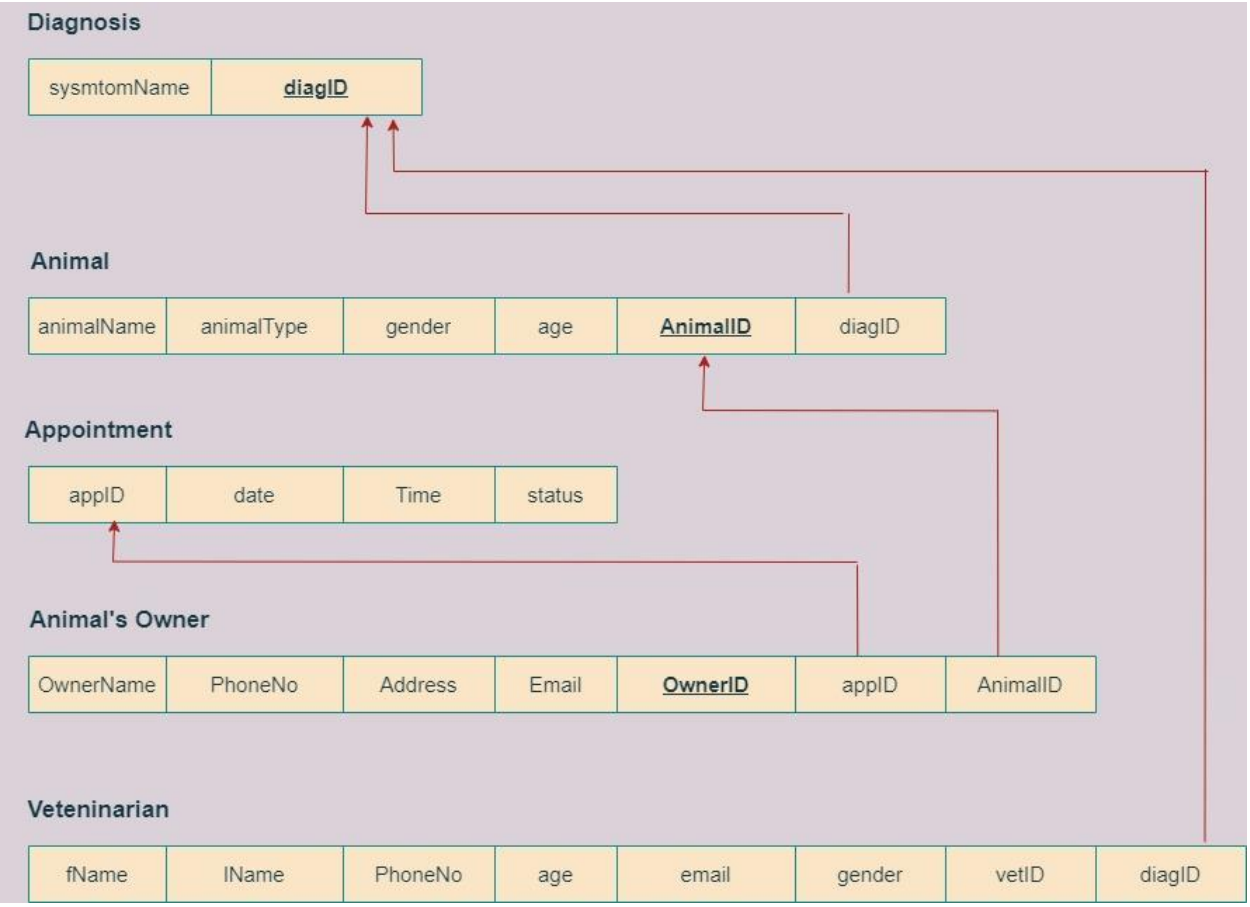
The Schema with applying Normalization Forms



Data Normalization Forms

Data normalization can be divided into different types of normal forms. The most popular ones are 1NF, 2NF, 3NF, and BCNF. Let us dive into all these normal forms with the help of an example. Assume that a company has a database of all their employees and their key skills as shown in the table below.

1NF - First Normal Form

The most basic form of data normalization is 1NF which ensures there are no two same entries in a group. For a table to be in the first normal form, it should satisfy the following rules:

- Each cell should contain a single value
- Each record should be unique

Tables

- Animal
- Animal_Owner
- Appointment
- Diagnosis
- Veteninarian

are all not have any multi-valued attribute so all cels contains a single value

and each record is unique as each record depend only on primary key .

so 1NF is applied successfully here .

2NF - Second Normal Form

In a 2NF table, all the subsets of data that can be placed in multiple rows are placed in separate tables. For a table to be in the second normal form, it should satisfy the following rules:

- It should be in 1F
- The primary key should not be functionally dependant on any subset of candidate key

so 2NF is applied successfully on our tables.

3NF - Third Normal Form

For a table to be in the third normal form, it should satisfy the following rules:

- It should be in 2F
- It should not have any transitive functional dependencies

A transitive functional dependency is when a change in a column (which is not a primary key) may cause any of the other columns to change.

So it applied successfully on our tables.

```
create database Veterinary_Database;

create table Animal(
AnimalID int primary key not null,
animalName varchar(25) not null ,
animalType varchar(15) not null ,
gender varchar(10) not null ,
age int not null,
Species varchar(30) not null
)
;
create table Appointment(
appID int primary key not null ,
appDate date not null ,
appTime time not null ,
status varchar(20)

);
create table Animal_Owner(
OwnerID int primary key not null,
OwerName varchar(25) not null ,
PhoneNo varchar(40) not null ,
Address varchar(50) not null ,
```

```
Email varchar(30) not null ,
AnimalID int not null,
appID int not null ,
FOREIGN KEY (appID) REFERENCES Appointment(appID),
FOREIGN KEY (AnimalID) REFERENCES Animal(AnimalID)
);
create table Diagnosis(
diagID int primary key not null,
symptomName varchar(30) not null ,
AnimalID int not null,
FOREIGN KEY (AnimalID) REFERENCES Animal(AnimalID)
);
create table Veteninarian(
vetID int primary key not null,
vetName varchar(25) not null ,
PhoneNo varchar(40) not null ,
age int not null ,
gender varchar(10) not null ,
Email varchar(30) not null ,
diagID int not null,
FOREIGN KEY (diagID) REFERENCES Diagnosis(diagID)
);
```

Insert Data into Animal

```
use Veterinary_Database;
insert into Animal values
(1,"Cat","Type1","Male",2,"Spece1"),
(2,"Horse","Type2","Male",2,"Spece2"),
(3,"Dog","Type1","Female",3,"Spece1"),
(4,"Horse","Type2","F",2,"Spece3");
```

Insert Data into Appoitment

```
use Veterinary_Database;
insert into Appointment values
(1,CURDATE(),curtime(),'open'),
(2,CURDATE(),curtime(),'closed'),
(3,'2008-7-04','13:11:22','opened');
```

Insert Data into Animal_Owner

```
insert  into Animal_Owner values
(1,"Ghada","966-00-111-11","Saudia-Arabia","Ghada@twitter.com",1,1),
(2,"Mohamed","966-00-111-25","England","Mohamed123@gmail.com",2,2),
(3,"Hadeel","966-22-008-11","France","Hadeel66@yahoo.com",3,2),
(4,"Hadeer","966-001-111-99","Egy","Hadeer123@twitter.com",4,3);
```

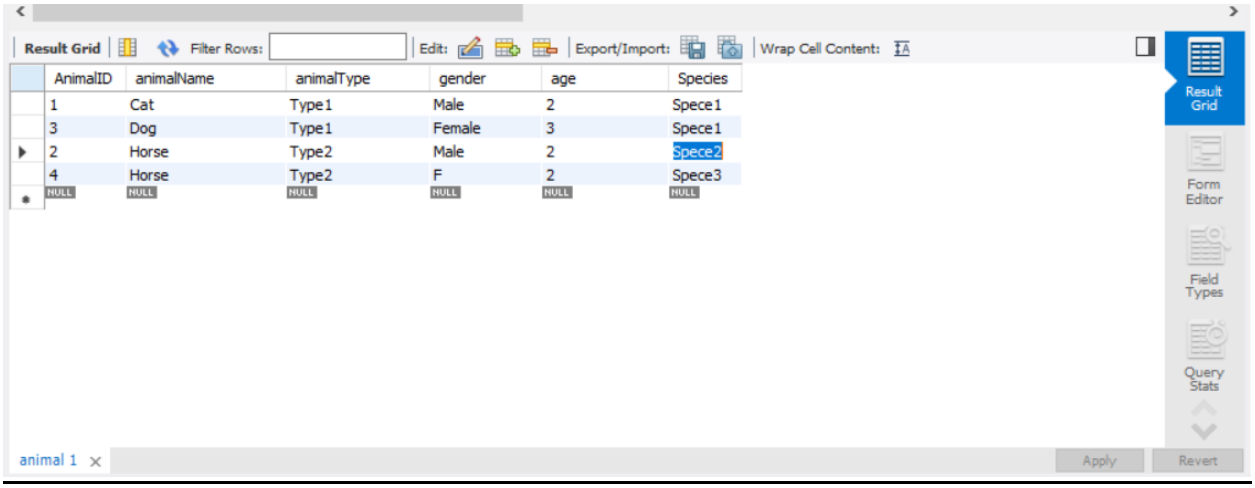
Insert Data into Diagnosis_

```
use Veterinary_Database;
insert into diagnosis values
(1,"diag1",2),
(2,"diag2",3),
(14,"diag14",4),
(6,"diag6",3),
(8,"diag8",2),
(10,"diag10",1);
```

Insert Data into Veteninarian

```
insert  into Veteninarian values
(1,"Ghada","966-00-111-11",30,'M',"Ghada@twitter.com",2),
(2,"Omar","966-12-111-11",30,'M',"Omar@hotmail.com",1),
(3,"Hagar","966-11-56-11",30,'F',"Hagar@gmail.com",8),
(4,"Rehab","966-00-11-11",30,'F',"Rehab456@facebook.com",2),
(5,"Omar","966-00-123-11",30,'M',"Omar123@hotmail.com",6),
(6,"Hagar","966-00-111-11",26,'F',"Hagar89@twitter.com",10);
```

Screenshots for Creating table

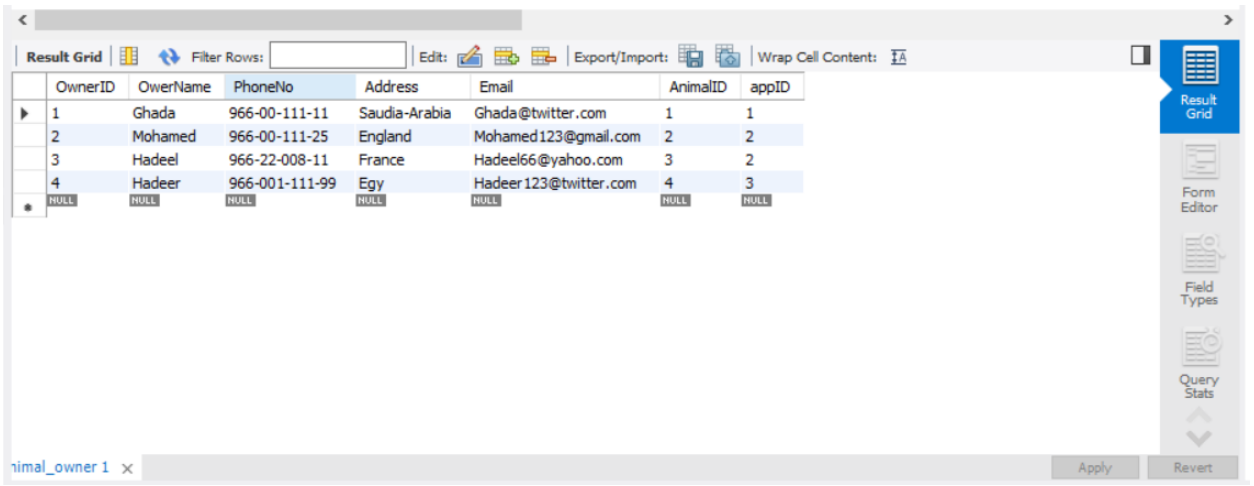


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

AnimalID	animalName	animalType	gender	age	Species
1	Cat	Type1	Male	2	Spece1
3	Dog	Type1	Female	3	Spece1
2	Horse	Type2	Male	2	Spece2
4	Horse	Type2	F	2	Spece3
NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats

animal 1 | Apply | Revert

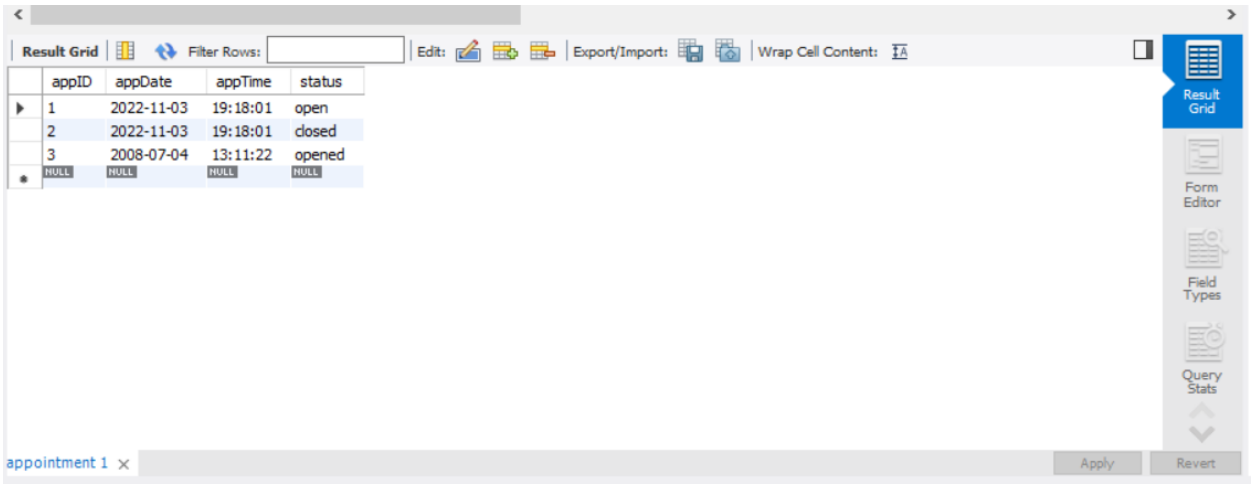


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

OwnerID	OwerName	PhoneNo	Address	Email	AnimalID	appID
1	Ghada	966-00-111-11	Saudia-Arabia	Ghada@twitter.com	1	1
2	Mohamed	966-00-111-25	England	Mohamed123@gmail.com	2	2
3	Hadeel	966-22-008-11	France	Hadeel66@yahoo.com	3	2
4	Hadeer	966-001-111-99	Egy	Hadeer123@twitter.com	4	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats

nimal_owner 1 | Apply | Revert

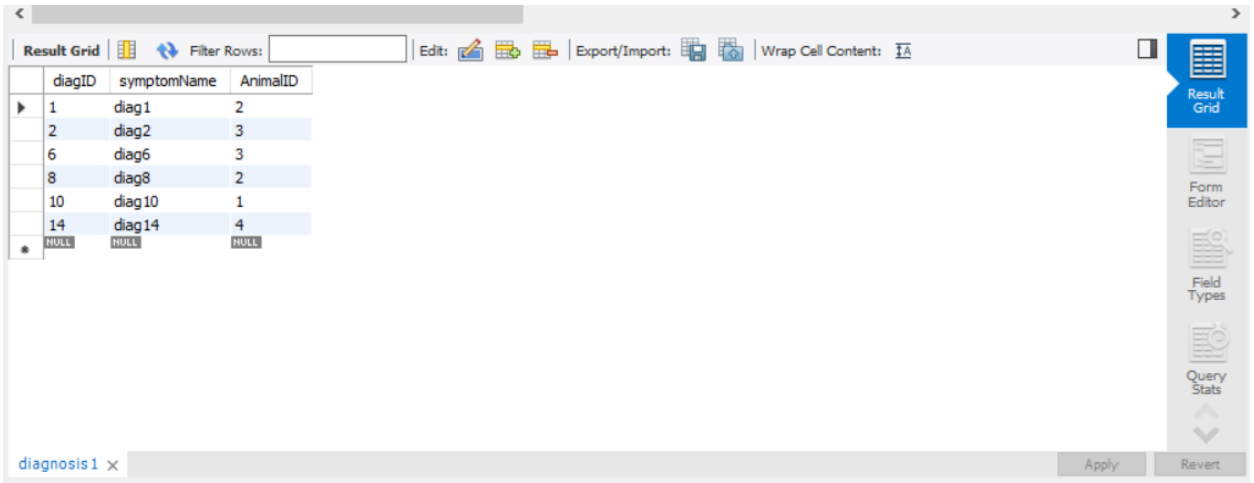


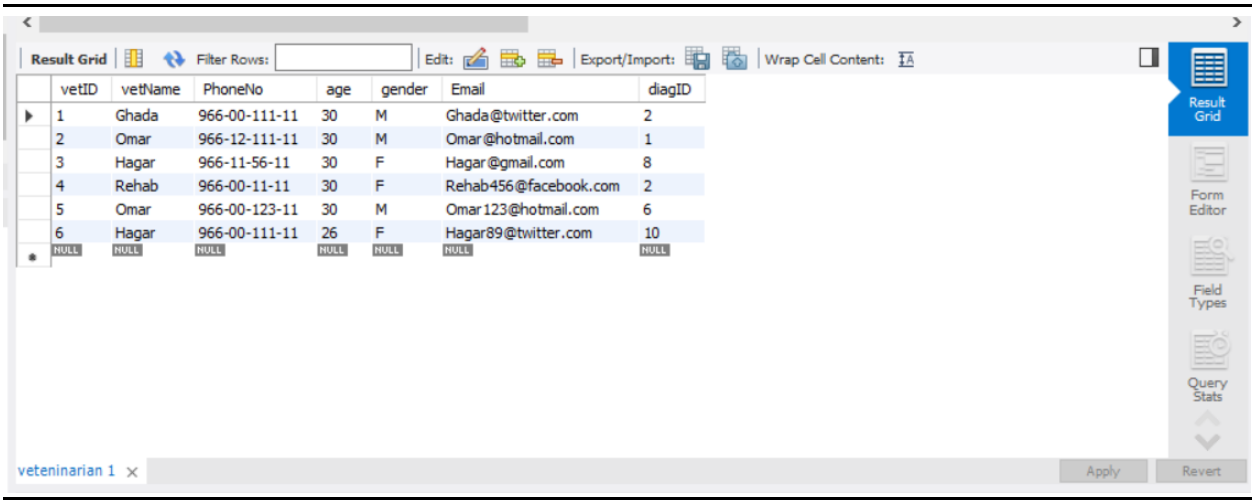
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

appID	appDate	appTime	status
1	2022-11-03	19:18:01	open
2	2022-11-03	19:18:01	closed
3	2008-07-04	13:11:22	opened
NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats

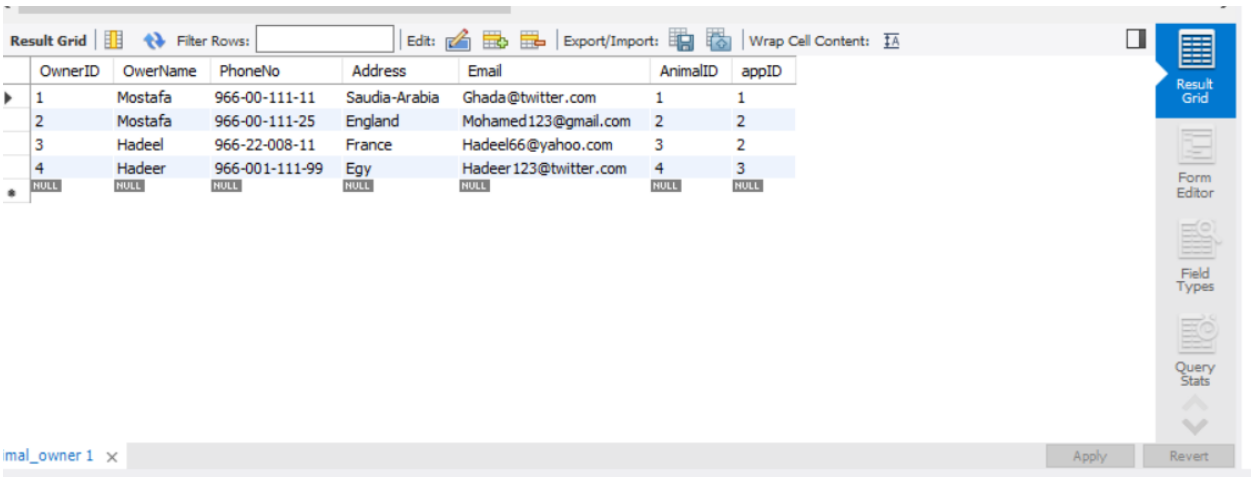
appointment 1 | Apply | Revert


Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
diagID symptomName AnimalID
1 diag1 2
2 diag2 3
6 diag6 3
8 diag8 2
10 diag10 1
14 diag14 4
NULL NULL NULL
Result Grid
Form Editor
Field Types
Query Stats
diagnosis 1
Apply
Revert


Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
vetID vetName PhoneNo age gender Email diagID
1 Ghada 966-00-111-11 30 M Ghada@twitter.com 2
2 Omar 966-12-111-11 30 M Omar@hotmail.com 1
3 Hagar 966-11-56-11 30 F Hagar@gmail.com 8
4 Rehab 966-00-11-11 30 F Rehab456@facebook.com 2
5 Omar 966-00-123-11 30 M Omar123@hotmail.com 6
6 Hagar 966-00-111-11 26 F Hagar89@twitter.com 10
NULL NULL NULL NULL NULL NULL NULL
Result Grid
Form Editor
Field Types
Query Stats
veteninarian 1
Apply
Revert

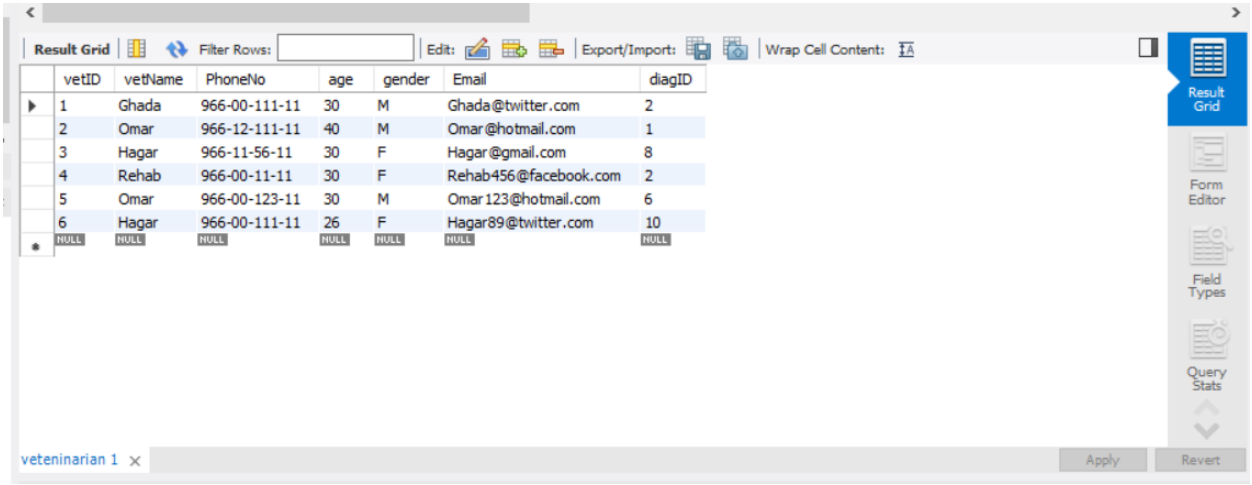
Update statements

update Animal_Owner set OwerName="Mostafa" where OwnerID=2 or OwnerID=1;



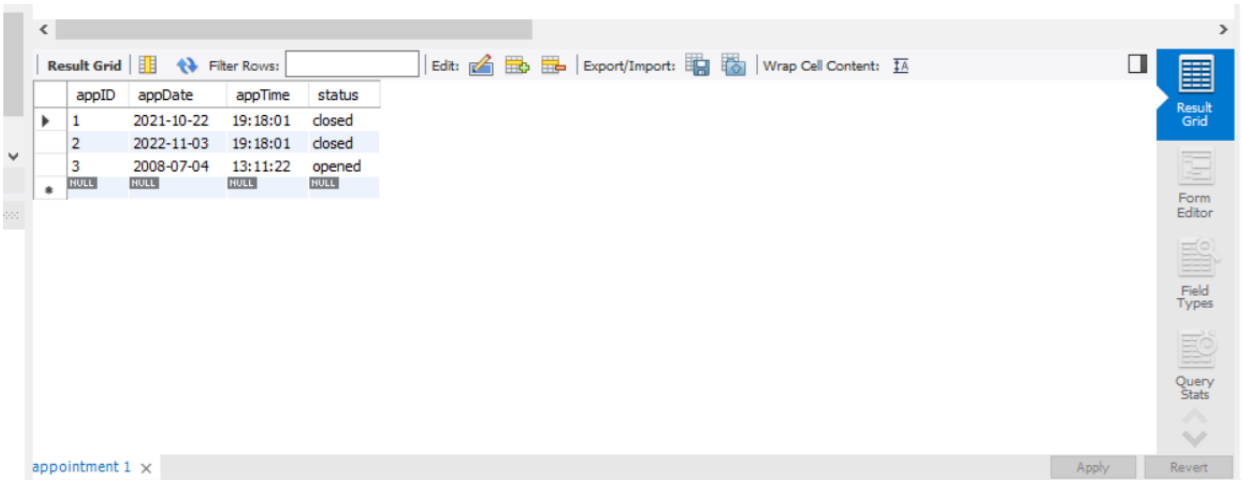
OwnerID	OwerName	PhoneNo	Address	Email	AnimalID	appID
1	Mostafa	966-00-111-11	Saudia-Arabia	Ghada@twitter.com	1	1
2	Mostafa	966-00-111-25	England	Mohamed123@gmail.com	2	2
3	Hadeel	966-22-008-11	France	Hadeel66@yahoo.com	3	2
4	Hadeer	966-001-111-99	Egy	Hadeer123@twitter.com	4	3

update vETENINARIAN set age = 40 where vetID=2;



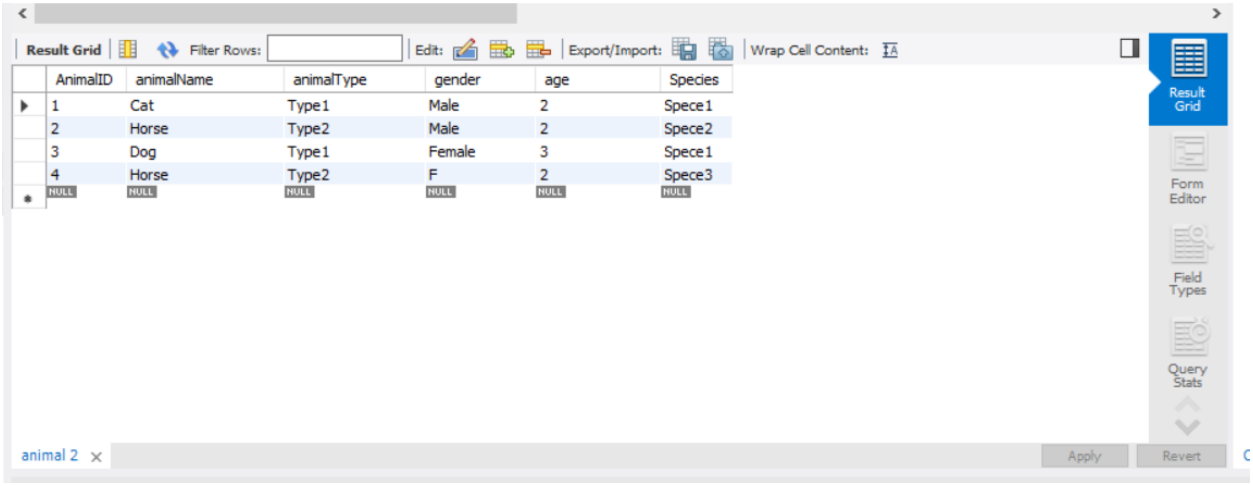
vetID	vetName	PhoneNo	age	gender	Email	diagID
1	Ghada	966-00-111-11	30	M	Ghada@twitter.com	2
2	Omar	966-12-111-11	40	M	Omar@hotmail.com	1
3	Hagar	966-11-56-11	30	F	Hagar@gmail.com	8
4	Rehab	966-00-11-11	30	F	Rehab456@facebook.com	2
5	Omar	966-00-123-11	30	M	Omar123@hotmail.com	6
6	Hagar	966-00-111-11	26	F	Hagar89@twitter.com	10

```
UPDATE appointment
SET appDate = '2021-10-22', status ='closed'
WHERE appID = 1;
```



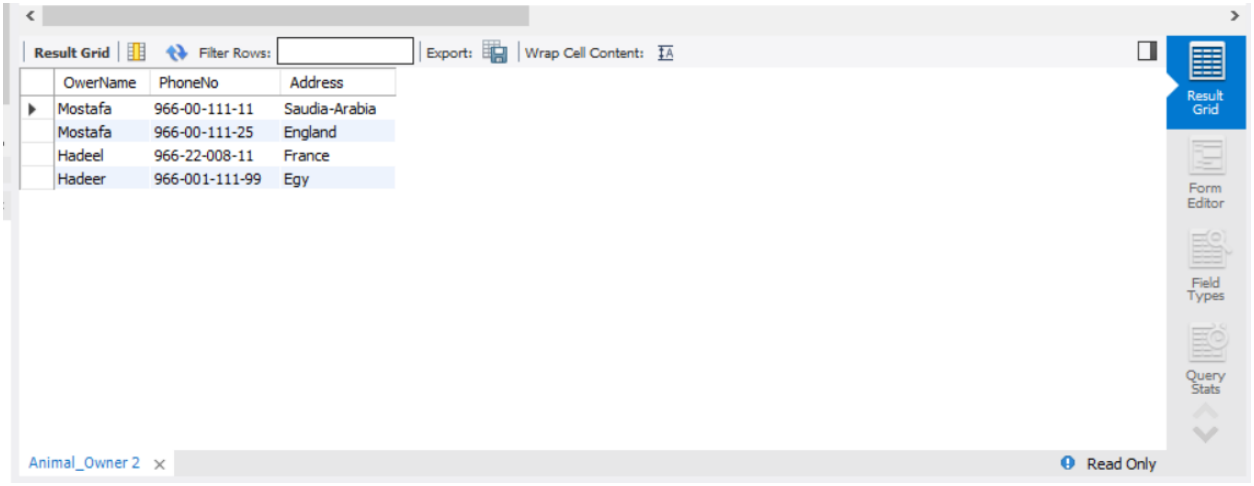
Select statements

select * from animal;



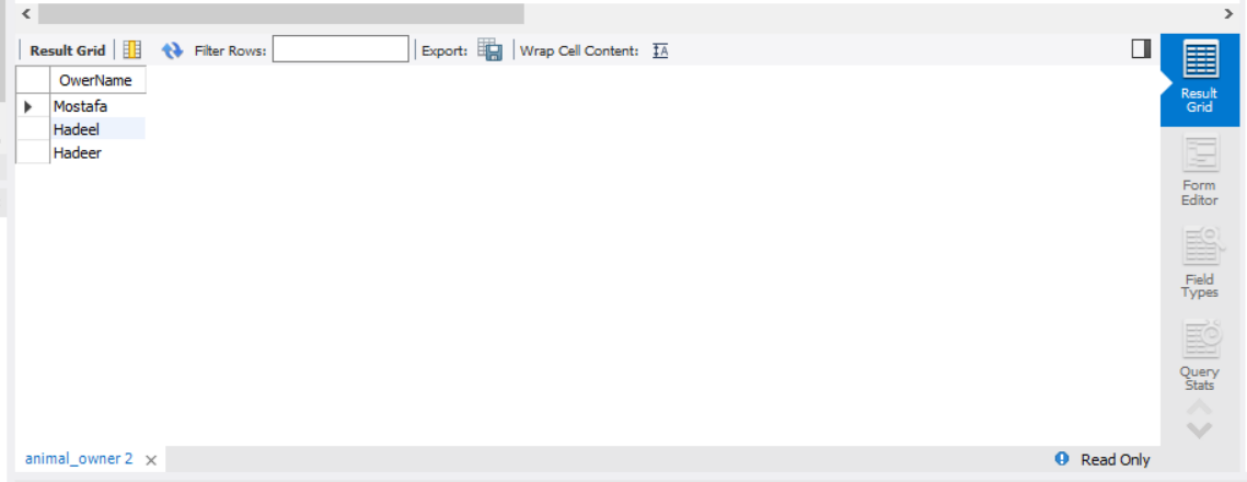
AnimalID	animalName	animalType	gender	age	Species
1	Cat	Type1	Male	2	Spece1
2	Horse	Type2	Male	2	Spece2
3	Dog	Type1	Female	3	Spece1
4	Horse	Type2	F	2	Spece3
NULL	NULL	NULL	NULL	NULL	NULL

select OwerName , PhoneNo , Address from Animal_Owner;



OwerName	PhoneNo	Address
Mostafa	966-00-111-11	Saudia-Arabia
Mostafa	966-00-111-25	England
Hadeel	966-22-008-11	France
Hadeer	966-001-111-99	Egy

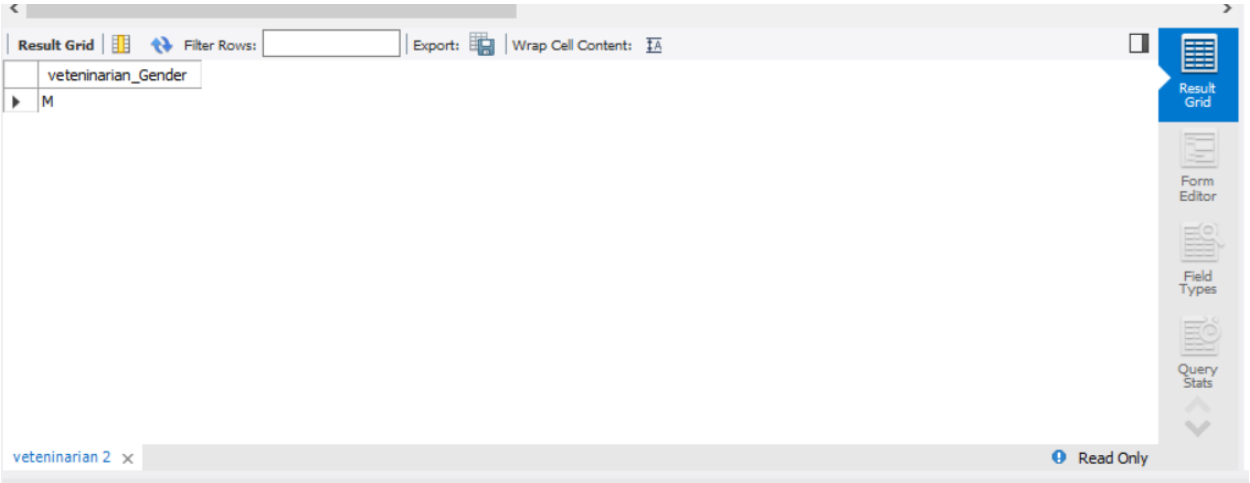
SELECT DISTINCT OwerName FROM animal_owner;



SELECT gender as 'veteninarian_Gender'

FROM veteninarian

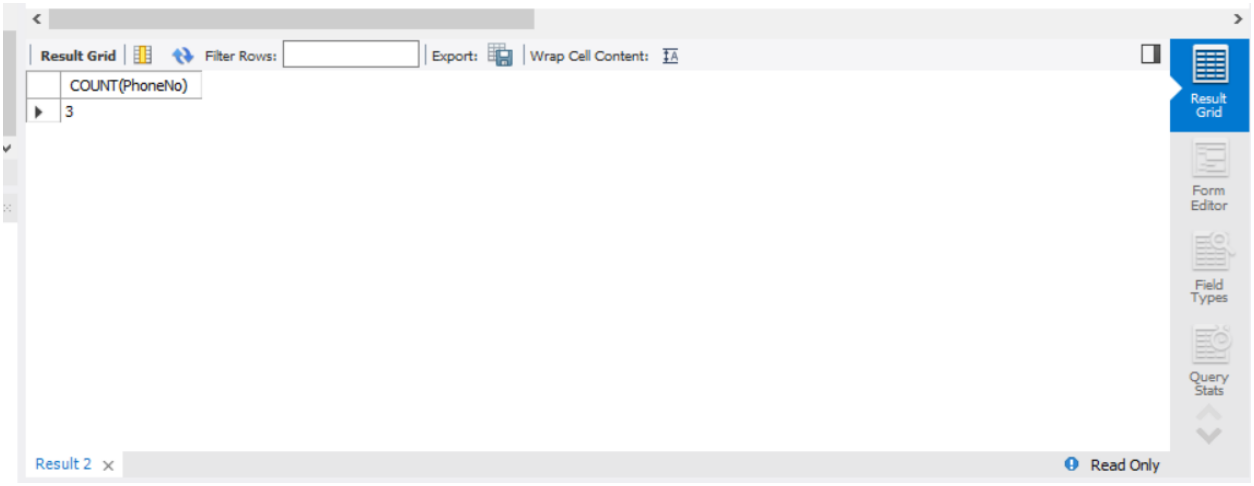
WHERE age >30;



SELECT COUNT(PhoneNo)

FROM animal_Owner

WHERE AnimalID >=2;



SELECT * FROM DIAGNOSIS

WHERE AnimalID NOT BETWEEN 2 AND 3;