

PROJECT REPORT
(SINDHUJA YERRAMALLA, VENKATESH PANGULURI)

1. INTRODUCTION

This project is about designing and implementing a simple Twitter-like social networking platform using both MySQL and Neo4j graph databases. By using this project, the user can create account, can post the tweets, can follow other users, can read their feeds, and like, comment and retweet the feeds.

STEPS FOLLOWED FOR DATABASE CREATION USING MYSQL:

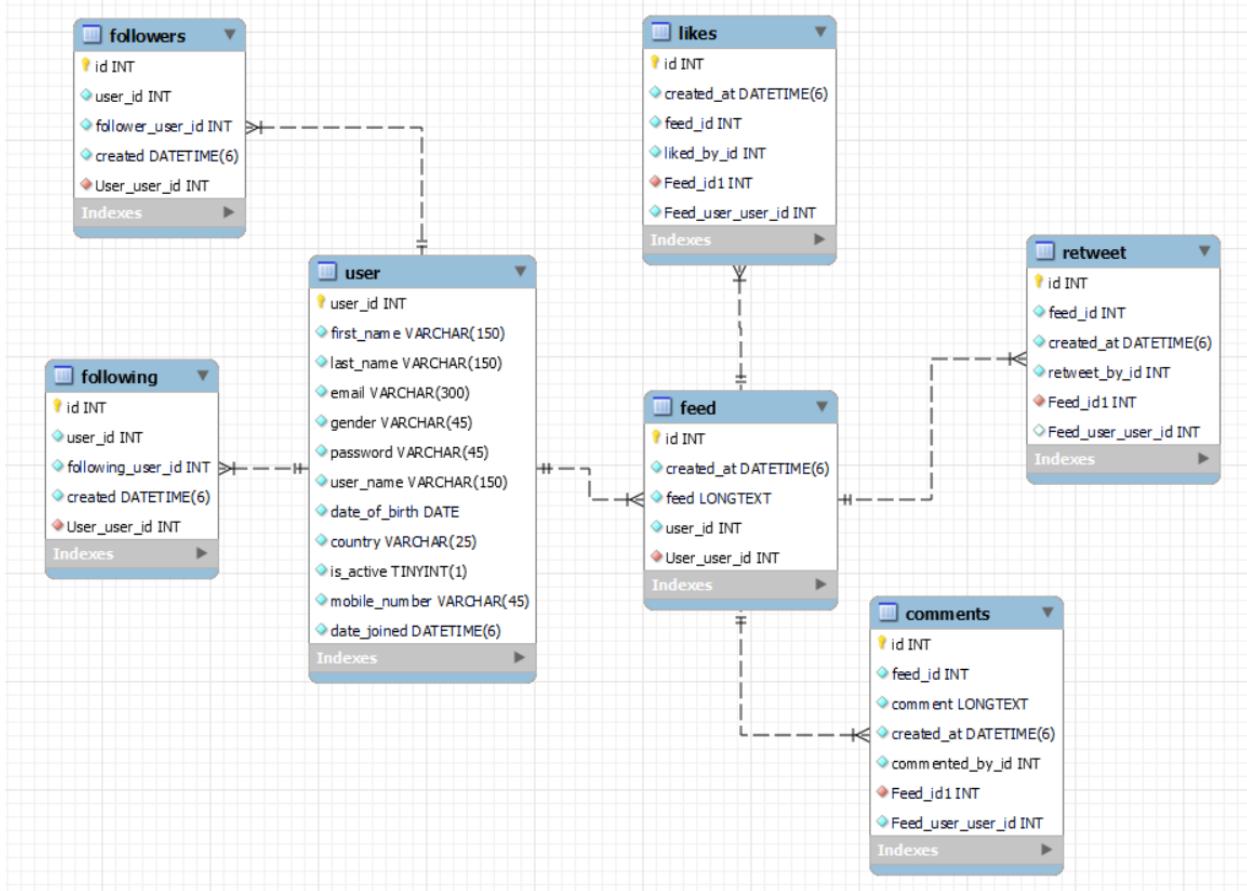
- First, we had designed our ER Diagram in MYSQL workbench with all the tables, columns, and their relations and then we did the Forward Engineering to get the script from the ER Diagram.
- Secondly, we had Inserted the sample data for all the tables.
- Then, tested the database with different types of queries to ensure the database is working properly which includes Insert, update, delete, count operations.
- Finally, we had noted 15 unique use cases and took the screenshots of their output.

STEPS FOLLOWED FOR DATABASE CREATION USING NEO4J:

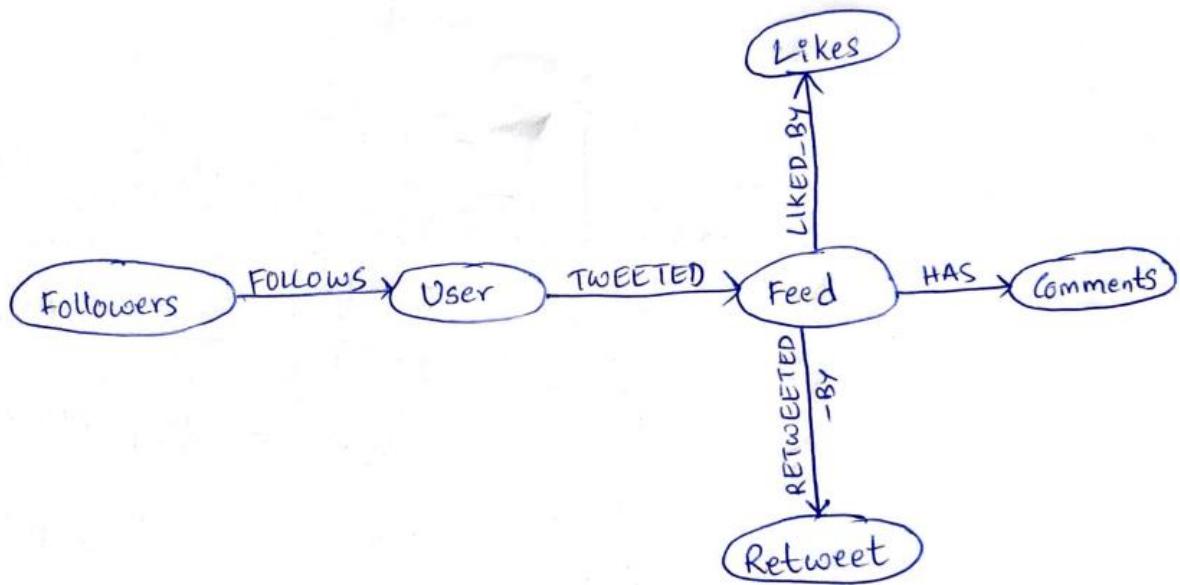
- Using the entities from ER Diagram as reference, we first created Node labels in Neo4j.
- Second step is about inserting data for each label. We made 50 insertions for each label.
- Tested with multiple queries
- Figured out 15 use cases which includes deleting, updating, retrieving, and creating operations in different forms.
- Took output screenshots of every Use case and mentioned below.

2. DATABASE DESIGN

2.1. ER DIAGRAM – MYSQL:



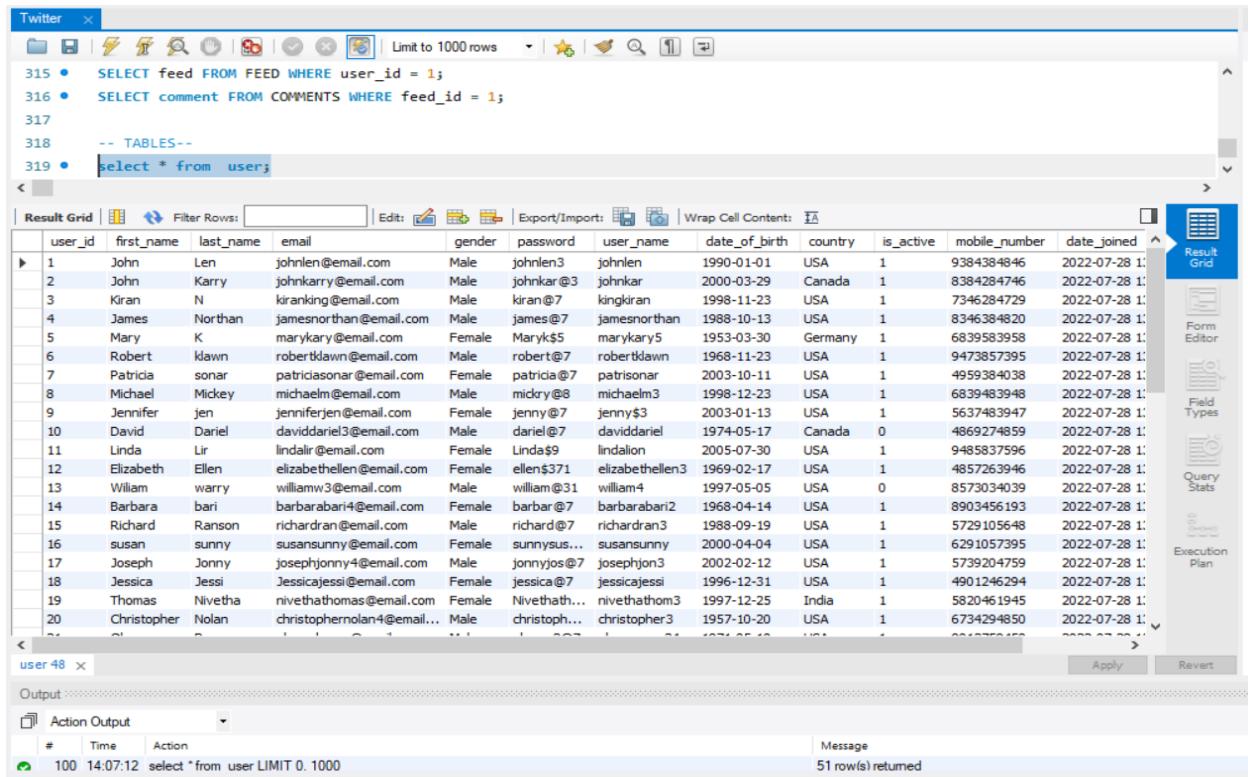
2.2. GRAPH DIAGRAM – Neo4j:



3. PHYSICAL DATABASE (TABLES)

MYSQL – DATABASE:

1. User Table:



The screenshot shows the MySQL Workbench interface with the 'Twitter' database selected. The 'User' table is currently displayed in the result grid. The table has 21 rows and 13 columns. The columns are: user_id, first_name, last_name, email, gender, password, user_name, date_of_birth, country, is_active, mobile_number, and date_joined. The data includes various names like John, Len, Mary, K, Robert, etc., with their corresponding details such as email addresses, genders, and birth dates. The 'is_active' column contains values 1 or 0, and the 'date_joined' column shows dates ranging from 2022-07-28 to 2022-07-29. The 'mobile_number' column contains some numbers starting with 9. The 'user_id' column ranges from 1 to 21. The 'password' column contains hashed passwords. The 'user_name' column contains unique names for each user. The 'date_of_birth' column shows various birth years from 1953 to 1998. The 'country' column shows USA, Canada, and India. The 'is_active' column shows both active (1) and inactive (0) users. The 'mobile_number' column shows some numbers starting with 9. The 'date_joined' column shows dates from 2022-07-28 to 2022-07-29.

user_id	first_name	last_name	email	gender	password	user_name	date_of_birth	country	is_active	mobile_number	date_joined
1	John	Len	johnlen@email.com	Male	johnlen3	johnlen	1990-01-01	USA	1	9384384846	2022-07-28 1:22
2	John	Karry	johnkarry@email.com	Male	johnkar@3	johnkar	2000-03-29	Canada	1	8384284746	2022-07-28 1:22
3	Kiran	N	kiranking@email.com	Male	kiran@7	kingkiran	1998-11-23	USA	1	7346284729	2022-07-28 1:22
4	James	Northan	jamesnorthan@email.com	Male	james@7	jamesnorthan	1988-10-13	USA	1	8346384820	2022-07-28 1:22
5	Mary	K	marykary@email.com	Female	Mary\$5	marykary5	1953-03-30	Germany	1	6839583958	2022-07-28 1:22
6	Robert	klawn	robertklawn@email.com	Male	robert@7	robertklawn	1968-11-23	USA	1	9473857395	2022-07-28 1:22
7	Patricia	sonar	patriciasonar@email.com	Female	patricia@7	patrisonar	2003-10-11	USA	1	4959384038	2022-07-28 1:22
8	Michael	Mickey	michaelm@email.com	Male	midky@8	michaelm3	1998-12-23	USA	1	6839483948	2022-07-28 1:22
9	Jennifer	jen	jenniferjen@email.com	Female	jenny@7	jenny\$3	2003-01-13	USA	1	5637483947	2022-07-28 1:22
10	David	Dariel	daviddariel3@email.com	Male	dariel@7	daviddariel	1974-05-17	Canada	0	4869274859	2022-07-28 1:22
11	Linda	Lir	lindalir@email.com	Female	Linda\$9	lindalir	2005-07-30	USA	1	9485837596	2022-07-28 1:22
12	Elizabeth	Ellen	elizabetthellen@email.com	Female	ellen\$371	elizabetthellen3	1969-02-17	USA	1	4857263946	2022-07-28 1:22
13	Willam	warry	williamw3@email.com	Male	william@31	william4	1997-05-05	USA	0	8573034039	2022-07-28 1:22
14	Barbara	bari	barbarabari4@email.com	Female	barbar@7	barbarabari2	1968-04-14	USA	1	8903456193	2022-07-28 1:22
15	Richard	Ranson	richardran@email.com	Male	richard@7	richardran3	1988-09-19	USA	1	5729105648	2022-07-28 1:22
16	susan	sunny	susansunny@email.com	Female	sunnysus...	susansunny	2000-04-04	USA	1	6291057395	2022-07-28 1:22
17	Joseph	Jonny	josephjony4@email.com	Male	jonnyjos@7	josephjony3	2002-02-12	USA	1	5739204759	2022-07-28 1:22
18	Jessica	Jessi	Jessicajessi@email.com	Female	jessica@7	jessicajessi	1996-12-31	USA	1	4901246294	2022-07-28 1:22
19	Thomas	Nivetha	nivethathomas@email.com	Female	Nivethath...	nivethathom3	1997-12-25	India	1	5820461945	2022-07-28 1:22
20	Christopher	Nolan	christophernolan4@email...	Male	christoph...	christopher3	1957-10-20	USA	1	6734294850	2022-07-28 1:22
21	Gl	o	glo@email.com	Male	gl@7	glo	2007-05-10	USA	1	2012750150	2022-07-28 1:22

Twitter

```

315 • SELECT feed FROM FEED WHERE user_id = 1;
316 • SELECT comment FROM COMMENTS WHERE feed_id = 1;
317
318 -- TABLES--
319 • select * from user;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Apply

user_id	first_name	last_name	email	gender	password	user_name	date_of_birth	country	is_active	mobile_number	date_joined
21	Obama	Barac	obamabarac@email.com	Male	obama3@7	obamaaarac34	1971-05-10	USA	1	9013759450	2022-07-28 13:42:37.000000
22	Nancy	Nerin	nancynerin@email.com	Female	nancymer@7	nancymerin3	1983-07-07	USA	1	6701235185	2022-07-28 13:42:37.000000
23	Amy	Jackson	amyjackson@email.com	Female	amyjack@7	amyjackson3	1970-02-18	USA	1	4859124385	2022-07-28 13:42:37.000000
24	Scott	Fleming	scottflem3@email.com	Male	scottflem@7	scottflem3	1984-06-27	USA	1	9017539546	2022-07-28 13:42:37.000000
25	Catherine	Teresa	catherineteresa@email.c...	Female	catherine@7	catherinetere...	1964-05-02	USA	1	8572934094	2022-07-28 13:42:37.000000
26	mahesh	gattamneni	maheshgattam3@email.c...	Male	maheshsha...	maheshbabu3	1973-10-11	India	1	8572934019	2022-07-28 13:42:37.000000
27	Tom	Crusie	tomcruse@email.com	Male	tomcruse@7	tomcruse5	1965-12-03	USA	1	9013254718	2022-07-28 13:42:37.000000
28	Selena	Gomez	selenagomez23@email.com	Female	selenago...	selenagomez5	1997-10-26	USA	1	9018549230	2022-07-28 13:42:37.000000
29	Christian	Corell	christiancorell3@email.com	Male	christian@7	christiancor3	1985-03-09	USA	1	5481045729	2022-07-28 13:42:37.000000
30	Grace	Garry	gracegarry@email.com	Female	gracegarr...	gracegarry5	2001-09-12	USA	1	9823418495	2022-07-28 13:42:37.000000
31	Jordan	jonath	jordanjonath3@email.com	Male	jordanjon...	jordanjonath1	1963-03-18	USA	1	9018543921	2022-07-28 13:42:37.000000
32	Sophia	Sony	sophiasony@email.com	Female	sophiason...	sophiasony5	1957-07-03	USA	1	4513859420	2022-07-28 13:42:37.000000
33	Albert	Einstein	einstein321@email.com	Male	einstein@7	alberteinstein3	1931-08-24	USA	1	8437591034	2022-07-28 13:42:37.000000
34	Newton	Alison	newtonalison3@email.com	Male	newtonalisi...	newtonalison3	1949-04-10	USA	1	9012396329	2022-07-28 13:42:37.000000
35	Jason	Roy	jasonroy3@email.com	Male	jasonroy@7	jasonroy1	1985-07-10	England	1	9018643026	2022-07-28 13:42:37.000000
36	Bruce	Lee	brucelee3@email.com	Male	brucelee@7	brucelee7	1950-05-21	USA	1	6028569236	2022-07-28 13:42:37.000000
37	Alice	Ally	aliceally3@email.com	Female	aliceally@91	aliceally	1989-04-05	USA	1	9814385492	2022-07-28 13:42:37.000000
38	Julia	Julius	juliajulius@email.com	Female	juliajulius@7	juliajulius4	1974-10-01	USA	1	9018560923	2022-07-28 13:42:37.000000
39	Kayla	Kali	kaylakali3@email.com	Female	kaylakali@7	kaylakali4	1992-03-19	USA	1	9017693740	2022-07-28 13:42:37.000000
40	Gerald	grandy	geraldgrandy3@email.com	Male	geraldgra...	geraldgrandy3	1996-05-03	USA	1	9048575710	2022-07-28 13:42:37.000000
41	Sara	Nara	saranara3@email.com	Female	saranara@7	saranara3	1995-06-29	USA	1	4517395748	2022-07-28 13:42:37.000000
42	Roger	Reign	rogerreign3@email.com	Male	rogerreig...	rogerreign	1978-04-29	USA	1	8475837582	2022-07-28 13:42:37.000000
43	Janice	Janial	janicejanial@email.com	Female	janialjanic...	janialjanice2	1988-12-01	USA	1	9012495472	2022-07-28 13:42:37.000000
44	Diana	Danny	DianaDany@email.com	Female	Dianadan...	dianadiana3	1999-08-25	USA	1	9013842120	2022-07-28 13:42:37.000000
45	Marie	Moza	mariemoza@email.com	Female	mariemoz...	mariemoza2	2000-10-04	USA	1	9016781452	2022-07-28 13:42:37.000000
46	Mason	George	georgemason3@email.com	Male	masongeo...	georgemason3	1993-07-22	USA	1	9815723491	2022-07-28 13:42:37.000000
47	Alexa	Ally	alexaaally3@email.com	Female	alexaaally@7	alexaaally5	1993-03-13	USA	1	6713491548	2022-07-28 13:42:37.000000
48	Logan	Larry	loganlarry5@email.com	Male	loganlarry...	loganlarry3	1992-01-13	USA	1	8329451023	2022-07-28 13:42:37.000000
49	will	smith	willsmith3@email.com	Male	willsmith@7	willsmith3	1963-03-29	USA	1	8710341954	2022-07-28 13:42:37.000000
50	Martha	Maggie	marthamaggie3@email.com	Female	marthama...	marthamaggie2	1977-07-03	USA	1	9012841953	2022-07-28 13:42:37.000000
51	sumanth	suss	sumanthsuss@email.com	Male	sumanth...	sumanth.suss	1998-06-23	USA	1	9015201954	2022-07-28 13:42:37.000000

Twitter

```

315 • SELECT feed FROM FEED WHERE user_id = 1;
316 • SELECT comment FROM COMMENTS WHERE feed_id = 1;
317
318 -- TABLES--
319 • select * from user;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Apply

user_id	first_name	last_name	email	gender	password	user_name	date_of_birth	country	is_active	mobile_number	date_joined
29	Christian	Corell	christiancorell3@email.com	Male	christian@7	christiancor3	1985-03-09	USA	1	5481045729	2022-07-28 13:42:37.000000
30	Grace	Garry	gracegarry@email.com	Female	gracegarr...	gracegarry5	2001-09-12	USA	1	9823418495	2022-07-28 13:42:37.000000
31	Jordan	jonath	jordanjonath3@email.com	Male	jordanjon...	jordanjonath1	1963-03-18	USA	1	9018543921	2022-07-28 13:42:37.000000
32	Sophia	Sony	sophiasony@email.com	Female	sophiason...	sophiasony5	1957-07-03	USA	1	4513859420	2022-07-28 13:42:37.000000
33	Albert	Einstein	einstein321@email.com	Male	einstein@7	alberteinstein3	1931-08-24	USA	1	8437591034	2022-07-28 13:42:37.000000
34	Newton	Alison	newtonalison3@email.com	Male	newtonalisi...	newtonalison3	1949-04-10	USA	1	9012396329	2022-07-28 13:42:37.000000
35	Jason	Roy	jasonroy3@email.com	Male	jasonroy@7	jasonroy1	1985-07-10	England	1	9018643026	2022-07-28 13:42:37.000000
36	Bruce	Lee	brucelee3@email.com	Male	brucelee@7	brucelee7	1950-05-21	USA	1	6028569236	2022-07-28 13:42:37.000000
37	Alice	Ally	aliceally3@email.com	Female	aliceally@91	aliceally	1989-04-05	USA	1	9814385492	2022-07-28 13:42:37.000000
38	Julia	Julius	juliajulius@email.com	Female	juliajulius@7	juliajulius4	1974-10-01	USA	1	9018560923	2022-07-28 13:42:37.000000
39	Kayla	Kali	kaylakali3@email.com	Female	kaylakali@7	kaylakali4	1992-03-19	USA	1	9017693740	2022-07-28 13:42:37.000000
40	Gerald	grandy	geraldgrandy3@email.com	Male	geraldgra...	geraldgrandy3	1996-05-03	USA	1	9048575710	2022-07-28 13:42:37.000000
41	Sara	Nara	saranara3@email.com	Female	saranara@7	saranara3	1995-06-29	USA	1	4517395748	2022-07-28 13:42:37.000000
42	Roger	Reign	rogerreign3@email.com	Male	rogerreig...	rogerreign	1978-04-29	USA	1	8475837582	2022-07-28 13:42:37.000000
43	Janice	Janial	janicejanial@email.com	Female	janialjanic...	janialjanice2	1988-12-01	USA	1	9012495472	2022-07-28 13:42:37.000000
44	Diana	Danny	DianaDany@email.com	Female	Dianadan...	dianadiana3	1999-08-25	USA	1	9013842120	2022-07-28 13:42:37.000000
45	Marie	Moza	mariemoza@email.com	Female	mariemoz...	mariemoza2	2000-10-04	USA	1	9016781452	2022-07-28 13:42:37.000000
46	Mason	George	georgemason3@email.com	Male	masongeo...	georgemason3	1993-07-22	USA	1	9815723491	2022-07-28 13:42:37.000000
47	Alexa	Ally	alexaaally3@email.com	Female	alexaaally@7	alexaaally5	1993-03-13	USA	1	6713491548	2022-07-28 13:42:37.000000
48	Logan	Larry	loganlarry5@email.com	Male	loganlarry...	loganlarry3	1992-01-13	USA	1	8329451023	2022-07-28 13:42:37.000000
49	will	smith	willsmith3@email.com	Male	willsmith@7	willsmith3	1963-03-29	USA	1	8710341954	2022-07-28 13:42:37.000000
50	Martha	Maggie	marthamaggie3@email.com	Female	marthama...	marthamaggie2	1977-07-03	USA	1	9012841953	2022-07-28 13:42:37.000000
51	sumanth	suss	sumanthsuss@email.com	Male	sumanth...	sumanth.suss	1998-06-23	USA	1	9015201954	2022-07-28 13:42:37.000000

2. Feed Table:

Twitter

```
316 •   SELECT comment FROM COMMENTS WHERE feed_id = 1;
317
318 -- TABLES--
319 •   select * from user;
320 •   select * from Feed;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed	user_id	User_user_id
▶	1	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems	1	1
	2	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-2	2	2
	3	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-3	3	3
	4	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-4	4	4
	5	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-5	5	5
	6	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-6	6	6
	7	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-7	7	7
	8	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-8	8	8
	9	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-9	9	9
	10	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-10	10	10
	11	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-11	11	11
	12	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-12	12	12
	13	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-13	13	13
	14	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-14	14	14
	15	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-15	15	15
	16	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-16	16	16
	17	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-17	17	17
	18	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-18	18	18
	19	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-19	19	19
	20	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-20	20	20
	21	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-21	21	21

Feed 1

Output:

#	Time	Action	Message
1	14:20:45	select * from Feed LIMIT 0, 1000	55 row(s) returned

Twitter

```

316 •  SELECT comment FROM COMMENTS WHERE feed_id = 1;
317
318 -- TABLES--
319 •  select * from user;
320 •  select * from Feed;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed	user_id	User_user_id
21	21	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-21	21	21
22	22	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-22	22	22
23	23	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-23	23	23
24	24	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-24	24	24
25	25	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-25	25	25
26	26	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-26	26	26
27	27	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-27	27	27
28	28	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-28	28	28
29	29	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-29	29	29
30	30	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-30	30	30
31	31	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-31	31	31
32	32	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-32	32	32
33	33	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-33	33	33
34	34	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-34	34	34
35	35	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-35	35	35
36	36	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-36	36	36
37	37	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-37	37	37
38	38	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-38	38	38
39	39	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-39	39	39
40	40	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-40	40	40
41	41	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-41	41	41

Feed 1

Output

Action Output

#	Time	Action	Message
1	14:20:45	select * from Feed LIMIT 0, 1000	55 row(s) returned

Twitter

```

316 •  SELECT comment FROM COMMENTS WHERE feed_id = 1;
317
318 -- TABLES--
319 •  select * from user;
320 •  select * from Feed;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed	user_id	User_user_id
37	37	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-37	37	37
38	38	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-38	38	38
39	39	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-39	39	39
40	40	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-40	40	40
41	41	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-41	41	41
42	42	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-42	42	42
43	43	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-43	43	43
44	44	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-44	44	44
45	45	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-45	45	45
46	46	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-46	46	46
47	47	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-47	47	47
48	48	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-48	48	48
49	49	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-49	49	49
50	50	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-50	50	50
51	51	2022-07-28 13:42:37.000000	Hey, How are you?	51	51
52	52	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-52	1	1
53	53	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-53	1	1
54	54	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-54	1	1
55	55	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-55	1	1

Feed 1

Apply

Output

Action Output

#	Time	Action	Message
1	14:20:45	select * from Feed LIMIT 0, 1000	55 row(s) returned

3. Following Table:

Twitter >

```

317
318    -- TABLES--
319 •   select * from user;
320 •   select * from Feed;
321 •   select * from followings;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	user_id	following_user_id	created	User_user_id
▶	1	1	2	2022-07-28 13:42:37.000000	1
	2	2	3	2022-07-28 13:42:37.000000	2
	3	3	4	2022-07-28 13:42:37.000000	3
	4	4	5	2022-07-28 13:42:37.000000	4
	5	5	6	2022-07-28 13:42:37.000000	5
	6	6	7	2022-07-28 13:42:37.000000	6
	7	7	8	2022-07-28 13:42:37.000000	7
	8	8	9	2022-07-28 13:42:37.000000	8
	9	9	10	2022-07-28 13:42:37.000000	9
	10	10	11	2022-07-28 13:42:37.000000	10
	11	11	12	2022-07-28 13:42:37.000000	11
	12	12	13	2022-07-28 13:42:37.000000	12
	13	13	14	2022-07-28 13:42:37.000000	13
	14	14	15	2022-07-28 13:42:37.000000	14
	15	15	16	2022-07-28 13:42:37.000000	15
	16	16	17	2022-07-28 13:42:37.000000	16
	17	17	18	2022-07-28 13:42:37.000000	17
	18	18	19	2022-07-28 13:42:37.000000	18
	19	19	20	2022-07-28 13:42:37.000000	19
	20	20	21	2022-07-28 13:42:37.000000	20
	21	21	22	2022-07-28 13:42:37.000000	21

following 3 >

Output ::

Action Output

#	Time	Action
1	14:24:32	select * from following LIMIT 0, 1000

Message
54 row(s) returned

Twitter >

```

317
318    -- TABLES--
319 •   select * from user;
320 •   select * from Feed;
321 •   select * from followings;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	user_id	following_user_id	created	User_user_id
▶	21	21	22	2022-07-28 13:42:37.000000	21
	22	22	23	2022-07-28 13:42:37.000000	22
	23	23	24	2022-07-28 13:42:37.000000	23
	24	24	25	2022-07-28 13:42:37.000000	24
	25	25	26	2022-07-28 13:42:37.000000	25
	26	26	27	2022-07-28 13:42:37.000000	26
	27	27	28	2022-07-28 13:42:37.000000	27
	28	28	29	2022-07-28 13:42:37.000000	28
	29	29	30	2022-07-28 13:42:37.000000	29
	30	30	31	2022-07-28 13:42:37.000000	30
	31	31	32	2022-07-28 13:42:37.000000	31
	32	32	33	2022-07-28 13:42:37.000000	32
	33	33	34	2022-07-28 13:42:37.000000	33
	34	34	35	2022-07-28 13:42:37.000000	34
	35	35	36	2022-07-28 13:42:37.000000	35
	36	36	37	2022-07-28 13:42:37.000000	36
	37	37	38	2022-07-28 13:42:37.000000	37
	38	38	39	2022-07-28 13:42:37.000000	38
	39	39	40	2022-07-28 13:42:37.000000	39
	40	40	41	2022-07-28 13:42:37.000000	40
	41	41	42	2022-07-28 13:42:37.000000	41

following 3 >

Output ::

Action Output

#	Time	Action
1	14:24:32	select * from following LIMIT 0, 1000

Message
54 row(s) returned

Twitter

```

317
318    -- TABLES--
319 •  select * from user;
320 •  select * from Feed;
321 •  select * from following;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	user_id	following_user_id	created	User_user_id
36	36	37	2022-07-28 13:42:37.000000	36	
37	37	38	2022-07-28 13:42:37.000000	37	
38	38	39	2022-07-28 13:42:37.000000	38	
39	39	40	2022-07-28 13:42:37.000000	39	
40	40	41	2022-07-28 13:42:37.000000	40	
41	41	42	2022-07-28 13:42:37.000000	41	
42	42	43	2022-07-28 13:42:37.000000	42	
43	43	44	2022-07-28 13:42:37.000000	43	
44	44	45	2022-07-28 13:42:37.000000	44	
45	45	46	2022-07-28 13:42:37.000000	45	
46	46	47	2022-07-28 13:42:37.000000	46	
47	47	48	2022-07-28 13:42:37.000000	47	
48	48	49	2022-07-28 13:42:37.000000	48	
49	49	50	2022-07-28 13:42:37.000000	49	
50	50	1	2022-07-28 13:42:37.000000	50	
51	1	3	2022-07-28 13:42:37.000000	1	
52	1	4	2022-07-28 13:42:37.000000	1	
53	1	5	2022-07-28 13:42:37.000000	1	
54	1	6	2022-07-28 13:42:37.000000	1	
*	NUL	NUL	NUL	NUL	NUL

following 3 ×

Output:

Action Output

#	Time	Action
1	14:24:32	select * from following LIMIT 0, 1000

Message: 54 row(s) returned

4. Followers Table:

Twitter

```

318    -- TABLES--
319 •  select * from user;
320 •  select * from Feed;
321 •  select * from following;
322 •  select * from followers;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	user_id	follower_user_id	created	User_user_id
1	1	2	2022-07-28 13:42:37.000000	1	
2	2	3	2022-07-28 13:42:37.000000	2	
3	3	4	2022-07-28 13:42:37.000000	3	
4	4	5	2022-07-28 13:42:37.000000	4	
5	5	6	2022-07-28 13:42:37.000000	5	
6	6	7	2022-07-28 13:42:37.000000	6	
7	7	8	2022-07-28 13:42:37.000000	7	
8	8	9	2022-07-28 13:42:37.000000	8	
9	9	10	2022-07-28 13:42:37.000000	9	
10	10	11	2022-07-28 13:42:37.000000	10	
11	11	12	2022-07-28 13:42:37.000000	11	
12	12	13	2022-07-28 13:42:37.000000	12	
13	13	14	2022-07-28 13:42:37.000000	13	
14	14	15	2022-07-28 13:42:37.000000	14	
15	15	16	2022-07-28 13:42:37.000000	15	
16	16	17	2022-07-28 13:42:37.000000	16	
17	17	18	2022-07-28 13:42:37.000000	17	
18	18	19	2022-07-28 13:42:37.000000	18	
19	19	20	2022-07-28 13:42:37.000000	19	
20	20	21	2022-07-28 13:42:37.000000	20	
21	21	22	2022-07-28 13:42:37.000000	21	

followers 5 ×

Output:

Action Output

#	Time	Action
1	14:27:34	select * from followers LIMIT 0, 1000

Message: 54 row(s) returned

Twitter

```

318 -- TABLES--
319 • select * from user;
320 • select * from Feed;
321 • select * from following;
322 • select * from followers;

```

Result Grid

	id	user_id	follower_user_id	created	User_user_id
21	21	22		2022-07-28 13:42:37.000000	21
22	22	23		2022-07-28 13:42:37.000000	22
23	23	24		2022-07-28 13:42:37.000000	23
24	24	25		2022-07-28 13:42:37.000000	24
25	25	26		2022-07-28 13:42:37.000000	25
26	26	27		2022-07-28 13:42:37.000000	26
27	27	28		2022-07-28 13:42:37.000000	27
28	28	29		2022-07-28 13:42:37.000000	28
29	29	30		2022-07-28 13:42:37.000000	29
30	30	31		2022-07-28 13:42:37.000000	30
31	31	32		2022-07-28 13:42:37.000000	31
32	32	33		2022-07-28 13:42:37.000000	32
33	33	34		2022-07-28 13:42:37.000000	33
34	34	35		2022-07-28 13:42:37.000000	34
35	35	36		2022-07-28 13:42:37.000000	35
36	36	37		2022-07-28 13:42:37.000000	36
37	37	38		2022-07-28 13:42:37.000000	37
38	38	39		2022-07-28 13:42:37.000000	38
39	39	40		2022-07-28 13:42:37.000000	39
40	40	41		2022-07-28 13:42:37.000000	40
41	41	42		2022-07-28 13:42:37.000000	41

followers 5

Output

Action Output

#	Time	Action	Message
1	14:27:34	select * from followers LIMIT 0, 1000	54 row(s) returned

Twitter

```

318 -- TABLES--
319 • select * from user;
320 • select * from Feed;
321 • select * from following;
322 • select * from followers;

```

Result Grid

	id	user_id	follower_user_id	created	User_user_id
36	36	37		2022-07-28 13:42:37.000000	36
37	37	38		2022-07-28 13:42:37.000000	37
38	38	39		2022-07-28 13:42:37.000000	38
39	39	40		2022-07-28 13:42:37.000000	39
40	40	41		2022-07-28 13:42:37.000000	40
41	41	42		2022-07-28 13:42:37.000000	41
42	42	43		2022-07-28 13:42:37.000000	42
43	43	44		2022-07-28 13:42:37.000000	43
44	44	45		2022-07-28 13:42:37.000000	44
45	45	46		2022-07-28 13:42:37.000000	45
46	46	47		2022-07-28 13:42:37.000000	46
47	47	48		2022-07-28 13:42:37.000000	47
48	48	49		2022-07-28 13:42:37.000000	48
49	49	50		2022-07-28 13:42:37.000000	49
50	50	1		2022-07-28 13:42:37.000000	50
51	1	3		2022-07-28 13:42:37.000000	1
52	1	4		2022-07-28 13:42:37.000000	1
53	1	5		2022-07-28 13:42:37.000000	1
54	1	6		2022-07-28 13:42:37.000000	1

followers 5

Output

Action Output

#	Time	Action	Message
1	14:27:34	select * from followers LIMIT 0, 1000	54 row(s) returned

5. Retweet Table:

Twitter ×

```

319 • select * from user;
320 • select * from Feed;
321 • select * from following;
322 • select * from followers;
323 • select * from retweets;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	feed_id	created_at	retweet_by_id	Feed_id1	Feed_user_user_id
▶	1	1	2022-07-28 13:42:37.000000	2	1	1
	2	2	2022-07-28 13:42:37.000000	3	2	2
	3	3	2022-07-28 13:42:37.000000	4	3	3
	4	4	2022-07-28 13:42:37.000000	5	4	4
	5	5	2022-07-28 13:42:37.000000	6	5	5
	6	6	2022-07-28 13:42:37.000000	7	6	6
	7	7	2022-07-28 13:42:37.000000	8	7	7
	8	8	2022-07-28 13:42:37.000000	9	8	8
	9	9	2022-07-28 13:42:37.000000	10	9	9
	10	10	2022-07-28 13:42:37.000000	11	10	10
	11	11	2022-07-28 13:42:37.000000	12	11	11
	12	12	2022-07-28 13:42:37.000000	13	12	12
	13	13	2022-07-28 13:42:37.000000	14	13	13
	14	14	2022-07-28 13:42:37.000000	15	14	14
	15	15	2022-07-28 13:42:37.000000	16	15	15
	16	16	2022-07-28 13:42:37.000000	17	16	16
	17	17	2022-07-28 13:42:37.000000	18	17	17
	18	18	2022-07-28 13:42:37.000000	19	18	18
	19	19	2022-07-28 13:42:37.000000	20	19	19
	20	20	2022-07-28 13:42:37.000000	21	20	20
	21	21	2022-07-28 13:42:37.000000	22	21	21

retweet 6 ×

Output

Action Output

#	Time	Action	Message
1	14:30:57	select * from retweet LIMIT 0, 1000	54 row(s) returned

Twitter ×

```

319 • select * from user;
320 • select * from Feed;
321 • select * from following;
322 • select * from followers;
323 • select * from retweets;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	feed_id	created_at	retweet_by_id	Feed_id1	Feed_user_user_id
▶	21	21	2022-07-28 13:42:37.000000	22	21	21
	22	22	2022-07-28 13:42:37.000000	23	22	22
	23	23	2022-07-28 13:42:37.000000	24	23	23
	24	24	2022-07-28 13:42:37.000000	25	24	24
	25	25	2022-07-28 13:42:37.000000	26	25	25
	26	26	2022-07-28 13:42:37.000000	27	26	26
	27	27	2022-07-28 13:42:37.000000	28	27	27
	28	28	2022-07-28 13:42:37.000000	29	28	28
	29	29	2022-07-28 13:42:37.000000	30	29	29
	30	30	2022-07-28 13:42:37.000000	31	30	30
	31	31	2022-07-28 13:42:37.000000	32	31	31
	32	32	2022-07-28 13:42:37.000000	33	32	32
	33	33	2022-07-28 13:42:37.000000	34	33	33
	34	34	2022-07-28 13:42:37.000000	35	34	34
	35	35	2022-07-28 13:42:37.000000	36	35	35
	36	36	2022-07-28 13:42:37.000000	37	36	36
	37	37	2022-07-28 13:42:37.000000	38	37	37
	38	38	2022-07-28 13:42:37.000000	39	38	38
	39	39	2022-07-28 13:42:37.000000	40	39	39
	40	40	2022-07-28 13:42:37.000000	41	40	40
	41	41	2022-07-28 13:42:37.000000	42	41	41

retweet 6 ×

Output

Action Output

#	Time	Action	Message
1	14:30:57	select * from retweet LIMIT 0, 1000	54 row(s) returned

Twitter

```

319 •   select * from user;
320 •   select * from Feed;
321 •   select * from following;
322 •   select * from followers;
323 •   select * from retweets;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	feed_id	created_at	retweet_by_id	Feed_id1	Feed_user_user_id
36	36	2022-07-28 13:42:37.000000	37	36	36	
37	37	2022-07-28 13:42:37.000000	38	37	37	
38	38	2022-07-28 13:42:37.000000	39	38	38	
39	39	2022-07-28 13:42:37.000000	40	39	39	
40	40	2022-07-28 13:42:37.000000	41	40	40	
41	41	2022-07-28 13:42:37.000000	42	41	41	
42	42	2022-07-28 13:42:37.000000	43	42	42	
43	43	2022-07-28 13:42:37.000000	44	43	43	
44	44	2022-07-28 13:42:37.000000	45	44	44	
45	45	2022-07-28 13:42:37.000000	46	45	45	
46	46	2022-07-28 13:42:37.000000	47	46	46	
47	47	2022-07-28 13:42:37.000000	48	47	47	
48	48	2022-07-28 13:42:37.000000	49	48	48	
49	49	2022-07-28 13:42:37.000000	50	49	49	
50	50	2022-07-28 13:42:37.000000	1	50	50	
51	1	2022-07-28 13:42:37.000000	3	1	1	
52	1	2022-07-28 13:42:37.000000	4	1	1	
53	1	2022-07-28 13:42:37.000000	5	1	1	
54	1	2022-07-28 13:42:37.000000	6	1	1	
*	HULL	HULL	HULL	HULL	HULL	HULL

retweet 6

Output:

Action Output

#	Time	Action
1	14:30:57	select * from retweet LIMIT 0, 1000

Message: 54 row(s) returned

6. Likes Table:

Twitter

```

320 •   select * from Feed;
321 •   select * from following;
322 •   select * from followers;
323 •   select * from retweets;
324 •   select * from likes;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed_id	liked_by_id	Feed_id1	Feed_user_user_id
1	1	2022-07-28 13:42:37.000000	1	2	1	1
2	2	2022-07-28 13:42:37.000000	2	3	2	2
3	3	2022-07-28 13:42:37.000000	3	4	3	3
4	4	2022-07-28 13:42:37.000000	4	5	4	4
5	5	2022-07-28 13:42:37.000000	5	6	5	5
6	6	2022-07-28 13:42:37.000000	6	7	6	6
7	7	2022-07-28 13:42:37.000000	7	8	7	7
8	8	2022-07-28 13:42:37.000000	8	9	8	8
9	9	2022-07-28 13:42:37.000000	9	10	9	9
10	10	2022-07-28 13:42:37.000000	10	11	10	10
11	11	2022-07-28 13:42:37.000000	11	12	11	11
12	12	2022-07-28 13:42:37.000000	12	13	12	12
13	13	2022-07-28 13:42:37.000000	13	14	13	13
14	14	2022-07-28 13:42:37.000000	14	15	14	14
15	15	2022-07-28 13:42:37.000000	15	16	15	15
16	16	2022-07-28 13:42:37.000000	16	17	16	16
17	17	2022-07-28 13:42:37.000000	17	18	17	17
18	18	2022-07-28 13:42:37.000000	18	19	18	18
19	19	2022-07-28 13:42:37.000000	19	20	19	19
20	20	2022-07-28 13:42:37.000000	20	21	20	20
21	21	2022-07-28 13:42:37.000000	21	22	21	21
*	HULL	HULL	HULL	HULL	HULL	HULL

likes 7

Output:

Action Output

#	Time	Action
1	14:32:53	select * from likes LIMIT 0, 1000

Message: 54 row(s) returned

Twitter

```

320 • select * from Feed;
321 • select * from following;
322 • select * from followers;
323 • select * from retweet;
324 • select * from likes;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed_id	liked_by_id	Feed_id1	Feed_user_user_id
21	21	2022-07-28 13:42:37.000000	21	22	21	21
22	22	2022-07-28 13:42:37.000000	22	23	22	22
23	23	2022-07-28 13:42:37.000000	23	24	23	23
24	24	2022-07-28 13:42:37.000000	24	25	24	24
25	25	2022-07-28 13:42:37.000000	25	26	25	25
26	26	2022-07-28 13:42:37.000000	26	27	26	26
27	27	2022-07-28 13:42:37.000000	27	28	27	27
28	28	2022-07-28 13:42:37.000000	28	29	28	28
29	29	2022-07-28 13:42:37.000000	29	30	29	29
30	30	2022-07-28 13:42:37.000000	30	31	30	30
31	31	2022-07-28 13:42:37.000000	31	32	31	31
32	32	2022-07-28 13:42:37.000000	32	33	32	32
33	33	2022-07-28 13:42:37.000000	33	34	33	33
34	34	2022-07-28 13:42:37.000000	34	35	34	34
35	35	2022-07-28 13:42:37.000000	35	36	35	35
36	36	2022-07-28 13:42:37.000000	36	37	36	36
37	37	2022-07-28 13:42:37.000000	37	38	37	37
38	38	2022-07-28 13:42:37.000000	38	39	38	38
39	39	2022-07-28 13:42:37.000000	39	40	39	39
40	40	2022-07-28 13:42:37.000000	40	41	40	40
41	41	2022-07-28 13:42:37.000000	41	42	41	41

likes 7

Output

Action Output

#	Time	Action	Message
1	14:32:53	select * from likes LIMIT 0, 1000	54 row(s) returned

Twitter

```

320 • select * from Feed;
321 • select * from following;
322 • select * from followers;
323 • select * from retweet;
324 • select * from likes;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed_id	liked_by_id	Feed_id1	Feed_user_user_id
36	36	2022-07-28 13:42:37.000000	36	37	36	36
37	37	2022-07-28 13:42:37.000000	37	38	37	37
38	38	2022-07-28 13:42:37.000000	38	39	38	38
39	39	2022-07-28 13:42:37.000000	39	40	39	39
40	40	2022-07-28 13:42:37.000000	40	41	40	40
41	41	2022-07-28 13:42:37.000000	41	42	41	41
42	42	2022-07-28 13:42:37.000000	42	43	42	42
43	43	2022-07-28 13:42:37.000000	43	44	43	43
44	44	2022-07-28 13:42:37.000000	44	45	44	44
45	45	2022-07-28 13:42:37.000000	45	46	45	45
46	46	2022-07-28 13:42:37.000000	46	47	46	46
47	47	2022-07-28 13:42:37.000000	47	48	47	47
48	48	2022-07-28 13:42:37.000000	48	49	48	48
49	49	2022-07-28 13:42:37.000000	49	50	49	49
50	50	2022-07-28 13:42:37.000000	50	1	50	50
51	51	2022-07-28 13:42:37.000000	1	3	1	1
52	52	2022-07-28 13:42:37.000000	1	4	1	1
53	53	2022-07-28 13:42:37.000000	1	5	1	1
54	54	2022-07-28 13:42:37.000000	1	6	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL

likes 7

Output

Action Output

#	Time	Action	Message
1	14:32:53	select * from likes LIMIT 0, 1000	54 row(s) returned

7. Comments Table:

Twitter ×

```

321 •   select * from following;
322 •   select * from followers;
323 •   select * from retweet;
324 •   select * from likes;
325 •   select * from comments;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	feed_id	comment	created_at	commented_by_id	Feed_id1	Feed_user_user_id
▶	1	1	Hello, Good Morning	2022-07-28 13:42:37.000000	2	1	1
	2	2	Hello, Good Morning-2	2022-07-28 13:42:37.000000	3	2	2
	3	3	Hello, Good Morning-3	2022-07-28 13:42:37.000000	4	3	3
	4	4	Hello, Good Morning-4	2022-07-28 13:42:37.000000	5	4	4
	5	5	Hello, Good Morning-5	2022-07-28 13:42:37.000000	6	5	5
	6	6	Hello, Good Morning-6	2022-07-28 13:42:37.000000	7	6	6
	7	7	Hello, Good Morning-7	2022-07-28 13:42:37.000000	8	7	7
	8	8	Hello, Good Morning-8	2022-07-28 13:42:37.000000	9	8	8
	9	9	Hello, Good Morning-9	2022-07-28 13:42:37.000000	10	9	9
	10	10	Hello, Good Morning-10	2022-07-28 13:42:37.000000	11	10	10
	11	11	Hello, Good Morning-11	2022-07-28 13:42:37.000000	12	11	11
	12	12	Hello, Good Morning-12	2022-07-28 13:42:37.000000	13	12	12
	13	13	Hello, Good Morning-13	2022-07-28 13:42:37.000000	14	13	13
	14	14	Hello, Good Morning-14	2022-07-28 13:42:37.000000	15	14	14
	15	15	Hello, Good Morning-15	2022-07-28 13:42:37.000000	16	15	15
	16	16	Hello, Good Morning-16	2022-07-28 13:42:37.000000	17	16	16
	17	17	Hello, Good Morning-17	2022-07-28 13:42:37.000000	18	17	17
	18	18	Hello, Good Morning-18	2022-07-28 13:42:37.000000	19	18	18
	19	19	Hello, Good Morning-19	2022-07-28 13:42:37.000000	20	19	19
	20	20	Hello, Good Morning-20	2022-07-28 13:42:37.000000	21	20	20
	21	21	Hello, Good Morning-21	2022-07-28 13:42:37.000000	22	21	21

comments 8 ×

Output

Action Output

#	Time	Action	Message
1	14:34:56	select * from comments LIMIT 0, 1000	54 row(s) returned

Twitter ×

```

321 •   select * from following;
322 •   select * from followers;
323 •   select * from retweet;
324 •   select * from likes;
325 •   select * from comments;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	feed_id	comment	created_at	commented_by_id	Feed_id1	Feed_user_user_id
▶	21	21	Hello, Good Morning-21	2022-07-28 13:42:37.000000	22	21	21
	22	22	Hello, Good Morning-22	2022-07-28 13:42:37.000000	23	22	22
	23	23	Hello, Good Morning-23	2022-07-28 13:42:37.000000	24	23	23
	24	24	Hello, Good Morning-24	2022-07-28 13:42:37.000000	25	24	24
	25	25	Hello, Good Morning-25	2022-07-28 13:42:37.000000	26	25	25
	26	26	Hello, Good Morning-26	2022-07-28 13:42:37.000000	27	26	26
	27	27	Hello, Good Morning-27	2022-07-28 13:42:37.000000	28	27	27
	28	28	Hello, Good Morning-28	2022-07-28 13:42:37.000000	29	28	28
	29	29	Hello, Good Morning-29	2022-07-28 13:42:37.000000	30	29	29
	30	30	Hello, Good Morning-30	2022-07-28 13:42:37.000000	31	30	30
	31	31	Hello, Good Morning-31	2022-07-28 13:42:37.000000	32	31	31
	32	32	Hello, Good Morning-32	2022-07-28 13:42:37.000000	33	32	32
	33	33	Hello, Good Morning-33	2022-07-28 13:42:37.000000	34	33	33
	34	34	Hello, Good Morning-34	2022-07-28 13:42:37.000000	35	34	34
	35	35	Hello, Good Morning-35	2022-07-28 13:42:37.000000	36	35	35
	36	36	Hello, Good Morning-36	2022-07-28 13:42:37.000000	37	36	36
	37	37	Hello, Good Morning-37	2022-07-28 13:42:37.000000	38	37	37
	38	38	Hello, Good Morning-38	2022-07-28 13:42:37.000000	39	38	38
	39	39	Hello, Good Morning-39	2022-07-28 13:42:37.000000	40	39	39
	40	40	Hello, Good Morning-40	2022-07-28 13:42:37.000000	41	40	40
	41	41	Hello, Good Morning-41	2022-07-28 13:42:37.000000	42	41	41

comments 8 ×

Output

Action Output

#	Time	Action	Message
1	14:34:56	select * from comments LIMIT 0, 1000	54 row(s) returned

Twitter

```

321 •   select * from following;
322 •   select * from followers;
323 •   select * from retweet;
324 •   select * from likes;
325 •   select * from comments;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	feed_id	comment	created_at	commented_by_id	Feed_id1	Feed_user_user_id
▶	36	36	Hello, Good Morning-36	2022-07-28 13:42:37.000000	37	36	36
	37	37	Hello, Good Morning-37	2022-07-28 13:42:37.000000	38	37	37
	38	38	Hello, Good Morning-38	2022-07-28 13:42:37.000000	39	38	38
	39	39	Hello, Good Morning-39	2022-07-28 13:42:37.000000	40	39	39
	40	40	Hello, Good Morning-40	2022-07-28 13:42:37.000000	41	40	40
	41	41	Hello, Good Morning-41	2022-07-28 13:42:37.000000	42	41	41
	42	42	Hello, Good Morning-42	2022-07-28 13:42:37.000000	43	42	42
	43	43	Hello, Good Morning-43	2022-07-28 13:42:37.000000	44	43	43
	44	44	Hello, Good Morning-44	2022-07-28 13:42:37.000000	45	44	44
	45	45	Hello, Good Morning-45	2022-07-28 13:42:37.000000	46	45	45
	46	46	Hello, Good Morning-46	2022-07-28 13:42:37.000000	47	46	46
	47	47	Hello, Good Morning-47	2022-07-28 13:42:37.000000	48	47	47
	48	48	Hello, Good Morning-48	2022-07-28 13:42:37.000000	49	48	48
	49	49	Hello, Good Morning-49	2022-07-28 13:42:37.000000	50	49	49
	50	50	Hello, Good Morning-50	2022-07-28 13:42:37.000000	1	50	50
	51	1	Hello, Good Morning-51	2022-07-28 13:42:37.000000	3	1	1
	52	1	Hello, Good Morning-52	2022-07-28 13:42:37.000000	4	1	1
	53	1	Hello, Good Morning-53	2022-07-28 13:42:37.000000	5	1	1
	54	1	Hello, Good Morning-54	2022-07-28 13:42:37.000000	6	1	1
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

comments 8 ×

Output

Action Output

#	Time	Action	Message
1	14:34:56	select * from comments LIMIT 0, 1000	54 row(s) returned

Neo4j Tables:

neo4j@bolt://localhost:7687/twitter - Neo4j Browser

File Edit View Window Help Developer

Database Information

Use database: twitter

Node Labels: *300 Comments Feed Followers Likes Retweet User

Relationship Types: *251 FOLLOWS HAS LIKED_BY RETWEETED_BY TWEETED

twitter\$

```
twitter$ CREATE (u1:User{ user_id:'u1',first_name:'Sindhuja',last_name:'Yerram... 
```

Added 300 labels, created 300 nodes, set 1450 properties, created 251 relationships, completed after 734 ms.

Table

Code

Added 300 labels, created 300 nodes, set 1450 properties, created 251 relationships, completed after 734 ms.

4. USE CASES

MYSQL QUERIES:

1. REGISTRATION QUERY: By using this query, the new user can register into our DB.

```
INSERT INTO Twitter.user (user_id, first_name, last_name, email, gender, password, user_name, date_of_birth, country, is_active, mobile_number, date_joined)
VALUES(52, 'siva', 'p', 'sivap@email.com', 'Male', 'siva@137', 'siva.p', '2000-03-26', 'USA', 1,
9015439210, '2022-07-28 13:42:37');
```

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Twitter'. The code entered is the registration query. Below the code, the results grid displays a table of user data with 52 rows. The columns are: user_id, first_name, last_name, email, gender, password, user_name, date_of_birth, country, is_active, mobile_number, and date_joined. The 'user' table has 52 rows. At the bottom, the 'Action Output' pane shows two log entries: one for the insert operation and another for the select query.

user_id	first_name	last_name	email	gender	password	user_name	date_of_birth	country	is_active	mobile_number	date_joined
47	Alexa	Ally	alexally3@email.com	Female	alexally@7	alexally5	1993-03-13	USA	1	6713491548	2022-07-28 13:42:37.000000
48	Logan	Larry	loganlarry5@email.com	Male	loganlarry...	loganlarry3	1992-01-13	USA	1	8329451023	2022-07-28 13:42:37.000000
49	will	smith	willsmith3@email.com	Male	willsmith@7	willsmith3	1963-03-29	USA	1	8710341954	2022-07-28 13:42:37.000000
50	Martha	Maggie	marthamaggie3@email.com	Female	marthamaggie...	marthamaggie2	1977-07-03	USA	1	9012841953	2022-07-28 13:42:37.000000
51	sumanth	suss	sumanthsuss@email.com	Male	sumanth...	sumanth.suss	1998-06-23	USA	1	9015201954	2022-07-28 13:42:37.000000
52	siva	p	sivap@email.com	Male	siva@137	siva.p	2000-03-26	USA	1	9015439210	2022-07-28 13:42:37.000000
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

2. LOGIN QUERY: By using this query, the user can login into his/her account.

```
SELECT CASE WHEN COUNT(*) > 0 THEN 1 ELSE 0
```

```
END STATUS
```

```
FROM USER WHERE user_name = 'siva.p' AND password = 'siva@137';
```

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Twitter'. The code entered is the login query, which uses a CASE statement to check if the user exists and returns 1 or 0. Below the code, the results grid shows a single row with the value '1'. The 'Action Output' pane at the bottom shows the log entry for the query execution.

STATUS
1

3. GETTING USER PROFILE: By using this query, we can get the user profile details for specific username of the user.

```
SELECT * FROM USER WHERE user_name = 'siva.p';
```

The screenshot shows the MySQL Workbench interface with a query editor titled "Twitter". The code executed is:

```
261    FROM USER WHERE user_name = 'siva.p' AND password = 'siva@137';
262
263 -- 3 GETTING USER PROFILE--
264 •  SELECT * FROM USER WHERE user_name = 'siva.p';
265
```

The result grid displays the following data:

user_id	first_name	last_name	email	gender	password	user_name	date_of_birth	country	is_active	mobile_number	date_joined
52	siva	p	sivap@gmail.com	Male	siva@137	siva.p	2000-03-26	USA	1	9015439210	2022-07-28 13:42:37.000000
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

The screenshot shows the MySQL Workbench interface with a query editor titled "USER 17". The output pane shows the following log entry:

```
Action Output
# Time Action
1 16:18:55 SELECT * FROM USER WHERE user_name = 'siva.p' LIMIT 0, 1000
Message
1 row(s) returned
```

4. ADDING A FEED: By using this query, we can post a new feed or tweet.

```
INSERT INTO Twitter.Feed (id, created_at, feed, user_id, User_user_id)
VALUES(56, '2022-07-28 13:42:37', 'Hello, I am Siva!', 52, 52);
```

The screenshot shows the MySQL Workbench interface with a query editor titled "Twitter". The code executed is:

```
266 -- 4 ADDING A FEED--
267 •  INSERT INTO Twitter.Feed (id, created_at, feed, user_id, User_user_id)
VALUES(56, '2022-07-28 13:42:37', 'Hello, I am Siva!', 52, 52);
268
269
270 •  select * from Feed;
```

The result grid displays the following data:

id	created_at	feed	user_id	User_user_id
51	2022-07-28 13:42:37.000000	Hey, How are you?	51	51
52	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-52	1	1
53	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-53	1	1
54	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-54	1	1
55	2022-07-28 13:42:37.000000	Hello, Welcome to Database systems-55	1	1
56	2022-07-28 13:42:37.000000	Hello, I am Siva!	52	52
*	HULL	HULL	HULL	HULL

The screenshot shows the MySQL Workbench interface with a query editor titled "Feed 19". The output pane shows the following log entries:

```
Action Output
# Time Action
1 16:22:15 INSERT INTO Twitter.Feed (id, created_at, feed, user_id, User_user_id) VALUES(56, '2022-07-28 13:42:37', 'Hello, I am Siva...', 1 row(s) affected
2 16:22:15 select * from Feed LIMIT 0, 1000
Message
56 row(s) returned
```

5. GETTING ALL FEEDS OF A SPECIFIC USER: By using this query, we can get all the feeds of a particular user at once.

```
SELECT * FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p');
```

Twitter

```
270 •    select * from Feed;
271
272      -- 5 GETTING ALL FEEDS OF A SPECIFIC USER--
273 •    SELECT * FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p');
274
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	created_at	feed	user_id	User_user_id
▶	56	2022-07-28 13:42:37.000000	Hello, I am Siva!	52	52
*	HULL	HULL	HULL	HULL	HULL

Feed 20

```
Output ::::::::::::
```

Action Output

#	Time	Action	Message
1	16:24:36	SELECT * FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p') LIMIT 0, 1000	1 row(s) returned

OR

```
SELECT feed FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p');
```

Twitter

```
271
272      -- 5 GETTING ALL FEEDS OF A SPECIFIC USER--
273 •    SELECT * FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p');
274      -- OR
275 •    SELECT feed FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

feed
Hello, I am Siva!

Feed 21

```
Output ::::::::::::
```

Action Output

#	Time	Action	Message
1	16:24:36	SELECT * FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p') LIMIT 0, 1000	1 row(s) returned
2	16:27:45	SELECT feed FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p') LIMIT 0, 1000	1 row(s) returned

6. UPDATING A FEED: By using this query, we can update any feed of any user with the help of the feed id.

UPDATE Feed SET feed = 'Hello all, I am Siva, How are you?' WHERE id = 56;

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query editor contains the following code:

```
276
277 -- 6 UPDATING A FEED --
278 • UPDATE Feed SET feed = 'Hello all, I am Siva, How are you?' WHERE id = 56;
279
280 • SELECT * from Feed where id = 56;
```

The results grid shows the updated feed record:

id	created_at	feed	user_id	User_user_id
56	2022-07-28 13:42:37.000000	Hello all, I am Siva, How are you?	52	52
•	NULL	NULL	NULL	NULL

The output pane shows the execution log:

#	Time	Action	Message
1	17:26:13	UPDATE Feed SET feed = 'Hello all, I am Siva, How are you?' WHERE id = 56	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
2	17:26:14	SELECT * from Feed where id = 56 LIMIT 0, 1000	1 row(s) returned

7. DELETING A FEED: By using this query, we can delete any feed of any user with the help of the feed id.

DELETE FROM Feed WHERE id = 51;

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query editor contains the following code:

```
S 274 -- OR--
C 275 • SELECT feed FROM Feed WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'siva.p');
I 276
Z 277 -- 6 UPDATING A FEED --
K 278 • UPDATE Feed SET feed = 'Hello all, I am Siva, How are you?' WHERE id = 56;
n 279
J 280 • SELECT * from Feed where id = 56;
D 281
L 282 -- 7 DELETING A FEED --
I 283 • DELETE FROM Feed WHERE id = 51;
C 284
D 285 -- 8 ADDING LIKE TO A FEED
R 286 • INSERT INTO Twitter.likes (id, created_at, feed_id, liked_by_id, Feed_id1, Feed_user_user_id)
r 287 VALUES(51, '2022-07-28 13:42:37', 51, 1, 51, 51);
?RR
```

The output pane shows the execution log:

#	Time	Action	Message
1	17:28:09	DELETE FROM Feed WHERE id = 51	1 row(s) affected

8. ADDING LIKE TO A FEED: By using this query, we can add a like to any feed.

```
INSERT INTO Twitter.likes (id, created_at, feed_id, liked_by_id, Feed_id1, Feed_user_user_id)
VALUES(55, '2022-07-28 13:42:37', 56, 1, 56, 52);
```

The screenshot shows the MySQL Workbench interface with two panes. The top pane is titled 'Twitter' and contains the SQL query: `INSERT INTO Twitter.likes (id, created_at, feed_id, liked_by_id, Feed_id1, Feed_user_user_id) VALUES(55, '2022-07-28 13:42:37', 56, 1, 56, 52);`. The bottom pane is titled 'Result Grid' and displays the data inserted into the 'LIKES' table:

id	created_at	feed_id	liked_by_id	Feed_id1	Feed_user_user_id
50	2022-07-28 13:42:37.000000	50	1	50	50
51	2022-07-28 13:42:37.000000	1	3	1	1
52	2022-07-28 13:42:37.000000	1	4	1	1
53	2022-07-28 13:42:37.000000	1	5	1	1
54	2022-07-28 13:42:37.000000	1	6	1	1
55	2022-07-28 13:42:37.000000	56	1	56	52
*	HULL	HULL	HULL	HULL	HULL

The bottom pane also shows the 'Action Output' log with two entries:

#	Time	Action	Message
1	17:31:40	INSERT INTO Twitter.likes (id, created_at, feed_id, liked_by_id, Feed_id1, Feed_user_user_id) VALUES(55, '2022-07-28 13:42:37.000000', 56, 1, 56, 52)	1 row(s) affected
2	17:31:40	SELECT * FROM LIKES LIMIT 0, 1000	55 row(s) returned

9. GETTING TOTAL NUMBER OF COMMENTS FOR A SPECIFIC FEED : By using this query, we can get the total number of comments for a specific feed.

```
SELECT COUNT(*) FROM Comments WHERE feed_id = 1;
```

The screenshot shows the MySQL Workbench interface with two panes. The top pane is titled 'Twitter' and contains the SQL query: `SELECT COUNT(*) FROM Comments WHERE feed_id = 1;`. The bottom pane is titled 'Result Grid' and displays the result of the query:

COUNT(*)
5

The bottom pane also shows the 'Action Output' log with one entry:

#	Time	Action	Message
1	20:45:56	SELECT COUNT(*) FROM Comments WHERE feed_id = 1 LIMIT 0, 1000	1 row(s) returned

10. GETTING TOTAL NUMBER OF LIKES FOR A SPECIFIC FEED: By using this query, we can get the total number of likes for a specific feed.

```
SELECT COUNT(*) FROM likes WHERE feed_id = 1;
```

The screenshot shows the MySQL Workbench interface. The query editor window titled 'Twitter' contains the following code:

```
294 -- 10 GETTING TOTAL NUMBER OF LIKES FOR A SPECIFIC FEED--
295 • SELECT COUNT(*) FROM likes WHERE feed_id = 1;
296
```

The result grid shows a single row with the value 5.

The screenshot shows the MySQL Workbench interface. The query editor window titled 'Result 30' contains the following code:

```
Output :: Action Output
# Time Action
1 21:29:13 SELECT COUNT(*) FROM likes WHERE feed_id = 1 LIMIT 0, 1000
```

The message pane indicates "1 row(s) returned".

11. GETTING TOTAL NUMBER OF RETWEETS FOR A SPECIFIC FEED: By using this query, we can get the total number of retweets for a specific feed by using the feed id.

```
SELECT COUNT(*) FROM Retweet WHERE feed_id = 1;
```

The screenshot shows the MySQL Workbench interface. The query editor window titled 'Twitter' contains the following code:

```
297 -- 11 GETTING TOTAL NUMBER OF RETWEETS FOR A SPECIFIC TWEET(FEED)--
298 • SELECT COUNT(*) FROM Retweet WHERE feed_id = 1;
299
```

The result grid shows a single row with the value 5.

The screenshot shows the MySQL Workbench interface. The query editor window titled 'Result 31' contains the following code:

```
Output :: Action Output
# Time Action
1 21:30:54 SELECT COUNT(*) FROM Retweet WHERE feed_id = 1 LIMIT 0, 1000
```

The message pane indicates "1 row(s) returned".

12. GETTING FOLLOWER DETAILS:

We can get all the follower details
SELECT * FROM Twitter.user WHERE user_id in (SELECT follower_user_id FROM Followers WHERE user_id = 1);

The screenshot shows the MySQL Workbench interface with a connection named 'Twitter'. A query window contains the following code:

```
298 •  SELECT COUNT(*) FROM Retweet WHERE feed_id = 1;
299
300 -- 12 GETTING FOLLOWER DETAILS--
301 •  SELECT * FROM Twitter.user WHERE user_id in (SELECT follower_user_id FROM Followers WHERE user_id = 1);
302
```

The result grid shows the following data:

user_id	first_name	last_name	email	gender	password	user_name	date_of_birth	country	is_active	mobile_number	date_joined
2	John	Karry	johnkarry@email.com	Male	johnkar@3	johnkar	2000-03-29	Canada	1	8384284746	2022-07-28 13:42:37.000000
3	Kiran	N	kiranking@email.com	Male	kiran@7	kingkiran	1998-11-23	USA	1	7346284729	2022-07-28 13:42:37.000000
4	James	Northan	jamesnorthan@email.com	Male	james@7	jamesnorthan	1988-10-13	USA	1	8346384820	2022-07-28 13:42:37.000000
5	Mary	K	marykary@email.com	Female	Maryk\$5	marykary5	1953-03-30	Germany	1	6839583958	2022-07-28 13:42:37.000000
6	Robert	klawn	robertklawn@email.com	Male	robert@7	robertklawn	1968-11-23	USA	1	9473857395	2022-07-28 13:42:37.000000
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

The screenshot shows the MySQL Workbench interface with a connection named 'user 34'. An output window displays the following message:

```
Output:
Action Output
# Time Action
1 21:46:39 SELECT * FROM Twitter.user WHERE user_id in (SELECT follower_user_id FROM Followers WHERE user_id = 1) LIMIT 0, ... 5 row(s) returned
```

13. GETTING TOTAL NUMBER OF FOLLOWERS FOR A SPECIFIC USER:

We can get the total number of followers for a specific user by using the user id or username.
SELECT COUNT(*) FROM Followers WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'johnlen');

The screenshot shows the MySQL Workbench interface with a connection named 'Twitter'. A query window contains the following code:

```
302
303 -- 13 GETTING TOTAL NUMBER OF FOLLOWERS FOR A SPECIFIC USER--
304 •  SELECT COUNT(*) FROM Followers WHERE user_id = 1;
305 -- OR
306 •  SELECT COUNT(*) FROM Followers WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'johnlen');
```

The result grid shows the following data:

COUNT(*)
5

The screenshot shows the MySQL Workbench interface with a connection named 'Result 35'. An output window displays the following message:

```
Output:
Action Output
# Time Action
1 21:49:19 SELECT COUNT(*) FROM Followers WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'johnlen') LIMI... 1 row(s) returned
```

14. DELETING A LIKE FOR A SPECIFIC FEED: By using this query, we can delete a like for a specific feed by using the id of that like.

```
DELETE FROM likes WHERE id = 41;
```

The screenshot shows the MySQL Workbench interface with a connection named 'Twitter'. The SQL editor contains the following code:

```
306 •  SELECT COUNT(*) FROM Followers WHERE user_id = (SELECT user_id FROM USER WHERE user_name = 'johnlen');
307
308 -- 14 DELETING A LIKE FOR A SPECIFIC FEED--
309 •  DELETE FROM likes WHERE id = 41;
310
```

The 'Output' tab shows the results of the DELETE query:

Action Output	#	Time	Action	Message
	1	21:52:13	DELETE FROM likes WHERE id = 41	1 row(s) affected

15. DELETING A RETWEET FOR A SPECIFIC FEED: By using this query, we can deselect a retweet for a specific feed with the help of the retweet id.

```
DELETE FROM Retweet WHERE id = 51;
```

The screenshot shows the MySQL Workbench interface with a connection named 'Twitter'. The SQL editor contains the following code:

```
310
311 -- 15 DELETING A RETWEET FOR A SPECIFIC TWEET(FEED)
312 •  DELETE FROM Retweet WHERE id = 51;
313
314
```

The 'Output' tab shows the results of the DELETE query:

Action Output	#	Time	Action	Message
	1	21:53:43	DELETE FROM Retweet WHERE id = 51	1 row(s) affected

Neo4j Database UseCases:

1. First Use case is about retrieving some followers details of a specific user.

Match(e:Followers) where e.user_id='u1' return e.user_id,e.follower_id,e.created_at

The screenshot shows the Neo4j Bolt Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database is 'twitter'. The main area contains a query window with the following content:

```
twitter$ Match(e:Followers) where e.user_id='u1' return e.user_id,e.follower_id,e.created_at
```

Below the query, a table displays the results:

	e.user_id	e.follower_id	e.created_at
1	"u1"	"u2"	"1/1/2022 12:00:00 AM"
2	"u1"	"u3"	"1/1/2022 12:56:00 AM"
3	"u1"	"u20"	"8/12/2017 11:20:24 AM"
4	"u1"	"u23"	"8/12/2017 11:20:25 AM"
5	"u1"	"u24"	"8/12/2017 11:21:24 AM"
6	"u1"	"u25"	"8/12/2017 11:21:24 AM"
7			

At the bottom of the table, a message states: "Started streaming 7 records in less than 1 ms and completed in less than 1 ms."

2. To get all the first names and last names of the users who live in India

Match(e:User) where e.country='India' return e.first_name,e.last_name,e.country

The screenshot shows the Neo4j Bolt Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database is 'twitter'. The main area contains a query window with the following content:

```
twitter$ Match(e:User) where e.country='India' return e.first_name,e.last_name,e.country
```

Below the query, a table displays the results:

	e.first_name	e.last_name	e.country
1	"Sindhuja"	"Yerramalla"	"India"
2	"Venkatesh"	"Panguluri"	"India"
3	"Amir"	"Khan"	"India"
4	"latha"	"mangeskar"	"India"
5	"krithi"	"shetty"	"India"
6	"ram gopal"	"varma"	"India"
7			

At the bottom of the table, a message states: "Started streaming 24 records in less than 1 ms and completed in less than 1 ms."

3. To know about the number of likes a specific tweet got

Match(e:Likes) where e(tweet_id='t1' return count(distinct e.liked_by)

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database is set to 'twitter'. In the main query panel, the following Cypher query is run:

```
twitter$ Match(e:Likes) where e(tweet_id='t1' return count(distinct e.liked_by)
```

The results are displayed in a table:

	count(distinct e.liked_by)
1	2

Below the table, a status message reads: "Started streaming 1 records after 8 ms and completed after 8 ms."

4. To know about the number of comments a specific tweet got

Match(e:Comments) where e(tweet_id='t1' return count(e.commented_by)

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is visible, showing the database is set to 'twitter'. In the main query panel, the following Cypher query is run:

```
twitter$ Match(e:Comments) where e(tweet_id='t1' return count(distinct e.commented_by)
```

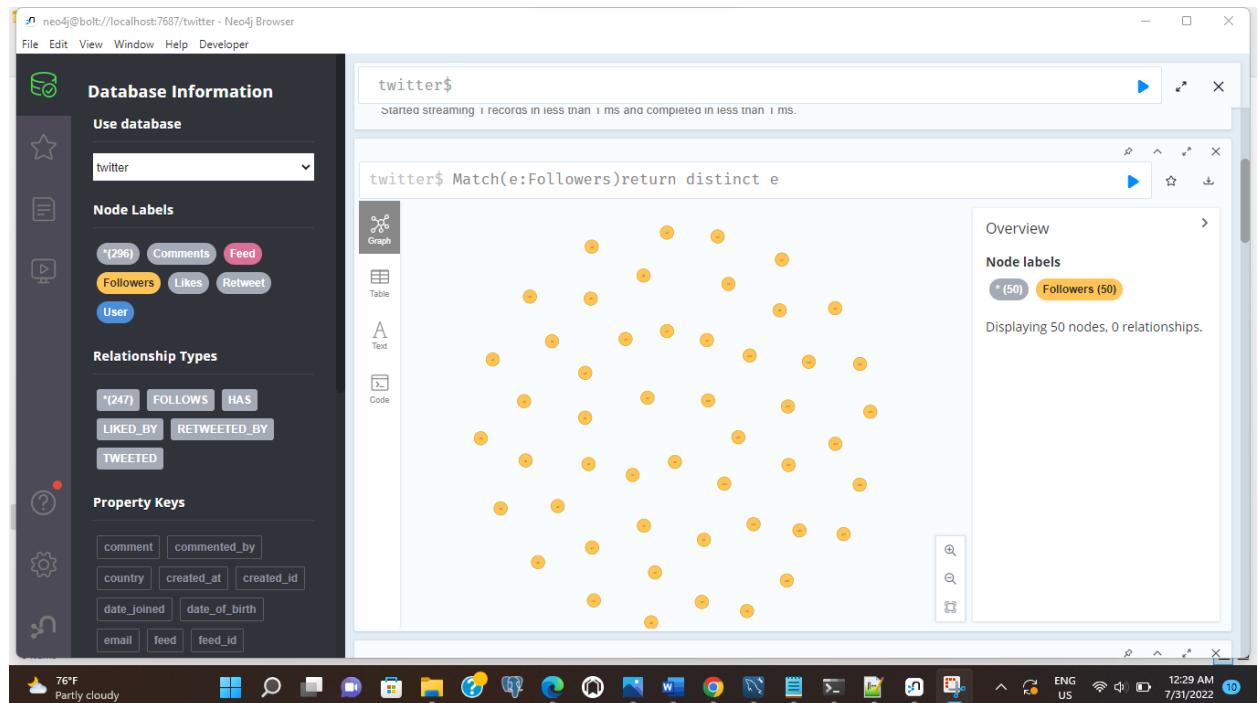
The results are displayed in a table:

	count(distinct e.commented_by)
1	3

Below the table, a status message reads: "Started streaming 1 records in less than 1 ms and completed in less than 1 ms."

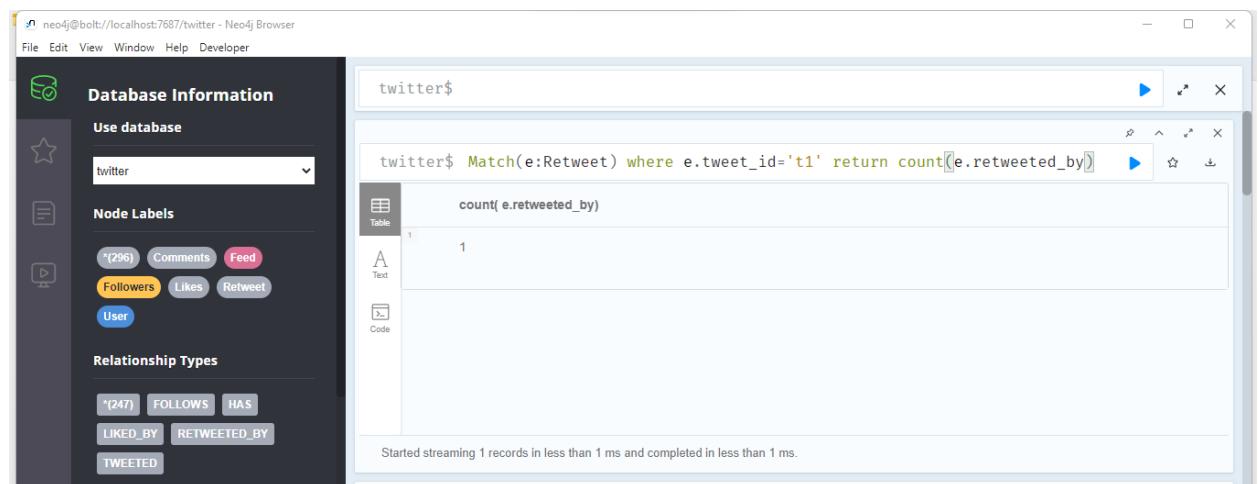
5. To get all the records of a particular label (Ex: Followers)

Match(e:Followers) return distinct e



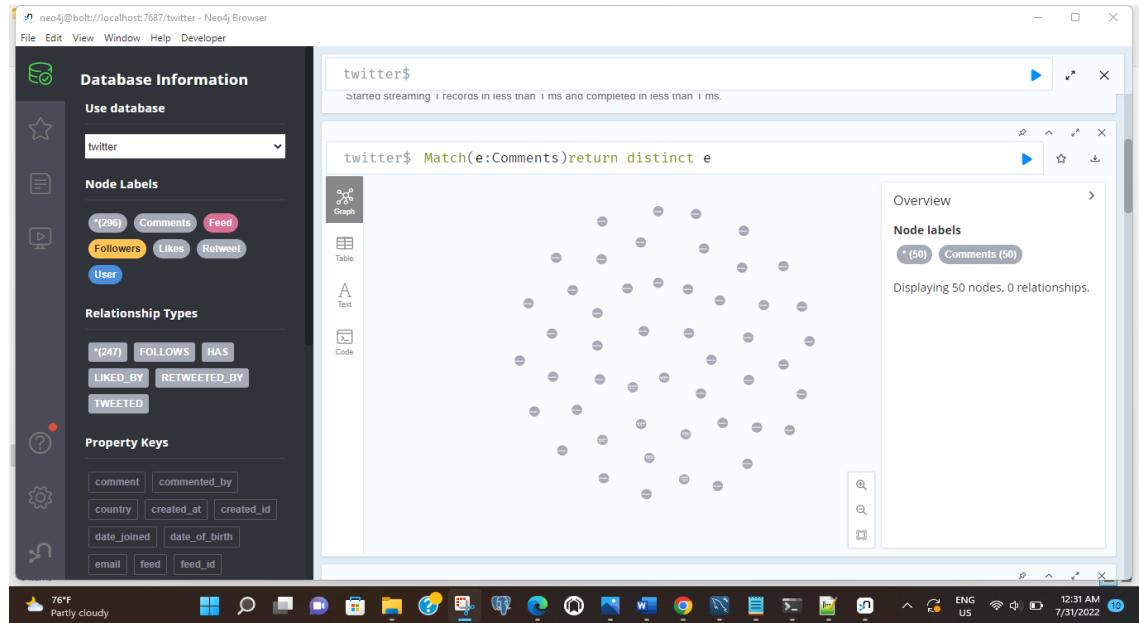
6. To know about the number of retweets a specific tweet got

Match(e:Retweet) where e.tweet_id='t1' return count(e.retweeted_by)



7. To get all the 50 records of Comments table

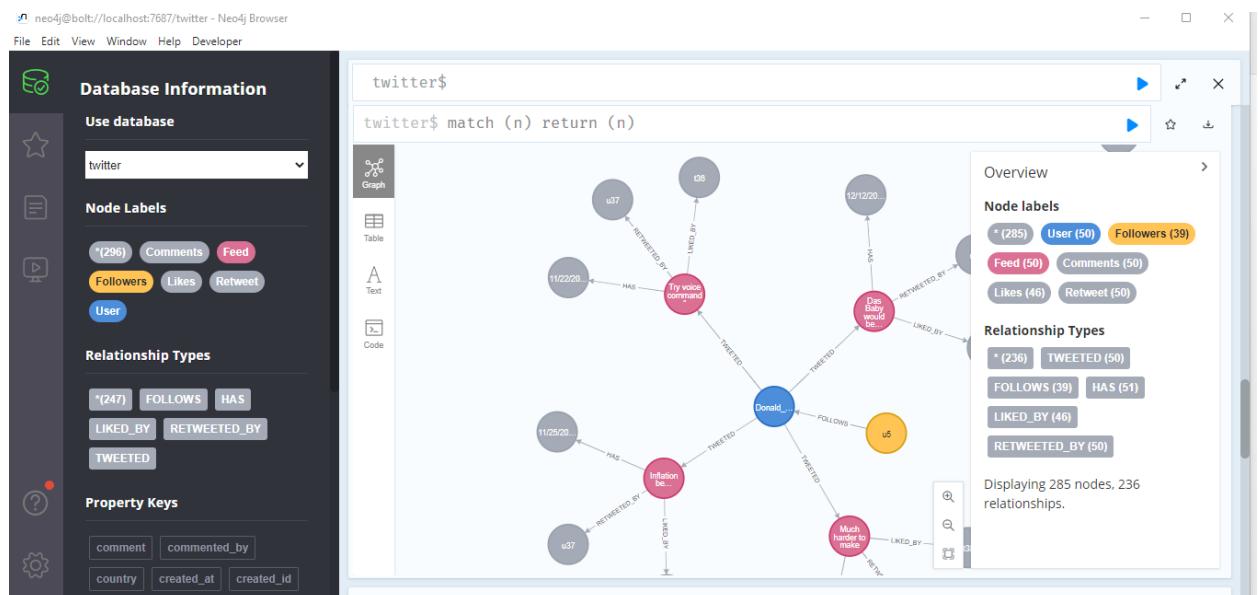
Match(e:Comments) return distinct e



8. To get all node labels and their relationships and complete database graph

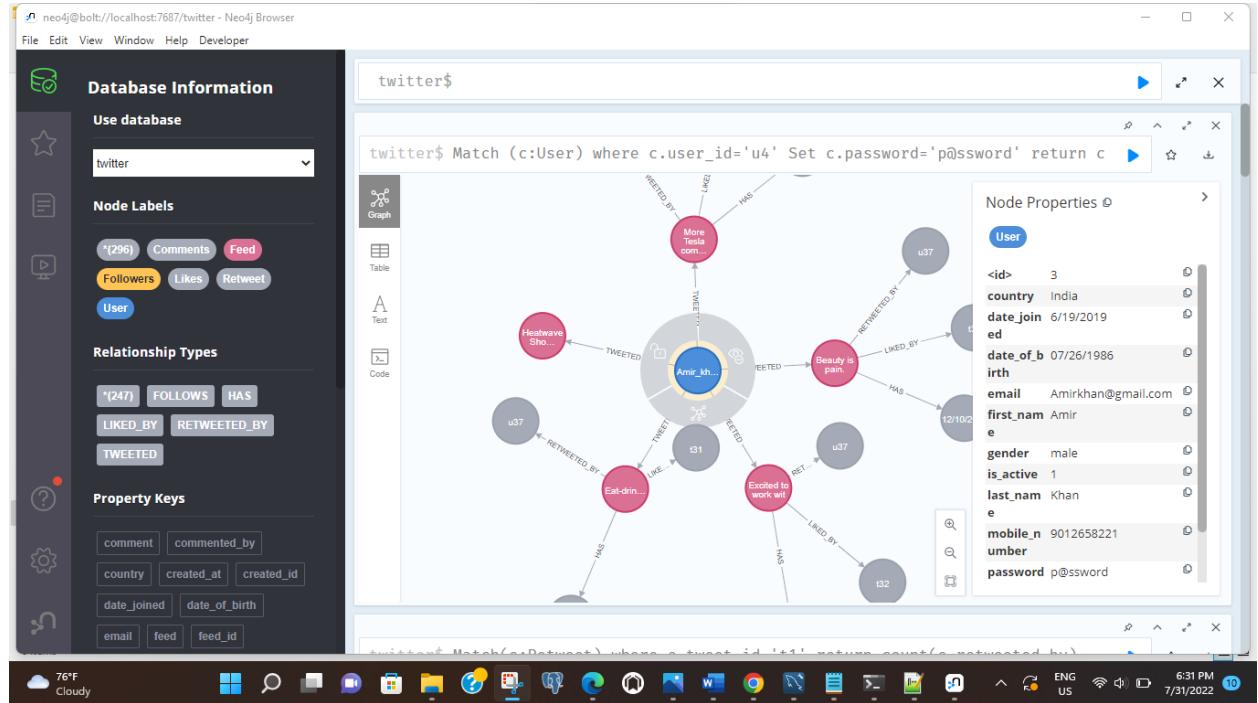
(We used this particular query to show all the relationships we made in this project in graph representation. Please zoom the picture for better view)

Match (n) return (n)



9. We can use this query when a specific user wants to update his/her password.

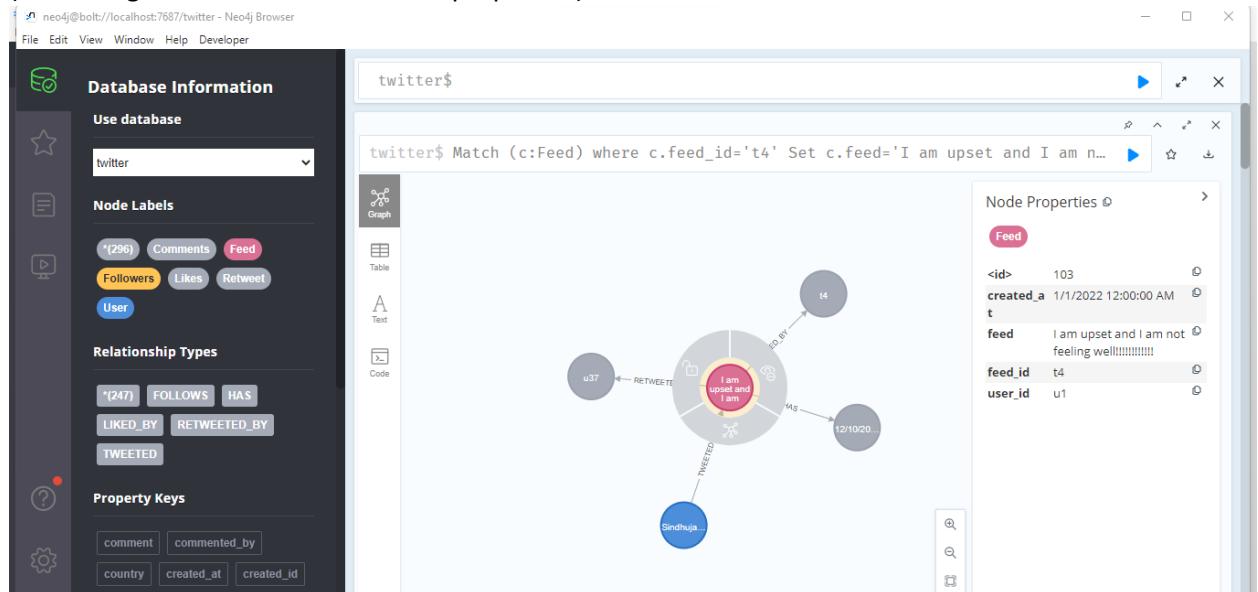
Match (c:User) where c.user_id='u4' Set c.password='P@ssword' return c
 (The change can be seen in the node properties.)



10. We can use this query when a particular user wants to modify his/her tweet.

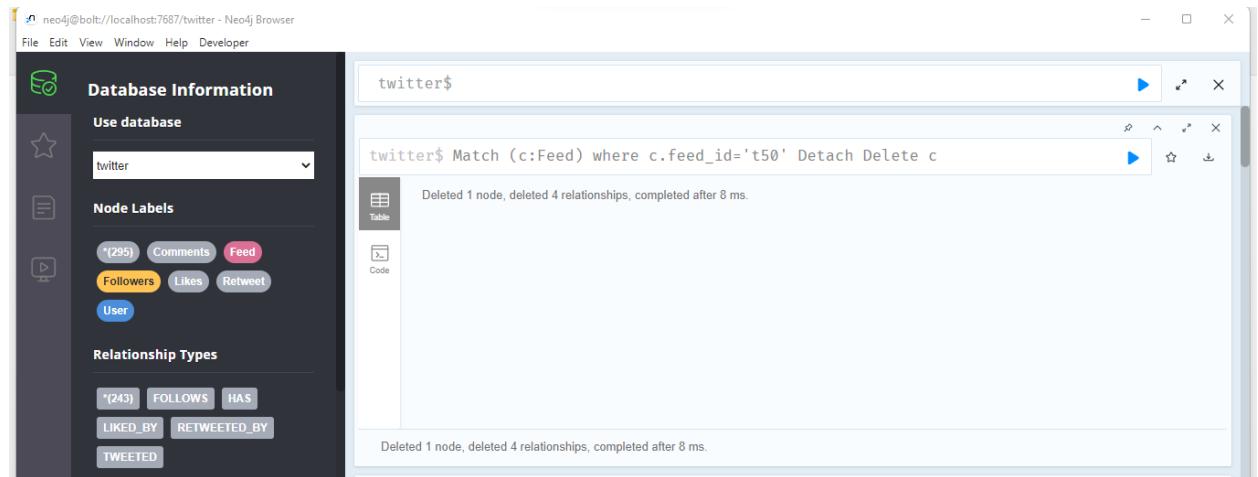
Match (c:Feed) where c.feed_id='t4' Set c.feed='I am upset and I am not feeling well!!!!!!'

(The change can be seen in the node properties)



11. We can use this query when a particular User wants to delete a particular tweet

Match (c:Feed) where c.feed_id='t50' Detach Delete c



12. When user wants to remove a specific follower.

Match (c:Follower) where c.follower_id = 'u5' Detach Delete c



13. To get the first name and last name of the users of all the tweets

Match(e:User)-[r:TWEETED]->(t:Feed) return e.first_name,e.last_name,t.feed_id

The screenshot shows the Neo4j Browser interface. On the left, the 'Database Information' sidebar is open, showing the database 'twitter' is selected. The 'Node Labels' section lists 'Comments', 'Feed', 'Followers', 'Likes', 'Retweet', and 'User'. The 'Relationship Types' section lists 'FOLLOWS', 'HAS', 'LIKED_BY', 'RETWEETED_BY', and 'TWEETED'. The 'Property Keys' section lists various keys like 'comment', 'commented_by', 'country', etc. The main window displays a table with the results of the Cypher query:

	e.first_name	e.last_name	t.feed_id
1	"Sindhuja"	"Yerramalla"	"t1"
2	"Venkatesh"	"Panguluri"	"t2"
3	"krithi"	"shetty"	"t3"
4	"Sindhuja"	"Yerramalla"	"t4"
5	"Venkatesh"	"Panguluri"	"t5"
6	"Iatha"	"mangeskar"	"t6"

Below the table, a message says 'Started streaming 49 records after 1 ms and completed after 2 ms.'

14. To get all the Feed of a specific User.

Match (c:Feed) where c.user_id='u1' return c

The screenshot shows the Neo4j Browser interface. The 'Database Information' sidebar is open, showing the database 'twitter' is selected. The 'Node Labels' section lists 'Comments', 'Feed', 'Followers', 'Likes', 'Retweet', and 'User'. The 'Relationship Types' section lists 'FOLLOWS', 'HAS', 'LIKED_BY', 'RETWEETED_BY', and 'TWEETED'. The main window displays a graph visualization with a central node 'Sindhuja' (blue) connected to four other nodes (red circles) via 'TWEETED' relationships. The nodes are labeled: 'wishing everyone', 'I am impressed I am', 'Working in pro...', and 'GOT ENGA...'. The right panel shows an 'Overview' section with 'Node labels' (User (1), Feed (4)), 'Relationship Types' (TWEETED (4)), and a note 'Displaying 5 nodes, 4 relationships.'

15. If a new follower is added to a specific user

```
CREATE (f51:Followers{  
follower_id:'u20',user_id:'u22',created_at:'9/11/2018 10:56:54 AM' })  
CREATE (f51)-[:FOLLOWS]->(u22)
```

