

---

# 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 עבר

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

בנוסף הבסיס נתונים ינהל את כל החיילים (מכונאי, נהג, 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	<u>VehicleID</u>	Primary Key (מפתח ראשי) של הטבלה, מזהה ייחודי לכל כלי רכב.
	Model	הדגם
	AcquisitionDate	תאריך רכישת הרכב

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

DriveAt	VehcileID	מפתח זר לרכבים
	SoldierID	מפתח זר לחיילי ם

## הסבר על נרמול NF3

נפרט על כל אחת מהטבלאות:

### Vehicle

VehicleID הוא המפתח הראשי.

Model, AcquisitionDate, Status, ו-BaselD תלויות ישירות במפתח הראשי VehicleID. BaselD הוא מפתח זר שמצביע על BaselD 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

BaselD הוא המפתח הראשי.

BaseName ו-Location תלויות ישירות במפתח הראשי BaselD.

הוא . MaintenanceRecord (רשומת תחזוקה) :

RecordID המפתח הראשי

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

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

### Soldier

SoldierID הוא המפתח הראשי.

Name, Rank, 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.

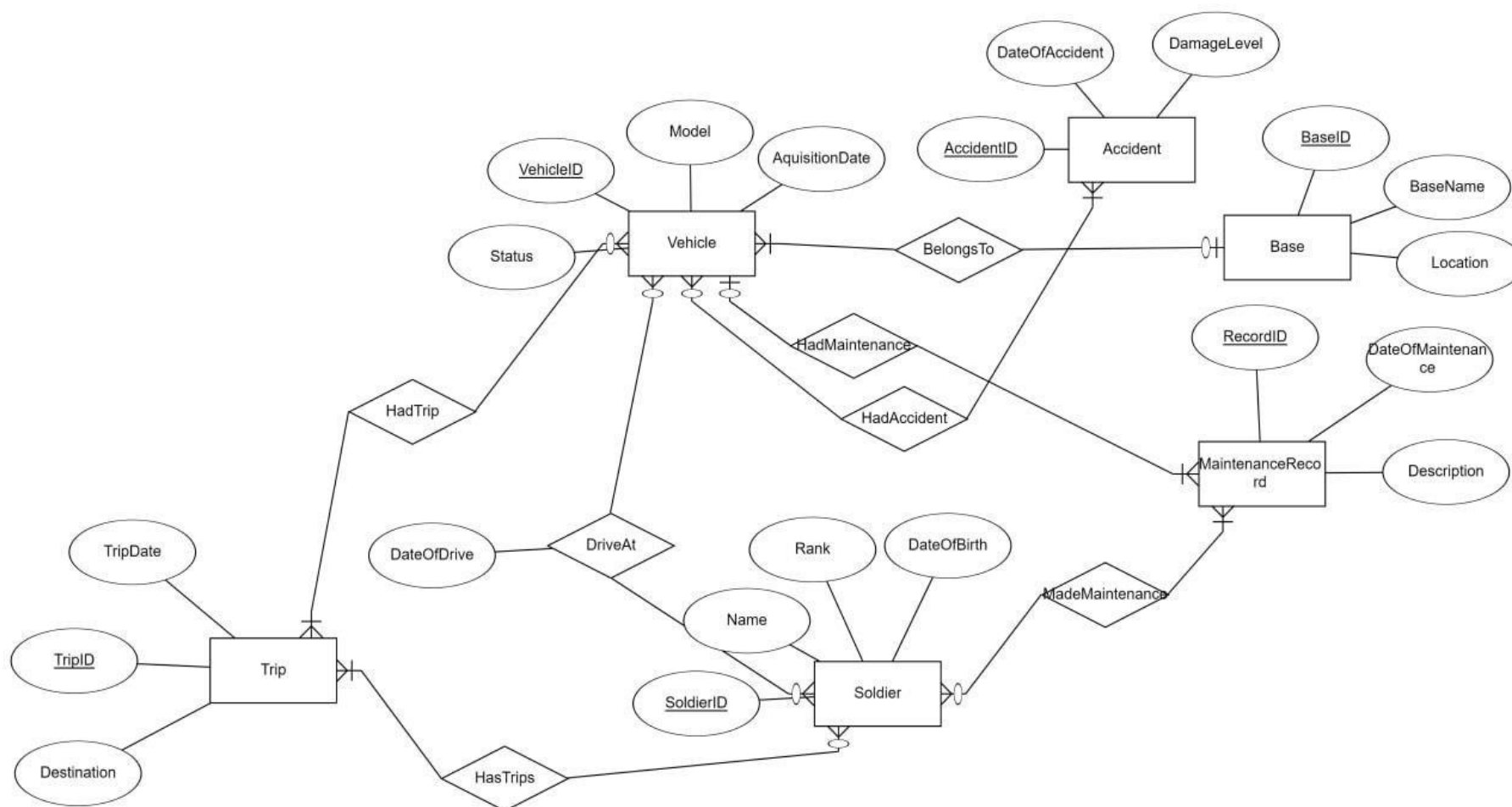
בהתאמה.

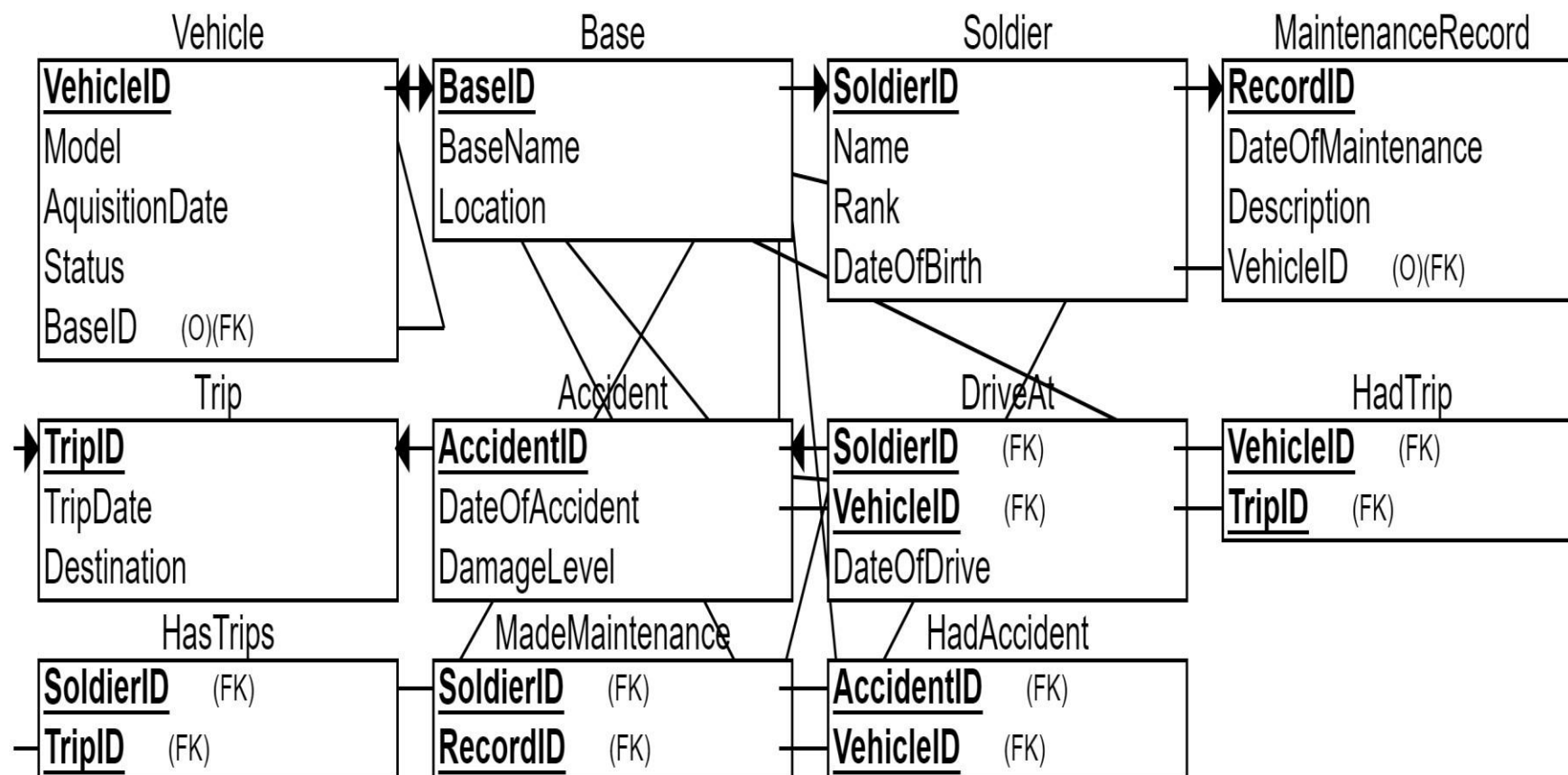
אין תלות פונקציונלית חלקית או טרנזיטיבית בין המפתחות הללו.

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

פונקציונליות בלתי רצויות.

# דיאגרמת ERD







## יצירת ואכלוס הטבלאות

במסמך הדרישות לפרויקט במודל התבקשנו לכתוב הסבר למפוי ה ERD לטבלאו 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

Field Name	Type	Options
BaseID	Digit Sequence	##### blank: 0 % Σ ×
BaseName	Custom List	Iron Fortress, Eagle's Nest, Thunderbolt Barracks, Lion's Den, Shadow Ridge Base, Phoenix Stronghold, random blank: 0 % Σ ×
Location_	Custom List	Tel Aviv, Jerusalem, Haifa, Eilat, Dead Sea, Galilee, Negev Desert, Caesarea, Jaffa, Tiberias, Nazareth, A random blank: 0 % Σ ×

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

# Rows: 500 Format: SQL Table Name: Base ☐ include CREATE TABLE

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

```
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



## 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

```
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()
```

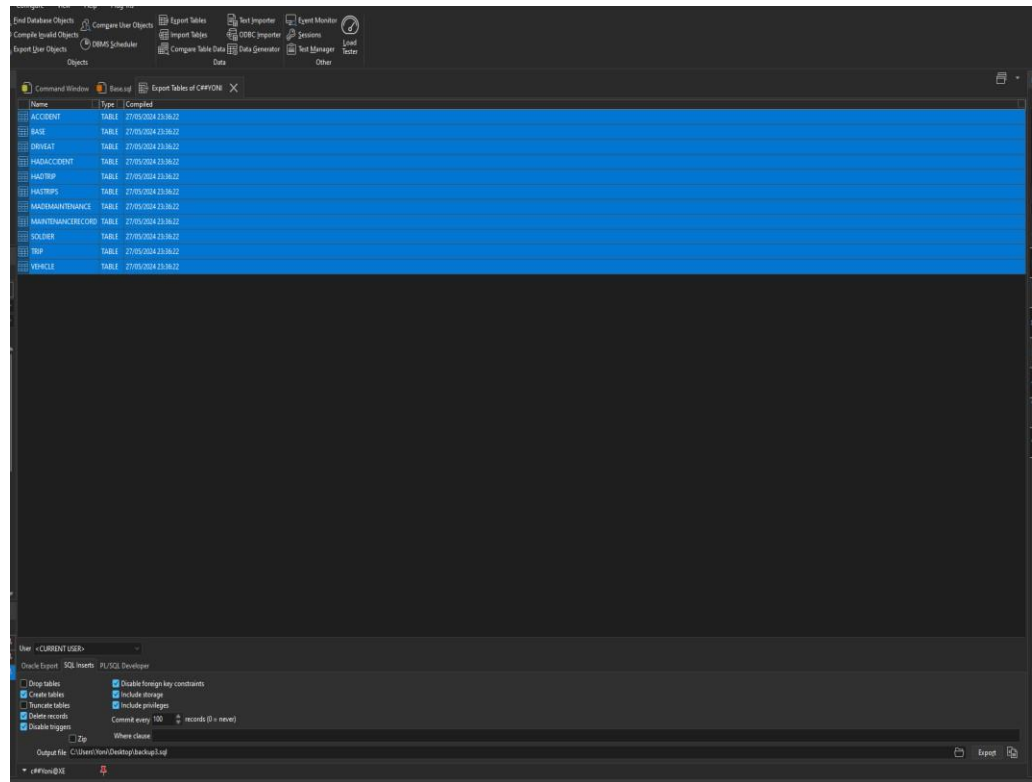
## 3. הצגת הנתונים ע"י פקודת select \*



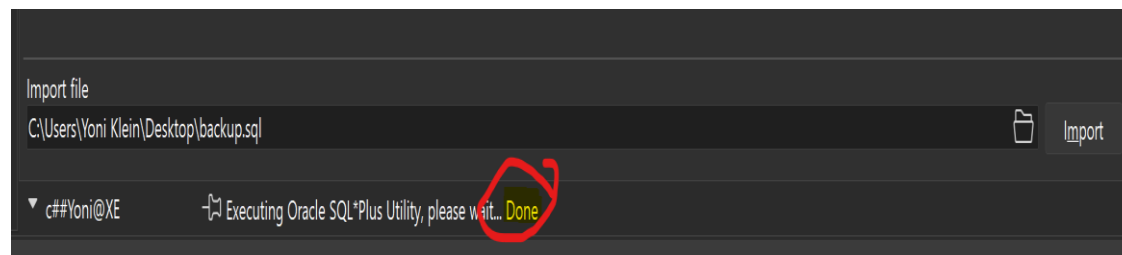
		SOLDIERID	NAME_		RANK_		DATEOFBIRTH	
▶	1	843643	Alexis Hopkins	...	Private	...	06/03/1992	...
	2	234622	Richard Riley	...	Lieutenant	...	04/03/2004	...
	3	372213	Travis Smith	...	Corporal	...	14/02/1988	...
	4	423436	Kenneth Mendoza	...	Lieutenant	...	14/11/2000	...
	5	943303	Luis Bean	...	Sergeant	...	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	Ganwaspxgaomgdyafvylcwdhqpqdfolfgpumrtmwpuresueofxgbhhitxmypj
2	334317	Merkava Mk.1	1/19/1970	Esdmaivpavviyfzxluezxcxivpgkadyctscnjhqnjhfwdwiqnlgpsfmnjjaekfhzjpnyc
3	484578	Magach 7	6/12/1970	Noklahmoihszcjgbtxncdqxkrctykehrbchwgdvlrbsfyflipxlsweochlgvjxaogdk
4	324351	Dohar	6/26/1970	Nxsedmnpkemvcjmwoznatvqpjisszjfuaahdamjktenolyupnjtuczitmlxkdzg
5	682623	Humvee	8/8/1970	Oowzscfzhldpscxegxegbgitygdkhbctrbldqyqverybtjuxkytwkdrzfwxaomz

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

```
/*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.

```

/* 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
<pre> /*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) = &amp;givenYear GROUP BY S.SoldierID, S.Name_ ORDER BY TripCount DESC; </pre>		

		SOLDIERID	NAME_	TRIPCOUN
▶	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 שאילות עם delete:

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

```
-- Delete all tripes 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.

```
-- 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'
```



צילום מסך לפני הרצת השאילתה.

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
4	778134	Dohar	7/14/1986	Deployed	568920
5	666262	Zibra	12/4/1986	Under Repair	760245
▶ 6	573952	Dohar	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	Dohar	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 מ-4 נסיעות לדרגת sergeant.

```
SQL Output Statistics
/*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	Dohar	***	7/14/1986	***	Deployed	***	568920
	5	666262	Zibra	***	12/4/1986	***	Under Repair	***	760245

צילום מסך לאחר הרצת השאילתה.

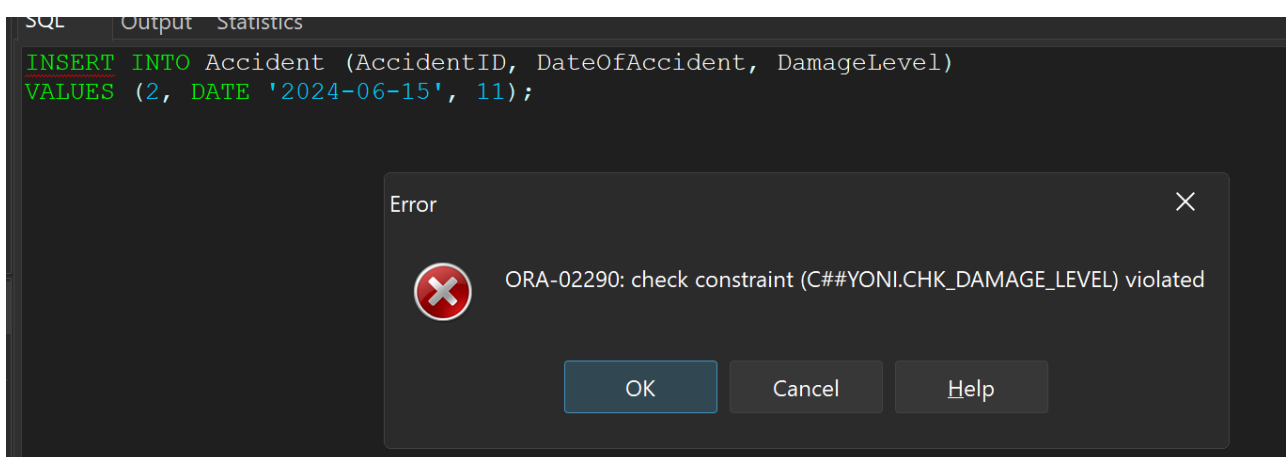
		VEHICLEID	MODEL_		AQUISITIONDATE		STATUS		BASEID
▶	1	414318	Pereh	***	3/26/1978	***	Under Repair	***	657775
	2	583224	Puma	***	12/15/2008	***	Deployed	***	816153
	3	953765	Merkava Mk.1	***	12/24/2000	***	Under Repair	***	819635
	4	778134	Dohar	***	7/14/1986	***	Deployed	***	568920
	5	666262	Zibra	***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

כתיבת constraints.

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.

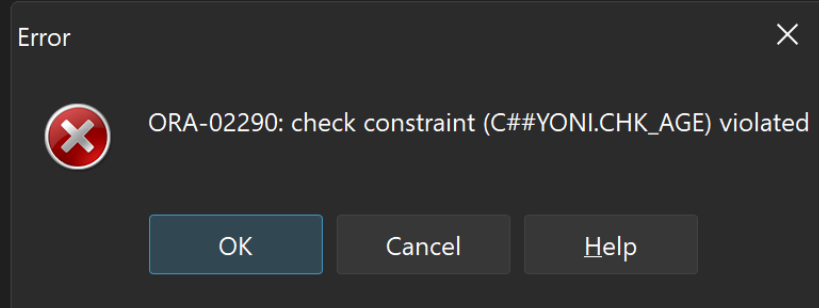


2. השאילתה בודקת שהגיל המינימלי של חייל הוא 18.

```
/*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	B
▶ 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.

```

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

```
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

```
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.

```
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.

```

de section  Loop  Code section  Update
1 CREATE OR REPLACE PROCEDURE UpdateVehicleMaintenance(p_VehicleIDs IN SYS_REFCURSOR)
2 IS
3     v_VehicleID Vehicle.VehicleID%TYPE;
4     v_Model Vehicle.Model_%TYPE;
5     v_AquisitionDate Vehicle.AquisitionDate%TYPE;
6     v_Status Vehicle.Status%TYPE;
7     v_RecordID NUMBER;
8 BEGIN
9     LOOP
10        FETCH p_VehicleIDs INTO v_VehicleID, v_Model, v_AquisitionDate, v_Status;
11        EXIT WHEN p_VehicleIDs%NOTFOUND;
12
13        -- Generate a unique RecordID using SYSTIMESTAMP and VehicleID
14        v_RecordID := TO_NUMBER(TO_CHAR(SYSTIMESTAMP, 'YYYYMMDDHH24MISSFF')) + v_VehicleID;
15
16        -- Update the status of the vehicle to 'Maintenance Completed'
17        UPDATE Vehicle
18        SET Status = 'Deployed'
19        WHERE VehicleID = v_VehicleID;
20
21        -- Insert a maintenance record for the vehicle
22        INSERT INTO MaintenanceRecord (RecordID, DateOfMaintenance, Description_, VehicleID)
23        VALUES (v_RecordID, SYSDATE, 'Routine maintenance completed', v_VehicleID);
24
25        -- Print the maintenance update
26        DBMS_OUTPUT.PUT_LINE('Vehicle ID: ' || v_VehicleID || ' | Status updated to Maintenance Completed.');

```

### 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

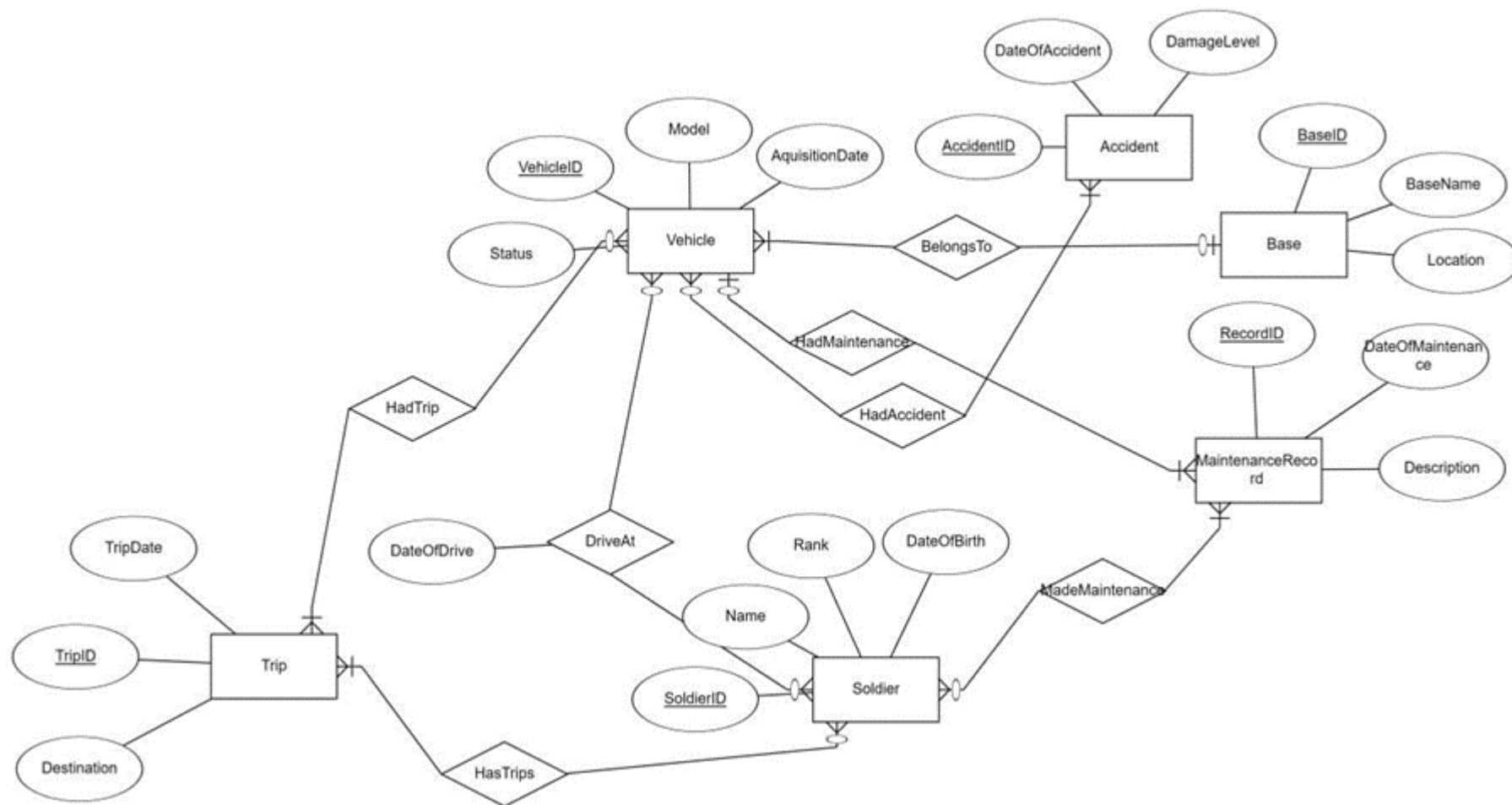
Statistics

SELECT \* FROM Vehicle  
where VehicleID = 119637 or VehicleID = 518899 or VehicleID = 387833;

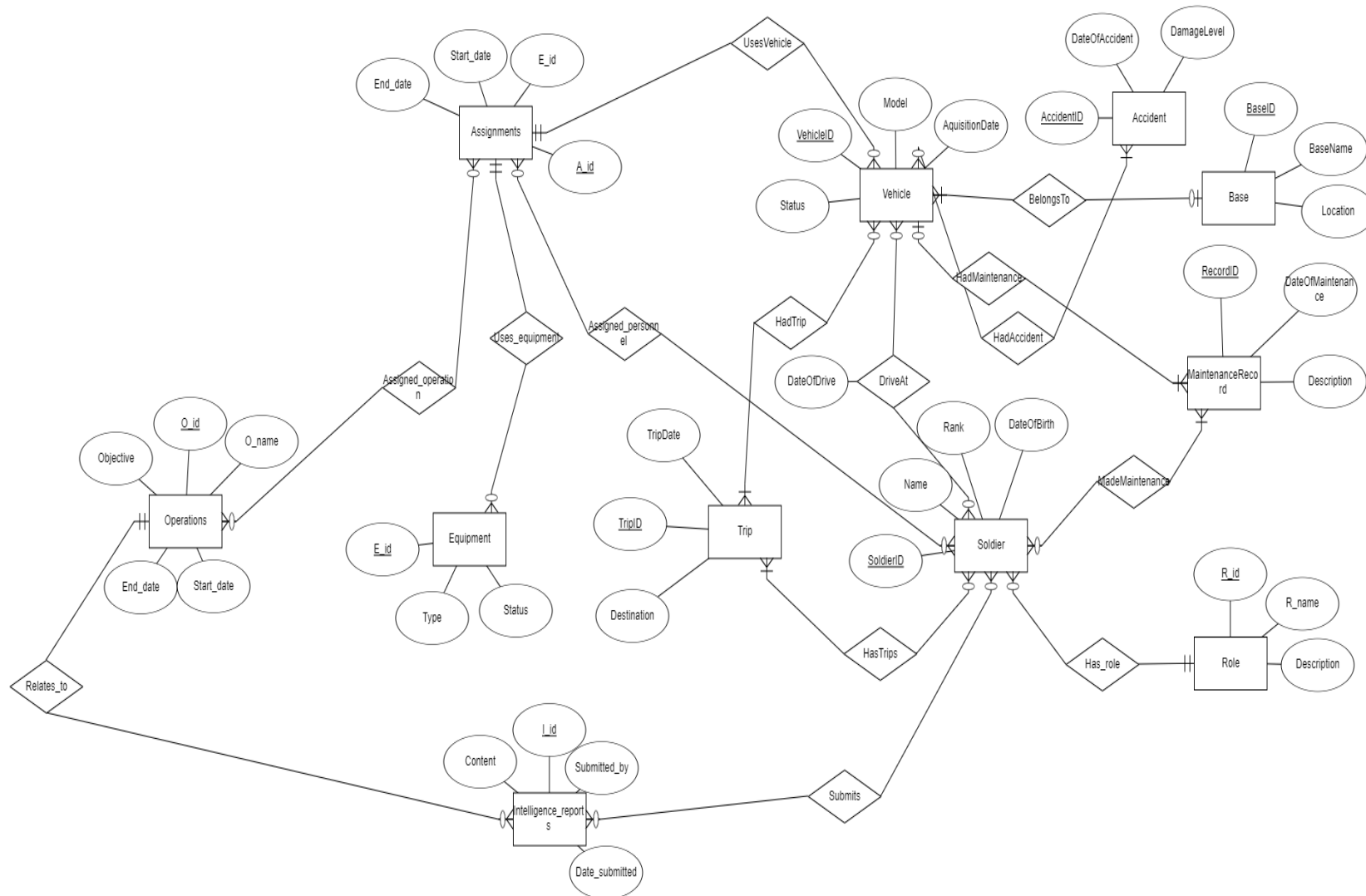
	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	BASEID
1	119637	Nagmachon	12/19/1970	Deployed	476658
2	387833	M270 MLRS	3/4/1970	Deployed	954472
3	518899	Merkava Mk.2	5/17/1970	Deployed	465888

שלב ד

1. דיאגרמת ERD מקורית



## 2. דיאגרמת ERD משותף



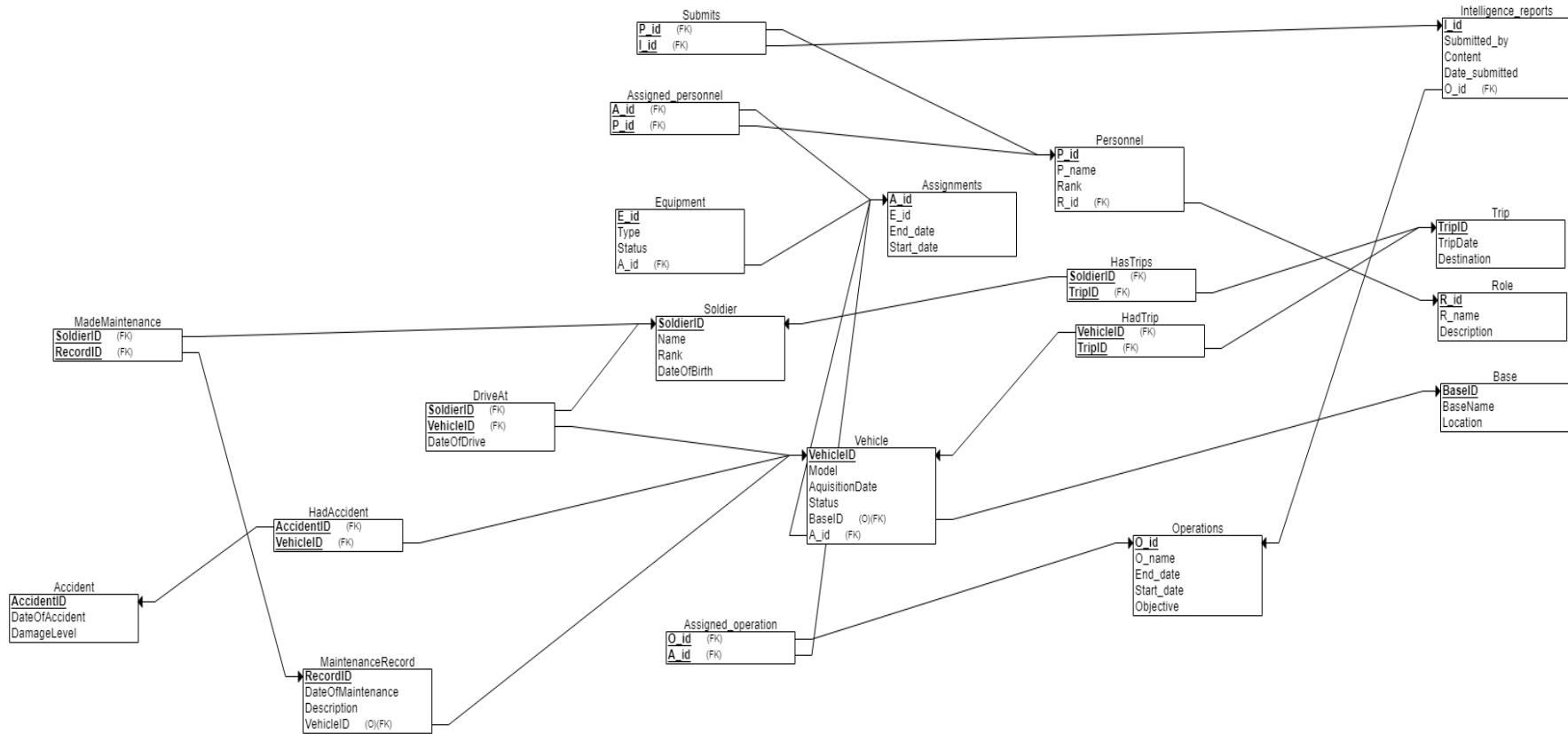


החלטות:

החלטנו לאחד בין טבלת חיילים (Soldier) בטבלה המקורית שלנו לטבלת האנשי צוות (Personnel) מכיוון שהם אותו דבר.

החלטנו לעשות קשר בין טבלת רכבים (Vehicles) לבין טבלת משימות (Assignment) כי לכל משימה צריך רכב.

### 3. תרשים DSD משותף



## 4. אינטגרציה

מצורף הקוד של יצירת הטבלאות החדשות שקיבלנו על ידי האינטגרציה (חלקי)

```
CREATE TABLE Operations_
(
  O_id INT NOT NULL,
  O_name VARCHAR2(50) NOT NULL,
  Start_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,
  E_id INT NOT NULL,
  Start_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(50) NOT NULL,
  Date_submitted DATE NOT NULL,
  Content_ VARCHAR2(1000) NOT NULL,
  PRIMARY KEY (I_id),
  FOREIGN KEY (O_id) REFERENCES Operations_(O_id)
);

CREATE TABLE Role_
(
  R_id INT NOT NULL,
  R_name VARCHAR2(50) NOT NULL,
  Description_ VARCHAR2(200) NOT NULL,
  PRIMARY KEY (R_id)
);

CREATE TABLE Equipment
(
  E_id INT NOT NULL,
  Type_ VARCHAR2(50) NOT NULL,
  Status VARCHAR2(20) NOT NULL,
  PRIMARY KEY (E_id)
);

CREATE TABLE Submits
(
  P_id INT NOT NULL,
  I_id INT NOT NULL,
  PRIMARY KEY (P_id, I_id),
  FOREIGN KEY (P_id) REFERENCES Soldier(SoldierID),
  FOREIGN KEY (I_id) REFERENCES Intelligence_reports(I_id)
);
```

```
-- add the column
ALTER TABLE Vehicle
ADD A_id INT;

-- make it foreign key
ALTER TABLE Vehicle
ADD CONSTRAINT fk_Assignments
FOREIGN KEY (A_id)
REFERENCES Assignments(A_id);

-- add the column
ALTER TABLE Soldier
ADD R_id INT;

-- make it foreign key
ALTER TABLE Soldier
ADD CONSTRAINT fk_Role
FOREIGN KEY (R_id)
REFERENCES Role_(R_id);
```

מצורף הקוד של המשך האינטגרציה. הוספת הקשר  
החדש וחיבור טבלת Soldier עם Role מכיוון  
שאיחדנו אותם עם Personnel

## 5. מבטים

הצגת המשימות והמבצעים והציוד שקשורים אליהם.

```
CREATE VIEW Assignments_View AS
SELECT
    a.A_id,
    a.E_id,
    a.Start_date,
    a.End_date,
    e.E_id AS EquipmentID,
    e.Type_,
    e.Status,
    o.O_id,
    o.O_name,
    o.Objective
FROM
    Assignments a
LEFT JOIN
    Equipment e ON a.E_id = e.E_id
LEFT JOIN
    Assigned_operation ao ON a.A_id = ao.A_id
LEFT JOIN
    Operations_ o ON ao.O_id = o.O_id;
```

	A_ID	E_ID	START_DATE	END_DATE	EQUIPMENTID	TYPE_	STATUS	O_ID	O_NAME	OBJECTIVE
1	1	49445	02/09/1968	30/10/1990	49445	TPC	Taken			
2	3	12256	10/10/1964	03/04/2000	12256	Tax	Taken	25	stable	Configurable web-enabled extranet
3	5	99558	05/10/1964	04/04/2022	99558	Dgc	Taken	275	focus group	Robust intermediate access
4	7	91983	17/07/1976	20/03/2002	91983	JkJ	Taken			
5	9	19841	15/05/1980	01/09/2010	19841	vhO	Taken	232	hierarchy	Re-contextualized optimizing firmware
6	11	87827	13/07/1965	28/09/2008	87827	nNa	Taken	178	flexibility	Future-proofed tangible firmware
7	11	87827	13/07/1965	28/09/2008	87827	nNa	Taken	338	budgetary management	Ergonomic analyzing internet solution
8	13	33004	23/05/1953	05/12/2017	33004	LdR	Taken			
9	15	99558	12/07/1969	22/02/2018	99558	Dgc	Taken	67	upward-trending	Optimized intangible ability
10	15	99558	12/07/1969	22/02/2018	99558	Dgc	Taken	102	asynchronous	Reduced interactive algorithm

דוגמת הרצה:

שאלת select ראשונה על המבט הראשון. מציג כמה משימות היו קשורים לכל ציוד.

```
SELECT
    EquipmentID,
    Type_,
    COUNT(A_id) AS assignment_count
FROM
    Assignments_View
GROUP BY
    EquipmentID,
    Type_;
```

תוצאת השאלתה:

	EQUIPMENTID	TYPE_	ASSIGNMENT_COUNT
1	87827	nNa	3
2	33004	LdR	4
3	63110	tqm	1
4	61604	cBn	1
5	58889	TyC	1
6	92267	dqz	4
7	13112	hMK	1
8	48961	OKk	3
9	51457	MgM	1
10	28805	uiR	2

שאלת select שניה על המבט הראשון.  
הוספה למבט את משך הזמן בימים של כל משימה.

```
SELECT
    A_id,
    E_id,
    Start_date,
    End_date,
    EquipmentID,
    Type_,
    Status,
    O_id,
    O_name,
    Objective,
    (End_date - Start_date) AS duration_days
FROM
    Assignments_View;
```

## תוצאת השאילתה:

	A_ID	E_ID	START_DATE	END_DATE	EQUIPMENTID	TYPE_	STATUS	O_ID	O_NAME	OBJECTIVE	DURATION_DAYS
1	1	49445	02/09/1968	30/10/1990	49445	TPC	Taken				8093
2	3	12256	10/10/1964	03/04/2000	12256	Tax	Taken	25	stable	Configurable web-enabled extranet	12959
3	5	99558	05/10/1964	04/04/2022	99558	Dgc	Taken	275	focus group	Robust intermediate access	21000
4	7	91983	17/07/1976	20/03/2002	91983	JkI	Taken				9377
5	9	19841	15/05/1980	01/09/2010	19841	vhO	Taken	232	hierarchy	Re-contextualized optimizing firmware	11066
6	11	87827	13/07/1965	28/09/2008	87827	nNa	Taken	178	flexibility	Future-proofed tangible firmware	15783
7	11	87827	13/07/1965	28/09/2008	87827	nNa	Taken	338	budgetary management	Ergonomic analyzing internet solution	15783
8	13	33004	23/05/1953	05/12/2017	33004	LdR	Taken				23572
9	15	99558	12/07/1969	22/02/2018	99558	Dgc	Taken	67	upward-trending	Optimized intangible ability	17757
10	15	99558	12/07/1969	22/02/2018	99558	Dgc	Taken	102	asynchronous	Reduced interactive algorithm	17757

```

CREATE VIEW Vehicle_View AS
SELECT
    v.VehicleID,
    v.Model_,
    v.AquisitionDate,
    v.Status,
    m.RecordID,
    m.DateOfMaintenance,
    m.Description_
FROM
    Vehicle v
JOIN
    MaintenanceRecord m ON v.VehicleID = m.VehicleID;

```

מבט שני מציג את הרכבים ודוחות הטיפול שלהם.

דוגמת הרצה:

	VEHICLEID	MODEL_	AQUISITIONDATE	STATUS	RECORDID	DATEOFMAINTENANCE	DESCRIPTION_
1	269282	Merkava Mk.2	25/02/1978	Under Review	256484	25/07/1977	Utmctiwidbjzgcsgwzbxkdkfwnqtiybapppgmvenuwnsohqifiktmmvmv
2	485274	Nakpadon	15/06/1992	Deployed	313727	15/08/2003	Wltwqlgccanmikxwspwjqtuokmxijrsqxugrxevkprxmgbkhhkhdnysgfeyia
3	334896	M270 MLRS	18/02/1981	Under Repair	514327	13/02/2010	Qxxdpeesbnbzdgylhibzixsbctqufjakesxnokpiyvowuqtgrxzxwnosnzm
4	814835	Achzarit	25/08/1977	Under Review	532129	11/08/1982	Fswllzwvtiljywjzuyqfavesrhvwxqxjrcyyujlwactdefljioexowxwbnolr
5	269578	David Light Armored Vehicle	14/08/1978	In Maintenance	411752	27/11/1980	Txnckfrsewqduueffrtadnzagundkkuzwhwfcvcjnoayjprsavafprkrcdjtnich
6	632974	Achzarit	04/03/2001	Under Repair	831538	21/06/1985	Rinuizwavneljrbremotbdkvbhkhhuuehdfdzapntqypbhhyfgyekyewektazimbq
7	325119	Gaash	03/03/1981	Under Review	287517	31/03/1977	Jriihpjmgaailwqxminzolunbbnqywrqbwbvbaavlayaoshvklfaxmxmtmtvu
8	814785	Nimrod	15/06/1985	Under Review	853842	25/07/2013	Yjfixnuamfkyhpcpptsxxxgyccpgfotbafkevqxcwbzjoaenknkhqebxkbjz
9	947427	LYNX	10/12/2012	In Maintenance	965452	08/01/1970	Ganwaspxgaomgdyafvylcwdhppqdfolfgpumrtmwpuresueofxgbhhitxmyj
10	325119	Gaash	03/03/1981	Under Review	195589	15/08/2013	Slhmykfkcxczfcqvnbsuhzkechbbhzvaixgyuyqdsbqvkmppxxfptkiuqkmmk
11	251756	Machbet	30/03/1977	Under Review	203123	08/04/2023	Wkchajclthbzvordhualufkredhwpjznpixcagdzazwksopniagzzwtdai

```
SELECT
    VehicleID,
    Model_,
    COUNT(RecordID) AS maintenance_count
FROM
    Vehicle_View
GROUP BY
    VehicleID,
    Model_;
```

שאלת select ראשונה על המבט השני. סופר לכל רכב את כמות הטיפולים שלו.

תוצאת השאלתה:

	VEHICLEID	MODEL_	MAINTENANCE_COUNT
1	269282	Merkava Mk.2	1
2	814835	Achzarit	2
3	325119	Gaash	3
4	182726	Achzarit	1
5	192413	Merkava Mk.3	1
6	484578	Magach 7	1
7	726562	M113 Zelda	3
8	975531	Sufa (Storm) Jeep	2
9	333691	Humvee	1
10	191535	Machbet	1
11	467767	MANTIS	2

שאילתת select שניה על המבט השני.

מראה את גיל הרכב.

```
SELECT
    VehicleID,
    Model_,
    AquisitionDate,
    ROUND(MONTHS_BETWEEN(SYSDATE, AquisitionDate) / 12, 2) AS vehicle_age_years
FROM
    Vehicle_View;
```

תוצאת השאילתה:

	VEHICLEID	MODEL_	AQUISITIONDATE	VEHICLE_AGE_YEARS
1	269282	Merkava Mk.2	25/02/1978	46.42
2	485274	Nakpadon	15/06/1992	32.11
3	334896	M270 MLRS	18/02/1981	43.43
4	814835	Achzarit	25/08/1977	46.92
5	269578	David Light Armored Vehicle	14/08/1978	45.95
6	632974	Achzarit	04/03/2001	23.39
7	325119	Gaash	03/03/1981	43.39
8	814785	Nimrod	15/06/1985	39.11
9	947427	LYNX	10/12/2012	11.62
10	325119	Gaash	03/03/1981	43.39