

Introduction

This database schema represents a social networking website where users can manage their profiles, posts, comments, images, links, and friendships. The system consists of several interconnected tables, each storing specific data:

- Users: Stores basic user information such as username, first and last names, password, description, and the date of joining.
- Links: Stores links to users' external profiles or websites, linked to the Users table.
- Entries: Stores posts made by users, including the post's ID, time, and the user who posted it.
- Comments: Stores comments on posts, allowing replies and hierarchical comment structures, linked to the Entries table.
- Images: Stores images related to posts, connected to the Entries table.
- Blurbs: Stores descriptive text for posts, linked to the Entries table.
- Friends: Stores friendship relationships between users, including the start date of the friendship, with links to the Users table.

These tables work together to enable features like posting, commenting, sharing images, and managing friendships, providing a comprehensive social networking experience.

Database Implementation

```
CREATE TABLE users (  
    login VARCHAR2(255) PRIMARY KEY,  
    firstName VARCHAR2(100),  
    lastName VARCHAR2(100),  
    password VARCHAR2(255),  
    description VARCHAR2(1000),  
    dateJoined DATE  
);  
  
CREATE TABLE links (  
    userI VARCHAR2(255),  
    linkNo INT,  
    link VARCHAR2(255),  
    CONSTRAINT PK_links PRIMARY KEY (userI, linkNo),  
    FOREIGN KEY (userI) REFERENCES users(login)  
);  
  
CREATE TABLE entries (  
    entryId INT PRIMARY KEY,  
    usere VARCHAR2(255),  
    time DATE,  
    FOREIGN KEY (usere) REFERENCES users(login)  
);  
  
CREATE TABLE comments (  
    entry INT,  
    parentEntry INT,  
    rootEntry INT,
```

```
text VARCHAR2(1000),  
CONSTRAINT PK_comments PRIMARY KEY (entry, parentEntry,  
rootEntry),  
FOREIGN KEY (entry) REFERENCES entries(entryId),  
FOREIGN KEY (parentEntry) REFERENCES entries(entryId),  
FOREIGN KEY (rootEntry) REFERENCES entries(entryId)  
);
```

```
CREATE TABLE images (  
entry INT PRIMARY KEY,  
caption VARCHAR2(255),  
image VARCHAR2(255),  
FOREIGN KEY (entry) REFERENCES entries(entryId)  
);
```

```
CREATE TABLE blurbs (  
entry INT PRIMARY KEY,  
text VARCHAR2(1000),  
FOREIGN KEY (entry) REFERENCES entries(entryId)  
);
```

```
CREATE TABLE friends (  
userf VARCHAR2(255),  
friend VARCHAR2(255),  
sinceDate DATE,  
CONSTRAINT PK_friends PRIMARY KEY (userf, friend),  
FOREIGN KEY (userf) REFERENCES users(login),  
FOREIGN KEY (friend) REFERENCES users(login)  
);
```

Inserted Values

Users

INSERT INTO Users (login, firstName, lastName, password, description, dateJoined)

VALUES ('tasneem123', 'Tasneem', 'Ibrahim', 'rNdoM1234', 'Computer Engineer', '2023-03-15');

INSERT INTO Users (login, firstName, lastName, password, description, dateJoined)

VALUES ('ahmad456', 'Ahmad', 'Mosameh', 'Z1h9i3T1z', 'Mathematics Teacher', '2022-10-22');

INSERT INTO Users (login, firstName, lastName, password, description, dateJoined)

VALUES ('hanan789', 'Hanan', 'Zebdeh', 'AaB3jP6@1', 'Math Student', '2021-08-10');

INSERT INTO Users (login, firstName, lastName, password, description, dateJoined)

VALUES ('yazan321', 'Yazan', 'Osama', 'Yz9WsX7\$1', 'Software Developer', '2022-12-01');

INSERT INTO Users (login, firstName, lastName, password, description, dateJoined)

VALUES ('mohammad123', 'Mohammad', 'Khaled', 'tX3yH2z1', 'Data Analyst', '2024-02-10');

```
SELECT * FROM users;
```

Output

login	firstName	lastName	password	description	dateJoined
tasneem123	Tasneem	Ibrahim	rNdoM1234	Computer Engineer	2023-03-15
ahmad456	Ahmad	Mosameh	Z1h9i3T1z	Mathematics Teacher	2022-10-22
hanan789	Hanan	Zebdeh	AaB3jP6@1	Math Student	2021-08-10
yazan321	Yazan	Osama	Yz9WsX7\$1	Software Developer	2022-12-01
mohammad123	Mohammad	Khaled	tX3yH2z1	Data Analyst	2024-02-10

links

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('tasneem123', 1,  
'https://www.linkedin.com/in/tasneemibrahim/');
```

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('tasneem123', 2, 'https://github.com/tasneem-portfolio');
```

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('ahmad456', 3,  
'https://www.linkedin.com/in/ahmadmosameh/');
```

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('ahmad456', 4, 'https://github.com/ahmad-art-portfolio');
```

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('hanan789', 1, 'https://www.linkedin.com/in/hananzebdeh/');
```

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('yazan321', 2, 'https://github.com/yazandevportfolio');
```

```
INSERT INTO links (userI, linkNo, link)
```

```
VALUES ('yazan321', 3, 'https://www.linkedin.com/in/yazan-osama/');
```

```
SELECT * FROM links ;
```

Output

userI	linkNo	link
tasneem123	1	https://www.linkedin.com/in/tasneemibrahim/
tasneem123	2	https://github.com/tasneem-portfolio
ahmad456	3	https://www.linkedin.com/in/ahmadmosameh/
ahmad456	4	https://github.com/ahmad-art-portfolio
hanan789	1	https://www.linkedin.com/in/hananzebdeh/
yazan321	2	https://github.com/yazandevportfolio
yazan321	3	https://www.linkedin.com/in/yazan-osama/

Entries

```
INSERT INTO entries (entryId, usere, time)
VALUES (1, 'tasneem123', '2024-01-10');
INSERT INTO entries (entryId, usere, time)
VALUES (2, 'ahmad456', '2024-01-12');
INSERT INTO entries (entryId, usere, time)
VALUES (3, 'hanan789', '2024-01-14');
INSERT INTO entries (entryId, usere, time)
VALUES (4, 'yazan321', '2024-01-15');
INSERT INTO entries (entryId, usere, time)
VALUES (5, 'mohammad123', '2024-01-16');
```

```
SELECT * FROM entries ;
```

Output

entryId	usere	time
1	tasneem123	2024-01-10
2	ahmad456	2024-01-12
3	hanan789	2024-01-14
4	yazan321	2024-01-15
5	mohammad123	2024-01-16

Comments

```
INSERT INTO comments (entry, parentEntry, rootEntry, text)
VALUES (2, 1, 1, 'This project is absolutely fantastic, Tasneem! The
attention to detail in your work is impressive. Keep it up!');
INSERT INTO comments (entry, parentEntry, rootEntry, text)
VALUES (3, 2, 1, 'Great work on this, Ahmad! I really admire how you
optimized the code for better performance. It looks really clean.');
```

```
INSERT INTO comments (entry, parentEntry, rootEntry, text)
```

VALUES (4, 3, 3, 'Noor, your designs are top-notch! The color choices and the overall layout are very user-friendly.');

INSERT INTO comments (entry, parentEntry, rootEntry, text)

VALUES (1, 4, 4, 'Omar, your analysis on the data trends is very thorough. It really helps in understanding the bigger picture.');

INSERT INTO comments (entry, parentEntry, rootEntry, text)

VALUES (1, 3, 3, 'Your coding journey has been inspiring, Tasneem. Your hard work and determination shine through your projects!');

```
SELECT * FROM comments ;
```

Output

entry	parentEntry	rootEntry	text
2	1	1	This project is absolutely fantastic, Tasneem! The attention to detail in your work is impressive. Keep it up!
3	2	1	Great work on this, Ahmad! I really admire how you optimized the code for better performance. It looks really clean.
4	3	3	Noor, your designs are top-notch! The color choices and the overall layout are very user-friendly.
1	4	4	Omar, your analysis on the data trends is very thorough. It really helps in understanding the bigger picture.
1	3	3	Your coding journey has been inspiring, Tasneem. Your hard work and determination shine through your projects!

Images

INSERT INTO images (entry, caption, image)

VALUES (1, 'Tasneems Coding Project', 'path/to/image1.jpg');

INSERT INTO images (entry, caption, image)

VALUES (2, 'Ahmads Debugging Tips', 'path/to/image2.jpg');

INSERT INTO images (entry, caption, image)

VALUES (3, 'Hanans Math Solutions', 'path/to/image3.jpg');

INSERT INTO images (entry, caption, image)

VALUES (4, 'Yazans Software Development', 'path/to/image4.jpg');

INSERT INTO images (entry, caption, image)

VALUES (5, 'Mohammads Data Analysis', 'path/to/image5.jpg');

```
SELECT * FROM images ;
```

Output

entry	caption	image
1	Tasneems Coding Project	path/to/image1.jpg
2	Ahmads Debugging Tips	path/to/image2.jpg
3	Hanans Math Solutions	path/to/image3.jpg
4	Yazans Software Development	path/to/image4.jpg
5	Mohammads Data Analysis	path/to/image5.jpg

Blurbs

```
INSERT INTO blurbs (entry, text)
```

```
VALUES (1, 'Tasneems project is incredibly well-structured and innovative.');
```

```
INSERT INTO blurbs (entry, text)
```

```
VALUES (2, 'Ahmad provides detailed insights into problem-solving techniques.');
```

```
INSERT INTO blurbs (entry, text)
```

```
VALUES (3, 'Hanan has an amazing ability to simplify complex concepts.');
```

```
INSERT INTO blurbs (entry, text)
```

```
VALUES (4, 'Yazan always focuses on clean and efficient coding.');
```

```
INSERT INTO blurbs (entry, text)
```

```
VALUES (5, 'Mohammads analysis is data-driven and always on point.');
```



```
SELECT * FROM blurbs ;
```

Output

entry	text
1	Tasneems project is incredibly well-structured and innovative.
2	Ahmad provides detailed insights into problem-solving techniques.
3	Hanan has an amazing ability to simplify complex concepts.
4	Yazan always focuses on clean and efficient coding.
5	Mohammads analysis is data-driven and always on point.

Friends

```
INSERT INTO friends (userf, friend, sinceDate)
VALUES ('tasneem123', 'ahmad456', '2022-03-10');
```

```
INSERT INTO friends (userf, friend, sinceDate)
VALUES ('ahmad456', 'hanan789', '2021-11-20');
```

```
INSERT INTO friends (userf, friend, sinceDate)
VALUES ('hanan789', 'yazan321', '2022-06-15');
```

```
INSERT INTO friends (userf, friend, sinceDate)
VALUES ('yazan321', 'mohammad123', '2023-05-01');
```

```
INSERT INTO friends (userf, friend, sinceDate)
VALUES ('mohammad123', 'tasneem123', '2024-01-25');
```

```
SELECT * FROM friends ;
```

Output

userf	friend	sinceDate
tasneem123	ahmad456	2022-03-10
ahmad456	hanan789	2021-11-20
hanan789	yazan321	2022-06-15
yazan321	mohammad123	2023-05-01
mohammad123	tasneem123	2024-01-25

left outer join

1-

Displays all users and their links

```
SELECT u.login, l.link
FROM users u
LEFT JOIN links l ON u.login = l.userl;
```

Output

login	link
ahmad456	https://www.linkedin.com/in/ahmadmosameh/
ahmad456	https://github.com/ahmad-art-portfolio
hanan789	https://www.linkedin.com/in/hananzebdeh/
mohammad123	
tasneem123	https://www.linkedin.com/in/tasneemibrahim/
tasneem123	https://github.com/tasneem-portfolio
yazan321	https://github.com/yazandevportfolio
yazan321	https://www.linkedin.com/in/yazan-osama/

2-

Displays all friends and their first names

```
SELECT f.userf, u.firstName
FROM friends f
LEFT JOIN users u ON f.friend = u.login;
```

Output

userf	firstName
ahmad456	Hanan
hanan789	Yazan
mohammad123	Tasneem
tasneem123	Ahmad
yazan321	Mohammad

right outer join

1-

Displays all users and their links

```
SELECT l.link, u.firstName
```

FROM links l RIGHT JOIN users u

ON l.userl = u.login;

firstName	link
Tasneem	https://www.linkedin.com/in/tasneemibrahim/
Tasneem	https://github.com/tasneem-portfolio
Ahmad	https://www.linkedin.com/in/ahmadmosameh/
Ahmad	https://github.com/ahmad-art-portfolio
Hanan	NULL
Yazan	NULL
Mohammad	NULL

2-

Displays all text and their time

SELECT c.text, e.time

FROM comments c

RIGHT JOIN entries e ON c.entry = e.entryId;

time	text
2024-01-10	This project is absolutely fantastic, Tasneem! ...
2024-01-12	Great work on this, Ahmad! ...
2024-01-14	Noor, your designs are top-notch! ...
2024-01-15	Omar, your analysis on the data trends is very thorough.
2024-01-16	NULL

full outer join

1-

Displays all users and their friends . If a user has no friend, shows NULL in the friend column, and if a friend has no user,shows NULL in the login column.

```
SELECT u.login, f.friend
```

```
FROM users u
```

```
FULL OUTER JOIN friends f ON u.login = f.userf;
```

friend	login
Ahmad	tasneem123
Hanan	ahmad456
Yazan	hanan789
Mohammad	yazan321
Tasneem	mohammad123

2-

Displays all entries and their blurbs . If there is no blurb for an entry, it shows NULL in the text column.

```
SELECT e.entryId, b.text
```

```
FROM entries e FULL OUTER JOIN blurbs b
```

```
ON e.entryId = b.entry;
```

entryId	text
1	Tasneems project is incredibly well-structured...
2	Ahmad provides detailed insights into problem-solving techniques.
3	Hanan has an amazing ability to simplify complex concepts.
4	Yazan always focuses on clean and efficient coding.

5	Mohammads analysis is data-driven and always on point.

Write an SQL to retrieve the names of users who have posted comments on entries made by their friends. Use a subquery to identify the entries created by each user's friends.

```
SELECT U.firstName AS UserName, F.friend AS FriendName, C.text AS CommentText
FROM (
    SELECT F.userf, F.friend, E.entryId
    FROM friends F
    JOIN entries E ON F.friend = E.usere
) F
JOIN comments C
ON F.entryId = C.parentEntry
JOIN entries E
ON C.entry = E.entryId
JOIN users U
ON E.usere = U.login;
```

Output

UserName	FriendName	CommentText
Ahmad	tasneem123	This project is absolutely fantastic, Tasneem! The attention to detail in your work is impressive. Keep it up!
Hanan	ahmad456	Great work on this, Ahmad! I really admire how you optimized the code for better performance. It looks really clean.
Yazan	hanan789	Noor, your designs are top-notch! The color choices and the overall layout are very user-friendly.
Tasneem	yazan321	Omar, your analysis on the data trends is very thorough. It really helps in understanding the bigger picture.
Tasneem	hanan789	Your coding journey has been inspiring, Tasneem. Your hard work and determination shine through your projects!

Write an SQL query to display the users and the number of friends for each user. Use the COUNT aggregate function with GROUP BY.

```
SELECT U.login AS UserLogin, U.firstName AS FirstName, COUNT(F.friend) AS FriendCount
FROM Users U LEFT JOIN friends F
ON U.login = F.userf
GROUP BY U.login, U.firstName;
```

Output

UserLogin	FirstName	FriendCount
ahmad456	Ahmad	1
hanan789	Hanan	1
mohammad123	Mohammad	1
tasneem123	Tasneem	1
yazan321	Yazan	1