

# מיני פרויקט בסיסי נתונים

מגישות:

הילה חג'בי: 213899412

מעין אלקיים: 212612972

## תוכן עניינים

- תיאור הארגון
- תרשים ERD
- תרשים DSD
- פירוט הישויות
- פירוט הקשרים בין הישויות
- טבלאות
- Script ליצירת טבלאות
- Script למחיקת טבלאות
- הכנסת נתונים ב 3 דרכים:
  - 1. קובץ txt
  - 2. data generator
  - 3. mockaroo
- גיבוי ושחזור נתונים

שם הארגון: איחוד הצלה.

תיאור הארגון:

איחוד הצלה הוא ארגון מתנדבים ישראלי שנוסד בשנת 2006 על מנת לאחד את מתנדבי "הצלה" ברחבי הארץ. הארגון מספק מענה רפואי ראשוני ומקצועי עד להגעת אמבולנס, ובימינו מונה מעל 6,500 מתנדבים פעילים מדין ועד אילת, פועל 24/7, גם בשבתות וחגים. המתנדבים מגישים סיוע רפואי חיוני ומגיעים מכל שכבות האוכלוסייה בישראל. ייעודו של הארגון הוא לטפל במקרים רפואיים ללא הבדל דת, גזע ומין, בתיאום עם גורמים פועלים להצלת חיים, ולהעניק טיפול רפואי ראשוני בתוך 90 שניות, על מנת להציל חיים ולמזער נזקים. מטרות הארגון כוללות הגשת סיוע רפואי בהתנדבות לכל נזקק בתוך 90 שניות, אפשרות למתנדבים לפעול לפי אורח חייהם, דתם ואמונתם, פיתוח יישומים וטכנולוגיות מתקדמות להצלת חיים, פעולה במקצועיות בכל תחומי העשייה, וציוד מקצועי ומתקדם למתנדבים. הפרויקט שלנו מתמקד בתחום הקורסים.

\* \* \*

ישויות:

1. משתתפים: Participants

- ת.ז של משתתף (מזהה) p\_id
- role - תפקיד של משתתף.
- p\_name - שם משתתף.
- gender - גבר/ אישה.
- p\_date - תאריך לידה של משתתף

2. קבוצה: StudentGroup

- ת.ז של קבוצה (מזהה) g\_id
- g\_hour - שעה בלוח זמנים שבה מתקיים הקורס של אותה קבוצה
- g\_day - יום בלוח זמנים שבה מתקיים הקורס של אותה קבוצה
- Max\_p - מספר מקסימלי של משתתפים.

3. חדר: Room

- ת.ז של חדר (מזהה) r\_id
- location - מיקום החדר שבו מתקיים הקורס.
- numplace - מס מקומות ישיבה בחדר.
- type - מעבדה / אולם / שטח.

4. ציוד רפואי: Equipment

- ת.ז של ציוד רפואי (מזהה) e\_id
- e\_id - ת.ז של ציוד
- e\_date - תאריך תפוגה של ציוד.
- amount - הכמות הכוללת של פריט הציוד הזמין במלאי.

5. קורסים : Courses

- ת.ז של קורס (מזהה) c\_id
- C\_name - שם קורס.
- categories - קטגוריה של קורס (חובשים/ מגיש עזרה ראשונה/ החייאה).
- pre-course - דרישות קדם לקורס הנוכחי.
- presence - נוכחות חובה בקורס (T/F).

6. מרצים : lecturers

- ת.ז של מרצה (מזהה) l\_id
- l\_name - שם של מרצה.
- l\_date - תאריך לידה.
- seniority - ותק של מרצה.
- training - הכשרה של מרצה.

קשרים:

1. StudentGroup Room (M)  $\Leftrightarrow$  (1)  
*Relation Name: Scheduled In*

2. StudentGroup (M)  $\Leftrightarrow$  (M) Participants  
*Relation Name: belongs*

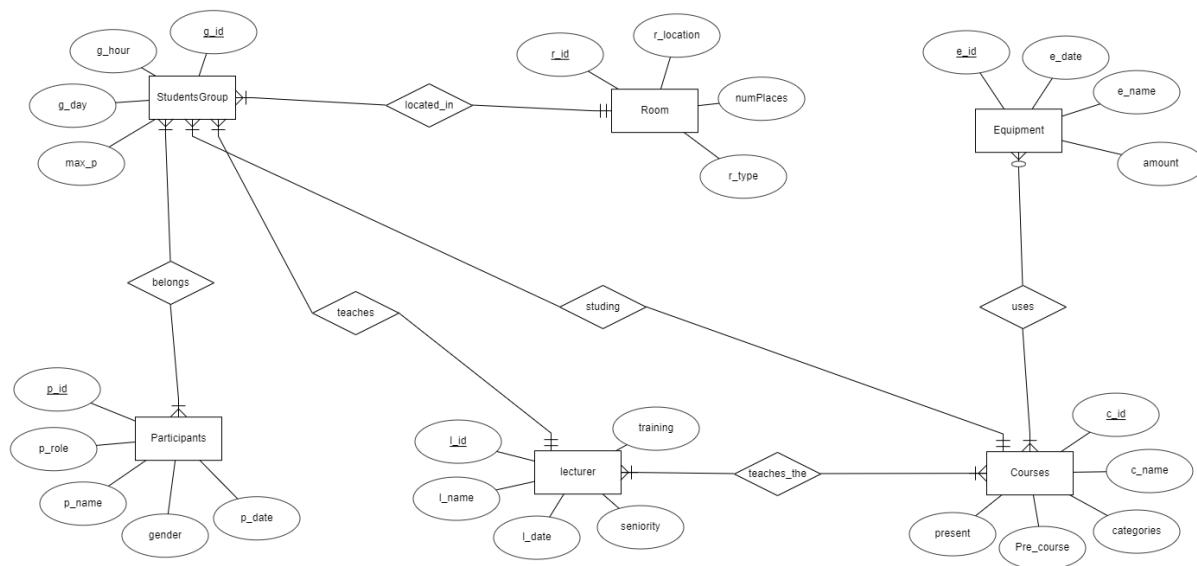
3. StudentGroup (1)  $\Leftrightarrow$  (M) lecturers  
*Relation Name: teaches*

4. StudentGroup (1)  $\Leftrightarrow$  (M) lecturers  
*Relation Name: teaches*

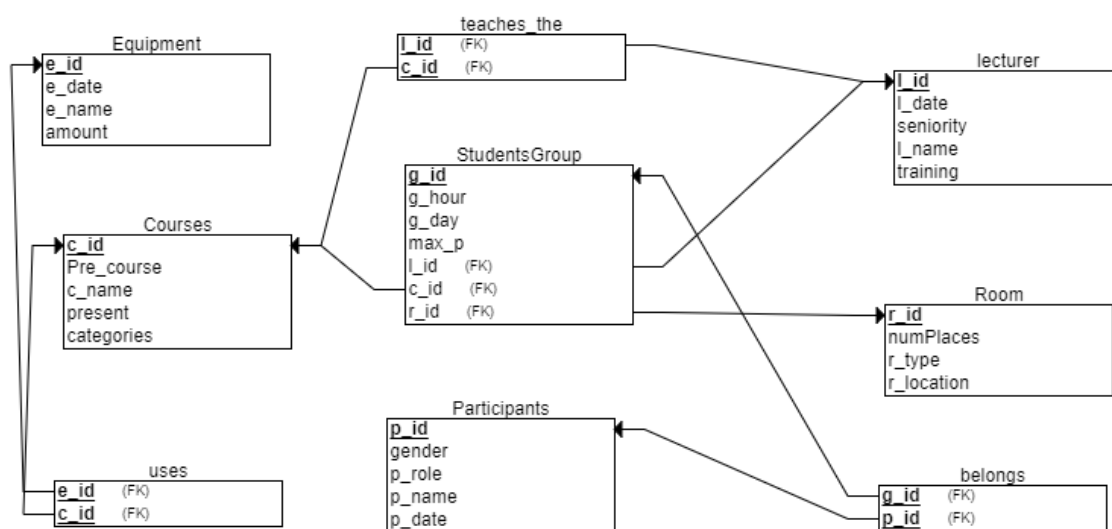
5. lecturers (M)  $\Leftrightarrow$  (M) Courses  
*Relation Name: teaches\_the*

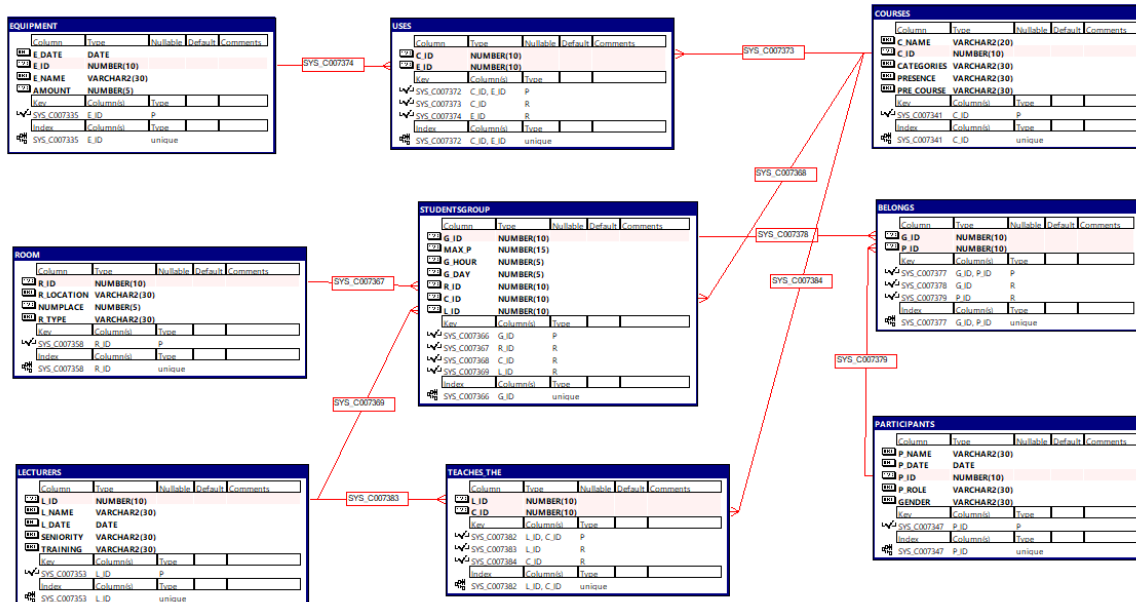
6. Courses (M)  $\Leftrightarrow$  (M) Equipment  
*Relation Name: teaches*

# תרשים ERD:



# תרשים DSD:





## יצירת הטבלאות:

CREATE TABLE Equipment

```
(
  e_date DATE NOT NULL,
  e_id NUMBER(10) NOT NULL,
  e_name VARCHAR2(30) NOT NULL,
  amount NUMBER(5) NOT NULL,
  PRIMARY KEY (e_id)
);
```

CREATE TABLE Courses

```
(
  C_name VARCHAR2(20) NOT NULL,
  c_id NUMBER(10) NOT NULL,
  categories VARCHAR2(30) NOT NULL,
  presence VARCHAR2(30) NOT NULL,
  pre_course VARCHAR2(30) NOT NULL,
  PRIMARY KEY (c_id)
);
```

CREATE TABLE Participants

```
(
  p_name VARCHAR2(30) NOT NULL,
  p_date DATE NOT NULL,
  p_id NUMBER(10) NOT NULL,
  p_role VARCHAR2(30) NOT NULL,
  gender VARCHAR2(30) NOT NULL,
  PRIMARY KEY (p_id)
);
```

CREATE TABLE lecturers

```
(
  l_id NUMBER(10) NOT NULL,
  l_name VARCHAR2(30) NOT NULL,
```

```

l_date DATE NOT NULL,
seniority VARCHAR2(30) NOT NULL,
training VARCHAR2(30) NOT NULL,
PRIMARY KEY (l_id)
);

CREATE TABLE Room
(
  r_id NUMBER(10) NOT NULL,
  r_location VARCHAR2(30) NOT NULL,
  numplace NUMBER(5) NOT NULL,
  r_type VARCHAR2(30) NOT NULL,
  PRIMARY KEY (r_id)
);

CREATE TABLE StudentsGroup
(
  g_id NUMBER(10) NOT NULL,
  Max_p NUMBER(15) NOT NULL,
  g_hour NUMBER(5) NOT NULL,
  g_day NUMBER(5) NOT NULL,
  r_id NUMBER(10) NOT NULL,
  c_id NUMBER(10) NOT NULL,
  l_id NUMBER(10) NOT NULL,
  PRIMARY KEY (g_id),
  FOREIGN KEY (r_id) REFERENCES Room(r_id),
  FOREIGN KEY (c_id) REFERENCES Courses(c_id),
  FOREIGN KEY (l_id) REFERENCES lecturers(l_id)
);

CREATE TABLE uses
(
  c_id NUMBER(10) NOT NULL,
  e_id NUMBER(10) NOT NULL,
  PRIMARY KEY (c_id, e_id),
  FOREIGN KEY (c_id) REFERENCES Courses(c_id),
  FOREIGN KEY (e_id) REFERENCES Equipment(e_id)
);

CREATE TABLE belongs
(
  g_id NUMBER(10) NOT NULL,
  p_id NUMBER(10) NOT NULL,
  PRIMARY KEY (g_id, p_id),
  FOREIGN KEY (g_id) REFERENCES StudentsGroup(g_id),
  FOREIGN KEY (p_id) REFERENCES Participants(p_id)
);

CREATE TABLE teaches_the
(
  l_id NUMBER(10) NOT NULL,
  c_id NUMBER(10) NOT NULL,
  PRIMARY KEY (l_id, c_id),
  FOREIGN KEY (l_id) REFERENCES lecturers(l_id),
  FOREIGN KEY (c_id) REFERENCES Courses(c_id)
);

```

## מחיקת הטבלאות:

```
drop table uses;
drop table teaches_the;
drop table belongs;
drop table StudentsGroup;
drop table Room;
drop table lecturers;
drop table Participants;
drop table Courses;
drop table Equipment;
```

## הכנסת נתונים לטבלה:

● פקודות insert:

```
--
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-01-15', 'YYYY-MM-DD'), 1,
'First Aid Kit', 50);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-02-20', 'YYYY-MM-DD'), 2,
'Stethoscope', 30);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-03-10', 'YYYY-MM-DD'), 3,
'Blood Pressure Monitor', 70);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-04-05', 'YYYY-MM-DD'), 4,
'Thermometer', 20);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-05-15', 'YYYY-MM-DD'), 5,
'Glucose Meter', 40);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-06-25', 'YYYY-MM-DD'), 6,
'Oxygen Tank', 25);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-07-30', 'YYYY-MM-DD'), 7,
'Defibrillator', 60);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-08-15', 'YYYY-MM-DD'), 8,
'Surgical Gloves', 350);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-09-10', 'YYYY-MM-DD'), 9,
'Face Mask', 500);
INSERT INTO Equipment (e_date, e_id, e_name, amount) VALUES (TO_DATE('2023-10-05', 'YYYY-MM-DD'), 10,
'CPR Manikin', 15);
--Courses
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Basic First Aid', 1, 'Medical',
'Required', 'None');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('CPR', 2, 'Medical', 'Required',
'Basic First Aid');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Advanced Life Support', 3,
'Medical', 'Required', 'CPR');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Trauma Care', 4, 'Medical',
'Required', 'Basic First Aid');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Pediatric Care', 5, 'Medical',
'Required', 'Basic First Aid');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Emergency Response', 6,
'Medical', 'Required', 'None');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Disaster Management', 7,
'Management', 'Optional', 'Emergency Response');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Health and Safety', 8, 'Safety',
'Required', 'None');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Wilderness First Aid', 9,
'Medical', 'Optional', 'Basic First Aid');
INSERT INTO Courses (C_name, c_id, categories, presence, pre_course) VALUES ('Infection Control', 10, 'Medical',
'Required', 'Health and Safety');
--Participants
```



```

INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('John Doe', TO_DATE('1990-01-01',
'YYYY-MM-DD'), 101, 'Paramedic', 'Male');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Jane Smith', TO_DATE('1992-02-15',
'YYYY-MM-DD'), 102, 'EMT', 'Female');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Alice Johnson', TO_DATE('1995-03-
20', 'YYYY-MM-DD'), 103, 'First Responder', 'Female');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Bob Brown', TO_DATE('1993-04-25',
'YYYY-MM-DD'), 104, 'Certified Nurse', 'Male');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Charlie Davis', TO_DATE('1988-05-30',
'YYYY-MM-DD'), 105, 'Paramedic', 'Non-binary');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Diana Evans', TO_DATE('1991-06-05',
'YYYY-MM-DD'), 106, 'First Responder', 'Female');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Edward Franklin', TO_DATE('1994-07-
10', 'YYYY-MM-DD'), 107, 'EMT', 'Male');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Fiona Green', TO_DATE('1989-08-15',
'YYYY-MM-DD'), 108, 'Certified Nurse', 'Female');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('George Harris', TO_DATE('1996-09-
20', 'YYYY-MM-DD'), 109, 'Paramedic', 'Male');
INSERT INTO Participants (p_name, p_date, p_id, p_role, gender) VALUES ('Helen Irvine', TO_DATE('1997-10-25',
'YYYY-MM-DD'), 110, 'First Responder', 'Female');

--lecturers
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (111, 'Dr. Emily White', TO_DATE('2022-
01-15', 'YYYY-MM-DD'), 'Senior', 'Medical');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (112, 'Prof. John Miller', TO_DATE('2021-
02-20', 'YYYY-MM-DD'), 'Expert', 'Trauma Care');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (113, 'Dr. Sarah Thompson',
TO_DATE('2020-03-10', 'YYYY-MM-DD'), 'Intermediate', 'CPR');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (114, 'Mr. Michael Brown',
TO_DATE('2019-04-05', 'YYYY-MM-DD'), 'Junior', 'First Aid');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (115, 'Mrs. Anna Wilson', TO_DATE('2018-
05-15', 'YYYY-MM-DD'), 'Senior', 'Emergency Response');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (116, 'Ms. Laura Davis', TO_DATE('2017-
06-25', 'YYYY-MM-DD'), 'Expert', 'Disaster Management');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (117, 'Dr. Kevin Clark', TO_DATE('2016-
07-30', 'YYYY-MM-DD'), 'Intermediate', 'Pediatric Care');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (118, 'Prof. Jessica Lewis',
TO_DATE('2015-08-15', 'YYYY-MM-DD'), 'Senior', 'Advanced Life Support');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (119, 'Mr. Brian Walker', TO_DATE('2014-
09-10', 'YYYY-MM-DD'), 'Junior', 'Health and Safety');
INSERT INTO lecturers (l_id, l_name, l_date, seniority, training) VALUES (120, 'Mrs. Nancy Scott', TO_DATE('2013-
10-05', 'YYYY-MM-DD'), 'Intermediate', 'Infection Control');

--Room
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (1, 'Building A, Floor 1', 50, 'Classroom');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (2, 'Building B, Floor 2', 40, 'Lecture Hall');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (3, 'Building C, Floor 3', 30, 'Training Room');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (4, 'Building A, Floor 2', 60, 'Auditorium');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (5, 'Building D, Floor 1', 35, 'Seminar Room');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (6, 'Building B, Floor 1', 45, 'Conference Room');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (7, 'Building C, Floor 2', 55, 'Workshop Room');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (8, 'Building D, Floor 2', 25, 'Computer Lab');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (9, 'Building A, Floor 3', 70, 'Training Center');
INSERT INTO Room (r_id, r_location, numplace, r_type) VALUES (10, 'Building B, Floor 3', 20, 'Discussion Room');

--StudentsGroup
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (1, 25, 9, 1, 1, 1, 114);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (2, 20, 10, 2, 2, 2, 113);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (3, 30, 11, 3, 3, 3, 118);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (4, 35, 13, 4, 4, 4, 112);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (5, 40, 14, 5, 5, 5, 117);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (6, 45, 15, 6, 6, 6, 115);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (7, 50, 9, 7, 7, 7, 116);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (8, 55, 10, 1, 8, 8, 119);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (9, 60, 11, 2, 9, 9, 111);
INSERT INTO StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) VALUES (10, 25, 13, 3, 10, 10, 120);

--uses
INSERT INTO uses (c_id, e_id) VALUES (1, 1);

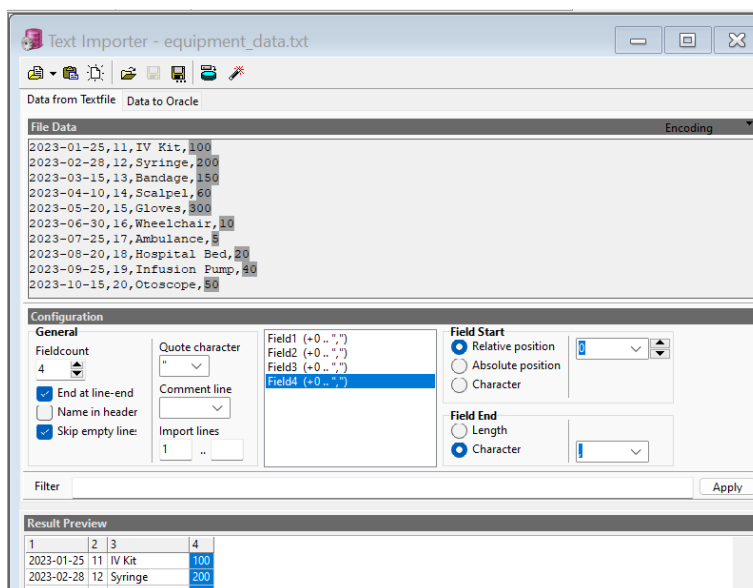
```

```

INSERT INTO uses (c_id, e_id) VALUES (2, 10);
INSERT INTO uses (c_id, e_id) VALUES (3, 7);
INSERT INTO uses (c_id, e_id) VALUES (4, 2);
INSERT INTO uses (c_id, e_id) VALUES (5, 5);
INSERT INTO uses (c_id, e_id) VALUES (6, 4);
INSERT INTO uses (c_id, e_id) VALUES (7, 6);
INSERT INTO uses (c_id, e_id) VALUES (8, 9);
INSERT INTO uses (c_id, e_id) VALUES (9, 1);
INSERT INTO uses (c_id, e_id) VALUES (10, 3);
--belongs
INSERT INTO belongs (g_id, p_id) VALUES (1, 101);
INSERT INTO belongs (g_id, p_id) VALUES (2, 102);
INSERT INTO belongs (g_id, p_id) VALUES (3, 103);
INSERT INTO belongs (g_id, p_id) VALUES (4, 104);
INSERT INTO belongs (g_id, p_id) VALUES (5, 105);
INSERT INTO belongs (g_id, p_id) VALUES (6, 106);
INSERT INTO belongs (g_id, p_id) VALUES (7, 107);
INSERT INTO belongs (g_id, p_id) VALUES (8, 108);
INSERT INTO belongs (g_id, p_id) VALUES (9, 109);
INSERT INTO belongs (g_id, p_id) VALUES (10, 110);
--teaches_the
INSERT INTO teaches_the (l_id, c_id) VALUES (111, 1);
INSERT INTO teaches_the (l_id, c_id) VALUES (112, 4);
INSERT INTO teaches_the (l_id, c_id) VALUES (113, 2);
INSERT INTO teaches_the (l_id, c_id) VALUES (114, 1);
INSERT INTO teaches_the (l_id, c_id) VALUES (115, 6);
INSERT INTO teaches_the (l_id, c_id) VALUES (116, 7);
INSERT INTO teaches_the (l_id, c_id) VALUES (117, 5);
INSERT INTO teaches_the (l_id, c_id) VALUES (118, 3);
INSERT INTO teaches_the (l_id, c_id) VALUES (119, 8);
INSERT INTO teaches_the (l_id, c_id) VALUES (120, 10);

```

- הכנסת נתונים ע"י קובץ txt:  
לטבלה Equipment



Text Importer - equipment\_data.txt

Data from Textfile Data to Oracle

General

Owner: EQUIPMENT

Table: EQUIPMENT

Commit every... 0

Overwrite duplicates

Ignore duplicates

Initializing Script

Finalizing Script

Fields

Field1 -> E\_DATE (DATE)

Field2 -> E\_ID (NUMBER)

Field3 -> E\_NAME (VARCHAR2)

Field4 -> AMOUNT (NUMBER)

Fieldtype: Number

SQL function

additional Oracle processing, for example: substr(, 1, 20)

Result Preview

1	2	3	4
2023-01-25	11	IV Kit	100
2023-02-28	12	Syringe	200

	E_DATE	E_ID	E_NAME	AMOUNT
1	15/01/2023	1	First Aid Kit	50
2	20/02/2023	2	Stethoscope	30
3	10/03/2023	3	Blood Pressure Monitor	70
4	05/04/2023	4	Thermometer	20
5	15/05/2023	5	Glucose Meter	40
6	25/06/2023	6	Oxygen Tank	25
7	30/07/2023	7	Defibrillator	60
8	15/08/2023	8	Surgical Gloves	350
9	10/09/2023	9	Face Mask	500
10	28/02/2023	12	Syringe	200
11	05/10/2023	10	CPR Manikin	15
12	25/01/2023	11	IV Kit	100
13	15/03/2023	13	Bandage	150
14	10/04/2023	14	Scalpel	60
15	20/05/2023	15	Gloves	300
16	30/06/2023	16	Wheelchair	10
17	25/07/2023	17	Ambulance	5
18	20/08/2023	18	Hospital Bed	20
19	25/09/2023	19	Infusion Pump	40
20	15/10/2023	20	Otoscope	50

לטבלה Courses:

Text Importer - courses\_data.txt

Data from Textfile Data to Oracle

File Data

Encoding

Basic Life Support,11,Medical,Required,None

First Aid Cert,12,Medical,Required,Basic Life Support

EM Technician,13,Medical,Required,First Aid Cert

Adv Trauma Life Sup,14,Medical,Required,Basic Life Support

Ped Adv Life Support,15,Medical,Required,Basic Life Support

Fire Safety Training,16,Safety,Required,None

Disaster Response,17,Management,Optional,Fire Safety Training

Occupational Health,18,Safety,Required,None

Remote Emerg Care,19,Medical,Optional,First Aid Cert

Epidemic Preparedness,20,Medical,Required,Occupational Health

Configuration

General

Fieldcount: 5

End at line-end

Name in header

Skip empty line

Quote character

Comment line

Import lines

Field1 (+0..,")

Field2 (+0..,")

Field3 (+0..,")

Field4 (+0..,")

Field5 (+0..,")

Field Start

Relative position

Absolute position

Character

Field End

Length

Character

Filter

Apply

Result Preview

1	2	3	4	5
Basic Life Support	11	Medical	Required	None
First Aid Cert	12	Medical	Required	Basic Life Support

Text Importer - courses\_data.txt

Data from Textfile Data to Oracle

General

Owner: COURSES Table: Clear Table

Commit every... 0 Overwrite duplicates Ignore duplicates

Initializing Script Finalizing Script

Fields

Field1 -> C\_NAME (VARCHAR2)  
Field2 -> C\_ID (NUMBER)  
Field3 -> CATEGORIES (VARCHAR2)  
Field4 -> PRESENCE (VARCHAR2)  
Field5 -> PRE\_COURSE (VARCHAR2)

Field: PRE\_COURSE (VARCH  
Fieldtype: String  
Create SQL  
SQL function: additional Oracle processing, for example: substr(%, 1, 20)

Result Preview

1	2	3	4	5
Basic Life Support	11	Medical	Required	None
First Aid Cert	12	Medical	Required	Basic Life Support

	C_NAME	C_ID	CATEGORIES	PRESENCE	PRE_COURSE
1	Basic First Aid	1	Medical	Required	None
2	CPR	2	Medical	Required	Basic First Aid
3	Advanced Life Support	3	Medical	Required	CPR
4	Trauma Care	4	Medical	Required	Basic First Aid
5	Pediatric Care	5	Medical	Required	Basic First Aid
6	Emergency Response	6	Medical	Required	None
7	Disaster Management	7	Management	Optional	Emergency Response
8	Health and Safety	8	Safety	Required	None
9	Wilderness First Aid	9	Medical	Optional	Basic First Aid
10	Infection Control	10	Medical	Required	Health and Safety
11	Basic Life Support	11	Medical	Required	None
12	First Aid Cert	12	Medical	Required	Basic Life Support
13	EM Technician	13	Medical	Required	First Aid Cert
14	Adv Trauma Life Sup	14	Medical	Required	Basic Life Support
15	Ped Adv Life Support	15	Medical	Required	Basic Life Support
16	Fire Safety Training	16	Safety	Required	None
17	Disaster Response	17	Management	Optional	Fire Safety Training
18	Occupational Health	18	Safety	Required	None
19	Remote Emerg Care	19	Medical	Optional	First Aid Cert
20	Epidemic Preparedness	20	Medical	Required	Occupational Health

- הכנסת נתונים ע"י mockaroo:  
לטבלה Room:

Field Name Type Options

r\_id Sequence start at: 11 step: 1 repeat: 1 restart at: 40 blank: 0 %

r\_location Formula - ("A".ord + random(0, 3)).chr + ", Floor " + random(1, 3).to blank: 0 %

numplace Number min: 30 max: 60 decimals: 0 blank: 0 %

r\_type Custom List Classroom, Lecture Hall, Training Room, Seminar Room, Comp sequential blank: 0 %

SQL Window - Room.sql

SQL Output Statistics

```

insert into Room (r_id, r_location, numplace, r_type) values (11, 'Building D, Floor 2', 49, 'Classroom');
insert into Room (r_id, r_location, numplace, r_type) values (12, 'Building C, Floor 3', 32, 'Lecture Hall');
insert into Room (r_id, r_location, numplace, r_type) values (13, 'Building D, Floor 3', 44, 'Training Room');
insert into Room (r_id, r_location, numplace, r_type) values (14, 'Building A, Floor 1', 47, 'Seminar Room');
insert into Room (r_id, r_location, numplace, r_type) values (15, 'Building D, Floor 3', 30, 'Computer Lab');
insert into Room (r_id, r_location, numplace, r_type) values (16, 'Building B, Floor 1', 48, 'Auditorium');
insert into Room (r_id, r_location, numplace, r_type) values (17, 'Building A, Floor 1', 43, 'Conference Room');
insert into Room (r_id, r_location, numplace, r_type) values (18, 'Building B, Floor 1', 32, 'Workshop Room');
insert into Room (r_id, r_location, numplace, r_type) values (19, 'Building D, Floor 3', 55, 'Training Center');
insert into Room (r_id, r_location, numplace, r_type) values (20, 'Building C, Floor 2', 57, 'Discussion Room');

```

	R_ID	R_LOCATION	NUMPLACE	R_TYPE
1	11	Building D, Floor 2	49	Classroom
2	12	Building C, Floor 3	32	Lecture Hall
3	13	Building D, Floor 3	44	Training Room
4	14	Building A, Floor 1	47	Seminar Room
5	1	Building A, Floor 1	50	Classroom
6	2	Building B, Floor 2	40	Lecture Hall
7	3	Building C, Floor 3	30	Training Room
8	4	Building A, Floor 2	60	Auditorium
9	5	Building D, Floor 1	35	Seminar Room
10	6	Building B, Floor 1	45	Conference Room
11	7	Building C, Floor 2	55	Workshop Room
12	8	Building D, Floor 2	25	Computer Lab
13	9	Building A, Floor 3	70	Training Center
14	10	Building B, Floor 3	20	Discussion Room
15	15	Building D, Floor 3	30	Computer Lab
16	16	Building B, Floor 1	48	Auditorium
17	17	Building A, Floor 1	43	Conference Room
18	18	Building B, Floor 1	32	Workshop Room
19	19	Building D, Floor 3	55	Training Center
20	20	Building C, Floor 2	57	Discussion Room

## StudentsGroup: לטבלה

Field Name	Type	Options
g_id	Sequence	start at: 11 step: 1 repeat: 1 restart at: 40 blank: 0 %
Max_p	Number	min: 30 max: 60 decimals: 0 blank: 0 %
g_hour	Number	min: 8 max: 20 decimals: 0 blank: 0 %
g_day	Number	min: 1 max: 5 decimals: 0 blank: 0 %
r_id	Sequence	start at: 1 step: 1 repeat: 1 restart at: 40 blank: 0 %
c_id	Sequence	start at: 1 step: 1 repeat: 1 restart at: 40 blank: 0 %
l_id	Sequence	start at: 120 step: -1 repeat: 1 restart at: 40 blank: 0 %

```

SQL Window - StudentsGroup.sql
SQL Output Statistics
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (11, 51, 18, 1, 1, 1, 120);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (12, 46, 11, 5, 2, 2, 119);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (13, 35, 9, 2, 3, 3, 118);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (14, 30, 17, 4, 4, 4, 117);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (15, 46, 9, 3, 5, 5, 116);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (16, 52, 11, 4, 6, 6, 115);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (17, 47, 8, 5, 7, 7, 114);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (18, 31, 16, 3, 8, 8, 113);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (19, 40, 15, 3, 9, 9, 112);
insert into StudentsGroup (g_id, Max_p, g_hour, g_day, r_id, c_id, l_id) values (20, 36, 11, 5, 10, 10, 111);

```

	G_ID	MAX_P	G_HOUR	G_DAY	R_ID	C_ID	L_ID
1	11	51	18	1	1	1	120
2	12	46	11	5	2	2	119
3	13	35	9	2	3	3	118
4	14	30	17	4	4	4	117
5	15	46	9	3	5	5	116
6	16	52	11	4	6	6	115
7	17	47	8	5	7	7	114
8	18	31	16	3	8	8	113
9	19	40	15	3	9	9	112
10	20	36	11	5	10	10	111
11	1	25	9	1	1	1	114
12	2	20	10	2	2	2	113
13	3	30	11	3	3	3	118
14	4	35	13	4	4	4	112
15	5	40	14	5	5	5	117
16	6	45	15	6	6	6	115
17	7	50	9	7	7	7	116
18	8	55	10	1	8	8	119
19	9	60	11	2	9	9	111
20	10	25	13	3	10	10	120

• הכנסת נתונים ע"י data generator:  
לטבלה Lecturers:

Data Generator - New

LECTURERS

Owner: SYS Table: LECTURERS Number of records: 400

Name	Type	Size	Data	Master
L_ID	NUMBER	10	Sequence(100,1)	...
L_NAME	VARCHAR2	30	FirstName + LastName	...
L_DATE	DATE	...	Random(01/01/1970, 01/01/2000)	...
SENIORITY	VARCHAR2	30	Random(1, 30)	...
TRAINING	VARCHAR2	30	List('Infection Control', 'Health and Safety', 'Advanced Life Support', 'Pediatric Care', 'Disaster Management', 'Emergency Response', 'First Aid', 'CPF ...	...

Definition Options Result

	L_ID	L_NAME	L_DATE	SENIORITY	TRAINING
1	398	LiquidShannon	26/02/1996	4	Advanced Life Support
2	399	SarahDouglas	08/02/1992	13	Health and Safety
3	400	MiriamGoodman	30/08/1981	22	Health and Safety
4	401	LynetteMcDonald	08/01/1977	21	Medical
5	402	SalmaTheron	08/12/1976	11	Infection Control
6	403	HarrisonQuaid	04/03/1973	20	Infection Control
7	404	GeorgeLofgren	05/01/1993	29	CPR
8	405	KimberlyFuray	21/12/1996	10	Pediatric Care"Disaster Manag
9	406	JuanCromwell	06/10/1978	15	CPR
10	407	JeanLonsdale	14/12/1989	27	Advanced Life Support
11	408	KayJanssen	22/03/1997	22	Health and Safety
12	409	SaraForrest	02/12/1977	8	Infection Control
13	410	NikkaHudson	04/05/1979	4	Infection Control
14	411	GlenStallone	17/02/1998	8	Advanced Life Support
15	412	AndreWeston	01/02/1972	26	First Aid
16	413	SallyBachman	10/08/1975	30	Trauma Care
17	414	LorraineOrton	01/12/1982	19	Health and Safety
18	415	FisherBurton	04/02/1980	19	Advanced Life Support

לטבלה Participants:

Data Generator - participants.gd

PARTICIPANTS

Owner: SYS Table: PARTICIPANTS Number of records: 400

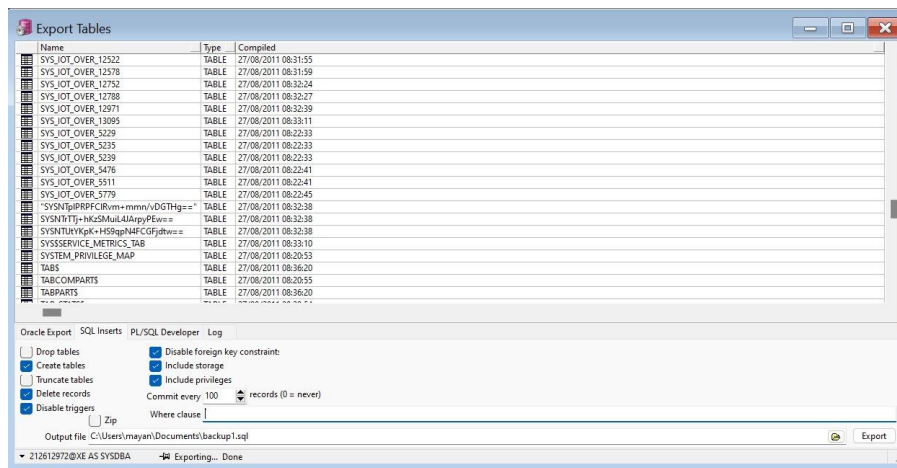
Name	Type	Size	Data	Master
P_NAME	VARCHAR2	30	FirstName + LastName	...
P_DATE	DATE	...	Random(01/01/1970, 01/01/2000)	...
P_ID	NUMBER	10	Sequence(500,1)	...
P_ROLE	VARCHAR2	30	List('First Responder', 'Paramedic', 'Certified Nurse', 'First Responder', 'EMT')	...
GENDER	VARCHAR2	30	List(female, male)	...

Definition Options Result

	P_NAME	P_DATE	P_ID	P_ROLE	GENDER
1	FrancoEsposito	12/06/1973	800	First Responder	female
2	EmmArkenstone	08/10/1992	801	EMT	female
3	PercyPigott-Smith	30/07/1986	802	First Responder	female
4	MarkFisher	28/02/1974	803	Certified Nurse	female
5	MollyRooker	12/05/1983	804	Paramedic	male
6	JohnnyTwilley	03/10/1978	805	Paramedic	female
7	MikaStone	15/07/1973	806	First Responder	female
8	JodyMoriarty	10/11/1986	807	EMT	female
9	DelbertFrancis	12/01/1974	808	First Responder	female
10	EdwardBall	01/01/1972	809	First Responder	male
11	LeVarCurfman	01/01/1974	810	Certified Nurse	female
12	AlessandroCreek	05/11/1994	811	EMT	female
13	WilliamPfeiffer	17/10/1997	812	Certified Nurse	male
14	Maefavreau	24/06/1993	813	EMT	male
15	WadeFlack	21/08/1987	814	Paramedic	male
16	QueenValentin	06/05/1974	815	Certified Nurse	female
17	MillieSummer	27/12/1991	816	Certified Nurse	female
18	TimothyVinton	05/03/1991	817	EMT	male
19	SheenaBalk	19/08/1988	818	First Responder	female
20	JaneaneSchneider	21/03/1990	819	First Responder	male
21	FrankieTravolta	01/10/1982	820	EMT	male
22	MauraTwilley	23/10/1975	821	Paramedic	male
23	PeteStormare	13/01/1994	822	First Responder	male
24	LaurenDomino	26/08/1992	823	First Responder	female
25	CaseyGoldwyn	09/10/1972	824	First Responder	female
26	NickelFonda	19/03/1997	825	First Responder	male
27	RoddyKirkwood	01/10/1996	826	First Responder	female
28	ChuckDean	13/09/1971	827	EMT	male
29	MarlonGershon	30/10/1988	828	First Responder	male

## גיבוי ושחזור נתונים

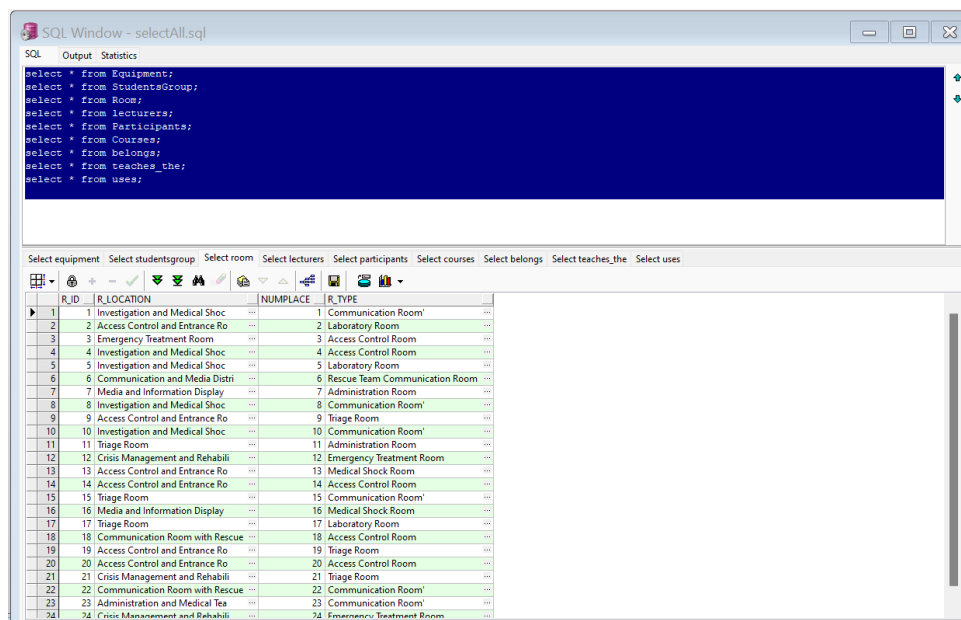
### ביצוע גיבוי במחשב אחד:



### שיחזור על מחשב השני:



### וכעת כל הטבלאות מלאות:



## שלב 2-שאלות

### שינויים:

הוספנו שדה בשם age לטבלה participants:

```
ALTER TABLE Participants
ADD age NUMBER;
```

```
select * from Participants;
select * from Courses;
select * from belongs;
select * from teaches_the;
select * from uses;
```

	P_NAME	P_DATE	P_ID	P_ROLE	GENDER	AGE
1	Antonio Satriani	20/06/1972	500	Paramedic	female	
2	Percy Bonham	18/07/1976	501	First Responder	male	
3	James Hall	30/06/1981	502	Paramedic	female	
4	Nicole Pleasure	08/11/1993	503	Certified Nurse	male	
5	Sean Applegate	30/05/1985	504	Paramedic	male	
6	Kate Head	25/05/1979	505	First Responder	female	
7	Emmylou Orlando	23/01/1976	506	First Responder	female	
8	Maria Giamatti	05/01/1972	507	Certified Nurse	male	
9	Jake Krumholtz	15/02/1979	508	EMT	female	
10	Avenned Gambon	18/06/1986	509	Certified Nurse	female	

```
UPDATE Participants SET age = EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM p_date);
```

```
select * from Participants;
select * from Courses;
select * from belongs;
select * from teaches_the;
select * from uses;
```

	P_NAME	P_DATE	P_ID	P_ROLE	GENDER	AGE
1	Antonio Satriani	20/06/1972	500	Paramedic	female	52
2	Percy Bonham	18/07/1976	501	First Responder	male	48
3	James Hall	30/06/1981	502	Paramedic	female	43
4	Nicole Pleasure	08/11/1993	503	Certified Nurse	male	31
5	Sean Applegate	30/05/1985	504	Paramedic	male	39
6	Kate Head	25/05/1979	505	First Responder	female	45
7	Emmylou Orlando	23/01/1976	506	First Responder	female	48
8	Maria Giamatti	05/01/1972	507	Certified Nurse	male	52
9	Jake Krumholtz	15/02/1979	508	EMT	female	45
10	Avenned Gambon	18/06/1986	509	Certified Nurse	female	38



## שאלות SELECT:

השאלתה נותנת מידע על הקבוצות של קורסים ששייכים לקטגוריות Medical, Emergency, Rescue כולל שם הקורס, קטגוריית הקורס, מספר קבוצה, שם המרצה, ותק המרצה, מיקום החדר סוג החדר ומספר המשתתפים בכל קבוצה.

התוצאות מסודרות לפי שם הקורס ומספר קבוצה.

```

SELECT
  c.C_name AS course_name,
  c.categories AS course_category,
  sg.g_id AS group_id,
  l.l_name AS lecturer_name,
  l.seniority AS lecturer_seniority,
  r.r_location AS room_location,
  r.r_type AS room_type,
  COUNT(b.p_id) AS num_participants
FROM
  StudentsGroup sg
JOIN
  Courses c ON sg.c_id = c.c_id
JOIN
  lecturers l ON sg.l_id = l.l_id
JOIN
  Room r ON sg.r_id = r.r_id
LEFT JOIN
  belongs b ON sg.g_id = b.g_id
WHERE
  c.categories IN ('Medical', 'Rescue', 'Emergency')
GROUP BY
  c.C_name, c.categories, sg.g_id, l.l_name, l.seniority,
  r.r_location, r.r_type
ORDER BY
  c.C_name, sg.g_id;

```

	COURSE_NAME	COURSE_CATEGORY	GROUP_ID	LECTURER_NAME	LECTURER_SENIORITY	ROOM_LOCATION	ROOM_TYPE	NUM_PARTICIPANTS
1	Advanced Life Support	Medical	48	Anne Atkins	22	Building B Floor 5	Discussion Room	4
2	Advanced Life Support	Medical	70	Rolando Weisz	17	Building D Floor 1	Auditorium	0
3	Advanced Life Support	Medical	72	Keith Pantoliano	18	Building A Floor 4	Computer Lab	1
4	Advanced Life Support	Medical	128	Jason Emmerich	16	Building D Floor 2	Computer Lab	1
5	Advanced Life Support	Medical	137	Kevin DiCaprio	7	Building D Floor 2	Computer Lab	0
6	Advanced Life Support	Medical	149	Crispin Stamp	17	Building B Floor 5	Auditorium	0
7	Advanced Life Support	Medical	187	Vonda Alexander	19	Building D Floor 2	Auditorium	1
8	Advanced Life Support	Medical	279	Harrison Broderick	4	Building E Floor 4	Classroom	1
9	Advanced Life Support	Medical	359	Art Farrell	9	Building C Floor 4	Auditorium	0
10	Basic First Aid	Medical	25	Joy Cruz	1	Building E Floor 1	Classroom	1

השאלתה מחזירה את גיל המשתתפים המינימלי והמקסימלי בקורס CPR :

```

SELECT
  c.C_name AS course_name,
  MIN(p.age) AS min_age,
  MAX(p.age) AS max_age
FROM
  Participants p
JOIN
  belongs b ON p.p_id = b.p_id
JOIN
  StudentsGroup sg ON b.g_id = sg.g_id
JOIN
  Courses c ON sg.c_id = c.c_id
WHERE
  c.C_name = 'CPR'
GROUP BY
  c.C_name;

```

	COURSE_NAME	MIN_AGE	MAX_AGE
1	CPR	25	54

השאלתה מחזירה את הציוד שמשתמשים בו הכי הרבה:

```

SELECT
    e.e_name AS equipment_name,
    u.course_usage_count
FROM
    Equipment e
JOIN
    (
        SELECT
            u.e_id,
            COUNT(u.c_Id) AS course_usage_count
        FROM
            uses u
        GROUP BY
            u.e_id
    ) u ON e.e_id = u.e_id
WHERE
    u.course_usage_count = (
        SELECT
            MAX(u2.course_usage_count)
        FROM
            (
                SELECT
                    u2.e_id,
                    COUNT(u2.c_Id) AS course_usage_count
                FROM
                    uses u2
                GROUP BY
                    u2.e_id
            ) u2
    );

```

	EQUIPMENT_NAME	COURSE_USAGE_COUNT
1	First Aid Kit	5

השאלתה נותנת מידע על כל קורס בטבלת Courses כולל שם הקורס, הקטגוריה שלו והקורס המקדים שלו. היא מחשבת גם את מספר המרצים השונים שמלמדים את הקורס ואת סך כל המשתתפים המקסימליים בקורס.

```

SELECT c.C_name, c.categories, c.pre_course,
    (SELECT COUNT(DISTINCT l.l_name)
     FROM teaches_the tt
     JOIN lecturers l ON tt.l_id = l.l_id
     WHERE tt.c_id = c.c_id) AS num_lecturers,
    (SELECT SUM(sg.Max_p)
     FROM StudentsGroup sg
     WHERE sg.c_id = c.c_id) AS total_participants,
    (SELECT r.r_location
     FROM StudentsGroup sg
     JOIN Room r ON sg.r_id = r.r_id
     WHERE sg.c_id = c.c_id
     AND ROWNUM = 1) AS room_location
FROM Courses c;

```

	C_NAME	CATEGORIES	PRE_COURSE	NUM_LECTURERS	TOTAL_PARTICIPANTS	ROOM_LOCATION
1	Disaster Management	Medical	Basic First Aid	3	91	Building C Floor 5
2	Trauma Care	Management	Basic First Aid	1	103	Building B Floor 5
3	Emergency Response	Safety	CPR	2	69	Building A Floor 3
4	Trauma Care	Management	Basic First Aid	0	43	Building B Floor 5
5	Emergency Response	Management	None	1	80	Building D Floor 2
6	Disaster Management	Medical	Basic First Aid	1		
7	Wilderness First Aid	Medical	Emergency Response	1		
8	CPR	Safety	Basic First Aid	0	84	Building A Floor 5
9	Wilderness First Aid	Safety	CPR	0		
10	Pediatric Care	Medical	Health and Safety	1		
11	Wilderness First Aid	Safety	None	1	32	Building A Floor 4
12	Infection Control	Management	None	0		
13	Pediatric Care	Medical	Emergency Response	1	205	Building E Floor 4

## שאלות DELETE:

השאלה מוחקת משתתפים מטבלת belongs משתתפים בקורסים בקטגוריית medical וגילם מעל גיל 50:

```
DELETE FROM belongs
WHERE p_id IN (
  SELECT b.p_id
  FROM belongs b
  JOIN Participants p ON b.p_id = p.p_id
  JOIN StudentsGroup sg ON b.g_id = sg.g_id
  JOIN Courses c ON sg.c_id = c.c_id
  WHERE c.categories = 'Management'
  AND p.age > 50
  GROUP BY b.p_id
  HAVING COUNT(*) > 0
);
```

שאלת ה select לפני הרצת שאלת המחיקה:

```
SELECT b.p_id
FROM belongs b
JOIN Participants p ON b.p_id = p.p_id
JOIN StudentsGroup sg ON b.g_id = sg.g_id
JOIN Courses c ON sg.c_id = c.c_id
WHERE c.categories = 'Management'
AND p.age > 50
GROUP BY b.p_id
HAVING COUNT(*) > 0;
```

	P_ID
1	815
2	554
3	759
4	696
5	536

לאחר ההרצה:

```
SELECT b.p_id
FROM belongs b
JOIN Participants p ON b.p_id = p.p_id
JOIN StudentsGroup sg ON b.g_id = sg.g_id
JOIN Courses c ON sg.c_id = c.c_id
WHERE c.categories = 'Management'
AND p.age > 50
GROUP BY b.p_id
HAVING COUNT(*) > 0;
```

	P_ID
--	------

השאלתה מוחקת ציוד מהטבלה uses שתאריך הציוד קטן מ-01/01/2025 ושם הציוד הוא Glucose Meter, Defibrillator, Oxygen Tank:

```
DELETE FROM uses
WHERE e_id IN (
    SELECT e.e_id
    FROM Equipment e
    WHERE e.e_date < TO_DATE('01/01/2025', 'DD/MM/YYYY')
    AND e.e_name IN ('Glucose Meter', 'Oxygen Tank', 'Defibrillator')
);
```

הרצת שאלת select לפני הרצת שאלת המחיקה:

```
SELECT *
FROM uses u
JOIN Equipment e ON u.e_id = e.e_id
WHERE e.e_date < TO_DATE('01/01/2025', 'DD/MM/YYYY')
AND e.e_name IN ('Glucose Meter', 'Oxygen Tank', 'Defibrillator');
```

	C_ID	E_ID	E_DATE	E_ID	E_NAME	AMOUNT
1	16	141	04/01/2024	141	Defibrillator	92
2	111	22	16/11/2024	22	Glucose Meter	99
3	201	164	18/07/2024	164	Glucose Meter	86
4	215	164	18/07/2024	164	Glucose Meter	86
5	236	5	17/10/2024	5	Oxygen Tank	43
6	276	5	17/10/2024	5	Oxygen Tank	43

לאחר ההרצה:

```
SELECT *
FROM uses u
JOIN Equipment e ON u.e_id = e.e_id
WHERE e.e_date < TO_DATE('01/01/2025', 'DD/MM/YYYY')
AND e.e_name IN ('Glucose Meter', 'Oxygen Tank', 'Defibrillator');
```

	C_ID	E_ID	E_DATE	E_ID	E_NAME	AMOUNT
--	------	------	--------	------	--------	--------

### שאלות UPDATE:

השאלתה הזו מבצעת עדכון מוסיפה עוד 40 לציוד מסוג 'CPR Manikin' וגם תאריך התפוגה שלהם יהיה ב-2026

```
UPDATE Equipment
SET amount = amount + 40
WHERE e_name = 'CPR Manikin'
AND e_id IN (
    SELECT e_id
    FROM Equipment
    WHERE EXTRACT(YEAR FROM e_date) = 2026
);
```

הרצת שאלת select לפני הרצת שאלת העדכון:

```
SELECT e_id, e_name, amount
FROM Equipment
WHERE e_name = 'CPR Manikin'
AND EXTRACT(YEAR FROM e_date) = 2026;
```

	E_ID	E_NAME	AMOUNT
1	73	CPR Manikin	70
2	194	CPR Manikin	87

לאחר ההרצה:

```
SELECT e_id, e_name, amount
FROM Equipment
WHERE e_name = 'CPR Manikin'
AND EXTRACT(YEAR FROM e_date) = 2026;
```

	E_ID	E_NAME	AMOUNT
▶ 1	73	CPR Manikin ...	110
2	194	CPR Manikin ...	127

השאלתה מעדכנת את מרצים שהם נהיים בגיל 40 השנה ומלמדים קורסים שנדרש נוכחות חובה, ומעדכן את הותק שלהם ע"י הוספת שנה:

```
UPDATE lecturers l
SET l.seniority = l.seniority + 1
WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM l.l_date) = 40
AND EXISTS (
    SELECT 1
    FROM teaches_the tt
    JOIN Courses c ON tt.c_Id = c.c_Id
    WHERE tt.l_Id = l.l_Id
    AND c.presence = 'Required'
);
```

הרצת שאלת select לפני הרצת שאלת העדכון:

```
SELECT *
FROM lecturers l
WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM l.l_date) = 40
AND EXISTS (
    SELECT 1
    FROM teaches_the tt
    JOIN Courses c ON tt.c_Id = c.c_Id
    WHERE tt.l_Id = l.l_Id
    AND c.presence = 'Required'
);
```

	L_ID	L_NAME	L_DATE	SENIORITY	TRAINING
▶ 1	107	Taye Bloch	10/06/1984	29	First Aid
2	248	Pamela Knight	26/10/1984	7	Medical
3	294	Todd MacLachlan	24/06/1984	4	Advanced Life Support
4	418	Loretta Postlethwaite	17/06/1984	4	Trauma Care
5	497	Marina Gooding	27/11/1984	13	Health and Safety

לאחר ההרצה:

```
SELECT *
FROM lecturers l
WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM l.l_date) = 40
AND EXISTS (
    SELECT 1
    FROM teaches_the tt
    JOIN Courses c ON tt.c_Id = c.c_Id
    WHERE tt.l_Id = l.l_Id
    AND c.presence = 'Required'
);
```

	L_ID	L_NAME	L_DATE	SENIORITY	TRAINING
▶ 1	107	Taye Bloch	10/06/1984	30	First Aid
2	248	Pamela Knight	26/10/1984	8	Medical
3	294	Todd MacLachlan	24/06/1984	5	Advanced Life Support
4	418	Loretta Postlethwaite	17/06/1984	5	Trauma Care
5	497	Marina Gooding	27/11/1984	14	Health and Safety

## שאלות עם פרמטרים:

השאלתה מחזירה רשימת משתתפים בקורס מסוים יחד עם שם המרצה שמלמד את הקורס ומיקום החדר שבו מתקיים הקורס עבור שם של הקורס כפרמטר:

```

SELECT
    p.p_name AS ParticipantName,
    l.l_name AS LecturerName,
    c.C_name AS CourseName,
    r.r_location AS RoomLocation
FROM
    Participants p
JOIN belongs b ON p.p_id = b.p_id
JOIN StudentsGroup sg ON b.g_id = sg.g_id
JOIN Courses c ON sg.c_id = c.c_id
JOIN lecturers l ON sg.l_id = l.l_id
LEFT JOIN Room r ON sg.r_id = r.r_id
WHERE c.C_name = <name="CourseName" type="string" list="select C_name from Courses">
order by p.p_name;

```

	PARTICIPANTNAME	LECTURERNAME	COURSENAME	ROOMLOCATION
1	Alannah McGowan	Leon Laurie	CPR	Building B Floor 3
2	Amy Tate	Leon Laurie	CPR	Building B Floor 3
3	Angelina Howard	Halle Allan	CPR	Building D Floor 5
4	Avenged Gambon	Azucar McBride	CPR	Building D Floor 2
5	Boz Tilly	Debbie Mathis	CPR	Building C Floor 4
6	Carlos Vaughn	Javon Porter	CPR	Building B Floor 2
7	Christopher Adler	Sissy Levin	CPR	Building D Floor 2
8	Clive Levine	Mira Shaw	CPR	Building A Floor 4
9	Courtney Sinise	Jean-Luc Atkinson	CPR	Building B Floor 5
10	Davis Eat World	Mira Blaine	CPR	Building B Floor 3
11	Eric Carrere	Javon Porter	CPR	Building B Floor 2

Variables

Name	Value
CourseName	CPR

Trauma Care

Emergency Response

Disaster Management

Wilderness First Aid

CPR

Wilderness First Aid

Pediatric Care

Wilderness First Aid

Infection Control

Pediatric Care

OK

השאלתה מחזירה רשימת קורסים שמשתמשים בצידוד מסוים עבור שם של צידוד כפרמטר:

```

SELECT
    c.C_name AS CourseName,
    c.c_id AS CourseID,
    e.e_name AS EquipmentName,
    e.e_id AS EquipmentID
FROM
    Courses c
JOIN uses u ON c.c_id = u.c_id
JOIN Equipment e ON u.e_id = e.e_id
WHERE
    e.e_name = <name=EquipmentName type="string">
ORDER BY
    c.C_name;

```

	COURSENAME	COURSEID	EQUIPMENTNAME	EQUIPMENTID
1	Advanced Life Support	343	CPR Manikin	218
2	Advanced Life Support	312	CPR Manikin	262
3	Advanced Life Support	399	CPR Manikin	149
4	Advanced Life Support	291	CPR Manikin	346
5	Advanced Life Support	318	CPR Manikin	373
6	Basic First Aid	358	CPR Manikin	312
7	Basic First Aid	196	CPR Manikin	344
8	Basic First Aid	320	CPR Manikin	396
9	CPR	349	CPR Manikin	32
10	CPR	31	CPR Manikin	262

Name	Value
EquipmentName	CPR Manikin

OK Cancel Clear

השאלתה מחזירה רשימת קורסים שמשתתף מסוים משתתף בהם עבור שם של משתתף כפרמטר:

```
SELECT
    p.p_name AS ParticipantName,
    c.C_name AS CourseName
FROM
    Participants p
JOIN belongs b ON p.p_id = b.p_id
JOIN StudentsGroup sg ON b.g_id = sg.g_id
JOIN Courses c ON sg.c_id = c.c_id
WHERE
    p.p_name = &<name="ParticipantName" type="string" required="true">
ORDER BY
    c.C_name;
```

	PARTICIPANTNAME	COURSENAME
1	Avenged Gambon	CPR
2	Avenged Gambon	Disaster Management

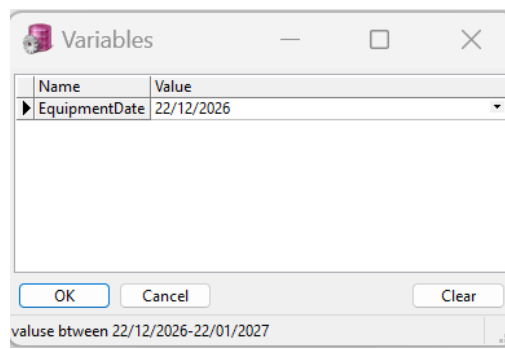
Name	Value
ParticipantName	Avenged Gambon

OK Cancel Clear

השאלתה מחזירה את הציוד שתאריך התפוגה שלו הוא התאריך שהתקבל כפרמטר:

```
SELECT
    e.e_id AS EquipmentID,
    e.e_name AS EquipmentName,
    e.e_date AS EquipmentDate,
    e.amount AS Amount
FROM
    Equipment e
WHERE
    e.e_date = &<name="EquipmentDate" type="date" hint="valuse btween 22/12/2026-22/01/2027">
ORDER BY
    e.e_name;
```

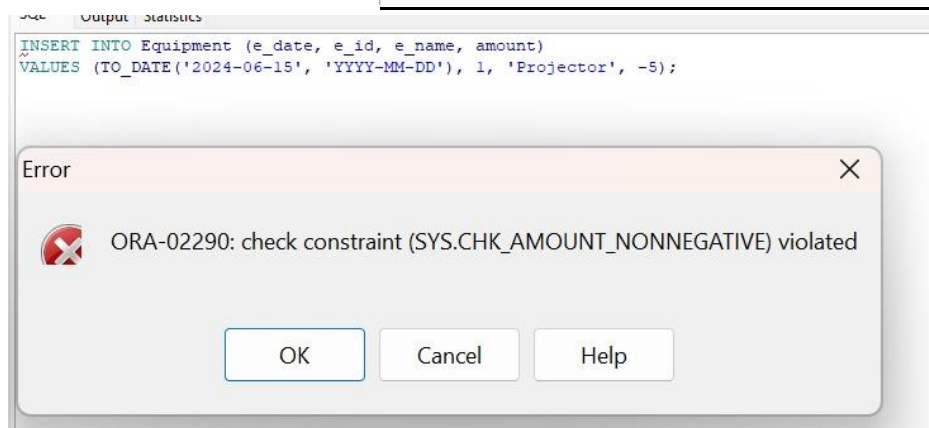
	EQUIPMENTID	EQUIPMENTNAME	EQUIPMENTDATE	AMOUNT
1	19	Defibrillator	22/12/2026	95



## אילוצים:

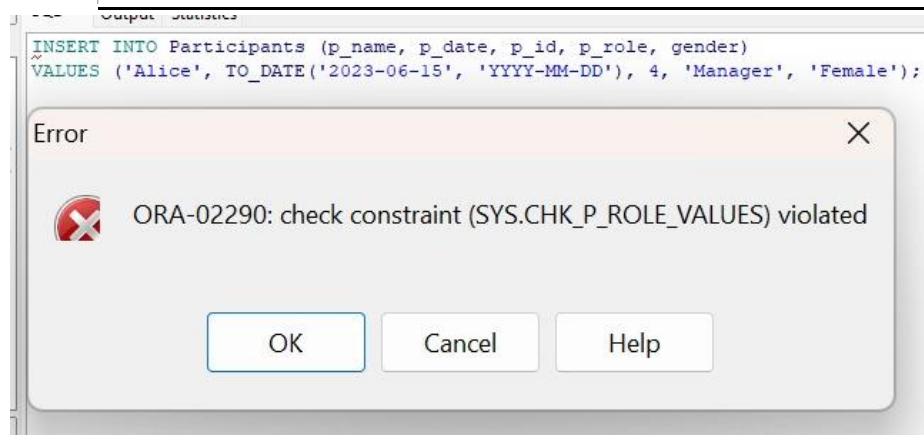
בודק שכמות הציוד היא חיובית ולא שלילית:

```
ALTER TABLE Equipment
ADD CONSTRAINT chk_amount_nonnegative
CHECK (amount >= 0);
```



בודק שתפקיד של משתתף קיים במערכת:

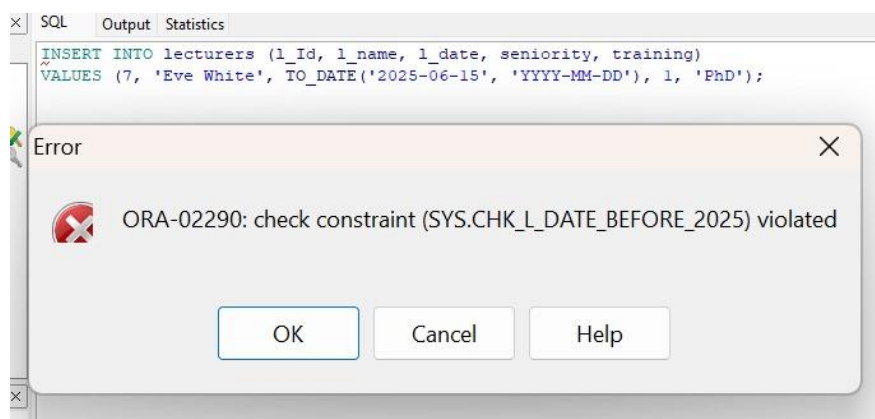
```
ALTER TABLE Participants
ADD CONSTRAINT chk_p_role_values
CHECK (p_role IN ('Certified Nurse','First Responder','EMT','Paramedic'));
```





בודק שהתאריך לידה של מרצה הוא לא אחרי השנה 2024:

```
ALTER TABLE lecturers
ADD CONSTRAINT chk_l_date_before_2025
CHECK (l_date < TO_DATE('2025-01-01', 'YYYY-MM-DD'));
```



## שלב 3-תוכניות

### פונקציה 1:

הפונקציה מקבלת תפקיד משתתף, ושם קורס. היא מחזירה משתתפים שתפקידם הוא התפקיד שניתן והם לומדים את הקורס שניתן, ומדפיסה את פרטי המשתתפים ואת מספרם הכולל, אם אין משתתפים עם התפקיד או הקורס שניתנו, הפונקציה תדפיס שגיאה מתאימה.

```
CREATE OR REPLACE FUNCTION GetParticipantsAndCourses (
    role IN VARCHAR2,
    course name IN VARCHAR2
) RETURN SYS_REFCURSOR IS
    participants_courses_cursor SYS_REFCURSOR;

    no_participants EXCEPTION;
    no_role EXCEPTION;
    no_course EXCEPTION;
    participant_count NUMBER := 0;
    role_count NUMBER := 0;
    course_count NUMBER := 0;

    TYPE participant_record IS RECORD (
        p_id Participants.p_id%TYPE,
        p_name Participants.p_name%TYPE,
        p_role Participants.p_role%TYPE
    );
    participant_rec participant_record;

BEGIN
    SELECT COUNT(*) INTO role_count
    FROM Participants
    WHERE p_role = role;

    IF role_count = 0 THEN
        RAISE no_role;
    END IF;

    SELECT COUNT(*) INTO course_count
    FROM Courses
    WHERE c_name = course_name;

    IF course_count = 0 THEN
        RAISE no_course;
    END IF;

    OPEN participants_courses_cursor FOR
        SELECT p.p_id, p.p_name, p.p_role
        FROM Participants p
        JOIN belongs b ON p.p_id = b.p_id
        JOIN StudentsGroup sg ON b.g_id = sg.g_id
        JOIN Courses c ON sg.c_id = c.c_id
        WHERE p.p_role = role AND c.c_name = course_name;

    LOOP
        FETCH participants_courses_cursor INTO participant_rec;
        EXIT WHEN participants_courses_cursor%NOTFOUND;
        participant_count := participant_count + 1;
        DBMS_OUTPUT.PUT_LINE('Participant ID: ' || participant_rec.p_id || ', Name: ' ||
            participant_rec.p_name || ', Role: ' || participant_rec.p_role);
    END LOOP;

    IF participant_count = 0 THEN
        RAISE no_participants;
    ELSE
        DBMS_OUTPUT.PUT_LINE('Total participants returned by the function: ' ||
            participant_count);
    END IF;

    OPEN participants_courses_cursor FOR
        SELECT p.p_id, p.p_name
        FROM Participants p
        JOIN belongs b ON p.p_id = b.p_id
        JOIN StudentsGroup sg ON b.g_id = sg.g_id
        JOIN Courses c ON sg.c_id = c.c_id
        WHERE p.p_role = role AND c.c_name = course_name;
```

```

RETURN participants_courses_cursor;

EXCEPTION
    WHEN no_participants THEN
        DBMS_OUTPUT.PUT_LINE('No participants found for the given role and course.');
```

```

        RETURN NULL;
    WHEN no_role THEN
        DBMS_OUTPUT.PUT_LINE('No participants found for the given role.');
```

```

        RETURN NULL;
    WHEN no_course THEN
        DBMS_OUTPUT.PUT_LINE('No participants found for the given course.');
```

```

        RETURN NULL;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
        RETURN NULL;
END GetParticipantsAndCourses;
```

## דוגמת הרצה:

### הרצת הפונקציה:

Test Window - Script for function GETPARTICIPANTSANDCOURSES@XE

Test script DBMS Output Statistics Profiler Trace

```

1 begin
2   -- Call the function
3   :result := getparticipantsandcourses(role => :role,
4                                       course_name => :course_name);
5 end;
```

Variable	Type	Value
result	Cursor	<Cursor>
role	String	Certified Nurse
course_name	String	Trauma Care

### תוצאות פלט הפונקציה:

Test Window - Script for function GETPARTICIPANTSANDCOURSES@XE

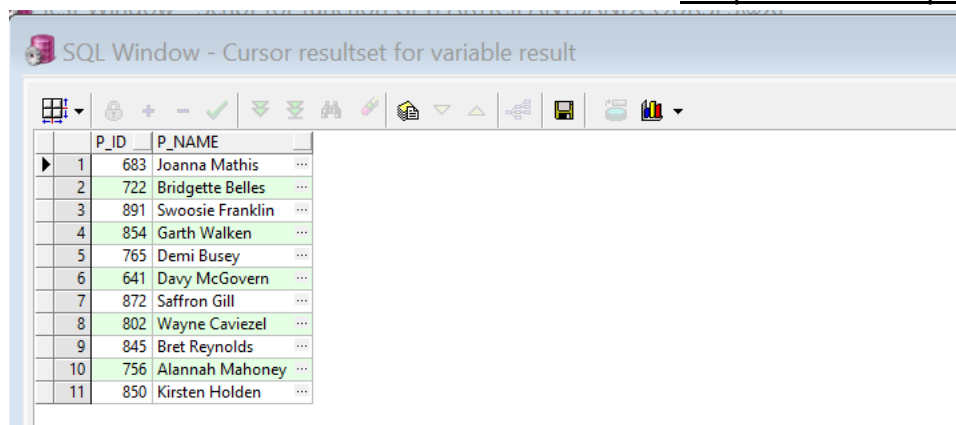
Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 Enabled

```

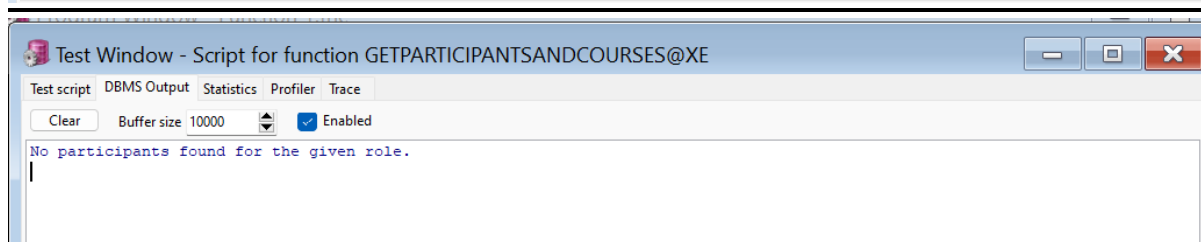
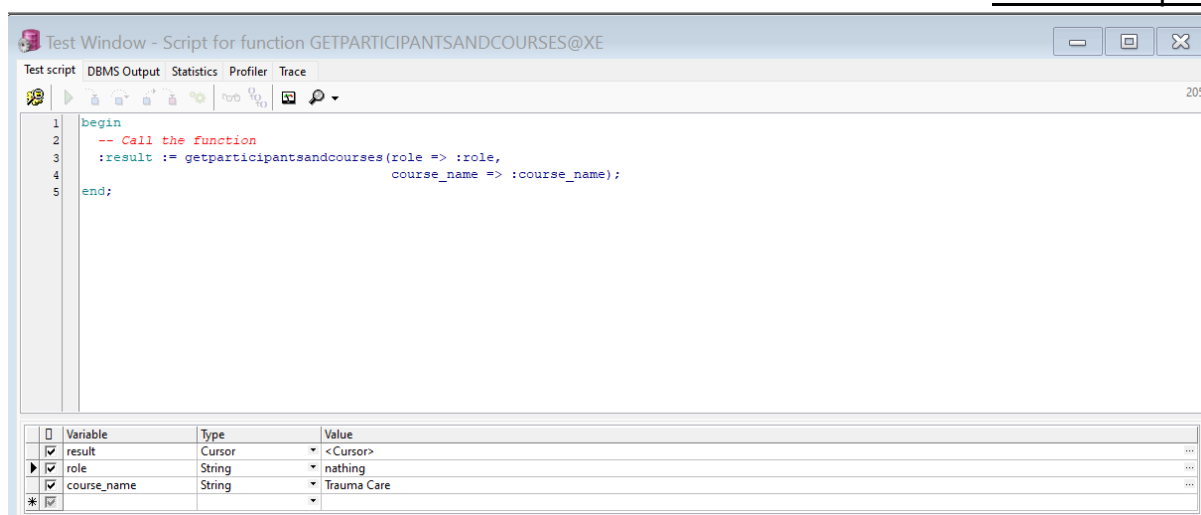
Participant ID: 683, Name: Joanna Mathis, Role: Certified Nurse
Participant ID: 722, Name: Bridgette Belles, Role: Certified Nurse
Participant ID: 891, Name: Swoosie Franklin, Role: Certified Nurse
Participant ID: 854, Name: Garth Walken, Role: Certified Nurse
Participant ID: 765, Name: Demi Busey, Role: Certified Nurse
Participant ID: 641, Name: Davy McGovern, Role: Certified Nurse
Participant ID: 872, Name: Saffron Gill, Role: Certified Nurse
Participant ID: 802, Name: Wayne Caviezel, Role: Certified Nurse
Participant ID: 845, Name: Bret Reynolds, Role: Certified Nurse
Participant ID: 756, Name: Alannah Mahoney, Role: Certified Nurse
Participant ID: 850, Name: Kirsten Holden, Role: Certified Nurse
Total participants returned by the function: 11
```

## ערך החוזר מהפונקציה:



	P_ID	P_NAME
1	683	Joanna Mathis
2	722	Bridgette Belles
3	891	Swoosie Franklin
4	854	Garth Walken
5	765	Demi Busey
6	641	Davy McGovern
7	872	Saffron Gill
8	802	Wayne Caviezel
9	845	Bret Reynolds
10	756	Alannah Mahoney
11	850	Kirsten Holden

## במקרה של חריגה



## פרוצדורה 1:

הפרוצדורה מקבלת שם משתתף, ת"ז משתתף ומספר קבוצה. היא מוסיפה את המשתתף לקבוצה אם הוא לא נמצא בה כבר, מדפיסה הודעה על הוספתו לקורס, ומעדכנת את מספר הקורסים בהם הוא משתתף. אם המשתתף כבר נמצא בקבוצה או אם מתרחשת שגיאה, הפרוצדורה מטפלת בשגיאות ומדפיסה הודעה מתאימה.

```

CREATE OR REPLACE PROCEDURE AddParticipantToGroup (
  participant_name IN Participants.p_name%TYPE,
  participant_id IN Participants.p_id%TYPE,
  group_id IN StudentsGroup.g_id%TYPE
) IS
  course count NUMBER := 0;
  v_exists NUMBER := 0;
  category Courses.categories%TYPE;
  course_name Courses.c_name%TYPE;

  message VARCHAR2(200);

  CURSOR participant_courses_cursor IS
    SELECT DISTINCT c.c_Id, c.categories
    FROM belongs b
```

```

        JOIN StudentsGroup sg ON b.g_id = sg.g_id
        JOIN Courses c ON sg.c_Id = c.c_Id
        WHERE b.p_id = participant_id;

BEGIN
    SELECT COUNT(*)
    INTO v_exists
    FROM belongs
    WHERE g_id = group_id
        AND p_id = participant_id;

    IF v_exists > 0 THEN
        message := 'The participant is already in the group.';
        DBMS_OUTPUT.PUT_LINE(message);
    ELSE
        INSERT INTO belongs (g_id, p_id)
        VALUES (group_id, participant_id);
        COMMIT;

        SELECT c.categories, c.c_name
        INTO category, course_name
        FROM StudentsGroup sg
        JOIN Courses c ON sg.c_Id = c.c_Id
        WHERE sg.g_id = group_id;

        message := participant_name || ' has been added to the course: ' || course_name;
        DBMS_OUTPUT.PUT_LINE(message);

        FOR course_rec IN participant_courses_cursor LOOP
            course_count := course_count + 1;
        END LOOP;

        message := participant_name || ' is enrolled in ' || course_count || ' courses.';
        DBMS_OUTPUT.PUT_LINE(message);
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Course category not found for the group.');
```

```

    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Participant is already in the group.');
```

```

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

## דוגמת הרצה:

מצב הטבלה לפני הרצת הפרוצדורה:

```
SELECT * FROM belongs
WHERE G_ID = 400;
```

	G_ID	P_ID
1	400	500
2	400	718

## הרצת הפרוצדורה:

Test Window - Script for procedure ADPPARTICIPANTTOGROUP@XE

Test script DBMS Output Statistics Profiler Trace

```

1 begin
2   -- Call the procedure
3   addparticipanttogroup(participant_name => :participant_name,
4     participant_id => :participant_id,
5     group_id => :group_id);
6 end;
```

Variable	Type	Value
participant_name	String	Eugene Moss
participant_id	Float	520
group_id	Float	400

## תוצאות פלט הפרוצדורה:

Test Window - Script for procedure ADPPARTICIPANTTOGROUP@XE

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 Enabled

```

Eugene Moss has been added to the course: Health and Safety
Eugene Moss is enrolled in 1 courses.
```

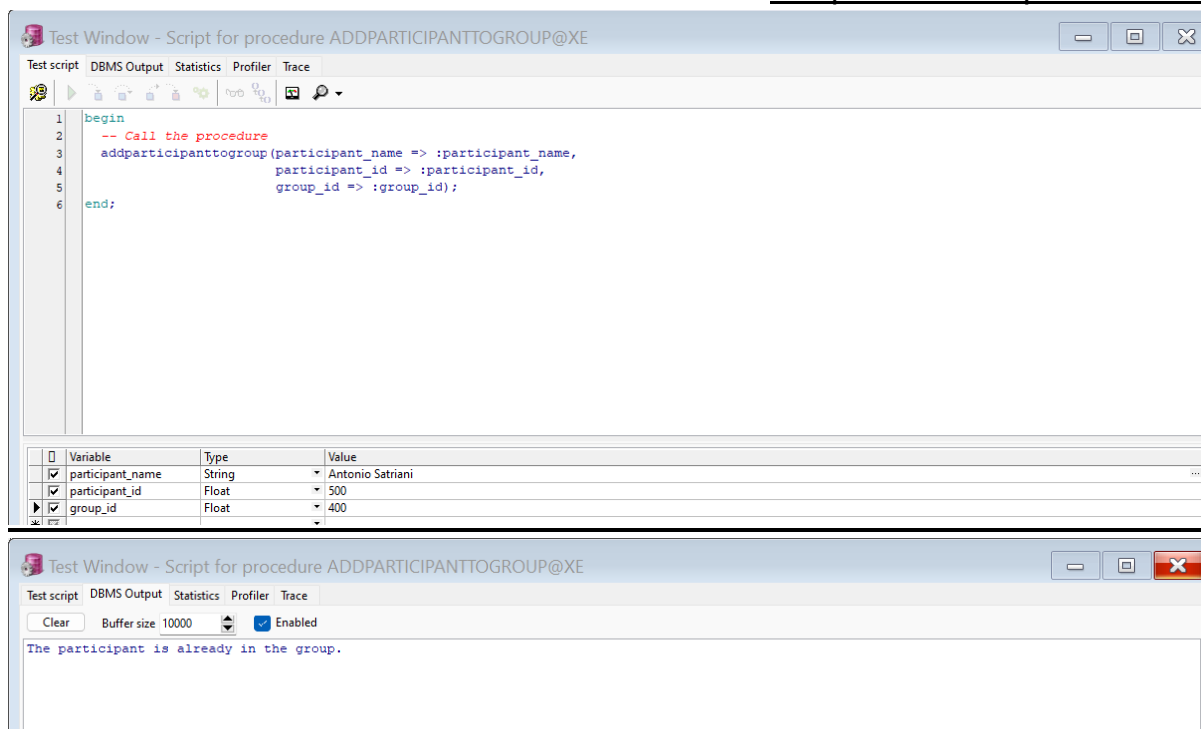
## מצב הטבלה לאחר הרצת הפרוצדורה:

```
SELECT * FROM belongs
WHERE G_ID = 400;
```

	G_ID	P_ID
1	400	500
2	400	520
3	400	718

## במקרה של חריגה:

### כאשר המשתתף כבר נימצא בקבוצה



## תוכנית 1:

התוכנית מקבלת קבוצה, תפקיד ושם קורס כקלט. היא מוציאה את כל המשתתפים עם התפקיד הנתון שלומדים בקורס הנתון בעזרת הפונקציה, בלולאה היא שולפת כל משתתף מהcursor החוזר מהפונקציה ומוסיפה אותו לקבוצה בעזרת הפרוצדורה, אם מתרחשת שגיאה התוכנית מדפיסה הודעת שגיאה מתאימה.

```

DECLARE
    participants_courses_cursor SYS_REFCURSOR;
    v_p_id Participants.p_id%TYPE;
    v_p_name Participants.p_name%TYPE;
    v_group_id StudentsGroup.g_id%TYPE := &group_id;
    v_role VARCHAR2(30) := '&p_role';
    v_course_name VARCHAR2(30) := '&course_name';
BEGIN
    participants_courses_cursor := GetParticipantsAndCourses(v_role, v_course_name);

    LOOP
        FETCH participants_courses_cursor INTO v_p_id, v_p_name;
        EXIT WHEN participants_courses_cursor%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('Adding participant: ' || v_p_name || ', ID: ' || v_p_id);

        AddParticipantToGroup(v_p_name, v_p_id, v_group_id);
    END LOOP;

    CLOSE participants_courses_cursor;

    DBMS_OUTPUT.PUT_LINE('Participants have been added to the new group.');
```

```

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

## דוגמת הרצה:

Variables

Name	Value
group_id	400
p_role	Certified Nurse
course_name	Trauma Care

OK

Cancel

Clear

## תוצאות פלט התוכנית:

SQL	Output	Statistics
	<div> <div>Clear</div> <div>Buffer size 10000</div> <div>Enabled</div> </div>	
	<p>Participant ID: 683, Name: Joanna Mathis, Role: Certified Nurse</p> <p>Participant ID: 722, Name: Bridgette Belles, Role: Certified Nurse</p> <p>Participant ID: 891, Name: Swoosie Franklin, Role: Certified Nurse</p> <p>Participant ID: 854, Name: Garth Walken, Role: Certified Nurse</p> <p>Participant ID: 765, Name: Demi Busey, Role: Certified Nurse</p> <p>Participant ID: 641, Name: Davy McGovern, Role: Certified Nurse</p> <p>Participant ID: 872, Name: Saffron Gill, Role: Certified Nurse</p> <p>Participant ID: 802, Name: Wayne Caviezel, Role: Certified Nurse</p> <p>Participant ID: 845, Name: Bret Reynolds, Role: Certified Nurse</p> <p>Participant ID: 756, Name: Alannah Mahoney, Role: Certified Nurse</p> <p>Participant ID: 850, Name: Kirsten Holden, Role: Certified Nurse</p> <p>Total participants returned by the function: 11</p> <p>Adding participant: Joanna Mathis, ID: 683</p> <p>Joanna Mathis has been added to the course: Health and Safety</p> <p>Joanna Mathis is enrolled in 4 courses.</p> <p>Adding participant: Bridgette Belles, ID: 722</p> <p>Bridgette Belles has been added to the course: Health and Safety</p> <p>Bridgette Belles is enrolled in 3 courses.</p> <p>Adding participant: Swoosie Franklin, ID: 891</p> <p>Swoosie Franklin has been added to the course: Health and Safety</p> <p>Swoosie Franklin is enrolled in 3 courses.</p> <p>Adding participant: Garth Walken, ID: 854</p> <p>Garth Walken has been added to the course: Health and Safety</p> <p>Garth Walken is enrolled in 2 courses.</p> <p>Adding participant: Demi Busey, ID: 765</p> <p>Demi Busey has been added to the course: Health and Safety</p> <p>Demi Busey is enrolled in 3 courses.</p> <p>Adding participant: Davy McGovern, ID: 641</p> <p>Davy McGovern has been added to the course: Health and Safety</p> <p>Davy McGovern is enrolled in 2 courses.</p> <p>Adding participant: Saffron Gill, ID: 872</p> <p>Saffron Gill has been added to the course: Health and Safety</p> <p>Saffron Gill is enrolled in 3 courses.</p> <p>Adding participant: Wayne Caviezel, ID: 802</p> <p>Wayne Caviezel has been added to the course: Health and Safety</p> <p>Wayne Caviezel is enrolled in 2 courses.</p> <p>Adding participant: Bret Reynolds, ID: 845</p> <p>Bret Reynolds has been added to the course: Health and Safety</p> <p>Bret Reynolds is enrolled in 4 courses.</p> <p>Adding participant: Alannah Mahoney, ID: 756</p> <p>Alannah Mahoney has been added to the course: Health and Safety</p> <p>Alannah Mahoney is enrolled in 2 courses.</p> <p>Adding participant: Kirsten Holden, ID: 850</p> <p>Kirsten Holden has been added to the course: Health and Safety</p> <p>Kirsten Holden has been added to the course: Health and Safety</p> <p>Kirsten Holden is enrolled in 4 courses.</p> <p>Participants have been added to the new group.</p>	



## מצב הטבלה לאחר הרצת התוכנית:

```
SELECT * FROM belongs
WHERE G_ID = 400;
```

	G_ID	P_ID
1	400	500
2	400	520
3	400	641
4	400	683
5	400	718
6	400	722
7	400	756
8	400	765
9	400	802
10	400	845
11	400	850
12	400	854
13	400	872
14	400	891

## פונקציה 2:

הפונקציה מקבלת שם קורס ושם מרצה כקלט. היא מוצאת את כל הציוד שמשמש בקורס הזה על ידי המרצה הזה, מדפיסה את פרטי הציוד ומחזירה סמן (cursor) עם תוצאות השאילתה. אם לא נמצא ציוד, היא מדפיסה הודעה מתאימה

```
CREATE OR REPLACE FUNCTION find_equipment_used_in_course(
    course_name IN VARCHAR2,
    lecturer_name IN VARCHAR2
)
RETURN SYS_REFCURSOR
AS
    equipment_cursor SYS REFCURSOR;
    internal_cursor SYS REFCURSOR;
    equipment_name VARCHAR2(100);
    equipment_id NUMBER;
    e_count NUMBER := 0;

    no_equipment_found EXCEPTION;
BEGIN
    OPEN internal_cursor FOR
        SELECT e.e_name, e.e_id
        FROM equipment e
        JOIN uses u ON e.e_id = u.e_id
        JOIN courses c ON u.c_id = c.c_id
        JOIN teaches_the tt ON c.c_id = tt.c_id
        JOIN lecturers l ON tt.l_id = l.l_id
        WHERE c.c_name = course_name
        AND l.l_name = lecturer_name;

    LOOP
        FETCH internal_cursor INTO equipment_name, equipment_id;
        EXIT WHEN internal_cursor%NOTFOUND;
        e_count := e_count + 1;
        DBMS_OUTPUT.PUT_LINE('Equipment used in course ' || course_name || ' by lecturer ' ||
            lecturer_name || ': ' || equipment_name || ' (ID: ' || equipment_id || ')');
    END LOOP;

    CLOSE internal_cursor;

    IF e_count = 0 THEN
        RAISE no_equipment_found;
    END IF;
END;
```

```

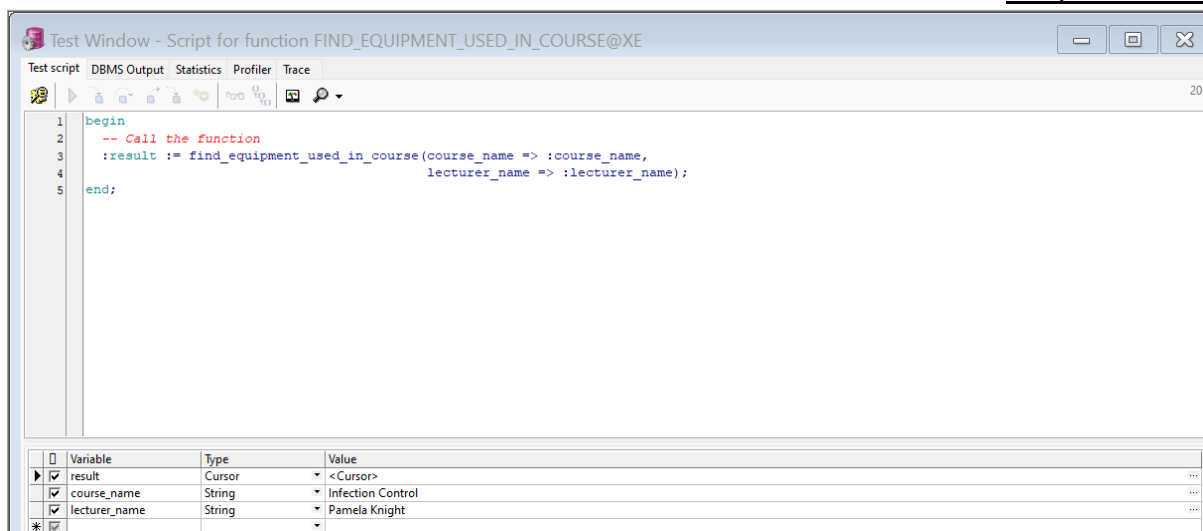
OPEN equipment_cursor FOR
  SELECT e.e name, e.e_id
  FROM equipment e
  JOIN uses u ON e.e id = u.e id
  JOIN courses c ON u.c_id = c.c_id
  JOIN teaches_the tt ON c.c_id = tt.c_id
  JOIN lecturers l ON tt.l_id = l.l_id
  WHERE c.c name = course_name
  AND l.l_name = lecturer_name;

RETURN equipment_cursor;

EXCEPTION
  WHEN no equipment found THEN
    DBMS_OUTPUT.PUT_LINE('No equipment found for the specified course and lecturer');
    RETURN NULL;
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error while fetching equipment used in course: ' || SQLERRM);
    RETURN NULL;
END;
```

## דוגמת הרצה:

### הרצת הפונקציה:



Test Window - Script for function FIND\_EQUIPMENT\_USED\_IN\_COURSE@XE

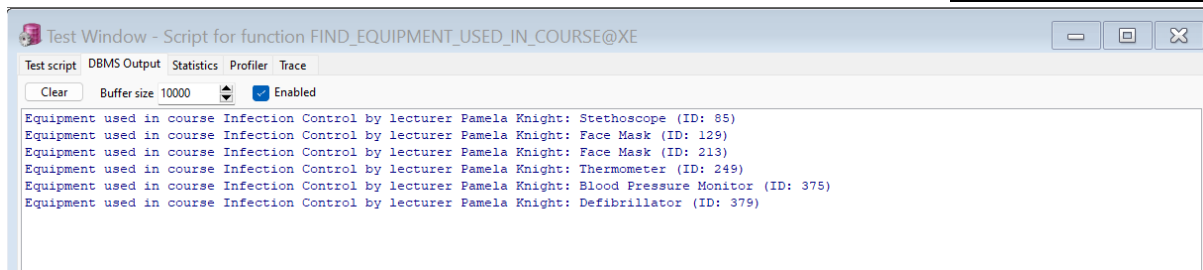
Test script DBMS Output Statistics Profiler Trace

```

1 begin
2   -- Call the function
3   :result := find_equipment_used_in_course(course_name => :course_name,
4                                           lecturer_name => :lecturer_name);
5 end;
```

Variable	Type	Value
result	Cursor	<Cursor>
course_name	String	Infection Control
lecturer_name	String	Pamela Knight

### תוצאות פלט הפונקציה:



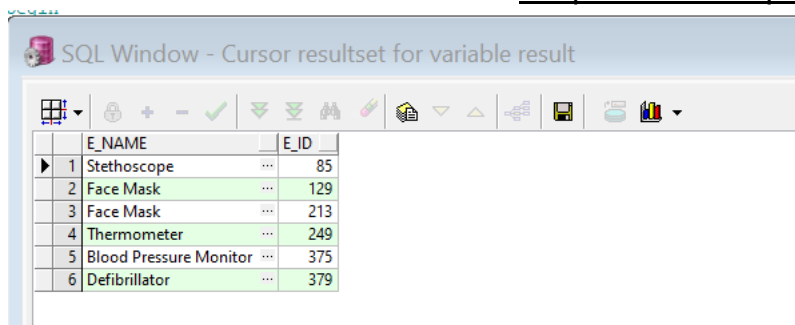
Test Window - Script for function FIND\_EQUIPMENT\_USED\_IN\_COURSE@XE

Test script DBMS Output Statistics Profiler Trace

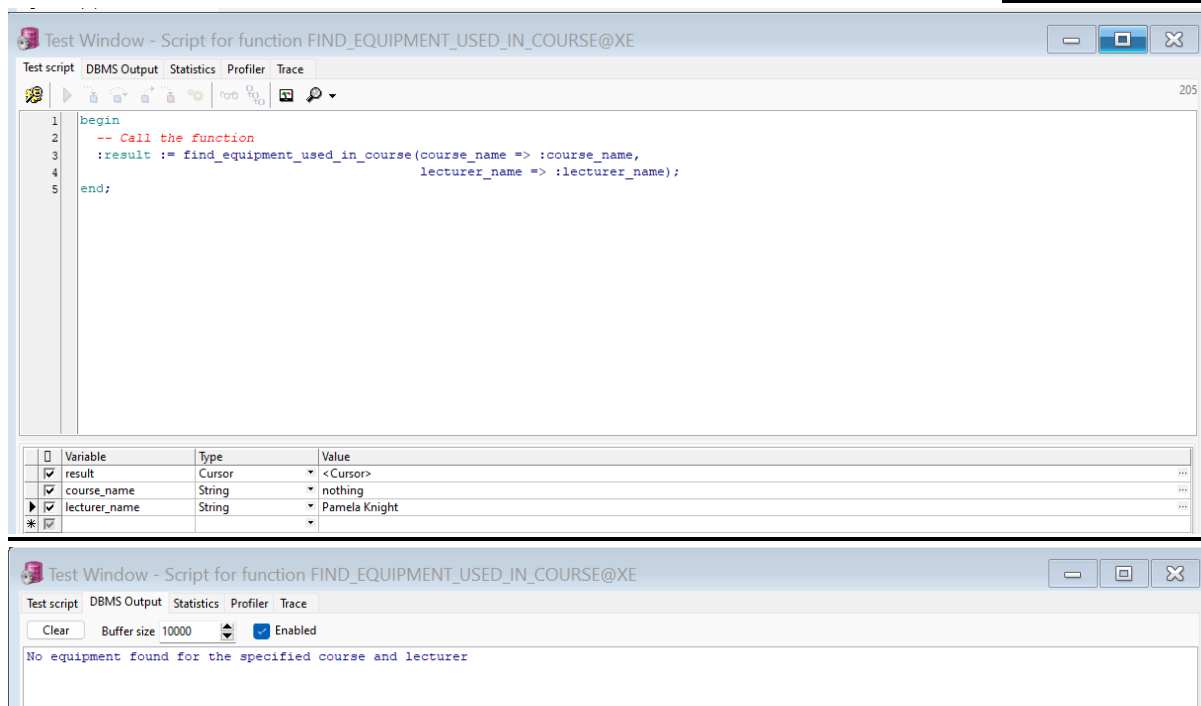
Clear Buffer size 10000 Enabled

```

Equipment used in course Infection Control by lecturer Pamela Knight: Stethoscope (ID: 85)
Equipment used in course Infection Control by lecturer Pamela Knight: Face Mask (ID: 129)
Equipment used in course Infection Control by lecturer Pamela Knight: Face Mask (ID: 213)
Equipment used in course Infection Control by lecturer Pamela Knight: Thermometer (ID: 249)
Equipment used in course Infection Control by lecturer Pamela Knight: Blood Pressure Monitor (ID: 375)
Equipment used in course Infection Control by lecturer Pamela Knight: Defibrillator (ID: 379)
```

ערך החוזר מהפונקציה:


	E_NAME	E_ID
1	Stethoscope	85
2	Face Mask	129
3	Face Mask	213
4	Thermometer	249
5	Blood Pressure Monitor	375
6	Defibrillator	379

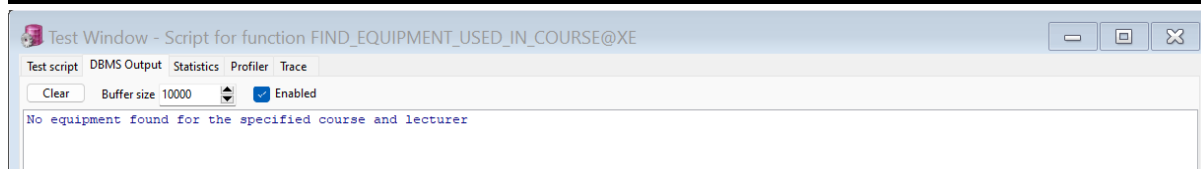
במקרה של חריגה:


```

1 begin
2   -- Call the function
3   :result := find_equipment_used_in_course(course_name => :course_name,
4   :lecturer_name => :lecturer_name);
5 end;

```

Variable	Type	Value
result	Cursor	<Cursor>
course_name	String	nothing
lecturer_name	String	Pamela Knight

Clear Buffer size 10000 ☒ Enabled

No equipment found for the specified course and lecturer

פרוצדורה 2:

הפרוצדורה מקבלת שם ציוד, מספר מזהה שלו, ומספר ומעדכנת את הכמות של הציוד במערכת על פי מזהה הציוד והכמות שניתן. לאחר מכן, היא מדפיסה את רשימת הקורסים והמרצים שמשתמשים בציוד זה. אם יש שגיאה במהלך הביצוע, היא מדפיסה הודעת שגיאה מתאימה.

```

CREATE OR REPLACE PROCEDURE process_equipment(
    equipment_name IN VARCHAR2,
    equipment_id IN NUMBER,
    amount_to_add IN NUMBER
)
IS
    course_name VARCHAR2(100);
    lecturer_name VARCHAR2(100);
    courses cur SYS REFCURSOR;
    rows_updated NUMBER;
    equipment_name_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO equipment_name_count
    FROM equipment
    WHERE e_name = equipment_name;

    IF equipment_name_count = 0 THEN
        RAISE NO_DATA_FOUND;
    END IF;

```

```

UPDATE equipment
SET amount = amount + amount_to_add
WHERE e_id = equipment_id;
rows_updated := SQL%ROWCOUNT;

IF rows_updated = 0 THEN
    RAISE NO_DATA_FOUND;
END IF;

DBMS_OUTPUT.PUT_LINE('Courses using equipment ' || equipment_name || ':');

OPEN courses_cur FOR
SELECT c.c_name AS course_name, l.l_name AS lecturer_name
FROM courses c
JOIN teaches tt ON c.c_id = tt.c_id
JOIN lecturers l ON tt.l_id = l.l_id
JOIN uses u ON c.c_id = u.c_id
WHERE u.e_id = equipment_id;

LOOP
    FETCH courses_cur INTO course_name, lecturer_name;
    EXIT WHEN courses_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('    Course Name: ' || course_name || ', Lecturer: ' ||
lecturer_name);
END LOOP;

CLOSE courses_cur;

EXCEPTION
WHEN NO_DATA_FOUND THEN
    IF equipment_name count = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No equipment found with name ' || equipment_name);
    ELSE
        DBMS_OUTPUT.PUT_LINE('No equipment found with ID ' || equipment_id);
    END IF;
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error processing equipment: ' || SQLERRM);
END;
```

## דוגמת הרצה:

מצב הטבלה לפני הרצת הפרוצדורה:

SQL Output Statistics					
select * from Equipment;					
	E_DATE	E_ID	E_NAME	AMOUNT	
1	03/11/2042	218	CPR Manikin	50	
2	08/12/2031	219	Thermometer	71	
3	11/01/2031	220	Defibrillator	92	
4	15/12/2039	221	First Aid Kit	46	
5	31/01/2040	222	Face Mask	71	
6	30/05/2031	223	Oxygen Tank	47	
7	31/10/2036	224	Defibrillator	84	
8	28/12/2026	225	Oxygen Tank	44	
9	20/04/2027	226	Oxygen Tank	47	
10	10/10/2025	227	Face Mask	82	
11	09/10/2029	228	Oxygen Tank	81	
12	06/08/2031	229	Stethoscope	90	
13	30/03/2042	230	Blood Pressure Monitor	77	
14	10/09/2041	231	Face Mask	80	
15	12/06/2031	232	Surgical Gloves	76	
16	07/04/2030	233	Glucose Meter	71	

## הרצת הפרוצדורה:

Test Window - Script for procedure PROCESS\_EQUIPMENT@XE

Test script DBMS Output Statistics Profiler Trace

```

1 begin
2   -- Call the procedure
3   process_equipment(equipment_name => :equipment_name,
4                     equipment_id => :equipment_id,
5                     amount_to_add => :amount_to_add);
6 end;

```

Variable	Type	Value
equipment_name	String	CPR Manikin
equipment_id	Float	218
amount_to_add	Float	1

## תוצאות פלט הפרוצדורה:

Test Window - Script for procedure PROCESS\_EQUIPMENT@XE

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 Enabled

Courses using equipment CPR Manikin:

Course Name: Pediatric Care, Lecturer: Brian Orbit

Course Name: Pediatric Care, Lecturer: Chris Sutherland

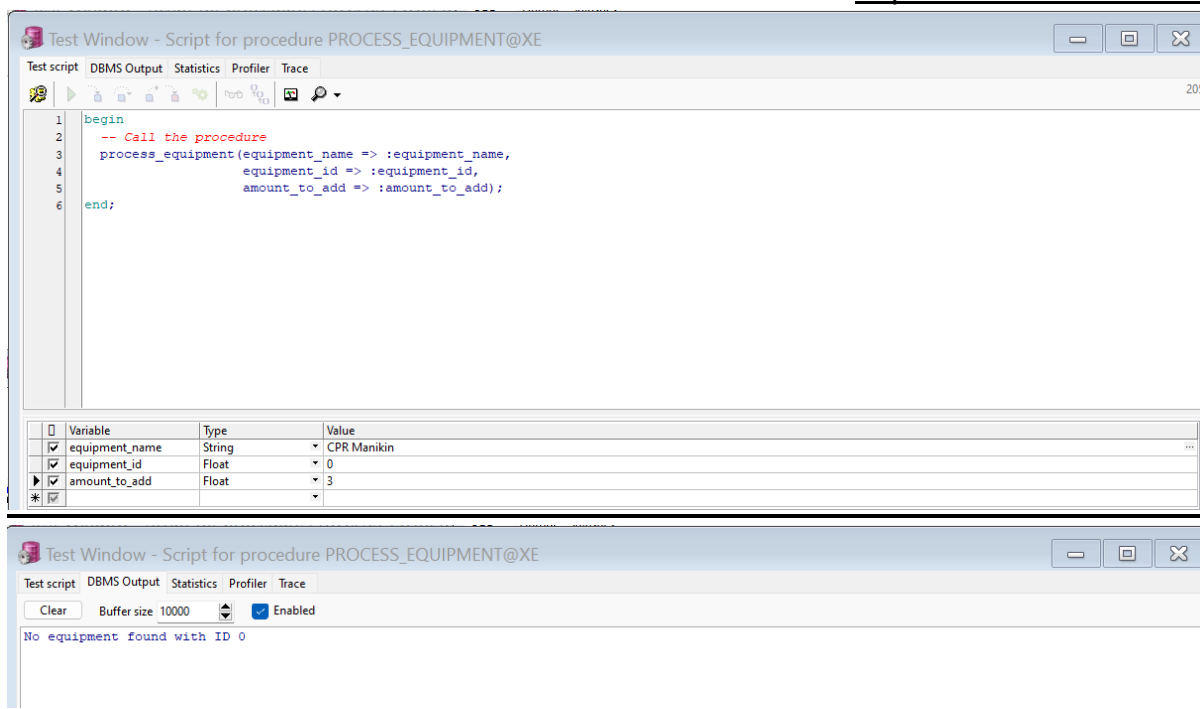
## מצב הטבלה לאחר הרצת הפרוצדורה:

SQL Output Statistics

```
select * from Equipment;
```

	E_DATE	E_ID	E_NAME	AMOUNT
1	03/11/2042	218	CPR Manikin	51
2	08/12/2031	219	Thermometer	71
3	11/01/2031	220	Defibrillator	92
4	15/12/2039	221	First Aid Kit	46
5	31/01/2040	222	Face Mask	71
6	30/05/2031	223	Oxygen Tank	47
7	31/10/2036	224	Defibrillator	84
8	28/12/2026	225	Oxygen Tank	44
9	20/04/2027	226	Oxygen Tank	47
10	10/10/2025	227	Face Mask	82
11	09/10/2029	228	Oxygen Tank	81
12	06/08/2031	229	Stethoscope	90
13	30/03/2042	230	Blood Pressure Monitor	77

## במקרה של חריגה: מספר מזהה של ציוד לא קיים



## תוכנית 2:

התוכנית מקבלת שם קורס, שם מרצה וכמות להוספה כקלטים. היא מוצאת את כל הציוד המשמש בקורס ובמרצה הניתנים באמצעות הפונקציה בלולאה היא שולפת כל ציוד מהcursor החוזר מהפונקציה ומעדכנת את כמות הציוד עבור כל פריט באמצעות הפרוצדורה, אם מתרחשת שגיאה במהלך הביצוע, התוכנית מדפיסה הודעת שגיאה מתאימה.

```

DECLARE
  v_course_name VARCHAR2(100) := '&course_name';
  v_lecturer_name VARCHAR2(100) := '&lecturer_name';
  v_amount_to_add NUMBER := &amount_to_add;
  v_equipment_cur SYS REFCURSOR;
  v_equipment_name VARCHAR2(100);
  v_equipment_id NUMBER;
BEGIN
  v_equipment_cur := find_equipment_used_in_course(v_course_name, v_lecturer_name);
  DBMS_OUTPUT.PUT_LINE('Retrieving equipment details for the specified course and lecturer.');
```

```

  LOOP
    FETCH v_equipment_cur INTO v_equipment_name, v_equipment_id;
    EXIT WHEN v_equipment_cur%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Processing equipment item: ' || v_equipment_name || ' (ID: ' || v_equipment_id || ').');
```

```

    process_equipment(v_equipment_name, v_equipment_id, v_amount_to_add);
  END LOOP;

  CLOSE v_equipment_cur;
  DBMS_OUTPUT.PUT_LINE('All equipment items have been successfully updated.');
```

```

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred during the execution of the program: ' || SQLERRM);
END;
```

## דוגמת הרצה:

מצב הטבלה לפני הרצת התוכנית:

```
SELECT e_name, e_id, amount
FROM equipment
WHERE e_id IN (85, 129, 213, 249, 375, 379);
```

	E_NAME	E_ID	AMOUNT
1	Stethoscope	85	67
2	Face Mask	129	87
3	Face Mask	213	53
4	Thermometer	249	93
5	Blood Pressure Monitor	375	64
6	Defibrillator	379	82

הרצת התוכנית:

Variables

Name	Value
course_name	Infection Control
lecturer_name	Pamela Knight
amount_to_add	10

OK
Cancel
Clear

תוצאות פלט התוכנית:

SQL	Output	Statistics
	Clear Buffer size: 10000 <input checked="" type="checkbox"/> Enabled	
	Equipment used in course Infection Control by lecturer Pamela Knight: Stethoscope (ID: 85) Equipment used in course Infection Control by lecturer Pamela Knight: Face Mask (ID: 129) Equipment used in course Infection Control by lecturer Pamela Knight: Face Mask (ID: 213) Equipment used in course Infection Control by lecturer Pamela Knight: Thermometer (ID: 249) Equipment used in course Infection Control by lecturer Pamela Knight: Blood Pressure Monitor (ID: 375) Equipment used in course Infection Control by lecturer Pamela Knight: Defibrillator (ID: 379) Retrieving equipment details for the specified course and lecturer. Processing equipment item: Stethoscope (ID: 85). Courses using equipment Stethoscope: Course Name: Infection Control, Lecturer: Pamela Knight Processing equipment item: Face Mask (ID: 129). Courses using equipment Face Mask: Course Name: Advanced Life Support, Lecturer: Pablo Pastore Course Name: Infection Control, Lecturer: Pamela Knight Processing equipment item: Face Mask (ID: 213). Courses using equipment Face Mask: Course Name: Basic First Aid, Lecturer: Kitty Lyonne Course Name: Infection Control, Lecturer: Pamela Knight Processing equipment item: Thermometer (ID: 249). Courses using equipment Thermometer: Course Name: Infection Control, Lecturer: Pamela Knight Processing equipment item: Blood Pressure Monitor (ID: 375). Courses using equipment Blood Pressure Monitor: Course Name: Emergency Response, Lecturer: Marina Delta Course Name: Emergency Response, Lecturer: Wesley King Course Name: Emergency Response, Lecturer: Rascal Evans Course Name: Emergency Response, Lecturer: Benjamin Hynde Course Name: Emergency Response, Lecturer: Avenged Phoenix Course Name: Emergency Response, Lecturer: Donal Sorvino Course Name: Infection Control, Lecturer: Pamela Knight Processing equipment item: Defibrillator (ID: 379). Courses using equipment Defibrillator: Course Name: Infection Control, Lecturer: Pamela Knight All equipment items have been successfully updated.	

מצב הטבלה לאחר הרצת התוכנית:

```
SELECT e_name, e_id, amount
FROM equipment
WHERE e_id IN (85, 129, 213, 249, 375, 379);
```

	E_NAME	E_ID	AMOUNT
1	Stethoscope	85	77
2	Face Mask	129	97
3	Face Mask	213	63
4	Thermometer	249	103
5	Blood Pressure Monitor	375	74
6	Defibrillator	379	92



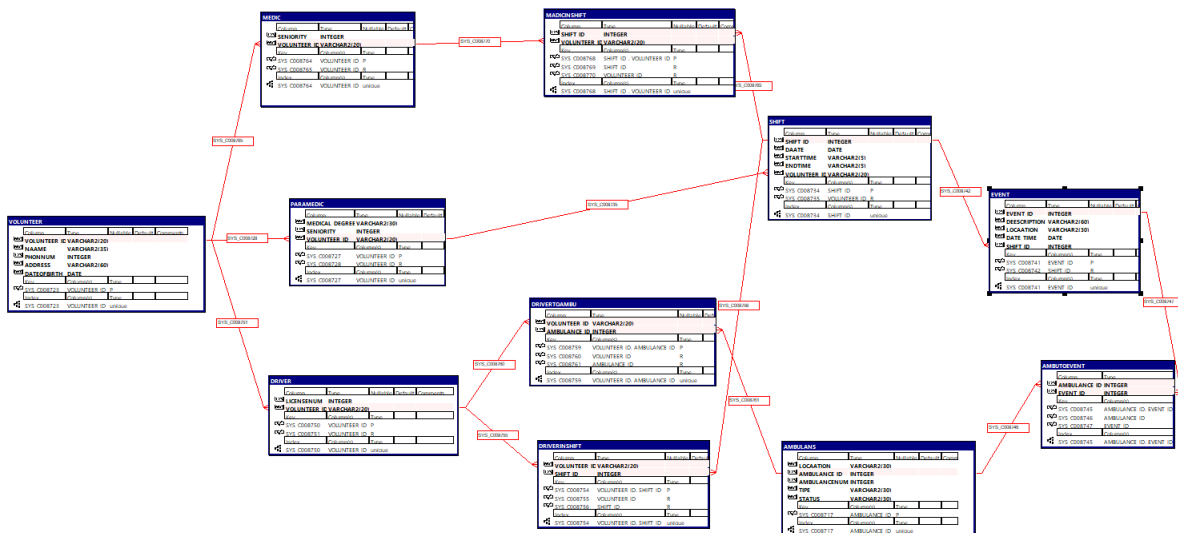
## שלב 4 – אינטגרציה

## שלב ההינדוס לאחר:

הרצת קובץ הגיבוי של הזוג השני:

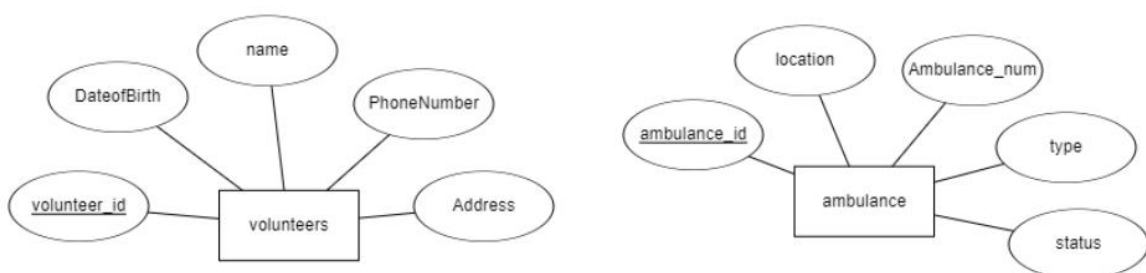


הטבלאות נוצרו ומהם בנינו את ה DSD :

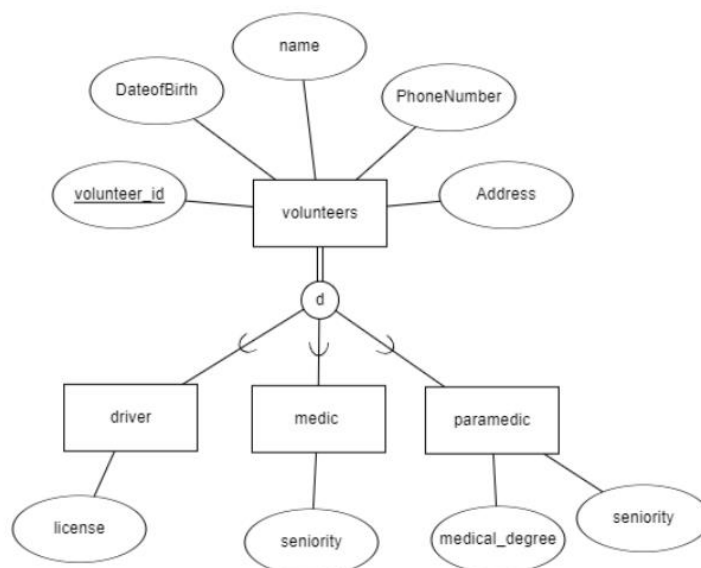


### בניית ה-ERD:

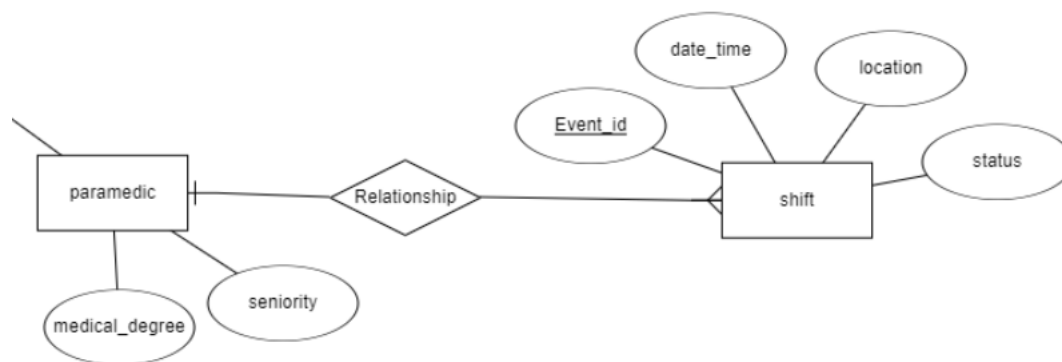
ניצור את הישויות ambulance, volunteers קודם כי הם אינם תלויות בישויות אחרות (אין להם מפתחות זרים):



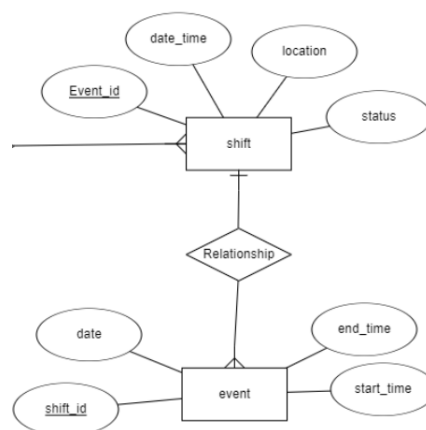
ניצור את הישויות medic, paramedic, driver ניתן לראות שהן יורשות של הישות volunteers מכיוון שלשלושתן יש את השדה מפתח שלה :



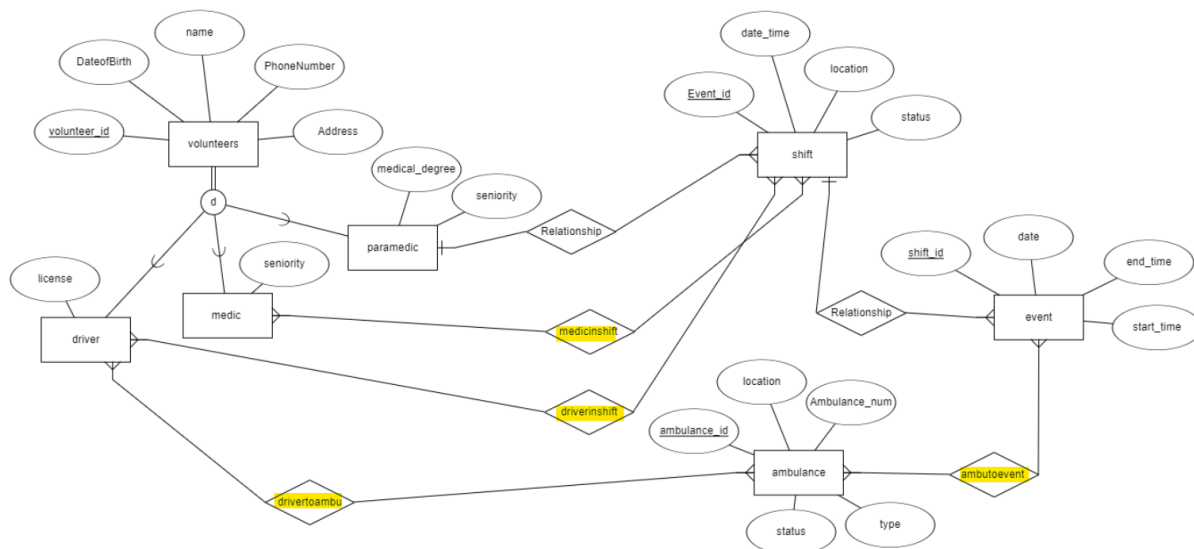
ניצור את הישות shift ניתן לראות שיש לה קשר של יחיד לרבים עם הישות paramedic מכיוון שיש לה את שדה המפתח שלה (מפתח זר):



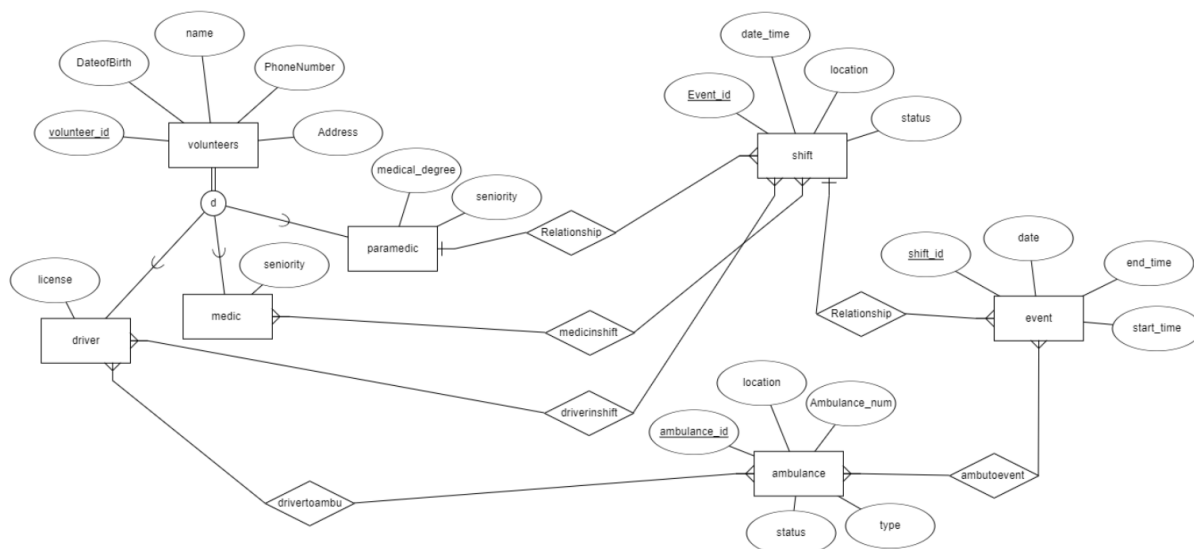
ניצור את הישות event ניתן לראות שגם לה יש קשר של יחיד לרבים עם הישות shift מכיוון שיש לה את השדה המפתח שלה (מפתח זר):



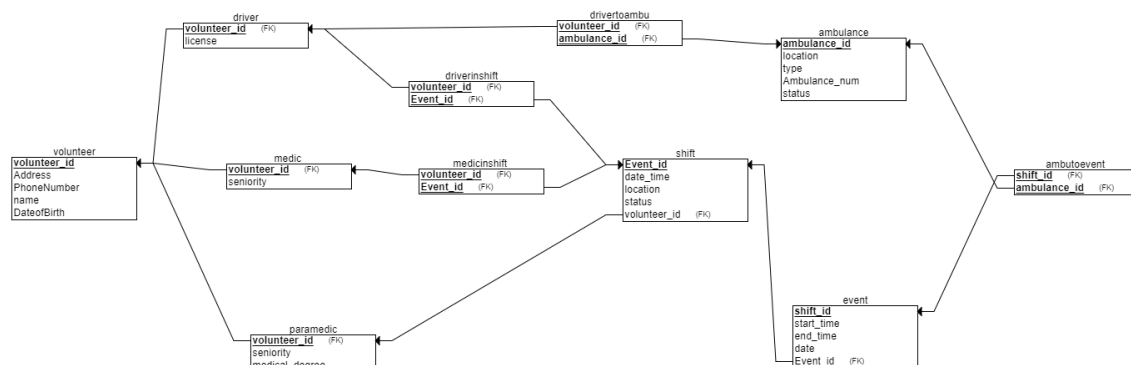
ניצור את הישויות medicinshift, drivertoambu, driverinshift, ambutoevent, שניתן לראות שהן הקשרים בין הישויות מכיוון שהן מכילות שני מפתחות זרים שמשמשים כמפתחות יחידים בטבלה:



התרשים הסופי:



:DSD

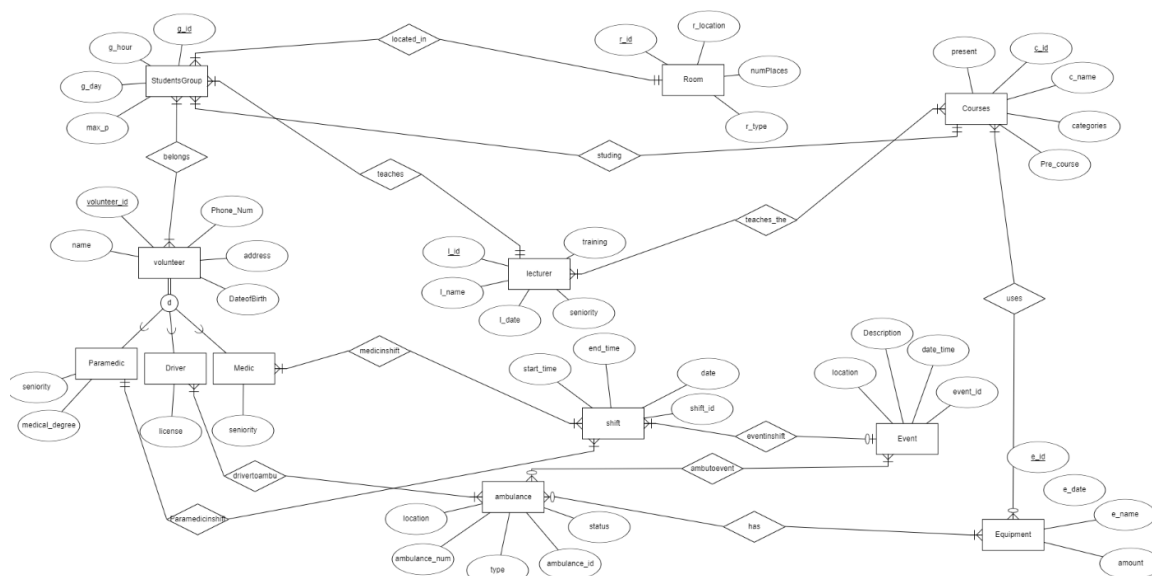


## שלב האינטגרציה:

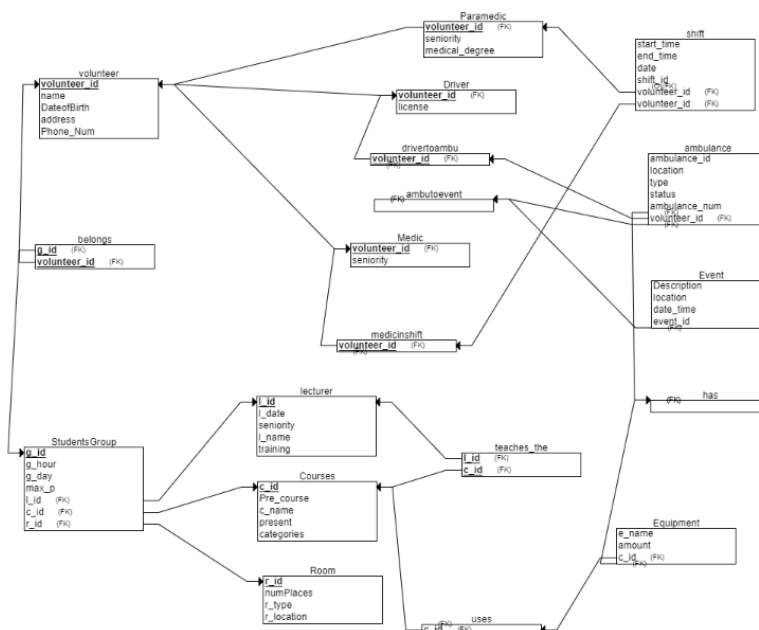
### החלטות לגבי השילוב:

- בפרויקט שלהם יש מתנדבים וסוגי מתנדבים היורשים ממנו ובפרויקט שלנו יש משתתפים נשים את המתנדבים שיהיו במקום המשתתפים עם כל היורשים שלה ונמחק את הטבלה של המשתתפים
- בפרויקט שלהם יש ישות אמבולנס ואצלנו יש ציוד , נעשה קשר בין אמבולנס לציוד והוא יהיה קשר של שייך שזה יבטא את כל הציוד שיש באמבולנס נעשה את הקשר שלא יהיה חובה שבאמבולנס יהיה את כל הציוד מכיוון שחלק מהציוד הוא ציוד של קורסים

### הERD המשולב:



### הDSD המשולב:

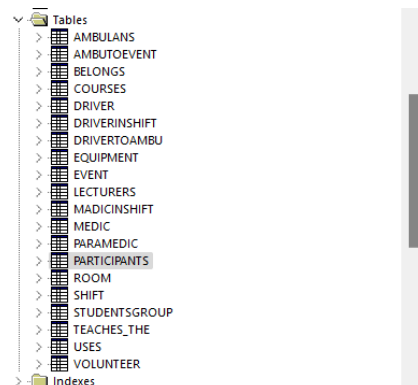


## כתיבת פקודות לשילוב הבסיסי הנתונים:

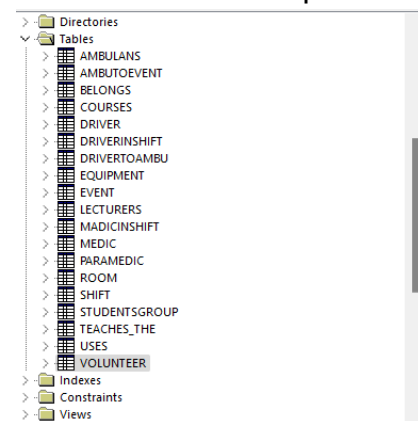
נמחק את טבלת משתתפים שלנו וכל ההגבלות הקשורות אליה:

```
DROP TABLE participants CASCADE CONSTRAINTS;
```

לפני הפקודה:



אחרי הפקודה:



נוסיף עמודה בשם volunteer\_id:

```
ALTER TABLE belongs ADD (volunteer_id VARCHAR2(20));
```

select \* from belongs;

	G_ID	P_ID	VOLUNTEER_ID
1	1	575	...
2	1	631	...
3	3	523	...
4	3	685	...
5	3	756	...
6	6	581	...
7	6	768	...
8	7	746	...
9	7	803	...
10	9	688	...
11	11	807	...
12	13	512	...
13	13	790	...
14	14	897	...
15	15	573	...
16	18	683	...
17	18	728	...
18	20	644	...
19	20	716	...
20	20	783	...
21	22	680	...

נקשר את העמודה הזאת למפתח של טבלת המתנדבים שיהיה המפתח זר שלו:

```
ALTER TABLE belongs ADD CONSTRAINT fk_belongs_volunteer FOREIGN KEY
(volunteer_id) REFERENCES volunteer(volunteer_id);
```

```
SELECT constraint_name
FROM user_constraints
WHERE table_name = 'BELONGS' AND constraint_type = 'R';
```

	CONSTRAINT_NAME	
1	SYS_C009042	...
2	FK_BELONGS_VOLUNTEER	...

נכניס נתונים לטבלה :

```
UPDATE belongs
SET volunteer_id = 11110 + ROWNUM
WHERE ROWNUM <= 400;
```

```
select * from belongs;
```

	G_ID	P_ID	VOLUNTEER_ID	
1	1	575	11111	...
2	1	631	11112	...
3	3	523	11113	...
4	3	685	11114	...
5	3	756	11115	...
6	6	581	11116	...
7	6	768	11117	...
8	7	746	11118	...
9	7	803	11119	...
10	9	688	11120	...
11	11	807	11121	...
12	13	512	11122	...
13	13	790	11123	...
14	14	897	11124	...
15	15	573	11125	...
16	18	683	11126	...
17	18	728	11127	...
18	20	644	11128	...
19	20	716	11129	...
20	20	783	11130	...
21	22	680	11131	...

נמחק את הגבלה של מפתח:

```
ALTER TABLE belongs
DROP PRIMARY KEY;
```

נמחק את העמודה p\_id:

```
ALTER TABLE belongs DROP COLUMN p_id;
```

```
select * from belongs;
```

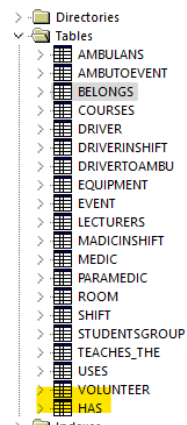
	G_ID	VOLUNTEER_ID	
1	1	11111	...
2	1	11112	...
3	3	11113	...
4	3	11114	...
5	3	11115	...
6	6	11116	...
7	6	11117	...

ניצור מפתח ראשי חדש:

```
ALTER TABLE belongs
ADD CONSTRAINT fk_belongs PRIMARY KEY (g_id, volunteer_id);
```

יצירת טבלת קשר בין אמבולנס לציוד:

```
CREATE TABLE HAS
(
  ambulance_id INTEGER NOT NULL,
  e_id NUMBER(10) NOT NULL,
  PRIMARY KEY (ambulance_id, e_id),
  FOREIGN KEY (ambulance_id) REFERENCES AMBULANS (ambulance_id),
  FOREIGN KEY (e_id) REFERENCES Equipment (e_id)
);
```



הכנסת נתונים לטבלה ובחירה בציוד שקשור לאמבולנס (יש ציוד ששייך לקורסים):

```
INSERT INTO HAS (ambulance_id, e_id)
SELECT ambulance_id, e_id
FROM (
  SELECT a.ambulance_id, e.e_id
  FROM AMBULANS a, Equipment e
  WHERE e.e_name IN ('Defibrillator', 'Stethoscope', 'Blood Pressure
Monitor', 'Oxygen Tank', 'First Aid Kit')
  AND ROWNUM <= 400
);
```

select \* from has;

	AMBULANCE_ID	E_ID
1	1	220
2	2	220
3	3	220
4	4	220
5	5	220
6	6	220
7	7	220
8	8	220
9	9	220
10	10	220
11	11111	220
12	11112	220
13	11113	220
14	11114	220
15	11115	220
16	11116	220

# VIEWS

## מבט 1:

המבט מכיל מידע משולב על קבוצות סטודנטים, המרצים שלהם, הקורסים שהם לומדים והמיקום של השיעורים:

```
CREATE VIEW view_groups_courses_lecturers AS
SELECT
    sg.g_id AS group_id,
    sg.g_hour AS group_hour,
    sg.g_day AS group_day,
    l.l_name AS lecturer_name,
    c.C_name AS course_name,
    r.r_location AS room_location
FROM
    StudentsGroup sg
JOIN
    lecturers l ON sg.l_id = l.l_id
JOIN
    Courses c ON sg.c_id = c.c_id
JOIN
    Room r ON sg.r_id = r.r_id;
```

select \* from view\_groups\_courses\_lecturers;

	GROUP_ID	GROUP_HOUR	GROUP_DAY	LECTURER_NAME	COURSE_NAME	ROOM_LOCATION
1	10	12	5	Nils Heche	Pediatric Care	Building D Floor 1
2	265	12	5	Wallace Heche	Basic First Aid	Building A Floor 4
3	201	11	4	Owen Bryson	Health and Safety	Building B Floor 4
4	142	13	4	Owen Bryson	Infection Control	Building B Floor 4
5	110	18	1	Owen Bryson	Health and Safety	Building C Floor 4
6	108	11	4	Owen Bryson	Pediatric Care	Building D Floor 4
7	182	11	5	Jose Bonneville	Infection Control	Building C Floor 5
8	317	13	1	Sophie Rivers	Trauma Care	Building B Floor 3
9	143	10	3	Taye Bloch	Pediatric Care	Building C Floor 3
10	111	17	3	Freddie Shand	Pediatric Care	Building C Floor 3
11	338	15	3	Vin Pony	Disaster Management	Building A Floor 4
12	234	15	3	Alfie Pullman	Advanced Life Support	Building C Floor 2
13	101	12	2	Alfie Pullman	Emergency Response	Building A Floor 1
14	37	16	2	Alfie Pullman	Basic First Aid	Building B Floor 4
15	347	20	5	Hank Hurt	Basic First Aid	Building C Floor 2
16	271	11	1	Hank Hurt	Emergency Response	Building A Floor 3

## שאלתה 1:

כל הקבוצות שהמרצה Owen Bryson מלמד, כולל מספר מזהה של קבוצה, שם המרצה ושם הקורס:

```
SELECT group_id,lecturer_name,course_name
FROM view_groups_courses_lecturers
WHERE lecturer_name = 'Owen Bryson';
```

	GROUP_ID	LECTURER_NAME	COURSE_NAME
1	108	Owen Bryson	Pediatric Care
2	110	Owen Bryson	Health and Safety
3	142	Owen Bryson	Infection Control
4	201	Owen Bryson	Health and Safety



## שאלת 2:

כל הקבוצות שלומדות את הקורס Emergency Response:

```
SELECT *
FROM view_groups_courses_lecturers
WHERE course_name = 'Emergency Response';
```

	GROUP_ID	GROUP_HOUR	GROUP_DAY	LECTURER_NAME	COURSE_NAME	ROOM_LOCATION
1	228	9	3	Ray Mann	Emergency Response	Building E Floor 4
2	152	11	4	Nicolas Delta	Emergency Response	Building E Floor 4
3	198	11	2	Avenged Phoenix	Emergency Response	Building D Floor 2
4	130	16	2	Timothy Wright	Emergency Response	Building B Floor 5
5	164	17	5	Jamie Addy	Emergency Response	Building A Floor 5
6	376	17	1	Oliver Waite	Emergency Response	Building A Floor 5
7	371	8	2	Javon Porter	Emergency Response	Building D Floor 3
8	351	17	2	Henry Connors	Emergency Response	Building E Floor 2
9	166	17	2	Gerald Finn	Emergency Response	Building C Floor 3
10	342	15	4	Javon Ramirez	Emergency Response	Building C Floor 3
11	90	12	3	Mira Jay	Emergency Response	Building D Floor 2
12	261	12	1	Gerald Finn	Emergency Response	Building D Floor 2
13	106	13	2	Gladys McDowell	Emergency Response	Building A Floor 3
14	247	16	3	Cyndi Malkovich	Emergency Response	Building D Floor 3
15	328	10	3	Johnnie Rudd	Emergency Response	Building E Floor 3
16	271	13	1	Hank Hurt	Emergency Response	Building A Floor 3

## מבט 2:

המבט מכיל את פרטי המשמרות יחד עם פרטי המתנדבים רק עבור פרמדיקים שהם גם חובשים, כלומר אלה שמשתתפים בשעות המשמרות שהוגדרו בטבלה:

```
CREATE VIEW view_shifts_medics AS
SELECT
    s.shift_id AS shift_id,
    s.date AS shift_date,
    s.starttime AS shift_start_time,
    s.endtime AS shift_end_time,
    v.volunteer_id AS volunteer_id,
    v.name AS volunteer_name
FROM
    SHIFT s
JOIN
    PARAMEDIC p ON s.volunteer_id = p.volunteer_id
JOIN
    MEDIC m ON s.volunteer_id = m.volunteer_id
JOIN
    VOLUNTEER v ON s.volunteer_id = v.volunteer_id;
```

```
select * from view_shifts_medics;
```

	SHIFT_ID	SHIFT_DATE	SHIFT_START_TIME	SHIFT_END_TIME	VOLUNTEER_ID	VOLUNTEER_NAME
1	1	01/01/2024	08:00	16:00	1	John Doe
2	2	01/02/2024	08:00	16:00	2	Jane Smith
3	3	01/03/2024	08:00	16:00	3	Emily Jones
4	4	01/04/2024	08:00	16:00	4	Michael Brown
5	5	01/05/2024	08:00	16:00	5	Sarah Davis
6	6	01/06/2024	08:00	16:00	6	David Wilson
7	7	01/07/2024	08:00	16:00	7	Laura Martinez
8	8	01/08/2024	08:00	16:00	8	James Garcia
9	9	01/09/2024	08:00	16:00	9	Emma Anderson
10	10	01/10/2024	08:00	16:00	10	Daniel Taylor
11	11	20/06/2024	08:00	14:00	1	John Doe
12	12	20/06/2024	08:00	14:00	10	Daniel Taylor
13	13	22/06/2024	08:00	16:00	8	James Garcia
14	11111	16/06/2020	12:00	16:00	11138	PatriciaDiBiasio
15	11112	17/01/2023	12:00	16:00	11151	MariePayne
16	11114	01/08/2020	08:00	16:00	11185	VingHenstridge

## שאלתה 1:

מחזירה את כל המשמרות שהיו אחרי תאריך 01/02/2024, כולל מספר מזהה זמן התחלה זמן סיום ותאריך:

```
SELECT shift_id,shift_date,shift_start_time,shift_end_time
FROM view_shifts_medics
WHERE TO_DATE(shift_date, 'DD/MM/YYYY') > TO_DATE('01/02/2024', 'DD/MM/YYYY');
```

	SHIFT_ID	SHIFT_DATE	SHIFT_START_TIME	SHIFT_END_TIME
1	3	01/03/2024	08:00	16:00
2	4	01/04/2024	08:00	16:00
3	5	01/05/2024	08:00	16:00
4	6	01/06/2024	08:00	16:00
5	7	01/07/2024	08:00	16:00
6	8	01/08/2024	08:00	16:00
7	9	01/09/2024	08:00	16:00
8	10	01/10/2024	08:00	16:00
9	11	20/06/2024	08:00	14:00
10	12	20/06/2024	08:00	14:00
11	13	22/06/2024	08:00	16:00

## שאלתה 2:

מחזירה את כל המשמרות של פרמדיקים:

```
SELECT *
FROM view_shifts_medics
WHERE volunteer_id IN (
    SELECT volunteer_id
    FROM PARAMEDIC
);
```

	SHIFT_ID	SHIFT_DATE	SHIFT_START_TIME	SHIFT_END_TIME	VOLUNTEER_ID	VOLUNTEER_NAME
1	1	01/01/2024	08:00	16:00	1	John Doe
2	2	01/02/2024	08:00	16:00	2	Jane Smith
3	3	01/03/2024	08:00	16:00	3	Emily Jones
4	4	01/04/2024	08:00	16:00	4	Michael Brown
5	5	01/05/2024	08:00	16:00	5	Sarah Davis
6	6	01/06/2024	08:00	16:00	6	David Wilson
7	7	01/07/2024	08:00	16:00	7	Laura Martinez
8	8	01/08/2024	08:00	16:00	8	James Garcia
9	9	01/09/2024	08:00	16:00	9	Emma Anderson
10	10	01/10/2024	08:00	16:00	10	Daniel Taylor
11	11	20/06/2024	08:00	14:00	1	John Doe