

پروژه قناری فاز اول

بردیا اردکانیان

9831072

REFERENCES

CORE TABLES

1. USER
2. TWEET
3. MESSAGE
4. HASHTAG

MAPPING TABLES (MAP TWO OR MORE ENTRIES TOGETHER)

1. LOGIN
2. FOLLOWER
3. BLOCKED
4. LIKED
5. REPLY

FUNCTIONS

1. Register (create new user)
2. Login
3. Login History
4. Post (tweet a new tweet)
5. Get Personal Tweets
6. Follow
7. Unfollow
8. Block
9. Unblock
10. Timeline (following activity)
11. User Activity
12. Add Reply
13. Show Replies
14. Add #
15. Get # (all tweets with a specific tag)
16. Like
17. Likes Count
18. Likes List
19. Get Trending Tweets
20. Send Message
21. Get Messages From User
22. Get Message Senders List

EER Diagram

CORE TABLES

USER

```
CREATE TABLE twitter.user(  
    FIRSTNAME NVARCHAR(30) NOT NULL,  
    LASTNAME NVARCHAR(30) NOT NULL,  
    ID NVARCHAR(20) NOT NULL,  
    PASSWORD_HASH NVARCHAR(128) NOT NULL,  
    BORN DATE NOT NULL,  
    JOINED DATE NOT NULL,  
    BIOGRAPHY NVARCHAR(64) NOT NULL,  
    PRIMARY KEY (ID))
```

TWEET

```
CREATE TABLE twitter.tweet(  
    ID INT AUTO_INCREMENT NOT NULL,  
    USER_ID NVARCHAR(20) NOT  
NULL,  
    CONTENT NVARCHAR(256) NOT  
NULL,  
    POSTAGE DATE NOT  
NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (USER_ID) REFERENCES twitter.user(ID))
```

MESSAGE

```
CREATE TABLE twitter.message(  
    SENDER_ID NVARCHAR(20) NOT NULL,  
    RECEIVER_ID NVARCHAR(20) NOT NULL,  
    TWEET_ID INT,  
    CONTENT NVARCHAR(256),  
    POSTAGE DATE,  
    TYPE INT NOT NULL,  
    FOREIGN KEY (SENDER_ID) REFERENCES twitter.user(ID),  
    FOREIGN KEY (RECEIVER_ID) REFERENCES twitter.user(ID),  
    FOREIGN KEY (TWEET_ID) REFERENCES twitter.tweet(ID))
```

```
/*  
    TYPE == 0 TWEET  
    TYPE == 1 CONTENT  
*/
```

HASHTAG

```
CREATE TABLE twitter.hashtag(  
    TWEET_ID INT NOT NULL,  
    TAG NVARCHAR(6) NOT NULL,  
    FOREIGN KEY (TWEET_ID) REFERENCES twitter.tweet(ID))
```

MAPPING

LOGIN

```
CREATE TABLE twitter.login(  
    ID NVARCHAR(20) NOT NULL,  
    TIME DATE NOT NULL,  
    FOREIGN KEY (ID) REFERENCES twitter.user(ID))
```

FOLLOWER

```
CREATE TABLE twitter.follower(  
    SRC_ID NVARCHAR(20),  
    DEST_ID NVARCHAR(20),  
    PRIMARY KEY (SRC_ID, DEST_ID),  
    FOREIGN KEY (SRC_ID) REFERENCES twitter.user(ID),  
    FOREIGN KEY (DEST_ID) REFERENCES twitter.user(ID))
```

BLOCKED

```
CREATE TABLE twitter.blocked(  
    SRC_ID NVARCHAR(20),  
    DEST_ID NVARCHAR(20),  
    PRIMARY KEY (SRC_ID, DEST_ID),  
    FOREIGN KEY (SRC_ID) REFERENCES twitter.user(ID),  
    FOREIGN KEY (DEST_ID) REFERENCES twitter.user(ID))
```

LIKED

```
CREATE TABLE liked(  
    TWEET_ID INT NOT NULL,  
    USER_ID NVARCHAR(20) NOT NULL,  
    PRIMARY KEY (TWEET_ID, USER_ID),  
    FOREIGN KEY (TWEET_ID) REFERENCES twitter.tweet(ID),  
    FOREIGN KEY (USER_ID) REFERENCES twitter.user(ID))
```

REPLY

```
CREATE TABLE twitter.reply(  
    ID                                INT AUTO_INCREMENT NOT NULL,  
    TWEET_ID                         INT NOT NULL,  
    REPLY_ID                         INT NOT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (TWEET_ID) REFERENCES twitter.tweet(ID),  
    FOREIGN KEY (REPLY_ID) REFERENCES twitter.tweet(ID))
```

FUNCTIONS

Register

```
INSERT INTO twitter.user VALUES('Bardia', 'Ardakanian',  
'BardiArdakanian', SHA2('11111111', 256), '2001-09-02', '2021-02-05',  
'CE AUT');
```

```
INSERT INTO twitter.user VALUES('Sina', 'Amini', 'Trueshinnyjesus',  
SHA2('22222222', 256), '2000-05-12', '2021-02-05', 'My name is  
jeff');
```

```
INSERT INTO twitter.user VALUES('Mohammad', 'Salehi', 'M_salehi_V',  
SHA2('33333333', 256), '1999-11-23', '2021-02-05', 'Mohammad Salehi  
son of Reza');
```

```
INSERT INTO twitter.user VALUES('Keivan', 'Ipchi', 'keivanIphci',  
SHA2('44444444', 256), '2002-04-15', '2021-02-05', 'Relationship  
status: Neftlix and icecream');
```

```
INSERT INTO twitter.user VALUES('Amirhossein', 'Najafi',  
'AmirNajafi21', SHA2('55555555', 256), '1998-01-03', '2021-02-05',  
'It was too long go read it yourself:');
```

```
INSERT INTO twitter.user VALUES('Omid', 'Dokhanchi', 'Stalin00',  
SHA2('66666666', 256), '2003-03-12', '2021-02-05', 'hail hentai');
```

Login

```
SELECT *  
FROM twitter.user  
WHERE ID='bardiArdakanian' AND PASSWORD_HASH=SHA2('11111111', 256);
```

Login History

Adding new record

```
INSERT INTO twitter.login VALUES('bardiArdakanian', current_date())
```

Check history

```
SELECT * FROM twitter.login ORDER BY TIME DESC
```

Post

```
INSERT INTO twitter.tweet (USER_ID, CONTENT, POSTAGE)  
VALUES('bardiArdakanian', 'I just took an exam... Im gonna be  
unemployed:)', current_date())
```

Get Personal Tweets

```
SELECT * FROM twitter.tweet WHERE USER_ID='bardiArdakanian'
```

Follow

```
INSERT INTO twitter.follower VALUES('bardiArdakanian', 'Stalin00')
```

Unfollow

```
DELETE FROM twitter.follower  
WHERE SRC_ID='bardiArdakanian' and DEST_ID='Stalin00'
```

Block

```
INSERT INTO twitter.blocked VALUES('keivanIphci', 'M_salehi_V')
```

Unblock

```
DELETE FROM twitter.blocked
WHERE SRC_ID='bardiArdakanian' and DEST_ID='Stalin00'
```

Timeline

```
SELECT tweet.USER_ID, tweet.CONTENT, tweet.POSTAGE
FROM twitter.tweet, twitter.follower, twitter.blocked
WHERE (follower.SRC_ID = blocked.SRC_ID
      and follower.DEST_ID != blocked.DEST_ID)
      and (follower.DEST_ID = tweet.USER_ID)
      and (follower.SRC_ID = 'bardiArdakanian');
```

User activity

```
SELECT T.USER_ID, T.CONTENT, T.POSTAGE
FROM twitter.tweet AS T
WHERE (T.USER_ID = 'M_salehi_V')
and not exists (
    SELECT *
    FROM twitter.blocked as B
    WHERE B.SRC_ID = 'M_salehi_V' and B.DEST_ID = 'Stalin00')
```

Add Reply

```
INSERT INTO twitter.reply(TWEET_ID, REPLY_ID) SELECT 6,11
WHERE NOT EXISTS (
    SELECT *
    FROM twitter.blocked AS B, twitter.tweet AS T1, twitter.tweet AS
T2
    WHERE (T1.ID = 10 AND T1.USER_ID = B.SRC_ID)
    AND (T2.ID = 11 AND T2.USER_ID = B.DEST_ID)
)
```

Show Replies

```
SELECT *
FROM twitter.reply AS R, twitter.tweet AS T1, twitter.tweet AS T2
WHERE (R.TWEET_ID = T1.ID) AND (R.REPLY_ID = T2.ID) AND TWEET_ID=6
AND NOT EXISTS (
    SELECT * FROM twitter.blocked AS B
    WHERE (T1.USER_ID = B.SRC_ID AND T2.USER_ID = B.DEST_ID)
    OR (T1.USER_ID = B.DEST_ID AND T2.USER_ID = B.SRC_ID)
)
```

Add #

```
INSERT INTO twitter.hashtag VALUES(11, '#fffff')
```

Get #

```
SELECT USER_ID, CONTENT, POSTAGE
FROM twitter.hashtag AS TAG, twitter.tweet AS T
WHERE TAG.TWEET_ID = T.ID AND TAG.TAG = '#abcde'
AND T.USER_ID = 'bardiArdakanian'
AND NOT EXISTS(
SELECT *
FROM twitter.blocked AS B
WHERE B.SRC_ID = T.USER_ID AND B.DEST_ID = 'M_salehi_V')
```

Like

```
INSERT INTO twitter.liked SELECT 2, 'M_salehi_V'
WHERE NOT EXISTS(
SELECT * FROM twitter.blocked AS B, twitter.tweet AS T
WHERE T.ID = 2 AND T.USER_ID = B.SRC_ID AND B.DEST_ID = 'M_salehi_V')
```

Likes Count

```
SELECT COUNT(USER_ID)
FROM twitter.liked AS L
WHERE L.TWEET_ID = 6
AND NOT EXISTS(
SELECT *
FROM twitter.blocked AS B, twitter.tweet AS T
WHERE T.ID = 6 AND T.USER_ID = B.SRC_ID AND B.DEST_ID = 'Stalin00')
```

Likes List

```
SELECT * FROM twitter.liked AS L
WHERE L.TWEET_ID = 6
AND L.USER_ID NOT IN ( -- remove blocked account
SELECT B.SRC_ID FROM twitter.blocked AS B
JOIN twitter.tweet AS T ON B.DEST_ID = T.USER_ID
WHERE L.TWEET_ID = T.ID)
AND NOT EXISTS( -- display nothing if the account has been blocked
SELECT * FROM twitter.blocked AS B, twitter.tweet AS T
WHERE T.ID = 6 AND T.USER_ID = B.SRC_ID AND B.DEST_ID =
'bardiArdakanian')
```


Get Trending Tweets

```
SELECT COUNT(TWEET_ID), T.ID, T.USER_ID, T.CONTENT, T.POSTAGE
FROM twitter.liked AS L
JOIN twitter.tweet AS T ON T.ID = L.TWEET_ID
GROUP BY L.TWEET_ID
ORDER BY COUNT(TWEET_ID) DESC
```

Send Message

```
INSERT
INTO twitter.message (SENDER_ID, RECEIVER_ID, TWEET_ID, POSTAGE,
TYPE) SELECT 'bardiArdakanian', 'M_salehi_V', 13, current_date(), 0
FROM twitter.tweet AS T
WHERE T.ID = 13
AND 'M_salehi_V' NOT IN (
    SELECT B.DEST_ID
    FROM twitter.blocked AS B
    WHERE T.USER_ID = B.SRC_ID)
AND NOT EXISTS(
    SELECT * FROM twitter.blocked AS B
    WHERE (B.SRC_ID = 'bardiArdakanian'
    AND B.DEST_ID = 'M_salehi_V') OR (B.SRC_ID = 'M_salehi_V'
    AND B.DEST_ID = 'bardiArdakanian'))
```

Delete Message

```
DELETE FROM twitter.message
WHERE SENDER_ID = 'Trueshinnyjesus' AND RECEIVER_ID = 'M_salehi_V'
```

Get Messages From User

```
SELECT *
FROM twitter.message AS M, twitter.tweet AS T
WHERE M.RECEIVER_ID = 'M_salehi_V' AND T.ID = M.TWEET_ID
AND 'M_salehi_V' NOT IN (
    SELECT B.DEST_ID
    FROM twitter.blocked AS B
    WHERE T.USER_ID = B.SRC_ID)
ORDER BY M.POSTAGE DESC
```

Get Message Senders List

```
SELECT SENDER_ID, COUNT(SENDER_ID) AS NEW  
FROM twitter.message AS M  
WHERE M.RECEIVER_ID = 'M_salehi_V'  
GROUP BY SENDER_ID  
ORDER BY POSTAGE DESC
```

EER Diagram

