

מיני פרויקט בקורס בסיסי נתונים:

מערכת תרומות הדם בארגון מד"א

מגישות:

שובל חדד 206846297

רעות צפלביץ 213114499

תוכן עניינים:

שלב א:

3.....	תיאור האירגון
4.....	תיאור הנתונים הנשמרים
5	טבלת ERD ו
6.....	טבלאות DSD ו
7-9.....	תוכן הקובץ createTable ו
10.....	פעולת describe ו
11.....	קובץ dropTable ו
11.....	קובץ selectAll ו
12-15.....	קובץ insertTable ו
16	גיבוי ושחזור הנתונים

שלב ב:

שאלות ללא פרמטרים:

17-20.....	שאלות SELECT ו
21-23.....	שאלות DELETE ו
24-26.....	שאלות UPDATE ו
27-34.....	שאלות עם פרמטרים
35-37.....	אילוצים
38.....	שינויים
39-40.....	פעולות ROLLBACK & COMMIT ו

שלב ג:

41-67.....	1 תוכנית
59-68.....	2 תוכנית

שלב א - תיאור המערכת

תיאור הארגון:

בפרויקט נבנה בסיס נתונים לתרומות דם לארגון מד"א. מגן דוד אדום (מד"א) הוא הארגון הלאומי של ישראל לסיוע רפואי והצלה במצבי חירום. הארגון פועל בתחומים שונים כמו סיוע רפואי חירום, סיוע פסיכוסוציאלי, תרומת דם, סיוע ראשוני, ועוד. מגן דוד אדום גם משמש כגוף להשתתפות בפעולות הומניטריות ברחבי העולם, כולל בזמן מבצעים צבאיים ואסונות טבע. הארגון מתבסס על צוות מתנדבים מוכשר ומקצועי, שמאומן לטיפול במגוון רחב של תרחישים חירום. מד"א מסתמך על תרומות דם לספק חומרים חיוניים לטיפול בפצועים ובחולים במצבי חירום. בפרויקט נתמקד בתחום תרומות הדם. מד"א עורך התרמות דם בנקודות התרמה שונות. חלקן הן ניידות אשר פרושות לאורך הארץ, וחלקן הן התרמות דם בשיתוף פעולה עם ארגונים שונים בכל הארץ אשר דואגים לתורמים פוטנציאליים. הפונקציונאליות העיקריות שנרצה לבצע ע"י המערכת הן למשל: בדיקת נתונים של תורם, כמו סוג הדם שלו ומתי הוא תרם לאחרונה, בשביל לבדוק אם הוא רשאי לתרום שוב, בנוסף, נוכל לבדוק נתוני אספקת דם לבתי החולים שונים, ונוכל לדעת כמה מנות תקינות מכל סוג דם יש במלאי בנק בדם.

תיאור הנתונים הנשמרים:

ישויות:

1. אדם (ישות אב)

2. תורם (ישות בן)

3. תרומת הדם

4. מקום ההתרמה

5. פרמדיק (ישות בן, האדם שלוקח את הדם מהתורם)

6. נמען (המוסד המקבל את הדם)

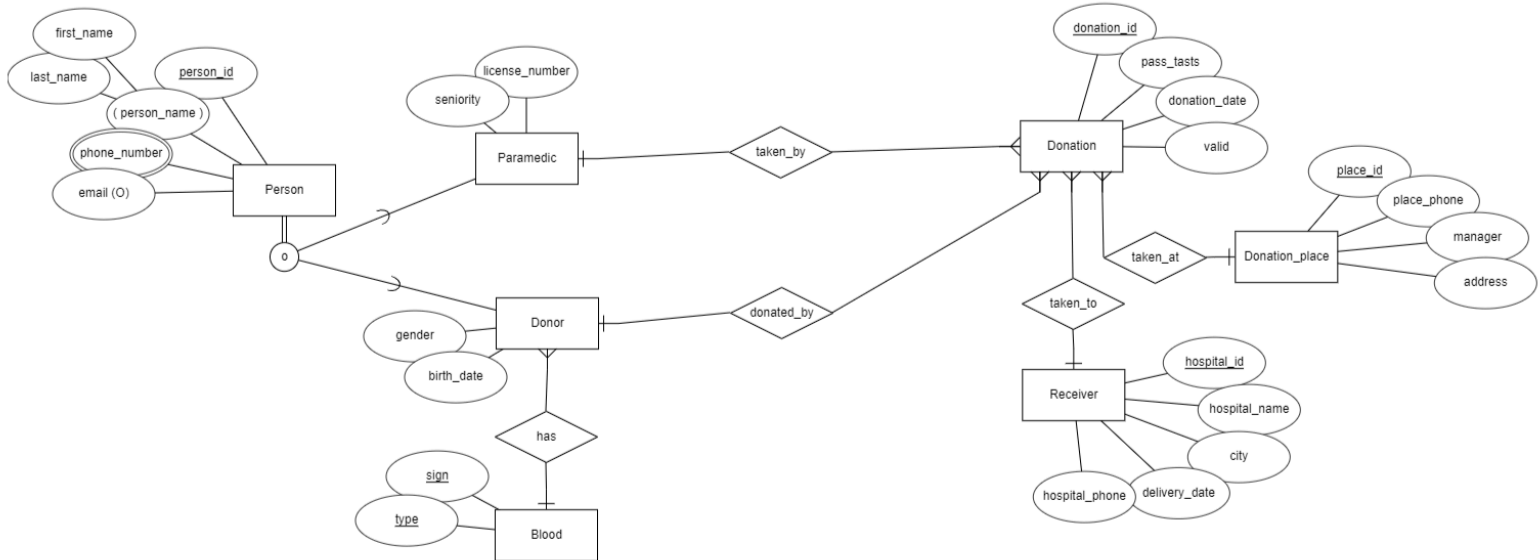
7. דם

היישות המרכזית היא יישות של תרומה, שיש לה מפתחות זרים של רוב הישויות האחרות. יש לנו ישות של אדם עם שדות בסיסיים כמו מספר ת"ז, מספר פלאפון וכו'. ממנה יורשים הישויות תורם ופרמדיק, תורם הוא אדם שממנו נלקח הדם והפרמדיק הוא האדם שלקח את הדם מן התורם. כיוון שבישות של אדם עשינו את התכונה מספר טלפון ככזו שיכולה לקבל מספר ערכים, ניצור גם עבור תכונה זו טבלה.

כמו כן יש ישות של מקום ההתרמה בו נערכה תרומת הדם וישות של ביה"ח שמקבל את מנת הדם לבסוף.

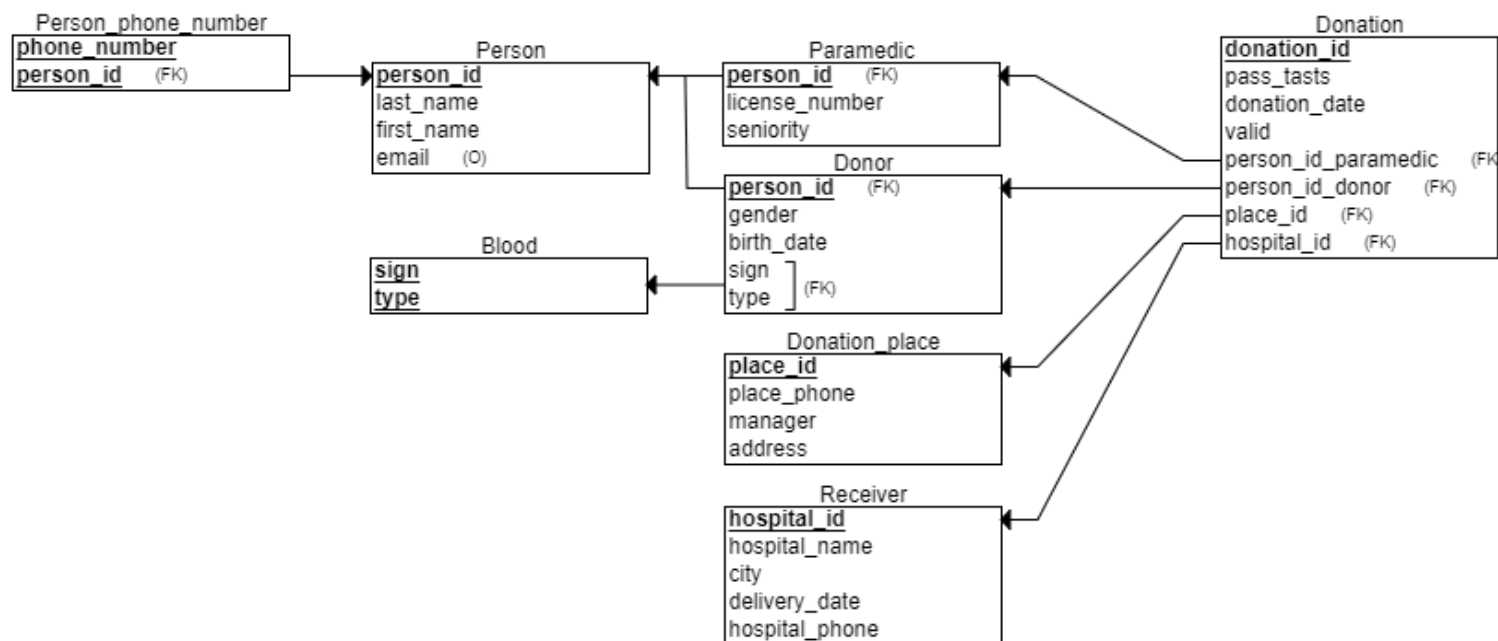
טבלת ERD -

(בתוכנת ERDPlus)

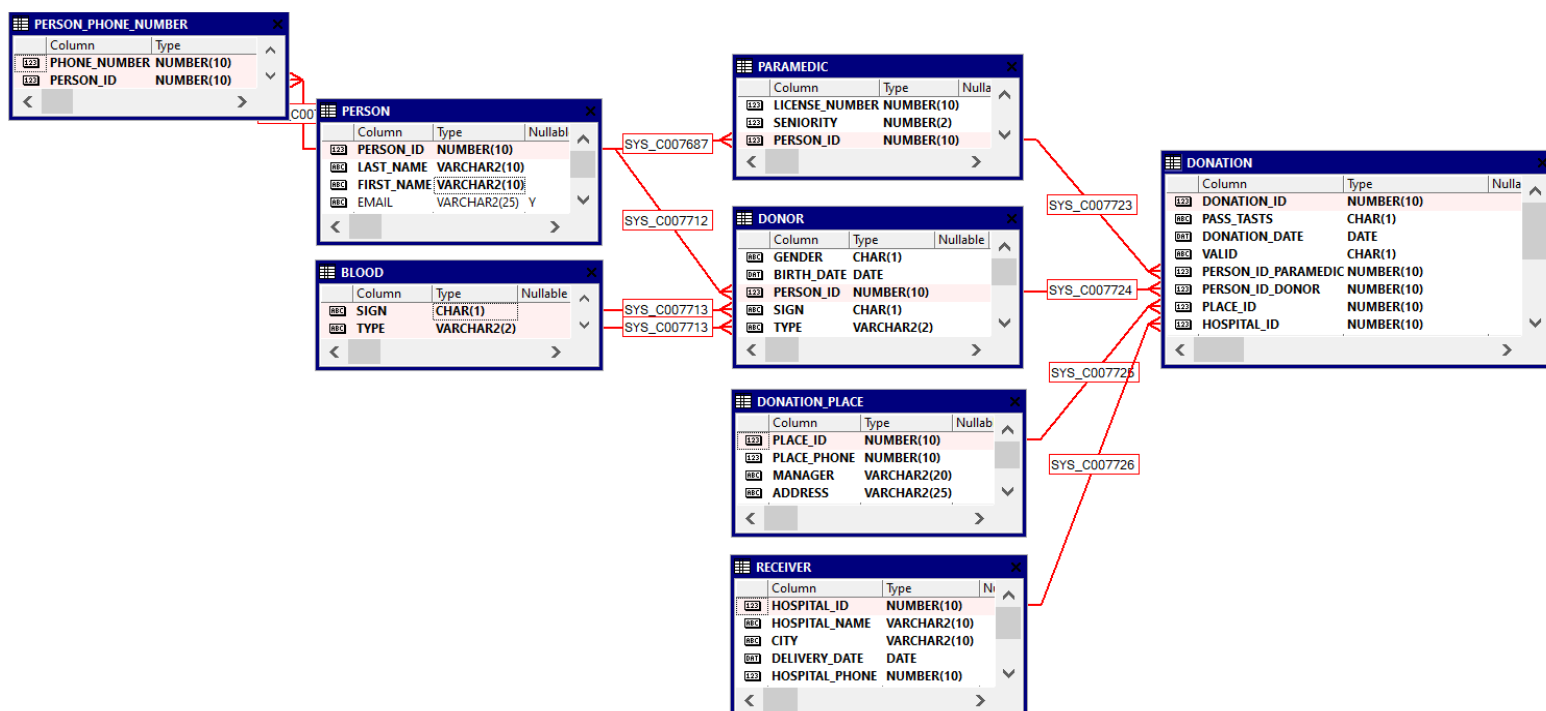


טבלת DSD -

(בתוכנת ERDPlus)



(בתוכנת Oracle)



- קובץ createTable

(את הקובץ createTables.sql ובו נמצא הסקריפט ליצירת הטבלאות העלנו גם לגיט)

```
CREATE TABLE Person

(
    person_id NUMERIC(10) NOT NULL,
    last_name VARCHAR(10) NOT NULL,
    first_name VARCHAR(10) NOT NULL,
    email VARCHAR(25),
    PRIMARY KEY (person_id)
);

CREATE TABLE Paramedic

(
    license_number NUMERIC(10) NOT NULL,
    seniority NUMERIC(2) NOT NULL,
    person_id NUMERIC(10) NOT NULL,
    PRIMARY KEY (person_id),
    FOREIGN KEY (person_id) REFERENCES Person(person_id)
);

CREATE TABLE Blood

(
    sign CHAR(1) NOT NULL,
    type VARCHAR(2) NOT NULL,
    PRIMARY KEY (sign, type)
);
```

```
CREATE TABLE Receiver
```

```
(  
    hospital_id NUMERIC(10) NOT NULL,  
    hospital_name VARCHAR(10) NOT NULL,  
    city VARCHAR(10) NOT NULL,  
    delivery_date DATE NOT NULL,  
    hospital_phone NUMERIC(10) NOT NULL,  
    PRIMARY KEY (hospital_id)  
);
```

```
CREATE TABLE Donation_place
```

```
(  
    place_id NUMERIC(10) NOT NULL,  
    place_phone NUMERIC(10) NOT NULL,  
    manager VARCHAR(20) NOT NULL,  
    address VARCHAR(25) NOT NULL,  
    PRIMARY KEY (place_id)  
);
```

```
CREATE TABLE Person_phone_number
```

```
(  
    phone_number NUMERIC(10) NOT NULL,  
    person_id NUMERIC(10) NOT NULL,  
    PRIMARY KEY (phone_number, person_id),  
    FOREIGN KEY (person_id) REFERENCES Person(person_id)  
);
```

```
CREATE TABLE Donor
```



```
(
    gender CHAR(1) NOT NULL,
    birth_date DATE NOT NULL,
    person_id NUMERIC(10) NOT NULL,
    sign CHAR(1) NOT NULL,
    type VARCHAR(2) NOT NULL,
    PRIMARY KEY (person_id),
    FOREIGN KEY (person_id) REFERENCES Person(person_id),
    FOREIGN KEY (sign, type) REFERENCES Blood(sign, type)
);
```

```
CREATE TABLE Donation
```

```
(
    donation_id NUMERIC(10) NOT NULL,
    pass_tests CHAR(1) NOT NULL,
    donation_date DATE NOT NULL,
    valid CHAR(1) NOT NULL,
    person_id_paramedic NUMERIC(10) NOT NULL,
    person_id_donor NUMERIC(10) NOT NULL,
    place_id NUMERIC(10) NOT NULL,
    hospital_id NUMERIC(10) NOT NULL,
    PRIMARY KEY (donation_id),
    FOREIGN KEY (person_id_paramedic) REFERENCES Paramedic(person_id),
    FOREIGN KEY (person_id_donor) REFERENCES Donor(person_id),
    FOREIGN KEY (place_id) REFERENCES Donation_place(place_id),
    FOREIGN KEY (hospital_id) REFERENCES Receiver(hospital_id)
);
```

פעולת describe -

ניתן לראות שהטבלאות נוצרו ע"י שימוש בפעולה describe:

Tables	
>	APEX\$_ACL
>	APEX\$_WS_FILES
>	APEX\$_WS_HISTORY
>	APEX\$_WS_LINKS
>	APEX\$_WS_NOTES
>	APEX\$_WS_ROWS
>	APEX\$_WS_TAGS
>	APEX\$_WS_WEBPG_SECTION_HISTORY
>	APEX\$_WS_WEBPG_SECTIONS
>	DEMO_CUSTOMERS
>	DEMO_ORDER_ITEMS
>	DEMO_ORDERS
>	DEMO_PRODUCT_INFO
>	DEMO_STATES
>	DEMO_USERS
>	DEPT
>	EMP
>	PERSON
>	PARAMEDIC
>	BLOOD
>	RECEIVER
>	DONATION_PLACE
>	PERSON_PHONE_NUMBER
>	DONOR
>	DONATION

Columns of BLOOD					
	Name	Type	Nullable	Default	Comments
1	SIGN	CHAR(1)			
2	TYPE	CHAR(2)			

Columns of PARAMEDIC					
	Name	Type	Nullable	Default	Comments
1	LICENSE_NUMBER	NUMBER(10)			
2	PERSON_ID	NUMBER(10)			

Columns of RECEIVER					
	Name	Type	Nullable	Default	Comments
1	HOSPITAL_ID	NUMBER(10)			
2	HOSPITAL_NAME	VARCHAR2(10)			
3	CITY	VARCHAR2(10)			
4	DELIVERY_DATE	DATE			
5	HOSPITAL_PHONE	NUMBER(10)			

Columns of PERSON					
	Name	Type	Nullable	Default	Comments
1	PERSON_ID	NUMBER(10)			
2	LAST_NAME	VARCHAR2(10)			
3	FIRST_NAME	VARCHAR2(10)			
4	EMAIL	VARCHAR2(25)	Y		

Columns of PERSON_PHONE_NUMBER					
	Name	Type	Nullable	Default	Comments
1	PHONE_NUMBER	NUMBER(10)			
2	PERSON_ID	NUMBER(10)			

Columns of DONOR					
	Name	Type	Nullable	Default	Comments
1	GENDER	CHAR(1)			
2	BIRTH_DATE	DATE			
3	PERSON_ID	NUMBER(10)			
4	SIGN	CHAR(1)			
5	TYPE	CHAR(2)			

Columns of DONATION_PLACE					
	Name	Type	Nullable	Default	Comments
1	PLACE_ID	NUMBER(10)			
2	PLACE_PHONE	NUMBER(10)			
3	MANAGER	VARCHAR2(20)			
4	ADDRESS	VARCHAR2(25)			

Columns of DONATION					
	Name	Type	Nullable	Default	Comments
1	DONATION_ID	NUMBER(10)			
2	PASS_TESTS	CHAR(1)			
3	DONATION_DATE	DATE			
4	VALID	CHAR(1)			
5	LICENSE_NUMBER	NUMBER(10)			
6	PERSON_ID_PARAMEDIC	NUMBER(10)			
7	PERSON_ID_DONOR	NUMBER(10)			
8	PLACE_ID	NUMBER(10)			
9	HOSPITAL_ID	NUMBER(10)			

קובץ dropTable -

כשכתבנו את הקובץ dropTables.sql סידרנו את סדר הטבלאות כך שקודם ימחקו הטבלאות שיש בהם מפתחות זרים לטבלאות אחרות ורק לאחר מכן הטבלאות שלא מסתמכות על טבלאות אחרות, כדי שתתאפשר המחיקה. (את הקובץ העלנו גם לגיט)

```
DROP TABLE Donation;

DROP TABLE Donor;

DROP TABLE Person_phone_number;

DROP TABLE Paramedic;

DROP TABLE Blood;

DROP TABLE Receiver;

DROP TABLE Donation_place;

DROP TABLE Person;
```

קובץ selectAll -

קובץ selectAll.sql מאפשר לנו לראות את כל הטבלאות שנוצרו עם הנתונים לאחר הכנסתם. (את הקובץ העלנו גם לגיט)

```
select * from Donation;

select * from Donor;

select * from Person_phone_number;

select * from Paramedic;

select * from Blood;

select * from Receiver;

select * from Donation_place;

select * from Person;
```

קובץ insertTable -

לאחר שהכנסנו את כל הנתונים בשלושה שיטות השונות, יצרנו את הקובץ insertTables.sql (את הקובץ העלנו לגיט), בקובץ זה כינסנו את כל הפקודות הכנסה של הפרויקט. את הפקודות ההכנסה של הפרויקט כינסנו לאחר שהכנסנו לכל הטבלאות נתונים ועשינו זאת כך:

1. לטבלאות Blood, Receiver ו- Person הכנסנו נתונים באמצעות הרצת קוד בפייתון (insertData.py שהעלנו לגיט) ושימוש בספריית Faker ע"מ להגדיל נתונים רנדומליים. הנתונים של הטבלאות Blood, Receiver ו- Person הוכנסו לתוך קובצי txt ולאחר מכן הוכנסו לטבלאות המתאימות בפרויקט.

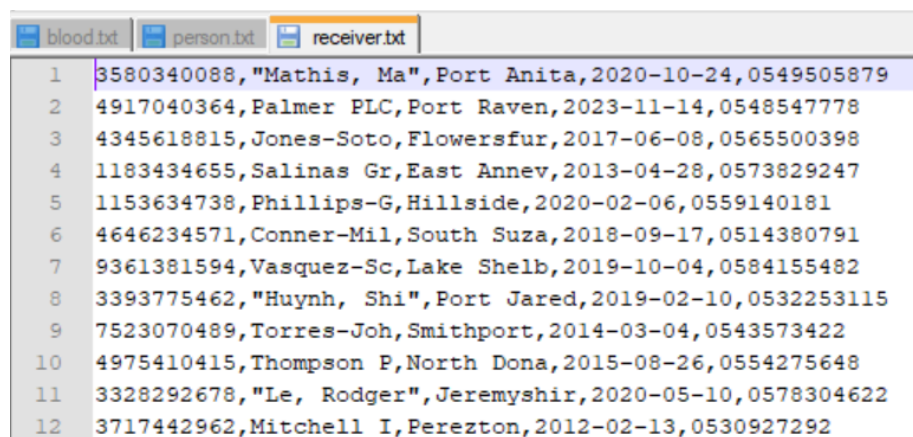
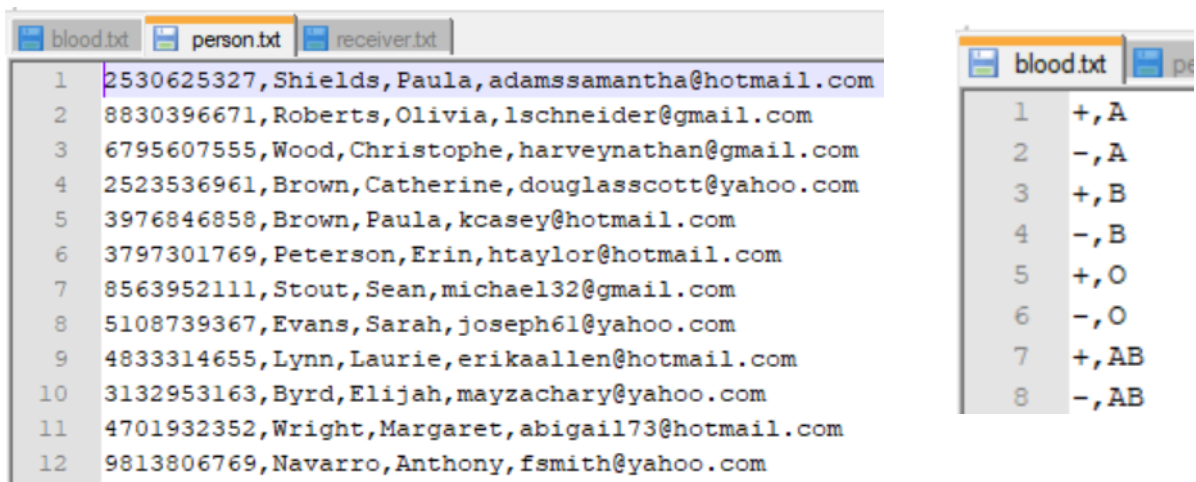
צילום מסך של חלק מהקוד-

```
# Generate 400 rows of random data for Receiver table
receiver_data = [
    {
        'hospital_id': unique_hospital_ids[i],
        'hospital_name': fake.company()[:max_length1],
        'city': fake.city()[:max_length1],
        'delivery_date': fake.date_between(start_date='-13y', end_date='today'),
        'hospital_phone': unique_phone_ids[i]
    }
    for i in range(400)
]

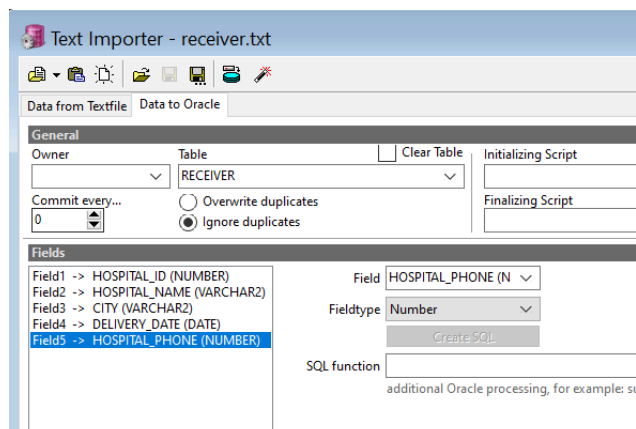
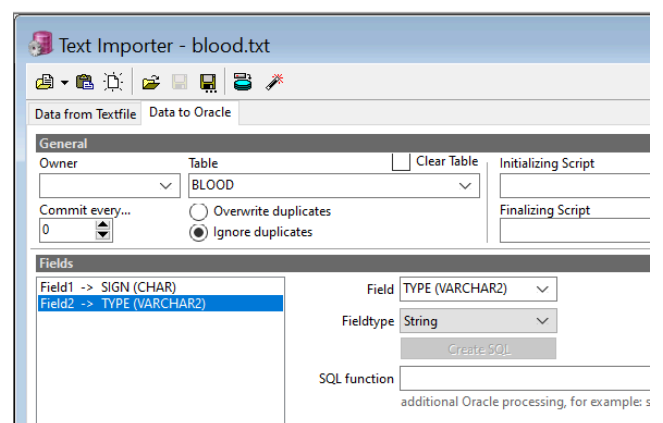
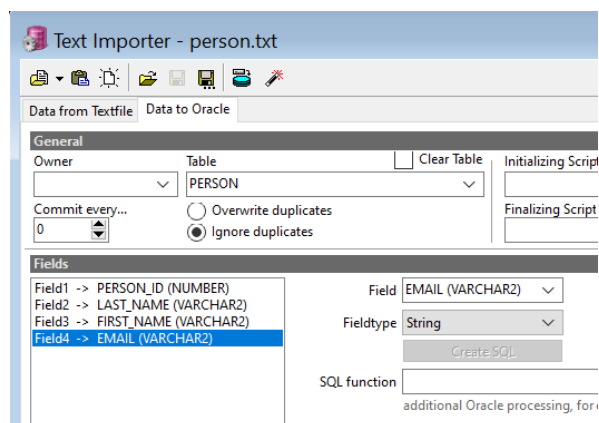
# Convert data to DataFrame
blood_df = pd.DataFrame(blood_data)
person_df = pd.DataFrame(person_data)
receiver_df = pd.DataFrame(receiver_data)

# Write DataFrames to TXT files for Blood and Person
blood_df.to_csv( path_or_buf: 'blood.txt', index=False, sep=',', header=False)
person_df.to_csv( path_or_buf: 'person.txt', index=False, sep=',', header=False)
receiver_df.to_csv( path_or_buf: 'receiver.txt', index=False, sep=',', header=False)
```

צילום מסך של חלק מקבצי txt שהתקבלו-



צילום מסך של חלק מתהליך הכנסת הקבצים לטבלאות באורקל-



2. בעזרת Mockaroo יצרנו קבצי SQL עבור הטבלה Donation_Place ושמרנו והכנסנו גם את הנתונים של הטבלה הזו לפרויקט.
(צילום מסך מאתר Mockaroo)

Field Name	Type	Options
place_id	Row Number	blank: 0 % Σ \times
phone_num	Phone	format: ##### ∇ blank: 0 % Σ \times
manager	Full Name	blank: 0 % Σ \times
address	Street Address	blank: 0 % Σ \times

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

Rows: 400 Format: SQL ∇ Table Name: DONATION_PLACE ☐ include CREATE TABLE

3. לטבלאות Donation, Donor, Person_phone_number ו- Paramedic הכנסנו נתונים בעזרת ה Data generator.

Data Generator - donor.gd

DONOR

Owner	Table	Number of records
SHOVAL	DONOR	400

Name	Type	Size	Data
GENDER	CHAR	1	List('F', 'M')
BIRTH_DATE	DATE		Random(1.1.1960, 1.5.2006)
PERSON_ID	NUMBER	10	List(select person_id from PERSON)
SIGN	CHAR	1	List(select sign from BLOOD)
TYPE	CHAR	2	List(select type from BLOOD)
*			

Data Generator - paramedic.gd

PARAMEDIC

Owner: SHOVAL Table: PARAMEDIC Number of records: 400

Name	Type	Size	Data
LICENSE_NUMBER	NUMBER	10	[1111111111]
SENIORITY	NUMBER	2	Random(1, 70)
PERSON_ID	NUMBER	10	List(select person_id from PERSON)
*			

Data Generator - person_phone_number.gd

PERSON_PHONE_NUMBER

Owner: SHOVAL Table: PERSON_PHONE_NUMBER Number of records: 600

Name	Type	Size	Data
PHONE_NUMBER	NUMBER	10	'05'+[1111111111]
PERSON_ID	NUMBER	10	List(select person_id from PERSON)
*			

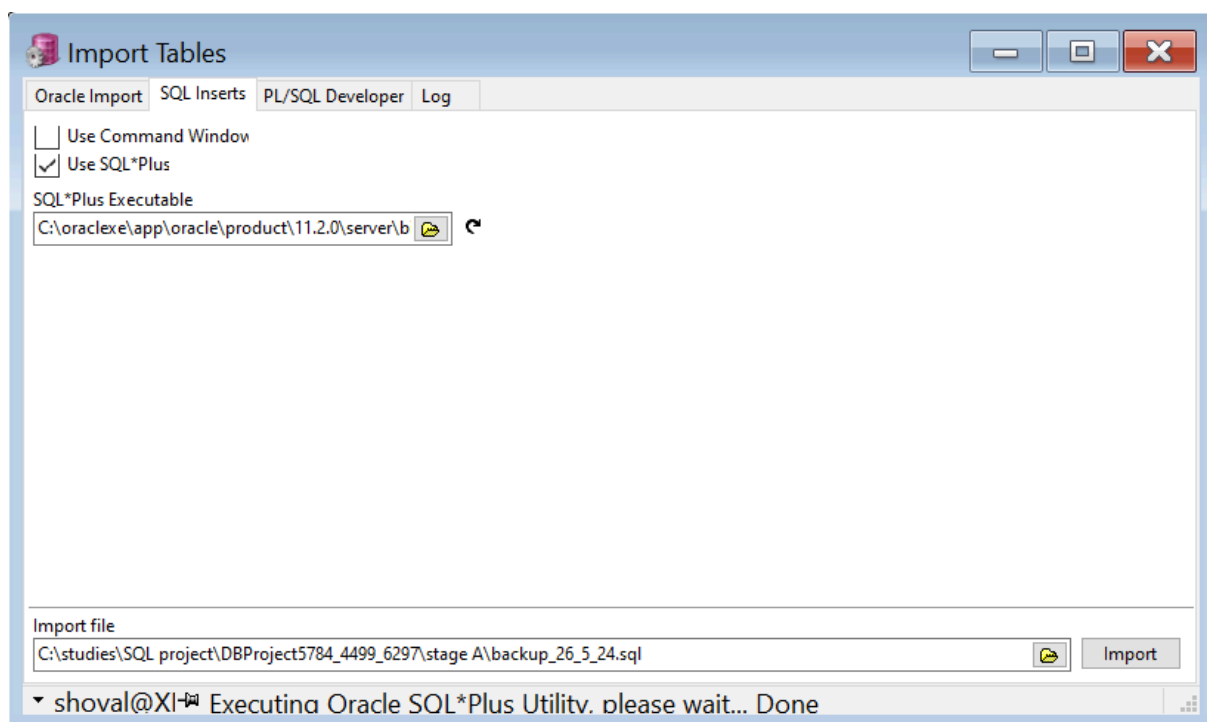
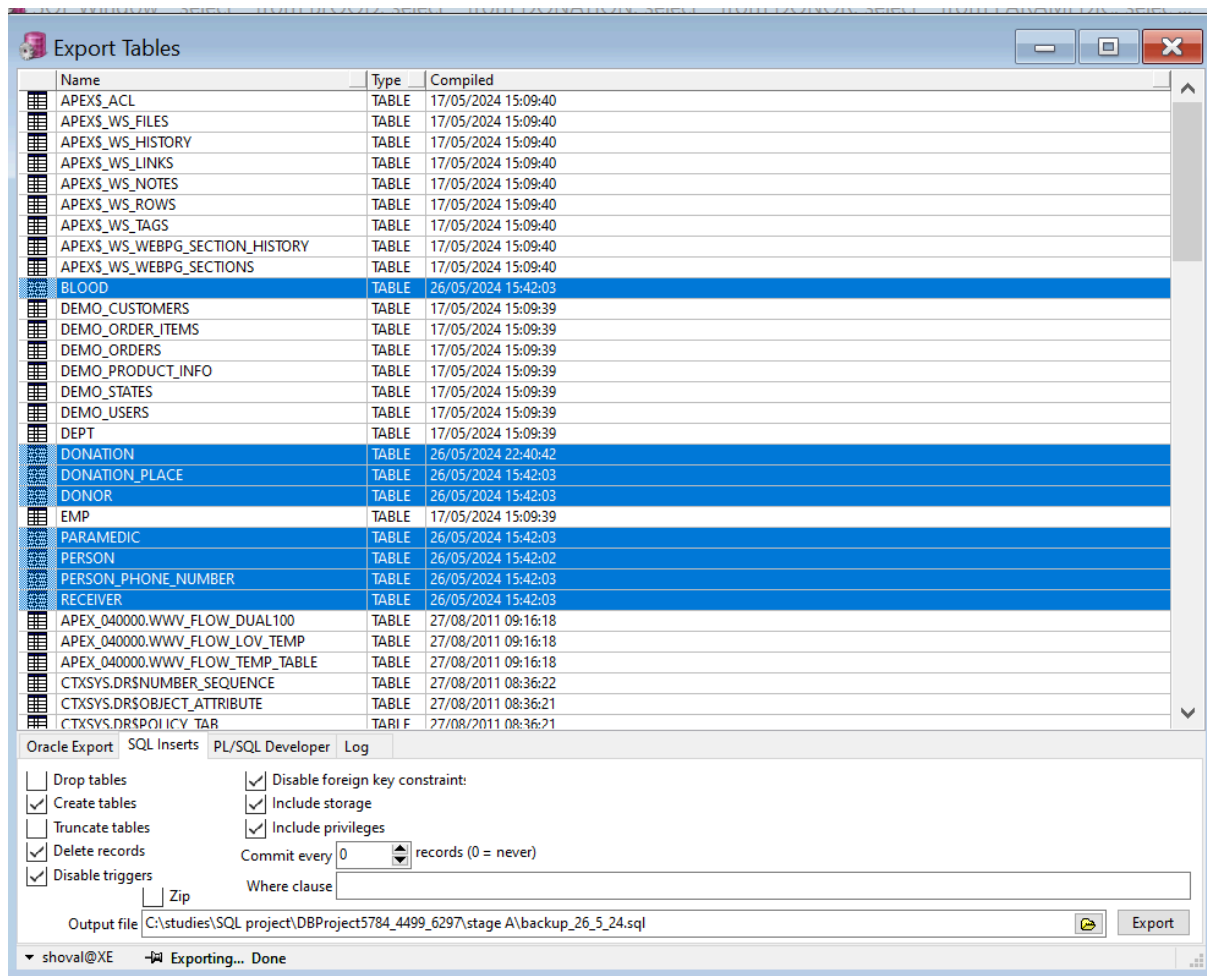
Data Generator - donation.gd

DONATION

Owner: SHOVAL Table: DONATION Number of records: 400

Name	Type	Size	Data
DONATION_ID	NUMBER	10	[1111111111]
PASS_TESTS	CHAR	1	List('Y', 'N')
DONATION_DATE	DATE		Random(1.1.2000, 1.5.2024)
VALID	CHAR	1	List('Y', 'N')
PERSON_ID_PARAMED	NUMBER	10	List(select person_id from PARAMEDIC)
PERSON_ID_DONOR	NUMBER	10	List(select person_id from DONOR)
PLACE_ID	NUMBER	10	List(select place_id from DONATION_PLACE)
HOSPITAL_ID	NUMBER	10	List(select hospital_id from RECEIVER)
*			

גיבוי ושחזור הנתונים:



שלב ב - שאלות ואילוצים

שאלות ללא פרמטרים:

(ניתן לראות את השאלות בקובץ Queries.sql)

• שאלות SELECT:

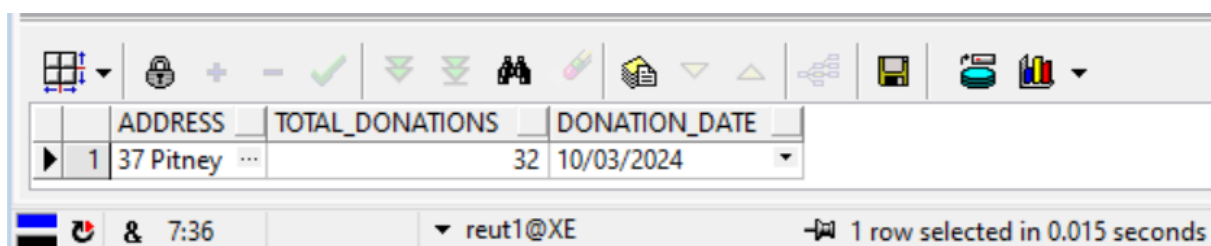
1. רוצים לדעת איפה נערכה ההתרמה הגדולה ביותר ביום יחיד בשנה האחרונה

וכמה תרומות דם נתרמו באותו יום בשביל הענקת פרס מטעם משרד הבריאות.

הקוד-

```
SELECT dp.address, daily_donations.donation_count AS total_donations,
       daily_donations.donation_date
FROM (
    SELECT d.place_id,
           d.donation_date,
           COUNT(*) AS donation_count
    FROM Donation d
    WHERE d.donation_date BETWEEN ADD_MONTHS(SYSDATE, -12) AND SYSDATE
    GROUP BY d.place_id, d.donation_date
    ORDER BY donation_count DESC
) daily_donations
JOIN Donation_place dp ON dp.place_id = daily_donations.place_id
WHERE ROWNUM = 1;
```

התוצאה -



	ADDRESS	TOTAL_DONATIONS	DONATION_DATE
1	37 Pitney	32	10/03/2024

2. מתרחשת התרמה ב 'Lily Ave 63', מספר תחנה: 69545, ורוצים לשלוח

הודעות אודות ההתרמה לאנשים שתרמו שם בעבר, אך שלא תרמו דם בשלושה חודשים האחרונים, כדי שיבואו לתרום.

הקוד -

```
SELECT P.pid AS person_id, P.f_name AS first_name, P.l_name AS last_name,
        LISTAGG (PPN.phone_number, ', ') WITHIN GROUP (ORDER BY
        PPN.phone_number) AS phone_numbers
FROM DONATION DN
JOIN DONOR D ON DN.donor_id = D.donor_id
JOIN PERSON P ON D.donor_id = P.pid
JOIN DONATION_PLACE DP ON DN.place_id = DP.place_id
JOIN PERSON_PHONE_NUMBER PPN ON P.pid = PPN.pid
WHERE DP.address = '63 Lily Ave'
      AND DP.place_id = 69545
      AND D.donor_id NOT IN (
        SELECT D1.donor_id
        FROM DONATION DN1
        JOIN DONOR D1 ON DN1.donor_id = D1.donor_id
        WHERE MONTHS_BETWEEN (SYSDATE, DN1.donation_date) <= 3
      )
GROUP BY P.pid, P.f_name, P.l_name;
```

התוצאה -

	PERSON_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBERS
1	004214981	Tracy	Gertner	053-209-8007, 053-209-8007
2	005988217	Rosanne	Chilton	051-411-2514, 051-411-2514, 051-411-2514, 051-411-2514
3	007507474	Jena	Horton	050-365-0841, 050-365-0841, 053-664-8631, 053-664-8631
4	014472246	Jim	Murdock	054-046-8463, 054-046-8463, 054-046-8463, 054-046-8463
5	025800738	Albertina	Creek	052-921-6102

161 rows selected in 0.061 seconds

3. עבור אדם שבא לתרום, נרצה לבדוק האם לא עברו 3 חודשים מהתרומה האחרונה שלו והאם יש יותר מ-4 תרומות שלו שלא עברו את הבדיקות ולכן הוא לא יכול לתרום (לפי תקנות משרד הבריאות).

הקוד -

```
SELECT P.f_name AS first_name,P.l_name AS last_name,
       D.type AS blood_type,D.sign AS rh_sign,
       (SELECT MAX(DN.donation_date)
        FROM DONATION DN
        WHERE DN.donor_id = P.pid) AS last_donation_date,
       CASE
         WHEN MONTHS_BETWEEN(SYSDATE, (SELECT MAX(DN.donation_date)
                                         FROM DONATION DN
                                         WHERE DN.donor_id = P.pid)) >= 3
         AND (SELECT COUNT(*)
              FROM DONATION DN
              WHERE DN.donor_id = P.pid
              AND DN.pass_tests = 'N') <= 4
         THEN 'Yes'
         ELSE 'No'
       END AS can_donate
FROM PERSON P
JOIN DONOR D ON P.pid = D.donor_id
WHERE P.pid = '238348956';
```

התוצאה -

	FIRST_NAME	LAST_NAME	BLOOD_TYPE	RH_SIGN	LAST_DONATION_DATE	CAN_DONATE
1	Howard	Adams	A	-	19/08/2023	No

1 row selected in 0.031 seconds

4. יש מחסור חמור בכל הארץ בדם מסוג 'O-', רוצים לקבל רשימה של כל האנשים עם סוג הדם הנ"ל, בקבוצות לפי מיקום תרומות קודמות שלהם כדי לשלוח להם בקשה לבוא לתרום, ולהבין איפה הכי כדאי לקיים התרמה לסוג דם הזה.

הקוד -

```
SELECT DP.address AS donation_place,

       LISTAGG(P.f_name || ' ' || P.l_name || ' (' || PPN.phone_number || ')',
              ', ') WITHIN GROUP (ORDER BY P.l_name) AS donors

FROM DONATION DN

JOIN DONOR D ON DN.donor_id = D.donor_id

JOIN PERSON P ON D.donor_id = P.pid

JOIN PERSON_PHONE_NUMBER PPN ON P.pid = PPN.pid

JOIN DONATION_PLACE DP ON DN.place_id = DP.place_id

WHERE D.sign LIKE '%+' AND D.type LIKE '%O%'

GROUP BY DP.address;
```

התוצאה -

	DONATION_PLACE	DONORS
1	1 Kyra Road	Geena Tillis (050-004-3135)
2	10 Albuquerque Ave	Quentin Addy (053-213-5158), Quentin Addy (055-765-3132)
3	11 Breslin Drive	Leslie Akins (054-898-5129)
4	11 Stockard Road	Ethan Roundtree (055-254-2641)
5	13rd Street	Boyd Seigny (058-862-5971)

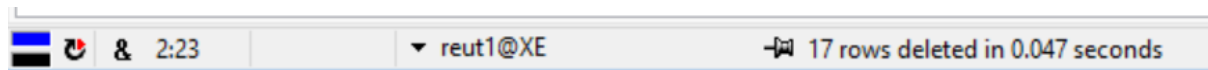
34 rows selected in 0.155 seconds

• שאלות DELETE:

1. רוצים למחוק מהרשומות את כל מנות הדם שרשום שנמסרו לפני שבכלל נלקחו בעקבות תקלה.

הקוד -

```
DELETE FROM Donation
WHERE donation_id IN (
    SELECT DN.donation_id
    FROM Donation DN
    JOIN Receiver R ON DN.hospital_id = R.hospital_id
    WHERE R.delivery_date < DN.donation_date
);
```



טבלת התרומות שבהן התרחשה התקלה לפני המחיקה -

SQL Window - SELECT * FROM Donation WHERE donation_id IN (SELECT DN.donation_id FROM Don...

SQL Output Statistics

```
SELECT *
FROM Donation
WHERE donation_id IN (
    SELECT DN.donation_id
    FROM Donation DN
    JOIN Receiver R ON DN.hospital_id = R.hospital_id
    WHERE R.delivery_date < DN.donation_date
);
```

	DONATION_ID	PASS_TESTS	DONATION_DATE	VALID	PARAMEDIC_ID	DONOR_ID	PLACE_ID
1	215841674	Y	24/03/2024	Y	624208805	853599144	91
2	298197542	Y	08/10/2023	Y	060138485	165323833	81
3	572476512	Y	06/02/2024	N	363254369	165323833	51
4	796985816	Y	17/12/2023	N	328635747	165323833	41
5	487472365	Y	12/12/2023	N	354683638	165323833	81

1:1 reut1@XE 17 rows selected in 0.438 seconds

טבלת התרומות שבהן התרחשה התקלה לאחר המחיקה -

A screenshot of a SQL execution bar. On the left, there are icons for SQL, Output, and Statistics. The status bar shows '0 rows selected in 0.032 seconds'.

2. רוצים למחוק מהרשומות את כל התורמים שגילם גדול מ-64 ולא תרמו בעשר השנים האחרונות ואת כל רשומות התרומות שלהם.

את פעולת המחיקה הזו חילקנו ל 2 פעולות מחיקה שונות.

המחיקה ראשונה היא מחיקת כל רשומות התרומות של תורם מעל גיל 64 שלא תרם ב-10 שנים האחרונות.

הקוד של מחיקה הראשונה -

```
DELETE FROM DONATION

WHERE donor_id IN (

    SELECT D.donor_id

    FROM DONOR D

    JOIN PERSON P ON D.donor_id = P.pid

    WHERE D.birth_date <= ADD_MONTHS(SYSDATE, -64*12)

    AND NOT EXISTS (

        SELECT 1

        FROM DONATION DN

        WHERE DN.donor_id = D.donor_id

        AND DN.donation_date >= ADD_MONTHS(SYSDATE, -120)

    )

);
```

מחיקה שנייה היא מחיקת התורם עצמו שמעל גיל 64 ושלא תרם דם ב-10 שנים האחרונות.

הקוד של המחיקה השנייה -

```
DELETE FROM DONOR

WHERE donor_id IN (

    SELECT D.donor_id

    FROM DONOR D

    JOIN PERSON P ON D.donor_id = P.pid
```

```

WHERE D.birth_date <= ADD_MONTHS(SYSDATE, -64*12)

AND NOT EXISTS (

SELECT 1

FROM DONATION DN

WHERE DN.donor_id = D.donor_id

AND DN.donation_date >= ADD_MONTHS(SYSDATE, -120)

)

);

```

טבלת התורמים מעל גיל 64 לפני המחיקה -

	PERSON_ID	FIRST_NAME	LAST_NAME	BIRTH_DATE
1	349734903	Chaka	Belushi	06/10/1959
2	853599144	Liv	Black	02/12/1959
3	165323833	Leslie	Broadbent	25/08/1953
4	608234327	Ian	Callow	09/12/1950
5	609830431	Lenny	Cronin	20/04/1954

15 rows selected in 0.031 seconds

טבלת התורמים מעל גיל 64 לאחר המחיקה -

	PERSON_ID	FIRST_NAME	LAST_NAME	BIRTH_DATE
1	349734903	Chaka	Belushi	06/10/1959
2	853599144	Liv	Black	02/12/1959
3	165323833	Leslie	Broadbent	25/08/1953

3 rows selected in 0.032 seconds

סה"כ היו 15 תורמים שגילם גדול מ 64 ומתוכם רק 3 המשיכו לתרום ב 10 שנים האחרונות.

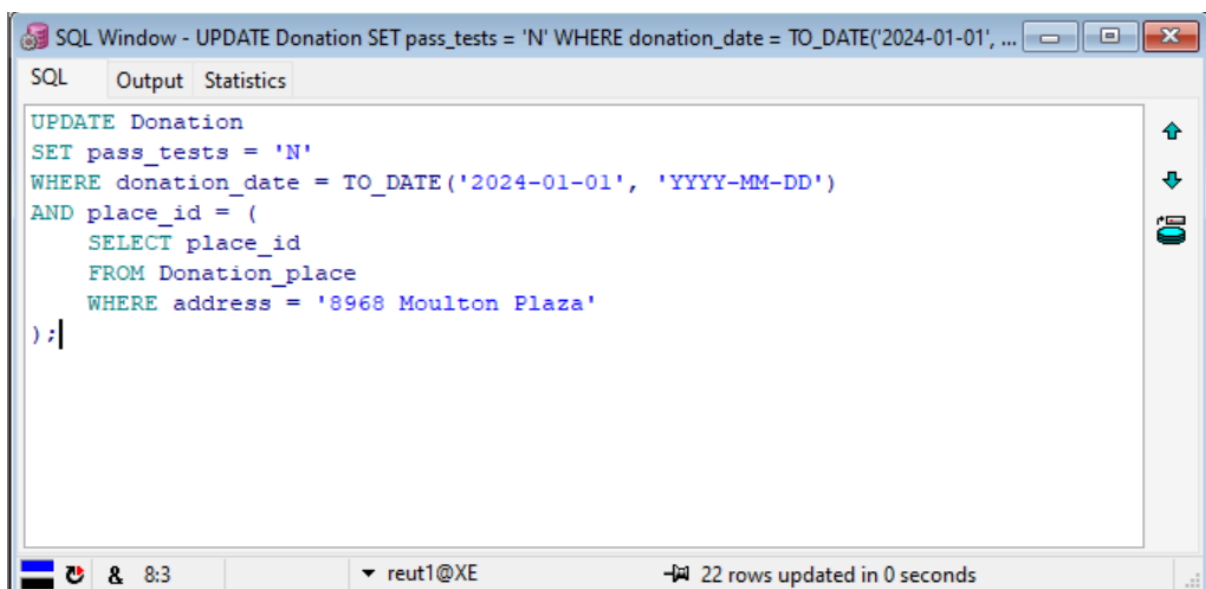
• שאלות UPDATE:

1. רוצים לסמן שהדם של כל האנשים שתרמו בתחנה מספר 69545 בתאריך 10.03.2024 לא עבר את הבדיקות כיוון שהייתה תקלה במקרר התרומות.

הקוד -

```
UPDATE Donation
SET pass_tests = 'Y', valid = 'N'
WHERE donation_date = TO_DATE('2024-01-01', 'YYYY-MM-DD')
AND place_id = (
    SELECT place_id
    FROM Donation_place
    WHERE address = '8968 Moulton Plaza'
);
```

התוצאה -



```
SQL Window - UPDATE Donation SET pass_tests = 'N' WHERE donation_date = TO_DATE('2024-01-01', ...
SQL Output Statistics
UPDATE Donation
SET pass_tests = 'N'
WHERE donation_date = TO_DATE('2024-01-01', 'YYYY-MM-DD')
AND place_id = (
    SELECT place_id
    FROM Donation_place
    WHERE address = '8968 Moulton Plaza'
);|
reut1@XE 22 rows updated in 0 seconds
```

ניתן לראות שפעולת העדכון בוצעה.

2. תחנת '82nd Street' רוצה לתת העלאה בשכר לפרמדיקים שערכו אצלה תרומות בשביל לעודד פרמדיקים חדשים לבוא לעבוד אצלם. כל פרמדיק שהוותק שלו גדול משנתיים והשתתף ביותר מ-6 תרומות בתחנה יקבל העלאה של 400 ש"ח בשכר.

הקוד -

```
UPDATE PARAMEDIC P

SET P.SALARY = P.SALARY + 400

WHERE P.PARAMEDIC_ID IN (

    SELECT D.PARAMEDIC_ID

    FROM DONATION D

    JOIN DONATION_PLACE DP ON D.PLACE_ID = DP.PLACE_ID

    WHERE DP.ADDRESS = '82nd Street'





    AND P.SENIORITY > 2

    GROUP BY D.PARAMEDIC_ID

    HAVING COUNT(D.DONATION_ID) > 6

);
```

תוצאת הרצת הקוד -

			11:3	▼ reut1@XE	 6 rows updated in 0.594 seconds
---	---	---	------	------------	---

ניתן לראות שפעולת העדכון בוצעה.

טבלת הפרמדיקים שהתרימו יותר מ 6 פעמים בתחנה לפני העדכון -

	FIRST_NAME	PARAMEDIC_ID	SALARY
1	Maceo	579667581	15300
2	Campbell	452597433	10000
3	Debra	408874915	7800
4	Amy	370632269	13200
5	Cherry	175321868	11900
6	Jimmie	036492757	15300

6 rows selected in 0.032 seconds

טבלת אותם פרמדיקים לאחר העדכון -

	FIRST_NAME	PARAMEDIC_ID	SALARY
1	Maceo	579667581	15700
2	Campbell	452597433	10400
3	Debra	408874915	8200
4	Amy	370632269	13600
5	Cherry	175321868	12300
6	Jimmie	036492757	15700

6 rows selected in 0.031 seconds

ניתן לראות שהמשכורת של ששת הפרמדיקים עלתה ב- 400 ש"ח.

שאלות עם פרמטרים:

(ניתן לראות את השאלות בקובץ ParamsQueries.sql)

1. כשמתרחשת התרמה רוצים לשלוח הודעות אודות ההתרמה לאנשים שתרמו שם בעבר, אך שלא תרמו דם בשלושה חודשים האחרונים, כדי שיבואו לתרום. לכן, השאלתה מקבלת כתובת של תחנת התרומה, מספר התחנה ותאריך שבו נקבעה ההתרמה ומחזירה את כל הפרטים לאנשים שצריך לשלוח להם הודעות.

הקוד -

```
SELECT P.pid AS person_id,

       P.f_name AS first_name,

       P.l_name AS last_name,

       LISTAGG (PPN.phone_number, ', ' ) WITHIN GROUP (ORDER BY
PPN.phone_number) AS phone_numbers

FROM DONATION DN

JOIN DONOR D ON DN.donor_id = D.donor_id

JOIN PERSON P ON D.donor_id = P.pid

JOIN DONATION_PLACE DP ON DN.place_id = DP.place_id

JOIN PERSON_PHONE_NUMBER PPN ON P.pid = PPN.pid

WHERE DP.address = &<name="Address" required="true" type="string">

      AND DP.place_id = &<name="Place id" required="true" type="string">

      AND D.donor_id NOT IN (

          SELECT D1.donor_id

          FROM DONATION DN1

          JOIN DONOR D1 ON DN1.donor_id = D1.donor_id

          WHERE MONTHS_BETWEEN(&<name="Donation date" required="true"
type="date">, DN1.donation_date) <= 3

      )

GROUP BY P.pid, P.f_name, P.l_name;
```

פרמטרים -

Variables

Name	Value
Address	63 Lily Ave
Place id	69545
Donation date	03/06/2024

OK
Cancel
Clear

התוצאות -

		PERSON_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBERS
	1	004214981	Tracy	Gertner	053-209-8007, 053-209-8007
	2	005988217	Rosanne	Chilton	051-411-2514, 051-411-2514, 051-411-2514, 051-411-2514
	3	007507474	Jena	Horton	050-365-0841, 050-365-0841, 053-664-8631, 053-664-8631
	4	014472246	Jim	Murdock	054-046-8463, 054-046-8463, 054-046-8463, 054-046-8463
	5	025800738	Albertina	Creek	052-921-6102

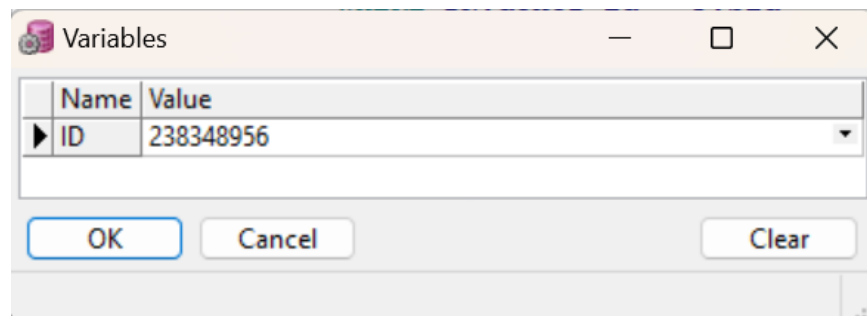
5:17
reut1@XE
155 rows selected in 0.266 seconds

2. עבור אדם שבא לתרום, נרצה לבדוק האם לא עברו 3 חודשים מהתרומה האחרונה שלו והאם יש יותר מ-4 תרומות שלו שלא עברו את הבדיקות ולכן הוא לא יכול לתרום (לפי תקנות משרד הבריאות).

הקוד -

```
SELECT P.f_name AS first_name,P.l_name AS last_name,
       D.type AS blood_type,D.sign AS rh_sign,
       (SELECT MAX(DN.donation_date)
        FROM DONATION DN
        WHERE DN.donor_id = P.pid) AS last_donation_date,
CASE
    WHEN MONTHS_BETWEEN(SYSDATE, (SELECT MAX(DN.donation_date)
                                   FROM DONATION DN
                                   WHERE DN.donor_id = P.pid)) >= 3
    AND (SELECT COUNT(*)
         FROM DONATION DN
         WHERE DN.donor_id = P.pid
         AND DN.pass_tests = 'N') <= 4
    THEN 'Yes'
    ELSE 'No'
END AS can_donate
FROM PERSON P
JOIN DONOR D ON P.pid = D.donor_id
WHERE P.pid = &<name="ID" type="string" required="true">;
```

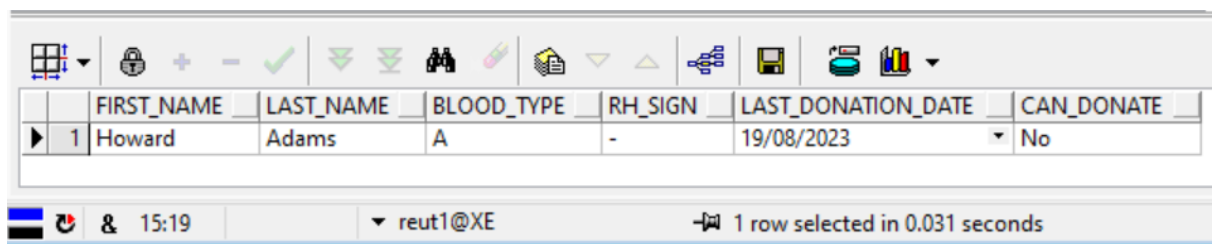
פרמטרים -



Name	Value
ID	238348956

OK Cancel Clear

התוצאות -



	FIRST_NAME	LAST_NAME	BLOOD_TYPE	RH_SIGN	LAST_DONATION_DATE	CAN_DONATE
1	Howard	Adams	A	-	19/08/2023	No

15:19 reut1@XE 1 row selected in 0.031 seconds

3. בית החולים מבקש שמות ומספרי טלפון של תורמים בעלי סוג דם מסוים שניתן לבחירה, בטווח גילאים מסויים שתרמו לפחות תרומה אחת שעברה את הבדיקות בשנה האחרונה.

הקוד -

```
SELECT  p.f_name AS First_name,  p.l_name AS Last_name,  pp.phone_number
FROM person p

        JOIN donor d ON p.pid = d.donor_id

        JOIN person_phone_number pp ON p.pid = pp.pid

WHERE  d.type=&<name="Blood type" list="SELECT DISTINCT type FROM BLOOD"
type="string">

        AND d.sign=&<name="Rh" list="SELECT DISTINCT sign FROM BLOOD"
type="string">

        AND ROUND(MONTHS_BETWEEN(SYSDATE, d.birth_date) / 12) BETWEEN

&<name="Min age" hint="The min age for donating is 17" required="true"
default="17">

AND &<name="Max age" required="true" default="60">

        AND EXISTS (SELECT 1

                        FROM    donation dn

                        WHERE    dn.donor_id = d.donor_id

                                AND dn.donation_date >=

                                TRUNC(SYSDATE) - INTERVAL '1' YEAR

                                AND dn.pass_tests = 'Y'

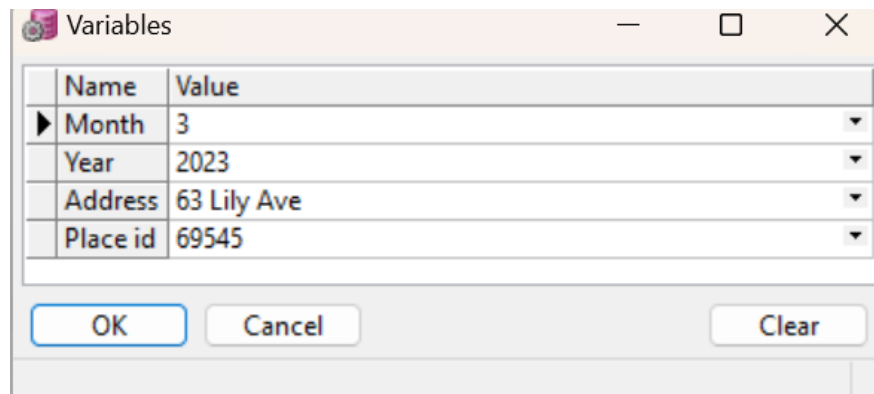
                                AND dn.valid = 'Y');
```


4. מנהלי תחנות התרומה ביקשו שיהיה להם אפשרות לראות עבור כל חודש איזה פרמדיקים התרימו אצלם בתחנה וכמה תרומות כל אחד מהם עשה.

הקוד -

```
SELECT p.f_name AS last_name, p.l_name AS first_name,
       p.pid AS paramedic_id, COUNT(d.donation_id) AS donation_count
FROM   person p
       JOIN PARAMEDIC pr ON p.pid = pr.paramedic_id
       JOIN DONATION d ON pr.paramedic_id = d.paramedic_id
       JOIN DONATION_PLACE dp ON d.place_id = dp.place_id
WHERE  EXTRACT(MONTH FROM d.donation_date) = &<name="Month" required="true"
type="integer">
       AND EXTRACT(YEAR FROM d.donation_date) = &<name="Year" required="true"
type="integer">
       AND dp.address = &<name="Address" required="true" type="string">
       AND dp.place_id = &<name="Place id" required="true" type="string">
GROUP BY p.f_name, p.l_name, p.pid
HAVING COUNT(d.donation_id) > 0
ORDER BY donation_count DESC;
```

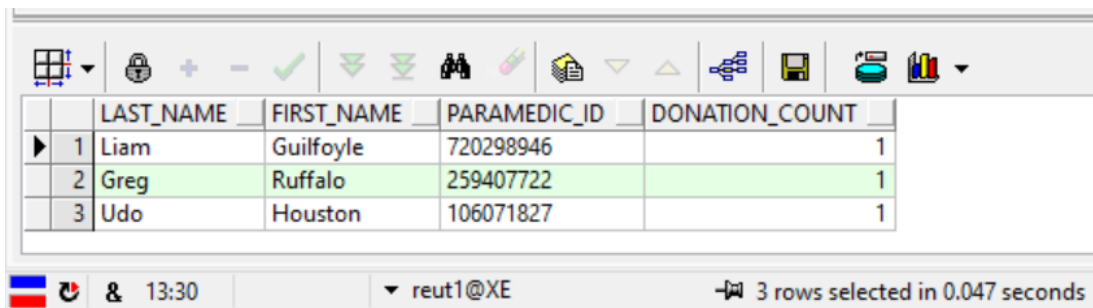
פרמטרים -



A dialog box titled "Variables" with a table of parameters. The table has two columns: "Name" and "Value". The parameters are: Month (3), Year (2023), Address (63 Lily Ave), and Place id (69545). Below the table are buttons for "OK", "Cancel", and "Clear".

Name	Value
Month	3
Year	2023
Address	63 Lily Ave
Place id	69545

התוצאות -



A screenshot of a database query results window. It shows a table with 5 columns: an index column, LAST_NAME, FIRST_NAME, PARAMEDIC_ID, and DONATION_COUNT. There are 3 rows of data. The second row is highlighted in green. The status bar at the bottom indicates "3 rows selected in 0.047 seconds".

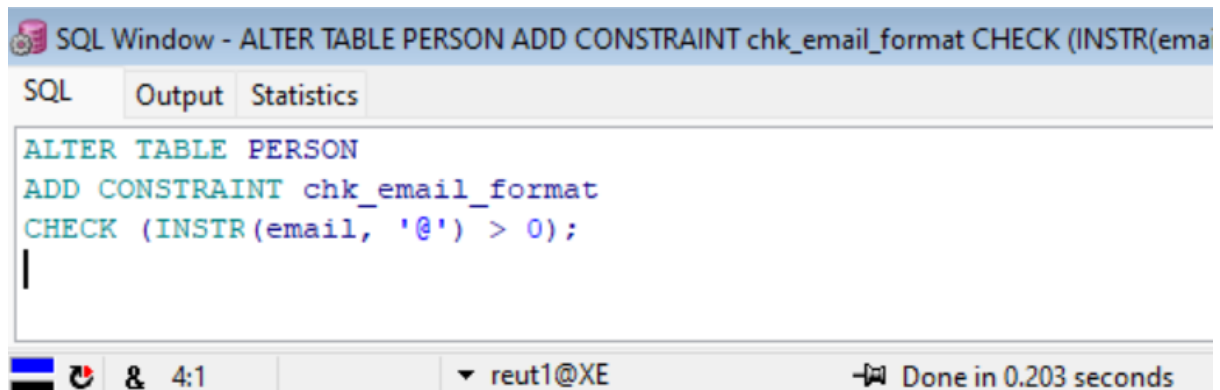
	LAST_NAME	FIRST_NAME	PARAMEDIC_ID	DONATION_COUNT
1	Liam	Guilfoyle	720298946	1
2	Greg	Ruffalo	259407722	1
3	Udo	Houston	106071827	1

אילוצים:

(ניתן לראות את השאלות בקובץ Constraints.sql)

1. אילוץ על טבלת PERSON -

כאשר מכניסים כתובת אימייל לטבלה של PERSON, ייבדק שהיא כוללת '@'.



SQL Window - ALTER TABLE PERSON ADD CONSTRAINT chk_email_format CHECK (INSTR(email, '@') > 0);

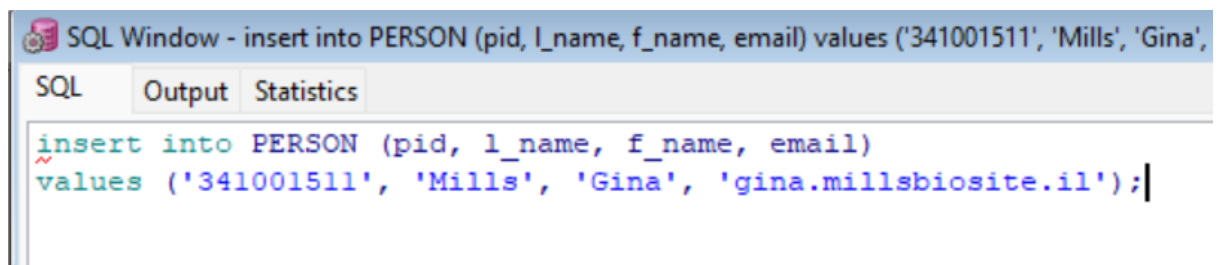
```

ALTER TABLE PERSON
ADD CONSTRAINT chk_email_format
CHECK (INSTR(email, '@') > 0);

```

Done in 0.203 seconds

בדיקה -



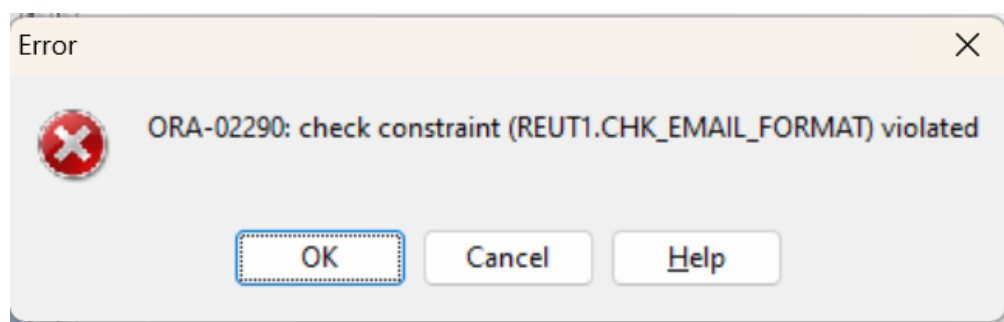
SQL Window - insert into PERSON (pid, l_name, f_name, email) values ('341001511', 'Mills', 'Gina', 'gina.millsbiosite.il');

```

insert into PERSON (pid, l_name, f_name, email)
values ('341001511', 'Mills', 'Gina', 'gina.millsbiosite.il');

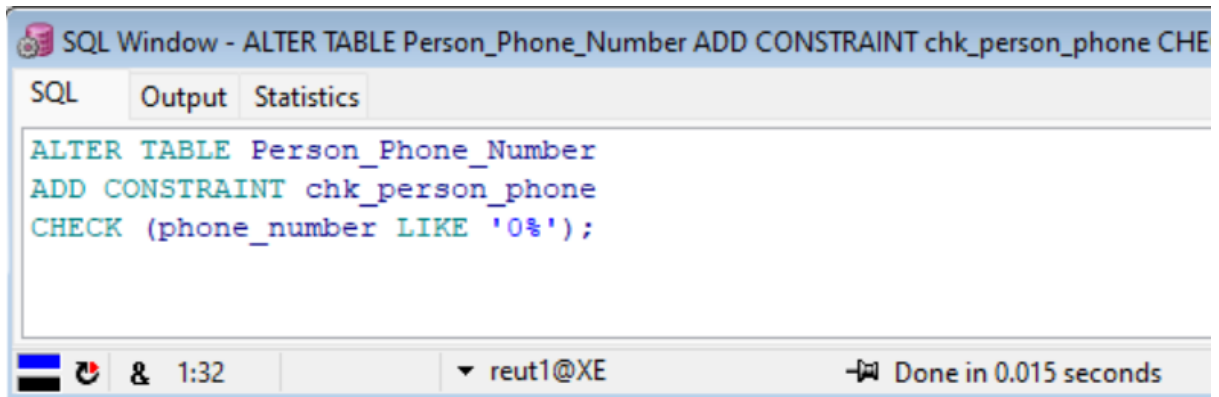
```

התקבלה שגיאה -



2. אילויז על טבלת PERSON_PHONE_NUMBER -

כאשר מכניסים מספר טלפון לטבלה של PERSON_PHONE_NUMBER, ייבדק שהוא מתחיל ב-0.



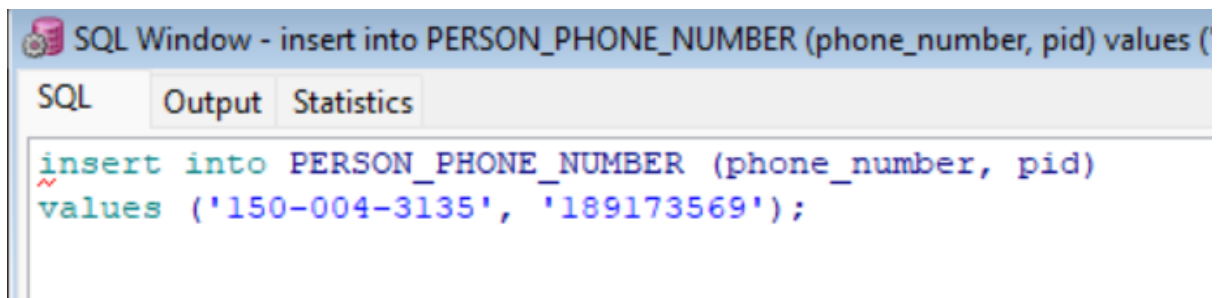
SQL Window - ALTER TABLE Person_Phone_Number ADD CONSTRAINT chk_person_phone CHE

SQL Output Statistics

```
ALTER TABLE Person_Phone_Number
ADD CONSTRAINT chk_person_phone
CHECK (phone_number LIKE '0%');
```

1:32 reut1@XE Done in 0.015 seconds

בדיקה -

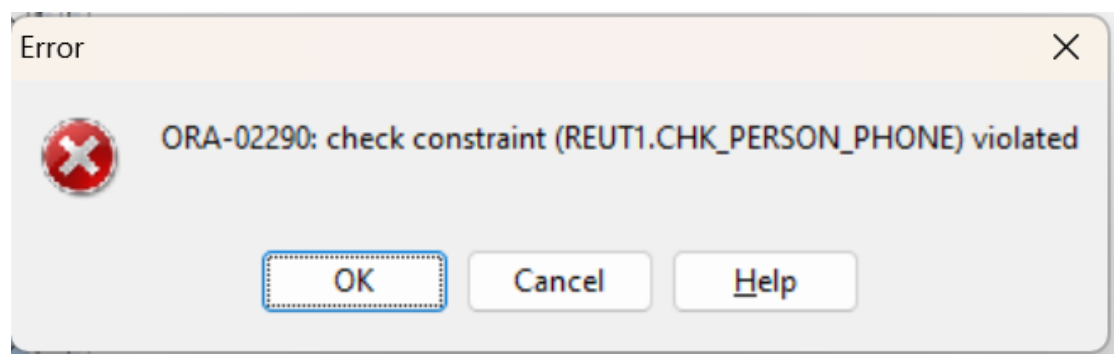


SQL Window - insert into PERSON_PHONE_NUMBER (phone_number, pid) values (

SQL Output Statistics

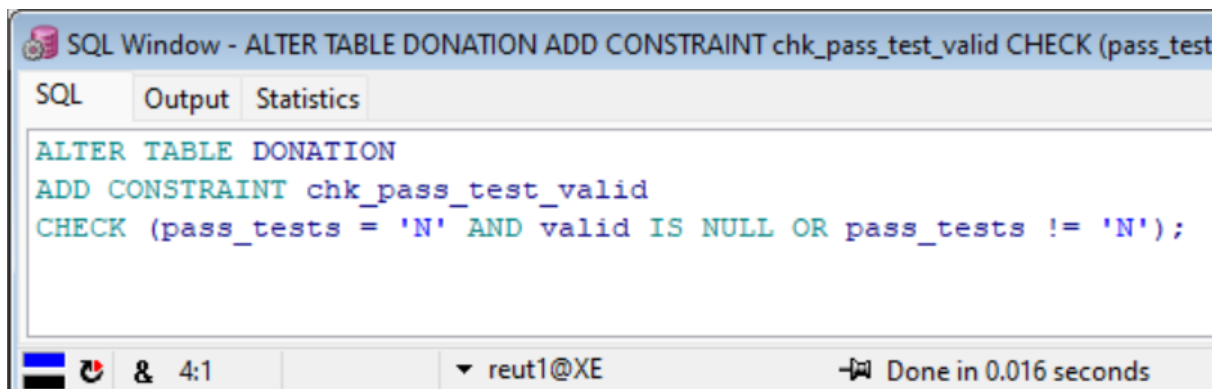
```
insert into PERSON_PHONE_NUMBER (phone_number, pid)
values ('150-004-3135', '189173569');
```

התקבלה שגיאה -



3. אילון על טבלת DONATION -

כאשר מכניסים תרומה לטבלה של DONATION, אם היא עוד לא עברה בדיקות, צריך שהשדה valid יהיה שווה ל NULL.

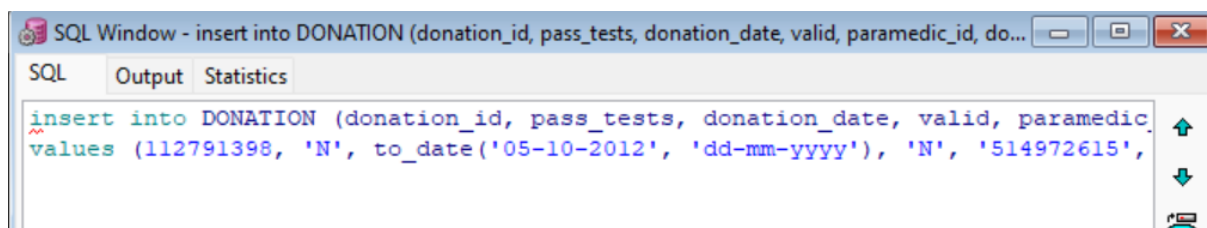


SQL Window - ALTER TABLE DONATION ADD CONSTRAINT chk_pass_test_valid CHECK (pass_test

```
SQL      Output  Statistics
ALTER TABLE DONATION
ADD CONSTRAINT chk_pass_test_valid
CHECK (pass_tests = 'N' AND valid IS NULL OR pass_tests != 'N');
```

reut1@XE Done in 0.016 seconds

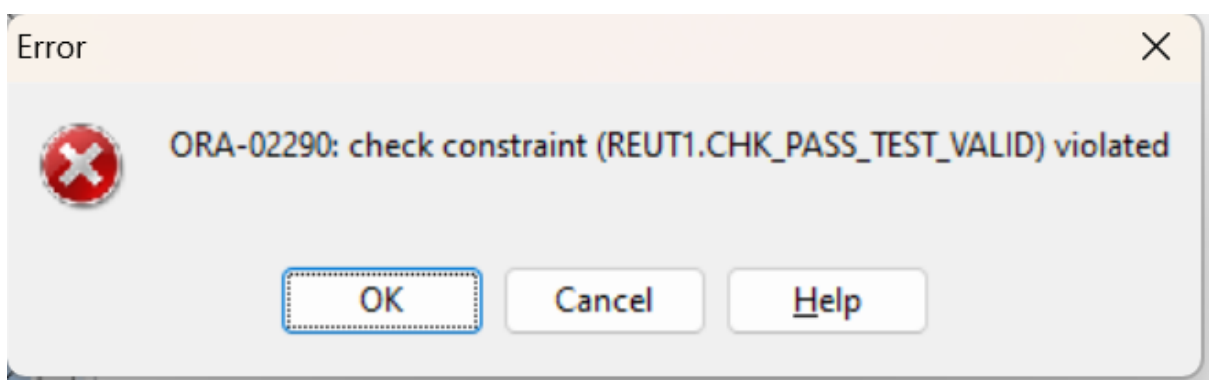
בדיקה -



SQL Window - insert into DONATION (donation_id, pass_tests, donation_date, valid, paramedic_id, do...

```
SQL      Output  Statistics
insert into DONATION (donation_id, pass_tests, donation_date, valid, paramedic_id, do...
values (112791398, 'N', to_date('05-10-2012', 'dd-mm-yyyy'), 'N', '514972615',
```

התקבלה שגיאה -



שינויים:

(ניתן גם לראות בקובץ AlterTables.sql)

חוץ מהשינויים לאילוצים עשינו רק שינוי אחד שהוא הוספה למטבלת PARAMEDIC ערך חדש של שכר.

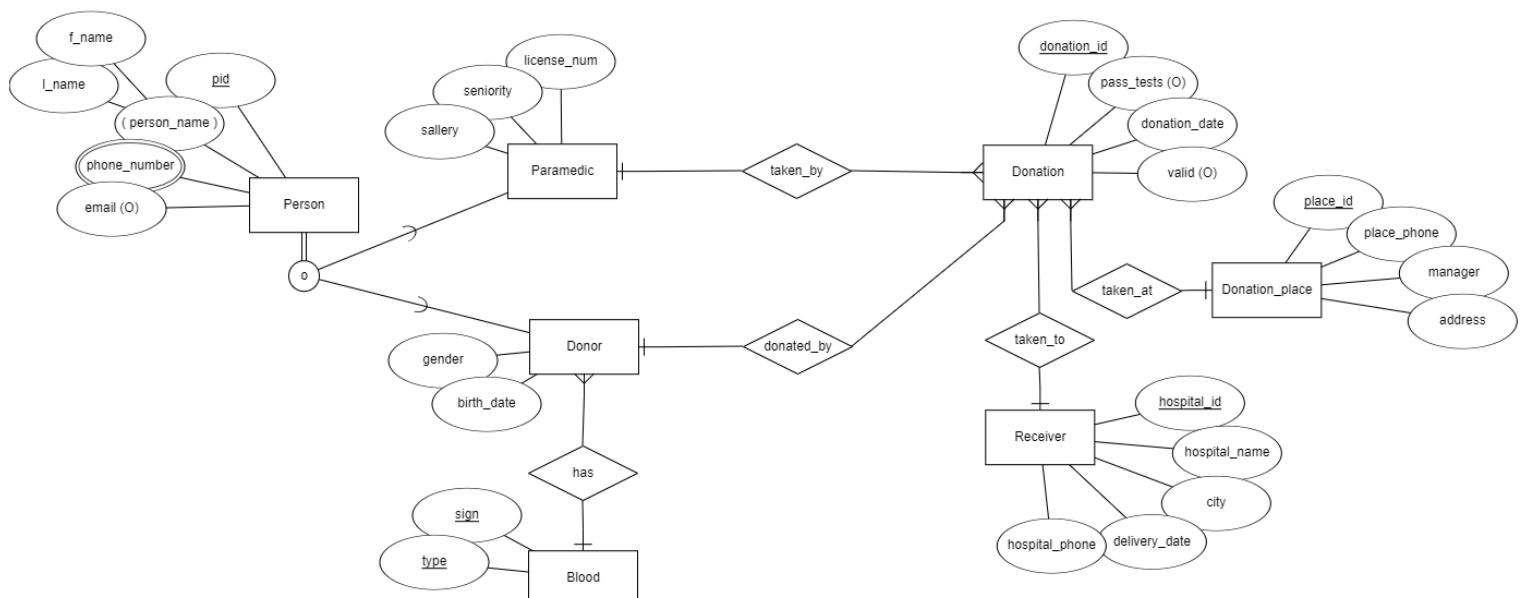
הקוד -

```
ALTER TABLE Paramedic
ADD salary NUMERIC(6) DEFAULT 0;
```

המטבלה לאחר הפקודה -

Type owner	Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments
	LICENSE_NUM	<input type="checkbox"/>	NUMBER(5)	<input type="checkbox"/>	...		
	SENIORITY	<input type="checkbox"/>	NUMBER(2)	<input checked="" type="checkbox"/>	0	...	
	PARAMEDIC_ID	<input type="checkbox"/>	VARCHAR2(9)	<input type="checkbox"/>	...		
	SALARY	<input type="checkbox"/>	NUMBER(6)	<input checked="" type="checkbox"/>	0	...	
*		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	

מטבלת ERD לאחר השינויים -



פעולת Commit ופעולת RollBack:

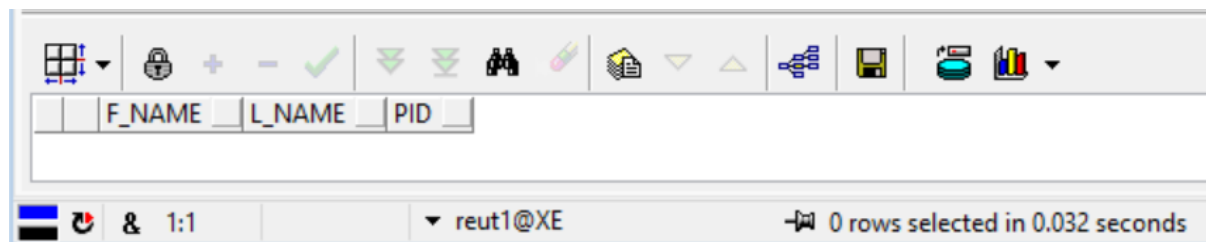
פעולת Commit משמשת לשמור באופן קבוע את כל השינויים שבוצעו במסד הנתונים במסגרת העסקה הנוכחית. לאחר ביצוע הפעולה, השינויים הופכים קבועים וזמינים למשתמשים אחרים במסד הנתונים.

פעולת RollBack משמשת לבטל את כל השינויים שבוצעו במסגרת העסקה הנוכחית ולהחזיר את מסד הנתונים למצבו הקודם לפני תחילת העסקה. זה מאפשר לך לבטל שינויים אם התגלתה שגיאה או אם החלטת לא לשמור את השינויים שביצעת.

דוגמת קוד-

```
SELECT f_name, l_name, pid
FROM PERSON
WHERE pid = 276484571
```

לאחר הרצה:

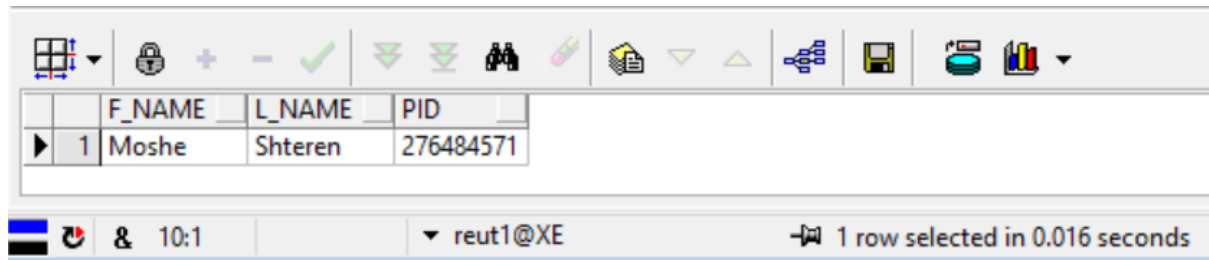


```
commit;

insert into person (PID, L_NAME, F_NAME, EMAIL)
values (276484571, 'Shteren', 'Moshe', 'moshe123@gmail.com');

SELECT f_name, l_name, pid
FROM PERSON
WHERE pid = 276484571
```

לאחר הרצה:



The screenshot shows the SQL Developer interface. At the top is a toolbar with various icons for editing and viewing data. Below the toolbar is a table with the following data:

	F_NAME	L_NAME	PID
1	Moshe	Shteren	276484571

At the bottom of the window, the status bar indicates: 10:1, reut1@XE, and 1 row selected in 0.016 seconds.

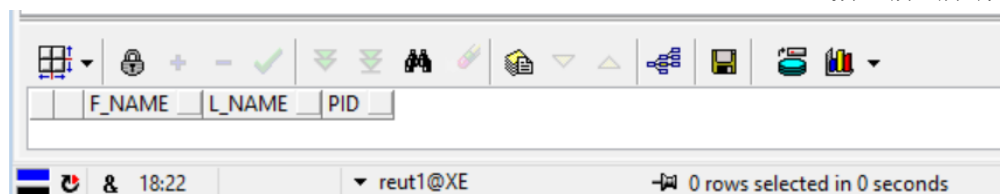
```
rollback;
```

```
SELECT f_name, l_name, pid
```

```
FROM PERSON
```

```
WHERE pid = 276484571
```

לאחר הרצה:



The screenshot shows the SQL Developer interface after a rollback. The table is empty. The status bar at the bottom indicates: 18:22, reut1@XE, and 0 rows selected in 0 seconds.

	F_NAME	L_NAME	PID
--	--------	--------	-----

שלב ג -

התוכניות:

1. חוקרים רצו לעשות מחקר סטטיסטי על תורמי הדם, במחקר הם רצו לדעת כמה

דברים על סוגי דם שונים וגילאי התורמים:

- מספר התרומות שהתקבלו (עברו את הבדיקות).
- היחס בין מספר התרומות שהתקבלו לבין למספר התרומות מאותו סוג שלא התקבלו.
- היחס בין הבדיקות שהתקבלו לאלו שלא.
- מספר התרומות שהתקבלו ע"י נשים.
- מספר התרומות שהתקבלו ע"י גברים.
- היחס בין התרומות שהתקבלו ע"י הנשים לאלו ע"י הגברים.
- הגיל הכי גדול תרומתו שהתקבלה.

תוכנית זו רושמת דו"ח מפורט על כל הנתונים שהחוקרים רצו ומדפיסה אותו בצורה נוחה להבנה, לאחר השגת המידע התוכנית מחזירה ע"י פרוצדורה (הפרוצדורה) תרומות דם מסוימות בחזרה לבנק הדם כדי שהחוקרים יוכלו לחקור אותם יותר ומדפיסה את מספר הזהות של תורמי תרומות הדם האלו.
(ניתן לראות את התוכנית בקובץ MainProgram1.sql)

נעזרנו בפונקציה `get_donation_statistics` שמקבלת את הקלט מהמשתמש (סוג דם, סימן, גיל מקסימלי ומינימלי של תורם) ומחזירה סמן ששומר את חישובי היחסים (סטטיסטיקות) שנעשו על הנתונים שהתקבלו.

נעזרנו בפרוצדורה `return_blood_to_bank` שמקבלת סמן ואת הנתונים שהכנסנו לפונקציה, הפרוצדורה מחליטה איזה תרומות להחזיר לבנק הדם ע"י בדיקת סכום הגדול מבין התרומות שעברו את המבחנים לבין אלו שלא ובנוסף בדיקת סכום הגדול מבין מספר הגברים לנשים, מתוכם תחזיר 5 או פחות תרומות לבנק.

הקוד של התוכנית -

```

DECLARE

    -- Declaring variables

    v_blood_type blood.type%type;

    v_blood_sign blood.sign%type;

    v_min_age NUMBER;

    v_max_age NUMBER;

    v_decision VARCHAR2(1);


    v_ratio_invalid NUMBER;

    v_ratio_gender NUMBER;


    -- Cursor declaration

    v_cursor SYS_REFCURSOR;

    v_total_received NUMBER;

    v_total_invalid NUMBER;

    v_total_valid NUMBER;

    v_total_females NUMBER;

    v_total_males NUMBER;

    v_max_age_donated NUMBER;


BEGIN

    -- Getting user input for blood type

    v_blood_type :=&<name="Blood type" list="SELECT DISTINCT type FROM BLOOD"
type="string">;

    -- Getting user input for blood sign

    v_blood_sign :=&<name="Blood sign" list="SELECT DISTINCT sign FROM BLOOD"
type="string">;

```

```

-- Getting user input for age range and decision

v_min_age := &<name="min age">;

v_max_age := &<name="max age">;

-- Getting cursor from function to fetch donors

v_cursor := get_donation_statistics(v_blood_type, v_blood_sign,
v_min_age, v_max_age);

-- Proceed if cursor is not null

IF v_cursor IS NOT NULL THEN

-- Fetching data from cursor

FETCH v_cursor INTO

    v_total_received,

    v_total_invalid,

    v_total_valid,

    v_total_females,

    v_total_males,

    v_max_age_donated;

-- Calculating ratio of received to invalid donations

IF v_total_invalid > 0 THEN

    v_ratio_invalid := v_total_received / v_total_invalid;

ELSE

    v_ratio_invalid := 0;

END IF;

-- Calculating ratio of donations received (females to males)

```

```

IF v_total_males > 0 THEN
    v_ratio_gender := v_total_females / v_total_males;
ELSE
    v_ratio_gender := 0;
END IF;

-- Printing the report

DBMS_OUTPUT.PUT_LINE('Blood Donation Report');
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('Blood Type: ' || v_blood_type || v_blood_sign);
DBMS_OUTPUT.PUT_LINE('Age Range: ' || v_min_age || ' to ' ||
v_max_age);
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('    Total donations That did pass the tests: ' ||
v_total_received);
DBMS_OUTPUT.PUT_LINE('    Total donations That did not pass the tests:
' || v_total_invalid);
DBMS_OUTPUT.PUT_LINE('    Ratio of pass to not pass: ' ||
TO_CHAR(v_ratio_invalid, '999.99'));
DBMS_OUTPUT.PUT_LINE('    Total donations that passed by females: ' ||
v_total_females);
DBMS_OUTPUT.PUT_LINE('    Total donations that passed by males: ' ||
v_total_males);
DBMS_OUTPUT.PUT_LINE('    Ratio of donations passed (females to males):
' || TO_CHAR(v_ratio_gender, '999.99'));
DBMS_OUTPUT.PUT_LINE('    The max Age that donated and passed the
tests: ' || v_max_age_donated);
DBMS_OUTPUT.PUT_LINE('');

-- Returning blood ( decision by a procedure)

return_blood_to_bank(v_cursor,v_blood_type, v_blood_sign, v_min_age,
v_max_age);

```

```

ELSE

    DBMS_OUTPUT.PUT_LINE('No donations found for the specified
criteria.');
```

END IF;

EXCEPTION

WHEN OTHERS THEN

```

    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

    ROLLBACK;

END;
```

פרמטרים -

Name	Value
Blood type	B
Blood sign	A
nim age	B
max age	O
	AB

Name	Value
Blood type	B
Blood sign	+
nim age	20
max age	40

תוצאת הרצה -

SQL Output Statistics

Clear Buffer size 10000 ☒ Enabled

Blood Donation Report

Blood Type: B+

Age Range: 20 to 40

Total donations That did pass the tests: 123

Total donations That did not pass the tests: 26

Ratio of pass to not pass: 4.73

Total donations that passed by females: 14

Total donations that passed by males: 11

Ratio of donations passed (females to males): 1.27

The max Age that donated and passed the tests: 40

Valid Female Donation returned: ID - 652327528

Valid Female Donation returned: ID - 684744404

Valid Female Donation returned: ID - 684744404

Valid Female Donation returned: ID - 616117735

Valid Female Donation returned: ID - 616117735

בדיקת שינויי בבסיס הנתונים -

טבלת תרומות הדם הנמצאות בבנק הדם לפני הרצת התוכנית

SQL Output Statistics

```
select * from Donation
where Hospital_id is null
and Pass_tests = 'Y';
```

	DONATION_ID	PASS_TESTS	DONATION_DATE	VALID	PARAMEDIC_ID	DONOR_ID	PLACE_ID	HOSPITAL_ID
1	732984353	Y	18/05/2012	Y	879327263	333334713	69545	
2	158382115	Y	10/07/2020	Y	080849238	495883869	69545	
3	282766864	Y	24/04/2012	Y	018933518	943811449	69545	
4	656741648	Y	22/05/2016	N	915523925	433660379	69545	
5	814497662	Y	13/04/2012	Y	984918806	160587152	69545	
6	379453987	Y	29/10/2014	N	999950098	333334713	69545	
7	654523321	Y	29/01/2011	Y	478202044	957568433	69545	
8	491761878	Y	16/11/2011	N	135639362	160587152	69545	
9	184733863	Y	30/11/2020	N	338160998	484771335	69545	
10	693229839	Y	18/03/2013	N	102771731	165537016	69545	
11	351453115	Y	02/10/2016	N	158592795	004214981	69545	
12	565637851	Y	23/01/2021	Y	175321868	901831943	69545	
13	159914895	Y	06/02/2014	Y	696511331	289180854	69545	
14	422195799	Y	05/10/2016	N	144998512	349136279	69545	
15	964538482	Y	08/09/2012	N	014700822	943811449	69545	
16	516812238	Y	15/12/2010	Y	319137744	433660379	69545	
17	818175367	Y	29/12/2011	N	478202044	004214981	69545	
18	595282789	Y	26/04/2012	Y	613689167	134423569	69545	
19	282527877	Y	19/04/2021	N	617833949	151254801	69545	

3:16 shoval@XE 169 rows selected in 0.172 seconds

הטבלה לאחר הרצת התוכנית -

SQL Output Statistics								
<pre>select * from Donation where Hospital_id is null and Pass_tests = 'Y';</pre>								
	DONATION_ID	PASS_TESTS	DONATION_DATE	VALID	PARAMEDIC_ID	DONOR_ID	PLACE_ID	HOSPITAL_ID
1	732984353	Y	18/05/2012	Y	879327263	333334713	69545	
2	158382115	Y	10/07/2020	Y	080849238	495883869	69545	
3	282766864	Y	24/04/2012	Y	018933518	943811449	69545	
4	656741648	Y	22/05/2016	N	915523925	433660379	69545	
5	814497662	Y	13/04/2012	Y	984918806	160587152	69545	
6	379453987	Y	29/10/2014	N	999950098	333334713	69545	
7	654523321	Y	29/01/2011	Y	478202044	957568433	69545	
8	491761878	Y	16/11/2011	N	135639362	160587152	69545	
9	446753579	Y	14/12/2011	Y	897793693	579628336	69545	
10	184733863	Y	30/11/2020	N	338160998	484771335	69545	
11	693229839	Y	18/03/2013	N	102771731	165537016	69545	
12	125331463	Y	28/02/2019	Y	159478573	837205710	69545	
13	351453115	Y	02/10/2016	N	158592795	004214981	69545	
14	565637851	Y	23/01/2021	Y	175321868	901831943	69545	
15	159914895	Y	06/02/2014	Y	696511331	289180854	69545	
16	422195799	Y	05/10/2016	N	144998512	349136279	69545	
17	479997782	Y	08/07/2013	Y	725569007	482138217	69545	
18	964538482	Y	08/09/2012	N	014700822	943811449	69545	
19	516812238	Y	15/12/2010	Y	319137744	433660379	69545	

ניתן לראות שלאחרת הרצת התוכנית מספר תרומות הדם שנמצאות בבנק הדם גדל.

הפרוצדורה return_blood_to_bank -

מקבלת מצביע (SYS_REFCURSOR) ואת הנתונים שהכנסנו לפונקציה. הפרוצדורה מחליטה איזה תרומות להחזיר לבנק הדם ע"י בדיקת סכום הגדול מבין התרומות שעברו את המבחנים לבין אלו שלא ובנוסף בדיקת סכום הגדול מבין מספר הגברים לנשים, מתוכם תחזיר 5 או פחות תרומות לבנק.

```
CREATE OR REPLACE PROCEDURE return_blood_to_bank (
    v_donors_cursor SYS_REFCURSOR,
    p_blood_type VARCHAR2,
    p_blood_sign VARCHAR2,
    p_min_age NUMBER,
    p_max_age NUMBER
) IS
```

```

-- values for input cursor

v_total_received NUMBER;

v_total_invalid NUMBER;

v_total_valid NUMBER;

v_total_females NUMBER;

v_total_males NUMBER;

v_max_age_donated NUMBER;


--values for counting

v_invalid_male_count NUMBER;

v_invalid_female_count NUMBER;

v_valid_male_count NUMBER;

v_valid_female_count NUMBER;


-- values for printing

v_donation_id donation.donation_id%TYPE;

v_donor_id donor.donor_id%TYPE;


-- cursors for more then 5 lines:

-- cursor for invalid and males

CURSOR v_invalid_male_cursor IS

    SELECT d.donation_id, dr.donor_id

    FROM donation d

    JOIN donor dr ON d.donor_id = dr.donor_id

    WHERE dr.type = p_blood_type

    AND dr.sign = p_blood_sign

    AND TRUNC (MONTHS_BETWEEN (SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

```



```

AND d.valid = 'N'

AND dr.gender = 'M';

-- cursor for invalid and females

CURSOR v_invalid_female_cursor IS

    SELECT d.donation_id, dr.donor_id

    FROM donation d

    JOIN donor dr ON d.donor_id = dr.donor_id

    WHERE dr.type = p_blood_type

    AND dr.sign = p_blood_sign

    AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

    AND d.valid = 'N'

    AND dr.gender = 'F';

-- cursor for valid and males

CURSOR v_valid_male_cursor IS

    SELECT d.donation_id, dr.donor_id

    FROM donation d

    JOIN donor dr ON d.donor_id = dr.donor_id

    WHERE dr.type = p_blood_type

    AND dr.sign = p_blood_sign

    AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

    AND d.valid = 'Y'

    AND dr.gender = 'M';

-- cursor for valid and females

CURSOR v_valid_female_cursor IS

```

```

SELECT d.donation_id, dr.donor_id

FROM donation d

JOIN donor dr ON d.donor_id = dr.donor_id

WHERE dr.type = p_blood_type

AND dr.sign = p_blood_sign

AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

AND d.valid = 'Y'

AND dr.gender = 'F';

BEGIN

-- Fetching data from cursor

FETCH v_donors_cursor INTO

    v_total_received,

    v_total_invalid,

    v_total_valid,

    v_total_females,

    v_total_males,

    v_max_age_donated;

IF v_total_valid < v_total_invalid THEN

    IF v_total_females < v_total_males THEN

        -- when there are more invalid and males

        -- Count invalid donations by males

        SELECT COUNT(*) INTO v_invalid_male_count

        FROM donation d

        JOIN donor dr ON d.donor_id = dr.donor_id

```

```

WHERE dr.type = p_blood_type

AND dr.sign = p_blood_sign

AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

AND d.valid = 'N'

AND dr.gender = 'M';

-- more then 5 invalid donations by males

IF v_invalid_male_count > 5 THEN

OPEN v_invalid_male_cursor;

FOR i IN 1..5 LOOP -- for loop to return the blood (update the
donation)

    FETCH v_invalid_male_cursor INTO v_donation_id,
v_donor_id;

    UPDATE donation

    SET hospital_id = NULL

    WHERE donation_id = v_donation_id;

    DBMS_OUTPUT.PUT_LINE('Invalid Male Donation returned: ID -
' || v_donor_id );

END LOOP;

CLOSE v_invalid_male_cursor;

-- less then 5 and more then 0 invalid donations by males

ELSIF v_invalid_male_count > 0 THEN

OPEN v_invalid_male_cursor;

LOOP

    FETCH v_invalid_male_cursor INTO v_donation_id,
v_donor_id;

    EXIT WHEN v_invalid_male_cursor%NOTFOUND;

    UPDATE donation

    SET hospital_id = NULL

```

```

        WHERE donation_id = v_donation_id;

        DBMS_OUTPUT.PUT_LINE('Invalid Male Donation returned: ID -
' || v_donor_id );

    END LOOP;

    CLOSE v_invalid_male_cursor;

    -- zero invalid donations by males

    ELSE DBMS_OUTPUT.PUT_LINE('No invalid male donation was
returned');

    END IF;

ELSE

    -- when there are more invalid and females

    -- Count invalid donations by females

    SELECT COUNT(*) INTO v_invalid_female_count

    FROM donation d

    JOIN donor dr ON d.donor_id = dr.donor_id

    WHERE dr.type = p_blood_type

    AND dr.sign = p_blood_sign

    AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

    AND d.valid = 'N'

    AND dr.gender = 'F';

    -- more than 5 invalid donations by females

    IF v_invalid_female_count > 5 THEN

    OPEN v_invalid_female_cursor;

    FOR i IN 1..5 LOOP -- for loop to return the blood (update the
donation)

        FETCH v_invalid_female_cursor INTO v_donation_id,
v_donor_id;

```

```

        UPDATE donation

        SET hospital_id = NULL

        WHERE donation_id = v_donation_id;

        DBMS_OUTPUT.PUT_LINE('Invalid Female Donation returned: ID
- ' || v_donor_id );

    END LOOP;

    CLOSE v_invalid_female_cursor;

    -- less than 5 and more than 0 invalid donations by females

    ELSIF v_invalid_female_count > 0 THEN

    OPEN v_invalid_female_cursor;

    LOOP

        FETCH v_invalid_female_cursor INTO v_donation_id,
v_donor_id;

        EXIT WHEN v_invalid_female_cursor%NOTFOUND;

        UPDATE donation

        SET hospital_id = NULL

        WHERE donation_id = v_donation_id;

        DBMS_OUTPUT.PUT_LINE('Invalid Female Donation returned: ID
- ' || v_donor_id );

    END LOOP;

    CLOSE v_invalid_female_cursor;

    -- zero invalid donations by females

    ELSE DBMS_OUTPUT.PUT_LINE('No invalid female donation was
returned');

    END IF;

END IF;

ELSE

    IF v_total_females < v_total_males THEN

        -- when there are more valid and males

```

```

-- Count valid donations by males

SELECT COUNT(*) INTO v_valid_male_count

FROM donation d

JOIN donor dr ON d.donor_id = dr.donor_id

WHERE dr.type = p_blood_type

AND dr.sign = p_blood_sign

AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

AND d.valid = 'Y'

AND dr.gender = 'M';

-- more then 5 valid donations by males

IF v_valid_male_count > 5 THEN

OPEN v_valid_male_cursor;

FOR i IN 1..5 LOOP -- for loop to return the blood (update the
donation)

    FETCH v_valid_male_cursor INTO v_donation_id, v_donor_id;

    UPDATE donation

    SET hospital_id = NULL

    WHERE donation_id = v_donation_id;

    DBMS_OUTPUT.PUT_LINE('Valid Male Donation returned: ID - '
|| v_donor_id );

END LOOP;

CLOSE v_valid_male_cursor;

-- less then 5 and more then 0 valid donations by males

ELSIF v_valid_male_count > 0 THEN

OPEN v_valid_male_cursor;

LOOP

    FETCH v_valid_male_cursor INTO v_donation_id, v_donor_id;

```

```

        EXIT WHEN v_valid_male_cursor%NOTFOUND;

        UPDATE donation

        SET hospital_id = NULL

        WHERE donation_id = v_donation_id;

        DBMS_OUTPUT.PUT_LINE('Valid Male Donation returned: ID - '
|| v_donor_id );

    END LOOP;

    CLOSE v_valid_male_cursor;

    -- zero valid donations by males

    ELSE DBMS_OUTPUT.PUT_LINE('No valid male donation was
returned');

    END IF;

ELSE

    -- when there are more valid and females

    -- Count valid donations by females

    SELECT COUNT(*) INTO v_valid_female_count

    FROM donation d

    JOIN donor dr ON d.donor_id = dr.donor_id

    WHERE dr.type = p_blood_type

    AND dr.sign = p_blood_sign

    AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age

    AND d.valid = 'Y'

    AND dr.gender = 'F';

    -- more then 5 valid donations by females

    IF v_valid_female_count > 5 THEN

    OPEN v_valid_female_cursor;

```

```

FOR i IN 1..5 LOOP -- for loop to return the blood (update the
donation)

    FETCH v_valid_female_cursor INTO v_donation_id,
v_donor_id;

    UPDATE donation

    SET hospital_id = NULL

    WHERE donation_id = v_donation_id;

    DBMS_OUTPUT.PUT_LINE('Valid Female Donation returned: ID -
' || v_donor_id );

    END LOOP;

    CLOSE v_valid_female_cursor;

    -- less then 5 and more then 0 valid donations by females

    ELSIF v_valid_female_count > 0 THEN

    OPEN v_valid_female_cursor;

    LOOP

        FETCH v_valid_female_cursor INTO v_donation_id,
v_donor_id;

        EXIT WHEN v_valid_female_cursor%NOTFOUND;

        UPDATE donation

        SET hospital_id = NULL

        WHERE donation_id = v_donation_id;

        DBMS_OUTPUT.PUT_LINE('Valid Female Donation returned: ID -
' || v_donor_id );

        END LOOP;

        CLOSE v_valid_female_cursor;

        -- zero valid donations by females

    ELSE DBMS_OUTPUT.PUT_LINE('No valid female donation was
returned');

    END IF;

END IF;

END IF;

```



```

commit; -- save changes

-- Closing the cursors after use

CLOSE v_donors_cursor;

;END

```

הפונקציה get_donation_statistics -

מחזירה סטטיסטיקות על תרומות הדם התקפות לפי סוג דם, סימן דם, וטווח גילאים שמוגדרים כקלט. היא מחזירה סוג SYS_REFCURSOR, שזה מצביע למערך של התוצאות, שמספק גישה לתוצאות בצורה סדורה.

```

CREATE OR REPLACE FUNCTION get_donation_statistics (

    p_blood_type VARCHAR2,

    p_blood_sign VARCHAR2,

    p_min_age NUMBER,

    p_max_age NUMBER

) RETURN SYS_REFCURSOR IS

    v_stats_cursor SYS_REFCURSOR;

BEGIN

    OPEN v_stats_cursor FOR

        SELECT

            COUNT(*) AS total_received,

            SUM(CASE WHEN d.valid = 'N' THEN 1 ELSE 0 END) AS total_invalid,

            COUNT(CASE WHEN d.valid = 'Y' THEN 1 END) AS total_valid,

            COUNT(CASE WHEN dr.gender = 'F' AND d.valid = 'Y' THEN 1 END) AS
total_females,

            COUNT(CASE WHEN dr.gender = 'M' AND d.valid = 'Y' THEN 1 END) AS
total_males,

            MAX(TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12)) AS
max_age_donated

        FROM donation d

```

```
JOIN donor dr ON d.donor_id = dr.donor_id

WHERE dr.type = p_blood_type

AND dr.sign = p_blood_sign

AND TRUNC(MONTHS_BETWEEN(SYSDATE, dr.birth_date) / 12) BETWEEN
p_min_age AND p_max_age;

RETURN v_stats_cursor;

;END
```

2. לאור העבודה הקשה והאינטנסיבית של הפרמדיקים לאורך כל תקופת המלחמה ולנוכח הקושי בגיוס פרמדיקים נוספים למד"א, החליטו במשרד הבריאות להוקיר תודה לארגון מד"א ועובדיו. במשרד החליט להעניק תעודות הוקרה לפרמדיקים מצטיינים שהשתתפו ביותר מ-15 התרמות וכן לפרמדיקים שתרמו דם יותר מ-13 פעמים.

כמו כן, החליט משרד הבריאות בשיתוף משרד האוצר להגדיל את התקציב השנתי של מד"א בכ-20 מיליון ש"ח בשנה. במד"א הוחלט להקדיש חלק גדול מהתקציב להעלאת המשכורות של פרמדיקים עם וותק של יותר מ-45 שנה וכן להעלאת המשכורות של כל הפרמדיקים שהתנדבו בשבעה לאוקטובר.

בתוכנית הבאה אנחנו מעדכנים את המשכורת של הפרמדיקים הנבחרים ע"י זימון הפרוצדורה Add_Grant, כמו כן, נדפיס את שמות כל הפרמדיקים שקיבלו העלאה ואת סכום ההעלאה שקיבלו.

בנוסף, התוכנית מוציאה את שמות כל הפרמדיקים המצטיינים ע"י הפונקציה Is_Paramedic_Outstanding ומדפיסה אותם ואת סיבות ההצטיינות כך שנוכל להעביר את הנתונים למשרד הבריאות.

(ניתן לראות את התוכנית בקובץ MainProgram2.sql)

הקוד של התוכנית -

```
DECLARE

v_paramedic_id VARCHAR2(9);

v_first_name VARCHAR2(10);

v_last_name VARCHAR2(10);

v_reason VARCHAR2(200);

v_cursor SYS_REFCURSOR;

v_paramedic_threshold NUMBER := 15; -- Set the paramedic threshold value

v_donor_threshold NUMBER := 13; -- Set the donor threshold value

BEGIN

-- Call the Add_Grant procedure first

DBMS_OUTPUT.PUT_LINE('Paramedics who received a grant: ');

DBMS_OUTPUT.PUT_LINE(' ');
```

```

Add_Grant;

DBMS_OUTPUT.PUT_LINE(' ');

DBMS_OUTPUT.PUT_LINE('Outstanding paramedics: ');

DBMS_OUTPUT.PUT_LINE(' ');

-- Call the Is_Paramedic_Outstanding function to get the cursor

v_cursor := Is_Paramedic_Outstanding(v_paramedic_threshold,
v_donor_threshold);

LOOP

    FETCH v_cursor INTO v_paramedic_id, v_reason;

    EXIT WHEN v_cursor%NOTFOUND;

BEGIN

    SELECT f_name, l_name INTO v_first_name, v_last_name

    FROM PERSON

    WHERE pid = v_paramedic_id;

    DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name || ': ' ||
v_reason);

    EXCEPTION

        WHEN NO_DATA_FOUND THEN

            DBMS_OUTPUT.PUT_LINE('No data found for paramedic ID: ' ||
v_paramedic_id);

        WHEN OTHERS THEN

            DBMS_OUTPUT.PUT_LINE('Error fetching paramedic name: ' || SQLERRM);

    END;

END LOOP;

CLOSE v_cursor;

```

```

EXCEPTION

WHEN OTHERS THEN

    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

    IF v_cursor%ISOPEN THEN

        CLOSE v_cursor;

    END IF;

END;
```

תוצאת הרצה -

Clear	Buffer size 10000	<input checked="" type="checkbox"/> Enabled
-------	-------------------	---

```

Paramedics who received a grant:

Garry Duschel (570331042) - Grant of: 700
Jamie Shandling (119591809) - Grant of: 700
Joanna McClinton (976460507) - Grant of: 700
Anna Lewin (182761224) - Grant of: 1200
Murray Garza (160587152) - Grant of: 700
Gloria Adler (740704902) - Grant of: 700
Edie Stuart (292509980) - Grant of: 700
Ashton Cash (002554127) - Grant of: 700
Christian Jones (360157179) - Grant of: 700
Adina Rodriguez (267941315) - Grant of: 700
Praga Warren (320257722) - Grant of: 700
Geoff Warden (539935524) - Grant of: 700
Treat Perrineau (311853595) - Grant of: 700
Rose Driver (617833949) - Grant of: 700
Karon Merchant (573459851) - Grant of: 700
Jessica Haslam (623731843) - Grant of: 700
Katrin Leguizamo (004437338) - Grant of: 700
Steve Holiday (998752021) - Grant of: 700
Etta Idle (276143046) - Grant of: 700
Albertina Curfman (264346014) - Grant of: 700
Freda Tsettos (610422260) - Grant of: 700
Edwin Stevens (566372714) - Grant of: 700
Lindsay O'Keefe (268303729) - Grant of: 700
```


Outstanding paramedics:

Manu Valentin: Participated in more than 15 donations
 Angelina Camp: Donated more than 13 times
 Nigel Parsons: Participated in more than 15 donations
 Clea Hopper: Donated more than 13 times
 Tamala Frost: Participated in more than 15 donations
 Jann O'Sullivan: Participated in more than 15 donations
 Geggy Paymer: Participated in more than 15 donations
 Nickel Iglesias: Participated in more than 15 donations
 Campbell Shearer: Participated in more than 15 donations
 Christmas McKennitt: Participated in more than 15 donations
 Jonny Lee Shawn: Participated in more than 15 donations
 Maceo Steiger: Participated in more than 15 donations
 Vickie McIntosh: Participated in more than 15 donations
 Jet Skaggs: Participated in more than 15 donations
 Denzel Flack: Donated more than 13 times
 Nikki Sobieski: Donated more than 13 times
 Manu Valentin: Participated in more than 15 donations
 Angelina Camp: Donated more than 13 times
 Nigel Parsons: Participated in more than 15 donations
 Clea Hopper: Donated more than 13 times
 Tamala Frost: Participated in more than 15 donations
 Jann O'Sullivan: Participated in more than 15 donations
 Geggy Paymer: Participated in more than 15 donations

בדיקת שינויים -

לדוגמה, Garry Duschel (ת"ז 570331042) קיבל העלאה של 700 ש"ח כיוון שיש לו ותק של 45 שנה.

לפני הרצת התוכנית -

SQL Output Statistics				
<pre>select * from paramedic where paramedic_id = 570331042</pre>				
				
	LICENSE_NUM	SENIORITY	PARAMEDIC_ID	SALARY
1	87951	50	570331042	36905

לאחר הרצת התוכנית -

SQL Output Statistics				
<pre>select * from paramedic where paramedic_id = 570331042</pre>				
	LICENSE_NUM	SENIORITY	PARAMEDIC_ID	SALARY
1	87951	50	570331042	37605

ניתן לראות שהשכר של הפרמדיק גרי (עם יותר מ-45 שנות וותק) אכן עלה ב-700 ש"ח.

הפרוצדורה Add_Grant -

פרוצדורה להוספת מענק (העלאה בשכר) לפרמדיקים שהשתתפו בהתרמות דם בשביעי לאוקטובר וכן לפרמדיקים עם וותק של יותר מ-45 שנה. ההעלאה לפרמדיקים שהשתתפו בשביעי באוקטובר היא 500 ש"ח ואילו לפרמדיקים ותיקים היא 700 ש"ח. הפרוצדורה עובדת לפי השלבים הבאים:

1. פתיחת קורסור על כל הפרמדיקים בטבלת PARAMEDIC.
2. לולאה על כל רשומה מהקורסור:
 - איפוס הסכום של המענק עבור הפרמדיק.
 - ניסיון לשלוף את השם הפרטי ושם המשפחה של הפרמדיק מתוך טבלת PERSON. אם אין נתונים, מגדירים את השמות כ-'Unknown'.
 - בדיקה אם הפרמדיק השתתף בתרומה בתאריך 07-10-2024 והוספת מענק של 500 ש"ח במידה וכן.
 - בדיקה אם לפרמדיק יש וותק של יותר מ-45 שנים והוספת מענק של 700 ש"ח במידה וכן.
 - אם ניתן מענק כלשהו, עדכון השכר של הפרמדיק בטבלת PARAMEDIC והצגת הודעה דרך DBMS_OUTPUT.
3. סגירת הקורסור.
4. ביצוע COMMIT לשינויים שנעשו.
5. טיפול בחריגות במקרה של שגיאה וביצוע ROLLBACK.

```

CREATE OR REPLACE PROCEDURE Add_Grant
IS
    v_paramedic_rec PARAMEDIC%ROWTYPE;
    v_donation_count NUMBER;
    v_first_name VARCHAR2(50);
    v_last_name VARCHAR2(50);
    CURSOR c_paramedics IS
        SELECT * FROM PARAMEDIC;
BEGIN
    OPEN c_paramedics;

    LOOP

        FETCH c_paramedics INTO v_paramedic_rec;

        EXIT WHEN c_paramedics%NOTFOUND;

        -- Initialize grant amount for reporting
        v_paramedic_rec.salary := 0;

        -- Get paramedic name
        BEGIN
            SELECT f_name, l_name INTO v_first_name, v_last_name
            FROM PERSON
            WHERE pid = v_paramedic_rec.paramedic_id;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                v_first_name := 'Unknown';
                v_last_name := 'Unknown';
        END;
    END LOOP;
END;

```



```

END;

-- Check for paramedics who made donations on 7/10/2024

SELECT COUNT(*) INTO v_donation_count
FROM DONATION
WHERE paramedic_id = v_paramedic_rec.paramedic_id
      AND donation_date = TO_DATE('07-10-2024', 'dd-mm-yyyy');

IF v_donation_count > 0 THEN
    v_paramedic_rec.salary := v_paramedic_rec.salary + 500;
END IF;

-- Check for paramedics with more than 45 years seniority
IF v_paramedic_rec.seniority > 45 THEN
    v_paramedic_rec.salary := v_paramedic_rec.salary + 700;
END IF;

-- Update the salary if any grant is given
IF v_paramedic_rec.salary > 0 THEN
    UPDATE PARAMEDIC
    SET salary = salary + v_paramedic_rec.salary
    WHERE paramedic_id = v_paramedic_rec.paramedic_id;

    DBMS_OUTPUT.PUT_LINE(v_first_name || ' ' || v_last_name || ' (' ||
v_paramedic_rec.paramedic_id || ') - Grant of: ' || v_paramedic_rec.salary);

END IF;

END LOOP;

CLOSE c_paramedics;

```

```
DBMS_OUTPUT.PUT_LINE('Grants added to paramedics based on donations and
seniority.');
```

```
COMMIT;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
ROLLBACK;
```

```
RAISE;
```

```
;END
```

הפונקציה Is_Paramedic_Outstanding -

מחזירה את כל הפרמדיקים המצטיינים ואת סיבות ההצטיינות שלהם. פרמדיק מצטיין הוא פרמדיק שהשתתף ביותר ממספר מסוים של התרומות ושתרם דם בעצמו יותר ממספר מסוים של פעמים (מספר הפעמים מתקבל ע"י פרמטר).

הפונקציה עובדת לפי השלבים הבאים:

1. פתיחת קורסור על כל הפרמדיקים בטבלת PARAMEDIC.
2. לולאה על כל רשומה מהקורסור:
 - חישוב מספר התרומות שבהן הפרמדיק השתתף כפרמדיק.
 - חישוב מספר התרומות שבהן הפרמדיק השתתף כתורם.
 - בדיקת תנאים והגדרת סיבה מתאימה אם הוא עומד באחד התנאים.
 - הכנסת הרשומה עם הסיבה לטבלה זמנית temp_paramedic_reasons.
3. פתיחת קורסור על הטבלה הזמנית temp_paramedic_reasons והחזרת ה-REF CURSOR עם התוצאות.

```
CREATE OR REPLACE FUNCTION Is_Paramedic_Outstanding(
    p_paramedic_threshold NUMBER,
    p_donor_threshold NUMBER
)
RETURN SYS_REFCURSOR
IS
    v_cursor SYS_REFCURSOR;
```

```

v_paramedic_rec PARAMEDIC%ROWTYPE;

v_donation_count_paramedic NUMBER;

v_donation_count_donor NUMBER;

v_reason VARCHAR2(200);

CURSOR c_paramedics IS

    SELECT * FROM PARAMEDIC;

BEGIN

    OPEN c_paramedics;

    LOOP

        FETCH c_paramedics INTO v_paramedic_rec;

        EXIT WHEN c_paramedics%NOTFOUND;

        -- Get the number of donations made by the paramedic as paramedic_id

        SELECT COUNT(*) INTO v_donation_count_paramedic

        FROM DONATION

        WHERE paramedic_id = v_paramedic_rec.paramedic_id;

        -- Get the number of donations made by the paramedic as donor_id

        SELECT COUNT(*) INTO v_donation_count_donor

        FROM DONATION

        WHERE donor_id = v_paramedic_rec.paramedic_id;

        -- Check conditions and set reasons

        IF v_donation_count_paramedic > p_paramedic_threshold THEN

            v_reason := 'Participated in more than ' || p_paramedic_threshold || '
donations';

        ELSIF v_donation_count_donor > p_donor_threshold THEN

            v_reason := 'Donated more than ' || p_donor_threshold || ' times';

```

```

ELSE

v_reason := NULL;

END IF;

IF v_reason IS NOT NULL THEN

INSERT INTO temp_paramedic_reasons (paramedic_id, reason)

VALUES (v_paramedic_rec.paramedic_id, v_reason);

END IF;

END LOOP;

CLOSE c_paramedics;

OPEN v_cursor FOR

SELECT paramedic_id, reason FROM temp_paramedic_reasons;

RETURN v_cursor;

EXCEPTION

WHEN NO_DATA_FOUND THEN

RETURN NULL;

WHEN OTHERS THEN

RAISE;

;END

```

בשביל הפונקציה יצרנו טבלה זמנית בשם temp_paramedic_reasons:

```

CREATE GLOBAL TEMPORARY TABLE temp_paramedic_reasons (

paramedic_id VARCHAR2(9) NOT NULL,

reason VARCHAR2(200) NOT NULL

) ON COMMIT DELETE ROWS;

```

