

[תאריך]

נדב צימרמן

[שם החברה]

[כתובת החברה]

תוכן

[שלב 4 2](#_Toc172192364)

[4.1 יצירת הDSD- של המערכת הנוספת 2](#_Toc172192365)

[4.2 יצירת הERD- של המערכת הנוספת 3](#_Toc172192366)

[4.3 יצירת הERD- המשולב של שתי המערכות 4](#_Toc172192367)

[4.4 יצירת הDSD- המשולב של שתי המערכות 5](#_Toc172192368)

[4.5 יצירת הסכמה החדשה שמשלבת בין שני בסיסי הנתונים 6](#_Toc172192369)

[4.6 הוספת מבטים ושאילתות עליהן 8](#_Toc172192370)

[4.6.1 – view מנקודת המבט של האגף שקיבלנו ("צד כחול") 8](#_Toc172192371)

[4.6.2 – שאילתה 1: מציאת יחידות עם מספר החיילים והמשאבים הגבוה ביותר 10](#_Toc172192372)

[4.6.3 – שאילתה 2: סיכום משאבים וחיילים לפי יחידות עם סטטוס פרויקט 11](#_Toc172192373)

[4.6.4 – view מנקודת המבט של האגף שלנו ("צד אדום") 12](#_Toc172192374)

[4.6.5 – שאילתה 1: מציאת כל האנשים עם רישומי מעצר ושיחות טלפון 14](#_Toc172192375)

[4.6.6 – שאילתה 2: סיכום משאבים וחיילים לפי יחידות עם סטטוס פרויקט 15](#_Toc172192376)

# שלב 4

## 4.1 יצירת הDSD- של המערכת הנוספת

#### הסבר מילולי:

בהתאם להנחיות, קיבלנו את קובץ ה-backup של המערכת המכילה את המידע שמצוי ב"אגף התכנון" של צה"ל. בתחילה בנינו את המחלקות השונות על ידי צפייה בשדות השונים של כל טבלה תוך הקפדה על סימון המפתחות והמפתחות הזרים בכל טבלה.

לאחר מכן התבוננו בקשרים השונים שבין הטבלאות – כך למשל במקרה בו כתוב:  
(X) REFERENCES TableB(Y) foreignkey

ברור כי קיים קשר הפונה מטבלה זו לטבלה B.

#### הdsd שפיענחנו מתוך גיבוי הטבלאות שקיבלנו:

תמונה שמכילה טקסט, תרשים, עיגול, גופן

התיאור נוצר באופן אוטומטימביצוע פעולות אלו הסקנו כי תרשים הDSD המקורי הוא כדלהלן:

## 4.2 יצירת הERD- של המערכת הנוספת

#### הסבר מילולי

יצירת המחלקות השונות והשדות של כל מחלקה נעשתה בדיוק לפי האמור בחלק הקודם. האתגר היה באפיון היחסים השונים שבין הישויות במערכת. לדוגמא מהם הקשרים שבן הישויות – רבים לרבים יחיד לרבים וכו', האם קיימים ישויות חלשות? ירושה בין ישויות וכו'

במקרה כמו הטבלה היה קל לראות כי קיימים רק שני מפתחות זרים ללא כל מפתח רגיל, מכאן כי טבלה זו הינה קשר של רבים לרבים.

באופן דומה: Unit - Resources: אחד-לרבים (Unit יכולה להכיל משאבים רבים)

Projects - Task: אחד-לרבים (פרויקט יכול להכיל משימות רבות),

Unit - Excellence: אחד-לרבים (יחידה יכולה להכיל הצטיינויות רבות)

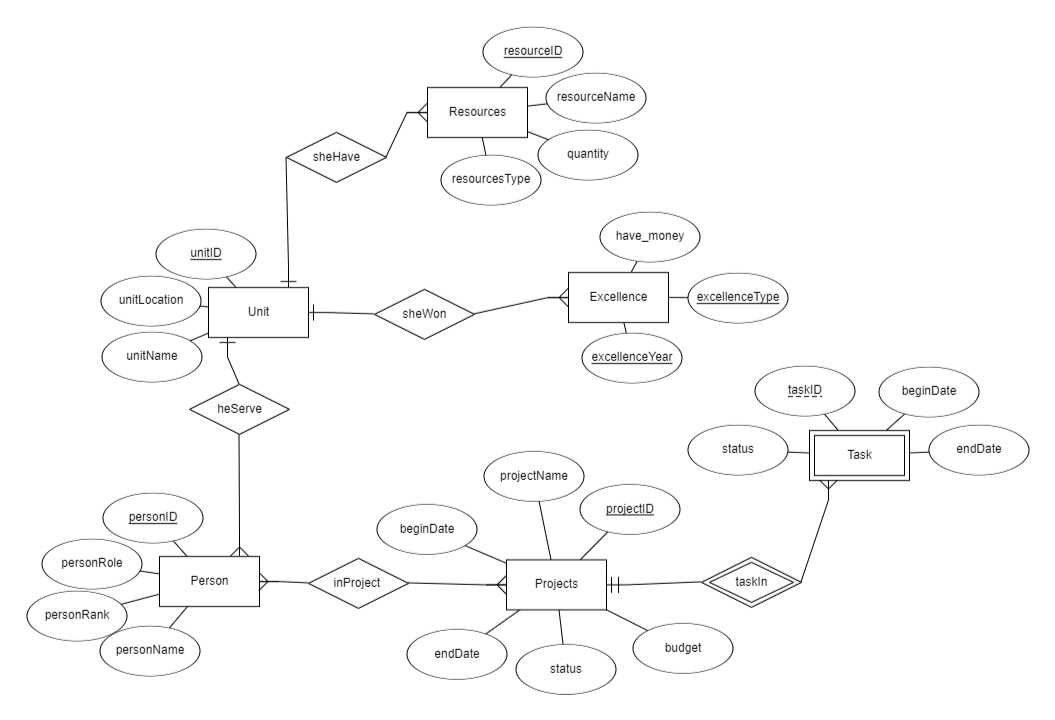
Unit - Person: אחד-לרבים (יחידה יכולה להכיל אנשים רבים)

Projects - inProject - Person: רבים-לרבים (פרויקט יכול להכיל אנשים רבים, ואדם יכול להשתתף בפרויקטים רבים).

בנוסף הסקנו כי הישות Task היא ישות חלשה, משום שהמפתח שלה מורכב בחלקו מן המפתח הזר של הישות Projects מה שמעיד כי אין קיום לTask במנותק מפרויקט.

היחס inProject אינו יחס חלש משום שאינו מקשר בין ישות חלשה לישות אחרת, אלא מהווה קישור בין שתי הטבלאות Person ו- Projects, לכן אינו מכיל אף ישות אחרת ולא מפתח מלבד שני המפתחות הזרים.

#### הerd המפוענח



## 4.3 יצירת הERD- המשולב של שתי המערכות

#### הסבר מילולי

בשלב זה של יצירת הERD המשולב של שני הפרויקטים, ישבנו לחשוב מהי נקודת המפגש בין שני הפרויקטים ואיזה חיבור ניתן למצוא בניהם. לאחר הרבה מחשבות והתייעצויות החלטנו שמאחר ושני הפרויקטים הם בנושא צבאי אולם אנחנו בחרנו להתמקד ב"צד אדום" (מחבלים ואויב) ולעומתנו הם בחרו להתמקד דווקא ב"צד כחול" (צה"ל עצמו) מצאנו שנקודת המפגש בין הפרויקטים תהיה יצירת טבלה שמקשרת בן הפרויקטים. טבלה שמכילה שתי עמודות של מפתחות זרים ושניהם יחד הם מפתח מקומי. עמודה אחת תהיה HumanId (תעודות הזהות של הפלסטינאים בעלי מסוכנות גבוהה שאותן צה"ל רוצה לעצור) והעמודה השנייה תהיה UnitId (מספר מזהה של יחידות צבאיות) כך שהטבלה מצמידה בין יחידה צבאית לתעודת זהות של פלסטינאי שתפקידה לעצור אותו. מלבד חיבור זה לא מצאנו ממשקים נוספים מאחר ושני הפרויקטים אכן מתעסקים בתחומים מאוד נפרדים

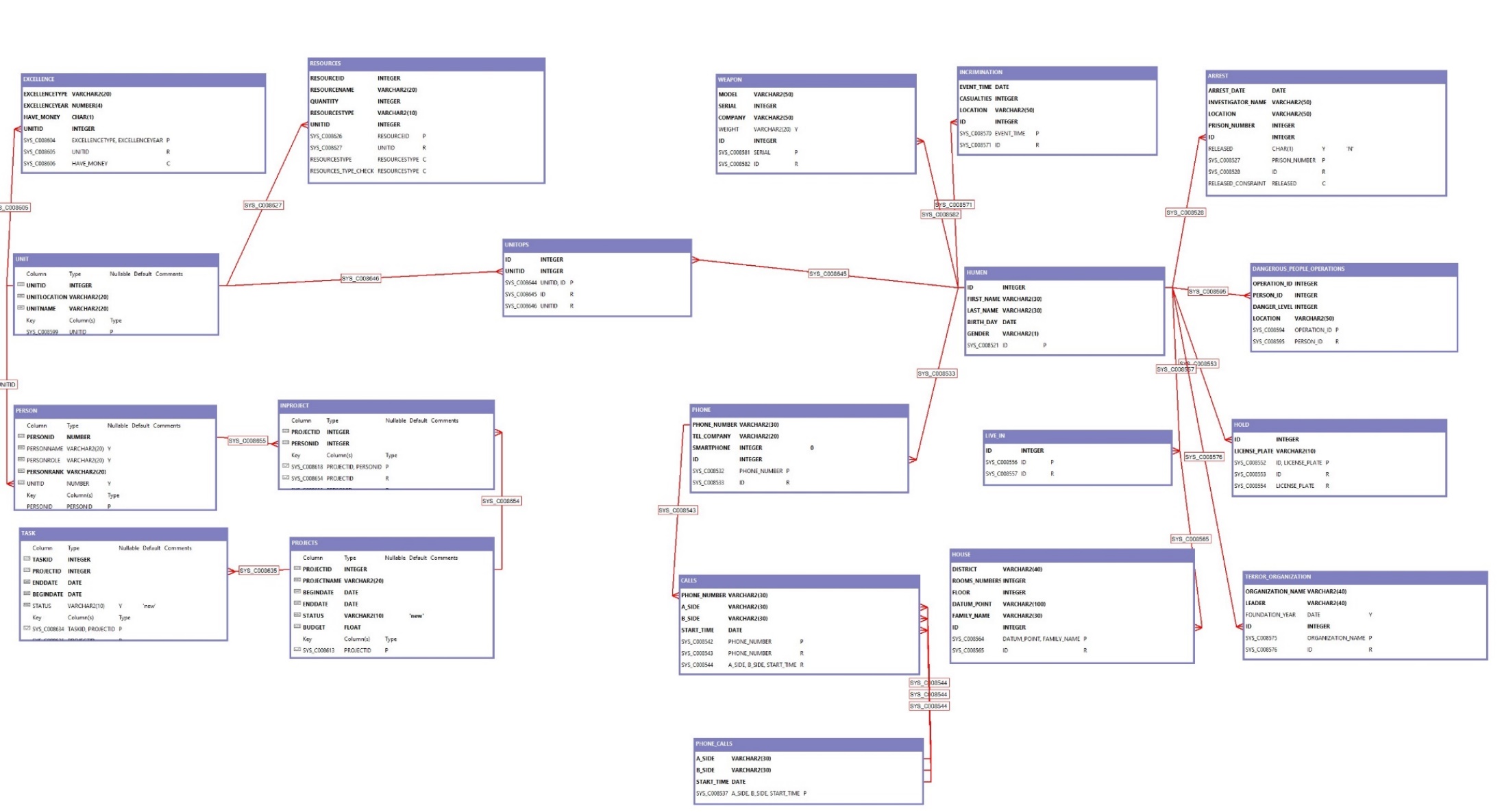
#### הerd המשולב לאחר הוספת האינטגרציה:

תמונה שמכילה ציור, שרטוט, תרשים, אומנות קווים

התיאור נוצר באופן אוטומטי

## 4.4 יצירת הDSD- המשולב של שתי המערכות

#### הdsd המשולב לאחר הוספת האינטגרציה:



## 4.5 יצירת הסכמה החדשה שמשלבת בין שני בסיסי הנתונים

#### הסבר מילולי

כדי ליצור את הסכמה החדשה בהתאם למה שתכננו בשלב יצירת הERD, יצרנו טבלה חדשה שמחברת בין הdb שאנו בנינו לאורך כל הפרויקט (טבלת Human) לבין הdb שקיבלנו בשלב הזה (טבלת Unit).  
לטבלה החדשה קראנו בשם Unitops ותפקידה הוא לאחד בין יעדי מעצר ובין היחידות האמונות על ביצוע המעצר של כל פעיל:

create table unitOps

(

id INT NOT NULL,

unitID INT NOT NULL,

PRIMARY KEY (unitID, id),

FOREIGN KEY (id) REFERENCES humen(ID),

FOREIGN KEY (unitID) REFERENCES Unit(unitID)

);

לצורך בחירת האנשים המסוכנים מתוך טבלת Human בנינו טבלת עזר בדומה לטבלה שהרצנו בשלב הקודם, אשר מחשבת את 100 היעדים המסוכנים ביותר מתוך טבלת Human ושומרת אותם בטבלה בשם dangerous\_people\_operations כולל שמירת הנתונים במאגר- ביצוע commit. (חשוב לשים לב! לאחר הכנסת 100 היעדים לא ניתן לבצע את הפונקציה פעם נוספת עד לריקון הטבלה כי הפונקציה תחשב שוב את אותם האנשים וההכנסה כי הid של כל יעד הינו מפתח, כך לא ייתכנו ערכים כפולים בטבלה.) ומתוך טבלת העזר הזו בחרנו את האנשים שאנו רוצים לקשר ליחידות על מנת שיעצרו אותם.  
לצורך בחירת היחידות אותן נכניס לטבלת unitOps בחרנו ערכים רנדומליים מתוך הטבלה של היחידות (Unit) וצימדנו אותם לאנשים המסכונים אותם אנו רוצים לעצור

לצורך הכנסת הנתונים לטבלת unitops כתבנו את הקוד הבא, אשר בוחר את האנשים המסוכנים ביותר ומכניס אותם לטבלה (מוגבל ל1,000 שורות):

*-- הכנסת נתונים לטבלת unitops עם שילובים רנדומליים*

INSERT INTO unitops (ID, unitID) *-- הכנסת נתונים לטבלה*

SELECT h.Id, u.unitID *-- בחירת העמודות Id ו-unitID מהשילוב הרנדומלי*

FROM (

*-- בחירת מזהי אנשים מסוכנים*

SELECT person\_id AS Id

FROM dangerous\_people\_operations

WHERE danger\_level > 50 *-- רק אנשים עם רמת סיכון מעל 50*

) h *-- מתן שם לטבלה הזמנית כ-h*

CROSS JOIN (

*-- בחירת 200 יחידות באופן רנדומלי*

SELECT unitID

FROM (

SELECT unitID

FROM Unit

ORDER BY DBMS\_RANDOM.VALUE *-- סידור רשומות היחידות באופן רנדומלי*

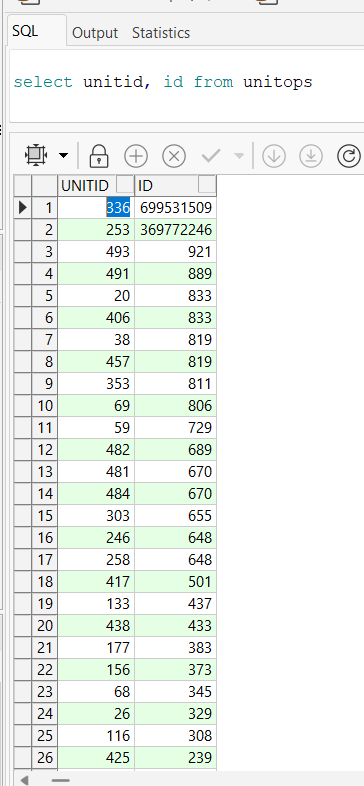
)

WHERE ROWNUM <= 5 *-- בחירת 5 היחידות באופן רנדומלי*

) u *-- מתן שם לטבלה הזמנית כ-u*

WHERE ROWNUM <= 3000; *-- הגבלת התוצאה ל3000 הרשומות הראשונות*

#### דוגמת הדפסה של הטבלה החדשה שיצרנו



## 4.6 הוספת מבטים ושאילתות עליהן

### 4.6.1 – view מנקודת המבט של האגף שקיבלנו ("צד כחול")

#### הסבר מילולי:

יצרנו View מנקודת המבט של האגף שקיבלנו בשם MilitaryUnitOverviewכדי לשלב מידע ממספר טבלאות מרכזיות בבסיס הנתונים של האגף שקיבלנו: יחידות, הצטיינות, משאבים, חיילים, פרויקטים ומשימות. ה- View שיצרנו מספק תמונה רחבה ומקיפה על כל יחידה, כולל מיקומה, שמה, סוגי ההצטיינות שקיבלה, המשאבים שברשותה, החיילים שמרכיבים אותה, הפרויקטים והמשימות הקשורות אליה. המטרה היא לאחד מידע ממספר טבלאות כדי לפשט שאילתות מורכבות, לתמוך בניהול משאבים ומעקב אחר החיילים ובניהול פרויקטים ומשימות בצורה יעילה. בזכות ה-View מנהלים יכולים לראות את הפרויקטים הפעילים, המשאבים והחיילים המוקצים לכל פרויקט, ולקבל החלטות מבוססות נתונים בצורה מסודרת ויעילה.

#### שאילתא ליצירת הview:

CREATE OR REPLACE VIEW MilitaryUnitOverview AS

SELECT

u.unitid,

u.unitlocation,

u.unitname,

e.excellencetype,

e.excellenceyear,

e.have\_money,

r.resourceid,

r.resourcename,

r.resourcestype,

r.quantity,

p.personid,

p.personname,

p.personrole,

p.personrank,

pr.projectid,

pr.projectname,

pr.begindate AS project\_begindate,

pr.enddate AS project\_enddate,

pr.status AS project\_status,

pr.budget,

ti.taskid,

ti.begindate AS task\_begindate,

ti.enddate AS task\_enddate,

ti.status AS task\_status

FROM

Unit u

LEFT JOIN

Excellence e ON u.unitid = e.unitid

LEFT JOIN

Resources r ON u.unitid = r.unitid

LEFT JOIN

Person p ON u.unitid = p.unitid

LEFT JOIN

inProject ip ON p.personid = ip.personid

LEFT JOIN

Projects pr ON ip.projectid = pr.projectid

LEFT JOIN

Task ti ON pr.projectid = ti.projectid

#### דוגמא לפלט של טבלת הview שיצרנו:

תמונה שמכילה טקסט, צילום מסך, מספר

התיאור נוצר באופן אוטומטי

### 4.6.2 – שאילתה 1: מציאת יחידות עם מספר החיילים והמשאבים הגבוה ביותר

#### הסבר מילולי:

שאילתה זו מוצאת את חמש היחידות עם המספר הגבוה ביותר של חיילים ומשאבים. השאילתה מחזירה את מזהה היחידה, שם היחידה, מספר החיילים ומספר המשאבים שיש לכל יחידה. התוצאות מקובצות לפי יחידות וממויינות לפי מספר החיילים והמשאבים בסדר יורד. שאילתה זו מסייעת למפקדים לזהות את היחידות עם העומס הגבוה ביותר של חיילים ומשאבים, וכך מאפשרת למפקדים להתמקד ביחידות אלו ולבצע החלטות ניהוליות בהתאם.

#### השאילתה:

SELECT

unitid,

unitname,

total\_soldiers,

total\_resources

FROM (

SELECT

unitid,

unitname,

COUNT(DISTINCT personid) AS total\_soldiers,

COUNT(DISTINCT resourceid) AS total\_resources

FROM

MilitaryUnitOverview

GROUP BY

unitid,

unitname

ORDER BY

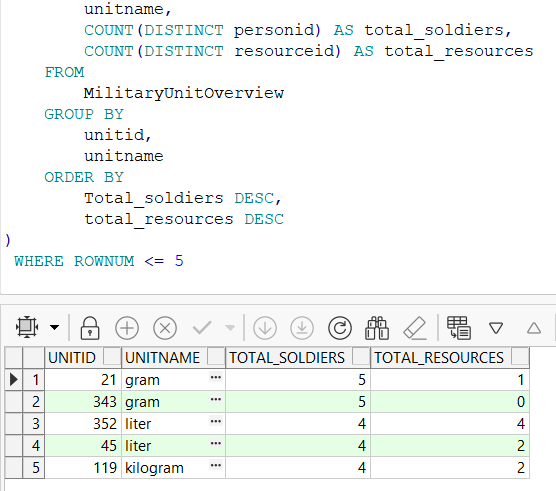
Total\_soldiers DESC,

total\_resources DESC

)

WHERE ROWNUM <= 5

#### דוגמא לפלט של השאילתה שיצרנו:



### 4.6.3 – שאילתה 2: סיכום משאבים וחיילים לפי יחידות עם סטטוס פרויקט

#### הסבר מילולי:

שאילתה זו סופרת את מספר המשאבים והחיילים בכל יחידה ומציגה את סטטוס הפרויקט. היא מחזירה את מזהה היחידה, שם היחידה, מספר המשאבים, מספר החיילים וסטטוס הפרויקט. השאילתה מבוצעת לפי חלוקה ליחידות המאפשרת לקבל מבט כולל על המשאבים ומצב החיילים בכל יחידה, יחד עם סטטוס הפרויקט הנוכחי שלה. שאילתה זו מסייעת למפקדים להבין את כמות המשאבים והחיילים המעורבים בפרויקטים השונים של כל יחידה.

#### השאילתה:

SELECT

unitid,

unitname,

COUNT(DISTINCT resourceid) AS resource\_count,

COUNT(DISTINCT personid) AS employee\_count,

project\_status

FROM

MilitaryUnitOverview

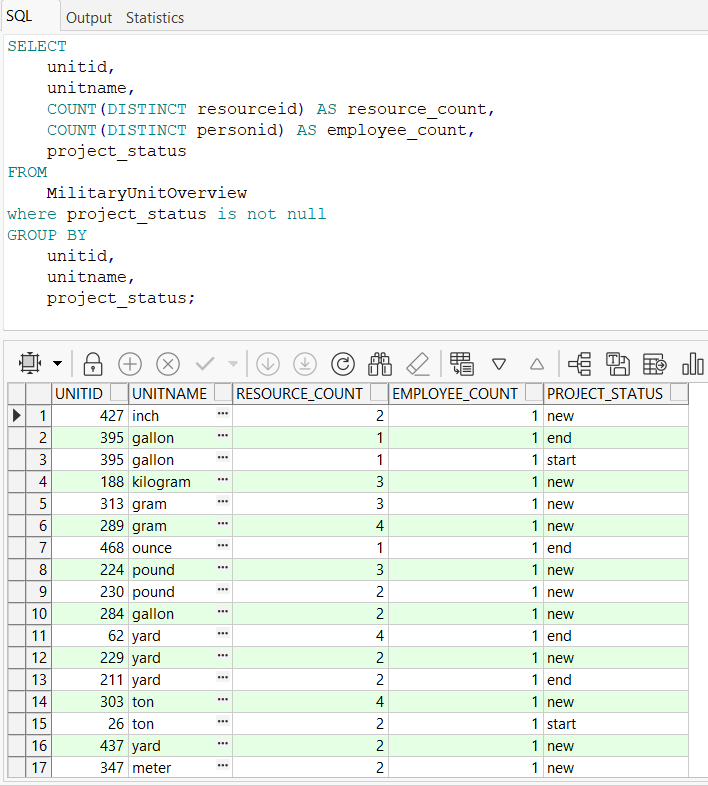
GROUP BY

unitid,

unitname,

project\_status;

#### דוגמא לפלט של השאילתה שיצרנו:



### 4.6.4 – view מנקודת המבט של האגף שלנו ("צד אדום")

#### הסבר מילולי:

יצרנו View מנקודת המבט של האגף שלנו בשם CriminalIntelDashboard כדי לשלב מידע ממספר טבלאות מרכזיות בבסיס הנתונים הקשורות לרישומים פליליים, כולל אנשים, מעצרים, החזקות נשק, אירועי הפללה, פרטי טלפונים, ארגוני טרור, כלי נשק, שיחות טלפון, מקום מגורים ורכבים. המטרה של ה-View היא לספק תמונה רחבה ומקיפה על כל פרט רלוונטי לאדם בעל רישום פלילי, ולהנגיש את המידע הרלוונטי בצורה פשוטה ונוחה למשתמשים שונים במערכת. ה-View יכול לשמש מנהלים, חוקרים ואנשי מקצוע נוספים שזקוקים לגישה מהירה ולמידע מקיף על חשודים, כולל פרטים אישיים, אירועים פליליים, פרטי טלפונים, נתוני תקשורות, פרטי רכבים ובתים, ונתונים נוספים.

#### שאילתא ליצירת הview:

CREATE OR REPLACE VIEW CriminalIntelDashboard AS

SELECT

h.id AS person\_id,

h.first\_name,

h.last\_name,

h.birth\_day,

h.gender,

a.arrest\_date,

a.investigator\_name,

a.location AS arrest\_location,

a.prison\_number,

ho.license\_plate,

i.event\_time,

i.casualties,

i.location AS incrimination\_location,

p.phone\_number,

p.tel\_company,

p.smartphone,

t.organization\_name,

t.leader,

t.foundation\_year,

w.model AS weapon\_model,

w.serial AS weapon\_serial,

w.company AS weapon\_company,

c.a\_side,

c.b\_side,

c.start\_time AS call\_start\_time,

v.license\_plate AS vehicle\_license\_plate,

v.color AS vehicle\_color,

v.car\_company AS vehicle\_company,

v.Date\_Manuacturing AS vehicle\_date\_manufacturing,

hs.district,

hs.rooms\_numbers,

hs.floor,

hs.datum\_point,

hs.family\_name

FROM

HUMEN h

LEFT JOIN

ARREST a ON h.id = a.id

LEFT JOIN

HOLD ho ON h.id = ho.id

LEFT JOIN

INCRIMINATION i ON h.id = i.id

LEFT JOIN

PHONE p ON h.id = p.id

LEFT JOIN

TERROR\_ORGANIZATION t ON h.id = t.id

LEFT JOIN

WEAPON w ON h.id = w.id

LEFT JOIN

CALLS c ON p.phone\_number = c.phone\_number

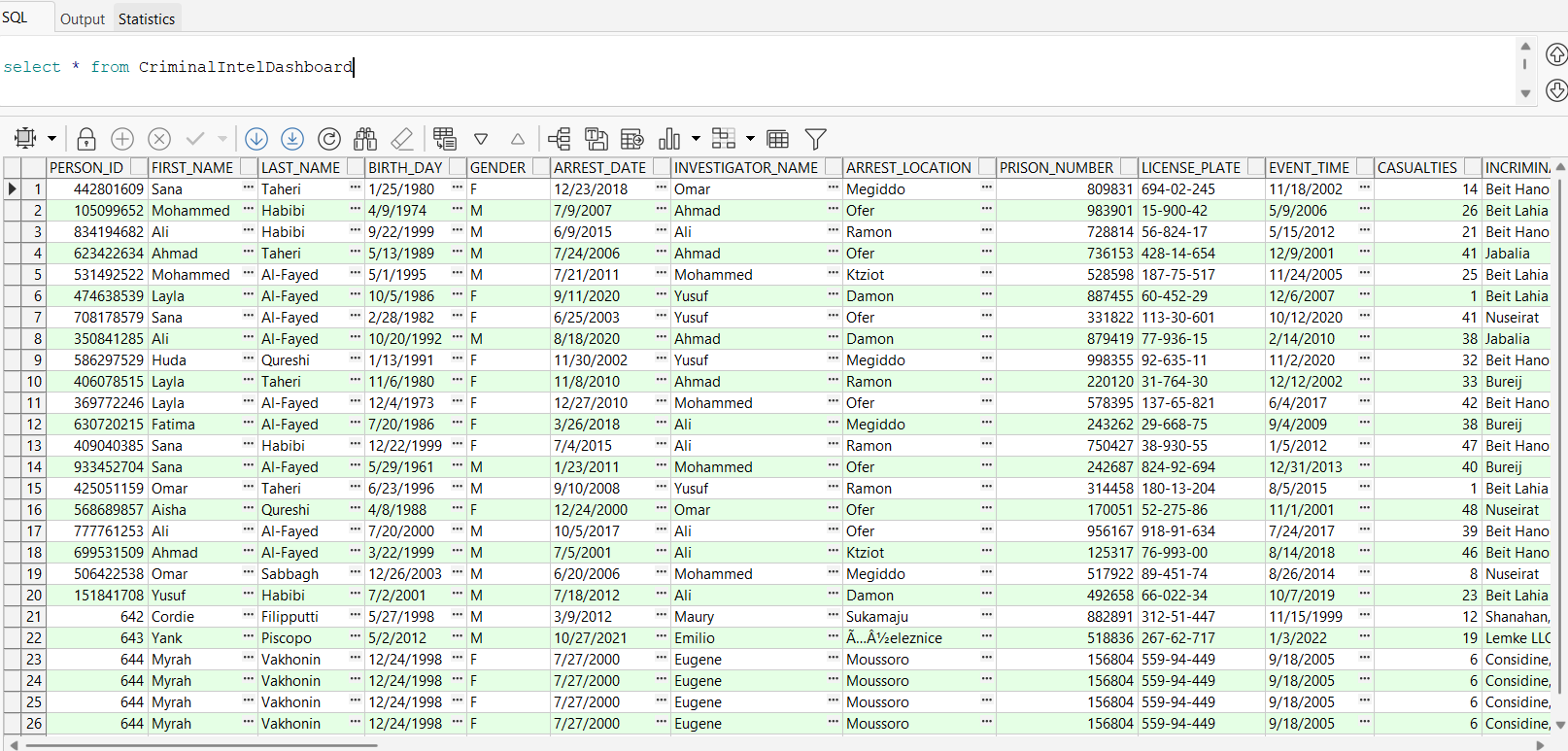
LEFT JOIN

HOUSE hs ON h.id = hs.id

LEFT JOIN

VEHICLE v ON ho.license\_plate = v.license\_plate;

#### דוגמא לפלט של טבלת הview שיצרנו:



### 4.6.5 – שאילתה 1: מציאת כל האנשים עם רישומי מעצר ושיחות טלפון

#### הסבר מילולי:

שאילתה זו מוצאת את כל האנשים שיש להם רישומי מעצר ושיחות טלפון. השאילתה מחזירה את מספר הזהות של האדם, שם פרטי, שם משפחה, תאריך מעצר, שם החוקר, מיקום המעצר ועל השיחות שעשה, זמן תחילת השיחה, צד א' וצד ב'. שאילתה זו מסייעת לחוקרים לקבל תמונה רחבה יותר על פעילויות החשודים בזמן ולאחר מעצר.

#### השאילתה:

SELECT

person\_id,

first\_name,

last\_name,

arrest\_date,

investigator\_name,

arrest\_location,

call\_start\_time,

a\_side,

b\_side

FROM

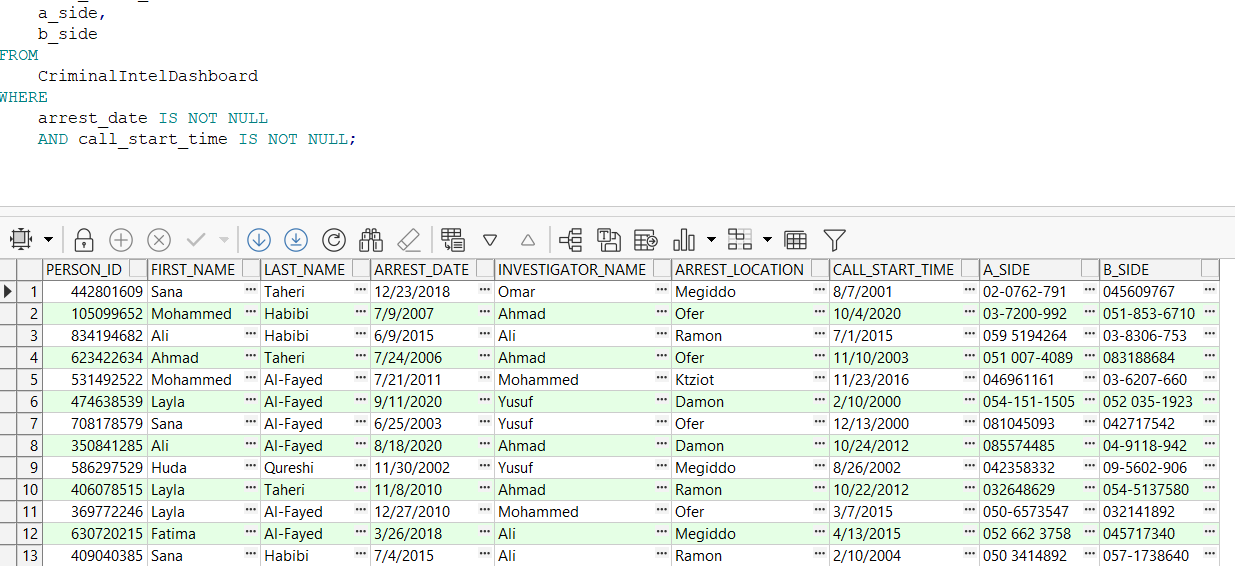
CriminalIntelDashboard

WHERE

arrest\_date IS NOT NULL

AND call\_start\_time IS NOT NULL;

#### דוגמא לפלט של השאילתה שיצרנו:



### 4.6.6 – שאילתה 2: סיכום משאבים וחיילים לפי יחידות עם סטטוס פרויקט

#### הסבר מילולי:

שאילתה זו מוצאת את כל האנשים שיש להם רישום על החזקת כלי נשק ושהיו מעורבים באירועים פליליים שהתרחשו במקומות ספציפיים. השאילתה מחזירה את מספר הזהות של המפגע, שם פרטי, שם משפחה, דגם הנשק, מספר סידורי, חברת הנשק, זמן האירוע ומיקום ההפללה. זה מסייע לחוקרים לזהות קשרים בין נשק לאירועים פליליים במיקומים ספציפיים.

#### השאילתה:

SELECT

person\_id,

first\_name,

last\_name,

weapon\_model,

weapon\_serial,

weapon\_company,

event\_time,

incrimination\_location

FROM

CriminalIntelDashboard

WHERE

weapon\_serial IS NOT NULL

AND incrimination\_location IN ('Beit Lahia', 'Bureij');

#### דוגמא לפלט של השאילתה שיצרנו:

