**Bel Shifa**

**(Medicine Ordering application)**

TA: Radwa Moustafa

Team ID: 37

|  |  |  |
| --- | --- | --- |
| **Name** | **Seat numbers** | **Department** |
| **كريمه صبحي ابو العلا عبد الخالق** | **20201701115** | **IS** |
| **علي ماهر طلعت السيد** | **20201700506** | **IS** |
| **علي عبد الله محمد عبد الله** | **20201700504** | **IS** |
| **عمر أحمد محمود النصيري** | **20201701110** | **IS** |
| **عبد الرحمن عبد الحليم العارف حسين** | **20201700442** | **IS** |
| **مصطفى ياسر السيد احمد** | **20201701143** | **IS** |

**--(****1)Tables**

**--table [1]**

CREATE TABLE Customer (

CustomerID number(10) PRIMARY KEY,

TrackOrderID number(10) ,

CustomerName VARCHAR2(25) NOT NULL,

CustomerEmail VARCHAR2(50) NOT NULL,

CustomerPassword VARCHAR2(50) NOT NULL

);

**--****table [2]**

CREATE TABLE Medicines (

MedicineID number(10) PRIMARY KEY,

MedicineName VARCHAR2(60) NOT NULL,

pharmacyid number ,

Description VARCHAR2(500),

Quantity number ,

image VARCHAR2(200),

Price DECIMAL(10,2) NOT NULL,

FOREIGN KEY (PharmacyID) REFERENCES Pharmacy(PharmacyID)

);

**--table [3]**

CREATE TABLE Pharmacy (

PharmacyID number(10) PRIMARY KEY,

PharmacyName VARCHAR2(50) NOT NULL,

Address VARCHAR2(100) NOT NULL,

PharmacyEmail VARCHAR2(50) NOT NULL,

PharmacyPassword VARCHAR2(50) NOT NULL,

phone\_number varchar2(25),

color varchar2(100)

);

**--table [4]**

CREATE TABLE Admin (

AdminID number(10) PRIMARY KEY,

PharmacyID number(10) NOT NULL,

AdminName VARCHAR2(30) NOT NULL,

AdminEmail VARCHAR2(50) NOT NULL,

AdminPassword VARCHAR2(50) NOT NULL,

FOREIGN KEY (PharmacyID) REFERENCES Pharmacy(PharmacyID)

);

**--table [5]**

CREATE TABLE Orders (

OrderID number(10) PRIMARY KEY,

CustomerID number(10) NOT NULL,

MedicineID number(10) NOT NULL,

PharmacyID number(10) NOT NULL,

Quantity number(10) NOT NULL,

Status VARCHAR2(20) ,

PaymentMethod VARCHAR2(100),

cost number(10) NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (MedicineID) REFERENCES Medicines(MedicineID),

FOREIGN KEY (PharmacyID) REFERENCES Pharmacy(PharmacyID)

);

**--table [6]**

CREATE TABLE Delivery (

DeliveryID number(10) primary key,

DeliveryEmail VARCHAR2(50) NOT NULL,

DeliveryPassword VARCHAR2(50) NOT NULL

);

**--(2) Procedures**

**--Procedures [1]**

create or replace

PROCEDURE ShowORDERS

(orders\_results OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN orders\_results FOR

SELECT orderid

FROM orders

WHERE status = 'pending';

END;

**--Procedures [2]**

create or replace

PROCEDURE CountCanceledOrders

(results out NUMBER)

AS

BEGIN

SELECT count(\*)

into results

FROM orders

WHERE status = 'canceled';

END;

**--Procedures [3]**

create or replace

PROCEDURE AuthenticateUser

(email in VARCHAR2,pass in VARCHAR2 , cemail out VARCHAR2, cpass out VARCHAR2)

AS

BEGIN

select CustomerEmail,CustomerPassword INTO cemail,cpass from Customer Where CustomerEmail = email and CustomerPassword =pass;

end;

**--Procedures [4]**

create or replace PROCEDURE get\_medicines

( pharmacy\_id IN NUMBER,medicine\_results OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN medicine\_results FOR

SELECT medicinename,quantity

FROM medicines

WHERE pharmacyid = pharmacy\_id;

END;

**--****Procedures[5]**

create or replace

PROCEDURE TrackOrder

(Order\_ID NUMBER,Customer\_Id NUMBER,state out VARCHAR2)

AS

id NUMBER;

BEGIN

-- Check if the customer exists

SELECT customerid into id FROM Customer WHERE CustomerId = Customer\_Id;

IF id is NULL then

RAISE\_APPLICATION\_ERROR('Customer with Id %d does not exist.',Customer\_Id);

end if;

SELECT Status into state FROM Orders WHERE OrderID = Order\_ID ;

END ;

**--Procedures [6]**

create or replace

PROCEDURE DeliverOrder

(Order\_ID NUMBER)

AS

BEGIN

UPDATE Orders SET Status = 'Delivered' WHERE OrderID = Order\_ID;

END;

**--Procedures [7]**

create or replace PROCEDURE AddMedicine

(Medicine\_Name VARCHAR2,description VARCHAR2,quantity NUMBER,price DECIMAL)

AS

med\_name VARCHAR2(60);

BEGIN

-- Check if the medicine exists

SELECT MedicineName INTO med\_name FROM Medicines WHERE MedicineName = Medicine\_Name;

IF med\_name is NULL then

RAISE\_APPLICATION\_ERROR(-20001,'this medicine is already exists.');

RETURN;

end if;

INSERT INTO Medicines (MedicineName, Description, Quantity, Price)

VALUES (Medicine\_Name, description, quantity, price);

END AddMedicine;

**--Procedures [8]**

create or replace PROCEDURE RemoveMedicine

(Medicine\_Id NUMBER)

AS

med\_id NUMBER;

BEGIN

-- Check if the medicine exists

SELECT MedicineID into med\_id FROM Medicines WHERE MedicineID = Medicine\_Id;

IF med\_id is NULL then

BEGIN

RAISE\_APPLICATION\_ERROR('Medicine with Id %d does not exist.',Medicine\_Id);

RETURN;

END;

end IF;

-- Delete the medicine from the inventory

DELETE FROM Medicines WHERE MedicineID = Medicine\_ID ;

DBMS\_OUTPUT.PUT\_LINE('Medicine has been removed from the inventory.');

END;

**--Procedures [9]**

create or replace PROCEDURE CheckAvailability

(med\_Name VARCHAR2)

AS

Available NUMBER(10);

BEGIN

SELECT quantity into Available FROM Medicines WHERE medicinename = med\_Name ;

IF Available = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('The medicine is available.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('The medicine is not available.');

END IF;

END;

**--Procedures [10]**

create or replace PROCEDURE AddPharmacy

(Pharmacy\_Name VARCHAR2,address VARCHAR2,Email VARCHAR2,Password VARCHAR2)

AS

P\_name VARCHAR2(50);

BEGIN

-- Check if the pharmacy exists

SELECT PharmacyName INTO P\_name FROM Pharmacy WHERE PharmacyName = Pharmacy\_Name;

IF P\_name is NULL then

RAISE\_APPLICATION\_ERROR(-20001,'this Pharmacy is already exists.');

RETURN;

end if;

INSERT INTO Pharmacy (PharmacyName, Address, PharmacyEmail,PharmacyPassword)

VALUES (Pharmacy\_Name,Address, Email,Password) ;

DBMS\_OUTPUT.PUT\_LINE('Pharmacy added successfully.') ;

END ;

**--Procedures [11]**

create or replace PROCEDURE DeletePharmacy

(Pharmacy\_Name VARCHAR2)

AS

p\_id NUMBER;

BEGIN

-- Check if the pharmacy exists

SELECT pharmacyid into p\_id FROM Pharmacy WHERE PharmacyName = Pharmacy\_Name ;

IF p\_id is NULL then

RAISE\_APPLICATION\_ERROR(-20001,'Pharmacy does not exist.');

RETURN;

end IF ;

DELETE FROM pharmacy WHERE pharmacyname = Pharmacy\_Name;

DBMS\_OUTPUT.PUT\_LINE('Pharmacy have been deleted successfully.');

END;

**--Procedures [12]**

create or replace

PROCEDURE CancelRequest

(Order\_ID IN NUMBER)

AS

BEGIN

UPDATE Orders

SET Status = 'Cancelled'

WHERE OrderID = Order\_ID;

END;

**--Procedures [13]**

create or replace

PROCEDURE ChoosePaymentMethod

(Order\_ID NUMBER,Payment\_Method VARCHAR2)

AS

BEGIN

UPDATE Orders

SET PaymentMethod = Payment\_Method

WHERE OrderID = Order\_ID;

END;