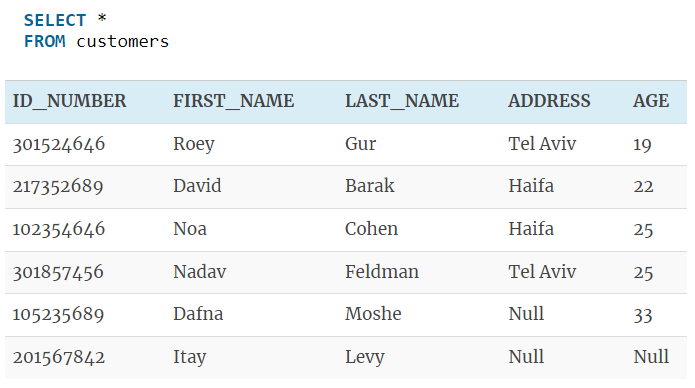
# טעויות נפוצות בתחום ה-NULL VALUES

טעויות בתחום ניתוח נתונים קורות לכולם, כל הזמן, ללא קשר לוותק במקצוע. הבעיה האמיתית מתחילה עם טעויות שאנחנו לא יודעים שבכלל עשינו. בפועל, אלה הבעיות הכי יומיומיות איתם מתמודד אנליסט. לרוב לומדים את זה בדרך הקשה אחרי שעשינו כמה טעויות בעבודה, אבל אפשר וגם רצוי, ללמוד מטעויות של אחרים. כמובן שלא תמצאו כאן רשימה ממצה של כל הטעויות שאתם יכולים לעשות, אבל לכל הפחות אפשר להתחיל לפתח מודעות לטעויות אפשריות, מה שיעזור להימנע מהן.

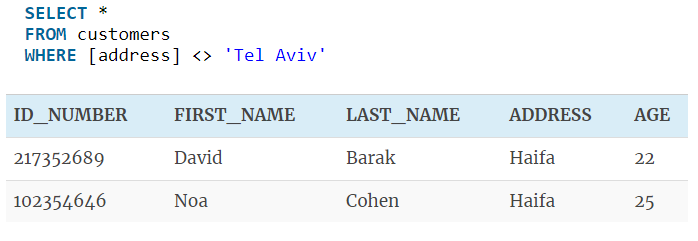
**עבודה עם Null Values -** אחד הנושאים המבלבלים ביותר בעבודה עם דאטה קשור לעבודה עם ערכי Null. מה שחשוב להכיר הוא ש-Null אינו Data Type אמיתי. למעשה אפשר להסתכל עליו כNon-Value.Null מייצג שדה ריק – הוא לא מייצג '0' וגם לא מייצג blank space. בואו נראה איזה בעיות זה עלול ליצור לנו.

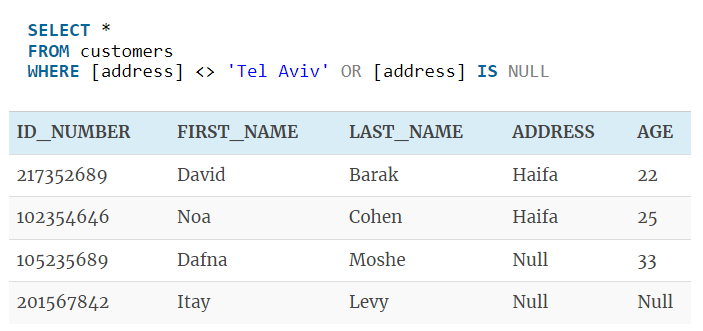
**טעות #1 – שימוש באופרטורים על NULL**

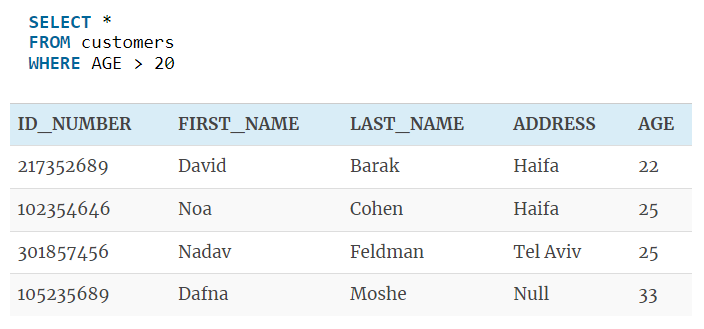
כאנליסטים יוצא לנו לעשות השוואות על שדות מסוג String וכן על שדות מספריים ( Int, floatוכו). באמצעות '>', '<' או '='. מכיוון שאמרנו ש-Null הוא Non Value, לא 0 ולא blank space, אי אפשר להשוות אליו ערך אחר. הטעות הכי חמורה של אי התייחסות לעמודה שמכילה Null תיווצר בפקודת WHERE. נסביר זאת עם כמה דוגמאות.

נתונה הטבלה הבאה של לקוחות החברה**:**

אם ננסה לחפש את לקוחות החברה שהכתובת שלהם אינה Tel Aviv, יתכן ונכתוב שאילתא כזו:

כפי שאתם רואים סיננו בטעות שני לקוחות שכתובתם הייתה Null - זאת משום שSQL מתעלם מכל סוג של Comparison Operations על Null (מאחורי הקלעים, הוא מתייחס לזה כ-Unkonwn ולא מציג את התוצאה).

שאילתא נכונה יותר הייתה צריכה לכלול התייחסות לערכי Null:

אותו קונספט חל על השוואת Null לכל דבר אחר. הנה דוגמא של השוואה על שדה מספרי באותה טבלה:

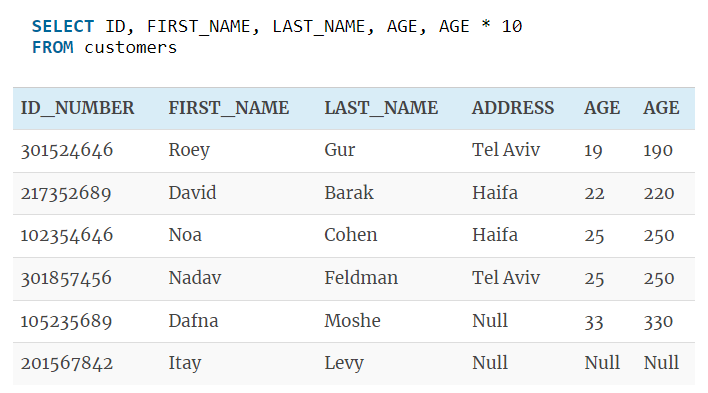
כפי שניתן לראות, שני ערכים סוננו מחוץ לטבלה: רועי גור בן ה19, ואיתי לוי שגילו בטבלה הוא NULL. השוואה ל-NULL איננה אפשרית כי Null אינו data type"אמיתי" ולכן SQL התעלם מתוצאה זו.

**טעות #2 – ביצוע פעולה חישובית על NULL**

כל פעולה חישובית (חיבור, חיסור, חילוק, הכפלה) על ערך Null אינה אפשרית משום שהוא Non-Value. לשם הדוגמא, Null \* 10 לא שווה 10, אלא Null.

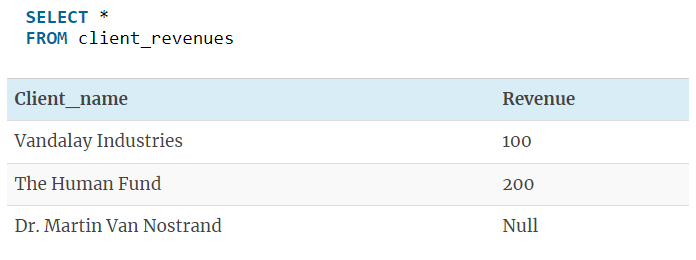
SQL יחזיר פעולת חישוב על Null כ-Null. למה? כי Null מייצג ערך לא קיים.

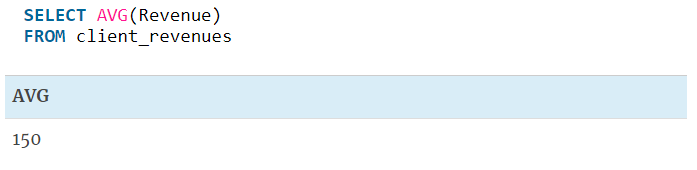
10 \* כלום = כלום.

אם נחזור לטבלת הדוגמא שלנו, חישוב על ערך Null יראה כך:

**טעות #3 – אגרגציה ללא התייחסות ל-NULL**

כפי שאמרנו מבחינת המחשב שלכם,Null מייצג שדה ריק. כשנעשה אגרגציה,SQL יפיל ערכים מסוג Null ויצור מעין עיוות בנתונים שיתכן שלא התכוונו אליו.

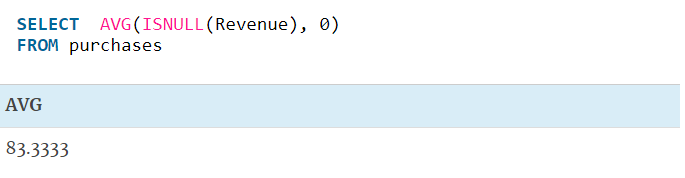
אדגים בעזרת דוגמא פשוטה. נתונה הטבלה הבאה ובה לקוחות וההכנסה שלהן:

נניח ונרצה לברר את ממוצע ההכנסות של כלל החברות אותן אנחנו מנהלים. ככל הנראה נכתוב קוד כזה:

מכיוון ש-Dr. Martin Van Nostrand לא ביצע עסקאות החודש, הוא קיבל את הערך NULL. ניתן לראות שחישוב הממוצע התבצע על שני הערכים שאינם Null וערך ה-Null סונן החוצה לפני ביצוע החישוב. כש-SQL מבצע אגרגציה, הוא לא מתייחס לערך Null כ-0 למרות שיתכן שלזה התכוונו כשביקשנו למצוא ממוצע לכל חודש.

מה עושים כדי לפתור את זה? משתמשים בפונקציית isnull. פונקציה שימושית מאוד שכופה על SQL את הלוגיקה הבאה: אם הערך הוא Null תהפוך אותו ל-X. במקרה שלנו, נרצה להפוך אותו ל-0.

הפונקציה מקבלת שני פרמטרים, הראשון הוא העמודה שתרצו לבדוק אם היא מכילה ערך Null והשני הוא הערך שתרצו שיקבל במקום. זה יכול להיות 0 אבל יכול להיות גם כל ערך אחר כמו String או Boolean.

ניתן לראות כי כאן השתמשנו ב-0 ועכשיו החישוב כלל גם את ההכנסה שלא התקבלה מ-Dr. Martin Van Nostrand.