

נושא: חנות רמי לוי

מגישות: רבקה לוי

חיה יזרוסק'

הרעון הכללי:

המחשה של היחסות והקשרים בסופר מarket כמו רמי לוי.

היחסות השונות:

Employee המפתח הוא empid

Buy המפתח הוא bid

Customer המפתח הוא custid

Product המפתח הוא prodid

ProdType המפתח הוא prodtypeid

CashRegister המפתח הוא cashregid

קשרים עיקריים:

הקשר BUYSWHAT הוא קשר בין מוצר לקנייה, כך שבכל קנייה אפשר לקנות מספר מוצרים. וכל קנייה נחברת לפחות בקשר WHOBUYS כך שלכל קנייה יהיה לפחות אחד. וכך נוכל לחבר בכל קנייה כמה מוצרים לאוותן.

כל קנייה מתרכשת בcashregister אחד, והם מקושרים בקשר BUYSBY.

הקשר ORGANIZE מקשר עובד לשסידר אותו.

הקשר TYPE מקשר בין מספר של סוג מוצר לשם סוג המוצר, נניח 1 זה תפו ווכא.

סוגי קשרים:

יחיד לרבים:

Whobuys

Buysby

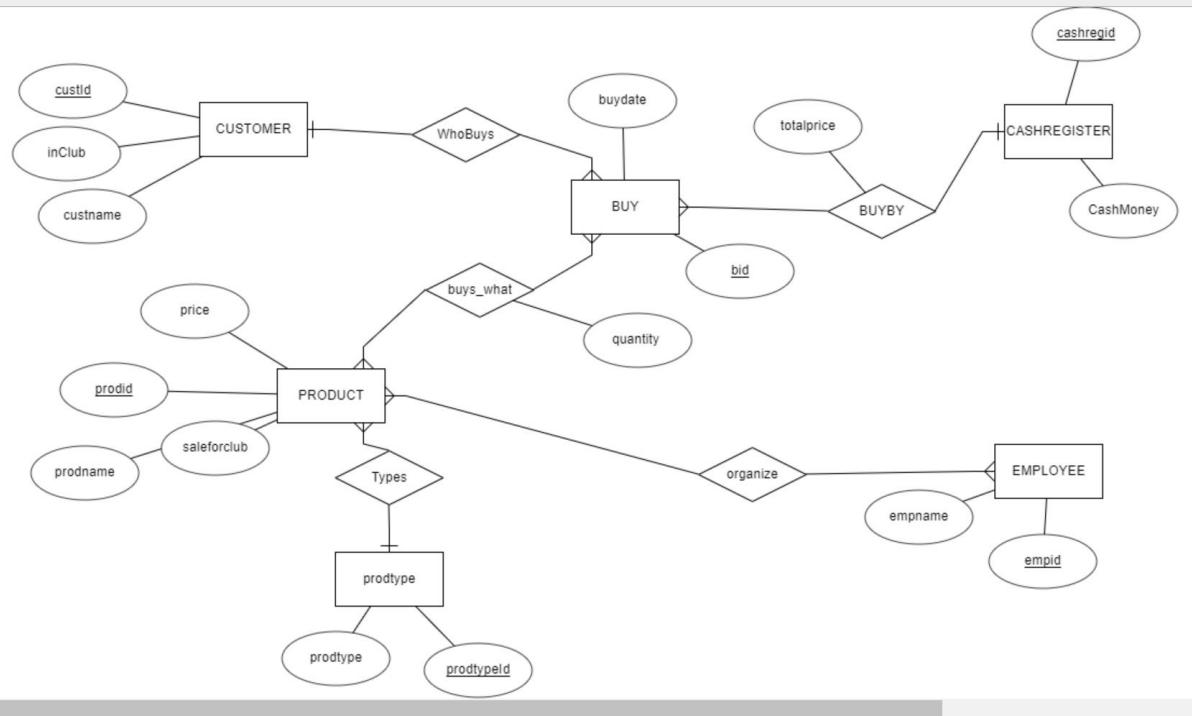
Types

רבים לרבים:

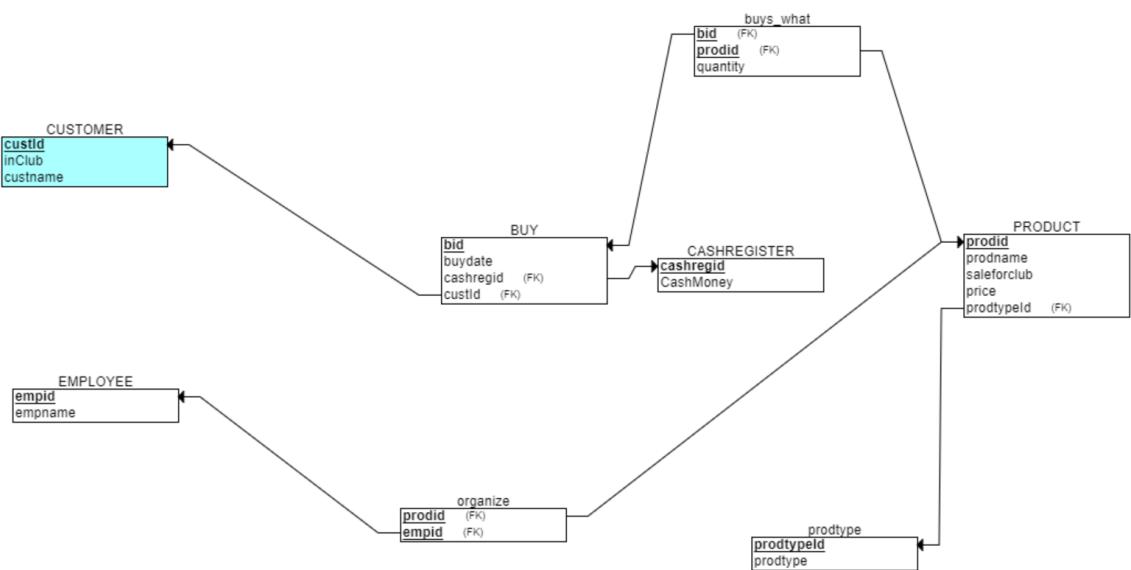
Organize

Buyswhat

ציור ERD



ציור ה DSD



SQL נטול

```

CREATE TABLE CASHREGISTER
(
    cashregid NUMERIC(3) NOT NULL,
    CashMoney NUMERIC(3) NOT NULL,
    PRIMARY KEY (cashregid)
);

CREATE TABLE CUSTOMER
(
    custId NUMERIC(3) NOT NULL,
    inClub VARCHAR(15) NOT NULL,
    custname VARCHAR(15) NOT NULL,
    PRIMARY KEY (custId)
);

CREATE TABLE EMPLOYEE
(
    empid NUMERIC(3) NOT NULL,
    empname VARCHAR(15) NOT NULL,
    PRIMARY KEY (empid)
);

CREATE TABLE prodtype
(
    prodtype VARCHAR(15) NOT NULL,
    prodtypeld NUMERIC(3) NOT NULL,
    PRIMARY KEY (prodtypeld)
);

CREATE TABLE BUY
(
    buydate DATE NOT NULL,
    bid NUMERIC(3) NOT NULL,
    cashregid NUMERIC(3) NOT NULL,
    custId NUMERIC(3) NOT NULL,
    PRIMARY KEY (bid),
    FOREIGN KEY (cashregid) REFERENCES CASHREGISTER(cashregid),
    FOREIGN KEY (custId) REFERENCES CUSTOMER(custId)
);

CREATE TABLE PRODUCT
(
    prodid NUMERIC(3) NOT NULL,
    prodname VARCHAR(15) NOT NULL,
    saleforclub VARCHAR(15) NOT NULL,
    price NUMERIC(3) NOT NULL,
    prodtypeld NUMERIC(3) NOT NULL,
    PRIMARY KEY (prodid),
    FOREIGN KEY (prodtypeld) REFERENCES prodtype(prodtypeld)
);

CREATE TABLE buys_what
(
    quantity_ NUMERIC(3) NOT NULL,
    bid NUMERIC(3) NOT NULL,

```

```

    prodid NUMERIC(3) NOT NULL,
    PRIMARY KEY (bid, prodid),
    FOREIGN KEY (bid) REFERENCES BUY(bid),
    FOREIGN KEY (prodid) REFERENCES PRODUCT(prodid)
);

```

CREATE TABLE organize

(

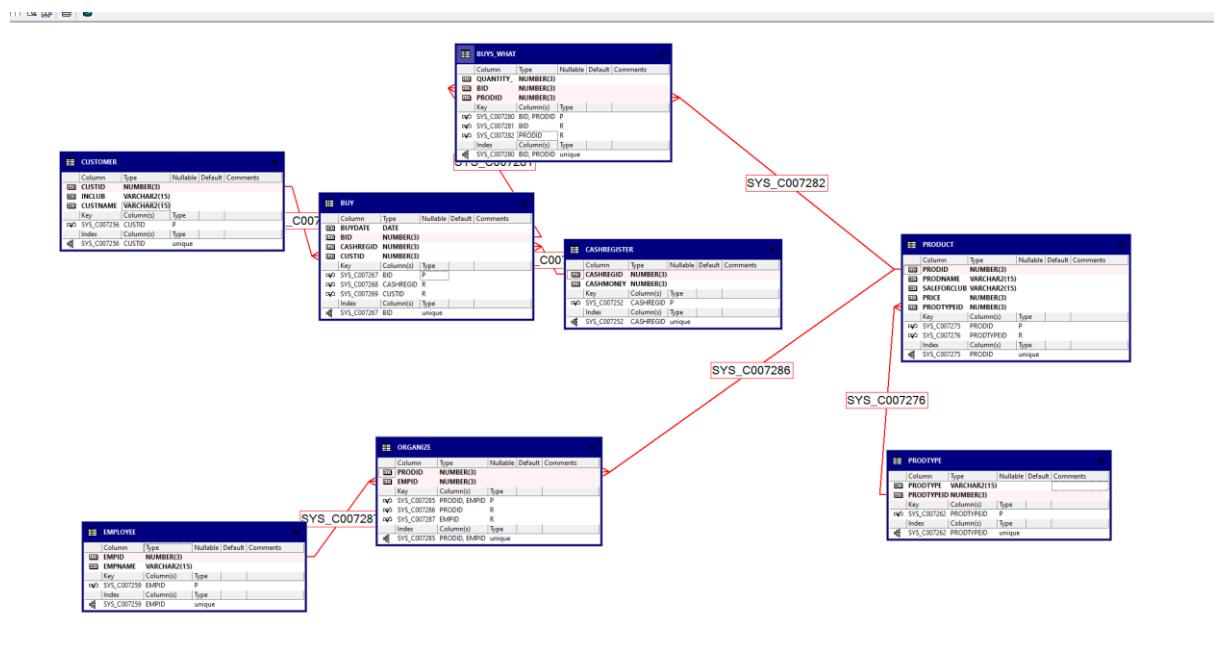
```

    prodid NUMERIC(3) NOT NULL,
    empid NUMERIC(3) NOT NULL,
    PRIMARY KEY (prodid, empid),
    FOREIGN KEY (prodid) REFERENCES PRODUCT(prodid),
    FOREIGN KEY (empid) REFERENCES EMPLOYEE(empid)

```

);

ולסימן - הדיאגרמה שנוצרת מה sqs שakan נראה כמו ה DSDM



INSERT וUPDATE מ

Select employee Select cashregister Select prodtype

EMPID | EMPNAME

1	31 chaya rose
2	32 eli cohen
3	33 shoshi levi

```
select * from EMPLOYEE;
select * from CASHREGISTER;
select * from PRODTYPE;
```

Select employee Select cashregister Select prodtype

CASHREGID | CASHMONEY

1	1
2	2
3	0

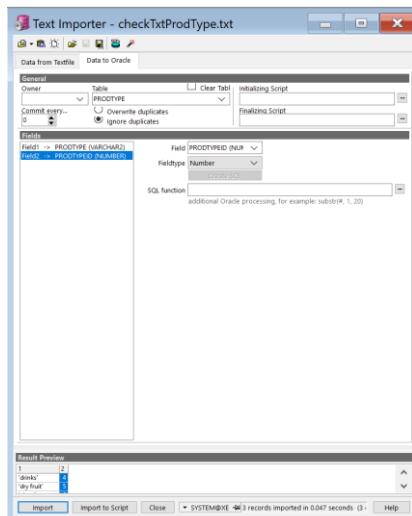
```
select * from EMPLOYEE;
select * from CASHREGISTER;
select * from PRODTYPE;
```

Select employee Select cashregister Select prodtype

PRODTYPE | PRODTYPEID

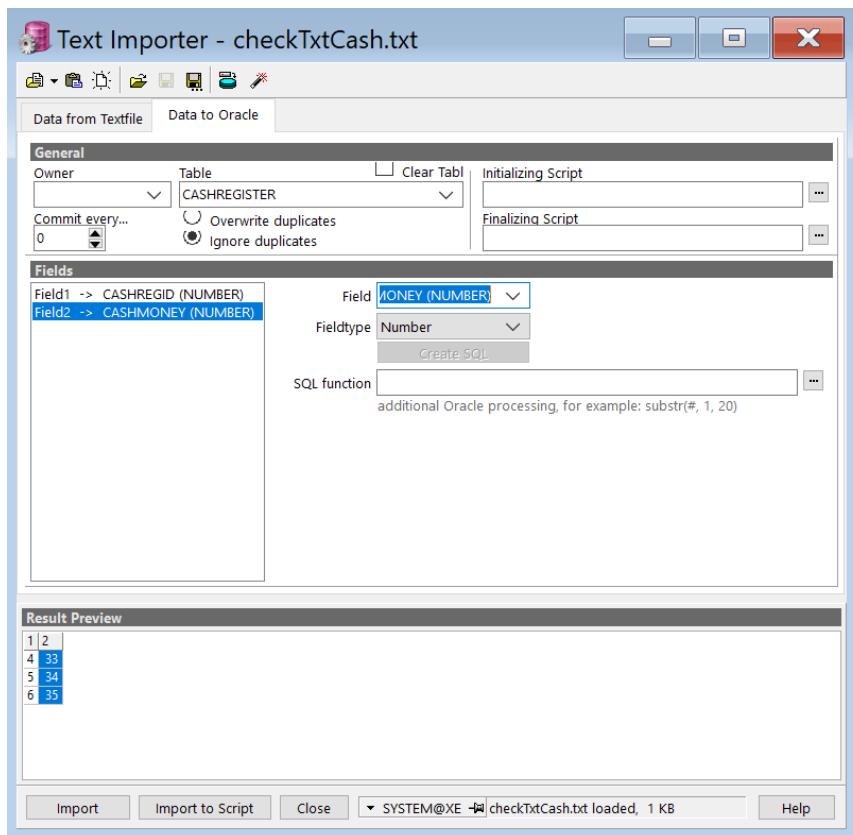
vegetable	1
meats	2
cheese	3

מ הוראות TXT



The screenshot shows the Oracle Database SQL Developer interface with three tabs: "Select employee", "Select cashregister", and "Select prodtype". The "Select prodtype" tab is active and displays a table with the following data:

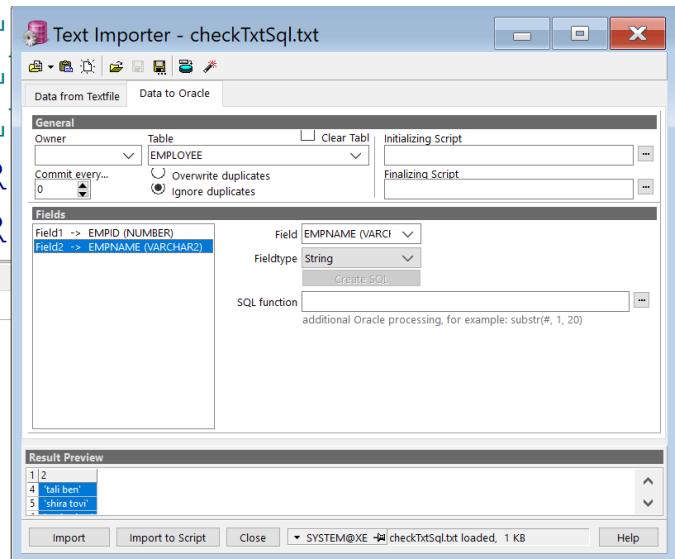
	PRODTYPE	PRODTYPEID
▶	1 vegetable	1
	2 meats	2
	3 cheese	3
	4 'drinks'	4
	5 'dry fruit'	5
	6 'cleaning mat'	6



Select employee Select cashregister Select prodtype

[grid icon] [refresh icon] [add icon] [edit icon] [checkmark icon] [down arrow icon] [up arrow icon] [list icon] [grid icon] [refresh icon] [add icon] [edit icon] [checkmark icon] [down arrow icon] [up arrow icon] [list icon]

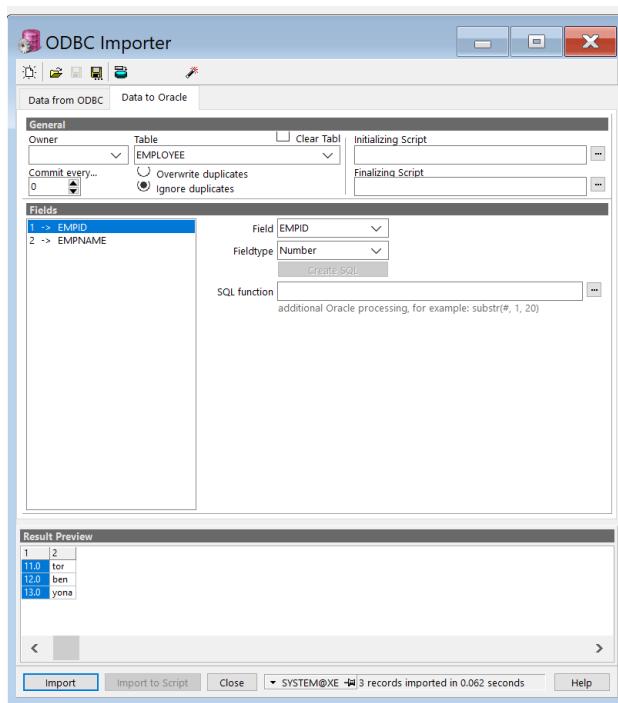
	CASHREGID	CASHMONEY
▶	1	20
	2	10
	3	0
	4	33
	5	34
	6	35



The screenshot shows the Oracle SQL Developer Data Pump Importer interface. The 'General' tab is selected, showing the 'Owner' as SYSTEM, 'Table' as EMPLOYEE, and 'Commit every...' set to 0. Under 'Fields', 'Field1' is mapped to 'EMPID (NUMBER)' and 'Field2' is mapped to 'EMPNAME (VARCHAR2)'. The 'Fieldtype' is set to 'String'. In the 'Result Preview' section, the data is shown in two rows:

	EMPID	EMPNAME
1	31	chaya rose
2	32	eli cohen
3	33	shoshi levi
4	4	'tali ben'
5	5	'shira tovi'
6	6	'rani yahav'

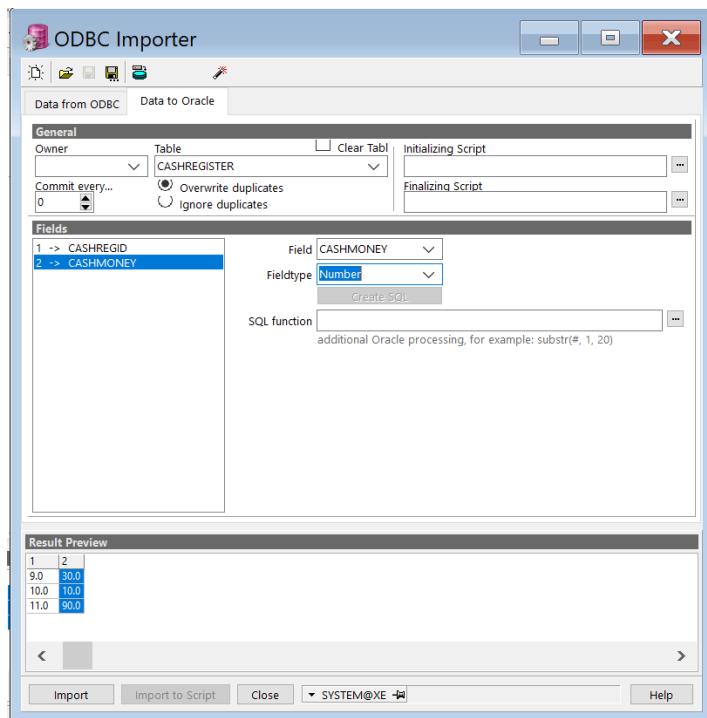
הבא מארקן



```
select * from EMPLOYEE;
select * from CASHREGISTER;
select * from PRODTYPE;
```

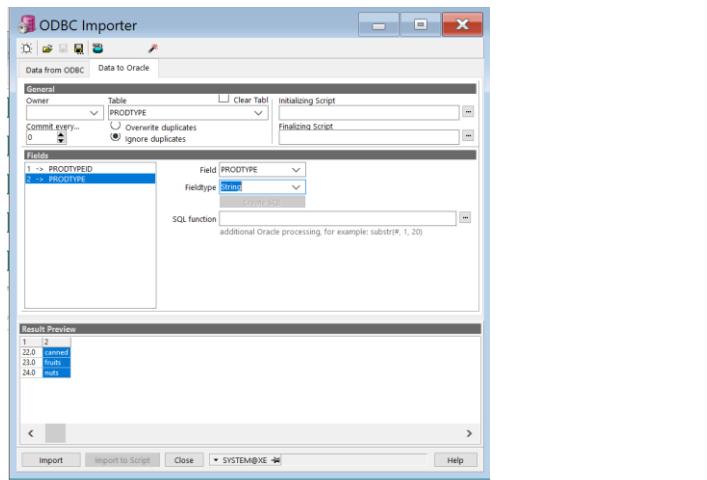
The screenshot shows a database application interface with three tabs at the top: 'Select employee', 'Select cashregister', and 'Select prodtype'. Below the tabs is a toolbar with various icons. The main area displays a table with columns 'EMPID' and 'EMPNAME'. The data is as follows:

	EMPID	EMPNAME
▶	1	31 chaya rose
2	32 eli cohen	
3	33 shoshi levi	
4	4 'tali ben'	
5	5 'shira tovi'	
6	6 'rani yahav'	
7	11 tor	
8	12 ben	
9	13 yona	



```
select * from EMPLOYEE;
select * from CASHREGISTER
select * from PRODTYPE;
```

Select employee		Select cashregister		Select prodtype	
1	CASHREGID	2	CASHMONEY	3	
1	1	2	20	4	
2	2	3	10	5	
3	3	4	0	6	
4	4	5	33	7	
5	5	6	34	8	
6	6	7	35	9	
7	9	8	30	10	
8	10	9	10	11	
9	11		90		



```

select * from EMPLOYEE;
select * from CASHREGISTER;
select * from PRODTYPE;

```

PRODTYPE	PRODTYPEID
vegetable	1
meats	2
cheese	3
bread	4
dry fruit	5
cleaning mat	6
flour	22
fruits	23
nuts	24

מחיקת טבלאות:

```
DROP TABLE organize;
```

```
DROP TABLE buys_what;
```

```
DROP TABLE PRODUCT;
```

```
DROP TABLE BUY;
```

```
DROP TABLE protype;
```

```
DROP TABLE EMPLOYEE;
```

```
DROP TABLE CASHREGISTER;
```

```
DROP TABLE CUSTOMER;
```

הניצור

Name	Type	Size	Data
EMPID	NUMBER	3	Sequence(50, [2])
EMPNAME	VARCHAR2	15	FirstName + ' ' + LastName
*			

את הקוד להוספה נשמר בנפרד כי הוא ארוך מדי להכניס פה

CASHREGISTER

Owner	Table	Number of records
SYSTEM	CASHREGISTER	300

Name	Type	Size	Data
CASHREGID	NUMBER	3	Sequence(20, [1])
CASHMONEY	NUMBER	3	Random(0, 999)
*			

CUSTOMER

Owner	Table	Number of records
SYSTEM	CUSTOMER	1000

Name	Type	Size	Data
CUSTID	NUMBER	3	Sequence(0, [1])
INCLUB	VARCHAR2	15	List('YES', 'NO')
CUSTNAME	VARCHAR2	15	FirstName + ' ' + LastName
*			

Data Generator - New

PRODTYPE

Owner	Table	Number of records
SYSTEM	PRODTYPE	300

Name	Type	Size	Data
PRODTYPE	VARCHAR2	15	[aaaaaaaaaa]
PRODTYPEID	NUMBER	3	Sequence(50,1)
*			

Data Generator - New

PRODUCT

Name	Type	Size	Data
PRODID	NUMBER	3	Sequence(0)
PRODNAME	VARCHAR2	15	[Aaaaaaaaa]
SALEFORCLUB	NUMBER	3	Random(10,99)
PRICE	NUMBER	3	Random(0, 999)
PRODTYPEID	NUMBER	3	List(select prodtypeid from prodtype)
*			

Data Generator - New

BUY

Name	Type	Size	Data
BUYDATE	DATE		Random(01-01-1970, 01-05-2023)
BID	NUMBER	2	Random(50, 1050)
CASHREGID	NUMBER	3	List(select cashregid from cashregister)
CUSTID	NUMBER	3	List(select custid from customer)
*			

Data Generator - New

Date format: dd-mm-yyyy

Commit every...: 0

Delay (mSec): 0

User Definitions Initialization Script

BUYS_WHAT

Name	Type	Size	Data
QUANTITY_	NUMBER	3	Random(0,999)
BID	NUMBER	3	List(select bid from buy)
► PRODID	NUMBER	3	List(select prodid from product)
*			

ORGANIZE

Name	Type	Size	Data
► PRODID	NUMBER	3	List(select prodid from product)
► EMPID	NUMBER	3	List(select empid from employee)
*			

חלק רביעי שאלות:

Select:

-1

בחר את כל הטיפוסים שיש מעל שיש מוצרים כאלה.

select prodtype,prodtypeid

from prodtype natural join product

group by prodtype,prodtypeid

having count(*)>6;

The screenshot shows a Microsoft Access query results grid. The columns are labeled 'PRODTYPE' and 'PRODTYPEID'. The data consists of nine rows, each containing a product name and its corresponding ID. Row 2 ('Xzqyqpcjlr') is highlighted in green.

	PRODTYPE	PRODTYPEID
▶ 1	Lgvdyhkpu	68
2	Xzqyqpcjlr	229
3	Lkqssbwvbtb	334
4	Lelwuyjdnez	211
5	Gebbmjcojz	77
6	Untutcltzn	146
7	Vywecoatww	157
8	Ldtpdlvhfyf	190
9	Hhbvesifod	279

2- בחר את העובדים שסידרו הכי הרבה מוצרים:

```
select empid
from organize
group by empid
having count(prodid)=(select max(A.cc)
from (select empid,count(*) cc
from organize
group by empid) A)
```

הכחול סופר לכל עובד כמה הוא סידר הkey תחתון מוצאת את המקס מביניהם.
ואז החיצוני שוב סופר לכל עובד כמה הוא סידר ובוחר את אלה ששו למקס

The screenshot shows a Microsoft Access query results grid. The column is labeled 'EMPID'. The data consists of two rows, each containing an employee ID. Row 2 ('275') is highlighted in green.

	EMPID
▶ 1	239
2	275

-3

בחר את הסכוםסהכ של קניה מס' 3.(לא כולל הנחות)

```
select sum(
price*quantity_
)
from buy
natural join buys_what
natural join product
```

where bid=3

קודם בחרנו את הקניות מסע buy_what את אלה שישיות לקניה 3. תשיג את המחיר מהנטורל ג'ין עם פרודקט ואת הכמות מסע buy_what

	SUM(PRICE*QUANTITY_)
► 1	1240516

4-

בחר את הרווח מהמוצרים שנකנו בתאריך המופיעים: (לא כולל הנחות) '08-11-2024'

```
select sum(price)
from buys_what
natural join buy
natural join product
where buydate= to_date('08-11-2024','dd-mm-yyyy')
```

בוחרים את הקניות שעשו שבתאריך הנל ומקשרים לכל המוצרים מהתאריך ואז סוכמים את המחירים שקיבלנו自然 JOIN עם פרודקט.

	SUM(PRICE)
► 1	289

update

-1

כל מי שהקופה עם פחות מ70 שח תוסיף שיהיה 100 שח (שינו חלק לכך בהרצה אין הכל)

	CASHREGID	CASHMONEY
▶	1	24
	2	70
	3	62

```
select *
from cashregister
where cashmoney<=70
```

```
UPDATE cashregister
SET cashmoney = 100
WHERE cashmoney <= 70;
```

	CASHREGID	CASHMONEY
▶	1	100

2- תחלק ל 2 את המחיר של המוצר ששמו 'Gbiobezyi'

```
UPDATE product
SET price = price/2
WHERE prodname='Gbiobezyi';
```

	PRICE	
▶	1	395

לפני

אחרי:

	PRICE
1	198

delete

1- תמחק את כל מי שגם עובד וגם לקוח

delete from employee where empid in

(select empid from customer

natural join employee

where custname=empname(

	EMPID
1	841

	EMPID	EMPNAME
1	841	Lorraine Madsen

הפנימי מושג את ה ID של אנשים שהשם שלהם נמצא גם בCUSTOMER וגם EMPLOYEE. החיצוני מוחק מי שזה ה ID שלו מתוך EMPLOYEE.

2-

מחק את כל הקניות שגם המוצרים מתחילה ב T ו גם ה bid הוא 100 כי לאוינו קניה יש כמה מוצרים.

delete from buys_what

where bid='100' and prodid in

(select prodid from product

where prodname like 'T("%

הפנימי- מושג את ה ID של מוצרים שמתחילה ב T. החיצוני מוחק מי שה ID שלו ברשימה הנל וגם נקינה בקניה buyswhat. (משיגים מי נקינה באיזה קניה מה buyswhat)

לפנוי- כshawishim select במקומ delete

	QUANTITY	BID	PROID
►	1	883	100

אחרי

	QUANTITY	BID	PROID
►	1	883	453

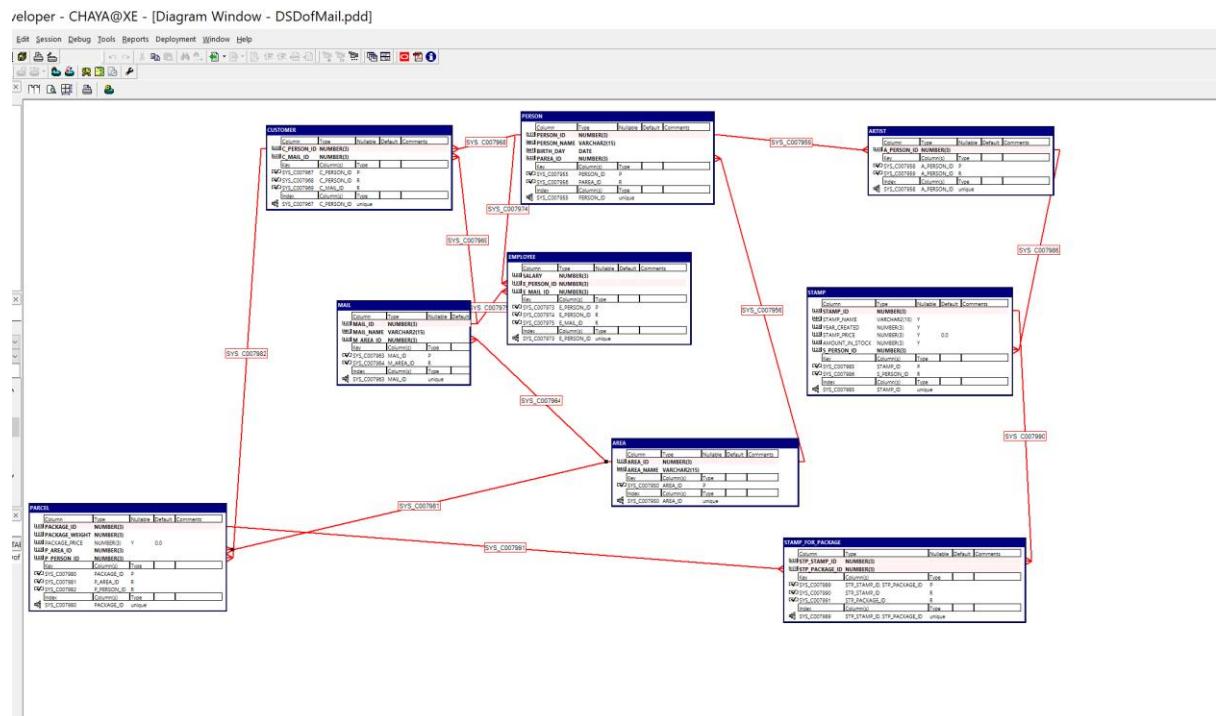
חלק 5- אינטגרציה

קיבלונו טבלאות סכמה של מיילים של חברות דואר.

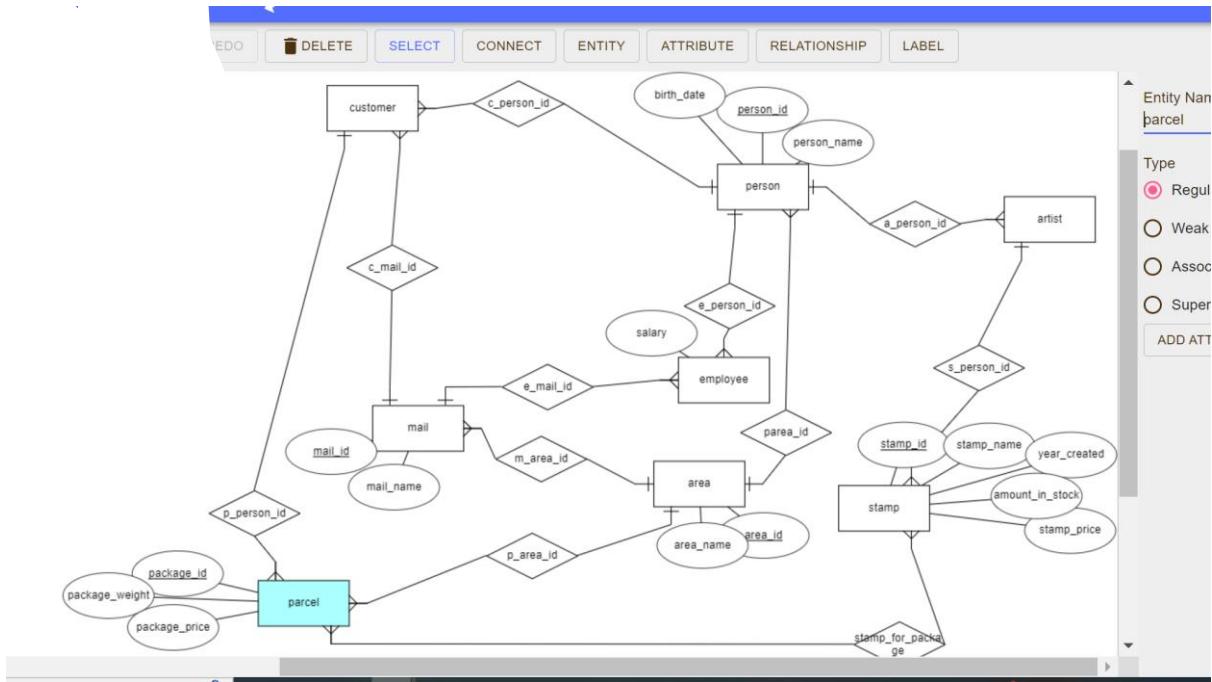
AREA- אזור

MAIL- כתובות דואר.

ה psd שנוצר מה create table שלם



ציור ה erd שלם



עכשו צריך לחבר בerd

עשינו שתי בחירות

- 1 על employee
- 2 על customer

נסביר את התהילה -

ນצטריך קודם להוסיף את כל מי שב customer ו employee שלא נמצא בתוך person.

```
insert into person (select custid, custname,null,null
```

```
from customer1
```

```
where custid not in (select person_id from person)
```

```
)
```

לפנינו אין צורך לשנות את העמודות החסרות בcustomer ו employee בcustomer ו employee שלא נמצאות ב person. אפשר ב person להכניס null

```
alter table
```

```
person
```

```
modify
```

```
) parea_id NULL;()
```

ורק לאחר מכן לחבר בין ה customer ו employee לשלהם וכו'.

נצרך להוסיף את כל מי שב customer שלהם לתוכו שלנו.

```
insert into customer1 (
    select c_person_id,'NO',null from customer where c_person_id not in (select custid from customer1)
```

(
כמובן שצרכי לאפשר לשמם של לקוחותosos נמחק את העמודה בהמשך)

נשנה מפתחות זרים:

כל מי שלהם קשור אליו עכשו שולמו קשור אליו-

מוחקים תלויות ב customer - אלה mail+person

```
--person foreign
--alter table customer drop constraint SYS_C008093
--mail foreign
--alter table customer drop constraint SYS_C008094
```

ומוסיף את התלויות הנל בcustomer1

Alter table customer1

```
add constraint SYS_C008094 foreign key (c_mail_id) references mail(mail_id);
```

alter table customer1

```
add constraint SYS_C008093 foreign key (custid) references person(person_id);
```

```
select* from person;
```

ושכל מי קשור לשלהם שייהי קשור לשלהנו-

מוחקים תלויות במילוי(parcel-customer

– customer1 – מוסיף את התלויות מאותו אובייקט אל

alter table parcel drop constraint SYS_C008107

alter table parcel

```
add constraint SYS_C008107 foreign key (p_person_id) references customer1(custid);
```

```
select * from customer1;
```

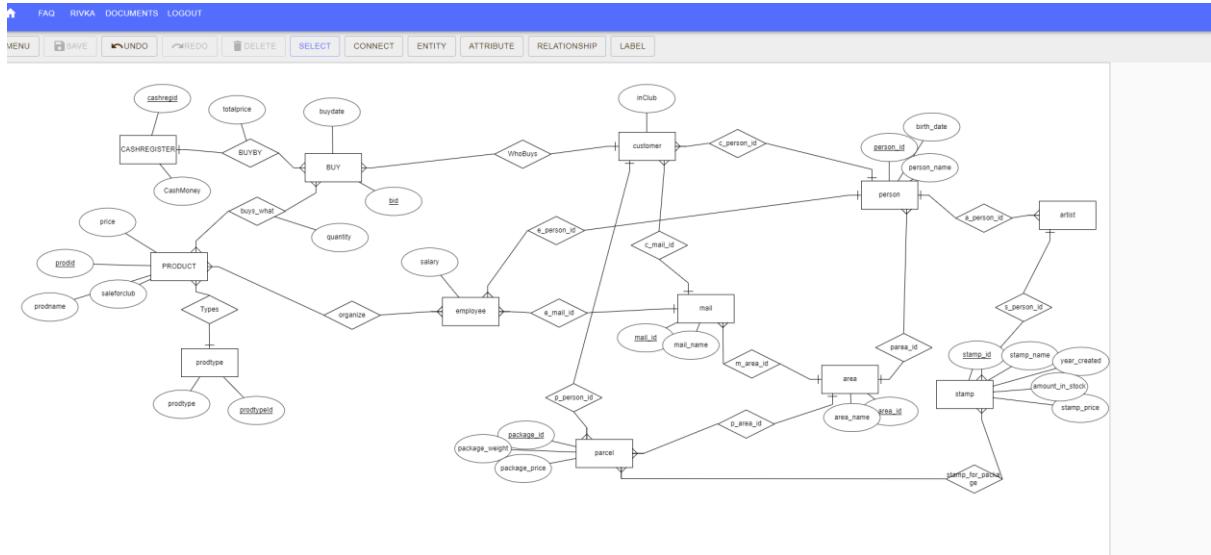
ומחקנו את העמודות המיותרות בcustomer1 ושיםנו את השם של customer

drop table customer

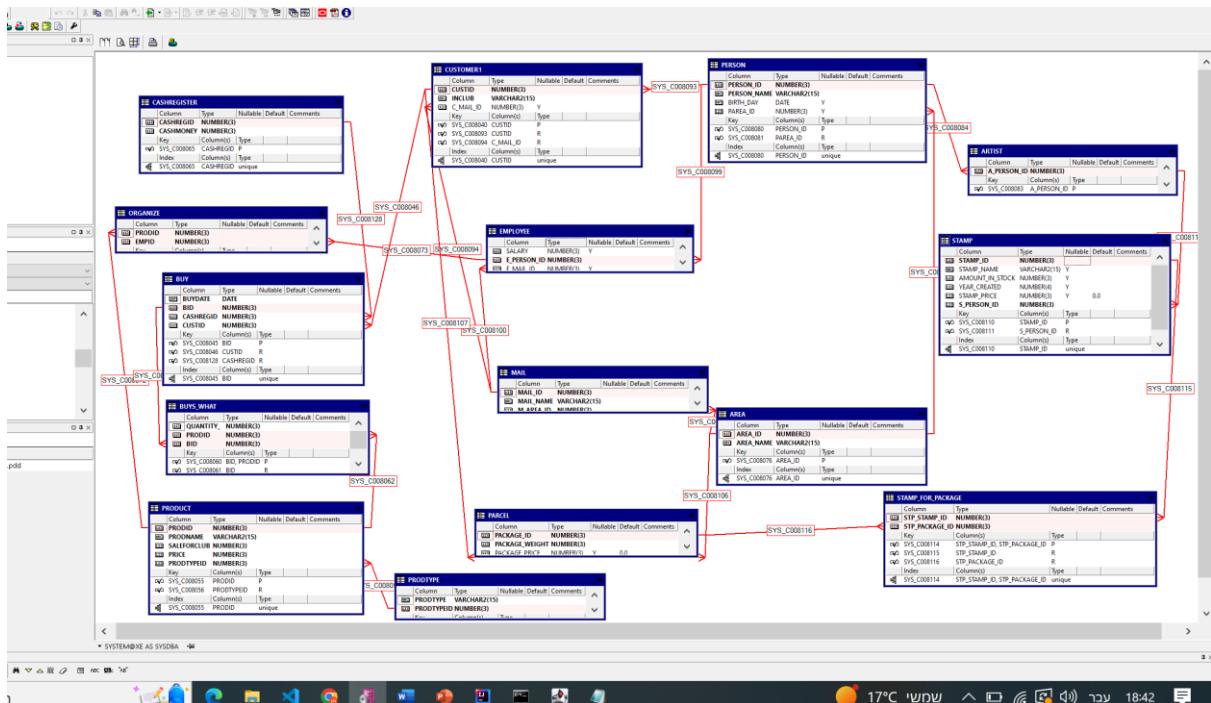
;rename customer1 to customer

אותו רעיון ביצנו בשפה החלטנו למחוק את ה `customer` שלו מכיוון שיש בו פחות חלקים. את המשכורת של מה ששלמו הכנסנו כתהן את היכולת להכניס ולחשין ידנית.

ה `erd` המחבר:



ה `psd` המחבר



הקשר היחיד שנמחק היה עם `cashregister`

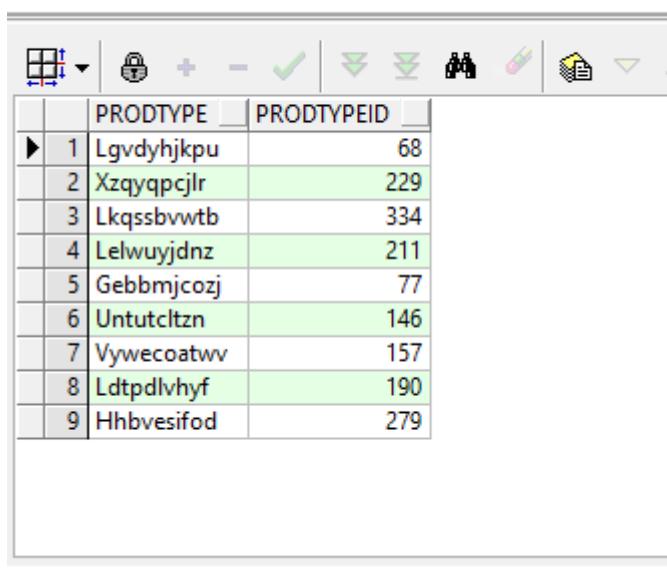
נבדוק שהשאילות עובדות:

Select:

1-

```
select prodtype,prodtypeid
from prodtype natural join product
group by prodtype,prodtypeid
having count(*)>6;
```

natural join



	PRODTYPE	PRODTYPEID
► 1	Lgvdyhjkpu	68
2	Xzqyqpcjlr	229
3	Lkqssbvwtb	334
4	Lelwuyjdzn	211
5	Gebbmjcozj	77
6	Untutcltn	146
7	Vywecoatvv	157
8	Ldtpdlvhyf	190
9	Hhbvesifod	279

עובד מעולה כי אף אחד מהן לא השתנה

2-

```
select sum(
price*quantity_
)
from buy
natural join buys_what
natural join product
where bid=3
כנ"ל:
```

	SUM(PRICE*QUANTITY_)
▶ 1	1240516

3-

בשאילתת הבאה אומנם הוספנו `to_date('08-11-2024','dd-mm-yyyy')` אבל כל העובדים החדשים לא סידרו מוצרים אז התוצאה נשארה זהה. רצינו את העובד שסידר כי הרבה מוצרים.

```
select empid
from organize
group by empid
having count(prodid)=(select max(A.cc)
from (select empid,count(*) cc
from organize
group by empid) A(
```

	EMPID
▶ 1	239
2	275

4-

גם בשאילתת הבאה הכל אותו דבר, כי לא שינו את הטבלאות הנ"ל

```
select sum(price)
from buys_what
natural join buy
natural join product
where buydate= to_date('08-11-2024','dd-mm-yyyy')
```

	SUM(PRICE)
▶ 1	289

עכשו הגיעו לחלק היותר מסובך:

Update:

UPDATE cashregister

SET cashmoney = 100

WHERE cashmoney <= 70;

עובד כמו שציריך כי cashregister לא תלוי באף אחד

נצלם תמונה של מה קורה אחרי כשמנסים לבחור קופות עם פחותות כסף

```
select *
from cashregister
where cashmoney<=70
```

CASHREGID	CASHMONEY
	100

UPDATE product

SET price = price/2

WHERE prodname='Gbiobezi';

גם כן עבדת כמו שציריך כי לא השנתה טבלת product

תמונה לפניו:

```
/*help for the querie above */
select price from product
where prodname='Gbiobezyi';
```

PRICE
395

תמונה אחריה:

```
UPDATE product
SET price = price/2
WHERE prodname='Gbiobezyi';
/*help for the querie above */
select price from product
where prodname='Gbiobezyi';
```

```
select * from product
```

PRICE
198

וכעת למסובך ביותר:

Delete:

```
delete from buys_what
where bid='100' and prodid in
(select prodid from product
```

where prodname like 'T("%

לפניכם:

```
select * from buys_what
where bid='100' and prodid in
(select prodid from product
where prodname like 'T%')
```

*/*help for later*/*

```
select empid from customer
natural join employee
where custname=empname
```

A screenshot of a database management system interface. At the top, there is a toolbar with various icons. Below the toolbar is a table with three columns: QUANTITY, BID, and PRODID. The table has one row with the values 1, 883, and 453 respectively.

	QUANTITY	BID	PRODID
▶ 1	883	100	453

אחרין:

```

select * from buys_what
where bid='100' and prodid in
(select prodid from product
where prodname like 'T%')

```

QUANTITY	BID	PRODID
10	100	100

כלומר השאלה זו עבדה בסדר כי אף אחד לא תלוי בשורה זו בתור מפתח זר. בכלל לא תלויים בטבלה זו בשל מפתח זר. היא תלוי באחרים, זה כן.

אפשר לראות שגם הערכים בתוך הטבלה זו לא השתנו. (יתכן שם הלקוח של הקנייה השתנה כי חיברנו כמה ליקוחות משנה הטבלאות עם אול' אותו זו אבל לא קרייטי כרגע) ולשאילתת האחרונה והכי מסובכת:

```

delete from employee where empid in
(select empid from customer
natural join employee
where custname=empname)

```

לצורך זה היינו צריכים להכניס מישחו לתוך הטבלה של employee שנמצא גם בתור לקוחות כי בpg לא היה.

```

insert into SYSTEM.EMPLOYEE (EMPID, EMPNAME)
values (841, 'Lorraine Madsen');

```

זה לא יעבד כי שמות העמודות השתנו, כי פה בניגוד לcustomer העדפנו את שמות העמודות שהתקבלו מבחן...

בנוסף יש לנו בעיה שקשה מאוד להוציא עבד למערכת כי צריך להוציא קודם קודם אדם. כך שכניסינו להכניס את השם choreanza היה הערך הלא נכון כי היה צריך להוציא מייל והוא גם צריך להוציא משכורת, (כלומר חסר עמודות וגם הורידן עמודת שם)

נוסיף את

```
insert into EMPLOYEE
```

```
values;(10 ,0,841)
```

וזה נכון.

עכשו השאלה לא עובדת כי שמות העמודות השתנו, אבל לפני שננסה לשנות נשים לב למשהו:

כרגע אין בכלל את העמודות של שמות בתוך employee ו customer.

ואם נשווה בין ה ρ זה לא ייתן תוצאה רצiosa כי יתרן שני ρ עם אותו שם. את השמות צריך להשיג מ person.

נראה את השאלה החדשה שיצרנו ואת התוצאה שיצאה (כרגע לפני המחקה)

```
select * from customer natural join person
```

```
where custid=person_id
```

```
and person_name in
```

```
(select unique person_name
```

```
from employee natural join person
```

```
where e_person_id=person_id;()
```

כלומר תבחר מתוך customer את האנשים שהשם שלהם נמצא ב employee. השאלה הפנימית משיגה את השמות שנמצאים בשנייהם. החיצונית מוחקמת כל מי שזה השם שלו.

הסיבה שזה(CC) מסובך היא שכשעשיתי join natural השמות של העמודות לא אותו דבר ב person ו customer לדוגמה, אז כדי שלא יבוצע כל שילוב אפשרי צריך לוודא שהוא ρ שווים.

נצלם מה יש לפני:

```

select * from customer natural join person
where custid=person_id
and person_name in
(select unique person_name
from employee natural join person
where e_person_id=person_id);

```

```

select * from customer natural join person w|
select * from employee natural join person w|

```

	CUSTID	INCLUB	C_MAIL_ID	PERSON_ID	PERSON_NAME	BIRTH_DAY	PAREA_ID
1	841	NO		841	Aenee Atatos	05/12/1971	598
2	847	YES		847	Hannah Maren	18/04/2018	550
3	856	NO		856	Mary Beth Evans	19/02/1942	659
4	859	YES		859	Geggy Dalley	11/04/1948	56
5	864	YES		864	Amanda Wright	16/01/1969	664
6	865	YES		865	Gwyneth Hewitt	08/06/1923	797
7	890	NO		890	Michelle Reilly	22/11/1936	76
8	892	NO		892	Jimmy Ruffalo	05/05/1942	532
9	901	YES		901	Jennifer Gyllen	21/07/1977	118
10	903	NO		903	Garland Holmes	19/08/1960	118
11	906	NO		906	Vonda Hurley	17/12/1949	582
12	911	YES		911	Julio Portman	10/12/1992	555
13	929	NO		929	Arturo Peet	09/12/2000	41
14	934	NO		932	Geoff Sheare	31/03/1957	180
15	947	YES		947	Marc Dean	14/05/2020	539
16	951	NO		951	Nicky Emmerich	06/08/1954	467
17	952	NO		952	Pete Fichtner	19/07/1988	38

הסיבה שזה בחר ככ הרבה שורות זה כי כל מי שנמצא ב `customer` וב `employee` נמצא גם ב `person`.

ובגלל שיש ככ הרבה אנשים ב `customer` כל מי שנמצא ב `person` נמצא גם ב `customer` ולכן כל מי שב `employee` נמצא ב `customer`. **אד-column ימחקו**

כי עכשו השמות לפני update הם תואימים תמיד!!!. ננסה למחוק:

`delete from employee`

`where e_person_id in`

`(select person_id from customer natural join person`

`where custid=person_id`

`and person_name in`

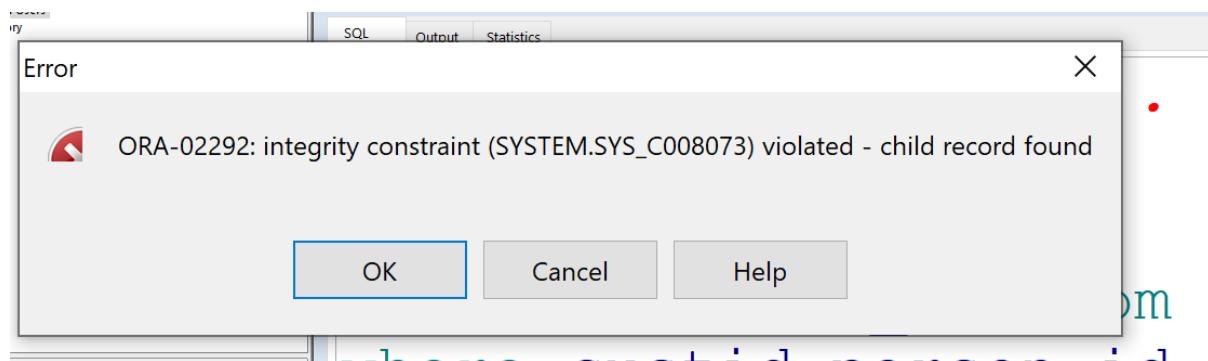
`(select unique person_name`

`from employee natural join person`

`where e_person_id=person_id;()`

הירוק זה השאלה שלפני כן הייתה `SELECT`.

קיבלו את השגיאה הנ"ל:



הסיבה היא כי organize צריך את העובדים האלה.

לכן נמחק קודם מתוך organize את מי שהמספר עובד הוא אחד מאליה:

```
delete from organize
```

```
where empid in
```

```
(select person_id from customer natural join person
```

```
where custid=person_id
```

```
and person_name in
```

```
(select unique person_name
```

```
from employee natural join person
```

```
where e_person_id=person_id;()
```

ועכשיו נמחק:

עכשו אין אנשים כאלה:

```

select person_id from customer natural join person
where custid=person_id
and person_name in
(select unique person_name
from employee natural join person
where e_person_id=person_id));

```

PERSON_ID

ניתור אילוצים:

1- לבדוק שבקופה חדשה יש מספיק כסף

Alter table cashregister

Add constraint enoughmoneyincashreg

Check cashmoney>70

ובאמת שהנכונה

;(400,60)insert into cashregister values

קיבלו



ORA-02290: check constraint (SYSTEM.ENOUGHMONEYINCASHREG) violated

OK

Cancel

Help

עשינו את זה דומה ל update כי אם מנסים להוסיף תנאי שהנתונים הקיימים לא מקיימים נראה שמתיקבלת שגיאה. כמו כן נראה שאין דרך להוסיף תות שאלתה ב check

ביצענו דוגמא נוספת כשהפכנו את העמודות שיוכלו להיות null

alter table

person

modify

) parea_id NULL; (

- דוגמא נוספת להלן:

(הינו צריכים לשנות ידנית את datatype ישיה null כדי שהיא לנו יכולה לשנות)

נסה להכני

insert into prodtype values(null,998)

	PRODTYPE	PRODTYPEID
288	Azgqlblpql	92
289	Otfqyhybue	93
290	Zvvfrhludq	94
291	Nsukiarjud	95
292	Vojxkmxiq	96
293	Laiwecwmat	97
294	Lpbtcwqnvm	98
295	Imevczacy	99
296	Vwokrakgj	100
297	Letpvqnucb	101
298	Squlmjijdy	102
299	Rqsrlruszu	103
300	Xcpukkognt	104
301	Ifpvfbuzvz	105
302	Xqecbfaebd	106
303	Ffjmomvdyh	107
304	Ijsxkbhpnc	108
305	Nbqlpdfoc	109
306	Mchvlbrapj	110
307	Nfcfsqczr	111
308	Ksnhqwjhsp	112
309	Womrdcixjh	113
310		998

ונצליח

הרצינו:

ALTER TABLE prodtype MODIFY(PRODTYPE NOT NULL);

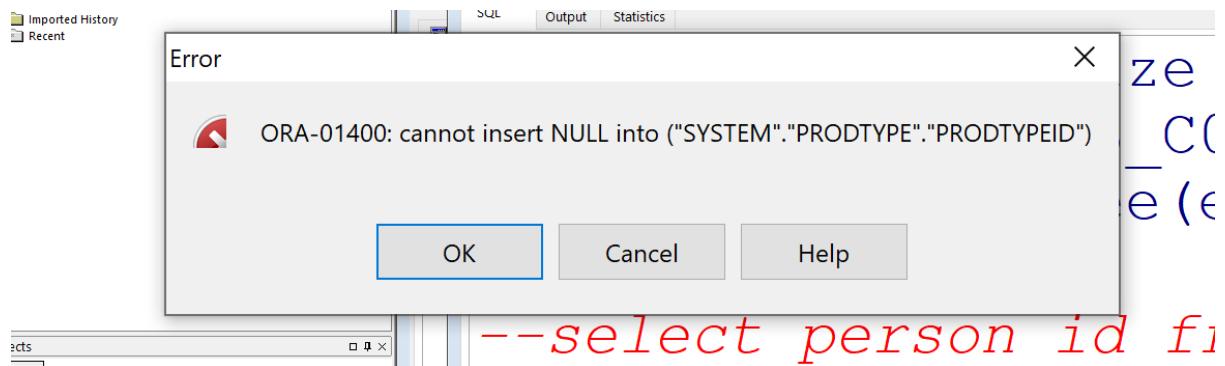
לא יעבוד כי יש נתון שסתור איז נמחק אותו ונוסיף את התנאי

שוב

(מחיקה ידנית ב data edit ואיז לחיצה על המינוס -)

```
insert into prodtype values(null,999,)
```

וקיבלנו שגיאה:



נוסף משהו עם default:

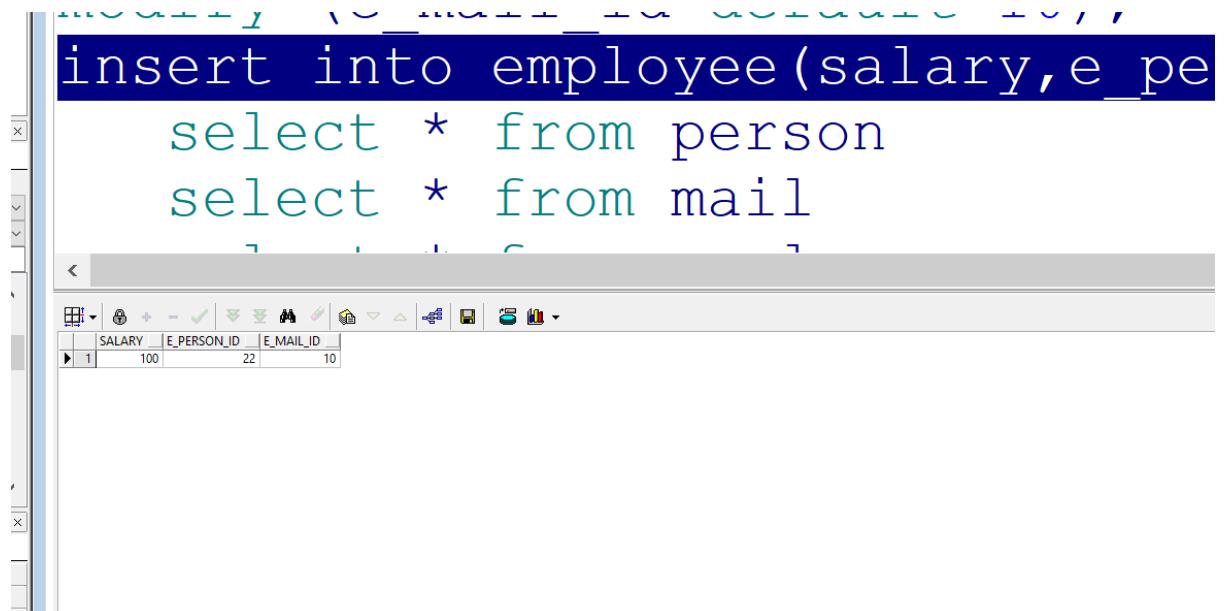
```
alter table employee
```

```
modify (e_mail_id default 10);
```

ונוסה להכניות:

```
insert into employee(salary,e_person_id) values(100,22);
```

ונקבל:



הערה:

נמחקו חלק מהנתונים ושיחזרנו הרוב כך שייתכנו נתונים שהיו באחת הטלנות ועכשו לא שם.

כמו כן כדי לחזור אחורה מחקנו חלק מהCONSTRAINTS

חלק 6

ניצור שאלות על החלק של המיל
(וב透ר כר גם על שלו כי יש לנו חלקים בcustomer נניח)

Select:

1- תשיג לכל חבילה כמה סהכ עולימ הבולים שלה .

היה פה טעות קודם כי עשינו NATURAL JOIN בלי להשוות את הערכים של STAMPID ו STPSTAMPID

```
select stp_package_id,sum(stamp_price)
from stamp_for_package,stamp
where stp_stamp_id=stamp_id
group by stp_package_id;
```

The screenshot shows the Oracle SQL Developer interface with a SQL window titled "SQL Window - queriesonmail.sql". The query has been executed, as indicated by the results grid below. The results show the total price for each package ID.

STP_PACKAGE_ID	SUM(STAMP_PRICE)
1	30
2	179
3	878
4	280
5	468
6	408

Execution details at the bottom of the window show "56 rows selected in 0.062 seconds".

2- לכל צייר מה הוא קנה במלולת. יתכן שקנה יותר ממוצר אחד. קיימים כאלה בגל של הצייר ולא בשם שלו. עשינו השוואה בין מספר האדם של הליקון והצייר ככה שלא צריך קישור ל

person

.

```
select prodname,a_person_id from
product natural join buys_what natural join buy natural join artist
where custid=a_person_id
group by stp_package_id;
--what does artist buy in supermaket
select prodname,a_person_id from
product natural join buys_what natural join buy natural join artis
where custid=a_person_id
select * from artist--Ppwvhgusp 932 yeeyy!!!
select * from buy natural join buys_what
select * from product
--to everv employee how manv people live in that area
```

	PRODNAME	A_PERSON_ID
▶	1 Ppwvhgusp	400
	2 Qoupqnsq	604
	3 Aodvdmcrd	128
	4 Dredagwgf	128
	5 Pgkkuhuvq	904
	6 Tankrzhe	128
	7 Qtmsuyaxs	904
	8 Akhimgdjg	604
	9	128
	10	128
	11	128
	12	128

12 rows selected in 0.234 seconds

3- לכל עובד מחזיר כמה אנשים עובדים באותו אזור. מחזיר רק 165 ולא 300 ומשהו כי יש עובדים שהאזור שלהם הוא null אך כשהנעשה = הם לא יבחרו

```
select count(*),e_person_id,p.person_name
from person p2 ,employee ,person p
where p.person_id=e_person_id
and p2.parea_id=p.PAREA_ID
group by e_person_id,p.person_name
```

אחריו שקיישרנו בין כל שני אנשים שగרים באותו אזור נעשה GROUP BY לפי ה

ID

 של הראשון(לא לדאוג שיש הכל כפול)

SQL Window - queriesonmall.sql

SQL Output Statistics

```

select * from product
--to every employee how many people live in that area
--checking answer: select * from person where person_id=34
select count(*),e_person_id,p.person_name
from person p2 ,employee ,person p
where p.person_id=e_person_id
and p2.parea_id=p.PAREA_ID
group by e_person_id,p.person_name
--does not get all employees because alot of them dont have an area
--what is total weight of parcels for specific person
select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id
<

```

	COUNT(*)	E_PERSON_ID	PERSON_NAME
▶	1	2	678 Henry_Winslet
	2	2	312 Sal_Finn
	3	3	183 Rowan_Fiorentin
	4	3	582 Max_Bachman
	5	2	123 Lynette_Browne
	6	2	180 Sinead_Colin_Bo

14:1 SYSTEM@XE 165 rows selected in 0.531 seconds

-4

לכל אדם, מה המשקל הממוצע של חבילות שהוא קנה. אפשר לחבר בלי CUSTOMER כי יש ID בשנייהם.

```

select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id
group by p_person_id, person_name

```

```

--group by e_person_id
--does not get all employees because alot of them dont have
--what is total weight of parcels for specific person
select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id

```

The screenshot shows the execution results of the SQL query. The results are displayed in a grid format with columns PERSON_NAME and AVG(PACKAGE_WEIGHT). The data shows six rows of results:

	PERSON_NAME	AVG(PACKAGE_WEIGHT)
1	Henry_Winslet	920
2	Sinead_Colin Yo	124
3	Candice_Duvall	440.666666666667
4	Rosie_Colin You	573.5
5	Neil_Cockburn	367
6	Rosie_Glover	763

Below the results, the status bar shows "86 rows selected in 0.422 seconds".

Update:

1- לכל העובדים שהם ציירים תעדכן את המשכורת:

```

update employee
set salary=salary+10
where
e_person_id in (select a_person_id from artist)

```

כדי להראות שעובד נראתה דוגמה של מישחו לפני ואחרי:

```

where
e_person_id in (select a_person_id fro

```

```

select * from employee;
select * from artist;
--for every customer update area of ma

```

The screenshot shows the execution results of the update query. The results are displayed in a grid format with columns SALARY, E_PERSON_ID, and E_MAIL_ID. The data shows one row of results:

	SALARY	E_PERSON_ID	E_MAIL_ID
1	650	24	10

אחרי:

```
select * from employee;
```

	SALARY	E_PERSON_ID	E_MAIL_ID
▶	1	660	24
▼	2	220	26

נראה שהבן אדם און ב-artist:

```
select * from employee;
select * from artist;
--for every customer update
```

	A_PERSON_ID
▶	1
▼	2
▼	3

הערה- לא ניתן לנו זמנים כשבשים update.

2-תעדכן את כל האזורי מיילים של customers להיות כמו של ה customer עצמו.

update mail M

```
set M.m_area_id=(select parea_id from
mail,customer,person
where c_mail_id=mail_id and custid=person_id
and M.Mail_id=mail.mail_id
)
where mail_id in (select mail_id from customer);
```

מכל השילובים האפשריים תכניס רק את מי שזה המיל שלו

השאילתת הפנימית מוצאת לכל איבר במיל שנמצא ברשימה מיילים של לקוחות(מכאן ה WHERE החיצוני) את האזור של הלוקו עם אותו מיל

הערה חשובה- בגלל שכשחכנו את הנתונים באינטגרציה בחרנו את המידע מתוך customer שלנו הלה לאיבוד כל ה-mail שלהם, לכן הכנסנו ידנית קודם נתונים לתוכה כדי להראות שבאמת יש שינויים.

update person

```
set parea_id=3
```

```
where person_id=838;
```

לאדם 838 יש עכשו איזור 3

update customer

```
set c_mail_id=11--area is 1 not 3
```

```
where custid=838;
```

לאדם 838 יש עכשו מיל של 11 אבל למיל 11 יש איזור לא מתאים.

(יתכן שנצטרך את השורה זו לעזור אם הערכים להשמה בהמשך הם null)

```
(alter table MAIL modify m_area_id null;
```

תקשר עכשו בין איזור שימושו גור לאיזור של המיל שלו. לא בוחר דברים שהם null בגלל ה=
(השוויה)

```
--lets try to do it with select
select m_area_id,parea_id from
mail,customer,person
where c_mail_id=mail_id and custid=person_id
--now make update out of it
update mail M
set M.m_area_id=(select parea_id from
mail,customer,person
```

	M_AREA_ID	PAREA_ID
▶	1	3

עכשו נרים את update

```
update mail M
set M.m_area_id=(select parea_id from
mail,customer,person
where c_mail_id=mail_id and custid=person_id
and M.Mail_id=mail.mail_id
```

```
)
where mail_id in (select mail_id from customer);
```

עשינו id מילוי התיקות לשאיילת החיצונית בתוך הפנימית כדי שימצא את מי שמלל השילובים האפשריים יש לו את הid המתאים, בהנחה שיש רק אחד כזה, כי אם יש כמה עם אותו מיליז זה לא יעבוד. גם זה פחות הגיוני. לא רצימ את כל הPAIRID רק את מי שהשתלב עם ה id מ m_area_id הנוכחי. ועושה ככה לכל שורה.
בוחר את ID PAREAID ומעדכן את המילוי.

```
השורה
where mail_id in (select mail_id from
customer);
```

זה כדי שאל תעדכן mail אם הוא לא נמצא בcustomer בכלל. אין צורך

```

update mail M
set M.m_area_id=(select parea_id from
mail,customer,person
where c_mail_id=mail_id and custid=person_id
and M.Mail_id=mail.mail_id
)
where mail_id in (select mail_id from customer);
--it did not work because we dont have any,
--all customers have nullable mail id
--create things we can change
update person
set parea_id=3
where person_id=838;
update customer
set c_mail_id=11--area is 1 not 3
where custid=838;
update customer
set c_mail_id=12--area is 1 not 147??
where custid=394;
update person
set parea_id=147

```

SYSTEM@XE 187 rows updated in 0.015 seconds

ונרץ שוב קישור בין איזור מגורים לאיזור דואר

```

select m_area_id,parea_id from
mail,customer,person
where c_mail_id=mail_id and custid=person_id
--now make update out of it
update mail M
set M.m_area_id=(select parea_id from

```

	M_AREA_ID	PAREA_ID
▶	1	3

אפשר לראות שזה אותו דבר
 אחרי כל שאילתת עשינו rollback ישמור השינויים.)

Delete:

-1

טמיהן את כל הבולים שלא בשימוש

```

delete from stamp
where stamp_id not in
)select stp_stamp_id from stamp_for_package(
```

```

--delete stamps who are not used by any parcer
select * from stamp
delete from stamp
where stamp_id not in
(select stp_stamp_id from stamp_for_package)
--delete all employees who live in area x
delete from employee
where e_person_id in
(
select person_id
from person
where parea_id=1
)
select * from employee where e_person_id=22
select * from person where person_id=22
```

SYSTEM@XE 99 rows deleted in 0.031 seconds

וונא כמה בולים יש לפני

```
--delete stamps who are not used by any parcel
select * from stamp
delete from stamp
where stamp_id not in
(select stp_stamp_id from stamp_for_package)
--delete all employees who live in area x
delete from employee
where e_person_id in
()
```

	STAMP_ID	STAMP_NAME	YEAR_CREATED	STAMP_PRICE	AMOUNT_IN_STOCK	S
▶	1	70	flower	1950	1	20
	2	71	Home	1951	4	30
	3	72	Banana	1968	5	46
	4	64	lithium	2017	497	445
	5	867	gallium	2049	820	133
60:1 SYSTEM@XE 156 rows selected in 0.673 seconds						

וונא כמה בולים יש אחריו:

```
--delete stamps who are not used by any parcel
select * from stamp
delete from stamp
where stamp_id not in
(select stp_stamp_id from stamp_for_package)
--delete all employees who live in area x
```

	STAMP_ID	STAMP_NAME	YEAR_CREATED	STAMP_PRICE	AMOUNT_IN_STOCK	S
▶	1	70	flower	1950	1	
	2	71	Home	1951	4	
	3	72	Banana	1968	5	
	4	601	aluminum	1861	478	
	5	802	thorium	2013	86	
	6	209	oxygen	1881	77	
	7	458	nobelium	2045	434	
	8	138	nickel	2023	14	
60:1 SYSTEM@XE 57 rows selected in 0.25						

2- תמחק את כל העובדים שגרם באזור 1

```
delete from employee
```

```
where e_person_id in
```

```
(
```

```

select person_id
from person
where parea_id=1
(
----- -----
(select stp_stamp_id from stamp_for_package)
--delete all employees who live in area x
delete from employee
where e_person_id in
(
select person_id
from person
where parea_id=1
)
select * from employee where e_person_id=22
select * from person where person_id=22

```

SYSTEM@XE 2 rows deleted in 0.016 seconds

נראה דוגמה של מישחו שזה משפיע עליו:
לפני:

/
select * from employee where e_person_id=22

	SALARY	E_PERSON_ID	E_MAIL_ID	
▶	1	500	22	10

אחריו:

```
select * from employee where e_person_id=22
```

The screenshot shows the Oracle SQL Developer interface. A query window is open with the following SQL statement:

```
select * from employee where e_person_id=22
```

The results grid displays three columns: SALARY, E_PERSON_ID, and E_MAIL_ID. There is one row of data shown:

SALARY	E_PERSON_ID	E_MAIL_ID
	22	

ובאמת הואAGR באזורי 1:

```
select * from person where person_id=22
```

The screenshot shows the Oracle SQL Developer interface. A query window is open with the following SQL statement:

```
select * from person where person_id=22
```

The results grid displays five columns: PERSON_ID, PERSON_NAME, BIRTH_DAY, and PAREA_ID. There is one row of data shown:

PERSON_ID	PERSON_NAME	BIRTH_DAY	PAREA_ID
22	Noam	01/03/2002	1

הווסף אינדקסים:

```
CREATE INDEX index_person_name  
ON person(person_name);
```

לפנינו:

SQL Window - queriesonmall.sql

SQL Output Statistics

```
select * from product
--to every employee how many people live in that area
--checking answer: select * from person where person_id=34
select count(*),e_person_id,p.person_name
from person p2 ,employee ,person p
where p.person_id=e_person_id
and p2.parea_id=p.PAREA_ID
group by e_person_id,p.person_name
--does not get all employees because alot of them dont have an area
--what is total weight of parcels for specific person
select person_name,avg(package_weight)
from parcel , person
where p person id=person id
<
```

	COUNT(*)	E_PERSON_ID	PERSON_NAME
▶	1	2	678 Henry_Winslet
	2	2	312 Sal_Finn
	3	3	183 Rowan_Fiorentin
	4	3	582 Max_Bachman
	5	2	123 Lynette_Browne
	6	2	180 Sinead_Colin_Vo

SYSTEM@XE 165 rows selected in 0.531 seconds

: אחר:

SQL Window - queriesonmail.sql

SQL Output Statistics

```

select * from product
--to every employee how many people live in that area
--checking answer: select * from person where person_id=34
select count(*),e_person_id,p.person_name
from person p2 ,employee ,person p
where p.person_id=e_person_id
and p2.parea_id=p.PAREA_ID
group by e_person_id,p.person_name
--does not get all employees because alot of them dont have an e
--what is total weight of parcels for specific person
select person_name,avg(package_weight)
from parcel , person
where p person id=person id
<
```

	COUNT(*)	E_PERSON_ID	PERSON_NAME
▶	1	2	678 Henry_Winslet
	2	2	312 Sal_Finn
	3	3	183 Rowan_Fiorentin
	4	3	582 Max_Bachman
	5	2	123 Lynette_Browne
	6	2	180 Sinead_Colin_Bo

14:1 SYSTEM@XE 165 rows selected in 0.328 seconds

CREATE INDEX index_package_weight

ON parcel (package_weight);

לפניכם:

```

group by e_person_id
--does not get all employees because alot of them dont have
--what is total weight of parcels for specific person
select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id

```

	PERSON_NAME	Avg(PACKAGE_WEIGHT)
1	Henry_Winslet	920
2	Sinead_Colin Yo	124
3	Candice_Duvall	440.6666666666667
4	Rosie_Colin You	573.5
5	Neil_Cockburn	367
6	Rosie_Glover	763

22:1 SYSTEM@XE 86 rows selected in 0.422 seconds

אחרי:

SQL Window - queriesonmail.sql

SQL Output Statistics

```

where p.person_id=e_person_id
and p2.parea_id=p.PAREA_ID
group by e_person_id,p.person_name
--does not get all employees because alot of them dont have an area
--what is total weight of parcels for specific person
select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id
group by p_person_id,person_name
--update salary of artist who are employees
update employee
set salary=salary+10
where

```

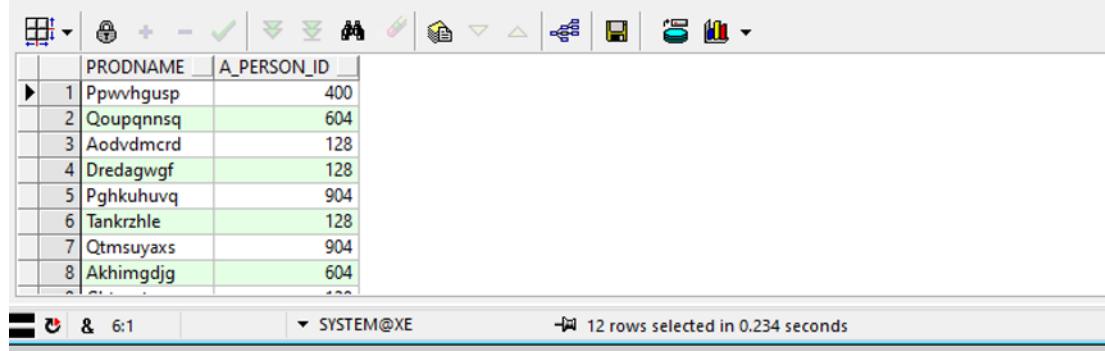
	PERSON_NAME	Avg(PACKAGE_WEIGHT)
1	Henry_Winslet	920
2	Sinead_Colin Yo	124
3	Candice_Duvall	440.6666666666667
4	Rosie_Colin You	573.5
5	Neil_Cockburn	367
6	Rosie_Glover	763

21:1 SYSTEM@XE 86 rows selected in 0.219 seconds

```
CREATE INDEX index_prod_name  
ON product (prodname);
```

לפניהם:

```
group by stp_package_id;  
--what does artist buy in supermaket  
select prodname,a_person_id from  
product natural join buys_what natural join buy natural join artis  
where custid=a_person_id  
select * from artist--Ppwvhgusp 932 yeyy!!!  
select * from buy natural join buys_what  
select * from product  
--to every employee how many people live in that area
```



	PRODNAME	A_PERSON_ID
▶	1 Ppwvhgusp	400
2	Qoupqnnsq	604
3	Aodvdmcrd	128
4	Dredagwgf	128
5	Pghkuhuvq	904
6	Tankrzble	128
7	Qtmsuyaxs	904
8	Akhimgdjg	604
...		...
		120

6:1 SYSTEM@XE 12 rows selected in 0.234 seconds

אחריהם:

SQL Window - queriesonmain.sql

SQL Output Statistics

```

group by stp_package_id;
--what does artist buy in supermaket
select prodname,a_person_id from
product natural join buys_what natural join buy natural join art
where custid=a_person_id
select * from artist--Ppwvhgusp 932 yeyy!!!
select * from buy natural join buys_what
select * from product
--to every employee how many people live in that area
--checking answer: select * from person where person_id=34
select count(*),e_person_id,p.person_name
from person p2 ,employee ,person p
where p.person_id=e_person_id

```

	PRODNAME	A_PERSON_ID
1	Ppwvhgusp	400
2	Qoupqnnsq	604
3	Aodvdmcrd	128
4	Dredagwgf	128
5	Pghkuhuvq	904
6	Tankzrhlz	128

SYSTEM@XE 12 rows selected in 0.126 seconds

חלק -7 :view

הVIEW הראשון:

```

CREATE OR REPLACE VIEW productsforUserplusTotalPrice
as
select
custid,inclub,saleforclub,buydate,bid,cashregid,prodname,
prodtype,buys_what.quantity_,price,price*quantity_
as total_price
from customer
natural join buy
natural join buys_what
natural join product
natural join prodtype

```

מקשר בין לקוחות למוצרים שהוא קנה.

SELECT

- 1- לכל קנייה להציג את המחיר שלה סהכ אחדי הנחה.

```
select  
bid, count(*), sum(total_price * (saleforclub/100))  
from productsforUserplusTotalPrice  
group by bid
```

--buydate (and quantity) which is buy table but every product has 1

```
select bid, count(*), sum(total_price * (saleforclub/100))  
from productsforUserplusTotalPrice  
group by bid  
--select customers that bought at least 3 types (!! ) of products.  
select custid, count(prodtype)  
from productsforUserplusTotalPrice  
group by custid
```

	BID	COUNT(*)	SUM(TOTAL_PRICE*(SALEFORCLUB/100))
▶	1	474	47245.72
	2	294	482312
	3	34	41729.1
	4	650	30343
	5	948	21015
	6	967	1190.64
	7	815	133383.6
	8	861	44007.6
	9	784	24210.72
	10	150	77525.72

- 2- בחר את כל הלקוחות שקנו יותר משני סוגים מוצריים

```
select custid, count(prodtype)  
from productsforUserplusTotalPrice  
group by custid  
having count(prodtype)>2
```

```

--buydate (and quantity) which is buy table but every pr
select bid, count(*), sum(total_price*(saleforclub/100))
from productsforUserplusTotalPrice
group by bid
--select customers that bought at least 3 types(!!) of p
select custid, count(prodtype)
from productsforUserplusTotalPrice
group by custid
having count(prodtype)>2

update productsforUserplusTotalPrice

```

CUSTID	COUNT(PRODTYPE)
1	560
2	128
3	759
4	211
5	867
6	301

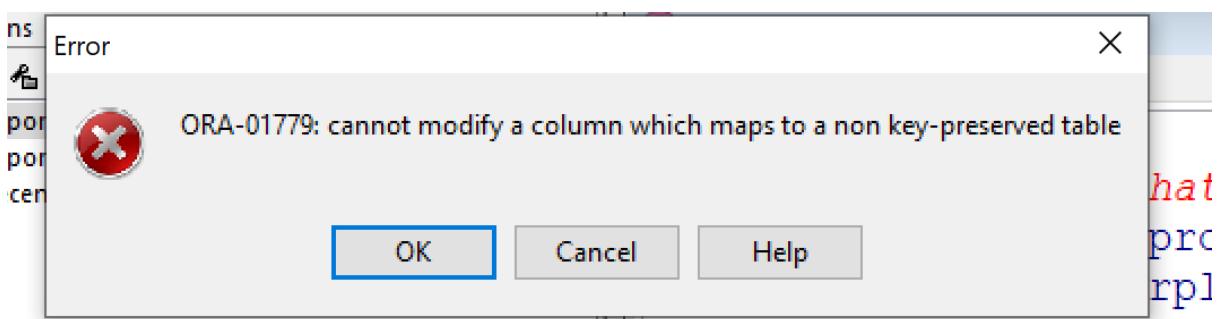
27:1 SYSTEM@XE 7 rows selected in 0.063 seconds

- לא עבד מכיוון שלכל שורה שיש לה מחיר כלומר ב product יש VIEW כמה תאריכי קנייה מותאימים. אז זה לא יודע למה להשווות. גם QUANTITY יש כמה.

```

update productsforUserplusTotalPrice
set price=0
where quantity_>100
      and buydate=to_date('11/03/2015', 'dd/mm/yyyy')

```

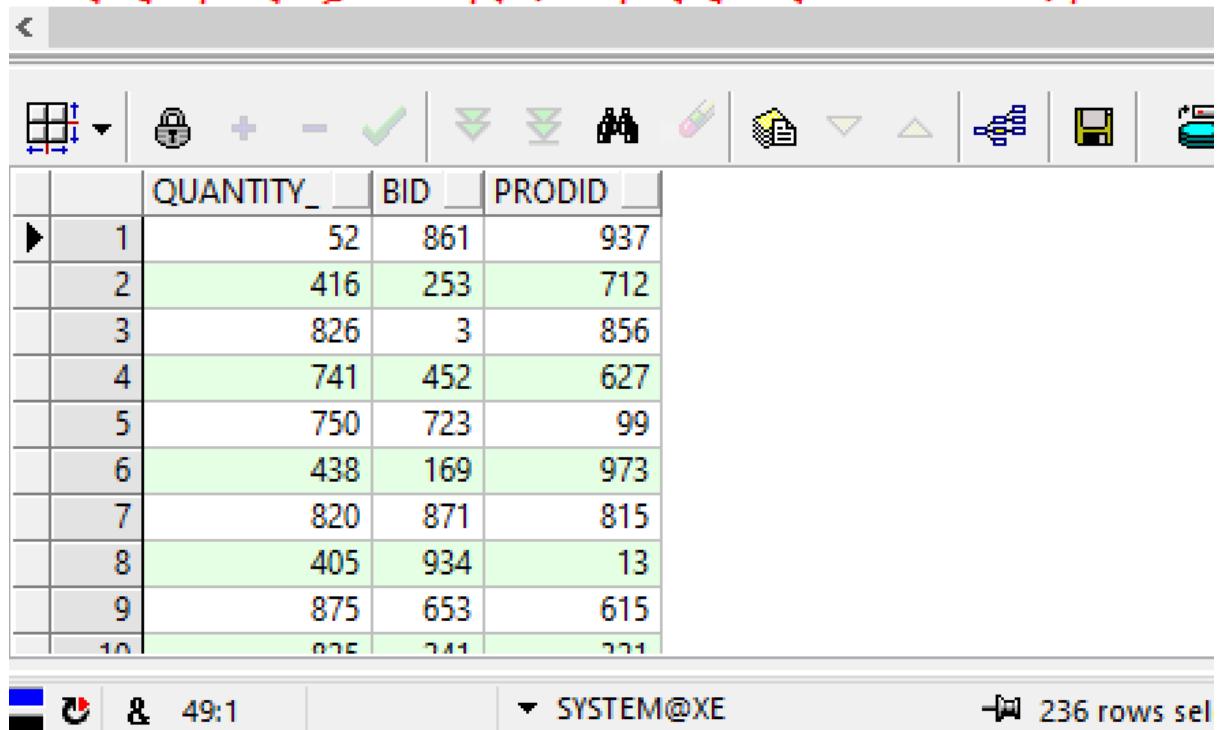


DELETE

```
delete from productsforUserplusTotalPrice
where total_price*(saleforclub/100)>100000
and inclub='YES'
```

זה דוקא כן עובד. בדקנו טבלה טבלה ומיל שמשתנה זה WHAT_BUYS. הסיבה לכך היא שזו טבלה של רבים לרבים. כך שכל שורה בה שיש בה SALEFORCLUB יש לה INCLUB אחד ולהפוך (כי לכל קנייה יש לפחות אחד וכל מוצר יש SALEFORCLUB אחד). נהייה שם אחרי הריצה פחות שורות.

--works because we don't have aggregation



	QUANTITY_	BID	PRODID
►	1	52	861
	2	416	253
	3	826	3
	4	741	452
	5	750	723
	6	438	169
	7	820	871
	8	405	934
	9	875	653
	10	256	221

לעומת לפני כן שיש 300 שורות כמעט.

VIEW מופר 2

```
CREATE OR REPLACE VIEW amountOfParcelssForStamp AS
--how many parcels have this stamp
select stamp_id,person_id,person_name,birth_day,
stamp_name,year_created,stamp_price,amount_in_stock,c
ount(*) as countingParcels,
sum(package_weight) counting_package_weight,sum(packag
e_price) as counting_package_price
from person,artist,stamp,stamp_for_package,parcel
where person_id=a_person_id
```

```

and s_person_id=a_person_id
and stamp.stamp_id=stamp_for_package.stp_stamp_id
and stp_package_id=parcel.package_id
group by
stamp_id, person_id, person_name, birth_day, stamp_name, y
ear_created, stamp_price, amount_in_stock
natural join.
הסיבה שזה(CC יותר מסובך זה כי אי אפשר לעשות

```

שינו לב עשינו פה group by.

:Update & delete

```

update amountOfParcelssForStamp
set amount_in_stock=100
where
counting_package_weight/counting_package_price>1
--does not work for same reason
delete from amountOfParcelssForStamp
where birth_day>to_date('15/03/1997','dd/mm/yyyy')
and stamp_name like "%a%

```

בשנים קיבלנו אותה הערה שאפשר לשנות את ה view .

הסיבה היא שעשינו BY GROUP.

SELECT

- לכל STAMP כמה הרווחנו ממנו ומידע על מי שציר אותו (**כולל מחיר הבולים**) (אבל רק על אלה שמשקל החבילות גדול מ100)

```

select stamp_id, person_name, birth_day,
stamp_name, year_created, (stamp_price+counting_package_price)
from amountOfParcelssForStamp
where counting_package_weight>100

```

	STAMP_ID	PERSON_NAME	BIRTH_DAY	STAMP_NAME	YEAR_CREATED	(STAMP_PRICE+COUNTING_PACKAGE_PRICE)
►	1	934 Noam	01/03/2002	osmium	2008	1434
	2	70 Noam	01/03/2002	flower	1950	133
	3	515 Jane_Lynch	26/02/1953	cerium	1978	1512
	4	697 Frances_Freeman	18/05/2008	cadmium	1986	1209
	5	306 Frances_Freeman	18/05/2008	uranium	1947	1152
	6	930 Hugh_Weston	12/02/2009	bromine	2002	1655
	7	629 Avril_Raybon	25/09/1981	nitrogen	1898	1946
	8	463 Noam	01/03/2002	arsenic	2008	955
	9	165 Dan	15/03/1997	rhenium	1980	1193
	10	Dan	15/03/1997	radon	1960	1224

SYSTEM@XE 51 rows selected in 0.078 seconds

לכל צייר כמה הרווחנו מהחבילות שהיא להם אותו (כולל מחיר הבולים) וכמה חבילות
כolumbia יש

```
select person_id, person_name, count(*)
       sum(stamp_price) + sum(counting_package_price)
  from amountOfParcelssForStamp
group by person_id, person_name

--for every artist how much we earned from his stamp. add how many .
select person_id, person_name, count(*),
       sum(stamp_price) + sum(counting_package_price)
  from amountOfParcelssForStamp
group by person_id, person_name
```

	PERSON_ID	PERSON_NAME	COUNT(*)	SUM(STAMP_PRICE)+SUM(COUNTING_)
▶	1	81 Elle_Larter	7	7712
	2	24 Gad	2	918
	3	22 Noam	5	4400
	4	623 Hugh_Weston	5	6663
	5	282 Kevin_Robbins	5	4945
	6	23 Dan	6	4876
	7	400 Olympia_Delta	6	6190
	8	904 Jane_Lynch	2	1759
	9	128 Frances_Freeman	5	4348
	10	761 Lorraine_Garcia	5	5500

אין הרבה שורות כי עושים GROUP לפי בול ואז GROUP לפי צייר

GRANT AND REVOKE

"יצרנו משתמש חדש בשם CHAYA . כשנכנסים מבקש סיסמא חדשה שהוחלפה
****7

```
create user CHAYA
identified by aaaa--password. new one is ****
temporary tablespace temp
profile default
password expire;
grant create session to CHAYA;
grant create table to CHAYA;
```

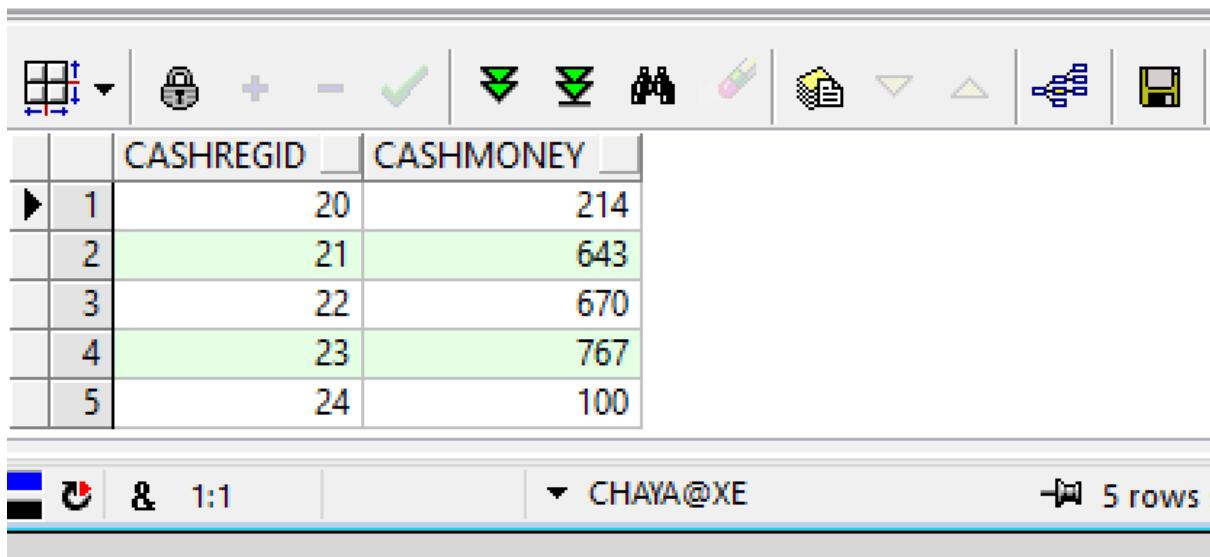
נתנו לדוגמה הרשאות גישה על CASHREGISTER

```
grant select, update, delete on cashregister to  
CHAYA  
  
alter user CHAYA quota unlimited on system;
```

נתינה כמה מקום בזיכרון שירצה המשתמש זהה כי סומכים עליון.

בתוך משתמש CHAYA מרצים

```
select * from SYSTEM.cashregister  
  
ו ניתן
```



	CASHREGID	CASHMONEY
▶	1	20
	2	643
	3	670
	4	767
	5	100

Bottom status bar: CHAYA@XE, 5 rows

נתינה הרשאה לשינוי גם. ניתן לעשות הכל.

```
grant all on cashregister to chaya
```

נכשינו בחיה אפשר לבצע שינויים שינוניים לatable

```

SQL Window - select * from SYSTEM.cashregister select * from SYSTEM.cashregister for update
SQL Output Statistics
select * from SYSTEM.cashregister
select * from SYSTEM.cashregister for update

CASHREGID CASHMONEY
123        999
20         214
21         643
22         670
23         767

```

נתינה SELECT לכל המשתמשים

```

grant select on cashregister to public

```

מחיקת הרשאות שנותנו לAYA;

```

revoke select on cashregister from CHAYA;

```

עדין ניתן לשוחות UPDATE וכו' . משום מה זה לא עובד אז

עשינו ידנית בEDIT מתוך SYSTEM.

מחיקת USER

```

drop user CHAYA

```

סעיף 8:

שאלות עם פרמטרים:

1- בחר מספר של חבילה וקבל את המחיר סהכ של הבולים עליה

```

select stp_package_id,sum(stamp_price)
from stamp_for_package natural join stamp

```

```

group by stp_package_id
having stp_package_id=&<name="package id"
required="true"
hint="between 0 to 999"
type=integer >
;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, there's a 'Report Window - report1.rep' tab and a 'SQL Window - queriesplusparams.sql' tab. The SQL window contains the following query:

```

select stp_package_id,sum(stamp_price)
from stamp_for_package natural join
group by stp_package_id
having stp_package_id=&<name="package id"
required="true"
hint="between 0 to 999"

```

Below the query is a result grid:

STP_PACKAGE_ID	SUM(STAMP_PRICE)	
1	408	70640

A 'Variables' dialog box is overlaid on the interface, containing a single entry:

Name	Value
package id	408

Buttons for 'OK', 'Cancel', and 'Clear' are visible at the bottom of the dialog.

This screenshot shows the same Oracle SQL Developer session after the query has been executed. The results grid now includes a third column:

STP_PACKAGE_ID	SUM(STAMP_PRICE)	
1	408	70640

The status bar at the bottom right indicates "1 row selected in 0.031 seconds".

נתנו רמז כדי שהיה מעניין!

```

Report Window - report1.rep
SQL Window - queriesplusparams.sql
SQL Output Statistics
select stp_package_id,sum(stamp_price)
from stamp_for_package natural join
group by stp_package_id
having stp_package_id=&<name="package
required="true"
hint="between 0 to 999"

```

STP_PACKAGE_ID	SUM(STAMP_PRICE)
1	408
2	70640

Variables

Name	Value
package id	408

OK Cancel Clear

between 0 to 999

- 2-חבר צייר ותבזוק מה הוא קנה בספר. נוון רשימה של ציירים שלא ייבחר סתם מישחו, בעזרת הLIST לא לשכוד TRUE=DESCRIPTION
- . LIST="select a_person_id, person_name from artist, person where a_person_id=person_id"

```

select prodbname,a_person_id, person_name from
product natural join buys_what natural join buy
natural join artist, person
where custid=a_person_id
and a_person_id=person_id
and a_person_id=&<name="choose artist"
list="select a_person_id, person_name from
artist, person where a_person_id=person_id"
"
description="true"
>;

```

The screenshot shows the Oracle SQL Developer interface. On the left, a SQL window displays a query script with a parameter placeholder: &choose artist. On the right, a 'Variables' dialog box is open, listing names and their corresponding values. The value for 'choose artist' is set to 'Gad'. A scroll bar on the right side of the dialog indicates there are more names listed below.

```
--what does artist buy at super
select prodname,a_person_id,person_name
from product natural join buys_what natural join buy natural join artist
where custid=a_person_id
and a_person_id=person_id
and a_person_id=&choose artist
list="select a_person_id, person_name from artist, person where
"
description="true"
```

Name	Value
choose artist	Gad
	Noam
	Dan
	Gad
	Olympia_Delta
	Jane_Lynch
	Frances_Freeman
	Jonny_Cruz
	Hugh_Weston
	Elle_Larter
	Tom_Branagh

כשהרצינו את הziיר נעם ראיינו שקנה

The screenshot shows the Oracle SQL Developer interface. The SQL window contains the same query script as the previous screenshot. Below the window, a results grid displays two rows of data. The columns are PRODNAME, A_PERSON_ID, and PERSON_NAME. The first row has PRODNAME 'Xzdwwdynr', A_PERSON_ID '22', and PERSON_NAME 'Noam'. The second row has PRODNAME 'Hzdczgghi', A_PERSON_ID '22', and PERSON_NAME 'Noam'.

```
--what does artist buy at super
select prodname,a_person_id,person_name
from product natural join buys_what natural join buy natural join artist
where custid=a_person_id
and a_person_id=person_id
and a_person_id=&choose artist
list="select a_person_id, person_name from artist, person where
"
description="true"
```

	PRODNAME	A_PERSON_ID	PERSON_NAME
▶	1 Xzdwwdynr	22	Noam
	2 Hzdczgghi	22	Noam

3-לכל אדם תשיג את ממוצע המשקלים של חבילות שלו. תבחר אם אתה רוצה את שמות האנשים בסדר עולה או יורד.

```
select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id
group by p_person_id, person_name
&<name="order type is descen"
checkbox="desc,asc"
prefix="order by person_name "
>
```

The screenshot shows the Oracle SQL Developer interface. In the top window, a query is being run with a variable dialog open. The variable 'order type is descen' has a checked value. The query itself includes logic to handle both descending and ascending order based on a checkbox parameter.

```

select person_name,avg(package_weight)
from parcel , person
where p_person_id=person_id
group by p_person_id, person_name
  &<name="order type is descen"
checkbox="desc,asc"
prefix="order by person_name "
>

```

In the bottom window, the results of the query execution are displayed in a grid format. The results show the average package weight for each person, ordered by person name.

	PERSON_NAME	AVG(PACKAGE_WEIGHT)
1	Yaphet_Bradford	724
2	Vonda_Hurley	324.5
3	Vendetta_Stiles	511
4	Vanessa_Cozier	797
5	Ilma_Ruson	212.5

Execution statistics at the bottom indicate 86 rows selected in 0.093 seconds.

4-כל המוצרים שמשנוקנו הכמות היא מעל פרמטר X והתאריך קניה הוא אחרי תאריך פרם Y. ארוך Ci זה המידע על המוצר והמידע על הליקו כמעט במלואו, וגם סהכ המחיר ל מוצר במכפלה מחיר וכמות.

```

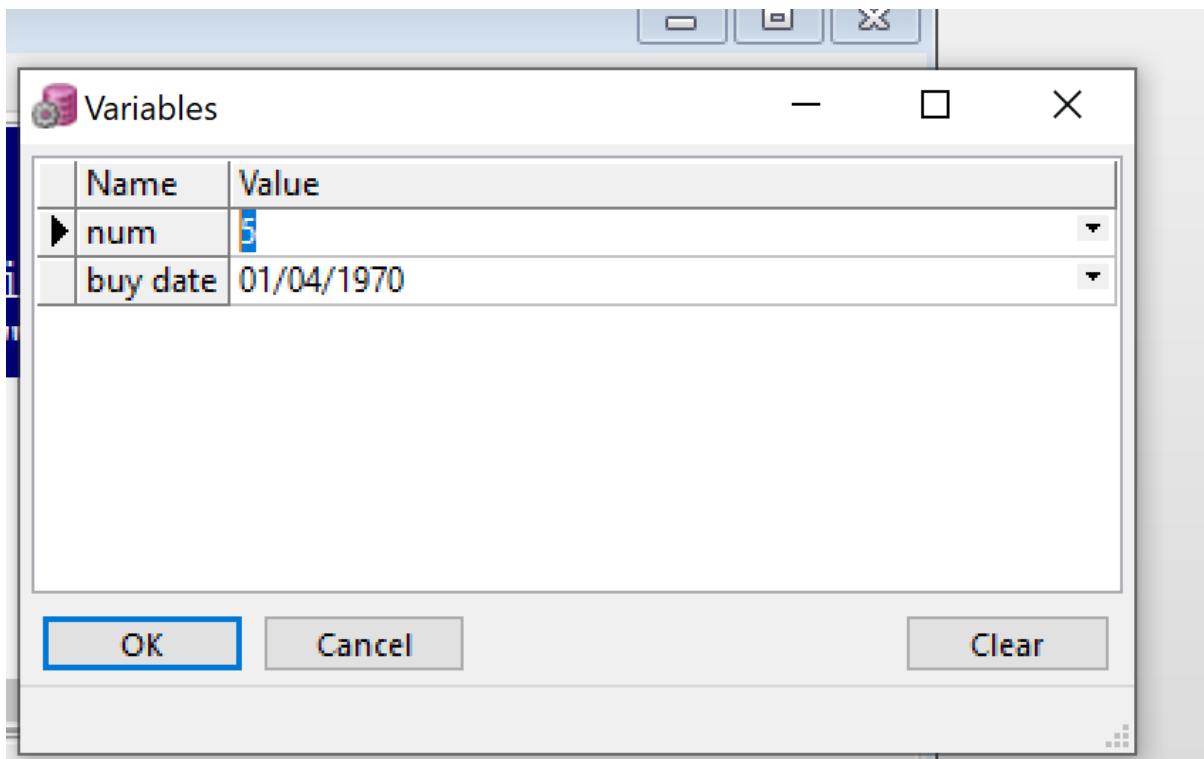
select * from (select
custid,inclub,saleforclub,buydate,bid,cashregid,prodname,prodtype,buys_what.quantity_,price,price*quantity_
as total_price
from customer
natural join buy
natural join buys_what
natural join product

```

```

natural join prodtype)
where quantity_>&lt;name="num" type="integer">
      and buydate>&lt;name="buy date" type="date">

```



אפשר לראות בתוצאות שהתקבל מה שרצינו:

	CUSTID	INCLUB	SALEFORCLUB	BUYDATE	PID	CASHREGID	PRODNAME	PRODTYPE	QUANTITY_	PRICE	TOT
▶	1	344	NO	80 11/03/2015	929	302	Ndbfjvilkw	Gvulchcrbi	999	945	
▶	2	932	NO	93 10/09/2011	992	128	Vckmknktu	Qtfqyhybue	278	715	
▶	3	549	YES	57 11/07/1976	359	250	Agjzrffqv	Jqsbbkjnww	511	907	
▶	4	668	NO	80 17/10/2004	184	69	Iauronnti	Hltdhluhmk	667	727	

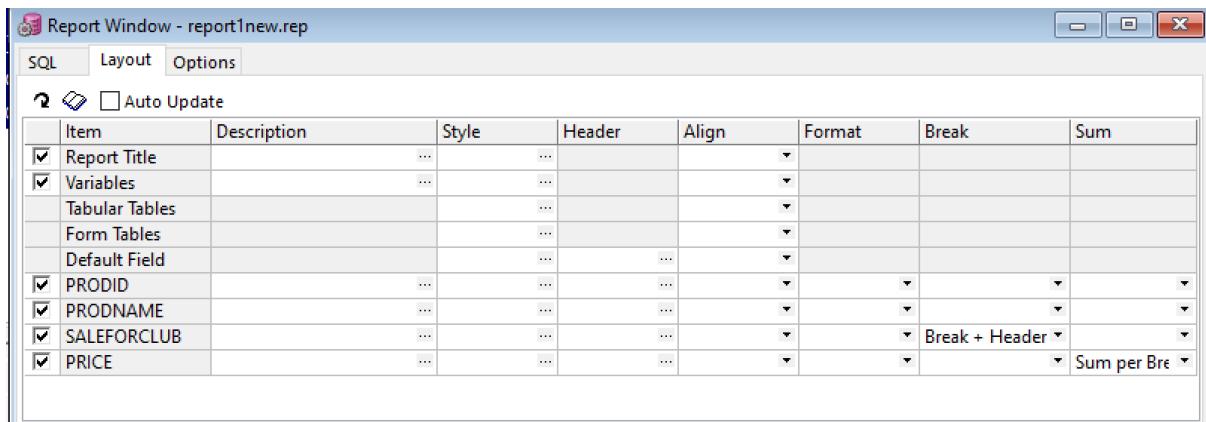
יצירת דוחות:

```

select prodid,prodname,saleforclub ,price
from product
order by saleforclub, prodname

```

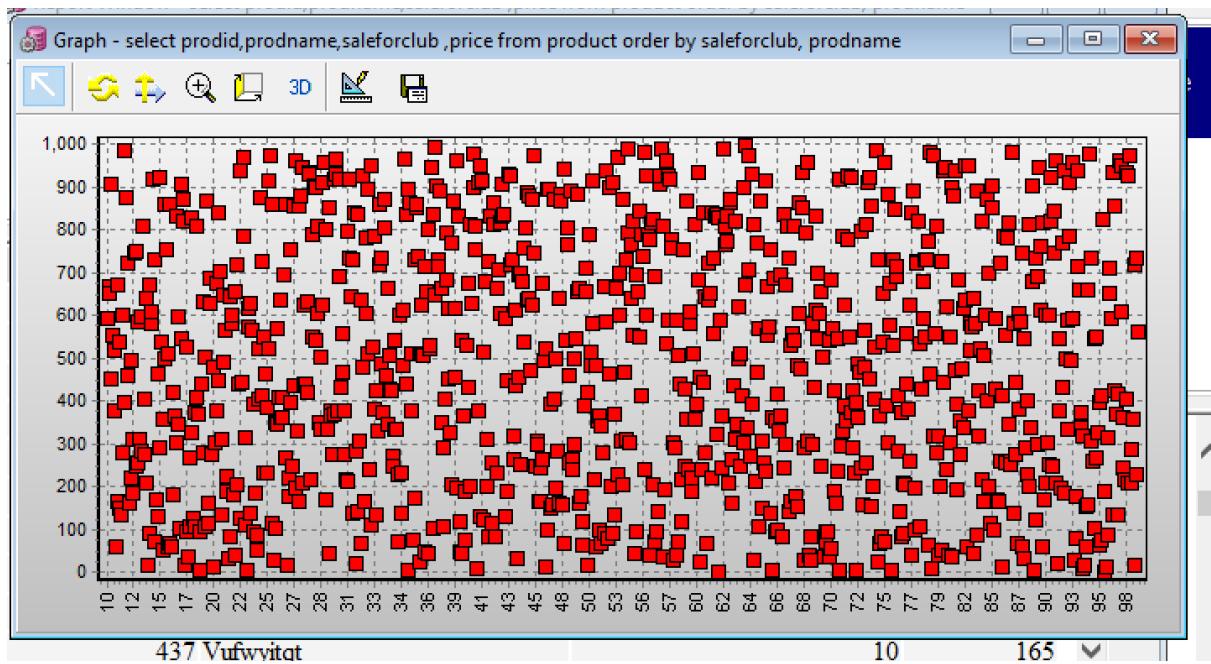
צילום הגדרות:



חלק מהדוח לדוגמא:

Prodid	Prodname	Saleforclub	Price
467	Bkbckvrus	10	595
443	Eooswfazn	10	669
594	Fevhjrkao	10	653
917	Fsfmnlcpx	10	906
696	Ihttztvjk	10	451
371	Ngvrjplo	10	554
469	Nwgpuurvx	10	377
660	Poiqlpuvv	10	517
710	Rhuzgftdh	10	57
437	Vufwyitqt	10	165
21	Xyplufprl	10	670
125	Yeqstruuq	10	539
104	Zjnlkbwxg	10	148
38	Zznwwnxka	10	133
			6434

ציר X אחזוֹן הנחה ציר Y מחיר מקורי



-2 דוח

```
select *
from stamp
where year_created>1970
order by year_created
```

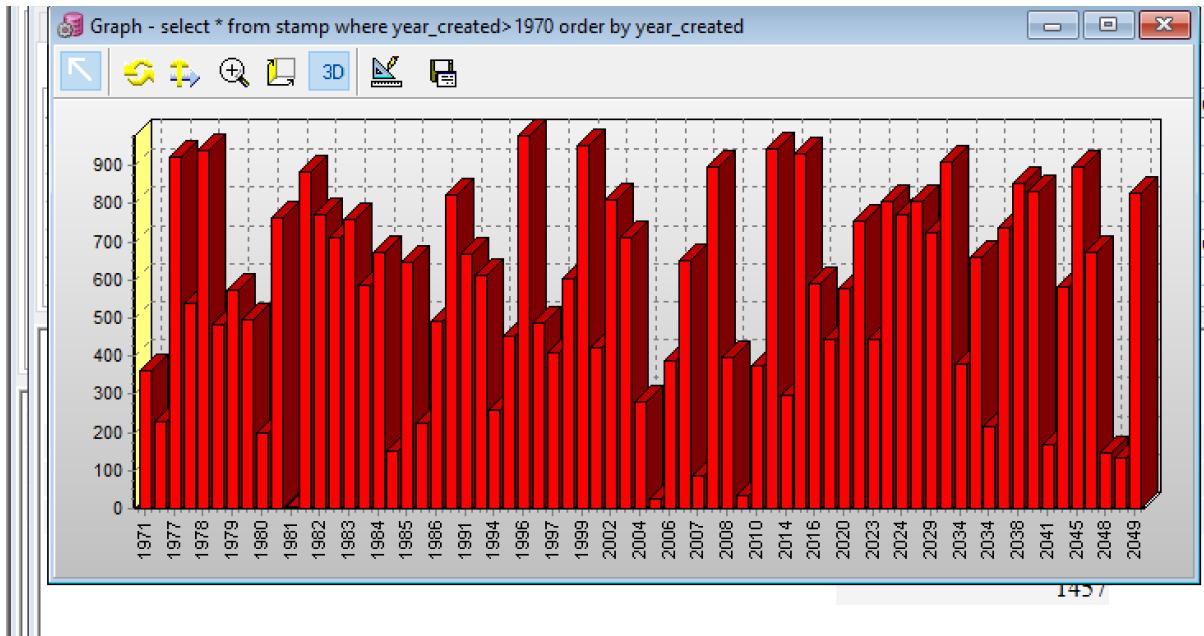
以色列的汇率：

SQL Layout Options								
<input checked="" type="checkbox"/> Auto Update								
Item	Description	Style	Header	Align	Format	Break	Sum	
<input checked="" type="checkbox"/> STAMP_ID		▼	▼	▼	
<input checked="" type="checkbox"/> STAMP_NAME		▼	▼	▼	
<input checked="" type="checkbox"/> YEAR_CREATED		▼	▼	Break	▼	
<input checked="" type="checkbox"/> STAMP_PRICE		▼	▼	▼	▼	
<input checked="" type="checkbox"/> AMOUNT_IN_STOCK		▼	▼	▼	Sum per Break	▼
<input checked="" type="checkbox"/> S_PERSON_ID		▼	▼	▼	▼	

דוגמה של חלק מהגרף שבו יש יותר משורה אחת:

628 platinum	1980	475	496	400
76 polonium	1980	119	199	81
165 rhenium	1980	846	762	23

ציר X שנה ציר Y כמות במחסן. שימוש לב שזה לא לכל השנה ביחיד אם יש כמות בולימית מסוימת בשנה. למשל ב-1980 כמו שראינו.



חלק 9- פונקציות וטריגרים:

פונקציה 1- לכל עובד שסידר יותר מ-X מוצרים היא מוסיף בונו י. מחזיר בסוף לכמה הוא שינה.

```
create or replace function organizedEnoughBonus(organizeAmount in number,
addToSalary in number)
```

```
return number is
```

```
Result number;
```

```
countHowManyChanged number;
```

```
cursor c1 is
```

```
select empid,count(*) as count
```

```
from organize
```

```
group by empid;
```

```
begin
```

```
countHowManyChanged:=0;
```

```
FOR row in c1 LOOP
```

```
if row.count>organizedAmount then
```

```
--must exist
```

```

update employee set salary=salary+addToSalary
where e_person_id=row.empid;
countHowManyChanged:=countHowManyChanged+1;
end If;
end LOOP;
Result:=countHowManyChanged;
return(Result);
end organizedEnoughBonus;

```

נראה דוגמה של הרצה:

```

Test Window - Script for function ORGANIZEDENOUGHTBONUS@XE
Test script DBMS Output Statistics Profiler Trace
1 begin
2   -- Call the function
3   :result := organizedenoughbonus(organizeamount => :organizeamount,
4                                   addtosalary => :addtosalary);
5 end;

```

	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Float	19
<input checked="" type="checkbox"/>	organizeamount	Float	2
<input checked="" type="checkbox"/>	addtosalary	Float	3
*			

1:1 SYSTEM@XE Executed in 0.016 seconds

נראה שבאמת זה מה שהוא-Amor לkerot: באמצעות Amor להשתנות 19 אנשים:

```

SQL Window - checkfuncupdate.sql
select * from organize
select empid, count(*) as count
from organize
group by empid
having count(*) > 2

```

EMPID	COUNT
100	4

19 rows selected in 0.204 seconds

הוסףנו בהמשך לפונקציה: exceptions:

```
create or replace function organizedBonusWithException(organizeAmount in
number, addToSalary in number)

return number is

Result number;

--dont use declare!!!

countHowManyChanged number;

cursor c1 is

select empid,count(*) as count
from organize
group by empid;

addIsNegative EXCEPTION;

pragma
exception_init(addIsNegative,-2292);

begin

if(organizeAmount<0 or addToSalary<0)
then raise addIsNegative;
END IF;

countHowManyChanged:=0;

FOR row in c1 LOOP
--begin
if row.count>organizeAmount then
--must exist
update employee set salary=salary+addToSalary
where e_person_id=row.empid;
countHowManyChanged:=countHowManyChanged+1;
end if;
end LOOP;

Result:=countHowManyChanged;
return(Result);

EXCEPTION
WHEN addIsNegative THEN
```

```

DBMS_OUTPUT.PUT_LINE('organizeAmount or addToSalary is negative');

--dont close exception!!!

result:=0;

return(Result);

end organizedBonusWithException;

```

הסביר: אם צריך להוריד משכורת למי שעבד מספיק או צריך לבדוק מי סידר מספר שלילי של מוציאים צריך לזרוק exception ולחדרס שיש טעות כדי שלא יעשה את שאר הקוד. ולהחזיר שיעדכ ל0 אנשים.

ובאמת בהרצה של

Variable	Type	Value
result	Float	0
organizeamount	Float	-3
addtosalary	Float	5

חזר 0 ומתקבלת הדפסת הטעות

פונקציה 2-

קצת יותר פשוטה, עשינו קודם כדי להבין איך עובד.

מתוך PARCEL תבחר עמודה כלשהי לעשות מנתה מקסימום או מינימום. עובד רק כי כל העמודות בטבלה מסווג מספר. אם הכנסנו שם לא טוב לעמודה או לא MAX\MIN צריך שגיאת מערכת.

```

CREATE OR REPLACE FUNCTION findMinMax(minOrMax IN VARCHAR,
columnName IN VARCHAR)
    RETURN NUMBER
IS
    Result NUMBER;
BEGIN
    EXECUTE IMMEDIATE 'SELECT ' || minOrMax || '(' || columnName || ')' FROM
parcel' INTO Result;

    RETURN Result;
END findMinMax;

```

שימוש לב שבביל לבצע את הקוד צריך לחבר **STRINGS** עם הפרמטרים ולבצע את השאלה כבר בתור `max\min` ואז לבצע אותה ואת התשובה להכניס כפרמטר לvariable RESULT שוחזר בסוף.

וכשMRIיצים מתקובל

	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Float	999
<input checked="" type="checkbox"/>	minormax	String	max
<input checked="" type="checkbox"/>	columnname	String	PACKAGE_PRICE

נרים את השאלה חיצונית לפונקציה:

	MAX(PACKAGE_PRICE)
1	999

פרוצדורה 1 - לכל ציר תדף מ- הבולים שהוא ציר. יש פה לולאה בתוך לולאה ואין צורך להגדיר cursor כי עושה את הלולאה על מה שחוזר מהשאילה כבר.

create or replace procedure printStampForArtist is

a_person_id artist.a_person_id%TYPE;

BEGIN

-- Loop through each artist

FOR artist_rec IN (SELECT * FROM artist) LOOP

a_person_id := artist_rec.a_person_id;

DBMS_OUTPUT.PUT_LINE('Artist: ' || a_person_id);

-- Loop through each stamp for the current artist

FOR stamp_rec IN

(SELECT * FROM stamp WHERE a_person_id = stamp.s_person_id)

LOOP

DBMS_OUTPUT.PUT_LINE('Stamp: ' || stamp_rec.stamp_name);

END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');

END LOOP;

end printStampForArtist;

הנה מה שמודפס ב-test

(כי זה פרוצדורה לא חוזרת שום דבר)

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled

-----
Artist: 904
Stamp: gallium
Stamp: boron
Stamp: praseodymium
Stamp: terbium
Stamp: cerium
Stamp: carbon
Stamp: europium
Stamp: curium
Stamp: iridium
Stamp: bismuth
Stamp: neptunium
Stamp: fluorine
Stamp: tellurium
-----
Artist: 998
Stamp: dysprosium
Stamp: holmium

```

183:1 SYSTEM@XE Executed in 0.015 :

לכל צייר הוא מופיע כמות שונה ושמות שונים לפי מה שהוא צייר. אבל גם אין הרבה ציירים.

פרוצדורה 2-

נרצה להציג לכל מחיר שהתקבל איזה בולים הוא צריך בשבייל זה.

כדי לפשט את האלגוריתם נציין לפי הסדר. בוחר את הבול הכי זול ובודק אם יש לנו מספיק כרגע (לוקח אחד מכל סוג) אונמנם זה לא אלגוריתם מדויק אבל הוא יותר פשוט כרגע ו לוקח פחות זמן לחשב.

זה מה שמודפס בבדיקה:

```
test script | Output | STATISTICS | Profiler | trace |  
Clear Buffer size 10000   Enabled  
we also need stamp flower with price of 1  
we also need stamp Home with price of 4  
you paid 5
```

כמובן שלא תמיד ימצא בדיק למשל אם נשים 20 הוא יסכם עד 28 כי זה מה שיש לנו.

```
Clear Buffer size 10000   Enabled  
we also need stamp flower with price of 1  
we also need stamp Home with price of 4  
we also need stamp rhenium with price of 4  
we also need stamp Banana with price of 5  
we also need stamp nickel with price of 14  
you paid 28
```

```
create or replace procedure findWhichStamps(amount in number) is  
cursor c1 is  
select stamp_price,stamp_name  
from stamp  
order by stamp_price;  
sumUntilNow number;  
numberOfStamps number;  
begin  
sumUntilNow:=0;  
for r in c1 loop  
numberOfStamps:=numberOfStamps+1;  
sumUntilNow:=sumUntilNow+r.stamp_price;
```

```

DBMS_OUTPUT.PUT_LINE('we also need stamp'||r.stamp_name||' with price of
'||r.stamp_price);

if(sumUntilNow>=amount) then
    exit;
end if;
end Loop;

DBMS_OUTPUT.PUT_LINE('you paid'||sumUntilNow);
--amount:=numberOfStamps;
end findWhichStamps;

```

יצירת טריגרים:

-1

```

SET SERVEROUTPUT ON;
create or replace trigger addArea
    before update on mail
    for each row
declare
    -- local variables here
    value_exists NUMBER;
begin
    SELECT COUNT(*)
        INTO value_exists
        FROM area
        WHERE area_id = :new.m_area_id;

    IF value_exists != 0 THEN
        DBMS_OUTPUT.PUT_LINE('updated someone');
    END IF;
    IF value_exists = 0 THEN
        insert into area(area_id,area_name) values(:new.m_area_id,'random name');
        DBMS_OUTPUT.PUT_LINE('value does not exist');
    END IF;

```

```
end addArea;
```

אם רוצים לשנות במייל איזור לאיזור שלא קיימ תיצור את האיזור.

נכשע את הפקודות הבאות

```
select * from mail where mail_id=100;
```

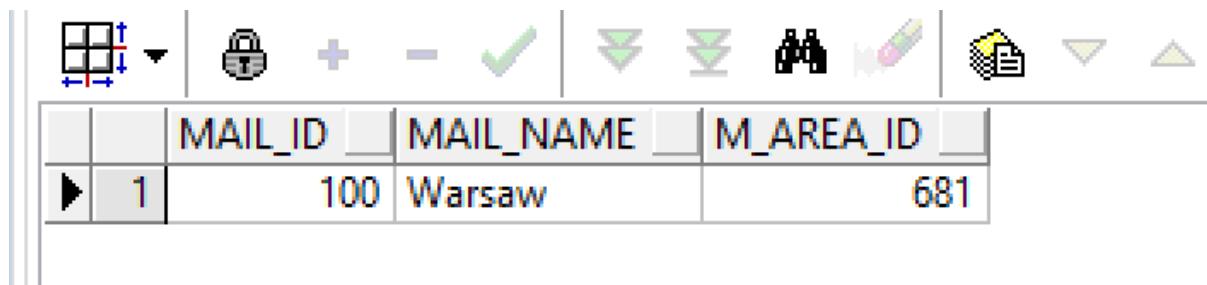
```
select * from area where area_id=999;
```

update mail

```
set m_area_id=999
```

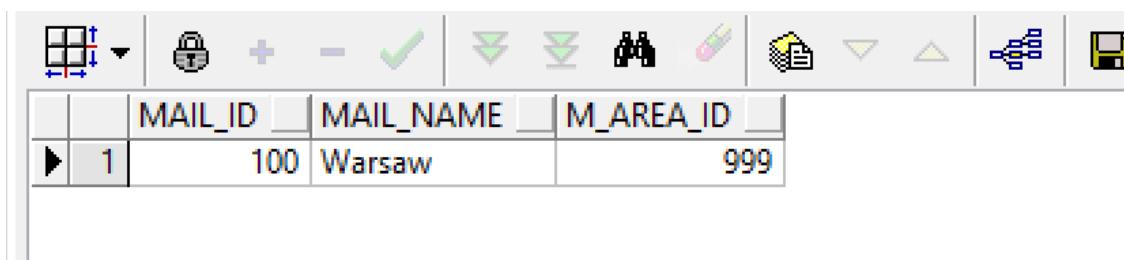
```
where mail_id=100
```

נקבל שהמייל קיים והאיזור לא.



	MAIL_ID	MAIL_NAME	M_AREA_ID
▶	1	100	Warsaw

לאחר הרצתה הגדיל על המайл נקלט במייל במייל עם ה-ID הנל את



	MAIL_ID	MAIL_NAME	M_AREA_ID
▶	1	100	Warsaw

זה מה שרצינו.

2-אם אדם עשה מספיק קניות תוסף לו .INCLUB

```
CREATE OR REPLACE TRIGGER trg_update_inclub
```

```
FOR INSERT ON buy
```

```
COMPOUND TRIGGER
```

```
-- Define a package-level collection to store customer IDs
```

```
TYPE customer_ids_type IS TABLE OF buy.custid%TYPE;
```

```
customer_ids customer_ids_type := customer_ids_type();
```

```
-- Before Each Row Section
BEFORE EACH ROW IS
BEGIN
    -- Store the customer ID in the collection
    customer_ids.EXTEND;
    customer_ids(customer_ids.LAST) := :new.custid;
END BEFORE EACH ROW;

-- After Statement Section
AFTER STATEMENT IS
BEGIN
    -- Iterate over the customer IDs in the collection
    FOR i IN 1..customer_ids.COUNT LOOP
        DECLARE
            v_purchase_count NUMBER;
        BEGIN
            -- Count the number of purchases for the customer
            SELECT COUNT(*) INTO v_purchase_count
            FROM buy
            WHERE custid = customer_ids(i)
            GROUP BY custid;

            -- Update the inclub status for the customer if purchase count exceeds 5
            IF v_purchase_count > 5 THEN
                UPDATE Customer
                SET inclub = 'YES'
                WHERE custid = customer_ids(i);
            END IF;
        END;
    END LOOP;
END;
```

```

END LOOP;

-- Clear the collection
customer_ids.DELETE;
END AFTER STATEMENT;

```

```
END trg_update_inclub;
```

למה הקוד הזה מסובך? כי אם אנחנו מנסים להשתמש במידע מתוך BUY כשאנחנו בתוך טריגר על BUY המחשב עשוה שגיאה, אז נצטרך לשמר את המידע מראש לפני ולהשתמש בו אחריו. געשה את זה עם טריגר מסווג COMPOUND שמאפשר לעשות משהו גם לפני וגם אחרי פעולה. יצרנו טבלה customer_id ושמרנו בה בכל פעם את הID הנוכחי. ואז אחרי INSERT בודקים אם לכל אחד מהלകחות בטבלה החדשה וביניהם זה ששיינו הרצע, אם יש להם מספיק שורות בBUY.

נרים את השורה

```
;select * from customer where custid=799
```

נקבל

	CUSTID	INCLUB	C_MAIL_ID
▶	1	799	NO

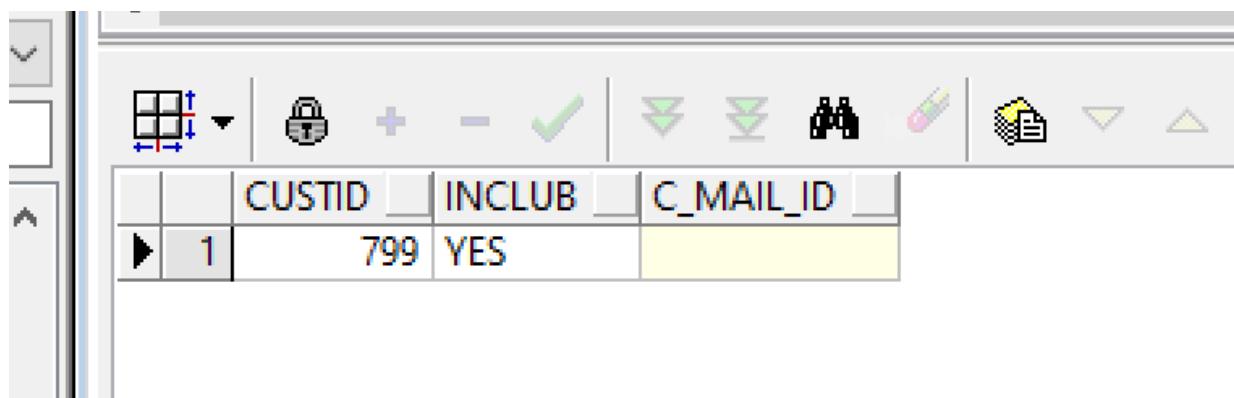
נפנה אדם רנדומלי כי יש יותר מדי קניות בטבלה BUY

```
;delete from buy where bid=889
```

עשינו כניסה קניה נוספת לעטן עם הליקו 799

```
insert into buy values(TO_DATE('04/11/1989', 'DD/MM/YYYY'),889,151,799)
```

ונראה שנוסף לו CLUB מועדי

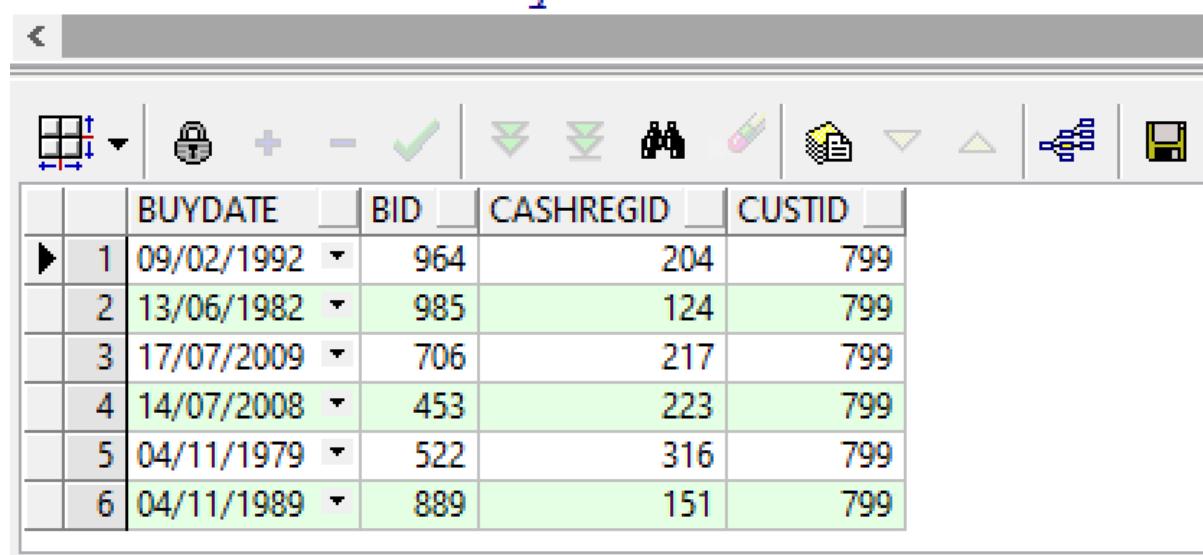


A screenshot of a database management system interface. At the top, there's a toolbar with various icons for operations like insert, delete, update, and search. Below the toolbar is a table with three columns: CUSTID, INCLUB, and C_MAIL_ID. A single row is selected, showing values 1, 799, and YES respectively.

	CUSTID	INCLUB	C_MAIL_ID
▶	1	799	YES

כִּי הוּא עֲשָׂה מַסְפֵּיק קְנִיתָ

```
select * from buy where custid=799
```



A screenshot of a database management system interface. At the top, there's a toolbar with various icons for operations like insert, delete, update, and search. Below the toolbar is a table with four columns: BUYDATE, BID, CASHREGID, and CUSTID. Six rows are listed, all corresponding to CUSTID 799. The rows show different dates and BID values.

	BUYDATE	BID	CASHREGID	CUSTID
▶	09/02/1992	964	204	799
2	13/06/1982	985	124	799
3	17/07/2009	706	217	799
4	14/07/2008	453	223	799
5	04/11/1979	522	316	799
6	04/11/1989	889	151	799