מיני פרויקט בבסיסי נתונים

מגישים:

רז ליבוביץ: 209521863

יונתן קליין: 322961764

שלב א - תיאור המערכת

תיאור מילולי

בפרויקט נבנה בסיס נתונים עבור רכבי צה"ל.

הרעיון הוא לנהל את הרכבים של צה"ל ולשייך כל רכב לבסיס שלו, בנוסף איזה טיפולים הרכב עב ר

ואיזה משימות יש בשבילו וגם יחזיק תיעוד של תאונות שהרכב עבר.

בנוסף הבסיס נתונים ינהל את כל החיילים)מכונאי, נה ג, וכו'(וייתן להם את התפקידים הדרושים בשבילם אם זה לטפל ברכב או לקחת את הרכב עבור משימה מיוחדת.

תיאור מילולי של הטבלאות

Accident(AccidentID, DateOfAccident, DamageLevel, VehicleID)

Base(BaseID, BaseName, Location)

MaintenanceRecord(RecordID, DateOfMaintenance, Description, VehicleID)

Soldier(SoldierID, Name, Rank, DateOfBirth)

Trip(TripID, TripDate, Destination, VehicleID, SoldierID)

DriveAt(VehicleID, SoldierID)

Table name	Attribute	Explanation (where needed)
Vehicle	VehicleID	מפתח ראשי)Primary Key(של הטבלה, מזהה ייחודי לכל כלי רכב.
	Model	הדגם
	AcquisitionDate	תאריך רכישת הרכב

	Status	הסטטוס הנוכחי של כלי הרכב
	BaseID	מפתח זר לבסיס
Accident	AccidentID	מפתח
	DateOfAccident	תאריך התאונה
	DamageLevel	רמת הנזק 1-10
	VehicleID	מפתח זר לטבלת Vehicle
Base	BaseID	מפתח
	BaseName	שם הבסי ס
	Location	מיקום הבסי ס
MaintenanceRecord	RecordID	מפתח
	DateOfMaintenace	תאריך התוחזקה
	Description	תיאור התחזוקה שבוצעה
	VehcileID	מפתח זר לרכבים
	SoldierID	מפתח זר לחיילי ם
Soldier	SoldierID	מפתח
	Name	שם החייל
	Rank	דרגת החיי ל
	DateOfBirth	תאריך הלידה של החיי ל
Trip	TripID	מפתח
	TripDate	תאריך של הנסיעה
	Destination	יעד הנסיעה
	VehicleID	מפתח זר לרכבי ם
	SoldierID	מפתח זר לחיילי ם

מפתח זר לרכבים	VehcileID	DriveAt
מפתח זר לחיילי ם	SoldierID	

הסבר על נרמול NF3

נפרט על כל אחת מהטבלאות:

Vehicle

VehicleID הוא המפתח הראשי.

Model, AcquisitionDate, Status, ו-BaseID תלויות ישירות במפתח הראשי VehicleID. BaseID הוא מפתח זר שמצביע על BaseID בטבלת Base, אך התלות היא ישירה)לא טרנזיטיבית(, כיוון שהוא מציין את הבסיס אליו שייך הרכב.

Accident

הוא המפתח הראשי.

AccidentID

DateOfAccident, DamageLevel, ו-VehicleID תלויות ישירות במפתח הראשי AccidentID. VehicleID הוא מפתח זר שמצביע על VehicleID בטבלת Vehicle, אך התלות היא ישירה כיוון שהוא מציין את הרכב שהיה מעורב בתאונה.

Base

BaseID הוא המפתח הראשי.

BaseName ו-Location תלויות ישירות במפתח הראשי BaseID.

הוא . MaintenanceRecord)רשומת תחזוקה(:

RecordID המפתח הראשי

DateOfMaintenance, Description, VehicleID, ו-SoldierID תלויות ישירות במפתח הראשי RecordID.

VehicleID ו-SoldierID הם מפתחות זרים שמצביעים על VehicleID בטבלת Vehicle ועל SoldierID בטבלת Soldier בהתאמה.

Soldier

SoldierID הוא המפתח הראשי.

Name, Rank, ו-DateOfBirth תלויות ישירות במפתח הראשי SoldierID.

Trip

TripID הוא המפתח הראשי.

TripDate, Destination, VehicleID, ו-SoldierID תלויות ישירות במפתח הראשי TripID.

VehicleID ו-SoldierID הם מפתחות זרים שמצביעים על VehicleID בטבלת Vehicle ועל SoldierID בטבלת Soldier בהתאמה. הסבר על הקשרים:

HadAccident

VehicleID ו-AccidentID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Vehicle ו-Accident בהתאמה.

אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

BelongsTo

VehicleID ו-BaseID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Vehicle ו-Base בהתאמה.

אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

HadMaintenance

VehicleID ו-RecordID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Vehicle ו -MaintenanceRecord בהתאמה.

אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

DriveAt

VehicleID ו-SoldierID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Vehicle ו-Soldier בהתאמה.

אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

HadTrip

VehicleID ו -TripID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Vehicle ו-Trip בהתאמה.

אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

HasTrips:

SoldierID ו -TripID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Soldier ו-Trip בהתאמה.

אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

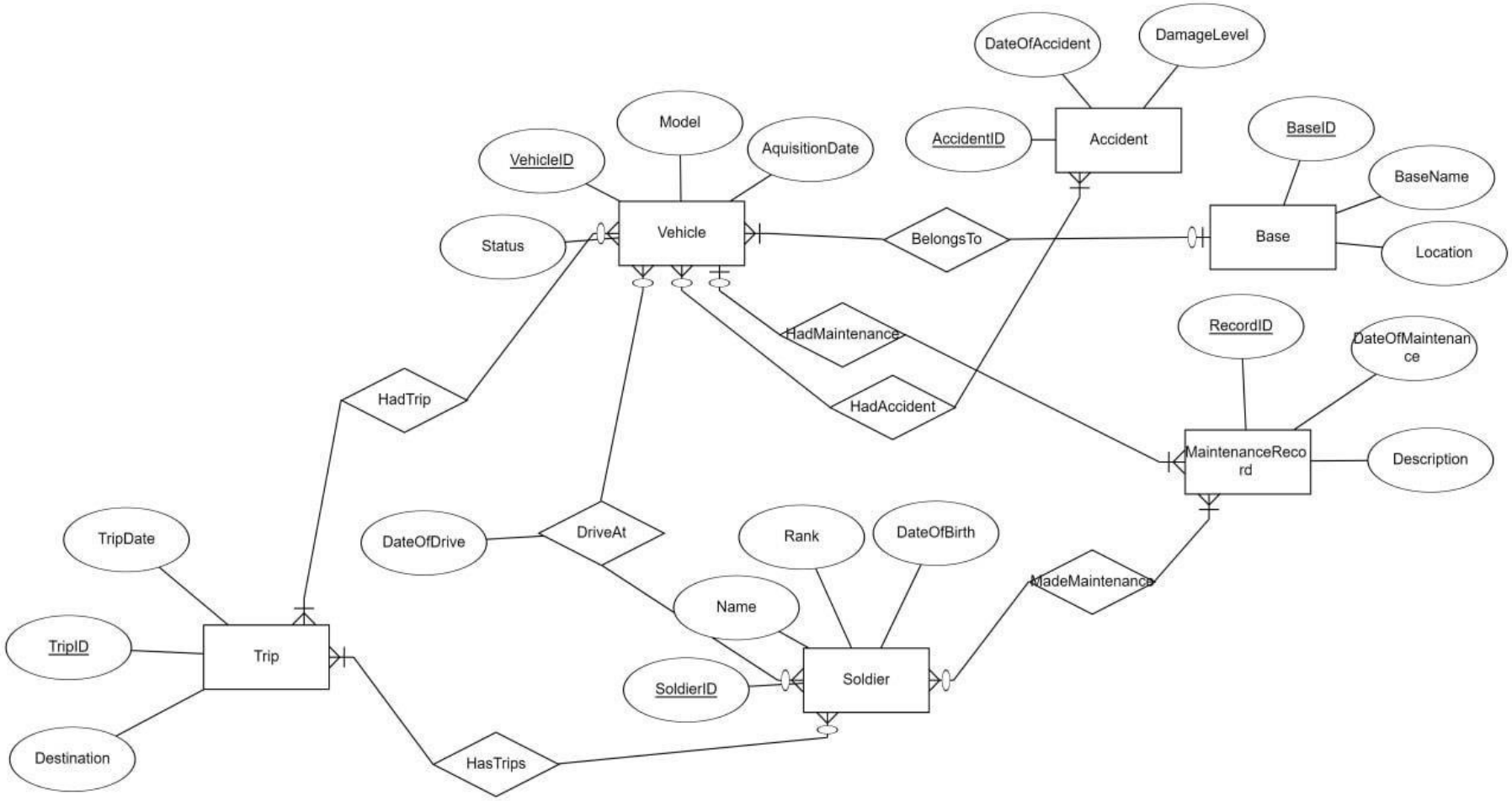
HadeMaintenance

SoldierID ו-RecordID הם מפתחות זרים שמצביעים על המפתחות הראשיים בטבלאות Soldier ו-MaintenanceRecord בהתאמה.

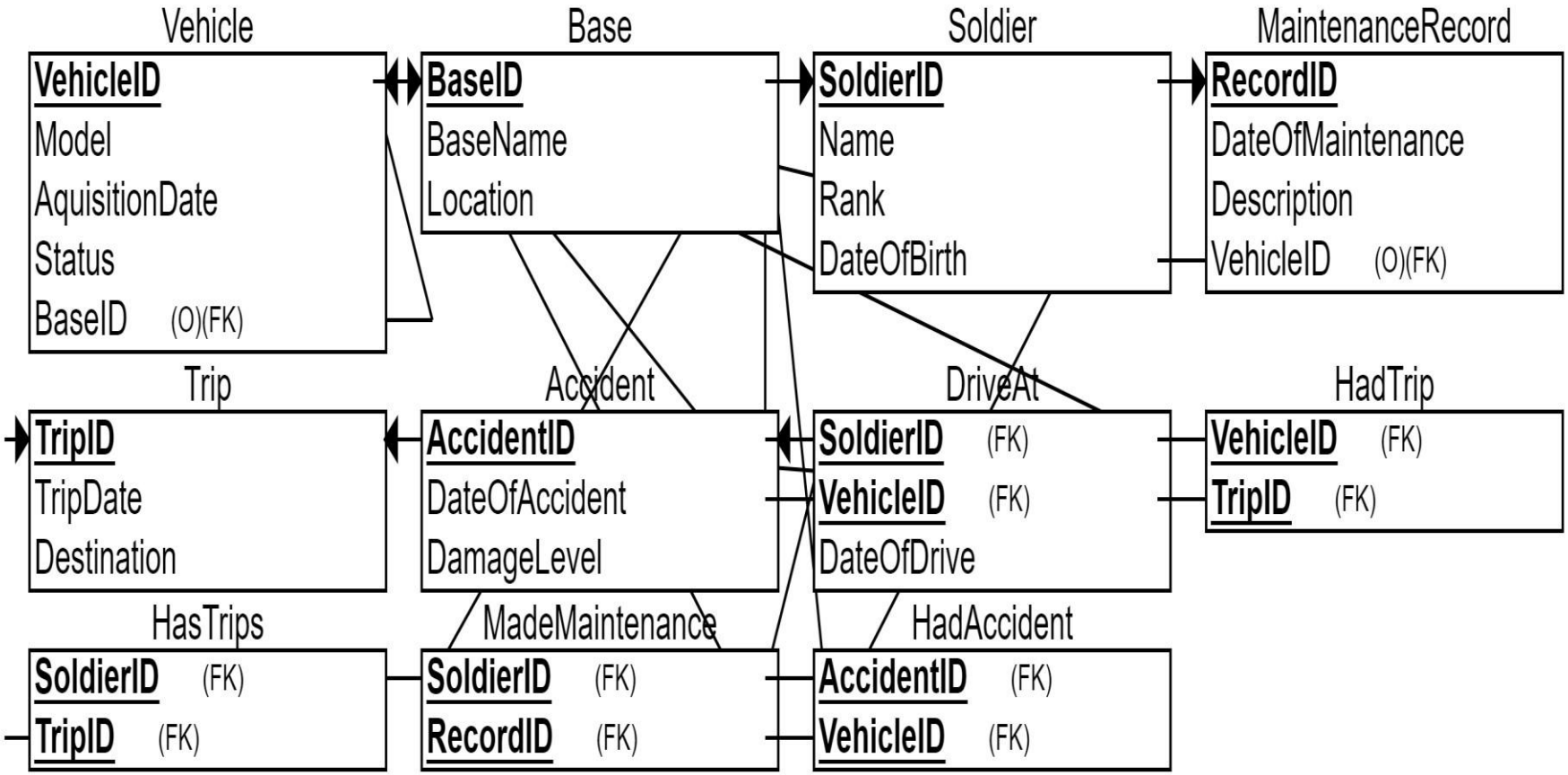
אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

בדרך זו, הטבלאות והקשרים מתארים מערכת נתונים מתוקננת שעומדת בתנאי נורמליזציה NF3, מבטיחה שהמידע הוא עקבי ומונעת כפילויות ותלותות פונקציונליות בלתי רצויות.

דיאגרמת ERD


DateOfAccident
DamageLevel
Model
AquisitionDate
AccidentID
Accident
BaseID
VehicleID
BaseName
Status
Vehicle
BelongsTo
Base
Location
HadMaintenance
RecordID
DateOfMaintenance
HadTrip
HadAccident
MaintenanceRecord
Description
TripDate
DateOfDrive
DriveAt
Rank
DateOfBirth
MadeMaintenance
Name
TripID
Trip
SoldierID
Soldier
Destination
HasTrips

תרשים DSD


Vehicle
VehicleID
Model
AquisitionDate
Status
BaseID (O)(FK)
Base
BaseID
BaseName
Location
Soldier
SoldierID
Name
Rank
DateOfBirth
MaintenanceRecord
RecordID
DateOfMaintenance
Description
VehicleID (O)(FK)
Trip
TripID
TripDate
Destination
Accident
AccidentID
DateOfAccident
DamageLevel
DriveAt
SoldierID (FK)
VehicleID (FK)
DateOfDrive
HadTrip
VehicleID (FK)
TripID (FK)
HasTrips
SoldierID (FK)
TripID (FK)
MadeMaintenance
SoldierID (FK)
RecordID (FK)
HadAccident
AccidentID (FK)
VehicleID (FK)

יצירת ואכלוס הטבלאות

במסמך הדרישות לפרויקט במודל התבקשנו לכתוב הסבר למפוי ה ERDלטבלאו ת

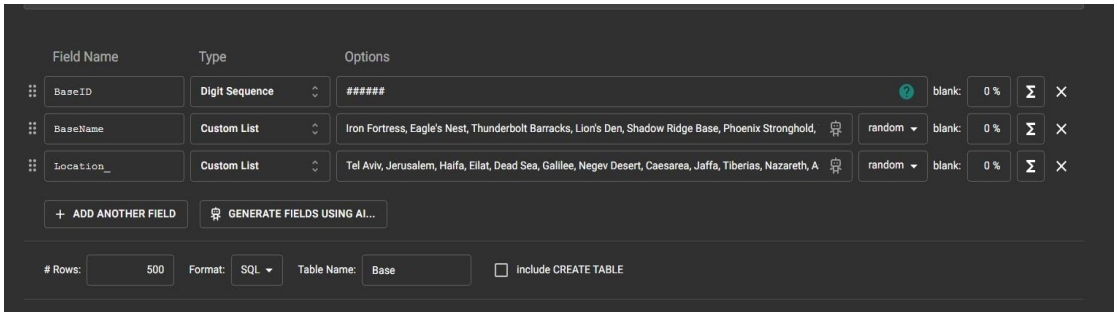
נראה דוגמא ל3 שיטות שהכנסנו דרכם את הנתונים.

Base

1. *יצירת הטבלה*

```
CREATE TABLE Base
(
  BaseID INT NOT NULL,
  BaseName VARCHAR2(30) NOT NULL,
  Location_ VARCHAR2(30) NOT NULL,
  PRIMARY KEY (BaseID)
);
```

2. *הכנסת נתונים ע"י אתר mackaroo*



3. *הצגת הנתונים ע"י פקודת select **

	BASEID	BASENAME	LOCATION_
1	716411	Wolfpack Camp	Galilee
2	299961	Lion's Den	Ramat Gan
3	21950	Thunderbolt Barracks	Holon
4	618595	Shadow Ridge Base	Tel Aviv
5	155077	Spartan Citadel	Nesher
6	522468	Thunderbolt Barracks	Qiryat Ono
7	611132	Wolfpack Camp	Negev Desert
8	656992	Phoenix Stronghold	Tzfat
9	755215	Wolfpack Camp	Tel Aviv
10	596693	Wolfpack Camp	Ashdod
11	647973	Spartan Citadel	Caesarea
12	286784	Thunderstrike Outpost	Ashkelon
13	480786	Iron Fortress	Or Yehuda
14	238744	Shadow Ridge Base	Eilat
15	267671	Lion's Den	Qiryat Motzkin
16	880598	Lion's Den	Lod
17	741586	Lion's Den	Petah Tikva
18	250479	Shadow Ridge Base	Acre
19	531268	Shadow Ridge Base	Beit She'an
20	713497	Shadow Ridge Base	Kiryat Malachi
21	865889	Wolfpack Camp	Ra'anana
22	426820	Eagle's Nest	Tiberias

Vehicle

1. יצירת הטבלה

```oracle
CREATE TABLE Vehicle
(
  VehicleID INT NOT NULL,
  Model_ VARCHAR2(30) NOT NULL,
  AquisitionDate DATE NOT NULL,
  Status VARCHAR2(30) NOT NULL,
  BaseID INT,
  PRIMARY KEY (VehicleID),
  FOREIGN KEY (BaseID) REFERENCES Base(BaseID)
);
```

2. הכנסת נתונים ע"י data generator

Name	Type	Size	Data	Master
VEHICLEID	NUMBER		[111111]	
MODEL_	VARCHAR2	30	List(Merkava Mk.4, Merkava Mk.3, Merkava Mk.2, Merkava Mk.1, Namer, Eitan, Achzarit, Magach 7, Magach 6, Magach 5, Sho't Kal, M113 Zelda, Nagmachon, Puma, Nakpadon, IDF Caterpillar D9, M109 Doher, Soltam M-71, M270 MLRS, ATMOS 2000, Sholef, Pereh, Tiran 4, Tiran 5, Tir	
AQUISITIONDATE	DATE		Random(1/1/1970, 1/5/2024)	
STATUS	VARCHAR2	30	List(Active, In Maintenance, Deployed, Out of Service, Under Repair)	
BASEID	NUMBER		List(select BaseID from Base)	

3. הצגת הנתונים ע"י פקודת select *

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
1	747914	Eitan	28/03/1978	In Maintenance	513500
2	381875	Tiran 6	15/10/1979	Out of Service	710007
3	817156	Tiran 5	07/06/1992	Under Repair	701603
4	468589	Tiran 5	05/12/1992	Under Repair	962591
5	452793	IDF Caterpillar D9	21/10/2013	Active	914132
6	292526	Pereh	25/06/2002	Out of Service	714309
7	431721	Magach 6	14/03/1975	Under Repair	186088
8	396138	M113 Zelda	17/07/1991	Active	484085
9	125596	M88 Hercules	06/01/2024	In Maintenance	255941
10	571538	RBY Mk 1	09/09/1983	Active	38609
11	543871	Magach 7	17/05/1985	Under Repair	368287
12	248842	Pereh	07/10/1990	Active	491038
13	641271	IDF Caterpillar D9	24/12/2006	Deployed	864124
14	713789	Tiran 6	02/02/2009	In Maintenance	756250
15	638187	Magach 7	09/01/1982	In Maintenance	647973
16	234455	David Light Armored Vehicle	01/08/1971	In Maintenance	76132
17	548715	M109 Doher	14/06/1973	Deployed	876383
18	271766	LYNX	28/06/2011	In Maintenance	965900
19	765847	Magach 5	23/12/2018	Active	715708
20	262347	Eyal	01/06/1994	Active	688633
21	461769	Merkava Mk.2	06/10/1991	Deployed	515315
22	828374	MANTAK	02/06/2005	Active	255941
23	234349	Merkava Mk.2	01/12/1987	Active	342517
24	732336	Sufa (Storm) Jeep	30/05/2014	In Maintenance	615770
25	744734	LYNX	06/12/1986	Out of Service	551839
26	331426	M270 Smashing	10/03/1999	Out of Service	750009
27	221661	Soltam M-71	08/09/1989	Active	23650
28	161998	Magach 7	22/09/1971	Active	994218
29	528415	M88 Hercules	02/01/2005	Out of Service	171641
30	857933	Doher	16/12/2018	In Maintenance	584448
31	959674	Machbet	07/07/1975	Out of Service	293957
32	189779	Magach 7	18/11/1993	Under Repair	989717
33	483711	Sufa (Storm) Jeep	24/12/2018	Out of Service	699803
34	559421	Panther Command Vehicle	29/03/1973	Under Repair	194232
35	561299	Keshet	04/06/2016	In Maintenance	63195
36	975563	Eyal	29/11/2003	Deployed	74836
37	449468	Zibra	04/08/1982	Under Repair	932705

Soldier

1. יצירת הטבלה

```
CREATE TABLE Soldier
(
  SoldierID INT NOT NULL,
  Name_ VARCHAR2(30) NOT NULL,
  Rank_ VARCHAR2(30) NOT NULL,
  DateOfBirth DATE NOT NULL,
  PRIMARY KEY (SoldierID)
);
```

2. הכנסת נתונים ע"י Python

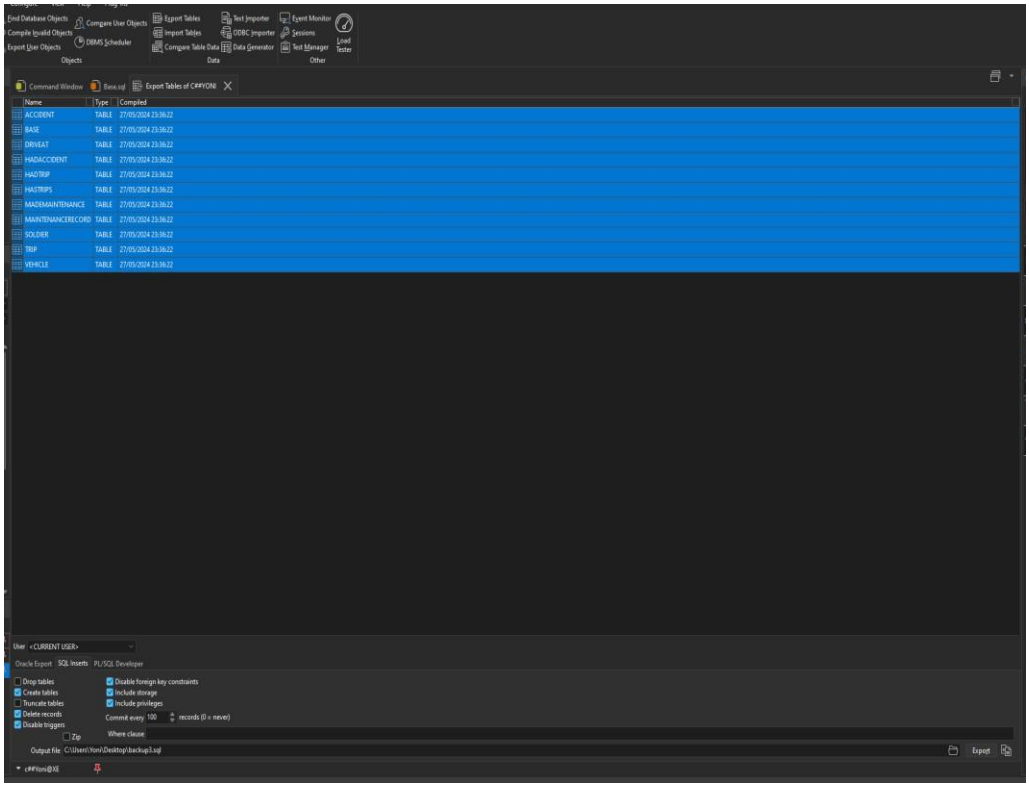
```python
def generate_fake_soldier():
    fake = Faker()
    soldier_id = random.randint(a: 100000, b: 999999)  # Generate random 6-digit SoldierID
    soldier_name = fake.name()
    rank = fake.random_element(elements=('Private', 'Corporal', 'Sergeant', 'Lieutenant', 'Captain'))
    date_of_birth = fake.date_of_birth(minimum_age=18, maximum_age=40)
    return (soldier_id, soldier_name, rank, date_of_birth)


# Function to insert fake data into the database
1 usage
def insert_soldiers(connection, cursor, num_rows):
    for _ in range(num_rows):
        soldier_data = generate_fake_soldier()
        query = "INSERT INTO soldier (SoldierID, Name_, Rank_, DateOfBirth) VALUES (:1, :2, :3, :4)"
        cursor.execute(query, soldier_data)
    connection.commit()
```

3. הצגת הנתונים ע"י פקודת select *

	SOLDIERID	NAME_	RANK_	DATEOFBIRTH
1	843643	Alexis Hopkins	Private	06/03/1992
2	234622	Richard Riley	Lieutenant	04/03/2004
3	372213	Travis Smith	Corporal	14/02/1988
4	423436	Kenneth Mendoza	Lieutenant	14/11/2000
5	943303	Luis Bean	Sergeant	26/01/2003
6	321807	Timothy Kennedy	Captain	26/01/1986
7	941183	Larry Anderson	Corporal	13/01/1992
8	367359	Ronald Patel	Lieutenant	04/02/1995
9	194799	Colton Ellis	Lieutenant	30/11/1991
10	531386	Emily Garcia	Captain	25/05/2002
11	581000	Edward Dominguez	Corporal	18/06/1984
12	489057	Jason Moore	Captain	25/02/1990
13	761979	Patrick Hatfield	Sergeant	06/10/1995
14	422058	Donna Russell	Sergeant	04/07/1995
15	303983	Jason Lyons	Captain	01/09/1984
16	180514	Lisa Rice	Sergeant	10/03/1994
17	890778	Christine Erickson	Sergeant	19/02/2000
18	563675	Marcus Hernandez	Lieutenant	14/07/1996
19	669983	Ryan Nunez	Captain	14/12/2005
20	896121	Katrina Hamilton	Corporal	24/08/1999
21	826512	Joseph Hicks	Private	17/11/1991
22	279562	Brian Frye	Sergeant	22/05/2004
23	996427	Arthur Lee	Lieutenant	05/03/2000
24	593987	Kristin Gomez	Sergeant	19/06/2001
25	856755	Angel Anderson	Private	15/05/1987
26	842562	Kevin Watson	Private	26/08/1998
27	536242	Roberto Nguyen	Sergeant	17/04/1984
28	931299	Tim Tran	Lieutenant	03/11/1983
29	780063	Paul Howard	Corporal	11/03/1992
30	214895	Sophia Mcdowell	Corporal	01/01/1987
31	873487	Joan Smith	Private	20/04/1998
32	554221	Michael Harrell	Corporal	22/03/1997
33	967089	Brian Miller DDS	Lieutenant	25/03/2003
34	530553	Shannon Lee	Captain	06/09/1990
35	944293	Adrian Hanson	Private	12/08/1995
36	426481	Andrew Suarez	Captain	05/05/2000
37	875100	Maria Williams	Sergeant	19/12/1998

צילומי מסך עבור הbackup

1. גיבוי הטבלה



2. שחזור הנתונים

שלב ב

כתיבת שאילתות select:

1. השאילתה מביאה את כל הרכבים שאף פעם לא היו בתאונה.

```
/*List all vehicles that have never been in an accident*/
SELECT V.VehicleID, V.Model_
FROM Vehicle V
LEFT JOIN HadAccident HA ON V.VehicleID = HA.VehicleID
WHERE HA.AccidentID IS NULL;
```

	VEHICLEID	MODEL_
1	122174	Magach 5
2	137139	LYNX
3	142314	Hafiz
4	146385	SandCat
5	152328	Mahatz

2. השאילתה מחזירה את רישום הטיפולים עבור רכב שגדול מ5 שנים.

```
/*Show maintenance history for vehicles older than 5 years*/
SELECT V.VehicleID, V.Model_, MR.DateOfMaintenance, MR.Description_
FROM Vehicle V
JOIN MaintenanceRecord MR ON V.VehicleID = MR.VehicleID
WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM V.AquisitionDate) > 5
ORDER BY MR.DateOfMaintenance;
```

	VEHICLEID	MODEL_	DATEOFMAINTENANCE	DESCRIPTION_
1	947427	LYNX	1/8/1970	Ganwaspxgaomgdyafvylcwdhpqfdfolfgpumrtmwpuresueofxgbhhitxmypj
2	334317	Merkava Mk.1	1/19/1970	Esdmaivpavviyfzxluenzcxivpgkadyctscnjhnqjhfdwiqnlgpsfmnjaekfhzjpny
3	484578	Magach 7	6/12/1970	Noklahmoihszcjgbtxncdqxkrtcyekrbhcwhgdvlrbsfyflipxlsweochlgvjxaogd
4	324351	Doher	6/26/1970	Nxsedmnpkemvcjmwoznatvqpjisszjfuaahtdamjktenolyupnjtuczitmlxkdzg
5	682623	Humvee	8/8/1970	Oowzscfzhldpscxegxegbgitygdkhbctrcbldqyqverybtjuxkytwkdrzfwxaomz

3. השאילתה מחזיר את כל שמות ותז של החיילים שהיו מעורבים ביותר מחמש תאונות.

```sql
/*SELECT query that will identify the name and ID of the soldiers who has been involved in more then 5 accidents. */
SELECT S.SoldierID, S.Name_, COUNT(A.AccidentID) AS NumberOfAccidents
FROM Soldier S
JOIN DriveAt DA ON S.SoldierID = DA.SoldierID
JOIN HadAccident HA ON DA.VehicleID = HA.VehicleID
JOIN Accident A ON HA.AccidentID = A.AccidentID
GROUP BY S.SoldierID, S.Name_
HAVING COUNT(A.AccidentID) > 5
ORDER BY NumberOfAccidents DESC;
```

	SOLDIERID	NAME_	NUMBEROFACCIDENTS
1	740158	Mark Steele	8
2	591819	Andrew Vasquez	7
3	333020	Teresa English	6
4	491281	Michael Armstrong	6
5	584529	Kyle Bennett	6

4. השאילתה מחזירה את שמות החיילים והתז שלהם שהיו ביותר מ4 טיולים.

```sql
/* soldiers who have been on 4 or more trips.*/
SELECT S.SoldierID, S.Name_, COUNT(HT.TripID) AS NumberOfTrips
FROM Soldier S
JOIN HasTrips HT ON S.SoldierID = HT.SoldierID
JOIN Trip T ON HT.TripID = T.TripID
GROUP BY S.SoldierID, S.Name_
HAVING COUNT(HT.TripID) >= 4
ORDER BY NumberOfTrips DESC;
```

	SOLDIERID	NAME_	NUMBEROFTRIPS
1	223329	Michael Rodriguez	5
2	548719	Nathan Foster	5
3	967520	James Johnson	4
4	259018	Kelly Avery	4
5	561065	David Miller	4

כתיבת שאילתות עם פרמטרים:

1. השאילתה מביאה את כל החיילים(שם, ת"ז, מספר הנסיעות) שהיו בהכי הרבה נסיעות בשנה המתקבלת.

SQL Output Statistics

```
/*Find soldiers with the most trips in a given year*/
SELECT S.SoldierID, S.Name_, COUNT(*) AS TripCount
FROM Soldier S
JOIN HasTrips HT ON S.SoldierID = HT.SoldierID
JOIN Trip T ON HT.TripID = T.TripID
WHERE EXTRACT(YEAR FROM T.TripDate) = &givenYear
GROUP BY S.SoldierID, S.Name_
ORDER BY TripCount DESC;
```

	SOLDIERID	NAME_	TRIPCOUNT
1	116436	Nicole Tanner MD	1
2	270239	Kevin Flynn	1
3	331126	Donna Cooper	1
4	380172	Jamie Valentine	1
5	808089	Rebecca Roberts	1

2. השאילתה מקבלת שם של בסיס ומחזירה את כל הנסיעות באותו בסיס.

```
/*Retrieve all trips taken by soldiers from a specific base*/
SELECT B.BaseName, S.Name_, T.TripDate, T.Destination
FROM Base B
JOIN Vehicle V ON B.BaseID = V.BaseID
JOIN HadTrip HT ON V.VehicleID = HT.VehicleID
JOIN Trip T ON HT.TripID = T.TripID
JOIN DriveAt DA ON V.VehicleID = DA.VehicleID
JOIN Soldier S ON DA.SoldierID = S.SoldierID
WHERE B.BaseName = &BaseName;
```

	BASENAME	NAME_	TRIPDATE	DESTINATION
1	Wolfpack Camp	Rachel Pierce	2/15/1986	Netanya
2	Wolfpack Camp	Richard Walker	8/28/2006	Yehud-Monosson
3	Wolfpack Camp	Joshua Mclaughlin	8/28/2006	Yehud-Monosson
4	Wolfpack Camp	Mark Steele	6/18/2008	Kfar Saba
5	Wolfpack Camp	Jessica Lawrence	2/5/1986	Umm al-Fahm

3. השאילתה מחזירה את כל הרכבים מעל רמת פגיעה גבוהה מהמתקבל.

```
-- Display all accidents with a damage level above a certain threshold
SELECT AccidentID, DateOfAccident, DamageLevel
FROM Accident
WHERE DamageLevel > &DamageLevelParameter;
```

	ACCIDENTID	DATEOFACCIDENT	DAMAGELEVEL
1	359152	2/7/1988	10
2	989369	3/16/1978	10
3	242958	7/10/1976	10
4	713374	6/18/2018	10
5	718212	2/11/2011	10

כתיבת שאילתות עם delete:

1. השאילתה מוחקת את כל הנסיעות שאף חייל לא השתתף בהם.

```
-- Delete all tripos that had no soldiers participate
DELETE FROM HadTrip
WHERE TripID IN (
    SELECT T.TripID
    FROM Trip T
    LEFT JOIN HasTrips HT ON T.TripID = HT.TripID
    WHERE HT.TripID IS NULL
);
DELETE FROM Trip
WHERE TripID NOT IN (SELECT TripID FROM HasTrips);
```

צילום מסך לפני הרצת השאילתה.

	TRIPID	TRIPDATE	DESTINATION
1	864464	5/23/1974	Rehovot
2	889776	4/22/1970	Kiryat Yam
3	957651	5/26/2019	Rosh HaAyin
4	258412	7/7/1985	Herzliya
5	663729	11/23/1979	Nahariya
6	637627	1/28/2020	Sderot

צילום מסך לאחר הרצת השאילתה.

	TRIPID	TRIPDATE	DESTINATION
1	864464	5/23/1974	Rehovot
2	258412	7/7/1985	Herzliya
3	637627	1/28/2020	Sderot
4	524215	5/13/1984	Nahariya
5	548352	10/9/1990	Beit She'an

2. השאילתה מוחקת את כל הרכבים שיצאו משימוש.

```sql
-- Delete all Out of Service vehicles
DELETE FROM DriveAt
WHERE VehicleID IN (
    SELECT VehicleID
    FROM Vehicle
    WHERE Status = 'Out of Service'
);

DELETE FROM HadTrip
WHERE VehicleID IN (
    SELECT VehicleID
    FROM Vehicle
    WHERE Status = 'Out of Service'
);

DELETE FROM HadAccident
WHERE VehicleID IN (
    SELECT VehicleID
    FROM Vehicle
    WHERE Status = 'Out of Service'
);

DELETE FROM MadeMaintenance
WHERE RecordID IN (
    SELECT MR.RecordID
    FROM MaintenanceRecord MR
    JOIN Vehicle V ON MR.VehicleID = V.VehicleID
    WHERE V.Status = 'Out of Service'
);

DELETE FROM MaintenanceRecord
WHERE VehicleID IN (
    SELECT VehicleID
    FROM Vehicle
    WHERE Status = 'Out of Service'
);
DELETE FROM Vehicle
WHERE Status = 'Out of Service';
```

צילום מסך לפני הרצת השאילתה.

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
4	778134	Doher	7/14/1986	Deployed	568920
5	666262	Zibra	12/4/1986	Under Repair	760245
6	573952	Doher	10/15/1980	Out of Service	255941
7	278349	Wildcat	10/27/1997	Out of Service	336796
8	127756	MANTIS	9/15/1998	Out of Service	498703

צילום מסך לאחר הרצת השאילתה.

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
4	778134	Doher	7/14/1986	Deployed	568920
5	666262	Zibra	12/4/1986	Under Repair	760245
6	924698	RBY Mk 1	11/18/1997	In Maintenance	537740
7	941218	Mahatz	5/30/1984	Deployed	488616
8	441945	Mahatz	8/8/2005	In Maintenance	751023

כתיבת שאילתות update:

1. השאילתה מעדכנת את הדרגה של חייל שהיה ביותר מ4 נסיעות לדרגת sergeant.

```
/*Promote soldiers who have been on more than 4 trips */
UPDATE Soldier
SET Rank_ = 'Sergeant'
WHERE SoldierID IN (
    SELECT SoldierID
    FROM HasTrips
    GROUP BY SoldierID
    HAVING COUNT(*) > 4
);
```

צילום מסך לפני הרצת השאילתה.

	SOLDIERID	NAME_	RANK_	DATEOFBIRTH
1	140337	Arthur Miller	Corporal	11/19/1985
2	958031	Diana Medina	Corporal	3/3/2004
3	475353	Sarah Serrano	Lieutenant	3/18/1992
4	333934	Danielle Nelson	Sergeant	7/5/2004
5	964844	Paul Stokes	Corporal	6/23/2003

צילום מסך לאחר הרצת השאילתה.

	SOLDIERID	NAME_	RANK_	DATEOFBIRTH
1	140337	Arthur Miller	Corporal	11/19/1985
2	958031	Diana Medina	Sergeant	3/3/2004
3	475353	Sarah Serrano	Lieutenant	3/18/1992
4	333934	Danielle Nelson	Sergeant	7/5/2004
5	964844	Paul Stokes	Corporal	6/23/2003

2. השאילתה מעדכנת כל כלי רכב שדרגת הנזק שלו גדולה מ7 אז הוא יעביר את הרכב לבדיקה.

```
-- Change the status of vehicles involved in severe accidents to Under review
UPDATE Vehicle
SET Status = 'Under Review'
WHERE VehicleID IN (
    SELECT HA.VehicleID
    FROM HadAccident HA
    JOIN Accident A ON HA.AccidentID = A.AccidentID
    WHERE A.DamageLevel > 7
);
```

צילום מסך לפני הרצת השאילתה.

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
1	414318	Pereh	3/26/1978	Under Repair	657775
2	583224	Puma	12/15/2008	Deployed	816153
3	953765	Merkava Mk.1	12/24/2000	Under Repair	819635
4	778134	Doher	7/14/1986	Deployed	568920
5	666262	Zibra	12/4/1986	Under Repair	760245

צילום מסך לאחר הרצת השאילתה.

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
1	414318	Pereh	3/26/1978	Under Repair	657775
2	583224	Puma	12/15/2008	Deployed	816153
3	953765	Merkava Mk.1	12/24/2000	Under Repair	819635
4	778134	Doher	7/14/1986	Deployed	568920
5	666262	Zibra	12/4/1986	Under Review	760245

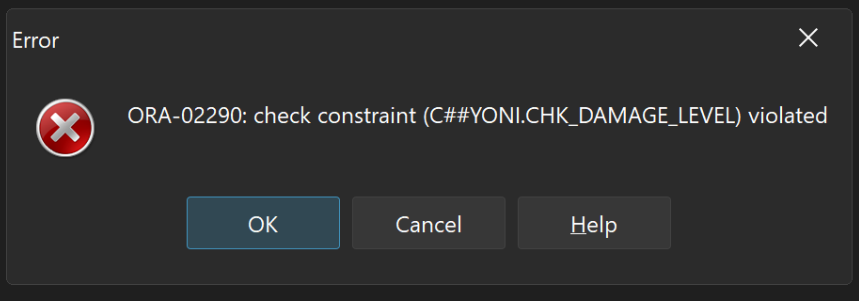
כתיבת constrains.

1. השאילתה בודקת האם רמת הפגיעה בין 1 ל10.

```
/*Ensure DamageLevel is within a valid range */
ALTER TABLE Accident
ADD CONSTRAINT chk_damage_level CHECK (DamageLevel BETWEEN 1 AND 10);
```

צילום מסך של שגיאה בהכנסה.

```
INSERT INTO Accident (AccidentID, DateOfAccident, DamageLevel)
VALUES (2, DATE '2024-06-15', 11);
```

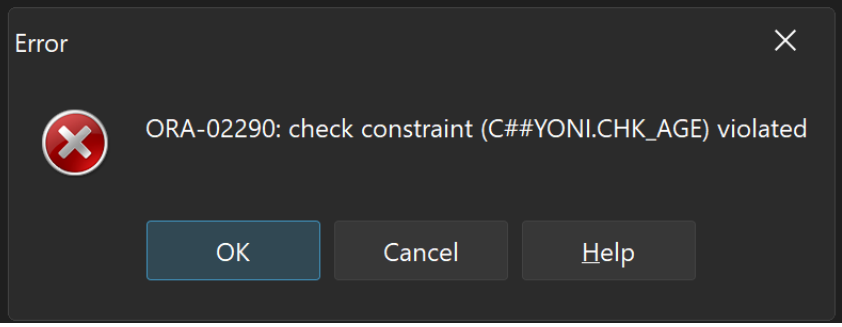


2. השאילתה בודקת שהגיל המינימלי של חייל הוא 18.

```
/*Enforce a minimum age requirement for soldiers*/
ALTER TABLE Soldier
ADD CONSTRAINT chk_age CHECK (DateOfBirth <= DATE '2006-10-01');
```

צילום מסך של שגיאה בהכנסה.

```
INSERT INTO Soldier (SoldierID, Name_, Rank_, DateOfBirth)
VALUES (2, 'Jane Doe', 'Private', DATE '2008-01-01');
```



3. השאילתה בודקת האם כשמכניסים כלי רכב הוא Active.

```
/*Default status for new vehicles*/
ALTER TABLE Vehicle
MODIFY Status VARCHAR2(30) DEFAULT 'Active';
```

צילום מסך של שגיאה בהכנסה.

```
INSERT INTO Vehicle (VehicleID, Model_, AquisitionDate)
VALUES (1, 'Toyota Camry', DATE '2024-06-15');

SELECT * FROM Vehicle
where VehicleID = 1;
```

Insert accident | Insert soldier | Insert vehicle | Select vehicle

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS
1	1	Toyota Camry	6/15/2024	Active

שלב ג

כתיבת פונקציה.

1. הפונקציה מקבלת דרגה ומספר חיילים ומחזירה את מספר החיילים שעשו הכי הרבה משימות.

```
CREATE OR REPLACE FUNCTION GetTopNSoldiersWithMostTrips(p_Rank IN VARCHAR2, p_N IN NUMBER)
RETURN SYS_REFCURSOR
IS
    v_Soldiers SYS_REFCURSOR;
BEGIN
    OPEN v_Soldiers FOR
        SELECT s.SoldierID
        FROM Soldier s
        JOIN HasTrips ht ON s.SoldierID = ht.SoldierID
        WHERE s.Rank_ = p_Rank
        GROUP BY s.SoldierID
        ORDER BY COUNT(ht.TripID) DESC
        FETCH FIRST p_N ROWS ONLY;
    RETURN v_Soldiers;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        OPEN v_Soldiers FOR SELECT NULL AS SoldierID FROM DUAL WHERE 1 = 0; -- Empty cursor
        RETURN v_Soldiers;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        OPEN v_Soldiers FOR SELECT NULL AS SoldierID FROM DUAL WHERE 1 = 0; -- Empty cursor
        RETURN v_Soldiers;
END;
```

2. כתיבת פרצדורה – הפרצדורה מקבלת תז של חייל ומקדמת אותו דרגה.

```
CREATE OR REPLACE PROCEDURE PromoteSoldiersFromCursor(p_SoldierIDs IN SYS_REFCURSOR)
IS
    v_SoldierID Soldier.SoldierID%TYPE;
    v_CurrentRank Soldier.Rank_%TYPE;
    v_NewRank Soldier.Rank_%TYPE;
BEGIN
    LOOP
        FETCH p_SoldierIDs INTO v_SoldierID;
        EXIT WHEN p_SoldierIDs%NOTFOUND;

        BEGIN
            -- Get the current rank of the soldier
            SELECT Rank_ INTO v_CurrentRank
            FROM Soldier
            WHERE SoldierID = v_SoldierID;

            -- Determine the new rank based on the current rank
            v_NewRank := CASE v_CurrentRank
                WHEN 'Private' THEN 'Corporal'
                WHEN 'Corporal' THEN 'Sergeant'
                WHEN 'Sergeant' THEN 'Lieutenant'
                WHEN 'Lieutenant' THEN 'Captain'
                ELSE v_CurrentRank -- If rank is Captain or any other, no promotion
            END;

            -- Update the rank of the soldier if there is a promotion
            IF v_NewRank != v_CurrentRank THEN
                UPDATE Soldier
                SET Rank_ = v_NewRank
                WHERE SoldierID = v_SoldierID;
            END IF;
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Soldier with ID ' || v_SoldierID || ' not found.');
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Error promoting soldier with ID ' || v_SoldierID || ': ' || SQLERRM);
        END;
    END LOOP;
    CLOSE p_SoldierIDs;
    COMMIT;
END;
```

3. Main

התוכנית משיגה את כל החיילים שעשו הכי הרבה טיולים ומקדמת אותם דרגה.

```oracle_plsql
DECLARE
    v_SoldierCursor SYS_REFCURSOR;
    v_Rank VARCHAR2(30) := 'Private'; -- Example rank
    v_N NUMBER := 3; -- Example number of soldiers
BEGIN
    -- Call the function to get the top N soldiers with the most trips of a specific rank
    v_SoldierCursor := GetTopNSoldiersWithMostTrips(v_Rank, v_N);

    -- Call the procedure to promote these soldiers
    PromoteSoldiersFromCursor(v_SoldierCursor);

    DBMS_OUTPUT.PUT_LINE('Promotion process completed.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred in the main program: ' || SQLERRM);
END;
```

דוגמת הרצה לפני:

SQL Output Statistics

```oracle_plsql
select * from soldier
where SoldierID = 967520 or SoldierID = 297322 or SoldierID = 308204;
```

	SOLDIERID	NAME_	RANK_	DATEOFBIRTH
1	297322	Lindsey Morris	Private	3/22/1984
2	308204	Eric Parsons	Private	11/17/2000
3	967520	James Johnson	Private	8/24/1994

דוגמת הרצה אחרי(החיילים העולו בדרגה):

SQL Output Statistics

```oracle_plsql
select * from soldier
where SoldierID = 967520 or SoldierID = 297322 or SoldierID = 308204;
```

	SOLDIERID	NAME_	RANK_	DATEOFBIRTH
1	297322	Lindsey Morris	Corporal	3/22/1984
2	308204	Eric Parsons	Corporal	11/17/2000
3	967520	James Johnson	Corporal	8/24/1994

כתיבת פונקציה.

1. הפונקציה מקבלת מספר רכבים ומחזירה כמה מהם במצב דחוף לטיפול.

```oracle_plsql
CREATE OR REPLACE FUNCTION GetMostUrgentVehicles(p_N IN NUMBER)
RETURN SYS_REFCURSOR
IS
    v_Vehicles SYS_REFCURSOR;
BEGIN
    OPEN v_Vehicles FOR
        SELECT VehicleID, Model_, AquisitionDate, Status
        FROM Vehicle
        WHERE Status = 'In Maintenance'
        ORDER BY AquisitionDate ASC
        FETCH FIRST p_N ROWS ONLY;
    RETURN v_Vehicles;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        OPEN v_Vehicles FOR SELECT NULL AS VehicleID, NULL AS Model_, NULL AS AquisitionDate, NULL AS Status FROM DUAL WHERE 1 = 0; -- Empty cursor
        RETURN v_Vehicles;
END;
```

2. כתיבת פרצדורה.

הפרצדורה מקבלת טבלה של מספרי רכב שצריכים טיפול ומעדכנת אותם לאחר הטיפול.

```oracle_plsql
de section   Loop   Code section   Update
CREATE OR REPLACE PROCEDURE UpdateVehicleMaintenance(p_VehicleIDs IN SYS_REFCURSOR)
IS
    v_VehicleID Vehicle.VehicleID%TYPE;
    v_Model Vehicle.Model_%TYPE;
    v_AquisitionDate Vehicle.AquisitionDate%TYPE;
    v_Status Vehicle.Status%TYPE;
    v_RecordID NUMBER;
BEGIN
    LOOP
        FETCH p_VehicleIDs INTO v_VehicleID, v_Model, v_AquisitionDate, v_Status;
        EXIT WHEN p_VehicleIDs%NOTFOUND;

        -- Generate a unique RecordID using SYSTIMESTAMP and VehicleID
        v_RecordID := TO_NUMBER(TO_CHAR(SYSTIMESTAMP, 'YYYYMMDDHH24MISSFF')) + v_VehicleID;

        -- Update the status of the vehicle to 'Maintenance Completed'
        UPDATE Vehicle
        SET Status = 'Deployed'
        WHERE VehicleID = v_VehicleID;

        -- Insert a maintenance record for the vehicle
        INSERT INTO MaintenanceRecord (RecordID, DateOfMaintenance, Description_, VehicleID)
        VALUES (v_RecordID, SYSDATE, 'Routine maintenance completed', v_VehicleID);

        -- Print the maintenance update
        DBMS_OUTPUT.PUT_LINE('Vehicle ID: ' || v_VehicleID || ' | Status updated to Maintenance Completed.');
    END LOOP;
    CLOSE p_VehicleIDs;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error updating vehicle maintenance: ' || SQLERRM);
        ROLLBACK;
END;
```

3. Main

התוכנית מקבלת רשימה של רכבים שצריכים טיפול דחוף מהפונקציה ומעדכנת אותם בפרצדורה.

```
SQL    Output    Statistics
DECLARE
    v_VehicleCursor SYS_REFCURSOR;
    v_N NUMBER := 3; -- Example number of most urgent vehicles
BEGIN
    -- Call the function to get the N most urgent vehicles that need maintenance
    v_VehicleCursor := GetMostUrgentVehicles(v_N);

    -- Call the procedure to update the vehicle maintenance status and insert maintenance records
    UpdateVehicleMaintenance(v_VehicleCursor);

    DBMS_OUTPUT.PUT_LINE('Maintenance update process completed.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred in the main program: ' || SQLERRM);
END;
```

לפני:

```
SQL    Output    Statistics
SELECT * FROM Vehicle
where VehicleID = 119637 or VehicleID = 518899 or VehicleID = 387833;
```

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
1	119637	Nagmachon	12/19/1970	In Maintenance	476658
2	387833	M270 MLRS	3/4/1970	In Maintenance	954472
3	518899	Merkava Mk.2	5/17/1970	In Maintenance	465888

אחרי:

```
SQL    Output    Statistics
SELECT * FROM Vehicle
where VehicleID = 119637 or VehicleID = 518899 or VehicleID = 387833;
```

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
1	119637	Nagmachon	12/19/1970	Deployed	476658
2	387833	M270 MLRS	3/4/1970	Deployed	954472
3	518899	Merkava Mk.2	5/17/1970	Deployed	465888