

DATABASE DESIGN FOR PHARMACY

CS 6360.002

FINAL PROJECT

TEAM MEMBERS:

AKHIL BITRA (AXB170061)

YASH JAIN(YXJ180004)

Table of Contents

Requirements	3
Services offered by Pharmacy	3
Functional Requirements	3
Employee Structure	4
Minimum Wage Act	4
Technology used in Pharmacy Operations.....	4
Operations performed by Pharmacies.....	5
Modeling of Requirements as ER-Diagram	6
Mapping of ERD in Relational Schema	8
Normalization of Relational Schema	10
SQL Statements to create Relations in DB and Add Constraints	11
SQL Statements to Insert data into Relations	14
PL/SQL – Triggers.....	16
Trigger I	16
Trigger II	17
PL/SQL- Procedures.....	18
Procedure I.....	18
Procedure II	19

Requirements

Services offered by Pharmacy

Numerous services are being offered by pharmacies and some of them are listed below.

- **Patient Consultations:**
One-on-one consulting between patients and pharmacists to learn about medication use, possible side effects, and medication adherence.
- **Home Delivery:**
Medication delivery is provided by pharmacies for those who have trouble reaching the pharmacy and purchasing medicines.
- **Immunizations:**
Pharmacies provide vaccinations for influenza, shingles, and pneumonia.
- **Clinic checks:**
Pharmacies hold specialized clinic days to help learn about specific health conditions such as heart disease or diabetes.
- **Lab testing:**
This is done by pharmacies to improve medication compliance and effectiveness and enhance patient safety.
- **Compliance aids:**
These aids help you to remember when you have taken your medication.
- **Compounding prescriptions:**
If your medications are not commercially available in the dose or form, they were prescribed, pharmacies will mix these medications for you.
- **Smoking cessation support:**
In addition to recommending over the counter medications to help you quit smoking, pharmacies can prescribe certain medications for smoking cessation.
- **Counselling services:**
Pharmacies can provide a range of advice and education about prescriptions, over-the-counter medications, natural health products, medical devices and more.

Functional Requirements:

There are certain functions done by the pharmacy such as - store the necessary information of drugs, prepare bill for the medicine, give week reports, easily searching of medicine, working in two languages, Update, delete and save medicines data.

- **Generate report:** The Pharmacy management system generates weekly report on the information about the drugs and it exports the information as output document.

- **Store the necessary information of the drugs:** The Pharmacy system stores the detailed information about each medicine including actual name, formula of medicine and how it is importance and for which disease it is required.
- **Searching Medicine and other Data:** The Pharmacy system can easily search for medicines which shows in which shelf the medicines are put and the behavior of the medicines. The searching process is based on the name of the given data or the identification of the item. Here when the user searches the item on search bar the related things were displayed on the screen and can select the actual item that the user needs.
- **Alerting pharmacy Data's in the system:** Changing medicines to another because of outdated medicine, modifying the saved medicine data due to some incorrect data, deleting data of the pharmacy.

Employee Structure:

There are several people with different designations such as Clinical Pharmacist, Checking Technician, Dispensary Manager, Medicines Supply Service Manager, Pharmacy Stores Manager etc.

Minimum Wage Act:

Under the final rule issued by the Department of Labor (DoL), the initial increase to the Fair Labor Standards Act minimum salary level to qualify for an exemption is from \$455 to \$913 per week (i.e., from \$23,660 to \$47,476 annually). Future automatic updates to the thresholds will occur every 3 years, starting on January 1, 2020.

Technology used in Pharmacy operations:

Various types of technology have found application in pharmacy operations. They can be grouped into mechanization technologies, quality assessment technology, information and communications technologies (ICTs), automation technologies and the newly evolving biotechnology.

Mechanisation Technologies:

Use of electrical mixing vessel in compounding medicines is an example of mechanization technology. The technology for compounding medicines include facilities for making individualized doses of medicines such as intravenous feeding solution for patients unable to eat, or doses of anticancer chemotherapy.

Quality Assessment Technology:

Quality assessment technology is used for the task of assessing product quality. The technology used to assess the quality of drug products spans a wide spectrum of technological artifacts, some of which are

chemical reagents, chemical and biochemical equipment and instruments as well as high precision instruments for pharmaceutical analysis.

Information and communications technologies:

The overall ICT infrastructure comprises the computer and communication technologies and the shareable technical platforms and databases. Most of the technologies adopted in pharmacy operations are ICT-based technologies. The key ICT-based technologies to be discussed include the computer, other computer-based technologies and the telephone.

Operations performed by Pharmacies:

Enrollment/Patient Intake:

Enrollment information can be received at the pharmacy in various forms from multiple sources. As a result, the technology that supports the enrollment process will need to ensure receipt capabilities from physician/hospital faxed forms, payer electronic feeds, and third-party providers via telephone calls and e-mails.

Insurance Verification:

Pharmacies will typically attempt an adjudication transaction based on the information received during the enrollment work flow operation. Performing automated insurance verification at this point is critical to work flow efficiency, as information obtained from and confirmed by third party services can help to ensure claims will process successfully.

Order/Item Entry:

Pharmacies may automatically create orders and order supplemental items (ie, supplies) utilizing information from enrollment and insurance verification work flows as well as information based on a pharmacy's standard operating procedure for certain drugs. Successful technologies for order entry work flow should support the association of prescription images received from a provider to manual entries.

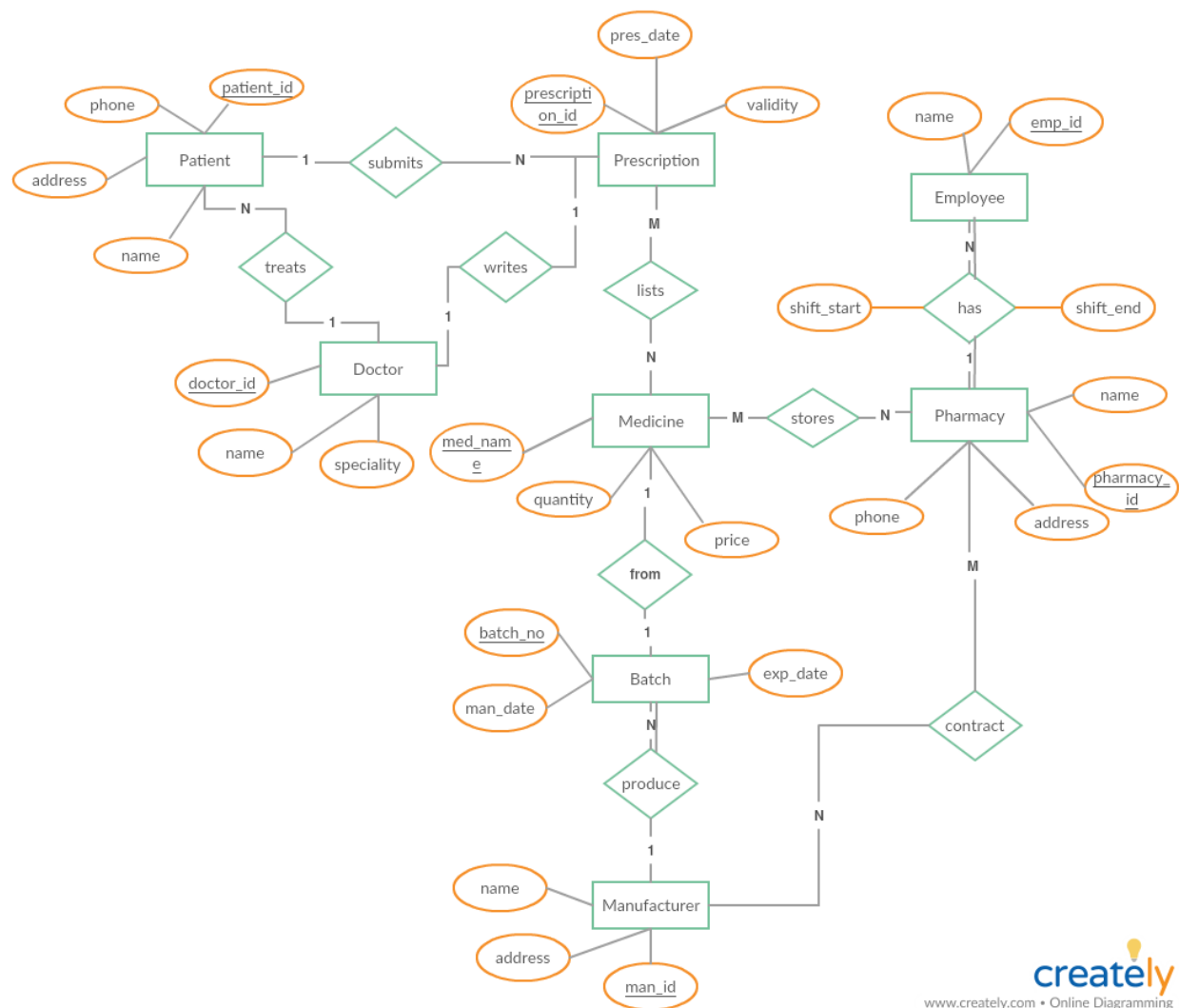
Prior Authorization:

Certain payers may require a Prior Authorization for certain drugs. Successful technologies should be configured to associate Prior Authorizations at payer/drug level in order to begin processing during the enrollment work flow operation.

Scheduling:

Coordinating the delivery of drugs with the patient and provider is a critical work flow. Without ensuring proper delivery, patients may not receive the necessary medication in a timely manner. As a result, successful technologies should manage overnight packaging requirements and shipping restriction days in accordance with certain drug parameters.

Modeling of Requirements as ER-Diagram:



The requirements can be summarized from the ERD as –

1. A patient can submit one or more prescriptions represented as N in the diagram.
2. A doctor treats one or more patients.
3. A doctor writes one and only one prescription.
4. One or more prescriptions lists one or more medicines which becomes an M:N relation where M prescriptions can contain N relations.
5. One Medicine is only from one Batch represented as 1:1
6. A single manufacturer can produce one more batch and hence it becomes 1:N relation.
7. Pharmacies store many medicines and thus becomes M:N

8. A pharmacy contains 1 or more employees which becomes a 1:N relation and the employees should work at a pharmacy and vice versa which implies it becomes a total participation at both sides.
9. Multiple pharmacies can get contract from more than 1 manufacturer and thus becomes a M:N relation.
10. Batches should be produced by manufacturer and so it holds a total participation.

Mapping of ERD in Relational Schema

1. PATIENT

<u>Patient_id</u>	Phone	Name	Address	Doc_id
-------------------	-------	------	---------	--------

- PRIMARY KEY : PATIENT_ID
- FOREIGN KEY : FOREIGN KEY(DOC_ID) REFERENCES DOCTOR(DOCTOR_ID)

2. PRESCRIPTION

<u>Prescription_id</u>	Validity	Docid	Pres_date	Patient_id
------------------------	----------	-------	-----------	------------

- PRIMARY KEY : PRESCRIPTION_ID
- FOREIGN KEY : FOREIGN KEY(PATIENT_ID) REFERENCES PATIENT(PATIENT_ID)

3. DOCTOR

<u>Doctor_id</u>	Specialty	Name
------------------	-----------	------

- PRIMARY KEY : DOCTOR_ID
- FOREIGN KEY : None

4. MEDICINE

<u>Med_name</u>	Quantity	Price	Batch_no
-----------------	----------	-------	----------

- PRIMARY KEY : MED_NAME
- FOREIGN KEY : FOREIGN KEY(BATCH_NO) REFERENCES BATCH(BATCH_NO)

5. PHARMACY

<u>Pharmacy_id</u>	Name	Phone	Address
--------------------	------	-------	---------

- PRIMARY KEY : PHARMACY_ID
- FOREIGN KEY :

6. EMPLOYEE

<u>Emp_id</u>	Name	Shift_start	Shift_end	Pharmacy_id
---------------	------	-------------	-----------	-------------

- PRIMARY KEY: EMP_ID
- FOREIGN KEY: FOREIGN KEY(PHARMACY_ID) REFERENCES PHARMACY(PHARMACY_ID)

7. BATCH

<u>Batch_no</u>	Expiry	Man_date	Man_id
-----------------	--------	----------	--------

- PRIMARY KEY: BATCH_NO
- FOREIGN KEY: FOREIGN KEY(MAN_ID) REFERENCES MANUFACTURER(MAN_ID)

8. MANUFACTURER

<u>Man_id</u>	Name	Address
---------------	------	---------

- PRIMARY KEY: MAN_ID
- FOREIGN KEY: NONE

9. PRES_MEDICINE

<u>Prescription_id</u>	<u>Med_name</u>
------------------------	-----------------

- PRIMARY KEY: PRESCRIPTION_ID, MED_NAME
- FOREIGN KEY: FOREIGN KEY(PRESCRIPTION_ID) REFERENCES PRESCRIPTION (PRESCRIPTION_ID), FOREIGN KEY(MED_NAME) REFERENCES MEDICINE (MED_NAME)

10. MED_PHAR

<u>Med_name</u>	<u>Pharmacy_id</u>
-----------------	--------------------

- PRIMARY KEY: MED_NAME, PHARMACY_ID
- FOREIGN KEY: FOREIGN KEY(MED_NAME) REFERENCES MEDICINE (MED_NAME), FOREIGN KEY(PHARMACY_ID) REFERENCES PHARMACY (PHARMACY_ID)

11. MAN_PHAR

<u>Man_id</u>	<u>Phar_id</u>
---------------	----------------

- PRIMARY KEY: MAN_ID, PHAR_ID
- FOREIGN KEY: FOREIGN KEY(MAN_ID) REFERENCES MANUFACTURER(MAN_ID), FOREIGN KEY(PHAR_ID) REFERENCES PHARMACY(PHARMACY_ID)

Normalization of Relational Schema

The following functional dependencies exist on the above relational tables

1. PATIENT {Patient_id -> phone, name, address, doc_id}
2. PRESCRIPTION {Prescription_id -> validity, doc_id, pres_date, patient_id}
3. DOCTOR {Doctor_id -> speciality, name}
4. MEDICINE {Med_name -> quantity, price, batch_no}
5. PHARMACY {Pharmacy_id -> name, phone, address}
6. EMPLOYEE {Emp_id -> name, shift_start, shift_end, pharmacy_id}
7. BATCH {Batch_no -> expiry, man_date, man_id}
8. MANUFACTURER {Man_id -> name, address}

The functional dependencies make the relational schema to be in third normal form.

SQL Statements to create Relations in DB and Add Constraints

```
Create table Patient(  
Patient_id int,  
Phone int,  
Name varchar(10),  
Address varchar(50),  
Doctor_id int,  
primary key(Patient_id));
```

```
Create table Prescription(  
Prescription_id int,  
Pres_date date,  
Validity date,  
Doctor_id int,  
Patient_id int,  
primary key(Prescription_id),  
CONSTRAINT chk_validity CHECK (Validity >= Pres_date));
```

```
Create table Pres_Med(  
Prescription_id int,  
Med_Name varchar(10),  
primary key(Prescription_id, Med_Name));
```

```
Create table Medicine(  
Med_Name varchar(10),  
Quantity int,  
Price int,  
Batchno int,  
primary key(Med_Name));
```

```
Create table Doctor(  
Doctor_id int,  
Name varchar(10),  
Speciality varchar(15),  
primary key(Doctor_id));
```

```
Create table Batch(  
Batchno int,
```

```
Man_date date,  
Exp_date date,  
Man_id int,  
primary key(Batchno),  
constraint chk_exp check (Exp_date > Man_date));
```

```
Create table Manufacturer(  
Man_id int,  
Name varchar(10),  
Address varchar(50),  
primary key(Man_id));
```

```
Create table Pharmacy(  
Pharmacy_id int,  
Name varchar(10),  
Phone int,  
Address varchar(50),  
primary key(Pharmacy_id));
```

```
Create Table Employee(  
Emp_id int,  
Name varchar(10),  
Shift_start date,  
Shift_end date,  
Pharmacy_id int,  
primary key(Emp_id),  
constraint chk_endtime check (Shift_end > Shift_start));
```

```
Create table Pharm_Med(  
Pharmacy_id int,  
Med_Name varchar(10),  
primary key(Pharmacy_id, Med_Name));
```

```
create table man_phar(  
man_id int,  
pharmacy_id int,  
primary key(man_id,pharmacy_id));
```

```
Alter table Employee add foreign key (Pharmacy_id) references Pharmacy(Pharmacy_id) on delete  
cascade;
```

```
Alter table Manufacturer add foreign key (Pharmacy_id) references Pharmacy(Pharmacy_id) on delete  
cascade;
```

```
Alter table Batch add foreign key (Man_id) references Manufacturer(Man_id) on delete cascade;
```

```
Alter table Medicine add foreign key (Batchno) references Batch(Batchno) on delete cascade;
```

```
Alter table Patient add foreign key (Doctor_id) references Doctor(Doctor_id) on delete cascade;
```

```
Alter table Prescription add foreign key (Doctor_id) references Doctor(Doctor_id) on delete cascade;
```

```
Alter table Prescription add foreign key (Patient_id) references Patient(Patient_id) on delete cascade;
```

Alter table Pres_Med add foreign key (Prescription_id) references Prescription(Prescription_id) on delete cascade;
Alter table Pres_Med add foreign key (Med_Name) references Medicine(Med_Name) on delete cascade;
Alter table Pharm_Med add foreign key (Pharmacy_id) references Pharmacy(Pharmacy_id) on delete cascade;
Alter table Pharm_Med add foreign key (Med_Name) references Medicine(Med_Name) on delete cascade;
Alter table man_phar add foreign key(man_id) references manufacturer(man_id) on delete cascade;
Alter table man_phar add foreign key(pharmacy_id) references pharmacy(pharmacy_id) on delete cascade;

SQL Statements to insert data into relations

```
alter session set nls_date_format = 'DD/MM/YYYY HH24:MI:SS';
```

```
insert into doctor values(1,'House','Physician');
insert into doctor values(2,'Wilson','Pediatrician');
insert into doctor values(3,'Cuddy','Orthopedist');
```

```
insert into patient values(1,44455577,'John','Frankford',2);
insert into patient values(2, 44455566,'Cynthia','Coppell',3);
insert into patient values(3, 66655577, 'Nick','Irving',1);
insert into patient values(4, 44466677, 'Ramesh','Campbell',3);
insert into patient values(5, 55577766, 'Kyle','Plano',1);
```

```
insert into pharmacy values(1,'CVS',222111333,'Plano');
insert into pharmacy values(2,'Walgreens',222444333,'Irving');
insert into pharmacy values(3,'Apollo',222666333,'Coppell');
insert into pharmacy values(4,'Redcross',222111666,'Plano');
insert into pharmacy values(5,'Bluehealth',222111555,'Frankford');
```

```
insert into employee
values(1,'Steve',TO_DATE('09:00:00','HH24:MI:SS'),TO_DATE('17:00:00','HH24:MI:SS'),2);
insert into employee
values(2,'Bill',TO_DATE('08:00:00','HH24:MI:SS'),TO_DATE('15:00:00','HH24:MI:SS'),3);
insert into employee
values(3,'Bezos',TO_DATE('12:00:00','HH24:MI:SS'),TO_DATE('22:00:00','HH24:MI:SS'),1);
insert into employee
values(4,'Elon',TO_DATE('10:00:00','HH24:MI:SS'),TO_DATE('20:00:00','HH24:MI:SS'),5);
insert into employee
values(5,'Mark',TO_DATE('14:00:00','HH24:MI:SS'),TO_DATE('23:59:59','HH24:MI:SS'),4);
```

```
insert into manufacturer values(1,'Abbott','Coppell');
insert into manufacturer values(2,'Alkem','Plano');
insert into manufacturer values(3,'Pfizer','Frankford');
insert into manufacturer values(4,'Novartis','Irving');
insert into manufacturer values(5,'GSK','Preston');
```

```
insert into batch
values(1,TO_DATE('05/01/2018','DD/MM/YYYY'),TO_DATE('04/01/2020','DD/MM/YYYY'),4);
insert into batch
values(2,TO_DATE('05/06/2018','DD/MM/YYYY'),TO_DATE('04/11/2019','DD/MM/YYYY'),3);
insert into batch
values(3,TO_DATE('16/02/2017','DD/MM/YYYY'),TO_DATE('04/12/2018','DD/MM/YYYY'),1);
insert into batch
values(4,TO_DATE('12/10/2018','DD/MM/YYYY'),TO_DATE('14/11/2020','DD/MM/YYYY'),5);
```

```

insert into batch
values(5,TO_DATE('06/09/2017','DD/MM/YYYY'),TO_DATE('28/04/2019','DD/MM/YYYY'),2);

insert into medicine values('Vicodin',200,60,3);
insert into medicine values('Metacin',700,35,2);
insert into medicine values('Demerol',350,85,1);
insert into medicine values('Soframycin',500,30,2);
insert into medicine values('Adderall',950,70,5);

insert into prescription
values(1,TO_DATE('06/09/2018','DD/MM/YYYY'),TO_DATE('06/12/2018','DD/MM/YYYY'),3,4);
insert into prescription
values(2,TO_DATE('10/11/2018','DD/MM/YYYY'),TO_DATE('09/02/2019','DD/MM/YYYY'),1,3);
insert into prescription
values(3,TO_DATE('07/10/2018','DD/MM/YYYY'),TO_DATE('07/01/2019','DD/MM/YYYY'),1,5);
insert into prescription
values(4,TO_DATE('25/06/2018','DD/MM/YYYY'),TO_DATE('25/09/2018','DD/MM/YYYY'),3,2);
insert into prescription
values(5,TO_DATE('16/09/2018','DD/MM/YYYY'),TO_DATE('16/12/2018','DD/MM/YYYY'),2,1);

insert into pres_med values(1,'Demerol');
insert into pres_med values(2,'Vicodin');
insert into pres_med values(3,'Adderall');

insert into pharm_med values(1,'Soframycin');
insert into pharm_med values(2,'Demerol');
insert into pharm_med values(3,'Metacin');

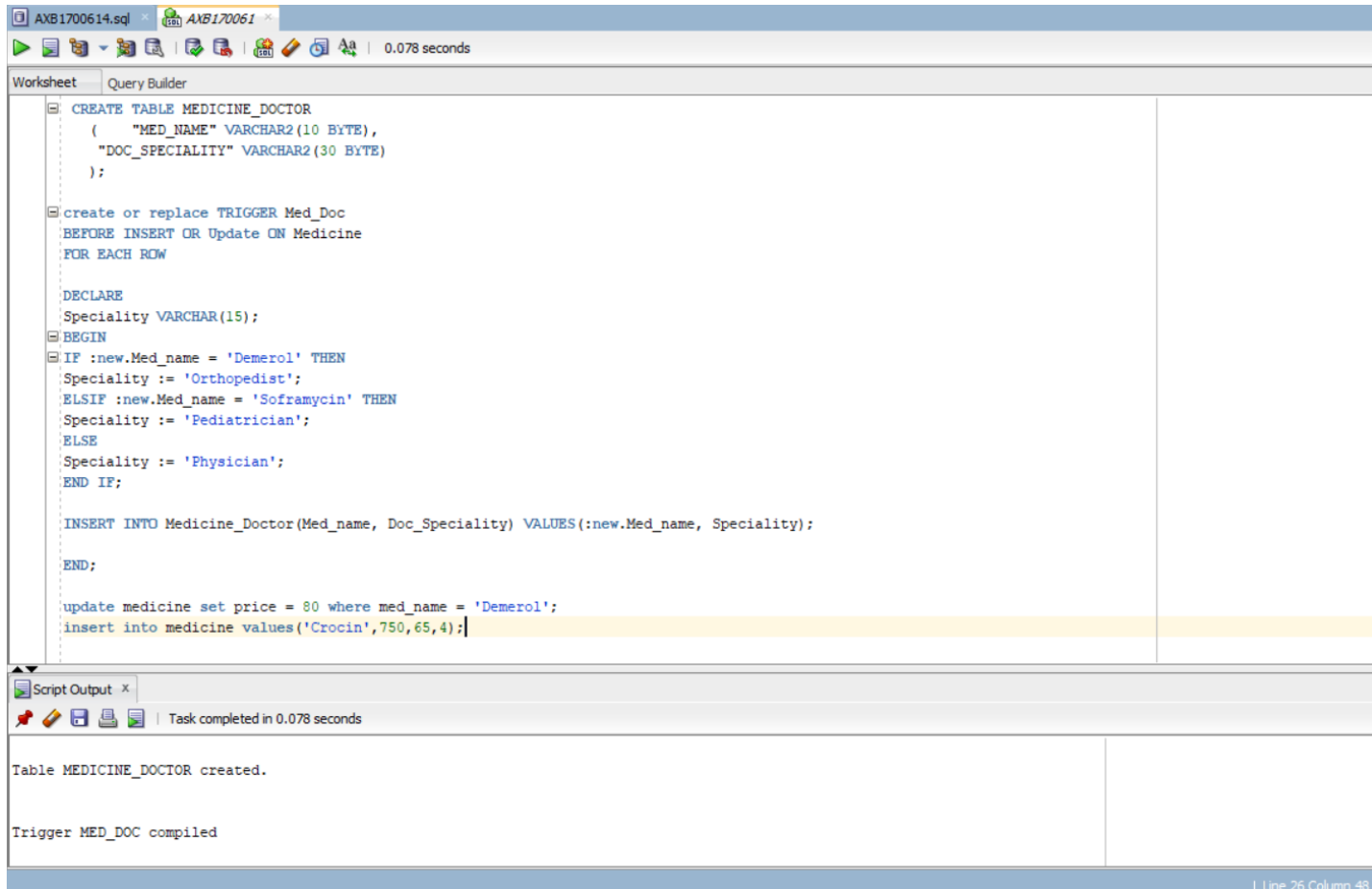
insert into man_phar values(1,3);
insert into man_phar values(2,1);
insert into man_phar values(3,5);
insert into man_phar values(4,2);
insert into man_phar values(5,4);

```

PL/SQL – Triggers

Trigger-I

Whenever there is any insert or update on medicine table, check the medicine name and Medicine_doctor table will be updated accordingly.



The screenshot displays the SQL Developer interface. The top pane shows a PL/SQL script with the following code:

```
CREATE TABLE MEDICINE_DOCTOR
(
  "MED_NAME" VARCHAR2(10 BYTE),
  "DOC_SPECIALITY" VARCHAR2(30 BYTE)
);

create or replace TRIGGER Med_Doc
BEFORE INSERT OR Update ON Medicine
FOR EACH ROW

DECLARE
Speciality VARCHAR(15);
BEGIN
IF :new.Med_name = 'Demerol' THEN
Speciality := 'Orthopedist';
ELSIF :new.Med_name = 'Soframycin' THEN
Speciality := 'Pediatrician';
ELSE
Speciality := 'Physician';
END IF;

INSERT INTO Medicine_Doctor(Med_name, Doc_Speciality) VALUES(:new.Med_name, Speciality);

END;

update medicine set price = 80 where med_name = 'Demerol';
insert into medicine values('Crocine',750,65,4);
```

The bottom pane, titled "Script Output", shows the execution results:

```
Table MEDICINE_DOCTOR created.

Trigger MED_DOC compiled
```

The status bar at the bottom right indicates "Line 26 Column 48".

When an update is made, Medicine_doctor will be updated

When an insert is made, Medicine_doctor is updated because of triggering action.

Medicine_doctor table after two triggering events:

	MED_NAME	DOC_SPECIALITY
1	Demerol	Orthopedist
2	Crocine	Physician

TRIGGER- II

Whenever there is any insert or update on medicine table, Med_manu table will be updated accordingly.

```
create table med_manu
(
  med_name varchar2(20),
  manufacturer_name varchar2(20));

create or replace TRIGGER MED_MANUFACTURER
BEFORE INSERT OR Update ON MEDICINE
FOR EACH ROW

DECLARE
  man_name VARCHAR(15);
BEGIN
  IF :new.Med_name = 'Demerol' THEN
    man_name := 'Abbott';
  ELSIF :new.Med_name = 'Soframycin' THEN
    man_name := 'Pfizer';
  ELSE
    man_name := 'GSK';
  END IF;

  INSERT INTO med_manu(med_name, manufacturer_name) VALUES(:new.Med_name, man_name);

END;

insert into medicine values('Demoral',650,45,4);
```

Script Output x

Task completed in 0.074 seconds

Table MED_MANU created.

Trigger MED_MANUFACTURER compiled

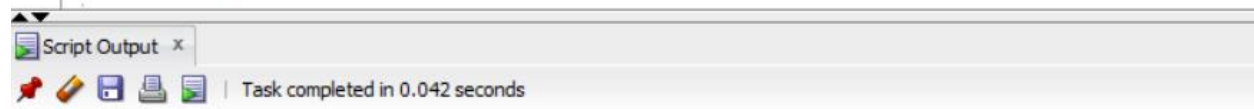
PROCEDURES:

Procedure I:

When this procedure is executed, it will display the medicines which are available at pharmacy stores.

```
create or replace procedure med_pharm
as
medicine_name medicine.med_name%type;
pharmacy_name pharmacy.name%type;
CURSOR medical IS
SELECT medicine.med_name,pharmacy.name
from patient, medicine, pharmacy, prescription, pres_med, pharm_med
where patient.patient_id = prescription.patient_id
and prescription.prescription_id = pres_med.prescription_id
and pres_med.med_name = medicine.med_name
and pharm_med.med_name = medicine.med_name
and pharm_med.pharmacy_id = pharmacy.pharmacy_id and patient.name = 'Kyle';
BEGIN
OPEN medical;
LOOP
FETCH medical into medicine_name,pharmacy_name;
EXIT WHEN(medical%NOTFOUND);
DBMS_OUTPUT.PUT_LINE('Medicine:'||medicine_name||' Pharmacy:'||pharmacy_name);
END LOOP;
CLOSE medical;
END;

set serveroutput on;
EXECUTE MED_PHARM;
```



Procedure MED_PHARM compiled

Medicine:Adderall Pharmacy:CVS

Medicine:Soframycin Pharmacy:Bluehealth

PL/SQL procedure successfully completed.

PROCEDURE II:

When this procedure is executed, it displays employees who work after 10 A.M. and the place where they work.

```
create or replace procedure emp_pharm
as
emp_name employee.name%type;
pharmacy_name pharmacy.name%type;
CURSOR shift IS
select employee.name, pharmacy.name
from employee, pharmacy
where employee.pharmacy_id = pharmacy.pharmacy_id
and EXTRACT(HOUR from cast(employee.shift_start AS TIMESTAMP)) >= 10;
BEGIN
OPEN shift;
LOOP
FETCH shift into emp_name, pharmacy_name;
EXIT WHEN (shift%NOTFOUND);
DBMS_OUTPUT.PUT_LINE('Employee:' || emp_name || ' Pharmacy:' || pharmacy_name);
END LOOP;
CLOSE shift;
END;

set serveroutput on;
EXECUTE EMP_PHARM;
```

Script Output x

Task completed in 0.084 seconds

Employee:Bezos Pharmacy:CVS
Employee:Mark Pharmacy:Redcross
Employee:Elon Pharmacy:Bluehealth

PL/SQL procedure successfully completed.