

מעיין סעידוב 322903410 maayan18403@gmail.com הדס זאבי 325543353 ho548510062@gmail.com



תוכן עניינים:

וניאור ווארגוןוניאור וואר וואר וואר וואר וואר וואר ווואר ווואר ווואר ווואר ווואר וווווווו	
תרשים Erd	
Dsd תרשים	5
פירוט הישויות	6
קשרים בין הישויות	
createTable	8
insertTable	9
DropTable	10
DescTable	
SelectAll	12
מילוי הנתונים בטבלאות	13-22
גיבוי ושחזור	23-24
שלב 2	25
חדש Erd תרשים	26
חדש Dsd תרשים	27
ללא פרמטרים select שאילתות	
ללא פרמטרים update ללא פרמטרים	
ללא פרמטרים delete שאילתות	38-42
פרמטרים select שאילתות	42-45
אילוצים	46-49

תיאור הארגון:

מטרופולין היא חברה של <u>תחבורה ציבורית</u>.

החברה מפעילה כ-200 קווי שירות באשכול הנגב, באשכול השרון ואשכול נתניה–תל אביב באמצעות כ-800 אוטובוסים.

בחברה מועסקים כ-1,000 עובדים והמטה שלה ממוקם בכפר סבא.

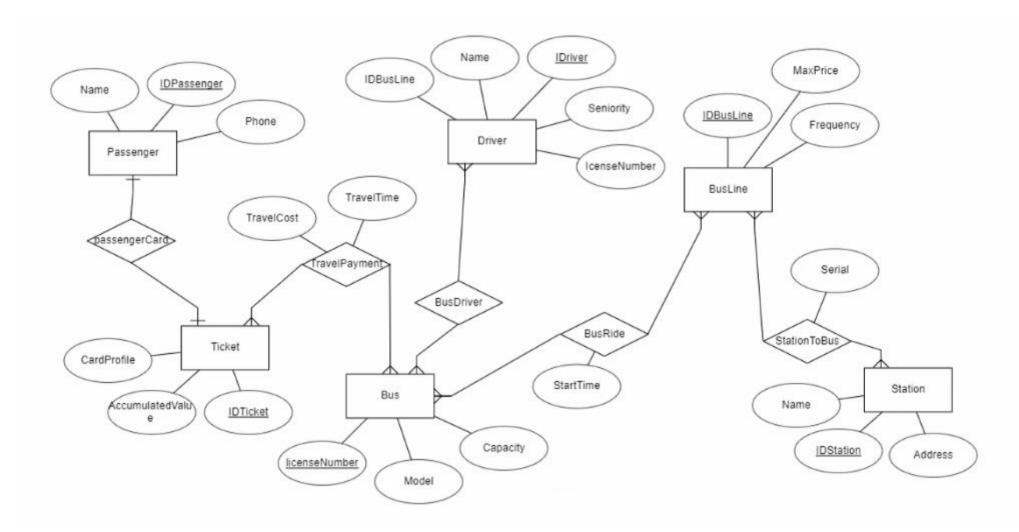
<u>הישויות הראשיות:</u>

- **אוטובוס:** לכל אוטובוס מוגדר לוחית הרישוי שלו, דגם ותכולה.
 - **נהג**: לכל קו נסיעה יש נהג.
- **תחנה:** לכל אוטובוס יש מסלול נסיעה המורכב ממספר תחנות.
- נוסע: לכל נוסע הנוסע בתחבורה הציבורית יש כרטיס באמצעותו הוא משלם.

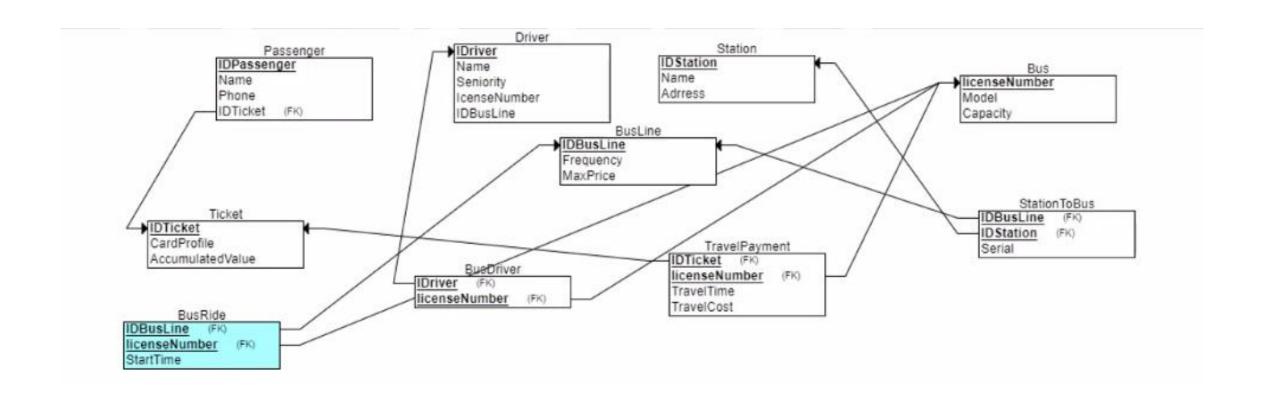
כאשר נוסע עולה לקו נסיעה הוא מעביר כרטיס אשר מכיל ערך צבור של כסף, וכך הוא משלם על הנסיעה. לכל נסיעה מחיר תשלום שונה בהתאם למרחק הנסיעה.

לכל קו נסיעה יש תדירות שונה, מספר נוסעים שונה ומסלול נסיעה שונה.

:ERD תרשים



: DSD תרשים



פירוט הישויות:

Ticket

פרופיל הכרטיס	Card Profile
מספר כרטיס	ID Ticket
ערך צבור	Accumulated Value

<u>Passenger</u>

שם	Name
מספר זיהוי	ID Passenger
טלפון	Phone
מספר כרטיס	ID Ticket

Bus Line

מספר קו נסיעה	ID Bus Line
תדירות	Frequency
מחיר מקסימלי לנסיעה	Max Price

Driver

מספר זיהוי	ID driver
שם	Name
ותק	Seniority
לוחית רישוי	License Number
מספר קו נסיעה	ld Bus Line

Station

שם	Name
מספר תחנה	ID Station
כתובת	Address

<u>Bus</u>

דגם	Model
לוחית רישוי	License Number
תכולה- מספר מושבים	Capacity

קשרים בין הישויות:

<u>שם הקשר</u>

לי	הקשר הוא קשר של רבים לרבים בין קו נסיעה לתחנה. לכל קו נסיעה יש מספר רב של תחנות בהן הוא עובר. לכל תחנה יש הרבה קווי נסיעה שעוברים דרכה.
לים	הקשר הוא קשר של רבים לרבים בין קו נסיעה לאוטובוס. לכל קו נסיעה יש כמה סוגים של אוטובוסים. לכל אוטובוס יש הרבה קווי נסיעה.
כי	הקשר הוא קשר של רבים לרבים בין האוטובוס לנהג. כל נהג יכול לנהוג על הרבה אוטובוסים. ולכל אוטובוס יש הרבה נהגים.
לפ	הקשר הוא קשר של רבים לרבים בין אוטובוס וכרטיס. לכל אוטובוס יש ניקובים רבים של כרטיסים. ע"י כרטיס אחד ניתן לנקב בהרבה אוטובוסים.
לי	הקשר הוא קשר של יחיד ליחיד בין הנוסע לכרטיס. לכל נוסע יש כרטיס נסיעה אחד. לכל כרטיס נסיעה יש בעלים יחידי.

```
IDriver INT NOT NULL.
  licenseNumber VARCHAR(15) NOT NULL,
  PRIMARY KEY (IDriver, licenseNumber),
  FOREIGN KEY (IDriver) REFERENCES Driver(IDriver).
  FOREIGN KEY (licenseNumber) REFERENCES Bus(licenseNumber)
CREATE TABLE Passenger
  Name VARCHAR (15) NOT NULL,
  IDPassenger INT NOT NULL,
  Phone VARCHAR (15) NOT NULL,
  IDTicket INT NOT NULL,
  PRIMARY KEY (IDPassenger).
  FOREIGN KEY (IDTicket) REFERENCES Ticket(IDTicket)
);
CREATE TABLE BusRide
 StartTime DATE NOT NULL,
 IDBusLine INT NOT NULL,
 licenseNumber VARCHAR(15) NOT NULL,
 PRIMARY KEY (IDBusLine, licenseNumber),
 FOREIGN KEY (IDBusLine) REFERENCES BusLine(IDBusLine),
 FOREIGN KEY (licenseNumber) REFERENCES Bus(licenseNumber)
 CREATE TABLE StationToBus
  Serial INT NOT NULL,
  IDBusLine INT NOT NULL,
  IDStation INT NOT NULL,
  PRIMARY KEY (IDBusLine, IDStation),
  FOREIGN KEY (IDBusLine) REFERENCES BusLine(IDBusLine),
  FOREIGN KEY (IDStation) REFERENCES Station(IDStation)
```

CREATE TABLE BusDriver

:CreatTable פקודת

```
CREATE TABLE Ticket
  CardProfile VARCHAR(15) NOT NULL,
  IDTicket INT NOT NULL,
  AccumulatedValue INT NOT NULL,
  PRIMARY KEY (IDTicket)
 CREATE TABLE BusLine
  IDBusLine INT NOT NULL,
  Frequency VARCHAR (15) NOT NULL,
  MaxPrice INT NOT NULL,
  PRIMARY KEY (IDBusLine)
CREATE TABLE Bus
 Model VARCHAR (15) NOT NULL.
 licenseNumber VARCHAR(15) NOT NULL,
 Capacity INT NOT NULL,
 PRIMARY KEY (licenseNumber)
CREATE TABLE TravelPayment
 TravelTime date NOT NULL,
 TravelCost INT NOT NULL,
 IDTicket INT NOT NULL,
 licenseNumber VARCHAR(15) NOT NULL,
 PRIMARY KEY (IDTicket, licenseNumber),
 FOREIGN KEY (IDTicket) REFERENCES Ticket(IDTicket),
 FOREIGN KEY (licenseNumber) REFERENCES Bus(licenseNumber)
```

```
CREATE TABLE Driver
(
   Name VARCHAR(15) NOT NULL,
   IDriver INT NOT NULL,
   Seniority INT NOT NULL,
   IcenseNumber VARCHAR(15) NOT NULL,
   IDBUSLINE INT NOT NULL,
   PRIMARY KEY (IDriver)
);

CREATE TABLE Station
(
   Name VARCHAR(15) NOT NULL,
   IDStation INT NOT NULL,
   Adrress VARCHAR(15) NOT NULL,
   PRIMARY KEY (IDStation)
);
```

```
insert into MAAYAN1.BUS (MODEL, LICENSENUMBER, CAPACITY)
values ('Higer KLQ6129Q', 537-79-64, 14);
insert into MAAYAN1.BUS (MODEL, LICENSENUMBER, CAPACITY)
values ('Volvo 7700', 130-26-50, 60);
insert into MAAYAN1.BUS (MODEL, LICENSENUMBER, CAPACITY)
values ('BYD K9 ', 355-61-55, 47);
insert into MAAYAN1.BUS (MODEL, LICENSENUMBER, CAPACITY)
values ('Higer KLQ6129Q', 975-10-65, 20);
insert into MAAYAN1.BUS (MODEL, LICENSENUMBER, CAPACITY)
values ('BYD K9 ', 489-12-22, 54);
insert into MAAYAN1.BUS (MODEL, LICENSENUMBER, CAPACITY)
values ('Volvo 7700', 426-81-51, 18);
insert into MAAYAN1.BUSDRIVER (IDRIVER, LICENSENUMBER)
values (813, 688-67-63);
insert into MAAYAN1.BUSDRIVER (IDRIVER, LICENSENUMBER)
values (788, 226-90-44);
insert into MAAYAN1.BUSDRIVER (IDRIVER, LICENSENUMBER)
values (538, 443-39-14);
insert into MAAYAN1.BUSDRIVER (IDRIVER, LICENSENUMBER)
values (412, 999-29-67);
insert into MAAYAN1.BUSDRIVER (IDRIVER, LICENSENUMBER)
values (326, 658-93-71);
```

:InsertTable פקודת

```
insert into MAAYAN1.BUSLINE (IDBUSLINE, FREQUENCY, MAXPRICE)
values (191, '35-min', 23);
insert into MAAYAN1.BUSLINE (IDBUSLINE, FREQUENCY, MAXPRICE)
values (536, '40-min', 58);
insert into MAAYAN1.BUSLINE (IDBUSLINE, FREQUENCY, MAXPRICE)
values (544, '30-min', 45);
insert into MAAYAN1.DRIVER (NAME, IDRIVER, SENIORITY, ICENSENUMBER, IDBUSLINE)
values ('Kay', 916, 26, 928-53-71, 769);
insert into MAAYAN1.DRIVER (NAME, IDRIVER, SENIORITY, ICENSENUMBER, IDBUSLINE)
values ('Bernie', 716, 6, 217-19-35, 713);
insert into MAAYAN1.DRIVER (NAME, IDRIVER, SENIORITY, ICENSENUMBER, IDBUSLINE)
values ('Lisa', 861, 34, 815-29-80, 659);
insert into MAAYAN1.DRIVER (NAME, IDRIVER, SENIORITY, ICENSENUMBER, IDBUSLINE)
values ('Nicky', 968, 20, 532-88-40, 147);
 insert into MAAYAN1.PASSENGER (NAME, IDPASSENGER, PHONE, IDTICKET)
 values ('Meryl', 622, 052-960-2646, 18411);
 insert into MAAYAN1.PASSENGER (NAME, IDPASSENGER, PHONE, IDTICKET)
values ('Carolyn', 934, 052-287-7783, 79563);
 insert into MAAYAN1.PASSENGER (NAME, IDPASSENGER, PHONE, IDTICKET)
values ('Joely', 859, 052-925-8640, 98293);
 insert into MAAYAN1.PASSENGER (NAME, IDPASSENGER, PHONE, IDTICKET)
 values ('Louise', 583, 054-706-7592, 45207);
```

:DropTable פקודת

```
drop table TRAVELPAYMENT;
drop table STATIONTOBUS;
drop table BUSDRIVER;
drop table PASSENGER;
drop table BUSRIDE;
drop table DRIVER;
drop table STATION;
drop table BUSLINE;
drop table BUSLINE;
drop table BUS;
drop table TICKET;
```

SOL> desc driver Nullable Default Comments VARCHAR2 (15) NAME INTEGER IDRIVER SENIORITY INTEGER ICENSENUMBER VARCHAR2 (15) IDBUSLINE INTEGER SOL> desc bus Name Nullable Default Comments MODEL VARCHAR2 (15) LICENSENUMBER VARCHAR2 (15) CAPACITY INTEGER SOL> desc busdriver Type Nullable Default Comments IDRIVER INTEGER LICENSENUMBER VARCHAR2 (15) SOL> desc busline Nullable Default Comments IDBUSLINE INTEGER FREQUENCY VARCHAR2 (15) MAXPRICE INTEGER SOL> desc stationtobus Type Nullable Default Comments SERIAL INTEGER IDBUSLINE INTEGER IDSTATION INTEGER SQL> desc busride Nullable Default Comments STARTTIME DATE IDBUSLINE INTEGER LICENSENUMBER VARCHAR2 (15)

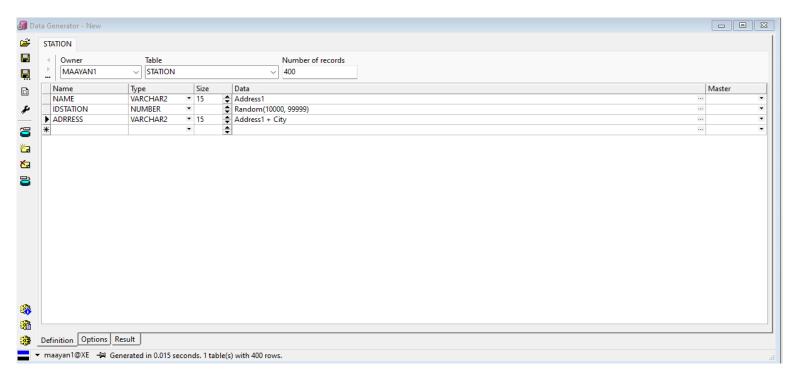
:DescTable פקודת

	passenger Type	Nullable Default Comments
NAME	VARCHAR2 (15)
IDPASSENG	ER INTEGER	
	VARCHAR2 (15)
IDTICKET	INTEGER	
SQL> desc	station	
Name	Type	Nullable Default Comments
	VARCHAR2 (15	
IDSTATION	N INTEGER	
ADRRESS	VARCHAR2 (15	5)
SQL> desc t		
Name	Type	Nullable Default Comments
CARDPROFILE	VARCHAR2 ((15)
IDTICKET	INTEGER	•
ACCUMULATED	VALUE INTEGER	
SQL> desc	travelpaym	
Name	Type	Nullable Default Comments
	ME DATE	
	ST INTEGE	
IDTICKET	INTEGE	R
LICENSEN	JMBER VARCHA	R2 (15)

:SelectAll פקודת

```
select * from busline
select * from busdriver
select * from busride
select * from driver
select * from passenger
select * from stationtobus
select * from station
select * from ticket
select * from travelpayment
select count (*) from bus
select count (*) from busline
select count (*) from busdriver
select count (*) from busride
select count (*) from driver
select count (*) from passenger
select count (*) from stationtobus
select count (*) from station
select count (*) from ticket
select count (*) from travelpayment
```

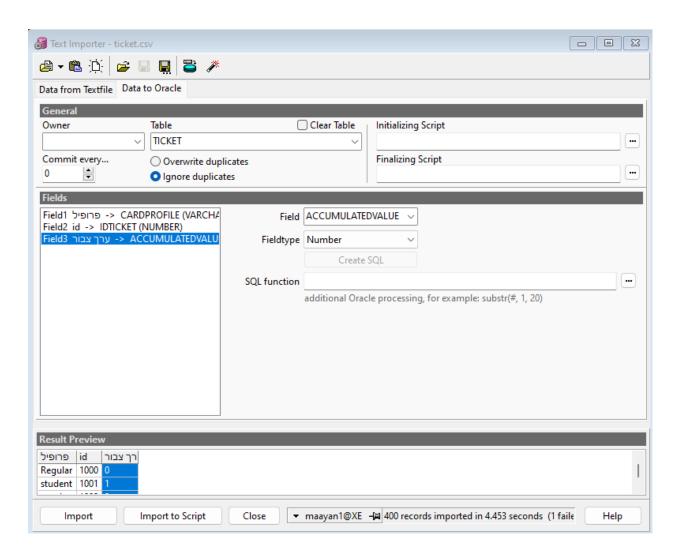
Data generator מילוי הטבלאות ע"י



1	48 Patrick Road	63319	48 Patrick Road
2	62nd Street	71940	62nd StreetMana
3	423 Alpharetta	89476	423 Alpharetta
4	31 Secada Stree	14067	31 Secada Stree
5	55 Reynolds Str	58192	55 Reynolds Str
6	54 Patrick Road	81102	54 Patrick Road
7	40 Nancy Street	80063	40 Nancy Street
8	55 Vega Road	49273	55 Vega RoadHig
9	10 Tottori Stre	30498	10 Tottori Stre
10	503 Sisto Road	62353	503 Sisto RoadR
11	22nd Street	12707	22nd StreetPeta

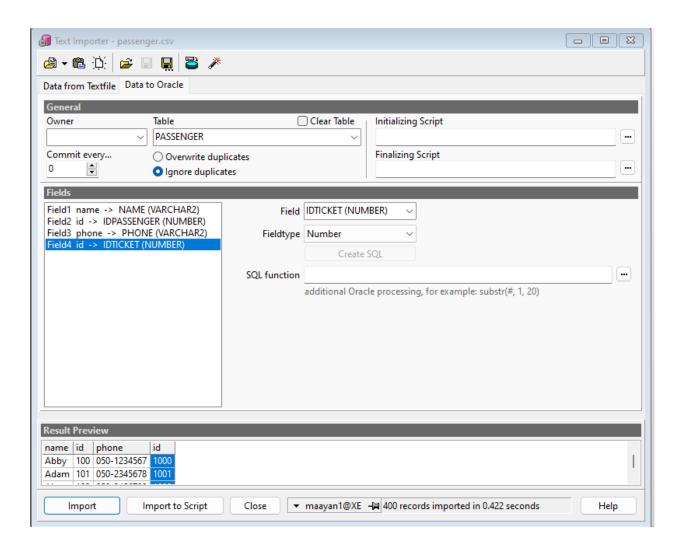
Station

IAME	IDSTATION	ADRRESS
8 Media	57094	88 MediaRoanoke
Davison Drive	60280	4 Davison Drive
Cole Street	58227	8 Cole StreetHi
9 Nashua Road	10997	89 Nashua RoadA
Aurora	90913	6 AuroraLe ches
6 Kilmer Drive	51324	46 Kilmer Drive
79 Weaving Driv	67532	79 Weaving Driv
5 Cheadle Road	96022	45 Cheadle Road
7 Bush Road	10270	77 Bush RoadKri
5 Kier Ave	71538	15 Kier AveWich
4 Meniketti Ro	13302	34 Meniketti Ro
20 Scorsese Ro	29337	820 Scorsese Ro
1st Street	22334	61st StreetOxfo
3 Dafoe Drive	50684	33 Dafoe DriveB
2 Ithaca Stree	20287	92 Ithaca Stree
7 København Ro	52820	37 København Ro
3 Pusan-city S	25168	63 Pusan-city S
4 Mclean Ave	10455	84 Mclean AveMe
49 Frost Road	65562	149 Frost RoadL
2 Clint Road	65756	22 Clint RoadCa
3 Judd Street	61167	93 Judd StreetK
58 Milano Stre	44491	658 Milano Stre



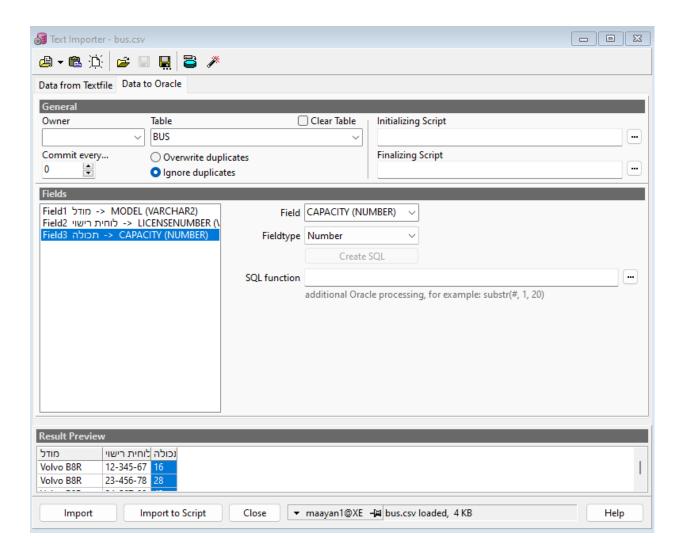
Ticket

	CARDPROFILE	IDTICKET	ACCUMULATEDVALUE
) 1	Regular	1000	0
2	student	1001	1
3	youth	1002	2
4	asoldier	1003	3
5	Veteran	1004	4
6	Regular	1005	5
7	student	1006	6
8	youth	1007	7
9	asoldier	1008	8
10	Veteran	1009	9
11	Regular	1010	10



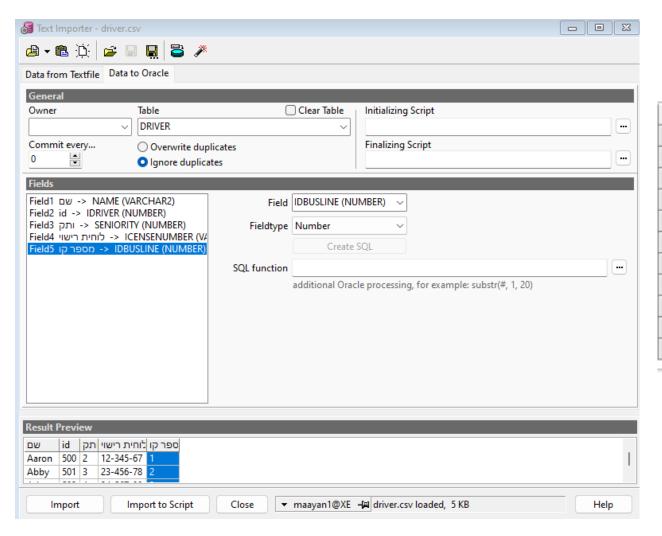
Passenger

		NAME	IDPASSENGER	PHONE	IDTICKET _
٠	1	Stan	342	088-3456789	1242
	2	Tate	343	088-4567890	1243
	3	Todd	344	088-5678901	1244
	4	Tom	345	088-6789012	1245
	5	Toni	346	088-7890123	1246
	6	Tory	347	088-8901234	1247
	7	Trey	348	088-9012345	1248
	8	Troy	349	088-0123456	1249
	9	Tyra	350	089-1234567	1250
	10	Vick	351	089-2345678	1251
	11	Walt	352	089-3456789	1252



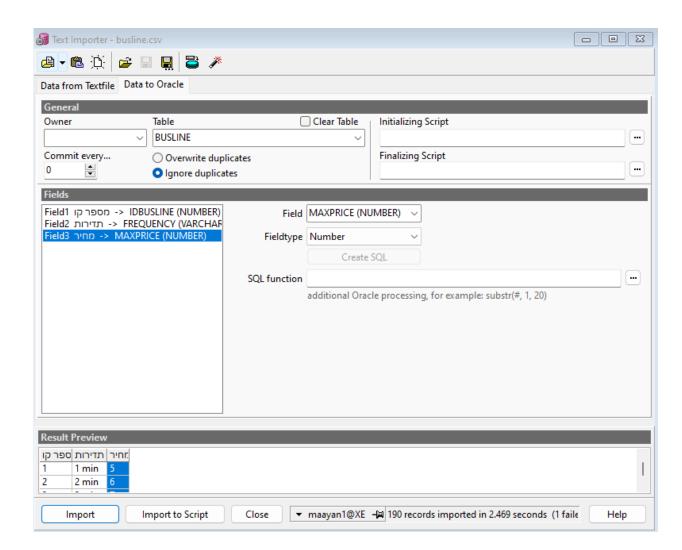
Bus

LICENSENUMBER	CAPACITY
12-345-67	16
23-456-78	28
34-567-89	43
45-678-90	21
56-789-01	35
67-890-12	50
78-901-23	25
89-012-34	14
90-123-45	10
01-234-56	11
12-345-68	38
	12-345-67 23-456-78 34-567-89 45-678-90 56-789-01 67-890-12 78-901-23 89-012-34 90-123-45 01-234-56



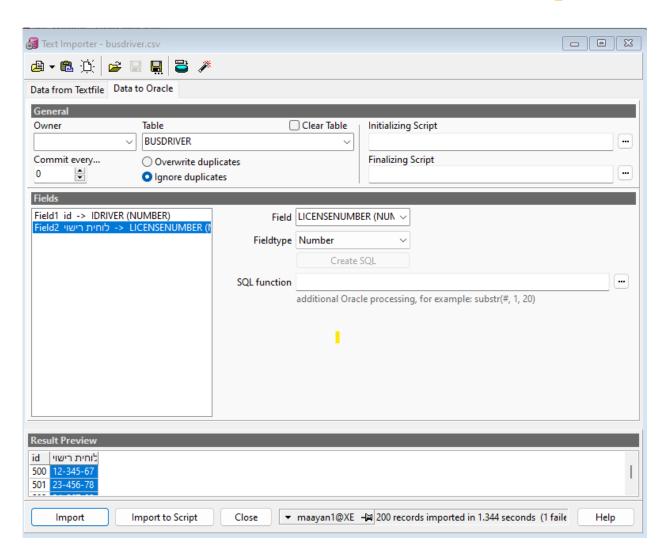
Driver

	NAME	IDRIVER	SENIORITY _	ICENSENUMBER	IDBUSLINE
1	Aaron	500	2	12-345-67	1
2	Abby	501	3	23-456-78	2
3	Adam	502	4	34-567-89	3
4	Alan	503	5	45-678-90	4
5	Alex	504	6	56-789-01	5
6	Alice	505	7	67-890-12	6
7	Allie	506	8	78-901-23	7
8	Alvin	507	9	89-012-34	8
9	Amber	508	10	90-123-45	9
10	Amy	509	11	01-234-56	10
11	Andy	510	12	12-345-68	11



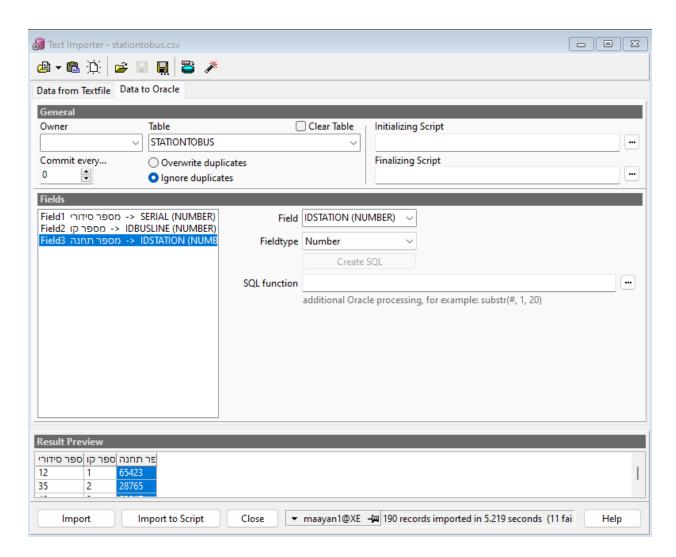
Bus Line

	IDBUSLINE	FREQUENCY	MAXPRICE
1	102	42 min	10
2	103	43 min	11
3	104	44 min	12
4	105	45 min	5
5	106	46 min	6
6	107	47 min	7
7	108	48 min	8
8	109	49 min	9
9	110	50 min	10
10	111	51 min	11
11	112	52 min	12



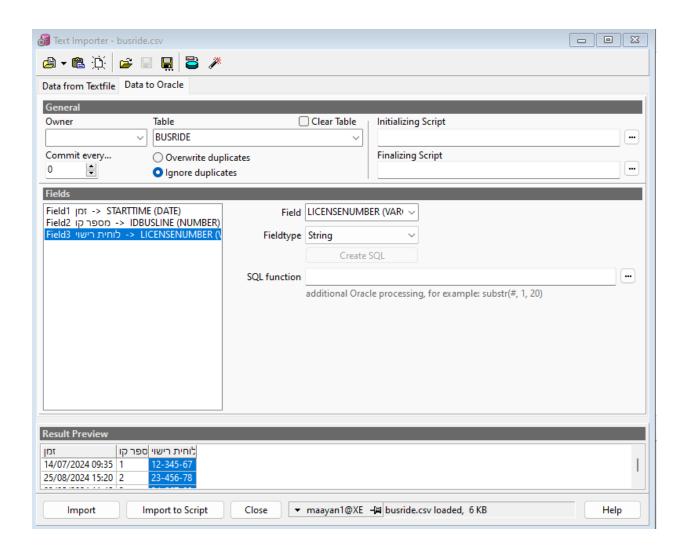
Bus Driver

	IDRIVER	LICENSENUMBER
1	500	12-345-67
2	501	23-456-78
3	502	34-567-89
4	503	45-678-90
5	504	56-789-01
6	505	67-890-12
7	506	78-901-23
8	507	89-012-34
9	508	90-123-45
10	509	01-234-56
11	510	12-345-68



Station to bus

	SERIAL	IDBUSLINE	IDSTATION _
1	15	102	51297
2	47	103	43928
3	20	104	62918
4	30	105	85426
5	8	106	29478
6	41	107	18635
7	2	108	52097
8	31	109	74385
9	35	110	58026
10	3	111	31745
11	46	112	94263



Bus Ride

	STARTTIME		IDBUSLINE	LICENSENUMBER
1	17/12/2024 14:55:00	•	102	23-456-88
2	27/07/2024 09:20:00	•	103	34-567-99
3	07/03/2024 13:45:00	•	104	45-678-10
4	18/11/2024 08:10:00	•	105	56-789-21
5	29/07/2024 12:35:00	•	106	67-890-32
6	09/02/2024 16:00:00	•	107	78-901-43
7	14/07/2024 09:35:00	•	108	89-012-54
8	25/08/2024 15:20:00	•	109	90-123-65
9	03/05/2024 11:45:00	•	110	01-234-76
10	06/11/2024 14:10:00	•	111	12-345-78
11	19/01/2024 10:55:00	•	112	23-456-89

הכנסת נתונים ע"י יצירת סקריפט בפייתון

```
import random
import cx_Oracle

def hours to hh mm(hours):
    hh = hours // 60
    mm = hours % 60
    return "{:02d}:{:02d}".format(hh, mm)

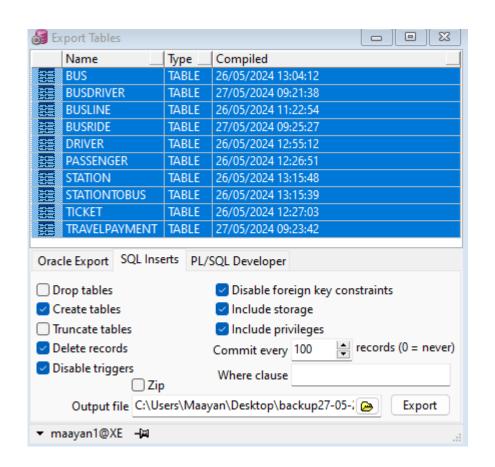
def read_file_content(file_path):
    try:
        with open(file_path, 'r') as file:
            content = file.readlines()
            return [line.strip() for line in content]
    except FileNotFoundError:
        print(f"File {file_path} not found.")
        return []
```

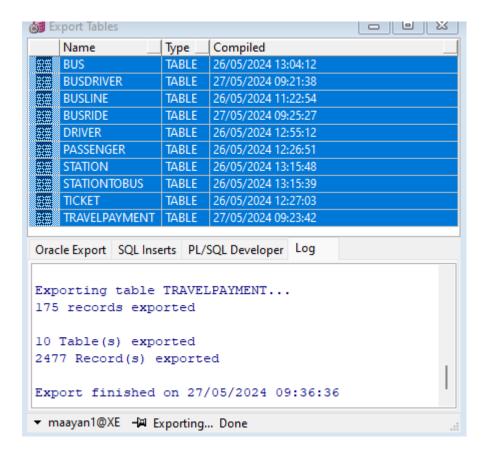
```
יצירת חיבור למסד הנתונים #
connection = cx_Oracle.connect(username, password, dsn)
cursor = connection.cursor()
# קריאת תוכן הקבצים
idtickets = read_file_content("idticket.txt")
licenses = read_file_content("license.txt")
if len(idtickets) == 0 or len(licenses) == 0:
    print("Failed to read tickets or licenses.")
else:
    שאילתת ה-INSERT לטבלה #
   insert_query = """
    INSERT INTO TravelPayment (TravelTime, TravelCost, IDTicket, licenseNumber)
    VALUES (TO DATE(:1, 'HH24:MI'), :2, :3, :4)
    for in range(400):
        הגרלת שעות ועלות #
        travel hours = random.randint(5 * 60, 12 * 60)
        travel cost = random.randint(5, 12)
        travel time = hours to hh mm(travel hours)
        # הגרלת כרטיס ורישיוו אקראיים
        id ticket = random.choice(idtickets)
        license number = random.choice(licenses)
        הכנסת הנתונים לטכלה #
        cursor.execute(insert query, (travel time, travel cost, id ticket, license number))
    שמירת השינויים במסד הנתונים #
    connection.commit()
    שליבור # סגירת ה-cursor והחיבור
    cursor.close()
    connection.close()
    print("Data inserted successfully.")
```

TravelPayMent

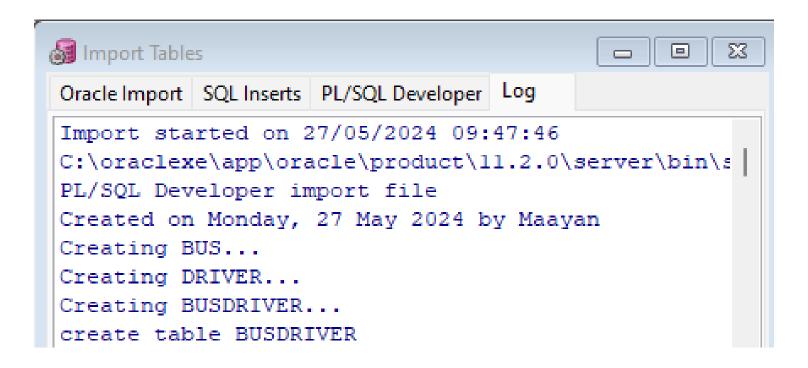
	TRAVELTIME		TRAVELCOST	IDTICKET	LICENSENUMBER
1	01/08/2024 11:30:00	•	10	1067	89-012-50
2	02/08/2024 13:40:00	•	11	1068	90-123-61
3	03/08/2024 09:15:00	•	7	1069	01-234-72
4	04/08/2024 10:45:00	•	11	1070	12-345-74
5	05/08/2024 14:25:00	•	11	1071	23-456-85
6	06/08/2024 08:50:00	•	12	1072	34-567-96
7	07/08/2024 11:20:00	•	12	1073	45-678-07
8	08/08/2024 13:10:00	•	7	1074	56-789-18
9	09/08/2024 08:20:00	•	5	1075	67-890-29
10	10/08/2024 10:55:00	•	7	1076	78-901-40
11	11/08/2024 14:35:00	•	5	1077	89-012-51





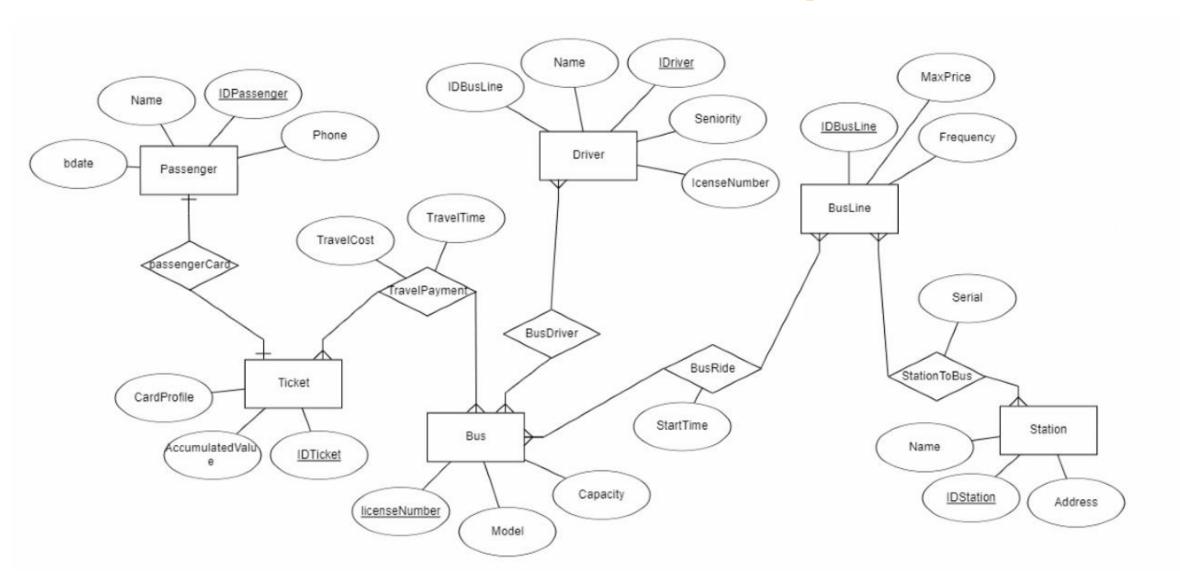


שחזור

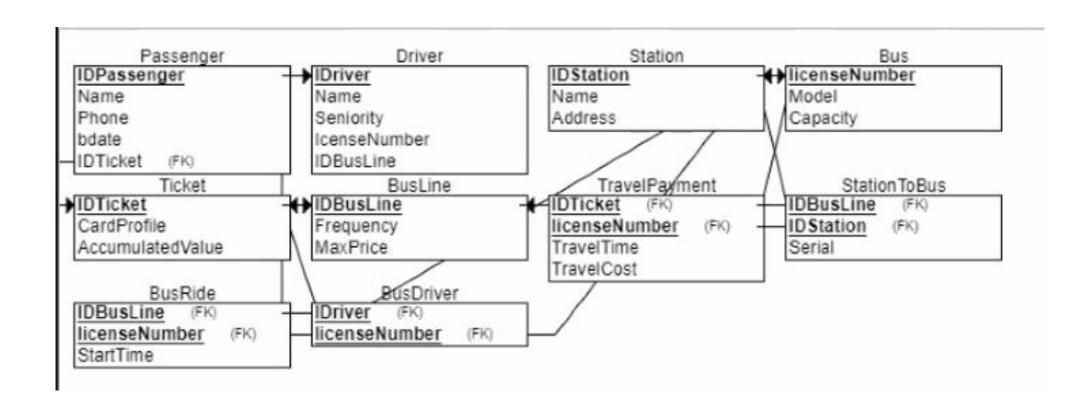


שלב 2 שאילתות

טבלת ERD מתוקנת

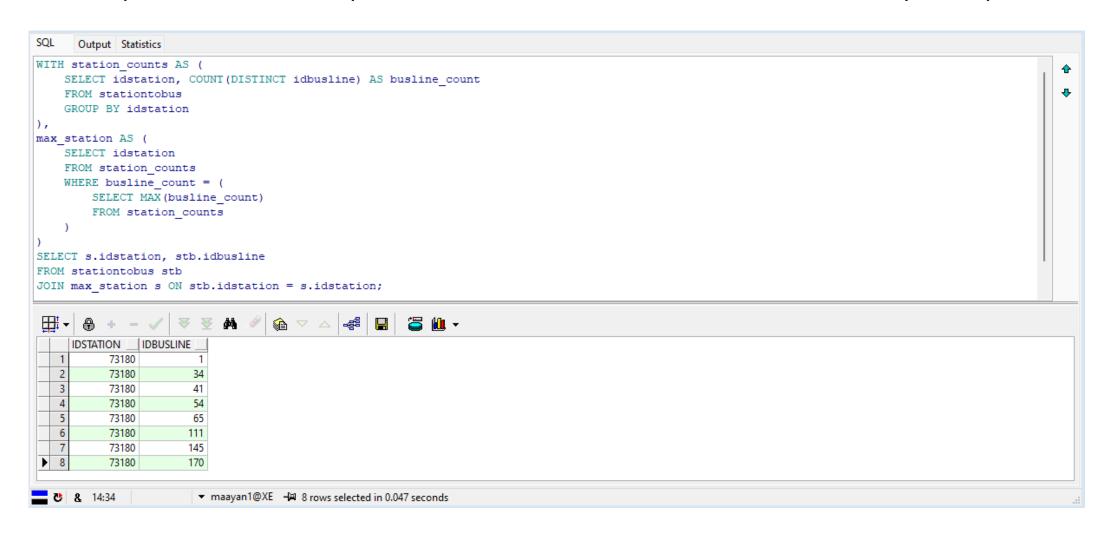


טבלת DSD מתוקנת



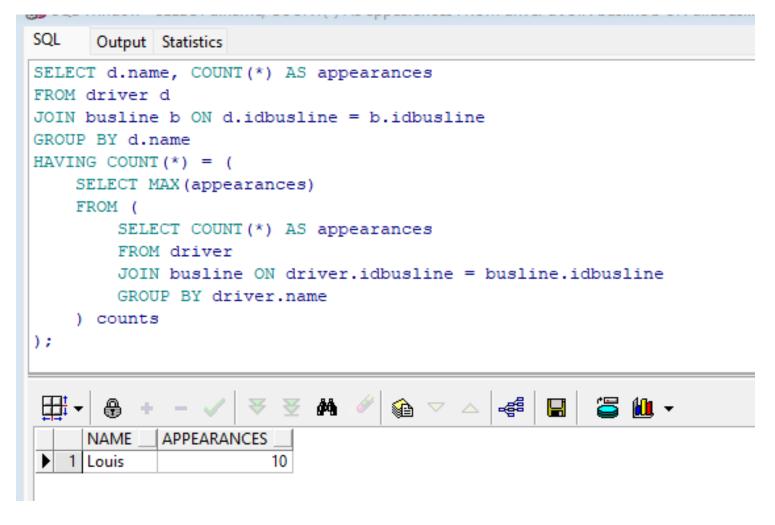
שאילתת select1 ללא פרמטרים

בשאילתה נרצה להציג את רשימת האוטובוסים שעוברים בתחנה ספציפית, כאשר התחנה היא התחנה שעוברים בה הכי הרבה קווים. ניתן לראות כי התחנה שמספרה 73180 היא התחנה שמספר הקווים שעוצרים בה הוא מקסימלי.



שאילתת select2 ללא פרמטרים

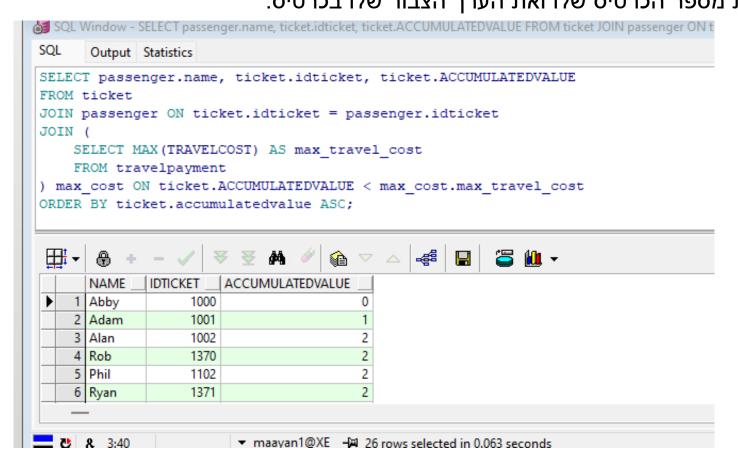
בשאילתה נרצה להציג את שם הנהג שנוהג על הכי הרבה קווים מבין כלל הנהגים בחברה. וכן הצגנו את מספר הקווים עליהם הוא נוהג.



שאילתת select3 ללא פרמטרים

בשאילתה נרצה להציג את הנוסעים שלא יכולים לנסוע בנסיעה היקרה ביותר בחברה.

לכל נוסע יש כרטיס שבאמצעותו הוא יכול להתנייד בקווי האוטובוס. בכל כרטיס שמור הערך הצבור שיש בו, כאשר הערך הצבור קטן ממחיר הנסיעה הנוסע לא יוכל לנסוע באותה נסיעה. הצגנו את הספר הכרטיס שלו ואת הערך הצבור שלו בכרטיס.



שאילתת select4 ללא פרמטרים

בשאילתה נרצה להציג את קווי האוטובוס שזמן התדירות שלהם קטן מחצי שעה וגם זמן היציאה שלהם למסלול הוא לפני השעה 13:00.

נציג אותם ממוינים בסדר עולה.

```
SELECT b.idbusline
FROM (
    SELECT idbusline
    FROM Busline
    WHERE TO_NUMBER(REGEXP_SUBSTR(frequency, '^\d+')) < 30
JOIN busride r ON b.idbusline = r.idbusline
WHERE TO CHAR(r.starttime, 'HH24:MI:SS') < '13:00:00'
ORDER BY b.idbusline ASC:
      IDBUSLINE
              11
              13
              16
              21
              23
              25
              32
```

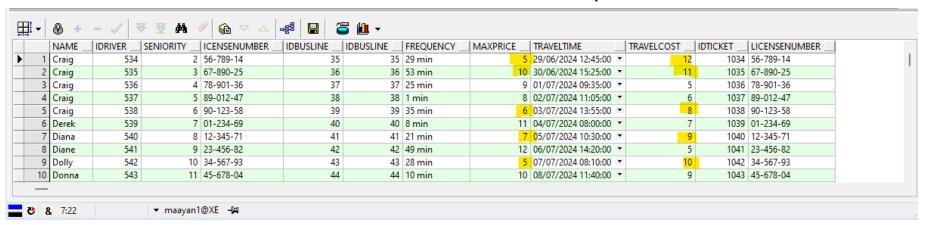
update1 שאילתת

בשאילתה נרצה לעדכן את מחיר הנסיעה באוטובוס כאשר, לכל קו אוטובוס יש מחיר מקסימלי ולכל נסיעה יש מחיר נסיעה, אם מחיר הנסיעה גדול מהמחיר המקסימלי נעדכן את מחיר הנסיעה להיות שווה למחיר המקסימלי.

```
UPDATE TravelPayment
SET travelcost = (
    SELECT BusLine.maxprice
    FROM Driver
    INNER JOIN BusLine ON Driver. Idbusline = BusLine. Idbusline
    WHERE TravelPayment.Licensenumber = Driver.Icensenumber
WHERE Licensenumber IN (
    SELECT TravelPayment.Licensenumber
    FROM TravelPayment
    INNER JOIN Driver ON TravelPayment.Licensenumber =
Driver. Icensenumber
    INNER JOIN BusLine ON Driver. Idbusline = BusLine. Idbusline
    WHERE TravelPayment.travelcost > BusLine.maxprice
```

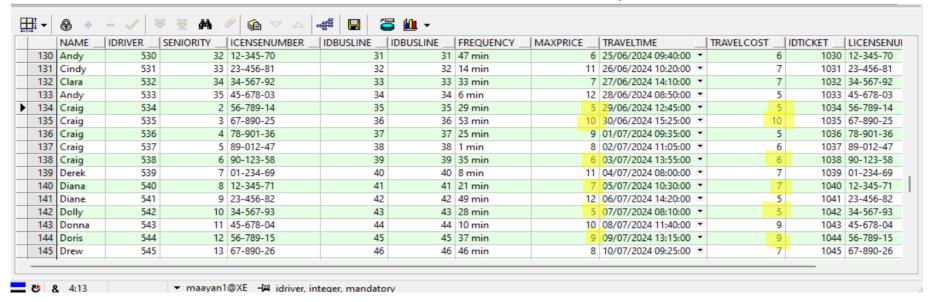
לפני עדכון השאילתה

ניתן לראות כי מחירי הנסיעה גדולים מהמחיר המקסימלי האפשרי



אחרי עדכון השאילתה

ניתן לראות כי מחירי הנסיעה גדולים מהמחיר המקסימלי האפשרי



update2 שאילתת

בשאילתה נרצה לעדכן את הפרופיל בכרטיס נסיעה של הנוסע בהתאם לתאריך הלידה שלו. לצורך כך היינו צריכות להוסיף עמודה נוספת לטבלת הנוסע בשם bdate שתשמור את תאריך הלידה של כל נוסע.

```
SQL Output Statistics חדשה ALTER TABLE passenger
ADD bdate DATE;
```

מילוי העמודה בתאריכים רנדומליים

```
UPDATE passenger
SET bdate = TO_DATE('1950-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE(0, 365 * 70)
```

הסרנו את הTIME השארנו רק את התאריך

```
UPDATE passenger
SET bdate = TRUNC(TO_DATE('1950-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE(0, 365 * 70))
```

אחרי הוספת העמודה

	NAME	IDPASSENGER	PHONE	IDTICKET	BDATE	
1	Stan	342	088-3456789	1242	18/07/1989	•
2	Tate	343	088-4567890	1243	18/02/1988	•
3	Todd	344	088-5678901	1244	23/10/1992	•
4	Tom	345	088-6789012	1245	26/01/1983	•
5	Toni	346	088-7890123	1246	11/07/1960	•
6	Tory	347	088-8901234	1247	03/08/1990	•
7	Trey	348	088-9012345	1248	12/04/2004	•
8	Troy	349	088-0123456	1249	05/03/1952	•
9	Tyra	350	089-1234567	1250	01/03/1965	•
10	Vick	351	089-2345678	1251	16/08/2018	•
11	Walt	352	089-3456789	1252	10/02/1974	•
12	Will	353	089-4567890	1253	07/04/2004	•
13	Yves	354	089-5678901	1254	11/05/2009	•
14	Zane	355	089-6789012	1255	26/08/1962	•
15	Zeke	356	089-7890123	1256	08/08/1998	•
16	Zola	357	089-8901234	1257	10/04/1997	•
17	Axel	358	089-9012345	1258	06/10/1988	•
18	Brad	359	089-0123456	1259	14/12/1967	•
19	Brie	360	090-1234567	1260	12/08/1959	•
20	Cary	361	090-2345678	1261	17/08/1969	•
21	Chad	362	090-3456789	1262	19/01/1992	•
22	Clay	363	090-4567890	1263	06/06/1967	•
23	Dale	364	090-5678901	1264	16/02/1960	•
24	Dean	365	090-6789012	1265	06/07/2015	•
25	Edie	366	090-7890123	1266	27/11/1970	•
26	Ella	367	090-8901234	1267	15/03/1965	•
27	Erik	368	090-9012345	1268	,,	•
28	Faye	369	090-0123456	1269	16/01/2013	•
29	Finn	370	091-1234567	1270	19/12/2012	•
30	Fred	371	091-2345678	1271	02/01/1985	•
31	Greg	372	091-3456789	1272	30/05/2012	•
32	Hank	373	091-4567890	1273	21/06/1975	•
33	Hugh	374	091-5678901	1274	23/06/1983	•

לפני הוספת העמודה

	NAME	IDPASSENGER	PHONE	IDTICKET
1	Stan	342	088-3456789	1242
2	Tate	343	088-4567890	1243
3	Todd	344	088-5678901	1244
4	Tom	345	088-6789012	1245
5	Toni	346	088-7890123	1246
6	Tory	347	088-8901234	1247
7	Trey	348	088-9012345	1248
8	Troy	349	088-0123456	1249
9	Tyra	350	089-1234567	1250
10	Vick	351	089-2345678	1251
11	Walt	352	089-3456789	1252
12	Will	353	089-4567890	1253
13	Yves	354	089-5678901	1254
14	Zane	355	089-6789012	1255
15	Zeke	356	089-7890123	1256
16	Zola	357	089-8901234	1257
17	Axel	358	089-9012345	1258
18	Brad	359	089-0123456	1259
19	Brie	360	090-1234567	1260
20	Cary	361	090-2345678	1261
21	Chad	362	090-3456789	1262
22	Clay	363	090-4567890	1263
23	Dale	364	090-5678901	1264
24	Dean	365	090-6789012	1265
25	Edie	366	090-7890123	1266
26	Ella	367	090-8901234	1267
27	Erik	368	090-9012345	1268
28	Faye	369	090-0123456	1269
29	Finn	370	091-1234567	1270
30	Fred	371	091-2345678	1271
31	Greg	372	091-3456789	1272
32	Hank	373	091-4567890	1273
33	Hugh	374	091-5678901	1274

בשאילתה נרצה לעדכן את הפרופיל בכרטיס נסיעה של הנוסע בהתאם לתאריך הלידה שלו. אחרי שהוספנו עמודה ומילאנו אותה,

יצרנו שאילתה שבודקת את טווח הגילאים של הנוסעים ולפי זה מקטלגת לאזרח ותיק (אזרח מעל גיל 65) ולנוער בגילאי 5-18).

```
UPDATE ticket
SET cardprofile =
    CASE
        WHEN EXTRACT (YEAR FROM (SELECT bdate FROM passenger WHERE idticket = ticket.idticket)) < EXTRACT (YEAR
FROM CURRENT DATE - INTERVAL '65' YEAR) THEN 'Veteran'
        WHEN EXTRACT (YEAR FROM CURRENT DATE) - EXTRACT (YEAR FROM (SELECT bdate FROM passenger WHERE idticket =
ticket.idticket)) BETWEEN 5 AND 18 THEN 'youth'
    END
WHERE EXISTS (
    SELECT 1
    FROM passenger
    WHERE idticket = ticket.idticket
    AND (
        EXTRACT (YEAR FROM bdate) < EXTRACT (YEAR FROM CURRENT DATE - INTERVAL '65' YEAR)
        OR EXTRACT (YEAR FROM CURRENT DATE) - EXTRACT (YEAR FROM bdate) BETWEEN 5 AND 18
```

_			_	
		BDATE		CARDPROFILE
Þ	1	22/03/2016	•	Regular
	2	14/01/2006	•	student
	3		•	Veteran
	4	01/01/2007	•	Veteran
	5	08/11/2016	•	Regular
	6			Veteran
	7	23/12/2016	•	Regular
	8	20/04/2011	•	asoldier
	9	01/06/1957	•	Veteran
	10	23/06/2006	•	youth
	-11	07/01/2008	•	Regular
	12	02/02/1958	•	youth
	13	18/03/1950		student
	14	22/06/1958	•	youth
	15	28/10/1958	•	student

Œ	# +	⊕ +	nie.	1	*	포	44	
4		BDATE		CAR	PRO	FILE		
1		22/03/2016	*	youth				
	2	14/01/2006	*	youth				
	3	30/06/2015	•	youth				
	4	01/01/2007	*	youth				
	5	08/11/2016	•	youth				
6		18/06/2015	•	youth				
7		23/12/2016	•	youth				
8		20/04/2011	•	youth				
9		01/06/1957		Veteran				
	10	23/06/2006	*	youth				
	11	07/01/2008	*	youth				
	12	02/02/1958	•	Veteran				
13		18/03/1950	950 Veteran					
	14	22/06/1958	▼ Veteran					
	15	28/10/1958	•	Veteran				
	16	04/11/1956	•	Veter	an			

לפני עדכון השאילתה

ניתן לראות כי עבור אזרח בגיל 8 רשום שהפרופיל הוא פרופיל רגיל.

אחרי עדכון השאילתה

ניתן לראות כי עבור אזרח בגיל 8 רשום שהפרופיל הוא פרופיל נוער.

delete1 שאילתת

בשאילתה נרצה למחוק את כל קווי האוטובוס שאין להם מסלול נסיעה ולא עוברים בשום תחנה וכן שמספר המושבים (תכולה) בהם קטן מ20.

```
DELETE FROM busline
WHERE idbusline IN (
    SELECT busline.idbusline
    FROM busline
    LEFT JOIN stationtobus ON busline.idbusline = stationtobus.idbusline
    LEFT JOIN busride ON busline.idbusline = busride.idbusline
    LEFT JOIN bus ON busride.licensenumber = bus.licensenumber
    WHERE stationtobus.idstation IS NULL OR bus.capacity < 20
;(</pre>
```

אחרי המחיקה

לפני המחיקה

```
SELECT busline.idbusline, bus.capacity
FROM busline
LEFT JOIN stationtobus ON busline.idbusline = stationtobus.idbusline
LEFT JOIN busride ON busline.idbusline = busride.idbusline
LEFT JOIN bus ON busride.licensenumber = bus.licensenumber
WHERE stationtobus.idstation IS NULL or bus.capacity < 20;
```

1	IE	⊕ + −	✓ ₹	\$
)	IE	- DIVISION IN		_
- 1	_	DBUSLINE	CAPACITY	
	1	97		10
	2	97		10
	3	97		10
	4	98		12
	5	100		52
	6	102		10
	7	103		47
	8	104		23
	9	107		39
10	0	108		54
1	1	109		10
12	2	109		10
13	3	112		10
14	4	113		15
1.	5	114		11
10	6	115		13
1	7	117		10
18	8	118		30
19	9	119		40
20	0	121		10
2	1	122		34
2	2	123		17
2.	3	124		19
24	4	124		19
2:	5	125		50
2	6	126		10

delete2 שאילתת

בשאילתה נרצה למחוק את כל הנהגים השמורים במערכת אך שאינם משובצים לשום קו נסיעה וגם שהותק שלהם קטן מ10 שנים.

.null יוכל להיות idbusline לצורך כך בטבלת מriver נצטרך לשנות שהשדה

```
SQL Output Statistics

ALTER TABLE driver
MODIFY (idbusline NULL);
```

שאילתת המחיקה

```
SQL Output Statistics

DELETE FROM driver
WHERE idbusline IS NULL AND seniority < 10;
```

אחרי המחיקה

לפני המחיקה

SELECT *
FROM driver

WHERE idbusline IS NULL and seniority<10

1	# +	+	- 🗸	₹ ₩		∉ □ =
Ī		NAME	IDRIVER _	SENIORITY	ICENSENUMBER _	IDBUSLINE
T	1	Spike	306	1	58-612-35	
T	2	Sinead	346	1	49-596-70	
Ť	3	Aidan	362	2	28-194-44	
Ī	4	Lance	341	3	91-863-10	
Ī	5	Devon	315	3	74-664-55	
Ī	6	Rawlins	303	1	14-279-71	
Ī	7	Cheech	375	1	61-807-79	
Ī	8	Pam	336	1	96-324-18	
I	9	Miki	335	2	47-887-86	
Ī	10	Barbara	324	3	76-401-14	
Ī	11	Roscoe	389	1	56-369-96	
Ī	12	April	305	3	93-457-17	
I	13	Chalee	332	1	75-669-63	
Ī	14	Praga	379	1	52-717-67	
Ī	15	Bret	382	3	46-184-30	
Ī	16	Amanda	352	3	24-452-84	
Ī	17	Hookah	326	3	83-269-18	
Ī	18	Al	350	2	70-252-31	
Ī	19	Pamela	325	3	46-246-52	
I	20	Emma	371	1	75-789-63	
I	21	Barbara	330	2	87-358-73	
Ī	22	Brent	309	1	67-322-22	
I	23	Richard	327	2	68-594-79	
I	24	Alice	301	5	12-345-67	
Ī	25	Bobby	302	3	23-456-78	

שאילתת select1 עם פרמטרים

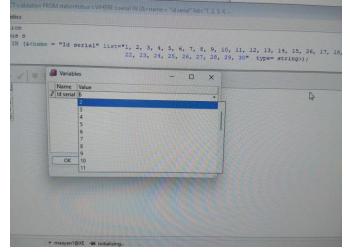
בשאילתה נרצה להציג את רשימת התחנות שמספרן הסידורי במסלול הוא הפרמטר שהוכנס, ואת התדירות של קווים העוברים בתחנה.

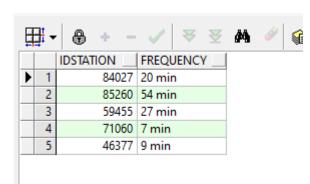
```
SELECT s.idstation, b.frequency
FROM stationtobus s
JOIN busline b ON s.idbusline = b.idbusline
WHERE s.serial IN (&<name = "Id serial" list="1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 26, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30" type= string>);
```

.list עם פרמטר מסוג select ניתן לראות כי יש כאן שאילתת

ניתן לבחור מספר מ1-30 המציין את המספר הסידורי של התחנה לפי סדר המסלול של כל קו.

כאשר נבחר מספר יוצגו לנו כל רשימת התחנות שמספרן הסידורי הוא המספר שנבחר.



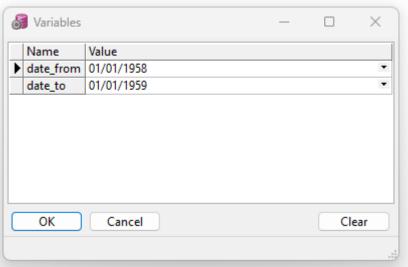


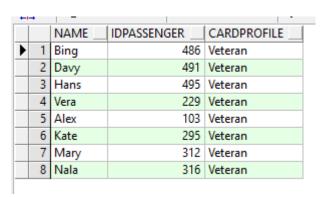
שאילתת select2 עם פרמטרים

בשאילתה נרצה להציג את רשימת כל הנוסעים בטווח גילאים מסוים לפי תאריך הלידה של כל נוסע והצגת הפרופיל שלהם.

```
SELECT p.name, p.idpassenger, t.cardprofile
FROM passenger p
JOIN ticket t ON p.idticket = t.idticket
WHERE p.bdate BETWEEN &<name=date_from type="date"> AND &<name=date_to type="date">;
```

ניתן לראות כי יש כאן שאילתת select עם פרמטר מסוג date. ניתן לבחור תאריכים בטווחים מסוימים, ונקבל את כל הנוסעים שנמצאים בטווח הגילאים הנרחר.



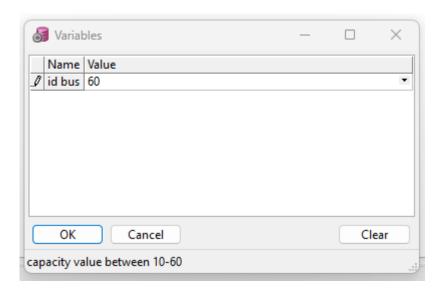


שאילתת select3 עם פרמטרים

בשאילתה נרצה להציג את רשימת קווי האוטובוס שהתכולה שלהם קטנה מהפרמטר שהוקש.

```
select d.idbusline,d.icensenumber, b.capacity
from driver d
join bus b on d.icensenumber= b.licensenumber
where b.capacity < &<name="id bus" hint= "capacity value between 10-60" type=integer required=true>
```

ניתן לראות כי יש כאן שאילתת select עם פרמטר מסוג hint. ניתן לבחור מספר בטווח של 10-60 ולקבל את רשימת האוטובוסים שעומדים בתכולה.



+1-		_		-
		IDBUSLINE	ICENSENUMBER	CAPACITY
Þ	1	35	56-789-14	10
	2	36	67-890-25	23
	3	37	78-901-36	49
	4	38	89-012-47	15
	5	39	90-123-58	31
	6	40	01-234-69	42
	7	41	12-345-71	10
	8	42	23-456-82	58
	9	43	34-567-93	27
	10	44	45-678-04	51
	-11	45	56-789-15	36
	12	46	67-890-26	24
	13	47	78-901-37	40
	14	48	89-012-48	10
	15	49	90-123-59	57
		2 3 4 5 6 7 8 9 10 11 12 13	DBUSLINE	IDBUSLINE ICENSENUMBER 2 36 67-890-14 2 36 67-890-25 3 37 78-901-36 4 38 89-012-47 5 39 90-123-58 6 40 01-234-69 7 41 12-345-71 8 42 23-456-82 9 43 34-567-93 10 44 45-678-04 11 45 56-789-15 12 46 67-890-26 13 47 78-901-37 14 48 89-012-48

שאילתת select4 עם פרמטרים

בשאילתה נרצה להציג את קווי האוטובוס שמחיר הנסיעה שלהם הוא הפרמטר שהוקש ואת הנהג של כל אחד מהקווים.

```
SQL Output Statistics

SELECT d.idbusline, d.name
FROM travelpayment t

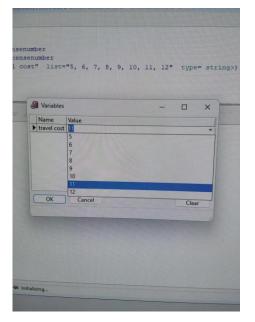
JOIN bus b ON b.licensenumber = t.licensenumber

JOIN driver d ON b.licensenumber = d.icensenumber

WHERE t.travelcost in (&<name= "travel cost" list="5, 6, 7, 8, 9, 10, 11, 12" type= string>)
```

.list עם פרמטר מסוג select ניתן לראות כי יש כאן שאילתת

ניתן לבחור מספר בטווח של 5-12 ולקבל את רשימת האוטובוסים שמחיר הנסיעה שלהם זהה למספר שהוקש.

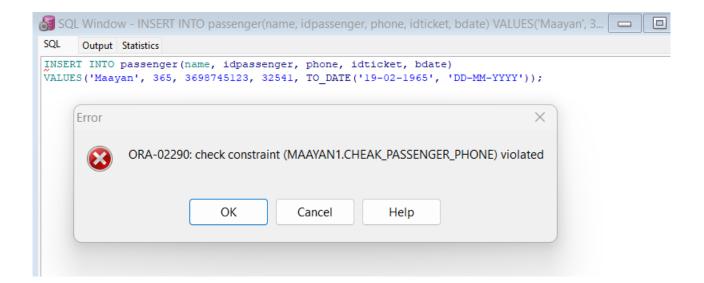


∓ 1										
		IDBUSLINE	NAME							
Þ	1	66	Grace							
	2	72	Helen							
	3	138	Nikki							
	4	144	Louis							
	5	152	Quinn							
	6	162	Rosa							
	7	168	Sarah							
	8	4	Alan							

אילוץ מספר 1

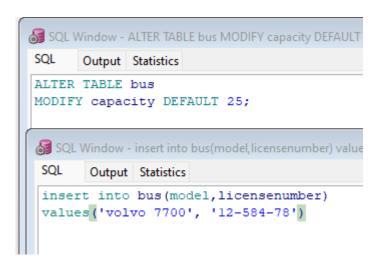
אילוץ מספר טלפון של נוסע. מספר הטלפון חייב להתחיל בתו 0.

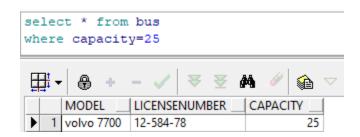
```
alter table passenger
add constraint cheak_passenger_phone
check (phone LIKE '0%');
```



אילוץ מספר 2

אילוץ על מספר המקומות באוטובוס.נהפוך את כמות המקומות לערך דיפולטיבי של 25.

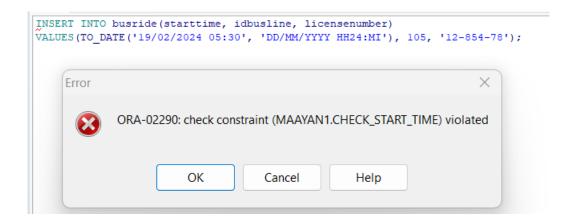




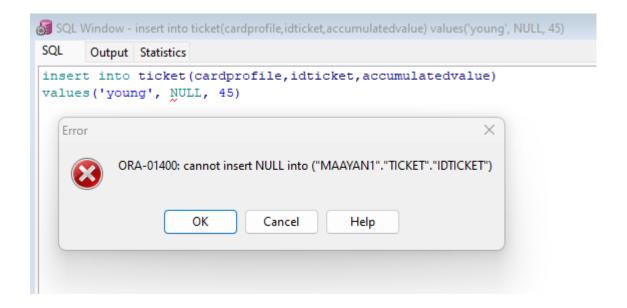
אילוץ מספר 3

אילוץ על שעת היציאה של קווי האוטובוס. כל קו אוטובוס יכול לצאת למסלול מהשעה 06:00 ועד השעה 23:00.

```
ALTER TABLE busride
ADD CONSTRAINT check_start_time
CHECK (TO_NUMBER(TO_CHAR(starttime, 'HH24')) BETWEEN 6 AND 23);
```



ALTER TABLE ticket MODIFY idticket NOT NULL;

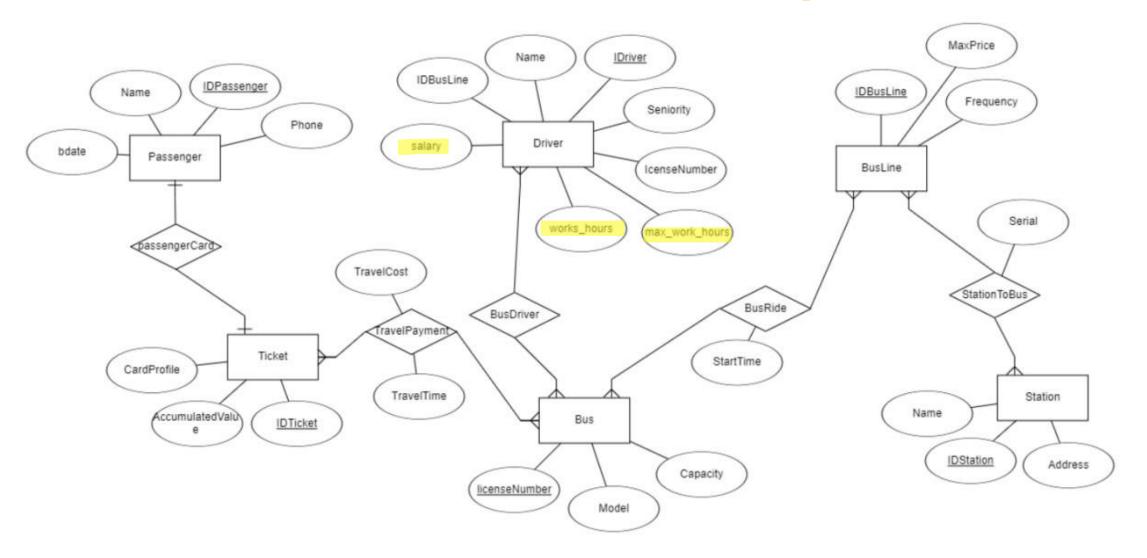


4 אילוץ מספר

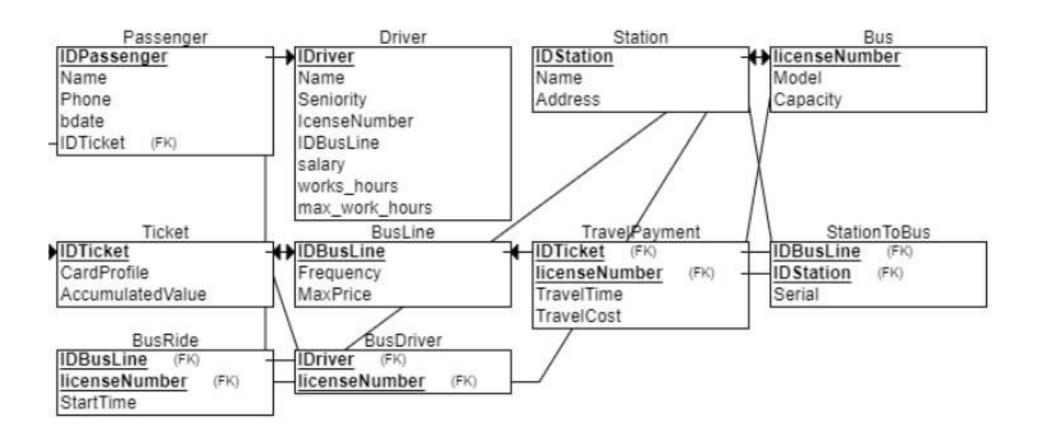
אילוץ על מספר כרטיס נסיעה.

שלב 3 תכנות

טבלת ERD מתוקנת



טבלת DSD מתוקנת



הטבלה של הdriver לאחר השינויים

		NAME _	IDRIVER	SENIORITY	ICENSENUMBER _	IDBUSLINE	WORK_HOURS	MAX_WORK_HOURS	SALARY
	-1	Craig	534	2	56-789-14	35	84	182	5497
	2	Craig	535	3	67-890-25	36	106	182	8875
	3	Craig	536	4	78-901-36	37	140	182	6009
	4	Craig	537	5	89-012-47	38	63	182	5452
- 3	5	Craig	538	6	90-123-58	39	150	182	5309
	6	Derek	539	7	01-234-69	40	59	182	8982
	7	Diana	540	8	12-345-71	41	79	182	9384
•	8	Diane	541	9	23-456-82	42	179	182	11647
	9	Dolly	542	10	34-567-93	43	55	182	9919
	10	Donna	543	11	45-678-04	44	112	182	9248
	11	Doris	544	12	56-789-15	45	76	182	7023
	12	Drew	545	13	67-890-26	46	58	182	5960
- 33	13	Dolly	546	14	78-901-37	47	177	182	12407
	14	Dolly	547	15	89-012-48	48	77	182	8249
	15	Edith	548	16	90-123-59	49	157	182	12346
	16	Edwin	549	17	01-234-60	50	82	182	6208
	17	Elena	550	18	12-345-72	51	84	182	7681
	18	Eli	551	19	23-456-83	52	55	182	9476
	19	Ellie	552	20	34-567-94	53	154	182	6889
- 1	20	Elton	553	21	45-678-05	54	161	182	14220

פונקציה מספר 1

```
הפונקציה Driver_Bonus בודקת:
```

.0-100 נמצא בטווח <u>threshold_hour</u> האם <u>threshold_hour</u> נמצא בטווח

עוברת על כל הנהגים בטבלה driver ומעדכנת את שכר הנהגים שעבדו שעות נוספות (מעל 150 שעות חודשיות) ומעדכנת את שכר הנהג בהתאם לאחוז הניתן (percent).

מחייבת את השינויים במאגר הנתונים (commit).

:מכינה REF CURSOR המכיל את הנהגים שעבדו שעות נוספות עם הפרטים הבאים

מזהה הנהג (idriver)

שם הנהג (name)

(new_salary) השכר החדש

(old_salary) השכר הישן

במקרה של חריגה, הפונקציה מתמודדת עם החריגות ומחזירה NULL

חריגה illegal_hour אם threshold_hour חריגה

חריגה illegal_percent אם illegal_percent

חריגות אחרות מטופלות על ידי הצגת הודעת שגיאה והתגלגלות לאחור (ROLLBACK).

```
CREATE OR REPLACE FUNCTION Driver Bonus (percent IN NUMBER, threshold hour IN INTEGER)
RETURN SYS REFCURSOR IS
  message VARCHAR2 (100);
CURSOR c drivers IS
    SELECT idriver, name, salary, work hours
    FROM driver
    FOR UPDATE OF salary;
  result cursor SYS REFCURSOR;
  illegal hour EXCEPTION;
                                                                               old salary
  illegal percent EXCEPTION;
                                                                                   FROM driver
 v new salary NUMBER;
 max work hours CONSTANT INTEGER := 182;
BEGIN
                                                                               EXCEPTION
IF threshold hour < 150 OR threshold hour > max work hours THEN
    message := 'Threshold hour should be an integer between 150 to 182';
RAISE illegal hour;
                                                                               RETURN NULL;
END IF;
  IF percent <= 0 OR percent > 100 THEN
                                                                               RETURN NULL;
    message := 'Percent should be a number between 1 to 100';
                                                                               WHEN OTHERS THEN
RAISE illegal percent;
END IF;
                                                                               | | SQLERRM);
                                                                               ROLLBACK;
FOR driver IN c drivers LOOP
                                                                               RETURN NULL;
IF driver.work hours BETWEEN 150 AND 182 THEN
                                                                               END;
      v new salary := driver.salary + ((driver.salary * percent) / 100);
UPDATE driver
      SET salary = v new salary
      WHERE CURRENT OF c drivers;
    END IF;
  END LOOP;
COMMIT;
```

```
OPEN result cursor FOR
    SELECT idriver, name,
           salary AS new salary,
          TRUNC(salary / (1 + (percent / 100))) AS
   WHERE work hours BETWEEN 150 AND 182;
 RETURN result cursor;
 WHEN illegal hour THEN
    DBMS OUTPUT.PUT LINE(message);
WHEN illegal percent THEN
    DBMS OUTPUT.PUT LINE(message);
    DBMS OUTPUT.PUT LINE ('An unexpected error occurred: '
```

דוגמת הרצה:

עבור: precent=10 threshold_hour=152

O	Variable	Type	Va	lue
	result	Cursor	▼ <	Cursor>
	percent	Float	* 10	
	threshold_hour	Float	· 15	2
V			₹	

```
Driver ID: 538, Name: Craig, New Salary: 14366, Old Salary: 13060
Driver ID: 541, Name: Diane, New Salary: 4414617, Old Salary: 4013288
Driver ID: 546, Name: Dolly, New Salary: 122400, Old Salary: 111272
Driver ID: 548, Name: Edith, New Salary: 37593, Old Salary: 34175
Driver ID: 552, Name: Ellie, New Salarv: 16247, Old Salarv: 14770
Driver ID: 553, Name: Elton, New Salarv: 58265, Old Salarv: 52968
Driver ID: 554, Name: Emily, New Salary: 221136, Old Salary: 201032
Driver ID: 555, Name: Emma, New Salary: 26489, Old Salary: 24080
Driver ID: 557, Name: Erin, New Salary: 64887, Old Salary: 58988
Driver ID: 561, Name: Frank, New Salary: 170232, Old Salary: 154756
Driver ID: 565, Name: Grace, New Salarv: 123644, Old Salarv: 112403
Driver ID: 568, Name: Gwen, New Salary: 130767, Old Salary: 118879
Driver ID: 571, Name: Helen, New Salary: 105534, Old Salary: 95940
Driver ID: 572, Name: Henry, New Salary: 22546, Old Salary: 20496
Driver ID: 584, Name: Jenny, New Salary: 226579, Old Salary: 205980
Driver ID: 591, Name: Joel, New Salarv: 11384, Old Salarv: 10349
```

דוגמת הרצה לחריגה:

עבור:

precent=3

threshold_hour=120

	O	Variable	Туре	Value
		result	Cursor	<cursor></cursor>
		percent	Float	3
١		threshold_hour	Float	120
÷	V		•	

Percent should be a number between 1 to 100

Error: ORA-01001: invalid cursor

פונקציה מספר 2

:Get_Busline_Stations הפונקציה

יצירת REF CURSOR, הפונקציה מחזירה REF CURSOR אשר יכיל את הנתונים של קווי האוטובוס והתחנות שלהם. c_buslines מפורש: c_buslines נפתח כדי לבחור את כל ה- id של קווי האוטובוס מתוך טבלת c_stationtobus. הגדרת CURSOR מרומז: c_stations יפתח בתוך לולאה כדי לבחור את התחנות של כל קו אוטובוס. משתני רשומה: משתנים v_station_id ו- v_station_name מוגדרים כדי לאחסן את נתוני התחנה.

.c_buslines עוברת על כל רשומה בקורסור FOR לולאת קווי אוטובוס: לולאה

פתיחת קורסור: פתיחת הקורסור c_stations כדי לבחור את תחנות הקו

לולאת תחנות: לולאת LOOP פנימית לעבור על כל התחנות שנבחרו בקורסור c_stations.

v_station_name-ו v_station_id שליפת נתוני התחנה הנוכחית לתוך המשתנים

עיבוד תחנה: הדפסת נתוני התחנה.

ומחזיר הודעה מתאימה במקרה של שגיאה.

פונקציה מספר 1

```
CREATE OR REPLACE FUNCTION Get Busline Stations
RETURN SYS REFCURSOR IS
 result cursor SYS REFCURSOR;
CURSOR c buslines IS
    SELECT DISTINCT idbusline
    FROM stationtobus;
c stations SYS REFCURSOR;
v station id station.idstation%TYPE;
v station name station.name%TYPE;
BEGIN
FOR busline record IN c buslines LOOP
UPDATE stationtobus
    SET idbusline = idbusline
   WHERE idbusline = busline_record.idbusline;
COMMIT;
OPEN c stations FOR
      SELECT s.idstation, s.name
      FROM stationtobus stb
      JOIN station s ON stb.idstation = s.idstation
      WHERE stb.idbusline = busline record.idbusline;
LOOP
      FETCH c stations INTO v station id, v station name;
      EXIT WHEN c stations%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Busline ID: ' || busline record.idbusline ||
                           ', Station ID: ' || v station id ||
                          ', Station Name: ' || v_station_name);
END LOOP;
```

```
CLOSE c_stations;
  END LOOP;
OPEN result_cursor FOR
    SELECT stb.idbusline, s.idstation, s.name AS
station_name
    FROM stationtobus stb
    JOIN station s ON stb.idstation = s.idstation
    ORDER BY stb.idbusline, s.idstation;
RETURN result_cursor;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An unexpected error occurred:
' || SQLERRM);
    RETURN NULL;
END;
```

דוגמת הרצה:

קיבלנו עבור כל קו אוטובוס את התחנות שבהן הוא עובר (המסלול שלו).

```
Busline ID: 5, Station ID: 15193, Station Name: 707 Gaza
Busline ID: 5, Station ID: 71060, Station Name: 25 Marin Drive
Busline ID: 6, Station ID: 44993, Station Name: 52 Junior Stree
Busline ID: 6, Station ID: 66103, Station Name: 82 Barry Blvd
Busline ID: 6, Station ID: 83043, Station Name: 8 Armatrading S
Busline ID: 6, Station ID: 85260, Station Name: 44 Beaverton St
Busline ID: 11, Station ID: 78359, Station Name: 56 Hilton
Busline ID: 18, Station ID: 81573, Station Name: 89 Mumbai Road
Busline ID: 20, Station ID: 35274, Station Name: 929 Vannelli Av
Busline ID: 23, Station ID: 41740, Station Name: 91 Jason Blvd
Busline ID: 31, Station ID: 46859, Station Name: 89 Andrea Road
Busline ID: 34, Station ID: 62469, Station Name: 84 Roberts Road
```

פרוצידורה מספר 1

:add_passenger_and_ticket הפרוצדורה

מוסיפה נוסע חדש ומנפיקה כרטיס חדש עבורו.

להלן פירוט הפעולות שהפרוצדורה מבצעת:

חישוב גיל הנוסע: הפרוצדורה מחשבת את גיל הנוסע לפי תאריך הלידה הנוכחי, קובעת את פרופיל הנוסע לפי הגיל: הפרוצדורה מגדירה את הפרופיל של הנוסע בהתאם לגיל המחושב. קיימים ארבעה פרופילים:

.18-לגילאים בין 5 ל <u>Student</u>

לגילאים בין 19 ל-29. <u>Youth</u>

.65 לגילאים בין 30 ל-Adult

.65 לגילאים מעל <u>Senior</u>

קביעת מספר כרטיס חדש: הפרוצדורה מוצאת את המספר הגבוה ביותר של כרטיס קיים בטבלת ticket ומגדילה אותו ב1 כדי לקבל מספר כרטיס חדש.

פרוצידורה מספר 1

```
CREATE OR REPLACE PROCEDURE add passenger and ticket (
    p passenger id NUMBER,
    p passenger name VARCHAR2,
    p passenger phone VARCHAR2,
    p passenger birth date DATE,
    p stored value NUMBER
) IS
    v profile VARCHAR2(20);
   v age NUMBER;
   v ticket number NUMBER;
BEGIN
v age := FLOOR(MONTHS BETWEEN(SYSDATE, p passenger birth date) / 12);
IF v age BETWEEN 5 AND 18 THEN
        v profile := 'Student';
    ELSIF v age BETWEEN 19 AND 29 THEN
        v profile := 'Youth';
    ELSIF v age BETWEEN 30 AND 65 THEN
        v profile := 'Adult';
    ELSIF v age > 65 THEN
        v profile := 'Senior';
    ELSE
       v profile := 'Invalid';
    END IF;
SELECT NVL (MAX (idticket), 0) + 1 INTO v ticket number FROM ticket;
INSERT INTO ticket (cardprofile, idticket, accumulatedvalue)
    VALUES (v profile, v ticket number, p stored value);
INSERT INTO passenger (name, idpassenger, phone, idticket, bdate)
    VALUES (p passenger name, p passenger id, p passenger phone, v ticket number,
p passenger birth date);
COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE APPLICATION ERROR(-20001, 'An error occurred: ' || SQLERRM);
END;
```

דוגמת הרצה:

הוספנו נוסע חדש, וכתוצאה מההוספה נוצר לנו כרטיס נסיעה חדש לאותו הנוסע.

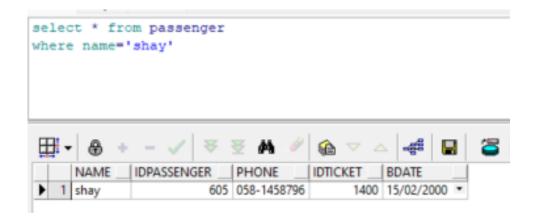
```
begin
         -- Call the procedure
        add passenger and ticket(p passenger id => :p passenger id,
                                   p passenger name => :p passenger name,
                                   p passenger phone => :p passenger phone,
                                   p_passenger_birth_date => :p_passenger_birth_date,
                                   p stored value => :p stored value);
       end;
 ● Variable
                      Type
                                      Value
 Float
                                     ▼ 605

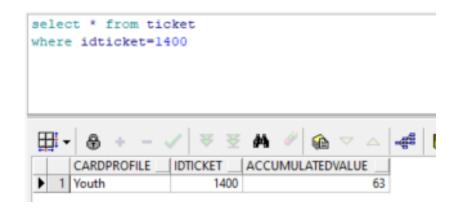
    □ p_passenger_name

                      String
                                     shay
 p_passenger_phone
                      String
                                     058-1458796
  p_passenger_birth_date
                      Date

    15/02/2000

▶ ▼ p_stored_value
                                    ▼ 63
```





פרוצידורה מספר 2

הפרוצדורה process_driver_salaries מעבדת את המשכורות של נהגים ומעדכנת אותם בהתאם לנתוני הוותק שלהם. הפרוצדורה בודקת אם שנות הוותק של הנהג הן 10 שנים ומעלה. אם כן, הפרוצדורה מעלה את משכורתו ב-1500 ומעדכנת את מספר הנהגים שקיבלו העלאה.

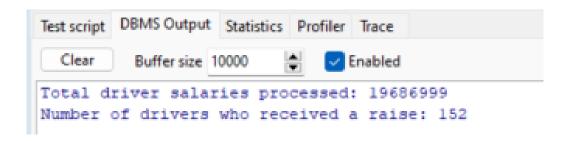
הפרוצדורה מוסיפה את משכורת הנהג (לאחר העדכון, אם היה) לסכום הכולל של כל המשכורות המעובדות. הפרוצדורה מציגה את הסכום הכולל של כל המשכורות המעובדות. הפרוצדורה מציגה את מספר הנהגים שקיבלו העלאה. במקרה שאין נתוני נהגים, הפרוצדורה מדפיסה הודעה מתאימה.

פרוצידורה מספר 2

```
CREATE OR REPLACE PROCEDURE process driver salaries IS
v driver id driver.idriver%TYPE;
   v driver name driver.name%TYPE;
   v experience driver.seniority%TYPE;
   v salary driver.salary%TYPE;
   v total salaries NUMBER := 0;
   v num drivers raised NUMBER := 0;
   v cursor SYS REFCURSOR;
BEGIN
OPEN v cursor FOR
       SELECT idriver, name, seniority, salary
       FROM driver:
LOOP
       FETCH v cursor INTO v driver id, v driver name, v experience, v salary;
       EXIT WHEN v cursor%NOTFOUND;
IF v experience >= 10 THEN
v salary := v salary + 1500;
           v num drivers raised := v num drivers raised + 1;
  END IF;
UPDATE driver SET salary = v salary WHERE idriver = v driver id;
v total salaries := v total salaries + v salary;
    END LOOP;
CLOSE v cursor;
DBMS OUTPUT.PUT LINE('Total driver salaries processed: ' || v total salaries);
DBMS OUTPUT.PUT LINE('Number of drivers who received a raise: ' ||
v num drivers raised);
    COMMIT;
EXCEPTION
    WHEN NO DATA FOUND THEN
        DBMS OUTPUT.PUT LINE('No driver data found.');
   WHEN OTHERS THEN
        ROLLBACK;
        RAISE APPLICATION_ERROR(-20001, 'An error occurred: ' || SQLERRM);
END;
```

דוגמת הרצה:

מספר הנהגים ששנות הותק שלהם הוא 10 ומעלה הוא 152. והסכום של המשכורות הוא 1,968,6999



DECLARE result bonus SYS REFCURSOR; result stations SYS REFCURSOR; idriver NUMBER; name VARCHAR2 (100); new salary NUMBER; old salary NUMBER; percent NUMBER := 10; threshold hour NUMBER := 160; Get Busline Stations busline id NUMBER; station id NUMBER; station name VARCHAR2(100); BEGIN BEGIN result bonus := Driver Bonus (percent, threshold hour); LOOP FETCH result bonus INTO idriver, name, new_salary, old_salary; EXIT WHEN result bonus%NOTFOUND; DBMS_OUTPUT.PUT_LINE('Driver ID: ' || idriver || ', Name: ' || name || ', New Salary: ' || new_salary || ', Old Salary: ' || TRUNC(old salary)); END LOOP; CLOSE result bonus; EXCEPTION WHEN OTHERS THEN DBMS OUTPUT.PUT LINE('Error in Driver Bonus: ' || SQLERRM); END; BEGIN result stations := Get Busline Stations();

תוכנית ראשית שקוראת לפונקציה 1 ולפרוצידורה 1

```
LOOP
      FETCH result stations INTO busline id,
station id, station name;
      EXIT WHEN result stations%NOTFOUND;
      DBMS OUTPUT.PUT LINE('Busline ID: ' ||
busline id ||
                            ', Station ID: ' ||
station_id ||
                            ', Station Name: ' ||
station_name);
    END LOOP;
Get Busline Stations
    CLOSE result stations;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS OUTPUT.PUT LINE('Error in
Get Busline Stations: ' || SQLERRM);
  END;
END;
```

```
תוכנית ראשית שקוראת לפונקציה 2 ולפרוצידורה 2
```

```
DECLARE
result SYS REFCURSOR;
  idriver NUMBER;
 name VARCHAR2 (100);
 new salary NUMBER;
  old salary NUMBER;
 percent NUMBER := 10;
threshold hour NUMBER := 160;
BEGIN
BEGIN
    result := Driver Bonus (percent, threshold hour);
LOOP
      FETCH result INTO idriver, name, new salary, old salary;
      EXIT WHEN result%NOTFOUND;
      DBMS OUTPUT.PUT LINE('Driver ID: ' || idriver ||
                           ', Name: ' || name ||
                           ', New Salary: ' || new_salary ||
                           ', Old Salary: ' || TRUNC(old salary));
    END LOOP;
CLOSE result;
  EXCEPTION
   WHEN OTHERS THEN
      DBMS OUTPUT.PUT LINE('Error calling Driver Bonus: ' || SQLERRM);
  END;
BEGIN
    process driver salaries;
 EXCEPTION
   WHEN OTHERS THEN
      DBMS OUTPUT.PUT LINE('Error calling process driver salaries: ' ||
SQLERRM);
  END;
END;
```