**בס"ד**

**מיני פרויקט בבסיסי נתונים**

# הפרויקט שלנו

הקבוצה שלנו בנתה את בסיס הנתונים הכללי ונתנה הרשאות לשאר הקבוצה. לפיכך יש לנו את הERD הכללי של כולם. הקבוצה שלנו התעסקה בעיקר בפרסומות, ובתחומים הקשורים. התעסקנו עם כמות פרסומות לסרט או לאזור, מחירם וכו'.

## תרשים ERD

תרשים הERD שלנו

## C:\Users\oehrlich\Downloads\erdplus-diagram (1) (1).png

## תיאור הישויות והקשרים

### ישויות

* Advertisement - ישות זאת אחראית על כל הפרסומות הנמצאות במערכת.

ישות זאת הינה חלשה, כיוון שתלויה בישות סרט. בנוסף יש טבלה של קשרים בין איזורים לפרוסומת.

* ADTYPE – סוג הפרוסמת ( טלוויזיה, שלט, חדשות, גדול/קטן/בינוני וכו')
* Adcost –עלות הפרסומת באלפי שקלים
* ADID – מזהה פרסומת
* Movieid – מזהה הסרט שעבורו הפרסומת. רפרנס.
* area – ישות זאת אחראית על כל האזורים.

ישות זאת הינה חזקה.

* areaid – מספר מזהה של האזור
* areaname – שם האזור
* country – שם המדינה
* city – ישות זאת אחראית על כל הערים הנמצאים במערכת.

ישות זאת הינה חלשה, כיוון שהיא תלויה בעיר.

* cityid – מספר מזהה של העיר
* Cityname – שם העיר
* areaid – מספר מזהה של האיזור. רפרנס.
* movies – ישות זאת אחראית על כל הסרטים הנמצאים במערכת.

ישות זאת הינה חזקה.

* movieid – מספר מזהה של סרט
* Removaldate –תאריך הורדה
* moviename – שם הסרט
* Genre – שם הז'אנר
* Rating – דירוג הסרט
* Releasedate – תאריך העלאה
* releaseYear – השנה של הסרט

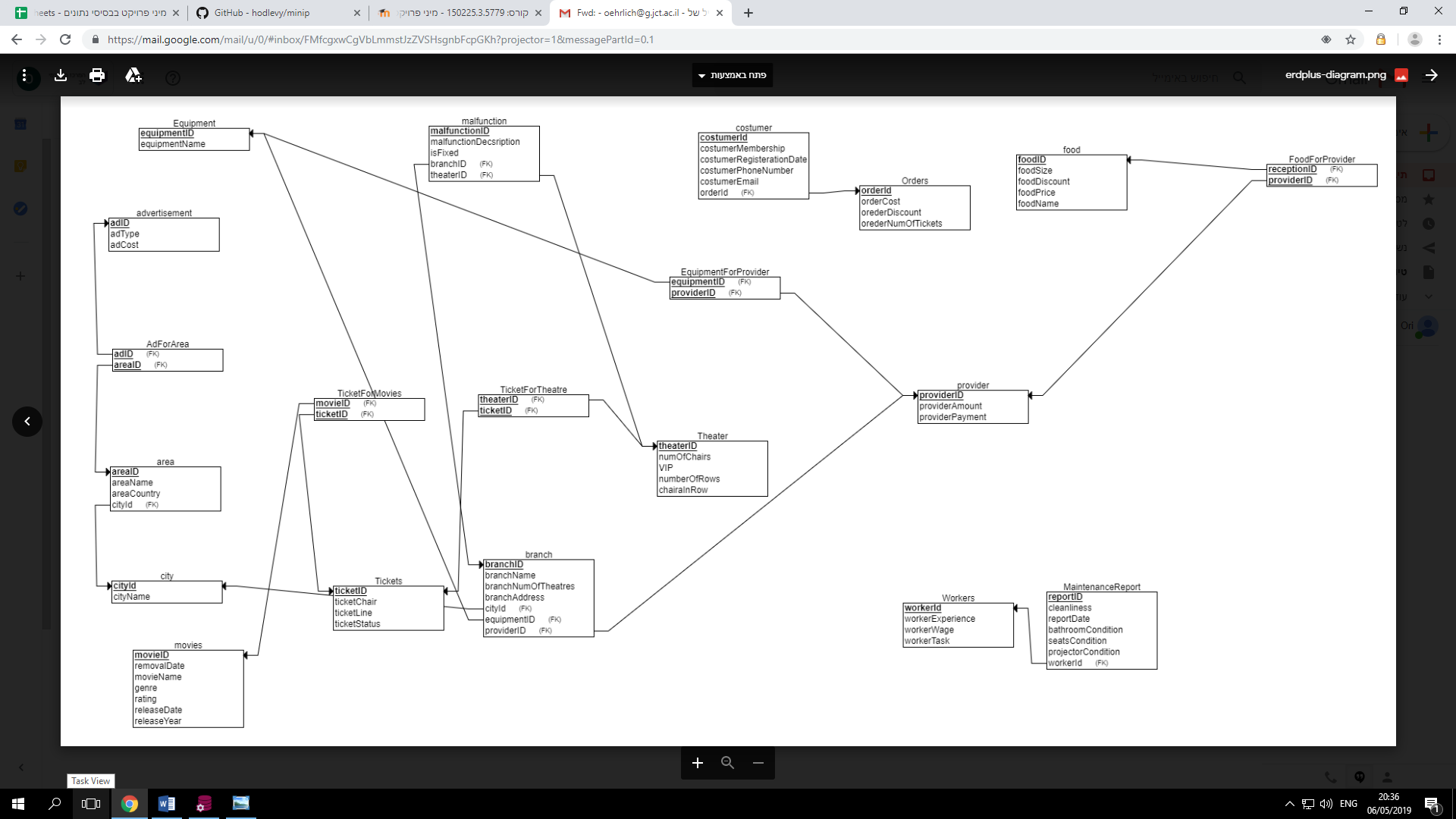
### קשרים

* adforarea – הקשר בין area לבין ad. הוא משמש כדי לדעת איזה פרסומות היו באיזה איזורים. הקשר הוא רבים לרבים כיוון שיש כמה פרסומות בכל אזור, וכל פרסומת בכמה איזורים.

## נרמול הטבלאות

היחסים עומדים ב- NF3 וב- BCNF : מכיוון שבכל טבלה, התלויות הפונקציונאליות הלא-טריוויאליות הן מהמפתח אל תכונות נוספות לכן מתקיים שלכל X Y , X הוא מפתח ולכן הם עומד בתנאים.

## תרשים DSD



## 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 הcreate table.

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

## הכנסת נתונים

השתמשנו בData generator של הPLSQL. קבצים בGIT.

## שאילתות SQL

### בחירה - SELECT

לאחר שיצרנו את בסיס הנתונים והכנסנו לתוכו מידע, כתבנו כמה שאילתות מעניינות על מנת לתשאל אותו.

1. קיבוץ כל הערים בהם יש יותר מ100 סנפים, ביחד עם מספר הסניפים. על מנת לדעת – באיזה איזורים יש סניפים רבים וכדאי להשקיע.
2. עבור כל עיר – כמה סניפים יש בממוצע לכל עיר? נפקא מינא איך להעריך איזורים מסוימים, במה להשקיע, איפה משתלם וכו'. באיזור עם המון סניפים יהיה כדאי להשקיע יותר, כמובן.
3. אנחנו מתעניינים בטלוויזיה – כמה פרסומות טלוויזניות יש, ומה מחירם הממוצע? האם זה משתלם? האם לא? תשאלו את רואה החשבון, אני בסך הכל DBA, מה אני אמור להבין בזה?
4. קיבוץ איזורים – עבור כל איזור – כמה פרסומות יש בו, ומה מחירם הממוצע? עבור כל אזור – כמה השקענו? האם זה משתלם?
5. קיבוץ סרטים שיש להם יותר משלוש פרסומות שהמחיר הממוצע שלהם מעל 50,000. זה נותן את המחיר הממוצע ומספרם.
6. ריכוז כל הסרטים שיצאו ב1998, ולהם יותר מ7 פרסומות. בשנת 1998 קרה X שחשוב למען סטטיסטיקות בהנהלה.
7. קיבוץ כל המדינות בהם יש פרסומת לסרט 890
8. קיבוץ הסרטים בהם סך המחיר עלה מעל 600,000. כדי לדעת כמה שילמנו על סרטים מאוד יקרים. זה דבר מאוד שימושי וחשוב.
9. בכל אזור כמה ערים יש

**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 (בעזרת אינדקס שמכיל את המזהים). 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.

את כל הבדיקות עשינו ללא cache ע"י שסגרנו את התוכנה ( ולפעמים כיבינו את המחשב ), על מנת שזה לא יושפע מהcache.

צילומי מסך והשאילתות נמצאות בגיט.

האינדקס הראשון שעשינו היה על rating, ולפיכך כאשר חיפשנו משהו בrating מסוים מאוד, זה היה מהיר בערך פי עשר. הסיבה לכך שיש המון נתונים, וrating מתפלג באופן כזה שעבור מספר נתון יש כמות קטנה שמתאימה, ואז השימוש באינדקס מצמצם חיפוש ארוך. זו שאילתא AAA בגיט.

לפני אינדקס – 0.429

אחרי אינדקס – 0.049

האינדקס השני שעשינו היה על releaseyear. יש עשרה אופציות והנתונים מתחלקים באופן שווה. מכיוון שזה המון אופציות, ולא אחוזים נמוכים מספיק, החיפוש בטבלת האינדקס רק מוסיף זמן – כי הוא דורש חיפוש מורכב יותר בתוך טבלאות.

לפיכך, בשאילתא KKK ראינו שהאינדקס סתם מריע את המצב.

לפני אינדקס - 0.161

אחרי אינדקס – 0.253

בפועל מה שיקרה שהשרת ילמד לא להשתמש באינדקס כי זה רק מאט, ויעבוד עם מטמון או בלי אינדקס.

האינדקס הנוסף היה על areacountry – יש המון מדינות, ובמדינות שבהם יש מעט אזורים יוצא יעיל להשתמש באינדקס. בחרנו נתון מסוים שיתאי לזה על מנת לבדוק – ויצא ששיפרנו בערך פי שניים. הסיבה לכך שזה מעט ערכים מתאימים מתו רבים מאוד.

לפני אינדקס – 0.079

אחרי אינדקס – 0.041