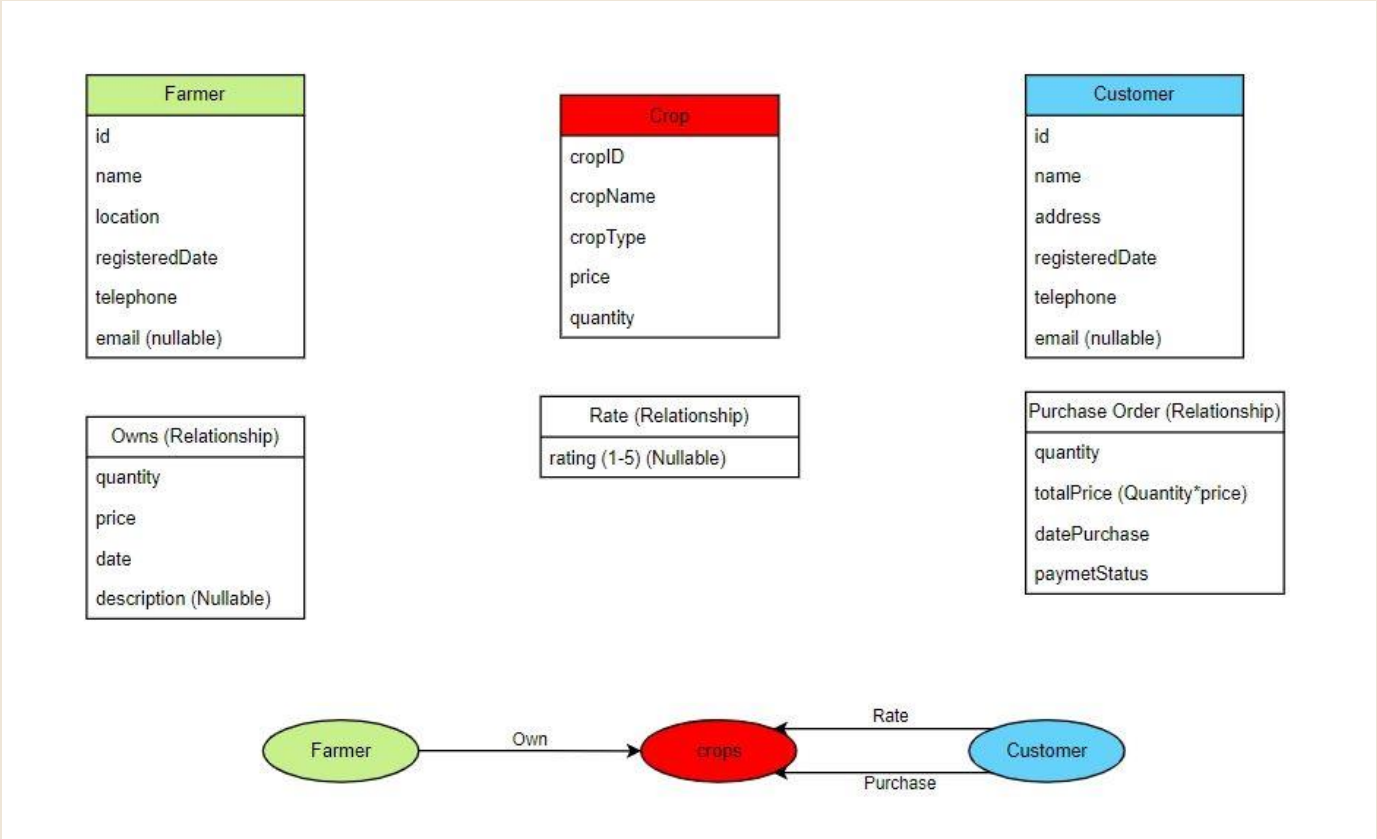
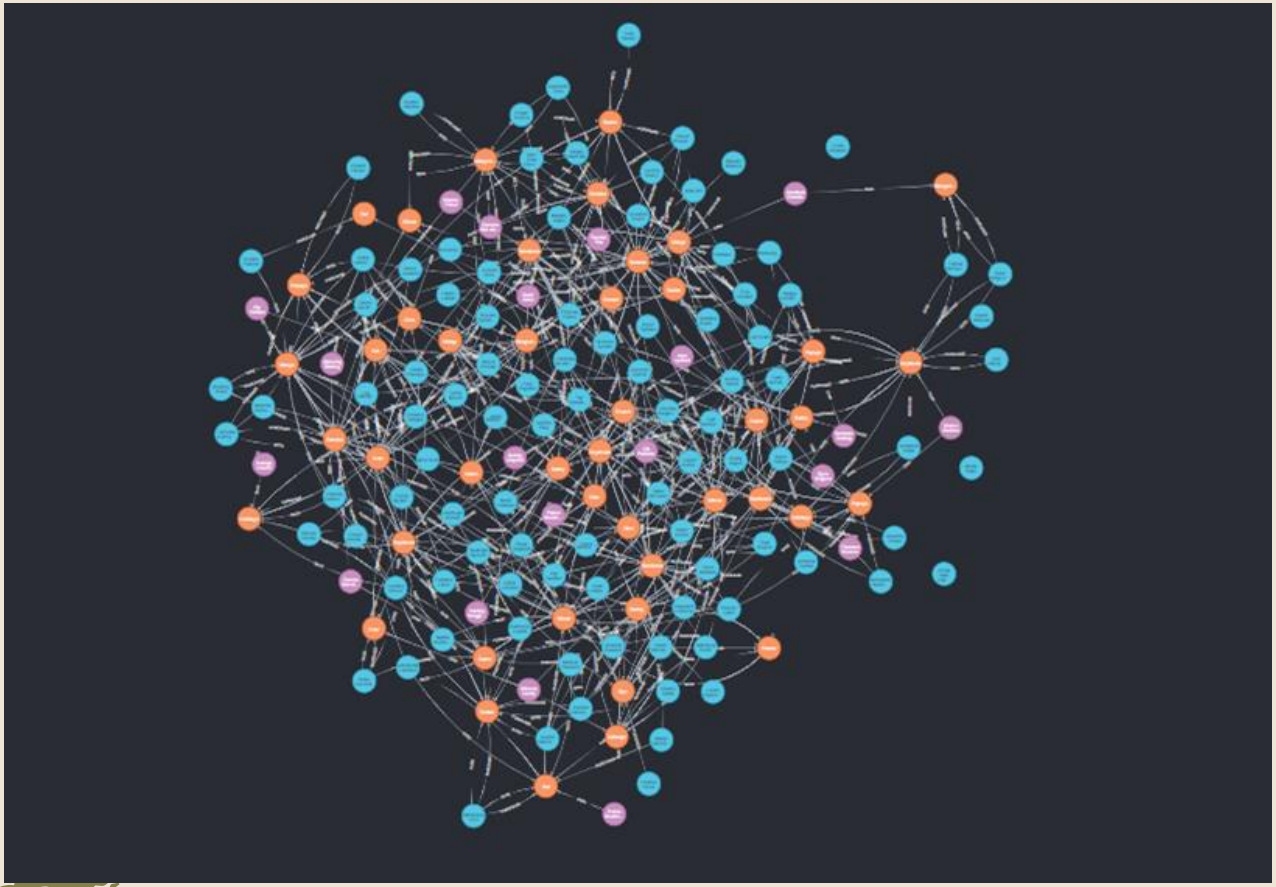
OVERVIEW OF OUR GRAPH MODEL



OVERVIEW OF OUR GRAPH



➢ Overview of our system through some scenarios

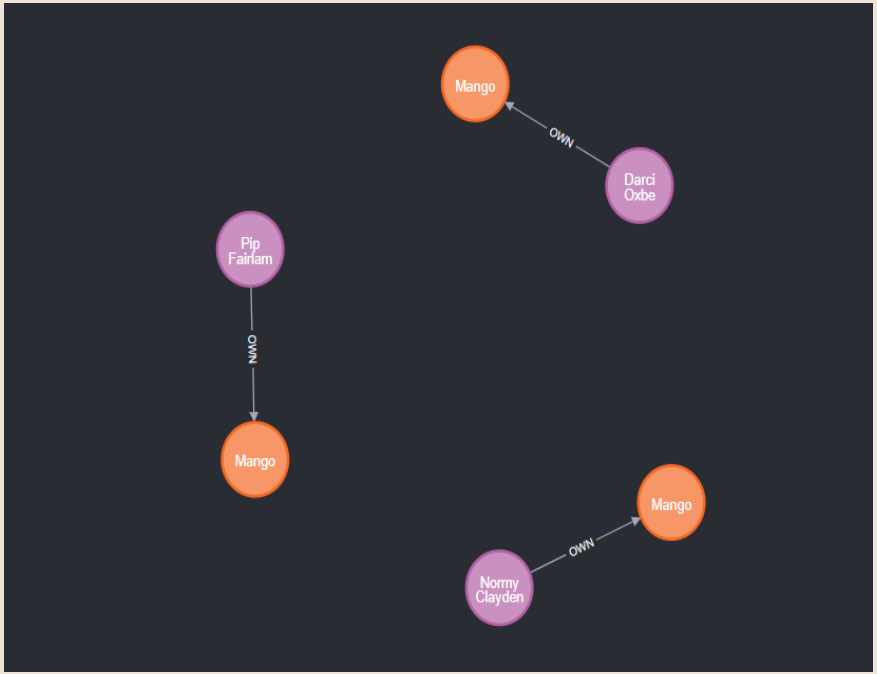
- A customer will look for different crop types stored in the system

```
neo4j$ match(c:Crop) return distinct c.type as CropType;
```

	CropType
1	"cereal"
2	"fruit"
3	"bean"
4	"rice"
5	"vegetable"

- Now, he/she will only look for the Mango (fruit) and related farmers

```
match (f:Farmer)-[o:OWN]->(c:Crop)
where c.name="Mango"
return f, c;
```



- Now he/she will select the farmers with mango and rating >= 3

```
MATCH (f:Farmer)-[:OWN]->(cr:Crop)<-[r:RATE]-(c:Customer)
WHERE cr.name = 'Mango' and r.rating > 3
RETURN f.name, cr.name, r.rating
```

	f.name	cr.name	r.rating
1	"Normy Clayden"	"Mango"	5
2	"Normy Clayden"	"Mango"	5
3	"Normy Clayden"	"Mango"	5
4	"Darci Oxbe"	"Mango"	5
5	"Pip Fairlam"	"Mango"	4
6	"Pip Fairlam"	"Mango"	4

- We assume that farmer 2 sold 20 kg of Barley.

  MATCH (f:Farmer)-[o:OWN]->(cr:Crop), (c:Customer)

  WHERE f.id = 2 AND cr.name = "Barley" AND c.name = "Archibold Fibbit"

  CREATE (cr)<-[p:PURCHASE{datePurchase:date(), paymentStatus: "Done", quantity: 20, totalPrice : o.price * 20}]-(c)

  Return p

  - Right now he has 187 quantity

```
MATCH (f:Farmer)-[o:OWN]->(cr:Crop)
WHERE f.id=2 and cr.name="Barley"
RETURN cr, o
```

"cr"	"o"
{"name":"Barley","id":2,"type":"cereal"}	{"date":"2022-10-26","quantity":187,"price":47.22}

- So, update the data accordingly.

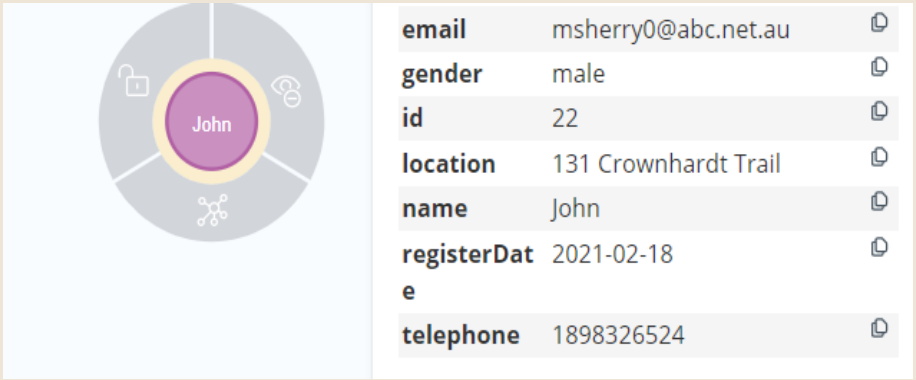
```
MATCH (f:Farmer)-[o:OWN]->(cr:Crop)
WHERE f.id=2 and cr.name="Barley"
SET o.quantity = 167
RETURN o
```

```
╒═══════════════════════════════════════════════════════════╕
│"o"                                                        │
╞═══════════════════════════════════════════════════════════╡
│{"date":"2022-10-26","quantity":167,"price":47.22}         │
└───────────────────────────────────────────────────────────┘
```

FEW BASIC OPERATIONS

- Create a farmer node

```
CREATE
(f22:Farmer{id: 22, name: "John", gender: "male", location: "131 Crownhardt Trail", registerDate: "2021-02-18", telephone: "1898326524", email: "msherry0@abc.net.au" })
```

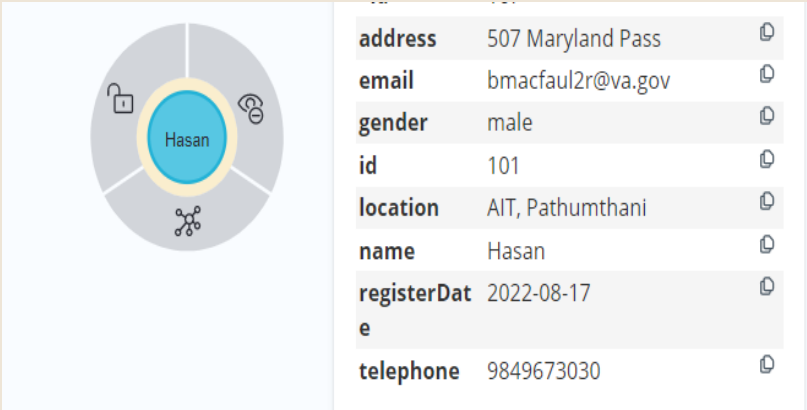


➢ Create a customer node

```
CREATE
(c101:Customer{id: 101, name: "Hasan", gender: "male", address: "507 Maryland Pass", email: "bmacfaul2r@va.gov", telepho
ne: "9849673030", registerDate: "2022-08-17" })
RETURN c101
```

➢ Update Customer address

```
match (c:Customer)
WHERE c.name='Hasan'
SET c.location='AIT, Pathumthani'
return c
```



CUSTOMER ANALYSIS

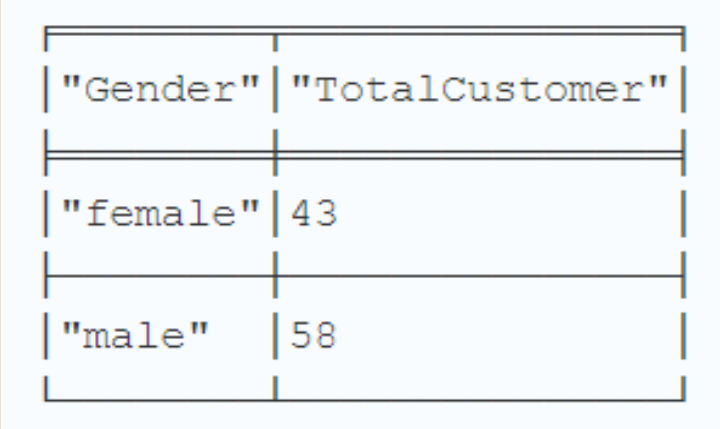
- Number of customers from different locations.

```
match (c:Customer)
return c.address as location, count(c) as TotalCustomer
```

"location"	"TotalCustomer"
"45 Maywood Circle"	1
"7447 Marquette Road"	1
"51 Orin Parkway"	1
"4 Columbus Crossing"	1
"433 Monterey Center"	1
"83 David Avenue"	1
"796 Toban Drive"	1

- Sort customers by gender

```
match (c:Customer)
return c.gender as Gender, count(c) as TotalCustomer
```



"Gender"	"TotalCustomer"
"female"	43
"male"	58

- Number of purchase group by gender

```
match (c:Customer)-[p:PURCHASE] ->(cr:Crop)
return c.gender as Gender, count(c) as TotalPrchase
```

Gender	TotalPrchase
"female"	117
"male"	183

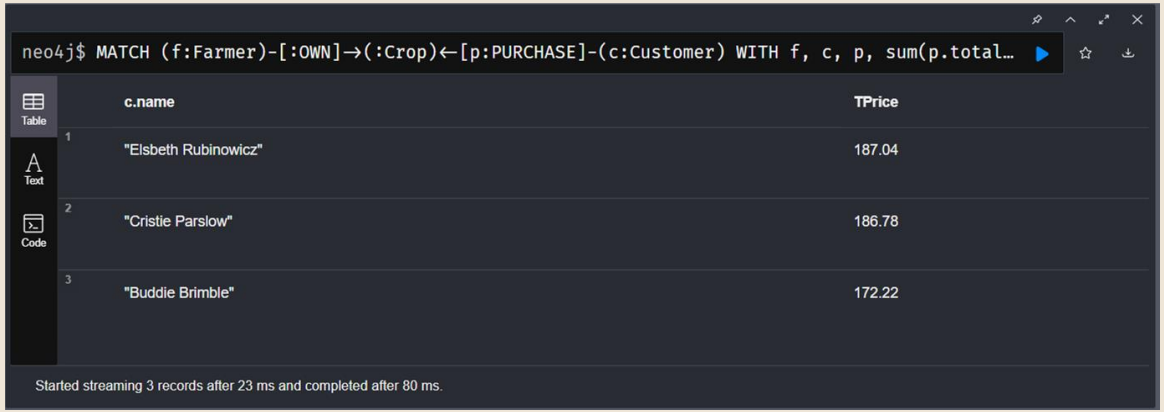
- Customer who ordered most

```
match (c:Customer)-[p:PURCHASE] ->(cr:Crop)
return c.name as customer, count(c) as TotalOrder
order by TotalOrder DESC
```

	customer	TotalOrder
1	"Tan Brentnall"	7
2	"Kennie Rasell"	7
3	"Stacie Silverthorne"	6
4	"Cal Scutts"	6
5	"Addia Vandenhoff"	6
6	"Amandy Haacker"	6

- Find top 3 customers who have a total price of purchasing with Albin the farmer.

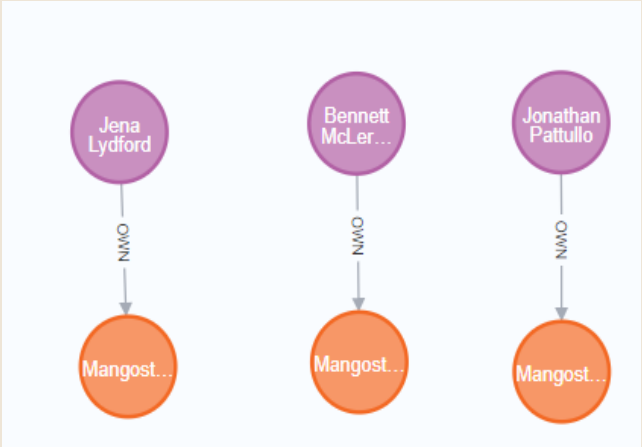
```
MATCH (f:Farmer)-[:OWN]->(:Crop)<-[p:PURCHASE]-(c:Customer)
WITH f, c, p, sum(p.totalPrice) as TPrice
WHERE f.name = 'Albin Raj Maskey'
RETURN DISTINCT c.name, TPrice
ORDER BY TPrice DESC LIMIT 3
```



FARMER ANALYSIS

- Looking for farmer with specific crop (Mangosteen)

```
match (f:Farmer)-[o:OWN]->(c:Crop)
where c.name="Mangosteen"
return f, c;
```



- Farmer with rating > 4

```
MATCH (f:Farmer)-[:OWN]->(cr:Crop)<-[r:RATE]-(c:Customer)
WHERE  r.rating > 4
RETURN Distinct cr.name as Crop, f.name as Farmer, r.rating as Rating
```

Crop	Farmer	Rating
"Soyabean"	"Albin Raj Maskey"	5
"Papaya"	"Albin Raj Maskey"	5
"Peanut"	"Egon Deppen"	5
"Mango"	"Darci Oxbe"	5
"Rambutan"	"Kliment Lansly"	5
"Banana"	"Bennett McLernon"	5

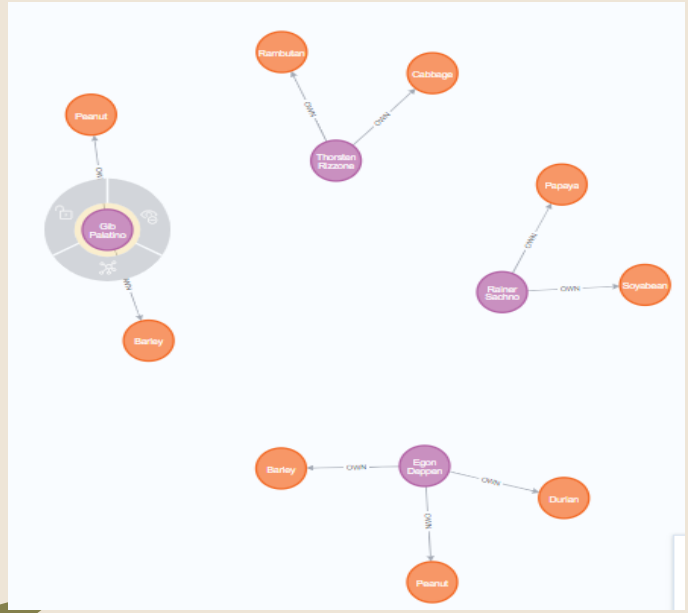
- Average rating of a farmer

```
MATCH (f:Farmer)-[:OWN]->(cr:Crop)<-[r:RATE]-(c:Customer)
WHERE  f.name="Rainer Sachno"
RETURN round(avg(r.rating), 2) as AvgRating
```

AvgRating
3.92

Sales data (farmer and crop information) for the month October

```
match (c:Customer)-[p:PURCHASE]->(cr:Crop)<-[:OWN]-(f:Farmer)
where p.datePurchase > "2022-09-31"
return Distinct f, cr
```



- Top 5 farmers with most sales for the month October

```
match (c:Customer)-[p:PURCHASE]->(cr:Crop)<-[:OWN]-(f:Farmer)
where p.datePurchase > "2022-09-31"
return Distinct f as farmer, count(f) as TotalSale
order by TotalSale DESC LIMIT 5
```

"farmer"	"TotalSale"
{"gender":"female","name":"Parker Newborn","telephone":"2704327282","location":"72 Muir Hill","id":11,"registerDate":"2021-10-14"}	23
{"gender":"male","name":"Bennett McLernon","telephone":"4342210625","location":"924 Monument Court","id":5,"email":"bmclernon4@e-recht24.de","registerDate":"2022-08-19"}	21
{"gender":"male","name":"Pip Fairlam","telephone":"7792145739","location":"4 Chive Plaza","id":18,"email":"pfairlamh@g.co","registerDate":"2022-10-06"}	19
{"gender":"male","name":"Zaneta Sheehan","location":"0 Briar Crest Parkway","telephone":"1746168575","id":10,"email":"zsheehan9@prnewswire.com","registerDate":"2022-09-24"}	17
{"gender":"male","name":"Albin Raj Maskey","telephone":"1898326524","location":"131 Crownhardt Trail","id":1,"email":"msherry0@abc.net.au","registerDate":"2021-02-18"}	17

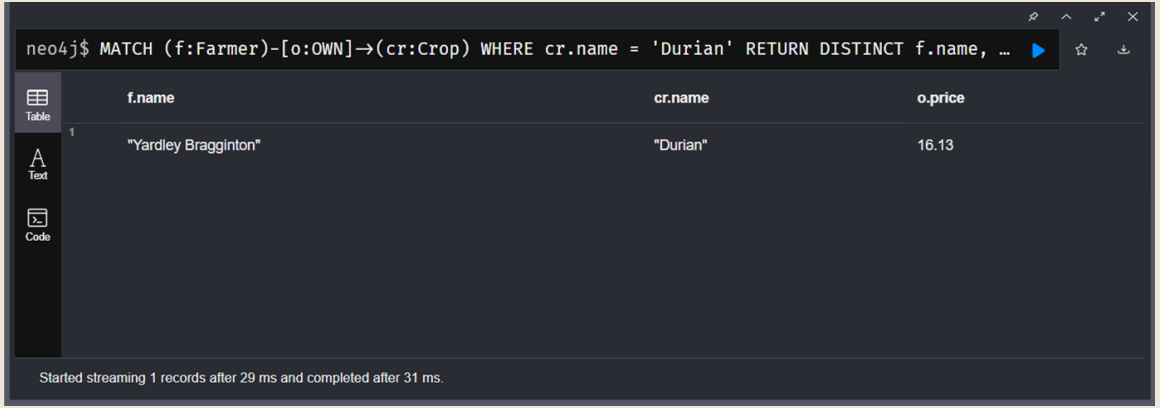
- Farmer Rating (> 3) in terms of gender

```
MATCH (f:Farmer)-[:OWN]->(cr:Crop)<-[r:RATE]-(c:Customer)
WHERE r.rating > 3
RETURN f.gender as Gender, count(f) as TotalFarmer
```

Gender	TotalFarmer
"male"	68
"female"	35

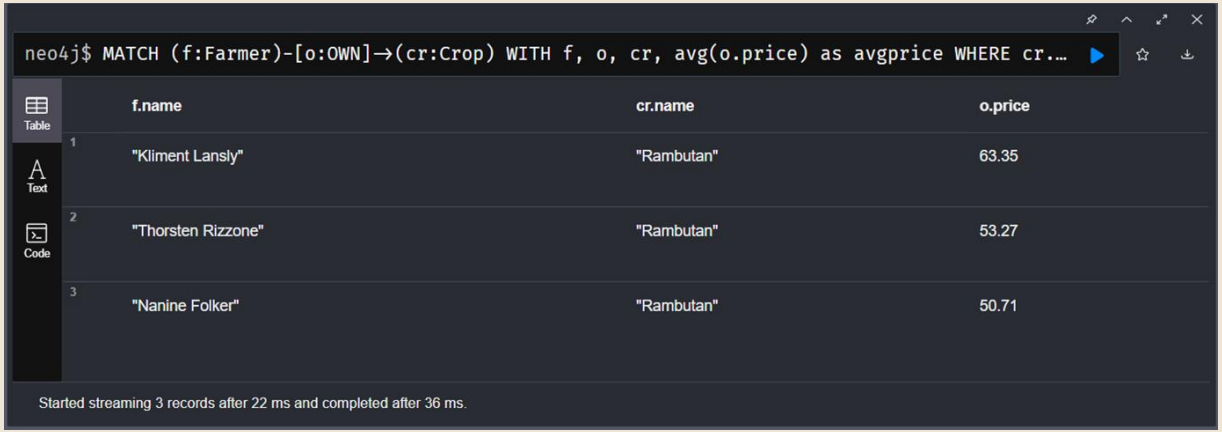
- Find the farmer who sell the cheapest durian

```
MATCH (f:Farmer)-[o:OWN]->(cr:Crop)
WHERE cr.name = 'Durian'
RETURN DISTINCT f.name, cr.name, o.price
ORDER BY o.price ASC LIMIT 1
```



- Find farmers and the price of rambutan that is lower than the average price.

```
MATCH (f:Farmer)-[o:OWN]->(cr:Crop)
WITH f, o, cr, avg(o.price) as avgprice
WHERE cr.name = 'Rambutan' and o.price <= avgprice
RETURN DISTINCT f.name, cr.name, o.price
```



CROP ANALYSIS

- Available crop type and crops

```
match (c:Crop)
return Distinct c.type as CropType
```

	CropType
1	"cereal"
2	"fruit"
3	"bean"
4	"rice"
5	"vegetable"

```
match (c:Crop)
return Distinct c.name as Crop
```

Crop
"Wheat"
"Barley"
"Oat"
"Corn"
"Mangosteen"
"Papaya"
"Durian"
"Mango"
"Rambutan"
"Banana"

"Soyabean"
"Peanut"
"Rice"
"Cabbage"
"Carrot"

- Average prices of different types of crops.

```
match (f:Farmer)-[o:OWN]->(cr:Crop)
return cr.type as CropType, round(avg(o.price), 2) as AvgPrice
```

CropType	AvgPrice
"bean"	43.68
"fruit"	37.26
"cereal"	32.05
"rice"	13.11
"vegetable"	45.7

- Average prices of different fruits

```
match (f:Farmer)-[o:OWN]->(cr:Crop)
Where cr.type="fruit"
return cr.name as FruitName, round(avg(o.price), 2) as AvgPrice
```

FruitName	AvgPrice
"Mangosteen"	23.24
"Papaya"	39.57
"Durian"	32.71
"Mango"	35.07
"Rambutan"	55.78
"Banana"	37.22

- Top 5 crop with best sales in 2022

```
match(c:Customer)-[p:PURCHASE]->(cr:Crop)
where p.datePurchase > "2021-12-31"
return cr.name as Crop, sum(p.quantity) as Total
order by Total DESC LIMIT 5
```

	Crop	Total
1	"Banana"	768
2	"Wheat"	701
3	"Soyabean"	693
4	"Rice"	580
5	"Barley"	541

- **Most sold crop types in 2022**

```
match(c:Customer)-[p:PURCHASE]->(cr:Crop)
where p.datePurchase > "2021-12-31"
return cr.type as CropType, sum(p.quantity) as Total
order by Total DESC
```

	CropType	Total
1	"fruit"	3025
2	"cereal"	1848
3	"bean"	1025
4	"vegetable"	772
5	"rice"	580

- Crop sell in winter

```
match(c:Customer)-[p:PURCHASE]->(cr:Crop)
where p.datePurchase > "2022-10-31" and p.datePurchase < "2022-12-31"
return cr.name as Crop, sum(p.quantity) as TotalWinterSale
order by TotalWinterSale DESC
```

	Crop	TotalWinterSale
1	"Banana"	184
2	"Rice"	160
3	"Wheat"	155
4	"Durian"	146
5	"Mangosteen"	141
6	"Cabbage"	137
7	"Carrot"	125
8	"Oat"	114
9	"Soyabean"	94
10	"Peanut"	91

SALES DATA ANALYSIS

- Crops with least sales

```
MATCH
(c:Customer)-[p:PURCHASE]->(cr:Crop)<-[:OWN]-(f:Farmer)
RETURN cr.name as cropName, count(p.quantity) as quantity
order by quantity
```

	cropName	quantity
1	"Oat"	12
2	"Peanut"	15
3	"Cabbage"	15
4	"Carrot"	15
5	"Papaya"	16
6	"Corn"	17
7		

ted streaming 15 records after 4 ms and completed after 33 ms.

- Which farmer sold fruit most

```
MATCH (c:Customer)-[p:PURCHASE]->(cr:Crop)<-[:OWN]-
(f:Farmer)
RETURN f.name as farmerName, cr.name as cropName,
sum(p.totalPrice) as totalAmount, count(p.quantity) as quantity
order by totalAmount  desc
```

"Jonathan Pattullo"	"Banana"	1388.4699999999998	14
"Albin Raj Maskey"	"Soyabean"	1265.92	10
"Normy Clayden"	"Rice"	1254.35	9
"Parker Newborn"	"Wheat"	1246.77	13
"Haleigh Issacof"	"Corn"	1090.06	10
"Parker Newborn"	"Soyabean"	1071.49	10

ed streaming 45 records after 3 ms and completed after 30 ms.

- Pending payment till date

```
MATCH (c:Customer)-[p:PURCHASE]->(cr:Crop)<-[:OWN]-(f:Farmer)
WHERE  p.datePurchase>"2022-11-10" and p.paymentStatus="Pending"
RETURN c.name as customerName,c.email as contact, p.datePurchase
as  dateofPurchase, p.paymentStatus as status
```

	customerName	contact	dateofPurchase	status
1	"Wynn Runge"	"wrunge1y@liveinternet.ru"	"2022-11-03"	"Pending"
2	"Kiah Lindback"	"klindbackq@marriott.com"	"2022-10-29"	"Pending"
3	"Gerard Alvis"	"galvis1n@example.com"	"2022-11-05"	"Pending"
4	"Erhard House"	"ehouse26@freewebs.com"	"2022-10-14"	"Pending"
5	"Dewain Spellissy"	"dspellissy1v@unicef.org"	"2022-11-08"	"Pending"
6	"Cal Scutts"	*null*	"2022-11-07"	"Pending"
7				

rted streaming 119 records after 4 ms and completed after 30 ms.

- Check the customer who still pending their payment and show how many order he/she still pending with total price

```
match (c:Customer)-[p:PURCHASE]->(cr:Crop)<-[:OWN]-
(f:Farmer) where p.paymentStatus="Pending"
return c.name as customerName,c.email as contact,
count(p.paymentStatus) as pendingorders,sum(p.totalPrice) as
totalAmount, p.paymentStatus as status
```

	customerName	contact	pendingorders	totalAmount	status
1	"Wenona Ilyenko"	"wilyenko24@printfriendly.com"	3	334.42	"Pending"
2	"Wynn Runge"	"wrunge1y@liveinternet.ru"	3	201.1	"Pending"
3	"Kiah Lindback"	"klindbackq@marriott.com"	2	260.42	"Pending"
4	"Gerard Alvis"	"galvis1n@example.com"	2	122.47	"Pending"
5	"Doyle Topham"	*null*	1	60.39	"Pending"
6	"Erhard House"	"ehouse26@freewebs.com"	2	313.63	"Pending"
7					

arted streaming 79 records after 3 ms and completed after 53 ms.

- Check each farmers' net value of crops remains (in case of they have to do something with insurance etc.).

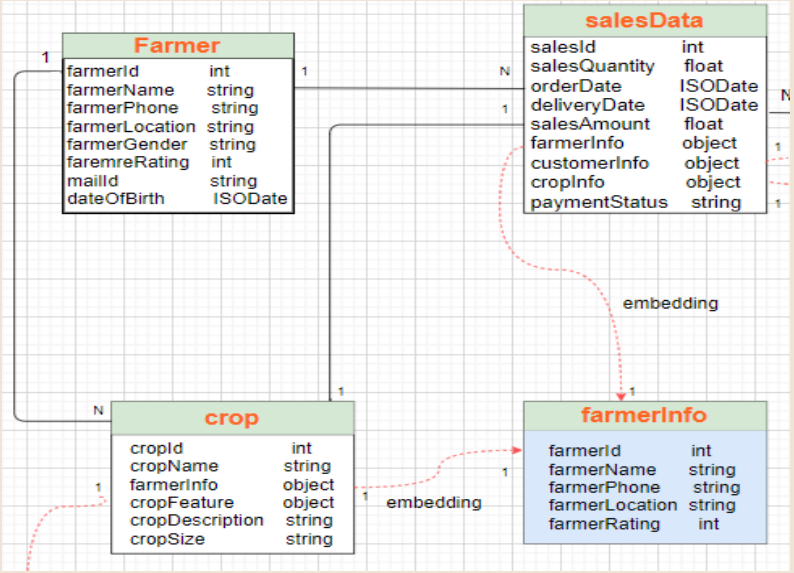
```
MATCH (f:Farmer)-[o:OWN]->(cr:Crop)
RETURN f.name as farmerName, sum(o.price) as netValue order by netValue
desc
```

	farmerName	netValue
1	"Egon Deppen"	161.89
2	"Thorsten Rizzone"	119.66
3	"Kliment Lansly"	118.89
4	"Nanine Folker"	109.16
5	"Bennett McLernon"	105.94999999999999
6	"Rochette Korting"	102.83000000000001
7		

arted streaming 20 records after 3 ms and completed after 7 ms.

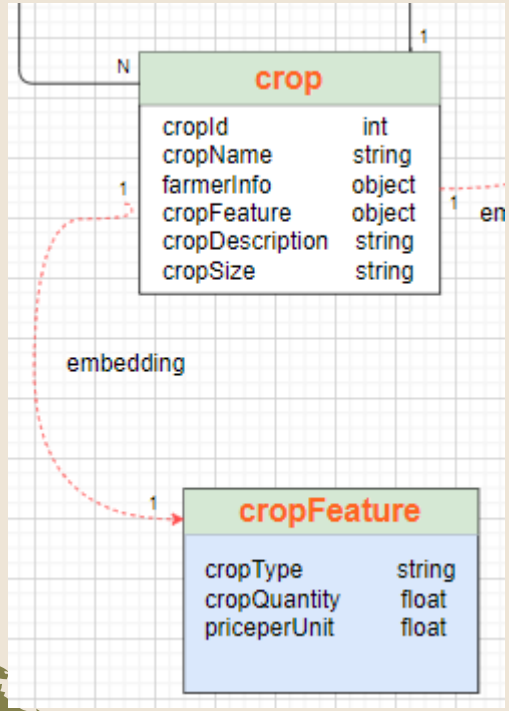
Comparison among RDBMS, Document and Graph model in terms of our project : benefits and limitations we faced

Benefits of using Document model



We see some benefits. Such as, ‘farmer’ collection has many attributes. We have embedded some key attributes from ‘farmer’ into ‘crop’. So, while we need farmer information related to crops, we directly extract that from the crop collection. In **RDBMS** we might have to join ‘farmer’ and ‘crop’ table. This makes our queries smooth.

Benefits of using Document model and Graph model in our project



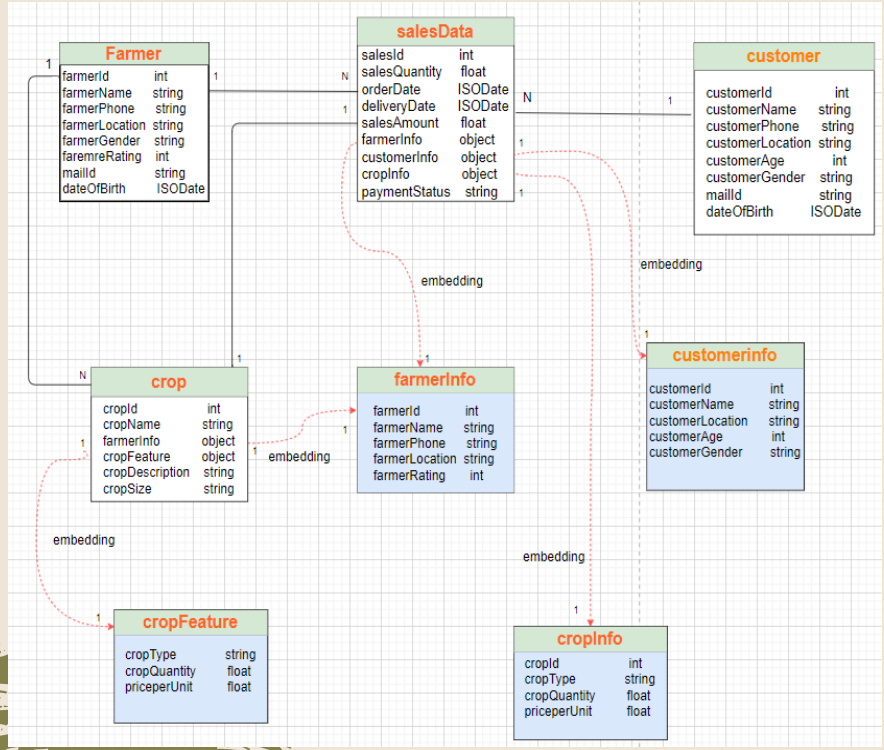
For some of our crops we have cropDescription and size. Such as, for fruits we have crop size, but for vegetable the crops don't have size. **This is the flexibility of NoSql.** We see similar benefits both in graph and document model. If it was **RDBMS** we might have a schema and some fixed attributes.

Benefits of using Graph model in our project

- We found that Graph DB Model visualize how the node related to each other which helped us for the quick implementation of queries.

- A user's favorites entity can be represented as edge in Graph DB Model as it shows the relationship between farmers, customers and crops, while in Relational and Document DB Model, it needs to be a table and a collection respectively.

Challenges of using Document model in our project



Such as, since we have embedded farmerInfo, cropInfo, and customerInfo into SalesData, it creates data duplication. Although it helps in terms of query implementation but it adds those additional data in the system. And, also we have to maintain data integrity. Such as, farmer information in famer collection and in the embedded farmerInfo are similar.

Challenges of using Graph model in our project

The queries that span the whole database might be tough to handle by graph model. Such as, to get the farmers with rating > 4 for his/her different crops we need to span through few relationships.

```
MATCH (f:Farmer)-[:OWN]->(cr:Crop)<-[r:RATE]-(c:Customer)
WHERE r.rating > 4
RETURN Distinct cr.name as Crop, f.name as Farmer, r.rating as Rating
```

If there were more complex queries it would be tiring to go through so many relationships.

Few final words from our observations

Relational databases provide a structured approach to data, whereas **graph and document databases are agile** and focus on quick data relationship insight. So, we think no one is meant to replace another. Although in terms of designing and implementing queries **we found document model more flexible for us.**