizer and if he gets more than 1 order at a time then there should be some criteria whom to deliver first and availability stock of fertilizer must be changed accordingly. So in such case of simultaneous access, DBMS is feasible solution.

- **Isolation of data:**

Examples:

→There might be possibility that if application adds new kind of user type with allocated features for them then there is possibility that new features are added by new logical language and as platform is about connecting different kind of users than platform must show compatibility with new technology and so application's database should be made using DBMS else there will be data isolation.

→As our project is about application for agribusiness, so there are high chances that with time passing we need to add some features and if we use file system to create database and if in past we used specific language to add features and now we want to add another feature using another language than there might be possibility that we can't connect certain data of past in file system with new language and it can create data isolation problem.

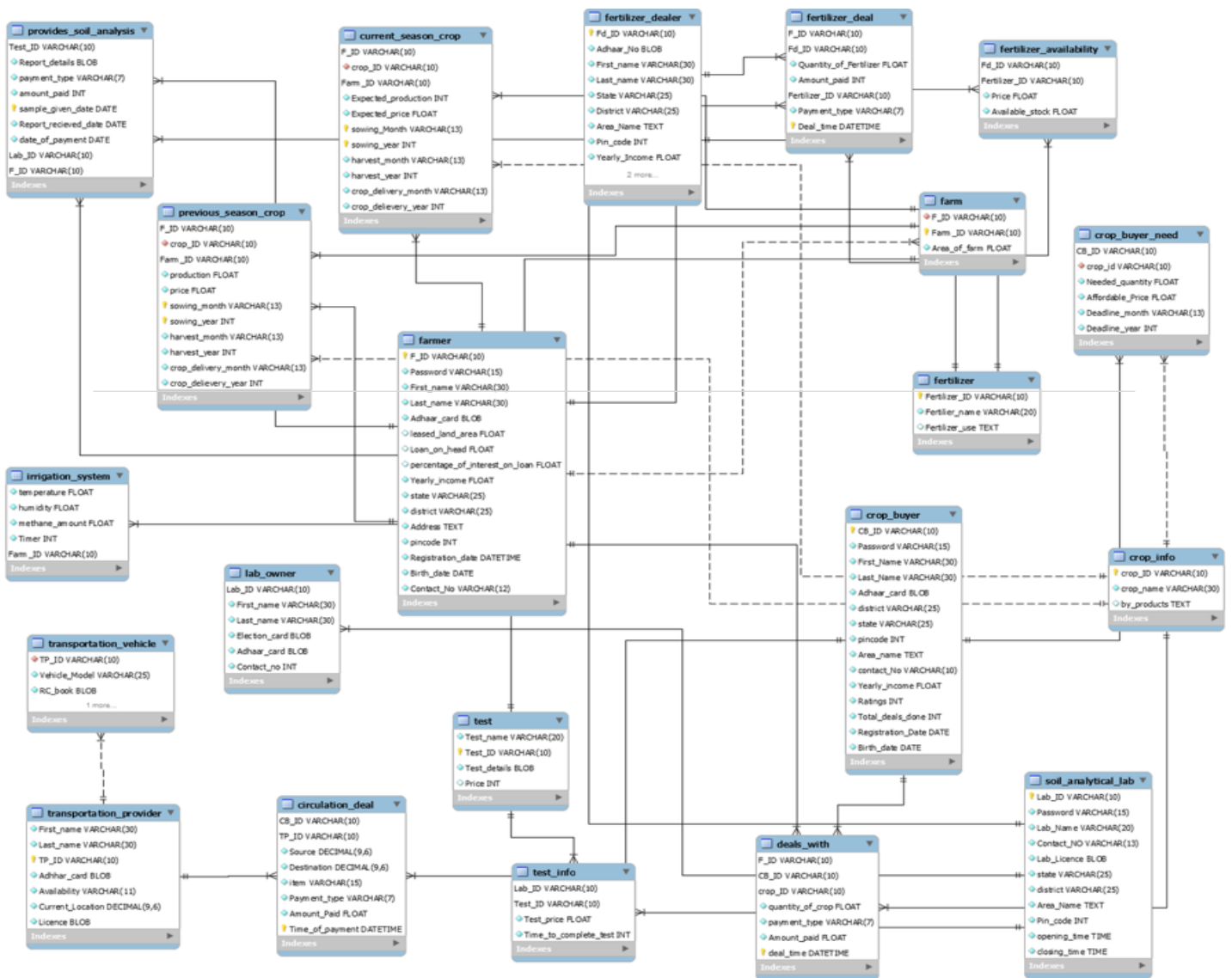
→If we add new features for farmers by seeking some information of them by using new language than the language used for existing data than there should be compatibility between new and old data and as DBMS is compatible with different languages it can be done and data isolation will be avoided.

→In our project, if we want to add voice feature for transportation customer for giving information about vehicle location and to do so we have to make additional data set required for voice feature by connecting location data base that might be added using different language than language is used for voice feature but still we have to have compatibility and it is possible using DBMS only and there will not be data isolation.

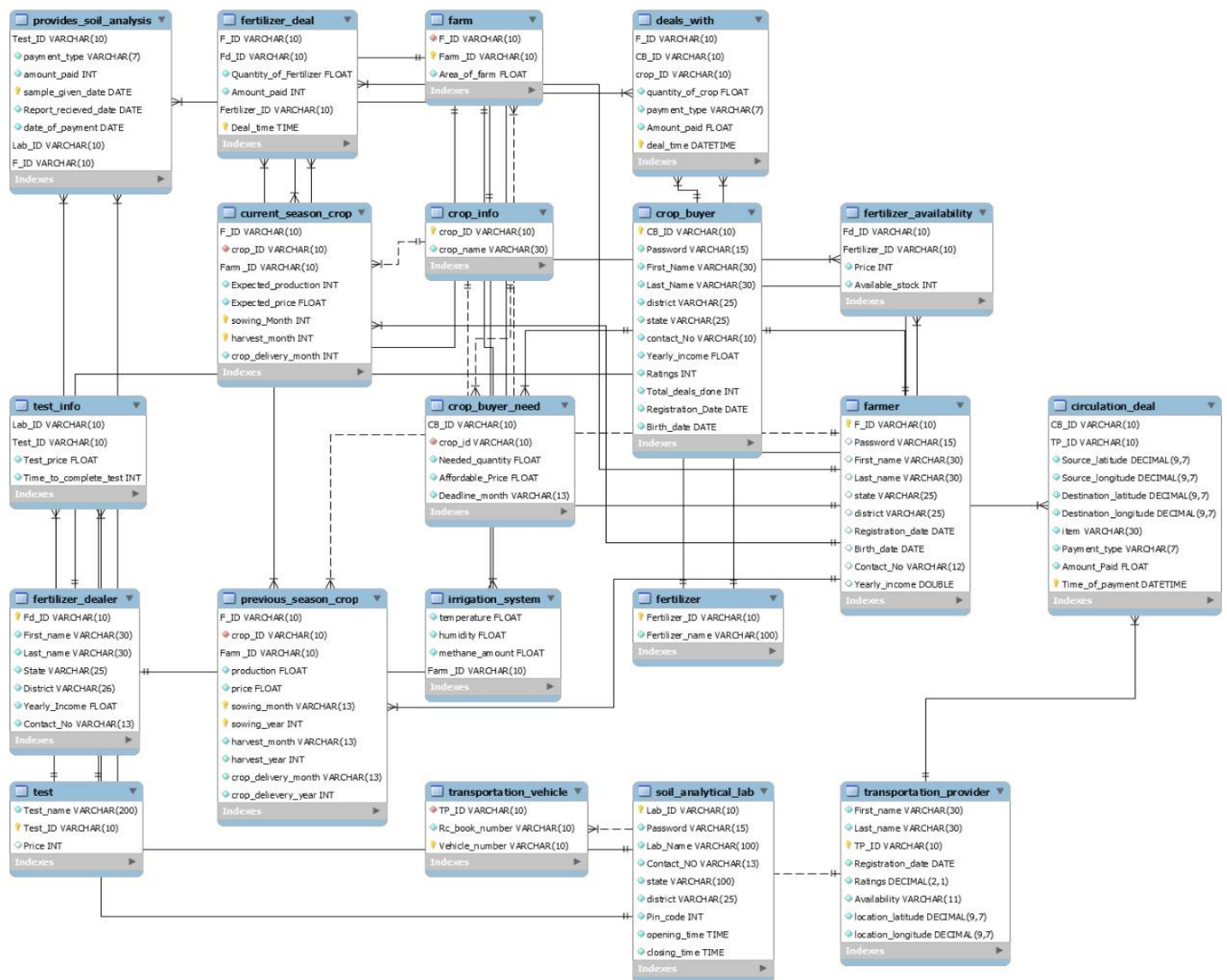
RELATIONAL MODEL (DIAGRAM)

A basic relational model is made to get a look of how our system will look. Nothing is perfect and we will keep on improving our system always seeking the perfect.

(If the image is not clear we have uploaded it in a folder of relational model)



Relational model (Updated)



Relational Model (final Updated)

Sr. No.	Tables
1.	<p>Table name: crop_buyer</p> <p>Primary key: CB_ID</p> <p>Candidate key: CB_ID</p> <p>Foreign key: No key</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. CB_ID , Password 2. CB_ID , First_Name 3. CB_ID , Last_Name 4. CB_ID , district 5. CB_ID , state 6. CB_ID , contact_No 7. CB_ID , Yearly_income 8. CB_ID , Ratings 9. CB_ID , Total_deals_done 10. CB_ID , Registration_Date 11. CB_ID , Birth_date 12. CB_ID , Password , First_Name 13. CB_ID , Password , Last_Name 14. CB_ID , Password , district 15. CB_ID , Password , state 16. CB_ID , Password , contact_No 17. CB_ID , Password , Yearly_income 18. CB_ID , Password , Ratings 19. CB_ID , Password , Total_deals_done 20. CB_ID , Password , Registration_Date 21. CB_ID , Password , Birth_date 22. CB_ID , Password , First_Name , Last_Name 23. CB_ID , Password , First_Name , district 24. CB_ID , Password , First_Name , state

	25. CB_ID , Password , First_Name , contact_No
	26. CB_ID , Password , First_Name , Yearly_income
	27. CB_ID , Password , First_Name , Ratings
	28. CB_ID , Password , First_Name , Total_deals_done
	29. CB_ID , Password , First_Name , Registration_Date
	30. CB_ID , Password , First_Name , Birth_date
	31. CB_ID , Password , First_Name , Last_Name , district
	32. CB_ID , Password , First_Name , Last_Name , state
	33. CB_ID , Password , First_Name , Last_Name , contact_No
	34. CB_ID , Password , First_Name , Last_Name , Yearly_income
	35. CB_ID , Password , First_Name , Last_Name , Ratings
	36. CB_ID , Password , First_Name , Last_Name , Total_deals_done
	37. CB_ID , Password , First_Name , Last_Name , Registration_Date
	38. CB_ID , Password , First_Name , Last_Name , Birth_date
	39. CB_ID , Password , First_Name , Last_Name , district , state
	40. CB_ID , Password , First_Name , Last_Name , district , contact_No
	41. CB_ID , Password , First_Name , Last_Name , district , Yearly_income
	42. CB_ID , Password , First_Name , Last_Name , district , Ratings
	43. CB_ID , Password , First_Name , Last_Name , district , Total_deals_done
	44. CB_ID , Password , First_Name , Last_Name , district , Registration_Date
	45. CB_ID , Password , First_Name , Last_Name , district , Birth_date
	46. CB_ID , Password , First_Name , Last_Name , district , state , contact_No
	47. CB_ID , Password , First_Name , Last_Name , district , state , Yearly_income
	48. CB_ID , Password , First_Name , Last_Name , district , state , Ratings
	49. CB_ID , Password , First_Name , Last_Name , district , state , Total_deals_done
	50. CB_ID , Password , First_Name , Last_Name , district , state , Registration_Date
	51. CB_ID , Password , First_Name , Last_Name , district , state , Birth_date
	52. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income

	<p>53. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Ratings</p> <p>54. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Total_deals_done</p> <p>55. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Registration_Date</p> <p>56. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Birth_date</p> <p>57. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings</p> <p>58. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Total_deals_done</p> <p>59. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Registration_Date</p> <p>60. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Birth_date</p> <p>61. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings , Total_deals_done</p> <p>62. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings , Registration_Date</p> <p>63. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings , Birth_date</p> <p>64. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings , Total_deals_done , Registration_Date</p> <p>65. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings , Total_deals_done , Birth_date</p> <p>66. CB_ID , Password , First_Name , Last_Name , district , state , contact_No , Yearly_income , Ratings , Total_deals_done , Registration_Date , Birth_date</p>
2.	<p>Table name: crop_buyer_need</p> <p>Primary key: CB_ID</p> <p>Candidate key: CB_ID</p> <p>Foreign key:</p> <ol style="list-style-type: none"> 1. CB_ID 2. crop_id <p>Super key:</p>

	<ol style="list-style-type: none"> 1. CB_ID , crop_id 2. CB_ID , Needed_quantity 3. CB_ID , Affordable_Price 4. CB_ID , Deadline_month 5. CB_ID , crop_id , Needed_quantity 6. CB_ID , crop_id , Affordable_Price 7. CB_ID , crop_id , Deadline_month 8. CB_ID , crop_id , Needed_quantity , Affordable_Price 9. CB_ID , crop_id , Needed_quantity , Deadline_month 10. CB_ID , crop_id , Needed_quantity , Affordable_Price , Deadline_month
3.	<p>Table name: crop_info</p> <p>Primary key: crop_ID</p> <p>Candidate key: crop_ID</p> <p>Foreign key: No key</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. crop_ID , crop_name
4.	<p>Table name: current_season_crop</p> <p>Primary key: F_ID , Farm_ID , sowing_Month , harvest_month</p> <p>Candidate key: F_ID , Farm_ID , sowing_Month , harvest_month</p> <p>Foreign key:</p> <ol style="list-style-type: none"> 1. F_ID 2. crop_ID 3. Farm_ID <p>Super key:</p> <ol style="list-style-type: none"> 1. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID 2. F_ID , Farm_ID , sowing_Month , harvest_month , Expected_production 3. F_ID , Farm_ID , sowing_Month , harvest_month , Expected_price 4. F_ID , Farm_ID , sowing_Month , harvest_month , crop_delivery_month 5. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID , Expected_production

	<p>6. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID , Expected_price</p> <p>7. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID , crop_delivery_month</p> <p>8. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID , Expected_production , Expected_price</p> <p>9. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID , Expected_production , crop_delivery_month</p> <p>10. F_ID , Farm_ID , sowing_Month , harvest_month , crop_ID , Expected_production , Expected_price , crop_delivery_month</p>
5.	<p>Table name: farm</p> <p>Primary key: Farm_ID</p> <p>Candidate key: Farm_ID</p> <p>Foreign key:</p> <p>1. F_ID</p> <p>Super key:</p> <p>1. Farm_ID , F_ID</p> <p>2. Farm_ID , Area_of_farm</p> <p>3. Farm_ID , F_ID , Area_of_farm</p>
6.	<p>Table name: provides_soil_analysis</p> <p>Primary key: Fertilizer_ID</p> <p>Candidate key: Fertilizer_ID</p> <p>Foreign key: No key</p> <p>Super key:</p> <p>1. Fertilizer_ID , Fertilizer_name</p>
7.	<p>Table name: provides_soil_analysis</p> <p>Primary key: Lab_ID , F_ID , Test_ID , sample_given_date</p> <p>Candidate key: Lab_ID , F_ID , Test_ID , sample_given_date</p> <p>Foreign key:</p> <p>1. Lab_ID</p> <p>2. Test_ID</p> <p>3. F_ID</p>

	<p>Super key:</p> <ol style="list-style-type: none"> 1. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type 2. Lab_ID , F_ID , Test_ID , sample_given_date , amount_paid 3. Lab_ID , F_ID , Test_ID , sample_given_date , Report_recieved_date 4. Lab_ID , F_ID , Test_ID , sample_given_date , date_of_payment 5. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type , amount_paid 6. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type , Report_recieved_date 7. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type , date_of_paymen 8. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type , amount_paid , Report_recieved_date 9. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type , amount_paid , date_of_payment 10. Lab_ID , F_ID , Test_ID , sample_given_date, payment_type , amount_paid , Report_recieved_date , date_of_payment
8.	<p>Table name: soil_analytical_lab</p> <p>Primary key: Lab_ID</p> <p>Candidate key: Lab_ID</p> <p>Foreign key: No key</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. Lab_ID , Password 2. Lab_ID , Lab_Name 3. Lab_ID , Contact_NO 4. Lab_ID , state 5. Lab_ID , district 6. Lab_ID , Pin_code 7. Lab_ID , opening_time 8. Lab_ID , closing_time 9. Lab_ID , Password , Lab_Name

	10. Lab_ID , Password , Contact_NO 11. Lab_ID , Password , state 12. Lab_ID , Password , district 13. Lab_ID , Password , Pin_code 14. Lab_ID , Password , opening_time 15. Lab_ID , Password , closing_time 16. Lab_ID , Password , Lab_Name , Contact_NO 17. Lab_ID , Password , Lab_Name , state 18. Lab_ID , Password , Lab_Name , district 19. Lab_ID , Password , Lab_Name , Pin_code 20. Lab_ID , Password , Lab_Name , opening_time 21. Lab_ID , Password , Lab_Name , closing_time 22. Lab_ID , Password , Lab_Name , Contact_NO , state 23. Lab_ID , Password , Lab_Name , Contact_NO , district 24. Lab_ID , Password , Lab_Name , Contact_NO , Pin_code 25. Lab_ID , Password , Lab_Name , Contact_NO , opening_time 26. Lab_ID , Password , Lab_Name , Contact_NO , closing_time 27. Lab_ID , Password , Lab_Name , Contact_NO , state , district 28. Lab_ID , Password , Lab_Name , Contact_NO , state , Pin_code 29. Lab_ID , Password , Lab_Name , Contact_NO , state , opening_time 30. Lab_ID , Password , Lab_Name , Contact_NO , state , closing_time 31. Lab_ID , Password , Lab_Name , Contact_NO , state , district , Pin_code 32. Lab_ID , Password , Lab_Name , Contact_NO , state , district , opening_time 33. Lab_ID , Password , Lab_Name , Contact_NO , state , district , closing_time 34. Lab_ID , Password , Lab_Name , Contact_NO , state , district , Pin_code , opening_time 35. Lab_ID , Password , Lab_Name , Contact_NO , state , district , Pin_code , closing_time 36. Lab_ID , Password , Lab_Name , Contact_NO , state , district , Pin_code , opening_time , closing_time
9.	Table name: test

	<p>Primary key: Test_ID</p> <p>Candidate key: Test_ID</p> <p>Foreign key: No key</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. Test_ID , Test_name 2. Test_ID , Price 3. Test_ID , Test_name , Price
10.	<p>Table name: test_info</p> <p>Primary key: Lab_ID , Test_ID</p> <p>Candidate key: Lab_ID , Test_ID</p> <p>Foreign key:</p> <ol style="list-style-type: none"> 1. Lab_ID 2. Test_ID <p>Super key:</p> <ol style="list-style-type: none"> 1. Lab_ID , Test_ID , Test_price 2. Lab_ID , Test_ID , Time_to_complete_test 3. Lab_ID , Test_ID , Test_price , Time_to_complete_test
11.	<p>Table name: transportation_vehicle</p> <p>Primary key: Vehicle_number</p> <p>Candidate key: Vehicle_number</p> <p>Foreign key: TP_ID</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. Vehicle_number , TP_ID 2. Vehicle_number , Rc_book_number 3. Vehicle_number , TP_ID , Rc_book_number
12	<p>Table name: fertilizer</p> <p>Primary key: Fertilizer_ID</p> <p>Candidate key: Fertilizer_ID</p> <p>Foreign key: no key</p> <p>Super key: Fertilizer_ID, fertilizer_name</p>

13	<p>Table name: fertilizer_deal</p> <p>Primary key: F_ID ,Fd_ID Fertilizer_ID, Deal_tim</p> <p>Candidate key: F_ID ,Fd_ID Fertilizer_ID, Deal_tim</p> <p>Foreign key:</p> <ol style="list-style-type: none"> 1. Fd_ID 2. Fertilizer_ID 3. F_ID <p>Super key: F_ID ,Fd_ID Fertilizer_ID, Deal_time, Price</p>
14	<p>Table name: fertilizer_availability</p> <p>Primary key: Fd_ID, Fertilizer_ID</p> <p>Candidate key: Fd_ID, Fertilizer_ID</p> <p>Foreign key: Fd_ID,</p> <p>Fertilizer_ID</p> <p>Super key; Fd_ID, Fertilizer_ID, price</p>
15	<p>Table name: fertilizer_dealer</p> <p>Primary key: Fd_ID</p> <p>Candidate key: Fd_ID</p> <p>Foreign key: no key</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. Fd_ID, First_name 2. Fd_ID, First_name, Last_name 3. Fd_ID, First_name, Last_name 4. Fd_ID, First_name, Last_name, State 5. Fd_ID, First_name, Last_name, State, District 6. Fd_ID, First_name, Last_name, State, District, Yearly_Income 7. Fd_ID, First_name, Last_name, State, District, Yearly_Income, Contact_No
16	<p>Table name: irrigation_system</p> <p>Primary key: Farm_ID</p> <p>Candidate key: Farm_ID</p> <p>Foreign key: no key</p>

	<p>Super key:</p> <ol style="list-style-type: none"> 1. Farm_ID, temperature 2. Farm_ID, temperature, humidity 3. Farm_ID, temperature, humidity, methane_amount 4. Farm_ID, temperature, humidity, methane_amount, Timer
17	<p>Table name: deals_with</p> <p>Primary key: F_ID, CB_ID , Crop_ID, deal_time</p> <p>Candidate key: F_ID, CB_ID , Crop_ID, deal_time</p> <p>Foreign key:</p> <ol style="list-style-type: none"> 1. F_ID 2. CB_ID 3. Crop_ID <p>Super key:</p> <ol style="list-style-type: none"> 1. F_ID, CB_ID , Crop_ID, deal_time, quantity_of_crop 2. F_ID, CB_ID , Crop_ID, deal_time, quantity_of_crop, payment_type 3. F_ID, CB_ID , Crop_ID, deal_time, quantity_of_crop, payment_type, Amount_paid
18	<p>Table name: farmer</p> <p>Primary key: F_ID</p> <p>Candidate key: F_ID</p> <p>Foreign key: no key</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. F_ID, First_name 2. F_ID, First_name, Last_name 3. F_ID, First_name, Last_name, Adhaar_card 4. F_ID, First_name, Last_name, Adhaar_card, leased_land_area 5. F_ID, First_name, Last_name, Adhaar_card, leased_land_area, Loan_on_head 6. F_ID, First_name, Last_name, Adhaar_card, leased_land_area, Loan_on_head , Yearly_income

	<p>7. F_ID, First_name, Last_name, Adhaar_card, State leased_land_area, Loan_on_head , Yearly_income, District</p> <p>8. F_ID, First_name, Last_name, Adhaar_card, leased_land_area, Loan_on_head , Yearly_income, State, District, Address</p> <p>9. F_ID, First_name, Last_name, Adhaar_card, leased_land_area, Loan_on_head , Yearly_income, State, District, Address, Registration_date</p> <p>10. F_ID, First_name, Last_name, Adhaar_card, leased_land_area, Loan_on_head , Yearly_income, State, District, Address, Registration_date, Birth_date</p> <p>11. F_ID, First_name, Last_name, Adhaar_card, leased_land_area, Loan_on_head , Yearly_income, State, District, Address, Registration_date, Birth_date , Contact_No</p>
19	<p>Table name: current_season_crop</p> <p>Primary key: F_ID,Farm_ID,sowing_Month ,harvest_month</p> <p>Candidate key: F_ID,Farm_ID,sowing_Month ,harvest_month</p> <p>Foreign key:</p> <ol style="list-style-type: none"> 1. F_ID 2. `Farm_ID 3. crop_ID <p>Super key:</p> <ol style="list-style-type: none"> 1. F_ID,Farm_ID,sowing_Month ,harvest_month, Expected_production 2. F_ID,Farm_ID,sowing_Month ,harvest_month, Expected_production , Expected_price 3. F_ID,Farm_ID,sowing_Month ,harvest_month, Expected_production, Expected_price , sowing_year 4. F_ID,Farm_ID,sowing_Month ,harvest_month, Expected_production , Expected_price , sowing_year, harvest_year 5. F_ID,Farm_ID,sowing_Month ,harvest_month, Expected_production, Expected_price, sowing_year, harvest_year, crop_delivery_month
20	Table name: circulation_deal

	<p>Primary key: CB_ID, TP_ID, Time_of_payment</p> <p>Candidate key: CB_ID, TP_ID, Time_of_payment</p> <p>Foreign key:</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. CB_ID, TP_ID, Time_of_payment , Source 2. CB_ID, TP_ID, Time_of_payment , Source, Destination 3. CB_ID, TP_ID, Time_of_payment, Source, Destination, item 4. CB_ID, TP_ID, Time_of_payment, Source, Destination, item, Payment_type 5. CB_ID, TP_ID, Time_of_payment, Source, Destination, item, Payment_type, Amount_Paid
21	<p>Table name: transportation_provider</p> <p>Primary key: TP_ID</p> <p>Candidate key: TP_ID</p> <p>Foreign key:</p> <p>Super key:</p> <ol style="list-style-type: none"> 1. TP_ID, First_name 2. TP_ID, First_name. Last_name 3. TP_ID, First_name. Last_name, Adhhar_card 4. TP_ID, First_name. Last_name, Adhhar_card, Availability 5. TP_ID, First_name. Last_name, Adhhar_card, Availability, Current_Location 6. TP_ID, First_name. Last_name, Adhhar_card , Availability, Current_Location, Licence

Relational Algebra

1. This query returns Crop buyers(CB_ID) having maximum ratings and expecting the crop that farmer sow in current season.

→ $\Pi_{CB_ID}[\Pi_{CB_ID, Ratings(crop_buyer)} \times \Pi_{CB_ID, Ratings(crop_buyer)}$
– $\sigma_{Ratings < Ratings2}(\Pi_{CB_ID, Ratings(crop_buyer)})$
 $\times \rho_{Ratings2/Ratings}(\Pi_{CB_ID, Ratings(crop_buyer)})]$
 $\cap \Pi_{CB_ID}[\Pi_{CB_ID, crop_ID}(crop_buyer_need)$
 $\bowtie_{crop_buyer_need.crop_ID = current_season_crop.crop_ID}$
 $\Pi_{crop_ID}(current_season_crop)]$

2. This query returns Labs(Lab_ID, Contact_NO), When a farmer searches labs residing in the same city by Test_name, Test_price.

→ $\Pi_{Lab_ID, Contact_NO} [\Pi_{Lab_ID, Contact_NO, Pin_code} (soil_analytical_lab)$
 $\bowtie_{soil_analytical_lab.Pin_code = farmer.pincode} \Pi_{F_ID, pincode} (farmer)]$
 $\bowtie \Pi_{Lab_ID} [\Pi_{Lab_ID, Test_ID} (test_info)$
 $\bowtie \Pi (\Pi_{Test_ID} (\sigma_{Test_name = "name"}(test))$
 $\cap \Pi_{Test_ID} (\sigma_{Test_price \leq "price"} (test_info)))]$

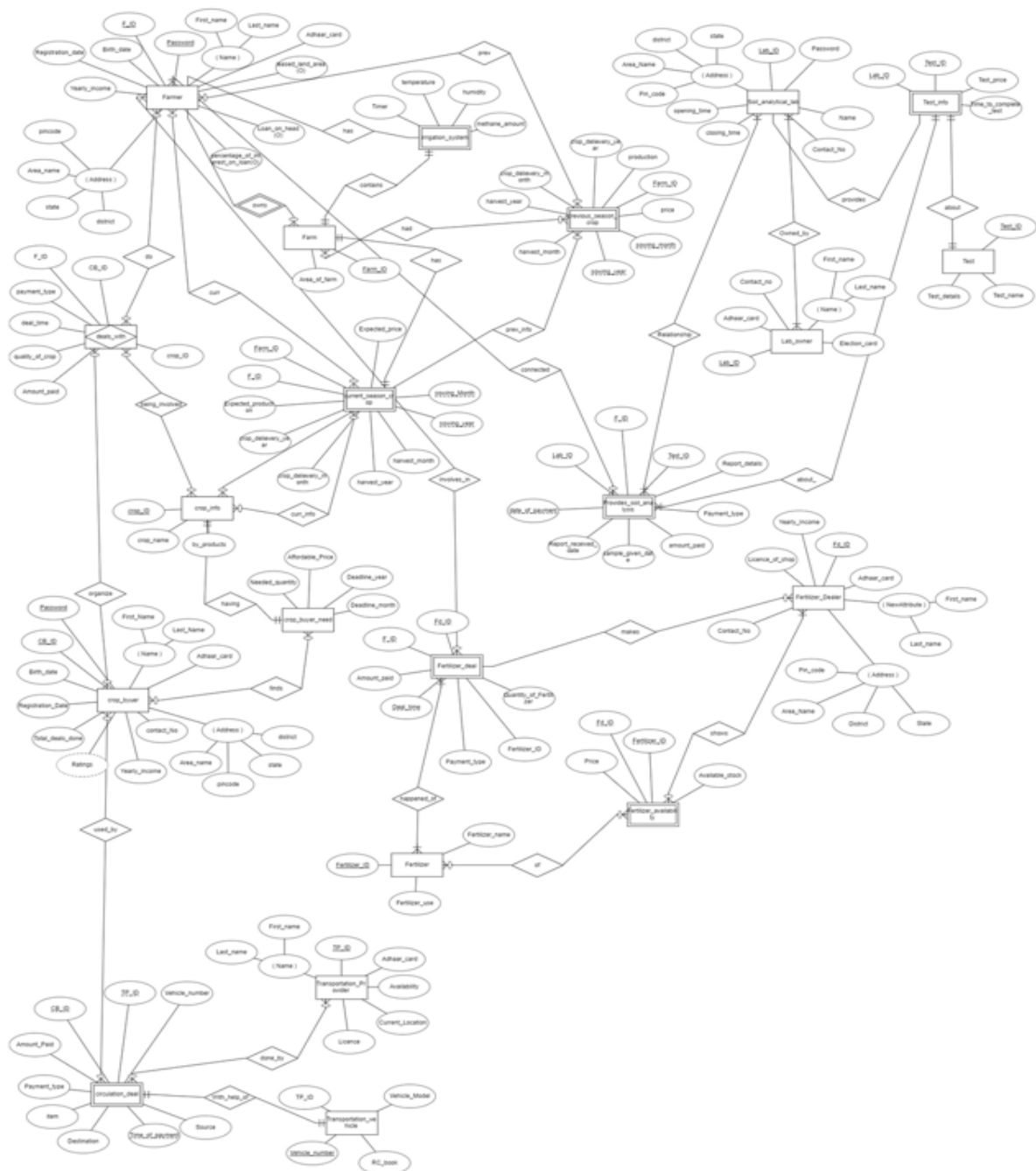
3. This query returns Farmers' details (F_ID, First_name, Contact_No, district, state) whose current season's crop production is higher than previous season's crop production for the same crop.

→ $\sigma_{state = "Gujarat"} [\Pi_{F_ID, First_name, Contact_No, district, state} (farmer)$
 $\bowtie \sigma_{current_season_crop.Expected_production > previous_season_crop.production}$
 $(\Pi_{F_ID, crop_ID, Expected_production} (current_season_crop)$
 $\bowtie \Pi_{F_ID, crop_ID, production} (previous_season_crop))$
 $\bowtie \Pi_{crop_ID, crop_name} (crop_info)]$

4. This query returns Fertilizer dealers' details (Fd_ID, First_name, Contact_No) whose shop's pincode is 382007 and who has Urea fertilizer with available stock greater than or equal to 5.

→ $\Pi_{Fd_ID, First_name, Contact_No} [$
 $\sigma_{Pin_code = 382007} (\Pi_{Fd_ID, First_name, Pin_code, Contact_No} (fertilizer_dealer)$
 $\bowtie \sigma_{Available_stock \geq 5} (\Pi_{Fertilizer_ID} (\sigma_{Fertilizer_name = "Urea"} (fertilizer))$
 $\bowtie (fertilizer_availability)))$
 $]$

ER Model

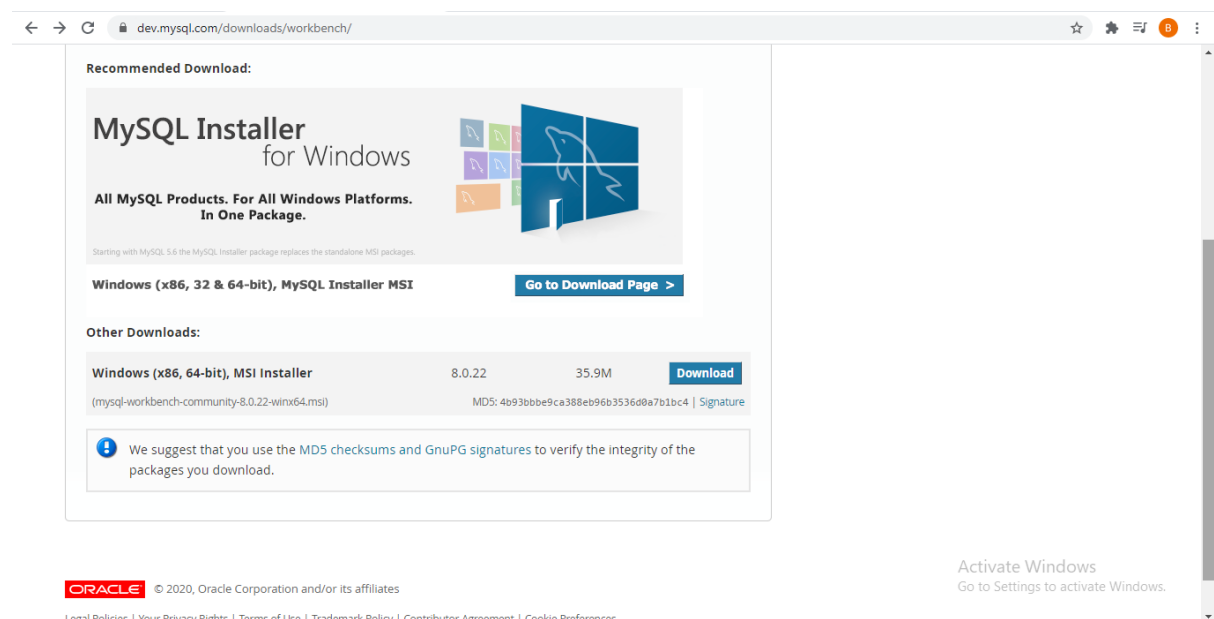


Er diagram (first created diagram)

Installation of MYSQL workbench

- ➔ To make the database of our project and to do task given/assigned by faculty related to SQL queries, All the team members installed MYSQL workbench.
- ➔ Throughout the project we have run our queries on MYSQL workbench when needed and during the lab experiments also, we have used MYSQL workbench.
- ➔ We have downloaded MYSQL workbench from <https://dev.mysql.com/downloads/workbench/>.

Screenshot:



Create table syntax

DROP DATABASE IF EXISTS `agriculture`;

CREATE DATABASE IF NOT EXISTS `agriculture`;

USE `agriculture`;

→ Create Table command for Table named 'Farmer'

```
CREATE TABLE `farmer` (  
  `F_ID` varchar(10) NOT NULL,  
  `Password` varchar(15) DEFAULT NULL,  
  `First_name` varchar(30) DEFAULT NULL,  
  `Last_name` varchar(30) DEFAULT NULL,  
  `state` varchar(25) DEFAULT NULL,  
  `district` varchar(25) DEFAULT NULL,  
  `Registration_date` date DEFAULT NULL,  
  `Birth_date` date DEFAULT NULL,  
  `Contact_No` varchar(12) DEFAULT NULL,  
  `Yearly_income` double DEFAULT NULL,  
  PRIMARY KEY (`F_ID`),  
  CONSTRAINT `farmer_chk_1` CHECK  
  ((length(_cp850'Password') > 6))  
);
```

→ Create Table command for Table named 'Farmer'

```
CREATE TABLE `farm` (  
  `F_ID` varchar(10) NOT NULL,  
  `Farm_ID` varchar(10) NOT NULL,  
  `Area_of_farm` float NOT NULL,  
  PRIMARY KEY (`Farm_ID`),  
  KEY `F_ID` (`F_ID`),  
  CONSTRAINT `farm_ibfk_1` FOREIGN KEY (`F_ID`)  
  REFERENCES `farmer` (`F_ID`)  
);
```

→ Create Table command for Table named 'crop_info'

```
CREATE TABLE `crop_info` (  
  `crop_ID` varchar(10) NOT NULL,  
  `crop_name` varchar(30) NOT NULL,  
  PRIMARY KEY (`crop_ID`)  
);
```

→Create Table command for Table named 'current_season_crop'

```
CREATE TABLE `current_season_crop` (  
  `F_ID` varchar(10) NOT NULL,  
  `crop_ID` varchar(10) NOT NULL,  
  `Farm_ID` varchar(10) NOT NULL,  
  `Expected_production` int NOT NULL,  
  `Expected_price` float NOT NULL,  
  `sowing_Month` int NOT NULL,  
  `harvest_month` int NOT NULL,  
  `crop_delivery_month` int NOT NULL,  
  PRIMARY KEY  
  (`F_ID`,`Farm_ID`,`sowing_Month`,`harvest_month`),  
  KEY `crop_ID` (`crop_ID`),  
  KEY `Farm_ID` (`Farm_ID`),  
  CONSTRAINT `current_season_crop_ibfk_1` FOREIGN KEY  
  (`F_ID`) REFERENCES `farmer` (`F_ID`),  
  CONSTRAINT `current_season_crop_ibfk_2` FOREIGN KEY  
  (`crop_ID`) REFERENCES `crop_info` (`crop_ID`),  
  CONSTRAINT `current_season_crop_ibfk_3` FOREIGN KEY  
  (`Farm_ID`) REFERENCES `farm` (`Farm_ID`)  
);
```

→Create Table command for Table named 'crop_info'

```
CREATE TABLE `circulation_deal` (  
  `CB_ID` varchar(10) NOT NULL,  
  `TP_ID` varchar(10) NOT NULL,  
  `Source_latitude` decimal(9,7) NOT NULL,  
  `Source_longitude` decimal(9,7) NOT NULL,  
  `Destination_latitude` decimal(9,7) NOT NULL,  
  `Destination_longitude` decimal(9,7) NOT NULL,  
  `item` varchar(30) NOT NULL,  
  `Payment_type` varchar(7) NOT NULL,  
  `Amount_Paid` float NOT NULL,  
  `Time_of_payment` datetime NOT NULL,  
  PRIMARY KEY (`CB_ID`,`TP_ID`,`Time_of_payment`),  
  KEY `TP_ID` (`TP_ID`),  
  CONSTRAINT `circulation_deal_ibfk_1` FOREIGN KEY  
  (`CB_ID`) REFERENCES `crop_buyer` (`CB_ID`),  
  CONSTRAINT `circulation_deal_ibfk_2` FOREIGN KEY  
  (`TP_ID`) REFERENCES `transportation_provider` (`TP_ID`)  
);
```

→Create Table command for Table named 'crop_info'

```
CREATE TABLE `crop_buyer` (  
  `CB_ID` varchar(10) NOT NULL,  
  `Password` varchar(15) NOT NULL,  
  `First_Name` varchar(30) NOT NULL,  
  `Last_Name` varchar(30) NOT NULL,  
  `district` varchar(25) NOT NULL,  
  `state` varchar(25) NOT NULL,  
  `contact_No` varchar(10) NOT NULL,  
  `Yearly_income` float NOT NULL,  
  `Ratings` int NOT NULL,  
  `Total_deals_done` int NOT NULL,  
  `Registration_Date` date NOT NULL,  
  `Birth_date` date NOT NULL,  
  PRIMARY KEY (`CB_ID`),  
  CONSTRAINT `crop_buyer_chk_1` CHECK  
  ((length(_cp850'Password') > 6))  
);
```

→Create Table command for Table named 'crop_buyer_need'

```
CREATE TABLE `crop_buyer_need` (  
  `CB_ID` varchar(10) NOT NULL,  
  `crop_id` varchar(10) NOT NULL,  
  `Needed_quantity` float NOT NULL,  
  `Affordable_Price` float NOT NULL,  
  `Deadline_month` varchar(13) NOT NULL,  
  PRIMARY KEY (`CB_ID`),  
  KEY `crop_id` (`crop_id`),  
  CONSTRAINT `crop_buyer_need_ibfk_1` FOREIGN KEY  
  (`CB_ID`) REFERENCES `crop_buyer` (`CB_ID`),  
  CONSTRAINT `crop_buyer_need_ibfk_2` FOREIGN KEY  
  (`crop_id`) REFERENCES `crop_info` (`crop_ID`)  
);
```

→Create Table command for Table named 'deals_with'

```
CREATE TABLE `deals_with` (  
  `F_ID` varchar(10) NOT NULL,  
  `CB_ID` varchar(10) NOT NULL,  
  `crop_ID` varchar(10) NOT NULL,  
  `quantity_of_crop` float NOT NULL,  
  `payment_type` varchar(7) NOT NULL,  
  `Amount_paid` float NOT NULL,  
  `deal_time` datetime NOT NULL,  
  PRIMARY KEY (`F_ID`,`CB_ID`,`crop_ID`,`deal_time`),  
  KEY `crop_ID` (`crop_ID`),  
  KEY `CB_ID` (`CB_ID`),
```

```

        CONSTRAINT `deals_with_ibfk_1` FOREIGN KEY (`F_ID`)
REFERENCES `farmer` (`F_ID`),
        CONSTRAINT `deals_with_ibfk_2` FOREIGN KEY (`crop_ID`)
REFERENCES `crop_info` (`crop_ID`),
        CONSTRAINT `deals_with_ibfk_3` FOREIGN KEY (`CB_ID`)
REFERENCES `crop_buyer` (`CB_ID`)
);

```

→ Create Table command for Table named 'fertilizer'

```

CREATE TABLE `fertilizer` (
  `Fertilizer_ID` varchar(10) NOT NULL,
  `Fertilizer_name` varchar(100) NOT NULL,
  PRIMARY KEY (`Fertilizer_ID`)
);

```

→ Create Table command for Table named 'fertilizer_availability'

```

CREATE TABLE `fertilizer_availability` (
  `Fd_ID` varchar(10) NOT NULL,
  `Fertilizer_ID` varchar(10) NOT NULL,
  `Price` int NOT NULL,
  `Available_stock` int NOT NULL,
  PRIMARY KEY (`Fd_ID`, `Fertilizer_ID`),
  KEY `Fertilizer_ID` (`Fertilizer_ID`),
  CONSTRAINT `fertilizer_availability_ibfk_1` FOREIGN KEY
(`Fd_ID`) REFERENCES `fertilizer_dealer` (`Fd_ID`),
  CONSTRAINT `fertilizer_availability_ibfk_2` FOREIGN KEY
(`Fertilizer_ID`) REFERENCES `fertilizer` (`Fertilizer_ID`)
);

```

→ Create Table command for Table named 'fertilizer_deal'

```

CREATE TABLE `fertilizer_deal` (
  `F_ID` varchar(10) NOT NULL,
  `Fd_ID` varchar(10) NOT NULL,
  `Quantity_of_Fertilizer` float NOT NULL,
  `Amount_paid` int NOT NULL,
  `Fertilizer_ID` varchar(10) NOT NULL,
  `Deal_time` time NOT NULL,
  PRIMARY KEY (`F_ID`, `Fd_ID`, `Fertilizer_ID`, `Deal_time`),
  KEY `Fertilizer_ID` (`Fertilizer_ID`),
  KEY `Fd_ID` (`Fd_ID`),
  CONSTRAINT `fertilizer_deal_ibfk_1` FOREIGN KEY
(`Fertilizer_ID`) REFERENCES `fertilizer` (`Fertilizer_ID`),
  CONSTRAINT `fertilizer_deal_ibfk_2` FOREIGN KEY (`F_ID`)
REFERENCES `farmer` (`F_ID`),
  CONSTRAINT `fertilizer_deal_ibfk_3` FOREIGN KEY
(`Fd_ID`) REFERENCES `fertilizer_dealer` (`Fd_ID`)
);

```

→Create Table command for Table named 'fertilizer_dealer'

```
CREATE TABLE `fertilizer_dealer` (  
  `Fd_ID` varchar(10) NOT NULL,  
  `First_name` varchar(30) NOT NULL,  
  `Last_name` varchar(30) NOT NULL,  
  `State` varchar(25) NOT NULL,  
  `District` varchar(26) NOT NULL,  
  `Yearly_Income` float NOT NULL,  
  `Contact_No` varchar(13) NOT NULL,  
  PRIMARY KEY (`Fd_ID`)  
);
```

→Create Table command for Table named 'irrigation_system'

```
CREATE TABLE `irrigation_system` (  
  `temperature` float NOT NULL,  
  `humidity` float NOT NULL,  
  `methane_amount` float NOT NULL,  
  `Farm_ID` varchar(10) NOT NULL,  
  PRIMARY KEY (`Farm_ID`),  
  CONSTRAINT `irrigation_system_ibfk_1` FOREIGN KEY  
  (`Farm_ID`) REFERENCES `farm` (`Farm_ID`)  
);
```

→Create Table command for Table named 'provides_soil_analysis'

```
CREATE TABLE `provides_soil_analysis` (  
  `Test_ID` varchar(10) NOT NULL,  
  `payment_type` varchar(7) NOT NULL,  
  `amount_paid` int NOT NULL,  
  `sample_given_date` date NOT NULL,  
  `Report_recieved_date` date NOT NULL,  
  `date_of_payment` date NOT NULL,  
  `Lab_ID` varchar(10) NOT NULL,  
  `F_ID` varchar(10) NOT NULL,  
  PRIMARY KEY  
  (`Lab_ID`,`F_ID`,`Test_ID`,`sample_given_date`),  
  KEY `Test_ID` (`Test_ID`),  
  KEY `F_ID` (`F_ID`),  
  CONSTRAINT `provides_soil_analysis_ibfk_1` FOREIGN KEY  
  (`Lab_ID`) REFERENCES `soil_analytical_lab` (`Lab_ID`),  
  CONSTRAINT `provides_soil_analysis_ibfk_2` FOREIGN KEY  
  (`Test_ID`) REFERENCES `test` (`Test_ID`),  
  CONSTRAINT `provides_soil_analysis_ibfk_3` FOREIGN KEY  
  (`F_ID`) REFERENCES `farmer` (`F_ID`)  
);
```

→Create Table command for Table named 'soil_analytical_lab'

```
CREATE TABLE `soil_analytical_lab` (  
  `Lab_ID` varchar(10) NOT NULL,  
  `Password` varchar(15) NOT NULL,  
  `Lab_Name` varchar(100) NOT NULL,  
  `Contact_NO` varchar(13) NOT NULL,  
  `state` varchar(100) NOT NULL,  
  `district` varchar(25) NOT NULL,  
  `Pin_code` int NOT NULL,  
  `opening_time` time NOT NULL,  
  `closing_time` time NOT NULL,  
  PRIMARY KEY (`Lab_ID`)  
);
```

→Create Table command for Table named 'test'

```
CREATE TABLE `test` (  
  `Test_name` varchar(200) NOT NULL,  
  `Test_ID` varchar(10) NOT NULL,  
  `Price` int DEFAULT NULL,  
  PRIMARY KEY (`Test_ID`)  
);
```

→Create Table command for Table named 'test_info'

```
CREATE TABLE `test_info` (  
  `Lab_ID` varchar(10) NOT NULL,  
  `Test_ID` varchar(10) NOT NULL,  
  `Test_price` float NOT NULL,  
  `Time_to_complete_test` int NOT NULL,  
  PRIMARY KEY (`Lab_ID`, `Test_ID`),  
  KEY `Test_ID` (`Test_ID`),  
  CONSTRAINT `test_info_ibfk_1` FOREIGN KEY (`Lab_ID`)  
  REFERENCES `soil_analytical_lab` (`Lab_ID`),  
  CONSTRAINT `test_info_ibfk_2` FOREIGN KEY (`Test_ID`)  
  REFERENCES `test` (`Test_ID`)  
);
```

→Create Table command for Table named 'transportation_provider'

```
CREATE TABLE `transportation_provider` (  
  `First_name` varchar(30) NOT NULL,  
  `Last_name` varchar(30) NOT NULL,  
  `TP_ID` varchar(10) NOT NULL,  
  `Registration_date` date NOT NULL,
```



```
`Ratings` decimal(2,1) NOT NULL,  
`Availability` varchar(11) NOT NULL,  
`location_latitude` decimal(9,7) NOT NULL,  
`location_longitude` decimal(9,7) NOT NULL,  
PRIMARY KEY (`TP_ID`)  
);
```

→ Create Table command for Table named 'transportation_vehicle'

```
CREATE TABLE `transportation_vehicle` (  
  `TP_ID` varchar(10) NOT NULL,  
  `Rc_book_number` varchar(10) NOT NULL,  
  `Vehicle_number` varchar(10) NOT NULL,  
  PRIMARY KEY (`Vehicle_number`),  
  KEY `TP_ID` (`TP_ID`),  
  CONSTRAINT `transportation_vehicle_ibfk_1` FOREIGN KEY  
  (`TP_ID`) REFERENCES `transportation_provider` (`TP_ID`)  
);
```

Filled Tables

1. farmer

	F_ID	Password	First_name	Last_name	state	district	Registration_date	Birth_date
	F000315	BVFTHGkc4b	sanjay	chabbria	Gujarat	Kheda	2004-08-02	1950-12-09
	F000322	YPCqDQ	anuradha	sarwe	Andaman Nicobar	Sirsa	2004-08-31	1976-01-27
	F000555	I9A6xkz	geeta	ramelwar	Andaman Nicobar	Sukma	2005-02-18	1962-05-26
	F001381	tlqkcOV	gappu	gaitonde	Madhya Pradesh	Chhatarpur	2018-06-18	1953-02-01
	F001521	K6lxOfwec3	ram kishan	darbari	Jammu Kashmir	Reasi	2015-05-05	1989-03-14
	F003268	5QCinVzE82A	durgesh	ahuja	Arunachal Pradesh	Central Siang	2014-05-27	1967-06-20
	F003753	KFscDDL	sarvesh	bhattacha...	Chhattisgarh	Narayanpur	2011-11-26	1969-10-26
	F003877	WhymCLMN	naval kish...	gaur	Madhya Pradesh	Khandwa	2003-03-26	1980-03-06
	F004912	P5xvW4stc	sohna1 ku...	indulkar	Mizoram	Champhai	2018-07-26	1971-07-21
	F007111	atXx11dB8Dx	bhola sarkar	chaturvedi	Haryana	Jhajjar	2007-12-20	1986-01-23

2. farm

	F_ID	Farm_ID	Area_of_farm
▶	FA12329	F12329	1450
	FA12330	F12330	1400
	FA12340	F12340	1600
	FA12341	F12341	1700
	FA12345	F12345	2000
	FA23329	F23329	2200
	FA34330	F34330	1700
	FA45340	F45340	1600
	FA67341	F67341	1700
	FA89345	F89345	1800

3. crop_info

	crop_ID	crop_name
▶	crop000315	wkkkdhun
	crop000322	qznnftlp
	crop000555	ouexgcmx
	crop001195	rwqarqhp
	crop001295	rkypolrz
	crop001381	uxypktje
	crop001521	bihfajnb
	crop002118	gcxoaelp
	crop002390	rswhdapf
	crop003268	npsrtjhx

4. current season crop

	F_ID	crop_ID	Farm_ID	Expected_production	Expected_price	sowing_Month	harvest_month	crop_delivery_month
▶	F000315	crop567	FA000315	215	1769	2	7	3
	F001381	crop013	FA001381	223	1620	2	6	3
	F001521	crop710	FA001521	337	1357	1	8	2
	F003268	crop263	FA003268	217	1268	2	11	2
	F003753	crop375	FA003753	307	1147	6	9	3
	F003877	crop784	FA003877	278	825	6	10	3
	F004912	crop165	FA004912	375	1594	2	8	2
	F007111	crop094	FA007111	398	1567	1	11	4
	F007606	crop666	FA007606	268	1559	4	6	2
	F008336	crop763	FA008336	245	1302	3	7	1

5. crop buyer

	CB_ID	Password	First_Name	Last_Name	district	state	contact_No	Yearly_income	Ratings	Total_deals_done	Registration_Date	Birth_date
▶	CB000315	BVFTHGkgc4b	raj kumar	mohanty	Junagadh	Gujarat	9634764164	282650	2	99	2004-08-02	1950-12-09
	CB000322	YPCqDQ	manjeet	jadi	Rohtak	Andaman Nicobar	9625015321	396868	3	60	2004-08-31	1976-01-27
	CB000555	I9A6xkz	ramkaran	gore	Rajnandgaon	Andaman Nicobar	9635315138	354588	1	248	2005-02-18	1962-05-26
	CB001195	ko9yE0y	vivek auhari	bundela	Kaushambi	Uttar Pradesh	9665595597	352622	2	269	2017-03-08	1961-06-28
	CB001295	vAGHZv2GizMH	aman	agvan	Bharatpur	Rajasthan	9607351430	266282	4	424	2009-07-29	1980-08-08
	CB001381	tlqkcOV	reena	pawar	Chachaura	Madhya Pradesh	9624750290	628161	2	178	2018-06-18	1953-02-01
	CB001521	K6xOfwec3	vandana	nehra	Ramban	Jammu Kashmir	9627532788	741488	1	385	2015-05-05	1989-03-14
	CB002118	oIgcCaK	nausad	ameer	Kota	Rajasthan	9611194857	702965	2	138	2005-03-21	1971-03-23
	CB002390	dIpKH88ACZh	deepak	bhan	Unakoti	Tripura	9605890435	518705	2	486	2011-11-28	1990-11-13
	CB003268	5QCInVzE82A	goldy	karwande	Anjaw	Arunachal Pradesh	9664330417	608358	4	56	2014-05-27	1967-06-20

6. crop buyer need

	CB_ID	crop_id	Needed_quantity	Affordable_Price	Deadline_month
▶	CB000315	crop567	15	537	2
	CB000322	crop821	13	519	2
	CB000555	crop211	16	370	2
	CB001195	crop784	13	351	4
	CB001381	crop013	24	454	4
	CB001521	crop710	17	351	2
	CB002390	crop047	19	361	2
	CB003268	crop263	16	456	4
	CB003753	crop375	20	455	3
	CB003877	crop784	23	354	4

7. deals with

	F_ID	CB_ID	crop_ID	quantity_of_crop	payment_type	Amount_paid	deal_time
►	F000315	CB000315	crop013	67	Offline	4020	2012-11-07 05:13:52
	F000322	CB000322	crop014	59	Offline	3540	2009-10-17 00:11:51
	F000555	CB000555	crop027	17	Offline	1020	2013-07-12 17:43:34
	F001381	CB001195	crop047	31	Offline	1860	2007-07-20 07:43:12
	F001521	CB001295	crop050	39	Offline	2340	2007-03-30 21:04:19
	F003268	CB001381	crop055	40	Offline	2400	2006-09-08 17:00:40
	F003753	CB001521	crop063	50	Offline	3000	2002-03-19 05:48:47
	F003877	CB002118	crop070	24	Offline	1440	2017-05-19 04:10:24
	F004912	CB002390	crop076	16	Offline	960	2006-11-29 05:17:27
	F007111	CB003268	crop080	62	Offline	3720	2014-10-12 09:27:31

8. soil analytical lab

	Lab_ID	Password	Lab_Name	Contact_NO	state	district	Pin_code	opening_time	closing_time
►	L000806	qqvsQw35	KVK Dhamtari Mini Soil Testing Lab	7746164508	Chhattisgarh	Balod	81307	11:04:00	17:10:00
	L001734	fsIf7gz	District Soil Testing Lab , Sitamarhi	1216195996	Bihar	Araria	738013	08:36:00	18:57:00
	L002601	SQ7mjgPM	KAMANI SCIENCE and PRATAPRAI ARTS COLL...	4087441269	Gujarat	Ahmadabad	170551	08:02:00	18:11:00
	L002603	TId3vsK3SYWc	Professor & Head,Department of Chemistry,NMCA	6143703525	Gujarat	Ahmadabad	436530	10:00:00	18:39:00
	L003340	cCqtJgikYv	KVK, Kaimur (Bhabua) Soil & Water Testing Lab	7519660931	Bihar	Araria	383097	09:45:00	19:20:00
	L003631	YpYC9kz	KVK, Darang	6868733011	Assam	Baksa	45955	11:17:00	18:33:00
	L004377	5r2kA9tV	Soil Testing Lab,Kulgam	5695768890	Jammu And Kashmir	Anantnag	296239	09:35:00	17:28:00
	L004881	OKW80h02	MSTL_BARAMULLA HORTICULTURE	1554767292	Jammu And Kashmir	Anantnag	334010	11:33:00	18:11:00
	L007204	41xyIzOQxi	Soil Testing Laboratory,Agriculture complex,Doda.	6679263815	Jammu And Kashmir	Anantnag	312027	11:54:00	17:48:00
	L009494	abuN7X	District Agriculture Office, KAMRUP METRO , AS...	5958040367	Assam	Baksa	138584	08:11:00	17:42:00

9. test

	Test_name	Test_ID	Price
►	Nitrogen (N)-50, Phosph...	T000315	888
	Nitrogen (N)-20, Phosph...	T001521	1493
	Nitrogen (N)-13, Phosph...	T003268	746
	Nitrogen (N)-0, Phospho...	T003753	1161
	Nitrogen (N)-1	T007111	1331
	Nitrogen (N)-20, Phosph...	T007606	1115
	Phosphorus (P)-0, EC-0,...	T008336	1430
	Nitrogen (N)-0, Phospho...	T010667	1339
	Nitrogen (N)-2, Phospho...	T012386	923
	Nitrogen (N)-10, Phosph...	T017958	683

10.test_info

	Lab_ID	Test_ID	Test_price	Time_to_complete_test
▶	L000806	T796675	1365	13
	L001734	T774524	864	9
	L002601	T123485	1323	8
	L002603	T270442	571	11
	L003340	T063040	865	12
	L003631	T283920	1037	14
	L004377	T020441	1159	8
	L004881	T872523	771	12
	L007204	T095219	1337	11
	L009494	T270442	571	10

11.provides_soil_analysis

	Test_ID	payment_type	amount_paid	sample_given_date	Report_recieved_date	date_of_payment	Lab_ID	F_ID
▶	T123485	Offline	1207	2020-03-24	2020-04-22	2020-05-29	L002601	F354130
	T063040	Offline	549	2020-03-09	2020-04-19	2020-05-24	L003340	F040952
	T283920	Offline	1280	2020-03-04	2020-04-19	2020-05-21	L003631	F812816
	T872523	Offline	605	2020-03-16	2020-04-10	2020-05-23	L004881	F786045
	T095219	Offline	864	2020-03-03	2020-04-05	2020-05-17	L007204	F633762
	T397732	Offline	999	2020-03-13	2020-04-15	2020-05-22	L018644	F395889
	T029080	Offline	763	2020-03-24	2020-04-20	2020-05-08	L021699	F615000
	T079992	Offline	1234	2020-03-25	2020-04-04	2020-05-13	L021868	F784744
	T478185	Offline	600	2020-03-29	2020-04-10	2020-05-02	L036614	F758025
	T437652	Offline	1399	2020-03-16	2020-04-06	2020-05-06	L037182	F274790

12.fertilizer_dealer

	Fd_ID	First_name	Last_name	State	District	Yearly_Income	Contact_No
▶	FD000315	sanjay	chabbria	Gujarat	Kheda	282650	9634764164
	FD000322	anuradha	sarwe	Andaman Nicobar	Sirsa	396868	9625015321
	FD000555	geeta	ramelwar	Andaman Nicobar	Sukma	354588	9635315138
	FD001195	pankaj	molkar	Uttar Pradesh	Kheri	352622	9665595597
	FD001295	bhavesdass	kelkar	Rajasthan	Bhilwara	266282	9607351430
	FD001381	gappu	gaitonde	Madhya Pradesh	Chhatarpur	628161	9624750290
	FD001521	ram kishan	darbari	Jammu Kashmir	Reasi	741488	9627532788
	FD002118	suresh	khobragade	Rajasthan	Nagaur	702965	9611194857
	FD002390	parbhat	mallya	Tripura	West Tripura	518705	9605890435
	FD003268	durgesh	ahuja	Arunachal Pradesh	Central Siang	608358	9664330417

13.fertilizer

	Fertilizer_ID	Fertilizer_name
▶	FE003268	N.P.K. (22-22-11)
	FE012386	N.P.K. (17-17-17)
	FE063905	N.P.K. 15::9:20
	FE079992	Single Superphosphate (16% P2O 5 Granulated)
	FE100616	Urea (Granular)
	FE1232	Nitrogen
	FE167851	Urea (46% N) (While free flowing)
	FE2131	Phosphorus
	FE241530	Anhydrous Ammonia
	FE249857	Single Superphosphate (16% P2O 5 Powdered)

14.fertilizer availability

	Fd_ID	Fertilizer_ID	Price	Available_stock
	FD983603	FE100616	1384	648
	FD983728	FE347751	1445	983
	FD984784	FE702920	1485	991
	FD989949	FE003268	1501	597
	FD990607	FE100616	1675	938
	FD991172	FE100616	1257	751
	FD992438	FE495777	1974	649
	FD994623	FE493093	1457	554
	FD995409	FE493744	1359	726

15.fertilizer deal

	F_ID	Fd_ID	Quantity_of_Fertilizer	Amount_paid	Fertilizer_ID	Deal_time
	F983322	FD983322	352	540	FE964717	12:07:00
	F983603	FD983603	197	4521	FE100616	08:42:00
	F995409	FD995409	420	4697	FE493744	07:11:00
	F996834	FD996834	270	871	FE249857	21:58:00
	FA12329	FD1234	200	2000	FE1232	00:00:00
	FA12330	FD2312	250	3000	FE1232	00:00:00
	FA12340	FD4562	210	2500	FE1232	00:00:00
	FA12341	FD6532	220	2000	FE1232	00:00:00
	FA12345	FD8267	230	1800	FE1232	00:00:00

16. irrigation_system

	temperature	humidity	methane_amount	Farm_ID
►	15	30	4	FA000315
	32	21	2	FA001381
	21	11	4	FA001521
	17	37	5	FA003268
	25	29	10	FA003753
	13	20	9	FA003877
	14	23	1	FA004912
	15	26	3	FA007111
	33	37	1	FA007606
	10	17	4	FA008336

17. transportation_provider

	First_name	Last_name	TP_ID	Registration_date	Ratings	Availability	location_latitude	location_longitude
►	Kirbie	Lampbrecht	TP0004	2012-04-30	5.0	Available	22.8275820	47.2174870
	Lorena	Klass	TP0021	2012-09-20	3.9	Available	39.9042000	35.8939570
	Melissa	Jewks	TP0038	2009-03-05	4.5	Available	40.5397100	48.8270850
	Ev	Trumble	TP0070	2009-07-28	1.7	Available	-3.8008890	29.5785800
	Harper	Caseborne	TP0083	2012-01-01	1.2	Available	0.5617340	57.0237450
	Baxy	Palke	TP0097	2008-12-29	2.3	Available	35.6958420	50.0622000
	Jere	Kettlesing	TP0104	2016-03-16	4.2	Unavailable	23.8752080	-35.0026610
	Levey	Bracegirdle	TP0114	2013-06-16	3.7	Available	14.5925110	13.8151890
	Sheilah	Newlan	TP0118	2017-01-12	1.7	Available	-19.1097390	55.8389850
	Elvin	Vasenvkov	TP0127	2013-10-04	3.8	Available	33.3876840	35.6963950

18. transportation_vehicle

	TP_ID	Rc_book_number	Vehicle_number
►	TP0004	1219476	AN01AO7582
	TP0021	6583911	AN01EI4293
	TP0038	8501497	AN01FO3436
	TP0070	8092152	AN01UJ2029
	TP8113	0444222	AN02BZ2297
	TP8233	7980652	AN02CU4683
	TP7993	7911963	AN02DB9779
	TP8058	5490227	AN02GK1858
	TP8098	4976830	AN02JV0729
	TP8238	0661815	AN02KR1887

19.circulation deal

	CB_ID	TP_ID	Source_latitude	Source_longitude	Destination_latitude	Destination_longitude	item	Payment_type	Amount_Paid	Time_of_payment
▶	CB000315	TP0004	22.8275820	47.2174870	14.3839330	35.2853170	Banana	Offline	840	2004-01-20 20:17:36
	CB000322	TP0021	39.9042000	35.8939570	51.7783030	58.4986110	Plums	Offline	2480	2016-05-29 06:47:05
	CB000555	TP0038	40.5397100	48.8270850	43.0429120	53.1286980	Peach	Offline	4480	2020-06-18 16:42:34
	CB001195	TP0070	-3.8008890	29.5785800	55.6771780	48.4034050	Khesari	Offline	3360	2010-08-27 10:17:40
	CB001295	TP0083	0.5617340	57.0237450	13.5419180	58.7028900	Blackgram	Offline	2000	2012-12-15 02:03:19
	CB001381	TP0097	35.6958420	50.0622000	46.2372400	49.7582150	Cashewnut	Offline	3920	2009-03-09 14:07:37
	CB001521	TP0104	23.8752080	-35.0026610	-8.5039040	24.3180670	Cond-spcs other	Offline	3320	2019-10-04 21:18:51
	CB002118	TP0114	14.5925110	13.8151890	4.0627140	0.2221600	Total foodgrain	Offline	5920	2017-08-21 01:11:48
	CB002390	TP0118	-19.1097390	55.8389850	41.5659300	61.2370210	Beet Root	Offline	400	2008-12-23 01:41:04
	CB003268	TP0127	33.3876840	35.6963950	40.8246050	-27.4794140	Bean	Offline	2480	2017-12-14 08:47:41

SQL Queries

(Using aggregate functions)

1. Display highest amount_paid with their F_ID , Lab_ID , Test_ID ,sample_given_date , Report_recieved_date for those farmers who have a higher amount_paid in a day is within the range 1400 and 1500.

➔ SELECT F_ID , Lab_ID , Test_ID , sample_given_date , Report_recieved_date , MAX(amount_paid)

FROM provides_soil_analysis

GROUP BY F_ID , Lab_ID

HAVING MAX(amount_paid) BETWEEN 1400 AND 1500;

Output:

	F_ID	Lab_ID	Test_ID	sample_given_date	Report_recieved_date	MAX(amount_paid)
▶	F003753	L138703	T740014	2020-03-02	2020-04-25	1415
	F010667	L643687	T713748	2020-03-19	2020-04-18	1490
	F084178	L986103	T146029	2020-03-18	2020-04-02	1488
	F146738	L843981	T052859	2020-03-25	2020-04-28	1462
	F167851	L643372	T167851	2020-03-29	2020-04-08	1476
	F196465	L727405	T007111	2020-03-27	2020-04-19	1417
	F214496	L744587	T356682	2020-03-15	2020-04-04	1430
	F376738	L427896	T063040	2020-03-08	2020-04-09	1476
	F399590	L826627	T220279	2020-03-08	2020-04-06	1424
	F444998	L513302	T354130	2020-03-21	2020-04-28	1496
	F615149	L468936	T602091	2020-03-14	2020-04-04	1491
	F643381	L057298	T330534	2020-03-14	2020-04-22	1425
	F683521	L223715	T796784	2020-03-11	2020-04-14	1480
	F698720	L555794	T311184	2020-03-22	2020-04-05	1411
	F762417	L366224	T143680	2020-03-06	2020-04-17	1429
	F808610	L524557	T683521	2020-03-18	2020-04-12	1491
	F894884	L076448	T894884	2020-03-14	2020-04-06	1482

2. Display the number of deals happened between Crop buyer and transportation provider in 2015 to 2016 year.

➔ SELECT COUNT(*)

FROM circulation_deal

WHERE Time_of_payment >= '2015-01-01 00:00:00.000' and
Time_of_payment < '2016-01-01 00:00:00.000 ' ;

Output:

	COUNT(*)
▶	34

3. Display average amount earned within a month 2020-01-01 to 2020-02-01 of farmers with their F_ID.

➔ SELECT F_ID , AVG(Amount_paid) as amount_earned

FROM deals_with

WHERE deal_time >= '2020-01-01 00:00:00.000' and deal_time < '2020-02-01 00:00:00.000 '

GROUP BY F_ID ;

Output:

	F_ID	amount_earned
▶	F178032	1800
	F325730	1020
	F571915	3000
	F794803	2880
	F840542	3480

4. Display Sum of Amount_Paid done on online within 2019 to 2020 year between transportation provider and crop buyer.

➔ SELECT SUM(Amount_paid)

FROM circulation_deal

WHERE Payment_type = 'Online' and Time_of_payment >= '2019-01-01 00:00:00.000' and Time_of_payment < '2020-01-01 00:00:00.000 ' ;

Output:

	SUM(Amount_paid)
▶	61440

SQL Queries

1. This query is for finding work_status of lab on the basis of time taken to complete soil analytical test and price taken by lab. This query can be used to check labs are doing their work properly or not.

➔ SELECT provides_soil_analysis.Lab_ID, Lab_Name, Contact_No, Area_Name, provides_soil_analysis.Test_ID, amount_paid, F_ID, DATEDIFF(Report_recieved_date,sample_given_date) as days_for_test,

CASE WHEN

DATEDIFF(Report_recieved_date,sample_given_date)>14 AND amount_paid>1800 AND provides_soil_analysis.Test_ID='T30789' THEN 'Expensive and late'

WHEN DATEDIFF(Report_recieved_date,sample_given_date)>14 AND amount_paid<=1800 AND provides_soil_analysis.Test_ID='T30789' THEN 'late with ok price'

WHEN DATEDIFF(Report_recieved_date,sample_given_date)<14 AND amount_paid>1800 AND provides_soil_analysis.Test_ID='T30789' THEN 'On time and expensive'

ELSE 'On time with ok price'

END as deal_status

FROM soil_analytical_lab,provides_soil_analysis

WHERE soil_analytical_lab.Lab_ID=provides_soil_analysis.Lab_ID;

Output:

	Lab_ID	Lab_Name	Contact_No	Area_Name	Test_ID	amount_paid	F_ID	days_for_test	deal_status
►	SL1234	Best_lab	7867890909	Gurukrupa_society	T30789	1800	FA12329	17	late with ok price
	SL1245	Shree_Lab	9089099890	vellapur	T30789	2200	FA89345	17	Expensive and late
	SL2897	scratch	9089717171	Ganeshnagar	T30789	3500	FA12330	14	On time with ok price
	SL3098	Shubh	7667566565	khamakadi	T30789	3500	FA12340	16	Expensive and late
	SL5676	Yeris	9876866555	sukunpura	T30789	2000	FA12341	12	On time and expensive

2. This query gives farmer's current season's wheat production details, price_status and give the output in the ascending order of expected_price. This query will be used by crop_buyer.

➔ SELECT current_season_crop.F_ID,First_name, Contact_No,
Expected_production, Expected_price,

CASE WHEN Expected_price>220 THEN 'expensive'

WHEN Expected_price<=220 THEN 'affordable'

END AS price_status

FROM farmer, current_season_crop

WHERE state='Gujarat' AND Expected_production>200 AND
crop_ID='WH123' AND crop_delivery_month='July' AND
crop_delievery_year='2000' AND
current_season_crop.F_ID=farmer.F_ID

ORDER BY(Expected_price) ASC

LIMIT 5;

Output:

F_ID	First_name	Contact_No	Expected_production	Expected_price	price_status
FA45340	Yogesh	9676788888	560	220	affordable
FA89345	Shravan	5553455355	686	220	affordable
FA23329	John	2672762772	300	230	expensive
FA67341	Girish	3323232133	546	240	expensive
FA34330	Rickon	2231313333	780	250	expensive

3. This query gives farmer's list fulfilling some conditions about their last season crop details. This query can be used for analysis purpose.

➔ SELECT farmer.F_ID,First_name, Contact_No,Address,
previous_season_crop.production,state,district

FROM farmer,previous_season_crop

WHERE sowing_month='January' AND harvest_month='May' AND
sowing_year=2000

AND harvest_year=2000 AND previous_season_crop.crop_ID='WH123'
AND previous_season_crop.F_ID=farmer.F_ID
AND production= ANY (SELECT production
FROM previous_season_crop
WHERE production
BETWEEN 200.0 AND 500.0)

ORDER BY(production) ASC;

Output:

F_ID	First_name	Contact_No	Address	production	state	district
FA12330	Dinesh	7878676567	Vedalampur	220	Tamilnadu	Chennai
FA12329	Aryan	9898764646	hurrey_nagar	250	Gujrat	Banasakantha
FA12340	vijay	9898709009	saurav_zone	300	Bengal	kolakata
FA12341	baageshwar	9023432321	famous_zone	400	Gujarat	Mehasana
FA12345	Suresh	7089763553	Gurukrupa_nagar	500	Gujarat	Himmatnagar

4. This query gives detail about those fertilizer dealer selling fertilizer than normal price and having stock more than normal.

➔ SELECT fertilizer_dealer.Fd_ID, Contact_No, First_name,
fertilizer_availability.Price, Available_stock, District, Area_Name

FROM fertilizer_dealer, fertilizer_availability

WHERE fertilizer_availability.Fertilizer_ID='FE1232'

AND EXISTS (SELECT price
FROM fertilizer_availability

UNION

SELECT Available_stock
FROM fertilizer_availability

WHERE price>500 AND Available_stock>=20) AND
fertilizer_availability.Fd_ID= fertilizer_dealer.Fd_ID AND state='gujarat'

ORDER BY (fertilizer_availability.Price) ASC;

Output:

Fd_ID	Contact_No	First_name	Price	Available_stock	District	Area_Name
FD2312	2672727722	Ajey	5000	30	Itnagar	Itpur
FD1234	6754626626	Rohit	5500	20	Mehsana	haripur
FD4562	2727888288	Haresh	6000	20	Sirsa	sirspur
FD6532	2626272828	Jenny	6600	20	Batinda	khubpur
FD8267	2771655554	Mahesh	7000	40	Gandhinagar	gandhinagar

Normalization

Table: farmer

F_ID(PK)	Password	First_name	Last_name	state	district	Registration_date
Birth_date	Contact_No	Yearly_income				

FD	Minimal cover
F_ID → First_name	F_ID → First_name
F_ID → Last_name	F_ID → Last_name
F_ID → State	F_ID → State
F_ID → Birth_date	F_ID → Birth_date
F_ID → Contact_no	F_ID → Contact_no
F_ID → Registration_date	F_ID → Registration_date
F_ID → Yearly_income	F_ID → Yearly_income

Candidate Key:- F_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: farm

F_ID	Farm_ID (PK)	Area_of_farm
------	--------------	--------------

FD	Minimal cover
Farm_ID → F_ID	Farm_ID → F_ID
Farm_ID → Area	Farm_ID → Area

Candidate Key:- Farm_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: crop_info

crop_ID (PK)	crop_name
--------------	-----------

FD	Minimal cover
crop_ID → crop_name	crop_ID → crop_name

Candidate Key:- crop_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: current_season_crop

F_ID (PK)	crop_ID	Farm_ID (PK)	Expected_production	Expected_price
sowing_Month (PK)	harvest_month (PK)	crop_delivery_month		

FD	Minimal cover
Farm_ID → F_ID Farm_ID → crop_ID Farm_ID → sowing_month Farm_ID → sowing_year Farm_ID → harvest_month Farm_ID → harvest_year	Farm_ID → F_ID Farm_ID → crop_ID Farm_ID → sowing_month Farm_ID → sowing_year Farm_ID → harvest_month Farm_ID → harvest_year

Candidate Key:- Farm_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: crop_buyer

CB_ID (PK)	Password	First_name	Last_name	state	district	Registration_date
Birth_date	Ratings	Yearly_income	Total_deals_done			

FD	Minimal cover
CB_ID → Ratings CB_ID → Total_deals_done CB_ID → First_name CB_ID → Last_name CB_ID → State CB_ID → Birth_date CB_ID → Contact_no CB_ID → Registration_date CB_ID → Yearly_income	CB_ID → Ratings CB_ID → Deals_done CB_ID → First_name CB_ID → Last_name CB_ID → State CB_ID → Birth_date CB_ID → Contact_no CB_ID → Registration_date CB_ID → Yearly_income

Candidate Key:- CB_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: crop_buyer_need

CB_ID(PK)	crop_id	Needed_quantity	Affordable_Price	Deadline_month
-----------	---------	-----------------	------------------	----------------

FD	Minimal cover
(CB_ID)(crop_ID) → needed_quantity	(CB_ID)(crop_ID) → needed_quantity
(CB_ID)(crop_ID) → Affordable_price	(CB_ID)(crop_ID) → Affordable_price
(CB_ID)(crop_ID) → Deadline_month	(CB_ID)(crop_ID) → Deadline_month
(CB_ID)(crop_ID) → Deadline_year	(CB_ID)(crop_ID) → Deadline_year

Candidate Key:- (CB_ID)(crop_ID)**Normalization:**

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: deals_with

F_ID(PK)	CB_ID (PK)	crop_ID (PK)	quantity_of_crop	payment_type	Amount_paid	deal_time (PK)
----------	---------------	-----------------	------------------	--------------	-------------	-------------------

FD	Minimal cover
(CB_ID)(crop_ID)(F_ID)(deal_time) → quantity_of_crop	(CB_ID)(crop_ID)(F_ID)(deal_time) → quantity_of_crop
(CB_ID)(crop_ID)(F_ID)(deal_time) → payment_type	(CB_ID)(crop_ID)(F_ID)(deal_time) → payment_type
(CB_ID)(crop_ID)(F_ID)(deal_time) → Amount_paid	(CB_ID)(crop_ID)(F_ID)(deal_time) → Amount_paid

Candidate Key:- (CB_ID)(crop_ID)(F_ID)(deal_time)

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: soil analytical lab

Lab_ID (PK)	Password	Lab_Name	Contact_NO	state	district	Pin_code	opening_time
closing_time							

FD	Minimal cover
Lab_ID → Lab_name	Lab_ID → Lab_name
Lab_ID → Contact_NO	Lab_ID → contact_no
Lab_ID → state	Lab_ID → state
Lab_ID → district	Lab_ID → district
Lab_ID → opening_time	Lab_ID → opening_time
Lab_ID → closing_time	Lab_ID → closing_time

Candidate Key:- Lab_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: test

Test_name	Test_ID (PK)	Price
-----------	--------------	-------

FD	Minimal cover
Test_ID → Test_name	Test_ID → Test_name

Candidate Key:- Test_ID**Normalization:**

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: test_info

Lab_ID(PK)	Test_ID (PK)	Test_price	Time_to_complete_test
------------	--------------	------------	-----------------------

FD	Minimal cover
(Test_ID)(Lab_ID) → Test_price (Test_ID)(Lab_ID) → Days_for_test	(Test_ID)(Lab_ID) → Test_price (Test_ID)(Lab_ID) → Days_for_test

Candidate Key:- (Test_ID)(Lab_ID)

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: provides soil analysis

Test_ID(PK)	payment_type	amount_paid	sample_given_date (PK)
Report_recieved_date	date_of_payment	Lab_ID (PK)	F_ID (PK)

FD	Minimal cover
(Test_ID)(sample_given_date) (Lab_ID)(F_ID) → (Payment_type)(amount) (Report_received_date)	(Test_ID)(sample_given_date) (Lab_ID)(F_ID) → (Payment_type)(amount) (Report_received_date)

Candidate Key:- (Test_ID)(sample_given_date)(Lab_ID)(F_ID)

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: fertilizer_dealer

Fd_ID(PK)	First_name	Last_name	State	District	Pin_code	Yearly_Income	Contact_No
-----------	------------	-----------	-------	----------	----------	---------------	------------

FD	Minimal cover
FD_ID → First_name FD_ID → Last_name FD_ID → State FD_ID → Birth_date FD_ID → Contact_no FD_ID → Registration_date FD_ID → Yearly_income	FD_ID → First_name FD_ID → Last_name FD_ID → State FD_ID → Birth_date FD_ID → Contact_no FD_ID → Registration_date FD_ID → Yearly_income

Candidate Key:- FD_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: fertilizer

Fertilizer_ID (PK)	Fertilier_name
--------------------	----------------

FD	Minimal cover
Fertilizer_ID → Fertilizer_name	Fertilizer_ID → Fertilizer_name

Candidate Key:- Fertilizer_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: fertilizer_availability

Fd_ID (PK)	Fertilizer_ID (PK)	Price	Available_stock
------------	--------------------	-------	-----------------

FD	Minimal cover
(FD_ID)(Fertilizer_ID) → Price (FD_ID)(Fertilizer_ID) → Available_stock	(FD_ID)(Fertilizer_ID) → Price (FD_ID)(Fertilizer_ID) → Available_stock

Candidate Key:- (FD_ID)(Fertilizer_ID)**Normalization:**

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: fertilizer_deal

F_ID(PK)	Fd_ID (PK)	Quantity_of_Fertilizer	Amount_paid	Fertilizer_ID (PK)	Payment_type	Deal_time (PK)
----------	------------	------------------------	-------------	--------------------	--------------	----------------

FD	Minimal cover
(F_ID)(FD_ID)(Fertilizer_ID) (Deal_time)→ (Amount_paid)(Payment_type) (quantity_of_fertilizer)	(F_ID)(FD_ID)(Fertilizer_ID) (Deal_time)→ (Amount_paid)(Payment_type) (quantity_of_fertilizer)

Candidate Key:- (F_ID)(FD_ID)(Fertilizer_ID)(Deal_time)

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: irrigation system

Farm_ID (PK)	temperature	humidity	methane_amount
--------------	-------------	----------	----------------

FD	Minimal cover
Farm_ID → temperature Farm_ID → humidity Farm_ID → methane_amount	Farm_ID → temperature Farm_ID → humidity Farm_ID → methane_amount

Candidate Key:- Farm_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: transportation_provider

First_name	Last_name	TP_ID (PK)	Registration_date	Ratings	Availability
location_latitude	location_longitude				

FD	Minimal cover
TP_ID → First_name TP_ID → Last_name TP_ID → Registration_date TP_ID → Ratings TP_ID → Availability TP_ID → location_latitude TP_ID → location_longitude	TP_ID → First_name TP_ID → Last_name TP_ID → Registration_date TP_ID → Ratings TP_ID → Availability TP_ID → location_latitude TP_ID → location_longitude

Candidate Key:- TP_ID

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: transportation_vehicle

TP_ID	Rc_book_number	Vehicle_number (PK)
-------	----------------	---------------------

FD	Minimal cover
Vehicle_number → TP_ID Vehicle_number → Vehicle_model Vehicle_number → Rc_book_number	Vehicle_number → TP_ID Vehicle_number → Vehicle_model Vehicle_number → Rc_book_number

Candidate Key:- Vehicle_number

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Table: circulation deal

CB_ID (PK)	TP_ID (PK)	Source_latitude	Source_longitude	Destination_latitude	Destination_longitude
item	Payment_type	Amount_Paid	Time_of_payment (PK)		

FD	Minimal cover
(CB_ID)(TP_ID)(Time_of_payment) → Source	(CB_ID)(TP_ID)(Time_of_payment) → Source
(CB_ID)(TP_ID)(Time_of_payment) → Destination,item	(CB_ID)(TP_ID)(Time_of_payment) → Destination,item
(CB_ID)(TP_ID)(Time_of_payment) → Payment_type	(CB_ID)(TP_ID)(Time_of_payment) → Payment_type
(CB_ID)(TP_ID)(Time_of_payment) → Amount_Paid	(CB_ID)(TP_ID)(Time_of_payment) → Amount

Candidate Key:- (CB_ID)(TP_ID)(Time_of_payment)

Normalization:

1NF → Yes

2NF → Yes

3NF → Yes

BCNF → Yes

Assignment Topic

(Stored Procedures)

1. This query will merge dataset of farmer's current and past season's crop state wise.

```
CREATE PROCEDURE `statewise_farmer_detail`(IN current_cropID
VARCHAR(10),
IN previous_cropID VARCHAR(10),IN state_ VARCHAR(25))
BEGIN
SELECT farmer.First_name, current_season_crop.crop_ID as current_crop_ID,
previous_season_crop.crop_ID as
previous_crop_ID,current_season_crop.Expected_production,
previous_season_crop.production as produced_crop
FROM ((farmer INNER JOIN current_season_crop ON farmer.F_ID =
current_season_crop.F_ID) INNER JOIN previous_season_crop ON
farmer.F_ID =
previous_season_crop.F_ID) WHERE current_season_crop.crop_ID=
current_cropID
AND previous_season_crop.crop_ID= previous_cropID AND state=state_;
END
CALL statewise_farmer_detail ('gujarat')
```

Output:

	First_name	current_crop_ID	previous_crop_ID	Expected_production	produced_crop
►	John	WH123	RI405	300	350
	Rickon	WH123	RI405	780	750
	Yogesh	WH123	RI405	560	580
	Girish	WH123	RI405	546	550
	Shravan	WH123	RI405	686	700

2. This query will give the crop buyer detail with their experience status based on their ratings and total deals done.

```
CREATE PROCEDURE `Experienced_crop_buyer`(IN state_
VARCHAR(25), IN District_ VARCHAR(25))
BEGIN
SELECT CB_ID, First_Name,contact_No, district,Ratings>Total_deals_done,
CASE WHEN Total_deals_done>=500 THEN 'experienced'
WHEN Total_deals_done>=300 AND Total_deals_done<500 THEN 'quite
experienced'
ELSE 'not_experienced'
END AS experience_status FROM crop_buyer
WHERE state='gujarat' AND crop_buyer.CB_ID= ANY (SELECT
crop_buyer.CB_ID FROM
crop_buyer WHERE Total_deals_done>200) AND state=state_ AND
district=district_
ORDER BY(Total_deals_done);
END
```

Output:

	CB_ID	First_Name	contact_No	district	Ratings	Total_deals_done	experience_status
►	CB45345	Mahesh	9888477444	Banaskantha	5	300	quite experienced
	CB23345	Prem	9448373484	Banaskantha	5	400	quite experienced
	CB32345	Rahul	9439234776	Banaskantha	5	500	experienced
	CB12345	Ajay	90191919	Banaskantha	4	600	experienced
	CB56345	Haresh	8787788998	Banaskantha	4	700	experienced

3. This query will give average price and average stalk available per fertilizer dealer per state.

```
CREATE PROCEDURE `fertilizer_detail`(IN state_ VARCHAR(25),  
IN Fertilizer_ID_ VARCHAR(10))  
BEGIN  
SELECT AVG(fertilizer_availability.Price) As Avg_price,  
AVG(Available_stock) As Avg_available_stock  
FROM fertilizer_availability, fertilizer_dealer, fertilizer  
WHERE fertilizer.Fertilizer_ID=  
fertilizer_availability.Fertilizer_ID=Fertilizer_ID_  
AND Fertilizer_dealer.state=state_;  
END  
CALL fertilizer_detail('gujarat','FE1232')
```

Output:

Result Grid			Filter Rows:		Exp
	Avg_price	Avg_available_stock			
▶	6020	26			

4. This query will give list of top fertilizer dealer in particular state on the basis of quantity of fertilizer they have sold.

```
CREATE PROCEDURE `top_fertilizer_dealers`(IN State_ VARCHAR(25))
BEGIN
SELECT fertilizer_dealer.Fd_ID, Contact_No, First_name, District
FROM fertilizer_dealer,fertilizer_deal
WHERE fertilizer_deal.Fertilizer_ID='FE1232' AND
EXISTS (SELECT Quantity_of_Fertilizer FROM fertilizer_deal WHERE
Quantity_of_Fertilizer>190 ) AND fertilizer_dealer.state=state_
AND fertilizer_dealer.Fd_ID= fertilizer_deal.Fd_ID
ORDER BY (Amount_paid) ASC;
END
CALL top_fertilizer_dealers('gujarat')
```

Output:

Fd_ID	Contact_No	First_name	District
FD8267	2771655554	Mahesh	Gandhinagar
FD1234	6754626626	Rohit	Mehsana
FD6532	2626272828	Jenny	Batinda
FD4562	2727888288	Haresh	Sirsa
FD2312	2672727722	Ajey	Itanagar


Research Publication

(Book Chapter)

We have submitted the book chapter of “Agrisociety Network Management System” on easy chair.

Snapshots of submission page and mail are given below:

The submission has been saved!

Submission 12	
Title:	Agrisociety network management system
Paper:	 (Dec 09, 08:54 GMT)
Author keywords:	price rise controller integrated system connection maker hoarding protector
Abstract:	Algorithm of this project is based on "Larger a network, more the opportunities". In abstract this project will lead us to a platform that will provide more opportunities for agricultural growth by developing a proper business system and avoiding problems like price rise in a society. On our platform farmers or any professional related to agriculture business people will get good profit for their efforts. Hoarding will be less because of transparency provided by the platform. This platform will be "Price rise controller of basic needs" as sellers have to show their price details, and due to competition on platform they will not be able to afford consumer loss and so they will not do unnecessary price rise. This platform as a agribusiness network will connect around 53% to 60% population of agrarian country like India. The platform can be helpful to make organized, transparent and efficient agricultural cycle in India.
Submitted:	Dec 09, 08:54 GMT
Last update:	Dec 09, 08:54 GMT

Authors						
first name	last name	email	country	affiliation	Web page	corresponding?
Bhavik	Bhatt	bhavik.bict18@sot.pdpu.ac.in	India	Pandit Deendayal Petroleum University		✓
Aaryan	Satpal	aaryan.sict18@sot.pdpu.ac.in	India	Pandit Deendayal Petroleum University		✓
Jaimin	Rana	jaimin.rict18@sot.pdpu.ac.in	India	pandit deendayal petroleum university		✓
Manan	Bharwad	manan.bict18@sot.pdpu.ac.in	India	pandit deendayal petroleum university		✓
Nishant	Doshi	Nishant.Doshi@sot.pdpu.ac.in	India	pandit deendayal petroleum university		✓

IECE 2021 submission 12 Inbox x



IECE 2021 <iece2021@easychair.org>

to me ▾

Dear authors,

We received your submission to IECE 2021 (Internet of Energy :
Connecting Energy Anywhere Anytime):

Authors : Bhavik Bhatt, Aaryan Satpal, Jaimin Rana, Manan Bharwad and **Nishant Doshi**

Title : Agrisociety network management system

Number : 12

The submission was uploaded by Bhavik Bhatt
<bhavik.bict18@sot.pdpu.ac.in>. You can access it via the IECE 2021
EasyChair Web page

<https://easychair.org/conferences/?conf=iece2021>

Thank you for submitting to IECE 2021.

Best regards,
EasyChair for IECE 2021.

Our Contribution

- Before we started our work for project, we decided that we will work in a manner that each member will get to know about every corner of project and will be able to learn every topic attached to the DBMS project.
 - So during the project, there were some topics in which, we simply divided our work into four parts, those tasks were:
 1. DBMS Vs File system examples
 2. SQL queries
 3. Relational Algebra
 4. Extra topic (Stored Procedure)
 - But there were some tasks in which we distributed work table wise.
 - Tasks like:
 1. Creating ER diagram
 2. Creating schema diagram
 3. Creating relational diagram
 4. create table commands
- ❖ During the project when the task was oriented or related to table/s, members completed tasks according to the tables they selected. Here, members and their selected tables' information is given:

Member Name	Allocated table Name
Bhavik	'Farmer', 'Farm', 'Crop_info', 'Current_season_crop' 'Crop_buyer', 'Crop_buyer_need'
Aaryan	'Lab_owner', 'soil_analytical_lab', 'provides_soil_analysis', 'test', 'test_info'
Jaimin	'circulation_deal', 'deals_with', 'transportation_provider', 'transportation_vehicle'
Manan	'Fertilizer', 'Fertilizer_dealer', 'Fertilizer_deal', 'Fertilizer_availability', 'Irrigation_system'

- In tasks like stage 1, stage 2 and research paper, it was never like one person is doing all the tasks, we made sure that each member can know everything going on in tasks, so we made several meetings, during these tasks and made sure that all members can contribution in every possible way.

- So in overall project, we can say that work done by each team member was neither more nor less and each member contributed equally.
- ➔ Throughout the project, we have divided the complex work like making queries for the project. And for the book chapter, we distributed topics among the team members.
- ➔ We divided each and every task considering the unique skills of each team member.

Member Name	Work(Stage-2)
Bhavik	Stored procedure, My SQL queries
Aaryan	MY SQL QUERIES, Algebra queries
Jaimin	Algebra queries, Stored procedure
Manan	Algebra queries, My SQL queries

Member Name	Work(Book chapter)
Bhavik	Proposed work & Feature description of project
Aaryan	Literature survey and analysis of the papers
Jaimin	Introduction and abstraction of project
Manan	Query Analysis and Conclusion of project

- How we added IOT devices to our database?

➔ In our sensor based irrigation system we embedded moisture sensor, temperature sensor and methane sensor.

➔ The reading those will be shown by these sensors will be stored in table named Irrigation_system.

➔ And the developers will take the data from tables like farm_area from farm table, moisture, temperature and methane information from irrigation_system table and growing crop info from farm table and the needed levels of element will be compared with the existing and the call like watering crop will be taken.

➔ Here is some information about our IOT devices:

IOT device name	In which table it is?
Moisture sensor	irrigation_system
Temperature sensor	irrigation_system
Methane sensor	irrigation_system

IOT device name	What it does?
Moisture sensor	Shows moisture level
Temperature sensor	Shows temperature
Methane sensor	Methane element level