**דו"ח מיני"פ - חיל האוויר**

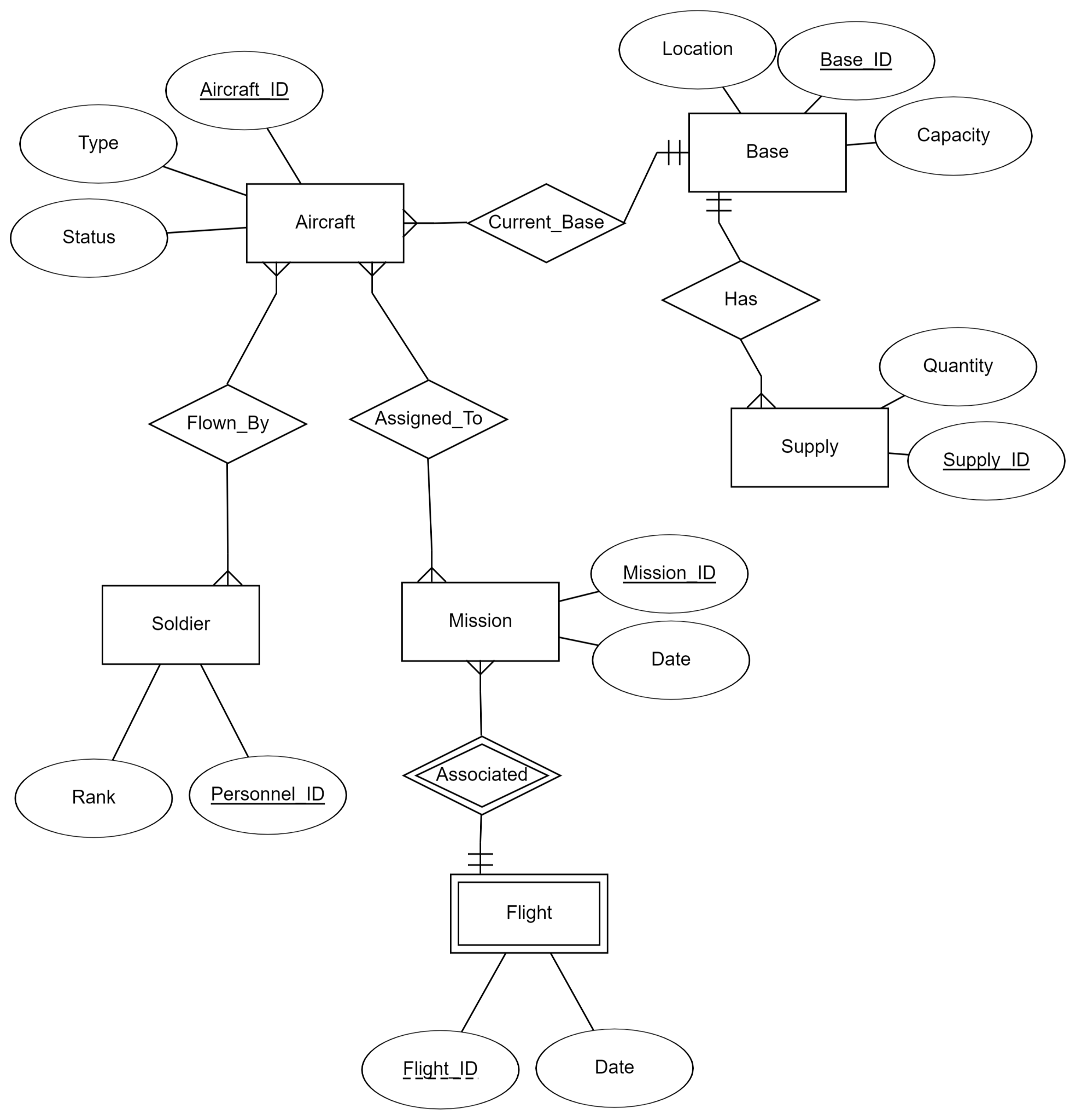
מגישים: ידידיה שאולי ואורי נשר

**תיאור הארגון במילים**

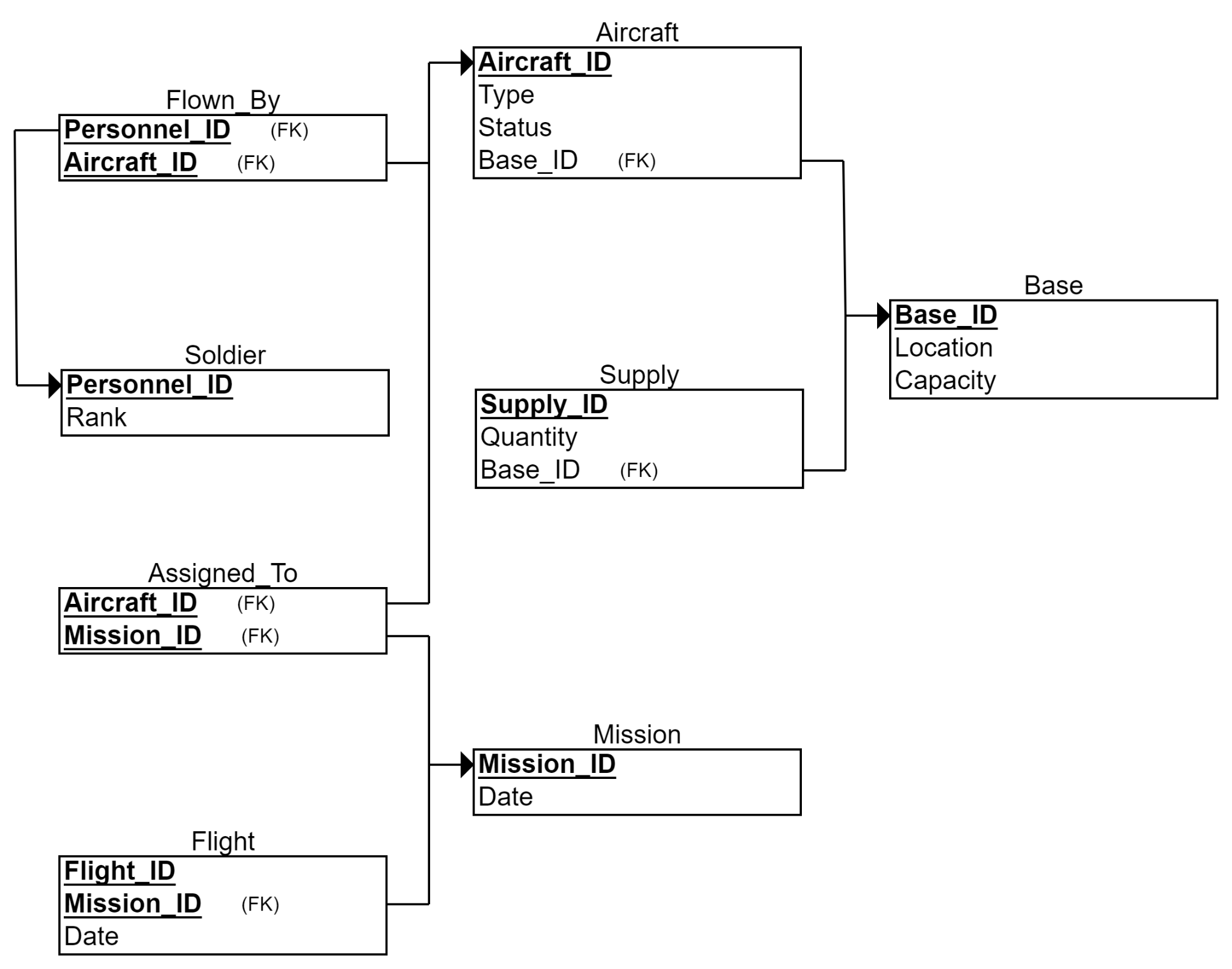
חיל האוויר של צה"ל פועל עם מערכת מסד נתונים מובנית בקפידה שנועדה לנהל את דרישותיו המבצעיות ביעילות.   
מסד נתונים זה משלב ישויות חיוניות כגון מטוסים, חיילים, בסיסים, משימות, אספקה וטיסות.   
הוא משמש עמוד השדרה לתזמון ותיאום משימות, לוגיסטיקה ותחזוקה, ומבטיח הקצאת משאבים מיטבית.  
מידע מפורט על תחזוקת מטוסים, היסטוריית תפעול ומדדי ביצועים מתועד, בעוד שהגישה לנתונים רגישים מוגבלת לצוות מורשה.   
באמצעות גישה שיטתית זו, חיל האוויר משפר את מוכנותו ויעילותו המבצעית, ומקל על הצלחת המשימה.

בנוסף לשמירת נתונים כגון כמויות האספקה בכל בסיס, שמירת הבסיס הנוכחי של כל מטוס, ותיעוד הקצאת כלי הטיס למשימות השונות, המערכת גם שומרת אילו חיילים מטיסים איזה כלי טיס.

## **תרשים ERD**



## **DSD תרשים**



**תיאור כללי של הישויות והקשרים**

### **סכמות**

Aircraft (Aircraft\_ID, Type, Status, Base\_ID)

Soldier (Personnel\_ID, Rank)

Flight (Flight\_ID, Date,Mission\_ID)

Supply (Supply\_ID, Quantity, Base\_ID)

Flown\_By (Personnel\_ID, Aircraft\_ID)

Mission (Mission\_ID, Date)

Base (Base\_ID, Location, Capacity)

Assigned\_To(Aircraft\_ID, Mission\_ID)

### בדיקת נרמול:

* Aircraft

הסוג והסטטוס ובסיס הבית תלויים במזהה כלי הטייס ואין תלות ביניהם - מתקיים 3NF

* Soldier

הדרגה תלויה במזהה-חייל ולכן מתקיים 3NF

* Flight

התאריך והמזהה של המשימה תלויים במזהה-הטיסה ולכן מתקיים 3NF

* Supply

כמות האספקה ומזהה הבסיס תלויים במספר המזהה של האספקה ולכן מתקיים 3NF

* Flown\_By

אין תכונות מלבד המפתח ולכן מתקיים 3NF

* Mission

Date תלוי במפתח, מתקיים 3NF

* Base

המיקום והקיבולת של כל בסיס תלויים במספר המזהה שלו, מתקיים 3NF

* Assigned\_To

אין תכונות מלבד המפתח ולכן מתקיים 3NF

### **קשרים בין ישויות**

* Flown\_By

קשר רבים לרבים, כל חייל יכול להטיס כמה כלי טיס, וכל כלי טיס כמה חיילים יכולים להטיס אותו.

* Assigned\_To

קשר רבים ליחיד (אחד לכל היותר), לכל משימה יכול להיות אפס או יותר כלי טיס שהוקצו לה, אבל לכל כלי טיס יכול להיות רק משימה אחת אליה הוא הוקצה בכל זמן נתון.

* Current\_Base

קשר רבים ליחיד (בדיוק אחד), כל מטוס יכול להיות רק בבסיס אחד בכל זמן נתון (גם כשהוא יוצא למשימות עדיין יש לו בסיס אליו הוא צריך לחזור בתום המשימה) ובבסיס יכולים להיות הרבה כלי טייס שונים באותו זמן.

* Has

קשר רבים ליחיד (בדיוק אחד), כל אספקה יכולה להיות בבסיס אחד בלבד. בבסיס יכולות להיות אספקות רבות מכל מיני סוגים בבת אחת.

* Associated

לא קשר אמיתי אלא Flight היא ישות חלשה ביחס לMission כי אין טיסות שהן לא חלק ממשימה.

## **script של create table**

-- Table for Soldier

CREATE TABLE Soldier (

Personnel\_ID INT NOT NULL,

Rank VARCHAR(25) NOT NULL,

PRIMARY KEY (Personnel\_ID)

);

-- Table for Mission

CREATE TABLE Mission (

Mission\_ID INT NOT NULL,

Date\_Of\_Mission DATE NOT NULL,

PRIMARY KEY (Mission\_ID)

);

-- Table for Base

CREATE TABLE Base (

Base\_ID INT NOT NULL,

Location VARCHAR(25) NOT NULL,

Capacity INT NOT NULL,

PRIMARY KEY (Base\_ID)

);

-- Table for Flight

CREATE TABLE Flight (

Flight\_ID INT NOT NULL,

Date\_Of\_Flight DATE NOT NULL,

Mission\_ID INT NOT NULL,

PRIMARY KEY (Flight\_ID, Mission\_ID),

FOREIGN KEY (Mission\_ID) REFERENCES Mission(Mission\_ID)

);

-- Table for Supply

CREATE TABLE Supply (

Supply\_ID INT NOT NULL,

Quantity INT NOT NULL,

Base\_ID INT NOT NULL,

PRIMARY KEY (Supply\_ID),

FOREIGN KEY (Base\_ID) REFERENCES Base(Base\_ID)

);

-- Table for Aircraft

CREATE TABLE Aircraft (

Aircraft\_ID INT NOT NULL,

Type VARCHAR(25) NOT NULL,

Status VARCHAR(25) NOT NULL,

Base\_ID INT NOT NULL,

PRIMARY KEY (Aircraft\_ID),

FOREIGN KEY (Base\_ID) REFERENCES Base(Base\_ID)

);

-- Table for Assigned\_To

CREATE TABLE Assigned\_To (

Aircraft\_ID INT NOT NULL,

Mission\_ID INT NOT NULL,

PRIMARY KEY (Aircraft\_ID, Mission\_ID),

FOREIGN KEY (Aircraft\_ID) REFERENCES Aircraft(Aircraft\_ID),

FOREIGN KEY (Mission\_ID) REFERENCES Mission(Mission\_ID)

);

-- Table for Flown\_By

CREATE TABLE Flown\_By (

Personnel\_ID INT NOT NULL,

Aircraft\_ID INT NOT NULL,

PRIMARY KEY (Personnel\_ID, Aircraft\_ID),

FOREIGN KEY (Personnel\_ID) REFERENCES Soldier(Personnel\_ID),

FOREIGN KEY (Aircraft\_ID) REFERENCES Aircraft(Aircraft\_ID)

);

);

CREATE TABLE Assigned\_To

(

Aircraft\_ID INT NOT NULL,

Mission\_ID INT NOT NULL,

PRIMARY KEY (Aircraft\_ID, Mission\_ID),

FOREIGN KEY (Aircraft\_ID) REFERENCES Aircraft(Aircraft\_ID),

FOREIGN KEY (Mission\_ID) REFERENCES Mission(Mission\_ID)

);

CREATE TABLE Flown\_By

(

Personnel\_ID INT NOT NULL,

Aircraft\_ID INT NOT NULL,

PRIMARY KEY (Personnel\_ID, Aircraft\_ID),

FOREIGN KEY (Personnel\_ID) REFERENCES Soldier(Personnel\_ID),

FOREIGN KEY (Aircraft\_ID) REFERENCES Aircraft(Aircraft\_ID)

);

## **script של drop table**

-- Drop the Mission table

DROP TABLE IF EXISTS Mission;

-- Drop the Flown\_By table

DROP TABLE IF EXISTS Flown\_By;

-- Drop the Base table

DROP TABLE IF EXISTS Base;

-- Drop the Supply table

DROP TABLE IF EXISTS Supply;

-- Drop the Flight table

DROP TABLE IF EXISTS Flight;

-- Drop the Soldier table

DROP TABLE IF EXISTS Soldier;

-- Drop the Aircraft table

DROP TABLE IF EXISTS Aircraft;

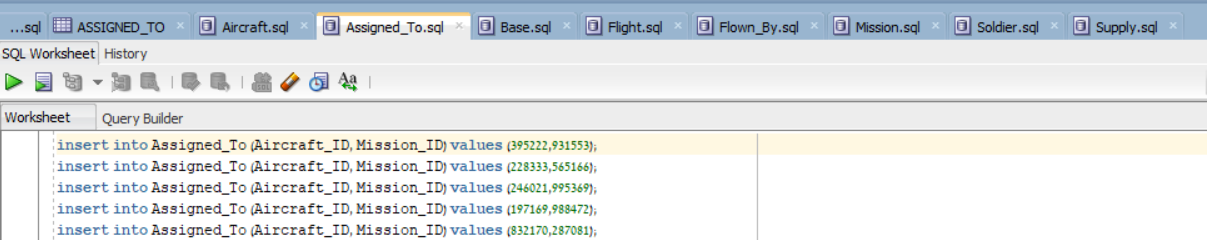
-- Drop the Assigned\_To table

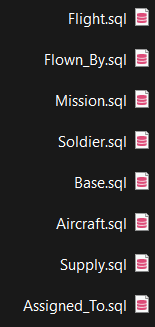
DROP TABLE IF EXISTS Assigned\_To;

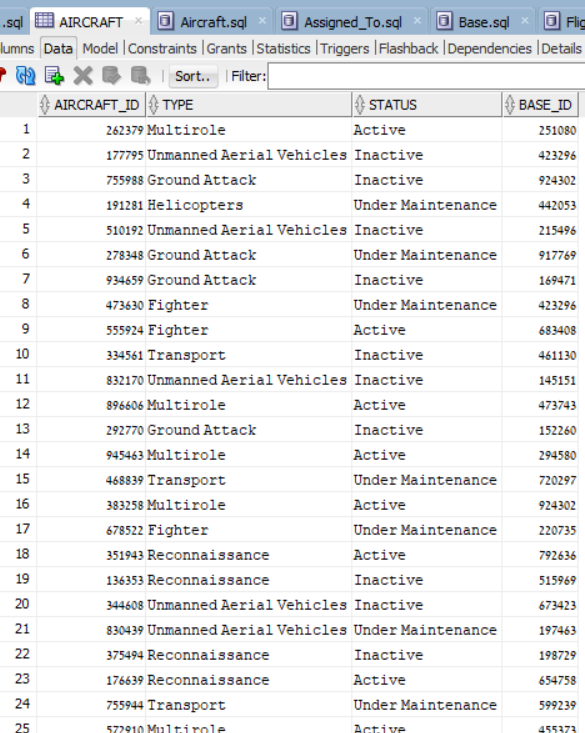
## הכנסת נתונים באמצעות Data Generator

### Mockaroo

יצרנו את כל הסכמות באתר, וביצענו ג'ינרוט קבצים מסוג SQL.

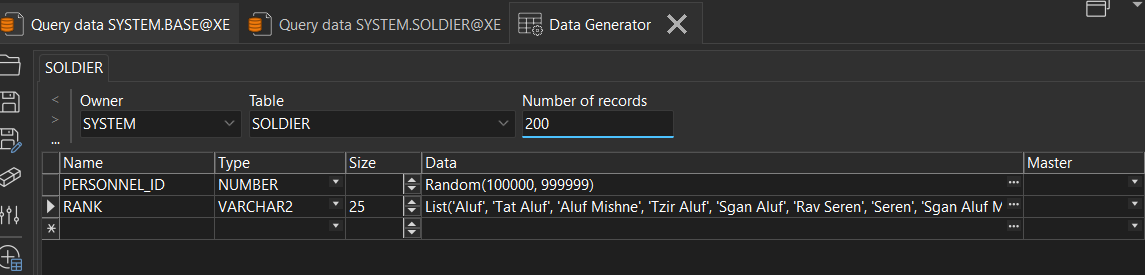
ולאחר מכן פתחנו את כולם בSQL developer והרצנו את הקוד שלהם.

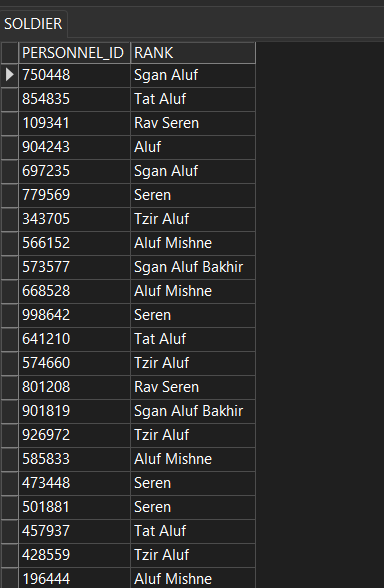




### 

### Data Generator





### Python

import sqlite3

import random

# List of airbase locations

locations = [

'Ramat David Airbase',

'Tel Nof Airbase',

'Hatzerim Airbase',

'Hatzor Airbase',

'Nevatim Airbase',

'Ovda Airbase',

'Palmachim Airbase',

'Sde Dov Airbase',

'Uvda Airbase'

]

# Generate data for 200 rows

bases = []

for i in range(1, 201):

base\_id = random.randint(100000, 999999) #

location = random.choice(locations)

capacity = random.randint(2000, 10000) # Assuming capacity ranges between 500 and 5000

bases.append((base\_id, location, capacity))

# Connect to SQLite database (or create it if it doesn't exist)

conn = sqlite3.connect('military.db')

# Create a cursor object

cur = conn.cursor()

# Create the Base table

cur.execute('''

CREATE TABLE IF NOT EXISTS Base (

Base\_ID INT NOT NULL,

Location VARCHAR(25) NOT NULL,

Capacity INT NOT NULL,

PRIMARY KEY (Base\_ID)

);

''')

# Insert sample data into the Base table

cur.executemany('INSERT INTO Base (Base\_ID, Location, Capacity) VALUES (?, ?, ?);', bases)

# Commit the transaction

conn.commit()

# Query the database to confirm data insertion

cur.execute('SELECT \* FROM Base')

rows = cur.fetchall()

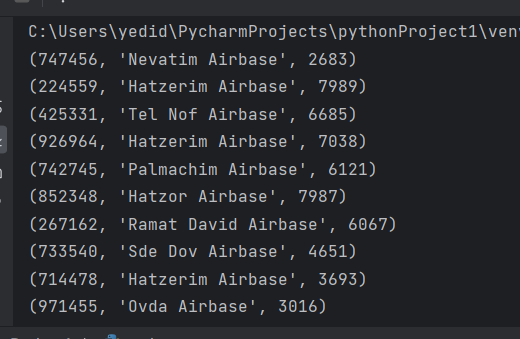
# Print the inserted data

for row in rows:

print(row)

# Close the connection

conn.close()



## גיבוי

