Solution of task 9

--Users Table

CREATE TABLE Users ( UserID INT PRIMARY KEY IDENTITY(1,1), Username NVARCHAR(50) NOT NULL, Email NVARCHAR(100), RegistrationDate DATETIME NOT NULL DEFAULT(GETDATE()) );

--Posts Table

CREATE TABLE Posts ( PostID INT PRIMARY KEY IDENTITY(1,1), UserID INT FOREIGN KEY REFERENCES Users(UserID), Title NVARCHAR(100) NOT NULL, Content NVARCHAR(MAX) NOT NULL, PostDate DATETIME NOT NULL DEFAULT(GETDATE()) );

--Comments Table

CREATE TABLE Comments ( CommentID INT PRIMARY KEY IDENTITY(1,1), PostID INT FOREIGN KEY REFERENCES Posts(PostID), UserID INT FOREIGN KEY REFERENCES Users(UserID), CommentText NVARCHAR(MAX) NOT NULL, CommentDate DATETIME NOT NULL DEFAULT(GETDATE()) );

--Step 1: Insert Data into Users Table

INSERT INTO Users (Username, Email, RegistrationDate) VALUES ('JohnDoe', 'johndoe@example.com', '2022-01-10'), ('JaneSmith', 'janesmith@example.com', '2022-01-15'), ('AliceJones', 'alicejones@example.com', '2022-01-20'), ('BobBrown', 'bobbrown@example.com', '2022-01-25'), ('CharlieDavis', 'charliedavis@example.com', '2022-02-01'), ('DianaEvans', 'dianaevans@example.com', '2022-02-05'), ('EvanFoster', 'evanfoster@example.com', '2022-02-10'), ('FionaGreen', 'fionagreen@example.com', '2022-02-15'), ('GeorgeHill', 'georgehill@example.com', '2022-02-20'), ('HannahIvy', 'hannahivy@example.com', '2022-02-25');

--Step 2: Insert Data into Posts Table

INSERT INTO Posts (UserID, Title, Content, PostDate) VALUES (1, 'First Post', 'This is the content of the first post.', '2022-02-26'), (2, 'Second Post', 'This is the content of the second post.', '2022-03-01'), (1, 'Third Post', 'This is the content of the third post.', '2022-03-05'), (3, 'Fourth Post', 'This is the content of the fourth post.', '2022-03-10'), (4, 'Fifth Post', 'This is the content of the fifth post.', '2022-03-15'), (5, 'Sixth Post', 'This is the content of the sixth post.', '2022-03-20'), (2, 'Seventh Post', 'This is the content of the seventh post.', '2022-03-25'), (3, 'Eighth Post', 'This is the content of the eighth post.', '2022-03-30'), (4, 'Ninth Post', 'This is the content of the ninth post.', '2022-04-04'), (5, 'Tenth Post', 'This is the content of the tenth post.', '2022-04-08');

--Step 3: Insert Data into Comments Table

INSERT INTO Comments (PostID, UserID, CommentText, CommentDate) VALUES (1, 2, 'Great first post!', '2022-02-27'), (1, 3, 'Looking forward to more!', '2022-02-28'), (2, 1, 'Very interesting read.', '2022-03-02'), (3, 4, 'I agree with your points.', '2022-03-06'), (4, 5, 'This topic is very relevant.', '2022-03-11'), (5, 1, 'Thanks for sharing.', '2022-03-16'), (6, 2, 'I learned a lot from this post.', '2022-03-21'), (7, 3, 'Cannot wait for your next post!', '2022-03-26'), (8, 4, 'Very detailed and informative.', '2022-03-31'), (9, 5, 'Your writing style is engaging.', '2022-04-05');

--Basic Join Queries

--Join Users and Posts: Get all posts with their corresponding user names.

SELECT Users.Username, Posts.Title, Posts.Content

FROM Posts

INNER JOIN Users ON Posts.UserID = Users.UserID;

--Join Users and Comments: Find all comments made by each user.

SELECT Users.Username, Comments.CommentText

FROM Comments

INNER JOIN Users ON Comments.UserID = Users.UserID;

--Join Posts and Comments: List all posts and any comments made on them.

SELECT Posts.Title, Comments.CommentText

FROM Posts

LEFT JOIN Comments ON Posts.PostID = Comments.PostID;

--Multiple Joins: Get all comments with the corresponding post title and user name.

SELECT Posts.Title, Users.Username, Comments.CommentText

FROM Comments

INNER JOIN Posts ON Comments.PostID = Posts.PostID

INNER JOIN Users ON Comments.UserID = Users.UserID;

--Count of Posts by User: Display users along with the number of posts they've made.

SELECT Users.Username, COUNT(Posts.PostID) AS PostCount

FROM Users

LEFT JOIN Posts ON Users.UserID = Posts.UserID

GROUP BY Users.Username;

--Advanced Join Queries

--Posts with Comment Count: Show all posts along with the total number of comments on each post.

SELECT Posts.Title, COUNT(Comments.CommentID) AS CommentCount

FROM Posts

LEFT JOIN Comments ON Posts.PostID = Comments.PostID

GROUP BY Posts.Title;

--Users with No Posts: Find users who haven't made any posts.

SELECT Users.Username

FROM Users

LEFT JOIN Posts ON Users.UserID = Posts.UserID

WHERE Posts.PostID IS NULL;

--Posts without Comments: List all posts that have no comments.

SELECT Posts.Title

FROM Posts

LEFT JOIN Comments ON Posts.PostID = Comments.PostID

WHERE Comments.CommentID IS NULL;

--Latest Post by Each User: Find the most recent post by each user.

SELECT Users.Username, MAX(Posts.PostDate) AS LatestPostDate

FROM Users

INNER JOIN Posts ON Users.UserID = Posts.UserID

GROUP BY Users.Username;

--Comments on a Specific User's Posts: Retrieve comments for posts made by a specific user (e.g., UserID = 1).

SELECT Posts.Title, Comments.CommentText

FROM Posts

INNER JOIN Comments ON Posts.PostID = Comments.PostID

WHERE Posts.UserID = 1;

--Complex Join Queries

--Users, Their Posts, and Comment Counts on Each Post: Show user names, their post titles, and the number of comments on each post.

SELECT Users.Username, Posts.Title, COUNT(Comments.CommentID) AS CommentCount

FROM Users

INNER JOIN Posts ON Users.UserID = Posts.UserID

LEFT JOIN Comments ON Posts.PostID = Comments.PostID

GROUP BY Users.Username, Posts.Title;

--Find Users Who Have Commented on Every Post: Identify users who have commented on every post.

SELECT Users.Username

FROM Users

JOIN Comments ON Users.UserID = Comments.UserID

GROUP BY Users.UserID, Users.Username

HAVING COUNT(DISTINCT Comments.PostID) = (SELECT COUNT(\*) FROM Posts);

--All Posts and Any Users Who Have Not Commented on Them: List all posts and users who haven't commented on them.

SELECT Posts.Title, Users.Username

FROM Posts

CROSS JOIN Users

LEFT JOIN Comments ON Posts.PostID = Comments.PostID AND Users.UserID = Comments.UserID

WHERE Comments.CommentID IS NULL;

--Users and Their First Post: Get each user's first post based on the post date.

SELECT Users.Username, MIN(Posts.PostDate) AS FirstPostDate, Posts.Title

FROM Users

INNER JOIN Posts ON Users.UserID = Posts.UserID

GROUP BY Users.Username, Posts.Title

HAVING MIN(Posts.PostDate) IS NOT NULL;

--Rank Posts by Comment Count: Rank posts based on the number of comments.

SELECT Posts.Title, COUNT(Comments.CommentID) AS CommentCount, RANK() OVER (ORDER BY COUNT(Comments.CommentID) DESC) AS Rank

FROM Posts

LEFT JOIN Comments ON Posts.PostID = Comments.PostID

GROUP BY Posts.Title;

1. Users and Posts: Get all posts with their corresponding user names. (1 mark)
2. Users and Comments: Find all comments made by each user. (1 mark)
3. Posts and Comments: List all posts and any comments made on them. (1 mark)
4. Multiple Joins: Get all comments with the corresponding post title and user name. (2 marks)
5. Count of Posts by User: Display users along with the number of posts they've made. (1 mark)
6. Posts with Comment Count: Show all posts along with the total number of comments on each post. (1 mark)
7. Users with No Posts: Find users who haven't made any posts. (1 mark)
8. Posts without Comments: List all posts that have no comments. (1 mark)
9. Latest Post by Each User: Find the most recent post by each user. (2 marks)
10. Comments on a Specific User's Posts: Retrieve comments for posts made by a specific user (e.g., UserID = 1). (1 mark)
11. Users, Their Posts, and Comment Counts on Each Post: Show user names, their post titles, and the number of comments on each post. (2 marks)
12. Find Users Who Have Commented on Every Post: Identify users who have commented on every post. (2 marks)
13. All Posts and Any Users Who Have Not Commented on Them: List all posts and users who haven't commented on them. (2 marks)
14. Users and Their First Post: Get each user's first post based on the post date. (2 marks)
15. Rank Posts by Comment Count: Rank posts based on the number of comments. (1 mark)