

# מיני פרוייקט בבסיסי נתונים

## חברת הסעות



מגישים: אריאל זקן ואבישי גונן

## תוכן העניינים

4	ישויות נתונים וקשרים
5	פונקציונליות
6	תרשים ERD
6	תרשים DSD
7	יצירת הטבלאות
7	יצירת טבלת Bus
7	יצירת טבלת Line
7	יצירת טבלת Driver
8	יצירת טבלת Schedule
8	יצירת טבלת Station
8	יצירת טבלת Taxi
9	יצירת טבלת BusRide
9	יצירת טבלת DriveInTaxi
10	שיטות למילוי הטבלאות
10	אנחנו נמלא את הנתונים ב 3 דרכים שונות:
10	שימוש ב mockaroo
10	הכנסה לטבלה Bus
10	הכנסה לטבלה Driver
10	הכנסה לטבלה Taxi
11	הכנסה לטבלה Schedule
11	הכנסה לטבלה Line
11	הכנסה לטבלה DrivesInTaxi
12	הכנסה לטבלה BusRide
12	שימוש ב EXCEL
12	שימוש ב data generator
12	הכנסה לטבלה Station
13	גיבוי ושחזור
<b>14</b>	<b>שאלות ללא פרמטרים</b>
14	שאלתא 1
15	שאלתא 2
15	שאלתא 3
16	שאלתא 4
<b>17</b>	<b>שאלות מחיקה</b>
17	שאלתא 5
19	שאלתא 6
<b>20</b>	<b>שאלות עדכון</b>
20	שאלתא 7

22	שאלתא 8
<b>23</b>	<b>שאלתות עם פרמטרים</b>
23	שאלתא 9
24	שאלתא 10
25	שאלתא 11
26	שאלתא 12
<b>27</b>	<b>אילוצים</b>
27	אילוץ 1
28	אילוץ 2
28	אילוץ 3
<b>30</b>	<b>פונקציות</b>
30	פונקציה 1
31	פונקציה 2
<b>32</b>	<b>פרוצדורות</b>
32	פרוצדורה 1
33	פרוצדורה 2
<b>36</b>	<b>תכניות ראשיות</b>
36	תכנית ראשית 1
37	דוגמת הרצה:
37	דוגמת הרצה שכוללת חריגה:
38	תכנית ראשית 2
39	דוגמת הרצה:
39	דוגמת הרצה שכוללת חריגה:

## תיאור הפרויקט

- מערכת זו מיועדת לניהול ומעקב אחר פעילות תחבורה של חברת הסעות.
- היא כוללת נתונים על אוטובוסים, מוניות, נהגים, תחנות, מסלולים ולוחות זמנים.
- הפונקציונליות העיקרית היא לאפשר תכנון, ניהול ומעקב של נסיעות באוטובוס ובמוניות.

### ישויות נתונים וקשרים

#### • אוטובוס (Buses):

- מזהה ייחודי לכל אוטובוס (BusID)
- דגם (Model)
- קיבולת נוסעים (Capacity)
- תאריך רכישה (PurchaseDate)
- לוחית רישוי (LicensePlate)

#### • מונית (Taxi):

- מזהה ייחודי למונית (TaxiID) (PK)
- נהג (מפתח זר לטבלת נהגים) (DriverID) (FK)
- דגם (Model)
- קיבולת נוסעים (Capacity)
- תאריך רכישה (PurchaseDate)
- לוחית רישוי (LicensePlate)

#### • נהג (Driver):

- מזהה ייחודי לכל נהג (DriverID)
- שם מלא (FullName)
- מספר רישיון נהיגה (LicenseNumber)
- תאריך קליטה לעבודה (HireDate)

#### • תחנה (Station):

- מזהה ייחודי לכל תחנה (StationID)
- שם התחנה (StationName)
- כתובת התחנה (StationAddress)

#### • קו (Line):

- מזהה ייחודי לכל מסלול (LineID)
- שם המסלול (LineName)
- לוח זמנים (מפתח זר לטבלת לוחות זמנים) (ScheduleID)
- תחנת התחלה (מפתח זר לטבלת תחנות) (StartStationID)
- תחנת סיום (מפתח זר לטבלת תחנות) (EndStationID)

• **לוחות זמנים (Schedule):**

- מזהה ייחודי ללוח זמנים (ScheduleID)
- תדירות (Frequency)
- זמן יציאה ראשון (FirstDepartureTime)
- זמן יציאה אחרון (LastDepartureTime)

• **נסיעת אוטובוס (BusDriving): (ישות חלשה)**

- מזהה למסלול (LineID)
- מזהה לנהג (DriverID)
- מזהה לאוטובוס (BusID)
- שעת התחלת עבודה (StartTime)
- שעת סיום עבודה (FinishTime)

• **נסיעה במונית (DrivesInTaxi): (קשר)**

- מזהה לנהג (DriverID)
- מזהה למונית (TaxiID)
- שעת התחלת עבודה (StartTime)
- שעת סיום עבודה (FinishTime)
- איזור עבודה (WorkingZone)

• **תחנה סופית של קו (EndOf): (קשר)**

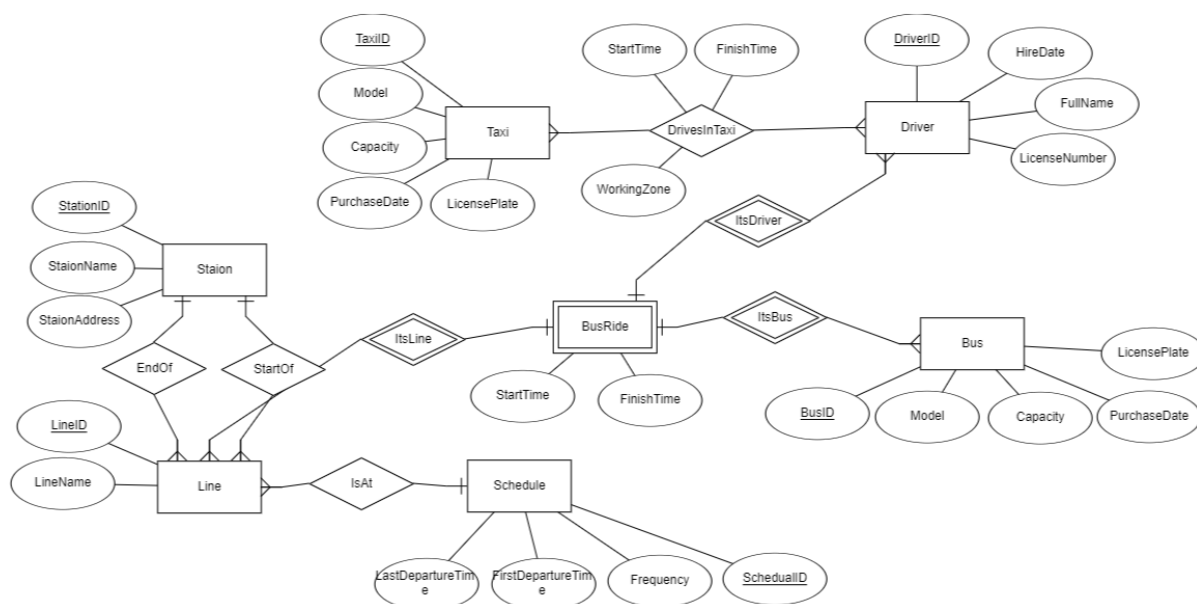
• **תחנה התחלתית של קו (StartOf): (קשר)**

• **לוח זמנים של קו (IsAt): (קשר)**

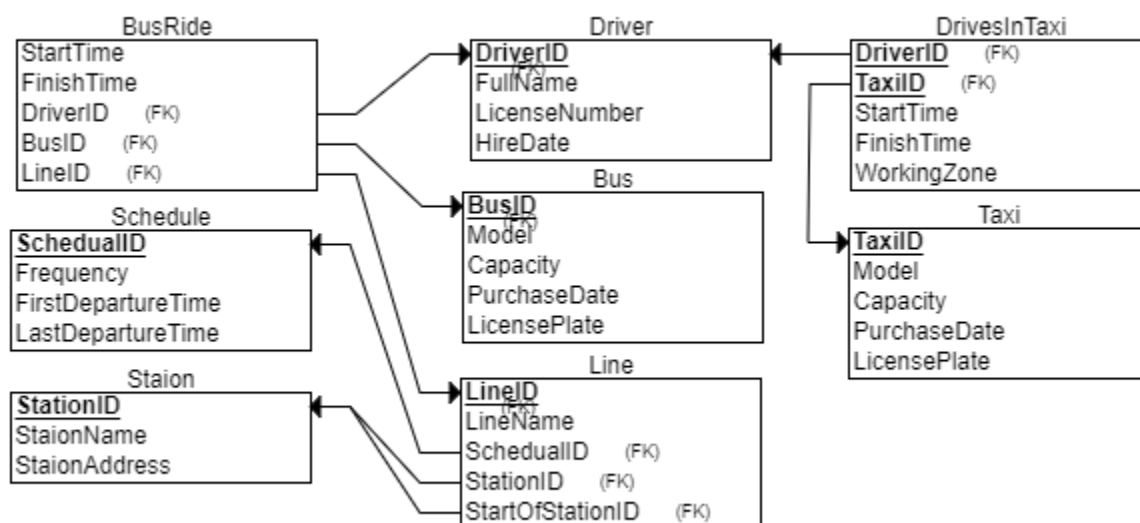
**פונקציונליות**

- **ניהול מידע על אוטובוסים, נהגים מוניות ותחנות**
  - הוספה, עדכון ומחיקה של פרטי אוטובוסים, מוניות, נהגים ותחנות במערכת
- **הגדרת מסלולים ולוחות זמנים לנסיעות**
  - יצירה וניהול של מסלולי נסיעה ולוחות זמנים מתוזמנים
- **תכנון ושיוך נסיעות לאוטובוסים, מוניות, נהגים ומסלולים ספציפיים**
  - שיוך אוטובוסים, מוניות, נהגים ומסלולים לנסיעות ספציפיות על בסיס זמינות ודרישות
- **מעקב אחר ביצוע הנסיעות בפועל**
  - רישום והצגת נתוני נסיעות בפועל, כולל תאריכים, אוטובוסים, מוניות, נהגים ומסלולים
- **ניתוח נתוני נסיעות לצורכי תחזוקה, תפעול ושיפור השירות**
  - הפקת דוחות וניתוחים סטטיסטיים לגבי נתוני הנסיעות לשיפור התפעול והשירות

## תרשים ERD



## תרשים DSD



## יצירת הטבלאות

יצירת טבלת **Bus**

-- Create Bus Table

```
CREATE TABLE Bus (
  BusID INT NOT NULL,
  LicensePlate VARCHAR(100) NOT NULL, -- License plate
  Model VARCHAR(100) NOT NULL, -- Model of the bus
  Capacity INT NOT NULL, -- Capacity of the bus
  PurchaseDate DATE NOT NULL, -- Purchase date of the bus
  PRIMARY KEY (BusID)
);
```

```
SQL> desc bus
Name          Type          Nullable Default Comments
-----
BUSID         INTEGER
LICENSEPLATE VARCHAR2(100)
MODEL         VARCHAR2(100)
CAPACITY      INTEGER
PURCHASEDATE DATE
```

יצירת טבלת **Line**

```
-- Create Line Table
CREATE TABLE Line (
  LineID INT NOT NULL,
  LineName VARCHAR(100) NOT NULL, -- Name of the
  ScheduleID INT NOT NULL, -- Foreign Key referen
  StartOfStationID INT NOT NULL, -- Foreign Key r
  EndOfStationID INT NOT NULL, -- Foreign Key referencing Station (end station)
  PRIMARY KEY (LineID),
  FOREIGN KEY (ScheduleID) REFERENCES Schedule(ScheduleID),
  FOREIGN KEY (StartOfStationID) REFERENCES Station(StationID),
  FOREIGN KEY (EndOfStationID) REFERENCES Station(StationID)
);
```

```
SQL> desc line
Name          Type          Nullable Default Comments
-----
LINEID        INTEGER
LINENAME      VARCHAR2(100)
SCHEDULEID    INTEGER
STARTOFSTATIONID INTEGER
ENDOFSTATIONID INTEGER
```

יצירת טבלת **Driver**

```
-- Create Driver Table
CREATE TABLE Driver (
  DriverID INT NOT NULL,
  FullName VARCHAR(100) NOT NULL, -- Name of the driver
  LicenseNumber INT NOT NULL, -- License number of the driver
  HireDate DATE NOT NULL, -- Hire date of the driver
  PRIMARY KEY (DriverID)
);
```

```
SQL> desc driver
Name          Type          Nullable Default Comments
-----
DRIVERID      NUMBER(38)
FULLNAME      VARCHAR2(100)
LICENSENUMBER VARCHAR2(50)
HIREDATE      DATE
```

## יצירת טבלת Schedule

```
-- Create Schedule Table
CREATE TABLE Schedule (
  ScheduleID INT NOT NULL, -- Unique
  Frequency INT NOT NULL, -- Frequency of the schedule
  FirstDepartureTime TIME NOT NULL, -- First departure time
  LastDepartureTime TIME NOT NULL, -- Last departure time
  PRIMARY KEY (ScheduleID)
);
```

```
SQL> desc schedule
Name                               Type                               Nullable Default Comments
-----
SCHEDULEID                         INTEGER                           YES
FREQUENCY                         INTEGER                           YES
FIRSTDEPARTURETIME                 VARCHAR2(8)                       YES
LASTDEPARTURETIME                 VARCHAR2(8)                       YES
```

## יצירת טבלת Station

```
-- Create Station Table
CREATE TABLE Station (
  StationID INT NOT NULL,
  StationName VARCHAR(100) NOT NULL, -- Name of the station
  StationAddress VARCHAR(255) NOT NULL, -- Address of the station
  PRIMARY KEY (StationID)
);
```

```
SQL> desc station
Name                               Type                               Nullable Default Comments
-----
STATIONID                         NUMBER(38)                         YES
STATIONNAME                       VARCHAR2(100)                      YES
STATIONADDRESS                     VARCHAR2(255)                      YES
```

## יצירת טבלת Taxi

```
-- Create Taxi Table
CREATE TABLE Taxi (
  TaxiID INT NOT NULL,
  LicensePlate VARCHAR(100) NOT NULL, -- License plate
  Model VARCHAR(100) NOT NULL, -- Model of the bus
  Capacity INT NOT NULL, -- Capacity of the bus
  PurchaseDate DATE NOT NULL, -- Purchase date of the bus
  PRIMARY KEY (TaxiID)
);
```

```
SQL> desc taxi
Name                               Type                               Nullable Default Comments
-----
TAXIID                           INTEGER                           YES
LICENSEPLATE                     VARCHAR2(100)                     YES
MODEL                           VARCHAR2(100)                     YES
CAPACITY                         INTEGER                           YES
PURCHASEDATE                     DATE                             YES
```



## יצירת טבלת BusRide

```
SQL> desc busride
Name          Type          Nullable Default Comments
-----
BUSID         INTEGER
LINEID        INTEGER
DRIVERID      INTEGER
STARTTIME     VARCHAR2(8)
FINISHTIME    VARCHAR2(8)
```

```
-- Create BusRide Table
CREATE TABLE BusRide (
  BusID INT NOT NULL, -- Foreign Key referencing Bus
  LineID INT NOT NULL, -- Foreign Key referencing Line
  DriverID INT NOT NULL, -- Foreign Key referencing Driver
  StartTime VARCHAR2(8) NOT NULL, -- Start time of the taxi shift in HH24:MI:SS format
  FinishTime VARCHAR2(8) NOT NULL, -- Finish time of the taxi shift in HH24:MI:SS format

  PRIMARY KEY (BusID, LineID, DriverID),
  FOREIGN KEY (BusID) REFERENCES Bus(BusID),
  FOREIGN KEY (LineID) REFERENCES Line(LineID),
  FOREIGN KEY (DriverID) REFERENCES Driver(DriverID),
  CONSTRAINT chk_StartTime1 CHECK (REGEXP_LIKE(StartTime, '^d{1,2}:d{1,2}:d{1,2}$')),
  CONSTRAINT chk_FinishTime1 CHECK (REGEXP_LIKE(FinishTime, '^d{1,2}:d{1,2}:d{1,2}$'))
);
```

## יצירת טבלת DriveInTaxi

```
SQL> desc drivesintaxi
Name          Type          Nullable Default Comments
-----
DRIVERID      INTEGER
TAXIID        INTEGER
STARTTIME     VARCHAR2(8)
FINISHTIME    VARCHAR2(8)
WORKINGZONE    VARCHAR2(100)
```

```
-- Create DrivesInTaxi Table
CREATE TABLE DrivesInTaxi (
  DriverID INT NOT NULL, -- Foreign Key referencing Driver
  TaxiID INT NOT NULL, -- Foreign Key referencing Taxi
  StartTime VARCHAR2(8) NOT NULL, -- Start time of the DriveInTaxi shift in HH24:MI:SS format
  FinishTime VARCHAR2(8) NOT NULL, -- Finish time of the DriveInTaxi shift in HH24:MI:SS format
  WorkingZone VARCHAR(100) NOT NULL, -- Working zone of the DriveInTaxi

  PRIMARY KEY (DriverID, TaxiID),
  FOREIGN KEY (DriverID) REFERENCES Driver(DriverID),
  FOREIGN KEY (TaxiID) REFERENCES Taxi(TaxiID),
  CONSTRAINT chk_StartTime2 CHECK (REGEXP_LIKE(StartTime, '^d{1,2}:d{1,2}:d{1,2}$')),
  CONSTRAINT chk_FinishTime2 CHECK (REGEXP_LIKE(FinishTime, '^d{1,2}:d{1,2}:d{1,2}$'))
);
```

שיטות למילוי הטבלאות

אנחנו נמלא את הנתונים ב 3 דרכים שונות:

שימוש ב mockaroo

הכנסה לטבלה Bus

Field Name	Type	Options
BusID	Row Number	blank: 0 %
LicensePlate	Digit Sequence	###-###-##
Model	Car Model	blank: 0 %
Capacity	Number	min: 20 max: 60 decimals: 0 blank: 0 %
PurchaseDate	Datetime	06/04/2017 to 06/04/2024 format: m/d/yyyy blank: 0 %

	BUSID	LICENSEPLATE	MODEL	CAPACITY	PURCHASEDATE
1	1	25-594-39	Quest	36	8/17/2018
2	2	46-862-66	ES	59	4/6/2019
3	3	58-756-53	Nitro	47	9/26/2019
4	4	52-350-93	929	26	2/24/2018
5	5	17-475-49	Taurus	24	10/27/2017
6	6	14-604-74	Murano	22	2/29/2020
7	7	81-439-09	S10	50	7/26/2023
8	8	26-908-79	F430	39	8/11/2019
9	9	53-020-75	F-Series	49	7/5/2022
10	10	24-114-03	GranTurismo	47	9/26/2022

הכנסה לטבלה Driver

Field Name	Type	Options
DriverID	Row Number	blank: 0 %
FullName	Full Name	blank: 0 %
LicenseNumbere	Digit Sequence	#####-####
HireDate	Datetime	05/30/2018 to 05/30/2024 format: m/d/yyyy blank: 0 %

	DRIVERID	FULLNAME	LICENSENUMBER	HIREDATE
1	1	Travers McGrane	3476-8840-8675	10/31/2018
2	2	Antonin Fominov	9517-1392-5878	10/18/2022
3	3	Gracia Pamplin	3170-9041-2787	5/27/2021
4	4	Adele Brozsset	3225-3617-6123	2/26/2020
5	5	Cosette Swinnard	0048-0764-5397	8/12/2020
6	6	Hermon Ascraft	4095-8246-9689	9/24/2023
7	7	Glynnis Strase	0709-3537-7335	9/13/2020
8	8	Zsazsa Elkington	8401-4035-3911	9/19/2023
9	9	Noble Logie	7810-2427-3690	10/9/2023
10	10	Findley Ullett	6662-6916-3131	11/23/2021

הכנסה לטבלה Taxi

Field Name	Type	Options
TaxiID	Row Number	blank: 0 %
LicensePlate	Digit Sequence	##-###-##
Model	Car Model	blank: 0 %
Capacity	Number	min: 20 max: 60 decimals: 0 blank: 0 %
PurchaseDate	Datetime	06/04/2017 to 06/04/2024 format: m/d/yyyy blank: 0 %

	TAXIID	LICENSEPLATE	MODEL	CAPACITY	PURCHASEDATE
1	1	15-193-82	Fox	7	4/2/2022
2	2	94-642-54	Tracer	6	1/22/2019
3	3	92-205-32	MX-6	3	5/3/2022
4	4	65-691-37	V50	5	12/14/2021
5	5	71-491-92	911	2	10/18/2018
6	6	38-227-49	Fox	7	5/7/2023
7	7	77-247-65	3500	3	9/28/2019
8	8	13-040-19	Park Avenue	4	7/8/2023
9	9	20-244-23	GL-Class	10	2/7/2018
10	10	43-487-38	New Beetle	4	1/21/2023

## הכנסה לטבלה Schedule

Field Name	Type	Options
ScheduleID	Row Number	blank: 0 %
Frequency	Number	min: 10 max: 720
FirstDepartureTime	Time	from 12:00 AM to 11:59 PM
LastDepartureTime	Time	from 12:00 AM to 11:59 PM

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

	SCHEDULEID	FREQUENCY	FIRSTDEPARTURETIME	LASTDEPARTURETIME
1	1	367	11:18:47	0:03:28
2	2	33	15:26:30	17:29:15
3	3	172	19:19:39	10:15:25
4	4	660	1:00:20	11:12:55
5	5	504	1:04:19	19:33:06
6	6	424	14:14:35	7:52:39
7	7	347	4:51:35	0:08:54
8	8	334	13:50:29	9:18:13
9	9	529	7:09:42	4:32:02
10	10	666	20:17:07	21:26:17

## הכנסה לטבלה Line

Field Name	Type	Options
LineID	Row Number	blank: 0 %
LineName	Digit Sequence	###
ScheduleID	Number	min: 1 max: 400 decimals: 0 blank: 0 %
StartOfStationID	Number	min: 1 max: 400 decimals: 0 blank: 0 %
EndOfStationID	Number	min: 1 max: 400 decimals: 0 blank: 0 %

	LINEID	LINENAME	SCHEDULEID	STARTOFSTATIONID	ENDOFSTATIONID
1	1	554	289	399	241
2	2	71	328	334	214
3	3	814	172	224	361
4	4	467	226	295	345
5	5	10	111	229	226
6	6	754	264	388	365
7	7	109	256	285	124
8	8	186	111	186	86
9	9	623	43	65	265
10	10	779	339	340	95

## הכנסה לטבלה DrivesInTaxi

Field Name	Type	Options
DriverID	Number	min: 1 max: 400
TaxiID	Number	min: 1 max: 400
StartTime	Time	from 12:00 AM to 11:59 PM format: 24 Hour w/seconds blank: 0 %
FinishTime	Time	from 12:00 AM to 11:59 PM format: 24 Hour w/seconds blank: 0 %
WorkingZone	City	blank: 0 %

	DRIVERID	TAXIID	STARTTIME	FINISHTIME	WORKINGZONE
1	348	178	6:17:08	3:59:11	Satte
2	314	390	15:47:00	14:06:23	Si Wilai
3	374	364	5:02:56	8:07:21	Biting
4	100	110	15:51:51	14:42:33	Wyszogród
5	243	295	6:44:49	17:07:51	San Martín
6	26	289	13:34:19	7:25:19	Ponta Porã
7	150	259	11:45:31	8:13:38	Diofior
8	217	272	0:11:46	22:39:03	Yejia
9	326	196	10:41:07	18:16:10	Balut
10	93	32	0:54:37	23:40:57	Panorama

## הכנסה לטבלה BusRide

Field Name	Type	Options
BusID	Number	min: 1 max:
LineID	Number	min: 1 max:
DriverID	Number	min: 1 max:
StartTime	Time	from 12:00 AM to 11:59 PM
FinishTime	Time	from 12:00 AM to 11:59 PM format: 24 Hour w/seconds blank: 0 % Σ X

	BUSID	LINEID	DRIVERID	STARTTIME	FINISHTIME
1	183	88	49	22:23:23	14:54:50
2	351	356	143	12:19:43	9:07:18
3	381	110	164	11:08:08	17:21:52
4	128	237	349	10:27:33	14:44:38
5	245	337	90	13:33:20	9:39:28
6	159	197	351	1:18:47	5:47:02
7	102	193	375	2:53:10	1:44:39
8	15	8	16	6:14:23	2:51:30
9	314	290	170	2:15:30	12:04:28
10	166	257	299	8:28:34	10:10:44

שימוש ב EXCEL

A	B	C	D	E
BusID	Model	Capacit	PurchaseDate	LicensePlate
1	Toyota2004	60	9/10/2021	49-391-34
2	BMW 34	19	6/7/2023	12-361-04
3	Pejo 3004	42	6/3/2022	32-692-79
4	Lamborghini 45	64	6/17/2023	84-284-85
5	Mazda3	56	12/30/2022	23-593-23
6	Suburban 2500	35	5/21/2024	67-938-91

Table / Query	
<input checked="" type="radio"/> Import Table	<input type="radio"/> Import Query Result
C:\Users\avish\OneDrive\Desktop\basnat\Tar\DB:	
View Data	

Result Preview				
BusID	Model	Capacity	PurchaseDate	LicensePlate
1.0	Toyota2004	60.0	9/10/2021	49-391-34
2.0	BMW 34	19.0	6/7/2023	12-361-04
3.0	Pejo 3004	42.0	6/3/2022	32-692-79
4.0	Lamborghini 45	64.0	6/17/2023	84-284-85
5.0	Mazda3	56.0	12/30/2022	23-593-23
6.0	Suburban 2500	35.0	5/21/2024	67-938-91

שימוש ב data generator

## הכנסה לטבלה Station

STATION			
Owner	Table	Number of	
C##AVISHAI	STATION	400	
Name	Type	Size	Data
STATIONID	NUMBER		Sequence(1 [1], [400])
STATIONNAME	VARCHAR2	100	'Station ' + LastName
STATIONADDRESS	VARCHAR2	255	Address1

	STATIONID	STATIONNAME	STATIONADDRESS
1	1	Station Close	31 Downey Ave
2	2	Station Calle	62 Hanks Drive
3	3	Station Penn	30 Spike Road
4	4	Station May	59 Barbara Road
5	5	Station Wopat	797 Mattea Street
6	6	Station Gates	35 Zwolle Street
7	7	Station Patrick	80 Diehl Street
8	8	Station Dillon	9 Chilton Street
9	9	Station Cohn	455 Oldenburg Street
10	10	Station Spine	51 Gertner Street

## גיבוי ושחזור

Export Tables of C##AVISHAI

Name	Type	Compiled
BUS	TABLE	5/30/2024 2:11:15 PM
BUSRIDE	TABLE	5/30/2024 2:11:15 PM
DRIVER	TABLE	5/30/2024 2:11:15 PM
DRIVESINTAXI	TABLE	5/30/2024 2:11:15 PM
LINE	TABLE	5/30/2024 2:11:15 PM
SCHEDULE	TABLE	5/30/2024 2:11:15 PM
STATION	TABLE	5/30/2024 2:11:15 PM
TAXI	TABLE	5/30/2024 2:11:15 PM

User <CURRENT USER>

Oracle Export

SQL Inserts

PL/SQL Developer

☐ Drop tables

☒ Create tables

☐ Truncate tables

☐ Delete records

☒ Disable triggers

☐ Disable foreign key constraints

☐ Include storage

☐ Include privileges

Commit every 0 records (0 = never)

Where clause

☐ Zip

Output file C:\Users\avish\OneDrive\Desktop\basnat\Tar1\DB5784\_7001\_1752\stageA\backup.sql

Export

C##avishai@XE

Import Tables

Oracle Import

SQL Inserts

PL/SQL Developer

☐ Use Command Window

☒ Use SQL\*Plus

SQL\*Plus Executable

C:\app\avish\product\21c\dbhomeXE\bin\sqlpl

Import file

C:\Users\avish\OneDrive\Desktop\basnat\Tar1\DB5784\_7001\_1752\stageA\backup.sql

Import

## חלק ב

### שאלות ללא פרמטרים

#### שאלת 1

איציק המנהל ביקש רשימה מסודרת של כל המוניות שכרגע פועלות בשטח - מה הלוחית רישוי שלהן, איך קוראים לנהג שלהן וכן את איזור העבודה שלהן.  
שאלתא שמביאה את כל הנהגים שנוהגים במוניות כולל את הלוחית רישוי של המונית ואיזור העבודה של הנהג והמונית שלה.

```
-- Query 1: get the driver names that drive in a Taxi and there Taxi
licenses plate and there working zone and order by the name
SELECT Driver.FullName, Taxi.LicensePlate, DrivesInTaxi.WorkingZone
FROM Driver
JOIN DrivesInTaxi ON Driver.DriverID = DrivesInTaxi.DriverID
JOIN Taxi ON DrivesInTaxi.TaxiID = Taxi.TaxiID
ORDER BY Driver.FullName;
```

	FULLNAME	LICENSEPLATE	WORKINGZONE
1	Adelina Rolph	47-133-52	Zhoutie
2	Adelina Rolph	08-621-86	Alfenas
3	Agnese Lofty	27-562-32	Shibetsu
4	Agnese Lofty	56-419-31	GardÃ"z
5	Agosto Tingle	49-645-74	San Francisco
6	Agosto Tingle	66-801-83	Sanyantang
7	Alana Warrington	08-621-86	Marataizes
8	Alana Warrington	29-420-65	Guizi
9	Alberto Ghioni	47-961-58	AngoulÃ"me
10	Aldric Carette	45-657-41	Muaratiga
11	Alejandra Gooddie	45-005-29	Marseille
12	Alejandra Gooddie	17-277-00	Mondlo
13	Alejandra Gooddie	24-231-39	Eá¹- á¹-aiyiba
14	Alexandra Lackeye	88-212-94	Chiduo
15	Alexandra Lackeye	47-133-52	Hamburg
16	Alexandra Lackeye	63-064-70	SÃ£o Bento
17	Alexandra Lackeye	84-093-62	Sedati
18	Alfy Dax	82-950-43	Bantengan
19	Alfy Dax	35-019-58	Loshnitsa
20	Alica Hawse	59-708-87	Flagstaff
21	Alica Hawse	00-981-43	Shatou
22	Alica Hawse	85-515-11	Kanshan
23	Alistair Hanny	39-770-98	Yuanqiao
24	Almeria Gornall	38-763-53	ã€"Arab ar RashÄydah
25	Almeria Gornall	31-962-08	Kagadi

## שאלתא 2

בחברה שלנו רוצים לייעל את התפוקה שאנחנו מקבלים מכל אוטובוס, לשם כך, נבצע סקר שיבדוק מה החברה של האוטובוס שבו אנחנו משתמשים הכי הרבה.

שאלתא שמחזירה את החברה של האוטובוס שאנחנו משתמשים הכי הרבה באוטובוסים שלה

```
-- Query 2: Select the bus model that has the highest number of associated
rides in the BusRide table
SELECT b.Model, COUNT(*) AS TotalRides
FROM Bus b
JOIN BusRide br ON b.BusID = br.BusID
GROUP BY b.Model
ORDER BY TotalRides DESC
FETCH FIRST 1 ROWS ONLY;
```

Select driver

Select bus

Select line

Select busride

</

```
AND TO_TIMESTAMP(schd.FirstDepartureTime, 'HH24:MI:SS') >=
TO_TIMESTAMP('17:00:00', 'HH24:MI:SS');
```

		LINENAME		FIRSTDEPARTURETIME	
▶	1	472a	***	19:19:39	
	2	412	***	20:17:07	
	3	367c	***	20:58:15	
	4	489	***	21:59:03	

## שאלתא 4

בעקבות רבעון רווחי שהחברה עברה, היא החליטה לצ'פר את כל העובדים שיש להם וותק של לפחות שנתיים בחברה, ביום כיף בימית 2000.

```
-- Query 4: Select all bus rides with drivers who have been hired before a
certain date
SELECT driver.FullName, driver.HireDate
FROM BusRide busride
JOIN Driver driver ON busride.DriverID = driver.DriverID
WHERE MONTHS_BETWEEN(SYSDATE, driver.HireDate) / 12 >= 2;
```

		FULLNAME		HIREDATE	
▶	1	Ferdinanda Coundley	***	8/30/2018	***
	2	Sadie Alloisi	***	11/24/2021	***
	3	Kermit Sully	***	5/19/2019	***
	4	Bridie Golthorpp	***	5/4/2021	***
	5	Marlene Paszak	***	11/26/2020	***
	6	Loleta MacDowall	***	4/4/2022	***
	7	Catie Novak	***	11/22/2020	***
	8	Antonius Djakovic	***	3/9/2020	***
	9	Sheila Schoroder	***	5/5/2019	***
	10	Beniamino Crowther	***	6/2/2021	***
	11	Bebe Pardy	***	11/14/2018	***
	12	Parry Rasor	***	6/25/2019	***
	13	Diahann Abdee	***	5/11/2021	***
	14	Golda Madgwick	***	4/4/2019	***
	15	Marilyn Da Costa	***	2/3/2019	***
	16	Katherine Beardwell	***	1/16/2021	***
	17	Alistair Hanny	***	9/7/2020	***
	18	Hammad Wilsher	***	4/24/2019	***
	19	Isabella Dryden	***	11/18/2020	***
	20	Gracia Pamplin	***	5/27/2021	***



## שאלות מחיקה

### שאלתא 5

עקב קיצוצים בסניף בנהריה, הוחלט להוציא משירות את כל המוניות שעובדות בין השעות 2 בלילה ל 7 בבוקר.

```
-- Step 1: Verify the existing records
SELECT *
FROM DrivesInTaxi dit
WHERE dit.WorkingZone = 'Nahariya';

-- Step 2: Delete related records in DrivesInTaxi
DELETE FROM DrivesInTaxi dit
WHERE dit.WorkingZone = 'Nahariya'
AND TO_TIMESTAMP(dit.StartTime, 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP('02:00:00', 'HH24:MI:SS') AND TO_TIMESTAMP('07:00:00',
'HH24:MI:SS')
AND TO_TIMESTAMP(dit.FinishTime, 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP('02:00:00', 'HH24:MI:SS') AND TO_TIMESTAMP('07:00:00',
'HH24:MI:SS');

-- Step 3: Verify the deletion
SELECT *
FROM DrivesInTaxi dit
WHERE dit.WorkingZone = 'Nahariya';
```

Select drivesintaxi Delete drivesintaxi Select drivesintaxi

	DRIVERID	TAXIID	STARTTIME	FINISHTIME	WORKINGZONE
1	55	128	12:55:54	23:05:01	Nahariya
2	184	88	7:28:22	16:13:24	Nahariya
3	81	48	11:30:30	8:12:08	Nahariya
4	323	368	12:52:32	10:48:03	Nahariya
5	160	208	10:40:17	21:02:24	Nahariya
6	107	128	8:33:52	6:24:38	Nahariya
7	126	88	5:32:51	17:05:12	Nahariya
8	322	208	0:05:51	20:46:09	Nahariya
9	193	88	11:48:40	22:23:32	Nahariya
10	135	128	13:46:44	13:39:07	Nahariya
11	30	88	19:22:33	17:57:41	Nahariya
12	82	88	7:20:58	23:14:03	Nahariya
13	297	328	1:54:44	15:11:03	Nahariya
14	291	48	7:28:32	4:30:36	Nahariya
15	79	48	9:00:20	11:37:54	Nahariya
16	400	168	23:19:49	18:43:57	Nahariya
17	156	328	20:54:46	6:30:51	Nahariya
18	360	208	3:49:34	14:59:48	Nahariya
19	121	248	15:31:25	5:48:29	Nahariya
20	235	208	16:51:23	3:15:45	Nahariya
21	39	128	18:23:22	10:11:04	Nahariya
22	385	248	17:50:59	20:36:56	Nahariya
23	191	368	4:21:18	6:31:01	Nahariya
24	264	248	15:02:01	18:08:00	Nahariya
25	19	328	22:08:21	9:09:34	Nahariya
26	306	48	10:28:03	18:00:29	Nahariya

1 of 27 C##avishai@XE [11:30:37 PM] 27 rows selected in 0.055 seconds

Select drivesintaxi Delete drivesintaxi Select drivesintaxi

	DRIVERID	TAXIID	STARTTIME	FINISHTIME	WORKINGZONE
1	55	128	12:55:54	23:05:01	Nahariya
2	184	88	7:28:22	16:13:24	Nahariya
3	81	48	11:30:30	8:12:08	Nahariya
4	323	368	12:52:32	10:48:03	Nahariya
5	160	208	10:40:17	21:02:24	Nahariya
6	107	128	8:33:52	6:24:38	Nahariya
7	126	88	5:32:51	17:05:12	Nahariya
8	322	208	0:05:51	20:46:09	Nahariya
9	193	88	11:48:40	22:23:32	Nahariya
10	135	128	13:46:44	13:39:07	Nahariya
11	30	88	19:22:33	17:57:41	Nahariya
12	82	88	7:20:58	23:14:03	Nahariya
13	297	328	1:54:44	15:11:03	Nahariya
14	291	48	7:28:32	4:30:36	Nahariya
15	79	48	9:00:20	11:37:54	Nahariya
16	400	168	23:19:49	18:43:57	Nahariya
17	156	328	20:54:46	6:30:51	Nahariya
18	360	208	3:49:34	14:59:48	Nahariya
19	121	248	15:31:25	5:48:29	Nahariya
20	235	208	16:51:23	3:15:45	Nahariya
21	39	128	18:23:22	10:11:04	Nahariya
22	385	248	17:50:59	20:36:56	Nahariya
23	264	248	15:02:01	18:08:00	Nahariya
24	19	328	22:08:21	9:09:34	Nahariya
25	306	48	10:28:03	18:00:29	Nahariya

1 of 26 C##avishai@XE [11:30:34 PM] 26 rows selected in 0.049 seconds

## שאלתא 6

אנחנו רוצים להוציא משימוש את כל האוטובוסים הישנים מלפני 2022. (לא למחוק אותם מהטבלה של האוטובוסים אלא רק להוציא אותם משימוש)

```
-- Query 2: Delete all bus rides for a specific bus that has a purchase
date before 2015
SELECT *
FROM BusRide br;

DELETE FROM BusRide
WHERE BusID IN (SELECT BusID FROM Bus WHERE PurchaseDate <
TO_DATE('2022-01-01', 'YYYY-MM-DD'));

SELECT *
FROM BusRide br;
```

לפני מחיקה

Select busride Select bus Delete busride Select busride				
	BUSID	LINEID	DRIVERID	
1	146	238	228	
2	239	319	43	
3	132	223	96	
4	253	361	10	
5	108	188	238	
6	184	215	133	
7	26	138	47	
8	322	12	352	
9	9	228	316	
10	354	209	297	
11	165	324	95	
12	385	228	62	

400 rows selected in 0.480 seconds

אחרי מחיקה

Select busride Select bus Delete busride Select busride				
	BUSID	LINEID	DRIVERID	
1	146	238	228	
2	239	319	43	
3	132	223	96	
4	253	361	10	
5	108	188	238	
6	184	215	133	
7	26	138	47	
8	322	12	352	
9	9	228	316	
10	354	209	297	
11	165	324	95	
12	385	228	62	

1 of 340 C#avishai@XE [6:06:43 PM] 340 rows selected in 0.231 seconds

## שאלות עדכון

### שאלת 7

בעקבות המתח הביטחוני ששורר בגזרת שכם, הוחלט להעביר את כל הקווים שמסתיימים בחווארה אל אלון מורה.

השאלתא תחפש את כל ה busRides שהתחנה הסופית שלהם היא חווארה, ותשנה להם את התחנה הסופית לאלון מורה.

```
-- UPDATE:

-- Select all records from Line where EndOfStationID matches the StationID
of 'Elon Morre'
SELECT lne.LineName, station.StationName AS endStation,
station.StationAddress AS endStationAddress
FROM Line lne
JOIN Station station ON station.StationID = lne.EndOfStationID
WHERE lne.EndOfStationID = (
    SELECT station.StationID
    FROM Station station
    WHERE station.StationName = 'Hawarra'
    FETCH FIRST 1 ROWS ONLY
);

UPDATE Line
SET EndOfStationID = (
```

```

SELECT station.StationID
FROM Station station
WHERE station.StationName = 'Elon Morre'
FETCH FIRST 1 ROWS ONLY
)
WHERE EndOfStationID = (
SELECT station.StationID
FROM Station station
WHERE station.StationName = 'Hawarra'
FETCH FIRST 1 ROWS ONLY
);


SELECT lne.LineName, station.StationName AS endStation,
station.StationAddress AS endStationAddress
FROM Line lne
JOIN Station station ON station.StationID = lne.EndOfStationID
WHERE lne.EndOfStationID = (
SELECT station.StationID
FROM Station station
WHERE station.StationName = 'Elon Morre'
FETCH FIRST 1 ROWS ONLY
);


```


Select line


Update line


Select line




























		LINENAME		ENDSTATION		ENDSTATIONADDRESS	
▶	1	425a	***	Elon Morre	***	Hateena 10	***
	2	76	***	Elon Morre	***	Hateena 10	***
	3	209	***	Elon Morre	***	Hateena 10	***
	4	770	***	Elon Morre	***	Hateena 10	***


Select line


Update line


Select line


























		LINENAME		ENDSTATION		ENDSTATIONADDRESS	
▶	1	425a	***	Hawarra	***	Jabel Dawlla 11	***
	2	76	***	Hawarra	***	Jabel Dawlla 11	***
	3	209	***	Hawarra	***	Jabel Dawlla 11	***
	4	770	***	Hawarra	***	Jabel Dawlla 11	***

## שאלתא 8

חברות Tracer, Camaro, RAV4 הציעו שיפוץ בחינם לאוטובוסים שלהם שבאופן מפתיע וקסום יגרום לאוטובוסים שלהם להיות עם 60 מקומות כל אחד.

שאלתא שמעדכנת את האוטובוסים של החברות האלה ל-60 מקומות.

```
-- UPDATE:

-- Query 2: Update the capacity of buses purchased in the year 2020 to 60
SELECT *
FROM Bus b
WHERE b.Capacity = 60;

UPDATE Bus
SET Capacity = 60
WHERE Capacity < 60 AND (Model = 'Tracer' or Model = 'Camaro' or Model = 'RAV4');

SELECT *
FROM Bus b
WHERE b.Capacity = 60;
```

לפני עדכון

Select bus Update bus Select bus					
	BUSID	MODEL	CAPACITY	PURCHASEDATE	
1	1	Protege	60	9/10/2021	...
2	48	Impala	60	6/16/2022	...
3	78	Festiva	60	9/4/2023	...
4	140	57	60	2/1/2024	...
5	156	TT	60	1/14/2023	...
6	196	NV1500	60	2/28/2023	...
7	221	LeSabre	60	8/6/2022	...
8	264	B-Series Plus	60	7/13/2021	...
9	380	F450	60	1/21/2023	...
10	393	Firebird	60	1/6/2024	...

Select bus   Update bus   Select bus					
	BUSID	MODEL	CAPACITY	PURCHASEDATE	
1	1	Protege	60	9/10/2021	
2	17	Camaro	60	7/7/2022	←
3	31	RAV4	60	12/20/2023	←
4	48	Impala	60	6/16/2022	
5	78	Festiva	60	9/4/2023	
6	107	Tracer	60	6/19/2023	←
7	108	Tracer	60	5/9/2024	←
8	140	57	60	2/1/2024	←
9	146	RAV4	60	4/30/2022	
10	156	TT	60	1/14/2023	
11	196	NV1500	60	2/28/2023	
12	221	LeSabre	60	8/6/2022	
13	264	B-Series Plus	60	7/13/2021	
14	272	RAV4	60	2/7/2023	←
15	351	Camaro	60	11/25/2023	←
16	380	F450	60	1/21/2023	
17	393	Firebird	60	1/6/2024	

## שאלות עם פרמטרים

### שאלתא 9

בשביל שנוכל לנהל את הנהגים בכל איזור באופן יעיל, אנחנו צריכים לדעת מי פנוי, איפה ומתי. לשם כך, יצרנו שאלתא שמקבלת איזור עבודה, ואיזשהו טווח עבודה, ומחזירה את כל מי שפנוי במקום המסויים בשעות שניתן לשאלתא. לדוגמא, במקרה שלנו היה צורך בנהגים באיזור נהריה, שאינם עובדים בין השעה 6:00 לשעה 15:00.

```
-- List all drivers available within a specific time range.
SELECT driver.DriverID, driver.FullName
FROM Driver driver
JOIN Drivesintaxi drivesintaxi ON driver.driverid = drivesintaxi.driverid
WHERE driver.DriverID NOT IN (
    SELECT DriverID
    FROM BusRide busride
    WHERE (TO_TIMESTAMP('&StartTime', 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP(busride.StartTime, 'HH24:MI:SS') AND
TO_TIMESTAMP(busride.FinishTime, 'HH24:MI:SS'))
```

```

        OR TO_TIMESTAMP('&FinishTime', 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP(busride.StartTime, 'HH24:MI:SS') AND
TO_TIMESTAMP(busride.FinishTime, 'HH24:MI:SS'))
    )
AND driver.DriverID NOT IN (
    SELECT DriverID
    FROM DrivesInTaxi drivesintaxi
    WHERE (TO_TIMESTAMP('&StartTime', 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP(drivesintaxi.StartTime, 'HH24:MI:SS') AND
TO_TIMESTAMP(drivesintaxi.FinishTime, 'HH24:MI:SS')
        OR TO_TIMESTAMP('&FinishTime', 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP(drivesintaxi.StartTime, 'HH24:MI:SS') AND
TO_TIMESTAMP(drivesintaxi.FinishTime, 'HH24:MI:SS'))
    )
AND drivesintaxi.WorkingZone = '&WorkingZone';

```

Variables

Name	Value
StartTime	06:00:00
FinishTime	15:00:00
WorkingZone	Nahariya

	DRIVERID	FULLNAME
1	30	Rex Wildbore
2	81	Walt Baumaier
3	323	Lucretia Landsbury

OK

Cancel

Clear

## שאלתא 10

בשביל לעשות קצת סדר למנהל, ולהראות לו שבאמת העובדים שלנו מבצעים את עבודתם בכל שעות היום, אנחנו רוצים ליצור רשימה שבהינתן איזשהו טווח של שעות, תסכום לנו לכל נהג אוטובוס כמה פעמים הוא עובד, מבצע נסיעות, בטווח השעות הנ"ל. במקרה שלנו, טווח השעות הנ"ל הוא בין 2:00 ל 4:00.

```

-- Query 2: List drivers with the most bus rides in a specific period.
SELECT driver.DriverID, driver.FullName, COUNT(busride.BusID) AS
NumberOfRides

```



```
FROM Driver driver
JOIN BusRide busride ON driver.DriverID = busride.DriverID
WHERE TO_TIMESTAMP(busride.StartTime, 'HH24:MI:SS') BETWEEN
TO_TIMESTAMP('&StartPeriod', 'HH24:MI:SS') AND TO_TIMESTAMP('&EndPeriod',
'HH24:MI:SS')
GROUP BY driver.DriverID, driver.FullName
ORDER BY NumberOfRides DESC;
```

Name	Value
StartPeriod	02:00:00
EndPeriod	04:00:00

	DRIVERID	FULLNAME	NUMBEROFRIDES
1	80	Brade Margrie	3
2	311	Elladine Dilleston	2
3	344	Sarge Lawley	2
4	340	Edik Lyddon	2
5	182	Suzanna Yurygyn	2
6	57	Drusy Risborough	2
7	276	Marcel Ladds	2
8	170	Galven Netti	2
9	320	Kermit Sully	2
10	375	Dianemarie Brunker	1
11	6	Hermon Ascraft	1
12	264	Desdemona Durbin	1
13	139	Antonina Gavin	1
14	238	Juliette Mougel	1
15	302	Dacie McLleese	1
16	176	Aldric Carette	1
17	4	Adele Brozsset	1
18	171	Darren Scarlin	1
19	242	Maynard Atty	1
20	146	Gideon Doornbos	1
21	270	Rosanne Vennings	1

## שאלת 11

המנהל רוצה לצ'פר את נהגי המונית המצטיינים (האלו שעובדים הכי הרבה שעות) בכל איזור. לשם כך, הוא צריך שאלתא שתקבל איזור מסויים, ותחזיר לו את העובד שעובד הכי הרבה שעות באיזור זה.

```
-- Query 3: Retrieve the driver who works the most hours in a specific
working zone.
SELECT
    driverinstaxi.DriverID,
    driver.FullName,
    SUM(
```

```
(EXTRACT(HOUR FROM (TO_TIMESTAMP(driverinstaxi.FinishTime,
'HH24:MI:SS') - TO_TIMESTAMP(driverinstaxi.StartTime, 'HH24:MI:SS'))) * 60
+ EXTRACT(MINUTE FROM (TO_TIMESTAMP(driverinstaxi.FinishTime,
'HH24:MI:SS') - TO_TIMESTAMP(driverinstaxi.StartTime, 'HH24:MI:SS')))) /
60
) AS TotalHours
FROM DrivesInTaxi driverinstaxi
JOIN Driver driver ON driverinstaxi.DriverID = driver.DriverID
WHERE driverinstaxi.WorkingZone = '&WorkingZone'
GROUP BY driverinstaxi.DriverID, driver.FullName
ORDER BY TotalHours DESC
FETCH FIRST 1 ROWS ONLY;
```

Name	Value
WorkingZone	Nahariya

	DRIVERID	FULLNAME	TOTALHOURS
1	322	Mendie Toffetto	20.66666666666667

## שאלתא 12

בשביל שנוכל לצוות בקרים לקווים, אנחנו צריכים לדעת את כל הקווים שמתחילים בנקודה מסוימת, כדי שנוכל לבחור לאיזה קו להעלות כל בקר. לשם כך, ניצור שאלתא שמקבלת בתור פרמטר תחנת התחלה, ומחזירה את כל הקווים שמתחילים מתחנה זו.

```
-- Query 4: Retrieve lines starting from a specific station
SELECT
    line.LineID,
    start_station.StationName AS StartStationName,
    end_station.StationName AS EndStationName,
    line.LineName
FROM Line line
JOIN Station start_station ON line.StartOfStationID =
start_station.StationID
JOIN Station end_station ON line.EndOfStationID = end_station.StationID
WHERE start_station.StationName = '&<name="StationName" list="SELECT
StationName from Station">'
```

Name	Value
StationName	Station Wopat
	Station Close
	Station Calle
	Station Penn
	Station May
	Station Wopat
	Station Gates
	Station Patrick
	Station Dillon
	Station Cohn
	Station Spine

		LINEID	STARTSTATIONNAME	ENDSTATIONNAME	LINENAME
▶	1	233	Station Wopat	*** Station Sampson	*** 772
	2	326	Station Wopat	*** Station Eckhart	*** 115
	3	347	Station Wopat	*** Station McIntyre	*** 759

## אילוצים

### אילוץ 1

בעולם מתוקן, כל הלוחיות רישוי הן ייחודיות, לכן הוספנו את האילוץ הזה.

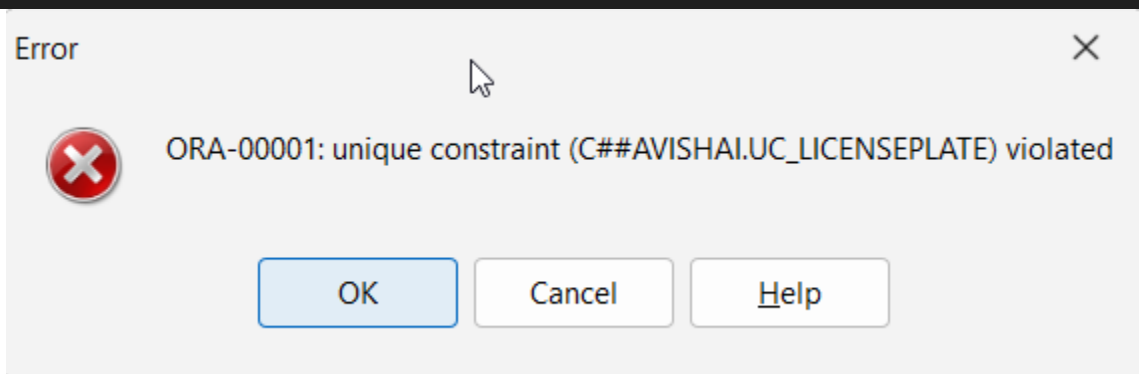
```
-- Constraint: Unique License Plate
ALTER TABLE Taxi
ADD CONSTRAINT uc_LicensePlate UNIQUE (LicensePlate);
```

תיאור השינוי: בפקודת ALTER TABLE נוספה אילוץ ייחודי על העמודה LicensePlate בטבלה Taxi, כך שלא יהיה ניתן להכניס יותר מרכב אחד עם אותו מספר רישוי.

ניסיון להכניס נתונים שמפרים את האילוץ: ננסה להכניס נתונים שמפרים את האילוץ על LicensePlate ייחודי.

```
-- insert data that violates constraint 1
INSERT INTO Taxi (TaxiID, LicensePlate, Model, Capacity, PurchaseDate)
VALUES (560, '56-023-12', 'Toyota', 7, to_date('11-08-2020',
'dd-mm-yyyy'));
```

```
-- same license plate
INSERT INTO Taxi (TaxiID, LicensePlate, Model, Capacity, PurchaseDate)
VALUES (561, '56-023-12', 'Mercedes', 5, to_date('19-11-2021',
'dd-mm-yyyy'));
```



## אילוח 2

אם במקרה יש נהג שחסר לו שם, שם ברירת המחדל יהיה "דקל וקנין", כי מתאים לו להיות נהג.

```
-- Constraint: Default value for driver's name
ALTER TABLE Driver
ALTER COLUMN FullName SET DEFAULT 'Dekel Vaknin';
```

תיאור השינוי: בפקודת ALTER TABLE עודכנה העמודה FullName בטבלה Driver כך שתקבל ערך ברירת מחדל של 'Dekel Vaknin'.

ניסיון להכניס נתונים שמפרים את האילוח: ננסה להכניס נתונים בלי הכנסה לעמודה Fullname, מה שיגרום להשמט ערך דיפולטיבי.

```
-- insert data without FullName, to show constraint 2
INSERT INTO Driver (DriverID, LicenseNumber, HireDate)
VALUES (450, '39-102-39', to_date('19-07-2020', 'dd-mm-yyyy'));

Select * from Driver
Where DriverID = 450
```

		DRIVERID	FULLNAME	LICENSENUMBER	HIREDATE
▶	1	450	Dekel Vaknin	39-102-39	7/19/2020

## אילוח 3

התאריך שבו הנהג מתחיל לעבוד לא יכול להיות יותר מאוחר מאשר התאריך כיום, כלומר בעתיד. השתמשתי פה בטריגר כי אי אפשר להשתמש באילוח, מכיוון שהשימוש בזמן זה משהו לא דטרמינסטי, ולא ניתן לעשות את עם אילוח.

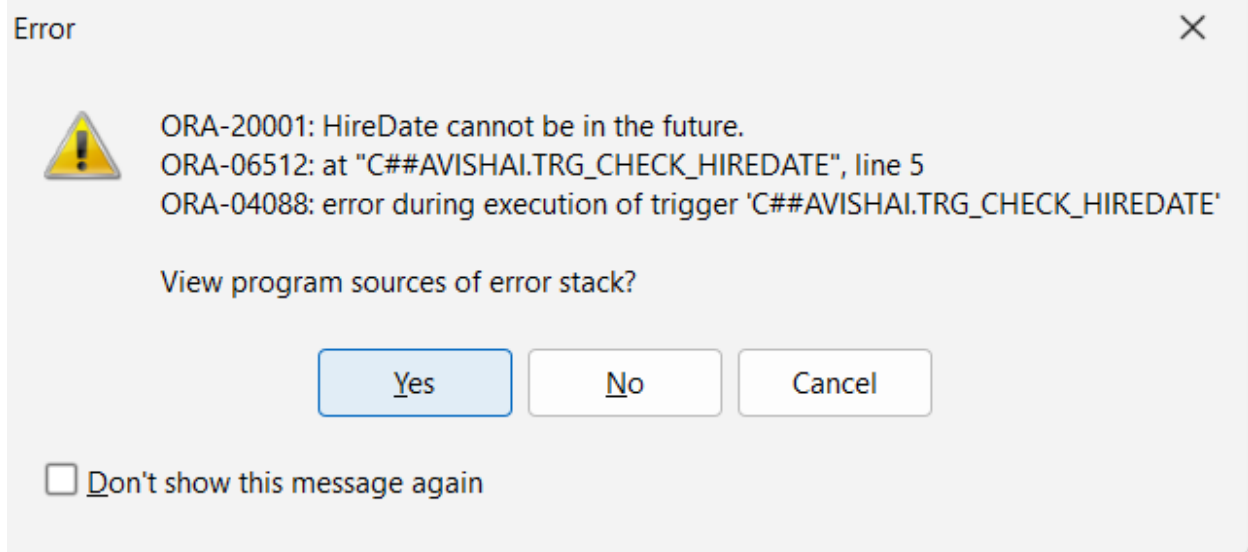
```
-- Constraint 3: check that HireDate is not in the future
```

```
-- i used here trigger, because SYSDATE is non-deterministic, and it can't
be done with constraint
CREATE OR REPLACE TRIGGER trg_check_hiredate
BEFORE INSERT OR UPDATE ON Driver
FOR EACH ROW
DECLARE
    cur_date DATE := TRUNC(SYSDATE);
BEGIN
    IF :NEW.HireDate > cur_date THEN
        RAISE_APPLICATION_ERROR(-20001, 'HireDate cannot be in the
future.');
```

**תיאור השינוי:** יצרנו טריגר לפני הכנסה או עדכון בטבלה Driver שבודק שה-HireDate אינו בעתיד מבחינת ה-SYSDATE.

**ניסיון להכניס נתונים שמפרים את האילוץ:** ננסה להכניס נתונים שמפרים את האילוץ על HireDate שאמור להיות בעתיד.

```
-- insert data that violates constraint 3
INSERT INTO Driver (DriverID, FullName, LicenseNumber, HireDate)
VALUES (459, 'John Smith', '69-291-39', SYSDATE + 1);
```



## חלק ג

### פונקציות

#### פונקציה 1

פונקציה שמקבלת שני תאריכים, ומחזירה את כל הנהגים (Drivers) שהתחילו לעבוד בחברה בין התאריכים הנ"ל, כלומר,  $\text{inputDate1} > \text{HireDate} > \text{inputDate2}$ . הפונקציה תחזיר ref cursor ל ID של כל הנהגים האלו. במידה והתאריכים לא תקינים (התאריך הראשון גדול מהשני), הפונקציה תזרוק חריגה.

```
CREATE OR REPLACE FUNCTION getDriversBetweenDates (
    inputDate1 DATE,
    inputDate2 DATE
) RETURN SYS_REFCURSOR IS
    ref_cur SYS_REFCURSOR;
BEGIN
    -- Check if input dates are valid
    IF inputDate1 < inputDate2 THEN
        -- Open the ref cursor for drivers hired between inputDate2 and
        inputDate1
        OPEN ref_cur FOR
            SELECT DriverID
            FROM Driver
            WHERE HireDate < inputDate2 AND HireDate > inputDate1;

        -- Return the ref cursor
        RETURN ref_cur;
    ELSE
        -- Throw an exception if input dates are invalid
        RAISE_APPLICATION_ERROR(-20001, 'Invalid input dates: inputDate1
        should be greater than inputDate2');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        -- Handle any exceptions and ensure ref cursor is closed
        IF (ref_cur IS NOT NULL) THEN
            CLOSE ref_cur;
        END IF;
        -- Re-raise the exception for the caller
        RAISE;
```

```
END getDriversBetweenDates;
/
```

## פונקציה 2

פונקציה שמקבלת רשימה של לוחיות רישוי ותחזיר רשימה של כל הנהגים (Drivers) שנוהגים על המוניות או האוטובוסים עם לוחיות הרישוי האלו. הפונקציה תחזיר רשומה של DriverIDs.

```
CREATE OR REPLACE FUNCTION getDriversByLicensePlates (
    license_plates_list VARCHAR2
) RETURN SYS_REFCURSOR IS
    ref_cur SYS_REFCURSOR;
BEGIN
    -- Open the ref cursor for drivers with given license plates
    OPEN ref_cur FOR
        SELECT DISTINCT DriverID
        FROM DrivesInTaxi
        WHERE TaxiID IN (
            SELECT TaxiID
            FROM Taxi
            WHERE LicensePlate IN (
                SELECT trim(regexp_substr(license_plates_list, '[^,]+', 1,
LEVEL)) license_plate
                FROM dual
                CONNECT BY LEVEL <= regexp_count(license_plates_list, ',')
+ 1
            )
        )
        UNION
        SELECT DISTINCT DriverID
        FROM BusRide
        WHERE BusID IN (
            SELECT BusID
            FROM Bus
            WHERE LicensePlate IN (
                SELECT trim(regexp_substr(license_plates_list, '[^,]+', 1,
LEVEL)) license_plate
                FROM dual
                CONNECT BY LEVEL <= regexp_count(license_plates_list, ',')
+ 1
            )
        )
```

```

);

-- Return the ref cursor
RETURN ref_cur;
EXCEPTION
    WHEN OTHERS THEN
        -- Handle any exceptions and ensure ref cursor is closed
        IF (ref_cur IS NOT NULL) THEN
            CLOSE ref_cur;
        END IF;
        -- Re-raise the exception for the caller
        RAISE;
END getDriversByLicensePlates;
/

```

## פרוצדורות

### פרוצדורה 1

פרוצדורה שמקבלת ref cursor לרשימת IDים של נהגים, מדפיסה את השמות של כל הנהגים הנ"ל ומוחקת אותם ממאגר המידע. במידה והיא מקבלת רשימה ריקה, היא תזרוק חריגה.

```

CREATE OR REPLACE PROCEDURE printAndDeleteDrivers(
    ref_cur IN SYS_REFCURSOR
) IS
    driver_id Driver.DriverID%TYPE;
    driver_name Driver.FullName%TYPE;
BEGIN
    -- Check if ref cursor is valid
    IF ref_cur IS NULL THEN
        RAISE_APPLICATION_ERROR(-20002, 'Input ref cursor is NULL');
    END IF;

    LOOP
        FETCH ref_cur INTO driver_id;
        EXIT WHEN ref_cur%NOTFOUND;

        -- Get driver name
        SELECT FullName INTO driver_name
        FROM Driver

```



```

WHERE DriverID = driver_id;

-- Print driver name
DBMS_OUTPUT.PUT_LINE('Deleting driver: ' || driver_name);

-- Delete instances from table BusRide where driverID = driverID
DELETE FROM BusRide
WHERE DriverID = driver_id;

-- Delete instances from table Drives in taxi where driverID =
driverID
DELETE FROM DrivesInTaxi;
WHERE DriverID = driver_id;

-- Delete driver from Driver table
DELETE FROM Driver
WHERE DriverID = driver_id;
END LOOP;

CLOSE ref_cur;
EXCEPTION
    WHEN OTHERS THEN
        -- Handle any exceptions and ensure ref cursor is closed
        IF (ref_cur IS NOT NULL) THEN
            CLOSE ref_cur;
        END IF;
        -- Re-raise the exception for the caller
        RAISE;
END printAndDeleteDrivers;
/

```

## פרוצדורה 2

פרוצדורה שמקבלת רשימה של נהגים (ID) ומוחקת את כל הנהגים שמספר הרישיון שלהם הוא לא תקין (הנוסחא היא סכימת כל הספרות, חוץ מהאחרונה, לעשות מודולו 10 עליה, ולבדוק אם זה שווה לספרה האחרונה). הפרוצדורה למעשה מורכבת מפרוצדורה שמשתמשת בפרדיקט שבדק אם לוחית הרישוי היא תקינה. במידה והפרוצדורה מקבלת ID של נהג שלא קיים במערכת, היא תזרוק חריגה.

```

CREATE OR REPLACE FUNCTION isValidLicenseNumber(license_number IN
VARCHAR2)
RETURN BOOLEAN
IS
    v_length NUMBER;
    v_sum NUMBER := 0;
    v_last_digit NUMBER;
BEGIN
    -- Check if the license number is exactly 14 characters long
    (including hyphens)
    v_length := LENGTH(license_number);
    IF v_length != 14 THEN
        RETURN FALSE;
    END IF;

    -- Calculate the sum of the digits excluding the hyphens
    FOR i IN 1..13 LOOP
        IF SUBSTR(license_number, i, 1) != '-' THEN
            v_sum := v_sum + TO_NUMBER(SUBSTR(license_number, i, 1));
        END IF;
    END LOOP;

    -- Get the last digit
    v_last_digit := TO_NUMBER(SUBSTR(license_number, 14, 1));

    -- Check if the last digit is equal to the sum of the other digits mod
    10
    IF v_last_digit = MOD(v_sum, 10) THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RETURN FALSE;
END isValidLicenseNumber;
/

CREATE OR REPLACE PROCEDURE deleteDriversWithInvalidLicenses(

```

```

        driver_ids_list VARCHAR2
    ) IS
        v_driver_id_list VARCHAR2(4000); -- Variable to hold comma-separated
list of DriverIDs
        v_driver_id Driver.DriverID%TYPE;
        v_driver_name Driver.FullName%TYPE;
BEGIN
    -- Check if input driver_ids_list is empty
    IF driver_ids_list IS NULL OR driver_ids_list = '' THEN
        RAISE_APPLICATION_ERROR(-20003, 'Input driver IDs list is empty');
    END IF;

    -- Delete drivers with invalid license numbers
    FOR driver_rec IN (
        SELECT trim(regexp_substr(driver_ids_list, '[^,]+', 1, LEVEL))
driver_id
        FROM dual
        CONNECT BY LEVEL <= regexp_count(driver_ids_list, ',') + 1
    ) LOOP
        v_driver_id := TO_NUMBER(driver_rec.driver_id);

        -- Check if driver exists
        SELECT COUNT(*)
        INTO v_driver_id_list
        FROM Driver
        WHERE DriverID = v_driver_id;

        IF v_driver_id_list > 0 THEN
            -- Validate license number and delete driver if invalid
            IF NOT isValidLicenseNumber(v_driver_id) THEN
                -- Get driver name
                SELECT FullName INTO v_driver_name
                FROM Driver
                WHERE DriverID = v_driver_id;

                -- Delete dependent records first (if needed)
                DELETE FROM DrivesInTaxi WHERE DriverID = v_driver_id;
                DELETE FROM BusRide WHERE DriverID = v_driver_id;

                -- Print driver name (optional for debugging)

```

```

        DBMS_OUTPUT.PUT_LINE('Deleting driver: ' ||
v_driver_name);

        -- Delete driver from Driver table
        DELETE FROM Driver
        WHERE DriverID = v_driver_id;
    END IF;
ELSE
    -- Throw exception if driver does not exist
    RAISE_APPLICATION_ERROR(-20004, 'Driver with ID ' ||
v_driver_id || ' does not exist');
    END IF;
END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        NULL; -- Handle no data found scenario
    WHEN OTHERS THEN
        -- Handle any exceptions
        RAISE;
END deleteDriversWithInvalidLicenses;
/

```

## תכניות ראשיות

### תכנית ראשית 1

תכנית שמבקשת תאריך מהמשתמש, ומוחקת את כל הנהגים שהתחילו לעבוד בתאריך זה, עד לתאריך הנוכחי. התכנית תשלח את הקלט מהמשתמש + התאריך הנוכחי אל פונקציה 1, והפלט של פונקציה 1 יעבור על פרוצדורה 1.

```

DECLARE
    input_date DATE := TO_DATE('2023-01-01', 'YYYY-MM-DD');
    current_date DATE := SYSDATE;
    ref_cur SYS_REFCURSOR;
    driver_id INT;
BEGIN
    input_date := TO_DATE('&input_date', 'YYYY-MM-DD');

    -- Call Function 1 to get drivers hired between input_date and
current_date
    ref_cur := getDriversBetweenDates(input_date, current_date);

```

```
printAndDeleteDrivers(ref_cur);

-- Close the cursor after fetching
CLOSE ref_cur;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
```

Variables

Name	Value
input_date	2023-10-10

SQL Output Statistics

Clear Buffer size 10000 ☒ Enabled

Deleting driver: Mack Degue  
 Deleting driver: Natal Bein  
 Deleting driver: Sharyl Schoenrock  
 Deleting driver: Michele Dicey  
 Deleting driver: Tess Perford  
 Deleting driver: Cornelia Downgate  
 Deleting driver: Katrinka Janks  
 Deleting driver: Nealon Gladbeck

דוגמת הרצה:

Variables

Name	Value
input_date	2025-10-10

SQL Output Statistics

Clear Buffer size 10000 ☒ Enabled

Error: ORA-20001: Invalid input dates: inputDate1 should be greater than inputDate2

OK Cancel Clear

דוגמת הרצה שכוללת חריגה:

## תכנית ראשית 2

תכנית שתבקש רשימה של לוחיות רישוי מהמשתמש, ותמחוק את כל הנהגים שמספר הרישוי שלהם הוא לא תקני. התכנית תשלח את הקלט מהמשתמש אל פונקציה 2, והפלט של פונקציה 2 יעבור אל פרוצדורה 2.

```
DECLARE
    license_plates_list VARCHAR2(200); -- Declare variable to hold license
plates list
    ref_cur SYS_REFCURSOR;
    driver_id INT;
    driver_ids_list VARCHAR2(4000); -- Variable to hold comma-separated
list of DriverIDs

BEGIN
    -- Prompt user for input
    license_plates_list := '&Enter_license_plates_list';

    -- Initialize driver_ids_list
    driver_ids_list := '';

    -- Call Function 2 to get drivers driving taxis or buses with given
license plates
    ref_cur := getDriversByLicensePlates(license_plates_list);
    LOOP
        FETCH ref_cur INTO driver_id;
        EXIT WHEN ref_cur%NOTFOUND;

        -- Concatenate driver_id to driver_ids_list
        IF driver_ids_list IS NULL THEN
            driver_ids_list := TO_CHAR(driver_id);
        ELSE
            driver_ids_list := driver_ids_list || ',' ||
TO_CHAR(driver_id);
        END IF;
    END LOOP;

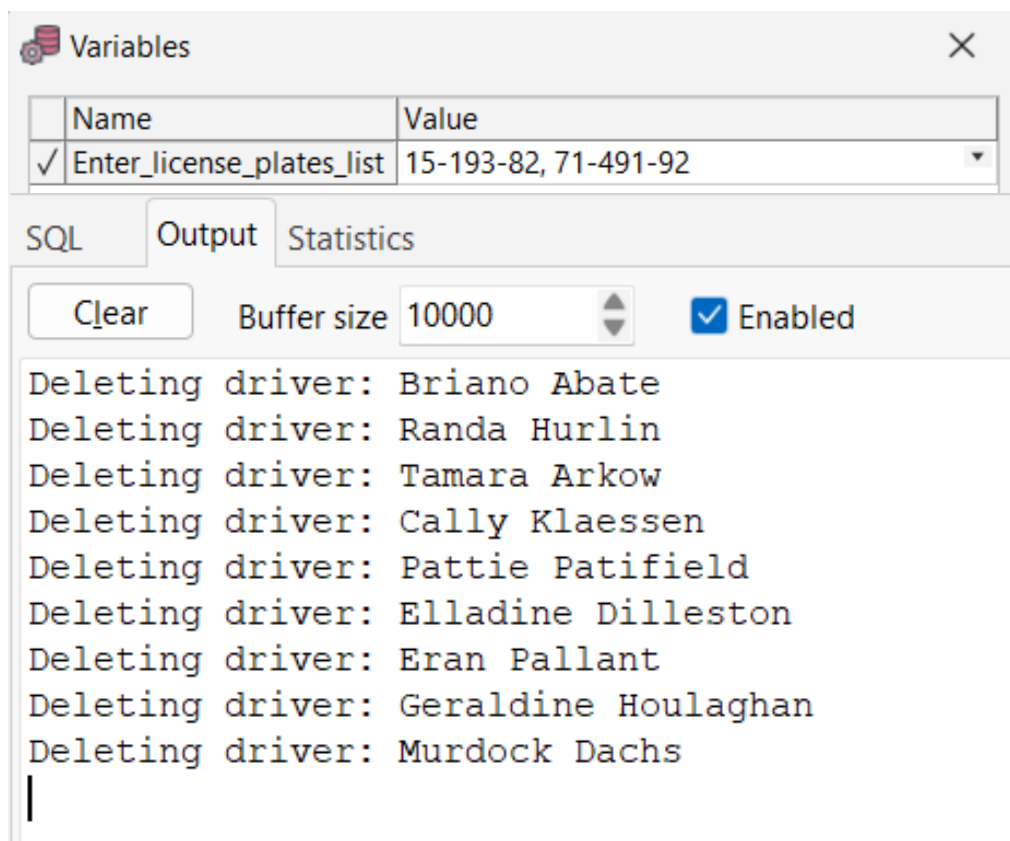
    -- Call Procedure 2 to delete drivers with invalid license plates
deleteDriversWithInvalidLicenses(driver_ids_list);

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

```
END;
```

```
/
```

דוגמת הרצה:



**Variables**

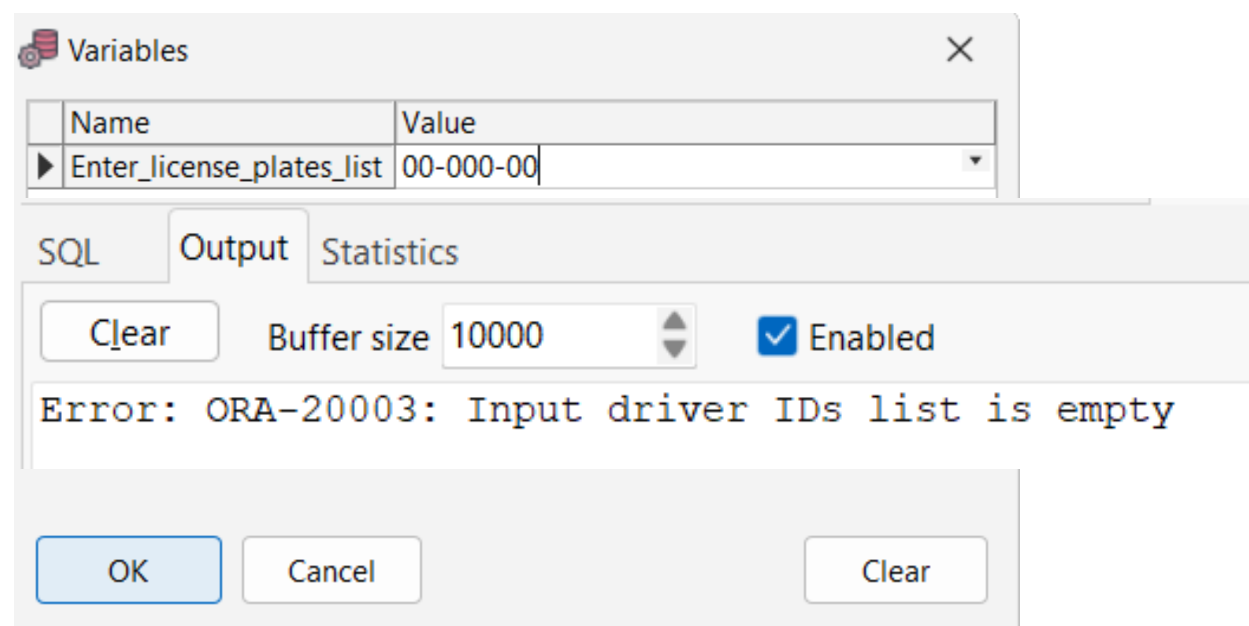
Name	Value
Enter_license_plates_list	15-193-82, 71-491-92

SQL Output Statistics

Clear Buffer size 10000 ☒ Enabled

```
Deleting driver: Briano Abate
Deleting driver: Randa Hurlin
Deleting driver: Tamara Arkow
Deleting driver: Cally Klaessen
Deleting driver: Pattie Patifield
Deleting driver: Elladine Dilleston
Deleting driver: Eran Pallant
Deleting driver: Geraldine Houlaghan
Deleting driver: Murdock Dachs
```

דוגמת הרצה שכוללת חריגה:



**Variables**

Name	Value
Enter_license_plates_list	00-000-00

SQL Output Statistics

Clear Buffer size 10000 ☒ Enabled

```
Error: ORA-20003: Input driver IDs list is empty
```

OK Cancel Clear