Milestone 1

Pharmaceutical Database

Course: ITEC4220 A
Term: Fall 2020

Team members:

Shidie Ren – 215439409
Ritish Bajaj – 215563323
Yuerui Yang – 215211030
Tushar Sroya – 215647639
Nikki Hosseini-Bidi – 216810715

Topic: Planning on making a health info tracking DBMS. This would be for a pharmacy so that they can keep track of new and returning customers and their prescriptions, and preexisting medical records.

3. *Identify major user views:*

a. *Identify possible user groups (no more than 3)*

1. Pharmacists
2. Patients (People prescribed medicine)

b. *Information requirements for each user view.*

1. Pharmacists
    - Date of Birth
    - PharmacistID
    - Male/Female
    - Name
    - Recent Date of Prescription
    - Family Doctor Name

    This would be used to access patient records, and bring down the time used in preparing the paperwork for a prescription
2. Medicine Data
    - Medicine Name
    - Brand Name
    - Stock
    - Price
3. Billing System
    - Patient Name
    - Tablets Per day
    - Drug ID/Medicine Name
    - Instructions
    - Date of Prescription
    - Total Price
4. Customer personnel
    - Name
    - Date of Birth
    - ID
    - Age
    - Gender
    - Phone Number
    - Insurance Info
5. Business Information
    - Name of pharmacy's
    - Address
    - Phone Number
    - Business License Number

4. Determine the scope of a database application. Describe the problem domain clearly articulating basic business rules. The description should be about 1 page long. All your business rules must be reflected on your class diagram and all parts of your class diagram must be justified by your business rules.

In this database, we are looking to solve a manual system within a pharmacy. In this case, the pharmacy decided to maintain its business procedures for prescriptions, stock, payment, pharmacy info, and patient and pharmacist personnel lists through paper-based systems. Fast forward to today, with the technology at hand, people have started to transition towards OODBMS to analyze their business procedures in a more organized and purposeful way. In this database system, we are including different roles within the system. This begins with the development of both a ManagementSystem class to track individual pharmacists' pay and salary within the unique pharmacy within the flow of both the pharmacist class and the pharmacy class. and the PharmacyManagement class that tracks the flow of medicine stock in and out of the Supply class to MedicineStock class process.

Initially, we have an Insurance class that allows the pharmacy database to keep a record of all patients' insurance information. Things such as a patient's insurance company and contact are stored in the class. This class is connected to the patient class.

The Patient class, it's connected to allow for the database to track and store patient information and the insurance company they\patient is associated with. For example, we can keep track of a patient's contact information. This class is connected to the Patient_Account, Insurance, and Prescription classes.

Moreover, the Patient-Account class exists to be used in the database to keep track of what patient accounts exist and the information regarding them. This class also allows for the access of any current actions being done on the specific business account. This class is connected to the Patient, Order, and the Bill class.

The Bill class holds information that relates to the bills that a patient would have to pay. Bills are organized by BillNo and referenced with a PatientID to ensure the bill is for the specific patient. Also, the Bill class allows for access to payment tender information, and payment calculations. Furthermore, the Bill class is connected order, Patient_Account, Medicene_Stock and contains the subclasses Credit_Payment, Cash_Payment, and Debit_Payment.
These subclasses contain very important functions in terms of bill payment. Credit_Payment makes credit payment easier by storing the patient's credit information. Cash_Payment collects information on how much cash the patient used to fulfill payment and how much change they received back. Debit_Payment makes the debit payments more database forward by storing the patient's debit information.

The Prescription class exists to ensure what prescription is being given to which patient, this is kept track of by having a Prescription and a referenced PatientID. The Prescription class is connected to the Patient and Pharmacist classes.

The Pharmacist class that Prescription is connected to, keeps track of information about pharmacists. Examples of things that are kept track of are PharmacistLicenseNo and their contact information. This allows for the retrieval, ordering, and viewing of all things medical related. This class connects to the Order, Prescription, Pharmacy, and Medicine_Stock classes.

The order class is connected to the Pharmacist class, and it records what a patient orders and the price regarding the order. The class also allows for a return on the price of said order. The following class is connected to the Bill, Patient-Account, and Pharmacist class.

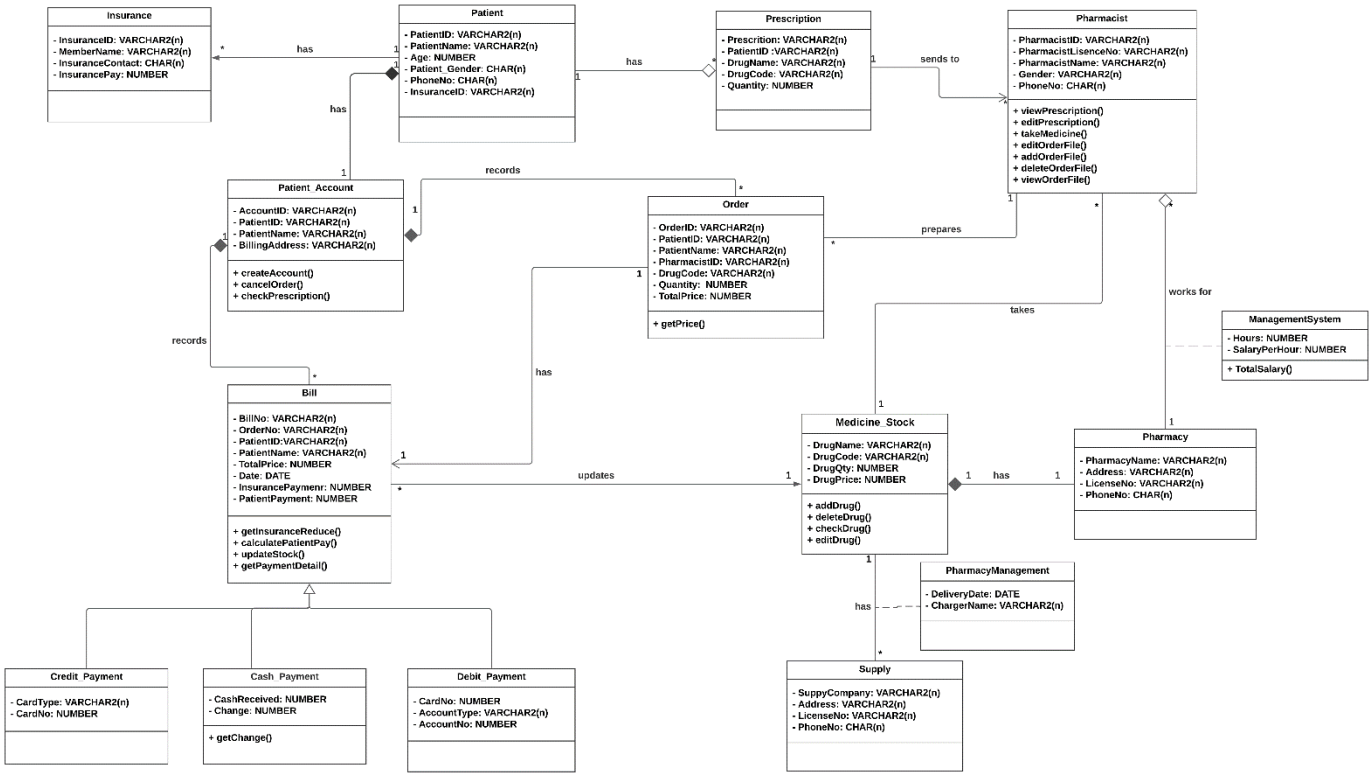
The Pharmacy class that is connected to the Pharmacist class exists to store information about the pharmacy with things such as the pharmacy's address. This class is connected to both the Medicine_Stock class and the Pharmacist class.

The Medicine_Stock class that is linked from the Pharmacy class, enables a view of the stock of medicine a pharmacy has to supply its customers. Medicine stock is updated through the classes many methods, and viewable because of the class. This class is connected to Supply, Pharmacy, Bill, and the Pharmacist class.

The Supply class exists in the database to keep track of medicine suppliers and their company information. The class is connected to the Medicine-Stock class.

Conceptual Design

Class Diagram for the Pharmacy Management System:



All attributes:

Attributes in Patient class: PatientID: VARCHAR2(n), PatientName: VARCHAR2(n), Age: NUMBER, Patient_Gender: VARCHAR2(n), PhoneNo: CHAR(n), InsuranceID: VARCHAR2(n).

Attributes in Insurance class: InsuranceID: VARCHAR2(n), MemberName: VARCHAR2(n), InsuranceContact: CHAR(n), InsurancePay: NUMBER.
Attributes in Patient_Account class: AccountID: VARCHAR2(n), PatientID: VARCHAR2(n), PatientName: VARCHAR2(n), BillingAddress: VARCHAR2(n).
Attributes in Bill class: BillNo: VARCHAR2(n), OrderNo: VARCHAR2(n), PatientID: VARCHAR2(n), PatientName: VARCHAR2(n), TotalPrice: NUMBER, Date: DATE, InsurancePaymenr: NUMBER, PatientPayment: NUMBER.
Attributes in Credit_Payment class: CardType: VARCHAR2(n), CardNo: NUMBER.
Attributes in Cash_Payment class: CashReceived: NUMBER, Change: NUMBER
Attributes in Debit_Payment class: CardNo: NUMBER, AccountType: VARCHAR2(n), AccountNo: NUMBER
Attributes in Prescrition class: Prescrition: VARCHAR2(n), PatientID: VARCHAR2(n), DoctorCode: VARCHAR2(n), DrugName: VARCHAR2(n), Quantity: NUMBER.
Attributes in Order class: OrderID: VARCHAR2(n), PatientID: VARCHAR2(n), PatientName: VARCHAR2(n), PharmacistID: VARCHAR2(n), DrugCode: VARCHAR2(n), Quantity: NUMBER, TotalPrice: NUMBER.
Attributes in Pharmacist class: PharmacistID: VARCHAR2(n), PharmacistLisenceNo: VARCHAR2(n), PharmacistName: VARCHAR2(n), Gender: VARCHAR2(n), PhoneNo: CHAR(n).
Attributes in Pharmacy class: PharmacyName: VARCHAR2(n), Address: VARCHAR2(n), LicenseNo: VARCHAR2(n), PhoneNo: CHAR(n).
Attributes in Medicine_Stock class: DrugName: VARCHAR2(n), DrugCode: VARCHAR2(n), DrugQty: NUMBER, DrugPrice: NUMBER.
Attributes in ManagementSystem class: Hours: NUMBER, SalaryPerHour: NUMBER.
Attributes in PharmacyManagement class: DeliveryDate: DATE, ChargerName: VARCHAR2(n).
Attributes in Supply class: SuppyCompany: VARCHAR2(n), Address: VARCHAR2(n), LicenseNo: VARCHAR2(n), PhoneNo: CHAR(n).

Cardinality:

Patient-Patient_Account (1:1)
Patient-Insurance (1:*)
Patient-Prescription (1:*)
Patient_Account-Order (1:*)
Patient_Account-Bill (1:*)
Order-Bill (1:1)
Bill-Medicine_Stock(*:1)
Pharmacy-Medicine_Stock (1:1)
Pharmacist-Medicine Stock(*:1)
Pharmacist-Order (1:*)
Pharmacist-Prescription (1:*)
Pharmacist-Pharmacy (*:1)
Medicine_Stock-Supply (1:*)

Object Methods

Patient_Account.createAccount(): add information into AccountID, PatientID, PatientName and BillingAddress.
Patient_Account.cancelOrder(): record orderID which is the order that customer canceled.
Patient_Account.checkPrescription(): check the information from the patient and record them.

Order.getPrice(): calculate price using DrugQty * DrugPrice, return the total price for the order.

Bill.getInsuranceReduce(): calculate insurance coverage for the order.
Bill. calculatePatientPay (): by deducting insurance cover and calculate total bill for the order.
Bill.updateStock(): after patient pay the bill, updating the medicine amount in the stock.
Bill.getPaymentDetail(): get the way of payment information.

Cash_Payment.getChange(): calculate cash change by using received cash amount deduct bill.

Pharmacist.viewPrescription(): check information from the prescription.
Pharmacist.editPrescription(): edit information from the prescription.
Pharmacist.takeMedicine(): calculate medicine amount that patient needs.
Pharmacist.editOrderFile(): edit information from the order records.
Pharmacist.addOrderFile(): add a new order record into the order files.
Pharmacist.deleteOrderFile(): delete an order record from order files.
Pharmacist.viewOrderFile(): check information from the order records.

Medicine_Stock.addDrug(): add new kinds of medicine into the inventory.
Medicine_Stock.deleteDrug(): delete medicine information from the inventory.
Medicine_Stock.checkDrug(): check medicine information and inventory amount.
Medicine_Stock.editDrug(): edit inventory amount.

ManagementSystem.TotalSalary(): calculate employees' salary by using Hours * SalaryPerHour.

Group #7 SQL Queries

1. **Find the names of the companies supplying drugs for prescription of Patient John Collins. (Required by Prof)**

```
Select d.supplierName.SupplyCompany, ph.Prescribe.pid.PatientName
From DrugOrder_t d,PrescriptionHandler_t ph
Where ph.Prescribe.pid.PatientName = 'John Collins'
AND d.drugcode.DrugCode = ph.dcode.DrugCode;
```

2. **find the name, phone number and refill date for patient who have more than one refill.**

```
Select ph.Prescribe.pid.PatientName, ph.Prescribe.pid.PhoneNo, ph.RefillDate,ph.refillsLeft()
From PrescriptionHandler_t ph
Where ph.refillsLeft(5)>1
Order by ph.RefillDate;
```

3. **Find bill Numbers and patient names for card payment that were paid by Humana Insurance company in the month of April 2020.**

```
Select cd.BillNo, ph.Prescribe.pid.PatientName, cd.billDate
From PrescriptionHandler_t ph, Card_Payment_t cd
Where ph.Prescribe.pid.insurid.insuranceName = 'Humana'
AND cd.billDate >='01-Apr-2020' And cd.billDate<='30-Apr-2020'
AND cd.pid.PatientID=ph.Prescribe.pid.PatientID
Order by ph.Prescribe.pid.PatientName;
```

4. **Find Patients and Account IDs that had drugs supplied by Apotex.**

```
Select ph.Prescribe.pid.PatientName, pa.AccountId
From DrugOrder_t d,PrescriptionHandler_t ph, Patient_Account_t pa
Where d.supplierName.SupplyCompany= 'Apotex'
AND d.drugcode.DrugCode = ph.dcode.DrugCode
```

```
AND ph.Prescribe.pid.PatientId = pa.pid.PatientId;
```

5. **Find patientname and the relative prescribed drugname for those patients that have a prescription for a medicine that needs to be restocked.**

```
Select ph.drugQuantity, ph.dcode.DrugQty, ph.Prescribe.pid.PatientName, ph.dcode.DrugName
From PrescriptionHandler_t ph
Where ph.drugQuantity > ph.dcode.DrugQty;
```

6. **Find all PatientNames and the relevant prescriptionID that didn't have a drug from Sanofi Pasteur.**

```
Select ph.Prescribe.pid.PatientName, ph.Prescribe.prescriptionID
From DrugOrder_t d,PrescriptionHandler_t ph
Where d.supplierName.SupplyCompany!= 'Sanofi Pasteur'
AND d.drugcode.DrugCode = ph.dcode.DrugCode;
```

7. **Find PatientNames that had a perscritpion for Metformin.**

```
Select ph.Prescribe.pid.PatientName, ph.dcode.DrugName
From PrescriptionHandler_t ph
Where ph.dcode.DrugName = 'Metformin';
```

8. **Find the billing address for patients made payment using debit card.**

```
Select cd.CardType, cd.pid.PatientName, pa.BillingAdress
From Card_Payment_t cd, Patient_Account_t pa
Where cd.CardType = 'Debit'
AND cd.pid.PatientId = pa.pid.PatientId;
```

9. **Find the patient name , bill number and date for cash payment over 50 dollars.(Overriding method)**

```
Select c.pid.PatientName,c.BillNo, c.BillDate, round(c.CalculatePatientPay(),2)
From Cash_Payment_t c
```

```
Where c.CalculatePatientPay()>50;
```

10. **Compare prices of other drug orders to the price of an order of acetaminophen. Listed from lowest price to highest**

```
Select d.drugcode.DrugName, d.supplierName.SupplyCompany,d.price,

d.sortOrder(DrugOrder('5565543', (select ref(m) from Medicine_Stock_t m where m.DrugCode =
'3816452222'), 5, 5303, '15-Jul-2020', (select ref(s) from Supplier_t s where s.SupplyCompany =
'Sanofi Pasteur')))

From DrugOrder_t d

Order by Value(d);
```

11. **Find drug name in the prescription for patients over 35 years old . (using map method)**

```
Select ph.Prescribe.pid.mapPatient() ,ph.Prescribe.pid. PatientName, ph.dcode.DrugName from
PrescriptionHandler_t ph

Where ph.Prescribe.pid.mapPatient() > 35

Order by ph.Prescribe.pid.mapPatient();
```

```
Enter user-name: Grp7@studb10g
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options


SQL>
create type Insurance as object (InsuranceID varchar2(10), InsuranceName
varchar2(20), InsuranceContact char(10), InsurancePercentOff number(10));
  2
  3   /
create type Insurance as object (InsuranceID varchar2(10), InsuranceName
varchar2(20), InsuranceContact char(10), InsurancePercentOff number(10));
            *
ERROR at line 1:
ORA-00955: name is already used by an existing object


begin
   for trec in ( select table_name from user_object_tables )
    loop
    execute immediate 'drop table '||trec.table_name||' purge';
    end loop;
   end;
  7    /
begin
*
ERROR at line 1:
ORA-00054: resource busy and acquire with NOWAIT specified or timeout
expired
ORA-06512: at line 4


begin
  for  trec in ( select type_name from user_types )
   loop
   execute immediate 'drop type '||trec.type_name||' force';
   end loop;
   end;
  7     /

PL/SQL procedure successfully completed.


SQL>
create type Insurance as object (InsuranceID varchar2(10), InsuranceName
varchar2(20), InsuranceContact char(10), InsurancePercentOff number(10));
  2
  3   /
```

```
Type created.


SQL>
SQL> create type Patient as object (PatientID varchar2(10), PatientName
varchar2(20), BirthYear number(4), Patient_Gender char(1), PhoneNo
char(10), Insurid ref Insurance, map member function mapPatient return
integer);
  2  /

Type created.


SQL>
create type Patient_Account as object (AccountID varchar2(10), pid ref
Patient, BillingAdress varchar2(40), HomeAdress varchar2(50));
  2
  3   /

Type created.


SQL>
create type Bill as object (BillNo varchar2(10), pid ref Patient,
TotalPrice number(6,2), billDate date, InsuranceOff number(10) , member
function  getInsuranceReduce return number, member function
calculatePatientPay return number) NOT FINAL;
  2
  3   /

Type created.


SQL>
SQL> create type Card_Payment under Bill(CardType varchar2(10), CardNo
varchar2(12));
  2  /

Type created.


SQL>
SQL> create type Cash_Payment under Bill(CashReceived number(6),
CashPercentageOff number(6), member function getChange return number,
overriding member function calculatePatientPay return number);
  2  /

Type created.


SQL>
create type Medicine_Stock as object (DrugCode varchar2(10), DrugName
varchar2(20), DrugPrice number(6,2), DrugQty integer,
```

```
  2
  3  member function getDrugQty(drugCode in varchar2) return integer,
member procedure updateStock(drugCode in varchar2, quantity in number));
  4  /

Type created.


SQL>
SQL> create type Prescription as object(PrescriptionID varchar2(10), pid
REF Patient);
  2  /

Type created.

SQL> create type  PrescriptionHandler as object(handlerID varchar2(10),
prescribe ref Prescription, dcode ref Medicine_Stock, drugQuantity
integer, Refills number(3),  RefillDate date, member function refillsLeft
return number);
  2  /

Type created.

SQL> create type Supplier as object (SupplyCompany varchar2(20), Address
varchar2(20), LicenseNo varchar2(15), PhoneNo varchar2(12));
  2  /

Type created.

SQL> create type DrugOrder as object(OrderNo varchar2(10), drugcode ref
Medicine_Stock, OrderQty integer, Price number, OrderDate date,
supplierName ref Supplier, member function getDiscount return number,
member function getTotalPrice return number, order member function
sortOrder(orders in DrugOrder) return integer);
  2  /

Type created.

CREATE TABLE Insurance_t of Insurance (InsuranceID primary key);


Table created.

SQL> SQL>
CREATE TABLE Patient_t of Patient (PatientID primary key);


Table created.

 CREATE TABLE Patient_Account_t of Patient_Account (AccountID primary
key);
```

```
CREATE TABLE Bill_t of Bill (BillNo primary key);


CREATE TABLE Cash_Payment_t of Cash_Payment (BillNo primary key);


CREATE TABLE Card_Payment_t  of Card_Payment (BillNo primary key);


CREATE TABLE Medicine_Stock_t of Medicine_Stock (DrugCode primary key);

Table created.


CREATE TABLE Prescription_t of Prescription (PrescriptionID primary key);


CREATE TABLE PrescriptionHandler_t of PrescriptionHandler (handlerID
primary key);


CREATE TABLE Supplier_t of Supplier (SupplyCompany primary key);


CREATE TABLE DrugOrder_t of DrugOrder (orderNo primary key);


Table created.

SQL> SQL> SQL> SQL>
Table created.

SQL> SQL> SQL> SQL>
Table created.

SQL> SQL> SQL> SQL>
Table created.

SQL> SQL> SQL> SQL>
Table created.

SQL> SQL> SQL> SQL>
Table created.

SQL> SQL> SQL> SQL>
Table created.
```

```
SQL> SQL> SQL> SQL> CREATE TABLE DrugOrder_t of DrugOrder (orderNo primary
key)
             *
ERROR at line 1:
ORA-00955: name is already used by an existing object


 create type body Patient as map member function mapPatient return integer
  2
is
  4
begin
  6
return 2020 - self.BirthYear;
  8
end;
 10
 11  end;
 12  /

Type body created.

create type body Bill as member function getInsuranceReduce return number
  2
is
  4
begin
  6
return self.TotalPrice*self.insuranceOff;
  8
end;
 10
member function calculatePatientPay return number
 12
is
 14
begin
 16
return self.TotalPrice*(1-self.InsuranceOff/100);
 18
end;
 20
 21  end;
 22  /

Type body created.

create type body Cash_Payment as overriding member function
calculatePatientPay return number
  2
is
  4
begin
```

```
  6
return self.TotalPrice*(1-InsuranceOff/100)*(1-CashPercentageOff/100);
  8
end;
 10
 member function getChange return number
 12
is
 14
begin
 16
 return self.CashReceived - self.TotalPrice*(1-InsuranceOff/100)*(1-
CashPercentageOff/100);
 18
end;
 20
 21  end;
 22  /

Type body created.

create type body Medicine_Stock as member function getDrugQty(drugCode in
varchar2) return integer
  2
is
  4
drugNum integer := 0;
  6
begin
  8
select self.DrugQty into drugNum from Medicine_Stock_t
 10
where self.DrugCode = drugCode;
 12
return self.DrugQty;
 14
end;
 16
member procedure updateStock(drugCode in varchar2, quantity in number)
 18
is
 20
drugQ integer := 0;
 22
begin
 24
self.DrugQty := quantity;
 26
select self.DrugQty into drugQ from Medicine_Stock_t
 28
where self.DrugCode = drugCode;
 30
DBMS_OUTPUT.PUT_LINE(self.DrugQty);
 32
```

```
end;
 34
 35  end;
 36  /

Type body created.

create type body DrugOrder as member function getDiscount return number
  2
is
  4
begin
  6
if self.OrderQty >10 then
  8
return self.Price*self.OrderQty*0.05;
 10
end if;
 12
end;
 14
member function getTotalPrice return number
 16
is
 18
begin
 20
if self.getDiscount > 0 then
 22
return self.Price*self.OrderQty*0.95;
 24
else
 26
return self.Price*self.OrderQty;
 28
end if;
 30
end;
 32
order member function sortOrder(orders in DrugOrder) return integer
 34
is
 36
result integer := 0;
 38
begin
 40
if self.Price < orders.Price then
 42
result := -1;
 44
else
 46
if self.Price > orders.Price then
```

```
 48
result := 1;
 50
else
 52
result := 0;
 54
end if;
 56
end if;
 58
return result;
 60
end;
 62
end;
 64
 65   /

Type body created.


SQL>
create type body PrescriptionHandler as member function refillsLeft return
number
  2
is
  4
begin
  6
Return self.DrugQuantity - self.Refills ;
  8
end;
 10
 11  end;
 12  /

Type body created.


SQL>
insert into Insurance_t values(Insurance('3847343255', 'PharmaAID',
'6477056399', 32));

1. row created.

SQL>
insert into Insurance_t values(Insurance('43432434', 'Rexall+',
'4163563116', 43));

1. row created.

SQL>
```

```
insert into Insurance_t values(Insurance('46556555', 'Humana',
'4165551232', 54));

1. row created.

SQL>
SQL> insert into Insurance_t values(Insurance('3321134', 'Anthem',
'6473552192', 12));

1. row created.


SQL>
insert into Patient_t values(Patient('6838123233', 'Rick Astley', 1983,
'M', '4169276542', (select ref(i) from Insurance_t i where i.InsuranceID =
'3847343255')));

1. row created.

SQL>

SQL>

SQL>
insert into Patient_t values(Patient('444332312', 'John Collins', 1975,
'M', '4163554642', (select ref(i) from Insurance_t i where i.InsuranceID =
'43432434')));

1. row created.

SQL>

SQL>

SQL>
insert into Patient_t values(Patient('343434323', 'Peter Parker', 1995,
'M', '4163819322', (select ref(i) from Insurance_t i where i.InsuranceID =
'3321134')));

1. row created.

SQL>

SQL>

SQL>

SQL>
SQL> insert into Patient_t values(Patient('75748333', 'Maurice Loveheart',
1981, 'F', '4160992444', (select ref(i) from Insurance_t i where
i.InsuranceID = '46556555')));

1. row created.
```

```
insert into Patient_Account_t values(Patient_Account(1342657322, (select
ref(p) from Patient_t p where p.PatientID = '6838123233'), '245 Eisenhower
Crescent ', '245 Eibenhower Crescent '));

1. row created.

SQL>

SQL>
insert into Patient_Account_t values(Patient_Account('55833636', (select
ref(p) from Patient_t p where p.PatientID = '444332312'), '45 Harlove Road
', '82 Dove Street '));

1. row created.

SQL>

SQL>
insert into Patient_Account_t values(Patient_Account('3477767', (select
ref(p) from Patient_t p where p.PatientID = '343434323'), '83 Humber
Avenue ', '83 Humber Avenue '));

1. row created.

SQL>

SQL>
SQL> insert into Patient_Account_t values(Patient_Account('164223',
(select ref(p) from Patient_t p where p.PatientID = '75748333'), '75
Brampton Road ', '75 Brampton Road '));

1. row created.

Insert into Card_Payment_t values(Card_Payment('132324',(select ref(p)
from Patient_t p where p.PatientID = '6838123233'), 233.45, '10-Mar-2020',
32, 'Credit', 450566473838));

1. row created.

SQL>

SQL>
Insert into Card_Payment_t values(Card_Payment('234555',(select ref(p)
from Patient_t p where p.PatientID = '444332312'), 99.52, '1-Jan-2020',
43, 'Debit', '949473722347'));

1. row created.

SQL>

SQL>
Insert into Card_Payment_t values(Card_Payment('32651',(select ref(p) from
Patient_t p where p.PatientID = '343434323'), 56.22, '20-Feb-2020', 12,
'Debit', '383846632887'));
```

```
1. row created.

SQL>

SQL>
SQL> Insert into Card_Payment_t values(Card_Payment('573778',(select
ref(p) from Patient_t p where p.PatientID = '75748333'), 122.13, '27-Apr-
2020', 54, 'Credit', '927555523124'));

1. row created.

Insert into Cash_Payment_t values(Cash_Payment('110032020',(select ref(p)
from Patient_t p where p.PatientID = '6838123233'), 343.45, '20-Mar-2020',
32, 250, 10));

1. row created.

SQL>

SQL>
Insert into Cash_Payment_t values(Cash_Payment('18042020',(select ref(p)
from Patient_t p where p.PatientID = '343434323'), 30.45, '18-Apr-2020',
12 , 28, 5));

1. row created.

SQL>

SQL>
Insert into Cash_Payment_t values(Cash_Payment('20052020',(select ref(p)
from Patient_t p where p.PatientID = '75748333'), 123.8, '20-May-2020',
54, 60, 5));

1. row created.

SQL>

SQL>
SQL> Insert into Cash_Payment_t values(Cash_Payment('5032020',(select
ref(p) from Patient_t p where p.PatientID = '444332312'), 102.5, '5-Mar-
2020',43, 60, 5));

1. row created.

insert into Medicine_Stock_t values(Medicine_Stock('6327272244',
'Metformin', 1323.32, 13));

1. row created.

SQL>
insert into Medicine_Stock_t values(Medicine_Stock('3816452222',
'Acetaminophen', 237.98, 25));
```

```
1. row created.

SQL>
insert into Medicine_Stock_t values(Medicine_Stock('8783111133',
'Amoxicillin', 373.43, 52));

1. row created.

SQL>
SQL> Insert into Medicine_Stock_t values(Medicine_Stock('6354427262',
'ibuprofen', 572.43, 38));

1. row created.


SQL>
Insert into Prescription_t values(Prescription( '8383524', (select ref(p)
from Patient_t p where p.PatientID = '6838123233')));

1. row created.

SQL>

SQL>
Insert into Prescription_t values(Prescription( '4522234', (select ref(p)
from Patient_t p where p.PatientID = '444332312')));

1. row created.

SQL>

SQL>
Insert into Prescription_t values(Prescription( '7462653', (select ref(p)
from Patient_t p where p.PatientID = '343434323')));

1. row created.

SQL>

SQL>
SQL> Insert into Prescription_t values(Prescription( '1288634', (select
ref(p) from Patient_t p where p.PatientID = '75748333')));

1. row created.


SQL>
Insert into PrescriptionHandler_t values
(PrescriptionHandler('00000001',(select ref(pr) from Prescription_t pr
where pr.PrescriptionID = '8383524'), (select ref(m) from Medicine_Stock_t
m where m.DrugCode = '6327272244'), 12, 12, '12-MAY-2020'));

1. row created.
```

```
SQL>

SQL>
Insert into PrescriptionHandler_t values(
PrescriptionHandler('00000002',(select ref(pr) from Prescription_t pr
where pr.PrescriptionID = '4522234'), (select ref(m) from Medicine_Stock_t
m where m.DrugCode = '3816452222'), 32, 25, '28-Feb-2020'));

1. row created.

SQL>

SQL>
Insert into PrescriptionHandler_t
values(PrescriptionHandler('00000003',(select ref(pr) from Prescription_t
pr where pr.PrescriptionID = '7462653'), (select ref(m) from
Medicine_Stock_t m where m.DrugCode = '8783111133'), 15, 5, '13-Mar-
2020'));

1. row created.

SQL>

SQL>
SQL> Insert into PrescriptionHandler_t
values(PrescriptionHandler('00000004',(select ref(pr) from Prescription_t
pr where pr.PrescriptionID = '1288634'), (select ref(m) from
Medicine_Stock_t m where m.DrugCode = '6354427262'), 11, 8, '14-Feb-
2020'));

1. row created.

Insert Into Supplier_t values(Supplier('Sanofi Pasteur', '1755 Steeles
Ave', 'SN10077', '4166672700'));

1. row created.

SQL>

SQL>
Insert Into Supplier_t values(Supplier('Apotex', '150 Signet Dr ',
'AN67821', '8002684623'));

1. row created.

SQL>

SQL>
SQL> Insert Into Supplier_t values(Supplier('Pfizer Canada', '110 Maple
Dr', 'SI07671', '4164986037'));

1. row created.
```

```
SQL>
SQL> Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m)
from Medicine_Stock_t m where m.DrugCode = '3816452222'), 5, 5303, '15-
Jul-2020', (select ref(s) from Supplier_t s where s.SupplyCompany =
'Sanofi Pasteur')));
Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m) from
Medicine_Stock_t m where m.DrugCode = '3816452222'), 5, 5303, '15-Jul-
2020', (select ref(s) from Supplier_t s where s.SupplyCompany = 'Sanofi
Pasteur')))
           *
ERROR at line 1:
ORA-04063: table "GRP7.DRUGORDER_T" has errors


SQL> Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m)
from Medicine_Stock_t m
  2. where m.DrugCode = '3816452222'), 5, 5303, '15-Jul-2020', (select
     ref(s) from Supplier_t s where
  3. s.SupplyCompany = 'Sanofi Pasteur')));
Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m) from
Medicine_Stock_t m
           *
ERROR at line 1:
ORA-04063: table "GRP7.DRUGORDER_T" has errors


SQL> drop table DrugOrder;
drop table DrugOrder
          *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> CREATE TABLE DrugOrder_t of DrugOrder (orderNo primary key);
CREATE TABLE DrugOrder_t of DrugOrder (orderNo primary key)
             *
ERROR at line 1:
ORA-00955: name is already used by an existing object


SQL> Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m)
from Medicine_Stock_t m where m.DrugCode = '3816452222'), 5, 5303, '15-
Jul-2020', (select ref(s) from Supplier_t s where s.SupplyCompany =
'Sanofi Pasteur')));
Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m) from
Medicine_Stock_t m where m.DrugCode = '3816452222'), 5, 5303, '15-Jul-
2020', (select ref(s) from Supplier_t s where s.SupplyCompany = 'Sanofi
Pasteur')))
           *
ERROR at line 1:
ORA-04063: table "GRP7.DRUGORDER_T" has errors
ORA-04063: table "GRP7.DRUGORDER_T" has errors
ORA-04063: table "GRP7.DRUGORDER_T" has errors
```

```
SQL> drop table DrugOrder_t;

Table dropped.

SQL> CREATE TABLE DrugOrder_t of DrugOrder (orderNo primary key);

Table created.


SQL>
SQL> Insert into DrugOrder_t values(DrugOrder('5565543', (select ref(m)
from Medicine_Stock_t m where m.DrugCode = '3816452222'), 5, 5303, '15-
Jul-2020', (select ref(s) from Supplier_t s where s.SupplyCompany =
'Sanofi Pasteur')));

1. row created.


SQL>
SQL> Insert into DrugOrder_t values(DrugOrder('8725384', (select ref(m)
from Medicine_Stock_t m where m.DrugCode = '8783111133'), 15, 6200, '31-
Oct-2020', (select ref(s) from Supplier_t s where s.SupplyCompany =
'Apotex')));

1. row created.

Insert into DrugOrder_t values(DrugOrder('4322347', (select ref(m) from
Medicine_Stock_t m where m.DrugCode = '6327272244'), 23, 800, '10-Sep-
2020', (select ref(s) from Supplier_t s where s.SupplyCompany = 'Pfizer
Canada')));

1. row created.

SQL>
SQL>  Select d.supplierName.SupplyCompany, ph.Prescribe.pid.PatientName
  2. From DrugOrder_t d,PrescriptionHandler_t ph
  3. Where ph.Prescribe.pid.PatientName = 'John Collins'
  4. AND d.drugcode.DrugCode = ph.dcode.DrugCode;

SUPPLIERNAME.SUPPLYC PRESCRIBE.PID.PATIEN
-------------------- --------------------
Sanofi Pasteur        John Collins

SQL> Select ph.Prescribe.pid.PatientName, ph.Prescribe.pid.PhoneNo,
ph.RefillDate,ph.refillsLeft()
  5. From PrescriptionHandler_t ph
  6. Where ph.refillsLeft()>1
  7. Order by ph.RefillDate;

PRESCRIBE.PID.PATIEN PRESCRIBE. REFILLDAT PH.REFILLSLEFT()
-------------------- ---------- --------- ----------------
Maurice Loveheart    4160992444 14-FEB-20           3
John Collins          4163554642 28-FEB-20          7
```

```
Peter Parker          4163819322 13-MAR-20         10

SQL> Select cd.BillNo, ph.Prescribe.pid.PatientName, cd.billDate
  8. From PrescriptionHandler_t ph, Card_Payment_t cd
  9. Where ph.Prescribe.pid.insurid.insuranceName = 'Humana'
  10. AND cd.billDate >='01-Apr-2020' And cd.billDate<='30-Apr-2020'
  11. AND cd.pid.PatientID=ph.Prescribe.pid.PatientID
  12. Order by ph.Prescribe.pid.PatientName;

BILLNO        PRESCRIBE.PID.PATIEN BILLDATE
---------- -------------------- ---------
573778        Maurice Loveheart   27-APR-20

SQL> Select ph.Prescribe.pid.PatientName, pa.AccountId
  13. From DrugOrder_t d,PrescriptionHandler_t ph, Patient_Account_t pa
  14. Where d.supplierName.SupplyCompany= 'Apotex'
  15. AND d.drugcode.DrugCode = ph.dcode.DrugCode
  16. AND ph.Prescribe.pid.PatientId = pa.pid.PatientId;

PRESCRIBE.PID.PATIEN ACCOUNTID
-------------------- ----------
Peter Parker          3477767

SQL> Select ph.drugQuantity, ph.dcode.DrugQty,
ph.Prescribe.pid.PatientName, ph.dcode.DrugName
  17. From PrescriptionHandler_t ph
  18. Where ph.drugQuantity > ph.dcode.DrugQty;

DRUGQUANTITY DCODE.DRUGQTY PRESCRIBE.PID.PATIEN DCODE.DRUGNAME
------------ ------------- -------------------- --------------------
             32        25 John Collins  Acetaminophen


SQL>
SQL> Select ph.Prescribe.pid.PatientName, ph.Prescribe.prescriptionID
  19. From DrugOrder_t d,PrescriptionHandler_t ph
  20. Where d.supplierName.SupplyCompany!= 'Sanofi Pasteur'
  21. AND d.drugcode.DrugCode = ph.dcode.DrugCode;

PRESCRIBE.PID.PATIEN PRESCRIBE.
-------------------- ----------
Rick Astley     8383524
Peter Parker          7462653

SQL> Select ph.Prescribe.pid.PatientName, ph.dcode.DrugName
  22. From PrescriptionHandler_t ph
  23. Where ph.dcode.DrugName = 'Metformin';

PRESCRIBE.PID.PATIEN DCODE.DRUGNAME
-------------------- --------------------
Rick Astley     Metformin

SQL> Select cd.CardType, cd.pid.PatientName, pa.BillingAdress
  24. From Card_Payment_t cd, Patient_Account_t pa
```

```
  25. Where cd.CardType = 'Debit'
  26. AND cd.pid.PatientId = pa.pid.PatientId;

CARDTYPE   PID.PATIENTNAME   BILLINGADRESS
---------- ------------------- ---------------------------------------
Debit    John Collins  45 Harlove Road
Debit    Peter Parker  83 Humber Avenue


SQL>
SQL> Select c.pid.PatientName,c.BillNo, c.BillDate,
round(c.CalculatePatientPay(),2)
  27. From Cash_Payment_t c
  28. Where c.CalculatePatientPay()>50;

PID.PATIENTNAME      BILLNO  BILLDATE  ROUND(C.CALCULATEPATIENTPAY(),2)
-------------------- ---------- --------- -------------------------------
Rick Astley     110032020   20-MAR-20                  210.19
Maurice Loveheart    20052020     20-MAY-20                    54.1
John Collins          5032020     05-MAR-20                    55.5


SQL>
SQL> Select d.drugcode.DrugName, d.supplierName.SupplyCompany,d.price,
  29. d.sortOrder(DrugOrder('5565543', (select ref(m) from
      Medicine_Stock_t m where m.DrugCode = '3816452222'), 5, 5303, '15-
      Jul-2020', (select ref(s) from Supplier_t s where s.SupplyCompany =
      'Sanofi Pasteur')))
  30. From DrugOrder_t d
  31. Order by Value(d);

DRUGCODE.DRUGNAME    SUPPLIERNAME.SUPPLYC      PRICE
-------------------- -------------------- ----------
D.SORTORDER(DRUGORDER('5565543',(SELECTREF(M)FROMMEDICINE_STOCK_TMWHEREM.D
RUGCOD
--------------------------------------------------------------------------
------
Metformin        Pfizer Canada           800

                      -1

Acetaminophen         Sanofi Pasteur         5303

                       0

Amoxicillin      Apotex            6200

                       1


SQL> Select ph.Prescribe.pid.mapPatient() ,ph.Prescribe.pid.PatientName,
ph.dcode.DrugName  from PrescriptionHandler_t ph
  32. Where ph.Prescribe.pid.mapPatient() > 35
  33. Order by ph.Prescribe.pid.mapPatient();
```

```
PH.PRESCRIBE.PID.MAPPATIENT() PRESCRIBE.PID.PATIEN DCODE.DRUGNAME
------------------------------ -------------------- --------------------
                                37 Rick Astley   Metformin
                                39 Maurice Loveheart    ibuprofen
                                45 John Collins        Acetaminophen

SQL>
```

Milestone 3

Pharmaceutical Database

Course: ITEC4220 A
Term: Fall 2020

Team members:

Shidie Ren – 215439409
Ritish Bajaj – 215563323
Yuerui Yang – 215211030
Tushar Sroya – 215647639
Nikki Hosseini-Bidi – 21681071

NUMBER TWO

1. **Pharmacist – 3 Queries**

Find drugname, drug quantities that Peter Parker's bought.

```
select xmlroot(xmlelement("Prescriptionlist", xmlagg(xmlelement(
"PatientPrescription", xmlattributes(ph.Prescribe.pid.PatientName as "pname"),
xmlagg( xmlelement("drug", xmlattributes(ph.dcode.DrugCode as "drugcode"),
xmlelement("drugname",ph.dcode.DrugName), xmlelement("drugquantities", ph.drugQuantity)
)))),version '1.0') as result
from PrescriptionHandler_t ph
where ph.Prescribe.pid.PatientName ='Peter Parker'
group by ph.Prescribe.pid.PatientName;
```

Find customer accountID, home address, refills and refill date for patient who have more than three refills.

```
select xmlroot(xmlelement("Prescriptionlist", xmlagg(xmlelement(
"PatientPrescription",xmlagg(xmlelement("patientaccountid", xmlattributes(pt.AccountID as
"aid"),xmlelement("homeaddress", pt.HomeAdress),xmlelement("refilldate",ph.RefillDate),
xmlelement("refills",ph.refillsLeft()))))),version '1.0') as result
from PrescriptionHandler_t ph, Patient_Account_t pt
where ph.refillsLeft()>3
And ph.Prescribe.pid.PatientID = pt.pid.PatientID
group by pt.AccountID;
```

Find drug prescriptions to patients with between 5 to 25 refills and display their associating customer info

```
select xmlroot(xmlelement("drugs", xmlagg(xmlelement("patientinfo",
xmlagg(xmlelement("patient", xmlattributes(ph.prescribe.pid.PatientId as "pid"),
xmlelement("patientname", ph.prescribe.pid.patientname),
XMLELEMENT("prescription",ph.dcode.DrugName))))),version '1.0')

from PrescriptionHandler_t ph where ph.refills > 5 AND ph.refills < 25

group by ph.prescribe.pid.insurid.InsuranceID;
```

2. **Medicine Data – 3 Queries**

Find all drugs' names grouped by supplier:

```
Select xmlroot(xmlelement("druglist", xmlagg(xmlelement("Supplier",
xmlattributes(do.supplierName.SupplyCompany as "sid"), xmlagg(xmlelement("drug",
xmlattributes(do.drugcode.DrugCode as "DrugID"), xmlelement("DrugName",
do.drugcode.DrugName), xmlelement( "Quantity", do.drugcode.DrugQty))))), version '1.0') as
result

From DrugOrder_t do

Group By do.supplierName.SupplyCompany;
```

Find drugs being given to patients:

```
Select xmlroot(xmlelement("DrugsAssigned",xmlagg(xmlelement("Patient",
xmlattributes(ph.Prescribe.pid.PatientID as "pid"), xmlelement("PatientName",
ph.Prescribe.pid.PatientName), xmlelement("Prescription",ph.dcode.DrugName), xmlelement
("DrugCode",ph.dcode.DrugCode)))),version '1.0')

from PrescriptionHandler_t ph

group by ph.Prescribe.pid.PatientID;
```

Find which kinds of drug code, drug name needs to be restocked with the patient name. The patient needs to take the drugs from the prescription.

```
select xmlroot(xmlelement ("prescriptionlist", xmlagg(xmlelement(

"patient", xmlattributes(ph.Prescribe.pid.PatientName as
"pname"),xmlagg(xmlelement("restockeddrug", xmlelement("drugcode",ph.dcode.DrugCode),
xmlelement("drugname", ph.dcode.DrugName))))),version '1.0') as result
```

from PrescriptionHandler_t ph

where ph.drugQuantity>ph.dcode.DrugQty

group by ph.Prescribe.pid.PatientName;

3. **Billing System – 3 Query**

Find the bill from Peter Parker with its insurance contact and insurance coverage.

select xmlroot(xmlelement("billlist", xmlagg(xmlelement("patientbill", xmlattributes(c.BillNo as "billno"),xmlagg(xmlelement("patient", xmlattributes(c.pid.PatientID as "id"), xmlelement("patientname", c.pid.PatientName), xmlelement("insurancecontact", c.pid.Insurid.InsuranceContact), xmlelement("insurancecoverage", round(c.getInsuranceReduce(),2)))))), version '1.0') as result

from Cash_Payment_t c

where c.pid.PatientName ='Peter Parker'

group by c.BillNo;

Find BillNo, patient names for card payment. Also show drug price and drug quantities for that patient bought.

select xmlroot(xmlelement("Billlist", xmlagg(xmlelement(

"bill", xmlattributes(cd.BillNo as "billno"),xmlagg(xmlelement("patient", xmlforest(ph.Prescribe.pid.PatientName as"pname", ph.Prescribe.pid.PatientID as "pid"),xmlelement("drugprice", ph.dcode.DrugPrice),xmlelement("drugquantites", ph.dcode.DrugQty)))))),version '1.0') as result

from PrescriptionHandler_t ph, Card_Payment_t cd

where cd.pid.PatientID = ph.Prescribe.pid.PatientID

group by cd.BillNo;

Find Maurice Loveheart’s prescriptionID and get change from cash payment.

select xmlroot(xmlelement("Prescriptionlist", xmlagg(xmlelement(

```
"patient", xmlattributes(cd.pid.PatientID as "pid"),
xmlagg(
xmlelement("patientname", xmlattributes(cd.pid.PatientName as "pname"),
xmlelement("prescriptionid",p.PrescriptionID),
xmlelement("cashchange",cd.getChange())
))))),
version '1.0') as result
from Prescription_t p, Cash_Payment_t cd
where p.pid.PatientName = 'Maurice Loveheart'
And p.pid.PatientID = cd.pid.PatientID
group by cd.pid.PatientID;
```

4. **Customer Personnel – 3 Queries**

Customer names that paid by card

```
select xmlroot(xmlelement("patientlist", xmlagg(xmlelement("patient",
xmlagg(xmlelement("patient", xmlattributes(ca.pid.PatientID as "pid"),
xmlelement("patientname", ca.pid.PatientName))))), version '1.0')
from Card_Payment_t ca
group by ca.pid.PatientID;
```

Find bills which are over 100 and get the prescription details.

```
select xmlroot(xmlelement("Prescriptionlist", xmlagg(xmlelement("patient",
xmlattributes(cd.pid.PatientID as "pid"), xmlagg( xmlelement("patientname",
xmlattributes(cd.pid.PatientName as "pname"),
xmlelement("prescriptionid",p.PrescriptionID),
xmlelement("cashchange",cd.getChange()))))))), version '1.0') as result
from Prescription_t p, Cash_Payment_t cd
where cd.TotalPrice>100 And p.pid.PatientID = cd.pid.PatientID
group by cd.pid.PatientID;
```

Find all patient and their insurance name.

```
select xmlroot(xmlelement("patientlist", xmlagg(xmlelement("patient",
xmlagg(xmlelement("patientaccount", xmlattributes(pt.AccountID as "aid"),
xmlelement("patientname", pt.pid.PatientName), xmlelement("insurancename",
pt.pid.Insurid.InsuranceName))))), version '1.0')

from Patient_Account_t pt

group by pt.pid.Insurid.InsuranceID;
```

5. **Inventory System – 3 Queries**

Find all drug name, their supply name and which have stock in Medicine_Stock.

```
Select xmlroot(xmlelement("druglist", xmlagg(xmlelement("Drug",

xmlagg(

xmlelement("Drugid", xmlattributes(ph.dcode.DrugCode as "ID"),

xmlelement("Name", ph.dcode.DrugName),

xmlelement("Supplier", d.supplierName.SupplyCompany))))),

version '1.0')

From PrescriptionHandler_t ph, DrugOrder_t d

Where ph.dcode.DrugQty > 0

And ph.dcode.DrugCode = d.drugcode.DrugCode

Group by ph.dcode.DrugCode;
```

Find the supplier company and suppliers' address for the drug order that drug order price over 5000.

```
select xmlroot(xmlelement("orderlist", xmlagg(xmlelement("order",

xmlagg(xmlelement("orderno", xmlattributes(d.OrderNo as

"orderno"),xmlelement("supplier", xmlforest(d.supplierName.SupplyCompany as
"suppliername", d.supplierName.Address as "address"),xmlelement("price",
d.Price))))),version '1.0')
```

```
from DrugOrder_t d

where d.Price>5000

group by d.OrderNo;
```

Find the supplier company is Sanofi Pasteur and which patient buy drugs from the supplier and the patient gender.

```
select xmlroot(xmlelement("orderlist", xmlagg(xmlelement("order", xmlattributes(d.OrderNo as "orderno"),xmlagg(xmlelement("patient", xmlforest(ph.Prescribe.pid.PatientName as "pname", ph.Prescribe.pid.Patient_Gender as "gender"),xmlelement("supplyname",d.supplierName.SupplyCompany)))))),version '1.0') as result

from PrescriptionHandler_t ph, DrugOrder_t d

where d.supplierName.SupplyCompany = 'Sanofi Pasteur'

And ph.dcode.DrugCode = d.drugcode.DrugCode

group by d.OrderNo;
```

NUMBER THREE

1. **Pharmacist**

Find prescribed drug, amount of refills and the price of the drug for all patients

```
OracleXML getXML -user "grp7/here4grp7" \

-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \

-rowsetTag "JohnCollins" -rowTag "drugPrescribed" \

"select ph.Prescribe.pid.PatientName as PatientName, ph.dcode.DrugName as DrugName, ph.Refills as Refills, ph.dcode.DrugPrice as price \

from PrescriptionHandler_t ph "
```

2. **Medicine Data**

Find drug code, stock quantity and supplier name for the medicine “Amoxicillin”

```
OracleXML getXML -user "grp7/here4grp7" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowsetTag "Amoxicillin" -rowTag "drugInfo" \
"select d.drugcode.DrugCode as code, d.drugcode.DrugName as Name,
d.drugcode.DrugQty as Quantity, d.supplierName.SupplyCompany as Supplier \
from DrugOrder_t d where d.drugcode.DrugName = 'Amoxicillin'"
```

3. **Billing System**

For all customers that paid by credit, show Patient name, the total amount the patient paid, the amount covered by insurance and insurance company

```
OracleXML getXML -user "grp7/here4grp7" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowsetTag "payment" -rowTag "credit" \
"select ca.pid.PatientName as Name, round(ca.calculatePatientPay(),2) as TotalPaid,
round(ca.getInsuranceReduce(),2) as InsurancePay, ca.pid.Insurid.InsuranceName as
InsuranceCompany \
from Card_Payment_t ca where ca.CardType ='Credit'"
```

4. **Customer Personnel**

Find all patient names with their Account numbers and InsurancePercentOff who are with PharmaAID insurance

```
OracleXML getXML -user "grp7/here4grp7" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowsetTag "PharmaAID" -rowTag "patient" \
"select pa.pid.PatientName as Name, pa.AccountID as accountNo,
pa.pid.Insurid.InsurancePercentOff as InsurancePercentage \
from Patient_Account_t pa where pa.pid.Insurid.InsuranceName = 'PharmaAID'"
```

5. **Inventory System**

Find all drugs with their current stock quantity, order quantity and supplier name

```
OracleXML getXML -user "grp7/here4grp7" \
-conn "jdbc:oracle:thin:@sit.itec.yorku.ca:1521:studb10g" \
-rowsetTag "Inventory" -rowTag "drug" \
"select d.drugcode.DrugName as DrugName, d.drugcode.DrugQty as Stock, d.OrderQty as OrderQyantity, d.supplierName.SupplyCompany as Supplier \
from DrugOrder_t d"
```

NUMBER FOUR

Create XML Resources in XML Repository

1. Creating a folder:

```
declare
ret boolean;
begin
ret
:=dbms_xdb.createfolder('/public/group7_2020');
commit;
end;
/
```

2. Creating a resource:

1. XML file: pharmacist.xml

```
declare
ret boolean;
begin
ret :=
```

```
dbms_xdb.createresource('/public/group7_2020/pharmacist.xml',
'<Prescriptionlist>
  <PatientPrescription presid="4522234">
    <pname pid="444332312">John Collins</pname>
    <homeaddress>82 Dove Street </homeaddress>
    <drug drugcode="3816452222">
      <drugname>Acetaminophen</drugname>
      <drugquantities>32</drugquantities>
      <refilldate>2020-02-28</refilldate>
      <refills>7</refills>
    </drug>
  </PatientPrescription>
  <PatientPrescription presid="1288634">
    <pname pid="75748333">Maurice Loveheart</pname>
    <homeaddress>75 Brampton Road </homeaddress>
    <drug drugcode="6354427262">
      <drugname>ibuprofen</drugname>
      <drugquantities>11</drugquantities>
      <refilldate>2020-02-14</refilldate>
      <refills>3</refills>
    </drug>
  </PatientPrescription>
  <PatientPrescription presid="7462653">
    <pname pid="343434323">Peter Parker</pname>
    <homeaddress>83 Humber Avenue </homeaddress>
    <drug drugcode="8783111133">
      <drugname>Amoxicillin</drugname>
      <drugquantities>15</drugquantities>
```

```
      <refilldate>2020-03-13</refilldate>
      <refills>10</refills>
    </drug>
  </PatientPrescription>
  <PatientPrescription presid="8383524">
    <pname pid="6838123233">Rick Astley</pname>
    <homeaddress>245 Eibenhower Crescent </homeaddress>
    <drug drugcode="6327272244">
      <drugname>Metformin</drugname>
      <drugquantities>12</drugquantities>
      <refilldate>2020-05-12</refilldate>
      <refills>0</refills>
    </drug>
  </PatientPrescription>
</Prescriptionlist>
  ');
commit;
end;
/
```

2. XML file: bill.xml

```
  declare
  ret boolean;
  begin
  ret :=
  dbms_xdb.createresource('/public/group7_2020/bill.xml',
'<billlist>
    <patientbill id="6838123233">
```

```
    <patientname>Rick Astley</patientname>
    <cashbill billno="110032020">
        <billdate>2020-03-20</billdate>
    <totalprice>343.45</totalprice>
    </cashbill>
    <cardbill billno="132324">
        <billdate>2020-03-10</billdate>
    <totalprice>233.45</totalprice>
    </cardbill>
</patientbill>
<patientbill id="343434323">
    <patientname>Peter Parker</patientname>
    <cashbill billno="18042020">
        <billdate>2020-04-18</billdate>
    <totalprice>30.45</totalprice>
    </cashbill>
    <cardbill billno="32651">
        <billdate>2020-02-20</billdate>
    <totalprice>56.22</totalprice>
    </cardbill>
</patientbill>
<patientbill id="75748333">
    <patientname>Maurice Loveheart</patientname>
    <cashbill billno="20052020">
        <billdate>2020-05-20</billdate>
        <totalprice>123.8</totalprice>
    </cashbill>
    <cardbill billno="573778">
```

```
        <billdate>2020-04-27</billdate>
      <totalprice>122.13</totalprice>
      </cardbill>
   </patientbill>
   <patientbill id="444332312">
      <patientname>John Collins</patientname>
      <cashbill billno="5032020">
         <billdate>2020-03-05</billdate>
         <totalprice>102.5</totalprice>
      </cashbill>
      <cardbill billno="234555">
         <billdate>2020-01-01</billdate>
      <totalprice>99.52</totalprice>
      </cardbill>
   </patientbill>
 </billlist>
 ');
 commit;
 end;
 /
```

3. XML file: insurance.xml

```
declare
 ret boolean;
 begin
 ret :=
 dbms_xdb.createresource('/public/group7_2020/insurance.xml',
'<insurancelist>
```

```
    <insurance insurid="3847343255">
        <insurancecompany>PharmaAID</insurancecompany>
        <insuranceoff>32</insuranceoff>
        <contactnumber>6477056399</contactnumber>
        <pname pid="6838123233">Rick Astley</pname>
    </insurance>
    <insurance insurid="43432434">
        <insurancecompany>Rexall+</insurancecompany>
        <insuranceoff>43</insuranceoff>
        <contactnumber>4163563116</contactnumber>
        <pname pid="444332312">John Collins</pname>
    </insurance>
    <insurance insurid="46556555">
        <insurancecompany>Humana</insurancecompany>
        <insuranceoff>54</insuranceoff>
        <contactnumber>4165551232</contactnumber>
        <pname pid="75748333">Maurice Loveheart</pname>
    </insurance>
    <insurance insurid="3321134">
        <insurancecompany>Anthem</insurancecompany>
        <insuranceoff>32</insuranceoff>
        <contactnumber>6473552192</contactnumber>
        <pname pid="343434323">Peter Parker</pname>
    </insurance>
</insurancelist>
  ');
  commit;
  end;
```

/

4. XML file: MedicineData.xml

```
declare
  ret boolean;
  begin
  ret :=
  dbms_xdb.createresource('/public/group7_2020/medicinedata.xml',
'<druglist>
        <drugDetails>
                <drugCode dcode="6327272244" >
                        <drugName>Metformin</drugName>
                        <stockQty>13</stockQty>
                        <price>1323.32</price>
                        <supplyCompany>Pfizer Canada</supplyCompany>
                </drugCode>
        </drugDetails>
        <drugDetails>
                <drugCode dcode="3816452222" >
                        <drugName>Acetaminophen</drugName>
                        <stockQty>25</stockQty>
                        <price>237.98</price>
                        <supplyCompany>Sanofi Pasteur</supplyCompany>
                </drugCode>
        </drugDetails>
        <drugDetails>
                <drugCode dcode="8783111133" >
                        <drugName>Amoxicillin</drugName>
```

```
                    <stockQty>52</stockQty>
                    <price>373.43</price>
                    <supplyCompany>Apotex</supplyCompany>
                </drugCode>
        </drugDetails>
        <drugDetails>
                <drugCode dcode="6354427262" >
                    <drugName>ibuprofen</drugName>
                    <stockQty>38</stockQty>
                    <price>572.43</price>
                    <supplyCompany>Sanofi Pasteur</supplyCompany>
                </drugCode>
        </drugDetails>
</druglist>
  ');
  commit;
  end;
  /
```

5.. XML file: inventory.xml

```
declare
  ret boolean;
  begin
  ret :=
  dbms_xdb.createresource('/public/group7_2020/inventory.xml',
'<pharmInventory>
                <SupplyCompany cid = "Sanofi Pasteur">
                   <drugDetails>
```

```
            <DrugName>Acetaminophen</DrugName>
            <DrugQty>25</DrugQty>
        </drugDetails>
        <orderDetails orderNo = "5565543" >
            <OrderQty>5</OrderQty>
            <Price>5303</Price>
        </orderDetails>
    </SupplyCompany>
    <SupplyCompany cid = "Apotex" >
        <drugDetails>
            <DrugName>Amoxicillin</DrugName>
            <DrugQty>52</DrugQty>
        </drugDetails>
        <orderDetails orderNo = "8725384" >
            <OrderQty>15</OrderQty>
            <Price>6200</Price>
        </orderDetails>

    </SupplyCompany>
    <SupplyCompany cid = "Sanofi Pasteur" >
        <drugDetails>
            <DrugName>Ibuprofen</DrugName>
            <DrugQty>0</DrugQty>
        </drugDetails>
        <orderDetails orderNo = "0000000" >
            <OrderQty>0</OrderQty>
            <Price>0</Price>
        </orderDetails>
```

```
        </SupplyCompany>
        <SupplyCompany cid = "Pfizer Canada" >
          <drugDetails>
              <DrugName>Metformin</DrugName>
              <DrugQty>13</DrugQty>
          </drugDetails>
          <orderDetails orderNo = "4322347" >
              <OrderQty>23</OrderQty>
              <Price>800</Price>
          </orderDetails>

        </SupplyCompany>
</pharmInventory>
');
 commit;
 end;
 /
```

NUMBER FIVE-XQUERY

1. **Pharmacist**

Find customer name for patient who have more than five refills.

```
  xquery
  let $c:=doc("/public/group7_2020/pharmacist.xml")
  for $d in $c/Prescriptionlist/PatientPrescription
  where $d/drug/refills > 5
  return $d/pname
  /
```

```
 xquery
    let $c:=doc("/public/group7_2020/pharmacist.xml")
    for $d in $c/Prescriptionlist/PatientPrescription
    where $d/drug/refills > 5
    return $d/pname
  6      /

Result Sequence
--------------------------------------------------------------------------------
<pname pid="444332312">John Collins</pname>
<pname pid="343434323">Peter Parker</pname>
```

Find drug code that customer bought and the drug quantities less than 15.

xquery

let $c:=doc("/public/group7_2020/pharmacist.xml")

for $d in $c/Prescriptionlist/PatientPrescription/drug[drugquantities < 15]

return $d/@drugcode

/

```
SQL> xquery
  2  let $c:=doc("/public/group7_2020/pharmacist.xml")
  3  for $d in $c/Prescriptionlist/PatientPrescription/drug[drugquantities < 15 ]
  4  return $d/@drugcode
  5  /

Result Sequence
--------------------------------------------------------------------------------
6354427262
6327272244
```

Find total price for customer who bought drug ibuprofen and paid by card.

xquery

let $c:=doc("/public/group7_2020/pharmacist.xml")

for $m in $c/Prescriptionlist/PatientPrescription

let $s:=doc("/public/group7_2020/bill.xml")

for $e in $s/billlist/patientbill

where $m/pname/@pid = $e/@id

and $m/drug/drugname ='ibuprofen'

return $e/cardbill/totalprice

/

```
SQL> xquery
  2   let $c:=doc("/public/group7_2020/pharmacist.xml")
  3  for $m in $c/Prescriptionlist/PatientPrescription
  4  let $s:=doc("/public/group7_2020/bill.xml")
  5  for $e in $s/billlist/patientbill
  6  where $m/pname/@pid = $e/@id and $m/drug/drugname ='ibuprofen'
  7  return $e/cardbill/totalprice
  8  /

Result Sequence
--------------------------------------------------------------------------------
<totalprice>122.13</totalprice>
```

2. **Medicine Data**

Get price for drugcode "3816452222"

xquery

let $c:=doc("/public/group7_2020/medicinedata.xml")

for $d in $c/druglist/drugDetails

where $d/drugCode/@dcode='3816452222'

return $d/drugCode/price

/

```
xquery
let $c:=doc("/public/group7_2020/medicinedata.xml")
for $d in $c/druglist/drugDetails
where $d/drugCode/@dcode='3816452222'
return $d/drugCode/price
  6  /

Result Sequence
------------------------------------------------------------
<price> 237.98 </price>
```

For stock quantity below 20, show corresponding prescription number.

xquery

let $c:=doc("/public/group7_2020/medicinedata.xml")

for $d in $c/druglist/drugDetails/drugCode

let $e:=doc("/public/group7_2020/pharmacist.xml")

for $m in $e/Prescriptionlist/PatientPrescription

where $d/@dcode= $m/drug/@drugcode

and $d/stockQty < 20

return $m/@presid

/

```
SQL> xquery
let $c:=doc("/public/group7_2020/medicinedata.xml")
for $d in $c/druglist/drugDetails/drugCode
let $e:=doc("/public/group7_2020/pharmacist.xml")
for $m in $e/Prescriptionlist/PatientPrescription
where $d/@dcode= $m/drug/@drugcode
and $d/stockQty < 20
return $m/@presid
  9  /

Result Sequence
------------------------------------------------------------
8383524
```

For drug stock lower than 50 units, show drug code, drug price and stock quantity listed from low to high

xquery

let $c:=doc("/public/group7_2020/medicinedata.xml")

for $d in $c/druglist/drugDetails

where $d/drugCode/stockQty < 50

order by xs:decimal($d/drugCode/stockQty)

return <drugstock drugcode="{$d/drugCode/@dcode}">

<stock>{$d/drugCode/stockQty/text()}</stock>

{$d/drugCode/price}

</drugstock>

/

```
SQL>
xquery
let $c:=doc("/public/group7_2020/medicinedata.xml")
for $d in $c/druglist/drugDetails
where $d/drugCode/stockQty < 50
order by xs:decimal($d/drugCode/stockQty)
return <drugstock drugcode="{$d/drugCode/@dcode}">
<stock>{$d/drugCode/stockQty/text()}</stock>
{$d/drugCode/price}
</drugstock>
 10  /

Result Sequence
--------------------------------------------------------------------------------
<drugstock drugcode="6327272244"><stock>13</stock><price>1323.32</price></drugst
ock>

<drugstock drugcode="3816452222"><stock>25</stock><price>237.98</price></drugsto
ck>

<drugstock drugcode="6354427262"><stock>38</stock><price>572.43</price></drugsto
ck>
```

3. Billing System

Find cash bill number, bill date and total price is over 100.

```
xquery
let $c:=doc("/public/group7_2020/bill.xml")
for $d in $c/billlist/patientbill
where $d/cashbill/totalprice > 100
return <cashbill billno = "{$d/cashbill/@billno}">
<date> {$d/cashbill/billdate/text()}</date>
<totalprice> {$d/cashbill/totalprice/text()}</totalprice>
</cashbill>
/
```

```
    xquery
    let $c:=doc("/public/group7_2020/bill.xml")
    for $d in $c/billlist/patientbill
    where $d/cashbill/totalprice > 100
return <cashbill billno = "{$d/cashbill/@billno}">
<date> {$d/cashbill/billdate/text()}</date>
<totalprice> {$d/cashbill/totalprice/text()}</totalprice>
</cashbill>
    /



Result Sequence
------------------------------------------------------------------------------
<cashbill billno="110032020"><date>2020-03-20</date><totalprice>343.45</totalpri
ce></cashbill>

<cashbill billno="20052020"><date>2020-05-20</date><totalprice>123.8</totalprice
></cashbill>

<cashbill billno="5032020"><date>2020-03-05</date><totalprice>102.5</totalprice>
</cashbill>
```

Find card bill date and home address for patient John Collins

```
xquery
let $c :=doc("/public/group7_2020/bill.xml")
for $m in $c/billlist/patientbill
let $s :=doc("/public/group7_2020/pharmacist.xml")
for $e in $s/Prescriptionlist/PatientPrescription
where $m/@id = $e/pname/@pid
and $m/patientname = 'John Collins'
return <cardbill billno = "{$m/cardbill/@billno}">
        <date>{$m/cashbill/billdate/text()} </date>
        <address>{$e/homeaddress/text()}</address>
</cardbill>
/
```

```
xquery
    let $c :=doc("/public/group7_2020/bill.xml")
for $m in $c/billlist/patientbill
    let $s :=doc("/public/group7_2020/pharmacist.xml")
for $e in $s/Prescriptionlist/PatientPrescription
where $m/@id = $e/pname/@pid
and $m/patientname = 'John Collins'
return <cardbill billno = "{$m/cardbill/@billno}">
<date>{$m/cashbill/billdate/text()} </date>
<address>{$e/homeaddress/text()}</address>
</cardbill>
 12  /

Result Sequence
--------------------------------------------------------------------------------
<cardbill billno="234555"><date>2020-03-05</date><address>82 Dove Street </addre
ss></cardbill>
```

Find cash bill number for date is 2020-04-18.

xquery

let $c:=doc("/public/group7_2020/bill.xml")

for $m in $c/billlist/patientbill

where $m/cashbill/billdate = '2020-04-18'

return $m/cashbill/@billno

/

```
xquery
    let $c:=doc("/public/group7_2020/bill.xml")
for $m in $c/billlist/patientbill
where $m/cashbill/billdate =  '2020-04-18'
    return $m/cashbill/@billno
/


Result Sequence
--------------------------------------------------------------------------------
18042020
```

4. **Customer Personnel**

Find insurance company names and coverage rate for patient who have all bill payment over 100. (use insurance.xml and bill.xml)

Xquery

let $c:=doc("/public/group7_2020/bill.xml")

```
for $m in $c/billlist/patientbill
let $s:=doc("/public/group7_2020/insurance.xml")
for $e in $s/insurancelist/insurance
where $m/@id = $e/pname/@pid and
$m/cashbill/totalprice > 100 and $m/cardbill/totalprice > 100
return <insuranceinfo patientname = "{$e/pname}">
<provider>{$e/insurancecompany/text()}</provider>
<insuranceoff>{$e/insuranceoff/text()}</insuranceoff>
</insuranceinfo>
/
```

```
SQL> xquery
let $c:=doc("/public/group7_2020/bill.xml")
for $m in $c/billlist/patientbill
let $s:=doc("/public/group7_2020/insurance.xml")
for $e in $s/insurancelist/insurance
where $m/@id = $e/pname/@pid and
$m/cashbill/totalprice > 100 and $m/cardbill/totalprice > 100
return <insuranceinfo patientname = "{$e/pname}">
<provider>{$e/insurancecompany/text()}</provider>
<insuranceoff>{$e/insuranceoff/text()}</insuranceoff>
</insuranceinfo>
 12  /

Result Sequence
--------------------------------------------------------------------------------
<insuranceinfo patientname="Rick Astley"><provider>PharmaAID</provider><insuranc
eoff>32</insuranceoff></insuranceinfo>

<insuranceinfo patientname="Maurice Loveheart"><provider>Humana</provider><insur
anceoff>54</insuranceoff></insuranceinfo>
```

Find patient ids with insurance company "Humana" (result without tag)

```
xquery
let $c:=doc("/public/group7_2020/insurance.xml")
for $m in $c/insurancelist/insurance
where $m/insurancecompany = 'Humana'
return $m/pname/@pid
/
```

```
xquery
let $c:=doc("/public/group7_2020/insurance.xml")
for $m in $c/insurancelist/insurance
where $m/insurancecompany = 'Humana'
return $m/pname/@pid
  6  /

Result Sequence
--------------------------------------------------------------------------------
75748333
```

Find insurance id, all bill number and amount for patient id “343434323” (use insurance.xml and bill.xml)

xquery

let $c:=doc("/public/group7_2020/bill.xml")

for $m in $c/billlist/patientbill

let $s:=doc("/public/group7_2020/insurance.xml")

for $e in $s/insurancelist/insurance

where $m/@id = $e/pname/@pid and

$e/pname/@pid = '343434323'

return <billinfo patientname = "{$e/pname}">

<insuranceid>{$e/@insurid}

<cashbill>{$m/cashbill/@billno}

<amount>{$m/cashbill/totalprice/text()}</amount></cashbill>

<cardbill>{$m/cardbill/@billno}

<amount>{$m/cardbill/totalprice/text()}</amount></cardbill>

</insuranceid>

</billinfo>

/

```
SQL> xquery
let $c:=doc("/public/group7_2020/bill.xml")
for $m in $c/billlist/patientbill
let $s:=doc("/public/group7_2020/insurance.xml")
for $e in $s/insurancelist/insurance
where $m/@id = $e/pname/@pid and
$e/pname/@pid = '343434323'
return <billinfo patientname = "{$e/pname}">
<insuranceid>{$e/@insurid}
<cashbill>{$m/cashbill/@billno}
<amount>{$m/cashbill/totalprice/text()}</amount></cashbill>
<cardbill>{$m/cardbill/@billno}
<amount>{$m/cardbill/totalprice/text()}</amount></cardbill>
</insuranceid>
</billinfo>
 16  /

Result Sequence
--------------------------------------------------------------------------------
<billinfo patientname="Peter Parker"><insuranceid insurid="3321134"><cashbill bi
llno="18042020"><amount>30.45</amount></cashbill><cardbill billno="32651"><amoun
t>56.22</amount></cardbill></insuranceid></billinfo>
```

For patient id "6838123233" show insurance company contact number

xquery

let $c:=doc("/public/group7_2020/insurance.xml")

for $d in $c/insurancelist/insurance

where $d/pname/@pid ='6838123233'

return $d/contactnumber/text()

/

```
xquery
let $c:=doc("/public/group7_2020/insurance.xml")
for $d in $c/insurancelist/insurance
where $d/pname/@pid ='6838123233'
return $d/contactnumber/text()
  6  /

Result Sequence
----------------------------------------------------
6477056399
```

5. **Inventory System**

Find Drugs with quantity over 10

xquery

let $c:=doc("/public/group7_2020/inventory.xml")

for $d in $c/pharmInventory/SupplyCompany/drugDetails

where $d/DrugQty > 10

return $d/DrugName/text()

/

```
SQL>
xquery
    let $c:=doc("/public/group7_2020/inventory.xml")
    for $d in $c/pharmInventory/SupplyCompany/drugDetails
    where $d/DrugQty > 10
    return $d/DrugName/text()
  6      /

Result Sequence
--------------------------------------------------------------------------------
Acetaminophen
Amoxicillin
Metformin
```

Find All drugs with price under 6000, print all name, drugqty and price, group low to high price

xquery

let $c:=doc("/public/group7_2020/inventory.xml")

for $d in $c/pharmInventory/SupplyCompany

where $d/orderDetails/Price < 6000

order by xs:decimal($d/orderDetails/Price)

return

<drug>

<name>{$d/drugDetails/DrugName/text()}</name>

<drugqty>{$d/drugDetails/DrugQty/text()}</drugqty>

<price>{$d/orderDetails/Price/text()}</price>

</drug>

/

```
SQL> xquery
let $c:=doc("/public/group7_2020/inventory.xml")
for $d in $c/pharmInventory/SupplyCompany
where $d/orderDetails/Price < 6000
order by xs:decimal($d/orderDetails/Price)
return
<drug>
<name>{$d/drugDetails/DrugName/text()}</name>
<drugqty>{$d/drugDetails/DrugQty/text()}</drugqty>
<price>{$d/orderDetails/Price/text()}</price>
</drug>
 12  /

Result Sequence
--------------------------------------------------------------------------------
<drug><name>Ibuprofen</name><drugqty>0</drugqty><price>0</price></drug>
<drug><name>Metformin</name><drugqty>13</drugqty><price>800</price></drug>
<drug><name>Acetaminophen</name><drugqty>25</drugqty><price>5303</price></drug>
```

Find Drug details for cid = Supply company Pfizer along with its corresponding drugcode

xquery

```
let $c:=doc("/public/group7_2020/inventory.xml")
for $d in $c/pharmInventory/SupplyCompany
let $e:=doc("/public/group7_2020/medicinedata.xml")
for $f in $e/druglist/drugDetails
where $d/@cid = $f/drugCode/supplyCompany and $d/@cid= 'Pfizer Canada'
return
<orderdetails orderno = "{$d/orderDetails/@orderNo}" >
<orderqty>{$d/orderDetails/OrderQty}</orderqty>
<price>{$d/orderDetails/Price/text()}</price>
<quantity>{$d/drugDetails/DrugQty/text()}</quantity>
<drug drugCode = "{$f/drugCode/@dcode}" >{$f/drugCode/drugName/text()}</drug>
</orderdetails>
/
```

```
xquery
let $c:=doc("/public/group7_2020/inventory.xml")
for $d in $c/pharmInventory/SupplyCompany
let $e:=doc("/public/group7_2020/medicinedata.xml")
for $f in $e/druglist/drugDetails
where $d/@cid = $f/drugCode/supplyCompany and $d/@cid= 'Pfizer Canada'
return
<orderdetails orderno = "{$d/orderDetails/@orderNo}" >
<orderqty>{$d/orderDetails/OrderQty}</orderqty>
<price>{$d/orderDetails/Price/text()}</price>
<quantity>{$d/drugDetails/DrugQty/text()}</quantity>
<drug drugCode = "{$f/drugCode/@dcode}" >{$f/drugCode/drugName/text()}</drug>
</orderdetails>
 14  /

Result Sequence
--------------------------------------------------------------------------------
<orderdetails orderno="4322347"><orderqty><OrderQty>23</OrderQty></orderqty><pri
ce>800</price><quantity>13</quantity><drug drugCode="6327272244">Metformin</drug
></orderdetails>
```