פתרון תרגיל כתה בנושא

יצירת טבלאות באמצעות פקודות DDL– סביבת MS-SQL

מערכת match me

אילוצים שלא ניתן לאכוף ברמת סכמת הטבלאות:

אילוצי חובה (משתמעים במפורש מהסיפור)

1. באפליקציה מסוימת לא יכול להיות שני משתמשים עם כינוי זהה. לכן, בעת רישום משתמש לאפליקציה מסוימת, המערכת תבדוק כי לא קיים משתמש אחר בעל אותו כינוי באפליקציה זאת. במידה וכבר קיים משתמש בעל הכינוי המבוקש ההכנסה לא תאופשר (ניתן לפתור בעזרת אילוצי ייחודיות בעת יצירת הטבלה או בעזרת טריגר)

2. כאשר משתמש מסויים מסמן כי איננו מעוניין במותאם, יהיה עלינו להגדיר כי גם המותאם לא מעוניין במשתמש על מנת למנוע מצב בו הוא יוצג לו (ניתן לפתור בעזרת טריגר)

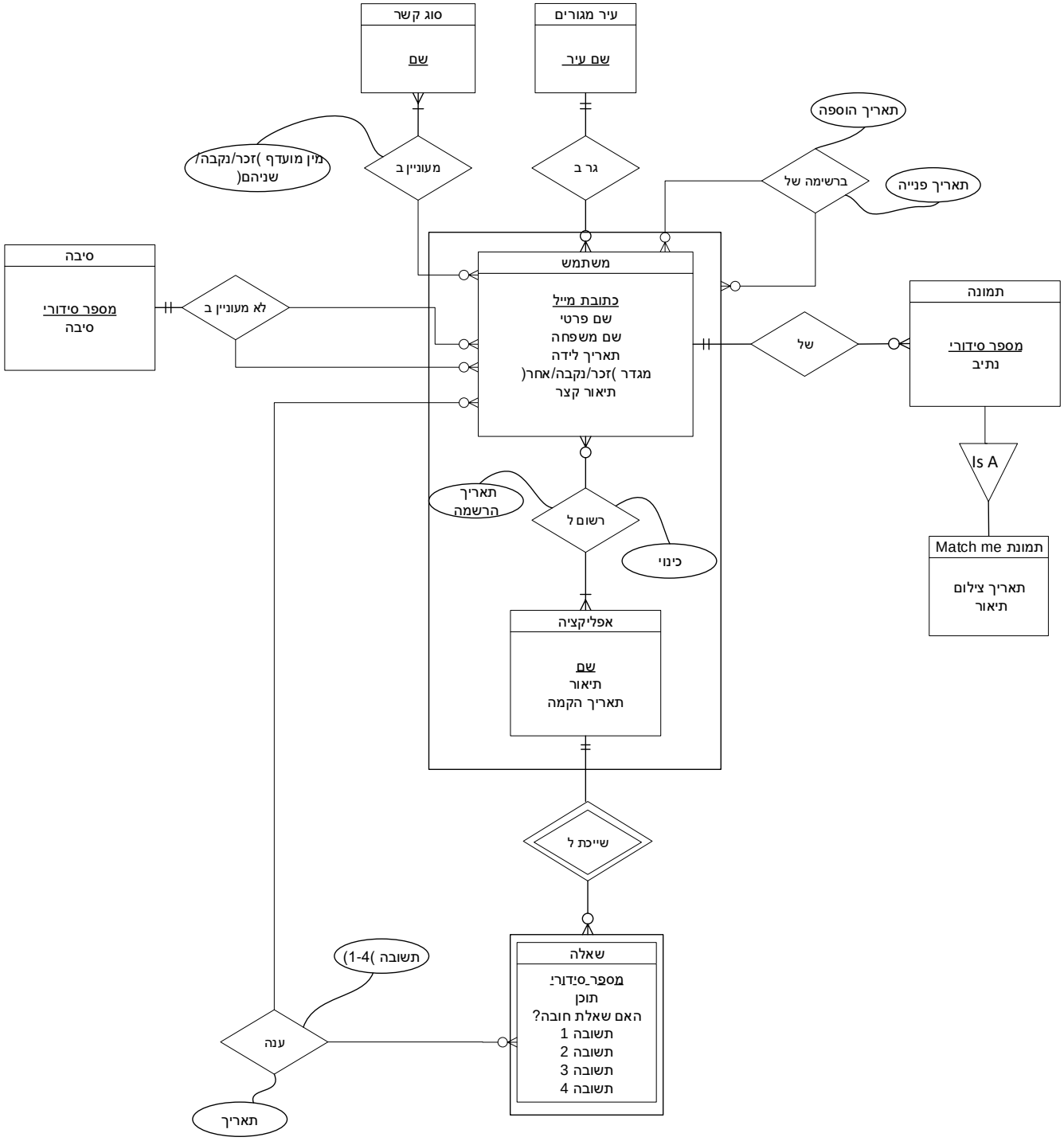
3. יש לדאוג למספור עוקב עבור שאלות באפליקציות בעת הכנסת השאלות (ניתן לפתור בעזרת טריגר או ברמת ממשק המשתמש)

4. יש לדאוג לא לאפשר להוסיף את אותו מותאם יותר מפעם אחת לאותו משתמש, כל פעם באפליקציה אחרת (ניתן לפתור בעזרת אילוצי ייחודיות בעת יצירת הטבלה או בעזרת טריגר)

אילוצי רשות לדוגמא

5. על מנת למנוע מרובוטים לענות על שאלות כדי לשבש את נתוני האפליקציה, יש לבדוק כי משתמש שעונה על שאלה אכך רשום לאפליקציה שאליה השאלה שייכת. אחרת המענה על השאלה לא יתאפשר (ניתן לפתור בעזרת טריגר)

1. תרשים ה- ERD:


סוג קשר
שם
עיר מגורים
שם עיר
תאריך הוספה
מין מועדף)זכר/נקבה/
שניהם(
מעוניין ב
גר ב
ברשימה של
תאריך פנייה
סיבה
מספר סידורי
סיבה
לא מעוניין ב
משתמש
כתובת מייל
שם פרטי
שם משפחה
תאריך לידה
מגדר)זכר/נקבה/אחר(
תיאור קצר
של
תמונה
מספר סידורי
נתיב
Is A
תמונת Match me
תאריך צילום
תיאור
תאריך
הרשמה
רשום ל
כינוי
אפליקציה
שם
תיאור
תאריך הקמה
שייכת ל
שאלה
מספר_סידורי
תוכן
האם שאלת חובה?
תשובה 1
תשובה 2
תשובה 3
תשובה 4
תשובה)1-4(
ענה
תאריך

טבלאות מנורמולות 3NF (לפי הכללים):

שם הטבלה	שדות הטבלה
עיר מגורים City	שם עיר cityName
סוג קשר RelationType	שם קשר relationName
משתמש Users	כתובת מייל, שם פרטי, שם משפחה, תאריך לידה, מגדר (זכר/נקבה/אחר), תיאור, שם עיר (מ.ז. לעיר) userEmail, firstName, lastName, dateOfBirth, gender(F, M, O), description, cityName (fk_City)
מעוניין ב (משתמש- סוג קשר) InterestedIn	כתובת מייל (מ.ז. משתמש), שם קשר (מ.ז. קשר), מין (זכר/ נקבה/ שניהם) userEmail(fk_User), relationName (fk_RelationType), gender(F, M, B)
דחייה Reject	סיבת דחייה rejectReason
לא מעוניין ב DoesntLike	כתובת מייל (מ.ז. משתמש), כתובת מייל דחוי (מ.ז. משתמש), סיבת דחייה (מ.ז. דחייה) userEmailHate (fk_User), userEmailHateed (fk_User), rejectReason (fk_Reason) * כתובת המייל בעמודה הראשונה תציין את הדוחה והכתובת המייל בעמודה השנייה את הנדחה
תמונה Picture	מספר סידורי, נתיב, מייל משתמש (מ.ז. משתמש) picSerialNum, path, userEmail(fk_User)
תמונת Match me MatchMePicture	מספר סידורי (מ.ז. תמונה), תאריך צילום, תיאור picSerialNum (fk_Picture), photoDate, description
אפליקציה App	שם אפליקציה, תיאור, תאריך הקמה appName, description, creationDate
רשום ל (משתמש – אפליקציה) RegisteredTo	כתובת מייל (מ.ז. משתמש), שם אפליקציה (מ.ז. אפליקציה), תאריך הרשמה, כינוי useEmail(fk_User), appName(fk_App), registrationDate, nickName
שאלה Question	שם אפליקציה (מ.ז. אפליקציה), מספר שאלה, תוכן, האם חובה? תשובה 1, תשובה 2, תשובה 3, תשובה 4 appName (fk_App), questionSerialNum, content, IsMandatory?, answer1, answer2, answer3, answer4
ענה על (משתמש – שאלה) tblAnsweredOm	כתובת מייל (מ.ז. משתמש),שם אפליקציה, מספר שאלה (יחד מ.ז. לשאלה), תשובה (1-4), תאריך מענה userEmail(fk_User), appName + questionSerialNum (fk_Question), answer (1-4), answerDate
ברשימה של (משתמש- רשום ל) List	כתובת מייל (מ.ז. משתמש), כתובת מייל של המותאם, שם אפליקציה (יחד מ.ז. רשום ל), תאריך הוספה, תאריך פנייה userEmailAddTo (fk_Users), userEmailAddedTo + appName (fk_RegisteredTo), addDate, talkDate

פקודות ה- SQL DDL היוצרות את טבלאות ב- SQL Server:

1) טבלת עיר מגורים

```
create table city
(
cityName nvarchar(20) primary key
)
```

2) טבלת משתמש

```
create table users
(
userEmail varchar(50) primary key check(userEmail like '_%@_%._%'),
firstName nvarchar(20) not null,
surName nvarchar(20) not null,
birthDate date check(datediff(year,birthDate,getdate()) >= 18) not
null,
gender char(1) check(gender in('M','F','O')) default 'O',
description nvarchar(255),
cityName nvarchar(20) references city(cityName)
)
```

3) טבלת סיבה

```
create table Reject
(
rejectName nvarchar(20) primary key
)
```

4) טבלת לא מעוניין ב

```
create table DoesntLike
(
userEmailHate varchar(50) references Users(userEmail),
userEmailHated varchar(50) references Users(userEmail),
rejectName nvarchar(20) references Reject(rejectName) not null
primary key(userEmailHate,userEmailHated),
check(userEmailHate != userEmailHated)
)
```

5) טבלת סוג קשר

```
create table relationType
(
relationName nvarchar(20) primary key
)
```

6) טבלת מעוניין ב

```
create table userRelation
(
userEmail varchar(50) references users(userEmail),
relationName nvarchar(20) references relationtype(relationName),
gender char(1) check(gender in ('M', 'F', 'B')) not null,
primary key(userEmail,relationName)
)
```

7) טבלת תמונה

```sql
create table Picture
(
picSerialNum int identity(3,10) primary key,
path nvarchar(255) not null,
userEmail varchar(50) references Users(userEmail) not null
)
```

8) טבלת תמונת match me

```sql
create table matchMePic
(
picSerialNum int references picture(picSerialNum) primary key,
picDate date not null,
description nvarchar(255)
)
```

9) טבלת אפליקציה

```sql
create table App
(
appName nvarchar(20) primary key,
description nvarchar(255) not null,
creationDate date check(creationDate <= getdate()) not null
)
```

10)טבלת רשום ל

```sql
create table RegisteredTo
(
userEmail varchar(50) references Users(userEmail),
appName nvarchar(20) references App(appName),
registrationDate date,
nickName nvarchar(20),
primary key(userEmail,appName),
unique(appName,nickName)
   )
```

11)טבלת שאלה

```sql
create table Question
(
appName nvarchar(20) references App(appName) on delete cascade,
questionSerialNum int,
content nvarchar(255) not null,
mandatory bit not null,
ans1 nvarchar(50) not null,
ans2 nvarchar(50) not null,
ans3 nvarchar(50) not null,
ans4 nvarchar(50) not null,
primary key(appName,questionSerialNum)
)
```

12)טבלת ענה על

```sql
create table Answers
(
userEmail varchar(50) references Users(userEmail),
appName nvarchar(20),
questionSerialNum int,
ansDate date not null,
ans char(1) check(ans in ('1','2','3','4')) not null,
foreign key(appName,questionSerialNum) references
Question(appName,questionSerialNum),
primary key(userEmail,appName,questionSerialNum)
)
```

13)טבלת ברשימה של

```sql
create table List
(
userEmailAddTo varchar(50) references Users(userEmail),
userEmailAddedTo varchar(50),
appName nvarchar(20),
addDate date not null,
talkDate date,
foreign key (userEmailAddedTo,appName) references
RegisteredTo(userEmail,appName),
primary key(userEmailAddTo,userEmailAddedTo,appName),
check(userEmailAddTo <> userEmailAddedTo) ,
check(addDate <= talkDate)
)
```