x x

אבינועם סבע – 337966642

אליעזר לנקרי - 588939

<https://github.com/AvinoamSebbah/BSISEINETOUNIM>

**בס"ד**

**מיני פרויקט בבסיסי נתונים**

תוכן עניינים

[מבוא 2](#_Toc43377489)

[עבודת הכנה והכרת התוכנה 3](#_Toc43377490)

[תרשים ERD 3](#_Toc43377491)

[תיאור הישויות והקשרים 4](#_Toc43377492)

[תרשים DSD 5](#_Toc43377522)

[הפרויקט שלנו 6](#_Toc43377523)

[תרשים ERD 6](#_Toc43377524)

[תיאור הישויות והקשרים 7](#_Toc43377525)

[נרמול הטבלאות 8](#_Toc43377528)

[תרשים DSD 8](#_Toc43377529)

[יצירת הטבלאות 9](#_Toc43377530)

[הכנסת נתונים אכלוס הטבלאות 10](#_Toc43377531)

[שאילתות SQL 11](#_Toc43377532)

[אינדקסים 12](#_Toc43377533)

[הרשאות 16](#_Toc43377534)

[שאילתות על כמה אגפים 17](#_Toc43377535)

[Views 18](#_Toc43377536)

[גרף 21](#_Toc43377537)

[פרוצדרות 24](#_Toc43377538)

[פונקציות 26](#_Toc43377539)

[סוף פרויקט 28](#_Toc43377540)

[נספחים 0](#_Toc43377541)

[נספח ראשון: SQL 0](#_Toc43377542)

[נספח שני: אכלוס טבלאות 1](#_Toc43377543)

[נספח שלישי: שאילתות ואינדקסים 3](#_Toc43377544)

[נספח רביעי : שאילתות על איחוד אגפים 6](#_Toc43377545)

[נספח חמישי : View וגרפים 8](#_Toc43377546)

[נספח שישי : פונקציות ופרוצדרות 12](#_Toc43377547)

# מבוא

* **אגך הרפואה**

בית חולים :שם, כתובת, מחלקות, מספר זהות, קיבולת.

אנשי רפואה :דרגה, ת"ז, שם, מחלקה, גיל, שכר, מקום מגורים, לוז זהות, מספר טלפון.

חולים :מצב הנוכחי, ת"ז, שם, גיל, מקום מגורים, סביבתו, מספר טלפון.

* **אגף המדעים**

מעבדות : מספר זהות, שם, רמת הגנה, מיקום.

מדענים : ת"ז, שם, גיל, שכר, התמחויות, מספר טלפון.

וירוסים : שם, קוד ה-דנא/מס' זהות, מקור, רמת סכנה.

תרופות : שם, סוג, מלאי, מספר זהות, חברת ייצור, כמות.

* **אגף הבית חולים**

בית חולים :שם, כתובת, מחלקות, מספר זהות, קיבולת.

חדרים פנויים : סוג, שם, מספר זהות, קיבולת, פנוי.

חולים :מצב הנוכחי, ת"ז, שם, גיל, מקום מגורים, סביבתו, מספר טלפון.

* **אגף האוכלוסייה**

עיר התפשטות : שם, מספר זהות, מצב נוכחי, מספר תושבים.

רשתות בדיקה : מיקום, סוג בדיקה, קיבולת, ת"ז רשת.

אנשים : בהסגר,מחכים לבדיקה

* **אגף עובדים**

מתנדבים : ת"ז, שם, כתובת, מס' טלפון, תפקיד התנדבות.

אמבולנס : מספר חברה, מס' טלפון, מס' רישיון, מרכז רפואי.

בית חולים :שם, כתובת, מחלקות, מספר זהות, קיבולת.

* **אגף המלאי**

מלאי רפואי : שם, מספר זהה, כמות, מקור.

רשתות משלוחים אוכל תרופות : חברה, אזור, מספר זהות, מלאי, יבואן.

חברות ייצור ציוד רפואי : שם, מספר זהות, סוג ציוד.

* **אגף הלוגיסטיקה**

חדרים פנויים : סוג, שם, מספר זהות, קיבולת, פנוי.

לוח זמנים : שעות, מחלקה, יום\לילה, לוז זהות.

ארוע : זמן התחלה, זמן סיום, מיקום, תיאור, אירוע זהות, לוז זהות.

מלון להסגר : שם, מיקום, קיבולת, אנשים במלון, מלון זהות.

* **אגף המתנדבים**

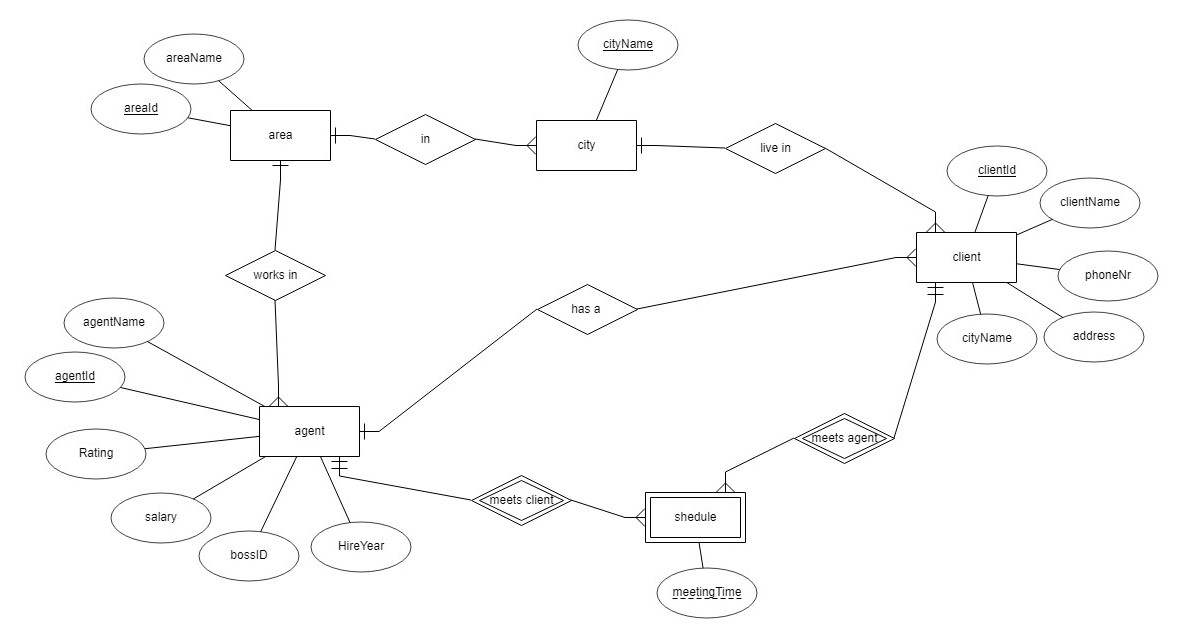
מתנדבים : ת"ז, שם, כתובת, עיר, התמחויות, מס טלפון.

רשתות התנדבות : חברה, מס זהות ,עיר, התמחויות, ת.ז מתנדבים.

רכזי התנדבות : שם, ת.ז, מס טלפון, כתובת.

עבודת הכנה והכרת התוכנה

## **תרשים ERD**



## תיאור הישויות והקשרים

### ישויות

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Client | ת.ז של הלקוח | שם | כתובת | מספר פלאפון | עיר מגורים | מספר סוכן |  |
| Agent | ת.ז של הסוכן | שם סוכן | מספר אזור | דירוג | תק | ת.ז של הבוס | משכורת |
| Area | מספר אזור | שם האזור |  |  |  |  |  |
| City | מספר אזור | שם העיר |  |  |  |  |  |
| Schedule | ת.ז של הסוכן | ת.ז של הלקוח | זמן הפגישה |  |  |  |  |

### קשרים

|  |  |
| --- | --- |
| לכל אזור | יכולות להיות כמה ערים ויכולים להיות הרבה סוכנים שעובדים באזור זה |
| לכל עיר | יכולים להיות הרבה לקוחות שגרים בה והיא יכולה להיות באזור יחידי |
| לכל סוכן | יש אזור עבודה יחידי, יכולים להיות הרבה לקוחות ויכולים להיות לו הרבה פגישות עם לקוחות |
| לכל לקוח | יש סוכן יחיד והוא גר בעיר יחידה ויכול להיות לו הרבה פגישות עם הסוכן שלו |
| לכל לוח זמנים | חייב להיות סוכן יחיד ולקוח יחיד הנפגשים בזמן מסוים |

### פרוקים

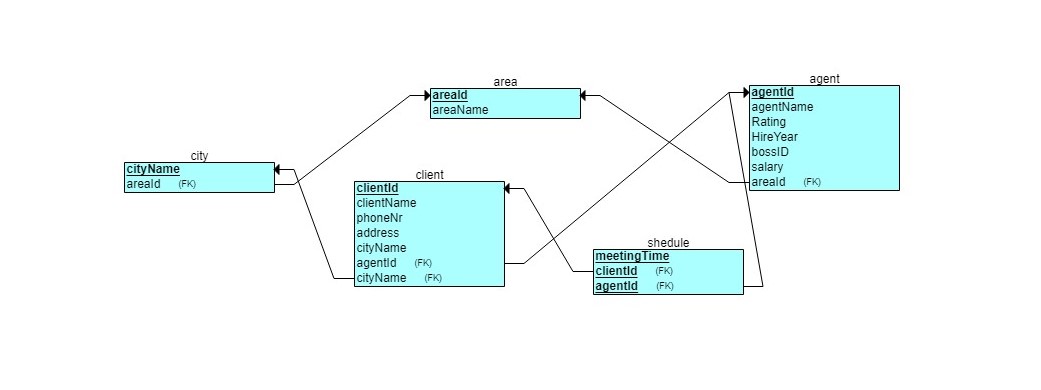
כל התלויות הן BCNF כיוון שה X בתלות הינו מפתח על. ולכן כמובן כל התלויות הינן 3NF .

### נרמול הטבלאות

* + area (areaID, areaName)
  + city (cityName, areaID)
  + agent (agentID, rating, hireYear, boosID, salary, areaID)
  + client (clientID, clientName, address, phoneNr, cityName)
  + schedule (agentID, clientID, meetingTime)

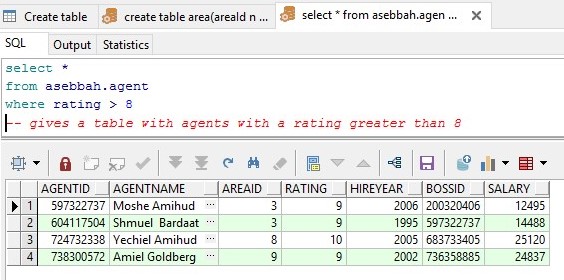
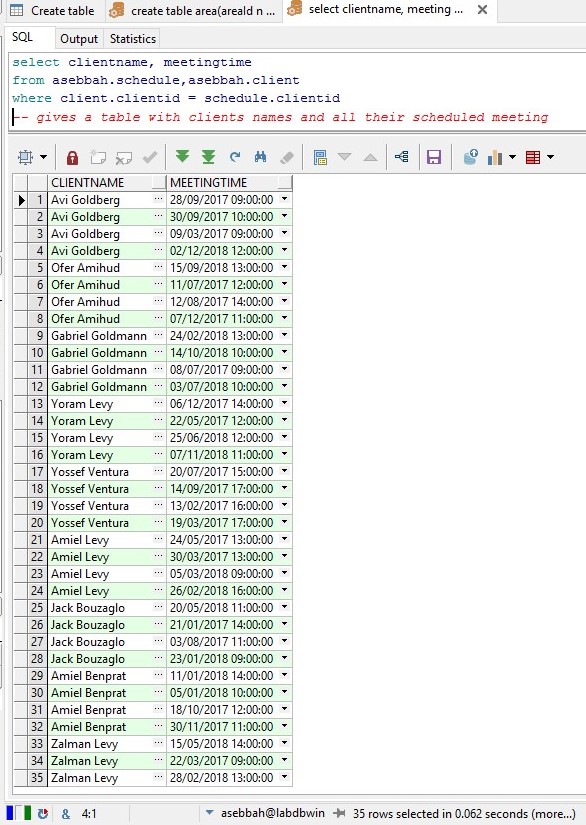
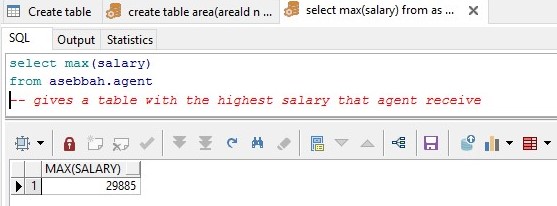
## תרשים DSD

כאן ניתן לראות את ה- DSD שהפקנו מתרשים ה- ERD שיצרנו.

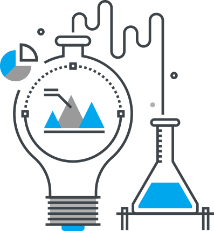
****

**טעינת בסיס הנתונים** **הגרעיני ל-PL/SQL Developer**

לאחר יצירת הטבלאות וטעינת נתונים, אני אנסה כמה שאילתות לראות אם הכל עובד טוב עם הבסיס הנתונים הגרעיני שקיבלנו :



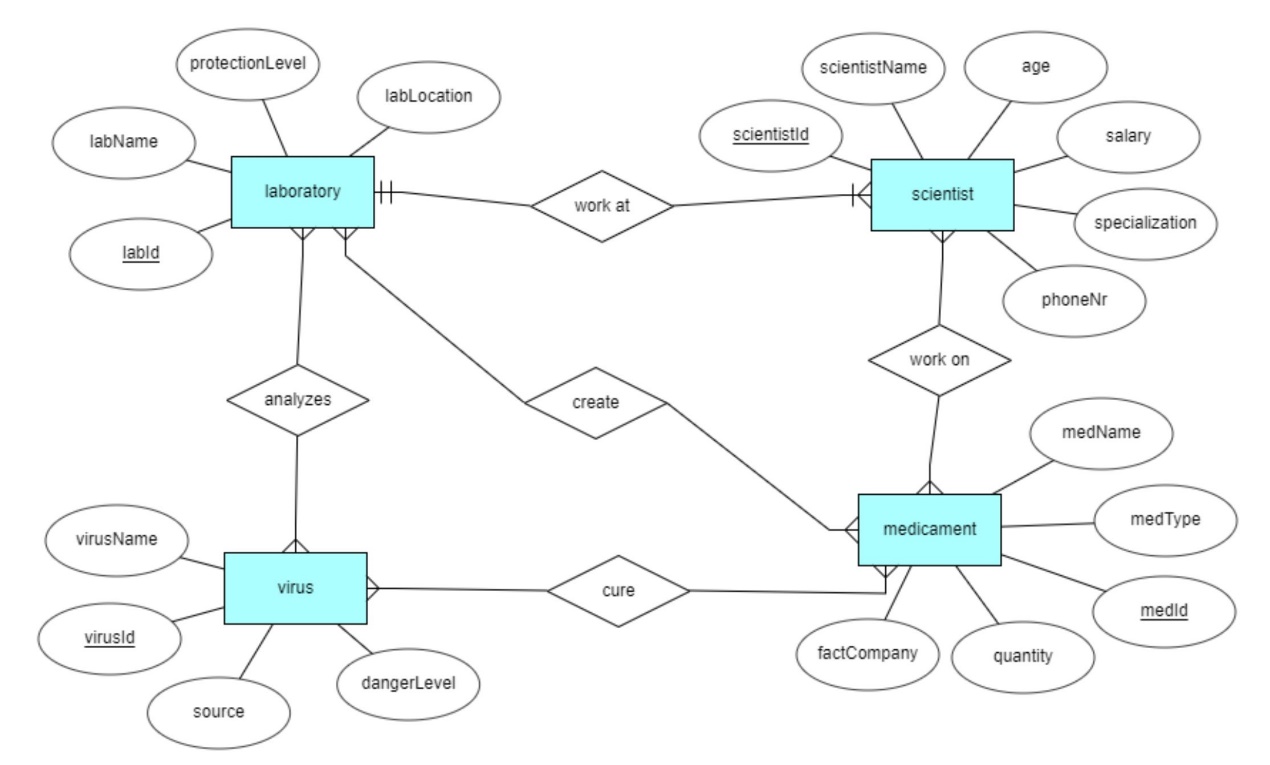
# הפרויקט שלנו



לפרויקט זה, נעבוד על בביס נתונים של אגף המדע הארצי.

מאגר מידע של אגף זו יכול לשמש בעיקר על ידי משרד הבריאות (למידע על תרופות, נגיפים וכו '), משרד הכלכלה (מידע על שכר, מחירים וכו') ואחרים (אזרחים, waze ,ארגונים ..).

**תרשים ERD**

****כפי שהזכרנו, באגף שלנו, ישנן 4 ישויות : מעבדות, מדענים, וירוסים, תרופות. בשלב הראשון יצרנו תרשים ERD שיתאר את הקשרים בין הישויות הללו ואת התכונות שלהן.

**תיאור הישויות והקשרים**

**ישויות**

* Laboratory – ישות זאת אחראית על כל מעבדות הנמצאים במערכת.
* labId – מספר מזהה של המעבדה (PK)
* labName – שם המעבדה
* protectionLevel – רמת הגנה של המעבדה
* labLocation – מיקום המעבדה
* Scientist – ישות זאת אחראית על כל המדענים הנמצאים במערכת.
* scientistId – מספר מזהה של המדען (PK)
* scientistName – שם המדען
* age – גיל המדען
* salary – שכר המדען
* specialization – התמחויות
* phoneNr – מספר טלפון של המדען
* labId – מספר מזהה של המעבדה (FK)
* Virus – ישות זאת אחראית על כל הוירוסים הנמצאים במערכת.
* virusId – מספר מזהה של הוירוס (PK)
* virusName – שם הוירוס
* source – מקורו
* dangerLevel – רמת סכנת
* Medicament – ישות זאת אחראית על כל התרופות הנמצאים במערכת.
* medId – מספר מזהה של תרופה (PK)
* medName– שם התרופה
* medType – סוג התרופה (חיסון או תרופה)
* quantity – כמות
* factCompany – חברת יצור
* Analyzes – ישות זאת מגדירה אילו מדענים מנתחים אילו וירוסים.
* labId – מספר מזהה של המעבדה (FK)
* virusId – מספר מזהה של הוירוס (FK)
* Creates – ישות זאת מגדירה אילו מעבדות יוצרים אילו תרופות.
* labId – מספר מזהה של המעבדה (FK)
* medId – מספר מזהה של תרופה (FK)
* Cure – ישות זאת מגדירה אילו תרופות מרפאות אילו וירוסים.
* medId – מספר מזהה של תרופה (FK)
* virusId – מספר מזהה של הוירוס (FK)
* Work\_On – ישות זאת מגדירה אילו מדענים עובדים אילו תרופות.
* scientistId – מספר מזהה של המדען (FK)
* medId – מספר מזהה של תרופה (FK)

**קשרים**

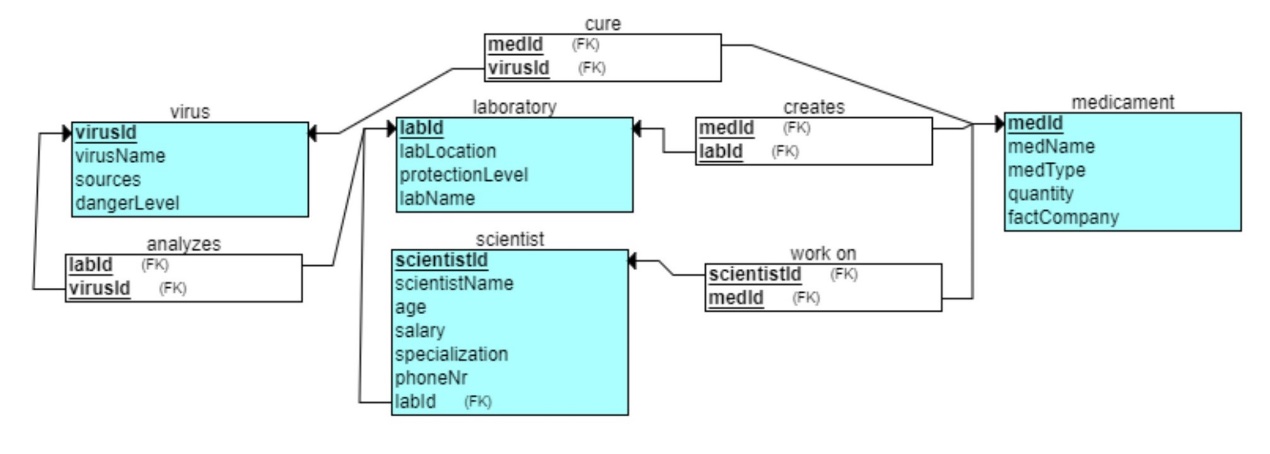
* work at – הקשר בין laboratory לבין scientist. הקשר הוא M:1 משום שמדען אחד יכול לעבוד רק במעבדה אחת, ובמעבדה אחת יכול לעבוד כמה מדענים.
* work on – הקשר בין scientist לבין medicament. הקשר הוא M:M משום שמדען אחד יכול לעבוד על כמה תרופות, ותרופה אחת יכולה להיות מעובד על ידי כמה מדענים..
* analyzes – הקשר בין laboratory לבין virus. הקשר הוא M:M משום שמעבדה אחת יכולה לנתח כמה וירוסים, ווירוס אחד יכול להיות מנותח בכמה מעבדות.
* create – הקשר בין laboratory לבין medicament. הקשר הוא M:M משום שמעבדה אחת יכולה ליצור כמה תרופות, ותרופה אחת יכולה להיות מיוצרת בכמה מעבדות..
* cure – הקשר בין virus לביןmedicament . הקשר הוא M:M משום שוירוס אחד יכול להיות מטופל עם כמה תרופות, ותרופה אחת יכולה לטפל כמה וירוסים.

**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**

על פי תרשים ה- ERD ועל ידי הבנת הקשרים בין הישויות, יצרנו תרשים DSD עבור האגף שלנו במערכת: אגף המדע.



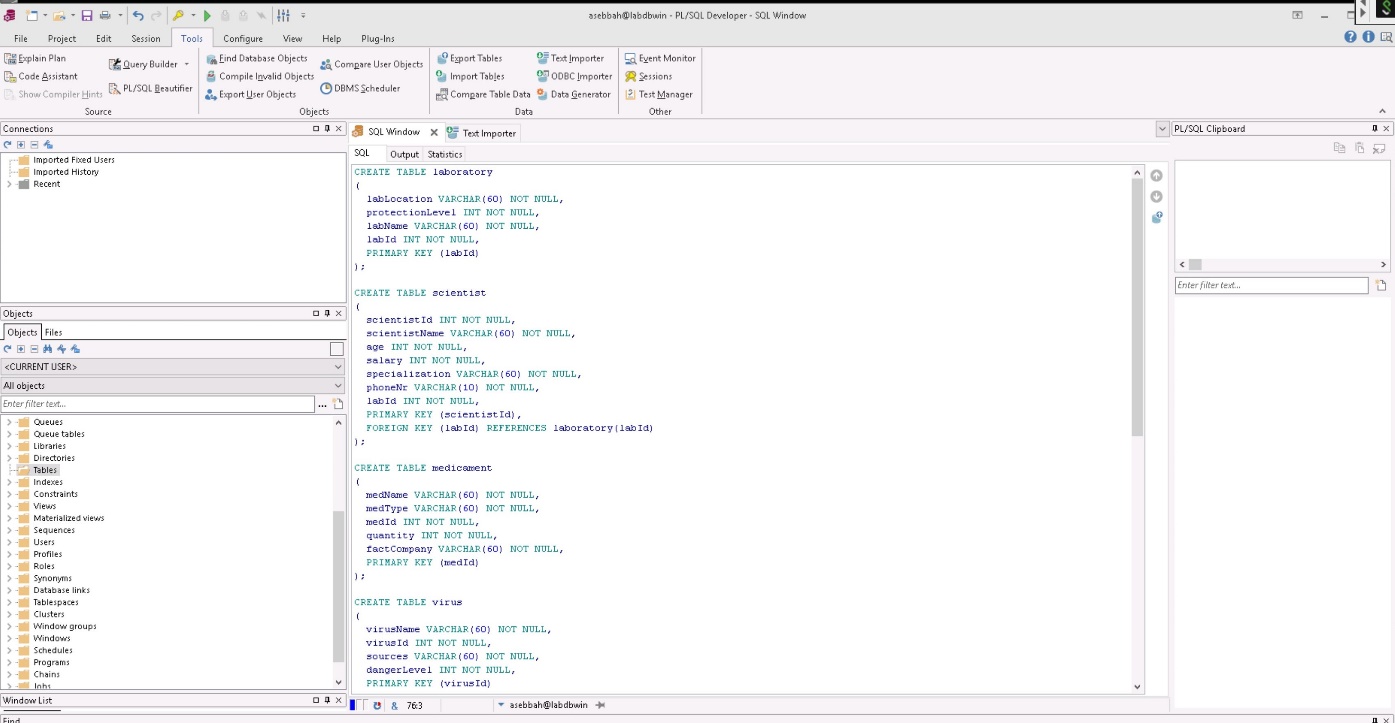
## 

## 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.

הקוד נמצא בנספח בסוף הדוח.



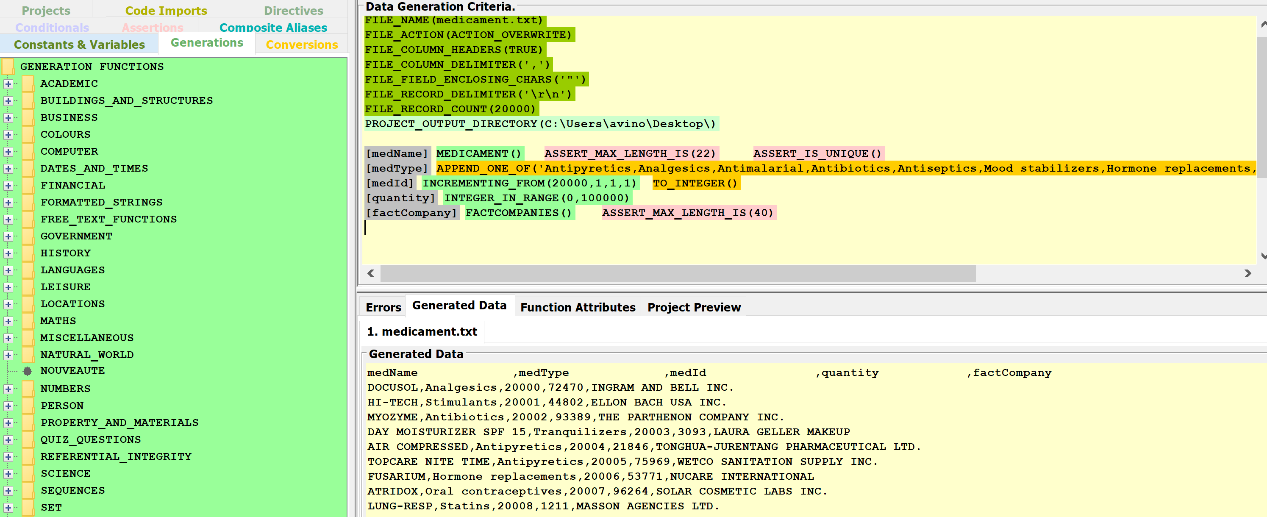
## הכנסת נתונים אכלוס הטבלאות

על מנת לאכלס את הטבלאות שיצרנו בנתונים, השתמשנו ביבוא מידע מקובץ טקסט לתוך הטבלאות בבסיס הנתונים. יצרנו קובץ טקסט ובעזרת תכנה בשם GenerateDataDSL090 עם בסיס נתונים עצום, הוא בחינם לחלוטין ומאפשר לך ליצור מיליוני נתונים ללא בעיות.

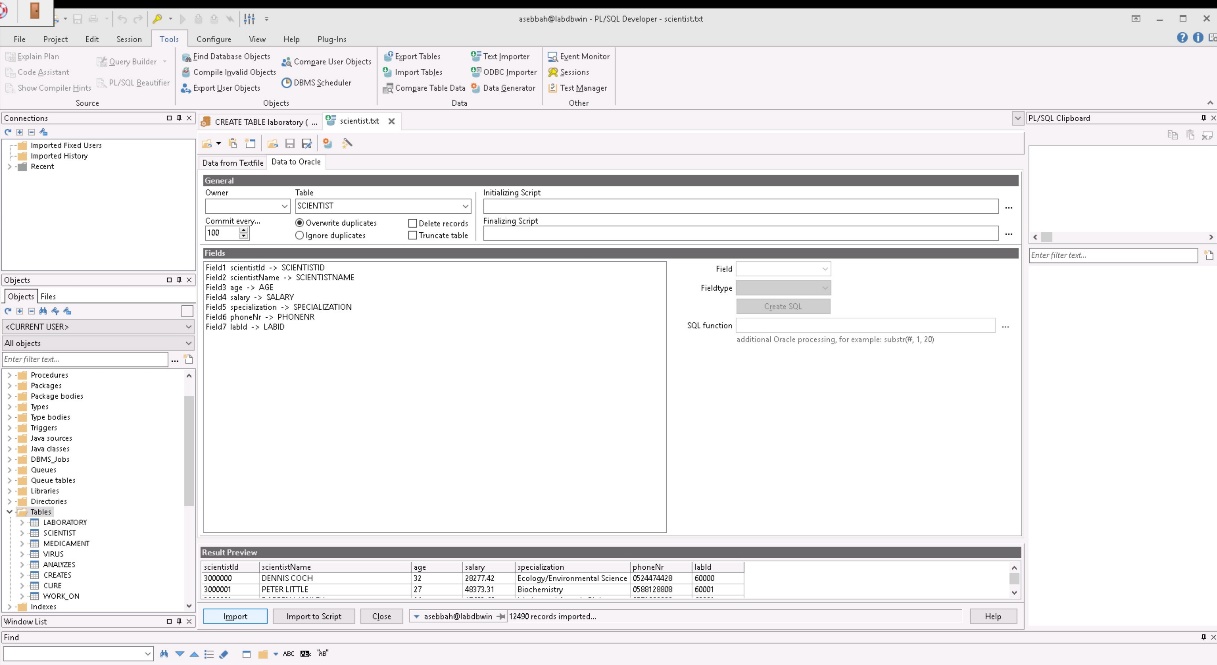
<- <https://sourceforge.net/projects/generatedata/files/>

מה שמעניין הוא שהתוכנה נותנת אלפי פונקציות להגדרת תצורה של הבסיס הנתונים !!

לדוגמא : בהוספת פקודה assert\_is\_unique() לעמודה מסייומת אני מחייב את העמודה להשתמש רק פעם אחד בנתון הזה.. (הקוד לכל הישויות נמצא בנספח השני)



אחר כך השתמשנו בtext importer בתוכנת ה- PL/SQL כדי להכניס את הנתונים לטבלאות. כך לדוגמא, הכנסנו נתונים לתוך הטבלה : SCIENTIST



## 

## שאילתות SQL

לאחר שיצרנו את בסיס הנתונים והכנסנו לתוכו מידע, כתבנו כמה שאילתות מעניינות על מנת לתשאל אותו. (הקוד בנספח השלישי)

1. מעוניין בתקציב הנוכחי, משרד הכלכלה רוצה לדעת כמה היא משלמת למחלקת המדע להתמודד עם הנגיף הנוכחי (קוביד-19).
   * לכן כתבנו שאילתה שמחשב את הסכום שכרם של כל המדענים העובדים על תרופות המרפאות את הנגיף/וירוס המדובר.
2. לאחר תלונות של כמה בתי חולים על מלאי התרופות למגפה , משרד הבריאות רוצה לדעת אילו תרופות הנמצאות במלאי יש כמות קריטי כדי להיות מסוגלות להצטייד במלאי.

* לכן כתבנו שאילתה המדפיסה את כל התרופות הנמצאות מתחת לסף הקריטי.

1. איחוד המדענים העולמי רוצה לקבל רשימת מדענים העובדים במעבדה מסוימת כדי להבטיח את תפקודה התקין של המערכת .

* לכן כתבנו שאילתה המחזירה רשימת שמות המדענים העובדים שם.

1. אחרי כמה בעיות ביטחונית בכמה מעבדות, משרד הבריאות רוצה לעצור את המעבדות העובדות על וירוסים מסוכנים, למרות הרמת ההגנה הנמוכה שלהן.

* לכן כתבנו שאילתה המחזירה טבלה של שמות המעבדות עם רמת הגנה נמוכה ושעובדים על וירוסים מסוכנים.

1. הראש ממשלה רוצה להודות לכל המדענים שעובדים על הנגיף שעובר כרגע על המדינה.

* לכן כתבנו שאילתה המחסירה טבלה של שמות המדענים העובדים על וירוס מסוים.

1. המדענים רוצים לדעת איזה תרופות מרפאות הכמות ההכי גדולה של וירוסים כדי לדעת באיזה כיוון להמשיך את העבודה שלהם.

* לכן כתבנו שאילתה המחזירה שמות התרופות עם מספר הוירוסים שכל אחד מרפא.

1. העיתון הארצי רוצה להרגיע את האוכלוסייה על ידי כתיבה בעמוד הראשון שמות הרופאים שעובדים בתחום החיסונים ושעובדים על וירוסים מסוכנים לעיתון למחר.

* לכן כתבנו שאילתה המחזירה טבלה עם שמות המדענים שעובדים על וירוסים מסוכנים ושהתחום שלהם החיסונים בלי הכפילויות במקרה שהוא עובד על כמה וירוסים מסוכנים.

1. כדי להגן על התקדמותה המדעית במדינה, שירות המודיעין הלאומי רוצה לקבל רשימה של כל המדענים העובדים במעבדות לאומיות כיום כדי להבטיח את ביטחונם.

* לשם כך כתבנו שאילתה המחזירה טבלה עם שמות המדענים העובדים במעבדות שנמצאות בתוך המדינה.

## 

## 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 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.

לכן יצרנו אינדקסים (הקוד בנספח השלישי) שמקצרים את תהליך ביצוע השאילתות.

חשוב לציין כמה דברים :

* מאוד קשה לקבל זמני ריצה מדויקים כי זה תלוי במהירות המחשב, ואם במקביל המחשב או התוכנה מבצעים פעולות אחרות, יכולות להיות לכך השלכות של מספר שניות !!

הזמני ריצה המוצגים הן תוצאות בתנאים הטובים ביותר (לפני ואחרי האינדקסים), אך זה עשוי להשתנות.

* בכדי להבהיר למרצה, השארנו את חלון אובייקט התוכנה, כדי להראות מתי השתמשנו באינדקסים ומתי לא השתמשנו (לפני ואחרי).

כאשר משתמשים באינדקס הוא מודגש.

תוכן אינדקסים :

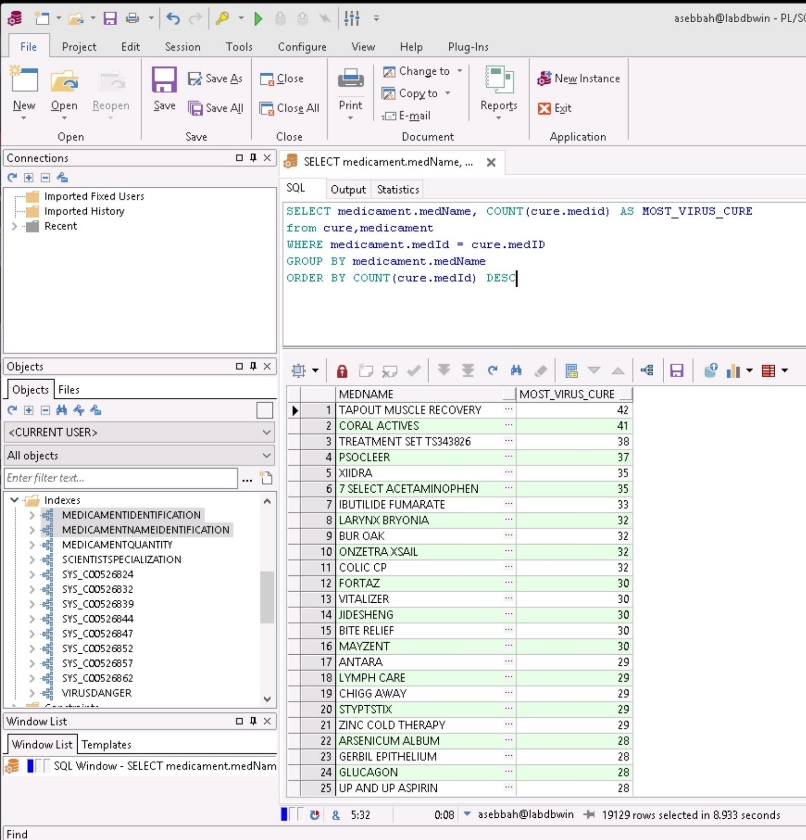
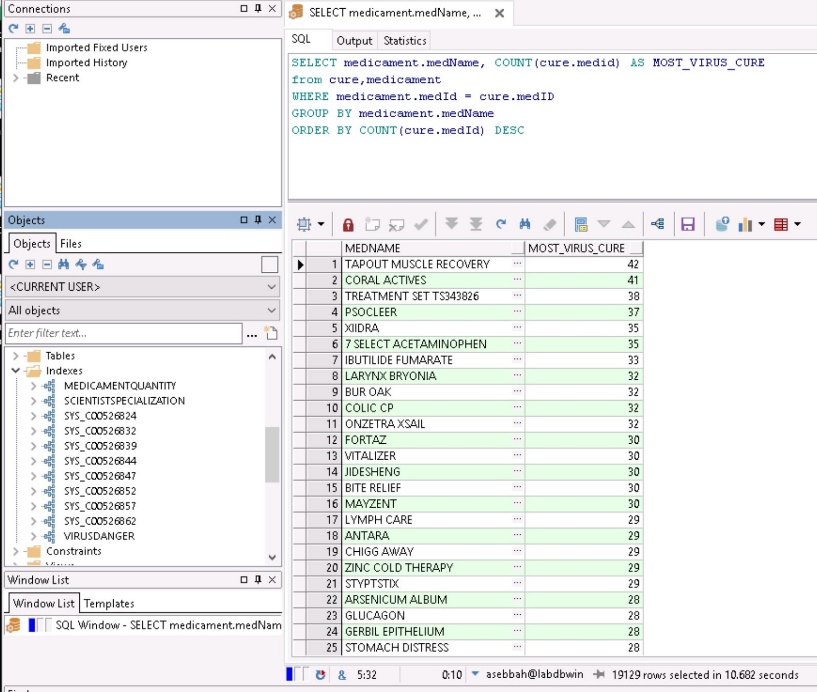
1. אינדקס הראשון : על העמודה של (medID) הנמצאת בתוך טבלת [cure] ועל העמודה (medName) הנמצאת בתוך טבלת [medicament].
2. אינדקס השני : על התמחויות של מדענים ויותר מדויק על התמחות’Immunology’ .
3. אינדקס השלישי : על עמודת(quantity) הנמצאת בטבלת [medicament]

(קוד האינדקסים בנספח השלישי בסוף הדוח)

1. שאלנו שאילתה המחזירה טבלה של שמות התרופות עם מספר הווירוסים שכל אחד מרפא.

* האינדקס שיצרנו היה על העמודה של (medID)הנמצאת בתוך טבלת [cure]ועל העמודה (medName) הנמצאת בתוך טבלת .[medicament]

ולכן כשחיפשנו בשאילתה את שמות/מספר זהות של כל התרופות החוזרות על עצמם בתוך הטבלת [cure] , שזה בעצם אומר כל פעם שהוא מופיע הוא מרפא עוד וירוס אחד, היה מהיר יותר לקבל את התוצאה.



לפני

אחרי

אינדקס הראשון : לפני הגדרת האינדקס הייתה לנו זמן ביצוע של 10.682s, לאחר הגדרתו קיבלנו זמן ביצוע של 8.933s.

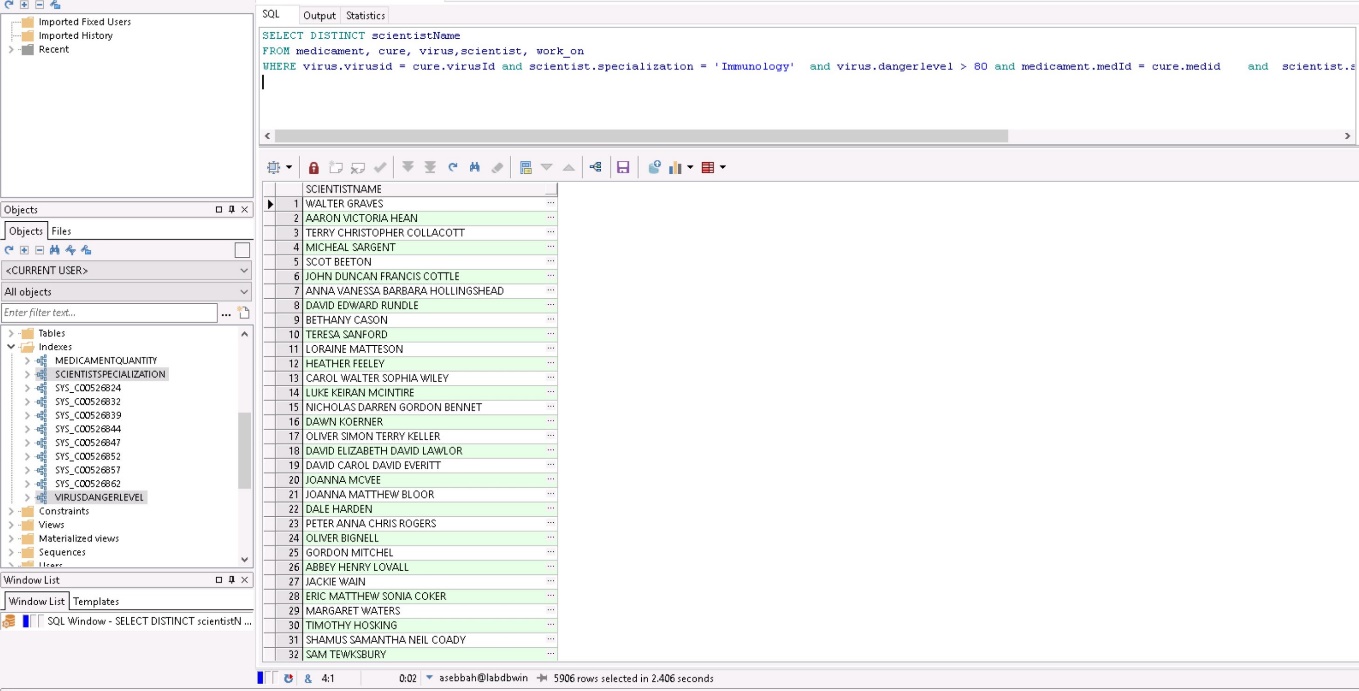
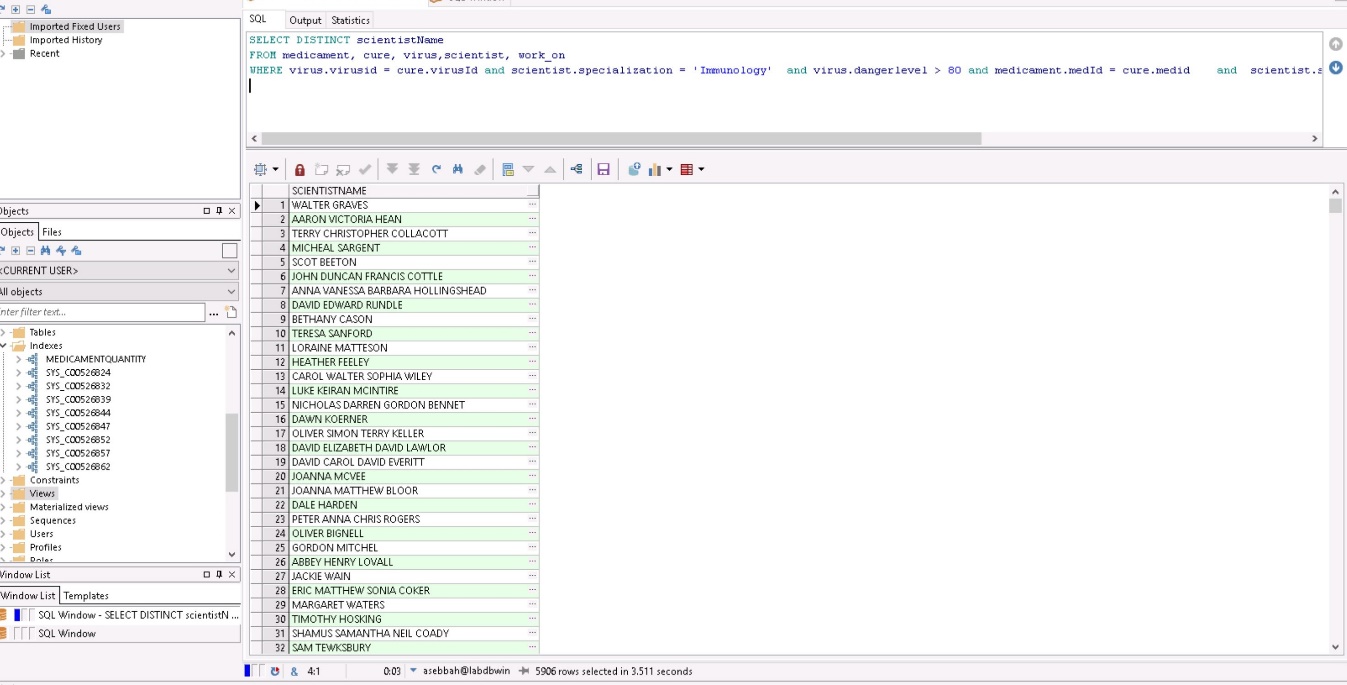
בזכות האינדקס, אנו מרוויחים 2 שניות וחצי פחות על המהירות ! (2.5s)

1. שאלנו שאילתה המחזירה טבלה עם שמות המדענים שעובדים על וירוסים מסוכנים ושהתחום שלהם החיסונים בלי הכפילויות במקרה שהוא עובד על כמה וירוסים מסוכנים.

* האינדקס שיצרנו היה על התמחויות של מדענים ויותר מדויק על התמחות’Immunology’ , ולכן כשחיפשנו בשאילתה את שמות הרופאים שהם מומחים בחיסונים, היה מהיר יותר לקבל את התוצאה כי ההסתברות לקבל שם של רופא ירד ל1/16.

לפני

אחרי



אינדקס השני : לפני הגדרת האינדקס הייתה לנו זמן ביצוע של 03.511s, לאחר הגדרתו קיבלנו זמן ביצוע של 02.406s.

בזכות האינדקס, אנו מרוויחים שנייה אחת פחות על המהירות ! (1.1s)

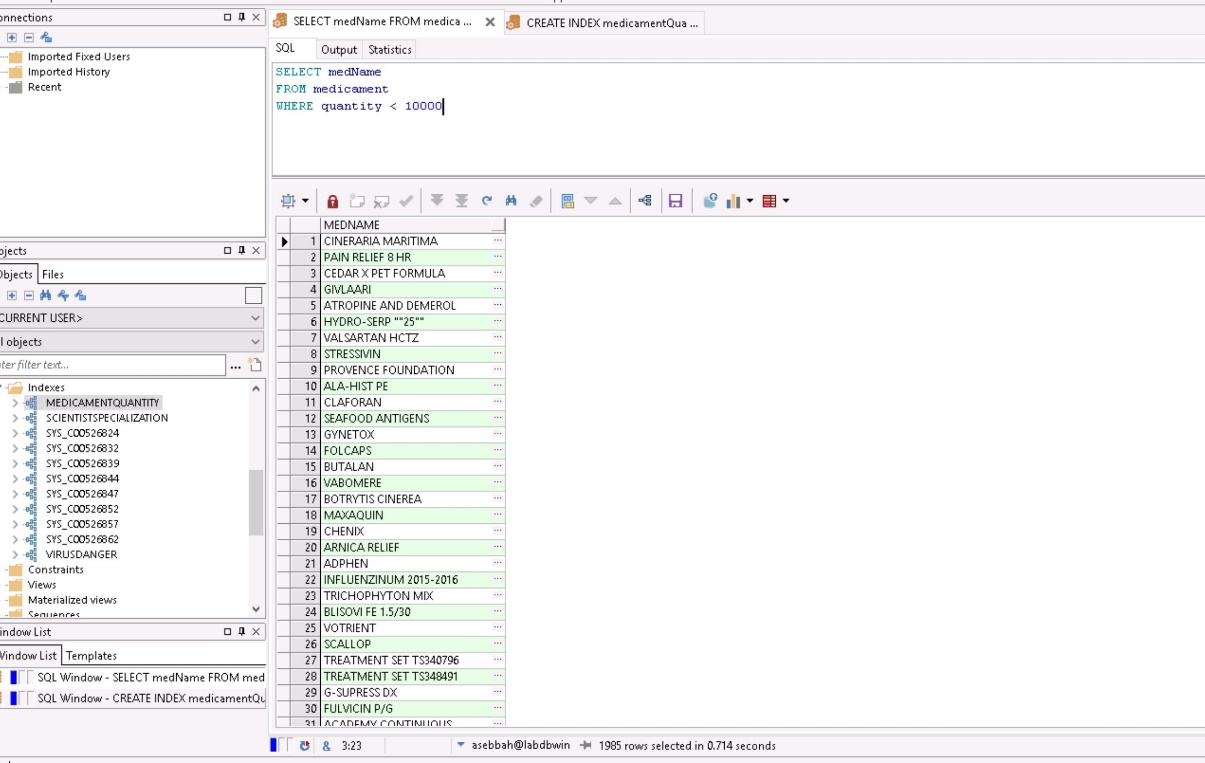
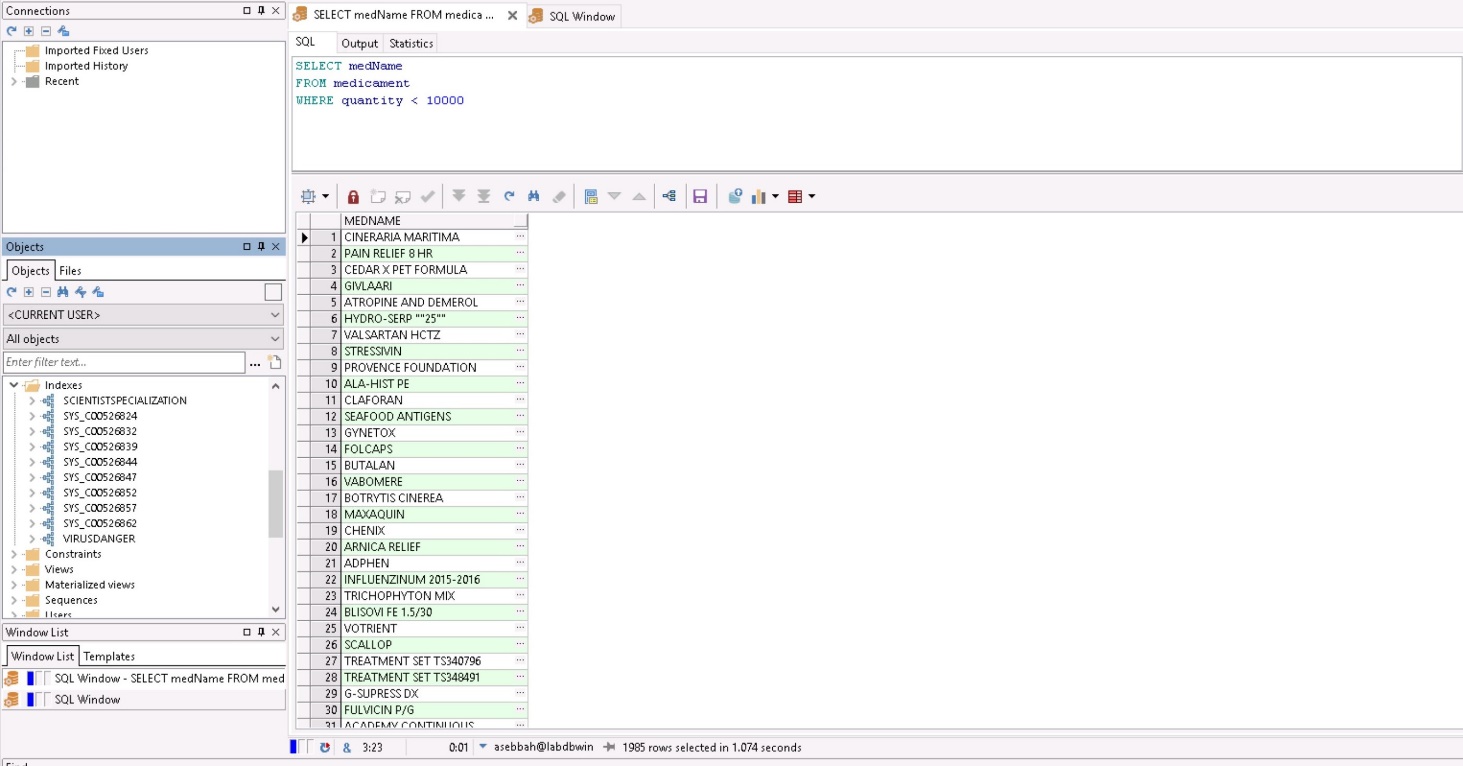
1. שאלנו שאילתה המדפיסה את כל התרופות שהכמות שלהם עברה סף הקריטי..

* האינדקס שיצרנו היה על עמודת(quantity) הנמצאת בטבלת [medicament]

(quantity) הוא מספר רנדומלי בין 0 ל-100000, לכן מבחינה הסתברותית יש לנו פחות quantity שנמצאות מתחת לסף הקריטי ולכן זמן החיפוש היה מעט קצר יותר.

לפני

אחרי



אינדקס השלישי : לפני הגדרת האינדקס הייתה לנו זמן ביצוע של 01.074s, לאחר הגדרתו קיבלנו זמן ביצוע של 00.714s.

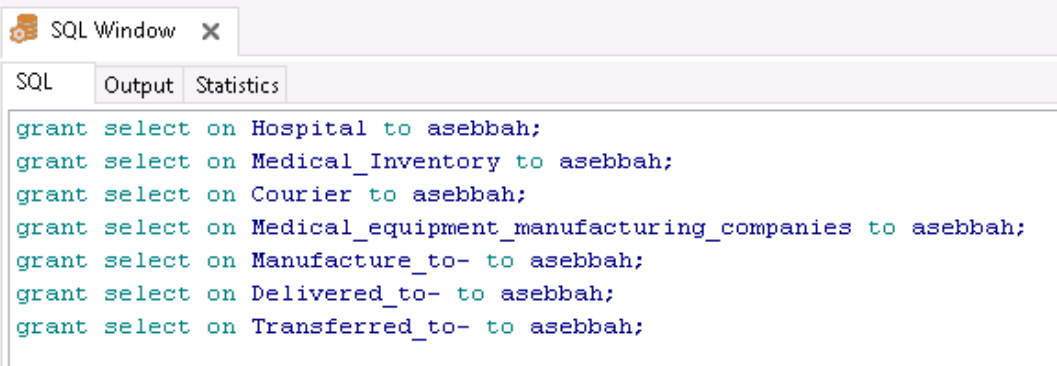
בזכות האינדקס, אנו מרוויחים חצי שנייה פחות על המהירות ! (0.5s)

## הרשאות

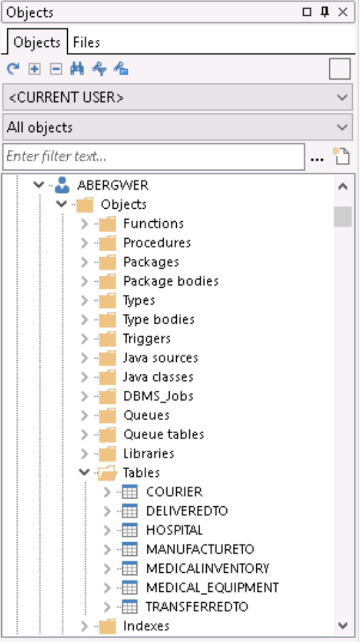
כדי שנוכל לעבוד ככיתה שלמה המפתחת בסיס נתונים כאשר כל קבוצה בונה טבלאות שונות לבסיס הנתונים ואנו צריכים לגשת לטבלאות שלהם כדי לקחת מידע אנו צריכים לקבל מיוצרי הטבלאות הרשאה כדי שנוכל להשתמש בטבלאות שלהם.

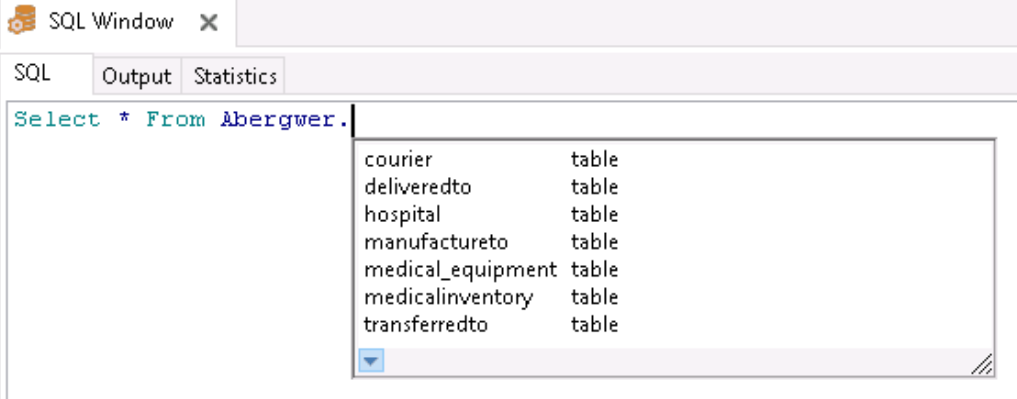
לקחנו מאשר ורפאל את הישויות.. הם עבדו על אגף המלאי שעוסק על חברות ציוד רפואי, מלאי ציוד רפואי במדינה, משלוח, ובית חולים.

נשים לב שהתרופות הן חלק מהציוד רפואי שלהם ואצלינו תרופות נמצאים בישות המתאים. עבדנו ביחד כדי לא להיווצר בעייות התנגשויות בינינו..

ואז הם אישרו לנו עם הפקודות הנ"ל להשתמש בישויות/טבלאות שלהם..

וכפי שניתן לראות, אנחנו יכולים עכשיו לגשת לכל הטבלאות שלהם :



 ונוכל לכתוב פקודות כאלה :

## שאילתות על איחוד אגפים

לאחר שקיבלנו הרשאה מהזוג השני, כתבנו כמה שאילתות מעניינות על מנת לתשאל כמה אגפים ביחד. (קוד שאילתות ותמונות של תוצאות בנספח הרביעי)

1. המדינה תהיה מעוניינת לבצע חוזים חדשים עם חברות יצור ציוד,כדי לחדש את מלאי לתרופות הנמצאות בכמות קריטי.
   * לכן כתבנו שאילתה המדפיסה שמות חברות יצור ציוד רפואי העובדים על תרופות הנמצאות בכמות קריטי.
2. סינדיקט המעבדות הלאומית מעוניין להשתמש בשירות חברות משלוחים מקומי כדי להקל למדענים ולמעבדות על שיתוף מסמכים ודגימות וירוסים / תרופות ביניהם .

* לכן כתבנו שאילתה המדפיסה שמות מעבדות ושמות חברות משלוחים שנמצא באותו מיקום.

1. כמה אזרחים שרצו להגיע לבית חולים בזכות אפליקצית Waze©, היגעו ליעד הלא נכון, מכיוון שלכמה מעבדות במדינה יש שמות זהים לבתי חולים !!!

כתוצאה מכך ,משרד הבריאות רוצה בדחיפות לדעת אילו שמות בעייתיים בכדי לשנותם..

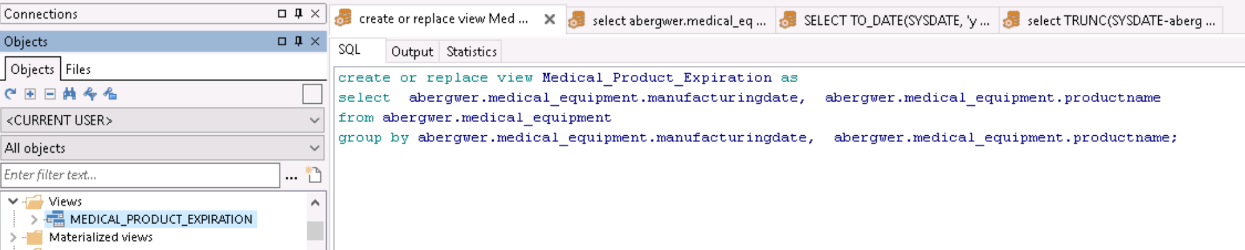
* לכן כתבנו שאילתה המחזירה רשימת שמות המעבדות שיש להם שם בעייתי (זהה לשם בית חולים) והכתובת שלהם בשביל עדכון האפליקציה.

## Views

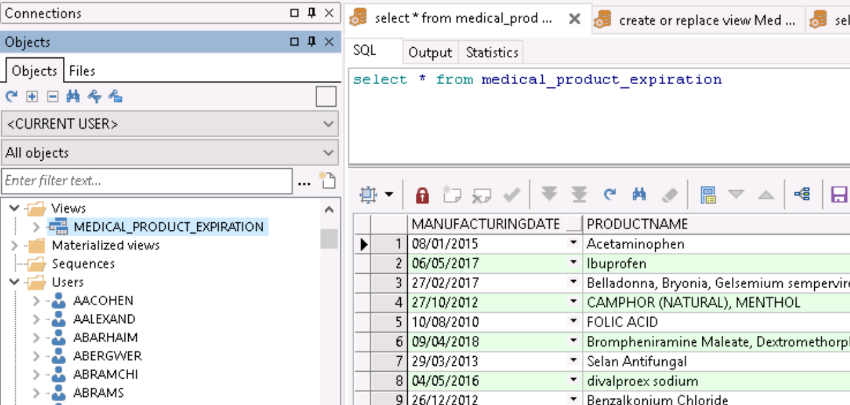
1. משרד הבריאות מעוניין לקבל טבלה שמתעדכנת כל העת, על ציודי רפואיים שפג תוקפם (על פי החוק מעל 20 שנה, לפי התאריך הנוכחי), בכדי להוציא אותם מהשוק ולהחליפם.

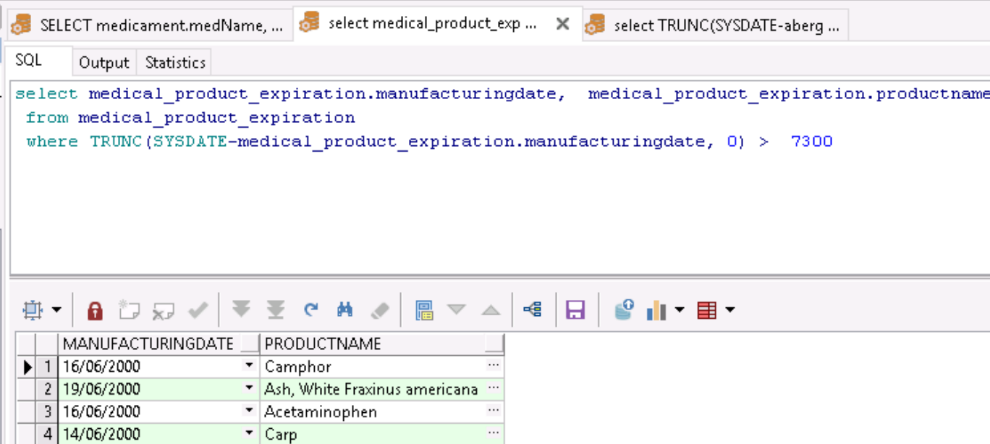
יצרנו view עם עמודות ProductName ו-ManufacturingDate כיוון שמשרד הבריאות משתמש כל הזמן בשתי העמודות הללו כדי לקבל את המידע עדכני .

ולכן יעיל יותר לעבוד עם שתי עמודות אלו באופן נפרד, בלי התייחסות לטבלאות המלאות. (הקוד המלא נמצא בנספח החמישי)

יצירת ה-view :

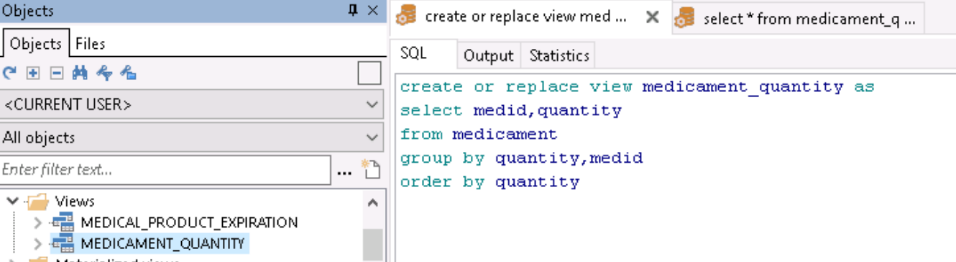
ואז משתמשים בטבלה :

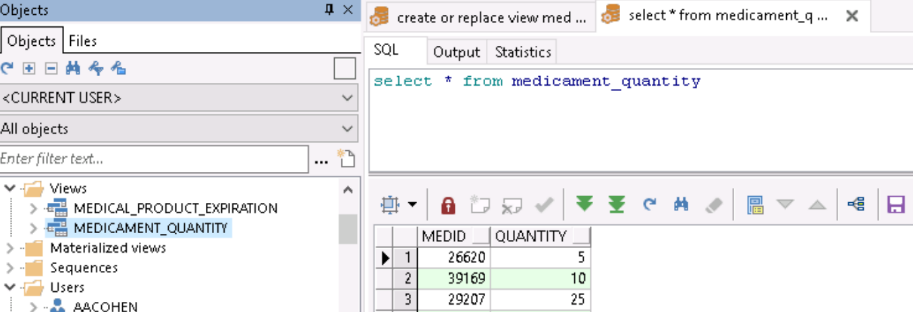


משרד הבריאות יוכל כעת לגשת ביתר קלות לטבלאות, הם יוכלו לדעת כל יום אילו תרופות פג תוקפן :

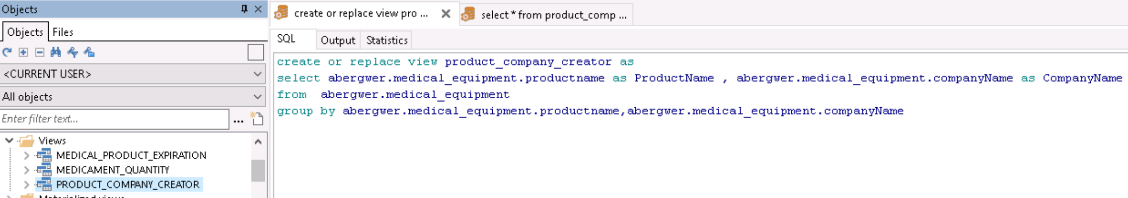
1. יצרנו view עם עמודות medId וquantity כיוון שרבות מהשאילתות שלנו מתייחסות לעמודות אלו ולכן יעיל יותר לעבוד עם שתי עמודות אלו באופן נפרד, בלי התייחסות לטבלאות המלאות. (הקוד עצמו נמצא בנספח החמישי).

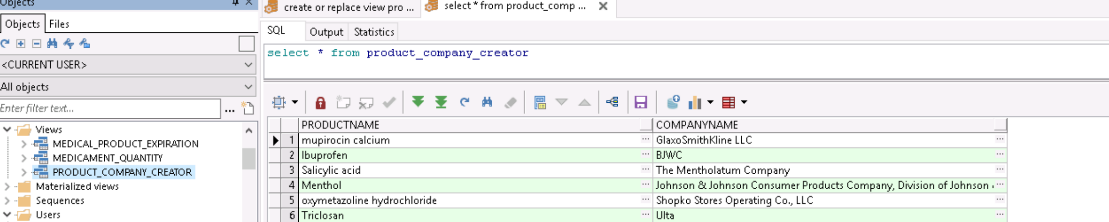
הוספנו מיון לפי כמות התרופות כך שהטבלה תהיה קלה יותר לשימוש.

יצירת ה-view :

 ואז משתמשים בטבלה :

1. יצרנו view עם עמודות CompanyName וProductName כיוון שרבות מהשאילתות שלנו וגם כמה פונקציות שנראה בהמשך מתייחסות לעמודות אלו ולכן יעיל יותר לעבוד עם שתי עמודות אלו באופן נפרד, בלי התייחסות לטבלאות המלאות. (הקוד עצמו נמצא בנספח החמישי).

יצירת ה-view :

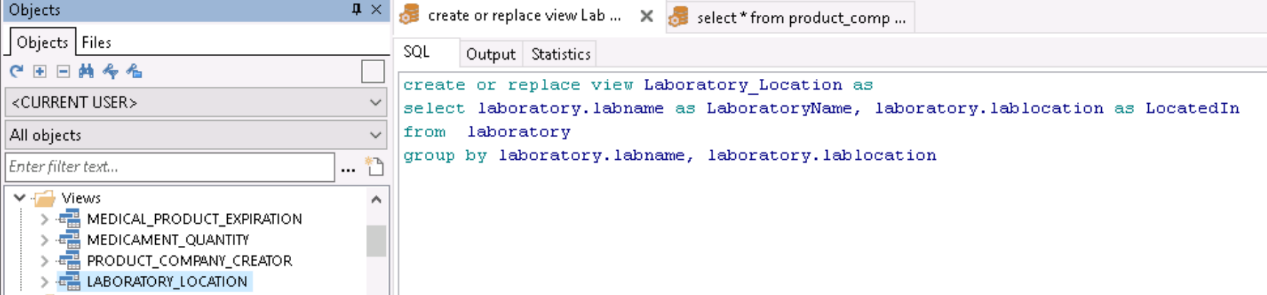
ואז משתמשים בטבלה :

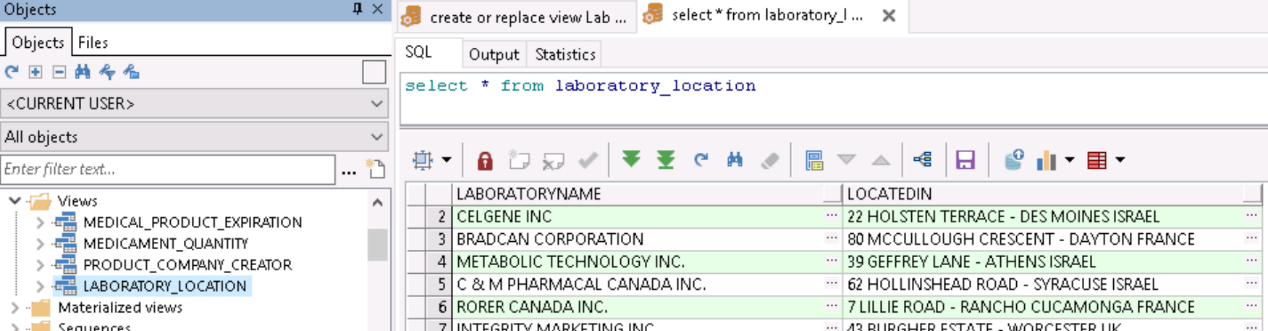
1. Waze זקוקה לטבלה המורכבת ממעבדות וכתובותיהם, על מנת שתוכל לעדכן כל העת את השירות שלהם..

יצרנו view עם עמודות labName ו-labLocation כיוון ש-Waze ישתמש כל הזמן בשתי העמודות הללו כדי לקבל את המידע עדכני .

ולכן יעיל יותר לעבוד עם שתי עמודות אלו באופן נפרד, בלי התייחסות לטבלאות המלאות. (הקוד המלא נמצא בנספח החמישי)

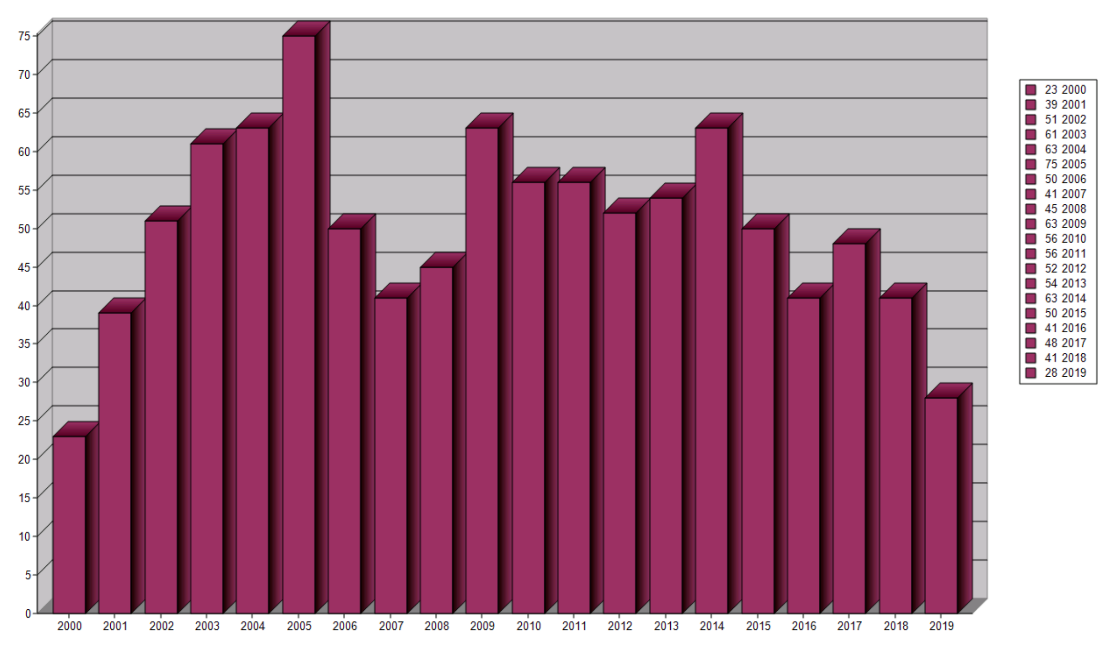
יצירת ה-view :

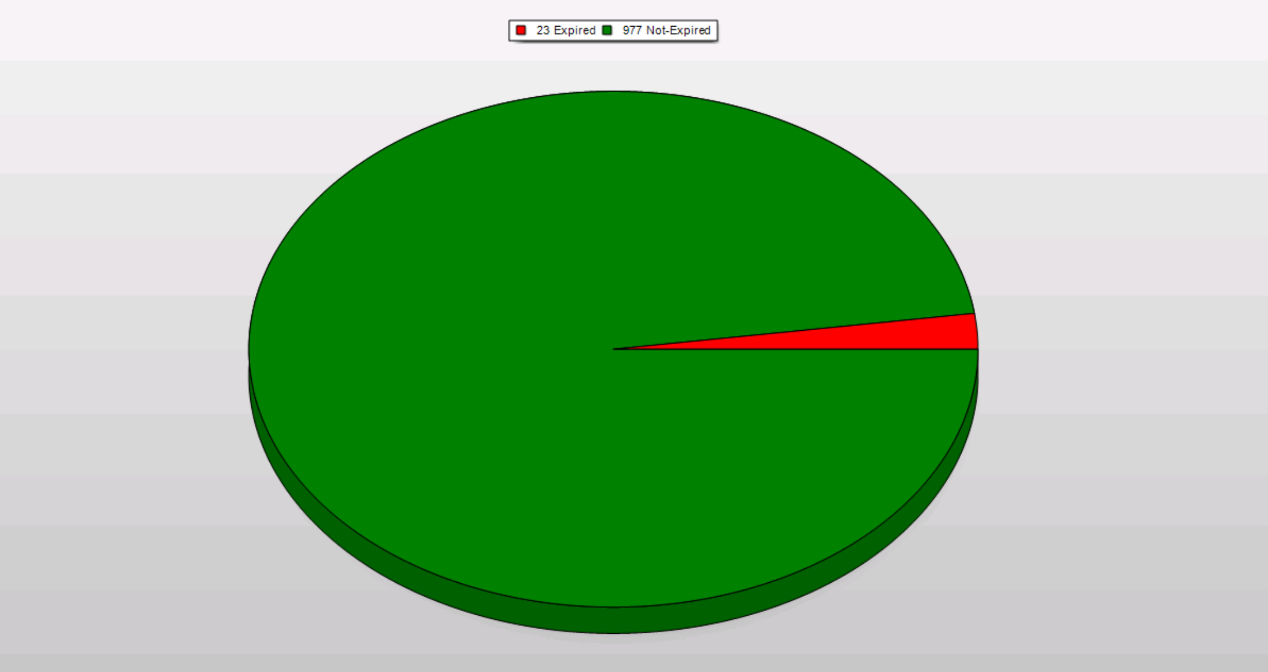


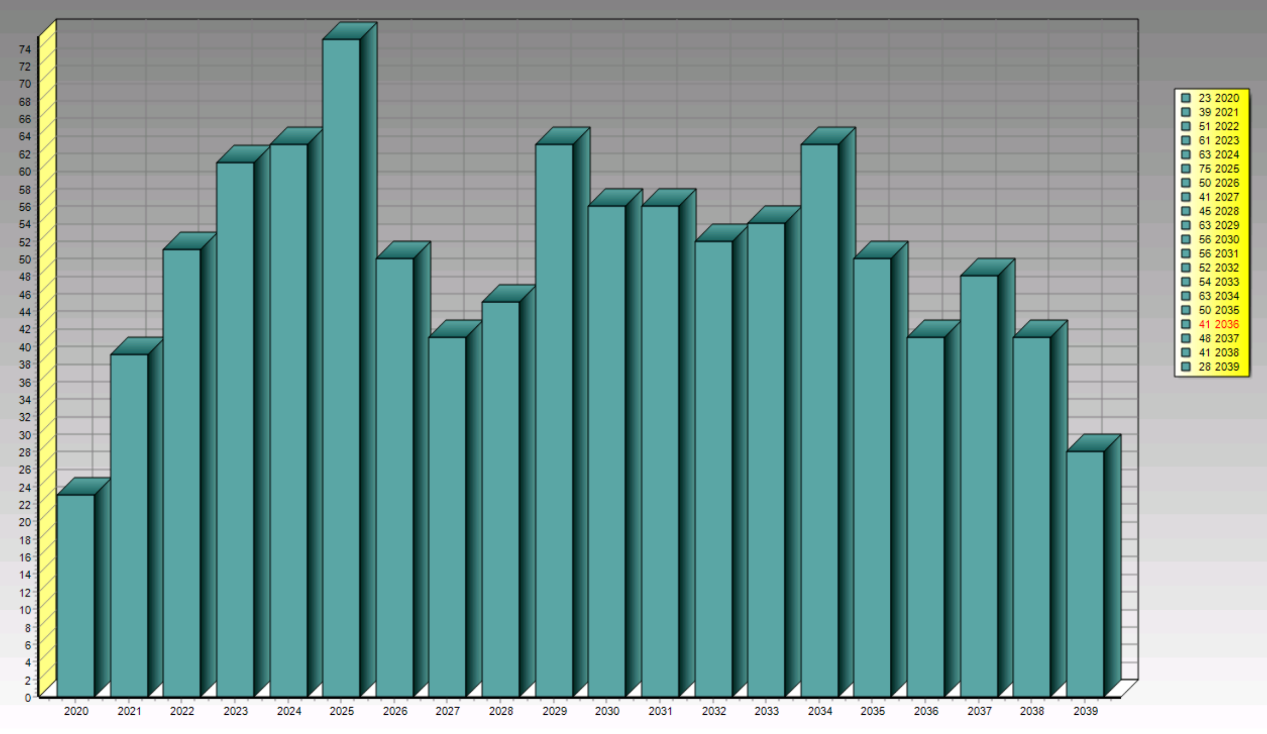
ואז משתמשים בטבלה :

## גרף

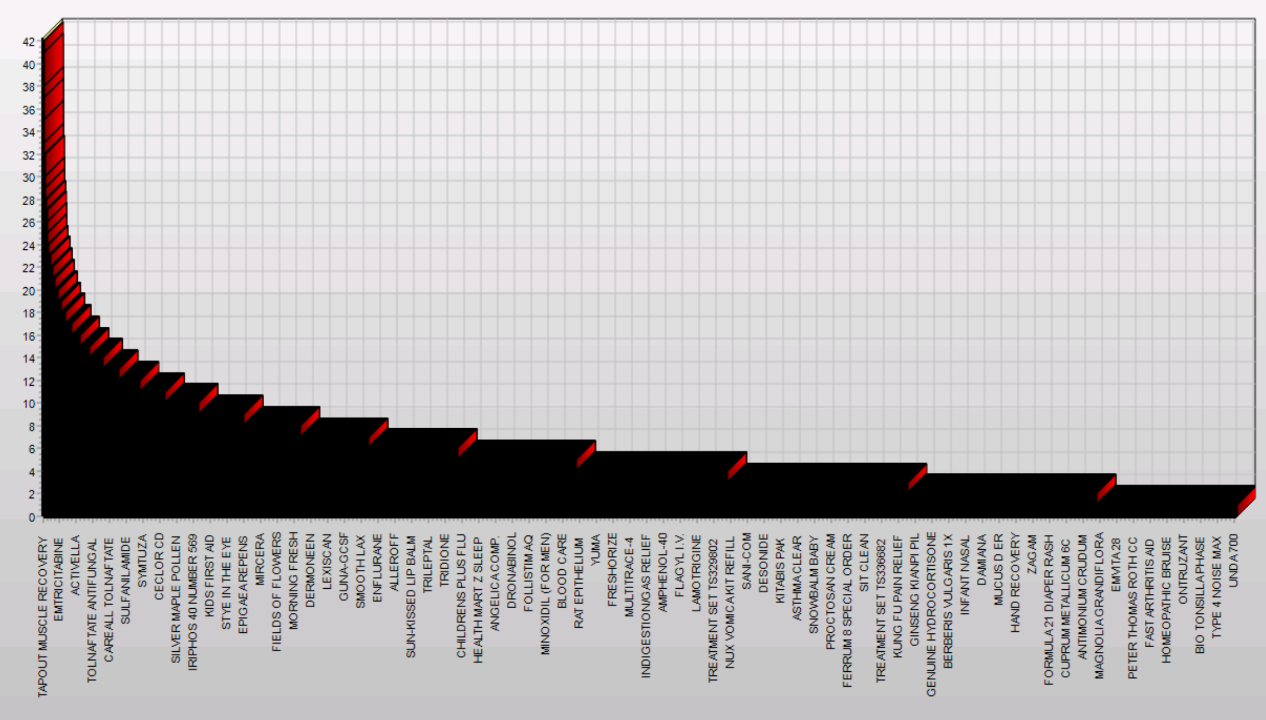
1. משרד הבריאות מעוניין אז לקבל תרשים של תאריכי יצירת המוצר לפי שנה על מנת לקבל סקירה של המוצרים העומדים לפוג.

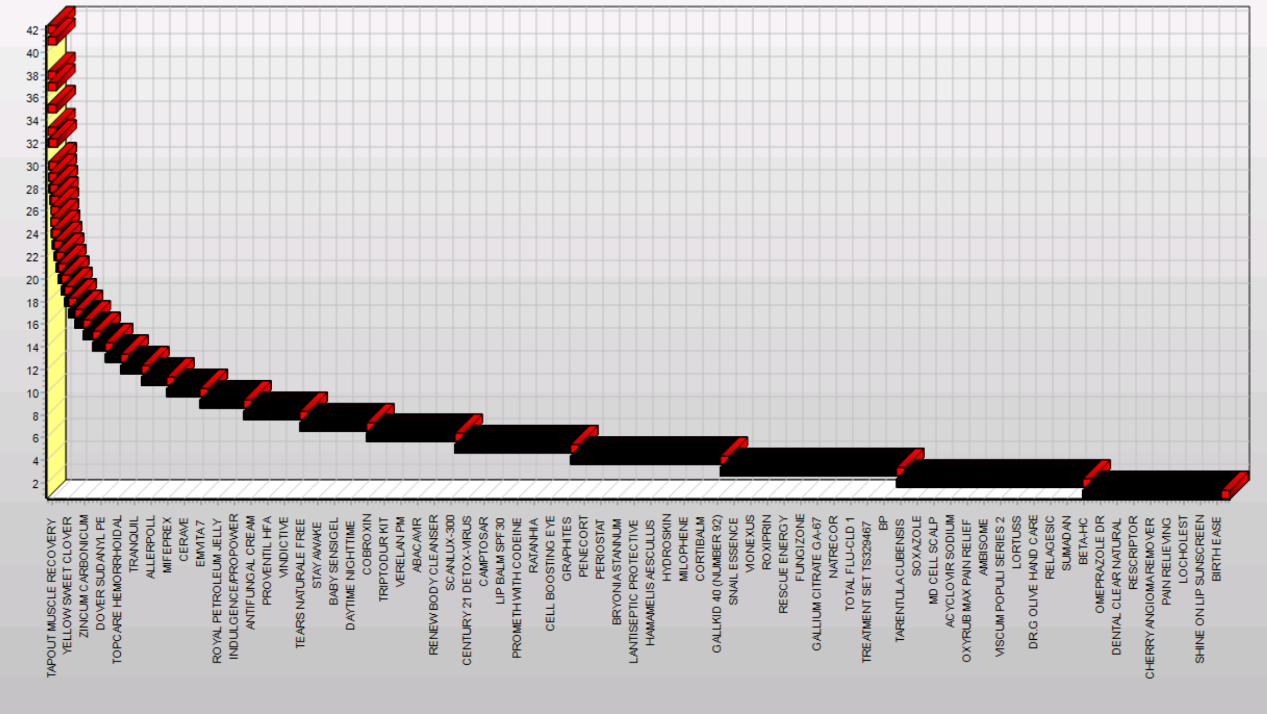
ולכו יצרנו שאילתה (קוד בנספח חמישי) ותרשים\גרף המייצגים את מספר המוצרים שנוצרו במהלך שנה מסוימת :

בצורה של תרשים עוגה : **(*Pie chart*)** (אדום מייצג כמות המוצרים שפג תוקפם)

משרד הבריאות יוכל גם להשתמש בגרף המייצג תאריכי תפוגה לפי שנה ומוצר )על מנת לקבל סקירה כללית של ההוצאות שיש לחדש את המלאי (:

1. מנכ"ל המעבדות הלאומיות רוצה לדעת עד כמה יעילות התרופות שנוצרו על ידי המדענים ביחס למספר הוירוסים.

וכפי שניתן לראות, זו לא תוצאה טובה עבור מדענים..

בתרשים זה אנו יכולים לראות ביתר קלות כיצד מרבית התרופות אינן יעילות כנגד וירוסים :

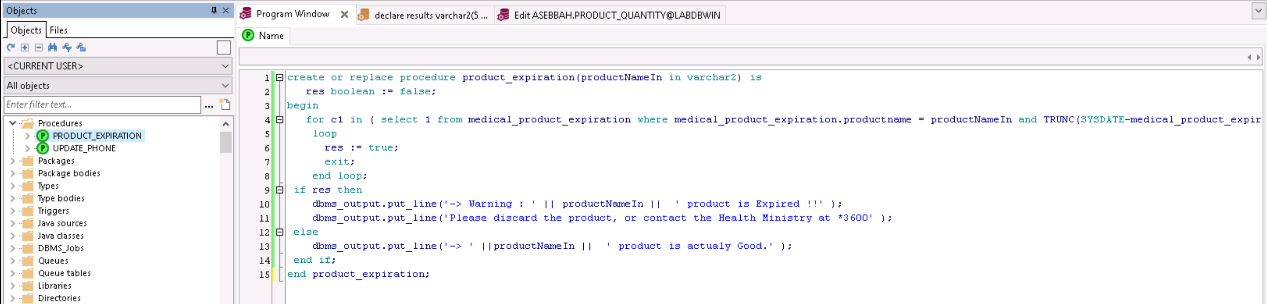
## פרוצדרות

1. במקרה של פקיעת מוצר, משרד הבריאות יסלק אותו מבתי מרקחת, בתי חולים וכו '.

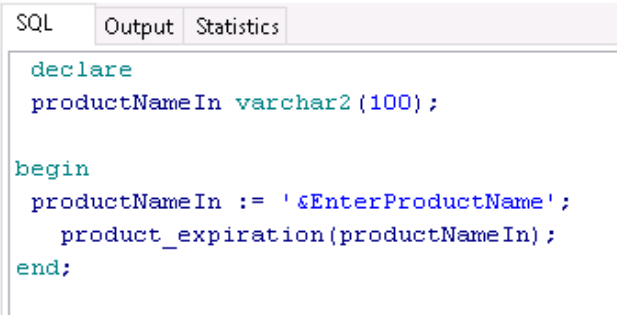
אך הם לא יוכלו לסלק את המוצרים שנמצאים אצל האזרחים, דבר שעלול להוות סכנה.

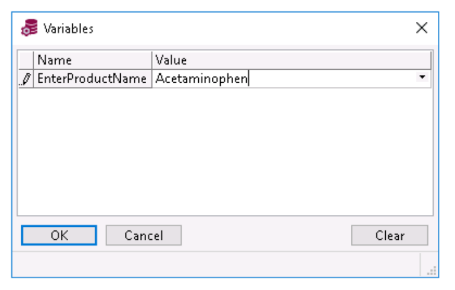
לכן משרד הבריאות החליט ליצור אפליקציה (פרוצדורה), המאפשרת לאזרחים לדעת אם פג תוקף של מוצר, על ידי הזנת שם המוצר. (הקוד המלא נמצא בניספח שישי)

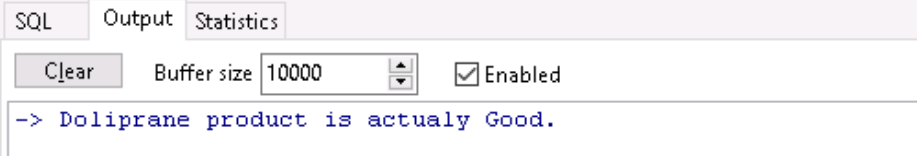
אנו יוצרים תחילה את הנוהל הפרוצדורה בשםproduct\_expiration  :



ואז נריץ את הפרוצדורה :

אנו מבקשים מהמשתמש להזין את שם המוצר שהוא רוצה לבדוק :



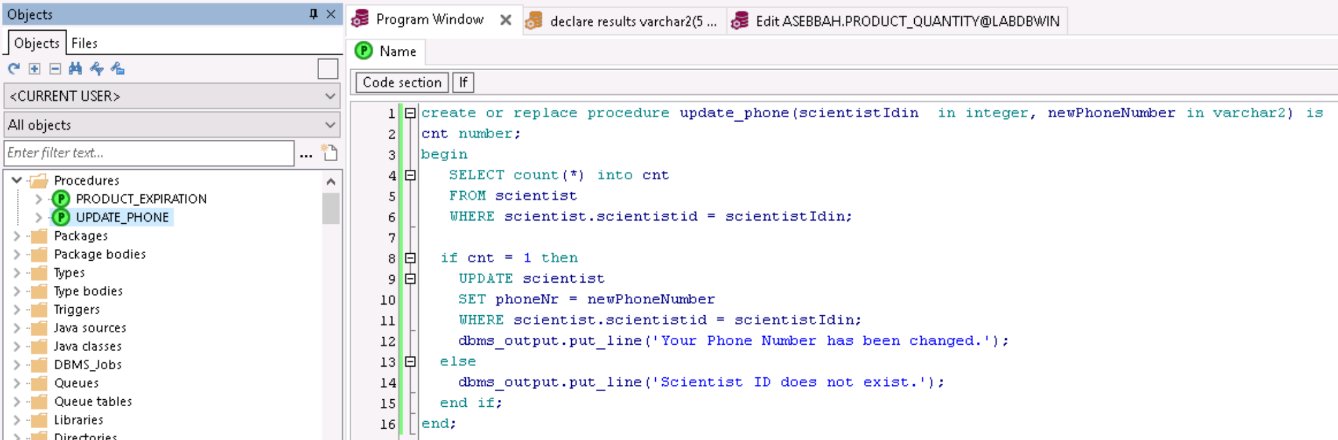
ואז אנו מציגים את תוצאות החיפוש למשתמש :

1. מזכירי המעבדות מתקשים יותר ויותר ליצור קשר עם מדענים מסוימים בגלל מספרי טלפון.

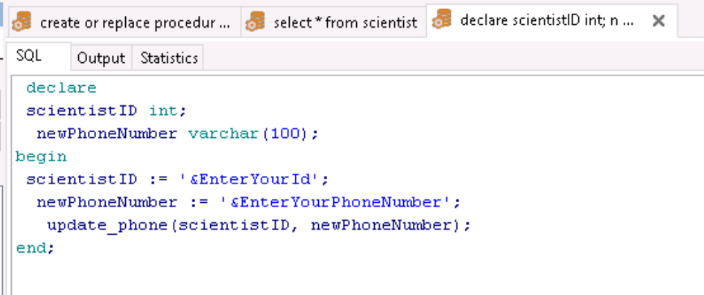
בפשטות, כמה מספרי טלפון השייכים להם אינם עדכניים במערכת, מכיוון ששום אפשרות לא נתנה להם לשנות לאחר ההרשמה.

לשם כך יצרנו פרוצדורה בשם Update\_Phone, המאפשר למדען להזין את המזהה שלו, ואת מספר הטלפון החדש, כדי לעדכן את בסיס הנתונים. (הקוד המלא נמצא בניספח שישי)

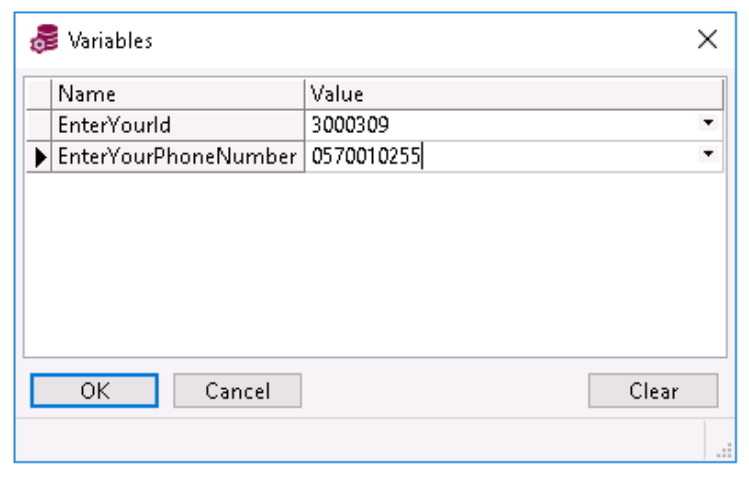
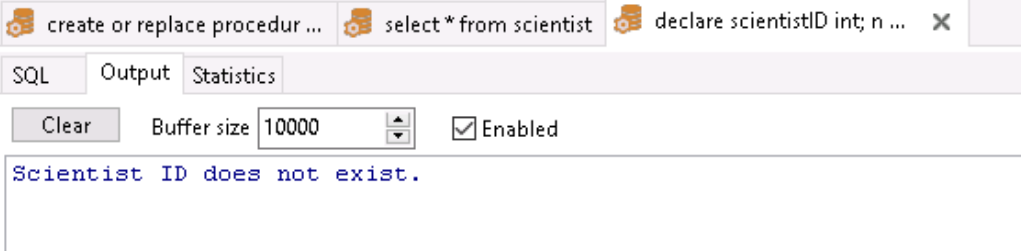
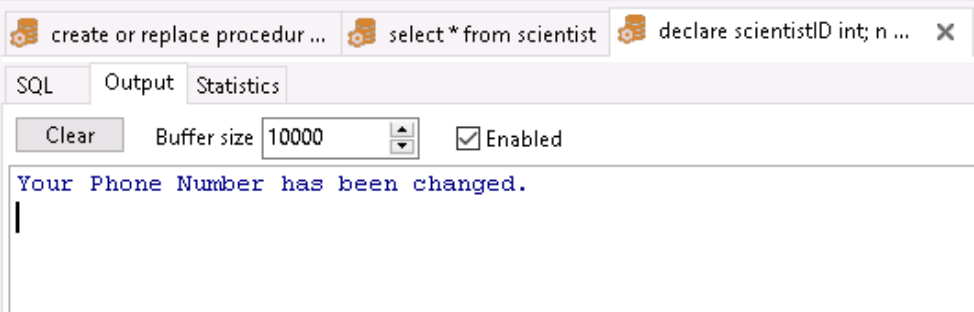
יצירת הפרוצדורה :



ואז נוכל לקרוא לפרוצדורה :



אנו מבקשים מהמדען להזין את המזהה ואת מספר טלפון החדש :



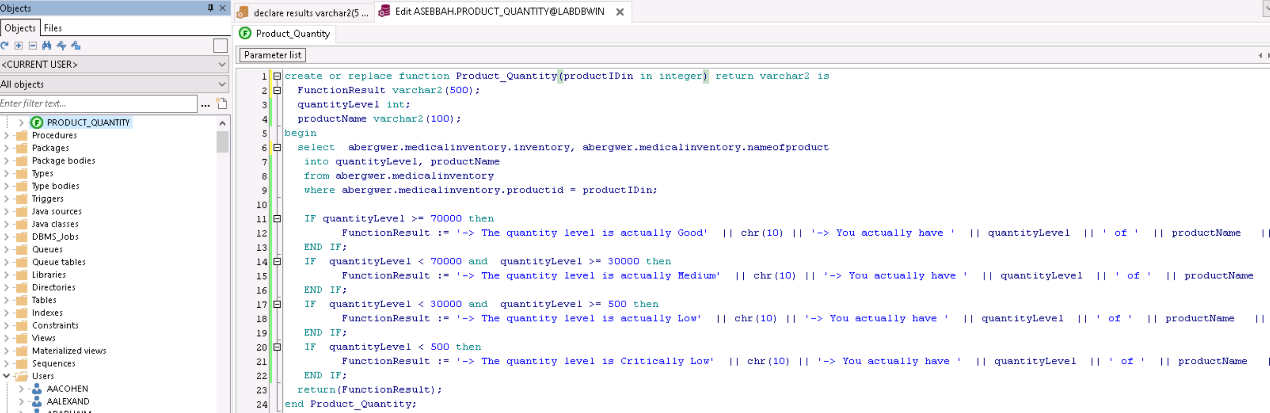
ואז אנו מקבלים אישור הרשמה, או שגיאה אם ​​המזהה אינו קיים.

## פונקציות

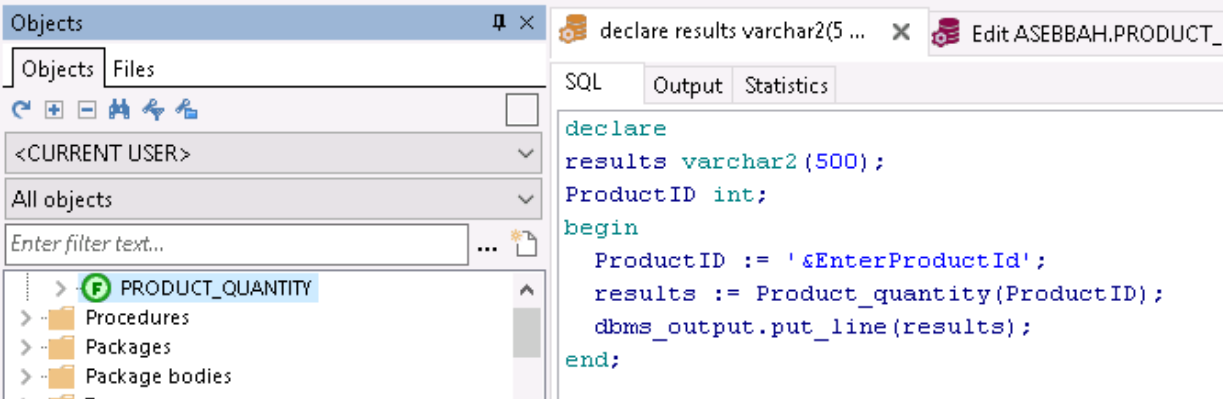
1. משרד הבריאות וחברות ייצור ציוד רפאוי מעוניינים להקים אפליקציה (פונקציה) המציגה אם כמות המוצר, כרגע :טובה / בינונית / נמוכה / נמוכה קריטית, בכדי לדעת האם כדאי להזמין/ליצור חדשים.

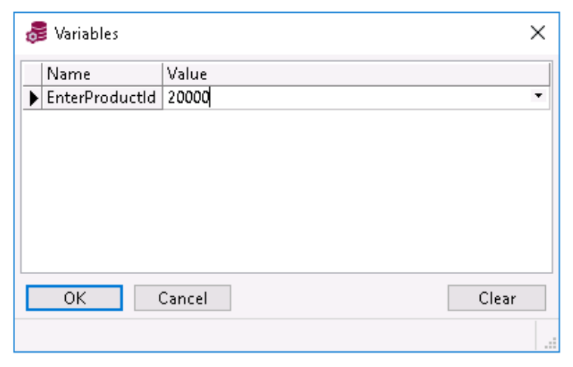
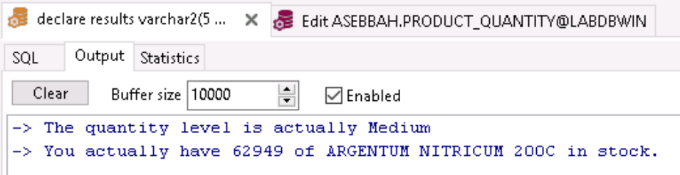
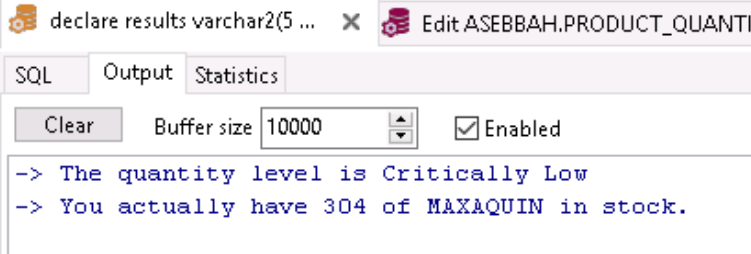
לשם כך יצרנו פונקציה בשם Product\_Quantity, המאפשר למשתמש להזין מספר מזהה של מוצר ולקבל בחזרה האם הכמות טובה/בינונית/קריטי. (הקוד המלא נמצא בניספח שישי)

יצירת הפונקציה :



ואז נוכל לקרוא לפונקציה :



אנו מבקשים מהמשתמש להזין את המזהה של המוצר שהוא רוצה לבדוק :

ואז אנו מקבלים מידע על המוצר.

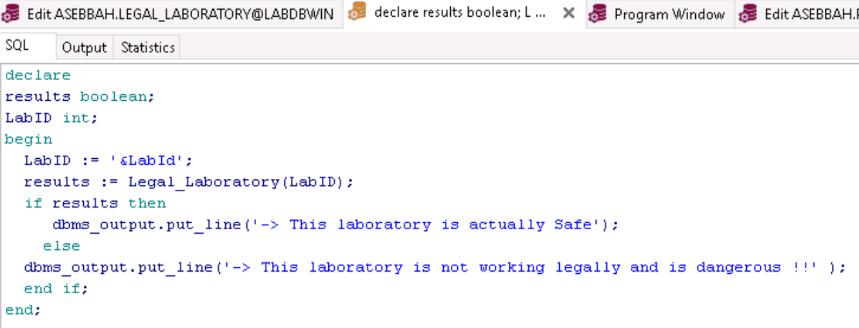
1. הארגון להגנת הסביבה מעוניין לקבל יישום (פונקציה) המאפשר להם לדעת אם מעבדה עובדת כחוק ובבטיחות (על פי החוקים הלאומיים, היא לא יכולה לעבוד על וירוסים מסוכנים עם רמת הגנה נמוכה מדי)

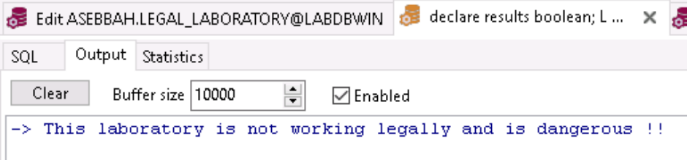
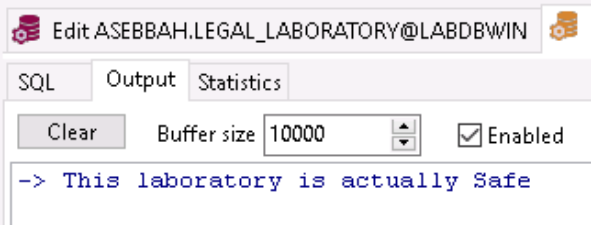
לשם כך יצרנו פונקציה בשם Product\_Quantity, המאפשר למשתמש להזין מספר מזהה של מעבדה ולקבל בחזרה בוליאן שפירושו : אמת – הכל בסדר עם המעבדה, שקר – המעבדה לא חוקית. (הקוד המלא נמצא בניספח שישי(

יצירת הפונקציה :

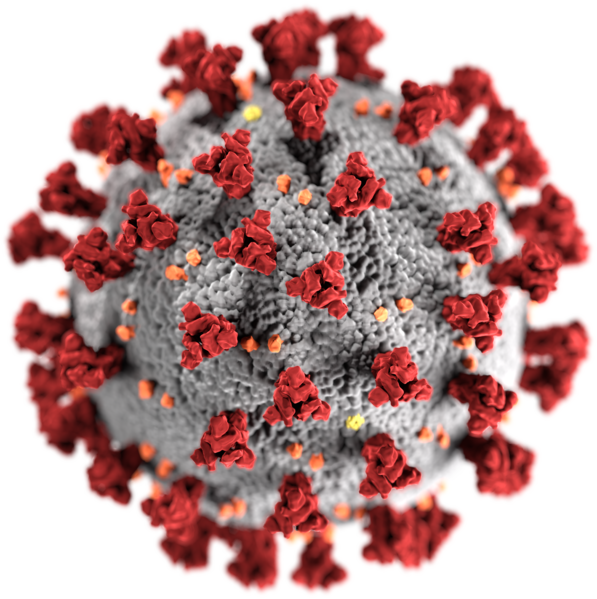


ואז נוכל לקרוא לפונקציה :



אנו מבקשים מהמשתמש להזין את המזהה של המעבדה שהוא רוצה לבדוק :

ואז אנו מקבלים מידע על המעבדה.



# סוף פרויקט

# נספחים

## נספח ראשון: SQL

CREATE TABLE laboratory

(

labLocation VARCHAR(60) NOT NULL,

protectionLevel INT NOT NULL,

labName VARCHAR(60) NOT NULL,

labId INT NOT NULL,

PRIMARY KEY (labId)

);

CREATE TABLE scientist

(

scientistId INT NOT NULL,

scientistName VARCHAR(60) NOT NULL,

age INT NOT NULL,

salary INT NOT NULL,

specialization VARCHAR(60) NOT NULL,

phoneNr VARCHAR(10) NOT NULL,

labId INT NOT NULL,

PRIMARY KEY (scientistId),

FOREIGN KEY (labId) REFERENCES laboratory(labId)

);

CREATE TABLE medicament

(

medName VARCHAR(60) NOT NULL,

medType VARCHAR(60) NOT NULL,

medId INT NOT NULL,

quantity INT NOT NULL,

factCompany VARCHAR(60) NOT NULL,

PRIMARY KEY (medId)

);

CREATE TABLE virus

(

virusName VARCHAR(60) NOT NULL,

virusId INT NOT NULL,

sources VARCHAR(60) NOT NULL,

dangerLevel INT NOT NULL,

PRIMARY KEY (virusId)

);

CREATE TABLE analyzes

(

labId INT NOT NULL,

virusId INT NOT NULL,

PRIMARY KEY (labId, virusId),

FOREIGN KEY (labId) REFERENCES laboratory(labId),

FOREIGN KEY (virusId) REFERENCES virus(virusId)

);

CREATE TABLE creates

(

medId INT NOT NULL,

labId INT NOT NULL,

PRIMARY KEY (medId, labId),

FOREIGN KEY (medId) REFERENCES medicament(medId),

FOREIGN KEY (labId) REFERENCES laboratory(labId)

);

CREATE TABLE cure

(

medId INT NOT NULL,

virusId INT NOT NULL,

PRIMARY KEY (medId, virusId),

FOREIGN KEY (medId) REFERENCES medicament(medId),

FOREIGN KEY (virusId) REFERENCES virus(virusId)

);

CREATE TABLE work\_on

(

scientistId INT NOT NULL,

medId INT NOT NULL,

PRIMARY KEY (scientistId, medId),

FOREIGN KEY (scientistId) REFERENCES scientist(scientistId),

FOREIGN KEY (medId) REFERENCES medicament(medId)

) ;

## נספח שני: אכלוס טבלאות

Laboratory (ID : 60000 to 60999)

FILE\_NAME(laboratory.txt)

FILE\_ACTION(ACTION\_OVERWRITE)

FILE\_COLUMN\_HEADERS(TRUE)

FILE\_COLUMN\_DELIMITER(',')

FILE\_FIELD\_ENCLOSING\_CHARS('')

FILE\_RECORD\_DELIMITER('\r\n')

FILE\_RECORD\_COUNT(1000)

PROJECT\_OUTPUT\_DIRECTORY(C:\Users\avino\Desktop\Semestre B\Projet Bsisei Netounim\Projet\DataBaseCopie\DataBaseFinal)

[labLocation] INTEGER\_IN\_RANGE(1,100) SET(' ') STREET\_NAME() SET(' - ') CITY\_TOWN\_NAMES\_US() SET(' ') ONE\_OF('ISRAEL, USA, FRANCE,ITALY,GERMANY,UK,SPAIN') ASSERT\_IS\_UNIQUE()

[protectionLevel] INTEGER\_IN\_RANGE(0,10)

[labName] LABNAME() ASSERT\_MAX\_LENGTH\_IS(40) ASSERT\_IS\_UNIQUE()

[labId] INCREMENTING\_FROM(60000,1,1,1) TO\_INTEGER()

Medicament (ID : 20000 to 39999)

FILE\_NAME(medicament.txt)

FILE\_ACTION(ACTION\_OVERWRITE)

FILE\_COLUMN\_HEADERS(TRUE)

FILE\_COLUMN\_DELIMITER(',')

FILE\_FIELD\_ENCLOSING\_CHARS('"')

FILE\_RECORD\_DELIMITER('\r\n')

FILE\_RECORD\_COUNT(20000)

PROJECT\_OUTPUT\_DIRECTORY(C:\Users\avino\Desktop\Semestre B\Projet Bsisei Netounim\Projet\DataBase\DataBaseFinal)

[medName] MEDICAMENT() ASSERT\_MAX\_LENGTH\_IS(22) ASSERT\_IS\_UNIQUE()

[medType] APPEND\_ONE\_OF('Antipyretics,Analgesics,Antimalarial,Antibiotics,Antiseptics,Mood stabilizers,Hormone replacements,Oral contraceptives,Stimulants,Tranquilizers,Statins')

[medId] INCREMENTING\_FROM(20000,1,1,1) TO\_INTEGER()

[quantity] INTEGER\_IN\_RANGE(0,100000)

[factCompany] FACTCOMPANIES() ASSERT\_MAX\_LENGTH\_IS(40)

Scientist (ID : 3000000 to 3099999)

FILE\_NAME(scientist.txt)

FILE\_ACTION(ACTION\_OVERWRITE)

FILE\_COLUMN\_HEADERS(TRUE)

FILE\_COLUMN\_DELIMITER(',')

FILE\_FIELD\_ENCLOSING\_CHARS('"')

FILE\_RECORD\_DELIMITER('\r\n')

FILE\_RECORD\_COUNT(100000)

PROJECT\_OUTPUT\_DIRECTORY(C:\Users\avino\Desktop\Semestre B\Projet Bsisei Netounim\Projet\DataBase\DataBaseFinal)

[scientistId] INCREMENTING\_FROM(3000000,1,1,1) TO\_INTEGER()

[scientistName] PERSON\_FULLNAME()

[age] INTEGER\_IN\_RANGE(18,75)

[salary] IN\_RANGE(10000,80000) ROUND\_DECIMAL\_PLACES(2)

[specialization] APPEND\_ONE\_OF('Biochemistry,Bioinformatics,Biostatistics,Biophysics,Cell and Molecular Biology,Ecology/Environmental Science,Entomology,Genetics,Immunology,Marine and Aquatic Biology,Microbiology,Neuroscience,Nutrition and Food Science,Pharmacology,Physiology')

[phoneNr] TELEPHONE\_NUMBER\_FORMATTED(05########)

[labId] INTEGER\_IN\_RANGE(60000,61999)

Virus (ID : 80000 to 83999)

FILE\_NAME(virus.txt)

FILE\_ACTION(ACTION\_OVERWRITE)

FILE\_COLUMN\_HEADERS(TRUE)

FILE\_COLUMN\_DELIMITER(',')

FILE\_FIELD\_ENCLOSING\_CHARS('"')

FILE\_RECORD\_DELIMITER('\r\n')

FILE\_RECORD\_COUNT(4000)

PROJECT\_OUTPUT\_DIRECTORY(C:\Users\avino\Desktop\Semestre B\Projet Bsisei Netounim\Projet\DataBase\DataBaseFinal)

[virusName] VIRUS() ASSERT\_IS\_UNIQUE()

[virusId] INCREMENTING\_FROM(80000,1,1,1) TO\_INTEGER()

[sources] COUNTRY\_NAMES()

[dangerLevel] INTEGER\_IN\_RANGE(1,100)

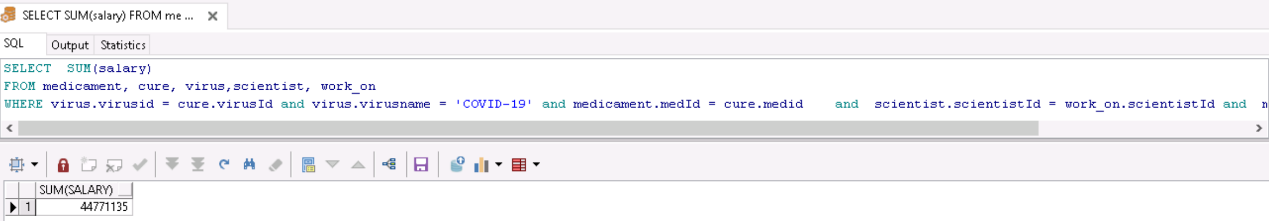


## נספח שלישי: שאילתות ואינדקסים

(האינדקסים מודגשים)

1. SELECT SUM(salary)

FROM medicament, cure, virus,scientist, work\_on

WHERE virus.virusid = cure.virusId and virus.virusname = 'COVID-19' and medicament.medId = cure.medid and scientist.scientistId = work\_on.scientistId and medicament.medId = work\_on.medid

****

1. SELECT medName

FROM medicament

WHERE quantity < 10000

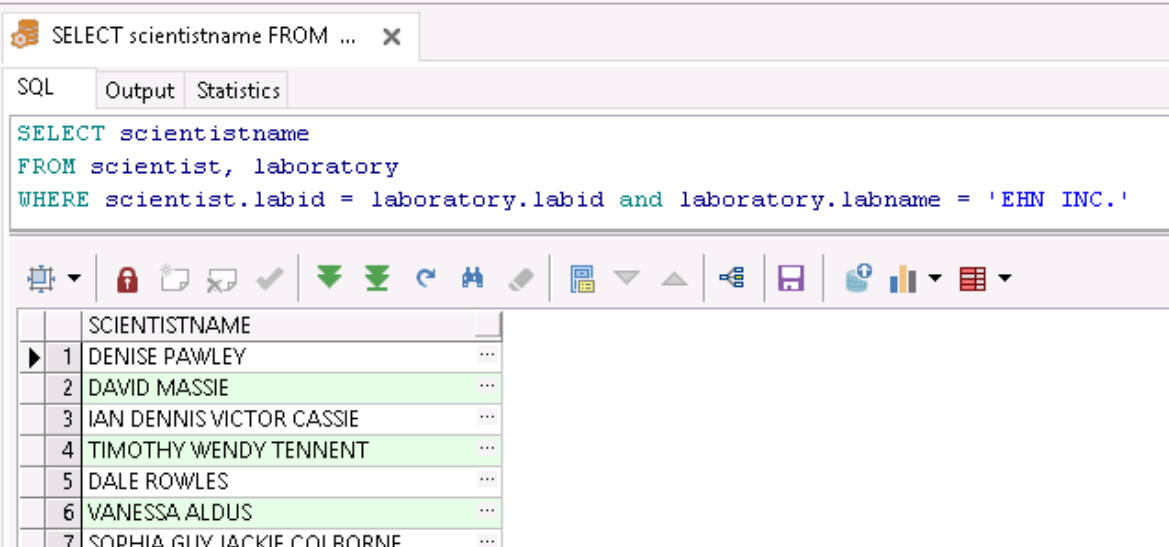
**CREATE INDEX medicamentQuantity**

**ON medicament (quantity)**

1. SELECT scientistname

FROM scientist, laboratory

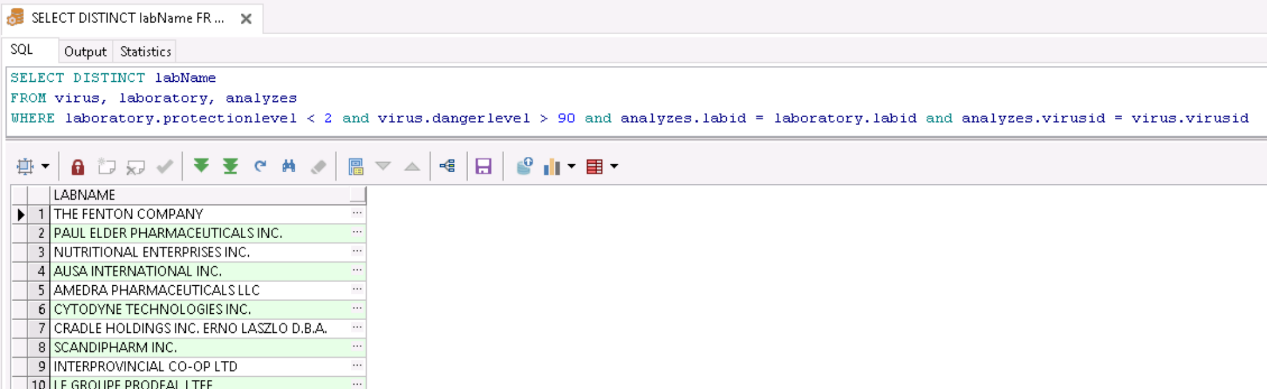
WHERE scientist.labid = laboratory.labid and laboratory.labname = 'EHN INC.'

****

1. SELECT DISTINCT labName

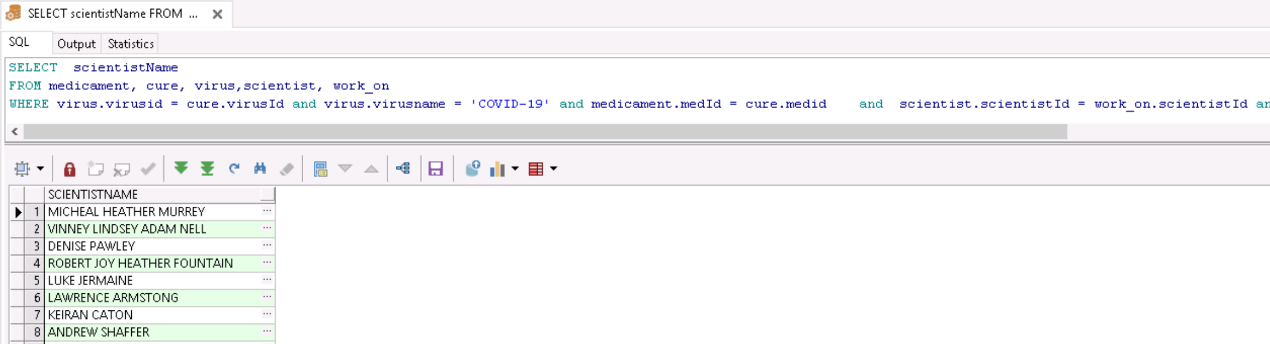
FROM virus, laboratory, analyzes

WHERE laboratory.protectionlevel < 2 and virus.dangerlevel > 90 and analyzes.labid = laboratory.labid and analyzes.virusid = virus.virusid



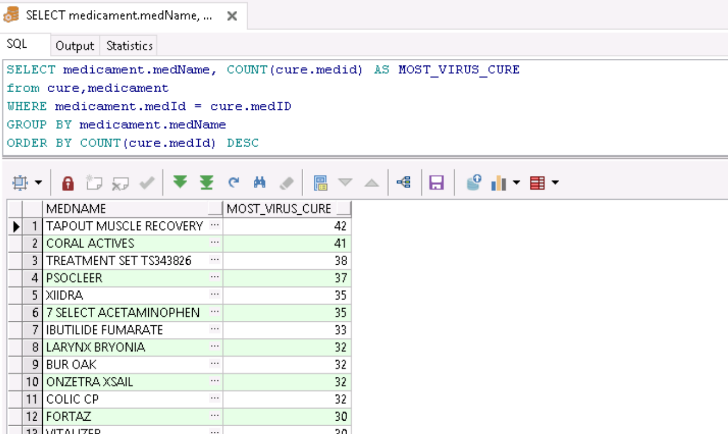
1. SELECT scientistName

FROM medicament, cure, virus,scientist, work\_on

WHERE virus.virusid = cure.virusId and virus.virusname = 'COVID-19' and medicament.medId = cure.medid and scientist.scientistId = work\_on.scientistId and medicament.medId = work\_on.medid

1. SELECT medicament.medName, COUNT(cure.medid) AS MOST\_VIRUS\_CURE

from cure,medicament

WHERE medicament.medId = cure.medID

GROUP BY medicament.medName

ORDER BY COUNT(cure.medId) DESC

**CREATE INDEX medicamentIdentification**

**ON cure (medId)**

**CREATE INDEX medicamentNameIdentification**

**ON medicament (medName)**

1. SELECT DISTINCT scientistName

FROM medicament, cure, virus,scientist, work\_on

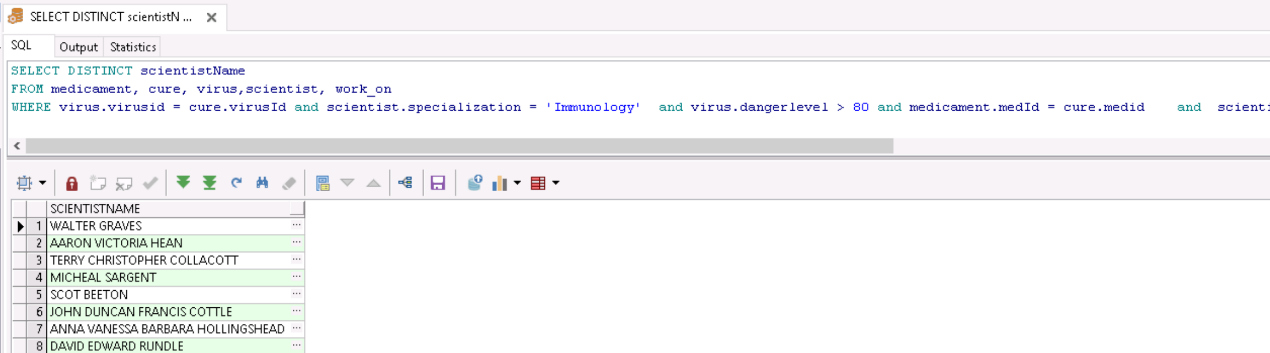
WHERE virus.virusid = cure.virusId and scientist.specialization = 'Immunology' and virus.dangerlevel > 80 and medicament.medId = cure.medid and scientist.scientistId = work\_on.scientistId and medicament.medId = work\_on.medid

**CREATE INDEX scientistSpecialization**

**ON scientist ('Immunology')**

**CREATE INDEX virusDanger**

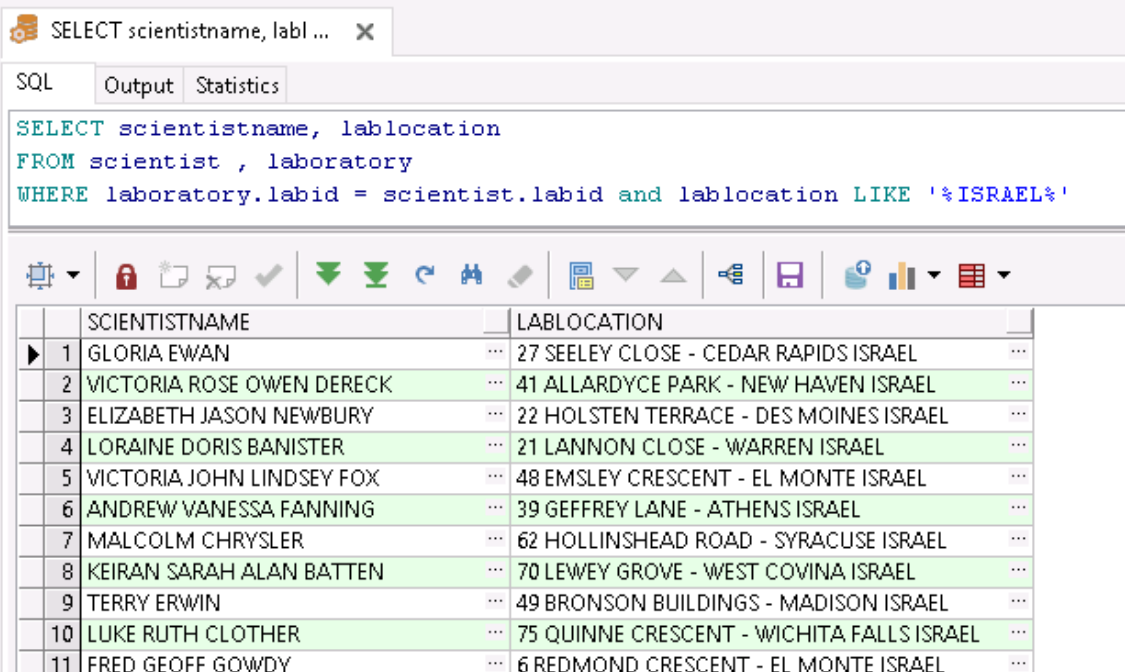
**ON virus (dangerlevel)**



1. SELECT scientistname, lablocation

FROM scientist , laboratory

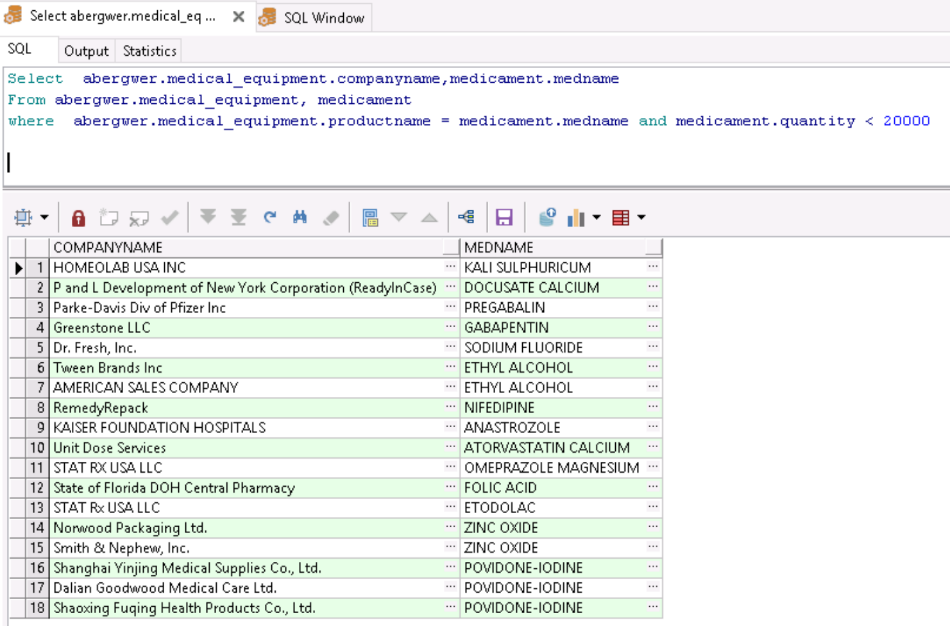
WHERE laboratory.labid = scientist.labid and lablocation LIKE '%ISRAEL%'



## נספח רביעי : שאילתות על איחוד אגפים

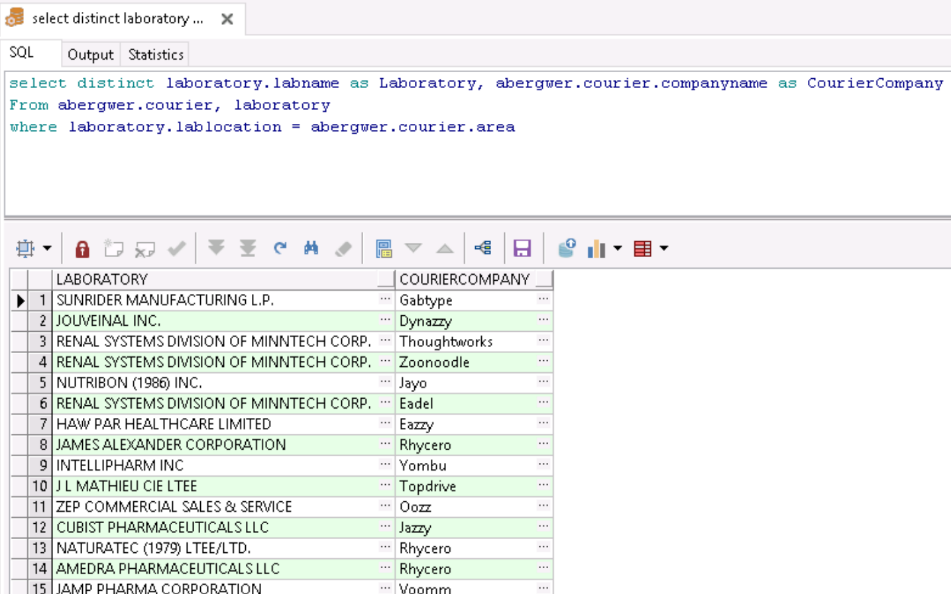
1. SELECT **abergwer**.medical\_equipment.companyname, medicament.mednam

FROM **abergwer**.medical\_equipment, medicament

WHERE **abergwer**.medical\_equipment.productname = medicament.medname and medicament.quantity < 20000

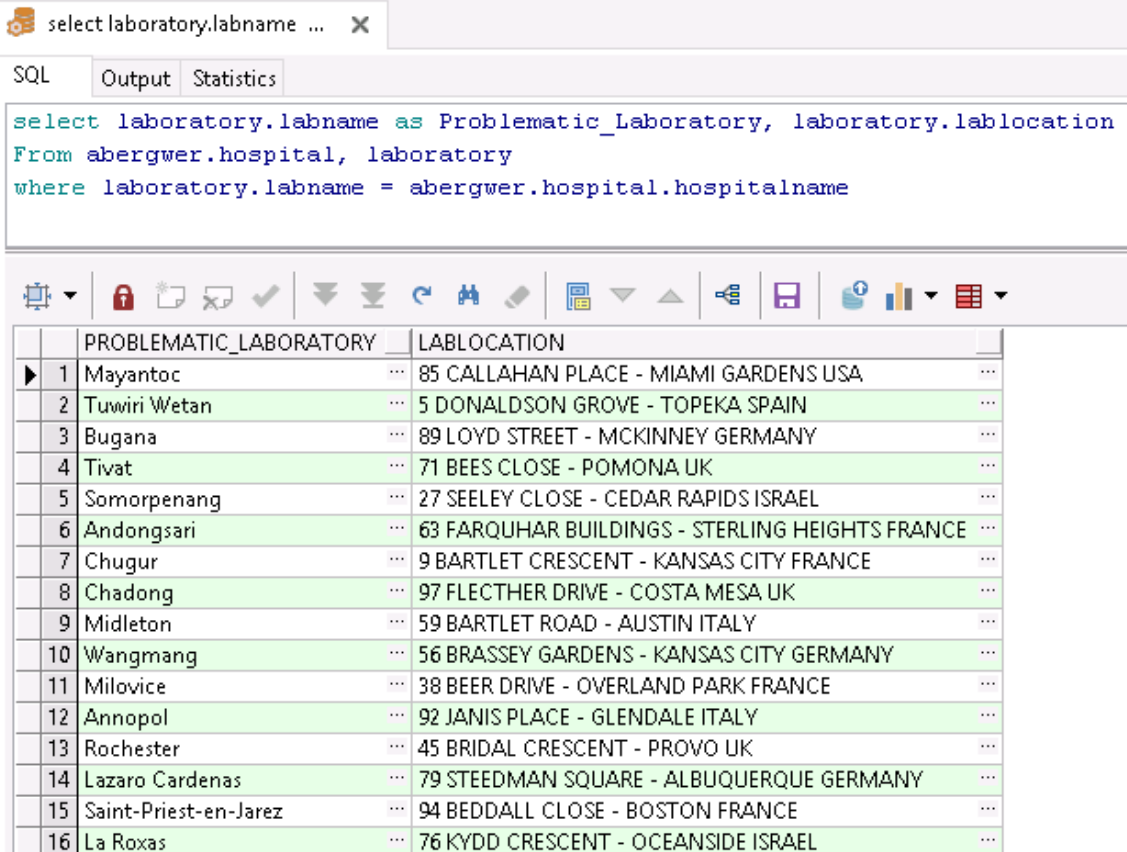
1. SELECT DISTINCT laboratory.labname as Laboratory, **abergwer**.courier.companyname as CourierCompany

FROM **abergwer**.courier, laboratory

WHERE laboratory.lablocation = **abergwer**.courier.area

1. SELECT laboratory.labname as Problematic\_Laboratory, laboratory.lablocation

FROM **abergwer**.hospital, laboratory

WHERE laboratory.labname = **abergwer**.hospital.hospitalname

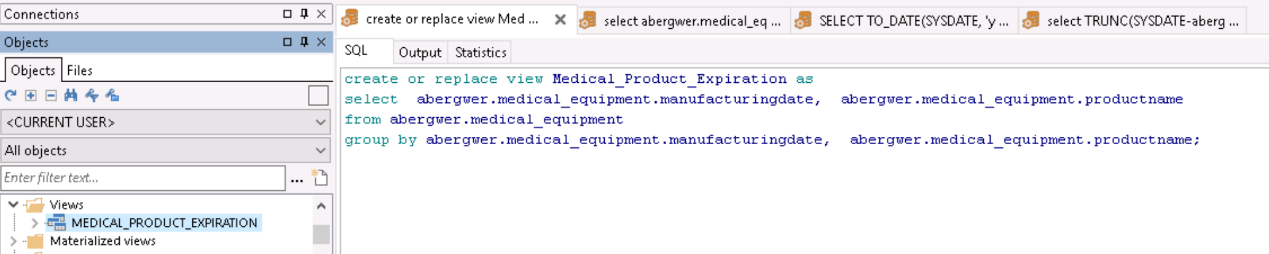
## נספח חמישי : View וגרפים

View 1 :

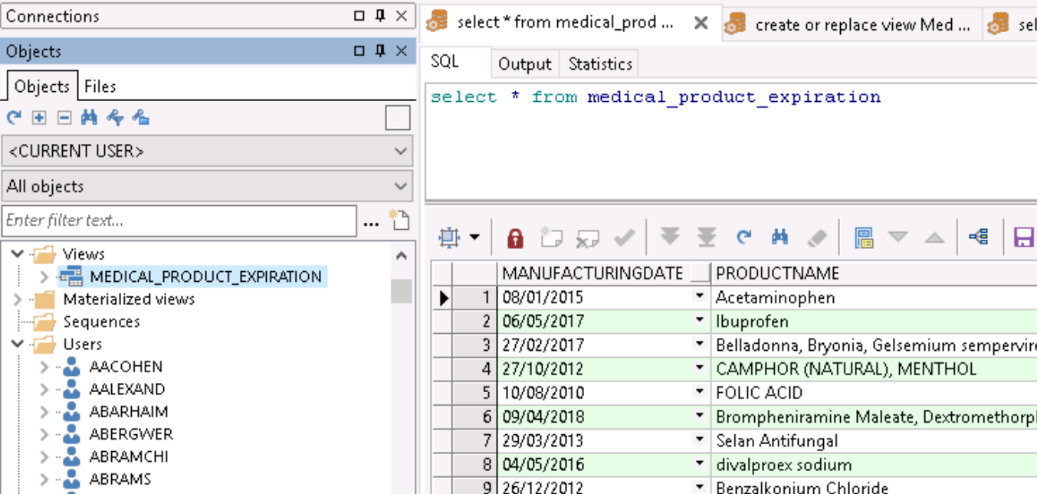
1. CREATE or REPLACE VIEW Medical\_Product\_Expiration as

SELECT **abergwer**.medical\_equipment.manufacturingdate, **abergwer**.medical\_equipment.productname

FROM **abergwer**.medical\_equipment

GROUP BY **abergwer**.medical\_equipment.manufacturingdate, **abergwer**.medical\_equipment.productname;

1. SELECT\* FROM medical\_product\_expiration



1. SELECT medical\_product\_expiration.manufacturingdate, medical\_product\_expiration.productname

FROM medical\_product\_expiration

WHERE TRUNC(SYSDATE-medical\_product\_expiration.manufacturingdate, 0) > 7300

) <-7300 הוא מספר הימים ב 20 שנה(

View 2 :

1. CREATE or REPLACE VIEW medicament\_quantity as

SELECT medid,quantity

FROM medicament

GROUP BY quantity,medid

ORDER BY quantity

1. SELECT \* from medicament\_quantity

View 3 :

1. CREATE or REPLACE VIEW product\_company\_creator as

SELECT **abergwer**.medical\_equipment.productname as ProductName , **abergwer**.medical\_equipment.companyName as CompanyName

FROM abergwer.medical\_equipment

GROUP BY **abergwer**.medical\_equipment.productname,**abergwer**.medical\_equipment.companyName

1. SELECT \* from product\_company\_creator

View 4 :

1. CREATE or REPLACE VIEW Laboratory\_Location as

SELECT laboratory.labname as LaboratoryName, laboratory.lablocation as LocatedIn

FROM laboratory

GROUP BY laboratory.labname, laboratory.lablocation

1. SELECT \* from medicament\_quantity

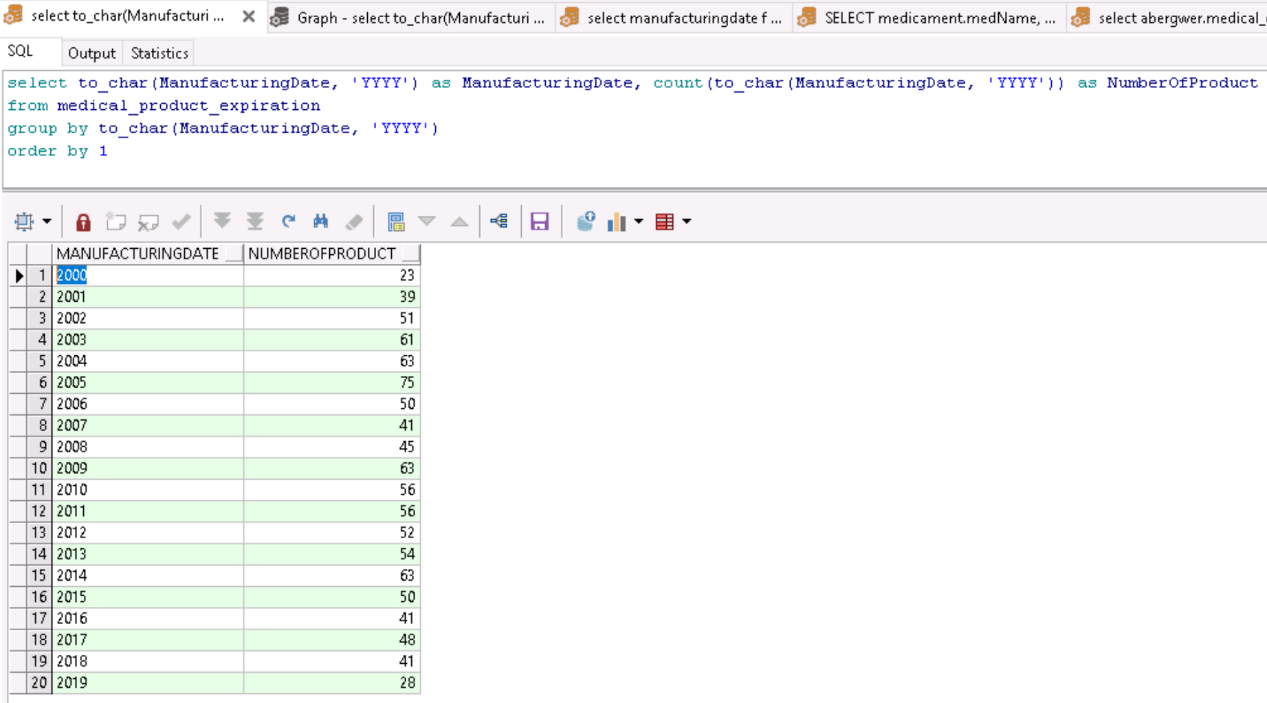
Graph 1:

1. SELECT to\_char(ManufacturingDate, 'YYYY') as ManufacturingDate, Count(to\_char(ManufacturingDate, 'YYYY')) as NumberOfProduct

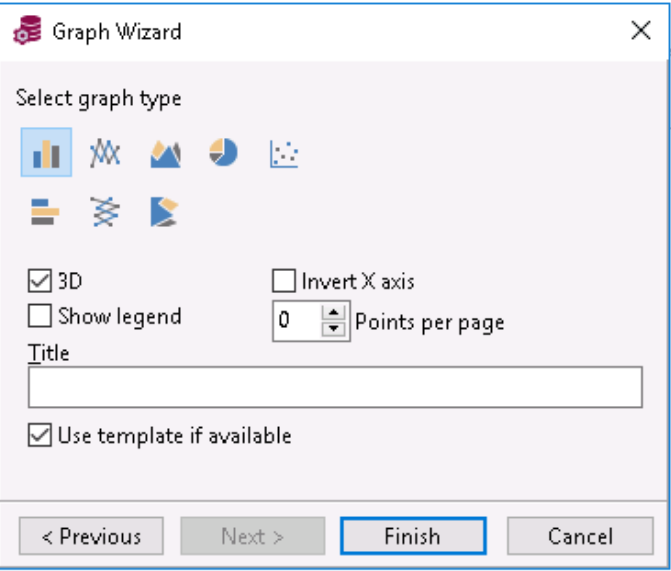
FROM medical\_product\_expiration

GROUP BY to\_char(ManufacturingDate, 'YYYY')

ORDER BY 1



ואז אנו בוחרים את הנתונים על ציר ה- X (שנים) ועל ציר ה- Y (מספר המוצרים).

ולבסוף אנו בוחרים את צורת הגרף שלנו.

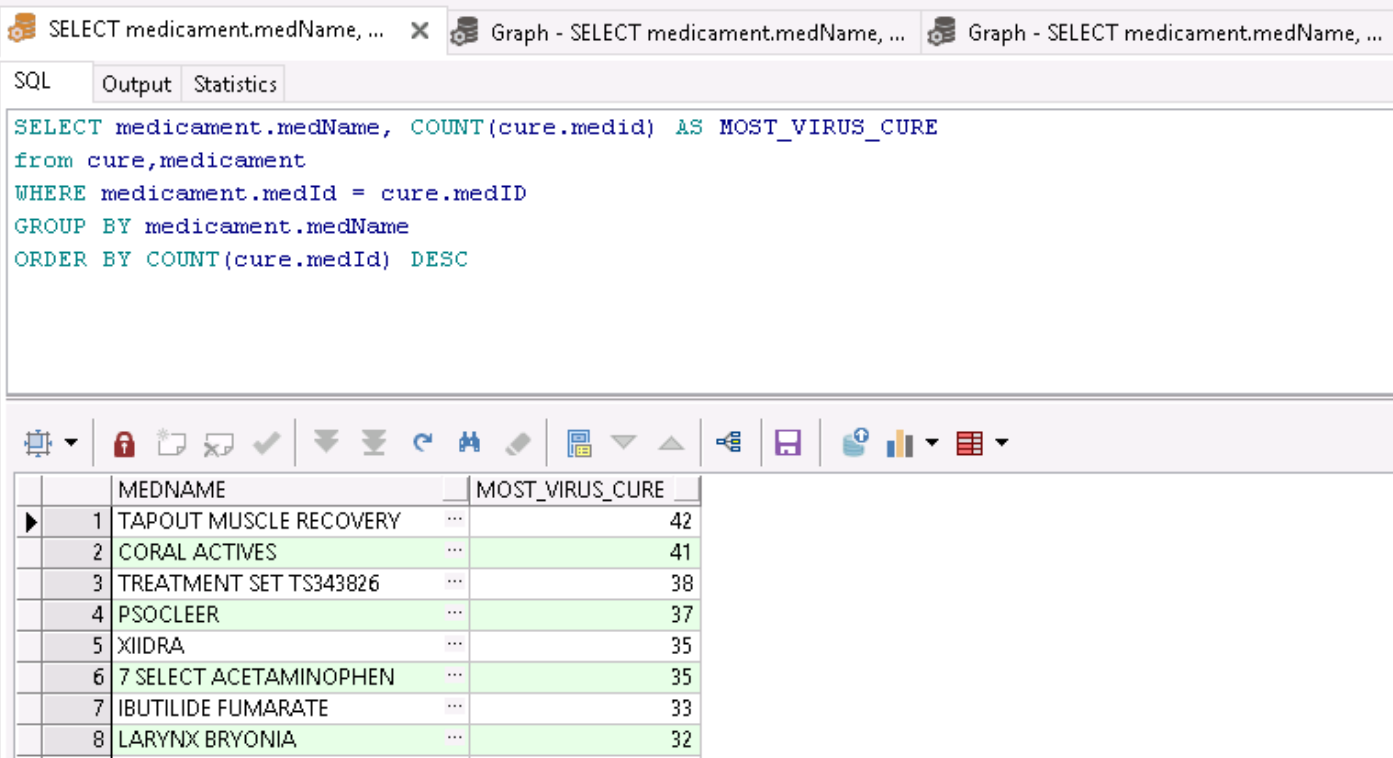
Graph 2 :

1. SELECT medicament.medName, COUNT(cure.medid) AS MOST\_VIRUS\_CURE

from cure,medicament

WHERE medicament.medId = cure.medID

GROUP BY medicament.medName

ORDER BY COUNT(cure.medId) DESC

## נספח שישי : פונקציות ופרוצדרות

Procedure 1:

1. CREATE or REPLACE PROCEDURE **product\_expiration**(productNameIn in varchar2)

IS

res boolean := false;

BEGIN

FOR c1 in ( select 1 from medical\_product\_expiration where medical\_product\_expiration.productname = productNameIn and TRUNC(SYSDATE-medical\_product\_expiration.manufacturingdate, 0) > 7300 )

LOOP

res := true;

exit;

END LOOP;

IF res THEN

dbms\_output.put\_line('-> Warning : ' || productNameIn || ' product is Expired !!' );

dbms\_output.put\_line('Please discard the product, or contact the Health Ministry at \*3600' );

ELSE

dbms\_output.put\_line('-> ' ||productNameIn || ' product is actualy Good.' );

END IF;

END product\_expiration;

Procedure call :

1. DECLARE

productNameIn varchar2(100);

BEGIN

productNameIn := '&EnterProductName';

product\_expiration(productNameIn);

END;

Procedure 2:

1. CREATE or REPLACE PROCEDURE **update\_phone**(scientistIdin in integer, newPhoneNumber in varchar2)

IS

cnt number;

BEGIN

SELECT count(\*) into cnt

FROM scientist

WHERE scientist.scientistid = scientistIdin;

IF cnt = 1 THEN

UPDATE scientist

SET phoneNr = newPhoneNumber

WHERE scientist.scientistid = scientistIdin;

dbms\_output.put\_line('Your Phone Number has been changed.');

ELSE

dbms\_output.put\_line('Scientist ID does not exist.');

END IF;

END;

Procedure call :

1. DECLARE

scientistID int;

newPhoneNumber varchar(100);

BEGIN

scientistID := '&EnterYourId';

newPhoneNumber := '&EnterYourPhoneNumber';

update\_phone(scientistID, newPhoneNumber);

END;

Function 1:

1. CREATE or REPLACE FUNCTION **Product\_Quantity**(productIDin in integer) return varchar2 is

FunctionResult varchar2(500);

quantityLevel int;

productName varchar2(100);

BEGIN

SELECT **abergwer**.medicalinventory.inventory, **abergwer**.medicalinventory.nameofproduct

INTO quantityLevel, productName

FROM **abergwer**.medicalinventory

WHERE **abergwer**.medicalinventory.productid = productIDin;

IF quantityLevel >= 70000 then

FunctionResult := '-> The quantity level is actually Good' || chr(10) || '-> You actually have ' || quantityLevel || ' of ' || productName || ' in stock.';

END IF;

IF quantityLevel < 70000 and quantityLevel >= 30000 THEN

FunctionResult := '-> The quantity level is actually Medium' || chr(10) || '-> You actually have ' || quantityLevel || ' of ' || productName || ' in stock.';

END IF;

IF quantityLevel < 30000 and quantityLevel >= 500 THEN

FunctionResult := '-> The quantity level is actually Low' || chr(10) || '-> You actually have ' || quantityLevel || ' of ' || productName || ' in stock.';

END IF;

IF quantityLevel < 500 THEN

FunctionResult := '-> The quantity level is Critically Low' || chr(10) || '-> You actually have ' || quantityLevel || ' of ' || productName || ' in stock.';

END IF;

RETURN(FunctionResult);

END Product\_Quantity;

Function call :

1. DECLARE

results varchar2(500);

ProductID int;

BEGIN

ProductID := '&EnterProductId';

results := Product\_quantity(ProductID);

dbms\_output.put\_line(results);

END;

Function 2:

1. CREATE or REPLACE FUNCTION **legal\_laboratory**(labIDin in integer) return boolean is

FunctionResult boolean;

cnt number;

BEGIN

SELECT count(\*) into cnt

FROM virus, laboratory, analyzes

WHERE laboratory.protectionlevel < 2 and virus.dangerlevel > 90 and analyzes.labid = laboratory.labid and analyzes.virusid = virus.virusid and analyzes.labid = labIDin;

IF cnt >= 1 THEN

FunctionResult := False;

ELSE

FunctionResult := TRUE;

END IF;

RETURN(FunctionResult);

END legal\_laboratory;

Function call :

1. DECLARE

results boolean;

LabID int;

BEGIN

LabID := '&LabId';

results := Legal\_Laboratory(LabID);

IF results then

dbms\_output.put\_line('-> This laboratory is actually Safe');

ELSE

dbms\_output.put\_line('-> This laboratory is not working legally and is dangerous !!' );

END IF;

END;