

אבינועם סבע – 337966642

אליעזר לנקרי - 588939

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

**בס"ד**

**מיני פרויקט בבסיסי נתונים**

תוכן עניינים

[מבוא 3](#_Toc517367483)

[עבודת הכנה והכרת התוכנה 4](#_Toc517367484)

[תרשים ERD 4](#_Toc517367485)

[תיאור הישויות והקשרים 5](#_Toc517367486)

[ישויות 5](#_Toc517367487)

[קשרים 5](#_Toc517367488)

[נרמול הטבלאות 5](#_Toc517367489)

[פרוקים 5](#_Toc517367490)

[תרשים DSD 6](#_Toc517367491)

[הפרויקט שלנו 7](#_Toc517367492)

[תרשים ERD 7](#_Toc517367493)

[תיאור הישויות והקשרים 7](#_Toc517367494)

[ישויות 7](#_Toc517367495)

[קשרים 8](#_Toc517367496)

[נרמול הטבלאות 8](#_Toc517367497)

[תרשים DSD 9](#_Toc517367498)

# 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**

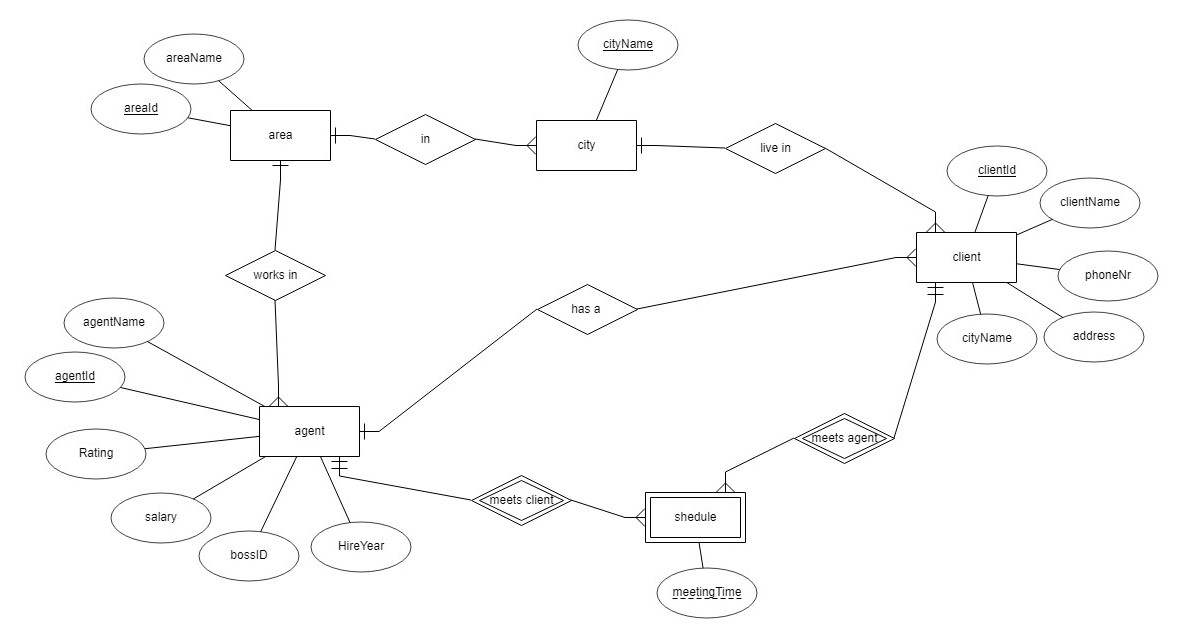
מתנדבים : ת"ז, שם, כתובת, עיר, התמחויות, מס טלפון.

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

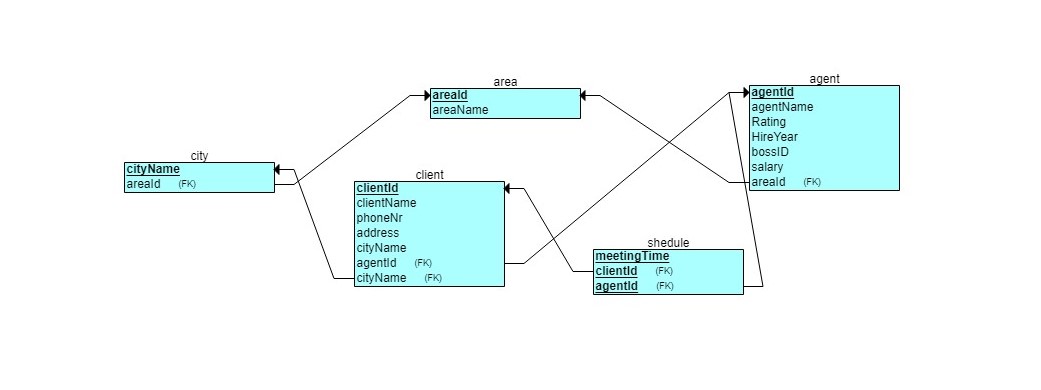
כל התלויות הן BCNF כיוון שה X בתלות הינו מפתח על. ולכן כמובן כל התלויות הינן 3NF .

### 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)

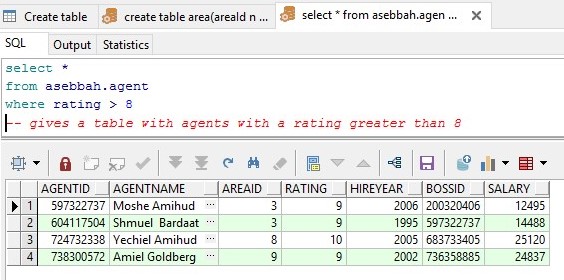
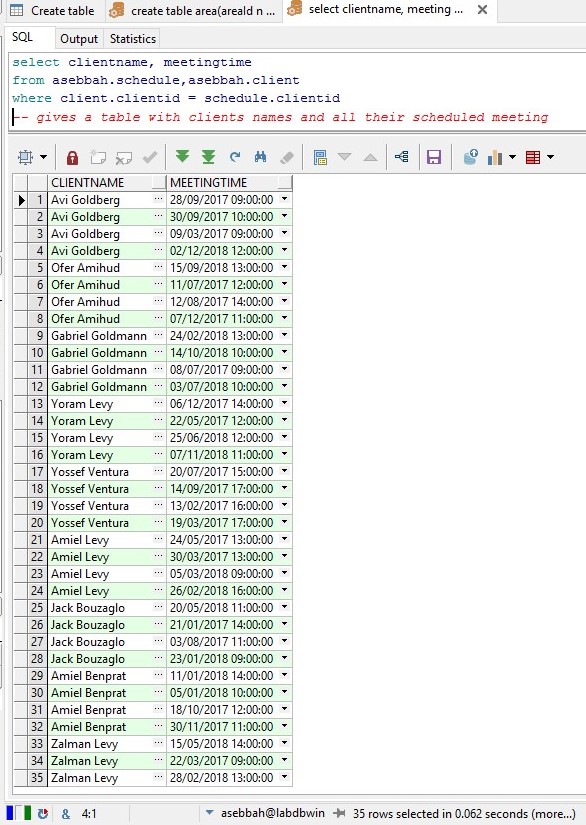
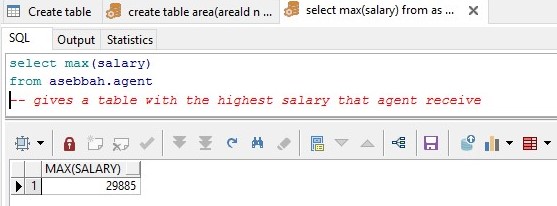
## תרשים DSD

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

****

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

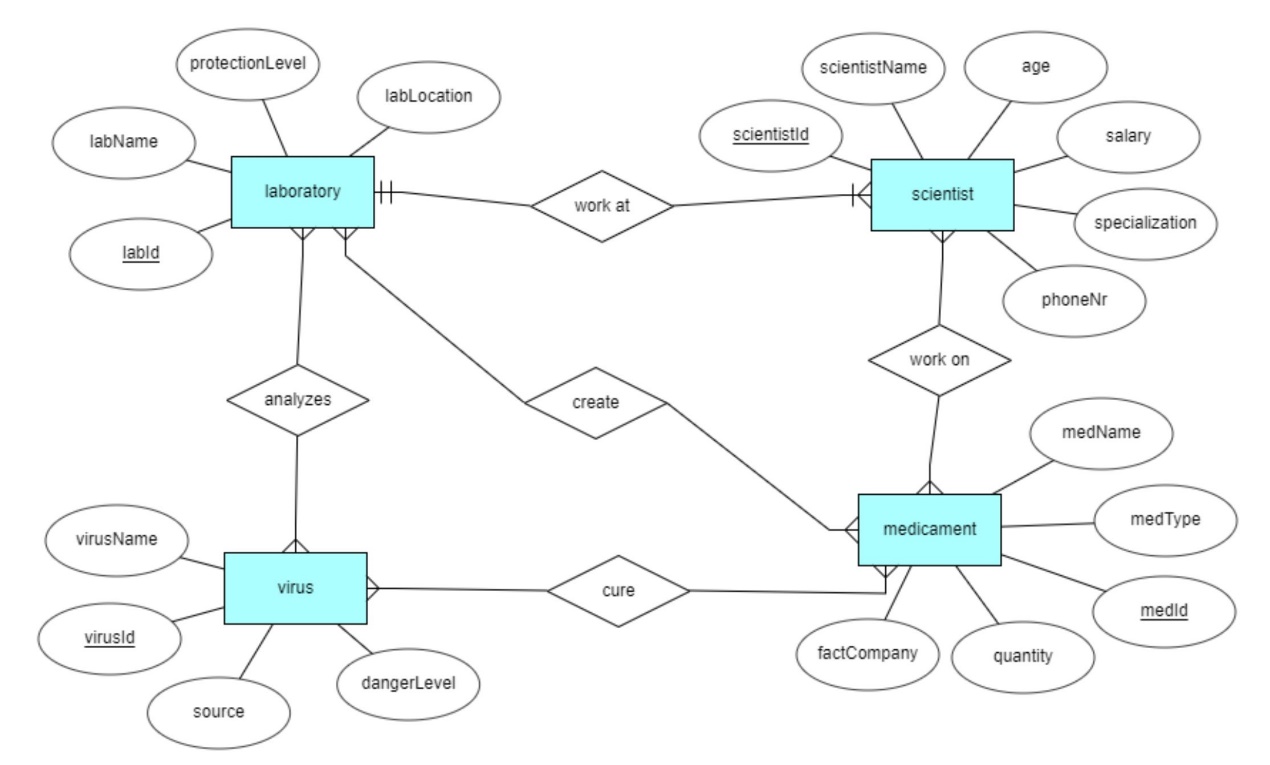
לאחר יצירת הטבלאות וטעינת נתונים, אני אנסה כמה שאילתות לראות אם הכל עובד טוב עם הבסיס הנתונים הגרעיני שקיבלנו :



# הפרויקט שלנו

**תרשים ERD**

כפי שהזכרנו, במחלקה 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)(FK)
* 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
* Virus – ישות זאת אחראית על כל הוירוסים הנמצאים במערכת.
* virusId – מספר מזהה של הוירוס (PK) (FK)
* virusName – שם הוירוס
* source – מקורו
* dangerLevel – רמת סכנת
* Medicament – ישות זאת אחראית על כל התרופות הנמצאים במערכת.
* medId – מספר מזהה של תרופה (PK) (FK)
* medName– שם התרופה
* medType – סוג התרופה (חיסון או תרופה)
* quantity – כמות
* factCompany – חברת יצור

**קשרים**

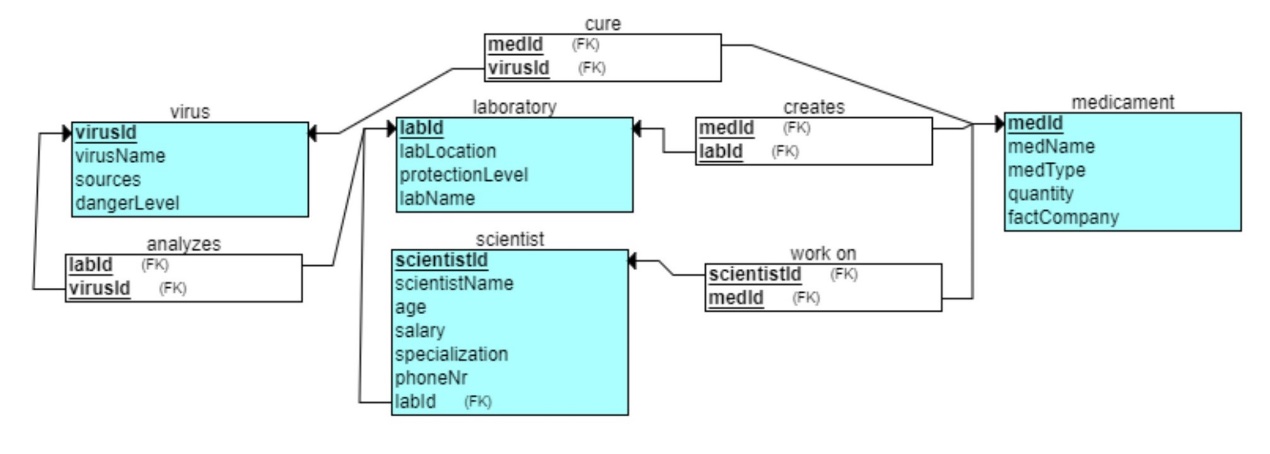
* work at – הקשר בין laboratory לבין scientist. הקשר הוא M:1 משום שמדען אחד יכול לעבוד רק במעבדה אחת, ובמעבדה אחת יכול לעבוד כמה מדענים.
* work on – הקשר בין scientist לבין medicament. הקשר הוא M:M משום שמדען אחד יכול לעבוד על כמה תרופות, ותרופה אחת יכולה להיות מעובד על ידי כמה מדענים..
* analyzes – 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.

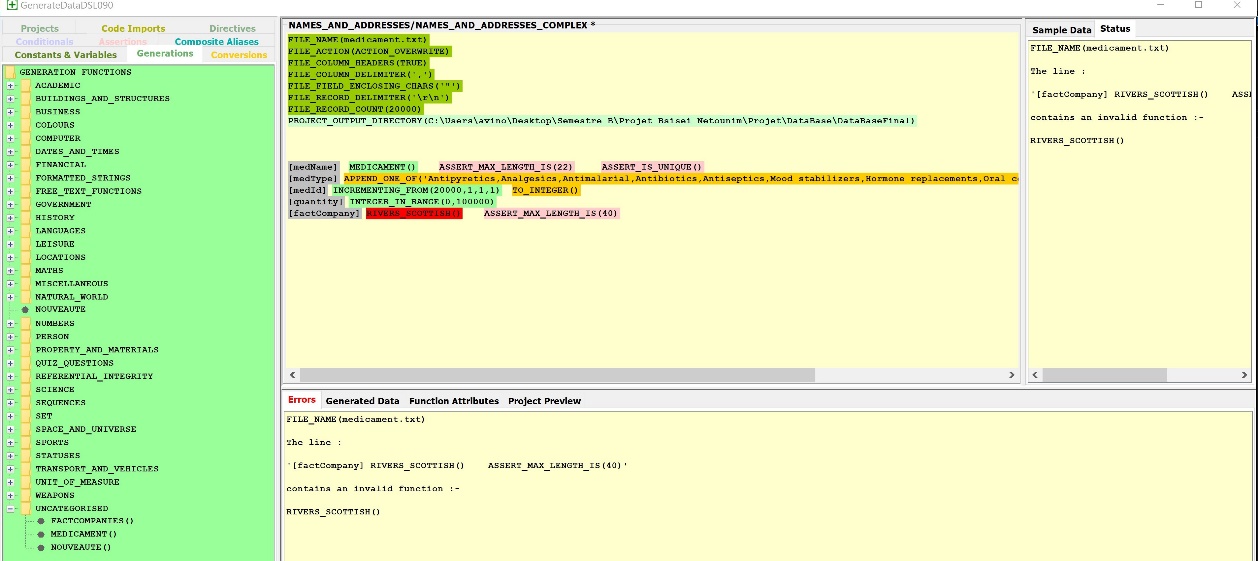
## הכנסת נתונים

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.

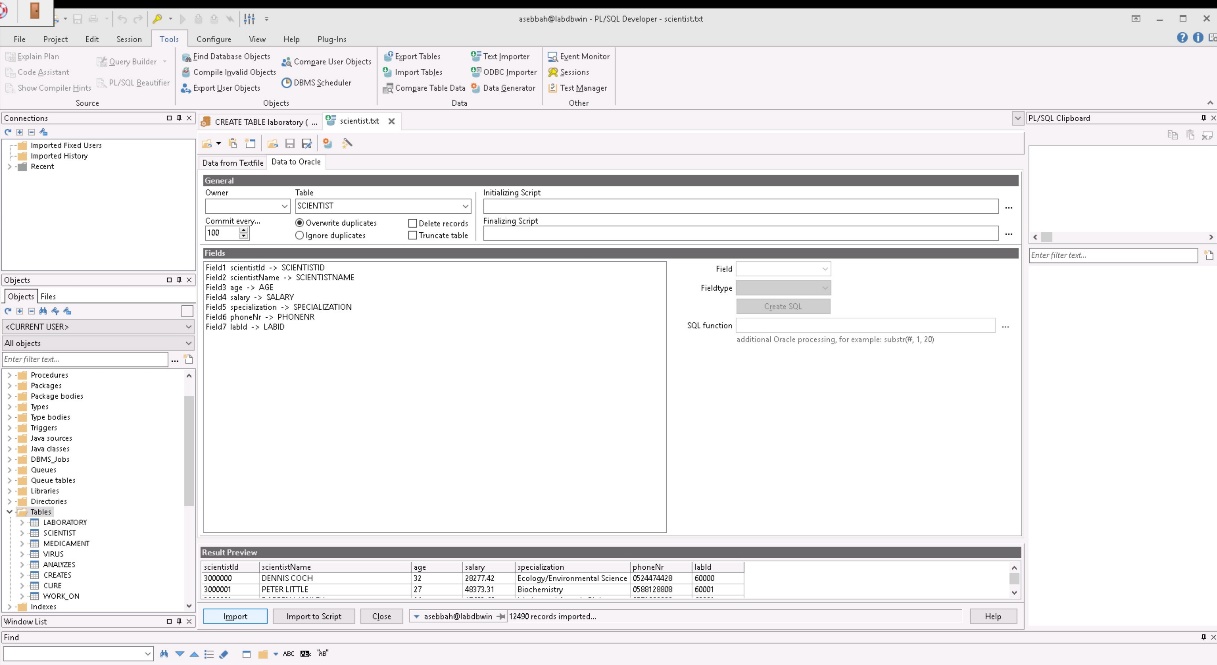
<- <https://sourceforge.net/projects/generatedata/files/>

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..



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.

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 השני) 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 זמני ריצה :

1. 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. 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)

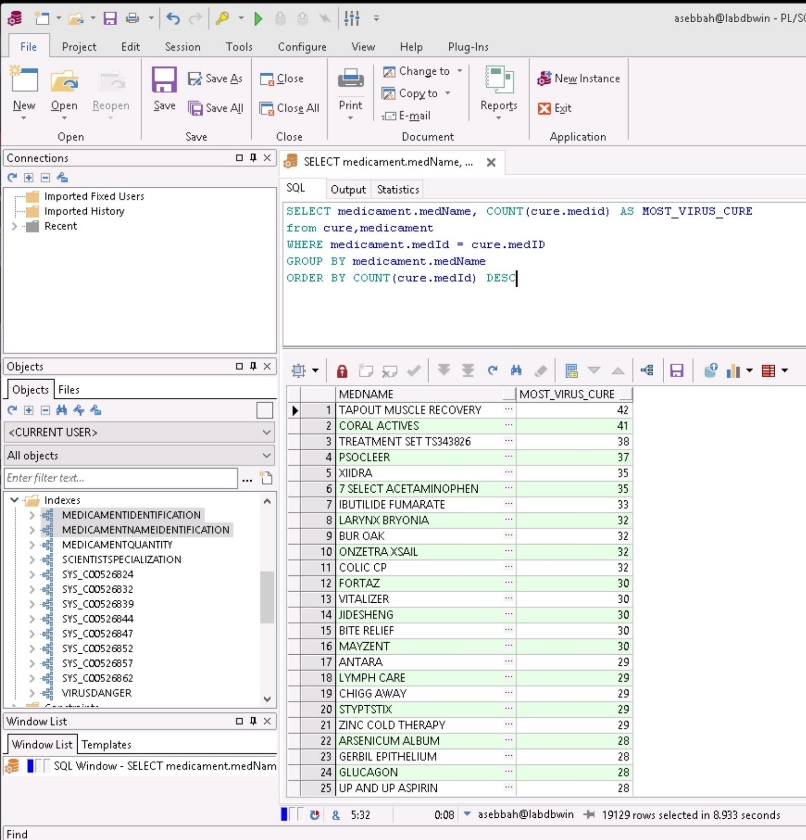
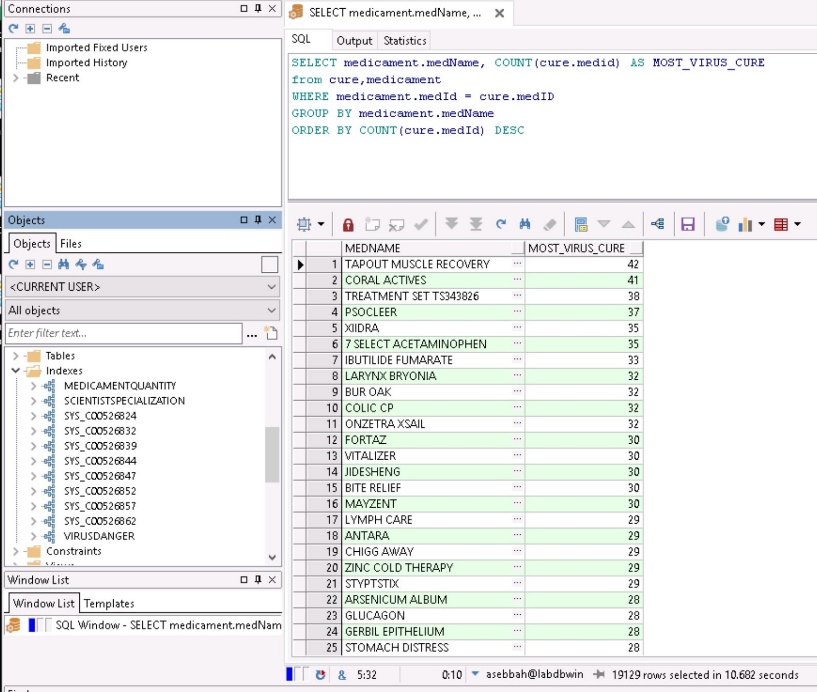
1. אינדקס השלישי : לפני הגדרת האינדקס הייתה לנו זמן ביצוע של 01.074s, לאחר הגדרתו קיבלנו זמן ביצוע של 00.714s.

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

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

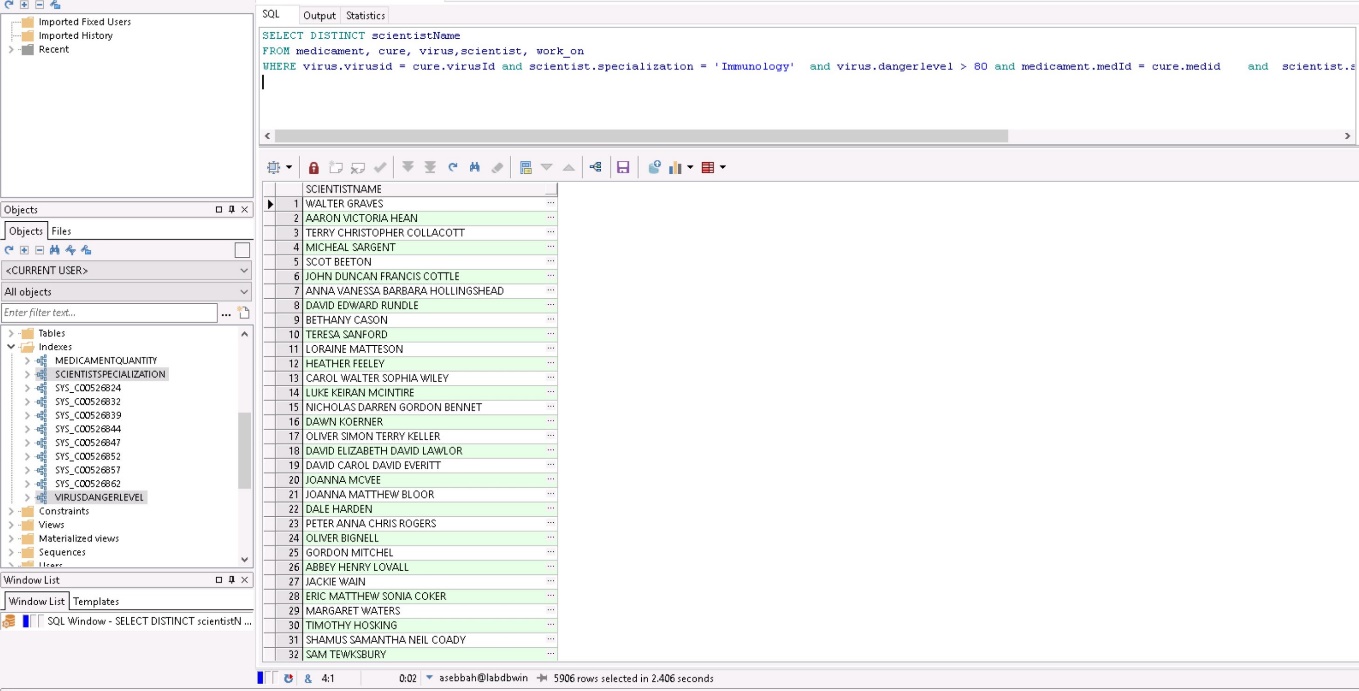
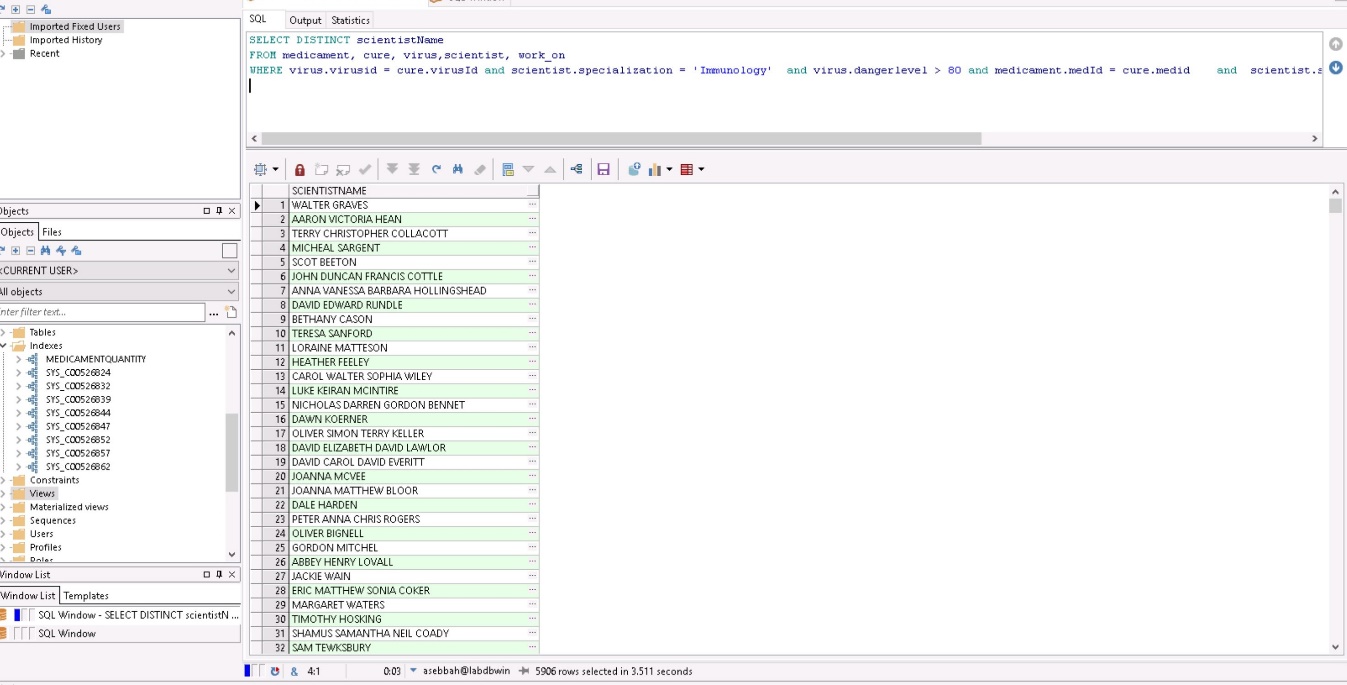
אחרי

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



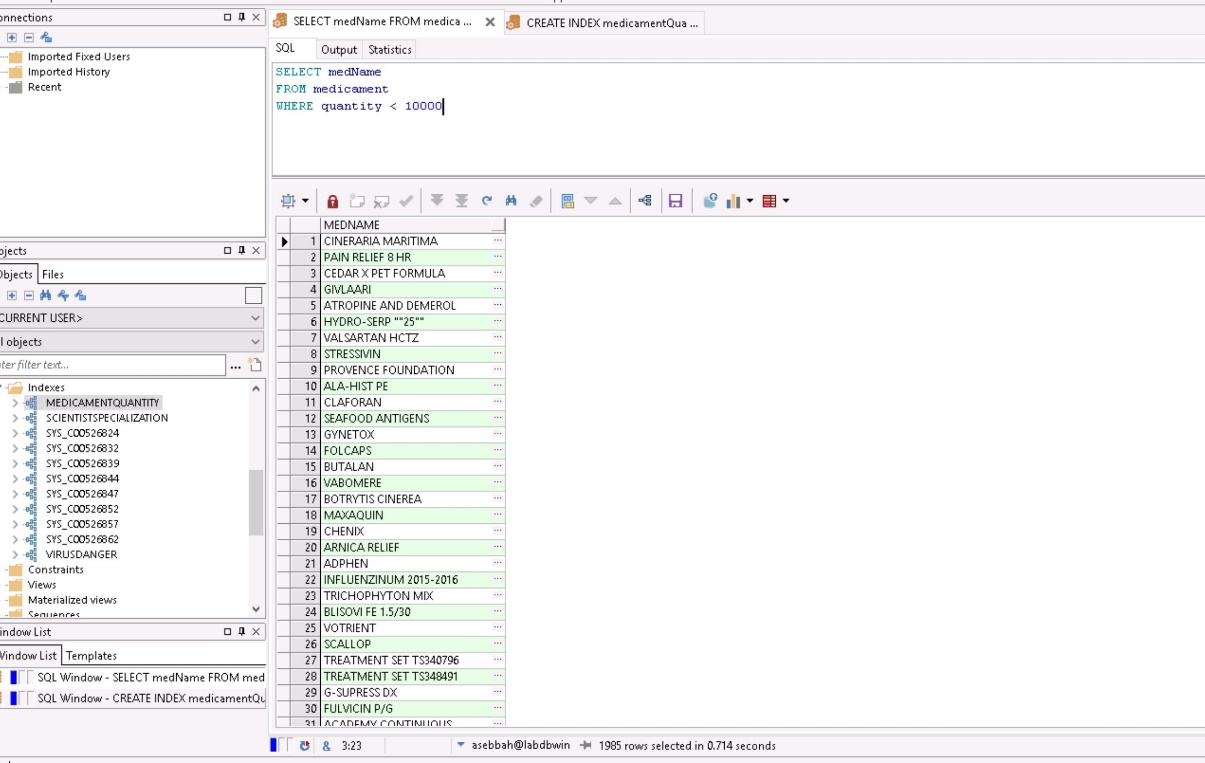
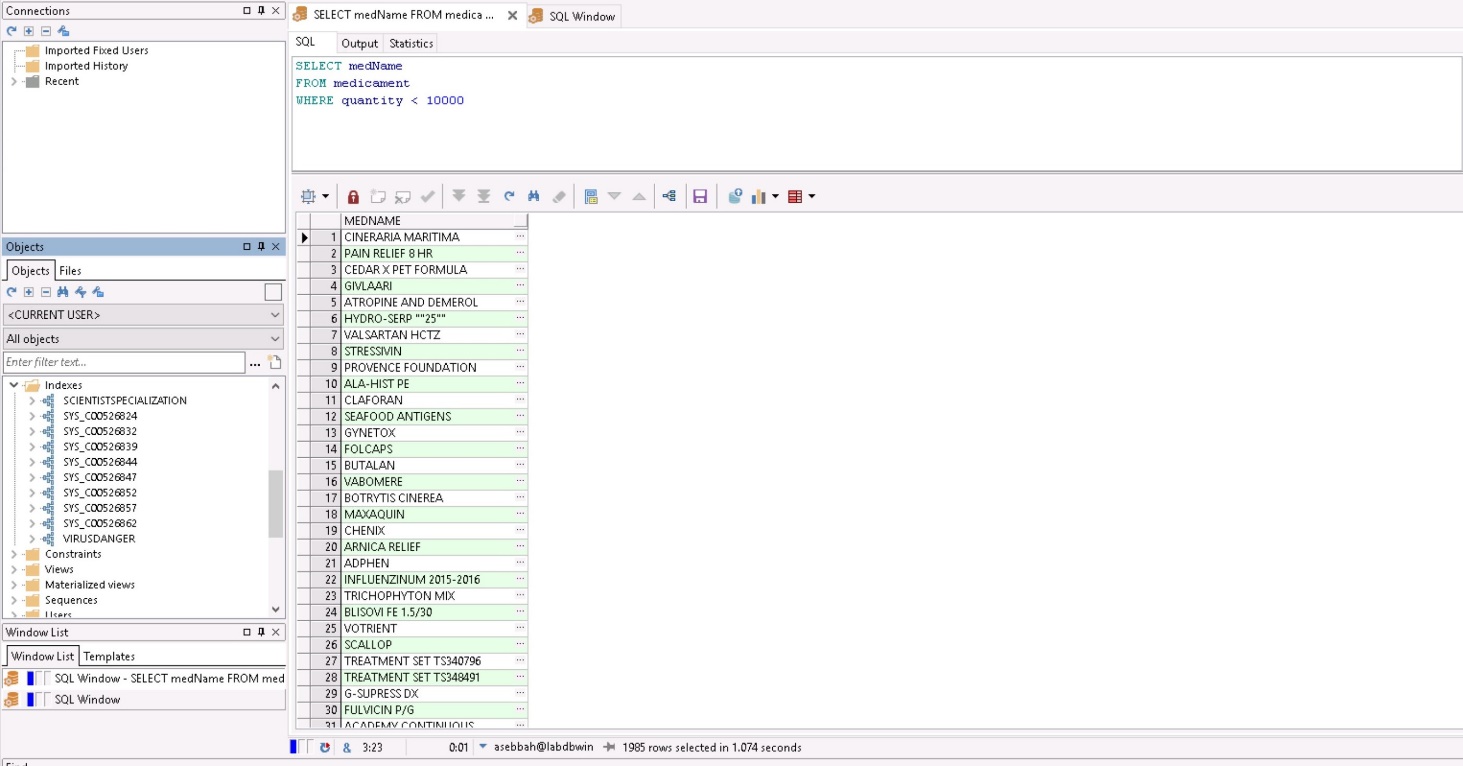
1. שאלנו שאילתה המדפיסה את כל התרופות שהכמות שלהם עברה סף הקריטי..

* 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



# 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,

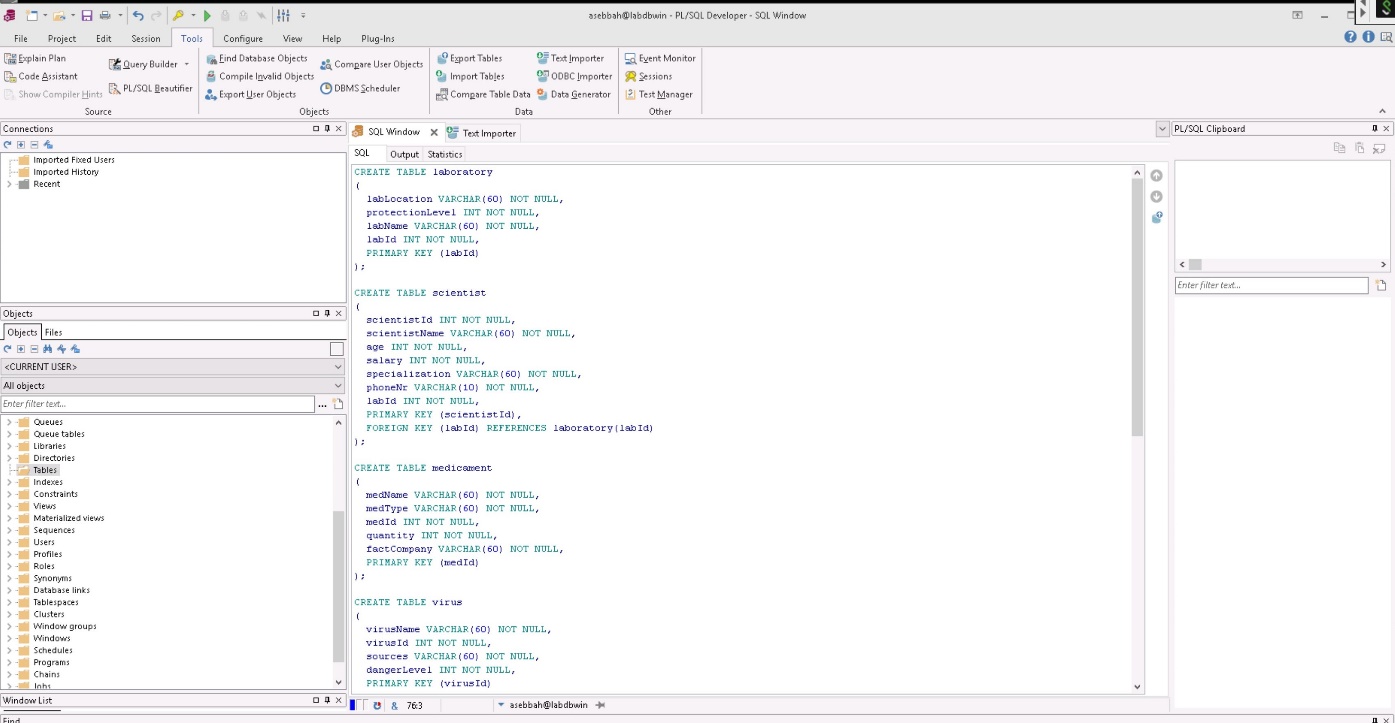
medId INT NOT NULL,

PRIMARY KEY (scientistId, medId),

FOREIGN KEY (scientistId) REFERENCES scientist(scientistId),

FOREIGN KEY (medId) REFERENCES medicament(medId)

) ;



## נספח שני: 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%'