

מיני פרויקט בבסיסי נתונים

חיל המודיעין של צה"ל

תוכן עניינים

2	שלב 1
2	תיאור הארגון - מבוא :
2	ERD & DSD :
3	תרשים ERD של הארגון :
3	הטבלאות המתקבלות :
4	בדיקת נרמול :
4	דיאגרמת DSD :
5	פקודות create table :
7	שלב הכנסת הנתונים :
10	גיבוי ואחזור :
11	שלב 2
11	שאלות
17	שלב 3
17	תכנות

שלב 1

תיאור הארגון - מבוא:

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 יעוץ למפקדי צה"ל ולממשלה, בהקשר של תכנון מבצעי ואסטרטגי.

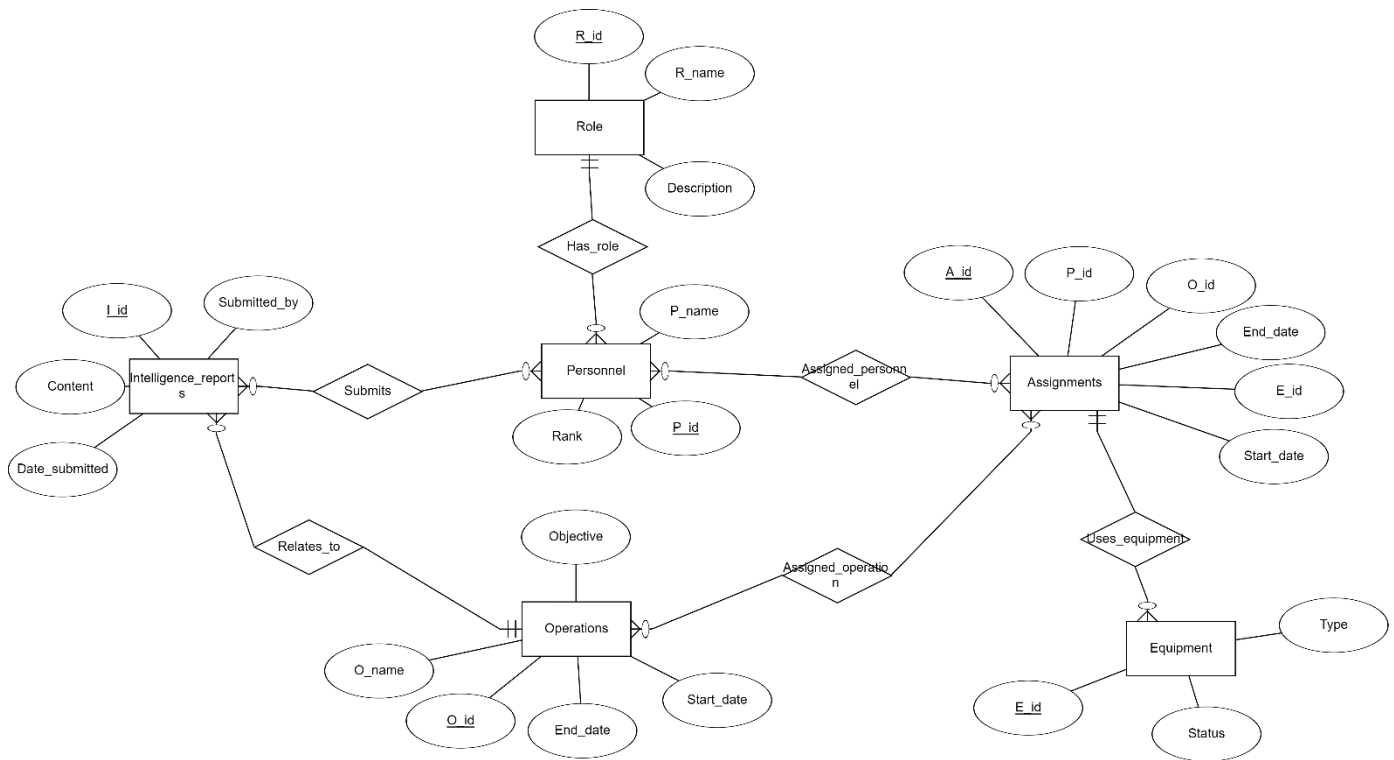
יחידות מרכזיות בחיל המודיעין:

- אמ"ן (אגף מודיעין ומחקר):** אחראי על איסוף וניתוח מודיעין אסטרטגי.
- יחידת 8200:** יחידה ללוחמה אלקטרונית ואיסוף מודיעין סיגני"ט.
- 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 הקצאת 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)

✓ **Equipment**: סוג הציוד והסטטוס תלויים במפתח E_id, ולכן מתקיים 3NF.

✓ **Personnel**: כל השדות תלויים במפתח P_id וכן אין שום תלות נוספת בין שני שדות שאינם מפתחות, ולכן

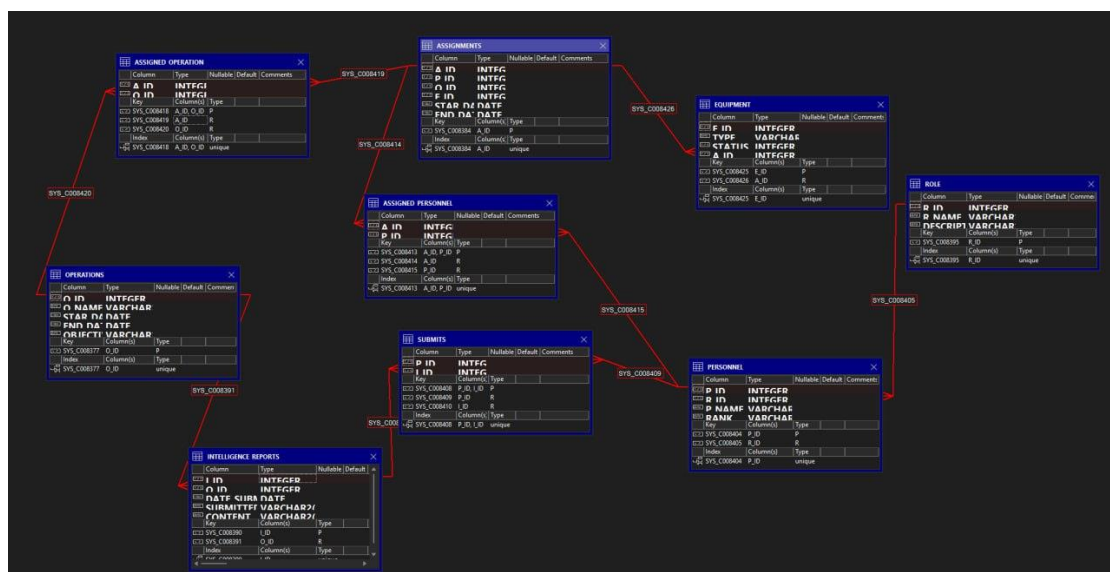
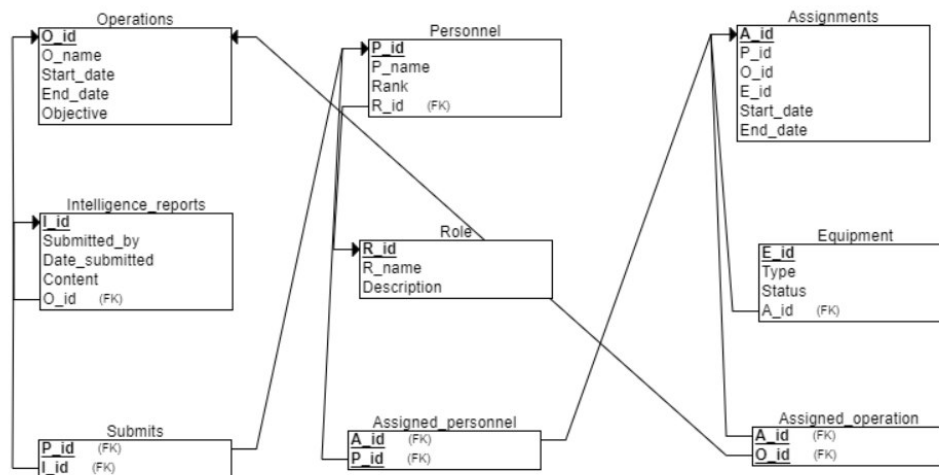
✓ **Operations:** השם והתאריכים תלויים במפתח O_id ואין תלות בניהם - מתקיים 3NF.

Intelligence_Reports ✓ לפי מספר הדו"ח נוכל לקבוע את כל שאר הפרטים כמו שם ותאריכים ולכן מתקיים

✓ **Assignments:** התלות היחידה שמתקיימת היא כל השדות במפתח A_id, ולכן מתקיים 3NF.

Submits, Assigned_personnel, Assigned_operation : בכולם יש רק 2 שדות ולכן מתקיים BCNF כלומר ✓

דיאגרמת DSD:



פקודות create table:

```
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,  
    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,  
    PRIMARY KEY (P_id),  
    FOREIGN KEY (R_id) REFERENCES Role_(R_id)  
);
```

```
CREATE TABLE Submits
```

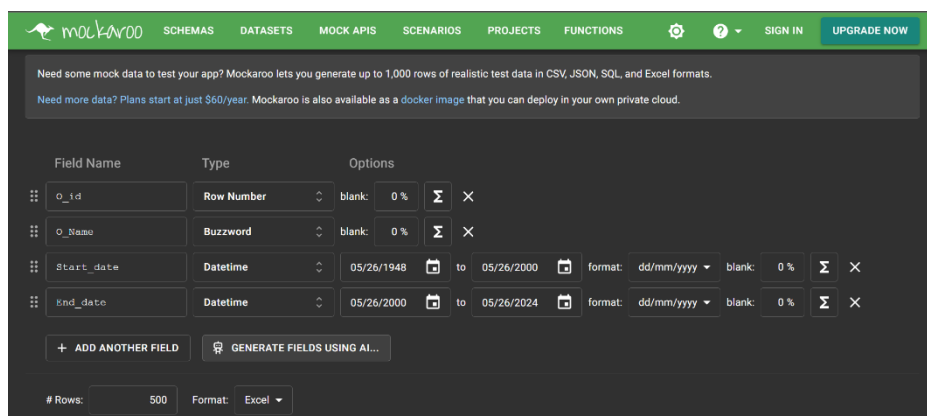
```
(  
    P_id INT NOT NULL,  
    I_id INT NOT NULL,  
    PRIMARY KEY (P_id, I_id),  
    FOREIGN KEY (P_id) REFERENCES Personnel(P_id),  
    FOREIGN KEY (I_id) REFERENCES Intelligence_reports(I_id)  
);
```

```
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:

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

עבור Operations:

להלן הפקודות
שנוצרו:

```

Operations_sql X insertToPersonnel.gd insertToRole.gd select * from personnel;
SQL Output Statistics
insert into Operations (O_id, O_Name, Start_date, End_date) values (1, 'zero defect', TO_DATE('07/12/1996', 'MM/DD/YYYY'), TO_DATE('07/12/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (2, 'internet solution', TO_DATE('07/12/1996', 'MM/DD/YYYY'), TO_DATE('07/12/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (3, 'solution-oriented', TO_DATE('07/12/1996', 'MM/DD/YYYY'), TO_DATE('07/12/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (4, 'De-engineered', TO_DATE('08/01/1996', 'MM/DD/YYYY'), TO_DATE('08/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (5, 'portal', TO_DATE('11/03/1984', 'MM/DD/YYYY'), TO_DATE('11/03/1984', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (6, 'service-desk', TO_DATE('01/01/1996', 'MM/DD/YYYY'), TO_DATE('01/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (7, 'implementation', TO_DATE('07/12/1996', 'MM/DD/YYYY'), TO_DATE('07/12/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (8, 'Intuitive', TO_DATE('11/09/1996', 'MM/DD/YYYY'), TO_DATE('11/09/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (9, 'conglomeration', TO_DATE('03/01/1996', 'MM/DD/YYYY'), TO_DATE('03/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (10, 'Diverse', TO_DATE('15/09/1997', 'MM/DD/YYYY'), TO_DATE('15/09/1997', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (11, 'multimedia', TO_DATE('18/09/1996', 'MM/DD/YYYY'), TO_DATE('18/09/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (12, 'Seamless', TO_DATE('27/08/1996', 'MM/DD/YYYY'), TO_DATE('27/08/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (13, 'bottom-line', TO_DATE('15/04/1996', 'MM/DD/YYYY'), TO_DATE('15/04/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (14, 'transitional', TO_DATE('25/11/1996', 'MM/DD/YYYY'), TO_DATE('25/11/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (15, 'asynchronous', TO_DATE('16/01/1996', 'MM/DD/YYYY'), TO_DATE('16/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (16, 'Synchronised', TO_DATE('05/01/1996', 'MM/DD/YYYY'), TO_DATE('05/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (17, 'Triple-buffered', TO_DATE('02/01/1996', 'MM/DD/YYYY'), TO_DATE('02/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (18, 'knowledge base', TO_DATE('05/01/1996', 'MM/DD/YYYY'), TO_DATE('05/01/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (19, 'Cloned', TO_DATE('19/07/1977', 'MM/DD/YYYY'), TO_DATE('19/07/1977', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (20, 'contingency', TO_DATE('05/03/1996', 'MM/DD/YYYY'), TO_DATE('05/03/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (21, 'algorithm', TO_DATE('01/08/1996', 'MM/DD/YYYY'), TO_DATE('01/08/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (22, 'mobile', TO_DATE('29/08/1996', 'MM/DD/YYYY'), TO_DATE('29/08/1996', 'MM/DD/YYYY'));
insert into Operations (O_id, O_Name, Start_date, End_date) values (23, 'mobile', TO_DATE('29/08/1996', 'MM/DD/YYYY'), TO_DATE('29/08/1996', 'MM/DD/YYYY'));

```

דרך ב - data generator של PLSQL:

The screenshot shows the 'Data Generator' window in PL/SQL Developer. The 'PERSONNEL' table is selected, and 500 records are generated. The table structure is as follows:

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'	...

At the bottom, it states: "Generated in 0.031 seconds. 1 table(s) with 500 rows."

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

דרך ג - יצירה ע"י פייתון:

```

main.py x
1 import random
2 import string
3
4 # Function to generate a random string for R_name and Description
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

```

להלן הפקודות שנוצרו:

```

insert_queries.sql x
1 INSERT INTO Role_ (R_id, R_name, Description) VALUES (1, 'gl
2 INSERT INTO Role_ (R_id, R_name, Description) VALUES (2, 'nE
3 INSERT INTO Role_ (R_id, R_name, Description) VALUES (3, 'sg
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, 'Sc
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

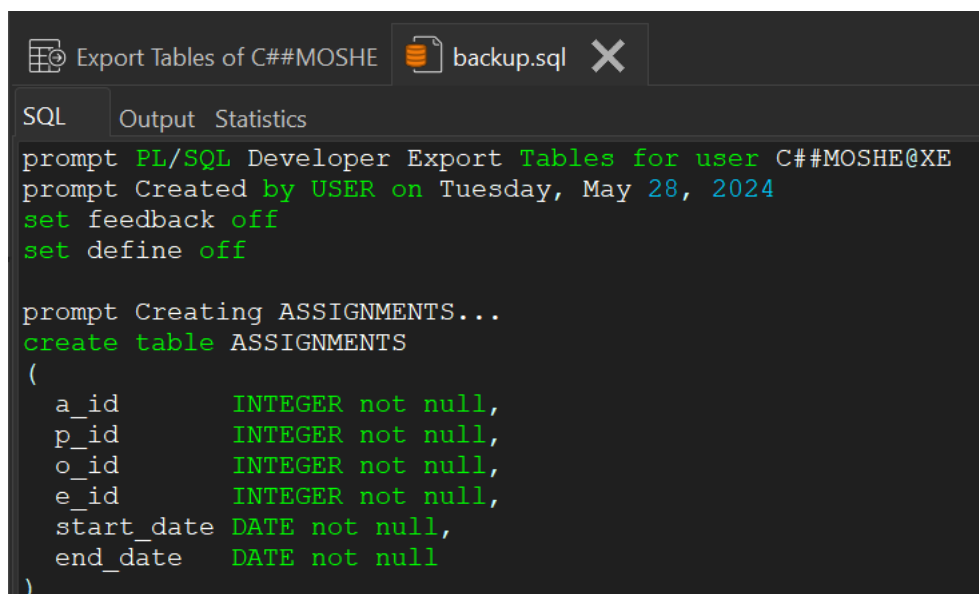
```

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)
```

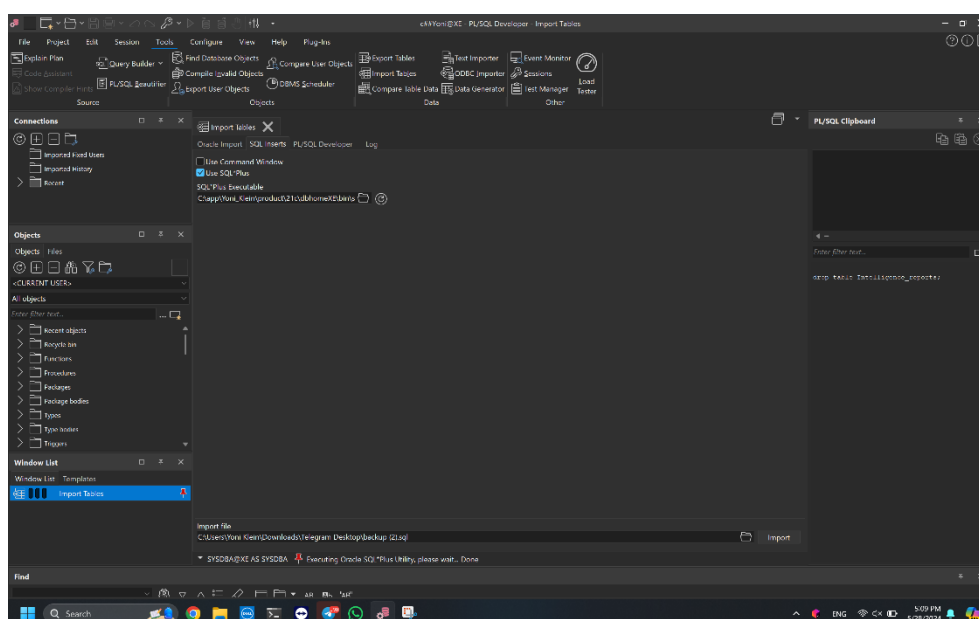
גיבוי ואחזור

צילום מסך של קובץ הגיבוי והשחזור:



```
Export Tables of C##MOSHE backup.sql X
SQL Output Statistics
prompt PL/SQL Developer Export Tables for user C##MOSHE@XE
prompt Created by USER on Tuesday, May 28, 2024
set feedback off
set define off

prompt Creating ASSIGNMENTS...
create table 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
)
```



שלב 2

שאלות

:SELECT

SQL Output Statistics

```
SELECT A.A_id, A.Start_date, A.End_date, O.O_name,
       (SELECT COUNT(*)
        FROM Assigned_personnel AP
        WHERE AP.A_id = A.A_id) AS Personnel_Count
FROM Assignments A
JOIN Assigned_operation AO ON A.A_id = AO.A_id
JOIN Operations_ O ON AO.O_id = O.O_id
ORDER BY A.A_id;
```

Select assigned_personnel Select intelligence_reports Select operations_ Select select

	A_ID	START_DATE	END_DATE	O_NAME	PERSONNEL_COUNT
1	3	10/10/1964	4/3/2000	stable	0
2	5	10/5/1964	4/4/2022	focus group	2
3	9	5/15/1980	9/1/2010	hierarchy	1
4	11	7/13/1965	9/28/2008	flexibility	0
5	11	7/13/1965	9/28/2008	budgetary management	0
6	15	7/12/1969	2/22/2018	upward-trending	1
7	15	7/12/1969	2/22/2018	asynchronous	1

רשימה של כל המשימות (מזהה, תאריך התחלה, תאריך סיום) יחד עם שמות הפעולות שהן חלק מהן. בנוסף, חישוב מספר הכולל של הצוותים שהוקצו לכל משימה.

שאלתה זו עוזרת לנהל ולעקוב אחר מטלות על ידי:

- 1. ספירת כוח אדם:** מראה כמה עובדים הוקצו לכל משימה.
- 2. פעולות קישור:** מחבר מטלות לפעולות שלהן בהתאמה לבהירות.
- 3. ניהול משאבים:** מסייע בהערכה והתאמת הקצאת כוח אדם ליעילות.

SQL Output Statistics

```
SELECT IR.Submitted_by, P.P_name,
       (SELECT COUNT(*)
        FROM Intelligence_reports IR2
        WHERE IR2.Submitted_by = IR.Submitted_by) AS Report_Count
FROM Intelligence_reports IR
JOIN Personnel P ON IR.Submitted_by = P.P_id
GROUP BY IR.Submitted_by, P.P_name
ORDER BY Report_Count DESC;
```

Select assigned_personnel Select intelligence_reports Select operations_ Select select

	SUBMITTED_BY	P_NAME	REPORT_COUNT
1	572376377	Dionne-Leguizamo	5
2	266057750	Christmas-Johnson	5
3	653091007	Elle-Lonsdale	4
4	532856593	Kim-Teng	4
5	304545484	Ann-Bright	4
6	702388971	Randall-Shawn	4
7	521806306	Jonathan-Watson	4

שאלתה זו מסכמת את ספירת הדוחות המודיעיניים שהוגשו על ידי כל צוות, ומציגה את זיהוי הצוות, השם והמספר הכולל של הדוחות שהם הגישו.

- 1. מעקב אחר ביצועים:** זיהוי כוח אדם פעיל.
- 2. הקצאת משאבים:** הבטחת חלוקה שווה של הגשת הדוחות.
- 3. תובנות תפעוליות:** מעקב ואופטימיזציה של זרימת דוחות המודיעין.

SQL	Output	Statistics																								
<pre> SELECT TO_CHAR(O.Start_date, 'YYYY-MM') AS Start_Month, (SELECT AVG(O2.End_date - O2.Start_date) FROM Operations_O2 WHERE TO_CHAR(O2.Start_date, 'YYYY-MM') = TO_CHAR(O.Start_date, 'YYYY-MM')) AS Avg_Duration_Days FROM Operations_O GROUP BY TO_CHAR(O.Start_date, 'YYYY-MM') ORDER BY Start_Month;</pre>																										
<div> Select assigned_personnel Select intelligence_reports Select operations_ Select select </div> <div> </div> <table> <thead> <tr> <th></th><th>START_MONTH</th><th>AVG_DURATION_DAYS</th></tr> </thead> <tbody> <tr><td>1</td><td>1948-07</td><td>19268</td></tr> <tr><td>2</td><td>1948-08</td><td>12639</td></tr> <tr><td>3</td><td>1948-09</td><td>23395</td></tr> <tr><td>4</td><td>1948-11</td><td>14987</td></tr> <tr><td>5</td><td>1949-01</td><td>26286</td></tr> <tr><td>6</td><td>1949-02</td><td>20317</td></tr> <tr><td>7</td><td>1949-05</td><td>18477</td></tr> </tbody> </table>				START_MONTH	AVG_DURATION_DAYS	1	1948-07	19268	2	1948-08	12639	3	1948-09	23395	4	1948-11	14987	5	1949-01	26286	6	1949-02	20317	7	1949-05	18477
	START_MONTH	AVG_DURATION_DAYS																								
1	1948-07	19268																								
2	1948-08	12639																								
3	1948-09	23395																								
4	1948-11	14987																								
5	1949-01	26286																								
6	1949-02	20317																								
7	1949-05	18477																								

שאלתה זו מחשבת את משך הפעולות הממוצע בימים. היא מחשבת את משך הזמן הממוצע עבור כל חודש על ידי השוואת תאריכי ההתחלה והסיום של הפעולות באותו חודש.

1. ניתוח זמן: הבנת משך הפעולה הממוצע לפי חודש.

2. מדדי ביצועים: זיהוי מגמות ותכנון משאבים בצורה יעילה יותר.

SQL

Output

Statistics

```
SELECT O_id, O_name, Start_date, End_date, Num_Assigned_Personnel
FROM (
    SELECT O.O_id, O.O_name, O.Start_date, O.End_date,
           COUNT(AP.P_id) AS Num_Assigned_Personnel
    FROM Operations_O
    JOIN Assigned_operation AO ON O.O_id = AO.O_id
    JOIN Assignments A ON AO.A_id = A.A_id
    JOIN Assigned_personnel AP ON A.A_id = AP.A_id
    GROUP BY O.O_id, O.O_name, O.Start_date, O.End_date
)T
ORDER BY Num_Assigned_Personnel DESC
FETCH FIRST 3 ROWS ONLY;
```

Select assigned_personnel

Select intelligence_reports

Select operations_

Select select

	O_ID	O_NAME	START_DATE	END_DATE	NUM_ASSIGNED_PERSONNEL
▶ 1	202	actuating	7/24/1954	11/11/2001	7
2	272	functionalities	5/6/1950	6/21/1992	6
3	369	forecast	11/12/1971	4/15/2011	6

שאלתה זו מאחזרת את 3 הפעולות המובילות עם המספר הגבוה ביותר של כוח אדם שהוקצה. היא עושה זאת על ידי קיבוץ תחילה של המשימות לפי פעולה וספירת מספר הצוותים שהוקצו לכל פעולה. לאחר מכן, היא בוחרת את פרטי הפעולה יחד עם ספירת הצוות שהוקצה וממיינת את הפעולות בסדר יורד, ומביאה רק את 3 התוצאות המובילות.

1. זיהוי פעולות מפתח: מציאה במהירות פעולות עם מעורבות כוח האדם הגבוהה ביותר.

2. הקצאת משאבים: עוזר להבין אילו פעולות דורשות את מירב המשאבים.

שאילתה זו מעדכנת את מצב הציוד ל'Taken' עבור כל הציוד שהוקצה.

```
UPDATE Equipment
SET Status = 'Taken'
WHERE E_id IN (SELECT E_id FROM Assignments);
```

```
SQL Output Statistics
```

```
select * from equipment;
```

	E_ID	TYPE_	STATUS
▶ 1	48952	kRs ...	Taken ...

אחרי

```
UPDATE Personnel
SET Rank_ = 'Samal'
WHERE P_id IN (
    SELECT P_id
    FROM Assigned_personnel
    GROUP BY P_id
    HAVING COUNT(A_id) > 3
) AND Rank_ = 'Turai';
```

```
SQL | Output | Statistics
```

```
UPDATE Personnel  
SET Rank_ = 'Samal'  
WHERE P_id IN (  
    SELECT P_id  
    FROM Assigned_personnel  
    GROUP BY P_id  
    HAVING COUNT(A_id) > 3  
) AND Rank_ = 'Turai';
```

```
select distinct * from Personnel p where p.p_name = 'Gailard-Thomson';
```

Update personnel | Select personnel

	P ID	R ID	P NAME	RANK
▶ 1	479692269	206	Gailard-Thomson	Samal

	P_ID	P_NAME	RANK	OPERATIONS_COUNT
1	563581797	Julia-Hopper	*** Sgan Aluf ***	8
2	932038464	Petula-Kilmer	*** Aluf ***	8
3	479692269	Gailard-Thomson	*** Turai ***	7
4	219371383	Brittany-Mantegna	*** Aluf Mishne ***	7
5	958699305	Derrick-Coyote	*** Aluf ***	7
6	789046930	Cesar-Aniston	*** Samal Rishon ***	7

לפני

:DELETE

```
DELETE FROM Intelligence_reports
WHERE Date_submitted < TO_DATE('08/04/1948', 'DD/MM/YYYY');
```

מחק את כל דוחות המודיעין שהוגשו לפני 08/04/1948.

להלן צילום של בסיס הנתונים לפני ואחרי העדכון:

SQL

Output















Statistics

```
DELETE FROM Intelligence_reports
WHERE Date_submitted < TO_DATE('08/04/1948', 'DD/MM/YYYY');

select * from intelligence_reports
where I_id = 0;
```

Delete intelligence_reports

Select intelligence_reports



I_ID

O_ID

SUBMITTED_BY

DATE_SUBMITTED

CONTENT

אחרי

SQL

Output

Statistics

```
select * from intelligence_reports
where I_id = 0;
```

לפני

```
DELETE FROM Equipment
WHERE E_id NOT IN (SELECT E_id FROM Assignments);
```

מחק את כל ערכי הציוד שאינם מוקצים כעת לאף משימה.

להלן צילום של בסיס הנתונים לפני ואחרי העדכון:

SQL

Output


Statistics


```
DELETE FROM Equipment
WHERE E_id NOT IN (SELECT E_id FROM Assignments);


select * from equipment
where E_ID = 1;
```


Delete equipment


Select equipment

 ▾














▾

















E_ID	TYPE	STATUS
------	------	--------

אחרי

SQL

Output Statistics

```
select * from equipment
where E_ID = 1;
```

לפני

:SELECT WITH PARAMS

מאחזרת רשימה של פעולות שמתחילות בתאריך התחלה מוגדר או אחריו ומסתיימות בתאריך סיום מוגדר או לפניו. היא ממיינת את התוצאות לפי תאריך ההתחלה של הפעולות.

SQL

```
SELECT
  O.O_id,
  O.O_name,
  O.Start_date,
  O.End_date,
  O.Objective
FROM
  Operations_ O
WHERE
  O.Start_date >= <name="Start Date" hint="Enter the start date in format DD/MM/YYYY" type="date">
  AND O.End_date <= <name="End Date" hint="Enter the end date in format DD/MM/YYYY" type="date">
ORDER BY
  O.Start_date;
```

	O_ID	O_NAME	START DATE	END DATE	OBJECTIVE
1	1	24/7	3/6/1959	12/13/2001	Object-based multi-state product
2	118	success	3/16/1959	4/6/1995	Up-sized logistical alliance
3	202	website	3/22/1959	1/5/1992	Persevering actuating workforce

מאחזרת רשימה של ציוד על סמך סטטוס מוגדר ("Taken" or "Available") היא ממיינת את התוצאות לפי מזהה הציוד.

SQL

```
SELECT
  E.E_id,
  E.Type,
  E.Status
FROM
  Equipment E
WHERE
  E.Status = <name="Equipment Status" list="Taken,Available" hint="Choose the status of the equipment">
ORDER BY
  E.E_id;
```

	E_ID	TYPE	STATUS
1	231	gUn	Taken
2	1222	uNu	Taken

מאחזרת רשימה של אנשי צוות בעלי דירוג מוגדר, ומצרף לטבלת Role כדי לכלול את שם התפקיד. היא ממיינת את התוצאות לפי מזהה כוח אדם.

SQL

```
SELECT
  P.P_id,
  P.P_name,
  P.Rank,
  R.R_name
FROM
  Personnel P
JOIN
  Role_R ON P.R_id = R.R_id
WHERE
  P.Rank = <name="Rank" hint="Enter the personnel rank" type="string">
ORDER BY
  P.P_id;
```

	P_ID	P_NAME	RANK	R NAME
1	169454937	Sammy-Ferrer	Aluf	CW-27
2	199263551	Cliff-Mac	Aluf	LQ-99
3	208052994	Queen-Overstreet	Aluf	ZS-28

מאחזרת רשימה של דוחות מודיעין שהוגשו על ידי אדם מסוים, וממיינת את התוצאות לפי תאריך הגשת הדוחות.

SQL

```
SELECT
  I.I_id,
  I.O_id,
  I.Submitted_by,
  I.Date_submitted,
  I.Content_
FROM
  Intelligence_reports I
WHERE
  I.Submitted_by = <name="Submitted By" hint="Enter the submitter's name" type="string">
ORDER BY
  I.Date_submitted;
```

	I_ID	O_ID	SUBMITTED_BY	DATE SUBMITTED	CONTENT
1	321	288	304545484	10/30/1973	XSzfvmVMeGbkaARWHY
2	1	88	304545484	9/21/2001	HHjGHfHXDFTSUeMPY
3	454	262	304545484	2/24/2023	ugylmCDaMNFfjjaH
4	30	53	304545484	3/31/2024	ruRGMjkPtbHZoZjgfl

:CONSTRAINTS

```

SQL | Output | Statistics
ALTER TABLE Personnel
ADD CONSTRAINT chk_rank CHECK (Rank_ IN ('Turai', 'Sgan Aluf', 'Samal Rishon', 'Aluf Mishne', 'Samal', 'Tat Aluf', 'Rasan', 'Segen Mishne', 'Aluf', 'Segen', 'Rabat', 'Seren', 'Rav Seren'));

ALTER TABLE Assignments
ADD CONSTRAINT chk_date CHECK (End_date > Start_date);

ALTER TABLE Equipment
MODIFY Status DEFAULT 'Available';

```

תיאור השינויים שנעשו בעזרת הפקודה :ALTER TABLE**Personnel:**

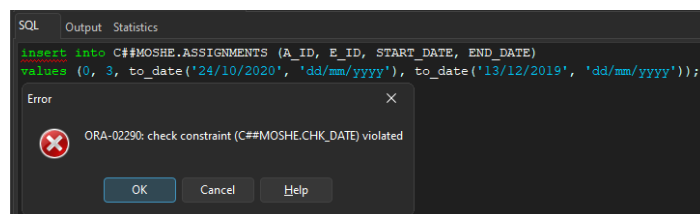
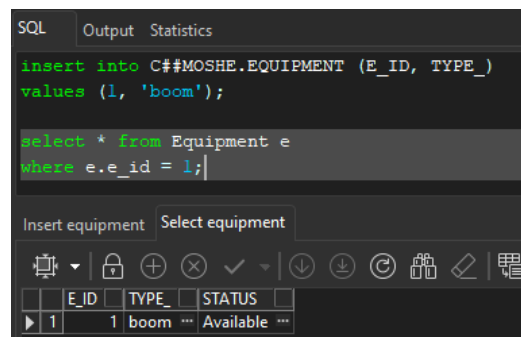
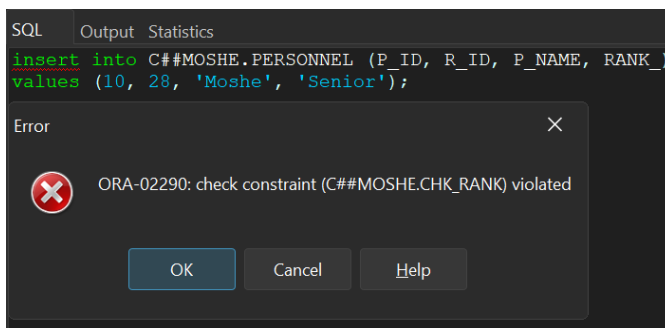
אילוץ זה מבטיח שהעמודה Rank_ בטבלת Personnel יכולה להכיל רק אחת מהדרגות הצבאיות שצוינו. זה עוזר לשמור על שלמות הנתונים על ידי הגבלת הערכים האפשריים לרשימה מוגדרת מראש.

Assignments:

אילוץ זה מבטיח ש- End_date בטבלה Assignments הוא תמיד מאוחר מ-Start_date. זה עוזר לשמור על עקביות לוגית על ידי הבטחה שלמשימות יש טווחי תאריכים חוקיים.

Equipment:

אילוץ זה מגדיר את ערך ברירת המחדל של העמודה סטטוס בטבלת הציוד ל-'Available' אם לא מסופק ערך. זה עוזר לפשט את הזנת הנתונים ומבטיח שהציוד מסומן כזמין כברירת מחדל אלא אם צוין אחרת.

להלן ניסיון הכנסת נתונים אשר סותרים את האילוצים ושיאוות הרצה שמתקבלות:

הערה: באילוץ על טבלת Equipment ניתן לראות שלא הוגדר סטטוס בזמן ההכנסה, ובכל זאת רשום 'Available'.

שלב 3

תכנות

:FUNCTIONS

```

create or replace function get_report_count_for_operation(p_operation_id in int)
return int is
    v_report_count int := 0;
    v_report_id intelligence_reports.i_id%type;
    cursor report_cursor is
        select i_id
        from intelligence_reports
        where o_id = p_operation_id;
begin
    open report_cursor;
    loop
        fetch report_cursor into v_report_id;
        exit when report_cursor%notfound;
        v_report_count := v_report_count + 1;
    end loop;
    close report_cursor;

    return v_report_count;
exception
    when no_data_found then
        dbms_output.put_line('no data found.');
```

get_report_count_for_operation מחשבת את ספירת דיווחי המודיעין עבור מבצע נתון. משתמשת ב cursor כדי להביא ולספור i_id מדוחות מודיעין שבהם o_id תואם ל- p_operation_id שסופק. מחזירה את הספירה או מטפלת בשגיאות על ידי החזרת 0 או -1.

```

create or replace function get_personnel_rank(p_personnel_id in int)
return varchar2 is
    v_rank personnel.rank_%type;
begin
    select rank_
    into v_rank
    from personnel
    where p_id = p_personnel_id;

    return v_rank;
exception
    when no_data_found then
        return 'unknown';
    when others then
        dbms_output.put_line('an unexpected error occurred.');
```

get_personnel_rank מחזירה את הדרגה של איש צוות על סמך תעודת הזהות שלו. אם לא נמצא נתונים, היא תחזיר 'unknown'. אם מתרחשת שגיאה אחרת, היא תדפיס את השגיאה ותחזיר 'error'.

:PROCEDURES

```

create or replace procedure update_equipment_status(
  p_equipment_ids in sys.odcinumberlist,
  p_new_status in varchar2,
  p_refcur out sys_refcursor
) is
begin
  if p_new_status not in ('Available', 'Taken') then
    raise_application_error(-20001, 'Invalid status value');
  else
    -- Update equipment statuses
    forall i in 1..p_equipment_ids.count
      update equipment
        set status = p_new_status
        where e_id = p_equipment_ids(i);

    -- Open ref cursor to fetch the updated equipment IDs
    open p_refcur for
      select e_id
      from equipment
      where e_id in (select column_value from table(p_equipment_ids))
        and status = p_new_status;
    commit;
  end if;
exception
  when others then
    dbms_output.put_line('An error occurred: ' || sqlerrm);
    if p_refcur%isopen then
      close p_refcur;
    end if;
    rollback;
    raise;
end;

```

update_equipment_status מעדכנת את סטטוס הציוד על סמך מזהי הציוד שסופקו וסטטוס חדש. פותחת ref cursor כדי לשחזר מזהי ציוד מעודכנים. מטפלת בשגיאות ומחזירה לאחר שיוניים ב database אם מתרחשות בעיות כלשהן.

```

create or replace procedure assign_personnel_to_assignment(p_assignment_id in int, p_personnel_id in int) is
type assignment_record is record (
  a_id assignments.a_id%type,
  p_id personnel.p_id%type
);
v_assignment assignment_record;
begin
  v_assignment.a_id := p_assignment_id;
  v_assignment.p_id := p_personnel_id;

  begin
    insert into assigned_personnel (a_id, p_id)
      values (v_assignment.a_id, v_assignment.p_id);
  exception
    when dup_val_on_index then
      dbms_output.put_line('personnel already assigned to this assignment.');
```

assign_personnel_to_assignment מקצה חבר צוות למשימה על ידי הוספת רשומה לטבלה assigned_personnel. אם הצוות כבר מוקצה, היא רושמת הודעה מתאימה. כל שגיאה בלתי צפויה מתועדת ומועלת מחדש. בסיום מתבצע שמירה ל database אם תצליח.

בסיס הנתונים לפני ואחרי הרצות תוכניות ה MAIN:

:main1

```

SQL Output Statistics
declare
v_operation_id int := 432;
v_report_count int;
v_equipment_ids sys.odcinumberlist;
v_new_status varchar2(20) := 'Taken';
v_equipment_refcur sys_refcursor;
v_e_id int;
begin
-- Fetch equipment IDs to be changed
select cast(collect(e_id) as sys.odcinumberlist)
into v_equipment_ids
from equipment
where status = 'Available';

-- Print equipment IDs before updating
dbms_output.put_line('Fetched equipment IDs:');
for i in 1..v_equipment_ids.count loop
    dbms_output.put_line('Equipment ID: ' || v_equipment_ids(i));
end loop;

-- Get report count for the operation
v_report_count := get_report_count_for_operation(v_operation_id);
dbms_output.put_line('Number of reports for operation ' || v_operation_id || ': ' || v_report_count);

-- Update equipment statuses
update_equipment_status(v_equipment_ids, v_new_status, v_equipment_refcur);

-- Print the equipment IDs that were updated to new status
loop
    dbms_output.put_line('List of updated equipment IDs:');
    fetch v_equipment_refcur into v_e_id;
    exit when v_equipment_refcur%notfound;
    dbms_output.put_line('Updated Equipment ID: ' || v_e_id);
end loop;
close v_equipment_refcur;

exception
when others then
    dbms_output.put_line('An error occurred: ' || sqlerrm);
end;
```

```

SQL Output Statistics
Clear Buffer size 10000 Enabled
Number of reports for operation 432: 4
List of updated equipment IDs:
Updated Equipment ID: 1
```

main1 לאחר הרצת get report count for operation ו update equipment status:

```

SQL Output Statistics
select * from equipment where e_id = 1
```

E_ID	TYPE	STATUS
1	boom	Taken

אחרי

```

SQL Output Statistics
select * from equipment where status = 'Available'
```

E_ID	TYPE	STATUS
1	boom	Available

לפני

:main2

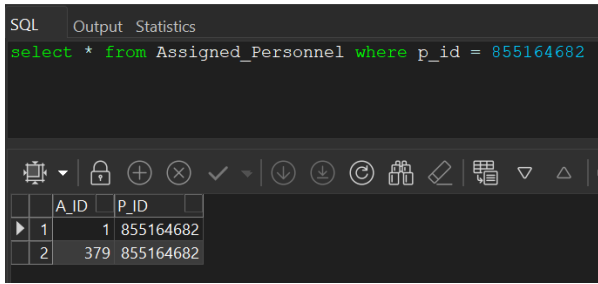
```

SQL Output Statistics
declare
v_personnel_id int := 855164682;
v_rank varchar2(50);
v_assignment_id int := 1;
begin
v_rank := get_personnel_rank(v_personnel_id);
dbms_output.put_line('personnel rank: ' || v_rank);

assign_personnel_to_assignment(v_assignment_id, v_personnel_id);
dbms_output.put_line('personnel assigned to assignment.');
```

```

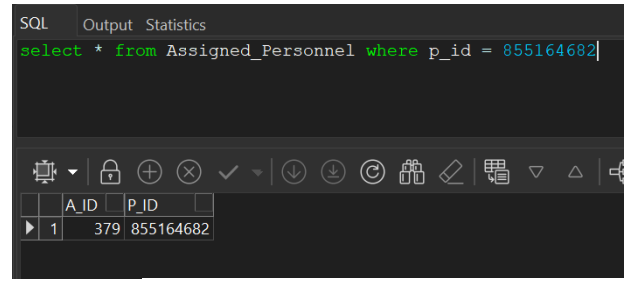
SQL Output Statistics
Clear Buffer size 10000 Enabled
personnel rank: Turai
personnel assigned to assignment.
```



The screenshot shows a SQL query interface with a dark theme. The query is `select * from Assigned_Personnel where p_id = 855164682`. Below the query, there is a toolbar with various icons. The results are displayed in a table with two columns: A_ID and P_ID. The first row has A_ID 1 and P_ID 855164682. The second row has A_ID 379 and P_ID 855164682.

	A_ID	P_ID
▶	1	855164682
	379	855164682

אחרי



The screenshot shows the same SQL query interface. The query is `select * from Assigned_Personnel where p_id = 855164682`. The results are displayed in a table with two columns: A_ID and P_ID. The first row has A_ID 1 and P_ID 855164682. The second row is empty.

	A_ID	P_ID
▶	1	855164682

לפני