

# Medlife Pharmacy

Group: Project\_0502\_01

Presented by: Yash Makadia

Date: 7th December 2022





# Agenda





# Background

There are three main sources of data that are being managed in this system at Medlife Pharmacy:

1. The data coming from different Pharmaceutical companies that manufacture drugs.
2. The Employees who work at Medlife
3. The customers who purchase drugs from our stores

As a Company that sources its products from different locations and sells it at different locations, Medlife Pharmacy needs to keep track of it's inventory purchases and sales. At a smaller scale, it is easy but Medlife has stores all over the country and thus needs to ensure that the data is tabulated and managed in an appropriate manner.

# Introduction

Medlife Pharmacy is a system that stores data related to medication use processes within pharmacies in an orderly and systematic manner.

## Mission Statement

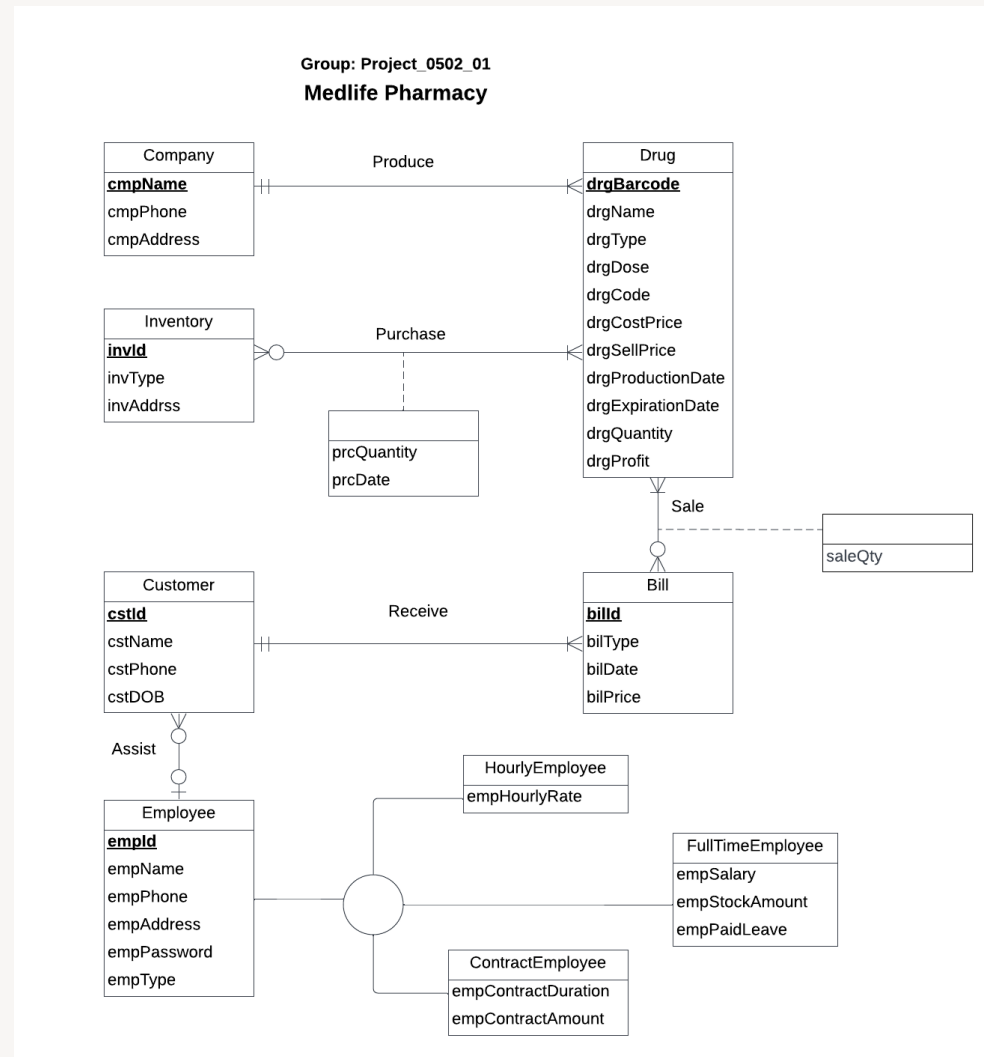
The main aim of this project is to develop the system such that the pharmaceutical store can effectively manage its behemoth data and stay ahead of the curve.

## Mission Objective

- To develop a system that deals with the day to day requirement of any pharmacy.
- To ensure safety, accuracy, and efficiency within the pharmaceutical store.
- To provide competitive advantage to the pharmacy.



# Conceptual Database Design



# Logical Database Design

Company (cmpName, cmpPhone, cmpAddress)

Drug (drgBarcode, drgName, drgType, drgDose, drgCode, drgCostPrice, drgSellPrice, drgProductionDate, drgExpirationDate, drgQuantity, drgProfit, *cmpName*)

Inventory(invId, invType, invAddress)

Employee (empld, empName, empPhone, empAddress, empPassword, empType)

HourlyEmployee (empld, empHourlyRate)

FullTimeEmployee (empld, empSalary, empStockAmount, empPaidLeave)

ContractEmployee (empld, empContractDuration, empContractAmount)

Customer (cstId, cstName, cstPhone, cstDOB, *empld*)

Bill (billd, bilType, bilPrice, bilDate, *cstId*)

Purchase (invId, drgBarcode, prcQuantity, prcDate)

Sale (drgBarcode, billd, saleQty)

# Physical Database Design

1. CREATE TABLE Company (  
    cmpName VARCHAR (30) NOT NULL,  
    cmpPhone CHAR(10),  
    cmpAddress VARCHAR(50),  
    CONSTRAINT pk\_Company\_cmpName PRIMARY KEY (cmpName))
1. CREATE TABLE Drug (  
    drgBarcode CHAR (10) NOT NULL,  
    drgName VARCHAR(30),  
    drgType VARCHAR(20),  
    drgDose INTEGER,  
    drgCode VARCHAR(10),  
    drgCostPrice DECIMAL(6,2),  
    drgSellPrice DECIMAL(6,2),  
    drgProductionDate DATE,  
    drgExpirationDate DATE,  
    drgQuantity INTEGER,  
    drgProfit DECIMAL(6,2),  
    cmpName VARCHAR (30) NOT NULL,  
    CONSTRAINT pk\_Drug\_drgBarcode PRIMARY KEY (drgBarcode),  
    CONSTRAINT fk\_Drug\_cmpName FOREIGN KEY (cmpName )  
        REFERENCES Company (cmpName )  
        ON DELETE NO ACTION ON UPDATE CASCADE)

# Physical Database Design

## 3. CREATE TABLE Inventory (

invId CHAR(10) NOT NULL,

invType VARCHAR(20),

invAddress VARCHAR (50),

CONSTRAINT pk\_Inventory\_invId PRIMARY KEY (invId))

## 4. CREATE TABLE Employee (

empId VARCHAR (10) NOT NULL,

empName VARCHAR (20),

empPhone CHAR (12),

empAddress VARCHAR (50),

empPassword VARCHAR (20),

emp\_type VARCHAR(20),

CONSTRAINT pk\_Employee\_empId PRIMARY KEY (empId) )



# Physical Database Design

5. CREATE TABLE Customer (

cstId VARCHAR (10) NOT NULL,

cstName VARCHAR (20),

cstPhone CHAR (10),

cstDOB DATE,

empId VARCHAR (10) NOT NULL,

CONSTRAINT pk\_Customer\_cstId PRIMARY KEY (cstId),

CONSTRAINT fk\_Customer\_cmpName FOREIGN KEY (empId)

REFERENCES Employee (empId)

ON DELETE NO ACTION ON UPDATE CASCADE)

# Physical Database Design

6. CREATE TABLE Bill (

    bilId CHAR (10) NOT NULL,

    bilType VARCHAR (20),

    bilDate DATE,

    bilPrice DECIMAL (8,2),

    cstId VARCHAR (10) NOT NULL,

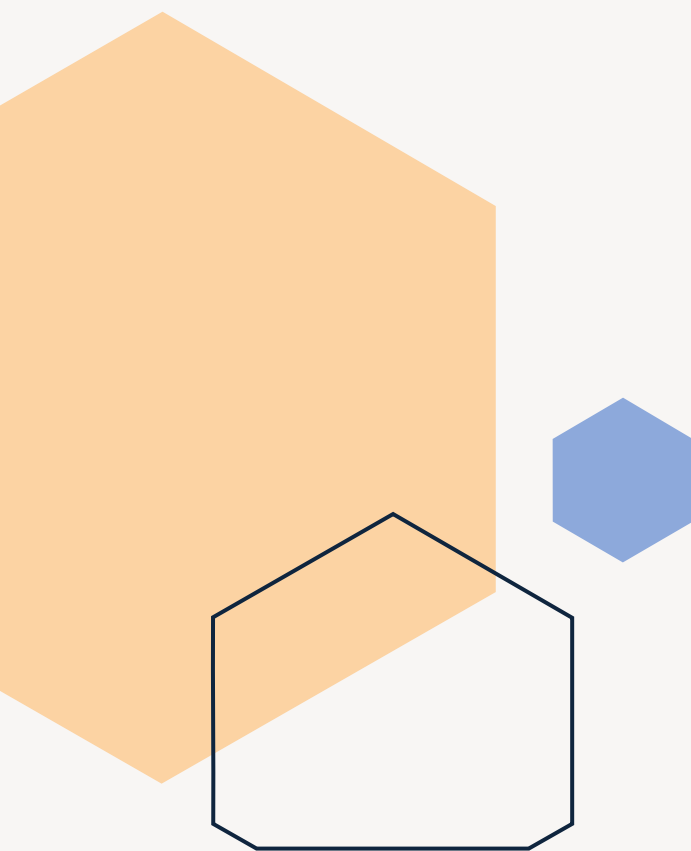
    CONSTRAINT pk\_Bill\_bilId PRIMARY KEY (bilId),

    CONSTRAINT fk\_Bill\_cstId FOREIGN KEY (cstId)

        REFERENCES Customer (cstId)

        ON DELETE NO ACTION ON UPDATE CASCADE )

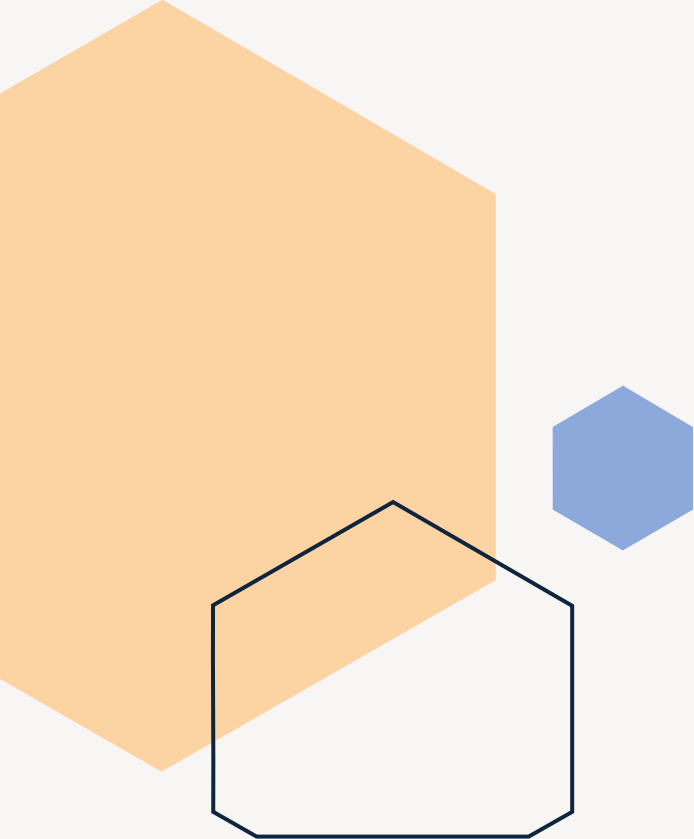
# Physical Database Design



```
7. CREATE TABLE HourlyEmployee(  
    empId VARCHAR (10) NOT NULL,  
    empHourlyRate DECIMAL (6,2),  
    CONSTRAINT pk_HourlyEmployee_empId PRIMARY KEY (empId),  
    CONSTRAINT fk_HourlyEmployee_cmpName FOREIGN KEY (empId)  
        REFERENCES Employee (empId)  
        ON DELETE CASCADE ON UPDATE CASCADE)
```

```
8. CREATE TABLE FullTimeEmployee (  
    empId VARCHAR (10) NOT NULL,  
    empSalary DECIMAL (10,2),  
    empStockAmount DECIMAL (9,2),  
    empPaidLeave INTEGER,  
    CONSTRAINT pk_FullTimeEmployee_empId PRIMARY KEY (empId),  
    CONSTRAINT fk_FullTimeEmployee_cmpName FOREIGN KEY (empId)  
        REFERENCES Employee (empId)  
        ON DELETE CASCADE ON UPDATE CASCADE)
```

# Physical Database Design



```
9. CREATE TABLE ContractEmployee (  
    empId VARCHAR (10) NOT NULL,  
    empContractDurationMonths INTEGER,  
    empContractAmount DECIMAL (9,2),  
    CONSTRAINT pk_ContractEmployee_empId PRIMARY KEY (empId),  
    CONSTRAINT fk_ContractEmployee_cmpName FOREIGN KEY (empId)  
        REFERENCES Employee (empId)  
    ON DELETE CASCADE ON UPDATE CASCADE)
```

# Physical Database Design

10. CREATE TABLE Purchase (

invId CHAR (10) NOT NULL,

drgBarcode CHAR (10) NOT NULL,

prcQuantity INTEGER,

prcDate DATE,

CONSTRAINT pk\_Purchase\_invId\_drgBarcode PRIMARY KEY (invId, drgBarcode),

CONSTRAINT fk\_Purchase\_invId FOREIGN KEY (invId)

REFERENCES Inventory (invId)

ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT fk\_Purchase\_drgBarcode FOREIGN KEY (drgBarcode)

REFERENCES Drug (drgBarcode)

ON DELETE CASCADE ON UPDATE CASCADE)

# Physical Database Design

11. CREATE TABLE Sale (

drgBarcode CHAR (10) NOT NULL,

billd CHAR (10) NOT NULL,

saleQty INTEGER,

CONSTRAINT pk\_Sale\_drgBarcode\_billd PRIMARY KEY (drgBarcode, billd),

CONSTRAINT fk\_Sale\_drgBarcode FOREIGN KEY (drgBarcode)

REFERENCES Drug (drgBarcode)

ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT fk\_Sale\_billd FOREIGN KEY (billd)

REFERENCES Bill (billd)

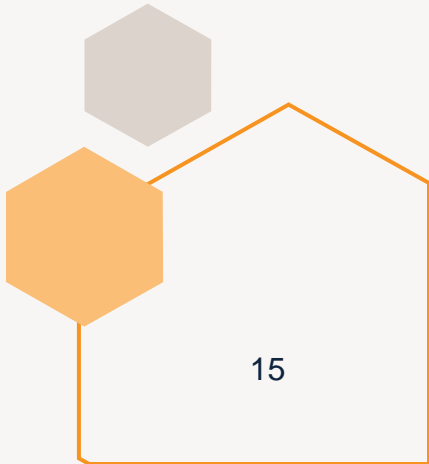
ON DELETE CASCADE ON UPDATE CASCADE)

# Business Transactions

1. As part of our customer experience, we would like to know how our employees are doing with customer assistance. Are customers who purchase from our pharmacy, helped by Full-time employees or part-time (Hourly)?

```
SELECT e.empType AS 'Employee Type' , COUNT(e.empId) AS 'Number of Customers'  
FROM Employee e where empId IN ( SELECT c.empId FROM Customer c,  
Bill b WHERE c.cstId = b.cstId)  
GROUP BY e.empType  
ORDER BY COUNT(e.empId) DESC;
```

Employee Type	Number of Customers
Contract	8
FullTime	6
Hourly	2

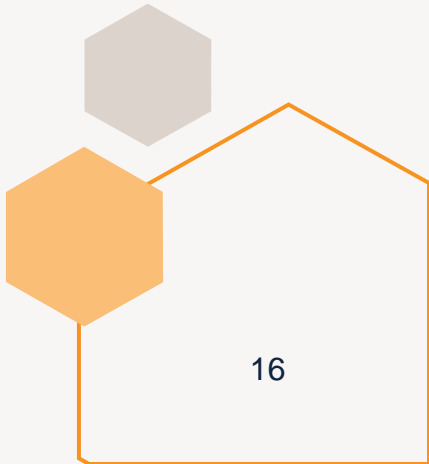


# Business Transactions

2. For the purpose of efficient Inventory management, the manager wants to know the purchase quantity of drugs for any particular month and year. This will help the manager in determining which drugs to buy in bulk and also determine the kind of contracts he/she wants to make with drug selling companies.

```
SELECT d.drgName AS 'Drug Name', p.drgBarcode AS 'Drug Barcode', d.cmpName AS 'Company Name',  
SUM(p.prcQuantity) AS 'Purchase Quantity'  
FROM Purchase p, Drug d  
WHERE YEAR(p.prcDate) = '2021' AND MONTH(p.prcDate) = '09' AND p.drgBarcode = d.drgBarcode  
GROUP BY p.drgBarcode, d.drgName, d.cmpName  
ORDER BY SUM(p.prcQuantity) DESC;
```

Drug Name	Drug Barcode	Company Name	Purchase Quantity
Levothyroxine	1112908901	AbbVie Inc	30
Metformin	1112908902	Johnson & Johnson	30
Atorvastatin	1112908900	Pfizer Inc	15





# Business Transactions

3. The consumption pattern of drugs changes yearly. With the rise in variety of diseases and global outbreaks, the most-selling drugs are different for every year. The Sales manager wants to see the most-selling drugs every year to analyze the trends and patterns and for forecasting purposes.

```
SELECT s.drgBarcode AS 'Drug Barcode', d.drgName AS 'Drug Name', d.cmpName AS  
'Company Name', YEAR(b.bilDate) AS 'Year of Sale', SUM(s.saleQty) AS 'Sale Quantity'  
FROM Drug d, Sale s, Bill b  
WHERE (Year(b.bilDate) IN (SELECT Year(b.bilDate) FROM Bill b GROUP BY YEAR(b.bilDate) )  
AND (s.drgBarcode = d.drgBarcode AND s.billd = b.billd))  
GROUP BY s.drgBarcode, d.drgName, d.cmpName, YEAR(b.bilDate)  
ORDER BY SUM(s.saleQty) DESC, d.drgName;
```

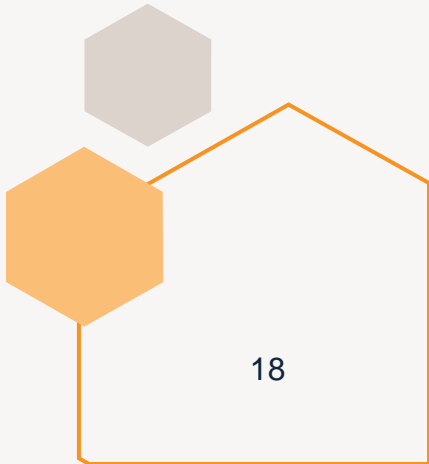
Drug Barcode	Drug Name	Company Name	Year of Sale	Sale Quantity
1112908902	Metformin	Johnson & Johnson	2022	87
1112908901	Levothyroxine	AbbVie Inc	2022	53
1112908900	Atorvastatin	Pfizer Inc	2022	30
1112908903	Fosinopril	Pfizer Inc	2022	24
1112908900	Atorvastatin	Pfizer Inc	2021	20
1112908901	Levothyroxine	AbbVie Inc	2021	20
1112908902	Metformin	Johnson & Johnson	2021	20
1112908904	Captopril	AbbVie Inc	2022	19
1112908904	Captopril	AbbVie Inc	2021	12

# Business Transactions

4. Profitability is an important part of any business. Medlife wants to estimate which drugs are the most profitable for them. This will help the Sales manager and the Relationship manager to enter into long-term contracts for companies whose drugs are sold in highest number by Medlife.

```
SELECT d.drgType AS 'Drug Type', d.drgName AS 'Drug Name', d.cmpName
AS 'Company
Name', SUM(s.saleQty) AS 'Sale Quantity',
((SUM(d.drgProfit)*SUM(s.saleQty))/COUNT(s.drgBarcode)) AS 'Total
Profit'
FROM Drug d, Sale s
WHERE d.drgBarcode = s.drgBarcode
GROUP BY d.drgType, d.cmpName, d.drgName
ORDER BY d.drgType ASC , SUM(s.saleQty) DESC;
```

Drug Type	Drug Name	Company Name	Sale Quantity	Total Profit
Biguanides	Metformin	Johnson & Johnson	107	1337.500000
Hormone	Levothyroxine	AbbVie Inc	73	518.300000
Inhibitor	Atorvastatin	Pfizer Inc	50	325.000000
Inhibitor	Captopril	AbbVie Inc	31	313.100000
Inhibitor	Fosinopril	Pfizer Inc	24	300.000000



# Thank You

