

דו"ח פעילות- השכרת רכב  
מיני פרויקט בבסיסי נתונים  
יעקב אברמוביץ  
חיים מילצקי 203077326

## תוכן עניינים

דו"ח פעילות- השכרת רכב.....	1
תיאור.....	3
RED.....	4
CREATE TABLE: .....	6
יצירת נתונים .....	8
שאלות .....	9
אינדקסים .....	17
אינטגרציה.....	23
VIEW'S.....	25
תרשימים .....	27
פרוצדורות.....	29
פונקציות .....	31

## תיאור

אנו בונים מסד נתונים עבור חברה להשכרת רכבים. כחלק מבנית הפרוייקט

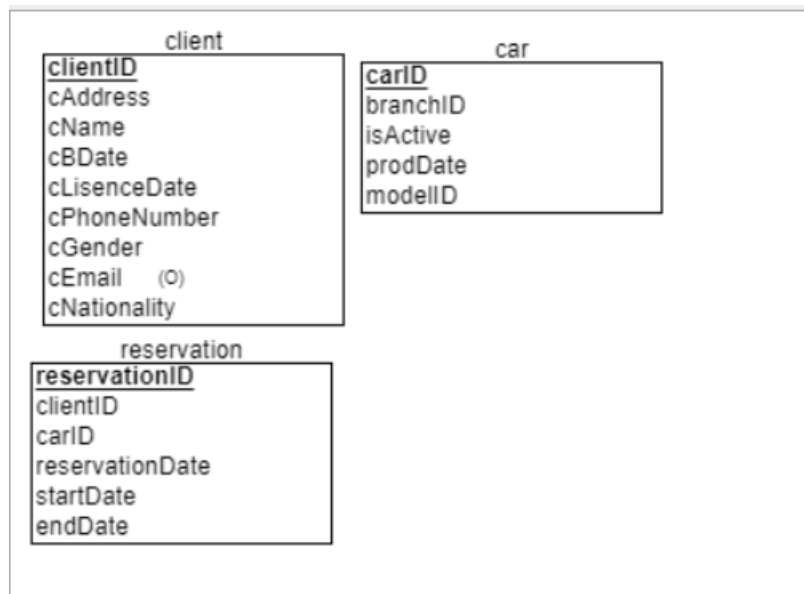
נדרשנו לאפיין שלושה ישויות:

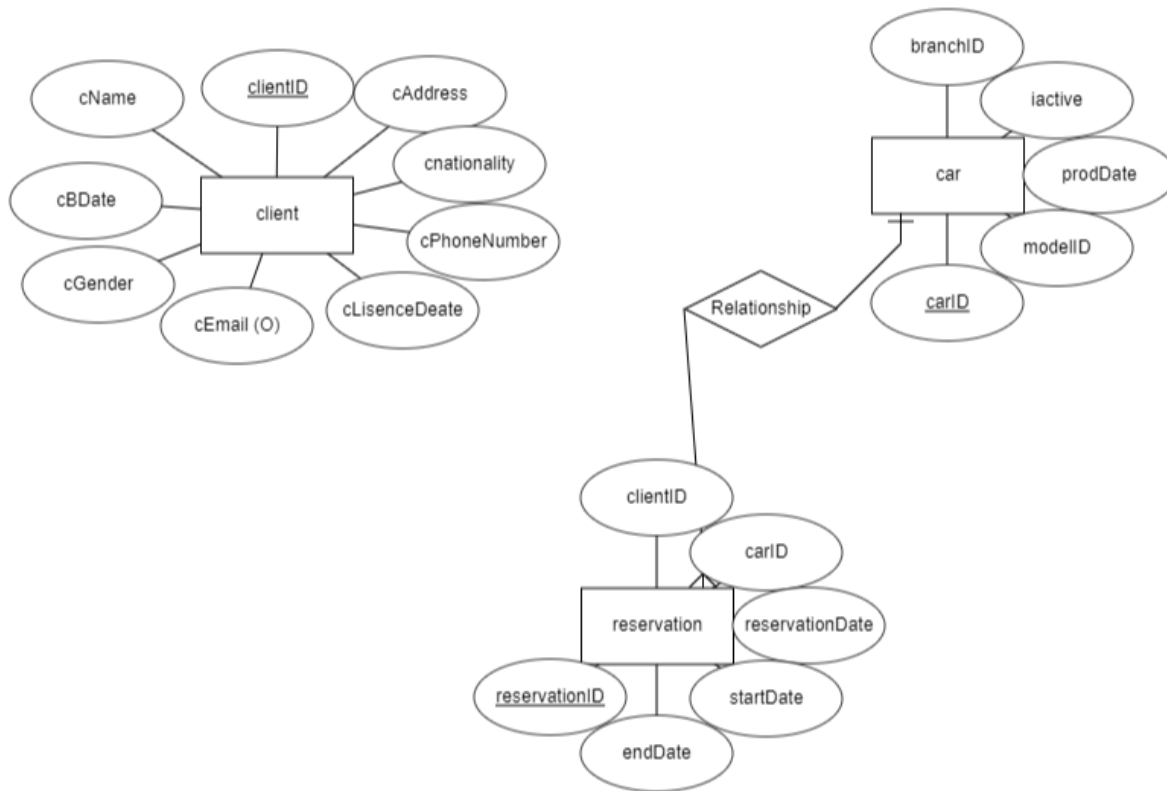
ישות 1: CAR (רכב)

ישות 2: CLIENT (לקוח)

ישות 3: RESERVATION (הזמנת רכב בחברה)

RED





## :CREATE\_TABLE

### סכמה עבור לקוח:

```
CREATE TABLE client
)
clientID NUMERIC(9) NOT NULL,
cAddress VARCHAR(50) NOT NULL,
cName VARCHAR(20) NOT NULL,
cBDate DATE NOT NULL,
cLicenceDate DATE NOT NULL,
cPhoneNumber NUMERIC(9) NOT NULL,
cGender CHAR(1) NOT NULL,
cEmail VARCHAR(25)
cNationality VARCHAR(10) NOT NULL,
PRIMARY KEY (clientID)
; (
```

### סכמה עבור הישות "רכב":

```
CREATE TABLE car
)
branchID NUMERIC(6) NOT NULL,
isActive CHAR(1) NOT NULL,
prodDate DATE NOT NULL,
modelID NUMERIC(20) NOT NULL,
carID NUMERIC(10) NOT NULL,
PRIMARY KEY (carID)
; (
```

**סכמה עבור הזמנות:**

```
CREATE TABLE reservation
)
clientID NUMERIC(9) NOT NULL,
carID NUMERIC(10) NOT NULL,
reservationDate DATE NOT NULL,
startDate DATE NOT NULL,
endDate DATE NOT NULL,
reservationID NUMERIC(15) NOT NULL,
PRIMARY KEY (reservationID)
;{
```

## יצירת נתונים

חלק קוד שכתבנו ליצירת פייק דאטה

```
from faker.providers import BaseProvider
from faker import *
import csv
import random

# I don't know what is wrong but it fail at running here but at colab it run successfully

class NationalityProvider(BaseProvider):
    def nationality(self):
        return random.choice(['USA', 'ASIAN', 'ITALY', 'SPAIN', 'EURO_UNION', 'JAPAN', 'ISRAEL'])

class GenderProvider(BaseProvider):
    def gender(self):
        return random.choice(['F', 'M'])
```

## [קישור לקוד](#)

יצרנו מסמכי CSV בעזרת הסקריפט  
את התוצר הכנסנו לבסיס הנתונים



## שאלות

מעניין לדעת מי הלקוחות היותר קבועים של החברה

שאלתא 1:

כל הלקוחות שביצעו יותר מ 3 הזמנות

נבחרו 408 שורות ב0.389 שניות

SQL

Output

Statistics

```
select *
from client
where clientid in ( select distinct clientid
from client natural join reservation
group by clientid
having count(*)>3
)
```

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

Statistics

SQL

Output

כחלק ממחקר שנערך בחברה, כמה מהלקוחות באמת מעוניינים ברכב חדש. אנו מעוניינים לדעת כמה אחוזים מכלל הלקוחות אכן שמים דגש על כך

**שאלתא 2: כל הלקוחות ששכרו רכב ששנת הייצור שלו קדמה לתאריך 15/7/2005**

The screenshot shows a SQL query executed in a database client. The query filters clients based on the date of their last rental. The results are displayed in a table with the following columns: CLIENTID, ADDRESS, CNAME, CBDATE, CLISENCEDATE, and CPH.

	CLIENTID	ADDRESS	CNAME	CBDATE	CLISENCEDATE	CPH
20	302	2514 Burke AlleyShepardmouth, CA 69845	Steven White	22/07/1971	02/05/1995	
21	306	236 Kenneth TerraceLaneland, ME 91195	Michelle Kelly	14/01/1991	07/01/1981	
22	308	4013 Stephanie TraceStevenport, OK 22583	Tammy Salazar	23/04/1953	01/01/1960	
23	309	PSC 9429, Box 8161APO AE 27922	Robert Flowers	24/03/1971	18/02/1943	
24	311	28998 Christopher FieldSummersville, KY 03480	Natasha Thomas	05/03/1944	03/04/1974	
25	312	0309 Washington ViaSouth Craig, LA 60071	Jeffrey Nelson	28/07/1999	30/09/1935	
26	313	6995 Huff ForgeLake Kaitlyn, WA 51647	Laura Sanchez	17/03/1972	29/10/1970	
27	314	PSC 5113, Box 6103APO AE 30872	Jonathan Mcdonald	13/03/1935	20/02/1937	
28	318	5838 Daniel Points Apt. 470Fordville, ID 59138	Catherine McGee	05/03/1951	11/03/1994	
29	319	71671 Torres PlazaKarenchester, NY 17987	Timothy Clark	28/01/1982	15/02/1970	
30	320	26077 Melinda ValleysFergusonton, VT 67713	Aaron Moreno	31/12/1994	23/12/2002	
31	321	5858 Shaun Center Apt. 089Williamsfurt, DC 36969	Amy George	09/03/1949	05/12/1969	
32	322	77966 Chan SpurTaylorborrough, WY 52027	Debra Charles	30/12/1974	01/03/1959	
33	325	10630 Hector PortsOliviaborough, WA 66465	Ashley Shaffer	12/01/1958	30/01/1950	

At the bottom of the window, a status bar indicates: 11420 rows selected in 4.642 seconds.

### שאלתא 3: שמות ות.ז. הלקוחות שהזמינו רכב בטווח תאריכים מסוים

CREATE TABLE reservation ( ... select distinct cname ,cli ...

SQL Output Statistics

```
select distinct cname ,clientid
from reservation natural join client
where reservationDate> TO_DATE('01/01/2000', 'DD/MM/YY') and reservationDate< TO_DATE('01/01/2010', 'DD/MM/YY')
```

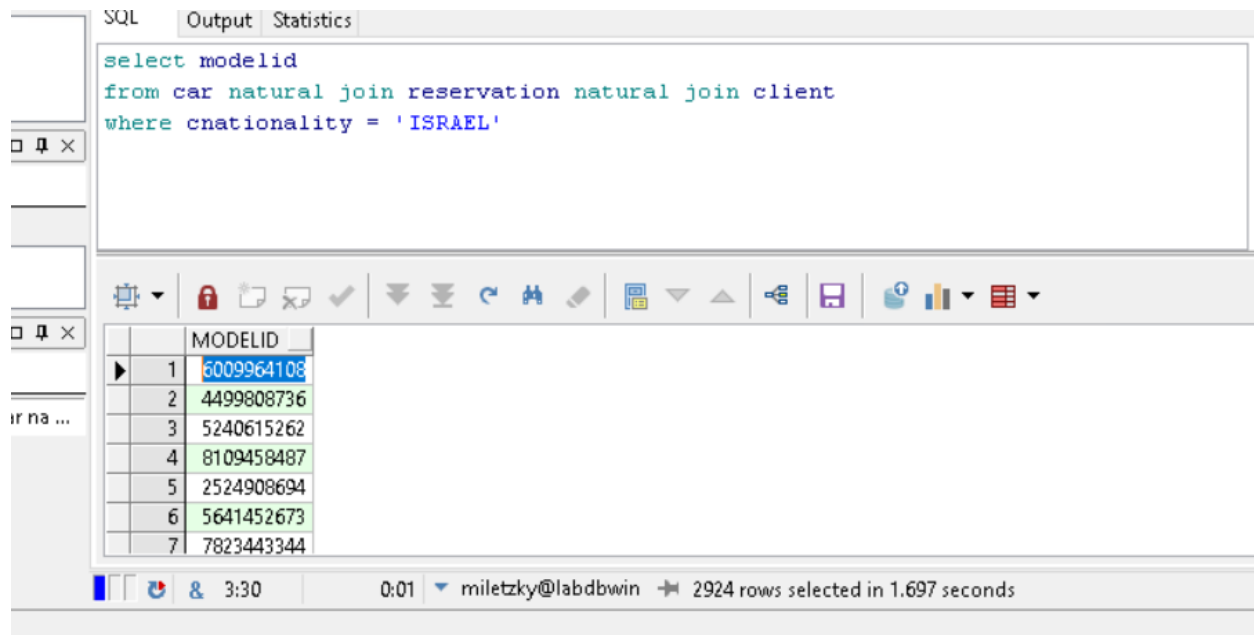
	CNAME	CLIENTID
2	Melissa Nguyen	292
3	Tammy Salazar	308
4	Adam Hughes	350
5	Linda Hunt	359
6	George Alvarez	9
7	Lisa Nelson	60
8	Renee Ferguson	102
9	Dr. Eric Todd	254
10	Jason Ramirez	442
11	Jason Freeman	555
12	Amanda Mathews	783
13	Peter Henson	819
14	Andrew Martinez	491
15	Melvin Cunningham	531
16	Abigail Lowe	310
17	Denise Murray	1147

3:32 miletzky@labdbwin 3556 rows selected in 0.821 seconds

אנו מעוניינים לדעת אלו מודלים נפוצים בקרב הלקוחות המתגוררים במדינת ישראל

שאלתא 4:

כל המודלים שהזמינו לקוחות במדינה פלונית



The screenshot shows a SQL query execution window with the following SQL query:

```
select modelid
from car natural join reservation natural join client
where cnationality = 'ISRAEL'
```

The results are displayed in a table with 7 rows and 1 column (MODELID):

	MODELID
1	6009964108
2	4499808736
3	5240615262
4	8109458487
5	2524908694
6	5641452673
7	7823443344

The status bar at the bottom indicates: 0:01 | miletzky@labdbwin | 2924 rows selected in 1.697 seconds

כחלק ממחקר שנערך על ידי מרשם האוכלוסין האמריקאי נדרשנו להעביר את שמות הלקוחות בארה"ב

### שאלתה 5:

#### שמות הלקוחות שגרים בUSA

The screenshot shows the PL/SQL Developer interface. The main window displays a SQL query: `select cname  
from client  
where cnationality = 'USA'`. Below the query, the results are shown in a table with 6 rows. The status bar at the bottom indicates that 2927 rows were selected in 0.703 seconds.

	CNAME
1	Rachel Erickson
2	Tommy Robinson
3	Tracey Davis
4	Morgan Estes
5	Jason Harmon
6	Robert Flowers

## שאלתה 6:

כל ההזמנות שהוזמנו בטווח תאריכים מסוים

SQL Output Statistics

```
select*
from reservation
where reservationdate >TO_DATE('1/1/2018' , 'DD/MM/YY') and reservationdate< TO_DATE('1/1/2022' , 'DD/MM/YY')
```

	CLIENTID	CARID	RESERVATIONDATE	STARTDATE	ENDDATE	RESERVATIONID
1	3877	1207	10/06/2019	15/02/1974	28/02/1998	703
2	13202	214	09/02/2019	02/10/2017	08/07/1975	723
3	9860	250	09/08/2020	07/10/2005	31/03/1989	724
4	1062	1296	27/09/2021	04/11/1999	20/09/1996	733
5	6149	535	03/02/2020	18/02/1980	21/09/2015	734
6	2211	1633	10/07/2021	29/05/2005	30/01/1995	768
7	14489	1662	05/09/2020	12/08/2012	15/12/2016	769
8	12207	561	04/04/2021	02/05/2002	18/01/1973	775
9	17601	580	25/04/2018	25/06/2009	25/08/1986	822
10	17548	84	03/08/2020	04/01/1993	01/10/1974	855
11	5359	523	31/07/2020	05/06/2010	09/01/1986	861
12	16409	1894	30/10/2018	20/07/2006	12/09/2004	862
13	19786	1848	11/01/2019	28/07/1985	13/09/2000	876
14	19052	252	26/10/2019	17/05/1985	15/03/2002	10
15	19086	751	18/11/2018	09/06/2004	05/08/2009	36
16	17621	1579	27/06/2019	05/08/2020	26/07/2019	87
17	2601	998	25/10/2019	05/11/2011	14/09/2010	99
18	2633	294	12/02/2018	02/02/2005	28/03/1985	112
19	5600	747	10/02/2018	01/05/2000	07/01/2000	110

5:109 miletzky@labdbwin 1629 rows selected in 0.621 seconds

reserva  
TO\_DATE  
1/1/111  
select  
from cl  
where c  
select  
from cs  
where c  
select  
from re  
where r  
TO\_DATE

שאלתה 7:

כחלק מתוכנית התייעלות של החברה אנו מעוניינים לקבל את רשימת הרכבים הפעילים

```
select*
from car
where isActive = 1
```

	BRANCHID	ISACTIVE	PRODDATE	MODELID	CARID
1	432	1	23/03/1952	9831342330	1
2	774	1	01/06/1990	4271643110	3
3	503	1	03/12/1996	6052230695	4
4	782	1	21/04/1960	1027720104	7
5	273	1	16/01/1947	3403836402	10
6	993	1	18/05/1973	1902483743	12
7	626	1	10/07/1943	2485674520	13
8	884	1	08/09/1934	5455907711	16
9	948	1	16/08/2001	2217458561	20
10	448	1	17/07/1973	5790842187	24
11	726	1	18/03/1965	4724069405	26
12	376	1	21/06/1994	3756006546	31
13	755	1	06/09/1948	7010676533	33
14	866	1	02/02/2015	9730950191	34
15	668	1	19/12/1996	187972137	37
16	383	1	15/09/1939	3080631069	42
17	995	1	31/10/1997	9643280826	46
18	270	1	23/05/1955	8822178606	47
19	601	1	27/04/2017	4561065067	49

miletzky@labdbwin 969 rows selected in 0.691 seconds

### שאלתה 8:

בחברה מעוניינים לקבל את הרשומות עבור הנהגים שהם בטווח סיכון מוגבר לבצע תאונות:

**שמות כל הלקוחות שהגיל שלהם קטן מ-20**

The screenshot shows a database management interface with a SQL query editor and a results grid. The query is:

```
select*  
from client  
where cbdate > {current_date - 20*365}
```

The results grid displays the following data:

	CLIENTID	CADDRESS	CNAME	CBDATE
1	358	646 Kelly HarborNorth Shannonville, MI 77116	Kara Webster	18/11/
2	31	3611 Jennifer Locks Suite 150New Daniel, SD 88911	Richard Stevens	23/04/
3	55	4818 Michelle Dam Suite 529Jesusburgh, AR 63937	Erin Garcia	23/01/
4	68	69011 Elizabeth Ways Apt. 650East Elaine, ID 20073	Connie Flynn	13/10/
5	132	4019 Holmes LocksWashingtonshire, AK 70743	Nicholas Byrd	04/12/
6	219	754 Avers Isle Apt. 081Johnsonton. IA 13149	Michael Stevens	01/06/

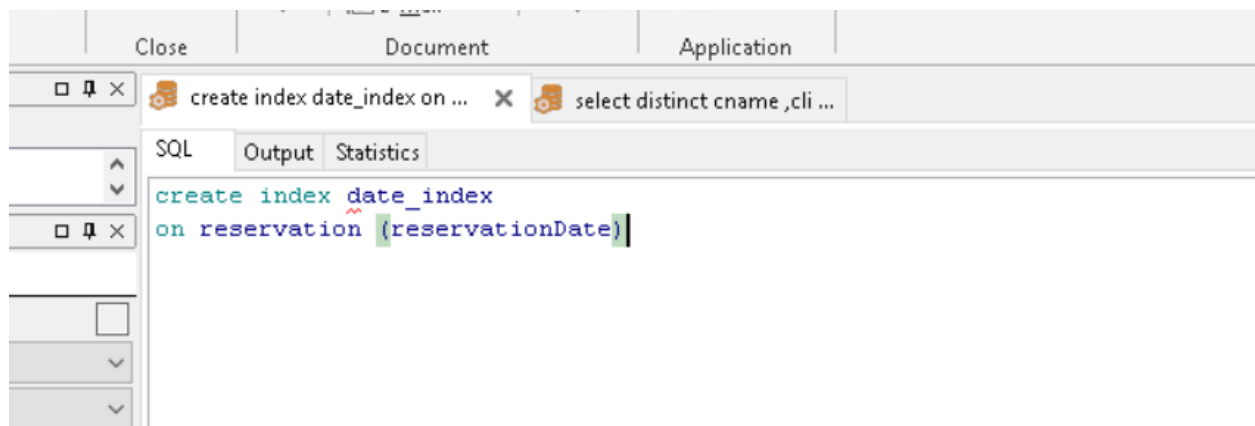
The status bar at the bottom indicates: 3:39, miletzky@labdbwin, 647 rows selected in 0.306 seconds.



## אינדקסים

אינדקס 1:

ביצענו אינדקס על התאריכים של 'תאריך הזמנה' בטבלה RESERVATION



לאחר הרצה שוב של שאילתא 3 קיבלנו תוצאות יותר גרועות. 1.127 שניה במקום 0.821

הפרש של 0.306 מעניין למה קיבלנו תוצאה פחות טובה.

SQL Output Statistics

```
select*
from client
where clientid in(
select clientid
from car natural join reservation
where proddate < TO_DATE('15/07/2005', 'DD/MM/YY'))
```

	CLIENTID	CADDRESS	CNAME	CBDATE	CLISENCEI
1	256	94690 Todd TerraceTurnerfurt, NY 12795	Laura Arnold	14/10/1938	08/01/1961
2	258	89507 Amber Views Apt. 084Lake Jaredbury, KY 64264	Curtis Dunn	04/12/1985	26/06/1981
3	259	9372 Timothy VistaWest Dianeview, IN 57974	Nancy Williams	17/12/1988	14/04/1981
4	261	09856 Boyer Lights Apt. 865Davisborough, DC 33968	Bianca Hernandez	27/07/1988	13/01/1991
5	262	0320 Scott ShoreKathyburgh, WA 22273	Leah Cole	18/10/1994	12/03/1981
6	263	3774 Jones Alley Suite 950Trevorbury, LA 03724	Jermaine Villarreal	18/01/1938	08/11/1991
7	264	4928 Sean Loaf Suite 727North Ashleyfort, AZ 49515	Jenny Hunt	27/10/1972	31/01/1971

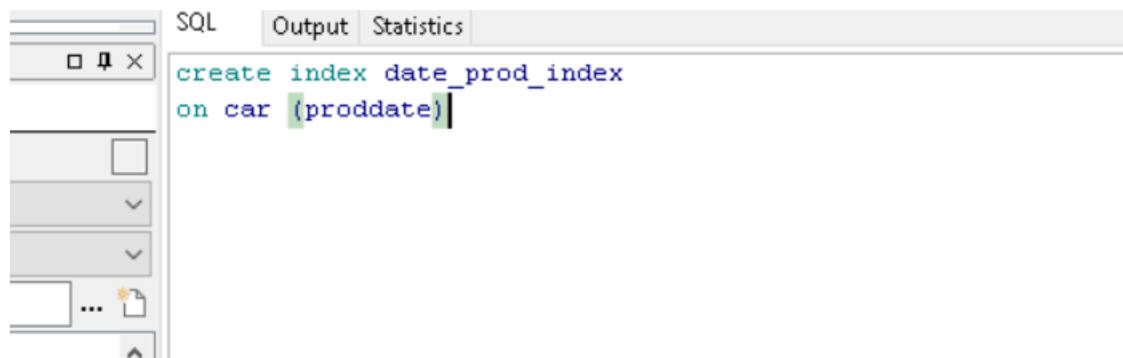
1 of 23 miletzky@labdbwin 23 rows selected in 0.043 seconds (more...)

```
select distinct cname ,clientid
from reservation natural join client
where reservationDate> TO_DATE('01/01/2000', 'DD/MM/YY') and reservationDate< TO_DATE('01/01/2010', 'DD/MM/YY')
```

	CNAME	CLIENTID
1	Bianca Hernandez	261
2	Melissa Nguyen	292
3	Tammy Salazar	308
4	Adam Hughes	350
5	Linda Hunt	359
6	George Alvarez	9
7	Lisa Nelson	60
8	Renee Ferguson	102
9	Dr. Eric Todd	254
10	Jason Ramirez	442
11	Jason Freeman	555
12	Amanda Mathews	783
13	Peter Henson	819
14	Andrew Martinez	491
15	Melvin Cunningham	531
16	Abigail Lowe	310
17	Denise Murray	1147
18	Cassie Walsh	1148
19	Joshua Stewart	1178
20	Clifford Miller	1257

1 of 23 0:01 miletzky@labdbwin 3556 rows selected in 1.127 seconds

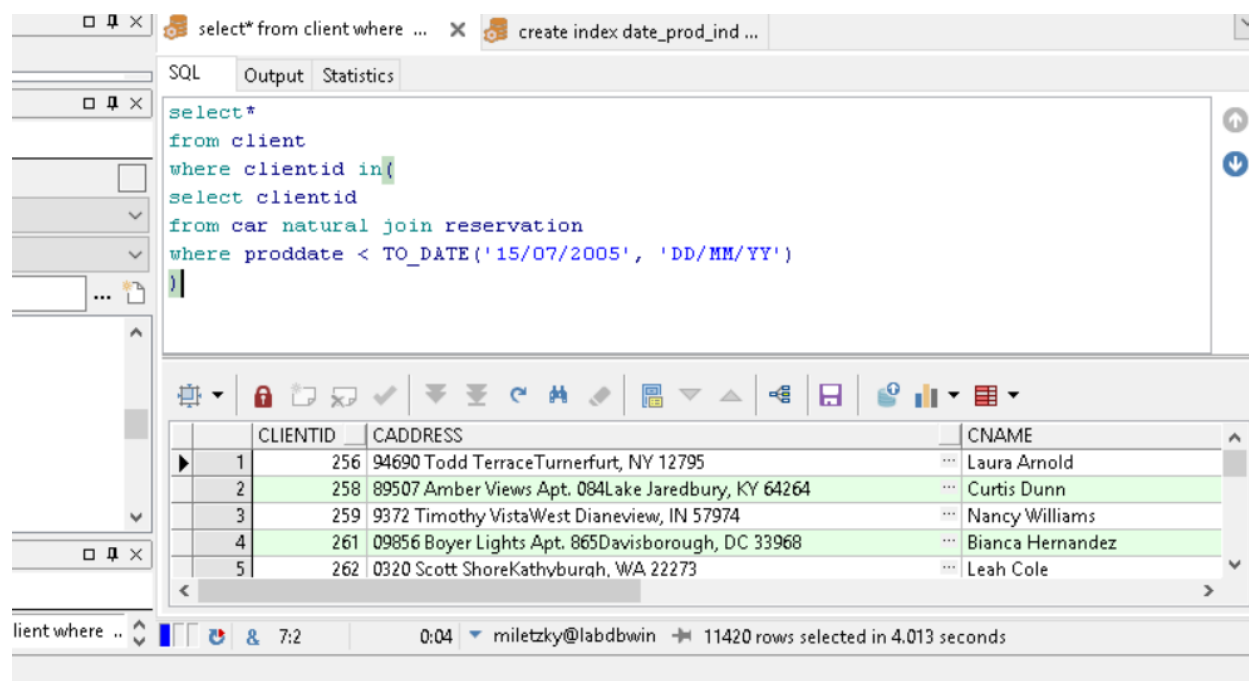
אינדקס 2: הרצנו את שאילתה 2 שוב אחרי שיצרנו את האינדקס הנל על תאריך יצור בטבלה CAR



```
SQL
Output
Statistics

create index date_prod_index
on car (proddate)
```

זה התוצאה עם האינדקס שיפור בערך של 12% לעומת הרצה ללא האינדקס



```
select* from client where ...
create index date_prod_ind ...

SQL
Output
Statistics

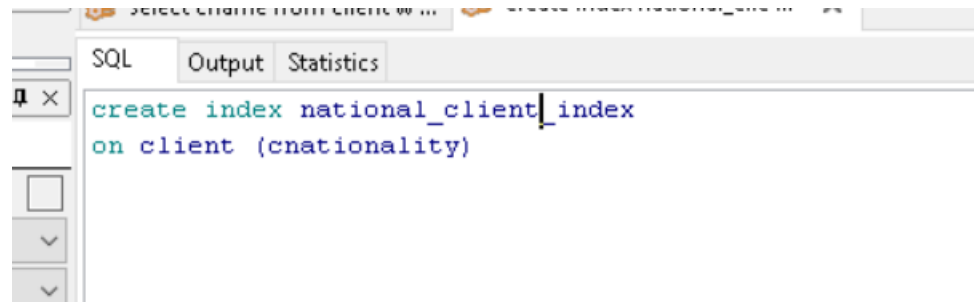
select*
from client
where clientid in(
select clientid
from car natural join reservation
where proddate < TO_DATE('15/07/2005', 'DD/MM/YY'))
```

	CLIENTID	CADDRESS	CNAME
1	256	94690 Todd TerraceTurnerfurt, NY 12795	Laura Arnold
2	258	89507 Amber Views Apt. 084Lake Jaredbury, KY 64264	Curtis Dunn
3	259	9372 Timothy VistaWest Dianeview, IN 57974	Nancy Williams
4	261	09856 Boyer Lights Apt. 865Davisborough, DC 33968	Bianca Hernandez
5	262	0320 Scott ShoreKathyburgh, WA 22273	Leah Cole

lient where .. 7:2 0:04 miletzky@labdbwin 11420 rows selected in 4.013 seconds

אינדקס 3:

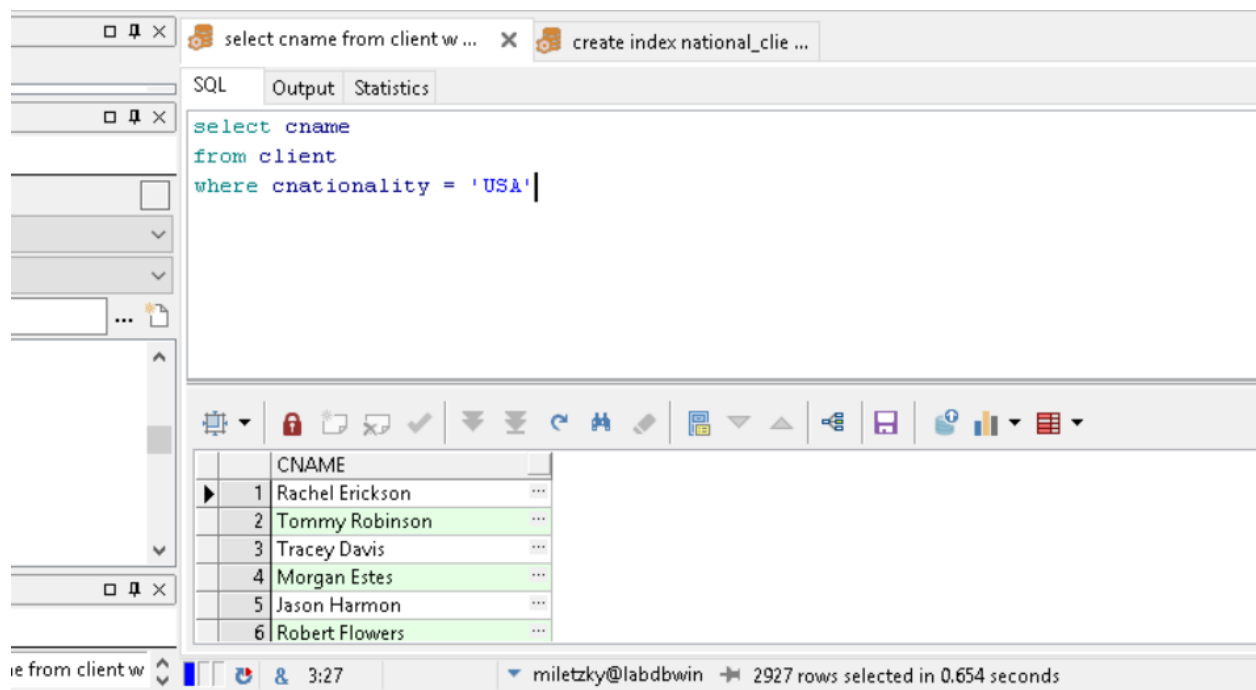
יצרנו אינדקס על ארץ מוצא של לקוח עבור CLIENT



The screenshot shows the SQL Developer interface with the 'SQL' tab selected. The SQL editor contains the following text:

```
create index national_client_index  
on client (cnationality)
```

שיפור של 10 אחוז



The screenshot shows the SQL Developer interface with the 'SQL' tab selected. The SQL editor contains the following text:

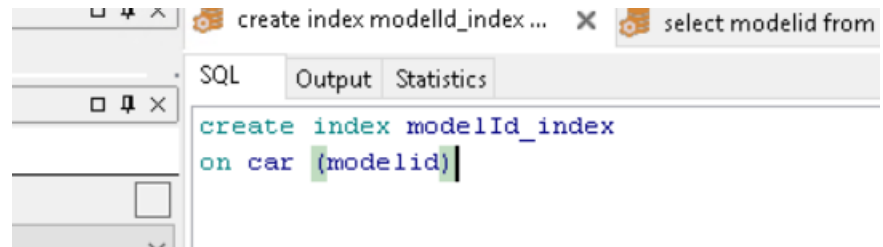
```
select cname  
from client  
where cnationality = 'USA'
```

Below the SQL editor, the 'Output' tab is selected, displaying a table with 6 rows and 1 column (CNAME). The table contains the following data:

	CNAME
1	Rachel Erickson
2	Tommy Robinson
3	Tracey Davis
4	Morgan Estes
5	Jason Harmon
6	Robert Flowers

The status bar at the bottom indicates: 2927 rows selected in 0.654 seconds.

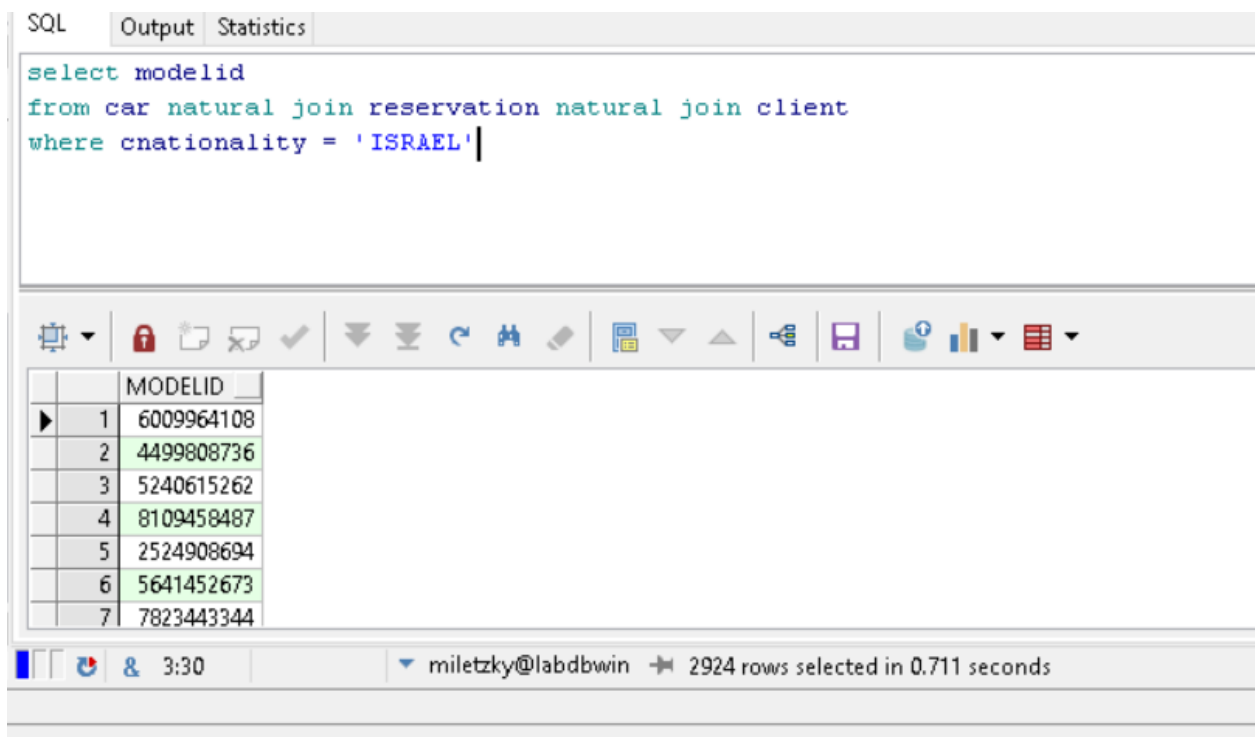
אינדקס 4: הוספנו עוד אינדקס על model id של car



The screenshot shows a SQL IDE window with two tabs: 'create index modelId\_index ...' and 'select modelid from'. The 'SQL' tab is active, displaying the following code:

```
create index modelId_index
on car (modelid)
```

זה נתן לנו שיפור של 50% ביחס להרצה של שאילתה 4 זה קרה גם בגלל האינדקס הקודם על הלאום של הלקוח



The screenshot shows a SQL IDE window with the 'SQL' tab active, displaying the following query:

```
select modelid
from car natural join reservation natural join client
where cnationality = 'ISRAEL'
```

Below the query editor, there is a toolbar and a table of results. The table has two columns: an index and 'MODELID'. The results are as follows:

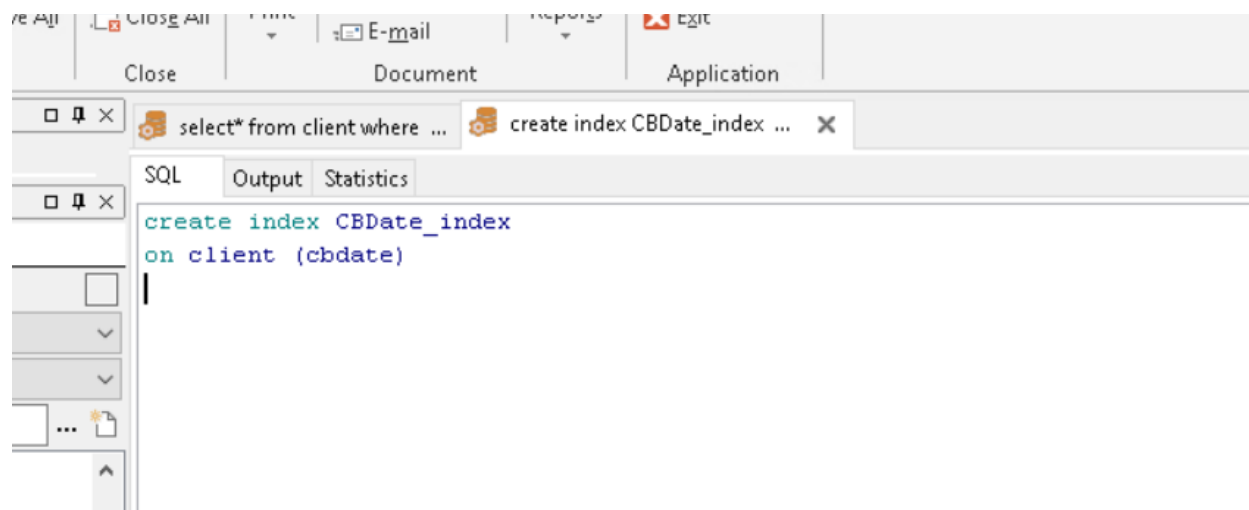
	MODELID
1	6009964108
2	4499808736
3	5240615262
4	8109458487
5	2524908694
6	5641452673
7	7823443344

At the bottom of the IDE, the status bar shows 'miletzky@labdbwin' and '2924 rows selected in 0.711 seconds'.

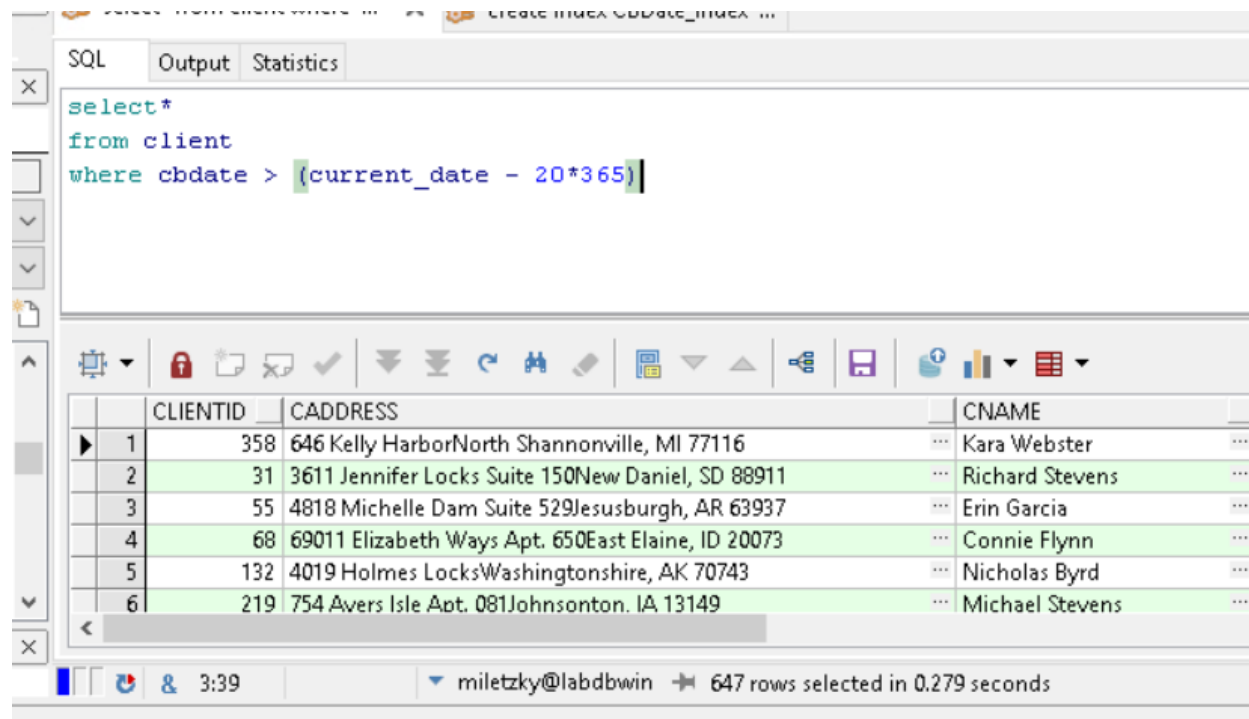
אינדקס 5:

אינדקס על עמודת תאריך לידה של לקוח

משפיע על שאלתה 8



תוצאה: שיפור של 30%



## אינטגרציה

קיבלנו grant מ אלעד ואחיה את הישויות

אלעד אחיה 2,4,6,15,12,13

שאלתה 1: הסיבות שבגללם ביטלו הזמנות בשנת 2020

The screenshot shows a SQL query in a database client interface. The query is:

```
select distinct cause
from reservation natural join elafishe.cancellations
where reservationdate >TO_DATE('1/1/2020' , 'DD/MM/YY') and reservationdate< TO_DATE('1/1/2021', 'DD/MM/YY')
```

The results are displayed in a table with 15 rows. The first row is highlighted in blue, and the rest are highlighted in green. The status bar at the bottom indicates "7 of 15" rows and "15 rows selected in 0.015 seconds".

CAUSE
1 my grandparents bought me a new car
2 I had an unpleasant experience with the company
3 it costs too much for us to justify it
4 I killed my wife and I should go to jail
5 my kids are annoying
6 the ac is not good enough for Israel in the summer
7 I am very disappointed because a medieval knight didn't attack me and
8 because
9 the bride decided to run
10 I moved to the competitor
11 I moved to Canada, It's too hot here in Israel
12 I was too happy to drive
13 a medieval knight attacked me with an axe and I am now mortally wounded
14 I became addicted to fantasy books so I don't want to drive any more
15 an unexpected event happened to my seventh child

שאלתה 2:

סכום התשלומים שהתקבלו ברבעון הראשון של שנת 2019

The screenshot shows a SQL query in a database client interface. The query is:

```
SELECT SUM(sum)
FROM reservation natural join elafishe.payments
WHERE reservationdate >TO_DATE('1/1/2019' , 'DD/MM/YY') and reservationdate< TO_DATE('1/4/2019', 'DD/MM/YY');
```

The results are displayed in a table with 1 row. The first row is highlighted in blue. The status bar at the bottom indicates "1" row.

SUM(SUM)
1 992503692

### שאלתה 3:

כל התוספות שנרכשו בשנה האחרונה

SQL Output Statistics

```
select distinct description
from elafishe.extras e , elafishe.extrasfororders efo
where efo.order_id in(
select me.reservationid
from reservation me , elafishe.extrasfororders efo
where me.reservationdate > TO_DATE('1/1/2019', 'DD/MM/YY') and
me.reservationdate < TO_DATE('1/1/2020', 'DD/MM/YY')
);
```

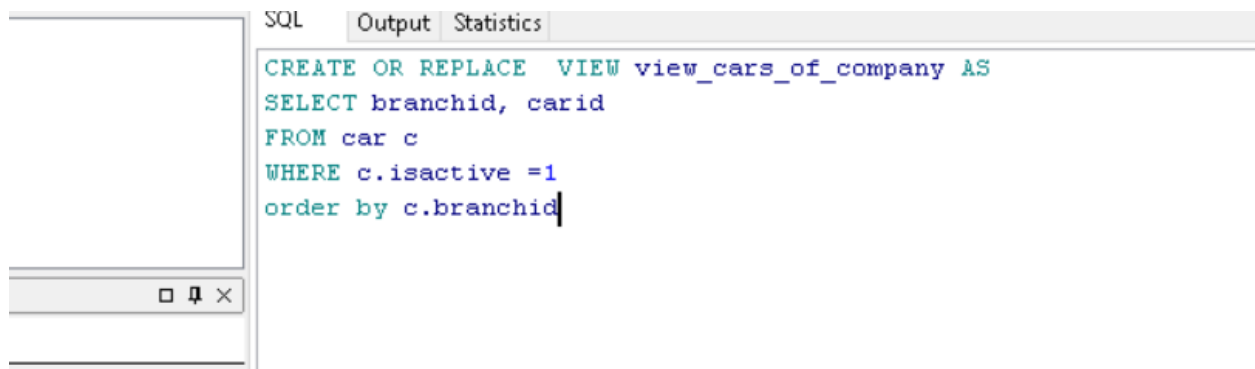
	DESCRIPTION
1	charger
2	multimedia_system
3	booster_chair
4	tablet



## VIEW'S

(1) יצירת VIEW עבור מנהלי הסניפים בכדי שיוכלו לשלוף בקלות מידע על כל הרכבים הפעילים בחברה

והיכן הם נמצאים בכל רגע נתון

A screenshot of a SQL IDE window. The window has three tabs: 'SQL', 'Output', and 'Statistics'. The 'SQL' tab is active and contains the following SQL code:

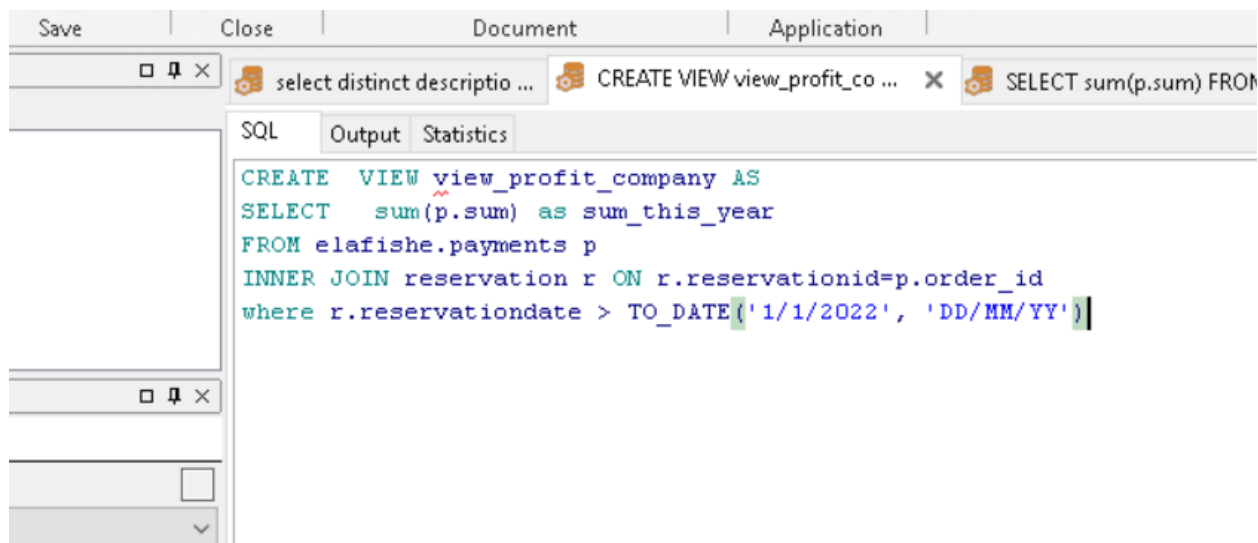
```
CREATE OR REPLACE VIEW view_cars_of_company AS
SELECT branchid, carid
FROM car c
WHERE c.isactive =1
order by c.branchid
```

The code is color-coded: 'CREATE OR REPLACE' is blue, 'VIEW' is green, 'view\_cars\_of\_company' is blue, 'AS' is green, 'SELECT' is blue, 'branchid, carid' is blue, 'FROM' is blue, 'car c' is blue, 'WHERE' is blue, 'c.isactive =1' is blue, and 'order by c.branchid' is blue. The 'Output' and 'Statistics' tabs are empty. The window has a standard Windows-style title bar and a toolbar with icons for undo, redo, and save.

(2) יצירת VIEW עבור מנהלי הסניפים כמה הזמנות התבצעו היום

```
CREATE OR REPLACE VIEW view_cars_reseve_today AS
SELECT *
FROM reservation c
WHERE c.reservationdate =current_date
```

(3) יצירת VIEW עבור הנהלת חשבונות בכדי לשלוף את מחזור הכספים לנקודת זמן זו מתחילת שנה.



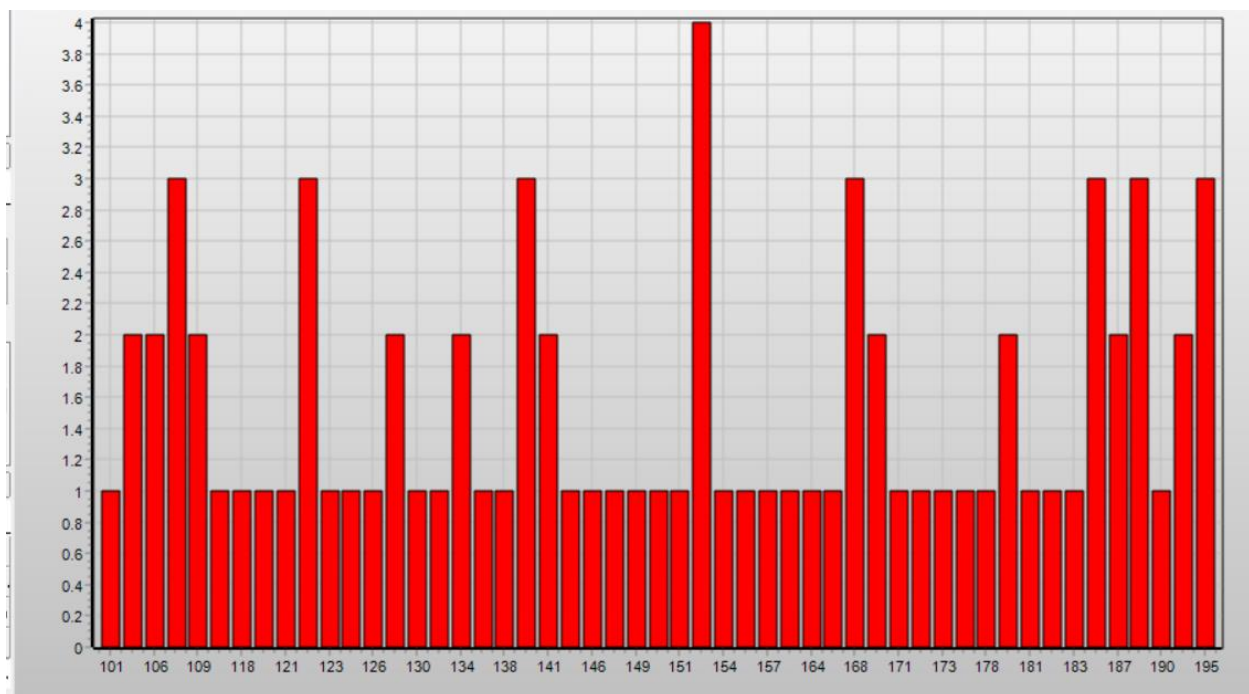
(4) הלקוחות שביצעו הזמנה בחודש האחרון

```
CREATE OR REPLACE VIEW view_res_this_month AS
SELECT c.clientid
FROM reservation r inner join client c
on c.clientid= r.clientid
WHERE r.reservationdate > current_date - 30;
```

## תרשימים

תרשים 1:

מייצג התפלגות של כמות כלי הרכב ביחס לסניפים באיזור 100 עד 200. ציר X זה מספר הסניף, ציר Y מייצג את כמות הרכבים בסניף



כאשר השאילתה היא:

```
SQL      Output  Statistics
SELECT branchid, count(*) as b
FROM car c
WHERE c.isactive =1 and c.branchid>100 and c.branchid < 200
group by c.branchid
order by c.branchid
```

תרשים 2:

התפלגות הרווחים ביחס לחודשים 3,4,5 על פני 20 השנים האחרונות.



השאלתה שהשיגה את הנתונים:

SQL

```
SELECT p.payment_month, sum(p.sum)
FROM elafishe.payments p
INNER JOIN reservation r ON r.reservationid=p.order_id
where r.reservationdate > TO_DATE('1/1/2000', 'DD/MM/YY')
group by p.payment_month
```

Output

PAYEMENT_MONTH	SUM(P.SUM)
3	592943
2	1705061
1	1865613

## פרוצדורות

(1) ברצוננו להדפיס למסך מאיזה רוב מגדרי מורכב מאגר הלקוחות של החברה

הגדרת הפרוצדורה:

```
create or replace procedure genderMajority as
women number:=0;
men number:=0;

begin
  select count(*) into women from client c where c.cgender = 'f';
  select count(*) into men from client c where c.cgender = 'm';

  if(women>men)
  then dbms_output.put_line('There is more women clients then men ');
  else if(women<=men) then dbms_output.put_line('There is more men clients then women ');
  end if;end if;
end;
```

שימוש:

```
begin
  gendermajority;
end
;
```

ההדפסה:

```
There is more men clients then women
There is more men clients then women
```

## פרוצדורה 2:

(2) הדפס את מס ההזמנות שהתבצעו בחודש מסוים, המתקבל כפרמטר, לאורך השנים.

---

```
create or replace procedure get_max_res_month (monthtosearch in number) is

    monthmax number;
    resmax number;
cursor ord_month is
select * from(
    select  EXTRACT(month FROM reservation.reservationdate) as Montht,
    COUNT(reservation.reservationdate) as No_of_ord
    FROM reservation
    GROUP BY EXTRACT(month FROM reservation.reservationdate)
    ORDER BY No_of_ord DESC) s
where s.montht = monthtosearch;

begin
    open
        ord_month;
    loop

        FETCH ord_month into monthmax, resmax;
        EXIT WHEN ord_month%notfound;

        dbms_output.put_line(resmax|| ' reservations made in month ' || monthmax )
    END LOOP;
    CLOSE ord_month;
END;
```

קריאה לפרוצדורה:

```
begin
    get_max_res_month(11);
end;
```

ההדפסה:

---

```
1688 reservations made in month 11
|
```

## פונקציות

### פונקציה 1:

מקבלת כפרמטר מדינה שלקוחות מסוימים מגיעים משם ומחזירה את אחוז הלקוחות המגיעים ממדינה זו.

```
create or replace function national_percent (nationalp in char)
return number is
total_clients number;
amount_this_nation number;

begin

select count(*) into total_clients from client;
select count(*) into amount_this_nation from client c where c.cnationality = nationalp;

return(amount_this_nation /total_clients );
end;
```

הקריאה:

```
declare rec number;
begin
rec := national_percent('ISRAEL');
dbms_output.put_line(rec);
end;
```

הדפסה:

0.14225

|

## פונקציה 2:

מקבלת כפרמטר מדינה וגיל. מחזירה את מספר הלקוחות ממדינה זו מעל הגיל הנתון

```
create or replace function old_from_age_percent (age in number , nationalp in char)
return number is
total_clients number;
amount_above number;

begin

select count(*) into total_clients from client;
select count(*) into amount_above from client c where c.cnationality = nationalp
and c.cbdate < (current_date - 365*age);
  dbms_output.put_line('there is ' || amount_above || ' cliens from ' || nationalp || ' above ' || age || ' age' );

return(amount_above );
end;
```

הקריאה:

```
declare rec number;
begin
  rec := old_from_age_percent(20,'ISRAEL');
| end;
```

הדפסה:

```
there is 2771 cliens from ISRAEL above 20 age
|
```

# סוף!