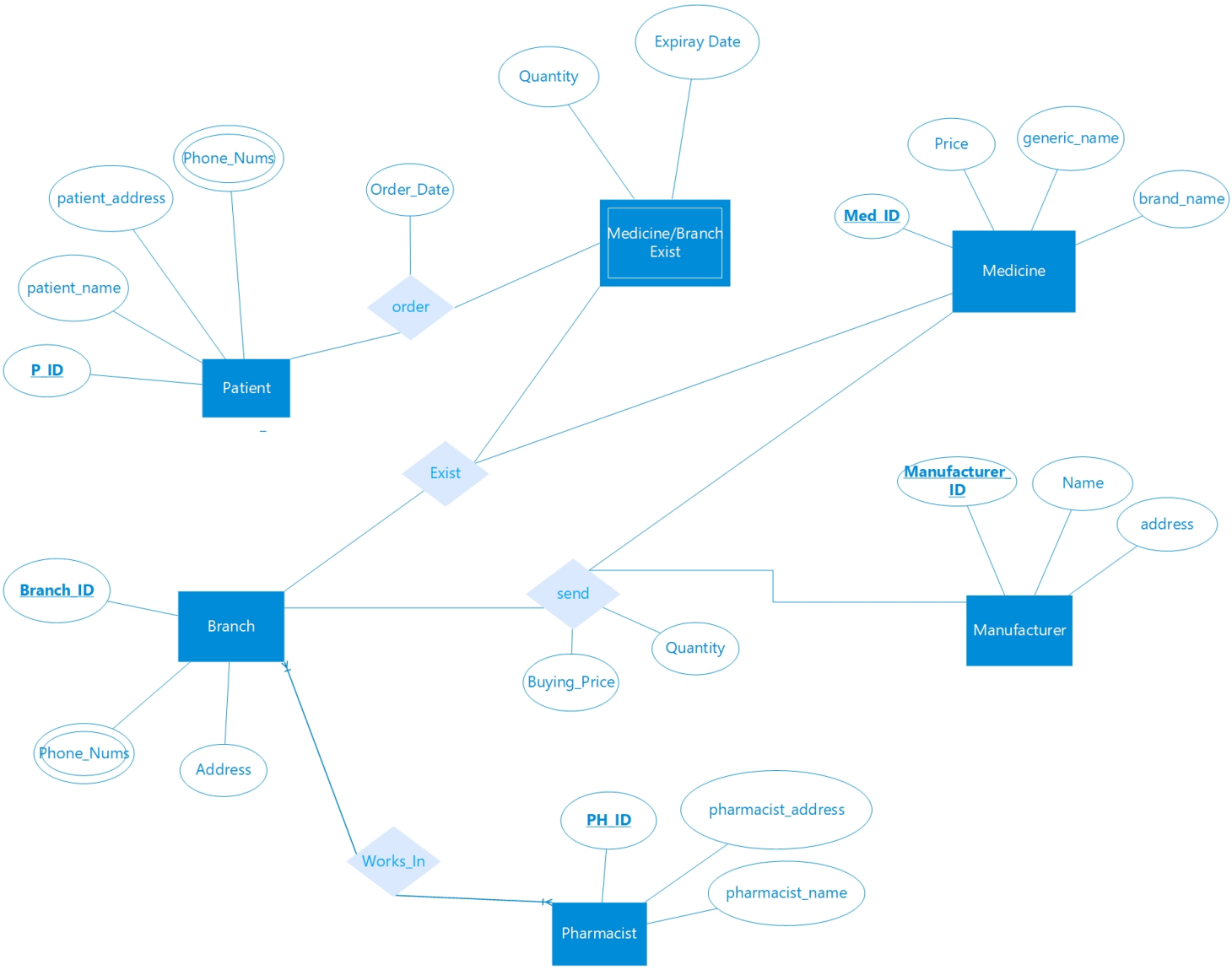
The database
project
2022-2023

---

the project's title : Pharmacy Database

description:

Entity diagram (ERD)



ERD description:

1. General:

The ERD consists of 6 Entities and their respective attributes and 4 relationships.
Entities are in blue, Relationships in gray, Attributes in white

2.Entities:

2.1.Patient

Attributes:

1.P_ID: a patient unique identifier (primary key)

2.patient_name

3.patient_address: where the patient lives.

4.Phone_Nums: a patient's phonenumber(s)

2.2.Medicine:

Attributes:

1.Med_ID: a medicine unique identifier (primary key)

2.generic_name: medicine's active ingredient that makes it work.

3.brand_name: given by the pharmaceutical company that markets the medicine.

4.Price: each medicine has its own price

2.3 .Branch

1.Branch_ID: a branch unique identifier (primary key)

2.Address: a branch's geographical location.

Phone_Nums: a branch's phonenumber(s)

2.4.Pharmacist :

Attributes:

1.PH_ID: a pharmacist unique identifier (primary key)

2.pharmacist_name

3.pharmacist_address: where a pharmacist live.

2.5.Medicine/Branch Exist

Attributes:

Quantity: a medicine exist in a branch with a certain quantity.

ExpiryDate: determined date after which medicine should no longer be used,

2.5.Manufacturer:

1.Manufacturer ID: a manufacturer unique identifier.

2.Name: a manufacturer name.

3.Address: a manufacturer geographical location.

3.Relationships:

3.1.Works_In: a pharmacist works in a certain branch.

3.2.Exist: a medicine exist in a branch with a certain quantity.

3.3:Order: a patient orders medicine from existing ones

Attributes:

1.Order_Date.

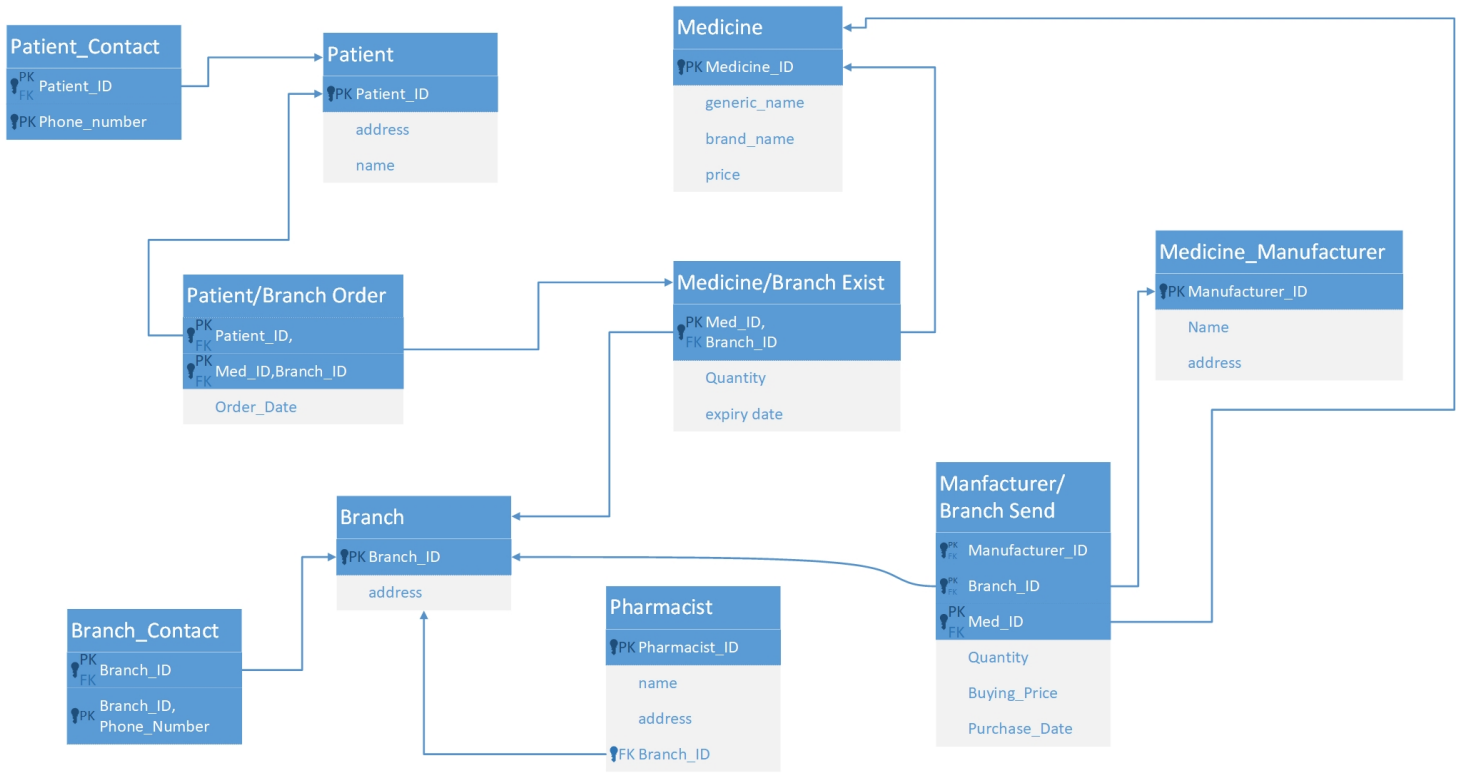
3.4.Send: a manufacturer send a medicine to a branch.

Attributes:

1.Quantity: number sent of certain medicine.

2.Buying_Price: the sum or amount of money for which a medicine is bought.

Relational Schema



SQL

1: Table Creation

1.1: Base Entities

```
Create table Patient(
patient_ID int primary key identity,
patient_address varchar(100),
patient_name varchar(868),
);

Create table Pharmacist(
pharmacist_ID int primary key identity,
pharmacist_name varchar(868),
pharmacist_address varchar(100),

branch_ID int,
);

Create table Medicine(
med_ID int primary key identity,
generic_name varchar(50),
brand_name varchar(50) not null,
price decimal (5,2),
);

Create table Branch(
branch_ID int primary key identity,
branch_address varchar(100),
);

Create table Manufacturer(
manufacturer_ID int primary key identity,
manufacturer_name varchar(50) unique not null,
manufacturer_address varchar(100),
);
```

1.2 : MultiValue attributes

```
Create table Patient_Contact(
patient_ID int,
phone_number varchar(15),
primary key(patient_ID,phone_number),
);

Create table Branch_Contact(
branch_ID int,
phone_number varchar(15),
primary key(branch_ID,phone_number),
);
```

1.3: Relationships

```
Create table Patient_Order(
patient_ID int,
med_ID int,
 int,
order_date date,
primary key(patient_ID,med_ID,branch_ID),
);

Create table Med_Exist(
med_ID int,
branch_ID int,
quantity int,
expiryDate date,
primary key(med_ID,branch_ID)
);

Create table Manufacturer_Send(
manufacturer_ID int,
branch_ID int,
med_ID int,
med_quantity int,
buying_price decimal(5,2),
purchase_date date,
primary key(manufacturer_ID,branch_ID,med_ID),
);
```

2.Defining Constraints (PK, FK)

```
alter table Pharmacist
add constraint Pharmacist_Branch_FK foreign key(branch_ID) references Branch;


alter table Med_Exist
add constraint med_exsitingMed_FK foreign key(med_ID) references Medicine;

alter table Med_Exist
add constraint exsitingMed_branch_FK foreign key(branch_ID) references Branch;

--Patient Order from Existing_Medicine in a branch
alter table Patient_Order
add constraint patient_order_FK foreign key(patient_ID) references patient;

alter table Patient_Order
add constraint order_existingMed_FK foreign key(med_ID,branch_ID) references Med_Exist;


alter table Manufacturer_Send
add constraint manufacturer_Send_FK foreign key(manufacturer_ID) references Manufacturer;

alter table Manufacturer_Send
add constraint Send_Medicine_FK foreign key(med_ID) references Medicine;

alter table Manufacturer_Send
add constraint Medicine_Branch_FK foreign key(branch_ID) references Branch;


alter table Patient_Contact
add constraint patientContact_patient foreign key(patient_ID) references Patient;

alter table Branch_Contact
add constraint branchContact_Branch foreign key(branch_ID) references Branch;
```

3.Queries.

3.1 : selecting the name and number of different medicines in each branch (retrieve data using join).

```
select B.branch_ID, M.brand_name , quantity
from Med_Exist as ME
join Medicine as M
on M.med_ID = ME.med_ID
join Branch as B
on B.branch_ID = ME.branch_ID;
```

3.2: selecting Pharmacists names and which branch they work on (retrieve data using subqueries).

```
select Ph.pharmacist_name, branch_ID
from Pharmacist as Ph
where Ph.branch_ID IN(
select B.branch_ID from
Branch as B
)
```

4.Aggregate Functions.

4.1 : Pharmacist expenses.

```
select SUM(buying_price) as 'Pharmacist expenses'
from Manufacturer_Send;
```

4.2 : Pharmacist revenues

```
select SUM(Price) as 'Pharmacist revenues'
from Orders;
```

4.3: Average medicine price

```
select AVG(price) as 'Average Medicine Price'
from Medicine;
```

4.4: (retrieve data using a group by and having).

4.4.1 : certain Medicine expenses.

```
select med_ID,SUM(buying_price) as 'Medicine expenses'
from Manufacturer_Send
group by med_ID;
```

4.4.2: Certain Medicine revenues.

```
select Medicine,SUM(Price) as 'revenues'
from Orders
group by(Medicine);
```

5.Views.

```
Create view Orders
as
select patient_name 'Customer Name', brand_name 'Medicine', price  as 'Price', branch_address as 'Pharmacy Branch'
from Patient_Order as PO
join Patient as P
    on PO.patient_ID = P.patient_ID
join Branch as B
    on B.branch_ID = PO.branch_ID
join Med_Exist as ME
    on ME.med_ID = PO.med_ID AND ME.branch_ID = B.branch_ID
join Medicine as M
    on M.med_ID = ME.med_ID
```

6.Stored procedures.

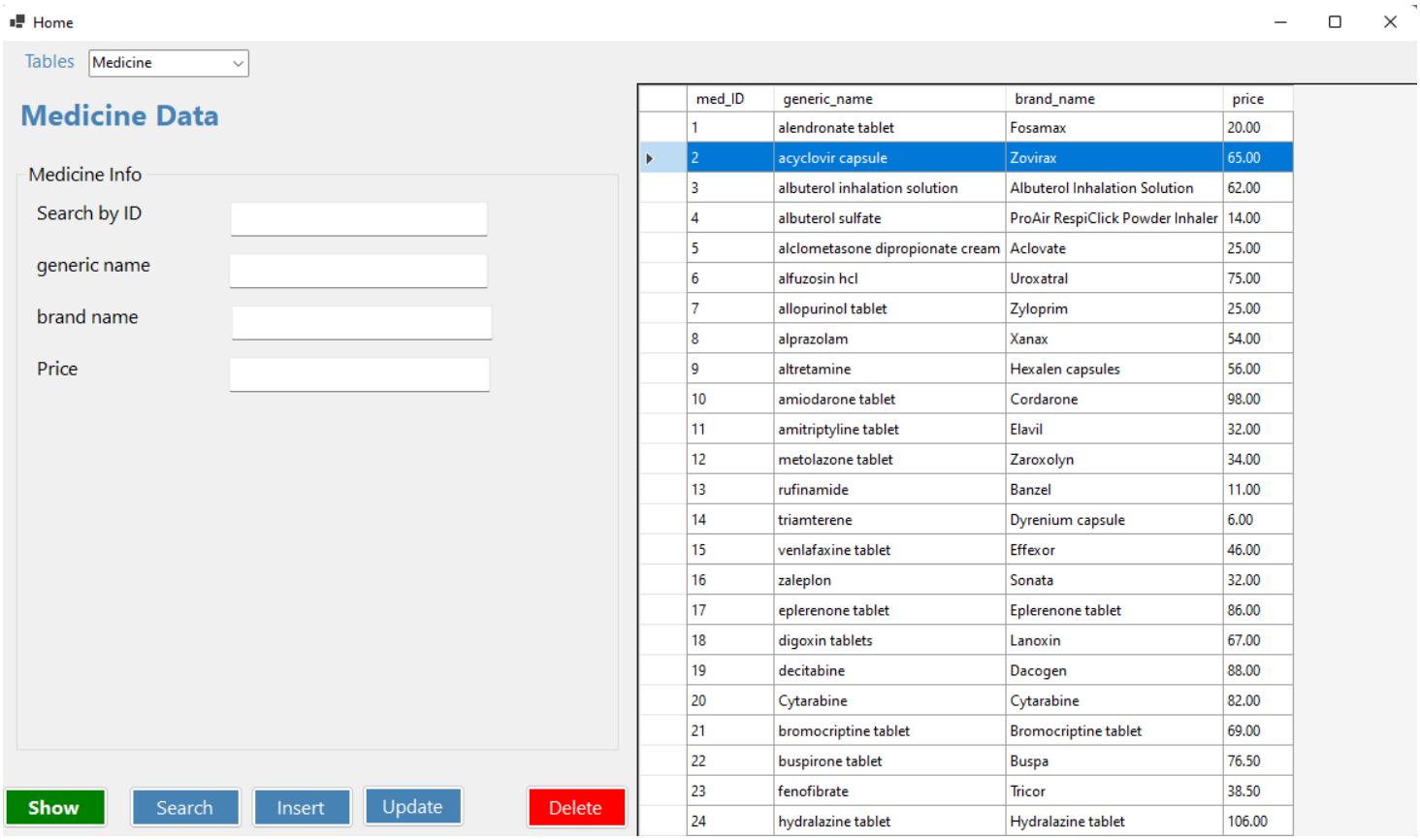
```
create proc OrderMedicine(
@patientID int,
@medID int,
@branchID int,
@orderDate date
)
as
begin

Insert into Patient_Order
Values (@patientID,@medID,@branchID,@orderDate);

end
```

---

7. User Interface (UI)



med_ID	generic_name	brand_name	price
1	alendronate tablet	Fosamax	20.00
2	acyclovir capsule	Zovirax	65.00
3	albuterol inhalation solution	Albuterol Inhalation Solution	62.00
4	albuterol sulfate	ProAir RespiClick Powder Inhaler	14.00
5	alclometasone dipropionate cream	Aclovate	25.00
6	alfuzosin hcl	Uroxatral	75.00
7	allopurinol tablet	Zyloprim	25.00
8	alprazolam	Xanax	54.00
9	altretamine	Hexalen capsules	56.00
10	amiodarone tablet	Cordarone	98.00
11	amitriptyline tablet	Elavil	32.00
12	metolazone tablet	Zaroxolyn	34.00
13	rufinamide	Banzel	11.00
14	triamterene	Dyrenium capsule	6.00
15	venlafaxine tablet	Effexor	46.00
16	zaleplon	Sonata	32.00
17	eplerenone tablet	Eplerenone tablet	86.00
18	digoxin tablets	Lanoxin	67.00
19	decitabine	Dacogen	88.00
20	Cytarabine	Cytarabine	82.00
21	bromocriptine tablet	Bromocriptine tablet	69.00
22	buspirone tablet	Buspa	76.50
23	fenofibrate	Tricor	38.50
24	hydralazine tablet	Hydralazine tablet	106.00