פרויקט בבסיסי נתונים

חיל המודיעין של צה"ל

תוכן עניינים

שלב 1 2
עיצוב ויצירת DB 2
שלב 2 9
שאילתות 9
שלב 3 15
תכנות 15
שלב 4 18
אינטגרציה ומבטים 18

שלב 1

עיצוב ויצירת DB

עיצוב , בניה, אכלוס נתונים וגיבוי

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 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 8200 : יחידה ללוחמה אלקטרונית ואיסוף מודיעין סיגינ"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.

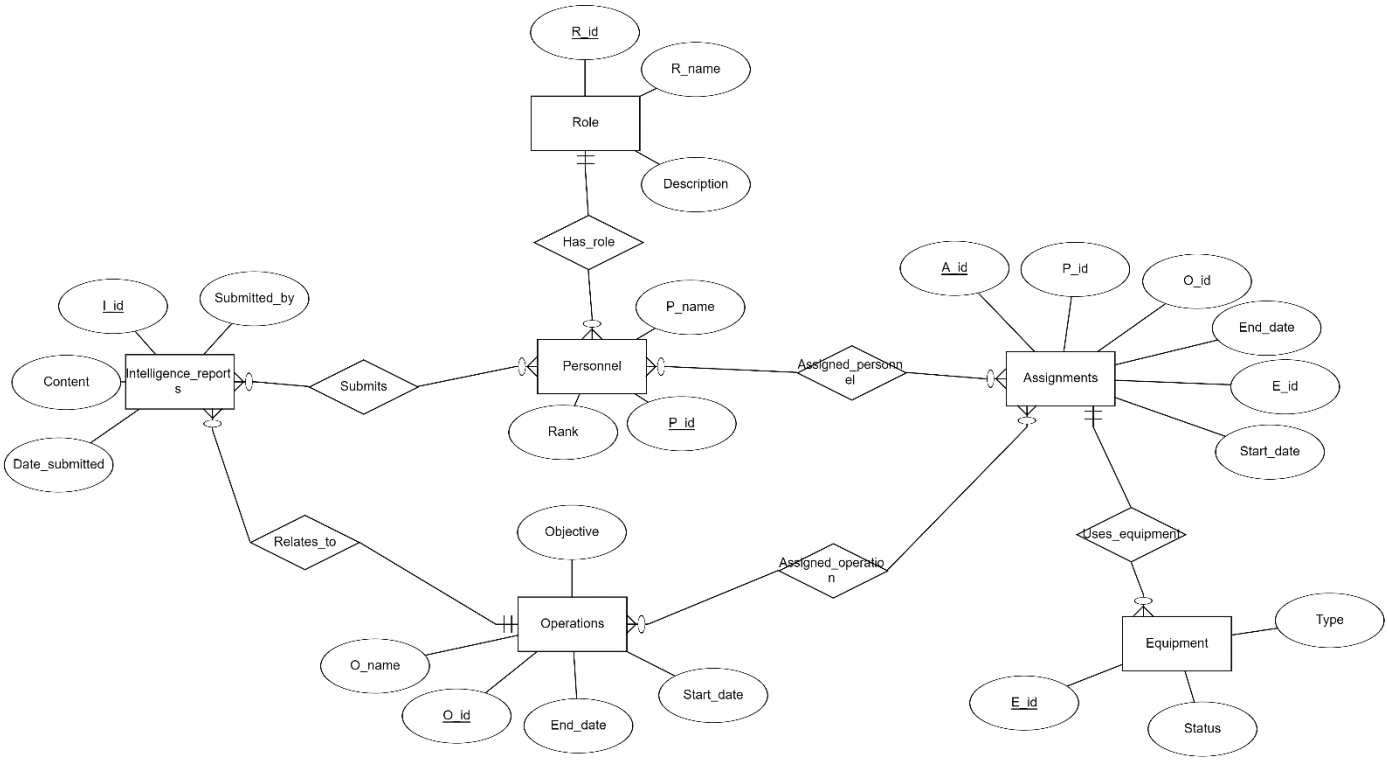
ניהול כוח אדם: כוח אדם הוא מרכזי במערכת, כל אחד מזוהה על ידי מזהה ייחודי, ויש להם תכונות כמו שם ודירוג. לעובדים מוקצים תפקידים בתוך הארגון, המתוארים בשמות התפקידים ובתיאוריהם.

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)

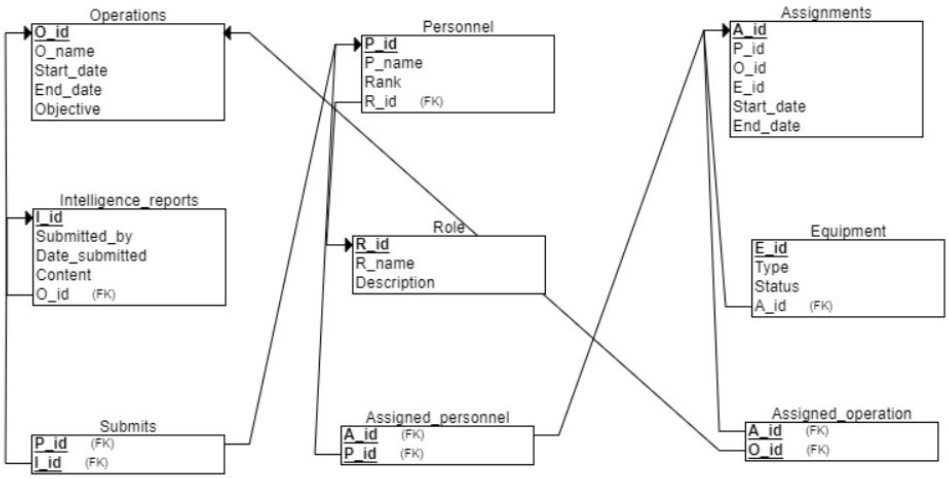
Assigned_personnel (A_id, P_id)

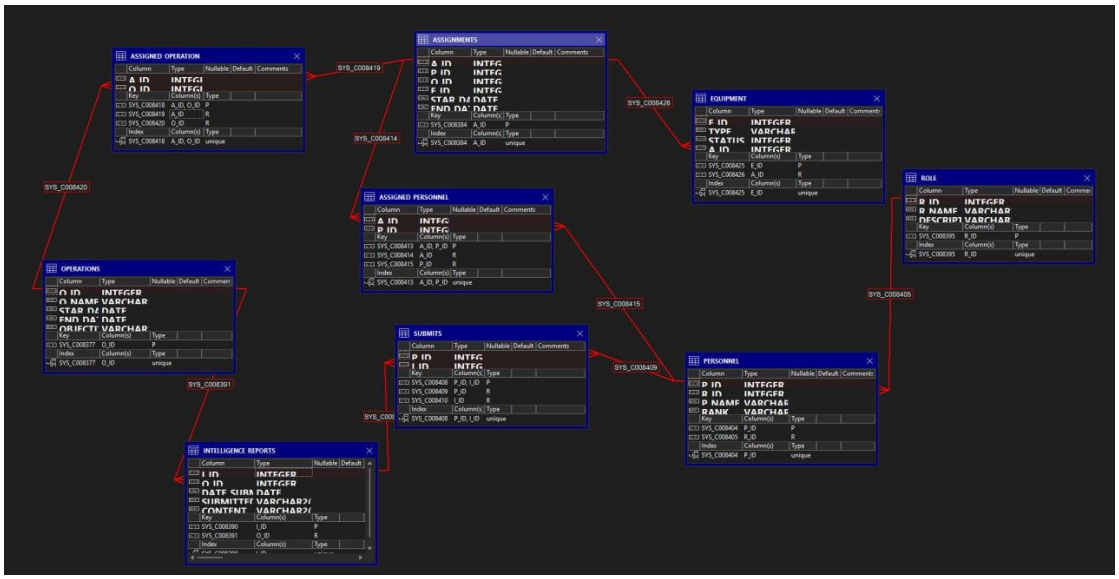
Submits (P_id, I_id)

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.
- ✓ **Assigned_operation, Assigned_personnel, Submits:** בכולם יש רק 2 שדות ולכן מתקיים BCNF כלומר מתקיים 3NF.

דיאגרמת DSD:





פקודות: CREATE TABLE

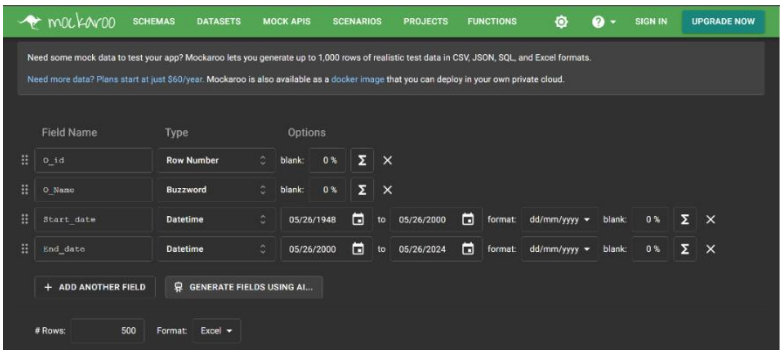
```
CREATE TABLE Operations_
(
    O_id INT NOT NULL,
    O_name VARCHAR2(20) NOT NULL,
    Star_date DATE NOT NULL,
    End_date DATE NOT NULL,
    Objective VARCHAR2(200) NOT NULL,
    PRIMARY KEY (O_id)
);

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 (Lid),
  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,
  L_id INT NOT NULL,
  PRIMARY KEY (P_id, l_id),
  FOREIGN KEY (P_id) REFERENCES Personnel (P_id),
  FOREIGN KEY (I_id) REFERENCES Intelligence_reports(l_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:

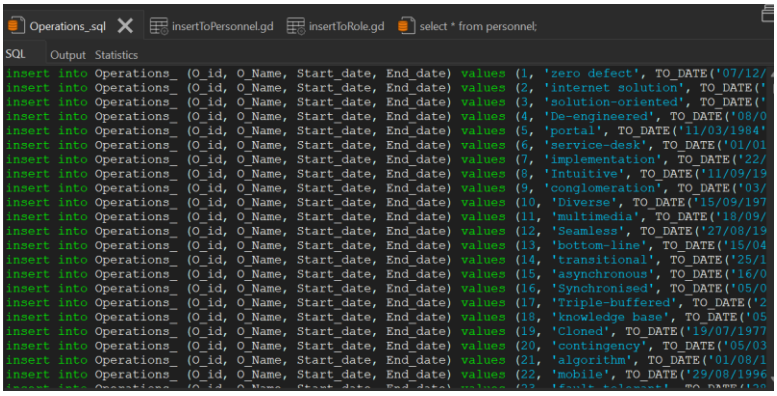
דרך א – mockaroo:

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

עבור Opertions:

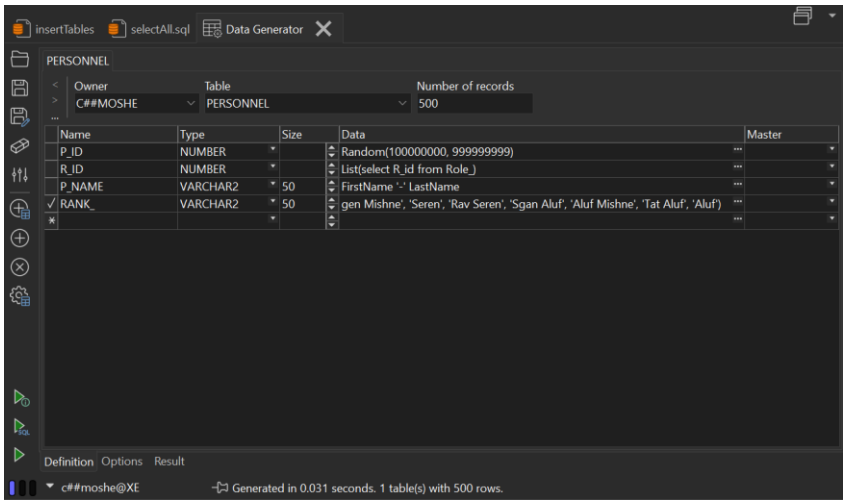


להלן הפקודות שנוצרו:



דרך ב - data generator של PLSQL:

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



דרך ג - יצירה ע"י פייתון:

```python
import random
import string

# Function to generate a random string for R_name and Description
def random_string(length):
    letters = string.ascii_letters
    return ''.join(random.choice(letters) for _ in range(length))

# Number of records to insert
num_records = 500

# Generate SQL insert queries
queries = []
for i in range(1, num_records + 1):
    r_id = i
    r_name = random_string(10)  # Generate a random string of length 10 for R_name
    description = random_string(20)  # Generate a random string of length 20 for Description
    query = f"INSERT INTO Role_ (R_id, R_name, Description) VALUES ({r_id}, '{r_name}', '{description}');"
    queries.append(query)

# Write queries to a file
with open('insert_queries.sql', 'w') as f:
    for query in queries:
        f.write(query + '\n')

print("SQL insert queries have been generated and saved to insert_queries.sql.")
```

להלן הפקודות שנוצרו:

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

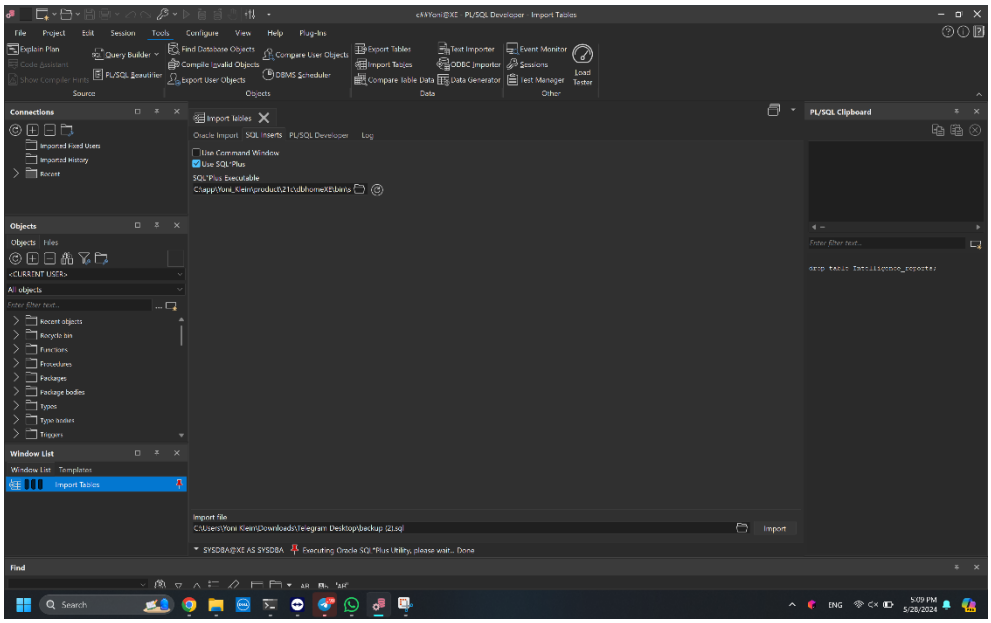
גיבוי ואחזור:

צילום מסך של קובץ הגיבוי והשחזור:

Export Tables of C##MOSHE | backup.sql

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

שאילתות

SELECT:

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.

```
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

שאילתא 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.

```
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

שאילתה 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.

שאילתה 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 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;
```

Select assigned_personnel | Select intelligence_reports | Select operations_ | Select select

	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

שאילתה 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.

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

UPDATE:

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

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

להלן צילום של בסיס הנתונים לפני ואחרי העדכון:

```
SQL  Output  Statistics
select * from equipment;
```

	E_ID	TYPE_	STATUS
1	48952	kRs	Taken

לפני

```
SQL  Output  Statistics
UPDATE Equipment
SET Status = 'Available'
WHERE E_id IN (SELECT E_id FROM Assignments);

select * from equipment;
```

Update equipment Select equipment

	E_ID	TYPE_	STATUS
1	48952	kRs	Available

אחרי

שאילתה זו מעדכנת את דרגת איש צוות מ'טוראי' ל'סמ"ל' אם הוא מוקצה ליותר משלוש משימות.

```
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';
```

להלן צילום של בסיס הנתונים לפני ואחרי העדכון:

	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

```
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

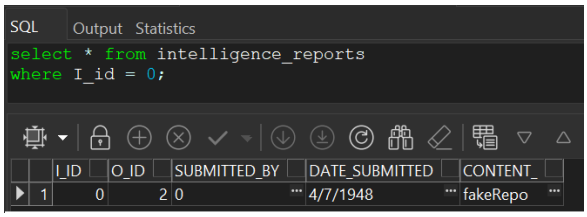
אחרי

DELETE:

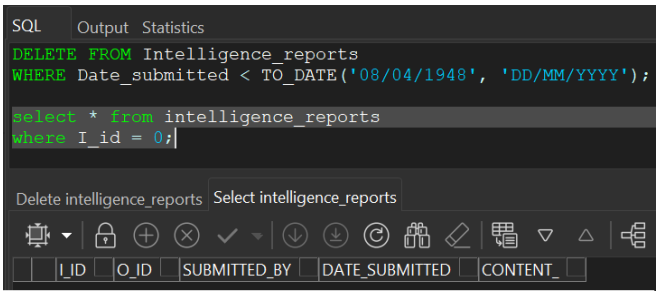
מחק את כל דוחות המודיעין שהוגשו לפני 08/04/1948.

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

להלן צילום של בסיס הנתונים לפני ואחרי העדכון:



לפני

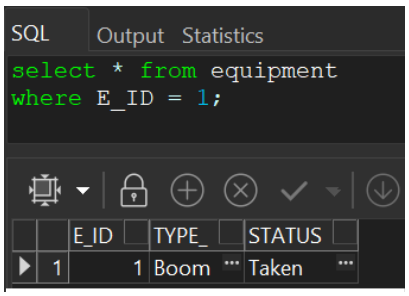


אחרי

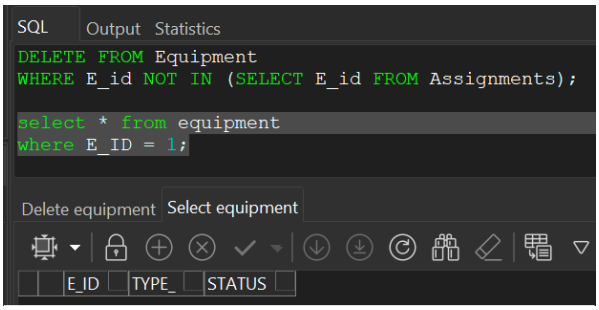
מחק את כל ערכי הציוד שאינם מוקצים כעת לאף משימה.

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

להלן צילום של בסיס הנתונים לפני ואחרי העדכון:



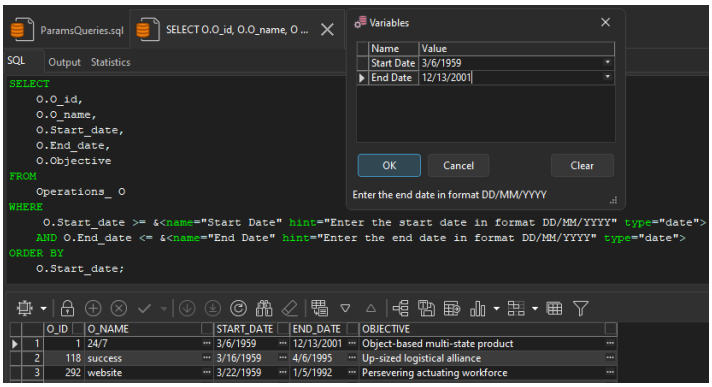
לפני



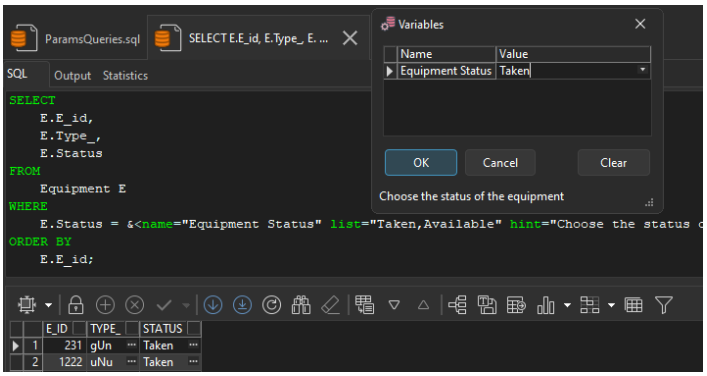
אחרי

SELECT WITH PARAMS:

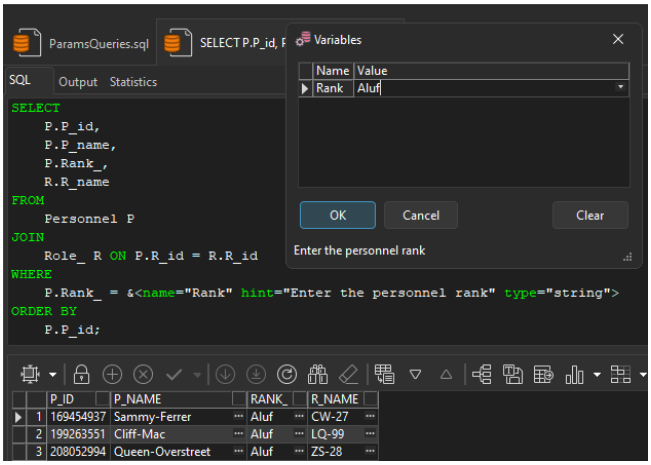
מאחזרת רשימה של פעולות שמתחילות בתאריך התחלה מוגדר או אחריו ומסתיימות בתאריך סיום מוגדר או לפניו. היא ממיינת את התוצאות לפי תאריך ההתחלה של הפעולות.



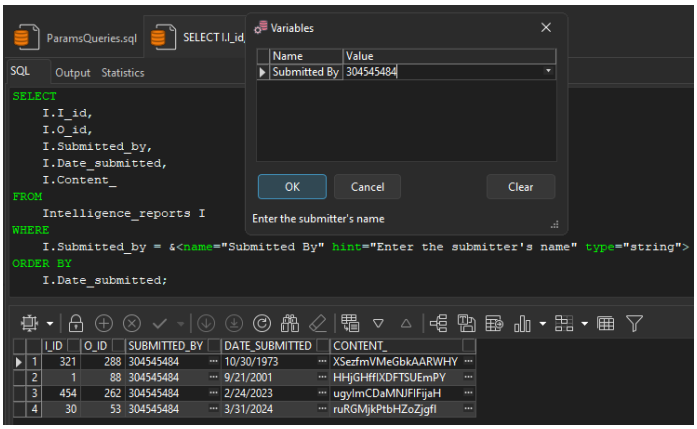
מאחזרת רשימה של ציוד על סמך סטטוס מוגדר ("Taken" or "Available"). היא ממיינת את התוצאות לפי מזהה הציוד.



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



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



CONSTRAINTS:

```
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';
```

<u>תיאור השינויים שנעשו בעזרת הפקודה ALTER TABLE:</u>

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.

<u>להלן ניסיון הכנסת נתונים אשר סותרים את האילוצים ושגיאות הרצה שמתקבלות:</u>

```
insert into C##MOSHE.PERSONNEL (P_ID, R_ID, P_NAME, RANK_)
values (10, 28, 'Moshe', 'Senior');
```

Error

ORA-02290: check constraint (C##MOSHE.CHK_RANK) violated

OK Cancel Help

```
insert into C##MOSHE.EQUIPMENT (E_ID, TYPE_)
values (1, 'boom');

select * from Equipment e
where e.e_id = 1;
```

Insert equipment | Select equipment

	E_ID	TYPE_	STATUS
1	1	boom	Available

```
insert into C##MOSHE.ASSIGNMENTS (A_ID, E_ID, START_DATE, END_DATE)
values (0, 3, to_date('24/10/2020', 'dd/mm/yyyy'), to_date('13/12/2019', 'dd/mm/yyyy'));
```

Error

ORA-02290: check constraint (C##MOSHE.CHK_DATE) violated

OK Cancel Help

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

שלב 3

תכנות

FUNCTIONS:

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.

```oracle_plsql
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.');
    return 0;
  when others then
    dbms_output.put_line('an unexpected error occurred.');
    return -1;
end;
```

get_personnel_rank מאחזרת 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'.

```oracle_plsql
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.');
    return 'error';
end;
```

PROCEDURES:

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 לאחור שינויים ב database אם מתרחשות בעיות כלשהן.

```oracle
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;
```

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.

```oracle
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.');
    when others then
      dbms_output.put_line('an unexpected error occurred.');
      raise;
  end;

  commit;
exception
  when others then
    rollback;
    raise;
end;
```

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
  dbms_output.put_line('List of updated equipment IDs:');
  loop
    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 status = 'Available'
```

	E_ID	TYPE_	STATUS
1	1	boom	Available

לפני

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

	E_ID	TYPE_	STATUS
1	1	boom	Taken

אחרי

main2:

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

```
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.');
exception
  when others then
    dbms_output.put_line('an error occurred.');
end;
```

main2 לאחר הרצת get_personnel_rank ו assign_personnel_to_assignment:

```
SQL  Output  Statistics
select * from Assigned_Personnel where p_id = 855164682
```

	A_ID	P_ID
1	1	855164682
2	379	855164682

לפני

```
SQL  Output  Statistics
select * from Assigned_Personnel where p_id = 855164682
```

	A_ID	P_ID
1	379	855164682

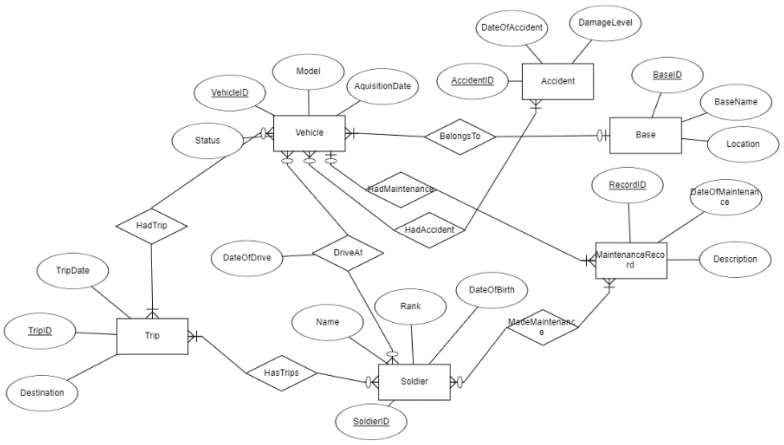
אחרי

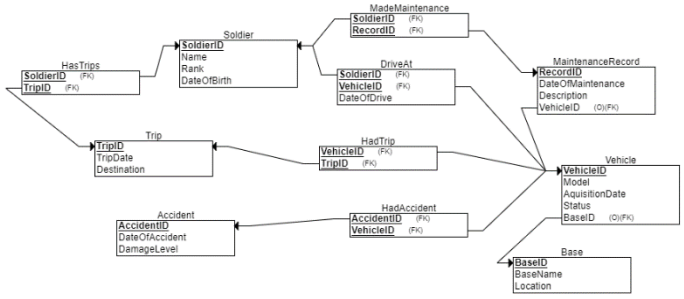
שלב 4

אינטגרציה ומבטים

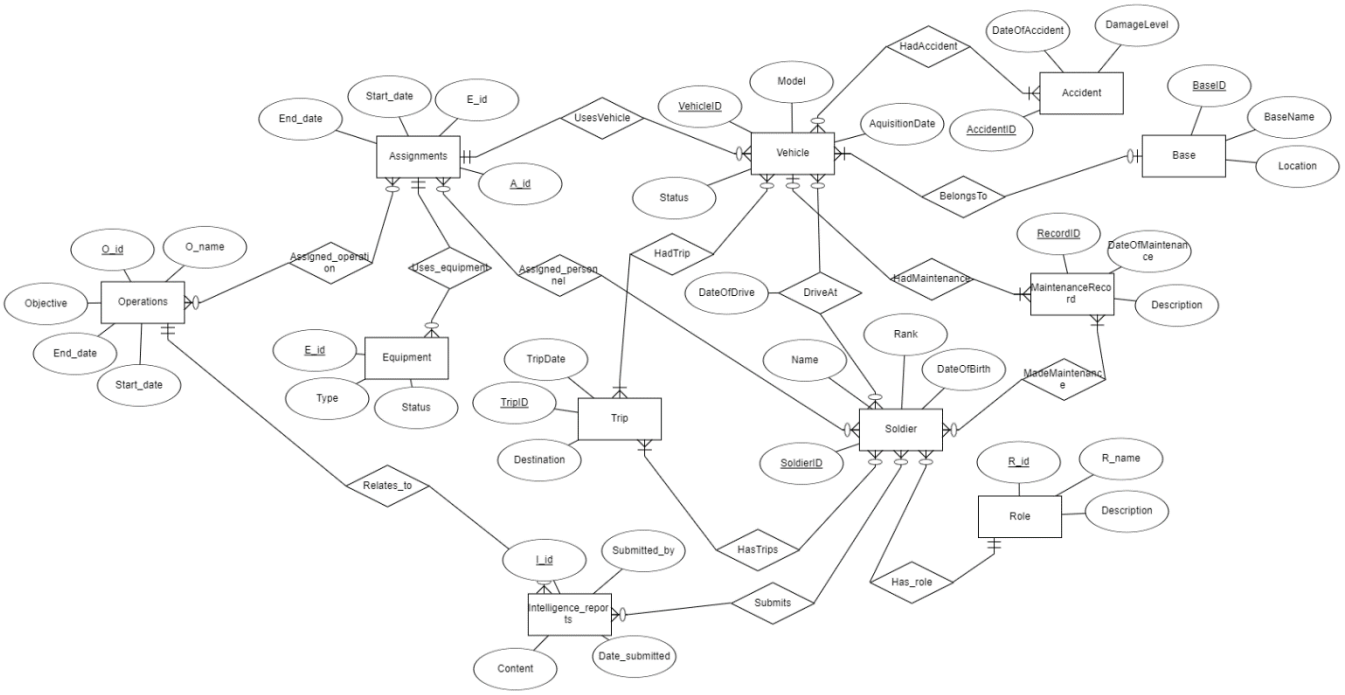
תמונות מסך של תרשימי ה DSD ERD:

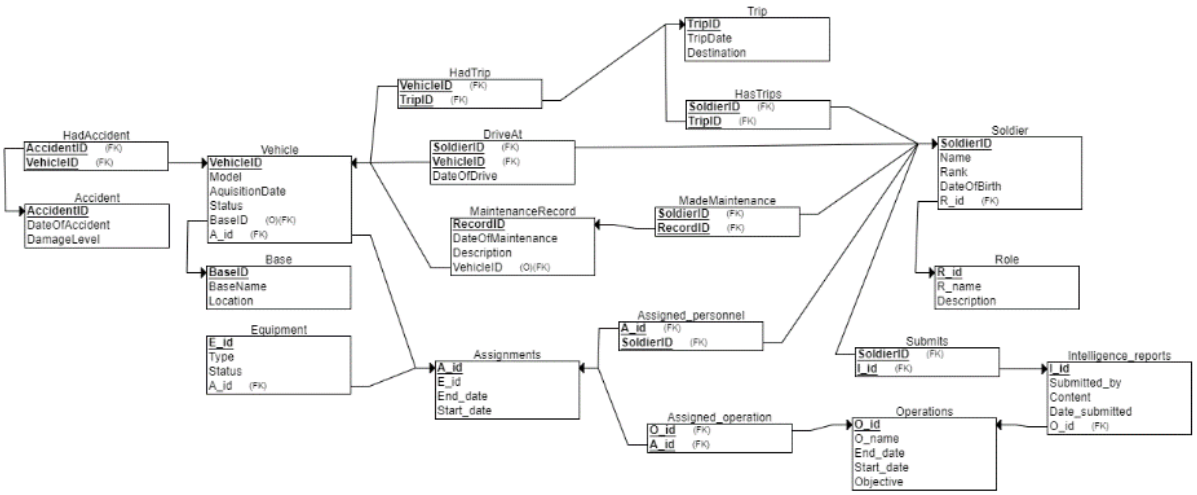
מאגר חדש - ERD,DSD





מאגר משותף - ERD,DSD





החלטות שנעשו בשלב האינטגרציה:

חיבור טבלת הרכב:

החלטתנו לחבר את טבלת Vehicle מהמסד החדש לטבלת Assignment במסד הנתונים הקיים שלנו. החלטה זו נבחרה כדי לקשר את נתוני הרכב עם רישומי המשימות, ובכך לשפר את היכולת לעקוב אחר אילו כלי רכב שימשו במשימות שונות ולהבטיח תצוגה מקיפה של ניצול הרכב על פני המשימות

שינוי שם טבלה והוספת עמודה:

הטבלה המקורית בשם Personnel במסד הנתונים שלנו התייחסה לאותה ישות כמו טבלת Soldier במסד הנתונים החדש. כדי לייעל את האינטגרציה ולהבטיח עקביות, שינינו את שם טבלת ה- Personnel ל Soldier כדי להתאים לטבלת מסד הנתונים החדשה. בנוסף, הוספנו עמודת תאריך לידה לטבלת החייל כדי ליישר קו עם מבנה מסד הנתונים החדש ולשפר את שלמות המידע על כוח האדם.

הסבר על התהליך והפקודות:

<u>פקודות ושינויים:</u>

יצירת טבלה: יצרנו טבלאות חדשות (Base, Trip, Accident, HasTrips, Vehicle, MaintenanceRecord, DriveAt, HadTrip, MadeMaintenance, HadAccident) כדי לשלב את מסד הנתונים החדש עם הסכימה הקיימת שלנו. טבלאות אלו נועדו לנהל נתונים נוספים הקשורים לבסיסים, נסיעות, תאונות, כלי רכב ורישומי תחזוקה.

<u>עדכון טבלת כוח אדם:</u>

הוספת עמודה: הוספנו עמודת DateOfBirth לטבלת ה-Personal כדי ליישר קו עם הסכימה החדשה.

שינוי שם טבלה: שינינו את שם טבלת כוח האדם ל Soldier לצורך עקביות.

שנוי שם עמודות: שינה P_id ל-SoldierID ו-P_name ל-Name_ כדי להתאים למוסכמות של מסד הנתונים החדש.

יצירת תאריכי לידה: השתמשנו ב-PL/SQL כדי לאכלס את העמודה החדשה של תאריך לידה בתאריכים אקראיים עבור רשומות קיימות.

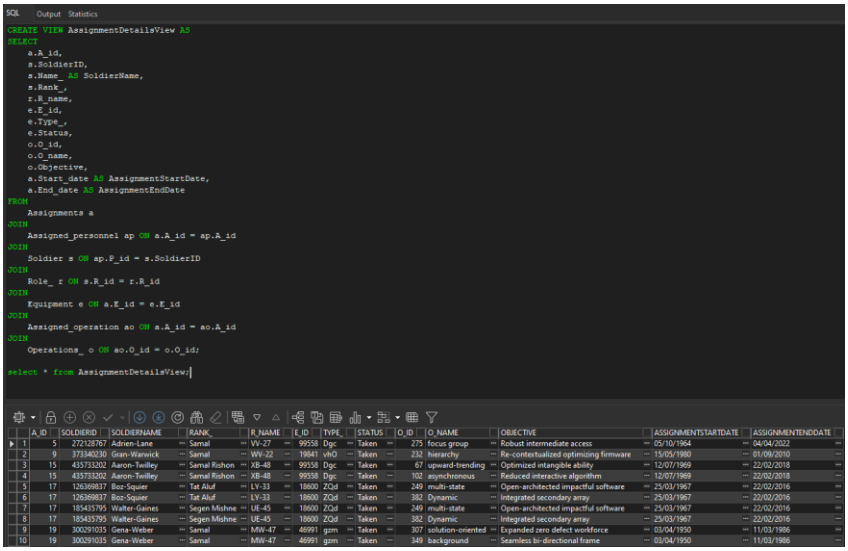
פקודות אלו מבטיחות תאימות ושילוב נכון של רכיבי מסד הנתונים החדשים עם המערכת הקיימת שלנו.

תיאור על המבטים החדשים ושאילתות:

מאגר מקורי

AssignmentDetailsView:

טבלת view זו מאגדת מידע מפורט על מטלות, כולל פרטי חייל, תפקידים, ציוד ופעולות. היא מצרפת את טבלאות Assignments, Assigned_personnel, Soldier, Role_, Equipment, Assigned_operation, ו-Operations_ כדי לספק תצוגה מקיפה של כל משימה, עם מידע על החיילים, תפקידיהם, הציוד בהם נעשה שימוש והפעולות הנלוות.

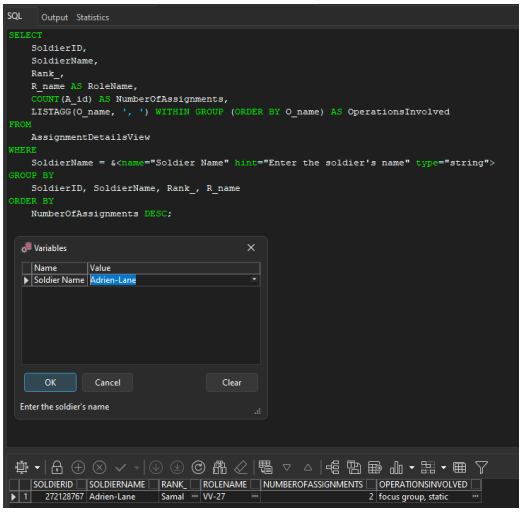


```
CREATE VIEW AssignmentDetailsView AS
SELECT
    a.A_id,
    s.SoldierID,
    s.Name_ AS SoldierName,
    s.Rank_,
    r.R_name,
    e.E_id,
    e.Type_,
    e.Status,
    o.O_id,
    o.O_name,
    o.Objective,
    a.Start_date AS AssignmentStartDate,
    a.End_date AS AssignmentEndDate
FROM
    Assignments a
JOIN
    Assigned_personnel ap ON a.A_id = ap.A_id
JOIN
    Soldier s ON ap.P_id = s.SoldierID
JOIN
    Role_ r ON s.R_id = r.R_id
JOIN
    Equipment e ON a.E_id = e.E_id
JOIN
    Assigned_operation ao ON a.A_id = ao.A_id
JOIN
    Operations_ o ON ao.O_id = o.O_id;

select * from AssignmentDetailsView;
```

	A_ID	SOLDIERID	SOLDIERNAME	RANK_	R_NAME	E_ID	TYPE_	STATUS	O_ID	O_NAME	OBJECTIVE	ASSIGNMENTSTARTDATE	ASSIGNMENTENDDATE
1	5	272128767	Adrien-Lane	Samal	VV-27	99558	Dgc	Taken	275	focus group	Robust intermediate access	05/10/1964	04/04/2022
2	9	373340230	Gran-Warwick	Samal	WV-22	19841	vhO	Taken	232	hierarchy	Re-contextualized optimizing firmware	15/05/1980	01/09/2010
3	15	435733202	Aaron-Twilley	Samal Rishon	XB-48	99558	Dgc	Taken	67	upward-trending	Optimized intangible ability	12/07/1969	22/02/2018
4	15	435733202	Aaron-Twilley	Samal Rishon	XB-48	99558	Dgc	Taken	102	asynchronous	Reduced interactive algorithm	12/07/1969	22/02/2018
5	17	126369837	Boz-Squier	Tat Aluf	LY-33	18600	ZQd	Taken	249	multi-state	Open-architected impactful software	25/03/1967	22/02/2016
6	17	126369837	Boz-Squier	Tat Aluf	LY-33	18600	ZQd	Taken	382	Dynamic	Integrated secondary array	25/03/1967	22/02/2016
7	17	185435795	Walter-Gaines	Segen Mishne	UE-45	18600	ZQd	Taken	249	multi-state	Open-architected impactful software	25/03/1967	22/02/2016
8	17	185435795	Walter-Gaines	Segen Mishne	UE-45	18600	ZQd	Taken	382	Dynamic	Integrated secondary array	25/03/1967	22/02/2016
9	19	300291035	Gena-Weber	Samal	MW-47	46991	gzm	Taken	307	solution-oriented	Expanded zero defect workforce	03/04/1950	11/03/1986
10	19	300291035	Gena-Weber	Samal	MW-47	46991	gzm	Taken	349	background	Seamless bi-directional frame	03/04/1950	11/03/1986

שאילתות

שאילתה לרשימת משימות ופעולות של חייל: שאילתה זו מאחזרת את מספר המטלות הכולל של כל חייל ואת הפעולות בהן היה מעורב. היא מקבצת את הנתונים לפי זיהוי חייל, שם, דרגה ותפקיד, ולאחר מכן סופרת את מספר המטלות ומפרטת את הנתונים.
השאילתה תסנן לפי שם חייל ספציפי הניתן כפרמטר ומסדרת את התוצאות לפי מספר המטלות בסדר יורד.



```
SELECT
    SoldierID,
    SoldierName,
    Rank_,
    R_name AS RoleName,
    COUNT(A_id) AS NumberOfAssignments,
    LISTAGG(O_name, ', ') WITHIN GROUP (ORDER BY O_name) AS OperationsInvolved
FROM
    AssignmentDetailsView
WHERE
    SoldierName = &<name="Soldier Name" hint="Enter the soldier's name" type="string">
GROUP BY
    SoldierID, SoldierName, Rank_, R_name
ORDER BY
    NumberOfAssignments DESC;
```

Variables

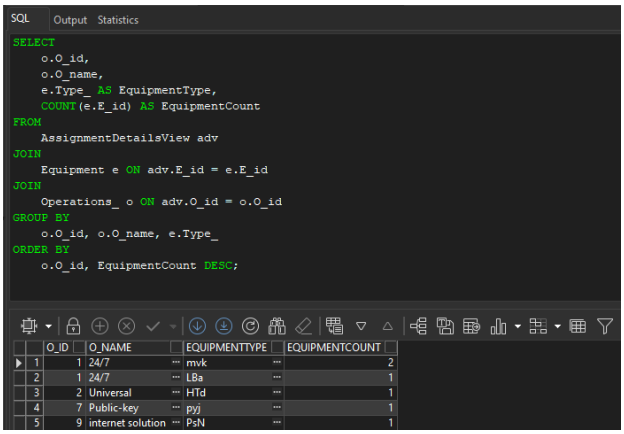
Name	Value
Soldier Name	Adrien-Lane

OK Cancel Clear

Enter the soldier's name

	SOLDIERID	SOLDIERNAME	RANK_	ROLENAME	NUMBEROFASSIGNMENTS	OPERATIONSINVOLVED
1	272128767	Adrien-Lane	Samal	VV-27	2	focus group, static

שאילתה לספירת סוגי ציוד המשמשים בפעולות: שאילתה זו מאחזרת את הספירה של כל סוג ציוד המשויך לפעולות שונות. היא מקבצת את התוצאות לפי מזהה פעולה, שם וסוג הציוד, וסופרת את מספר פריטי הציוד עבור כל סוג.
השאילתה מסדרת את התוצאות לפי מזהה הפעולה וספירת הציוד בסדר יורד.



```
SELECT
    o.O_id,
    o.O_name,
    e.Type_ AS EquipmentType,
    COUNT(e.E_id) AS EquipmentCount
FROM
    AssignmentDetailsView adv
JOIN
    Equipment e ON adv.E_id = e.E_id
JOIN
    Operations_ o ON adv.O_id = o.O_id
GROUP BY
    o.O_id, o.O_name, e.Type_
ORDER BY
    o.O_id, EquipmentCount DESC;
```

	O_ID	O_NAME	EQUIPMENTTYPE	EQUIPMENTCOUNT
1	1	24/7	mvk	2
2	1	24/7	LBa	1
3	2	Universal	HTd	1
4	7	Public-key	pyj	1
5	9	internet solution	PsN	1

מאגר חדש

IntegratedDetailsView: טבלת view זו משלבת נתונים מנסיעות, כלי רכב, בסיסים, תאונות ורישומי תחזוקה. היא מספקת סקירה מקיפה על ידי קישור נסיעות לכלי רכב, המחוברים עוד יותר לבסיסים, תאונות ותחזוקה שלהם. היא כוללת פרטים כגון מידע על נסיעה, דגם ומצב רכב, מיקום בסיס, פרטי תאונות ורישומי תחזוקה.

```
CREATE VIEW IntegratedDetailsView AS
SELECT
    t.TripID,
    t.TripDate,
    t.Destination,
    v.VehicleID,
    v.Model_ AS VehicleModel,
    v.AquisitionDate,
    v.Status AS VehicleStatus,
    b.BaseID,
    b.BaseName,
    b.Location_ AS BaseLocation,
    a.AccidentID,
    a.DateOfAccident,
    a.DamageLevel,
    mr.RecordID,
    mr.DateOfMaintenance,
    mr.Description_ AS MaintenanceDescription
FROM
    Trip t
LEFT JOIN
    HadTrip ht ON t.TripID = ht.TripID
LEFT JOIN
    Vehicle v ON ht.VehicleID = v.VehicleID
LEFT JOIN
    Base b ON v.BaseID = b.BaseID
LEFT JOIN
    HadAccident ha ON v.VehicleID = ha.VehicleID
LEFT JOIN
    Accident a ON ha.AccidentID = a.AccidentID
LEFT JOIN
    MaintenanceRecord mr ON v.VehicleID = mr.VehicleID;

select * from IntegratedDetailsView
```

	TRIPID	TRIPDATE	DESTINATION	VEHICLEID	VEHICLEMODEL	AQUISITIO
1	113196	03/06/2004	Carmiel			
2	115294	22/08/1991	Tel Aviv	449621	Mahatz	25/01/202
3	115294	22/08/1991	Tel Aviv	799136	LYNX	13/03/202
4	118619	24/11/2018	Rishon LeZion	849224	RAMTA RBY	04/03/202
5	118619	24/11/2018	Rishon LeZion	849224	RAMTA RBY	04/03/202
6	118619	24/11/2018	Rishon LeZion	879477	Tiran 6	16/02/201
7	118619	24/11/2018	Rishon LeZion	548157	Sufa (Storm) Jeep	04/06/202
8	118619	24/11/2018	Rishon LeZion	594746	Wolf Armored Vehicle	23/06/197
9	121324	22/04/1996	Rosh HaAyin			
10	124673	11/04/2000	Rehovot	418488	RBY Mk 1	05/10/201

שאילתות

פרטי תחזוקה אחרונים: שאילתה זו מאחזרת פרטים של רישומי תחזוקה עבור כלי רכב, כולל מזהה הרכב, הדגם, שם הבסיס, מיקום הבסיס, מזהה רשומת התחזוקה, תאריך התחזוקה והתיאור. היא מסננת את התוצאות כדי להציג רק רשומות תחזוקה עם תאריכים לאחר תאריך מוגדר ומוכנס כפרמטר, ומסדרת את התוצאות לפי תאריך התחזוקה בסדר יורד.

```
SELECT
    VehicleID,
    VehicleModel,
    BaseName,
    BaseLocation,
    RecordID,
    DateOfMaintenance,
    MaintenanceDescription
FROM
    IntegratedDetailsView
WHERE
    DateOfMaintenance > TO_DATE(&<name="Maintenance Date" hint="Enter the maintenance date in format DD/MM/YYYY" t
ORDER BY
    DateOfMaintenance DESC;
```

Variables

Name	Value
Maintenance Date	01/01/2000

OK Cancel Clear

Enter the maintenance date in format DD/MM/YYYY

	VEHICLEID	VEHICLEMODEL	BASENAME	BASELOCATION	RECORDID	DATEOFMAINTENANCE	MAINTENANCEDESCRIPTION
1	468277	Machbet	Thunderbolt Barracks	Dimona	643244	20/11/2023	Fjhwtvptwhqokohwhkohflmly
2	468277	Machbet	Thunderbolt Barracks	Dimona	643244	20/11/2023	Fjhwtvptwhqokohwhkohflmly
3	594746	Wolf Armored Vehicle	Spartan Citadel	Qiryat Ata	245625	06/10/2023	Msrskpnnvsnstbqmtkcszfujhm
4	594746	Wolf Armored Vehicle	Spartan Citadel	Qiryat Ata	245625	06/10/2023	Msrskpnnvsnstbqmtkcszfujhm
5	941218	Mahatz	Eagle's Nest	Petah Tikva	925215	27/09/2023	Kqteosghzzbomeuylesifhrfhkt
6	935332	M270 MLRS	Viper Valley Base	Ashdod	335152	08/09/2023	Kqeomfabhimoatlskscccgehk

סך הנסיעות לפי רכב: שאילתה זו מאחזרת את מספר הנסיעות הכולל עבור כל רכב על ידי ספירת הנסיעות המשויכות לכל רכב. התוצאות מקובצות לפי מזהה רכב ודגם, ומסודרות לפי סך הנסיעות בסדר עולה.

```
SELECT
    VehicleID,
    VehicleModel,
    COUNT(TripID) AS TotalTrips
FROM
    IntegratedDetailsView
GROUP BY
    VehicleID, VehicleModel
ORDER BY
    TotalTrips;
```

	VEHICLEID	VEHICLEMODEL	TOTALTRIPS
1	913868	MANTAK	1
2	928564	Sho't Kal	1
3	935332	M270 MLRS	1
4	953765	Merkava Mk.1	1
5	958923	M109 Doher	1
6	987167	Nakpadon	1
7	144226	Panther Command Vehicle	1
8	148158	Nimda Shoet	1
9	163671	PULS	1
10	171715	RBY Mk 1	1