Friendships

*** Query: Create 6 new users***

INSERT INTO users(first\_name, last\_name) VALUES ("Amy","Giver");

INSERT INTO users(first\_name, last\_name) VALUES ("Eli","Byears");

INSERT INTO users(first\_name, last\_name) VALUES ("MArky","Mark");

INSERT INTO users(first\_name, last\_name) VALUES ("Kermit","The Frog");

INSERT INTO users(first\_name, last\_name) VALUES ("Big","Bird");

*** Query: Have user 1 be friends with user 2, 4 and 6***

Query: Have user 2 be friends with user 1, 3 and 5

INSERT INTO frienships(user\_id, friend\_id) VALUES (3,2);

INSERT INTO frienships(user\_id, friend\_id) VALUES (3,4);

INSERT INTO frienships(user\_id, friend\_id) VALUES (3,6);

*** Query: Have user 3 be friends with user 2 and 5***

INSERT INTO frienships(user\_id, friend\_id) VALUES (4,3);

INSERT INTO frienships(user\_id, friend\_id) VALUES (4,6);

*** Query: Have user 4 be friends with user 3***

INSERT INTO frienships(user\_id, friend\_id) VALUES (5,4);

** Query: Have user 5 be friends with user 1 and 6**

INSERT INTO frienships(user\_id, friend\_id) VALUES (6,2);

INSERT INTO frienships(user\_id, friend\_id) VALUES (6,8);

*** Query: Have user 6 be friends with user 2 and 3***

INSERT INTO frienships(user\_id, friend\_id) VALUES (8,3);

INSERT INTO frienships(user\_id, friend\_id) VALUES (8,4);

 Query: Display the relationships created as shown in the table in the above image



 NINJA Query: Return all users who are friends with the first user, make sure their names are displayed in results.

 NINJA Query: Return the count of all friendships

 NINJA Query: Find out who has the most friends and return the count of their friends.

 NINJA Query: Return the friends of the third user in alphabetical order

Bottom of Form