תוכן

[שלב 4 3](#_Toc171448637)

[4.1 יצירת הDSD- של המערכת הנוספת 3](#_Toc171448638)

# 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.

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 workin 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)

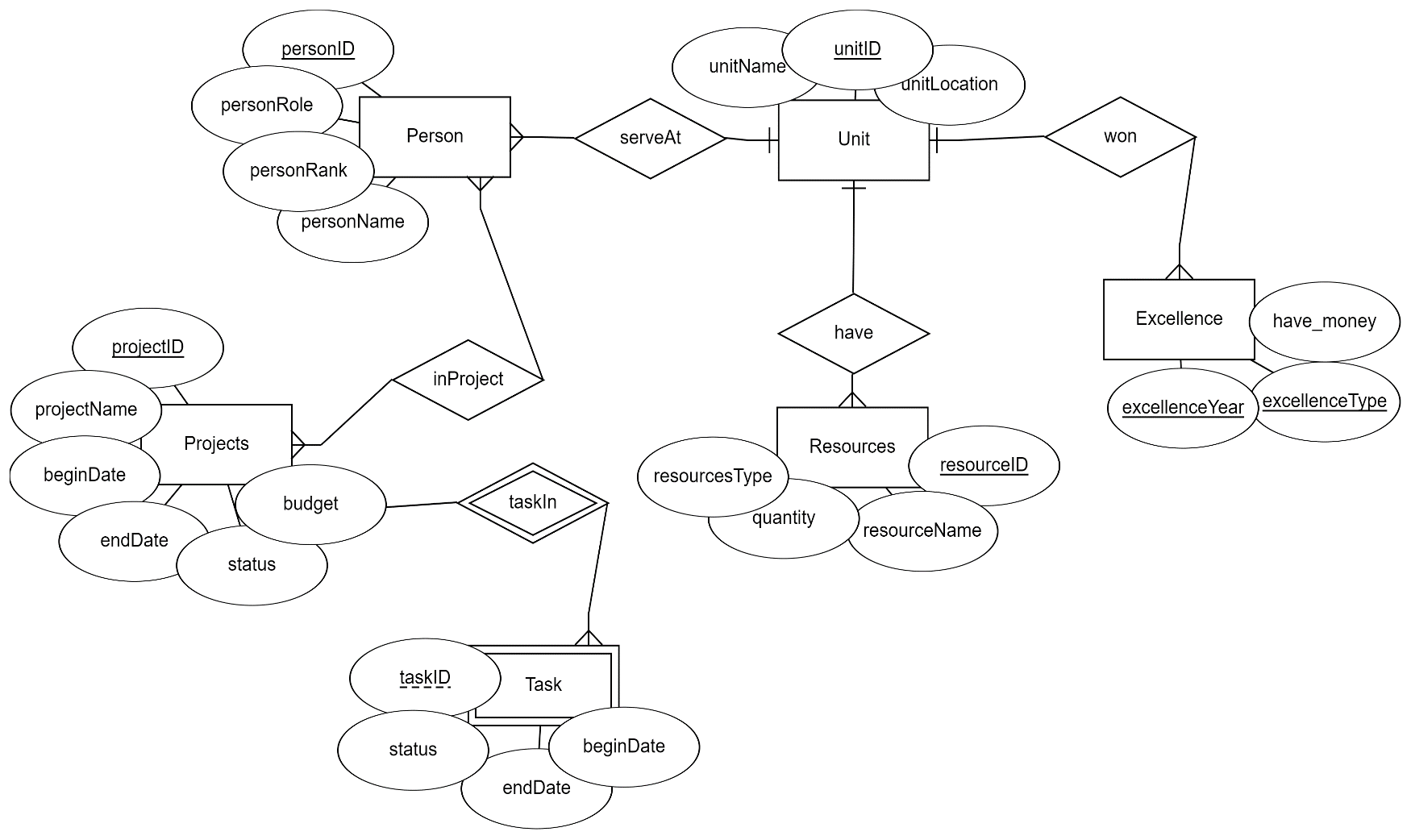
Projects - Task: אחד-לרבים (פרויקט יכול להכיל משימות רבות),

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

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

Projects - workin - 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 workin 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.

## 4.3 יצירת הERD- המשולב של שתי המערכות

יצרנו טבלה Unitops שמאחדת 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 הרצנו את הפונקציה משלב קודם אשר מחשבת את 100 היעדים המסוכנים ביותר ושומרת אותם בטבלה 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.

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*