

מיני פרויקט בבסיסי נתונים

אגף חיל השריון של צה"ל



מגישים: מאיר שאנני - 216241596

אריאל סיון - 329524839

מרצה: מר יעקב ברזילי

קישור לגיטהאב: <https://github.com/trhtk242/minipbasnat>

תוכן

3	שלב 1 - עיצוב ויצירת DB, שימוש בERDPLUS 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 ב3NF
7	קוד SQL של יצירת הטבלאות (CREATE)
9	תיאור הטבלאות (DESC)
10	קוד SQL של זריקת הטבלאות (DROP)
10	הכנסת מידע לטבלאות
10	הכנסת מידע לSOLIDERS – דרך 1, Data Generator
12	הכנסת מידע לCOMMANDER – דרך 1, Data Generator
13	הכנסת מידע לCREWMATE – דרך 2, סקריפט בפיתוח
14	הכנסת מידע לUNIT – דרך 3, קובץ טקסט
16	הכנסת מידע לTANK – דרך 2, סקריפט בפיתוח
17	הכנסת מידע לMISSION - דרך 4, קובץ EXCEL
18	הכנסת מידע לparticipate – דרך 1, Data Generator
19	גיבוי הטבלאות
20	שחזור הטבלאות

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

תיאור המערכת

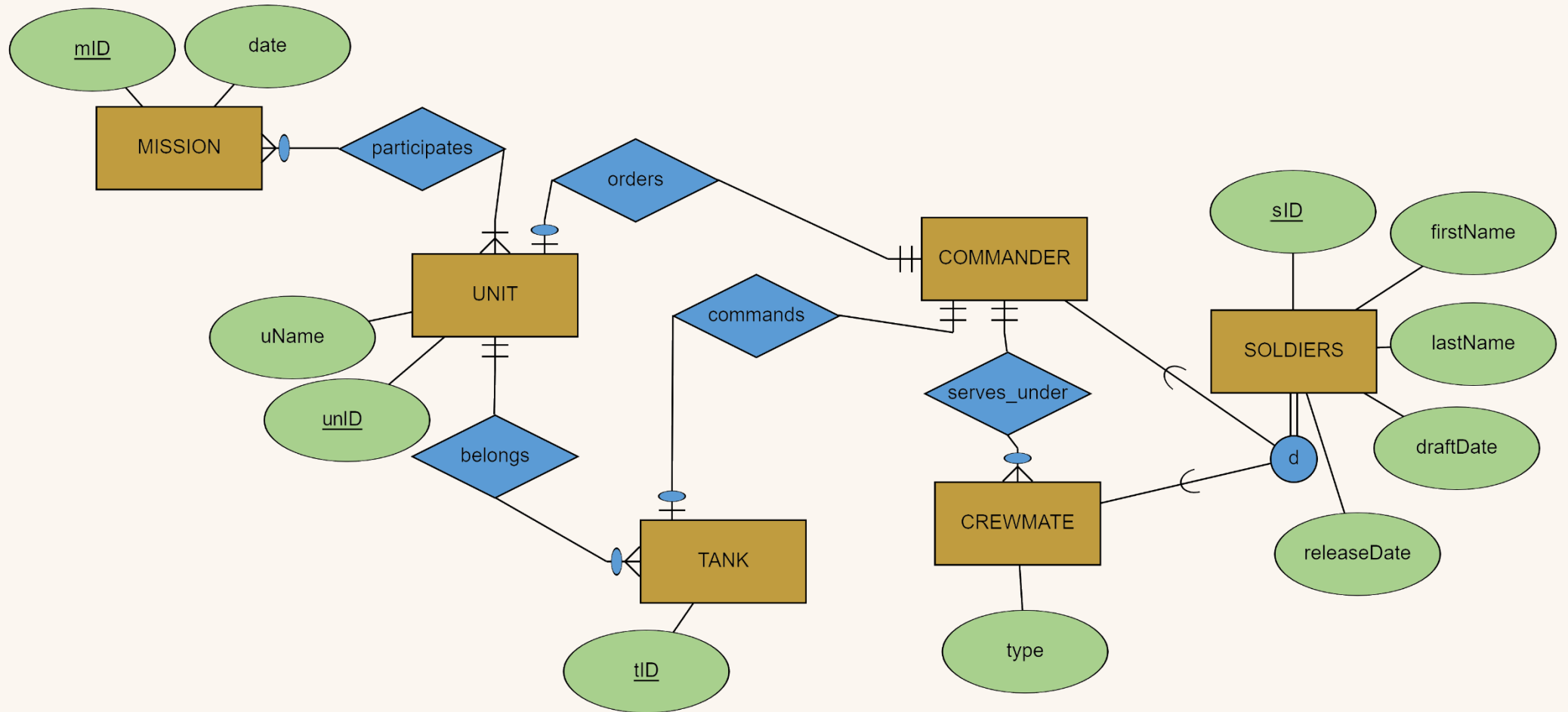
זרוע השריון הישראלית, המוכרת גם כחיל השריון (חש"ן), עומדת כחוד החנית של צבא הגנה לישראל (צה"ל). זרוע זו, מהווה כוח התקפי עיקרי בשדה הקרב, ומשלבת טכנולוגיה מתקדמת, אומץ לב ורוח לחימה עזה.

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.

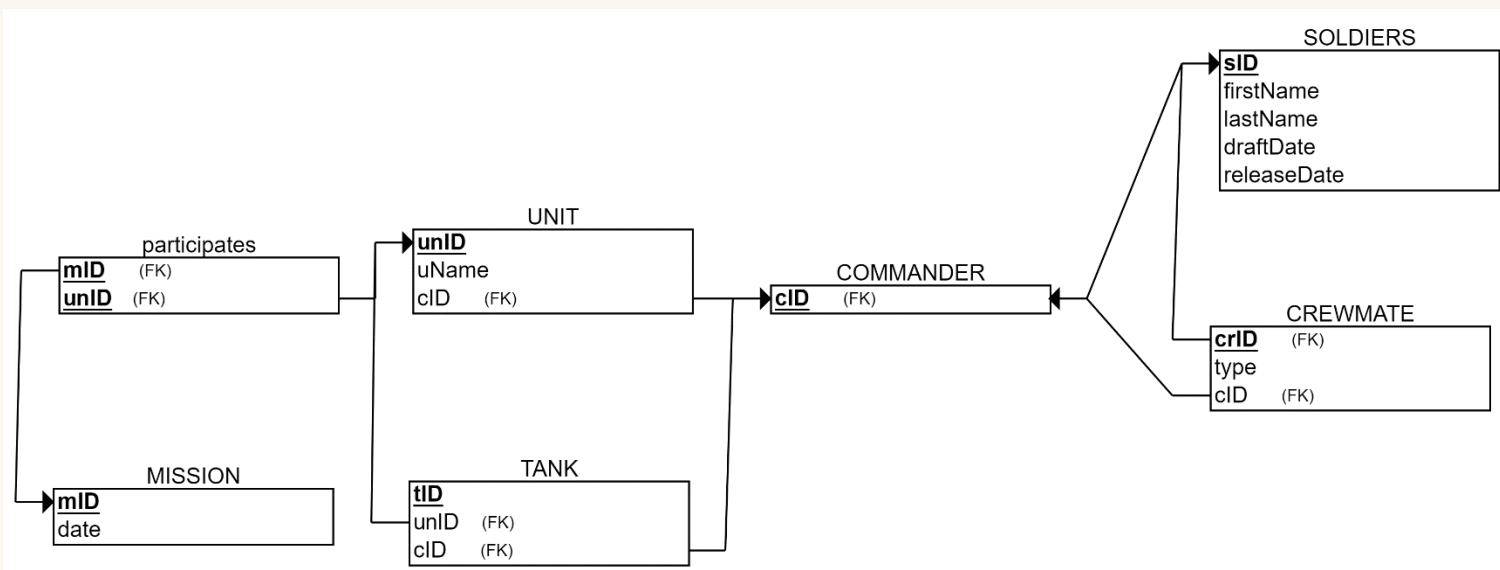
המערכת שנבנה תספק בסיס נתונים לחיל השריון. היא תתחלק לשלושה חלקים:

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 הישויות והקשרים



תרשים DSD



תיאור הישויות והתכונות שלהן

קבוצה של כל החיילים	SOLIDERS (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>tlD</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(tlD,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 cID וגם crID גם קשור ישירות ל-cID. כמפתח זר. הטבלה ב3NF.

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

TANK (טנק) – uID 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 uID 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)

```
CREATE TABLE SOLDIERS
(
    sID NUMBER(9) NOT NULL,
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,
    draftDate DATE NOT NULL,
    releaseDate DATE NOT NULL,
    PRIMARY KEY (sID)
);

CREATE TABLE MISSION
(
    mdate DATE NOT NULL,
    mID NUMBER(9) NOT NULL,
    PRIMARY KEY (mID)
);
```

```
CREATE TABLE COMMANDER
(
    cID NUMBER(9) NOT NULL,
    PRIMARY KEY (cID),
    FOREIGN KEY (cID) REFERENCES SOLDIERS(sID)
);

CREATE TABLE CREWMATE
(
    type VARCHAR(20) NOT NULL,
    crID NUMBER(9) NOT NULL,
    cID NUMBER(9) NOT NULL,
    PRIMARY KEY (crID),
    FOREIGN KEY (crID) REFERENCES SOLDIERS(sID),
    FOREIGN KEY (cID) REFERENCES COMMANDER(cID)
);

CREATE TABLE UNIT
(
    unID NUMBER(9) NOT NULL,
    uName VARCHAR(20) NOT NULL,
    cID NUMBER(9) NOT NULL,
    PRIMARY KEY (unID),
    FOREIGN KEY (cID) REFERENCES COMMANDER(cID)
);

CREATE TABLE TANK
(
    tID NUMBER(9) NOT NULL,
    unID NUMBER(9) NOT NULL,
    cID NUMBER(9) NOT NULL,
    PRIMARY KEY (tID),
    FOREIGN KEY (unID) REFERENCES UNIT(unID),
    FOREIGN KEY (cID) REFERENCES COMMANDER(cID)
);

CREATE TABLE participates
(
    mID NUMBER(9) NOT NULL,
    unID NUMBER(9) NOT NULL,
    PRIMARY KEY (mID, unID),
    FOREIGN KEY (mID) REFERENCES MISSION(mID),
```



```
FOREIGN KEY (unID) REFERENCES UNIT(unID)
);
```

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 הData בעזרת מחולל המידע של Oracle (Data Generator)

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)

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 Data ל SQL INSERT לפקודות

```
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('03-09-2009', 'dd-mm-yyyy'))

insert into SYS.SOLDIERS (SID, FIRSTNAME, LASTNAME, DRAFTDATE, RELEASEDATE)
```

נבדוק 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)

Name	Type	Size	Data
CID	NUMBER	9	Sequence(1)

בדוק האם המידע נקלט בעזרת הפקודה
Select * from COMMANDER

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

```
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)
```

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.

```
import random

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

def generate_crewmate_data(num_records):
    data = []
    for x in range(num_records):
        id = x + 601
        cid = int((id - 600) / 4 ) + 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 = 1320 # 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', rewmate_data)
```

נכניס את שורות הInsertn אל לתוך מסך הנתונים

SQL	Output	Statistics
<pre> INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (604,2,'Loader'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (605,2,'Gunner'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (606,2,'Driver'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (607,2,'Loader'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (608,3,'Gunner'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (609,3,'Driver'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (610,3,'Loader'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (611,3,'Gunner'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (612,4,'Driver'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (613,4,'Loader'); INSERT INTO CREWMATE (CRID, CID, TYPE) VALUES (614,4,'Gunner'); </pre>		

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

End at line-end: ☒ Name in header: ☐ Skip empty lines: ☒

Quote character: " Comment line: Import lines: 1 ..

Field2 (+0 .. ",") Field3 (+0 .. ",")

relative position: Absolute position: Character: Field End: Length: Character:

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	Ikvot 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 ל-TANK – דרך 2, סקריפט בפייתון

```
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)
```

הסקריפט יצור לנו 350 פקודות INSERT ב-TANK

```
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 שאלתת
 select * from TANK;

	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

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

הכנסת מידע ל-MISSION - דרך 4, קובץ Excel

נשתמש במידע מתוך קובץ Excel

נפתח את קובץ Excel בעזרת ODBC Importer

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

- Connection:**
 - User / System DSN: Excel Files
 - User Name: asivan
 - Password: (masked with dots)
 - Buttons: Connect, Disconnect
- Table / Query:**
 - Import Table (selected)
 - View Data button
- 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:

Commit every...: ☒ Overwrite duplicates ☐ Ignore duplicates ☐ Delete records ☐ Truncate table

Initializing Script:

Finalizing Script:

Fields

MDATE -> MDATE
MID -> MID

Field: Fieldtype:

Result Preview

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

נבדוק שהData אכן עבר מהExcel אל מסד הנתונים

	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

הכנסת מידע לparticipate – דרך 1, Data Generator

נשתמש בData Generator בשביל ליצור משימות ולהכניס אותן אל בסיס הנתונים

PARTICIPATES

< Owner: Table: Number of records:

>

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);
```

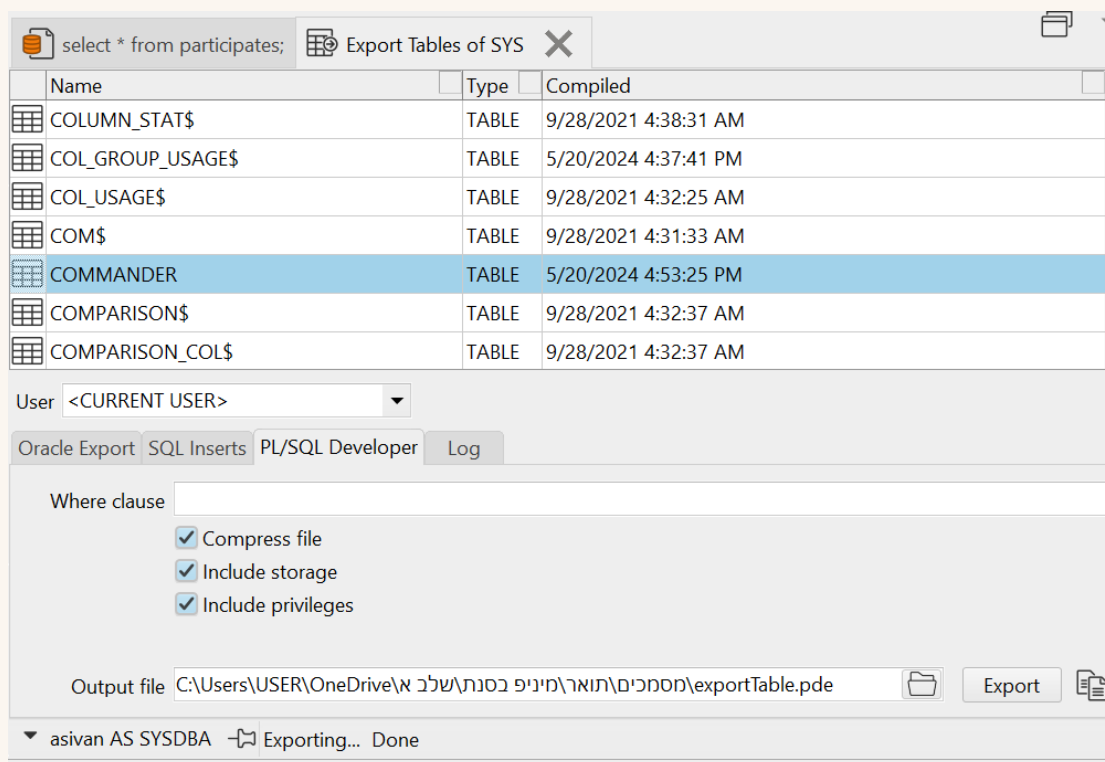
אחרי שהרצנו את הפקודות בתמונה הקודמת,
נבדוק בעזרת פקודת Select האם המידע אכן
נקלט

	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

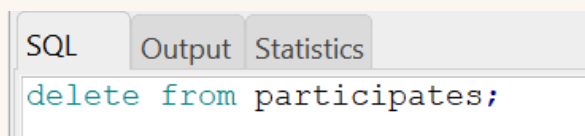
בשביל להבטיח שהמידע של בסיס הנתונים שלנו תמיד נשמר נצטרך לגבות את כל
המידע שבתוך הטבלאות למקור שמירה חיצוני.

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

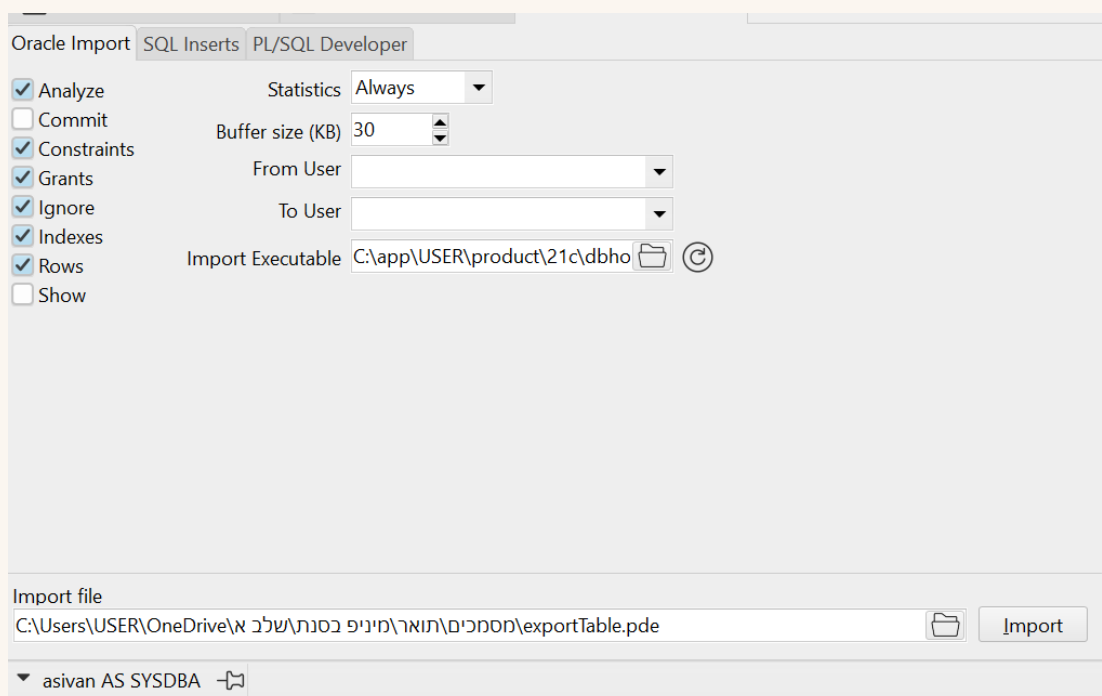


שחזור הטבלאות

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