



AgriSociety Management System

Team 30

Team Members

Bhavik Bhatt(18BIT010)

Aaryan Satpal(18BIT001)

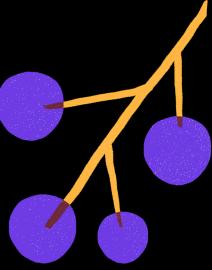
Jaimin Rana(18BIT036)

Manan Bharwad(18BIT061)



Product Flow Control

Most importantly this platform is being made by taking care of sensitive issues like hoardings, as here production from farmers can be analysed and on the basis of that data administrator can find if produced crops are being circulated in proper way or not, and if not than circulation can be traced and further actions can be taken.



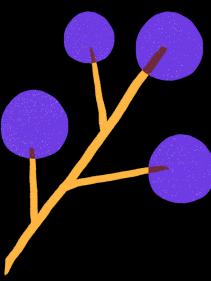
Compatible Data for Business

Information taken from users will be in such a way that their current needs will be shown on platform so if users can fulfil each other needs in profitable way they will communicate with each other.

Unique Features

Provides Integrated Agribusiness

Platform is from small professions to bigger factories, it sets proper path for agribusiness, and provides feature of integrity of scattered agribusiness.



Sensor based Irrigation System

Will manage to water crops by Methane Sensor, Humidity Sensor, Temperature Sensor, PCLs and RTU based electronic system.



Schema Diagram:

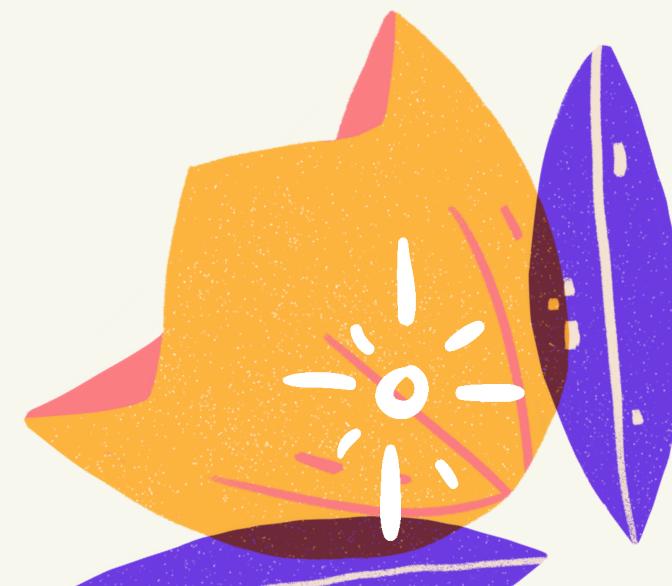
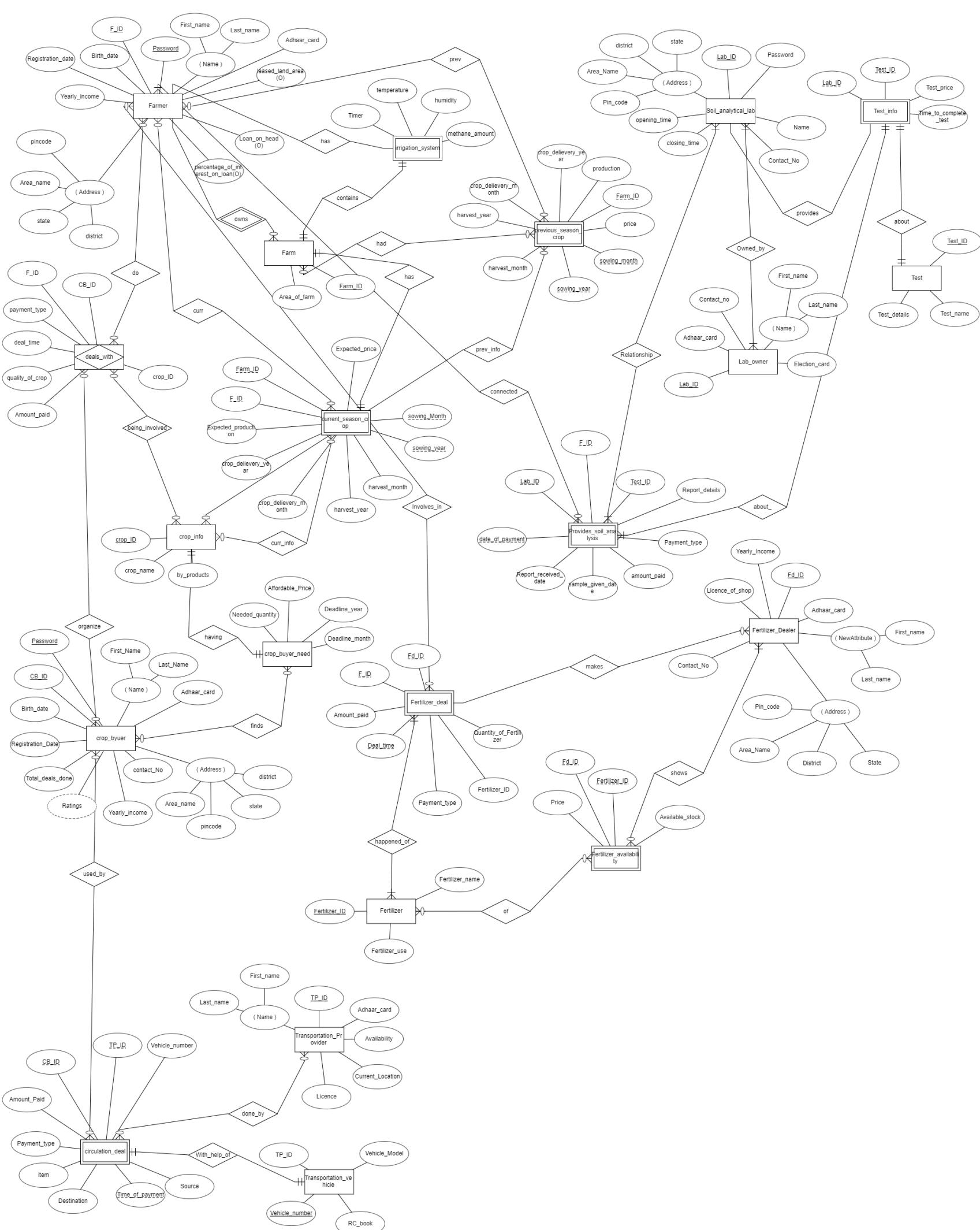
- can be represented in a visual diagram
- shows the database objects
- shows objects' relationship with each other.

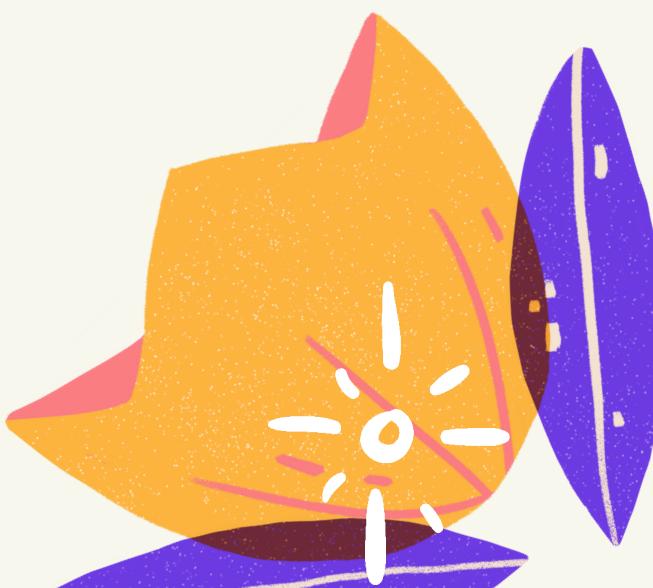
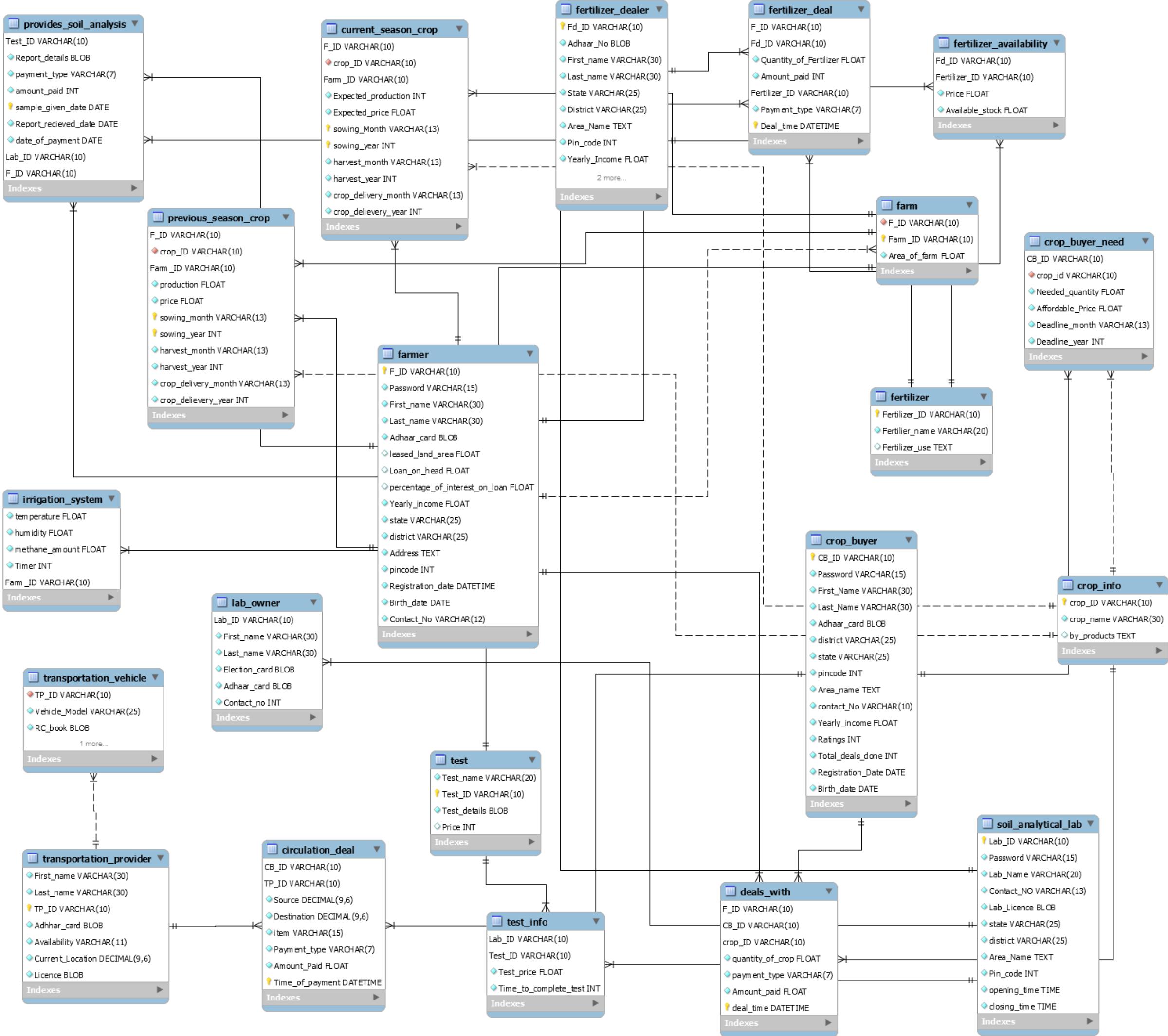


Entity-Relationship Diagram:

- is a type of flowchart that illustrates how “entities” are related to each other







SQL Queries

SQL Operators

SELECT:
It is used to select data from a database.

WHERE:
It is used to extract only those records that fulfill a specified condition.

AND:
It displays a record if all the conditions separated by AND are TRUE.

ORDER BY:
It is used to sort the result-set in ascending or descending order.

COUNT:
It returns the number of rows that matches a specified criterion.

BETWEEN:
It selects values within a given range.

INNER JOIN:
IT selects records that have matching values in both tables.

UNION:
IT is used to combine the result-set of two or more SELECT statements.

QUERY 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

QUERY 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_delivery_year='2000' AND  
current_season_crop.F_ID=farmer.F_ID  
ORDER BY(Expected_price) ASC  
LIMIT 5;
```



Output

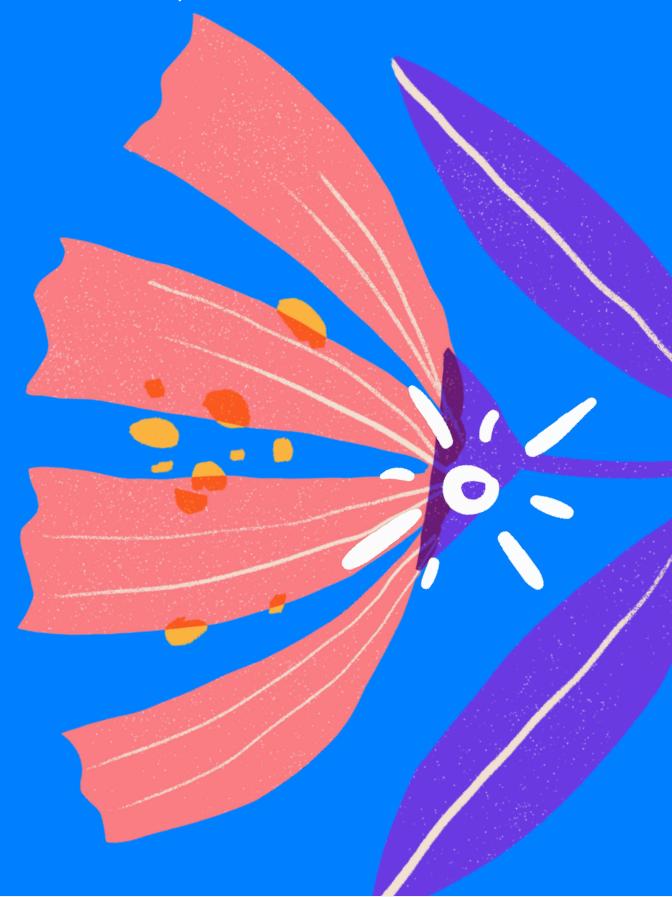


F_ID	First_name	Contact_No	Expected_production	Expected_price	price_status
FA45340	Yogesh	9676788888	560	220	affordable
FA89345	Shravan	5553455355	686	220	affordable
FA23329	John	2672762772	300	230	expensive
FA67341	Girish	3323232133	546	240	expensive
FA34330	Rickon	2231313333	780	250	expensive

QUERY 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	Vedalamppur	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

QUERY 4 This query gives detail about those fertilizer dealer selling fertilizer than normal price and having stalk 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

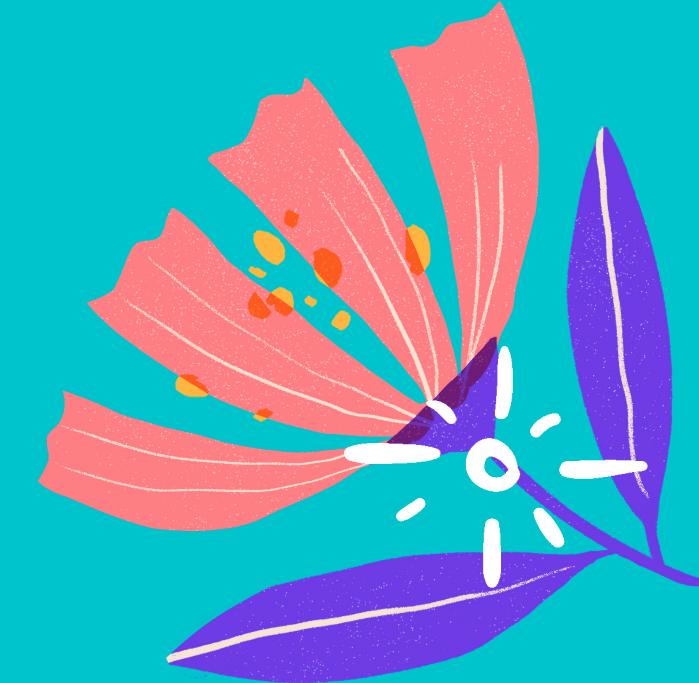
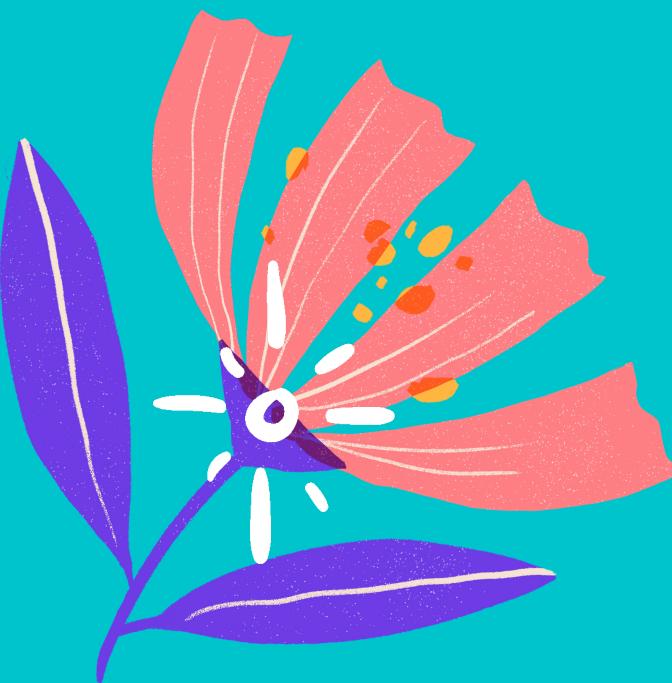
Fd_ID	Contact_No	First_name	Price	Available_stock	District	Area_Name
FD2312	2672727722	Ajey	5000	30	Itnagar	Itpur
FD1234	6754626626	Rohit	5500	20	Mehsana	haripur
FD4562	2727888288	Haresh	6000	20	Sirsa	sirspur
FD6532	2626272828	Jenny	6600	20	Batinda	khubpur
FD8267	2771655554	Mahesh	7000	40	Gandhinagar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

Queries References and content involved in them:

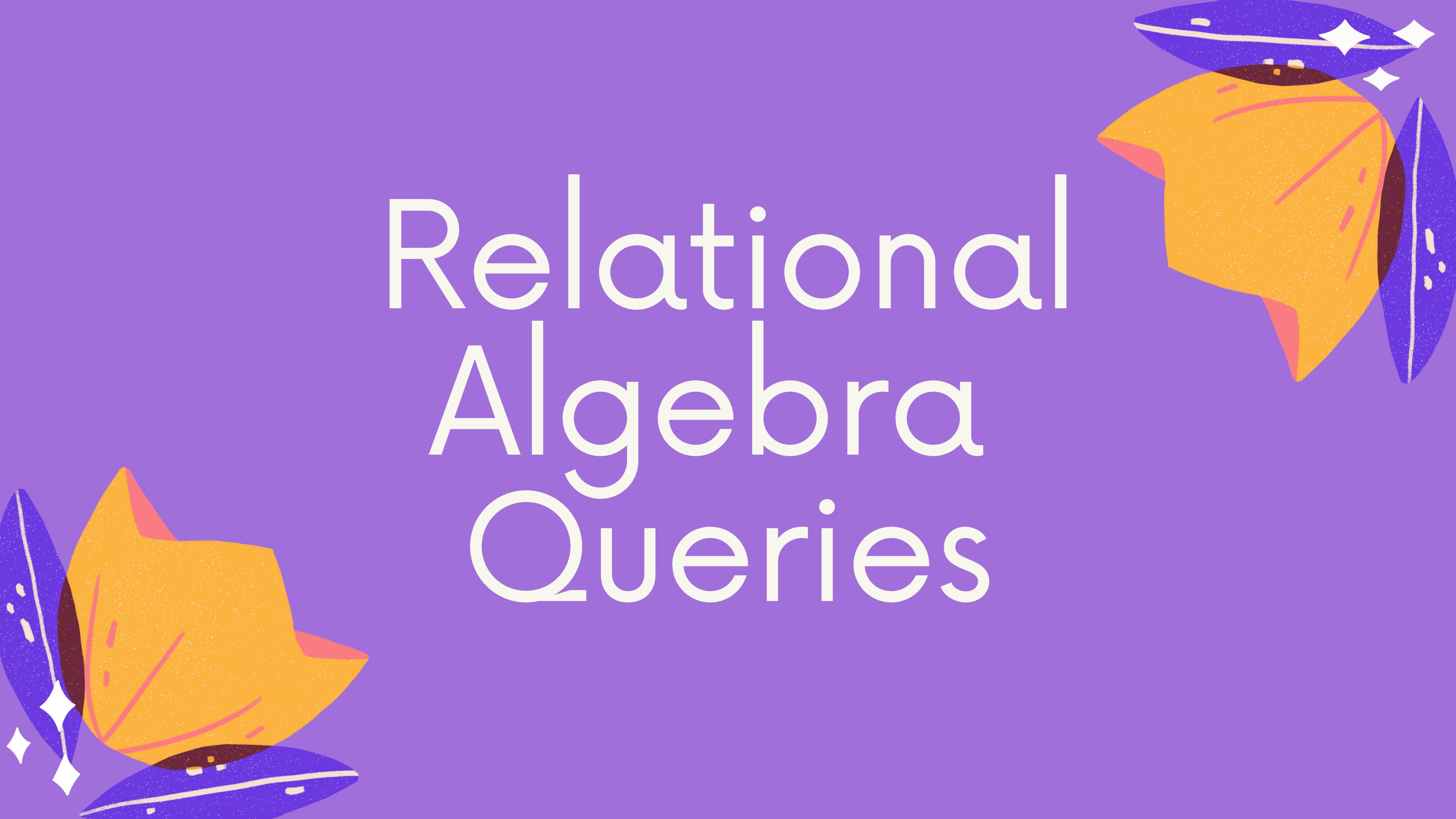
All the SQL queries selected or made by our team were based on real life Scenarios.

There were some references in making of these SQL queries:
www.w3schools.com
www.javapoint.com

Queries have been run on:
MySQL workbench
Oracle Live SQL



Relational Algebra Queries

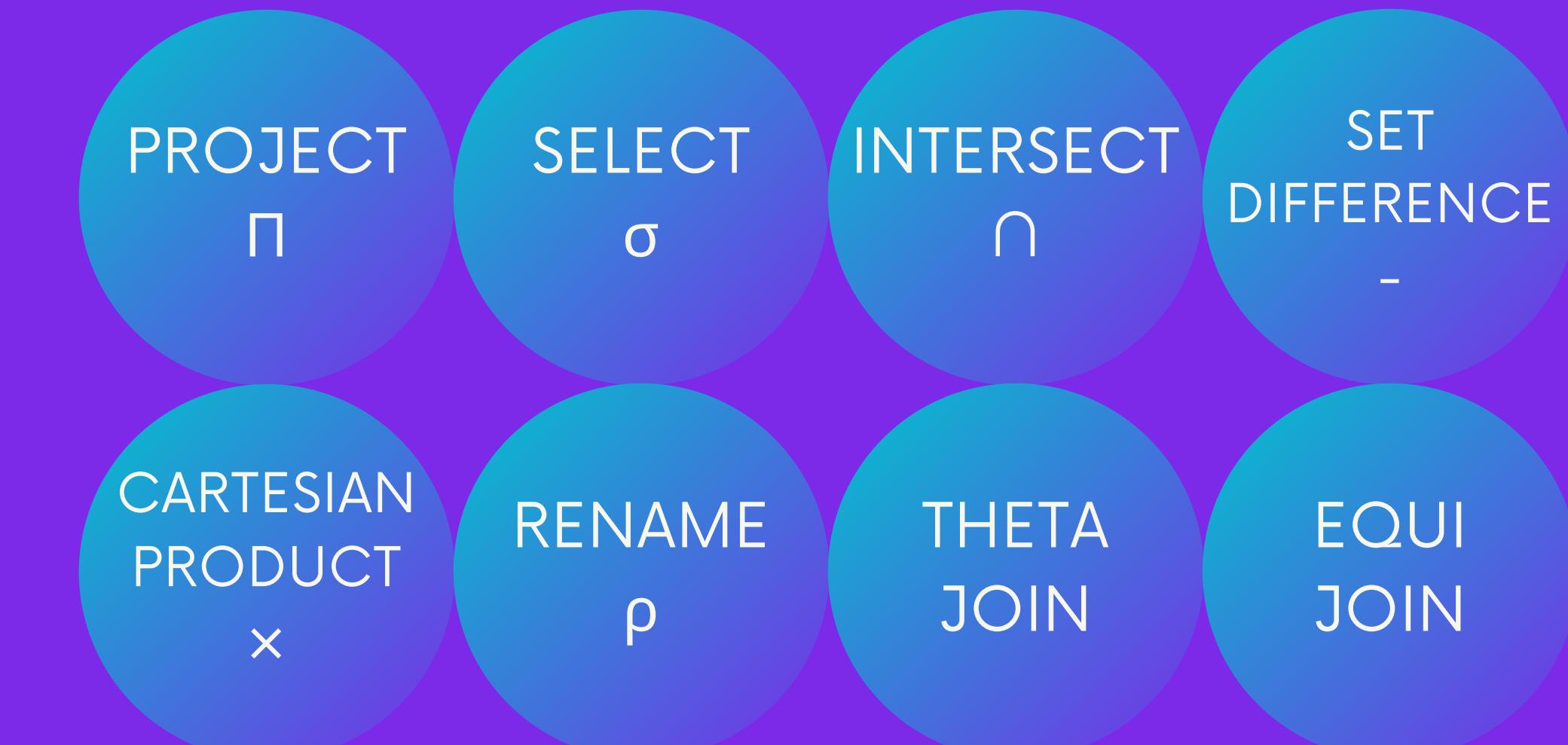


Introduction:

- Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output and uses operators to perform queries. An operator can be either unary or binary. They accept relation as their input and yield relations as output. It is performed recursively on a relations & intermediate results are also considered relations.



The Fundamental Operations



QUERY 1

This query will return Crop buyers(CB_ID) having maximum ratings and are expecting the crop(crop_ID) that farmer sow in current season(crop_ID).

$\Pi_{CB_ID}[\Pi_{CB_ID, Ratings}(crop_buyer) \times \Pi_{CB_ID, Ratings}(crop_buyer)]$

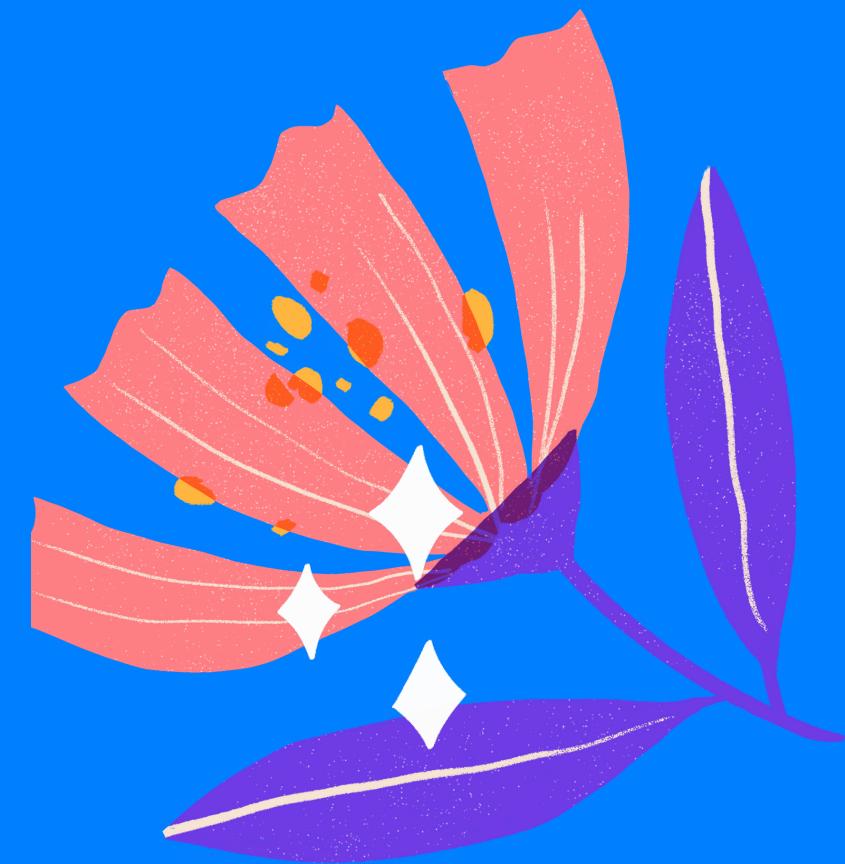
$- \sigma_{Ratings < Ratings2}(\Pi_{CB_ID, Ratings}(crop_buyer))$

$\times \rho_{Ratings2/Ratings}(\Pi_{CB_ID, Ratings}(crop_buyer))]$

$\cap \Pi_{CB_ID}[\Pi_{CB_ID, crop_ID}(crop_buyer_need)$

$\bowtie_{crop_buyer_need.crop_ID = current_season_crop.crop_ID}$

$\Pi_{crop_ID}(current_season_crop)]$



QUERY 2

This query will return The Labs(Lab_ID, Contact_NO), When a farmer searches for labs residing in the same city by Test_name ,Test_price.

$\Pi_{Lab_ID, Contact_NO} [\Pi_{Lab_ID, Contact_NO, Pin_code} (soil_analytical_lab)]$

$\bowtie_{soil_analytical_lab.Pin_code = farmer.pincode} \Pi_{F_ID, pincode} (farmer)]$

$\bowtie \Pi_{Lab_ID} [\Pi_{Lab_ID, Test_ID} (test_info)]$

$\bowtie \Pi (\Pi_{Test_ID} (\sigma_{Test_name = "name"} (test)))$

$\cap \Pi_{Test_ID} (\sigma_{Test_price \leq "price"} (test_info)))]$



QUERY 3

This query will return Farmers' details (F_ID, First_name, Contact_No, district, state) living in "Gujarat" whose current season's crop production is much higher than previous season's crop production for the same crop.

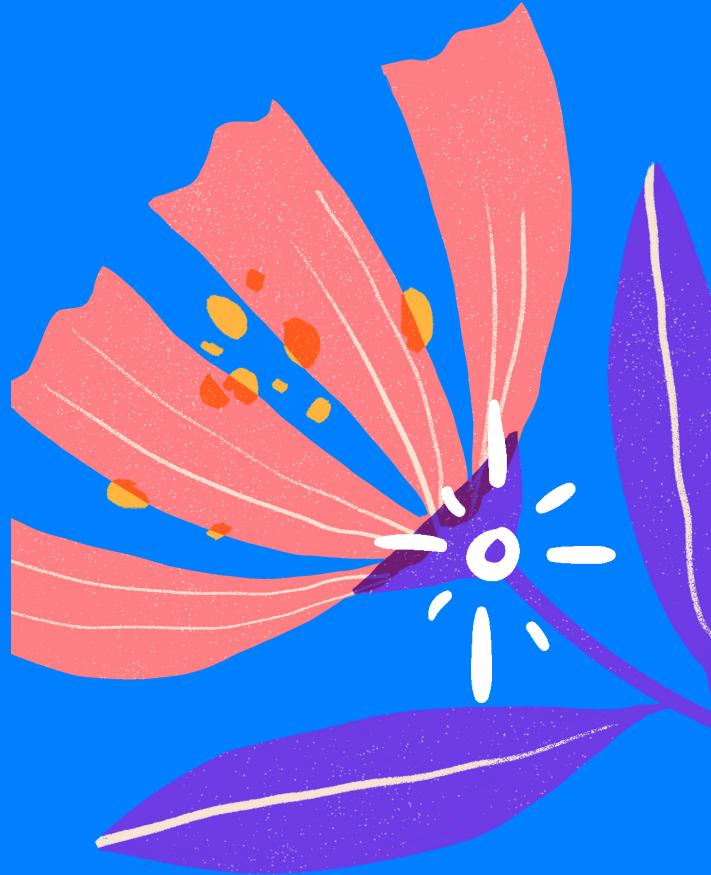
$\sigma_{\text{state}=\text{"Gujarat"} \left[\Pi_{\text{F_ID}, \text{First_name}, \text{Contact_No}, \text{district}, \text{state}} (\text{farmer}) \right]$

$\bowtie \sigma_{\text{current_season_crop.Expected_production} >> \text{previous_season_crop.production}}$

$(\Pi_{\text{F_ID}, \text{crop_ID}, \text{Expected_production}} (\text{current_season_crop})$

$\bowtie \Pi_{\text{F_ID}, \text{crop_ID}, \text{production}} (\text{previous_season_crop}))$

$\bowtie \Pi_{\text{crop_ID}, \text{crop_name}} (\text{crop_info})]$



QUERY 4

This query will return Fertilizer dealers' details (Fd_ID, First_name, Contact_No) whose shop will be in city having pincode=382007 and who has "Urea" fertilizer with available stock>=5.

```
ΠFd_ID,First_name,Contact_No [  
    σPin_code=382007 ( ΠFd_ID,First_name,Pin_code,Contact_No (fertilizer_dealer)  
        ⋈ σAvailable_stock>=5 ( ΠFertilizer_ID (σFertilizer_name="Urea" (fertilizer))  
            ⋈ (fertilizer_availability) ))  
    ]
```



Stored Procedures



QUERY 1 This query will merge the 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

QUERY 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

QUERY 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
```

Output

CALL fertilizer_detail('gujarat','FE1232')

Result Grid | Filter Rows: | Export

	Avg_price	Avg_available_stock
▶	6020	26

QUERY 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
```

Output

CALL top_fertilizer_dealers('gujarat')

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

Contribution of team members:

BHAVIK

- Stored Procedure
- MY SQL queries
- Data Entry

AARYAN

- MY SQL queries
- Algebra queries
- Data Entry

JAIMIN

- Algebra queries
- Stored Procedure
- Relational Model

MANAN

- ER diagram
- Relational mode
- Algebra queries

Team Management



- We used various platforms for communicating with each other.
- We used **Microsoft Teams** for major team meetings.
- We used a **Whatsapp** group and **Google drive** for sharing updates and suggestion of the project.
- To work efficiently, Team meetings and work distribution were managed as per team members' time schedule, their abilities and their knowledge of interest.



Thank You