<u>- פרויקט בבסיסי נתונים בנושא מכירה וקניית דירות</u> חברת <u>חזון</u>

מגישות: נעה אייזן מגני 207568965

ונחמה ברון 328958830



<u>תוכן ענייני ם</u>

מבוא-תיאור המערכת	
3ERD	
4	4
5 טבלאות	Į
7 desc פקודת	-
הכנסת נתונים לטבלה	
גיבוי ושיחזור	
14 ללא הפרמטרים select השאילתות של ה	
שאילתות מחיקה	1
שאילתות עדכון	2
שאילתות עם פרמטרים	
29	
מכנות: פונקציה ופרוצדורה א	3
94 ופרוצדורה ב	34

: מבוא-תיאור המערכת

המערכת המתוארת היא מערכת לניהול ומכירת נדל"ן, המיועדת לחברת תיווך המעסיקה מספר רב של מתווכים ופועלת במכירת דירות בכל ערי הארץ. הנתונים העיקריים שמאוחסנים במערכת כוללים:

- 1. בעלי דירות: מידע על בעלי הדירות המעוניינים למכור את נכסיהם, כולל פרטים כמו מספר זיהוי, שם מלא ופרטי יצירת קשר.
- 2. קונים: פרטי לקוחות המעוניינים לרכוש דירה, כולל פרטי יצירת קשר והעדפות נדל"ן.
- 3. מתווכים: נתונים על המתווכים המפעילים במערכת, כולל פרטים אישיים ופרטי יצירת קשר.
- 4. ערים: מידע על הערים בהן נמצאות הדירות הנמכרות, כולל דירוג ומידע כללי על הערים.
- 5. דירות: פרטי הנכסים הנמצאים למכירה, כולל מספר חדרים, מחיר, גודל, קומה וכתובת.
 - 6. פגישות :(VISIT) רישום פגישות לביקור בדירות עם המתווכים והלקוחות, כולל תאריך ושעה של הפגישה.
- 7. מכירת דירה: רישום עסקאות מכירת הדירות, כולל מידע על המכירה עצמה כמו מספר סידורי, עמלת סוכן ותאריך חתימה.

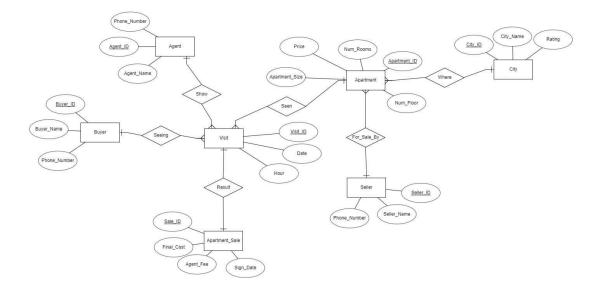
פונקציונליות המערכת כוללת:

- 1. ניהול נתונים: הוספה, עריכה ומחיקה של נתוני בעלי דירות, קונים, מתווכים, ערים ודירות במערכת.
- 2. קישור ותיאום: קישור בין בעלי הדירות, הקונים והמתווכים, ותיאום פגישות לביקור בדירות.
 - 3. רישום ומעקב: רישום פגישות ועסקאות מכירה, ומעקב אחר תהליכי המכירה.
- 4. דיווח וניתוח: יצירת דוחות וניתוחים על פעילות המכירה והתפלגות הדירות לפי מספר פרמטרים, לדוגמה, לפי עיר או סוג דירה.

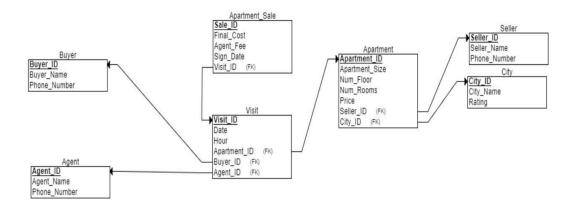
תיאור מילולי של טבלאות:

```
Seller(Seller_ID, Seller_Name,phone_Number)
Buyer(Buyer_ID, Buyer_Name, phone_Number)
Agent(Agent_ID,Agent_Name,phone_Number)
City(City_ID, City_Name,rating)
Apartment(Apartment_ID, Num_Rooms, Price, Apartment_Size, Num_Floor,
Seller_ID) Visit(Visit_ID, Date, Hour)
Apartment_Sale(Sale_ID, Final_Cost, Agent_Fee, Sign_Date)
```

-ERD



-DSD



פקודת יצירת טבלאות:

```
CREATE TABLE Seller
 Seller ID NUMERIC(3) NOT NULL,
Seller_Name VARCHAR(15),
 Phone Number NUMERIC(5),
 PRIMARY KEY (Seller ID)
);
CREATE TABLE Buyer
 Buyer ID NUMERIC(3) NOT NULL,
 Buyer Name VARCHAR(15),
 Phone_Number NUMERIC(5),
 PRIMARY KEY (Buyer ID)
);
CREATE TABLE City
 City ID NUMERIC(3) NOT NULL,
 City_Name VARCHAR(15),
 Rating NUMERIC(3),
 PRIMARY KEY (City_ID)
);
CREATE TABLE Agent Person
 Agent_ID NUMERIC(3) NOT NULL,
 Agent Name VARCHAR(15),
 Phone Number NUMERIC(5),
 PRIMARY KEY (Agent_ID)
);
CREATE TABLE Apartment_Sale
 Sale ID NUMERIC(3) NOT NULL,
 Final Cost NUMERIC(10),
 Agent Fee NUMERIC(6),
 Sign_Date DATE,
 Visit ID NUMERIC(3),
 PRIMARY KEY (Sale_ID),
 FOREIGN KEY (Visit ID) REFERENCES Visit(Visit ID)
CREATE TABLE Apartment
 Apartment_ID NUMERIC(3) NOT NULL,
 Apartment Size NUMERIC(3),
 Num Floor INT,
 Num Rooms INT,
 Price INT,
 Seller_ID NUMERIC(3),
5
```

```
City ID NUMERIC(3),
 PRIMARY KEY (Apartment_ID),
 FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID),
 FOREIGN KEY (City_ID) REFERENCES City(City_ID)
);
CREATE TABLE Visit
 Visit_ID NUMERIC(3) NOT NULL,
 Date of visit DATE,
 Hour_of_visit NUMERIC(2),
 Apartment ID NUMERIC(3),
 Buyer ID NUMERIC(3),
 Agent_ID NUMERIC(3),
 PRIMARY KEY (Visit ID),
 FOREIGN KEY (Apartment_ID) REFERENCES Apartment(Apartment_ID),
 FOREIGN KEY (Buyer ID) REFERENCES Buyer(Buyer ID),
 FOREIGN KEY (Agent ID) REFERENCES Agent Person(Agent ID),
UNIQUE (Date_of_visit)
);
```

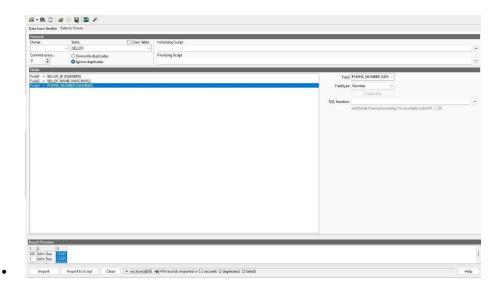
<u>:decs - פקודת</u>

SQL> desc	Buyer					
Name	Type		Nullable	Default	Comments	
A CONTRACTOR OF THE PARTY OF TH	NUMBER (3)					
BUYER_NAME	VARCHAR2 (15)	Y			
PHONE_NUME	BER NUMBER(5)	NUMBER (5)		Y		
SQL> desc	Seller					
Name	Type		Nullable	Default	Comments	
		000			5000000	
	NUMBER (3)					
SELLER_NAM	ME VARCHAR2 (VARCHAR2 (15)		Y		
PHONE_NUME	BER NUMBER (5)	NUMBER (5)		Y		
SQL> desc	Agent_Person					
Name	Type		Nullable	Default	Comments	
AGENT ID	NUMBER (3)					
		VARCHAR2 (15)				
THE RESERVE AND ADDRESS OF THE PARTY OF THE	BER NUMBER (5)					
SQL> desc						
Name	Type	Nu.	llable De:	fault Con	mments	
		1000				
CITY_ID	NUMBER (3)					
CITY_NAME	VARCHAR2 (15)	Y				
RATING	NUMBER (3)	Y				

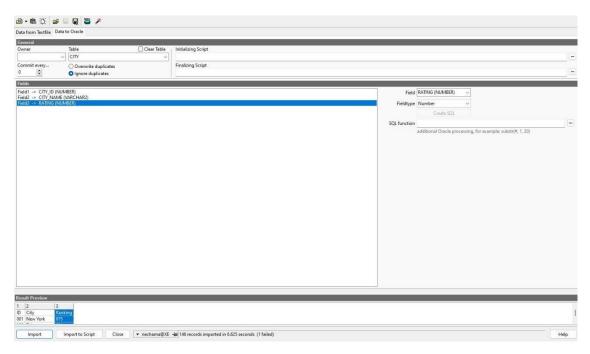
הכנסת נתונים לטבלאות:

:השתמשנו ב3 שיטות

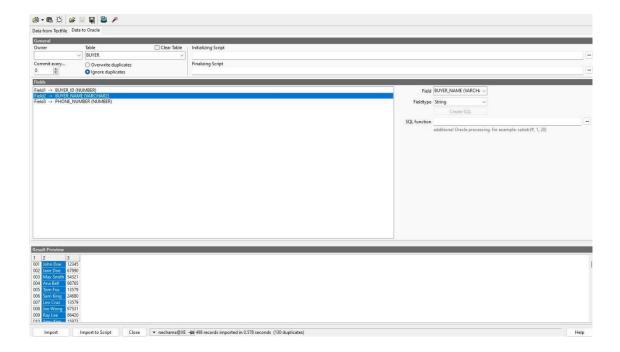
1. הכנסת נתונים מתוך קובץ טקסט, הכנסנו לתוך טבלאות לפחות 400 נתונים לטבלת של SELLER:



הכנסנו גם לטבלת CITY כ100 נתונים לפחות:

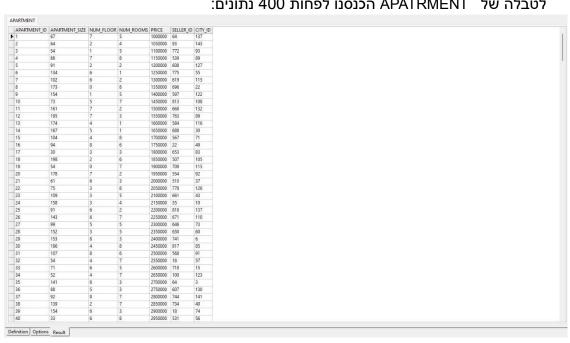


הכנסנו לתוך טבלאות לפחות 400 נתונים לטבלת של BUYER :



• דרך נוספת שהכנסנו נתונים זה עם ב dataGenerator

לטבלה של APATRMENT הכנסנו לפחות 400 נתונים:



: APATRMENT _SALE לטבלה של



PERSON_ AGENT הכנסנו לפחות 400 נתונים לטבלה של



VISIT הכנסת נתונים לטבלה של

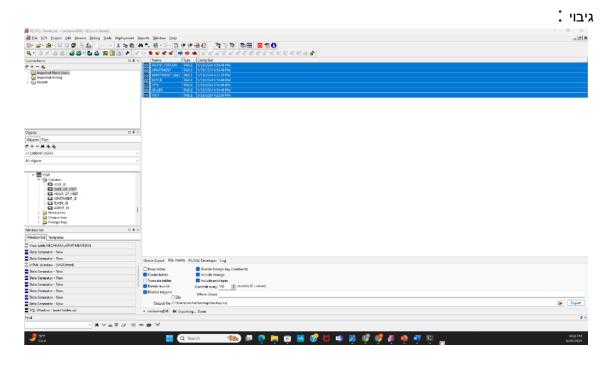


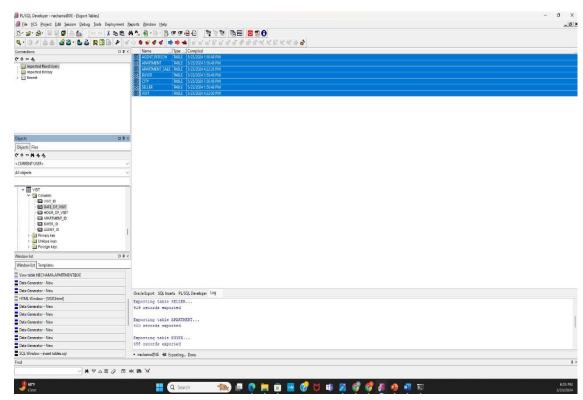
לטבלה של INSERT הכנסנו נתונים בעזרת פקודת + obsect לטבלה של BUY

```
SOL> insert into Buyer values(666,'noa edr',93848);
1 row inserted
SOL> insert into Buyer values(800,'rachel yajo',93458);
1 row inserted
SOL> insert into Buyer values(900,'rachel yayay',94458);
1 row inserted
SOL>
```

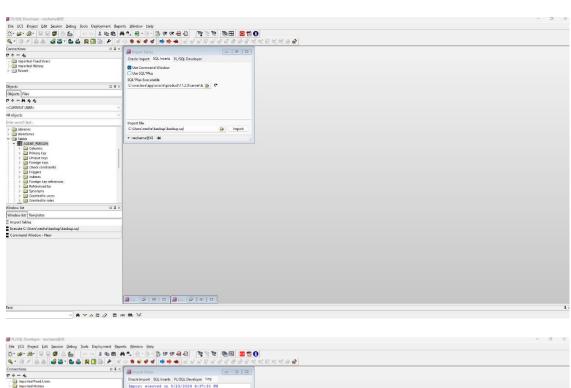
המרצה בהוראות שכתבה לכיתה אמרה שרק 4 טבלאות יהיו עם לפחות 400 נתונים .

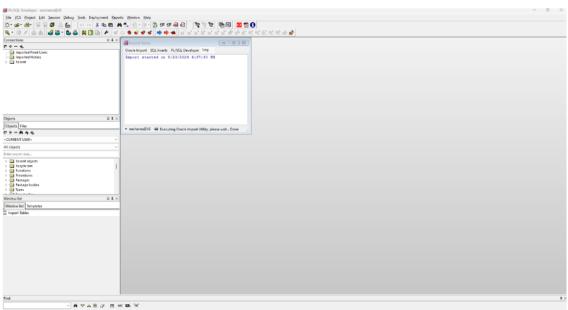
גיבוי ושחזור:





<u>שיחזור:</u>





שאילתות SELECT ללא פרמטרים:

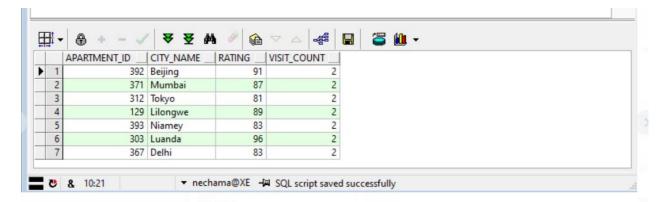
1. שאילתה המציגה דירות מעיר עם דירוג 80 שביקרו בה יותר מפעמיים. הרצה:

```
- - X

SQL Window - SelectApartmentWithRating80AND5Visits.sql

SQL
     Output Statistics
SELECT A.Apartment_ID, C.City_Name, C.Rating, V.Visit_Count
                                                                                                 ŵ
FROM Apartment A
                                                                                                 4
JOIN City C ON A.City_ID = C.City_ID
JOIN (
    SELECT Apartment_ID, COUNT(*) AS Visit_Count
    FROM Visit
    GROUP BY Apartment_ID
    HAVING COUNT (*) >= 2
) V ON A.Apartment ID = V.Apartment ID
WHERE C.Rating > 80;
```

<u>תוצאה:</u>



2. שאילתא המייצגת את כל הסוכנים שהיו ביותר מפגישה.

<u>הרצה:</u>

```
SQL Window - SelectAgentWithMinOneVisit.sql

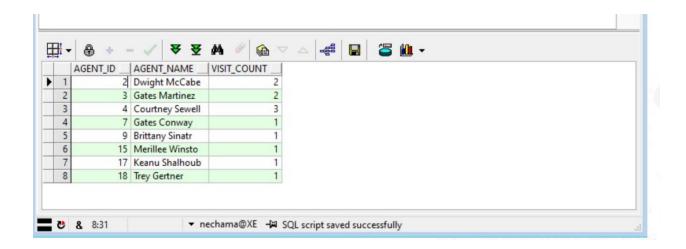
SQL Output Statistics

SELECT A.Agent_ID, A.Agent_Name, V.Visit_Count
FROM Agent_Person A

JOIN (
SELECT Agent_ID, COUNT(*) AS Visit_Count
FROM Visit
GROUP BY Agent_ID
HAVING COUNT(*) >= 1

) V ON A.Agent_ID = V.Agent_ID
```

<u>תוצאה:</u>



3. שאילתא select המציגה את מספר הפגישות שהיו עבור כל דירה לפני שהיא נקנתה מסודרת לפי המספר המזהה של הדירה.

SQL Window - SELECT A.Apartment_ID, Apartment_Sale.Final_Cost, Apartment_Sale.Sign_Date, VC.Visit_Count FROM ...

SQL Output Statistics

SELECT A.Apartment_ID, Apartment_Sale.Final_Cost, Apartment_Sale.Sign_Date, VC.Visit_Count FROM Apartment_Sale

JOIN Visit ON Apartment_Sale.Visit_ID = Visit.Visit_ID

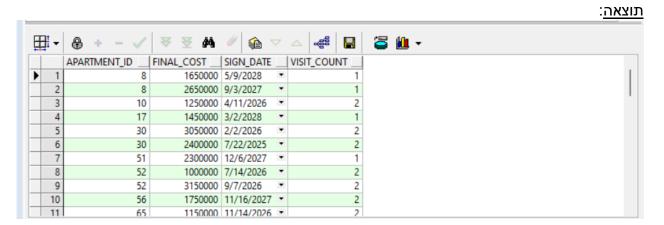
JOIN Apartment A ON Visit.Apartment_ID = A.Apartment_ID

JOIN (

SELECT Apartment_ID, COUNT(*) AS Visit_Count
FROM Visit
GROUP BY Apartment_ID

) VC ON A.Apartment_ID = VC.Apartment_ID

ORDER BY A.Apartment_ID;



4. שאילתה המייצגת את הביקורים בתווך מסויים מסודרים לפי תאריך. <u>הרצה:</u>

```
SELECT

V.Visit_ID,

V.Apartment_ID,

V.DATE_OF_VISIT,

C.City_Name

FROM

Visit V

JOIN

Apartment A ON V.Apartment_ID = A.Apartment_ID

JOIN

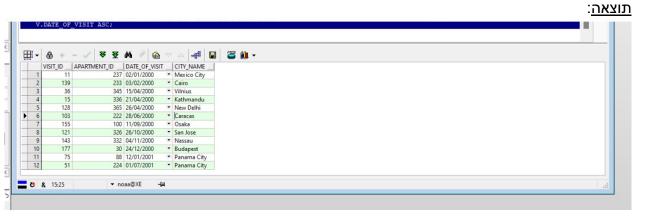
City_C ON A.City_ID = C.City_ID

WHERE

V.DATE_OF_VISIT_BETWEEN TO_DATE('1986-01-01', 'YYYY-MM-DD') AND TO_DATE('2029-12-31', 'YYYY-MM-DD')

ORDER BY

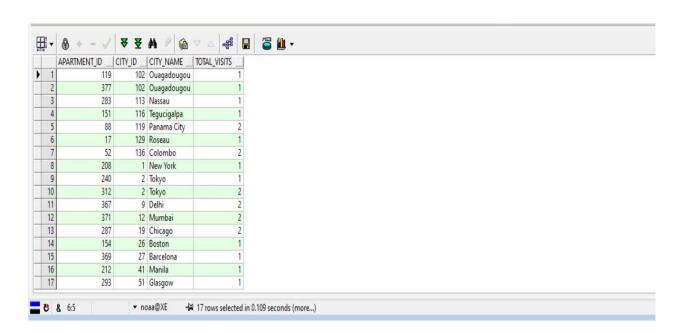
V.DATE_OF_VISIT_ASC;
```



. .5

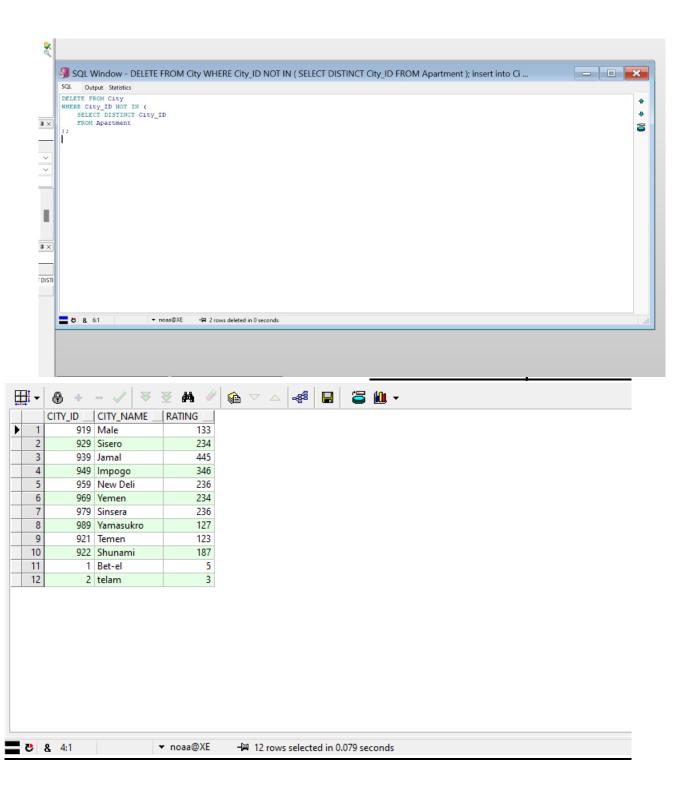
שאילתה של כמה פעמים היום בכל דירה ומחזיר נתונים של איזה עיר זה. <u>הרצה</u>:

<u>תוצאה:</u>

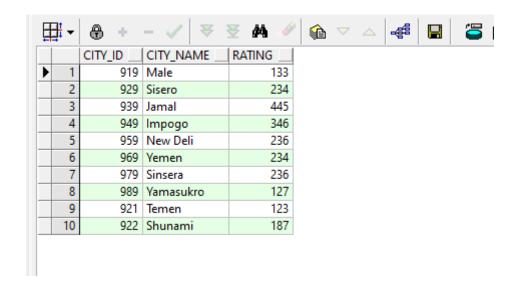


שאילתות מחיקה:

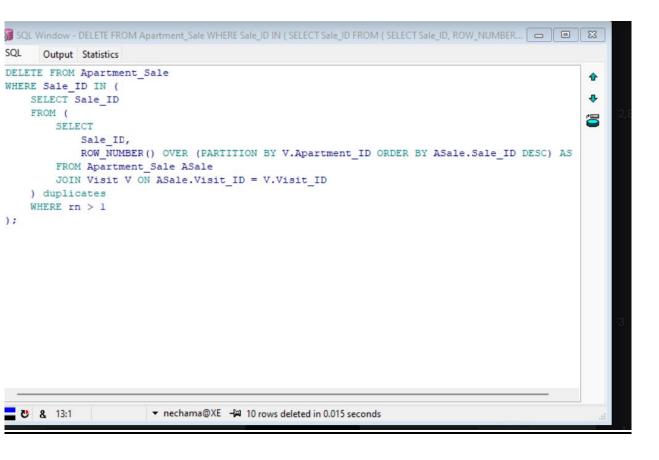
שאילתה שהיא מוחקת ערים שאין להם דירות במאגר. <u>הרצה</u>:



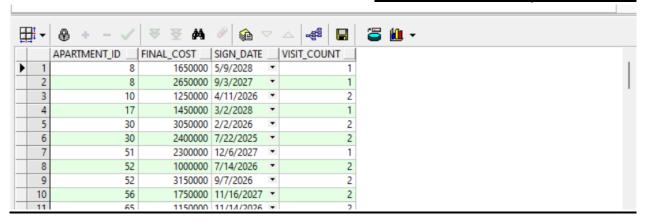
לאחר המחיקה הבסיס נתונים:



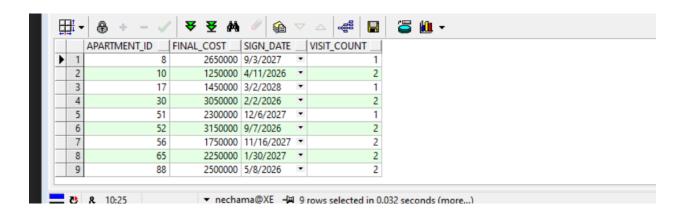
שאילתה מחיקת נתוני הדירות של יהיו כפולים הרצה:



לפני המחיקה הבסיס נתונים:



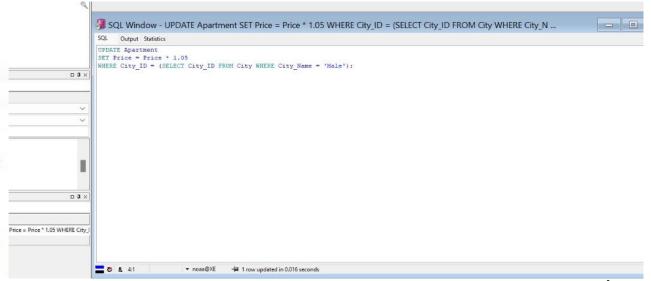
לאחר המחיקה הבסיס נתונים:



:שאילתות עדכון

המכירים בעיר מסויים עלו - כל המחירים בעיר מתעדקנות להיות 5% יותר

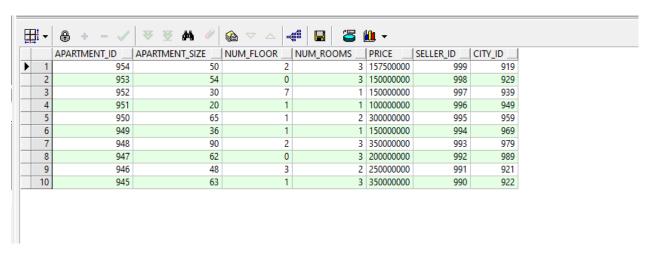
הרצה: לדוגמא עדכון מחיר של דירה מלין



לפני עדכון בסיס הנתונים:



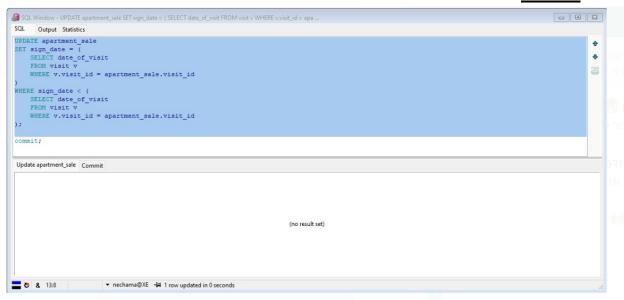
<u>לאחר עדכון בסיס הנתונים:</u>



שאילת עדכון:

אם תאריך הביקור היה אחרי תאריך החתימה, תאריך החתימה מתעדכן להיות אותו תאריך של הביקור. (זאת אומרת שהם ביקרו וחתמו באותו יום)

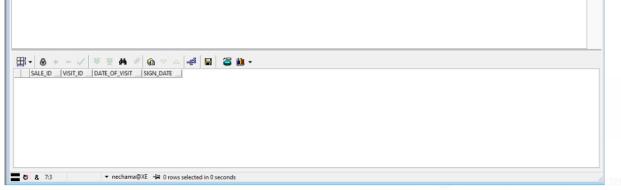
:הרצה



לפני עדכון בסיס הנתונים:



<u>אחרי עדכון בסיס הנתונים:</u>



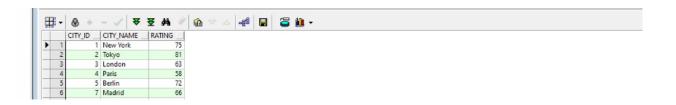
שאילתות עם פרמטרים:

1. שאילתה שמציגה את הערים לפי דירוג שהמשתמש מכניס.

<u>:הרצה</u>



תוצאה:



.2

שאילתה המציגה החל מתאריך מסויים את כל הפגישות שבהם קיבלו מתנה מסויימת. הוספנו שדה בשם PRESENT_STATUS בעזרת ALTER_TABLES.

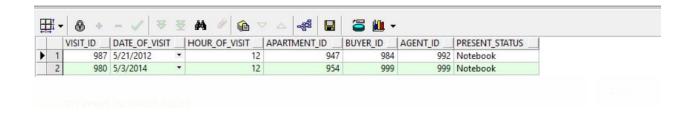
<u>:הרצה</u>

```
SELECT * FROM Visit

WHERE date_of_visit >= TO_DATE('&visit_date', 'YYYY-MM-DD')

AND present_status = &<name = p_status type=String list="Notebook, Pen, Nothing" hint="Enter status of Present" required=true>;
```

:תוצאה



שאילתה שמציגה את הדירות מעיר מסויים מתוך 3 ערים הכי פופלריים לפי בחירת המשתמש ועם מחיר החל ממחיר שמצויין על ידי המשתמש

הרצה:

```
SELECT City.City_ID, City_Name, Apartment_ID, Apartment_Size, Num_Floor, Num_Rooms, Price
FROM City
JOIN Apartment ON City_ID = Apartment.City_ID
AND City_Name = &<name = city_name type=String list="New Deli, Jamal, Sisero" hint="Enter city" required=true>
AND Price > &<name = min_price type=integer hint="Enter minimum" required=true>;
```

תוצאה:



.4

שאילתא שמציגה את הדירות של מוכר מסויים לפי פרמטר של המשתמש.

:הרצה



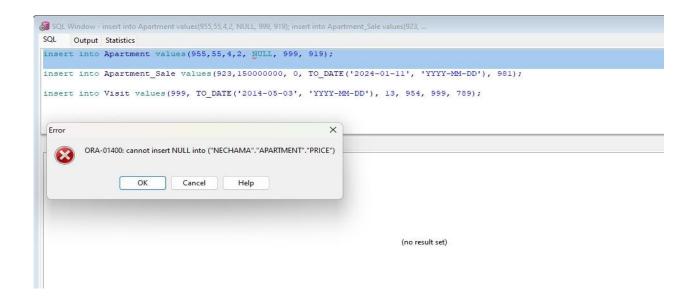
■ 8 5:1 ▼ noaa@XE → 1 row selected in 0.172 seconds

<u>אילוצים</u>:

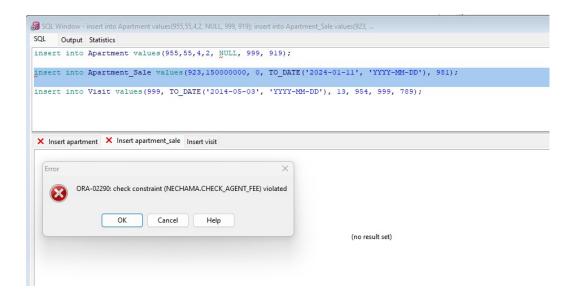
הוספנו את האילוצים הבאים:



בטבלה Apartment: המחיר אינו יכול להיות NULL



בטבלה Apartment_Sale: עמלת תיווך של הסוכן חייב להיות גדול מ0

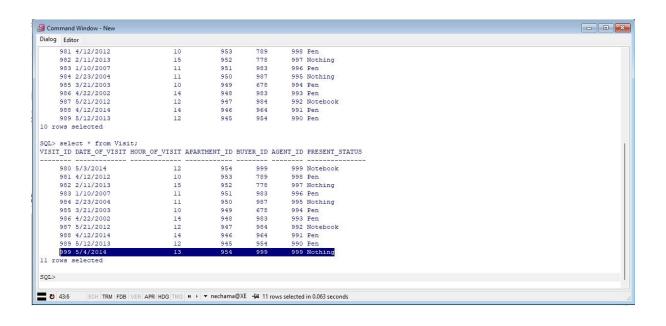


:Visit בטבלה

ברירת מחדל עבור העמודה Present_Status המייצת את המתנה שנתנו לקונה הוא כלום – 'Nothing'. ניתן לראות שעכשיו אפשר להכניס נתונים בלי לציין מה יש בעמודה הזאת ובפועל מוכנס ערך 'nothing'.

insert into Visit (Visit_Id, Date_Of_Visit, HOUR_OF_VISIT, Apartment_Id, Buyer_Id, Agent_Id)
values(999, TO_DATE('2014-05-04', 'YYYY-MM-DD'), 13, 954, 999, 999);



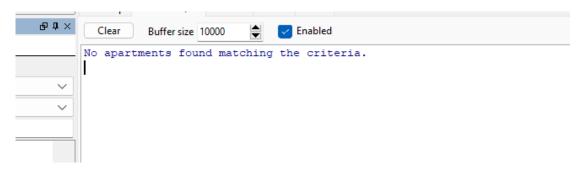


שלב 3: תכנות: פונקציה ופרוצדורה א

פונקציה שמביאה דוח על דירות שלא נמכרו בעיר לפי נתונים שמשתמש מכניס כמו שם עיר , טווח חדרים או טווח מחירים:

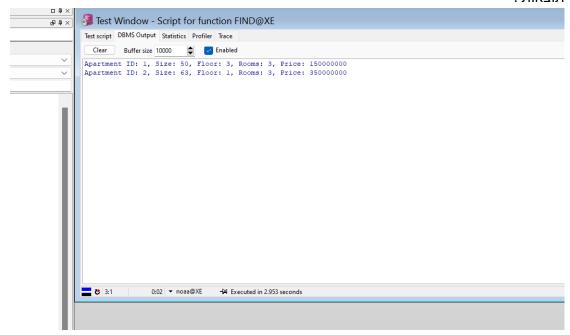
תוצאות:

אין דירות בהתאם למה שמשתמש הכניס:



המשתמש הכניס נתונים:

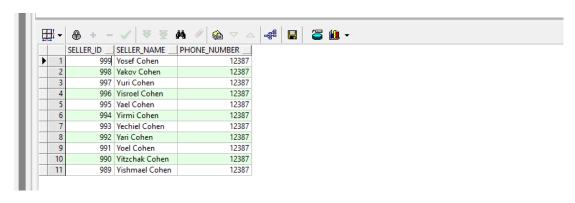
: תוצאות



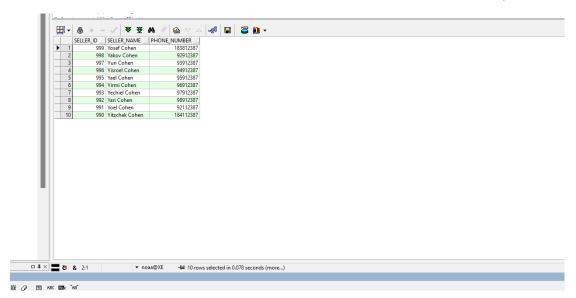
הוספנו עמודה לטבלה של דירה של האם הדירה נמכרה או לא.

פרוצודה שמעדכנת את מספרי הטלפון לפי קידומת של העיר:

בסיס הנתונים לפני עדכון:



אחרי עדכון עם פרוצדורה:

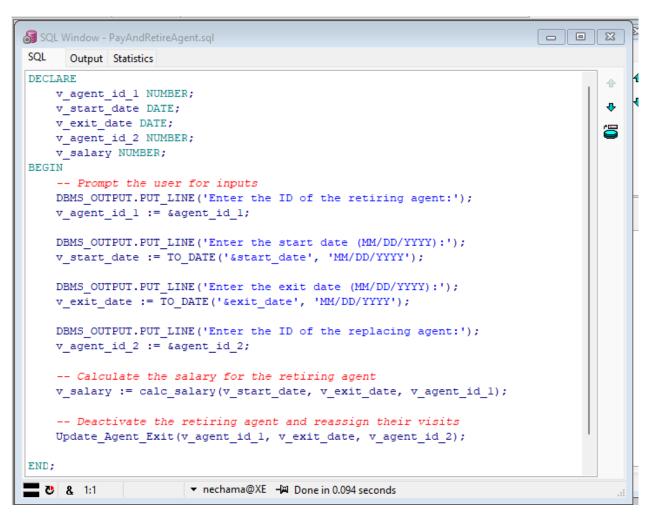


השתמשנו פה בעדכון גודל מספר הטלפון

פונקציה ופרוצדורה ב

בתוכנית הראשית, המשתמש מזין את מספר של הסוכן שיוצא לפנסיה, את התאריך הראשון שממנו צריך לשלם לסוכן, ואת תאריך הפרישה. המשתמש גם מזין את המספר של הסוכן שיחליף אותו בפגישות שנקבעו לתאריכים שהם אחרי תאריך הפרישה. התכנית מעדכנת שהוא כבר לא פעיל, מחשבת את שכרו עבור תקופת עבודתו ומעבירה את הפגישות שנקבעו לתאריכים אחרי תאריך הפרישה לסוכן אחר.

תוכנית ראשית:



חישוב השכר:

הסוכן מקבל שכר קבוע עבור כל פגישה בתקופת עבודתו

הסובן מקבל אחוז מהעמלה לחברת תיווך עבור כל מכירה שנחתמה התקופת עבודתו

החישוב נעשה בעזרת 2 פונקציות:

מחשב את השכר עבור פגישות Calc visit salary

מחשב את השכר עבור מכירות Calc sale salary

הפונקציה calc_Salary קוראת לשניהם.

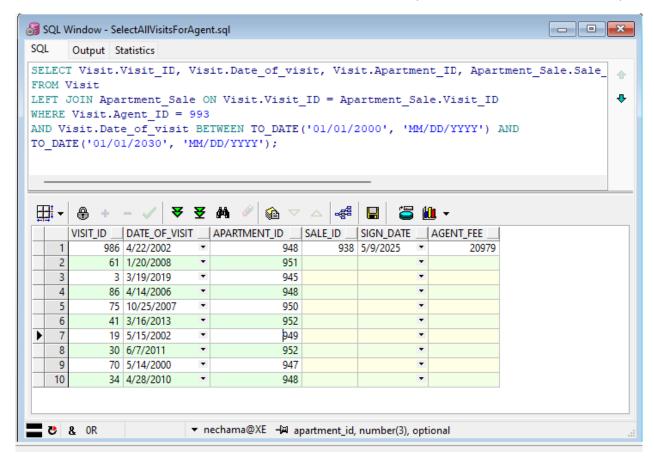
התוצאות מודפסות למסך.

עדכון היציאה לפנסיה:

הפרוצדורה Update_Agent_Exit מעדכנת את הסוכן להיות לא פעיל ומעבירה את הפגישות שלו לסוכן אחר. הפרוצדורה בודקת שהסוכן המחליף הוא באמת פעיל, אחרת הפגישות אינם מועברים לסוכן החדש.

דומגת הרצה:

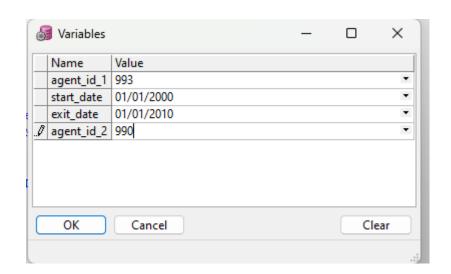
סוכן 993 יוצא לפנסיה בשנה 2010. כאן טבלת הפגישות ומכירות שלו לפני שעדכנו את בסיס הנתונים:



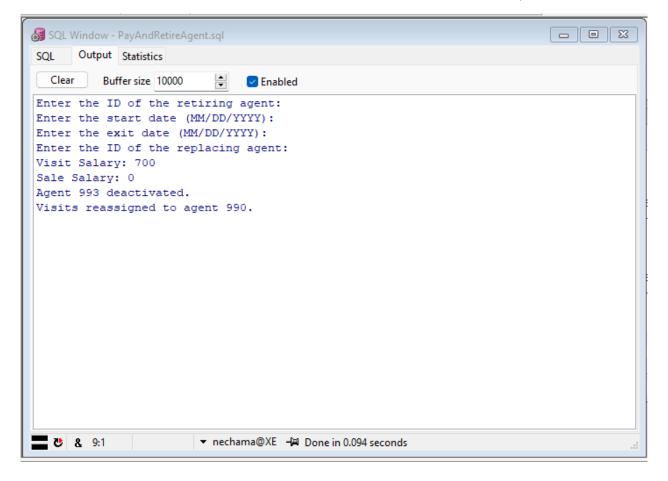
וכאן ניתן לראות שהוא סוכן פעיל:

נרצה לשלם לו עבור פגישות ומכירות שהתקיימו לפני 2010. הפגישות שנקבעו לתאריכים אחרי 2010 יועברו לסוכן 990.

הזנת נתונים:

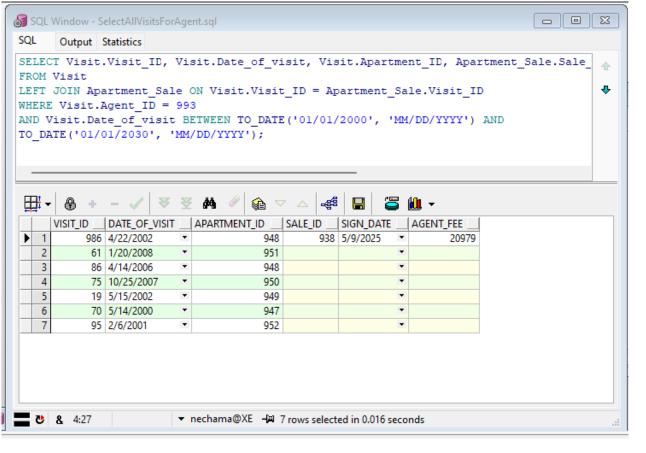


התוצאות הודפסו למסך:



השכר שמגיע לסוכן עבור תקופת עבודתו היא 700 על 7 פגישות. אין מכירות שהתקיימו בזמן הזה. הפגישות שנקבעו לתאריכים אחרי שהוא פרש עברו לסוכן 990.

כאן הטבלה המעודכנת של סוכן 993:



ניתן לראות שאין לו פגישות אחרי 2010. הוא גם התעדבן להיות לא פעיל:

הפגישות שאחרי 2010 עברו לסוכן 990:

