מיני פרויקט בבסיסי נתונים

אגף חיל השריון של צה"ל



מגישים: יניב טל - 216196550

אהוביה בצלאל - 329228431

מרצה : מר יעקב ברזילי

קישור לgithub:

<https://github.com/yaniv-tal/DBProject_216196550_329228431>

תוכן עניינים

[שלב 1 - עיצוב ויצירת DB, שימוש בERDPLUS והכנסת נתונים 3](#_Toc197535375)

[תיאור המערכת 3](#_Toc197535376)

[תרשים ERD של הישויות והקשרים 4](#_Toc197535377)

[תרשים DSD 5](#_Toc197535378)

[תיאור הישויות והתכונות שלהן 5](#_Toc197535379)

[תיאור הקשרים 6](#_Toc197535380)

[רשימת הסכמות של בסיס הנתונים 6](#_Toc197535381)

[הסכמות של הישויות 6](#_Toc197535382)

[הסכמות של הקשרים 6](#_Toc197535383)

[הוכחה שהסכמות מנורמלות בNF3 7](#_Toc197535384)

[קוד SQL של יצירת הטבלאות (CREATE) 7](#_Toc197535385)

[קוד SQL של זריקת הטבלאות (DROP) 9](#_Toc197535386)

[הכנסת מידע לטבלאות 9](#_Toc197535387)

[הכנסת מידע לSOLIDERS – דרך 1, Mockaroo file 9](#_Toc197535388)

[הכנסת מידע לCOMMANDER – דרך 2, קובץ Excel 11](#_Toc197535389)

[הכנסת מידע לCREWMATE – דרך 3, סקריפט בפייתון 13](#_Toc197535390)

[הכנסת מידע לUNIT – דרך 4, קובץ טקסט 14](#_Toc197535391)

[הכנסת מידע לTANK – דרך 3, סקריפט בפייתון 16](#_Toc197535392)

[הכנסת מידע לMISSION – דרך 1, Mockaroo file 17](#_Toc197535393)

[הכנסת מידע לparticipate – דרך 2, קובץ Excel 20](#_Toc197535394)

[גיבוי הטבלאות 22](#_Toc197535395)

[שחזור הטבלאות 22](#_Toc197535396)

[שלב 2 - שאילתות (וכן השאילתות עם פרמטרים) 25](#_Toc197535397)

[שאילתות 25](#_Toc197535398)

[8 שאילתות select: 25](#_Toc197535399)

[3 שאילתות delete: 33](#_Toc197535400)

[3 שאילתות update: 37](#_Toc197535401)

[שאילתת Update עם Commit: 40](#_Toc197535402)

[אילוצים 42](#_Toc197535403)

# שלב 1 - עיצוב ויצירת DB, שימוש בERDPLUS והכנסת נתונים

## תיאור המערכת

זרוע השריון הישראלית, המוכרת גם כחיל השריון (חש"ן), עומדת כחוד החנית של צבא הגנה לישראל (צה"ל). זרוע זו, מהווה כוח התקפי עיקרי בשדה הקרב, ומשלבת טכנולוגיה מתקדמת, אומץ לב ורוח לחימה עזה.

זרוע השריון היא זרוע עוצמתית וחיונית בצה"ל, הממלאת תפקיד מרכזי בהגנת המדינה. כוח השריון מאפשר לצה"ל לבצע מגוון רחב של משימות בשדה הקרב, ולהוות גורם הרתעה משמעותי מול אויביה.

המערכת שנבנה תספק בסיס נתונים לחיל השריון. היא תתחלק לשלושה חלקים:

1. ניהול מידע על החיילים – המערכת תשמור עבור כל חייל את פרטיו האישיים ואת המפקד שלו. עבור מפקד המערכת תשמור גם את תחום פיקודו.
2. ניהול יחידות – המערכת תספק מידע לכל יחידה. שם היחידה, המפקד שלה והטנקים העומדים לרשותה.
3. ניהול משימות – עבור כל משימה המערכת תשמור את זמני המשימה והיחידות המשתתפות

## תמונה שמכילה צילום מסך, פתקייה, עיגול, ריבוע התיאור נוצר באופן אוטומטי תרשים ERD של הישויות והקשרים

## תמונה שמכילה טקסט, תרשים, שרטוט טכני, תוכנית התיאור נוצר באופן אוטומטיתרשים DSD

## תיאור הישויות והתכונות שלהן

|  |  |
| --- | --- |
| SOLIDERS (חיילים) | קבוצה של כל החיילים |
| sID | תעודת זהות |
| firstName | שם פרטי |
| lastName | שם משפחה |
| draftDate | תאריך גיוס |
| releaseDate | תאריך שחרור |

|  |  |
| --- | --- |
| COMMANDER (מפקד) | מפקד על טנק או על יחידה. יורש מחיילים |
| cID | מפתח, תעודת זהות |

|  |  |
| --- | --- |
| CREWMATE (איש צוות) | איש צוות בתוך טנק. יורש מחיילים |
| crID | מפתח, תעודת זהות |
| crID | מפתח זר, תעודת הזהות של מפקד הצוות |
| type | סוג החייל (נהג, טען או תותחן) |

|  |  |
| --- | --- |
| UNIT (יחידה) | יחידה בחיל השריון |
| unID | מפתח, מספר יחידה מזהה |
| cID | מפתח זר, תעודת הזהות של מפקד היחידה |
| uName | שם היחידה |

|  |  |
| --- | --- |
| TANK (טנק) | טנק בחיל השריון |
| tID | מפתח, מספר טנק מזהה |
| cID | מפתח זר, תעודת הזהות של מפקד הטנק |
| unID | מפתח זר, מספר היחידה המזהה של הטנק |

|  |  |
| --- | --- |
| MISSION (משימה) | משימה בחיל השריון |
| mID | מפתח, מספר משימה מזהה |
| mDate | התאריך בו יוצאת המשימה לפועל |

## תיאור הקשרים

|  |  |
| --- | --- |
| participates (משתתף) | רבים לרבים - יחידה שמשתתפת במשימה |
| mID | מפתח, מספר המשימה מזהה |
| unID | מפתח, מספר היחידה מזהה |

1. serves\_under (משרת) – יחיד לרבים, איש צוות שמשרת תחת מפקד.
2. commands (מפקד) – יחיד לרבים, מפקד שמפקד על טנק.
3. belongs (משתייך) – יחיד לרבים, טנק ששייך ליחידה.
4. orders (נותן הוראות) – יחיד לרבים, מפקד שנותן הוראות ליחידה.

## רשימת הסכמות של בסיס הנתונים

### הסכמות של הישויות

SOLIDERS(sID,firstName,lastName,draftDate,releaseDate)  
COMMANDER(cID)  
CREWMATE(crID,cID,type)  
TANK(tID,unID,cID)  
UNIT(unID,cID,uName)  
MISSION(mID,mdate)

### הסכמות של הקשרים

participates(mID,unID)

## הוכחה שהסכמות מנורמלות בNF3

SOLDIERS(חיילים) – המפתח הראשי הוא sID. וכל שאר התכונות תלויות ישירות במפתח הראשי sID. ולכן הטבלה בNF3.

COMMANDER (מפקד) – cID הוא מפתח ראשי וגם מפתח זר המפנה ל-sID בטבלה SOLDIERS. אין מאפיינים נוספים ולכן הטבלה בNF3.

CREWMATE (איש צוות) – type תלוי ב-cID וגם crID גם קשור ישירות ל-crID כמפתח זר. הטבלה ב3NF.

UNIT (יחידה) – uName ו-cID תלויים במפתח הראשי .unID הטבלה נמצאת בNF3.

TANK (טנק) – unID ו-cID הם תכונות עם יחסים של מפתחות זרים. אבל, אין תלות טרנזיטיבית כיוון ש-tID הוא המפתח הראשי. ולכן הטבלה בNF3.

MISSION (משימה) – מכיוון שmdate תלוי ישירות במפתח הראשי mID, הטבלה בNF3.

participates(משתתף) – גם mID וגם unID הם מפתחות זרים, והם יוצרים יחד מפתח מורכב. כל מאפיין במפתח המורכב קובע לחלוטין את המאפיינים האחרים.

למסקנה כל הטבלאות עונות על התנאים ל-3NF.

## קוד SQL של יצירת הטבלאות (CREATE)

CREATE TABLE SOLDIERS

(

  sID numeric(9) NOT NULL,

  firstName VARCHAR(20) NOT NULL,

  lastName VARCHAR(20) NOT NULL,

  draftDate DATE NOT NULL,

  releaseDate DATE NOT NULL,

  PRIMARY KEY (sID)

);

CREATE TABLE MISSION

(

  mdate DATE NOT NULL,

  mID numeric(9) NOT NULL,

  PRIMARY KEY (mID)

);

CREATE TABLE COMMANDER

(

  cID numeric(9) NOT NULL,

  PRIMARY KEY (cID),

  FOREIGN KEY (cID) REFERENCES SOLDIERS(sID)

);

CREATE TABLE CREWMATE

(

  type VARCHAR(20) NOT NULL,

  crID numeric(9) NOT NULL,

  cID numeric(9) NOT NULL,

  PRIMARY KEY (crID),

  FOREIGN KEY (crID) REFERENCES SOLDIERS(sID),

  FOREIGN KEY (cID) REFERENCES COMMANDER(cID)

);

CREATE TABLE UNIT

(

  unID numeric(9) NOT NULL,

  uName VARCHAR(20) NOT NULL,

  cID numeric(9) NOT NULL,

  PRIMARY KEY (unID),

  FOREIGN KEY (cID) REFERENCES COMMANDER(cID)

);

CREATE TABLE TANK

(

  tID numeric(9) NOT NULL,

  unID numeric(9) NOT NULL,

  cID numeric(9) NOT NULL,

  PRIMARY KEY (tID),

  FOREIGN KEY (unID) REFERENCES UNIT(unID),

  FOREIGN KEY (cID) REFERENCES COMMANDER(cID)

);

CREATE TABLE participates

(

  mID numeric(9) NOT NULL,

  unID numeric(9) NOT NULL,

  PRIMARY KEY (mID, unID),

  FOREIGN KEY (mID) REFERENCES MISSION(mID),

  FOREIGN KEY (unID) REFERENCES UNIT(unID)

);

## קוד SQL של זריקת הטבלאות (DROP)

DROP TABLE CREWMATE;

DROP TABLE TANK;

DROP TABLE participates;

DROP TABLE MISSION;

DROP TABLE UNIT;

DROP TABLE COMMANDER;

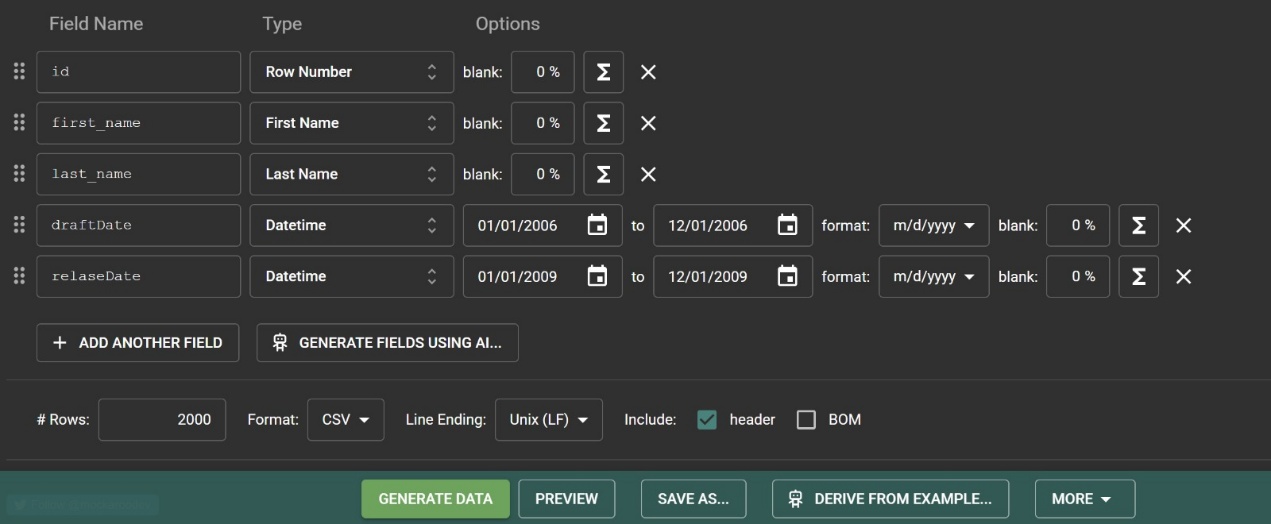
DROP TABLE SOLDIERS;

## הכנסת מידע לטבלאות

נכניס את כל הנתונים שלנו לתוך מסד הנתונים. לשם כך נוכל להשתמש בשיטות שונות.

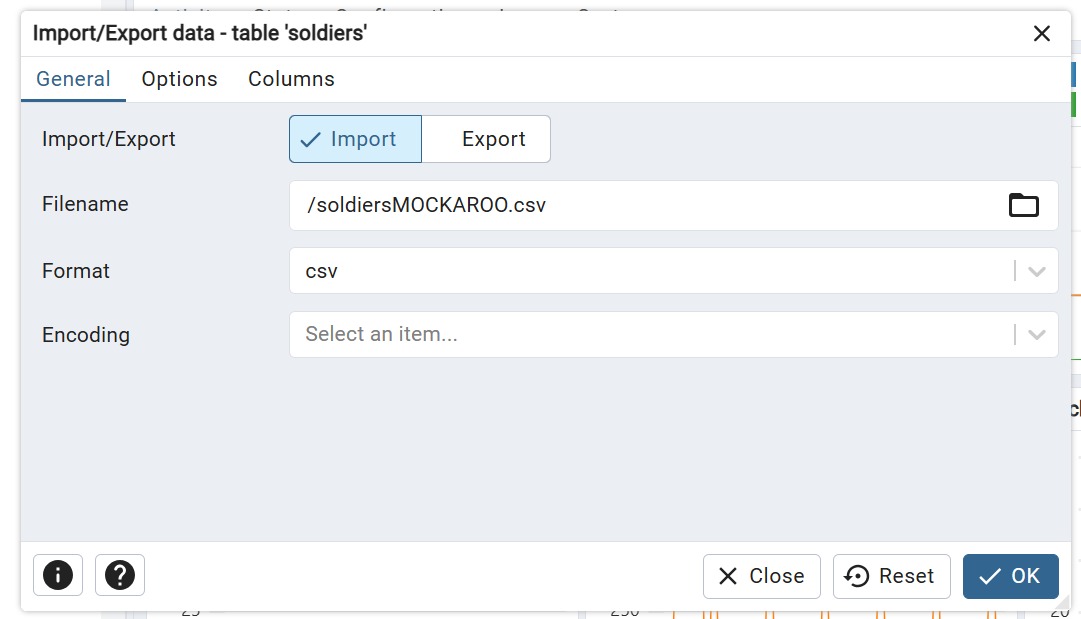
### הכנסת מידע לSOLIDERS – דרך 1, Mockaroo file

ניצור ונכניס את הData בעזרת מחולל המידע של Mockaroo file



ככה נראה המידע המפוברק שקיבלנו ממחולל הנתונים. תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

נייבא את קובץ ה-csv בעזרת מערכת הקבצים של Postgres

נבדוק שהמידע אכן נקלט במערכת בעזרת הפקודה Select \* from SOLDIERS



### הכנסת מידע לCOMMANDER – דרך 2, קובץ Excel

נשתמש במידע מתוך קובץ Excel

תמונה שמכילה טקסט, צילום מסך, מספר

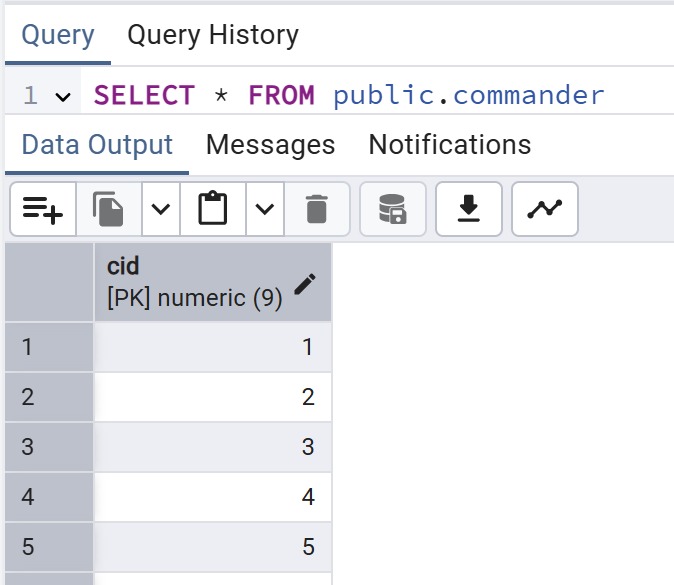
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

נייבא את קובץ ה-csv בעזרת מערכת הקבצים של Postgres

תמונה שמכילה טקסט, צילום מסך, תוכנה, תצוגה

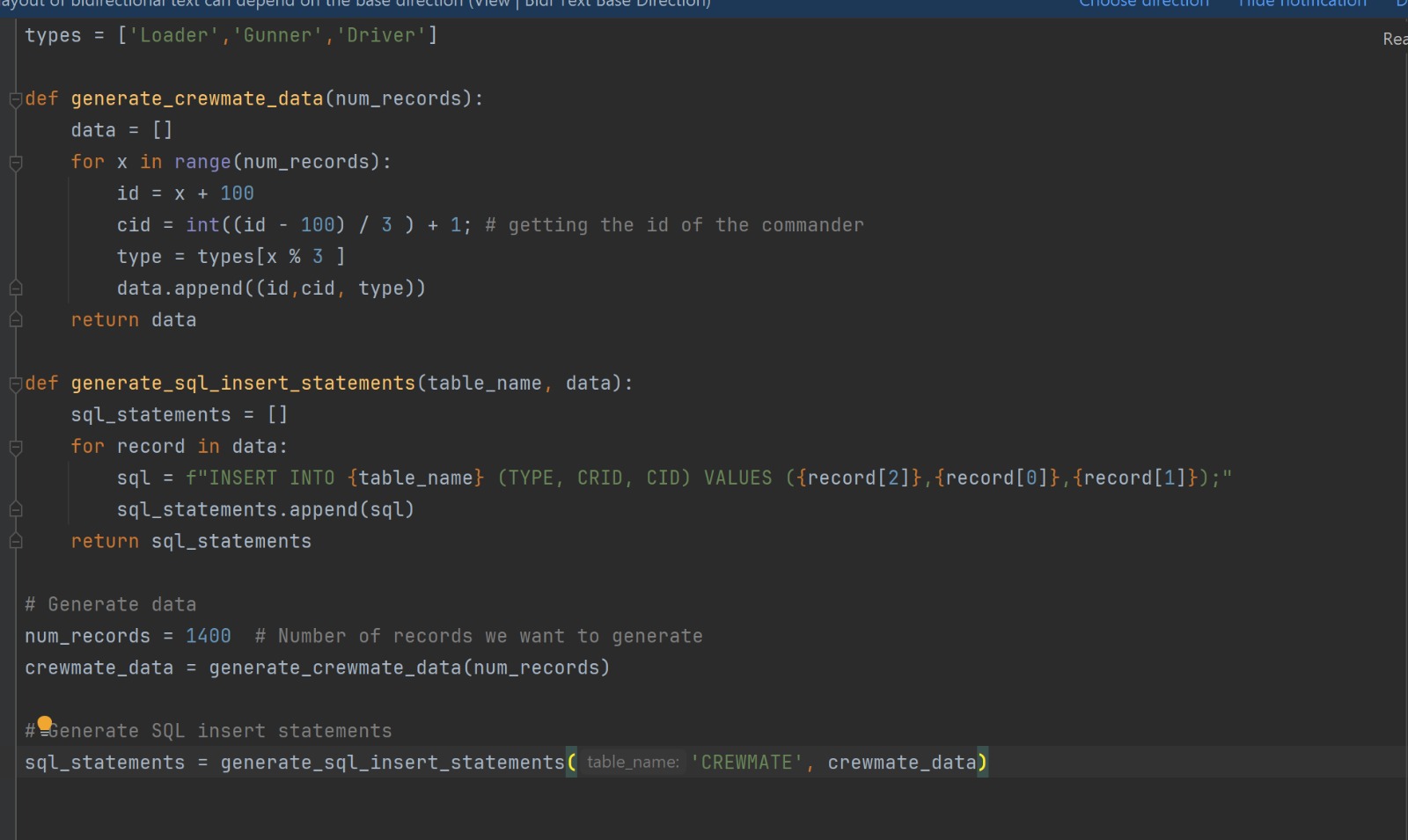
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

נבדוק האם המידע נקלט בעזרת הפקודה Select \* from COMMANDER



### הכנסת מידע לCREWMATE – דרך 3, סקריפט בפייתון

הפעם ניקח דרך שונה. ניצור סקריפט בפייתון שיצור רשימה של פקודות INSERT בהן נשתמש בשביל להכניס מידע לאנשי הצוות.



נכניס את שורות הInsert אל לתוך מסך הנתונים



ונוודא שמידע נקלט בעזרת פקודת SELECT

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

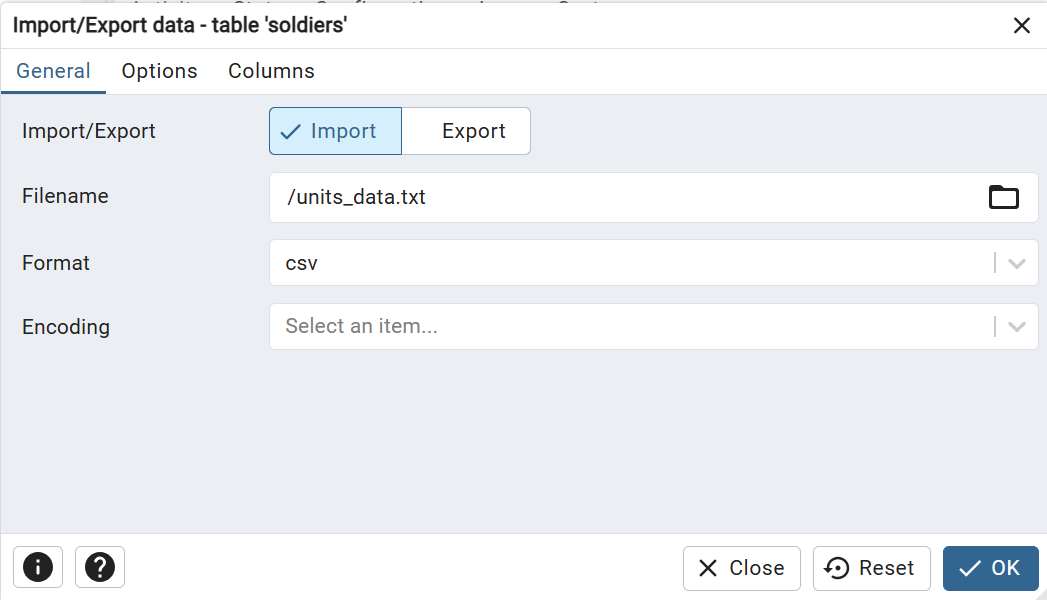
### הכנסת מידע לUNIT – דרך 4, קובץ טקסט

הפעם נכניס למסד הנתונים בטבלה UNIT מידע מקובץ טקסט, ניעזר בבינה מלאכותית ובסיס הנתונים של צה"ל על מנת להכניס מידע ריאליסטי ככל הניתן.

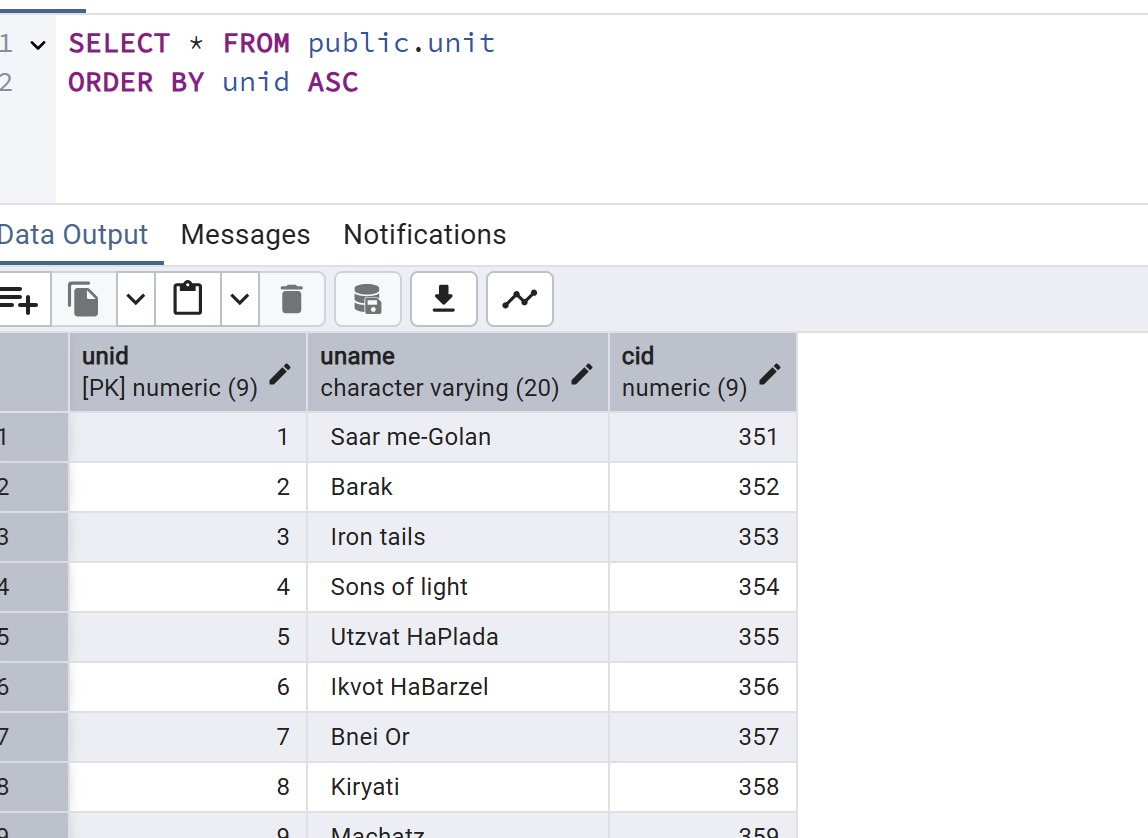
תמונה שמכילה טקסט, צילום מסך, מסמך, גופן

התיאור נוצר באופן אוטומטי

נייבא את קובץ ה-csv בעזרת מערכת הקבצים של Postgres



נוודא שאכן המידע נשמר במערכת



### הכנסת מידע לTANK – דרך 3, סקריפט בפייתון

תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

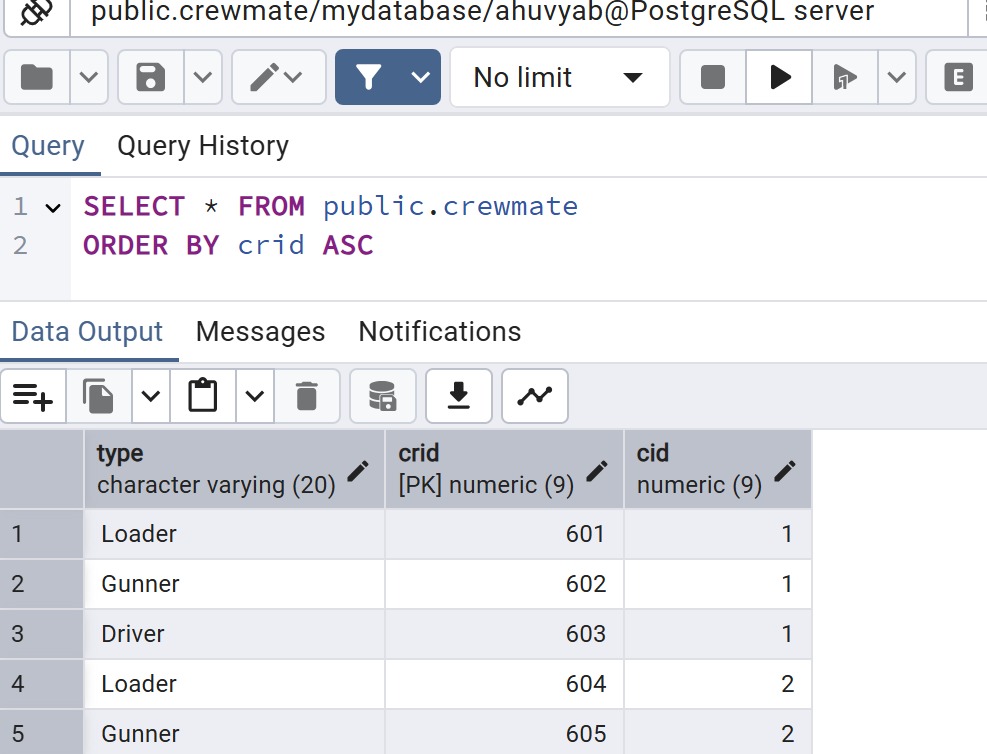
הסקריפט יצור לנו פקודות INSERT בSQL

נריץ את הפקודות דרך הquery tool ב Postgres

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

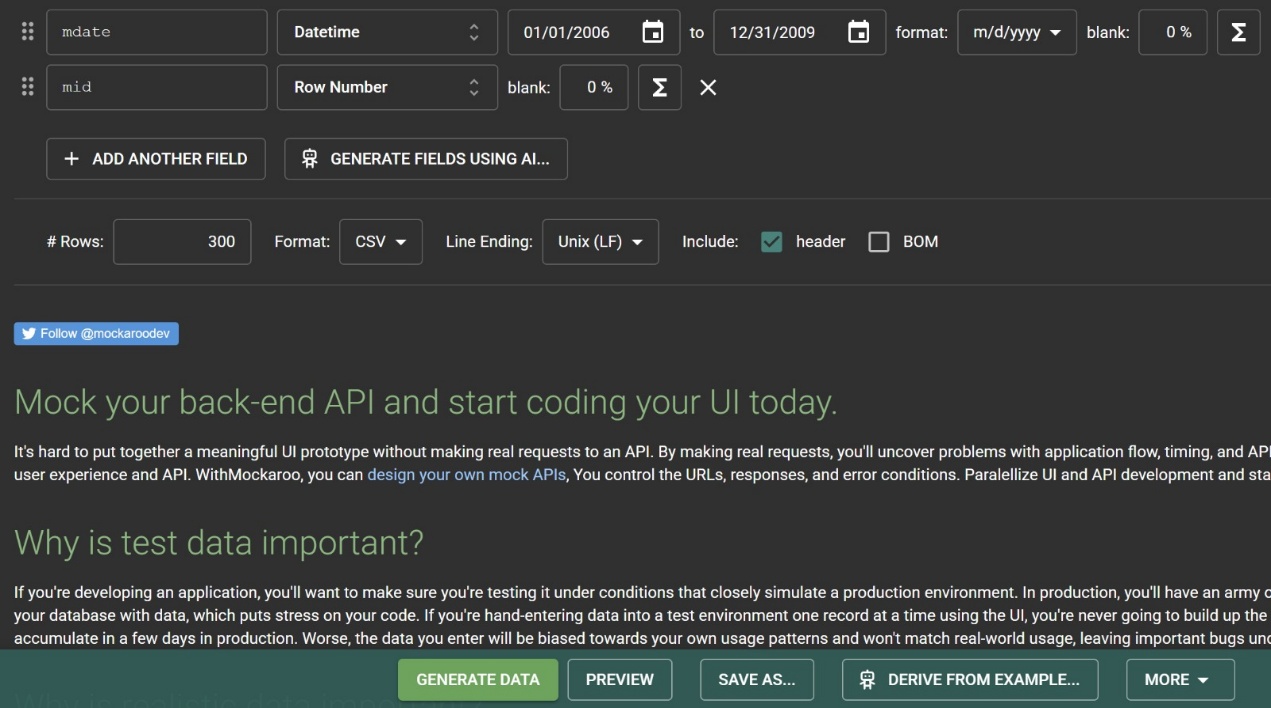
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

נוודא שהמידע נקלט במערכת בעזרת שאילתת select \* from TANK;



### הכנסת מידע לMISSION – דרך 1, Mockaroo file

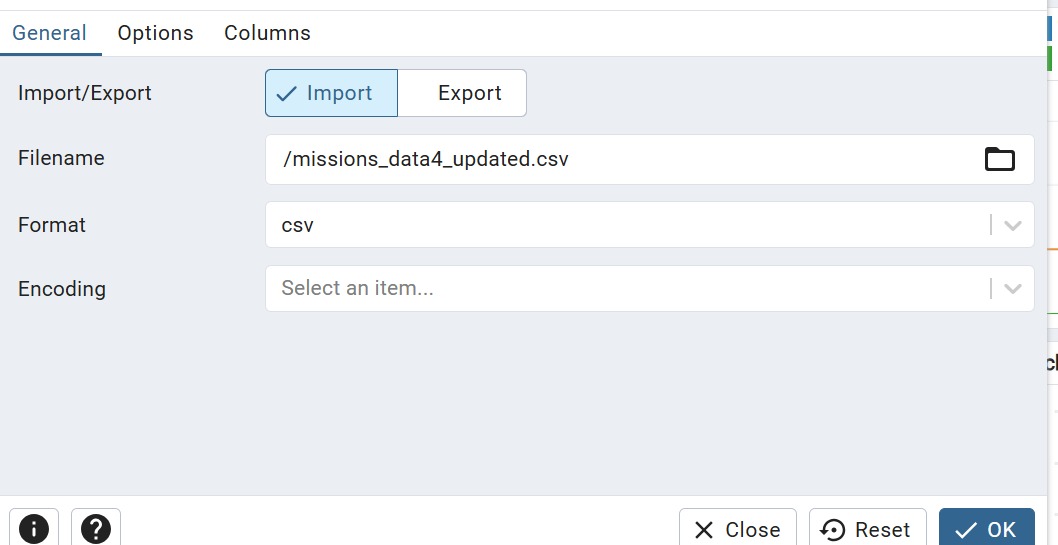
ניצור ונכניס את הData בעזרת מחולל המידע של Mockaroo file



ככה נראה המידע המפוברק שקיבלנו ממחולל הנתונים. תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

נייבא את קובץ ה-csv בעזרת מערכת הקבצים של Postgres



נבדוק שהמידע אכן נקלט במערכת בעזרת הפקודה Select \* from mission

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

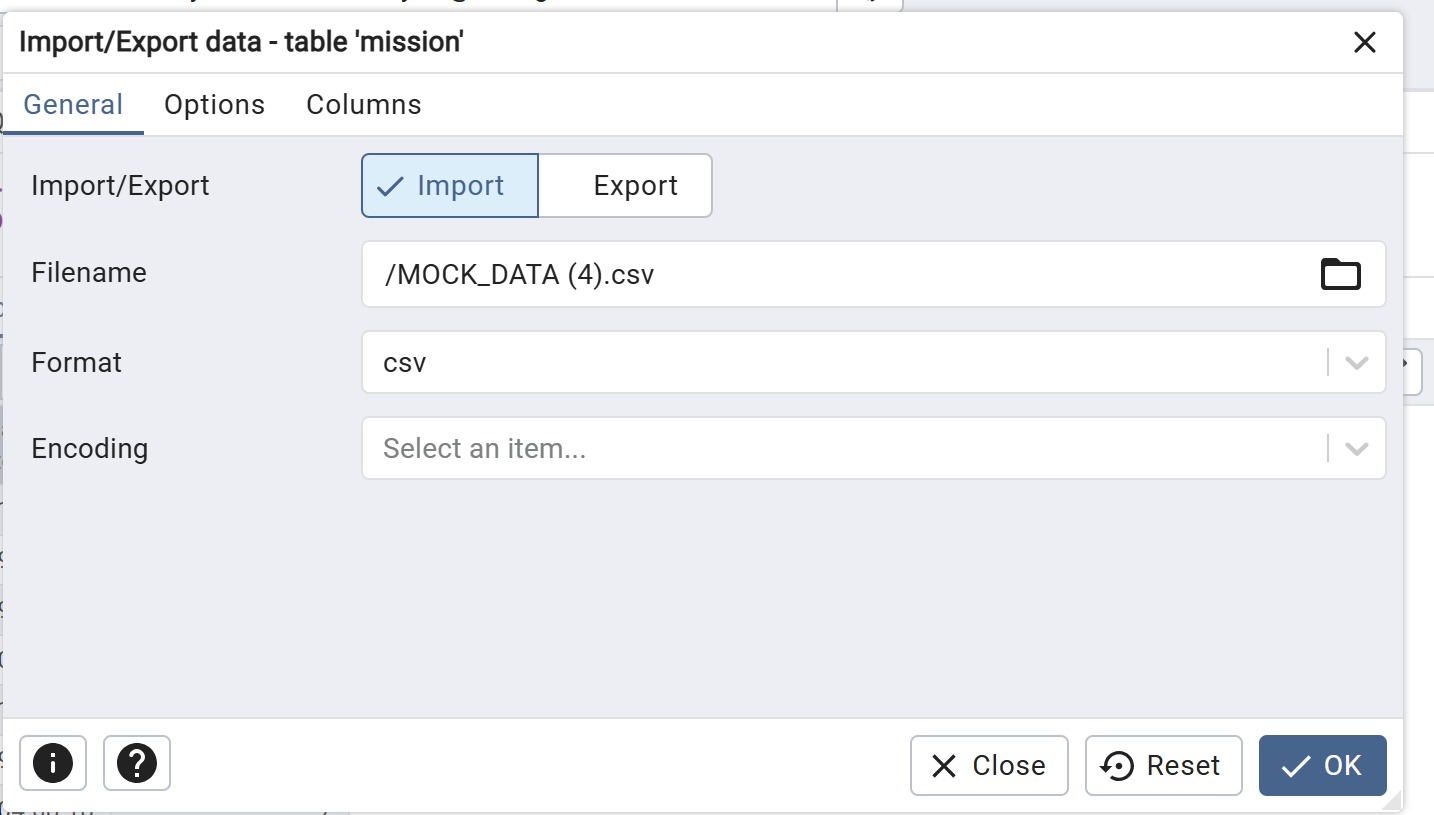
### הכנסת מידע לparticipate – דרך 2, קובץ Excel

נשתמש במידע מתוך קובץ Excel

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

נייבא את קובץ ה-csv בעזרת מערכת הקבצים של Postgres



נבדוק האם המידע נקלט בעזרת הפקודה Select \* from participates

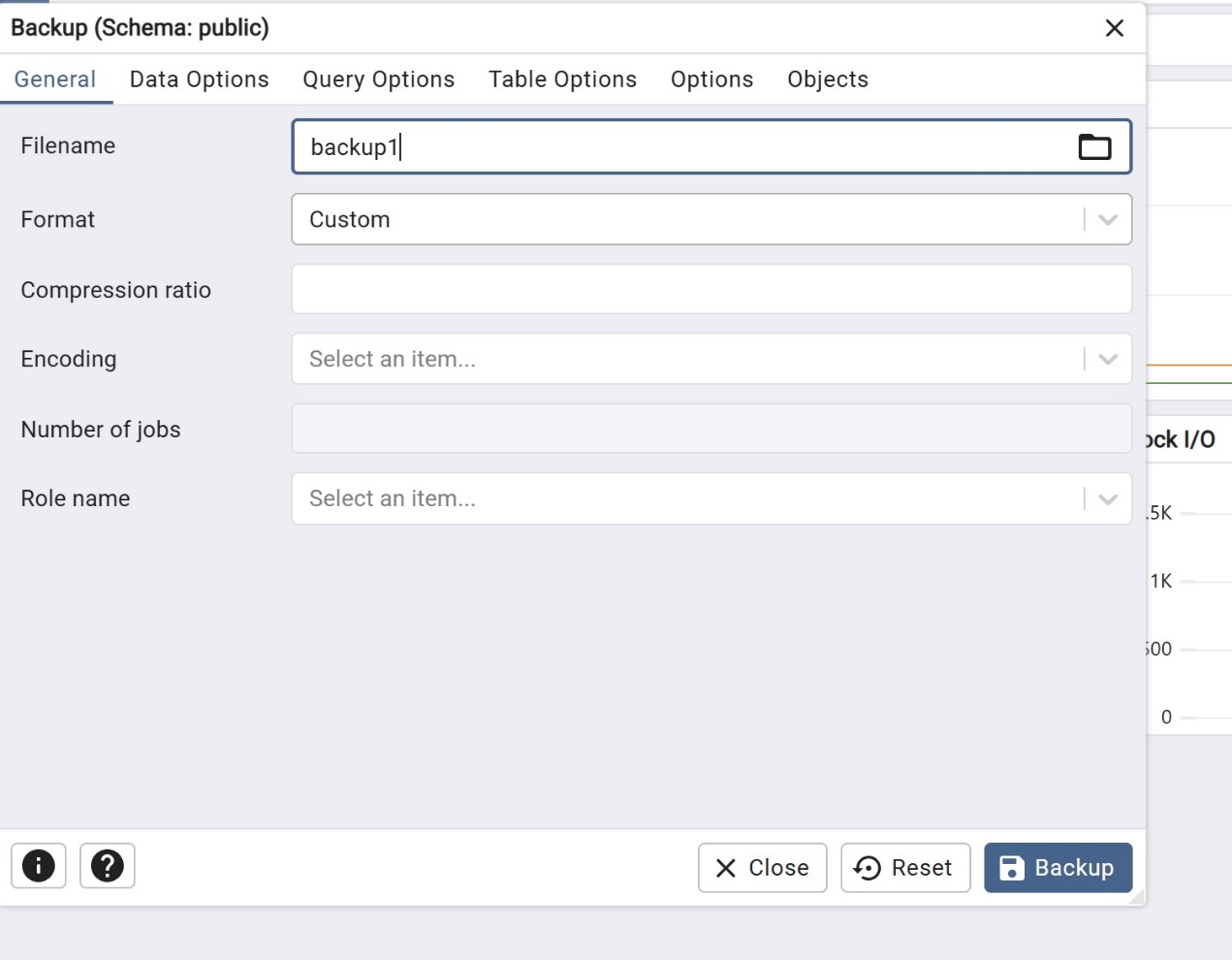
תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

## גיבוי הטבלאות

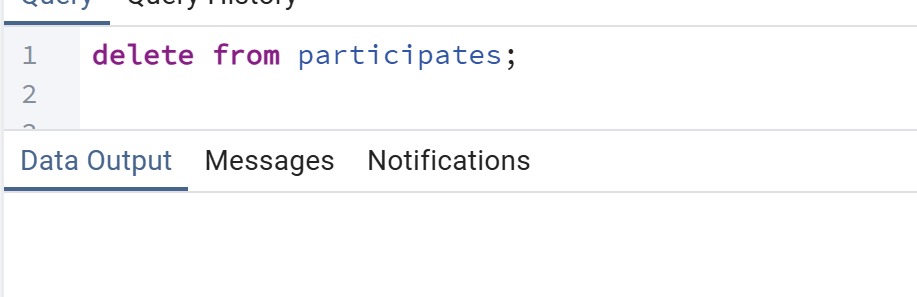
בשביל להבטיח שהמידע של בסיס הנתונים שלנו תמיד נשמר נצטרך לגבות את כל המידע שבתוך הטבלאות למקור שמירה חיצוני.

אחרי שסימנו את כל הטבלאות שאנחנו רוצים לשמור, נייצא את המידע שבהן לתוך קובץ מסוג יצוא וככה נוכל גם להעביר את הטבלאות למחשבים אחרים וגם לשמור עליהן



## שחזור הטבלאות

אחרי שגיבינו את כל הטבלאות, נרצה לראות שזה אכן עבד. החלטנו למחוק את המידע שבטבלה participates בעזרת הפקודה Delete.



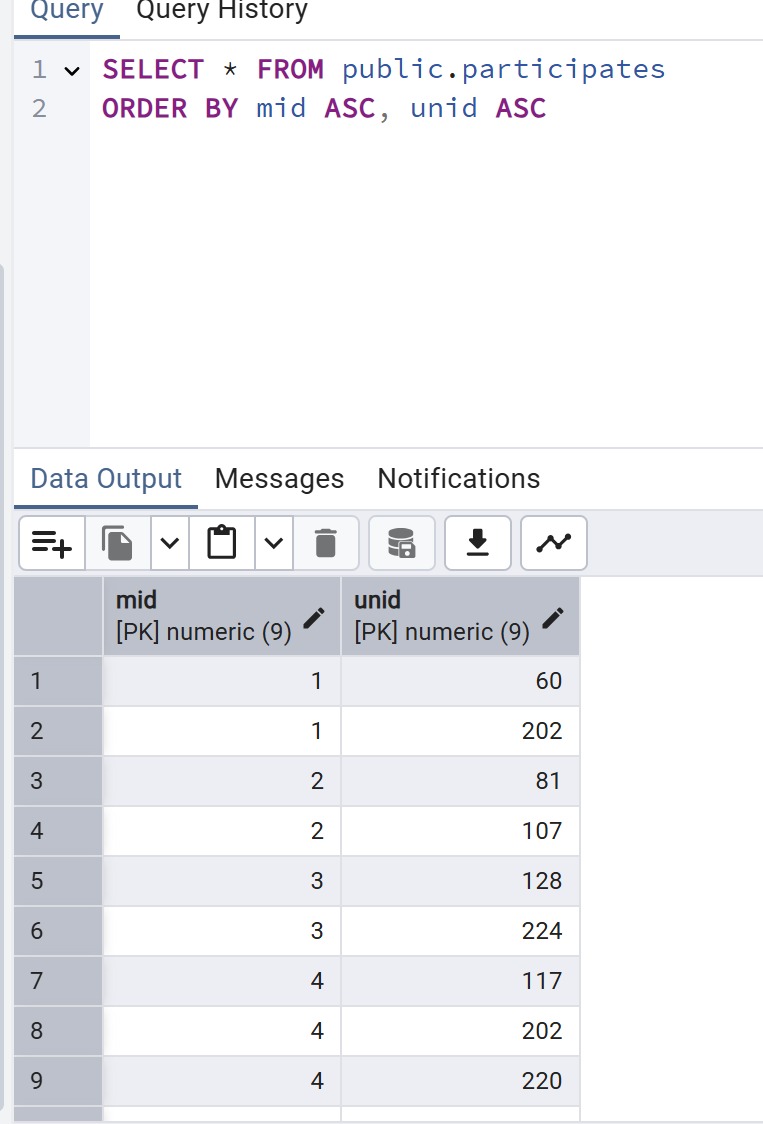
נטען עכשיו את המידע מקובץ היצוא החיצוני שלנו בשביל לבדוק האם המידע באמת נשמר

תמונה שמכילה טקסט, צילום מסך, תוכנה, תצוגה

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

המידע נשמר 🥳

אנחנו יכולים לגשת אליו גם אחרי שמחקנו אותו



# שלב 2 - שאילתות (וכן השאילתות עם פרמטרים)

## שאילתות

### 8 שאילתות select:

שאילתה 1 : מציאת זמן השירות הממוצע של חיילים ששירתו בתור מפקדים, והשוואה עם זמן השירות הממוצע של אילו שאינם מפקדים.

-- Subquery for Commanders

SELECT 'Commander' AS role,

    AVG(s.releaseDate - s.draftDate) AS avgServiceDuration

FROM SOLDIERS s

JOIN COMMANDER c ON s.sID = c.cID

UNION ALL

-- Subquery for Non-Commanders

SELECT 'Non-Commander' AS role,

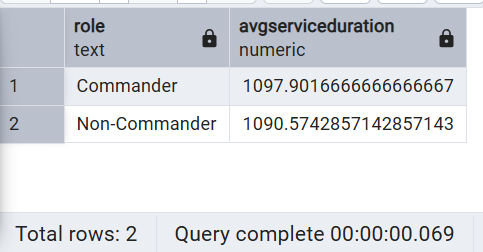
    AVG(s.releaseDate - s.draftDate) AS avgServiceDuration

FROM SOLDIERS s

LEFT JOIN COMMANDER c ON s.sID = c.cID

WHERE c.cID IS NULL;

התוצאה שתתקבל מהרצת השאילתה היא



שאילתה 2 : רשימה של כל היחידות שהשתתפו ביותר משימות ממספר המשימות ליחידה הממוצע

SELECT u.unID, u.uName, COUNT(p.mID) AS missionCount

FROM UNIT u

JOIN participates p ON u.unID = p.unID

GROUP BY u.unID, u.uName

HAVING COUNT(p.mID) > (

    SELECT AVG(missionCount)

    FROM (

        SELECT COUNT(p.mID) AS missionCount

        FROM participates p

        GROUP BY p.unID

    ) subquery

)

ORDER BY missionCount DESC;

התוצאה שתתקבל מהרצת השאילתה היא

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

שאילתה 3 : כל היחידות ממוימות לפי מספר הטנקים בכל יחידה

SELECT u.uName AS unitName, COUNT(t.tID) AS numberOfTanks

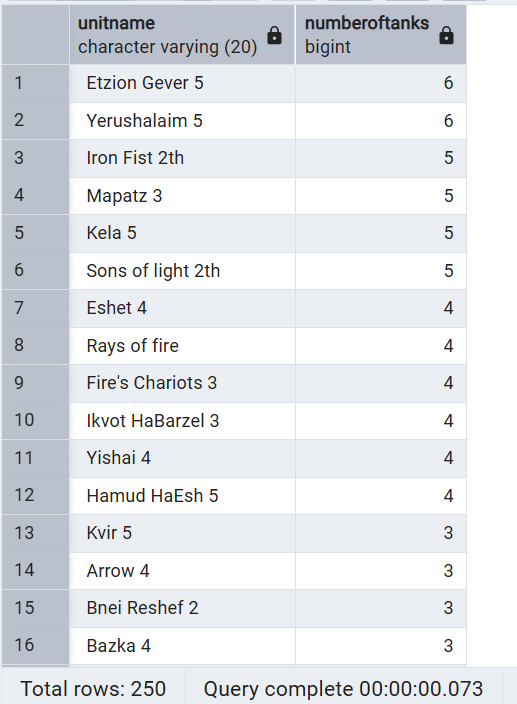
FROM UNIT u

LEFT JOIN TANK t ON u.unID = t.unID

GROUP BY u.uName

ORDER BY numberOfTanks DESC;

התוצאה שתתקבל מהרצת השאילתה היא



שאילתה 4 : אחרי מבצע מוצלח רצו בפיקוד להביא לחיילים צ'ופר, לכל חייל תיק עם הדפס של היחידה. בשביל כל החיילים שהתגייסו במרץ, הודפס התיק "גיוס מרץ שריון".  
יחידת האפסנאות צריכה רשימה של כל החיילים שהתגייסו בסביבות מרץ אפריל.

SELECT \* FROM CREWMATE

WHERE crID IN (

  SELECT sID

  FROM SOLDIERS

  WHERE EXTRACT(MONTH FROM draftDate) BETWEEN 3 AND 4

);

התוצאה שתתקבל מהרצת השאילתה היא

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

שאילתה 5 : מציאת 10 החיילים הוותיקים ביותר.

SELECT

  s.firstName || ' ' || s.lastName AS commanderName,

  s.draftDate,

  s.releaseDate,

  FLOOR((s.releaseDate - s.draftDate)/365) AS yearsServed,

  FLOOR(MOD((s.releaseDate - s.draftDate), 365)/30) AS monthsServed,

  MOD((s.releaseDate - s.draftDate), 30) AS daysServed

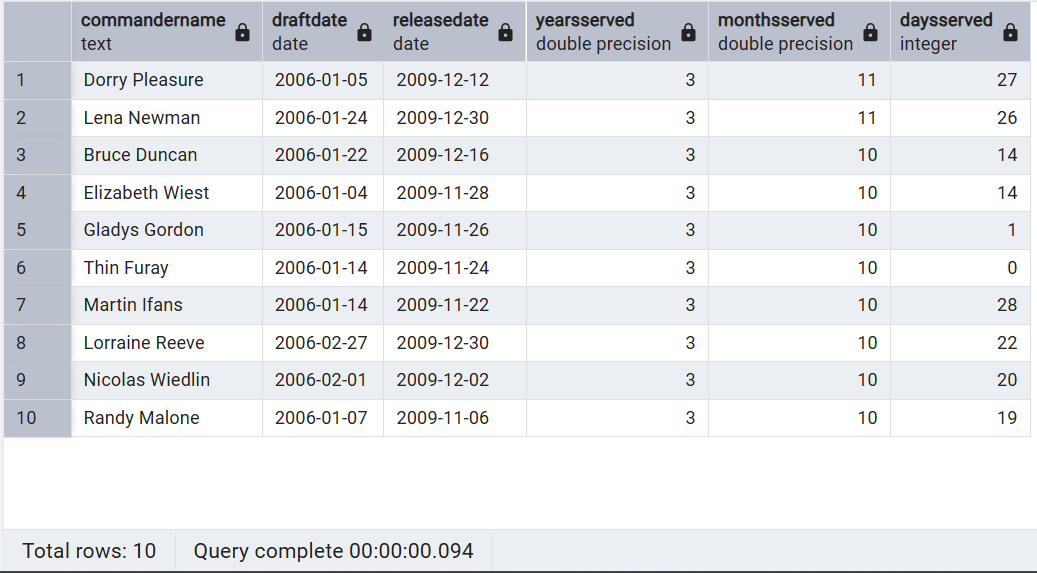
FROM SOLDIERS s

JOIN COMMANDER c ON s.sID = c.cID

ORDER BY (s.releaseDate - s.draftDate) DESC

FETCH FIRST 10 ROWS ONLY;

התוצאה שתתקבל מהרצת השאילתה היא



שאילתה 6 : רשימת מפקדים של יחידות שהשתתפו בלפחות 3 משימות, כולל שם המפקד, מזהה יחידה, מספר המשימות.

SELECT

  s.firstName,

  s.lastName,

  u.unID,

  COUNT(p.mID) AS missionCount

FROM COMMANDER c

JOIN SOLDIERS s ON c.cID = s.sID

JOIN UNIT u ON u.cID = c.cID

JOIN participates p ON u.unID = p.unID

GROUP BY s.firstName, s.lastName, u.unID

HAVING COUNT(p.mID) >= 3

ORDER BY missionCount DESC;

התוצאה שתתקבל מהרצת השאילתה היא

תמונה שמכילה טקסט, צילום מסך, מספר

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

שאילתה 7 : כמה חיילים משרתים תחת כל מפקד, כולל שם המפקד ומספר החיילים, ממויין לפי מספר החיילים.

SELECT

  s.firstName AS commanderFirstName,

  s.lastName AS commanderLastName,

  COUNT(cm.crID) AS soldiersUnderCommand

FROM COMMANDER c

JOIN SOLDIERS s ON c.cID = s.sID

JOIN CREWMATE cm ON c.cID = cm.cID

GROUP BY s.firstName, s.lastName

ORDER BY soldiersUnderCommand DESC;

התוצאה שתתקבל מהרצת השאילתה היא

תמונה שמכילה טקסט, צילום מסך, מספר

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

שאילתה 8 : עשרת המפקדים שיחידתם השתתפה בהכי הרבה משימות לצורך הענקת עיטור העוז.

SELECT

  s.firstName || ' ' || s.lastName AS commanderName,

  u.uName AS unitName,

  COUNT(p.mID) AS missionCount

FROM COMMANDER c

JOIN SOLDIERS s ON c.cID = s.sID

JOIN UNIT u ON c.cID = u.cID

JOIN participates p ON u.unID = p.unID

GROUP BY s.firstName, s.lastName, u.uName

ORDER BY missionCount DESC

FETCH FIRST 10 ROWS ONLY;

התוצאה שתתקבל מהרצת השאילתה היא

תמונה שמכילה טקסט, צילום מסך, תוכנה, תצוגה

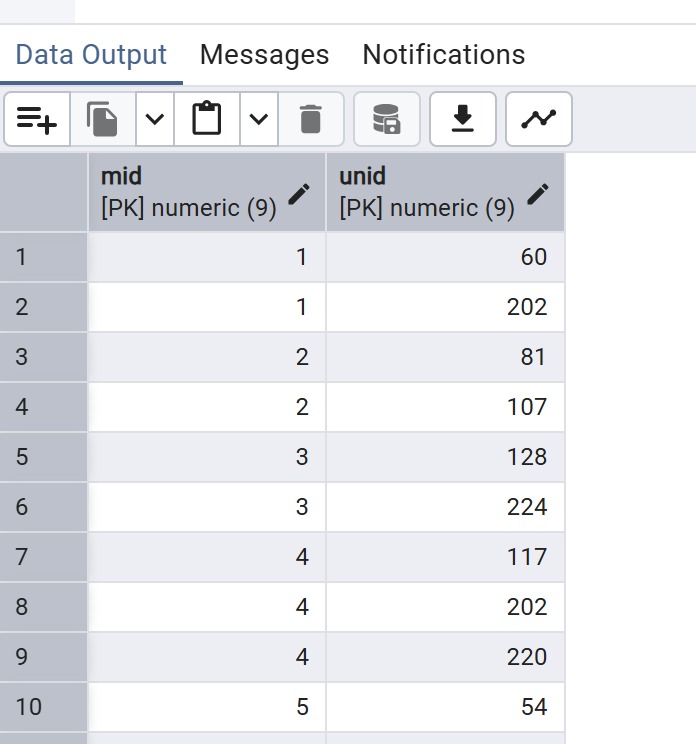
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

### 3 שאילתות delete:

שאילתה 1 : בעקבות תפקודה המופתי של יחידה 101 בעזה, הוחלט בדרג הפיקודי העליון על הפיכת היחידה ליחידה מיוחדת ומסווגת ועל כן, יש צורך במחיקת כל הרישומים המעידים על קיום היחידה.

DELETE FROM participates p

WHERE p.UNID = 101;



נמחקו נתוני היחידה 101.

לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה.

שאילתה 2 : יחידה 5 התפרקה עקב בעיות משמעת, נמחק את חברי הצוות בטנקים של היחידה.

DELETE FROM CREWMATE

WHERE crID IN (

  SELECT cr.crID

  FROM CREWMATE cr

  JOIN COMMANDER c ON cr.cID = c.cID

  JOIN UNIT u ON u.cID = c.cID

  WHERE u.unID = 5

);

rollback;



לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה.

שאילתה 3 ושימוש בפקודת rollback:

הוחלט לשחרר משירות מילואים את כל החיילים שאינם בשירות פעיל כבר יותר מ15 שנה ושאינם מפקדים או נהגים. נמחק אותם מבסיס הנתונים.

begin;

DELETE FROM CREWMATE

WHERE crID IN (

    SELECT sID

    FROM SOLDIERS

    WHERE sID NOT IN (SELECT cID FROM COMMANDER)

      AND sID NOT IN (

          SELECT crID FROM CREWMATE WHERE LOWER(type) = 'driver'

      )

      AND releaseDate < CURRENT\_DATE - INTERVAL '15 years'

);

DELETE FROM SOLDIERS

WHERE sID NOT IN (SELECT cID FROM COMMANDER)

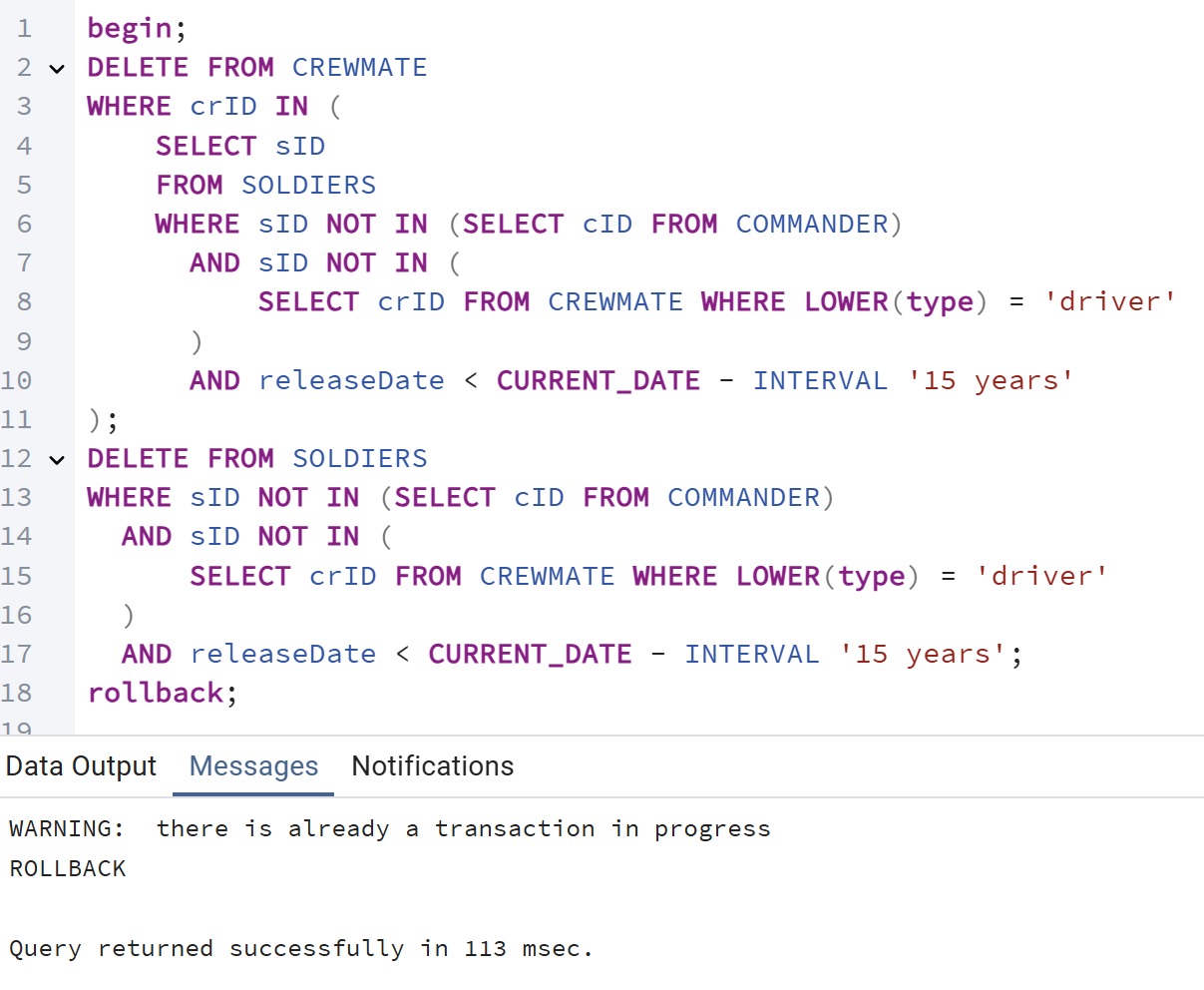
  AND sID NOT IN (

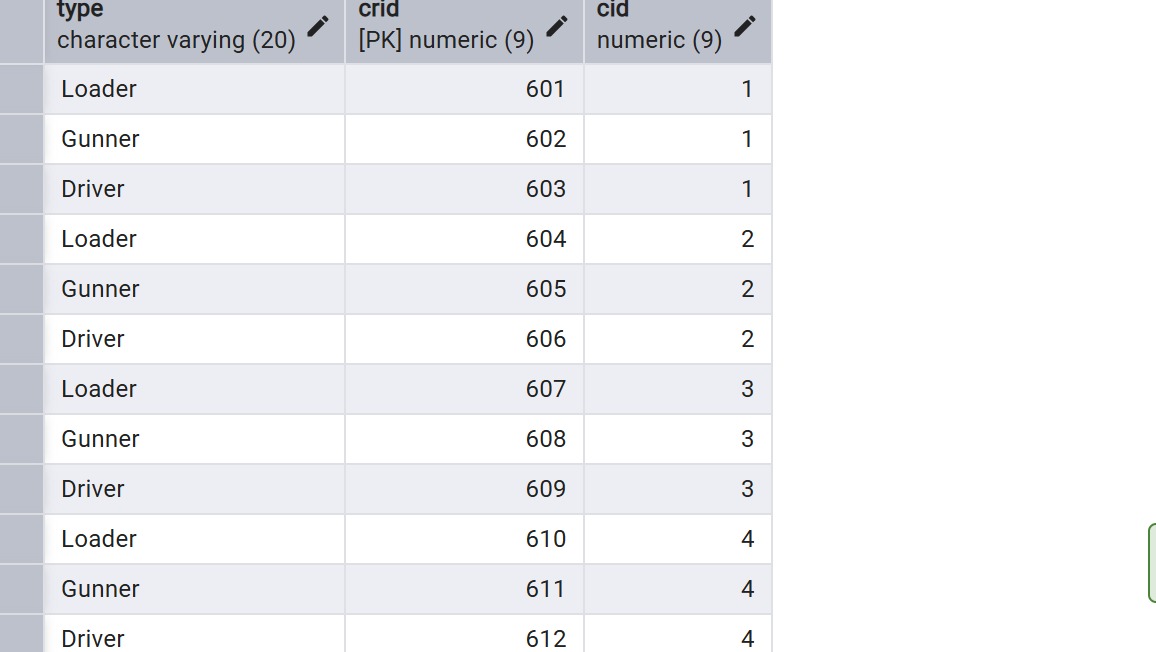
      SELECT crID FROM CREWMATE WHERE LOWER(type) = 'driver'

  )

  AND releaseDate < CURRENT\_DATE - INTERVAL '15 years';

rollback;





לאחר הרצת פקודת המחיקה עם שימוש בפקודת rollback בסיס הנתונים נשמר.

DELETE FROM CREWMATE

WHERE crID IN (

    SELECT sID

    FROM SOLDIERS

    WHERE sID NOT IN (SELECT cID FROM COMMANDER)

      AND sID NOT IN (

          SELECT crID FROM CREWMATE WHERE LOWER(type) = 'driver'

      )

      AND releaseDate < CURRENT\_DATE - INTERVAL '15 years'

);

DELETE FROM SOLDIERS

WHERE sID NOT IN (SELECT cID FROM COMMANDER)

  AND sID NOT IN (

      SELECT crID FROM CREWMATE WHERE LOWER(type) = 'driver'

  )

  AND releaseDate < CURRENT\_DATE - INTERVAL '15 years';



לאחר הרצת פקודת המחיקה ללא rollback נמחקו 935 חיילים מבסיס הנתונים.

### 3 שאילתות update:

שאילתה 1 : שינוי שם לinactive Unit ליחידות שאינן פעילות.

UPDATE UNIT

SET uName = 'Inactive Unit'

WHERE unID NOT IN (

    SELECT p.unID

    FROM participates p

    WHERE p.mID IN (

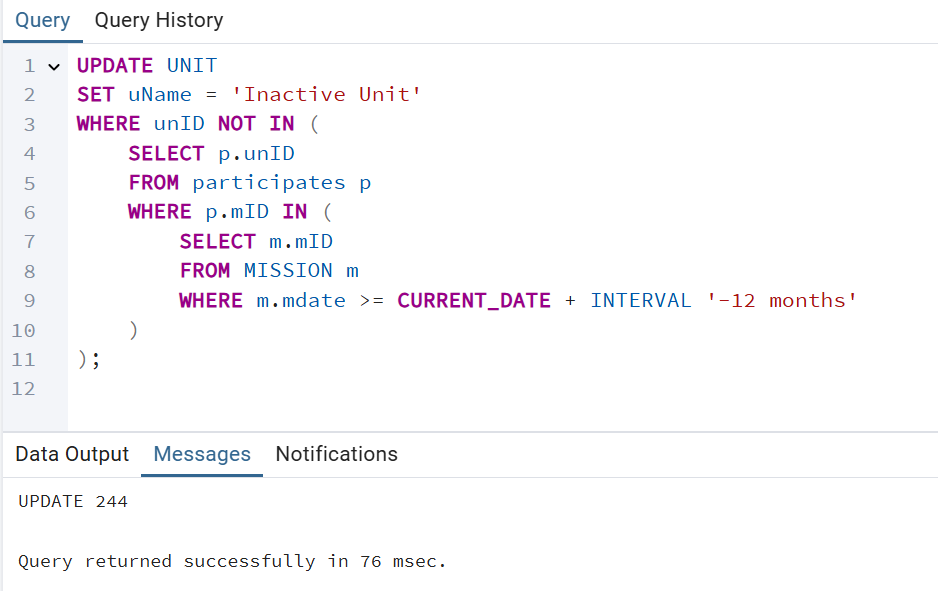
        SELECT m.mID

        FROM MISSION m

        WHERE m.mdate >= CURRENT\_DATE + INTERVAL '-12 months'

    )

);



לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה.

שאילתה 2 : לאחר שיחידה 101 הפכה למסווגת, גם המפקד שלה נדרש לרדת לצללים ועל כן הוא שינה את שמו ל"אנו נימי"

UPDATE SOLDIERS

SET firstName = 'ANO',

    lastName = 'NIMI'

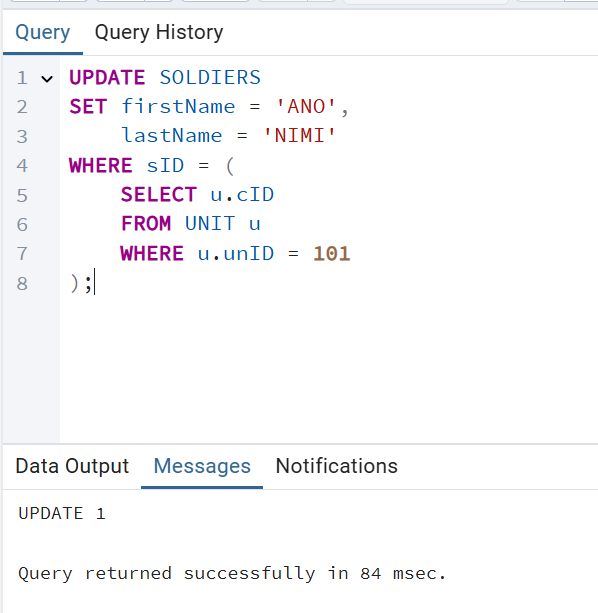
WHERE sID = (

    SELECT u.cID

    FROM UNIT u

    WHERE u.unID = 101

);



לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה.

שאילתה 3: עדכון תאריך שחרור (releaseDate) לכל חייל שהוא מפקד, ששירת עד כה פחות משנתיים וחצי – תוספת של שנה לשחרור.

UPDATE SOLDIERS s

SET releaseDate = releaseDate + INTERVAL '12 months'

WHERE sID IN (

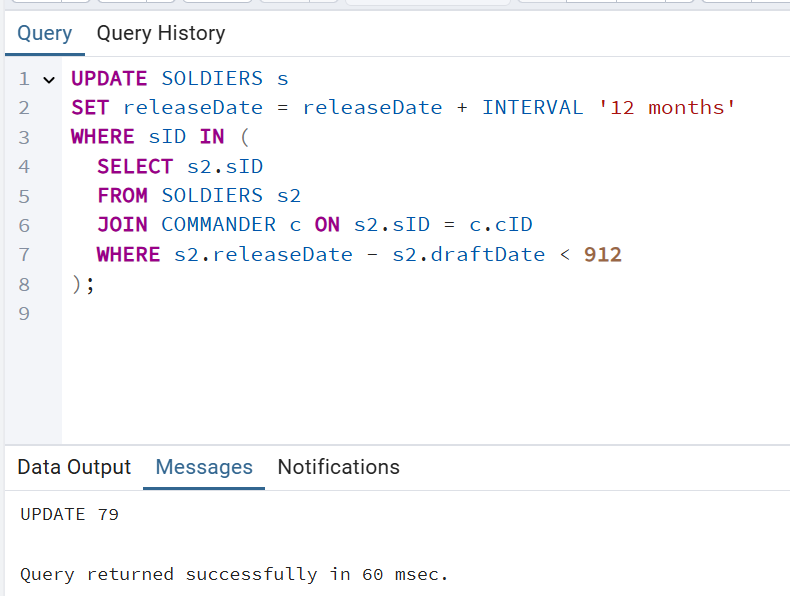
  SELECT s2.sID

  FROM SOLDIERS s2

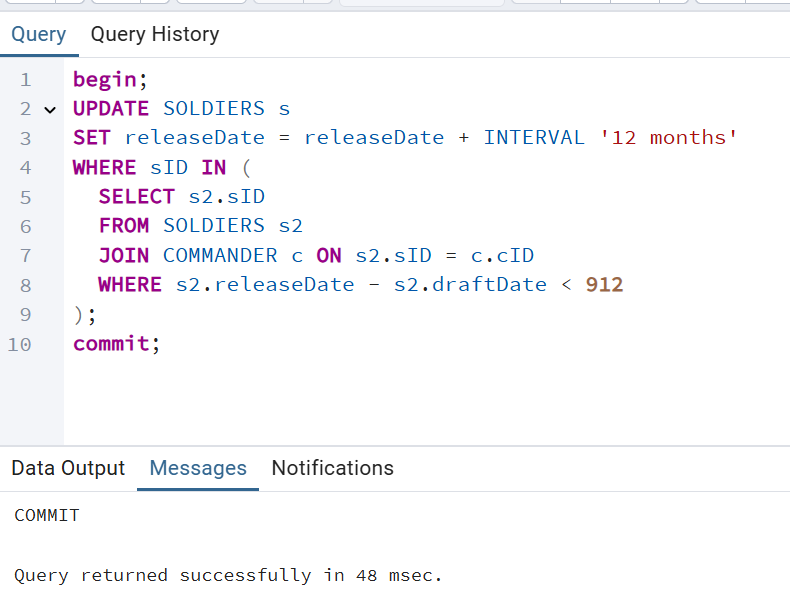
  JOIN COMMANDER c ON s2.sID = c.cID

  WHERE s2.releaseDate - s2.draftDate < 912

);



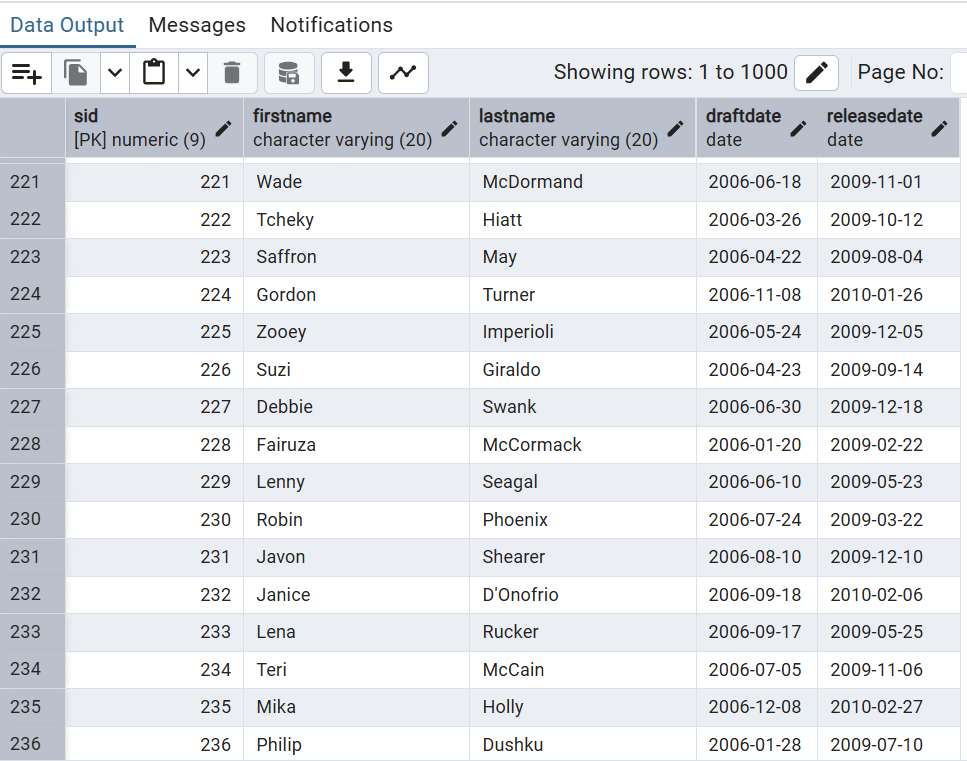
## שאילתת Update עם Commit:



לפני השינוי:

תמונה שמכילה טקסט, צילום מסך, תפריט, מספר

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

אחרי השינוי: 

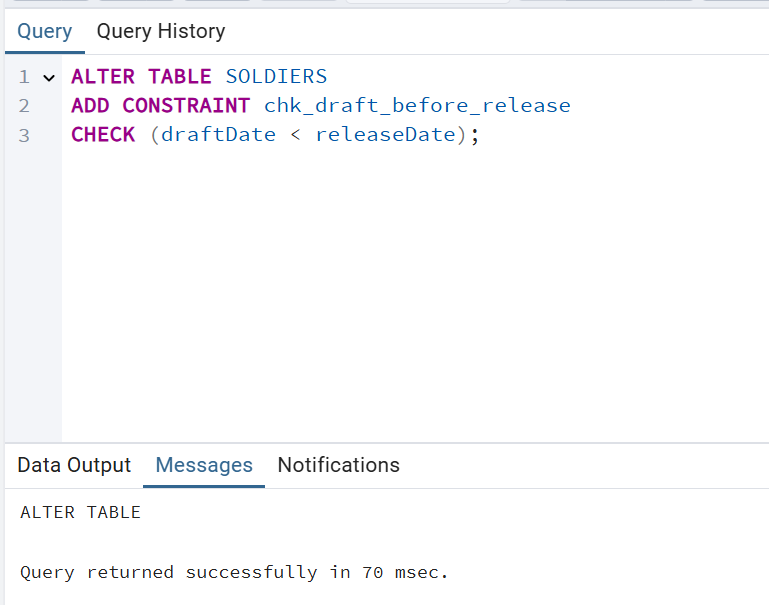
## אילוצים

אילוץ 1 : האילוץ מכריח כי לכל חייל תאריך השחרור חייב להיות אחרי תאריך הגיוס

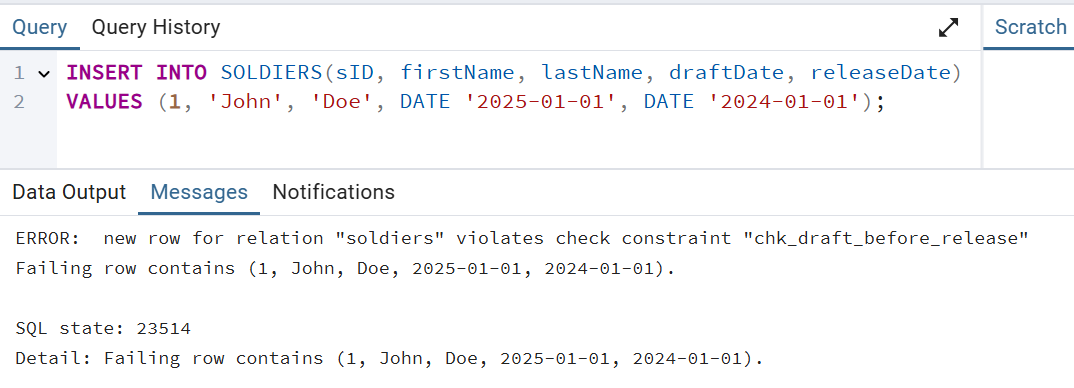
ALTER TABLE SOLDIERS

ADD CONSTRAINT chk\_draft\_before\_release

CHECK (draftDate < releaseDate);



ננסה לבצע פקודה שמפרה את האילוץ



אילוץ 2 : האילוץ מחייב שלכל יחידה יהיה שם מיוחד רק לה

ALTER TABLE UNIT

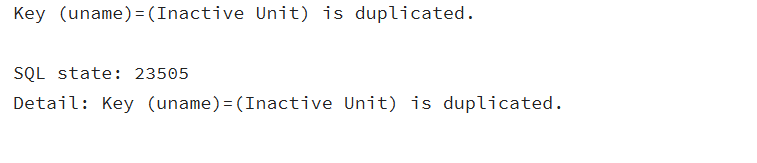
ADD CONSTRAINT unique\_unit\_name

UNIQUE (uName);

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

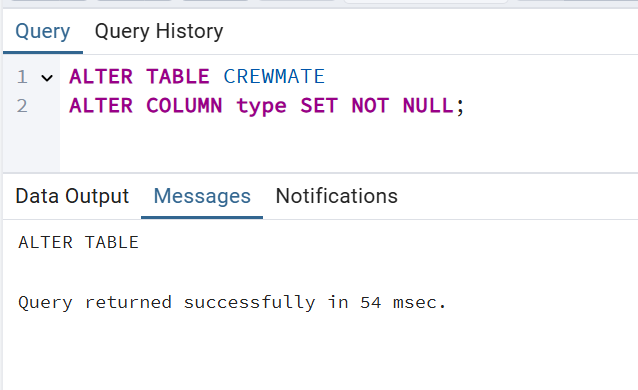
ננסה לבצע פקודה שמפרה את האילוץ



אילוץ 3 : האילוץ מכריח שלכל איש צוות בטנק חייב להיות תפקיד

ALTER TABLE CREWMATE

ALTER COLUMN type SET NOT NULL;



ננסה לבצע פקודה שמפרה את האילוץ

