
מיני פרויקט

בוסיסי נתונים

דו"ח מסכם

בנושא

יחידת יהלום (זה"ל)



מגישיים:
יואב בביוף
אביישי שחור

	תוכן
2.....	תיאור הארגון
4.....	תרשים ה-ERD
5.....	תרשים DSD
6.....	פירוט היחסות והקשרים
8.....	נרטול הסכמה
9.....	פקודות create table
11	הפעלה desc
13.....	פקודות drop tables
14.....	פקודות select all
15.....	יצירת מידע
15.....	שיטה 1 – Mockaroo –
22.....	שיטה 2 – python code –
32.....	שיטה 3 – data generator (PLSQL developer) –
37.....	גיבוי נתונים
38.....	החזרת נתונים מגיבוי

חולג א

תיאור הארגון

הארגון הוא היחידה הצבאית המובחרת **יחידה הנדרשת למשימות מיוחדות**, בראשי תיבות: **יהל"ם** (נוהga: **יהלום**). היחידה היא יחידה קרבית מובחרת וככזו היא יוצאת למבצעים מיוחדים, מוקצת לה רב צבא מיוחד, וכל חיל בה חבר בצוות מסוים.

חברי היחידה:

היחידה מורכבת מהיחידה הבסיסית **soldier (חיל)**, כאשר ישנו חיילים עם תפקידים מיוחדים.

Soldier (חיל):

היחידה הבסיסית של הארגון, הוא האדם ממנו מורכבים צוותים והוא יוצא למשימות, **רבנים צבאיים (military Rabbi), מפקדים (commander) ורופאים צבאיים (combat medic)** הם גם חיילים.

לכל **חיל יש** מספר זהה (id) המזזה אותו באופן חד ערכי, ותכונות נוספות של חיל כגון גונן תא裏ר לידה, שם, דת ודרגה. כל **חיל** הוא **חיל-ב-(in) soldier** 0 או יותר **צוותים (team)** אשר בכל אחד מהם יש לו **תפקיד מסוים (responsibility)**.

רב צבאי (military Rabbi):

חיל האחראי על הצבעון הדתי של החברה היהודית של היחידה, בנוסף למאפיינים הרגילים של חיל יש לו גם עדת (ethnicity), האם הוא דרשן (preacher), האם הוא פיטן (payytan), מגזר (sector) והאם הוא מוסמך לרבנות (certified Rabbi).

רופא צבאי (combat medic):

חיל האחראי על שמירת הבריאות של חברי **צוות**, הוא המטפל בהם במבצעים מיוחדים (אם הוא נמצא בהם) ואחראי לבריאותם. בנוסף למאפיינים הרגילים של חיל יש לו מאפיין של הסמכה (qualification) המתארת את הפעולות הרפואיות אותן הוא מוסמך לעשות, והכשרה (in training) המתארת האם הוא בתקופת הכשרה. כל **רופא צבאי** גם יכול **להשתתף (take part)** במבצע צבאי.

מפקד (commander):

חיל המפקד על **commands (commands on)** צוות ואלו גם מפקד על **(on) מבצע**, כאשר אז הוא יכול לחתול חלק בקרב (active in field), או לא. בנוסף למאפיינים הרגילים של חיל יש לו כינוי (nickname) אופציוני.

קבוצות ביחידה:

ביחידה ישנן קבוצות מיוחדות המאגדת חברי יחידה למטרת משותפת.

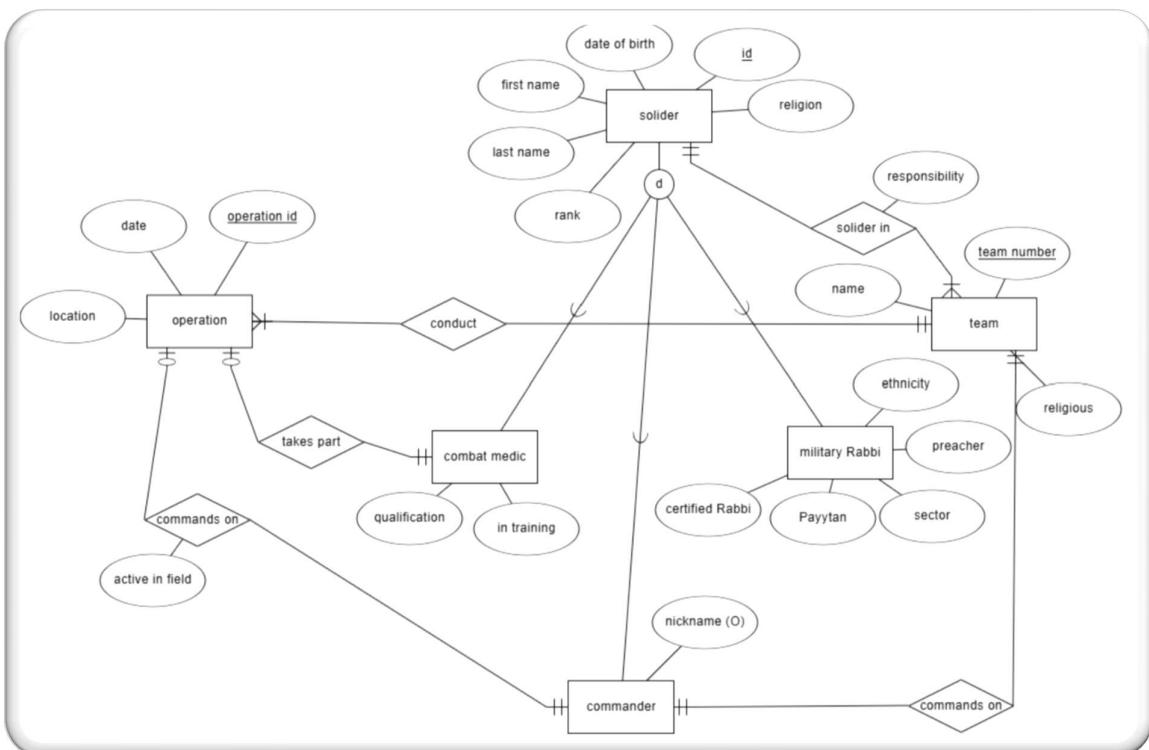
צוות (team):

היחידה הבסיסית המאגדת חברי צוות. לכל צוות בהכרח יש מפקד המפקד על **(commands on)** הצוות. כל צוות גם **מבצע (conducts)** מבצע. צוות מסוים מזוהה על פי מספר הצוות (team number).

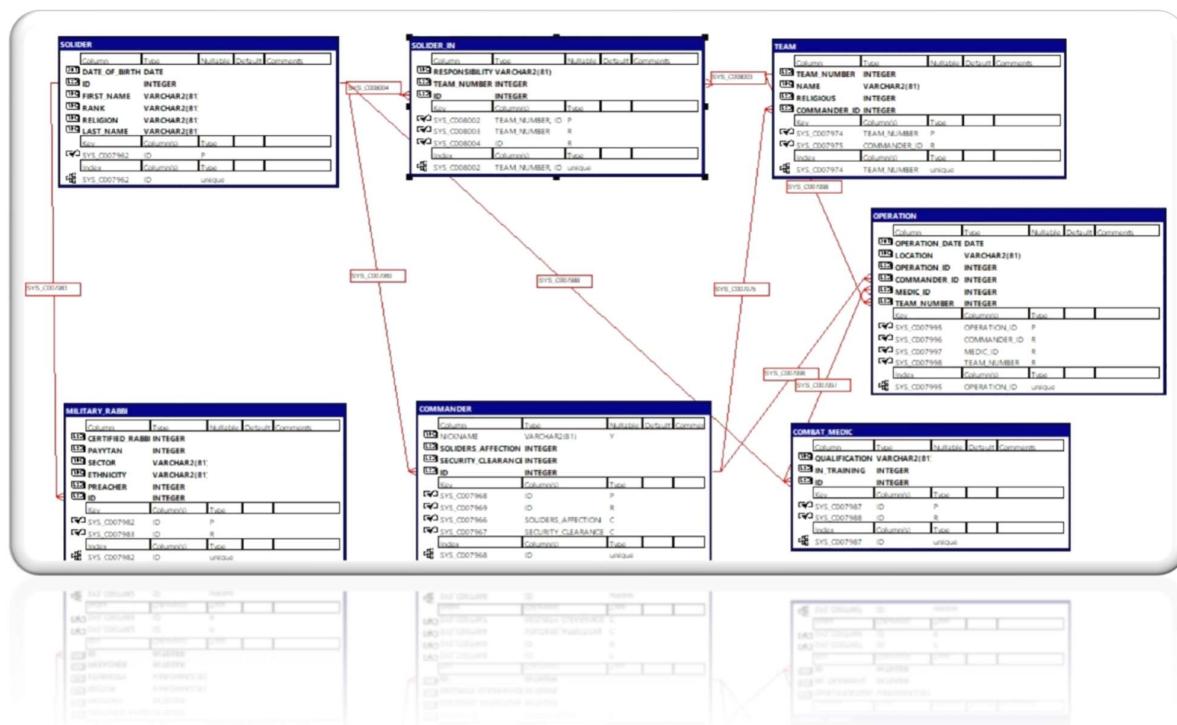
מבצע (operation):

ביצוע צבאי הממשש את היכולות אותן סיגלו לעצם **חילים**, בכל מבצע יכולים להשתתף כמה צוותים, ויש לו בהכרח **רופא צבאי** אחד הדואג לחילים המשתתפים בו ומפקד אחד המפקד עליהם. מבצע מזוהה לפי זהה המבצע (id operation), ויש לו תכונות נוספות כגון גונן תא裏ר המבצע ומיקומו.

תרשים ה-ERD



DSD תרשימים



פירוט הישויות והקשרים

חיל – Solider

<u>מזהה</u>	<u>Id</u>
תאריך לידה	Date of birth
שם פרטי	First name
דרגה	Rank
דת	Religion
שם משפחה	Last name

מפקד – commander

<u>מזהה</u>	<u>Id (fk)</u>
כינוי	Nickname (o)
ערצאות המפקד (ע"י החילים)	Solider affection
סיווג בטחוני	First name
מספר צוות	Rank

רב צבאי – military rabbi

<u>מזהה</u>	<u>Id (fk)</u>
האם הוא רב מוסמך	Certified rabbi
האם הרב פ'יטן	payytan
מגזר	sector
עדת	ethnicity
האם הרב דרשן	preacher

חובש קרב – combat medic

<u>מזהה</u>	<u>Id (fk)</u>
-------------	----------------

יואב בביוף
אבייש שchor

תחום התמחות	Qualification
האם עדין בהכשרה	In Training

צוות – team

<u>מספר צוות</u>	Team number
מזהה המפקד	Commander Id (fk)
שם הצוות	Name
האם הצוות הינו צוות דתי	Religious

מבצע צבאי – Operation

<u>מזהה מבצע</u>	Operation ID
תאריך בו בוצע המבצע	Date
מזהה מפקד	Commander Id (fk)
מזהה חובש קרב'	medic Id (fk)
מקום המבצע	Location
מספר צוות בו חבר החיל	Team Number (fk)

חיל בצוות – solider in

<u>מזהה החיל</u>	ID (fk)
מספר הצוות בו החיל נמצא	Team number (fk)
תפקיד החיל בצוות	responsibility

נרטוֹל הסכמה

הסכם מנוירמלת לפי NF3, שכן לכל הטבלאות שלנו, כל תכונה תלויה בכל התכונות שבמפתח, ורק בהן.

פקודות create table

```
-- Create table for soldiers
CREATE TABLE solider
(
    date_of_birth DATE NOT NULL,
    id INT NOT NULL,
    first_name VARCHAR(81) NOT NULL,
    rank VARCHAR(81) NOT NULL,
    religion VARCHAR(81) NOT NULL,
    last_name VARCHAR(81) NOT NULL,
    PRIMARY KEY (id)
);

-- Create table for commanders
CREATE TABLE commander
(
    nickname VARCHAR(81),
    soliders_affection INT NOT NULL, CHECK (soliders_affection >= 1 AND
soliders_affection <= 10),
    security_clearance INT NOT NULL, CHECK (security_clearance >= 1 AND
security_clearance <= 5),
    id INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id) REFERENCES solider(id)
);

-- Create table for teams
CREATE TABLE team
(
    team_number INT NOT NULL,
    name VARCHAR(81) NOT NULL,
    religious INT NOT NULL,
    commander_id INT NOT NULL,
    PRIMARY KEY (team_number),
    FOREIGN KEY (commander_id) REFERENCES commander(id)
);

-- Create table for military rabbis
CREATE TABLE military_Rabbi
(
    certified_Rabbi INT NOT NULL,
    Payttan INT NOT NULL,
    sector VARCHAR(81) NOT NULL,
    ethnicity VARCHAR(81) NOT NULL,
    preacher INT NOT NULL,
    id INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id) REFERENCES solider(id)
);

-- Create table for combat medics
CREATE TABLE combat_medic
(
    qualification VARCHAR(81) NOT NULL,
    in_training INT NOT NULL,
    id INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id) REFERENCES solider(id)
);
```

ואב בביוף
אבייש שchor

```
-- Create table for military operations
CREATE TABLE operation
(
    operation_date DATE NOT NULL,
    location VARCHAR(81) NOT NULL,
    operation_id INT NOT NULL,
    commander_id INT NOT NULL,
    medic_id INT NOT NULL,
    team_number INT NOT NULL,
    PRIMARY KEY (operation_id),
    FOREIGN KEY (commander_id) REFERENCES commander(id),
    FOREIGN KEY (medic_id) REFERENCES combat_medic(id),
    FOREIGN KEY (team_number) REFERENCES team(team_number)
);

-- create table for solider-team connection
CREATE TABLE solider_in
(
    responsibility VARCHAR(81) NOT NULL,
    team_number INT NOT NULL,
    id INT NOT NULL,
    PRIMARY KEY (team_number, id),
    FOREIGN KEY (team_number) REFERENCES team(team_number),
    FOREIGN KEY (id) REFERENCES solider(id)
);
```

הפעלה desc

נבדוק שהטבלאות נוצרו:

```
SQL> SELECT table_name FROM user_tables;  
  
TABLE_NAME  
-----  
SOLIDER  
COMMANDER  
TEAM  
MILITARY_RABBI  
COMBAT_MEDIC  
OPERATION  
SOLIDER_IN  
  
7 rows selected
```

יכן הן נוצרו.

נפעיל את פקודות DESC[RE]BIE על מנת לוודא שהטבלאות נוצרו כראוי:

```
SQL> desc military_rabbi  
Name          Type      Nullable Default Comments  
-----  
CERTIFIED_RABBI  INTEGER  
PAYYTAN        INTEGER  
SECTOR          VARCHAR2(81)  
ETHNICITY        VARCHAR2(81)  
PREACHER         INTEGER  
ID              INTEGER  
  
SQL> desc team  
Name          Type      Nullable Default Comments  
-----  
TEAM_NUMBER    INTEGER  
NAME           VARCHAR2(81)  
RELIGIOUS       INTEGER  
COMMANDER_ID   INTEGER  
  
SQL> desc operation  
Name          Type      Nullable Default Comments  
-----  
OPERATION_DATE DATE  
LOCATION        VARCHAR2(81)  
OPERATION_ID    INTEGER  
COMMANDER_ID    INTEGER  
MEDIC_ID        INTEGER  
TEAM_NUMBER     INTEGER  
  
SQL> desc solider_in  
Name          Type      Nullable Default Comments  
-----  
RESPONSIBILITY VARCHAR2(81)  
TEAM_NUMBER    INTEGER  
ID             INTEGER
```

```
SQL> desc solider
Name          Type           Nullable Default Comments
-----
DATE_OF_BIRTH DATE
ID            INTEGER
FIRST_NAME    VARCHAR2(81)
RANK          VARCHAR2(81)
RELIGION      VARCHAR2(81)
LAST_NAME     VARCHAR2(81)

SQL> desc commander
Name          Type           Nullable Default Comments
-----
NICKNAME      VARCHAR2(81)  Y
SOLDIERS_AFFECTION INTEGER
SECURITY_CLEARANCE INTEGER
ID            INTEGER

SQL> desc combat_medic
Name          Type           Nullable Default Comments
-----
QUALIFICATION VARCHAR2(81)
IN_TRAINING   INTEGER
ID            INTEGER
```

ואכן כל הattribut נוצרו כנדרש.

פקודות drop tables

```
-- Drop the combat_medic table
DROP TABLE combat_medic CASCADE CONSTRAINTS;

-- Drop the military_rabbi table
DROP TABLE military_rabbi CASCADE CONSTRAINTS;

-- Drop the operation table
DROP TABLE operation CASCADE CONSTRAINTS;

-- Drop the team table
DROP TABLE team CASCADE CONSTRAINTS;

-- Drop the commander table
DROP TABLE commander CASCADE CONSTRAINTS;

-- Drop the solider table
DROP TABLE solider CASCADE CONSTRAINTS;

-- Drop the solider-in table
DROP TABLE solider_in CASCADE CONSTRAINTS;
```

פקודות select all

```
-- Select all records from the solider table
SELECT * FROM solider;

-- Select all records from the commander table
SELECT * FROM commander;

-- Select all records from the team table
SELECT * FROM team;

-- Select all records from the military_Rabbi table
SELECT * FROM military_Rabbi;

-- Select all records from the combat_medic table
SELECT * FROM combat_medic;

-- Select all records from the operation table
SELECT * FROM operation;

-- Select all records from the solider-n table
SELECT * FROM solider_in;
```

יצירת מידע

Mockaroo – 1 שיטה

Military Rabii:

Field Name	Type	Options
certified_Rabbi	Number	min: 0 max: 1 decimals: 0 blank: 0 % Σ X
Payytan	Number	min: 0 max: 1 decimals: 0 blank: 0 % Σ X
sector	Custom List	Religious Zionism, Haredi-Litai, Haredi-Hasidic  random  blank: 0 % Σ X
ethnicity	Custom List	Ashkenazi, Sephard, Edot-Mizrah  random  blank: 0 % Σ X
preacher	Number	min: 0 max: 1 decimals: 0 blank: 0 % Σ X
id	Number	min: 10000000 max: 999999999 decimals: 0 blank: 0 % Σ X

Combat Medic:

Field Name	Type	Options
qualification	Custom List	MD, DO, DDS, DMD, RN, NP, PA, PharmD, DPT, DC  random  blank: 0 % Σ X
in_training	Number	min: 0 max: 1 decimals: 0 blank: 0 % Σ X
id	Row Number	blank: 0 % Σ X

Commander:

Field Name	Type	Options
nickname	First Name	blank: 0 % Σ X
soldiers_affection	Number	min: 1 max: 10 decimals: 0 blank: 0 % Σ X
security_clearance	Number	min: 1 max: 5 decimals: 0 blank: 0 % Σ X
id	Row Number	blank: 0 % Σ X

Team:

Field Name	Type	Options
team_number	Row Number	blank: 0 % Σ X
name	Animal Scientific Na...	blank: 0 % Σ X
religious	Number	min: 0 max: 1 decimals: 0 blank: 0 % Σ X
commander_id	Row Number	blank: 0 % Σ X

Operation:

ויאב בביוף

אכיש שחר

Field Name	Type	Options
operation_date	Datetime	01/01/2000 <input type="button" value="Calendar"/> to 12/31/2022 <input type="button" value="Calendar"/> format: m/d/yyyy blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
location	Street Address	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
operation_id	Row Number	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
commander_id	Row Number	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
medic_id	Row Number	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
team_number	Row Number	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>

Solider:

Field Name	Type	Options
date_of_birth	Datetime	01/01/1980 <input type="button" value="Calendar"/> to 12/31/2022 <input type="button" value="Calendar"/> format: m/d/yyyy blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
id	Row Number	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
first_name	First Name	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
rank	Custom List	Private, Corporal, Sergeant, Lieutenant, Captain, Major, Colonel, General <input type="button" value="random"/> blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
religion	Custom List	Jewish, Christian, Muslim, Hindu, Buddhist, Druze, Bahai, Sikh, Other <input type="button" value="random"/> blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
last_name	Last Name	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>

Solider-In:

Field Name	Type	Options
responsibility	Custom List	sniper, scout, demolition expert, communications specialist, heavy gunner, engineer, reconnaissance, <input type="button" value="random"/> random blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
team_number	Row Number	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
id	Number	min: 1 max: 1200 decimals: 0 blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>

Generated SQL code can be found in the GitHub directory.

Here is a sample of the generated SQL code and it's result:

```
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1998-02-09',1,'Madelene','Corporal','Druze','McDougal');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2011-3-4',2,'Kerwin','Lieutenant','Zoroastrian','Cordingly');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1997-4-6',3,'Blakeley','Lieutenant','Muslim','Alecock');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2003-2-02',4,'Domingo','General','Jewish','Wildgoose');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1998-9-06',5,'Lori','Captain','Zoroastrian','Slorance');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1986-9-9',6,'Trixie','Sergeant','Christian','Pettaward');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2015-6-05',7,'Paco','Colonel','Zoroastrian','Campelli');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1985-1-01',8,'Rochelle','Colonel','Jewish','Shotter');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1999-02-10',9,'Kile','Private','Hindu','Bebis');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2008-11-11',10,'Hodge','Private','Hindu','Upton');

INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1989-06-08',401,'Fallon','Lieutenant','Sikh','Mullard');
```

יאוב בביוף**אבייש שוחר**

```

INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2009-07-03',402,'Ed','Sergeant','Sikh','Kesteven');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2020-03-
07',403,'Birgitta','Corporal','Zoroastrian','Frediani');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1997-05-06',404,'Martie','Private','Jewish','Malia');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2005-8-9',405,'Townie','Colonel','Christian','Bromby');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2015-4-5',406,'Heindrick','Colonel','Sikh','Lett');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2016-7-4',407,'Lianna','Captain','Sikh','Jouannot');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1993-01-09',408,'Iggie','Private','Sikh','Kelf');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2012-04-12',409,'Andra','Captain','Druze','Gascoigne');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2017-2-12',410,'Theressa','General','Druze','Janic');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1999-08-05',411,'Bertrando','Major','Sikh','Murfill');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1996-01-04',412,'Alberto','Corporal','Muslim','Russilll');

INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2008-10-12',801,'Kellina','Corporal','Zoroastrian','Storer');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1996-02-03',802,'Adah','Corporal','Muslim','Rubes');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2022-1-7',803,'Hercules','General','Sikh','Ferri');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1991-07-05',804,'Farris','Private','Christian','Steel');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2014-2-1',805,'Fancie','Lieutenant','Jewish','Fielder');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1986-8-6',806,'Emalia','Private','Sikh','Wheatland');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2017-2-4',807,'Karil','Private','Druze','Ravillas');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1985-12-10',808,'Arty','Lieutenant','Sikh','Pinney');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '1992-2-8',809,'Marion','Colonel','Jewish','MacCrann');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2019-5-12',810,'Justin','Lieutenant','Zoroastrian','Gillet');
INSERT INTO solider (date_of_birth,id,first_name,rank,religion,last_name)
VALUES (DATE '2015-1-3',811,'Jose','Major','Muslim','Mellsop');

insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (1,1,'Haredi-Litai','Ashkenazi',1,1);
insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (0,1,'Haredi-Litai','Sphard',1,2);
insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (0,1,'Haredi-Hasidic','Ashkenazi',1,3);
insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (0,1,'Haredi-Hasidic','Edot-Mizrah',1,4);
insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (0,1,'Religious Zionism','Sphard',0,5);
insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (1,0,'Haredi-Hasidic','Sphard',0,6);
insert into military_rabbi (certified_Rabbi, Paytan, sector, ethnicity,
preacher, id) values (1,0,'Religious Zionism','Edot-Mizrah',1,7);

```

```

insert into combat_medic (qualification,in_training,id) values
('RN',1,401);
insert into combat_medic (qualification,in_training,id) values
('DPT',1,402);
insert into combat_medic (qualification,in_training,id) values
('DPT',1,403);
insert into combat_medic (qualification,in_training,id) values
('MD',0,404);
insert into combat_medic (qualification,in_training,id) values
('NP',1,405);
insert into combat_medic (qualification,in_training,id) values
('DMD',1,406);

insert into commander (nickname, soliders_affection, security_clearance,
id) values ('benchmark', 9, 4, 801);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('firmware', 6, 5, 802);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('4th generation', 10, 3, 803);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('knowledge user', 7, 4, 804);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('artificial intelligence', 1, 3, 805);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('user-facing', 5, 5, 806);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('user-facing', 5, 5, 807);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('user-facing', 5, 5, 808);
insert into commander (nickname, soliders_affection, security_clearance,
id) values ('user-facing', 5, 5, 809);

insert into team (team_number, name, religious, commander_id) values (1,
'Stenella coeruleoalba', 1, 801);
insert into team (team_number, name, religious, commander_id) values (2,
'Neophron percnopterus', 1, 802);
insert into team (team_number, name, religious, commander_id) values (3,
'Ardea golieth', 0, 803);
insert into team (team_number, name, religious, commander_id) values (4,
'Cyrtodactylus louisianensis', 0, 804);
insert into team (team_number, name, religious, commander_id) values (5,
'Eremophila alpestris', 1, 805);
insert into team (team_number, name, religious, commander_id) values (6,
'Vanessa indica', 1, 806);
insert into team (team_number, name, religious, commander_id) values (7,
'Sus scrofa', 0, 807);

insert into operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) values (DATE '2015-1-
3','China',2358239,801,401,1);
insert into operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) values (DATE '2016-2-
10','Brazil',2358240,802,402,2);
insert into operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) values (DATE '2003-1-
10','Russia',2358241,803,403,3);
insert into operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) values (DATE '2010-4-
12','China',2358242,804,404,4);

```

ואב בביוף

אבייש שחר

```

insert into operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) values (DATE '2007-5-
3','Japan',2358243,805,405,5);
insert into operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) values (DATE '2003-6-
2','Brazil',2358244,806,406,6);

insert into solider_in (responsibility, team_number, id) values
('communications specialist', 1, 402);
insert into solider_in (responsibility, team_number, id) values
('engineer', 2, 804);
insert into solider_in (responsibility, team_number, id) values
('marksman', 3, 7);
insert into solider_in (responsibility, team_number, id) values ('heavy
gunner', 4, 807);
insert into solider_in (responsibility, team_number, id) values ('heavy
gunner', 5, 404);

commit;

```

processing select-all:

SOLIDER-IN:

	RESPONSIBILITY	TEAM_NUMBER	ID
1	communications specialist	1	402
2	engineer	2	804
3	marksman	3	7
4	heavy gunner	4	807
5	heavy qunner	5	404

SOLIDER:

	DATE_OF_BIRTH	ID	FIRST_NAME	RANK	RELIGION	LAST_NAME
1	09/02/1998	1	Madelene	Corporal	Druze	McDougal
2	04/03/2011	2	Kerwin	Lieutenant	Zoroastrian	Cordingly
3	06/04/1997	3	Blakeley	Lieutenant	Muslim	Alecock
4	02/02/2003	4	Domingo	General	Jewish	Wildgoose
5	06/09/1998	5	Lori	Captain	Zoroastrian	Slorance
6	09/09/1986	6	Trixie	Sergeant	Christian	Pettward
7	05/06/2015	7	Paco	Colonel	Zoroastrian	Campelli
8	01/01/1985	8	Rochelle	Colonel	Jewish	Shotter
9	10/02/1999	9	Kile	Private	Hindu	Bebis
10	11/11/2008	10	Hodge	Private	Hindu	Upton
11	08/06/1989	401	Fallon	Lieutenant	Sikh	Mullard
12	03/07/2009	402	Ed	Sergeant	Sikh	Kesteven
13	07/03/2020	403	Birgitta	Corporal	Zoroastrian	Frediani
14	06/05/1997	404	Martie	Private	Jewish	Malia

ואב בביוף

אבי שchor

COMMANDER:

	NICKNAME	SOLDERS_AFFECTION	SECURITY_CLEARANCE	ID
1	benchmark	9	4	801
2	firmware	6	5	802
3	4th generation	10	3	803
4	knowledge user	7	4	804
5	artificial intelligence	1	3	805
6	user-facing	5	5	806
7	user-facing	5	5	807
8	user-facing	5	5	808
9	user-facing	5	5	809

TEAM:

	TEAM_NUMBER	NAME	RELIGIOUS	COMMANDER_ID
1	1	Stenella coeruleoalba	1	801
2	2	Neophron percnopterus	1	802
3	3	Ardea goliath	0	803
4	4	Cyrtodactylus louisiadensis	0	804
5	5	Eremophila alpestris	1	805
6	6	Vanessa indica	1	806
7	7	Sus scrofa	0	807

MILITARY RABBI:

	CERTIFIED_RABBI	PAYYTAN	SECTOR	ETHNICITY	PREACHER	ID
1	1	1	Haredi-Litai	Ashkenazi	1	1
2	0	1	Haredi-Litai	Sphard	1	2
3	0	1	Haredi-Hasidic	Ashkenazi	1	3
4	0	1	Haredi-Hasidic	Edot-Mizrah	1	4
5	0	1	Religious Zionism	Sphard	0	5
6	1	0	Haredi-Hasidic	Sphard	0	6
7	1	0	Reliaious Zionism	Edot-Mizrah	1	7

COMBAT MEDIC:

	QUALIFICATION	IN_TRAINING	ID
1	RN	1	401
2	DPT	1	402
3	DPT	1	403
4	MD	0	404
5	NP	1	405
6	DMD	1	406

בש"ד
מini-פרויקט בסיסי נתונים

ואב בביוף

אבייש שchor

OPERATION:

	OPERATION_DATE	LOCATION	OPERATION_ID	COMMANDER_ID	MEDIC_ID	TEAM_NUMBER
1	03/01/2015	▼ China	... 2358239	801	401	1
2	10/02/2016	▼ Brazil	... 2358240	802	402	2
3	10/01/2003	▼ Russia	... 2358241	803	403	3
4	12/04/2010	▼ China	... 2358242	804	404	4
5	03/05/2007	▼ Japan	... 2358243	805	405	5
6	02/06/2003	▼ Brazil	... 2358244	806	406	6

שיטה 2 – python code

הקוד הבא יוצר קובץ sql אשר בו פקודות insert של נתונים המיצרים אקרואית לכל טבלה:

```
import random
import string
import datetime

# Define possible values for VARCHAR fields
possible_values = {
    'first_name': ['John', 'Jane', 'Alice', 'Bob', 'Charlie'],
    'last_name': ['Smith', 'Doe', 'Johnson', 'Williams', 'Brown'],
    'rank': ['Private', 'Sergeant', 'Lieutenant', 'Captain', 'Major'],
    'religion': ['Christianity', 'Judaism', 'Islam', 'Hinduism',
    'Buddhism'],
    'nickname': ['Eagle', 'Tiger', 'Wolf', 'Bear', 'Lion'],
    'sector': ['North', 'South', 'East', 'West', 'Central'],
    'ethnicity': ['Caucasian', 'African American', 'Asian', 'Hispanic',
    'Other'],
    'qualification': ['Basic', 'Advanced', 'Expert'],
    'location': ['Base A', 'Base B', 'Base C', 'Base D', 'Base E'],
    'team_name': ['Alpha', 'Bravo', 'Charlie', 'Delta', 'Echo'],
    'responsibility': ['sniper', 'scout', 'demolition expert',
    'communications specialist', 'heavy gunner', 'engineer',
    'reconnaissance', 'marksman', 'grenadier'],
    'sharpshooter']
}

def generate_random_string(length=10):
    return ''.join(random.choices(string.ascii_letters + string.digits,
k=length))

def generate_date_help():
    start_date = datetime.date(1980, 1, 1)
    end_date = datetime.date(2000, 12, 31)
    return start_date + (end_date - start_date) * random.random()

def generate_date():
    date = "DATE " + generate_date_help().strftime("%Y-%m-%d") + " "
    return date

def generate_sql_file(num_soldiers, num_commanders, num_teams, num_rabbis,
num_medics, num_operations, num_soliders_in):
    soldiers = []
    commanders = []
    teams = []
    military_rabbis = []
    combat_medics = []
    operations = []
    soliders_in = []

    for i in range(1, num_soldiers + 1):
        soldier = {
            'date_of_birth': generate_date(),
            'id': i,
```

יאוב בביוף
אבייש שחר

```
'first_name': random.choice(possible_values['first_name']),
'rank': random.choice(possible_values['rank']),
'religion': random.choice(possible_values['religion']),
'last_name': random.choice(possible_values['last_name']),
'team_number': random.randint(1, num_teams)
}
soldiers.append(soldier)

for i in range(1, num_commanders + 1):
    commander = {
        'nickname': random.choice(possible_values['nickname']),
        'soliders_affection': random.randint(1, 10),
        'security_clearance': random.randint(1, 5),
        'id': soldiers[i % num_soldiers]['id']
    }
    commanders.append(commander)

for i in range(1, num_teams + 1):
    team = {
        'team_number': i,
        'name': random.choice(possible_values['team_name']),
        'religious': random.randint(0, 1),
        'commander_id': commanders[i % num_commanders]['id']
    }
    teams.append(team)

for i in range(1, num_rabbis + 1):
    rabbi = {
        'certified_Rabbi': random.randint(0, 1),
        'Payytan': random.randint(0, 1),
        'sector': random.choice(possible_values['sector']),
        'ethnicity': random.choice(possible_values['ethnicity']),
        'preacher': random.randint(0, 1),
        'id': soldiers[i % num_soldiers]['id']
    }
    military_rabbis.append(rabbi)

for i in range(1, num_medics + 1):
    medic = {
        'qualification':
random.choice(possible_values['qualification']),
        'in_training': random.randint(0, 1),
        'id': soldiers[i % num_soldiers]['id']
    }
    combat_medics.append(medic)

for i in range(1, num_operations + 1):
    operation = {
        'operation_date': generate_date(),
        'location': random.choice(possible_values['location']),
        'operation_id': i,
        'commander_id': commanders[i % num_commanders]['id'],
        'medic_id': combat_medics[i % num_medics]['id'],
        'team_number': teams[i % num_teams]['team_number']
    }
    operations.append(operation)

for i in range(1, num_soliders_in + 1):
    solider_in = {
        'responsibility':
random.choice(possible_values['responsibility']),
        'team_number': teams[i % num_teams]['team_number'],
        'id': soldiers[i % num_soldiers]['id']
    }
    soliders_in.append(solider_in)
```

ואב בביוף

אביישי שחור

```

        'id': soldiers[i % num_soldiers]['id'],
        'team_number': teams[i % num_teams]['team_number']
    }
    soliders_in.append(solider_in)

    # Create SQL file
    with open('insertTables.sql', 'w') as f:
        # Insert soldiers
        for soldier in soldiers:
            f.write(
                f"INSERT INTO solider (date_of_birth, id, first_name, rank,
religion, last_name) VALUES ({soldier['date_of_birth']}, {soldier['id']},
'{soldier['first_name']}', '{soldier['rank']}', '{soldier['religion']}',
'{soldier['last_name']}");\n")

        # Insert commanders
        for commander in commanders:
            f.write(
                f"INSERT INTO commander (nickname, soliders_affection,
security_clearance, id) VALUES ('{commander['nickname']}',
{commander['soliders_affection']}, {commander['security_clearance']},
{commander['id']});\n")

        # Insert teams
        for team in teams:
            f.write(
                f"INSERT INTO team (team_number, name, religious,
commander_id) VALUES ({team['team_number']}, '{team['name']}',
{team['religious']}, {team['commander_id']});\n")

        # Insert military rabbis
        for rabbi in military_rabbis:
            f.write(
                f"INSERT INTO military_Rabbi (certified_Rabbi, Payytan,
sector, ethnicity, preacher, id) VALUES ({rabbi['certified_Rabbi']},
{rabbi['Payytan']}, '{rabbi['sector']}', '{rabbi['ethnicity']}',
{rabbi['preacher']}, {rabbi['id']});\n")

        # Insert combat medics
        for medic in combat_medics:
            f.write(
                f"INSERT INTO combat_medic (qualification, in_training, id)
VALUES ('{medic['qualification']}', {medic['in_training']},
{medic['id']});\n")

        # Insert operations
        for operation in operations:
            f.write(
                f"INSERT INTO operation (operation_date, location,
operation_id, commander_id, medic_id, team_number) VALUES
({operation['operation_date']}), '{operation['location']}',
{operation['operation_id']}, {operation['commander_id']},
{operation['medic_id']}, {operation['team_number']});\n")

        # Insert solider-team connections
        for solider_in in soliders_in:
            f.write(
                f"INSERT INTO solider_in (responsibility, id, team_number)
VALUES ('{solider_in['responsibility']}', '{solider_in['id']}',
{solider_in['team_number']});\n")

```

ואב בביוף
אבייש שוחר

```
if __name__ == '__main__':
    # Specify the number of entries for each table
    num_soldiers = 1200
    num_commanders = 400
    num_teams = 400
    num_rabbis = 400
    num_medics = 400
    num_operations = 400
    num_soliders_in = 400

    generate_sql_file(num_soldiers, num_commanders, num_teams, num_rabbis,
num_medics, num_operations, num_soliders_in)
```

דוגמת קוד עבור כמה כניסה נموנאה:

```
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1984-08-09', 1, 'Charlie', 'Lieutenant',
'Buddhism', 'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1996-08-28', 2, 'Bob', 'Sergeant', 'Buddhism',
'Doe');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1985-06-05', 3, 'John', 'Sergeant', 'Hinduism',
'Smith');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '2000-07-01', 4, 'Bob', 'Lieutenant', 'Buddhism',
'Williams');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1988-06-05', 5, 'John', 'Captain', 'Judaism',
'Smith');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1985-05-15', 6, 'John', 'Captain', 'Hinduism',
'Johnson');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1989-08-25', 7, 'John', 'Sergeant', 'Judaism',
'Smith');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1988-12-07', 8, 'Alice', 'Sergeant', 'Islam',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1997-06-29', 9, 'Charlie', 'Captain', 'Judaism',
'Doe');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1993-05-26', 10, 'John', 'Private', 'Hinduism',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1983-04-05', 11, 'Alice', 'Private', 'Judaism',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1982-11-30', 12, 'Jane', 'Sergeant', 'Judaism',
'Williams');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1981-06-17', 13, 'Bob', 'Sergeant',
'Christianity', 'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1987-11-17', 14, 'John', 'Sergeant', 'Judaism',
'Smith');
```

ויאב בביוף

אבייש שחר

```

INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1995-12-01', 15, 'Charlie', 'Private', 'Judaism',
'Smith');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1985-02-18', 16, 'Alice', 'Lieutenant', 'Islam',
'Williams');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1981-09-13', 17, 'John', 'Major', 'Judaism',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1986-12-31', 18, 'Jane', 'Private', 'Buddhism',
'Doe');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1987-06-29', 19, 'Charlie', 'Sergeant', 'Judaism',
'Johnson');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '2000-12-23', 20, 'Charlie', 'Private', 'Hinduism',
'Williams');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1994-02-27', 21, 'Bob', 'Sergeant', 'Islam',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1992-11-17', 22, 'Jane', 'Sergeant', 'Islam',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1985-07-23', 23, 'Bob', 'Sergeant', 'Judaism',
'Brown');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1994-05-15', 24, 'Charlie', 'Captain',
'Christianity', 'Doe');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1997-12-26', 25, 'Charlie', 'Sergeant',
'Buddhism', 'Williams');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1985-09-11', 26, 'Bob', 'Private', 'Buddhism',
'Smith');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1995-06-09', 27, 'Alice', 'Captain', 'Judaism',
'Williams');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1987-01-08', 28, 'Bob', 'Lieutenant',
'Christianity', 'Johnson');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1982-05-24', 29, 'Charlie', 'Sergeant',
'Christianity', 'Smith');
INSERT INTO solider (date_of_birth, id, first_name, rank, religion,
last_name) VALUES (DATE '1991-01-27', 30, 'Jane', 'Lieutenant', 'Islam',
'Williams');
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Eagle', 7, 2, 2);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Bear', 7, 2, 3);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Tiger', 3, 4, 4);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Wolf', 10, 4, 5);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Bear', 7, 5, 6);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Bear', 6, 1, 7);

```

ואב בביוף

אבייש שחר

```

INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Tiger', 9, 5, 8);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Tiger', 9, 1, 9);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Wolf', 8, 1, 10);
INSERT INTO commander (nickname, soliders_affection, security_clearance,
id) VALUES ('Lion', 1, 1, 11);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (1,
'Alpha', 0, 3);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (2,
'Bravo', 1, 4);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (3,
'Echo', 0, 5);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (4,
'Echo', 0, 6);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (5,
'Delta', 0, 7);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (6,
'Charlie', 0, 8);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (7,
'Charlie', 1, 9);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (8,
'Bravo', 1, 10);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (9,
'Bravo', 0, 11);
INSERT INTO team (team_number, name, religious, commander_id) VALUES (10,
'Echo', 1, 2);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (1, 0, 'South', 'African American', 0, 2);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (1, 0, 'Central', 'Asian', 1, 3);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (0, 0, 'East', 'Other', 0, 4);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (0, 1, 'West', 'African American', 1, 5);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (0, 1, 'West', 'Hispanic', 0, 6);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (0, 0, 'South', 'Caucasian', 1, 7);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (0, 0, 'West', 'African American', 1, 8);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (1, 0, 'West', 'Caucasian', 0, 9);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (1, 1, 'South', 'Caucasian', 1, 10);
INSERT INTO military_Rabbi (certified_Rabbi, Payytan, sector, ethnicity,
preacher, id) VALUES (0, 0, 'East', 'Hispanic', 0, 11);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Expert',
0, 2);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Basic',
0, 3);
INSERT INTO combat_medic (qualification, in_training, id) VALUES
('Advanced', 0, 4);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Basic',
0, 5);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Basic',
1, 6);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Expert',
0, 7);

```

יאוב בביוף**אבייש שחר**

```

INSERT INTO combat_medic (qualification, in_training, id) VALUES
('Advanced', 1, 8);
INSERT INTO combat_medic (qualification, in_training, id) VALUES
('Advanced', 0, 9);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Expert',
1, 10);
INSERT INTO combat_medic (qualification, in_training, id) VALUES ('Expert',
1, 11);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1981-09-30', 'Base B',
1, 3, 3, 2);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1992-12-17', 'Base C',
2, 4, 4, 3);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1995-10-24', 'Base B',
3, 5, 5, 4);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1995-09-16', 'Base E',
4, 6, 6, 5);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1993-03-17', 'Base B',
5, 7, 7, 6);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1987-01-11', 'Base E',
6, 8, 8, 7);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1985-11-18', 'Base A',
7, 9, 9, 8);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1992-08-29', 'Base C',
8, 10, 10, 9);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '2000-05-30', 'Base B',
9, 11, 11, 10);
INSERT INTO operation (operation_date, location, operation_id,
commander_id, medic_id, team_number) VALUES (DATE '1992-11-19', 'Base B',
10, 2, 2, 1);
INSERT INTO solider_in (responsibility, id, team_number) VALUES ('scout',
'2', 2);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('marksman', '3', 3);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('sharpshooter', '4', 4);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('engineer', '5', 5);
INSERT INTO solider_in (responsibility, id, team_number) VALUES ('scout',
'6', 6);
INSERT INTO solider_in (responsibility, id, team_number) VALUES ('scout',
'7', 7);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('reconnaissance', '8', 8);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('engineer', '9', 9);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('sharpshooter', '10', 10);
INSERT INTO solider_in (responsibility, id, team_number) VALUES
('demolition expert', '11', 1);
commit;

```

בש"ד

מינ-פרוייקט בסיסי נתונים

ויאב בביוף

אכיש שחר

והתוצאה לאחר all:

SOLIDER-IN:

	RESPONSIBILITY	TEAM_NUMBER	ID
1	scout	2	2
2	marksman	3	3
3	sharpshooter	4	4
4	engineer	5	5
5	scout	6	6
6	scout	7	7
7	reconnaissance	8	8
8	engineer	9	9
9	sharpshooter	10	10
10	demolition expert	11	11

SOLIDER:

	DATE_OF_BIRTH	ID	FIRST_NAME	RANK	RELIGION	LAST_NAME
1	09/08/1984	1	Charlie	Lieutenant	Buddhism	Brown
2	28/08/1996	2	Bob	Sergeant	Buddhism	Doe
3	05/06/1985	3	John	Sergeant	Hinduism	Smith
4	01/07/2000	4	Bob	Lieutenant	Buddhism	Williams
5	05/06/1988	5	John	Captain	Judaism	Smith
6	15/05/1985	6	John	Captain	Hinduism	Johnson
7	25/08/1989	7	John	Sergeant	Judaism	Smith
8	07/12/1988	8	Alice	Sergeant	Islam	Brown
9	29/06/1997	9	Charlie	Captain	Judaism	Doe
10	26/05/1993	10	John	Private	Hinduism	Brown
11	05/04/1983	11	Alice	Private	Judaism	Brown
12	30/11/1982	12	Jane	Sergeant	Judaism	Williams
13	17/06/1981	13	Bob	Sergeant	Christianity	Brown
14	17/11/1987	14	John	Sergeant	Judaism	Smith

COMMANDER:

	NICKNAME	SOLDERS_AFFECTION	SECURITY_CLEARANCE	ID
1	Eagle	7	2	2
2	Bear	7	2	3
3	Tiger	3	4	4
4	Wolf	10	4	5
5	Bear	7	5	6
6	Bear	6	1	7
7	Tiger	9	5	8
8	Tiger	9	1	9
9	Wolf	8	1	10
10	Lion	1	1	11

ואב בביוף

אבייש שחרור

TEAM:

	TEAM_NUMBER	NAME	RELIGIOUS	COMMANDER_ID
1	1	Alpha	0	3
2	2	Bravo	1	4
3	3	Echo	0	5
4	4	Echo	0	6
5	5	Delta	0	7
6	6	Charlie	0	8
7	7	Charlie	1	9
8	8	Bravo	1	10
9	9	Bravo	0	11
10	10	Echo	1	2

MILITARY RABBI:

	CERTIFIED_RABBI	PAYYTAN	SECTOR	ETHNICITY	PREACHER	ID
1	1	0	South	African American	0	2
2	1	0	Central	Asian	1	3
3	0	0	East	Other	0	4
4	0	1	West	African American	1	5
5	0	1	West	Hispanic	0	6
6	0	0	South	Caucasian	1	7
7	0	0	West	African American	1	8
8	1	0	West	Caucasian	0	9
9	1	1	South	Caucasian	1	10
10	0	0	East	Hispanic	0	11

COMBAT MEDIC:

	QUALIFICATION	IN_TRAINING	ID
1	Expert	0	2
2	Basic	0	3
3	Advanced	0	4
4	Basic	0	5
5	Basic	1	6
6	Expert	0	7
7	Advanced	1	8
8	Advanced	0	9
9	Expert	1	10
10	Expert	1	11

בש"ד
מINI-PROJECT בבסיסי נתונים

ואב בביוף

אבייש שחר

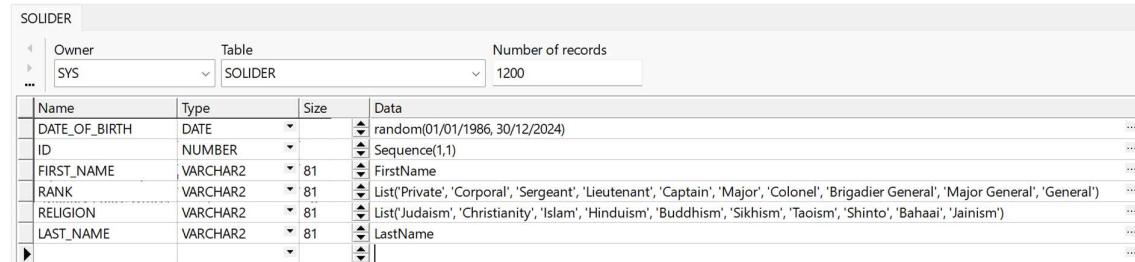
OPERATION:

	OPERATION_DATE	LOCATION	OPERATION_ID	COMMANDER_ID	MEDIC_ID	TEAM_NUMBER
1	30/09/1981	▼ Base B	...	1	3	3
2	17/12/1992	▼ Base C	...	2	4	4
3	24/10/1995	▼ Base B	...	3	5	5
4	16/09/1995	▼ Base E	...	4	6	6
5	17/03/1993	▼ Base B	...	5	7	7
6	11/01/1987	▼ Base E	...	6	8	8
7	18/11/1985	▼ Base A	...	7	9	9
8	29/08/1992	▼ Base C	...	8	10	10
9	30/05/2000	▼ Base B	...	9	11	11
10	19/11/1992	▼ Base B	...	10	2	2

שיטה 3 – data generator (PLSQL developer)

בשיטת זו אנו משתמשים ב-**data generator** המובנה של PLSQL developer

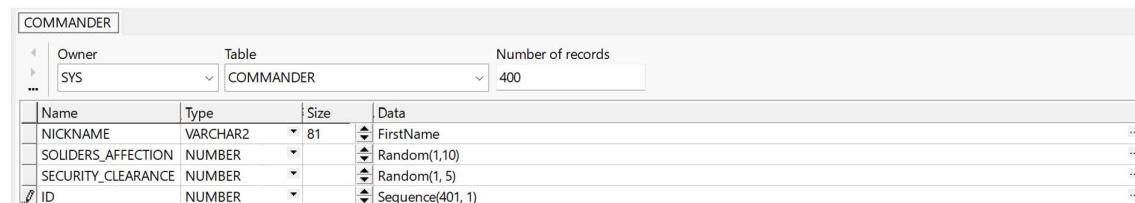
Solider:



The screenshot shows the Oracle SQL Developer Data Generator interface for the SOLIDER table. At the top, it displays the table name 'SOLIDER', owner 'SYS', and number of records '1200'. Below this is a detailed view of the table structure with columns: DATE_OF_BIRTH, ID, FIRST_NAME, RANK, RELIGION, and LAST_NAME. Each column has its data type (DATE, NUMBER, VARCHAR2), size (81), and a dropdown menu showing generation logic: random(01/01/1986, 30/12/2024) for DATE_OF_BIRTH, Sequence(1,1) for ID, FirstName for FIRST_NAME, List('Private', 'Corporal', 'Sergeant', 'Lieutenant', 'Captain', 'Major', 'Colonel', 'Brigadier General', 'Major General', 'General') for RANK, List('Judaism', 'Christianity', 'Islam', 'Hinduism', 'Buddhism', 'Sikhism', 'Taoism', 'Shinto', 'Bahaai', 'Jainism') for RELIGION, and LastName for LAST_NAME.

	DATE_OF_BIRTH	ID	FIRST_NAME	RANK	RELIGION	LAST_NAME	
1	07/09/1993	1	Rachel	General	Shinto	Hewett	...
2	17/10/2007	2	Amanda	Major	Bahaai	Salonga	...
3	11/03/2008	3	Petula	Private	Shinto	Shawn	...
4	11/08/2006	4	Ben	Captain	Bahaai	Lucien	...
5	25/10/1991	5	Ronnie	Major	Islam	McConaughey	...
6	06/05/2018	6	Juan	Private	Islam	Morton	...
7	25/01/1988	7	Julianna	Captain	Hinduism	Bacon	...
8	30/10/2001	8	Scarlett	Captain	Sikhism	Guest	...
9	15/03/2000	9	Ryan	Corporal	Shinto	Hutton	...
10	12/04/1993	10	Juliette	General	Buddhism	Griffin	...
11	16/11/2006	11	Hugh	Brigadier General	Judaism	Kretschmann	...
12	26/04/2015	12	Cheryl	Colonel	Sikhism	Shelton	...
13	20/01/2009	13	Marty	Colonel	Islam	Andrews	...
14	21/01/2024	14	Terri	Colonel	Bahaai	Faria	...

Commander:



The screenshot shows the Oracle SQL Developer Data Generator interface for the COMMANDER table. At the top, it displays the table name 'COMMANDER', owner 'SYS', and number of records '400'. Below this is a detailed view of the table structure with columns: NICKNAME, SOLDERS_AFFECTION, SECURITY_CLEARANCE, and ID. Each column has its data type (VARCHAR2, NUMBER, NUMBER, NUMBER), size (81), and a dropdown menu showing generation logic: FirstName for NICKNAME, Random(1,10) for SOLDERS_AFFECTION, Random(1, 5) for SECURITY_CLEARANCE, and Sequence(401, 1) for ID.

	NICKNAME	SOLDERS_AFFECTION	SECURITY_CLEARANCE	ID
	VARCHAR2	NUMBER	NUMBER	NUMBER
	81			
	FirstName	Random(1,10)	Random(1, 5)	Sequence(401, 1)

ואב בביוף
אBITS' שchor

	NICKNAME	SOLDIERS_AFFECTION	SECURITY_CLEARANCE	ID
1	Barry	8	3	401
2	Annette	9	2	402
3	Tobey	3	4	403
4	Rufus	5	4	404
5	Hugh	2	1	405
6	Ethan	5	3	406
7	Helen	7	1	407
8	Kristin	7	2	408
9	Lennie	10	2	409
10	Emmylou	3	1	410
11	Danny	4	3	411
12	Ray	2	3	412
13	Franz	7	2	413
14	Rich	4	3	414

Combat medic:

COMBAT_MEDIC				
Owner	Table	Number of records		
SYS	COMBAT_MEDIC	400		
	Name	Type	Size	Data
	QUALIFICATION	VARCHAR2	81	List('MD', 'DO', 'RN', 'NP', 'PA', 'DDS', 'DMD', 'PharmD', 'DVM', 'OD')
	IN_TRAINING	NUMBER		Random(0, 1)
	ID	NUMBER		Sequence(1, 1)

	QUALIFICATION	IN_TRAINING	ID
1	DMD	0	1
2	DMD	0	2
3	PA	1	3
4	DDS	1	4
5	RN	1	5
6	PA	1	6
7	PA	1	7
8	NP	1	8
9	DO	1	9
10	DO	1	10
11	PharmD	1	11
12	PA	1	12
13	DDS	0	13
14	DO	0	14

Military rabbi:

MILITARY_RABBI				
Owner	Table	Number of records		
SYS	MILITARY_RABBI	400		
	Name	Type	Size	Data
	CERTIFIED_RABBI	NUMBER		Random(0, 1)
	PAYYTAN	NUMBER		Random(0, 1)
	SECTOR	VARCHAR2	81	List('Religious-Zionism', 'Haredi-Litai', 'Haredi-Hasidic')
	ETHNICITY	VARCHAR2	81	List('Ashkenazi', 'Iraqi', 'Buchari', 'Teman', 'Sphard')
	PREACHER	NUMBER		Random(0, 1)
	ID	NUMBER		Sequence(801, 1)

ואב בביוף
אבייש שchor

	CERTIFIED_RABBI	PAYYTAN	SECTOR	ETHNICITY	PREACHER	ID
1	1	1	Haredi-Hasidic	Buchari	0	801
2	1	0	Haredi-Hasidic	Temani	1	802
3	1	0	Religious-Zionism	Ashkenazi	0	803
4	0	0	Haredi-Hasidic	Temani	0	804
5	1	0	Haredi-Hasidic	Sphard	0	805
6	1	1	Haredi-Hasidic	Buchari	0	806
7	0	1	Haredi-Hasidic	Sphard	1	807
8	0	1	Haredi-Hasidic	Temani	1	808
9	1	1	Haredi-Litai	Buchari	1	809
10	0	1	Haredi-Litai	Iraqi	1	810
11	1	1	Haredi-Litai	Sphard	1	811
12	1	1	Haredi-Litai	Sphard	1	812
13	1	1	Haredi-Litai	Temani	1	813
14	0	1	Haredi-Litai	Buchari	0	814

Team:

TEAM			
Owner	Table	Number of records	
SYS	TEAM	400	
	Name	Type	Size
	TEAM_NUMBER	NUMBER	Sequence(1, 1)
	NAME	VARCHAR2	81
	RELIGIOUS	NUMBER	Random(0, 1)
	COMMANDER_ID	NUMBER	Sequence(401,1)

	TEAM_NUMBER	NAME	RELIGIOUS	COMMANDER_ID
1	1	Zaandam	1	401
2	2	Salem	0	402
3	3	Akita	1	403
4	4	Golden	1	404
5	5	Colorado Springs	0	405
6	6	Herne	1	406
7	7	Bielefeld	0	407
8	8	St Jean de Soudain	1	408
9	9	Bristol	1	409
10	10	Bergara	1	410
11	11	Tucson	1	411
12	12	Antwerpen	1	412
13	13	Rorschach	0	413
14	14	Double Oak	0	414

יאוב בביוי'

אבייש שchor

Operation:

OPERATION

Name	Type	Size	Data
OPERATION_DATE	DATE	81	random(01/01/1986, 30/12/2024)
LOCATION	VARCHAR2	81	Country
OPERATION_ID	NUMBER	...	Random(1, 99999999)
COMMANDER_ID	NUMBER	...	Sequence(401,1)
MEDIC_ID	NUMBER	...	Sequence(1, 1)
TEAM_NUMBER	NUMBER	...	Sequence(1, 1)

	OPERATION_DATE	LOCATION	OPERATION_ID	COMMANDER_ID	MEDIC_ID	TEAM_NUMBER
1	24/04/1988	USA	35188343	401	1	1
2	16/03/2005	USA	731001008	402	2	2
3	17/09/1993	Canada	605736777	403	3	3
4	26/12/2010	USA	534349540	404	4	4
5	20/02/2010	Belgium	86975604	405	5	5
6	01/10/2012	USA	545678862	406	6	6
7	13/07/1994	Italy	483903501	407	7	7
8	18/08/2016	USA	900754043	408	8	8
9	10/05/2003	United Kingdom	683925872	409	9	9
10	29/05/2001	Norway	690369426	410	10	10
11	05/12/2011	USA	961505079	411	11	11
12	26/03/2015	Germany	931749871	412	12	12
13	21/02/2003	Canada	691187826	413	13	13
14	28/11/1991	Taiwan	358042001	414	14	14

Soldier_in:

SOLDIER_IN

Name	Type	Size	Data
RESPONSIBILITY	VARCHAR2	81	List("Infantry Soldier", "Sniper", "Combat Engineer", "Special Forces Operator", "Armored Crewman", "Artillery Gunner", "Intelligence Officer", "Medical Specialist", "Logistics Specialist", "Cyber Warfare Specialist", "Parachute Rigger", "Mine Detector", "Search and Rescue Specialist", "Forward Observer", "Military Police Officer")
TEAM_NUMBER	NUMBER	...	Sequence(1,1)
ID	NUMBER	...	Sequence(2,2)

	RESPONSIBILITY	TEAM_NUMBER	ID
1	"Forward Observer"	1	2
2	"Forward Observer"	2	4
3	"Combat Engineer"	3	6
4	"Sniper"	4	8
5	"Forward Observer"	5	10
6	Infantry Soldier	6	12
7	"Military Police Officer"	7	14
8	"Sniper"	8	16
9	"Combat Engineer"	9	18
10	Infantry Soldier	10	20
11	Infantry Soldier	11	22
12	"Forward Observer"	12	24
13	"Forward Observer"	13	26
14	Infantry Soldier	14	28

גיבוי נתונים

Name	Type	Compiled
METAXSL\$	TABLE	28/09/2021 4:32:46
METAXLPPARAM\$	TABLE	28/09/2021 4:32:47
METAXLPPARAMDESC\$	TABLE	28/09/2021 4:32:47
METHODS\$	TABLE	28/09/2021 4:32:28
MIGRATE\$	TABLE	28/09/2021 4:31:28
MILITARY_RABBI	TABLE	26/05/2024 22:19:03
MLOG_REFCOL\$	TABLE	28/09/2021 4:32:32
MODEL\$	TABLE	28/09/2021 4:32:47
MODELALGS\$	TABLE	28/09/2021 4:36:39
MODELATT\$	TABLE	28/09/2021 4:32:48
MODELGTTRAWS\$	TABLE	28/09/2021 4:32:48
MODELPART\$	TABLE	28/09/2021 4:32:48
MODELPARTCOL\$	TABLE	28/09/2021 4:32:48
MODELSETS\$	TABLE	28/09/2021 4:32:48
MODELTABS\$	TABLE	28/09/2021 4:32:48
MODELXFM\$	TABLE	28/09/2021 4:32:48
MON_MODS\$	TABLE	28/09/2021 4:32:25
MON_MODS_ALL\$	TABLE	28/09/2021 4:32:25
MVREFS_CHANGE_STATS	TABLE	28/09/2021 4:39:57
MVREFS_DML_STATS	TABLE	28/09/2021 4:39:57

User <CURRENT USER>

Oracle Export SQL Inserts PL/SQL Developer Log

Drop tables Disable foreign key constraints
 Create tables Include storage
 Truncate tables Include privileges
 Delete records Commit every 100 records (0 = never)
 Disable triggers
 Zip
Where clause
Output file C:\Users\Yoav\Desktop\minip basnat\backup - 26_05_2024\aw.sql

The backup can be found in the GitHub of the project.

שחזר נתונים מגיבוי

The screenshot shows the Oracle SQL Developer interface with the 'Import' tab selected. The 'Import file' dropdown menu is open, showing the path 'C:\Users\Yoav\Desktop\minip_basnat\backup - 26_05_2024\aw.sql'. The 'Import' button is located at the bottom right of the import dialog.

ובמבחן התוצאה:

```
-- Select all records from the team table
SELECT * FROM team;

-- Select all records from the military_Rabbi table
SELECT * FROM military_Rabbi;

-- Select all records from the combat_medic table
SELECT * FROM combat_medic;

-- Select all records from the operation table
SELECT * FROM operation;

-- Select all records from the solider_in table
SELECT * FROM solider_in;
```

The screenshot shows the results of the executed SQL queries in the Oracle SQL Developer interface. The results are displayed in a grid format with three columns: 'RESPONSIBILITY', 'TEAM_NUMBER', and 'ID'. The data includes various soldier roles and their corresponding team numbers and IDs.

	RESPONSIBILITY	TEAM_NUMBER	ID
►	1 "Intelligence Analyst"	...	223 446
	2 "Communications Specialist"	...	224 448
	3 "Forward Observer"	...	225 450
	4 "Artillery Gunner"	...	226 452
	5 Infantry Soldier	...	227 454
	6 "Communications Specialist"	...	228 456
	7 Infantry Soldier	...	229 458
	8 "Special Forces Operator"	...	230 460
	9 "Sniper"	...	231 462
	10 Infantry Soldier	...	232 464
	11 "Sniper"	...	233 466
	12 "Communications Specialist"	...	234 468
	13 Infantry Soldier	...	235 470

חלה ב

SELECT שאלות

שאלת ראשונה – מחלת לבנון

התגלה מחלת לבנון שהתרצה בין השנים 1999 ו-2005. כתוצאה לכך רצתה יחידת יהלום לדעת את הפרטים של החילימ השותפו במצבים לבנון באותה תקופה, על מנת לשוח אוטם לבדיקות. בנוסף, היחידה רצתה לדעת כמה מצבים חיל צהה השתתף, על מנת לנתח את הסיכון הנטו לו.

```
-- operations in Lebanon between the specified dates (when the disease
happened)
WITH lebanon_operations AS (
    SELECT
        o.team_number,
        o.operation_date
    FROM
        operation o
    WHERE
        o.location = 'Lebanon' -- Where the location is Lebanon
        AND o.operation_date BETWEEN TO_DATE('01/01/1999', 'DD/MM/YYYY')
        AND TO_DATE('01/01/2005', 'DD/MM/YYYY') -- And the operation date is
        between January 1, 1999, and January 1, 2005
),
-- get number of relevant operations every soldier was part of
soldiers_in_operations AS (
    SELECT
        si.id AS soldier_id,
        COUNT(*) AS number_of_operations -- Counting the number of
        operations for each soldier
        -- from natural join of soldier-in and lebanon operations to match the
        team numbers
    FROM
        soldier_in si
    JOIN
        lebanon_operations lo ON si.team_number = lo.team_number
    GROUP BY
        si.id -- Grouping by soldier's ID to count the operations per
        soldier
)
-- retrieve soldier details along with the number of operations they were
involved in
SELECT
    s.id AS soldier_id,
    s.first_name,
    s.last_name,
    s.rank,
    s.date_of_birth,
    sio.number_of_operations
FROM
```

וְאֶבֶן בְּבִזְעִיף**אֲבִישֵׁי שְׁחוֹר**

soldier s -- From the join of solider and sio to get view with details of soldiers in operation

JOIN

soldiers_in_operations sio ON s.id = sio.solider_id;

ובהרצתה לדוגמא:

	SOLIDER_ID	FIRST_NAME	LAST_NAME	RANK	DATE_OF_BIRTH	NUMBER_OF_OPERATIONS
▶	1	198	Chaka	... Sewell	... Major General	... 12/03/2006 ...
	2	262	Lynette	... Gordon	... Sergeant	... 11/11/2023 ...
	3	280	Kristin	... Atkinson	... Corporal	... 17/08/1988 ...
	4	328	Lucy	... Aaron	... Brigadier General	... 04/05/2019 ...
	5	374	Edward	... Dern	... Major General	... 31/03/2024 ...
	6	520	Jean-Luc	... Niven	... Major General	... 04/09/2001 ...
	7	636	Jason	... Fraser	... Lieutenant	... 29/12/2005 ...
	8	734	Danny	... Choice	... Brigadier General	... 17/11/2011 ...
	9	790	Robby	... Stampley	... Major General	... 14/11/2011 ...

שאילתה שנייה – חיילים בעלי פוטנציאל גבוה

יחידת יחל"ם רוצה למצוא את החילים בעלי הפוטנציאל הגבוה ביותר ביוטר משורותיה. על מנת למצוא אותם, היא רוצה לקבל את פרטי החילים שהשתתפו במקרה שעלי פיקד חיל בעל הדרגה הגבוהה ביותר ביחידת General, ואת כמות המבצעים בהם השתתפו, מסודר לפי כמות המבצעים ואז תאריך הלידה של החיל בסדר יורד (על פי ההנחה שחייל צער יותר שהשתתף במקרה זה הוא בעל פוטנציאל גבוה יותר):

```
-- Get details of soldiers born between 2001 and 2007 that participated in
-- an operation with their commander being the highest-ranking commanders, and
-- the operation was in Iran. Also, count the number of operations each
-- soldier participated in.
SELECT
    s.id AS soldier_id,
    s.first_name,
    s.last_name,
    s.date_of_birth,
    s.rank AS soldier_rank,
    COUNT(o.operation_id) AS num_operations, -- Count the number of
operations each soldier participated in
    o.location
FROM
    (SELECT
        *
    FROM
        solider
    WHERE
        date_of_birth
        BETWEEN TO_DATE('01/01/1995', 'DD/MM/YYYY')
        AND
        TO_DATE('01/01/2006', 'DD/MM/YYYY')
    ) s -- Soldiers born between 2001 and 2007
JOIN
    solider_in si ON s.id = si.id
JOIN
    operation o ON si.team_number = o.team_number
JOIN
    commander cm ON o.commander_id = cm.id
JOIN
    solider c ON cm.id = c.id
WHERE
    c.rank = 'General'
GROUP BY
    s.id, s.first_name, s.last_name, s.date_of_birth, s.rank, o.location -- -
- Grouping by soldier details and operation location
ORDER BY
    num_operations DESC, s.date_of_birth DESC; -- --
Ordering results by soldier's date of birth and amount of operations
```

וברצה לדוגמא:

ואב בביוף

אבייש שוחר

	SOLDIER_ID	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	SOLDIER_RANK	NUM_OPERATIONS	LOCATION
►	1	232 Roddy	„ Tate	„ 18/01/1995	„ Lieutenant	„	1 Saudi Arabia „
	2	250 Adina	„ Duke	„ 14/11/1995	„ Major	„	1 Iran „
	3	272 Julianne	„ Yulin	„ 15/07/1996	„ Private	„	1 Iran „
	4	110 Tia	„ Waits	„ 18/04/1999	„ Private	„	1 Saudi Arabia „
	5	220 Joanna	„ McNeice	„ 25/06/1999	„ Major	„	1 Saudi Arabia „
	6	90 Meredith	„ Kadison	„ 25/07/2000	„ Lieutenant	„	1 Iraq „
	7	20 Jeremy	„ Tilly	„ 13/02/2001	„ Lieutenant	„	1 Iran „
	8	18 Carrie-Anne	„ Chambers	„ 01/07/2001	„ Brigadier General	„	1 Iran „
	9	456 Talvin	„ Heron	„ 25/09/2001	„ Brigadier General	„	1 Lebanon „
	10	432 Simon	„ Tambor	„ 19/12/2001	„ Colonel	„	1 Lebanon „

שאילתת שלישית – אהדת חילים לפי דת המפקד

יחידת יחל"ם רוצה לדעת האם יש קשר בין דת המפקד לבין אהדת החילים כלפיו. על מנת לבדוק זאת, יצרנו את השאלה הזו, אשר מאגדת את כל המפקדים לפי דתם ובמוצע על חיליהם כלפייהם. על מנת שההנחתה תהיה נכוןה, הוגבל המدى רק למפקדים בין גילאים 25 ל-65 (על מנת שהמפקדים לא יהיו מבוגרים מדי או צעירים מדי), ונבדקו רק צוותים אשר יש בהם לפחות שני חילים עם אמונה דתית שונאות, מתוך הנחה שצוותים יותר פלורליסטיים יהיו בעלי הטיה פחות חמורה כלפי דת מסוימת.

בנוסף, על מנת שהחידה תוכל להסיק מהנתונים מסקנות, הוספנו את המובוקות הסטטיסטיות של הנתונים, כאשר נתון הוא מובוק סטטיסטית אם הוא נעשה בממוצע על 30 חילים לפחות (לפי עיקרון המספרים הגדולים), ובנוסף ניתן גיל ממוצע של המפקד באותה דת על מנת להתמודד עם הטיות גילאיות.

```
-- get average affection towards commanders
-- correlated with age and religion of commander,
-- commander between ages 25 and 65
-- and pluralism between their's soliders
SELECT
    cs.religion AS commander_religion,
    ROUND(AVG(EXTRACT(YEAR FROM cs.date_of_birth)) - 1975) AS
avg_commander_age,
    ROUND(AVG(c.soliders_affection),3) AS avg_soldier_affection,
    CASE WHEN COUNT(DISTINCT s.id) >= 30 THEN 'Yes' ELSE 'No' END AS
statisticaly_significant
FROM
    commander c
JOIN
    solider cs ON c.id = cs.id
JOIN
    team t ON c.id = t.commander_id
JOIN
    solider_in si ON t.team_number = si.team_number
JOIN
    solider s ON si.id = s.id
WHERE
    cs.date_of_birth BETWEEN TO_DATE('01/01/1990', 'DD/MM/YYYY') AND
TO_DATE('31/12/2020', 'DD/MM/YYYY')
    AND EXISTS ( -- commands on soliders with at least two different
religions
        SELECT *
        FROM
            solider_in si2
        JOIN
            solider s2 ON si2.id = s2.id
        WHERE
            si2.team_number = t.team_number
            AND EXISTS (
                SELECT *
                FROM
```

יואב בביוף
אבייש שchor

```

        solider_in_si3
    JOIN
        solider s3 ON si3.id = s3.id
    WHERE
        s3.religion <> s2.religion
    )
)
GROUP BY
    cs.religion
ORDER BY
    avg_soldier_affection DESC;

```

ובחרצת לדוגמא:

	COMMANDER_RELIGION	AVG_COMMANDER_AGE	AVG_SOLDIER_AFFECTION	STATISTICALY_SIGNIFICANT
► 1	Jainism	31	6.167	No
2	Judaism	30	5.857	Yes
3	Sikhism	29	5.852	No
4	Buddhism	28	5.543	Yes
5	Christianity	30	5.515	Yes
6	Hinduism	33	5.353	Yes
7	Islam	29	5.286	No
8	Taoism	29	4.938	Yes
9	Shinto	28	4.938	Yes
10	Bahaai	29	4.794	Yes

שאילתה רביעית – מתחזים

יחידת יחל"ם גילתה שבחלק מהמבצעים שהיו בין 1995 ל-2005, היו בצוותים שביצעו את המבצעים מסתננים – חיללים שדתם אינה יהודות אך הצהירו שהם רבנים צבאים, ובכך נכנסו אל צוות המבצע ביותר קלות. זהה הפרת בטחון מידע חמורה, ולכן היהידה רוצה רשותה ובה כל המבצעים בהם היה מסתנן צזה, כמוות המסתננים, ומפקד המבצע, כאשר הסיווג הבטחוני שלו הוא לפחות "סודי" (3), ובצווות לפחות שני חיילים (כך שייתר מפרק המפקד הושפע מזה). הנתונים מסודרים לפי כמוות המתחזים, ואז לפיה תאריך המבצע.

```
-- operations between 1995 and 2005, where there is at least one non-Jewish
military rabbi in the team.
-- also shows how many of them were in the team, and the commander of the
operation
SELECT
    o.operation_id,
    o.operation_date,
    o.location,
    c.id AS commander_id,
    cs.first_name AS commander_first_name,
    cs.last_name AS commander_last_name,
    COUNT(DISTINCT mr.id) AS non_jewish_rabbi_count
FROM
    operation o
JOIN
    team t ON o.team_number = t.team_number
JOIN
    solider_in si ON t.team_number = si.team_number
JOIN
    military_Rabbi mr ON si.id = mr.id
JOIN
    solider s ON mr.id = s.id -- Join to get the religion of the military
rabbi
JOIN
    commander c ON o.commander_id = c.id
JOIN
    solider cs ON c.id = cs.id -- Join to get commander details
WHERE
    o.operation_date BETWEEN TO_DATE('01/01/1995', 'DD/MM/YYYY') AND
TO_DATE('31/12/2005', 'DD/MM/YYYY')
        AND s.religion <> 'Judaism'
        AND c.security_clearance <= 3
        -- teams with at least two members
        AND t.team_number IN (
            SELECT
                si.team_number
            FROM
                solider_in si
            GROUP BY
                si.team_number
            HAVING
                COUNT(DISTINCT si.id) >= 2
        )
GROUP BY
    o.operation_id, o.operation_date, o.location, t.team_number, c.id,
    cs.first_name, cs.last_name
ORDER BY
    non_jewish_rabbi_count DESC, o.operation_date DESC;
```

בס"ד
מיני-פרוייקט בבסיסי נתונים

ואב בביוף
אבייש שchor

ובהרצת לדוגמא:

	OPERATION_ID	OPERATION_DATE	LOCATION	COMMANDER_ID	COMMANDER_FIRST_NAME	COMMANDER_LAST_NAME	NON_JEWISH_RABBI_COUNT
► 1	48635911	15/01/2005	„ Israel	... 456	Talvin	„ Heron	...
2	514784399	07/11/2002	„ Iran	... 468	Viggo	„ McReady	1
3	107993598	24/11/2001	„ Saudi Arabia	... 458	Cheryl	„ Schock	1
4	2519626	21/08/2001	„ Iran	... 409	Dabney	„ McGill	1
5	49078312	31/01/2000	„ Saudi Arabia	... 408	Tea	„ Williams	1
6	222620172	24/07/1997	„ Iraq	... 451	Burt	„ Curry	1
7	476266295	30/03/1996	„ Saudi Arabia	... 478	Scott	„ Ferry	1
8	188010190	30/01/1996	„ Israel	... 440	Kelli	„ Li	1
9	375539055	10/07/1995	„ Lebanon	... 437	Tcheky	„ Phoenix	1
10	594655844	11/04/1995	„ Israel	... 423	Earl	„ Blaine	1

שאילתות DELETE

שאילתת ראשונה – מדריכים

נודע לייחידת יחל"ם כי במבצע האחרון בו השתתפו הדלי'ף אחד מחברי הצוות מידע לאויב. ידוע עליו כי שמו הפרטני הינו Maury, והוא בדרגת רב פקד (Major). נרצה לשחרר משירות כל אדם שמתאים לתיאור זהה, על מנת לשמר על בט"מ.

```
-- delete leakers: named David, and participated in the last operation)
DELETE FROM solider
WHERE id IN (
    SELECT s.id
    FROM solider s
    JOIN solider_in si ON s.id = si.id
    JOIN operation o ON si.team_number = o.team_number
    JOIN commander c ON o.commander_id = c.id
    WHERE s.first_name = 'Maury'
        AND s.rank = 'Major'
        AND o.operation_date = (
            SELECT MAX(o2.operation_date)
            FROM operation o2
        )
);
)
```

שאילתת שנייה – תקנית במבצע

יחידת יחל"ם גילתה כי הייתה תקנית מבצעית במבצע מסוים ועל כן היא רוצה להעיף כל מי שהוא במבצע מצה"ל.

```
DELETE FROM solider
WHERE id IN (
    SELECT s.id
    FROM solider s
    JOIN solider_in si ON s.id = si.id
    JOIN operation o ON si.team_number = o.team_number
    WHERE o.operation_id = 721577260 AND s.id != o.commander_id
);
)
```

שאילות UPDATE

שאילה ראשונה – עדכון פרטיים אישיים

הייתה טעות ברישום פרטי החילימ ובטעות חילימ בשם Goldie נרשמו כ- Goldie ולהיפך. זה קרה רק עבור חילימ שנולדו בחודש נובמבר. רצאה להחלף את פרטייהם.

```
-- Rename all soliders named Goldie to Golada
UPDATE solider
SET first_name = CASE
    WHEN first_name = 'Goldie' AND EXTRACT(MONTH FROM date_of_birth) = 11 THEN 'Golda'
    WHEN first_name = 'Golda' AND EXTRACT(MONTH FROM date_of_birth) = 11 THEN 'Goldie'
END
WHERE (first_name = 'Goldie' AND EXTRACT(MONTH FROM date_of_birth) = 11)
    OR (first_name = 'Golda' AND EXTRACT(MONTH FROM date_of_birth) = 11);
```

שאילה שנייה – עדכון דרגות

חידת יהל"ם רצאה לתגמל את החילימ שהשתתפו בפעולות חילוץ החטופים האחרונות, ולהעלוות אותן בדרגה.

```
-- give better rank for soliders participated in the last operation
UPDATE solider
SET rank = CASE
    WHEN rank = 'Private' THEN 'Corporal'
    WHEN rank = 'Corporal' THEN 'Sergeant'
    WHEN rank = 'Sergeant' THEN 'Lieutenant'
    ELSE rank
END
WHERE id IN (
    SELECT s.id
    FROM solider s
    JOIN soldier_in si ON s.id = si.id
    JOIN operation o ON si.team_number = o.team_number
    WHERE o.operation_date = (
        SELECT MAX(o2.operation_date)
        FROM operation o2
    )
);
```

שאלות עם פרמטרים

שאלת ראשונה – כל החיילים תחת מפקד

מאתגרת את כל החיילים תחת מפקד מסוים ובדרגות מסוימות (פרמטרים).

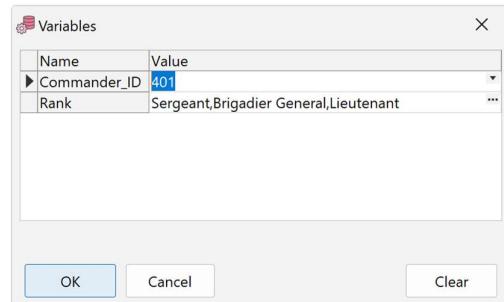
-- Query 1: List all soldiers who are part of a team commanded by a specific commander and have a rank equal to given rank.

```
SELECT s.id, s.first_name, s.last_name, s.rank, t.name AS team_name
FROM solider s
JOIN solider_in si ON s.id = si.id
JOIN team t ON si.team_number = t.team_number
JOIN commander c ON t.commander_id = c.id
WHERE c.id = (&<name = "Commander_ID"
              type = "integer"
              List = "SELECT id FROM commander"
              required = "true">)
          AND
              s.rank in (&<name = "Rank"
              type = "String"
              List = "SELECT DISTINCT rank FROM solider"
              required = "true"
              multiselect = "true">)
ORDER BY s.rank DESC;
```

הרצה:

-- Query 1: List all soldiers who are part of a team commanded by a specific commander and have a rank equal to :

```
SELECT s.id, s.first_name, s.last_name, s.rank, t.name AS team_name
FROM solider s
JOIN solider_in si ON s.id = si.id
JOIN team t ON si.team_number = t.team_number
JOIN commander c ON t.commander_id = c.id
WHERE c.id = (&<name = "Commander_ID"
              type = "integer"
              List = "SELECT id FROM commander"
              required = "true">)
          AND
              s.rank in (&<name = "Rank"
              type = "String"
              List = "SELECT DISTINCT rank FROM solider"
              required = "true"
              multiselect = "true">)
ORDER BY s.rank DESC;
```



שאילהה 2 – מבצעים מיוחדים עם מהמהה רפואיה

יחידת יהל"ם רוצה למצוא את כל הרפואיים עם רופא צבאי בהתחמות מסוימת (מתוך רשימה), בין שני זמנים מסוימים (פרמטרים גם הם).

```

SELECT o.operation_id, o.operation_date, o.location, cm.qualification,
t.name AS team_name
FROM operation o
JOIN combat_medic cm ON o.medic_id = cm.id
JOIN team t ON o.team_number = t.team_number
WHERE
o.operation_date
between
(&<name = "First Date"
type = "date"
default = "sysdate"
list = "select operation_date from operation"
required = "true">)
AND
(&<name = "Last Date"
type = "date"
default = "sysdate"
list = "select operation_date from operation"
required = "true">)
AND cm.qualification In (&<name = "Qualification"
type = "String"
list = "SELECT DISTINCT qualification
FROM combat_medic"
required = "true"
multiselect = "true">)
ORDER BY o.operation_date ASC

```

The screenshot shows a database interface with a results grid and a configuration dialog.

Results Grid:

OPERATION_ID	OPERATION_DATE	LOCATION	QUALIFICATION
1	967262895	29/10/2007	„ Israel „ DVM „ Gen
2	895013054	18/11/2008	„ Iran „ OD „ Balti
3	7511403	03/12/2008	„ Saudi Arabia „ OD „ Kura
4	206163450	14/12/2009	„ Saudi Arabia „ OD „ Salv
5	866400181	24/12/2011	„ Saudi Arabia „ DVM „ Bue
6	397884367	22/02/2012	„ Iran „ OD „ Macassar
7	913355692	28/08/2013	„ Israel „ DVM „ Kochi
8	943160488	14/05/2014	„ Iraq „ DVM „ West Launceston
9	213932068	15/06/2014	„ Israel „ DVM „ Baltimore
10	47041999	13/02/2015	„ Iraq „ DVM „ Maryville
11	123888550	19/08/2016	„ Lebanon „ DVM „ Kagoshima
12	661397990	25/11/2016	„ Iran „ DVM „ Suwon
13	440828632	14/12/2016	„ Iran „ DVM „ Dietikon

Variables Dialog:

Name	Value
First Date	06/06/2007
Last Date	18/07/2023
Qualification	OD,DVM

שאילה 3 – אहדת חילימ גבואה

יחידת יחל"ם רוצה לדעת את הפרטים של כל המפקדים עם אהדת חילימ מעל כמה מסוימת על מנת לסמם כאחובי החילימ.

```
SELECT c.id, c.nickname, s.first_name, s.last_name,
c.soliders_affection, s.rank
FROM commander c
JOIN team t ON c.id = t.commander_id
JOIN solider_in si ON t.team_number = si.team_number
JOIN solider s ON si.id = s.id
WHERE c.soliders_affection > (&<name = "Affection Rating"
                                list = "select distinct soliders_affection
from commander"
                                type = "integer"
                                required = "true">)
ORDER BY c.soliders_affection DESC
```

ובהרצה:

```
SELECT c.id, c.nickname, s.first_name, s.last_name, c.soliders_affection, s.rank
FROM commander c
JOIN team t ON c.id = t.commander_id
JOIN solider_in si ON t.team_number = si.team_number
JOIN solider s ON si.id = s.id
WHERE c.soliders_affection > (&<name = "Affection Rating"
                                list = "select distinct soliders_affection from commander"
                                type = "integer"
                                required = "true">)
ORDER BY c.soliders_affection DESC
```

The screenshot shows a database interface with a grid of data and a 'Variables' dialog box. The grid has columns: ID, NICKNAME, FIRST_NAME, LAST_NAME, SOLDIERS_AFFECTION, and RANK. The data includes rows for Judy, Night, Zooey, Carrie-Anne, Ramsey, Debi, Wallace, Roy, Jackson, and Rebeka, each with their respective nicknames, first names, last names, affection levels (ranging from 10 to 100), and ranks (Major, Corporal, Colonel, Major General, Sergeant, Lieutenant, Private, Sergeant, Sergeant, Brigadier General). To the right of the grid is a 'Variables' dialog box with a table containing a single row: 'Affection Rating' with value '3'. Buttons for 'OK' and 'Cancel' are at the bottom of the dialog.

ID	NICKNAME	FIRST_NAME	LAST_NAME	SOLDIERS_AFFECTION	RANK
1	503 Judy	''' Dustin	''' Solido	...	10 Major
2	481 Night	''' Kid	''' Bush	...	10 Corporal
3	520 Zooey	''' Teena	''' Saxon	...	10 Colonel
4	538 Carrie-Anne	''' Sal	''' Spacek	...	10 Major General
5	545 Ramsey	''' Laurie	''' Swayne	...	10 Corporal
6	561 Debi	''' Larry	''' Giraldo	...	10 Sergeant
7	564 Wallace	''' Rachael	''' Donovan	...	10 Lieutenant
8	581 Roy	''' Mickey	''' Fierstein	...	10 Private
9	591 Melba	''' Jason	''' Gilliam	...	10 Sergeant
10	602 Jackson	''' Johnnie	''' Rucker	...	10 Sergeant
11	603 Rebeka	''' Elle	''' McNeice	...	10 Brigadier General

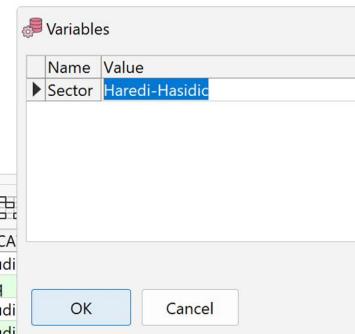
שאילתה רבעית –Robins לוחמים מגזר מסוים

יחידת יחל"ם רוצה להיות מסוגלת לדעת בכל מגזר עלRobins שהם גםRobins מסוימים, גםDRShנים וגם השתתפו במבצע כלשהו (גם לוחמים).

```
SELECT mr.id, s.first_name, s.last_name, mr.sector, o.operation_id,
o.operation_date, o.location
FROM military_Rabbi mr
JOIN solider s ON mr.id = s.id
Join solider_in si on si.id = s.id
JOIN operation o ON si.team_number = o.team_number
WHERE mr.certified_Rabbi = 1
AND mr.preacher = 1
AND mr.sector = (&<name = "Sector"
type = "String"
List = "SELECT DISTINCT sector FROM
military_Rabbi"
required = "true">)
ORDER BY s.last_name, o.operation_date
```

ובהרצה:

```
SELECT mr.id, s.first_name, s.last_name, mr.sector, o.operation_id, o.operation_date, o.location
FROM military_Rabbi mr
JOIN solider s ON mr.id = s.id
Join solider_in si on si.id = s.id
JOIN operation o ON si.team_number = o.team_number
WHERE mr.certified_Rabbi = 1
AND mr.preacher = 1
AND mr.sector = (&<name = "Sector"
type = "String"
List = "SELECT DISTINCT sector FROM military_Rabbi"
required = "true">)
ORDER BY s.last_name, o.operation_date
```



ID	FIRST_NAME	LAST_NAME	SECTOR	OPERATION_ID	OPERATION_DATE	LOCATION
1	835 Freddie	Askew	Haredi-Hasidic	49078312	31/01/2000	Saudi
2	835 Freddie	Askew	Haredi-Hasidic	544872435	05/07/2002	Iraq
3	874 Giancarlo	Bailey	Haredi-Hasidic	789682026	15/11/2020	Saudi
4	877 Ricardo	Cleese	Haredi-Hasidic	533234647	23/11/1989	Saudi
5	1100 Javon	Hampton	Haredi-Hasidic	24514083	15/05/1991	Israel
6	871 Caroline	Jay	Haredi-Hasidic	870685839	24/08/2017	Iraq
7	1140 Miles	Karyo	Haredi-Hasidic	752719132	09/06/2019	Saudi Arabia
8	920 Campbell	Leary	Haredi-Hasidic	182821698	13/06/1992	Israel
9	847 Jaime	McCormack	Haredi-Hasidic	502043500	08/04/2021	Iran
10	841 Hookah	Quinones	Haredi-Hasidic	976607567	21/07/2008	Lebanon
11	1085 Colin	de Lanie	Haredi-Hasidic	107993598	24/11/2001	Saudi Arabia

אילוצים

```
-- check boolean
ALTER TABLE combat_medic
DROP CONSTRAINT chk_combat_medic_bool;

ALTER TABLE combat_medic
ADD CONSTRAINT chk_combat_medic_bool CHECK (in_training BETWEEN 0 AND 1);

-- make boolean set false
ALTER TABLE team
MODIFY religious DEFAULT 1;

-- check boolean
ALTER TABLE military_rabbi
DROP CONSTRAINT chk_military_rabbi_bool;

ALTER TABLE military_rabbi
ADD CONSTRAINT chk_military_rabbi_bool CHECK (payytan BETWEEN 0 AND 1);
commit;
```

חלק ג

פונקציות

פונקציה ראשונה – גיל ממוצע

הfonקציה מאפשרת ליהלום לדעת את הגיל הממוצע של חיילים במצב מסוים:

```
CREATE OR REPLACE FUNCTION get_team_avg_age(p_team_number IN NUMBER)
RETURN NUMBER
IS
    v_avg_age NUMBER;
    v_count NUMBER := 0;
    v_total_age NUMBER := 0;

    -- Cursor to get soldier birthdates in the team
    CURSOR c_soldier_ages IS
        SELECT s.date_of_birth
        FROM solider s
        JOIN solider_in si ON s.id = si.id
        WHERE si.team_number = p_team_number;

    -- Record to store soldier birthdate
    r_soldier_birth DATE;

    -- Exception for when no soldiers are found in the team
    no_soldiers_exception EXCEPTION;
BEGIN
    -- Loop through soldier birthdates
    OPEN c_soldier_ages;
    LOOP
        FETCH c_soldier_ages INTO r_soldier_birth;
        EXIT WHEN c_soldier_ages%NOTFOUND;

        v_total_age := v_total_age + TRUNC(MONTHS_BETWEEN(SYSDATE,
r_soldier_birth) / 12);
        v_count := v_count + 1;
    END LOOP;
    CLOSE c_soldier_ages;

    -- Check if any soldiers were found
    IF v_count = 0 THEN
        RAISE no_soldiers_exception;
    END IF;

    -- Calculate average age
    v_avg_age := v_total_age / v_count;

    RETURN v_avg_age;
EXCEPTION
    WHEN no_soldiers_exception THEN
        RAISE_APPLICATION_ERROR(-20001, 'No soldiers found in the specified
team');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'An error occurred: ' || SQLERRM);
END;
/
```

פונקציה 2 – פרטי חיילים בדרגה מסוימת

פונקציה זו מאפשרת ליחידת יחל"ם להציג את כל הפרטים שהם רוצים לגבי חיילים בדרגה מסוימת.

```

CREATE OR REPLACE FUNCTION get_rank_statistics(p_rank IN VARCHAR2)
RETURN SYS_REFCURSOR
IS
    v_result SYS_REFCURSOR;
    v_commander_count NUMBER := 0;
    v_non_commander_count NUMBER := 0;
    v_total_soldiers NUMBER := 0;
    v_total_affection NUMBER := 0;
    v_affection NUMBER := 0;
    v_avg_soldiers_per_commander NUMBER;
    v_avg_soldier_affection NUMBER;
    v_team_numbers VARCHAR2(4000) := '';
    -- Record to store soldier information
    TYPE soldier_record IS RECORD (
        id NUMBER,
        is_commander NUMBER
    );
    v_soldier soldier_record;
    -- Exception for when no soldiers with the given rank exist
    no_soldiers_exception EXCEPTION;
    -- Cursor to get soldiers with the given rank
    CURSOR c_soldiers IS
        SELECT s.id,
               CASE WHEN c.id IS NOT NULL THEN 1 ELSE 0 END AS is_commander
        FROM solider s
        LEFT JOIN commander c ON s.id = c.id
        WHERE s.rank = p_rank;
BEGIN
    -- Open the cursor and fetch soldiers
    OPEN c_soldiers;
    LOOP
        FETCH c_soldiers INTO v_soldier;
        EXIT WHEN c_soldiers%NOTFOUND;
        v_total_soldiers := v_total_soldiers + 1;
        -- Branching to count commanders and non-commanders
        IF v_soldier.is_commander = 1 THEN
            v_commander_count := v_commander_count + 1;
            -- Get soldier affection for commanders
            SELECT soliders affection
            INTO v_affection
            FROM commander
            WHERE id = v_soldier.id;
            v_total_affection := v_total_affection + v_affection;
        ELSE
            v_non_commander_count := v_non_commander_count + 1;
        END IF;
        -- Get team numbers for this soldier
    END LOOP;
    CLOSE c_soldiers;
    RETURN v_result;
END;

```

ואב בביוף

אבייש שchor

```

FOR team_rec IN (SELECT DISTINCT team_number
                  FROM solider_in
                  WHERE id = v_soldier.id) LOOP
    v_team_numbers := v_team_numbers || team_rec.team_number ||
    ',';
END LOOP;
CLOSE c_soldiers

-- Check if any soldiers were found
IF v_total_soldiers = 0 THEN
    RAISE no_soldiers_exception;
END IF;

-- Calculate averages
v_avg_soldiers_per_commander := CASE WHEN v_commander_count > 0
                                         THEN v_total_soldiers /
                                         v_commander_count
                                         ELSE 0 END;
v_avg_soldier_affection := CASE WHEN v_commander_count > 0
                                         THEN v_total_affection /
                                         v_commander_count
                                         ELSE 0 END;

-- Remove trailing comma from team numbers
v_team_numbers := RTRIM(v_team_numbers, ',');

-- Prepare the result cursor
OPEN v_result FOR
    SELECT v_avg_soldiers_per_commander AS avg_soldiers_per_commander,
           v_avg_soldier_affection AS avg_soldier_affection,
           v_commander_count AS commander_count,
           v_non_commander_count AS non_commander_count,
           v_total_soldiers AS total_soldiers,
           v_team_numbers AS team_numbers
      FROM dual;

-- Example DML command (update): Increase security clearance for
-- commanders of this rank
UPDATE commander c
SET c.security_clearance = LEAST(c.security_clearance + 1, 5)
WHERE c.id IN (SELECT s.id FROM solider s WHERE s.rank = p_rank);

RETURN v_result;

EXCEPTION
    WHEN no_soldiers_exception THEN
        RAISE_APPLICATION_ERROR(-20001, 'No soldiers with this rank
exist');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20002, 'An error occurred: ' || SQLERRM);
END;
/

```

תוכנה ראשית – פונקציה

התוכנה הראשית הבאה נותנת לנו את כל הפרטים על חיילים בדרגת coporal, ועל כל הוצאותם שיש בהם חיילים בדרגה זו:

```

DECLARE
    p_rank VARCHAR2(81);
    p_team_number NUMBER;
    v_rank_stats SYS_REFCURSOR;
    v_avg_soldiers_per_commander NUMBER;
    v_avg_soldier_affection NUMBER;
    v_commander_count NUMBER;
    v_non_commander_count NUMBER;
    v_total_soldiers NUMBER;
    v_team_numbers VARCHAR2(4000);

    v_team_avg_age NUMBER;
    v_team_number VARCHAR2(10);
    v_start_pos NUMBER;
    v_end_pos NUMBER;
BEGIN
    p_rank := 'Corporal';
    p_team_number := 32;

    -- Get rank statistics
    v_rank_stats := get_rank_statistics(p_rank);
    FETCH v_rank_stats INTO v_avg_soldiers_per_commander,
    v_avg_soldier_affection,
                           v_commander_count, v_non_commander_count,
                           v_total_soldiers, v_team_numbers;
    CLOSE v_rank_stats;

    -- Display results for rank statistics
    DBMS_OUTPUT.PUT_LINE('Analysis for rank: ' || p_rank);
    DBMS_OUTPUT.PUT_LINE('Average soldiers per commander: ' || 
v_avg_soldiers_per_commander);
    DBMS_OUTPUT.PUT_LINE('Average soldier affection: ' || 
v_avg_soldier_affection);
    DBMS_OUTPUT.PUT_LINE('Number of commanders: ' || v_commander_count);
    DBMS_OUTPUT.PUT_LINE('Number of non-commanders: ' || 
v_non_commander_count);
    DBMS_OUTPUT.PUT_LINE('Total soldiers: ' || v_total_soldiers);
    DBMS_OUTPUT.PUT_LINE('Team numbers: ' || v_team_numbers);

    -- Process each team number
    DBMS_OUTPUT.PUT_LINE('Details for each team:');
    v_start_pos := 1;
    LOOP
        v_end_pos := INSTR(v_team_numbers, ',', v_start_pos);
        IF v_end_pos = 0 THEN
            v_team_number := SUBSTR(v_team_numbers, v_start_pos);
        ELSE
            v_team_number := SUBSTR(v_team_numbers, v_start_pos, v_end_pos
- v_start_pos);
        END IF;

        -- Get and display team average age
        BEGIN
            v_team_avg_age := get_team_avg_age(TO_NUMBER(v_team_number));
            DBMS_OUTPUT.PUT_LINE('Team ' || v_team_number || ':' );
        
```

ואב בביוף

אבייש שchor

```

DBMS_OUTPUT.PUT_LINE(' Average age of soldiers: ' ||
ROUND(v_team_avg_age, 2) || ' years');

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Team ' || v_team_number || ': Error
getting average age - ' || SQLERRM);
    END;

    EXIT WHEN v_end_pos = 0;
    v_start_pos := v_end_pos + 1;
END LOOP;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/

```

דוגמת הרצה:

```

Analysis for rank: Corporal
Average soldiers per commander: 2.76744186046511627906976744186046511628
Average soldier affection: 5.30232558139534883720930232558139534884
Number of commanders: 43
Number of non-commanders: 76
Total soldiers: 119
Team numbers: 111,116,126,128,132,145,153,168,194,196,205,208,214,224,228,239,:.
Details for each team:
Team 111:
    Average age of soldiers: 23 years
Team 116:
    Average age of soldiers: 2 years
Team 126:
    Average age of soldiers: 35 years
Team 128:
    Average age of soldiers: 3 years
Team 132:
    Average age of soldiers: 6 years
Team 132:
    Average age of soldiers: 6 years
Team 145:
    Average age of soldiers: 14 years
Team 153:
    Average age of soldiers: 9 years
Team 168:
    Average age of soldiers: 0 years
Team 194:
    Average age of soldiers: 31 years
Team 196:
    Average age of soldiers: 2 years

```

פרוצדורות

פרוצדורה ראשונה – חיילים ראויים לציון

פרוצדורה האפשרת ליהלום את היכולת לקדם חיילים שהשתתפו בזמן האחרון במבצע:

```
CREATE OR REPLACE PROCEDURE PromoteSoldiersInRecentOperations IS
  TYPE solider_rec IS RECORD (
    id solider.id%TYPE,
    rank solider.rank%TYPE,
    first_name solider.first_name%TYPE,
    last_name solider.last_name%TYPE
  );
  CURSOR solider_cursor IS
    SELECT DISTINCT s.id, s.rank, s.first_name, s.last_name
    FROM solider s
    JOIN solider_in si ON s.id = si.id
    JOIN operation o ON si.team_number = o.team_number
    WHERE o.operation_date > SYSDATE - 30;
  solider_record solider_rec;
  v_updated_count NUMBER := 0;
BEGIN
  FOR solider_record IN solider_cursor LOOP
    BEGIN
      UPDATE solider
      SET rank = CASE
        WHEN rank = 'Private' THEN 'Corporal'
        WHEN rank = 'Corporal' THEN 'Sergeant'
        WHEN rank = 'Sergeant' THEN 'Lieutenant'
        ELSE rank
      END
      WHERE id = solider_record.id
      AND rank IN ('Private', 'Corporal', 'Sergeant');

      IF SQL%ROWCOUNT > 0 THEN
        v_updated_count := v_updated_count + 1;
        DBMS_OUTPUT.PUT_LINE('Promoted soldier: ' ||
          solider_record.first_name || ' ' || solider_record.last_name ||
          ' from ' || solider_record.rank || ' to ' ||
          CASE
            WHEN solider_record.rank = 'Private' THEN
              'Corporal'
            WHEN solider_record.rank = 'Corporal' THEN
              'Sergeant'
            WHEN solider_record.rank = 'Sergeant' THEN
              'Lieutenant'
            ELSE solider_record.rank
          END);
      END IF;

      EXCEPTION
      WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error updating soldier with ID: ' ||
          solider_record.id || ' - ' || SQLERRM);
    END;
  END LOOP;
END;
```

ואב בביוף
אבייש שchor

```
DBMS_OUTPUT.PUT_LINE('Total soldiers promoted: ' || v_updated_count);

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

פרוצדורת שנייה – קידום חיילים בצוות מסוים

פרוצדורת המקדמת חיילים הנמצאים בצוות מסוים.

```

CREATE OR REPLACE PROCEDURE update_soldier_ranks_and_responsibilities(
    p_team_number IN team.team_number%TYPE
)
IS
    TYPE soldier_record IS RECORD (
        id solider.id%TYPE,
        first_name solider.first_name%TYPE,
        last_name solider.last_name%TYPE,
        current_rank solider.rank%TYPE,
        date_of_birth solider.date_of_birth%TYPE,
        responsibility solider_in.responsibility%TYPE
    );
    TYPE soldier_list IS TABLE OF soldier_record;
    v_soldiers soldier_list;

    v_team_name team.name%TYPE;
    v_commander_id team.commander_id%TYPE;
    v_commander_nickname commander.nickname%TYPE;

    CURSOR c_team_soldiers IS
        SELECT s.id, s.first_name, s.last_name, s.rank,
        s.date_of_birth, si.responsibility
        FROM solider s
        JOIN solider_in si ON s.id = si.id
        WHERE si.team_number = p_team_number;

    v_age NUMBER;
    v_new_rank solider.rank%TYPE;
    v_new_responsibility solider_in.responsibility%TYPE;

    e_invalid_team EXCEPTION;
    PRAGMA EXCEPTION_INIT(e_invalid_team, -20001);

BEGIN
    -- Get team information
    SELECT name, commander_id INTO v_team_name, v_commander_id
    FROM team
    WHERE team_number = p_team_number;

    IF v_team_name IS NULL THEN
        RAISE e_invalid_team;
    END IF;

    -- Get commander nickname
    SELECT nickname INTO v_commander_nickname
    FROM commander
    WHERE id = v_commander_id;

    -- Fetch soldiers into collection
    OPEN c_team_soldiers;
    FETCH c_team_soldiers BULK COLLECT INTO v_soldiers;
    CLOSE c_team_soldiers;

    -- Process each soldier
    FOR i IN 1..v_soldiers.COUNT LOOP

```

ואב בביוף

אבייש שchor

```

-- Calculate age
v_age := TRUNC(MONTHS_BETWEEN(SYSDATE,
v_soldiers(i).date_of_birth) / 12);

-- Determine new rank based on age
IF v_age < 20 THEN
    v_new_rank := 'Private';
ELSIF v_age < 25 THEN
    v_new_rank := 'Corporal';
ELSIF v_age < 30 THEN
    v_new_rank := 'Sergeant';
ELSE
    v_new_rank := 'Lieutenant';
END IF;

-- Determine new responsibility based on new rank
CASE v_new_rank
    WHEN 'Private' THEN v_new_responsibility := 'Support';
    WHEN 'Corporal' THEN v_new_responsibility := 'Tactical';
    WHEN 'Sergeant' THEN v_new_responsibility := 'Squad Leader';
    WHEN 'Lieutenant' THEN v_new_responsibility := 'Platoon
Leader';
    ELSE v_new_responsibility := 'Specialized';
END CASE;

-- Update soldier rank and responsibility
UPDATE solider
SET rank = v_new_rank
WHERE id = v_soldiers(i).id;

UPDATE solider_in
SET responsibility = v_new_responsibility
WHERE id = v_soldiers(i).id AND team_number = p_team_number;

-- Log the update
DBMS_OUTPUT.PUT_LINE('Updated soldier: ' ||
v_soldiers(i).first_name || ' ' || v_soldiers(i).last_name || ' - New Rank: ' || v_new_rank || ', New
Responsibility: ' || v_new_responsibility);
END LOOP;

COMMIT;
DBMS_OUTPUT.PUT_LINE('Team ' || v_team_name || ' updated
successfully. Commander: ' || v_commander_nickname);

EXCEPTION
    WHEN e_invalid_team THEN
        DBMS_OUTPUT.PUT_LINE('Error: Invalid team number provided.');
        ROLLBACK;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END update_soldier_ranks_and_responsibilities;
/

```

תוכנית ראשית - פרוצדורות

```
-- Declare variables
DECLARE
    v_team_number team.team_number%TYPE;
BEGIN
    -- First, promote soldiers who participated in recent operations
    DBMS_OUTPUT.PUT_LINE('Executing PromoteSoldiersInRecentOperations:');
    PromoteSoldiersInRecentOperations;

    -- Get a team number (you may want to replace this with a specific team
    -- number)
    SELECT MIN(team_number) INTO v_team_number FROM team;

    -- Then, update ranks and responsibilities for a specific team
    DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Executing
update_soldier_ranks_and_responsibilities:');
    update_soldier_ranks_and_responsibilities(v_team_number);

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

נץ ונראה שאכן עבד:

```
Executing PromoteSoldiersInRecentOperations:
Promoted soldier: Emma Neuwirth from Private to Corporal
Total soldiers promoted: 1

Executing update_soldier_ranks_and_responsibilities:
Updated soldier: Vincent Crudup - New Rank: Corporal, New
Responsibility: Tactical
Updated soldier: Gloria Price - New Rank: Private, New
Responsibility: Support
Updated soldier: Cary Whitaker - New Rank: Lieutenant, New
Responsibility: Platoon Leader
Updated soldier: Ossie Stanley - New Rank: Sergeant, New
Responsibility: Squad Leader
Updated soldier: Harrison Rivers - New Rank: Sergeant, New
Responsibility: Squad Leader
Team Chapel hill updated
successfully. Commander: Franz
```

תוכניות ראשיות

תוכנית ראשית ראשונה

```

DECLARE
    p_rank VARCHAR2(20) := 'Sergeant'; -- Example rank
    p_team_number NUMBER := 101; -- Example team number
    v_result SYS_REFCURSOR;
    v_avg_soldiers_per_commander NUMBER;
    v_avg_soldier_affection NUMBER;
    v_commander_count NUMBER;
    v_non_commander_count NUMBER;
    v_total_soldiers NUMBER;
    v_team_numbers VARCHAR2(4000);
BEGIN
    DBMS_OUTPUT.PUT_LINE('Starting Main Program 1');
    DBMS_OUTPUT.PUT_LINE('-----');

    -- Call the get_rank_statistics function
    v_result := get_rank_statistics(p_rank);

    -- Fetch the results from the cursor
    FETCH v_result INTO
        v_avg_soldiers_per_commander,
        v_avg_soldier_affection,
        v_commander_count,
        v_non_commander_count,
        v_total_soldiers,
        v_team_numbers;

    -- Display the results
    DBMS_OUTPUT.PUT_LINE('Rank Statistics for ' || p_rank || ':');
    DBMS_OUTPUT.PUT_LINE('Average soldiers per commander: ' || 
v_avg_soldiers_per_commander);
    DBMS_OUTPUT.PUT_LINE('Average soldier affection: ' || 
v_avg_soldier_affection);
    DBMS_OUTPUT.PUT_LINE('Number of commanders: ' || v_commander_count);
    DBMS_OUTPUT.PUT_LINE('Number of non-commanders: ' || 
v_non_commander_count);
    DBMS_OUTPUT.PUT_LINE('Total soldiers: ' || v_total_soldiers);
    DBMS_OUTPUT.PUT_LINE('Team numbers: ' || v_team_numbers);

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

    -- Call the update_soldier_ranks_and_responsibilities procedure
    update_soldier_ranks_and_responsibilities(p_team_number);

    DBMS_OUTPUT.PUT_LINE('Main Program 1 completed successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred in Main Program 1: ' || 
SQLERRM);
END;
/

```

ואב בביוף
אבייש שחר

```
Starting Main Program 1
-----
Rank Statistics for Sergeant:
Average soldiers per commander: 3.4
Average soldier affection: 5.45
Number of commanders: 40
Number of non-commanders: 96
Total soldiers: 136
Team numbers: 104,113,117,130,134,140,146,147,161,191,200,202,213.
-----
Updated soldier: Gena Brando - New Rank: Private, New
Responsibility: Support
Team Thames Ditton updated
successfully. Commander: Christmas
Main Program 1 completed successfully
```

תוכנית ראשית שנייה

```

DECLARE
    p_team_number NUMBER := 101; -- Example team number
    v_avg_age NUMBER;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Starting Main Program 2');
    DBMS_OUTPUT.PUT_LINE('-----');

    -- Call the get_team_avg_age function
    v_avg_age := get_team_avg_age(p_team_number);

    -- Display the result
    DBMS_OUTPUT.PUT_LINE('Average age of team ' || p_team_number || ': ' ||
    ROUND(v_avg_age, 2) || ' years');

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

    -- Call the PromoteSoldiersInRecentOperations procedure
    PromoteSoldiersInRecentOperations;

    DBMS_OUTPUT.PUT_LINE('Main Program 2 completed successfully');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred in Main Program 2: ' ||
        SQLERRM);
END;
/

```

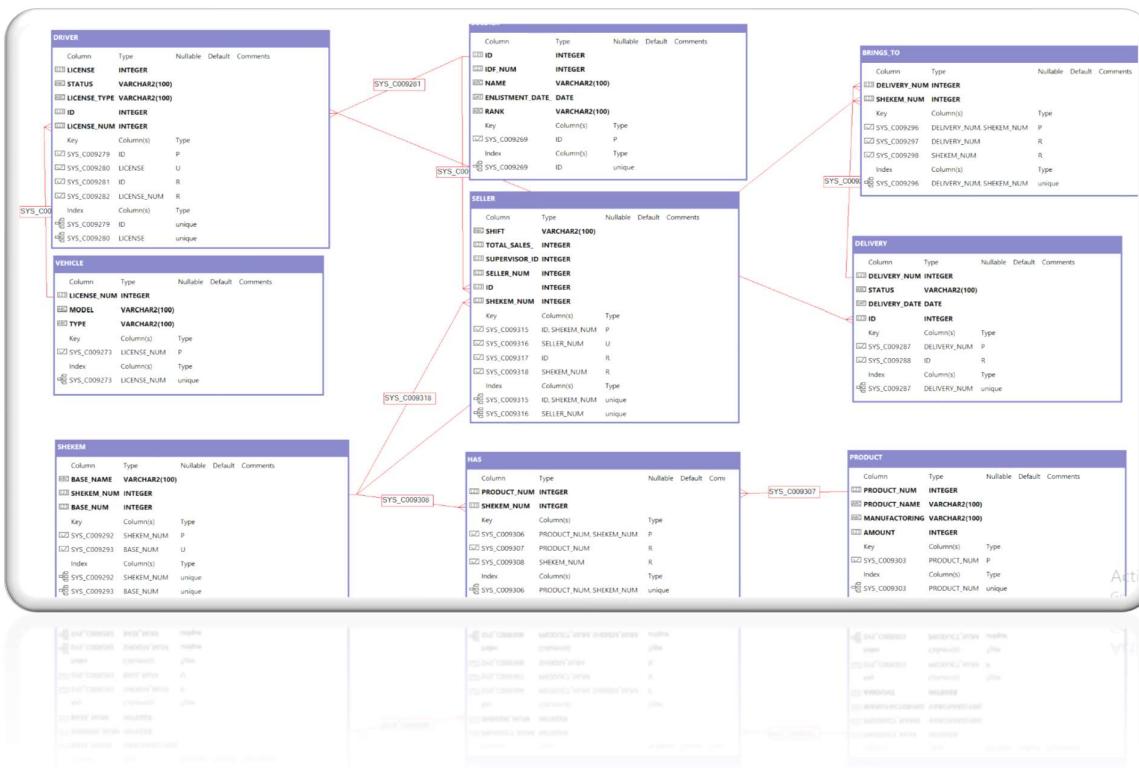
```

Starting Main Program 2
-----
Average age of team 101: 12 years
-----
Promoted soldier: Emma Neuwirth from Corporal to Sergeant
Total soldiers promoted: 1
Main Program 2 completed successfully
|

```

חלק 4

תרשים DSD של הפרויקט השני:



יצירת ERD מדיגרמת ה-DSD

:Has •

המשמעות היא שזיה קשור ולא ישות, והוא מקשר בין product ו-shekem כפ' שנייתן לראות מהמפתח (shekem_num, product_num). לא חייב להיות קשר לכל אחד זיה קשור many to many (המפתח הוא שניהם).

:Seller •

המשמעות היא שזאת ישות. המפתח שלו הוא sum_num_ID, shekem_num_ID הוא ID של solider, מכאן שהוא יורש many-to-soldier, המכון שזאת מפתח של.

:Driver •

המשמעות היא שזאת ישות, המפתח שלו הוא ID, license_num_ID הוא תכונת מפתח של.

:Product •

אם הוא ישות, עם מפתח product_num.

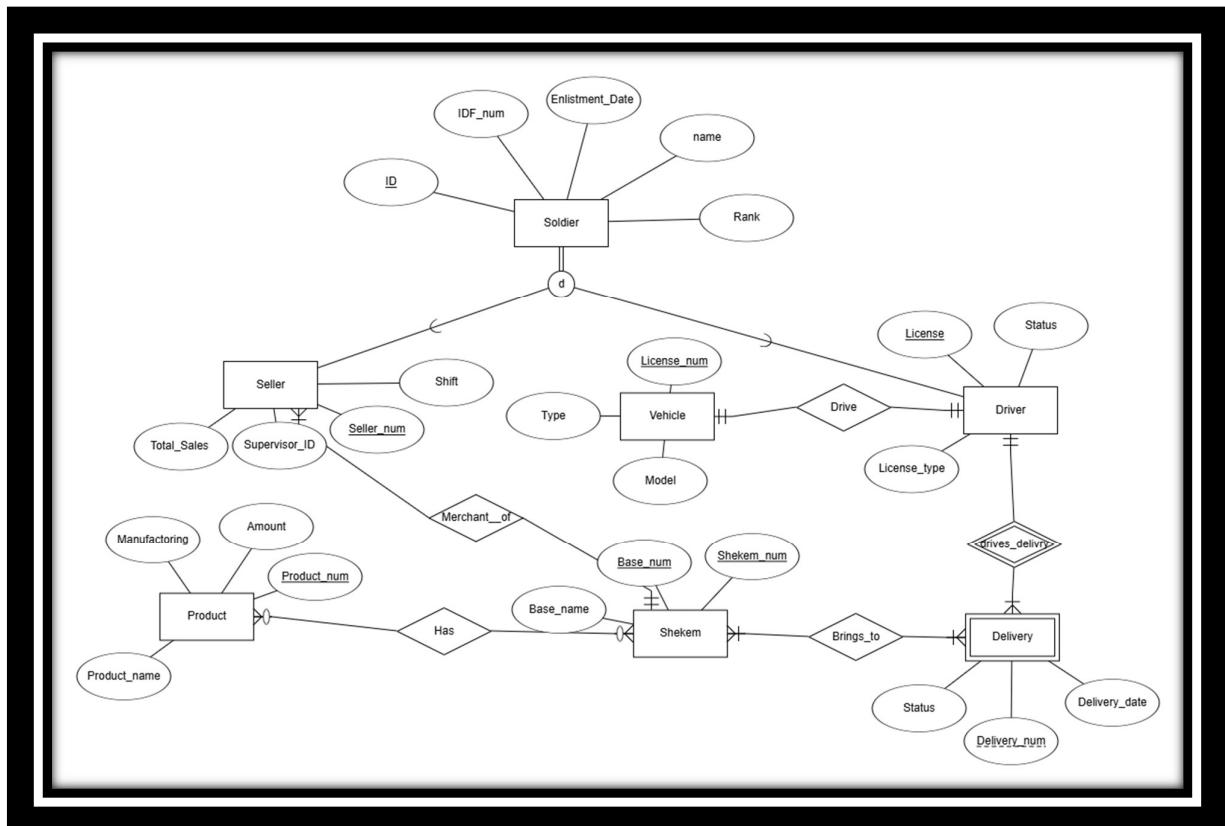
:Vehicle •

אם הוא ישות, עם מפתח licence_num.

יואב בביוף
אבי שchor

- :Solider** •
גם הוא ישות, עם מפתח ID.
- :Brings_to** •
הוא קשר many-to-many בין .shekem-delivery
- :Delivery** •
ישות, עם מפתח num ,Delivery ,ומפתחZR ID של driver. מפני ש-ID הוא unique, אצל delivery הוא null not, נראה שהוא למעשה מעשה ישות של שלושה שותחים ב-driver.
- :Shekem** •
ישות, עם מפתח num shekem_num ו-base_num שהוא unique, ככלומר גם חלק מהמפתח.
- :Merchant_of** •
ובן שקיים קשר בין Seller ל-Shekem, מפני שמכר מוכר בשק"ם, אך הוספנו את זה. זה mandatory כובן, ומוכר יכול למקומ בכמה שק"מים.
- :Drives** •
ובן גם שכל מכונית נוהגת ע"י נהג, لكن נוסף ביניהם קשר drive. מפני שבמדינת ישראל יש יקר מחיה, לכל נהג יש מכונית אחת בלבד ולהיפך. لكن זה one to one mandatory.

דיאגרמת ה-ERD שיצרנו:



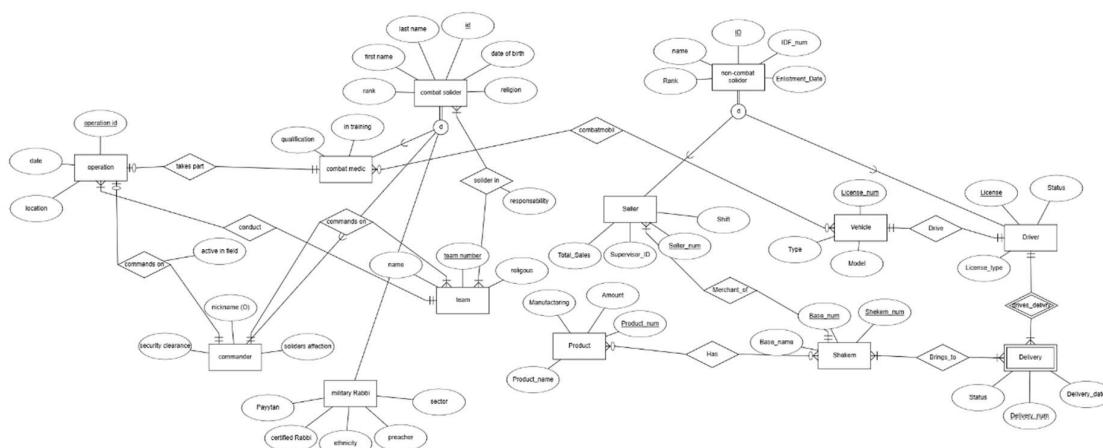
שילוב בסיסי הנתונים

cut נבצע אינטגרציה בין בין הסכמה שלנו לשכמתו של הפרויקט השני (להלן: "השני").

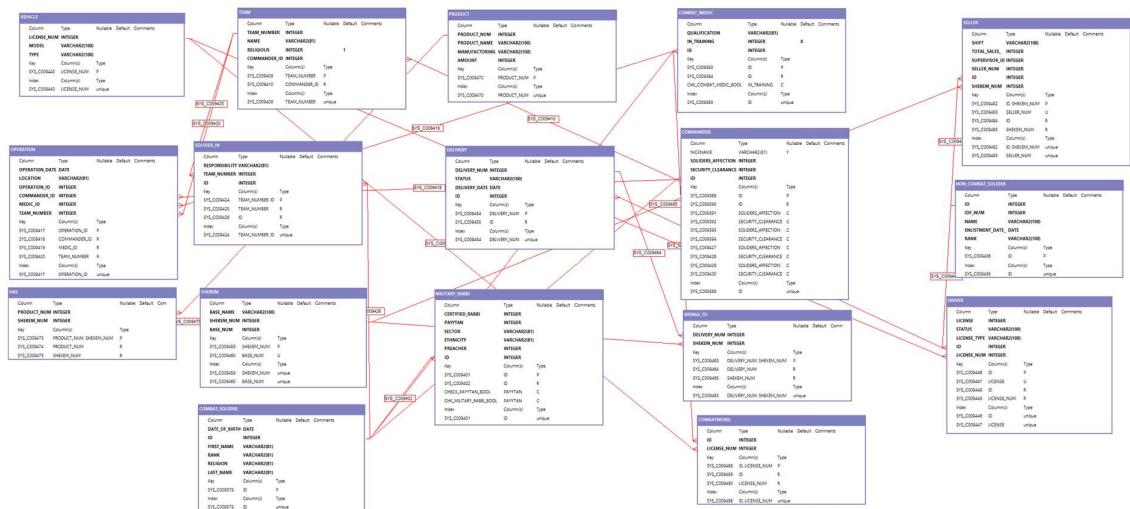
ההחלטות שקיבלנו:

- לשני יש solider, اي לkr ובהתאם לkr, נכנה את שלו combat soldier ואות של השני non-combat soldier
- יצרנו קשר בין combatmobil, שמקשר בין חובש צבאי לרכב, הוא מתאר את השתייכותו של חובש לאמבולנס מסוים. הרבה רופאים יכולים להיות בהרבה אמבולנסים ולכן זה many to many, הם יכולים שלא להיות, لكن זה גם optional.

תרשים ERD משולב



תרשים DSD משולב



שילוב בקוד

Integrate code

הקוד הנ"ל יוצר את ה-database כפ' שאנו רוצים שיראה עbor ה-database המשולב:

```
-- rename out solider
ALTER TABLE solider RENAME TO combat_solider;
-- rename their soldier
ALTER TABLE soldier RENAME TO non_combat_solider;
-- create relation between databases
CREATE TABLE combatmobil
(
    id INT NOT NULL,
    License_num INT NOT NULL,
    PRIMARY KEY (id, License_num),
    FOREIGN KEY (id) REFERENCES combat_medic(id),
    FOREIGN KEY (License_num) REFERENCES Vehicle(License_num)
);
```

```
Here comes long lines of putting values to combatmobil
commit;
```

כפי שניתן לראות, אנו למשה מושנים את solider של שני בסיסי הנתונים ויצרים את הקשר החדש, ומכוונים לו נתונים.

הכנסת נתונים

בוצע ע"י Data Generator

The screenshot shows the Data Generator interface for the COMBATMOBIL table. The configuration pane at the top shows the owner as MERGED_BASNAT, the table as COMBATMOBIL, and the number of records as 400. Below this, the table structure is defined with columns ID and LICENSE_NUM, both of type NUMBER. The 'Data' column for each has a dropdown menu with options like 'List(select ...)' and 'List(select ... from Vehicle)'. The bottom pane displays four generated SQL insert statements:

```

insert into MERGED_BASNAT.COMBATMOBIL (ID, LICENSE_NUM)
values (255, 6858460);

insert into MERGED_BASNAT.COMBATMOBIL (ID, LICENSE_NUM)
values (53, 2263322);

insert into MERGED_BASNAT.COMBATMOBIL (ID, LICENSE_NUM)
values (242, 6106673);

```

מבטים

מבט ראשון – יהלום

מבט זה מביט באגד שלנו בצורה הבאה: הסתכלות מאוחדת על כל החילאים, כאשר לכל חיל נתונים כל הפרטים שלו, כמו המבצעים בהם השתתף, והתקידים שלו (מפקד, רב צבאי וכו'):

```
CREATE OR REPLACE VIEW soldier_roles_and_operations AS
SELECT
    s.id,
    s.first_name,
    s.last_name,
    s.rank,
    -- Determine if the soldier is a Commander
    CASE WHEN c.id IS NOT NULL THEN 'Commander' END AS role1,
    -- Determine if the soldier is a Military Rabbi
    CASE WHEN mr.id IS NOT NULL THEN 'Military Rabbi' END AS role2,
    -- Determine if the soldier is a Combat Medic
    CASE WHEN cm.id IS NOT NULL THEN 'Combat Medic' END AS role3,
    -- Calculate total operation count (sum of commander and medic
operations)
    NVL(commander_ops.op_count, 0) + NVL(medic_ops.op_count, 0) AS
operation_count
FROM
    combat_soldier s
-- Join with commander table to identify commanders
LEFT JOIN commander c ON s.id = c.id
-- Join with military_Rabbi table to identify military rabbis
LEFT JOIN military_Rabbi mr ON s.id = mr.id
-- Join with combat_medic table to identify combat medics
LEFT JOIN combat_medic cm ON s.id = cm.id
-- Subquery to count operations for commanders
LEFT JOIN (
    SELECT commander_id, COUNT(*) as op_count
    FROM operation
    GROUP BY commander_id
) commander_ops ON s.id = commander_ops.commander_id
-- Subquery to count operations for medics
LEFT JOIN (
    SELECT medic_id, COUNT(*) as op_count
    FROM operation
    GROUP BY medic_id
) medic_ops ON s.id = medic_ops.medic_id;
-- see the view
select * from soldier_roles_and_operations;
```

נריץ: על מנת לראות את המבט:

	ID	FIRST_NAME	LAST_NAME	RANK	ROLE1	ROLE2	ROLE3	OPERATION_COUNT
▶	1	1 Harrison	"" Rivers	"" Sergeant	""		Combat Medic	1
	2	2 Elvis	"" de Lanie	"" Private	""		Combat Medic	1
	3	3 Vincent	"" Crudup	"" Corporal	""		Combat Medic	1
	4	4 Gloria	"" Price	"" Private	""		Combat Medic	1
	5	5 Cary	"" Whitaker	"" Lieutenant	""		Combat Medic	1
	6	6 Ahmad	"" Stills	"" Lieutenant	""		Combat Medic	1
	7	7 Derrick	"" Rispoli	"" Captain	""		Combat Medic	1
	8	8 Bridgette	"" Bancroft	"" Major General	""		Combat Medic	1
	9	9 Melanie	"" Ranger	"" Captain	""		Combat Medic	1
	10	10 Jared	"" Randal	"" Private	""		Combat Medic	1
	11	11 Cliff	"" Lowe	"" Private	""		Combat Medic	1
	12	12 Susan	"" Osment	"" General	""		Combat Medic	1
	13	13 Art	"" Shue	"" Colonel	""		Combat Medic	1
	14	14 Chalee	"" Holiday	"" Lieutenant	""		Combat Medic	1
	15	15 Shannon	"" Spinosa	"" Colonel	""		Combat Medic	1

כפי שניתן לראות, לכל חיל נתונים פרטיים, התפקידים שלו (במקרה של אלה עם ה-פ' הראשונים הוא רק חובש צבאי) וכמות המבצעים הצבאים שהשתתף בהם.

מבחן 2 – האגר השוני

מבחן זה מראה לנו פרטים שלמים על כל המשלוחים שנעשו. את פרטי החיל שמבצע את המשלוח (שם, דרגה מס' רישוי הנהיגה וכו'), פרטי הרכב, מס' הרשות, הסטטוס שלו והתאריך שלו.

```
CREATE OR REPLACE VIEW DriverDeliveryInfo AS
SELECT
    d.ID,
    s.name AS Driver_Name,
    s.Rank AS Driver_Rank,
    d.License AS Driver_License,
    d.License_type,
    v.Model AS Vehicle_Model,
    v.Type AS Vehicle_Type,
    del.Delivery_num,
    del.Status AS DeliveryStatus,
    del.Delivery_date
FROM
    Driver d
-- Join with non_combat_solidier to get driver's personal information
JOIN non_combat_solidier s ON d.ID = s.ID
-- Join with Vehicle to get information about the vehicle assigned to the
driver
JOIN Vehicle v ON d.License_num = v.License_num
-- Left join with Delivery to include all drivers, even those without
deliveries
LEFT JOIN Delivery del ON d.ID = del.ID;
```

נ裏ץ עם נקבל:

	ID	DRIVER_NAME	DRIVER_RANK	DRIVER_LICENSE	LICENSE_TYPE	VEHICLE_MODEL	VEHICLE_TYPE	DELIVERY_NUM	DELIVERYSTATUS	DELIVERY_DATE
▶	1	353343650 Evangelin	... Rav Aluf	... 621925253 C	... Rogue	... Westfield	...	32285915	Packaging	... 16/02/2025
	2	858873709 Walliw	... Samal Rishon	... 1047199520 A	... Tracker	... Hummer	...	531756019	PickedUp	... 14/01/2003
	3	476745216 Arlan	... Rav Turai	... 2134192591 C	... Zetao	... Morgan	...	926925120	OutforDelivery	... 05/02/2019
	4	278413709 Jesse	... Segev Mishneh	... 619613627 A	... Strada	... KTM	...	450092955	Delivered	... 17/09/2013
	5	572888847 Joshua	... Rav Seren	... 649708092 C	... Dart	... Opel	...	32108321	ReturnedtoSender	... 20/05/2002
	6	538200293 Dedie	... Samal Rishon	... 1840132101 A	... Zohr	... Fiat	...	953866905	Delayed	... 24/04/2000
	7	861096098 Rice	... Rav Turai	... 1413989817 A	... Puma	... Thunder Power	...	773048663	Shipped	... 30/12/2000
	8	757334285 Brandie	... Samal	... 373958388 B	... Koenigsegg	... KTM	...	453521520	ReturnedtoSender	... 10/07/2001
	9	509670721 Otho	... Seren	... 364208206 C	... Telluride	... Honda	...	572284234	AttemptedDelivery	... 09/02/2003
	10	756949281 Rivkah	... Rav Turai	... 1502442967 D	... Tharson	... GTA Hispano	...	403985452	Packaging	... 27/11/2021
	11	379370561 Derwin	... Rav Seren	... 796605740 A	... BrZ	... Volkswagen	...	200955157	Shipped	... 11/05/2017
	12	620670112 Giffy	... Turai	... 206733726 B	... Jimmy	... Vector Motors	...	485340533	PickedUp	... 25/07/2018
	13	787587330 Elsinore	... Rav Aluf	... 966588751 C	... Compass	... Ascaris	...	270974268	PickedUp	... 23/11/2019
	14	7599704 Horatio	... Turai	... 936736989 A	... Zonda	... Saturn	...	712643220	PickedUp	... 10/02/2020
	1E	21140417C Vandalia	... Tami Diskha	... 110225217 C	... Indiana	... Linkster	...	214027010	ArrivedInWarehouse	... 20/11/2024

כפי שניתן לראות, לכל משלוח ישנו פרטי המשלוח, נהג הרכב, ופרטי הרכב המבצע את המשלוח.

שאילתות

מבחן ראשון – שאלה ראשונה

שאילתת זו מביאה לנו את רשימת כל החילים, עם כמות התפקידים שהם מלאים וכמות המבצעים שהם ביצעו:

```
SELECT
    id,
    first_name || ' ' || last_name AS full_name,
    rank,
    -- Count the number of non-null roles
    (CASE WHEN role1 IS NOT NULL THEN 1 ELSE 0 END +
     CASE WHEN role2 IS NOT NULL THEN 1 ELSE 0 END +
     CASE WHEN role3 IS NOT NULL THEN 1 ELSE 0 END) AS role_count,
    operation_count
FROM soldier_roles_and_operations
ORDER BY operation_count DESC;
```

נראז:

	ID	FULL_NAME	RANK	ROLE_COUNT	OPERATION_COUNT
▶	1	401 Colleen Hoffman	Sergeant	1	2
	2	278 Ed Idol	Sergeant	1	2
	3	1 Harrison Rivers	Sergeant	1	1
	4	2 Elvis de Lanie	Private	1	1
	5	3 Vincent Crudup	Corporal	1	1
	6	4 Gloria Price	Private	1	1
	7	5 Cary Whitaker	Lieutenant	1	1
	8	6 Ahmad Stills	Lieutenant	1	1
	9	7 Derrick Rispoli	Captain	1	1
	10	8 Bridgette Bancroft	Major General	1	1
	11	9 Melanie Ranger	Captain	1	1
	12	10 Jared Randal	Private	1	1
	13	11 Cliff Lowe	Private	1	1
	14	12 Susan Osment	General	1	1
	15	13 Art Shue	Colonel	1	1

מבט ראשון – שאלה שניה

שאלתנו זו מביאה לנו את החיללים מכל תפקיד (מפקד, רב צבאי וכו') שהיו בכמות הרבה ביותר של מבצעים:

```
WITH unpivoted_roles AS (
    SELECT id, first_name, last_name, rank, operation_count, role
    FROM soldier_roles_and_operations
    UNPIVOT (role FOR role_column IN (role1, role2, role3))
    WHERE role IS NOT NULL
    UNION ALL
    -- Add 'Regular Soldier' for those with no special roles
    SELECT id, first_name, last_name, rank, operation_count, 'Regular
    Soldier' AS role
    FROM soldier_roles_and_operations
    WHERE role1 IS NULL AND role2 IS NULL AND role3 IS NULL
)
-- Main query to find top performer in each role
SELECT *
FROM (
    SELECT
        role,
        id,
        first_name || ' ' || last_name AS full_name,
        rank,
        operation_count,
        -- Rank soldiers within each role based on operation count
        ROW_NUMBER() OVER (PARTITION BY role ORDER BY operation_count DESC)
    AS operations_rank
    FROM unpivoted_roles
    WHERE role IN ('Combat Medic', 'Commander', 'Military Rabbi', 'Regular
    Soldier')
)
WHERE operations_rank = 1 -- first place!
ORDER BY role;
```

ונרץ:

	ROLE	ID	FULL_NAME	RANK	OPERATION_COUNT	OPERATIONS_RANK
▶	1 Combat Medic	278	Ed Idol	... Sergeant ...	2	1
	2 Commander	401	Colleen Hoffman	... Sergeant ...	2	1
	3 Military Rabbi	894	Bradley Esposito	... Major General ...	0	1

mbt שני – שאילתת ראשונה

שאילתת זו מספקת לנו מידע אודות כל נהג שביצע משלוח: שם, דרגתם, סוג הרכב שלהם, כמוות המשלוחים שהם מבצעים, כמוות המשלוחים שכבר ביצעו ורמת הניסיון שלהם (על פי כמהות המשלוחים שביצעו).

```
SELECT
    Driver_Name,
    Driver_Rank,
    Vehicle_Model,
    COUNT(DISTINCT Delivery_num) AS Total_Deliveries,
    SUM(CASE WHEN DeliveryStatus = 'Completed' THEN 1 ELSE 0 END) AS
    Completed_Deliveries,
    CASE
        WHEN COUNT(DISTINCT Delivery_num) > 10 THEN 'Experienced'
        WHEN COUNT(DISTINCT Delivery_num) > 5 THEN 'Intermediate'
        ELSE 'Novice'
    END AS Driver_Experience
FROM
    DriverDeliveryInfo
GROUP BY
    Driver_Name, Driver_Rank, Vehicle_Model
HAVING
    COUNT(DISTINCT Delivery_num) > 0
ORDER BY
    COUNT(DISTINCT Delivery_num) DESC;
```

ונרץ:

	DRIVER_NAME	DRIVER_RANK	VEHICLE_MODEL	TOTAL_DELIVERIES	COMPLETED_DELIVERIES	DRIVER_EXPERIENCE
► 1	Vachel	... Turai Rishon	... Elantra	...	1	0 Novice
2	Vinni	... Segen Rishon	... Telluride	...	1	0 Novice
3	Virginia	... Seren	... Tiago	...	1	0 Novice
4	Mirilla	... Segen Mishneh	... Zonda	...	1	0 Novice
5	Moshe	... Aluf Mishne	... ProAce	...	1	0 Novice
6	Ara	... Segen Rishon	... Cascada	...	1	0 Novice
7	Gabriele	... Segen	... Replay	...	1	0 Novice
8	Yolande	... Aluf	... Crosspolo	...	1	0 Novice
9	Dael	... Segen Rishon	... Escape	...	1	0 Novice
10	Rice	... Rav Turai	... Puma	...	1	0 Novice
11	Ardisj	... Tat Aluf	... Esprit	...	1	0 Novice
12	Murvyn	... Rav Aluf	... Maxima	...	1	0 Novice
13	Conni	... Segen Mishneh	... Revero	...	1	0 Novice
14	Sondra	... Rav Aluf	... Galant	...	1	0 Novice
15	Oliver	... Rav Samal	... Rio	...	1	0 Novice

GBT שני – שאילתת שניה

שאילתת זו מספקת מידע על רכבים – עבור כל סוג רכב, כמה משלוחים בוצעו בסוג זה של רכב, ומתי הפעם הראשונה בה נעשה בו שימוש.

```
SELECT
    d.Vehicle_Type,
    MIN(d.Delivery_date) AS first_use,
    COUNT(Delivery_num) AS Total_Deliveries
FROM
    DriverDeliveryInfo d
GROUP BY
    Vehicle_Type
HAVING
    COUNT(Delivery_num) > 0
ORDER BY
    Total_Deliveries DESC, Vehicle_Type;
```

ונרץ:

	VEHICLE_TYPE	FIRST_USE	TOTAL_DELIVERIES
▶ 1	Honda	... 21/03/2000 ...	12
2	Bugatti	... 13/04/2000 ...	9
3	Plymouth	... 23/12/2003 ...	9
4	Audi	... 04/04/2001 ...	8
5	Kleemann	... 02/04/2000 ...	8
6	Lexus	... 27/06/2001 ...	8
7	Nissan	... 12/11/2004 ...	8
8	Roewe	... 11/11/2008 ...	8
9	Volkswagen	... 09/05/2010 ...	8
10	Bristol	... 28/11/2003 ...	7
11	Dacia	... 21/12/2008 ...	7
12	FAW	... 11/09/2001 ...	7
13	Lincoln	... 30/09/2011 ...	7
14	RAM	... 12/01/2004 ...	7
15	Vector Motors	... 26/09/2008 ...	7

תפֵּן וְגִשְׁלָם הַקּוֹנְדָס

RELATIONAL DATABASE

