

מיני פרויקט בבסיסי נתונים

מגישים
שגב בורשטיין & יקיר דמרי

מרצה: יאיר גולדשטיין | 15.6.2021

תוכן

2 מבוא
3 אפיון הטבלאות
3 תרשים ERD
4 תיאור הישויות והקשרים
4 ישויות
4 קשרים
5 נרמול הטבלאות
5 תרשים DSD
5 יצירת הטבלאות
6 אכלוס טבלאות
7 שאילתות
7 שאילתות SQL
11 אינדקסים
14 הרשאות
15 Views
19 תרשימים
21 פונקציות ופרוצדורות
21 Functions
22 Procedures
23 סיכום

מבוא

הרעיון שנבחר במהלך הדיון בכיתה הוא לממש בסיס נתונים המספק את הפונקציונליות על מנת לתמוך ברשת חברתית. הרשת שלנו מאפשרת למשתמש מגוון עולמות לנוחותו כגון פיתוח קשרים עם משתמשים אחרים ותקשורת עמהם, גישה לקבוצות ופגישות חברתיות המכילות תוכן משותף וכמובן היכולת לשתף את דעתו האישית של המשתמש עם שאר המשתמשים ברשת ולשמוע את דעתם על כך.

על כן, אלו הן המחלקות השונות המנהלות את הרשת לרוחבה:

- **פרופיל משתמש**
- **פרסום פוסט חדש**
 - סוג הפוסט
 - תגובה על פוסט
- **התקהלות קבוצתית**
 - אירוע
 - קבוצה
 - עמוד עסקי
 - סוג העסק
- **התראה**
- **התפתחות חברתית**
 - הצעת חברות
 - בקשת חברות
- **חבר**
- **נראות**
- **פעולות משתמש אקטיבי**
 - לייק
 - שיתוף
 - תיוג

אפיון הטבלאות

כחלק מתפקוד של רשת חברתית, יש צורך לספק למשתמש את האפשרות ליצור קשרים חדשים בין משתמשים קיימים במערכת, עדכנו בתהליכים שנעשים סביבו, חברים חדשים שהוא עשוי להכיר דרך קשרים שהוא כבר פיתח ועוד.

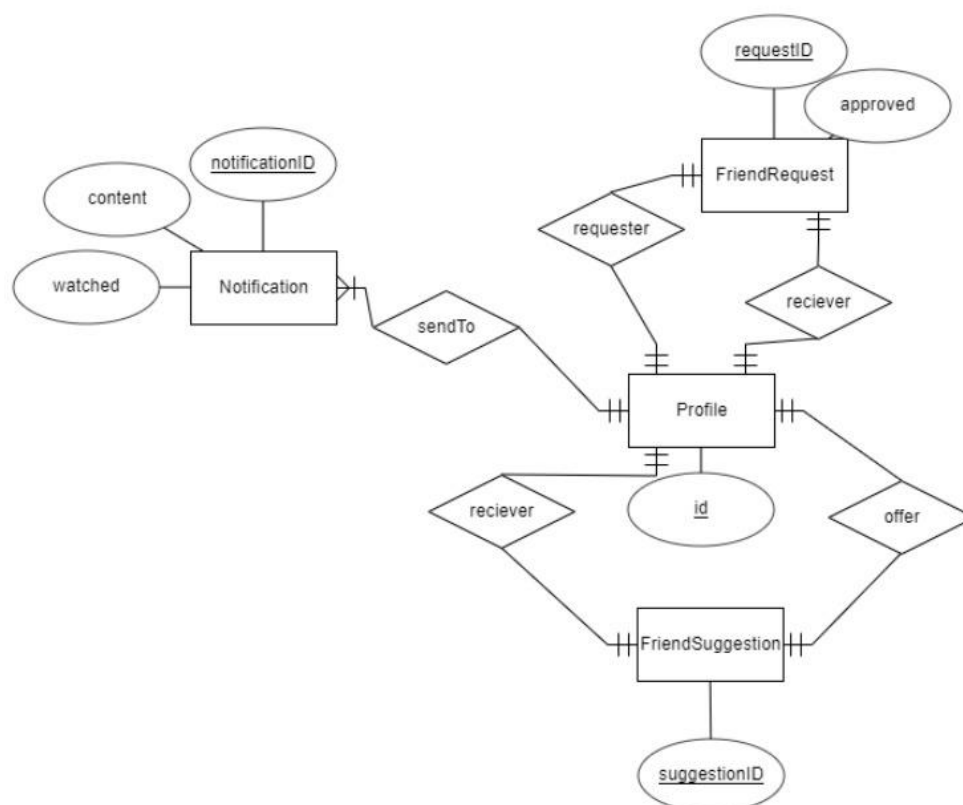
כחלק מתפקידנו במערכת הכוללת, לקחנו על עצמנו את יישומם של הטבלאות הבאות:

- **התראות** משתמש בעדכונים מהנעשה בסביבתו (פוסטים חדשים שהועלו, הודעות חדשות בקבוצות בהם הוא לוקח חלק ועוד)
- **בקשת חברות** באמצעותה המשתמש מסוגל להרחיב את מעגל הקשרים שלו.
- **הצעת חברות**. המלצות מטעם המערכת בדבר חברים חדשים שכדאי למשתמש להכיר בהתבסס על נתונים קודמים.

נתונים אלו מרוכזים בטבלאות שלנו ובאמצעות אינטראקציה מספקים למנהל המערכת אפשרות צפייה ומעקב במתרחש ברשת.

תרשים ERD

כפי שהזכרנו, במחלקה שלנו, ישנן 3 ישויות: התראות, בקשת חברות והצעת חברות. בשלב הראשון יצרנו תרשים ERD שיתאר את הקשרים בין הישויות הללו ואת התכונות שלהן. בעזרת תרשים זה הבנו את התנהלות המערכת ואת האפשרויות השונות העומדות לרשותנו מבחינת פונקציונליות הניתנת להציע למנהל המערכת. (מנהל מערכת בסיס המידע).



תיאור הישויות והקשרים

ישויות

- Notification – ישות זאת אחראית על כל ההתראות בסביבת המשתמש ברשת החברתית.
ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.
 - notification_id – מספר מזהה של ההתראה (PK).
 - profile_id – מזהה הפרופיל שקיבל את ההתראה (FK).
 - content – תוכן ההתראה.
 - watched – האם ההודעה נצפתה.
- FriendshipSuggestion – ישות זאת אחראית על כל הצעות החברות שהמערכת מסוגלת להציע למשתמש בהתבסס על נתונים ופעולות קודמות שהמשתמש ביצע.
ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.
 - suggestion_id – מספר מזהה של ההצעה (PK)
 - profile1_id – מספר מזהה של הפרופיל המוצע (FK)
 - profile2_id – מספר מזהה של הפרופיל המקבל את ההצעה (FK).
- Friendship-Request – ישות זאת אחראית על כל בקשות החברות שהמשתמש מפנה כלפי משתמשים אחרים ברשת החברתית על מנת להגדיל את הסביבה האישית שלו.
ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.
 - request_id – מספר מזהה של הבקשה (PK).
 - profile1_id – מספר מזהה של המשתמש המקבל (FK).
 - profile2_id – מספר מזהה של המשתמש המציע (FK)
 - isApproved – האם הבקשה אושרה. [כן/לא/מחכה לתשובה].

קשרים

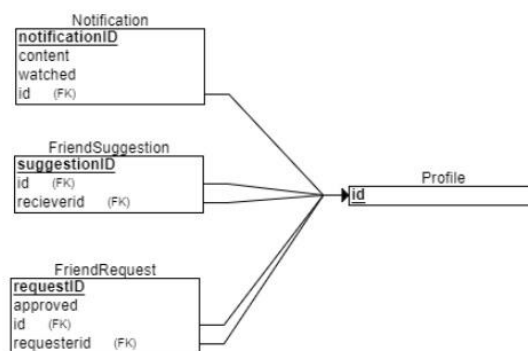
- **sendto** – הקשר בין Notification לבין Profile. הקשר הינו חזק, כיוון שמקשר בין שתי ישויות חזקות. הקשר הוא M:1 משום שיכול להיות הרבה התראות למשתמש בודד אבל לא יכול להיות הרבה משתמשים להתראה בודדת.
- **Receiver** – הקשר בין profile לבין friendshipSuggestion. הקשר הינו חזק, כיוון שמקשר בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל הצעה יכול להיות רק אדם אחד שמקבל אותה (אלו הצעות מותאמות אישית פר פרופיל).
- **Receiver** – הקשר בין profile לבין friendshipRequest. הקשר הינו חזק, כיוון שמקשר בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל בקשה יכול להיות רק אדם אחד שמקבל אותה.
- **Requester** – הקשר בין profile לבין friendshipRequest. הקשר הינו חזק, כיוון שמקשר בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל בקשה יכול להיות רק אדם אחד ששולח אותה.
- **Offer** – הקשר בין profile לבין friendshipSuggestion. הקשר הינו חזק, כיוון שמקשר בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל הצעה יכול להיות רק אדם אחד שמציעים אותו בה.

נרמול הטבלאות

היחסים עומדים ב- 3NF וב- BCNF : מכיוון שבכל טבלה, התלויות הפונקציונאליות הלא-טריוויאליות הן מהמפתח אל תכונות נוספות לכן מתקיים שלכל $X, X \rightarrow Y$ הוא מפתח ולכן הם עומדים בתנאים.

תרשים DSD

על פי תרשים ה- ERD ועל ידי הבנת הקשרים בין הישויות, יצרנו תרשים DSD עבור החלק שלנו במערכת: מחלקת מתקני המלון.



יצירת הטבלאות

אחרי שהבנו כיצד בסיס הנתונים צריך להראות בצורה מדויקת, מה תכיל כל טבלה ומהם הקשרים בין כל הטבלאות, ניגשנו ליצירת הטבלאות בפועל בעזרת פקודות create table.

יצרנו קוד לייצור הטבלאות באמצעות SQL export של האתר erdPlus, יצרנו קובץ SQL ואז העתקנו את קוד ה- SQL של כל טבלה אל תוכנת ה-plsql לשם יצירת הטבלאות בפועל.

```

create table Notification (
    notification_id number(9) primary key,
    profile_id number(9) foreign key references profile(profile_id)
    content varchar(256) not null,
    watched bit(1) not null
);

create table FriendshipRequest (
    requ_id number(9) primary key,
    requester_id number(9) foreign key references profile(profile_id),
    reciever_id number(9) foreign key references profile(profile_id),
    approved bit(1)
);

create table FriendshipSuggestion (
    suggestionID number(9) primary key,
    recieverID number(9) foreign key references profile(profile_id),
    offerID number(9) foreign key references profile(profile_id)
);
    
```

אכלוס טבלאות

לאחר יצירת הטבלאות, אנו נדרשים למלא אותן במידע. על מנת לעשות זאת, השתמשנו במגוון על מנת לאכלס את הטבלאות. להלן הכלים והתוצאות:

- Data Generator מתוכנת ה-PL/SQL להכנסת ערכים לטבלת ה-FriendshipRequest:

FRIENDSHIPREQUEST				
Owner	Table	Number of records		
YDEMRI	FRIENDSHIPREQUEST	20000		
Name	Type	Size	Data	
REQU_ID	NUMBER	9	Sequence(0)	
REQUESTER_ID	NUMBER	9	List(select profile_id from afeder.profile)	
RECIEVER_ID	NUMBER	9	List(select profile_id from afeder.profile)	
APPROVED	NUMBER	1	List(0, 1,)	

ניתן לראות כי בעמודה Data Approved הוא List(0, 1). זוהי לא טעות. הפסיק האחרון מסמל שגם null הוא אופציה עבור עמודה זו.

- באמצעות האתר ¹generatedata.com להכנסת ערכים לתוך Notification:

DATA SET

Order	Table Column	Data Type	Examples	Options	Help	Del
1	notification_id	Auto-increment	1, 2, 3, 4, 5, 6...	Start at: 0 Increment: 1 Placeholder string:	?	
2	profile_id	Number Range	No examples available.	Between 0 and 9999	?	
3	content	Random Number of Words	No examples available.	Start with "Lorem Ipsum..." Generate #1 to #15 words	?	
4	watched	Boolean	0 or 1	0 1	?	

EXPORT TYPES

CSV Excel HTML JSON LDIF Programming Language SQL XML - hide data format options

Database table name: Notification

Database Type: Oracle

Misc Options:

☐ Include CREATE TABLE query

☐ Include DROP TABLE query

☐ Enclose table and field names with backquotes

Statement Type: ☒ INSERT ☐ INSERT IGNORE ☐ UPDATE

INSERT batch size: 10

Primary Key: ☐ None ☒ Add default auto-increment column

Generate 100 rows ☒ Generate in-page ☐ New window/tab ☐ Prompt to download ☐ Zip?

כמו שאפשר לראות, באתר זה ניתן להכין רק 100 מופעים שונים לטבלה. לאור זאת שהיינו זקוקים ל-20 אלף ערכים, כתבנו script קצר ב-python הלוחץ על כפתור ה-Generate באופן אוטומטי וכך קיבלנו את מספר הערכים הרצוי.

- סקריפט בפייטון המייצר קובץ עם שאליות Insert. להלן הסקריפט:

```
import random

insert_query = 'INSERT INTO FriendshipSuggestion (suggestionid, recieverid, offerid) VALUES (%d, %d, %d);\n'

def createInsertQuery(rows=20_000):
    with open("insertQuery.sql", "w") as SQL_file:
        for i in range(rows):
            SQL_file.write(insert_query % (i, random.randint(0, 9999), random.randint(0, 9999)))

createInsertQuery()
```

תוצאות סקריפט זה וקובץ הפלט GenerateData נמצאים בגיט.

¹ <https://www.generatedata.com/>

שאלות

שאלות SQL

שאלתה 1 - כל הבקשות שמחכות לתשובה:

התרחיש: כחלק מהרצון של ההנהלה לדעת את מצב זמינות המשתמשים, ופנאי השימוש במערכת, ההנהלה ביקשה לדעת את כמות ההתראות שמחכות למענה בקרב המשתמשים. לכן כתבנו שאלתה המחזירה טבלה של כל הבקשות שמחכות למענה.

הגדרה: במערכת המידע שלנו, 0/1 מסמנים התקבל/נדחה בהתאמה ו-null מסמן 'ממתין למענה'.

הקוד:

```
-- query 1 : all pending requests [TRIVIAL]
select requ_id, requester_id, reciever_id
from friendshiprequest
where approved is null;
```

תוצאה וזמן ריצה:

התוצאה: 6616 שורות ב-6.427 שניות

REQU_ID	REQUESTER_ID	RECIEVER_ID
1	787	2550
2	789	8448
3	795	2860
4	800	4422
5	801	8434
6	802	9907
7	804	2960
8	806	5666
9	811	7777
10	817	751
11	823	4877
12	825	2520
13	828	4689
14	829	1328
15	830	9000
16	831	4902
17	834	40

0:12 ydemri@labdbwin 6616 rows selected in 6.427 seconds

שאלתה 2 - הפרופילים הכי פעילים:

התרחיש: ההנהלה החליטה לפנק במתנה את כל המשתמשים הפעילים ביותר - בסט סטיקרים חדש. לכן ההנהלה ביקשה רשימה של כל המשתמשים הפעילים. כתבנו שאלתה המבצעת את הבקשה הנ"ל.

הגדרה: פרופיל שמקבל יותר מ-5 התראות נחשב 'משתמש פעיל'.

הקוד:

```
-- query 2 : most active (more then X notifications) profiles [TRIVIAL]
select profile_id, first_name, last_name, count(*) as notif_num
from notification N natural join afeder.profile P
group by profile_id, first_name, last_name
having count(*) > 5;
```

תוצאה וזמן ריצה:

התוצאה: 166 שורות ב-6.535 שניות

PROFILE_ID	FIRST_NAME	LAST_NAME	NOTF_NUM
1	7686	Anjelica Larter	6
2	1173	Rawlins Jane	7
3	8457	Tia Weisz	6
4	1617	Lizzy Aglukark	6
5	2842	Sylvester Thornton	6
6	7008	Natacha Butier	7
7	3591	Parker Linney	6
8	2591	Gary Bachman	6
9	9604	Avenged Swinton	6
10	4224	Lindsay Adler	6
11	993	Mary Beth Elwes	6
12	6405	Sissy Uggams	6
13	6075	Dan Bacharach	6
14	3717	Mary Holly	6
15	8470	Lamelle Shearer	8
16	3570	Tori von Sydow	7
17	2135	Dar Keith	6

0:12 ydemri@labdbwin 166 rows selected in 6.535 seconds

שאלתה 3 - הפרופילים המומלצים ביותר:

התרחיש: כחלק מניסוי כלים של אלגוריתם יצור ההמלצות בניסיון לשפרו במידת הצורך, ביקשו מרכז הפיתוח רשימה של כל המשתמשים המומלצים ביותר.

לכן כתבנו שאלתה המחזירה את הפרופילים המומלצים ביותר ע"י המערכת בקרב משתמשים שונים ברשת החברתית.

הגדרה: פרופילים שהוצעו יותר מ-5 פעמים.

הקוד:

```
-- query 3 : most recommended profiles [TRIVIAL]
select offerid, first_name, last_name, count(*) as rec_num
from friendshipsuggestion FS join afeder.profile P on offerid = profile_id
group by offerid, first_name, last_name
having count(*) > 5;
```

תוצאה וזמן ריצה:

התוצאה: 156 שורות ב6.663 שניות

	OFFERID	FIRST_NAME	LAST_NAME	REC_NUM
1	9471	Liam	Farris	6
2	2894	Robert	Washington	7
3	852	Lila	Singletary	6
4	458	Nikki	Lorenz	6
5	3902	Ahmad	Solido	6
6	9989	Pamela	Weisberg	6
7	475	Trey	Tennison	6
8	4241	Kris	Frakes	6
9	9274	Dave	Nunn	6
10	639	Ruth	Mraz	6
11	7274	Anthony	Barry	7
12	4276	Rowan	Johansson	6
13	3183	Derek	Kutcher	6
14	2239	Christmas	Gaines	7
15	1348	Debi	Richards	7
16	6389	Jeanne	Tucker	8
17	7028	Mae	Bransagh	7

0:12 ydemni@labdbwin 156 rows selected in 6.633 seconds

שאלתה 4 – פרופילים שנחשדים כבוטים:

התרחיש: חוקרי אבטחת המידע של הרשת החברתית מנסים להבחין בבוטים חשודים שמפיצים בקשות חברות בצורה מוגזמת על מנת ליצור פרסום בפני קהל אותו הם יכולים להשכיר ללקוחות חיצוניים.

לכן כתבנו שאלתה המחזירה טבלה של כל המשתמשים החשודים. מצב שכזה יועבר לניתוח מעמיק יותר ותפיסת המפעילים.

הגדרה: משתמשים אשר שלחו יותר מ-5 בקשות.

הקוד:

```
-- query 4 : most desprate profiles [TRIVIAL]
select requester_id, first_name, last_name, count(*) as req_num
from friendshiprequest FR join afeder.profile P on requester_id = profile_id
group by requester_id, first_name, last_name
having count(*) > 5;
```

תוצאה וזמן ריצה:

התוצאה: 173 שורות ב6.752 שניות

	REQUESTER_ID	FIRST_NAME	LAST_NAME	REQ_NUM
1	1868	Viggo	Ellis	7
2	8078	Irene	Adams	6
3	1654	Suzy	Heald	6
4	5117	Bill	James	6
5	7168	Val	Boorem	7
6	4143	Boz	Lightfoot	6
7	7484	Carole	Roundtree	7
8	6796	Harvey	Chaplin	6
9	6459	Susan	Barry	8
10	5176	Jonny Lee	O'Donnell	6
11	5013	Dean	McNariand	6
12	5251	Jodie	Tirantino	6
13	9785	Nikki	Vorn	6
14	9276	Gabriel	Pastore	9
15	3030	Tara	Pepper	6
16	2894	Robert	Washington	6
17	7519	Christmas	Hagar	7

0:12 ydemni@labdbwin 173 rows selected in 6.752 seconds

שאלתה 5 - הצעות החברות המוצלחות:

התרחיש: ההנהלה חוששת שאלגוריתם המלצת החברים איננו איכותי ומאפשר ליצור קשרים שמתקיימים בפועל. רוצים לבדוק שאכן הצעה לאדם מסוים יוצרת קשר הדדי בין שני משתמשים. לכן כתבנו שאלתה המחזירה את כל הצעות החברות המוצלחות שנעשו ע"י המערכת.

הגדרה: מוצלח: הפרופיל שאליו נשלחה ההצעה, שלח את הבקשה לפרופיל המוצע, והנ"ל אישר את הבקשה.

הקוד:

```
-- query 5 : all good suggestions
select FS.SUGGESTIONID,
       FS.RECIEVERID,
       P1.FIRST_NAME as RECIEVER_FNAME,
       P1.LAST_NAME as RECIEVER_LNAME,
       FS.OFFERID,
       P2.FIRST_NAME as OFFER_FNAME,
       P2.LAST_NAME as OFFER_LNAME
from friendshipsuggestion FS
join
  afeder.profile P1
on FS.RECIEVERID = P1.PROFILE_ID
join afeder.profile P2
on FS.OFFERID = P2.PROFILE_ID
where exists (select *
              from friendshiprequest FR
              where FS.RECIEVERID = FR.REQUESTER_ID
                  and FS.OFFERID = FR.RECIEVER_ID
                  and FR.Approved = 1);
```

תוצאה וזמן ריצה:

התוצאה: 240 שורות ב6.883 שניות

	SUGGESTIONID	RECIEVERID	RECIEVER_FNAME	RECIEVER_LNAME	OFFERID	OFFER_FNAME	OFFER_LNAME
1	20000	6961	Rodney	Carlisle	6923	Lorraine	Cale
2	20624	5080	Jonny	O'Neill	6955	Kitty	Ajona
3	20580	8783	Burt	Brown	5456	Lloyd	Wiest
4	20046	5700	Denise	Sainte-Marie	5478	Vin	Levert
5	20866	765	Debbie	Carrington	5516	Chi	Singh
6	20369	778	Deborah	Woods	5595	Anna	Imperioli
7	20559	1075	Dean	Saxon	5612	Milla	Satnani
8	20708	2336	Louise	Chandler	5619	Javon	Sledge
9	20804	5092	Etta	Pinzner	5698	Meredith	Hirsch
10	20551	2431	Donal	Shatner	5720	Avenged	Moore
11	20786	245	Penelope	MacDonald	5751	Nickel	Stevenson
12	20997	9092	Colleen	Farina	5780	Faye	Gleeson
13	20001	6981	Vendetta	McCormack	5810	Buffy	McDowell
14	20941	2739	Ming-Na	Webb	5839	Horace	Paymer
15	20335	6168	Nickel	Guzman	5934	Ronnie	Stewart
16	20115	7401	Helen	Manning	5955	Kirk	Duschel
17	20009	9463	Lorraine	Lynne	5952	Avril	Sirtis

שאלתה 6 - כל ההתראות על המלצות/בקשות:

התרחיש: משתמשים רבים העלו פנייה בפני שירות הלקוחות של הרשת אודות התראות סרק לאחר שמקבלים הצעת חברות או שולחים אחת. ההנהלה ביקשה לנתח התראות סרק אלו ולהבין את מקור הבאג. לכן כתבנו שאלתה המחזירה את כל ההתראות הקשורות להמלצות חברות או בקשות חברות.

הקוד:

```
-- query 6 : all watched notifications on friend request/suggestion
select *
from notification N
where N.Watched = 1 and
       N.PROFILE_ID in ((select FR.RECIEVER_ID
                        from friendshiprequest FR)
                        union
                        (select FS.RECIEVERID
                        from friendshipsuggestion FS));
```

תוצאה וזמן ריצה:

התוצאה: 9807 שורות ב12.549 שניות

	NOTIFICATION_ID	PROFILE_ID	CONTENT	WATCHED
1	8575	1	Anne officis omnis deleniti quos est in occaecat.	1
2	13739	2	Tony quod.	1
3	458	2	Devon maiores voluptate excepteur et dolor et.	1
4	18712	3	Anita cillum dolor rerum corrupti sint.	1
5	4513	5	Mili recusandae id et esse et maxime doloribus.	1
6	14112	6	Joan itaque at id quidem quod est ex repellendus tempore.	1
7	12322	6	Solomon voluptatum mollit repellat ut similique saepe.	1
8	10865	7	Cate est esse.	1
9	1935	7	Denis cupiditate lorem officis voluptate id non.	1
10	15202	8	Teena laboris.	1
11	13262	8	Christopher.	1
12	10552	9	Coley optio quos sint autem accusamus.	1
13	7710	9	Rawlins qui hic animi id similique.	1
14	17437	11	Rhys temporibus dolorum hic dolor tempore.	1
15	18511	14	Julio fugiat omnis deserunt amet eligendi.	1
16	16015	15	Morris et nobis fugiat occaecati assumenda.	1
17	12685	15	Ron Duis nobis dolorum amet laborum culpa autem in cupidatat.	1

שאלתה 7 - כל הפרופילים הפופולריים:

התרחיש: הנהלת הרשת רוצה לעלות פרסומת חדשה למשך משתמשים חדשים. לפיכך היא מעוניינת להזמין את המשתמשים הפופולריים ביותר (בדר"כ אלו מפורסמים) על מנת שייציגו את היתרונות וההנאה בשימוש ברשת החברתית.

לכן כתבנו שאלתה המחזירה את כל אותם משתמשים מפורסמים.

הגדרה: פופולרי – פרופיל ששלחו אליו יותר מ-5 בקשות חברות.

הקוד:

```
-- query 7 : all popular profiles - more then 5 requests
select FR.RECIEVER_ID, first_name, last_name, count(*) as req_num
from friendshiprequest FR join afeder.profile P on FR.RECIEVER_ID = P.PROFILE_ID
group by FR.RECIEVER_ID, first_name, last_name
having count(*) > 5;
```

תוצאה וזמן ריצה:

התוצאה: 176 שורות ב-12.624 שניות

	RECIEVER_ID	FIRST_NAME	LAST_NAME	REQ_NUM
1	6744	Nanci	Dalley	6
2	8934	Rufus	Kravitz	6
3	8967	Thomas	Giamatti	6
4	4201	Nickel	Seagel	6
5	8105	Samantha	Gayle	7
6	8160	Winona	Matthau	7
7	5117	Bill	James	6
8	526	Jessica	Candy	6
9	9111	Bernard	Turturo	6
10	7994	Adina	Rizzo	6
11	5952	Avril	Sirtis	6
12	9662	Jose	Hatchet	6
13	1299	Boz	Griggs	6
14	7700	Rosanna	Rubinek	6
15	4185	Gary	Cherny	6
16	3789	Joely	Robbins	6
17	3284	Rosanne	Parr	7

שאלתה 8 - פעילות חשודה:

התרחיש: בתקופה האחרונה עלו פניות רבות בקרב אגף אבטחת המשתמש בשירות הלקוחות של השרת אודות משתמשים שמטרידים משתמשות. ההנהלה, יחד עם המשטרה רוצים לתפוס את החשודים ולהביאם אל עונשם.

לשם כך כתבנו שאלתה המחפשת את כל המשתמשים הבעייתיים הללו ומחזירה את פרטיהם.

הגדרה: פרופיל ששלח יותר מ-3 הצעות חברות לנשים שונות.

הקוד:

```
---- query 8 : suspicious activity
select P.profile_id, first_name, last_name
from afeder.profile P
where P.GENDER = 'M' and
(select count(*)
 from friendshiprequest FR join afeder.profile P1 on FR.RECIEVER_ID = P1.PROFILE_ID
 where FR.REQUESTER_ID = P.PROFILE_ID and P1.GENDER = 'F') >= 3
```

תוצאה וזמן ריצה:

התוצאה: 114 שורות ב-12.950 שניות

	PROFILE_ID	FIRST_NAME	LAST_NAME
1	6994	Nikki	Aida
2	5439	Sean	LaMond
3	5453	Will	Pearce
4	5485	Dylan	Sweet
5	5546	Nils	Spacey
6	5597	Don	Reilly
7	5623	Daryle	Davison
8	5792	Kazem	Shandling
9	5847	Taryn	Brooke
10	5893	Rose	Kutcher
11	5927	Chuck	Arnold
12	6125	Benjamin	Gold
13	6190	Toni	Colon
14	7771	Stephen	McBride
15	7945	Jude	Tah
16	9771	Garland	Blackwell
17	9690	Tea	Tippe

אינדקסים

אינדקסים עוזרים למצוא במהירות גדולה יותר נתונים שנשמרו בטבלאות בבסיס הנתונים. אפשר לדמות את האינדקסים כמו מראה מקום בספר. במקום שנקרא את כל הספר כדי למצוא את מה שאנחנו מחפשים נלך למראה מקום שיראה לנו את כל המקומות שבהם מוזכר הנושא שאנחנו מחפשים. השימוש באינדקסים יחסוך לנו זמן ויהפוך את תהליך החיפוש ליעיל יותר. מהבחינה הזו האינדקסים בטבלאות של ה-SQL זהים לאינדקס בספר.

במידה ולא נגדיר אינדקס לטבלה אז בכל שאילתה על הטבלה השאילתה תגרום למעבר על כל הרשומות בטבלה עד שתמצא את כל הרשומות העונות למה שחיפשנו. כשנגדיר אינדקס מתאים אז החיפוש יהיה מהיר יותר כי הפניה לבסיס הנתונים תגרום לזה שמנוע החיפוש בבסיס הנתונים יפנה קודם לאינדקס וילך לרשומות המתאימות על פי מה שרשום באינדקס.

האינדקסים שכתבנו הם (גודלן של הטבלאות – 20K נתונים):

```
create index idx_requester on ydemri.Friendshiprequest(requester_id);
create index idx_reciever on ydemri.Friendshipsuggestion(recieverid);
create index idx_notf_profile on ydemri.Notification(profile_id);
```

הסיבה לבחירת מאפייני האינדקסים הללו היא לאור זאת שבשאילתות שכתבנו, אלו הן העמודות המבוקשות ביותר. דהיינו, הסינון של מופעי הטבלה מתבצע בעיקר על ידם. לדוג' אפשר לראות בשאילתה 8 שמתבצע סינון על בסיס המידע המגיע מעמודה requester_id:

```
---- query 8 : suspicious activity
select P.profile_id, first_name, last_name
from afeder.profile P
where P.GENDER = 'M' and
(select count(*)
 from friendshiprequest FR join afeder.profile P1 on FR.RECIEVER_ID = P1.PROFILE_ID
 where FR.REQUESTER_ID = P.PROFILE_ID and P1.GENDER = 'F') >= 3
```

עוד דוגמא היא שאילתה מספר 2 בה מבצעים group by על profile_id של Notification:

```
-- query 2 : most active (more than X notifications) profiles [TRIVIAL]
select profile_id, first_name, last_name, count(*) as notf_num
from notification N natural join afeder.profile P
group by profile_id, first_name, last_name
having count(*) > 5;
```

חשוב להדגיש כי עמודות אינן ממוינות בשום צורה. לדוגמא, נציג פלט לשאילתה הפשוטה:

SELECT * FROM FriendshipSuggestion

	SUGGESTIONID	RECIEVERID	OFFERID
1	861	8952	2736
2	862	9515	3599
3	863	7181	6536
4	864	9512	9681
5	865	9860	7157
6	866	190	5903
7	867	9756	2070
8	868	7730	2158
9	869	7205	8104
10	870	3063	6994
11	871	8258	3713
12	872	9134	6638
13	873	8097	8837
14	874	181	6248
15	875	7547	7998

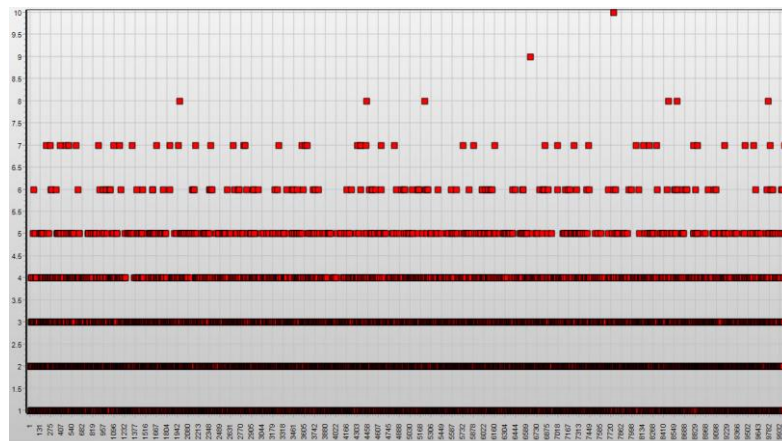
בנוסף, עמודות אלה מלאות בתוכן שסביר מאוד שחוזר על עצמו:

- Notification(profile_id) מייצג את הפרופיל שקיבל את הנוטיפיקציה. ייתכן מאוד כי פרופיל אחד קיבל מספר נוטיפיקציות.
- FriendshipSuggestion(recieverid) מייצג את הפרופיל שקיבל את ההצעה לחברות כלשהי מהמערכת. ייתכן מאוד כי פרופיל אחד קיבל מספר הצעות.
- FriendshipRequest(requester_id) מייצג את הפרופיל ששלח את בקשת החברות. ייתכן מאוד כי פרופיל אחד שלח מספר בקשות.

על מנת לחזק טענה זו, נציג גרף לדוגמא של התפלגות מספר הנוטיפיקציות כפונקציה של המשתמש. כלומר מספר הנוטיפיקציות עבור כל משתמש. השאלתה:

```
select N.PROFILE_ID, count(*)
from Notification N
group by N.PROFILE_ID
order by N.PROFILE_ID
```

את התוצאה נציג בscatter plot:



קל לראות כי בדוגמא זו, קשה מאוד לחזות את מספר הנוטיפיקציות עבור פרופיל נתון והתפלגות דיי אחידה. באופן דמה כך ההתפלגות של כל העמודות שנבחרו לאינדקסים.

לפיכך, חישוב מקדים באמצעות אינדקסים על עמודות אלה מאפשר לצמצם את זמן החיפוש בצורה משמעותית. כעת נציג שיפור של שלושה שאלות מדגמיות מתוך שמונת השאלות שצינו בחלק הקודם (עמוד 10).

1. שאלתה מס' (1):

	REQU_ID	REQUESTER_ID	RECIEVER_ID
1	787	2550	8261
2	789	8448	8874
3	795	2860	6495
4	800	4422	2300
5	801	8434	5125
6	802	9903	9671
7	804	2960	1376
8	806	5666	9056
9	811	7777	7194
10	817	751	2913
11	823	4877	5834
12	825	2520	2534
13	828	4689	4731
14	829	1328	6744
15	830	9000	1384
16	831	4902	2213
17	834	40	7378

66:20 0:09 ydemri@labdbwin 6616 rows selected in 3.832 seconds

- לפני: 6.427 (sec)
- אחרי: 3.832 (sec)
- מסקנה: מדובר ב-2.595 (sec), 40.377% שיפור.

2. שאילתה מס' (6):

	NOTIFICATION_ID	PROFILE_ID	CONTENT	WATCHED
1	8575	1	Anne officiis omnis deleniti quos est in occaecat.	1
2	13739	2	Tony quod.	1
3	458	2	Devon maiores voluptate excepteur et dolor et.	1
4	18712	3	Anita cillum dolor rerum corrupti sint.	1
5	4513	5	Mili recusandae id et esse et maxime doloribus.	1
6	14112	6	Joan itaque at id quidem quod est ex repellendus tempore.	1
7	12322	6	Solomon voluptatum mollit repellat ut similique saepe.	1
8	10865	7	Cate est esse.	1
9	1935	7	Denis cupiditate lorem officiis voluptate id non.	1
10	15202	8	Teena laboris.	1
11	13262	8	Christopher	1
12	10552	9	Coley optio quos sint autem accusamus.	1
13	7710	9	Rawlins qui hic animi id similique.	1
14	17437	11	Rhys temporibus dolorum hic dolor tempore.	1
15	18511	14	Julio fugiat omnis deserunt amet eligendi.	1
16	16015	15	Morris et nobis fugiat occaecati assumenda.	1
17	12685	15	Ron duis nobis dolorum amet laborum culpa autem in cupidatat.	1

- לפני: 12.549 (sec).- אחרי: 9.832 (sec).- מסקנה: מדובר ב-2.717 (sec), 21.652% שיפור.

3. שאילתה מס' (8):

	PROFILE_ID	FIRST_NAME	LAST_NAME
1	6934	Nikki	Alda
2	5439	Sean	LaMond
3	5453	Will	Pearce
4	5485	Dylan	Sweet
5	5546	Nils	Spacey
6	5597	Don	Reilly
7	5623	Daryle	Davison
8	5792	Kazem	Shandling
9	5847	Taryn	Brooke
10	5893	Rose	Kutcher
11	5927	Chuck	Arnold
12	6125	Benjamin	Gold
13	6190	Toni	Colon
14	7771	Stephen	McBride
15	7945	Jude	Tah
16	9771	Garland	Blackwell
17	9890	Tea	Tippe

- לפני: 12.950 (sec).- אחרי: 9.963 (sec).- מסקנה: מדובר ב-2.987 (sec), 23.066% שיפור.

להלן תקציר השיפורים בכל 8 השאילתות:

```

Resultes:
Query 1 - 6616 rows in 3.832 sec (6.427 before)
Query 2 - 166 rows in 3.891 sec (6.535 before)
Query 3 - 156 rows in 3.953 sec (6.663 before)
Query 4 - 173 rows in 4.044 sec (6.752 before)
Query 5 - 240 rows in 4.146 sec (6.883 before)
Query 6 - 9807 rows in 9.832 sec (12.549 before)
Query 7 - 176 rows in 9.923 sec (12.624 before)
Query 8 - 114 rows in 9.963 sec (12.950 before)

```

הרשאות

כדי שנוכל לעבוד ככיתה שלמה המפתחת בסיס נתונים כאשר כל קבוצה בונה טבלאות שונות לבסיס הנתונים ואנו צריכים לגשת לטבלאות שלהם כדי לקחת מידע אנו צריכים לקבל מיוצרי הטבלאות הרשאה כדי שנוכל להשתמש בטבלאות שלהם.

אנחנו התבקשנו לתת הרשאות לאבי פדר ודניאל יוחנן עבור כל הטבלאות שלנו. בחלטנו לתת להם אך ורק הרשאת select, כיוון שאין סיבה שישנו את הטבלה או יימחקו חלקים ממנה. לכן, כתבנו שאילתה קצרה על מנת לתת הרשאה למשתמש של אבי פדר – afeder:

```
grant select on Notification to afeder;
grant select on FriendshipRequest to afeder;
grant select on FriendshipSuggestion to afeder;
```

לעומת זאת אנחנו קיבלנו הרשאות לקריאה (select) מהטבלאות הבאות:

Profile	- פרופיל
SocialGroup	- קבוצה חברתית
BuisnessPage	- עמוד עסקי
Event	- אירוע
Visibility	- נראות
BuisnessType	- סוג עסק

Views

VIEWS הם טבלאות וירטואליות. VIEWS מכילים הגדרות של עמודות וסוגי מידע שאותן עמודות יכולות להכיל. ההבדל בין הטבלאות לבין ה-VIEWS הוא שבטבלאות נשמרים נתונים באופן פיזי ואילו ב-VIEWS הנתונים לא נשמרים באופן פיזי בתוכם אלא הם רק מציגים נתונים הנשמרים בטבלאות. לכן לא ניתן לעדכן או להוסיף נתונים ל-VIEWS כפי שעושים לטבלאות.

ארבעת התצוגות הבאות שנציג מאפשרות למשתמשים שונים המנהלים את התפתחות הרשת החברתית ותפקודה לקבל אינדיקציה על מגוון רחב של אלמנטים שונים.

Data & AI – פיתוח

(1) תצוגה זו מאפשרת למפתחים לצפות בשינוי פלטי האלגוריתם המקשר בין שני משתמשים שאינם מכירים על בסיסי מאפיינים ומכנים משותפים. לאור זאת שהשינויים תדירים בזמן הפיתוח, יכולת זו חיונית מאוד לעיבוד הנתונים ושיפורו של ה-suggestions המסופקים למשתמשים ע"י המערכת המתבססת על האלגוריתם לאיתור צמדים מתאימים.

הקוד:

```
-- View 1 ---- best suggestions -----
create view best_suggestions as
select FS.SUGGESTIONID, FS.RECIEVERID,
       P1.FIRST_NAME as RECIEVER_FNAME,
       P1.LAST_NAME as RECIEVER_LNAME,
       FS.OFFERID,
       P2.FIRST_NAME as OFFER_FNAME,
       P2.LAST_NAME as OFFER_LNAME
from friendshipssuggestion FS
join
  afeder.profile P1
on FS.RECIEVERID = P1.PROFILE_ID
join afeder.profile P2
on FS.OFFERID = P2.PROFILE_ID
where exists (select *
              from friendshiprequest FR
              where FS.RECIEVERID = FR.REQUESTER_ID
                  and FS.OFFERID = FR.RECIEVER_ID
                  and FR.Approved = 1);
```

הפלט:

	SUGGESTIONID	RECIEVERID	RECIEVER_FNAME	RECIEVER_LNAME	OFFERID	OFFER_FNAME	OFFER_LNAME
1	20000	6961	Rodney	Carlisle	6923	Lorraine	Cale
2	20624	5080	Jonny	O'Neill	6955	Kitty	Arjona
3	20580	8783	Burt	Brown	5456	Lloyd	Wiest
4	20046	5700	Denise	Sainte-Marie	5478	Vin	Levert
5	20866	765	Debbie	Carrington	5516	Chi	Singh
6	20369	778	Deborah	Woods	5595	Anna	Imperioli
7	20559	1075	Dean	Saxon	5612	Milla	Satriani
8	20708	2336	Louise	Chandler	5619	Javon	Sledge

3:39 0:06 ydemri@labdbwin 240 rows selected in 0.206 seconds

(2) על מנת לאבחן את החשיפה של כלל המשתמשים להתראות המסתובבות ברשת ולאפשר התאמה אישית לתדירות השליחה, המפתחים מעוניינים, בשלבי אימון מודל ה-AI, לנתח את הקשרים שבין כמות התראות שאדם מקבל לאלו שהוא אכן השקיע זמן לצפות בהם.

הקוד:

```
-- View 2 ---- Notification Statistics -----
create view notification_stats as
select profile_id, first_name, last_name,
       count(notification_id) as total_notf,
       sum(watched) as total_watched,
from afeder.profile P
     natural left outer join
     notification N
group by profile_id, first_name, last_name;
```

הפלט:

		PROFILE_ID	FIRST_NAME	LAST_NAME	TOTAL_NOTF	TOTAL_WATCHED
▶	1	9953	Bebe	Hubbard	2	1
	2	7318	Gwyneth	Hawke	2	1
	3	3935	Jackie	Ponty	5	2
	4	1735	Ruth	Askew	3	3
	5	1491	Tim	Hewett	4	1
	6	700	Ceili	Hanks	2	1
	7	8002	Mac	Colman	2	1
	8	5969	Vin	MacDowell	1	1

3:39 0:06 ydemri@labdbwin 10000 rows selected in 6.137 seconds

מחלקת שיווק

(1) נתח שוק גדול שפונה לרשת החברתית הן חברות הפרסום. לאור הבנה זו, מעוניינים למצוא את האזורים "החמים" ביותר בהם כדאי להכניס תכני פרסומות ומודעות. אזורים אלו בדר"כ מאופיינים במפגשים ואירועים המנוהלים בידי ישויות פופולריות ובהן מרוכזות מסה של משתמשים. להלן האזורים הללו ברשת החברתית.

קוד:

```
-- View 3 ---- all popular profiles & their events ----
create view popular_pf_events as
select profile_id, first_name, last_name,
       event_id, event_name,
       e.location,
       e.event_date,
       e.participants_num
from nshushan.event E
natural right outer join
(select FR.RECIEVER_ID as profile_id,
 first_name, last_name
 from friendshiprequest FR
 join afeder.profile P
 on FR.RECIEVER_ID = P.PROFILE_ID
 group by FR.RECIEVER_ID, first_name, last_name
 having count(*) > 5) T;
```

הפלט:

	PROFILE_ID	FIRST_NAME	LAST_NAME	EVENT_ID	EVENT_NAME	LOCATION	EVENT_DATE	PARTICIPANTS_NUM
1	4196	William	Lynskey	56	Jazz workshop	62nd Street	07/11/2015	24
2	9560	Ozzy	Kimball	130	Jazz workshop	401 Andrews Road	16/08/2006	342
3	9868	Isaiah	Arkenstone	217	Memorial Day ceremony	79 Hauer	19/12/2009	240
4	7218	Eliza	Neil	253	party	1 Stevie Drive	12/06/2014	29
5	8568	Clint	O'Neal	349	Watching the World Cup togethe	55 Paul Road	25/11/2006	29
6	5615	Kristin	Curry	364	music show	56 Loreena Road	08/08/2010	17
7	9957	Davey	Bell	367	Watching the World Cup togethe	18 Mary Street	16/01/2010	168
8	7657	Marc	Connick	431	Jazz workshop	19 Arlington Street	28/01/2013	236

3:39 0:06 ydemri@labdbwin 213 rows selected in 6.283 seconds

(2) לאור ההצלחה בפרסום, עלייה ניכרת בחשיפה של משתמשים לתכני פרסומות ורווחים, החליטה ההנהלה על המשך פריסת הפרסומות גם בקבוצות משתמשים גדולות המאופיינים בישות מנהלת פעילה. להלן אזורים אלו ברשת.

הקוד:

```
-- View 4 ---- all active profiles & their big social groups ----
create view active_pf_groups as
select profile_id, first_name, last_name,
       sg.group_id, sg.group_name, sg.members_num, sg.about, v.visibility_name
from nshushan.socialgroup sg
natural join
nshushan.visibility v
natural right outer join
(select profile_id as Profile_Id,
       first_name as First_Name,
       last_name as Last_Name
from notification N
natural join
afeder.profile P
group by profile_id, first_name, last_name
having count(*) > 5) T
where members_num > 500000;
```

הפלט:

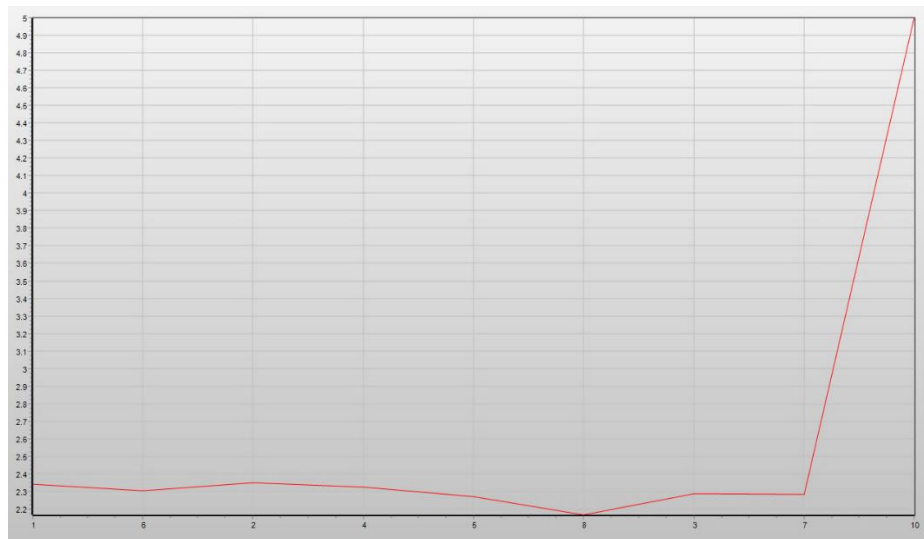
	PROFILE_ID	FIRST_NAME	LAST_NAME	GROUP_ID	GROUP_NAME	MEMBERS_NUM	ABOUT	VISABILITY_NAME
1	5567	Roscoe	Aniston	49	germanium	1642778	Combo Intel P4 1.8Ghz retail CPU + Gigabyte GA-8ID533 M/B + Fan.Ge	friends
2	3454	Gabrielle	Bancroft	1720	carbon	573480	Combo Intel P4 2.4ghz CPU + Fan(BOx) FSB800 with Hyper Threading w/	private
3	5144	Toshiro	Torn	1168	tungsten	666664	SIS Xabre200 64MB DDR AGP 8X DVI+TV+CRT by Power ColorW	public
4	1957	Ronny	Klein	2535	radon	1986436	128Mb SDRam PC133 KingstonRn	public
5	9089	Greg	Beckham	3490	helium	1579515	CNet External 56k V.92 Fax modem (Serial).He	friends
6	7957	France	Place	3440	lutetium	1782981	Printer CableLu	public
7	4436	Gloria	Prinze	3859	silver	545192	Combo Intel P4 1.6Ghz (512k cache) 845WN M/B & Fan (478pin)Ag	public
8	1045	Tanya	Hawke	4341	phosphorus	1021791	Intel Celeron 1.3GHz 256k S370 FCPGA2 (Retail Box)P	friends

תרשימים

לאחר שאספנו די מידע אודות החלק שלנו, נגשנו לראות כיצד מתנהגת הרשת החברתית שלנו מבחינת הדינמיקה שבין הנתונים ומסקנות מעניינות אחרות. לכן, הבאנו כאן שני תרשימים הממחישים נקודות מעין אלו.

תרשים א' – כמות בקשות החברות כפונקציה של מספר ההתראות

כידוע, משתמשים שמתמצאים ברשת החברתית ומבליים בה רבות בדר"כ מגיעים למפגשים חברתיים באופן פיזי/ דרך המחשב. השערה זו רצינו לבסס באמצעות המידע שנתון לנו בבסיס הנתונים העומד לרשותנו.



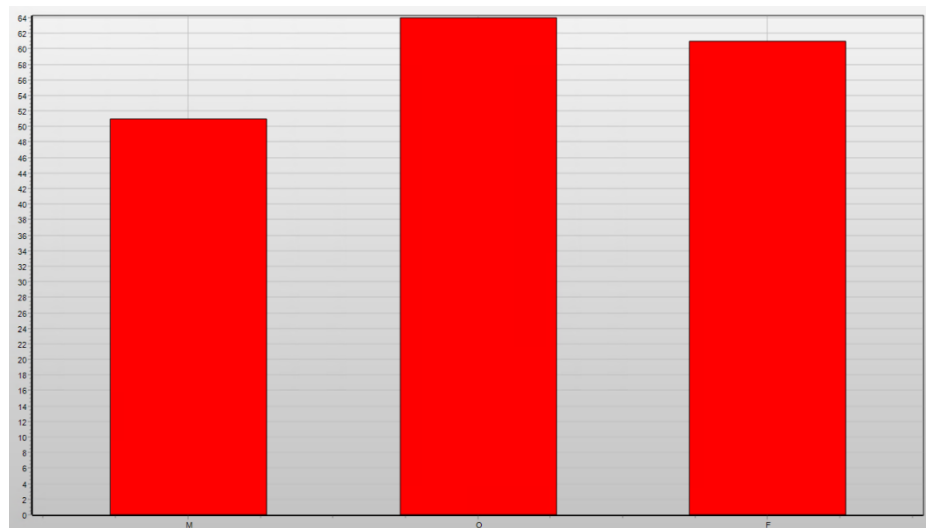
מסקנה: כפי שאפשר לראות, משתמשים בעלי תדירות התראות נמוך מאופיינים בבקשות חברות מעוטות. הערך איננו משתנה עבור מס' התראות נמוך מ-10 כפי שאפשר לראות אבל כאשר מתקרבים ל-10, אכן משתנה הערך ומשתמשים אלו דווקא מאופיינים בכמות בקשות חברות מרובות.

הקוד:

```
-- Graph 1: number of friend requests as a function of number of notifications
select num_notf, avg(num_requ)
from (
  select count(distinct N.NOTIFICATION_ID) as num_notf,
         count(distinct FR.REQU_ID) as num_requ
  from notification N
       natural join
       afeder.profile P
       join
       ydemri.friendshiprequest FR
  on profile_id = FR.RECIEVER_ID
 group by profile_id)
 group by num_notf;
```

תרשים ב' – פופולריות כפונקציה של מגדר

לאור חופש הדעות והבמה הפתוחה לכל משתמש שמאפשרת הרשת החברתית, רצינו לבדוק האם קיים איזה שהוא קשר בין האופן שמגדיר עצמו בן-אדם לבין הפופולריות שלו בחברה. כפי שכבר הגדרנו, **פופולריות** של משתמש ניתנת להגדרה כסה"כ בקשות החברות שמקבל דרך קבע. להלן התוצאה:



מסקנה: כפי שאפשר לראות, אנשים הנוטים לאפיין עצמם באופן שונה [Other] מהדרך הרגילה, נוטים להיות פופולריים יותר ממגדרים אחרים. אחריהם מגיעות הנשים [Female], באחוז לא גדול בכלל ולבסוף – גברים [Male].

הקוד:

```
-- Graph 2: popularity based on gender
select gender, count(*)
from (
  select FR.RECIEVER_ID as profile_id, first_name, last_name, gender
  from friendshiprequest FR join afeder.profile P on FR.RECIEVER_ID = P.PROFILE_ID
  group by FR.RECIEVER_ID, first_name, last_name, P.gender
  having count(*) > 5) T
group by gender
```

פונקציות ופרוצדורות

פונקציה בשפת SQL היא צורה מיוחדת של פקודה אשר מאפשרת ביצוע שאילתה מורכבת בצורה כזו שמשלבת פעולות שונות על המידע בבסיס הנתונים כדוג' הוספה, עדכון, מחיקה ושליפה. ההבדל בין פרוצדורה לפונקציה היא העובדה שפרוצדורה שאינה מחזירה ערך בגמר הפעולה לעומת פונקציה שכן מחזירה ערך כלשהו.

תוספות אלו למערך ניהול המידע מאפשר לנו, כמפתחי ומתחזקי בסיס הנתונים לבצע מניפולציות על ערכים המוחזרים משאילות מה שמאפשר ניתוח מעמיק יותר ושיפור השירות.

Functions

1. מפתחי ה-AI מעוניינים לקבל נתוני חשיפה להתראות על מנת לתחזק את האלגוריתם להצגת התראות. להלן פונקציה המחזירה את היחס הממוצע בין מספר ההתראות שנקראו לסך ההתראות שהתקבלו.

הקוד:

```
create or replace function get_notf_watched_ratio_avg return float as
res number;
begin
select avg(NS.total_watched / NS.total_notf)
into res
from ydemri.notification_stats NS;

return res;
end;
/
```

2. צוות ה-UI רוצה לדעת כיצד לתכנן את תצוגת החברים של המשתמש. לשם כך, עליהם לקבל את כמות החברים שיש למשתמש מסוים. פונקציה זו מקבלת מס' מזהה של משתמש ומחזירה את מס' חבריו.

הקוד:

```
create or replace function get_num_of_friends (profile_id in number) return number as
res number;
begin
select count(*)
into res
from friendshiprequest FR
where FR.RECIEVER_ID = profile_id
and FR.APPROVED = 1;

return res;
end;
/
```

Procedures

1. צוות האבטחה של הרשת מעוניין ביכולת להסיר משתמשים החשודים כבוטים (ע"פ ההגדרה שהצגנו בפרקים הקודמים). פרוצדורה זו עוברת על בסיס הנתונים ומסירה את כל המשתמשים ששולחים יותר מדי בקשות חברות.

הקוד:

```
create or replace procedure delete_all_bots as
begin
  delete from afeder.profile P
  where P.PROFILE_ID in (select requester_id as Requester_Id
                        from friendshiprequest FR join
                        afeder.profile P on requester_id = profile_id
                        group by requester_id, first_name, last_name
                        having count(*) > 5);
end;
```

2. מחלקת שירות משתמש הציעה להוסיף בתקופות מסוימות של השנה כותרות מיוחדות ומצחיקות להתראה ("שנה טובה", "שבת-שלום" וכו'). פרוצדורה זו עוברת על כלל ההתראות ומאפשרת החלפה של כלל ההתראות או שרשור של מחרוזת חדשה.

הקוד:

```
create or replace procedure update_notf_content (notf_id in number,
                                                new_content in varchar,
                                                app in number default 0) as
begin
  if app = 0 then
    update notification N
    set N.Content = new_content
    where N.Notification_Id = notf_id;
  else
    update notification N
    set N.Content = N.Content || new_content
    where N.Notification_Id = notf_id;
  end if;
end;
```

סיכום

לאחר מימוש חלקנו בפרויקט במהלך הסמסטר, למדנו להכיר את תהליך האפיון של הטבלאות ואת חשיבותו כבסיס לפיתוח איכותי בהמשך. הבנו כיצד מתבצעת חלוקת העבודה בין משתתפים שונים וכיצד מתאפשרת אינטגרציה בין חלקי הפיתוח השונים לכדי מכלול שלם.

אנחנו מרגישים שהקורס הנ"ל תרם רבות לידע האישי שלנו בתכנון והקמת בסיס נתונים כמו גם תחזוקו לאורך זמן ומרגישים כעת בטחון רב יותר בעיסוק עם מאגרי נתונים.

כמובן, על גבי שלל היכולות שבנינו במהלך הסמסטר, ניתן כעת להוסיף ולשפר את הרשת החברתית כך שתכלול יותר אינפורמציה, לרתום את כלל שכבת הנתונים למנוע לוגי המבצע מניפולציות שונות ומיוחדות וכמו כן לממשק משתמש נוח להצגת הנתונים.

את הקבצים השונים המכילים את קוד הפרויקט כמו גם את הדו"ח הזה ניתן למצוא באתר ה-GitHub כקוד פתוח בכתובת:

<https://github.com/yakir0/minipDB>