

מINI פרויקט **בבסיסי נתונים מכבי- מרפאת שיניים**

הדו זהה:

hadasszoh@gmail.com , 325032670

מيري אורדנטלי:

mariaordent@gmail.com , 213687346

תוכן

4.....	פרויקט מרפאת שניים:
5.....	תרשים ERD
6.....	הסבר תרשימים-
8.....	:DSD
10.....	פקודת: create Table
14.....	פקודות drop:
16.....	צילום מסך של הפעלת פקודה desk:
18.....	קובץ sqls.insertTable
24.....	קובץ sqls.selectAll
26.....	שיטות הכנסת data:
26.....	Data generator
36.....	Generatedata- Website
42.....	Tex importer:
43.....	קובץ גיבוי Backup.sql
45.....	שאליות:
45.....	Select 1:
46.....	Select 2
47.....	Select 3:
48.....	Select 4:
50.....	שאלת מחיקה:
50.....	1
52.....	2
54.....	שאלת update
54.....	update1
56.....	2 update
59.....	שאליות עם פרמטרים
59.....	שאלת 1 עם פרמטר
61.....	שאלת 2 עם פרמטר:
63.....	שאלת 3 עם פרמטר:
64.....	שאלת 4 עם פרמטר:
66.....	אלוצים:

66.....	1.
66.....	2.
67.....	3
68.....	4
70.....	פרוצדורה 1
73.....	פונקציה 1
77.....	תוכנית ראשית 1
79.....	פרוצדורה 2
84.....	פונקציה 2
87.....	תוכנית ראשית 2
90.....	שלב 4
90.....	DSD אגף חדש
91.....	ERD אגף חדש
92.....	ERD מאוחד
93.....	העברה הנתונים
93.....	העברה נתוני הסטודנט אל המטופל:
97.....	קשר בין סטודנט לשכר לימוד
99.....	העברה נתונים מ stuff teacher ל-
102.....	VIEW אגף בית ספר
103.....	שאילתות על VIEW אגף בית ספר
105.....	View מרפאת שניים:
106.....	שאילתת 1;
106.....	שאילתת 2:

פרויקט מרפאת שניים:

מרפאת שניים רב תחומית, עצשוית, המזינה לטיפולים מגוון טיפולים דנטליים מתקדמים מכל הסוגים, תחת קורת גג אחד, תוך הקפדה יתרה על עבודתה בתנאים סטריליים מחמירים, טיפול באויריה רגועה ומתן יחס אישי לכל לטיפול.

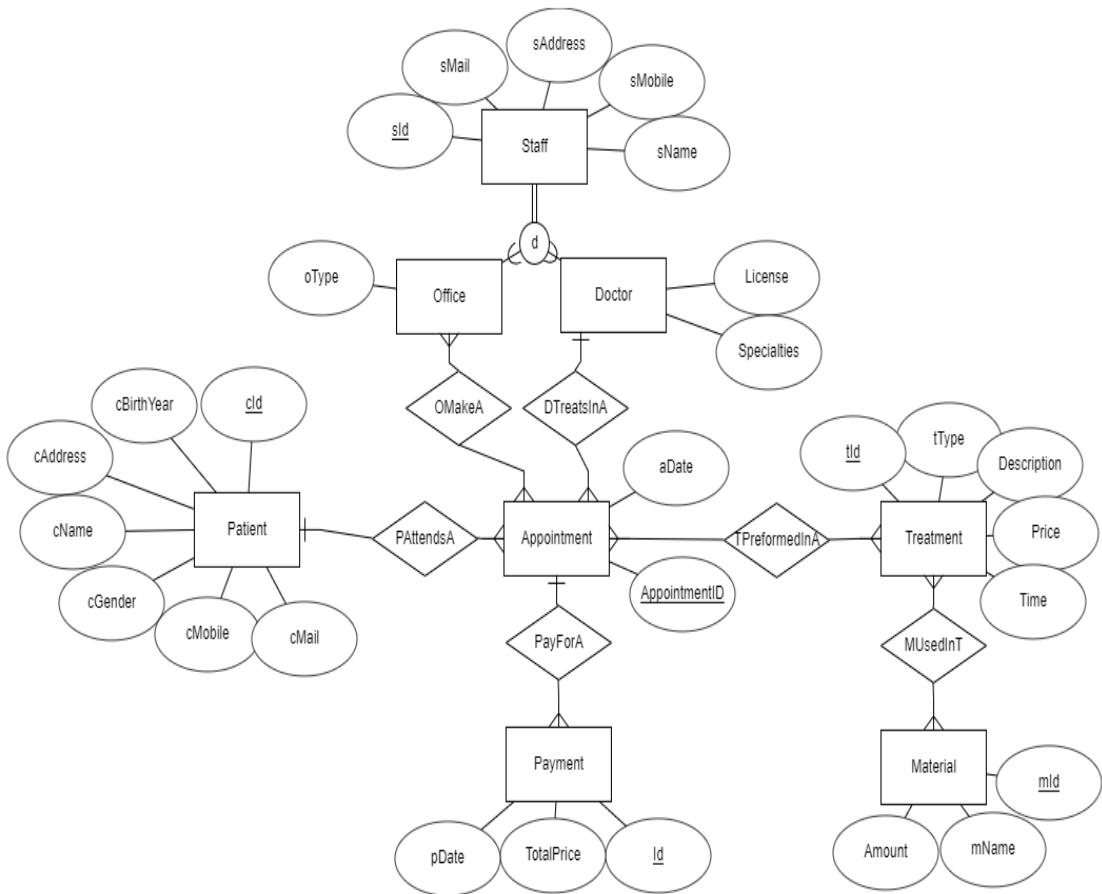
המרפאה כוללת צוות מקצועי מהשרה הראשונה, עם וותק וניסיון רב, כולל רופאים מומחים לרפואה ושיקום הפה, כירורגיה יישור שניים (אורטודונטיה), רפואת חניכיים (פריזודונטיה), טיפול שורש ורפואת שניים לילדים וכן יועצים מומחים מהטוביים בתחומים, שניניות וסיעות וכלל מtan מענה לטיפולן חירום כאשריות הטיפול והרמה המקצועית עומדות מעל לכל.

הישויות הראשיות- צוות- צוות המרפאה שמתחלק לרופאים ולצוות המשרד ל��חות- הלוקחות הבאים למרפאה על מנת לקבוע תור ולקבל טיפול

הפעולות בתחום המרפאה-

צוות המשרד יכול לקבוע ולתאם תורים עבור לקוחות, לכל תור משיר רפואי וכן נקבעים סוג הטיפולים שייתבצעו בו, ניתן לבצע כמה בתור אחד. עבור כל תור ניתן לשלם בכמה תשלום אחד וכן ניתן לשלם עבור כמה תורים בתשלום אחד. עבור כל טיפול נקבע באילו חומרים ישמשו

תרשים ERD



הסבר תרשימים-

פירוט ישויות:

Staff - טבלת צוות המרפאה:

ת.ז.	sID
כתובת	sAddress
טלפון	sMobile
שם	sName
מייל	sMail

Doctor - Doctor Table:

רישיון רופא	license
סוג התמחות	specialties

Office:

Type - Type of Work in the Ministry

Treatment - Treatment Table:

מספר זהוי של הטיפול	tID
סוג הטיפול	tType
תיאור הטיפול	Description
מחיר	Price
זמן הטיפול	Time

Appointment - Appointment Table:

מספר זהוי לתור	AppointmentId
Date of appointment	aDate

Patient - Patient Table:

ת.ז. ליקוי	cID
שנת לידיה של הליקוי	CBirth Year
כתובת הליקוי	cAddress
שם ליקוי	cName
מין הליקוי	cGender
טלפון הליקוי	cMobile
מייל הליקוי	cMail

ק- טבלת התשלום-Payment:

מספר זהוי של העסקה	ID
מחיר כולל	TotalPrice
תאריך העסקה	pDate

טבלת האספקה-Material:

תעודת זהוי של האספקה	mID
שם מוצר האספקה	mName
כמות האספקה	Amount

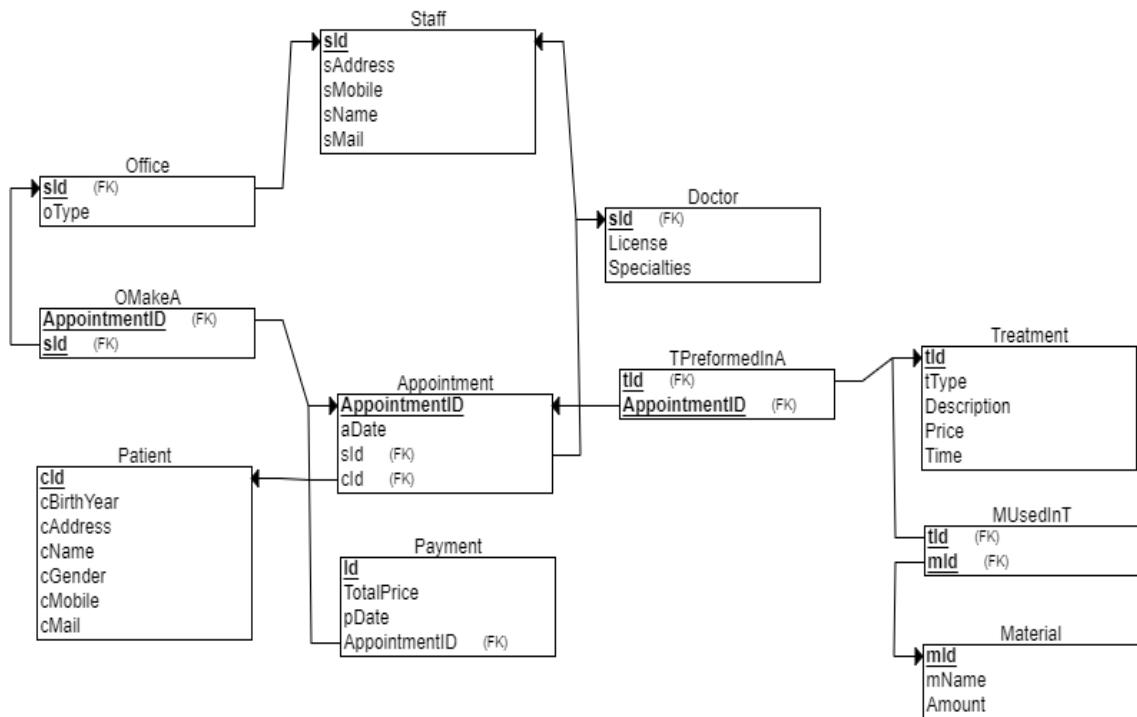
הערות כלליות-

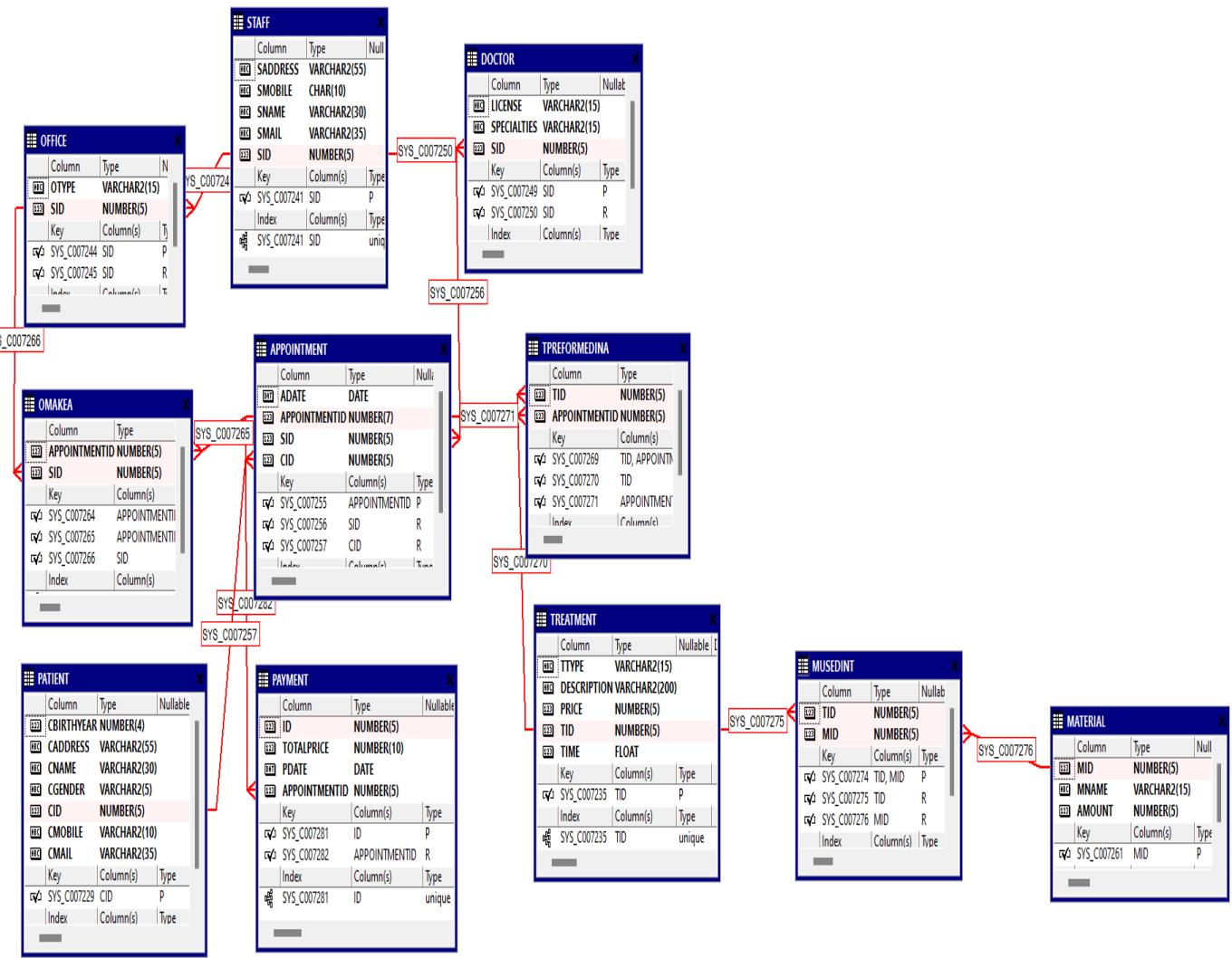
נשים לב כי מכיוון שיש ירושה לכל אדם השיר Doctor Office בפועל יש תוכנות נוספות כי הוא שיר גם ל'uff Staff (ובכך עונה על התנאי של מינימום 3 תוכנות. באישור המרצה)

הערות מבניות-

הERD מנוורמלת לפי NF3 כי קודם כל אין תלויות פונקציונליות לא טריוויאליות בין המפתחות המועמדים למאפיינים שאינם מפתח בתוך כל ישות. כל מאפיין שאינו מפתח תלוי במלואו במפתח המועמד. אין תלויות מעבריות בין מאפיינים שאינם מפתח בתוך כל ישות. כל המאפיינים שאינם מפתח תלויים ישירות במפתח המועמד. היחסות מחולקות כראוי ליחידות נפרדות, ואין קבוצות חוזרות של מאפיינים.

:DSD





פקולת : create Table

CREATE TABLE Patient

)

,cBirthYear NUMERIC(4) NOT NULL

,cAddress VARCHAR(55) NOT NULL

,cName VARCHAR(30) NOT NULL

,cGender VARCHAR(5) NOT NULL

,cId NUMERIC(5) NOT NULL

,cMobile VARCHAR(10) NOT NULL

,cMail VARCHAR(35) NOT NULL

PRIMARY KEY (cId)

:)

CREATE TABLE Treatment

)

,tType VARCHAR(15) NOT NULL

,Description VARCHAR(200) NOT NULL

,Price NUMERIC(5) NOT NULL

,tId NUMERIC(5) NOT NULL

,Time FLOAT NOT NULL

PRIMARY KEY (tId)

:)

CREATE TABLE Staff

)

,sAddress VARCHAR(55) NOT NULL

,sMobile CHAR(10) NOT NULL

,sName VARCHAR(30) NOT NULL

,sMail VARCHAR(35) NOT NULL

,sId NUMERIC(5) NOT NULL

PRIMARY KEY (sId)

:1

CREATE TABLE Office

)

,oType VARCHAR(15) NOT NULL

,sId NUMERIC(5) NOT NULL

PRIMARY KEY (sId)

FOREIGN KEY (sId) REFERENCES Staff(sId)

:1

CREATE TABLE Doctor

)

,License VARCHAR(15) NOT NULL

,Specialties VARCHAR(15) NOT NULL

,sId NUMERIC(5) NOT NULL

PRIMARY KEY (sId)

FOREIGN KEY (sId) REFERENCES Staff(sId)

:1

CREATE TABLE Appointment

)

,aDate DATE NOT NULL

,AppointmentID NUMERIC(7) NOT NULL

,sId NUMERIC(5) NOT NULL

,cId NUMERIC(5) NOT NULL

,PRIMARY KEY (AppointmentID)
,FOREIGN KEY (sId) REFERENCES Doctor(sId)
FOREIGN KEY (cId) REFERENCES Patient(cId)

:1

CREATE TABLE Material

)

,mId NUMERIC(5) NOT NULL
,mName VARCHAR(15) NOT NULL
,Amount NUMERIC(5) NOT NULL
PRIMARY KEY (mId)

:1

CREATE TABLE OMakeA

)

,AppointmentID NUMERIC(5) NOT NULL
,sId NUMERIC(5) NOT NULL
,PRIMARY KEY (AppointmentID, sId)
,FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID)
FOREIGN KEY (sId) REFERENCES Office(sId)

:1

CREATE TABLE TPreformedInA

)

,tId NUMERIC(5) NOT NULL
,AppointmentID NUMERIC(5) NOT NULL
,PRIMARY KEY (tId, AppointmentID)
,FOREIGN KEY (tId) REFERENCES Treatment(tId)

FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID)

:1

CREATE TABLE MUsedInT

)

,tId NUMERIC(5) NOT NULL

,mId NUMERIC(5) NOT NULL

,PRIMARY KEY (tId, mId)

,FOREIGN KEY (tId) REFERENCES Treatment(tId)

FOREIGN KEY (mId) REFERENCES Material(mId)

:1

CREATE TABLE Payment

)

,Id NUMERIC(5) NOT NULL

,TotalPrice NUMERIC(10) NOT NULL

,pDate DATE NOT NULL

,AppointmentID NUMERIC(5) NOT NULL

,PRIMARY KEY (Id)

FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID)

:1

פקודות :drop

-- Drop the Payment table first

```
DROP TABLE Payment;
```

-- Documentation: The Payment table is dropped first as per the specified order.

-- Drop the MUsedInT table next

```
DROP TABLE MUsedInT;
```

-- Documentation: The MUsedInT table is dropped next as per the specified order.

-- Drop the PreformedIn table

```
DROP TABLE Tpreformedina;
```

-- Documentation: The PreformedIn table is dropped as per the specified order.

-- Drop the OMakeA table

```
DROP TABLE OMakeA;
```

-- Documentation: The OMakeA table is dropped as per the specified order.

-- Drop the Treatment table

```
DROP TABLE Treatment;
```

-- Documentation: The Treatment table is dropped as per the specified order.

-- Drop the Material table

DROP TABLE Material;

-- Documentation: The Material table is dropped as per the specified order.

-- Drop the Appointment table

DROP TABLE Appointment;

-- Documentation: The Appointment table is dropped as per the specified order.

-- Drop the Patient table

DROP TABLE Patient;

-- Documentation: The Patient table is dropped as per the specified order.

-- Drop the Office table

DROP TABLE Office;

-- Documentation: The Office table is dropped as per the specified order.

-- Drop the Doctor table

DROP TABLE Doctor;

-- Documentation: The Doctor table is dropped as per the specified order.

-- Drop the Staff table last

DROP TABLE Staff;

-- Documentation: The Staff table is dropped last as per the specified order.

צילום מסך של הפעלה פקודה :desk

```
SQL> desc musedint
Name Type      Nullable Default Comments
----- -----
TID  NUMBER(5)
MID  NUMBER(5)

SQL> desc tpreformedina
Name      Type      Nullable Default Comments
----- -----
TID       NUMBER(5)
APPOINTMENTID NUMBER(5)

SQL> desc patient
Name      Type      Nullable Default Comments
----- -----
CBIRTHYEAR NUMBER(4)
CADDRESS   VARCHAR2(55)
CNAME     VARCHAR2(30)
CGENDER    VARCHAR2(5)
CID        NUMBER(5)
CMOBILE   VARCHAR2(10)
CMAIL     VARCHAR2(35)

SQL> desc omakea
Name      Type      Nullable Default Comments
----- -----
APPOINTMENTID NUMBER(5)
SID       NUMBER(5)

SQL> desc payment
Name      Type      Nullable Default Comments
----- -----
ID        NUMBER(5)
TOTALPRICE NUMBER(10)
PDATE    DATE
APPOINTMENTID NUMBER(5)
```

```
SQL> desc treatment
Name      Type      Nullable Default Comments
-----
TTYPE     VARCHAR2(15)
DESCRIPTION VARCHAR2(200)
PRICE     NUMBER(5)
TID       NUMBER(5)
TIME      NUMBER

SQL> desc appointment
Name      Type      Nullable Default Comments
-----
ADATE    DATE
APPOINTMENTID NUMBER(7)
SID      NUMBER(5)
CID      NUMBER(5)

SQL> desc doctor
Name      Type      Nullable Default Comments
-----
LICENSE   VARCHAR2(15)
SPECIALTIES VARCHAR2(15)
SID      NUMBER(5)

SQL> desc staff
Name      Type      Nullable Default Comments
-----
SADDRESS  VARCHAR2(55)
SMOBILE   CHAR(10)
SNAME    VARCHAR2(30)
SMAIL    VARCHAR2(35)
SID      NUMBER(5)

SQL> desc office
Name  Type      Nullable Default Comments
-----
OTYPE   VARCHAR2(15)
SID    NUMBER(5)

SQL> desc material
Name  Type      Nullable Default Comments
-----
MID    NUMBER(5)
MNAME  VARCHAR2(15)
AMOUNT NUMBER(5)
```

קובץ :insertTable.sql

-- Insert data into Staff table

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('123 Main St', '123456789', 'John Doe', 'john@email.com', 10001);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('456 Oak Ave', '987654321', 'Jane Smith', 'jane@email.com', 10002);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('789 Elm St', '456789012', 'Michael Johnson', 'michael@email.com', 10003);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('321 Pine Rd', '789012345', 'Emily Davis', 'emily@email.com', 10004);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('654 Maple Ln', '345678901', 'David Wilson', 'david@email.com', 10005);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('987 Cedar Blvd', '678901234', 'Sarah Thompson', 'sarah@email.com', 10006);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('246 Oak Ct', '901234567', 'Robert Anderson', 'robert@email.com', 10007);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('579 Elm Way', '234567890', 'Jessica Taylor', 'jessica@email.com', 10008);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('813 Pine Ave', '567890123', 'Chris Brown', 'chris@email.com', 10009);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('159 Maple St', '890123456', 'Amanda Garcia', 'amanda@email.com', 10010);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('101 Birch St', '234123456', 'Oliver Martin', 'oliver@email.com', 10011);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('202 Spruce Ln', '567234567', 'Sophia Lee', 'sophia@email.com', 10012);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('303 Cherry Ave', '890345678', 'Liam Martinez', 'liam@email.com', 10013);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('404 Willow Rd', '123456789', 'Mia Rodriguez', 'mia@email.com', 10014);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('505 Poplar St', '234567890', 'Noah Davis', 'noah@email.com', 10015);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('606 Aspen Ct', '345678901', 'Isabella Lewis', 'isabella@email.com', 10016);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('707 Beech Blvd', '456789012', 'James White', 'james@email.com', 10017);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('88 Magnolia Way', '567890123', 'Charl Walker', 'charlotte@email.com', 10018);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('909 Oak Dr', '678901234', 'Ben Hall', 'benjamin@email.com', 10019);

INSERT INTO Staff (sAddress, sMobile, sName, sMail, sId) VALUES ('100 Redwood Ave', '789012345', 'Amelia Young', 'amelia@email.com', 10020);

--select * from Staff;

-- Insert data into Doctor table

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L1234', 'Orthodontist', 10001);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L5678', 'Cosmetic', 10002);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L9012', 'Dentist', 10003);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L3456', 'Dentist', 10004);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L7890', 'Cosmetic', 10005);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L2345', 'Dentist', 10006);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L6789', 'Dentist', 10007);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L0123', 'Orthodontist', 10008);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L4567', 'Orthodontist', 10009);

INSERT INTO Doctor (License, Specialties, sId) VALUES ('L8901', 'Dentist', 10010);

--select * from Doctor;

-- Insert data into Office table

```
INSERT INTO Office (oType, sId) VALUES ('Secretariat', 10011);
INSERT INTO Office (oType, sId) VALUES ('Sales', 10012);
INSERT INTO Office (oType, sId) VALUES ('Secretariat', 10013);
INSERT INTO Office (oType, sId) VALUES ('Sales', 10014);
INSERT INTO Office (oType, sId) VALUES ('Secretariat', 10015);
INSERT INTO Office (oType, sId) VALUES ('Secretariat', 10016);
INSERT INTO Office (oType, sId) VALUES ('Sales', 10017);
INSERT INTO Office (oType, sId) VALUES ('Shift Manager', 10018);
INSERT INTO Office (oType, sId) VALUES ('Shift Manager', 10019);
INSERT INTO Office (oType, sId) VALUES ('Sales', 10020);
```

--select * from Office;

-- Insert data into Patient table

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile,
cMail) VALUES (1980, '123 Main St', 'Alice Johnson', 'F', 20001, '123456789',
'alice@email.com');

INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile,
cMail) VALUES (1990, '456 Oak Ave', 'Bob Smith', 'M', 20002, '987654321',
'bob@email.com');

INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile,
cMail) VALUES (1985, '789 Elm St', 'Charlie Davis', 'M', 20003, '456789012',
'charlie@email.com');

INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile,
cMail) VALUES (1995, '321 Pine Rd', 'Danielle Wilson', 'F', 20004, '789012345',
'danielle@email.com');

INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile,
cMail) VALUES (1988, '654 Maple Ln', 'Evan Thompson', 'M', 20005, '345678901',
'evan@email.com');
```

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cld, cMobile,  
cMail) VALUES (1992, '987 Cedar Blvd', 'Fiona Anderson', 'F', 20006, '678901234',  
'fiona@email.com');
```

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cld, cMobile,  
cMail) VALUES (1982, '246 Oak Ct', 'George Taylor', 'M', 20007, '901234567',  
'george@email.com');
```

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cld, cMobile,  
cMail) VALUES (1998, '579 Elm Way', 'Hannah Brown', 'F', 20008, '234567890',  
'hannah@email.com');
```

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cld, cMobile,  
cMail) VALUES (1991, '813 Pine Ave', 'Ian Garcia', 'M', 20009, '567890123',  
'ian@email.com');
```

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cld, cMobile,  
cMail) VALUES (1987, '159 Maple St', 'Jill Roberts', 'F', 20010, '890123456',  
'jill@email.com');
```

--select * from Patient;

-- Insert data into Appointment table

```
INSERT INTO Appointment (aDate, AppointmentID, sld, cld) VALUES  
(TO_DATE('01-05-2023', 'DD-MM-YYYY'), 30001, 10001, 20001);
```

```
INSERT INTO Appointment (aDate, AppointmentID, sld, cld) VALUES  
(TO_DATE('02-05-2023', 'DD-MM-YYYY'), 30002, 10006, 20002);
```

```
INSERT INTO Appointment (aDate, AppointmentID, sld, cld) VALUES  
(TO_DATE('03-05-2023', 'DD-MM-YYYY'), 30003, 10006, 20003);
```

```
INSERT INTO Appointment (aDate, AppointmentID, sld, cld) VALUES  
(TO_DATE('04-05-2023', 'DD-MM-YYYY'), 30004, 10001, 20004);
```

```
INSERT INTO Appointment (aDate, AppointmentID, sld, cld) VALUES  
(TO_DATE('05-05-2023', 'DD-MM-YYYY'), 30005, 10005, 20005);
```

```
INSERT INTO Appointment (aDate, AppointmentID, sld, cld) VALUES  
(TO_DATE('06-05-2023', 'DD-MM-YYYY'), 30006, 10006, 20006);
```

--select * from Appointment;

-- Insert data into Payment table

INSERT INTO Payment (Id, TotalPrice, pDate, AppointmentID) VALUES (40001,
100.00, TO_DATE('01-05-2023', 'DD-MM-YYYY'), 30001);

INSERT INTO Payment (Id, TotalPrice, pDate, AppointmentID) VALUES (40002,
150.50, TO_DATE('02-05-2023', 'DD-MM-YYYY'), 30002);

INSERT INTO Payment (Id, TotalPrice, pDate, AppointmentID) VALUES (40003,
75.25, TO_DATE('03-05-2023', 'DD-MM-YYYY'), 30003);

INSERT INTO Payment (Id, TotalPrice, pDate, AppointmentID) VALUES (40004,
200.00, TO_DATE('04-05-2023', 'DD-MM-YYYY'), 30004);

INSERT INTO Payment (Id, TotalPrice, pDate, AppointmentID) VALUES (40005,
125.75, TO_DATE('05-05-2023', 'DD-MM-YYYY'), 30005);

INSERT INTO Payment (Id, TotalPrice, pDate, AppointmentID) VALUES (40006,
175.50, TO_DATE('06-05-2023', 'DD-MM-YYYY'), 30006);

--select * from Payment;

-- Insert data into Material table

INSERT INTO Material (mId, mName, Amount) VALUES (50001,'Gauze',100);

INSERT INTO Material (mId, mName, Amount) VALUES (50002,'Bandages',75);

INSERT INTO Material (mId, mName, Amount) VALUES (50003,'Syringes',200);

INSERT INTO Material (mId, mName, Amount) VALUES (50004,'Scalpels',50);

INSERT INTO Material (mId, mName, Amount) VALUES (50005,'Gloves',300);

INSERT INTO Material (mId, mName, Amount) VALUES (50006,'Sutures',125);

INSERT INTO Material (mId, mName, Amount) VALUES (50007,'Cotton
Swabs',400);

INSERT INTO Material (mId, mName, Amount) VALUES (50008,'Disinfectant',150);

INSERT INTO Material (mId, mName, Amount) VALUES
(50009,'Thermometers',75);

INSERT INTO Material (mId, mName, Amount) VALUES
(50010,'Stethoscopes',25);

--select * from Material;

-- Insert data into Treatment table

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Surgery','Appendectomy',10000,60001,2.5);

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Medication','Antibiotics',50,60002,0.5);

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Therapy','Physical Therapy',100,60003,1.0);

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Procedure','X-Ray',200,60004,0.25);

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Surgery','Knee Replacement',15000,60005,3.0);

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Medication','Pain Relievers',25,60006,0.25);

--select * from Treatment;

קובץ :SelectAll.sql

-- Select all from Doctor table

Select * from Doctor; -- Retrieves all doctor records

-- Select all from Patient table

Select * from Patient; -- Retrieves all patient records

-- Select all from Appointment table

Select * from Appointment; -- Retrieves all appointment records

-- Select all from Payment table

Select * from Payment; -- Retrieves all payment records

-- Select all from Stuff table

Select * from Staff; -- Retrieves all staff records (corrected from "Staff")

-- Select all from MusedinT table

Select * from MusedinT; -- Retrieves records from MusedinT table
(materials/treatments used)

-- Select all from Treatment table

Select * from Treatment; -- Retrieves all treatment records

-- Select all from Material table

Select * from Material; -- Retrieves all material records

-- Select all from TperformedInA table

Select * from Tpreformedina; -- Retrieves records from TperformedInA table
(treatments and appointments)

-- Select all from OMakeA table

Select * from OMakeA; -- Retrieves records from OMakeA table (offices and
appointments)

-- For each Select - Checking the number of rows in the table

Select count(*) from Office;

Select count(*) from Doctor;

Select count(*) from Patient;

Select count(*) from Appointment;

Select count(*) from Payment;

Select count(*) from Staff;

Select count(*) from MusedinT;

Select count(*) from Treatment;

Select count(*) from Material;

Select count(*) from Tpreformedina;

Select count(*) from OMakeA;

שיטות הכנסה :data

שיטה 1:

Data generator

הכנסה לטבלה Appointment:

The screenshot shows the Oracle SQL Developer Data Generator interface. In the top left, there's a tree view with 'APPOINTMENT' selected. Below it, a table definition is shown with columns: Name, Type, Size, Data, and Master. The 'Data' column for ADATE contains 'Random(01/01/2024, 01/01/2030)'. The 'Master' column for CID contains 'List(select CID from patient)'. The 'Number of records' is set to 400.

The screenshot shows the generated data for the APPOINTMENT table. The table has columns: ADATE, APPOINTMENTID, SID, and CID. The data consists of 400 rows of appointment details, such as dates ranging from 16.12.2027 to 01.09.2029, and various SID and CID values.

	ADATE	APPOINTMENTID	SID	CID
1	16.12.2027	72466	10001	759
2	21.12.2027	29763	10007	715
3	19.12.2024	87464	27192	610
4	13.07.2024	86374	81284	135
5	07.10.2029	85249	39960	149
6	05.03.2028	84333	65613	100
7	17.01.2028	91451	60010	774
8	12.03.2028	54837	25922	363
9	14.06.2029	35641	45140	418
10	01.07.2027	96185	10004	631
11	28.01.2028	87242	10010	343
12	03.12.2026	87694	51672	584
13	27.01.2026	37495	13794	705
14	12.07.2026	12895	86343	705
15	21.04.2026	12897	10001	654
16	12.11.2024	17827	21157	453
17	06.12.2024	38581	68296	374
18	19.08.2024	45765	11630	647
19	28.01.2025	22749	21998	445
20	08.04.2026	69943	10003	410
21	01.06.2028	16480	11630	841
22	08.09.2028	16416	72071	100
23	30.10.2029	99732	14460	553
24	18.06.2029	19354	77700	562
25	04.01.2024	27877	90716	930
26	25.04.2029	73779	91539	975
27	04.02.2029	84624	17595	980
28	16.11.2026	82863	51524	672
29	16.04.2026	45347	33532	632
30	03.01.2027	99523	90901	100
31	30.06.2025	12988	93442	218
32	12.01.2029	79326	32397	332
33	05.12.2028	54489	22238	832
34	21.12.2028	63398	38187	292
35	30.09.2024	18259	61435	556
36	20.07.2028	46565	47507	689
37	08.09.2028	77515	10011	127
38	24.02.2027	12375	69221	468
39	01.09.2029	38615	68229	690

SQL Output Statistics

```
select * from appointment
```

The screenshot shows a database query results window. The table has columns: ADATE, APPOINTMENTID, SID, and CID. The data includes various dates from 2024-09-20 to 2025-01-10, and IDs ranging from 1 to 994. The interface includes standard toolbar icons and a status bar at the bottom.

ADATE	APPOINTMENTID	SID	CID
24/09/2024	23487	22338	984
08/01/2025	79857	51672	140
02/03/2027	34432	10012	104
27/09/2027	85761	34941	362
26/02/2025	82511	46150	690
26/10/2029	48010	90000	980
15/05/2023	30001	10001	20001
02/05/2023	30002	10006	20002
03/05/2023	30003	10006	20003
04/05/2023	30004	10001	20004
05/05/2023	30005	10005	20005
06/05/2023	30006	10006	20006
07/05/2023	30007	10001	20007
08/05/2023	25145	57567	829
09/05/2023	39772	76580	434
20/04/2025	73199	79819	783
02/08/2025	96556	79374	601
08/08/2028	79819	79819	783
09/09/2028	35457	74545	803
25/06/2026	35792	54615	178
30/06/2029	29532	10652	475
28/10/2029	54399	63961	799
23/07/2025	31716	10009	874
05/08/2025	23080	10009	867
25/11/2024	62227	63036	513
26/17/02/2027	97354	21570	736
27/06/2026	37916	51524	903
08/09/2025	64388	86483	302
29/17/05/2029	74732	11630	391
30/07/2025	62053	86483	302
31/09/2029	39616	31845	633
32/12/2025	17975	99799	552
33/22/10/2025	99822	61696	198
34/24/07/2029	16487	10013	839
35/18/11/2028	66153	86483	776
36/01/01/2029	62053	86483	302
37/13/07/2029	55657	39306	250
38/16/06/2027	55127	69469	568
39/02/05/2024	43974	84092	800

1:26 SYSTEM\XE 405 rows selected in 0.249 seconds

הכנסה לطبלה :DOCTOR

DOCTOR

The screenshot shows a table definition for 'DOCTOR'. It has three columns: LICENSE (VARCHAR2(15), not null, master), SPECIALTIES (VARCHAR2(15), not null, master), and SID (NUMBER(5), not null, master). The table contains 200 records.

Name	Type	Size	Data	Master
LICENSE	VARCHAR2	15	[A]00000	...
SPECIALTIES	VARCHAR2	15	List('Dentist', 'Cosmetic', 'Orthodontist')	...
SID	NUMBER	5	List(select SID from STAFF)	...

DOCTOR

The screenshot shows the data for the DOCTOR table. It lists 200 rows of information, each containing a unique ID, license number, specialties, and SID. The specialties column includes values like Orthodontist, Cosmetic, Dentist, and others.

ID	LICENSE	SPECIALTIES	SID
R21734	Orthodontist	82187	
X14775	Cosmetic	44023	
I11009	Dentist	49969	
WT77145	Cosmetic	83211	
H57744	Dentist	18371	
P97044	Dentist	43762	
B000982	Orthodontist	50906	
U60050	Dentist	61039	
M632541	Orthodontist	89219	
Y607144	Dentist	91660	
Z37623	Cosmetic	10709	
D11009	Dentist	56479	
T032986	Cosmetic	34982	
F041679	Orthodontist	51861	
WT44714	Orthodontist	13227	
H695140	Orthodontist	51162	
C579087	Dentist	40041	
G599547	Dentist	13416	
A970447	Dentist	41152	
R21544	Orthodontist	52809	
W694558	Orthodontist	59663	
C77619	Dentist	55149	
B018507	Orthodontist	51740	
T89992	Cosmetic	48188	
O522327	Cosmetic	27192	
F11009	Dentist	51539	
Y599324	Dentist	51524	
P033161	Dentist	71670	
B726553	Cosmetic	47962	
O660703	Dentist	70250	
5996963	Cosmetic	11311	
C62481	Dentist	40795	
Q512057	Dentist	41109	
N131782	Orthodontist	27991	
L353466	Orthodontist	70482	
5996931	Cosmetic	98750	
O327631	Dentist	76479	
M27875	Orthodontist	61696	
N691773	Dentist	88762	

Definition Options Result SYSTEM\XE Generated in 0.031 seconds. 1 table(s) with 200 rows.

LICENSE	SPECILATIES	ID
1 L1234	Orthodontist	10001
2 L2345	Cosmetic	10002
3 L3456	Dentist	10003
4 L3456	Dentist	10004
5 L3456	Cosmetic	10005
6 L3456	Dentist	10006
7 L7890	Dentist	10007
8 L7890	Orthodontist	10008
9 L4567	Orthodontist	10009
10 L8901	Dentist	10010
11 L987654	Cosmetic	10011
12 P980988	Orthodontist	33794
13 A933261	Orthodontist	61434
14 A933261	Cosmetic	61334
15 H007115	Orthodontist	95120
16 H007115	Orthodontist	20000
17 H007115	Dentist	34476
18 A400352	Orthodontist	61435
19 A400352	Cosmetic	61436
20 A79750	Orthodontist	74897
21 A648515	Orthodontist	60240
22 A648515	Cosmetic	60241
23 P800902	Cosmetic	27762
24 A887730	Orthodontist	69350
25 A887730	Cosmetic	69351
26 H008108	Dentist	27719
27 H008108	Cosmetic	19303
28 H008108	Orthodontist	20000
29 U000205	Cosmetic	61546
30 U000205	Orthodontist	61547
31 U000205	Dentist	96792
32 B62092	Cosmetic	34609
33 B62092	Orthodontist	25295
34 E98487	Orthodontist	63432
35 H008108	Cosmetic	27199
36 H008108	Orthodontist	20000
37 C091751	Cosmetic	53206
38 H456372	Orthodontist	39661
39 H105043	Cosmetic	39296

הכנסה לטבלה :Material

MATERIAL		
MID	MNAME	AMOUNT
74716	Anesthetic Pcah	175
44836	Anesthetic Emxz	37
54383	Anesthetic Royx	141
36681	Antibiotic Jyzz	120
19478	Septanest Mmlha	8
23314	Septanest Dymenf	219
57353	Pills Zugqab	6
42232	Ointment Cvbgc	94
63934	Septanest Bamvl	192
53564	Anesthetic Zsss	113
78293	Antibiotic Mbjz	71
57648	Antibiotic Lenn	219
57655	Septanest Cipq	126
89847	Ointment Matnvl	81
68235	Septanest Izepi	105
66981	Septanest Djkg	28
99943	Antibiotic Voee	3
85727	Ointment Altrud	195
53718	Septanest Mgney	55
41993	Pills Norar	29
99837	Antibiotic Maxz	68
55993	Pills Ofkeqj	245
94795	Anesthetic Qcms	5
69546	Septanest Rojiv	27
52584	Septanest Evnra	11
77373	Pills Ibbnd	156
19777	Ointment Yrgina	175
48969	Pills Cupkag	13
18611	Pills Nebkq	142
13383	Ointment Fngw	17
39962	Ointment Hdmb	162
17462	Ointment Thybyd	226
24514	Anesthetic Oide	77
32878	Pills Ngono	224
75879	Antibiotic Lnnl	145
37447	Pills Otdqr	37
37564	Pills Lgch	186
29911	Antibiotic Drve	67
79575	Pills Ebdkrc	130

Definition Options Result

• SYSTEM@XE - Generated in 0.046 seconds. 1 table(s) with 400 rows.

Screenshot of Oracle SQL Developer showing the creation of a table named 'MATERIAL' and its data.

MATERIAL

Name	Type	Size	Data	Master
MID	NUMBER	5	[11111]	...
MNAME	VARCHAR2	15	[List:Antibiotic,'Anesthetic','Pills','Ointment','Septanest'] + [Aaaaaaa]	...
AMOUNT	NUMBER	5	Random(1,250)	...

Definition | Options | Result

SYSTEM@XE Generated in 0 seconds. 1 table(s) with 17 rows.

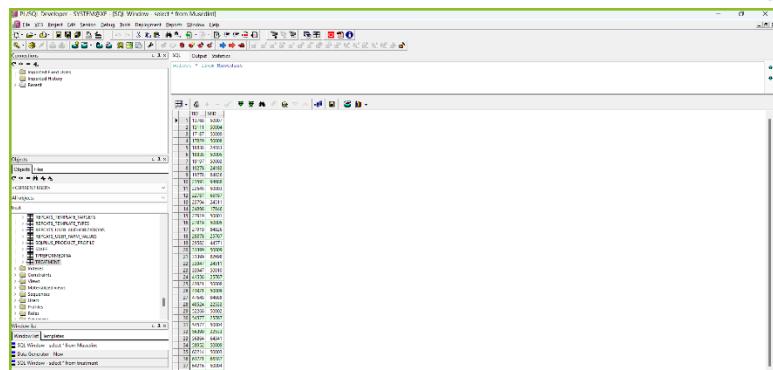
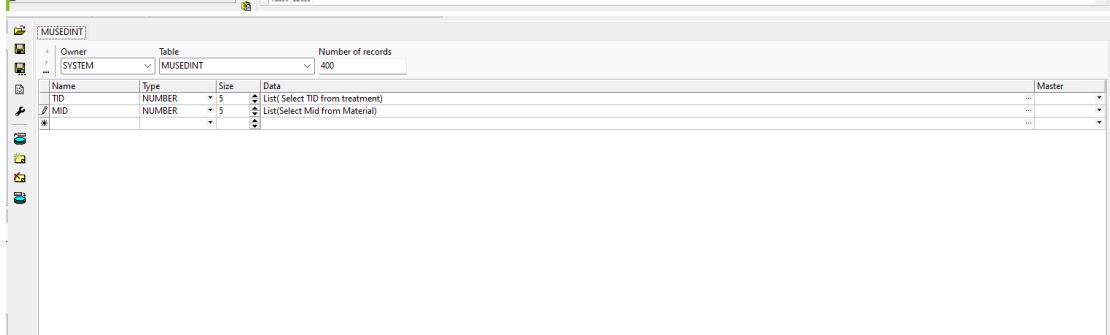
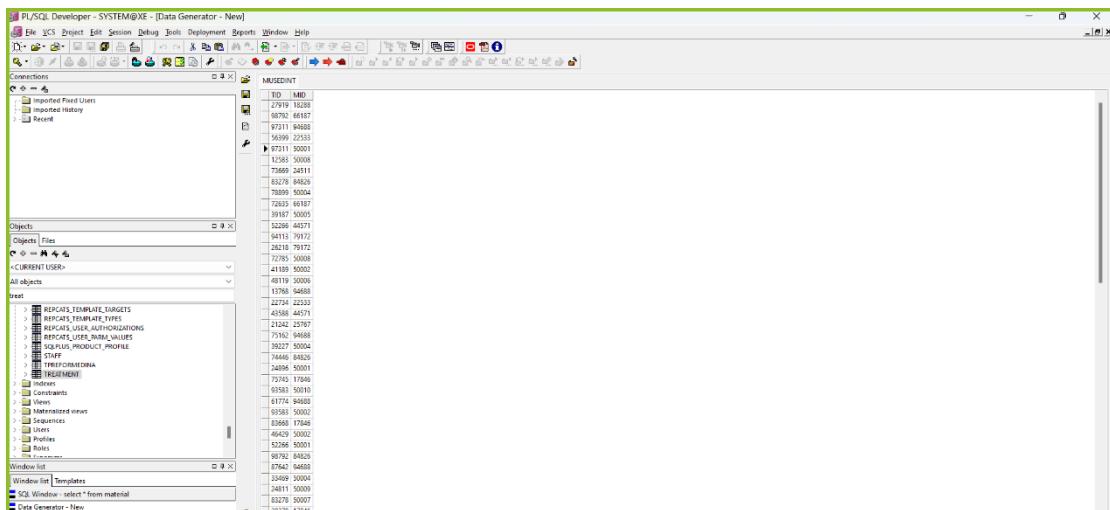
SQL Output Statistics

```
select * from MATERIAL
```

	MID	MNAME	AMOUNT
179	51142	Anesthetic Qbnk	187
180	13449	Ointment Fmctqu	249
181	96364	Anesthetic Xsgu	45
182	74315	Pills Qjxgbd	199
183	32992	Septanest Lztg	5
184	83936	Antibiotic Myij	5
185	44631	Ointment Fpbfd	250
186	91849	Septanest Oenyr	90
187	63651	Anesthetic Numh	89
188	53673	Septanest Vugtw	203
189	95165	Septanest Troke	78
190	54674	Septanest Ogvbn	13
191	29138	Antibiotic Vsmi	81
192	56631	Ointment Azill	169
193	15199	Antibiotic Wcqg	104
194	94219	Anesthetic Orik	51
195	41311	Anesthetic Gxjv	206
196	68746	Anesthetic Bbge	47
197	11323	Antibiotic Yeek	182
198	34646	Ointment Drlrcp	232
199	75214	Pills Axbsrh	97
200	62615	Antibiotic Jkxt	137
201	22792	Pills Qhjjs	74

1:23 SYSTEM@XE 439 rows selected in 0.202 seconds

הכנסה לطلبת :Musedint



הכנסה לטרבלה :office

SQL Output Statistics

```
select * from OFFICE
```

	OTYPE	SID
1	Secretariat	10011
2	Sales	10013
3	Secretariat	10013
4	Sales	10014
5	Secretariat	10015
6	Secretariat	10016
7	Sales	10017
8	Shift Manager	10018
9	Shift Manager	10019
10	Sales	10020
11	Shift Manager	12979
12	Sales	68062
13	Customer Servic	29121
14	Secretariat	73973
15	Shift Manager	87171
16	Secretariat	78572
17	Secretariat	41224
18	Shift Manager	86483
19	Sales	57569
20	Secretariat	29872
21	Secretariat	25578
22	Customer Servic	86779
23	Secretariat	26921
24	Shift Manager	69582
25	Secretariat	38296
26	Shift Manager	64636
27	Customer Servic	45839
28	Counter	34002
29	Secretariat	18802
30	Secretariat	55385
31	Secretariat	10394
32		49969
33	Counter	51383
34	Shift Manager	65813
35	Sales	10334
36	Counter	16712
37	Customer Servic	95956
38	Secretariat	47970
391	Customer Servic	29127

SYSTEM@XE 1:21 SYSTEM@XE 278 rows selected in 0.233 seconds

OFFICE

Owner	Table	Number of records
SYSTEM	OFFICE	400

Name	Type	Size	Data
OTYPE	VARCHAR2	15	List(Shift Manager;Sales Secretariat;Counter;Customer Service;IT)
SID	NUMBER	5	List>Select SID from staff)

Master

PL/SQL Developer - SYSTEM@XF - [Data Generator - New]

File JCS Project Edit Session Debug Tools Deployment Reports Window Help

Connections

- Imported Fixed Users
- Imported History
- Recent

Objects

Objects Files

<CURRENT USER>

All objects

staff

- REPLICATE_TEMPLATE_REFGROUPS
- REPLICATE_TEMPLATE_SITES
- REPLICATE_TEMPLATE_USERS
- REPLICATE_TEMPLATE_TARGETS
- REPLICATE_TEMPLATE_TYPES
- REPLICATE_TEMPLATE_DOMAINS
- REPLICATE_USER_ALARM_VALUES
- GOLULUS_PRODUCT_PROFILE
- TRIGGER
- TRIGGERED_BY
- TRIGGER_CRONDATA
- TREATMENT
- audit
- Constraints
- Views
- Materialized views
- Sequences
- Ita_schemas

Modeler

Windows

- Templates
- SQL Window - select* from office
- Data Generator - New

Definition Options Result

SYSTEM@XE 1 Generated in 0.031 seconds. 1 table(s) with 400 rows.

הכנסה לטבלה :omakea

OMAKEA

APPOINTMENTID	SID
19422	10013
34262	12979
43913	98375
83776	16712
62227	69592
71381	86483
96548	41224
34362	16712
49699	68429
63768	26921
59549	32397
83194	26921
63768	10020
30006	10013
32826	29872
27341	45839
83194	51383
37638	10020
37869	78572
25798	10013
66549	41470
38729	29127
72766	10019
96463	81894
73721	59041
19974	29127
44623	50598
33315	41224
31554	41470
76832	88762
58994	10017
54755	33360
14235	68062
97927	10018
43913	26921
39953	45839
69358	55385
79857	36074
37916	47970

OMAKEA

Owner	Table	Number of records
SYSTEM	OMAKEA	400

OMAKEA

Name	Type	Size	Data	Master
APPOINTMENTID	NUMBER	5	List(select APPOINTMENTID from APPOINTMENT)	...
SID	NUMBER	5	List(select SID from office)	...

Definition Options Result

SYSTEM@XE Generated in 0.016 seconds. 1 table(s) with 400 rows.

Definition Options Result

SYSTEM@XE 400 records generated in 4.719 seconds

SQL Output Statistics

```
select * from omaked;
```

	APPOINTMENTID	SID
1	11391	41224
2	11416	83782
3	11912	53269
4	12777	12979
5	13121	10018
6	13121	65613
7	13184	32397
8	13235	53975
9	13881	10018
10	13987	47970
11	13944	36074
12	14115	10018
13	14235	10334
14	14235	50598
15	14235	74972
16	14235	87171
17	14211	69962
18	15747	87171
19	16193	10015
20	16437	10011
21	16511	10020
22	17642	83782
23	18374	10334
24	18423	57969
25	18423	10012
26	18423	33360
27	18621	29127
28	18621	64636
29	19338	78572
30	19338	98375
31	19487	16712
32	19548	50598
33	19548	14012
34	21759	83782
35	22473	10017
36	22473	29561
37	22473	47970
38	22834	16712
39	22834	87171

1:21 SYSTEM@XE 409 rows selected in 0.266 seconds

הכנסה לטבלה Trefoemedina :TREFOEMEDINA

TREFOEMEDINA

Owner	Table	Number of records
SYSTEM	TREFOEMEDINA	400
	Name Type Size Data	Master
	TID NUMBER 5 List(Select TID from treatment)	...
	APPOINTMENTID NUMBER 5 List>Select APPOINTMENTID from APPOINTMENT	...

PLSQL Developer - SYSTEM@XE - [Data Generator - New]

File | Project | Edit | Session | Debug | Tools | Deployment | Reports | Window | Help

Connections

- System
- Imported Fixed Users
- Imported History
- Recent

Objects

- Objects | Files
- <CURRENT USER>
- All objects

Tree

- REPLICAS_TEMPLATES_IMAGES
- REPLICAS_TEMPLATES_TYPES
- REPLICAS_USER_AUTHORIZATIONS
- REPLICAS_USER_PRIVILEGES
- SOLVUS_PRODUCT_PROFILE
- STAFF
- TREATMENT
- TRTREFORMEDINA
- TRTREFORMEDINA
- TREATMENT
- Users
- Constraints
- Views
- Materialized views
- Sequences
- Users
- Profiles
- Roles

Window list

- Window list | Templates
- SQL Window - select * from Musedint
- Data Generator - New
- SQL Window - select * from treatment

SQL Output Statistics

```
select * from TRTREFORMEDINA
```

ID	APPOINTMENTID
1	11353
2	11355
3	11356
4	12265
5	12265
6	13622
7	13622
8	13622
9	14353
10	14353
11	14543
12	14543
13	14552
14	14552
15	14675
16	14675
17	14839
18	14839
19	15973
20	15972
21	16211
22	16211
23	16365
24	16754
25	16666
26	17119
27	17141
28	18332
29	18675
30	18841
31	19389
32	19434
33	21241
34	21241
35	21241
36	21534
37	21534
38	21534
39	22218

SYSTEM@XE 398 rows selected in 0.219 seconds

הכנסה לutable: Treatment

SQL Output Statistics

```
select * from treatment
```

TTYPE	DESCRIPTION	PRICE	ID	TIME
1 Orthodontics	Extraction	13000	61916	3
2 Orthodontics	Teeth filling	14352	62262	1
3 Rehabilitation	Gum	9141	59883	5
4 Rehabilitation	Braces	2181	55534	3
5 Rehabilitation	Gum	6397	82913	4
6 Aesthetic	Extraction	4312	71983	2
7 Aesthetic	Gum	3899	63726	3
8 Aesthetic	Gum	14364	26847	3
9 Aesthetic	Teeth filling	20000	61452	4
10 Rehabilitation	Teeth filling	6248	43949	3
11 Orthodontics	Braces	1821	91954	5
12 Aesthetic	Teeth filling	5821	14839	3
13 Rehabilitation	Gum	4366	85777	4
14 Aesthetic	Extraction	4633	43825	6
15 Orthodontics	Extraction	6154	83652	1
16 Orthodontics	Teeth filling	2637	73179	2
17 Orthodontics	Extraction	6154	93262	3
18 Rehabilitation	Gum	9004	46956	5
19 Aesthetic	Gum	1179	46956	2
20 Rehabilitation	Teeth filling	9718	46312	7
21 Orthodontics	Gum	5043	36836	4
22 Rehabilitation	Teeth filling	9393	85741	8
23 Aesthetic	Extraction	9172	31183	4
24 Orthodontics	Teeth filling	12887	55837	4
25 Rehabilitation	Extraction	998	31771	7
26 Aesthetic	Teeth filling	7000	27147	9
27 Aesthetic	Braces	13773	70002	6
28 Rehabilitation	Braces	12511	21241	9
29 Rehabilitation	Braces	3793	51348	7
30 Rehabilitation	Extraction	4856	54551	7
31 Orthodontics	Teeth filling	3994	83824	4
32 Rehabilitation	Teeth filling	9978	55941	7
33 Orthodontics	Braces	4053	83715	9
34 Rehabilitation	Gum	7817	17141	5
35 Orthodontics	Braces	8000	59523	5
36 Orthodontics	Teeth filling	6541	65452	5
37 Aesthetic	Teeth filling	3624	72198	6
38 Rehabilitation	Braces	10087	72385	8
39 Aesthetic	Gum	1109	43414	6

TREATMENT

TYPE	DESCRIPTION	PRICE	TID	TIME
Orthodontics	Extraction	8056	56699	8
Orthodontics	Gum	6806	42933	6
Rehabilitation	Braces	12208	46612	2
Rehabilitation	Extraction	897	55477	3
Orthodontics	Teeth filling	2316	37179	1
Rehabilitation	Gum	12515	31433	3
Rehabilitation	Gum	9905	18125	9
Orthodontics	Gum	6179	95651	1
Rehabilitation	Gum	1949	42417	4
Rehabilitation	Braces	12106	95468	7
Aesthetic	Gum	3361	64193	4
Rehabilitation	Teeth filling	12368	28443	8
Aesthetic	Teeth filling	9920	82689	7
Orthodontics	Gum	13610	98131	1
Orthodontics	Gum	11022	64361	4
Rehabilitation	Teeth filling	14243	31948	7
Rehabilitation	Gum	2583	39691	7
Orthodontics	Extraction	6610	73447	2
Orthodontics	Extraction	11328	38691	1
Aesthetic	Teeth filling	12661	29731	2
Orthodontics	Braces	11965	96641	4
Orthodontics	Teeth filling	11575	59523	1
Aesthetic	Teeth filling	1857	23271	1
Orthodontics	Gum	6100	95957	4
Aesthetic	Extraction	3505	29343	6
Orthodontics	Braces	2405	46771	4
Rehabilitation	Extraction	5041	36612	9
Orthodontics	Gum	3248	41881	9
Orthodontics	Teeth filling	5765	48288	6
Aesthetic	Extraction	4642	18318	4
Rehabilitation	Extraction	604	95127	5
Orthodontics	Gum	7752	38637	5
Aesthetic	Braces	10692	68245	6
Rehabilitation	Extraction	2855	47588	3
Aesthetic	Teeth filling	7821	93835	8
Aesthetic	Braces	8607	98754	3
Orthodontics	Braces	3957	73538	5
Rehabilitation	Braces	11995	72784	3
Rehabilitation	Braces	6185	19541	9

Definition Options Result

TREATMENT

Owner	Table	Number of records
SYSTEM	TREATMENT	400

Name	Type	Size	Data	Master
TYPE	VARCHAR2	15	List('Orthodontics';'Rehabilitation';'Aesthetic')	...
DESCRIPTION	VARCHAR2	200	List("Teeth filling";'Extraction';'Braces';'Gum')	...
PRICE	NUMBER	5	Random(50,15000)	...
TID	NUMBER	5	[11111]	...
TIME	NUMBER	5	[1]	...
*				...

Definition Options Result

SYSTEM@XE - 400 records generated in 3.312 seconds

שיטת 2

Generatedata- Website

הכנסה לטבלה :material

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'SQL', 'Output', and 'Statistics'. Below the tabs, a code editor contains the SQL query:

```
select * from material
```

The main area displays the results of the query in a grid format. The columns are labeled 'MID', 'MNAME', and 'AMOUNT'. The data consists of 45 rows, each containing a unique ID, a product name, and a quantity. The names include various medical supplies like Septanest Dkthy, Ointment Yttrch, Pills Ugdnqz, etc.

MID	MNAME	AMOUNT
429	Septanest Dkthy	163
430	Ointment Yttrch	46
431	Pills Ugdnqz	243
432	Septanest Ybrhx	238
433	Septanest Owlmu	211
434	Septanest Eavhj	212
435	Septanest Ngdzh	60
436	Pills Onglod	61
437	Anesthetic Hcdn	116
438	Ointment Wcvqrw	224
439	Anesthetic Eapy	178
440	Antibiotic	226
441	Thermometers	37
442	Disinfectant	59
443	Septanest	83
444	Cotton Swabs	120
445	Bandages	136
446	Gauze	167
447	Pills	168
448	Gloves	54
449	Anesthetic	190
450	Syringes	136
451	Pills	69

At the bottom of the interface, there are several toolbars and status indicators, including a connection icon, a session number (2:1), and a status message: SYSTEM@XE -> mid, number(5), mandatory.

Data from Textfile Data to Oracle

General

Owner: Table MATERIAL Clear Table
 Commit every: Overwrite duplicates Finalizing Script
 Ignore duplicates

Fields

Field1 MID -> MID
 Field2 MNAME -> MNAME
 Field3 AMOUNT -> AMOUNT

Field Fieldtype Create SQL
 SQL function additional Oracle processing, for example: substr(#, 1, 20)

Result Preview

MID	MNAME	AMOUNT
86698	Antibiotic	226
14726	Thermometers	37
72115	Disinfectant	59
12570	Septanes	83
11654	Cotton Swabs	120
24208	Bandages	136
61185	Gauze	167
14470	Pills	168
70154	Gloves	54
99078	Anesthetic	190
73385	Syringes	136
16596	Pills	69

Import Import to Script Close SYSTEM@XE material.txt loaded, 1 KB Help

Data from Textfile Data to Oracle

File Data

```

MID      MNAME      AMOUNT
86698  Antibiotic  226
14726  Thermometers 37
72115  Disinfectant 59
12570  Septanes 83
11654  Cotton Swabs 120
24208  Bandages 136
61185  Gauze 167
14470  Pills 168
70154  Gloves 54
99078  Anesthetic 190
73385  Syringes 136
16596  Pills 69
  
```

Encoding

Configuration

General

Fieldcount: 3 Quote character: * Comment line:
 End at line-end
 Name in header
 Skip empty lines Import lines: 1 ..

Field Start: Field1 (+0..<tab>) MID
 Field2 (+0..<tab>) MNAME
 Field3 (+0..<tab>) AMOUNT

Field End: Relative position
 Absolute position
 Character Length
 Character

Apply

Result Preview

MID	MNAME	AMOUNT
86698	Antibiotic	226
14726	Thermometers	37
72115	Disinfectant	59
12570	Septanes	83
11654	Cotton Swabs	120
24208	Bandages	136
61185	Gauze	167
14470	Pills	168
70154	Gloves	54
99078	Anesthetic	190
73385	Syringes	136
16596	Pills	69

Import Import to Script Close SYSTEM@XE material.txt loaded, 1 KB Help

הכנסה לטבלה :patient

	CBIRTHYEAR	CADDRESS	CNAME	CGENDER	CID	CMOBILE	CMAIL	...
290	1959	Ap #905-980 Est Ave	Herrod Ingram	F	322	0575339523	auctor.mauris@google.org	...
291	1996	Ap #220-2101 Egestas Av.	Whitney Holt	F	249	0580933718	eget@hotmail.com	...
292	1945	434-1217 Non Street	Hyatt Gay	F	495	0528533319	lacus.vestibulum@yahoo.org	...
293	2005	Ap #508-4203 Amet, Rd.	Mollie Nguyen	F	221	0590449083	sodales.nisi@icloud.org	...
294	1916	Ap #204-7890 Neque, Road	Hop Wright	M	919	0556847954	eu.lacus@icloud.net	...
295	1952	P.O. Box 394, 9604 Quis Rd.	David Sexton	F	241	0531781517	mattis.cras@protonmail.com	...
296	1939	401-1982 Erat, Street	Harrison Estrada	F	524	0581638292	vivamus.sit.amet@google.ca	...
297	2021	875-4218 Aliquam St.	Martin Pacheco	M	864	0586515972	pede.cum@protonmail.org	...
298	1951	P.O. Box 288, 6324 Primis Rd.	Jordan Mathis	M	833	0583375862	arcu.iaculis@protonmail.co.uk	...
299	1941	Ap #346-6650 Faucibus Road	Dahlia Reilly	F	812	0508338651	diam.pellentesque@hotmail.co.uk	...
300	1954	P.O. Box 321, 9761 Arcu, Rd.	Thomas Mendez	M	725	0527817282	tristique.senectus@hotmail.ca	...
301	1925	Ap #405-7497 Enim, Road	Isadora Monroe	M	169	0512156476	odio.etiam@protonmail.net	...
302	1943	893-9162 Egestas, Av.	Marcia Arnold	F	874	0560675237	rutrum.eu@aol.com	...
303	2000	Ap #671-8494 Eu, Ave	Simone Suarez	F	713	0571085153	posuere.at@aol.org	...
304	2003	4712 Nec Rd.	Galvin Reeves	M	448	0576128449	vehicula.risus nulla@aol.edu	...
305	1983	392-9303 Duis Road	Kenyon Hall	F	841	0566685505	elementum.dui@protonmail.ca	...
306	1959	205-4410 Fringilla Rd.	Charlotte Duffy	M	690	0582366338	nunc@aol.ca	...
307	2018	P.O. Box 435, 2678 Sed Rd.	Brock Valenzuela	F	281	0500338868	et.ultrices.posuere@google.edu	...
308	1938	P.O. Box 124, 5624 Eget St.	Gannon Garza	M	696	0533315498	voluptat.nulla@hotmail.org	...
309	1942	Ap #110-4949 Nulla, St.	Cullen Greene	M	952	0586368553	tincidunt.donec@yahoo.net	...
310	2012	P.O. Box 145, 8216 Nascentur St.	Unity Ayala	M	798	059037161	placerat@icloud.net	...
311	1947	Ap #339-3186 Velle, Av.	Xander Richards	M	616	0511854727	ut.odio@google.ca	...
312	1997	P.O. Box 243, 784 Facilisis Street	Shea Shepherd	M	707	058528867	pellentesque@icloud.edu	...
313	1947	202-555-1414 Rd.	Sophie Parker	M	457	0555032373	l."	...

Data from Textfile Data to Oracle

General

Owner Table Clear Table Initializing Script
 Overwrite duplicates
 Ignore duplicates Finalizing Script

Commit every... 0 Field

Fields

Field1 CNAME -> CNAME (VARCHAR2)	Field
Field2 CMOBILE -> CMOBILE (VARCHAR2)	Fieldtype
Field3 CMAIL -> CMAIL (VARCHAR)	Create SQL
Field4 CBIRTHYEAR -> CBIRTHYEAR (NUMBER)	SQl function
Field5 CGENDER -> CGENDER (VARCHAR2)	additional Oracle processing, for example: substr(#, 1, 20)
Field6 CID -> CID (NUMBER)	
Field7 CADRESS -> CADRESS (VARCHAR2)	

Result Preview

CNAME	CMOBILE	CMAIL	CBIRTHYEAR	CGENDER	CID	CADDRESS
Louis Harding	0576085009	commodo@yahoo.edu	1987	M	783	5429 Arcu. Street
Bruce Walton	0584338727	diam.duis.mi@yahoo.co.uk	1986	F	601	P.O. Box 643, 736 Mauris Rd.
Flynn Jensen	0582426580	ligula.aliquam@aol.net	1916	F	871	Ap #852-970 Neque. Street
Noelle James	0563243310	erat@yaho.net	1977	F	410	2289 Cras Ave
Kathleen Santana	051139756	neque.non.quam@protonmail.edu	2003	M	373	564-2259 Cras Ave
Macey Tucker	0544411142	malesuada.fames.ac@google.co.uk	1996	M	168	Ap #265-7286 Sollicitudin St.
Amanda Mcmahon	0512426537	morbi.tristique@protonmail.ca	2011	F	946	P.O. Box 640, 8387 Pulvinar Street
Eleanor Jefferson	0582175221	ac.metus@outlook.ca	2004	F	577	Ap #958-7751 Vivera. Ave
Ira Day	0524324167	et.nunc.cuisque@yahoo.ca	1945	M	838	P.O. Box 175, 6500 Quisque Av.

Import Import to Script Close SYSTEM@XE -# updated patient.txt loaded, 33 KB Help

Data from Textfile Data to Oracle

File Data

CNAME CMOBILE CMAIL CBIRTHYEAR CGENDER CID CADRESS
Louis Harding 0576085009 commodo@yahoo.edu 1987 M 783 5429 Arcu. Street
Bruce Walton 0584338727 diam.duis.mi@yahoo.co.uk 1986 F 601 P.O. Box 643, 736 Mauris Rd."
Flynn Jensen 0582426580 ligula.aliquam@aol.net 1916 F 871 Ap #852-970 Neque. Street
Noelle James 0563243310 erat@yaho.net 1977 F 410 2289 Cras Ave
Kathleen Santana 051139756 neque.non.quam@protonmail.edu 2003 M 373 564-2259 Cras Ave
Macey Tucker 0544411142 malesuada.fames.ac@google.co.uk 1996 M 168 Ap #265-7286 Sollicitudin St.
Zeph Beach 058485333 egestas.a.dui@hotmail.ca 1984 F 371 369-2700 Consectetur Av.
Eleanor Jefferson 0582175221 ac.metus@outlook.ca 2004 F 577 Ap #958-7751 Vivera. Ave
Ira Day 0524324167 et.nunc.cuisque@yahoo.ca 1945 M 838 "P.O. Box 175, 6500 Quisque Av."
Maisie Hampton 0512468528 penatibus.era@yahoo.com 1934 F 802 5479 Mi St.
Malachi Hampton 0564423517 tellus@google.ca 1980 F 597 Ap #839-3424 Mauris Rd.
Marcia McKnight 0542472697 mattis.semper@hotmail.net 1935 F 648 886-9121 Augue Rd.
Ariana Macias 0582426510 euismod.sed@outlook.net 1931 F 4753 2289 Cras Ave
Doreen Martinez 05040626510 phasellus.libero@outlook.edu 1941 F 104 766-2061 Quam Street
Darryl Lewis 0583347334 ipsum@protonmail.net 1933 F 377 274-2912 Morbi Rd.
Finn Henson 0532156133 eget@icloud.net 1989 M 134 897-649 In Rd.

Configuration

General

Fieldcount 7	Quote character "	Field1 (+, -, "tab") CNAME	Field Start
<input checked="" type="checkbox"/> End at line-end	Comment line	Field2 (+, -, "tab") CMOBILE	<input type="radio"/> Relative position
<input checked="" type="checkbox"/> Name in header	Import lines 1	Field3 (+, -, "tab") CMAIL	<input type="radio"/> Absolute position
<input checked="" type="checkbox"/> Skip empty lines		Field4 (+, -, "tab") CBIRTHYEAR	<input type="radio"/> Character
		Field5 (+, -, "tab") CGENDER	
		Field6 (+, -, "tab") CID	
		Field7 (+, -, "tab") CADRESS	
		Field End	<input type="radio"/> Length
			<input type="radio"/> Character

Result Preview

CNAME	CMOBILE	CMAIL	CBIRTHYEAR	CGENDER	CID	CADDRESS
Louis Harding	0576085009	commodo@yahoo.edu	1987	M	783	5429 Arcu. Street
Bruce Walton	0584338727	diam.duis.mi@yahoo.co.uk	1986	F	601	P.O. Box 643, 736 Mauris Rd.
Flynn Jensen	0582426580	ligula.aliquam@aol.net	1916	F	871	Ap #852-970 Neque. Street
Noelle James	0563243310	erat@yaho.net	1977	F	410	2289 Cras Ave
Kathleen Santana	051139756	neque.non.quam@protonmail.edu	2003	M	373	564-2259 Cras Ave
Macey Tucker	0544411142	malesuada.fames.ac@google.co.uk	1996	M	168	Ap #265-7286 Sollicitudin St.
Zeph Beach	058485333	egestas.a.dui@hotmail.ca	1984	F	371	369-2700 Consectetur Av.
Eleanor Jefferson	0582175221	ac.metus@outlook.ca	2004	F	577	Ap #958-7751 Vivera. Ave
Ira Day	0524324167	et.nunc.cuisque@yahoo.ca	1945	M	838	P.O. Box 175, 6500 Quisque Av.

Import Import to Script Close SYSTEM@XE -# updated patient.txt loaded, 33 KB Help

הכנסה לטבלה :staff

```
select * from staff
```

	SADDRESS	SMOBILE	SNAME	SMAIL	SID
397	Ap #719-4855 Ante St.	... 535721837	Abra Clemons	abraclemons2857@aol.org	... 84940
398	809 Urna Street	... 575013566	Harrison Stewart	harrisonstewart@gmail.net	... 19098
399	Ap #901-2030 Vel Street	... 542991526	Jeremy Santana	jeremysantana@gmail.co.uk	... 69373
400	827-8375 Aliquam Rd.	... 527581368	Constance Navarro	constancenavarro@icloud.edu	... 31845
401	383-3411 Nam Rd.	... 566718812	Colby Shaffer	colbyshaffer7949@icloud.co.uk	... 25922
402	279-4046 A, St.	... 594524576	Stacey Mcleod	staceymcleod@aol.com	... 26135
403	4212 Odio Av.	... 521145828	Gannon Vance	gannonvance@google.org	... 77934
404	P.O. Box 408, 8813 Voluptat. Avenue	... 517871957	Alan Sullivan	alansullivan@aol.com	... 83782
405	8129 Facilisis Av.	... 598355463	Joelle Hoover	joellehoover4378@yahoo.co.uk	... 90376
406	Ap #331-8940 Ut Road	... 521151267	Kane Velazquez	kanevelazquez5579@icloud.com	... 90152
407	7215 Vestibulum Road	... 556769693	Halee Boyer	haleeboyer8640@yahoo.co.uk	... 61039
408	P.O. Box 933, 8375 Enim Rd.	... 582388653	Jordan Stanton	jordanstanton@gmail.net	... 72497
409	112-260 Varius Av.	... 576107777	Jescie Reynolds	jescireynolds4525@icloud.co.uk	... 44135
410	Ap #299-1591 Dictum Street	... 505456764	Harding Ballard	hardingballard776@icloud.ca	... 41470
411	442-2426 Ut Street	... 526678882	Gavin Foley	gavin Foley@yahoo.ca	... 16017
412	Ap #878-4716 Magna Rd.	... 554512612	Kennedy Atkins	kennedyatkins5511@yahoo.net	... 68062
413	369-2220 Proin Av.	... 556043206	Keely Lambert	keelylambert@aol.co.uk	... 87909
414	Ap #778-3523 Nunc Street	... 521473627	Kerry Cantrell	kerrycantrell3347@google.co.uk	... 37037
415	214-5228 Turpis St.	... 540508616	Nehru Kane	nehrukane@gmail.edu	... 13854
416	861-3422 Et, Rd.	... 548580479	Evan Norman	evannorman3424@yahoo.net	... 20080
417	P.O. Box 302, 6246 Risus. St.	... 538180461	Flynn Fulton	flynnfulton@google.edu	... 48224
418	Ap #727-177 Pede Rd.	... 545651455	Indigo Rios	indigorios2663@aol.net	... 74972
419	242-6132 Laoreet Ave	... 535495636	Dexter Brock	dexterbrock@google.edu	... 90220

Data from Textfile Data to Oracle

General

Owner: STAFF Clear Table
 Overwrite duplicates Ignore duplicates

Commit every... 0

Fields

Field1 -> SNAME (VARCHAR2)	Field2 -> MOBILE (CHAR)	Field3 -> SMAIL (VARCHAR2)	Field4 -> SADDRESS (VARCHAR2)
Field5 -> SID (NUMBER)			

Field: SID (NUMBER) Fieldtype: Number Create SQL
 SQL function additional Oracle processing, for example: substr(#, 1, 20)

Result Preview

1	2	3	4	5
Shaine Sosa	0576802140	sheinesosa8233@gmail.co.uk	196-696 Tincidunt Av.	71870
Kirsten Foreman	0576163875	kirstenforeman55@google.org	Ap #517-3634 Sed Street	53162
Hillary Garrett	0577647211	hillarygarrett@yahoo.org	Ap #105-1406 Loren Av.	19365
Susan Lester	0535653718	susanlester2407@google.com	P.O. Box 103, 7570 Vestibulum St.	69468
Brianna Holcomb	051274695	briannaholcomb304@cloud.ca	P.O. Box 128, 3885 Vitae St.	94537
Imani Clayton	0533026653	imanicleyton@icloud.com	5640 Justo St.	31070
Colleen Nichols	0554287212	colleenichols341@gmail.co.uk	235-3854 Ipsum St.	49506
Constance Francis	0596111846	constancefrancis@yahoo.ne	764-1812 Velit Street	51534
Ava Nolan	0530216427	avandanl@gmail.org	P.O. Box 712, 8640 Odio. Rd.	22376
Kato Merrill	0536481753	katomerrill99@aol.co.uk	560-8645 Reus, Av.	57610

Import Import to Script Close SYSTEM@XE UPDATED STAFF.txt loaded, 31 KB Help

File Data

Shaine Sosa 0576802140 sheinesosa8233@gmail.co.uk 196-696 Tincidunt Av. 71870
Kirsten Foreman 0576163875 kirstenforeman55@google.org Ap #517-3634 Sed Street 53162
Hillary Garrett 0577647211 hillarygarrett@yahoo.org Ap #105-1406 Loren Av. 19365
Susan Lester 0535653718 susanlester2407@google.com P.O. Box 103, 7570 Vestibulum St. 69468
Brianna Holcomb 051274695 briannaholcomb304@cloud.ca P.O. Box 128, 3885 Vitae St. 94537
Imani Clayton 0533026653 imaniclayton@icloud.com 5640 Justo St. 31070
Colleen Nichols 0554287212 colleenichols341@gmail.co.uk 235-3854 Ipsum St. 49506
Constance Francis 0596111846 constancefrancis@yahoo.net P.O. Box 712, 8640 Odio. Rd. 22376
Ava Nolan 0530216427 avandanl@gmail.org "P.O. Box 712, 8640 Odio. Rd." 22376
Kato Merrill 0536481753 katomerrill99@aol.co.uk 560-8645 Reus, Av. 57610
Colleen Nichols 0554287212 colleenichols341@gmail.co.uk 235-3854 Ipsum St. 49506
Heidi Rogers 0518248264 heidirogers@gmail.net Ap #169-5910 Tincidunt Av. 53235
Keelie Shannon 0594678748 keelieshannon@aol.net 9432 Pede, Rd. 89350
Kaye Stephenson 0545145098 kayatephenson@google.com Ap #509-7935 Tempus Ave 44433
Imani Cantu 0533844066 imanicantu@gmail.co.uk 850-1622 Pretium Road 72479
Len Hayes 0590386181 lenhayes3989@gmail.com 750-1000 Blanda, Rd. 30039
Markell Reeves 0571141691 markellreeves335@gmail.com 3170 Elit Rd. 30039
Jerome Walters 05694085264 jeromekwalters8359@icloud.com 8509 Cras Avenue 90364
Sydney McLaughlin 05585695831 sydneymclaughlin342@gmail.com "P.O. Box 437, 6016 Dolor. Rd." 61435
Valentino Forbes 0586966378 valentinoforbes2516@yahoo.ca Ap #39-3448 Mollis Street 63036
Lisandra Carson 0506169440 lisandracarson@icloud.com 489-892 Id Rd. 12474
Florence Herrera 0577187944 florenceherreia057@gmail.com 246-4726 Tellus, Av. 44023

Configuration

General

Fieldcount: 5 Quote character: " End at line-end: Comment line: Import lines: 1 ..

Field1 (+,"-tab")	Field Start: Relative position
Field2 (+,"-tab")	Absolute position
Field3 (+,"-tab")	Character
Field4 (+,"-tab")	Field End: Length
Field5 (+,"-tab")	Character

Filter Apply

Result Preview

1	2	3	4	5
Shaine Sosa	0576802140	sheinesosa8233@gmail.co.uk	196-696 Tincidunt Av.	71870
Kirsten Foreman	0576163875	kirstenforeman55@google.org	Ap #517-3634 Sed Street	53162
Hillary Garrett	0577647211	hillarygarrett@yahoo.org	Ap #105-1406 Loren Av.	19365
Susan Lester	0535653718	susanlester2407@google.com	P.O. Box 103, 7570 Vestibulum St.	69468
Brianna Holcomb	051274695	briannaholcomb304@cloud.ca	P.O. Box 128, 3885 Vitae St.	94537
Imani Clayton	0533026653	imanicleyton@icloud.com	5640 Justo St.	31070

Import Import to Script Close SYSTEM@XE UPDATED STAFF.txt loaded, 31 KB Help

שיטה 3

Tex importer:
הכנסה של patient ו גם של staff

The image displays three separate screenshots of the generatedata.com/generator web application, illustrating the configuration of data sets for different output formats.

MySQL Screenshot:

- Data Type:** Names, Column Name: SNAME, Example: Alex Smith.
- Options:** Name Surname, SOURCE: NAME FIELDS.
- Data Type:** Phone / Fax, Column Name: SMOBILE, Example: North America.
- Options:** 0xxxxxxxxx.
- Data Type:** Email, Column Name: SMAIL, Example: No examples available.
- Options:** SOURCE: RANDOM.
- Data Type:** Street Address, Column Name: SADDRESS, Example: No examples available.
- Options:** No options available.
- Data Type:** Number Range, Column Name: SID, Example: No examples available.
- Options:** Between 100 and 999.

MySQL SQL Output:

```
1 DROP TABLE IF EXISTS `myTable`;
2
3 CREATE TABLE `myTable` (
4     `id` mediumint(8) unsigned NOT NULL auto_increment,
5     `SNAME` varchar(255) default NULL,
6     `SMOBILE` varchar(100) default NULL,
7     `SMAIL` varchar(255) default NULL,
8     `SADDRESS` varchar(255) default NULL,
9     `SID` mediumint default NULL,
10    PRIMARY KEY (`id`),
11 ) AUTO_INCREMENT=1;
12
13 INSERT INTO `myTable` (`SNAME`, `SMOBILE`, `SMAIL`, `SADDRESS`, `SID`)
```

CSV Screenshot:

- Data Type:** Names, Column Name: CNAME, Example: Alex Smith.
- Options:** Name Surname, SOURCE: RANDOM.
- Data Type:** Phone / Fax, Column Name: CMOBILE, Example: North America.
- Options:** 0xxxxxxxxx.
- Data Type:** Email, Column Name: CMAIL, Example: No examples available.
- Options:** SOURCE: RANDOM.
- Data Type:** Date, Column Name: CBIRTHYEAR, Example: May 26, 2024.
- Options:** Jan 1, 1915 → May 23, 2025, Format code: y.
- Data Type:** Names, Column Name: CGENDER, Example: Alex Smith.
- Options:** F X M X.
- Data Type:** Number Range, Column Name: CID, Example: No examples available.
- Options:** Between 100 and 999.
- Data Type:** Street Address, Column Name: CADRESS, Example: No examples available.
- Options:** No options available.

CSV Output:

```
1 CNAME,CMOBILE,CMAIL,CBIRTHYEAR,CGENDER,CID,CADDRESS
2 Raymond Potter,0544414405,mauris.e@eol.com,1993,F,722,266-3441 Urna. Street
3 Alyssa Beard,0529986435,ne,urna.suscipit@notnull.ca,2017,F,733,Ap #195-2539 Nutrum Rd.
4 Raven Melendez,051187952,ipsum.donec.solicitudin@google.com,1979,M,385,"P.O. Box 347, 4713 Pharetra. Avenue"
5 Cain Mercado,0535481844,velit.justus@outlook.co.uk,2009,M,433,"4448 Non, Ave"
```

קובץ גיבוי :Backup.sql

הדגמת גיבוי:

The screenshot shows the 'Export' dialog in Oracle SQL Developer. The 'Tables' tab is selected, displaying a list of tables from the 'SYSTEM@XE' schema. The tables listed include: REPCATS_INSTANTIATION_DDL, REPCATS_KEY_COLUMNS, REPCATS_OBJECT_PARMS, REPCATS_OBJECT_TYPES, REPCATS_PARAMETER_COLUMN, REPCATS_PRIORITY, REPCATS_PRIORITY_GROUP, REPCATS_REFRESH_TEMPLATES, REPCATS_REFRESH_TYPE, REPCATS_REPOCATLOG, REPCATS_REPOCOLUMN, REPCATS_REPOGROUP_PRIVS, REPCATS_REPOOBJECT, REPCATS_REPOPROP, REPCATS_REPOSCHEMA, REPCATS_RESOL_STATS_CTRL, REPCATS_RESOL_STATS_METHOD, REPCATS_RESOLUTION_METHOD, REPCATS_RESOLUTION_STATISTICS, REPCATS_RUNTIME_PARMS, REPCATS_SITE_OBJECTS, REPCATS_SITE_USERS, REPCATS_SHAPEGROUP, REPCATS_TEMPLATE_OBJECTS, REPCATS_TEMPLATE_PARMS, REPCATS_TEMPLATE_PRIVS, REPCATS_TEMPLATE_SITES, REPCATS_TEMPLATE_STATUS, REPCATS_TEMPLATE_TARGETS, REPCATS_TREATMENT, REPCATS_USER_AUTHORIZATIONS, REPCATS_USER_VALUES, SQLPLUS_PRODUCT_PROFILE, STAFF, TPRECPYRADMIN, and TREATMENT. Below the table list, the 'Log' section contains various export options like 'Drop tables', 'Create tables', 'Delete records', etc. The 'Output file' is set to 'C:\Users\manit\Desktop\backup.sql'. The status bar at the bottom shows 'SYSTEM@XE' and 'Exporting... Done'.

מחיקה:

The screenshot shows the SQL tab in Oracle SQL Developer with the following SQL code:

```
SQL Output Statistics
DROP TABLE Patient;
-- Documentation: The Patient table is dropped as per the specified order.

-- Drop the Office table
DROP TABLE Office;
-- Documentation: The Office table is dropped as per the specified order.

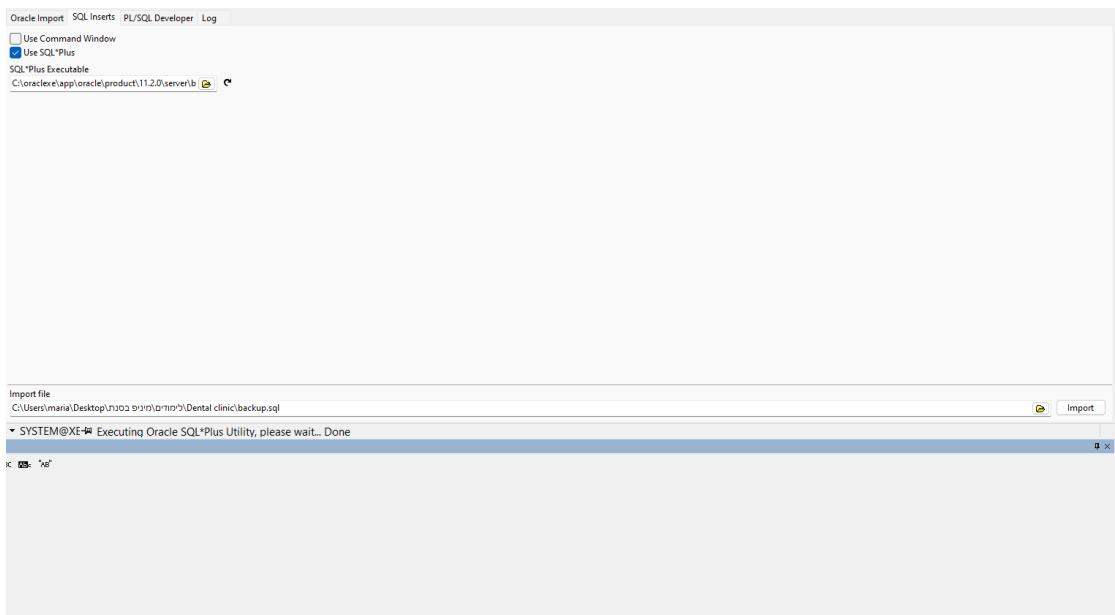
-- Drop the Doctor table
DROP TABLE Doctor;
-- Documentation: The Doctor table is dropped as per the specified order.

-- Drop the Staff table last
DROP TABLE Staff;
-- Documentation: The Staff table is dropped last as per the specified order.
```

The results pane below the SQL tab shows '(no result set)'.

The status bar at the bottom indicates the session is connected to 'SYSTEM@XE' and the command was completed 'Done in 0.031 seconds'.

שחזור:



שאילתות:
שאילתת select

Select 1:

הסבר: עקב ניסיון ייעול משאבי המרפאה נרצה לראות לכל פגישה של לקוחות מידע עליון וכמה עלתה כל הפגישה על כל הטיפולים השונים שבה ללקוח לעומת זאת למרפאה עבור לקוחות שלא ביצעו פגישה וטיפולים לא נראה מידע כמפורט

הקוד:

SELECT DISTINCT

,a.appointmentId
,p.cName
,p.cGender
,a.aDate
SUM(pa.TotalPrice) OVER (PARTITION BY a.AppointmentID) AS
,Cost_To_Patient
SUM(t.Price) OVER (PARTITION BY a.AppointmentID) AS Cost_To_Clinic
FROM system.patient p
LEFT JOIN system.appointment a ON p.cld = a.cld
LEFT JOIN system.tPreformedInA tpia ON a.AppointmentID =
tpia.AppointmentID
LEFT JOIN system.treatment t ON tpia.tld = t.tld
LEFT JOIN system.payment pa ON a.AppointmentID = pa.AppointmentID
WHERE p.cName IS NOT NULL
AND p.cGender IS NOT NULL
AND a.aDate IS NOT NULL
AND pa.TotalPrice IS NOT NULL
AND t.Price IS NOT NULL
;ORDER BY p.cName, a.aDate

תוצאות :

SQL Output Statistics

```

SELECT DISTINCT
    a.appointmentId,
    p.cName,
    p.cGender,
    a.aDate,
    SUM(pa.TotalPrice) OVER (PARTITION BY a.AppointmentID) AS Cost_To_Patient,
    SUM(t.Price) OVER (PARTITION BY a.AppointmentID) AS Cost_To_Clinic
FROM system.patient p
LEFT JOIN system.appointment a ON p.cId = a.cId
LEFT JOIN system.tPreformedInA tpia ON a.AppointmentID = tpia.AppointmentID
LEFT JOIN system.treatment t ON tpia.tId = t.tId
LEFT JOIN system.payment pa ON a.AppointmentID = pa.AppointmentID
WHERE p.cName IS NOT NULL
    AND p.cGender IS NOT NULL
    AND a.aDate IS NOT NULL
    AND pa.TotalPrice IS NOT NULL
    AND t.Price IS NOT NULL
ORDER BY p.cName, a.aDate;

```

	APPOINTMENTID	CNAME	CGENDER	ADATE	COST_TO_PATIENT	COST_TO_CLINIC
1	42733	Abraham Barlow	F	30/07/2029	3484	11175
2	78169	Acton Santana	M	21/05/2025	2506	42198
3	19548	Acton Santana	M	02/01/2027	1551	4856
4	59588	Addison Tran	M	19/10/2024	3972	26724
5	82439	Addison Tran	M	27/11/2025	984	19236
6	26926	Alan Head	F	28/07/2027	2470	6273
7	37647	Alexa Nielsen	F	15/07/2028	2960	43611
8	41673	Alexa Nielsen	F	10/11/2028	1390	42618
9	14715	Aline Ramos	M	16/07/2028	2462	33372
10	96463	Aline Ramos	M	11/11/2028	2281	658
11	43637	Alma Herrera	M	07/03/2027	3988	19561
12	24984	Alma Welch	M	13/01/2028	7112	17336
13	51446	Alma Welch	M	25/04/2028	1649	1145
14	84414	Althea Randolph	M	16/03/2025	4654	2169
15	89616	Althea Randolph	M	29/09/2027	4478	28901
16	33732	Anika Hopper	F	02/07/2027	2054	12119
17	89358	Anika Hopper	F	11/07/2028	736	876
18	92587	Arden Riley	M	02/04/2024	4476	33796
19	57934	Arden Riley	M	08/04/2025	795	7314
20	44623	Ariana Macias	F	20/09/2027	3027	26509
21	23487	Ashely Wilcox	F	24/09/2024	2628	17157

Select 2:

הסבר: כחלק מניהול המלאי נרצה לבדוק מיילו חומרים יש מעט מד' ובנוסף הם שימושיים.

נציג כל מוצר שיש ממנו פחות מ200 ועם הוא משוויר לפחות 5 תורמים

הקוד:

```

SELECT m.mId, m.mName, m.Amount, COUNT(DISTINCT tpia.AppointmentID)
AS appointment_count
FROM system.material m
JOIN system.musedint mui ON m.mId = mui.mId
JOIN system.treatment t ON mui.tId = t.tId
JOIN system.tpreformedina tpia ON t.tId = tpia.tId
WHERE m.Amount < 200
GROUP BY m.mId, m.mName, m.Amount
HAVING COUNT(DISTINCT tpia.AppointmentID) >= 5
;ORDER BY m.mId

```

תוצאות:

MID	MNAME	AMOUNT	APPOINTMENT_COUNT
1	Ointment Klsbdv	88	5
2	Antibiotic Nvgj	70	5
3	Anesthetic Naha	144	7
4	Antibiotic Qnrr	173	5
5	Pills Zjorvz	3	5
6	Septanest AjcyI	99	8
7	Septanest KqsdP	55	5
8	Anesthetic Swift	8	5
9	Septanest Wvlys	160	7
10	Antibiotic Tdtw	142	5
11	Ointment Xmaudd	196	6
12	Pills Zirneu	158	10
13	Ointment Ppjadm	153	5
14	Anesthetic Spnm	61	9
15	Antibiotic Itmt	60	6
16	Anesthetic Wwjx	86	8
17	Ointment Azqill	169	5
18	Ointment Cjgny	17	5
19	Anesthetic Atyu	114	5
20	Ointment Udfwh	41	5
21	Septanest Lzxau	137	5
22	Antibiotic Yxl	93	5
23	Pills Fanzbo	114	6
24	Pills Gnfrr	80	5
25	Anesthetic Rpil	173	5
26	Septanest Mktnj	59	5
27	Anesthetic Ukof	117	6
28	Anesthetic Khqg	93	5
29	Anesthetic Lyeg	7	5
30	Ointment Cfuzwa	192	7

Select 3:

הסבר:

נרצה להציג את עובדי המשרד הייעלים ביותר,

עובד משרד יעיל הוא עובד אשר קבוע לפחות 10 תורים, נציג אותו על פי מספר הפגישות

שהקבעו

קוד:

```

SELECT o.sId, s.sName, COUNT(DISTINCT a.AppointmentID) AS
    appointment_count
FROM system.office o
JOIN system.staff s ON o.sId = s.sId
JOIN system.omakea oma ON o.sId = oma.sId
JOIN system.appointment a ON oma.AppointmentID = a.AppointmentID
GROUP BY o.sId, s.sName
HAVING COUNT(DISTINCT a.AppointmentID) >= 10
;ORDER BY appointment_count DESC

```

תוצאות:

The screenshot shows a SQL query window with the following content:

```
SQL Output Statistics
SELECT o.sId, s.sName, COUNT(DISTINCT a.AppointmentID) AS appointment_count
FROM system.office o
JOIN system.staff s ON o.sId = s.sId
JOIN system.oMkaea oma ON o.sId = oma.sId
JOIN system.appointment a ON oma.AppointmentID = a.AppointmentID
GROUP BY o.sId, s.sName
HAVING COUNT(DISTINCT a.AppointmentID) >= 10
ORDER BY appointment_count DESC;
```

The results table has columns SID, SNAME, and APPOINTMENT_COUNT. The data is as follows:

SID	SNAME	APPOINTMENT_COUNT
1	James White	14
2	Todd Jimenez	13
3	Charl Walker	12
4	Indigo Rios	12
5	Isabella Lewis	11
6	Hayden Green	11
7	Macey Leonard	11
8	Silas Berger	10
9	Josiah Pacheco	10
10	Caleb Battle	10
11	Jacqueline Fleming	10
12	Nayda Olson	10
13	Gil Morse	10

Select 4:

הסבר:

בשביל לבצע סקר ולהבין יותר טוב את קהל היעד של המרפאה, המרפאה רוצה להבין את פרופיל הגילאים של המטופלים שלה באופן מדויק יותר ולתכנן את השירותים והטיפולים שלה בהתאם לצרכיהם הספציפיים. היא ממיינת את המטופלים שלה בשאלתה זו לפי גיל ממוצע

קוד:

SELECT

,p.cGender

```
,ROUND(AVG(EXTRACT(YEAR FROM SYSDATE) - p.cBirthYear)) AS avg_age
COUNT(DISTINCT p.cId) AS patient_count
FROM system.patient p
JOIN system.appointment a ON p.cId = a.cId
WHERE a.aDate >= ADD_MONTHS(SYSDATE, -6)
AND a.aDate <= SYSDATE
GROUP BY p.cGender
;ORDER BY avg_age DESC
```

תוצאות

SQL Output Statistics

```
SELECT
    p.cGender,
    ROUND(AVG(EXTRACT(YEAR FROM SYSDATE) - p.cBirthYear)) AS avg_age,
    COUNT(DISTINCT p.cId) AS patient_count
FROM system.patient p
JOIN system.appointment a ON p.cId = a.cId
WHERE a.aDate >= ADD_MONTHS(SYSDATE, -6)
    AND a.aDate <= SYSDATE
GROUP BY p.cGender
ORDER BY avg_age DESC;
```

	CGENDER	AVG_AGE	PATIENT_COUNT
▶	1 M	60	8
	2 F	44	15

שאילתת מחיקה:

1

הסבר:

התגלה כי חלה טעות ברישום הטיפולים שעתידיים להתקיים בתורים בחודש ממרץ 2026.

המציאות תרצה למחוק את הנתונים כדי לבצע כניסה ידנית מחדש

קוד:

--Show the treatments that are in appointments in march 2026 (To see before and after)

```
SELECT tpia.*, a.ADATE
FROM system.tpreformedina tpia
INNER JOIN system.appointment a ON a.AppointmentID = tpia.AppointmentID
WHERE EXTRACT(MONTH FROM a.aDate) = 3
      AND EXTRACT(YEAR FROM a.aDate) = 2026;
```

--Delete them

```
DELETE FROM system.tpreformedina tpia
WHERE tpia.AppointmentID IN (
    SELECT a.AppointmentID
    FROM system.appointment a
    WHERE EXTRACT(MONTH FROM a.aDate) = 3
          AND EXTRACT(YEAR FROM a.aDate) = 2026
);
```

(בשאילתת מחיקה הוספנו חוץ לשאילתת המחיקה עוד קוד select שיוכל להראות את הנתונים לפני ואחרי המחיקה)

齊лом הנתונים לפני מחיקה:

SQL Output Statistics

```
--Show the treatments that are in appointments in march 2026 (To see before and after)
SELECT tpia.*, a.ADATE
FROM system.tpreformedina tpia
INNER JOIN system.appointment a ON a.AppointmentID = tpia.AppointmentID
WHERE EXTRACT(MONTH FROM a.ADATE) = 3
      AND EXTRACT(YEAR FROM a.ADate) = 2026;

--Delete them
DELETE FROM system.tpreformedina tpia
WHERE tpia.AppointmentID IN (
    SELECT a.AppointmentID
    FROM system.appointment a
    WHERE EXTRACT(MONTH FROM a.ADate) = 3
          AND EXTRACT(YEAR FROM a.ADate) = 2026
);
```

	TID	APPOINTMENTID	ADATE
▶	1	16866	59864 31/03/2026 ▾
2	23243	52874 01/03/2026 ▾	
3	26551	53336 28/03/2026 ▾	
4	36999	57217 12/03/2026 ▾	
5	55142	59864 31/03/2026 ▾	
6	57499	62316 16/03/2026 ▾	
7	57734	52254 08/03/2026 ▾	
8	57734	53336 28/03/2026 ▾	
9	59859	38929 25/03/2026 ▾	
10	69323	59923 03/03/2026 ▾	
11	69548	52874 01/03/2026 ▾	
12	69989	35517 18/03/2026 ▾	
13	73179	59923 03/03/2026 ▾	
14	76324	52874 01/03/2026 ▾	
15	83436	59923 03/03/2026 ▾	
16	83715	31948 09/03/2026 ▾	
17	86775	44833 01/03/2026 ▾	
18	86933	76832 20/03/2026 ▾	
19	87911	84117 07/03/2026 ▾	
20	99588	47365 03/03/2026 ▾	

13:27 SYSTEM@XE 20 rows selected in 0.047 seconds

齊лом הנתונים אחרי המבוקש:

SQL Output Statistics

```
--Show the treatments that are in appointments in march 2026 (To see before and after)
SELECT tpia.*, a.ADATE
FROM system.tpreformedina tpia
INNER JOIN system.appointment a ON a.AppointmentID = tpia.AppointmentID
WHERE EXTRACT(MONTH FROM a.ADATE) = 3
      AND EXTRACT(YEAR FROM a.ADate) = 2026;

--Delete them
DELETE FROM system.tpreformedina tpia
WHERE tpia.AppointmentID IN (
    SELECT a.AppointmentID
    FROM system.appointment a
    WHERE EXTRACT(MONTH FROM a.ADate) = 3
          AND EXTRACT(YEAR FROM a.ADate) = 2026
);
```

	TID	APPOINTMENTID	ADATE
--	-----	---------------	-------

הסבר:

כחלק מהמעבר למערכת החדשה, הנהלת המרפאה החליטה שכל התשלומים והחיבורים שבוצעו בשנת 2023 ימחקו ממסד הנתונים הישן, כדי להתחיל את השנה החדשה עם מסד נתונים נקי ומעודכן.

קוד:

payment deateles before delete - select 2023--

```
SELECT p.Id, p.TotalPrice, p.pDate, a.aDate
FROM system.Payment p
```

```
JOIN system.Appointment a ON p.AppointmentID = a.AppointmentID
```

```
;WHERE EXTRACT(YEAR FROM p.pDate) = 2023
```

delete 2023--

```
DELETE FROM system.Payment p
```

```
;WHERE EXTRACT(YEAR FROM p.pDate) = 2023
```

צילום הנתונים לפני מתיקת:

```
-- 2023 payment deateles before delete
SELECT p.Id, p.TotalPrice, p.pDate, a.aDate
FROM system.Payment p
JOIN system.Appointment a ON p.AppointmentID = a.AppointmentID
WHERE EXTRACT(YEAR FROM p.pDate) = 2023;

-- 2023 delete
--DELETE FROM system.Payment p
--WHERE EXTRACT(YEAR FROM p.pDate) = 2023;
```

ID	TOTALPRICE	PDATE	ADATE
40001	100	01/05/2023	01/05/2023
40002	151	02/05/2023	02/05/2023
40003	75	03/05/2023	03/05/2023
40004	200	04/05/2023	04/05/2023
40005	126	05/05/2023	05/05/2023
40006	176	06/05/2023	06/05/2023

צילום נתונים לאחר המבוקש:

The screenshot shows a database management interface with two main sections: a SQL editor at the top and a results grid below it.

SQL Editor:

```
--2023 payment deateles before delete
SELECT p.Id, p.TotalPrice, p.pDate, a.aDate
FROM system.Payment p
JOIN system.Appointment a ON p.AppointmentID = a.AppointmentID
WHERE EXTRACT(YEAR FROM p.pDate) = 2023;

-- 2023 delete
--DELETE FROM system.Payment p
--WHERE EXTRACT(YEAR FROM p.pDate) = 2023;
```

Results Grid:

ID	TOTALPRICE	PDATE	ADATE

שאילתת update:

update 1

הסבר:

בשביל ניהול תקין של מלאי החומרים נרצה למצוא מי הם 20 החומרים המשמשים בהם הכי הרבה וזמן ממהם עוד עשר העשרה - בכל תוך עושים טיפולים שונים ולכל טיפול מסויכים חומרים דרושים לו.

קוד:

-- Update the amount for the top 20 most used materials

UPDATE system.material m

SET m.Amount = m.Amount + 10

WHERE m.mId IN (

SELECT mId

FROM (

SELECT m.mId

FROM system.material m

JOIN system.musedint mui ON m.mId = mui.mId

JOIN system.treatment t ON mui.tId = t.tId

JOIN system.tpreformedina tpi ON t.tId = tpi.tId

GROUP BY m.mId

ORDER BY COUNT(*) DESC

)

WHERE ROWNUM <= 20

);

-- Find the top 20 most used (in appointments) materials (use to see the before
and after the change)

SELECT *

FROM (

SELECT m.mId, m.mName, m.amount, COUNT(*) AS usageCount

```

        FROM system.material m
        JOIN system.musedint mui ON m.mId = mui.mId
        JOIN system.treatment t ON mui.tId = t.tId
        JOIN system.tpreformedina tpi ON t.tId = tpi.tId
        GROUP BY m.mId, m.mName, m.amount
        ORDER BY COUNT(*) DESC
    )
    WHERE ROWNUM <= 20

```

(גם כאן נשים שאלתת select שתראה את הנתונים לפני ואחרי עדכון)

צילום לפני עדכון :

-- Show the materials (before and after change)				
	MID	MNAME	AMOUNT	USAGECOUNT
▶	1	37596 Pills Zizme	148	10
	2	43389 Anesthetic Spkm	51	9
	3	49865 Anesthetic Wwvj	76	9
	4	49835 Ointment Czwa	202	8
	5	27219 Septanest Ajyl	89	8
	6	19558 Anesthetic Naha	134	7
	7	33316 Septanest Wdys	150	7
	8	99117 Ointment Czawa	182	7
	9	36647 Ointment Xnsaud	186	6
	10	93815 Anesthetic Ukol	107	6
	11	28379 Anesthetic Mbuu	215	6
	12	17771 Ointment Odald	289	6
	13	14183 Ointment Sjw	204	6
	14	29767 Ointment Vynolg	239	6
	15	69498 Pills Fenzbo	104	6
	16	63597 Pills Maylo	216	6
	17	44888 Antibiotic Itmt	50	6
	18	15869 Antibiotic Nvjj	60	5
	19	24124 Pills Zjorvz	3	5
	20	67682 Antibiotic Yx1	93	5

8 2021 SYSTEM@XE 20 rows selected in 0.047 seconds

צילום אחרי עדכון:

```
-- Update the amount for the top 20 most used materials
UPDATE system.material m
SET m.Amount = m.Amount + 10
WHERE m.mId IN (
    SELECT mId
    FROM (
        SELECT m.mId
        FROM system.material m
        JOIN system.musedint mui ON m.mId = mui.mId
        JOIN system.treatment t ON mui.tId = t.tId
        JOIN system.tpreformedina tpia ON t.tId = tpia.tId
        GROUP BY m.mId
        ORDER BY COUNT(*) DESC
    )
    WHERE ROWNUM <= 20
);

-- Show the materials (before and after change)
SELECT *
FROM (
    SELECT m.mId, m.mName, m.amount, COUNT(*) AS usageCount
    FROM system.material m
    JOIN system.musedint mui ON m.mId = mui.mId
    JOIN system.treatment t ON mui.tId = t.tId
    JOIN system.tpreformedina tpia ON t.tId = tpia.tId
    GROUP BY m.mId, m.mName, m.amount
    ORDER BY COUNT(*) DESC
)
WHERE ROWNUM <= 20
```

	MID	MNAME	AMOUNT	USAGECOUNT
1	37596	Pills Zizmeu	158	10
2	43389	Anesthetic Spxm	61	9
3	49899	Anesthetic Wwxj	86	9
4	27219	Septanest Ajcyl	99	8
5	12353	Anesthetic Vfmi	212	8
6	15959	Anesthetic Naha	144	7
7	99117	Ointment Cfuzwa	192	7
8	33316	Septanest Wylys	160	7
9	24183	Septanest	214	6
10	63597	Pills Mlavz	226	6
11	28379	Anesthetic Mbuw	225	6
12	69498	Pills Fanzbo	114	6
13	93815	Anesthetic Ukok	117	6
14	17776	Ointment Ddoald	299	6

:2 update

הסבירו:

בשל יוקר המחיה שפגע בישראל בשנת 2023 הוחלט להעלות את מחירי הטיפול במרפאות השנתיים

המרפאה تعدכן את המחיר של כל הטיפולים שבוצעו מתחילת השנה, ומעלה אותו ב-5%

קוד:

-- the Select

```
SELECT t.tId, t.tType, t.Description, t.Price
FROM system.Treatment t
JOIN system.TPreformedInA tpia ON t.tId = tpia.tId
JOIN system.Appointment a ON tpia.AppointmentID = a.AppointmentID
WHERE a.aDate >= TO_DATE('2023-01-01', 'YYYY-MM-DD')
ORDER BY a.aDate;
```

-- the update query

```

UPDATE system.Treatment
SET Price = Price * 1.05
WHERE tId IN (
    SELECT t.tId
    FROM system.Treatment t
    JOIN system.TPreformedInA tPIA ON t.tId = tPIA.tId
    JOIN system.Appointment a ON tPIA.AppointmentID = a.AppointmentID
    WHERE a.aDate >= TO_DATE('2023-01-01', 'YYYY-MM-DD')
);

```

צילום לפני עדכון:

SQL Window - SELECT t.tId, t.type, t.Description, t.Price, a.aDate FROM system.Treatment t JOIN system.TPreformedInA tPIA ON t.tId = tPIA.tId JOIN system.Appointment a ON tPIA.AppointmentID = a.AppointmentID WHERE a.aDate >= TO_DATE('2023-01-01', 'YYYY-MM-DD') ORDER BY a.aDate;

TID	TTTYPE	DESCRIPTION	PRICE	ADATE
1	Orthodontics	Braces	10124	02/05/2023
2	Rehabilitation	Extraction	12521	03/05/2023
3	Aesthetic	Gum	7131	06/05/2023
4	Rehabilitation	Teeth filling	4955	18/01/2024
5	Orthodontics	Teeth filling	2637	18/01/2024
6	Orthodontics	Teeth filling	10385	21/01/2024
7	Aesthetic	Teeth filling	10105	21/01/2024
8	Rehabilitation	Teeth filling	6749	23/01/2024
9	Aesthetic	Teeth filling	10489	17/02/2024
10	Rehabilitation	Braces	1035	17/02/2024
11	Aesthetic	Teeth filling	6242	09/03/2024
12	Orthodontics	Teeth filling	7421	09/03/2024
13	Rehabilitation	Gum	14785	19/03/2024
14	Aesthetic	Gum	6273	02/04/2024
15	Aesthetic	Braces	10625	02/04/2024
16	Aesthetic	Teeth filling	5749	05/04/2024
17	Aesthetic	Braces	8510	05/04/2024

SYSTEM@XE 378 rows selected in 0.264 seconds

צילום לאחר עדכון:

SQL Output Statistics

```
-- the Select
SELECT t.tId, t.tType, t.Description, t.Price,a.aDate
FROM system.Treatment t
JOIN system.TPreformedInA tpia ON t.tId = tpia.tId
JOIN system.Appointment a ON tpia.AppointmentID = a.AppointmentID
WHERE a.aDate >= TO_DATE('2023-01-01', 'YYYY-MM-DD')
ORDER BY a.aDate;

-- the update query
--UPDATE system.Treatment
--SET Price = Price * 1.05
--WHERE tId IN (
--SELECT t.tId
--FROM system.Treatment t
--JOIN system.TPreformedInA tpia ON t.tId = tpia.tId
--JOIN system.Appointment a ON tpia.AppointmentID = a.AppointmentID
--WHERE a.aDate >= TO_DATE('2023-01-01', 'YYYY-MM-DD')
--);

```

TID	TTYPE	DESCRIPTION	PRICE	ADATE
1	93545	Orthodontics Braces	13567	02/05/2023
2	82416	Rehabilitation Extraction	16779	03/05/2023
3	86775	Aesthetic Gum	9556	06/05/2023
4	45421	Rehabilitation Teeth filling	6640	18/01/2024
5	73179	Orthodontics Teeth filling	3533	18/01/2024
6	35666	Orthodontics Teeth filling	13916	21/01/2024
7	12265	Aesthetic Teeth filling	13542	21/01/2024
8	89886	Rehabilitation Teeth filling	9044	23/01/2024
9	64213	Aesthetic Teeth filling	14055	17/02/2024
10	52941	Rehabilitation Braces	1387	17/02/2024
11	87874	Aesthetic Teeth filling	8364	09/03/2024
12	79982	Orthodontics Teeth filling	9946	09/03/2024

18:3 hadasz@Xe 12 rows selected in 0.032 seconds (more...)

ABC "N"

שאלות עם פרמטרים

שאלת 1 עם פרמטר :

הסבר:

השאלה היא-

עליך ליצור תקציב הוחלט לעשות בירור עבור עובדים אשר לא תורמים למשרד, רופא X נמצא לא פעיל,

נרצה בזמן אותו לבירור, ובינתיים נשנה את כל הפגישות שלו כך שרופא אחר באותו התחום יעשה אותן במקום מקומו.

השאלה תמציא את הפגישות של הרופא המושהה, תחפש רופא אחר באותו התחום מומחיות באופן אקראי ותשנה כך שהוא יעשה את הפגישה

קוד:

--Move the appointments that belong to the suspended doctor to a random doctor with the same Specialties

UPDATE system.appointment

SET sId = (

SELECT sId FROM (

SELECT d.sId

FROM system.doctor d

WHERE d.Specialties = (

SELECT Specialties

FROM system.doctor

WHERE sId = (&<name=Doctor_id list="select distinct doc.sId from doctor doc order by doc.sId" hint="Doctor Id has 5 digits" required= true type=integer >)

)

AND d.sId <> &Doctor_id

ORDER BY dbms_random.value

)

WHERE ROWNUM = 1

)

WHERE sId = &Doctor_id;

--The followings are just to check that everything worked well, No need to run

--View the doctors with the same Specialties as the suspended doctor

SELECT * from system.doctor d

WHERE d.Specialties = (

SELECT Specialties

FROM system.doctor

WHERE sId = (&<name=Doctor_id list="select distinct doc.sId from doctor doc
order by doc.sId" hint="Doctor Id has 5 digits" required= true type=integer >)

)

AND d.sId <> &Doctor_id

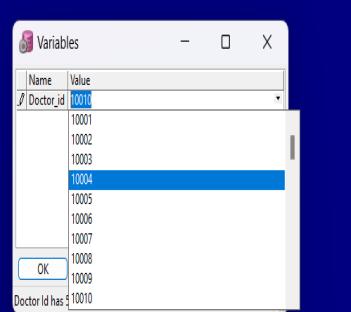
--View the appointments of the suspended doctor (Will be empty after the
change)

SELECT * from system.appointment

WHERE sId = (&<name=Doctor_id list="select distinct doc.sId from doctor doc
order by doc.sId" hint="Doctor Id has 5 digits" required= true type=integer >)

הכנסת ת"ז של רופא מומחה בפונקציית העדכון:

```
--Move the appointments that belong to the suspended doctor to a random doctor with the same Specialties
UPDATE system.appointment
SET sId = (
    SELECT sid FROM (
        SELECT d.sId
        FROM system.doctor d
        WHERE d.Specialties = (
            SELECT Specialties
            FROM system.doctor
            WHERE sId = (&<name=Doctor_id list="select distinct doc.sId from doctor doc order by doc.sId" hint="Doctor Id has 5 digits" required= true type=integer >)
        )
        AND d.sId <> &Doctor_id
        ORDER BY dmra_random.value
    )
    WHERE ROWNUM = 1
)
WHERE sId = &Doctor_id;
```



רופאים עם אותה מומחיות כמו הרופא המומחה:

```
--View the doctors with the same Specialties as the suspended doctor
SELECT * from system.doctor d
WHERE d.Specialties = (
    SELECT Specialties
    FROM system.doctor
    WHERE sid = (4<name=Doctor_id list="select distinct doc.sid from doctor doc order by doc.sid" hint="Doctor Id has 5 digits" required= true type=integer >
)
AND d.sid <> 4<doctor_id>
```

LICENSE	SPECIALTIES	SID
1 D757117	Dentist	43931
2 E209644	Dentist	62435
3 O465110	Dentist	81184
4 K655114	Dentist	73375
5 H13872	Dentist	79819
6 X325759	Dentist	31696
7 X491823	Dentist	25922
8 E67996	Dentist	87909
9 P768257	Dentist	69296
10 T588884	Dentist	46150
11 P742424	Dentist	20156
12 F11833	Dentist	10010
13 J098247	Dentist	13261
14 Q224232	Dentist	69880
15 K392960	Dentist	31845
16 N449816	Dentist	25991
17 A359840	Dentist	32397
18 B35680	Dentist	50906
19 L131155	Dentist	84301
20 D965117	Dentist	70746
21 D9651174	Dentist	91660
22 W006605	Dentist	95541
23 Z569941	Dentist	96640
24 P260530	Dentist	32568
25 N481123	Dentist	74972

48:4 SYSTEM@XE 40 rows selected in 0.093 seconds (more...)

פגישות של הרופא המושהה לפני העדכון:

```
--View the appointments of the suspended doctor (Will be empty after the change)
SELECT * from system.appointment
WHERE sid = (4<name=Doctor_id list="select distinct doc.sid from doctor doc order by doc.sid" hint="Doctor Id has 5 digits" required= true type=integer >
```

ADATE	APPOINTMENTID	SID	CID
1 01/03/2026	52874	10010	648
2 29/12/2025	65466	10010	250
3 31/01/2029	34362	10010	134

פגישות של הרופא המושהה לאחר העדכון:

```
--View the appointments of the suspended doctor (Will be empty after the change)
SELECT * from system.appointment
WHERE sid = (4<name=Doctor_id list="select distinct doc.sid from doctor doc order by doc.sid" hint="Doctor Id has 5 digits" required= true type=integer >
```

ADATE	APPOINTMENTID	SID	CID

שאילתה 2 עם פרמטר:

הסבר:

נרצה לדעת בתוך תקופת זמן מסוימת אילו מטופלים לטפלו בסוג טיפול מסוים.

אפשר כניסה טווח תאריכים וכן סוג טיפול והאם למיין את הרשימה בסדר עולה או יורד

קוד:

```

SELECT DISTINCT
    p.cId ,p.cName,p.cGender,a.aDate,t.tType,t.Price
FROM system.patient p
JOIN system.appointment a ON p.cId = a.cId
JOIN system.tpreformedina tbia ON a.AppointmentID = tbia.AppointmentID
JOIN system.treatment t ON tbia.tId = t.tId
WHERE a.aDate BETWEEN &<name=start_date hint="Enter start date (dd/mm/yyyy)" type=date>
    AND &<name=end_date hint="Enter end date (dd/mm/yyyy)" type=date>
AND t.tType IN (&<name=treatment_types list=""Rehabilitation','Orthodontics',
'Aesthetic'" multiselect="true" required=true>)
ORDER BY p.cId &<name="Desc or Asc order" checkbox="desc,asc">;

```

Result using asc:

SQL Output Statistics

```

SELECT DISTINCT
    p.cId ,p.cName,p.cGender,a.aDate,t.tType,t.Price
FROM system.patient p
JOIN system.appointment a ON p.cId = a.cId
JOIN system.tpreformedina tbia ON a.AppointmentID = tbia.AppointmentID
JOIN system.treatment t ON tbia.tId = t.tId
WHERE a.aDate BETWEEN &<name=start_date hint="Enter start date (dd/mm/yyyy)" type=date>
    AND &<name=end_date hint="Enter end date (dd/mm/yyyy)" type=date>
AND t.tType IN (&<name=treatment_types list=""Rehabilitation','Orthodontics',
'Aesthetic'" multiselect="true" required=true>)
ORDER BY p.cId &<name="Desc or Asc order" checkbox="desc,asc">;

```

Variables

Name	Value
start_date	01/01/2024
end_date	01/01/2025
treatment_types	'Rehabilitation','Aesthetic'
Desc or Asc order	<input checked="" type="checkbox"/>

Enter start date (dd/mm/yyyy)

10:04 SYSTEM@XE Initializing...

CID	CNAME	CGENDER	ADATE	TTYPE	PRICE
1	Cole Bernard	F	18/01/2024	Rehabilitation	4955
2	Brittany Guerra	M	09/08/2024	Rehabilitation	4332
3	Cora Maddox	M	18/07/2024	Rehabilitation	14537
4	Beau Ewing	M	18/11/2024	Aesthetic	3528
5	Pearl Warner	M	05/04/2024	Aesthetic	5749
6	Pearl Warner	M	05/04/2024	Aesthetic	8510
7	Pearl Warner	M	05/04/2024	Aesthetic	10354
8	Pearl Warner	M	05/04/2024	Rehabilitation	13998
9	Pearl Warner	M	05/04/2024	Rehabilitation	14991
10	Connor Gardner	M	21/12/2024	Rehabilitation	10860
11	Addison Tran	M	19/10/2024	Aesthetic	3162
12	Addison Tran	M	19/10/2024	Aesthetic	10200
13	Jesse Dominguez	F	09/03/2024	Aesthetic	6242
14	Amaya Watson	F	12/07/2024	Rehabilitation	91
15	Giselle Jimenez	M	23/01/2024	Rehabilitation	6749
16	Hyatt Gay	F	18/06/2024	Rehabilitation	9141
17	Madison Lott	M	17/02/2024	Aesthetic	10489
18	Madison Lott	M	17/02/2024	Rehabilitation	1035
19	Brady Potts	F	15/12/2024	Aesthetic	12652
20	Reagan Franks	F	10/11/2024	Aesthetic	4231
21	Reagan Franks	F	10/11/2024	Aesthetic	11901
22	Benjamin Joyce	F	27/04/2024	Aesthetic	3439
23	Benjamin Joyce	F	27/04/2024	Rehabilitation	4366
24	Madison Madden	F	21/01/2024	Aesthetic	10105
25	Blythe Stephenson	F	19/03/2024	Rehabilitation	14785
26	Megan Molina	F	28/07/2024	Aesthetic	5821
27	Shellie Burke	F	10/08/2024	Aesthetic	11926
28	Jordan Mathis	M	16/05/2024	Aesthetic	11068
29	Tate Wolfe	F	20/09/2024	Rehabilitation	9004

Result using desc:

The screenshot shows the Oracle SQL Developer interface. On the left, there is a code editor window containing a SQL query. On the right, there is a 'Variables' dialog box with several entries. Below the code editor is a table with 30 rows of data.

```

SQL Output Statistics
SELECT DISTINCT
    p.cId, p.cName, p.cGender, a.aDate, t.tType, t.Price
FROM system.patient p
JOIN system.appointment a ON p.cId = a.cId
JOIN system.treatmentmedina tmia ON a.AppointmentID = tmia.AppointmentID
JOIN system.treatment t ON tmia.tId = t.tId
WHERE a.aDate Between :start_date AND :end_date
    AND t.tType IN (:treatment_types)
    AND a.aDate Between :start_date AND :end_date
    AND t.tType IN (:treatment_types)
ORDER BY p.cId <?name="Desc or Asc order" checkbox="desc,asc">;

```

CID	CNAME	CGENDER	A DATE	TTYPE	PRICE
1	20006 Fiona Anderson	F	16/08/2024	Aesthetic	14902
2	984 Ashely Wilcox	F	24/09/2024	Rehabilitation	7383
3	975 Brody Alvarado	M	12/12/2024	Aesthetic	1317
4	981 Braxton Gandy	M	03/04/2024	Aesthetic	1073
5	941 Aiden Riley	M	03/04/2024	Aesthetic	10625
6	890 Regina Puckett	F	19/09/2024	Rehabilitation	1259
7	885 Tate Wolfe	F	20/09/2024	Rehabilitation	9004
8	833 Jordan Mathis	M	16/05/2024	Aesthetic	11068
9	743 Shellie Burke	F	10/08/2024	Aesthetic	11926
10	717 Megan Molina	F	28/01/2024	Aesthetic	5821
11	690 Michael Stevenson	F	19/01/2024	Rehabilitation	14783
12	676 Madison Madden	F	21/01/2024	Aesthetic	1000
13	624 Benjamin Joyce	F	27/04/2024	Aesthetic	3439
14	624 Benjamin Joyce	F	27/04/2024	Rehabilitation	4366
15	618 Reagan Franks	F	10/11/2024	Aesthetic	4231
16	618 Reagan Franks	F	10/11/2024	Aesthetic	11901
17	572 Brady Potts	F	15/12/2024	Aesthetic	12652
18	553 Madison Lott	M	10/02/2024	Aesthetic	10489
19	553 Madison Lott	M	17/02/2024	Rehabilitation	1050
20	495 Hyatt Gay	F	18/06/2024	Rehabilitation	9141
21	468 Giselle Jimenez	M	23/01/2024	Rehabilitation	6749
22	412 Amaya Watson	F	12/07/2024	Rehabilitation	91
23	387 Jesse Dominguez	F	09/03/2024	Aesthetic	6242
24	324 Addison Tan	M	19/10/2024	Aesthetic	3162
25	324 Addison Tan	M	19/10/2024	Aesthetic	10200
26	314 Madison Warner	M	10/04/2024	Rehabilitation	10800
27	309 Pearl Warner	M	05/04/2024	Aesthetic	749
28	309 Pearl Warner	M	05/04/2024	Aesthetic	8510
29	309 Pearl Warner	M	05/04/2024	Aesthetic	10354

שאילתה 3 עם פרמטר:

הסבר:

לצורך תרגול בוחן פתע של מס הכנסה - המזכירות צריכות להיות מותרגלות לשלו' נתונים על כל פרטיינט

- אż השאלתה תחזיר לי פרטי מטופל מסוים, הטיפולים שלו ומתי התוור שלו לטיפול

קוד:

SELECT

system.p.cId AS PatientID,

system.p.cName AS PatientName,

system.t.tId AS TreatmentID,

system.t.tType AS TreatmentType,

system.a.aDate AS AppointmentDate

FROM

system.Patient p

JOIN

system.Appointment a ON system.p.cId = system.a.cId

JOIN

```

system.TPreformedInA tp ON system.a.AppointmentID =
    system.tp.AppointmentID
JOIN
system.Treatment t ON system.tp.tId = system.t.tId
WHERE
system.p.cId = &list
ORDER BY
system.a.aDate;

```

תוצאה:

The screenshot shows a SQL interface with the following details:

- SQL Editor:** Contains the following query:


```

SELECT
    system.p.cId AS PatientID,
    system.p.cName AS PatientName,
    system.p.cId AS PatientID,
    system.t.tId AS TreatmentID,
    system.t.tType AS TreatmentType,
    system.a.aDate AS AppointmentDate
FROM
    system.Patient p
JOIN
    system.Appointment a ON system.p.cId = system.a.cId
JOIN
    system.TPreformedInA tp ON system.a.AppointmentID = system.tp.AppointmentID
JOIN
    system.Treatment t ON system.tp.tId = system.t.tId
WHERE
    system.p.cId =(&name=Patient_id list="select distinct p.cId from Patient p order by p.cId" hint="Patient Id has 5 digits" required= true type=integer >)
ORDER BY
    system.a.aDate;
      
```
- Variables Dialog:** Shows a single entry: Patient_id = 20006. A note below says "Patient Id has 5 digits".
- Results Grid:** Displays 5 rows of data for Fiona Anderson, with columns: PATIENTID, PATIENTNAME, TREATMENTID, TREATMENTTYPE, and APPOINTMENTDATE.

PATIENTID	PATIENTNAME	TREATMENTID	TREATMENTTYPE	APPOINTMENTDATE
1	20006 Fiona Anderson ...	86775	Aesthetic	06/05/2023
2	20006 Fiona Anderson ...	89835	Orthodontics	16/07/2024
3	20006 Fiona Anderson ...	64425	Aesthetic	16/08/2024
4	20006 Fiona Anderson ...	87568	Orthodontics	16/08/2024
5	20006 Fiona Anderson ...	22389	Orthodontics	05/08/2028

שאילה 4 עם פרמטר:

הסבר:

השאילה מוצאת דוקטורים שעובדים בסוף שבוע בחודש מסוים ומספקים טיפולים שעולים יותר מסכום כלשהו

קוד:

```

SELECT
    s.sName AS DoctorName,
    d.Specialties,
    a.aDate,
    TO_CHAR(a.aDate, 'DAY') AS DayOfWeek,
    t.tType,
    t.Price

```

```

        FROM system.staff s
        JOIN system.doctor d ON s.sId = d.sId
        JOIN system.appointment a ON d.sId = a.sId
        JOIN system.tpreformedina tpia ON a.AppointmentID = tpia.AppointmentID
        JOIN system.treatment t ON tpia.tId = t.tId
WHERE a.aDate BETWEEN &<name=month_start hint="Enter first day of month
(YYYY-MM-01)" type=date>
        AND LAST_DAY(&<name=month_start hint="Enter first day of month
(YYYY-MM-01)" type=date>
        AND TO_CHAR(a.aDate, 'DY') IN ('SAT', 'SUN')
        AND t.Price > &<name=min_price hint="Enter minimum treatment price"
type=INTEGER>
ORDER BY a.aDate, s.sName;

```

תוצאות:

The screenshot shows the Oracle SQL Developer interface. On the left, the SQL tab displays the following query:

```

SQL Output Statistics
SELECT
    s.sName AS DoctorName,
    d.Specialties,
    a.aDate,
    TO_CHAR(a.aDate, 'DAY', 'NLS_DATE_LANGUAGE=ENGLISH') AS DayOfWeek,
    t.tType,
    t.Price
FROM system.staff s
JOIN system.doctor d ON s.sId = d.sId
JOIN system.appointment a ON d.sId = a.sId
JOIN system.tpreformedina tpia ON a.AppointmentID = tpia.AppointmentID
JOIN system.treatment t ON tpia.tId = t.tId
WHERE a.aDate BETWEEN &<name=start_date hint="Enter start date (dd/mm/yyyy)" type=date>
        AND &<name=end_date hint="Enter end date (dd/mm/yyyy)" type=date>
        AND TO_CHAR(a.aDate, 'DY', 'NLS_DATE_LANGUAGE=ENGLISH') IN ('5', '7') -- 1 for Sunday, 7 for Saturday
        AND t.Price > &<name=min_price hint="Enter minimum treatment price" type=INTEGER>
ORDER BY a.aDate, s.sName;

```

To the right of the SQL tab is a Variables dialog box. It contains three entries:

Name	Value
start_date	01/09/2025
end_date	30/09/2025
min_price	3000

At the bottom of the Variables dialog is a note: "Enter end date (dd/mm/yyyy)".

Below the Variables dialog is a grid showing the results of the query. The columns are: DOCTORNAME, SPECIALTIES, ADATE, DAYOFWEEK, TTYPE, and PRICE. The data is as follows:

DOCTORNAME	SPECIALTIES	ADATE	DAYOFWEEK	TTYPE	PRICE
1 Darryl Schmidt	Cosmetic	04/09/2025	THURSDAY	Aesthetic	7078
2 Darryl Schmidt	Cosmetic	04/09/2025	THURSDAY	Rehabilitation	3534
3 Darryl Schmidt	Cosmetic	04/09/2025	THURSDAY	Orthodontics	5043
4 Chris Brown	Orthodontist	27/09/2025	SATURDAY	Orthodontics	6154

אילוצים:

1.

: קוד:

--Create a default date for appointments

```
ALTER TABLE appointment  
MODIFY aDate DATE DEFAULT TRUNC(SYSDATE);
```

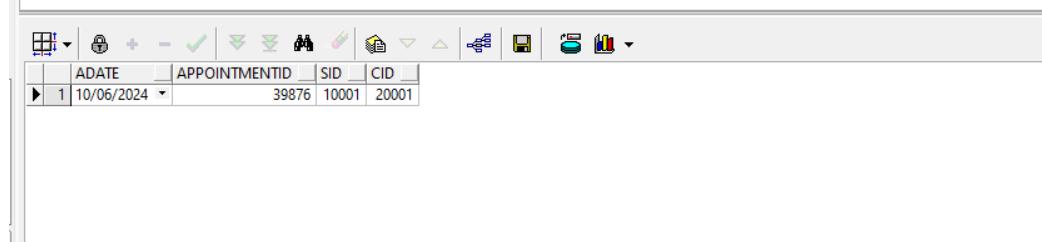
```
INSERT INTO Appointment ( AppointmentID, sId, cId) VALUES ( 39876, 10001,  
20001);
```

select * from appointment

where AppointmentID=39876

create a default date for appointments:

```
--Create a default date for appointments  
ALTER TABLE appointment  
MODIFY aDate DATE DEFAULT TRUNC(SYSDATE);  
  
INSERT INTO Appointment ( AppointmentID, sId, cId) VALUES ( 39876, 10001, 20001);  
  
select * from appointment  
where AppointmentID=39876
```



ID	DATE	APPOINTMENTID	SID	CID
1	10/06/2024	39876	10001	20001

2.

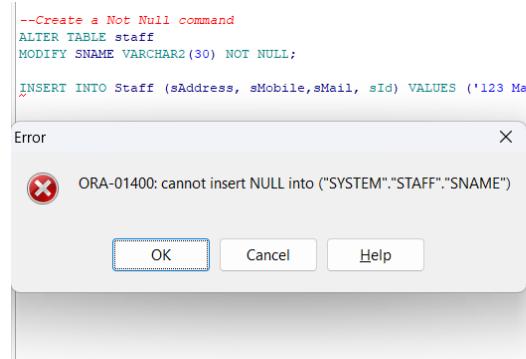
: Tip

--Create a Not Null command

```
ALTER TABLE staff  
MODIFY SNAME VARCHAR2(30) NOT NULL;
```

```
INSERT INTO Staff (sAddress, sMobile,sMail, sId) VALUES ('123 Main St',
'123456789', 'john@email.com', 10789);
```

Create a not null command:



The screenshot shows a SQL command window with the following code:

```
--Create a Not Null command
ALTER TABLE staff
MODIFY SNAME VARCHAR2(30) NOT NULL;

INSERT INTO Staff (sAddress, sMobile,sMail, sId) VALUES ('123 Main St', '123456789', 'john@email.com', 10789);
```

Below the code, an 'Error' dialog box is displayed. The message in the dialog is:

ORA-01400: cannot insert NULL into ("SYSTEM"."STAFF"."SNAME")

Buttons in the dialog box include OK, Cancel, and Help.

.3

:TIP

--Make sure that phone numbers start with 0

```
ALTER TABLE PATIENT
ADD CONSTRAINT check_mobile_patient
CHECK (CMOBILE LIKE '0%');
```

```
INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile,
cMail) VALUES (1990, '456 Oak Ave', 'Bob Smith', 'M', 25671, '987654321',
'bob@email.com');
```

select * from patient

Make sure that phone number start with 0:

The screenshot shows the Oracle SQL Developer interface. In the SQL tab, there is a red error message: "ORA-02290: check constraint (SYSTEM.CHECK_MOBILE_PATIENT) violated". This occurs because the mobile phone number '987654321' does not start with '0'. The error dialog has buttons for OK, Cancel, and Help.

```
--Make sure that phone numbers start with 0
ALTER TABLE PATIENT
ADD CONSTRAINT check_mobile_patient
CHECK (cMobile LIKE '0%');

INSERT INTO Patient (cBirthYear, cAddress, cName, cGender, cId, cMobile, cMail) VALUES (1990, '456 Oak Ave', 'Bob Smith', 'M', 25671, '987654321', 'bob@email.com');

select * from patient
```

.4

:717

--Make sure treatment price isn't <=0

ALTER TABLE treatment

ADD CONSTRAINT check_price

CHECK (PRICE > 0);

```
INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES
('Medication','Antibiotics',-15,60002,0.5);
```

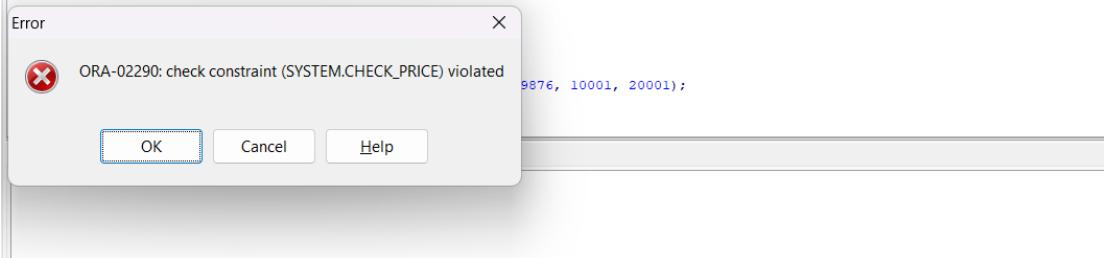
select * from treatment

Make sure treatment price is valid

```
--Make sure treatment price isn't <=0
ALTER TABLE treatment
ADD CONSTRAINT check_price
CHECK (PRICE > 0);

INSERT INTO Treatment (tType, Description, Price, tId, Time) VALUES ('Medication', 'Antibiotics', -15, 60002, 0.5);

select * from treatment
```



גיבוי:

The screenshot shows the 'Tables' section of the Oracle SQL Developer interface. It lists numerous tables under the schema 'REPCATS', such as REPCATS_INSTANTIATION_DDL, REPCATS_KEY_COLUMNS, REPCATS_LOGONS, REPCATS_OBJECT_TYPES, REPCATS_PARAMETER_COLUMN, REPCATS_PRIORITY, REPCATS_PRIORITY_GROUP, REPCATS_REFRESH_TEMPLATE, REPCATS_REFRESH_TEMPLATE, REPCATS_REPOLOG, REPCATS_REPOCOLUMN, REPCATS_REPOGROUP_PRIVS, REPCATS_REPOBJECT, REPCATS_REPOPROP, REPCATS_REPSHEMA, REPCATS_RESOL_STATS_CONTROL, REPCATS_RESOLUTION_METHOD, REPCATS_RESOLUTION_STATISTICS, REPCATS_RUNTIME_PARMS, REPCATS_SITE_OBJECTS, REPCATS_SITES_NEWS, REPCATS_SNAPGROUP, REPCATS_TEMPLATE_OBJECT, REPCATS_TEMPLATE_PARAMETERS, REPCATS_TEMPLATE_REFGROUPS, REPCATS_TEMPLATE_SITES, REPCATS_TEMPLATE_STATUS, REPCATS_TEMPLATE_TARGETS, REPCATS_TEMPLATE_TYPES, REPCATS_USER_AUTHORIZATIONS, REPCATS_USER_PARM_VALUES, and TREATMENT. The 'TREATMENT' table is highlighted with a blue selection bar at the bottom.

Below the table list, there is an 'Oracle Export' configuration panel. It includes options for 'Drop tables', 'Create tables', 'Truncate tables', 'Delete records', 'Disable triggers', 'Disable foreign key constraint', 'Include storage', 'Include privileges', 'Commit every 100 records (0 = never)', 'Zip', and 'Where clause'. The 'Output file' is set to 'C:\Users\datan\Desktop\2\جداول فريدة\New shortcut.lnk'. The connection is set to 'system@XE'.

פרוצדורה 1

הסביר - נרצה ליצור שיעבוד בצורה מקצועית.
יתבצעו בדיקות תקינות לכל הקליטים כמו האם התאריך עבר כבר, האם המטופל קיים ועוד
בנוסף נקבע גם את הטיפול שיתרחש בעתoor מעבר למועד התוור כי שהיה תמיד

הפרוצדורה-

```
CREATE OR REPLACE PROCEDURE create_appointment (
    p_doctor_id IN Doctor.sId%TYPE,
    p_patient_id IN Patient.cId%TYPE,
    p_appointment_date IN Appointment.aDate%TYPE,
    p_treatment_id IN Treatment.tId%TYPE,
    p_appointment_cursor OUT SYS_REFCURSOR
) AS
    -- Record type for appointment details
    TYPE appointment_record IS RECORD (
        appointment_id Appointment.AppointmentID%TYPE,
        doctor_name Staff.sName%TYPE,
        patient_name Patient.cName%TYPE,
        appointment_date Appointment.aDate%TYPE,
        appointment_type Treatment.tType%TYPE
    );
    v_appointment appointment_record;
    v_doctor_exists NUMBER;
    v_patient_exists NUMBER;
    v_treatment_exists NUMBER;
    v_appointment_id Appointment.AppointmentID%TYPE;
    v_current_date DATE := TRUNC(SYSDATE);
    v_doctor_appointments NUMBER;
    -- Custom exceptions
    e_invalid_doctor EXCEPTION;
    e_invalid_patient EXCEPTION;
    e_invalid_treatment EXCEPTION;
    e_past_date EXCEPTION;
    e_doctor_unavailable EXCEPTION;
BEGIN
    -- Check if doctor exists
    SELECT COUNT(*) INTO v_doctor_exists
    FROM Doctor
    WHERE sId = p_doctor_id;
    IF v_doctor_exists = 0 THEN
        RAISE e_invalid_doctor;
    END IF;
    -- Check if patient exists
    SELECT COUNT(*) INTO v_patient_exists
    FROM Patient
    WHERE cId = p_patient_id;
    IF v_patient_exists = 0 THEN
        RAISE e_invalid_patient;
    END IF;
    -- Check if treatment exists
    SELECT COUNT(*) INTO v_treatment_exists
    FROM Treatment
    WHERE tId = p_treatment_id;
    IF v_treatment_exists = 0 THEN
        RAISE e_invalid_treatment;
    END IF;
    -- Check if appointment date is in the future
    IF p_appointment_date <= v_current_date THEN
        RAISE e_past_date;
    END IF;
    -- Check doctor's availability (assuming max 10 appointments per day)
    SELECT COUNT(*) INTO v_doctor_appointments
    FROM Appointment
    WHERE sId = p_doctor_id AND aDate = p_appointment_date;
    IF v_doctor_appointments >= 10 THEN
        RAISE e_doctor_unavailable;
    END IF;
    -- Generate new appointment ID
    SELECT NVL(MAX(AppointmentID), 0) + 1 INTO v_appointment_id
    FROM Appointment;
    -- Insert new appointment
    INSERT INTO Appointment (AppointmentID, aDate, sId, cId)
    VALUES (v_appointment_id, p_appointment_date, p_doctor_id, p_patient_id);
    -- Insert treatment for the appointment
    INSERT INTO TREFORMEDINA (tId, AppointmentID)
    VALUES (p_treatment_id, v_appointment_id);
    -- Fetch appointment details
    SELECT a.AppointmentID, s.sName, p.cName, a.aDate, t.tType
    INTO v_appointment
    FROM Appointment a
    JOIN Staff s ON a.sId = s.sId;
```

```

create_appointment
-----[Code section]-----[Select]
> Parameter lis
> Declaration
> Code section
> -> Exception ha
 76   -- Insert new appointment
 77   INSERT INTO Appointment (AppointmentID, aDate, sid, cId)
 78   VALUES (v_appointment_id, p_appointment_date, p_doctor_id, p_patient_id);
 79
 80
 81   -- Insert treatment for the appointment
 82   INSERT INTO TPREFORMEDINA (tid, AppointmentID)
 83   VALUES (p_treatment_id, v_appointment_id);
 84
 85   -- Fetch appointment details
 86   SELECT a.AppointmentID, s.sName, p.cName, a.aDate, t.tType
 87   INTO v_appointment
 88   FROM Appointment a
 89   JOIN Staff s ON a.sid = s.sid
 90   JOIN Patient p ON a.cid = p.cid
 91   JOIN TPREFORMEDINA tpa ON a.AppointmentID = tpa.AppointmentID
 92   JOIN Treatment t ON tpa.tid = t.tid
 93   WHERE a.AppointmentID = v_appointment_id;
 94
 95   -- Return appointment details via REF CURSOR
 96   OPEN p_appointment_cursor FOR
 97     SELECT v_appointment.appointment_id AS appointment_id,
 98           v_appointment.doctor_name AS doctor_name,
 99           v_appointment.patient_name AS patient_name,
100          v_appointment.appointment_date AS appointment_date,
101          v_appointment.treatment_type AS treatment_type
102     FROM DUAL;
103
104   COMMIT;
105
106 EXCEPTION
107   WHEN e_invalid_doctor THEN
108     RAISE_APPLICATION_ERROR(-20001, 'Invalid doctor ID');
109   WHEN e_invalid_patient THEN
110     RAISE_APPLICATION_ERROR(-20002, 'Invalid patient ID');
111   WHEN e_invalid_treatment THEN
112     RAISE_APPLICATION_ERROR(-20003, 'Invalid treatment ID');
113   WHEN e_past_date THEN
114     RAISE_APPLICATION_ERROR(-20004, 'Appointment date must be in the future');
115   WHEN e_doctor_unavailable THEN
116     RAISE_APPLICATION_ERROR(-20005, 'Doctor is fully booked for this date');
117   WHEN OTHERS THEN
118     ROLLBACK;
119     RAISE;
120 END create_appointment;

```

תוצאות הרצה:

הרצה נכונה-

	Variable	Type	Value
✓	Enter_Doctor_ID	String	38187
✓	Enter_Patient_ID	String	971
▶ ✓	Enter_Appointment_Date	String	2025.12.03
✓	Enter_Treatment_ID	String	69548
*			

תוצאתה:

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Appointment created: ID=99863, Doctor=Raymond Warner, Patient=Portia Foster, Date=03.12.2025, Treatment=Aesthetic

```

שגיאות-

ת.ז. רופא לא תקין

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Error: ORA-20001: Invalid doctor ID

```

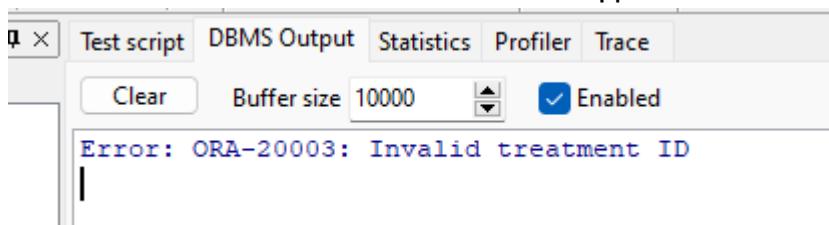
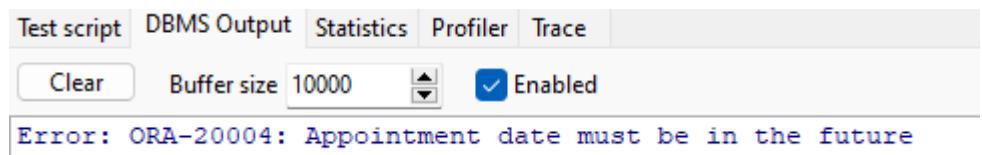
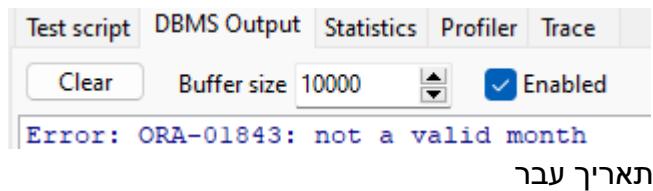
ת.ז. מטופל לא תקין

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Error: ORA-20002: Invalid patient ID

```

תאריך לא תקין



פונקציה 1

נרצה להעניק פרס לעריץ הגדל של המרפאה!
עריץ הוא רופא או מטופל שהוici הרבה פעמים במרפאה,
אם הוא רופא הוא יזכה בתור חופשי. תור עתידי כלשהו שלו יועבר לרופא אחר בעל אותה
המוחיות כמו והוא יעניק לו הפסקה
אם הוא מטופל הוא יזכה בטיפול אקדמי חינם על חשבון המרפאה
במקרה של שוויון יוכרז שוויון אך איש לא יקבל פרס

הפונקציה-

```
CREATE OR REPLACE FUNCTION secondMostVisitsPerson(fromDate IN appointment.aDate%TYPE, toDate IN appointment.aDate%TYPE)
RETURN VARCHAR
IS
    fanPatientId Patient.cId%TYPE;
    PatientNumApps INTEGER := 0;
    fanDoctorID Doctor.sId%TYPE;
    doctorNumApps INTEGER := 0;
    almostFanId INTEGER := 0;
    almostFanType VARCHAR2(20);
    almostFanName VARCHAR2(100);
    almostFanNumArrivals INTEGER := 0;
    FunctionResult VARCHAR2(200);
    v_new_sId Doctor.sId%TYPE;
    v_rows_updated INTEGER;
    v_today DATE := TRUNC(SYSDATE);

    -- Define a record type to hold patient prize information
    TYPE prize_rec IS RECORD (
        patient_id Patient.cId%TYPE,
        treatment_id Treatment.tId%TYPE,
        treatment_name Treatment.tType%TYPE
    );
    -- Create a variable of the record type
    v_prize prize_rec;

    -- Define a cursor to select a free treatment from the Treatment table
    CURSOR free_treatment_cursor IS
        SELECT tid, ttype FROM Treatment WHERE Price > 0
        ORDER BY DBMS_RANDOM.VALUE;

    -- Define a variable to hold treatment details
    v_treatment_id Treatment.tId%TYPE;
    v_treatment_name Treatment.tType%TYPE;
BEGIN
    -- Find the second most frequent patient
    BEGIN
        SELECT cId, num_apps INTO fanPatientId, PatientNumApps
        FROM (
            SELECT cId, COUNT(*) AS num_apps,
                ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS rn
            FROM appointment
            WHERE aDate BETWEEN fromDate AND toDate
            GROUP BY cId
        )
    
```

```

        WHERE rn = 1;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('No second most fan patient found.');
            RETURN 'No second most fan patient found.';
    END;

    -- Find the second most frequent doctor
BEGIN
    SELECT sId, num_apps INTO fanDoctorId, doctorNumApps
    FROM (
        SELECT sId, COUNT(*) AS num_apps,
               ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS rn
        FROM appointment
        WHERE aDate BETWEEN fromDate AND toDate
        GROUP BY sId
    )
    WHERE rn = 1;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No second most fan doctor found.');
        RETURN 'No second most fan doctor found.';
    END;

DBMS_OUTPUT.PUT_LINE('Almost most fan Patient: ' || fanPatientId || ' visited here: ' || PatientNumApps || ' times!');
DBMS_OUTPUT.PUT_LINE('Almost most fan doctor: ' || fanDoctorId || ' served here: ' || doctorNumApps || ' times!');

IF PatientNumApps > doctorNumApps THEN
    almostFanId := fanPatientId;
    almostFanType := 'Patient';
    almostFanNumArrivals := PatientNumApps;
    SELECT cName INTO almostFanName FROM Patient WHERE cId = almostFanId;
    -- Assign a free treatment to the patient
    OPEN free_treatment_cursor;
    --FETCH free_treatment_cursor INTO v_treatment_id, v_treatment_name;

    --Make a loop instead
    LOOP
        FETCH free_treatment_cursor INTO v_treatment_id, v_treatment_name;
        -- Exit the loop if no more rows to fetch
        EXIT WHEN free_treatment_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Fetched Treatment ID: ' || v_treatment_id || ', Name: ' || v_treatment_name);
    EXIT; -- Exit after fetching the first treatment
    END LOOP;
    CLOSE free_treatment_cursor;
    IF v_treatment_id IS NOT NULL THEN
        -- Create a prize record with the patient ID and treatment details
        v_prize.patient_id := almostFanId;
        v_prize.treatment_id := v_treatment_id;
        v_prize.treatment_name := v_treatment_name;

        -- Print or store the prize information
        DBMS_OUTPUT.PUT_LINE('Congratulations to Patient ' || v_prize.patient_id || '! You have won a free treatment: ' || v_prize.treatment_name || '.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('No available treatments for the prize.');
    END IF;

ELSIF doctorNumApps > PatientNumApps THEN
    almostFanId := fanDoctorId;
    almostFanType := 'Doctor';
    almostFanNumArrivals := doctorNumApps;
    SELECT sName INTO almostFanName FROM Staff WHERE sId = almostFanId;

    -- Find a suitable replacement doctor and update appointment
    BEGIN
        -- Check if there are any future appointments
        DECLARE
            v_future_appointments INTEGER;
        BEGIN
            SELECT COUNT(*) INTO v_future_appointments
            FROM system.appointment
            WHERE sId = fanDoctorId AND aDate > SYSDATE;

            IF v_future_appointments > 0 THEN
                -- Find a suitable replacement doctor
                BEGIN
                    SELECT d.sId INTO v_new_sId
                    FROM (
                        SELECT d.sId
                        FROM system.doctor d
                        WHERE d.Specialties =
                            (SELECT d2.Specialties
                            FROM system.doctor d2
                            WHERE d2.sId = fanDoctorId
                            )
                        AND d.sId <> fanDoctorId
                        ORDER BY dbms_random.value
                    ) d
                END;
            END IF;
        END;
    END;
END IF;

```

```

secondMostVisitsPerson
  131      )
  132      AND d.id <> fanDoctorId
  133      ORDER BY dbms_random.value
  134    ) d
  135    WHERE ROWNUM = 1;
  136
  137    -- Update one future appointment for the second most popular doctor
  138    UPDATE system.appointment
  139    SET sid = v_new_sid
  140    WHERE sid = fanDoctorId
  141    AND adate > v_today
  142    AND ROWNUM = 1;
  143
  144    v_rows_updated := SQL%ROWCOUNT;
  145
  146    IF v_rows_updated > 0 THEN
  147      DBMS_OUTPUT.PUT_LINE('Appointment updated for doctor: ' || fanDoctorId);
  148    ELSE
  149      DBMS_OUTPUT.PUT_LINE('Failed to update appointment for doctor: ' || fanDoctorId);
  150    END IF;
  151    Commit;
  152  EXCEPTION
  153    WHEN OTHERS THEN
  154      DBMS_OUTPUT.PUT_LINE('Error finding or updating appointment: ' || SQLERRM);
  155  END;
  156
  157  ELSE
  158    DBMS_OUTPUT.PUT_LINE('No future appointments found for doctor: ' || fanDoctorId);
  159  END IF;
  160
  161  END;
  162  EXCEPTION
  163    WHEN OTHERS THEN
  164      DBMS_OUTPUT.PUT_LINE('Error in nested block: ' || SQLERRM);
  165  END;
  166
  167  ELSIF PatientNumApps = doctorNumApps THEN
  168    almostFanType := 'Tie';
  169    almostFanName := 'Both Patient and Doctor have the same number of appointments!';
  170    almostFanArrivals := PatientNumApps; -- or doctorNumApps, they're equal in this case
  171  END IF;
  172
  173  FunctionResult := 'To sum up, the almost most fan of us is the ' || almostFanType || ':' || almostFanName ||
  174  ' who was here ' || almostFanArrivals || ' times!!!';
  175  DBMS_OUTPUT.PUT_LINE(FunctionResult);
  176
  177  RETURN FunctionResult;
END secondMostVisitsPerson;

```

הרצה-

	Variable	Type	Value
▶	fromDate	Date	01/01/2000
▶	toDate	Date	01/01/2030
▶	result	String	To sum up, the almost most fan of us is the Doctor: Jada Hatfield who was here 6 times!!!
*			

תוצאות-

תיק-

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Almost most fan Patient: 572 visited here: 6 times!
Almost most fan doctor: 21570 served here: 6 times!
To sum up, the almost most fan of us is the Tie: Both Patient and Doctor have the same number of appointments! who was here 6 times!!!

```

מתופל ניצח-

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Almost most fan Patient: 20006 visited here: 9 times!
Almost most fan doctor: 39663 served here: 8 times!
Fetched Treatment ID: 36449, Name: Aesthetic
Congratulations to Patient 20006! You have won a free treatment: Aesthetic.
To sum up, the almost most fan of us is the Patient: Fiona Anderson who was here 9 times!!!

```

רופא ניצח-

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Almost most fan Patient: 20006 visited here: 9 times!
Almost most fan doctor: 39663 served here: 10 times!
Appointment updated for doctor: 39663
To sum up, the almost most fan of us is the Doctor: Hanna Whitfield who was here 10 times!!!

```

הפגישות שלה לפני שינוי-

SQL Output Statistics

```
select * from appointment
where sId=39663
```

A screenshot of a database management system interface showing the results of a query. The results are displayed in a table with the following columns: ADATE, APPOINTMENTID, SID, and CID. The data consists of 10 rows, each containing a unique ID (1-10), an appointment date, and two identifiers (SID and CID). The dates range from 30/06/2027 to 04/08/2026.

	ADATE	APPOINTMENTID	SID	CID
► 1	30/06/2027	84988	39663	119
2	29/08/2028	91912	39663	660
3	13/06/2028	49699	39663	327
4	27/07/2024	59693	39663	130
5	31/07/2027	59991	39663	119
6	07/06/2029	69782	39663	221
7	28/07/2025	68242	39663	856
8	23/06/2027	31819	39663	20008
9	01/01/2027	96736	39663	994
10	04/08/2026	57292	39663	559

אחרי שינוי-

SQL Output Statistics

```
select * from appointment
where sId=39663
```

A screenshot of a database management system interface showing the results of a query. The results are displayed in a table with the following columns: ADATE, APPOINTMENTID, SID, and CID. The data consists of 9 rows, each containing a unique ID (1-9), an appointment date, and two identifiers (SID and CID). The dates range from 29/08/2028 to 04/08/2026.

	ADATE	APPOINTMENTID	SID	CID
► 1	29/08/2028	91912	39663	660
2	13/06/2028	49699	39663	327
3	27/07/2024	59693	39663	130
4	31/07/2027	59991	39663	119
5	07/06/2029	69782	39663	221
6	28/07/2025	68242	39663	856
7	23/06/2027	31819	39663	20008
8	01/01/2027	96736	39663	994
9	04/08/2026	57292	39663	559

תוכנית ראשית 1

```

DECLARE
    -- Variables for secondMostVisitsPerson
    v_result VARCHAR2(200);
    v_from_date DATE;
    v_to_date DATE;

    -- Variables for create_appointment
    l_appointment_cursor SYS_REFCURSOR;
    l_appointment_id NUMBER;
    l_doctor_name VARCHAR2(100);
    l_patient_name VARCHAR2(100);
    l_appointment_date DATE;
    l_treatment_type VARCHAR2(15);

    l_doctor_id Doctor.sId%TYPE;
    l_patient_id Patient.cId%TYPE;
    l_treatment_id Treatment.tId%TYPE;
    l_input_date VARCHAR2(30);

    -- Helper function for date conversion
    FUNCTION convert_date(p_date IN VARCHAR2) RETURN DATE IS
    BEGIN
        RETURN TO_DATE(p_date, 'DD.MM.YYYY');
    EXCEPTION
        WHEN OTHERS THEN
            RETURN TO_DATE(p_date, 'YYYY-MM-DD');
    END;

BEGIN
    DBMS_OUTPUT.PUT_LINE('Testing secondMostVisitsPerson function:');
    DBMS_OUTPUT.PUT_LINE('-----');

    -- Test secondMostVisitsPerson
    BEGIN
        v_result := secondMostVisitsPerson(fromDate=> :fromDate, toDate=>:toDate);
        DBMS_OUTPUT.PUT_LINE('Result: ' || v_result);
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error in secondMostVisitsPerson: ' || SQLERRM);
    END;

    DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Testing create_appointment function:');
    DBMS_OUTPUT.PUT_LINE('-----');

    -- Test create_appointment
    BEGIN
        l_doctor_id := :Enter_Doctor_ID;
        l_patient_id := :Enter_Patient_ID;
        l_treatment_id := :Enter_Treatment_ID;
        l_input_date := :Enter_Appointment_Date;

        l_appointment_date := convert_date(l_input_date);

        create_appointment(
            p_doctor_id => l_doctor_id,
            p_patient_id => l_patient_id,
            p_appointment_date => l_appointment_date,
            p_treatment_id => l_treatment_id,
            p_appointment_cursor => l_appointment_cursor
        );

        FETCH l_appointment_cursor INTO l_appointment_id, l_doctor_name, l_patient_name, l_appointment_date, l_treatment_type;
        CLOSE l_appointment_cursor;

        DBMS_OUTPUT.PUT_LINE('Appointment created: ID=' || l_appointment_id ||
                            ', Doctor=' || l_doctor_name ||
                            ', Patient=' || l_patient_name ||
                            ', Date=' || TO_CHAR(l_appointment_date, 'DD.MM.YYYY') ||
                            ', Treatment=' || l_treatment_type);
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
    END;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('An unexpected error occurred in the main block: ' || SQLERRM);
    END;

```

Variable	Type	Value
Enter_Doctor_ID	String	38187
Enter_Patient_ID	String	971
Enter_Appointment_Date	String	18.02.2025
Enter_Treatment_ID	String	69548
fromDate	Date	01/01/2000
toDate	Date	01/01/2030

הרצה משותפת -

```
Test script DBMS Output Statistics Profiler Trace
DECLARE
    -- Variables for secondMostVisitsPerson
    v_result VARCHAR2(200);
    v_from_date DATE;
    v_to_date DATE;
    ...
    -- Variables for create_appointment
    l_appointment_cursor SYS_REFCURSOR;
    l_appointment_id NUMBER;
    l_doctor_name VARCHAR2(100);
    l_patient_name VARCHAR2(100);
    l_appointment_date DATE;
    l_treatment_type VARCHAR2(15);
    ...
    l_doctor_id Doctor.sId%TYPE;
    l_patient_id Patient.cId%TYPE;
    l_treatment_id Treatment.tId%TYPE;
    l_input_date VARCHAR2(30);
    ...
    -- Helper function for date conversion
    FUNCTION convert_date(p_date IN VARCHAR2) RETURN DATE IS
    BEGIN
        RETURN TO_DATE(p_date, 'DD.MM.YYYY');
    EXCEPTION
        WHEN OTHERS THEN
            RETURN TO_DATE(p_date, 'YYYY-MM-DD');
    END;
    ...
BEGIN
    DBMS_OUTPUT.PUT_LINE('Testing secondMostVisitsPerson function:');
    DBMS_OUTPUT.PUT_LINE('-----');
    ...
    -- Test secondMostVisitsPerson
    BEGIN
        v_result := secondMostVisitsPerson(fromDate=> :fromDate, toDate=>:toDate);
        DBMS_OUTPUT.PUT_LINE('Result: ' || v_result);
    EXCEPTION
    ...
END;

```

Variable	Type	Value
Enter_Doctor_ID	String	38187
Enter_Patient_ID	String	971
Enter_Appointment_Date	String	18.02.2025
Enter_Treatment_ID	String	69548
fromDate	Date	01/01/2000
toDate	Date	01/01/2030

63:32 SYSTEM@XE Executed in 0.875 seconds

פלט משותף -

```
Testing secondMostVisitsPerson function:
-----
Almost most fan Patient: 971 visited here: 10 times!
Almost most fan doctor: 38187 served here: 10 times!
To sum up, the almost most fan of us is the Tie: Both Patient and Doctor have the same number of appointments! who was here 10 times!!!
Result: To sum up, the almost most fan of us is the Tie: Both Patient and Doctor have the same number of appointments! who was here 10 times!!!

Testing create_appointment function:
-----
Appointment created: ID=99870, Doctor=Raymond Warner, Patient=Portia Foster, Date=18.02.2025, Treatment=Aesthetic
```

פרויקט זהה 2

הסבר:

בעקבות המצב במשק הישראלי הוחלט להעלות את מחיר הטיפול במרפאה- לפי הטיפול הכי פופולארי ,

הקוד:

```
CREATE OR REPLACE PROCEDURE PRICE
```

```
IS
```

```
Cursor to fetch treatment details --
```

```
CURSOR treatment_cur IS
```

```
SELECT t.tId, t.tType, t.Price, COUNT(tpa.AppointmentID) as usage_count
```

```
FROM Treatment t
```

```
LEFT JOIN TPreformedInA tpa ON t.tId = tpa.tId
```

```
GROUP BY t.tId, t.tType, t.Price
```

```
;ORDER BY usage_count DESC
```

```
Record to store treatment details --
```

```
:treatment_rec treatment_cur%ROWTYPE
```

```
Variables --
```

```
:v_price_increase NUMBER
```

```
:v_update_count NUMBER := 0
```

```
:v_old_price NUMBER
```

```
:v_new_price NUMBER
```

```
BEGIN
```

```
Open the cursor --
```

```
;OPEN treatment_cur
```

Loop through treatments --

LOOP

;FETCH treatment_cur INTO treatment_rec

;EXIT WHEN treatment_cur%NOTFOUND

Calculate price increase based on usage (example logic) --

IF treatment_rec.usage_count > 1 THEN

v_price_increase := 0.15; -- 15% increase for very popular treatments

ELSIF treatment_rec.usage_count > 5 THEN

v_price_increase := 0.10; -- 10% increase for moderately popular treatments

ELSIF treatment_rec.usage_count > 2 THEN

v_price_increase := 0.05; -- 5% increase for somewhat popular treatments

ELSE

v_price_increase := 0; -- No increase for unpopular treatments

;END IF

Update price if there's an increase --

IF v_price_increase > 0 THEN

;v_old_price := treatment_rec.Price

;v_new_price := v_old_price * (1 + v_price_increase)

UPDATE Treatment

SET Price = v_new_price

;WHERE tId = treatment_rec.tId

;v_update_count := v_update_count + 1

Detailed output for each price update --

```
;DBMS_OUTPUT.PUT_LINE('Treatment updated:')

;DBMS_OUTPUT.PUT_LINE(' ID: ' || treatment_rec.tId)

;DBMS_OUTPUT.PUT_LINE(' Type: ' || treatment_rec.tType)

DBMS_OUTPUT.PUT_LINE(' Old price: ' || TO_CHAR(v_old_price,
;'999999.99'))

DBMS_OUTPUT.PUT_LINE(' New price: ' || TO_CHAR(v_new_price,
;'999999.99'))

DBMS_OUTPUT.PUT_LINE(' Price increase: ' || TO_CHAR((v_new_price -
;v_old_price), '999999.99'))

DBMS_OUTPUT.PUT_LINE(' Percentage increase: ' ||
;TO_CHAR(v_price_increase * 100, '999.99') || '%')

DBMS_OUTPUT.PUT_LINE(' Usage count: ' ||
;treatment_rec.usage_count)

;('-----')DBMS_OUTPUT.PUT_LINE

;END IF

;END LOOP
```

Close the cursor --

```
;CLOSE treatment_cur
```

Final output --

```
DBMS_OUTPUT.PUT_LINE('Price update complete. ' || v_update_count || '
;treatments updated.')
```

EXCEPTION

WHEN OTHERS THEN

Error handling --

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

```

Close cursor if it's still open --
IF treatment_cur%ISOPEN THEN
;CLOSE treatment_cur
;END IF

Rollback any changes --
;ROLLBACK
;END

```

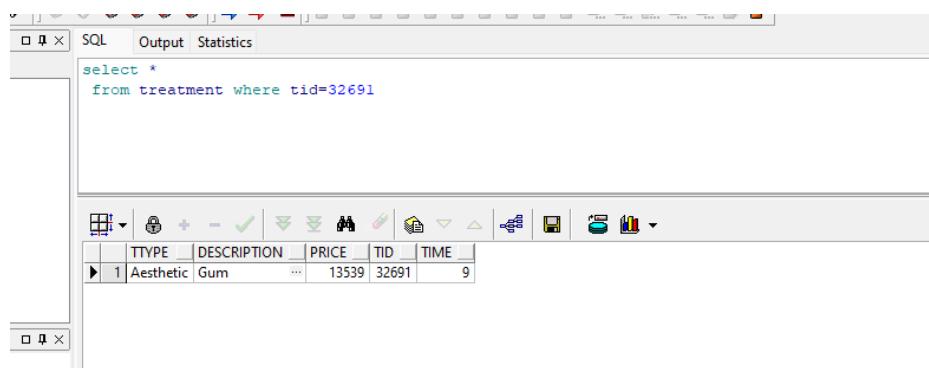
(הסביר קוד: אני אשלח לפרוצדורה 2 משתנים:
 אחד הוא כמות מינימאלית של טיפולים שנחשבים לפופולריים
 שני הוא בכמה אחוזים אני רוצה להעלות
 צילום ש�示ה שמהפונקציה עובדת :

	TTYPE	DESCRIPTION	PRICE	TID	TIME
▶	1	Aesthetic	Gum	1177	32691

צילום מחיר טיפול לפני פרוצדורה:

	TTYPE	DESCRIPTION	PRICE	TID	TIME
▶	1	Aesthetic	Gum	1177	32691

צילום מחיר מעודכן אחרי הרצת הפרוצדורה:



A screenshot of a SQL query window. The title bar says "SQL Output Statistics". The query in the SQL pane is:

```
select *  
from treatment where tid=32691
```

The results pane shows a single row of data in a table format:

	TTYPE	DESCRIPTION	PRICE	TID	TIME
▶	1	Aesthetic Gum	...	13539	32691

פונקציה 2

הסבר:

אנו רוצים לדעת מי הרופא הכי טוב במרפאה בשביל לשלוח אותו לחופשה בקריטיטים, הפרמטרים שקבענו לרופא הכי טוב מורכב ממספר הפגישות הרבה יותר שיש לרופא ומהיר ככל של הטיפולים שהוא טיפול

קוד:

```
CREATE OR REPLACE FUNCTION Best_Doc RETURN VARCHAR2 IS  
;(100)Result VARCHAR2
```

Explicit cursor for doctors --

```
CURSOR doctor_cur IS  
SELECT s.sId, s.sName, d.Specialties, d.License  
FROM Staff s  
;JOIN Doctor d ON s.sId = d.sId
```

Variables --

```
;(30)v_best_doctor_name VARCHAR2  
;v_max_performance_score NUMBER := 0
```

Record type for doctor details --

```
) TYPE doctor_rec_type IS RECORD  
,id NUMBER  
,(30)name VARCHAR2  
,(15)specialty VARCHAR2  
,(15)license VARCHAR2  
,appointment_count NUMBER  
,total_price NUMBER  
performance_score NUMBER
```

```
;(  
;v_doctor_rec doctor_rec_type  
  
BEGIN  
Loop through doctors using explicit cursor --  
FOR doctor_row IN doctor_cur LOOP  
Use implicit cursor to get appointment details --  
SELECT COUNT(a.AppointmentID), NVL(SUM(p.TotalPrice), 0)  
INTO v_doctor_rec.appointment_count, v_doctor_rec.total_price  
FROM Appointment a  
LEFT JOIN Payment p ON a.AppointmentID = p.AppointmentID  
;WHERE a.sId = doctor_row.sId  
  
;v_doctor_rec.id := doctor_row.sId  
;v_doctor_rec.name := doctor_row.sName  
;v_doctor_rec.specialty := doctor_row.Specialties  
;v_doctor_rec.license := doctor_row.License  
  
Calculate performance score (you can adjust this formula as needed) --  
v_doctor_rec.performance_score := v_doctor_rec.appointment_count *  
;v_doctor_rec.total_price / 1000  
  
Check if this doctor has the best performance so far --  
IF v_doctor_rec.performance_score > v_max_performance_score THEN  
;v_max_performance_score := v_doctor_rec.performance_score  
;v_best_doctor_name := v_doctor_rec.name  
;END IF  
;END LOOP
```

```

Check if a best doctor was found --
IF v_best_doctor_name IS NOT NULL THEN
;Result := 'The best doctor is ' || v_best_doctor_name
ELSE
;'.Result := 'No doctors found or no appointments scheduled
;END IF

;RETURN Result

EXCEPTION
WHEN OTHERS THEN
;Result := 'An error occurred: ' || SQLERRM
;RETURN Result
;END Best_Doc

```

צילום מסך של שאלתה שמביאה בדיק לפי אותה פרמטרים את הרופא הכי טוב במרפאה:

```

SELECT doctor_name, performance_score
FROM (
  SELECT
    s.name AS doctor_name,
    (COUNT(a.AppointmentID) * NVL(SUM(p.TotalPrice), 0) / 1000) AS performance_score
  FROM
    Staff s
  JOIN
    Doctor d ON s.sId = d.sId
  LEFT JOIN
    Appointment a ON s.sId = a.sId
  LEFT JOIN
    Payment p ON a.AppointmentID = p.AppointmentID
  GROUP BY
    s.sId, s.sName
  ORDER BY
    performance_score DESC
)
WHERE ROWNUM = 1;

```

DOCTOR_NAME	PERFORMANCE_SCORE
Weren Beddy	0

צילום מסך שמרתה את התוצאה בפונקציה שלי:

תוכנית ראשית 2

main

```
CREATE OR REPLACE PROCEDURE main_class_management(
    p_user_threshold IN NUMBER DEFAULT NULL
) AS
    TYPE class_id_array IS TABLE OF NUMBER;
    v_class_ids class_id_array;
    v_popularity_threshold NUMBER;
    v_evaluation_result class_popularity_result;
    v_total_popularity NUMBER := 0;
    v_class_count NUMBER := 0;
    v_result_cursor SYS_REFCURSOR;
BEGIN
    -- Step 1: Fetch all class IDs
    SELECT class_number BULK COLLECT INTO v_class_ids
        FROM class_lesson
        ORDER BY class_number;

    -- Step 2: Evaluate all classes and calculate average popularity
    DBMS_OUTPUT.PUT_LINE('Evaluating all classes.');
    DBMS_OUTPUT.PUT_LINE('-----');
    FOR i IN 1..v_class_ids.COUNT LOOP
        v_evaluation_result := evaluate_class_popularity(v_class_ids(i),
            v_result_cursor);
        v_total_popularity := v_total_popularity +
            v_evaluation_result.popularity_rate;
        v_class_count := v_class_count + 1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Average popularity: ' || TO_CHAR(v_total_popularity / v_class_count));
END;
```

```

        END LOOP;

-- Calculate threshold

IF p_user_threshold IS NOT NULL THEN

    v_popularity_threshold := p_user_threshold;

ELSE

    -- Set threshold to average popularity rate + 10%

    v_popularity_threshold := (v_total_popularity / v_class_count) + 10;

END IF;

DBMS_OUTPUT.PUT_LINE('Popularity threshold set to: ' ||
                     v_popularity_threshold || '%');

DBMS_OUTPUT.PUT_LINE('-----');

-- Step 3: Display popular classes

DBMS_OUTPUT.PUT_LINE('Popular classes:');
DBMS_OUTPUT.PUT_LINE(' ');

FOR i IN 1..v_class_ids.COUNT LOOP

    v_evaluation_result := evaluate_class_popularity(v_class_ids(i),
                                                       v_result_cursor);

    IF v_evaluation_result.popularity_rate >= v_popularity_threshold THEN

        DBMS_OUTPUT.PUT_LINE('Class Number: ' ||
                             v_evaluation_result.class_number ||

                             ', Popularity Rate: ' || v_evaluation_result.popularity_rate ||

                             '%, Recommendation: ' || v_evaluation_result.recommendation);

    END IF;
END LOOP;

-- Step 4: Update popular classes

DBMS_OUTPUT.PUT_LINE('-----');

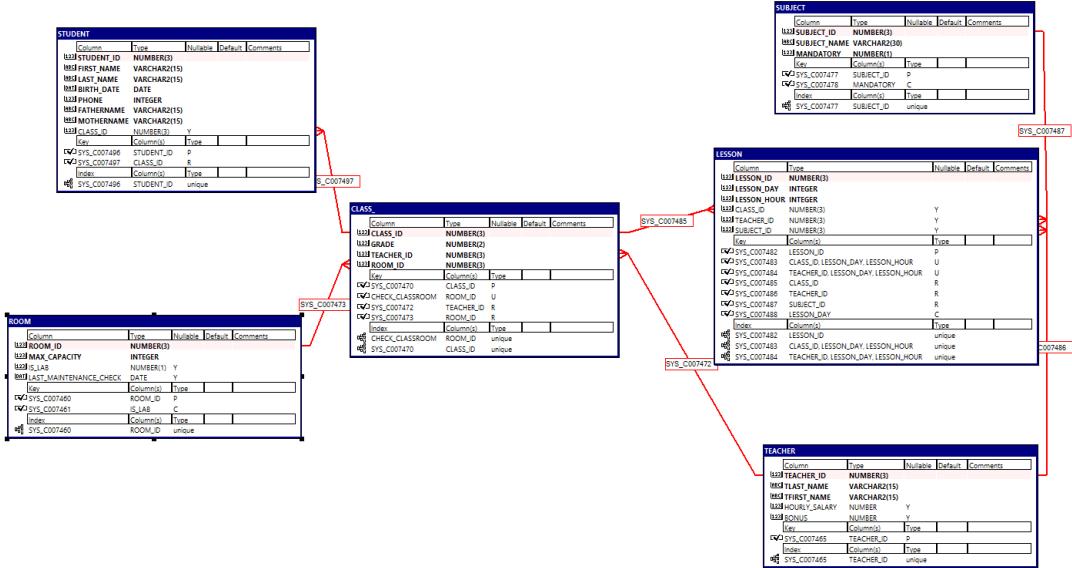
```

```
        DBMS_OUTPUT.PUT_LINE('Updating popular classes:');
                DBMS_OUTPUT.PUT_LINE(' ');
update_classes_by_popularity(v_popularity_threshold);
        DBMS_OUTPUT.PUT_LINE('Class updates completed.');

        EXCEPTION
        WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('An error occurred in main_class_management: ' ||
SQLERRM);
        RAISE;
END main_class_management;
```

שלב 4 אגף חדש

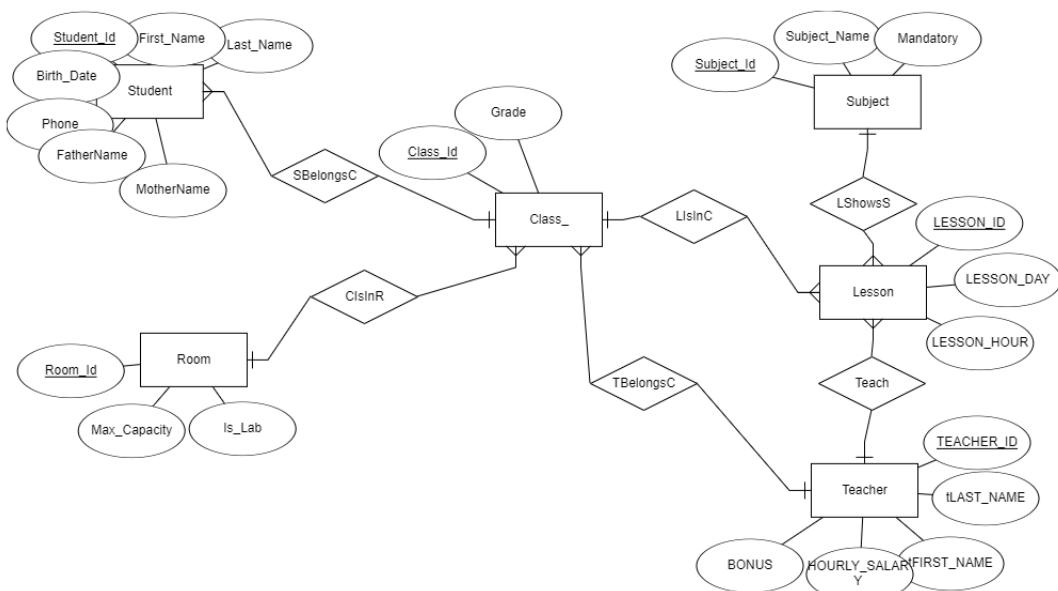
הוספנו את הגיבוי של האגף החדש אל הפרויקט וקיבלנו את ה DSD הבא



ארכ אוף חדש ERD

כעת נרצה לנתח ממנו ERD
 יצרנו יישיות חדשות עבור כל טבלה על פי השמות כולל התכונות של כל ישות והמפתח שלה.
 כעת נדרשו לנתח את הקשרים ביניהן, נעזרנו במקורות הזרים שמופיעים בטבלאות.
 לסטודנט למשל מופיע מפתח זר של כיתה, זהו קשר של רבים לאחד, הרבה סטודנטים
 שייכים לכיתה אך כל סטודנט שיר לכיתה אחת בלבד
 לכיתה יש שני מפתחות זרים- מורה וחדר, שוב שני קשרים של רבים לאחד. לכל כיתה יש
 מורה אחד אבל מורה מלמד כמה כיתות. וכן לכל כיתה יש חדר אחד וכו'
 נתנו שמות לקשרים ביניהם ובשביל בדיקה המרנו לדס DSDM צירה דרך האתר ERDPLUS כדי
 לבדוק את התוצאה

-ERDה



ERD מאוחז

פירוט החלטות-

בפרויקט שלנו יש ישות של מטופל (פטציינט) ובางף החדש באים לידי בית ספר לטיפול במרפאה. ליד יש תכונות יהודיות שאין למטופל והן למשל שמות ההורים וכן החלטנו להוסיף ישות חדשה בשם סטודנט שתירש ממטופל. יהיה לה את כל התכונות שלו (ע"י ירשה) ובנוסף את התכונות- שם אב ושם אם וכן מפתח זר של שייכות לכיתה

שינויים נוספים בעקבות הפעולה הזאת-

لتלמיד יש תוכנה של שם פרטי ואחרת של שם משפחה. נאחד אותן לתוכנית הקיימת כבר אצלנו- שם.

لتלמיד יש תאריך לידה ולמטופל יש רק שנת לידה- כדי למנוע את הצורך להכניס לכל המטופלים (ויש במאות, מרפאה מצילה סך הכל) תאריך מלא החלטנו לקחת מהרשומות החדשות של האגף החדש רק את השנה ובה להשתמש. לעומת לכל סטודנט נשולף את השנה מתוך תאריך הלידה ואוthon נשומר לתוך השדה הקיים שלנו.

בפרויקט שלנו יש גם ישות של צוות, נרצה להוסיף לו את הישות של מורה שהיא מהאגף החדש.

נוסיף ישות חדשה בשם מורה שתירש מהישות צוות ובה נשים את התכונות הייחודיות למורה- בונוס, שכר שעתי'

גם כן נאחד כאן את התכונות שם ושם משפחה אל התוכנה הקיימת שלנו של שם

בנוסף נרצה להוסיף קשרים בין ילד בבית הספר לבין תשלום, הקשר יוצר מכון שהילדים משלמים על הלימודים, נקרא PayForStudies, קשר של יחיד לרבים. בכל תשלום יש ילד אחד אבל כל ילד משלם הרבה תשלום

בנוסף הוספנו מתוך פרויקט בית הספר את הישויות והתכונות:

ישות כיתה Class_Id, Grade _Class עם התכונות-

ישות נושא Subject_Id, Subject_Name, Mandatory Subject עם התכונות-

ישות שיעור Lesson_ID Lesson LESSON_HOUR, LESSON_DAY, LESSON_Room LESSON עם התכונות-

ישות חדר Room Room_Id Room_Max_Capacity, Is_Lab Room_Is_Lab עם התכונות-

העברה הנתונים

העברה נתוני הסטודנט אל המטופל:

הסטודנט ירש מהמטופל ולכן צריך להעביר את הנתונים שלו אל המטופל ולהשאיר תכונות ייחודיות כפי שציינו אצלן.

היתה בעיה בהעברה של נתונים הטלפון של הסטודנטים כי הם היו מסוג INTEGER ושלנו מסוג varchar ולכן ביצענו את המרה הבאה-
הוסףנו עמודה חדשה חדשה אליה נעביר את הטלפונים לאחר שינוי הסוג

כך-

General							Columns	Keys	Checks	Indexes	Privileges	Triggers
Type owner	Name											
Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments						
CBIRTHYEAR		NUMBER(4)				...						
CADDRESS		VARCHAR2(55)				...						
CNAME		VARCHAR2(30)				...						
CGENDER		VARCHAR2(5)				...						
CID		NUMBER(5)				...						
CMOBILE		VARCHAR2(10)				...						
CMAIL		VARCHAR2(65)				...						
newPhone		INTEGER				...						
*						...						

אוחץ הרצינו את השורות הבאות-

```
SQL Output Statistics
update patient set newPhone = to_char(cMobile);
alter table patient drop column cMobile;
alter table patient rename column newPhone to cMobile;
select * from patient|
```

	CBIRTHYEAR	CADDRESS	CNAME	CGENDER	CID	CMAIL	CMOBILE
1	1972	567-8655 Integer Ave	Reagan Berry	M	727	pede.nec@outlook.edu	513417574
2	2003	7681 Loren, Avenue	Bradley Baxter	F	764	enim.nunc.ut@sol.net	515263776
3	1957	126-8581 Donec Rd.	Rajah Wilder	F	934	posuere.at@yahoo.net	555376138
4	1954	Ap #936-210 Erat Avenue	Dana Wyatt	F	354	dis.parturient.montes@aol.ca	541603349
5	1938	228-2863 Curabitur Rd.	Megan Molina	F	717	mauris.vel@outlook.net	572442876
6	1952	Ap #177-2381 Imperdiet St.	Rooney Jennings	F	350	eu.erat@protonmail.ca	556334347
7	1975	Ap #625-170 Vivamus Ave	Thomas Macias	M	148	gravida.aliquam@protonmail.net	565667249
8	1970	481-6951 Nascentur Ave	Adam Bond	M	600	donec@outlook.co.uk	533935363
9	1958	349-3264 Pede, Avenue	Gloria McKee	F	910	massa.mauris.vestibulum@yahoo.co.uk	594712703
10	1998	P.O. Box 567, 6464 Ligula Rd.	Basil Russo	F	989	vitae.risus@google.edu	526357271
11	2023	Ap #446-6388 Etiam St.	Shay Dillon	F	856	accumulo@cloud.com	524682525
12	1970	Ap #261-7637 Magna Av.	Quamar Welch	M	392	vestibulum.ut@yahoo.org	531137654
13	1990	Ap #774-3051 Erat Street	Cody Salinas	M	163	amet.lorem@yahoo.net	522779526
14	2000	505-6353 A Av.	Brittany Guerra	M	219	magna.tellus@cloud.org	572180937
15	2000	613-6144 Sollicitudin Avenue	Murphy Rodriguez	F	599	vulputate.dui@sol.ca	543917465
16	1987	P.O. Box 920, 5357 Vel Av.	Kyle Hood	M	290	magnis.dis@protonmail.ca	543506775
17	1950	Ap #496-1940 Nunc Rd.	Timon Dickson	M	610	donec.luctus@protonmail.org	561604265
18	1988	526-8631 Tellus, Rd.	Galvin Mueller	M	308	quam quis@protonmail.ca	57192579
19	1982	P.O. Box 189, 7664 Eget St.	Maxime Schmidt	M	433	nec.tempus@outlook.edu	565531776
20	2012	P.O. Box 754, 6798 Orci Avenue	Leo Jordan	M	889	et.eros.prin@google.co.uk	585112072
21	2011	P.O. Box 903, 1703 Lorem Av.	Melissa Bennett	F	408	et.magnis.dis@google.org	514454657
22	1974	967-451 Sed, Street	Madison Madden	F	676	lorem.u@aol.ca	571078114
23	1951	826 Euismod Rd.	Madison Madden	M	468	consectetur.mauris@outlook.co.uk	546537375
24	1951	7838 Nec, St.	Giselle Jimenez	M	736	metus@protonmail.com	524684538
25	1935	Ap #189-8861 Eget Ave	Charlotte Nguyen	M	538	mi.enim@google.edu	572109396
26	2024	Ap #292-1074 Tristique Avenue	Cecilia Butler	F	332	erat.voluptat@protonmail.org	577455669
27	2021	244-6311 Semper St.	Shad Vance	M	832	semper.cursum.integer@yahoo.net	51884874
28	1919	P.O. Box 512, 3296 Risus Av.	Xavier Bowen	F	662	ullamcorper.nisl@aol.net	503221446
29	1998	Ap #923-8893 Non, St.	Alma Herrera	M	657	laoreet.lectus@yahoo.edu	515626643
30	1938	4526 Elit, Road	Fredericka Alexander	M	777	hendrerit.negue@protonmail.net	514117865
31	2014	873-1295 Et Street	Rinah Conley	M	634	eros.nam@icloud.edu	528954826
32	2001	4166 Selerisque Avenue	Lamar Conrad	F	672	fringilla@yahoo.org	572746354
33	2015	Ap #892-6022 Nisi St.	May Mcleod	F	420	eu.odio@aol.edu	593277604
34	2010	P.O. Box 597, 2340 Malesuada St.	Josiah Barker	F	758	lorem.ipsum@aol.ca	512575114
35	1920	Ap #174-6408 Tortor, Ave	Irene Rodriguez	M	328	nunc.in.att@hotmail.org	564760474
36	1965	Ap #862-6080 Dictum Road	Kerry Winters	M	180	lorem.ipsum@icloud.ca	592340126

והשינויים נראים בטבלה עצמה כך-

Type owner	Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments
►	CBIRTHYEAR	<input type="checkbox"/>	NUMBER(4)	<input type="checkbox"/>	...		המשר Shinoyim-
	CADDRESS	<input type="checkbox"/>	VARCHAR2(55)	<input type="checkbox"/>	...		
	CNAME	<input type="checkbox"/>	VARCHAR2(30)	<input type="checkbox"/>	...		
	CGENDER	<input type="checkbox"/>	VARCHAR2(5)	<input type="checkbox"/>	...		
	CID	<input type="checkbox"/>	NUMBER(5)	<input type="checkbox"/>	...		
	CMAIL	<input type="checkbox"/>	VARCHAR2(35)	<input type="checkbox"/>	...		
*	CMOBILE	<input checked="" type="checkbox"/>	INTEGER	<input type="checkbox"/>	...		

המשר Shinoyim-
שינוי ID של סטודנט - כדי להמנע מקרה בו קיימں כבר מטופל עם ID זהה נעה את ID
של הסטודנט למעבר לשול המטופל כר-

```
--Change the id to bigger to evoide troubles
UPDATE STUDENT
SET student_id = student_id + 1000;
```

בשביל לבצע זאת היה צריך גם להגדיל את גודל המשתנה של ID שהיה 3, העלהנו ל-5 וcutת
הוא תואם את ההגדרה שלנו.
כמובן שבדקנו אם יש טבלאות התלוויות בID ויכולות להיפגע מהשינוי אך לא היו טבלאות
שקייםו כמפתח זר את המפתח של הסטודנט.

העברת הנתונים בפועל מהסטודנט למטופל
(הנתונים הריקים הם של הסטודנט שעברו, לא יהיה לו את הנתונים האלה מה שמקל על
הזהוי). כמובן שכחלק מהשינוי איפשרנו להכניס NULL בנתונים אלו בטבלת המטופל
דאגנו לאחד בין נתונים השמות כפי שציינו בדרישות וכן במקום תאזר לידה בזמן ההמרה
שלפנו את שנת הלידה ואותה שמרנו

CBIRTHYEAR	CADDRESS	CNAME	CGENDER	CID	CMAIL	CMOBILE
533	2013	... Jonathan Moore ...	443	540772325
534	2007	... Cecilia Mitchell ...	450	521163097
535	2012	... Albert Rogers ...	452	528833675
536	2017	... Dennis Hernandez ...	460	568083338
537	2016	... Isaiah Allen ...	461	530961548
538	2013	... Laura Lewis ...	462	557403039
539	2008	... Sheila Nguyen ...	465	585737384
540	2010	... Adriana Thomas ...	472	520359017
541	2009	... Leah Taylor ...	474	545745339
542	2007	... Lydia Miller ...	477	559178362
543	2011	... Kristen Hernandez ...	482	528972190

כעת יהיה צריך למחוק מטבלת הסטודנט את כל הנתונים שעברו למטופל ולהשאיר את התכונות הייחודיות של הסטודנט עצמו.

```
-- Drop columns that are now in PATIENT
ALTER TABLE STUDENT DROP COLUMN first_name;
ALTER TABLE STUDENT DROP COLUMN last_name;
ALTER TABLE STUDENT DROP COLUMN birth_date;
ALTER TABLE STUDENT DROP COLUMN phone;

-- Add foreign key reference to PATIENT
ALTER TABLE STUDENT ADD CONSTRAINT FK_STUDENT_PATIENT
FOREIGN KEY (student_id) REFERENCES PATIENT(cid);

-- Rename student_id to match PATIENT's primary key name
ALTER TABLE STUDENT RENAME COLUMN student_id TO cid;

-- Add primary key constraint
ALTER TABLE STUDENT ADD CONSTRAINT PK_STUDENT PRIMARY KEY (cid);
```

בסוף טבלה הסטודנט נראה כך:

The screenshot shows the 'Keys' tab of the table structure editor for the STUDENT table. The table has four columns: CID (primary key), FATHERNAME, MOTHERNAME, and CLASS_ID. The CLASS_ID column is nullable and has a default value.

Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments
CID	<input type="checkbox"/>	NUMBER(5)	<input type="checkbox"/>			
FATHERNAME	<input type="checkbox"/>	VARCHAR2(15)	<input type="checkbox"/>			
MOTHERNAME	<input type="checkbox"/>	VARCHAR2(15)	<input type="checkbox"/>			
CLASS_ID	<input type="checkbox"/>	NUMBER(3)	<input checked="" type="checkbox"/>			

```
select * from student
```

The screenshot shows a database interface with a toolbar at the top containing various icons for operations like insert, update, delete, and search. Below the toolbar is a table titled 'student'. The table has columns: CID, FATHERNAME, MOTHERNAME, and CLASS_ID. The data consists of 20 rows, each representing a student record. The first few rows are: 1. CID: 1001, FATHERNAME: Derek, MOTHERNAME: Deborah, CLASS_ID: 31; 2. CID: 1003, FATHERNAME: Lucas, MOTHERNAME: Samantha, CLASS_ID: 10; 3. CID: 1004, FATHERNAME: Charles, MOTHERNAME: Cheryl, CLASS_ID: 32; 4. CID: 1005, FATHERNAME: Douglas, MOTHERNAME: Brenda, CLASS_ID: 7; etc.

	CID	FATHERNAME	MOTHERNAME	CLASS_ID
▶	1	1001	Derek	Deborah
	2	1003	Lucas	Samantha
	3	1004	Charles	Cheryl
	4	1005	Douglas	Brenda
	5	1007	Henry	Jennifer
	6	1009	Joseph	Patricia
	7	1010	Adam	Lucia
	8	1012	Patrick	Janet
	9	1014	Travis	Pamela
	10	1015	Justin	Helena
	11	1016	Ethan	Mia
	12	1017	Eric	Stephanie
	13	1018	Diego	Phoebe
	14	1019	Walter	Phoebe
	15	1020	Ronald	Lisa
	16	1025	Paul	Brenda
	17	1028	Richard	Agnes
	18	1029	Austin	Abigail
	19	1030	Eric	Natalie
	20	1031	Walter	Alicia

32:1

SYSTEM@XE

436 rows selected in 0.234 se

נמchioש שהפרטים באמת עברו על ידי שנראה את שאר פרטיו של סטודנט 1001 בפרטיו-

```
select * from patient  
where cId = 1001
```

The screenshot shows a database interface with a toolbar at the top containing various icons for operations like insert, update, delete, and search. Below the toolbar is a table titled 'patient'. The table has columns: CBIRTHYEAR, CADDRESS, CNAME, CGENDER, CID, CMAIL, and CMOBILE. The data consists of 1 row, representing the patient record for CID 1001. The details are: CBIRTHYEAR: 2011, CADDRESS: ... Alexandra Wright ..., CGENDER: ..., CID: 1001, CMAIL: ..., and CMOBILE: 541868000.

	CBIRTHYEAR	CADDRESS	CNAME	CGENDER	CID	CMAIL	CMOBILE
▶	1	2011	... Alexandra Wright	1001	...	541868000

קשר בין סטודנט לשכר לימוד

שינוי של- סטודנטים יכולים לשלם על שכר לימוד (קשר בין סטודנט לתשלום)

זה מחייב מאיתנו כמה שינויים בטבלה תשלום.

הנה היא לפני שינוי-

Type	owner	Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments
		ID	<input type="checkbox"/>	NUMBER(5)	<input type="checkbox"/>
		TOTALPRICE	<input type="checkbox"/>	NUMBER(10)	<input type="checkbox"/>
		PDATE	<input type="checkbox"/>	DATE	<input type="checkbox"/>
		APPOINTMENTID	<input type="checkbox"/>	NUMBER(5)	<input type="checkbox"/>
*		*	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

עת השינויים- נרצה להוסיף שיהיה סוג עבור התשלום והגבלות כי אם הסטודנט רוצה לשלם עבור שכר לימוד שלא יוכל להכניס את הת.ז. של התור שם כי זה לא תשלום עבור זה. וכן תשלום עבור תור לא יוכל להכניס את הID של הסטודנט מסיבה דומה.
לכן נוסיף עמודה של סוג תור וכן עמודה של ת.ז. סטודנט שהוא מפתחZR מסטודנט

נסמן את כל התשלומים שהיו עד כה כתשלומים של תורים ונוסף בדיקה

```
SQL Output Statistics
ALTER TABLE Payment
ADD PaymentType VARCHAR2(20) CONSTRAINT CK_PaymentType CHECK (PaymentType IN ('School', 'Appointment')
ADD Student_ID NUMBER(5)
MODIFY AppointmentID NUMBER(5) NULL;

-- Add a foreign key for Student_ID
ALTER TABLE Payment
ADD CONSTRAINT FK_Payment_Student
FOREIGN KEY (Student_ID) REFERENCES Student(cid);

-- all the existing records are payments for appointments
UPDATE Payment
SET PaymentType = 'Appointment'
WHERE AppointmentID IS NOT NULL;

-- Modify the constraint to allow for non-student appointment payments
ALTER TABLE Payment
ADD CONSTRAINT CK_Payment_Type
CHECK (
    (PaymentType = 'School' AND Student_ID IS NOT NULL AND AppointmentID IS NULL) OR
    (PaymentType = 'Appointment' AND AppointmentID IS NOT NULL)
);

select * from payment
```

ID	TOTALPRICE	PDATE	APPOINTMENTID	PAYMENTTYPE	STUDENT_ID
1	37328	2308	20/12/2024	Appointment	...
2	55543	1465	12/09/2026	Appointment	...
3	17421	1915	03/04/2026	Appointment	...
4	21271	640	02/03/2027	Appointment	...
5	94623	1397	19/04/2028	Appointment	...
6	48576	1215	04/07/2025	Appointment	...
7	15756	2306	13/09/2025	Appointment	...
8	68514	2388	09/04/2025	Appointment	...
9	14935	1009	16/11/2024	Appointment	...
10	46339	743	22/08/2026	Appointment	...
11	59368	882	05/09/2025	Appointment	...
12	88362	2445	19/01/2026	Appointment	...
13	55971	994	13/01/2027	Appointment	...
14	42529	167	28/10/2027	Appointment	...

העברה נתונים מ teacher ל-stuff

stuff לפני השינוי:

Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments
SADDRESS		VARCHAR2(55)				
SMOBILE		CHAR(10)				
SNAME		VARCHAR2(30)				
SMAIL		VARCHAR2(35)				
SID		NUMBER(5)				
*						

staff - השינוי: נבדק שתכונות ה stuff הרצויות יהיו Nullable (לא עשינו שום שינוי במספרים כי הם מתאימים לנו)

Name	Virtual	Type	Nullable	Default/Expr.	Storage	Comments
SADDRESS		VARCHAR2(55)	✓			
SMOBILE		CHAR(10)	✓			
SNAME		VARCHAR2(30)				
SMAIL		VARCHAR2(35)	✓			
SID		NUMBER(5)				
*						

השינויים: מראה לפני ואחרי:
(עשינו גם בדיקה לראות שה id לא חוזר על עצמו גם במורה וגם בצוות)

מספר מ: stuff

```
--check tables;
select count(*) from staff;
select count(*) from teacher;

--insert the data from teacher to staff
insert into staff (sid, sname)
select teacher_id as sid, tfirst_name || ' ' || tlast_name
from teacher;
```

Select staff | Select teacher

COUNT(*)	419
----------	-----

מספר מ: teacher

```
--check tables;
select count(*) from staff;
select count(*) from teacher;

--insert the data from teacher to staff
insert into staff (sid, sname)
select teacher_id as sid, tfirst_name || ' ' || tlast_name
from teacher;
```

Select staff		Select teacher															
		COUNT(*)	1	396													

מספר ב stuff לאחר שינוי: (רואים שהמספר גדול)

```
--check tables;
select count(*) from staff;
select count(*) from teacher;

--insert the data from teacher to staff
insert into staff (sid, sname)
select teacher_id as sid, tfirst_name || ' ' || tlast_name
from teacher
where teacher.teacher_id not in (select sid from staff);
```

Select staff		Select teacher															
		COUNT(*)	1	815													

מחיקת עמודות עם שם- כי זה נמצא באבא:

```
select * from teacher;

--insert the data from teacher to staff
insert into staff (sid, sname)
select teacher_id as sid, tfirst_name || ' ' || tlast_name
from teacher
where teacher.teacher_id not in (select sid from staff);

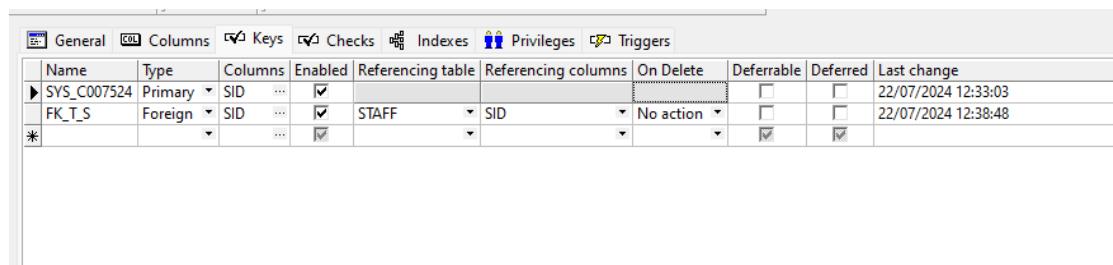
--drop unnecessary columns from teacher
alter table teacher
drop column tfirst_name;
alter table teacher
drop column tlast_name;
```

TEACHER_ID	HOURLY_SALARY	BONUS
1	312	35
2	313	35
3	314	35
4	316	35
5	317	35
6	320	35
7	321	35
8	322	35
9	323	35
10	326	38.5
11	327	35
12	328	35
13	329	35
14	330	35
15	331	35
16	332	35
17	333	35
18	334	35
19	335	35

system@XE 19 rows selected in 0.062 seconds (more...)

טיפול במפתח:

עכשו נרצה ש: ID teacher יהיה מפתח זר – אני אשנה את שם המפתח:



Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Last change
SYS_C007524	Primary	...	<input checked="" type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22/07/2024 12:33:03
FK_T_S	Foreign	...	<input checked="" type="checkbox"/>	STAFF	SID	No action	<input type="checkbox"/>	<input checked="" type="checkbox"/>	22/07/2024 12:38:48
*									

ושינויו את המספר מ-3-> 5 (כי הוא גדול מビיניהם).

VIEW בית ספר אגף

תצוגה שמחזירה נתונים על שיעורים:
 מספר השיעור, שמו, יום השבוע והשעה שבה הוא חל, ת.ז. ושם המורה של אותה כיתה
 ומספר האנשים הרשומים לכיתה

```

SQL Output Statistics
CREATE OR REPLACE VIEW ClassDetailedOverview AS
SELECT
    1.Lesson_ID AS LessonNumber,
    subj.Subject_Name AS LessonName,
    1.Lesson_Day AS DayOfWeek,
    1.Lesson_Hour AS Time,
    lst.sname AS TeacherName,
    c.Class_ID,
    COUNT(s.cid) AS NumberStudents
FROM
    Class_ c
LEFT JOIN Room r ON c.Room_ID = r.Room_ID
LEFT JOIN Lesson l ON c.Class_ID = l.Class_ID
LEFT JOIN Subject subj ON l.Subject_ID = subj.Subject_ID
LEFT JOIN Teacher lt ON l.Teacher_ID = lt.SID
LEFT JOIN Staff lst ON lt.SID = lst.sId
LEFT JOIN Student s ON c.Class_ID = s.Class_ID
GROUP BY
    1.Lesson_ID,
    subj.Subject_Name,
    1.Lesson_Day,
    1.Lesson_Hour,
    lst.sname,
    c.Class_ID;
SELECT * FROM ClassDetailedOverview;

```

Create classdetailedoverview		Select classdetailedoverview						
	LESSONNUMBER	LESSONNAME	DAYOFWEEK	TIME	TEACHERNAME	CLASS_ID	NUMBERSTUDENTS	
131	1	World History	...	3	Nigel Wincott	...	18	15
280	2	Sociology	...	3	Amy Conlee	...	1	10
183	3	Chemistry	...	1	Allan Shepard	...	36	9
132	4	Science (General)	...	4	Ossie Wen	...	14	4
231	5	Jewish History	...	4	Catherine Jeter	...	6	10
133	6	Astronomy	...	5	Bernie Himmelman	...	16	8
▶ 1	7	Foreign Language (French)	...	3	Adina Lauper	...	2	13
50	8	Chorus	...	5	Sammy Wariner	...	2	13
281	9	Visual Arts	...	4	Dave Aiken	...	18	15
282	10	Robotics	...	6	Carol O'Connor	...	10	13
51	11	United States History	...	3	Stanley Rhys-Davies	...	17	11
92	12	United States History	...	5	Lois Warden	...	1	10
283	13	Mathematics	...	1	Victoria Borgnine	...	22	11
340	14	Science (General)	...	3	Beverley Goldwyn	...	6	10
93	15	Film Studies	...	1	Scarlett Berry	...	5	10
232	16	Calculus	...	5	Campbell Ness	...	8	7

🕒 & 16:40 | SYSTEM@XE | 389 rows selected in 0.984 seconds

שאילות על VIEW אגף בית ספר

1. מיציאת השיעור הכי פופולרי_Per יום -

מציג את המידע וממי הוא השיעור שאלוי רשומים הכי הרבה סטודנטים ביום מסוים, ולבסוף
מציג עבור כל יום בשבוע מי השיעור המוביל

SQL Output Statistics

```
WITH RankedLessons AS (
    SELECT
        LessonNumber,
        LessonName,
        DayOfWeek,
        Time,
        TeacherName,
        NumberStudents,
        ROW_NUMBER() OVER (PARTITION BY DayOfWeek ORDER BY NumberStudents DESC) AS rn
    FROM
        ClassDetailedOverview
)
SELECT
    DayOfWeek,
    LessonNumber,
    LessonName,
    Time,
    TeacherName,
    NumberStudents
FROM
    RankedLessons
WHERE
    rn = 1;
```

Create classdetailedoverview | Select classdetailedoverview | With

	DAYOFWEEK	LESSONNUMBER	LESSONNAME	TIME	TEACHERNAME	NUMBERSTUDENTS
▶	1	255	Film Studies	8	Hal Stevens	16
	2	400	Robotics	15	David Underwood	17
	3	62	English Language Arts	8	Frankie Thornton	17
	4	315	Business and Entrepreneurship	15	Kathleen Hershey	17
	5	90	Foreign Language (Hebrew)	12	Eliza McLachlan	17
	6	184	Astronomy	15	Lauren Cartlidge	17

2. מציאת המורים המוציאים

מציאה והציגת מי המורים העסוקים ביותר בשבוע, כמה שיעורים הם מעבירים, בכמה נושאים שונים ומה המספר הממוצע של סטודנטים בשיעור

```
SELECT
    TeacherName,
    COUNT(DISTINCT LessonNumber) AS NumOfLessonsPerWeek,
    COUNT(DISTINCT LessonName) AS NumOfDifferentSubjects,
    AVG(NumberStudents) AS AverageStudentsPerLesson
FROM
    ClassDetailedOverview
GROUP BY
    TeacherName
ORDER BY
    NumOfLessonsPerWeek DESC,
    NumOfDifferentSubjects DESC;

WITH RankedLessons AS (
    SELECT
        LessonNumber,
        LessonName,
        DayOfWeek,
        Time,
        TeacherName,
```

	TEACHERNAME	NUMOFLESSONSPERWEEK	NUMOFDIFFERENTSUBJECTS	AVERAGESTUDENTSPERLESSON
30	Dave Aiken	3	2	13.666666666666667
31	Adina Lauper	3	2	13
32	Dean Zappacosta	3	2	13.33333333333333
33	Aaron Payton	2	2	9.5
34	Aida Holmes	2	2	10
35	Amy Conlee	2	2	10
36	Andrea Hawn	2	2	12.5
37	Armand Lucien	2	2	11.5
38	Austin Leachman	2	2	12
39	Azucar Ryder	2	2	11
40	Barbara DeLuise	2	2	9.5
41	Beverley Goldwyn	2	2	12.5
42	Boyd Underwood	2	2	11.5
43	Buddy DeVito	2	2	10.5
44	Cameron Griffin	2	2	15

View מרפאת שיניים:

מבט המציג נתונים של פגישות טיפול Shinim :
סוג טיפול, תאריך, רופא, תשלום

```
CREATE VIEW TreatmentDetails AS
SELECT
    t.tType AS TreatmentType,
    a.aDate AS AppointmentDate,
    s.sName AS DoctorName,
    p.TotalPrice AS Payment
FROM
    Treatment t
JOIN TPreformedInA tpa ON t.tId = tpa.tId
JOIN Appointment a ON tpa.AppointmentID = a.AppointmentID
JOIN Staff s ON a.sId = s.sId
JOIN Doctor d ON s.sId = d.sId
LEFT JOIN Payment p ON a.AppointmentID = p.AppointmentID;

-- לשים מוקדם ב.CommandText:
SELECT * FROM TreatmentDetails;
```

	TREATMENTTYPE	APPOINTMENTDATE	DOCTORNAME	PAYMENT
▶	1 Aesthetic	02/04/2027	Christen Rocha	1749
2	Aesthetic	02/04/2027	Christen Rocha	2108
3	Orthodontics	09/12/2025	Stacey Mcleod	1129
4	Orthodontics	09/12/2025	Stacey Mcleod	1906
5	Rehabilitation	29/08/2028	James White	1294
6	Orthodontics	01/12/2027	Tanner Peters	1627
7	Orthodontics	01/12/2027	Tanner Peters	1696
8	Rehabilitation	12/04/2028	Amena Bowers	506
9	Rehabilitation	15/10/2028	Maite Delacruz	548
10	Rehabilitation	12/04/2028	Amena Bowers	950
11	Rehabilitation	15/10/2028	Maite Delacruz	957
12	Rehabilitation	12/04/2028	Amena Bowers	1853
13	Aesthetic	26/01/2029	Minerva Knowles	460
14	Aesthetic	26/01/2029	Minerva Knowles	1139
15	Orthodontics	30/01/2029	Haviva Roth	1093
16	Rehabilitation	26/08/2029	Ahmed Carey	
17	Aesthetic	17/02/2024	Todd Jimenez	
18	Aesthetic	20/11/2029	Liam Martinez	
19	Aesthetic	07/06/2029	Hanna Whitfield	2106
20	Aesthetic	07/06/2029	Hanna Whitfield	430
21	Aesthetic	27/05/2029	Cheryl Rojas	292
22	Aesthetic	21/04/2027	Zena Howard	1132

ABC AB: "AB" 16:1 system@XE 22 rows selected in 0.11 seconds (more...)

שאילתה 1:

השאילתה מוצאת את הרופא המטפל בהכי הרבה מטופלים, ומספקת מידע על כמות הטיפולים שלו וממוצע התשלומים עבורם

```
WITH DoctorTreatments AS (
    SELECT
        DoctorName,
        COUNT(*) AS TreatmentCount,
        AVG(Payment) AS AverageTreatmentCost
    FROM
        TreatmentDetails
    GROUP BY
        DoctorName
)
SELECT
    DoctorName,
    TreatmentCount,
    AverageTreatmentCost
FROM
    DoctorTreatments
WHERE
    TreatmentCount = (SELECT MAX(TreatmentCount) FROM DoctorTreatments);
```

The screenshot shows a database query results window. At the top, there is a toolbar with various icons for managing the query and its results. Below the toolbar, a table is displayed with three columns: DOCTORNAME, TREATMENTCOUNT, and AVERAGE TREATMENT COST. There is one row of data, which corresponds to the result of the query above.

	DOCTORNAME	TREATMENTCOUNT	AVERAGE TREATMENT COST
1	Amena Bowers ...	22	1156.7727272727273

שאילתה 2:

השאילתה מחזירה את הטיפול הכי פופולרי (מופיע בשורה הראשונה) וסדר יורד של

הטיפולים לפי הפופולריות שלהם

The screenshot shows a SQL development environment with the following components:

- Toolbar:** Standard database toolbar with icons for connection, schema browser, table editor, and various data manipulation functions.
- Tab Bar:** Shows tabs for "SQL", "Output", and "Statistics".
- Query Editor:** Displays the following SQL code:

```
SELECT
    TreatmentType,
    COUNT(*) AS TreatmentCount
FROM
    TreatmentDetails
GROUP BY
    TreatmentType
ORDER BY
    TreatmentCount DESC;
```
- Result Grid:** Shows the output of the query as a table:

	TREATMENTTYPE	TREATMENTCOUNT
▶	1 Aesthetic	204
▶	2 Orthodontics	179
▶	3 Rehabilitation	164