

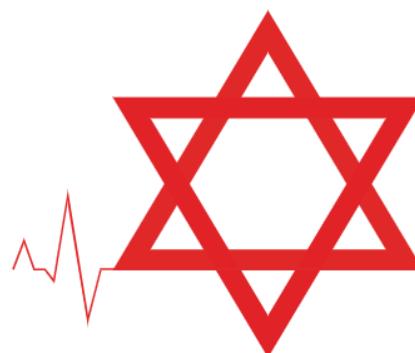
מִינֵי פְּרוּיִקְט

בסי"ס נתוניים

תשפ"ד-2024

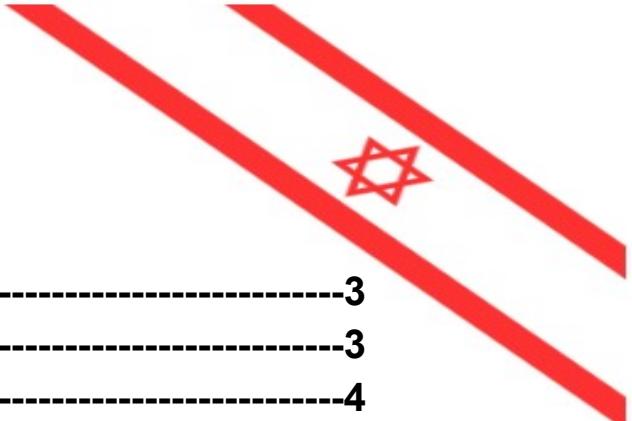
מג'ישות:

הדר סרое hadarsurus21@gmail.com - 213383110
לאה חיים leahaim36@gmail.com - 214085888



שותפים בהצלת חיים

תוכן עניינים:

	
מבוא ותיאור הארגון	3
פירוט הישויות	3
פירוט הקשרים	4
TERSHIM	4
DSD	4
פקודות create Table	5
פקודות desc	6
הכנסת נתונים	8
גיבוי נתונים	10
שחזור נתונים	11
select	12
delete	16
update	18
אלצים	20
פונקציה 1	21
פראצורה 1	22
תכנית ראשית 1	23
הרצה של תוכנית 1	24
פונקציה 2	25
פראצורה 2	26
תכנית ראשית 2	27
הרצה של תוכנית 2	28
שלב 4	31

שם הארגון: מד"א- מגן דוד אדום.

תיאור הארגון:

מגן דוד אדום לישראל הוא ארגון חירום ציבורי המספק סיוע רפואי והצלה במצב חירום. במסגרת מד"א, פעילים אלפי עובדים, כולל חובשים ונוהגים, המוחולקים לצוותים. תפקידם העיקרי של הצוותים הוא לטפל בקויות חירום שמגיעות ממטופלים, ולספק להם את הטיפול הרפואי הנדרש. חלק מטופיקדים, הם מצטיידים בצד הנדרש, על מנת לספק טיפול מיידי ומקצועי לפצועים ולחולמים במצב חירום. בנוסף לשיער רפואי במקומות האירוע, מגן דוד אדום גם מספק שירות פינוי של חולמים לבתי חולים.

ישויות:

1. ציוד -Equipment

תכונות:

- מזהה ציוד -id •
- שם - e_name •
- סטטוס- status_equipment •
- תאריך השאלה- date_borrow •

2. עובד -Employee

תכונות:

- מזהה עובד -id •
- שם-emp_name •
- משכורת- salary •

3. צוות - Team

תכונות:

- מזהה צוות- id_team •
- כמות עובדים בצוות-amount •
- שם הצוות- name_of_team •

4. קריאה - Call

תכונות:

- מזהה קריאה- id_call •
- דחיפות- urgency •
- פרטי קריאה- details •
- תאריך הקריאה- date_call •

5. בית חולים - Hospital

תכונות:

- מזהה בית חולים- id_hospital •
- עיר- city •
- טלפון- phone •

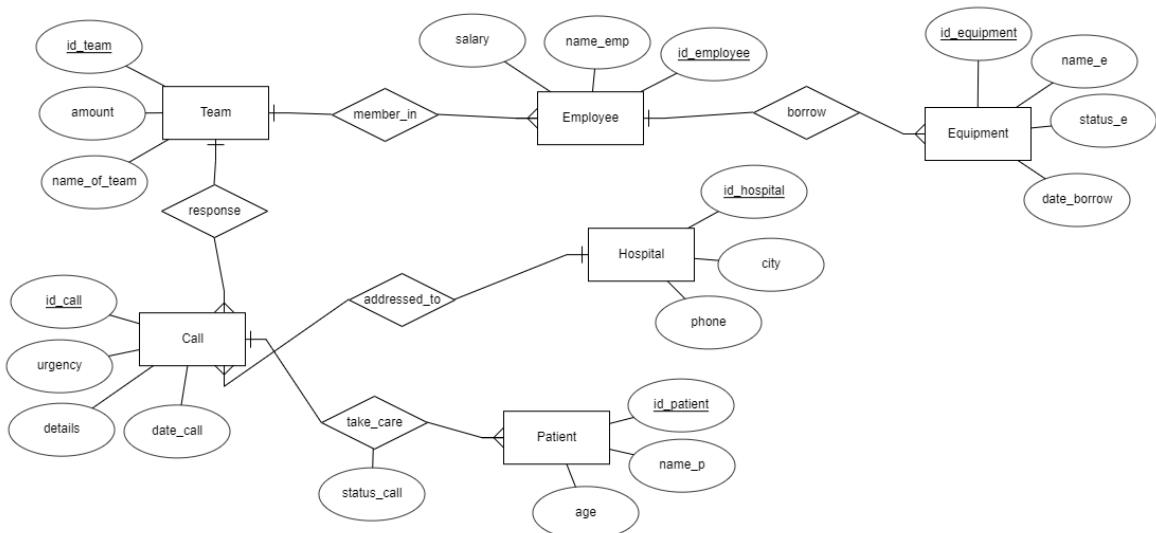
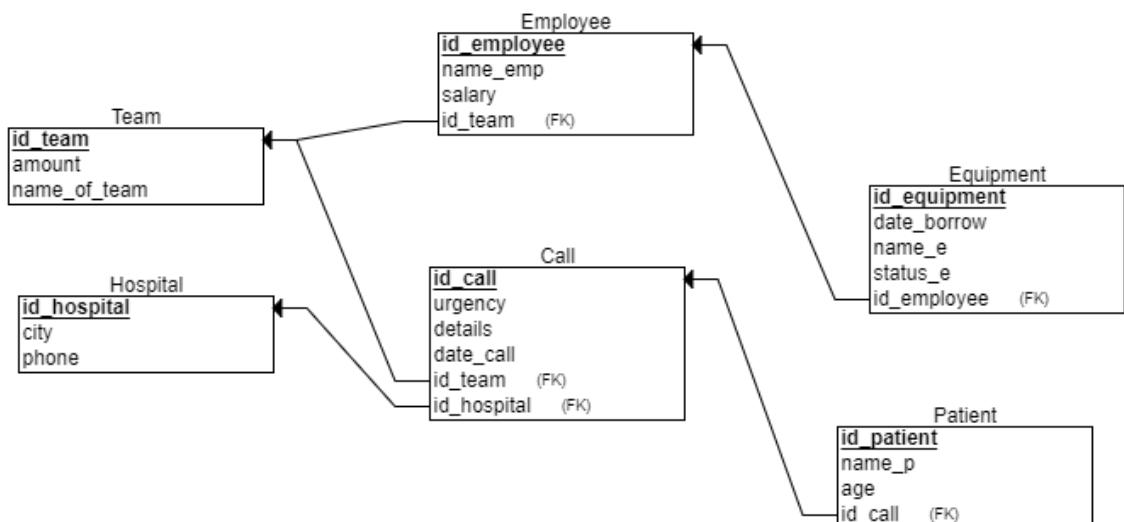
6. מטופל - Patient

תכונות:

- מזהה חוליה - id_patient •
- שם- k_name •
- גיל- age •

קשרים:

1. משכיר -borrow: קשר יחיד לרבים , כאשר כל עובד יכול להשכיר כמה ציודים , ולכל ציוד יש עובד אחד ששוכר אותו.
2. חבר ב-in member: קשר יחיד לרבים, כאשר בכל צוות יכולים להיות הרבה עובדים, וכל עובד יכול להיות בצוות אחד.
3. מגיב-response: קשר יחיד לרבים , כאשר כל צוות יכול לטפל בכמה קרייאות וכל קריאה מטופלת ע"ז צוות אחד.
4. מטפל-care take: קשר יחיד לרבים, כאשר קריאה יכולה לטפל במספר חולמים וחולה מטופל ע"ז קריאה אחת. לקשר יש תכונה: סטטוס-call status_call - מה הסטטוס של החולים כשהתבצעה קריאה
5. מפנהה ל-to address: קשר יחיד לרבים, כאשר כמה קרייאות יכולות לפנות אותה בית חולים , וקריאה יכולה להתפנות לבית החולים אחד.

תרשים : ERD**תרשים : DSD**

פקיות CREATE

```
CREATE TABLE Hospital
(
    id_hospital NUMERIC(3) NOT NULL,
    city VARCHAR(15),
    phone NUMERIC(10),
    PRIMARY KEY (id_hospital)
);
```

```
CREATE TABLE Call
(
    id_call NUMERIC(3) NOT NULL,
    urgency NUMERIC(1),
    details VARCHAR(15),
    date_call DATE,
    id_team NUMERIC(3),
    id_hospital NUMERIC(3),
    PRIMARY KEY (id_call),
    FOREIGN KEY (id_team) REFERENCES Team(id_team),
    FOREIGN KEY (id_hospital) REFERENCES Hospital(id_hospital)
);
```

```
CREATE TABLE Employee
(
    id_employee NUMERIC(3) NOT NULL,
    name_emp VARCHAR(15),
    salary NUMERIC(6),
    id_team NUMERIC(3),
    PRIMARY KEY (id_employee),
    FOREIGN KEY (id_team) REFERENCES Team(id_team)
);
```

```
CREATE TABLE Equipment
(
    date_borrow DATE NOT NULL,
    name_e VARCHAR(15),
    id_equipment NUMERIC(3),
    status_e NUMERIC(5),
    id_employee NUMERIC(3),
    PRIMARY KEY (id_equipment),
    FOREIGN KEY (id_employee) REFERENCES Employee(id_employee)
);
```

```
CREATE TABLE Team
(
    id_team NUMERIC(3) NOT NULL,
    amount NUMERIC(3),
    name_of_team VARCHAR(15),
    PRIMARY KEY (id_team)
);
```

```

CREATE TABLE Patient
(
    id_patient NUMERIC(3) NOT NULL,
    name_p VARCHAR(15),
    age NUMERIC(3),
    id_call NUMERIC(3),
    PRIMARY KEY (id_patient),
    FOREIGN KEY (id_call) REFERENCES Call(id_call)
);

```

:desc נידוחן

CREATE AS TEAM

SQL> desc TEAM;

Name	Type	Nullable	Default	Comments
ID_TEAM	NUMBER(3)			
AMOUNT	NUMBER(3)	Y		
NAME_OF_TEAM	VARCHAR2(15)	Y		

SQL>

SQL> desc CALL;

Name	Type	Nullable	Default	Comments
ID_CALL	NUMBER(3)			
URGENCY	NUMBER(1)	Y		
DETAILS	VARCHAR2(15)	Y		
DATE_CALL	DATE	Y		
ID_TEAM	NUMBER(3)	Y		
ID_HOSPITAL	NUMBER(3)	Y		

```
SQL> desc EQUIPMENT;
Name          Type           Nullable Default Comments
-----  -----
DATE_BORROW   DATE
NAME_E        VARCHAR2(15)  Y
ID_EQUIPMENT NUMBER(3)
STATUS_E      NUMBER(5)    Y
ID_EMPLOYEE   NUMBER(3)    Y
```

```
SQL> desc EMPLOYEE;
Name          Type           Nullable Default Comments
-----  -----
ID_EMPLOYEE   NUMBER(3)
NAME_EMP      VARCHAR2(15)  Y
SALARY        NUMBER(6)    Y
ID_TEAM       NUMBER(3)    Y
```

```
SQL> desc HOSPITAL;
Name          Type           Nullable Default Comments
-----  -----
ID_HOSPITAL   NUMBER(3)
CITY          VARCHAR2(15)  Y
PHONE         NUMBER(10)   Y
```

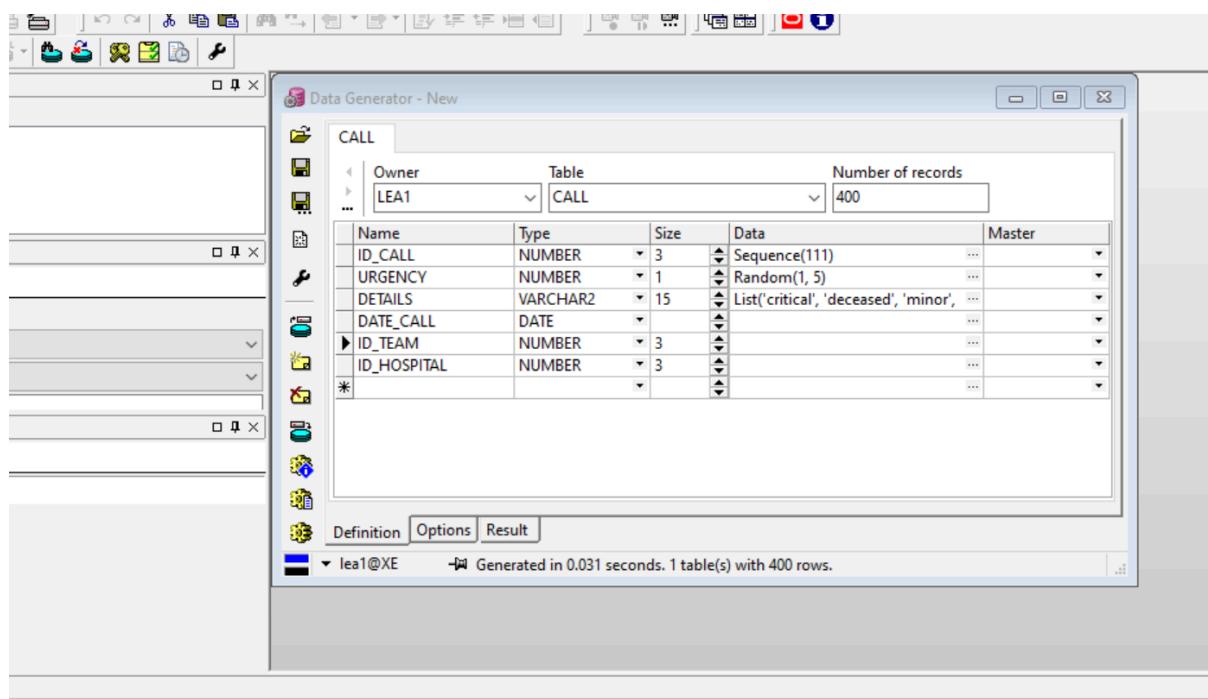
```
SQL> desc PATIENT;
Name          Type           Nullable Default Comments
-----  -----
ID_PATIENT    NUMBER(3)
NAME_P        VARCHAR2(15)  Y
AGE           NUMBER(3)    Y
ID_CALL       NUMBER(3)    Y
```

שלושת השיטות להכנסת נתונים שבחרנו הן:

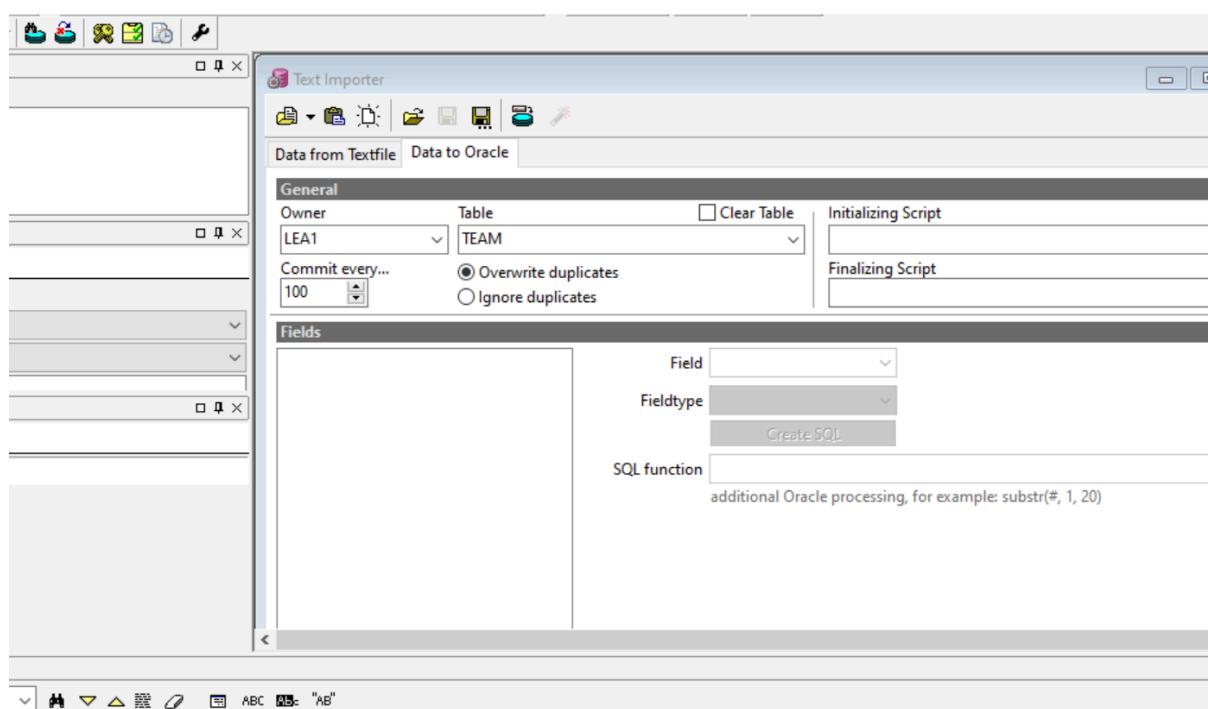
1. שיטת data generator
2. יבוא נתונים מקובצים (קובץ csv/excel)
3. שימוש באתר mockaroo

צילומי מסך:

שיטת 1:



שיטת 2:



שיטה 3:



mockaroo

SCHEMAS DATASETS MOCK APIs SCENARIOS

Looking to generate **fake data** based on your **production data**?

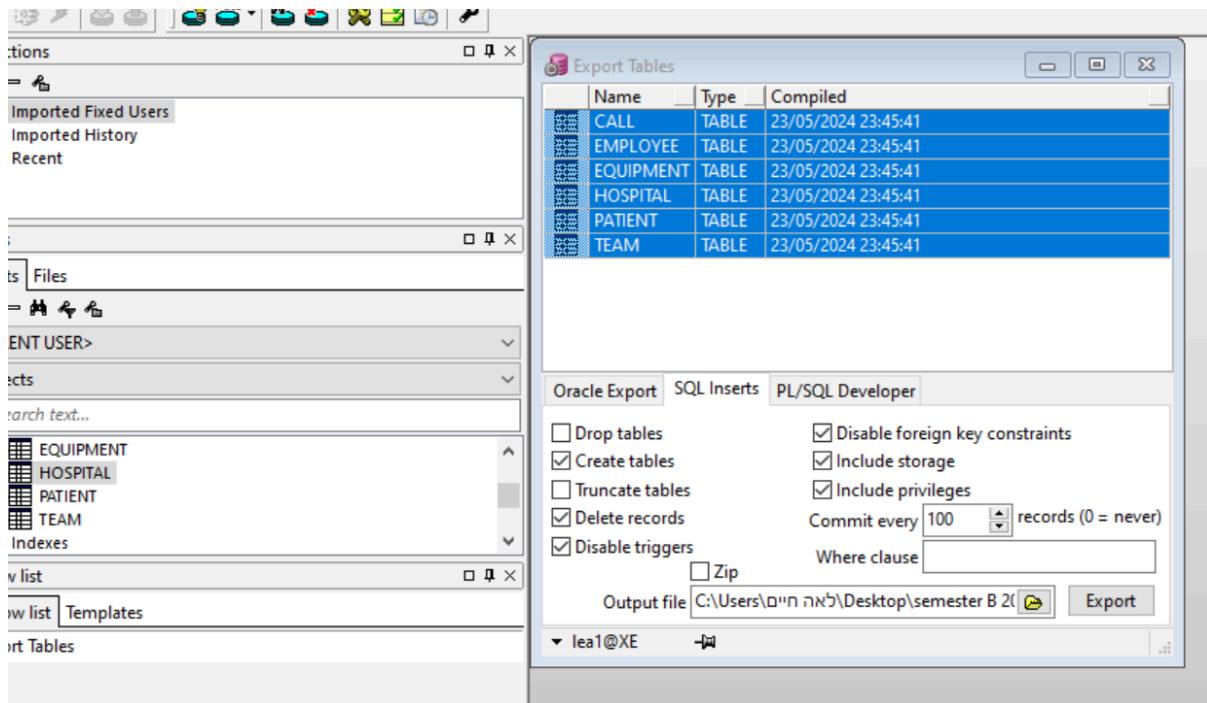
Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of real-world data.

Need more data? Plans start at just \$60/year. Mockaroo is also available as a [docker image](#).

Field Name	Type	Options
id_hospital	Sequence	start at: 1
city	City	blank: 0 %
phone	Phone	format: ####-####-####

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

ביצוע גיבוי נתונים:



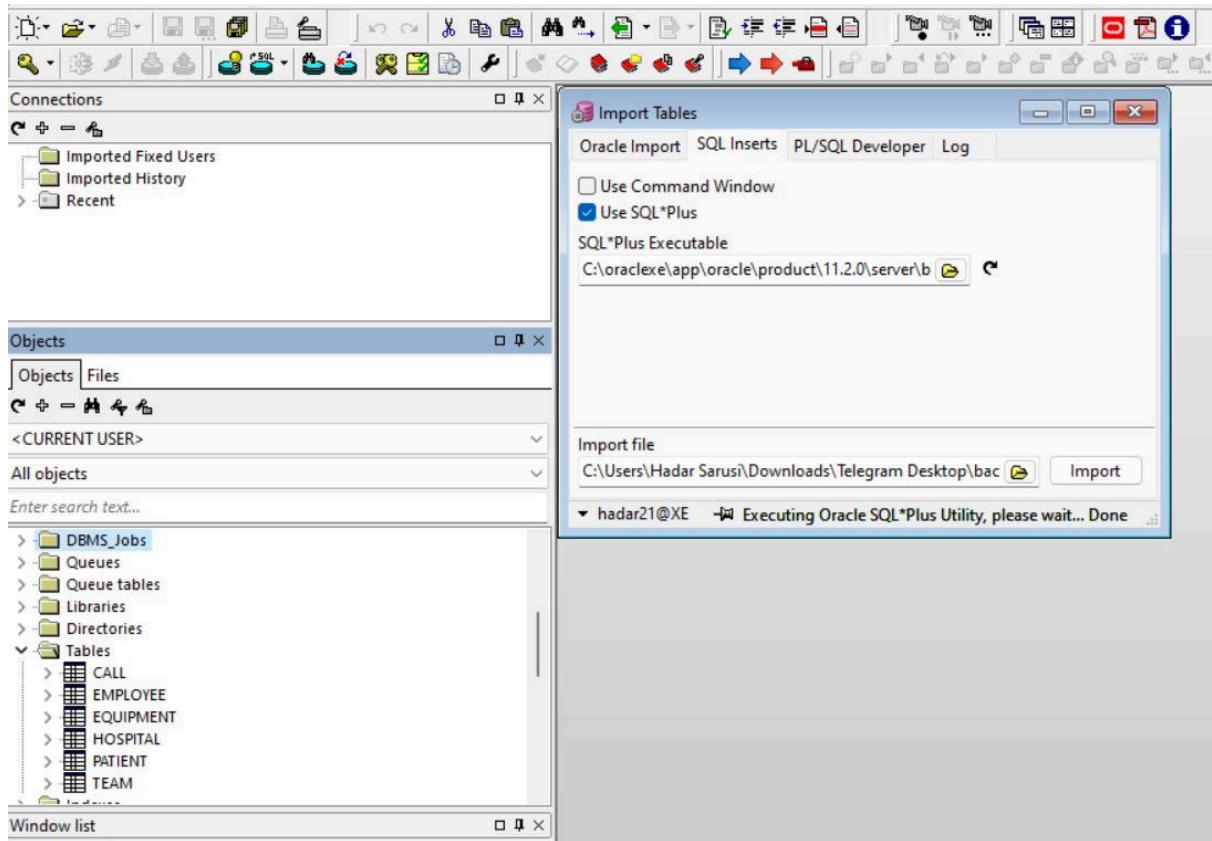
קובץ הגיבוי:

```

prompt PL/SQL Developer import file
prompt Created on יומ שישי 24 מאי 2024 by לאה דיבין
set feedback off
set define off
...prompt Creating HOSPITAL
create table HOSPITAL
)
,id_hospital NUMBER(3) not null
,(15)city      VARCHAR2
(10)phone      NUMBER
;
alter table HOSPITAL
add primary key (ID_HOSPITAL)
; using index

...prompt Creating TEAM
create table TEAM
)
,id_team      NUMBER(3) not null
,(3)amount     NUMBER
(15)name_of_team VARCHAR2
;
alter table TEAM
add primary key (ID_TEAM)
; using index

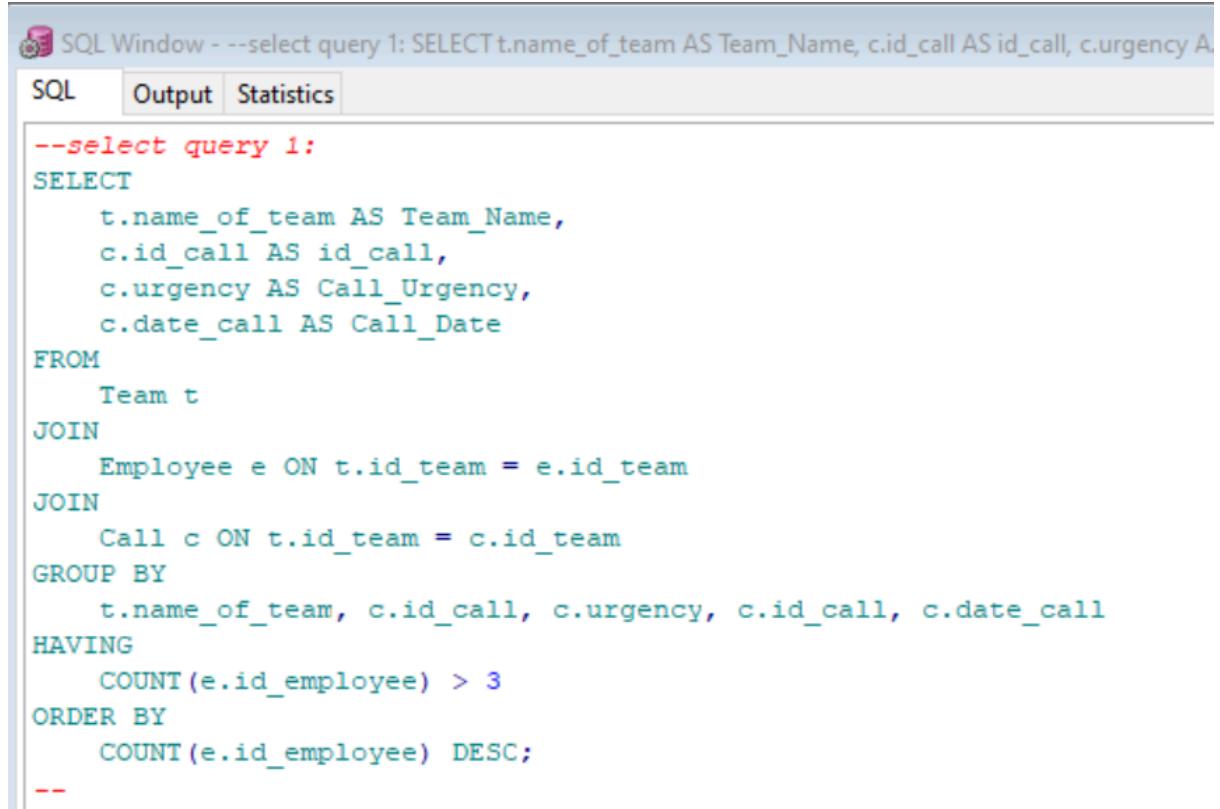
...prompt Creating CALL
create table CALL
)
,id_call      NUMBER(3) not null
;
```

שחזור הנתונים:

שאילות: select

:1

השאילה מבקשת מציגה פרטים של קריאות שהוצאות שטיפל בהן מנה יותר מ-3 אנשים.
יצג למשתמש: שם הצוות , מזחה קריאה , הדחיפות של הקריאה, ותאריך הקריאה.
צילום הריצה:



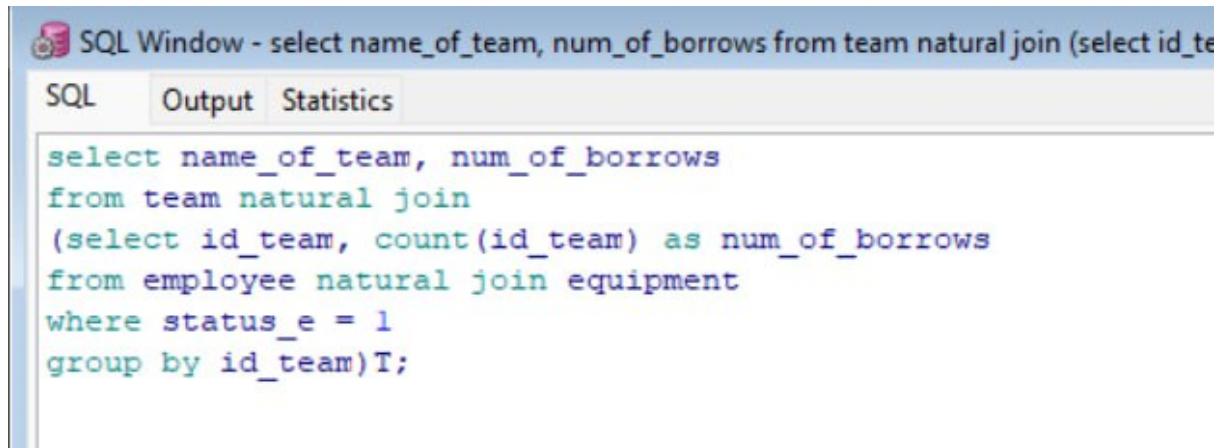
--select query 1:
SELECT
 t.name_of_team AS Team_Name,
 c.id_call AS id_call,
 c.urgency AS Call_Urgency,
 c.date_call AS Call_Date
FROM
 Team t
JOIN
 Employee e ON t.id_team = e.id_team
JOIN
 Call c ON t.id_team = c.id_team
GROUP BY
 t.name_of_team, c.id_call, c.urgency, c.id_call, c.date_call
HAVING
 COUNT(e.id_employee) > 3
ORDER BY
 COUNT(e.id_employee) DESC;
--

צילום פלט:

	TEAM_NAME	ID_CALL	CALL_URGENCY	CALL_DATE
► 1	Skajo	310	2	13/09/1994 ▾
2	Blogtags	363	1	08/10/2022 ▾
3	Skajo	248	5	07/09/1973 ▾
4	Photojam	430	4	14/12/1995 ▾
5	Kwimbee	447	2	05/11/2008 ▾
6	Blogpad	349	1	22/07/1993 ▾
7	Realfire	196	2	16/02/1994 ▾

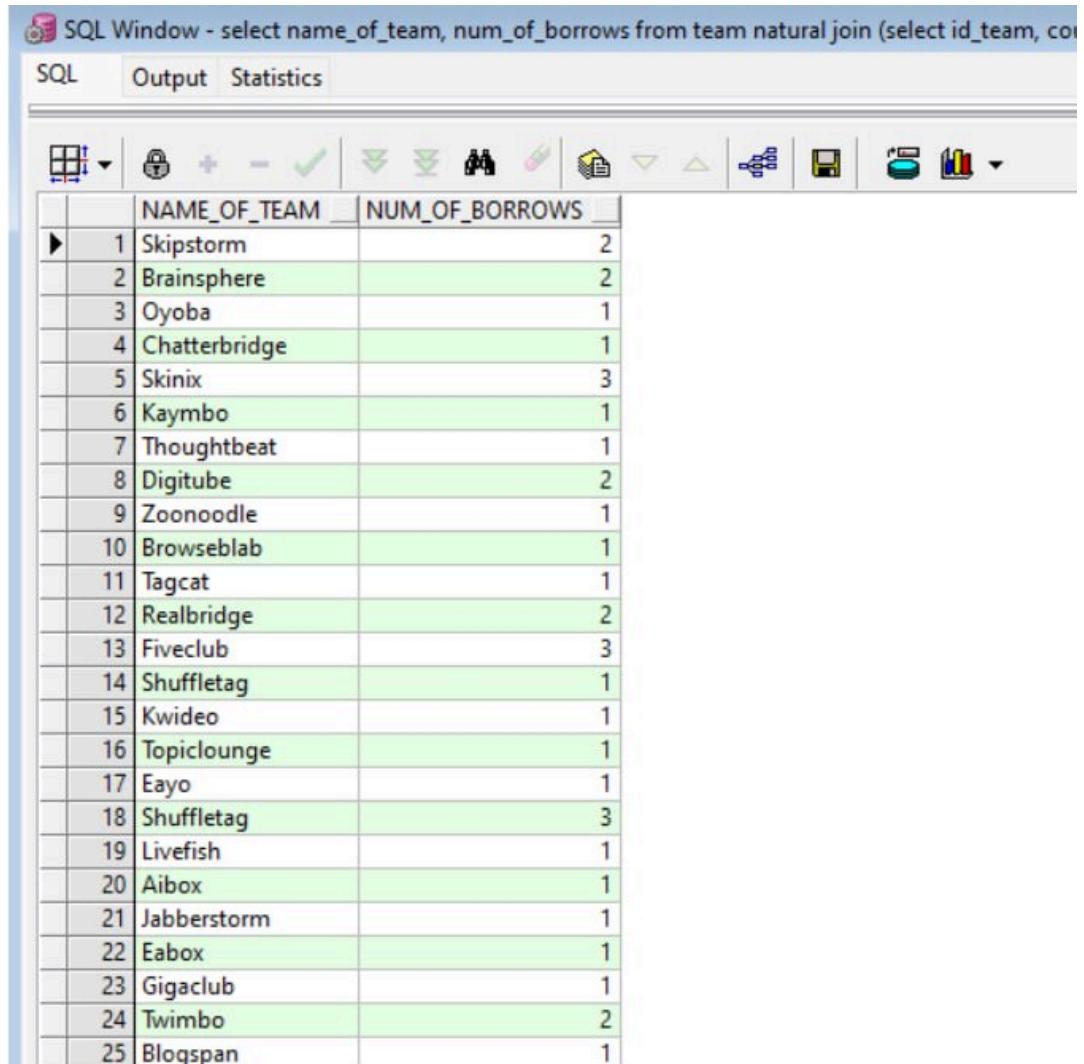
:2

השאילתא מציגה את כמות הציוד הרפואי שכרגע נמצא בהשקרה אצל העובדים שבצווות.
ויצג למשתמש: שם הצוות וכמות הציוד שמשאל.
צילום הרצה:



SQL Window - select name_of_team, num_of_borrows from team natural join (select id_team, count(id_team) as num_of_borrows from employee natural join equipment where status_e = 1 group by id_team)T;

צילום פלט:



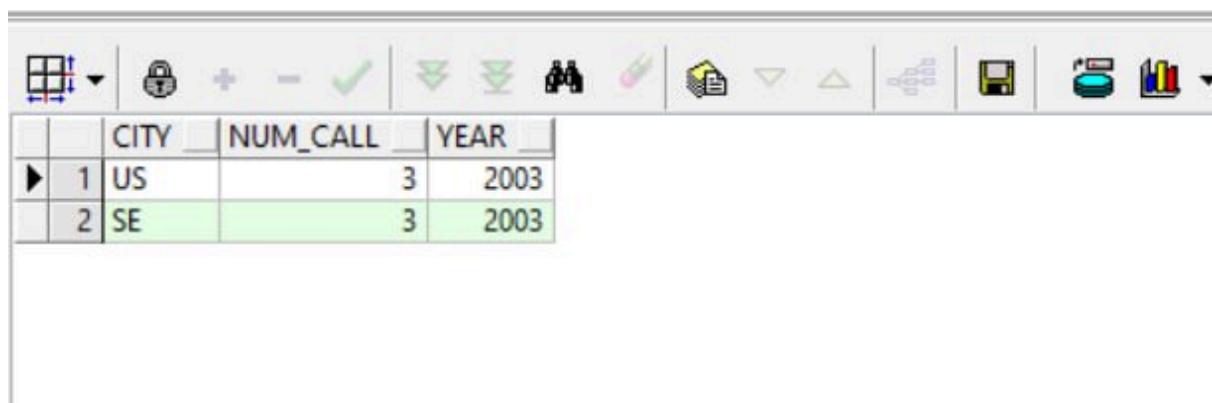
	NAME_OF_TEAM	NUM_OF_BORROWS
1	Skipstorm	2
2	Brainsphere	2
3	Oyoba	1
4	Chatterbridge	1
5	Skinix	3
6	Kaymbo	1
7	Thoughtbeat	1
8	Digitube	2
9	Zoonoodle	1
10	Browseblab	1
11	Tagcat	1
12	Realbridge	2
13	Fiveclub	3
14	Shuffletag	1
15	Kwideo	1
16	Topiclounge	1
17	Eayo	1
18	Shuffletag	3
19	Livefish	1
20	Aibox	1
21	Jabberstorm	1
22	Eabox	1
23	Gigaclub	1
24	Twimbo	2
25	Bloqspan	1

:3

:3

השאילתה מציגה את העיר/ערים שבהה לה הכי הרבה פניות בשנת 2003.
 יוצג למשתמש: שם העיר , מס הפלניות ואת השנה .
 צילום הרצה ופלט:

```
select city, num_call, 2003 as year|
from
(select city, count(*) as num_call
from call natural join hospital natural join patient
where EXTRACT(YEAR FROM date_call) = 2003
group by city)T
where num_call >= all
(select count(*) as num_call
from call natural join hospital natural join patient
where EXTRACT(YEAR FROM date_call) = 2003
group by city);
```



	CITY	NUM_CALL	YEAR
▶	1	US	3 2003
	2	SE	3 2003

:4

השאילתת מציגה פרטיהם של צוותים שלא טיפולו בקוריאות מדריכות 5 ופרטיה הקרייה הינם critical יוצג למשתמש: שם הצוות , מספר עובדים בצוות.

צילום הרצה ופלט:

The screenshot shows a software interface for running SQL queries. At the top, there are three tabs: 'SQL', 'Output', and 'Statistics'. The 'SQL' tab is active, displaying the following query:

```
select name_of_team, amount
from team
where id_team not in(
select id_team
from call
where urgency = 5 and details = 'critical');
```

Below the query, the 'Output' tab is active, displaying a table with the results of the query. The table has two columns: 'NAME_OF_TEAM' and 'AMOUNT'. The data is as follows:

	NAME_OF_TEAM	AMOUNT
1	Oyoloo	3
2	Jazzy	2
3	Brainbox	8
4	Skyba	10
5	Photolist	9
6	Skipstorm	7
7	Oyoba	5
8	Realmix	2
9	Voonyx	6
10	Brainsphere	6
11	Realcube	8
12	Meembee	10
13	Thoughtbridge	6

שאלות delete:

1. מחיקת כל הציוד (equipment) שהושאל על ידי עובד שכרכו מתחת ל-3000 ולא הוחזר עדין.
- צילום הרצה:

--delete query 1:

```
DELETE FROM Equipment
WHERE id_employee IN (
    SELECT id_employee FROM Employee WHERE salary <3000
) AND status_e = 1;
```

צילום בסיס הנתונים לפני המחקה:

	ID_EMPLOYEE	NAME_EMP	SALARY	ID_TEAM	DATE_BORROW	NAME_E	ID_EQUIPMENT	STATUS_E
52	612	Ava	3590	490	01/09/2015	▼ Axe	106	1
53	330	Amy	1706	116	10/01/2007	▼ Helmet	372	1
54	935	Eli	12590	181	13/02/2024	▼ Wrench	191	0
55	330	Amy	1706	116	29/11/2021	▼ Saw	815	0
56	882	Ben	14147	421	14/06/2021	▼ Utility knife	608	0
57	520	Amy	14906	416	04/09/2013	▼ Screwdriver	813	1
58	915	Amy	13173	418	18/07/2011	▼ Wrench	419	1
59	859	Amy	1096	196	20/07/2009	▼ Ladder	282	1
60	422	Ava	10947	357	14/02/2006	▼ Gloves	670	1
61	823	Ava	3704	130	22/06/2018	▼ Shovel	238	0
62	845	Eli	12824	435	29/03/2022	▼ Goggles	632	1
63	842	Ben	2452	352	15/03/2024	▼ Heat gun	200	0
64	433	Ava	12402	441	07/04/2015	▼ Flashlight	814	1
65	978	Eli	3496	473	26/11/2018	▼ Screwdriver	565	1
66	879	Ava	1528	352	27/05/2006	▼ Ladder	177	0
67	374	Ben	2059	283	05/05/2004	▼ Utility knife	246	1
68	554	Amy	9876	300	26/04/2015	▼ Toolbox	121	0
69	965	Amy	14860	404	01/09/2019	▼ Screwdriver	622	1
70	949	Amy	9152	271	17/04/2011	▼ Crowbar	134	1

ניתן לראות שהציוד ש-330 amy השכירה צריך להימחק (כМОון שעוד נתנו אותה כדוגמה)
צילום בסיס הנתונים לאחר המחקה:

ניתן לראות שהציוד ש-330 amy השכירה נמחק מהרשימת ציודים.

```
select id_equipment
from equipment where id_equipment =372
```

ID_EQUIPMENT

2. מחיקת כל הפסיכינטים שטיפלו בהם צוותים עם פחות מ 3 עובדים.
צילום הרצה:

--delete query 2:

```
DELETE FROM Patient WHERE id_call IN (
    SELECT id_call FROM Call WHERE id_team IN (
        SELECT id_team FROM Employee
        GROUP BY id_team
        HAVING COUNT(id_employee) < 3
    )
);
```

צילום בסיס הנתונים לפני המחיקה:
כל הפסיכינטים בראשימה הבאה צריכים להימחק:

	ID_PATIENT	NAME_P	AGE	URGENCY	DETAILS	DATE_CALL	ID_HOSPITAL	AMOUNT	NAME_C
▶	1	139	Ned	7	4 stable	08/05/1981	370	2	Pixope
▶	2	390	Lou	21	3 deceased	29/09/1991	135	2	Abata
▶	3	525	Kit	49	1 critical	20/12/2014	390	2	Browset
▶	4	882	Lia	111	1 unconscious	23/04/2014	434	2	Dablist
▶	5	770	Ted	37	3 deceased	02/11/2009	421	2	Tagcat
▶	6	654	Eve	11	3 minor	26/01/2014	412	2	Kwideo
▶	7	111	Pam	107	4 deceased	05/04/1989	428	2	Snaptop
▶	8	233	Kit	120	4 missing	02/11/1989	342	2	Nloung
▶	9	731	Lia	114	5 critical	16/05/1970	139	2	Ntags
▶	10	659	Zed	77	3 minor	26/01/2014	412	2	Kwideo
▶	11	204	Ted	64	3 critical	22/09/1974	195	2	Jaxbean
▶	12	692	Hal	27	4 unconscious	11/05/1994	331	2	Jaxwork
▶	13	115	Hal	31	5 critical	16/05/1970	139	2	Ntags

צילום בסיס הנתונים לאחר המחיקה:

```
select * from patient where id_patient=139;
```

	ID_PATIENT	NAME_P	AGE	ID_CALL
▶	1	Ned	7	

כאן ניתן לראות שבאמת פ齊ינט מס' 139 נמחק מרשימת הפסיכינטים.
(וככה אם נבדוק עבור כלום נראה שהוא נמחקו)

שאלות update:

1. השאלה מוסיפה בונוס של 500 שקלים לכל העובדים שהיו בקריאות בחודש 04 בשנת 2010.
- צילום הרצה:

```
--update query 1:
```

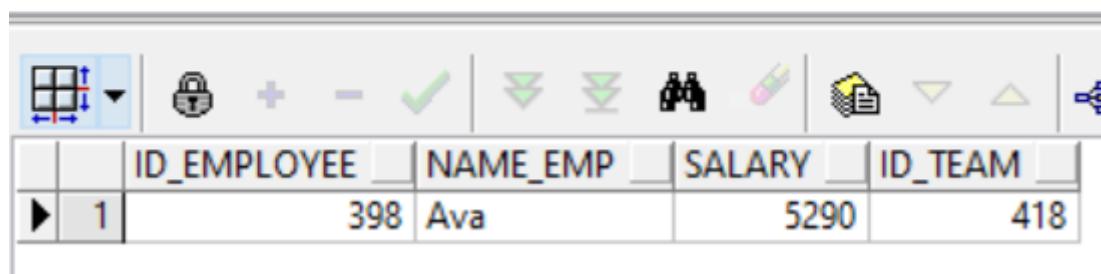
```
UPDATE Employee
SET salary = salary + 500
WHERE id_employee IN (
    SELECT id_employee
    FROM Call natural join employee
    WHERE EXTRACT(MONTH FROM date_call) = 4
    AND EXTRACT(YEAR FROM date_call) = 2010);
```

צילום בסיס הנתונים לפני העדכון:

ID_EMPLOYEE	NAME_EMP	SALARY	ID_CALL	URGENCY	DETAILS	DATE_CALL
898	Amy	10327	413	4	minor	10/11/2015
408	Ben	10012	414	1	missing	25/09/2003
520	Amy	14906	416	5	stable	10/12/1973
816	Eli	9977	417	3	missing	22/10/1981
398	Ava	4790	418	4	minor	22/04/2010
915	Amy	13173	418	4	minor	22/04/2010
932	Ava	8631	419	3	critical	06/12/1974
882	Ben	14147	421	3	deceased	02/11/2009
317	Ava	4356	422	1	deceased	26/01/1995

צילום בסיס הנתונים לאחר העדכון:

```
select *
from employee where id_employee=398;
```



The screenshot shows the Oracle SQL Developer interface with a database table viewer. The table has columns: ID_EMPLOYEE, NAME_EMP, SALARY, and ID_TEAM. A single row is selected, showing ID_EMPLOYEE 398, NAME_EMP Ava, SALARY 5290, and ID_TEAM 418. The salary value 5290 is highlighted in yellow, indicating it was updated. The top toolbar includes icons for new table, insert, delete, and search.

	ID_EMPLOYEE	NAME_EMP	SALARY	ID_TEAM
▶	1	398	5290	418

כמו שניתן לראות אכן המשכורת של העובד התעדכונה ל 500 שקלים יותר.

.2

השאילתת מעכנת את דחיפות הקרייה (urgency) ל-5 אם מספר העובדים במצב (amount) קטן מ-3 ופרטיה הקרייה (details) שווה ל-'critical'.
צילום הרצה:

```
--update query 2:
UPDATE Call
SET urgency = 5
WHERE details = 'critical'
AND id_team IN (
    SELECT id_team
    FROM Team
    WHERE amount < 3
);
```

צילום בסיס הנתונים לפני העדכון:

	ID_TEAM	ID_CALL	URGENCY	DETAILS	DATE_CALL	ID_HOSPITAL	AMOUNT	NAME_OF_TEAM
1	456	456	4	critical	20/01/2005	456	2	Dabvine
2	492	492	1	critical	30/06/1972	492	2	Jabberbean
3	241	408	4	critical	05/07/1982	176	2	Dabvine
4	195	195	3	critical	22/09/1974	195	2	Jaxbean
▶ 5	338	338	2	critical	08/07/2016	338	2	Agimba
6	139	139	5	critical	16/05/1970	139	2	Ntags
7	390	390	1	critical	20/12/2014	390	2	Browsertype

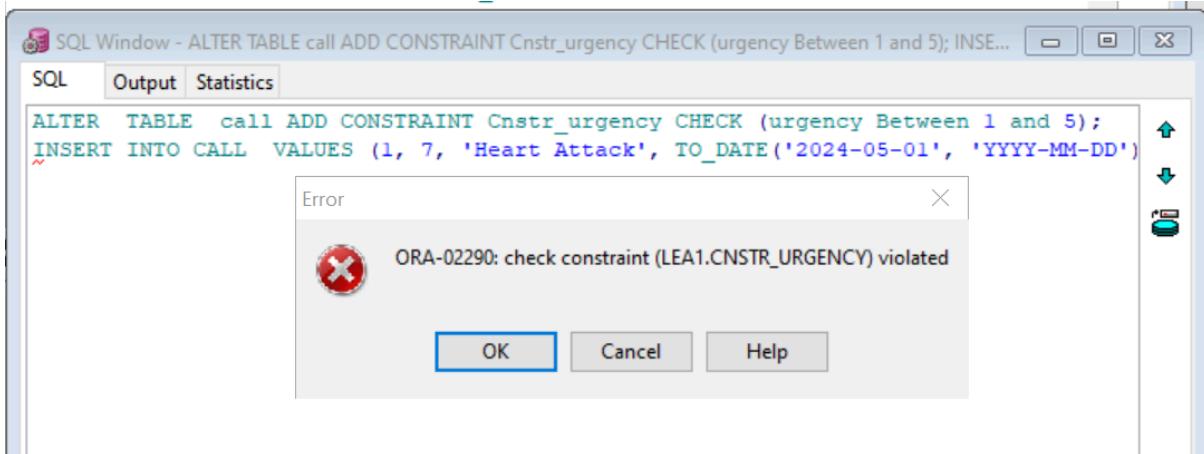
צילום בסיס הנתונים לאחר העדכון:

	ID_TEAM	ID_CALL	URGENCY	DETAILS	DATE_CALL	ID_HOSPITAL	AMOUNT	NAME_OF_TEAM
▶ 1	456	456	5	critical	20/01/2005	456	2	Dabvine
2	492	492	5	critical	30/06/1972	492	2	Jabberbean
3	241	408	5	critical	05/07/1982	176	2	Dabvine
4	195	195	5	critical	22/09/1974	195	2	Jaxbean
5	338	338	5	critical	08/07/2016	338	2	Agimba
6	139	139	5	critical	16/05/1970	139	2	Ntags
7	390	390	5	critical	20/12/2014	390	2	Browsertype

אלוצים:

אלוץ ראשון:

הוסף אלוץ check בדחיפות קרייה כך שייהי חייבים להכניס מספר בין 1-5
כأن בתמונה ניסינו להכניס דחיפות 7 וקפזה הודעת שגיאה.

**אלוץ שני:**

הוספנו בדיקת מחדל בסעיפים של ציוד שם לא הכניסו סעיפים - הסעיפים יהיה 0.
כמו שניתן לראות הכננו ציוד בלי סעיפים ואוטומטית נכנס 0 בסעיפים.

ALTER TABLE equipment MODIFY status_e DEFAULT 0; INSERT INTO EQUIPMENT(date_borrow,name_e,id_equipment,id_employee) VALUES (TO_DATE('2024-05-01', 'YYYY-MM-DD'), 'Laptop', 800, 335); select * from equipment where id_equipment=800;

אלוץ שלישי:

אי אפשר להשאיר את העמודה של תאריך בקרייה null - חייב להכניס תאריך.

--Constraint query 3: ALTER TABLE Call MODIFY date_call DATE NOT NULL; INSERT INTO Call (id_call, urgency, details, date_call, id_team, id_hospital) VALUES (999, 5, 'critical', NULL, 100, 202);

שלב 3: תכנות:

פונקציה 1 : assign_emergency_team
תיאור: מקבלת פרמטר מס' קרייה ומוצא לקרייה צוות.

```

CREATE OR REPLACE FUNCTION assign_emergency_team(
    p_call_id IN Call.id_call%TYPE) RETURN SYS_REFCURSOR
IS
    v_team_cursor SYS_REFCURSOR;
    v_team_id Team.id_team%TYPE;
    --v_employee_count NUMBER;
BEGIN
    -- Assign team
    SELECT id_team INTO v_team_id
    FROM (
        SELECT t.id_team, COUNT(e.id_employee) as employee_count
        FROM Team t
        LEFT JOIN Employee e ON t.id_team = e.id_team
        GROUP BY t.id_team
        HAVING COUNT(e.id_employee) BETWEEN 1 AND 10
        ORDER BY
            CASE
                WHEN COUNT(e.id_employee) = 3 THEN 1
                ELSE 2
            END,
            DBMS_RANDOM.VALUE
    )
    WHERE ROWNUM = 1;
    UPDATE Call
    SET id_team = v_team_id
    WHERE id_call = p_call_id;

    OPEN v_team_cursor FOR
        SELECT t.id_team, t.name_of_team, e.id_employee, e.name_emp
        FROM Team t
        JOIN Employee e ON t.id_team = e.id_team
        WHERE t.id_team = v_team_id;

    RETURN v_team_cursor;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001, 'No suitable team found for the emergency call');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'An error occurred: ' || SQLERRM);
END;

```

פרוצדורה 1 : manage_equipment_assignment
תיאור: מקבל מזהה קרייה ומקרה לעובדים של הוצאות שמתפל בה ציוד.

```

CREATE OR REPLACE PROCEDURE manage_equipment_assignment(
    p_call_id IN Call.id_call%TYPE)
IS
    TYPE equipment_record IS RECORD (
        id_equipment Equipment.id_equipment%TYPE,
        name_e Equipment.name_e%TYPE
    );
    TYPE equipment_table IS TABLE OF equipment_record;
    v_equipment equipment_table;
    v_team_id Team.id_team%TYPE;
    v_employee_id Employee.id_employee%TYPE;

    equipment_already_assigned EXCEPTION;
    PRAGMA EXCEPTION_INIT(equipment_already_assigned, -20003);

BEGIN
    -- Get the team assigned to the call
    SELECT id_team INTO v_team_id
    FROM Call
    WHERE id_call = p_call_id;

    -- Get available equipment
    SELECT id_equipment, name_e
    BULK COLLECT INTO v_equipment
    FROM Equipment
    WHERE status_e = 0
    AND ROWNUM <= 5; -- Limit to 5 pieces of equipment

    -- Assign equipment to employees in the team
    FOR i IN 1..v_equipment.COUNT LOOP
        BEGIN
            -- Get an available employee from the team
            SELECT id_employee INTO v_employee_id
            FROM Employee
            WHERE id_team = v_team_id
            AND id_employee NOT IN (
                SELECT id_employee
                FROM Equipment
                WHERE status_e = 1
            )
            AND ROWNUM = 1;
        END;
        -- Assign equipment to the employee
        UPDATE Equipment
        SET id_employee = v_employee_id,
            status_e = 1,
            date_borrow = SYSDATE
        WHERE id_equipment = v_equipment(i).id_equipment;

        -- Log the assignment (now just printing to output instead of inserting to a table)
        DBMS_OUTPUT.PUT_LINE('Assigned equipment ' || v_equipment(i).id_equipment ||
                             ' to employee ' || v_employee_id ||
                             ' for call ' || p_call_id);

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('No available employee for equipment: ' || v_equipment(i).name_e);
            WHEN equipment_already_assigned THEN
                DBMS_OUTPUT.PUT_LINE('Equipment already assigned: ' || v_equipment(i).name_e);
        END;
    END LOOP;

    COMMIT;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('No available employee for equipment: ' || v_equipment(i).name_e);
        WHEN equipment_already_assigned THEN
            DBMS_OUTPUT.PUT_LINE('Equipment already assigned: ' || v_equipment(i).name_e);
    END;
END LOOP;

COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20004, 'An error occurred in equipment assignment: ' || SQLERRM);
END;

```

תכנית ראשית 1:

תיאור:

התוכנית יוצרת קרייה חדשה.

ואז מקצת לקרייה צוות שיטפל בקרייה ע"י קרייה לפונקציה assign_emergency_team אשר מכנה התוכנית מציגה את הפרטים של הצוות שהוקצה ואת העובדים שבחברים בו.

אחרי שהוקצה צוות לטפל בקרייה צריך להקצות ציוד.
לכן התוכנית קוראת לפונקצורה assign_emergency_team כדי להקצות ציוד לחבריו הצוות.
בנוסף הקצאות הציוד מודפסות למשתמש.

```

DECLARE
    v_call_id Call.id_call%TYPE;
    v_team_cursor SYS_REFCURSOR;
    v_team_id Team.id_team%TYPE;
    v_team_name Team.name_of_team%TYPE;
    v_employee_id Employee.id_employee%TYPE;
    v_employee_name Employee.name_emp%TYPE;
BEGIN
    -- Simulate creating a new emergency call
    INSERT INTO Call (id_call, urgency, details, date_call, Id_Hospital)
    VALUES (932, 5, 'critical', SYSDATE, 412)
    RETURNING id_call INTO v_call_id;

    DBMS_OUTPUT.PUT_LINE('New emergency call created with ID: ' || v_call_id);

    -- Assign an emergency team to the call
    v_team_cursor := assign_emergency_team(v_call_id);

```

```

    -- Display assigned team information
    DBMS_OUTPUT.PUT_LINE('Assigned team details:');
    LOOP
        FETCH v_team_cursor INTO v_team_id, v_team_name, v_employee_id, v_employee_name;
        EXIT WHEN v_team_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Team ID: ' || v_team_id || ', Team Name: ' || v_team_name ||
                             ', Employee ID: ' || v_employee_id || ', Employee Name: ' || v_
    END LOOP;
    CLOSE v_team_cursor;

    -- Manage equipment assignment for the call
    manage_equipment_assignment(v_call_id);

    DBMS_OUTPUT.PUT_LINE('Emergency response process completed for call ID: ' || v_call_id)

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        ROLLBACK;

```

FND:

הרצה:

```

v_emergency_call_employee_name_employee_name_employee_name
BEGIN
    -- Simulate creating a new emergency call
    INSERT INTO Call (id_call, urgency, details, date_call, Id_Hospital)
    VALUES (932, 5, 'critical', SYSDATE, 412)
    RETURNING id_call INTO v_call_id;

```

ניתן לראות שפרט הקריאה נכנסו למערכת:

The screenshot shows the Oracle SQL Developer interface with the 'Call' table selected. The table has columns: ID_CALL, URGENCY, DETAILS, DATE_CALL, ID_TEAM, and ID_HOSPITAL. A single row is displayed with values: 1, 932, 5, 'critical', 10/07/2024 18:07:52, 292, and 412 respectively.

	ID_CALL	URGENCY	DETAILS	DATE_CALL	ID_TEAM	ID_HOSPITAL
▶	1	932	5 critical	10/07/2024 18:07:52	292	412

וכפי שניתן לראות הוקצה צוות לטפל בקריאה.

```

New emergency call created with ID: 932
Assigned team details:
Team ID: 292, Team Name: Flashpoint, Employee ID: 369, Employee Name: Ava
Team ID: 292, Team Name: Flashpoint, Employee ID: 909, Employee Name: Eli
Team ID: 292, Team Name: Flashpoint, Employee ID: 590, Employee Name: Ava

```

מודפס למסך את העובדים של הצוות.

The screenshot shows the Oracle SQL Developer interface with the 'employee' table selected. A query 'select * from employee where id_team=292;' is run. The results show three employees: Ava (Employee ID 369), Eli (Employee ID 909), and another Ava (Employee ID 590).

	ID_EMPLOYEE	NAME_EMP	SALARY	ID_TEAM
▶	1	369	Ava	13233
	2	909	Eli	14666
	3	590	Ava	4483

עכשו מתבצעת קראה לפואדרה כדי להקצת ציוד לטיפול בקריאה:

```

Team ID: 292, Team Name: Flashpoint, Employee ID: 590, Employee Name: Ava
Assigned equipment 254 to employee 590 for call 932
Assigned equipment 964 to employee 909 for call 932
No available employee for equipment: Pliers
No available employee for equipment: Sander
No available employee for equipment: Goggles
Emergency response process completed for call ID: 932

```

כמו שניתן לראות הודפס למסך איזה ציוד הוקצת לעובדים בשבייל לטפל בקריאה.

בז"ד
הנתונים
עדכן
והוקצת
ציוד
לעובד.

The screenshot shows the Oracle SQL Developer interface with the 'equipment' table selected. A query 'select * from equipment where id_employee in (select id_employee from employee where id_team = 292);' is run. The results show two pieces of equipment assigned to employees 590 and 909: Stud finder (Equipment ID 254) and Compressor (Equipment ID 964).

	DATE_BORROW	NAME_E	ID_EQUIPMENT	STATUS_E	ID_EMPLOYEE
▶	10/07/2024 18:07:52	Stud finder	254	1	590
	10/07/2024 18:07:52	Compressor	964	1	909

פונקציה 2:

תיאור: הפונקציה `get_team_most_borrowed` נועדה להחזיר את מזהה הצוות (ID team) אשר ש悲哀יל את מספר הזרים הגדול ביותר של ציוד.

```

  RETURN Team.id_team%TYPE
AS
  -- Explicit cursor declaration
  CURSOR team_cursor IS
    SELECT e.id_team, COUNT(eq.id_equipment) AS equipment_count
    FROM Employee e
    JOIN Equipment eq ON e.id_employee = eq.id_employee
    GROUP BY e.id_team
    ORDER BY COUNT(eq.id_equipment) DESC;

  -- Record type for storing team data
  TYPE team_rec_type IS RECORD (
    id_team Team.id_team%TYPE,
    equipment_count NUMBER
  );
  team_rec team_rec_type;

  -- Variable to hold the team with most borrowed equipment
  v_team_id Team.id_team%TYPE;
BEGIN
  -- Open explicit cursor
  OPEN team_cursor;

  -- Fetch the first row from the explicit cursor
  FETCH team_cursor INTO team_rec;
  IF team_cursor%FOUND THEN
    v_team_id := team_rec.id_team;
  ELSE
    v_team_id := NULL;
  END IF;

  -- Close explicit cursor
  CLOSE team_cursor;

  RETURN v_team_id;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN NULL;
  WHEN OTHERS THEN
    RAISE;
END;

```

פרוצדורה 2:

תיאור: הפרוצדורה update_team_salary_equipment מעודכנת את השכר של העובדים בצוות מסוים בהתאם על מספר הפרויקטים שהם השאילו, ומחזירה מידע על העובדים המעודכנים.

```

CREATE OR REPLACE PROCEDURE update_team_salary_equipment(p_team_id NUMBER)
IS
    -- Explicit cursor declaration
    CURSOR emp_cursor IS
        SELECT e.id_employee, COUNT(eq.id_equipment) AS equipment_count
        FROM Employee e
        LEFT JOIN Equipment eq ON e.id_employee = eq.id_employee
        WHERE e.id_team = p_team_id
        GROUP BY e.id_employee;

    -- Ref cursor declaration for returning results
    TYPE ref_cursor_type IS REF CURSOR;
    ref_cursor ref_cursor_type;

    -- Record type for employee data
    TYPE emp_rec_type IS RECORD (
        id_employee Employee.id_employee%TYPE,
        equipment_count NUMBER
    );
    emp_rec emp_rec_type;

    -- Record type for ref cursor data
    TYPE ref_emp_rec_type IS RECORD (
        id_employee Employee.id_employee%TYPE,
        name_emp Employee.name_emp%TYPE,
        salary Employee.salary%TYPE
    );
    ref_emp_rec ref_emp_rec_type;

    -- Variables for dynamic salary adjustment
    salary_increment NUMBER;

BEGIN
    -- Open the explicit cursor
    OPEN emp_cursor;

    -- Fetch each employee record
    LOOP
        FETCH emp_cursor INTO emp_rec;
        EXIT WHEN emp_cursor%NOTFOUND;

        -- Calculate salary increment based on equipment count
        IF emp_rec.equipment_count > 5 THEN
            salary_increment := 1000; -- Increment for more than 20 equipment
        ELSIF emp_rec.equipment_count > 2 THEN
            salary_increment := 500; -- Increment for 11-20 equipment
        ELSE
            salary_increment := 100; -- Increment for 10 or fewer equipment
        END IF;

        -- Update employee salary based on equipment count
        UPDATE Employee
        SET salary = salary + salary_increment
        WHERE id_employee = emp_rec.id_employee;

        -- Conditional action based on equipment count
        IF emp_rec.equipment_count > 10 THEN
            DBMS_OUTPUT.PUT_LINE('Employee ID ' || emp_rec.id_employee || ' has borrowed more than 10 pieces of equipment.');
        END IF;
    END LOOP;

    -- Close the explicit cursor
    CLOSE emp_cursor;

    -- Commit the transaction
    COMMIT;

```

```
-- Commit the transaction
COMMIT;

-- Return a ref cursor with updated employee details
OPEN ref_cursor FOR
  SELECT e.id_employee, e.name_emp, e.salary
  FROM Employee e
  WHERE e.id_team = p_team_id;

-- Additional processing with the ref cursor if needed
LOOP
  FETCH ref_cursor INTO ref_emp_rec;
  EXIT WHEN ref_cursor%NOTFOUND;
END LOOP;
CLOSE ref_cursor;

EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
    RAISE;
END;
```

תוכנית ראשית 2:

תיאור: איתור הוצאות עם מספר השאלות הגבוה ביותר: קוראת לפונקציה `get_team_most_borrowed` כדי לקבל את מזוהה הוצאות (`id_team`) עם מספר הפריטים הגבוה ביותר ש问道ה ש问道ה. בדיקת קיום צוות: אם נמצא צוות עם השאלות, מושיכים; אחרת, מוצגת הודעה שאין צוות צזה. הצגת מידע שכר נוכחי: לפני העדכון, מוצג מידע השכר הנוכחי של כל העובדים בצוות. עדכון שכר העובדים בצוות: קוראת לפורצתו `update_team_salary_equipment` כדי לעדכן את השכר של כל העובדים בצוות בהתאם על מספר הפריטים שה问道ה.

הצגת תוצאות לאחר העדכון:

- מוצג מספר הפריטים שה问道ה על ידי הוצאות.
- מוצג מידע השכר המעודכן של כל העובדים בצוות.

הצגת הודעה אם לא נמצא צוות: אם לא נמצא צוות עם השאלות, מוצגת הודעה מתאימה. **טיפול בשגיאות:** במקרה של שגיאה, מוצגת הודעה שגיאה ומועליה החיריג מחדש.

```
CREATE OR REPLACE PROCEDURE main_combined_procedure IS
    v_team_id Team.id_team%TYPE;
    v_equipment_count NUMBER;
BEGIN
    -- Get the team with the most borrowed equipment
    v_team_id := get_team_most_borrowed;

    IF v_team_id IS NOT NULL THEN
        -- Display current salary information before the update
        DBMS_OUTPUT.PUT_LINE('Current Salary Information:');
        FOR emp_rec IN (
            SELECT e.id_employee, e.name_emp, e.salary
            FROM Employee e
            WHERE e.id_team = v_team_id
        ) LOOP
            DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_rec.id_employee ||
                ', Name: ' || emp_rec.name_emp ||
                ', Salary: ' || emp_rec.salary);
        END LOOP;

        -- Update salary for employees in the team based on equipment count
        update_team_salary_equipment(v_team_id);

        -- Display results after update
        SELECT COUNT(*) INTO v_equipment_count
        FROM Equipment e
        JOIN Employee emp ON e.id_employee = emp.id_employee
        WHERE emp.id_team = v_team_id;

        DBMS_OUTPUT.PUT_LINE('Team ID with most borrowed equipment: ' || v_team_id);
        DBMS_OUTPUT.PUT_LINE('Number of borrowed equipment: ' || v_equipment_count);

        -- Display updated salary information
        DBMS_OUTPUT.PUT_LINE('Updated Salary Information:');
        FOR emp_rec IN (
            SELECT e.id_employee, e.name_emp, e.salary
            FROM Employee e
        ) LOOP
            DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_rec.id_employee ||
                ', Name: ' || emp_rec.name_emp ||
                ', Salary: ' || emp_rec.salary);
        END LOOP;
    END IF;
END;
```

```
FOR emp_rec IN (
    SELECT e.id_employee, e.name_emp, e.salary
    FROM Employee e
    WHERE e.id_team = v_team_id
) LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_rec.id_employee ||
                         ', Name: ' || emp_rec.name_emp ||
                         ', Salary: ' || emp_rec.salary);
END LOOP;

ELSE
    DBMS_OUTPUT.PUT_LINE('No team found with borrowed equipment.');
END IF;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        RAISE;
END;
```

הרצה:

Current Salary Information:

Employee ID: 989, Name: Amy, Salary: 25910

Employee ID: 532, Name: Amy, Salary: 25469

Employee ID: 463, Name: Ben, Salary: 28489

Team ID with most borrowed equipment: 237

Number of borrowed equipment: 8

Updated Salary Information:

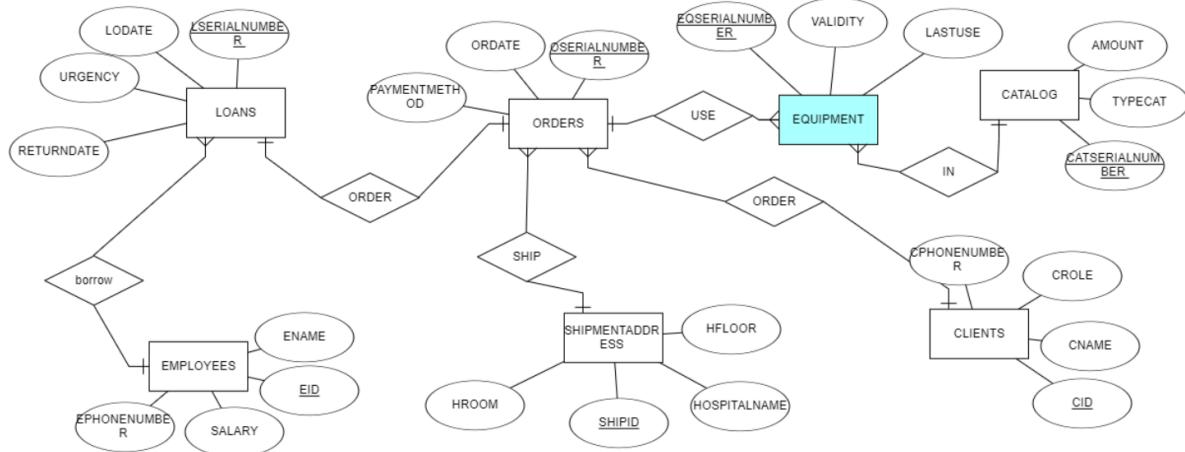
Employee ID: 989, Name: Amy, Salary: 26010

Employee ID: 532, Name: Amy, Salary: 25569

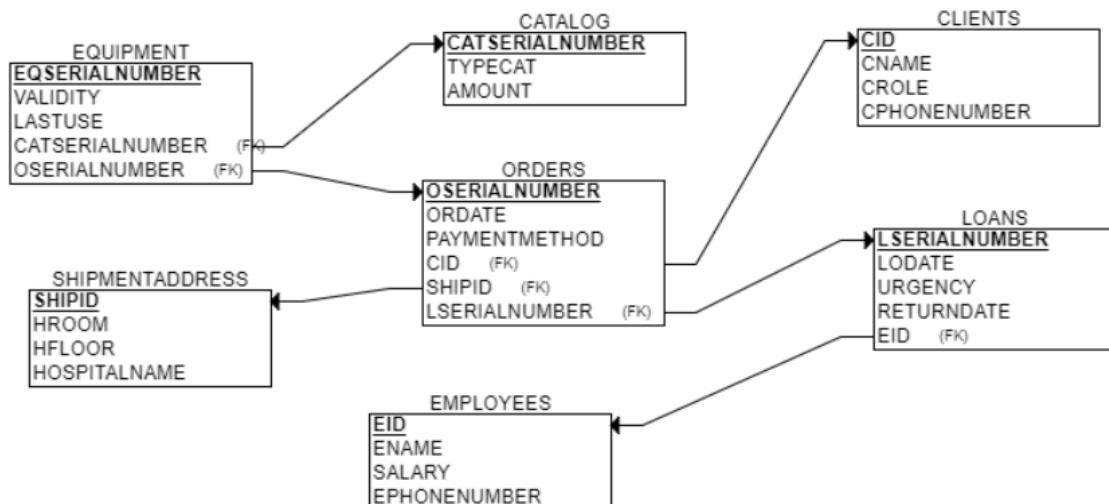
Employee ID: 463, Name: Ben, Salary: 28989

שלב 4: אינטגרציה:

תרשים ERD החדש:



תרשים DSD חדש:



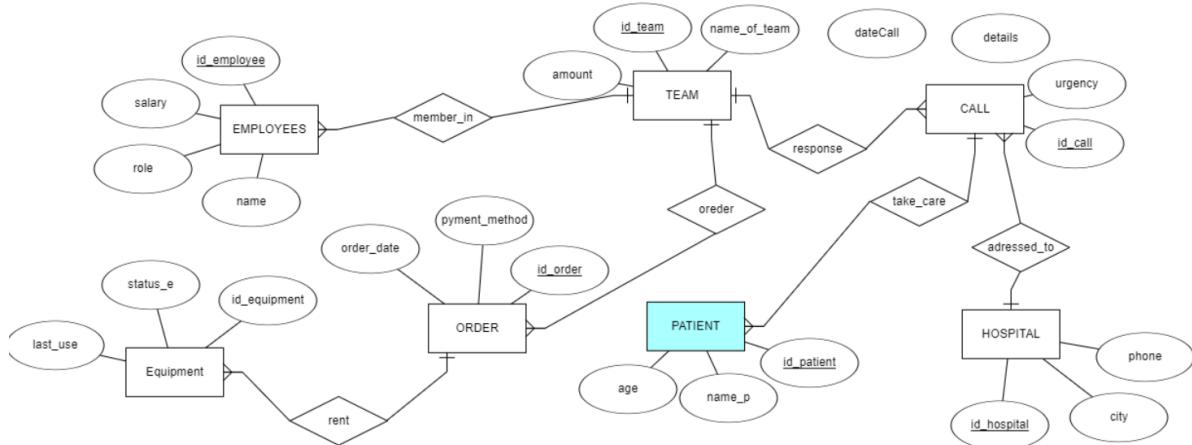
אינטגרציה:

בבסיס הנתונים שלנו היה מערכת ניהול ארגון מד"א - ניהול הקריאות, חולמים, צוותים , עובדים ... אנחנו קיבלנו מערכת שמנהלת השכורת ציוד ושיילבנו אותה במערכת שלנו.

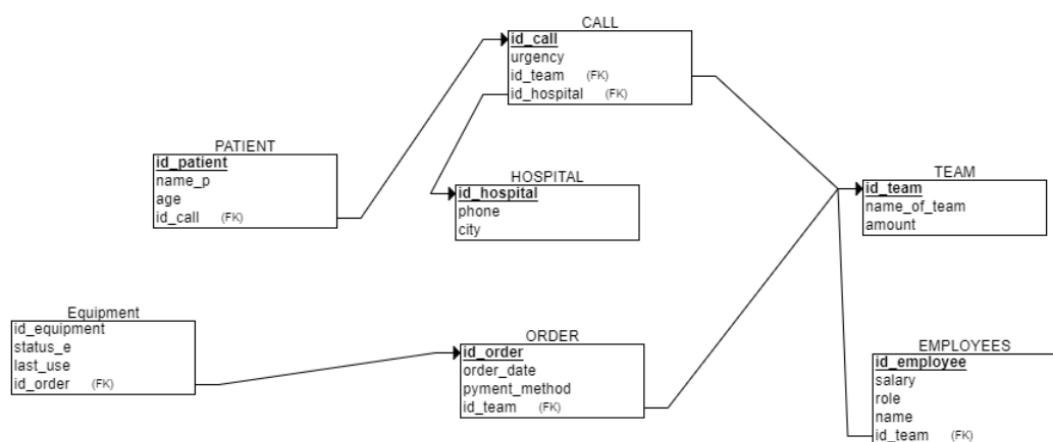
הסתכלנו על ה-ERD שקיבלנו וראינו שטבלת equipment שלנו היא בעצם שילוב של loans equipments ו-equipment בתရשים ERD שקיבלנו ולכן שיילבנו את הטבלאות הללו לטבלה אחת שנקראת equipment

הוסףנו מה ERD שקיבלנו את הטבלה order וחיבורנו אותה ל team כך שכעת צוות יכולים לבצע הזמנה של ציוד.

תרשים ERD מעודכן:



תרשים DSD מעודכן:



מבחן 1: על בסיס הנתונים שלנו:

```
SELECT *
FROM EmployeeTeams;

SELECT *
FROM EmployeeTeams
WHERE avg_team_salary > 5000;
```

	ID_EMPLOYEE	NAME_EMP	SALARY	NAME_OF_TEAM	AVG_TEAM_SALARY	
►	1	111	Paul	9264	Yodo	9264
	2	121	Aiden	12681	Jaxnation	13770.5
	3	140	Leah	14621	Innojam	8742.5
	4	171	Alice	2349	Rhyzero	8214.5
	5	300	Ben	5972	Twitternation	5972
	6	951	Eli	14392	Agimba	14392
	7	411	Ava	5627	Voonder	5627
	8	965	Amy	14860	Jaxnation	13770.5
	9	937	Eli	13899	Blogpad	11283.5
	10	538	Amy	13349	Ntags	13349
	11	447	Ava	10726	Oyoloo	6178.5

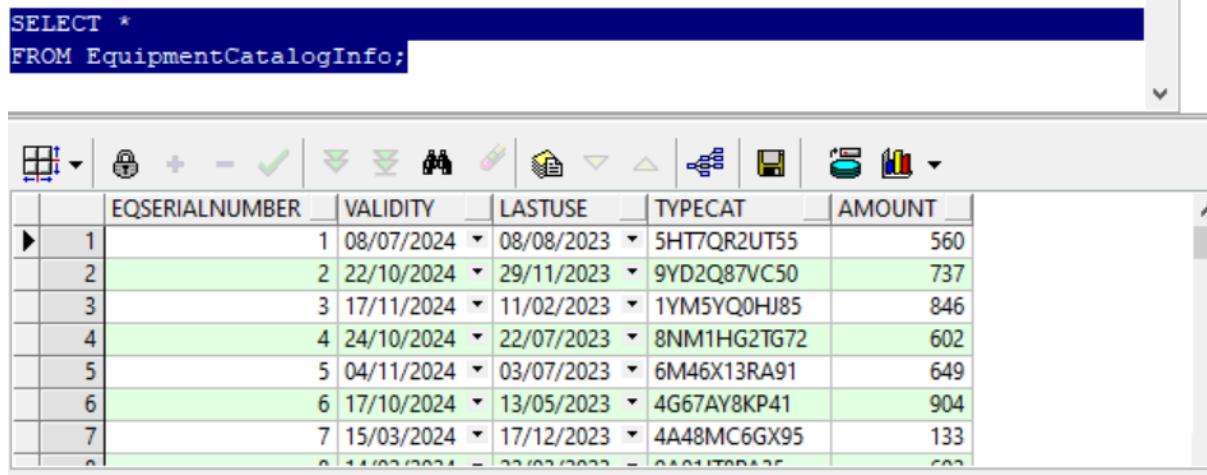
המבחן מציג נתונים עובדים יחד עם שם הצוות שלהם ומחשב את השכר הממוצע של כל צוות.

מבחן 2:

המבחן מציג נתונים על ציוד יחד עם פרטי הקטלוג המתאימים, כולל סוג וערך הציוד, על סמך חיבור בין טבלאות EQUIPMENT CATALOG.

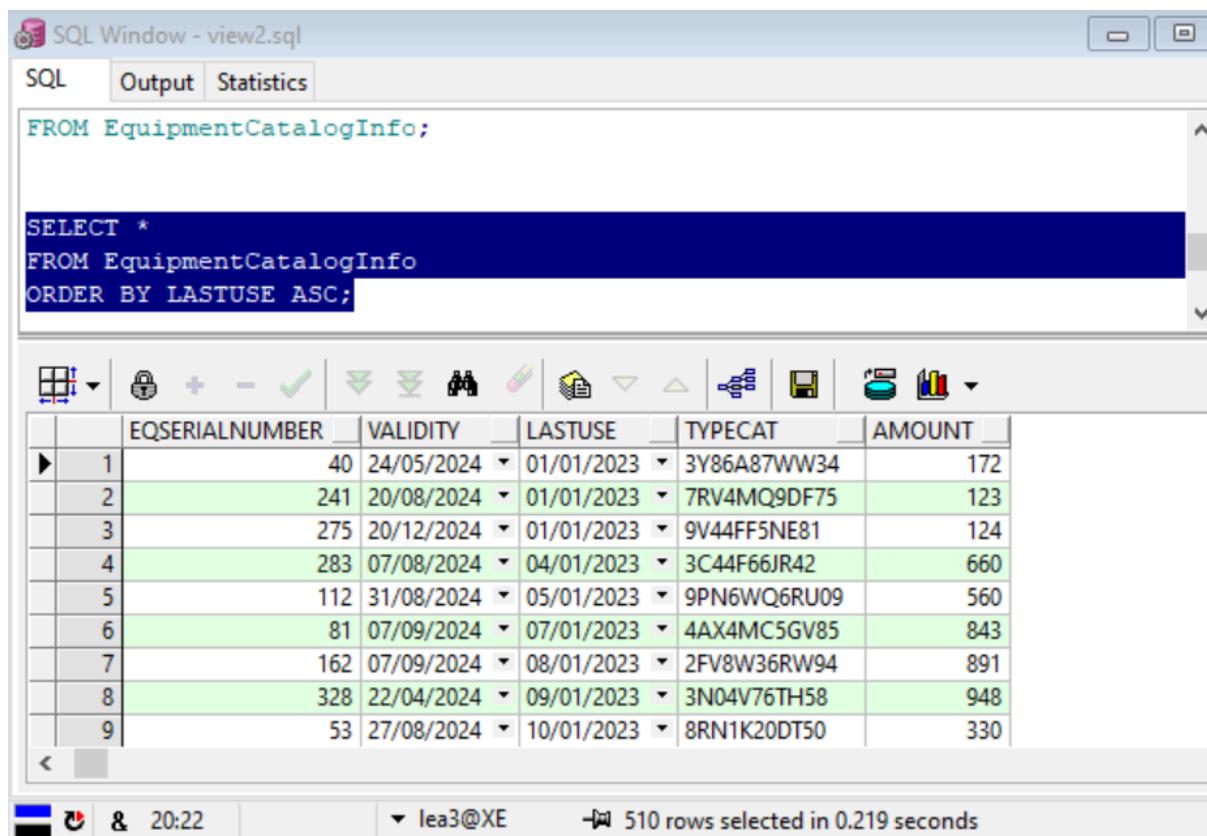
	EQSERIALNUMBER	VALIDITY	LASTUSE	TYPECAT	AMOUNT	
►	1	40	24/05/2024	01/01/2023	3Y86A87WW34	172
	2	241	20/08/2024	01/01/2023	7RV4MQ9DF75	123
	3	275	20/12/2024	01/01/2023	9V44FF5NE81	124
	4	283	07/08/2024	04/01/2023	3C44F66JR42	660
	5	112	31/08/2024	05/01/2023	9PN6WQ6RU09	560
	6	81	07/09/2024	07/01/2023	4AX4MC5GV85	843
	7	162	07/09/2024	08/01/2023	2FV8W36RW94	891
	8	328	22/04/2024	09/01/2023	3N04V76TH58	948
	9	53	27/08/2024	10/01/2023	8RN1K20DT50	330
	10	29	23/12/2024	10/01/2023	9V44FF5NE81	124
	11	126	06/11/2024	12/01/2023	5AM9AG8TX07	207
	12	291	24/08/2024	12/01/2023	6T50TX8AK21	87
	13	54	30/12/2024	13/01/2023	3U26Q14WX47	133
	14	385	06/03/2024	13/01/2023	5VF1YE1HV96	172
	15	475	19/02/2024	13/01/2023	8CF6HH1KE58	951

שאילתא 1 על מבט 2:
השאילתת `SELECT * FROM EquipmentCatalogInfo;` מציגה את כל הנתונים מה מבט `EquipmentCatalogInfo`, הcoliים פרטי ציוד וקטלוג כמו מס' סידורי, תוקף, שימוש אחרון, ועוד ערך הציון.



	EQSERIALNUMBER	VALIDITY	LASTUSE	TYPECAT	AMOUNT
► 1	1	08/07/2024	08/08/2023	5HT7QR2UT55	560
2	2	22/10/2024	29/11/2023	9YD2Q87VC50	737
3	3	17/11/2024	11/02/2023	1YM5YQ0HJ85	846
4	4	24/10/2024	22/07/2023	8NM1HG2TG72	602
5	5	04/11/2024	03/07/2023	6M46X13RA91	649
6	6	17/10/2024	13/05/2023	4G67AY8KP41	904
7	7	15/03/2024	17/12/2023	4A48MC6GX95	133

שאילתא 2 על מבט 2:
השאילתת `SELECT * FROM EquipmentCatalogInfo ORDER BY LASTUSE ASC;` מציגה את כל הנתונים מה מבט `EquipmentCatalogInfo`, וממיינת אותם בסדר עולה לפי עמודת `LASTUSE`.



	EQSERIALNUMBER	VALIDITY	LASTUSE	TYPECAT	AMOUNT
► 1	40	24/05/2024	01/01/2023	3Y86A87WW34	172
2	241	20/08/2024	01/01/2023	7RV4MQ9DF75	123
3	275	20/12/2024	01/01/2023	9V44FF5NE81	124
4	283	07/08/2024	04/01/2023	3C44F66JR42	660
5	112	31/08/2024	05/01/2023	9PN6WQ6RU09	560
6	81	07/09/2024	07/01/2023	4AX4MC5GV85	843
7	162	07/09/2024	08/01/2023	2FV8W36RW94	891
8	328	22/04/2024	09/01/2023	3N04V76TH58	948
9	53	27/08/2024	10/01/2023	8RN1K20DT50	330

20:22 & lea3@XE 510 rows selected in 0.219 seconds

שאילתה 1 על מבט1:

השאילתה `SELECT * FROM EmployeeTeams`; מציגה את כל הנתונים מה מבט `EmployeeTeams` הכוללים פרטי עובדים, שם הצוות שלהם, והשכר הממוצע בצוות.

	ID_EMPLOYEE	NAME_EMP	SALARY	NAME_OF_TEAM	AVG_TEAM_SALARY
▶	1	111	Paul	Yodo	9264
	2	112		Jetwire	2715
	3	121	Aiden	Jaxnation	13770.5
	4	140	Leah	Innojam	8742.5
	5	171	Alice	Rhyzero	8214.5
	6	200	Vera	Brightbean	1058
	7	300	Ben	Twitternation	5972
	8	951	Eli	Agimba	14392
	9	411	Ava	Voonder	5627

שאילתה 2 על מבט1:

השאילתה ``;SELECT * FROM EmployeeTeams WHERE avg_team_salary > 5000`` מציגה את כל הנתונים מה מבט `EmployeeTeams` רק עבור צוותים שבהם השכר הממוצע גבוה מ-5000.

	ID_EMPLOYEE	NAME_EMP	SALARY	NAME_OF_TEAM	AVG_TEAM_SALARY
▶	1	111	Paul	Yodo	9264
	2	121	Aiden	Jaxnation	13770.5
	3	140	Leah	Innojam	8742.5
	4	171	Alice	Rhyzero	8214.5
	5	300	Ben	Twitternation	5972
	6	951	Eli	Agimba	14392
	7	411	Ava	Voonder	5627
	8	965	Amy	Jaxnation	13770.5
	9	937	Eli	Blogpad	11283.5
	10	538	Amy	Ntaqs	13349

