

מיני פרויקט בבסיסי נתונים אגף חיל השריון של צה"ל



מגישים: מאיר שאנני - 216241596

אריאל סיון - 329524839

מרצה: מר יעקב ברזילי

קישור לגיטהב:

https://github.com/trhtk242/DBProject_216241596_329524839

תוכן

3	שלב 1 - עיצוב ויצירת DB, שימוש ב-ERDPLUS והכנסת נתונים
3	תיאור המערכת
4	תרשים ERD של הישויות והקשרים
5	תרשים DSD
5	תיאור הישויות והתכונות שלהן
6	תיאור הקשרים
6	רשימת הסכמות של בסיס הנתונים
6	הסכמות של הישויות
6	הסכמות של הקשרים
7	הוכחה שהסכמות מנורמלות ב-3NF
7	קוד SQL של יצירת הטבלאות (CREATE)
7	תיאור הטבלאות (DESC)
8	קוד SQL של זריקת הטבלאות (DROP)
9	הכנסת מידע לטבלאות
9	הכנסת מידע ל-SOLIDERS – דרך 1, Data Generator
10	הכנסת מידע ל-COMMANDER – דרך 1, Data Generator
11	הכנסת מידע ל-CREWMATE – דרך 2, סקריפט בפיתוח
13	הכנסת מידע ל-UNIT – דרך 3, קובץ טקסט
14	הכנסת מידע ל-TANK – דרך 2, סקריפט בפיתוח
16	הכנסת מידע ל-MISSION – דרך 4, קובץ Excel
17	הכנסת מידע ל-participate – דרך 1, Data Generator
18	גיבוי הטבלאות
19	שחזור הטבלאות
21	שלב 2 - שאילתות (וכן השאילתות עם פרמטרים)
21	שאילתות
21	4 שאילתות select:
23	2 שאילתות delete:
25	2 שאילתות update:
29	4 שאילתות עם פרמטרים:
32	אילוצים
34	שלב 3 – תכנות
34	פונקציות
37	פרוצדורות
39	תוכניות ראשיות

שלב 1 - עיצוב ויצירת DB, שימוש ב-ERDPLUS והכנסת נתונים

תיאור המערכת

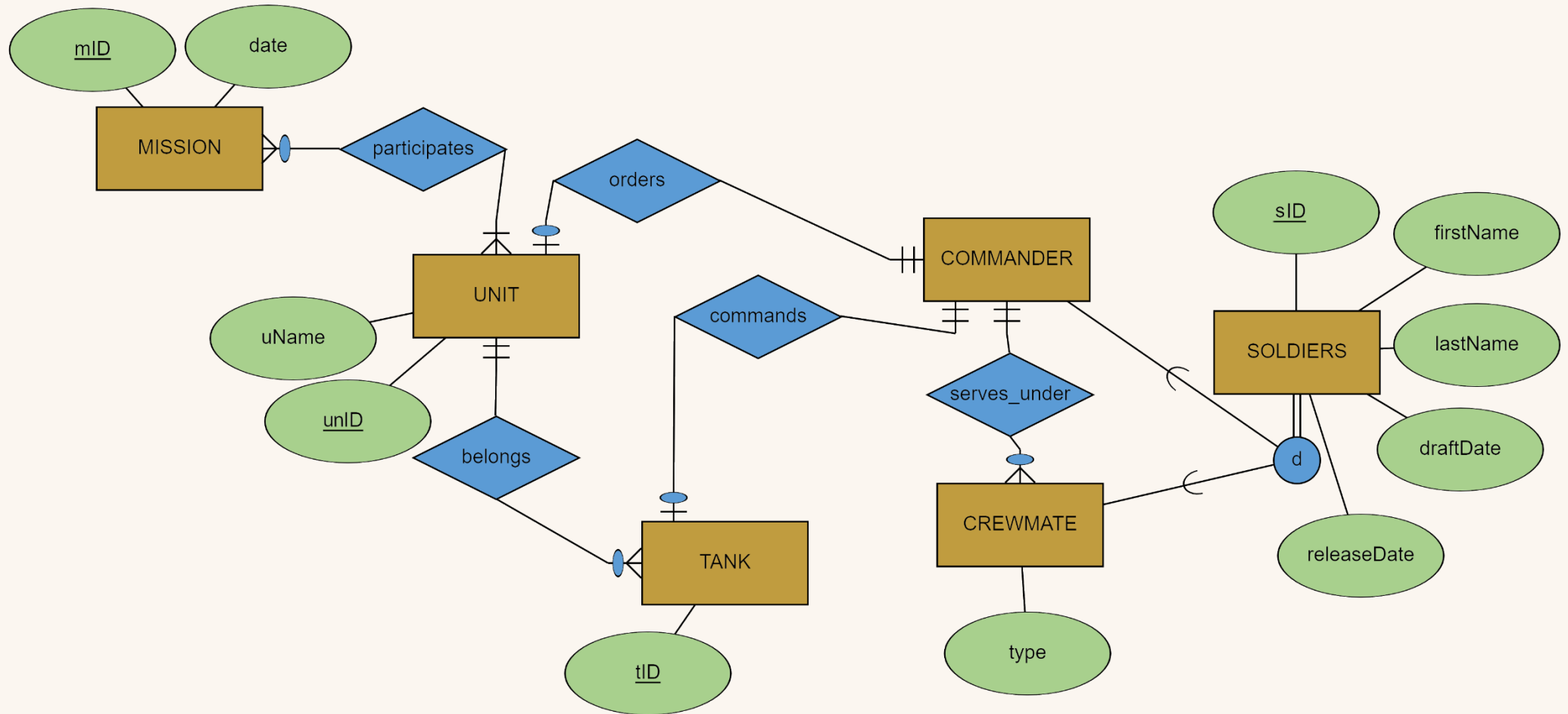
זרוע השריון הישראלית, המוכרת גם כחיל השריון (חש"ן), עומדת כחוד החנית של צבא הגנה לישראל (צה"ל). זרוע זו, מהווה כוח התקפי עיקרי בשדה הקרב, ומשלבת טכנולוגיה מתקדמת, אומץ לב ורוח לחימה עזה.

זרוע השריון היא זרוע עוצמתית וחיונית בצה"ל, הממלאת תפקיד מרכזי בהגנת המדינה. כוח השריון מאפשר לצה"ל לבצע מגוון רחב של משימות בשדה הקרב, ולהוות גורם הרתעה משמעותי מול אויביה.

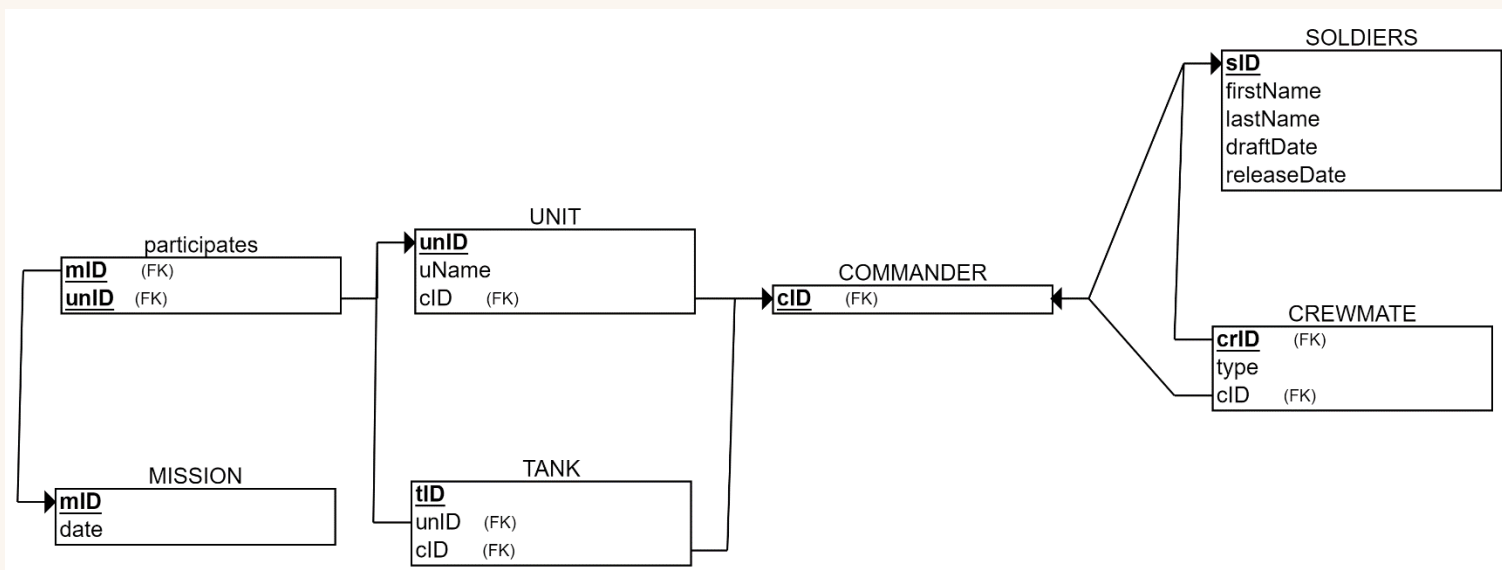
המערכת שנבנה תספק בסיס נתונים לחיל השריון. היא תתחלק לשלושה חלקים:

1. ניהול מידע על החיילים – המערכת תשמור עבור כל חייל את פרטיו האישיים ואת המפקד שלו. עבור מפקד המערכת תשמור גם את תחום פיקודו.
2. ניהול יחידות – המערכת תספק מידע לכל יחידה. שם היחידה, המפקד שלה והטנקים העומדים לרשותה.
3. ניהול משימות – עבור כל משימה המערכת תשמור את זמני המשימה והיחידות המשתתפות

תרשים ERD של הישיות והקשרים



תרשים DSD



תיאור הישויות והתכונות שלהן

קבוצה של כל החיילים	SOLDIERS (חיילים)
תעודת זהות	<u>sID</u>
שם פרטי	firstName
שם משפחה	lastName
תאריך גיוס	draftName
תאריך שחרור	releaseDate

מפקד על טנק או על יחידה. יורש מחיילים	COMMANDER (מפקד)
מפתח, תעודת זהות	<u>cID</u>

איש צוות בתוך טנק. יורש מחיילים	CREWMATE (איש צוות)
מפתח, תעודת זהות	<u>crID</u>
מפתח זר, תעודת הזהות של מפקד הצוות	crID
סוג החייל (נהג, טען או תותחן)	type

יחידה בחיל השריון	UNIT (יחידה)
מפתח, מספר יחידה מזהה	<u>unID</u>
מפתח זר, תעודת הזהות של מפקד היחידה	cID
שם היחידה	uName

TANK (טנק)	טנק בחיל השריון
<u>tID</u>	מפתח, מספר טנק מזהה
cID	מפתח זר, תעודת הזהות של מפקד הטנק
unID	מפתח זר, מספר היחידה המזהה של הטנק

MISSION (משימה)	משימה בחיל השריון
<u>mID</u>	מפתח, מספר משימה מזהה
date	התאריך בו יוצאת המשימה לפועל

תיאור הקשרים

participates (משתתף)	רבים לרבים - יחידה <u>שמשתתפת</u> במשימה
<u>mID</u>	מפתח, מספר המשימה מזהה
<u>unID</u>	מפתח, מספר היחידה מזהה

1. serves_under (משרת) – יחיד לרבים, איש צוות שמשרת תחת מפקד.
2. commands (מפקד) – יחיד לרבים, מפקד שמפקד על טנק.
3. belongs (משתייך) – יחיד לרבים, טנק שמשייך ליחידה.
4. orders (נותן הוראות) – יחיד לרבים, מפקד שנותן הוראות ליחידה.

רשימת הסכמות של בסיס הנתונים

הסכמות של הישויות

SOLIDERS(sID, firstName, lastName, draftName, releaseDate)
 COMMANDER(cID)
 CREWMATE(crID, cID, type)
 TANK(tID, unID, cID)
 UNIT(unID, cID, uName)
 MISSION(mID, mdate)

הסכמות של הקשרים

participates(mID, unID)

הוכחה שהסכמות מנורמלות ב3NF

SOLDIERS (חיילים) – המפתח הראשי הוא sID. וכל שאר התכונות תלויות ישירות במפתח הראשי sID. ולכן הטבלה ב3NF.

COMMANDER (מפקד) – cID הוא מפתח ראשי וגם מפתח זר המפנה ל-sID בטבלה SOLDIERS. אין מאפיינים נוספים ולכן הטבלה ב3NF.

CREWMATE (איש צוות) – type תלוי ב-cID וגם crID גם קשור ישירות ל-crID כמפתח זר. הטבלה ב3NF.

UNIT (יחידה) – uName ו-cID תלויים במפתח הראשי unID. הטבלה נמצאת ב3NF.

TANK (טנק) – unID ו-cID הם תכונות עם יחסים של מפתחות זרים. אבל, אין תלות טרנזיטיבית כיוון ש-tID הוא המפתח הראשי. ולכן הטבלה ב3NF.

MISSION (משימה) – מכיוון ש-mdate תלוי ישירות במפתח הראשי mID, הטבלה ב3NF.

participates (משתתף) – גם mID וגם unID הם מפתחות זרים, והם יוצרים יחד מפתח מורכב. כל מאפיין במפתח המורכב קובע לחלוטין את המאפיינים האחרים.

למסקנה כל הטבלאות עונות על התנאים ל-3NF.

קוד SQL של יצירת הטבלאות (CREATE)

h

תיאור הטבלאות (DESC)

```
SQL> DESC SOLDIERS;
```

Name	Null?	Type
SID	NOT NULL	NUMBER(9)
FIRSTNAME	NOT NULL	VARCHAR2(20)
LASTNAME	NOT NULL	VARCHAR2(20)
DRAFTDATE	NOT NULL	DATE
RELEASEDATE	NOT NULL	DATE

```
SQL> DESC MISSION;
```

Name	Null?	Type
MDATE	NOT NULL	DATE
MID	NOT NULL	NUMBER(9)

```
SQL> DESC COMMANDER;
Name          Null?   Type
-----
CID           NOT NULL NUMBER(9)
```

```
SQL> DESC CREWMATE;
Name          Null?   Type
-----
TYPE          NOT NULL VARCHAR2(20)
CRID          NOT NULL NUMBER(9)
CID           NOT NULL NUMBER(9)
```

```
SQL> DESC UNIT;
Name          Null?   Type
-----
UNID          NOT NULL NUMBER(9)
UNAME         NOT NULL VARCHAR2(20)
CID           NOT NULL NUMBER(9)
```

```
SQL> DESC TANK;
Name          Null?   Type
-----
TID           NOT NULL NUMBER(9)
UNID          NOT NULL NUMBER(9)
CID           NOT NULL NUMBER(9)
```

```
SQL> DESC participates;
Name          Null?   Type
-----
MID           NOT NULL NUMBER(9)
UNID          NOT NULL NUMBER(9)
```

קוד SQL של זריקת הטבלאות (DROP)

```
DROP TABLE CREWMATE;
DROP TABLE TANK;
DROP TABLE participates;
DROP TABLE MISSION;
DROP TABLE UNIT;
```



```
DROP TABLE COMMANDER;
DROP TABLE SOLDIERS;
```

הכנסת מידע לטבלאות

נכניס את כל הנתונים שלנו לתוך מסד הנתונים. לשם כך נוכל להשתמש בשיטות שונות.

הכנסת מידע ל-SOLIDERS – דרך 1, Data Generator

ניצור ונכניס את ה-Datan בעזרת מחולל המידע של Oracle (Data Generator)

ככה נראה המידע המפוברק שקיבלנו ממחולל הנתונים.

SOLDIERS					
	SID	FIRSTNAME	LASTNAME	DRAFTDATE	RELEASEDATE
	1	Demi	Callow	24-08-2006	13-05-2009
	2	Lauren	Laurie	04-10-2006	20-03-2009
	3	Rip	Gershon	30-09-2006	11-01-2009
	4	Elvis	Durning	21-06-2006	25-03-2009
	5	Meryl	Beckham	04-09-2006	05-09-2009
	6	Kirsten	Bush	28-11-2006	19-05-2009
	7	Wally	Ramirez	10-03-2006	21-12-2009
	8	Sydney	Peebles	09-06-2006	25-03-2009
	9	Joaquin	Benoit	29-01-2006	18-01-2009
	10	Walter	Hatchet	18-03-2006	01-09-2009
	11	Vivica	Lever	14-04-2006	08-02-2009
	12	Debra	Rawls	21-03-2006	09-01-2009
	13	Casey	Broadbent	01-05-2006	18-07-2009
	14	Miguel	Tripplehorn	08-08-2006	22-09-2009
	15	Meredith	Baldwin	20-01-2006	03-09-2009
	16	Grace	Davidson	25-04-2006	20-07-2009

כאשר נוכל להמיר את ה-Datan לפקודות INSERT בשפת SQL

Name	Type	Size	Data
SID	NUMBER	9	Sequence(1)
FIRSTNAME	VARCHAR2	20	FirstName
LASTNAME	VARCHAR2	20	LastName
DRAFTDATE	DATE		random(01/01/2006,31/12/2006)
RELEASEDATE	DATE		random(01/01/2009,31/12/2009)

```

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (1, 'Marc', 'Dunaway', to_date('27-10-2006', 'dd-mm-yyyy'), to_date('06-06-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (2, 'Rade', 'Wheel', to_date('14-07-2006', 'dd-mm-yyyy'), to_date('29-06-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (3, 'Jackson', 'Place', to_date('06-12-2006', 'dd-mm-yyyy'), to_date('30-12-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (4, 'Sarah', 'Crouch', to_date('28-07-2006', 'dd-mm-yyyy'), to_date('07-06-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (5, 'Sheryl', 'Stowe', to_date('22-10-2006', 'dd-mm-yyyy'), to_date('29-03-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (6, 'Rosco', 'Wells', to_date('03-08-2006', 'dd-mm-yyyy'), to_date('30-12-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (7, 'Jeff', 'Beckinsale', to_date('06-10-2006', 'dd-mm-yyyy'), to_date('03-06-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
values (8, 'Linda', 'Lloyd', to_date('05-06-2006', 'dd-mm-yyyy'), to_date('15-04-2009', 'dd-mm-yyyy'))

```

בדוק שהמידע אכן נקלט במערכת בעזרת הפקודה `Select * from SOLDIERS`

	SID	FIRSTNAME	LASTNAME	DRAFTDATE	RELEASEDATE
1	1	Jet	Gore	8/24/2006	4/9/2009
2	2	Latin	Garber	9/25/2006	8/15/2009
3	3	Sona	Valentin	11/14/2006	5/23/2009
4	4	Nicolas	Alston	12/9/2006	7/4/2009
5	5	Candice	Sheen	11/22/2006	4/2/2009
6	6	Naomi	Webb	1/7/2006	9/1/2009
7	7	Linda	Lloyd	6/5/2006	4/15/2009
8	8	Xander	Flemyng	11/11/2006	5/4/2009
9	9	Edwin	Kimball	2/14/2006	12/13/2009
10	10	Rawlins	Dolenz	7/31/2006	2/10/2009
11	11	Lupe	Pitney	9/11/2006	4/16/2009
12	12	Nelly	Harris	6/30/2006	1/19/2009
13	13	Ryan	MacDowell	5/4/2006	1/27/2009
14	14	Mia	De Almeida	7/6/2006	5/7/2009
15	15	Kim	Greenwood	6/20/2006	7/5/2009

הכנסת מידע לCOMMANDER – דרך 1, Data Generator

גם כאן ניצור ונכניס את המידע בעזרת מחולל המידע של Oracle (Data Generator)

נבדוק האם המידע נקלט בעזרת הפקודה `Select * from COMMANDER`

Name	Type	Size	Data
CID	NUMBER	9	Sequence(1)

הכנסת מידע לCREWMATE – דרך 2, סקריפט בפייתון

הפעם ניקח דרך שונה. ניצור סקריפט בפייתון שיצור רשימה של פקודות INSERT בהן נשתמש בשביל להכניס מידע לאנשי הצוות.

	CID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14

```
types =
['Loader','Gunner','Driver']
```

```
def
```

```
insert into SYS.COMMANDER (CID)
values (1);
```

```
insert into SYS.COMMANDER (CID)
values (2);
```

```
insert into SYS.COMMANDER (CID)
values (3);
```

```
insert into SYS.COMMANDER (CID)
values (4);
```

```
insert into SYS.COMMANDER (CID)
values (5);
```

```
insert into SYS.COMMANDER (CID)
values (6);
```

```
insert into SYS.COMMANDER (CID)
values (7);
```

```
insert into SYS.COMMANDER (CID)
```

```
generate_crewmate_data(num_records):
```

```
data = []
```

```

for x in range(num_records):
    id = x + 601
    cid = int((id - 601) / 3) + 1; # getting the id of the commander
    type = types[x % 3]
    data.append((id,cid, type))
return data

def generate_sql_insert_statements(table_name, data):
    sql_statements = []
    for record in data:
        sql = f"INSERT INTO {table_name} (CRID, CID, TYPE) VALUES ({record[0]},{record[1]},{record[2]});"
        sql_statements.append(sql)
    return sql_statements

# Generate data
num_records = 1400 # Number of records we want to generate
crewmate_data = generate_crewmate_data(num_records)

# Generate SQL insert statements
sql_statements = generate_sql_insert_statements('CREWMATE',
crewmate_data)

```

נכניס את שורות הInsert אל לתוך מסך הנתונים

SQL	Output	Statistics
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1642,348,Loader);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1643,348,Gunner);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1644,348,Driver);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1645,349,Loader);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1646,349,Gunner);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1647,349,Driver);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1648,350,Loader);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1649,350,Gunner);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1650,350,Driver);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1651,351,Loader);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1652,351,Gunner);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1653,351,Driver);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1654,352,Loader);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1655,352,Gunner);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1656,352,Driver);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1657,353,Loader);		
INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (1658,353,Gunner);		

ונוודא שמידע נקלט בעזרת פקודת SELECT

	TYPE	CRID	CID
1	Loader	601	1
2	Gunner	602	1
3	Driver	603	1
4	Loader	604	2
5	Gunner	605	2
6	Driver	606	2
7	Loader	607	2
8	Gunner	608	3
9	Driver	609	3

הכנסת מידע לUNIT – דרך 3, קובץ טקסט
 הפעם נכניס למסד הנתונים בטבלה UNIT מידע מקובץ טקסט

```

a - Notepad
File Edit Format View Help
1, 351, Saar me-Golan
2, 352, Barak
3, 353, Iron tails
4, 354, Sons of light
5, 355, Utzvat HaPlada
6, 356, Ikvot HaBarzel
7, 357, Bnei Or
8, 358, Kiryati
9, 359, Machatz
10, 360, Ram
11, 361, Iron Fist
12, 362, Yiftach
13, 363, Sinai
14, 364, Harel

```

הטקסט יעבור לOracle בעזרת Text Importer

Data Generator Data Generator a.txt X

Data from Textfile Data to Oracle

Fieldcount: 3

Quote Character: "

Field2 (+0 .. ",")

Field3 (+0 .. ",")

relative position

Absolute position

Character

Field End

Length

Character

End at line-end

Name in header

Skip empty lines

Comment line

Import lines: 1 ..

Filter

Apply

Result Preview

1	2	3
1	351	Saar me-Golan
2	352	Barak
3	353	Iron tails
4	354	Sons of light
5	355	Utzvat HaPlada

Import Import to Script Close asivan AS SYSDBA a.txt loaded, 5 KB Help

Data from Textfile Data to Oracle

General

Owner: Table: UNIT

Commit every...: 100

Overwrite duplicates

Ignore duplicates

Delete records

Truncate table

Initializing Script

Finalizing Script

Fields

Field1 -> UNID (NUMBER)

Field2 -> CID (NUMBER)

Field3 -> UNAME (VARCHAR2)

Field: UNAME (VARCHAR2)

Fieldtype: String

Create SQL

SQL function

additional Oracle processing, for example: substr(#, 1, 20)

נוודא שאכן המידע נשמר במערכת

	UNID	UNAME	CID
1	1	Saar me-Golan	351
2	2	Barak	352
3	3	Iron tails	353
4	4	Sons of light	354
5	5	Utzvat HaPlada	355
6	6	Ikvtot HaBarzel	356
7	7	Bnei Or	357
8	8	Kiryati	358
9	9	Machatz	359
10	10	Ram	360
11	11	Iron Fist	361

הכנסת מידע לTANK – דרך 2, סקריפט בפיתוח

```
import random
```

```
def generate_tank_data(num_records):
    data = []
    for x in range(num_records):
        id = x
        cid = x
        unid = random.randrange(1,250) #
    getting the id of the unit
        data.append((id,cid, unid))
    return data

def
generate_sql_insert_statements(table_name,
data):
    sql_statements = []
    for record in data:
        sql = f"INSERT INTO {table_name} (TID, CID, UNID) VALUES
({record[0]},{record[1]},{record[2]});"
        sql_statements.append(sql)
    return sql_statements

# Generate data
num_records = 350 # Number of records we want to generate
tank_data = generate_tank_data(num_records)

# Generate SQL insert statements
sql_statements = generate_sql_insert_statements('TANK', tank_data)
```

	TID	UNID	CID
1	1	218	1
2	2	232	2
3	3	211	3
4	4	195	4
5	5	62	5
6	6	185	6
7	7	38	7
8	8	68	8
9	9	234	9
10	10	92	10
11	11	130	11
12	12	221	12

הסקריפט יצור לנו 350 פקודות INSERT בSQL

```
INSERT INTO TANK (TID, CID, UNID) VALUES (0,0,156);
INSERT INTO TANK (TID, CID, UNID) VALUES (1,1,218);
INSERT INTO TANK (TID, CID, UNID) VALUES (2,2,232);
INSERT INTO TANK (TID, CID, UNID) VALUES (3,3,211);
INSERT INTO TANK (TID, CID, UNID) VALUES (4,4,195);
INSERT INTO TANK (TID, CID, UNID) VALUES (5,5,62);
INSERT INTO TANK (TID, CID, UNID) VALUES (6,6,185);
INSERT INTO TANK (TID, CID, UNID) VALUES (7,7,38);
INSERT INTO TANK (TID, CID, UNID) VALUES (8,8,68);
```

נוודא שהמידע נקלט במערכת בעזרת שאילתת: `select * from TANK;`

הכנסת מידע לMISSION - דרך 4, קובץ Excel

נשתמש במידע מתוך קובץ Excel

MID	MDATE
1	15-01-1990
2	7/2/1990
3	12/3/1990
4	19-04-1990
5	3/5/1990
6	27-06-1990
7	14-07-1990
8	21-08-1990
9	5/9/1990
10	18-10-1990

נפתח את קובץ Excel בעזרת ODBC Importer

The screenshot shows the ODBC Importer window with the following details:

- Toolbar:** Includes icons for file operations and a query icon.
- Query:** The text "select * from tank;" is entered in the top bar.
- Connection:**
 - User / System DSN:** Excel Files
 - User Name:** asivan
 - Password:** Masked with dots
 - Buttons:** Connect, Disconnect
- Table / Query:**
 - Import Table:** Selected option.
 - view Data:** Button to view the data.
- Result Preview:**

MDATE	MID
15-01-1990	1.0
07-02-1990	2.0
12-03-1990	3.0
19-04-1990	4.0

General

Owner: Table: MISSION

Commit every...: 100

☒ Overwrite duplicates ☐ Delete records

☐ Ignore duplicates ☐ Truncate table

Initializing Script:

Finalizing Script:

Fields

MDATE -> MDATE
MID -> MID

Field: MID

Fieldtype: (Float)

Result Preview

MDATE	MID
15-01-1990	1.0
07-02-1990	2.0
12-03-1990	3.0
19-04-1990	4.0

	MDATE	MID
1	1/15/1990	1
2	2/7/1990	2
3	3/12/1990	3
4	4/19/1990	4
5	5/3/1990	5
6	6/27/1990	6
7	7/14/1990	7
8	8/21/1990	8
9	9/5/1990	9
10	10/18/1990	10

נבדוק שהData אכן עבר מהExcel אל מסד הנתונים

הכנסת מידע לparticipate – דרך 1, Data Generator

נשתמש בData Generator בשביל ליצור משימות ולהכניס אותן אל בסיס הנתונים

PARTICIPATES

< Owner: SYS Table: PARTICIPATES Number of records: 200

>

Name	Type	Size	Data	Master
MID	NUMBER	9	Random(1, 300)	...
UNID	NUMBER	9	Random(1, 250)	...
				...

מחולל המידע הופך את
הנתונים שלנו לפקודות
Insert בשפת SQL

```
insert into SYS.PARTICIPATES (MID, UNID)
values (122, 41);

insert into SYS.PARTICIPATES (MID, UNID)
values (168, 63);

insert into SYS.PARTICIPATES (MID, UNID)
values (212, 165);

insert into SYS.PARTICIPATES (MID, UNID)
values (23, 116);

insert into SYS.PARTICIPATES (MID, UNID)
values (9, 46);

insert into SYS.PARTICIPATES (MID, UNID)
values (102, 106);

insert into SYS.PARTICIPATES (MID, UNID)
values (172, 55);
```

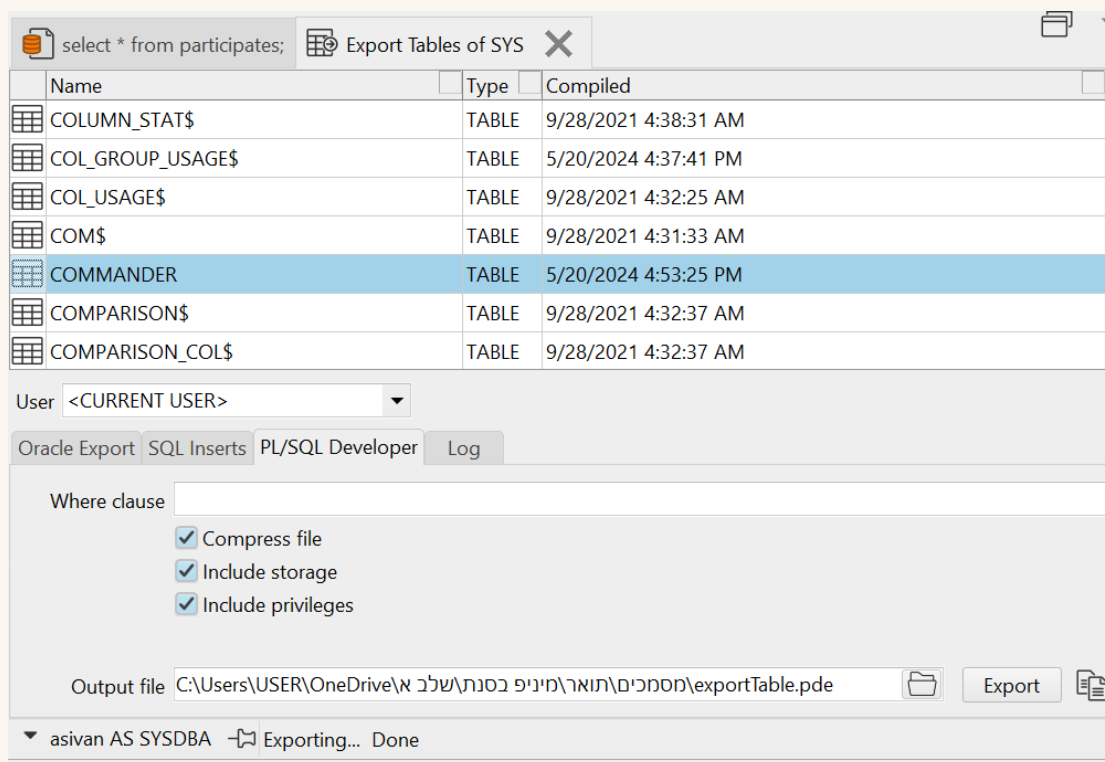
אחרי שהרצנו את הפקודות בתמונה הקודמת,
נבדוק בעזרת פקודת Select האם המידע אכן
נקלט

	MID	UNID
1	11	70
2	14	54
3	14	228
4	16	167
5	18	136
6	21	46
7	22	72
8	22	176
9	25	85
10	27	140
11	28	145
12	29	149

גיבוי הטבלאות

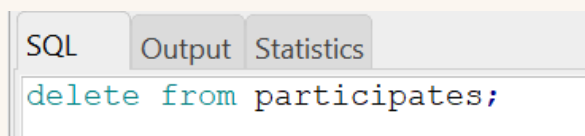
בשביל להבטיח שהמידע של בסיס הנתונים שלנו תמיד נשמר נצטרך לגבות את כל המידע שבתוך הטבלאות למקור שמירה חיצוני.

אחרי שסימנו את כל הטבלאות שאנחנו רוצים לשמור, נייצא את המידע שבהן לתוך קובץ מסוג יצוא וככה נוכל גם להעביר את הטבלאות למחשבים אחרים וגם לשמור עליהן

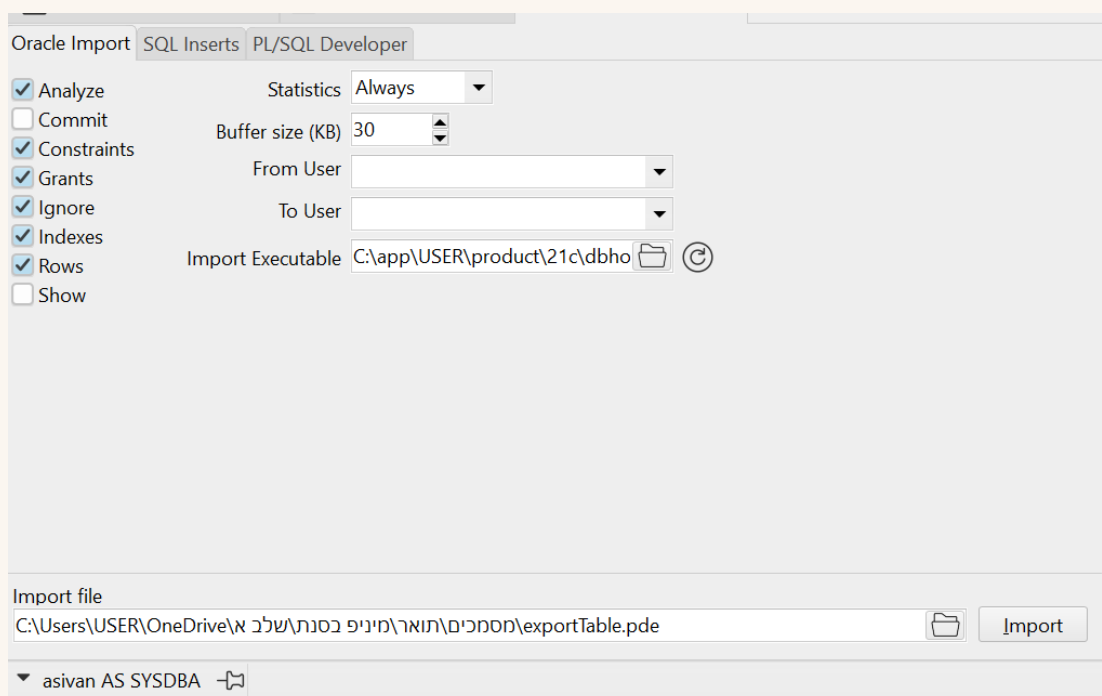


שחזור הטבלאות

אחרי שגיבנו את כל הטבלאות, נרצה לראות שזה אכן עבד. החלטנו למחוק את המידע שבטבלה participates בעזרת הפקודה Delete.



נטען עכשיו את המידע מקובץ היצוא החיצוני שלנו בשביל לבדוק האם המידע באמת נשמר



המידע נשמר 🤔

אנחנו יכולים לגשת אליו גם אחרי שמחקנו אותו

select * from participates;

SQL Output Statistics

select * from participates;

🖨️ 🔒 + × ✓ ⬇️

	MID	UNID
▶ 1	11	70
2	14	54
3	14	228
4	16	167
5	18	136
6	21	46
7	22	72
8	22	176
9	25	85
10	27	140
11	28	145
12	29	149
13	32	15
14	32	159

שלב 2 - שאילות (וכן השאילות עם פרמטרים)

שאילות

4 שאילות select:

שאילתה 1 : מציאת זמן השירות הממוצע של חיילים ששירתו בתור מפקדים, והשוואה עם זמן השירות הממוצע של אילו שאינם מפקדים

```
-- Subquery for Commanders
SELECT 'Commander' AS role,
      AVG(s.releaseDate - s.draftDate) AS avgServiceDuration
FROM SOLDIERS s
JOIN COMMANDER c ON s.sID = c.cID

UNION ALL

-- Subquery for Non-Commanders
SELECT 'Non-Commander' AS role,
      AVG(s.releaseDate - s.draftDate) AS avgServiceDuration
FROM SOLDIERS s
LEFT JOIN COMMANDER c ON s.sID = c.cID
WHERE c.cID IS NULL;
```

התוצאה שתתקבל מהרצת השאילתה היא

	ROLE	AVGSERVICEDURATION
1	Commander	1097.90166666667
2	Non-Commander	1090.57428571428

asivan AS SYSDBA [4:46:03 PM] 2 rows selected in 0.059 seconds

שאילתה 2 : רשימה של כל היחידות שהשתתפו ביותר משימות ממספר המשימות ליחידה הממוצע

```
SELECT u.unID, u.uName, COUNT(p.mID) AS missionCount
FROM UNIT u
JOIN participates p ON u.unID = p.unID
GROUP BY u.unID, u.uName
HAVING COUNT(p.mID) > (
  SELECT AVG(missionCount)
  FROM (
```

```

SELECT COUNT(p.mID) AS missionCount
FROM participates p
GROUP BY p.unID

) subquery
)
ORDER BY missionCount DESC;

```

התוצאה שתתקבל מהרצת השאילתה היא

	UNID	UNAME	MISSIONCOUNT
1	168	Gaash 4	10
2	162	Bazka 4	10
3	129	Iron tails 4	9
4	218	Kiryati 6	9
5	11	Iron Fist	9

5:39 asivan AS SYSDBA [5:06:18 PM] 5 rows selected in 0.052 seconds (more...)

שאילתה 3 : כל היחידות ממוימות לפי מספר הטנקים בכל יחידה

```

SELECT u.uName AS unitName, COUNT(t.tID) AS numberOfTanks
FROM UNIT u
LEFT JOIN TANK t ON u.unID = t.unID
GROUP BY u.uName
ORDER BY numberOfTanks DESC;

```

התוצאה שתתקבל מהרצת השאילתה היא

	UNITNAME	NUMBEROFTANKS
1	Yerushalaim 5	6
2	Etzion Gever 5	6
3	Mapatz 3	5
4	Sons of light 2th	5
5	Kela 5	5
6	Iron Fist 2th	5
7	Ikvot HaBarzel 3	4
8	Yishai 4	4

6:33 asivan AS SYSDBA [5:19:58 PM] 8 rows selected in 0.028 seconds (more...)

שאילתה 4 : אחרי מבצע מוצלח רצו בפיקוד להביא לחיילים צ'ופר, לכל חייל תיק עם הדפס של היחידה. בשביל כל החיילים שהתגייסו במרץ, הודפס התיק "גיוס מרץ שריון".

יחידת האפסנאות צריכה רשימה של כל החיילים שהתגייסו בסביבות מרץ אפריל.

```

SELECT * FROM CREWMATE

```

```
WHERE crID IN (
  SELECT sID
  FROM SOLDIERS
  WHERE EXTRACT(MONTH FROM draftDate) BETWEEN 3 AND 4
);
```

התוצאה שתתקבל מהרצת השאילתה היא

		TYPE		CRID		CID	
▶	1	Loader	***	1708		370	
	2	Gunner	***	1727		376	
	3	Gunner	***	1730		377	
	4	Loader	***	1732		378	
	5	Driver	***	1740		380	
	6	Loader	***	1741		381	
	7	Gunner	***	1742		381	
	8	Loader	***	1744		382	

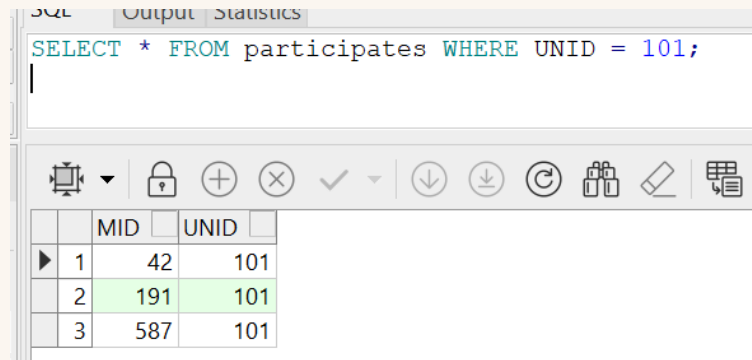
5:54 asivan AS SYSDBA [5:23:55 PM] 8 rows selected in 0.034 seconds (more...)

2 שאילות delete:

שאילתה 1 : בעקבות תפקודה המופתי של יחידה 101 בעזה, הוחלט בדרג הפיקודי העליון על הפיכת היחידה ליחידה מיוחדת ומסווגת ועל כן, יש צורך במחיקת כל הרישומים המעידים על קיום היחידה

```
DELETE FROM participates AS p
WHERE p.UNID = 101;
```

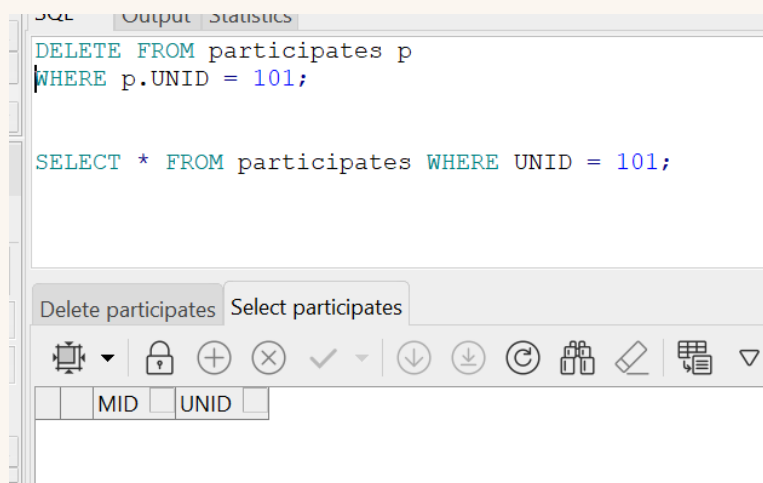
לפני הרצת פקודת המחיקה :



The screenshot shows a SQL IDE with a query editor and a results pane. The query is: `SELECT * FROM participates WHERE UNID = 101;`. The results pane shows a table with 3 rows and 2 columns: MID and UNID.

	MID	UNID
1	42	101
2	191	101
3	587	101

לאחר הרצת פקודת המחיקה :



The screenshot shows a SQL IDE with two queries. The first query is: `DELETE FROM participates p WHERE p.UNID = 101;`. The second query is: `SELECT * FROM participates WHERE UNID = 101;`. The results pane shows a table with 2 columns: MID and UNID.

	MID	UNID
--	-----	------

לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה

שאלתה 2 : אחרי מקרה מצער בו כל אנשי הצוות של טנק מספר 186 נפלו בקרב, נרצה להסיר אותם מבסיס הנתונים.

```
DELETE FROM CREWMATE cr
WHERE cr.crID IN (
  SELECT cr.crID
  FROM CREWMATE cr
  JOIN TANK t ON t.cID = cr.cID
  JOIN UNIT u ON u.unID = t.unID
  WHERE u.unID = 186
);
```


לפני הרצת פקודת המחיקה :

```

select * FROM CREWMATE cr
WHERE cr.crID IN (
  SELECT cr.crID
  FROM CREWMATE cr
  JOIN TANK t ON t.cID = cr.cID
  JOIN UNIT u ON u.unID = t.unID
  WHERE u.unID = 186
);

```

	TYPE	CRID	CID
1	Loader	1534	312
2	Gunner	1535	312
3	Driver	1536	312
4	Gunner	674	25
5	Driver	675	25
6	Loader	883	95

לאחר הרצת פקודת המחיקה :

SQL Output Statistics

```

WHERE cr.crID IN (
  SELECT cr.crID
  FROM CREWMATE cr
  JOIN TANK t ON t.cID = cr.cID
  JOIN UNIT u ON u.unID = t.unID
  WHERE u.unID = 186
);

select * FROM CREWMATE cr
WHERE cr.crID IN (
  SELECT cr.crID
  FROM CREWMATE cr
  JOIN TANK t ON t.cID = cr.cID

```

Delete crewmate Select crewmate Rollback

	TYPE	CRID	CID
--	------	------	-----

לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה

2 שאלות update:

שאלתה 1 :

```

UPDATE UNIT u
SET u.uName = 'Inactive Unit'

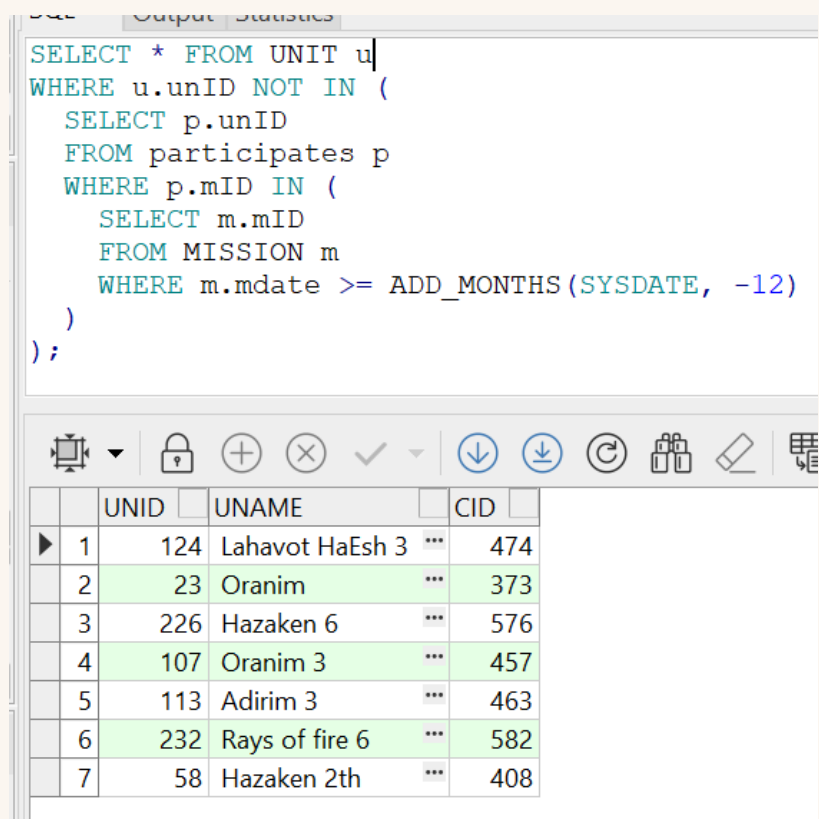
```

```

WHERE u.unID NOT IN (
  SELECT p.unID
  FROM participates p
  WHERE p.mID IN (
    SELECT m.mID
    FROM MISSION m
    WHERE m.mdate >= ADD_MONTHS(SYSDATE, -12)
  )
);

```

לפני הרצת פקודת העדכון :



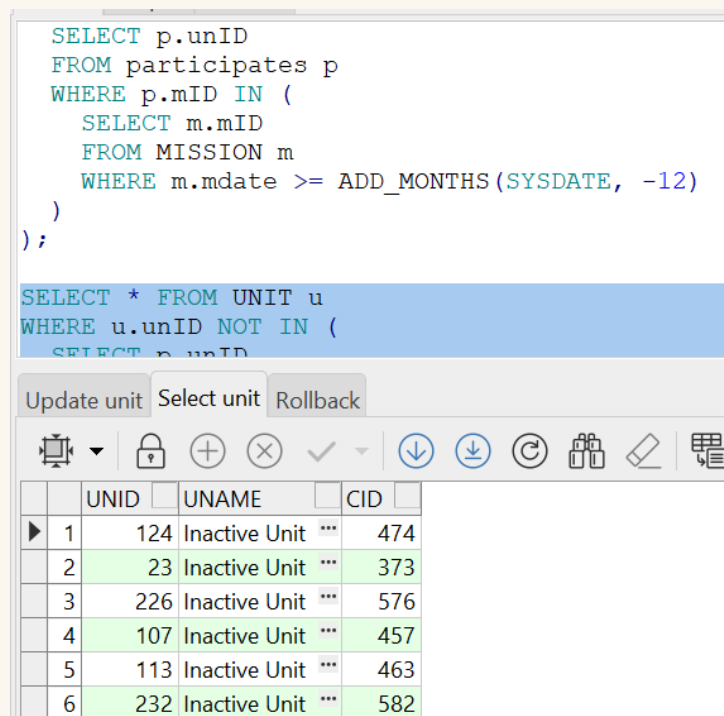
```

SELECT * FROM UNIT u
WHERE u.unID NOT IN (
  SELECT p.unID
  FROM participates p
  WHERE p.mID IN (
    SELECT m.mID
    FROM MISSION m
    WHERE m.mdate >= ADD_MONTHS(SYSDATE, -12)
  )
);

```

	UNID	UNAME	CID
1	124	Lahavot HaEsh 3	474
2	23	Oranim	373
3	226	Hazaken 6	576
4	107	Oranim 3	457
5	113	Adirim 3	463
6	232	Rays of fire 6	582
7	58	Hazaken 2th	408

לאחר הרצת פקודת העדכון :



לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה

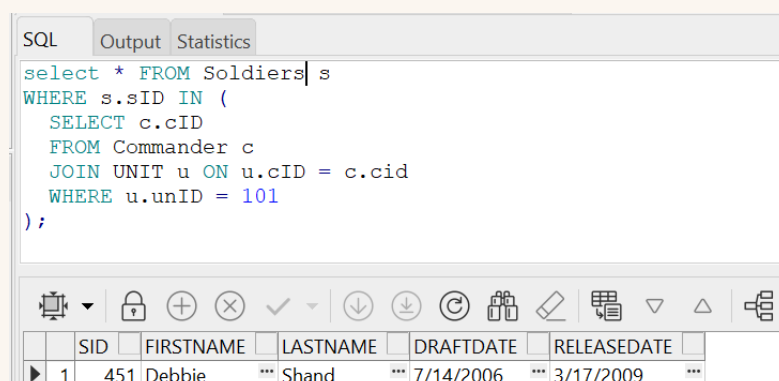
שאלתה 2 : לאחר שיחידה 101 הפכה למסווגת, גם המפקד שלה נדרש לרדת לצללים ועל כן הוא שינה את שמו ל"אנו נימי"

```

UPDATE COMMANDER
SET firstName = 'Ano',
    lastName = 'Nimi'
WHERE cID = (
    SELECT u.cID
    FROM UNIT u
    WHERE u.unID = 101
);

```

לפני הרצת פקודת העדכון :



לאחר הרצת פקודת העדכון :

```
UPDATE Soldiers s
SET s.firstName = 'Ano',
    s.lastName = 'Nimi'
WHERE s.sID IN (
    SELECT c.cID
    FROM Commander c
    JOIN UNIT u ON u.cID = c.cid
    WHERE u.unID = 101
);

select * FROM Soldiers s
WHERE s.sID IN (
    SELECT c.cID
    FROM Commander c
    JOIN UNIT u ON u.cID = c.cid
    WHERE u.unID = 101
);
```

Update soldiers | Select soldiers | Rollback

	SID	FIRSTNAME	LASTNAME	DRAFTDATE	RELEASEDATE
▶ 1	451	Ano	Nimi	7/14/2006	3/17/2009

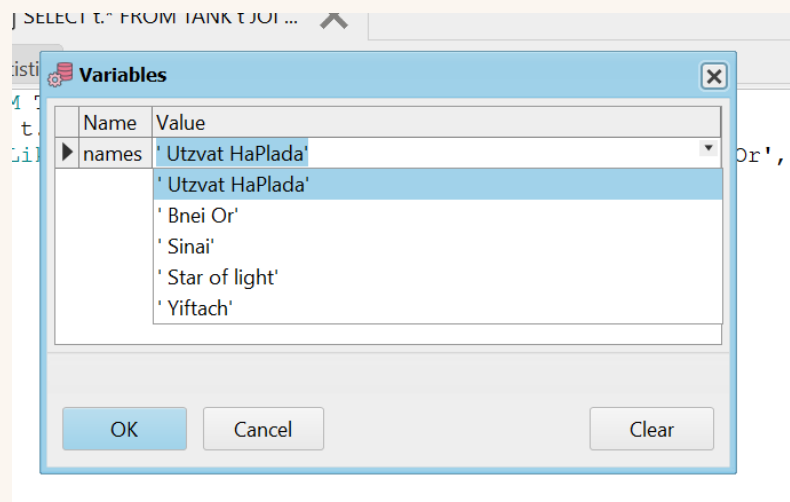
לאחר הרצת פקודת rollback הנתונים יחזרו להיות כמו שהם היו בתמונה הראשונה

4 שאלות עם פרמטרים:

שאלה 1 :

```
SELECT t.* FROM TANK t
JOIN UNIT u ON t.unID = u.unID
WHERE u.uName Like '<name="names" list=" ' Utzvat HaPlada', ' Bnei Or', '
Sinai', ' Star of light', ' Yiftach' " >
```

השימוש בפרמטר



התוצאה שתקבל מהרצת השאלה היא

	TID	UNID	CID
1	94	5	94
2	243	5	243

asivan AS SYSDBA [5:33:51 PM] 2 rows selected in 0.037 seconds

שאלה 2 :

```
SELECT m.mID AS missionID,
SELECT m.mID AS missionID,
      m.mdate AS missionDate,
      COUNT(p.unID) AS numberOfParticipatingUnits
FROM MISSION m
JOIN participates p ON m.mID = p.mID
GROUP BY m.mID, m.mdate
HAVING COUNT(p.unID) > '<name = "number of units" hint="number of
units can be from 1 to 5" type="integer">
ORDER BY numberOfParticipatingUnits DESC;
```

השימוש בפרמטר

Name	Value
✓ number of units	3

number of units can be from 1 to 5

OK Cancel Clear

התוצאה שתתקבל מהרצת השאילתה היא

	MISSIONID	MISSIONDATE	NUMBEROFPARTICIPATINGUNITS
1	62	2/5/2022	5
2	538	4/27/2006	5
3	523	4/6/2016	5
4	116	2/10/2014	4
5	501	12/4/2004	4
6	590	10/27/1990	4

asivan AS SYSDBA [5:51:46 PM] 6 rows selected in 0.070 seconds (more...)

שאילתה 3 :

```
UPDATE SOLDIERS s
SET releaseDate = TO_DATE(&<name="new release date" type="date">,
'YYYY-MM-DD')
WHERE s.firstName = &<name = "first name">;
```

השימוש בפרמטר

Name	Value
▶ new release date	12/2/12
first name	Percy

OK Cancel Clear

התוצאה שתקבל מהרצת השאילתה היא

```

UPDATE SOLDIERS s
SET releaseDate = TO_DATE(&<name="new release date" type="date">, 'YYYY-MM-DD')
WHERE s.firstName = &<name = "first name" type="string">;

select * from SOLDIERS s
WHERE s.firstName = &<name = "first name">;

rollback;

```

Update soldiers | Select soldiers | Rollback

	SID	FIRSTNAME	LASTNAME	DRAFTDATE	RELEASEDATE
1	1877	Percy	Malone	6/14/2006	12/12/0002
2	1564	Percy	Wincott	12/22/2006	12/12/0002

שאילתה 4 :

```

DELETE FROM participates
WHERE mID = &<name="mission id" type="integer" hint="mission id is
between 0 and 2000">;

```

השימוש בפרמטר

Variables

Name	Value
mission id	6

mission id is between 0 and 2000

OK Cancel Clear

התוצאה שתקבל מהרצת השאילתה היא

```

DELETE FROM participates
WHERE mID = &<name="mission id" type="integer" hint="mission id is betw

Select * FROM participates
WHERE mID = &<name="mission id" type="integer" hint="mission id is betw

rollback;

```

Delete participates | Select participates | Rollback

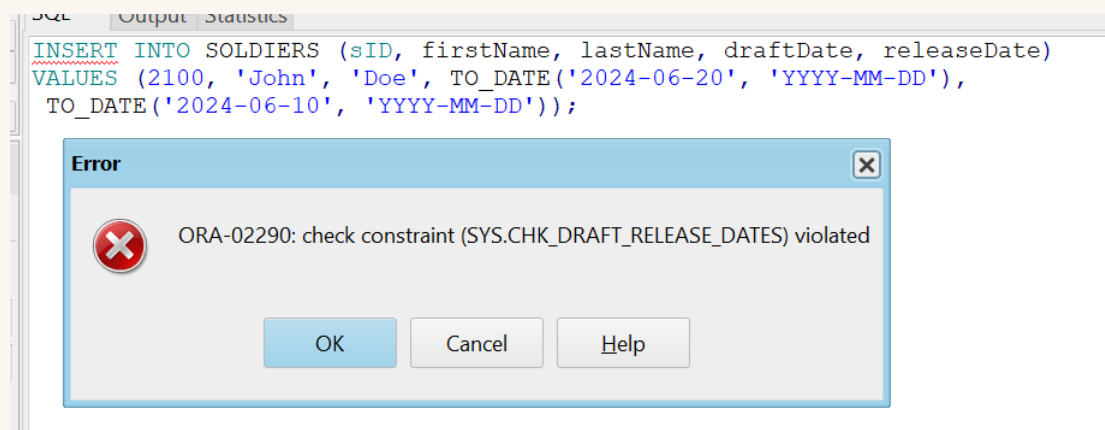
	MID	UNID
--	-----	------

אילוצים

אילוץ 1 : האילוץ מכריח כי לכל חייל תאריך השחרור חייב להיות אחרי תאריך הגיוס

```
ALTER TABLE SOLDIERS
ADD CONSTRAINT chk_draft_release_dates CHECK (draftDate <
releaseDate);
```

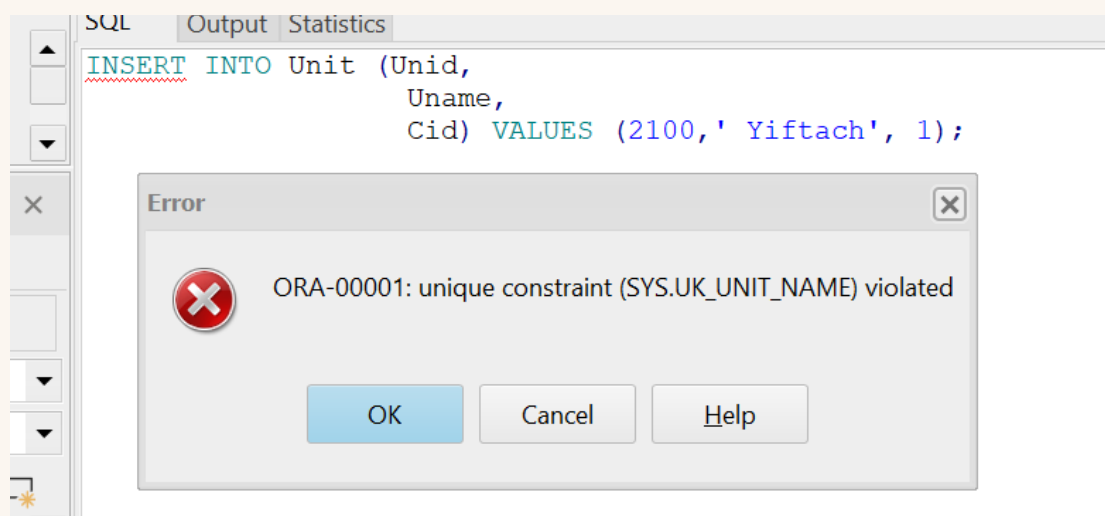
ננסה לבצע פקודה שמפרה את האילוץ



אילוץ 2 : האילוץ מחייב שלכל יחידה יהיה שם מיוחד רק לה

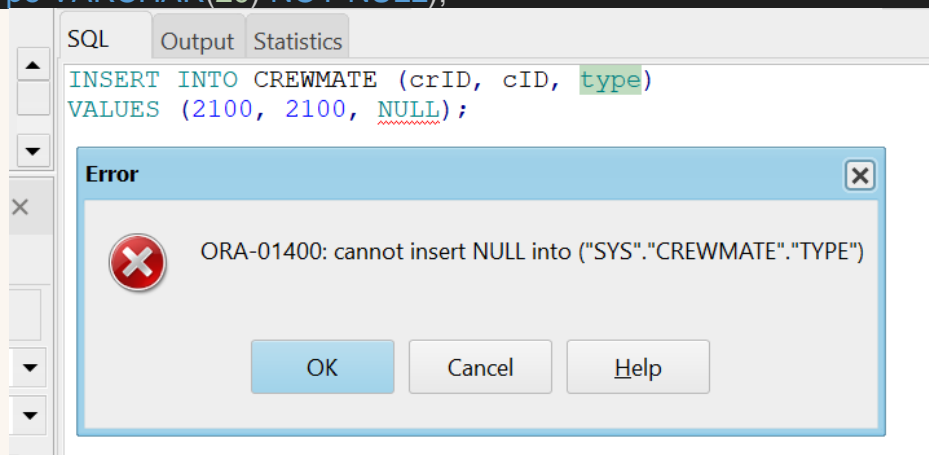
```
ALTER TABLE UNIT
ADD CONSTRAINT uk_unit_name UNIQUE (uName);
```

נכנס קיימת יחידה בשם 'Yiftach'



אילוץ 3 : האילוץ מכריח שלכל איש צוות בטנק חייב להיות תפקיד

```
ALTER TABLE CREWMATE  
MODIFY (type VARCHAR(20) NOT NULL);
```



שלב 3 – תכנות

פונקציות

פונקציה 1 : הפונקציה CountSoldiersInUnits מקבלת מערך של מזהי יחידות ומחזירה מערך של מספרי החיילים בכל יחידה.

```
CREATE OR REPLACE FUNCTION CountSoldiersInUnits (
    p_units SYS.ODCINUMBERLIST
) RETURN SYS.ODCINUMBERLIST IS
    -- הכרזת משתנה לאחסון התוצאה הסופית: מערך של מספרי החיילים בכל יחידה
    v_soldiersCount SYS.ODCINUMBERLIST :=
SYS.ODCINUMBERLIST();

    v_count NUMBER; -- משתנה לאחסון מספר החיילים הזמני
    v_index NUMBER := 1; -- אינדקס עבור הלולאות

    -- מפורש לבחירת מספר החיילים בכל יחידה לפי מזהה יחידה
    CURSOR soldiers_cur IS
        SELECT COUNT(*) AS soldier_count
        FROM SOLDIERS s
        JOIN CREWMATE cr ON s.sID = cr.crID
        JOIN UNIT u ON cr.cID = u.cID
        WHERE u.unID = p_units(v_index);
BEGIN
    -- (Implicit Cursor) לא מפורש
    FOR i IN 1..p_units.COUNT LOOP

        SELECT COUNT(*) * 4 -- בכל טנק יש 4 חיילים
        INTO v_count
        FROM Tank t
        WHERE t.unID = p_units(i);

        -- הוספת מספר החיילים למערך התוצאה
        v_soldiersCount.EXTEND;
        v_soldiersCount(i) := v_count;
    END LOOP;

    RETURN v_soldiersCount;
EXCEPTION
```

```

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
    RETURN v_soldiersCount; -- במקרה של שגיאה, מחזירה רשימה ריקה
END CountSoldiersInUnits;
/

```

פונקציה 2 : הפונקציה IsCommanderOf מקבלת תעודת זהות של חייל ובודקת, האם הוא מפקד על טנק, יחידה, גם וגם או שהוא בכלל לא מפקד.

```

CREATE OR REPLACE FUNCTION IsCommanderOf(p_cID NUMBER)
RETURN VARCHAR2 IS
    v_count_tanks NUMBER := 0;
    v_count_units NUMBER := 0;
    v_result VARCHAR2(50);

BEGIN
    -- בדיקת אם המפקד הוא מפקד של טנק
    SELECT COUNT(*)
    INTO v_count_tanks
    FROM TANK
    WHERE cID = p_cID;

    -- בדיקת אם המפקד הוא מפקד של יחידה
    SELECT COUNT(*)
    INTO v_count_units
    FROM UNIT
    WHERE cID = p_cID;

    -- הסתעפות על פי התוצאות
    IF v_count_tanks > 0 AND v_count_units > 0 THEN
        v_result := 'Commander of both tank and unit';
    ELSIF v_count_tanks > 0 THEN
        v_result := 'Commander of tank';
    ELSIF v_count_units > 0 THEN
        v_result := 'Commander of unit';
    ELSE
        v_result := 'Not a commander';
    END IF;

    RETURN v_result;
EXCEPTION
    WHEN OTHERS THEN

```

```

    RETURN 'Error: ' || SQLERRM;
END IsCommanderOf;
/

```

פונקציה 3 : הפונקציה GetSoldierName מקבלת תעודת זהות של חייל (p_cID) ומחזירה את שמו המלא (שם פרטי ושם משפחה). במקרה שלא נמצא חייל עם תעודת הזהות הנתונה, הפונקציה מחזירה הודעת שגיאה מתאימה.

```

CREATE OR REPLACE FUNCTION GetSoldierName(p_cID NUMBER)
RETURN VARCHAR2 IS
    -- הכרזת משתנים לאחסון שמות הפרטי והמשפחה של החייל
    v_firstName SOLDIERS.firstName%TYPE;
    v_lastName SOLDIERS.lastName%TYPE;
    v_fullName VARCHAR2(50);
BEGIN
    -- שליפת השמות הפרטי והמשפחה של החייל מתוך הטבלה לפי תעודת הזהות של החייל
    SELECT s.firstName, s.lastName
    INTO v_firstName, v_lastName
    FROM SOLDIERS s
    WHERE s.sID = p_cID;

    -- שילוב השם הפרטי ושם המשפחה לשם מלא
    v_fullName := v_firstName || ' ' || v_lastName;

    -- החזרת השם המלא
    RETURN v_fullName;
EXCEPTION
    -- טיפול במקרה שלא נמצא חייל עם המזהה הנתון
    WHEN NO_DATA_FOUND THEN
        RETURN 'Soldier not found';
    -- טיפול בכל שגיאה אחרת
    WHEN OTHERS THEN
        RETURN 'Error: ' || SQLERRM;
END GetSoldierName;
/

```

פרוצדורות

פרוצדורה 1 : הפרוצדורה CreateNewMissionWithUnits יוצרת משימה חדשה עם תאריך נתון ומוסיפה לparticipates יחידות שמתתפות במשימה. הפרוצדורה מקבלת מערך של מזהי יחידות (p_units) ותאריך המשימה (p_mDate).

```
CREATE OR REPLACE PROCEDURE CreateNewMissionWithUnits (
  p_units IN SYS.ODCINUMBERLIST,
  p_mDate IN DATE
) IS
  v_mID NUMBER;
BEGIN
  -- יצירת מזהה חדש למשימה
  SELECT NVL(MAX(mID), 0) + 1 INTO v_mID FROM MISSION;

  -- הוספת המשימה החדשה לטבלת המשימות
  INSERT INTO MISSION (mdate, mID)
  VALUES (p_mDate, v_mID);

  -- הוספת היחידות המתתפות
  FOR i IN 1..p_units.COUNT LOOP
    INSERT INTO participates (mID, unID)
    VALUES (v_mID, p_units(i));
  END LOOP;

  COMMIT;

  -- הצגת הודעה על יצירת משימה חדשה
  DBMS_OUTPUT.PUT_LINE('New mission created with ID: '
  || v_mID);
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END CreateNewMissionWithUnits;
/
```

פרוצדורה 2 : הפרוצדורה AssignTankToUnit מעבירה טנק מיחידת האם ליחידה חדשה, ומחזירה מחזירה Ref Cursor עם מידע על הטנק שעודכן.

```
CREATE OR REPLACE PROCEDURE AssignTankToUnit (  
    p_tID NUMBER,  
    p_unID NUMBER,  
    p_tankInfo OUT SYS_REFCURSOR  
) IS  
BEGIN  
    -- עדכון יחידת הטנק  
    UPDATE TANK  
    SET unID = p_unID  
    WHERE tID = p_tID;  
  
    -- פתיחת Ref Cursor להחזרת מידע על הטנק שעודכן--  
    OPEN p_tankInfo FOR  
        SELECT tID, unID, cID  
        FROM TANK  
        WHERE tID = p_tID;  
  
    COMMIT;  
  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE('Tank not found');  
    WHEN OTHERS THEN  
        ROLLBACK;  
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);  
END AssignTankToUnit;  
/
```

תוכניות ראשיות

תכנית ראשית 1 :

- יוצרת משימה חדשה עם יחידות באמצעות CreateNewMissionWithUnits.
- מחשבת את מספר החיילים בכל יחידה באמצעות CountSoldiersInUnits.
- מציגה את מספר החיילים בכל יחידה.
- סוכמת את סך כל החיילים בכל היחידות.
- מוצאת ומציגה את היחידה עם מספר החיילים הגבוה ביותר.

```

DECLARE
    v_units SYS.ODCINUMBERLIST := SYS.ODCINUMBERLIST(1, 2,
3); -- מערך מזהי יחידות לדוגמה
    v_soldiersCount SYS.ODCINUMBERLIST;
    v_missionDate DATE := SYSDATE; -- תאריך המשימה
    v_totalSoldiers NUMBER := 0;
    v_maxSoldiersUnit NUMBER := 0;
    v_maxSoldiersCount NUMBER := 0;
BEGIN
    -- יצירת משימה חדשה עם היחידות הנתונות
    CreateNewMissionWithUnits(v_units, v_missionDate);

    -- קבלת מספר החיילים בכל יחידה
    v_soldiersCount := CountSoldiersInUnits(v_units);

    -- חישובים והצגת מספר החיילים בכל יחידה
    FOR i IN 1..v_soldiersCount.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE('Unit ' ||
            v_units(i) ||
            ' has ' ||
            v_soldiersCount(i) ||
            ' soldiers. ');
        v_totalSoldiers :=
            v_totalSoldiers + v_soldiersCount(i);

        IF v_soldiersCount(i) > v_maxSoldiersCount THEN
            v_maxSoldiersCount := v_soldiersCount(i);
            v_maxSoldiersUnit := v_units(i);
        END IF;
    END LOOP;

```

```

-- הצגת סך כל החיילים
DBMS_OUTPUT.PUT_LINE('Total soldiers in mission: ' ||
  v_totalSoldiers);

-- הצגת היחידה עם מספר החיילים הגבוה ביותר
DBMS_OUTPUT.PUT_LINE('Unit with most soldiers: ' ||
  v_maxSoldiersUnit ||
  ' (' ||
  v_maxSoldiersCount ||
  ' soldiers)');
END;
/

```

הoutput של הרצת התוכנית

```

New mission created with ID: 2001
Unit 1 has 4 soldiers.
Unit 2 has 8 soldiers.
Unit 3 has 12 soldiers.
Total soldiers in mission: 24
Unit with most soldiers: 3 (12 soldiers)
|

```

תכנית ראשית 2 :

- בודקת אם חייל הוא מפקד של טנק או יחידה באמצעות `.IsCommanderOf`.
- מציגה את שמו המלא של החייל באמצעות `.GetSoldierName`.
- מעבירה טנק ליחידה חדשה באמצעות `.AssignTankToUnit`.
- מציגה את המידע על הטנק שעודכן.
- סוכמת ומציגה את מספר הטנקים והיחידות שהחייל מפקד עליהם.
- בודקת ומציגה אם החייל לא מפקד על טנקים או יחידות.

```

DECLARE
  v_cID NUMBER := 123; -- תעודת זהות של חייל לדוגמה
  v_tID NUMBER := 1; -- מזהה טנק לדוגמה
  v_unID NUMBER := 2; -- מזהה יחידה חדשה לדוגמה
  v_tankInfo SYS_REFCURSOR; -- משתנה להחזקת המידע על הטנק

```



```

v_result VARCHAR2(50); -- משתנה להחזקת תוצאת הפונקציה
IsCommanderOf
v_fullName VARCHAR2(50); -- משתנה להחזקת שם מלא של החייל
v_tankRow TANK%ROWTYPE; -- משתנה להחזקת רשומת טנק
v_tankCount NUMBER := 0;
v_unitCount NUMBER := 0;
BEGIN
    -- IsCommanderOf קריאה לפונקציה
    v_result := IsCommanderOf(v_cID);

    DBMS_OUTPUT.PUT_LINE('Commander status: ' ||
v_result);

    -- GetSoldierName קריאה לפונקציה
    v_fullName := GetSoldierName(v_cID);

    DBMS_OUTPUT.PUT_LINE('Soldier name: ' || v_fullName);

    -- קריאה לפרוצדורה AssignTankToUnit והצגת מידע על הטנק שעודכן
    AssignTankToUnit(v_tID, v_unID, v_tankInfo);

    LOOP
        FETCH v_tankInfo INTO v_tankRow;
        EXIT WHEN v_tankInfo%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Tank ID: ' ||
            v_tankRow.tID ||
            ', Unit ID: ' ||
            v_tankRow.unID ||
            ', Commander ID: ' ||
            v_tankRow.cID);
        v_tankCount := v_tankCount + 1;
    END LOOP;

    CLOSE v_tankInfo;

    -- הצגת מספר הטנקים של המפקד
    DBMS_OUTPUT.PUT_LINE('Total tanks commanded by ' ||
        v_fullName || ': ' || v_tankCount);

    -- הצגת מספר היחידות של המפקד
    BEGIN
        SELECT COUNT(*)

```

```

    INTO v_unitCount
  FROM UNIT
 WHERE cID = v_cID;

  DBMS_OUTPUT.PUT_LINE('Total units commanded by ' ||
    v_fullName || ': ' || v_unitCount);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error in counting units: '
|| SQLERRM);
  END;

  -- בדיקת אם המפקד לא מפקד על טנקים או יחידות
  IF v_tankCount = 0 AND v_unitCount = 0 THEN
    DBMS_OUTPUT.PUT_LINE(v_fullName ||
      ' is not a commander of any tanks or units.');
```

END IF;

END;

/

הoutput של הרצת התוכנית

```

Commander status: Commander of tank
Soldier name: Rachael Eat World
Tank ID: 1, Unit ID: 2, Commander ID: 1
Total tanks commanded by Rachael Eat World: 1
Total units commanded by Rachael Eat World: 0
|
```