

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



### תוכן עניינים:

וניאור וואר גון	
תרשים Erd	
Dsd תרשים	5
פירוט הישויות	6
קשרים בין הישויות	7
createTable	8
insertTable	9
DropTable	10
DescTable	
SelectAll	12
מילוי הנתונים בטבלאות	13-22
גיבוי ושחזור	23-24
שלב 2	25
חדש Erd חדש	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	46-50

### תוכן עניינים:

שלב 3שלב	51
Erd תרשים	52
Dsd תרשים	
1 פונקציה	
2 פונקציהפונקציה.	60-61
פרוצידורה 1פרוצידורה 1	62-64
פרוצידורה ?	65-67
Main 1	68
Main 2	

### 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.

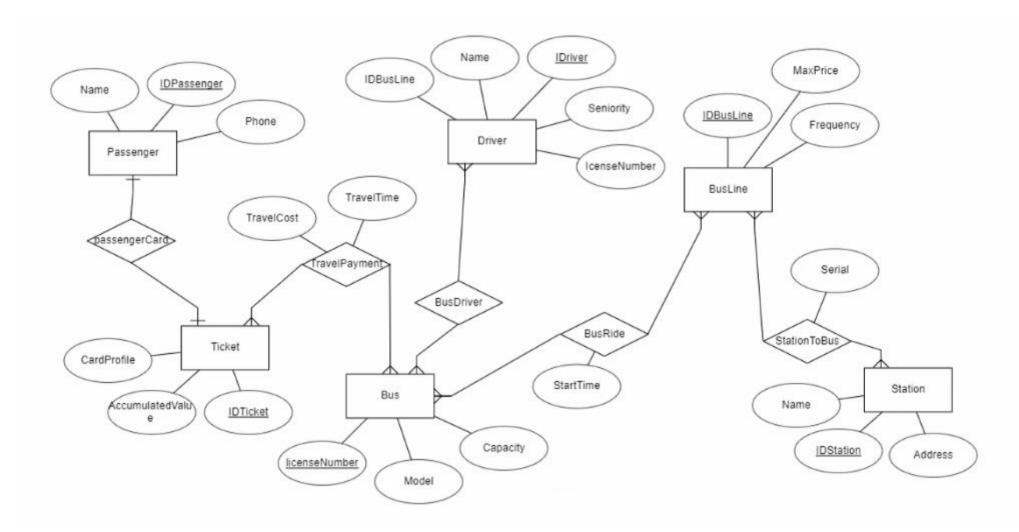
#### <u>הישויות הראשיות:</u>

- 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.

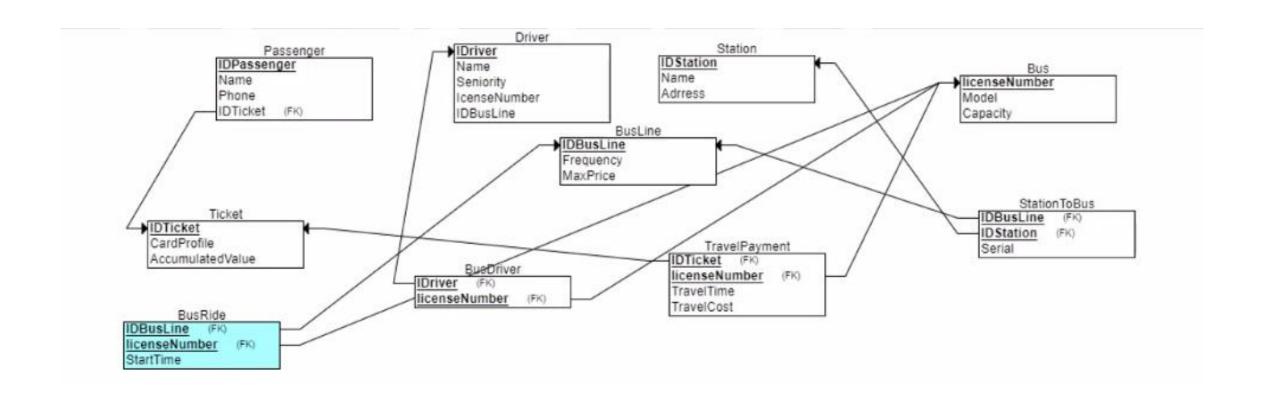
כאשר נוסע עולה לקו נסיעה הוא מעביר כרטיס אשר מכיל ערך צבור של כסף, וכך הוא משלם על הנסיעה. לכל נסיעה מחיר תשלום שונה בהתאם למרחק הנסיעה.

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)

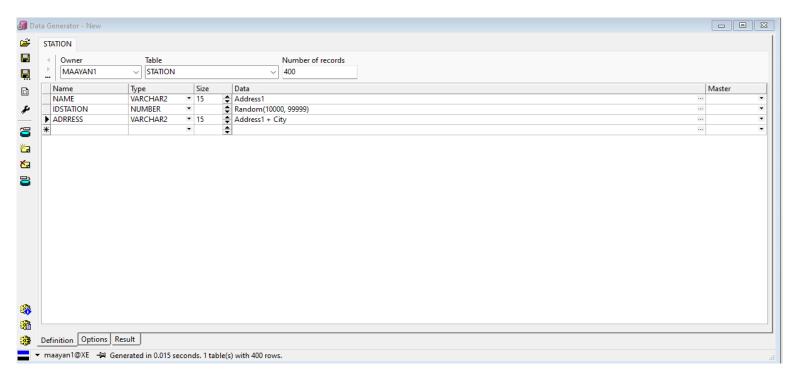
### :DescTable פקודת

	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
```

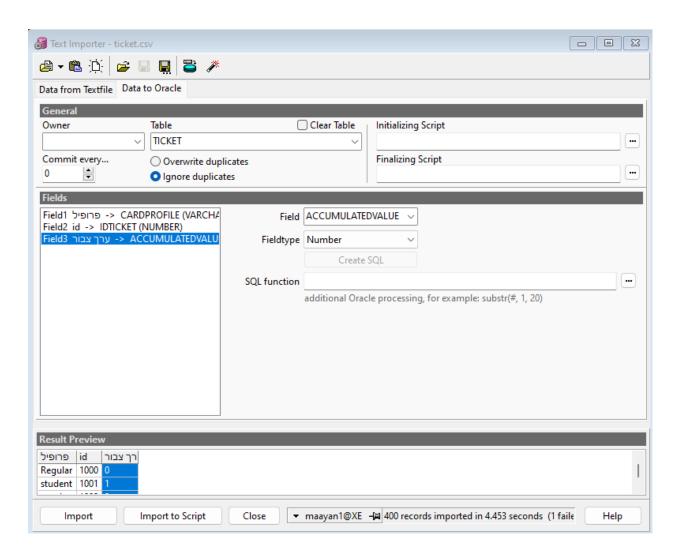
# Data generator מילוי הטבלאות ע"י



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

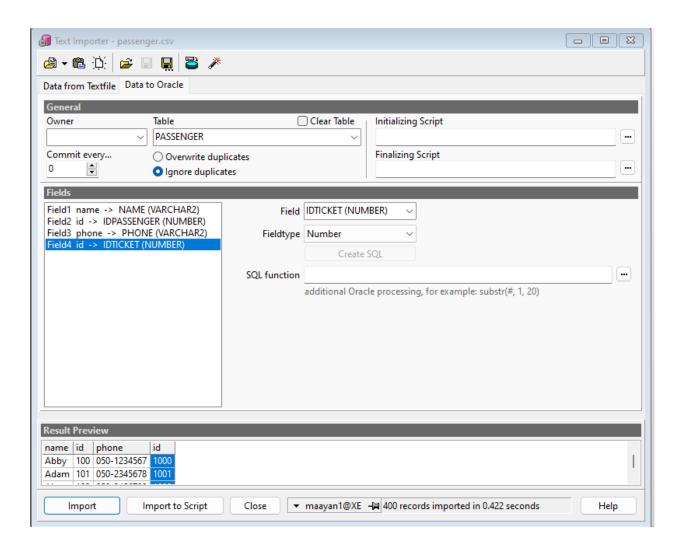
#### **Station**

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



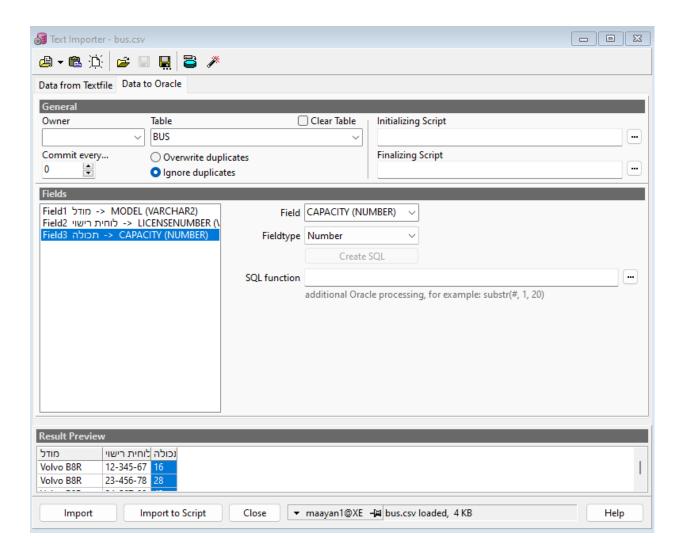
#### **Ticket**

	CARDPROFILE	IDTICKET _	ACCUMULATEDVALUE
<b>)</b> 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



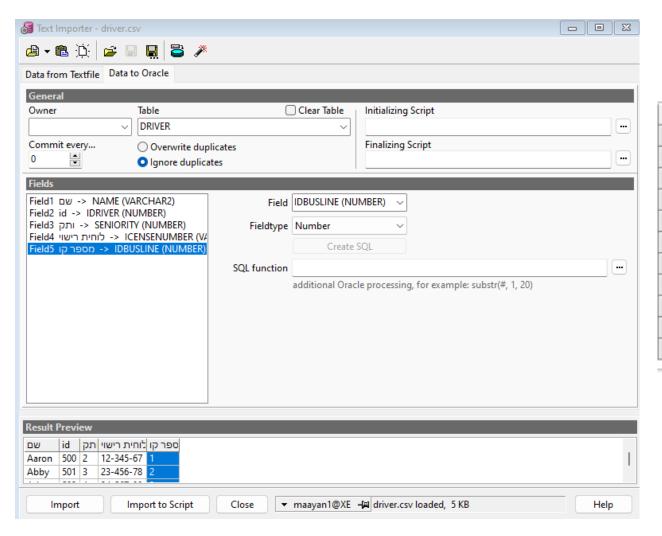
#### **Passenger**

		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



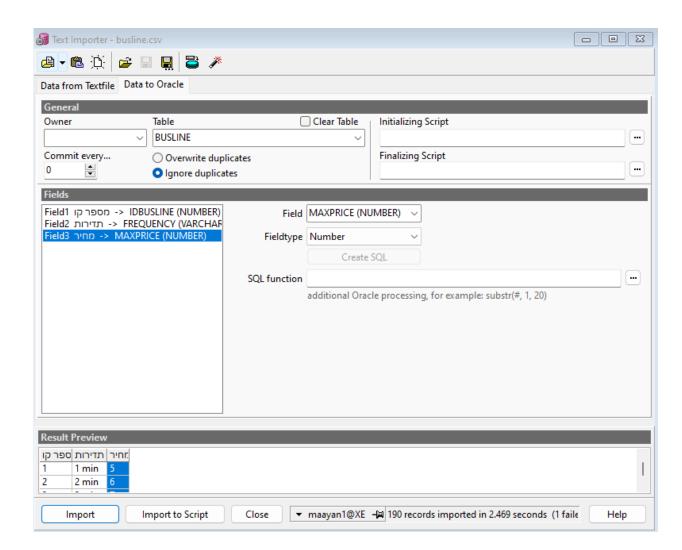
#### Bus

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



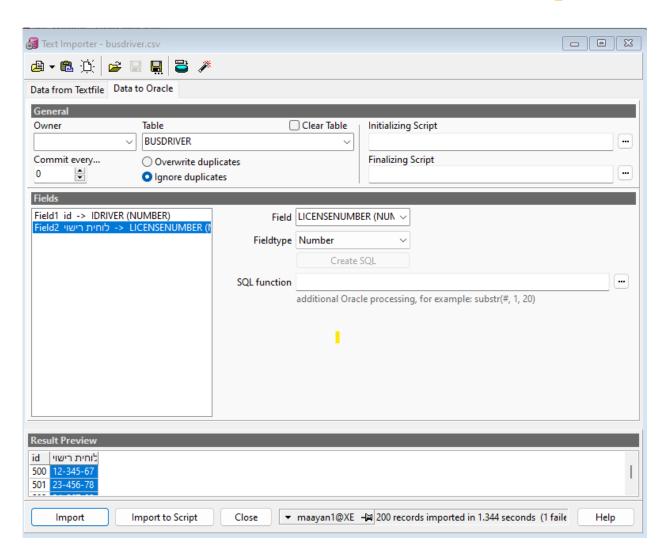
#### **Driver**

	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



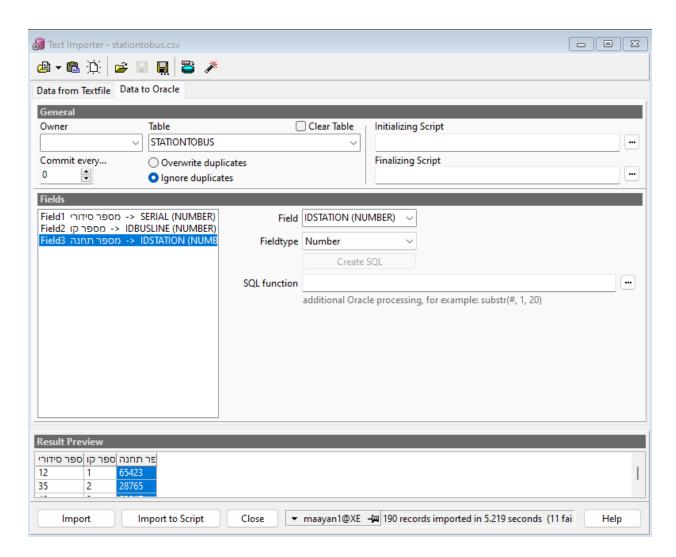
#### **Bus Line**

	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



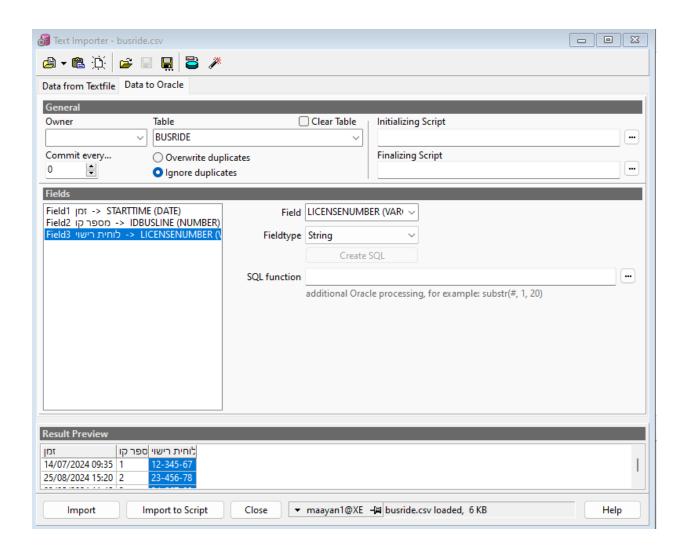
#### **Bus Driver**

	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)

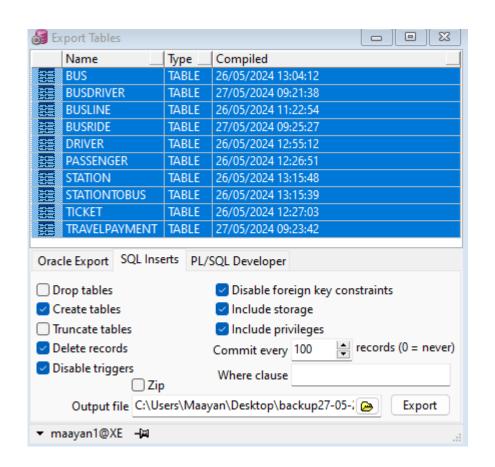
def read_file_content(file_path):
    try:
        with open(file_path, 'r') as file:
            content = file.readlines()
            return [line.strip() for line in content]
    except FileNotFoundError:
        print(f"File {file_path} not found.")
        return []
```

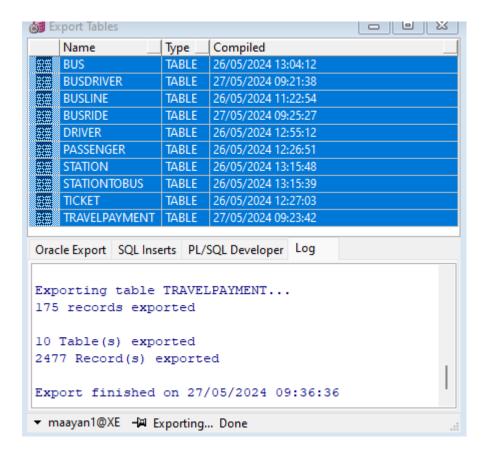
```
יצירת חיבור למסד הנתונים #
connection = cx_Oracle.connect(username, password, dsn)
cursor = connection.cursor()
# קריאת תוכן הקבצים
idtickets = read_file_content("idticket.txt")
licenses = read_file_content("license.txt")
if len(idtickets) == 0 or len(licenses) == 0:
    print("Failed to read tickets 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**

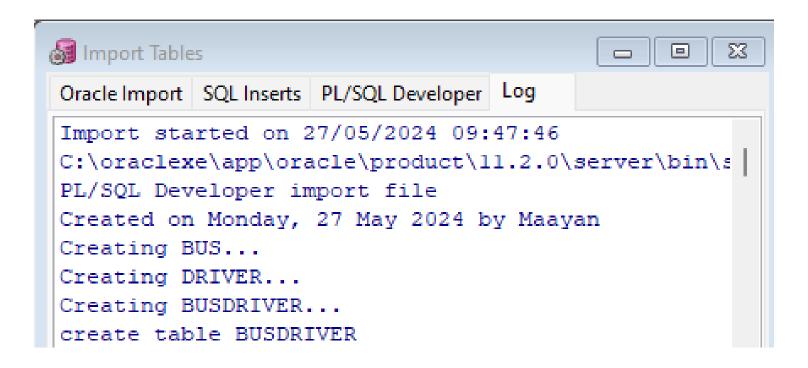
	TRAVELTIME		TRAVELCOST	IDTICKET	LICENSENUMBER
1	01/08/2024 11:30:00	•	10	1067	89-012-50
2	02/08/2024 13:40:00	•	11	1068	90-123-61
3	03/08/2024 09:15:00	•	7	1069	01-234-72
4	04/08/2024 10:45:00	•	11	1070	12-345-74
5	05/08/2024 14:25:00	•	11	1071	23-456-85
6	06/08/2024 08:50:00	•	12	1072	34-567-96
7	07/08/2024 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





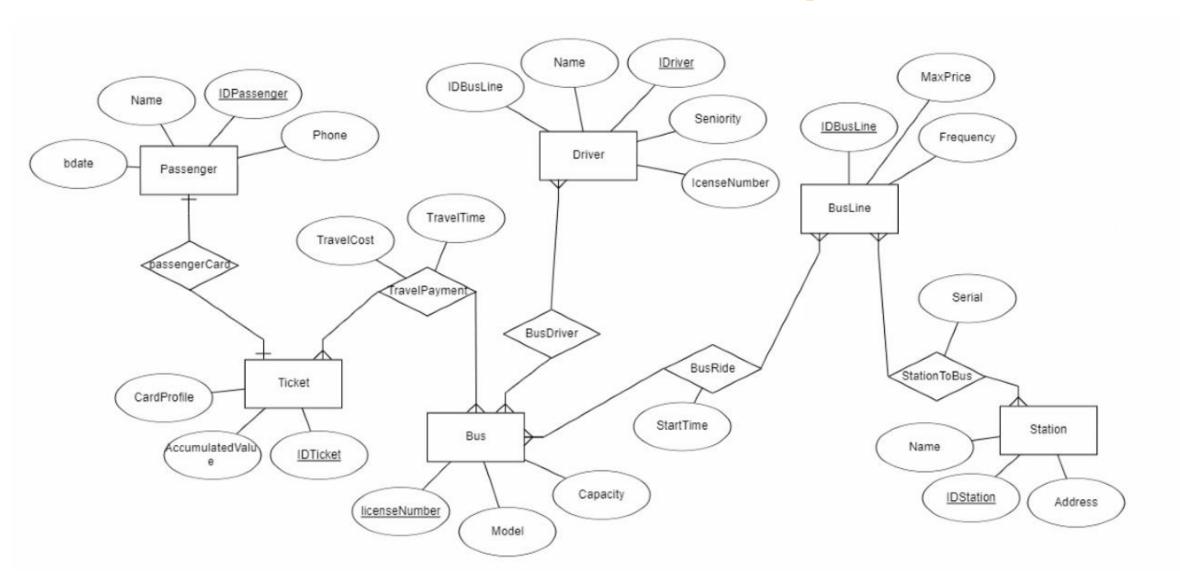


#### שחזור

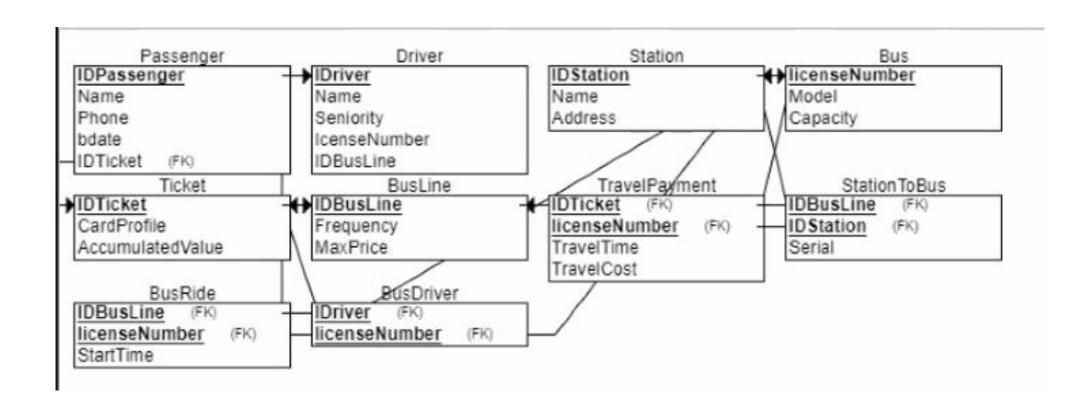


# שלב 2 שאילתות

# טבלת ERD מתוקנת

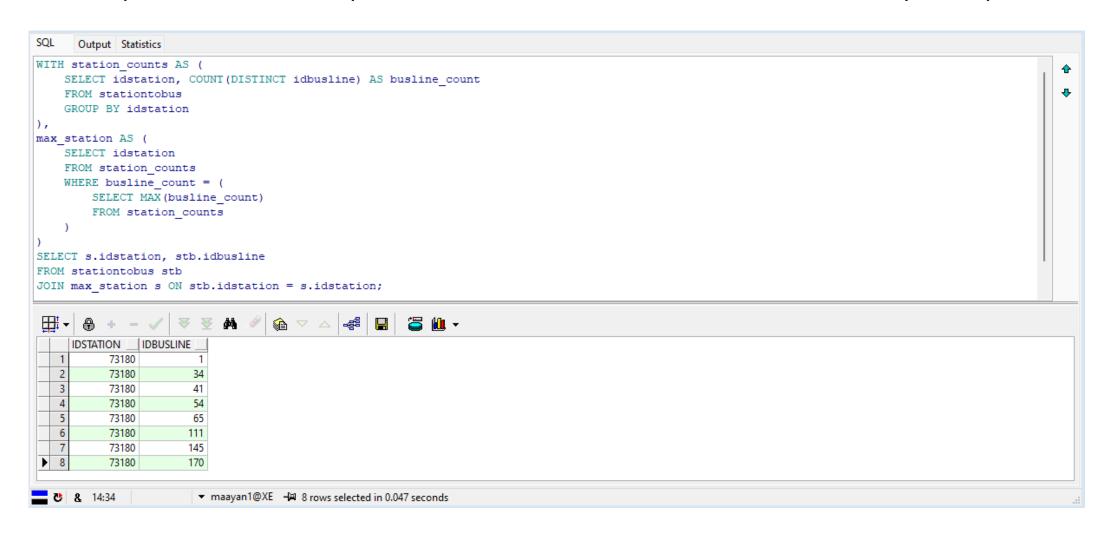


### טבלת DSD מתוקנת



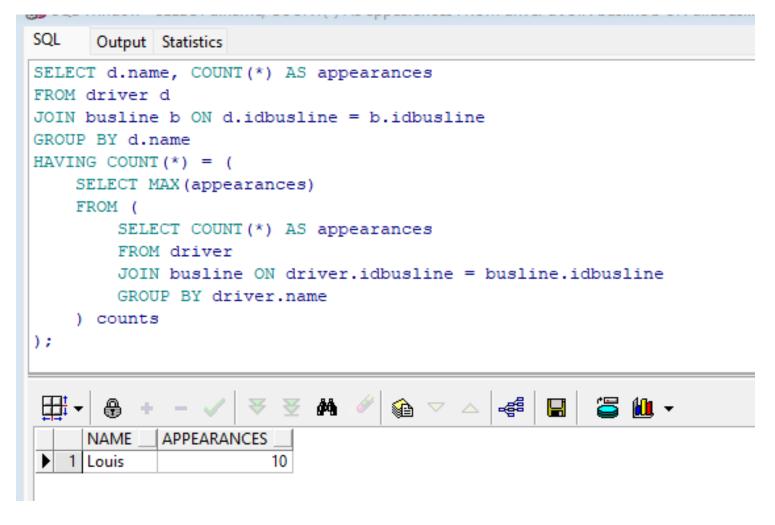
### שאילתת select1 ללא פרמטרים

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

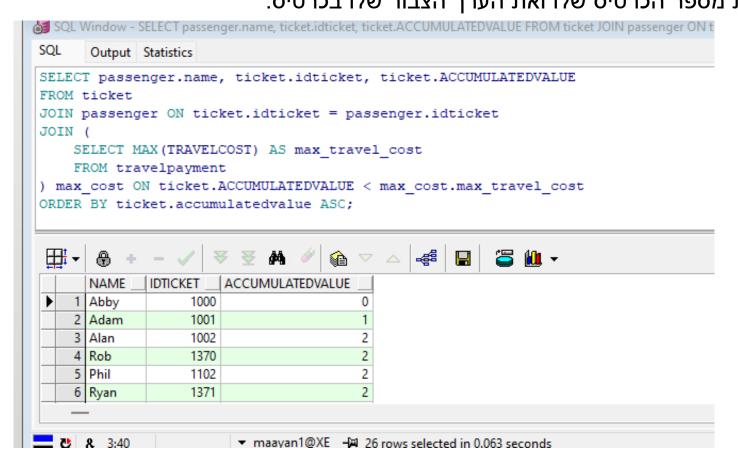
בשאילתה נרצה להציג את שם הנהג שנוהג על הכי הרבה קווים מבין כלל הנהגים בחברה. וכן הצגנו את מספר הקווים עליהם הוא נוהג.



### 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
```

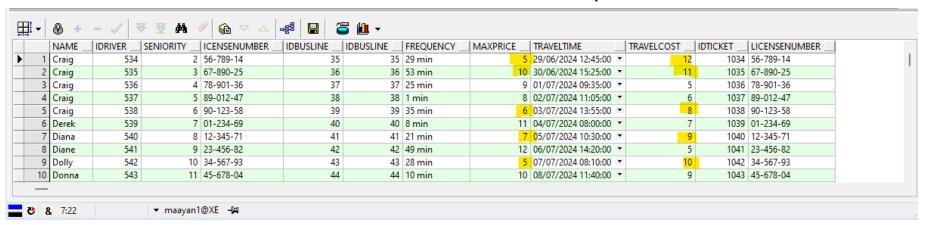
### update1 שאילתת

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
```

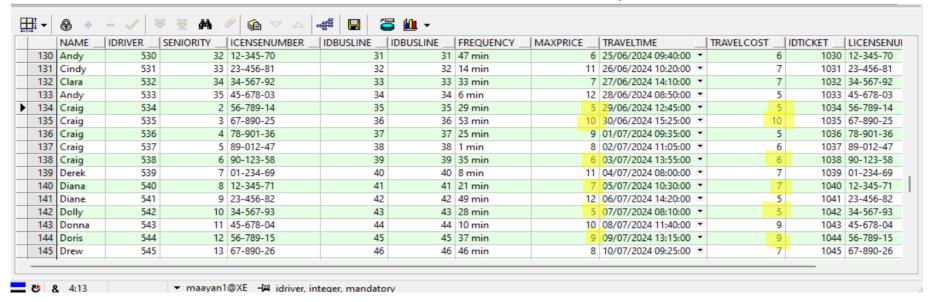
#### לפני עדכון השאילתה

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	# +	<b>a</b> +	nie.	1	*	포	44
4		BDATE		CAR	PRO	FILE	
Þ	- 1	22/03/2016	*	yout	1		
	2	14/01/2006	*	yout	1		
	3	30/06/2015	•	yout	1		
	4	01/01/2007	*	yout	1		
	5	08/11/2016	•	yout	n		
	6	18/06/2015	•	yout	n		
	7	23/12/2016	•	yout	n		
	8	20/04/2011	•	yout	1		
	9	01/06/1957		Veter	an		
	10	23/06/2006	*	yout	1		
	11	07/01/2008	*	yout	1		
	12	02/02/1958	•	Veter	an		
	13	18/03/1950	*	Veter	an		
	14	22/06/1958	*	Veter	an		
	15	28/10/1958	•	Veter	an		
	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;
```

	‡ -	<b>+</b> + -	<b>✓</b>   ₹ ₹
		IDBUSLINE	CAPACITY
<b>)</b>	1	97	10
1	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	108	54
	11	109	10
	12	109	10
	13	112	10
	14	113	15
	15	114	11
П	16	115	13
П	17	117	10
	18	118	30
	19	119	40
	20	121	10
	21	122	34
	22	123	17
	23	124	19
	24	124	19
	24	121	
	25	125	50

### 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	# +	+	- 🗸	<b>₹ ₩</b>		<b>∉</b> □ <b>=</b>
Ī		NAME	IDRIVER _	SENIORITY	ICENSENUMBER _	IDBUSLINE
T	1	Spike	306	1	58-612-35	
T	2	Sinead	346	1	49-596-70	
Ī	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

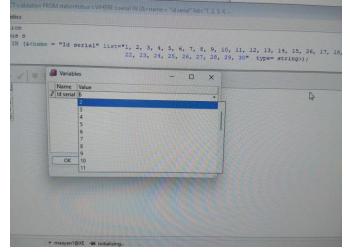
בשאילתה נרצה להציג את רשימת התחנות שמספרן הסידורי במסלול הוא הפרמטר שהוכנס, ואת התדירות של קווים העוברים בתחנה.

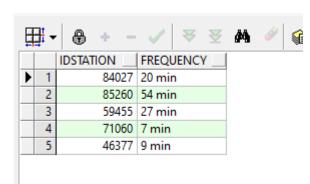
```
SELECT s.idstation, b.frequency
FROM stationtobus s
JOIN busline b ON s.idbusline = b.idbusline
WHERE s.serial IN (&<name = "Id serial" list="1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 26, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30" type= string>);
```

.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.



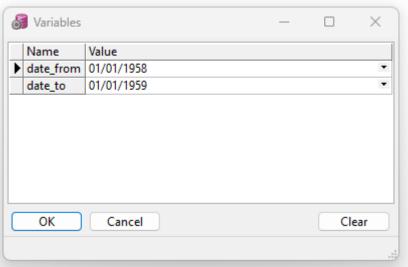


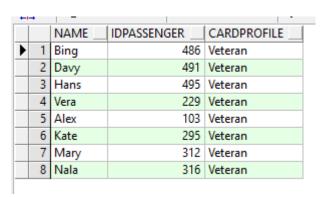
### שאילתת select2 עם פרמטרים

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.

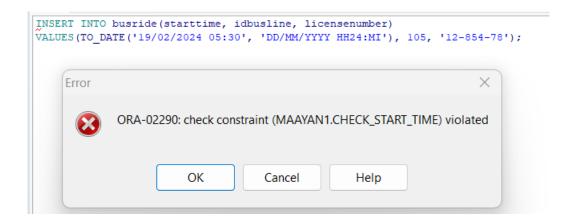




# אילוץ מספר 3

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);
```

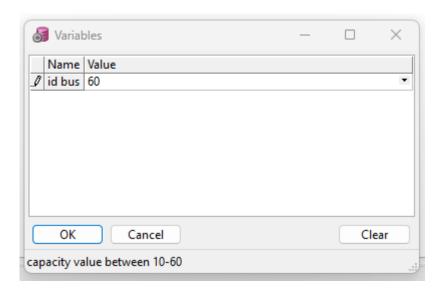


### שאילתת select3 עם פרמטרים

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	<b>-</b>	_		-
		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

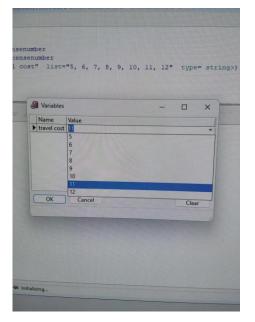
JOIN bus b ON b.licensenumber = t.licensenumber

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.

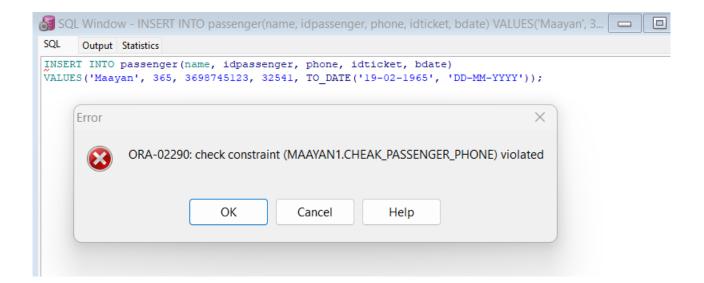


+1	<b>→</b> .	_	
		IDBUSLINE	NAME
Þ	1	66	Grace
	2	72	Helen
	3	138	Nikki
	4	144	Louis
	5	152	Quinn
	6	162	Rosa
	7	168	Sarah
	8	4	Alan

## אילוץ מספר 1

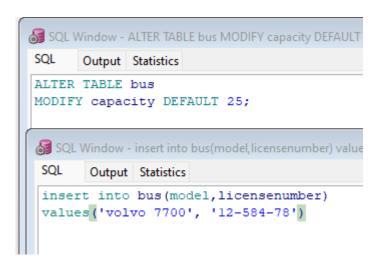
אילוץ מספר טלפון של נוסע. מספר הטלפון חייב להתחיל בתו 0.

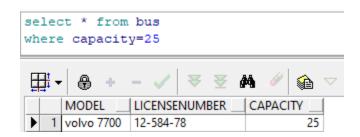
```
alter table passenger
add constraint cheak_passenger_phone
check (phone LIKE '0%');
```



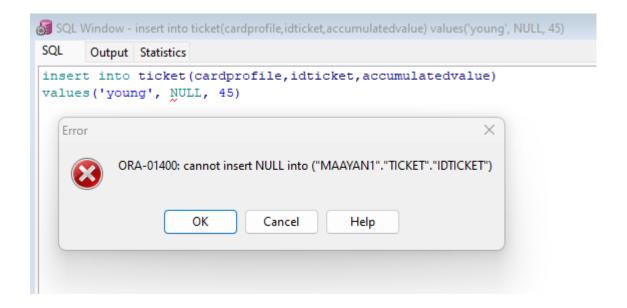
# אילוץ מספר 2

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





# ALTER TABLE ticket MODIFY idticket NOT NULL;

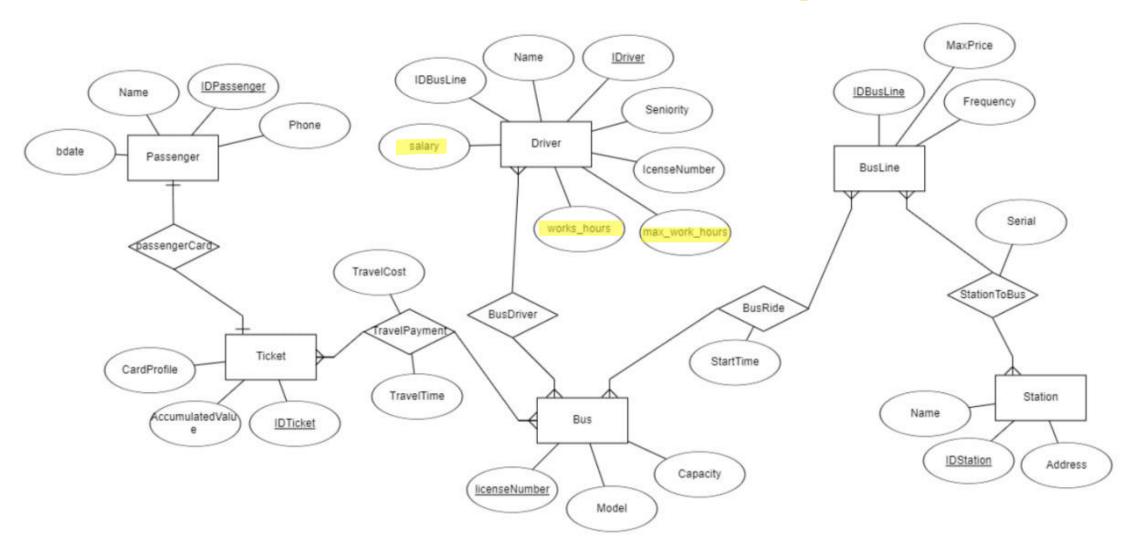


# 4 אילוץ מספר

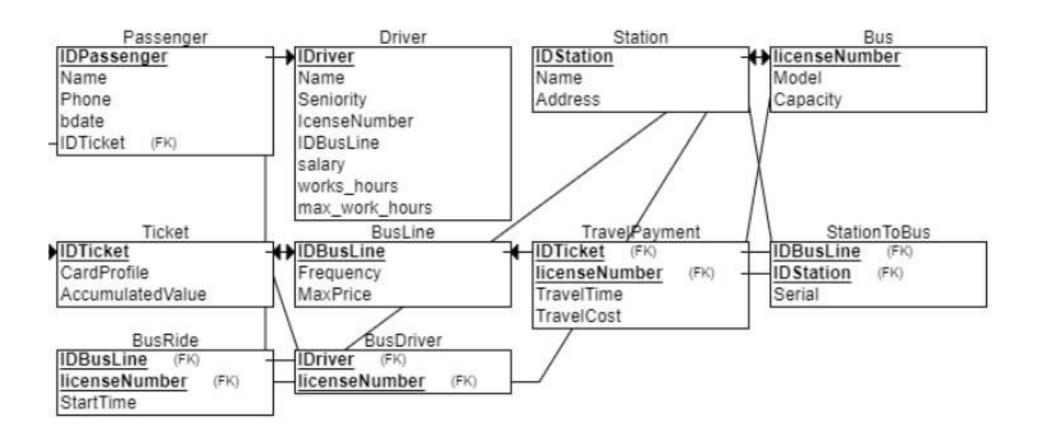
אילוץ על מספר כרטיס נסיעה.

# 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
	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
-7	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;
  illegal percent EXCEPTION;
 v new salary NUMBER;
  max work hours CONSTANT INTEGER := 182;
BEGIN
  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;
  END IF;
  IF percent <= 0 OR percent > 100 THEN
   message := 'Percent should be a number between 1 to 100';
   RAISE illegal percent;
  END IF:
 FOR driver IN c drivers LOOP
   IF driver.work hours BETWEEN 150 AND 182 THEN
     v new salary := driver.salary + ((driver.salary * percent) / 100);
     UPDATE driver
     SET salary = v new salary
     WHERE CURRENT OF c drivers;
   END IF:
 END LOOF:
 COMMIT:
```

```
OPEN result cursor FOR
  SELECT idriver, name,
         salary AS new salary,
         TRUNC(salary / (1 + (percent / 100))) AS old salary
  FROM driver
  WHERE work hours BETWEEN 150 AND 182;
RETURN result cursor;
EXCEPTION
  WHEN illegal hour THEN
   DBMS OUTPUT.PUT LINE(message);
   RETURN NULL:
 WHEN illegal percent THEN
   DBMS OUTPUT.PUT LINE(message);
   RETURN NULL:
  WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
   ROLLBACK:
   RETURN NULL;
END;
```

#### דוגמת הרצה:

עבור: precent=10 threshold hour=180

	Name	Value	
	percent_input	10	•
Þ	$threshold\_hour\_input$	180	•

```
:הפלט
Driver ID: 538, Name: Craig, Old Salary: 7982, New Salary: 8781
Driver ID: 541, Name: Diane, Old Salary: 9060, New Salary: 9967
Driver ID: 546, Name: Dolly, Old Salary: 8143, New Salary: 8958
Driver ID: 548, Name: Edith, Old Salary: 8143, New Salary: 8958
Driver ID: 552, Name: Ellie, Old Salary: 9150, New Salary: 10066
Driver ID: 553, Name: Elton, Old Salary: 6861, New Salary: 7548
Driver ID: 554, Name: Emily, Old Salary: 8193, New Salary: 9013
Driver ID: 555, Name: Emma, Old Salarv: 8201, New Salarv: 9022
Driver ID: 557, Name: Erin, Old Salary: 8946, New Salary: 9841
Driver ID: 561, Name: Frank, Old Salary: 7702, New Salary: 8473
Driver ID: 565, Name: Grace, Old Salary: 8819, New Salary: 9701
Driver ID: 568, Name: Gwen, Old Salary: 8305, New Salary: 9136
Driver ID: 571, Name: Helen, Old Salary: 7950, New Salary: 8746
Driver ID: 572, Name: Henry, Old Salary: 8002, New Salary: 8803
Driver ID: 584, Name: Jenny, Old Salary: 9627, New Salary: 10590
Driver ID: 591, Name: Joel, Old Salary: 6626, New Salary: 7289
Driver ID: 596, Name: Julia, Old Salary: 6760, New Salary: 7437
Driver ID: 597, Name: Julie, Old Salary: 8213, New Salary: 9035
```

### דוגמאות הרצה לחריגות:

#### דוגמת הרצה לאחוז שאינו בטווח של 0-100:

	Name	Value	
	percent_input	120	ı
▶	threshold_hour_input	180	

precent=120 threshold\_hour=180

Percent should be a number between 1 to 100 Error in Driver Bonus: ORA-01001: invalid cursor

הפלט:

הפלט:

#### דוגמת הרצה לסף שעות שאינו בטווח של 150-182:

precent=10

Name	Value
percent_input	10
threshold_hour_input	120

threshold\_hour=120

Threshold hour should be an integer between 150 to 182 Error in Driver Bonus: ORA-01001: invalid cursor

### פונקציה מספר 2

: Get\_Busline\_Info הפונקציה

הפונקציה מקבלת מספר קו ומחזירה עבורו את זמן המסלול שלו מתחנת המוצא ועד התחנה הסופית, ומציגה את התחנות בהן הקו עובר.

חישוב זמן המסלול: כל תחנה במסלול מוסיפה 5 דקות לזמן המסלול.

סופרים כמה תחנות יש לקו האוטובוס וכופלים ב5 כדי למצוא את זמן המסלול הכולל.

v\_duration := v\_station\_count \* 5;

```
CREATE OR REPLACE FUNCTION Get Busline Info (
   p_line_id_IN_NUMBER
) RETURN SYS REFCURSOR IS
    result cursor SYS REFCURSOR;
    v_station_count NUMBER;
    v duration NUMBER;
BEGIN
    -- Fetching the number of stations for the given bus line
    SELECT COUNT (*)
    INTO v station count
    FROM StationToBus
    WHERE IdBusLine = p line id;
    -- Calculate the duration based on the number of stations
    v duration := v station count * 5;
    -- Prepare REF CURSOR result with busline info and stations
    OPEN result cursor FOR
        SELECT stb.idbusline,
               s.idstation,
               s.name AS station name,
               v duration AS total duration
        FROM stationtobus stb
        JOIN station s ON stb.idstation = s.idstation
        WHERE stb.idbusline = p line id
        ORDER BY s.idstation;
```

#### פונקציה מספר 2

```
RETURN result_cursor;

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('No data found for the given bus line.');

RETURN NULL;

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);

RETURN NULL;

END;
```

#### דוגמת הרצה:

:הפלט

עבור קו האוטובוס שמספרו: 65

	Name	Value
Þ	p_line_id_input	65

```
Total Duration: 55 minutes
Stations:
                                                                       1. משך זמן המסלול
Station ID: 19299, Station Name: 419 Utsunomiya
                                                                   2. התחנות בהן הקו עובר
Station ID: 23866, Station Name: 46 Ernie Street
Station ID: 29757, Station Name: 75 Kane
Station ID: 37296, Station Name: 39 Eindhoven Dr
Station ID: 39679, Station Name: 96 Curfman Ave
Station ID: 47843, Station Name: 94 Natacha Driv
Station ID: 50322, Station Name: 6 Graham Street
Station ID: 65642, Station Name: 80 Shannon Blvd
Station ID: 73180, Station Name: 39 Mifune Road
Station ID: 79822, Station Name: 58 Bosco Street
Station ID: 85260, Station Name: 44 Beaverton St
```

### 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.

```
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
    -- Calculate the age of the passenger
   v age := FLOOR(MONTHS BETWEEN(SYSDATE, p passenger birth date) / 12);
    -- Determine the profile based on age
   IF v age BETWEEN 5 AND 18 THEN
                                                                                 COMMIT;
        v profile := 'Student';
    ELSIF v age BETWEEN 19 AND 29 THEN
                                                                             EXCEPTION
       v profile := 'Youth';
    ELSIF v age BETWEEN 30 AND 65 THEN
       v profile := 'Adult';
    ELSIF v age > 65 THEN
                                                                             END;
       v profile := 'Senior';
    ELSE
       v profile := 'Invalid'; -- In case the age doesn't fall into any range
    END IF:
    -- Find the highest ticket number and increment by one
    SELECT NVL(MAX(idticket), 0) + 1 INTO v ticket number FROM ticket;
    -- Insert the ticket into the tickets table
    INSERT INTO ticket (cardprofile, idticket, accumulatedvalue)
   VALUES (v profile, v ticket number, p stored value);
    -- Insert the passenger into the passengers table
    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);

#### פרוצידורה מספר 1

```
-- Commit the transaction
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.

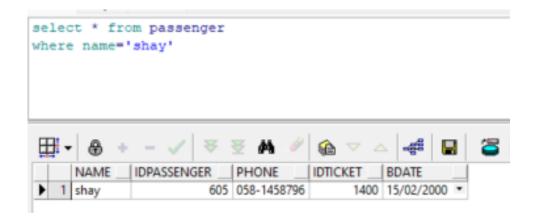
```
begin
         -- Call the procedure
        add passenger and ticket(p passenger id => :p passenger id,
                                   p passenger name => :p passenger name,
                                   p passenger phone => :p passenger phone,
                                   p_passenger_birth_date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

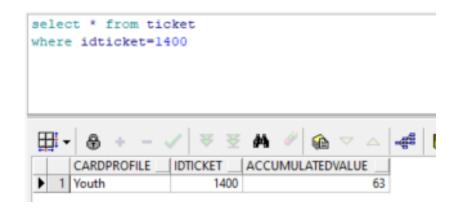
    □ p_passenger_name

                      String
                                     shay
 p_passenger_phone
                      String
                                     058-1458796
  p_passenger_birth_date
                      Date

    15/02/2000

▶ ▼ p_stored_value
                                    ▼ 63
```





### פרוצידורה מספר 2

:Get\_BusLine\_DepartureTimes הפונקציה

מקבלת קו אוטובוס, ומחזירה את התדירות של הקו ואת כל שעות היציאה שלו מתחנת המוצא. חישוב שעות היציאה:

כל קו אוטובוס פועל בין השעות 06:00-00:00

ולכן בין השעות האלה הקו יכול לצאת למסלול חדש.

הפונקציה תחזיר את כל שעות היציאה בהתאם לתדירות,

:דוגמא

.06:00 וכו' כשאר התדירות של הקו היא 42 דק.

#### פרוצידורה מספר 2

```
CREATE OR REPLACE PROCEDURE GetBusLineDepartureTimes (
    p line id IN NUMBER
) IS
    v frequency string VARCHAR2(20);
    v frequency NUMBER;
    v start time DATE := TRUNC(SYSDATE) + INTERVAL '6' HOUR; -- Start time at 06:00 AM
    v end time DATE := TRUNC(SYSDATE) + INTERVAL '24' HOUR - INTERVAL '1' SECOND; -- End time at 11:59:59 PM
    v departure time DATE;
    CURSOR c departures IS
        SELECT BusLine.Frequency
        FROM BusLine
        WHERE Idbusline = p line id;
BEGIN
    -- Fetching frequency using explicit cursor
    OPEN c departures;
    FETCH c departures INTO v_frequency_string;
    CLOSE c departures;
    IF v frequency string IS NULL THEN
        RAISE APPLICATION ERROR (-20001, 'Invalid line id or no data found.');
    END IF:
    -- Extracting frequency as number from the string
    v frequency := TO NUMBER(REGEXP SUBSTR(v frequency string, '^\d+'));
    -- Debugging output
    DBMS OUTPUT.PUT LINE('the frequency: ' || v frequency string);
     -- Loop to print departure times
    v departure time := v start time;
    WHILE v_departure_time <= v_end_time LOOP
        DBMS OUTPUT.PUT LINE(TO CHAR(v departure time, 'HH24:MI'));
        v_departure_time := v_departure_time + NUMTODSINTERVAL(v_frequency, 'MINUTE');
    END LOOP:
```

EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
RAISE;
END:

### דוגמת הרצה:

עבור קו האוטובוס שמספרו: 65



23:30

the frequency: 42 min				
06:00			:ს	הפז
06:42			L	4
07:24		הקו.	התדירות של	.1
08:06	התאם לתדירות הקו.	ועב ועלו בי	בל שוחם בוע	2
08:48	ווונאם זונו יו וונ וואוו.	אוו פיו בו	כז שעוונ וויצ	
09:30				
10:12				
10:54				
11:36				
12:18				
13:00				
13:42				
14:24				
15:06				
15:48				
16:30				
17:12				
17:54				
18:36				
19:18				
20:00				
20:42				
21:24				
22:06				
22:48				

```
DECLARE
```

#### Main 1

```
תוכנית ראשית עבור:
```

```
percent NUMBER := &percent input;
threshold hour NUMBER := &threshold_hour_input;
p passenger id NUMBER := &p passenger id input;
p passenger name VARCHAR2(100) := '&p passenger name input';
p_passenger_phone VARCHAR2(15) := '&p_passenger_phone_input';
p_passenger_birth_date DATE := TO_DATE('&p_passenger_birth_date_input', 'DD-MM-YYYY');
p_stored_value NUMBER := &p_stored_value_input;
result bonus SYS REFCURSOR;
idriver NUMBER;
name VARCHAR2(100);
new salary NUMBER;
old salary NUMBER;
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;
```

```
add_passenger_and_ticket(
   p_passenger_id,
   p_passenger_name,
   p_passenger_phone,
   p_passenger_birth_date,
   p_stored_value
);
DBMS_OUTPUT.PUT_LINE('Passenger and ticket added successfully.');

EXCEPTION
   WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);

END;
//
```

```
DECLARE
    p line id NUMBER := &p line id input;
    result cursor SYS REFCURSOR;
    v idbusline NUMBER;
    v idstation NUMBER;
    v station name VARCHAR2(100);
    v total duration NUMBER;
BEGIN
    -- קריאה לפונקעיה Get Busline Info
    result cursor := Get Busline Info(p line id);
    -- קר את השורה הראשונה מה --cursor
    FETCH result_cursor INTO v_idbusline, v_idstation, v_station_name, v_total_duration;
    בדיקה זם לא נמצאו נתונים --
    IF result cursor%NOTFOUND THEN
        DBMS OUTPUT.PUT_LINE('No data found for the given bus line.');
    ELSE
        הדפסת מן המסלול --
        DBMS_OUTPUT.PUT_LINE('Total Duration: ' || v total_duration || ' minutes');
       המשך ל לאה להדפסת שאר התחנות --
       LOOP
           FETCH result cursor INTO v idbusline, v idstation, v station name, v total duration;
           EXIT WHEN result cursor%NOTFOUND;
           DBMS OUTPUT.PUT LINE('Bus Line ID: ' || v idbusline ||
                                ', Station ID: ' || v_idstation ||
                                ', Station Name: ' || v station name);
       END LOOF;
   END IF;
   CLOSE result cursor;
   -- קריאה לפרועדורה GetBusLineDepartureTimes
   DBMS_OUTPUT.PUT_LINE('Departure Times for Bus Line ID: ' || p_line_id);
   GetBusLineDepartureTimes(p line id);
EXCEPTION
   WHEN OTHERS THEN
       DBMS OUTPUT.PUT LINE('An error occurred: ' || SQLERRM);
END;
```

#### Main 2

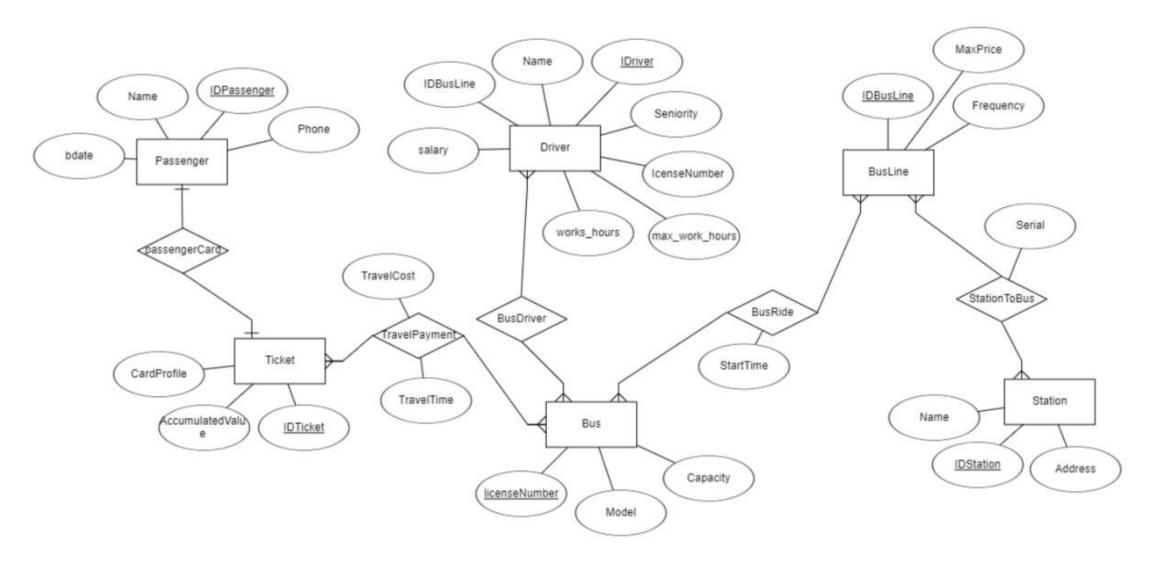
תוכנית ראשית עבור:

GetBusLineDepartureTimes

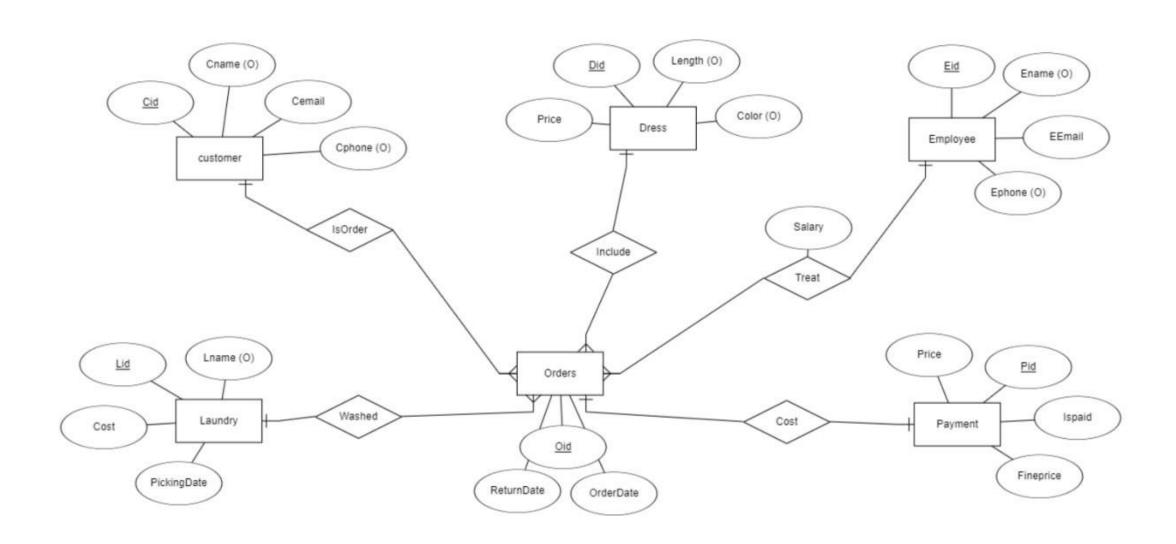
Get Busline Info

# שלב 4 אנטגרציה ומבטים

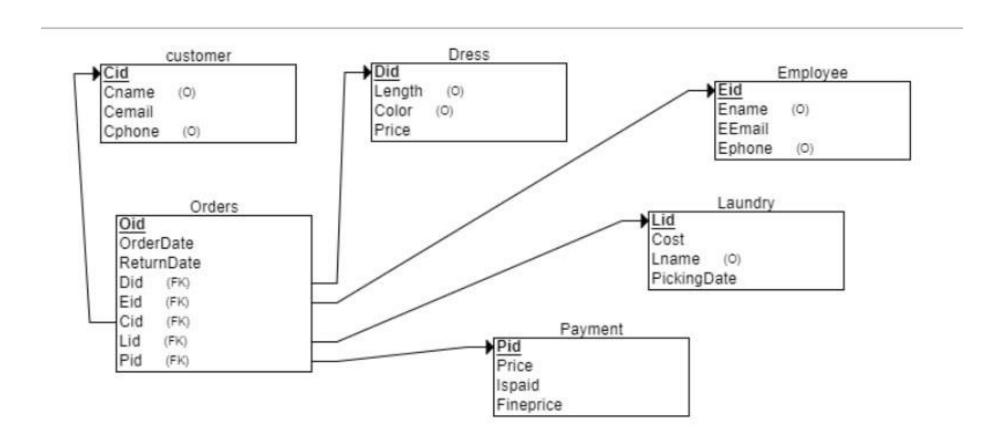
## טבלת ERD של הפרויקט שלנו



### טבלת ERD של הפרויקט החדש איתו אנחנו מתמזגות



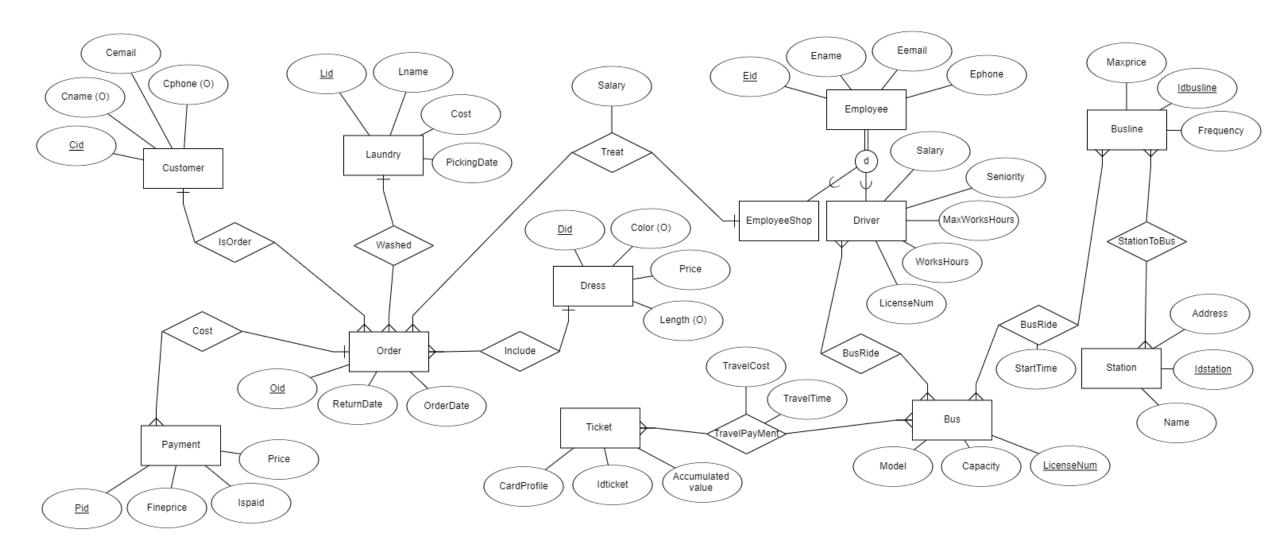
## טבלת DSD של הפרויקט החדש איתו אנחנו מתמזגות



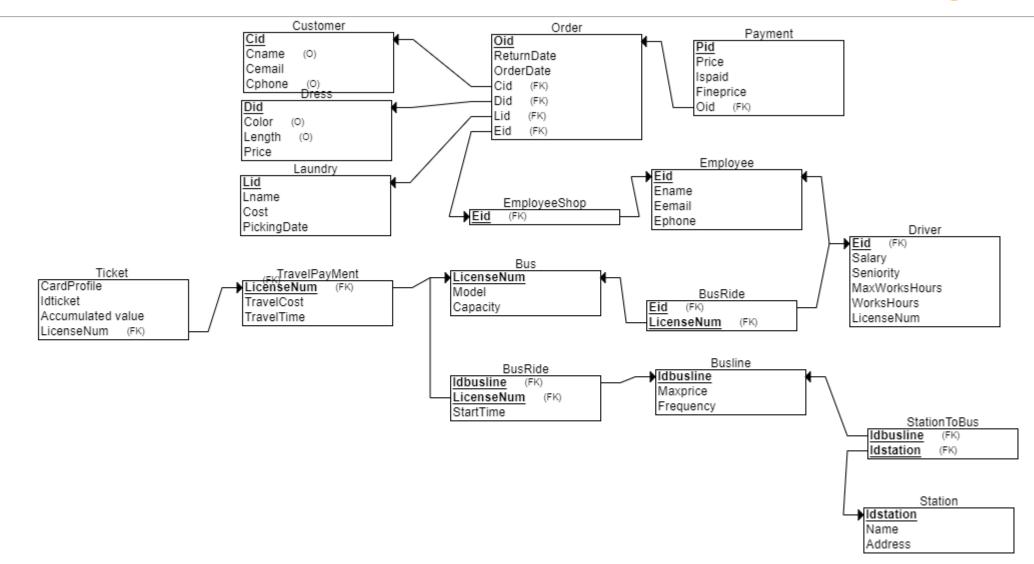
#### החלטות:

- 1. הemployee של הפרויקט האחר הפך להיות ישות ממנה יורשים 2 סוגים של עובדים: driver - נהג שליחויות לפיזור ההזמנה. ו employeeShop - עובד בחנות אשר אחראי לקבל את ההזמנות ולבצע אותן.
- 2. הpassenger בפרויקט שלנו הפך להיות customer אשר מזמין את השמלה. נשאיר לו את אותם תכונות חוץ מהשדה bdate
  - .empolyeeShop ובין orders .3
    - ticket passenger לticket. מחיקת הקשר בין
      - order לemployee מחיקת הקשר בין

## הERD המשולב



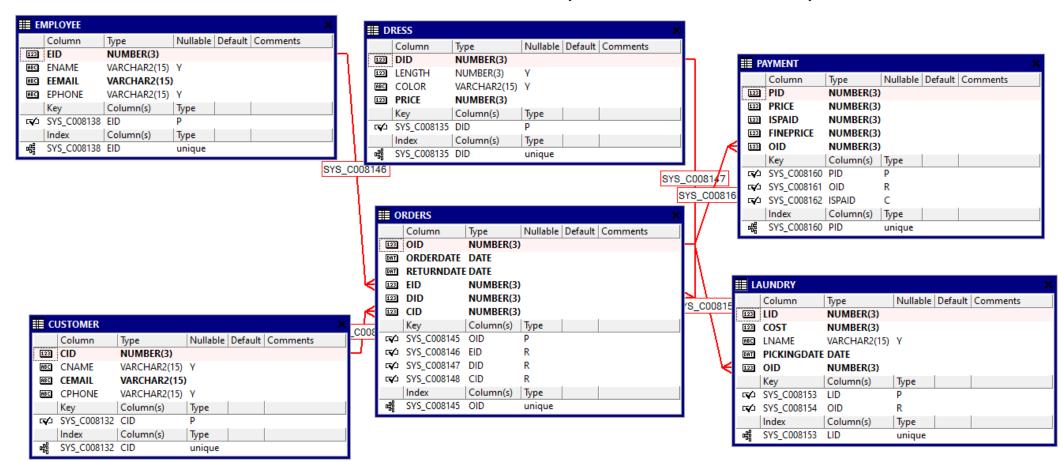
## הDSD המשולב



## שחזור הנתונים של הפרויקט האחר ע"י גובץ הגיבוי

הרצנו את קובץ הגיבוי, והגיבוי עבר בהצלחה.

ניתן לראות כי כל הטבלאות והקשרים החדשים של הפרויקט האחר נוספו.



## passengera עם טבלת customera איחוד טבלת

בטבלת הלקוח יש לנו 4 שדות אותן נרצה להשאיר ללא שינוי.

נאחד את טבלת הנוסע עם הלקוח ע"י כך שנעביר את הנתונים של הנוסעים לטבלת הלקוח. השדה היחידי שלא משותף בין 2 הטבלאות הוא השדה של המייל (Cemail) ואותו נמלא לבד.

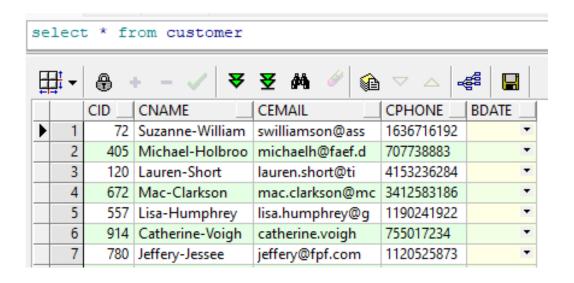
ALTER TABLE customer ADD bdate Date

נוסיף לטבלת הלקוח את השדות שיש בטבלת הנוסעים ואין בטבלת הלקוח

ALTER TABLE customer
MODIFY bdate Date NULL;

נאפשר שהשדה bdateיוכל להיות עם ערך null:

ונקבל: טבלה של לקוח שמכיל 5 שדות שהיו גם בלקוח וגם בנוסע.



#### passengera עם טבלת customera איחוד טבלת

נרצה לאחד את הנתונים שיש בטבלת הנוסעים יחד עם הנתונים שיש בטבלת הלקוחות

```
select * from passenger where passenger.idpassenger not in (select cid from customer ) :אחר ביצוע הפקודה
```

ראינו כי יש 280 נתונים שונים בין הטבלאות. נמזג את 280 השורות הנ"ל:

ונראה כי טבלת הלקוחות התרחבה מ326 ערכים ל606 ערכים:

→ 606 rows selected in 0.938 seconds

## passengera עם טבלת customera איחוד טבלת

	CID	CNAME	CEMAIL	CPHONE	BDATE			
584	143	Glen	G@example.com	055-4567890	•			
585	150	Jack	J@example.com	056-1234567	•			
586	153	Jane	J@example.com	056-4567890	•			
587	157	Jeff	J@example.com	056-8901234	▼			
588	160	Jill	J@example.com	057-1234567	•			
589	164	John	J@example.com	057-5678901	▼			
590	174	Leah	L@example.com	058-5678901	•			
591	180	Lori	L@example.com	059-1234567	▼			
592	185	Mace	M@example.com	059-6789012	•			
593	199	Owen	O@example.com	072-0123456	•			
594	214	Sam	S@example.com	074-5678901	•			
595	228	Ту	T@example.com	076-9012345	•			
596	241	Bing	B@example.com	078-2345678	•			
597	285	Jame	J@example.com	082-6789012	▼			
598	305	Liza	L@example.com	084-6789012	▼			
599	326	Paul	P@example.com	086-7890123	▼			
600	125	Dana	D@example.com	053-6789012	•			
601	166	Juan	J@example.com	057-7890123	•			
602	605	shay	s@example.com	058-1458796	▼			
603	606	yosi	y@example.com	054-8596487	▼			
604	903	maayan	m@example.com	0512214074	•			
605	904	maayan	m@example.com	0512214074	▼			
606	906	naama	n@example.com	0518745698	•			

סך הכל קיבלנו ישות אחת בשם customer עם 606 ערכים משותפים גם מטבלת הנוסעים וגם מטבלת הלקוחות.

## employee לבין driver, employeeShop הוספת הורשה בין

ניצור ישות חדשה בשם employeeShop שהיא תשמור לנו את כל העובדים בחנות. הישות יורשת מהישות employee את כל השדות.

```
CREATE TABLE EmployeeShop (
    EmployeeID NUMBER PRIMARY KEY,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(Eid)
);

INSERT INTO EmployeeShop (EmployeeID)
SELECT Eid
FROM Employee;
```

וקיבלנו: נשמור בטבלה רק את הid של העובדים

+1	<del></del>	_
		EMPLOYEEID
Þ	1	101
	2	102
	3	104
	4	106
	5	109
	6	110

## employee לבין driver, employeeShop לבין

ניקח ישות קיימת בשם driver שהיא תשמור לנו את כל הנהגים שאחראים לפזר את המשלוחים. הישות יורשת מהישות employee את כל השדות, ונוסיף על זה את השדות של הdriver.

```
ווא הכנסת הנתונים:

SELECT

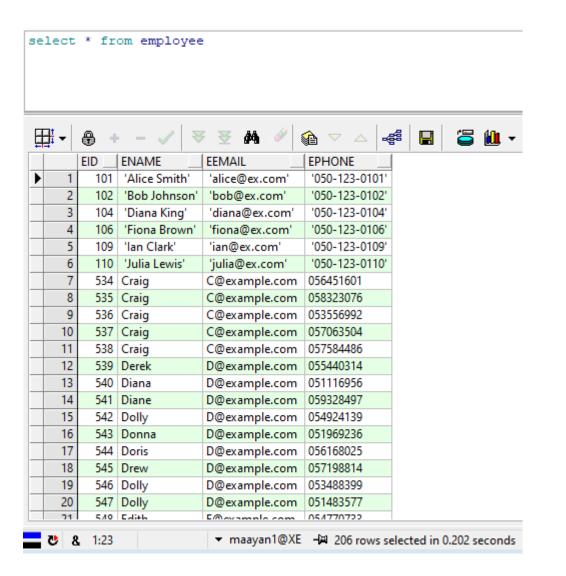
d.idriver,
d.name,
SUBSTR(d.name, 1, 1) || '@example.com' AS eemail,
'05' || TRIM(TO_CHAR(DBMS_RANDOM.VALUE(1000000, 9999999), '9999999')) AS ephone

FROM
driver d

WHERE
d.idriver NOT IN (SELECT e.eid FROM employee e);
```

מתוך טבלת הdriver driver מתוך טבלת הname; :employee שזה הרי קיים אצל הame מתוך טבלת ה

## employee לבין driver, employeeShop הוספת הורשה בין



וקיבלנו:

הוספנו לטבלת העובדים גם את הנהגים שלנו.

## מחיקת הקשר בין ה employee ל

נמחוק את הקשר של היחיד לרבים שקיים בין הטבלאות.

ALTER TABLE orders DROP CONSTRAINT SYS\_C008146;

לאחר מחיקת הקשר נראה כי אחר הוא נמחק:

```
ORDERS

ORDERS

Columns

Primary key

Unique keys

Foreign keys

SYS_C008148 (CUSTOMER)

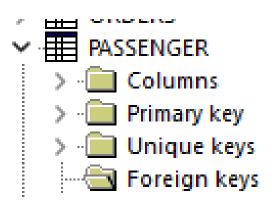
SYS_C008147 (DRESS)
```

## ticket passenger מחיקת הקשר בין

נמחוק את הקשר של היחיד ליחיד שקיים בין הטבלאות.

ALTER TABLE passenger DROP CONSTRAINT sys\_c007928;

לאחר מחיקת הקשר נראה כי אחר הוא נמחק:



## employeeShopל orders הוספת קשר בין

```
orders לטבלה employeeld נוסיף קודם עמודה
ALTER TABLE orders
ADD (employeeid NUMBER);
                                            נוסיף את הקשר בין הid של הטבלאות:
ALTER TABLE orders
ADD CONSTRAINT fk employee shop
FOREIGN KEY (Employeeid)
REFERENCES employeeShop (Employeeid);
                                                            ונקבל קשר חדש:

✓ ■ ORDERS

  > · i Columns
  > · Primary key
  > • Unique keys
```

> √√ SYS\_C008148 (CUSTOMER)

> TV FK\_EMPLOYEE\_SHOP (EMPLOYEESHOP)

> TV SYS\_C008147 (DRESS)

# Views

נעשה view ראשון על הפרויקט של השכרת השמלות.

ה- View שנוצר בשם CustomerPaymentDetails נועד להציג פרטים חשובים על לקוחות והזמנות בחנות. הוא מספק תצוגה משולבת של נתונים מטבלאות הלקוחות, ההזמנות והתשלומים. ה- View נועד להציג מידע מקיף על לקוחות ושמלות שהושכרו, כולל תשלום ותאריך החזרה.

## SELECT \* FROM CustomerPaymentDetails; :viewa הפעלת ה

הצגת הפלט:

#### SELECT \* FROM CustomerPaymentDetails;

<u> </u>	<b>□</b> + - ✓   ₹ ₹ <b>M</b> Ø   <b>6</b> ▼ △   <b>4</b> □   <b>3 6</b> ▼									
		CID	CNAME	CPHONE	CEMAIL	RENTAL_PAYMENT	RETURNDATE	DRESS		
Þ	1	405	Michael-Holbroo	707738883	michaelh@faef.d	997	11/09/2022 *	855		
	2	557	Lisa-Humphrey	1190241922	lisa.humphrey@g	771	15/03/2024 *	596		
	3	780	Jeffery-Jessee	1120525873	jeffery@fpf.com	375	23/06/2022 *	525		
	4	374	Albertina-Roth	3814200202	albertina.roth@	904	01/01/2024 *	459		
	5	374	Albertina-Roth	3814200202	albertina.roth@	559	01/01/2024	459		
	6	828	Marley-Armatrad	631747546	marley.armatrad	756	05/10/2022 *	847		
	7	486	Darius-Hector	4048366738	darius.hector@w	453	03/07/2022	501		
	8	486	Darius-Hector	4048366738	darius.hector@w	772	03/07/2022 *	501		
	9	486	Darius-Hector	4048366738	darius.hector@w	201	03/07/2022 *	501		
	10	486	Darius-Hector	4048366738	darius.hector@w	362	17/10/2022 *	125		

#### שאילתה 1: מציאת סכום התשלומים הכולל של לקוח ספציפי

```
SELECT
cid,
cname,
SUM(rental_payment) AS total_rental_payment
FROM
CustomerPaymentDetails
WHERE
cid =405
GROUP BY
cid, cname;
```

#### פלט השאילתה:

		1	l I	I I
		CID	CNAME	TOTAL_RENTAL_PAYMENT
H	1	405	Michael-Holbroo	997

#### שאילתה 2: ניתוח הלקוחות והשמלות שהושכרו בטווח זמן מסוים

```
SELECT
    cname,
    rental_payment,
    returndate
FROM
    CustomerPaymentDetails
WHERE
    returndate BETWEEN TO_DATE('2022-07-01', 'YYYY-MM-DD') AND TO_DATE('2023-07-31', 'YYYY-MM-DD');
```

	CNAME	RENIAL_PAYMENT	KETUKNDATE _	_
- 1	Dar-Mohr	220	21/10/2022	•
2	Jean-Claude-McC	709	23/09/2022	•
3	Lionel-Union	847	19/05/2023	•
4	Dylan-Favreau	706	03/10/2022	•
5	Rhona-Perry	590	10/08/2022	•
6	Geoffrey-Playbo	849	26/06/2023	•
7	Lili-Singh	683	07/02/2023	•
8	Tommy-Aaron	707	08/03/2023	•
9	Thomas-Eckhart	820	05/07/2023	•
10	Debi-Borden	114	16/07/2022	•
11	Dionne-Lachey	771	22/05/2023	•
12	Fionnula-Sartai	545	07/02/2023	•
13	Holland-Bright	483	09/10/2022	•
14	lan-Sepulveda	312	28/12/2022	•

CNIANE DENTAL DAVINENT DETURNIDATE

:פלט השאילתה

#### views

נעשה view שני על הפרויקט של חברת האוטובוסים.

ה- View בשם TicketPassengerDetails נוצר כדי להציג מידע משולב על נוסעים, כרטיסים, ועלויות נסיעה בצורה נוחה לשימוש. הוא מחבר נתונים משלוש טבלאות: ticket, passenger ו-travelpayment

```
CREATE OR REPLACE VIEW TicketPassengerDetails AS

SELECT

p.name AS passenger_name,
t.idticket AS ticket_id,
t.accumulatedvalue AS accumulated_value,
tp.travelcost AS travel_cost,
(SELECT MAX(travelcost) FROM travelpayment) AS max_travel_cost

FROM
ticket t

JOIN
passenger p ON t.idticket = p.idticket

JOIN
travelpayment tp ON tp.idticket = t.idticket;
```

SELECT \* FROM TicketPassengerDetails;

:viewa הפעלת

הצגת הפלט:

	PASSENGER_NAME	TICKET_ID	ACCUMULATED_VALUE	TRAVEL_COST	MAX_TRAVEL_COST
1	Abby	1000	0	7	12
2	Alex	1003	3	11	12
3	Luke	1084	84	9	12
4	Mara	1086	86	6	12
5	Mark	1087	87	8	12
6	Mary	1088	88	7	12
7	Matt	1089	89	9	12
8	Max	1090	90	5	12
9	Maya	1091	91	9	12
10	Meg	1092	92	9	12
11	Mike	1093	93	8	12
12	Nate	1094	94	6	12
13	Neal	1095	95	6	12
14	Nick	1096	96	6	12

#### שאילתה 1: מציאת 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 הנוסע, את מספר הכרטיס שלו ואת הערך הצבור שלו בכרטיס.

```
SELECT

passenger_name,

ticket_id,

accumulated_value

FROM

TicketPassengerDetails

WHERE

accumulated_value < max_travel_cost

ORDER BY

accumulated_value ASC;
```

פלט השאילתה:

	NAME	IDTICKET	ACCUMULATEDVALUE	Ш
_ 1	Abby	1000		0
2	Adam	1001		1
3	Alan	1002		2
4	Rob	1370		2
5	Phil	1102		2
6	Ryan	1371		2
7	Quinn	1367		2
8	Raul	1104		3
9	Sam	1114		3

#### שאילתה 2: מציאת נוסעים עם כרטיסים שפג תוקפם

נרצה לראות אילו כרטיסים הם בעלי ערך מצטבר של 0, כלומר, נוסעים שלא יכולים להשתמש בכרטיסיהם כרגע.

```
SELECT

passenger_name,

ticket_id,

accumulated_value,

travel_cost

FROM

TicketPassengerDetails

WHERE

accumulated_value = 0

ORDER BY

travel_cost DESC;
```

#### :פלט השאילתה

		1		I .		I I	
		PASSENGER_NAME	TICKET_ID	ACCUMULATED_	VALUE _	_ TRAVEL_	COST
	1	Abby	1000		0	)	7