

רשת חברתית

דוח פרויקט

מאת:

דניאל יוחנן 322406232

&

אבי פדר 208199638

המרצה: מר יאיר גלודשטיין



3.....	רשת חברתית – תיאור המערכת
4.....	חלק א – ישויות ותרשימים
4.....	ישויות
6.....	תרשים ERD
7.....	תרשים Relational schema
8.....	חלק ב – יצירת הטבלאות והכנסת נתונים
8.....	פקודות יצירת הטבלאות
9.....	הכנסת הנתונים לטבלאות
12.....	חלק ג – שאילתות ואינדקסים
12.....	שאילתות
16.....	אינדקסים
17.....	חלק ד – אינטגרציה ושאילתות נוספות
17.....	אינטגרציה
18.....	שאילתות נוספות
20.....	חלק ה – גרפים views
20.....	גרפים
22.....	Views
26.....	חלק ו – פונקציות ופרוצדורות
26.....	פונקציות
27.....	פרוצדורות
28.....	נספח א – קוד הכנסת הנתונים
29.....	נספח ב – קוד של השאילתות
30.....	נספח ג – קוד של האינדקסים
31.....	נספח ד – קוד של השאילתות הנוספות
32.....	נספח ה – קוד של הגרף הראשון
32.....	נספח ו – קוד של הגרף השני
32.....	נספח ז – קוד עבור שני הVIEW הראשונים
33.....	נספח ח – קוד עבור שני הVIEW השניים
33.....	נספח ט – קוד עבור הפונקציות
34.....	נספח י – קוד עבור הפרוצדורות

רשת חברתית – תיאור המערכת

רשת חברתית (Social network) היא מבנה חברתי המורכב מקבוצה של מספר גורמים חברתיים (כגון אנשים או ארגונים), המקיימים ביניהם מערכת של קשרים הדדיים בעוצמות שונות.

הרשת כוללת חברים, קבוצות, ארגונים ועוד. הרשת כוללת קשרים בין גורמים כאלו ומתבססת על פעילותם.

רשת חברתית מקוונת היא פלטפורמה או אתר אינטרנט המתמקד בבנייה או שיקוף של קשרים חברתיים בין אנשים. באמצעותן יכול כל אדם לגלות מי החברים של חבריו, לקרוא את הפרופיל שלהם וליצור איתם קשר.

בפרויקט שלנו אנחנו נבנה בסיס נתונים של רשת חברתית.

כל דוג קיבל מספר ישויות שהם חלק מרשת חברתית.

אנחנו אחראים על הישויות:

- 1. profile
- 2. post
- 3. post_type
- 4. post_comment

מהלך הפרויקט נתמקד בעיקר בישויות שתחת אחריותנו.

בין היתר ניתן למצוא עוד ישויות עיקריות בפרויקט כגון:

- 1. SocialGroup
- 2. Friendship
- 3. Like
- 4. Share

את רשימת הישויות המלאה ואת חלוקת העבודה ניתן למצוא [פה](#).

חלק א – ישויות ותרשימים

ישויות

• Profile

ישות זו מייצגת פרופיל משתמש

מספר מזהה של המשתמש	INT	Profile_id
שם פרטי של המשתמש	VARCHAR(20)	First_name
שם משפחה של המשתמש	VARCHAR(20)	Last_name
כתובת האימייל של המשתמש	VARCHAR(30)	Email
סיסמת התחברות של המשתמש	VARCHAR(20)	Pass
מין המשתמש	CHAR(1)	Gender

• Post

ישות זו מייצגת פוסט של משתמש

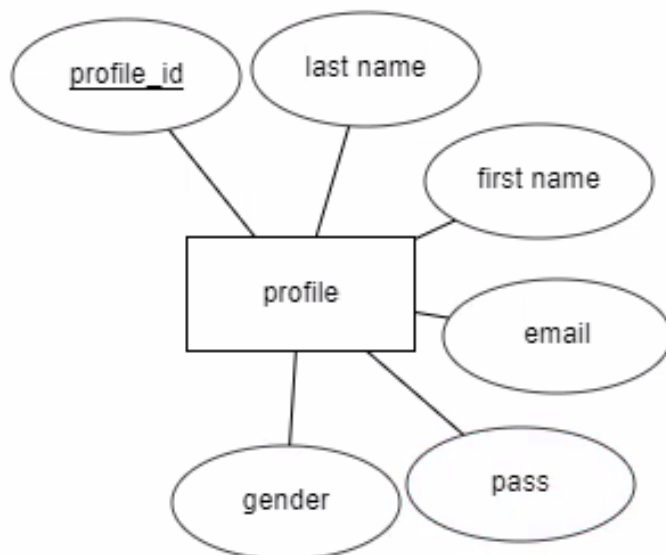
מספר מזהה של הפוסט	INT	Post_id
מספר מזהה של המשתמש	INT	Profile_id
תאריך כתיבת הפוסט	DATE	Post_date
נראות הפוסט (פרטי או ציבורי)	NUMERIC(10)	Visability_code
מספר הלייקים לפוסט	INT	Number_of_likes
תוכן הפוסט	VARCHAR(500)	Post_content
סוג הפוסט (פרטי או של קבוצה)	VARCHAR(20)	Post_type
אם זה פוסט של קבוצה – מספר הקבוצה	INT	Group_id

• Comment

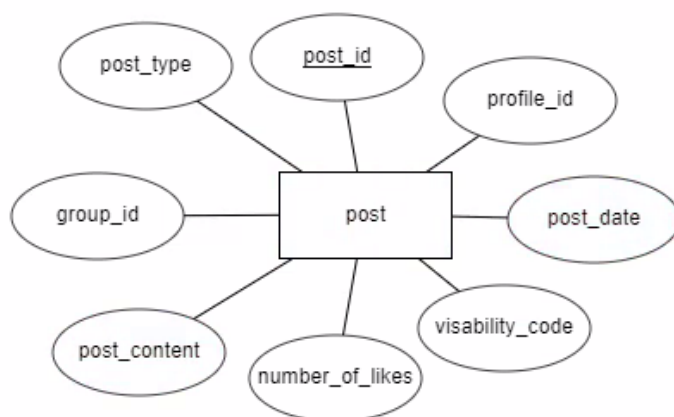
ישות זו מייצגת תגובה על פוסט

מספר מזהה של התגובה	INT	Comment_id
מספר מזהה של הפוסט	INT	Post_id
מספר מזהה של המגיב	INT	Profile_id
תוכן התגובה	VARCHAR(500)	Comment_content

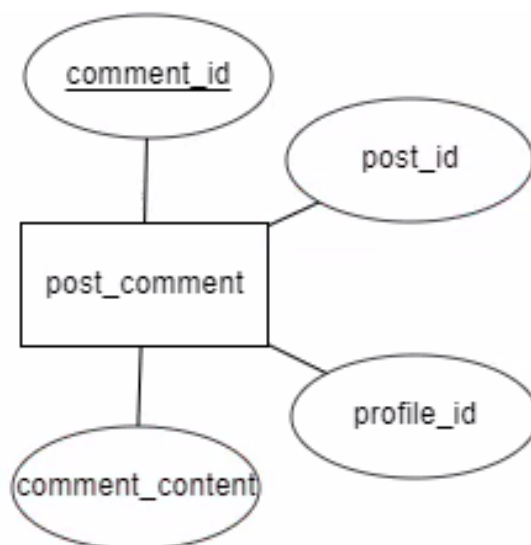
Profile •

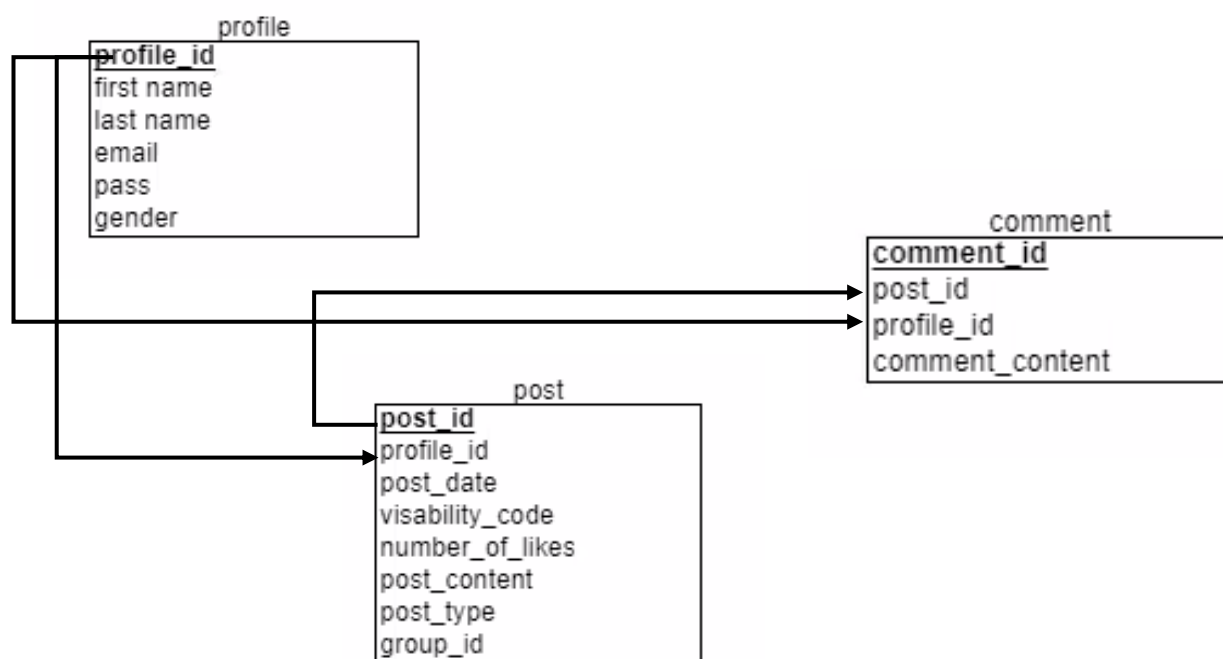


Post •



Comment •





חלק ב – יצירת הטבלאות והכנסת נתונים

פקודות יצירת הטבלאות

מכיוון שבין הטבלאות מתקיימים יחסי תלות שונים, יש חשיבות לסדר יצירת הטבלאות.

הקוד ליצירת הטבלאות הוא:

```
CREATE TABLE profile
(
  profile_id INT NOT NULL,
  first_name VARCHAR(20) NOT NULL,
  last_name VARCHAR(20) NOT NULL,
  email VARCHAR(30) NOT NULL,
  pass VARCHAR(20) NOT NULL,
  gender CHAR(1) CHECK(gender='F' or gender='M') NOT NULL,
  PRIMARY KEY (profile_id)
);

CREATE TABLE post
(
  post_id INT NOT NULL,
  profile_id INT NOT NULL,
  post_date DATE NOT NULL,
  visability_code NUMERIC(10) NOT NULL,
  number_of_likes INT NOT NULL,
  post_content VARCHAR(500) NOT NULL,
  post_type VARCHAR(20) NOT NULL,
  group_id INT NOT NULL,
  PRIMARY KEY (post_id),
  foreign key(profile_id) references profile(profile_id)
);

CREATE TABLE post_comment
(
  comment_id INT NOT NULL,
  post_id INT NOT NULL,
  profile_id INT NOT NULL,
  comment_content VARCHAR(500) NOT NULL,
  PRIMARY KEY (comment_id),
  foreign key(profile_id) references profile(profile_id),
  foreign key(post_id) references post(post_id)
);
```


הכנסת הנתונים לטבלאות
(את הקוד של הנתונים ניתן לראות בנספח א או [בגיט](#))

• Profile

ראשית, נכניס נתונים לטבלה הראשונה שלנו – profile.

לצורך כן נשתמש באתר mockaroo שמאפשר לנו ליצור נתונים ולהוריד אותם כקובץ CSC.

Field Name	Type	Options
profile_id	Sequence	start at: 9001 step: 1 repeat: 1 restart at: blank: 0 %
first_name	First Name	blank: 0 %
last_name	Last Name	blank: 0 %
email	Email Address	blank: 0 %
gender	Gender (abbrev)	blank: 0 %
pass	Password	blank: 0 %

ADD ANOTHER FIELD

Rows: 10000 Format: SQL Table Name: profile ☐ include CREATE TABLE

Append Dataset: choose a dataset...

לאחר הורדת הנתונים, נטען את קובץ הCSC לתוכנה ונמלא את הטבלה:

	PROFILE_ID	FIRST_NAME	LAST_NAME	EMAIL	PASS	GENDER
1	6997	Terrence	Gilley	terrence.gilley@yumbrands.com	887880404	F
2	6998	Richard	Alston	richard.a@httprint.ca	229526463	O
3	6999	Leo	Sweeney	leo.sweeney@providenceservice.com	707536111	M
4	7000	Taye	Klugh	tklugh@directdata.uk	728477299	F
5	7001	Kenneth	Evanswood	kenneth.evanswood@appriss.com	828246673	F
6	7002	Lorraine	Callow	lorraine.callow@pinnaclestaffing.lt	985576344	M
7	7003	Bridgette	Fisher	bridgette.f@healthscribe.com	601869273	F
8	6881	Gin	Calle	gin.calle@ptg.com	206373139	O
9	6882	Chalee	Belle	chalee.belle@northhighland.br	735106242	M
10	6883	Hazel	Kotto	hazel.kotto@visionarysystems.ch	529107382	F
11	6884	First	Glover	first.g@refinery.com	468508279	O
12	6885	Jodie	Rhymes	jodie.r@lms.com	786589939	M
13	6886	Marie	Griggs	mgriggs@electricalsolutions.com	805261409	F
14	6887	Maggie	Seigny	mseigny@ibfh.br	587230922	O
15	6888	Regina	Thorton	regina@ccb.com	491590791	O
16	6889	Wesley	Ledger	ww.ledger@kingland.be	255143046	F
17	6890	Armand	Dukakis	armand@providentbancorp.com	189400825	M
18	6891	Kyle	Root	k.root@johnkeeler.com	824168823	M
19	6892	Chely	Dourif	chelyd@qestrel.com	217946050	M
20	6893	Mary	Spector	mary.spector@venoco.be	471738661	M
21	6894	Angela	Moody	angela@tama.com	937700658	M
22	6895	Harriet	Kelly	harriet.kelly@kitba.nl	889654542	F
23	6896	Rosie	Atlas	rosie.atlas@lemproducts.it	317430747	O
24	6897	Jimmy	Starr	jimmy.starr@integramedamerica.nl	432663749	F

Post •

לצורך מילוי טבלת Post, ניצור טבלה נוספת בשם post_type ונוסיף לה שני סוגי פוסטים, פוסט רגיל ופוסט קבוצה:

	POST_TYPE_CODE	POST_TYPE_NAME	
1	1	NORMAL	...
2	0	GROUP	...

לאחר מכן נכתוב סקריפט בפייטון כדי ליצור פקודות insert data ולמלא את הטבלה post:

```
for x in f:
    c += 1
    x = x.replace("date", "post_date")
    x = x.replace("content", "post_content")
    counter += 1
    count += 1
    if count % 1001 == 0:
        count = 1
    y = x
    profile_id = random.randint(1, 10000)
    arr[profile_id] += 1
    post_id = arr[profile_id]
    visibility = random.randint(0, 1)
    likes = random.randint(0, 2000)
    type = random.randint(0, 1)
    group = 'NULL'
    if type == 0:
        group = random.randint(1, 500)
    lst = y.split("values")
    prop = lst[1].split(',')
    post_id_new = prop[0].replace(str(counter), str(post_id))
    profile_id_new = prop[1].replace(str(count), str(profile_id))
    date_new = change_format_date(prop[2].split("'")[1])
    visibility_new = prop[3].replace(str(count), str(visibility))
    likes_new = prop[4].replace(str(count), str(likes))
    type_new = prop[6].replace(str(count), str(type))
    group_new = prop[7].replace(str(count), str(group))
    y = "values".join([lst[0], ',', '.join([post_id_new, profile_id_new, date_new, visibility_new, likes_new, prop[5], type_new, gr
    p.write(y)
```

נריץ את פקודות insert data בתוכנה ונמלא את הטבלה בנתונים:

	POST_ID	PROFILE_ID	POST_DATE	VISIBILITY_CODE	NUMBER_OF_LIKES	POST_CONTENT	POST_TYPE_CODE	GROUP_ID
1	1	2462	10/05/2020	0	563	vehicula condimentum	1	...
2	1	176	15/04/2020	1	1587	metus vitae ipsum aliquam	0	480
3	1	8102	01/12/2020	1	47	eu interdum eu tincidunt in	1	...
4	1	1649	26/01/2021	1	1026	in hac	1	...
5	1	6786	17/10/2020	1	911	habitasse platea	1	...
6	1	5831	08/03/2021	1	1068	etiam pretium iaculis justo in	0	129
7	1	9330	26/01/2021	1	543	sed lacus	0	298
8	1	5146	27/06/2020	0	1242	arcu libero rutrum ac lobortis	0	208
9	1	2729	01/08/2020	1	1870	mauris vulputate elementum nullam varius	0	480
10	1	2001	06/03/2021	0	826	ipsum aliquam non mauris morbi	0	414
11	1	3169	04/05/2020	1	1652	tempus sit amet	1	...
12	1	9326	29/04/2020	0	1237	mus etiam	1	...
13	1	734	21/05/2020	0	865	nulla nisl	0	427
14	1	9445	21/05/2020	1	1544	ultrices eu nibh quisque	1	...
15	1	7248	30/10/2020	0	1454	lacinia sapien quis libero nullam	1	...
16	1	9143	31/03/2021	0	1599	diam vitae	1	...
17	1	1818	29/03/2021	1	1792	proin leo odio porttitor id	1	...
18	1	9347	19/02/2021	1	1533	nisi at nibh in	0	401
19	1	6615	23/09/2020	0	744	magna bibendum imperdiet nullam orci	1	...
20	2	8253	26/12/2020	1	813	porta volutpat erat quisque	0	361
21	1	885	11/05/2020	0	1184	donec quis orci	1	...
22	1	9232	29/07/2020	1	913	sapien cum	0	455
23	1	3812	23/12/2020	1	1087	faucibus orci luctus et ultrices	0	108
24	1	9480	08/07/2020	0	1800	dolor sit amet consectetur adipiscing	1	...

Comment •

בטבלה הזאת נשתמש בData generator של SQL כדי להוסיף נתונים.

לאחר הרצת Data generator נקבל טבלה מלאה בנתונים:

	COMMENT_ID	POST_ID	PROFILE_ID	COMMENT_CONTENT
1	1	1	1125	Morbi porttitor lorem id ligula. Suspendisse ornare consequat lectus. In e...
2	1	1	8901	Mauris sit amet eros. Suspendisse accumsan tortor quis turpis. Sed ante. ...
3	1	1	7938	Vivamus metus arcu, adipiscing molestie, hendrerit at, vulputate vitae, ni...
4	1	2	2836	Aenean auctor gravida sem. Praesent id massa id nisl venenatis lacinia. A...
5	1	1	5473	Nulla ac enim. In tempor, turpis nec euismod scelerisque, quam turpis ac...
6	1	1	9705	Maecenas tristique, est et tempus semper, est quam pharetra magna, ac...
7	1	1	9207	In hac habitasse platea dictumst. Etiam faucibus cursus urna. Ut tellus. N...
8	1	1	5184	Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere i...
9	1	1	8969	Vivamus metus arcu, adipiscing molestie, hendrerit at, vulputate vitae, ni...
10	1	1	1167	Vivamus metus arcu, adipiscing molestie, hendrerit at, vulputate vitae, ni...
11	1	1	6488	Proin interdum mauris non ligula pellentesque ultrices. Phasellus id sapi...
12	1	1	5313	Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta...
13	1	1	9146	Cum sociis natoque penatibus et magnis dis parturient montes, nascetur...
14	1	1	232	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim...
15	1	1	8985	Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta...
16	1	1	9001	Suspendisse potenti. Nullam porttitor lacus at turpis. Donec posuere met...
17	1	1	1402	Phasellus id sapien in sapien iaculis congue. ...
18	1	1	4649	Aenean lectus. Pellentesque eget nunc. Donec quis orci eget orci vehicul...
19	1	1	4245	Phasellus sit amet erat. Nulla tempus. Vivamus in felis eu sapien cursus v...
20	1	1	8210	Cum sociis natoque penatibus et magnis dis parturient montes, nascetur...
21	1	1	3378	Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilis...
22	1	1	5689	Proin eu mi. Nulla ac enim. In tempor, turpis nec euismod scelerisque, q...
23	1	1	9944	In blandit ultrices enim. Lorem ipsum dolor sit amet, consectetur adipis...
24	1	1	8679	In sagittis dui vel nisl. Duis ac nibh. Fusce lacus purus, aliquet at, feugiat...

חלק ג – שאילות ואינדקסים

שאילות

נציג 8 שאילות מתקדמות על הקוד שלנו.

בכל שאילתה נסביר מה השאילתה מחזירה, מה מטרת השיאלתה וכמה ערכים קיבלנו בתוצאה.

(את הקוד של השאילות ניתן לראות בנספח ב או [בגיט](#))

שאילתה 1:

- מה השאילתה נותנת:

השאילתה מחזירה טבלה שמפרטת לפי כל מין את מספר הלייקים המקסימלי לפוסט, מספר הלייקים המינימלי לפוסט, מספר הלייקים הממוצע לפוסט, וכמה פוסטים יש מפרופילים מאותו מין.

כדי שנוכל ללמוד על השוק לפי פלחים שונים.

- כמה שורות יש בתוצאה:

3 שורות.

	GENDER	AVG(NUMBER_OF_LIKES)	MAX(NUMBER_OF_LIKES)	MIN(NUMBER_OF_LIKES)	COUNT(*)
1	M	1007.81987000929	2000	0	3231
2	O	991.822833481219	2000	2	3381
3	F	1012.82821723731	2000	0	3388

שאילתה 2:

- מה השאילתה נותנת:

משתמשים שיש להם פוסט שבו מספר הלייקים גבוה מהממוצע.

מכיוון שנרצה למצוא ולאתר טרנדים חדשים.

- כמה שורות יש בתוצאה:

5001 שורות.

	PROFILE_ID	FIRST_NAME	LAST_NAME	POST_ID	NUMBER_OF_LIKES
1	7000	Taye	Klugh	1	1416
2	7001	Kenneth	Evanswood	3	1762
3	7001	Kenneth	Evanswood	1	1429
4	7002	Lorraine	Callow	1	1552
5	6881	Gin	Calle	1	1660
6	6883	Hazel	Kotto	1	1765
7	6884	First	Glover	1	1947
8	6892	Chely	Dourif	1	1983

שאלתה 3:

- מה השאלתה נותנת:
פרופילים שיש להם פוסטים עם מספר ליקיים גבוה מהממוצע וגם מספר תגובות שהגיבו להם גבוה מהממוצע.
מכיוון שנרצה למצוא פרופילים פופולריים מאוד.
כמה שורות יש בתוצאה:
3956 שורות.

		PROFILE_ID	FIRST_NAME	LAST_NAME	GENDER
▶	1	3	Halle	Rydell	F
	2	10	Mitchell	Zellweger	F
	3	11	Adrien	Love	F
	4	12	Franz	Tsettos	O
	5	14	Bonnie	Russell	F

שאלתה 4:

- מה השאלתה נותנת:
פרופילים שהגיבו מספר תגובות גבוה מהממוצע.
מכיוון שנרצה למצוא פרופילים פעילים מאוד.
כמה שורות יש בתוצאה:
416 שורות.

		PROFILE_ID	FIRST_NAME	LAST_NAME	GENDER
▶	1	5489	Ramsey	Rooker	O
	2	5785	Denise	Tennison	M
	3	5865	Carole	Atkins	O
	4	6112	Emmylou	McCann	O
	5	6193	Taye	McIntyre	M
	6	8090	Ann	Sevigny	M
	7	8752	Rod	McGriff	O
	8	9144	Tori	Sladon	M

שאלתה 5:

- מה השאלתה נותנת:
מתוך האנשים שלא הגיבו על אף פוסט, אנחנו שולפים את בעלי מספר הלייקים הרב ביותר ואת מגדרם.
זאת על מנת לנתח את אופי נתינת הלייקים לפרופילים שלא מגיבים ולא משתתפים פעילים בפלטפורמה.
- כמה שורות יש בתוצאה:
5 שורות.

	PROFILE_ID	POST_ID	NUMBER_OF_LIKES
1	7406	1	2000
2	8202	1	2000
3	8497	3	2000
4	3325	3	2000
5	9832	2	2000

שאלתה 6:

- מה השאלתה נותנת:
פרטים על פרופילים מובילים בעלי ממוצע לייקים הגדול מ 1500.
מכיוון ש 1500 הינו כ-15% מכלל הפרופילים בפלטפורמה וככה נמצא פרופילים בעלי עניין.
- כמה שורות יש בתוצאה:
263 שורות.

	PROFILE_ID	FIRST_NAME	LAST_NAME	AVG(NUMBER_OF_LIKES)	NUM_OF_POST
1	6978	Robert	Tah	1950	2
2	5565	Seann	Walken	1526.33333333333	3
3	5755	Pablo	Merchant	1581.5	2
4	5775	Jet	Saucedo	1607.66666666667	3
5	5913	Antonio	Salt	1730	2
6	6184	Jake	Pollak	1967	2
7	9719	Larenz	Allan	1568.5	2
8	7524	Meryl	Melvin	1804	2
9	4938	Patricia	Hatchet	1857.5	2

שאלתה 7:

- מה השאלתה נותנת:
השאלתה מחזירה פרטים על משתמשים אשר פרסמו פוסטים מכל הסוגים.
זאת על מנת לראות ולנתח את השימוש של משתמשים בסוגי פוסטים שונים.
- כמה שורות יש בתוצאה:
1590 שורות.

		PROFILE_ID	FIRST_NAME	LAST_NAME	GENDER
▶	1	7001	Kenneth	Evanswood	F
	2	6883	Hazel	Kotto	F
	3	6897	Jimmy	Starr	F
	4	6900	Vonda	Idol	O
	5	6902	Nick	Astin	F
	6	6906	Madeline	Whitman	O
	7	6923	Lorraine	Cale	F
	8	6935	Ceili	Tyson	F
	9	6936	Tammy	Northman	M

שאלתה 8:

- מה השאלתה נותנת:
פרטים על משתמש שמגיבים על פוסטים "רגילים".
על מנת לקבל אינפורמציה אופי המשתמשים.
- כמה שורות יש בתוצאה:
592 שורות.

		PROFILE_ID	FIRST_NAME	LAST_NAME	MAX(P.NUMBER_OF_LIKES)
▶	1	5532	Phoebe	Sutherland	1576
	2	5613	Murray	Irving	142
	3	5755	Pablo	Merchant	1950
	4	5851	Kylie	Sinatra	930
	5	5868	Mindy	Gold	1683
	6	6009	Chalee	Lynn	1599
	7	6077	Herbie	Lerner	1827
	8	6128	Victor	Turturro	974
	9	6193	Taye	McIntyre	682

ע"מ לשפר את זמן הריצה, ניצור אינדקסים לטבלאות.

אינדקס הוא טבלה המאפשרת גישה ישירה לרשומות על-פי מפתח, מבלי שיהיה צורך לסרוק את כל הנתונים, באמצעות התאמה בין המפתח של הרשומה לבין כתובתה בזכרון.

המפתח של הרשומה יכול להיות שדה כלשהו בתוכה המזהה אותה באופן יחיד ומאפשר את אחזור (לדוגמה מספר זהות ברשומה העוסקת באדם, מספר קטלוגי ברשומה העוסקת בפריט במלאי וכדומה).

ע"י יצירת אינדקסים, ניתן לשפר את זמן הריצה של השאילתות שלנו.

ניצור אינדקסים עבור עמודות שימושיות בשאילתות ונבחן את שינוי הזמנים:

(את הקוד של האינדקסים ניתן לראות בנספח ג או [בגיט](#))

```
create index index1 ON AFEDER.PROFILE(profile_id,first_name,last_name,gender);
create index index2 ON AFEDER.POST(post_id,number_of_likes);
create index index3 ON AFEDER.POST_COMMENT(comment_id,post_id,profile_id);
```

נתמקד רק בשאילתות שזמן הריצה שלהם היה מעל לשניה.

שאילתה מספר	זמן ריצה לפני יצירת האינדקס	זמן ריצה לאחר יצירת האינדקס	אחוז השינוי בזמן הריצה
2	2.4	6.6	הרעה של 4.2 שניות כלומר תוספת של 175 אחוז בזמן
3	4.2	4.4	הרעה של 0.2 שניות כלומר תוספת של 4 אחוז בזמן
4	4.4	2.8	הטבה של 1.6 שניות כלומר חיסכון של 65 אחוז בזמן
6	4.5	2.7	הטבה של 1.8 שניות כלומר חיסכון של 40 אחוז בזמן
7	5.7	2.5	הטבה של 2.7 שניות כלומר חיסכון של 57 אחוז בזמן
8	5.7	1.7	הטבה של 4 שניות כלומר חיסכון של 70 אחוז בזמן

מכיוון שאינדקס הוא אובייקט בפני עצמו, שגם לו זמן ריצה, כפי שניתן לראות לפעמים נקבל הרעה בזמן הריצה.

חלק ד - אינטגרציה ושאלות נוספות

אינטגרציה

ע"מ להמשיך בפרויקט, עלינו לבצע אינטגרציה לטבלאות של זוגות נוספים ולתת לזוגות אחרים גישה לטבלאות שלנו.

נוסיף למספר זוגות שפנו אלינו גישה לטבלאות.

הטבלה PROFILE היא הטבלה המבוקשת ביותר מצד זוגות אחרים, מכיוון שהיא מכילה את המשתמשים של הרשת החברתית ועליה מתבסס הפרויקט:

AFEDER.PROFILE@LABDBWIN										
<div>General Columns Keys Checks Indexes Privileges Triggers</div>										
Grantee	Select	Insert	Update	Delete	References	Alter	Index	Read	Debug	
ASEBBAG	Yes				Yes			Yes		
MZUCKERB	Yes				Yes			Yes		
NSHUSHAN	Yes				Yes			Yes		
YDEMRI	Yes				Yes			Yes		

גם לטבלה POST קיבלנו בקשה לתת הרשאות:

AFEDER.POST@LABDBWIN										
<div>General Columns Keys Checks Indexes Privileges Triggers</div>										
Grantee	Select	Insert	Update	Delete	References	Alter	Index	Read	Debug	
MZUCKERB	Yes				Yes	Yes				

כפי שניתן לראות, נתנו גישה רק לקריאת הנתונים מהטבלאות אך לא לבצע שינויים בנתונים. זאת ע"מ לשמור על עקרון separate privilege, כלומר כל אחד יקבל רק את הרשאות שהוא צריך ולא יותר.

אנחנו ביקשנו מזוגות אחרים גישה לטבלאות הבאות:

1. Share
2. Friends
3. Tag

על טבלאות אלו נכתוב שאלות נוספות.

שאלות נוספות

(את הקוד של השאלות הנוספות ניתן לראות בנספח ד או [בגיט](#))

שאלה 1:

מה השאלה נותנת:

השאלה מחזירה עבור כל פרופיל, כמה הצעות חברות הוא מקבל ששלחו לו בקשת חברות. השאלה תשמש את המערכת על מנת לצמצם חשיפה מיותרת ולמקד את ההצעות למשתמש.

	RECIEVERID	NUM_OF_SUGGESTION ▲
1596	2104	1
1525	2016	1
5898	7807	1
134	183	1
2576	3390	1
5030	6636	0
5021	6623	0
5029	6633	0
5031	6637	0
5019	6621	0

שאלה 2:

מה השאלה נותנת:

השאלה מחזירה את הפרופילים, שלהם יש הצעות חברות למשתמשים מסוימים כאשר הפרופיל מופיע לאותם הצעות גם בהצעות חברות (וכמה כאלה יש לו).

(שכל אחד מופיע לשני בהצעת חברות)

השאלה תשמש את המערכת על מנת לבדוק את רמת הדיוק של ההצעות ולמקד את ההצעות למשתמש.

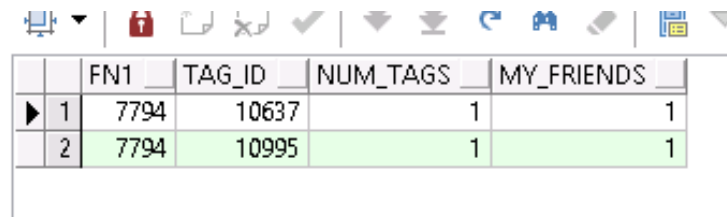
	RECIEVERID	COMMON_SUGGESTION
▶ 1	9417	1
2	7421	1
3	8636	1

שאלתה 3:

מה השאלתה נותנת:

השאלתה מחזירה את הפרופילים, שתויגו בתמונה שבה כל שאר המתויגים נמצאים ברשימת החברים שלהם.

השאלתה תשמש את המערכת לזהות מעגלי חברות ולשפר את דיוק הצעות החברות למשתמשים.



	FN1	TAG_ID	NUM_TAGS	MY_FRIENDS
1	7794	10637	1	1
2	7794	10995	1	1

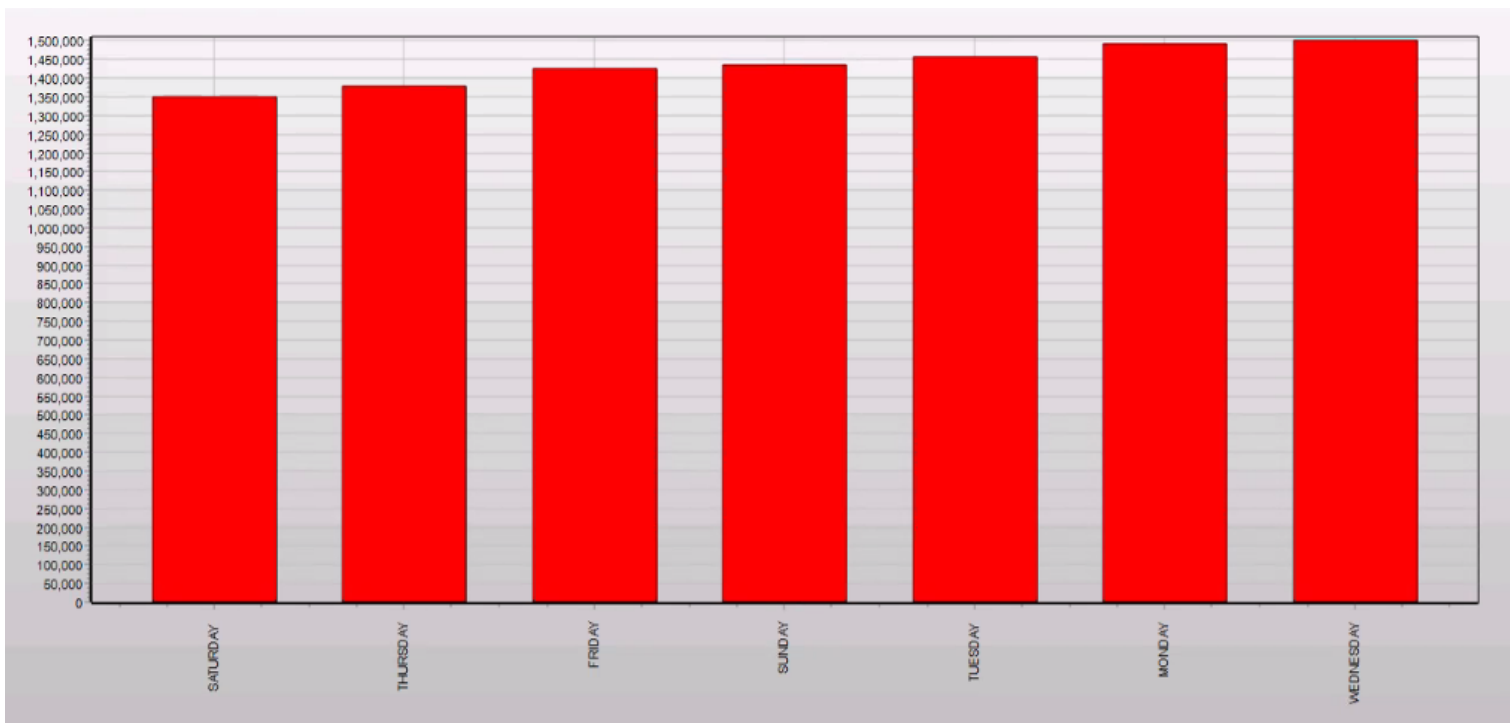
חלק ה - גרפים וviews

גרפים

1. הגרף הראשון שלנו יתמקד בלייקים לפוסטים. ויציג את מספר הלייקים שניתנו בפלטפורמה לפוסטים בחלוקה לפי ימים.

בציר Y נראה את מספר הלייקים ובציר X את יום פרסום הפוסט:

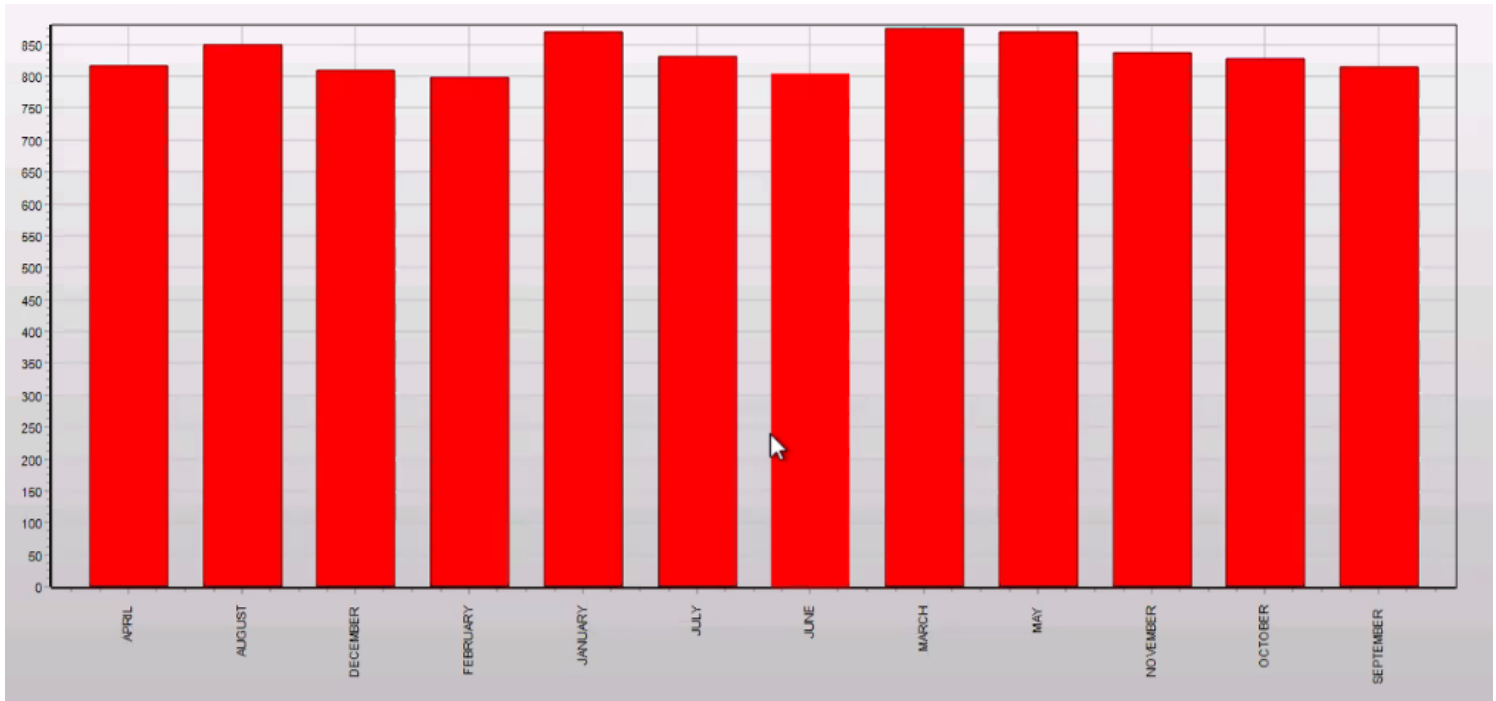
(את הקוד של הגרף ניתן לראות בנספח ה או [בגיט](#))



1. הגרף השני יתמקד בפוסטים שעלו בפלטפורמה ויצוג לנו כמה פוסטים עלו בכל חודש.

בציר Y נראה את כמות הפוסטים שעלו ובציר X את החודש.

(את הקוד של הגרף ניתן לראות בנספח I או [בגיט](#))



View זו טבלה דינאמית על בסיס הנתונים, שמכילה שאילתה שרצה מחדש כל פעם שה View נקרא.
ניצור 4 Views, עבור שני סוגי משתמשים - 2 עבור כל משתמש.

- Views עבור משתמש רגיל

(את הקוד של ה Views ניתן לראות בנספח 2 או [בגיט](#))

1. ניצור View בשם trand_profile שייתן משתמשים שיש להם פוסט שבו מספר הלייקים גבוה מהמוצע.
כל משתמש בפלטפורמה יוכל לראות מה הטרנדים חדשים.

	PROFILE_ID	FIRST_NAME	LAST_NAME	POST_ID	NUMBER_OF_LIKES
1	3	Halle	Rydell	4	1728
2	3	Halle	Rydell	2	1407
3	3	Halle	Rydell	1	1245
4	10	Mitchell	Zellweger	2	1552
5	11	Adrien	Love	1	1717
6	12	Franz	Tsettos	1	1905
7	14	Bonnie	Russell	1	1019
8	16	Mira	Norton	1	1233
9	19	Brad	Thornton	2	1172
10	22	Hal	Malkovich	2	1563
11	22	Hal	Malkovich	1	1300
12	23	Bill	Galecki	1	1056
13	31	Bobbi	Gore	1	1858
14	35	Peabo	Osment	1	1828
15	37	Mekhi	Spears	1	1119
16	38	Colin	Conley	1	1609
17	39	Merle	Russell	2	1602
18	39	Merle	Russell	1	1588
19	44	Cliff	Hall	1	1100
20	46	Clea	Shaw	2	1436

```
create or replace view trand_profile as
(
select profile_id, first_name, last_name, post_id, number_of_likes
from post natural join profile
where number_of_likes >= (
        select avg(number_of_likes)
        from post
    )
);

select * from trand_profile ;
```

2. ניצור View בשם active_profile שייתן משתמשים שהגיבו מספר תגובות גבוה מהממוצע.
כל משתמש בפלטפורמה יוכל לראות מי הם הפרופילים הפעילים והטרנדים.

	PROFILE_ID	FIRST_NAME	LAST_NAME	GENDER
1	383	Phil	Brody	O
2	423	Juliana	Horizon	M
3	546	Daryle	Fisher	O
4	668	Rich	Porter	F
5	743	Vern	Heslov	O
6	908	Wade	Stanley	O
7	1021	Dustin	Bacharach	O
8	1036	Kitty	Stuermer	M
9	1504	Sarah	Marie	O
10	1536	Davey	Collette	O
11	1550	Franz	Folds	F
12	2532	Jody	Day	F
13	3003	Diamond	Ferry	F
14	3041	Isaac	Tillis	F
15	3090	Rutger	Margulies	M
16	3106	Radney	Uggams	O
17	3295	Rodney	Arthur	O
18	3574	Murray	Bragg	M
19	3715	Oro	Chaplin	O
20	3733	Michelle	Bracco	O

```

create view active_profile as
(
select profile_id, first_name, last_name, gender
from profile natural join post_comment
group by profile_id, first_name, last_name, gender
having count(*) > (select avg(c) from (
                                select profile_id, count(*) as c
                                from post_comment
                                group by profile_id
                                ) )
);

select * from active_profile ;

```

- Views עבור מנהל

(את הקוד של הViews ניתן לראות בנספח ח או [בגיט](#))

1. ניצור View בשם active_friendship שייתן כמה חברים יש לכל משתמש במערכת. באמצעות View זה והView הבא מנהל מערכת יוכל למצוא חשבונות לא פעילים.

	USER_ID	FRIENDS
▶ 1	1	44
2	22	5
3	25	2
4	30	3
5	34	8
6	42	6
7	43	7
8	51	5
9	54	2
10	57	6
11	78	4
12	83	5
13	87	5
14	91	6

```
create view active_friendship as(
select friend_id1 as user_id, count(*) as friends
from bmarcus.friends
group by friend_id1
);

select * from active_friendship;
```


2. ניצור View בשם friendship_request שייתן כמה בקשות חברות יש לכל משתמש במערכת. באמצעות View זה והקודם מנהל מערכת יוכל למצוא חשבונות לא פעילים.

```
create view friendship_request as(  
select reciever_id as user_id, count(*) as requests  
from ydemri.friendshiprequest  
group by reciever_id  
);  
  
select * from friendship_request;
```

		USER_ID	REQUESTS
▶	1	6131	2
	2	6108	6
	3	2213	2
	4	6034	3
	5	4882	2
	6	5030	1
	7	3832	3
	8	1184	2
	9	8313	3
	10	1876	2
	11	701	2
	12	9470	5
	13	2039	3
	14	8263	3
	15	6731	7
	16	1412	1
	17	6904	2
	18	9510	5

חלק ו – פונקציות ופרוצדורות

פונקציות

(את הקוד של הפונקציות ניתן לראות בנספח ט או [בגיט](#))

1. פונקציה שתקבל שם (VARCHAR) ותחזיר כמה פרופילים יש עם כזה שם פרטי או משפחה (NUMBER).

• מקרה בדיקה 1

נכניס את השם Gilley ונקבל שיש 5 משתמשים עם כזה שם פרטי או משפחה:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Float	5
<input checked="" type="checkbox"/>	user_name	String	Gilley
<input checked="" type="checkbox"/>	*		

• מקרה בדיקה 2

נכניס את השם Avi ונקבל שיש 0 משתמשים עם כזה שם פרטי או משפחה:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Float	0
<input checked="" type="checkbox"/>	user_name	String	Avi
<input checked="" type="checkbox"/>	*		

2. פונקציה שתקבל מספר פרופיל (NUMBER) ותחזיר כמה חברים של אותו פרופיל, תייגו אותו בפוסט כלשהו (NUMBER).

• מקרה בדיקה 1

נכניס מספר פרופיל 6774 ונקבל שיש 0 משתמשים שעונים על התנאי הזה:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Float	0
<input checked="" type="checkbox"/>	profile_id	Float	6774
<input checked="" type="checkbox"/>	*		

• מקרה בדיקה 2

נכניס מספר פרופיל 8155 ונקבל שיש 1 משתמשים שעונים על התנאי הזה:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Float	1
<input checked="" type="checkbox"/>	profile_id	Float	8155
<input checked="" type="checkbox"/>	*		

פרוצדורות

(את הקוד של הפרוצדורות ניתן לראות בנספח י' או [בגיט](#))

1. פרוצדורה שמקבלת מספר פרופיל ומסכמת את המידע על אותו משתמש. הפרוצדורה תדפיס למסך את סיכום הנתונים כמו: סכום הלייקים, כמות הפוסטים, כמות התיוגים, מספר החברים וכו'.

נכניס פרופיל מספר 6775:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	profile_id	Float	6775
<input checked="" type="checkbox"/>	*		

ונקבל:

USER INFO								
ID	POST	LIKES	SHARES	TAGS	FREINDS	SUGGESTION	REQUEST	NOTIFICATION
6775	1	412	3	1	5	2	1	6

2. פרוצדורה שתדפיס לי את 10 אחוז המשתמשים המתויגים ביותר במערכת. הפרוצדורה לא מקבלת ערכים ולא מחזירה ערכים אלא רק סורקת את המסד נתונים ומדפיסה את 10 אחוז המשתמשים המתויגים ביותר במערכת בסדר יורד:

USER INFO	
ID	NUM OF TAGS
9097	10
8634	9
6627	9
3081	8
2413	8
1963	8
1366	8
1837	8
4240	8
4800	8

```

for x in f:
    c += 1
    x = x.replace("date", "post_date")
    x = x.replace("content", "post_content")
    counter += 1
    count += 1
    if count % 1001 == 0:
        count = 1
    y = x
    profile_id = random.randint(1, 10000)
    arr[profile_id] += 1
    post_id = arr[profile_id]
    visibility = random.randint(0, 1)
    likes = random.randint(0, 2000)
    type = random.randint(0, 1)
    group = 'NULL'
    if type == 0:
        group = random.randint(1, 500)
    lst = y.split("values")
    prop = lst[1].split(',')
    post_id_new = prop[0].replace(str(counter), str(post_id))
    profile_id_new = prop[1].replace(str(count), str(profile_id))
    date_new = change_format_date(prop[2].split(" ")[1])
    visibility_new = prop[3].replace(str(count), str(visibility))
    likes_new = prop[4].replace(str(count), str(likes))
    type_new = prop[6].replace(str(count), str(type))
    group_new = prop[7].replace(str(count), str(group))
    y = "values".join([lst[0], ','.join([post_id_new, profile_id_new, date_new, visibility_new, likes_new, prop[5], type_new, group_new])])
    p.write(y)

```

PyCharm 2020.2.5 available
Update...

```

commentId = []
for i in range(10001):
    commentId += [0]

for x in comment:
    y = x
    randomPost = random.randint(1, 1000)
    commentId[randomPost] = commentId[randomPost] + 1
    comment_id = commentId[randomPost]
    post_id = postTuple[randomPost][0]
    profile_id = postTuple[randomPost][1]
    part1 = y.split("values")
    part2 = part1[1].split(',')
    part2[0] = '(' + str(comment_id)
    part2[1] = post_id
    part2[2] = profile_id
    end = part1[0] + "values "
    for i in part2:
        end = end + i + ','
    end = end[0:len(end)-1]
    end = end.replace('MOCK_DATA', "post_comment")
    end = end.replace('content', "comment_content")
    newComment.write(end)

newComment.close()

```

נספח ב – קוד של השאלות

```
select gender, avg(number_of_likes), max(number_of_likes), min(number_of_likes), count(*)
from post natural join profile
group by gender;
```

```
select profile_id, first_name, last_name, post_id, number_of_likes
from post natural join profile
where number_of_likes >= (
    select avg(number_of_likes)
    from post
);
```

```
select profile_id, first_name, last_name, gender
from post natural join profile
where post_id in (select post_id
from (select post_id, count(*)
from post_comment
group by post_id
having count(*) >= (
    select avg(c)
    from (select post_id, count(*) as c
    from post_comment
    group by post_id) t
) )
) )
```

intersect

```
select profile_id, first_name, last_name, gender
from post natural join profile
where number_of_likes >= (
    select avg(number_of_likes)
    from post
);
```

```
select profile_id, first_name, last_name, gender
from profile natural join post_comment
group by profile_id, first_name, last_name, gender
having count(*) > (select avg(c) from (
    select profile_id, count(*) as c
    from post_comment
    group by profile_id
) ));
```

```
SELECT profile_id, post_id, number_of_likes
FROM post p1
WHERE number_of_likes = (
    SELECT max(number_of_likes)
    FROM post p2 natural join
        ( SELECT profile_id
        FROM profile
        minus
        SELECT profile_id
        FROM post_comment))
;
```

```

SELECT pr.profile_id, first_name, last_name, avg(number_of_likes), count(*) as num_of_post
FROM profile pr, post po
WHERE pr.profile_id = po.profile_id
GROUP BY pr.profile_id, first_name, last_name
HAVING count(*) > 1 and avg(number_of_likes) > 1500
;
SELECT profile_id, first_name, last_name, gender
FROM profile pr
WHERE NOT EXISTS(
    SELECT post_type_code
    FROM post_type pt
    WHERE NOT EXISTS(
        SELECT po.post_id
        FROM post po
        WHERE po.post_type_code = pt.post_type_code and po.profile_id = pr.profile_id))
;

```

```

SELECT profile_id , first_name, last_name, max(p.number_of_likes)
FROM post p natural join
(SELECT profile_id
FROM post_comment natural join (select post_id, post_type_code from post) T1 natural join post_type
where post_type_name = 'NORMAL') T
natural join profile
group by profile_id , first_name, last_name
;

```

נספח ג – קוד של האינדקסים

```

create index index1 ON AFEDER.PROFILE(profile_id,first_name,last_name,gender);
create index index2 ON AFEDER.POST(post_id,number_of_likes);
create index index3 ON AFEDER.POST_COMMENT(comment_id,post_id,profile_id);

```

נספח ד – קוד של השאילתות הנוספות

```
select recieverid, count(*) as num_of_suggestion
from ydemri.friendshipsuggestion join ydemri.friendshiprequest
on recieverid=reciever_id and offerid=requester_id
group by recieverid
union
select recieverid, 0
from ydemri.friendshipsuggestion join ydemri.friendshiprequest
on recieverid=reciever_id
where recieverid not in (select recieverid
from ydemri.friendshipsuggestion join ydemri.friendshiprequest
on recieverid=reciever_id and offerid=requester_id
);
```

```
select f1.recieverid, count(*) as common_suggestion
from ydemri.friendshipsuggestion f1, ydemri.friendshipsuggestion f2
where f1.offerid = f2.recieverid and f1.recieverid = f2.offerid
and not exists
(
select offerid
from ydemri.friendshipsuggestion t1
where t1.recieverid = f1.recieverid
minus
select offerid
from ydemri.friendshipsuggestion t2
where t2.recieverid = f2.recieverid
)
group by f1.recieverid;
```

```
SELECT *
FROM
(
SELECT FRIEND_ID1 AS FN1, TAG_ID, NUM AS NUM_TAGS
FROM (
(SELECT TAG_ID, COUNT(*) AS NUM
FROM BMARCUS.TAG
GROUP BY TAG_ID)

NATURAL JOIN

(SELECT DISTINCT FRIEND_ID1, TAG_ID
FROM BMARCUS.TAG, BMARCUS.FRIENDS
WHERE FRIEND_ID1 = PROFILE_ID2)))

NATURAL JOIN

(SELECT FN1, TAG_ID, COUNT(*) AS MY_FRIENDS
FROM BMARCUS.TAG NATURAL JOIN (
SELECT DISTINCT FRIEND_ID1 AS FN1, TAG_ID
FROM BMARCUS.TAG, BMARCUS.FRIENDS
WHERE FRIEND_ID1 = PROFILE_ID2) -- מי תויג ואיפה
-- מי תויג מי תייג איפה תויג ומי עוד תויג--
WHERE PROFILE_ID2 IN (SELECT FRIEND_ID2
FROM BMARCUS.FRIENDS
WHERE FRIEND_ID1 = FN1) --המתויג הראשון חבר של המתויג השני--
GROUP BY FN1, TAG_ID)

WHERE NUM_TAGS = MY_FRIENDS; -- גקיבוע לפי המתויג הראשון ומספר תיו
```

נספח ה – קוד של הגרף הראשון

```
select TO_CHAR(post_date, 'DAY') as day, sum(number_of_likes) as number_of_likes
from post
group by TO_CHAR(post_date, 'DAY')
order by number_of_likes;
```

נספח ו – קוד של הגרף השני

```
select count(*) as num_of_post, TO_CHAR(post_date, 'MONTH') as month
from post
group by TO_CHAR(post_date, 'MONTH')
order by TO_CHAR(post_date, 'MONTH');
```

נספח ז – קוד עבור שני הVIEW הראשונים

```
create or replace view trand_profile as
(
select profile_id, first_name, last_name, post_id, number_of_likes
from post natural join profile
where number_of_likes >= (
select avg(number_of_likes)
from post
)
);

select * from trand_profile ;

create view active_profile as
(
select profile_id, first_name, last_name, gender
from profile natural join post_comment
group by profile_id, first_name, last_name, gender
having count(*) > (select avg(c) from (
select profile_id, count(*) as c
from post_comment
group by profile_id
) )
);

select * from active_profile ;
```


נספח ח – קוד עבור שני הVIEW השניים

```
create view friendship_request as(
select reciever_id as user_id, count(*) as requests
from ydemri.friendshiprequest
group by reciever_id
);

select * from friendship_request;

create view active_friendship as(
select friend_id1 as user_id, count(*) as friends
from bmarcus.friend
group by friend_id1
);

select * from active_friendship;
```

נספח ט – קוד עבור הפונקציות

```
create or replace function equalNamae(user_name in VARCHAR2) return number is
    FunctionResult number;
begin

    select count(*)
    into FunctionResult
    from afeder.profile p
    where p.first_name = user_name or p.last_name = user_name;

    return(FunctionResult);
end equalNamae;
```

```
create or replace function howFriednsTag(profile_id in number) return number is
    FunctionResult number;
begin

    select count(*)
    into FunctionResult
    from bmarcus.tag t join bmarcus.friends f on f.friend_id2 = t.profile_id1
    where f.friend_id1 = profile_id and t.profile_id2 = profile_id;

    return(FunctionResult);
end howFriednsTag;
```

נספח י – קוד עבור הפרוצדורות

```

create or replace procedure allInfo(profile_id in INTEGER) is
    proId INTEGER := profile_id;
    numOfPost number;
    numOfLikes number;
    numOfFriends number;
    numOfShare number;
    numOfTags number;
    numOfSuggestion number;
    numOfRequest number;
    numOfNotification number;
begin
    select count(*)
    into numOfRequest
    from ydemri.friendshiprequest f1
    where f1.reciever_id = proId;

    select count(*)
    into numOfSuggestion
    from ydemri.friendshipsuggestion s1
    where s1.recieverid = proId;

    select count(*)
    into numOfNotification
    from ydemri.notification n
    where n.profile_id = proId;

    select count(*)
    into numOfShare
    from bmarcus.share_post s2
    where s2.profile_id = proId;

    select count(*)
    into numOfFriends
    from bmarcus.friends f2
    where f2.friend_id1 = proId;

    select count(*)
    into numOfTags
    from bmarcus.tag t
    where t.profile_id2 = proId;

    select sum(p1.number_of_likes)
    into numOfLikes
    from afeder.post p1
    where p1.profile_id = proId;

    IF numOfLikes is null then
        numOfLikes := 0;
    end if;

    select count(*)
    into numOfPost
    from afeder.post p2
    where p2.profile_id = proId;

    dbms_output.put_line('-----');
    dbms_output.put_line('                                USER INFO');
    dbms_output.put_line('ID | POST | LIKES | SHARES | TAGS | FREINDS | SUGGESTION | REQUEST | NOTIFICATION');
    dbms_output.put_line('-----');
    dbms_output.put_line(proId || ' | ' || numOfPost || ' | ' || numOfLikes || ' | ' || numOfShare ||
        ' | ' || numOfTags || ' | ' || numOfFriends || ' | ' || numOfSuggestion ||
        ' | ' || numOfRequest || ' | ' || numOfNotification );

end allInfo;

```

```
create or replace procedure thePop is
```

```
CURSOR top_users IS
select t2.pro_id, t2.c
from
(
select t1.profile_id2 as pro_id , count(*) as c, ROW_NUMBER() OVER (ORDER BY count(*)) as rnum
from bmarcus.tag t1
group by t1.profile_id2
) t2

where t2.rnum >=
(
select count( distinct t1.profile_id2)
from bmarcus.tag t1
) * 0.9

order by t2.c desc;

top_user top_users%rowtype;

BEGIN
open top_users;
fetch top_users into top_user;
if top_users%notfound then
dbms_output.put_line('error');
elsif top_users%found then

dbms_output.put_line('-----USER INFO-----');
dbms_output.put_line('ID | NUM OF TAGS ');
dbms_output.put_line('-----');
loop
exit when top_users%notfound ;
dbms_output.put_line(top_user.pro_id || ' | ' || top_user.c);
fetch top_users into top_user;
end loop;
close top_users;
end if;

end thePop;
```