גמ"ח לב
בנק הלוואות ללא ריבית
הפרויקט שלנו מתאר ניהול ומעקב
עבור בנק הלוואות גמ"ח לב
בסיס הנתונים מנהל את מספר סניפים
של גמ"ח לב, אשר קיימים
בערים שונות בישראל.
מרצה: מר אריה ויזן
מגישים: שמעון מזרחי 203375563 , יהודה משה בנימין 328972427

תוכן עניינים

חלק א' 6
1.1 תיאור המערכת 6
1.2 רשימת ישויות 6
1.3 דיאגרמת ERD 8
1.4 תיאור הקשרים 8
1.5 תרשים DSD 9
1.6 יצירת הטבלאות ע"י SQL 9
1.6.1 טבלת Person 9
1.6.2 טבלת Recipient 9
1.6.3 טבלת Clerk 9
1.6.4 טבלת Manager 10
1.6.5 טבלת Fund 10
1.6.6 טבלת Loan 10
1.6.7 טבלת Payments 10
1.7 מחיקת הטבלאות (לפי הסדר הנכון) 11
1.8 הכנסת נתונים לטבלאות ע"י שימוש ב Data Generator 11
1.8.1 טבלת Person 11
1.8.2 טבלת Recipient 12
1.8.3 טבלת Clerk 13
1.8.4 טבלת Manager 14
1.8.5 טבלת Fund 16
1.8.6 טבלת Loan 17
1.8.7 טבלת Payments 18
1.9 יבוא ע"י Text Importer 19
1.9.1 יבוא טבלת Fund 19
1.9.2 יבוא קובץ טקסט של Manager 21
1.10 יבוא דרך ODBC Importer 23
1.10.1 יבוא של טבלת Person 23
1.10.2 יבוא של טבלת Clerk 24
1.11 יצוא בשיטת Oracle Express 26
1.12 שאילתות 27
1.12.1 קידום עובדים מצטיינים 27
1.12.2 נאמנות הלווה והערב שלו 28
1.12.3 לווים ששילמו את רוב חובותיהם 29
1.12.4 מנהלים שמקבלים פחות ממוצע העובדים הזוטרים 30

1.12.5 הבנקים שהלוו הכי הרבה כסף לפי טווח תאריכים 31
1.12.6 קבלת נתונים של תשלומים למנהל 33
1.12.7 הערכת זמן סיום פירעון החוב 34
1.12.8 לווים שלא עומדים בתשלומים 35
1.13 שאילתות Update 36
1.13.1 הוספת מספר תשלומים עבור הלוואות גדולות 36
1.13.2 העלאות משכורת..... 37
1.14 שאילתות Delete 39
1.14.1 מחיקת חובות ישנים..... 39
1.14.2 פיטורי עובדים שאינם איכותיים..... 40
1.15 Commit ו Rollback 43
חלק ב' 46
2.1 הרחבת המערכת..... 46
2.2 רשימת ישויות 46
2.2.1 רשימת ישויות שקיימים כבר 46
2.2.2 קשרים וישויות חדשים שיתווספו..... 47
2.3 אלגוריתם האינטגרציה:..... 47
2.4 דיאגרמת ERD 49
2.5 תרשים DSD..... 51
2.6 יצירת הטבלאות ע"י SQL 51
2.7 הכנסת נתונים לטבלאות ע"י שימוש ב Data Generator..... 55
2.7.1 הכנסת הטבלה Staff 55
2.7.2 הכנסת הטבלה Reader..... 56
2.7.3 הכנסת הטבלה Customer 58
2.7.4 הכנסת הטבלה Publisher..... 59
2.7.5 הכנסת הטבלה Category 61
2.7.6 הכנסת הטבלה Product 62
2.7.7 הכנסת הטבלה Store..... 64
2.7.8 הכנסת הטבלה Sale 65
2.7.9 הכנסת הטבלה CommisionSale..... 67
2.7.10 הכנסת הטבלה Contain 68
הגדרות הסכמה: 68
2.7.11 הכנסת הטבלה Store..... 69
2.7.12 הכנסת הטבלה Supplier 71
2.7.13 הכנסת הטבלה Have 72
2.7.14 הכנסת הטבלה StoreEmployee 74
2.7.15 הכנסת הטבלה BorrowBy 75

2.7.16 הכנסת הטבלה Book 76
2.7.17 הכנסת הטבלה Maintain 78
2.8 פקודות Alter Table 80
2.8.1 שינוי סכמת StoreEmployee 80
2.8.2 שינוי סכמת Book 83
2.8.3 שינוי סכמת Sale 84
2.8.4 שינוי סכמת Customer 85
2.8.5 הוספת אילוץ חדש 86
2.9 מבטים (Views) 88
2.9.1 מידע לפקיד על לווה פוטנציאלי 88
2.9.2 סניפי הבנק אשר הלוו את סכום הכסף הרב ביותר 89
2.10 אינדקסים 92
2.10.1 זמן החזר הלוואה מוערך 92
2.10.2 שכר רב שנתי של מנהל שקיבל את סכום הכסף הרב ביותר 94
2.10.3 לווים שהחזירו את רוב החוב שלקחו 95
2.11 Grant and Revoke 98
2.12 שאילתות על בסיס הנתונים המשולב 102
2.12.1 רווחים עבור מכירות עם עמלה 102
2.12.2 סניף הגמ"ח שסיבסד את כמות הקניות המרבי 103
חלק ג' 104
3.1 שאילתות עם פרמטרים 104
3.1.1 מידע על חובות שאחוז מסויים מהם כבר שולם 104
3.1.2 סכום סך כל המשכורות עבור עובד כלשהו 106
3.1.3 מידע על מי שקונה הרבה ספרים לפני הזמן האחרון לפירעון 108
3.1.4 קבלת מידע על אנשים שלא מחזירים הלוואות 110
3.2 דוחות 112
3.2.1 עלויות סבסוד קניות 112
3.2.2 עשרת המנהלים שקיבלו את סכומי המשכורות הגבוהים ביותר 113
3.3 גרפים 115
3.3.1 אחוזי רווחים ממכירת ספרים 115
3.3.2 כמות תשלומי הבונוסים למנהלים ביחס לעלות תשלום כוח האדם הכללי 118
3.4 פרוצדורות 119
3.4.1 עדכון מוניטין של לווים הגונים 119
3.4.2 הגדלת מאזן של סניפים עקב מגפת הקורונה 122
3.5 פונקציות 124
3.5.1 פיטורי מנהלים שסרחו 124
3.5.2 סך סכום ההלוואות בתאריכים רצויים עבור כל סניף של הגמ"ח 127

3.6 טריגרים 129
3.6.1 עדכון מאזן הסניף מיד לאחר ביצוע הלוואה חדשה 129
3.6.2 הוספת עובד בעל הרשאות מרובות לטבלת המנהלים 132
3.7 סיכום 133

חלק א'

1.1 תיאור המערכת

המערכת מאפשרת ניהול ומעקב עבור בנק הלוואות גמ"ח לב.

המערכת מנהלת מספר סניפים של גמ"ח לב, אשר קיימים בערים שונים בישראל.

מטרות המערכת הינם:

- ניהול תשלומי ההלוואה תוך בקרה על החזר התשלומים בזמנם.
- אחסון נתונים בנוגע לאמינות של מקבלי ההלוואות, וכן אמינות של ערבי הלווים, על מנת שיתקבל ניהול עתידי טוב יותר.
- חישובים כספיים בנוגע למאזן הכספי של הסניפי הגמ"ח. בנוסף המערכת מציגה סך יתרת תשלום לפי תאריך נוכחי למקבל ההלוואה, לפי תעודת זהות ומספר הלוואה.
- ניהול עובדים, כגון משכורות לפי דירוג וכן ותק העובד.

1.2 רשימת ישויות

Person — ישות חזקה אשר מכילה את בסיס כל העובדים ולוקחי ההלוואה:

- ID מפתח תעודת זהות, ייחודי לזיהוי אדם.
- FirstName שם פרטי.
- LastName שם משפחה.
- Email כתובת דואר אלקטרוני.
- Address כתובת מגורים.

Clerk — ישות חלשה – מייצגת עובד:

- ID תעודת זהות – מפתח זר של Person.
- Hiredate תאריך תחילת עבודה (לחישוב הערכת עובד).
- Rating רמת שירות.
- Salary משכורת העובד.
- Authorization הרשאות העובד.

Manager — ישות חלשה המתארת מנהל גמ"ח:

- Id תעודת זהות של עובד (מפתח זר).
- SalaryBonusPercentange אחוזי הבונוס למשכורת של המנהל ממשכורת הבסיס.

Recipient — ישות חלשה – מייצגת אדם שמקבל\קיבל הלוואה:

- ID תעודת זהות – מפתח זר של Person.
- Credibility את טיבו של מבקש ההלוואה.
- MaximalAllowedLoan סכום מקסימלי אשר ניתן לתת לאדם זה.
- GuarantorId מספר תעודת זהות הערב של מבקש ההלוואה.

Loan — ישות המתארת הלוואה:

- LoanNumber מספר סידורי של הלוואה.
- ID: תעודת זהות של לוקח ההלוואה (מפתח זר)
- DateTaken תאריך לקיחת ההלוואה.
- InitialLoanAmount סכום הכסף שנלקח בהלוואה.
- TotalNumberPayments מספר התשלומים המקסימלי ללוקח ההלוואה.
- BranchCode קוד הסניף של הסניף ממנו נלקחה ההלוואה(מפתח זר).

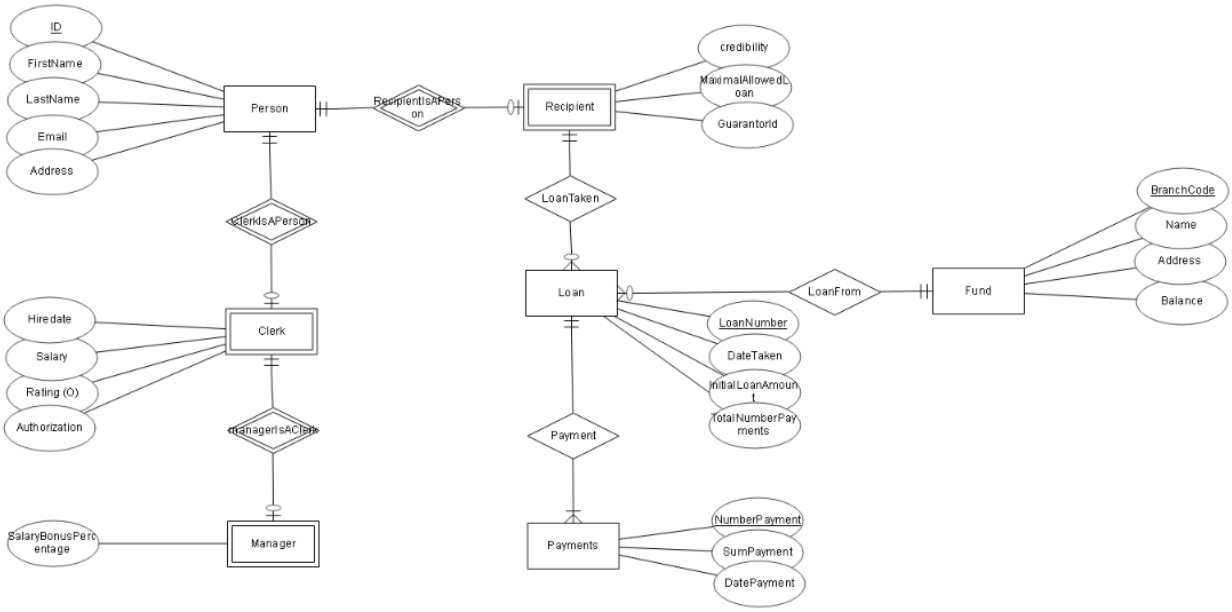
Payments — ישות המתארת החזר תשלום אחד מההלוואה:

- NumberPayment מספר התשלום המוחזר מההלוואה (מתחיל מ1).
- LoanNumber מספר הסידורי של ההלוואה (מפתח זר).
- SumPayment סכום ההחזר.
- DatePayment תאריך ההחזר.

Fund — ישות המתארת סניף של הגמ"ח:

- BranchCode מפתח - קוד הסניף.
- Balance מאזן הכספים בסניף.
- Address כתובת הסניף.
- Name שם הסניף.

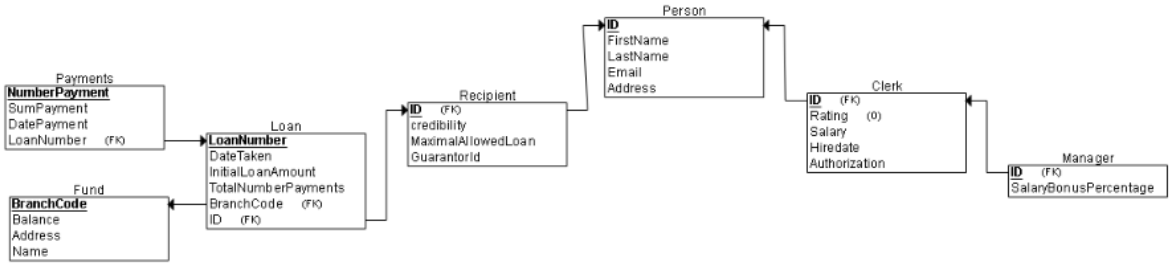
1.3 דיאגרמת ERD



1.4 תיאור הקשרים

- **ClerkIsAPerson** פקיד הינו אדם, קשר יחיד ליחיד.
- **managerIsAClerk** מנהל הינו פקיד, קשר יחיד ליחיד.
- **RecipientIsAPerson** מקבל ההלוואה הינו אדם, קשר יחיד ליחיד.
- **LoanTaken** הלוואה נלקחת ע"י מקבל ההלוואה, קשר יחיד לרבים, מכיוון שאדם אחד יכול לקחת מספר הלוואות.
- **Payment** להלוואה בודדת ישנם מספר תשלומים קבוע מראש, לפיכך קשר יחיד לרבים.
- **LoanFrom** לכל הלוואה ישנו סניף בנק בודד, אך בכל סניף בנק יכול להיות מספר של הלוואות, לפיכך קשר יחיד לרבים, כאמור לסניף יש מס' הלוואות.

1.5 תרשים DSD



1.6 יצירת הטבלאות ע"י SQL

1.6.1 טבלת Person

```
CREATE TABLE Person
(
  ID INT NOT NULL,
  FirstName VARCHAR(25) NOT NULL,
  LastName VARCHAR(25) NOT NULL,
  Email VARCHAR(30) NOT NULL,
  Address VARCHAR(40) NOT NULL,
  PRIMARY KEY (ID)
);
```

1.6.2 טבלת Recipient

```
CREATE TABLE Recipient
(
  credibility INT NOT NULL,
  MaximalAllowedLoan INT NOT NULL,
  GuarantorId INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY(ID),
  FOREIGN KEY (ID) REFERENCES Person(ID)
);
```

1.6.3 טבלת Clerk

```
CREATE TABLE Clerk
(
```

```
  Rating INT,
  Salary INT NOT NULL,
  Hiredate DATE NOT NULL,
  Authorization INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ID) REFERENCES Person(ID)
);
```

1.6.4 טבלת Manager

```
CREATE TABLE Manager
(
  SalaryBonusPercentage INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ID) REFERENCES Clerk(ID)
);
```

1.6.5 טבלת Fund

```
CREATE TABLE Fund
(
  Balance INT NOT NULL,
  Address VARCHAR(40) NOT NULL,
  Name VARCHAR(25) NOT NULL,
  BranchCode INT NOT NULL,
  PRIMARY KEY (BranchCode)
);
```

1.6.6 טבלת Loan

```
CREATE TABLE Loan
(
  LoanNumber INT NOT NULL,
  DateTaken DATE NOT NULL,
  InitialLoanAmount INT NOT NULL,
  TotalNumberPayments INT NOT NULL,
  BranchCode INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (LoanNumber),
  FOREIGN KEY (BranchCode) REFERENCES Fund(BranchCode),
  FOREIGN KEY (ID) REFERENCES Recipient(ID)
);
```

1.6.7 טבלת Payments

```
CREATE TABLE Payments
(
  NumberPayment INT NOT NULL,
  SumPayment INT NOT NULL,
```

```
  DatePayment DATE NOT NULL,
  LoanNumber INT NOT NULL,
  PRIMARY KEY (NumberPayment),
  FOREIGN KEY (LoanNumber) REFERENCES Loan(LoanNumber)
);
```

1.7 מחיקת הטבלאות (לפי הסדר הנכון)

```
DROP TABLE Manager;
DROP TABLE Clerk;
DROP TABLE Payments;
DROP TABLE Loan;
DROP TABLE Fund;
DROP TABLE Recipient;
DROP TABLE Person;
```

1.8 הכנסת נתונים לטבלאות ע"י שימוש ב Data Generator

1.8.1 טבלת Person

:Data Generator

PERSON

Owner: SHIMIZRA | Table: PERSON | Number of records: 1000

Name	Type	Size	Data
ID	NUMBER		[100000000]
FIRSTNAME	VARCHAR2	25	FirstName
LASTNAME	VARCHAR2	25	LastName
EMAIL	VARCHAR2	30	Email
ADDRESS	VARCHAR2	40	Address1 + ', ' + City

הצגת פקודת Insert into:

```
insert into SHIMIZRA.PERSON (ID, FIRSTNAME, LASTNAME, EMAIL, ADDRESS)
values (877823523, 'Merillee', 'Nielsen', 'm.nielsen@lemproducts.de', '61 Dinslaken Road, Leimen');

insert into SHIMIZRA.PERSON (ID, FIRSTNAME, LASTNAME, EMAIL, ADDRESS)
values (159953080, 'Garland', 'Edmunds', 'garland.edmunds@newhorizons.co', '4 Shearer Drive, Caguas');

insert into SHIMIZRA.PERSON (ID, FIRSTNAME, LASTNAME, EMAIL, ADDRESS)
values (379736606, 'Penelope', 'Piven', 'penelope.piven@usdairyproducer', '23 Bismarck Road, Bristol');

insert into SHIMIZRA.PERSON (ID, FIRSTNAME, LASTNAME, EMAIL, ADDRESS)
values (302340384, 'Dylan', 'Gill', 'd.gill@paisley.au', '80 Norton Road, Springfield');
```

הצגת הנתונים של הטבלה:

PERSON

ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
529864579	Giancarlo	Withers	giancarlo.withers@jsa.br	52nd Street, Mogi Guacu
909110541	Tanya	Lucien	tanya.lucien@apexsystems.com	20 Brooks Street, New hartford
129417750	Goran	Brooks	g.brooks@stonebrewing.jp	177 Baranski Road, Chiba
115529574	Lynette	Visnjic	lynettev@shar.jp	51st Street, Kobe
142548809	Nanci	Rucker	nanci.rucker@mavericktechnolog	74 Winter, Friedrichshafe
428653401	Chad	Blaine	chad@mms.com	94 Rourke Ave, Syracuse

1.8.2 טבלת Recipient

Data Generator:

RECIPIENT

Owner	Table	Number of records
SHIMIZRA	RECIPIENT	350

Name	Type	Size	Data
CREDIBILITY	NUMBER		Random(1, 5)
MAXIMALALLOWEDLC	NUMBER		Random(1000, 10000)
GUARANTORID	NUMBER		List(select ID from Person)
ID	NUMBER		List(select ID from Person)

הצגת פקודת Insert into:

Recipient.gd

```
insert into SHIMIZRA.RECIPIENT (CREDIBILITY, MAXIMALALLOWEDLOAN, GUARANTORID, ID)
values (3, 7065, 692524093, 274659397);

insert into SHIMIZRA.RECIPIENT (CREDIBILITY, MAXIMALALLOWEDLOAN, GUARANTORID, ID)
values (1, 5542, 320190174, 922781695);

insert into SHIMIZRA.RECIPIENT (CREDIBILITY, MAXIMALALLOWEDLOAN, GUARANTORID, ID)
values (3, 7518, 383754402, 443736649);

insert into SHIMIZRA.RECIPIENT (CREDIBILITY, MAXIMALALLOWEDLOAN, GUARANTORID, ID)
values (4, 6117, 655986524, 142255146);

insert into SHIMIZRA.RECIPIENT (CREDIBILITY, MAXIMALALLOWEDLOAN, GUARANTORID, ID)
values (2, 5060, 934498106, 767445817);
```

הצגת הנתונים של הטבלה:

RECIPIENT

CREDIBILITY	MAXIMALALLOWEDLOAN	GUARANTORID	ID
5	7544	983901109	421883297
4	8376	358324402	252143561
3	9791	238038479	693362481
3	9251	199720310	300679224
1	1062	947018586	704608454
1	5508	402215503	180801408
5	3225	278208564	656054660

1.8.3 טבלת Clerk

:Data Generator

Clerk.gd

CLERK

Owner: SHIMIZRA | Table: CLERK | Number of records: 300

Name	Type	Size	Data
RATING	NUMBER		Random(1, 10)
SALARY	NUMBER		Random(6000, 12000)
HIREDATE	DATE		Random(01/01/1980, 31/12/2020)
AUTHORIZATION	NUMBER		Random(1, 3)
ID	NUMBER		List(select ID from Person)

הצגת פקודת Insert into:

Clerk.gd

```
insert into SHIMIZRA.CLERK (RATING, SALARY, HIREDATE, AUTHORIZATION, ID)
values (6, 8234, to_date('27-10-1985', 'dd-mm-yyyy'), 1, 754519550);

insert into SHIMIZRA.CLERK (RATING, SALARY, HIREDATE, AUTHORIZATION, ID)
values (5, 8764, to_date('12-04-2003', 'dd-mm-yyyy'), 3, 165582729);

insert into SHIMIZRA.CLERK (RATING, SALARY, HIREDATE, AUTHORIZATION, ID)
values (9, 8923, to_date('11-11-2008', 'dd-mm-yyyy'), 1, 958368699);

insert into SHIMIZRA.CLERK (RATING, SALARY, HIREDATE, AUTHORIZATION, ID)
values (7, 6097, to_date('28-12-1993', 'dd-mm-yyyy'), 3, 784409867);

insert into SHIMIZRA.CLERK (RATING, SALARY, HIREDATE, AUTHORIZATION, ID)
values (1, 10031, to_date('22-09-2002', 'dd-mm-yyyy'), 1, 591730391);
```

הצגת הנתונים של הטבלה:

CLERK

RATING	SALARY	HIREDATE	AUTHORIZATION	ID
4	6633	20-10-1982	2	734114005
3	11732	27-06-1985	3	859180905
8	9811	10-08-1981	1	150304456
9	6434	02-05-1984	2	323033804
1	7558	16-11-2016	3	187749120
7	8858	27-12-1999	3	448749900
9	7292	29-07-2019	3	570459039
2	8397	08-09-1993	1	373181133

1.8.4 טבלת Manager

:Data Generator

Manager.gd

MANAGER

Owner	Table	Number of records
SHIMIZRA	MANAGER	200

Name	Type	Size	Data
SALARYBONUSPERCEN	NUMBER		Random(1, 100)
ID	NUMBER		List(select ID from Clerk)

הצגת פקודת Insert into:

Manager.gd

```
insert into SHIMIZRA.MANAGER (SALARYBONUSPERCENTAGE, ID)
values (54, 659332807);

insert into SHIMIZRA.MANAGER (SALARYBONUSPERCENTAGE, ID)
values (64, 144834961);

insert into SHIMIZRA.MANAGER (SALARYBONUSPERCENTAGE, ID)
values (90, 163024959);

insert into SHIMIZRA.MANAGER (SALARYBONUSPERCENTAGE, ID)
values (84, 934498106);

insert into SHIMIZRA.MANAGER (SALARYBONUSPERCENTAGE, ID)
values (85, 259174931);
```

הצגת הנתונים של הטבלה:

MANAGER

SALARYBONUSPERCENTAGE	ID
73	602461236
81	320971263
81	167685433
57	130633897
79	990603376
77	819186500
79	213960186
14	158671163
37	339612730

1.8.5 טבלת Fund

:Data Generator

FUND

Owner	Table	Number of records
SHIMIZRA	FUND	200

Name	Type	Size	Data
BALANCE	NUMBER		Random(0, 1500)
ADDRESS	VARCHAR2	40	Address2 + ',' + City
NAME	VARCHAR2	25	Company
BRANCHCODE	NUMBER		Sequence(1)

הצגת פקודת Insert into:

```
insert into SHIMIZRA.FUND (BALANCE, ADDRESS, NAME, BRANCHCODE)
values (620, ', San Francisco', 'Kelmoore Investment', 1);

insert into SHIMIZRA.FUND (BALANCE, ADDRESS, NAME, BRANCHCODE)
values (1226, ', Deerfield', 'AT&T Wireless', 2);

insert into SHIMIZRA.FUND (BALANCE, ADDRESS, NAME, BRANCHCODE)
values (850, ', Campinas', 'General Motors Corp.', 3);

insert into SHIMIZRA.FUND (BALANCE, ADDRESS, NAME, BRANCHCODE)
values (649, ', Augsburg', 'Market First', 4);

insert into SHIMIZRA.FUND (BALANCE, ADDRESS, NAME, BRANCHCODE)
values (97, ', Chapel hill', 'ComGlobal Systems', 5);
```

הצגת הנתונים של הטבלה:

FUND

BALANCE	ADDRESS	NAME	BRANCHCODE
32	, Dartmouth	Sammy's Woodfired Pizza	1
328	, Kaiserslautern	RS Information Systems	2
1477	, Monroe	Unicru	3
776	, Horsham	Automated License Systems	4
279	, Lisbon	Progressive Medical	5
62	, Durban	Nestle	6
967	, Rome	Sequoia System Internatio	7
1081	, Oak Park	Baesch Computer Consultin	8
1355	Suite 100, Vista	Access Systems	9

1.8.6 טבלת Loan

:Data Generator

LOAN

Owner	Table	Number of records
SHIMIZRA	LOAN	300

Name	Type	Size	Data
LOANNUMBER	NUMBER		Sequence(1)
DATETAKEN	DATE		Random(01/01/1980, 31/12/2020)
INITIALLOANAMOUNT	NUMBER		Random(1000, 10000)
TOTALNUMBERPAYMI	NUMBER		Random(1, 100)
BRANCHCODE	NUMBER		List(select BranchCode from Fund)
ID	NUMBER		List(select ID from Recipient)

הצגת פקודת Insert into:

```
insert into SHIMIZRA.LOAN (LOANNUMBER, DATETAKEN, INITIALLOANAMOUNT, TOTALNUMBERPAYMENTS, BRANCHCODE, ID)
values (1, to_date('28-01-1980', 'dd-mm-yyyy'), 5364, 55, 44, 138131429);

insert into SHIMIZRA.LOAN (LOANNUMBER, DATETAKEN, INITIALLOANAMOUNT, TOTALNUMBERPAYMENTS, BRANCHCODE, ID)
values (2, to_date('25-05-2005', 'dd-mm-yyyy'), 5085, 1, 10, 323033804);

insert into SHIMIZRA.LOAN (LOANNUMBER, DATETAKEN, INITIALLOANAMOUNT, TOTALNUMBERPAYMENTS, BRANCHCODE, ID)
values (3, to_date('09-01-2004', 'dd-mm-yyyy'), 6163, 13, 4, 576365939);

insert into SHIMIZRA.LOAN (LOANNUMBER, DATETAKEN, INITIALLOANAMOUNT, TOTALNUMBERPAYMENTS, BRANCHCODE, ID)
values (4, to_date('16-07-1985', 'dd-mm-yyyy'), 6178, 19, 141, 754519550);

insert into SHIMIZRA.LOAN (LOANNUMBER, DATETAKEN, INITIALLOANAMOUNT, TOTALNUMBERPAYMENTS, BRANCHCODE, ID)
values (5, to_date('10-04-2008', 'dd-mm-yyyy'), 1241, 90, 9, 437874230);
```

הצגת הנתונים של הטבלה:

LOAN

LOANNUMBER	DATETAKEN	INITIALLOANAMOUNT	TOTALNUMBERPAYMENTS	BRANCHCODE	ID
1	13-08-2010	9590	51	117	776802949
2	17-07-1987	9956	44	15	115583228
3	26-07-1987	4797	92	114	191866376
4	19-12-1999	1323	42	3	619894511
5	26-06-2008	9129	51	112	640223822
6	19-11-2007	8288	46	47	985888986

1.8.7 טבלת Payments

:Data Generator

PAYMENTS

Owner: SHIMIZRA | Table: PAYMENTS | Number of records: 250

Name	Type	Size	Data
NUMBERPAYMENT	NUMBER		Sequence(1)
SUMPAYMENT	NUMBER		Random(1, 1000)
DATEPAYMENT	DATE		Random(01/01/1980, 31/12/2020)
LOANNUMBER	NUMBER		List(select LoanNumber from Loan)

הצגת פקודת Insert into:

```
insert into SHIMIZRA.PAYMENTS (NUMBERPAYMENT, SUMPAYMENT, DATEPAYMENT, LOANNUMBER)
values (1, 935, to_date('23-12-2001', 'dd-mm-yyyy'), 117);

insert into SHIMIZRA.PAYMENTS (NUMBERPAYMENT, SUMPAYMENT, DATEPAYMENT, LOANNUMBER)
values (2, 202, to_date('04-12-1988', 'dd-mm-yyyy'), 50);

insert into SHIMIZRA.PAYMENTS (NUMBERPAYMENT, SUMPAYMENT, DATEPAYMENT, LOANNUMBER)
values (3, 953, to_date('08-12-2019', 'dd-mm-yyyy'), 7);

insert into SHIMIZRA.PAYMENTS (NUMBERPAYMENT, SUMPAYMENT, DATEPAYMENT, LOANNUMBER)
values (4, 707, to_date('06-07-1993', 'dd-mm-yyyy'), 180);

insert into SHIMIZRA.PAYMENTS (NUMBERPAYMENT, SUMPAYMENT, DATEPAYMENT, LOANNUMBER)
values (5, 764, to_date('17-02-1986', 'dd-mm-yyyy'), 134);
```

הצגת הנתונים של הטבלה:

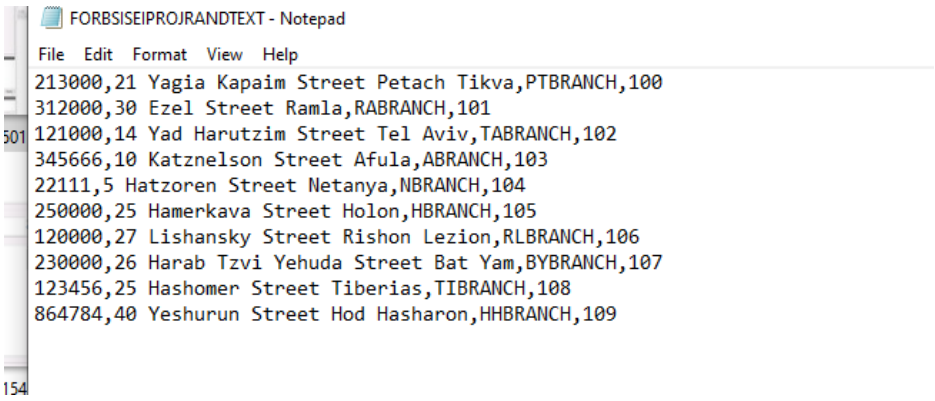
PAYMENTS

NUMBERPAYMENT	SUMPAYMENT	DATEPAYMENT	LOANNUMBER
1	186	24-07-1988	34
2	260	31-03-1992	76
3	139	05-02-1991	267
4	470	14-04-2003	30
5	502	29-09-1990	4
6	882	23-11-1980	232
7	523	07-06-1991	134

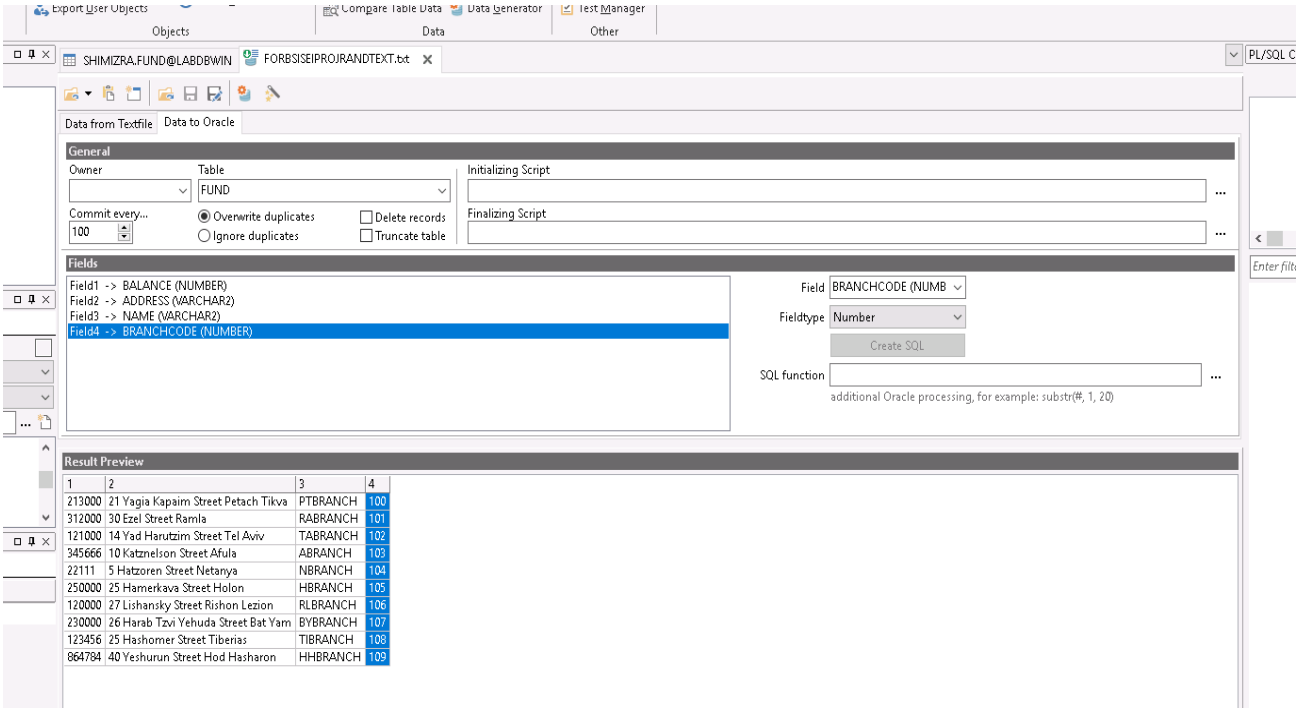
1.9 יבוא ע"י Text Importer

1.9.1 יבוא טבלת Fund

קובץ txt עבור טבלת Fund:



הגדרות היבוא:



קובץ insert שיוצר:

scriptfund - Notepad

File Edit Format View Help

```
insert into fund (balance, address, name, branchcode)
values (213000, '21 Yagia Kapaim Street Petach Tikva', 'PTBRANCH', 100);

insert into fund (balance, address, name, branchcode)
values (312000, '30 Ezel Street Ramla', 'RABRANCH', 101);

insert into fund (balance, address, name, branchcode)
values (121000, '14 Yad Harutzim Street Tel Aviv', 'TABRANCH', 102);

insert into fund (balance, address, name, branchcode)
values (345666, '10 Katznelson Street Afula', 'ABRANCH', 103);

insert into fund (balance, address, name, branchcode)
values (22111, '5 Hatzoren Street Netanya', 'NBRANCH', 104);

commit;

insert into fund (balance, address, name, branchcode)
values (250000, '25 Hamerkava Street Holon', 'HBRANCH', 105);

insert into fund (balance, address, name, branchcode)
values (120000, '27 Lishansky Street Rishon Lezion', 'RLBRANCH', 106);

insert into fund (balance, address, name, branchcode)
values (230000, '26 Harab Tzvi Yehuda Street Bat Yam', 'BYBRANCH', 107);

insert into fund (balance, address, name, branchcode)
values (123456, '25 Hashomer Street Tiberias', 'TIBRANCH', 108);

insert into fund (balance, address, name, branchcode)
values (864784, '40 Yeshurun Street Hod Hasharon', 'HHBRANCH', 109);

commit;
```

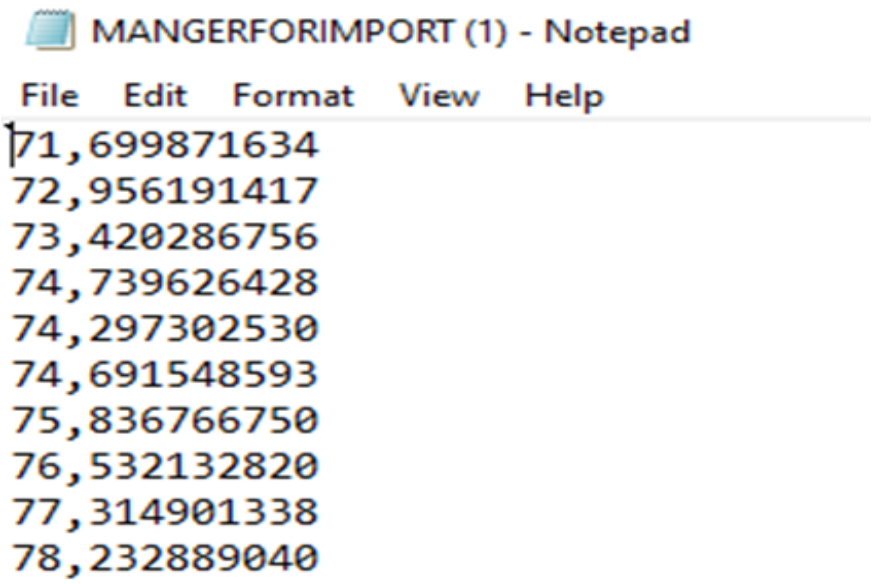
נוודא שהיבוא צלח:

```
select * from fund order by branchcode
```

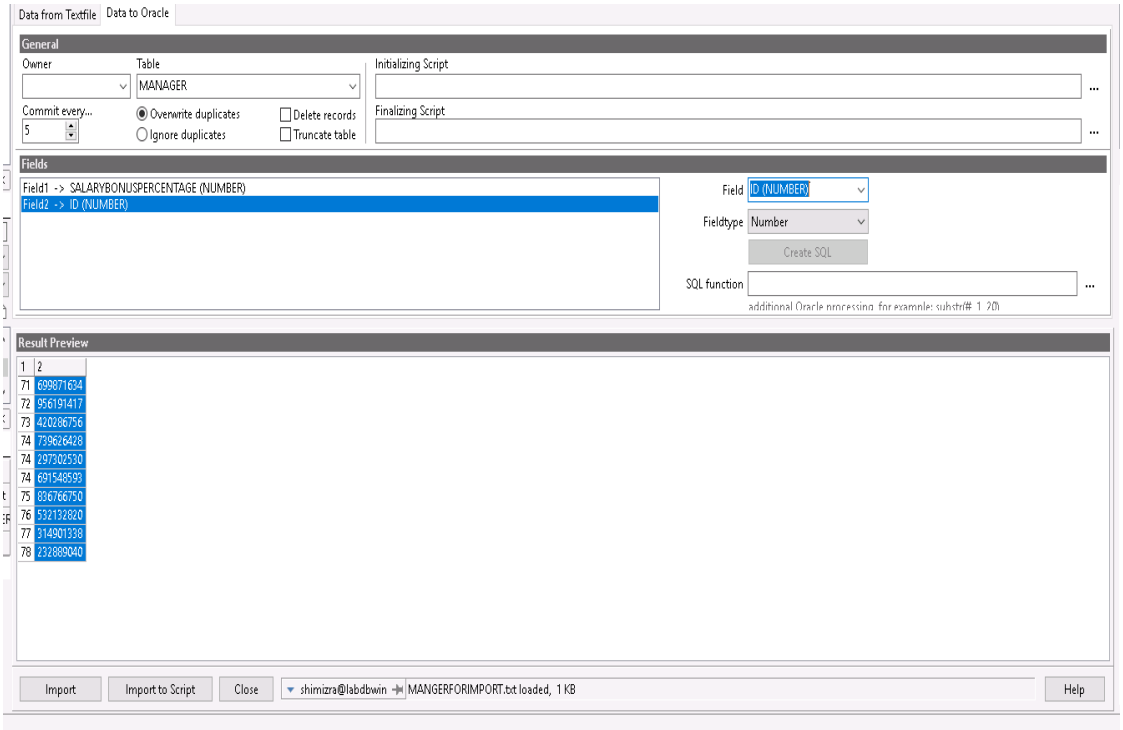
	BALANCE	ADDRESS	NAME	BRANCHCODE
100	213000	21 Yagia Kapaim Street Petach Tikva	PTBRANCH	100
101	312000	30 Ezel Street Ramla	RABRANCH	101
102	121000	14 Yad Harutzim Street Tel Aviv	TABRANCH	102
103	345666	10 Katznelson Street Afula	ABRANCH	103
104	22111	5 Hatzoren Street Netanya	NBRANCH	104
105	250000	25 Hamerkava Street Holon	HBRANCH	105
106	120000	27 Lishansky Street Rishon Lezion	RLBRANCH	106
107	230000	26 Harab Tzvi Yehuda Street Bat Yam	BYBRANCH	107
108	123456	25 Hashomer Street Tiberias	TIBRANCH	108
109	864784	40 Yeshurun Street Hod Hasharon	HHBRANCH	109

1.9.2 יבוא קובץ טקסט של Manager

קובץ txt:



הגדרות היבוא:



נוודא שהיבוא צלח:

```
select * from manager order by salarybonuspercentage
```

	SALARYBONUSPERCENTAGE	ID
106	71	800264620
107	71	312162379
108	71	206324745
109	72	956191417
110	72	765108935
111	73	406366356
112	73	420286756
113	73	245808338
114	73	158333669
115	73	512222332
116	74	739626428
117	74	297302530
118	74	691548593
119	75	130633897
120	75	836766750
121	75	940085223
122	75	237357477
123	76	843003478
124	76	676804174
125	76	532132820
126	77	644927493
127	77	314901338
128	78	232889040

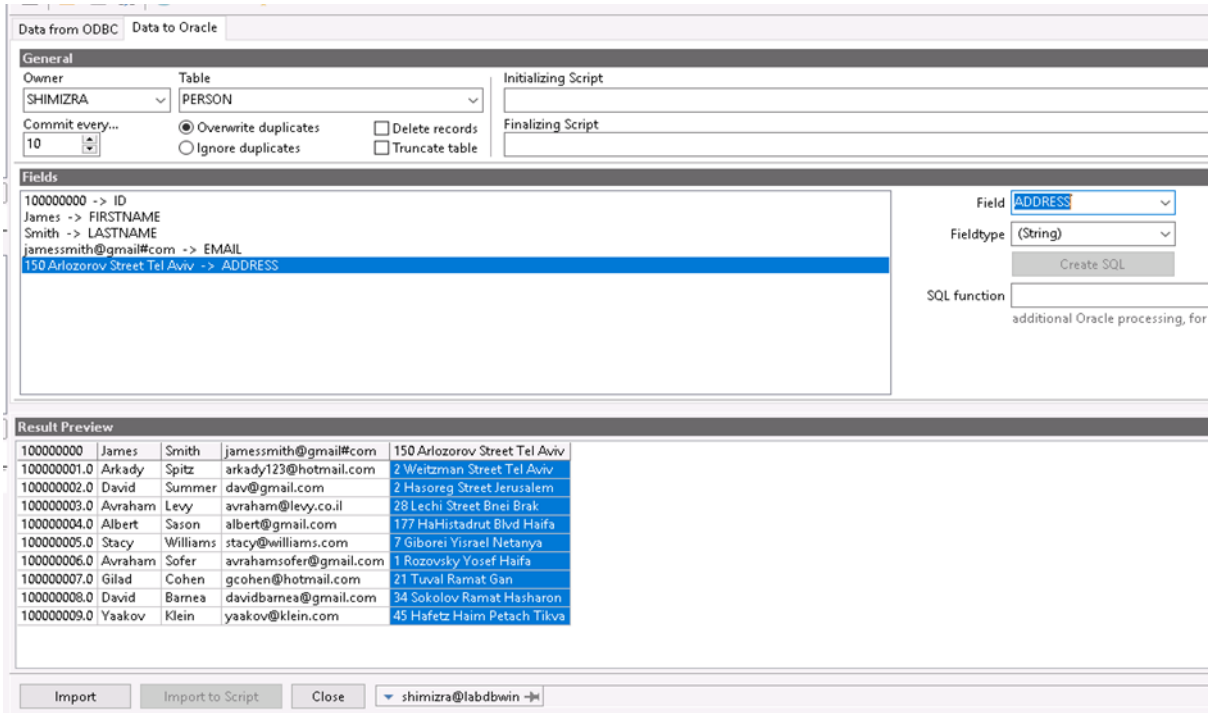
1.10 יבוא דרך ODBC Importer

1.10.1 יבוא של טבלת Person

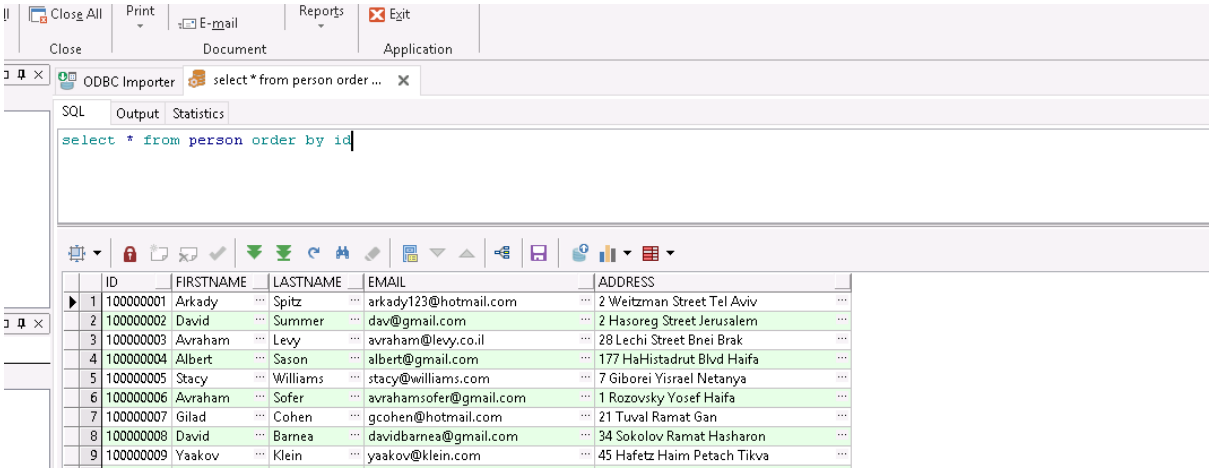
קובץ אקסל:

A	B	C	D	E
100000000	James	Smith	jamessmith@gmail.com	150 Arlozorov Street Tel Aviv
100000001	Arkady	Spitz	arkady123@hotmail.com	2 Weitzman Street Tel Aviv
100000002	David	Summer	dav@gmail.com	2 Hasoreg Street Jerusalem
100000003	Avraham	Levy	avraham@levy.co.il	28 Lechi Street Bnei Brak
100000004	Albert	Sason	albert@gmail.com	177 HaHistadrut Blvd Haifa
100000005	Stacy	Williams	stacy@williams.com	7 Giborei Yisrael Netanya
100000006	Avraham	Sofer	avrahamsofer@gmail.com	1 Rozovsky Yosef Haifa
100000007	Gilad	Cohen	gcohen@hotmail.com	21 Tuval Ramat Gan
100000008	David	Barnea	davidbarnea@gmail.com	34 Sokolov Ramat Hasharon
100000009	Yaakov	Klein	yaakov@klein.com	45 Hafetz Haim Petach Tikva

הגדרות הייבוא:



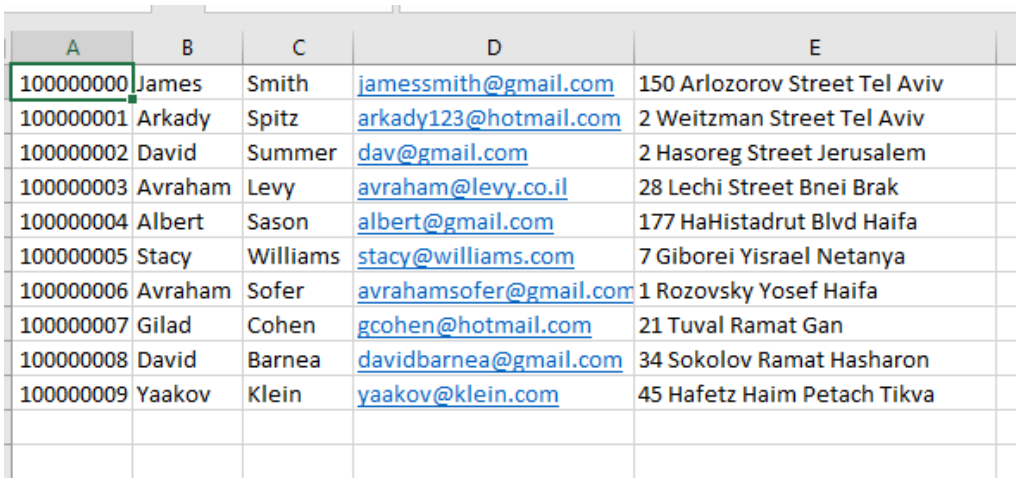
נוודא שהיבוא צלח:



	ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
1	100000001	Arkady	Spitz	arkady123@hotmail.com	2 Weitzman Street Tel Aviv
2	100000002	David	Summer	dav@gmail.com	2 Hasoreg Street Jerusalem
3	100000003	Avraham	Levy	avraham@levy.co.il	28 Lechi Street Bnei Brak
4	100000004	Albert	Sason	albert@gmail.com	177 HaHistadrut Blvd Haifa
5	100000005	Stacy	Williams	stacy@williams.com	7 Giborei Yisrael Netanya
6	100000006	Avraham	Sofer	avrahamsofer@gmail.com	1 Rozovsky Yosef Haifa
7	100000007	Gilad	Cohen	gcohen@hotmail.com	21 Tuval Ramat Gan
8	100000008	David	Barnea	davidbarnea@gmail.com	34 Sokolov Ramat Hasharon
9	100000009	Yaakov	Klein	yaakov@klein.com	45 Hafetz Haim Petach Tikva

1.10.2 יבוא של טבלת Clerk

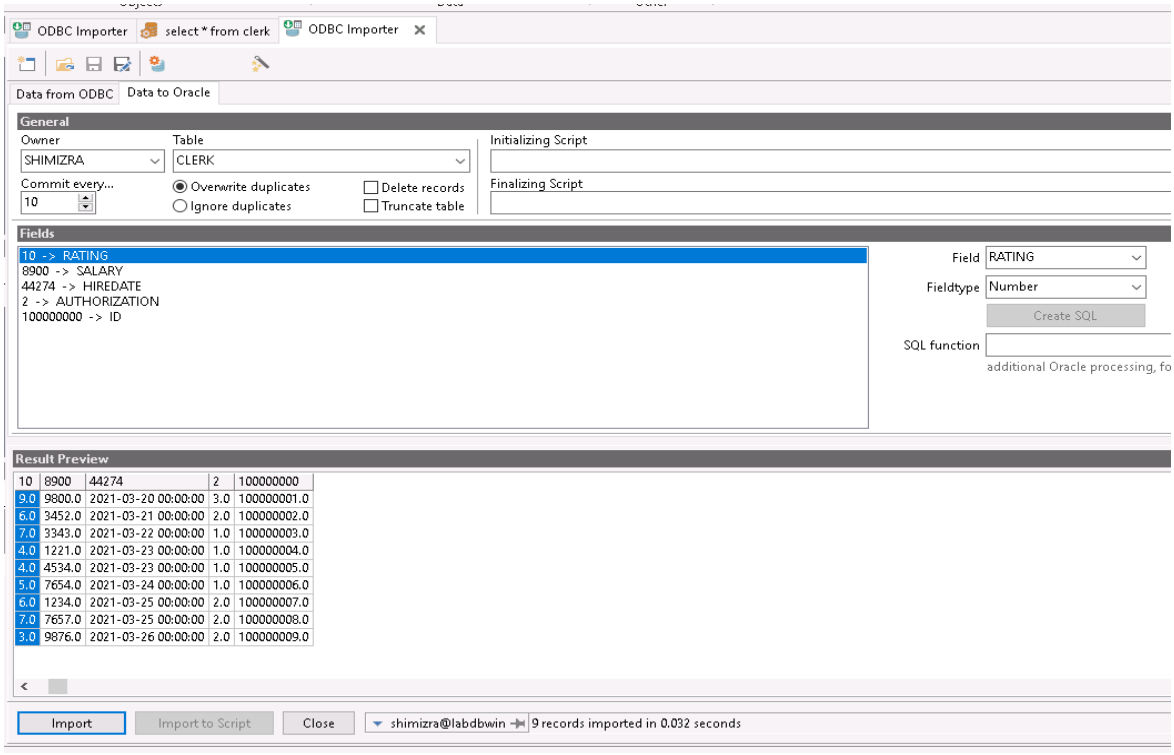
קובץ אקסל של Clerk:



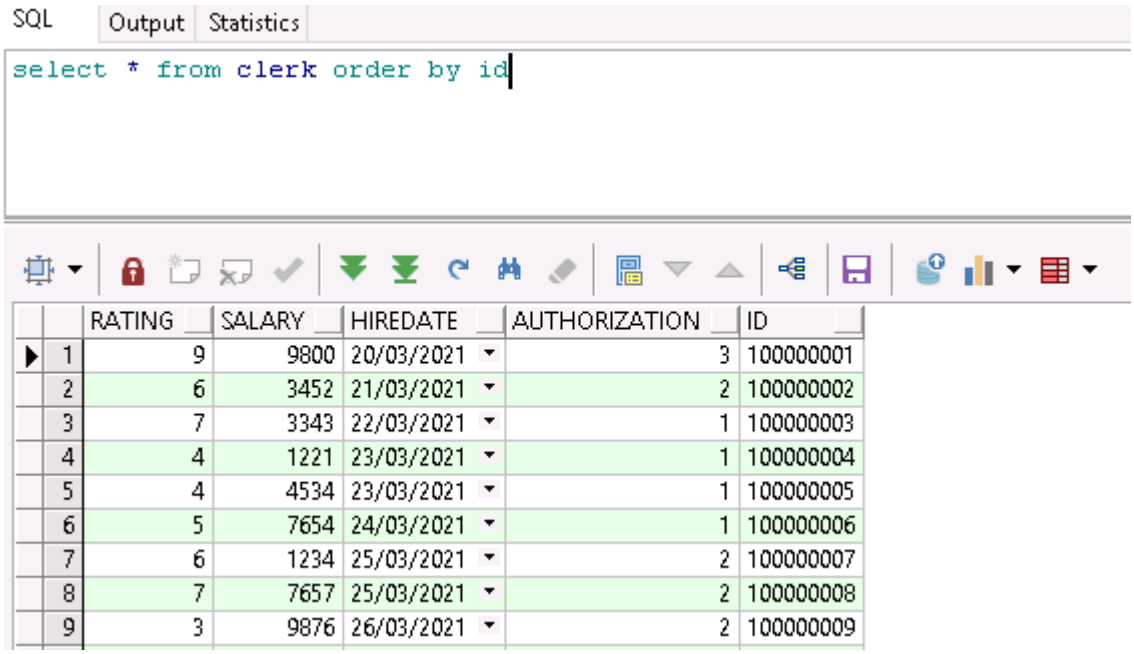
A	B	C	D	E
100000000	James	Smith	jamessmith@gmail.com	150 Arlozorov Street Tel Aviv
100000001	Arkady	Spitz	arkady123@hotmail.com	2 Weitzman Street Tel Aviv
100000002	David	Summer	dav@gmail.com	2 Hasoreg Street Jerusalem
100000003	Avraham	Levy	avraham@levy.co.il	28 Lechi Street Bnei Brak
100000004	Albert	Sason	albert@gmail.com	177 HaHistadrut Blvd Haifa
100000005	Stacy	Williams	stacy@williams.com	7 Giborei Yisrael Netanya
100000006	Avraham	Sofer	avrahamsofer@gmail.com	1 Rozovsky Yosef Haifa
100000007	Gilad	Cohen	gcohen@hotmail.com	21 Tuval Ramat Gan
100000008	David	Barnea	davidbarnea@gmail.com	34 Sokolov Ramat Hasharon
100000009	Yaakov	Klein	yaakov@klein.com	45 Hafetz Haim Petach Tikva

בס"ד מיני פרויקט בסיסי נתונים גמ"ח לב, בנק הלוואות ללא ריבית

הגדרות היבוא:



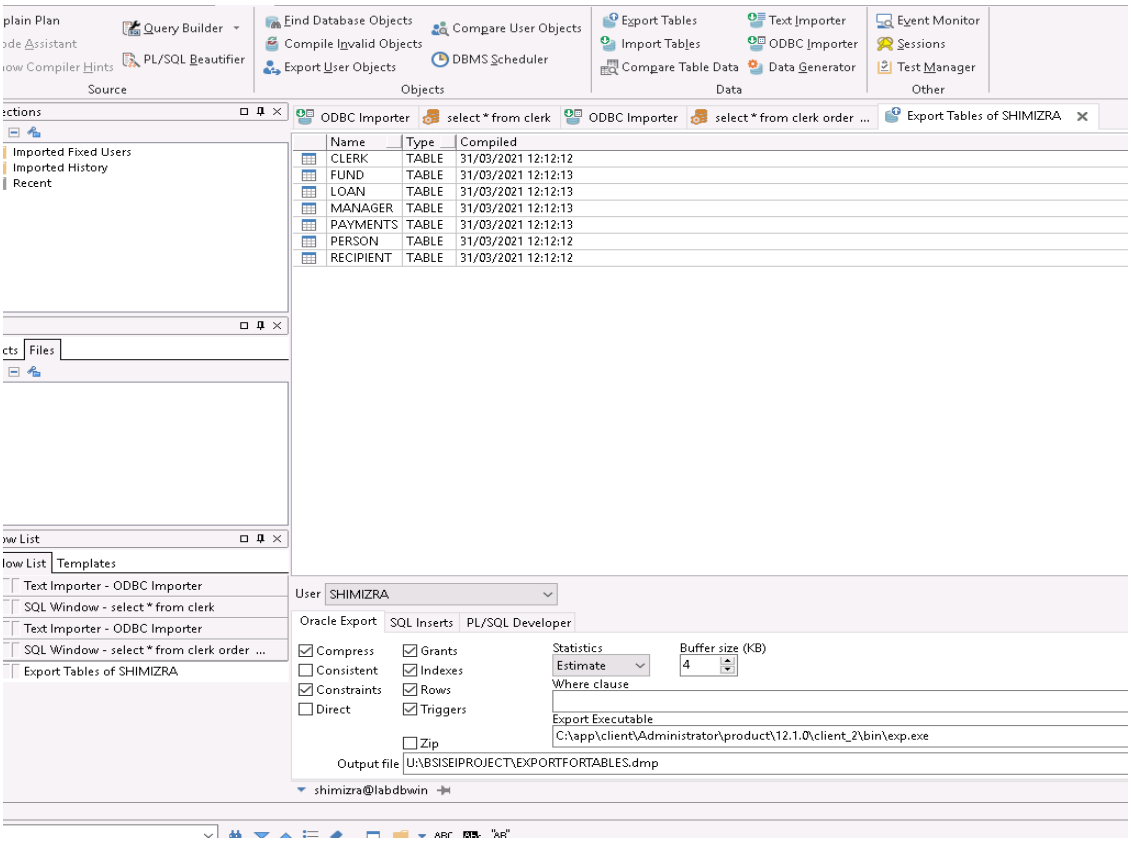
נוודא שהיבוא צלח:



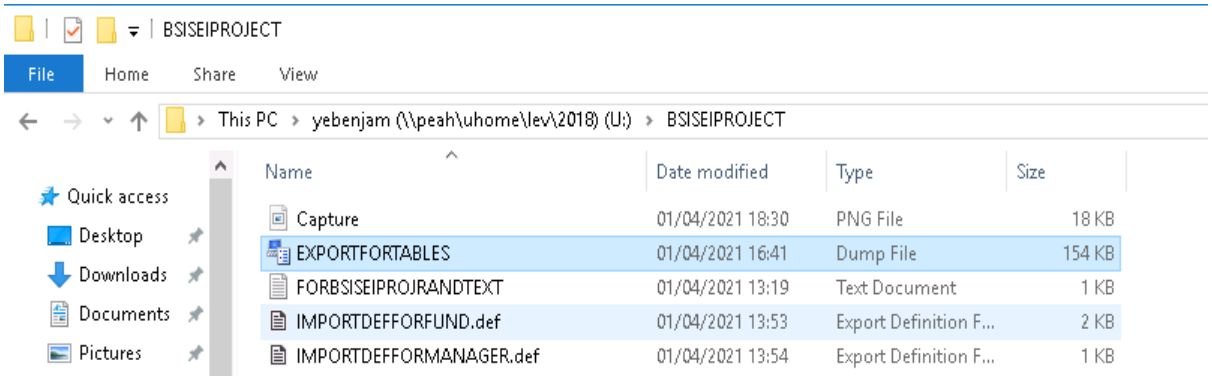
	RATING	SALARY	HIREDATE	AUTHORIZATION	ID
1	9	9800	20/03/2021	3	100000001
2	6	3452	21/03/2021	2	100000002
3	7	3343	22/03/2021	1	100000003
4	4	1221	23/03/2021	1	100000004
5	4	4534	23/03/2021	1	100000005
6	5	7654	24/03/2021	1	100000006
7	6	1234	25/03/2021	2	100000007
8	7	7657	25/03/2021	2	100000008
9	3	9876	26/03/2021	2	100000009

1.11 יצוא בשיטת Oracle Express

נייצא את כל הטבלאות :



קיבלנו קובץ גיבוי:



1.12 שאילתות

1.12.1 קידום עובדים מצטיינים

תיאור: מטרת השאילתה להחזיר את העובדים אשר מדורגים הכי גבוהה וכמו כן קיבלו את ההרשאות הגבוהות ביותר אבל הם עדיין לא קודמו למנהלים וכן משכורתם לא עולה על סכום 10,000 בחודש.

הקוד ב – SQL:

```
select *
from clerk c
where c.ID in
        ((
         select ID
         from clerk c1
         where c1.Rating >= all (
                        select c2.Rating
                        from clerk c2)
         intersect

          select c3.ID
          from clerk c3
          where c3.Authorization >= all (
                        select c4.Authorization
                        from clerk c4))
         minus

         (select c5.ID
         from clerk c5
         where c5.ID  in(
                  select m.ID
                  from manager m)))
and c.Salary < '10000'
order by hiredate
```

תוצאת הריצה:

	RATING	SALARY	HIREDATE	AUTHORIZATION	ID
1	10	8026	21/08/2004	3	6458107866
2	10	9330	22/06/2011	3	9179612224
3	10	7716	08/09/2016	3	1128890106
4	10	7570	05/07/2020	3	2953186314

14:24 yebenjam@labdbwin 4 rows selected in 0.021 seconds

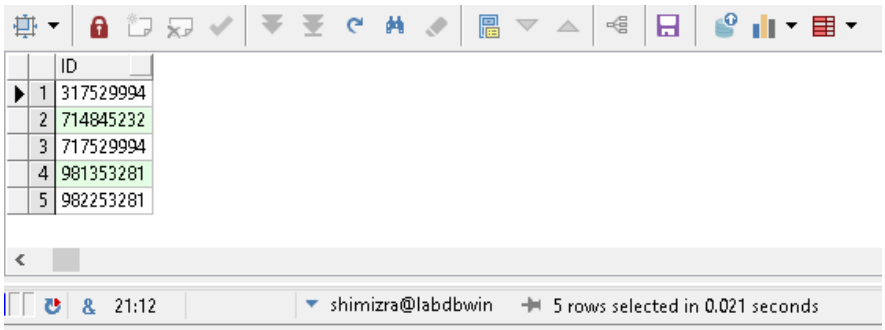
1.12.2 נאמנות הלווה והערב שלו

תיאור: מטרת השאילתה להחזיר את הלווים שגרים בניו יורק, אשר נמצא שהם נאמנים וגם הערבים שלהם נאמנים. או לחילופין השאילתה תחזיר את הלווים אשר הם לא נאמנים, אבל הם גרים בניו יורק וגם הערבים שלהם גרים בניו יורק.

הקוד ב – SQL:

```
    select p1.id
from (
    select p.Id
    from Person p
    where p.Address like '%New York'
    and p.id in(
            select r1.id
            from Recipient r1
            where (r1.credibility <> '1'
            and r1.guarantorid not in(
                        select r2.id
                        from recipient r2
                        where r2.credibility = '1'
                        ))
             or (r1.credibility = '1'
             and r1.guarantorid in(
                        select p2.id
                        from Person p2
                        where p2.Address like '%New York'
                        )))) p1
order by id
```

תוצאת הריצה:



	ID
1	317529994
2	714845232
3	717529994
4	981353281
5	982253281

21:12 shimizra@labdbwin 5 rows selected in 0.021 seconds

1.12.3 לווים ששילמו את רוב חובותיהם

תיאור: מטרת השאילתה הינה להחזיר נתונים על כל הלווים ששילמו יותר מ50 אחוז מסך כל ההלוואות שלקחו עד כה.

הקוד ב – SQL:

```
select  ID, LastName ,FirstName ,L.Total_Loans_Amount, P.Total_Payments,
        (L.Total_Loans_Amount - P.Total_Payments) as Rest_To_Pay,
        round((P.Total_Payments / L.Total_Loans_Amount)*100,0) as The_Percent
from
       (select ID, sum(SumPayment) as Total_Payments
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
    natural join
        Person
where (P.Total_Payments / L.Total_Loans_Amount) > 0.5
order by LastName, FirstName
```

תוצאת הריצה:

	ID	LASTNAME	FIRSTNAME	TOTAL_LOANS_AMOUNT	TOTAL_PAYMENTS	REST_TO_PAY	THE_PERCENT
1	384994481	Broderick	Kasey	2592	1659	933	64
2	589078366	Butler	Victoria	1034	1352	-318	131
3	507145453	Caan	Ivan	1001	1073	-72	107
4	301015308	Hewitt	Franco	5384	3644	1740	68
5	158333669	Lopez	William	1900	1363	537	72
6	202679017	Lynne	Rodney	2333	2005	328	86
7	847427209	Price	Val	4028	3028	1000	75

15:29 shimizra@labdbwin 10 rows selected in 0.015 seconds

1.12.4 מנהלים שמקבלים פחות ממוצע העובדים הזוטרים

תיאור: קבלת מידע על מנהלים בעלי הרשאות בינוניות ומעלה(2 או 3) שמקבלים פחות מהממוצע של העובדים שאינם מנהלים. השאילתה תיתן עבור כל מנהל את פרטיו האישיים, רמת ההרשאות, המשכורת, ההפרש בין ממוצע משכורות עבור אלו שאינם מנהלים לבין משכורתו, ובכמה אחוזים פחות הוא מקבל מהממוצע הנ"ל.

הקוד ב – SQL

```
select firstname,lastname,authorization,round(salary+salary*0.01*salarybonuspercentage,2)
as MANAGER_SALARY,
round((select avg(salary)  from clerk where id
not in
(select id from manager)
),2) as AVG_CLERK_SALARY,
round((select avg(salary) from clerk where id not in (select id from manager))-
(salary+salary*0.01*salarybonuspercentage),2) as DIFFERENCE,

round(((select avg(salary) from clerk where id not in (select id from manager))-
(salary+salary*0.01*salarybonuspercentage))/
 (select avg(salary) from clerk where id not in (select id from manager))*100,2) as
Percentage_Less_Clerk_Avg

from clerk natural join person NATURAL JOIN manager
where  salary+salary*0.01*salarybonuspercentage<
(select AVG(salary) from clerk where id not in (select id from manager))
and
authorization>ANY(select authorization from clerk)
order by percentage_less_clerk_avg desc
```

תוצאת הריצה:

	FIRSTNAME	LASTNAME	AUTHORIZATION	MANAGER_SALARY	AVG_CLERK_SALARY	DIFFERENCE	PERCENTAGE_LESS_CLERK_AVG
1	Charlize	Gosdin	2	9072	53917.69	44845.69	83.17
2	Angie	Niven	2	10352	53917.69	43565.69	80.8
3	Chubby	Todd	2	10537.45	53917.69	43380.24	80.46
4	Tara	Borden	2	10629.36	53917.69	43288.33	80.29
5	Sharon	Rodgers	3	10776.48	53917.69	43141.21	80.01
6	Jude	Tilly	2	11750.55	53917.69	42167.14	78.21
7	Philip	Ball	3	12453.96	53917.69	41463.73	76.9

16:40 shimizra@labdbwin 44 rows selected in 0.059 seconds

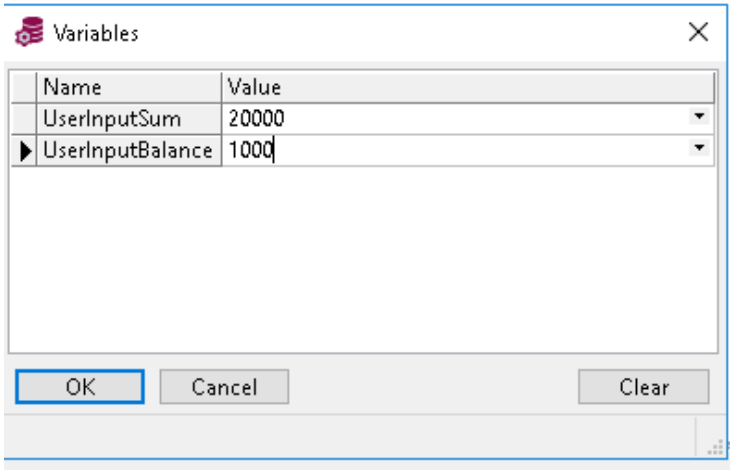
1.12.5 הבנקים שהלוו הכי הרבה כסף לפי טווח תאריכים

תיאור: מטרת השאלתה היא להחזיר מידע אלו סניפי בנק הלוו הכי הרבה כסף לפי טווח תאריכים החל מ 01/01/1991 עד ל 01/01/2020 לפי בחירת סכום שקובע המשתמש, כמו כן המשתמש יבחר את סניפי הבנק לפי היתרה שנותר להם.

הקוד ב – SQL

```
select branchcode, name, balance, sum_Of_Loan, Num_Of_Loans

from fund f natural join (
        select l.branchcode, sum(l.initialloanamount) as sum_Of_Loan ,count(*) as
Num_Of_Loans
        from loan l
        where l.datetaken between to_date('01-01-1991','dd-mm-yyyy') and
          to_date('01-01-2020','dd-mm-yyyy')
        group by l.branchcode
        having sum(l.initialloanamount) > '&UserInputSum') j

where f.balance < '&UserInputBalance'
order by sum_Of_Loan desc
```

תוצאת הריצה כאשר קלט המשתמש הינו:



תוצאת הריצה:

	BRANCHCODE	NAME	BALANCE	SUM_OF_LOAN	NUM_OF_LOANS
1	12	Royal Gold	825	23973	4
2	33	Mosaic	576	22366	3
3	47	Operational Technologies	166	22022	4
4	18	Hershey Foods Corp.	409	21442	3

2:1 shimizra@labdbwin 4 rows selected in 0.010 seconds

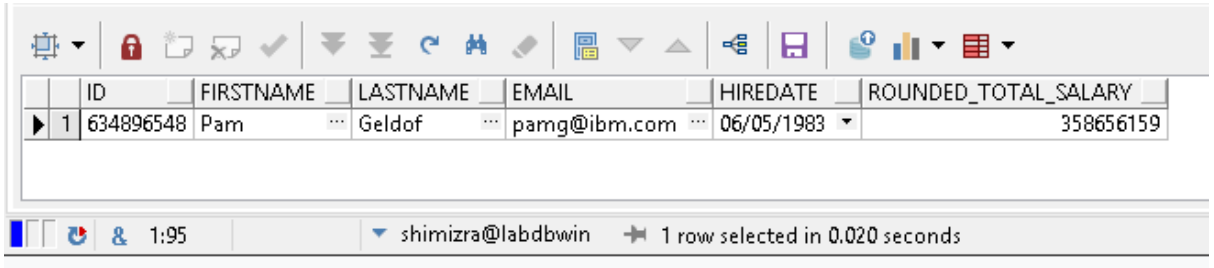
1.12.6 קבלת נתונים של תשלומים למנהל

תיאור: אחזור נתוני המנהל אשר קיבל את סך המשכורות המקסימלי מתחילת עבודתו עד היום.

הקוד ב – SQL:

```
 select id, FirstName, LastName, Email, HireDate, round(Total_Salary,0) as
Rounded_Total_Salary
from Person p natural join  (
                select id, ((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
                *
                months_between(CURRENT_DATE, hireDate)) as Total_Salary
                from manager g natural join clerk c) T1
                natural join Clerk c
where Total_Salary =
                (select max(((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
                *
                months_between(CURRENT_DATE, hireDate)))
                from manager g1 natural join clerk c1)
```

התוצאה שהתקבלה:



	ID	FIRSTNAME	LASTNAME	EMAIL	HIREDATE	ROUNDED_TOTAL_SALARY
1	634896548	Pam	Geldof	pamg@ibm.com	06/05/1983	358656159

1.12.7 הערכת זמן סיום פירעון החוב

תיאור: הערכה לגבי זמן סיום תשלום החוב אם מהיום והלאה הלווה יחזיר כל חודש כמו ההחזר הממוצע ששילם עד עכשיו. השאילתה תחזיר את פרטיו האישיים, מספר ההלוואה, תאריך אחרון להחזר כל החוב, וזמן מוערך לסיום תשלומי החוב באופן המוסבר לעיל.

הקוד ב – SQL:

```
select * from person natural join

(select T.ID as ID, T.LOANNUMBER, T.Loan_Amount-T.Paid_So_Far as Left_To_Pay,

trunc(Add_Months(CURRENT_DATE,ceil((T.Loan_Amount-T.Paid_So_Far)/(T.Min_Paid))))
as Predicted_End_Loan,T.Min_Paid,T.Due_Date from

(select id,min(initialLoanAmount)as Loan_Amount,
min(DateTaken) as Date_Taken, sum(SumPayment) as Paid_So_Far, min(SumPayment) as
Min_Paid, min(TotalNumberPayments) as Num_Payments, LoanNumber,
Add_Months(min(DateTaken),min(TotalNumberPayments)) as Due_Date

from Loan natural join Payments
group by id,LoanNumber)T

where T.LoanNumber in

(select LoanNumber from Loan where Add_Months(DateTaken,TotalNumberPayments)-
CURRENT_DATE>0)
and (T.Paid_So_Far<T.Loan_Amount))
where Predicted_End_Loan>CURRENT_DATE
order by Predicted_End_Loan
```

התוצאה שהתקבלה:

	ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS	LOANNUMBER	LEFT_TO_PAY	PREDICTED_END_LOAN	MIN_PAID	DUE_DATE
1	158333669	William	Lopez	william.lopez@tigris.au	25 Carolyn Street, Perth	226	537	06/07/2021	545	21/04/2023
2	160649811	Saul	Midler	saul.midler@viacom.fr	6 Anderson Road, Toulouse	12	529	06/07/2021	547	01/09/2026
3	937043772	Oded	Solido	oded.solido@qestrel.de	69 Liotta Road, Luedenscheid	217	1876	06/09/2021	838	27/03/2027
4	186364609	Cary	Li	cary.li@mwh.com	54 Maarssen Street, Lexington	11	1719	06/09/2021	645	03/04/2023
5	957395911	Nicholas	Oakenfold	nicholas.oakenfold@kellogg.de	92 Rodgers, Koeln	113	4432	06/11/2021	916	30/01/2025
6	535709318	Famke	Wright	fwright@learningvoyage.uk	1 Cale Drive, Sale	252	1513	06/12/2021	253	09/10/2023
7	259174931	Mindy	Pitt	mindy.pitt@consultants.de	392 Stormare Blvd, Lengdorf	173	4580	06/01/2022	763	11/09/2027
8	271124379	Marina	Tyson	marina.tyson@cocacola.com	55 Mika Street, Pusan-city	92	5182	06/02/2022	697	10/07/2022

1:1 shimizra@labdbwin 15 rows selected in 0.052 seconds

1.12.8 לווים שלא עומדים בתשלומים

תיאור: אחזור פרטים אישיים וסכום החיוב הכללי עבור אנשים בעלי מוסר תשלומים לקוי – אנשים שלא החזירו אף תשלום עבור יותר מהלוואה אחת מההלוואות שלקחו לפני תחילת השנה האזרחית הנוכחית.

הקוד ב – SQL:

```
select * from person natural join (select id, sum(initialloanamount) as Owed_Money

from loan
where DateTaken < to_date('01-01-2021','dd-mm-yyyy')
group by id
having id in
        (select id from loan natural left outer join payments
                where sumpayment IS NULL
                        group by id
                having count(*)>1))

order by OWED_MONEY desc
```

התוצאה שהתקבלה:

	ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS	OWED_MONEY
1	776071203	Jonny	Collette	jonny.collette@randomwalk.com	408 Warrenton Street, Waldorf	28156
2	955623508	Mary	Wong	mary.wong@mathis.ca	54 Shatner Street, Powell River	26353
3	283593811	Balthazar	Moffat	balthazar.moffat@networkdispla	76 de Lancie, Niigata	26340
4	105986204	Loreena	Chaykin	loreena.c@perfectorder.com	613 Chaykin Street, Herndon	20441
5	260589988	Suzy	Plummer	splummer@target.com	43rd Street, Bartlesville	20138
6	167714899	Aidan	Bell	aidan.bell@idas.com	84 Rancho Palos Verdes Road, Melrose par	16463
7	704608454	Rhett	Cassel	rhett.cassel@intel.at	65 Cook Road, Ternitz	13965
8	756621257	Thin	Vince	thin.vince@procter.com	93 Crystal, Middletown	13649
9	781419615	Madeline	Cocker	madeline.cocker@appriss.com	23 Fort gordon Street, Slmea	13240

4:16 shimizra@labdbwin 21 rows selected in 0.119 seconds

1.13 שאילתות Update

1.13.1 הוספת מספר תשלומים עבור הלוואות גדולות

תיאור: כל אדם שלקח הלוואה שסכומה גדול מ — 8800 יקבל תשלומים כמספר התשלומים עד עכשיו ועוד חצי כפול מספר התשלומים הקודם.

לפני ביצוע העדכון:

SQL Output Statistics

```
select *
from Loan
where initialLoanAmount > 8800
```

	LOANNUMBER	DATETAKEN	INITIALLOANAMOUNT	TOTALNUMBERPAYMENTS	BRANCHCODE	ID
1	15	03/01/2007	8952	51	140	776071203
2	55	02/04/2007	8960	60	156	138204140
3	64	11/09/2008	8951	79	63	283593811
4	84	06/06/1981	8843	21	116	540351159
5	89	16/04/2017	8919	87	71	159254374
6	173	11/11/2020	9620	82	162	259174931
7	215	17/02/2016	8886	65	55	633667386
8	238	24/06/2014	8860	31	18	928873849

הקוד ב — SQL :

```
UPDATE Loan
SET TotalNumberPayments = TotalNumberPayments + (TotalNumberPayments / 2)
WHERE initialLoanAmount > 8800;
```

לאחר ביצוע העדכון ניראה שכמות התשלומים המקסימלית המותרת להלוואות הנ"ל עלה.

select * from Loan where i ... | SQL Window

SQL | Output | Statistics

```
select *
from Loan
where initialLoanAmount > 8800
```

	LOANNUMBER	DATETAKEN	INITIALLOANAMOUNT	TOTALNUMBERPAYMENTS	BRANCHCODE	ID
1	15	03/01/2007	8952	77	140	776071203
2	55	02/04/2007	8960	90	156	138204140
3	64	11/09/2008	8951	119	63	283593811
4	84	06/06/1981	8843	32	116	540351159
5	89	16/04/2017	8919	131	71	159254374
6	215	17/02/2016	8886	98	55	633667386
7	238	24/06/2014	8860	47	18	928873849

1.13.2 העלאות משכורת

הוחלט לתת לעובדים העלאה במשכורת לפי הדירוג שלהם. כל אחד יקבל את המשכורת החדשה לפי הנוסחה: המשכורת הקודמת ועוד 0.1* המשכורת הקודמת * רמת הדירוג.

לפני ביצוע העדכון:

```
select *
from clerk
```

	RATING	SALARY	HIREDATE	AUTHORIZATION	ID
1	10	83881	24/06/1993	2	699871634
2	6	52140	06/05/1983	1	634896548
3	4	34150	11/05/2001	2	283593811
4	1	8644	18/10/2002	3	320971263
5	2	13918	26/03/1981	2	797759641
6	7	53379	18/07/1981	3	643887578
7	6	55788	25/10/2005	2	499591417
8	1	8520	24/02/2020	2	683441311
9	10	77962	27/02/2020	1	708677061
10	10	104272	03/09/1980	2	956191417

הקוד ב – SQL:

```
UPDATE Clerk
SET Salary = Salary + Salary * 0.01 * Rating;
```

לאחר ביצוע העדכון נקבל:

```
select *
from clerk
```

	RATING	SALARY	HIREDATE	AUTHORIZATION	ID
1	10	92269	24/06/1993	2	699871634
2	6	55268	06/05/1983	1	634896548
3	4	35516	11/05/2001	2	283593811
4	1	8730	18/10/2002	3	320971263
5	2	14196	26/03/1981	2	797759641
6	7	57116	18/07/1981	3	643887578
7	6	59135	25/10/2005	2	499591417
8	1	8605	24/02/2020	2	683441311
9	10	85758	27/02/2020	1	708677061
10	10	114699	03/09/1980	2	956191417

1.14 שאילתות Delete

1.14.1 מחיקת חובות ישנים

הגמ"ח התייאש מלגבות שאר החוב עבור חובות שזמן האחרון לפירעון שלהם היה לפני חמש עשרה שנה, ולכן החליט למחוק רישומי חובות אלו.

לפני המחיקה נקבל ע"י הקוד ב — SQL.

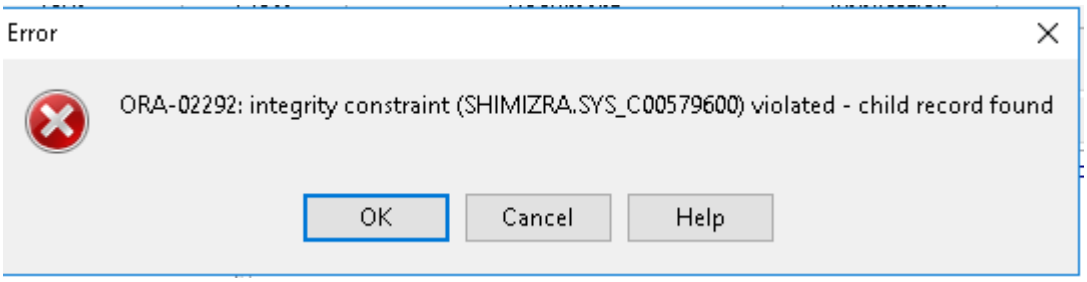
```
select * from loan where ((CURRENT_DATE-
ADD_MONTHS(DateTaken,TotalNumberPayments))/365)>15
```

	LOANNUMBER	DATETAKEN	INITIALLOANAMOUNT	TOTALNUMBERPAYMENTS	BRANCHCODE	ID
1	1	20/06/1990	3974	24	147	985888986
2	2	17/04/1981	4489	23	42	183954065
3	3	13/08/1996	3793	99	24	824144990
4	4	05/09/1987	4433	70	143	300153670
5	6	29/06/1990	904	73	197	751812019
6	9	04/09/1996	3825	47	125	781419615
7	17	01/05/1991	6015	22	63	597831737
8	18	07/02/2000	5647	31	8	508408145
9	19	21/11/1989	6434	78	174	307000637
10	24	03/08/1996	4964	87	101	655986524
11	25	18/03/1994	3810	59	16	282626834

ננסה למחוק על ידי השאילתה הבאה:

```
delete from loan where ((CURRENT_DATE-
ADD_MONTHS(DateTaken,TotalNumberPayments))/365)>15
```

ניסיון זה נכשל ומופיעה האזהרה הבאה:



כלומר לא ניתן למחוק רשומות מטבלה Loan כי יש לטבלה בן Payments שיש לו שורות שמתייחסות להלוואות המיועדות למחיקה, לכן נמחק קודם את השורות בPayment שמצביעות על הערכים המיועדים למחיקה ורק אז מחיקה מטבלת Loan.

הקוד ב — SQL:

```
delete from Payments where LoanNumber in(select LoanNumber from Loan where

((CURRENT_DATE-ADD_MONTHS(DateTaken,TotalNumberPayments))/365)>15)
```

נבדוק האם המחיקה הצליחה.

```
select * from Payments where LoanNumber in(select LoanNumber from Loan where ((CURRENT_DATE-ADD_MONTHS(DateTaken,TotalNumberPayments))/365)>15)
```

NUMBERPAYMENT	SUMPAYMENT	DATEPAYMENT	LOANNUMBER

ניתן לראות שהמחיקה הצליחה.

shimizra@labdbwin 101 rows deleted in 0.012 seconds

כעת ננסה שוב למחוק את הערכים מטבלת Loan כדלעיל.

```
delete from loan where ((CURRENT_DATE-ADD_MONTHS(DateTaken,TotalNumberPayments))/365)>15
```

1:89 shimizra@labdbwin 152 rows deleted in 0.011 seconds

זה צלח. נוודא זאת באמצעות פקודת select.

```
select * from loan where ((CURRENT_DATE-ADD_MONTHS(DateTaken,TotalNumberPayments))/365)>15
```

LOANNUMBER	DATETAKEN	INITIALLOANAMOUNT	TOTALNUMBERPAYMENTS	BRANCHCODE	ID

1:9 shimizra@labdbwin 0 rows selected in 0.017 seconds

הפעולה בוצעה בהצלחה.

1.14.2 פיטורי עובדים שאינם איכותיים

עקב העלויות הגבוהות של תפעול הגמ"ח הוחלט לפטר את העובדים שמזלזלים בעבודתם (עובדים שמונו לפני 1990 ובעלי דירוג פחות מחמש)

לפני המחיקה נקבל ע"י הקוד ב – SQL:

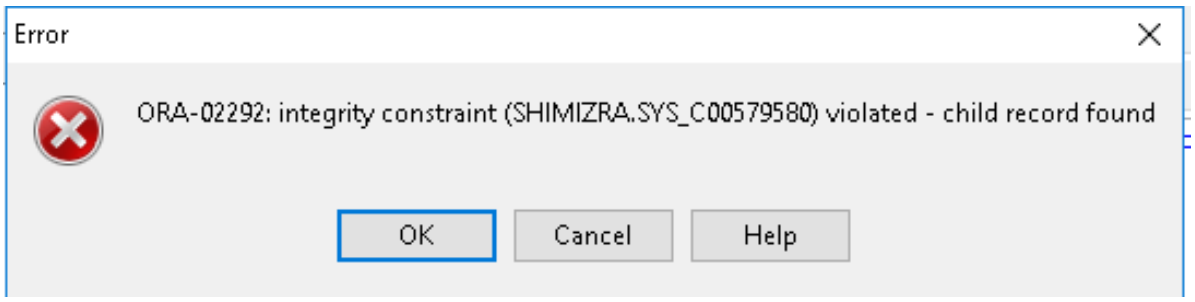
```
select * from clerk where hiredate<to_date('01-01-1990','dd-mm-yyyy') and rating<5
```

	RATING	SALARY	HIREDATE	AUTHORIZATION	ID
1	2	14196	26/03/1981	2	797759641
2	2	24099	09/02/1984	2	930583195
3	2	14122	09/04/1989	3	119368173
4	3	37382	03/11/1981	3	239371713
5	3	27080	28/03/1989	1	712017578
6	3	26159	21/10/1983	2	618708413
7	3	28319	28/11/1985	2	380855405
8	3	19995	15/12/1986	2	101176724
9	3	28213	05/12/1984	3	470356344
10	3	30582	22/11/1983	2	530842062

ננסה למחוק על ידי השאילתה הבאה:

```
delete from clerk where hiredate<to_date('01-01-1990','dd-mm-yyyy') and rating<5
```

מופיעה האזהרה הבאה:

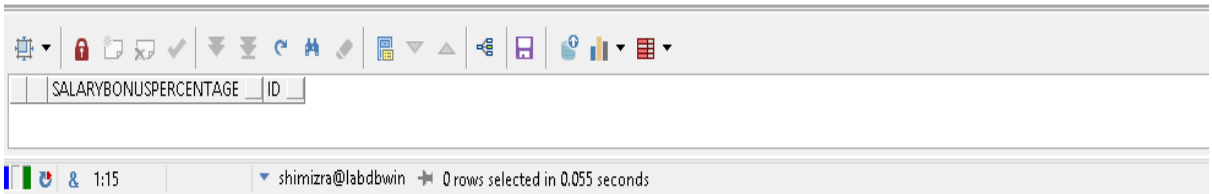


מכיוון שחלק העובדים המיועדים למחיקה נמצאים גם בטבלת הבן של Manager.

לכן נמחק קודם את השורות המצביעות:

נוודא זאת.

```
select * from manager where id in (select id from clerk where hiredate<to_date('01-01-1990','dd-mm-yyyy') and rating<5)
```



הפעולה בוצעה בהצלחה.

```
delete from clerk where hiredate<to_date('01-01-1990','dd-mm-yyyy') and rating<5
```

1:81 shimizra@labdbwin 22 rows deleted in 0.001 seconds

כעת נמחק את השורות המתאימות מטבלת האב.

הקוד ב SQL למחיקת טבלת manager.

```
delete manager where id in (select id from clerk where hiredate<to_date('01-01-1990','dd-mm-yyyy') and rating<5)
```

2:1 shimizra@labdbwin 13 rows deleted in 0.007 seconds

הפעולה בוצעה.

נוודא שהמחיקה הצליחה.

```
select * from clerk where hiredate<to_date('01-01-1990','dd-mm-yyyy') and rating<5
```

RATING	SALARY	HIREDATE	AUTHORIZATION	ID

1:9 shimizra@labdbwin 0 rows selected in 0.011 seconds

הפעולה בוצעה בהצלחה.

1.15 Commit ו Rollback

תיאור: עובד בשם Osment מעוניין להכפיל את משכורתו ב 10 פעמים על מנת לגנוב כסף מהגמ"ח לאחר שווידא כי הוא מפוטר.

המצב לפני ביצוע עדכון המשכורת הינו:

SQL Output Statistics

```
select *
from(
        select * from clerk
        where id ='320971263') C1 natural join Person
```

	ID	RATING	SALARY	HIREDATE	AUTHORIZATION	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
1	320971263	1	8730	18/10/2002	3	Jaime	Osment	jaime@contract.de	12 Craddock Road, Ehningen

ביצוע העדכון:

select * from(select * fr ... | UPDATE clerk SET salary = ... | select * from clerk

SQL Output Statistics

```
UPDATE clerk
SET salary = salary * 10
where id ='320971263';
```

לצערנו העדכון הצליח ונקבל:

select * from(select * fr ... | select * from(select * fr ... | select * from(select * fr ...

SQL Output Statistics

```
select *
from(
        select * from clerk
        where id ='320971263') C1 natural join Person
```

	ID	RATING	SALARY	HIREDATE	AUTHORIZATION	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
1	320971263	1	87300	18/10/2002	3	Jaime	Osment	jaime@contract.de	12 Craddock Road, Ehningen

מנהל הראשי מצא את תכנית הזדונית הנ"ל וביצע Rollback.

```
rollback;

select *
from(
        select * from clerk
        where id ='320971263') C1 natural join Person
```

	ID	RATING	SALARY	HIREDATE	AUTHORIZATION	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
1	320971263	1	8730	18/10/2002	3	Jaime	Osment	jaime@contract.de	12 Craddock Road, Ehningen

עובד החכם יותר היה פועל בדרך הבאה, כך שלמנהל שלו לא יתאפשר לבצע Rollback.

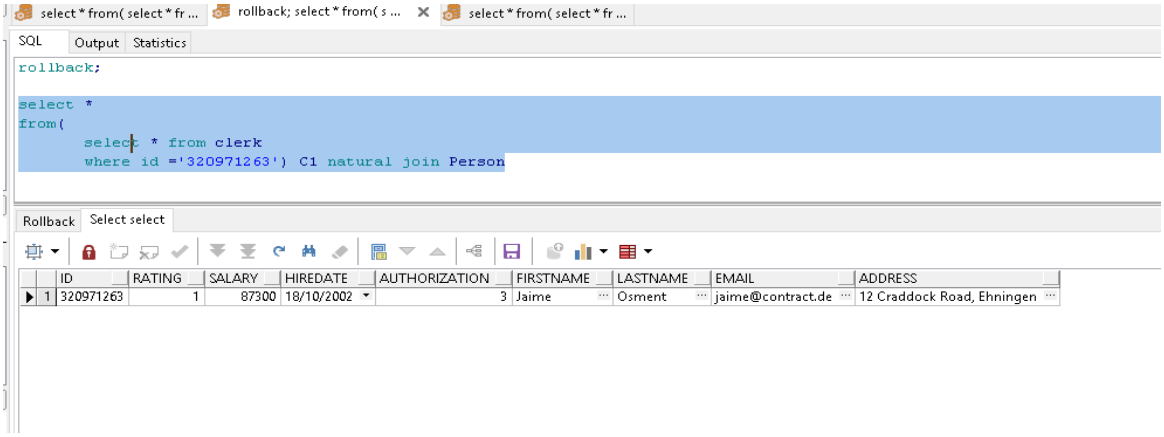
```
UPDATE clerk
SET salary = salary * 10
where id =320971263;
commit;
```

אכן העובד הצליח לקבל משכורת גדולה.

```
select *
from(
        select * from clerk
        where id ='320971263') C1 natural join Person
```

	ID	RATING	SALARY	HIREDATE	AUTHORIZATION	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
1	320971263	1	87300	18/10/2002	3	Jaime	Osment	jaime@contract.de	12 Craddock Road, Ehningen

כעת נראה תרחיש שהמנהל שלו מבצע Rollback ולא מצליח לשחזר למצב הקודם.



חלק ב'

2.1 הרחבת המערכת

מיני פרויקטים שייקחו חלק במערכת המורחבת:

גמ"ח הלוואות ללא ריבית (שמעון מזרחי ויהודה משה בנימין) מערכת המנהלת מעקב על פעילות של גמ"ח הלוואות ללא ריבית.

רשת חנויות (שגב ושילה אפרתי) מערכת המנהלת מידע על פעילות של רשת חנויות לעניין מידע על העובדים קניות מספקים ומכירת מוצרים.

ספריה (עומרי כוכבי ויעקב ברין) מערכת המנהלת מידע על פעילות של ספרייה כגון השאלת ספרים ומידע על מנויים.

2.2 רשימת ישויות

2.2.1 רשימת ישויות שקיימים כבר

מהפרויקט "גמ"ח הלוואות ללא ריבית" ניקח:

Person - ישות המתארת אדם.

Recipient - מקבל הלוואה..

Clerk - ישות המתארת פקיד בגמ"ח.

Manager - ישות המתארת מנהל בגמ"ח.

Fund - ישות המתארת סניף של הגמ"ח.

Loan - ישות המתארת הלוואה.

Payments - ישות המתארת תשלום מתוך החזרים על הלוואה.

מהפרויקט "ספרייה" ניקח:

Book - ישות המתארת ספר.

Category - ישות המתארת קטגוריה של ספרים.

Publisher - ישות המתארת בית הוצאה לאור.

Staff - ישות המתארת עובד של הספריה.

Reader - ישות המתארת קורא בספריה.

מהפרויקט "רשת חנויות" ניקח:

StoreEmployee (**Employees** לשעבר) - ישות המתארת עובד בחנות.

Store - ישות המתארת חנות.

Supplier - ישות המתארת ספק מוצרים לחנויות.

Customer - ישות המתארת לקוח של חנות.

Product - ישות המתארת מוצר.

Sale - ישות המתארת מכירה של אחד או יותר מוצרים בחנות.

2.2.2 קשרים וישויות חדשים שיתווספו

CommisionSale

ישות זו מתארת מכירה של ספר מהספרייה על ידי החנות עם עמלות, כאשר העמלה תעבור כתרומה לגמ"ח. ישות זו תכיל כיל את השדות הבאים:

CommisionPercentage - אחוזי העמלה עבור המכירה.

SalesNumber - מספר עסקה.

Subsidizes- סבסוד של קנייה על ידי הגמ"ח.

BookPurchase - קניית ספר מהספרייה במכירה הנערכת בחנות.

CSIsASale- קשר בין CommissionSale לבין הישות מרשת החנויות שמתארת מכירה-

Sale.

במערכת החדשה יתאפשר לנהל מעקב על הקניות המסובסדות בחנויות על ידי הגמ"ח על ידי הקשר החדש Subsidizes בין הישות של Fund - קרן הגמ"ח לבין הישות Sales שמייצגת מכירה בחנות של רשת החנויות. כך יהיה אפשר לראות נתונים על היקף התמיכה ומידע על הקונים בקניות המסובסדות.

בנוסף תהיה אפשרות חדשה למכור ספרים של הספרייה במכירה פומבית על ידי החנות עבור עמלה. יהיה ניתן לראות נתונים על קטגוריות השכיחות במכירות האלו הרווחים וכדאיות המהלך החדש של המכירות עם עמלות

במערכת החדשה יהיה ניתן לבצע שאילתות על כמות המכירות הפומביות והקשרים בין סוגי

הספרים, המחירים והעמלות המתקבלים על מכירתם.

2.3 אלגוריתם האינטגרציה:

עבור כלל המערכת:

לכל תת מערכת S:

אם יש צורך ביצירת קשרים חדשים וטבלאות חדשות בין S לבין תת מערכת אחרת 'S – בצע זאת

לכל טבלה T בתוך S:

אם צריך את T למערכת המורחבת:

כל אטריבוט A בתוך T יהפוך להיות אטריבוט חדש 'A בתוך טבלה חדשה 'T

עבור כל קשר חדש בין 'T לטבלה אחרת "T

אם יש כבר אטריבוט 'A ב – 'T שמתאים להיות מפתח זר לאטריבוט "A ב – "T :

בצע עדכון לטבלה כך ש – 'A תהיה מפתח זר לאטריבוט "A

אחרת:

הוסף אטריבוט "A ל – 'T והפוך אותה למפתח זר ל – "A ב – "T

לכל אטריבוט 'A ב – 'T:

אם 'A לא נצרך – מחק אותו

אם 'A צריך להפוך להיות חלק מהמפתח הראשי – בצע זאת

כעת נפרט את הפעולות הדרושות על בסיס האלגוריתם הכללי עבור כל אחת מהטבלאות הבאות:

Person- ניקח אותה כפי שהיא – מכיוון שהיא חשובה לייצוג אנשים בגמ"ח וגם עובדי ולקוחות חנות וספריה.

Recipient - ניקח אותה כפי שהיא מכיוון שבמערכת המורחבת נשמרת אופן רישום מקבלי ההלוואות של הגמ"ח כמקודם.

Clerk - ללא שינוי כי המידע על עובדי הגמ"ח לא צריכה להשתנות.

Manager - ללא שינוי בדומה לטבלה הקודמת.

Fund - סניפי הגמ"ח לא ישתנו רק ייתנו מענה גם לסבסוד קניות ולכן אין לשנות ישות זו.

Loan - ללא שינוי כי ההלוואות יהיו אותן סוג של הלוואות כמקודם.

Payments - ללא שינוי בדומה לטבלה הקודמת.

מהפרויקט "ספרייה" ניקח:

Book - פעילות השאלת הספרים תישאר כפי שהייתה. אך תהיה אופציה למכירת ספרים בעמלות, ולכן נוסיף שדה חדש בשם SalesNumber. נדאג לכך שזה יהיה מפתח זר לאטריבוט בעל שם זה שנמצא ב — CommisionSale.

Category - הקטגוריות לא ישתנו ולכן ניקח טבלה זו כפי שהן.

Publisher - הטבלה לא תשתנה בדומה לטבלה הקודמת.

Staff - העובד יהיה ישות חלשה ל – Person של הגמ"ח ולשם כך יש להוסיף לה אטריבוט תעודת זהות (id) שתהיה מפתח זר ל– id ב – Person ששם יהיה שאר פרטיו.

Reader - נעשה בדומה לטבלה הקודמת.

מהפרויקט "רשת חנויות" ניקח:

StoreEmployee– יילקח ללא שינוי מכיוון שהמידע על העובדים בחנות תהיה זהה כמקודם.

Store - הפרטים הפיזיים של החנות לא ישתנו במערכת המורחבת, אלא שגם תהיה אופציה למכירה בעמלות ולכן אין צורך לשנות טבלה זו.

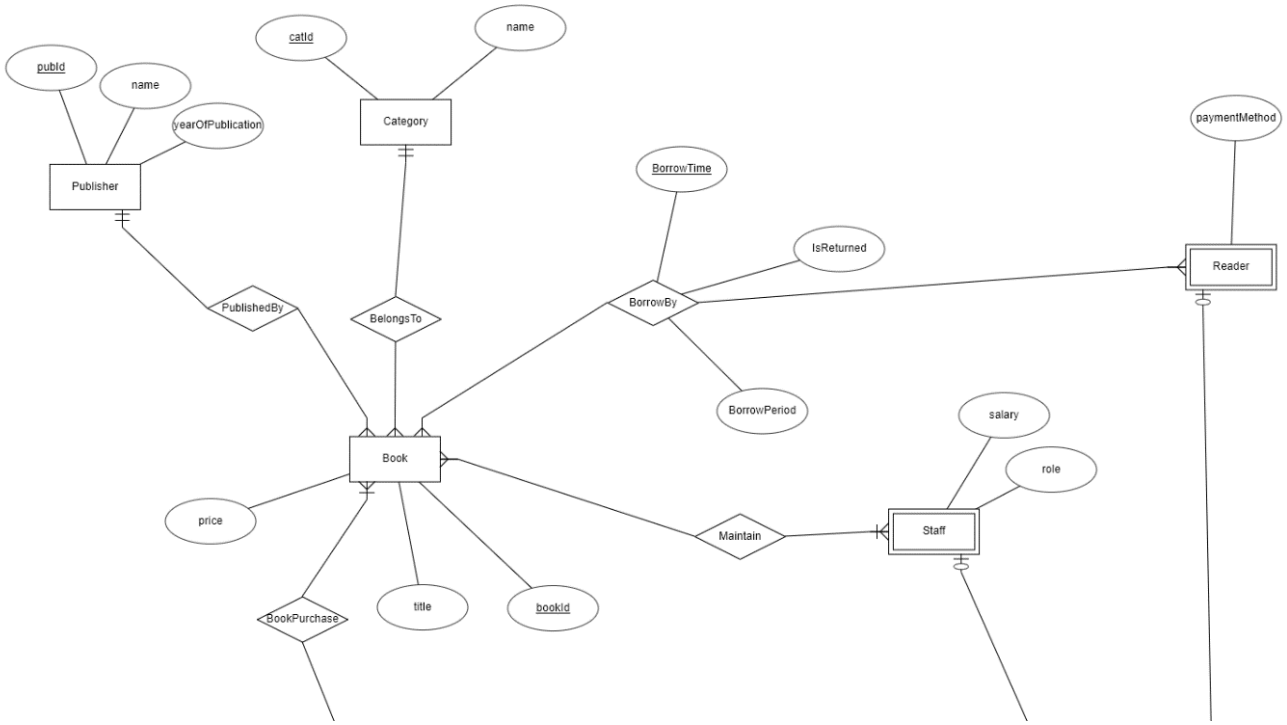
Supplier - הרשת תמשיך לקנות מאותם ספקים כמקודם ולכן אין שינוי בטבלה זו.

Customer - נעשה בדומה לטבלה Staff.

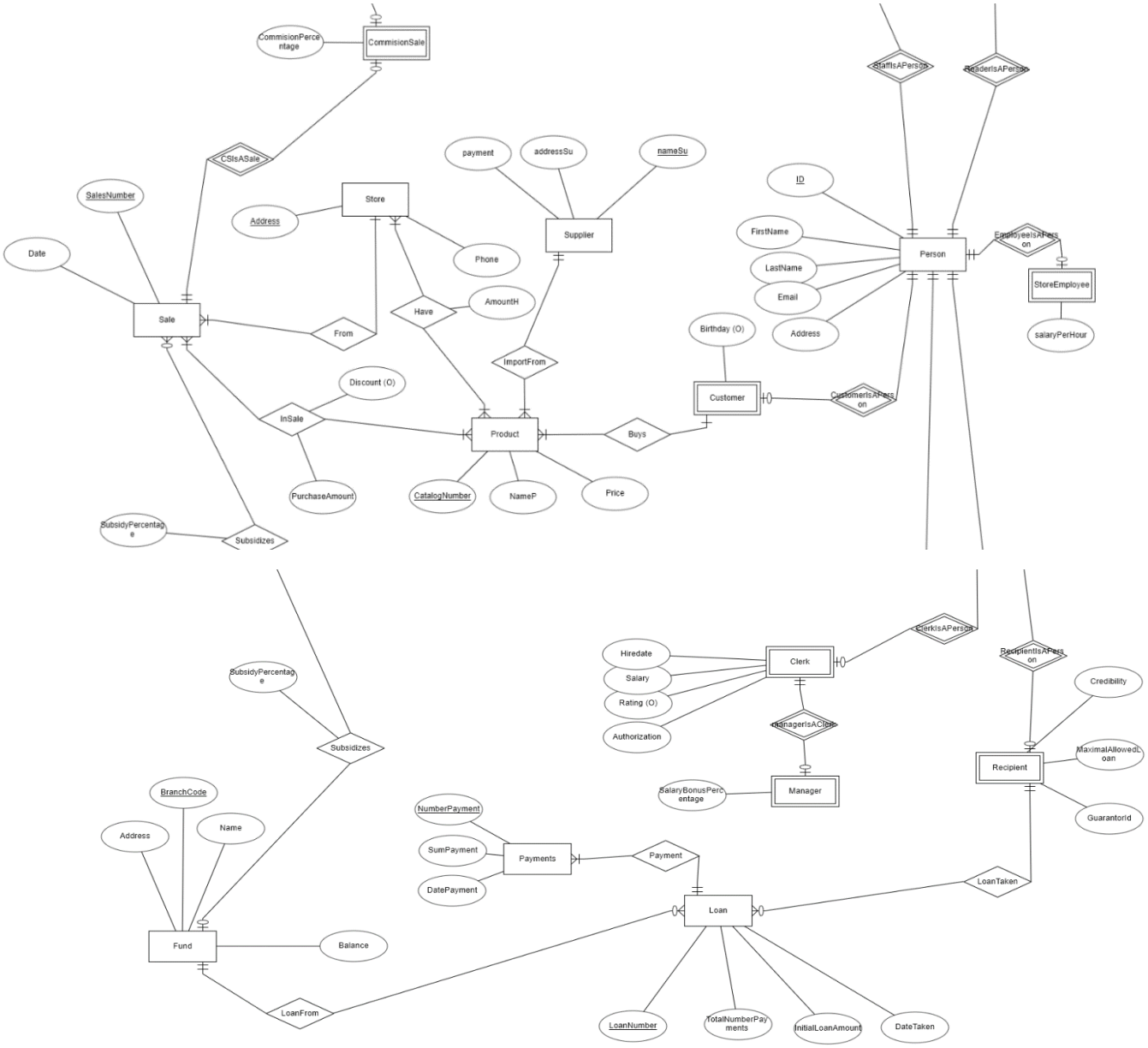
Product - יישאר ללא שינוי מכיוון שמוצרים ימשיכו להימכר כמקודם.

Sale - יקבל אטריבוט חדש BranchCode שתהיה מפתח זר לאטריבוט בעל שם זהה שנמצא ב – Fund וזאת כדי לאפשר מידע על הסניף שמסבסד קניה זאת. כמו כן נוסיף אטריבוט חדש בשם SubsidyPercentage שתייצג את מספר אחוזי הסבסוד מהגמ"ח.

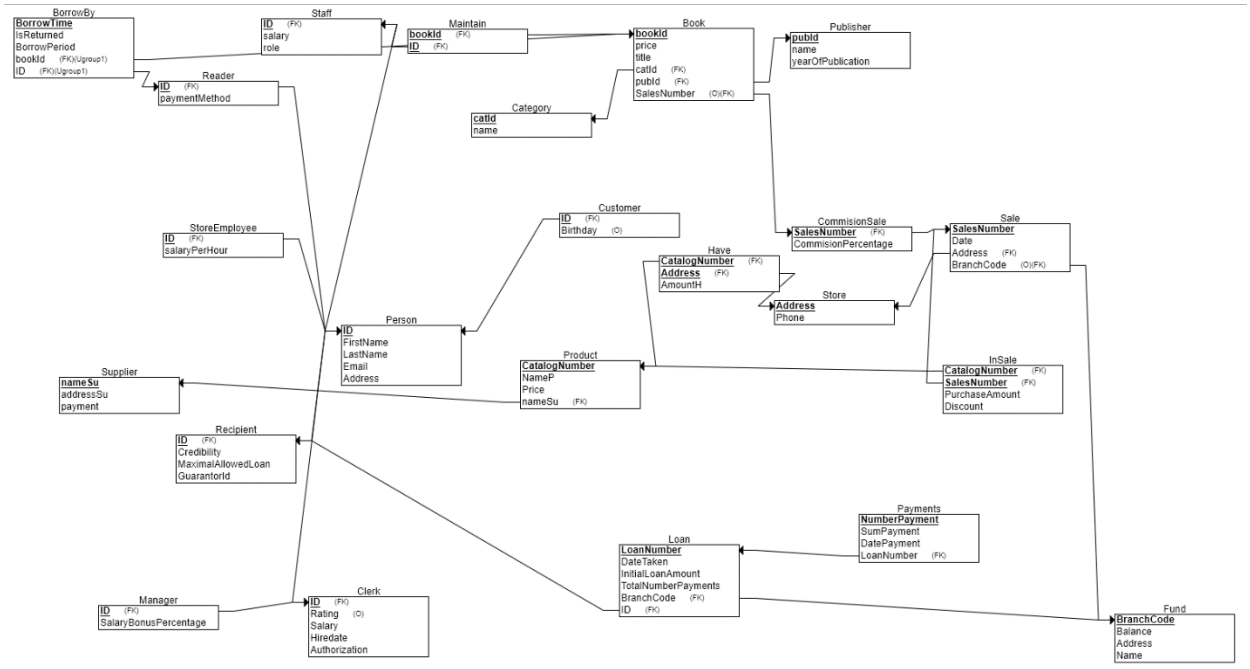
2.4 דיאגרמת ERD



בס"ד מיני פרויקט בסיסי נתונים גמ"ח לב, בנק הלוואות ללא ריבית



2.5 תרשים DSD



כעת ניצור את כל הטבלאות החדשות ונכניס את כלל הסכמות למאגר הנתונים:

את ישויות הגמ"ח הכנסנו כבר למערכת בשלב א, כעת נכניס את שאר הישויות של תתי הקבוצות האחרות והישויות החדשות.

2.6 יצירת הטבלאות ע"י SQL

```
CREATE TABLE Customer
(
  Birthday DATE,
  ID INT NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ID) REFERENCES Person(ID)
);

CREATE TABLE Publisher
(
  name VARCHAR(50) NOT NULL,
```

```
  yearOfPublication INT NOT NULL,
  pubId INT NOT NULL,
  PRIMARY KEY (pubId)
);

CREATE TABLE Category
(
  catId INT NOT NULL,
  name VARCHAR(50) NOT NULL,
  PRIMARY KEY (catId)
);


CREATE TABLE Reader
(
  paymentMethod INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ID) REFERENCES Person(ID)
);


CREATE TABLE Staff
(
  salary INT NOT NULL,
  role VARCHAR(50) NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ID) REFERENCES Person(ID)
);

CREATE TABLE StoreEmployee
(
  salaryPerHour INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ID) REFERENCES Person(ID)
);


CREATE TABLE Store
(
  Address VARCHAR(50) NOT NULL,
  Phone INT NOT NULL,
  PRIMARY KEY (Address)
);


CREATE TABLE Sale
(
  SalesNumber INT NOT NULL,
  DateSale DATE NOT NULL,
  Address VARCHAR(50) NOT NULL,
  BranchCode INT,
```

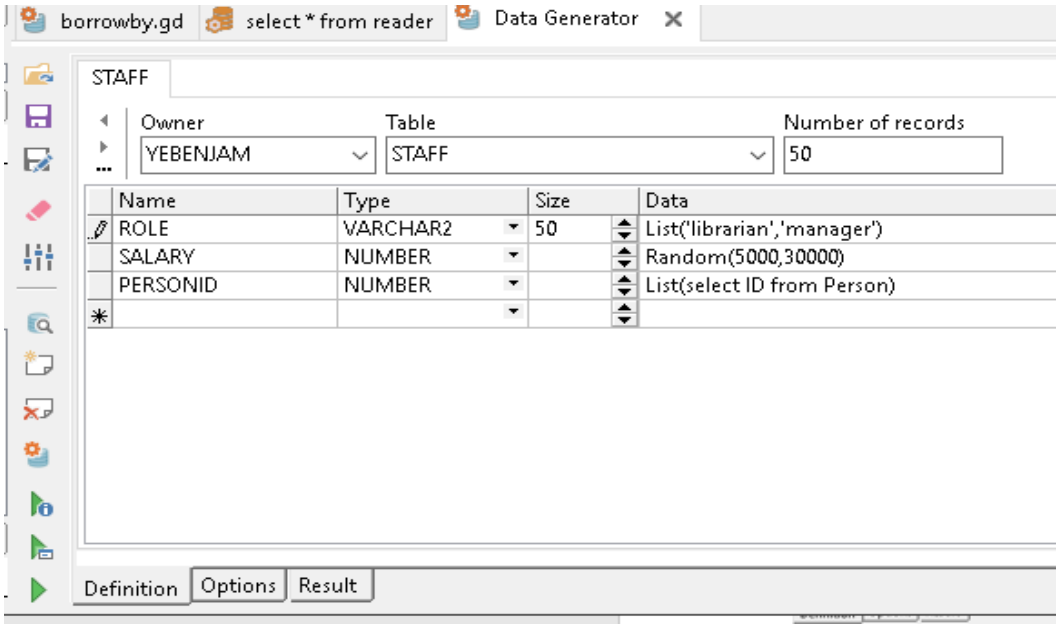
```
  Idc INT NOT NULL,
  PRIMARY KEY (SalesNumber),
  FOREIGN KEY (Address) REFERENCES Store(Address),
  FOREIGN KEY (BranchCode) REFERENCES Fund(BranchCode),
  FOREIGN KEY Idc references Customer(Id)
);


CREATE TABLE Supplier
(
  nameSu VARCHAR(50) NOT NULL,
  addressSu VARCHAR(50) NOT NULL,
  payment INT NOT NULL,
  PRIMARY KEY (nameSu)
);


CREATE TABLE Product
(
  CatalogNumber INT NOT NULL,
  NameP VARCHAR(50) NOT NULL,
  Price INT NOT NULL,
  PRIMARY KEY (CatalogNumber)
);


CREATE TABLE CommisionSale
(
  CommisionPercentage INT NOT NULL,
  SalesNumber INT NOT NULL,
  PRIMARY KEY (SalesNumber),
  FOREIGN KEY (SalesNumber) REFERENCES Sale(SalesNumber)
);


CREATE TABLE Have
(
  AmountH INT NOT NULL,
  CatalogNumber INT NOT NULL,
  Address VARCHAR(50) NOT NULL,
  PRIMARY KEY (CatalogNumber, Address),
  FOREIGN KEY (CatalogNumber) REFERENCES Product(CatalogNumber),
  FOREIGN KEY (Address) REFERENCES Store(Address)
);


CREATE TABLE InSale
(
  PurchaseAmount INT NOT NULL,
  Discount INT NOT NULL,
  CatalogNumber INT NOT NULL,
  SalesNumber INT NOT NULL,
  PRIMARY KEY (CatalogNumber, SalesNumber),
```

```
  FOREIGN KEY (CatalogNumber) REFERENCES Product(CatalogNumber),
  FOREIGN KEY (SalesNumber) REFERENCES Sale(SalesNumber)
);


CREATE TABLE Book
(
  price INT NOT NULL,
  bookId INT NOT NULL,
  title VARCHAR(50) NOT NULL,
  catId INT NOT NULL,
  pubId INT NOT NULL,
  SalesNumber INT,
  PRIMARY KEY (bookId),
  FOREIGN KEY (catId) REFERENCES Category(catId),
  FOREIGN KEY (pubId) REFERENCES Publisher(pubId),
  FOREIGN KEY (SalesNumber) REFERENCES CommisionSale(SalesNumber)
);


CREATE TABLE Maintain
(
  bookId INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (bookId, ID),
  FOREIGN KEY (bookId) REFERENCES Book(bookId),
  FOREIGN KEY (ID) REFERENCES Staff(ID)
);


CREATE TABLE BorrowBy
(
  IsReturned INT NOT NULL,
  BorrowTime DATE NOT NULL,
  BorrowPeriod INT NOT NULL,
  bookId INT NOT NULL,
  ID INT NOT NULL,
  PRIMARY KEY (BorrowTime),
  FOREIGN KEY (bookId) REFERENCES Book(bookId),
  FOREIGN KEY (ID) REFERENCES Reader(ID),
  UNIQUE (bookId, ID)

)
```

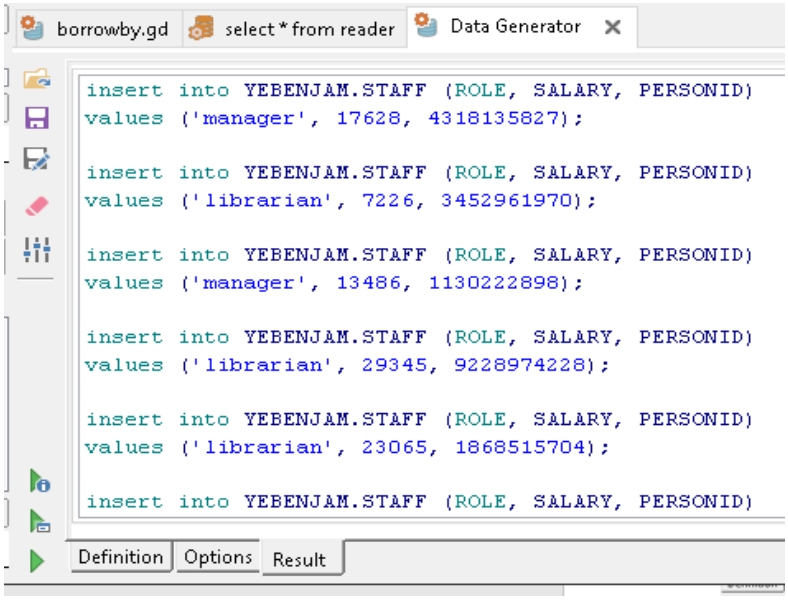
2.7 הכנסת נתונים לטבלאות ע"י שימוש ב Data Generator

2.7.1 הכנסת הטבלה Staff

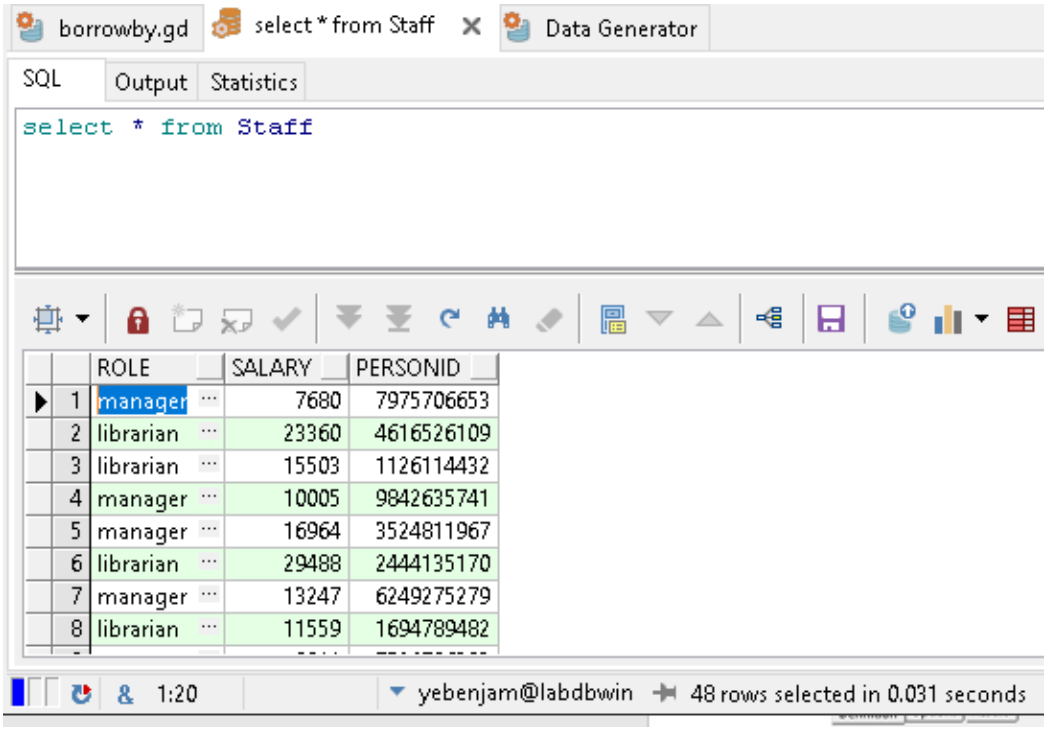
הגדרות הטבלה:



פקודות ההכנסה:

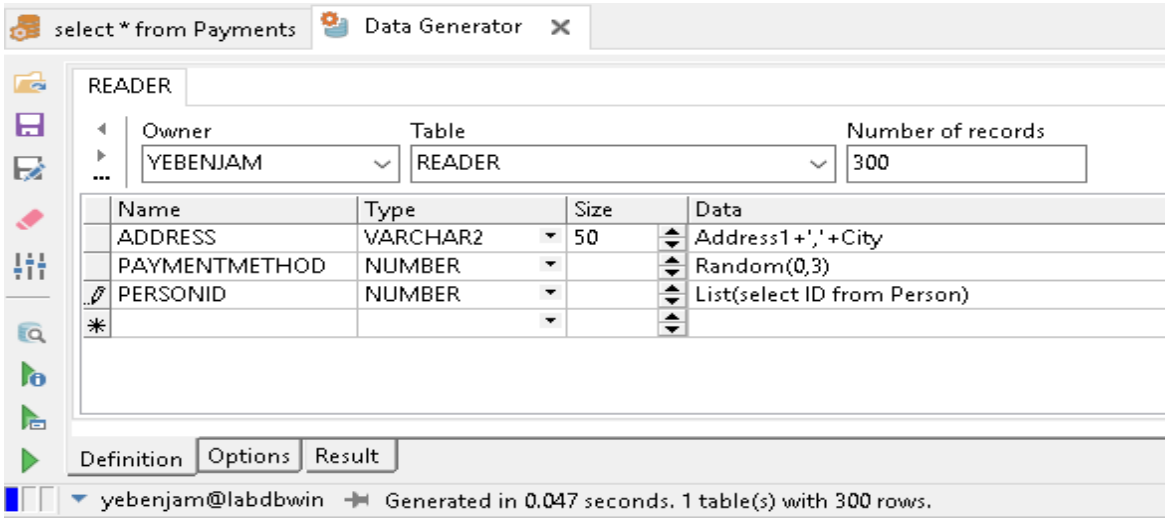


נוודא שההכנסה עבדה:



2.7.2 הכנסת הטבלה Reader

הגדרות הסכמה:



קוד הכנסת הנתונים:

select * from Payments | Data Generator

```
insert into YEBENJAM.READER (ADDRESS, PAYMENTMETHOD, PERSONID)
values ('2 Heubach Street,Kochi', 2, 5543782380);

insert into YEBENJAM.READER (ADDRESS, PAYMENTMETHOD, PERSONID)
values ('60 Ledger Drive,Helsingborg', 2, 6597173277);

insert into YEBENJAM.READER (ADDRESS, PAYMENTMETHOD, PERSONID)
values ('41 Paraju Street,Ettlingen', 2, 2512924213);

insert into YEBENJAM.READER (ADDRESS, PAYMENTMETHOD, PERSONID)
values ('66 Lynn Street,Ben-Gurion', 2, 6729710138);

insert into YEBENJAM.READER (ADDRESS, PAYMENTMETHOD, PERSONID)
values ('83rd Street,Los Alamos', 0, 8426568457);

insert into YEBENJAM.READER (ADDRESS, PAYMENTMETHOD, PERSONID)
values ('727 Uetikon am See Street,Utrecht', 1, 1151234343);
```

Definition | Options | Result

yebenjam@labdbwin 300 records generated in 0.062 seconds

נוודא שההכנסה הצליחה.

select * from Reader | Data Generator

SQL | Output | Statistics

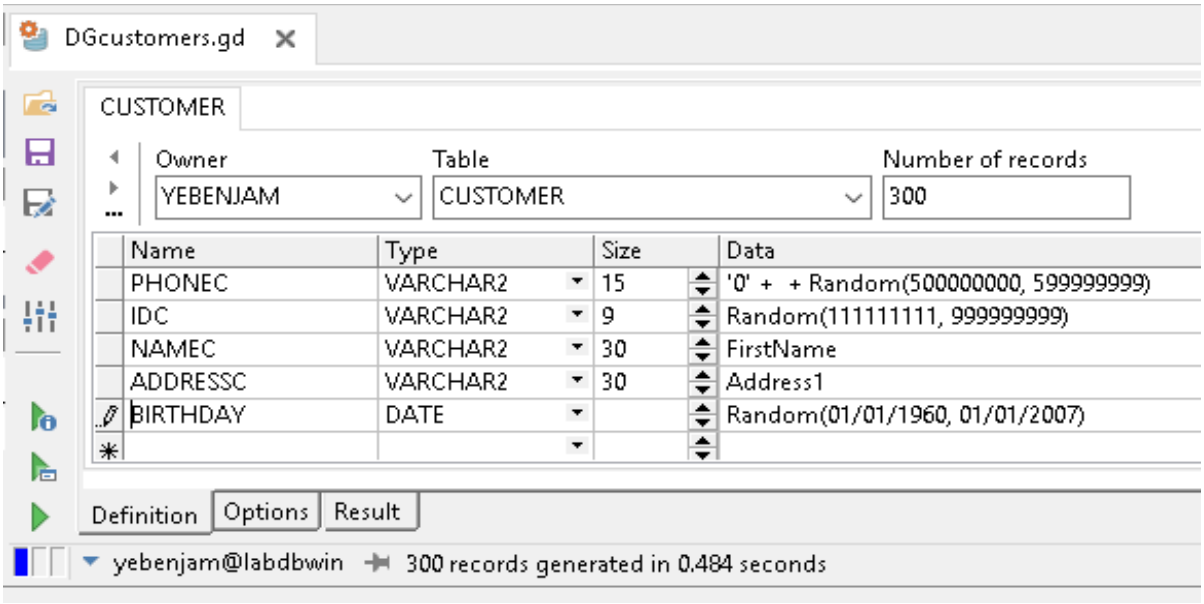
```
select * from Reader
```

	ADDRESS	PAYMENTMETHOD	PERSONID
1	539 Calgary Street,Dresden	3	7402687953
2	46 Craig Road,Rochester	2	2709268152
3	73 Washington Road,Kerava	1	4464361336
4	94 Smurfit Street,Horsens	3	8448333219
5	162 Yamagata Road,Fountain Hills	1	8378750892
6	61st Street,Budapest	0	3352400815
7	80 Smith Road,Horsham	0	2027493651
8	128 Paula Drive,Cambridge	1	6342426880
9	84 Balaban Drive,Suffern	1	3010170008
10	35 Rolando Street,Kaunas	1	8104977579

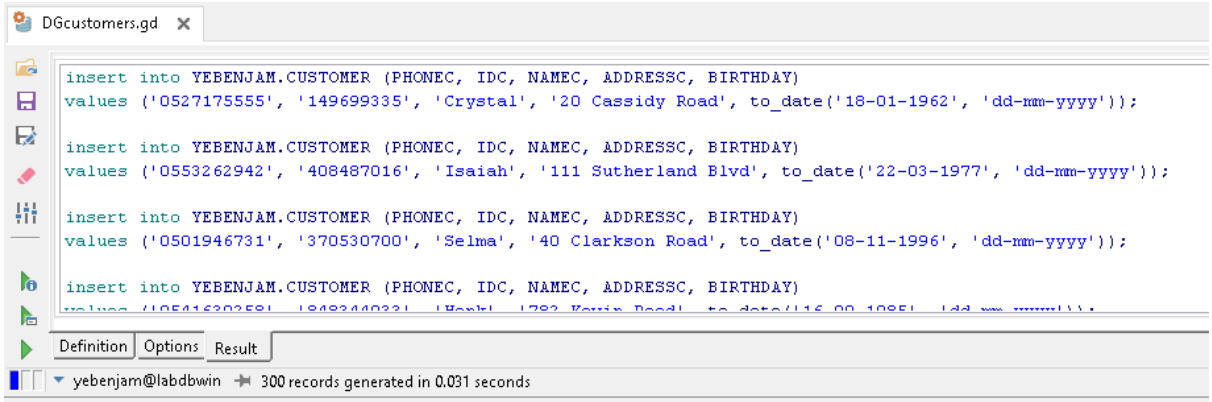
1:21 yebenjam@labdbwin 372 rows selected in 0.151 seconds

2.7.3 הכנסת הטבלה Customer

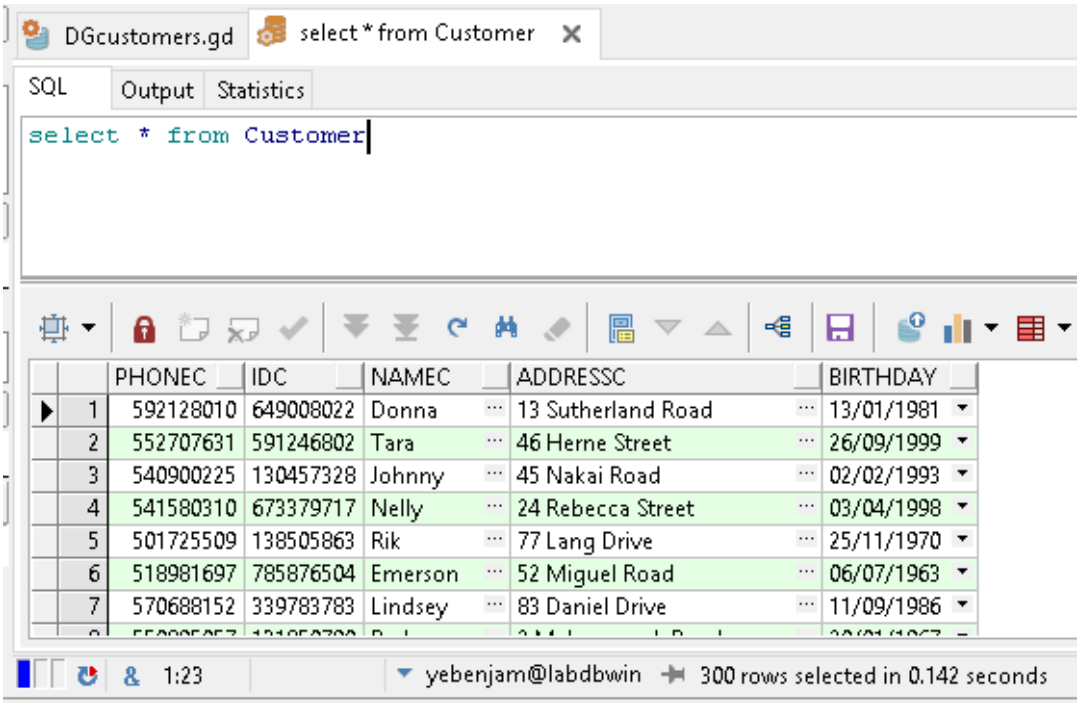
הגדרות הסכמה:



קוד הכנסת הנתונים:

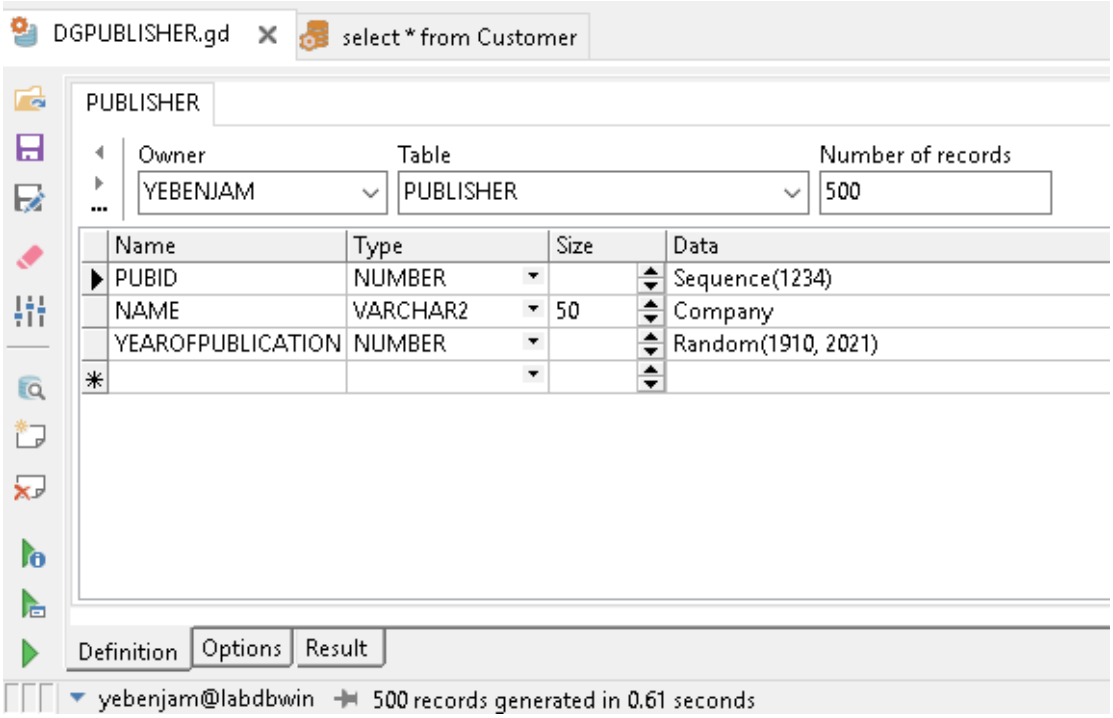


בדיקת תקינות ההכנסה:



2.7.4 הכנסת הטבלה Publisher

הגדרות הסכמה:



קוד הכנסת הנתונים:

DGPUBLISHER.gd | select * from Customer

```
insert into YEBENJAM.PUBLISHER (PUBID, NAME, YEAROFPUBLICATION)
values (1234, 'Infinity Software Development', 1989);

insert into YEBENJAM.PUBLISHER (PUBID, NAME, YEAROFPUBLICATION)
values (1235, 'MHF Logistical Solutions', 1970);

insert into YEBENJAM.PUBLISHER (PUBID, NAME, YEAROFPUBLICATION)
values (1236, 'Berkshire Hathaway', 2013);

insert into YEBENJAM.PUBLISHER (PUBID, NAME, YEAROFPUBLICATION)
values (1237, 'Contract Counsel', 2002);

insert into YEBENJAM.PUBLISHER (PUBID, NAME, YEAROFPUBLICATION)
values (1238, 'Greenwich Technology Partners', 1987);
```

Definition | Options | Result

yebenjam@labdbwin 500 records generated in 0.032 seconds

נוודא שההכנסה עבדה.

DGPUBLISHER.gd | select * from Publisher

SQL | Output | Statistics

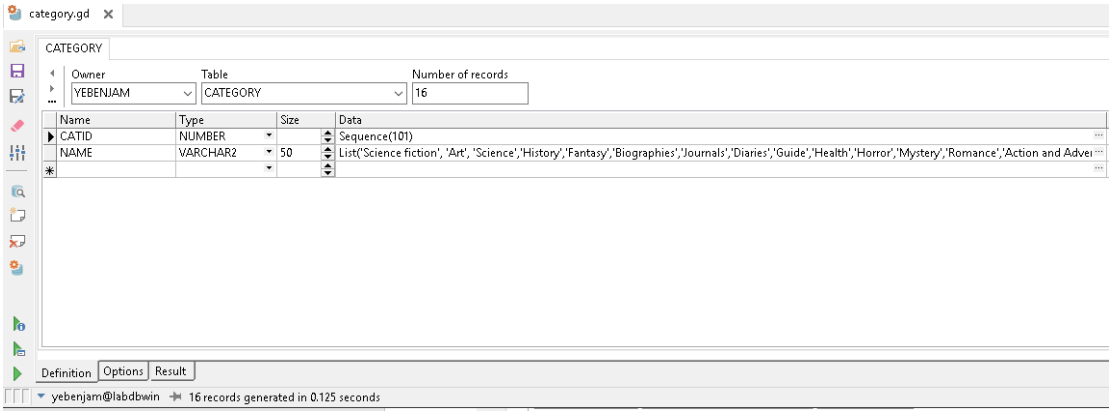
```
select * from Publisher
```

	PUBID	NAME	YEAROFPUBLICATION
1	1234	Priority Leasing	1963
2	1235	Canterbury Park	1978
3	1236	American Pan & Engineering	1943
4	1237	OnStaff	1961
5	1238	Denali Ventures	1952
6	1239	Intrasphere Technologies	1978
7	1240	Banfe Products	2010
8	1241	Harrison & Shriftman	1942

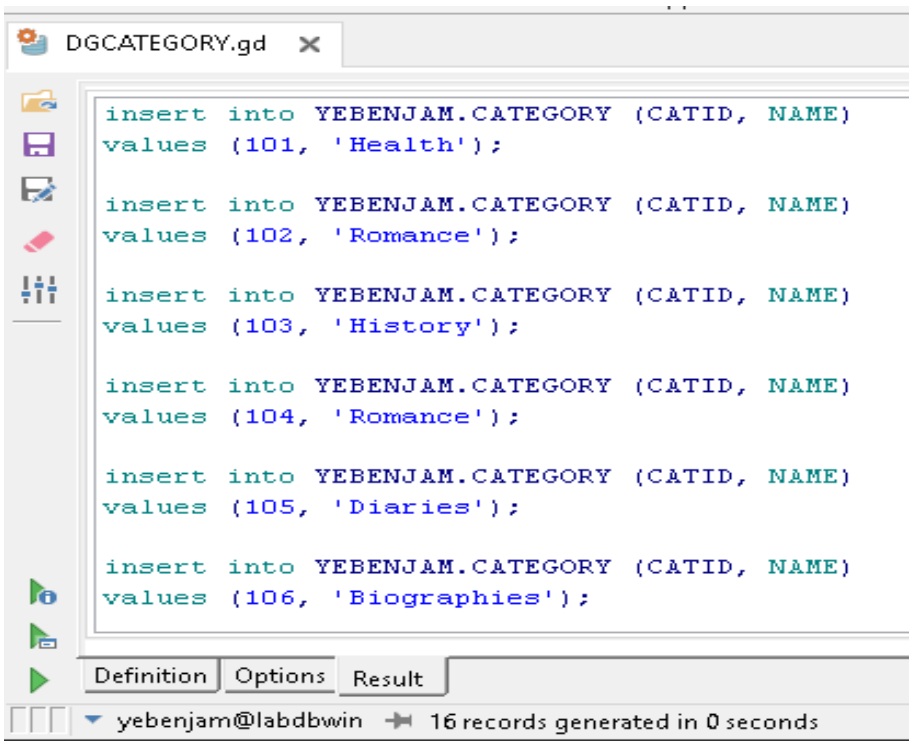
1:24 yebenjam@labdbwin 500 rows selected in 0.189 seconds

2.7.5 הכנסת הטבלה Category

הגדרות הסכמה:



קוד הכנסת הנתונים:



נוודא שההכנסה הצליחה:

select * from Category

SQL | Output | Statistics

```
select * from Category
```

	CATID	NAME
1	101	Action and Adventure
2	102	Drama
3	103	Art
4	104	Science fiction
5	105	Diaries
6	106	Mystery
7	107	Diaries
8	108	Science
9	109	Science fiction
10	110	Art
11	111	Science

1:23 yebenjam@labdbwin 16 rows selected in 0.023 seconds

2.7.6 הכנסת הטבלה Product

הגדרות הסכמה:

Data Generator

PRODUCT

Owner: YEBENJAM | Table: PRODUCT | Number of records: 200

Name	Type	Size	Data
CATALOGNUMBER	NUMBER		Sequence(1)
NAMEP	VARCHAR2	50	Company
PRICE	NUMBER		Random(1000,5000)

Definition | Options | Result

yebenjam@labdbwin 200 records generated in 60.016 seconds

פקודות ההכנסה:

Data Generator | select * from Product

```
insert into YEBENJAM.PRODUCT (CATALOGNUMBER, NAMEP, PRICE)
values (1, 'Cima Labs', 1755);

insert into YEBENJAM.PRODUCT (CATALOGNUMBER, NAMEP, PRICE)
values (2, 'Keller Williams Realty Ahwatukee Foothills', 4322);

insert into YEBENJAM.PRODUCT (CATALOGNUMBER, NAMEP, PRICE)
values (3, 'GHR Systems', 3762);

insert into YEBENJAM.PRODUCT (CATALOGNUMBER, NAMEP, PRICE)
values (4, 'GulfMark Offshore', 1153);

insert into YEBENJAM.PRODUCT (CATALOGNUMBER, NAMEP, PRICE)
values (5, 'Turner Professional Services', 1397);

insert into YEBENJAM.PRODUCT (CATALOGNUMBER, NAMEP, PRICE)
values (6, 'ACS International Resources', 3262);
```

Definition | Options | Result

yebenjam@labdbwin — 200 records generated in 0.016 seconds

נוודא שזה נכנס בהצלחה.

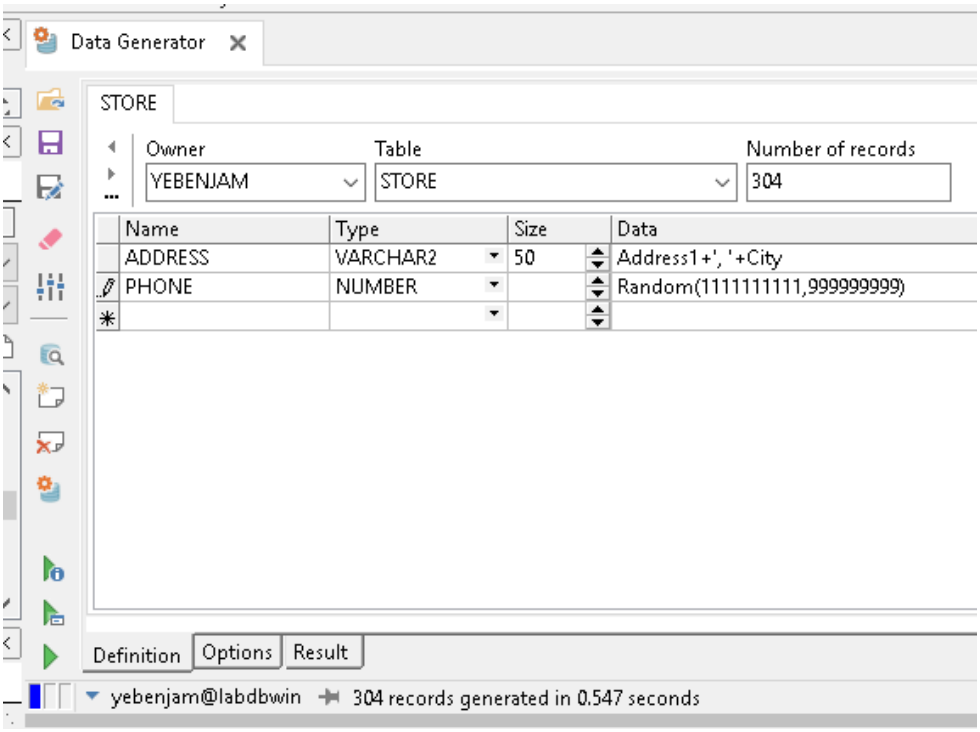
```
select * from Product
```

	CATALOGNUMBER	NAMEP	PRICE
1	1	Oneida Financial	1674
2	2	Balchem	3141
3	3	Consultants' Choice	1898
4	4	S.C. Johnson & Son	3715
5	5	Wells Financial	1004
6	6	Circuit City Stores	2231
7	7	Auction Systems Auctioneers & Appraisers	1784
8	8	Circuit City Stores	3321
9	9	All Star Consulting	4097

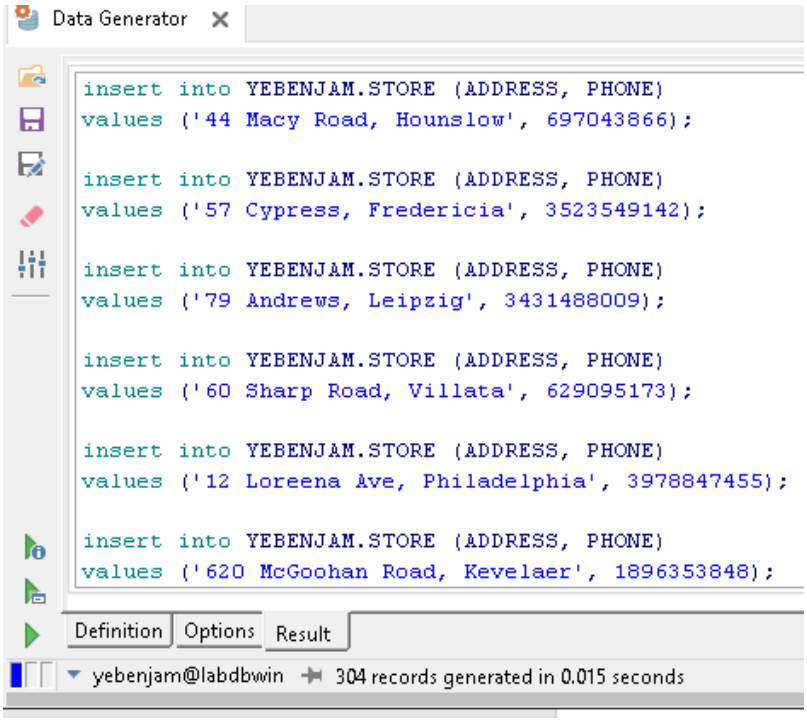
1:22 — yebenjam@labdbwin — 200 rows selected in 0.124 seconds

2.7.7 הכנסת הטבלה Store

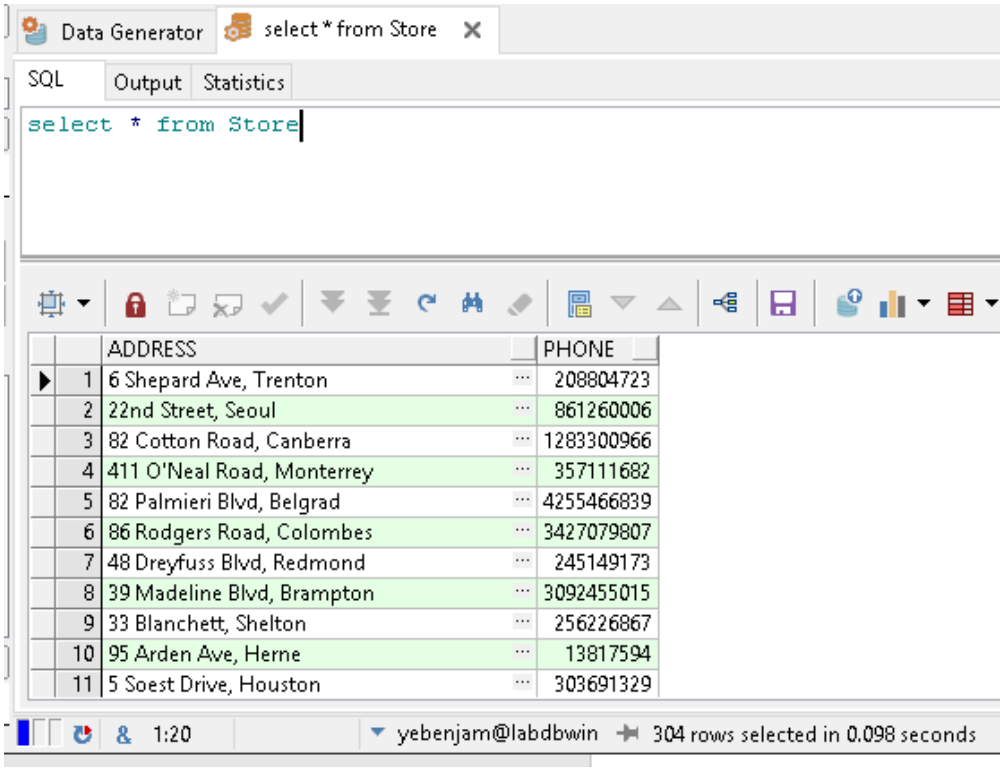
הגדרות הסכמה:



פקודות ההכנסה:



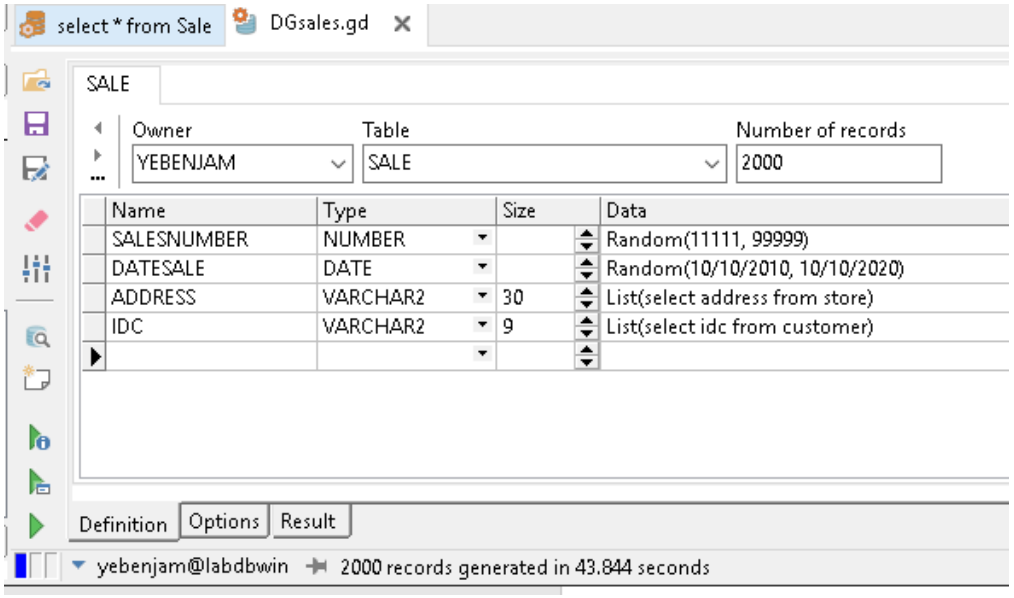
נוודא שזה הוכנס בהצלחה.



	ADDRESS	PHONE
1	6 Shepard Ave, Trenton	208804723
2	22nd Street, Seoul	861260006
3	82 Cotton Road, Canberra	1283300966
4	411 O'Neal Road, Monterrey	357111682
5	82 Palmieri Blvd, Belgrad	4255466839
6	86 Rodgers Road, Colombes	3427079807
7	48 Dreyfuss Blvd, Redmond	245149173
8	39 Madeline Blvd, Brampton	3092455015
9	33 Blanchett, Shelton	256226867
10	95 Arden Ave, Herne	13817594
11	5 Soest Drive, Houston	303691329

2.7.8 הכנסת הטבלה Sale

הגדרות הסכמה:



Name	Type	Size	Data
SALESNUMBER	NUMBER		Random(11111, 99999)
DATESALE	DATE		Random(10/10/2010, 10/10/2020)
ADDRESS	VARCHAR2	30	List(select address from store)
IDC	VARCHAR2	9	List(select idc from customer)

פקודות ההכנסה:

```
insert into YEBENJAM.SALE (SALESNUMBER, DATESALE, ADDRESS, IDC)
values (57424, to_date('24-01-2013', 'dd-mm-yyyy'), '33 Blanchett, Shelton', '899586569');

insert into YEBENJAM.SALE (SALESNUMBER, DATESALE, ADDRESS, IDC)
values (34832, to_date('22-04-2016', 'dd-mm-yyyy'), '142 Clarence Road, Sarasota', '708833643');

insert into YEBENJAM.SALE (SALESNUMBER, DATESALE, ADDRESS, IDC)
values (71358, to_date('13-01-2020', 'dd-mm-yyyy'), '65 Brugherio Ave, Riverdale', '816709553');

insert into YEBENJAM.SALE (SALESNUMBER, DATESALE, ADDRESS, IDC)
values (68667, to_date('26-01-2015', 'dd-mm-yyyy'), '40 Blair Street, Linz', '587397887');

insert into YEBENJAM.SALE (SALESNUMBER, DATESALE, ADDRESS, IDC)
```

Definition | Options | Result

yebenjam@labdbwin — 2000 records generated in 0.547 seconds

נוודא שההכנסה הצליחה.

select * from Sale | DGsales.gd

SQL | Output | Statistics

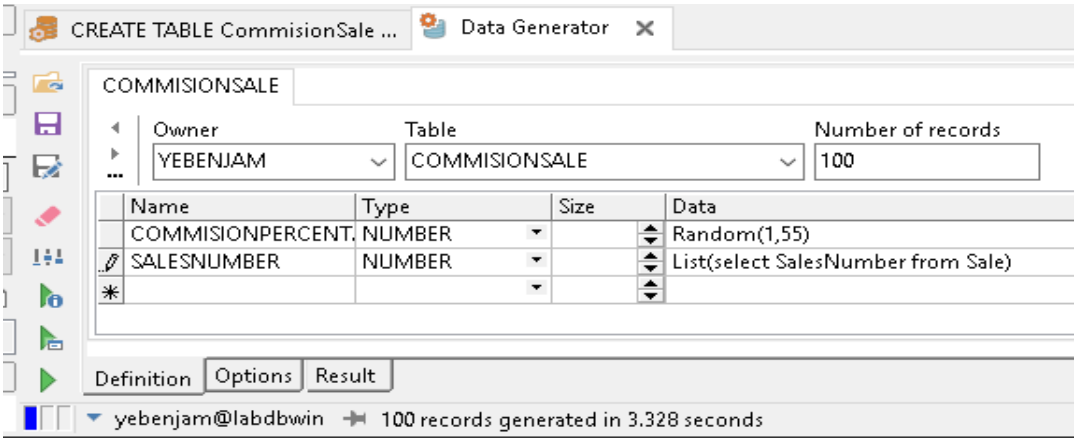
```
select * from Sale
```

	SALESNUMBER	DATESALE	ADDRESS	IDC
1	66636	03/09/2017	67 Richardson Drive, Yucca	792908958
2	48241	16/05/2018	91 Dianne Drive, Bruneck	133658580
3	68335	11/04/2019	33 Callow Street, Nanaimo	713018789
4	55030	21/06/2016	50 Tyler Street, San Antonio	201896355
5	64393	09/01/2014	93 de Lancie Drive, Nordhausen	824185175
6	14003	31/05/2020	94 Dern Road, St. Louis	298477806
7	63209	21/03/2017	053 Garry Ave, Warszawa	272442668

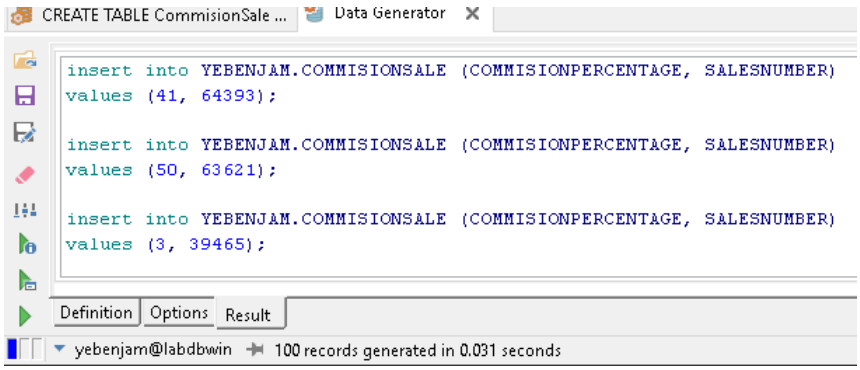
1:19 — yebenjam@labdbwin — 1813 rows selected in 0.545 seconds

2.7.9 הכנסת הטבלה CommisionSale

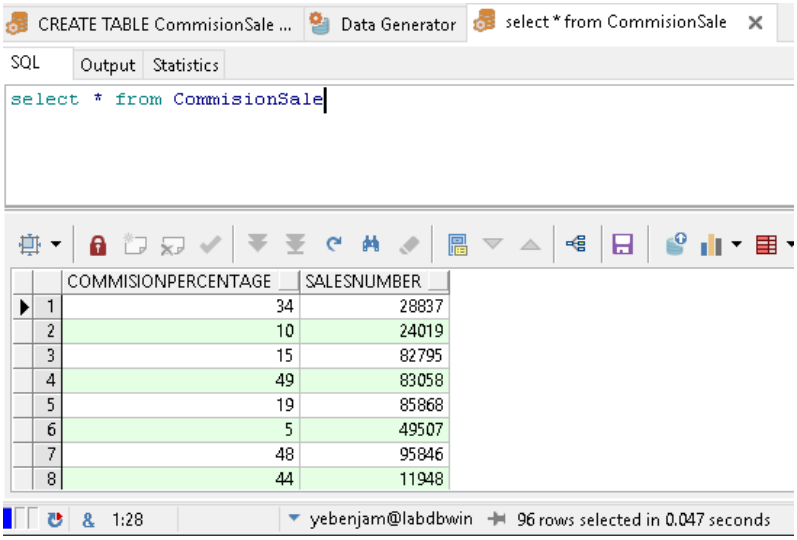
הגדרות הסכמה:



פקודות ההכנסה:

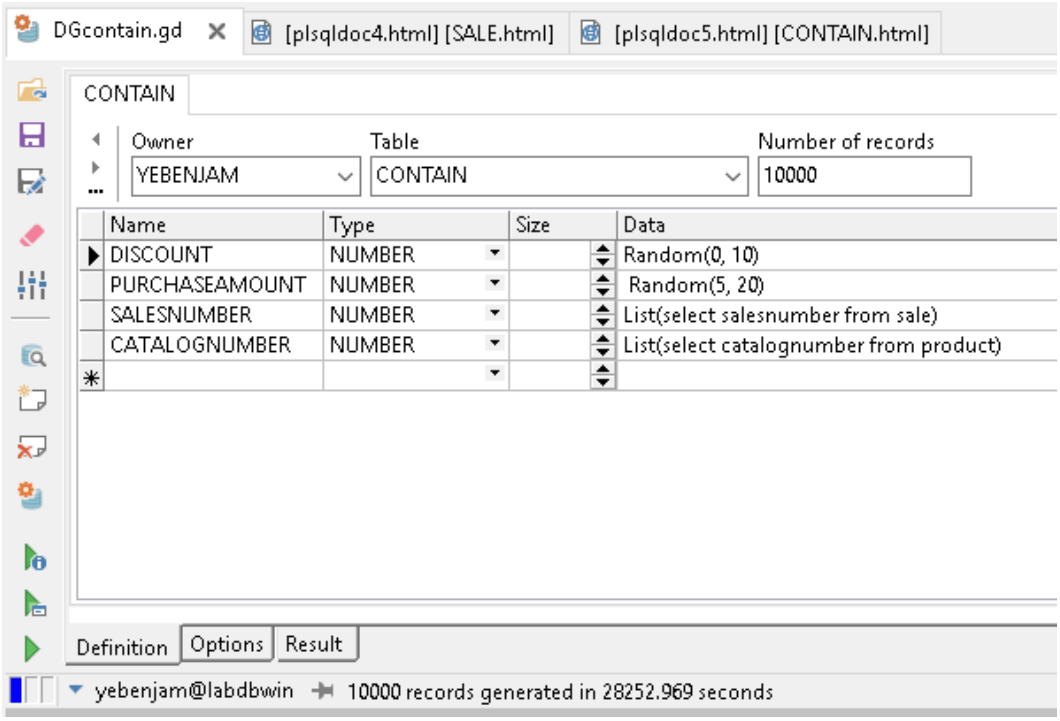


נוודא שההכנסה הצליחה.

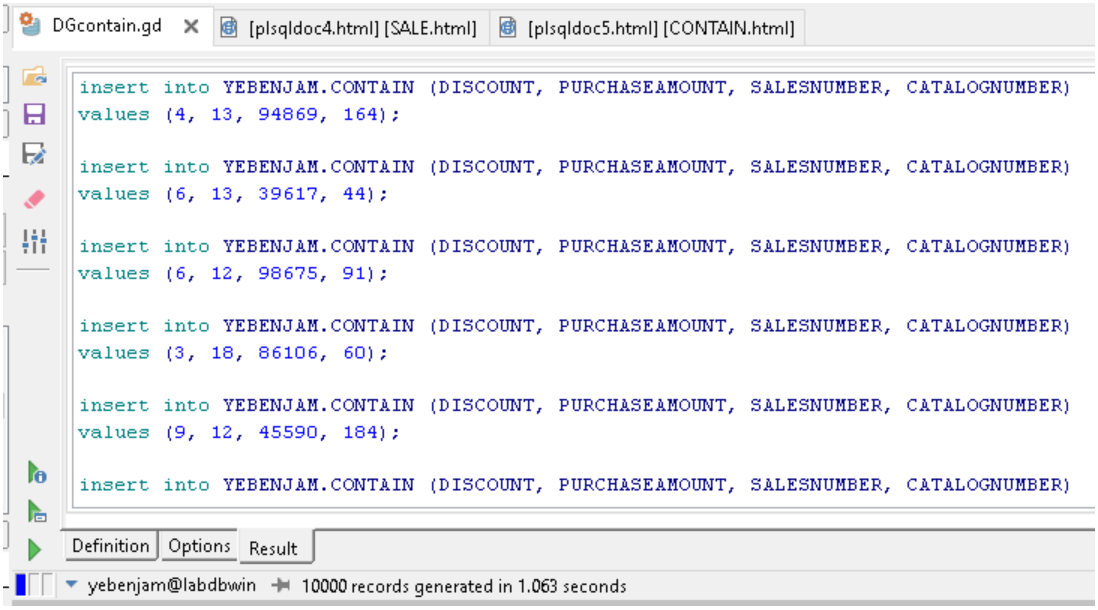


2.7.10 הכנסת הטבלה Contain

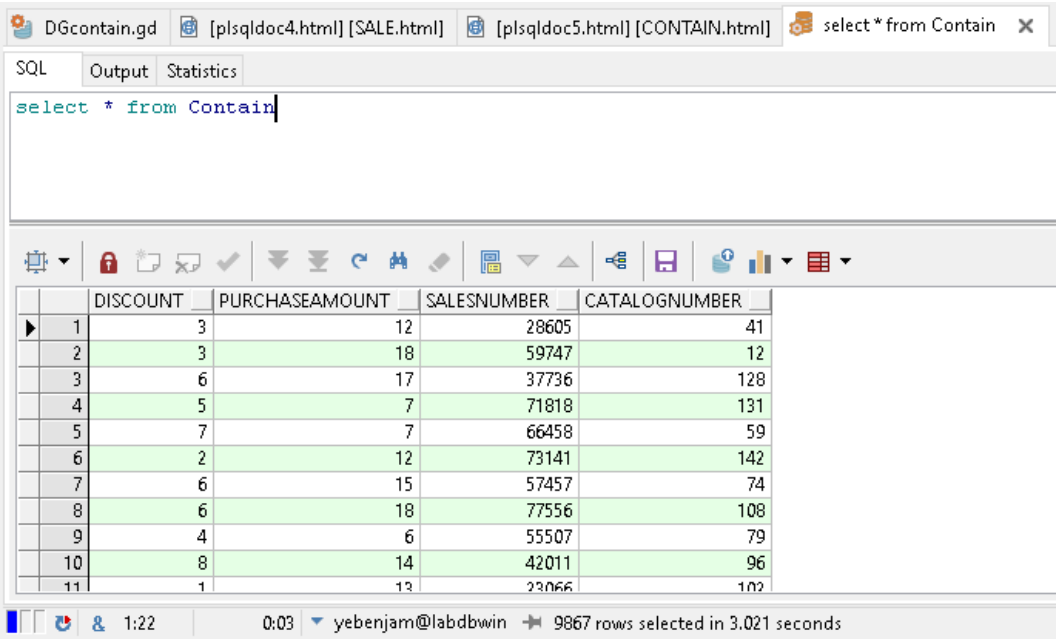
הגדרות הסכמה:



פקודות ההכנסה:



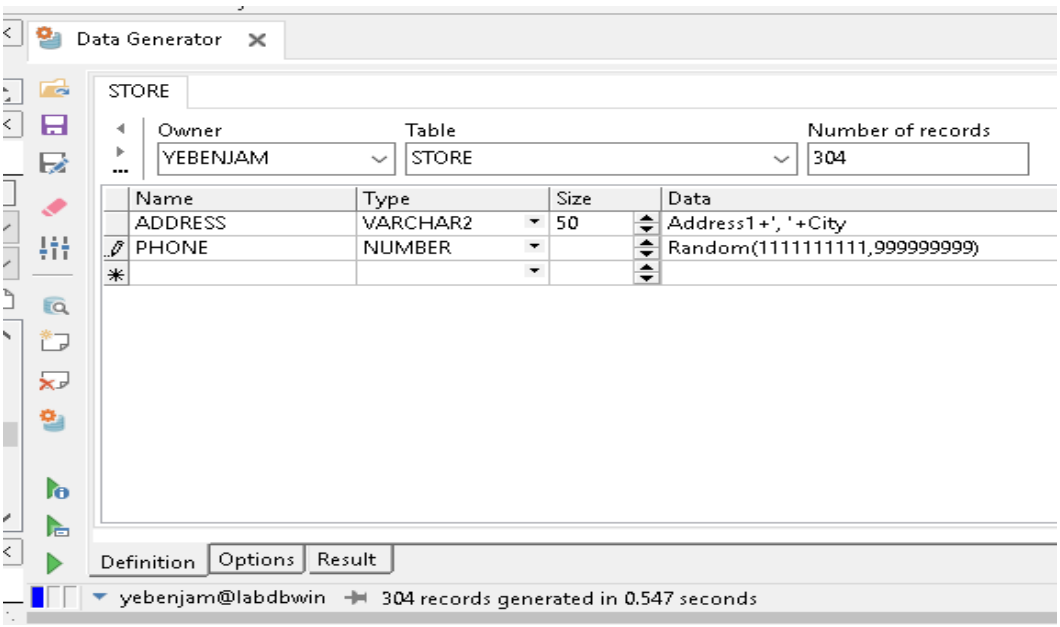
נוודא שההכנסה הצליחה.



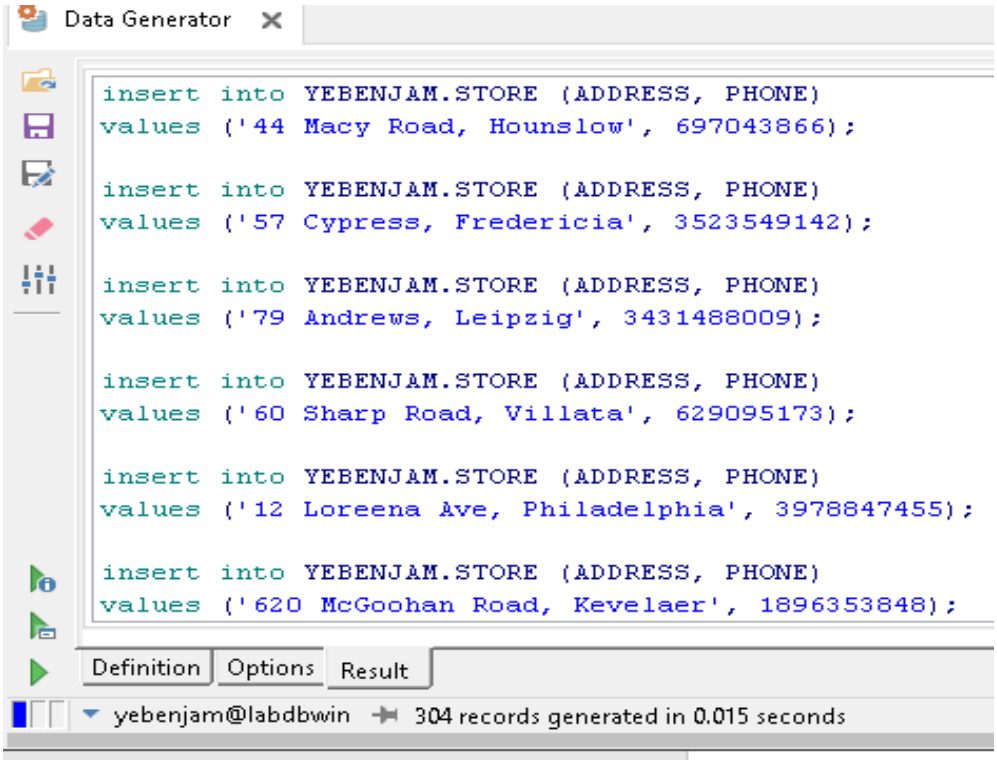
	DISCOUNT	PURCHASEAMOUNT	SALESNUMBER	CATALOGNUMBER
1	3	12	28605	41
2	3	18	59747	12
3	6	17	37736	128
4	5	7	71818	131
5	7	7	66458	59
6	2	12	73141	142
7	6	15	57457	74
8	6	18	77556	108
9	4	6	55507	79
10	8	14	42011	96
11	1	13	23066	102

2.7.11 הכנסת הטבלה Store

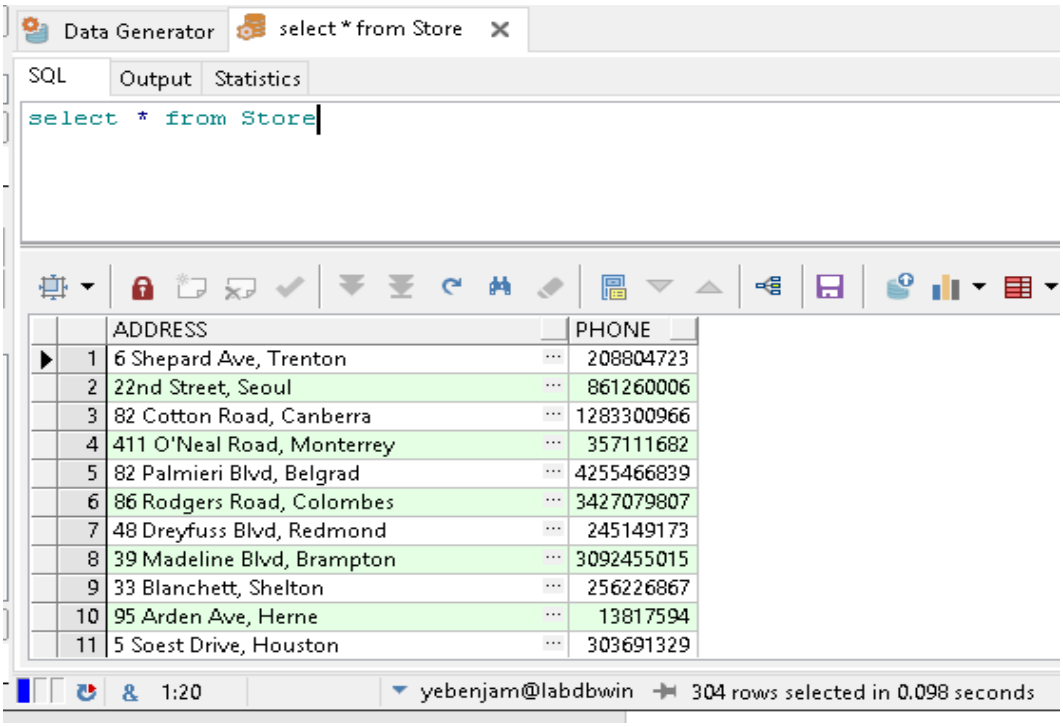
הגדרות הסכמה:



פקודות ההכנסה:

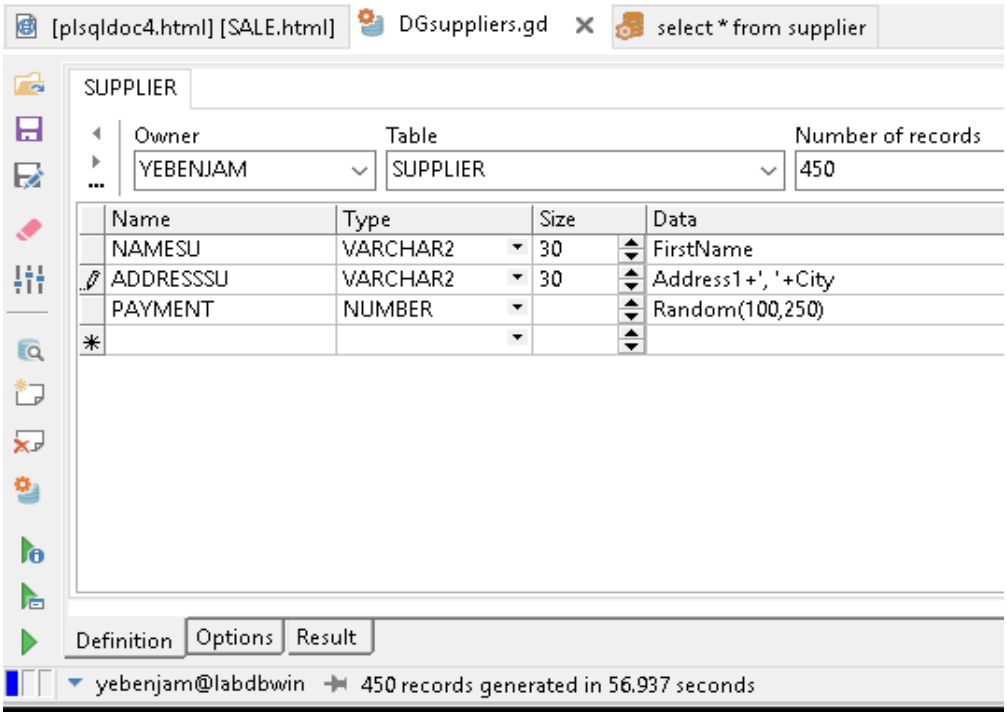


נוודא שזה הוכנס בהצלחה.

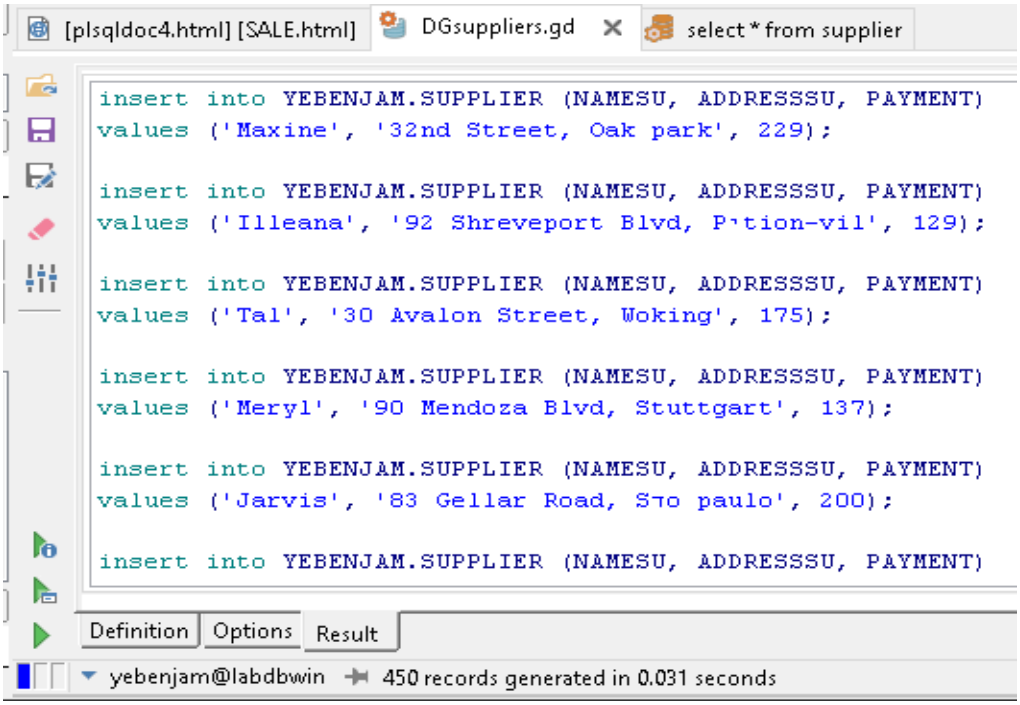


2.7.12 הכנסת הטבלה Supplier

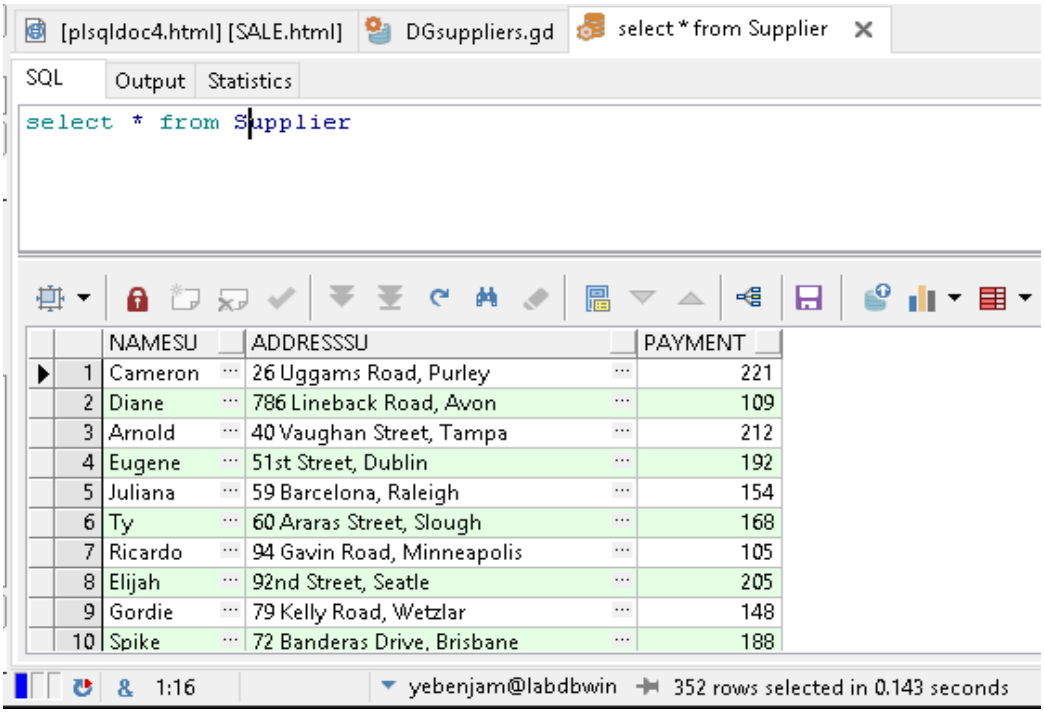
הגדרות הסכמה:



פקודות ההכנסה:

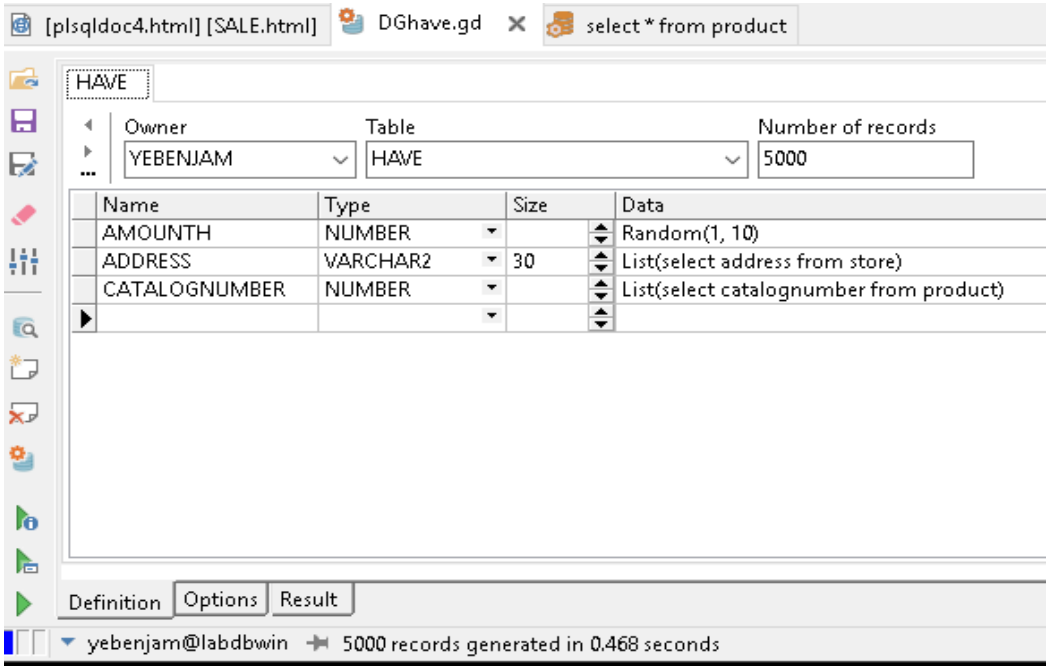


נוודא שההכנסה עבדה.



2.7.13 הכנסת הטבלה Have

הגדרות הסכמה:



פקודות ההכנסה:

[plsqldoc4.html] [SALE.html] DGhave.gd select * from product

```
insert into YEBENJAM.HAVE (AMOUNTH, ADDRESS, CATALOGNUMBER)
values (5, '88 Green Road, Anyang-si', 3);

insert into YEBENJAM.HAVE (AMOUNTH, ADDRESS, CATALOGNUMBER)
values (9, '69 Ray Road, Chapeco', 181);

insert into YEBENJAM.HAVE (AMOUNTH, ADDRESS, CATALOGNUMBER)
values (6, '46 Bad Oeynhausen Drive, Fount', 133);

insert into YEBENJAM.HAVE (AMOUNTH, ADDRESS, CATALOGNUMBER)
values (7, '91st Street, Zipf', 171);

insert into YEBENJAM.HAVE (AMOUNTH, ADDRESS, CATALOGNUMBER)
values (4, '11 Dillane Drive, Hochwald', 133);

insert into YEBENJAM.HAVE (AMOUNTH, ADDRESS, CATALOGNUMBER)
```

Definition Options Result

yebenjam@labdbwin 5000 records generated in 0.36 seconds

נוודא שההכנסה עבדה.

[plsqldoc4.html] [SALE.html] DGhave.gd select * from Have

SQL Output Statistics

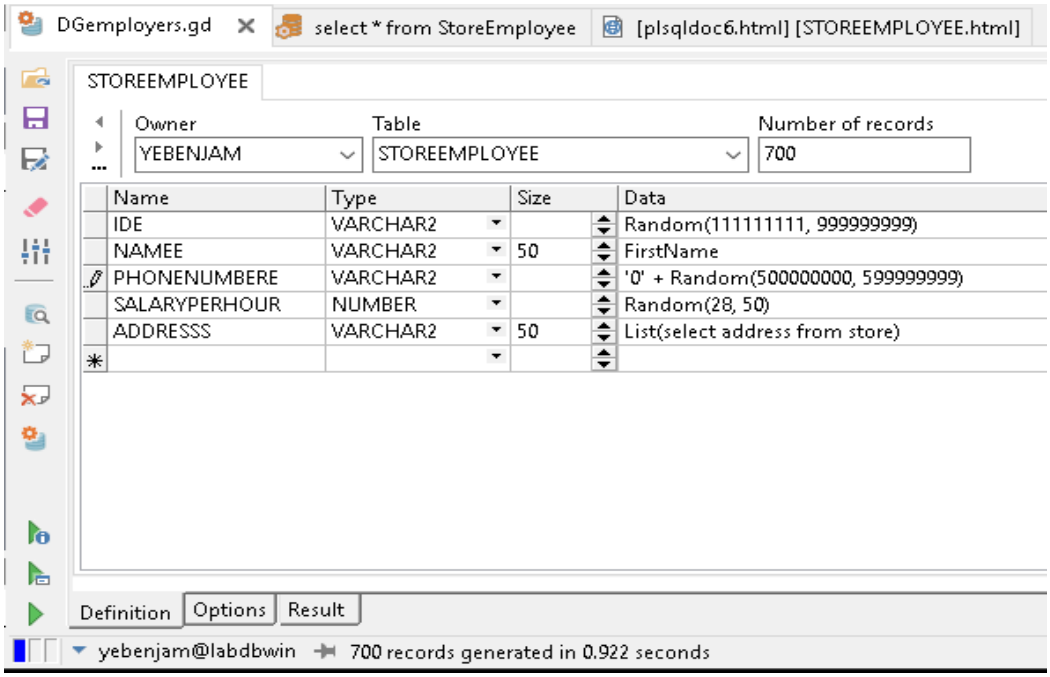
```
select * from Have
```

	AMOUNTH	ADDRESS	CATALOGNUMBER
1	6	86 Witherspoon Street, Zaandam	145
2	3	62 Quentin Street, Spring City	77
3	2	57 Pearce Blvd, Paris	114
4	1	57 Pearce Blvd, Paris	180
5	2	44 Rodgers Blvd, Wichita	72
6	1	62 Maxine Drive, Ljubljana	66
7	4	33rd Street, Fairborn	44
8	3	13 Steagall Road, Fornacette	93
9	1	50 Dunn Road, Monterey	56
10	7	26 Lynskey Road, Glendale	8
11	5	88 Hampton Drive, Kevelaer	80

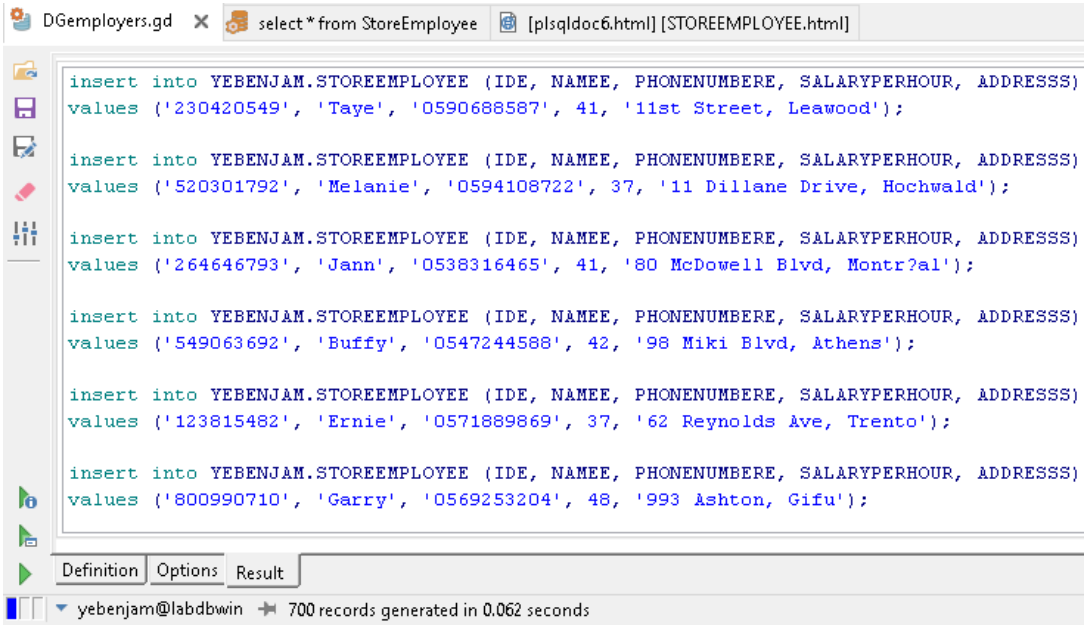
1:19 0:01 yebenjam@labdbwin 3545 rows selected in 1.136 seconds

2.7.14 הכנסת הטבלה StoreEmployee

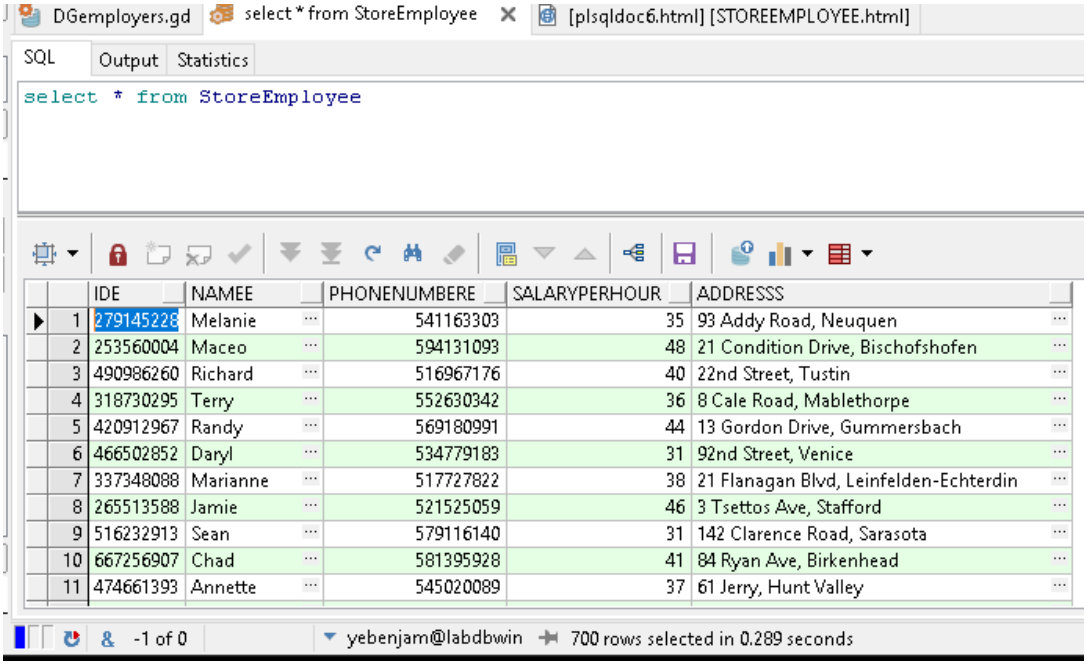
הגדרות הסכמה:



פקודות ההכנסה:



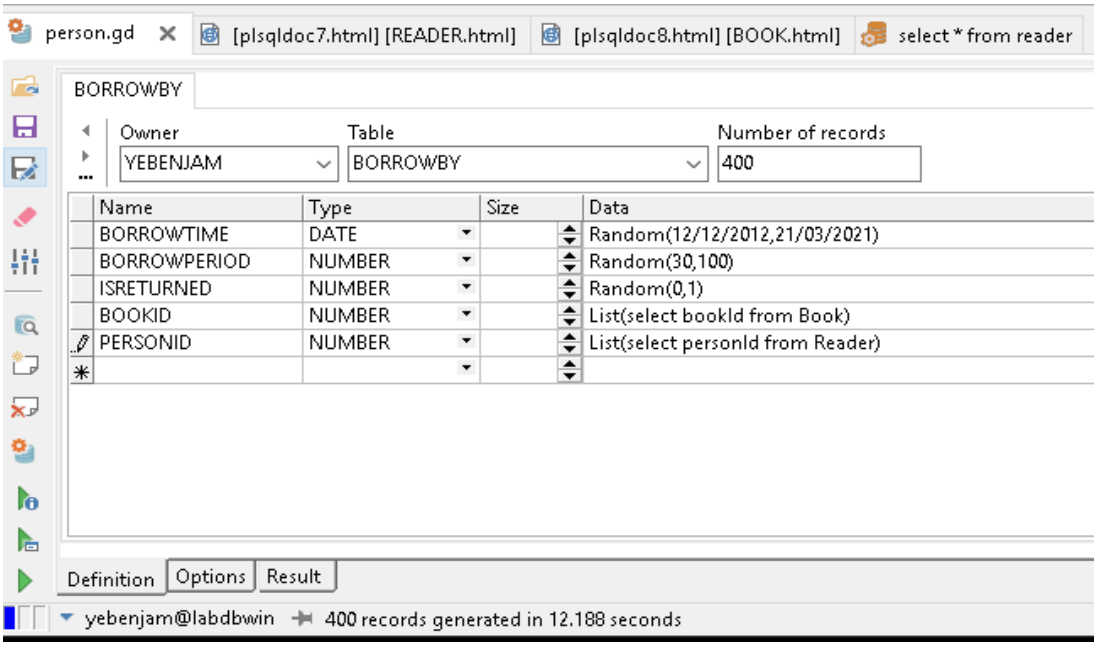
נוודא שההכנסה עבדה.



	IDE	NAMEE	PHONENUMBERE	SALARYPERHOUR	ADDRESSS
1	279145228	Melanie	541163303	35	93 Addy Road, Neuquen
2	253560004	Maceo	594131093	48	21 Condition Drive, Bischofshofen
3	490986260	Richard	516967176	40	22nd Street, Tustin
4	318730295	Terry	552630342	36	8 Cale Road, Mablethorpe
5	420912967	Randy	569180991	44	13 Gordon Drive, Gummersbach
6	466502852	Daryl	534779183	31	92nd Street, Venice
7	337348088	Marianne	517727822	38	21 Flanagan Blvd, Leinfelden-Echterdin
8	265513588	Jamie	521525059	46	3 Tsettos Ave, Stafford
9	516232913	Sean	579116140	31	142 Clarence Road, Sarasota
10	667256907	Chad	581395928	41	84 Ryan Ave, Birkenhead
11	474661393	Annette	545020089	37	61 Jerry, Hunt Valley

2.7.15 הכנסת הטבלה BorrowBy

הגדרות הסכמה:



פקודות ההכנסה:

borrowby.gd | select * from Staff | borrowby.gd

```
insert into YEBENJAM.BORROWBY (BORROWTIME, BORROWPERIOD, ISRETURNED, BOOKID, PERSONID)
values (to_date('07-12-2179 02:37:23', 'dd-mm-yyyy hh24:mi:ss'), 94, 0, 101861, 6491568051);

insert into YEBENJAM.BORROWBY (BORROWTIME, BORROWPERIOD, ISRETURNED, BOOKID, PERSONID)
values (to_date('05-06-2881 10:17:02', 'dd-mm-yyyy hh24:mi:ss'), 66, 1, 720682, 3017858626);

insert into YEBENJAM.BORROWBY (BORROWTIME, BORROWPERIOD, ISRETURNED, BOOKID, PERSONID)
values (to_date('13-02-2904 10:26:32', 'dd-mm-yyyy hh24:mi:ss'), 43, 0, 142829, 4888616887);

insert into YEBENJAM.BORROWBY (BORROWTIME, BORROWPERIOD, ISRETURNED, BOOKID, PERSONID)
values (to_date('01-07-2621 15:38:09', 'dd-mm-yyyy hh24:mi:ss'), 83, 1, 885289, 3278228251);

insert into YEBENJAM.BORROWBY (BORROWTIME, BORROWPERIOD, ISRETURNED, BOOKID, PERSONID)
values (to_date('01-11-2554 18:09:56', 'dd-mm-yyyy hh24:mi:ss'), 77, 0, 366579, 8722299533);
```

Definition | Options | Result

yebenjam@labdbwin — 400 records generated in 0.781 seconds

נוודא שההכנסה עבדה.

borrowby.gd | select * from BorrowBy

SQL | Output | Statistics

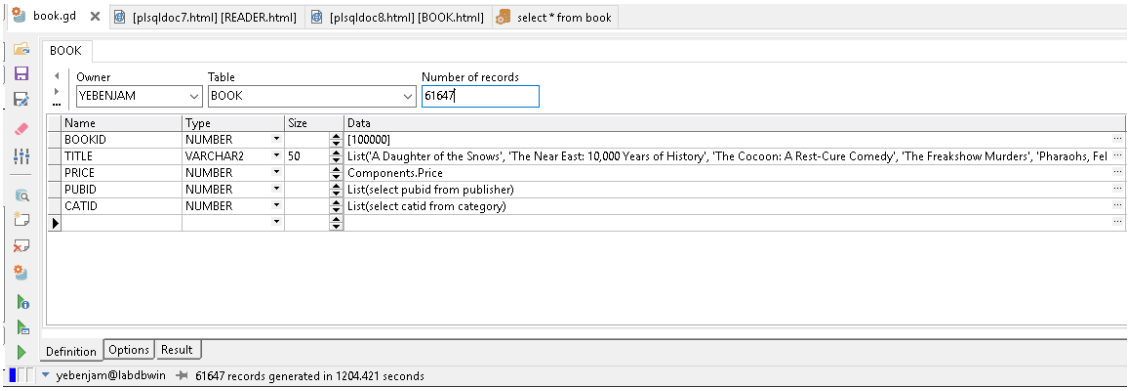
```
select * from BorrowBy
```

	BORROWTIME	BORROWPERIOD	ISRETURNED	BOOKID	PERSONID
1	27/07/2018	47	1	719181	8582455652
2	22/06/2016	70	0	458936	1781958702
3	02/10/2013	94	0	422782	7402687953
4	05/04/2020	92	0	148308	3174305802
5	21/06/2020	97	1	635607	2829340397
6	23/06/2016	47	1	241940	5011265197
7	12/02/2019	86	1	679202	8805872272
8	11/07/2018	32	1	791387	3958574620

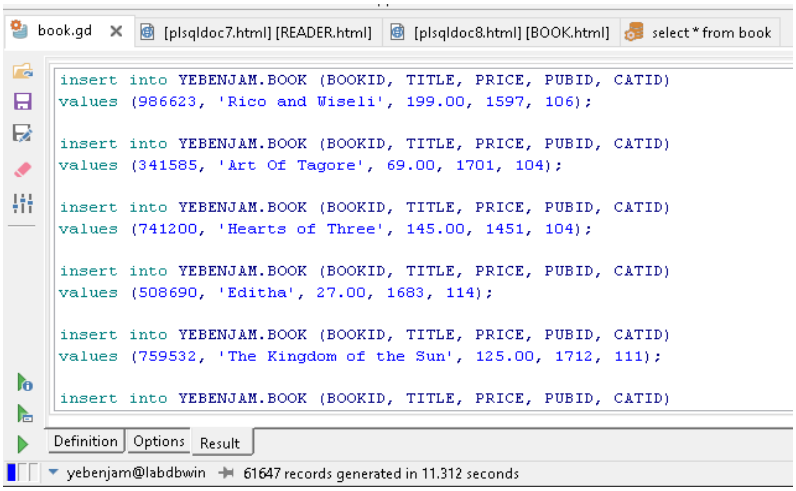
1:23 — yebenjam@labdbwin — 773 rows selected in 0.270 seconds

2.7.16 הכנסת הטבלה Book

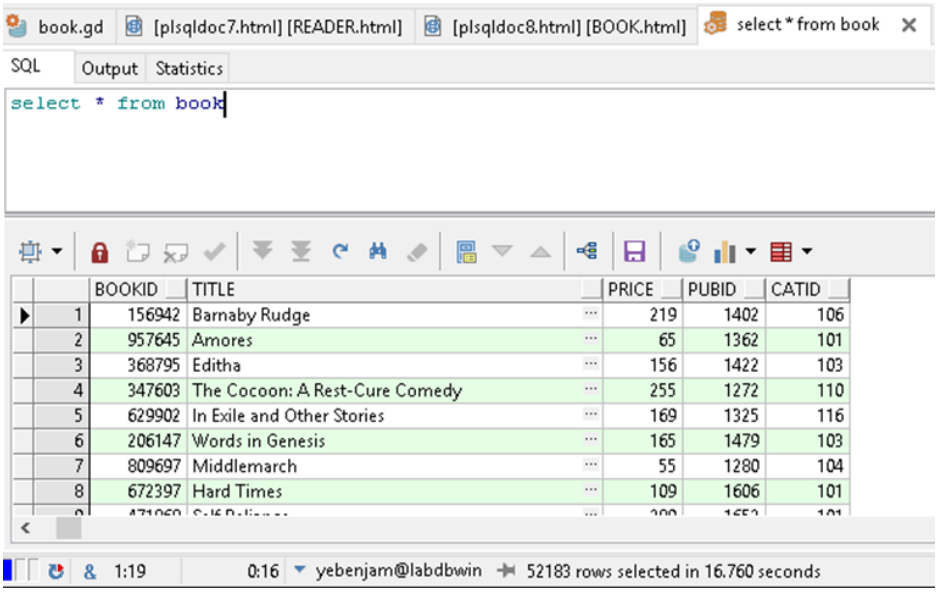
הגדרות הסכמה:



פקודות ההכנסה:

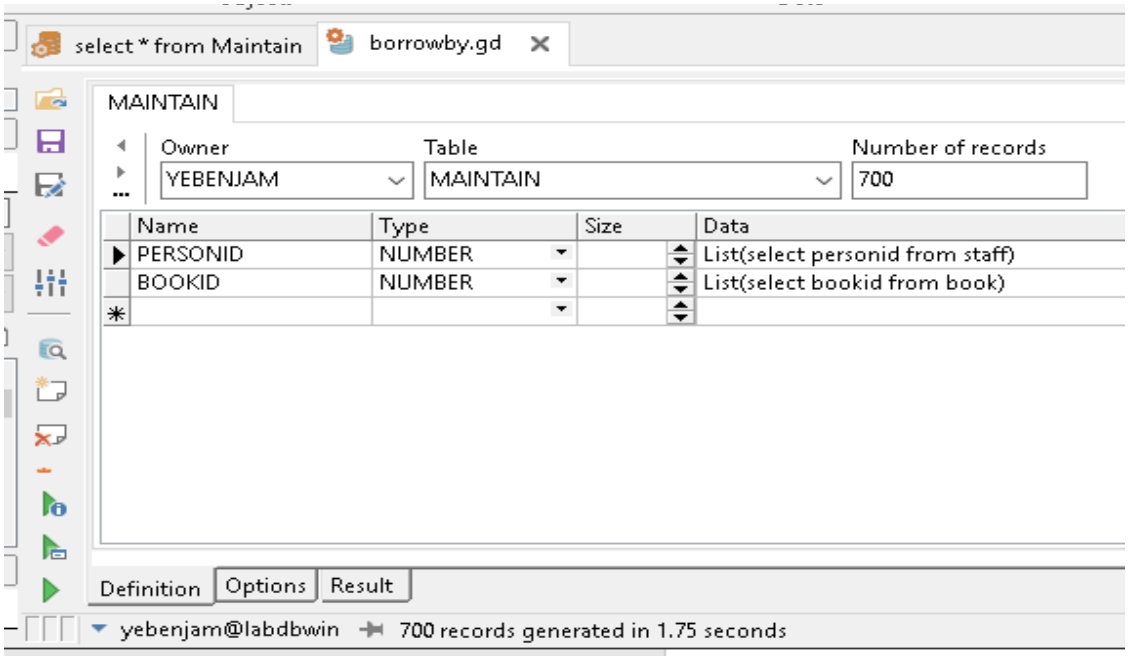


נוודא שההכנסה עבדה:

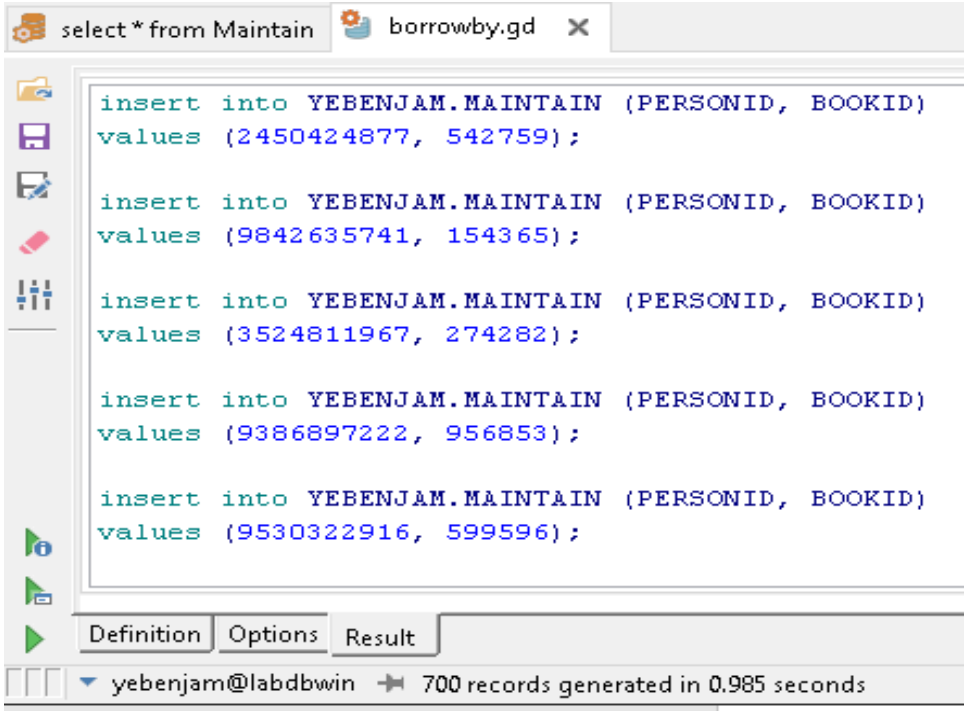


2.7.17 הכנסת הטבלה Maintain

הגדרות הסכמה:



פקודות ההכנסה:



נוודא שההכנסה עבדה.

SQL Output Statistics

```
select * from Maintain
```

	PERSONID	BOOKID
1	1126114432	103369
2	1126114432	129417
3	1126114432	135106
4	1126114432	207968
5	1126114432	260669
6	1126114432	293443
7	1126114432	305122
8	1126114432	309906
9	1126114432	455456

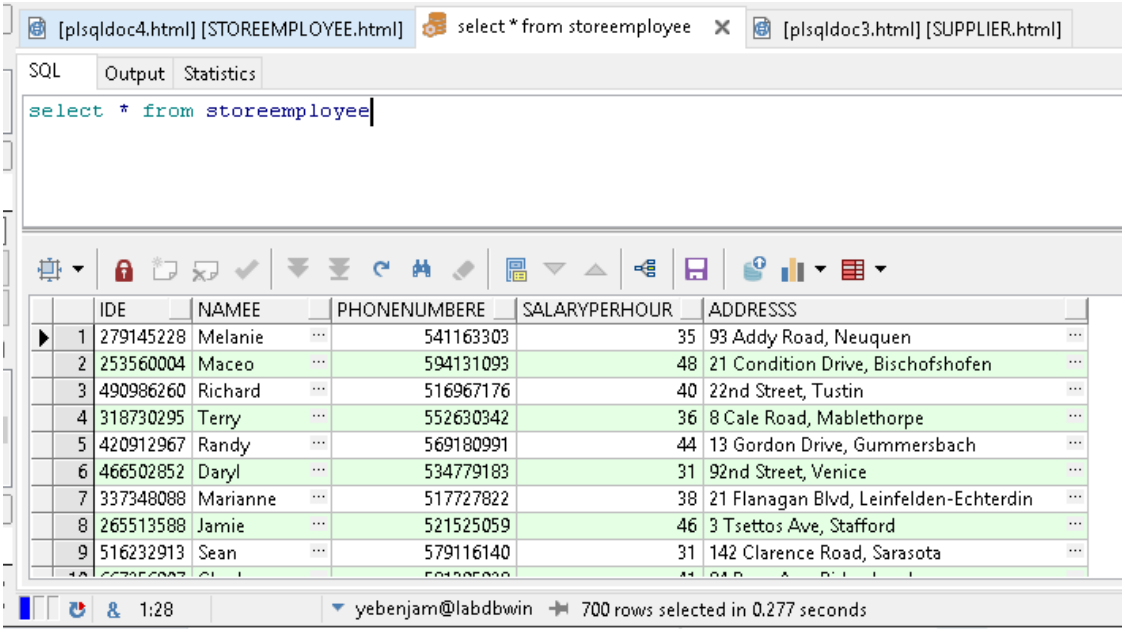
1:23 yebenjam@labdbwin 700 rows selected in 0.208 seconds

2.8 פקודות Alter Table

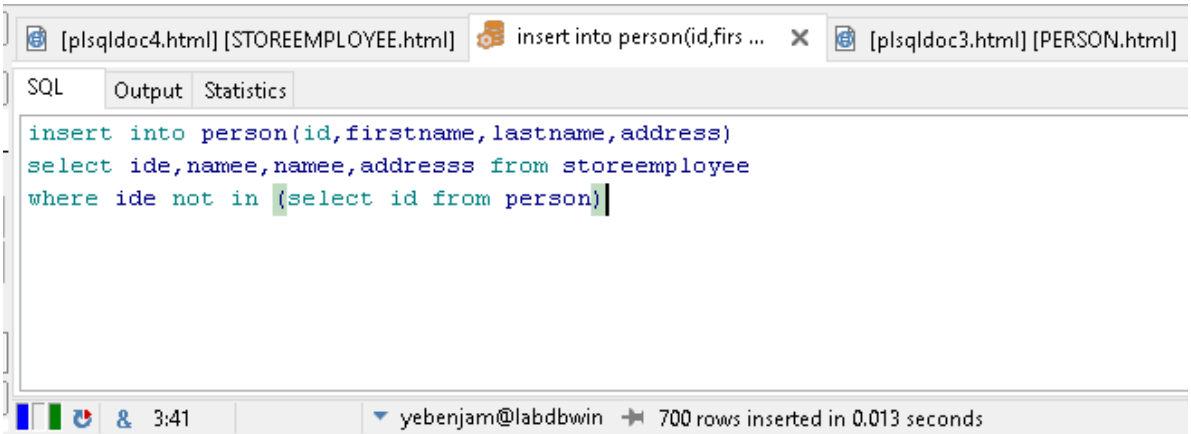
2.8.1 שינוי סכמת StoreEmployee

לפני איחוד הסכמות, בסכמה StoreEmployee היו שדות שישנם גם ב — Person. נרצה שלא תהיה חזרתיות ולכן נעדכן את הסכמה של עובד בחנות.

הסכמה לפני העדכונים:



נוסיף ל—Person את כל העמודות הרלוונטיות שב—StoreEmployee.



נרצה שבשונה מהחנות המקורית, עובד לא יהיה שייך בלעדית לחנות מסויימת. נעדכן בהתאם (הורדת אילוץ כתובת כמפתח זר):

מאותה סיבה נוריד את האילוץ המחייב את Address להיות מפתח ראשי ביחד עם ide של עובד

.

נחזיר את ide להיות מפתח ראשי בסכמה.

נמחק את העמודות המיותרות שיש כבר בסכמה Person.

alter table storeemployee ... | [plsqldoc3.html] [PERSON.html] | [plsqldoc5.html] [STOREEMPLOYEE.html]

SQL | Output | Statistics

```
alter table storeemployee
drop column addresss
```

2:21 | yebenjam@labdbwin | Done in 0.015 seconds

[plsqldoc4.html] [STOREEMPLOYEE.html] | alter table storeemployee ... | [plsqldoc3.html] [PERSON.html

SQL | Output | Statistics

```
alter table storeemployee
drop column namee
```

2:18 | yebenjam@labdbwin | Done in 0.052 seconds

נשנה את השם של העמודה ide ל-id עבור העקביות.

alter table storeemployee ... | [plsqldoc3.html] [PERSON.html] | [plsqldoc5.

SQL | Output | Statistics

```
alter table storeemployee
rename column ide to id
```

2:24 | yebenjam@labdbwin | Done in 0.014 seconds

הנה הסכמה לאחר כל השינויים:

select * from storeemployee | [plsqldoc3.html] [PERSON.html] | [plsqldoc5.html] [STOREEM

SQL | Output | Statistics

```
select * from storeemployee
```

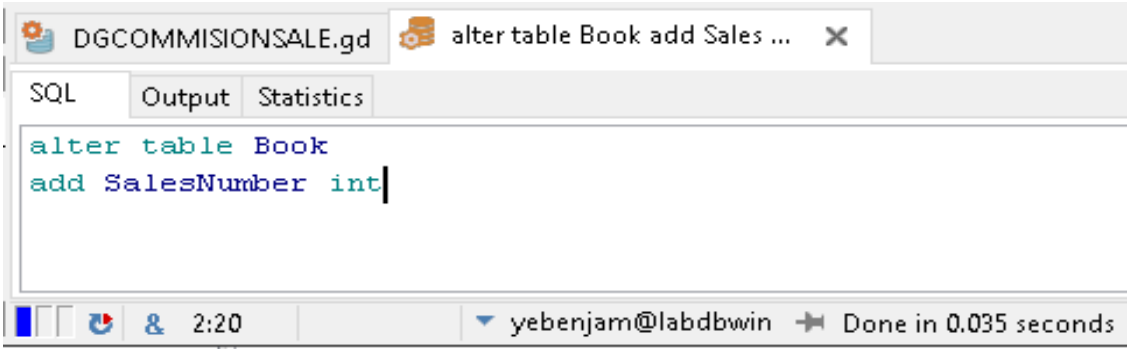
	ID	SALARYPERHOUR
1	279145228	35
2	253560004	48
3	490986260	40
4	318730295	36
5	420912967	44
6	466502852	31
7	337348088	38
8	265513588	46
9	516232913	31
10	667256907	41
11	474661393	37
12	402581942	46
13	832403234	43

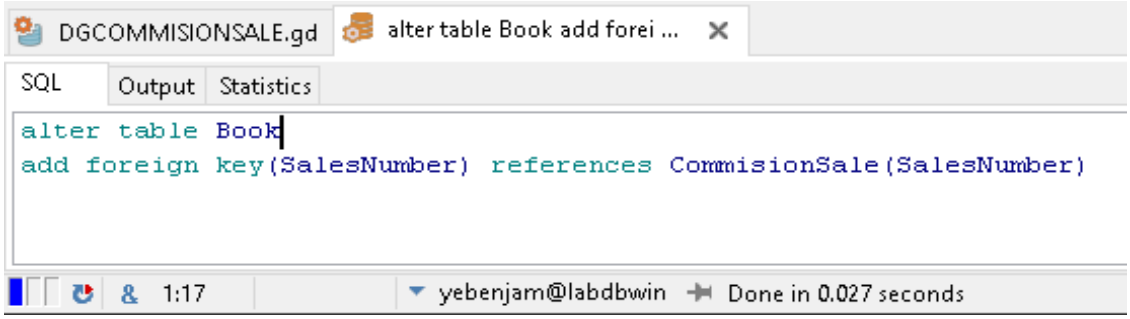
1:28 | yebenjam@labdbwin | 700 rows selected in 0.205 seconds

2.8.2 שינוי סכמת Book

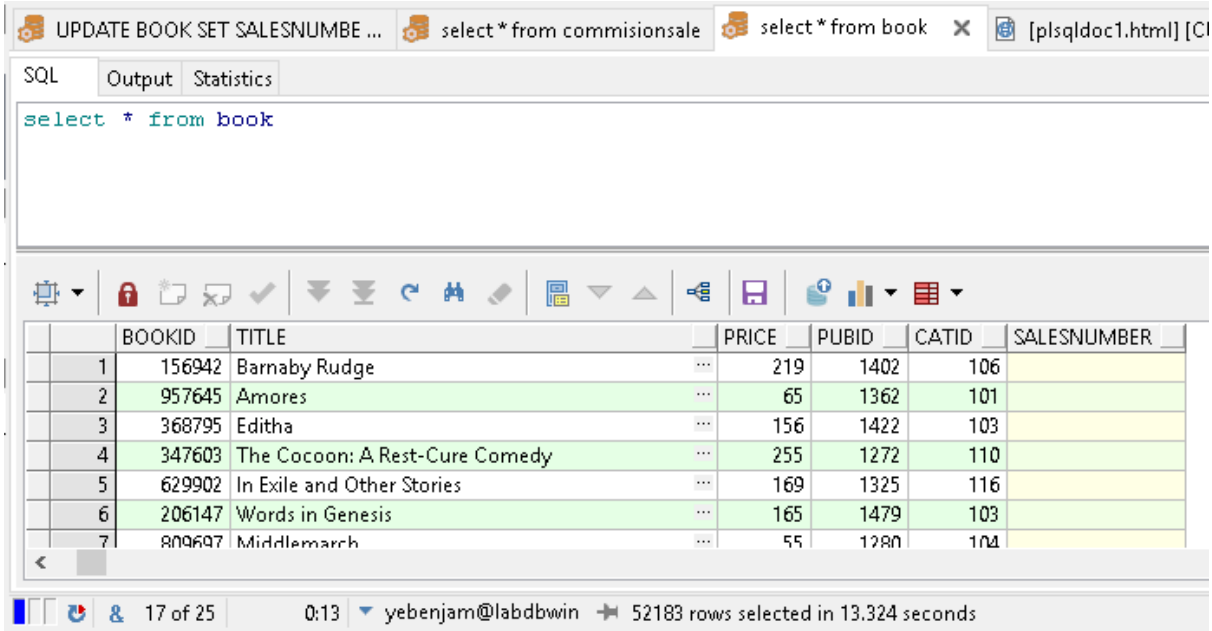
שינוי הסכמות בכדי לאפשר מכירה פומבית של ספרים:

הוספת שדה אופציונלי בשם SalesNumber שיהיה קשור ליישות CommisionSale.



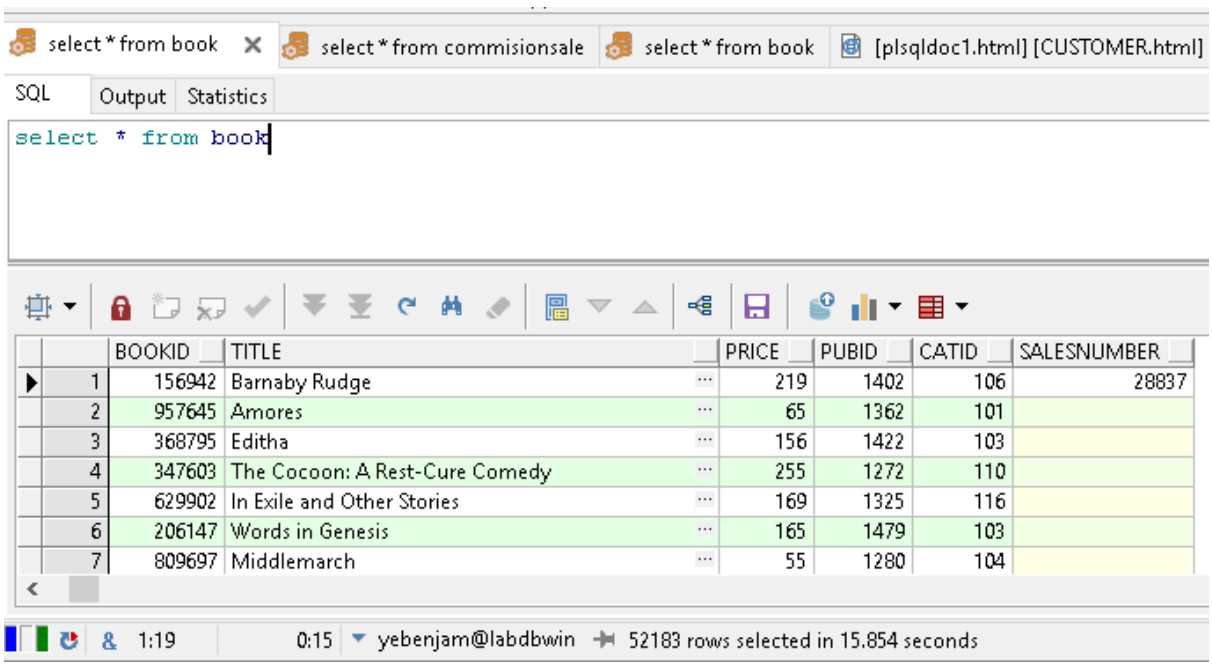


נראה שהעמודה התווספה כמו שצריך.



כעת אפשר לעשות עדכונים לטבלה ולשים ספר כנמכר במכירה פומבית.

נעשה זאת לספר ונקבל:



2.8.3 שינוי סכמת Sale

נרצה לעדכן את Sale כך שיכלול שדה אופציונלי של קוד הגמ"ח המסבסד קנייה זו.

נדאג שזה יהיה מפתח זר בסכמה.

alter table sale add forei ...

SQL Output Statistics

```
alter table sale
add foreign key(BranchCode) references Fund(BranchCode)
```

2:56 yebenjam@labdbwin Done in 0.011 seconds

נוודא שהשינויים עבדו.

```
select * from sale
```

	SALESNUMBER	DATESALE	ADDRESS	IDC	BRANCHCODE	SUBSIDYPERCENTAGE
1	66636	03/09/2017	67 Richardson Drive, Yucca	792908958		
2	48241	16/05/2018	91 Dianne Drive, Bruneck	133658580		
3	68335	11/04/2019	33 Callow Street, Nanaimo	713018789		
4	55030	21/06/2016	50 Tyler Street, San Antonio	201896355		
5	64393	09/01/2014	93 de Lancie Drive, Nordhausen	824185175		

1:19 yebenjam@labdbwin 1813 rows selected in 0.603 seconds

2.8.4 שינוי סכמת Customer

באופן דומה נעשה ש – id של Customer יצביע על מפתח של Person.

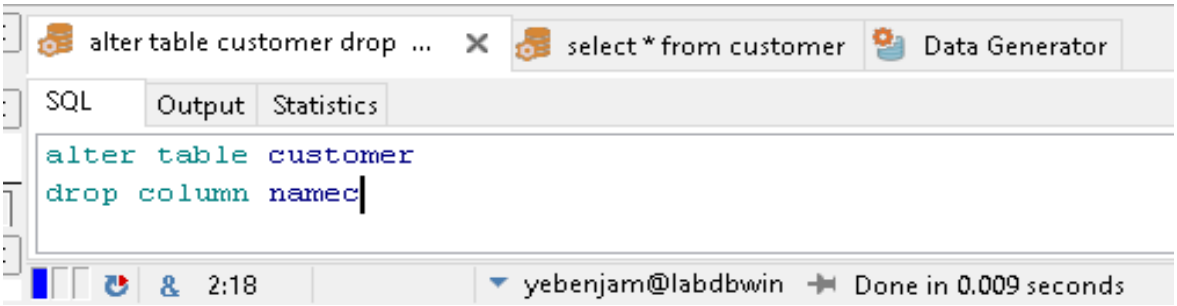
alter table customer add f ... select * from customer Data Generator

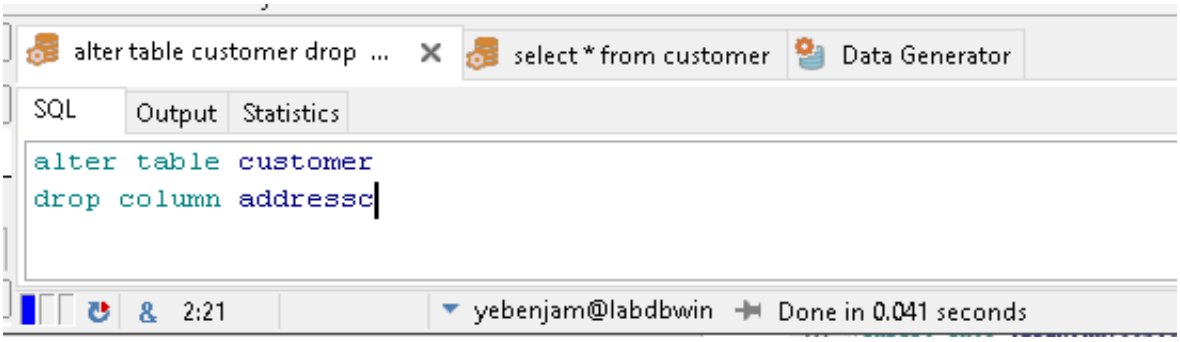
SQL Output Statistics

```
alter table customer
add foreign key(id) references person(id)
```

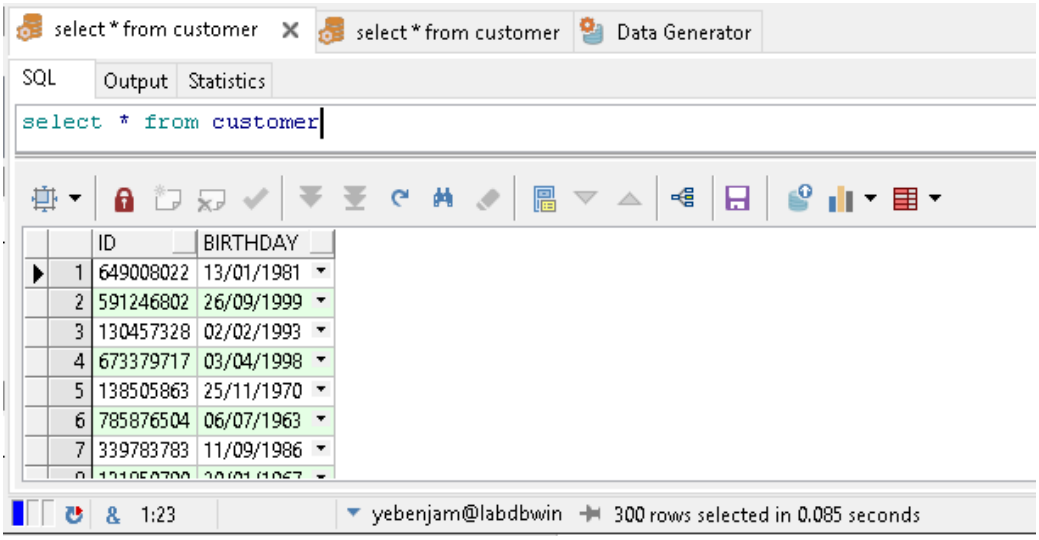
2:42 yebenjam@labdbwin Done in 0.009 seconds

מחיקת עמודות שאינן נצרכות:



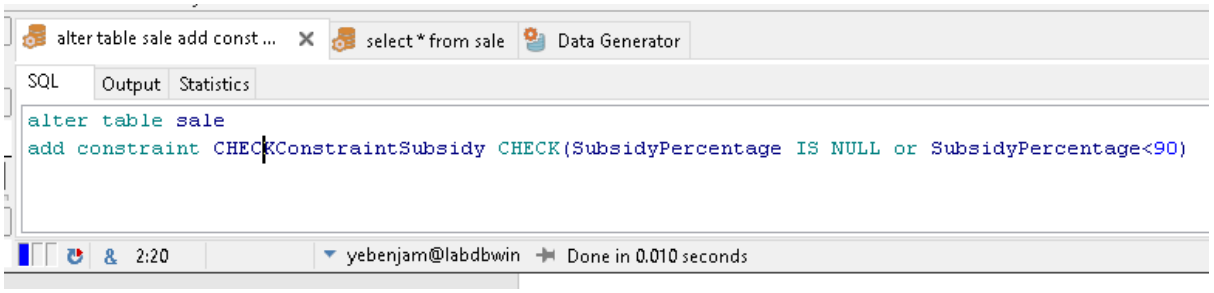


נראה את השינויים.

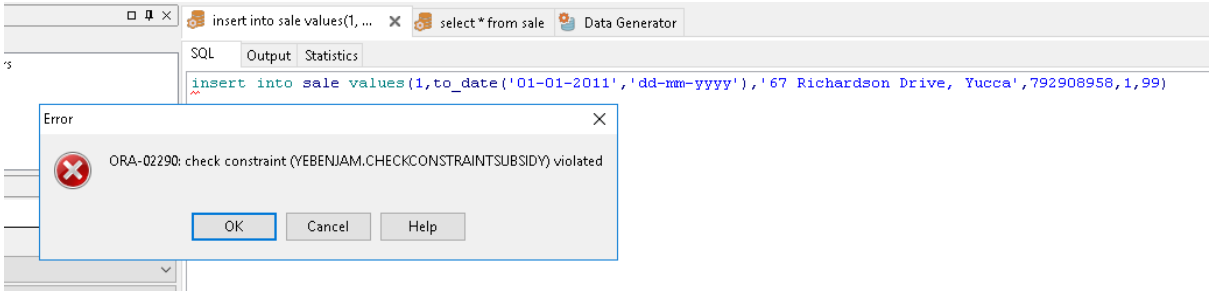


2.8.5 הוספת אילוץ חדש

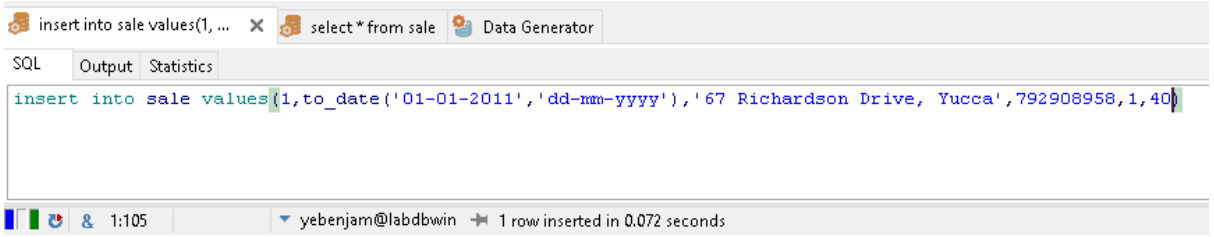
נעשה אילוץ חדש שלא תיתכן קנייה בה סבסוד הגמ"ח יעלה על סך 90 אחוזים מעלות הקנייה.



נבדוק שאכן אי אפשר להכניס נתונים באופן שכזה.



אך נוכל להכניס נתון העומד באילוץ הנ"ל.



2.9 מבטים (Views)

2.9.1 מידע לפקיד על לווה פוטנציאלי

תיאור הצורך:

ניצור מבט שעל ידו נבנה את "הטבלה שחורה", תפקיד הטבלה השחורה הינו לתת לפקיד הגמ"ח מידע האם ניתן לאשר הלוואה למבקש ההלוואה, הטבלה תכיל את תעודת הזהות, כתובת מגורים וכתובת דואר אלקטרוני, (לא נציין את השם הפרטי ושם המשפחה) של הלויים אשר שלא עמדו במועד פירעון תשלומים לגמ"ח ולכן הם לא נאמנים, או שהערבים שלהם כבר לא נאמנים.

יצירת המבט הקוד ב SQL:

```
create view Black_Table
as
select *
from (
        select p.id, p.email, p.address
        from Person p
 where p.id in(
  select r1.id
         from Recipient r1
        where (r1.credibility <> '1'
        and r1.guarantorid not in(
          select r2.id
         from recipient r2
          where r2.credibility = '1'))
 or (r1.credibility = '1'
         and r1.guarantorid in(
         select p2.id
 from Person p2 )))) p1
```

2.9.1.1 הצגת הטבלה השחורה לאחר יצירת המבט:

```
select *
from Black_Table
```

	ID	EMAIL	ADDRESS
1	824104927	drew@bestbuy.de	66 Vanian Road, Eschborn
2	593074465	rich.streep@maverick.uk	2 Carlisle Ave, Royston
3	578777984	diamond.howard@clorox.za	755 Lopez Blvd, Stellenbosch
4	389577957	darren.astin@usainstruments.co	76 Danny Ave, Tampa
5	683028244	collective.orlando@shirtfactor	18 Louise Blvd, New Delhi
6	824144990	lea.eatworld@scooterstore.com	13 Limeira Street, Laguna Bbeach
7	633667386	murray.stewart@nhr.com	416 Austin, Reston
8	232889040	ellen.blair@scooterstore.it	87 Bozeman Drive, Mantova

2.9.1.2 הצגת כל האנשים שגרים ב Austin:

SQL Output Statistics

```
select *
from Black_Table b
where b.ADDRESS like '%Austin'
```

	ID	EMAIL	ADDRESS
1	264379251	ted@shar.com	74 Desmond Street, Austin

2.9.2 סניפי הבנק אשר הלוו את סכום הכסף הרב ביותר

תיאור הצורך:

מנהל הגמ"ח מעוניין לשמור מיהם סניפי בנק אשר הלוו את סכום הכסף הרב ביותר, בדרך כלל יצירת הטבלאות נעשית לפי תקופת ריבעון שנתי, בדגמה זו נשמור מבט לפי טווח החל מ — 01/01/1990 עד ל — 01/01/2021 .

יצירת קוד המבט ב SQL:

```
create view Bank_Benefit
```

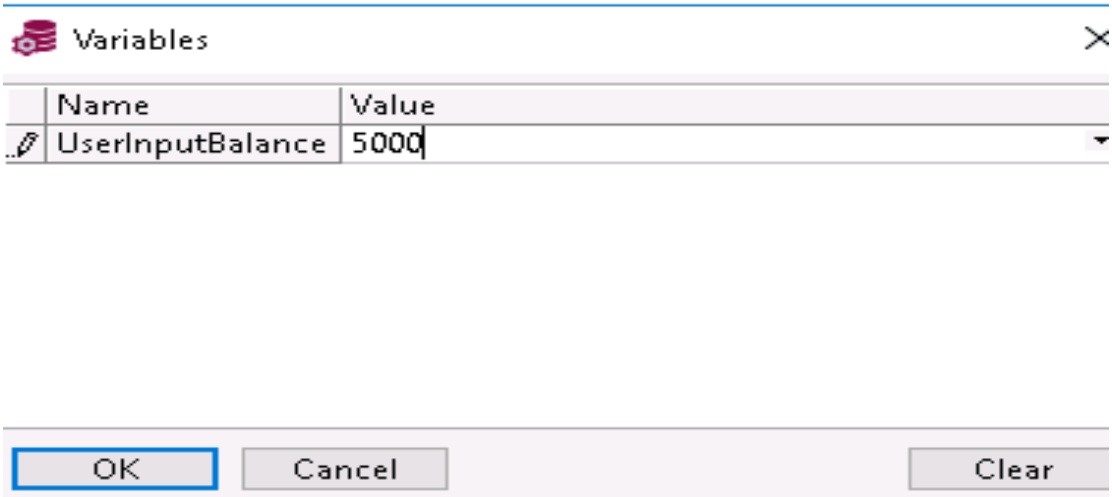
```
as
select branchcode, name, balance, sum_Of_Loan, Nun_Of_Loans
from fund f natural join (
 select l.branchcode, sum(l.initialloanamount) as sum_Of_Loan ,count(*) as Nun_Of_Loans
 from loan l
 where l.datetaken between to_date('01-01-1900','dd-mm-yyyy') and to_date('01-04-2020','dd-mm-yyyy')
 group by l.branchcode) j
order by sum_Of_Loan desc
```

2.9.2.1 משתמש בוחר סכום מקסימלי לחיפוש

הצגת סניפי הבנק אשר הלוו הכי הרבה כסף ע"י שהמשתמש בוחר סכום מקסימלי לחיפוש.

```
select *
from Bank_Benefit
where balance < '&UserInputBalance'
```

כאשר המשתמש מכניס קלט כך:



נקבל את התוצאה:

	BRANCHCODE	NAME	BALANCE	SUM_OF_LOAN	NUN_OF_LAONS
1	90	Genex Technologies	51	30577	6
2	116	Kia Motors Corp.	30	28582	5
3	35	Tilson HR	1191	27562	5
4	143	Innovate E-Commerce	443	26266	4
5	144	Noodles & Co.	1071	24356	4
6	12	Royal Gold	825	23973	4
7	173	Strategic Data Systems	812	23316	4
8	18	Hershey Foods Corp.	409	23184	4
9	39	Cendant Corp.	566	22996	4

2.9.2.2 הצגת הטבלה סניפי הבנק לאחר יצירת המבט

SQL Output Statistics

```
select *
from Bank_Benefit
```

	BRANCHCODE	NAME	BALANCE	SUM_OF_LOAN	NUN_OF_LAONS
1	90	Genex Technologies	51	30577	6
2	116	Kia Motors Corp.	30	28582	5
3	35	Tilson HR	1191	27562	5
4	143	Innovate E-Commerce	443	26266	4
5	144	Noodles & Co.	1071	24356	4
6	12	Royal Gold	825	23973	4
7	173	Strategic Data Systems	812	23316	4
8	18	Hershey Foods Corp.	409	23184	4
9	39	Cendant Corp.	566	22996	4
10	33	Mosaic	576	22366	3
11	47	Operational Technologies	166	22022	4

2.10 אינדקסים

2.10.1 זמן החזר הלוואה מוערך

השאילתה עליה נבחן את האינדקס.

השאילתה שעליה נבחן את האינדקס היא השאילתה השביעית שמטרתה להחזיר הערכה לגבי זמן סיום תשלום החוב אם מהיום והלאה הלווה יחזיר כל חודש כמו ההחזר הממוצע ששילם עד עכשיו.

```
 select * from person natural join

(select T.ID, T.LOANNUMBER, T.Loan_Amount-T.Paid_So_Far as Left_To_Pay,


Add_Months(CURRENT_DATE,ceil((T.Loan_Amount-T.Paid_So_Far)/(T.Min_Paid))) as
Predicted_End_Loan,T.Min_Paid,T.Due_Date from


(select id,min(initialLoanAmount)as Loan_Amount,

min(DateTaken) as Date_Taken, sum(SumPayment) as Paid_So_Far, min(SumPayment) as
Min_Paid, min(TotalNumberPayments) as Num_Payments, LoanNumber,

Add_Months(min(DateTaken),min(TotalNumberPayments)) as Due_Date


from Loan natural join Payments

group by id,LoanNumber)T


where T.LoanNumber in

(select LoanNumber from Loan where Add_Months(DateTaken,TotalNumberPayments)-
CURRENT_DATE>0)

and (T.Paid_So_Far<T.Loan_Amount))

where Predicted_End_Loan>CURRENT_DATE
```

```
select * from person natural join
(select T.ID, T.LOANNUMBER, T.Loan_Amount-T.Paid_So_Far as Left_To_Pay,
Add_Months(CURRENT_DATE,ceil((T.Loan_Amount-T.Paid_So_Far)/(T.Min_Paid))) as Predicted_End_Loan,T.Min_Paid,T.Due_Date from
(select id,min(initialLoanAmount)as Loan_Amount,
min(DateTaken) as Date_Taken, sum(SumPayment) as Paid_So_Far, min(SumPayment) as Min_Paid, min(TotalNumberPayments) as Num_Payments, LoanNumber,
Add_Months(min(DateTaken),min(TotalNumberPayments)) as Due_Date

from Loan natural join Payments
group by id,LoanNumber)T

where T.LoanNumber in
(select LoanNumber from Loan where Add_Months(DateTaken,TotalNumberPayments)-CURRENT_DATE>0)
and (T.Paid_So_Far<T.Loan_Amount))
where Predicted_End_Loan>CURRENT_DATE
```

	ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS	AGE	LOANNUMBER	LEFT_TO_PAY	PREDICTED_END_LOAN	MIN_PAID	DUE_DATE
1	957395911	Nicholas	Oakenfold	nicholas.oakenfold@kellogg.de	92 Rodgers, Koeln		113	4432	27/10/2021 12:44:52	916	30/01/2025
2	158333669	William	Lopez	william.lopez@tigris.au	25 Carolyn Street, Perth		226	537	27/06/2021 12:44:52	545	21/04/2023
3	160649811	Saul	Midler	saul.midler@viacom.fr	6 Anderson Road, Toulouse		12	529	27/06/2021 12:44:52	547	01/09/2026
4	535709318	Famke	Wright	fwright@learningvoyage.uk	1 Cale Drive, Sale		252	1513	27/11/2021 12:44:52	253	09/10/2023
5	937043772	Oded	Solido	oded.solido@qestrel.de	69 Liotta Road, Luedenscheid		217	1876	27/08/2021 12:44:52	838	27/03/2027
6	468307392	Allison	Jeter	a.jeter@prosum.com	6 LaPaglia Ave, Seongnam-si		61	4718	27/04/2022 12:44:52	454	05/09/2022
7	186364609	Cary	Li	cary.li@mwh.com	54 Maarssen Street, Lexington		11	1719	27/08/2021 12:44:52	645	03/04/2023

3:123 shimizra@labdbwin 7 rows selected in 0.056 seconds (more...)

זמן הריצה לפני יצירת האינדקס הינו 0.056

יצירת האינדקס:

```
create index Idx_DateTaken_Loan
on Loan(DateTaken);
```

יצרנו אינדקס על העמודה התארת את תאריך לקיחת ההלוואה בטבלת הלוואות.

כעת לאחר יצירת האינדקס נריץ את השאילתה:

SQL Output Statistics

```
select * from person natural join
(select T.ID, T.LOANNUMBER, T.Loan_Amount-T.Paid_So_Far as Left_To_Pay,
Add_Months(CURRENT_DATE,ceil((T.Loan_Amount-T.Paid_So_Far)/(T.Min_Paid))) as Predicted_End_Loan,T.Min_Paid,T.Due_Date from
(select id,min(initialLoanAmount)as Loan_Amount,
min(DateTaken) as Date_Taken, sum(SumPayment) as Paid_So_Far, min(SumPayment) as Min_Paid, min(TotalNumberPayments) as Num_Payments, LoanNumber,
Add_Months(min(DateTaken),min(TotalNumberPayments)) as Due_Date

from Loan natural join Payments
group by id,LoanNumber)T

where T.LoanNumber in
(select LoanNumber from Loan where Add_Months(DateTaken,TotalNumberPayments)-CURRENT_DATE>0)
and (T.Paid_So_Far<T.Loan_Amount))
where Predicted_End_Loan>CURRENT_DATE
```

11:23 shimizra@labdbwin 1 row selected in 0.013 seconds (more...)

ניתן לראות שכעת זמן הריצה החדש הינו 0.013 שניות.

תיאור השיפור:

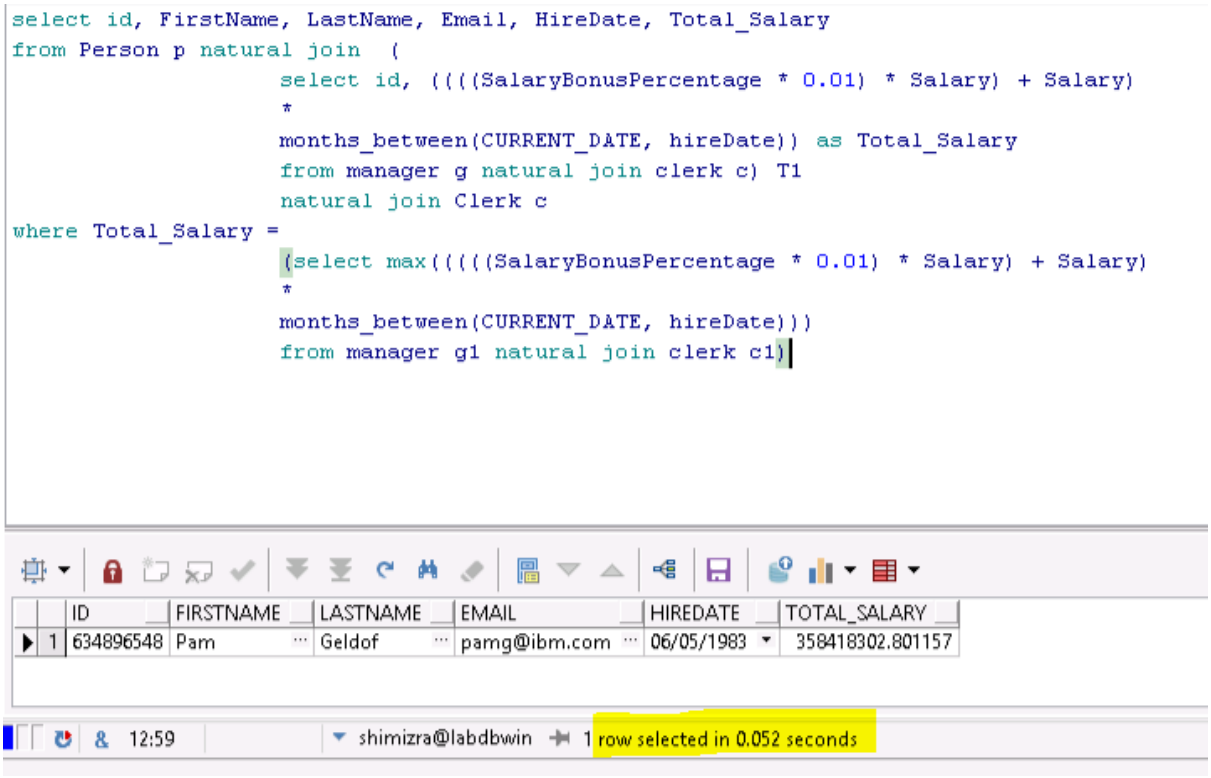
הרצת השאילתה לפני הוספת האינדקס לקחה 0.056 שניות. לאחר שהוספנו אינדקס זמן הריצה החדש נהיה 0.013שניות. רואים שהשיפור שהאינדקס יצר הוא שיפור ענק של כ 76.7% מזמן הריצה המקורי.

2.10.2 שכר רב שנתי של מנהל שקיבל את סכום הכסף הרב ביותר

השאילתה עליה נבחן את האינדקס:

השאילתה עליה נבחן את האינדקס היא שאילתה מספר 6. מטרת השאילתה להחזיר את הנתון המקסימלי של סך כל המשכורות שקיבל מנהל מיום תחילת עבודתו ועד היום (לפי מס' חודשים).

```
select id, FirstName, LastName, Email, HireDate, Total_Salary
from Person p natural join  (
        select id, ((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
        *
        months_between(CURRENT_DATE, hireDate)) as Total_Salary
        from manager g natural join clerk c) T1
        natural join Clerk c
where Total_Salary =
        (select max(((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
        *
        months_between(CURRENT_DATE, hireDate)))
        from manager g1 natural join clerk c1)
```



זמן הריצה לפני יצירת האינדקס הינו 0.052

יצירת האינדקס:

```
create index Idx_HireDate_clerk
on clerk(HireDate);
```

יצרנו אינדקס על עמודה תאריך התחלת עבודה של טבלת עובד.

כעת לאחר יצירת האינדקס נריץ את השאילתה:

```
select id, FirstName, LastName, Email, HireDate, Total_Salary
from Person p natural join  (
                    select id, ((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
                    *
                    months_between(CURRENT_DATE, hireDate)) as Total_Salary
                    from manager g natural join clerk c) T1
                    natural join Clerk c
where Total_Salary =
                    (select max(((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
                    *
```

	ID	FIRSTNAME	LASTNAME	EMAIL	HIREDATE	TOTAL_SALARY
1	634896548	Pam	Geldof	pamg@ibm.com	06/05/1983	358418469.525465

2:14 shimizra@labdbwin 1 row selected in 0.010 seconds

ניתן לראות שכעת זמן הריצה החדש הינו 0.010 שניות.

תיאור השיפור:

הרצת השאילתה לפני הוספת האינדקס לקחה 0.052 שניות. לאחר שהוספנו אינדקס זמן הריצה החדש נהיה 0.010שניות. רואים שהשיפור שהאינדקס יצר הוא שיפור ענק של כ 80% מזמן הריצה המקורי.

2.10.3 לווים שהחזירו את רוב החוב שלקחו

השאילתה עליה נבחן את האינדקס:

השאילתה עליה נבחן את האינדקס היא שאילתה מספר 3. מטרת השאילתה הינה להחזיר נתונים על כל הלווים ששילמו יותר מ50 אחוז מסך כל ההלוואות שלקחו עד כה.

```
select  ID, LastName ,FirstName ,L.Total_Loans_Amount, P.Total_Payments,
    (L.Total_Loans_Amount - P.Total_Payments) as Rest_To_Pay,
```

```
        (P.Total_Payments / L.Total_Loans_Amount) as The_Percent
from
       (select ID, sum(SumPayment) as Total_Payments
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
    natural join
        Person
where (P.Total_Payments / L.Total_Loans_Amount) > 0.5
order by LastName, FirstName
```

```
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
    natural join
        Person
where (P.Total_Payments / L.Total_Loans_Amount) > 0.5
order by LastName, FirstName
```

	ID	LASTNAME	FIRSTNAME	TOTAL_LOANS_AMOUNT	TOTAL_PAYMENTS	REST_TO_PAY	THE_PERCENT
1	384994481	Broderick	Kasey	2592	1659	933	0.640046296296296
2	589078366	Butler	Victoria	1034	1352	-318	1.30754352030948
3	507145453	Caan	Ivan	1001	1073	-72	1.07192807192807
4	301015308	Hewitt	Franco	5384	3644	1740	0.676820208023774
5	158333669	Lopez	William	1900	1363	537	0.717368421052632
6	202679017	Lynne	Rodney	2333	2005	328	0.859408486926704
7	847427209	Price	Val	4028	3028	1000	0.751737835153923
8	517243008	Sepulveda	Marc	2283	1298	985	0.568550153307052
9	273467288	Shaye	Trini	1288	1079	209	0.837732919254658
10	656054660	Soda	Maggie	2051	1420	631	0.692345197464651

15:29 shimizra@labdbwin 10 rows selected in 0.033 seconds

זמן הריצה לפני יצירת האינדקס הינו 0.033

יצירת האינדקס:

```
create index Idx_InitialLoanAmount_Loan
on Loan(InitialLoanAmount)
```

יצרנו אינדקס על עמודה תאריך סך סכום הלוואות שנלקחו לפי אדם על טבלת הלוואה.

כעת לאחר יצירת האינדקס נריץ את השאילתה:

```
select  ID, LastName ,FirstName ,L.Total_Loans_Amount, P.Total_Payments,
        (L.Total_Loans_Amount - P.Total_Payments) as Rest_To_Pay,
        (P.Total_Payments / L.Total_Loans_Amount) as The_Percent
from
       (select ID, sum(SumPayment) as Total_Payments
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
    natural join
```

	ID	LASTNAME	FIRSTNAME	TOTAL_LOANS_AMOUNT	TOTAL_PAYMENTS	REST_TO_PAY	THE_PERCENT
1	384994481	Broderick	Kasey	2592	1659	933	0.640046296296296
2	589078366	Butler	Victoria	1034	1352	-318	1.30754352030948
3	507145453	Caan	Ivan	1001	1073	-72	1.07192807192807
4	301015308	Hewitt	Franco	5384	3644	1740	0.676820208023774
5	158333669	Lopez	William	1900	1363	537	0.717368421052632
6	202679017	Lynne	Rodney	2333	2005	328	0.859408486926704
7	847427209	Price	Val	4028	3028	1000	0.751737835153923
8	517243008	Sepulveda	Marc	2283	1298	985	0.568550153307052

15:29 shimizra@labdbwin 8 rows selected in 0.015 seconds (more...)

ניתן לראות שכעת זמן הריצה החדש הינו 0.015 שניות.

תיאור השיפור:

הרצת השאילתה לפני הוספת האינדקס לקחה 0.033 שניות. לאחר שהוספנו אינדקס זמן הריצה החדש נהיה 0.015 שניות. רואים שהשיפור שהאינדקס יצר הוא שיפור ענק של כ 54% מזמן הריצה המקורי.

Grant and Revoke 2.11

בשביל להעניק ולהסיר הרשאות אני אשתמש בuser הקיים של יהודה בנימין.

ראשית נעניק ליהודה הרשאות גישה למסד הנתונים שלנו ולאחר מכן הוא יקבל הרשאות צפייה בטבלה Person, הרשאות עריכה בטבלה Fund והרשאות מחיקה בטבלה Clerk.

```
grant create session to yebenjam;
grant select on Person to yebenjam;
grant insert on Fund to yebenjam;
grant delete on Clerk to yebenjam;
```

כעת נבדוק האם הרשאות select עבור טבלה Person יעבדו עבור בנימין יהודה.

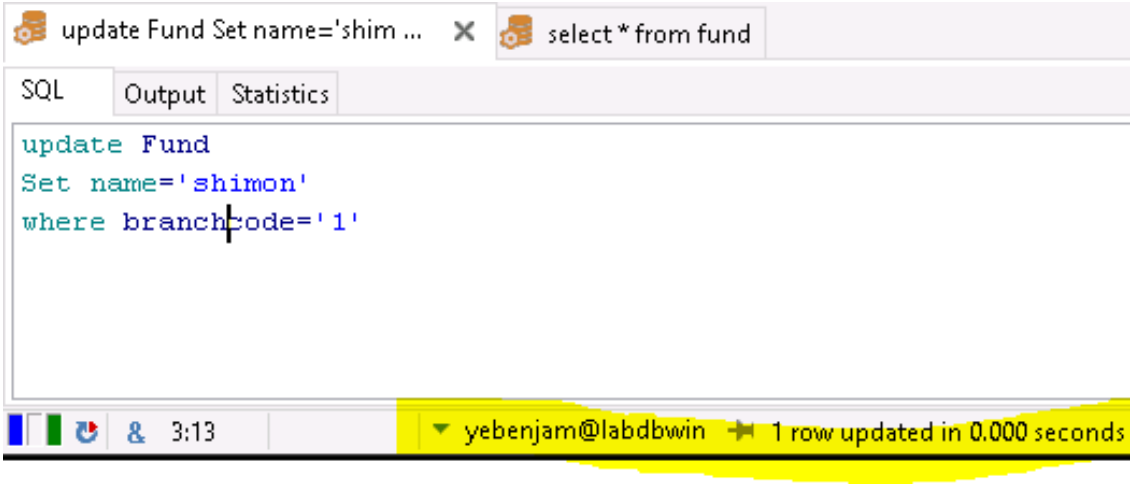
```
select *
from person
```

	ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS
1	6772116306	Randall	Conlee	randall@hospitalsolutions.com	100 Natal Drive,Kyu
2	6138868186	Alec	Union	alec@yashtechnologies.br	15 Nash Road,S?o p
3	5479264859	Rebeka	Gibson	rebeka.g@ogiointernational.com	89 Renfro Road,Mo
4	9376541493	Ronny	Tate	ronny.tate@isd.nl	96 Cobham,Veener
5	3641481556	Glenn	Coolidge	glenn.c@bps.com	19 Jackie Road,Frer
6	2405706407	Sam	Lynn	s.lynn@telepoint.com	42nd Street,Juno Be
7	3116650958	Nina	Braugher	nina.braugher@capellaeducation	58 Tankard Street,F
8	7700437898	Howard	Savage	howard.s@digitalmotorworks.pk	55 Flower mound A
9	4371854201	Toshiro	Forster	toshiro.forster@morganresearch	699 Athens Drive,Sa
10	3392354584	Laurence	Mantegna	laurence.mantegna@tilsonlandsc	293 Stans Drive,Par
11	2255077872	Clint	Bell	c.bell@sm.ca	57 Bellevue Road,W
12	3004092689	Mary Beth	Brickell	mbrickell@visionarysystems.com	48 Fishburne Road,
13	1826888505	Emmylou	Bentley	emmylou.b@primussoftware.za	79 Scott Ave,Vande

2:12 yebenjam@labdbwin 29 rows selected in 0.060 seconds (more...)

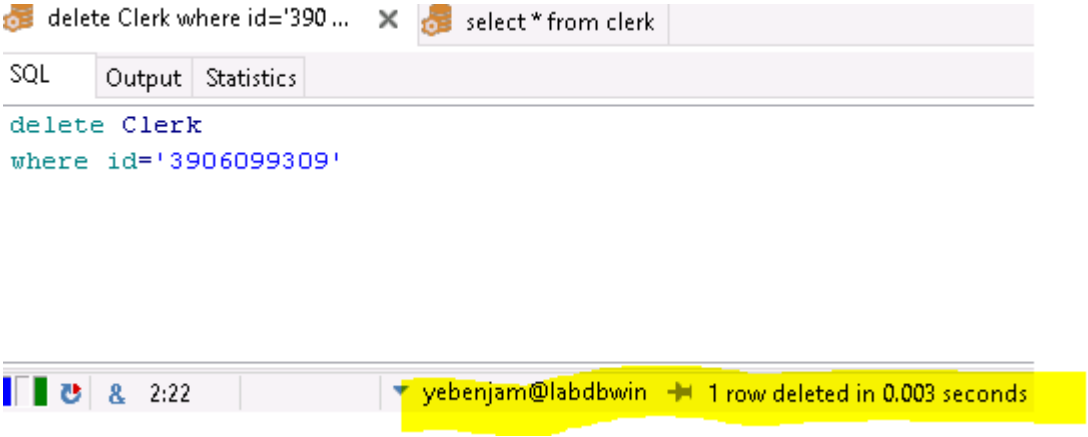
ניתן לראות ששאילתה select פעלה בהצלחה.

כעת נבדוק האם הרשאות update עבור טבלה Fund יעבדו עבור יהודה בנימין.



ניתן לראות שפעולת update פעלה בהצלחה.

כעת נבדוק האם הרשאות delete עבור טבלה Clerk יעבדו עבור בנימין יהודה.

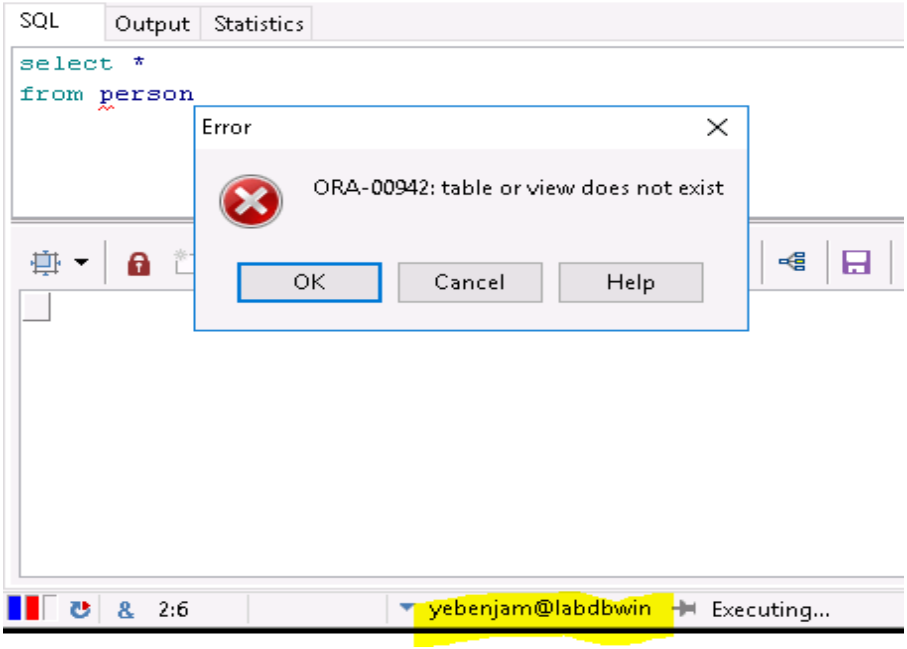


ניתן לראות שפעולת delete פעלה בהצלחה.

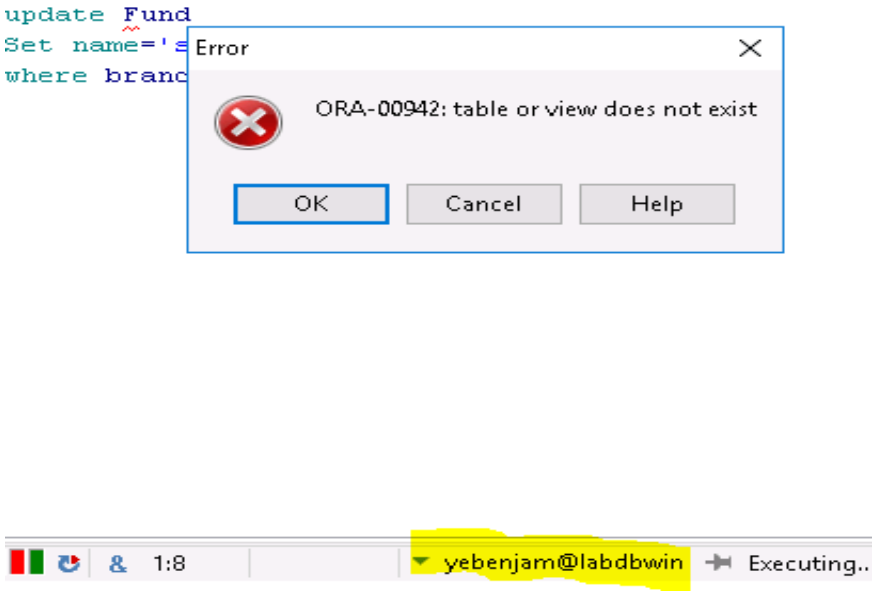
כעת לאחר שהענקנו הרשאות נרצה גם לבדוק את ביטול ההרשאות הקיימות, נשתמש בפקודת Revoke

```
revoke select on Person from yebenjam;
revoke insert on Fund from yebenjam;
revoke delete on Clerk from yebenjam;
```

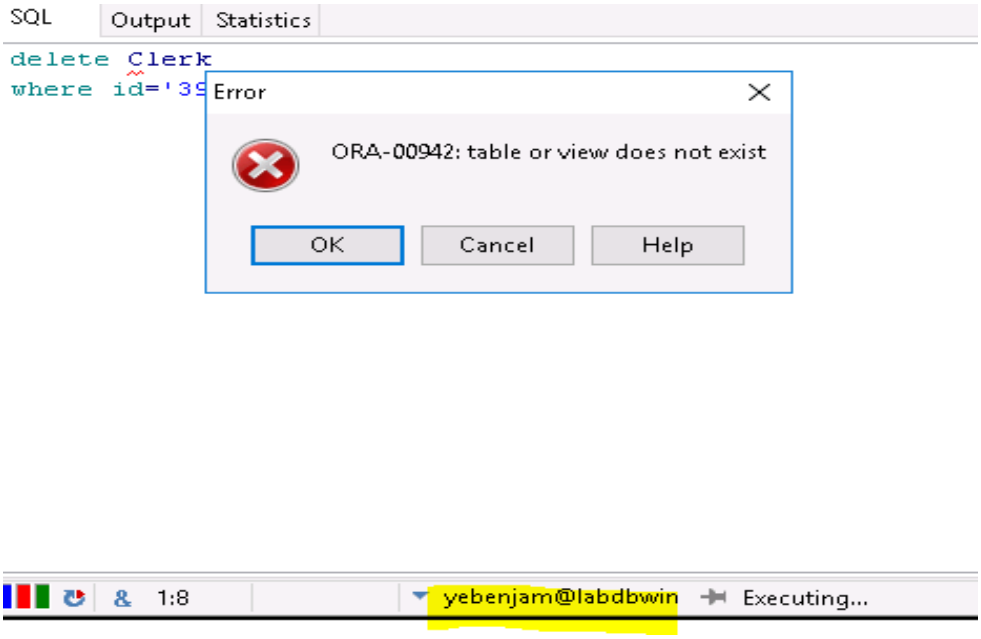
ניתן לראות שהרשאת select עבור טבלה Person נלקחה מבנימין יהודה.



ניתן לראות שהרשאת update עבור טבלה Fund נלקחה מבנימין יהודה.



ניתן לראות שהרשאת delete עבור טבלה Clerk נלקחה מבנימין יהודה.


SQL
Output
Statistics
delete Clerk
where id='39
Error
ORA-00942: table or view does not exist
OK
Cancel
Help
1:8
yebenjam@labdbwin
Executing...

2.12 שאילתות על בסיס הנתונים המשולב

2.12.1 רווחים עבור מכירות עם עמלה

מנכ"ל הגמ"ח כרת הסכם עם בעל החנות ובעלי הספריה שאת כל העמלות ממכירת ספרי ספריה שבעל החנות ימכור יעברו לגמ"ח.

המנכ"ל מעוניין לדעת מה הרווחיות של המכירות, ומאיזה קטגוריות הכי כדאי למכור.

יציג את סכום הרווחים ואחוזי העמלה הממוצעים עבור ספרים מהספרייה שנמכרו במכירה פומבית השייכים לקטגוריות שונות.

הקוד ב– SQL

```
select catid,name,sum(price*commisionPercentage*0.01) as
PROFIT_FOR_GMACH,round(avg(CommisionPercentage),2) as AVG_COMMISION

from book natural join commisionSale natural join Category

group by catid,name

order by PROFIT_FOR_GMACH desc
```

תוצאת השאילתה:

	CATID	NAME	PROFIT_FOR_GMACH	AVG_COMMISION
1	110	Art	611.27	42.56
2	104	Science fiction	548.97	38.88
3	108	Science	524.71	31.57
4	107	Diaries	480.35	33
5	112	Science	315.24	38.57
6	106	Mystery	292.32	35.17
7	101	Action and Adventure	251.54	30.8
8	115	Guide	201.55	16.4
9	109	Science fiction	179.18	21.6
10	114	Romance	170.5	28.17
11	116	Fantasy	161.91	18
12	113	Biographies	114.05	13.75
13	102	Drama	93.01	25.33
14	111	Science	91.12	19.4
15	103	Art	88.64	22.5

1:107 yebenjam@labdbwin 16 rows selected in 0.031 seconds

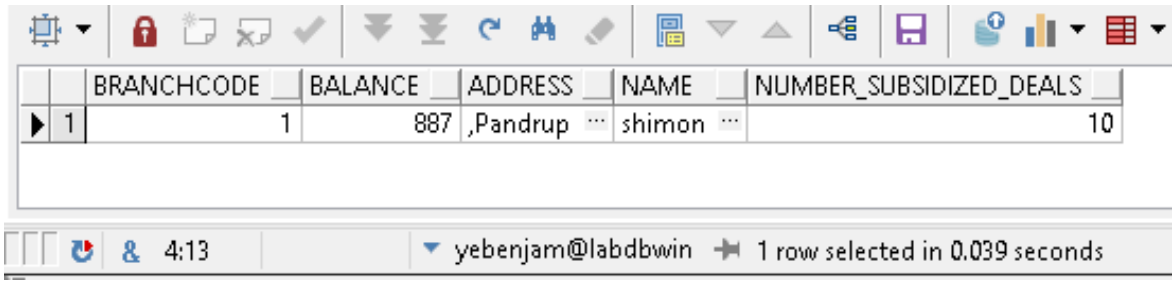
2.12.2 סניף הגמ"ח שסיבסד את כמות הקניות המרבי

בכדי לדעת טוב יותר מהם האיזורים בארץ הזקוקים לעזרה פיננסית במידה הרבה ביותר, ניצור שאילתה שתחזיר את פרטי סניף הגמ"ח שסבסד הכי הרבה קניות בחנות מבין כל סניפי הגמ"ח.

הקוד ב – SQL

```
select * from Fund natural join(select BranchCode,count(*) as
NUMBER_SUBSIDIZED_DEALS from Sale

where BranchCode in (select branchcode from fund)

group by BranchCode

having count(salesNumber)>=ALL

(select count(salesNumber)

from Sale

where BranchCode in (select BranchCode from fund)

group by BranchCode))
```

תוצאת השאילתה:



	BRANCHCODE	BALANCE	ADDRESS	NAME	NUMBER_SUBSIDIZED_DEALS
1	1	887	,Pandrup	shimon	10

4:13 yebenjam@labdbwin 1 row selected in 0.039 seconds

חלק ג'

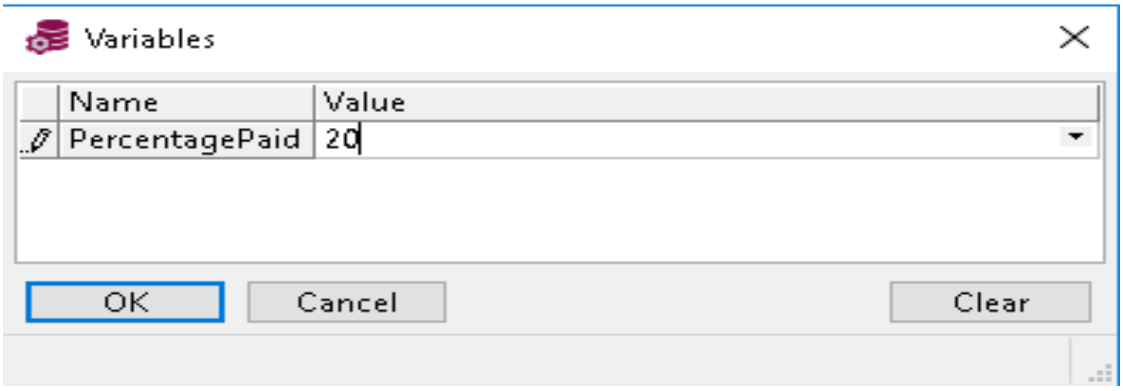
3.1 שאילתות עם פרמטרים

3.1.1 מידע על חובות שאחוז מסויים מהם כבר שולם

יקבל מהמשתמש מספר המייצג אחוזי הפירעון בין 0 ל–100 ויחזיר את כל פרטי הלווים וההלוואה שהחזירו יותר מאחוז זה מהלוואותיהם.

קוד ב–SQL:

```
select  ID, LastName ,FirstName ,L.Total_Loans_Amount, P.Total_Payments,
        (L.Total_Loans_Amount - P.Total_Payments) as Rest_To_Pay,
        round((P.Total_Payments / L.Total_Loans_Amount)*100,0) as The_Percent
from
       (select ID, sum(SumPayment) as Total_Payments
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
    natural join
        Person
where (P.Total_Payments / L.Total_Loans_Amount) > &<name="PercentagePaid"
type="integer" required="true">*0.01
order by LastName, FirstName
```



עבור מספר זה כפרמטר נקבל:

	ID	LASTNAME	FIRSTNAME	TOTAL_LOANS_AMOUNT	TOTAL_PAYMENTS	REST_TO_PAY	THE_PERCENT
4	259174931	Pitt	Mindy	7908	1732	6176	22
5	535709318	Wright	Famke	9076	1962	7114	22
6	776634794	Valentin	Kasey	3379	862	2517	26
7	927725375	Quatro	Amy	4124	1172	2952	28
8	754519550	Ammons	Julianna	3794	1050	2744	28

9:64 shimizra@labdbwin 27 rows selected in 0.031 seconds

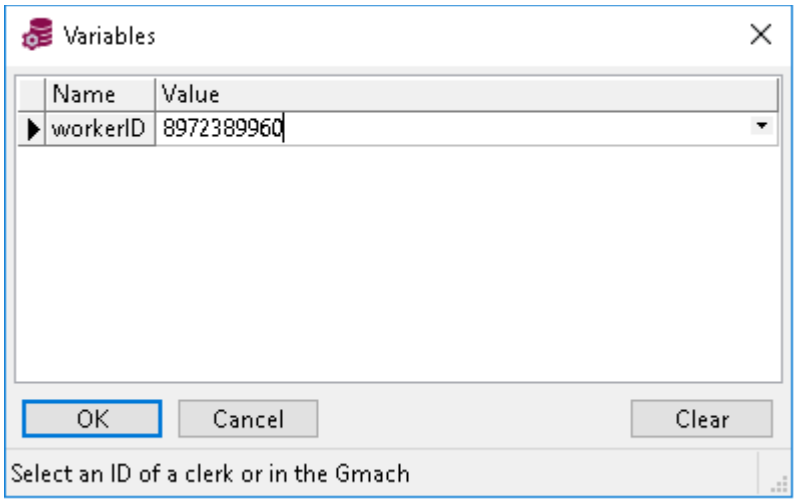
3.1.2 סכום סך כל המשכורות עבור עובד כלשהו

מנהל הגמ"ח מעוניין לדעת כמה הוציאה הגמ"ח על כל עובד. השאילתה תקבל כקלט את מספר תעודת הזהות של העובד ותפיק את סך ההוצאות שהגמ"ח הוציא על עובד זה מיום קליטתו בגמ"ח ועד עכשיו, יחד עם כל פרטיו האישיים.

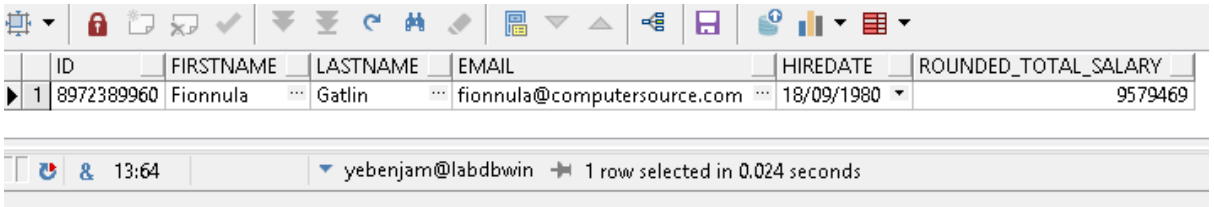
קוד הSQL:

```
select id, FirstName, LastName, Email, HireDate, round(Total_Salary,0) as
Rounded_Total_Salary
from Person p natural join  (
        select min(T.id) as id,min(T.hireDate) as hireDate,sum(salary) as Total_Salary
        from
        ((select id,hireDate,salary
        *
        months_between(CURRENT_DATE, hireDate) as salary
        from clerk
        where id=&<name="workerID" list = "select id from Clerk order by id"
        required=true hint="Select an ID of a clerk in the Gmach" type="integer">
        )
        UNION
        (select id,hireDate,salaryBonusPercentage *
        months_between(CURRENT_DATE, hireDate)*0.01*salary
        from clerk  natural join manager
        where id=&<name="workerID" list = "select id from Clerk order by id"
        required=true hint="Select an ID of a clerk in the Gmach" type="integer">
        ))T)
```

נחפש פרטים על עובד באופן שרירותי בגמ"ח:



נלחץ על OK ונקבל:



	ID	FIRSTNAME	LASTNAME	EMAIL	HIREDATE	ROUNDED_TOTAL_SALARY
1	8972389960	Fionnula	Gatlin	fionnula@computersource.com	18/09/1980	9579469

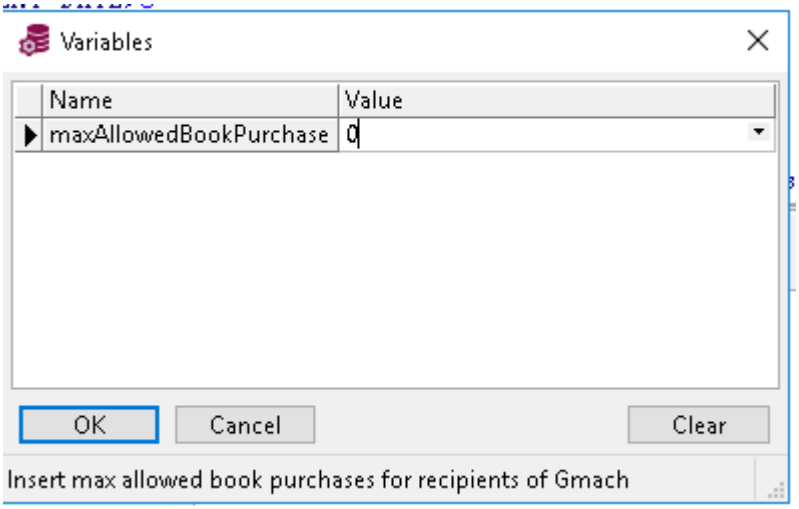
13:64 yebenjam@labdbwin 1 row selected in 0.024 seconds

3.1.3 מידע על מי שקונה הרבה ספרים לפני הזמן האחרון לפירעון

השאילתה תחזיר מידע על אנשים שקנו יותר מכמות מסוימת של ספרים לפני הזמן האחרון לפירעון, כגון תעודת זהות, שם פרטי, שם משפחה וכתובת.

```
select id,firstName,lastName,address
from person natural join
(select id from Recipient
where id in
(
select id from loan
where Add_Months(dateTaken,totalNumberPayments)-CURRENT_DATE>0
and id in
(select id from sale where id in
(select id from commisionSale))
group by id
having count(*)>=&<name="maxAllowedBookPurchase" required=true hint="Insert max
allowed book purchases for recipients of Gmach" type=integer>))
order by id
```

נכניס מספר 0 כפרמטר:



ונקבל:

	ID	FIRSTNAME	LASTNAME	ADDRESS
1	158333669	William	Lopez	25 Carolyn Street, Perth
2	159254374	Freddy	LaMond	76 Campbell Drive, Lake Forest
3	160649811	Saul	Midler	6 Anderson Road, Toulouse
4	167714899	Aidan	Bell	84 Rancho Palos Verdes Road, Melrose par
5	186364609	Cary	Li	54 Maarssen Street, Lexington
6	206258970	Marianne	Washington	59 Ljubljana Road, Trento
7	259174931	Mindy	Pitt	392 Stormare Blvd, Lengdorf
8	271124379	Marina	Tyson	55 Mika Street, Pusan-city
9	283593811	Balthazar	Moffat	76 de Lancie, Niigata
10	301015308	Franco	Hewitt	9 McDowell Street, Berkshire
11	313603065	Terri	Dunst	62nd Street, Clark
12	316266651	Selma	Brosnan	96 Ingelheim Street, Milton Keynes
13	345075034	Nicolas	Leto	76 Blair Road, Warrenton
14	454075156	Queen	Leary	96 Varzea grande Road, Cincinnati
15	468307392	Allison	Jeter	6 LaPaglia Ave, Seongnam-si
16	524114978	Carla	Red	50 Ingram Road, Valencia
17	535709318	Famke	Wright	1 Cale Drive, Sale
18	536769726	Mae	Byrd	2 Tobolowsky Road, Westport
19	578777984	Diamond	Howard	755 Lopez Blvd, Stellenbosch
20	633667386	Murray	Stewart	416 Austin, Reston
21	689042514	Rupert	Hartnett	21 Trento Drive, Herzlia
22	765108935	Adina	Mould	16 Laurel Blvd, G?vle
23	806972634	Juan	Roy Parnell	81 Strathairn Drive, Juneau
24	925242356	Patti	Kweller	374 Kristin Street, Cottbus

11:12 shimizra@labdbwin 28 rows selected in 0.050 seconds

3.1.4 קבלת מידע על אנשים שלא מחזירים הלוואות

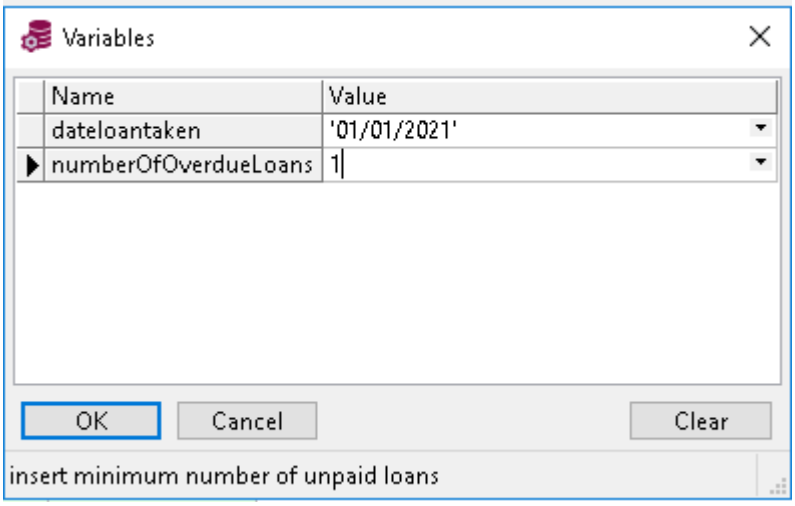
השאילתה תקבל כפרמטרים תאריך לקיחת החוב, וכמות ההלוואות המינימלית שלווה לקח ולא החזיר לה אפילו תשלום אחד.

היא תחזיר את פרטי האנשים שלא מחזירים הלוואות.

הקוד ב – SQL:

```
select * from person natural join (select id, sum(initialloanamount) as Owed_Money
from loan
where DateTaken < to_date(&dateloantaken,'dd/mm/yyyy')
group by id
having id in
(select id from loan natural left outer join payments
where sumpayment IS NULL
group by id
having count(*)>=&<name=numberOfOverdueLoans type=integer required=true hint="insert
minimum number of unpaid loans">))
order by OWED_MONEY desc
```

נכניס בתור פרמטרים:



כתוצאה נקבל את פרטי האנשים הבאים:

	ID	FIRSTNAME	LASTNAME	EMAIL	ADDRESS	OWED_MONEY
1	3461829111	Maureen	Rickles	maureen.rickles@tlsservicebure	89 Merchant Road,Reading	93323
2	2718724758	Bradley	Hagerty	bradley.hagerty@investorstitle	33 Plummer Drive,Chaam	87063
3	5063171775	Jackson	Mattea	jackson.m@hondamotor.es	49 Li Drive,El Masnou	80311
4	3929390584	Miriam	Vicious	m.vicious@keith.com	46 Scott Drive,Hercules	73320
5	8528991943	Davey	Furay	dfuray@jma.nl	11st Street,Oosterhout	70885
6	8177467898	Woody	Hedaya	woody.hedaya@kiamotors.ch	66 Pulheim-brauweiler,Gen?ve	64588
7	6868289226	Burton	Platt	burtonp@paintedword.com	28 Dubai Street,Kyunnam	62331
8	7644184436	Andrew	Sanchez	andrew.sanchez@hitechpharmacal	13rd Street,Nancy	62082
9	8937424788	Victor	Aykroyd	victor.aykroyd@infopros.za	91st Street,Gauteng	61571
10	2180990499	Claude	Balaban	claude.balaban@hersheyfoods.jp	62nd Street,Saitama	60933
11	4495495037	Jody	Foster	jody.foster@appriss.uk	25 Tippe,Berkshire	60839
12	5373245280	Bebe	Zeta-Jones	bzetajones@capitalbank.com	13 Doucette Road,Bedford	58577

9:110 yebenjam@labdbwin 32 rows selected in 0.048 seconds

3.2 דוחות

3.2.1 עלויות סבסוד קניות

מנהל הגמ"ח רוצה לדעת את נתוני סבסוד הקניות בשנה הלועזית האחרונה. לשם כך נכין דו"ח על סכומי הכסף שהושקעו בסניפים בסבסוד קניות של אנשים נזקקים בחנות.

קוד הSQL:

```
select BranchCode, sum(priceToPay) as SubsidyExpenses
 from fund natural join
(SELECT round((sum(price*purchaseAmount)-
((discount/100)*sum(price*purchaseAmount)))*(subsidyPercentage*0.01),0) as
priceToPay,salesNumber,branchCode
from insale natural join product natural join sale
where subsidyPercentage IS NOT null
and DateSale between to_date('01-01-2020','dd-mm-yyyy') and to_date('01-01-2021','dd-
mm-yyyy')
group by salesNumber,branchCode,discount,subsidyPercentage
)
group by branchCode

order by SubsidyExpenses
```

יצא:

Support Figures For Shopping

Branchcode	Subsidyexpenses
153	26999
35	90398
66	107207
1	136119

9:21 yebenjam@labdbwin 4 rows selected in 0.048 seconds

3.2.2 עשרת המנהלים שקיבלו את סכומי המשכורות הגבוהים ביותר

כחלק מההכנה לרישום הגמ"ח באופן חוקי כעמותה, נכין דו"ח על עשרת המנהלים שהרוויחו את סכומי הכסף הרבים ביותר מהגמ"ח מיום תחילת עבודתם עד היום בכדי שרשם העמותות יוכל להכריע אם היה ניהול תקין של הגמ"ח עד עכשיו.

שאילתה זו מבוססת על שאילתה 1.12.6 – "קבלת נתונים של תשלומים למנהל".

קוד ה – SQL:

```
select id, FirstName, LastName, Email,Rating, HireDate, round(Total_Salary,0) as
Rounded_Total_Salary
from Person p natural join  (
            select id, ((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
            *
            months_between(CURRENT_DATE, hireDate)) as Total_Salary
            from manager g natural join clerk c) T1
            natural join Clerk c
where Total_Salary>=any
            (select (((((SalaryBonusPercentage * 0.01) * Salary) + Salary)
            *
            months_between(CURRENT_DATE, hireDate)))
            from manager g1 natural join clerk c1)
order by ROUNDED_TOTAL_SALARY desc
fetch first 10 rows only
```

נעדכן את הגדרות התצוגה של הדו"ח עבור נראות טובה יותר:

SQL | Layout | Options

Auto Update

	Item	Description	Style	Header	Align	Format	Break	Sum
☑	Report Title	Top Ten Multi-year Earne	Default Rep		Default			
☑	Variables		Custom...		Default			
	Tabular Tables							
	Form Tables							
	Default Field							
☑	ID		Custom...		Left			
☑	FIRSTNAME		Custom...		Left			
☑	LASTNAME		Custom...		Left			
☑	EMAIL		Custom...		Left			
☑	RATING		Custom...		Left			
☑	HIREDATE		Custom...		Left			
☑	ROUNDED_TOTAL		Custom...		Left			

קיבלנו:

Top Ten Multi-year Earners

Id	Firstname	Lastname	Email	Rating	Hiredate	Rounded Total Salary
8972389960	Fionnula	Gatlin	fionnula@computersource.com	7	18/09/1980	9581336
3078331072	Jena	Westerberg	j.westerberg@microsoft.br	8	21/01/1985	8110443
7700437898	Howard	Savage	howard.s@digitalmotorworks.pk	2	26/01/1980	7517374
9882571566	Richie	Atkinson	richie.a@trafficmanagement.com	4	30/07/1983	7435874
6136570014	Nicole	Supernaw	nicole.supernaw@operationaltec	6	10/11/1981	7424593
9216667619	Rosanne	Jackson	rosanne.jackson@atlanticcredit	4	04/09/1985	7363942
3052787541	Yaphet	Biggs	yaphet.biggs@dcgroup.com	8	01/05/1984	7183989
2971959118	Elle	Hauer	ehauer@accessus.com	3	28/02/1993	7181183
5635035328	Dermot	Peebles	dermot.peebles@kellogg.de	9	01/05/1990	7078563
3000446743	Vivica	Moss	vmoss@cascadebancorp.com	8	02/02/1993	7011529

1:1 yebenjam@labdbwin 10 rows selected in 0.029 seconds

3.3 גרפים

3.3.1 אחוזי רווחים ממכירת ספרים

הגרף ממחיש אילו קטגוריות הסבו לגמ"ח מה היחס בין העמלה ממכירות ספרים לבין סכום המכירה.

הגרף מבוסס על שאילתה 2.12.2.

קוד ה–SQL:

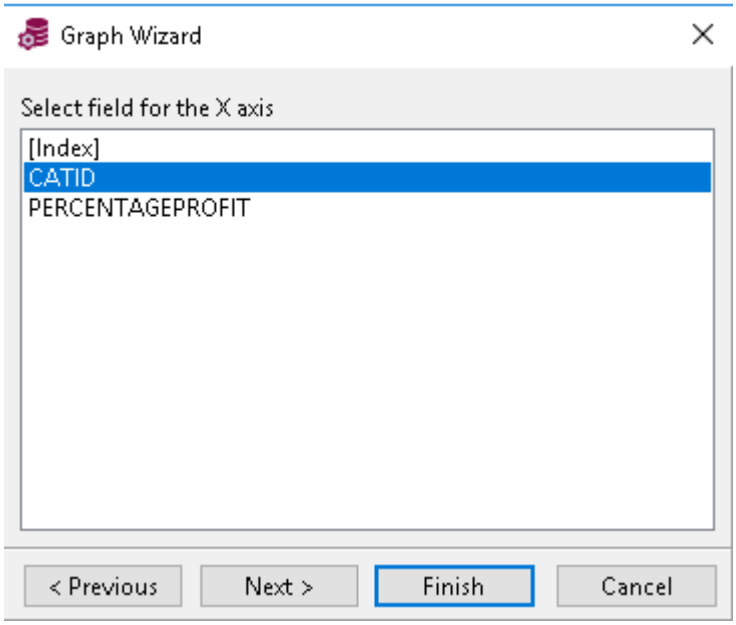
```
select name,round(sum(price*commisionPercentage*0.01)/sum(price)*100,0) as
PercentageProfit
from book natural join commisionSale natural join category
group by catid,name
order by PercentageProfit
```

קיבלנו טבלה:

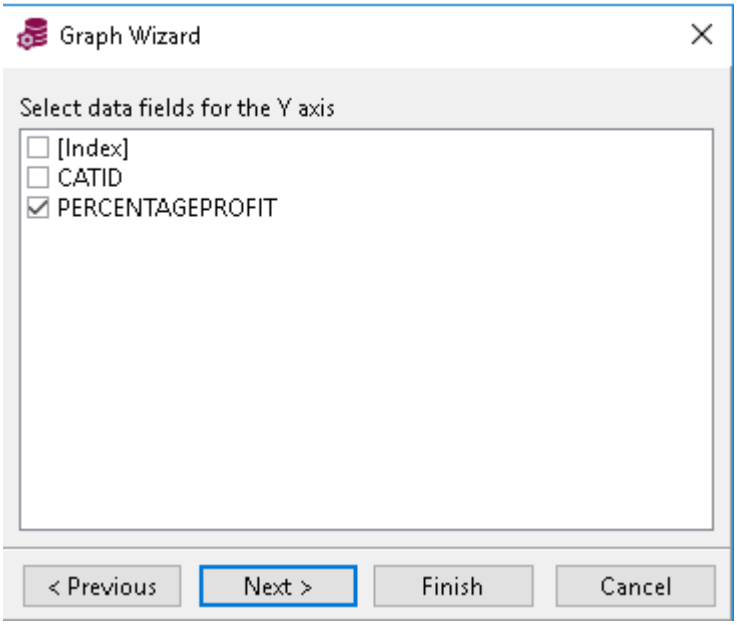
	CATID	PERCENTAGEPROFIT
1	105	11
2	113	12
3	111	13
4	115	18
5	102	18
6	116	23
7	109	24
8	103	27
9	114	28
10	106	30
11	101	31
12	107	34
13	108	36
14	112	39
15	104	40
16	110	46

1:1 yebenjam@labdbwin 16 rows selected in 0.025 seconds

נבחר את מספר הקטגוריה להיות בציר ה X:

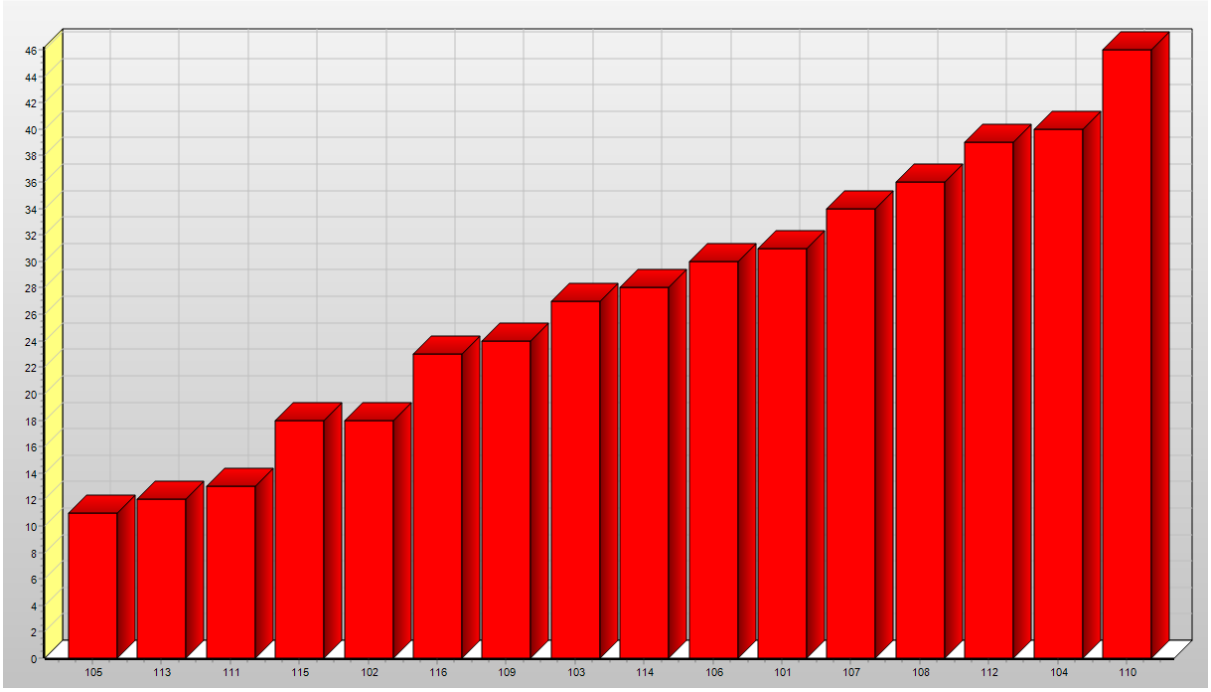


נבחר את אחוז הרווחים להיות בציר ה-Y:



בס"ד מיני פרויקט בסיסי נתונים גמ"ח לב, בנק הלוואות ללא ריבית

נקבל גרף:



3.3.2 כמות תשלומי הבונוסים למנהלים ביחס לעלות תשלום כוח האדם הכללי

הגרף מציג מה העלות של תשלומי הבונוסים למנהלים ביחס לעלות העסקת כל העובדים.

קוד ה –SQL:

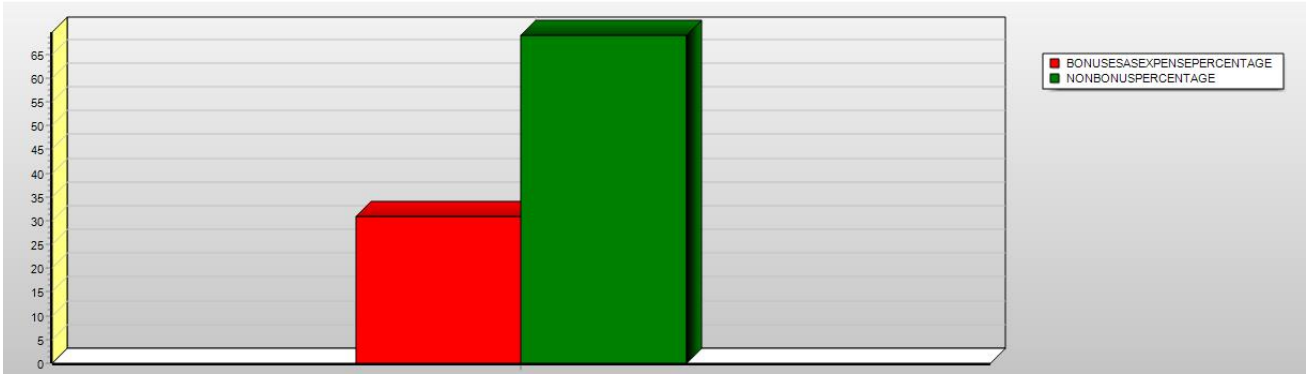
```
Select
round(sum(salary*0.01*salaryBonusPercentage)/sum(salary+salaryBonusPercentage*0.01*
salary)*100,0) as BonusesAsExpensePercentage,
100-
round(sum(salary*0.01*salaryBonusPercentage)/sum(salary+salaryBonusPercentage*0.01*
salary)*100,0) as nonBonusPercentage
from clerk natural left outer join manager
```

נקבל טבלה:

	BONUSESASEXPENSEPERCENTAGE	NONBONUSPERCENTAGE
1	31	69

4:1 yebenjam@labdbwin 1 row selected in 0.014 seconds

ניצור גרף. קיבלנו:



3.4 פרוצדורות

3.4.1 עדכון מוניטין של לווים הגונים

תיאור פרוצדורה:

מנהלי הגמ"ח בקשו שבכול סיום שנה יש לבצע מעבר על כל הלויים של הגמ"ח, וכל לווה ששילם את כל סכום הלוואותיו והמוניטין שלו קטן מ 5 אזי לעדכן את המוניטין שלו ל 5.

פרוצדורה זו אינה מקבלת פרמטרים מכיוון שפועלת על כל הלויים באופן שווה.

קוד ה – sql:

```
create or replace procedure UpdateRecipientCredibility  is
--vars
sumOfLoan integer:=0;
-- recipient tbl Iterator.
cursor aRecipient is
select *
from recipient;
aIterator recipient%rowtype;
--table of recipient who pay all louns.
cursor PayAllLouns is
select ID
from
      (select ID, sum(SumPayment) as Total_Payments
       from Loan natural join Payments
       group by ID) P
   natural join
      (select ID, sum(initialLoanAmount) as Total_Loans_Amount
       from Loan
       group by ID) L
where (P.Total_Payments - L.Total_Loans_Amount) >= 0;
bIterator PayAllLouns%rowtype;

-- begin
begin
  --interal for Recipient.
  for aIterator in aRecipient
    loop
    --external for PayAllLouns.
    for bIterator in PayAllLouns
      loop
      if ((bIterator.id = aIterator.id) and (aIterator.credibility <> 5)) then
        dbms_output.put_line('Recipien id: '||bIterator.id||' finished is debt!');
        dbms_output.put_line('The current credibility is: '||aIterator.credibility||' before update is
credibility.');
        dbms_output.put_line('The system modify the credibility from: '||aIterator.credibility||' to
5.');
        --Update query.
        UPDATE recipient
        SET credibility=5
        WHERE id= bIterator.id;
      end if;
```

```
    end loop;
  end loop;
end UpdateRecipientCredibility ;
```

נראה כעת את כל הלויים של הגמ"ח ששילמו את כל סכום ההלוואות שלקחו לפני ביצוע הפרוצדורה:

```
select ID, credibility
from
       (select ID, sum(SumPayment) as Total_Payments
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
        natural join recipient
where (P.Total_Payments - L.Total_Loans_Amount) >= 0;
```

	ID	CREDIBILITY
1	507145453	3
2	589078366	3

ניתן לראות שהמוניטין שלהם שווה לשלוש.

כעת נבצע את הפרוצדורה:

```
declare
begin
  UpdateRecipientCredibility;
end;
```

נקבל הדפסה מפורטת למסך על אודות הפעולה כפלט מהפרוצדורה.

```
Recipien id: 589078366 finished is debt!
The current credibility is: 2 before update is credibility.
The system modify the credibility from: 3 to 5.
Recipien id: 507145453 finished is debt!
The current credibility is: 2 before update is credibility.
The system modify the credibility from: 3 to 5.
```

כעת נריץ שוב את השאילתה על מנת לראות שהפרוצדורה אכן עדכנה את הנתונים.

```
select ID, credibility
from
       (select ID, sum(SumPayment) as Total_Payments
        from Loan natural join Payments
        group by ID) P
    natural join
       (select ID, sum(initialLoanAmount) as Total_Loans_Amount
        from Loan
        group by ID) L
        natural join recipient
where (P.Total_Payments - L.Total_Loans_Amount) >= 0;
```

	ID	CREDIBILITY
1	507145453	5
2	589078366	5

ניתן לראות שהעדכון בוצע בהצלחה.

3.4.2 הגדלת מאזן של סניפים עקב מגפת הקורונה

תיאור פרוצדורה:

עקב משבר וירוס הקורונה, כמות מבקשי ההלוואות הכפיל את עצמו, לפיכך מנהל גמ"ח הלוואות לב מעוניין להזרים כספיים לסניף גמ"ח שבו יש את הביקוש ההלוואות הגבוהה ביותר, בטווח תאריכים של תקופת וירוס קורונה, כך שיכפיל את מאזן החשבון של הסניף הנ"ל.

בפרוצדורה זו נשתמש בפונקציה מספר 2 שייצרנו, כאמור תפקידה של הפונקציה הוא להחזיר את סכום כל ההלוואות שסניף גמ"ח נתן לפי טווח תאריכים.

קוד ה – sql:

```
create or replace procedure DoubleBrancheAmout is
--vars
MaxOfLoan integer:=0;
TempMax integer:=0;
BrancheNum integer:=0;
BalanceNum integer:=0;
-- fund tbl cursor.
cursor aFund is
select *
from fund;
aIterator fund%rowtype;

-- begin
begin
  --for loop.
  for aIterator in aFund
    loop
      --call GetTotalAmount function.
      TempMax:=GetTotalAmount('11-JUN-2019',CURRENT_DATE,aIterator.branchcode);
      if(TempMax>MaxOfLoan) then
        --save the max of loan.
        MaxOfLoan:=TempMax;
        --save the branch code.
        BrancheNum:=aIterator.branchcode;
        --save the balance code.
        BalanceNum:=aIterator.balance;
      end if;
    end loop;
  --Update.
  if (BrancheNum <= 0) or (BrancheNum <=0) then
    dbms_output.put_line('ERROR not find branche num');
  else
    --Update query.
    dbms_output.put_line('The balance of Branchcode id: '||BrancheNum ||' before update is: '
|| BalanceNum);
    UPDATE fund
    SET balance=(BalanceNum*2)
    WHERE branchcode= BrancheNum;
    dbms_output.put_line('The balance of Branchcode id: '||BrancheNum ||' after update is: ' ||
BalanceNum*2);
  end if;
end DoubleBrancheAmout;
```

כאשר נפעיל את הפרוצדורה על ידי הקוד הבא:

```
declare
begin
  DoubleBrancheAmout;
end;
```

נקבל את הפתרון:

```
The balance of Branchcode id: 162 before update is: 233
The balance of Branchcode id: 162 after update is: 466
```

3.5 פונקציות

3.5.1 פיטורי מנהלים שסרחו

תיאור:

לאחר יצירת פונקציה מספר - 1, מנכ"ל העמותה קיבל לידו את דו"ח השנתי שהפונקציה הכינה עבורו. לתדהמתו מצא שיש גרעון בכספי הגמ"ח. לאחר בירור הגיעה שמועה לידי מנכ"ל הגמ"ח שישנה הונאה מצד כמה ממנהלי סניפים של הגמ"ח אשר אין להם הרשאה לקחת הלוואה (הרשאה ניתנת רק למנהל שיש לו Authorization=3) אך לקחו בכל זאת. מנכ"ל הגמ"ח מבקש לבנות פונקציה שתבדוק את הנ"ל, ולפטר לאלתר מנהלים שיתפסו, אם יתפסו.

הפעולה תתבצע כך, למנהל עם הרשאה 3 מותר לקחת הלוואה, מנהל עם הרשאה 2 יפוטר מלהיות מנהל והוא יעשה עובד רגיל, אך מנהל עם הרשאה 1 בלבד יפוטר לגמרי (כלומר גם אם הוא עובד הוא לא יישאר).

הפונקציה לא מקבל ארגומנטים מכיוון שעוברת על כל המנהלים, הפונקציה תבצע מחיקה מבסיס הנתונים כאשר תמצא מנהלים מושחתים, הפונקציה תחזיר את מספר המנהלים שנתפסו בשחיתות.

קוד ה – sql:

```
create or replace function DeleteManager
return integer is
--Vars
CountManager integer:=0;
-- cursor tbl manager.
cursor aManager is
select *
from manager
natural join loan
natural join clerk;
aIterator manager%rowtype;
-- cursor tbl loan.
cursor aLoan is
select *
from loan;
bIterator loan%rowtype;
-- begin
begin
  --external for manager.
  for aIterator in aManager
    loop
      if (aIterator.Authorization <> 3) then
        --internal for loan.
        for bIterator in aLoan
          loop
            if (bIterator.id = aIterator.id) then --find manager who takes loan.
              CountManager:=(CountManager+1);
              if (aIterator.Authorization = 2) then
                dbms_output.put_line('The manager id: ' || aIterator.id || ' holding a: ' ||
aIterator.Authorization || ' authorization');
```

```
            dbms_output.put_line('But he Takes Loan Number: ' || bIterator.loannumber || '
initial loan amount: ' || bIterator.initialloanamount);
            --' So he will be deleted from manager and wiil be clerk');
            --delete from manager
            DELETE FROM manager WHERE id=aIterator.id;
          end if;
          if (aIterator.Authorization = 1) then
            dbms_output.put_line('The manager id: ' || aIterator.id || ' holding a: ' ||
aIterator.Authorization || ' authorization');
            dbms_output.put_line('But he Takes Loan Number: ' || bIterator.loannumber || '
initial loan amount: ' || bIterator.initialloanamount);
            --' So he will be deleted from manager and clerk as well');
            --delete from manager and from clerk.
            DELETE FROM manager WHERE id=aIterator.id;
            DELETE FROM clerk WHERE id=aIterator.id;
          end if;
        end if;
      end loop;
    end if;
  end loop;

 --if  no manager that takes a loan
 if (CountManager = 0) then
   dbms_output.put_line('There are no manager who takes a loan');
 end if;
return CountManager;
end DeleteManager;
```

לפני שהוספתי לפונקציה את שאילתות מחיקה הרצתי ע"י הפקודה הבאה והתקבל:

```
declare
 --local vars.
 ResultFun integer:=0;
begin
 --call function.
 ResultFun:= DeleteManager;
 dbms_output.put_line('The number of manager that was deleted is: ' || ResultFun);
end;
```

```
The manager id: 167714899 holding a: 2 authorization
But he Takes Loan Number: 235 initial loan amount: 2554
The manager id: 167714899 holding a: 2 authorization
But he Takes Loan Number: 259 initial loan amount: 5289
The manager id: 232617841 holding a: 1 authorization
But he Takes Loan Number: 289 initial loan amount: 8510
The manager id: 433953550 holding a: 1 authorization
But he Takes Loan Number: 106 initial loan amount: 5511
The manager id: 433953550 holding a: 1 authorization
But he Takes Loan Number: 220 initial loan amount: 5760
The manager id: 433953550 holding a: 1 authorization
But he Takes Loan Number: 293 initial loan amount: 5349
The manager id: 618708413 holding a: 2 authorization
But he Takes Loan Number: 297 initial loan amount: 1868
The number of manager that was deleted is: 85
```

כעת נריץ את הפונקציה שוב כך ונקבל:

```
There are no manager who takes a loan
The number of manager that was deleted is: 0
```

ניתן לראות שכל ה85 המנהלים שנתפסו בשחיתות נמחקו (חלקם מלהיות מנהל וחלקם גם מלהיות עובדים).

3.5.2 סך סכום ההלוואות בתאריכים רצויים עבור כל סניף של הגמ"ח

תיאור:

בשעה טוב הגיע הזמן וגמ"ח הלוואת לב נהפך להיות עמותה, לשם כך רשם העמותות ביקש מעמותת לב לקבל דו"ח שנתי, (דו"ח אשר ישלח בסוף כל שנה) אשר יפרט מהם סכום ההלוואות שנתנו השנה הנוכחית לפי מספר כל סניף וסניף בבנק הלוואות, על מנת לראות פעילות שוטפת של העמותה.

מנהל הגמ"ח מבקש לבנות פונקציה שמקבלת כארגומנט תאריך התחלה ותאריך סיום וכן הפונקציה מקבלת מספר סניף, הפונקציה תחזיר סך כל ההלוואות שנתן הסניף בתאריכים הנדרשים.

קוד ה – sql:

```
create or replace function GetTotalAmount(FromDate in date, ToDate in date, BranchNum in
integer)
return integer is
 --Vars
SumBranchBank integer:=0;
branchFlage Boolean:=false;
-- loan tbl cursor.
Cursor aLoan is
select *
from loan;
aIterator loan%rowtype;
-- begin
begin
  --for Loan.
  For aIterator in aLoan
    loop
      if (BranchNum=aIterator.branchcode) then
        branchFlage:=true;
        if (aIterator.dateTaken between FromDate and ToDate) then
          SumBranchBank:=(SumBranchBank+aIterator.initialloanamount); --
sum+=initialloanamount;
          end if;
        end if;
  end loop;
  --print
  if (branchFlage = false) then
    dbms_output.put_line('There is no valid bank branch '|| BranchNum || '!');
  --else
  end if;
return SumBranchBank;
end GetTotalAmount ;
```

כעת נריץ את הפונקציה עם הארגומנטים הבאים:

FromDate= '11-JUN-1992'

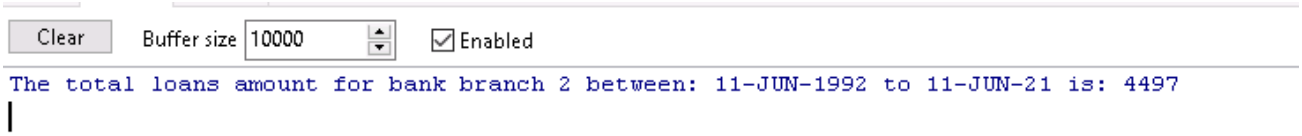
ToDate= CURRENT_DATE

BranchNum=2

קוד ההרצה:

```
declare
  --local vars.
  ResultFun integer:=0;
begin
  --call function.
  ResultFun:= GetTotalAmount('11-JUN-1992',CURRENT_DATE,2);
  dbms_output.put_line('The total loans amount for bank branch ' || 2 || ' between: '
   || '11-JUN-1992' || ' to ' || CURRENT_DATE || ' is: ' || ResultFun);
  --dbms_output.put_line(CURRENT_DATE);
end;
```

נקבל את התוצאה הבאה:



נ.ב אם המשתמש יזין סניף בנק שלא קיים כדוגמא 0 אזי יקבל את ההודעה הבאה:

3.6 טריגרים

3.6.1 עדכון מאזן הסניף מיד לאחר ביצוע הלוואה חדשה

בכדי לשמור על אמינות הנתונים וקלות השימוש במערכת, ניצור טריגר חדש אשר יקטין את מאזן הסניף באופן אוטומטי לאחר שמושכים ממנה כסף להלוואה.

קוד ה – SQL:

```
create or replace trigger ModifyFundBalance
 after insert
 on loan
 for each row
declare
pragma autonomous_transaction;
 -- local vars
LoanAmountNum integer:=0;
BranchCodeNum integer:=0;
flag boolean :=true;
amount integer:=0;

-- fund tbl cursor.
cursor aFund is
select *
from fund;
aIterator fund%rowtype;

--begin
begin

  --for loop.
  amount:= :new.initialloanamount;
  for aIterator in aFund
    loop
      if ((aIterator.branchcode = :new.branchcode) and (flag = true)) then
         LoanAmountNum:=aIterator.balance;
         BranchCodeNum:=aIterator.branchcode;
         flag:=false;

      end if;
    end loop;
 --insert
  if inserting then
    --if (flag = false) then
    --enf if;

    UPDATE fund
    SET balance=balance-amount
    WHERE branchcode= BranchCodeNum;
    dbms_output.put_line('After insert into new loan number: ' || BranchCodeNum);
    dbms_output.put_line('We will update the balance minus: ' || LoanAmountNum);
    dbms_output.put_line('The transaction completed successfully!');
   end if;
```

```
commit;
end ModifyFundBalance;
```

כעת נכניס את פקודת הכנסת לווה חדש:

```
Insert into SHIMIZRA.LOAN values(1607,to_date('13-06-2021','dd-mm-yyyy'),120,10,2,323033804)
```

ניתן לראות שמדובר בלווה שלווה מסניף מספר 2.

כאמור ניתן לראות שלפני שיופעל הטריגר מאזן סניף בנק מספר 2 הינו 1179.

SQL | Output | Statistics

```
select *
from fund
where branchcode = 2;
```

	BALANCE	ADDRESS	NAME	BRANCHCODE
1	1179	, Peachtree City	AC Technologies	2

לאחר הפעלת הטריגר נקבל:

```
After insert into new loan number: 2
We will update the balance minus: 120
The transaction completed successfully!
```

נבדוק כעת האם יש שינויים בבסיס הנתונים:

```
select *
from fund
where branchcode = 2;
```

	BALANCE	ADDRESS	NAME	BRANCHCODE
1	1059	, Peachtree City	AC Technologies	2

ניתן לראות שירדו 120 ₪ ממאזן הסניף ולכן פעולה בוצעה בהצלחה.

3.6.2 הוספת עובד בעל הרשאות מרובות לטבלת המנהלים

בכדי למנוע מצב אבסורדי בה יש מנהלים עם הרשאות נמוכות ועובדים זוטרים עם הרשאות מרובות, נחליט שמעתה והלאה כל פעם שעובד זוטר יקבל את רמת ההרשאות הגבוהה ביותר (3) הוא ימונה כמנהל עם בונוס למשכורת של עשרה אחוזים.

קוד ה – SQL:

```
create or replace trigger WorkerAfterwardsManager
after update
 on clerk
 for each row
declare
pragma autonomous_transaction;
 -- local vars
WorkerId integer;
WorkerAuthorization integer;
--begin
begin
  WorkerId:= :new.id;
  WorkerAuthorization:= :new.authorization;
  if WorkerAuthorization>=3 then
    insert into manager values(10,WorkerId);
    dbms_output.put_line(WorkerId ||' will become manager because the worker has high
authorization');
    dbms_output.put_line('Good for him/her!');
  end if;
commit;
end WorkerAfterwardsManager;
```

קלט:

פלט:

```
Clear   Buffer size 10000   Enabled
774376884 will become manager because the worker has high authorization
Good for him/her!
3:1   shimizra@labdbwin   1 row updated in 0.007 seconds
```

נבדוק שהטריגר באמת עובד:

```
select * from manager where id=774376884
```

	SALARYBONUSPERCENTAGE	ID
1	10	774376884

1:41 shimizra@labdbwin 1 row selected in 0.015 seconds

הוא מונה בהצלחה כמנהל.

3.7 סיכום

זו הפעם הראשונה שבה היה עלינו להקים בסיס נתונים רחב משלב התכנון עד לשלב האוטומציה עם פונקציות וטריגרים. השתפרנו מאוד בסמסטר הזה כמובן בשפת ה – SQL אבל יתרה מכך קיבלנו מהקורס יצירתיות, עבודה נכונה בצוות ויכולת טובה יותר לקחת דרישות ולעבד אותן למוצר מוגמר.

נרצה להודות לה' שנתן לנו את היכולת לעשות פרויקט זה, וגם למנחה הפרויקט מר אריה ויזן שתמיד דחף אותנו לעשות את העבודה הטובה ביותר שאנו יכולים.