**SOFTWARE-DOCS**

**DB Scheme structure:**

בציור של הDB להלן שמנו את כל השדות שאינם מפתח הטבלה באותו עיגול על מנת לחסוך במקום

Places

Addr

Has an

User

Has

Pics

Details

OpenHours

Reviews

Has

Has

Has

Has

הסברים:

לכל אחת מהטבלאות יש id ייחודי משלה. לטבלאות User וPlaces יש כתובת יחידה, על כן שמנו בטבלאות אלו שדה נוסף בשם addr\_id. ניתן היה לחסוך בטבלה על ידי הכנסת הכתובת כחלק מהפרטים בכל שורה תחת הטבלאות Iser וPlaces, אך לכתובת יש פרטים רבים, והיא הייתה מנפחת את גודל הטבלאות האחרות. בנוסף, רצינו לתמוך בהרחבה של הDB מאוחר יותר לטבלאות נוספות שיעשו שימוש בכתובת, ולכן העדפנו לשמור על טבלה זו כטבלה נפרדת.

להרבה מהטבלאות יש שדה בשם googlePlaceId, שזה המזהה בו גוגל משתמש כמזהה חד חד ערכי לכתובת. בחרנו לשמור אותו בטבלאות רבות, ולא רק בטבלת places על מנת לתמוך בהכנסת פרטים על מקומות שאין לנו עדיין בDB, אך ייתכן שיוכנסו אליו בהמשך. למרות קשר הone-to-many שיש בין Places לטבלאות הפרטים, אנחנו מאפשרים לשמור פרטים שאינם מקושרים למקום קיים.

בנוסף, בהרבה מהטבלאות השתמשנו במזהה googlePlaceId בתור אינדקס, על מנת לאפשר שליפות מהירות יותר.

**DB optimizations:**

טבלת Reviews – הטבלה מאפשרת שליפות fulltext, כאשר אינדקסנו את שדה text המכיל את הטקסט של הביקורת. על מנת לאפשר שליפות fulltext, השתמשנו במנוע MyISAM.

טבלת Places – רוב השליפות בטבלה זו מתבצעות על פי עמודות type ו googlePlaceId, על כן דאגנו לאנדקס את שתי העמודות האלו.

טבלת User – אינדקסנו את עמודת שם המשתמש משום שהבדיקה האם המשתמש קיים מתבצעת לפי עמודה זו.

בטבלאות Reviews, Places, Pics, OpenHours, Addr, Details אינדקסנו על פי googlePlaceId

**Query desctiptions:**

updateUserQuery, insertUserQuery, inserAddrQuery – שלוש שאילתות עדכון הDB שלנו. כאשר משתמש חדש מתחבר לתוכנה/רוצה לשנות פרטים, הוא צריך להכניס פרטים אישיים שיגיעו לטבלת Userעל ידי אחת משתי השאילתות הראשונות, וכתובת (בעזרת מנגנון השלמה אוטומאטי) שתעדכן את טבלת Addr בעזרת השאילתה האחרונה. הכתובת נשלפת מגוגל בעזרת הAPI שהם מספקים, וכל הפרטים הרלוונטיים נשמרים לטבלת הכתובות.

bestAvgTypeQuery – שאילתה זו מחפשת את סוג מקום הבילוי (מסעדה, באר, חנות, מלון, מועדון) שיש לו את הרייטינג הממוצע הכי גבוה. השאילתה ראשית מחשבת בשאילתה בפנימית את הרייטינג הממוצע הכי גבוה שיש לסוג מקום בילוי כלשהו, ואחר כך בודקת בשאילתה החיצונית לאיזה מקום בילוי יש רייטינג ממוצע זה.

כשלעצמה השאילתה לא מאוד מעניינת, אך אנחנו משתמשים בתוצאה שלה כדי לבצע שליפה נוספת שתביא למשתמש מקומות באזור שלו מהסוג עם הרייטינג הכי גבוה.

placesInDistQuery – שאילתה זו מחפשת את כל מקומות הבילוי מסוג כלשהו שנמצאים ברדיוס מקסימאלי מסויים מנקודה נתונה. הקלטים לשאילתה הם הרדיוס המקסימאלי, סוג מקום הבילוי המבוקש והנקודה בה המשתמש נמצא.

השאילתה נעזרת בשאילתה הפנימית placeAndPics ששולפת עבור כל מקום תמונה יחידה (ואת כל הפרטים על אותו מקום)

השאילתה מחזירה את כל הפרטים מטבלת Places על המקום שהיא מצאה, מזהה הurl (שאינו באמת url בפועל אלא מזהה שעוזר לשלוף תמונה מגוגל) של תמונה ממקום זה, ואת המרחק של המקום.

בנוסף, השאילתה משתמשת בתת שאילתה בשם sub שמחשבת עבור כל מקום את המרחק שלו מהמשתמש. במרחק זה משתמשים כדי לפלטר את התוצאות, והוא חוזר כפלט של השאילתה.

כשאנחנו משתמשים בשאילתה זו בקוד, אנחנו ממיינים את התוצאות שהיא מחזירה לפי מרחק או לפי רייטינג על ידי הוספת שורת order by מתאימה לקוד שלה לפני ההרצה שלה וכך אנחנו מקבלים את התוצאות הכי מתאימות לשליפה שרצינו לבצע.

שליפה זו מחפשת בטבלאות על פי googlePlaceId, ולפי Places.type. כדי להפוך אותה ליעילה יותר אינדקסנו שני שדות אלו בטבלאות המתאימות.

serchInReviewsQuery – שאילתת fulltext. מאפשרת לחפש טקסט רצוי בתוך עמודת הtext של טבלת Reviews. זאת כדי שמשתמש יוכל לחפש משהו מעניין שהוא רוצה בביקורות.

כדי לייעל שאילתה זו, שינינו את מנוע החיפוש של טבלת Reviews לMyISASM, הפכנו את עמודת text למאונדקסת fulltext, והפכנו את עמודת googlePlaceId למאונדקסת (שכן משתמשים גם בעמודה זו כדי לסנן את השליפה ולהתאים את התוצאות לטבלת Addr)

getPictures – שליפה שמאפשרת לקבל את כל התמונות