

מיני פרויקט בבסיסי נתונים:

# אפסנאות

מגישים:

דניאל צירקין – 214389751

ספיר בשן – 214103368

קישור לGitHub - [https://github.com/Mini\\_Project\\_DB](https://github.com/Mini_Project_DB)

## תוכן עניינים

2	תוכן עניינים
3	תיאור הארגון: ERD
4	דיאגרמת DSD
4	CREATE TABLE פקודות:
8	נירמול ל 3NF:
8	- מחסנים: (WAREHOUSES)
9	ציוד: (EQUIPMENTS)
10	ספקים: (SUPPLIERS)
10	חיילים: (SOLDIERS)
11	יחידות: (UNITS)
11	קצינים: (OFFICERS)
12	הזמנות: (ORDERS)
13	אכלוס טבלאות:
13	דרך א - MOCKAROO:
14	עבור WAREHOUSE
15	דרך ב - DATA GENERATOR:
16	עבור OFFICERS
18	דרך ג - PYTHON INSERT:
18	עבור ARMYUNIT
20	עבור SOLDIERS
22	עבור SUPPLIERS
24	עבור EQUIPMENT
26	עבור ORDERS
28	:DROP TABLES
29	גיבוי ושחזור:
29	גיבוי
30	שיחזור

## **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.

## **כעת, בוא נבחן יישויות עיקריות במערכת:**

### **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)**

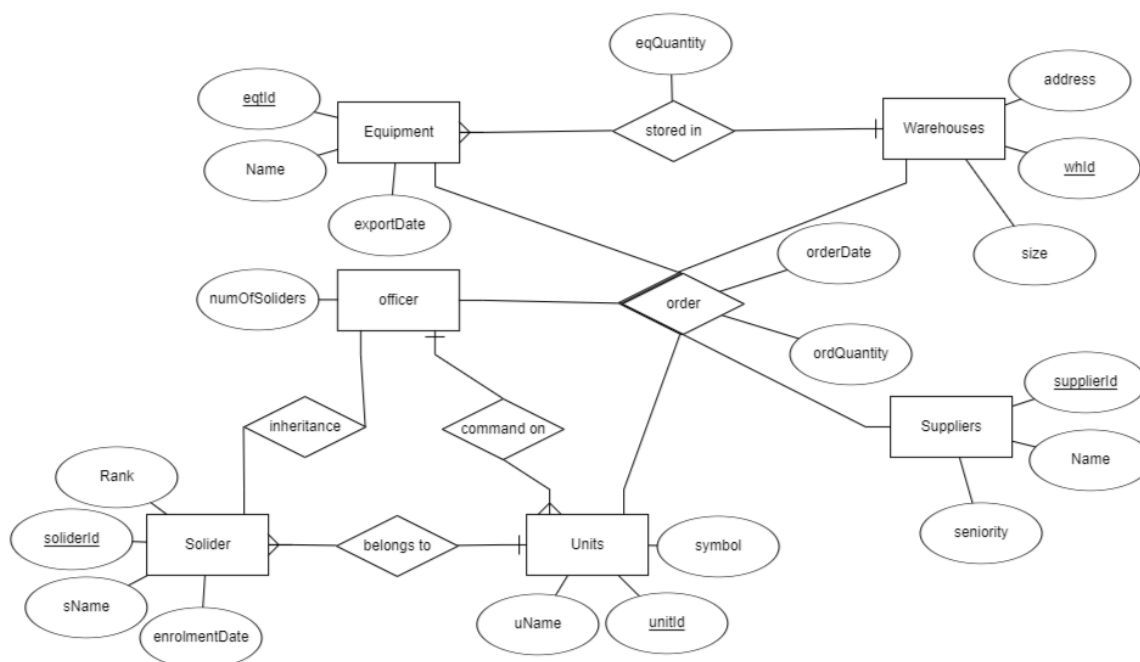
מייצגים את הספקים השונים שמספקים ציוד לצבא.

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

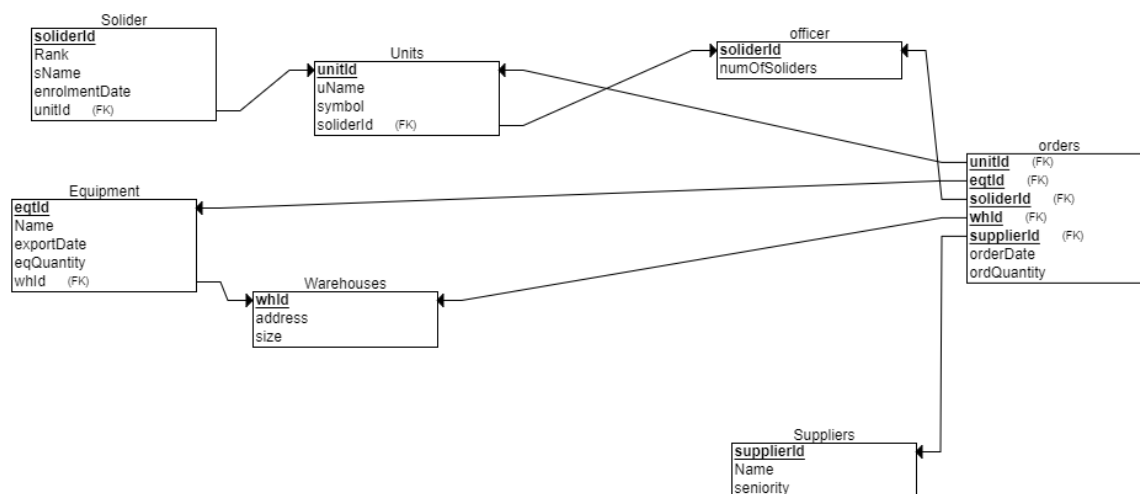
## **הזמנת ציוד עבור יחידה:**

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:



## create table פקודות:

```
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)
);
```

## Third Normal Form (3NF)

R is in **Third Normal Form (3NF)** if for every  $X \rightarrow Y$  implied by F, one of the following holds

- X is a superkey of R or
- for each  $A \in Y$ , either  $A \in X$  or A is an attribute in a key



R is in **Third Normal Form (3NF)** if for every  $X \rightarrow Y$  in F, one of the following holds

- X is a superkey of R or
- for each  $A \in Y$ , either  $A \in X$  or A is an attribute in a key

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  $X \rightarrow Y$  בקבוצת התלות הפונקציונלית F, אחד מהבאים חייב להתקיים:

X - הוא מפתח על של (R הטבלה).

- לכל תכונה A ב A, Y - נמצא ב X - או A היא תכונה במפתח.

בואו נבחן כל טבלה בסכימה שלכם ונחיל את האלגוריתם הזה:

(Warehouses) - מחסנים:

CREATE TABLE Warehouse



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

עפ"י הסכמה:

- whld -> address, size

כאן whld, הוא מפתח על (מפתח ראשי).

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

(Equipments) ציוד:

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

עפ"י הסכמה:

- eqtld -> Name, exportDate, eqQuantity, whld

כאן eqtld, הוא מפתח על (מפתח ראשי).

eqtld - 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).

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

## 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

### (Orders) הזמנות:

```
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) NF. בצורה נכונה ל-3.

אכלוס טבלאות:

הכנסנו את הנתונים בשלושה דרכים:

דרך א - mockaroo:

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

## עבור Warehouse

Need more data? Plans start at just \$60/year. Mockaroo is also available as a docker image that you can deploy in your own private cloud.

Field Name	Type	Options
address	Address Line 2	blank: 0 % $\Sigma$ $\times$
whId	Row Number	blank: 0 % $\Sigma$ $\times$
WhSize	Number	min: 0 max: 1000 decimals: 0 blank: 0 % $\Sigma$ $\times$

+ ADD ANOTHER FIELD GENERATE FIELDS USING ALL...

# Rows: 1000 Format: SQL Table Name: Warehouse ☐ Include CREATE TABLE

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

```
select* from Soldiers; first.sql CREATE TABLE Warehouses.sql SQL Window
```

```
SQL Output Statistics
```

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

והנה הפקודות שנוצרו.

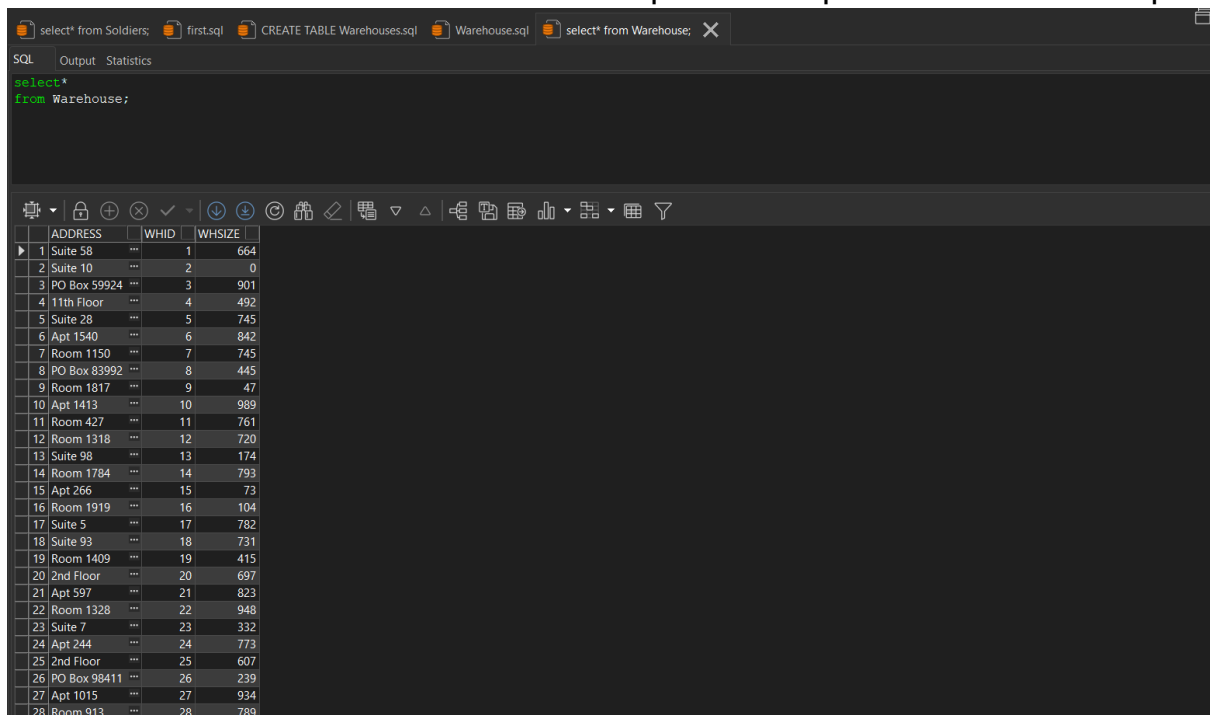
```
SYSDBA@XE as SYSDBA - PL/SQL Developer - Warehouse.sql
```

```
File Project Edit Session Tools Configure View Help Plug-Ins
```

```
SQL Output Statistics
```

```
insert into Warehouse (address, whId, WhSize) values ('Suite 90', 963, 553);
insert into Warehouse (address, whId, WhSize) values ('Room 1352', 964, 825);
insert into Warehouse (address, whId, WhSize) values ('Room 1784', 965, 287);
insert into Warehouse (address, whId, WhSize) values ('PO Box 65122', 966, 399);
insert into Warehouse (address, whId, WhSize) values ('Room 762', 967, 807);
insert into Warehouse (address, whId, WhSize) values ('Room 271', 968, 6);
insert into Warehouse (address, whId, WhSize) values ('15th Floor', 969, 922);
insert into Warehouse (address, whId, WhSize) values ('Apt 60', 970, 408);
insert into Warehouse (address, whId, WhSize) values ('12th Floor', 971, 233);
insert into Warehouse (address, whId, WhSize) values ('Suite 90', 972, 771);
insert into Warehouse (address, whId, WhSize) values ('PO Box 76387', 973, 846);
insert into Warehouse (address, whId, WhSize) values ('PO Box 88567', 974, 249);
insert into Warehouse (address, whId, WhSize) values ('PO Box 69847', 975, 683);
insert into Warehouse (address, whId, WhSize) values ('Suite 50', 976, 130);
insert into Warehouse (address, whId, WhSize) values ('14th Floor', 977, 770);
insert into Warehouse (address, whId, WhSize) values ('Room 1659', 978, 635);
insert into Warehouse (address, whId, WhSize) values ('Room 661', 979, 275);
insert into Warehouse (address, whId, WhSize) values ('Room 264', 980, 79);
insert into Warehouse (address, whId, WhSize) values ('PO Box 52175', 981, 637);
insert into Warehouse (address, whId, WhSize) values ('10th Floor', 982, 59);
insert into Warehouse (address, whId, WhSize) values ('PO Box 80967', 983, 939);
insert into Warehouse (address, whId, WhSize) values ('Apt 1442', 984, 157);
insert into Warehouse (address, whId, WhSize) values ('Room 85', 985, 773);
insert into Warehouse (address, whId, WhSize) values ('Suite 11', 986, 47);
insert into Warehouse (address, whId, WhSize) values ('Suite 53', 987, 803);
insert into Warehouse (address, whId, WhSize) values ('Room 738', 988, 666);
insert into Warehouse (address, whId, WhSize) values ('Suite 21', 989, 984);
insert into Warehouse (address, whId, WhSize) values ('Suite 31', 990, 132);
insert into Warehouse (address, whId, WhSize) values ('Suite 59', 991, 650);
insert into Warehouse (address, whId, WhSize) values ('Suite 40', 992, 258);
insert into Warehouse (address, whId, WhSize) values ('Room 1056', 993, 740);
insert into Warehouse (address, whId, WhSize) values ('Suite 78', 994, 805);
insert into Warehouse (address, whId, WhSize) values ('6th Floor', 995, 605);
insert into Warehouse (address, whId, WhSize) values ('Apt 1084', 996, 834);
insert into Warehouse (address, whId, WhSize) values ('12th Floor', 997, 788);
insert into Warehouse (address, whId, WhSize) values ('12th Floor', 998, 715);
insert into Warehouse (address, whId, WhSize) values ('5th Floor', 999, 344);
insert into Warehouse (address, whId, WhSize) values ('Apt 804', 1000, 527);
```

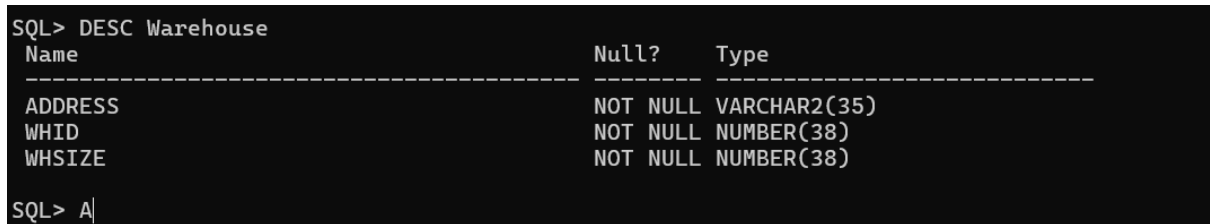
בדקנו שכל הרשומות אכן נכנסו לתוך הטבלה:



The screenshot shows a SQL IDE window with a query editor at the top containing the SQL statement: `select* from Warehouse;`. Below the editor, a table of results is displayed with 28 rows and 4 columns: ADDRESS, WHID, and WHSIZE. The data includes various locations like Suite 58, Suite 10, PO Box 59924, 11th Floor, Suite 28, Apt 1540, Room 1150, PO Box 83992, Room 1817, Apt 1413, Room 427, Room 1318, Suite 98, Room 1784, Apt 266, Room 1919, Suite 5, Suite 93, Room 1409, 2nd Floor, Apt 597, Room 1328, Suite 7, Apt 244, 2nd Floor, PO Box 98411, Apt 1015, and Room 913.

	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:



The screenshot shows a SQL command prompt with the command `SQL> DESC Warehouse` and its output. The output is a table with three columns: Name, Null?, and Type. The data rows are ADDRESS, WHID, and WHSIZE, all with Null? set to NOT NULL and appropriate data types.

Name	Null?	Type
ADDRESS	NOT NULL	VARCHAR2(35)
WHID	NOT NULL	NUMBER(38)
WHSIZE	NOT NULL	NUMBER(38)

דרך ב - data generator :

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



The screenshot shows a web application interface for generating a MySQL database schema. The main form is titled 'New Data Set' and contains two columns: 'numOfSoldiers' and 'soldierId'. The 'numOfSoldiers' column is set to 'Number Range' with a range from 999 to 999. The 'soldierId' column is set to 'Auto-Increment' with a start value of 1 and an increment of 1. A 'MySQL' tab on the right displays the generated SQL code:

```

1 DROP TABLE IF EXISTS `myTable`;
2
3 CREATE TABLE `myTable` (
4   `id` mediumint(8) unsigned NOT NULL auto_increment,
5   `numOfSoldiers` mediumint default NULL,
6   `soldierId` mediumint,
7   PRIMARY KEY (`id`)
8 ) AUTO_INCREMENT=1;
9
10 INSERT INTO `myTable` (`numOfSoldiers`,`soldierId`)
11 VALUES
12   (125700,1),
13   (795928,2),
14   (112661,3),
15   (467246,4),
16   (911369,5);
17

```

אנו מכניסים נתונים ל-CREATE TABLE זה:

The screenshot shows a SQL editor with the following SQL code:

```

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

```

הנה הפקודות שאנו מריצים:

The screenshot shows a SQL editor with the following SQL code:

```

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

```

```

CREATE TABLE Officers.sql | DataGeneratorOfficer.sql | DROP TABLE Officer; | INSERT INTO Officer
SQL | Output | Statistics
INSERT INTO `Officer` (`numOfSoldiers`,`soldierId`)
VALUES
  (191572,1),
  (70994,2),
  (104574,3)

```

והנתונים נכנסו בהצלחה:

```

CREATE TABLE Officers.sql | DataGeneratorOfficer.sql | Select* from Officer; | commit;
SQL | Output | Statistics
Select*
from Officer;

```

	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 desc על התרשים המבוקש

```

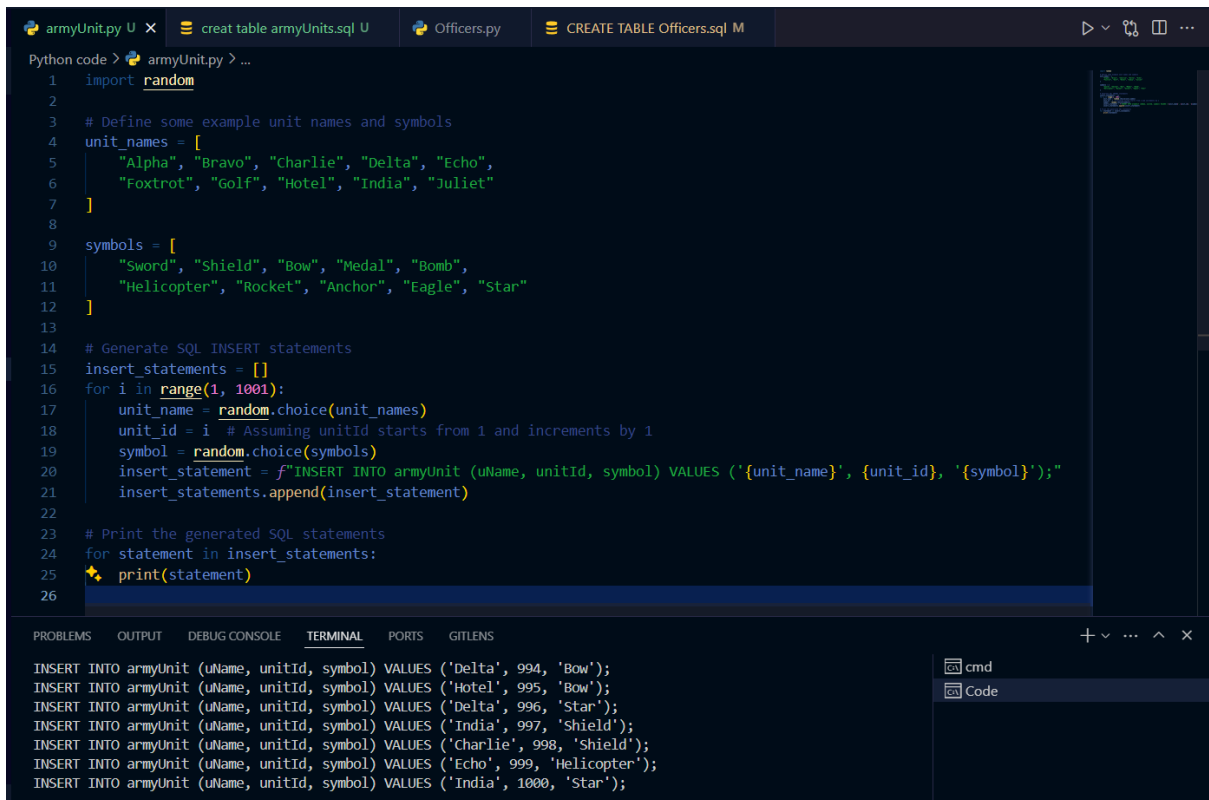
SQL> desc officer
Name                                     Null?   Type
-----
NUMOFSOLDIERS                          NOT NULL NUMBER(38)
SOLDIERID                               NOT NULL NUMBER(38)

```

דרך ג - python insert :

יצירה ע"י פייתון:

עבור armyUnit

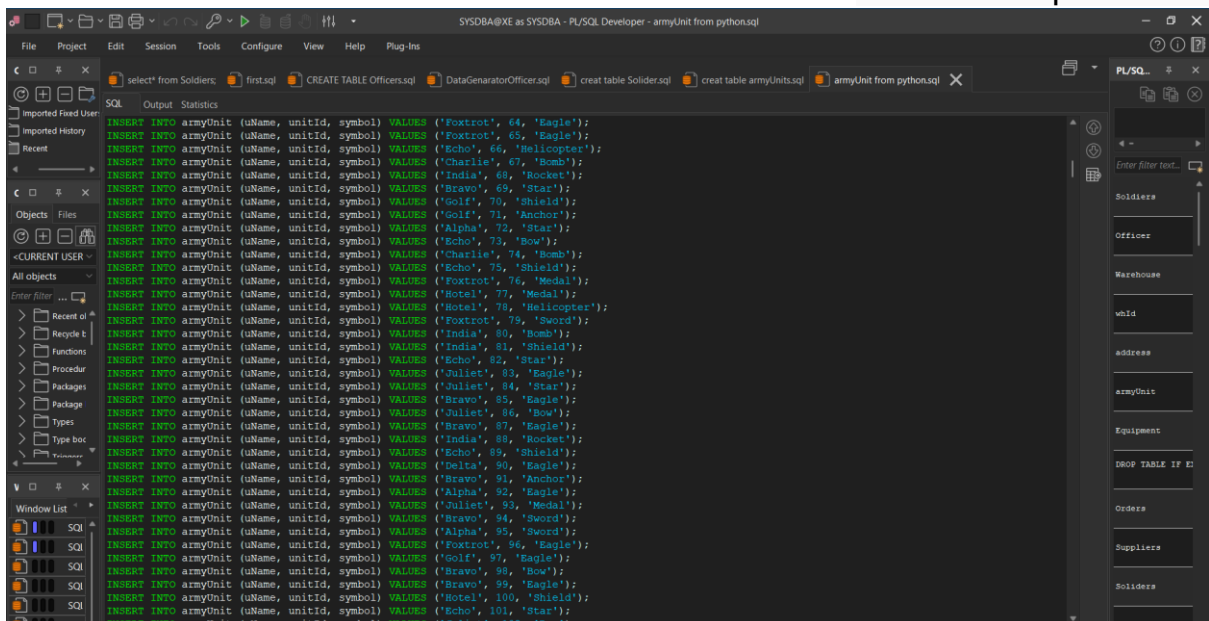


```
Python code > armyUnit.py > ...
1 import random
2
3 # Define some example unit names and symbols
4 unit_names = [
5     "Alpha", "Bravo", "Charlie", "Delta", "Echo",
6     "Foxtrot", "Golf", "Hotel", "India", "Juliet"
7 ]
8
9 symbols = [
10    "Sword", "Shield", "Bow", "Medal", "Bomb",
11    "Helicopter", "Rocket", "Anchor", "Eagle", "Star"
12 ]
13
14 # Generate SQL INSERT statements
15 insert_statements = []
16 for i in range(1, 1001):
17     unit_name = random.choice(unit_names)
18     unit_id = i # Assuming unitId starts from 1 and increments by 1
19     symbol = random.choice(symbols)
20     insert_statement = f"INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('{unit_name}', {unit_id}, '{symbol}');"
21     insert_statements.append(insert_statement)
22
23 # Print the generated SQL statements
24 for statement in insert_statements:
25     print(statement)
26
```

TERMINAL

```
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');
```

הנה הפקודות שנוצרו:



SQL

```
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Foxtrot', 64, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Foxtrot', 65, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 66, 'Helicopter');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Charlie', 67, 'Bomb');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('India', 68, 'Rocket');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 69, 'Star');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Golf', 70, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Golf', 71, 'Anchor');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Alpha', 72, 'Star');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 73, 'Bow');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Charlie', 74, 'Bomb');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 75, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Foxtrot', 76, 'Medal');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Hotel', 77, 'Medal');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Hotel', 78, 'Helicopter');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Foxtrot', 79, 'Sword');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('India', 80, 'Bomb');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('India', 81, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 82, 'Star');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Juliet', 83, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Juliet', 84, 'Star');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 85, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Juliet', 86, 'Bow');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 87, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('India', 88, 'Rocket');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 89, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Delta', 90, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 91, 'Anchor');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Alpha', 92, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Juliet', 93, 'Medal');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 94, 'Sword');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Alpha', 95, 'Sword');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Foxtrot', 96, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Golf', 97, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 98, 'Bow');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Bravo', 99, 'Eagle');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Hotel', 100, 'Shield');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Echo', 101, 'Star');
INSERT INTO armyUnit (uName, unitId, symbol) VALUES ('Juliet', 102, 'Bow');
```

SQL Output Statistics

```
select*
from armyUnit;
```

	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

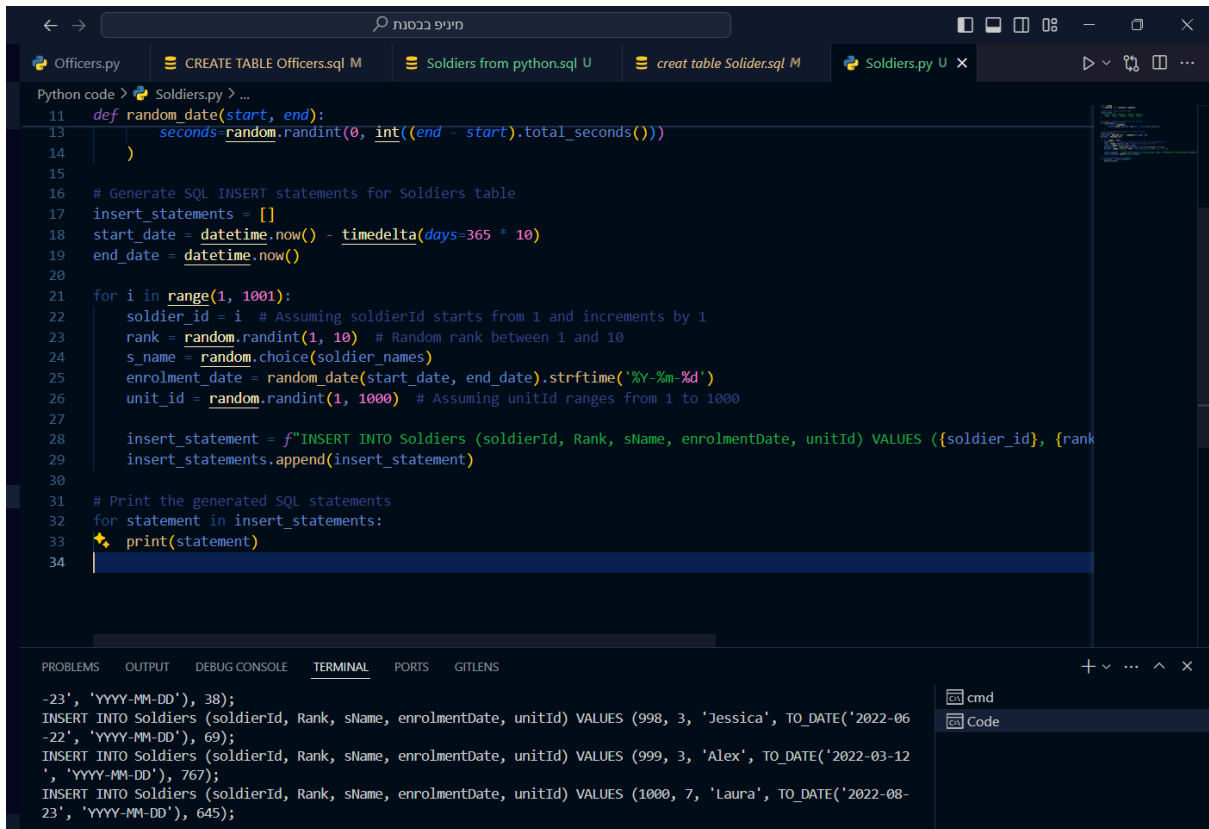
הפעלנו פקודת desc עבור armyUnit

```
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:

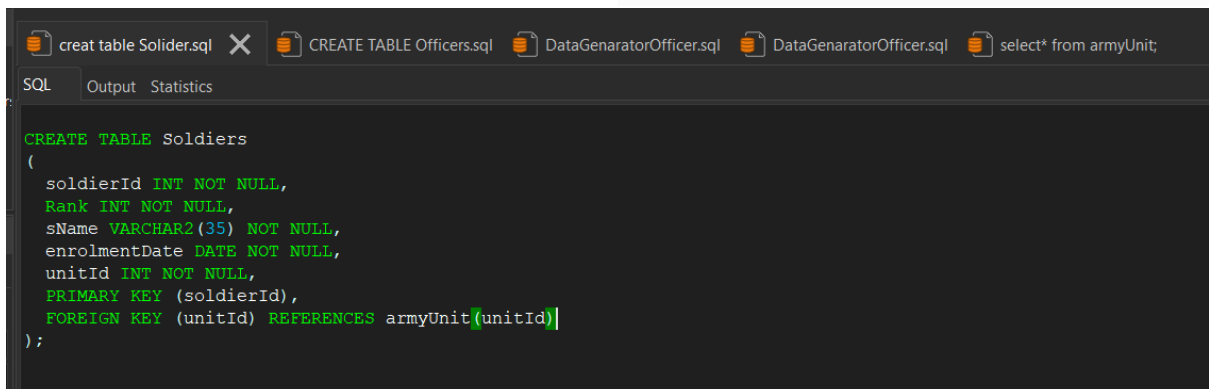
## עבוד Soldiers



```
Python code > Soldiers.py > ...
11 def random_date(start, end):
12     seconds=random.randint(0, int((end - start).total_seconds()))
13     )
14
15
16 # Generate SQL INSERT statements for Soldiers table
17 insert_statements = []
18 start_date = datetime.now() - timedelta(days=365 * 10)
19 end_date = datetime.now()
20
21 for i in range(1, 1001):
22     soldier_id = i # Assuming soldierId starts from 1 and increments by 1
23     rank = random.randint(1, 10) # Random rank between 1 and 10
24     s_name = random.choice(soldier_names)
25     enrolment_date = random_date(start_date, end_date).strftime('%Y-%m-%d')
26     unit_id = random.randint(1, 1000) # Assuming unitId ranges from 1 to 1000
27
28     insert_statement = f"INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES ({soldier_id}, {rank}, {s_name}, {enrolment_date}, {unit_id})"
29     insert_statements.append(insert_statement)
30
31 # Print the generated SQL statements
32 for statement in insert_statements:
33     print(statement)
34
```

```
-23', 'YYYY-MM-DD'), 38);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (998, 3, 'Jessica', TO_DATE('2022-06-22', 'YYYY-MM-DD'), 69);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (999, 3, 'Alex', TO_DATE('2022-03-12', 'YYYY-MM-DD'), 767);
INSERT INTO Soldiers (soldierId, Rank, sName, enrolmentDate, unitId) VALUES (1000, 7, 'Laura', TO_DATE('2022-08-23', 'YYYY-MM-DD'), 645);
```

## הנה ה create table וההכנסת נתונים:



```
creat table Solider.sql X CREATE TABLE Officers.sql DataGenaratorOfficer.sql DataGenaratorOfficer.sql select* from armyUnit;
SQL Output Statistics

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)
);
```

```
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:

	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 x Suppliers from python.sql U
Python code > suppliers.py > ...
You, 1 second ago | 1 author (You)
1 import random
2
3 # Define some example supplier names
4 supplier_names = [
5     "Supplier A", "Supplier B", "Supplier C", "Supplier D", "Supplier E",
6     "Supplier F", "Supplier G", "Supplier H", "Supplier I", "Supplier J"
7 ]
8
9 # Generate SQL INSERT statements for Suppliers table
10 insert_statements = []
11 for i in range(1, 1001):
12     supplier_id = i # Assuming supplierId starts from 1 and increments by 1
13     name = random.choice(supplier_names)
14     seniority = random.randint(1, 30) # Random seniority between 1 and 30 years
15
16     insert_statement = f"INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES ({supplier_id}, '{name}', {seniority})"
17     insert_statements.append(insert_statement)
18
19 # Print the generated SQL statements
20 for statement in insert_statements:
21     print(statement)
22
```

```
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);
```

```
CREATE TABLE Officers.sql DataGeneratorOfficer.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:

```
CREATE TABLE Officers.sql | DataGeneratorOfficers.sql | DataGeneratorOfficers.sql | select* from armyUnit; | Creat Table Suppliers.sql | Suppliers from python.sql X
```

```
SQL Output Statistics
```

```
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (1, 'Supplier D', 24);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (2, 'Supplier E', 12);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (3, 'Supplier J', 24);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (4, 'Supplier E', 2);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (5, 'Supplier B', 21);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (6, 'Supplier I', 15);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (7, 'Supplier G', 25);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (8, 'Supplier G', 9);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (9, 'Supplier J', 10);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (10, 'Supplier B', 22);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (11, 'Supplier I', 14);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (12, 'Supplier D', 7);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (13, 'Supplier B', 13);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (14, 'Supplier E', 18);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (15, 'Supplier C', 14);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (16, 'Supplier C', 8);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (17, 'Supplier E', 29);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (18, 'Supplier D', 30);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (19, 'Supplier B', 5);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (20, 'Supplier H', 30);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (21, 'Supplier H', 15);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (22, 'Supplier H', 5);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (23, 'Supplier E', 5);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (24, 'Supplier B', 13);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (25, 'Supplier F', 4);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (26, 'Supplier J', 24);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (27, 'Supplier D', 8);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (28, 'Supplier I', 21);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (29, 'Supplier D', 11);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (30, 'Supplier A', 24);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (31, 'Supplier B', 17);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (32, 'Supplier I', 24);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (33, 'Supplier F', 29);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (34, 'Supplier H', 12);
INSERT INTO Suppliers (supplierId, Name, Seniority) VALUES (35, 'Supplier C', 28);
```

## וכל הנתונים נכנסו באופן טוב:

CREATE TABLE Officers.sql | DataGeneratorOfficers.sql | DataGeneratorOfficers.sql | select\* from Suppliers; X | Creat Table Suppliers.sql | Suppliers from python.sql

SQL Output Statistics

```
select*
from Suppliers;
```

	SUPPLIERID	NAME	SENIORITY
1	1	Supplier D	24
2	2	Supplier E	12
3	3	Supplier J	24
4	4	Supplier E	2
5	5	Supplier B	21
6	6	Supplier I	15
7	7	Supplier G	25
8	8	Supplier G	9
9	9	Supplier J	10
10	10	Supplier B	22
11	11	Supplier I	14
12	12	Supplier D	7
13	13	Supplier B	13
14	14	Supplier E	18
15	15	Supplier C	14
16	16	Supplier C	8
17	17	Supplier E	29
18	18	Supplier D	30
19	19	Supplier B	5
20	20	Supplier H	30
21	21	Supplier H	15
22	22	Supplier H	5
23	23	Supplier E	5
24	24	Supplier B	13
25	25	Supplier F	4
26	26	Supplier J	24
27	27	Supplier D	8
28	28	Supplier I	21
29	29	Supplier D	11

## פעולת Desc על Suppliers:

SQL> Desc Suppliers

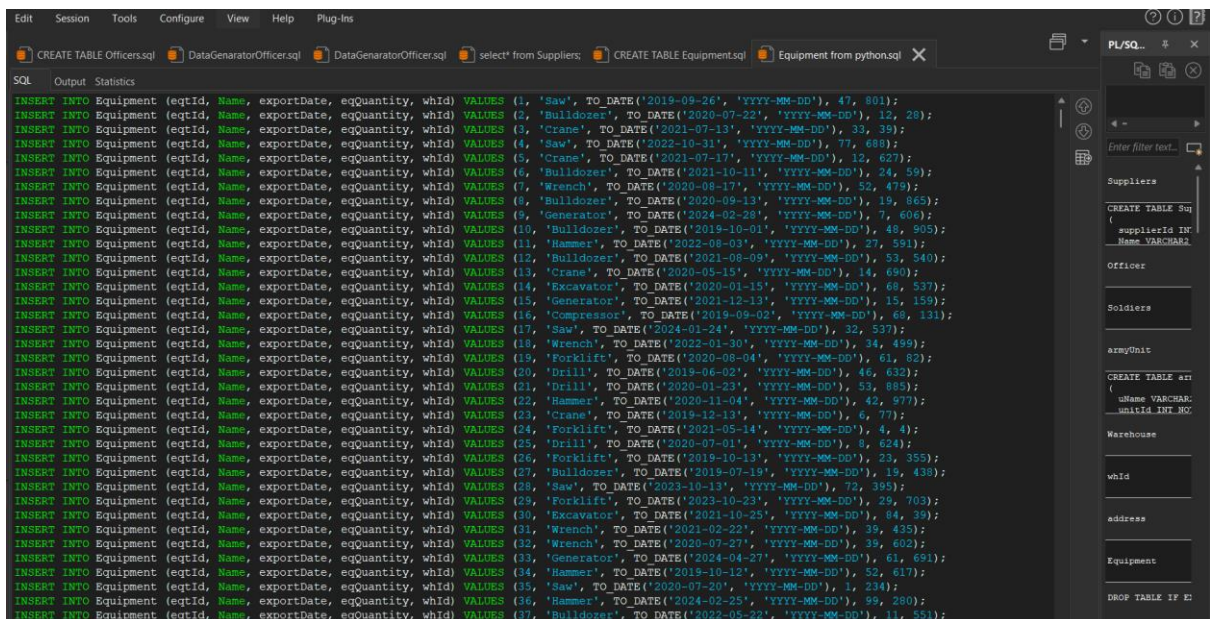
Name	Null?	Type
SUPPLIERID	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(35)
SENIORITY	NOT NULL	NUMBER(38)

## עבור Equipment

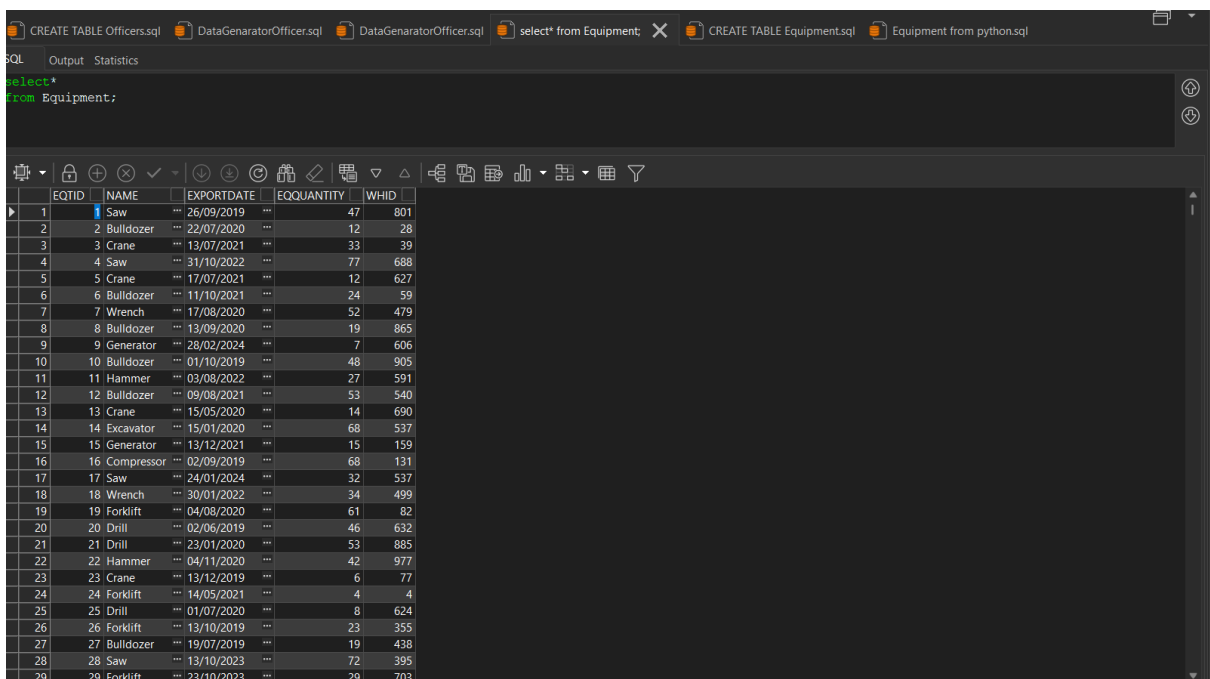
```
CREATE TABLE Equipment
(
    eqtId INT NOT NULL,
    Name VARCHAR2(35) NOT NULL,
    exportDate DATE NOT NULL,
    eqQuantity INT NOT NULL,
    whId INT NOT NULL,
    PRIMARY KEY (eqtId),
    FOREIGN KEY (whId) REFERENCES Warehouse (whId)
);
```

```
Equipment from python.sql U
Equipment.py U X
Python code > Equipment.py > ...
9
10 # Generate a random date within the last 5 years
11 def random_date(start, end):
12     return start + timedelta(
13         seconds=random.randint(0, int((end - start).total_seconds()))
14     )
15
16 # Generate SQL INSERT statements for Equipment table
17 insert_statements = []
18 start_date = datetime.now() - timedelta(days=365 * 5)
19 end_date = datetime.now()
20
21 for i in range(1, 1001):
22     eqt_id = i # Assuming eqtId starts from 1 and increments by 1
23     name = random.choice(equipment_names)
24     export_date = random_date(start_date, end_date).strftime('%Y-%m-%d')
25     eq_quantity = random.randint(1, 100) # Random quantity between 1 and 100
26     wh_id = random.randint(1, 1000) # Assuming whId ranges from 1 to 1000
27
28     insert_statement = f"INSERT INTO Equipment (eqtId, Name, exportDate, eqQuantity, whId) VALUES ({eqt_id}, '{name}', TO_DATE('{export_date}', 'YYYY-MM-DD'), {eq_quantity}, {wh_id});"
29     insert_statements.append(insert_statement)
30
31 # Print the generated SQL statements
32 for statement in insert_statements:
33     print(statement)
34
```

```
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);
```



ועכשיו כל הנתונים נכנסו כמו שצריך:



ופעולת desc עבור Equipment:

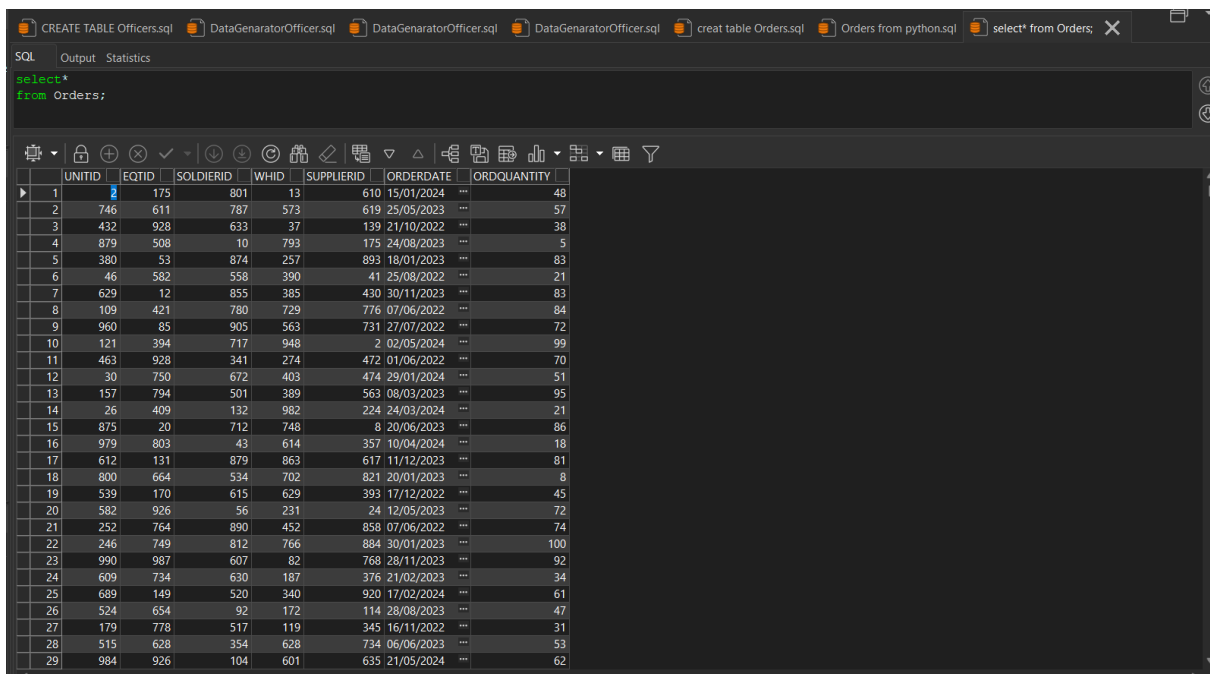
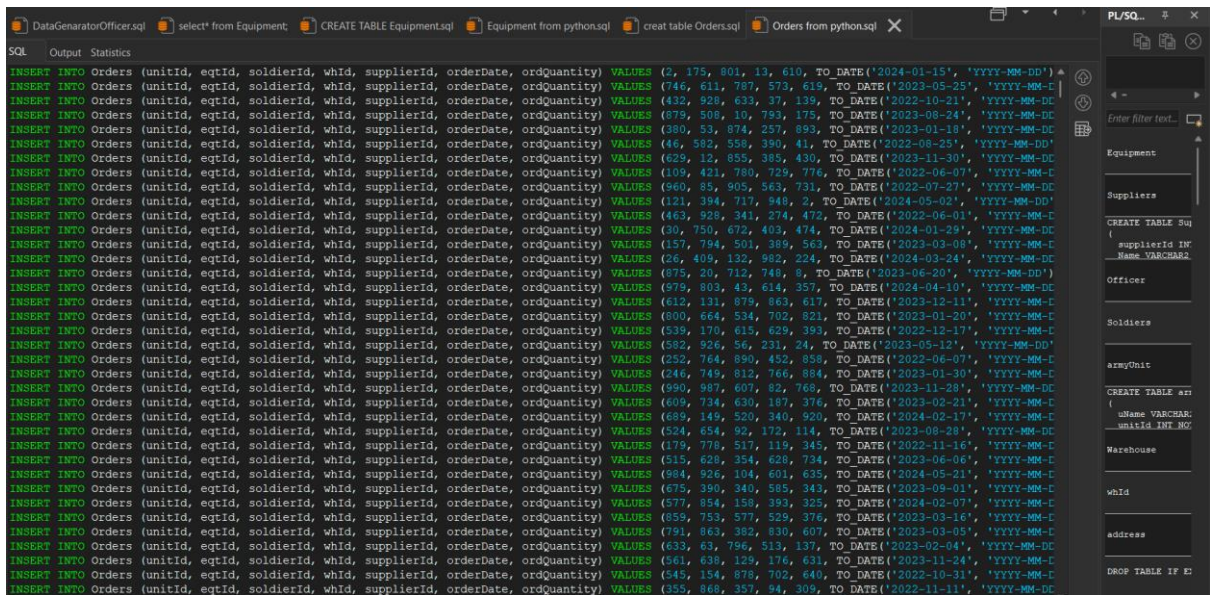
SQL> Desc Equipment		
Name	Null?	Type
-----	-----	-----
EQTID	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(35)
EXPORTDATE	NOT NULL	DATE
EQQUANTITY	NOT NULL	NUMBER(38)
WHID	NOT NULL	NUMBER(38)

## עבור Orders

```
Python code > Orders.py > ...
You, 60 seconds ago | 1 author (You)
1 import random
2 from datetime import datetime, timedelta
3
4 # Generate a random date within the last 2 years
5 def random_date(start, end):
6     return start + timedelta(
7         seconds=random.randint(0, int((end - start).total_seconds()))
8     )
9
10 # File path where the SQL insert statements will be saved
11 file_path = r'C:\Users\sapir\OneDrive\העבודה\נסיון\מסר ב שנה גס\ולחן העבודה\יניפ בבסנתמ\מסר ב שנה גס\ולחן העבודה\יניפ בבסנתמ\Insert Code\Orders from python.sql'
12
13 # Generate SQL INSERT statements for Orders table
14 insert_statements = []
15 start_date = datetime.now() - timedelta(days=365 * 2)
16 end_date = datetime.now()
17
18 for i in range(1, 1001):
19     unit_id = random.randint(1, 1000) # Assuming unitId ranges from 1 to 1000
20     eqt_id = random.randint(1, 1000) # Assuming eqtId ranges from 1 to 1000
21     soldier_id = random.randint(1, 1000) # Assuming soldierId ranges from 1 to 1000
22     wh_id = random.randint(1, 1000) # Assuming whId ranges from 1 to 1000
23     supplier_id = random.randint(1, 1000) # Assuming supplierId ranges from 1 to 1000
24     order_date = random_date(start_date, end_date).strftime('%Y-%m-%d')
25     ord_quantity = random.randint(1, 100) # Random order quantity between 1 and 100
26
27     insert_statement = f"INSERT INTO Orders (unitId, eqtId, soldierId, whId, supplierId, orderDate, ordQuantity) VALUES ("
28     insert_statements.append(insert_statement)
29
30 # Save the generated SQL statements to the specified file
31 with open(file_path, 'w') as file:
32     for statement in insert_statements:
33         file.write(statement + '\n')
34
35 print(f"SQL insert statements have been written to {file_path}")
36
```

```
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)
);
```



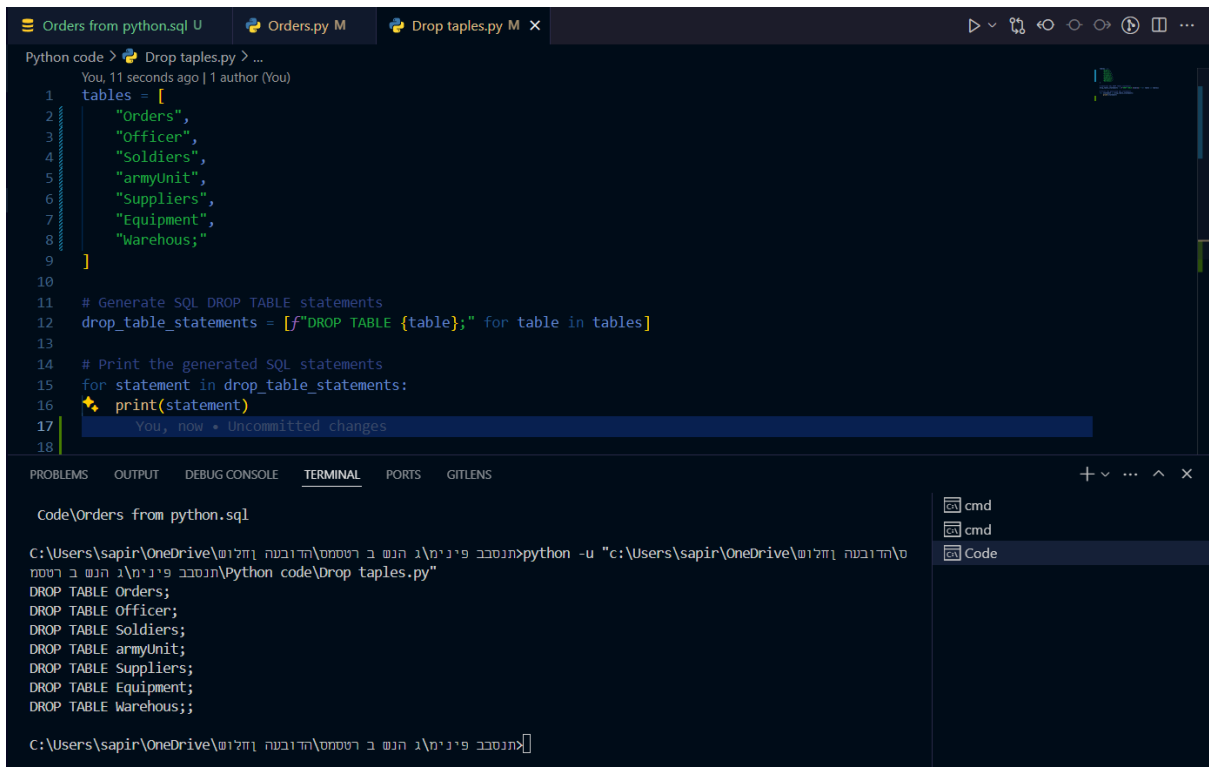


הוצאת פעולת desc על Orders:

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 בפיייתון שמייצר את פקודות ה drop tables



The screenshot shows a VS Code editor with a Python file named `Drop tables.py`. The code defines a list of table names and generates SQL `DROP TABLE` statements for each. The terminal output shows the execution of the script, which prints the generated SQL statements.

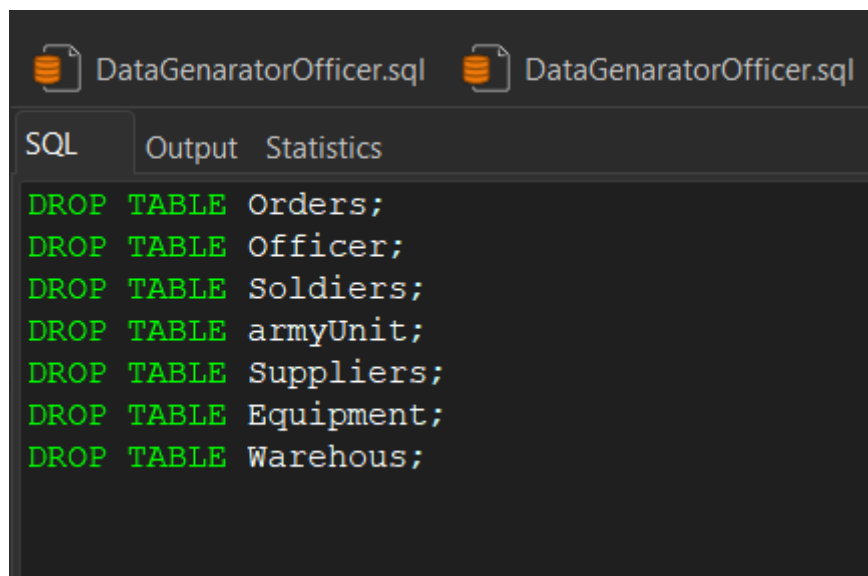
```
Python code > Drop tables.py > ...
You, 11 seconds ago | 1 author (You)

1 tables = [
2     "Orders",
3     "Officer",
4     "Soldiers",
5     "armyUnit",
6     "Suppliers",
7     "Equipment",
8     "Warehous;"
9 ]
10
11 # Generate SQL DROP TABLE statements
12 drop_table_statements = [f"DROP TABLE {table};" for table in tables]
13
14 # Print the generated SQL statements
15 for statement in drop_table_statements:
16     print(statement)
17
18 You, now * Uncommitted changes
```

Code\Orders from python.sql

```
C:\Users\sapir\OneDrive\מחשבים\הדיובה\Python code\Drop tables.py
DROP TABLE Orders;
DROP TABLE Officer;
DROP TABLE Soldiers;
DROP TABLE armyUnit;
DROP TABLE Suppliers;
DROP TABLE Equipment;
DROP TABLE Warehous;;
```

וכך יצאו לנו הפקודות עצמן:



The screenshot shows a SQL editor with a file named `DataGenaratorOfficer.sql`. The editor displays the generated SQL statements in the `SQL` tab.

```
DROP TABLE Orders;
DROP TABLE Officer;
DROP TABLE Soldiers;
DROP TABLE armyUnit;
DROP TABLE Suppliers;
DROP TABLE Equipment;
DROP TABLE Warehous;
```

גיבויי 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 insert.

לדוגמה – עבור הטבלה armyUnit:

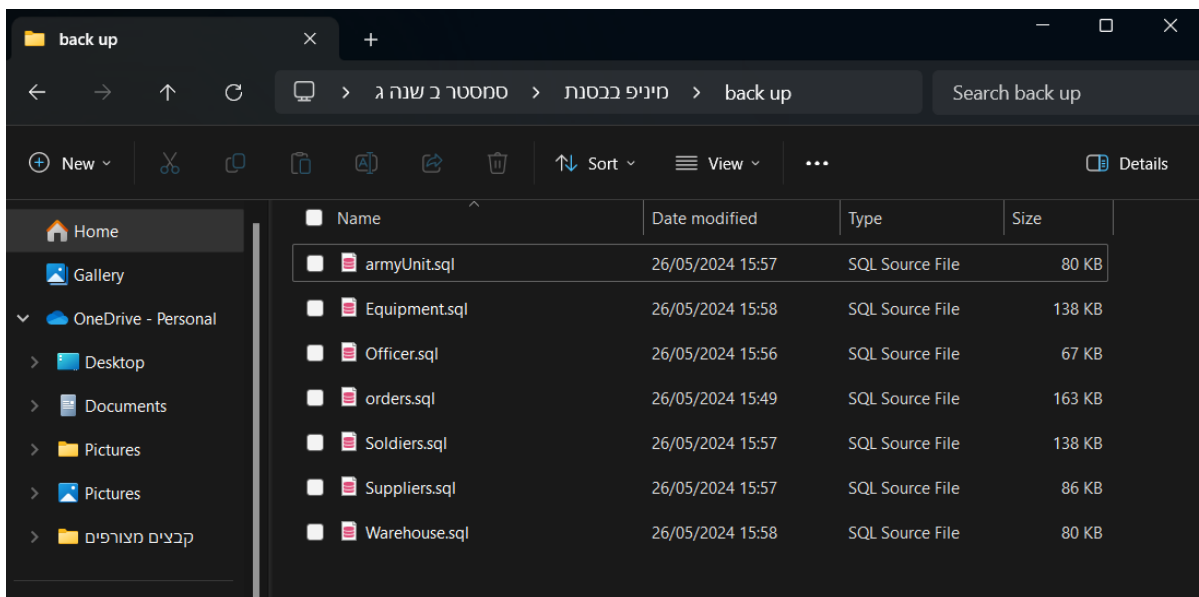
יצוא הקובץ כמתואר:

The screenshot shows a database management interface. At the top, there are tabs for 'CREATE TABLE Officers.sql', 'Export Tables of SYS', and 'select\* from armyUnit;'. Below the tabs, there are three buttons: 'SQL', 'Output', and 'Statistics'. The 'SQL' button is active, and the query 'select\* from armyUnit;' is entered in the text area. Below the text area, there is a toolbar with various icons. The main area displays a table with 27 rows and 4 columns: 'ID', 'UNAME', 'UNITID', and 'SYMBOL'. The table contains data for various units, including Golf, India, Bravo, Delta, Charlie, Hotel, Alpha, Echo, Juliet, and Charlie. A context menu is open over the table, showing options for exporting the data to various formats: CSV file, TSV file, HTML file, XML file, SQL file, Excel file, Column to files, and To clipboard as.

ID	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:





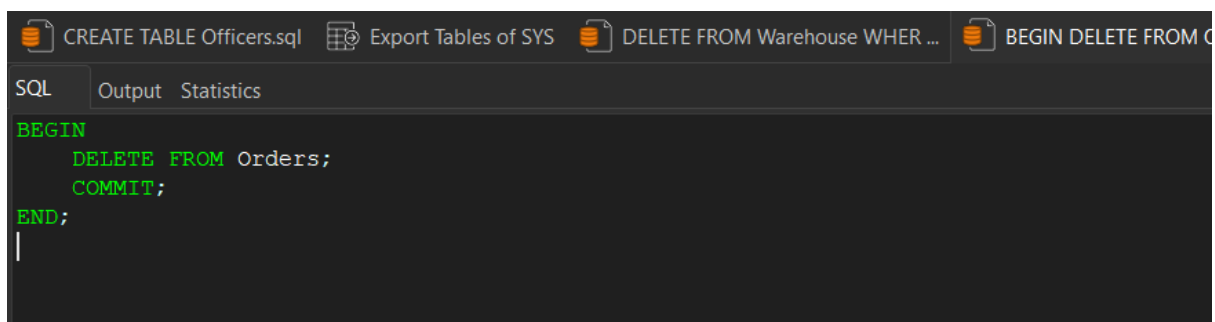
בכך גיבינו את כל טבלאותינו בהצלחה.

### שיחזור

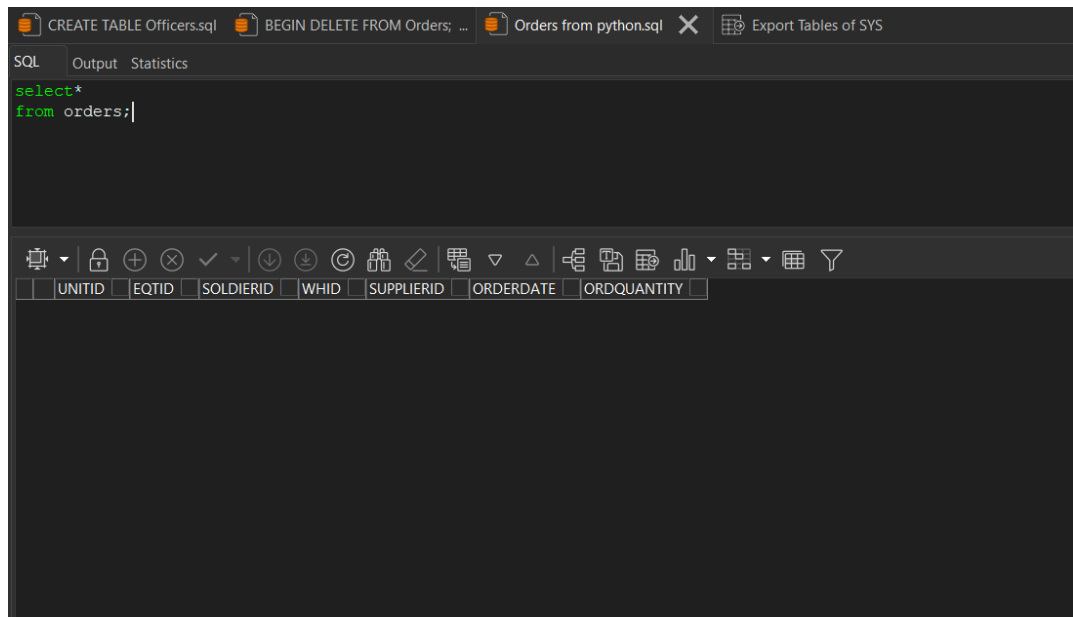
כעת כדי לשחזר את הקובץ – במידה ונצטרך, ניכנס לאופציה `import table->sql`, נבחר את הקובץ ממנו נרצה לשחזר את הטבלה למטה, ונלחץ `import`. מיד יופיעו לנו כל הinsertים שיחזירו לנו את הטבלה המקורית. כל שנשאר הוא ללחוץ על `execute`, ולבדוק האם השיחזור הצליח.

נמחיש את התהליך בדוגמה על הטבלה Orders:

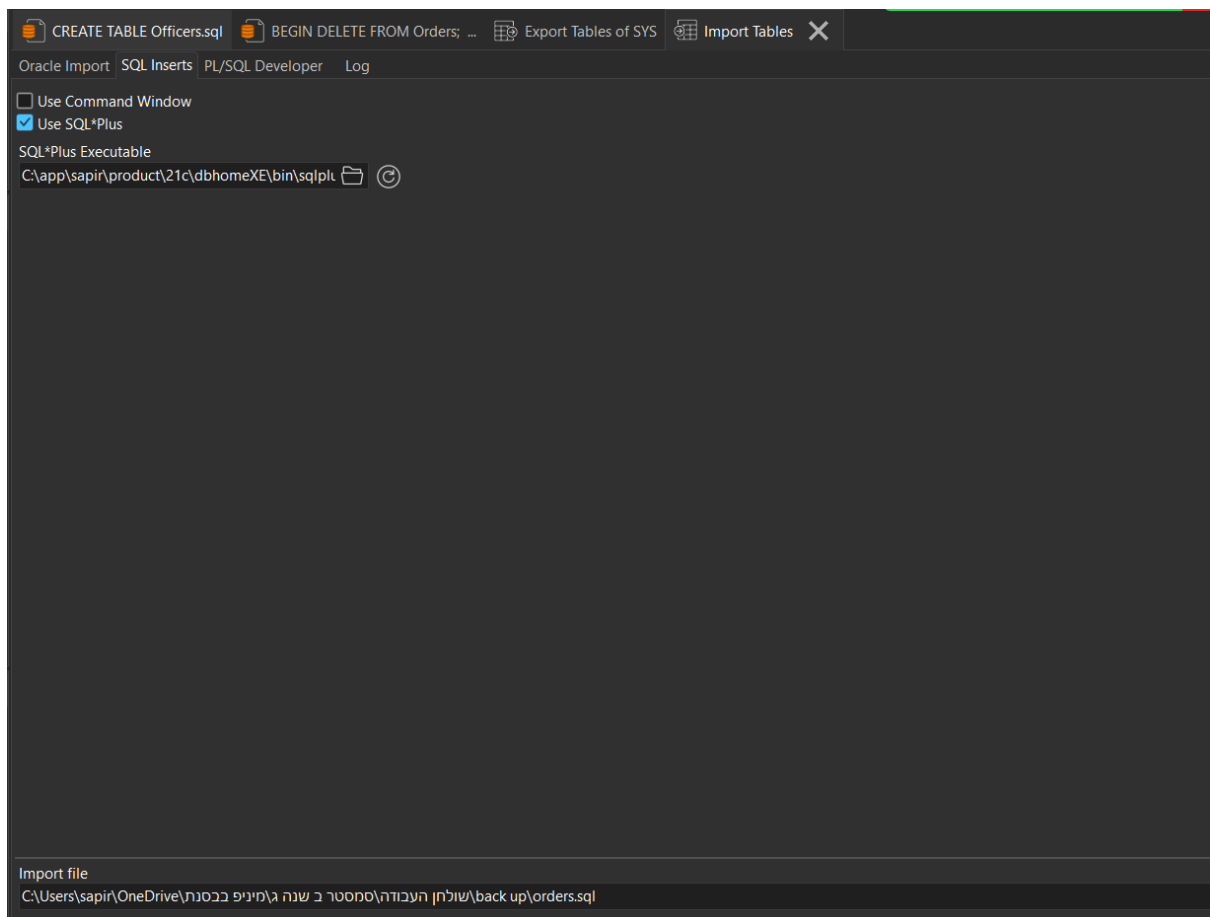
תחילה נראה שיש למחוק לפחות חלק מהטבלה – ע"מ לראות אם השיחזור מצליח. נמחק את נתוני הטבלה בשאלתה המתוארת בתמונה:



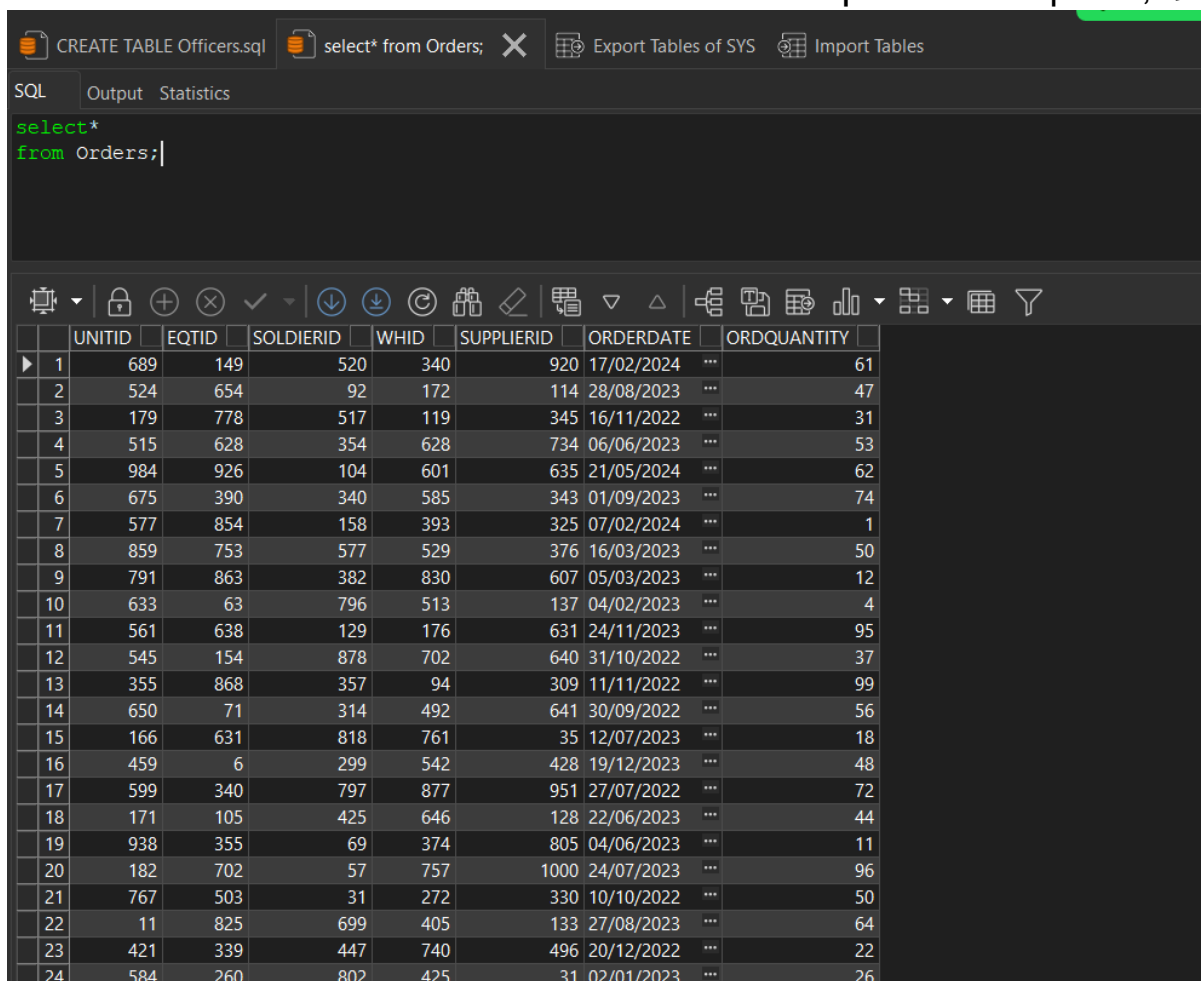
כעת, נראה שהנתונים אכן נמחקו:



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



כעת, נבדוק שהנתונים אכן שוחזרו:



	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!