DMDD Assignment 3 (Sociable)

Team members:

1. Harshila Jagtap
   NUID : 002743674

GitHubLink_Harshila

2. Yash Pawar
   NUID : 002747371

GitHubLink_Yash

About :

Today we are in the 21st century, where we can see a steep rise in people migrating to different places for study, work, lifestyle, and travel. They find it difficult to socialize and make new friends. Moreover, it is a difficult and cumbersome experience to find people and groups with similar interests in a short span of time. Socialize presents its users to easily find like-minded people amongst unknowns, quickly socialize and join social groups, attend events, converse, etc. This project aims to gather data, analyze and recommend like-minded suggestions by analyzing people's interests, and present statistics on the same.

ER and UML Diagrams :

ER and UML Diagrams

Follow Readme Document for detailed process to install and run the code :

README.md

Sources of Data :

➔ **Imported JSON Files to load the data**

Path to JSON Folder

Code File with output results of audit :

DMDD_EDA_Assignment_3.ipynb

---

10 Use-Cases :

1. Use Case: View Groups with specific interest category

Description: User views a list of groups with mentioned interest

Actors: User

Precondition: User must specify his interests and there must be groups present

Steps:

Actor action – User views groups with specified interests

System Responses – groups would be displayed

Post Condition: system displays group list

Query :

```
SELECT * from Group_Details
INNER JOIN Category
ON group_Details.category_id=Category.category_id
Where Category.name LIKE "Basketball";
```

2. Use Case: View Events attended by specific user

Description: User views a list of events attended

Actors: User

Precondition: User must attend an event

Steps:

Actor action – User views events attended

System Responses – events would be displayed

Post Condition: system displays attended event list

Query :

```
SELECT ed.event_name,ed.event_date from Event_Details ed
INNER JOIN User_Events ue
ON ed.event_id=ue.event_id
WHERE ue.user_id=2;
```

3. Use Case: View Matched User Details

Description: User views a list of users matched

Actors: User

Precondition: User must match another user

Steps:

Actor action – User views matched users

System Responses –Matched User List would be displayed

Post Condition: system displays Matched User list

Query :

```
SELECT u.nickname,u.country,u.age from User u
INNER JOIN Match_Details md
ON u.user_id=md.user1_id
where u.user_id=2;
```

4.Use Case: View Users with specific interest

Description: User views a list of users with mentioned interest

Actors: User

Precondition: User must have mentioned interests

Steps:

Actor action – User views other users with BasketBall interest

System Responses –Matched User List with specific list would be displayed

Post Condition: system displays Matched User list

Query :

```
SELECT u.name,u.nickname,u.country,u.age from User u
INNER JOIN User_InterestCategory ui
ON u.user_id=ui.user_id
INNER JOIN Category c
ON ui.category_id=c.category_id
where c.name LIKE "Basketball" AND u.user_id !=2;
```

5.Use Case: View list of Blocked users for specificed user

Description: User views a list of blocked users

Actors: User

Precondition: User should block atleast 1 user

Steps:

Actor action – User views other blocked users

System Responses –Matched User List with specific list would be displayed

Post Condition: system displays Matched User list

Query :

```
SELECT u.name,u.nickname,u.country,u.age from User u
INNER JOIN Blocked_user bu
ON u.user_id=bu.blocked_whom
where bu.blocked_by=2;
```

6.Use Case: View list of Matched users in the month of September

Description: admin views a list of matched users in september

Actors: admin

Precondition: there should be matches in database

Steps:

Actor action – admin views list of Matched users in the month of September

System Responses –Matched User List with specific month would be displayed

Post Condition: system displays Matched User list for September

Query :

```
SELECT md.user1_id,md.user2_id from Match_Details md
INNER JOIN User u
```

ON u.user_id=md.user1_id OR u.user_id=md.user2_id

where MONTH(md.match_time)=09;

7.Use Case: View list of Matched users with interest category Basketball

Description: admin views a list of matched users with common interest Category

Actors: admin

Precondition: there should be matches in database

Steps:

Actor action – admin views list of Matched users with common interest

System Responses –Matched User List with common interest would be displayed

Post Condition: system displays Matched User list

Query :

SELECT md.user1_id,md.user2_id from Match_Details md

INNER JOIN User u

ON u.user_id=md.user1_id OR u.user_id=md.user2_id

INNER JOIN Category c

ON md.category_id=c.category_id

where c.name LIKE "Basketball";

8.Use Case: View Conversation of Matched users along with their ids

Description: admin views Conversation of matched users

Actors: admin

Precondition: there should be matches in database

Steps:

Actor action – admin views conversation of Matched users with common interest

System Responses –Matched User List along with their messages would be displayed

Post Condition: system displays Matched User list with text messages

Query :

```
SELECT md.user1_id,md.user2_id, c.message from Match_Details md
INNER JOIN Conversation c
ON c.match_id=md.match_id;
```

9.Use Case: View groups with similar interest categories of a user

Description: User views List of groups with similar interest category

Actors: User

Precondition: there should be groups in database

Steps:

Actor action – User views List of groups with similar interest category

System Responses –Group List having having same interests

Post Condition: system displays list of groups

Query :

```
SELECT g.group_name from User u
INNER JOIN User_InterestCategory ui
ON u.user_id=ui.user_id
INNER JOIN Group_Details g
ON g.category_id=ui.category_id
where u.user_id=2;
```

10. Use Case: View events which share common liking with given user Description: User views List of events with common interest categoryActors: User

Precondition: there should be events in database

Steps:

Actor action – User views List of events with similar interest category

System Responses –event List having having same interests

Post Condition: system displays list of events

Query :

```
SELECT g.group_name, ed.event_name, ed.event_date from User u
INNER JOIN User_InterestCategory ui
ON u.user_id=ui.user_id
INNER JOIN Group_Details g
ON g.category_id=ui.category_id
INNER JOIN Event_Details ed
ON g.group_id=ed.group_id
where u.user_id=2;
```

Queries for the previous assignment created so far :

Q1. Fetch list of frequently conversed users for UserID = 2 in the month of September 2022.

Query :

```
SELECT md.user1_id,md.user2_id from Match_Details md
INNER JOIN User u
ON u.user_id=md.user1_id OR u.user_id=md.user2_id
where MONTH(md.match_time)=09;
```

Q2. Fetch username, nickname of blocked users by UserID = 2 with common interest sorted by username in descending order.

Query :

```
SELECT u.name,u.nickname from User u
INNER JOIN Blocked_user bu
ON u.user_id=bu.blocked_whom
where bu.blocked_by=2 order by u.name desc;
```

Q3. Fetch list of users who have not conversed with any other user.

Query :

```
SELECT * from User where user_id NOT IN (select user1_id from Match_Details);
```

Q4. Fetch top 5 most suitable matches for UserID=2 from native country and common interest.

Query :

```
select * from User u where u.country=(Select country from User where user_id=2) AND
u.user_id!=2 limit 0,5;
```

Q5. Fetch a list of users who attended maximum events in category sport from date range 1st September 2022 to date range 30th September,2022

Query :

```
SELECT u.name from User u

INNER JOIN User_Events ue

ON u.user_id=ue.user_id

INNER JOIN Event_Details ed

ON ue.event_id=ed.event_id

INNER JOIN Group_Details gd

ON ed.group_id=gd.group_id

INNER JOIN Category c

ON c.category_id=gd.category_id

WHERE c.name LIKE "SPORT" AND MONTH(ed.event_date)=12 AND
YEAR(ed.event_date)=2022;
```

Q6. Fetch a list of users who have created an account, but account verification is not done.

Query :

```
SELECT * from User where Is_Verified=0;
```

Q7. Fetch list of users who have deleted their account.

Query :

```
SELECT * from User where Is_Deleted=1;
```

Q8. Fetch list of users who are inactive/ not logged in for the past 2 months.

Query :

```
select name from user where last_login <= now()-interval 2 month;
```

Q9. Fetch each count of active male and female users who have registered the application.

Query :

```
SELECT gender,COUNT(*)
FROM user
GROUP BY gender;
```

Q10. Fetch a list of users who have connected with most connections today.

Query :

```
SELECT u.name,md.match_id, md.match_time,COUNT(*)
FROM match_details    md
  inner join user u
  on u.user_id= md.user1_id
GROUP BY match_time
HAVING COUNT(*) > 1
;
```