

מערכת דואר-חבילות

מגישות:

מוריה מזרחי 212319693

הודיה אבידן 213841505

2 תוכן עניינים
14 מבוא

מבוא:

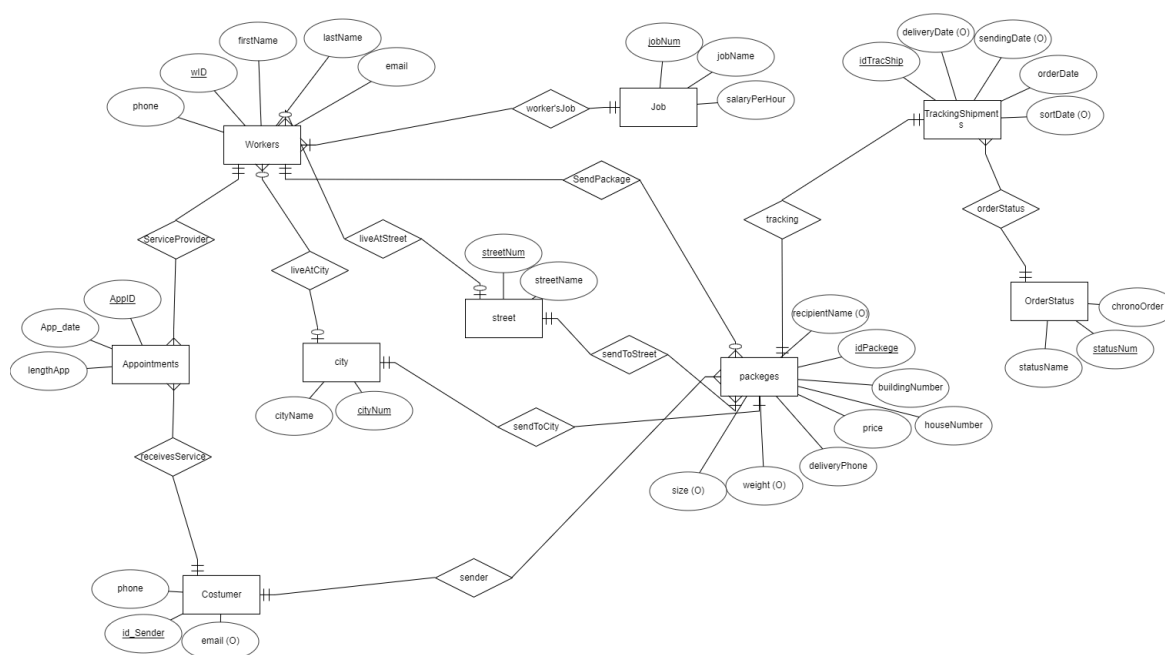
מערכת הדואר עיסוקה העיקרי יהיה על שילוח חבילות, כל המערכת תעסוק סביב זה. המערכת תשמור נתונים על **עובדים**, כמו שם פרטי, משפחה, ת.ז (שגם יהיה המפתח), כתובת וכו'...

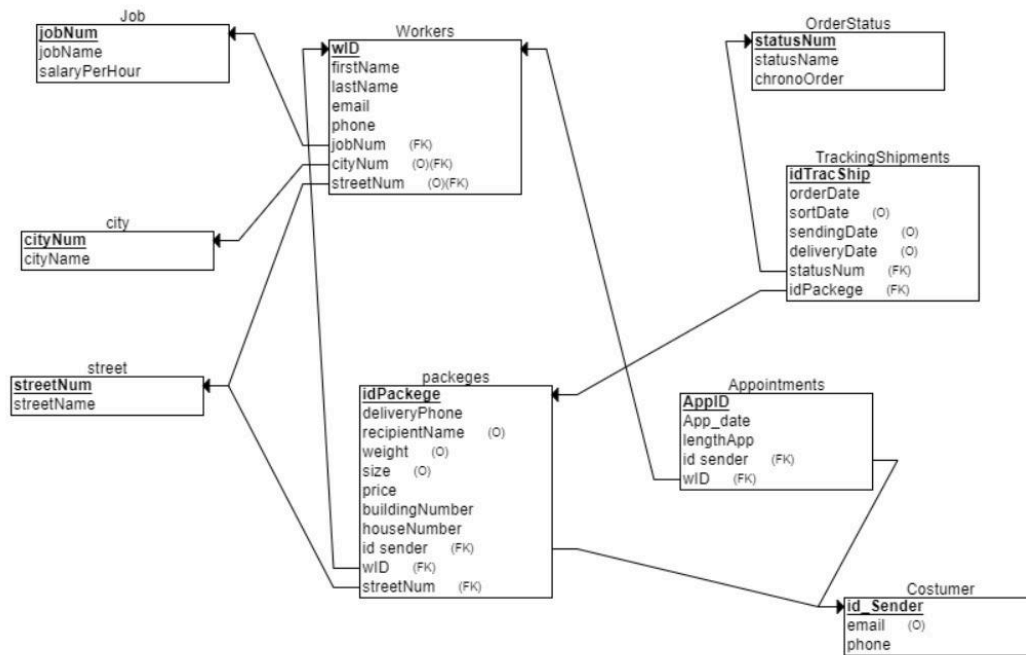
כמו כן המערכת תשמור נתונים על **לקוחות** ששולחים את החבילות. כמובן שהמערכת תשמור נתונים על **החבילות** עצמן, המידע על הלקוחות שמקבלים את החבילות ישמר בטבלת החבילות.

במידה ומקבל החבילה רוצה להחזיר את החבילה או לשלוח חבילה חדשה הוא נרשם כלקוח (עד עכשיו הפרטים שלו הופיעו רק בטבלת החבילה) ואותה חבילה נרשמת כחבילה חדשה ופרטי המקבל יהיו בטבלת החבילה.

במידה וכתובת הנמען והטלפון לא נכונים (מופיעים בחבילה), הדואר ירשום את החבילה כחבילה חדשה עם פרטי הלקוח ששלח כפרטי מקבל המשלוח.

בנוסף יהיה נתונים על **מעקב אחר חבילות וזימון תורים** עבור לקוחות שמוסרים חבילות.





עבור טבלת הנתונים של WORKERS פתחנו עוד טבלה JOB עבור תפקיד העובד, בטבלה זו יהיה מפורט כל תפקידי העובדים שעובדים בדואר הרלוונטיים למערכת זו, כמו פקיד דואר, שליח, ממין וכו'. טבלה זו תהיה מקושרת עם טבלת WORKERS ותסמן את תפקיד העובד.

עבור טבלת הנתונים TRACKINGSHIPMENTS פתחנו עוד טבלה ORDERSTATUS, טבלה זו תפרט את מיקום ההזמנה. טבלה זו תהיה מקושרת עם טבלת TRACKINGSHIPMENTS ותעדכן כל פעם את מיקום החבילה העדכני.

בנוסף פתחנו עוד טבלאות CITY, STREET שהתחברו עם טבלאות שמופיעות בהן כותבות כמו עובדים וחבילות.

עבור חבילה יהיה מקושר לקוח, שולח החבילה ועובדים שמתעסקים עם החבילה.

תיאור הקשרים בדיאגרמה:

:Worker's job

קשר המבטא את תיאור התפקיד של העובד על פי קוד תפקיד.

כל עובד חייב להיות משויך לתפקיד אחד, לעומת זאת אין כל הגבלה על התפקידים, יתכן וכמה עובדים יעבדו בתפקיד אחד, ויתכן תפקיד ללא עובדים כלל.

:Service provider

קשר המבטא עובד הנותן שירות בתור בדואר. כל תור חייב להיות משויך לעובד אחד, אך אין כל הגבלה על העובדים, יתכן

ועובד יטפל בכמה תורים ויתכן שלא יטפל בתורים כלל.

: receives service

קשר המבטא לקוח המקבל שירות בתור בדואר. כל תור חייב להיות משויך ללקוח אחד, אך אין כל הגבלה על הלקוחות, יתכן

ולקוח יזמין כמה תורים, ויתכן ולקוח לא יזמין תור כלל. (ישלח חבילה ללא תור)
Sender :

קשר המבטא שליחת חבילה על ידי לקוח. כל חבילה שייכת בהכרח ללקוח אחד בדיוק. לא יתכן חבילה שנשלחה על ידי כמה

לקוחות, אך יתכן שאותו לקוח שלח מספר חבילות.
Tracking :

קשר המבטא מעקב אחר חבילה. כל חבילה חייבת שיהיה עבורה מעקב אחד בדיוק. לא יתכן יותר ממעקב אחד לחבילה ולא

יתכן יותר מחבילה אחת למעקב מסוים.
orderStatus :

קשר המבטא סטטוס חבילה על ידי קוד של סטטוס מסוים. כל חבילה הנמצאת במעקב חייבת שיהיה עבורה סטטוס אחד

ספציפי עדכני לזמן הנוכחי. לא יתכן חבילה במעקב ללא סטטוס, אך יתכן כמה חבילות עם אותו סטטוס ויתכן סטטוס ללא חבילות כלל.

```
CREATE TABLE Job
)
,jobNum NUMERIC(2) NOT NULL
,jobName VARCHAR(15) NOT NULL
,salaryPerHour FLOAT NOT NULL
(PRIMARY KEY (jobNum
;{(
```

```
CREATE TABLE Costumer
)
,id_sender NUMERIC(3) NOT NULL
,(email VARCHAR(15
,phone NUMERIC(10) NOT NULL
(PRIMARY KEY (id_sender
;{(
```

```
CREATE TABLE OrderStatus
)
,statusNum NUMERIC(3) NOT NULL
,statusName VARCHAR(15) NOT NULL
,chronoOrder NUMERIC(1) NOT NULL
(PRIMARY KEY (statusNum
;{(
```

```
CREATE TABLE city
)
,cityName VARCHAR(15) NOT NULL
,cityNum NUMERIC(3) NOT NULL
(PRIMARY KEY (cityNum
;{(
```

```
CREATE TABLE street
)
,streetNum NUMERIC(3) NOT NULL
,streetName VARCHAR(15) NOT NULL
(PRIMARY KEY (streetNum
;{(
```

```
CREATE TABLE Workers
)
,wID NUMERIC(3) NOT NULL
,firstName VARCHAR(15) NOT NULL
```

```
,lastName VARCHAR(15) NOT NULL
,email VARCHAR(15)
,phone NUMERIC(10) NOT NULL
,jobNum NUMERIC(2) NOT NULL
,(cityNum NUMERIC(3)
,(streetNum NUMERIC(3)
,(PRIMARY KEY (wID
,(FOREIGN KEY (jobNum) REFERENCES Job(jobNum
,(FOREIGN KEY (cityNum) REFERENCES city(cityNum
,(FOREIGN KEY (streetNum) REFERENCES street(streetNum
;{
```

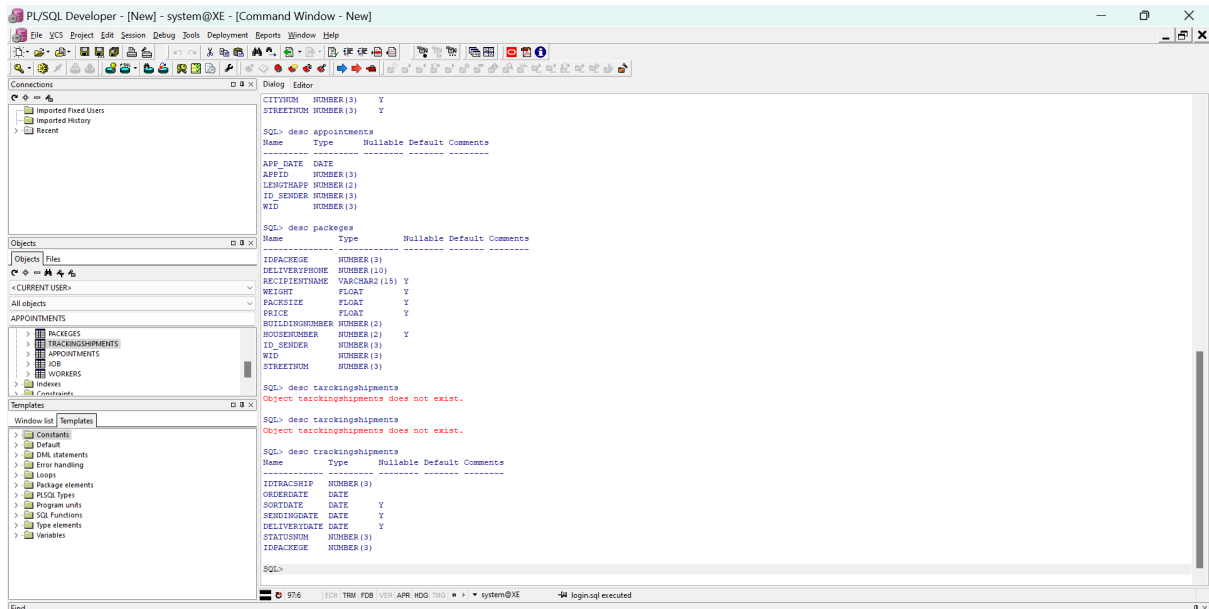
```
CREATE TABLE Appointments
)
,App_date DATE NOT NULL
,AppID NUMERIC(3) NOT NULL
,lengthApp NUMERIC(2) NOT NULL
,id_sender NUMERIC(3) NOT NULL
,wID NUMERIC(3) NOT NULL
,(PRIMARY KEY (AppID
,(FOREIGN KEY (id_sender) REFERENCES Costumer(id_sender
,(FOREIGN KEY (wID) REFERENCES Workers(wID
;{
```

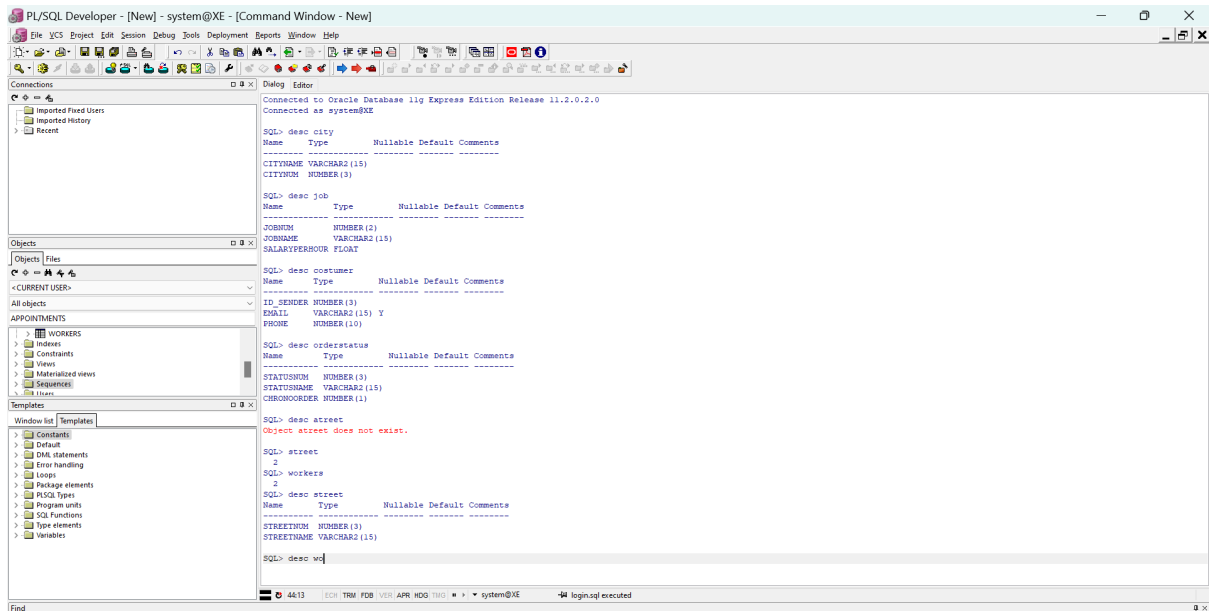
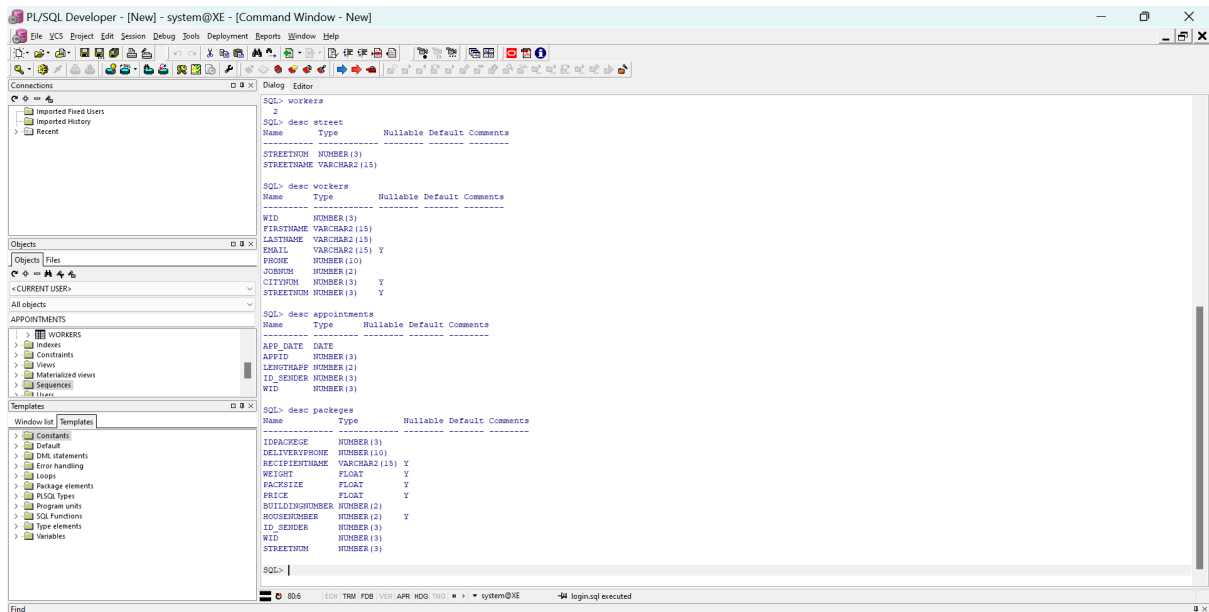
```
CREATE TABLE packageges
)
,idPackage NUMERIC(3) NOT NULL
,deliveryPhone NUMERIC(10) NOT NULL
,(recipientName VARCHAR(15)
,weight FLOAT
,packsize FLOAT
,price FLOAT
,buildingNumber NUMERIC(2) NOT NULL
,(houseNumber NUMERIC(2)
,id_sender NUMERIC(3) NOT NULL
,wID NUMERIC(3) NOT NULL
,streetNum NUMERIC(3) NOT NULL
,(PRIMARY KEY (idPackage
,(FOREIGN KEY (id_sender) REFERENCES Costumer(id_sender
,(FOREIGN KEY (wID) REFERENCES Workers(wID
,(FOREIGN KEY (streetNum) REFERENCES street(streetNum
;{
```

```
CREATE TABLE TrackingShipments
```

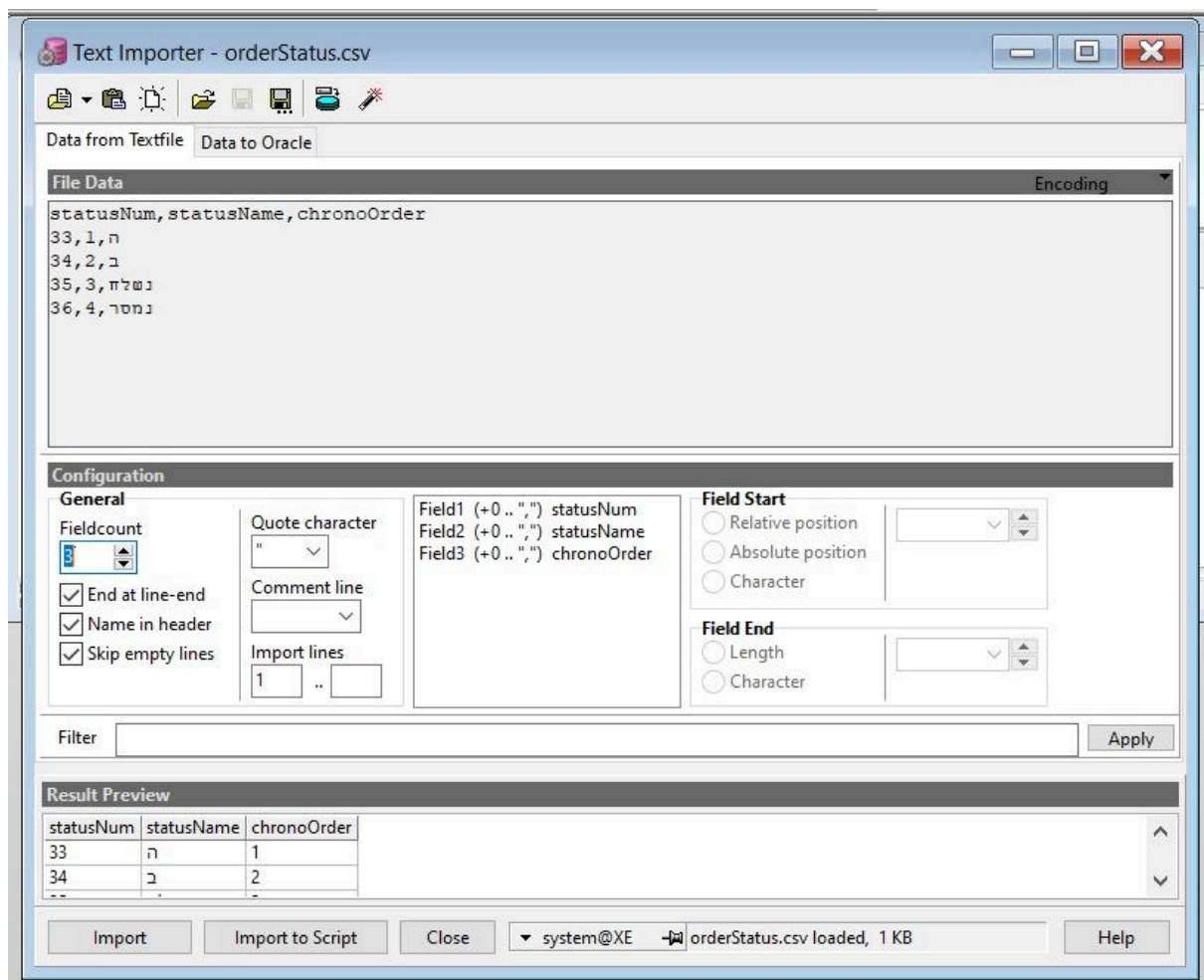
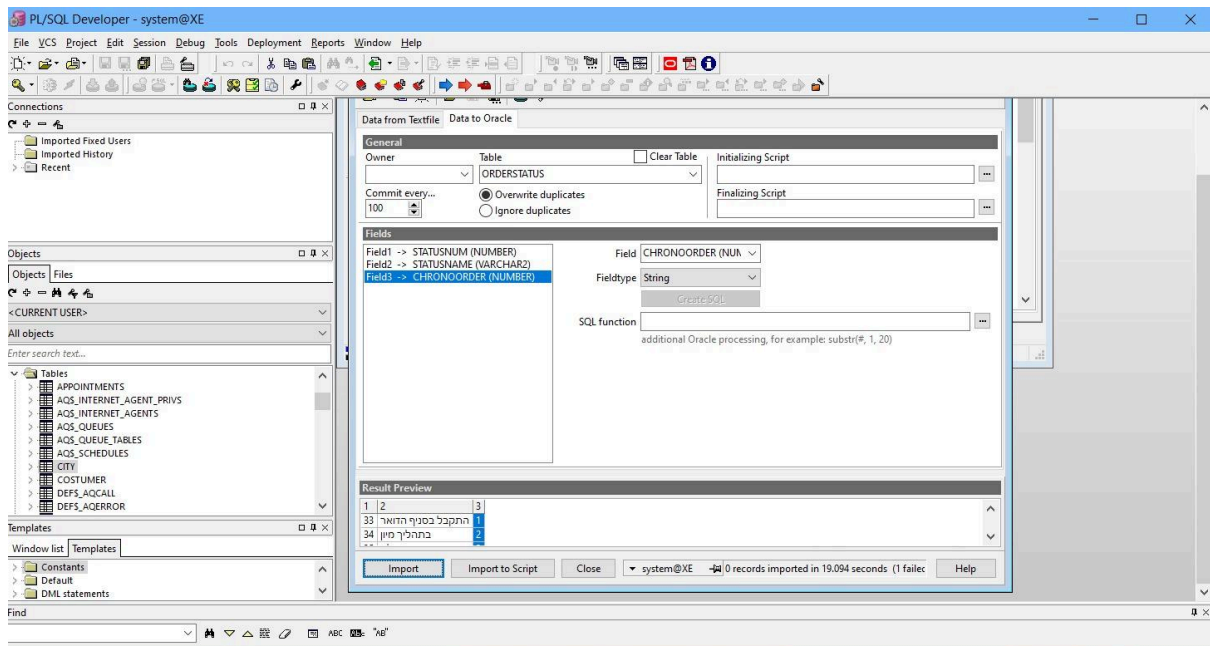
```
)  
,idTracShip NUMERIC(3) NOT NULL  
,orderDate DATE NOT NULL  
,sortDate DATE  
,sendingDate DATE  
,deliveryDate DATE  
,statusNum NUMERIC(3) NOT NULL  
,idPackege NUMERIC(3) NOT NULL  
,(PRIMARY KEY (idTracShip  
,(FOREIGN KEY (statusNum) REFERENCES OrderStatus(statusNum  
(FOREIGN KEY (idPackege) REFERENCES packeges(idPackege  
;(  

```

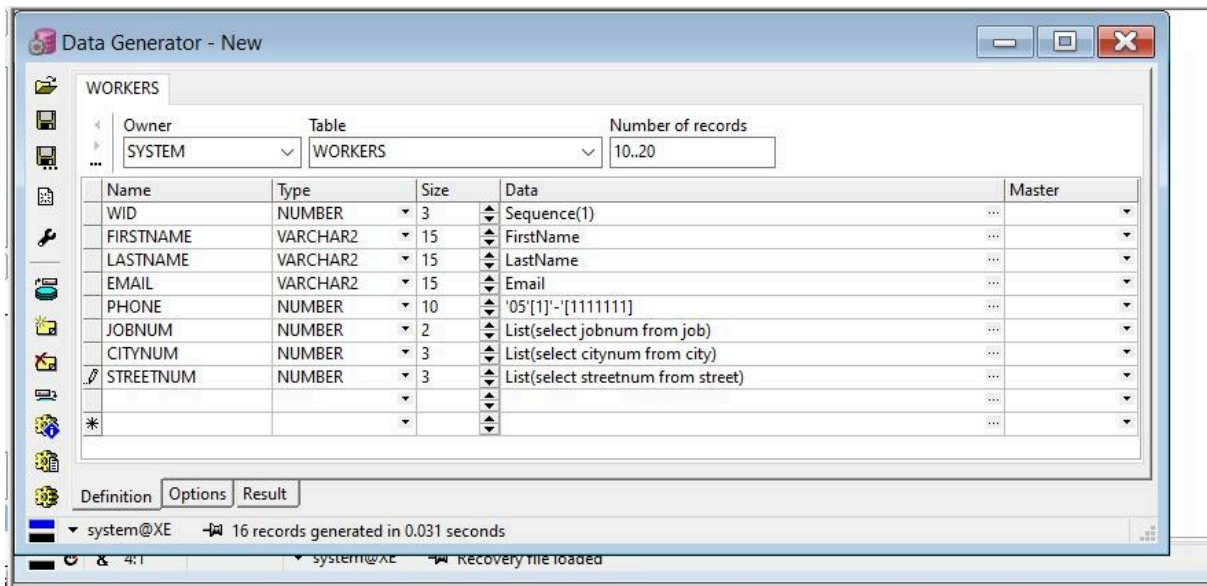





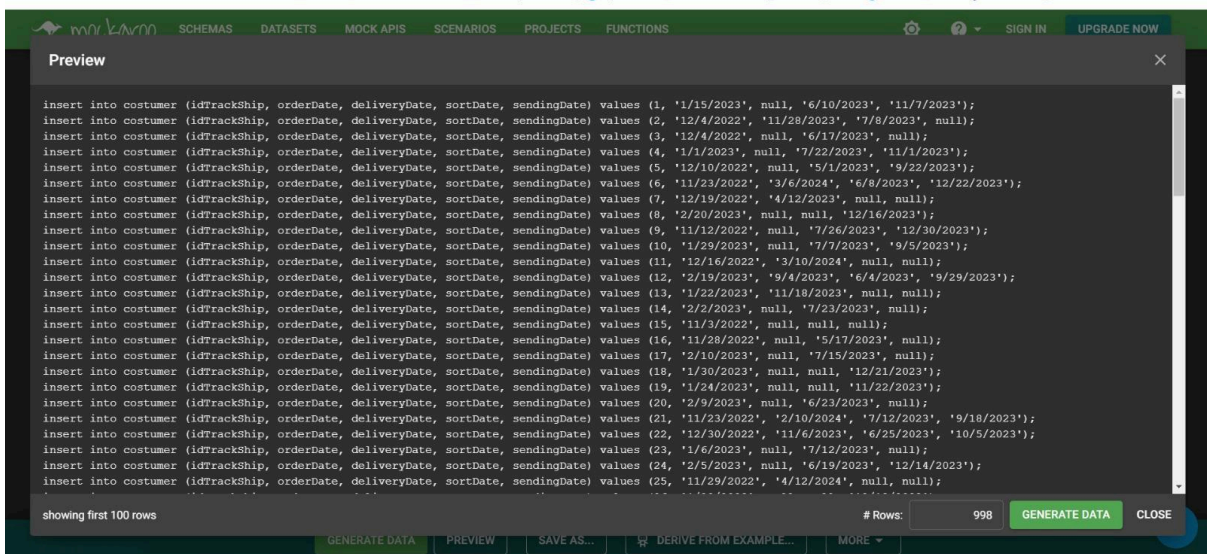
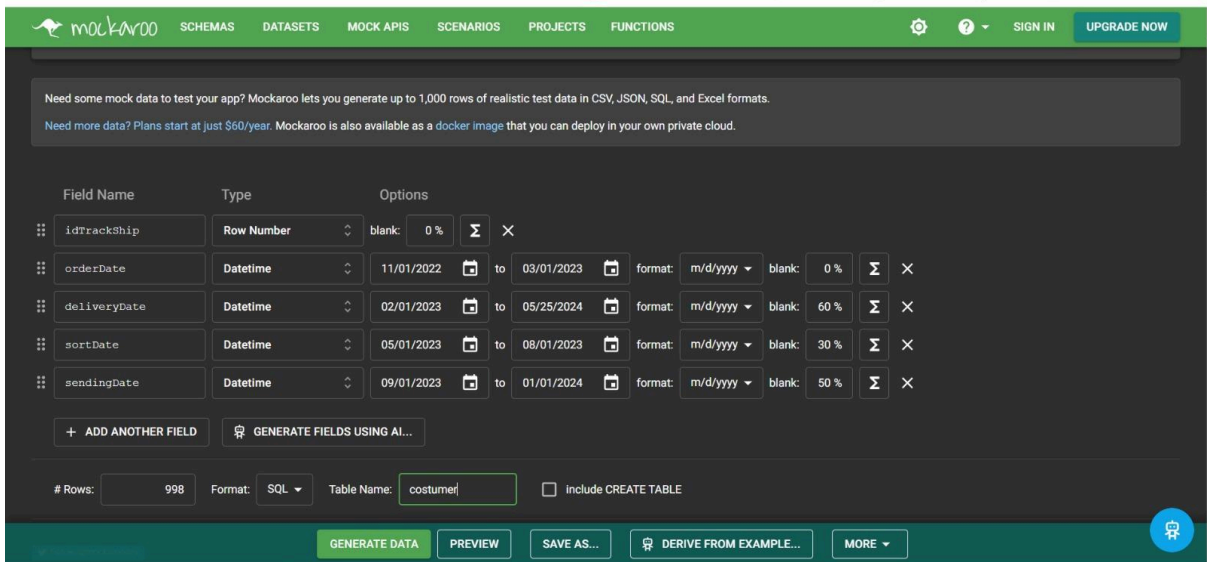
הכנסת הנתונים: 1. בחרנו להכניס נתונים על ידי קבצי CSV

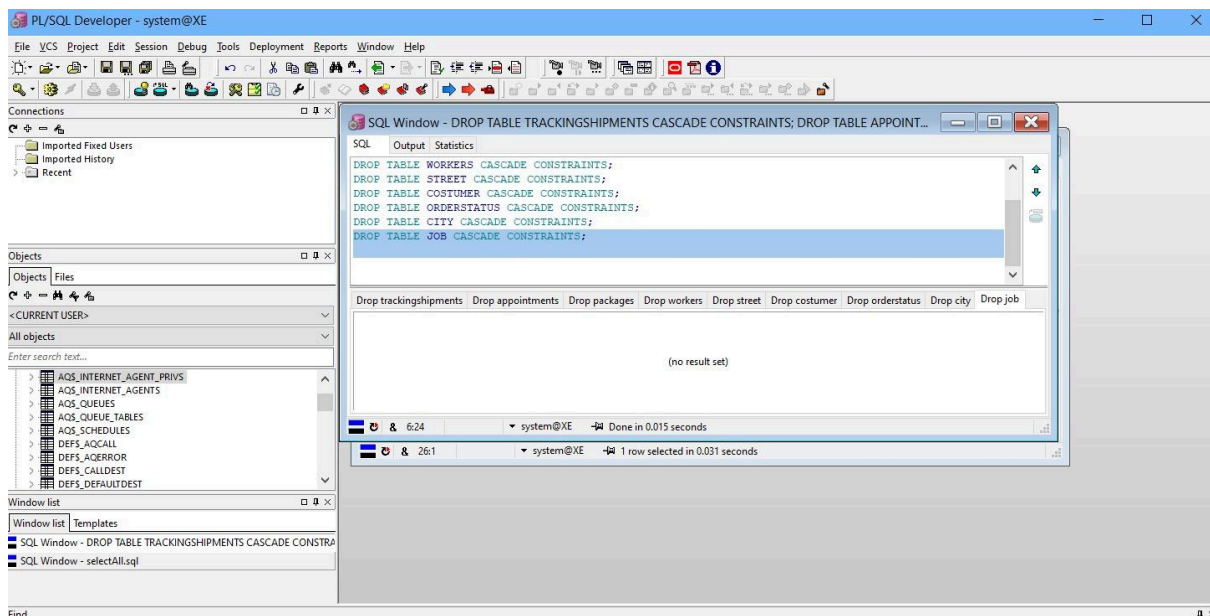
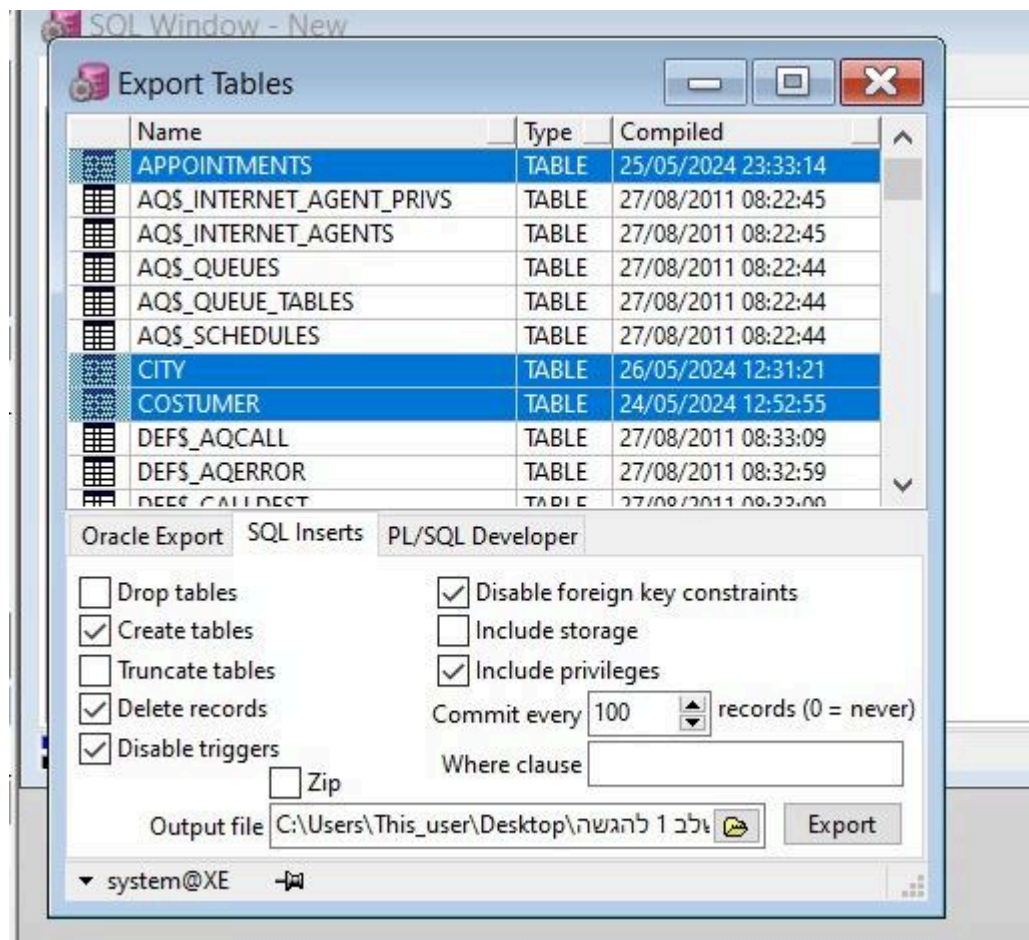


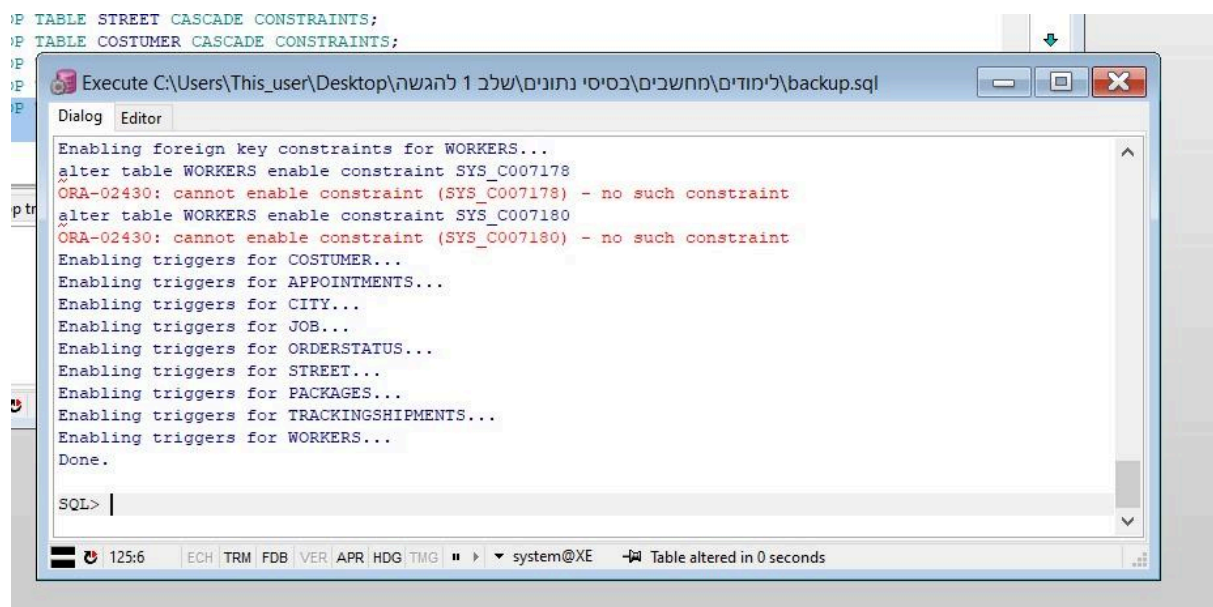
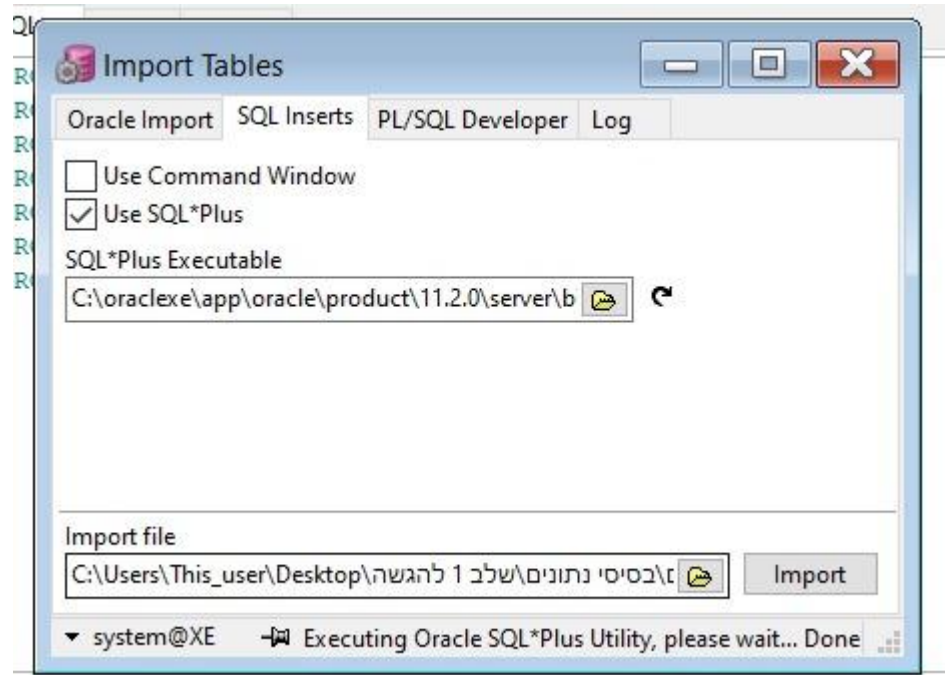
2. שיטת data generator



3.







שלב 2

תוספות שעשינו לפני השאילתות עצמם:

עדכון הסטטוס שיהיה תואם לתאריכים, אם אין תאריך מיון, אז שהסטטוס יהיה "התקבל", אם אין תאריך שילוח, הסטטוס יהיה "במיון" וכו'

```
--Correct status number update
update trackingshipments t
set t.statusnum=33
where t.orderdate is not null

update trackingshipments t
set t.statusnum=34
where t.sortdate is not null

update trackingshipments t
set t.statusnum=35
where t.sendingdate is not null

update trackingshipments t
set t.statusnum=36
where t.deliverydate is not null
```

עדכון חשוב נוסף: שהתאריכים יהיו בסדר כרונולוגי נכון, כלומר: לא יתכן שהתאריך של המסירה יהיה לפני תאריך המיון וכו' (עד עכשיו כן היה כך כי התאריכים הוגרלו רנדומלית) וכן לא יתכן שאין תאריך שילוח, אך יש תאריך מסירה וכן הלאה.

```
update trackingshipments
set sendingdate=null, deliverydate=null
where sortdate is null

update trackingshipments
set deliverydate=null
where sendingdate is null

update trackingshipments
set deliverydate=sendingdate+1
where sendingdate is not null and deliverydate<sendingdate

update trackingshipments
set sendingdate=sortdate+1
where sortdate is not null and sendingdate<sortdate
```

Queries:

search queries:

1 - פונקצית select:

שאלת חיפוש המציגה את רשימת כל העובדים (משלוחנים ופקידים) שביצעו פעילויות ב-6 החודשים האחרונים, עם סך הפעילויות לכל אחד מהם, סוג הפעילות ופרטי העובד.

```
WITH RecentAppointments AS (  
    SELECT w.wID, w.firstName, w.lastName, 'פקיד' AS activity_type, COUNT(a.AppID) AS activity_count  
    FROM Workers w  
    JOIN Appointments a ON w.wID = a.wID  
    WHERE a.App_date >= ADD_MONTHS(SYSDATE, -6) and jobnum=67  
    GROUP BY w.wID, w.firstName, w.lastName  
) ,  
RecentPackages AS (  
    SELECT w.wID, w.firstName, w.lastName, 'משלוח' AS activity_type, COUNT(p.idpackage) AS activity_count  
    FROM Workers w  
    JOIN packages p ON w.wID = p.wID  
    join trackingshipments t ON t.idpackage = p.idpackage  
    WHERE t.sendingdate >= ADD_MONTHS(SYSDATE, -6) and jobnum= 69  
    GROUP BY w.wID, w.firstName, w.lastName  
)  
SELECT *  
FROM RecentAppointments  
UNION ALL  
SELECT *  
FROM RecentPackages
```

תוצאת השאילתה:

	WID	FIRSTNAME	LASTNAME	ACTIVITY_TYPE	ACTIVITY_COUNT
1	71	Elle	Cusack	פקיד	1
2	76	Beth	Preston	פקיד	4
3	66	Joe	Shand	פקיד	4
4	2	Emily	Gilliam	פקיד	3
5	57	Sean	Hirsch	פקיד	3
6	89	Cherry	Scott	פקיד	2
7	60	Martin	Hornsby	פקיד	2
8	30	Barry	Coyote	פקיד	4
9	86	Alex	Oldman	פקיד	3
10	91	Anthony	Minogue	פקיד	4
11	34	Hal	Warwick	פקיד	5
12	12	Fats	Rodriguez	פקיד	6
13	18	Debby	Napolitano	פקיד	3
14	31	Joanna	Franklin	פקיד	5
15	77	Keith	Phifer	פקיד	3
16	29	Cathy	Gooding	פקיד	2
17	73	Allan	Wolf	שליח	3
18	33	Caroline	Gary	שליח	3
19	47	Rik	Baldwin	שליח	5
20	70	Jeffrey	Pollak	שליח	7
21	6	Glen	Simpson	שליח	3
22	40	Ruth	Winter	שליח	2
23	51	Lenny	Venora	שליח	4

system@XE 23 rows selected in 0.016 seconds (more...)

זמן הרצה:

53 rows selected in 0.062 seconds

2- שאילתת select

שאילתה המחשבת את המשכורת הסופית עבור כל עובד ומציגה את פרטי העובדים ומשכורותיהם.
חישוב המשכורת נעשה באופן הבא:
משלוחן - מקבל שכר בסיס לפי שעה + 10 ש"ח עבור כל משלוח שביצע
עובדים אחרים - שכר לפי שעה.
לצורך שאילתה זו הוספנו טור של מספר שעות עבודה עבור כל עובד:

```
ALTER TABLE workers
ADD WorkHours float;

UPDATE workers
SET WorkHours = 0

update workers
set workHours=ROUND(DBMS_RANDOM.VALUE(50.0,180.0),2)
```

ואז ערכנו את החישוב:

```
select wid, firstname, lastname, bonus+j.hourorpackagesalary*w.workhours+10*(
    select count(p.idpackage)
    from packages p join trackingshipments t on p.idpackage=t.idpackage
    where w.wid=p.wid
    and EXTRACT(YEAR from t.orderdate)=2024
    group by p.wid
) as finalSalary, j.jobname
from job j natural join workers w
where j.jobname='משלוחן'
UNION
select wid,firstname, lastname, w.bonus+j.hourorpackagesalary*w.workhours as finalSalary, j.jobname
from job j natural join workers w
where j.jobname!='משלוחן'
```

תוצאת השאילתה:

	WID	FIRSTNAME	LASTNAME	FINALSALARY	JOBNAME
1	1	Garry	Van Der Beek	12325.66	משלוחן
2	2	Emily	Gilliam	5735.384	פקיד
3	3	Amanda	Clarkson	8449.162	נהג
4	4	Natascha	Unger	9862.54	משלוחן
5	5	Patty	Costa	2660.098	נהג
6	6	Glen	Simpson	11868.8	משלוחן
7	7	Edgar	Gibbons	5094.798	נהג
8	8	Cary	Johansson	2234.775	ממין
9	9	Roger	Milsap	6435.434	ממין
10	10	Halle	Magnuson	4742.031	ממין
11	11	Larry	Branagh	7143.962	מנהל
12	12	Fats	Rodriguez	3144.84	פקיד
13	13	Liquid	Giraldo	4035.75	משלוחן
14	14	Marie	Mahoney	7752.386	נהג
15	15	Mika	Rain	6905.512	נהג
16	16	Graham	Ronstadt	4721.812	נהג
17	17	Renee	Carrington	9159.145	משלוחן
18	18	Debby	Napolitano	4030.134	פקיד
19	19	Mika	Kimball	10942.39	משלוחן
20	20	Kyra	Jay	3872.93	נהג

זמן הרצה:

100 rows selected in 0.094 seconds

3 - שאילתת select

שאילתה המספקת נתונים לפי רחובות חלוקה: מה מספר החבילות שחולקו ברחוב זה, מה המשקל הממוצע של החבילות, מה הגודל הממוצע של החבילה ומה הסכום הממוצע לכל חבילה באיזור זה. (השאילתה נועדה לעזור למנהלים לדעת האם באיזורים מסוימים יש יותר עומס של משלוחים ולדעת לפי זה כמה מחלקים לשלוח לכל איזור)

```
select streetname,avg_size,avg_weight,sum_price,num_packages
from street s natural join(
    select streetnum ,round(avg(packsize),2) as avg_size,round(avg(weight),2) as avg_weight,
        sum(price) as sum_price, count(*) as num_packages
    from street s
    natural join
        packages p
    natural join
        trackingshipments t
    group by streetnum)
```

תוצאת השאילתה:

	STREETNAME	AVG_SIZE	AVG_WEIGHT	SUM_PRICE	NUM_PACKAGES
1	גאולים	6.38	6.78	72.7	4
2	ארבל	19.67	9.83	131.8	6
3	באר אורה	34.25	7	96.8	4
4	בר	20.44	10.22	266.8	9
5	תאנה	28.55	10.18	221.3	11
6	אמיר	15.5	11.75	47.7	4
7	ארניה	28.71	10.86	162.7	7
8	אלרום	32	8.25	47.7	4
9	לילך	23	6.67	42.7	3
10	ורד	26	12.75	71.8	4
11	בר כוכבא	24.75	11	66.8	4
12	בראנד	19.36	10.3	168.6	7
13	הדר	17.75	0.25	30.9	2
14	פרחים	27.5	10.1	73.6	5
15	בדנר	38.4	12.2	73.6	5
16	אנקורים	12.75	9.5	105.9	4
17	רקפת	30.25	10	91.8	4
18	בנימין	22	3	35.9	2
19	נאגארה	2	0	5.9	1
20	איריס	19.59	8.37	293.6	11

זמן הרצה:

82 rows selected in 0.063 seconds

4 - שאילתת select

שאילתה המוצאת את הלקוחות VIP של המשרד. לקוח VIP זה לקוח שמספר החבילות שהוא שלח גדול שווה למספר החבילות הממוצע, וגם המחיר שהוא שילם עבור החבילות הוא גדול שווה למחיר החבילות הממוצע.

שאילתת חיפוש:

```
SELECT c.id_Sender,c.phone,c.email,p.total_packages,p.avg_package_price
FROM Costumer c
JOIN ( SELECT
      id_Sender,
      COUNT(*) AS total_packages, AVG(price) AS avg_package_price
      FROM packages
      GROUP BY id_Sender
    ) p ON c.id_Sender = p.id_Sender
WHERE
  p.total_packages > (
    SELECT AVG(package_count)
    FROM (
      SELECT COUNT(*) AS package_count
      FROM packages
      GROUP BY id_Sender
    )
  )
  AND p.avg_package_price > (
    SELECT AVG(subquery2.price)
    FROM (
      SELECT id_Sender, COUNT(*) AS package_count, AVG(price) AS price
      FROM packages
      GROUP BY id_Sender
    ) subquery2
    WHERE subquery2.package_count = p.total_packages
  );
```

תוצאה:

	ID_SENDER	PHONE	EMAIL	TOTAL_PACKAGES	AVG_PACKAGE_PRICE
1	93	535641491	patty@hps.pl	2	20
2	150	582962255	trink@irissoft	2	22.5
3	207	585616148	saule@swp.uk	2	30.45
4	242	546253666	rickie.farina@t	2	50
5	313	536323625	cuba.berenger@m	2	30.45
6	315	572227218	kstiers@extreme	2	20.45
7	325	575639176	vince.dalton@pr	2	20.45
8	345	539213257	fdelancie@tropi	2	30.45
9	378	516114758	forest@logistic	2	20.45
10	424	538451665	nick.royparnell	2	37.5
11	432	525942569	lea.crewson@sar	2	32.95
12	439	561415328	g.levert@daimle	2	27.95

זמן הרצה:

28 rows selected in 0.094 seconds

5 - שאלתת חיפוש

שאלתת ששולפת את שם העובד שקיבל הכי הרבה תורים יחד עם המידע של מספר התורים שהוא קיבל.

שאלתת עדכון:

```
WITH WorkerAppointmentCount AS (
    SELECT Wid, COUNT(AppID) AS totalAppointments
    FROM Appointments
    GROUP BY Wid
),
MaxAppointments AS (
    SELECT MAX(totalAppointments) AS maxAppointments
    FROM WorkerAppointmentCount
)
SELECT W.wID, W.firstName, W.lastName, COUNT(A.AppID) AS appointmentCount
FROM Workers W
JOIN Appointments A ON W.wID = A.wID
WHERE W.wID IN (
    SELECT wID
    FROM WorkerAppointmentCount WAC, MaxAppointments MA
    WHERE WAC.totalAppointments = MA.maxAppointments
)
GROUP BY W.wID, W.firstName, W.lastName
ORDER BY appointmentCount DESC;
```

תוצאה:

	WID	FIRSTNAME	LASTNAME	APPOINTMENTCOUNT
▶ 1	51	Lenny	Venora	18

אימות:

```
select *
from appointments
where wid=51
```

	APP_DATE	APPID	LENGTHAPP	ID_SENDER	WID
▶ 1	18/02/2023	141	5	400	51
2	27/08/2023	151	15	969	51
3	14/04/2024	179	15	15	51
4	15/03/2024	297	5	813	51
5	02/06/2023	320	15	306	51
6	08/02/2023	335	10	662	51
7	05/05/2024	345	5	468	51
8	13/06/2023	350	15	211	51
9	05/02/2024	375	10	207	51
10	14/05/2024	382	15	717	51
11	30/10/2023	524	5	149	51
12	25/04/2023	709	15	758	51
13	27/08/2023	717	10	720	51
14	14/07/2023	793	10	758	51
15	18/03/2024	794	15	878	51
16	28/07/2023	797	5	737	51
17	30/08/2023	879	10	880	51
18	04/10/2023	993	10	667	51

זמן הרצה:

1 row selected in 0.031 seconds

6 - שאילתת select

שאילתה שימושית מאד (עשינו אותה כתוספת כי היא מאד שימושית למרות שהיא לא כל כך מורכבת):
השאילתה מציגה את המשלוחים שתאריך השליחה שלהם היה בחודש האחרון אך הם עוד לא נמסרו.

צילום השאילתה:

```
SELECT
    p.wid, p.idPackage, p.recipientName, p.weight, c.id_Sender, c.email, c.phone, ts.orderdate, ts.sortdate, ts.sendingDate, ts.deliverydate
FROM packages p JOIN Costumer c ON p.id_Sender = c.id_Sender JOIN TrackingShipments ts ON p.idPackage = ts.idTracShip
WHERE ts.sendingDate >= ADD_MONTHS(SYSDATE, -1) AND ts.deliveryDate IS NULL
ORDER BY
    ts.orderdate DESC
```

תוצאה:

(כמובן שהפרש התאריכים לא מאד הגיוני אך בגלל שהוגרלו רנדומלית זה כך.
כמובן שעשינו בדיקה שתאריך המיון הוא אחרי תאריך הקבלה, ותאריך השילוח הוא אחרי תהליך המיון, ותאריך המסירה הוא אחרי תאריך השילוח.)

	WID	IDPACKAGE	RECIPIENTNAME	WEIGHT	ID_SENDER	EMAIL	PHONE	ORDERDATE	SORTDATE	SENDINGDATE	DELIVERYDATE
1	74	153	Dabney Carrack	18	91	sean.deejay@pri	592389933	27/11/2023	10/06/2024	11/06/2024	
2	22	283	Spike Williams	1	457	w.mckellen@zone	531431849	27/08/2023	23/05/2024	24/05/2024	
3	41	151	Lucinda Deejay	2	173	nicole.s@mainst	578562611	22/07/2023	07/06/2024	08/06/2024	
4	40	199	Jerry Aiken	12	730	morgan.guest@co	541439158	29/04/2023	09/07/2023	09/06/2024	
5	23	179	Earl Lerner	19	513	natasha.blosso	575138681	08/02/2023	29/05/2024	30/05/2024	
6	24	192	Swoosie Nivola	3	506	kathleen.rowlan	541153243	07/02/2023	02/06/2023	05/06/2024	

זמן ריצה:

6 rows selected in 0.031 seconds

update queries

1 - שאילתת עדכון

שאילתת הבדקת מי הם העובדים המצטיינים ומביאה להם בשכר בונוס של 500 שקל:
העובדים המצטיינים הן

העובד שחילק את מספר החבילות הגבוה ביותר, אם יש כמה שחילקו אותו מספר כולם יקבלו.
וכן העובד שקיבל את מספר התורים הגבוה ביותר, אם יש יותר מאחד כולם יקבלו.
הוספת הטור של בונוס בטבלת העובדים ואיפוסו בהתחלה -

```
ALTER TABLE workers
ADD bonus number(3);

UPDATE workers
SET bonus = 0
```

צילום לפני העדכון:

	WID	FIRSTNAME	LASTNAME	EMAIL	PHONE	JOBNUM	CITYNUM	STREETNUM	BONUS
50	50	Debby	Downie	debby.downie@pr	-8292506	71	9	34	0
51	51	Lenny	Venora	lenny.venora@vo	-5765629	69	5	74	0
52	52	Casey	Loeb	casey.loeb@roya	-1521836	69	6	33	0
53	53	Brian	Stone	brian.stone@inf	-8828322	70	5	36	0
54	54	Anna	McGovern	anna.mcgovern@o	-1272586	71	2	76	0
55	55	Cary	Stuerner	cary@sms.be	-7636095	69	10	29	0
56	56	Sinead	DiCaprio	sinead@signatur	-4375798	71	5	34	0
57	57	Sean	Hirsch	sean.h@usdairyp	-4684330	67	10	32	0
58	58	Johnette	Reiner	johnetter@prosp	-6643467	69	4	12	0
59	59	Kirsten	Carlisle	kirsten@esteela	-1143286	71	9	53	0
60	60	Martin	Hornsby	martin.hornsby@	-9746929	67	3	22	0
61	61	Joaquin	Travolta	j.travolta@sbc	-1751874	70	3	22	0
62	62	Candi	Hofstad	candi@cskbooks	-6147877	70	10	01	0

שאילתת עדכון:

```
UPDATE workers w
SET bonus = bonus + 500
where w.wid IN(
    select p.wid
    from packages p
    group by p.wid
    having count(*)>=ALL(
        select count(*)
        from packages p
        group by p.wid)
    )
or w.wid in (
    select a.wid
    from appointments a
    group by a.wid
    having count(*)>=ALL(
        select count(*)
        from appointments a
        group by a.wid)
    )
```

צילום לאחר העדכון: (של שני העובדים)

46	46	Judi	Trejo	judi.trejo@epiq	-5866947	69	10	30	0
47	47	Rik	Baldwin	rik.baldwin@mic	-3554445	69	4	46	0
48	48	Geoff	Steenburgen	geoff.steenburg	-3231537	70	6	49	0
49	49	Pamela	Leguizamo	pamelal@glaxosm	-8924469	70	1	6	0
50	50	Debby	Downie	debby.downie@pr	-8292506	71	9	34	0
51	51	Lenny	Venora	lenny.venora@vo	-5765629	69	5	74	500
52	52	Casey	Loeb	casey.loeb@roya	-1521836	69	6	33	0
53	53	Brian	Stone	brian.stone@inf	-8828322	70	5	36	0
54	54	Anna	McGovern	anna.mcgovern@o	-1272586	71	2	76	0
55	55	Cary	Stuerner	cary@sms.be	-7636095	69	10	29	0

95	95	Robby	Arden	r.arden@sourceg	-6886384	69	3	33	0
96	96	Arturo	McDowall	a.mcdowall@newt	-5581105	70	10	57	0
97	97	Cole	Peniston	cole.peniston@b	-9238483	70	9	2	0
98	98	Naomi	Wilson	naomi.wilson@ci	-5428457	69	10	22	0
99	99	Elizabeth	Chandler	elizabeth@cookt	-8292245	71	1	85	0
100	100	Brendan	Judd	brendan.judd@aq	-1221144	70	3	41	500

זמן הרצה -

2 rows updated in 0.015 seconds

2 - שאילתת עדכון

שאילתה המחפשת את הפקיד המצטיין (על פי מספר התורים שקיבל) ומקדמת אותו מדרגת פקיד לדרגת מנהל.

צילום של הפקיד לפני:

	JOBNUM	WID	JOBNAME
1	67	89	פקיד

שאילתת עדכון:

```
UPDATE workers w
SET w.jobnum = (
    SELECT jobnum
    FROM job
    WHERE jobname = 'מנהל'
)
WHERE w.wid = (
    SELECT wid
    FROM (
        SELECT w.wid, COUNT(*) AS appointment_count
        FROM appointments a
        JOIN workers w ON a.wid = w.wid
        JOIN job j ON w.jobnum = j.jobnum
        WHERE j.jobname = 'פקיד'
        GROUP BY w.wid
        ORDER BY appointment_count DESC
    )
    WHERE ROWNUM = 1
);
```

צילום אחרי:

130:1	system@XE	1 row updated in 0 seconds
-------	-----------	----------------------------

....

	JOBNUM	WID	JOBNAME
1	68	89	מנהל

זמן ריצה:

1 row updated in 0 seconds

Deletion queries:

1 - שאילתת מחיקה

השאילתה מוחקת עובדים שלא היה להם תורים בשנה האחרונה או עובדים שלא שלחו חבילות בכלל. השאילתה קודם מוחקת את כל הבנים שיש לטבלת העובדים שעונים על התנאי הנ"ל.

ניתן לראות כי בתחילה ישנם 100 עובדים:

	WID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19

העובדים אותם נרצה למחוק: (משלוחנים שלא חילקו בשנה האחרונה ופקידים שלא קיבלו תורים בחצי שנה האחרונה)

שאילתת מחיקה:

	WID
1	44
2	48
3	68
4	87

```
select wid
from appointments
minus
select distinct wid
from appointments a
where a.app_date>=ADD_MONTHS(SYSDATE, -6)
union
select wid
from packages
minus
select distinct wid
from packages p join trackingshipments t on t.idpackage=p.idpackage
where t.sendingdate>=ADD_MONTHS(SYSDATE, -12)
```

שאילתת מחיקה:

```
delete
from workers where wid in
(
select wid
from appointments
minus
select distinct wid
from appointments a
where a.app_date>=ADD_MONTHS(SYSDATE, -6)
union
select wid
from packages
minus
select distinct wid
from packages p join trackingshipments t on t.idpackage=p.idpackage
where t.sendingdate>=ADD_MONTHS(SYSDATE, -12)
)
```

תוצאה וזמן הרצה:

4 rows deleted in 0.015 seconds

2 - שאילתת מחיקה

השאילתה מוחקת תורים ישנים מלפני 16 חודשים ומעלה. בנוסף, השאילתה בודקת אם הלקוח שהזמין אות התור הזמין תור נוסף בשנה האחרונה, אם כן השאילתה לא מוחקת את התור הזה כי זה כנראה לקוח קבוע וכדאי שנשמור את היסטוריית התורים שלו.

כמה תורים צריך למחוק?

```
SELECT a.appid
FROM appointments a
Join costumer c ON c.id_sender = a.id_sender
where a.app_date <= ADD_MONTHS(SYSDATE, -16) and a.id_sender not in (select al.id_sender
                                                                    from appointments al
                                                                    where al.app_date >= ADD_MONTHS(SYSDATE, -12))
```

<- <- <-

50 rows selected in 0.078 seconds

שאילתת מחיקה:

```
DELETE FROM appointments
WHERE  appid IN(
    SELECT a.appid
    FROM appointments a
    Join costumer c ON c.id_sender = a.id_sender
    where a.app_date <= ADD_MONTHS(SYSDATE, -16) and a.id_sender not in (select al.id_sender
                                                                    from appointments al
                                                                    where al.app_date >= ADD_MONTHS(SYSDATE, -12)))
```

תוצאה וזמן הרצה:

50 rows deleted in 0.046 seconds

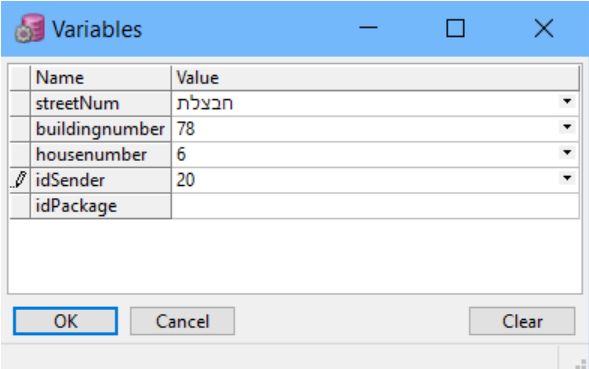
parameters queries

1 - שאלת פרמטרים

שאלת המעדכנת נתוני משלוח במידה והלקוח התקשר ורוצה לשנות את הכתובת. השאלת בודקת שהמשלוח עוד לא נמסר, ואם יש 2 משלוחים לאותו לקוח שעוד לא נשלחו, ולא הוזן מספר חבילה, השאלת לוקחת את המשלוח שהוזמן אחרון. נצלם לדוג' לקוח מספר 20 שיש לו 2 חבילות, נרצה לעדכן את החבילה הראשונה: (יש אפשרות לשים מספר חבילה אבל אנחנו נשאיר ריק לראות איך הוא לוקח את האחרונה מביניהם)

	IDPACKAGE	DELIVERYPHONE	RECIPIENTNAME	BUILDINGNUMBER	HOUSENUMBER	STREETNUM	ORDERDATE	SORTDATE	SENDINGDATE	DELIVERYDATE	STATUSNUM
▶ 1	155	556196259	Joseph Phillips	87	37	50	12/02/2024	13/02/2024	27/10/2023		35
2	211	531726921	Gavin Jane	88	17	55	06/02/2023	01/10/2023	23/10/2023	24/10/2023	36

בשם הרחוב צריך לבחור מתוך רשימת הרחובות שבאיזור החלוקה:



שאלת עדכון:

```
update packages p
set p.streetnum=<name=streetNum list="select streetNum, streetName from street" description=true type=integer restricted=true hint="choose the name of the street">
, p.buildingnumber=<name=buildingnumber type=integer required=true hint="enter the building number">
, p.housenumber=<name=housenumber type=integer required=true hint="enter the house number">
, p.deliveryphone=(
    select c.phone
    from costumer c
    where c.id_sender=<name=idSender type=integer required=true>)
where p.id_sender in (
    select c.id_sender
    from costumer c
    where c.id_sender=<name=idSender type=integer required=true>
)
and p.idpackage in (
    select t.idpackage
    from trackshipments t
    where t.deliverydate is null
)
and p.idpackage = COALESCE(<name=idPackage type=integer>, (
    select p.idpackage
    from packages p join trackshipments t on p.idpackage = t.idpackage
    where p.id_sender = <name=idSender type=integer required=true>
    and
    t.orderdate = (
        select max(t2.orderdate)
        from trackshipments t2 join packages p2 on t2.idpackage = p2.idpackage
        where p2.id_sender = <name=idSender type=integer required=true>
    )
));
```

תוצאה לאחר השינוי:

	IDPACKAGE	DELIVERYPHONE	RECIPIENTNAME	BUILDINGNUMBER	HOUSENUMBER	STREETNUM	ORDERDATE	SORTDATE	SENDINGDATE	DELIVERYDATE	STATUSNUM
▶ 1	155	566517274	Joseph Phillips	78	6	7	12/02/2024	13/02/2024	27/10/2023		35
2	211	531726921	Gavin Jane	88	17	55	06/02/2023	01/10/2023	23/10/2023	24/10/2023	36

זמן הרצה:

1 row updated in 0 seconds

2 - שאילתת פרמטרים

השאילתה מאפשרת להציג משלוחים מסוימים שעובד חילק, צריך לשים את פרטי העובד ואת השנה שבה רוצים לראות את החבילות שהוא חילק.
צריך לסמן את הסטטוס שבו רוצים לראות את החבילות.

נבחר עובד ונבקש לראות את החבילות בסטטוס נמסר משנת 2023:

Name	Value
Yeardate	2023
id	19
firstName	Mika
lastName	Kimball
status	נמסר

שאילתה:

```
select p.idpackage,p.weight,p.recipientname,p.packsize,p.streetnum,p.wid,w.firstname,w.lastname,t.orderdate,t.sortdate,t.sendingdate,t.deliverydate
from packages p join workers w on w.wid=p.wid join trackshipments t on t.idpackage=p.idpackage join orderstatus o on t.statusnum=o.statusnum
where p.idpackage in (
    select t.idpackage
    from trackshipments t
    where EXTRACT(YEAR from t.orderdate) =
        &<name="Yeardate" type=integer required=false ifempty=2024 hint="enter the year date you want to see the pakages from this date">
)
and w.wid=&<name="id" type=integer required=false>
and w.firstname = &<name="firstName" type=string hint="enter the first name of the worker">
and w.lastname = &<name="lastName" type=string hint="enter tha last name of the worker">
and o.statusnum = &<name="status" list="select statusnum, statusName from orderstatus" description=true type=integer restricted=true>
```

תוצאה:

(אצלנו יש בערך 400 חבילות ולכן אין הרבה חבילות עבור כל עובד, כמובן שבמציאות נראה רשימה הרבה יותר ארוכה)

	IDPACKAGE	WEIGHT	RECIPIENTNAME	PACKSIZE	STREETNUM	WID	FIRSTNAME	LASTNAME	ORDERDATE	SORTDATE	SENDINGDATE	DELIVERYDATE
1	212	13	Garth Bogguss	34	58	19	Mika	Kimball	28/05/2023	24/04/2024	25/04/2024	02/06/2023
2	240	5	Paul Sweeney	14	35	19	Mika	Kimball	29/03/2023	19/06/2023	29/07/2023	27/03/2024
3	360	10	Brenda Chandler	3	83	19	Mika	Kimball	16/12/2023	16/04/2024	17/04/2024	05/05/2024
4	407	6	Nastassja Gunto	10	76	19	Mika	Kimball	23/09/2023	24/09/2023	06/12/2023	30/01/2024
5	411	4	Hex Quaid	24	9	19	Mika	Kimball	26/01/2023	02/06/2023	31/05/2024	01/06/2024

זמן הרצה:

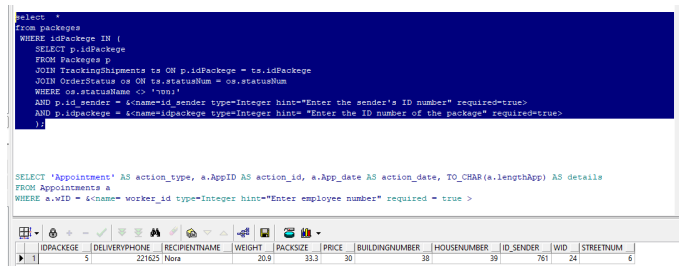
5 rows selected in 0.016 seconds

3 - שאילתת פרמטרים

זו שאילתת מחיקה עם פרמטרים, השאילתה מיועדת למקרה שלקוח ששלח חבילה התחרט ורוצה לבטל את החבילה, המחיקה תתאפשר רק אם החבילה עוד לא נמסרה. נצטרך למלא את תעודת זהות הלקוח ולבחור מתוך החבילות שלו את מספר החבילה אותה הוא רוצה למחוק.

לפני:

מספר חבילה, ת.ז. לקוח (הצגנו את אלה שעוד לא נמסרו, ואח"כ נמחק אחד מהם)



נפעיל את השאילתה:

```
DELETE
from packages
WHERE idPackage IN (
  SELECT p.idPackage
  FROM Packages p
  JOIN TrackingShipments ts ON p.idPackage = ts.idPackage
  JOIN OrderStatus os ON ts.statusNum = os.statusNum
  WHERE os.statusName <> 'נמסר'
  AND p.id_sender = <name=id_sender type=Integer hint="Enter the sender's ID number" required=true>
  AND p.idpackage = <name=idpackage type=Integer hint="Enter the ID number of the package" required=true>
);
```

מחקנו עם הפרמטרים הבאים:

Variables	
Name	Value
id_sender	761
idpackage	5

זמן ריצה:

1 row deleted in 0.015 seconds

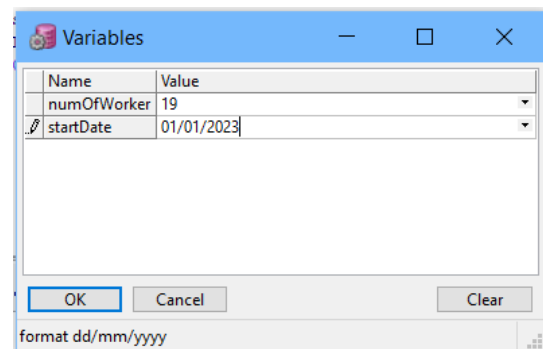
4 - שאלת פרמטרים

שאלת המבקשת מספר עובד ותאריך, ומחזירה את ממוצע הזמן שלקח מתאריך הזמנת החבילה עד תאריך מסירת החבילה של עובד מסוים החל מהתאריך המבוקש.

שאלתה:

```
select avg(t.deliverydate-t.orderdate)
from workers w join packages p on w.wid=p.wid join trackingshipments t on p.idpackage=t.idpackage
where w.wid=&<name=numOfWorkers hint="enter id of worker" type=integer required=true> and
orderdate > &<name=startDate type=date hint="format dd/mm/yyyy">
and t.deliverydate is not null
```

נמלא בפרמטרים:



תוצאה:

	AVG(T.DELIVERYDATE-T.ORDERDATE)
1	190

זמן הרצה:

1 row selected in 0.016 seconds

אימות:

```
select t.deliverydate-t.orderdate
from workers w join packages p on w.wid=p.wid join trackingshipments t on p.idpackage=t.idpackage
where w.wid=19
```

	T.DELIVERYDATE-T.ORDERDATE
1	9
2	
3	
4	
5	5
6	364
7	
8	
9	
10	
11	141
12	129
13	492

אילוצים

אילוץ 1

האילוץ מחייב שמשקל החבילה יהיה גדול או שווה ל-0. (בשביל מעטפות)

```
ALTER TABLE packages ADD CONSTRAINT chk_weight_positive CHECK (weight >= 0);
```

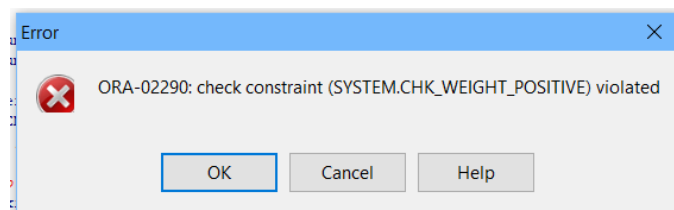
הרצה:

Done in 0.015 seconds

ניסיון הכנסה:

```
insert into packages ( idpackage, deliveryphone, recipientname, weight, packsize, price, buildingnumber, housenumber, id_sender, wid, streetnum)
values (999,0501234567,'mom',-3,54,67,4,7,2,5,4)
```

שגיאה:



אילוץ 2

האילוץ מחייב שבטבלה JOB לא יהיו 2 מפתחות עבור אותו תפקיד.

```
ALTER TABLE Job ADD CONSTRAINT unique_jobName UNIQUE (jobName);
```

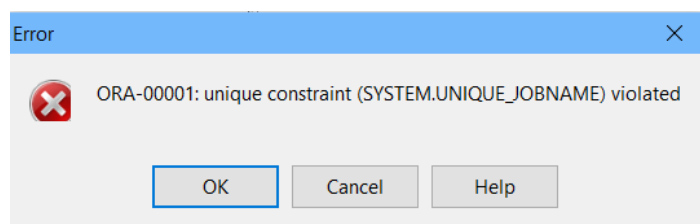
הרצה:

Done in 0.078 seconds

ניסיון הכנסה:

```
insert into Job (Jobnum, Jobname, Hourorpackagesalary) values (99, 'פקיד', 40)
```

שגיאה:



אילוח 3

האילוח קובע כי על כל חבילה שנכנסת לטבלת חבילות, אם לא הוגדר עבורה מחיר זה יהיה המחיר המינימלי 5 ש"ח.

```
ALTER TABLE packages MODIFY (price DEFAULT 5);
```

הרצה:

Done in 0 seconds

הכנסה ללא הערך מחיר:

```
insert into packages (idpackage, deliveryphone, recipientname, weight, packsize, buildingnumber, housenumber, id_sender, wid, streetnum)
values (420,0501234567,'mom',3,54,67,4,7,19,5)
```

השורה בטבלה לאחר מכן:

	IDPACKAGE	DELIVERYPHONE	RECIPIENTNAME	WEIGHT	PACKSIZE	PRICE	BUILDINGNUMBER	HOUSENUMBER	ID_SENDER	WID	STREETNUM
▶ 1	420	501234567	mom	3	54	5	67	4	7	19	5

אילוחים נוספים:

האילוח קובע כי על כל חבילה שנכנסת לטבלת מעקב משלוחים הערך ברירת מחדל של הסטטוס יהיה התקבל.

```
ALTER TABLE trackingshipments MODIFY (Statusnum DEFAULT 1);
```

האילוח קובע כי לא ניתן להכניס את תאריכי מעקב המשלוחים בסדר לא הגיוני.

```
ALTER TABLE TrackingShipments ADD CONSTRAINT check_dates
CHECK (orderDate < sortDate AND sortDate < sendingDate AND sendingDate < deliveryDate);
```


תכנית ראשית 1

התכנית הזו אחראית על בונסים ללקוחות ולעובדים.
התכנית תקרא לפרוצדורה שמעדכנת במשכורות העובדים בונים לעובדים הנבחרים ומדפיסה את העובדים שזכו.
והתכנית תקרא גם לפונקציה שמחזירה טבלה עם הלקוחות ששלחו הכי הרבה חבילות ועם חבילה רנדומלית מתוך החבילות שלהם. עבור הפונקציה התכנית תדפיס את התוצאות בעצמה.

הקוד:

```
DECLARE
-- הכרזת משתנים
year_input NUMBER := 2024; -- שנה לדוגמה
ref_cursor SYS_REFCURSOR;
rec CREDIT_RESULTS_TEMP%ROWTYPE; -- סוג נתוני ים כללי של הקורסור
BEGIN
-- קריאו לפ נקציה עם שנת הקלט
ref_cursor := creditToCostumer(year_input);

DBMS_OUTPUT.PUT_LINE('The costumers that got credit for a random package of year ' || year_input || ' are:');

-- לולאת קריאה מהקורסור
LOOP
    FETCH ref_cursor INTO rec;
    EXIT WHEN ref_cursor%NOTFOUND;

    -- הדפסת התוצאות מהקורסור
    DBMS_OUTPUT.PUT_LINE('Package ID: ' || rec.PackageID ||
        ', Customer ID: ' || rec.CustomerID ||
        ', Price: ' || rec.Price);
END LOOP;

CLOSE ref_cursor;

-- קריאה לירוצדורה לעדכון בונים לעובדים
bonustotopworkers(year_input);

EXCEPTION
    WHEN OTHERS THEN
        -- טיפול בשגיאות
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
```

חלון הפלט:

```
The costumers that got credit for a random package of year 2024 are:
Package ID: 367, Customer ID: 397, Price: 5.9
Package ID: 239, Customer ID: 570, Price: 25
Package ID: 43, Customer ID: 803, Price: 10.9
Top Month: 05/2024
Top Worker 1: worker id 37 Mekhi Whitley, Deliveries: 3
Top Worker 2: worker id 22 Larenz Stowe, Deliveries: 3
Top Worker 3: worker id 81 Gaby Dooley, Deliveries: 2
```

פרוצדורה 1

תיאור:

הפרוצדורה מקבלת כפרמטר מספר של שנה, ובודקת עבור שנה זו באיזה חודש חולקו הכי הרבה חבילות, בחודש הזה היא מעדכנת למספר עובדים מסוים (במקרה שלנו 3, אך ניתן לשנות בקלות) שחילקו אם מספר המשלוחים הגדול ביותר בחודש זה את השכר לגבוה יותר ב-300 שקלים. במקרה שיתקבל כקלט שנה שלא מופיעה בבסיס הנתונים תזרק חריגה.

הקוד:

```
CREATE OR REPLACE PROCEDURE bonustotopworkers(input_year IN NUMBER) IS
    max_count_packages NUMBER := 0;
    max_month NUMBER;
    workerCount NUMBER := 0;

    -- Cursor to find the month with the most deliveries
    CURSOR month_cursor IS
        SELECT EXTRACT(month FROM t.deliveryDate) AS delivery_month, COUNT(*) AS delivery_count
        FROM trackingshipments t
        WHERE EXTRACT(year FROM t.deliveryDate) = input_year
        GROUP BY EXTRACT(month FROM t.deliveryDate);

    -- Cursor to find the top workers in the month with the most deliveries
    CURSOR workAtThisMonth IS
        SELECT w.firstname || ' ' || w.lastname AS worker_name, w.wid AS worker_id, COUNT(*) AS delivery_count
        FROM packages p JOIN trackingshipments t ON p.idpackage = t.idpackage
        JOIN workers w ON p.wid = w.wid
        WHERE EXTRACT(month FROM t.deliveryDate) = max_month AND EXTRACT(year FROM t.deliveryDate) = input_year
        GROUP BY w.firstname, w.lastname, w.wid
        ORDER BY delivery_count DESC;

    -- Variables to check if the year exists and handle exceptions
    year_exists BOOLEAN := FALSE;
    year_not_found EXCEPTION;
    PRAGMA EXCEPTION_INIT(year_not_found, -20001);

BEGIN
    -- Check if the input year exists
    FOR record IN (SELECT 1 FROM trackingshipments t WHERE EXTRACT(year FROM t.deliveryDate) = input_year) LOOP
        year_exists := TRUE;
        EXIT;
    END LOOP;

    IF NOT year_exists THEN
        RAISE_APPLICATION_ERROR(-20001, 'illegal year');
    END IF;

    -- Find the month with the most deliveries
    FOR record IN month_cursor LOOP
        IF record.delivery_count > max_count_packages THEN
            max_count_packages := record.delivery_count;
            max_month := record.delivery_month;
        END IF;
    END LOOP;

    -- Print the selected month and year
    DBMS_OUTPUT.PUT_LINE('Top Month: ' || TO_CHAR(max_month, 'FM00') || '/' || input_year);

    -- Find the top 3 workers and print their details
    FOR record IN workAtThisMonth LOOP
        EXIT WHEN workerCount >= 3;
        DBMS_OUTPUT.PUT_LINE('Top Worker ' || (workerCount + 1) || ': ' || 'worker id ' || record.worker_id || ' ' ||
            record.worker_name || ', Deliveries: ' || record.delivery_count);

        -- Update the bonus for the top workers
        UPDATE workers
        SET bonus = bonus + 300
        WHERE wid = record.worker_id;
        commit;

        workerCount := workerCount + 1;
    END LOOP;

EXCEPTION
    WHEN year_not_found THEN
        DBMS_OUTPUT.PUT_LINE('Error: illegal year');
END bonustotopworkers;
```

נראה ניסיון חריגה:

1

begin

2

-- Call the procedure

3

bonustotopworkers(input_year => :input_year);

4

end;

	Variable	Type	Value
▶	input_year	Float	2025
*			

החריגה שנזרקה:

Test scriptDBMS OutputStatisticsProfilerTrace

ClearBuffer size10000Enabled

Error: illegal year

נכניס מספר שנה תקין:

1

begin

2

-- Call the procedure

3

bonustotopworkers(input_year => :input_year);

4

end;

	Variable	Type	Value
▶	input_year	Float	2024
*			

הפלט:

Test scriptDBMS OutputStatisticsProfilerTrace

ClearBuffer size10000Enabled

Top Month: 05/2024
Top Worker 1: worker id 37 Mekhi Whitley, Deliveries: 3
Top Worker 2: worker id 22 Larenz Stowe, Deliveries: 3
Top Worker 3: worker id 81 Gaby Dooley, Deliveries: 2

ניתן לראות כי העובדים עודכנו:

select bonus, firstname,lastname,wid
from workers w
where wid=37
union
select bonus, firstname,lastname,wid
from workers w
where wid=81
union
select bonus, firstname,lastname,wid
from workers w
where wid=22

	BONUS	FIRSTNAME	LASTNAME	WID
▶ 1	300	Gaby	Dooley	81
2	300	Larenz	Stowe	22
3	300	Mekhi	Whitley	37

פונקציה 1

תיאור:

הפונקציה מקבלת כפרמטר מספר שנה, ואז מוצאת את מקסימום 3 הלקוחות ששלחו הכי הרבה חבילות בשנה זו:
אם יש לדוג' 2 לקוחות ששלחו את המספר המקסימלי של החבילות הפונקציה תחשב רק עבורם,
ואם יש יותר מ-3 הפונקציה תגריל 3 רנדומליים מבין הלקוחות.
עבור כל לקוח שזכה הפונקציה תגריל חבילה רנדומלית מבין החבילות שלו ותחזיר cursor עם שם לקוח, מספר חבילה ומחיר חבילה.

הפונקציה תזרוק חריגה במקרה שנכנס קלט של שנה לא חוקית.
(הטבלה תודפס רק בתכנית הראשית, ואז הפקידים יוכלו להתקשר ללקוחות ולתת להם החזר כספי על החבילה, אם לדוג' יש מבצע שכל מי ששולח דרך סניף הדואר נכנס להגרלה על חבילה חינם)

הוספת טבלה לעזר:

```
ALTER TABLE COSTUMER  
DROP COLUMN credit_points;  
commit;
```

הקוד:

```
3 CREATE OR REPLACE FUNCTION creditToCostumer(year_num IN NUMBER) RETURN SYS_REFCURSOR IS  
3   Result SYS_REFCURSOR;  
   max_packages NUMBER;  
  
3   CURSOR id_costumers IS  
3     SELECT *  
     FROM (  
       SELECT idCostumer  
       FROM (  
         SELECT COUNT(*) AS countPackage, p.id_sender AS idCostumer  
         FROM packages p  
         JOIN trackingshipments t ON p.idpackage = t.idpackage  
         WHERE EXTRACT(YEAR FROM t.orderdate) = year_num  
         GROUP BY p.id_sender  
         ORDER BY DBMS_RANDOM.VALUE  
       )  
       WHERE countPackage = max_packages  
     )  
     WHERE ROWNUM <= 3;  
  
3   TYPE PackageRecord IS RECORD (  
     PackageID packages.idpackage%TYPE,  
     CustomerID packages.id_sender%TYPE,  
     Price packages.price%TYPE  
   );  
  
   rec PackageRecord;  
  
   year_exists BOOLEAN := FALSE;  
   year_not_found EXCEPTION;  
   PRAGMA EXCEPTION_INIT(year_not_found, -20001);  
  
BEGIN  
  -- Check if the input year exists  
3  FOR record IN (SELECT 1 FROM trackingshipments t WHERE EXTRACT(YEAR FROM t.deliveryDate) = year_num) LOOP  
    year_exists := TRUE;  
    EXIT;  
  END LOOP;  
  
3  IF NOT year_exists THEN  
    DBMS_OUTPUT.PUT_LINE('Error: illegal year');  
    OPEN Result FOR SELECT NULL AS PackageID, NULL AS CustomerID, NULL AS Price FROM DUAL WHERE 1 = 0;  
    RETURN Result;  
  END IF;
```

```

SELECT MAX(COUNT(*)) INTO max_packages
FROM packages p
JOIN trackingshipments t ON p.idpackage = t.idpackage
WHERE EXTRACT(YEAR FROM t.orderdate) = year_num
GROUP BY p.id_sender
ORDER BY COUNT(*) DESC;

OPEN id_costumers;
DELETE FROM CREDIT_RESULTS_TEMP;

LOOP
  FETCH id_costumers INTO rec.CustomerID;
  EXIT WHEN id_costumers%NOTFOUND;

  BEGIN
    SELECT price, packageId, id_sender INTO rec.Price, rec.PackageID, rec.CustomerID
    FROM (
      SELECT p.price AS price, p.idpackage AS packageId, p.id_sender AS id_sender
      FROM packages p
      JOIN trackingshipments t ON p.idpackage = t.idpackage
      WHERE EXTRACT(YEAR FROM t.orderdate) = year_num
      AND p.id_sender = rec.CustomerID
      ORDER BY DBMS_RANDOM.VALUE
    )
    WHERE ROWNUM = 1;

    INSERT INTO CREDIT_RESULTS_TEMP (PackageID, CustomerID, Price)
    VALUES (rec.PackageID, rec.CustomerID, rec.Price);

  END;
END LOOP;

CLOSE id_costumers;

OPEN Result FOR
  SELECT *
  FROM CREDIT_RESULTS_TEMP;

RETURN Result;

EXCEPTION
  WHEN year_not_found THEN
    DBMS_OUTPUT.PUT_LINE('Error: illegal year');
    OPEN Result FOR SELECT NULL AS PackageID, NULL AS CustomerID, NULL AS Price FROM DUAL WHERE 1 = 0;
    RETURN Result;
END creditToCostumer;

```

הרצה:

```

begin
  -- Call the function
  :result := credittocostumer(year_num => :year_num);
end;

```

נכניס פרמטר:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Cursor	<Cursor>
<input checked="" type="checkbox"/>	year_num	Float	2023
<input checked="" type="checkbox"/>	*		

אין תוצאה מודפסת. היא תודפס רק בתכנית הראשית.

ניסיון להכניס שנה לא חוקית:

1	begin
2	-- Call the function
3	:result := credittocustomer(year_num => :year_num);
4	end;

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Cursor	<Cursor>
<input checked="" type="checkbox"/>	year_num	Float	2025

שגיאה:

Test script
DBMS Output
Statistics
Profiler
Trace

Clear
Buffer size 10000
☒ Enabled

Error: illegal year

נוסיף צילומי מסך של הדפסה מהתכנית הראשית של 2 הרצות שונות על מנת שיהיה ניתן לראות שהחבילות באמת מוגרלות רנדומלית עבור הלקוחות הזוכים:

```
The costumers that got credit for a random package of year 2024 are:
Package ID: 70, Customer ID: 93, Price: 20
Package ID: 120, Customer ID: 207, Price: 50
Package ID: 41, Customer ID: 570, Price: 25
```

```
The costumers that got credit for a random package of year 2024 are:
Package ID: 367, Customer ID: 397, Price: 5.9
Package ID: 41, Customer ID: 570, Price: 25
Package ID: 237, Customer ID: 325, Price: 25
```

תוכנית ראשית 2:

```
DECLARE
package_ref_cursor SYS_REFCURSOR;
recP packages%ROWTYPE;
BEGIN

-- קר אה לירוצדורה וס מספר מינלי חבילות 5 ואחוזי הנחה 10
calculate_package_price(min_packages_for_discount => 5, discount_percentage => 10);

-- CURSOR-קו יאה לפונקציה וקבלת ה
package_ref_cursor := splittingLargePackages(43.0); -- , 45 , הגולת משקל של

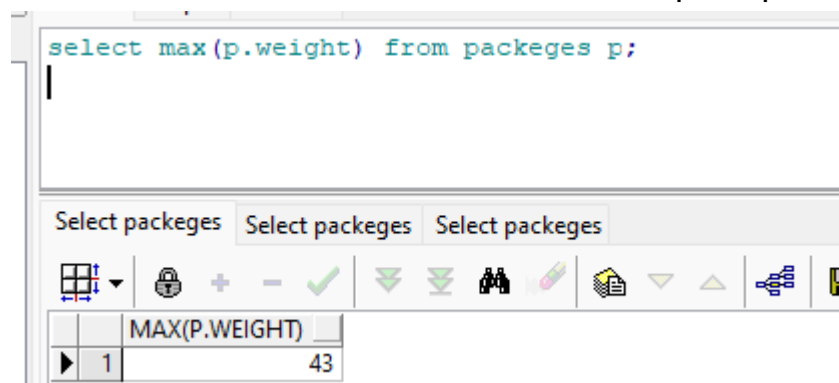
-- CURSOR-בי יקת אם ה
IF package_ref_cursor IS NOT NULL THEN
    LOOP
        FETCH package_ref_cursor INTO recP;
        EXIT WHEN package_ref_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('ID: ' || recP.idpackage ||
                               ', Phone: ' || recP.deliveryphone ||
                               ', Name: ' || recP.recipientname ||
                               ', Weight: ' || recP.weight ||
                               ', Size: ' || recP.packsize ||
                               ', Price: ' || recP.price ||
                               ', Building: ' || recP.buildingnumber ||
                               ', House: ' || recP.housenumber ||
                               ', Sender: ' || recP.id_sender ||
                               ', Worker: ' || recP.wid ||
                               ', Street: ' || recP.streetnum);
    END LOOP;
    CLOSE package_ref_cursor;
ELSE
    DBMS_OUTPUT.PUT_LINE('No packages to display.');
```

```
END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred in main block: ' || SQLERRM);
END;
```

הקוד הנ"ל הינו קוד של התוכנית הראשית.

התוכנית הראשית קוראת לפרוצדורה calculate package price עם שני פרמטרים מתאימים.

לאחר מכן התוכנית קוראת לפונקציה splitting Large Packages המקבלת כפרמטר מספר המייצג את המשקל המרבי. המספר שנשלח לפונקציה הינו 43 מכיוון שהוא כרגע המשקל המקסימלי שנמצא לנו בנתוני החבילות.



לבסוף התוכנית מדפיסה את הנתונים שהוחזרו מהפונקציה. להלן תיאור הפרוצדורה והפונקציה.

פרוצדורה 2:

calculate_package_price:

מקבלת שתי פרמטרים:

min_packages_for_discount - המייצגת את מספר המינימלי של חבילות שלקוח צריך לשלוח כדי לקבל הנחה על חבילה.

discount_percentage - אחוז הנחה שלקוח יקבל על חבילה אם שלח מעל מספר מינימלי של חבילות.

עוברת על כל החבילות ומחשבת את המחיר של כל חבילה לפי משקל וגודל של אותה חבילה. בנוסף, היא בודקת אם אותו לקוח שלח מעל למספר החבילות המינימלי (אותו מקבלת הפרוצדורה כפרמטר) ואם כן היא נותן הנחה על החבילה (אחוזי ההנחה גם כן נשלחים כפרמטר לפרוצדורה).
הפרוצדורה זורקת חריגות במקרים הבאים: 1. הוכנס 0 במשתנה min_packages_for_discount, כלומר, לא ניתן לתת הנחה ללקוח אם הוא שלח 0 חבילות.
2. הוכנס מספר גדול מ 100 למשתנה discount_percentage, כלומר לא ניתן לתת הנחה של יותר מ 100 אחוז.

הקוד:

```
1 CREATE OR REPLACE PROCEDURE calculate_package_price(  
2   min_packages_for_discount NUMBER,  
3   discount_percentage NUMBER  
4 ) IS  
5   -- הכרזת משתנים  
6   CURSOR package_cursor IS  
7     SELECT idPackage, id_sender, weight, packsize, price  
8       FROM packages;  
9  
10  TYPE ref_cursor_type IS REF CURSOR;  
11  package_ref_cursor ref_cursor_type;  
12  
13  rec package_cursor%ROWTYPE;  
14  total_price NUMBER;  
15  package_count NUMBER;  
16  discounted_price NUMBER;  
17  
18  -- חריגות  
19  no_packages_found EXCEPTION;  
20  discount_error EXCEPTION;  
21  minPacIsNull EXCEPTION;  
22  notDiscount EXCEPTION;  
23  
24  BEGIN  
25  
26  IF min_packages_for_discount = 0 THEN  
27    RAISE minPacIsNull;  
28  END IF;  
29  
30  IF discount_percentage > 100 THEN  
31    RAISE notDiscount;  
32  END IF;  
33  
34  -- פתיחת Cursor  
35  OPEN package_cursor;  
36  
37  LOOP  
38    FETCH package_cursor INTO rec;  
39    EXIT WHEN package_cursor%NOTFOUND;  
40  
41    -- חישוב המחיר הכולל לפי משקל וגודל  
42    total_price := rec.weight * 2 + rec.packsize * 1.5;  
43  END LOOP;
```



```

SELECT COUNT(*) INTO package_count
FROM packages
WHERE id_sender = rec.id_sender;

-- המעטפות 'העניות' הנחה וס מספר החבילות מעל למינימום
IF package_count > min_packages_for_discount THEN
    discounted_price := total_price * (1 - discount_percentage / 100);
ELSE
    discounted_price := total_price;
END IF;

-- עדכון מחיר ה'בילה' רק אם הוא לא שווה למחיר שחשבנו
IF rec.price != discounted_price THEN
    UPDATE packages
    SET price = discounted_price
    WHERE idPackage = rec.idPackage;
END IF;

END LOOP;

-- סגירת Cursor
CLOSE package_cursor;

-- Ref Cursor החזרת
OPEN package_ref_cursor FOR
SELECT idPackage, id_sender, weight, packsize, price
FROM packages;

-- הצגת חבילות מעודכנות
-- LOOP
-- FETCH package_ref_cursor INTO rec;
-- EXIT WHEN package_ref_cursor%NOTFOUND;

-- DBMS_OUTPUT.PUT_LINE('Package ID: ' || rec.idPackage || ', Sender ID: ' || rec.id_sender ||
-- ', Weight: ' || rec.weight || ', Size: ' || rec.packsize ||
-- ', Price: ' || rec.price);

-- END LOOP;

CLOSE package_ref_cursor;

-- סינויים Commit

```

```

70 -- END LOOP;
71
72 CLOSE package_ref_cursor;
73
74 -- סינויים Commit
75 COMMIT;
76
77 EXCEPTION
78 WHEN NO_DATA_FOUND THEN
79     DBMS_OUTPUT.PUT_LINE('No package found.');
```

החריגות:

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	min_packages_for_discount	Float	0
<input checked="" type="checkbox"/>	discount_percentage	Float	10
<input checked="" type="checkbox"/>	*		

Clear Buffer size 10000 ☒ Enabled

It is not possible to give a discount to a customer who sent 0 packages

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	min_packages_for_discount	Float	4
<input checked="" type="checkbox"/>	discount_percentage	Float	120
<input checked="" type="checkbox"/>	*		

Clear Buffer size 10000 ☒ Enabled

It is not possible to give a discount of more than 100 percent

ניתן לראות שאחרי הפעלת הפרוצדורה כל מחיר של חבילה הינו: משקל כפול 2 + גודל כפול 1.5, כפי שהפרוצדורה מעדכנת.

IDPACKAGE	DELIVERYPHONE	RECIPIENTNAME	WEIGHT	PACKSIZE	PRICE	BUILDINGNUMBER	HOUSENUMBER	ID_SENDER	WID	STREETNUM	IS_SPLIT
1	2	87114972 Busta	11	55	104.5	8	40	202	25	48	0
2	4	73358114 Stevie	12	60	114	83	26	383	25	4	0
3	7	25619788 Brian	2	21	35.5	24	32	713	16	48	0
4	9	45975639 Kate	8	35	68.5	87	6	503	27	50	0
5	6	63116374 Jimmie	18	7	46.5	72	19	445	27	50	0
6	11	43165715 Elisabeth	32	28	106	62	22	535	29	48	0
7	13	58791164 Jonny Lee	15	19	58.5	48	38	108	17	6	0

פונקציה 2: splittingLargePackages

הפונקציה מקבלת: פרמטר maxWeight - הקובע מהו המשקל המקסימלי של חבילה.

הפונקציה: עוברת על כל החבילות ובודקת אם ישנם חבילות שמעל למשקל המקסימלי, במידה ויש היא מפצלת את החבילה לשתי חבילות.

לצורך הפונקציה הוספנו עמודה is_splitable שתאוחל באופן רנדומלי ל 1 או 0. זה אם ניתן לפצל את החבילה ו0 לא ניתן לפצל את החבילה. במידה ויש חבילה עם משקל מקסימלי ולא ניתן לפצל את החבילה יודפס למערכת הודעה.

הפונקציה תחזיר: משנה מסוג Ref Cursor שבו נמצאים כל החבילות שפוצלו. לצורך כך הוספנו עמודה חדשה לטבלת החבילות הנקראת IS_SPLIT שוסמן ב1 עבור החבילות שהתפצלו. כמו כן הרחבנו את שדה מספר מזהה של חבילות ל 3 NUMBER ל 4 שיוכל להכיל יותר מ 999 חבילות.

הפונקציה תזרוק חריגה: במידה ואין חבילות עם משקל גדול או שווה למשקל המקסימלי שהוכנס כפרמטר.

הקוד:

```
CREATE OR REPLACE FUNCTION splittingLargePackages(maxWeight FLOAT) RETURN SYS_REFCURSOR IS
    CURSOR packagesCursor IS
        SELECT *
        FROM packages p
        WHERE p.weight >= maxWeight;

    recP packages%ROWTYPE;
    TYPE ref_cursor_type IS REF CURSOR;
    package_ref_cursor ref_cursor_type;
    numberPa NUMBER;
    maxIdPac packages.idpackage%TYPE;
    noBigPackages EXCEPTION;
BEGIN

    SELECT COUNT(*)
    INTO numberPa
    FROM packages p
    WHERE p.weight >= maxWeight;

    IF numberPa = 0 THEN
        RAISE noBigPackages;
    ELSE
        OPEN packagesCursor;
        LOOP
            FETCH packagesCursor INTO recP;
            EXIT WHEN packagesCursor%NOTFOUND;

            if recP.is_splitable = 0 THEN
                DBMS_OUTPUT.PUT_LINE('This package cannot be split');
            ELSE

                SELECT MAX(p.idpackage)
                INTO maxIdPac
                FROM packages p;

                INSERT INTO packages (idpackage, deliveryphone, recipientname, weight, packsize, price, buildingnumber, housenumber, id_sender, wid, streetnum, is_split)
                VALUES (maxIdPac+1, recP.deliveryphone, recP.recipientname, recP.weight / 2, recP.packsize / 2, recP.price / 2, recP.buildingnumber, recP.housenumber, recP.id_sender, recP.wid, recP.streetnum, 1);

                UPDATE packages
```

```

        price = recP.Price / 2,
        is_split = 1
        WHERE idpackage = recP.idpackage;
    END LOOP;
    CLOSE packagesCursor;
END IF;

-- חזרת CURSOR
OPEN package_ref_cursor FOR
    SELECT *
    FROM packages
    WHERE is_split = 1;

RETURN package_ref_cursor;
EXCEPTION
    WHEN noBigPackages THEN
        DBMS_OUTPUT.PUT_LINE('No large or equal weight packages found. ');
        RETURN NULL;
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        RETURN NULL;
END splittingLargePackages;

```

כרגע המשקל המקסימלי של החבילות הוא 20, אז נכניס פרמטר מספר מקסימלי 19:

```

1 begin
2     -- Call the function
3     :result := splittingLargePackages(maxweight => :maxweight);
4 end;

```

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	result	Cursor	<Cursor>
<input checked="" type="checkbox"/>	maxweight	Float	19
<input checked="" type="checkbox"/>	*		

לפני: נשים לב כי ישנם 15 חבילות בלתי ניתנות לפיצול

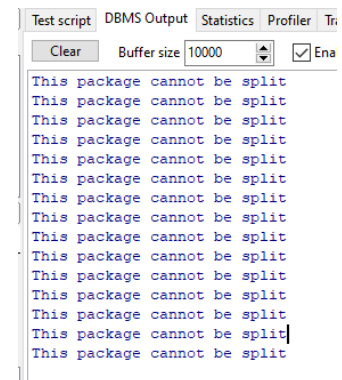
```

select weight, p.idpackage, p.is_splitable, p.is_split
from packages p
where weight > 19

```

	WEIGHT	IDPACKAGE	IS_SPLITABLE	IS_SPLIT
1	20	52	0	0
2	20	86	0	0
3	20	101	0	0
4	20	143	0	0
5	20	170	1	0
6	20	176	0	0
7	20	187	1	0
8	20	189	0	0
9	20	198	1	0
10	20	220	0	0
11	20	223	0	0
12	20	245	0	0
13	20	261	0	0
14	20	272	0	0
15	20	280	1	0
16	20	305	1	0
17	20	325	0	0
18	20	363	0	0
19	20	369	0	0
20	20	397	0	0
21	20	403	1	0

פלט: 15 פעמים הודעה כי לא ניתן לפצל את החבילה.



וניתן לראות כי בחבילות שגדולות מהמשקל המקסימלי נותרו רק 15, השאר פוצלו ל20 ומשקלם ירד.

```
select weight, p.idpackage, p.is_splitable, p.is_split
from packages p
where weight > 19
```

	WEIGHT	IDPACKAGE	IS_SPLITABLE	IS_SPLIT
1	20	52	0	0
2	20	86	0	0
3	20	101	0	0
4	20	143	0	0
5	20	176	0	0
6	20	189	0	0
7	20	220	0	0
8	20	223	0	0
9	20	245	0	0
10	20	261	0	0
11	20	272	0	0
12	20	325	0	0
13	20	363	0	0
14	20	369	0	0
15	20	397	0	0

הדפסת התוכנית שהרצנו:

```
ID: 16, Phone: 93234657, Name: Marc, Weight: 23, Size: 22.5, Price: 79.75, Building: 40, House: 24, Sender: 532, Worker: 24, Street: 4
ID: 96, Phone: 98946264, Name: Kiefer, Weight: 22.5, Size: 34, Price: 96, Building: 51, House: 28, Sender: 912, Worker: 9, Street: 2
ID: 101, Phone: 94473474, Name: Ronny, Weight: 21.5, Size: 35, Price: 95.5, Building: 56, House: 32, Sender: 362, Worker: 10, Street: 2
ID: 108, Phone: 16914919, Name: Marisa, Weight: 25, Size: 29, Price: 84.15, Building: 78, House: 24, Sender: 312, Worker: 23, Street: 28
ID: 111, Phone: 82686795, Name: Claire, Weight: 25, Size: 2, Price: 53, Building: 67, House: 35, Sender: 296, Worker: 30, Street: 30
ID: 45, Phone: 21166165, Name: Merle, Weight: 23.5, Size: 31, Price: 93.5, Building: 7, House: 4, Sender: 28, Worker: 1, Street: 50
ID: 46, Phone: 23154154, Name: Terrence, Weight: 25, Size: 28.5, Price: 92.75, Building: 80, House: 3, Sender: 34, Worker: 30, Street: 6
ID: 49, Phone: 26686974, Name: Nathan, Weight: 24, Size: 32.5, Price: 96.75, Building: 59, House: 37, Sender: 600, Worker: 12, Street: 25
ID: 59, Phone: 72592164, Name: Wes, Weight: 23, Size: 33, Price: 95.5, Building: 88, House: 23, Sender: 68, Worker: 22, Street: 2
ID: 61, Phone: 38864663, Name: Anita, Weight: 24, Size: 29.5, Price: 92.25, Building: 73, House: 22, Sender: 791, Worker: 20, Street: 48
ID: 106, Phone: 79914283, Name: Toshiro, Weight: 23, Size: 26, Price: 76.5, Building: 87, House: 36, Sender: 312, Worker: 24, Street: 4
ID: 148, Phone: 16914919, Name: Marisa, Weight: 25, Size: 29, Price: 84.15, Building: 78, House: 24, Sender: 312, Worker: 23, Street: 28
ID: 149, Phone: 82686795, Name: Claire, Weight: 25, Size: 2, Price: 53, Building: 67, House: 35, Sender: 296, Worker: 30, Street: 30
ID: 150, Phone: 23154154, Name: Terrence, Weight: 25, Size: 28.5, Price: 92.75, Building: 80, House: 3, Sender: 34, Worker: 30, Street: 6
ID: 800, Phone: 16914919, Name: Marisa, Weight: 26.5, Size: 35, Price: 94.95, Building: 78, House: 24, Sender: 312, Worker: 23, Street: 28
ID: 151, Phone: 26686974, Name: Nathan, Weight: 24, Size: 32.5, Price: 96.75, Building: 59, House: 37, Sender: 600, Worker: 12, Street: 25
ID: 152, Phone: 38864663, Name: Anita, Weight: 24, Size: 29.5, Price: 92.25, Building: 73, House: 22, Sender: 791, Worker: 20, Street: 48
ID: 801, Phone: 16914919, Name: Marisa, Weight: 26.5, Size: 35, Price: 94.95, Building: 78, House: 24, Sender: 312, Worker: 23, Street: 28
ID: 802, Phone: 21166165, Name: Merle, Weight: 23.5, Size: 31, Price: 93.5, Building: 7, House: 4, Sender: 28, Worker: 1, Street: 50
ID: 803, Phone: 93234657, Name: Marc, Weight: 23, Size: 22.5, Price: 79.75, Building: 40, House: 24, Sender: 532, Worker: 24, Street: 4
ID: 804, Phone: 72592164, Name: Wes, Weight: 23, Size: 33, Price: 95.5, Building: 88, House: 23, Sender: 68, Worker: 22, Street: 2
ID: 805, Phone: 79914283, Name: Toshiro, Weight: 23, Size: 26, Price: 76.5, Building: 87, House: 36, Sender: 312, Worker: 24, Street: 4
ID: 806, Phone: 98946264, Name: Kiefer, Weight: 22.5, Size: 34, Price: 96, Building: 51, House: 28, Sender: 912, Worker: 9, Street: 2
ID: 807, Phone: 94473474, Name: Ronny, Weight: 21.5, Size: 35, Price: 95.5, Building: 56, House: 32, Sender: 362, Worker: 10, Street: 2
```

ניתן לראות את כל החבילות שפוצלו, ביניהם גם את החבילה עם המשקל המקסימלי.

```
ID: 101, Phone: 94473474, Name: Ronny, Weight: 21.5, Size: 35, Price: 95.5, Building: 56, House: 32, Sender: 362, Worker: 10, Street: 2
ID: 807, Phone: 94473474, Name: Ronny, Weight: 21.5, Size: 35, Price: 95.5, Building: 56, House: 32, Sender: 362, Worker: 10, Street: 2
```

ניתן לראות שהחבילה אכן התפצלה לשניים.

אם נריץ שוב את התוכנית עם קריאה לפונקציה שוב עם מספר המקסימלי מאחר ופיצלנו את החבילה והוא כבר לא המשקל המקסימלי, ז"א לא קיים חבילה עם משקל זה יותר, התוכנית תזרוק שגיאה

```
No large or equal weight packages found.
No packages to display.
```

פרוצדורה נוספת שכתבנו על מנת לאתחל את הערכים בשדה is_splitable שיהיו בצורה רנדומלית בתוך הבחירה 0 או 1.

```
1 CREATE OR REPLACE PROCEDURE UpdateIsSplit AS
2   -- הכרזה על הקורסור
3   CURSOR pkg_cursor IS
4     SELECT *
5     FROM packages;
6
7   -- משתנים יעבודה עם הקורסור
8   v_rowid packages%ROWTYPE;
9   v_random_value NUMBER(1);
10
11 BEGIN
12   -- פתיחת הקורסור
13   OPEN pkg_cursor;
14
15   -- לולאת FETCH
16   LOOP
17     -- קבלת השווה הבאה מהקורסור
18     FETCH pkg_cursor INTO v_rowid;
19
20     -- יציאה מהלולאה אם לא נותרו שורות
21     EXIT WHEN pkg_cursor%NOTFOUND;
22
23     -- קביעת ירך רנדומלי 0 או 1
24     v_random_value := ROUND(DBMS_RANDOM.VALUE);
25
26     -- עדכון השדה is_splitable
27     UPDATE packages
28     SET is_splitable = v_random_value
29     WHERE idpackage = v_rowid.idpackage;
30   END LOOP;
31
32   -- סגירת הקורסור
33   CLOSE pkg_cursor;
34
35   -- אישור השינויים
36   COMMIT;
37
38 END;
```