DBMS - MINI PROJECT

BOTANICAL GARDEN DATABASE

Submitted By:

SANIKA M RANGAYYAN

PES1UG20CS901

V Semester Section C

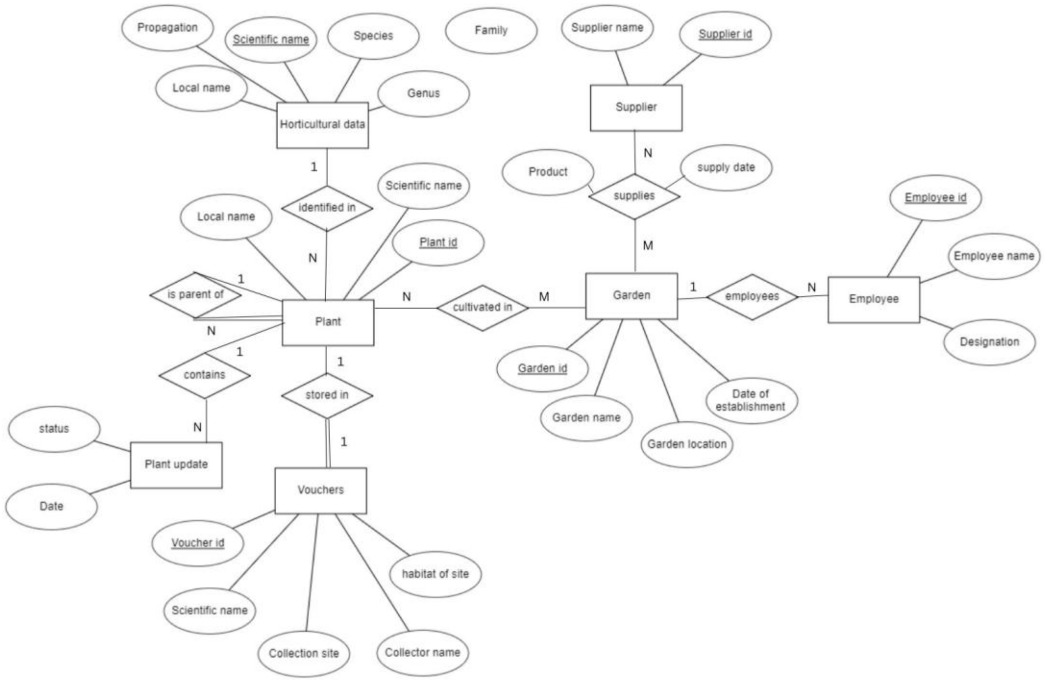
ABSTRACT

The project aims at building a database management system for botanical garden which has its division across various places. Considering the huge number of plants present in the garden either as live plants or as herbarium vouchers, managing the data efficiently helps in managing the garden adeptly. The end uses of the project are the garden managers across various gardens. The user can keep track of the plants that have been imported from other gardens or from other external sources, propagation of plants, plants that are preserved as herbarium vouchers and also periodic updates of each plant. Additionally, it also keeps a record of the suppliers who supply essential products and the employees working in the garden. The main purpose of the project is to keep track of the plant details efficiently so that the data can analysed in future to leverage the profit and plant culture of the garden. The project can be further developed to satisfy various real-time constrains and creating different portals to various echelons in the organization according to their requirement. The reference is taken from the botanical garden management application called BRAHMS.

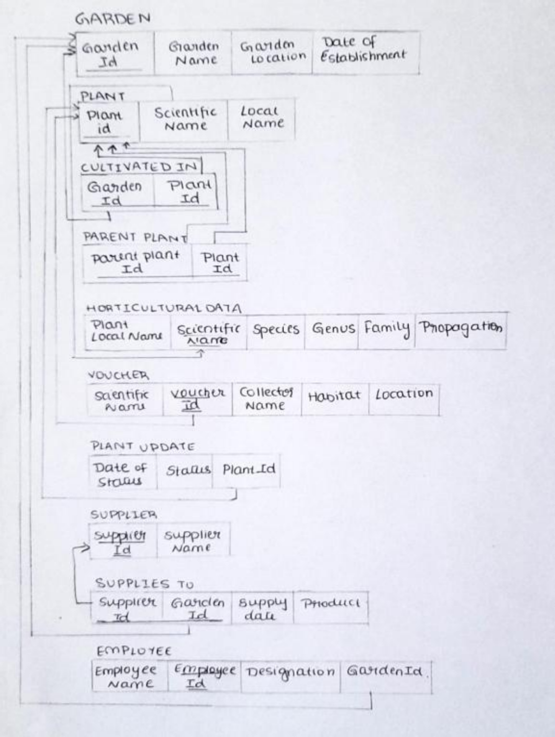
Constraints and Assumptions:

- Vouchers are made only for selective plants which are endangered. These plants are included in both plants and vouchers table. Whereas the live plants are recorded only in plants table.
- Plants are not removed from the plants table when they die. They are removed only from the cultivated_in table.
- It is assumed that a plant is propagated only from a single parent plant.

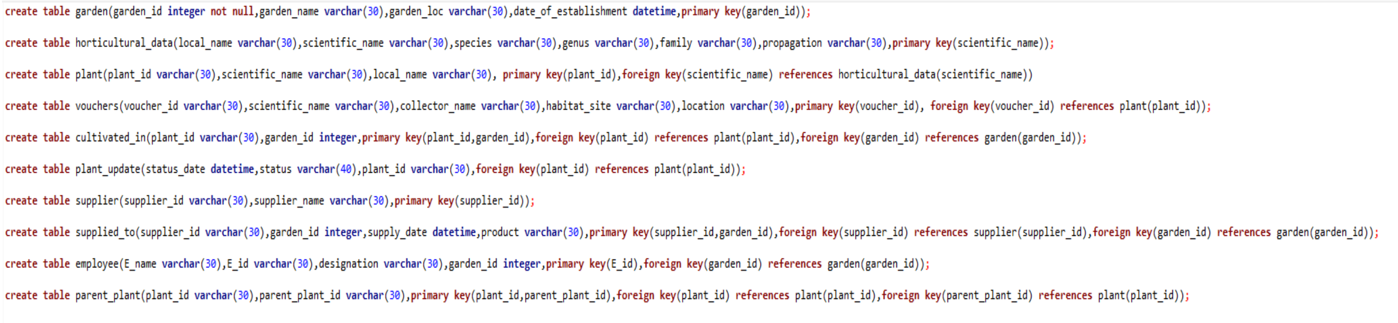
ER DIAGRAM



RELATIONAL SCHEMA

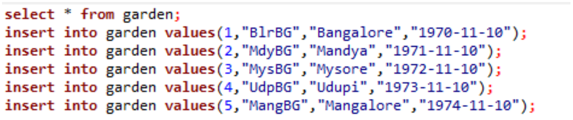


DDL STATEMENTS - BUILDING THE DATABASE

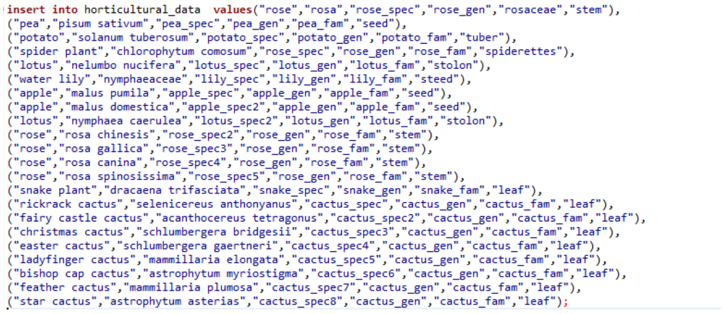


```
create table garden(garden_id integer not null,garden_name varchar(30),garden_loc varchar(30),date_of_establishment datetime,primary key(garden_id));

create table horticultural_data(local_name varchar(30),scientific_name varchar(30),species varchar(30),genus varchar(30),family varchar(30),propagation varchar(30),primary key(scientific_name));

create table plant(plant_id varchar(30),scientific_name varchar(30),local_name varchar(30), primary key(plant_id),foreign key(scientific_name) references horticultural_data(scientific_name))

create table vouchers(voucher_id varchar(30),scientific_name varchar(30),collector_name varchar(30),habitat_site varchar(30),location varchar(30),primary key(voucher_id), foreign key(voucher_id) references plant(plant_id));

create table cultivated_in(plant_id varchar(30),garden_id integer,primary key(plant_id,garden_id),foreign key(plant_id) references plant(plant_id),foreign key(garden_id) references garden(garden_id));

create table plant_update(status_date datetime,status varchar(40),plant_id varchar(30),foreign key(plant_id) references plant(plant_id));

create table supplier(supplier_id varchar(30),supplier_name varchar(30),primary key(supplier_id));

create table supplied_to(supplier_id varchar(30),garden_id integer,supply_date datetime,product varchar(30),primary key(supplier_id,garden_id),foreign key(supplier_id) references supplier(supplier_id),foreign key(garden_id) references garden(garden_id));

create table employee(E_name varchar(30),E_id varchar(30),designation varchar(30),garden_id integer,primary key(E_id),foreign key(garden_id) references garden(garden_id));

create table parent_plant(plant_id varchar(30),parent_plant_id varchar(30),primary key(plant_id,parent_plant_id),foreign key(plant_id) references plant(plant_id),foreign key(parent_plant_id) references plant(plant_id));
```

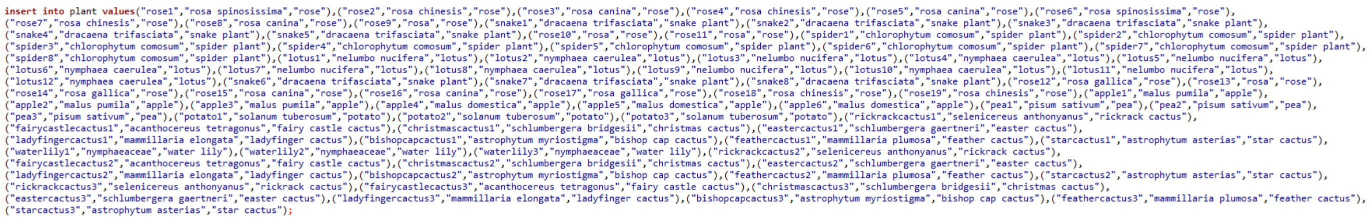
POPULATING THE DATABASE



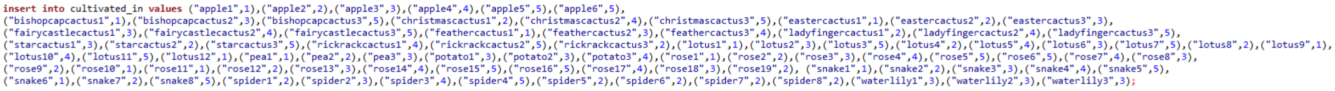
```
select * from garden;
insert into garden values(1,"BlrBG","Bangalore","1970-11-10");
insert into garden values(2,"MdyBG","Mandya","1971-11-10");
insert into garden values(3,"MysBG","Mysore","1972-11-10");
insert into garden values(4,"UdpBG","Udupi","1973-11-10");
insert into garden values(5,"MangBG","Mangalore","1974-11-10");
```



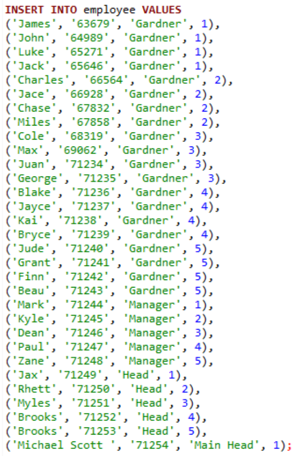
```
insert into horticultural_data  values("rose","rosa","rose_spec","rose_gen","rosaceae","stem"),
("pea","pisum sativum","pea_spec","pea_gen","pea_fam","seed"),
("potato","solanum tuberosum","potato_spec","potato_gen","potato_fam","tuber"),
("spider plant","chlorophytum comosum","rose_spec","rose_gen","rose_fam","spiderettes"),
("lotus","nelumbo nucifera","lotus_spec","lotus_gen","lotus_fam","stolon"),
("water lily","nymphaeaceae","lily_spec","lily_gen","lily_fam","steed"),
("apple","malus pumila","apple_spec","apple_gen","apple_fam","seed"),
("apple","malus domestica","apple_spec2","apple_gen","apple_fam","seed"),
("lotus","nymphaea caerulea","lotus_spec2","lotus_gen","lotus_fam","stolon"),
("rose","rosa chinesis","rose_spec2","rose_gen","rose_fam","stem"),
("rose","rosa gallica","rose_spec3","rose_gen","rose_fam","stem"),
("rose","rosa canina","rose_spec4","rose_gen","rose_fam","stem"),
("rose","rosa spinosissima","rose_spec5","rose_gen","rose_fam","stem"),
("snake plant","dracaena trifasciata","snake_spec","snake_gen","snake_fam","leaf"),
("rickrack cactus","selenicereus anthonyanus","cactus_spec","cactus_gen","cactus_fam","leaf"),
("fairy castle cactus","acanthocereus tetragonus","cactus_spec2","cactus_gen","cactus_fam","leaf"),
("christmas cactus","schlumbergera bridgesii","cactus_spec3","cactus_gen","cactus_fam","leaf"),
("easter cactus","schlumbergera gaertneri","cactus_spec4","cactus_gen","cactus_fam","leaf"),
("ladyfinger cactus","mammillaria elongata","cactus_spec5","cactus_gen","cactus_fam","leaf"),
("bishop cap cactus","astrophytum myriostigma","cactus_spec6","cactus_gen","cactus_fam","leaf"),
("feather cactus","mammillaria plumosa","cactus_spec7","cactus_gen","cactus_fam","leaf"),
("star cactus","astrophytum asterias","cactus_spec8","cactus_gen","cactus_fam","leaf");
```



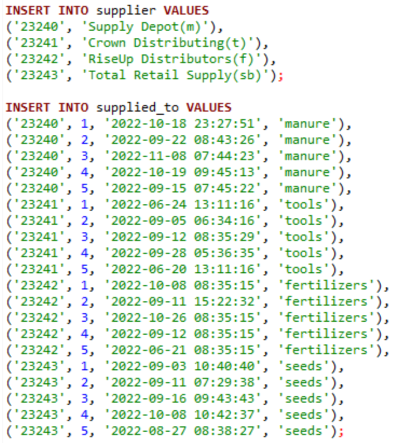
```
insert into plant values("rose1","rosa spinosissima","rose"),("rose2","rosa chinesis","rose"),("rose3","rosa canina","rose"),("rose4","rosa chinesis","rose"),("rose5","rosa canina","rose"),("rose6","rosa spinosissima","rose"),
("rose7","rosa chinesis","rose"),("rose8","rosa canina","rose"),("rose9","rosa","rose"),("snake1","dracaena trifasciata","snake plant"),("snake2","dracaena trifasciata","snake plant"),("snake3","dracaena trifasciata","snake plant"),
("snake4","dracaena trifasciata","snake plant"),("snake5","dracaena trifasciata","snake plant"),("rose10","rosa","rose"),("rose11","rosa","rose"),("spider1","chlorophytum comosum","spider plant"),("spider2","chlorophytum comosum","spider plant"),
("spider3","chlorophytum comosum","spider plant"),("spider4","chlorophytum comosum","spider plant"),("spider5","chlorophytum comosum","spider plant"),("spider6","chlorophytum comosum","spider plant"),("spider7","chlorophytum comosum","spider plant"),
("spider8","chlorophytum comosum","spider plant"),("lotus1","nelumbo nucifera","lotus"),("lotus2","nymphaea caerulea","lotus"),("lotus3","nelumbo nucifera","lotus"),("lotus4","nymphaea caerulea","lotus"),("lotus5","nelumbo nucifera","lotus"),
("lotus6","nymphaea caerulea","lotus"),("lotus7","nelumbo nucifera","lotus"),("lotus8","nymphaea caerulea","lotus"),("lotus9","nelumbo nucifera","lotus"),("lotus10","nymphaea caerulea","lotus"),("lotus11","nelumbo nucifera","lotus"),
("lotus12","nymphaea caerulea","lotus"),("snake6","dracaena trifasciata","snake plant"),("snake7","dracaena trifasciata","snake plant"),("snake8","dracaena trifasciata","snake plant"),("rose12","rosa gallica","rose"),("rose13","rosa","rose"),
("rose14","rosa gallica","rose"),("rose15","rosa canina","rose"),("rose16","rosa canina","rose"),("rose17","rosa gallica","rose"),("rose18","rosa chinesis","rose"),("rose19","rosa chinesis","rose"),("apple1","malus pumila","apple"),
("apple2","malus pumila","apple"),("apple3","malus pumila","apple"),("apple4","malus domestica","apple"),("apple5","malus domestica","apple"),("apple6","malus domestica","apple"),("pea1","pisum sativum","pea"),("pea2","pisum sativum","pea"),
("pea3","pisum sativum","pea"),("potato1","solanum tuberosum","potato"),("potato2","solanum tuberosum","potato"),("potato3","solanum tuberosum","potato"),("rickrackcactus1","selenicereus anthonyanus","rickrack cactus"),
("fairycastlecactus1","acanthocereus tetragonus","fairy castle cactus"),("christmascactus1","schlumbergera bridgesii","christmas cactus"),("eastercactus1","schlumbergera gaertneri","easter cactus"),
("ladyfingercactus1","mammillaria elongata","ladyfinger cactus"),("bishopcapcactus1","astrophytum myriostigma","bishop cap cactus"),("feathercactus1","mammillaria plumosa","feather cactus"),("starcactus1","astrophytum asterias","star cactus"),
("waterlily1","nymphaeaceae","water lily"),("waterlily2","nymphaeaceae","water lily"),("waterlily3","nymphaeaceae","water lily"),("rickrackcactus2","selenicereus anthonyanus","rickrack cactus"),
("fairycastlecactus2","acanthocereus tetragonus","fairy castle cactus"),("christmascactus2","schlumbergera bridgesii","christmas cactus"),("eastercactus2","schlumbergera gaertneri","easter cactus"),
("ladyfingercactus2","mammillaria elongata","ladyfinger cactus"),("bishopcapcactus2","astrophytum myriostigma","bishop cap cactus"),("feathercactus2","mammillaria plumosa","feather cactus"),("starcactus2","astrophytum asterias","star cactus"),
("rickrackcactus3","selenicereus anthonyanus","rickrack cactus"),("fairycastlecactus3","acanthocereus tetragonus","fairy castle cactus"),("christmascactus3","schlumbergera bridgesii","christmas cactus"),
("eastercactus3","schlumbergera gaertneri","easter cactus"),("ladyfingercactus3","mammillaria elongata","ladyfinger cactus"),("bishopcapcactus3","astrophytum myriostigma","bishop cap cactus"),("feathercactus3","mammillaria plumosa","feather cactus"),
("starcactus3","astrophytum asterias","star cactus");
```



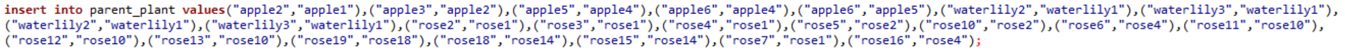
```
insert into cultivated_in values ("apple1",1),("apple2",2),("apple3",3),("apple4",4),("apple5",5),("apple6",5),
("bishopcapcactus1",1),("bishopcapcactus2",3),("bishopcapcactus3",5),("christmascactus1",2),("christmascactus2",4),("christmascactus3",5),("eastercactus1",1),("eastercactus2",2),("eastercactus3",3),
("fairycastlecactus1",3),("fairycastlecactus2",4),("fairycastlecactus3",5),("feathercactus1",1),("feathercactus2",3),("feathercactus3",4),("ladyfingercactus1",2),("ladyfingercactus2",4),("ladyfingercactus3",5),
("starcactus1",3),("starcactus2",2),("starcactus3",5),("rickrackcactus1",4),("rickrackcactus2",5),("rickrackcactus3",2),("lotus1",1),("lotus2",3),("lotus3",5),("lotus4",2),("lotus5",4),("lotus6",3),("lotus7",5),("lotus8",2),("lotus9",1),
("lotus10",4),("lotus11",5),("lotus12",1),("pea1",1),("pea2",2),("pea3",3),("potato1",3),("potato2",3),("potato3",4),("rose1",1),("rose2",2),("rose3",3),("rose4",4),("rose5",5),("rose6",5),("rose7",4),("rose8",3),
("rose9",2),("rose10",1),("rose11",1),("rose12",2),("rose13",3),("rose14",4),("rose15",5),("rose16",5),("rose17",4),("rose18",3),("rose19",2), ("snake1",1),("snake2",2),("snake3",3),("snake4",4),("snake5",5),
("snake6",1),("snake7",2),("snake8",5),("spider1",2),("spider2",3),("spider3",4),("spider4",5),("spider5",2),("spider6",2),("spider7",2),("spider8",2),("waterlily1",3),("waterlily2",3),("waterlily3",3);
```



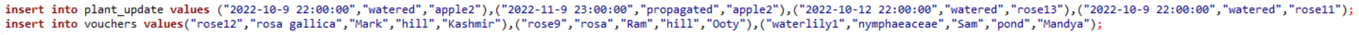
```
INSERT INTO employee VALUES
('James', '63679', 'Gardner', 1),
('John', '64989', 'Gardner', 1),
('Luke', '65271', 'Gardner', 1),
('Jack', '65646', 'Gardner', 1),
('Charles', '66564', 'Gardner', 2),
('Jace', '66928', 'Gardner', 2),
('Chase', '67832', 'Gardner', 2),
('Miles', '67858', 'Gardner', 2),
('Cole', '68319', 'Gardner', 3),
('Max', '69062', 'Gardner', 3),
('Juan', '71234', 'Gardner', 3),
('George', '71235', 'Gardner', 3),
('Blake', '71236', 'Gardner', 4),
('Jayce', '71237', 'Gardner', 4),
('Kai', '71238', 'Gardner', 4),
('Bryce', '71239', 'Gardner', 4),
('Jude', '71240', 'Gardner', 5),
('Grant', '71241', 'Gardner', 5),
('Finn', '71242', 'Gardner', 5),
('Beau', '71243', 'Gardner', 5),
('Mark', '71244', 'Manager', 1),
('Kyle', '71245', 'Manager', 2),
('Dean', '71246', 'Manager', 3),
('Paul', '71247', 'Manager', 4),
('Zane', '71248', 'Manager', 5),
('Jax', '71249', 'Head', 1),
('Rhett', '71250', 'Head', 2),
('Myles', '71251', 'Head', 3),
('Brooks', '71252', 'Head', 4),
('Brooks', '71253', 'Head', 5),
('Michael Scott ', '71254', 'Main Head', 1);
```



```
INSERT INTO supplier VALUES
('23240', 'Supply Depot(m)'),
('23241', 'Crown Distributing(t)'),
('23242', 'RiseUp Distributors(f)'),
('23243', 'Total Retail Supply(sb)');

INSERT INTO supplied_to VALUES
('23240', 1, '2022-10-18 23:27:51', 'manure'),
('23240', 2, '2022-09-22 08:43:26', 'manure'),
('23240', 3, '2022-11-08 07:44:23', 'manure'),
('23240', 4, '2022-10-19 09:45:13', 'manure'),
('23240', 5, '2022-09-15 07:45:22', 'manure'),
('23241', 1, '2022-06-24 13:11:16', 'tools'),
('23241', 2, '2022-09-05 06:34:16', 'tools'),
('23241', 3, '2022-09-12 08:35:29', 'tools'),
('23241', 4, '2022-09-28 05:36:35', 'tools'),
('23241', 5, '2022-06-20 13:11:16', 'tools'),
('23242', 1, '2022-10-08 08:35:15', 'fertilizers'),
('23242', 2, '2022-09-11 15:22:32', 'fertilizers'),
('23242', 3, '2022-10-26 08:35:15', 'fertilizers'),
('23242', 4, '2022-09-12 08:35:15', 'fertilizers'),
('23242', 5, '2022-06-21 08:35:15', 'fertilizers'),
('23243', 1, '2022-09-03 10:40:40', 'seeds'),
('23243', 2, '2022-09-11 07:29:38', 'seeds'),
('23243', 3, '2022-09-16 09:43:43', 'seeds'),
('23243', 4, '2022-10-08 10:42:37', 'seeds'),
('23243', 5, '2022-08-27 08:38:27', 'seeds');
```



```
insert into parent_plant values("apple2","apple1"),("apple3","apple2"),("apple5","apple4"),("apple6","apple4"),("apple6","apple5"),("waterlily2","waterlily1"),("waterlily3","waterlily1"),
("waterlily2","waterlily1"),("waterlily3","waterlily1"),("rose2","rose1"),("rose3","rose1"),("rose4","rose1"),("rose5","rose2"),("rose10","rose2"),("rose6","rose4"),("rose11","rose10"),
("rose12","rose10"),("rose13","rose10"),("rose19","rose18"),("rose18","rose14"),("rose15","rose14"),("rose7","rose1"),("rose16","rose4");
```



```
insert into plant_update values ("2022-10-9 22:00:00","watered","apple2"),("2022-11-9 23:00:00","propagated","apple2"),("2022-10-12 22:00:00","watered","rose13"),("2022-10-9 22:00:00","watered","rose11");
insert into vouchers values("rose12","rosa gallica","Mark","hill","Kashmir"),("rose9","rosa","Ram","hill","Ooty"),("waterlily1","nymphaeaceae","Sam","pond","Mandya");
```

TOOLD USED

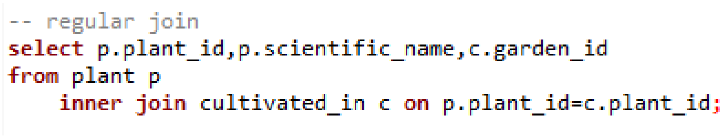
The queries are written in MySQL in the database tool called DBeaver. The front-end is built using streamlit and is connected to the backend using mysql connector.

QUERIES

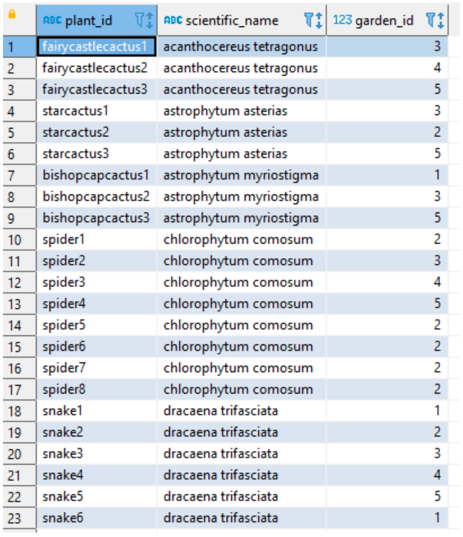
JOIN QUERIES

REGULAR JOINS:

1) Aim is to find the corresponding garden IDs of the given plant IDs

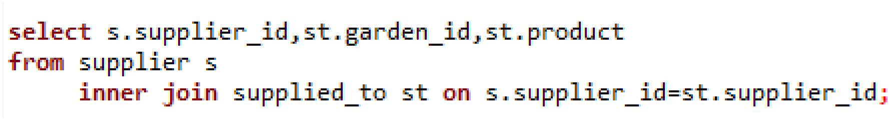


```
-- regular join
select p.plant_id,p.scientific_name,c.garden_id
from plant p
    inner join cultivated_in c on p.plant_id=c.plant_id;
```

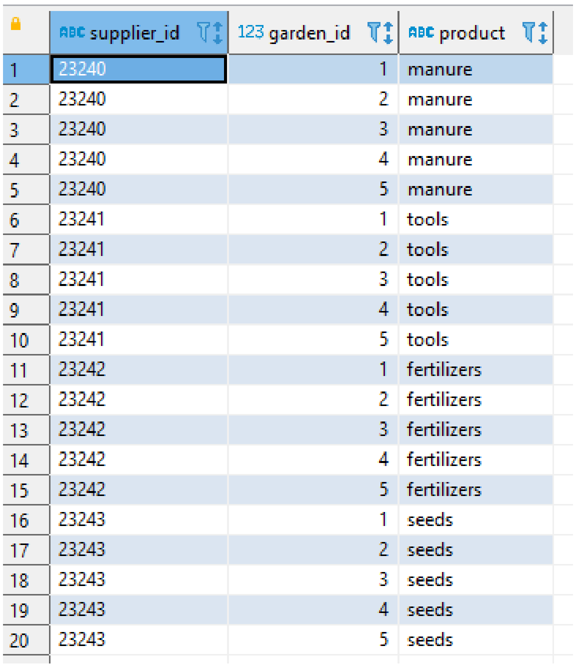


	plant_id	scientific_name	garden_id
1	fairycastlecactus1	acanthocereus tetragonus	3
2	fairycastlecactus2	acanthocereus tetragonus	4
3	fairycastlecactus3	acanthocereus tetragonus	5
4	starcactus1	astrophytum asterias	3
5	starcactus2	astrophytum asterias	2
6	starcactus3	astrophytum asterias	5
7	bishopcapcactus1	astrophytum myriostigma	1
8	bishopcapcactus2	astrophytum myriostigma	3
9	bishopcapcactus3	astrophytum myriostigma	5
10	spider1	chlorophytum comosum	2
11	spider2	chlorophytum comosum	3
12	spider3	chlorophytum comosum	4
13	spider4	chlorophytum comosum	5
14	spider5	chlorophytum comosum	2
15	spider6	chlorophytum comosum	2
16	spider7	chlorophytum comosum	2
17	spider8	chlorophytum comosum	2
18	snake1	dracaena trifasciata	1
19	snake2	dracaena trifasciata	2
20	snake3	dracaena trifasciata	3
21	snake4	dracaena trifasciata	4
22	snake5	dracaena trifasciata	5
23	snake6	dracaena trifasciata	1

2) Aim is to list the supplier IDs along with the garden IDs and respective product.



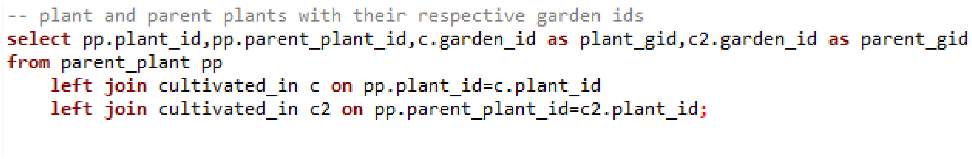
```
select s.supplier_id,st.garden_id,st.product
from supplier s
     inner join supplied_to st on s.supplier_id=st.supplier_id;
```



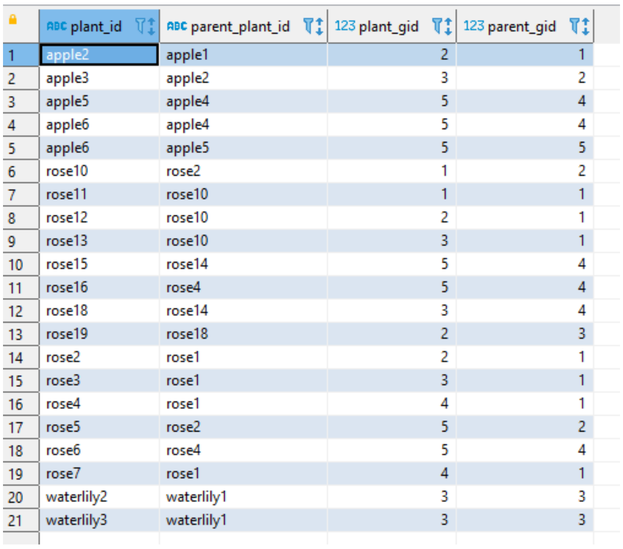
	supplier_id	garden_id	product
1	23240	1	manure
2	23240	2	manure
3	23240	3	manure
4	23240	4	manure
5	23240	5	manure
6	23241	1	tools
7	23241	2	tools
8	23241	3	tools
9	23241	4	tools
10	23241	5	tools
11	23242	1	fertilizers
12	23242	2	fertilizers
13	23242	3	fertilizers
14	23242	4	fertilizers
15	23242	5	fertilizers
16	23243	1	seeds
17	23243	2	seeds
18	23243	3	seeds
19	23243	4	seeds
20	23243	5	seeds

CO-RELATED JOINS:

1) The following code lists of the garden IDs of plant and its parent plant. This helps in analysing which plant has been propagated from plant and from which garden. This would help us to track the origins of the plant.

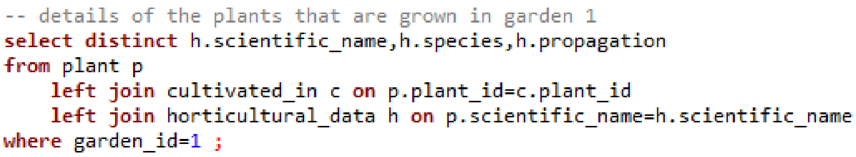


```
-- plant and parent plants with their respective garden ids
select pp.plant_id,pp.parent_plant_id,c.garden_id as plant_gid,c2.garden_id as parent_gid
from parent_plant pp
    left join cultivated_in c on pp.plant_id=c.plant_id
    left join cultivated_in c2 on pp.parent_plant_id=c2.plant_id;
```

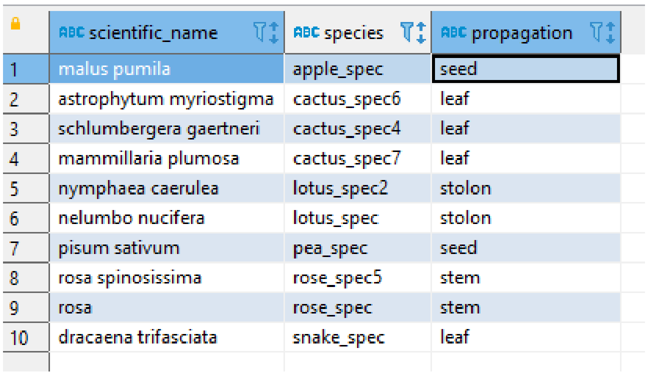


	plant_id	parent_plant_id	plant_gid	parent_gid
1	apple2	apple1	2	1
2	apple3	apple2	3	2
3	apple5	apple4	5	4
4	apple6	apple4	5	4
5	apple6	apple5	5	5
6	rose10	rose2	1	2
7	rose11	rose10	1	1
8	rose12	rose10	2	1
9	rose13	rose10	3	1
10	rose15	rose14	5	4
11	rose16	rose4	5	4
12	rose18	rose14	3	4
13	rose19	rose18	2	3
14	rose2	rose1	2	1
15	rose3	rose1	3	1
16	rose4	rose1	4	1
17	rose5	rose2	5	2
18	rose6	rose4	5	4
19	rose7	rose1	4	1
20	waterlily2	waterlily1	3	3
21	waterlily3	waterlily1	3	3

2) Listing the distinct species of plants that are grown in garden with ID 1



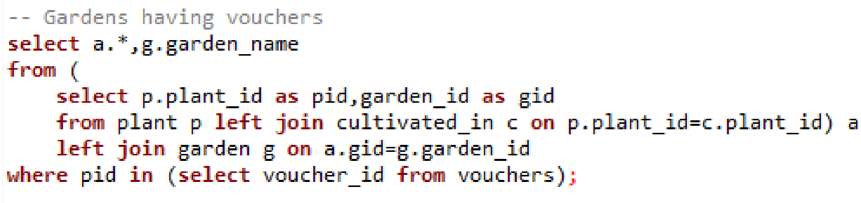
```
-- details of the plants that are grown in garden 1
select distinct h.scientific_name,h.species,h.propagation
from plant p
    left join cultivated_in c on p.plant_id=c.plant_id
    left join horticultural_data h on p.scientific_name=h.scientific_name
where garden_id=1 ;
```



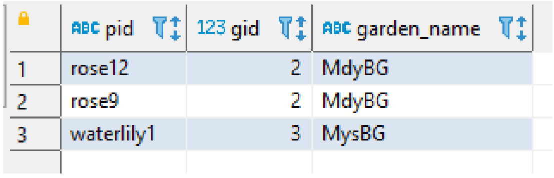
	scientific_name	species	propagation
1	malus pumila	apple_spec	seed
2	astrophytum myriostigma	cactus_spec6	leaf
3	schlumbergera gaertneri	cactus_spec4	leaf
4	mammillaria plumosa	cactus_spec7	leaf
5	nymphaea caerulea	lotus_spec2	stolon
6	nelumbo nucifera	lotus_spec	stolon
7	pisum sativum	pea_spec	seed
8	rosa spinosissima	rose_spec5	stem
9	rosa	rose_spec	stem
10	dracaena trifasciata	snake_spec	leaf

NESTED QUERIES:

1) The query lists out the plants from the plant table which are preserved as vouchers. Vouchers are maintained only for those species of plants that are endangered. The garden may contain live plant or plants in herbarium sheets or vouchers.

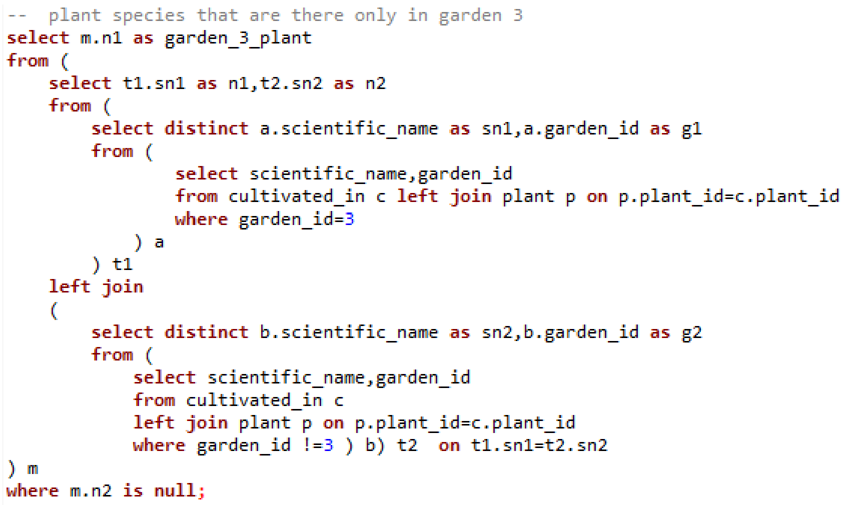


```
-- Gardens having vouchers
select a.*,g.garden_name
from (
    select p.plant_id as pid,garden_id as gid
    from plant p left join cultivated_in c on p.plant_id=c.plant_id) a
    left join garden g on a.gid=g.garden_id
where pid in (select voucher_id from vouchers);
```

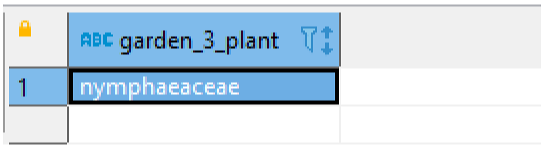


	pid	gid	garden_name
1	rose12	2	MdyBG
2	rose9	2	MdyBG
3	waterlily1	3	MysBG

2) The query lists the species of a plant that are present only in garden 3 and no other garden. This helps in analyzing the uniform distribution of plants among the gardens.



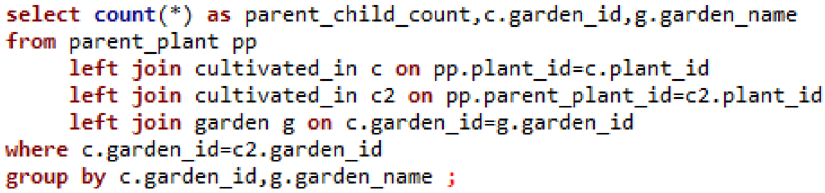
```
--  plant species that are there only in garden 3
select m.n1 as garden_3_plant
from (
    select t1.sn1 as n1,t2.sn2 as n2
    from (
        select distinct a.scientific_name as sn1,a.garden_id as g1
        from (
                select scientific_name,garden_id
                from cultivated_in c left join plant p on p.plant_id=c.plant_id
                where garden_id=3
            ) a
        ) t1
    left join
    (
        select distinct b.scientific_name as sn2,b.garden_id as g2
        from (
            select scientific_name,garden_id
            from cultivated_in c
            left join plant p on p.plant_id=c.plant_id
            where garden_id !=3 ) b) t2  on t1.sn1=t2.sn2
) m
where m.n2 is null;
```



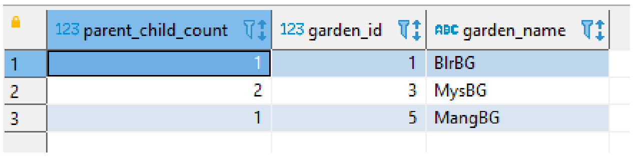
	garden_3_plant
1	nymphaeaceae

AGGREGATE FUNCTION:

1) The query finds the number of parent-child plant pairs that belong to the same garden along with their garden name.

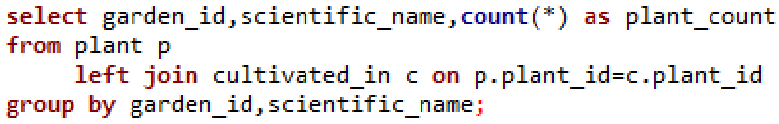


```
select count(*) as parent_child_count,c.garden_id,g.garden_name
from parent_plant pp
    left join cultivated_in c on pp.plant_id=c.plant_id
    left join cultivated_in c2 on pp.parent_plant_id=c2.plant_id
    left join garden g on c.garden_id=g.garden_id
where c.garden_id=c2.garden_id
group by c.garden_id,g.garden_name ;
```

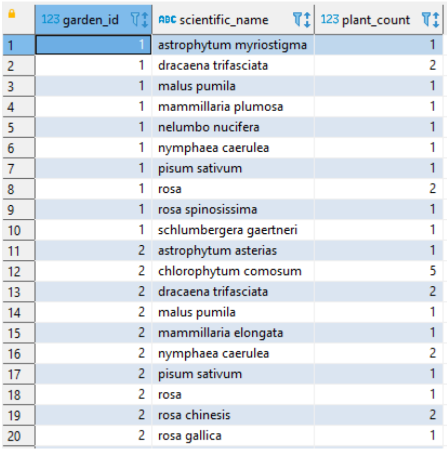


	parent_child_count	garden_id	garden_name
1	1	1	BlrBG
2	2	3	MysBG
3	1	5	MangBG

2) The query finds number of plants present in each species and in each garden.



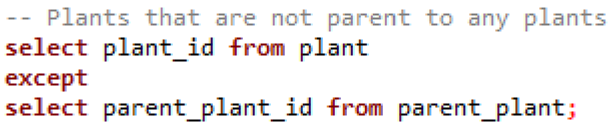
```
select garden_id,scientific_name,count(*) as plant_count
from plant p
    left join cultivated_in c on p.plant_id=c.plant_id
group by garden_id,scientific_name;
```



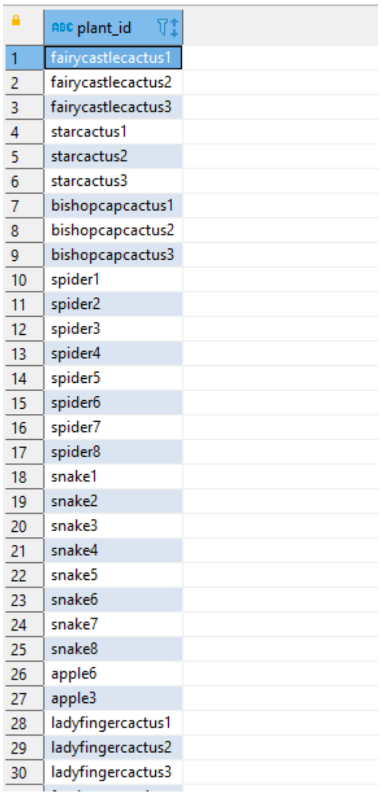
	garden_id	scientific_name	plant_count
1	1	astrophytum myriostigma	1
2	1	dracaena trifasciata	2
3	1	malus pumila	1
4	1	mammillaria plumosa	1
5	1	nelumbo nucifera	1
6	1	nymphaea caerulea	1
7	1	pisum sativum	1
8	1	rosa	2
9	1	rosa spinosissima	1
10	1	schlumbergera gaertneri	1
11	2	astrophytum asterias	1
12	2	chlorophytum comosum	5
13	2	dracaena trifasciata	2
14	2	malus pumila	1
15	2	mammillaria elongata	1
16	2	nymphaea caerulea	2
17	2	pisum sativum	1
18	2	rosa	1
19	2	rosa chinesis	2
20	2	rosa gallica	1

SET OPERATIONS:

1) The query lists out the plants that have not propagated.

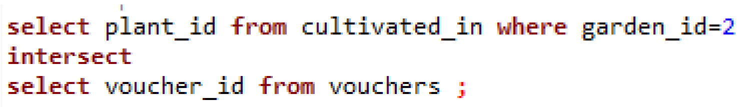


```
-- Plants that are not parent to any plants
select plant_id from plant
except
select parent_plant_id from parent_plant;
```

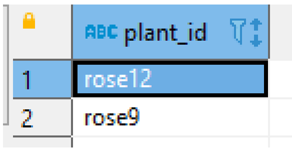


	plant_id
1	fairycastlecactus1
2	fairycastlecactus2
3	fairycastlecactus3
4	starcactus1
5	starcactus2
6	starcactus3
7	bishopcapcactus1
8	bishopcapcactus2
9	bishopcapcactus3
10	spider1
11	spider2
12	spider3
13	spider4
14	spider5
15	spider6
16	spider7
17	spider8
18	snake1
19	snake2
20	snake3
21	snake4
22	snake5
23	snake6
24	snake7
25	snake8
26	apple6
27	apple3
28	ladyfingercactus1
29	ladyfingercactus2
30	ladyfingercactus3

2) The query lists the plants that are cultivated in garden wit Id 2 and is a voucher. There are only two record because the proportion of plants maintained in the gardens as vouchers in less



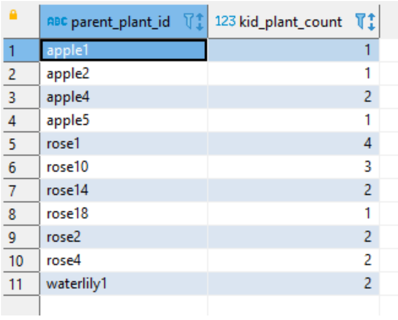
```
select plant_id from cultivated_in where garden_id=2
intersect
select voucher_id from vouchers ;
```



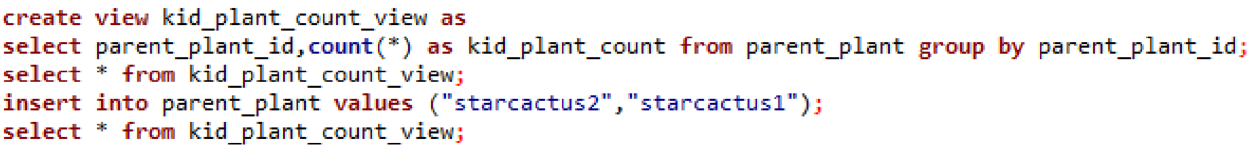
	plant_id
1	rose12
2	rose9

VIEWS:

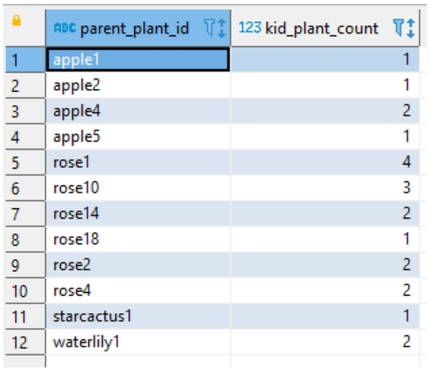
The view takes the plant and number of kids the plant has. In other words, the view contains the plant and count of the number of times the plant is propagated and maintained in the same garden chain. The propagated plant can be present in other garden of the same chain also. The view is updated with a new plant propagates.



	parent_plant_id	kid_plant_count
1	apple1	1
2	apple2	1
3	apple4	2
4	apple5	1
5	rose1	4
6	rose10	3
7	rose14	2
8	rose18	1
9	rose2	2
10	rose4	2
11	waterlily1	2



```
create view kid_plant_count_view as
select parent_plant_id,count(*) as kid_plant_count from parent_plant group by parent_plant_id;
select * from kid_plant_count_view;
insert into parent_plant values ("starcactus2","starcactus1");
select * from kid_plant_count_view;
```

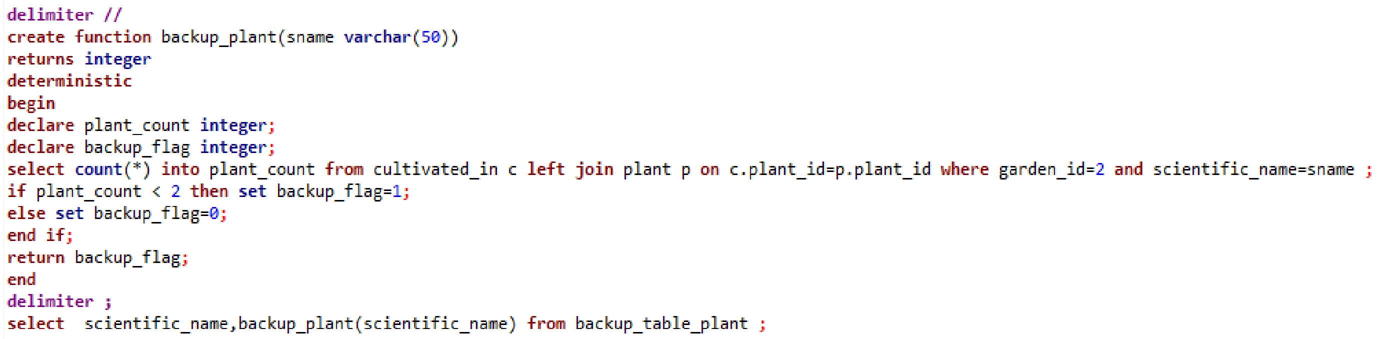


	parent_plant_id	kid_plant_count
1	apple1	1
2	apple2	1
3	apple4	2
4	apple5	1
5	rose1	4
6	rose10	3
7	rose14	2
8	rose18	1
9	rose2	2
10	rose4	2
11	starcactus1	1
12	waterlily1	2

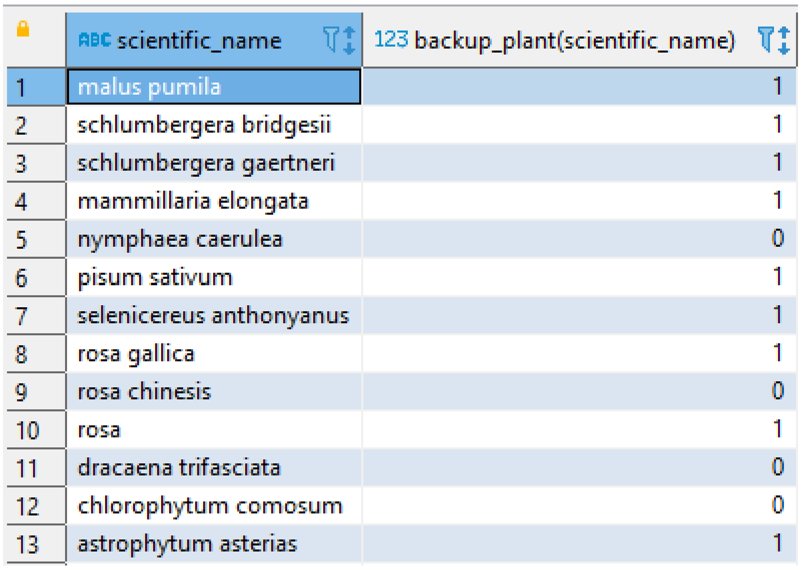
TRIGGERS (FUNCTIONS OR PROCEDURE)

FUNCTIONS:

For the given garden with ID 2 , the function notifies which plant species needs a backup. When there is only one plant of a given species, this calls for the time when another plant of the species must be imported or propagated. The backup flag 1 indicated that the plant count of the given species must be increased.



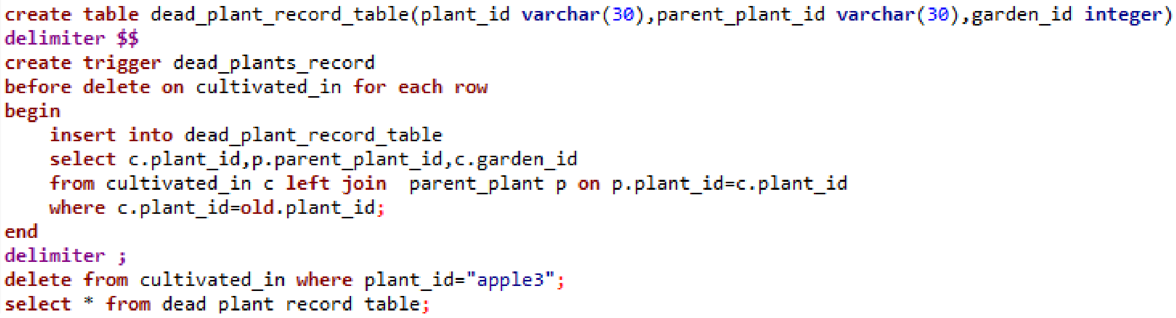
```
delimiter //
create function backup_plant(sname varchar(50))
returns integer
deterministic
begin
declare plant_count integer;
declare backup_flag integer;
select count(*) into plant_count from cultivated_in c left join plant p on c.plant_id=p.plant_id where garden_id=2 and scientific_name=sname ;
if plant_count < 2 then set backup_flag=1;
else set backup_flag=0;
end if;
return backup_flag;
end
delimiter ;
select  scientific_name,backup_plant(scientific_name) from backup_table_plant ;
```



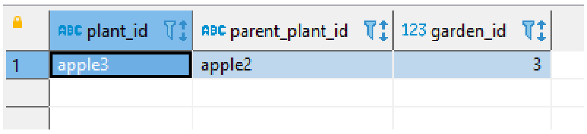
	scientific_name	backup_plant(scientific_name)
1	malus pumila	1
2	schlumbergera bridgesii	1
3	schlumbergera gaertneri	1
4	mammillaria elongata	1
5	nymphaea caerulea	0
6	pisum sativum	1
7	selenicereus anthonyanus	1
8	rosa gallica	1
9	rosa chinesis	0
10	rosa	1
11	dracaena trifasciata	0
12	chlorophytum comosum	0
13	astrophytum asterias	1

TRIGGER:

The trigger stores the plant details like plant ID, parent ID of the plant and the garden ID when the plant is removed from the cultivated_in table. It is assumed that when the plant dies, it is removed from the cultivated_in table. This helps in analyzing if there is any genetic pattern in deaths since the parent ID and garden ID is also stored .



```
create table dead_plant_record_table(plant_id varchar(30),parent_plant_id varchar(30),garden_id integer)
delimiter $$
create trigger dead_plants_record
before delete on cultivated_in for each row
begin
    insert into dead_plant_record_table
    select c.plant_id,p.parent_plant_id,c.garden_id
    from cultivated_in c left join  parent_plant p on p.plant_id=c.plant_id
    where c.plant_id=old.plant_id;
end
delimiter ;
delete from cultivated_in where plant_id="apple3";
select * from dead plant record table;
```



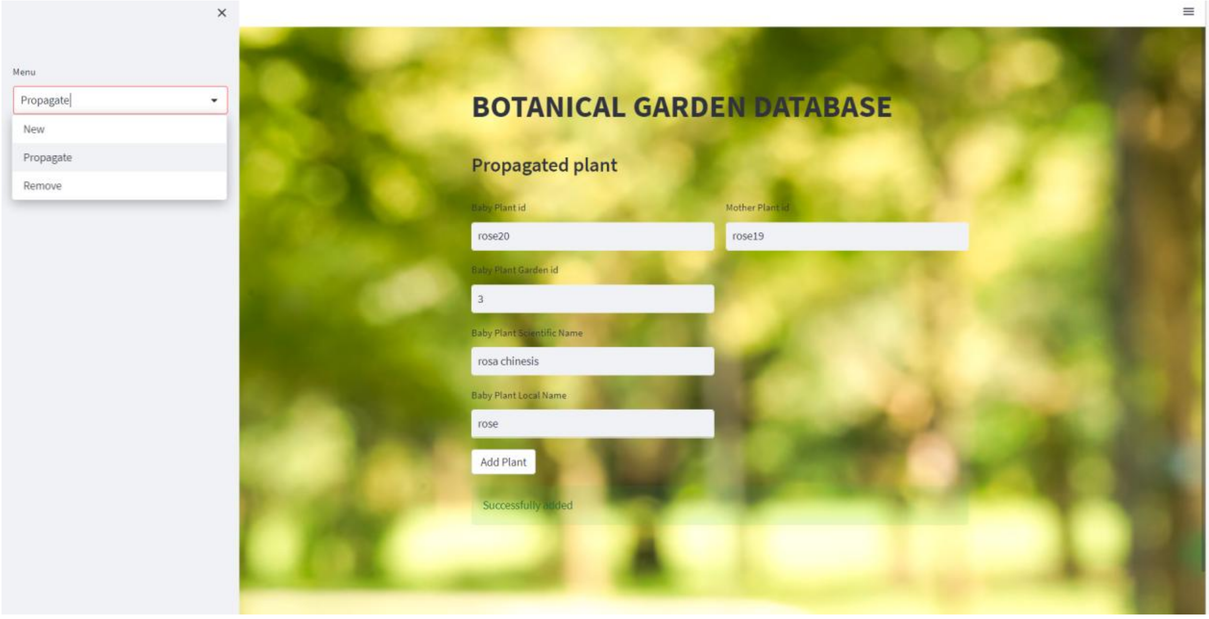
	plant_id	parent_plant_id	garden_id
1	apple3	apple2	3

FRONT-END:

The front-end has he following functionalities:

- Record added when a new plant comes to a garden.
- Record for propagated plant.
- Record deleted when the plant dies.
- Text box which runs SQL queries and displays the output.

1. Addition, Modification and Deletion of records from any chosen table



BOTANICAL GARDEN DATABASE

Propagated plant

Baby Plant id: rose20

Mother Plant id: rose19

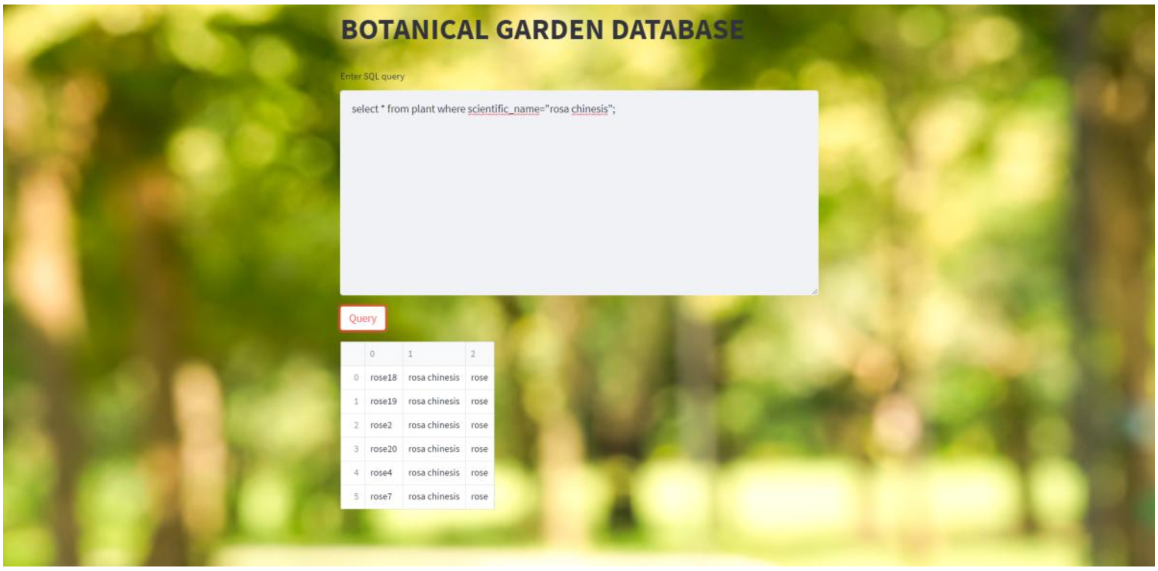
Baby Plant Garden id: 3

Baby Plant Scientific Name: rosa chinesis

Baby Plant Local Name: rose

Successfully added

2. There should be a window to accept and run any SQL statement and display the result



BOTANICAL GARDEN DATABASE

Enter SQL query

```
select * from plant where scientific_name="rosa chinesis";
```

	0	1	2
0	rose18	rosa chinesis	rose
1	rose19	rosa chinesis	rose
2	rose2	rosa chinesis	rose
3	rose20	rosa chinesis	rose
4	rose4	rosa chinesis	rose
5	rose7	rosa chinesis	rose