מיני פרויקט בבסיסי נתונים:

אפסנאות

מגישים:

דניאל צירקין – 214389751

ספיר בשן – 214103368

קישור לGItHub - https://github.com/ Mini_Project_DB

תוכן עניינים

תיאור הארגון:

הארגון הוא מערכת ניהולית המתמקדת בניהול ובשליטה בתהליכי האספקה והשיווק של ציוד צבאי ליחידות צבאיות שונות. כאמצעי עבודה עם מידע מרובה-מקורות, המערכת מתמקדת באכיפת התהליכים וההנחיות של הארגון בכל הנוגע לניהול האספקה.

<u>כעת, בוא נבחן יישויות עיקריות במערכת:</u>

חייל:(Soldier)

מייצגים את כלל החיילים בשירות הצבאי. כל חייל שייך ליחידה אחת.

יחידות:(Units)

מייצגות את יחידות השונות בצבא. לכל יחידה קצין (מפקד) אחד.

קצין (officer):

מייצגים את הקצינים של היחידות כאשר כל קצין הינו סוג של חייל.

ציוד:(Equipment)

מייצגים את הציוד הצבאי השונה, כגון כלי נשק, חומרי ציוד ותשלומים. לכל ציוד ישנו מחסן יחיד המאחסן אותו.

מחסנים:(Warehouses)

מייצגים את המקומות שבהם מאוחסן הציוד הצבאי.

ספקים:(Suppliers)

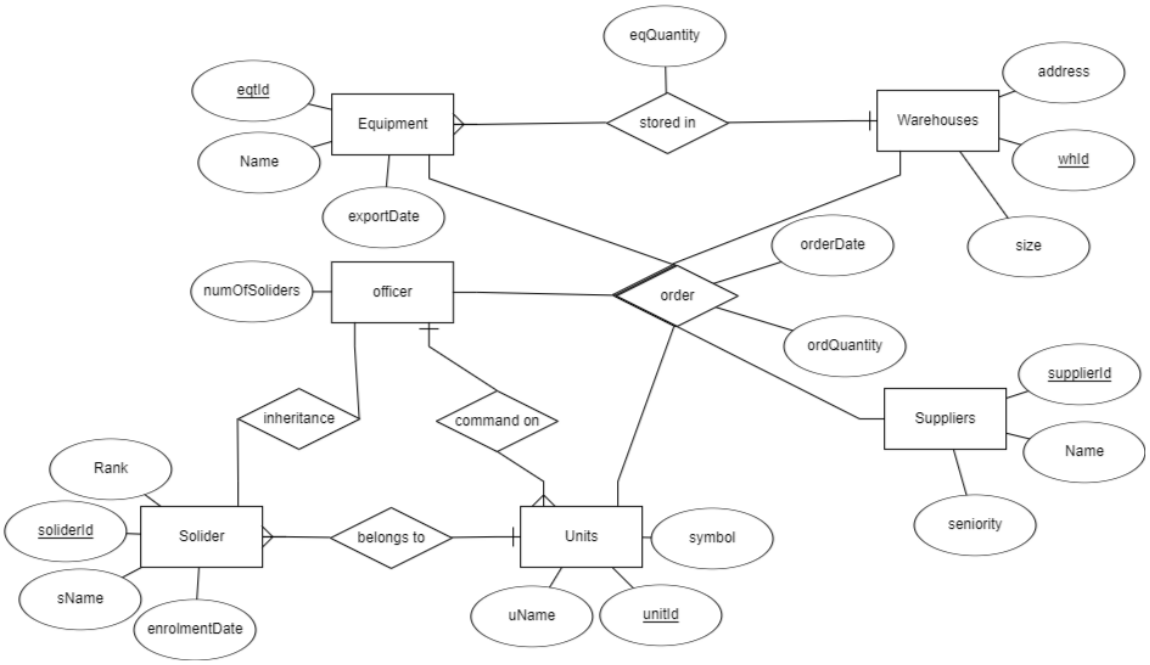
מייצגים את הספקים השונים שמספקים ציוד לצבא.

<u>כל ישות במערכת מכילה מאפיינים מזהים שמאפשרים זיהוי ייחודי ומאפיינים נוספים שמתארים את הישות במידת הצורך.</u>

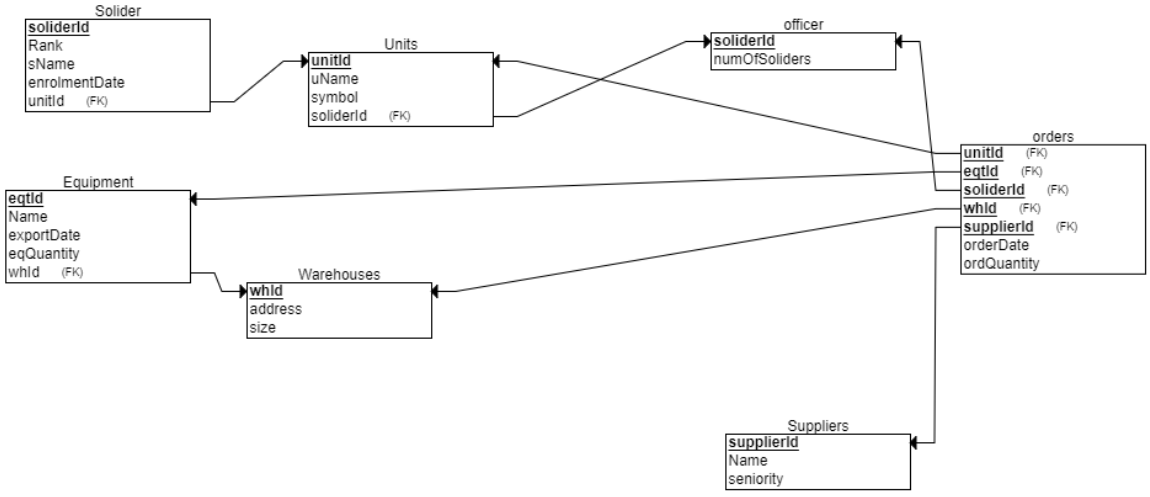
<u>הזמנת ציוד עבור יחידה:</u>

קצין חייל מבצע הזמנת ציוד צבאי עבור יחידה מסוימת. הפעולה כוללת בחירת הציוד הרצוי, ציון כמות הציוד, והגדרת זמן הזמנה. הציוד נבחר מתוך הציוד הזמין במחסנים המתאימים ליחידה. זמן הזמנה יכול להיות מוגדר מראש או להיות זמן ספציפי עבור ההזמנה הספציפית ועובר בין זמן גיוס לזמן גיוס. בכל הזמנה יוזמן סוג ציוד אחד.

דיאגרמת ERD:



דיאגרמת DSD:



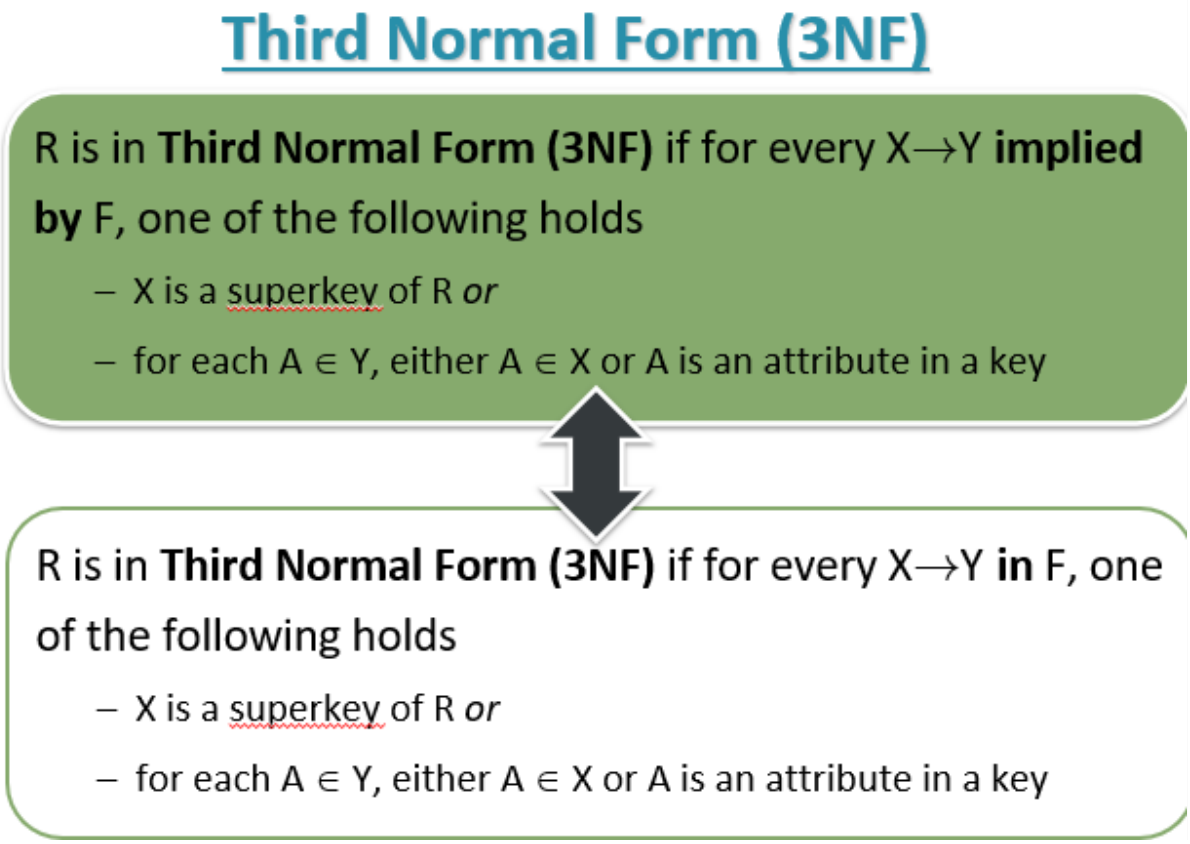
פקודות create table_:

```
CREATE TABLE Warehouse
(
 address VARCHAR2(35) NOT NULL,
 whId INT NOT NULL,
 size INT NOT NULL,
 PRIMARY KEY (whId)
);

CREATE TABLE Equipment
(
 eqtId INT NOT NULL,
 Name VARCHAR2(35) NOT NULL,
 exportDate DATE NOT NULL,
 eqQuantity INT NOT NULL,
 whId INT NOT NULL,
 PRIMARY KEY (eqtId),
 FOREIGN KEY (whId) REFERENCES Warehouse(whId)
);

CREATE TABLE Suppliers
(
 supplierId INT NOT NULL,
 Name VARCHAR2(35) NOT NULL,
 Seniority INT NOT NULL,
 PRIMARY KEY (supplierId)
);
```

```
CREATE TABLE Soldiers
(
 soldierId INT NOT NULL,
 Rank INT NOT NULL,
 sName VARCHAR2(35) NOT NULL,
 enrolmentDate DATE NOT NULL,
 unitId INT NOT NULL,
 PRIMARY KEY (soldierId),
 FOREIGN KEY (unitId) REFERENCES armyUnit(unitId)
);

CREATE TABLE armyUnit
(
 uName VARCHAR2(35) NOT NULL,
 unitId INT NOT NULL,
 symbol VARCHAR2(100) NOT NULL,
 PRIMARY KEY (unitId),
);

CREATE TABLE Officer
(
 numOfSoldiers INT NOT NULL,
 soldierId INT NOT NULL,
 PRIMARY KEY (soldierId),
 FOREIGN KEY (soldierId) REFERENCES Soldiers(soldierId)
);
```

```
CREATE TABLE Orders
(
 unitId INT NOT NULL,
 eqtId INT NOT NULL,
 soldierId INT NOT NULL,
 whId INT NOT NULL,
 supplierId INT NOT NULL,
 orderDate DATE NOT NULL,
 ordQuantity INT NOT NULL,
 PRIMARY KEY (unitId, eqtId, soldierId, whId, supplierId),
 FOREIGN KEY (unitId) REFERENCES armyUnit(unitId),
 FOREIGN KEY (eqtId) REFERENCES Equipment(eqtId),
 FOREIGN KEY (soldierId) REFERENCES Soldiers(soldierId),
 FOREIGN KEY (whId) REFERENCES Warehouse(whId),
 FOREIGN KEY (supplierId) REFERENCES Suppliers(supplierId)
);
```

נירמול ל3NF:



כדי לקבוע אם הסכימה הנתונה נמצאת בצורת נורמליזציה שלישית (3NF) באמצעות האלגוריתם שמוצג בתמונה, נצטרך לבדוק את התלות הפונקציונלית (FDs) לכל טבלה ולהבטיח שהן עונות על הקריטריונים:

עבור כל תלות פונקציונלית X -> Y בקבוצת התלות הפונקציונלית F, אחד מהבאים חייב להתקיים:

- X הוא מפתח על של (R הטבלה).
- לכל תכונה A ב Y, A -נמצא ב X -או A היא תכונה במפתח.

בואו נבחן כל טבלה בסכימה שלכם ונחיל את האלגוריתם הזה:

מחסנים: (Warehouses) -

CREATE TABLE Warehouse

```
(
 address VARCHAR2(35) NOT NULL,
 whId INT NOT NULL,
 size INT NOT NULL,
 PRIMARY KEY (whId)
);
```

עפ"י הסכמה:

- whId -> address, size

כאן whId, הוא מפתח על (מפתח ראשי).

whId - הוא מפתח על, כך שהטבלה נמצאת ב-NF.3

ציוד (Equipments):

```
CREATE TABLE Equipment
(
 eqtId INT NOT NULL,
 Name VARCHAR2(35) NOT NULL,
 exportDate DATE NOT NULL,
 eqQuantity INT NOT NULL,
 whId INT NOT NULL,
 PRIMARY KEY (eqtId),
 FOREIGN KEY (whId) REFERENCES Warehouse(whId)
);
```

עפ"י הסכמה:

- eqtId -> Name, exportDate, eqQuantity, whId

כאן eqtId, הוא מפתח על (מפתח ראשי).

eqtId - הוא מפתח על, כך שהטבלה נמצאת ב-NF.3

ספקים (Suppliers):

```
CREATE TABLE Suppliers
(
 supplierId INT NOT NULL,
 Name VARCHAR2(35) NOT NULL,
 Seniority INT NOT NULL,
 PRIMARY KEY (supplierId)
);
```

עפ"י הסכמה:

- supplierId -> Name, Seniority

כאן supplierId, הוא מפתח על (מפתח ראשי).

- supplierId 3-הוא מפתח על, כך שהטבלה נמצאת ב-NF.

חיילים (Soldiers):

```
CREATE TABLE Soldiers
(
 soldierId INT NOT NULL,
 Rank INT NOT NULL,
 sName VARCHAR2(35) NOT NULL,
 enrolmentDate DATE NOT NULL,
 unitId INT NOT NULL,
 PRIMARY KEY (soldierId),
 FOREIGN KEY (unitId) REFERENCES armyUnit(unitId)
);
```

עפ"י הסכמה:

- soldierId -> Rank, sName, enrolmentDate, unitId

כאן soldierId, הוא מפתח על (מפתח ראשי).

soldierId - הוא מפתח על, כך שהטבלה נמצאת ב-NF.3

יחידות:(Units)

```
CREATE TABLE armyUnit
(
 uName VARCHAR2(35) NOT NULL,
 unitId INT NOT NULL,
 symbol VARCHAR2(100) NOT NULL,
 PRIMARY KEY (unitId)
);
```

עפ"י הסכמה:

- unitId -> uName, symbol

כאן unitId, הוא מפתח על (מפתח ראשי).

unitId - הוא מפתח על, כך שהטבלה נמצאת ב-NF.3

קצינים: (Officers)

```
CREATE TABLE Officer
(
 numOfSoldiers INT NOT NULL,
 soldierId INT NOT NULL,
 PRIMARY KEY (soldierId),
 FOREIGN KEY (soldierId) REFERENCES Soldiers(soldierId)
);
```

עפ"י הסכמה:

- soldierId -> numOfSoldiers

כאן soldierId, הוא מפתח על (מפתח ראשי).

soldierId - הוא מפתח על, כך שהטבלה נמצאת ב-NF.3

הזמנות (Orders):

```
CREATE TABLE Orders
(
  unitId INT NOT NULL,
  eqtId INT NOT NULL,
  soldierId INT NOT NULL,
  whId INT NOT NULL,
  supplierId INT NOT NULL,
  orderDate DATE NOT NULL,
  ordQuantity INT NOT NULL,
  PRIMARY KEY (unitId, eqtId, soldierId, whId, supplierId),
  FOREIGN KEY (unitId) REFERENCES armyUnit(unitId),
  FOREIGN KEY (eqtId) REFERENCES Equipment(eqtId),
  FOREIGN KEY (soldierId) REFERENCES Soldiers(soldierId),
  FOREIGN KEY (whId) REFERENCES Warehouse(whId),
  FOREIGN KEY (supplierId) REFERENCES Suppliers(supplierId)
);
```

עפ"י הסכמה:

- {unitId, eqtId, soldierId, whId, supplierId} -> orderDate, ordQuantity

כאן, השילוב של unitId, eqtId, soldierId, whId, supplierId הוא מפתח על (מפתח ראשי מורכב).

-השילוב של unitId, eqtId, soldierId, whId, supplierId הוא מפתח על, כך שהטבלה נמצאת ב-NF3

לסיכום

על ידי יישום האלגוריתם הנתון, אישרנו שכל טבלה נמצאת בצורת נורמליזציה מכיוון שכל התלות הפונקציונלית עונות על התנאים: הצד השמאלי (3NF) שלישית הוא תמיד מפתח על ואין תלות טרנזיטיבית. לכן, הסכימה שלכם מנורמלת (X) בצורה נכונה ל-3NF.

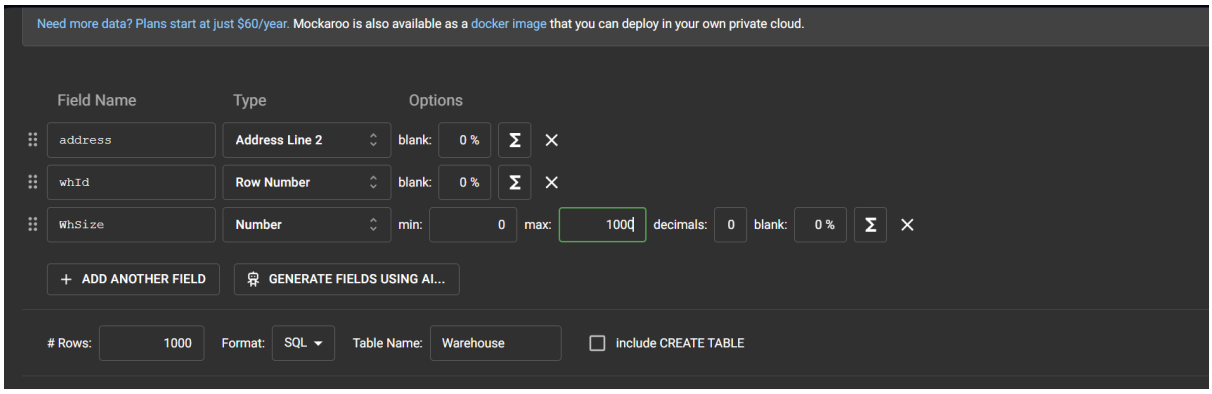
אכלוס טבלאות:

הכנסנו את הנתונים בשלושה דרכים:

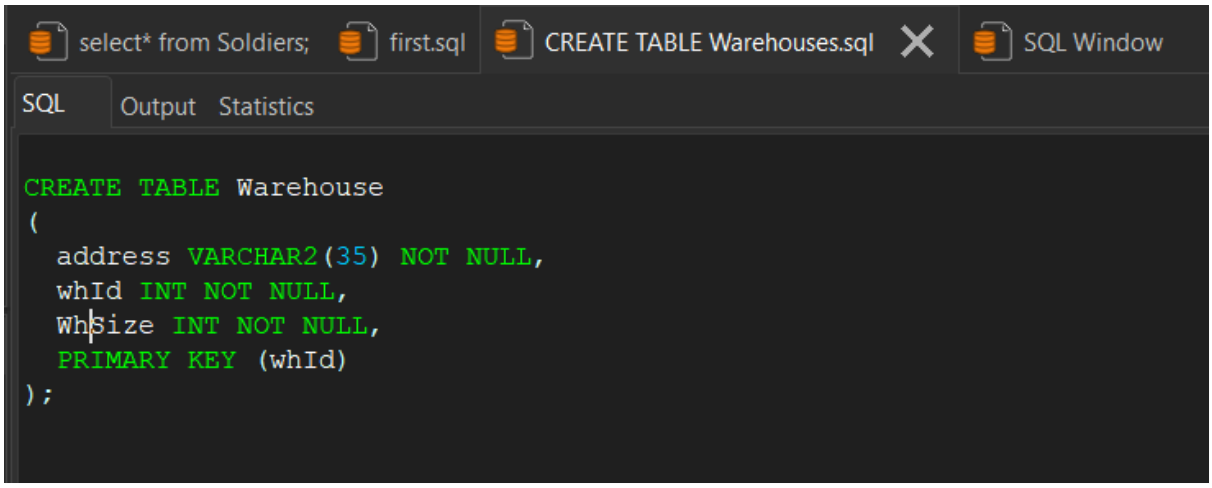
דרך א - mockaroo:

יצרנו נתונים ע"י אתר https://www.mockaroo.com.

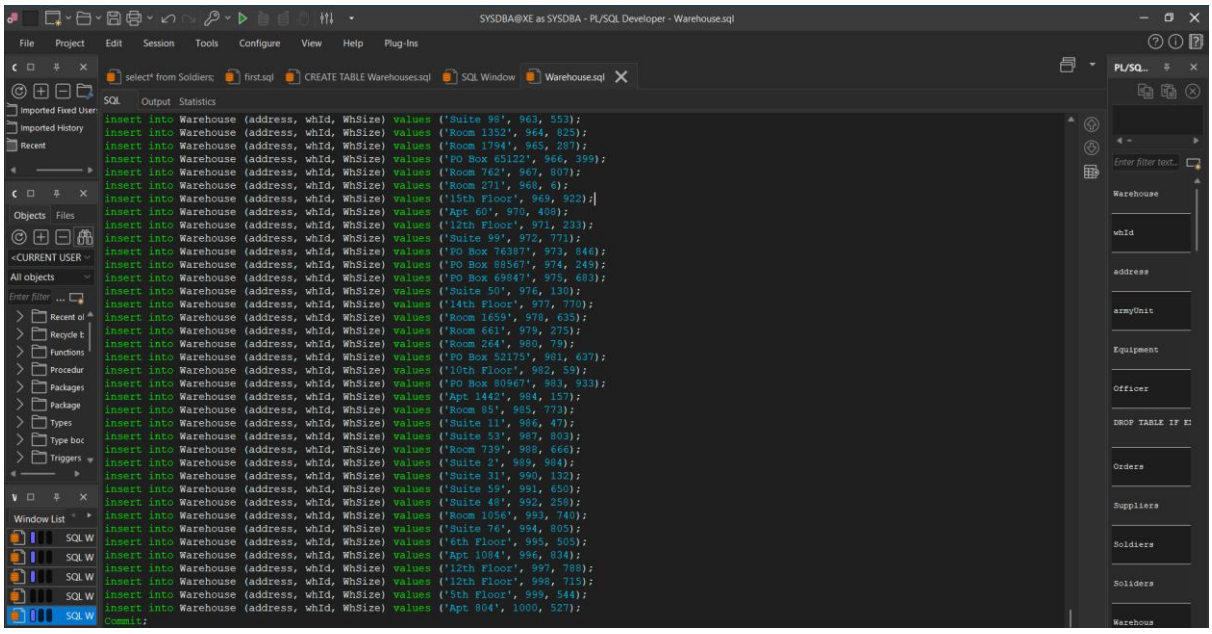
עבור Warehouse



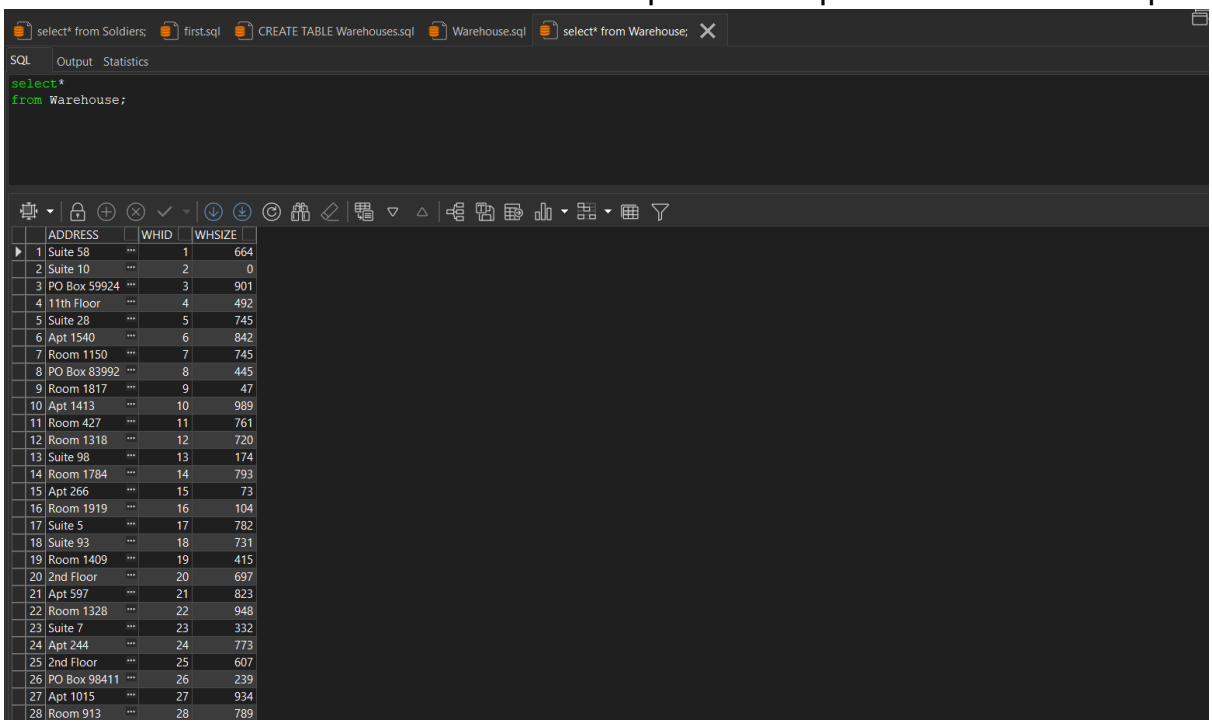
וכך יצרנו טבלת נתונים לכל הטבלאות שלנו באופן אוטומטי
התחלנו עם הטבלה הזאת:



והנה הפקודות שנוצרו.



בדקנו שכל הרשומות אכן נכנסו לתוך הטבלה:



	ADDRESS	WHID	WHSIZE
1	Suite 58	1	664
2	Suite 10	2	0
3	PO Box 59924	3	901
4	11th Floor	4	492
5	Suite 28	5	745
6	Apt 1540	6	842
7	Room 1150	7	745
8	PO Box 83992	8	445
9	Room 1817	9	47
10	Apt 1413	10	989
11	Room 427	11	761
12	Room 1318	12	720
13	Suite 98	13	174
14	Room 1784	14	793
15	Apt 266	15	73
16	Room 1919	16	104
17	Suite 5	17	782
18	Suite 93	18	731
19	Room 1409	19	415
20	2nd Floor	20	697
21	Apt 597	21	823
22	Room 1328	22	948
23	Suite 7	23	332
24	Apt 244	24	773
25	2nd Floor	25	607
26	PO Box 98411	26	239
27	Apt 1015	27	934
28	Room 913	28	789

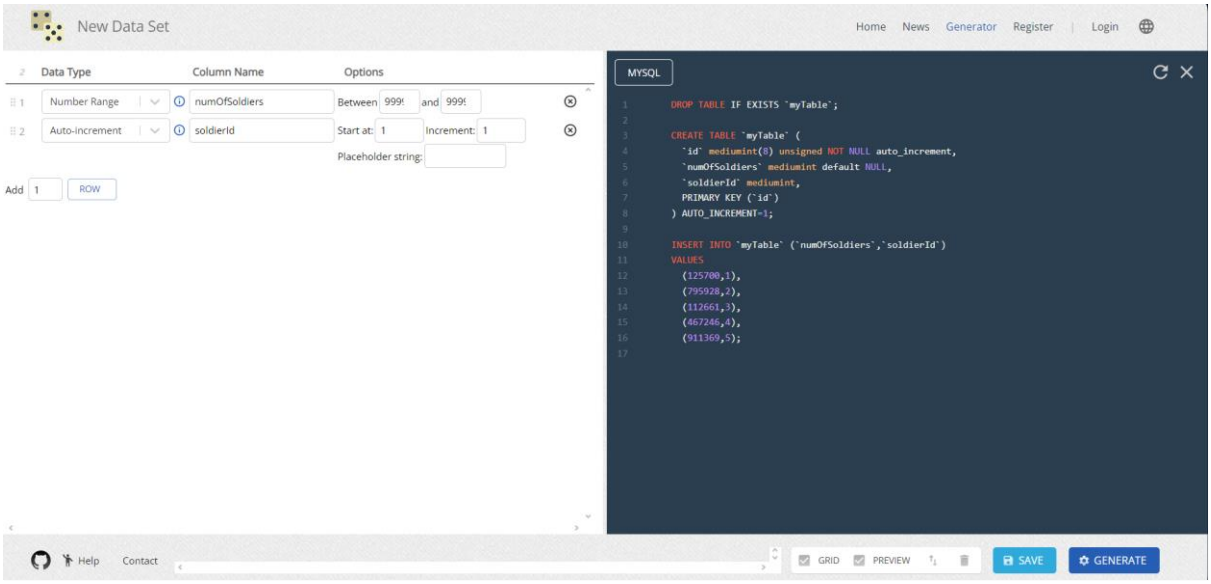
הפעלנו פעולת desc לtable:

```
SQL> DESC Warehouse
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADDRESS                                   NOT NULL VARCHAR2(35)
 WHID                                      NOT NULL NUMBER(38)
 WHSIZE                                    NOT NULL NUMBER(38)

SQL> A
```

דרך ב - data generator :

יצירת נתונים ע"י אתר https://generatedata.com/generator

עבור officers



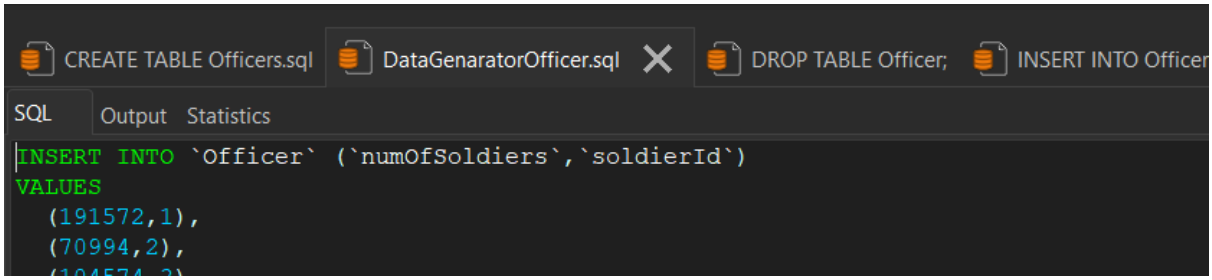
אנו מכניסים נתונים לCREATE TABLE זה:

SQL Output Statistics

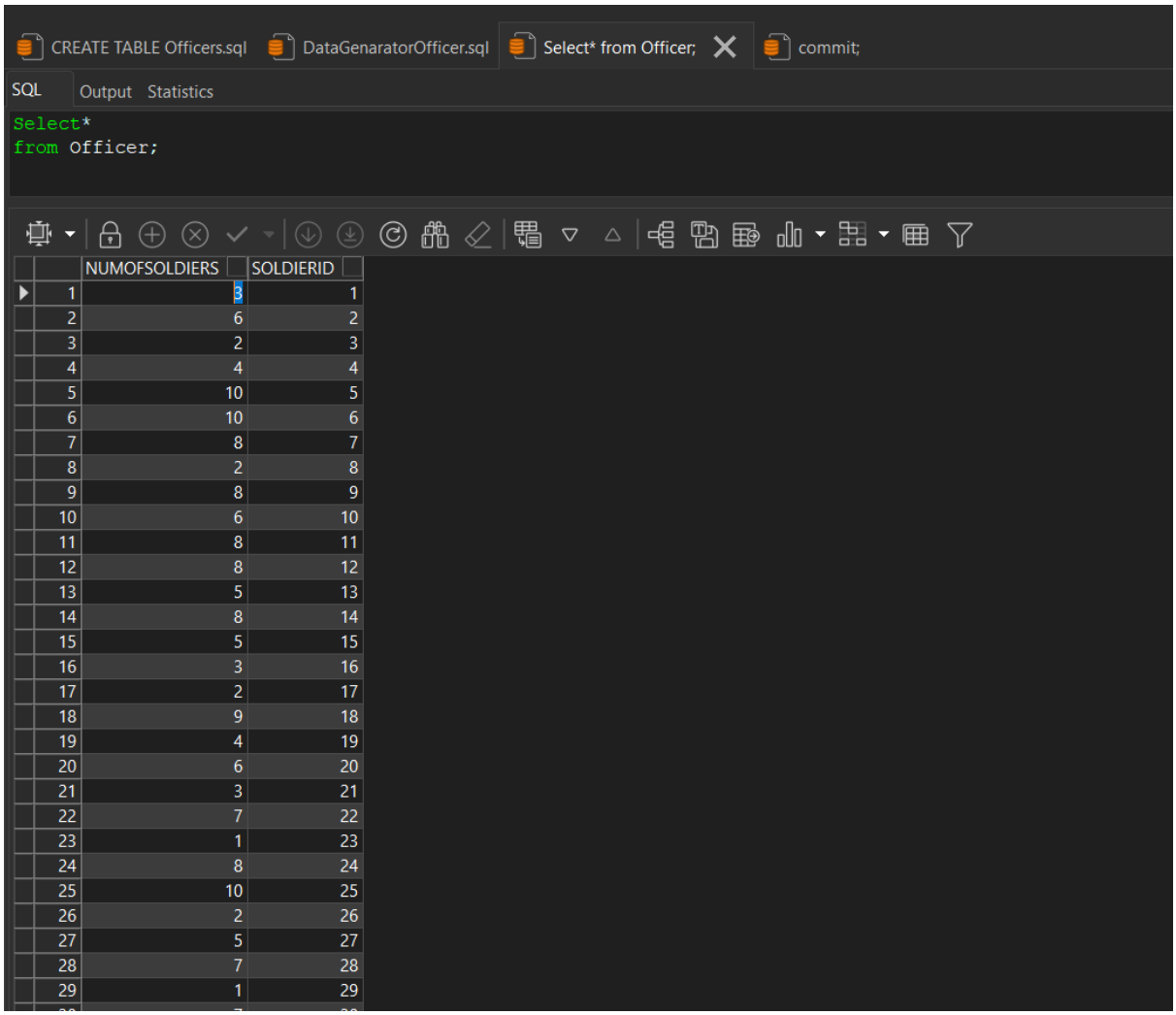
```
CREATE TABLE Officer
(
  numOfSoldiers INT NOT NULL,
  soldierId INT NOT NULL,
  FOREIGN KEY (soldierId) REFERENCES Soldiers(soldierId)
);
```

הנה הפקודות שאנו מריצים:

```
INSERT INTO `myTable` (`numOfSoldiers`,`soldierId`)
VALUES
  (125700,1),
  (795928,2),
  (112661,3),
  (467246,4),
  (911369,5);
```



והנתונים נכנסו בהצלחה:



	NUMOFSOLDIERS	SOLDIERID
1	3	1
2	6	2
3	2	3
4	4	4
5	10	5
6	10	6
7	8	7
8	2	8
9	8	9
10	6	10
11	8	11
12	8	12
13	5	13
14	8	14
15	5	15
16	3	16
17	2	17
18	9	18
19	4	19
20	6	20
21	3	21
22	7	22
23	1	23
24	8	24
25	10	25
26	2	26
27	5	27
28	7	28
29	1	29

בנוסף הפעלנו פעולת disc על התרשים המבוקש

```
SQL> desc officer
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NUMOFSOLDIERS                             NOT NULL NUMBER(38)
 SOLDIERID                                 NOT NULL NUMBER(38)
```

דרך ג - python insert :

יצירה ע"י פייתון:

armyUnit עבור

armyUnit.py U | creat table armyUnits.sql U | Officers.py | CREATE TABLE Officers.sql M

Python code > armyUnit.py > ...

```python
import random

# Define some example unit names and symbols
unit_names = [
    "Alpha", "Bravo", "Charlie", "Delta", "Echo",
    "Foxtrot", "Golf", "Hotel", "India", "Juliet"
]

symbols = [
    "Sword", "Shield", "Bow", "Medal", "Bomb",
    "Helicopter", "Rocket", "Anchor", "Eagle", "Star"
]

# Generate SQL INSERT statements
insert_statements = []
for i in range(1, 1001):
    unit_name = random.choice(unit_names)
    unit_id = i  # Assuming unitId starts from 1 and increments by 1
    symbol = random.choice(symbols)
    insert_statement = f"INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('{unit_name}', {unit_id}, '{symbol}');"
    insert_statements.append(insert_statement)

# Print the generated SQL statements
for statement in insert_statements:
    print(statement)
```

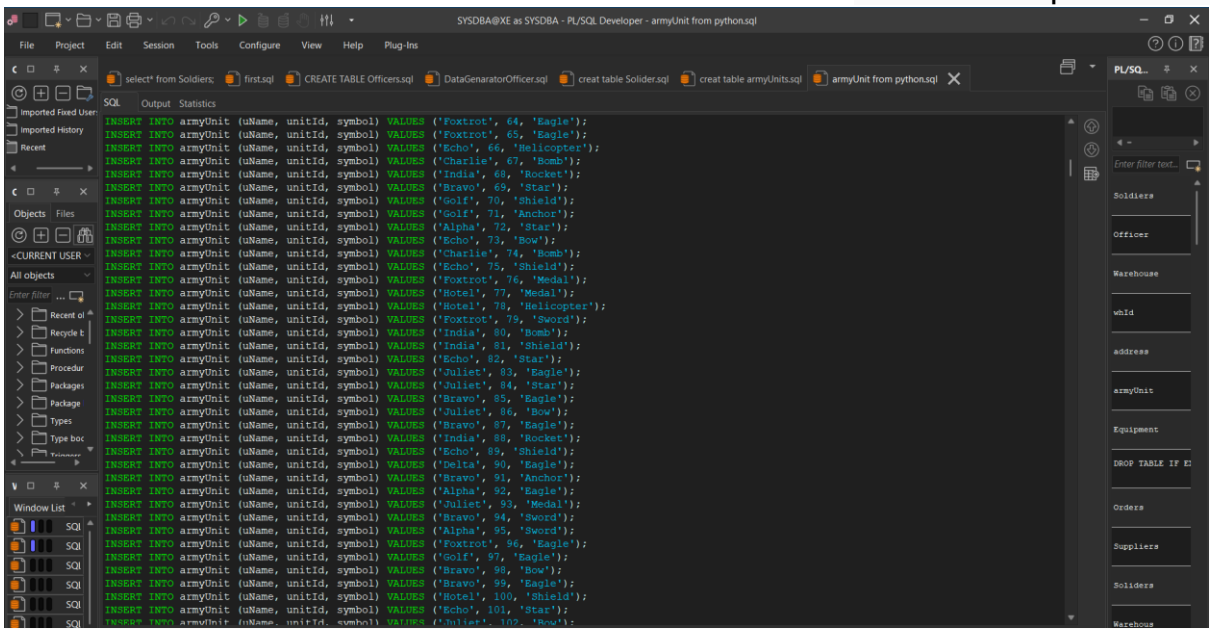
PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | GITLENS

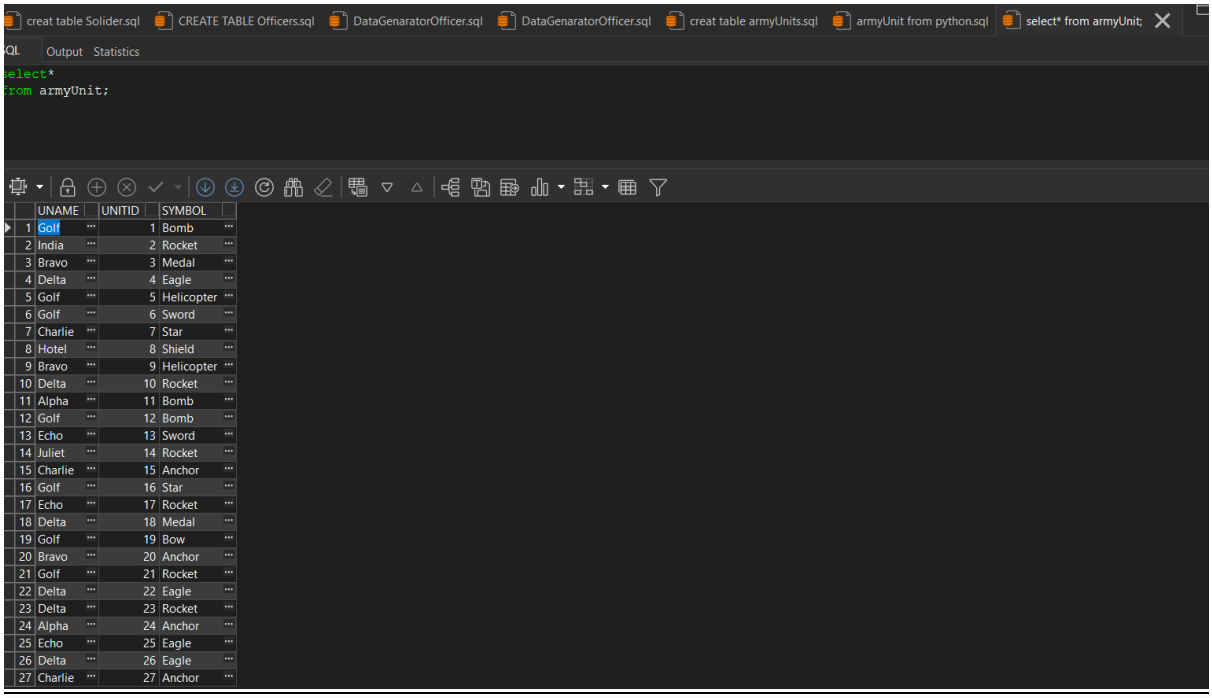
cmd

Code

```
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Delta', 994, 'Bow');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Hotel', 995, 'Bow');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Delta', 996, 'Star');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('India', 997, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Charlie', 998, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 999, 'Helicopter');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('India', 1000, 'Star');
```

והנה הפקודות שנוצרו:





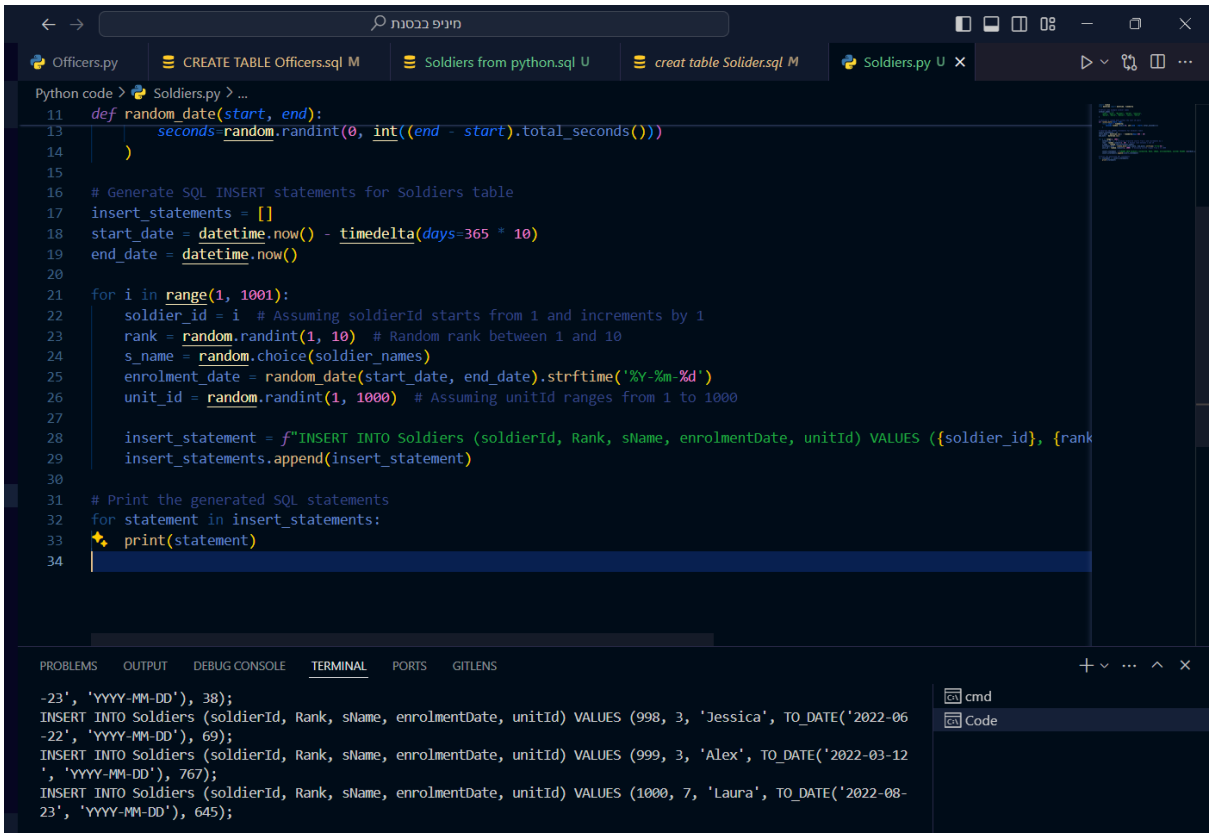
	UNAME	UNITID	SYMBOL
1	Golf	1	Bomb
2	India	2	Rocket
3	Bravo	3	Medal
4	Delta	4	Eagle
5	Golf	5	Helicopter
6	Golf	6	Sword
7	Charlie	7	Star
8	Hotel	8	Shield
9	Bravo	9	Helicopter
10	Delta	10	Rocket
11	Alpha	11	Bomb
12	Golf	12	Bomb
13	Echo	13	Sword
14	Juliet	14	Rocket
15	Charlie	15	Anchor
16	Golf	16	Star
17	Echo	17	Rocket
18	Delta	18	Medal
19	Golf	19	Bow
20	Bravo	20	Anchor
21	Golf	21	Rocket
22	Delta	22	Eagle
23	Delta	23	Rocket
24	Alpha	24	Anchor
25	Echo	25	Eagle
26	Delta	26	Eagle
27	Charlie	27	Anchor

הפעלנו פקודת desc עבור armyUnit

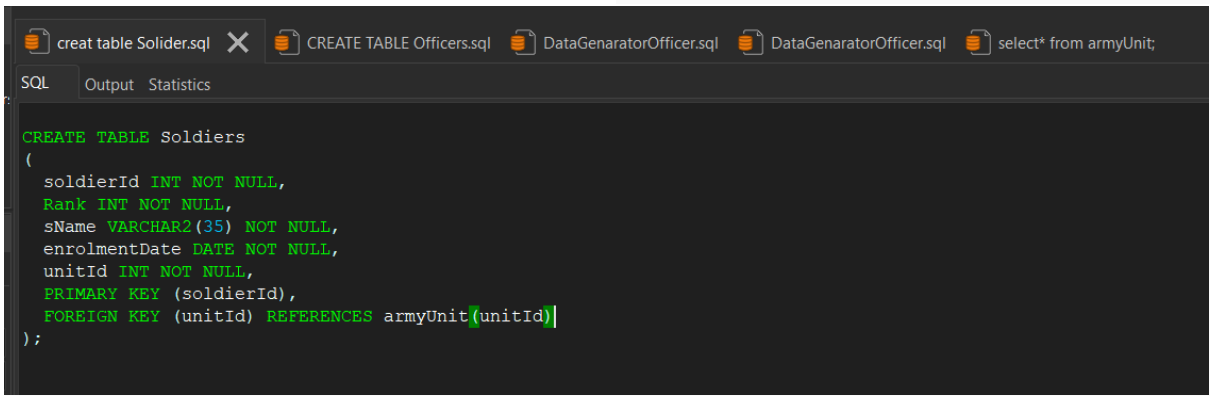
```
SQL> Desc armyUnit
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 UNAME                                     NOT NULL VARCHAR2(35)
 UNITID                                    NOT NULL NUMBER(38)
 SYMBOL                                    NOT NULL VARCHAR2(100)
```

מפה נכניס את כל שאר הטבלאות עם פייתון:

עבוד Soldiers



הנה ה create table וההכנסת נתונים:



creat table Solider.sql | CREATE TABLE Officers.sql | DataGenaratorOfficer.sql | DataGenaratorOfficer.sql | select* from armyUnit; | Soldiers from python.sql

SQL | Output | Statistics

```
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (1, 10, 'Daniel', TO_DATE('2016-10-17', 'YYYY-MM-DD'), 865);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (2, 1, 'Sarah', TO_DATE('2016-04-10', 'YYYY-MM-DD'), 256);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (3, 6, 'Laura', TO_DATE('2016-01-24', 'YYYY-MM-DD'), 985);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (4, 3, 'David', TO_DATE('2022-09-17', 'YYYY-MM-DD'), 785);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (5, 2, 'Laura', TO_DATE('2015-04-16', 'YYYY-MM-DD'), 19);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (6, 2, 'John', TO_DATE('2018-06-02', 'YYYY-MM-DD'), 697);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (7, 10, 'Jessica', TO_DATE('2018-12-27', 'YYYY-MM-DD'), 112);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (8, 4, 'Emily', TO_DATE('2020-04-08', 'YYYY-MM-DD'), 563);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (9, 4, 'David', TO_DATE('2017-04-18', 'YYYY-MM-DD'), 328);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (10, 1, 'Chris', TO_DATE('2015-03-10', 'YYYY-MM-DD'), 885);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (11, 9, 'David', TO_DATE('2018-08-15', 'YYYY-MM-DD'), 610);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (12, 10, 'Alex', TO_DATE('2020-05-17', 'YYYY-MM-DD'), 208);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (13, 3, 'John', TO_DATE('2024-03-09', 'YYYY-MM-DD'), 353);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (14, 1, 'Laura', TO_DATE('2015-12-29', 'YYYY-MM-DD'), 107);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (15, 6, 'Daniel', TO_DATE('2014-07-21', 'YYYY-MM-DD'), 932);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (16, 8, 'Daniel', TO_DATE('2022-01-02', 'YYYY-MM-DD'), 137);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (17, 7, 'Daniel', TO_DATE('2022-03-26', 'YYYY-MM-DD'), 639);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (18, 9, 'Michael', TO_DATE('2022-10-20', 'YYYY-MM-DD'), 798);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (19, 4, 'Jessica', TO_DATE('2023-10-15', 'YYYY-MM-DD'), 659);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (20, 8, 'Laura', TO_DATE('2015-11-11', 'YYYY-MM-DD'), 939);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (21, 9, 'David', TO_DATE('2020-03-08', 'YYYY-MM-DD'), 418);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (22, 1, 'Sarah', TO_DATE('2024-05-16', 'YYYY-MM-DD'), 45);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (23, 9, 'John', TO_DATE('2017-04-26', 'YYYY-MM-DD'), 572);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (24, 2, 'Alex', TO_DATE('2023-12-29', 'YYYY-MM-DD'), 696);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (25, 5, 'Alex', TO_DATE('2016-04-20', 'YYYY-MM-DD'), 559);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (26, 2, 'Emily', TO_DATE('2015-03-02', 'YYYY-MM-DD'), 292);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (27, 7, 'Emily', TO_DATE('2020-10-09', 'YYYY-MM-DD'), 845);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (28, 7, 'John', TO_DATE('2021-02-08', 'YYYY-MM-DD'), 999);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (29, 5, 'Emily', TO_DATE('2014-09-06', 'YYYY-MM-DD'), 31);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (30, 7, 'David', TO_DATE('2024-03-25', 'YYYY-MM-DD'), 655);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (31, 7, 'Daniel', TO_DATE('2023-12-07', 'YYYY-MM-DD'), 40);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (32, 8, 'Daniel', TO_DATE('2016-11-21', 'YYYY-MM-DD'), 318);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (33, 1, 'David', TO_DATE('2023-06-20', 'YYYY-MM-DD'), 452);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (34, 6, 'Laura', TO_DATE('2015-10-23', 'YYYY-MM-DD'), 553);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (35, 2, 'David', TO_DATE('2023-07-30', 'YYYY-MM-DD'), 615);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (36, 10, 'Sarah', TO_DATE('2020-10-05', 'YYYY-MM-DD'), 102);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (37, 9, 'Daniel', TO_DATE('2015-04-27', 'YYYY-MM-DD'), 27);
```

והנה הכל נכנס בצורה תקינה:

creat table Solider.sql | CREATE TABLE Officers.sql | DataGenaratorOfficer.sql | DataGenaratorOfficer.sql | select* from armyUnit; | Soldiers from python.sql | select* from Soldiers;

SQL | Output | Statistics

```
select*
from Soldiers;
```

	SOLDIERID	RANK	SNAME	ENROLMENTDATE	UNITID
1	1	10	Daniel	17/10/2016	865
2	2	1	Sarah	10/04/2016	256
3	3	6	Laura	24/01/2016	985
4	4	3	David	17/09/2022	785
5	5	2	Laura	16/04/2015	19
6	6	2	John	02/06/2018	697
7	7	10	Jessica	27/12/2018	112
8	8	4	Emily	08/04/2020	563
9	9	4	David	18/04/2017	328
10	10	1	Chris	10/03/2015	885
11	11	9	David	15/08/2018	610
12	12	10	Alex	17/05/2020	208
13	13	3	John	09/03/2024	353
14	14	1	Laura	29/12/2015	107
15	15	6	Daniel	21/07/2014	932
16	16	8	Daniel	02/01/2022	137
17	17	7	Daniel	26/03/2022	639
18	18	9	Michael	20/10/2022	798
19	19	4	Jessica	15/10/2023	659
20	20	8	Laura	11/11/2015	939
21	21	9	David	08/03/2020	418
22	22	1	Sarah	16/05/2024	45
23	23	9	John	26/04/2017	572
24	24	2	Alex	29/12/2023	696
25	25	5	Alex	20/04/2016	559
26	26	2	Emily	02/03/2015	292
27	27	7	Emily	09/10/2020	845

הפעלה של desc על Soldiers

```
SQL> desc Soldiers
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SOLDIERID                                 NOT NULL NUMBER(38)
 RANK                                      NOT NULL NUMBER(38)
 SNAME                                     NOT NULL VARCHAR2(35)
 ENROLMENTDATE                             NOT NULL DATE
 UNITID                                    NOT NULL NUMBER(38)
```

עבור Suppliers

suppliers.py M | Suppliers from python.sql U

Python code > suppliers.py > ...

You, 1 second ago | 1 author (You)

```python
import random

# Define some example supplier names
supplier_names = [
    "Supplier A", "Supplier B", "Supplier C", "Supplier D", "Supplier E",
    "Supplier F", "Supplier G", "Supplier H", "Supplier I", "Supplier J"
]

# Generate SQL INSERT statements for Suppliers table
insert_statements = []
for i in range(1, 1001):
    supplier_id = i  # Assuming supplierId starts from 1 and increments by 1
    name = random.choice(supplier_names)
    seniority = random.randint(1, 30)  # Random seniority between 1 and 30 years

    insert_statement = f"INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES ({supplier_id}, '{name}', {seniority})"
    insert_statements.append(insert_statement)

# Print the generated SQL statements
for statement in insert_statements:
    print(statement)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (995, 'Supplier B', 10);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (996, 'Supplier G', 27);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (997, 'Supplier E', 15);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (998, 'Supplier E', 17);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (999, 'Supplier D', 8);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (1000, 'Supplier F', 20);
```

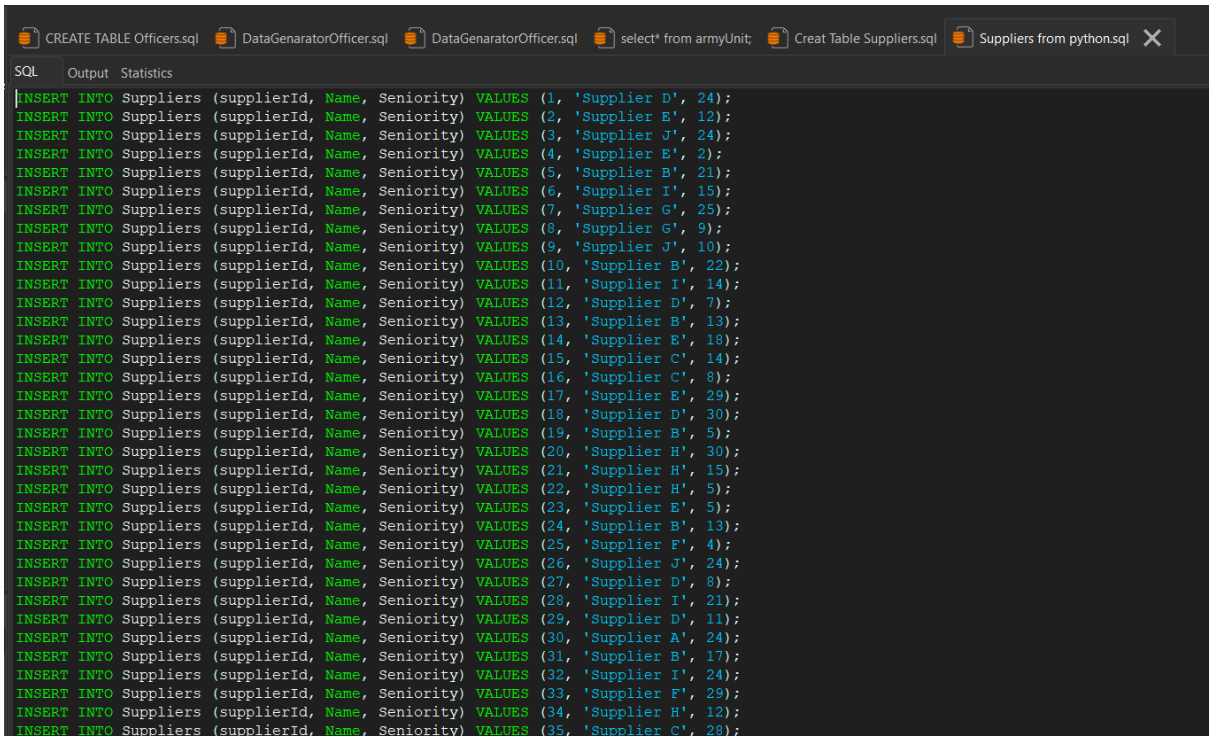
cmd
cmd
Code

CREATE TABLE Officers.sql | DataGenaratorOfficer.sql

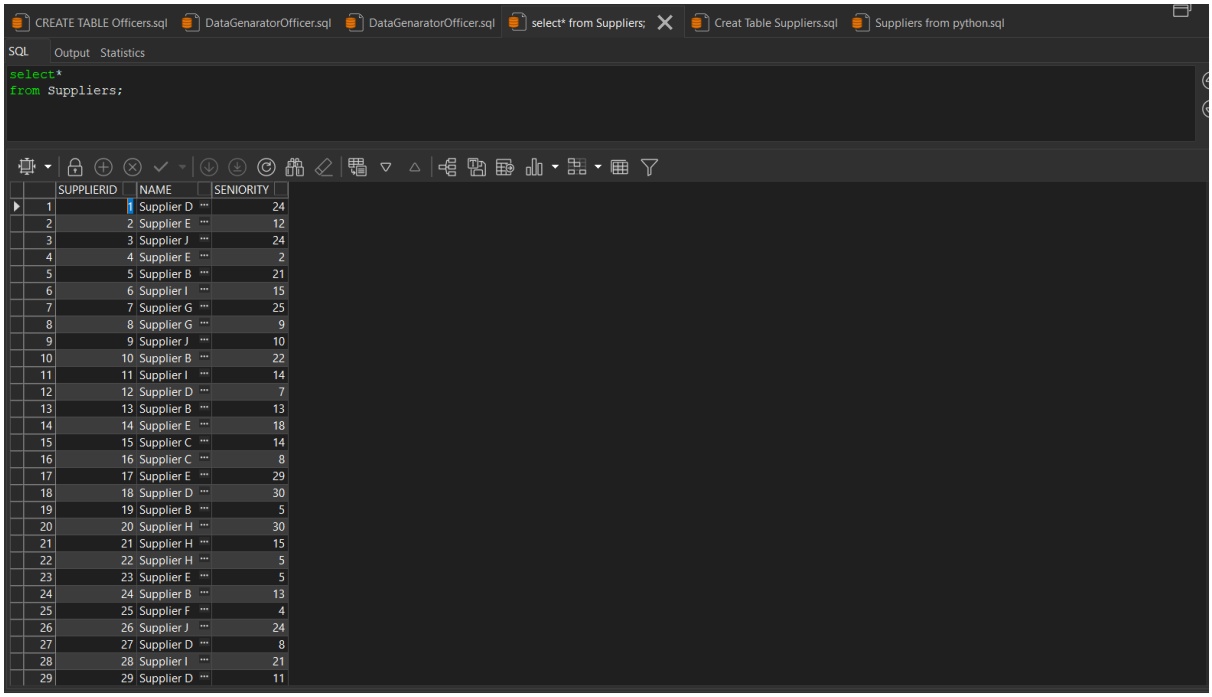
SQL Output Statistics

```
CREATE TABLE Suppliers
(
  supplierId INT NOT NULL,
  Name VARCHAR2(35) NOT NULL,
  Seniority INT NOT NULL,
  PRIMARY KEY (supplierId)
);
```

הנה הקוד להכנסת האיברים:



וכל הנתונים נכנסו באופן טוב:



פעולת Desc על Suppliers:

```
SQL> Desc Suppliers
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SUPPLIERID                                NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(35)
 SENIORITY                                 NOT NULL NUMBER(38)
```

Equipment עבור

CREATE TABLE Officers.sql | DataGenaratorOfficer.sql | DataGenaratorOfficer.sql | select* from Suppliers; | CREATE TABLE Equipment.sql

SQL | Output | Statistics

```sql
CREATE TABLE Equipment
(
  eqtId INT NOT NULL,
  Name VARCHAR2(35) NOT NULL,
  exportDate DATE NOT NULL,
  eqQuantity INT NOT NULL,
  whId INT NOT NULL,
  PRIMARY KEY (eqtId),
  FOREIGN KEY (whId) REFERENCES Warehouse(whId)
);
```

Equipment from python.sql U | Equipment.py U

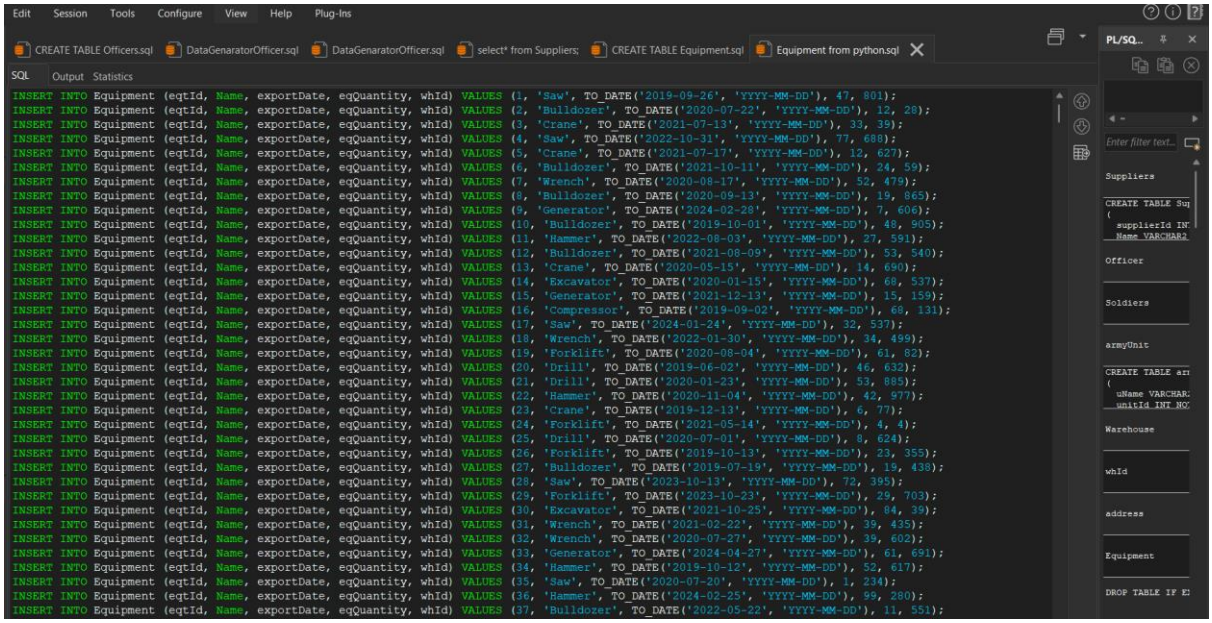
Python code > Equipment.py > ...

```python
# Generate a random date within the last 5 years
def random_date(start, end):
    return start + timedelta(
        seconds=random.randint(0, int((end - start).total_seconds()))
    )

# Generate SQL INSERT statements for Equipment table
insert_statements = []
start_date = datetime.now() - timedelta(days=365 * 5)
end_date = datetime.now()

for i in range(1, 1001):
    eqt_id = i  # Assuming eqtId starts from 1 and increments by 1
    name = random.choice(equipment_names)
    export_date = random_date(start_date, end_date).strftime('%Y-%m-%d')
    eq_quantity = random.randint(1, 100)  # Random quantity between 1 and 100
    wh_id = random.randint(1, 1000)  # Assuming whId ranges from 1 to 1000

    insert_statement = f"INSERT INTO Equipment (eqtId, Name, exportDate, eqQuantity, whId) VALUES ({eqt_id}, '{name}', TO
    insert_statements.append(insert_statement)

# Print the generated SQL statements
for statement in insert_statements:
    print(statement)
```

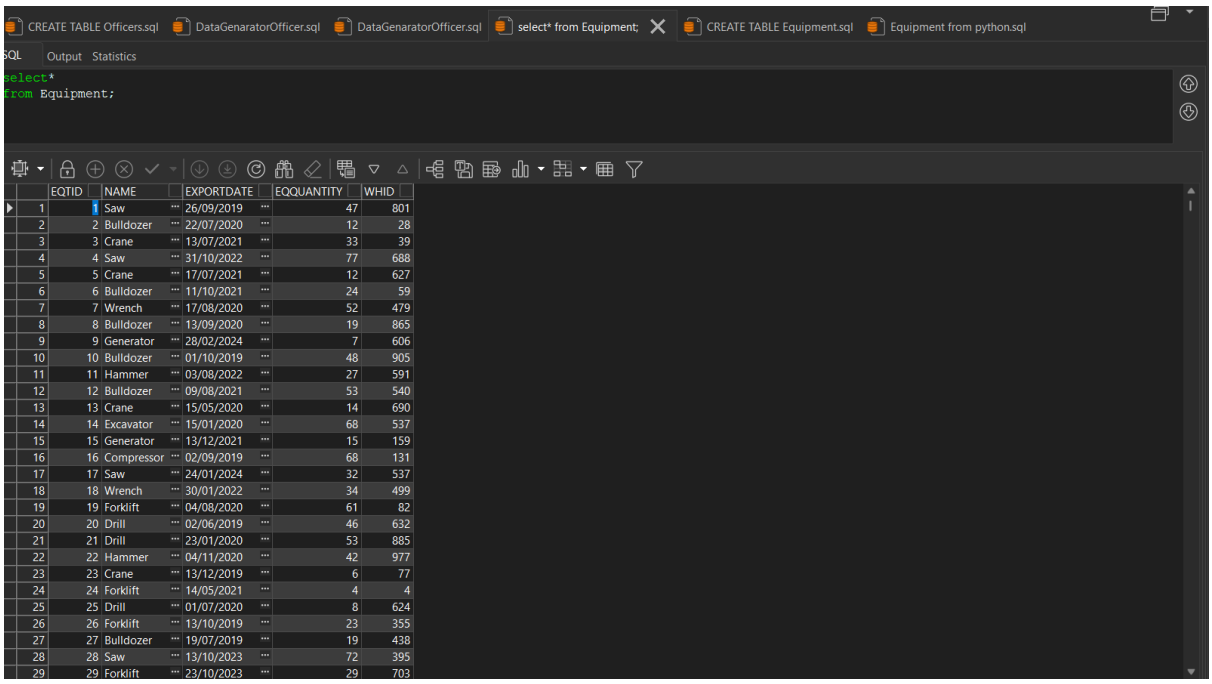
PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | GITLENS

```
INSERT INTO Equipment (eqtId, Name, exportDate, eqQuantity, whId) VALUES (998, 'Hammer', TO_DATE('2022-03-29', '
YYYY-MM-DD'), 24, 431);
INSERT INTO Equipment (eqtId, Name, exportDate, eqQuantity, whId) VALUES (999, 'Compressor', TO_DATE('2019-08-23
', 'YYYY-MM-DD'), 66, 203);
INSERT INTO Equipment (eqtId, Name, exportDate, eqQuantity, whId) VALUES (1000, 'Forklift', TO_DATE('2020-11-15'
, 'YYYY-MM-DD'), 79, 537);
```

cmd
cmd
Code



ועכשיו כל הנתונים נכנסו כמו שצריך:

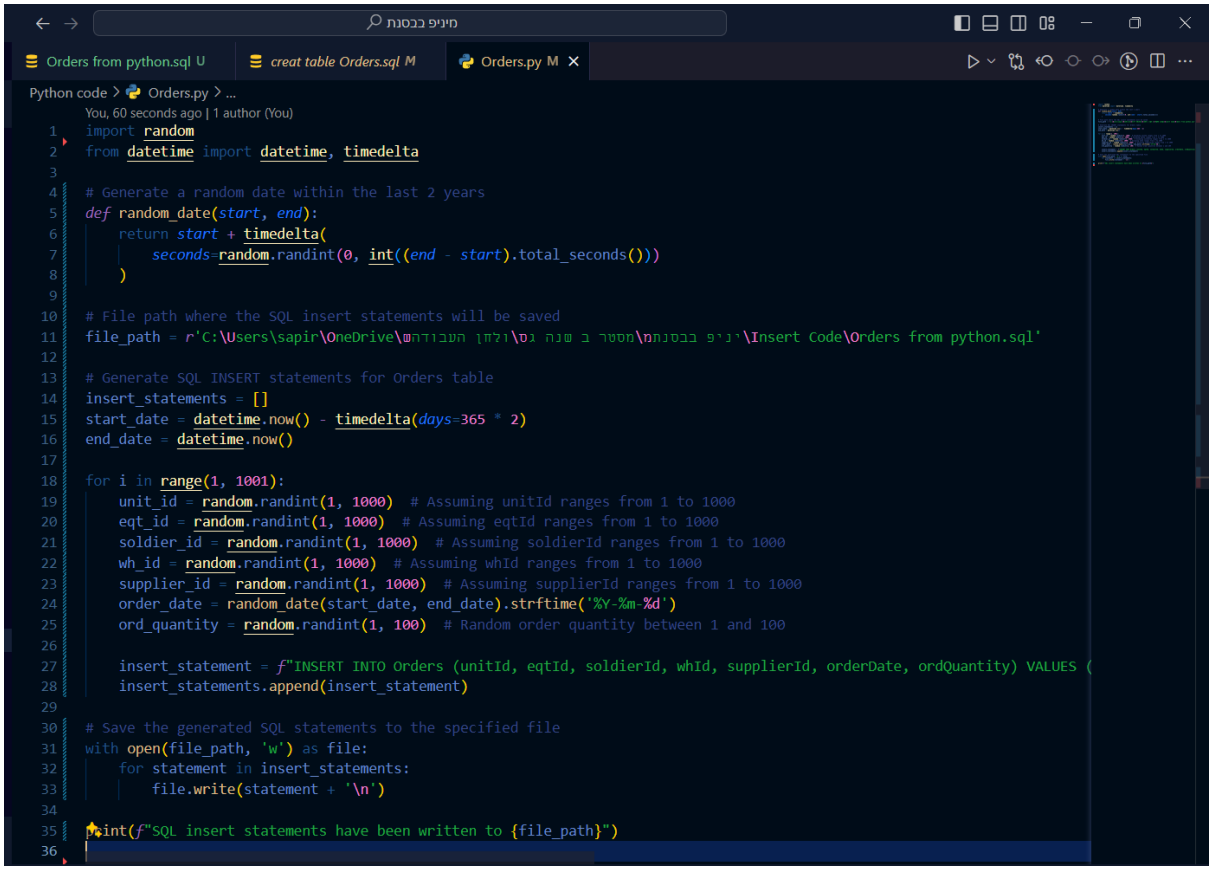


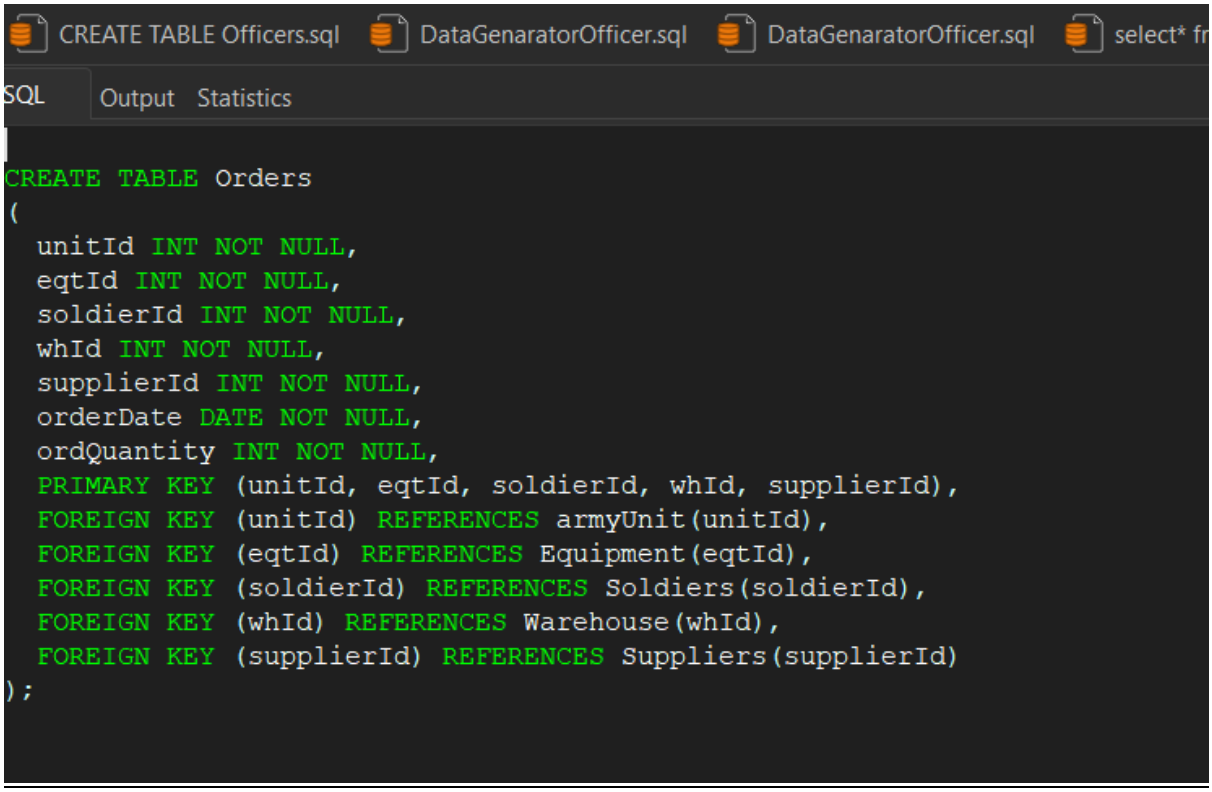
	EQTID	NAME	EXPORTDATE	EQQUANTITY	WHID
1	1	Saw	26/09/2019	47	801
2	2	Bulldozer	22/07/2020	12	28
3	3	Crane	13/07/2021	33	39
4	4	Saw	31/10/2022	77	688
5	5	Crane	17/07/2021	12	627
6	6	Bulldozer	11/10/2021	24	59
7	7	Wrench	17/08/2020	52	479
8	8	Bulldozer	13/09/2020	19	865
9	9	Generator	28/02/2024	7	606
10	10	Bulldozer	01/10/2019	48	905
11	11	Hammer	03/08/2022	27	591
12	12	Bulldozer	09/08/2021	53	540
13	13	Crane	15/05/2020	14	690
14	14	Excavator	15/01/2020	68	537
15	15	Generator	13/12/2021	15	159
16	16	Compressor	02/09/2019	68	131
17	17	Saw	24/01/2024	32	537
18	18	Wrench	30/01/2022	34	499
19	19	Forklift	04/08/2020	61	82
20	20	Drill	02/06/2019	46	632
21	21	Drill	23/01/2020	53	885
22	22	Hammer	04/11/2020	42	977
23	23	Crane	13/12/2019	6	77
24	24	Forklift	14/05/2021	4	4
25	25	Drill	01/07/2020	8	624
26	26	Forklift	13/10/2019	23	355
27	27	Bulldozer	19/07/2019	19	438
28	28	Saw	13/10/2023	72	395
29	29	Forklift	23/10/2023	29	703

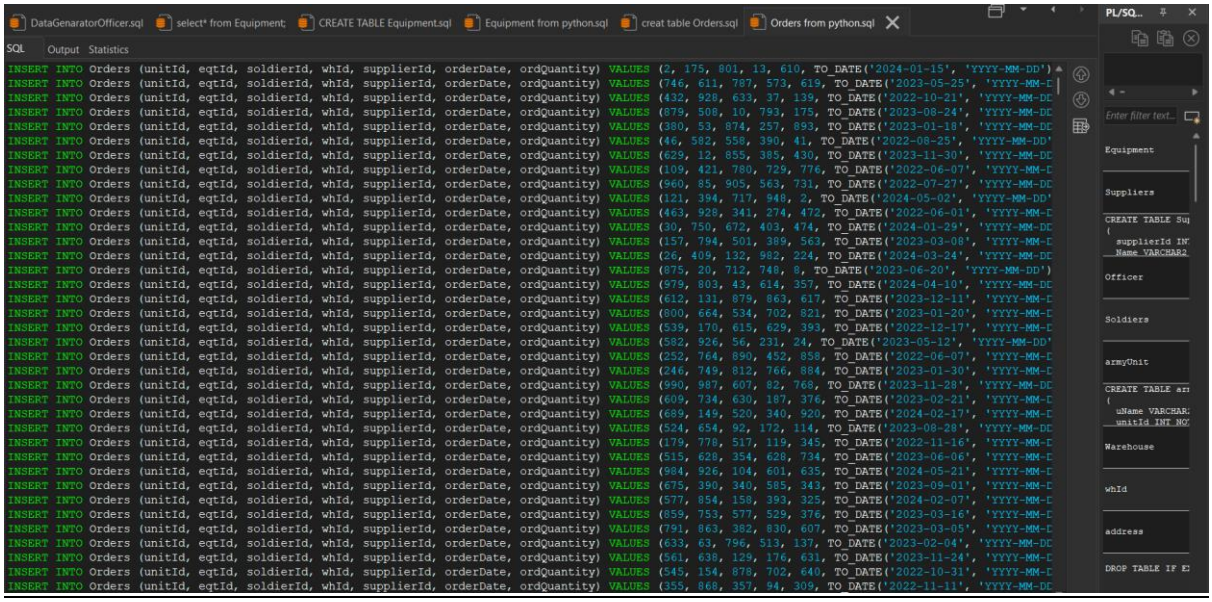
ופעולת desc עבוד Equipment:

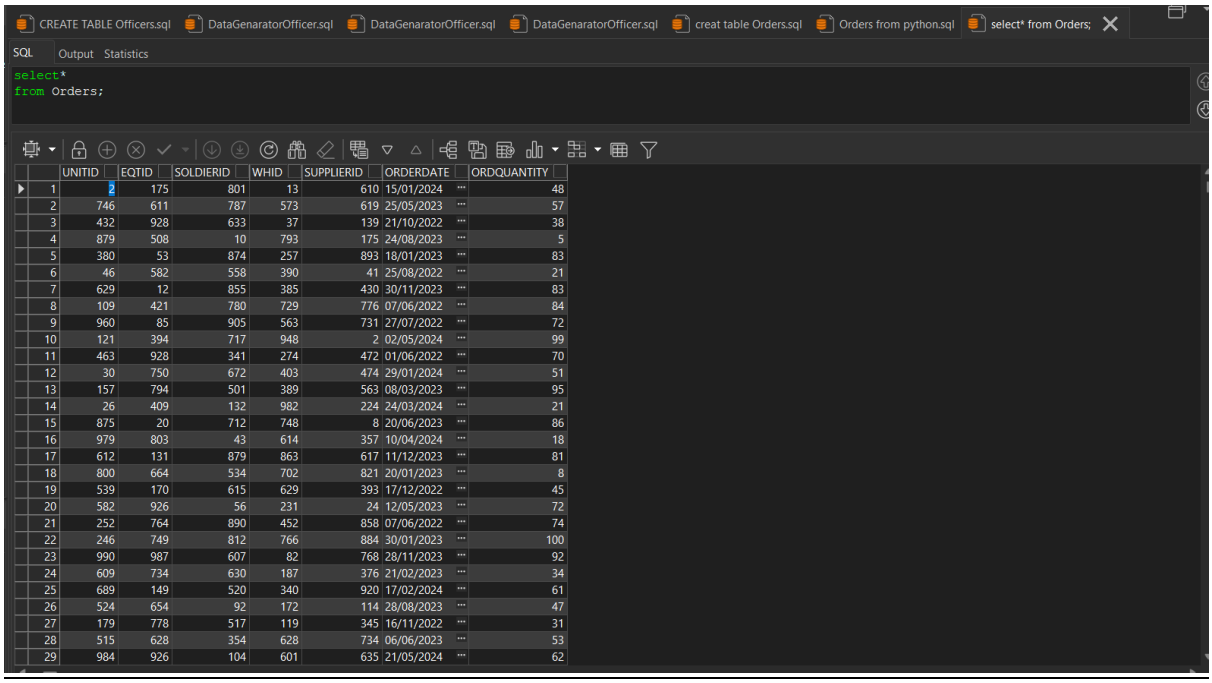
```
SQL> Desc Equipment
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EQTID                                     NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(35)
 EXPORTDATE                                NOT NULL DATE
 EQQUANTITY                                NOT NULL NUMBER(38)
 WHID                                      NOT NULL NUMBER(38)
```

עבור Orders


מיניף בבסנת
Orders from python.sql U
creat table Orders.sql M
Orders.py M
Python code > Orders.py > ...
You, 60 seconds ago | 1 author (You)
import random
from datetime import datetime, timedelta
# Generate a random date within the last 2 years
def random_date(start, end):
return start + timedelta(
seconds=random.randint(0, int((end - start).total_seconds()))
)
# File path where the SQL insert statements will be saved
file_path = r'C:\Users\sapir\OneDrive\מסטר ב שנה גמ\ולחן העבודהש\יניף בבסנתמ\Insert Code\Orders from python.sql'
# Generate SQL INSERT statements for Orders table
insert_statements = []
start_date = datetime.now() - timedelta(days=365 * 2)
end_date = datetime.now()
for i in range(1, 1001):
unit_id = random.randint(1, 1000) # Assuming unitId ranges from 1 to 1000
eqt_id = random.randint(1, 1000) # Assuming eqtId ranges from 1 to 1000
soldier_id = random.randint(1, 1000) # Assuming soldierId ranges from 1 to 1000
wh_id = random.randint(1, 1000) # Assuming whId ranges from 1 to 1000
supplier_id = random.randint(1, 1000) # Assuming supplierId ranges from 1 to 1000
order_date = random_date(start_date, end_date).strftime('%Y-%m-%d')
ord_quantity = random.randint(1, 100) # Random order quantity between 1 and 100
insert_statement = f"INSERT INTO Orders (unitId, eqtId, soldierId, whId, supplierId, orderDate, ordQuantity) VALUES (
insert_statements.append(insert_statement)
# Save the generated SQL statements to the specified file
with open(file_path, 'w') as file:
for statement in insert_statements:
file.write(statement + '\n')
print(f"SQL insert statements have been written to {file_path}")


CREATE TABLE Officers.sql
DataGenaratorOfficer.sql
DataGenaratorOfficer.sql
select* fr
SQL
Output
Statistics
CREATE TABLE Orders
(
unitId INT NOT NULL,
eqtId INT NOT NULL,
soldierId INT NOT NULL,
whId INT NOT NULL,
supplierId INT NOT NULL,
orderDate DATE NOT NULL,
ordQuantity INT NOT NULL,
PRIMARY KEY (unitId, eqtId, soldierId, whId, supplierId),
FOREIGN KEY (unitId) REFERENCES armyUnit(unitId),
FOREIGN KEY (eqtId) REFERENCES Equipment(eqtId),
FOREIGN KEY (soldierId) REFERENCES Soldiers(soldierId),
FOREIGN KEY (whId) REFERENCES Warehouse(whId),
FOREIGN KEY (supplierId) REFERENCES Suppliers(supplierId)
);





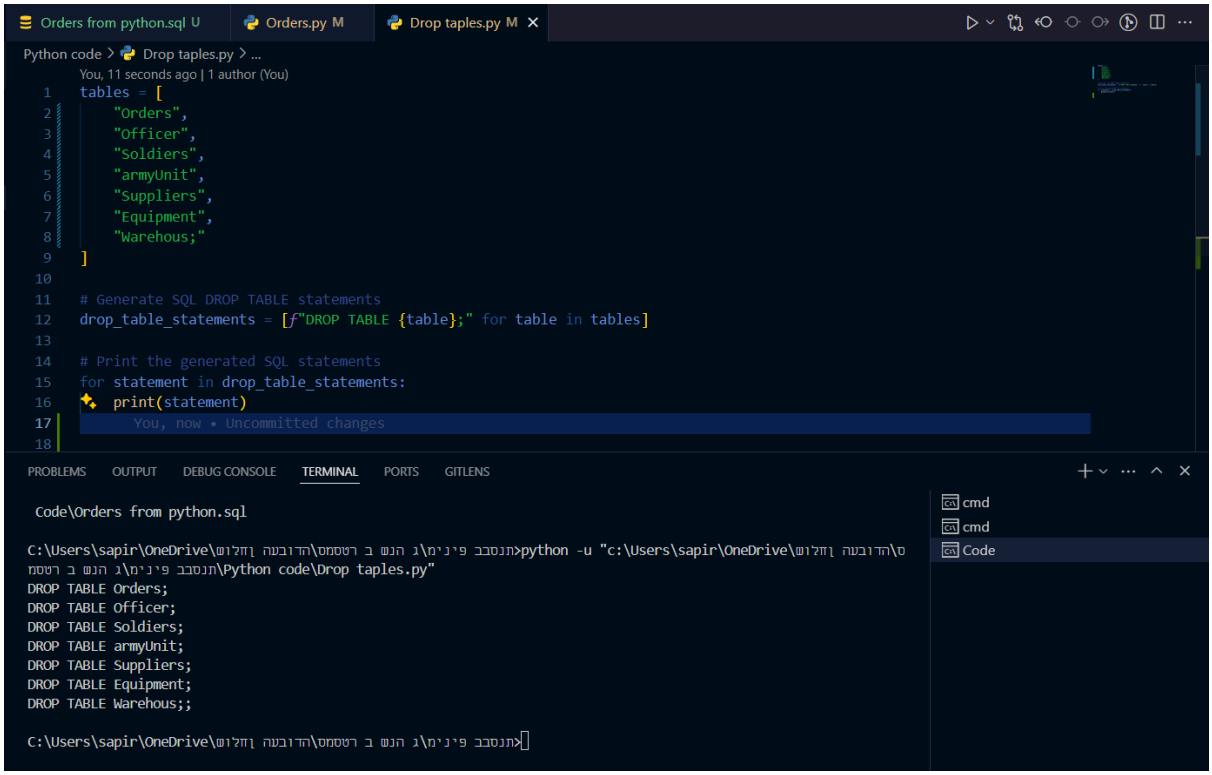
	UNITID	EQTID	SOLDIERID	WHID	SUPPLIERID	ORDERDATE	ORDQUANTITY
1	2	175	801	13	610	15/01/2024	48
2	746	611	787	573	619	25/05/2023	57
3	432	928	633	37	139	21/10/2022	38
4	879	508	10	793	175	24/08/2023	5
5	380	53	874	257	893	18/01/2023	83
6	46	582	558	390	41	25/08/2022	21
7	629	12	855	385	430	30/11/2023	83
8	109	421	780	729	776	07/06/2022	84
9	960	85	905	563	731	27/07/2022	72
10	121	394	717	948	2	02/05/2024	99
11	463	928	341	274	472	01/06/2022	70
12	30	750	672	403	474	29/01/2024	51
13	157	794	501	389	563	08/03/2023	95
14	26	409	132	982	224	24/03/2024	21
15	875	20	712	748	8	20/06/2023	86
16	979	803	43	614	357	10/04/2024	18
17	612	131	879	863	617	11/12/2023	81
18	800	664	534	702	821	20/01/2023	8
19	539	170	615	629	393	17/12/2022	45
20	582	926	56	231	24	12/05/2023	72
21	252	764	890	452	858	07/06/2022	74
22	246	749	812	766	884	30/01/2023	100
23	990	987	607	82	768	28/11/2023	92
24	609	734	630	187	376	21/02/2023	34
25	689	149	520	340	920	17/02/2024	61
26	524	654	92	172	114	28/08/2023	47
27	179	778	517	119	345	16/11/2022	31
28	515	628	354	628	734	06/06/2023	53
29	984	926	104	601	635	21/05/2024	62

הרצת פעולת desc על Orders:

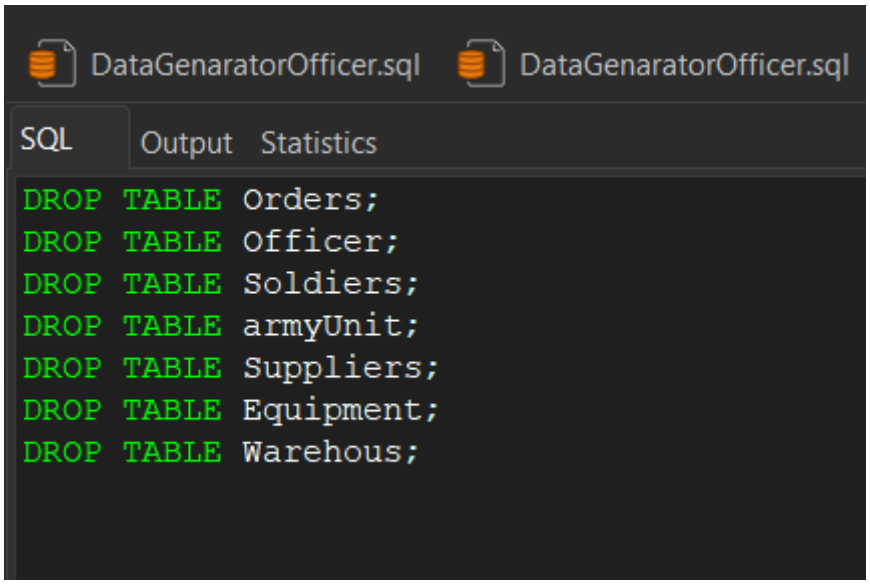
```
SQL> Desc Orders
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 UNITID                                    NOT NULL NUMBER(38)
 EQTID                                     NOT NULL NUMBER(38)
 SOLDIERID                                 NOT NULL NUMBER(38)
 WHID                                      NOT NULL NUMBER(38)
 SUPPLIERID                                NOT NULL NUMBER(38)
 ORDERDATE                                 NOT NULL DATE
 ORDQUANTITY                               NOT NULL NUMBER(38)
```

Drop tables:

כתבנו קוד בפייתון שמייצר את פקודות ה drop tables



וכך יצאו לנו הפקודות עצמן:



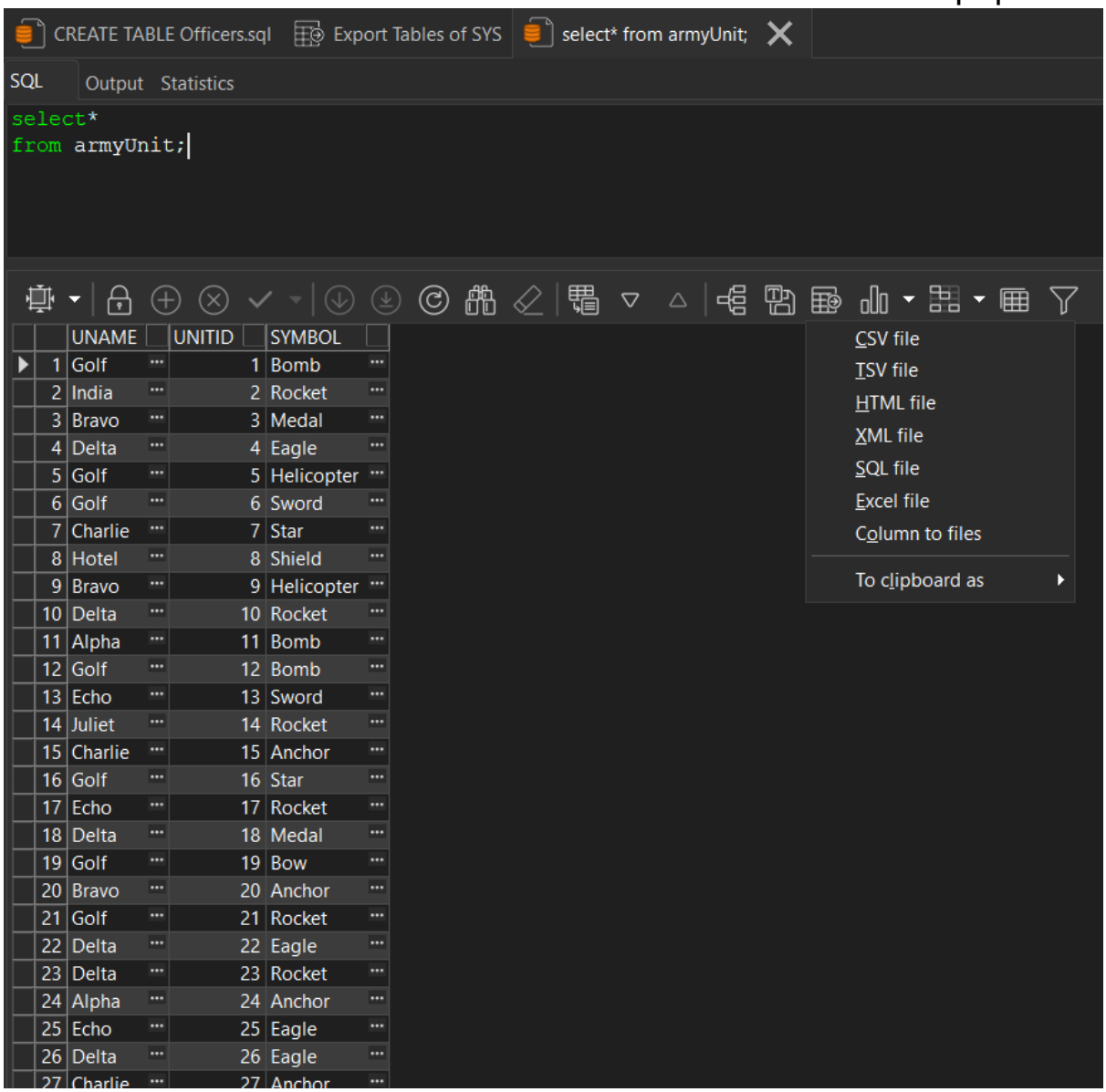
גיבוי ושחזור:

גיבוי

ע"מ לגבות את הקוד - עבור כל טבלה ניצור שאילתה המחזירה את כל הטבלה.
לאחר מכן, בקטע שבו הודפסה הטבלה – נלחץ על האפשרות: ניכנס ל-export table
sql file> ונבחר את המקום בו נרצה שישמר הקובץ. בכך יצרנו גיבוי של כל טבלה,
שנשמר ע"י insertים.

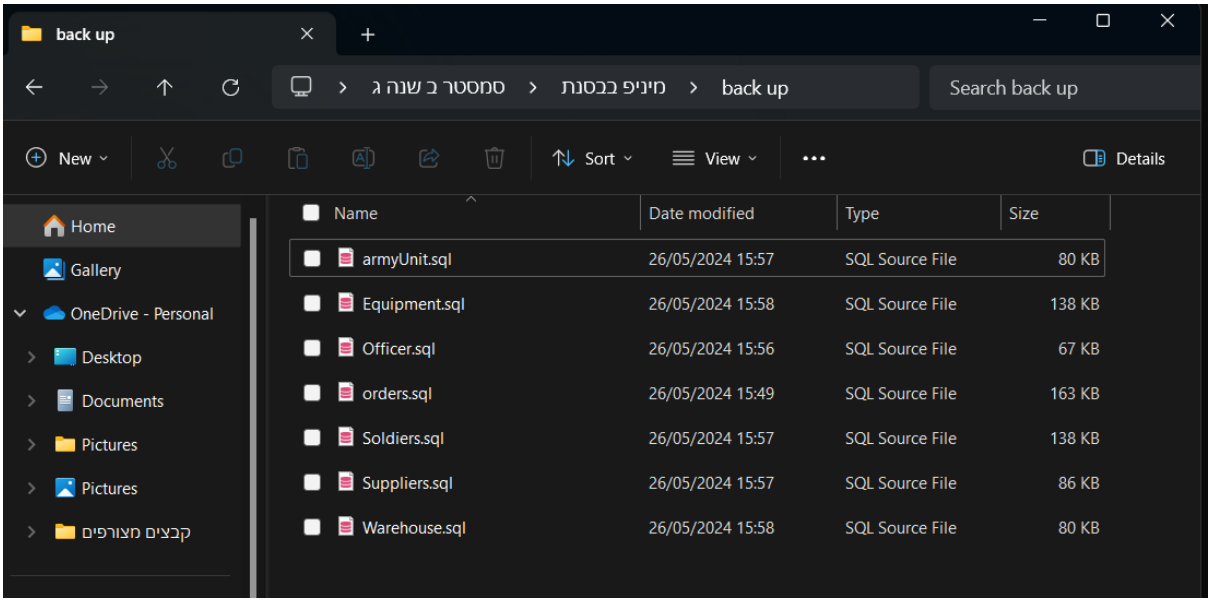
לדוגמה – עבור הטבלה armyUnit:

יצוא הקובץ כמתואר:



	UNAME	UNITID	SYMBOL
1	Golf	1	Bomb
2	India	2	Rocket
3	Bravo	3	Medal
4	Delta	4	Eagle
5	Golf	5	Helicopter
6	Golf	6	Sword
7	Charlie	7	Star
8	Hotel	8	Shield
9	Bravo	9	Helicopter
10	Delta	10	Rocket
11	Alpha	11	Bomb
12	Golf	12	Bomb
13	Echo	13	Sword
14	Juliet	14	Rocket
15	Charlie	15	Anchor
16	Golf	16	Star
17	Echo	17	Rocket
18	Delta	18	Medal
19	Golf	19	Bow
20	Bravo	20	Anchor
21	Golf	21	Rocket
22	Delta	22	Eagle
23	Delta	23	Rocket
24	Alpha	24	Anchor
25	Echo	25	Eagle
26	Delta	26	Eagle
27	Charlie	27	Anchor

כעת ניצור גיבויים לכל הטבלאות שלנו:



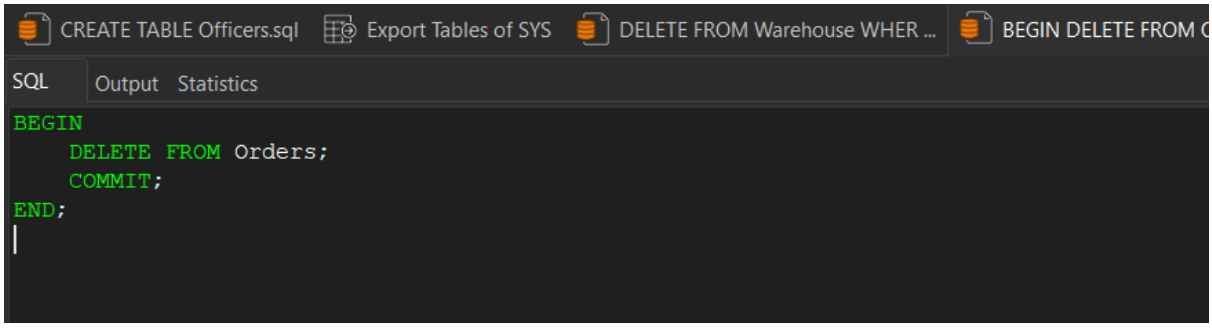
בכך גיבינו את כל טבלאותינו בהצלחה.

שיחזור

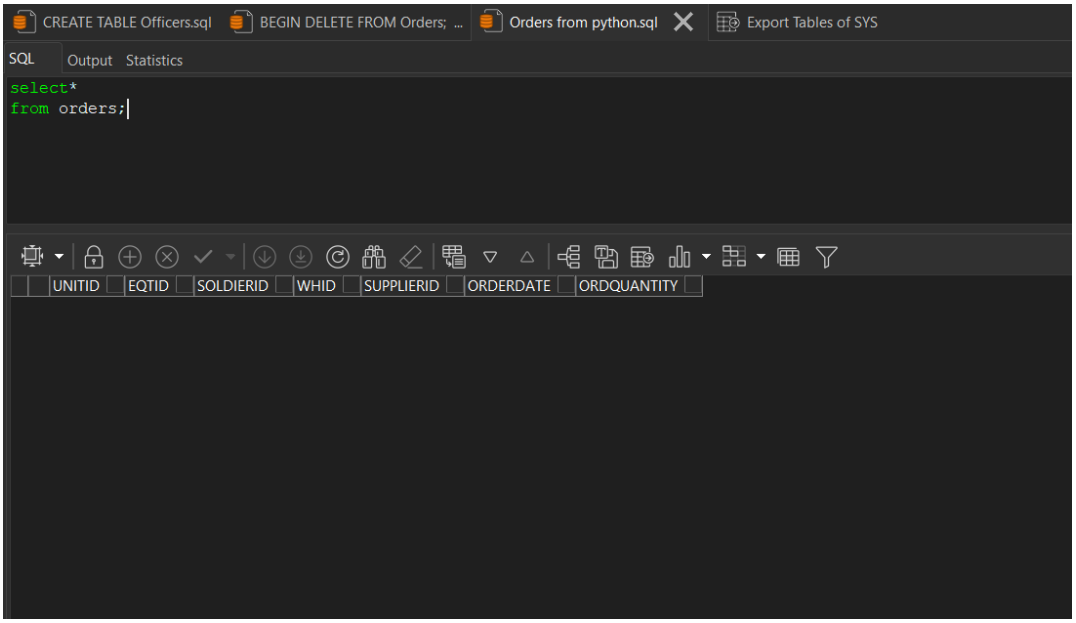
כעת כדי לשחזר את הקובץ – במידה ונצטרך, ניכנס לאופציה import table->sql, נבחר את הקובץ ממנו נרצה לשחזר את הטבלה למטה, ונלחץ import. מיד inserts יופיעו לנו כל הinsertים שיחזירו לנו את הטבלה המקורית. כל שנשאר הוא ללחוץ על execute, ולבדוק האם השיחזור הצליח.

נמחיש את התהליך בדוגמה על הטבלה Orders:

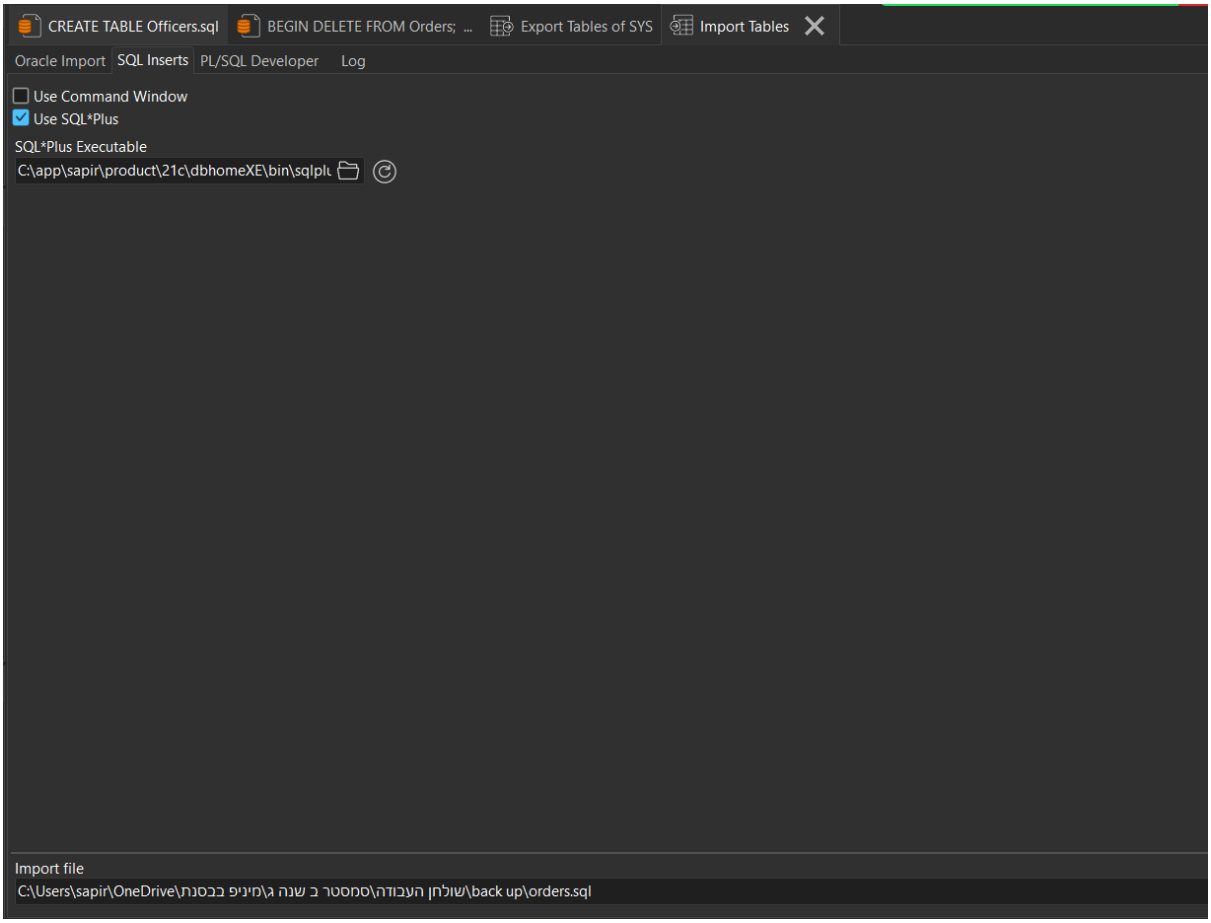
תחילה נראה שיש למחוק לפחות חלק מהטבלה – ע"מ לראות אם השיחזור מצליח.
נמחק את נתוני הטבלה בשאילתה המתוארת בתמונה:



כעת, נראה שהנתונים אכן נמחקו:



עכשיו, נבצע את השיחזור כמפורט למעלה – ניתן לראות זאת בתמונה שלהלן:



כעת, נבדוק שהנתונים אכן שוחזרו:

CREATE TABLE Officers.sql | select* from Orders; | Export Tables of SYS | Import Tables

SQL | Output | Statistics

```
select*
from Orders;
```

	UNITID	EQTID	SOLDIERID	WHID	SUPPLIERID	ORDERDATE	ORDQUANTITY
1	689	149	520	340	920	17/02/2024	61
2	524	654	92	172	114	28/08/2023	47
3	179	778	517	119	345	16/11/2022	31
4	515	628	354	628	734	06/06/2023	53
5	984	926	104	601	635	21/05/2024	62
6	675	390	340	585	343	01/09/2023	74
7	577	854	158	393	325	07/02/2024	1
8	859	753	577	529	376	16/03/2023	50
9	791	863	382	830	607	05/03/2023	12
10	633	63	796	513	137	04/02/2023	4
11	561	638	129	176	631	24/11/2023	95
12	545	154	878	702	640	31/10/2022	37
13	355	868	357	94	309	11/11/2022	99
14	650	71	314	492	641	30/09/2022	56
15	166	631	818	761	35	12/07/2023	18
16	459	6	299	542	428	19/12/2023	48
17	599	340	797	877	951	27/07/2022	72
18	171	105	425	646	128	22/06/2023	44
19	938	355	69	374	805	04/06/2023	11
20	182	702	57	757	1000	24/07/2023	96
21	767	503	31	272	330	10/10/2022	50
22	11	825	699	405	133	27/08/2023	64
23	421	339	447	740	496	20/12/2022	22
24	584	260	802	425	31	02/01/2023	26

'

הצלחנו!