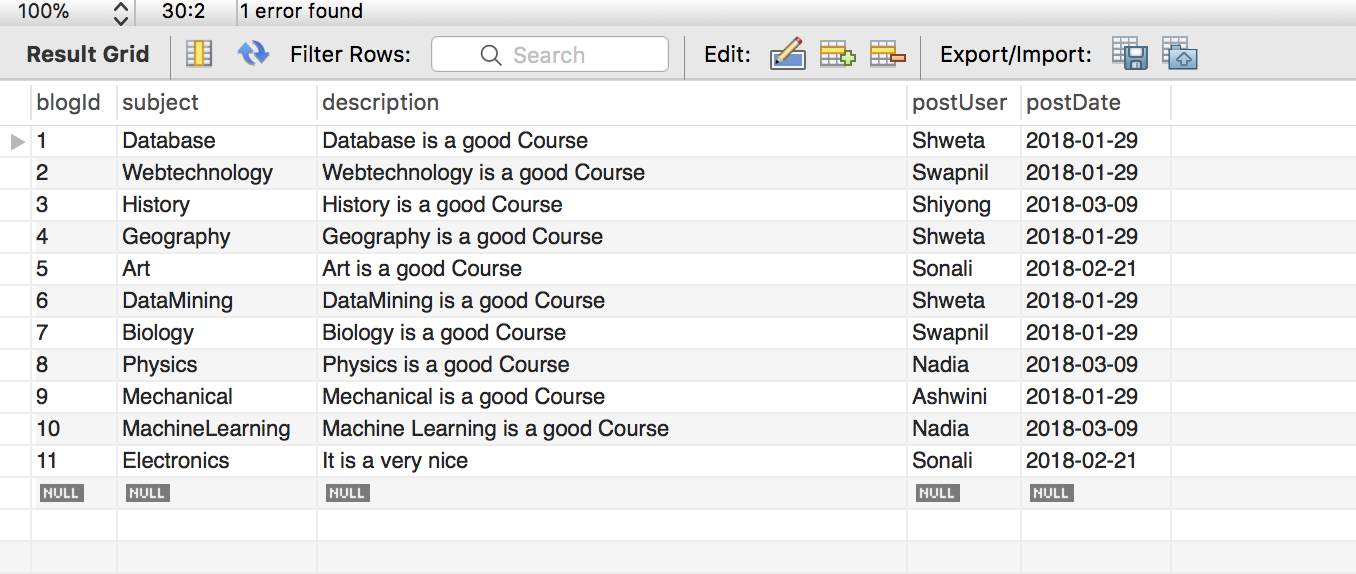
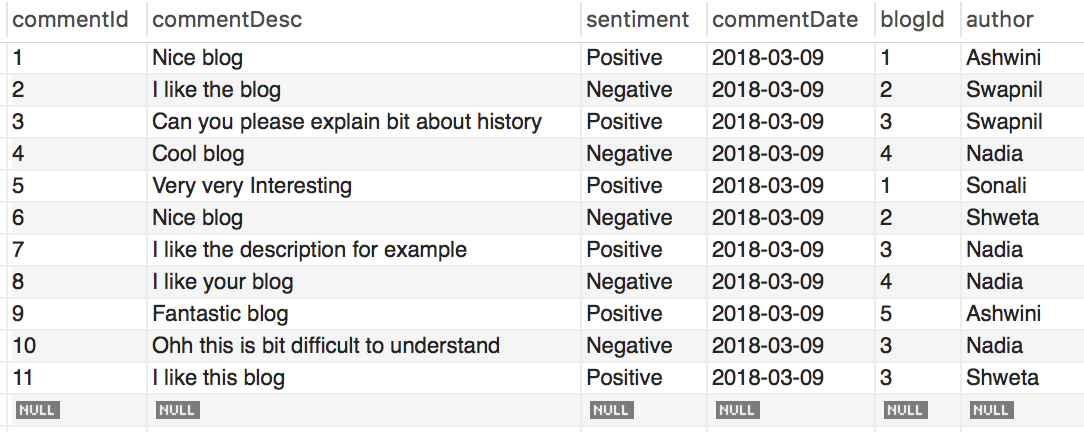
Assignment 2

Below are the table and it’s row on which I have wrote SQL queries-

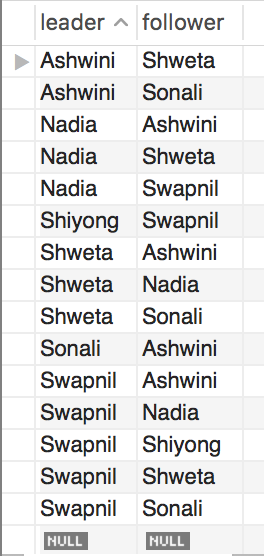
* **SELECT \* FROM Blogs**



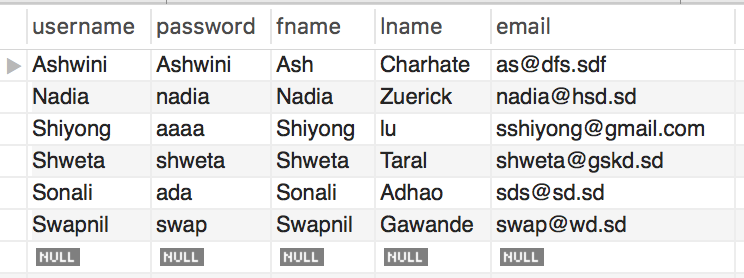
* **SELECT \* FROM Comments;**



* **SELECT \* FROM Followers;**



* **SELECT \* FROM tb\_user;**



1. Write a relational algebra expression and a corresponding SQL statement to list all users (just usernames) who posted at least two blogs on 1/29/2018.

**Solution-**

postUser(COUNT(\*)>=2 γpostUser postDate=‘2018-01-29’(Blogs))

SELECT postUser

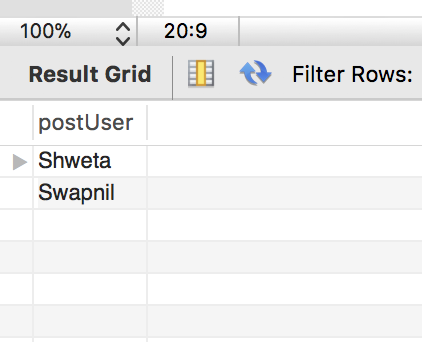
FROM Blogs

WHERE postDate="2018-01-29"

GROUP BY postUser

HAVING COUNT(\*)>=2;

**Result of the above query**



1. Write a relational algebra expression and a corresponding SQL statement to list all users (just usernames) who posted at least two blogs on the same day.

**Solution-**

postUser(COUNT(\*)>=2 γpostUser, postdate (Blogs))

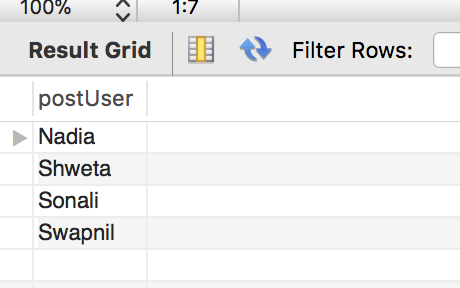
SELECT postUser

FROM Blogs

GROUP BY postUser,postDate

HAVING COUNT(\*)>=2;

**Result of the above query**



1. Write a relational algebra expression and a corresponding SQL statement to the users (just usernames) who give a positive comment to a blog written by one of its followers.

**Solution-**

T1.author(T1.sentiment=‘Positive’ ∧ T1.blogId=T2.blogId

 T3.leader=T1.author ∧ T3.follower=T2.postUser

(ρT1 (Comments) ρT2(Blogs) ρT3 (Follower))

SELECT T1.author

FROM Comments T1,Blogs T2

WHERE T1.sentiment="Positive"

AND T1.blogId=T2.blogId

AND (T1.author,T2.postUser)

IN(

SELECT T3.leader, T3.follower

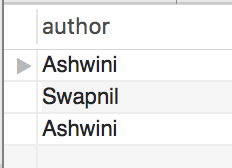
FROM Followers T3

WHERE T3.leader=T1.author

AND T3.follower=T2.postUser

);

**Result of the above query**



1. Write a relational algebra expression and a corresponding SQL statement to return the users (just usernames) who are followed by every other user. Assume a user cannot follow himself/herself.

**Solution-**

leader, follower(Followers) / username(tb\_user)

SELECT distinct leader

FROM Followers T1

WHERE NOT EXISTS(

SELECT username

FROM tb\_user

WHERE T1.leader <> username

AND username

NOT IN(

SELECT follower

FROM Followers T2

WHERE T2.leader=T1.leader));

**Result of the above query**



1. Write a relational algebra expression and a corresponding SQL statement to return blogs (just blogids) that are commented by both user A and user B, A and B are usernames.

**Solution-**

blogId(T1.author=‘**Ashwini’** (Comments))

blogId(T1.author=‘**Sonali’** (Comments))

SELECT T1.blogId

FROM Comments T1, Comments T2

WHERE T1.author="Sonali"

AND T2.author="Shweta"

AND T1.blogId=T2.blogId;

**Result of the above query**



1. Write an SQL statement to return the users (just usernames) who have the most number of followers. If there is a tie, select all such users.

SELECT \* FROM Followers;

**Solution-**

leader (ρT1Followers))T1.leader= maxFollowers.leader

 maxFollowers (leader,num) γleader; MAX(num) 🡪 leader (ρT2Followers))

leader(γleader (Followers))

SELECT leader

FROM Followers

GROUP BY leader

HAVING count(\*)=(

SELECT MAX(num) FROM

(SELECT leader, COUNT(\*) as num

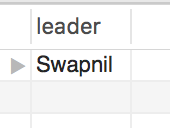
FROM Followers

GROUP BY leader

)AS maxFollowers

);

**Result of the above query**



1. Write a CREATE assertion statement to enforce that a user cannot comment on his/her own blogs.

**Solution-**

CREATE ASSERTION NoSelfComment

CHECK NOT EXISTS(

**SELECT \***

**FROM Comments T1**

**WHERE T1.author**

**NOT IN (**

**SELECT COUNT(\*)**

**FROM Blogs T2**

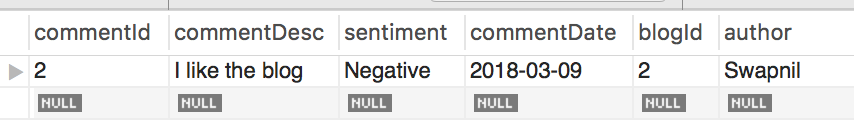
**WHERE T2.postUser=T1.author**

**AND T1.blogId=T2.blogId**

**)**

);

**Result of above inner query(marked in bold)**



1. Write a CREATE assertion statement to enforce that each user can post at most 5 blogs a day,

**Solution-**

CREATE ASSERTION NotMoreThanFiveBlogsPerDay

CHECK NOT EXISTS(

**SELECT postUser**

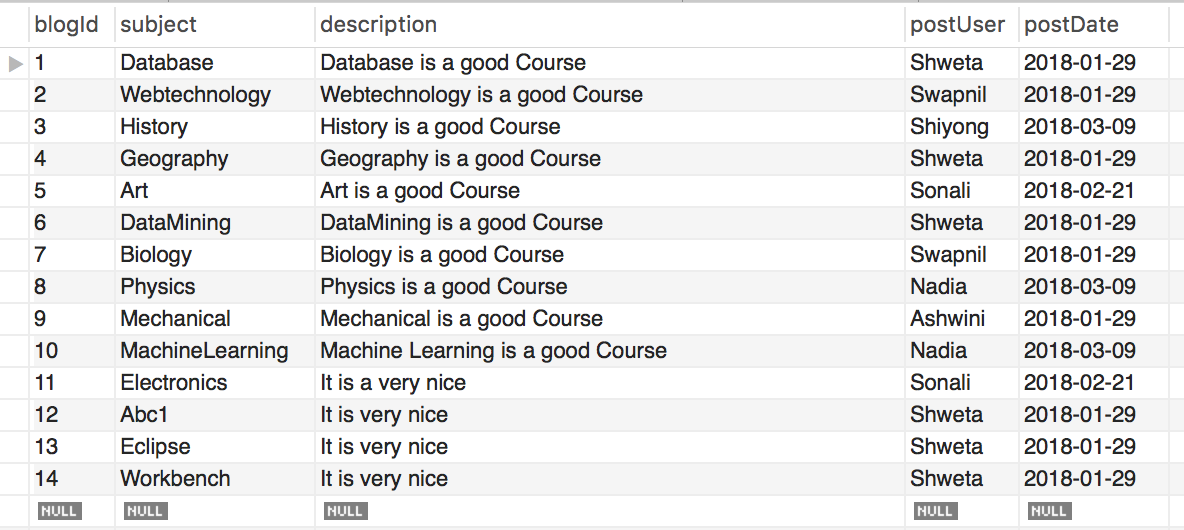
**FROM Blogs**

**GROUP BY postUser,postDate**

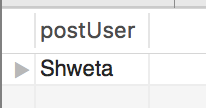
**HAVING COUNT(\*)>5**

);

**Blogs Table- added extra rows to get the result for assertion**



**Result of above inner query(marked in bold)**



1. Write a CREATE assertion statements to enforce that each user can give at most 7 comments in one day.

**Solution-**

CREATE ASSERTION NotMoreThanSevenCommentsPerDay

CHECK NOT EXISTS(

**SELECT author**

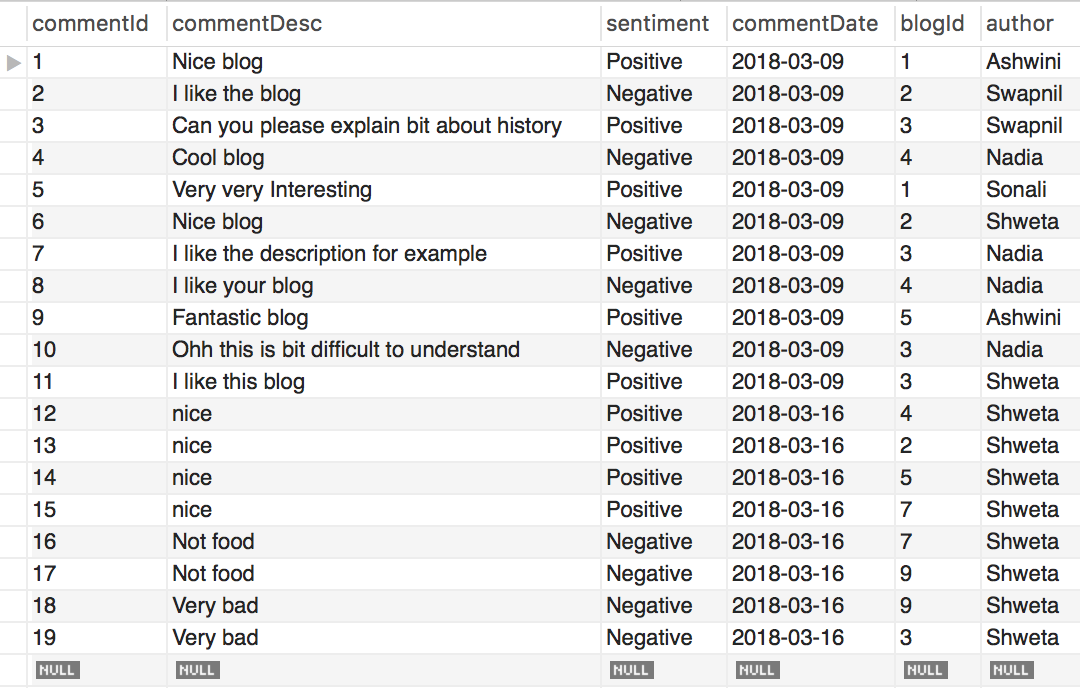
**FROM Comments**

**GROUP BY author,commentDate**

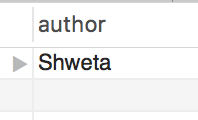
**HAVING COUNT(\*)>7**

);

**Comments Table- added extra rows to get the result for assertion**



**Result of above inner query(marked in bold)**

****

1. Write a CREATE TABLE or CREATE assertion statement to enforce that each user can give at most one comment to each blog.

**Solution-**

CREATE ASSERTION OnlyOneCommentPerBlog

CHECK NOT EXISTS(

**SELECT blogId,author**

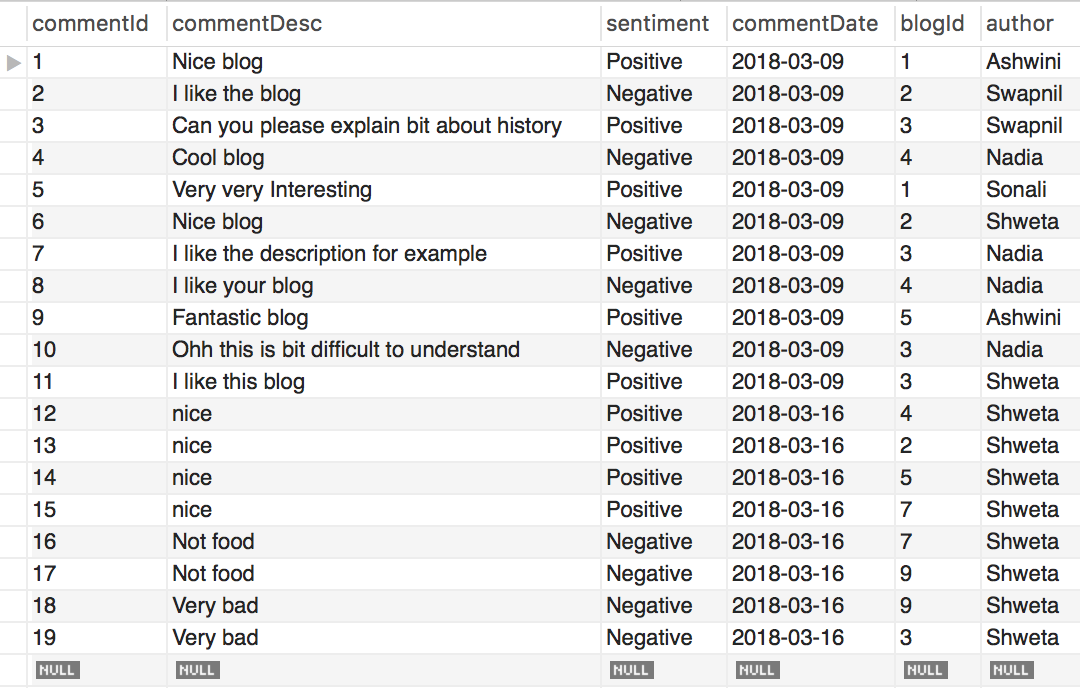
**FROM Comments**

**GROUP BY blogId,author**

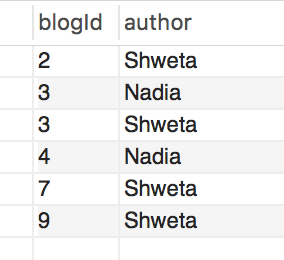
**HAVING COUNT(\*)>1**

);

**Comments Table-**



**Result of above inner query(marked in bold)**



Thank you