Medlife Pharmacy

README FILE

OVERVIEW

Medlife Pharmacy is a system that stores data related to medication use processes within pharmacies in an orderly and systematic manner. Thus, serving many purposes which ensures safety, accuracy, and efficiency within the pharmaceutical store. The main aim of this project is to develop the aforementioned system such that the pharmaceutical store can effectively manage its behemoth data and stay ahead of the curve.

This program deals with the day-to-day requirements about the pharmacy. It handles inventory details, drug details, customer details, employee details and the sales details. Thereupon, it will be facilitating the process for recording each of these data and fetch the necessary detail whenever required.

PRIMARY 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.
- To provide detailed information about the drugs in order to help locate it in the store easily.
- To streamline the use of inventory in the pharmacy

DATA SOURCES

- We have referenced the type of data by considering Pharmacy Management Systems like CVS Pharmacies which deal with collecting drugs from Suppliers and selling it to the customers.
- 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
- We have populated the tables by inserting data on our own, by keeping in mind the type of data that pharmacies like CVS use.

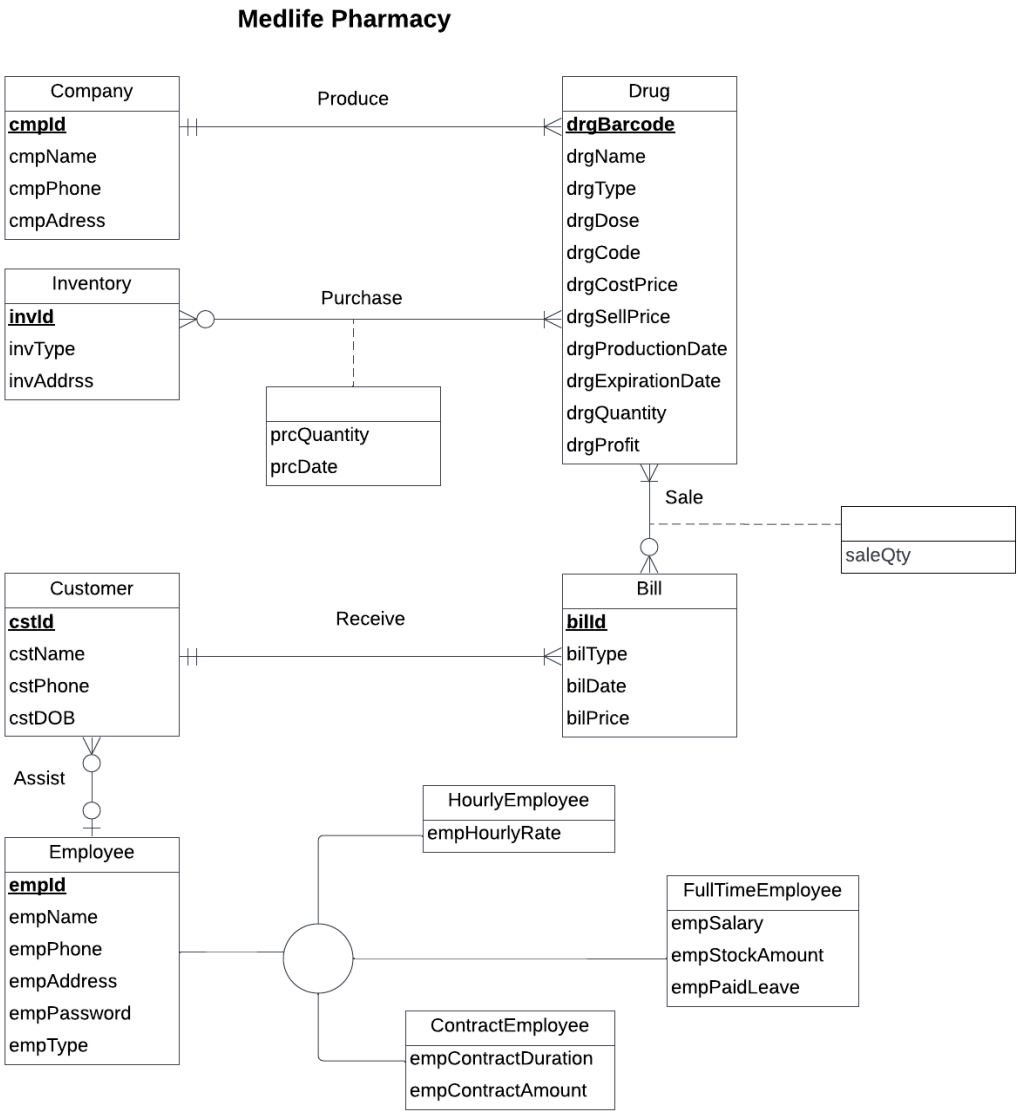
DATABASE DESIGN

TABLE DEFINITION

- **Company**: The company table contains the details of the companies that Medlife sources its drugs from. In our database, we store the company's name, address and phone number.
- **Drug**: The drug table contains the details of the drugs that Medlife purchases from the drug-producing companies. Details such as drug ID, drug Price, name of the company that produced the drug, etc is contained in the Drug table.
- **Inventory**: The inventory details contain the information about the type of inventory (e.g. Warehouse) and its address that is used by Medlife to store its drug shipments.
- **Employee**: The employee table contains details about the employees that work at Medlife and their details.
- **HourlyEmployee**: This is the subtype of the Employee table that contains details about the employees that are employed on an hourly basis. Information such as Hourly Rate of the employee is contained within this table.
- **FullTimeEmployee**: This is the subtype of the Employee table that contains details about the employees that are employed on a full-time basis. Information such as the employee salary is contained within this table.
- **ContractEmployee**: This is the subtype of the Employee table that contains details about the employees that are employed on a contract basis. Information such as the contract rate and the contract duration of the employee is contained within this table.
- **Customer**: The customer table consists of the data of the customers that visit Medlife and make a purchase.
- **Bill**: The Bill table consists of the data of the purchases made by a particular customer.
- **Purchase**: The purchase table consists of the data of the purchases made of a drug and which inventory it is stored in and also the data about the quantity and the date of purchase.
- **Sale**: The Sale table consists of the data of the drugs and their corresponding Bill number which can help us determine which customer is purchasing which drugs.

ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD) is a graphical representation that provides an overview of the scope of the project and how each data set is related to one another. Thus, helping to define the business processes and serving as the foundation for a relational database.



LOGICAL DATABASE DESIGN

Company (**cmpId**, cmpName, cmpPhone, cmpAddress)

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

Inventory(**invId**, invType, invAddress)

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

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

Employee (**empId**, empName, empPhone, empAddress, empPassword, empType)

HourlyEmployee (***empId***, empHourlyRate)

FullTimeEmployee (***empId***, empSalary, empStockAmount, empPaidLeave)

ContractEmployee (***empId***, empContractDuration, empContractAmount)

Purchase (***invId***, ***drgBarcode***, prcQuantity, prcDate)

Sale (***drgBarcode***, ***bilId***, saleQty)

TESTING THE PROJECT

CREATE TABLE

For our system, we require the following tables to store the pharmacy details:

- Company
- Drug
- Inventory
- Employee
- Customer
- Bill
- Purchase
- Sale
- Hourly Employee
- Full time employee
- Contract Employee

But, before we proceed with creating these tables, we need to make sure that they do not exist already and if they do, we need to drop them. For this, we will run the drop table query in the backwards order of the creation of each table

```
USE BUDT703_Project_0502_01

-- Drop the table if exists
DROP TABLE IF EXISTS ContractEmployee;
DROP TABLE IF EXISTS FullTimeEmployee;
DROP TABLE IF EXISTS HourlyEmployee;
DROP TABLE IF EXISTS Sale;
DROP TABLE IF EXISTS Purchase;
DROP TABLE IF EXISTS Bill;
DROP TABLE IF EXISTS Customer;
DROP TABLE IF EXISTS Employee;
DROP TABLE IF EXISTS Inventory;
DROP TABLE IF EXISTS Drug;
DROP TABLE IF EXISTS Company;
```

Now, we can move forward to create the tables

CREATE TABLE: COMPANY

```
CREATE TABLE Company (
        cmpName VARCHAR (30) NOT NULL,
        cmpPhone CHAR(10),
        cmpAddress VARCHAR(50),
        CONSTRAINT pk_Company_cmpName PRIMARY KEY (cmpName))
```

CREATE TABLE: DRUG

```
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)
```

CREATE TABLE: INVENTORY

```
CREATE TABLE Inventory (
    invId CHAR(10) NOT NULL,
    invType VARCHAR(20),
    invAddress VARCHAR (50),
    CONSTRAINT pk_Inventory_invId PRIMARY KEY (invId))
```

CREATE TABLE: EMPLOYEE

```
    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) )
```

CREATE TABLE: CUSTOMER

```
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)
```

CREATE TABLE: BILL

```
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 )
```

CREATE TABLE: PURCHASE

```
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)
```

CREATE TABLE: SALE

```
CREATE TABLE Sale (
        drgBarcode CHAR (10) NOT NULL,
        bilId CHAR (10) NOT NULL,
        saleQty INTEGER,
        CONSTRAINT pk_Sale_drgBarcode_bilId PRIMARY KEY (drgBarcode, bilId),
        CONSTRAINT fk_Sale_drgBarcode FOREIGN KEY (drgBarcode)
                REFERENCES Drug (drgBarcode)
                ON DELETE NO ACTION ON UPDATE CASCADE,
         CONSTRAINT fk_Sale_bilId FOREIGN KEY (bilId)
                REFERENCES Bill (bilId)
                ON DELETE CASCADE ON UPDATE CASCADE)
```

CREATE TABLE: HOURLY EMPLOYEE

```
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)
```

CREATE TABLE: FULL TIME EMPLOYEE

```
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)
```

CREATE TABLE: CONTRACT EMPLOYEE

```
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)
```

INSERT RECORDS OUTPUT

Some sample queries to insert into the tables is given below:

INSERT DATA IN THE TABLE: COMPANY

```
INSERT INTO Company (cmpName, cmpPhone, cmpAddress)
VALUES ('Pfizer Inc', '2127332323', '235 East 42nd Street New York NY 10017 USA'),
        ('AbbVie Inc' , '8002555162', '1 North Waukegan Road North Chicago, IL 60064
USA'),
        ('Johnson & Johnson' , '3017388172', 'New Brunswick, NJ 08933 USA');

SELECT * FROM COMPANY
```

OUTPUT:

	cmpName	cmpPhone	cmpAddress
1	AbbVie Inc	8002555162	1 North Waukegan Road North Chicago, IL 60064 USA
2	Astrazeneca	9009088172	33467, Baltimore Ave, MD 20740 USA
3	Johnson & Johnson	3017388172	New Brunswick, NJ 08933 USA
4	Merck	9007388172	12345, New Brunswick, NJ 08933 USA
5	Novartis	9007388172	1, South Wan Road, North Chicago, IL 60063, USA
6	Pfizer Inc	2127332323	235 East 42nd Street New York NY 10017 USA
7	Roche	2129032323	236, East 41st Street, New York, NY 10017, USA
8	Sanofi	9006788172	1345, New hampshire, VA 98983 USA

INSERT DATA IN THE TABLE: DRUG

```
INSERT INTO Drug (drgBarcode ,drgName, drgType, drgDose, drgCode, drgCostPrice,
                  drgSellPrice, drgProductionDate, drgExpirationDate, drgQuantity,
        drgProfit, cmpName)
VALUES ('1112908900', 'Atorvastatin', 'Prescription' , 2, '2d00', 23.4, 29.9,
        '2022-08-11', '2025-08-11', 12, 6.50, 'Pfizer Inc'),
        ('1112908901', 'Levothyroxine', 'Prescription' , 2, '3m50', 13.4, 20.5, '2020-07-
        12', '2022-07-12', 16, 7.10, 'AbbVie Inc'),
        ('1112908902', 'Metformin', 'Prescription' , 1, '4f30', 43.4, 55.9, '2021-11-13',
        '2025-11-13', 20, 12.50, 'Johnson & Johnson')
        ('1112908903', 'Fosinopril', 'Prescription' , 2, '2d34', 73.40, 85.90, '2022-08-
        11', '2025-08-11', 15, 'AbbVie Inc',  12.50),
        ('1112908904', 'Captopril', 'Prescription' , 2, '3m50', 13.4, 20.5, '2020-07-
        12', '2022-07-12', 16, 'AbbVie Inc', 10.10);

SELECT * FROM DRUG
```

OUTPUT:

	drgBarcode	drgName	drgType	drgDose	drgCode	drgCostPrice	drgSellPrice	drgProductionDate	drgExpirationDate	drgQuantity	cmpName	drgProfit
1	1112908900	Atorvastatin	Inhibitor	2	2d00	23.40	29.90	2022-08-11	2025-08-11	12	Pfizer Inc	6.50
2	1112908901	Levothyroxine	Hormone	2	3m50	13.40	20.50	2020-07-12	2022-07-12	16	AbbVie Inc	7.10
3	1112908902	Metformin	Biguanides	1	4f30	43.40	55.90	2021-11-13	2025-11-13	20	Johnson & Johnson	12.50
4	1112908903	Fosinopril	Inhibitor	2	2d34	73.40	85.90	2022-08-11	2025-08-11	15	Pfizer Inc	12.50
5	1112908904	Captopril	Inhibitor	2	3m55	15.40	25.50	2020-07-12	2022-07-12	20	AbbVie Inc	10.10
6	1112908905	Alecensa	Prescription	3	2d10	20.40	21.90	2021-07-11	2025-08-11	12	Roche	1.50
7	1112908906	Esbriet	Prescription	2	3h50	15.40	24.50	2022-07-12	2025-07-12	16	Roche	9.10
8	1112908907	Adakveo	Prescription	1	4f90	23.40	32.90	2021-11-13	2025-11-13	20	Novartis	9.50
9	1112908908	Arzerra	Prescription	1	4f00	143.40	155.90	2021-11-13	2025-11-13	10	Novartis	12.50
10	1112908909	Clarinex	Prescription	1	6s30	83.40	95.90	2022-11-13	2025-12-13	15	Merck	12.50
11	1112908910	Fosamax	Prescription	1	9j90	3.40	5.90	2020-07-13	2025-11-13	17	Merck	2.50
12	1112908911	Adacel	Prescription	1	sf30	23.40	35.90	2021-11-13	2023-11-13	25	Sanofi	12.50
13	1112908912	Admelog	Prescription	1	gg30	53.40	65.90	2022-09-13	2025-10-13	25	Sanofi	12.50
14	1112908913	Lokelma	Prescription	1	df20	47.40	57.90	2022-07-13	2027-11-13	22	Astrazeneca	10.50
15	1112908914	Nexium	Prescription	1	sf30	49.40	59.90	2022-05-13	2024-11-13	26	Astrazeneca	10.50

INSERT DATA IN THE TABLE: INVENTORY

```
INSERT INTO Inventory (invId, invType, invAddress)
VALUES ('A345678900', 'Warehouse', '3240 Fawn Circle, PA 204567'),
        ('A345678901', 'Warehouse', '6210 Belcrest Road, Hyattsville 20782'),
```

('A345678902', 'Warehouse', '4330 Heart Break road, College park 20742');

SELECT * FROM Inventory

OUTPUT:

invId	invType	invAddress
A345678900	Warehouse	3240 Fawn Circle, PA 204567
A345678901	Warehouse	6210 Belcrest Road, Hyattsville 20782
A345678902	Warehouse	4330 Heart Break road, College park 20742

INSERT DATA IN THE TABLE: EMPLOYEE

```
INSERT INTO Employee (empId, empName, empPhone, empAddress,
                      empPassword, empType)
VALUES ('1111192779','John Walker', '457890894', '571 Gulf Ave., Onalaska, WI
        54650', 'gjkTYU&56^', 'Hourly'),
        ('1111192780','Raja Syan', '2405678894', '972 N. Myrtle St., Southampton, PA
        18966', 'GHJKU&46^', 'Hourly'),
        ('1111192781','Yash Taylor', '2046378894', '829 Edgewater Street, Nashua, NH
        03060', 'LKJ78&56^', 'Hourly'),
       ('1111192782','Zeel Patel', '1206378894', '118 Spring Dr., Poughkeepsie, NY
       12601', 'UIOP34&5z^', 'FullTime'),
        ('1111192783','Vikram Walker', '2316378894', '78 Olive St., Vernon Hills, IL
        60061', 'LKJrtU&56^', 'FullTime'),
        ('1111192784','Harsh Zalavadia', '4845678894', '9743 Big Rock Cove Court,
Coventry,
        RI 02816', '9087Y&56^', 'Contract'),
        ('1111192785','Nishant Vun', '2406378894', '500 Fulton Rd., Delray Beach, FL
        33445', 'opLAU&56^', 'Contract'),
        ('1111192786','Geet Maine', '34046378894', '8082 Pilgrim Circle, Glendale, AZ
        85302', 'MNghU&56^', 'Contract'),
        ('1111192787','Riddhi Gawda', '8906378894', '9420 Harvard Dr., Maineville, OH
        45039', 'op><YU&56^', 'Contract');
```

SELECT * FROM Employee

OUTPUT:

	empId	empName	empPhone	empAddress	empPassword	empType
1	1111192779	John Walker	457890894	571 Gulf Ave., Onalaska, WI 54650	gjkTYU&56^	Hourly
2	1111192780	Raja Syan	2405678894	972 N. Myrtle St., Southampton, PA 18966	GHJKU&46^	Hourly
3	1111192781	Yash Taylor	2046378894	829 Edgewater Street, Nashua, NH 03060	LKJ78&56^	Hourly
4	1111192782	Zeel Patel	1206378894	118 Spring Dr., Poughkeepsie, NY 12601	UIOP34&5z^	FullTime
5	1111192783	Vikram Walker	2316378894	78 Olive St., Vernon Hills, IL 60061	LKJrtU&56^	FullTime
6	1111192784	Harsh Zalavadia	4845678894	9743 Big Rock Cove Court, Coventry, RI 02816	9087Y&56^	FullTime
7	1111192785	Nishant Vun	2406378894	500 Fulton Rd., Delray Beach, FL 33445	opLAU&56^	Contract
8	1111192786	Geet Maine	34046378894	8082 Pilgrim Circle, Glendale, AZ 85302	MNghU&56^	Contract
9	1111192787	Riddhi Gawda	8906378894	9420 Harvard Dr., Maineville, OH 45039	op><YU&56^	Contract
10	1111192788	Sam Walker	789090894	47 Pawnee Street Maspeth, NY 11378	eefYU&56^	Hourly
11	1111192789	Virat Syan	89455678894	51 Airport Street Morrisville, PA 19067	jklKU&46^	Hourly
12	1111192790	Zeel Taylor	9089778894	9836 Maiden Ave. Shrewsbury, MA 01545	qwe78&56^	Hourly
13	1111192791	Jack Patel	9088978894	9962 Kent Rd. Lawrenceville, GA 30043	hjkl34&5z^	Hourly
14	1111192792	Jhony Walker	6616378894	7412 Somerset Avenue Holly Springs, NC 27540	vbnrtU&56^	Hourly
15	1111192793	Harsh Jaskal	0945678894	9853 Golf Lane Nutley, NJ 07110	1089Y&56^	Hourly
16	1111192794	Nishant Bun	1206378894	7781 Del Monte St. Middle Village, NY 11379	cvbAU&56^	Hourly
17	1111192795	Sherine Maine	34046309780	9451 Henry Dr. Indiana, PA 15701	yuihU&56^	Hourly
18	1111192796	Katha Gawda	8906356789	814 Clinton Lane Downingtown, PA 19335	yt.*<YU&56^	Hourly
19	1111192797	Virat Kohli	124590894	90 Pawnee Street Maspeth, NY 11378	iopYU&56^	FullTime
20	1111192798	Zyan Malik	89455671234	71 Airport Street Morrisville, PA 19067	sdfKU&46^	FullTime
21	1111192799	Zeel Zalavadia	9089776523	7636 Maiden Ave. Shrewsbury, MA 01545	qwe78&56^	FullTime

INSERT DATA IN THE TABLE: CUSTOMER

```
INSERT INTO Customer (cstId, cstName, cstPhone, cstDOB, empId)
VALUES ('A33BH90121', 'Jerome Heff', '2239098767', '2002-09-09', '1111192779'),
        ('A33BH90122', 'Jafer Heff', '3456698767', '2003-09-09', '1111192781'),
        ('A33BH90123', 'Adam Heff', '9089098767', '2004-09-09', '1111192782');
SELECT * FROM Customer;
```

OUTPUT:

	cstId	cstName	cstPhone	cstDOB	empId
1	A33BH90121	Jerome Heff	2239098767	2002-09-09	1111192779
2	A33BH90122	Jafer Heff	3456698767	2003-09-09	1111192781
3	A33BH90123	Adam Heff	9089098767	2004-09-09	1111192782
4	A33BH90124	Jerome Heff	2239098767	2002-09-09	1111192806
5	A33BH90125	Jafer Heff	3456698767	2003-09-09	1111192807
6	A33BH90126	Adam Heff	9089098767	2004-09-09	1111192808
7	A33BH90127	Jerome Heff	2239098767	2002-09-09	1111192809
8	A33BH90128	Jafer Heff	3456698767	2003-09-09	1111192810
9	A33BH90129	Virat Heff	9089098767	2004-09-09	1111192812
10	A33BH90130	Jerome Patel	2239098767	2002-09-09	1111192813
11	A33BH90131	Harsh Heff	3456698767	2003-09-09	1111192814
12	A33BH90132	Jerome Heff	2239098767	2002-09-09	1111192797
13	A33BH90133	Jafer Heff	3456698767	2003-09-09	1111192798
14	A33BH90134	Virat Heff	9089098767	2004-09-09	1111192799
15	A33BH90135	Jerome Patel	2239098767	2002-09-09	1111192800
16	A33BH90136	Harsh Heff	3456698767	2003-09-09	1111192801

INSERT DATA IN THE TABLE: BILL

```
INSERT INTO Bill (bilId , bilType, bilDate, bilPrice, cstId)
VALUES('B234568201', 'Over The Counter', '2022-08-11', 1024.59, 'A33BH90121'),
        ('B234568202', 'Self Checkout', '2021-05-21', 126.19, 'A33BH90122'),
        ('B234568203', 'Over The Counter', '2022-03-30', 354.84, 'A33BH90123');
SELECT * FROM Bill;
```

OUTPUT:

	billd	bilType	bilDate	bilPrice	cstId
1	B234568201	Over The Counter	2022-08-11	1024.59	A33BH90121
2	B234568202	Self Checkout	2021-05-21	126.19	A33BH90122
3	B234568203	Over The Counter	2022-03-30	354.84	A33BH90123
4	B234568204	Over The Counter	2022-08-11	1024.59	A33BH90124
5	B234568205	Self Checkout	2021-05-21	126.19	A33BH90125
6	B234568206	Over The Counter	2022-03-30	354.84	A33BH90126
7	B234568207	Over The Counter	2022-08-11	1024.59	A33BH90127
8	B234568208	Self Checkout	2021-05-21	126.19	A33BH90128
9	B234568209	Over The Counter	2022-03-30	354.84	A33BH90129
10	B234568210	Over The Counter	2022-08-11	1024.59	A33BH90130
11	B234568211	Self Checkout	2021-05-21	126.19	A33BH90131
12	B234568212	Over The Counter	2022-08-11	1024.59	A33BH90132
13	B234568213	Self Checkout	2021-05-21	126.19	A33BH90133
14	B234568214	Over The Counter	2022-03-30	354.84	A33BH90134
15	B234568215	Over The Counter	2022-08-11	1024.59	A33BH90135
16	B234568216	Self Checkout	2021-05-21	126.19	A33BH90136

INSERT DATA IN THE TABLE: PURCHASE

```
INSERT INTO Purchase (invId, drgBarcode , prcQuantity, prcDate)
VALUES ('A345678900', '1112908900', 10, '2022-08-11'),
        ('A345678901', '1112908901', 20, '2021-09-10'),
        ('A345678902', '1112908902', 15, '2022-07-12');
SELECT * FROM Purchase;
```

OUTPUT:

invId	drgBarcode	prcQuantity	prcDate
A345678900	1112908900	10	2022-08-11
A345678900	1112908901	10	2021-09-11
A345678901	1112908901	20	2021-09-10
A345678901	1112908902	30	2021-09-10
A345678902	1112908900	15	2021-09-12
A345678902	1112908902	15	2022-07-12

INSERT DATA IN THE TABLE: SALE

```
INSERT INTO Sale (drgBarcode, bilId)
VALUES ('1112908900', 'B234568201'),
        ('1112908902', 'B234568202'),
        ('1112908901', 'B234568202');
SELECT * FROM Sale;
```

OUTPUT:

drgBarcode	billd	saleQty
1112908900	B234568201	30
1112908900	B234568202	20
1112908901	B234568201	30
1112908901	B234568202	20
1112908901	B234568203	23
1112908902	B234568201	30
1112908902	B234568202	20
1112908902	B234568203	57
1112908903	B234568201	24
1112908904	B234568201	19
1112908904	B234568202	12

INSERT DATA IN THE TABLE: HOURLY EMPLOYEE

```
INSERT INTO HourlyEmployee (empId, empHourlyRate)
VALUES ('1111192779', 15.80),
        ('1111192780', 17.90),
        ('1111192781', 23.60);
SELECT * FROM Employee;
```

OUTPUT:

	empId	empHourlyRate
1	1111192779	15.80
2	1111192780	17.90
3	1111192781	23.60
4	1111192788	16.80
5	1111192789	12.90
6	1111192790	24.60
7	1111192791	25.60
8	1111192792	13.60
9	1111192793	26.60
10	1111192794	21.60
11	1111192795	13.60
12	1111192796	10.60

INSERT DATA IN THE TABLE: FULL TIME EMPLOYEE

```
INSERT INTO FullTimeEmployee (empId, empSalary, empStockAmount, empPaidLeave)
VALUES ('1111192782', 23456.90, 34532.00, 25),
        ('1111192783', 63456.20, 4532.00, 20),
        ('1111192784', 53436.90, 14532.00, 35);
SELECT * FROM FullTimeEmployee;
```

OUTPUT:

	empId	empSalary	empStockAmount	empPaidLeave
1	1111192782	23456.90	34532.00	25
2	1111192783	63456.20	4532.00	20
3	1111192784	53436.90	14532.00	35
4	1111192797	33456.90	12532.00	23
5	1111192798	83456.20	5532.00	22
6	1111192799	12436.90	17532.00	20
7	1111192800	76456.90	56532.00	25
8	1111192801	23456.20	5432.00	20
9	1111192802	90436.90	39532.00	35
10	1111192803	12456.90	53532.00	25
11	1111192804	87456.20	9032.00	20
12	1111192805	45436.90	63532.00	35

INSERT DATA IN THE TABLE: CONTRACT EMPLOYEE

```
INSERT INTO ContractEmployee (empId, empContractDurationMonths,
empContractAmount)
VALUES ('1111192785', 15, 89098.00),
        ('1111192786', 25, 180040.00),
        ('1111192787', 5, 19398.00);
SELECT * FROM ContractEmployee;
```

OUTPUT:

	empId	empContractDurationMonths	empContractAmount
1	1111192785	15	89098.00
2	1111192786	25	180040.00
3	1111192787	5	19398.00
4	1111192806	12	90098.00
5	1111192807	15	190040.00
6	1111192808	5	21398.00
7	1111192809	25	91098.00
8	1111192810	15	200040.00
9	1111192811	10	22398.00
10	1111192812	24	89098.00
11	1111192813	25	230040.00
12	1111192814	5	22398.00

BUSINESS TRANSACTION 1 - QUERY & OUTPUT

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.emp_type 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.emp_type
ORDER BY COUNT(e.empId) DESC;
```

Employee Type	Number of Customers
Contract	8
FullTime	6
Hourly	2

BUSINESS TRANSACTION 2 - QUERY & OUTPUT

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 TRANSACTION 3 - QUERY & OUTPUT

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.bilId = b.bilId))
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 TRANSACTION 4 - QUERY & OUTPUT

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 the 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	1337.5000
Hormone	Levothyroxine	AbbVie Inc	73	518.30000
Inhibitor	Atorvastatin	Pfizer Inc	50	325.00000
Inhibitor	Captopril	AbbVie Inc	31	313.10000
Inhibitor	Fosinopril	Pfizer Inc	24	300.00000

REFERENCES

[1]

https://www.g2.com/categories/pharmacy-management-systems

[2]
https://en.wikipedia.org/wiki/Pharmacy_management_system#:~:text=The%20pharmacy%20management%20system%2C%20also,medication%20use%20process%20within%20pharmacies.

[3]

https://arirms.com/features-of-pharmacy-management-system