

מיני פרויקט בבסיסי נתונים**חיל המודיעין של צה"ל****תוכן עניינים**

1	שלב 1
1	תיאור הארגון: ERD & DSD
2	תרשים ERD של הארגון: הטבלאות המתקבלות: בדיקת נרמול: create table פקודות: שלב הכנסת הנתונים: mockaroo דרך א - דרך ב - data generator: דרך ג: פקודת desc עבור personnel:
2	
3	
3	
4	
6	
6	
8	
9	
10	

שלב 1**תיאור הארגון:**

חיל המודיעין של צה"ל, ידוע גם בשמו הרשמי "אגף המודיעין", הוא אחד מהיחידות המרכזיות והחשובות בצבא ההגנה לישראל. הוא אחראי על איסוף מודיעין, ניתוחו והפצתו לכלל גופי הביטחון במדינה. מטרתו העיקרית היא לספק תמונת מצב מדויקת ועדכנית שתאפשר לגורמים השונים לקבל החלטות מושכלות בזמן אמת.

תפקידים עיקריים של חיל המודיעין:

איסוף מודיעין: חיל המודיעין פועל במגוון דרכים כדי לאסוף מידע מהמגזר האזרחי והצבאי של מדינות זרות וארגונים. האיסוף מתבצע באמצעות אמצעים טכנולוגיים, אלקטרוניים, אוויריים ועוד.

ניתוח מודיעין: המידע שנאסף עובר ניתוח מקיף על ידי מומחים ואנליסטים, שמטרתם להבין את המגמות, לזהות מגמות ולחזות התפתחויות אפשריות.

הפצת מודיעין: המודיעין שניתוח מועבר לגורמים הרלוונטיים בצה"ל ובמערכת הביטחון, כדי לאפשר להם לפעול בהתאם למידע שהתקבל.

תמיכה בקבלת החלטות: חיל המודיעין משמש כזרוע ייעוץ למפקדי צה"ל ולממשלה, בהקשר של תכנון מבצעי ואסטרטגי.

יחידות מרכזיות בחיל המודיעין:

אמ"ן (אגף מודיעין ומחקר): אחראי על איסוף וניתוח מודיעין אסטרטגי.

יחידת המודיעין הטריטוריאלי: אחראית על איסוף מודיעין באזורים מסוימים ולפי גזרות.

יחידה למודיעין חוץ: עוסקת באיסוף מודיעין ממקורות חוץ.

חיל המודיעין מספק כלים קריטיים להבטחת ביטחון ישראל, ומשחק תפקיד חיוני בשמירה על יכולת ההרתעה וההגנה של המדינה.

:ERD & DSD

המערכת מנהלת היבטים שונים של תפעול, כוח אדם, ציוד ודיווח מודיעיני. להלן תיאור קצר של הפונקציונליות והמערכת:

דוחות מודיעין: המערכת מתעדת דוחות מודיעין המכילים פרטים כמו תוכן, תאריך הגשה והצוות ששלח אותם. ניתן לשייך דוחות אלה לפעולות ספציפיות.

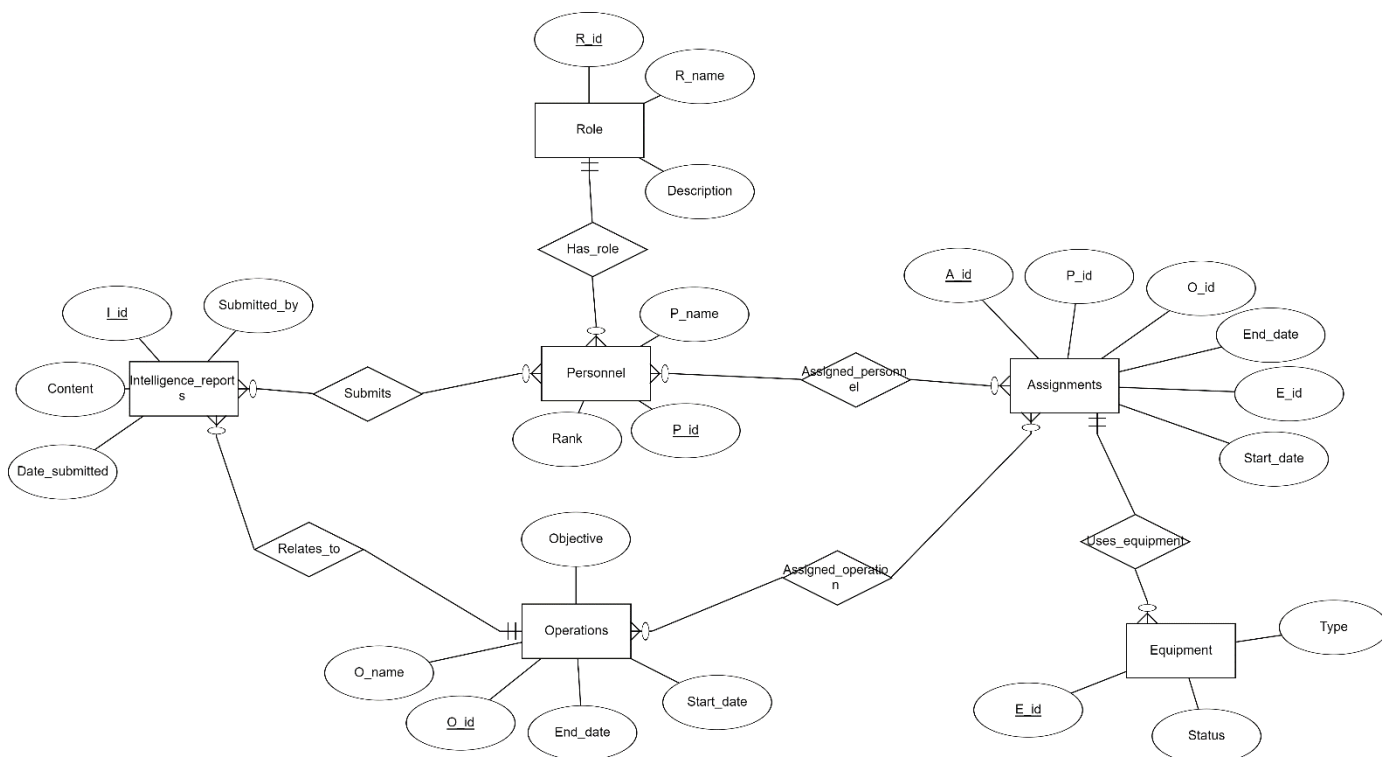
ניהול כוח אדם: כוח אדם הוא מרכזי במערכת, כל אחד מזוהה על ידי מזהה ייחודי, ויש להם תכונות כמו שם ודירוג. לעובדים מוקצים תפקידים בתוך הארגון, המתוארים בשמות התפקידים ובתיאוריהם.

ניהול תפעול: פעולות מוגדרות בתוך המערכת עם תכונות כמו שם, תאריך התחלה ותאריך סיום. פעולות מקושרות ליעדים ספציפיים, המתארים עוד יותר את המשימה או המטרות של פעולות אלו.

מטלות: המערכת עוקבת אחר מטלות על ידי קישור כוח אדם ותפעול. ניתן להקצות כוח אדם לפעולות ספציפיות, ומשימות אלו כוללות פרטים כגון תאריכי התחלה וסיום.

מעקב אחר ציוד: מעקב אחר הציוד מתבצע לפי סוג ומצב. המערכת מאפשרת קצאת ציוד לפעולות ספציפיות, מה שמאפשר ניהול משאבים מפורט.

תרשים ERD של הארגון:

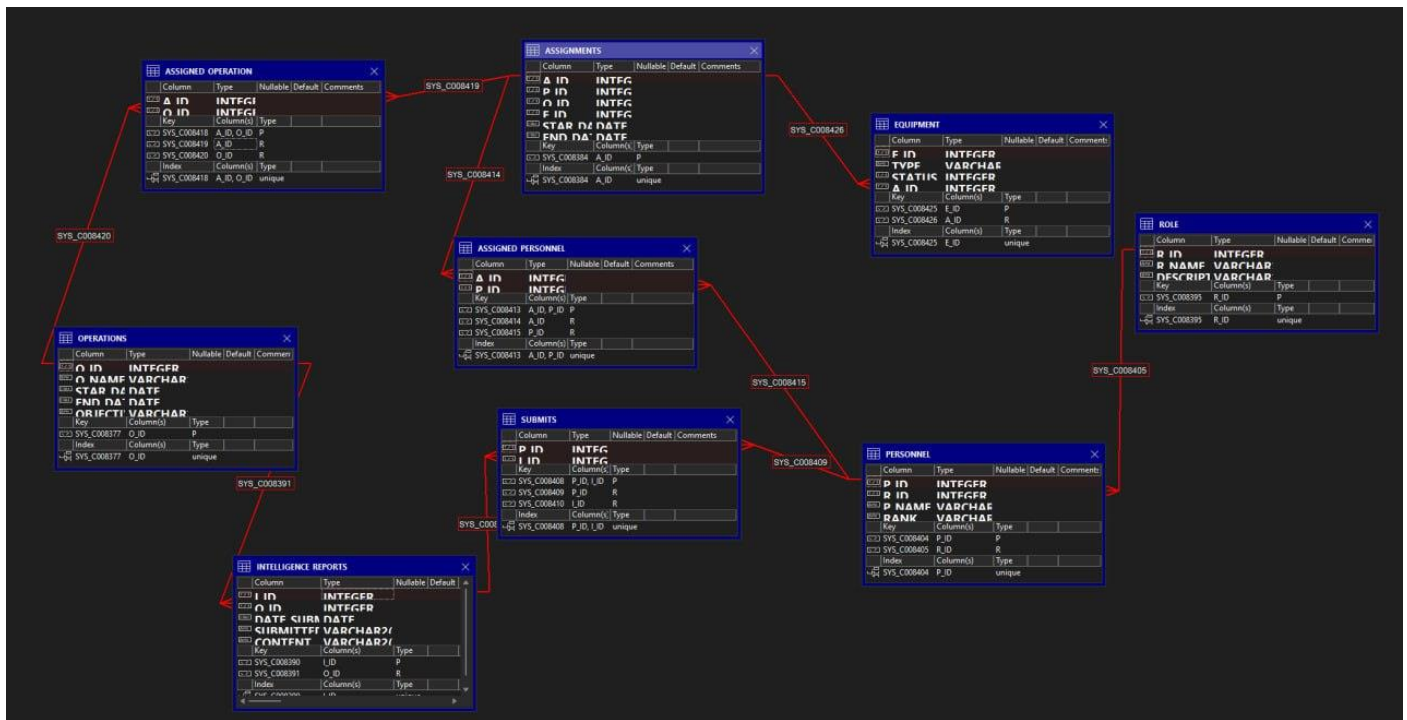


הטבלאות המתקבלות:

Personnel (P_id, P_name, R_id, Rank)**Role** (R_id, R_name, Description)**Operations** (O_id, O_name, Start_date, End_date, Objective)**Intelligence_Reports** (I_id, O_id, Submitted_by, Date_submitted, Content)**Equipment** (E_id, A_id, Type, Status)**Assignments** (A_id, P_id, O_id, E_ID, Start_date, End_date)**Assigned_operation** (A_id, O_id)**Assigned_personnel** (A_id, P_id)**Submits** (P_id, I_id)

בדיקת נרמול:

- ✓ **Equipment**: סוג הציוד והסטטוס תלויים במפתח E_id, ולכן מתקיים 3NF.
- ✓ **Role**: שם התפקיד והתיאור תלויים במפתח R_id, ולכן מתקיים 3NF.
- ✓ **Personnel**: כל השדות תלויים במפתח P_id וכן אין שום תלות נוספת בין שני שדות שאינם מפתחות, ולכן מתקיים 3NF.
- ✓ **Operations**: השם והתאריכים תלויים במפתח O_id ואין תלות בניהם - מתקיים 3NF.
- ✓ **Intelligence_Reports**: לפי מספר הדו"ח נוכל לקבוע את כל שאר הפרטים כמו שם ותאריכים ולכן מתקיים 3NF.
- ✓ **Assignments**: התלות היחידה שמתקיימת היא כל השדות במפתח A_id, ולכן מתקיים 3NF.
- ✓ **Submits, Assigned_personnel, Assigned_operation**: בכלם יש רק 2 שדות ולכן מתקיים BCNF כלומר מתקיים 3NF.



create table פקודות:

```
CREATE TABLE Operations_
(
O_id INT NOT NULL,
O_name VARCHAR2(20) NOT NULL,
Star_date DATE NOT NULL,
End_date DATE NOT NULL,
Objective VARCHAR2(200) NOT NULL,
PRIMARY KEY (O_id)
);
```

```
CREATE TABLE Assignments
(
A_id INT NOT NULL,
P_id INT NOT NULL,
O_id INT NOT NULL,
E_id INT NOT NULL,
Star_date DATE NOT NULL,
End_date DATE NOT NULL,
```

מגשים: משה שחר ואהרן כץ
PRIMARY KEY (A_id)

);

CREATE TABLE Intelligence_reports

(

I_id INT NOT NULL,

O_id INT NOT NULL,

Submitted_by VARCHAR2(20) NOT NULL,

Date_submitted DATE NOT NULL,

Content_ VARCHAR2(32767) NOT NULL,

PRIMARY KEY (I_id),

FOREIGN KEY (O_id) REFERENCES Operations_(O_id)

);

CREATE TABLE Role_

(

R_id INT NOT NULL,

R_name VARCHAR2(20) NOT NULL,

Description_ VARCHAR2(200) NOT NULL,

PRIMARY KEY (R_id)

);

CREATE TABLE Equipment

(

E_id INT NOT NULL,

Type_ VARCHAR2(20) NOT NULL,

Status INT NOT NULL,

PRIMARY KEY (E_id)

);

CREATE TABLE Personnel

(

P_id INT NOT NULL,

R_id INT NOT NULL,

P_name VARCHAR2(20) NOT NULL,

Rank_ VARCHAR2(20) NOT NULL,

מגשים: משה שחר ואהרן כץ

PRIMARY KEY (P_id),

FOREIGN KEY (R_id) REFERENCES Role_(R_id)

);

CREATE TABLE Submits

(

P_id INT NOT NULL,

I_id INT NOT NULL,

PRIMARY KEY (P_id, I_id),

FOREIGN KEY (P_id) REFERENCES Personnel(P_id),

FOREIGN KEY (I_id) REFERENCES Intelligence_reports(I_id)

);

CREATE TABLE Assigned_personnel

(

A_id INT NOT NULL,

P_id INT NOT NULL,

PRIMARY KEY (A_id, P_id),

FOREIGN KEY (A_id) REFERENCES Assignments(A_id),

FOREIGN KEY (P_id) REFERENCES Personnel(P_id)

);

CREATE TABLE Assigned_operation

(

A_id INT NOT NULL,

O_id INT NOT NULL,

PRIMARY KEY (A_id, O_id),

FOREIGN KEY (A_id) REFERENCES Assignments(A_id),

FOREIGN KEY (O_id) REFERENCES Operations_(O_id)

);

שלב הכנסת הנתונים:

דרך א - mockaroo:

עבור Operations:

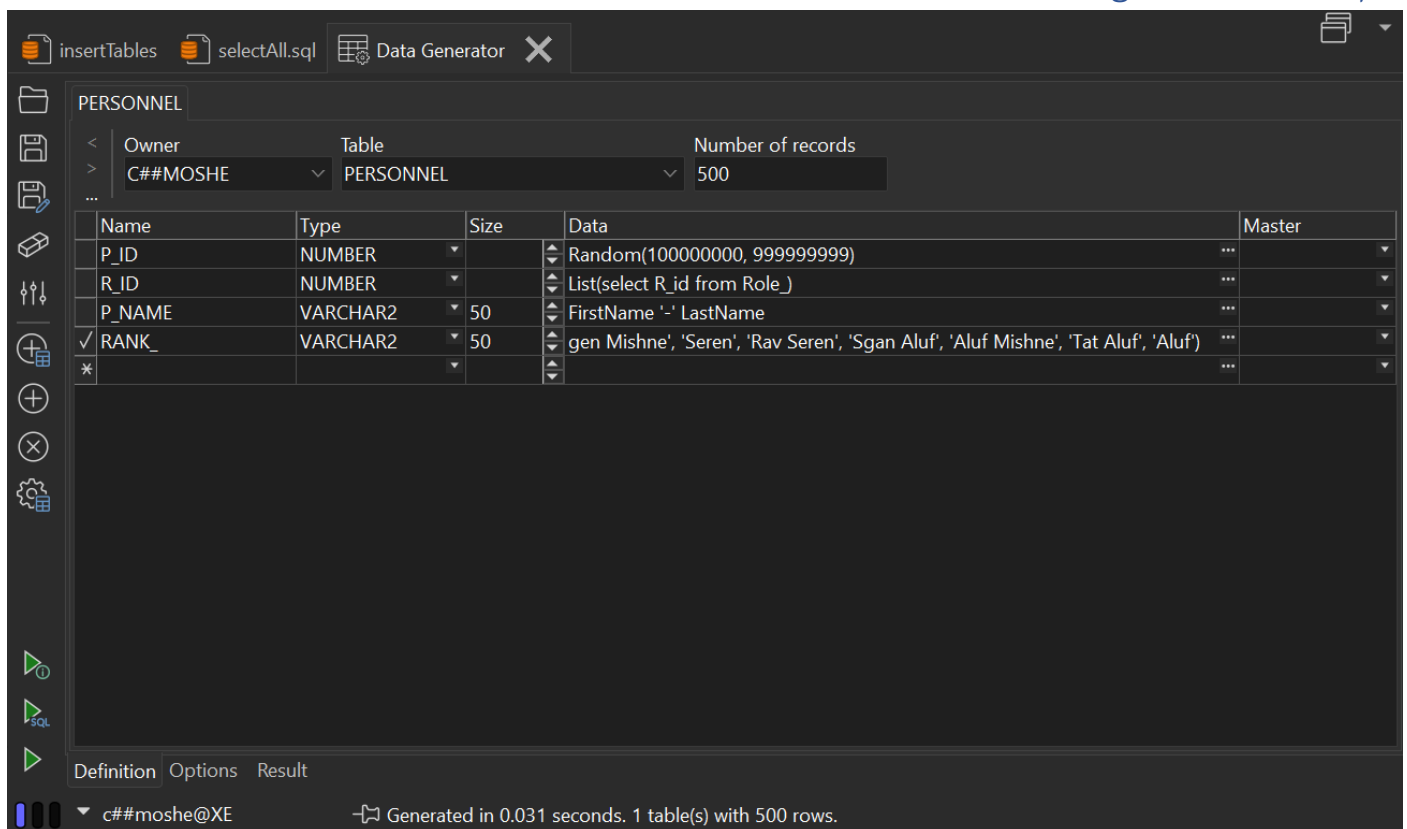
The screenshot shows the Mockaroo website interface. At the top, there's a navigation bar with links: SCHEMAS, DATASETS, MOCK APIS, SCENARIOS, PROJECTS, FUNCTIONS, a settings icon, a help icon, SIGN IN, and UPGRADE NOW. Below the navigation bar, there's a message: "Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats. 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." The main area is a form to configure the data generation. It has a table with columns: Field Name, Type, and Options. The table contains four rows: 1. Field Name: O_id, Type: Row Number, Options: blank: 0%, with a sum icon and a close icon. 2. Field Name: O_Name, Type: Buzzword, Options: blank: 0%, with a sum icon and a close icon. 3. Field Name: Start_date, Type: Datetime, Options: 05/26/1948 to 05/26/2000, format: dd/mm/yyyy, blank: 0%, with a sum icon and a close icon. 4. Field Name: End_date, Type: Datetime, Options: 05/26/2000 to 05/26/2024, format: dd/mm/yyyy, blank: 0%, with a sum icon and a close icon. Below the table, there are two buttons: "+ ADD ANOTHER FIELD" and "GENERATE FIELDS USING AI...". At the bottom, there are two input fields: "# Rows:" with the value "500" and "Format:" with the value "Excel".

והנה הפקודות שנוצרו:

The screenshot shows a SQL editor with a dark theme. The top bar has several tabs: "Operations_sql", "insertToPersonnel.gd", "insertToRole.gd", and "select * from personnel;". The "Operations_sql" tab is active. The editor shows a list of SQL queries, all starting with "insert into Operations_". Each query has a unique "O_id" and a unique "O_Name". The "Start_date" and "End_date" are also generated. The queries are as follows:

```
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (1, 'zero defect', TO_DATE('07/12/2019', 'dd/mm/yyyy'), TO_DATE('07/12/2019', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (2, 'internet solution', TO_DATE('07/12/2019', 'dd/mm/yyyy'), TO_DATE('07/12/2019', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (3, 'solution-oriented', TO_DATE('07/12/2019', 'dd/mm/yyyy'), TO_DATE('07/12/2019', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (4, 'De-engineered', TO_DATE('08/01/1984', 'dd/mm/yyyy'), TO_DATE('08/01/1984', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (5, 'portal', TO_DATE('11/03/1984', 'dd/mm/yyyy'), TO_DATE('11/03/1984', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (6, 'service-desk', TO_DATE('01/01/2000', 'dd/mm/yyyy'), TO_DATE('01/01/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (7, 'implementation', TO_DATE('22/01/2000', 'dd/mm/yyyy'), TO_DATE('22/01/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (8, 'Intuitive', TO_DATE('11/09/1999', 'dd/mm/yyyy'), TO_DATE('11/09/1999', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (9, 'conglomeration', TO_DATE('03/03/2000', 'dd/mm/yyyy'), TO_DATE('03/03/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (10, 'Diverse', TO_DATE('15/09/1977', 'dd/mm/yyyy'), TO_DATE('15/09/1977', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (11, 'multimedia', TO_DATE('18/09/2000', 'dd/mm/yyyy'), TO_DATE('18/09/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (12, 'Seamless', TO_DATE('27/08/1999', 'dd/mm/yyyy'), TO_DATE('27/08/1999', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (13, 'bottom-line', TO_DATE('15/04/2000', 'dd/mm/yyyy'), TO_DATE('15/04/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (14, 'transitional', TO_DATE('25/11/2000', 'dd/mm/yyyy'), TO_DATE('25/11/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (15, 'asynchronous', TO_DATE('16/01/2000', 'dd/mm/yyyy'), TO_DATE('16/01/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (16, 'Synchronised', TO_DATE('05/01/2000', 'dd/mm/yyyy'), TO_DATE('05/01/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (17, 'Triple-buffered', TO_DATE('12/01/2000', 'dd/mm/yyyy'), TO_DATE('12/01/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (18, 'knowledge base', TO_DATE('05/01/2000', 'dd/mm/yyyy'), TO_DATE('05/01/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (19, 'Cloned', TO_DATE('19/07/1977', 'dd/mm/yyyy'), TO_DATE('19/07/1977', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (20, 'contingency', TO_DATE('05/03/2000', 'dd/mm/yyyy'), TO_DATE('05/03/2000', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (21, 'algorithm', TO_DATE('01/08/1999', 'dd/mm/yyyy'), TO_DATE('01/08/1999', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (22, 'mobile', TO_DATE('29/08/1996', 'dd/mm/yyyy'), TO_DATE('29/08/1996', 'dd/mm/yyyy'));
insert into Operations_ (O_id, O_Name, Start_date, End_date) values (23, '15-line telephone', TO_DATE('12/01/2000', 'dd/mm/yyyy'), TO_DATE('12/01/2000', 'dd/mm/yyyy'));
```

דרך ב - data generator:



The screenshot shows the Oracle SQL Developer Data Generator tool. The window title is "Data Generator". It displays the "PERSONNEL" table selected, with 500 records generated. The table structure is shown in a table with columns: Name, Type, Size, Data, and Master. The "Data" column shows the generated values for each field.

Name	Type	Size	Data	Master
P_ID	NUMBER		Random(100000000, 999999999)	***
R_ID	NUMBER		List(select R_id from Role_)	***
P_NAME	VARCHAR2	50	FirstName '-' LastName	***
RANK_	VARCHAR2	50	gen Mishne', 'Seren', 'Rav Seren', 'Sgan Aluf', 'Aluf Mishne', 'Tat Aluf', 'Aluf')	***
*				***

Definition Options Result

c##moshe@XE Generated in 0.031 seconds. 1 table(s) with 500 rows.

השתמשנו ב generator של PLSQL על מנת להכניס נתונים לשאר הטבלאות.

דרך ג:

יצירה ע"י פייתון:

```
main.py ×
1 import random
2 import string
3
4 # Function to generate a random string for R_name and Description
  2 usages
5 def random_string(length):
6     letters = string.ascii_letters
7     return ''.join(random.choice(letters) for _ in range(length))
8
9 # Number of records to insert
10 num_records = 500
11
12 # Generate SQL insert queries
13 queries = []
14 for i in range(1, num_records + 1):
15     r_id = i
16     r_name = random_string(10) # Generate a random string of length 10 for R_name
17     description = random_string(20) # Generate a random string of length 20 for Description
18     query = f"INSERT INTO Role_ (R_id, R_name, Description) VALUES ({r_id}, '{r_name}', '{description}');"
19     queries.append(query)
20
21 # Write queries to a file
22 with open('insert_queries.sql', 'w') as f:
23     for query in queries:
24         f.write(query + '\n')
25
26 print("SQL insert queries have been generated and saved to insert_queries.sql.")
27
```

```

insert_queries.sql ×
1  INSERT INTO Role_ (R_id, R_name, Description) VALUES (1, 'gl
2  INSERT INTO Role_ (R_id, R_name, Description) VALUES (2, 'nF
3  INSERT INTO Role_ (R_id, R_name, Description) VALUES (3, 'sd
4  INSERT INTO Role_ (R_id, R_name, Description) VALUES (4, 'nZ
5  INSERT INTO Role_ (R_id, R_name, Description) VALUES (5, 'Eg
6  INSERT INTO Role_ (R_id, R_name, Description) VALUES (6, 'pm
7  INSERT INTO Role_ (R_id, R_name, Description) VALUES (7, 'pw
8  INSERT INTO Role_ (R_id, R_name, Description) VALUES (8, 'Co
9  INSERT INTO Role_ (R_id, R_name, Description) VALUES (9, 'Sq
10 INSERT INTO Role_ (R_id, R_name, Description) VALUES (10, 'B
11 INSERT INTO Role_ (R_id, R_name, Description) VALUES (11, 'K
12 INSERT INTO Role_ (R_id, R_name, Description) VALUES (12, 'V
13 INSERT INTO Role_ (R_id, R_name, Description) VALUES (13, 'W
14 INSERT INTO Role_ (R_id, R_name, Description) VALUES (14, 'J
15 INSERT INTO Role_ (R_id, R_name, Description) VALUES (15, 'J
16 INSERT INTO Role_ (R_id, R_name, Description) VALUES (16, 'g
17 INSERT INTO Role_ (R_id, R_name, Description) VALUES (17, 'J
18 INSERT INTO Role_ (R_id, R_name, Description) VALUES (18, 'D
19 INSERT INTO Role_ (R_id, R_name, Description) VALUES (19, 'i

```

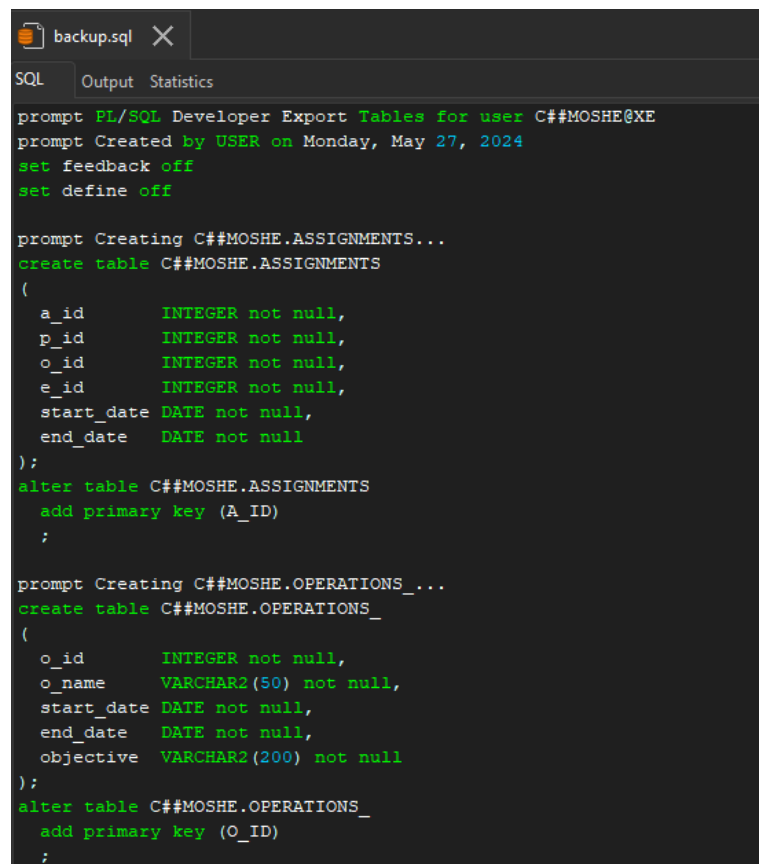
פקודת desc עבור personnel:

```

SQL> desc Personnel;
Name      Type      Nullable Default Comments
-----
P_ID      INTEGER
R_ID      INTEGER
P_NAME    VARCHAR2(50)
RANK_     VARCHAR2(50)

```

צילום מסך של קובץ הגיבוי:



```
backup.sql X
SQL Output Statistics
prompt PL/SQL Developer Export Tables for user C##MOSHE@XE
prompt Created by USER on Monday, May 27, 2024
set feedback off
set define off

prompt Creating C##MOSHE.ASSIGNMENTS...
create table C##MOSHE.ASSIGNMENTS
(
  a_id      INTEGER not null,
  p_id      INTEGER not null,
  o_id      INTEGER not null,
  e_id      INTEGER not null,
  start_date DATE not null,
  end_date  DATE not null
);
alter table C##MOSHE.ASSIGNMENTS
  add primary key (A_ID)
  ;

prompt Creating C##MOSHE.OPERATIONS_...
create table C##MOSHE.OPERATIONS_
(
  o_id      INTEGER not null,
  o_name    VARCHAR2(50) not null,
  start_date DATE not null,
  end_date  DATE not null,
  objective  VARCHAR2(200) not null
);
alter table C##MOSHE.OPERATIONS_
  add primary key (O_ID)
  ;
```