

דו"ח מיני-פרוייקט בבסיסי נתונים

התשפ"ב

מנחה:

ד"ר יאיר גולדשטיין

מגישים:

יוסף מלכה ואהרן קרמר

תוכן עניינים

2	מבוא.....
3	שלב האפיון:
3	אכלוס הטבלאות:
4	שאלות.....
5	אינטגרציה
6	אינדקסים.....
7	תרשימים.....
8	VIEWS.....
8	לקוחות החברה:.....
8	מתפעל מלאי הרכבים:.....
9	פונקציות ופרוצדורות.....
9	פונקציות:.....
9	פרוצדורות:

מבוא

נושא הפרויקט: בסיס מידע עבור חברה להשכרת רכב

פלח בסיס המידע בבעלותנו: מאגר הרכבים והנלווה

תיאור מורחב: בסיס המידע שמימשנו מהווה את לב ליבו של בסיס המידע הכללי: מלאי הרכבים, מודלים שונים, טיפולים ומוסכים.

הנחת העבודה היא כי השימוש בשאילתות למאגר המידע המכיל את פרטי הרכבים הוא השימוש העיקרי של בסיס הנתונים הכללי ולכן עליו להיבנות בדקדוק תוך בחינת הפרטים הרלוונטיים לטבלאות הנ"ל. במקביל וכהשלמה למלאי הרכבים, טבלאות הטיפולים והמוסכים מהווים את הנטל הכלכלי הכבד ביותר עבור החברה, ויתר על כן, המידע הנאגר בטבלאות אלו עשוי להיות פקטור משמעותי בבחירת סוג הרכבים שתבחר החברה להחזיק ברשותה. ניתן לראות התייחסות לכך בחלק מן השאילתות.

נציין כי על אף רכיב ההדמייה בפרויקט, השתדלנו לאסוף נתוני אמת על מנת לייצר תמונה מציאותית עד כמה שניתן. נציין את מאגרי המידע שבהם עשינו שימוש:

- מאגר מוסכים: מתוך מאגרי מידע ממשלתיים – <https://data.gov.il/dataset/musachim/resource/bb68386a-a331-4bbc-b668-bba2766d517d>
- מאגר רכבים לפי חברה ומודל: <https://www.autoevolution.com/cars/>
- לוחיות רישוי: הגרלה רנדומלית של ערכים תחת אילוף פורמט המחוייב על פי החוק הנוהג במדינה
- טבלת התפלגות צבעי המכוניות מבוססת על סטטיסטיקה קיימת (google) של צבעי המכוניות בעולם האמיתי

הערה: בשל אילוצי התוכנה שבה הונחנו לעבוד ובשל הרצון לייצר מצג אמת ולא להגריל שמות חסרי משמעות של מקומות בדויים שאינם מדמים מאגר אמיתי, המידע המחזיק את שמות המוסכים בפונט עברי מוצג כג'יבריש בתצוגת pl/sql, ניתן לראות כי שמות המוסכים הינם השמות המקוריים על ידי שאילתא שעושה שימוש בשם מוסך, התוצאה מופיעה בג'יבריש בעוד השאילתא מחזיקה שם של מוסך מקורי בפונט עברי (UTF-8). חיסרון זה יכול להפתר בנקל על ידי מעבר לתוכנת בסיסי נתונים מתקדמת יותר.

שלב האפיון:

את שלב האפיון בחרנו לבנות באמצעות ERDplus אונליין.

יתרונות המידול בתוכנה זו מתבטאים בעיקר במעבר קל ממודל ה-ERD לסכמות. כמו כן, בתוכנה זו ניתן ליצור סקריפטים של SQL באופן אוטומטי. השימוש האוטומטי באופן הנ"ל עוזר למנוע טעויות כתיבה וכדומה.

אכלוס הטבלאות:

עבור שלב זה השתמשנו בשני כלים עיקריים: TypeScript ו-Python, הבחירה ב-TypeScript נבעה בצורך לאסוף מידע ממאגרי HTML, לשפה זו כלים נוחים ויעילים לשימוש זה. לעומת זאת, על מנת לייצר קבצי CSV מהמאגרים השונים, בחרנו להשתמש בשפת python, ל-python קיימות ספריות רבות המאפשרות שימוש מגוון ונוח ליצירת הטבלאות.

כאמור, חלק מרכזי ממאגר המידע נמצא באתרי אינטרנט המכילים טבלאות בפורמט HTML, פורמט זה איננו נוח לשימוש ולכן חילצנו מהם את המידע על ידי שימוש ב-TypeScript.

את המידע שמרנו בפורמט CSV, הבחירה בפורמט זה היא הבחירה הטריטוריאלית לצורך בנייה עתידית של טבלאות SQL. הפורמט הוא הקל ביותר מבין האפשרויות השונות והפשוט ביותר לפירסור.

נעיר כי העדפנו שלא להשתמש בחלק מן הכלים השונים שהעמיד לרשותנו המנחה. הסיבה נובעת מטבעם הגנרי של כלים אלה. אנו בחרנו לעבוד עם נתוני אמת ואילוצים שונים המדמים אותם, לא ניתן לייצר נתונים מסוג זה בכלים אלה.

שאלות

לפנינו 10 שאלות, בבניית השאלות חשבנו על כמה צרכים: האחד הוא עבור הפעילות השוטפת של החברה, חיפוש רכבים במוסכים השונים, ניטור עומסים וכדומה. השני עבור דרישות לקוח מול החברה, בחירת צבע מסוים או מודל מסוים וכדומה. השלישי עבור תכנון כלכלי וראייה עתידית של החברה, למשל ניטור מודלים בעייתיים, והחרגת רכבים בעייתיים משירותי החברה.

1. `how_many_cars_we_have_from_each_model.sql`

שאלתא זו מנטרת את מספרי הרכבים (כמות ולא מספר רשיון) מכל סוג מודל. שאלתא זו הכרחית על מנת לאפשר לחברה לגוון בצורה נכונה ובקלות את סוגי הרכבים שברשותה.

2. `select_all_cars_with_specific_color.sql`

שאלתא זו מנטרת את כל הרכבים לפי מודל מצבע מסוים. שאלתא זו נותנת מענה לבקשות לקוחות החברה שפעמים רבות מעוניינים בהשכרת רכב בעל צבע מסוים, באם צבע זה איננו עבור מודל מסוים אזי באמצעות שאלתא זו מתאפשר לבחור רכב ממודל אחר בעל אותו צבע.

3. `select_all_new_cars.sql`

שאלתא זו מנטרת את כל הרכבים החדשים (לפי שנה). שאלתא זו שייכת בעיקר לקטגוריה של דרישות לקוח החברה, אולם גם עוזרת לניהול מלאי רכבים תקין

4. `select_all_new_models.sql`

שאלתא זו משלימה את השאלתא הקודמת, היא מחזירה את רשימת המודלים החדשים הקיימים בחברה

5. `select_all_cars_with_specific_company.sql`

שאלתא זו מנטרת את כל הרכבים השייכים לחברה מסוימת. לשאלתא זו שימושים רבים, בין השאר היא מועילה לשליטה על מלאי הרכבים ויצירת חוזים עם חברות הרכב השונות. בנוסף, בדומה לשאלתא הקודמת, ניתן להיעזר בשאלתא זו למתן מענה ללקוח החברה המעוניין דווקא בחברה זו.

6. `select_all_the_models_of_specific_company.sql`

שאלתא זו משלימה את השאלתא הקודמת, היא מחזירה את רשימת המודלים לפי חברה

7. `which car got the most fixes.sql`

שאלתא זו מנטרת את הרכב שנזקק למספר הרב ביותר של טיפולים. היא מחזירה את רשימת כל הרכבים ממויינים לפי מספר טיפולים. בין השאר, שאלתא זו מועילה לחברה להחליט איזה רכבים להחריג משירות החברה.

8. `which model needed the most treatments.sql`

שאלתא זו משלימה את השאלתא הקודמת. היא מחזירה את המודל שנזקק למפר הרב ביותר של טיפולים. בין השאר, שאלתא זו מועילה לסייע לחברה לדעת איזה מודל לא לרכוש עבור החברה.

9. `how_many_cars_in_each_garage.sql`

שאלתא זו מנטרת את מספרי הרכבים (כמות ולא מספר רשיון) הנמצאים בכל מוסך.
שאלתא זו מהווה חלק ממנגנון השליטה של החברה ברכביה. כמו כן שאלתא זו מאפשרת לחברה להחליט לאן לשלוח את רכביה לטיפול באופן שלא ייצר עומס במוסך.

10. `which_cars_are_in_specific_garage.sql`

שאלתא זו מחזירה את כל הרכבים הנמצאים במוסך מסויים.

אינטגרציה

בחרנו לבצע אינטגרציה עם הטבלאות הטבעיות ביותר לטבלאות הרכבים – טבלאות הסניפים. שילוב בין הסניפים והרכבים מרחיב את אפשרויות השאלתא ומייצר שאלות יעילות ומשופרות.

רשימת השאלות עבור אינטגרציה:

1. `THEY ARE GET ALL CARS FROM SPECIFIC COMPANY AND WHERE`

שאלתא זו מנטרת את כל הרכבים על פי חברה מסוימת ואת מיקומם לפי סניף.

2. `HOW MANY AVAILABLE CARS WE HAVE IN EACH BRANCH`

שאלתא זו אחת השאלות החשובות. תפקידה לנטר את כל הרכבים שאינם בטיפול עבור כל סניף וסניף.

3. `HOW MANY CARS IN EACH BRANCH`

שאלתא זו מחזירה את מספר הרכבים בכל סניף.

1. CARS_STOCK_BRANCH_ID.sql

האינדקס פועל על מזהה הסניף של רכב בטבלת מלאי הרכבים. השימוש באינדקס להפעלת השאילתא HOW MANY AVAILABLE CARS WE HAVE IN EACH BRANCH יצר שיפור משמעותי של 47% בזמן הרצת השאילתא (ללא הורדת הנתונים)

2. CARS_STOCK_MODEL_ID.sql

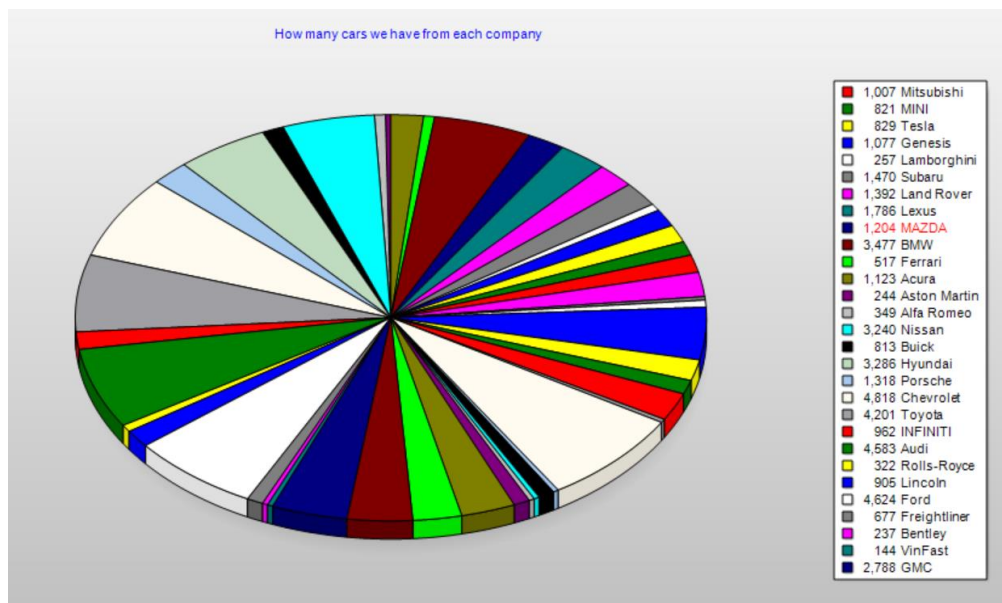
האינדקס פועל על מזהה הסניף של רכב בטבלת מלאי הרכבים. השימוש באינדקס להפעלת השאילתא WHICH MODEL NEEDED THE MOST TREATMENTS יצר שיפור משמעותי של 63% בזמן הרצת השאילתא (ללא הורדת הנתונים)

3. TREATMENTS_CSID.sql

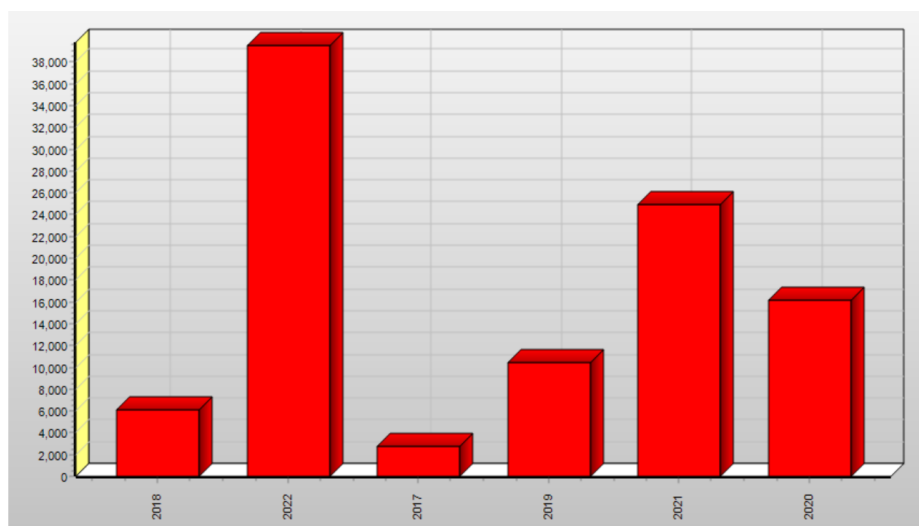
האינדקס פועל על מזהה הסניף של רכב בטבלת מלאי הרכבים. השימוש באינדקס להפעלת השאילתא which car got most fixes יצר שיפור משמעותי של 60% בזמן הרצת השאילתא (ללא הורדת הנתונים)

תרשימים

התרשימים שבנינו מסייעים לתחזוקת מלאי הרכבים. הרכיב הויזואלי מקל על אנשי התחזוקה בקריאת הנתונים. להלן תרשים של מספר הרכבים על פי סוג חברה שמחזיקה חברה ההשכרה:



התרשים הבא מייצג את מספר הטיפולים לפי שנה, התרשים נועד לסייע בבחינת הוצאות כלכליות שנתיות כאשר ההנחה היא שהוצאות התחזוקה הגבוהות ביותר הן על טיפולי רכב:



VIEWS

בניית טבלאות ה-VIES התבססה על שני משתמשים עיקריים – לקוחות החברה ומתפעל מלאי הרכבים.

לקוחות החברה:

1. FULL_CARS_DATA.sql

הטבלאות הנוגעות ללקוחות החברה צריכות להכיל מספר פרטים רב ככל הניתן של הרכבים, כך, לקוח המעוניין בסוג רכב מסוים, לפי צבע, מודל, חברה, מספר סניף, וכדומה, יוכל לקבל באופן מיידי את הרכבים העומדים לבחירתו על פי דרישותיו. לכן יש צורך בטבלה המכילה את כל פרטי הרכבים שניתן לייצר.

2. CARS_AVAILABLE_FULL

גם הצורך בטבלה זו הוא טריוויאלי, טבלה זו מכילה את המידע על כל הרכבים הזמינים, לפי סניף ומיקום.

מתפעל מלאי הרכבים:

3. FULL_TREATMENTS.sql

טבלת היסטוריית הטיפולים נועדה לתת מענה לאחראי תפעול הרכבים, טבלה זו מועילה לשימושים רבים, כמו למשל החרגת רכבים מהחברה, בחירת רכבים חדשים לקנייה, בדיקה אם רכב נמצא בטיפול כעת והיכן ועוד.

4. cars_in_each_model.sql

טבלה זו נוגעת לתחזוקת המלאי. היא מהווה למעשה ספירת מלאי של הרכבים לפי סוג המודל.

פונקציות ופרוצדורות

פונקציות:

נעשה שימוש ב- 4 פונקציות שנועדו להקל את חווית התפקוד השוטף של אחזקת הרכבים:

1. CARS_IN_BRANCH.fnc

פונקציה המקבלת מספר סניף ומחזירה את מספר הרכבים שנמצאים בו.

2. SMALLEST_BRANCH.fnc

פונקציה המחזירה את מספר הסניף בו קיים מספר קטן ביותר של רכבים.

3. largest_branch.fnc

פונקציה המחזירה את מספיר הסניף בו קיים מספר גדול ביותר של רכבים.

4. buy car.fnc

פונקציה המטפלת בהליך קניית רכב חדש לחברה, כל רכב שנכנס לחברה עובר טיפול של שלושה ימים במוסך, הפונקציה מקבלת מספר רכב ומחזירה את מספר המוסך המטפל ברכב.

פרוצדורות:

1. move_cars_from_big_branch_to_small_one.prc

הפרוצדורה נועדה לאזן בין מספרי הרכבים שבסניפים. היא מעבירה רכבים מהסניף בו מספר הרכבים הגדול ביותר לסניף בו מספר הרכבים הקטן ביותר. נעיר כי הפונקציה בנויה בצורה אטומית, כלומר היא פועלת על שני סניפים, אולם הרצת הפרוצדורה מספר רב של פעמים מייצרת איזון בסניפי החברה כולה.

2. sell_car.prc

הפרוצדורה הנ"ל למעשה מבצעת פעולת delete_car, לרוב פעולה זו נגזרת ממכירת רכב אולם גם החרגה של רכב ממלאי הרכבים וכדומה מייצר פעולה דומה. למעשה פרוצדורה זו מנקה את בסיס הנתונים מכל פרטי הרכב.