Magazine Management System

Assigned TA: Marwa Helaly

Team ID: 71

Team Members:

* سندس احمد عبدالله محمد 2021170240 IS
* سهيلة محمد احمد زكي عبدالهادي 2021170247 IS
* سارة عبد الله عباس السيد 2021170226 IS
* آية محمد فكري محمود 2021170110 IS
* إسراء حسن حسين حسن 2021170067 IS
* منار خالد عبدالجواد عبدالسلام 2021170542 IS
* علياء شريف رمضان الشريف 2020170514 CS

CREATE TABLE Guest (

UserID INT PRIMARY KEY,

Username VARCHAR2(50) NOT NULL,

Email VARCHAR2(100) UNIQUE NOT NULL,

Password VARCHAR2(100) NOT NULL

);

CREATE TABLE AdminPage (

UserID INT PRIMARY KEY,

Username VARCHAR2(50) NOT NULL,

Email VARCHAR2(100) UNIQUE NOT NULL,

Password VARCHAR2(100) NOT NULL

);

CREATE TABLE Publisher (

UserID INT PRIMARY KEY,

Username VARCHAR2(50) NOT NULL,

Email VARCHAR2(100) UNIQUE NOT NULL,

Password VARCHAR2(100) NOT NULL

);

CREATE TABLE Magazines (

MagazineID INT PRIMARY KEY,

PublisherID INT,

Title VARCHAR(100) NOT NULL,

Content CLOB, -- Using CLOB for large text content

Published NUMBER(1) DEFAULT 0,

FOREIGN KEY (PublisherID) REFERENCES Publisher(UserID)-- To track publication status (0 for false, 1 for true)

);

CREATE TABLE Emails (

EmailID INT PRIMARY KEY,

PublisherID INT,

AdminID INT,

MagazineID INT,

Content NUMBER(1) DEFAULT 0,

Published NUMBER(1) DEFAULT 0,

FOREIGN KEY (PublisherID) REFERENCES Publisher(UserID),

FOREIGN KEY (AdminID) REFERENCES AdminPage(UserID),

FOREIGN KEY (MagazineID) REFERENCES Magazines(MagazineID)

);

CREATE TABLE Ratings (

RatingID INT PRIMARY KEY,

UserID INT,

MagazineID INT,

Rating INT,

Review VARCHAR2(1000), -- Adjust the size as needed

FOREIGN KEY (UserID) REFERENCES Guest(UserID), -- corrected table name to Guests

FOREIGN KEY (MagazineID) REFERENCES Magazines(MagazineID)

);

INSERT INTO Guest (UserID, Username, Email, Password)

SELECT 1, 'guest\_user1', 'guest1@example.com', 'guestpassword1' FROM DUAL

UNION ALL

SELECT 2, 'guest\_user2', 'guest2@example.com', 'guestpassword2' FROM DUAL

UNION ALL

SELECT 3, 'guest\_user3', 'guest3@example.com', 'guestpassword3' FROM DUAL

UNION ALL

SELECT 4, 'guest\_user4', 'guest4@example.com', 'guestpassword4' FROM DUAL;

INSERT INTO AdminPage (UserID, Username, Email, Password)

SELECT 1, 'admin\_user1', 'admin1@example.com', 'adminpassword1' FROM DUAL

UNION ALL

SELECT 2, 'admin\_user2', 'admin2@example.com', 'adminpassword2' FROM DUAL

UNION ALL

SELECT 3, 'admin\_user3', 'admin3@example.com', 'adminpassword3' FROM DUAL

UNION ALL

SELECT 4, 'admin\_user4', 'admin4@example.com', 'adminpassword4' FROM DUAL;

INSERT INTO Publisher (UserID, Username, Email, Password)

SELECT 1, 'publisher\_user1', 'publisher1@example.com', 'publisherpassword1' FROM DUAL

UNION ALL

SELECT 2, 'publisher\_user2', 'publisher2@example.com', 'publisherpassword2' FROM DUAL

UNION ALL

SELECT 3, 'publisher\_user3', 'publisher3@example.com', 'publisherpassword3' FROM DUAL

UNION ALL

SELECT 4, 'publisher\_user4', 'publisher4@example.com', 'publisherpassword4' FROM DUAL;

INSERT INTO Magazines (MagazineID, PublisherID, Title, Content, Published)

SELECT 1, 1, 'Magazine 1', 'Content of Magazine 1', 0 FROM DUAL

UNION ALL

SELECT 2, 2, 'Magazine 2', 'Content of Magazine 2', 0 FROM DUAL

UNION ALL

SELECT 3, 3, 'Magazine 3', 'Content of Magazine 3', 0 FROM DUAL

UNION ALL

SELECT 4, 3, 'Magazine 4', 'Content of Magazine 4', 0 FROM DUAL;

INSERT INTO Ratings (RatingID, UserID, MagazineID, Rating, Review)

SELECT 1, 1, 1, 5, 'Great magazine!' FROM DUAL

UNION ALL

SELECT 2, 2, 2, 4, 'Interesting content.' FROM DUAL

UNION ALL

SELECT 3, 3, 3, 3, 'Good content.' FROM DUAL

UNION ALL

SELECT 4, 4, 4, 1, 'Bad content.' FROM DUAL;

INSERT INTO Emails (EmailID, PublisherID, AdminID, MagazineID, Content, Published)

VALUES

(1, 1, NULL, 1, 1, 1),

(2, 2, NULL, 2, 0, 1), -- Assuming 0 for false

(3, NULL, 3, 3, 1, 1),

(4, NULL, 4, 4, 0, 1); -- Assuming 0 for false

create or replace procedure GettingMagazinesID

(PID in varchar2 , CID out sys\_refcursor)

as

begin

open CID for

select M.MagazineID

from Magazines M , publisher P

where M.PublisherID = P.UserID

and P.UserID = PID;

end;

create or replace procedure GetUserID (UseID out number)

as

begin

select max(UserID)

into UseID

from Publisher;

end;

create or replace procedure GetPublisherID (PID out number)

as

begin

select max(PublisherID)

into PID

from Magazines;

end;