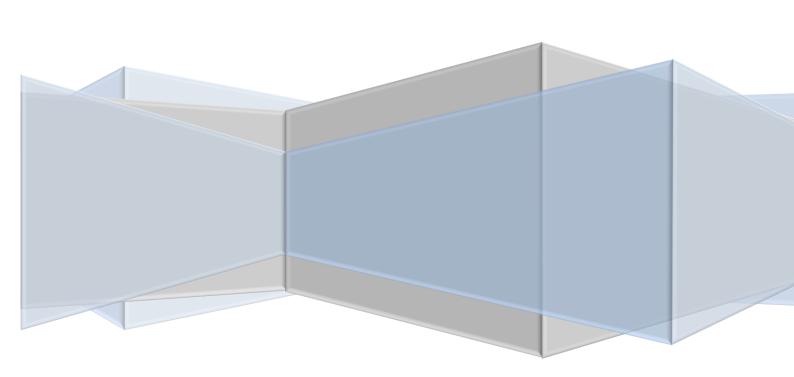
מיני פרויקט בבסיסי נתונים

שגב בורשטיין & יקיר דמרי 20.6.2021



<u>עוכן</u>

2	מבוא
ת התוכנה	עבודת הכנה והכר
3	ERD תרשים
קשרים	תיאור הישויות והי
3	ישויות
3	קשרים
Зл	נרמול הטבלאוו
3	פרוקים
4	תרשים DSD
5	הפרויקט שלנו
5	ERD תרשים
6	תיאור הישויות וה <i>י</i>
6	ישויות
6	קשרים
7	נרמול הטבלאות
7	תרשים DSD
7	יצירת הטבלאות
8	אכלוס טבלאות
9	שאילתות SQL שאילתות
13	אינדקסים
15	הרשאות
16	VIEWS
20	תרשימים
22	פונקציות ופרוצדו
24	סיכום

מבוא

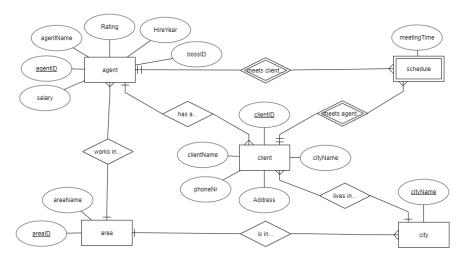
הרעיון שנבחר במהלך הדיון בכיתה הוא לממש בסיס נתונים המספק את הפונקציונליות על מנת לתמוך ברשת חברתית. הרשת שלנו מאפשרת למשתמש מגוון עולמות לנוחותו כגון פיתוח קשרים עם משתמשים אחרים ותקשורת עמהם, גישה לקבוצות ופגישות חברתיות המכילות תוכן משותף וכמובן היכולת לשתף את דעתו האישית של המשתמש עם שאר המשתמשים 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 ○
 - ס קבוצה
 - עמוד עסקי ○
- סוג העסק
 - התראה
- התפתחות חברתית
 - הצעת חברות ○
 - ∘ בקשת חברות
 - חבר
 - נראות
- פעולות משתמש אקטיבי
 - ס לייק
 - שיתוף \circ
 - ס תיוג

<u>עבודת הכנה והכרת התוכנה</u>

<u>תרשים ERD</u>



<u>תיאור הישויות והקשרים</u>

ישויות

- Client מאופיין ב- ת.ז של הלקוח, שם, כתובת, מספר פלאפון, עיר מגורים ומספר סוכן. Client •
- Agent מאופיין ב- ת.ז של הסוכן, שם סוכן, מספר אזור, דירוג, ותק, ת.ז של הבוס, משכורת.
 - . מאופיין ב- מספר אזור ובשם האזור Area
 - מאופיין ב- מספר אזור ובשם העיר. City •
 - Schedule ב- ת.ז של הסוכן, ת.ז של הלקוח ובזמן הפגישה. •

קשרים

- לכל סוכן- יש אזור עבודה יחידי, יכולים להיות הרבה לקוחות ויכולים להיות לו הרבה פגישות עם לקוחות.
 - לכל אזור- יכולות להיות כמה ערים ויכולים להיות הרבה סוכנים שעובדים באזור זה.
 - לכל עיר- יכולים להיות הרבה לקוחות שגרים בה והיא יכולה להיות באזור יחידי.
 - לכל לוח זמנים- חייב להיות סוכן יחיד ולקוח יחיד הנפגשים בזמן מסוים.
 - לכל לקוח-יש סוכן יחיד והוא גר בעיר יחידה ויכול להיות לו הרבה פגישות עם הסוכן שלו.

נרמול הטבלאות

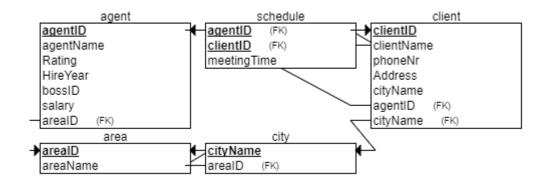
- Schedule (agentID, clientID, meetingTime)
- Agent (agentID, rating, hireYear, boosID, salary, areaID)
- Client (<u>clientID</u>, clientName, address, phoneNr, cityName)
- CityName (cityName, areaID)
- Area (areaID, areaName)

פרוקים

היחסים עומדים ב- 3NF וב- BCNF מכיוון שבכל טבלה, התלויות הפונקציונאליות הלא-טריוויאליות : BCNF היחסים עומדים ב- X תולכן מתקיים שלכל $X \to X$ הוא מפתח ולכן הם עומד בתנאים.

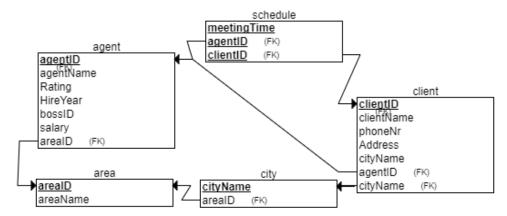
תרשים DSD

כאן ניתן לראות את ה- DSD שהפקנו מתרשים ה- ERD שיצרנו.



אך זה לא כמו מה שמומש ב- create table שניתן לנו כי שם ב- schedule אך זה לא כמו מה שמומש ב- praimery key כ- meetingTime

לכן נוסיף זאת לתרשים שלנו באופן ידני, ונקבל את התרשים הבא.



הפרויקט שלנו

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

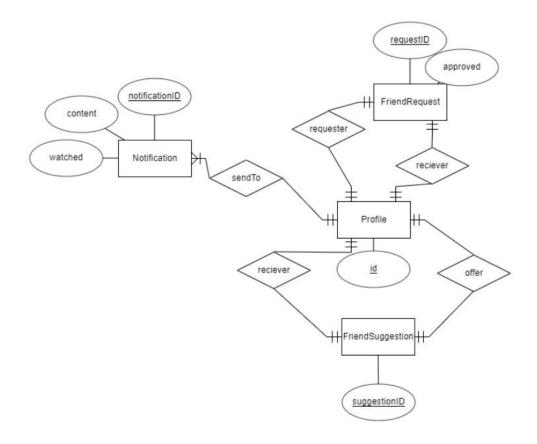
כחלק מתפקידנו במערכת הכוללת, לקחנו על עצמנו את יישומם של הטבלאות הבאות:

- **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).



<u>תיאור הישויות והקשרים</u>

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).
- o profile_id − מזהה הפרופיל שקיבל את ההתראה (FK).
 - התראה. − content o
 - האם ההודעה נצפתה. − watched o
- ישות זאת אחראית על כל הצעות החברות שהמערכת מסוגלת FriendshipSuggestion ישות זאת אחראית על כל הצעות החברות שהמשתמש ביצע. להציע למשתמש בהתבסס על נתונים ופעולות קודמות שהמשתמש ביצע. ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.
 - (PK) מספר מזהה של ההצעה − suggestion_id o
 - (FK) מספר מזהה של הפרופיל profile1_id \circ
 - .(FK) מספר מזהה של הפרופיל profile2_id \circ
- ישות זאת אחראית על כל בקשות החברות שהמשתמש מפנה Friendship-Request ישות זאת אחראית על מנת להגדיל את הסביבה האישית שלו.
 ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.
 - פר מזהה של הבקשה (PK). − request_id o
 - __ profile1_id __ מספר מזהה של המשתמש **המקבל (FK)**.
 - (FK) מספר מזהה של המשתמש profile2 id \circ
 - האם הבקשה אושרה. [כן/לא/מחכה לתשובה]. isApproved \circ

קשרים

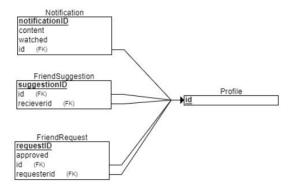
- הקשר בין Notification לבין Profile. הקשר הינו חזק, כיוון שמקשר בין שתי אפרול היות חזקות. הקשר הוא 1:1 משום שיכול להיות הרבה התראות למשתמש בודד אבל לא יכול להיות הרבה משתמשים להתראה בודדת.
- הקשר בין profile לבין friendshipSuggestion. הקשר הינו חזק, כיוון שמקשר profile הקשר בין שמיות חזקות. הקשר הוא 1:1 משום שלכל הצעה יכול להיות רק אדם אחד בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל הצעה יכול להיות רק אדם אחד שמקבל אותה (אלו הצעות מותאמות אישית פר פרופיל).
- profile הקשר בין profile לבין friendshipRequest. הקשר הינו חזק, כיוון שמקשר Receiver בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל בקשה יכול להיות רק אדם אחד שמקבל אותה.
- Profile לבין profile הקשר הינו חזק, כיוון שמקשר הקשר הינו חזק, כיוון שמקשר Requester בין שתי ישויות חזקות. הקשר הוא 1:1 משום שלכל בקשה יכול להיות רק אדם אחד ששולח אותה.
- profile לבין friendshipSuggestion. הקשר הינו חזק, כיוון שמקשר profile 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- BCNF טריוויאליות הן מהמפתח אל תכונות נוספות לכן מתקיים שלכל $X \to X$ 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 create table.

יצרנו קובץ SQL יצרנו קוד לייצור הטבלאות באמצעות erdPlus של האתר export SQL יצרנו קובץ SQL יצרנו קוד לייצור הטבלאות באמצעות 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
);

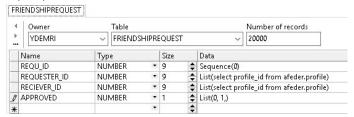
create table FriendshipRequest (
    requ_id number(9) primary key,
    requester_id number(9) foreign key references profile(profile_id),
    reciever_id number(9) foreign key references profile(profile_id),
    approved bit(1)
);

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

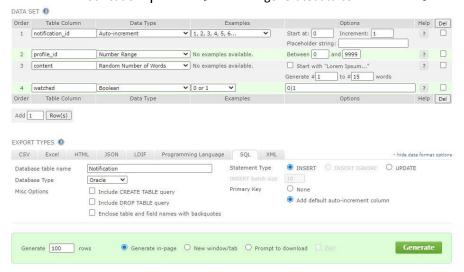
לאחר יצירת הטבלאות, אנו נדרשים למלא אותן במידע. על מנת לעשות זאת, השתמשנו במגוון על מנת לאכלס את הטבלאות. להלן הכלים והתוצאות:

:FriendshipRequest מתוכנת ה-PL/SQL מתוכנת ה-Data Generator -



ניתן לראות כי בעמודה Approved הוא (List(0, 1,) הוא (ביתן לראות כי בעמודה Approved האחרון מסמל null שגם null

באמצעות האתר generatedata.com¹ להכנסת ערכים לתוך



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 בפייתון המייצר קובץ עם שאילתות Insert. להלן הסקריפט:

```
import random
insert_query = 'INSERT INTO FriendshipSuggestion (suggestionid, recieverid, offerid) VALUES (%d, %d);\n'
def createInsertQuery(rows=20_000):
    with open("insertQuery.sq1","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 מGenerateData נמצאים בגיט.

https://www.generatedata.com/ 1

שאילתות SQL

שאילתה 1 - כל הבקשות שמחכות לתשובה:

<u>התרחיש:</u> כחלק מהרצון של ההנהלה לדעת את מצב זמינות המשתמשים, ופנאי השימוש במערכת, ההנהלה ביקשה לדעת את כמות ההתראות שמחכות למענה בקרב המשתמשים.

לכן כתבנו שאילתה המחזירה טבלה של כל הבקשות שמחכות למענה.

<u>הגדרה:</u> 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 ב6.427 שניות

	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

שאילתה 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 שאילתה המבצעת את הבקשה הנ"ל. <u>הגדרה:</u> 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 notf_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 ב6.535 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	Butler		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	200	6
12	6405	Sissy	Uggams	***	6
13	6075	Dan	Bacharach	200	6
14	3717	Mary	Holly		6
15	8470	Lamelle	Shearer		8
16	3570	Tori	von Sydow		7
17	2135	Dar	Keith		6

שאילתה 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 שאילתה המחזירה את הפרופילים המומלצים ביותר ע"י המערכת בקרב משתמשים שונים ברשת החברתית.

הגדרה: פרופילים שהוצעו יותר מ-5 פעמים.

הקוד:

```
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	***	- 1	7
12	4276	Rowan	***	Johansson	***		6
13	3183	Derek	***	Kutcher	***		6
14	2229	Christmas		Gaines		1	7
15	1348	Debi	***	Richards			7
16	6389	Jeanne	***	Tucker			8
17	7028	Mae		Branagh	101		7

שאילתה 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 שאילתה המחזירה טבלה של כל המשתמשים החשודים. מצב שכזה יועבר לניתוח מעמיק יותר ותפיסת המפעילים.

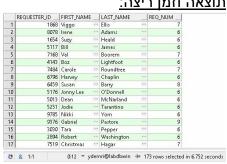
הגדרה: משתמשים אשר שלחו יותר מ-5 בקשות.

<u>הקוד:</u>

```
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
maving count(*) > 5;
```

תו<u>צאה וזמן ריצה:</u>

התוצאה: 173 שורות ב6.752 שניות



שאילתה 5 - הצעות החברות המוצלחות:

<u>התרחיש:</u> 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 שאילתה המחזירה את כל הצעות החברות המוצלחות שנעשו ע"י המערכת.

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

```
<u>הקוד:</u>
```

```
-- 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_ENAME,
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 friendshipsequest 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		Arjona	
3	20580	8783	Burt	***	Brown	225	5456	Lloyd		Wiest	
4	20046	5700	Denise		Sainte-Marie		5478	Vin		Levert	
5	20866	765	Debbie	***	Carrington	998	5516	Chi		Singh	
6	20369	778	Deborah		Woods		5595	Anna		Imperioli	
7	20559	1075	Dean	***	Saxon	888	5612	Milla		Satriani	
8	20708	2336	Louise	***	Chandler	0.8.0	5619	Javon		Sledge	
9	20804	5092	Etta	***	Prinze	199	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	114	Guzman	140	5934	Ronnie		Stewart	
16	20115	7401	Helen	***	Manning		5935	Kirk		Duschel	
17	20089	8463	Lorraine	444	Lynne	***	5952	Avril		Sirtis	

שאילתה 6 - כל ההתראות על המלצות/בקשות:

<u>התרחיש:</u> 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 שאילתה המחזירה את כל ההתראות הקשורות להמלצות חברות או בקשות חברות.

:הקוד

תוצאה וזמן ריצה:

התוצאה: 9807 שורות ב12.549 שניות

	NOTIFICATION_ID	PROFILE_ID	CONTENT		WATCHED
1	8575	1	Anne officiis omnis deleniti quos est in occaecat.	***	
2	13739	2	Tony quod.		
3	458	2	Devon maiores voluptate excepteur et dolor et.	100	
4	18712	3	Anita cillum dolor rerum corrupti sint.	***	
5	4513	5	Mili recusandae id et esse et maxime doloribus.	***	
6	14112	6	Joan itaque at id quidem quod est ex repellendus tempore.		
7	12322	6	Solomon voluptatum mollit repellat ut similique saepe.	***	
8	10865	7	Cate est esse.		
9	1935	7	Denis cupiditate lorem officiis voluptate id non.	***	
10	15202	8	Teena laboris.	***	
11	13262	8	Christopher	***	
12	10552	9	Coley optio quos sint autem accusamus.	***	
13	7710	9	Rawlins qui hic animi id similique.	***	
14	17437	11	Rhys temporibus dolorum hic dolor tempore.	***	
15	18511	14	Julio fugiat omnis deserunt amet eligendi.	***	
16	16015	15	Morris et nobis fugiat occaecati assumenda.		
17	12685	15	Ron duis nobis dolorum amet laborum culpa autem in cupidatat.	***	

שאילתה 7 - כל הפרופילים הפופולריים:

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

לכן כתבנו שאילתה המחזירה את כל אותם משתמשים מפורסמים.

<u>הגדרה:</u> 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 ב12.624 שניות



שאילתה 8 - פעילות חשודה:

<u>התרחיש:</u> 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 שאילתה המחפשת את כל המשתמשים הבעייתיים הללו ומחזירה את פרטיהם.

<u>הגדרה:</u> פרופיל ששלח יותר מ-3 הצעות חברות לנשים שונות.

הקוד:

תוצאה וזמן ריצה:

התוצאה: 114 שורות ב12.950 שניות



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 הגרום לזה שמנוע החיפוש בבסיס הנתונים יפנה קודם לאינדקס וילך לרשומות המתאימות על פי מה שרשום באינדקס.

האינדקסים שכתבנו הם:

```
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 על ידם. לכן, יצירת אינדקסים תאפשר למערכת לקצר את זמן החיפוש של שורות בבסיס המידע. כעת נציג שיפור של שלושה שאילתות מדגמיות מתוך שמונת השאילתות שציינו בחלק הקודם (עמוד 10).

1. שאילתה 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	

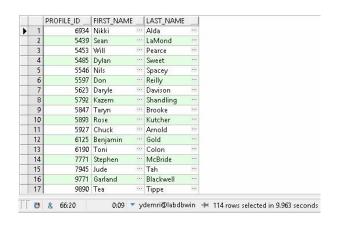
- לפני: (sec) .6.427.
- .3.832 (sec) אחרי: -
- מסקנה: מדובר ב-(sec) 40.377%, שיפור. -

2. שאילתה מס' (6):

		NOTIFICATION_ID	PROFILE_ID	CONTENT		WATCHED _
	1	8575	1	Anne officiis omnis deleniti quos est in occaecat.		
	2	13739	2	Tony quod.	***	
	3	458	2	Devon maiores voluptate excepteur et dolor et.		
	4	18712	3	Anita cillum dolor rerum corrupti sint.		
	5	4513	5	Mili recusandae id et esse et maxime doloribus.		
	6	14112	6	Joan itaque at id quidem quod est ex repellendus tempore.		
	7	12322	6	Solomon voluptatum mollit repellat ut similique saepe.		
	8	10865	7	Cate est esse.		
	9	1935	7	Denis cupiditate lorem officiis voluptate id non.	***	
1	0	15202	8	Teena laboris.	***	
- 1	1	13262	8	Christopher		
- 1	2	10552	9	Coley optio quos sint autem accusamus.		
1	3	7710	9	Rawlins qui hic animi id similique.		
1	4	17437	11	Rhys temporibus dolorum hic dolor tempore.		
1	5	18511	14	Julio fugiat omnis deserunt amet eligendi.		
1	6	16015	15	Morris et nobis fugiat occaecati assumenda.		
1	7	12685	15	Ron duis nobis dolorum amet laborum culpa autem in cupidatat.	***	

- .12.549 (sec) לפני:
- .9.832 (sec) אחרי: -
- <u>מסקנה:</u> מדובר ב-(sec), 2.717 שיפור.

3. שאילתה מס' (8):



- .12.950 (sec) <u>לפני:</u>
- .9.963 (sec) אחרי: -
- מסקנה: מדובר ב-(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.

במקרה שלנו לא היינו צריכים לתת הרשאות לאף אחד מכיוון שהטבלאות שלנו מייצגות מאפיינים ייחודים המתארים חלק ספציפי במודל הרשת החברתית. מהסיבה הזו אנחנו היינו תלויים באחרים.

קיבלנו הרשאות לקריאה 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 VIEWS הם טבלאות וירטואליות. VIEWS הוא שבטבלאות נשמרים נתונים עמודות יכולות להכיל. ההבדל בין הטבלאות לבין ה-S 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 לנובלאות

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

Data & AI – מחלקת 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 suggestions – של ה-suggestions המסופקים למשתמשים ע"י המערכת המתבססת על האלגוריתם לאיתור צמדים מתאימים.

<u>הקוד:</u>

```
View 1 ---- best suggestions
create view best suggestions as
      P1.FIRST_NAME as RECIEVER_FNAME,
      P1.LAST_NAME as RECIEVER_LNAME,
      P2.FIRST_NAME as OFFER_FNAME,
      P2.LAST_NAME as OFFER_LNAME
from friendshipsuggestion FS
    afeder.profile P1
    on FS.RECIEVERID = P1.PR0FILE_ID
     join afeder.profile P2
    on FS.OFFERID = P2.PR0FILE_ID
where exists (select *
             from friendshiprequest FR
             where FS.RECIEVERID = FR.REQUESTER_ID
                    and FS.OFFERID = FR.RECIEVER_ID
                    and FR.Approved = 1);
```

		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
1	7	20559	1075	Dean	Saxon		5612	Milla	Satriani
T	8	20708	2336	Louise	Chandler	123	5619	Javon	Sledge

(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.

<u>הקוד:</u>

		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
T	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

מחלקת שיווק 👃

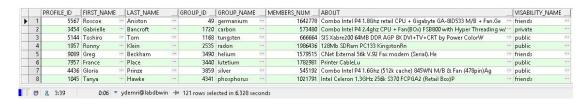
(1) נתח שוק גדול שפונה לרשת החברתית הן חברות הפרסום. לאור הבנה זו, מעוניינים למצוא את האזורים "החמים" ביותר בהם כדאי להכניס תכני פרסומות ומודעות. אזורים אלו בדר"כ מאופיינים במפגשים ואירועים המנוהלים בידי ישויות פופולרית ובהן מרוכזות מסה של משתמשים. להלן האזורים הללו ברשת החברתית.

<u>קוד:</u>

		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	٠	24
	2	9560	Ozzy		Kimball		130	Jazz workshop		401 Andrews Road		16/08/2006	•	342
1	3	9868	Isaiah		Arkenstone		217	Memorial Day ceremony		79 Hauer		19/12/2009	•	240
	4	7218	Eliza		Neil	1000	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	•	1
	7	9957	Davey	***	Bell		367	Watching the World Cup togethe		18 Mary Street	***	16/01/2010	•	168
1 3	8	7657	Marc		Connick		431	Jazz workshop		19 Arlington Street		28/01/2013		236

(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.

<u>הקוד:</u>

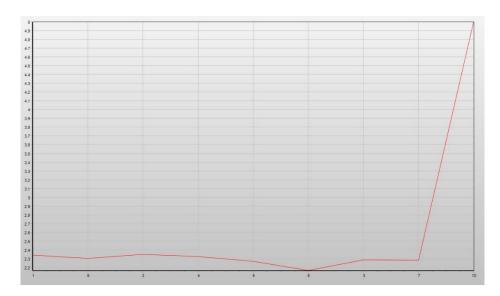


תרשימים

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

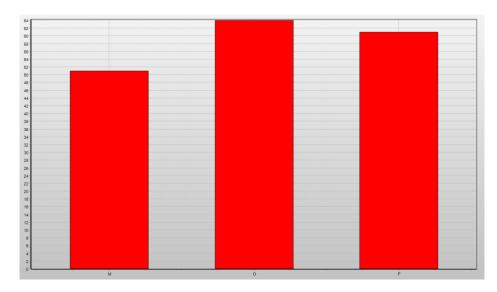
כידוע, משתמשים שמתמצאים ברשת החברתית ומבלים בה רבות בדר"כ מגיעים למפגשים חברתיים באופן פיזי/ דרך המחשב. השערה זו רצינו לבסס באמצעות המידע שנתון לנו בבסיס הנתונים העומד לרשותנו.



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.

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:



מהדרך (Other] מהדרך 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].

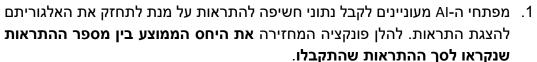
```
-- Graph 2: popularity based on gender
select gender, count(*)
from (
    select FR.RECIEVER_ID as profile_id, first_name, last_name, gender
    from friendshiprequest FR join afeder.profile P on FR.RECIEVER_ID = P.PROFILE_ID
    group by FR.RECIEVER_ID, first_name, last_name, P.gender
    having 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 משאילתות מה שמאפשר ניתוח מעמיק יותר ושיפור השירות.

:Functions 4



:הקוד

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**.

```
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 🔸

 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:

 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.

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