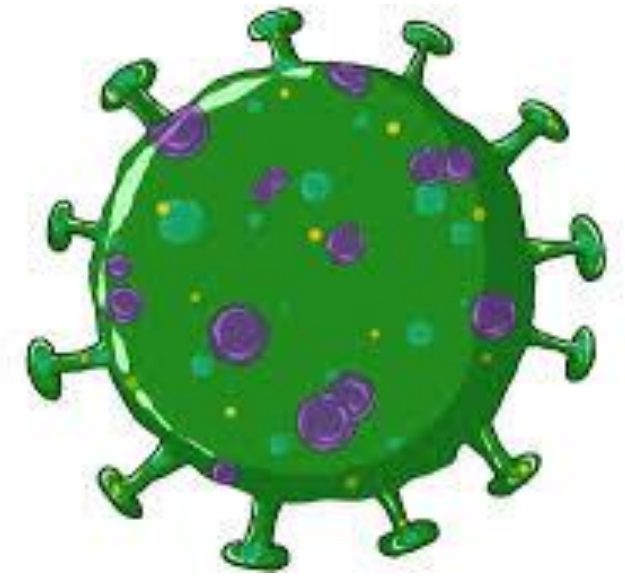
בס"ד

בית החולים שערי צדק
אגף קורונה



מגישות:

שירה יודייקין-פולק 315119461

ביילא ולנר 205823792

תוכן עניינים

שלב ראשון 4
תיאור האגף: 4
יצירת הטבלאות: 5
יצירת הטבלאות הבסיסיות: 5
יצירת 2 הטבלאות החדשות: 5
ווידוא שהטבלאות קיימות ע"י פקודת desc: 6
הכנסת הנתונים לטבלאות באמצעות insert: 7
הכנסת נתונים לטבלאות באמצעות data generator: 7
הוספת נתונים באמצעות text importer: 9
הוספת נתונים באמצעות ODBC importer: 10
בדיקת תכולת הטבלאות באמצעות select: 11
הנדוס לאחור - מעבר מטבלאות לתרשים ERD: 13
תרשים ERD: 14
תרשים DSD: 15
הוכחת צורה נורמלית (3NF \ BCNF): 15
גיבוי: 16
שלב שני 18
שאילתות: 18
שאילתה 1: 18
שאילתה 2: 18
שאילתה 3: 19
שאילתה 4: 19
שאילתה 5: 20
שאילתה 6: 20
אינדקסים 21
Updates: 24
עדכון 1: 24
עדכון 2: 26
Deletes: 28
מחיקה 1: 28
מחיקה 2: 29
דוחות: 31
דוח 1: 31
דוח 2: 32
דו"ח 3: 33
דו"ח 4: 34

שלב שלישי 35
הקדמה: 35
תיאור אגף מערך המח: 35
האינטגרציה 37
הוספת נתונים לטבלאות החדשות: 39
תרשים ERD מעודכן 41
צורה נורמלית של הטבלאות החדשות: 41
התאמת השאילתות לבסיס הנתונים המעודכן: 42
שאילתות על בסיס הנתונים החדש: 43
שאילתה 1: 43
שאילתה 2: 43
שאילתה 3: 43
Views: 44
view #1: 44
view #2: 45
שלב רביעי 47
פונקציות: 47
פונקציה 1: 47
פונקציה 2: 49
פונקציה 3: 51
פונקציה 4: 53
פרוצדורות: 55
פרוצדורה 1: 55
פרוצדורה 2: 57
טריגרים: 60
טריגר 1: 60
טריגר 2: 61
טריגר 3: 63
טריגר 4: 64
תכניות: 66
תכנית 1: 66
תכנית 2: 67

שלב ראשון

תיאור האגף:

אגף הקורונה הוא אגף חדש בבית החולים 'שערי צדק'. אגף זה נפתח במטרה לאפשר אשפוז וטיפול בתנאי בידוד מחמירים באנשים שנדבקו וחלו בנגיף הקורונה.

(הערה: בתיאור הטבלאות בשורות הבאות, השדה המסומן בקו תחתי הוא שדה מפתח).

בטבלה **client** נשמור את פרטי המטופל: מספר זהות, שם, כתובת, האחות המטפלת, מספר טלפון, שנת לידה, האם הוא נמצא בקבוצת סיכון (0 או 1), ומצב המחלה (קל, בינוני או קשה).

הטבלאות **city** ו-**area** שומרות את העיר ממנה מגיע המטופל ובאיזה אזור בארץ העיר נמצאת.

הטבלה **nurse** מאחסנת נתונים אודות האחיות המטפלות בחולי המחלקה: מספר זהות, שם, אזור מגורים, דירוג, השנה בה התחילה לעבוד, תעודת זהות של הבוס שלה, ומשכורת.

הטבלה **schedule** מייצגת פגישה בין מטופל ואחות רפואית ומכילה: ת.ז. של האחות, ת.ז. של המטופל, ותאריך הפגישה.

בטבלה **respirator** נאחסן מידע אודות מכונות ההנשמה שיש באגף: מספר המכונה, המטופל שמחובר אליה, תאריך האחרון בו נבדקה, ומספר ימי השימוש של המטופל הנוכחי.

הטבלה **epidemiology** מאחסנת את נתוני החקירה האפידמיולוגית שנערכה עבור כל מטופל: ת.ז. של המטופל, היכן שהה, מתי, ומספר האנשים שנחשפו אליו באותו זמן ומקום ויכולים היו להידבק.

יצירת הטבלאות:

יצירת הטבלאות הבסיסיות:

```
create table area(areaId   number(3) primary key,
                  areaName varchar(20));

create table city(cityName varchar(20) primary key,
                  areaId   number(3) references area);

create table nurse(nurseID    number(9) primary key,
                   nurseName  varchar(20),
                   areaId     number(3) references  area,
                   rating     number(2),
                   hireYear   number(4),
                   bossId     number(9),
                   salary     float);

create table client(clientId         number(9) primary key,
                    clientName       varchar(20),
                    nurseId          number(9) references  nurse,
                    phoneNr          varchar(10),
                    Address          varchar(25),
                    cityName         varchar(20) references city,
                    yearOfBirth      number(4),
                    riskGroup        number(1),
                    diseaseCondition varchar(10));

create table schedule(nurseId     number(9),
                      clientId   number(9),
                      meetingTime date,
                      primary key (nurseId, clientId, meetingTime),
                      foreign key (nurseId) references  nurse,
                      foreign key (clientId) references client);
```

יצירת 2 הטבלאות החדשות:

```
create table epidemiology(clientId  number(9),
                          stayPlace varchar(40),
                          stayTime  date,
                          exposedNr number(5),
                          primary key (clientId, stayPlace, stayTime),
                          foreign key (clientId) references client);

create table respirator(respiratorId number(3) primary key,
                        clientId   number(9) UNIQUE references client,
                        daysOfUse    number(5),
                        lastCheck    date);
```

<u>ווידוא שהטבלאות קיימות ע"י פקודת desc:</u>

Command Window

Dialog Editor

```
SQL> desc area
Name     Type          Nullable Default Comments
-------- ------------- -------- ------- --------
AREAID   NUMBER(3)
AREANAME VARCHAR2(20)  Y

SQL> desc city
Name     Type          Nullable Default Comments
-------- ------------- -------- ------- --------
CITYNAME VARCHAR2(20)
AREAID   NUMBER(3)     Y

SQL> desc nurse
Name      Type          Nullable Default Comments
--------- ------------- -------- ------- --------
NURSEID   NUMBER(9)
NURSENAME VARCHAR2(20)  Y
AREAID    NUMBER(3)     Y
RATING    NUMBER(2)     Y
HIREYEAR  NUMBER(4)     Y
BOSSID    NUMBER(9)     Y
SALARY    FLOAT         Y
```

```
SQL> desc client
Name              Type          Nullable Default Comments
----------------- ------------- -------- ------- --------
CLIENTID          NUMBER(9)
CLIENTNAME        VARCHAR2(20)  Y
NURSEID           NUMBER(9)     Y
PHONENR           VARCHAR2(10)  Y
ADDRESS           VARCHAR2(25)  Y
CITYNAME          VARCHAR2(20)  Y
YEAROFBIRTH       NUMBER(4)     Y
RISKGROUP         NUMBER(1)     Y
DISEASECONDITION  VARCHAR2(10)  Y
SQL> desc schedule
Name        Type      Nullable Default Comments
----------- --------- -------- ------- --------
NURSEID     NUMBER(9)
CLIENTID    NUMBER(9)
MEETINGTIME DATE

SQL> desc epidemiology
Name      Type          Nullable Default Comments
--------- ------------- -------- ------- --------
CLIENTID  NUMBER(9)
STAYPLACE VARCHAR2(40)
STAYTIME  DATE
EXPOSEDNR NUMBER(5)     Y
```

```
SQL> desc respirator
Name          Type      Nullable Default Comments
------------- --------- -------- ------- --------
RESPIRATORID  NUMBER(3)
CLIENTID      NUMBER(9) Y
DAYSOFUSE     NUMBER(5) Y
LASTCHECK     DATE      Y
```

הכנסת הנתונים לטבלאות באמצעות insert:

```
-- area insert
INSERT INTO area (areaId,areaName)
VALUES (002,'Jerusalem');

INSERT INTO area (areaId,areaName)
VALUES (003,'Center');

INSERT INTO area (areaId,areaName)
VALUES (004,'North');

INSERT INTO area (areaId,areaName)
VALUES (005,'Judea-Samaria');

INSERT INTO area (areaId,areaName)
VALUES (008,'South');

INSERT INTO area (areaId,areaName)
VALUES (009,'HaSharon');

--city
INSERT INTO city (cityName,areaId)
VALUES ('Acre',4);

INSERT INTO city (cityName,areaId)
VALUES ('Afula',4);

INSERT INTO city (cityName,areaId)
VALUES ('Arad',8);
```

הכנסת נתונים לטבלאות באמצעות data generator:

NURSE

Owner	Table	Number of records
SYS	NURSE	300

Name	Type	Size	Data
NURSEID	NUMBER	9	random(500000000,799999999)
NURSENAME	VARCHAR2	20	firstname + ' ' + lastname
AREAID	NUMBER	3	list(select areaid from area)
RATING	NUMBER	2	random(0,10)
HIREYEAR	NUMBER	4	random(1960,2002)
BOSSID	NUMBER	9	random(800000000,999999999)
SALARY	FLOAT	22	random(3000,15000)

CLIENT

Owner	Table	Number of records
SYS	CLIENT	600

Name	Type	Size	Data
CLIENTID	NUMBER	9	Random(100000000,599999999)
CLIENTNAME	VARCHAR2	20	firstname+' '+lastname
NURSEID	NUMBER	9	List(select nurseid from nurse)
PHONENR	VARCHAR2	10	List(05)+List(0,2,3,4,5,7,8)+Random(1000000,9999999)
ADDRESS	VARCHAR2	25	null
CITYNAME	VARCHAR2	20	List(select cityname from city)
YEAROFBIRTH	NUMBER	4	random(1920,2020)
RISKGROUP	NUMBER	1	list(0,1)
DISEASECONDITION	VARCHAR2	10	list('hard','medium','easy')

SCHEDULE

Owner	Table	Number of records
SYS	SCHEDULE	400

Name	Type	Size	Data
NURSEID	NUMBER	9	List(select nurseid from nurse)
CLIENTID	NUMBER	9	list(select clientid from client)
MEETINGTIME	DATE		random(01/01/2020, 29/07/2020)

EPIDEMIOLOGY

Owner	Table	Number of records
SYS	EPIDEMIOLOGY	1000

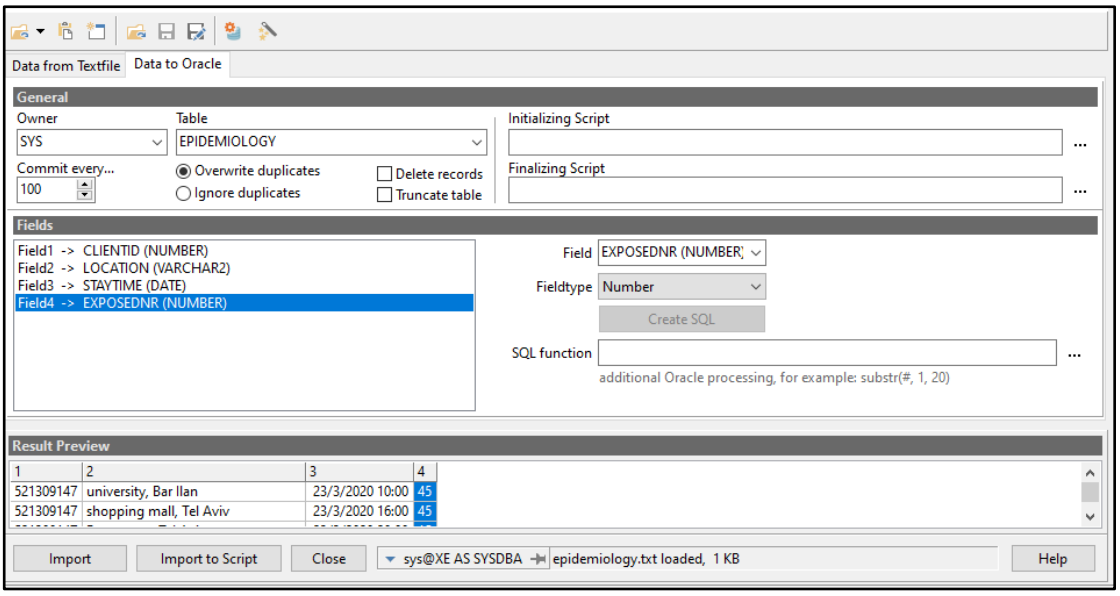
Name	Type	Size	Data
CLIENTID	NUMBER	9	list(select clientid from client)
STAYPLACE	VARCHAR2	40	list('Synagogue','School','Restaurant','Boardwalk','Hotel','Park','Toy Store',
STAYTIME	DATE		random(24/01/2020 06:00:00, 29/04/2020 23:59:59)
EXPOSEDNR	NUMBER	5	random(0,50)

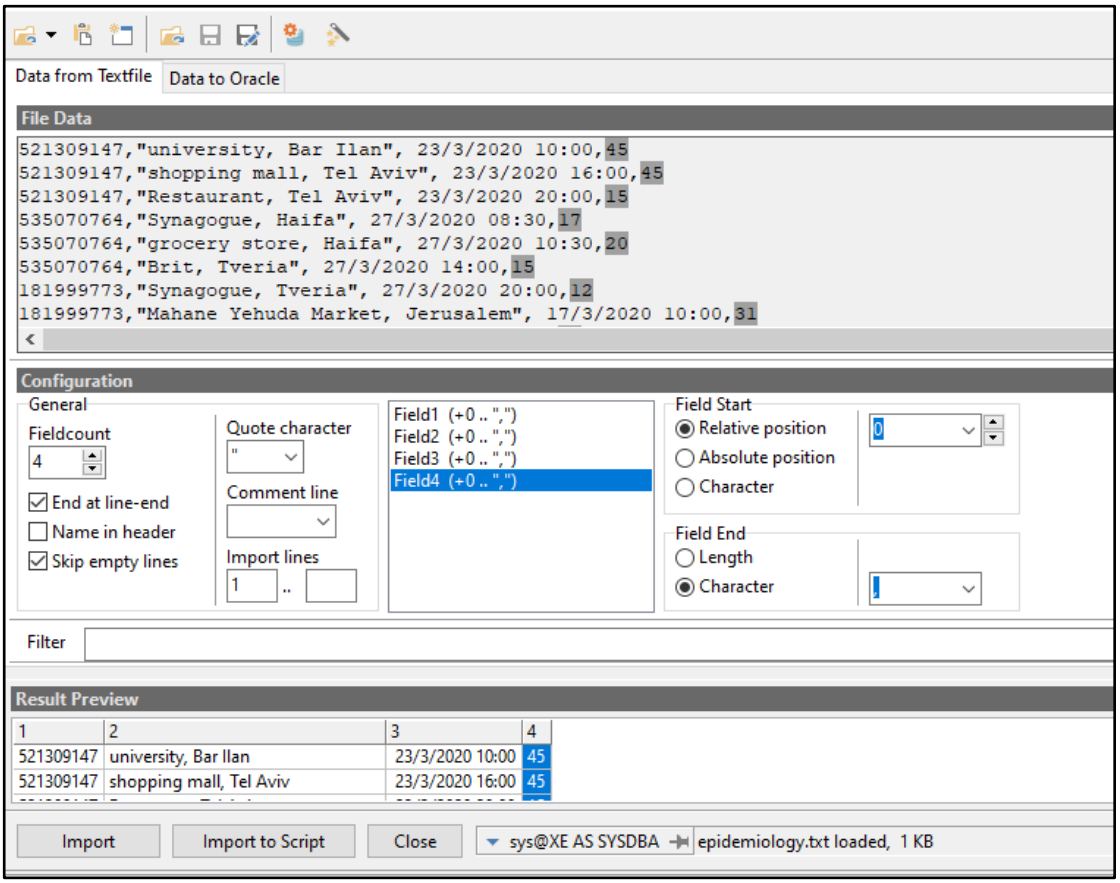
RESPIRATOR

Owner	Table	Number of records
SYS	RESPIRATOR	350

Name	Type	Size	Data
RESPIRATORID	NUMBER	3	Sequence(1, [Inc], [WithinParent])
CLIENTID	NUMBER	9	List(select clientid from client)
DAYSOFUSE	NUMBER	5	random(1,40)
LASTCHECK	DATE		Random(1.1.2000,1.1.2020)

הוספת נתונים באמצעות text importer:

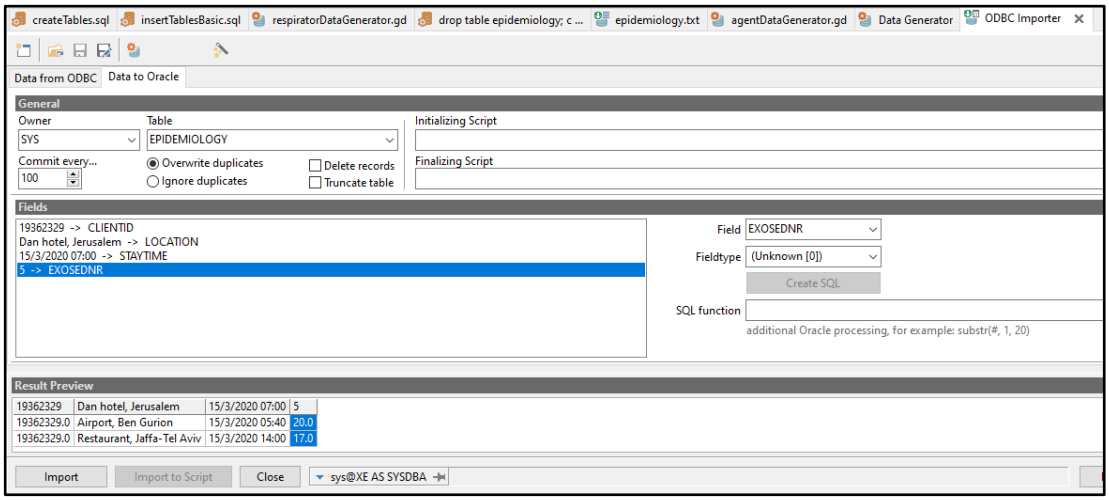


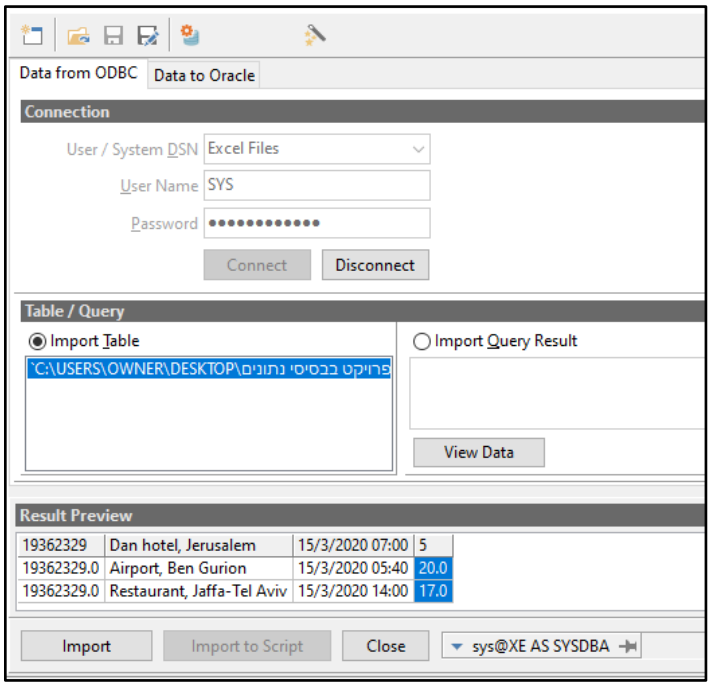


Select city | Select client | Select schedule | Select area | Select agent | Select epidemiology | Select respirator

	CLIENTID	LOCATION	STAYTIME	EXOSEDNR
1	2190855	wedding, Bney Brak	3/4/2020 3:30:00 PM	70
2	2190855	yeshiva, Bney Brak	3/5/2020 10:00:00 PM	20
3	521309147	university, Bar Ilan	3/23/2020 10:00:00 AM	45
4	521309147	shopping mall, Tel Aviv	3/23/2020 4:00:00 PM	45
5	521309147	Restaurant, Tel Aviv	3/23/2020 8:00:00 PM	15
6	535070764	Synagogue, Haifa	3/27/2020 8:30:00 AM	17
7	535070764	grocery store, Haifa	3/27/2020 10:30:00 AM	20
8	535070764	Brit, Tveria	3/27/2020 2:00:00 PM	15
9	181999773	Synagogue, Tveria	3/27/2020 8:00:00 PM	12
10	181999773	Mahane Yehuda Market, Jerusalem	3/17/2020 10:00:00 AM	31
11	103047956	Geula, Jerusalem	3/17/2020 12:30:00 PM	20
12	103047956	Central Station, Jerusalem	3/17/2020 3:00:00 PM	20

הוספת נתונים באמצעות ODBC importer:





בדיקת תכולת הטבלאות באמצעות select:

```
select* from area;
```

Select city | Select client | Select schedule | Select area | Select nurs

	AREAID	AREANAME
1	2	Jerusalem
2	3	Center
3	4	North
4	5	Judea-Samaria
5	8	South
6	9	HaSharon

```
select* from city;
```

Select city | Select client | Select schedule | Select area | Select n

	CITYNAME	AREAID
1	Acre	4
2	Afula	4
3	Arad	8
4	Ariel	5
5	Ashdod	8
6	Ashkelon	8
7	Baqa-Jatt	4
8	Bat Yam	3
9	Beersheba	8
10	Beit Shean	4
11	Beit Shemesh	2
12	Beitar Illit	5
13	Bnei Brak	3
14	Dimona	8
15	Efrath	5
16	Eilat	8
17	Elad	3

```
select* from nurse;
```

Select city | Select client | Select schedule | Select area | Select nurse | Select epidemiology | Select re

	NURSEID	NURSENAME	AREAID	RATING	HIREYEAR	BOSSID	SALARY
1	590153372	Taylor Balk	2	0	1986	862257359	3650
2	780991891	Regina Hoffman	9	10	1987	800546992	10842
3	640648066	Dorry Kotto	2	0	1992	868117967	4698
4	587502517	Deborah Garza	9	0	1997	924470025	3984
5	681311141	Luis Womack	2	0	1976	940150269	7530
6	533605958	Saul Raitt	4	7	2002	861987542	13796
7	553984174	Seth Albright	4	3	1997	990826799	14200
8	515147245	Teri Roth	5	0	1969	871763483	8185
9	536689933	Claire Malone	2	0	2002	896877903	4039
10	587639378	Garry Venora	5	0	1961	820469603	5467
11	607744152	Mia Paltrow	5	0	1982	986791741	7908
12	647513327	Geoff Holm	9	0	1991	870285466	4608
13	727005535	Burton Steiger	4	6	2002	928314285	14443
14	713368031	Diamond Mollard	4	9	1969	966398449	11479
15	517125754	Pierce Levine	5	0	1976	815949319	8305
16	607220395	Avril Madsen	5	0	1988	924503744	5097
17	744799451	Ryan Guilfoyle	2	0	1991	858458439	8438
18	535755295	Demi Evanswood	4	4	1970	918304354	11888

```
select* from client;
```

Select city | Select client | Select schedule | Select area | Select nurse | Select epidemiology | Select respirator

	CLIENTID	CLIENTNAME	NURSEID	PHONENR	ADDRESS	CITYNAME	YEAROFBIRTH	RISKGROUP	DISEASECONDITION
1	150986402	Merrilee Rudd	656425737	0589207382		Kiryat Ata	1961	0	easy
2	531958393	Gilbert Washington	695487349	0534161752		Beit Shemesh	2016	1	hard
3	240964888	Tanya Taha	535682507	0549445062		Safed	1986	0	easy
4	582040138	Diamond Baldwin	601556744	0536188505		Kiryat Yam	1975	0	hard
5	429647767	Rosario Phillips	658882943	0546388954		Rishon LeTzion	1965	1	hard
6	168611931	Rowan Connelly	778881565	0537370285		Kiryat Yam	1955	0	hard
7	203228910	Meg Rooker	502726228	0534170425		Safed	1957	0	easy
8	285483656	Etta Burton	632422277	0582071815		Kfar Saba	2001	1	easy
9	315165897	Mary McCormack	559610641	0587131530		Haifa	1939	0	easy
10	475436677	Tony Schiff	585926065	0521069999		Beitar Illit	2000	0	medium
11	311267194	Queen Sewell	722335567	0502151084		Ramat HaSharon	1951	1	hard
12	340076808	Davy Moraz	744739204	0553363516		Ramat HaSharon	2018	1	hard
13	316375915	Timothy Levy	635132755	0535956684		Nazareth Illit	1933	0	hard
14	425512863	Olympia Kelly	717849505	0506080709		Tayibe	1958	1	medium
15	348868753	Bo Palminteri	593843322	0577039576		Kiryat Ata	1947	1	hard
16	250317391	Hikaru Morton	587387117	0582799343		Kiryat Gat	2014	1	medium

```
select* from schedule;
```

Select city | Select client | Select schedule | Select ar

	NURSEID	CLIENTID	MEETINGTIME
1	501290343	278205588	7/27/2020
2	502726228	315165897	5/29/2020
3	502726228	516662620	5/18/2020
4	503241483	348647686	1/5/2020
5	503241483	583745521	1/8/2020
6	503241483	591891431	2/13/2020
7	507601433	142033424	1/24/2020
8	507601433	148299651	1/31/2020
9	507601433	472631221	3/1/2020

```
select* from epidemiology;
```

Select city | Select client | Select schedule | Select area | Select nurse | Select epidemiolog

	CLIENTID	STAYPLACE	STAYTIME	EXPOSEDNR
1	117704307	Hospital, Beit Shean	2/7/2020 6:18:25 AM	8
2	552386882	Central Station, Arad	3/15/2020 10:41:31 AM	37
3	463293385	University, Yokneam	1/29/2020 3:21:10 PM	6
4	512464379	Super Market, Kiryat Motzkin	3/27/2020 7:06:48 AM	16
5	384642471	Boardwalk, Even Yehuda	2/4/2020 5:54:31 PM	1
6	447201419	Hospital, Kfar Witkin	2/28/2020 4:16:10 AM	46
7	448481561	Synagogue, Raanana	4/28/2020 3:25:55 PM	32
8	473447591	Hospital, Sakhnin	3/16/2020 11:45:01 PM	24
9	183348504	Central Station, Tayibe	3/9/2020 7:48:00 AM	37
10	500940177	Park, Ashdod	2/26/2020 8:03:41 PM	46
11	449153460	Hospital, Acre	2/4/2020 2:42:17 PM	5
12	328941647	Restaurant, Nazareth	2/15/2020 12:35:25 PM	5
13	218392459	Hotel, Hertzliya	3/23/2020 12:22:36 AM	30
14	583654705	Park, Haifa	3/16/2020 4:18:28 PM	17

```
select* from respirator;
```

Select city | Select client | Select schedule | Select area | Select nu

	RESPIRATORID	CLIENTID	DAYSOFUSE	LASTCHECK
1	1	164410483	10	7/12/2007
2	2	199122531	11	10/26/2006
3	3	184444876	34	8/24/2016
4	4	407146650	11	6/3/2014
5	5	387848966	23	1/11/2002
6	6	240254927	5	9/10/2000
7	7	506111648	5	12/1/2008
8	8	419503903	18	10/9/2004
9	9	313232464	17	3/18/2007
10	11	589490221	27	8/11/2004
11	12	227340333	32	11/18/2003

הנדוס לאחור - מעבר מטבלאות לתרשים ERD:

1. עבור כל טבלה:
2. נוסיף את הטבלה כישות.
3. נעבור על כל אחת מעמודות הטבלה:

3.1. אם התכונה היא מפתח זר – לא נוסיף אותה כתכונה לישות אלא נוסיף מהישות הנוכחית קשר לישות אליה שייך המפתח הזר. אם הטבלה הזו לא קיימת – נעצור כאן, נבנה את הטבלה, ונחזור למקום בו עצרנו.

הצד בו אנו נמצאים הוא הרבים, הצד השני הוא היחיד.

(בחלק מהמקרים יהיה זה קשר של יחיד ליחיד, יש לבדוק לפי ההקשר. בקשר יחיד ליחיד אחד הקשרים מכיל את המפתח של הקשר השני).

3.2. אם בטבלה שני מפתחות זרים שהם גם מפתחות ראשיים בשתי טבלאות אחרות – זהו קשר של רבים לרבים וזו לא ישות.

3.3. אם יש תכונה שהיא מפתח זר וגם מפתח ראשי – זו ישות חלשה.

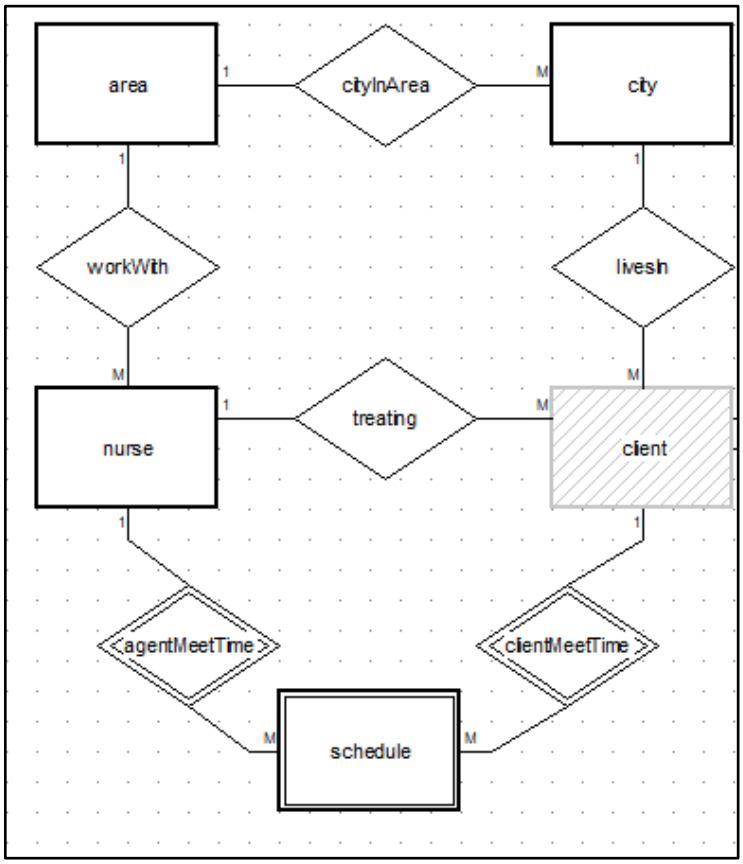
3.4. אחרת – נוסיף את העמודה כישות לתכונה.

3.5. אם התכונה היא מפתח ראשי – נציין זאת.

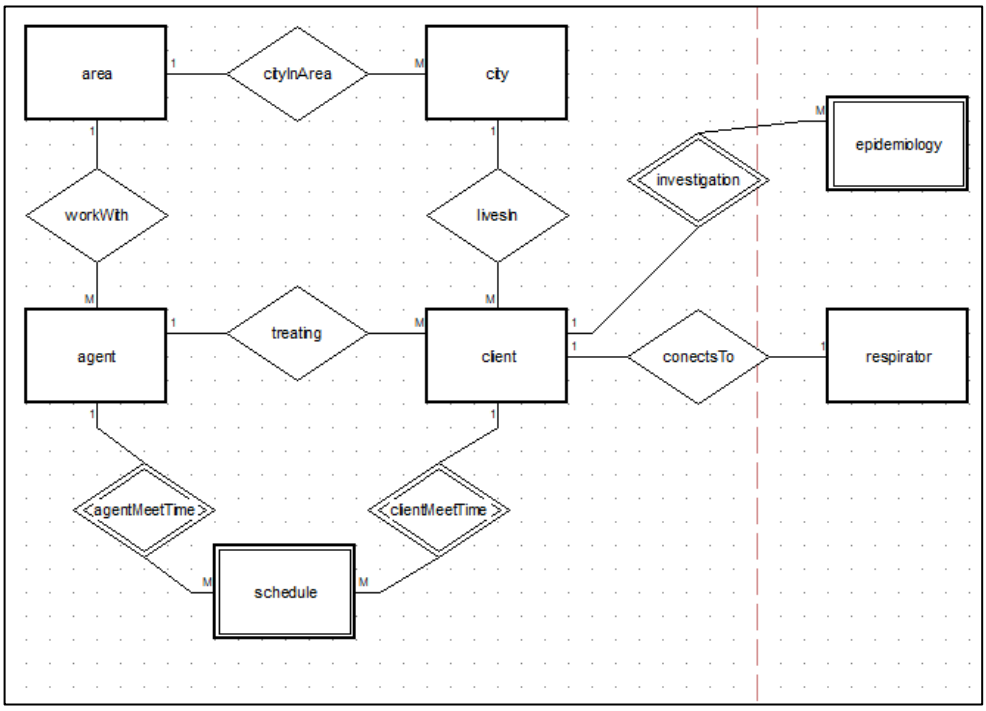
3.6. אם יש טבלה A שמכילה את כל התכונות של טבלה B ועוד תכונות נוספות, אז A יורשת מ-B.

תרשים ERD:

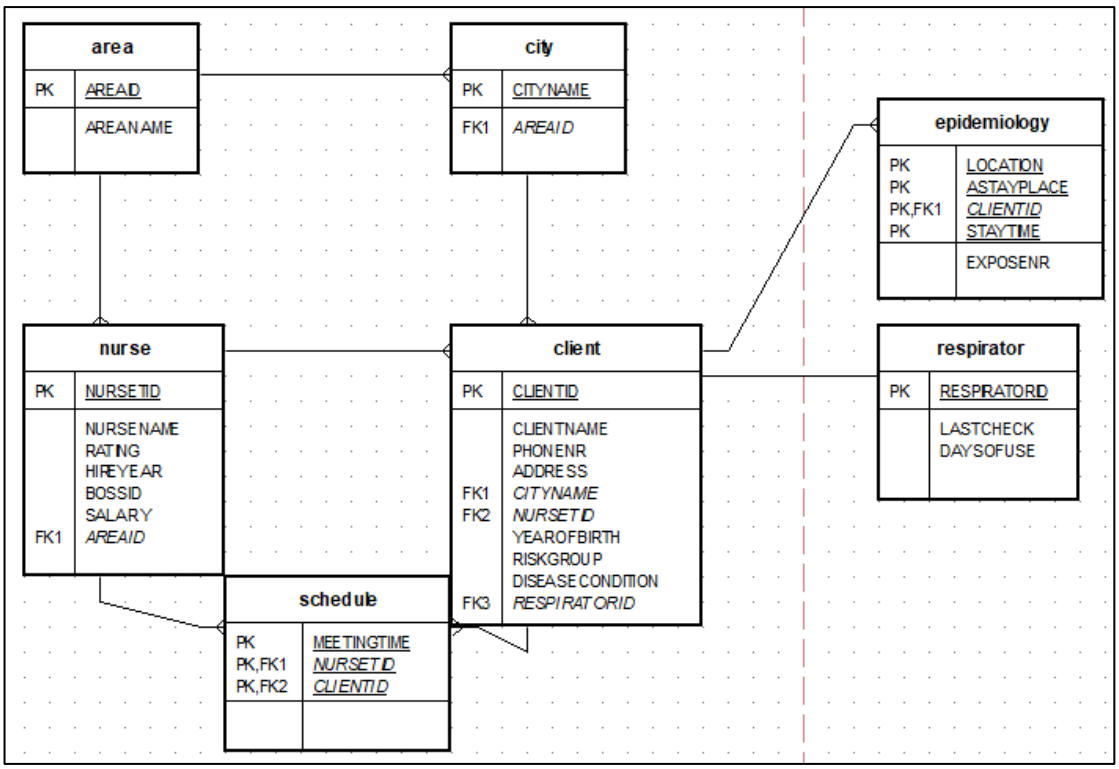
לפני הוספת 2 הישויות:



אחרי הוספת 2 הישויות:



תרשים DSD:



הערה: התכנה לקחה באופן אוטומטי את המפתח של respirator כמפתח זר לטבלת client. אך בבסיס הנתונים אנחנו השתמשנו במפתח של client כמפתח זר בטבלה של respirator.

הוכחת צורה נורמלית (3NF \ BCNF):

טבלת area: areaId -> areaName

טבלת city: cityName -> areaId

טבלת nurse: nurseId -> nurseName, areaId, Rating, HireYear, bossId, salary

טבלת client: clientId -> clientName, nurseId, phonNr, address, cityName, yearOfBirth, riskGroup, diseaseCondition

טבלת schedule: nurseId, clientId, meetingTime (המפתח הוא כל השדות)

טבלת epidemiology: clientId, location, stayTime -> exposeNr

טבלת respirator: respiratorId -> clientNr, lastCheck, daysOfUse

בכל הטבלאות, עבור כל תלות, הגורר הוא מפתח <u>על</u> ולכן הטבלאות כולן הן מנורמלות בצורת BCNF וכמובן שמנורמלות גם בצורת 3NF.

גיבוי:

select* from city; select* ... | Command Window | insertTablesExtra.sql | Export Tables of SYS

Name	Type	Compiled
ADO_IMPARAM$	TABLE	2/4/2019 4:51:41 AM
ADO_IMSEGSTAT$	TABLE	2/4/2019 4:51:41 AM
ADO_IMSEGTASKDETAILS$	TABLE	2/4/2019 4:51:41 AM
ADO_IMSTAT$	TABLE	2/4/2019 4:51:41 AM
ADO_IMTASKS$	TABLE	2/4/2019 4:51:41 AM
AGENT	TABLE	3/23/2020 11:10:37 PM
ALERT_QT	TABLE	2/4/2019 5:07:56 AM
ALL_UNIFIED_AUDIT_ACTIONS	TABLE	2/4/2019 5:03:19 AM
APPLY$_AUTO_CDR_COLUMN_GROUPS	TABLE	2/4/2019 4:45:53 AM
APPLY$_BATCH_SQL_STATS	TABLE	2/4/2019 4:45:53 AM
APPLY$_CDR_INFO	TABLE	2/4/2019 4:45:53 AM
APPLY$_CHANGE_HANDLERS	TABLE	2/4/2019 4:45:52 AM
APPLY$_CONF_HDLR_COLUMNS	TABLE	2/4/2019 4:45:52 AM
APPLY$_CONSTRAINT_COLUMNS	TABLE	2/4/2019 4:45:52 AM

Name	Type	Compiled
AQ$_SYS$SERVICE_METRICS_TAB_S	TABLE	2/4/2019 5:11:02 AM
AQ$_SYS$SERVICE_METRICS_TAB_T	TABLE	2/4/2019 5:11:02 AM
AQ_EVENT_TABLE	TABLE	2/4/2019 5:10:49 AM
AQ_PROP_TABLE	TABLE	2/4/2019 5:10:49 AM
AREA	TABLE	3/23/2020 11:10:37 PM
ARGUMENT$	TABLE	2/4/2019 6:04:04 AM
ASSEMBLY$	TABLE	2/4/2019 4:45:37 AM
ASSOCIATION$	TABLE	2/4/2019 4:45:46 AM
ATEMPTAB$	TABLE	2/4/2019 4:45:26 AM
ATTRCOL$	TABLE	2/4/2019 5:19:35 AM
ATTRIBUTE$	TABLE	2/4/2019 4:45:48 AM
ATTRIBUTE_TRANSFORMATIONS$	TABLE	2/4/2019 4:46:04 AM
AUD$	TABLE	2/4/2019 4:45:45 AM
AUDIT$	TABLE	2/4/2019 4:45:42 AM

select* from city; select* ... | Command Window | insertTablesExtra.sql | Export Tables of SYS

Name	Type	Compiled
CHNF$_QUERY_DEPENDENCIES	TABLE	2/4/2019 4:52:36 AM
CHNF$_QUERY_OBJECT	TABLE	2/4/2019 4:52:36 AM
CHNF$_REG_QUERIES	TABLE	2/4/2019 4:52:36 AM
CHNFDIRECTLOAD$	TABLE	2/4/2019 4:52:37 AM
CITY	TABLE	3/23/2020 11:10:37 PM
CLIENT	TABLE	3/23/2020 11:10:37 PM
CLI_INST$	TABLE	2/4/2019 4:46:03 AM
CLI_LOG$	TABLE	2/4/2019 4:46:03 AM
CLI_PART$	TABLE	2/4/2019 4:46:04 AM
CLI_TAB$	TABLE	2/4/2019 4:46:03 AM
CLI_TS$	TABLE	2/4/2019 4:46:04 AM
CLST$	TABLE	2/4/2019 4:45:26 AM
CLSTDIMENSION$	TABLE	2/4/2019 4:45:26 AM
CLSTJOIN$	TABLE	2/4/2019 4:45:26 AM

Name	Type	Compiled
EDITIONING_TYPES$	TABLE	2/4/2019 4:45:22 AM
ENC$	TABLE	2/4/2019 4:45:45 AM
EPG$_AUTH	TABLE	2/4/2019 5:20:01 AM
EPIDEMIOLOGY	TABLE	3/28/2020 11:14:24 PM
ERROR$	TABLE	2/4/2019 4:45:37 AM
EV$	TABLE	2/4/2019 4:45:23 AM
EVCOL$	TABLE	2/4/2019 4:45:23 AM
EXPACT$	TABLE	2/4/2019 4:45:59 AM
EXPDEPACT$	TABLE	2/4/2019 6:09:33 AM
EXPDEPOBJ$	TABLE	2/4/2019 6:09:33 AM
EXPIMP_TTS_CT$	TABLE	2/4/2019 4:52:49 AM
EXPPKGACT$	TABLE	2/4/2019 6:09:33 AM
EXPPKGOBJ$	TABLE	2/4/2019 6:09:33 AM
EXP_HEAD$	TABLE	2/4/2019 4:45:48 AM

Name	Type	Compiled
RESOURCE_MAPPING_PRIORITY$	TABLE	2/4/2019 4:45:41 AM
RESOURCE_PLAN$	TABLE	2/4/2019 4:45:41 AM
RESOURCE_PLAN_DIRECTIVE$	TABLE	2/4/2019 4:45:41 AM
RESOURCE_STORAGE_POOL_MAPPING$	TABLE	2/4/2019 4:45:41 AM
RESPIRATOR	TABLE	3/28/2020 11:14:25 PM
RESULT$	TABLE	2/4/2019 4:45:48 AM
RGCHILD$	TABLE	2/4/2019 4:45:51 AM
RGROUP$	TABLE	2/4/2019 4:45:51 AM
RLS$	TABLE	2/4/2019 4:45:44 AM
RLS_CSA$	TABLE	2/4/2019 4:45:45 AM
RLS_CTX$	TABLE	2/4/2019 4:45:44 AM
RLS_GRP$	TABLE	2/4/2019 4:45:44 AM
RLS_SC$	TABLE	2/4/2019 4:45:44 AM
RMTAB$	TABLE	2/4/2019 4:45:55 AM

select* from city; select* ... | Command Window | insertTablesExtra.sql | Export Tables of SYS

Name	Type	Compiled
RX$$SESSIONS	TABLE	2/4/2019 4:53:26 AM
RX$$SESSION_APPNS	TABLE	2/4/2019 4:53:18 AM
RX$$SESSION_ROLES	TABLE	2/4/2019 4:53:18 AM
SAM_SPARSITY_ADVICE	TABLE	2/4/2019 6:22:23 AM
SCHEDULE	TABLE	3/23/2020 11:10:37 PM
SCHEDULER$_CHAIN	TABLE	2/4/2019 4:52:47 AM
SCHEDULER$_CLASS	TABLE	2/4/2019 4:52:46 AM
SCHEDULER$_CONSTRAINTS	TABLE	2/4/2019 4:52:47 AM
SCHEDULER$_CONSTRAINTS_STATS	TABLE	2/4/2019 4:52:47 AM
SCHEDULER$_CREDENTIAL	TABLE	2/4/2019 4:52:47 AM
SCHEDULER$_DESTINATIONS	TABLE	2/4/2019 4:52:47 AM
SCHEDULER$_EVENT_LOG	TABLE	2/4/2019 4:52:46 AM
SCHEDULER$_EVENT_QTAB	TABLE	2/4/2019 5:07:52 AM
SCHEDULER$_EVTQ_SUB	TABLE	2/4/2019 4:52:46 AM

User <CURRENT USER>

Oracle Export | SQL Inserts | PL/SQL Developer | Log

- [ ] Drop tables
- [ ] Create tables
- [ ] Truncate tables
- [x] Delete records
- [x] Disable triggers
- [ ] Zip
- [x] Disable foreign key constraints
- [x] Include storage
- [x] Include privileges

Commit every 100 records (0 = never)

Where clause

Output file C:\Users\Owner\Desktop\סמסטר ב\פרויקט בבסיסי נתונים\sqlBackUp.sql Export

sys@XE AS SYSDBA Exporting... Done

שלב שני

שאילתות:

שאילתה 1: שמות המטופלים שנמצאים בקבוצת סיכון ומחוברים למכונת הנשמה אך גילם נמוך מ-60.

```
SELECT clientname, (2020-yearofbirth) as age
FROM (SELECT clientid, clientname, yearofbirth
      FROM client
      WHERE riskgroup = 1 and yearofbirth > 1960)
WHERE clientid in (select distinct clientid from respirator);
```

	CLIENTNAME	AGE
1	Gilbert Washington	4
2	Davy Moraz	2
3	Jeremy O'Keefe	9
4	Jimmy Wilson	55
5	Nicholas Griffin	53
6	Balthazar Myles	57
7	Howard Vinton	3
8	Ceili Ponce	35
9	Denise Chappelle	23
10	Temuera Kotto	25
11	Nigel Diddley	48
12	Cheech Gilley	18
13	Terry Sanchez	45
14	Geena Gallagher	6
15	France Bonneville	9

1 of 75 sys@XE AS SYSDBA [11:15:55 PM] 75 rows selected in 0.075 seconds

שאילתה 2: מספר המטופלים החולים קשה, בינוני וקל ממוין לפי מספר החולים.

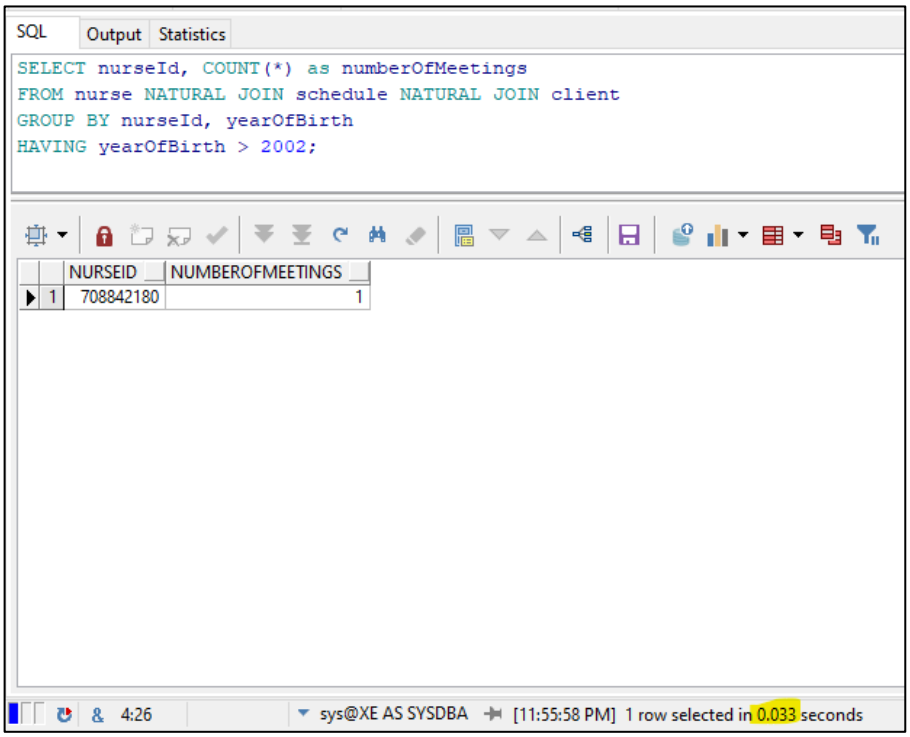
SQL Output Statistics

```
SELECT diseaseCondition, COUNT(diseaseCondition) as number_of_clients
FROM client
GROUP BY diseaseCondition
ORDER BY diseaseCondition;
```

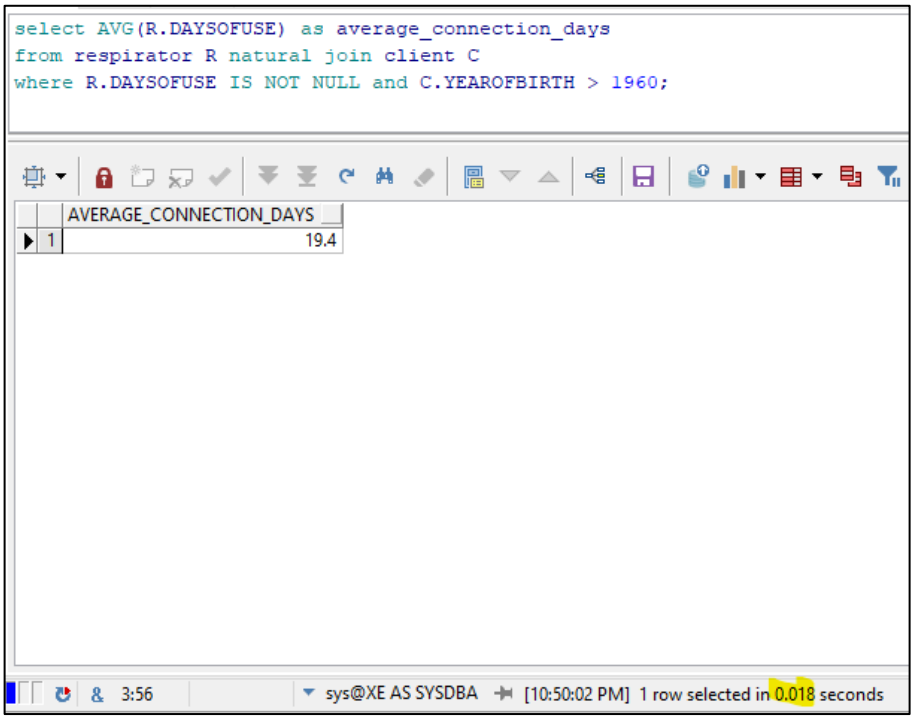
	DISEASECONDITION	NUMBER_OF_CLIENTS
1	easy	206
2	hard	206
3	medium	189

5:1 sys@XE AS SYSDBA [10:02:41 PM] 3 rows selected in 0.032 seconds

<u>שאילתה 3:</u> מספר הפגישות שיש לכל אחות עם מטופלים מתחת לגיל 18 ות.ז. של האחות.



<u>שאילתה 4:</u> מספר הימים הממוצע בו מחוברים למכונת הנשמה מטופלים שגילם גבוה מ-60.



<u>שאילתה 5:</u> ת.ז. של חולים מאזור השרון ומקומות בהם שהו ונחשפו אליהם מעל 30 אנשים בכל אחד מהמקומות.

SQL | Output | Statistics

```
select clientid,stayplace
from client C natural join (select cityname from city C where C.areaid = 9) natural join epidemiology
where exposednr > 30
ORDER BY clientid;
```

	CLIENTID	STAYPLACE
1	100399043	Wedding, Or Yehuda
2	102704705	Hospital, Rosh HaAyin
3	103413980	School, Holon
4	117531117	Central Station, Ashdod
5	121736560	Wedding, Shfar-am
6	121736560	Toy Store, Tirat Carmel
7	133836283	Wedding, Shderot
8	133836283	Wedding, Elad
9	145381088	Boardwalk, Petah Tikva
10	149302323	Super Market, Beit Shean
11	152871335	Toy Store, Or Yehuda
12	183108866	Synagogue, Tayibe
13	201052128	Mall, Umm al-Fahm
14	201052128	Park, Petah Tikva
15	212736567	Hospital, Kfar Witkin
16	232269018	Hotel, Rahat

& 1 of 62 sys@XE AS SYSDBA [8:37:50 PM] 62 rows selected in 0.081 seconds

<u>שאילתה 6:</u> ת.ז של מטופלים מעל גיל 70 שמחוברים למכונת הנשמה שלא נבדקה ב-4 השנים האחרונות, ממוינים לפי הגיל.

```
select clientid
from client natural join respirator
where client.yearofbirth < 1950
      and extract(year from lastcheck) < '2016'
order by yearofbirth;
```

	CLIENTID
1	504743251
2	304867344
3	512156402
4	569310320
5	596253805
6	183092238
7	468643489
8	496509050
9	429140338
10	296108197
11	127091841
12	512464379
13	225040532
14	155378647
15	452494355

& 1:1 sys@XE AS SYSDBA [11:05:15 PM] 63 rows selected in 0.060 seconds

אינדקסים:

אינדקסים המשפיעים על שאילתות 1,5,6:

```
create index idx_yearofbirth on client(yearofbirth);
create index idx_exposednr on epidemiology(exposednr);
create index idx_lastchek on respirator(lastcheck);
```

שאילתות לאחר הרצת האינדקסים:

שאילתה 1: (הסבר- המטופלים ממוינים לפי שנת הלידה שלהם ולכן הגיוני שכעת השליפה של המטופלים שגילם נמוך מ60 מהירה יותר)

```
SELECT clientname, (2020-yearofbirth) as age
FROM (SELECT clientid, clientname, yearofbirth
      FROM client
      WHERE riskgroup = 1 and yearofbirth > 1960)
WHERE clientid in (select distinct clientid from respirator);
```

	CLIENTNAME	AGE
1	Gilbert Washington	4
2	Davy Moraz	2
3	Jeremy O'Keefe	9
4	Jimmy Wilson	55
5	Nicholas Griffin	53
6	Balthazar Myles	57
7	Howard Vinton	3
8	Ceili Ponce	35
9	Denise Chappelle	23
10	Temuera Kotto	25
11	Nigel Diddley	48
12	Cheech Gilley	18
13	Terry Sanchez	45
14	Geena Gallagher	6
15	France Bonneville	9

5:62 sys@XE AS SYSDBA [8:58:29 PM] 75 rows selected in 0.058 seconds

שאילתה 5: (הסבר- החקירות האפידמיולוגיות ממוינות לפי מספר הנחשפים לחולה ולכן כעת קל יותר לשלוף את אלו שנחשפו אליהם יותר מ-30 אנשים).

```
select clientid,stayplace
from client C natural join (select cityname from city C where C.areaid = 9) natural join epidemiology
where exposednr > 30
ORDER BY clientid;
```

	CLIENTID	STAYPLACE
1	100399043	Wedding, Or Yehuda
2	102704705	Hospital, Rosh HaAyin
3	103413980	School, Holon
4	117531117	Central Station, Ashdod
5	121736560	Wedding, Shfar-am
6	121736560	Toy Store, Tirat Carmel
7	133836283	Wedding, Shderot
8	133836283	Wedding, Elad
9	145381088	Boardwalk, Petah Tikva
10	149302323	Super Market, Beit Shean
11	152871335	Toy Store, Or Yehuda
12	183108866	Synagogue, Tayibe
13	201052128	Mall, Umm al-Fahm
14	201052128	Park, Petah Tikva

4:19 sys@XE AS SYSDBA [8:54:23 PM] 62 rows selected in 0.055 seconds

שאילתה 6: (הסבר- מכונות ההנשמה ממוינות לפי השנה האחרונה בה נבדקו, ולכן כעת זמן השליפה של החולים שמחוברים למכונה שלא נבדקה ב-4 השנים האחרונות קצר יותר).

```
select clientid
from client natural join respirator
where client.yearofbirth < 1950 and extract(year from lastcheck)< '2016'
order by yearofbirth
```

	CLIENTID
1	504743251
2	304867344
3	512156402
4	569310320
5	596253805
6	183092238
7	468643489
8	496509050
9	429140338
10	296108197
11	127091841
12	512464379
13	225040532
14	155378647
15	452494355
16	516662620

4:21 sys@XE AS SYSDBA [9:13:56 PM] 63 rows selected in 0.051 seconds

מספר שאילתה	זמן ביצוע	זמן ביצוע עם אינדקס
1	0.075	0.058
5	0.081	0.055
6	0.060	0.051

Updates:

עדכון 1: כל הפגישות של המטופלים ששמם מתחיל באות A יעודכנו להיות עם האחות Wally Holiday שמספר הזהות שלה הוא 778881565.

לפני העדכון – שמות המטופלים והאחות איתה יש להם פגישה:

```
SELECT C.clientname, N.nursename
FROM (client C JOIN schedule S ON C.clientid = S.clientid)
     JOIN nurse N ON S.nurseid = N.Nurseid
WHERE C.clientname LIKE 'A%';
```

	CLIENTNAME	NURSENAME
1	Allison Loveless	Nanci Williamson
2	Angelina Peterson	Loreena Tisdale
3	Aimee Hurt	Jesse Ingram
4	Amy Gilley	Holly Paymer
5	Austin Dolenz	Gabriel Osborne
6	Albertina Moraz	Noah Guest
7	Amy MacIsaac	Rowan Frost
8	Allan Lang	Colin Shorter
9	Albertina Moraz	Teri McIntyre
10	Angie Wells	Alannah Craven
11	April Kershaw	Taryn McLean
12	Angelina Cochran	Kiefer Farris
13	Alice Kershaw	Ewan Stamp
14	Angie Wells	Patrick Parker
15	Allan Lang	Lena Warden
16	Amy MacIsaac	Alfie Sawa
17	Aaron Midler	Burt Hidalgo

אחרי העדכון:

```
UPDATE schedule S
SET nurseid = 778881565
WHERE clientid IN (SELECT clientid from client
                   WHERE clientname LIKE 'A%');

SELECT C.clientname, N.nursename
FROM (client C JOIN schedule S ON C.clientid = S.clientid)
     JOIN nurse N ON S.nurseid = N.Nurseid
WHERE C.clientname LIKE 'A%';
```

Update schedule | Select client

	CLIENTNAME	NURSENAME
1	Angelina Cochran	Wally Holiday
2	Alana Sarsgaard	Wally Holiday
3	Austin Dolenz	Wally Holiday
4	Allison Loveless	Wally Holiday
5	Angelina Peterson	Wally Holiday
6	Albertina Moraz	Wally Holiday
7	Albertina Moraz	Wally Holiday
8	Amy Gilley	Wally Holiday
9	Amy Gilley	Wally Holiday
10	Amy MacIsaac	Wally Holiday
11	Amy MacIsaac	Wally Holiday

אחרי שחזור באמצעות rollback:

```
ROLLBACK;

SELECT C.clientname, N.nursename
FROM (client C JOIN schedule S ON C.clientid = S.clientid)
     JOIN nurse N ON S.nurseid = N.Nurseid
WHERE C.clientname LIKE 'A%';
```

Rollback | Select client

	CLIENTNAME	NURSENAME
1	Allison Loveless	Nanci Williamson
2	Angelina Peterson	Loreena Tisdale
3	Aimee Hurt	Jesse Ingram
4	Amy Gilley	Holly Paymer
5	Austin Dolenz	Gabriel Osborne
6	Albertina Moraz	Noah Guest
7	Amy MacIsaac	Rowan Frost
8	Allan Lang	Colin Shorter
9	Albertina Moraz	Teri McIntyre
10	Angie Wells	Alannah Craven
11	April Kershaw	Taryn McLean
12	Angelina Cochran	Kiefer Farris
13	Alice Kershaw	Ewan Stamp

לאחר commit:

```
UPDATE schedule S
SET nurseid = 778881565
WHERE clientid IN (SELECT clientid from client
                   WHERE clientname LIKE 'A%');
commit;
rollback;
SELECT C.clientname, N.nursename
FROM (client C JOIN schedule S ON C.clientid = S.clientid)
     JOIN nurse N ON S.nurseid = N.Nurseid
WHERE C.clientname LIKE 'A%';
```

Update schedule | Commit | Rollback | Select client

	CLIENTNAME	NURSENAME
1	Angelina Cochran	Wally Holiday
2	Alana Sarsgaard	Wally Holiday
3	Austin Dolenz	Wally Holiday
4	Allison Loveless	Wally Holiday
5	Angelina Peterson	Wally Holiday
6	Albertina Moraz	Wally Holiday
7	Albertina Moraz	Wally Holiday
8	Amy Gilley	Wally Holiday
9	Amy Gilley	Wally Holiday

<u>עדכון 2:</u> כל החולים שמחוברים למכונת הנשמה והם בקבוצת סיכון, מצבם יהפוך לקשה.

כל החולים שמחוברים למכונת הנשמה:

```
select clientid, diseaseCondition
from client natural join respirator
where riskgroup = 1;
```

	CLIENTID	DISEASECONDITION
1	531958393	easy
2	311267194	hard
3	340076808	medium
4	348868753	easy
5	436220125	hard
6	527556515	easy
7	282781395	hard
8	460006703	hard
9	568468161	easy
10	531872360	hard
11	105870457	medium
12	563909409	hard
13	326590208	medium
14	385614150	easy
15	506111648	easy
16	138013346	easy
17	129386298	easy
18	510214007	hard

3:21 sys@XE AS SYSDBA

כל החולים שמחוברים למכונת הנשמה – אחרי העדכון

```
update client
set diseaseCondition='hard'
where riskgroup = 1 and clientid in (select clientid from respirator);

select clientid, diseaseCondition
from client natural join respirator
where riskgroup = 1;
```

Update client | Select client

	CLIENTID	DISEASECONDITION
1	531958393	hard
2	311267194	hard
3	340076808	hard
4	348868753	hard
5	436220125	hard
6	527556515	hard
7	282781395	hard
8	460006703	hard
9	568468161	hard
10	531872360	hard
11	105870457	hard
12	563909409	hard

SQL | Output | Statistics

```
rollback;
```

חזרנו למצב שהיינו לפני העדכון:

```
select clientid, diseaseCondition
from client natural join respirator
where riskgroup = 1;
```

	CLIENTID	DISEASECONDITION
1	531958393	easy
2	311267194	hard
3	340076808	medium
4	348868753	easy
5	436220125	hard
6	527556515	easy
7	282781395	hard
8	460006703	hard
9	568468161	easy
10	531872360	hard
11	105870457	medium
12	563909409	hard
13	326590208	medium
14	385614150	easy
15	506111648	easy
16	138013346	easy
17	129386298	easy
18	510314007	hard

כעת גם לאחר rollback העדכון נשאר.

```
update client
set diseaseCondition = 'hard'
where riskgroup = 1 and clientid in (select clientid from respirator);
commit;
rollback;
select clientid, diseaseCondition
from client natural join respirator
where riskgroup = 1;
```

Update client | Commit | Rollback | Select client

	CLIENTID	DISEASECONDITION
1	531958393	hard
2	311267194	hard
3	340076808	hard
4	348868753	hard
5	436220125	hard
6	527556515	hard
7	282781395	hard
8	460006703	hard
9	568468161	hard
10	531872360	hard
11	105870457	hard

Deletes:

מחיקה 1: מחיקת החקירות האפידמיולוגיות של המטופלים שגרים בצפת ומצבם קל.

לפני המחיקה:

```
SELECT clientid, stayplace, staytime
FROM epidemiology NATURAL JOIN client C
WHERE cityname = 'Safed' and diseasecondition = 'easy';
```

	CLIENTID	STAYPLACE	STAYTIME
1	240964888	Central Station, Kiryat Bialik	3/10/2020 6:23:11 AM
2	240964888	School, Eilat	2/27/2020 5:14:55 AM
3	203228910	Boardwalk, Holon	1/24/2020 3:29:31 PM

לאחר המחיקה:

```
DELETE
FROM epidemiology
WHERE clientid IN (SELECT clientid FROM client WHERE cityname = 'Safed' and diseasecondition = 'easy');

SELECT clientid, stayplace, staytime
FROM epidemiology NATURAL JOIN client C
WHERE cityname = 'Safed' and diseasecondition = 'easy';
```

Delete epidemiology | Select epidemiology

CLIENTID	STAYPLACE	STAYTIME

לאחר שחזור, הנתונים חוזרים לקדמותם:

```
ROLLBACK;

SELECT clientid, stayplace, staytime
FROM epidemiology NATURAL JOIN client C
WHERE cityname = 'Safed' and diseasecondition = 'easy';
```

Rollback | Select epidemiology

	CLIENTID	STAYPLACE	STAYTIME
1	240964888	Central Station, Kiryat Bialik	3/10/2020 6:23:11 AM
2	240964888	School, Eilat	2/27/2020 5:14:55 AM
3	203228910	Boardwalk, Holon	1/24/2020 3:29:31 PM

נראה שהנתונים לא ישוחזרו לאחר commit:

```
DELETE
FROM epidemiology
WHERE clientid IN (SELECT clientid FROM client WHERE cityname = 'Safed' and diseasecondition = 'easy');
commit;
ROLLBACK;
SELECT clientid, stayplace, staytime
FROM epidemiology NATURAL JOIN client C
WHERE cityname = 'Safed' and diseasecondition = 'easy';
```

Delete epidemiology | Commit | Rollback | Select epidemiology

CLIENTID	STAYPLACE	STAYTIME

<u>מחיקה 2:</u> מחיקת כל האנשים שפגישתם הייתה לפני ה- 1/5/2020

```
select clientid,meetingtime
from client C natural join schedule S
where extract(year from meetingtime)=2020 and extract(month from meetingtime)<05
```

	CLIENTID	MEETINGTIME
1	512295652	30/01/2020
2	397028200	23/04/2020

לאחר המחיקה:

```
delete
from schedule
where clientid in (select clientid from schedule
                   where extract(year from meetingtime)=2020 and extract(month from meetingtime)<05);
select clientid,meetingtime
from client C natural join schedule S
where extract(year from meetingtime)=2020 and extract(month from meetingtime)<05;
```

Roll | ct client

CLIENTID	MEETINGTIME

SQL | Output | Statistics

```
rollback;
```

חזרנו למצב לפני המחיקה:

```
select clientid,meetingtime
from client C natural join schedule S
where extract(year from meetingtime)=2020 and extract(month from meetingtime)<05
```

	CLIENTID	MEETINGTIME
1	512295652	30/01/2020
2	397028200	23/04/2020

לאחר commit גם לאחר rollback הפגישות נותרו מחוקות:

```
delete
from schedule
where clientid in (select clientid from schedule
                   where extract(year from meetingtime)=2020 and extract(month from meetingtime)<05);
commit;
rollback;

select clientid,meetingtime
from client C natural join schedule S
where extract(year from meetingtime)=2020 and extract(month from meetingtime)<05;
```

Rollback | Select client

CLIENTID	MEETINGTIME

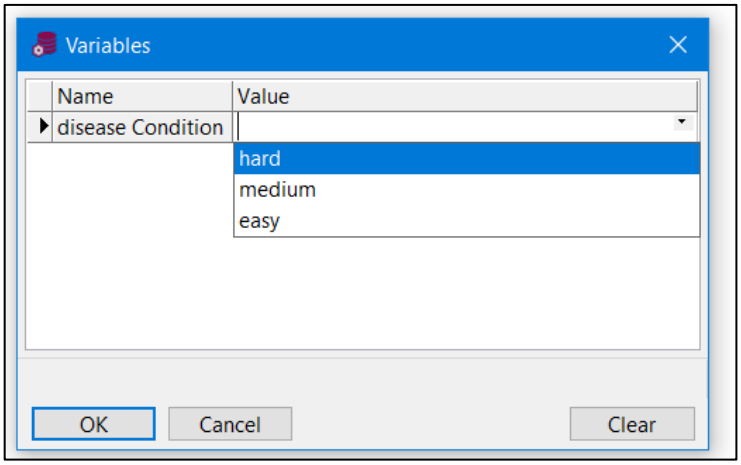
דוחות:

דוח 1: אנחנו רוצים לדעת מיהם החולים במצב קל בינוני וקשה ומאיזו עיר הם מגיעים

הרצנו את השאילתה הבאה:

```
select clientname,cityname
from client
where diseaseCondition= & <name="diseaseCondition"
                           type="string"
                           list="hard,easy,medium">
```

ונפתח לנו חלון שבו נכתוב את רמת החולי וקיבלנו רשימה של החולים הנמצאים באותה רמה:



וכשלחצנו/כתבנו hard לדוגמא קיבלנו את הטבלה הבאה:

disease Condition = hard

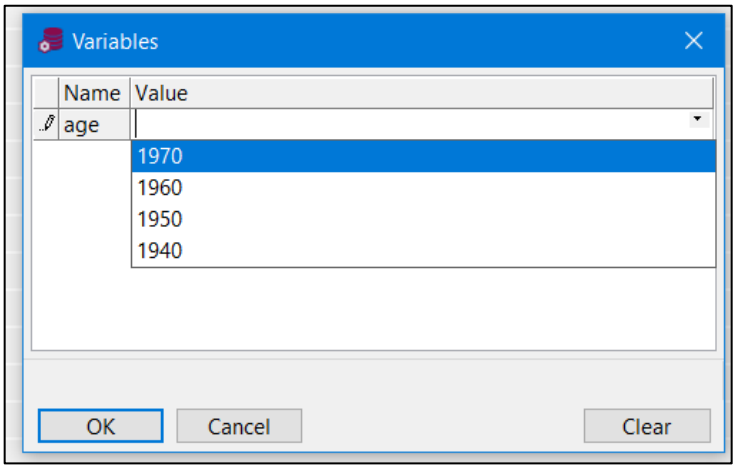
Clientname	Cityname
Keanu Caldwell	Ramat Gan
King Branch	Kiryat Bialik
Mindy Cash	Qalansawe
Ramsey Lee	Hod HaSharon
Debby Mason	Kiryat Yam
Billy Fichtner	Givat Shmuel
Rich Winans	Rehovot
Brooke Aniston	Bat Yam
Lea Brooks	Givatayim
Alfie Cocker	Even Yehuda
Dean Larter	Tamra
Mel Weiss	Kiryat Bialik
Maria Whitwam	Afula
Busta McNarland	Netivot
Edie McPherson	Maalot-Tarshiha
Peter Brolin	Tira

<u>דוח 2:</u> אנחנו רוצים לקבל את רשימת החולים שמעל גיל 50 או 60 או 70 או 80

הרצנו את השאילתה הבאה:

```
select clientname,(2020-yearofbirth)as age
from client
where yearofbirth < &<name="age" type="integer" list="1970,1960,1950,1940">
```

ונפתח לנו חלון שבו נכתוב את שנת הלידה הרצויה:



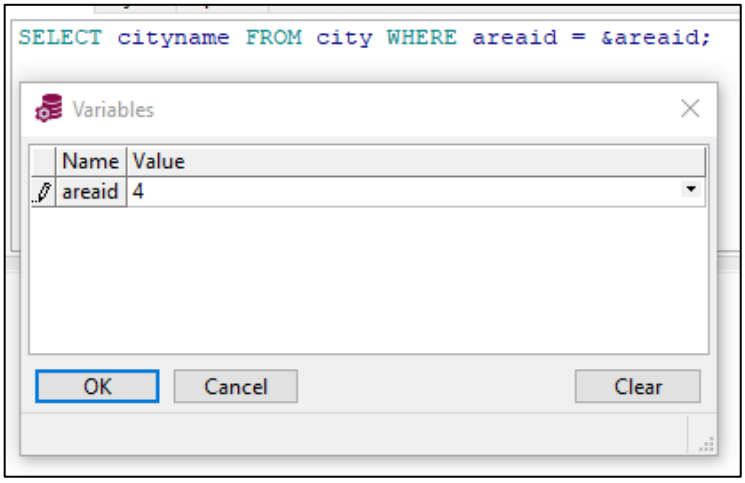
וכשלחצנו/כתבנו 1940 לדוגמא קיבלנו את הטבלה הבאה:

age = 1940

Clientname	Age
Debby Mason	89
Billy Fichtner	96
Faye Shandling	89
Elle Sepulveda	90
Trick Summer	84
Lauren White	85
Shelby Cazale	89
Ray Clayton	92
Rhett Nunn	86
Tia Spacek	89
Merrill Price	82
Bruce von Sydow	92
Brendan Ontiveros	87
Toni Quinones	92
Suzanne Kelly	84
Lance Copeland	83
Lindsay Potter	83
Merillee Craig	99
Kevin Doucette	81

<u>דו"ח 3:</u> נרצה לקבל פירוט של כל הערים שנמצאות באיזור מסוים, אך נוכל להחליט בכל פעם פירוט של אזור ספציפי.

לדוגמא נבחר אזור 4:



התקבלה הטבלה:

cities in area	
areaid = 4	
the names of cities	
Acre	
Afula	
Baqa-Jatt	
Beit Shean	
Hadera	
Haifa	
Karmiel	
Kiryat Ata	
Kiryat Bialik	
Kiryat Motzkin	
Kiryat Shmona	
Kiryat Yam	
Maalot-Tarshiha	
Migdal HaEmek	
Nahariya	
Nazareth	
Nazareth Illit	
Nesher	
Or Akiva	
Safed	
Sakhnin	
Shfar-am	
Tamra	
Tiberias	
Tirat Carmel	

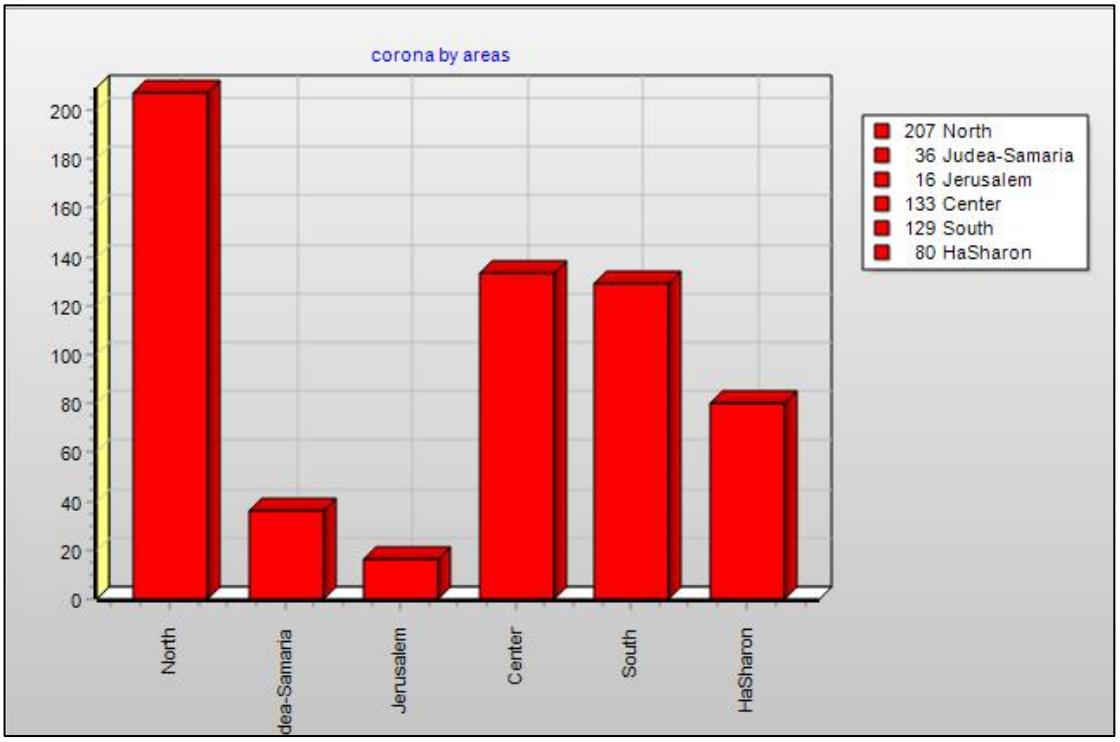
<u>דו"ח 4:</u> נרצה לקבל דו"ח המראה כמה חולים יש מכל עיר:

```
select areaname, COUNT(*)
from client NATURAL JOIN city NATURAL JOIN area
GROUP BY areaname;
```

corona by areas

name of area	number of patient from the area
North	207
Judea-Samaria	36
Jerusalem	16
Center	133
South	129
HaSharon	80

נמחיש את הנתונים באמצעות גרף:



שלב שלישי

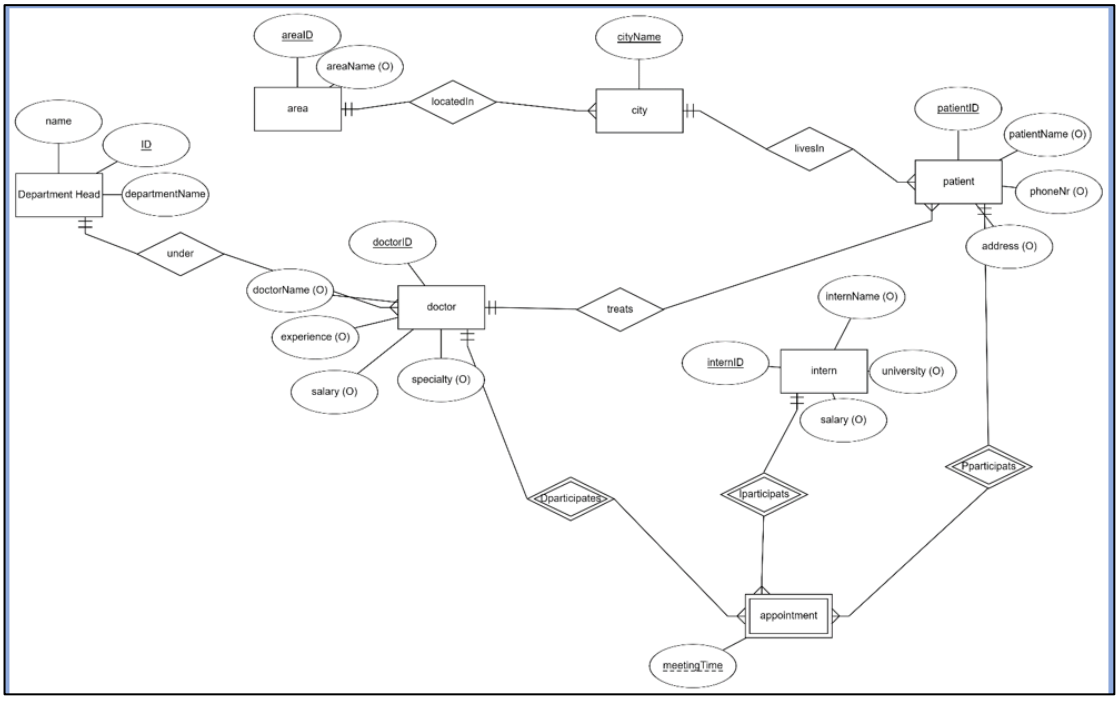
הקדמה:

בשלב זה נבצע אינטגרציה עם אגף מערך המח.

בשיעור נערך דיון באיזו דרך כדאי לבצע את האינטגרציה. אנחנו בחרנו להוסיף לבסיס הנתונים שלנו את הטבלאות שמופיעות בבסיס הנתונים של הזוג השני ואצלינו לא. כמו כן להוסיף באמצעות פקודות alter עמודות לטבלאות שכן קיימות וכן באמצעות פקודות אלו לשנות שמות של טבלאות קיימות, הכול בהתאם לשילוב שני האגפים.

תיאור אגף מערך המח:

תרשים ERD:



תיאור מילולי (לקוח מאגף מערך המח):

באגף מערך המח יש רופאים, מנהלי מחלקה, חולים ומתלמדים. החולים והרופאים מגיעים מכל הארץ. לכל פגישה יש חולה, רופא ומתלמד.
תיאור הישויות הקיימות במערכת:

- Appointment – ישות זו מיצגת פגישה בין חולה, רופא ומתלמד. היא ישות חלשה משום שהיא תלויה בכל המשתתפים.
  - DoctorID – ת"ז של הרופא.
  - ParientID – ת"ז של החולה.
  - MeetingTime – תאריך הטיפול.
  - InternID – ת"ז של המתלמד.
- Area – ישות זו מיצגת איזור בארץ. היא ישות חזקה מכיוון שהיא יכולה לעמוד בפני עצמה.
  - AreaID – מספר מזהה של האזור.
  - AreaName – שם האזור.
- City – ישות זו מייצגת עיר. היא ישות חזקה מכיוון שהיא יכולה לעמוד בפני עצמה.
  - CityName – שם העיר.
  - AreaID – מפתח זר לטבלת Area, מספר מזהה של האזור שבו העיר נמצאת.
- DepartmentHead – ישות זו מייצגת ראש מחלקה. היא ישות חזקה מכיוון שהיא יכולה לעמוד בפני עצמה.
  - ID – ת"ז של ראש המחלקה.
  - Name – שם הרופא.
  - DepartmentName – שם המחלקה עליה הרופא אחראי.
- Doctor – ישות זו מיצגת רופא שעובד באחת המחלקות. היא ישות חזקה מכיוון שהיא יכולה לעמוד בפני עצמה.
  - DoctorID (PK) – ת"ז של הרופא.
  - DoctorName – שם הרופא.
  - Experience – שנות נסיון.
  - Salary – משכורת.
  - Specialty – התמחות.
  - DepartmentHeadID – ת"ז של ראש המחלקה בה הרופא עובד.
- Intern – ישות זו מיצגת מתלמד שעושה סטאז' באגף מערך המח. היא ישות חזקה מכיוון שהיא יכולה לעמוד בפני עצמה.
  - InternID – ת"ז של המתלמד.
  - InternName – שם המתלמד.
  - Salary – משכורת.
  - University – האוניברסיטה בה הוא לומד.
- Patient – ישות זו מייצגת חולה באחד המחלקות. היא ישות חזקה מכיוון שהיא יכולה לעמוד בפני עצמה.
  - PatientID – ת"ז של החולה.
  - PatientName – שם החולה.
  - DoctorID – ת"ז של הרופא המטפל.
  - PhoneNUmber – מספר טלפון של החולה.
  - Address – כתובת מגורים.
  - CityName – שם העיר.

<u>האינטגרציה</u>

- הישויות הזהות הן city ו-area. לכן אין צורך להוסיפן.
- הישות patient היא הישות client בבסיס הנתונים שלנו. כמו כן, אצלינו יש לה שדות נוספים. לכן נשנה אצלינו מclient ל-patient, ונשאיר את השדות הנוספים שלנו. נשנה גם בטבלאות בהן יש reference את העמודות clientID לpatientID).

```
ALTER TABLE client RENAME TO patient;

ALTER TABLE patient
RENAME COLUMN clientid TO patientid;

ALTER TABLE patient
RENAME COLUMN clientname TO patientname;

ALTER TABLE schedule
RENAME COLUMN clientid TO patientid;

ALTER TABLE epidemiology
RENAME COLUMN clientid TO patientid;

ALTER TABLE respirator
RENAME COLUMN clientid TO patientid;

select* from patient;
```

Alter schedule | Alter epidemiology | Alter respirator | Select patient

	PATIENTID	PATIENTNAME	NURSEID	PHONENR	ADDRESS	CITYNAME	YEAROFBIRTH	RISKGROUP	DISEASECONDITION
1	471411815	Nathan Tyler	633139519	0554294827		Tirat Carmel	2009	0	medium
2	561709527	Wang Carrington	630697166	0532848860		Tirat Carmel	1981	0	medium
3	354000431	Giovanni Conway	748031390	0582027746		Hadera	2002	0	easy
4	311457592	Vienna Young	775230299	0504335479		Tel Mond	1934	0	medium

- נוסיף עוד שתי ישויות חדשות מאגף מערך המח:

ראש מחלקה – departmentHead, המכילה מספר זהות, שם, ושם המחלקה.

רופא מתמחה – intern, המכילה מספר זהות, שם, משכורת, ואוניברסיטה בה למד המתמחה.

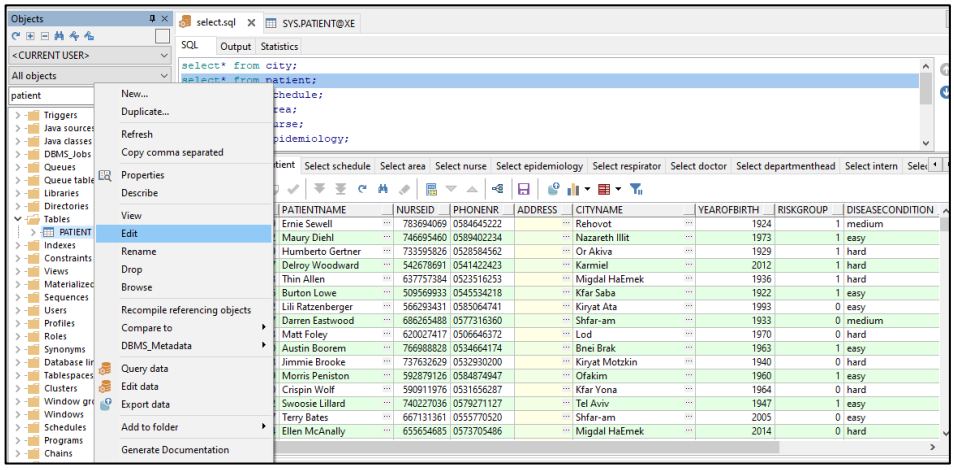
- הישות schedule (הבסיסית) מייצגת אצלם פגישה בין רופא, רופא מתמחה, ומטופל. אצלינו היא מייצגת פגישה בין אחות למטופל. לכן נוסיף ישות חדשה בשם appointment – פגישה של רופא, רופא מתמחה ומטופל.

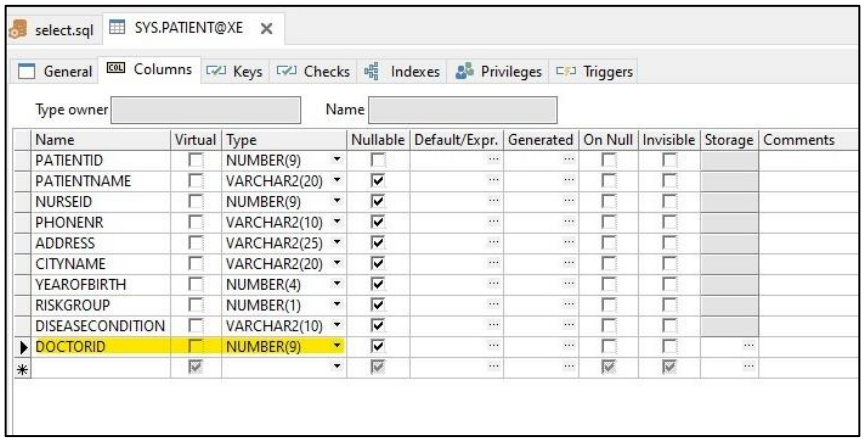
- הישות agent (הבסיסית) מתייחסת אצלם לרופא ואצלינו לאחות רפואית. ולכן נוסיף ישות חדשה בשם doctor בנוסף ל-nurse שקיימת אצלינו.

(סה"כ 4 טבלאות חדשות)

```
create table departmentHead(depHeadID   number(9) primary key,
                            depHeadName varchar(20),
                            Department  varchar(20));

create table doctor(doctorID         number(9) primary key,
                    doctorName       varchar(20),
                    experience       number(2),
                    specialty        varchar(20),
                    DepartmentHeadID number(9) references DepartmentHead,
                    salary           float);

create table intern(internID   number(9) primary key,
                    internName varchar(20),
                    university varchar(20),
                    salary     float);

create table appointment(doctorId number(9),
                         internId number(9),
                         patientId  number(9),
                         meetingTime date,
                         primary key (doctorId, internId, patientId, meetingTime),
                         foreign key (doctorId) references doctor,
                         foreign key (internId) references intern,
                         foreign key (patientId) references patient);
```

- נוסיף אצל client שדה של doctorId שהוא reference לטבלה doctor.





General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable	Deferred	Validated
SYS_C008081	Primary	PATIENTID	☑				☐	☐	☑
SYS_C008082	Foreign	NURSEID	☑	NURSE	NURSEID	No action	☐	☐	☑
SYS_C008083	Foreign	CITYNAME	☑	CITY	CITYNAME	No action	☐	☐	☑
SYS_C008137	Foreign	DOCTORID	☑	DOCTOR	DOCTORID	No action	☐	☐	☑
	Unique		☑				☐	☐	☑

הוספת נתונים לטבלאות החדשות:

הוספת נתונים באמצעות export לקבצי sql מבסיס הנתונים של אגף מערך המח

```
--departmentHead
insert into departmenthead (depHeadId, depHeadName, department)
values (559866704, 'Talya Broder', 'Neurosurgery');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866323, 'Yosef Genut', 'Neurosurgery');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866755, 'Eliau Cohen', 'Pediatric neurosurgery');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866425, 'Shifra Robin', 'Pediatric neurosurgery');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866221, 'Sara Broder', 'Neurology');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866667, 'Pnina Noi', 'Neurology');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866995, 'Malka Porush', 'Stroke');

insert into departmenthead (depHeadId, depHeadName, department)
values (559866780, 'Hadasa Frdedman', 'Stroke');

commit;
```

```
--doctor
insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (200550406, 'Shlomo Cohen', 6, 50800, 'Anesthesiology', 559866704);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (559895704, 'Shmuel  Ventura', 2, 19000, 'Dermatology', 559866704);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (523180780, 'Moshe Suissa', 2, 56800, 'radiology', 559866704);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (560453713, 'Moshe Cohen', 2, 21900, 'family medicine', 559866704);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (564501324, 'Yaacov Levy', 2, 42900, 'neurology', 559866704);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (569375478, 'Ofer Tahar', 2, 35700, 'Anesthesiology', 559866704);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (581621917, 'Gabriel Goldberg', 2, 15100, 'Anesthesiology', 559866667);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (597722737, 'Moshe Amihud', 3, 28200, 'Anesthesiology', 559866667);

insert into doctor (DOCTORID, DOCTORNAME, EXPERIENCE, SALARY, SPECIALTY, DEPARTMENTHEADID)
values (507512357, 'Yossef Gorodetski', 3, 35000, 'Anesthesiology', 559866667);
```

```
--intern
insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (478294894, 'Chedva Bro', 5000, 'jct');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (478294839, 'Rivki Olman', 6000, 'Oxford');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (478294029, 'Bina Aisner', 5000, 'Cambridge');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (783294730, 'Shin Lim', 6000, 'Oxford');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (89094730, 'Ellen Degeneres', 5000, 'jct');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (478394730, 'Justin Biber', 7000, 'Oxford');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (478292080, 'Michel Obama', 7000, 'mit');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (478250330, 'Donald Trump', 7000, 'Stanford');

insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (287294730, 'Harry Potter', 8000, 'Hogwarts');
```

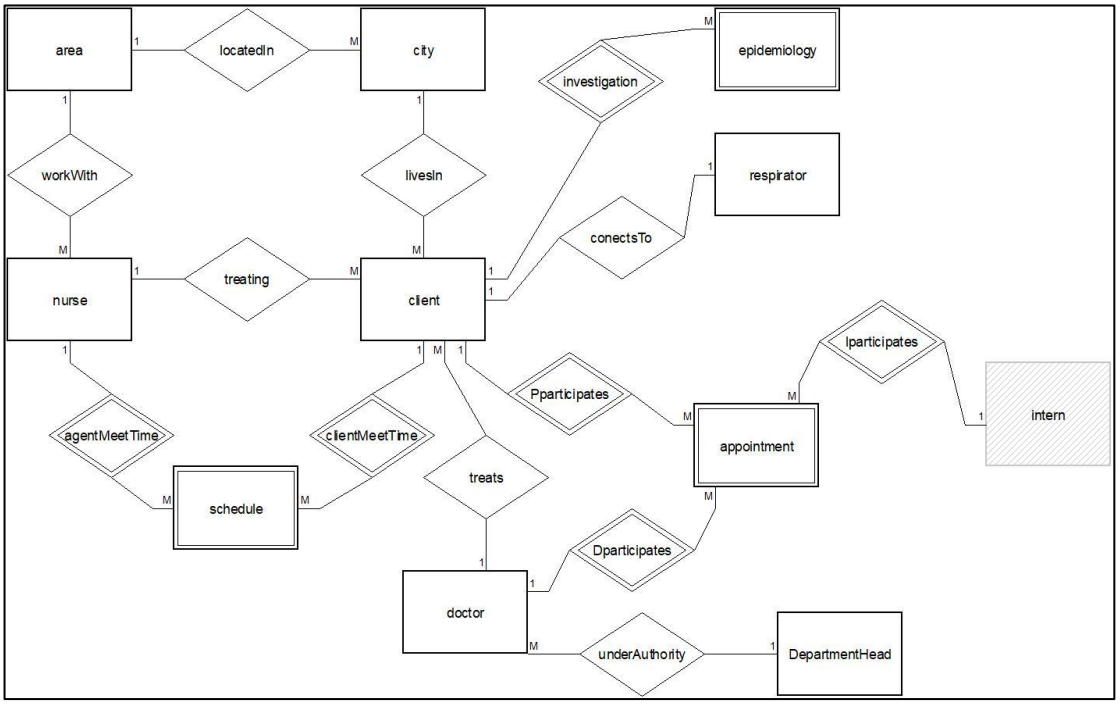
הוספת נתונים באמצעות data generator:

APPOINTMENT

Owner	Table	Number of records
SYS	APPOINTMENT	200

Name	Type	Size	Data
DOCTORID	NUMBER	9	list(select doctorid from doctor)
INTERNID	NUMBER	9	list(select internid from intern)
PATIENTID	NUMBER	9	list(select patientid from patient)
MEETINGTIME	DATE		random(01/01/2020, 29/07/2020)

תרשים ERD מעודכן



צורה נורמלית של הטבלאות החדשות:

טבלת departmentHead: depHeadId -> depHeadName, department

טבלת doctor: doctored -> doctorName, experience, specialty, departmentHeadId, salary

טבלת intern: internId -> internName, university, salary

טבלת appointment: doctorId, internId, patientId, meetingTime

טבלת patient: patientId -> patientName, nurseId, doctorId, phonNr, address, cityName, yearOfBirth, riskGroup, diseaseCondition

ניתן לראות שגם הטבלאות החדשות במנורמלות בצורת BCNF מכיוון שבכל תלות הגורר הוא מפתח על.

התאמת השאילתות לבסיס הנתונים המעודכן:

```
--query #1
--שמות המטופלים שנמצאים בקבוצת סיכון ומחוברים למכונת הנשמה אך גילם נמוך מ-60
SELECT patientname, (2020-yearofbirth) as age
FROM (SELECT patientid, patientname, yearofbirth
      FROM patient
      WHERE riskgroup = 1 and yearofbirth > 1960)
WHERE patientid in (select distinct patientid from respirator);

--query #2
--מספר המטופלים החולים קשה, בינוני וקל ממוין לפי מספר החולים
SELECT diseaseCondition, COUNT(diseaseCondition) as number_of_patients
FROM patient
GROUP BY diseaseCondition
ORDER BY diseaseCondition;

--query #3
--מספר הפגישות שיש לכל אחות עם מטופלים מתחת לגיל 18
SELECT nurseId, COUNT(*) as numberOfMeetings
FROM nurse NATURAL JOIN schedule NATURAL JOIN patient
GROUP BY nurseId, yearOfBirth
HAVING yearOfBirth > 2002;
```

```
--query #4
--ממוצע הימים שחולים מעל גיל 60 מחוברים למכונת הנשמה
select AVG(R.DAYSOFUSE)
from respirator R natural join patient C
where R.DAYSOFUSE IS NOT NULL and C.YEAROFBIRTH > 1960;

--query #5
--ת.ז. והמיקום שבו הסתובבו חולים שגרים באזור 9 (השרון) שנחשפו ליותר מ30 אנשים
select patientid,stayplace
from patient C natural join (select cityname from city C where C.areaid = 9) natural join epidemiology
where exposednr > 30
ORDER BY patientid;

--query #6
--ת.ז של חולים מעל גיל 70 שמכונת ההנשמה שלהם לא נבדקה ב4 שנים האחרונות
select patientid
from patient natural join respirator
where patient.yearofbirth < 1950 and extract(year from lastcheck)< '2016'
order by yearofbirth
```

שאילתות על בסיס הנתונים החדש:

שאילתה 1: כל המטופלים שיש להם בחודש מאי 2020 פגישה עם הרופא וגם פגישה עם האחות, ומספרי הזהות של המשתתפים בפגישות.

```
SELECT a.patientid, a.doctorid, a.internid, s.nurseid
FROM schedule s Join appointment a ON a.patientid = s.patientid
WHERE extract(year from a.meetingtime) = 2020 and extract(month from a.meetingtime) = 05
and extract(year from s.meetingtime) = 2020 and extract(month from s.meetingtime) = 05;
```

	PATIENTID	DOCTORID	INTERNID	NURSEID
1	401889548	784169339	627327614	627712905
2	185722612	281360066	627327716	693584509
3	422481246	543712835	627327744	627281234

שאילתה 2: שם ות.ז. של רופאים שהם ראש במחלקה pediatric neurosurgery ומטפלים בחולים מעל גיל 60 ויש להם יותר מ13 שנות ניסיון

```
select depheadid,depheadname
from patient NATURAL JOIN doctor NATURAL JOIN departmenthead
where yearofbirth>1960 AND experience>13 AND department='Pediatric neurosurgery'
order by depheadname;
```

	DEPHEADID	DEPHEADNAME
22	559866755	Eliau Cohen
23	559866755	Eliau Cohen
24	559866755	Eliau Cohen
25	559866755	Eliau Cohen
26	559866755	Eliau Cohen
27	559866425	Shifra Robin
28	559866425	Shifra Robin
29	559866425	Shifra Robin
30	559866425	Shifra Robin
31	559866425	Shifra Robin
32	559866425	Shifra Robin
33	559866425	Shifra Robin

15:1 sys@XE AS SYSDBA [21:34:54] 48 rows selected in 0.037 seconds

שאילתה 3: ממוצע השכר לראשי מחלקה עם ניסיון של יותר מ5 שנים ולא מטפלים בחולים שמצבם קל.

```
select AVG(salary) as salary_avarage
from doctor NATURAL JOIN departmenthead
where experience>5 and doctorid not in (select doctorid
                                        from patient
                                        where patient.diseasecondition = 'easy');
```

Select schedule | Select patient | Select doctor

	SALARY_AVARAGE
1	106869.558823529

Views:

view #1: ניצור view שיכיל את נתוני הרופאים וכן את נתוני ראש המחלקה שלהם:

```
CREATE or replace VIEW doctors_view AS
SELECT*
FROM doctor NATURAL JOIN departmenthead;
```

נראה את נתוני הview:

```
SELECT*
FROM doctors_view;
```

	DEPHEADID	DOCTORID	DOCTORNAME	EXPERIENCE	SPECIALTY	SALARY	DEPHEADNAME	DEPARTMENT
1	559866704	200550406	Shlomo Cohen	6	Anesthesiology	50800	Talya Broder	Neurosurgery
2	559866704	559895704	Shmuel Ventura	2	Dermatology	19000	Talya Broder	Neurosurgery
3	559866704	523180780	Moshe Suissa	2	radiology	56800	Talya Broder	Neurosurgery
4	559866704	560453713	Moshe Cohen	2	family medicine	21900	Talya Broder	Neurosurgery
5	559866704	564501324	Yaacov Levy	2	neurology	42900	Talya Broder	Neurosurgery
6	559866704	569375478	Ofer Tahar	2	Anesthesiology	35700	Talya Broder	Neurosurgery
7	559866667	581621917	Gabriel Goldberg	2	Anesthesiology	15100	Pnina Noi	Neurology
8	559866667	597722737	Moshe Amihud	3	Anesthesiology	28200	Pnina Noi	Neurology
9	559866667	507512357	Yossef Gorodetski	3	Anesthesiology	35000	Pnina Noi	Neurology
10	559866667	602517504	Shmuel Bardaat	3	Dermatology	32700	Pnina Noi	Neurology
11	559866667	609068105	Yehuda Azoulay	3	Dermatology	53500	Pnina Noi	Neurology
12	559866667	609002211	Yoram Amichai	3	Dermatology	59500	Pnina Noi	Neurology
13	559866995	855988674	Assaf Zalmanovitch	3	Dermatology	50850	Malka Porush	Stroke
14	559866995	653793789	Shimon Goldmann	3	Dermatology	33570	Malka Porush	Stroke

Update על ה-view:

כעת נעלה את המשכורת של כל הרופאים שראש המחלקה שלהם היא פנינה נוי ב-5000 ₪:

```
UPDATE doctors_view
SET salary = salary + 5000
WHERE depheadname = 'Pnina Noi';

SELECT*
FROM doctors_view;
```

	DEPHEADID	DOCTORID	DOCTORNAME	EXPERIENCE	SPECIALTY	SALARY	DEPHEADNAME	DEPARTMENT
1	559866704	200550406	Shlomo Cohen	6	Anesthesiology	50800	Talya Broder	Neurosurgery
2	559866704	559895704	Shmuel Ventura	2	Dermatology	19000	Talya Broder	Neurosurgery
3	559866704	523180780	Moshe Suissa	2	radiology	56800	Talya Broder	Neurosurgery
4	559866704	560453713	Moshe Cohen	2	family medicine	21900	Talya Broder	Neurosurgery
5	559866704	564501324	Yaacov Levy	2	neurology	42900	Talya Broder	Neurosurgery
6	559866704	569375478	Ofer Tahar	2	Anesthesiology	35700	Talya Broder	Neurosurgery
7	559866667	581621917	Gabriel Goldberg	2	Anesthesiology	20100	Pnina Noi	Neurology
8	559866667	597722737	Moshe Amihud	3	Anesthesiology	33200	Pnina Noi	Neurology
9	559866667	507512357	Yossef Gorodetski	3	Anesthesiology	40000	Pnina Noi	Neurology
10	559866667	602517504	Shmuel Bardaat	3	Dermatology	37700	Pnina Noi	Neurology
11	559866667	609068105	Yehuda Azoulay	3	Dermatology	58500	Pnina Noi	Neurology
12	559866667	609002211	Yoram Amichai	3	Dermatology	64500	Pnina Noi	Neurology
13	559866995	855988674	Assaf Zalmanovitch	3	Dermatology	50850	Malka Porush	Stroke
14	559866995	653793789	Shimon Goldmann	3	Dermatology	33570	Malka Porush	Stroke
15	559866995	624700530	Gabriel Amichai	4	radiology	11790	Malka Porush	Stroke

ניתן לראות שהמשכורת אכן עלתה.

שאילתה על ה-view:

נבדוק כמה רופאים יש בכל מחלקה:

```
SELECT department, count(*) as doctors
FROM doctors_view
GROUP BY department;
```

	DEPARTMENT	DOCTORS
1	Neurosurgery	68
2	Pediatric neurosurgery	56
3	Stroke	59
4	Neurology	58

view #2: ניצור view שיכיל את כל נתונים האחיות והפגישות שלהן

```
create or replace view nurse_schedule  as
select *
from schedule NATURAL JOIN nurse;
```

נראה את נתוני הview

```
select *
from nurse_schedule;
```

Create nurse_schedule | Select nurse_schedule

	NURSEID	PATIENTID	MEETINGTIME	NURSENAME	AREAID	RATING	HIREYEAR	BOSSID	SALARY
288	706958496	421621259	20/05/2020	Freddie Duchovny	8	6	2001	835790027	6903
147	610673529	257834928	20/05/2020	Dom Streep	2	6	1963	818015322	10121
49	538205806	227173579	20/05/2020	Don Neeson	2	7	1974	921895926	4546
117	585537131	457364967	20/05/2020	Thomas Hedaya	8	10	1964	919426047	4387
228	667983031	589774316	21/05/2020	Chi Vega	4	8	1973	984870760	5149
134	599387577	177737260	22/05/2020	Morgan Cale	5	1	1993	991077388	12707
155	620493650	292262873	24/05/2020	Daniel Frampton	9	8	1986	883574670	12448
245	673698654	386891687	27/05/2020	Bo Heston	4	9	1985	875223474	13828
18	518229611	258944588	29/05/2020	Gwyneth Graham	8	10	1961	878466786	13450
124	595804182	126301545	29/05/2020	Ricky Stiller	4	3	1981	998079387	10357
257	685919215	126301545	29/05/2020	Larnelle Feliciano	5	1	1991	847268826	6617
81	563103914	508128794	29/05/2020	Renee Minogue	9	1	1984	858362288	6310
328	742480233	182219619	30/05/2020	Avenged Gooding	2	9	1976	901066313	4879
125	595804182	181436865	30/05/2020	Ricky Stiller	4	3	1981	998079387	10357
360	774905470	466895822	30/05/2020	Natacha Brothers	8	5	1965	837031425	4221
8	510783277	351396161	30/05/2020	Lionel Kutcher	2	7	1991	902065950	3797
243	671712243	350665002	30/05/2020	Laurence Margulies	3	3	1992	969285595	14351
192	643811831	188383953	30/05/2020	Sona Orton	5	9	1987	942341421	5994
6	508963457	152411721	31/05/2020	Jerry Prinze	5	4	1987	810052382	7295

Update על ה-view:

נעביר את כל הפגישות שמתקיימות בתאריך 29/5/2020 לתאריך 31/5/2020

```
update nurse_schedule
set meetingtime=to_date('31-05-2020', 'dd-mm-yyyy')
where meetingtime=to_date('29-05-2020', 'dd-mm-yyyy');

select*
from nurse_schedule;
```

Create nurse_schedule | Select nurse_schedule | Update nurse_schedule | Select nurse_schedule

	NURSEID	PATIENTID	MEETINGTIME	NURSENAME	AREAID	RATING	HIREYEAR	BOSSID	SALARY
155	620493650	292262873	24/05/2020	Daniel Frampton	9	8	1986	883574670	12448
245	673698654	386891687	27/05/2020	Bo Heston	4	9	1985	875223474	13828
328	742480233	182219619	30/05/2020	Avenged Gooding	2	9	1976	901066313	4879
125	595804182	181436865	30/05/2020	Ricky Stiller	4	3	1981	998079387	10357
243	671712243	350665002	30/05/2020	Laurence Margulies	3	3	1992	969285595	14351
8	510783277	351396161	30/05/2020	Lionel Kutcher	2	7	1991	902065950	3797
192	643811831	188383953	30/05/2020	Sona Orton	5	9	1987	942341421	5994
360	774905470	466895822	30/05/2020	Natacha Brothers	8	5	1965	837031425	4221
124	595804182	126301545	31/05/2020	Ricky Stiller	4	3	1981	998079387	10357
6	508963457	152411721	31/05/2020	Jerry Prinze	5	4	1987	810052382	7295
320	727944901	417968065	31/05/2020	Orlando Rundgren	8	0	1977	880094441	5626
257	685919215	126301545	31/05/2020	Larnelle Feliciano	5	1	1991	847268826	6617
340	763665697	228156595	31/05/2020	Ellen Ifans	9	9	1981	846143748	4283
81	563103914	508128794	31/05/2020	Renee Minogue	9	1	1984	858362288	6310
18	518229611	258944588	31/05/2020	Gwyneth Graham	8	10	1961	878466786	13450
281	701932345	397471055	01/06/2020	Donal Goldwyn	2	6	1970	865076021	11346

ניתן לראות שהתאריך השתנה.

שאילתה על ה-view:

ממוצע השכר של אחיות שמקבלות מעל 10000 שח

```
select AVG(salary) as avg_salary
from nurse_schedule
where salary>10000;
```

Create nurse_schedule | Select nurse_schedule | Up

	AVG_SALARY
1	12673.8953488372

שלב רביעי

פונקציות:

פונקציה 1:

הפונקציה מקבלת ת.ז. של מטופל ומחזירה את מספר האנשים שנחשפו אליו סה"כ בזמן שהיה חולה, לפי החקירות האפידמיולוגיות.

```
--הפונקציה מקבלת מספר זהות של מטופל ומחזירה כמה אנשים נחשפו אליו בסה"כ
create or replace function how_many_exposed(idpatient in patient.patientid%type) return number is
  FunctionResult number(3);
  cursor epidemiology_cur is select epidemiology.exposednr
                             from epidemiology
                             where epidemiology.patientid = idpatient;
  exposed number(3) := 0;
begin
  for ep in epidemiology_cur
   loop
     exposed := exposed + ep.exposednr;
   end loop;
   dbms_output.put_line('the total exposed of: ' || idpatient || ' is: ' ||exposed);
   FunctionResult := exposed;
  return(FunctionResult);
end how_many_exposed;
```

נראה שהפונקציה עובדת:

נראה את החקירות האפידמיולוגיות של מטופל 366237484:

```
select *
from epidemiology
where epidemiology.patientid = 366237484;
```

	PATIENTID	STAYPLACE	STAYTIME	EXPOSEDNR
1	366237484	Hotel, Kiryat Ata	3/10/2020 6:27:12 PM	25
2	366237484	Hospital, Ofakim	2/14/2020 3:02:44 PM	28
3	366237484	Central Station, Kiryat Gat	3/17/2020 9:47:39 PM	33
4	366237484	Hotel, Ramat Gan	1/30/2020 6:25:46 AM	15

ניתן לראות ששהה ב-4 מקומות ובכל פעם נחשפו אליו: וסה"כ נחשפו אליו: 25+28+33+15 = 101

נריץ את הפונקציה על מטופל שלא קיימת עבורו חקירה אפידמיולוגית:

```
select *
from epidemiology
where patientid = 456864356;
```

PATIENTID	STAYPLACE	STAYTIME	EXPOSEDNR

הראינו שלא קיימות חקירות.

Test script | DBMS Output | Statistics | Profiler | Trace

```
1  begin
2    -- Call the function
3    :result := how_many_exposed(idpatient => :idpatient);
4  end;
```

Variable	Type	Value
result	Float	0
idpatient	Integer	456864356

התוצאה היא כמובן 0.

<u>פונקציה 2:</u>

הפונקציה מקבלת ת.ז. של מטופל ומחזירה אמת אם הוא מחובר למכונת הנשמה ושקר אם לא.

```
create or replace noneditionable function is_connected_to_respirator
(user_patient in patient.patientid%type) return boolean is
  FunctionResult boolean;
  cursor patients_cur is select respirator.patientid from respirator;
  --patient_id patient.patientid%type;
  is_connected boolean := false;
begin
  for pat in patients_cur
    loop
      if (pat.patientid = user_patient) then
        is_connected := true;
      end if;
    end loop;
  FunctionResult := is_connected;
  return(FunctionResult);
end is_connected_to_respirator;
```

נבדוק עבור מטופל שמחובר למכונת הנשמה -

קודם נוודא שהמטופל מחובר למכונה:

```
select respirator.patientid
  from respirator
  where respirator.patientid = 116996002;
```

Select respirator | Select respirator

	PATIENTID
1	116996002

וכעת נפעיל את הפונקציה עליו:

Test script | DBMS Output | Statistics | Profiler | Trace

```
declare
  -- Boolean parameters are translated from/to integers:
  -- 0/1/null <--> false/true/null
  result boolean;
begin
  -- Call the function
  result := is_connected_to_respirator(user_patient => :user_patient);
  -- Convert false/true/null to 0/1/null
  :result := sys.diutil.bool_to_int(result);
end;
```

Variable	Type	Value
result	Integer	1
user_patient	Integer	116996002

התוצאה היא 1 – אמת.

כעת נפעיל את הפונקציה על מטופל שלא מחובר למכונת הנשמה –

נוודא שאינו מחובר למכונה:

```
select respirator.patientid
  from respirator
  where respirator.patientid = 506945331;
```

Select respirator | Select respirator

PATIENTID

וכעת נפעיל עליו את הפונקציה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
declare
  -- Boolean parameters are translated from/to integers:
  -- 0/1/null <--> false/true/null
  result boolean;
begin
  -- Call the function
  result := is_connected_to_respirator(user_patient => :user_patient);
  -- Convert false/true/null to 0/1/null
  :result := sys.diutil.bool_to_int(result);
end;
```

Variable	Type	Value
result	Integer	0
user_patient	Integer	506945331

התוצאה היא 0 – שקר.

<u>פונקציה 3:</u>

הפונקציה מקבלת מקום וזמן. אם היה שם חולה קורונה מאומת – הפונקציה תחזיר את מספר הזהות שלו. אם לא היה שם חולה- הפונקציה תחזיר null.

```
create or replace function who_is_the_verified_patient(time_expose in epidemiology.staytime%type,
place_expose in epidemiology.stayplace%type) return number is
  FunctionResult number(9);
  cursor p_cur is select e.patientid
                  from epidemiology e
                  where e.stayplace = place_expose and e.staytime = time_expose;
  idpatient patient.patientid%type;
begin
  for p in p_cur
    loop
      if p_cur%found then
        dbms_output.put_line('You are exposed to the patient: ' || p.patientid);
        idpatient := p.patientid;
        exit when p_cur%found;
      end if;
    end loop;
  FunctionResult := idpatient;
  return(FunctionResult);
end who_is_the_verified_patient;
```

נראה את ריצת הפונקציה. נחפש את החקירות האפידמיולוגיות:

```
select* from epidemiology;
```

Select city | Select patient | Select schedule | Select area | Select nurse | Select epidemiology | Select respira

	PATIENTID	STAYPLACE	STAYTIME	EXPOSEDNR
1	295895094	Boardwalk, Jerusalem	3/31/2020 1:40:48 AM	11
2	468740795	Central Station, Shfar-am	3/14/2020 6:56:52 PM	33
3	366237484	Hotel, Kiryat Ata	3/10/2020 6:27:12 PM	25
4	236113201	University, Yavne	2/7/2020 6:47:48 PM	10
5	141080630	Boardwalk, Nesher	3/16/2020 1:08:06 PM	9
6	323603105	School, Ashkelon	4/14/2020 5:15:38 PM	33
7	166985646	Synagogue, Maalot-Tarshiha	3/8/2020 1:50:44 AM	7
8	442463339	Wedding, Yokneam	3/1/2020 4:45:15 PM	18
9	371735893	Wedding, Beit Shemesh	4/5/2020 12:21:18 AM	4
10	169396726	Wedding, Netivot	3/17/2020 10:52:27 PM	25
11	288593932	Super Market, Qalansawe	2/18/2020 3:01:17 AM	35
12	556593378	Mall, Ness Tziona	4/26/2020 3:36:43 AM	8
13	320830486	Toy Store, Umm al-Fahm	3/26/2020 11:28:10 PM	47
14	487904032	University, Kiryat Motzkin	2/27/2020 9:40:53 PM	8
15	464652162	Super Market, Bnei Brak	2/12/2020 11:32:55 AM	42
16	382434951	Hotel, Jerusalem	1/28/2020 12:27:04 PM	49
17	179107698	Toy Store, Hertzliya	2/28/2020 7:39:21 PM	23
18	148299370	Park, Nazareth	3/21/2020 6:27:38 AM	31
19	451471878	Boardwalk, Beitar Illit	3/28/2020 5:39:41 AM	48
20	336660622	University, Nahariya	4/20/2020 9:20:52 PM	28
21	577772430	Hospital, Umm al-Fahm	2/18/2020 5:28:19 AM	28
22	250141012	Central Station, Kiryat Yam	2/4/2020 11:00:14 PM	41

נריץ את הפונקציה על החקירה האפידמיולוגית הראשונה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
begin
  -- Call the function
  :result := who_is_the_verified_patient(time_expose => :time_expose,
                                         place_expose => :place_expose);
end;
```

Variable	Type	Value
result	Float	295895094
time_expose	Date	3/31/2020 1:40:48 AM
place_expose	String	Boardwalk, Jerusalem

ניתן לראות שהפונקציה החזירה את מספר הזהות של החולה המתאים.

נריץ מקום וזמן בו לא היה חולה קורונה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
1  begin
2    -- Call the function
3    :result := who_is_the_verified_patient(time_expose => :time_expose,
4                                           place_expose => :place_expose);
5  end;
```

Variable	Type	Value
result	Float	
time_expose	Date	1/1/2020 1:40:48 AM
place_expose	String	Kotel, Jerusalem

ניתן לראות שהפונקציה החזירה null.

<u>פונקציה 4:</u>

הפונקציה מקבלת מספר ת.ז. של רופא ומחזירה את מספר המטופלים שלו

```
create or replace function there_is_patient(cur_doctorid in patient.doctorid%type)
   return number is numPatient number(3);
   cursor his_patient is select patientid
                         from patient
                         where patient.doctorid=cur_doctorid;
   sum_patient number(3):=0;
  begin
    for idx in his_patient LOOP
      sum_patient:=his_patient%rowcount;
    END LOOP;
     dbms_output.put_line('sum of patient for '||cur_doctorid||' is '||sum_patient);
     return(sum_patient);

end there_is_patient;
```

נראה את טבלת החולים עם הת.ז של הרופאים שלהם

	PATIENTID	PATIENTNAME	NURSEID	DOCTORID	PHONENR	ADDRESS	CITYNAME	YEAROFBIRTH
227	447038968	Gavin Statham		100143662	0547418487		Ramat HaSharon	1958
231	282415670	Kim Gilliam		100866411	0571288706		Modiin-Maccabim-Reut	1936
369	506622656	Demi Caldwell		100866411	0535514110		Afula	1940
321	523515731	Armand Leary		100866411	0579500223		Kiryat Yam	1996
371	126301545	Geraldine Moorer		101893947	0576559478		Kiryat Shmona	1987
54	316596473	Miranda Driver		101893947	0536445499		Or Yehuda	1975
347	374221251	Caroline Balaban		101893947	0572392735		Nazareth	1982
337	440220974	Sandra Garofalo		101893947	0553226378		Kiryat Ata	1979
50	552024350	Rip Broza		101893947	0589703232		Tayibe	1964
165	375885557	Murray Rippy		107223017	0589991268		Sakhnin	2010
212	210775624	Lonnie Randal		108805248	0505655727		Or Yehuda	2007
173	475961129	Mika Ammons		108805248	0586826193		Gan Yavne	1946
35	186190478	Mili Byrne		109902179	0555358254		Efrath	1955
235	210326268	Arturo Cheadle		109902179	0582166866		Kafr Qasim	1935
353	469385833	Walter Hauser		109902179	0583671962		Eilat	1962
275	285447715	Danny Heatherly		113232505	0529486431		Ofakim	1944
220	537028591	Gary Henriksen		113232505	0552591314		Kiryat Shmona	1935
520	217166595	Jean-Claude Kane		113591151	0558748491		Netivot	2003
541	349534457	Micky Craig		113591151	0575311465		Bat Yam	1935
504	427022396	Robby Bailey		113591151	0544330205		Eilat	1944
349	213913442	Lauren Leary		119627004	0526095763		Netivot	1961
508	360680863	Mary Beth Hewitt		119627004	0582190153		Hertzliya	1973
109	398534116	Hex Collins		119627004	0574555162		Givatayim	1955
76	589886158	Latin Tah		119627004	0506261543		Bnei Brak	1963
223	284872405	Jon Diddley		121492247	0584631191		Or Akiva	1929

נריץ את הפונקציה על רופא שיש לו מטופלים:

Test script | DBMS Output | Statistics | Profiler | Trace

```
declare
  -- Local variables here
  result number;
begin
  -- Test statements here
  :result:=there_is_patient(cur_doctorid => :cur_doctorid);
end;
```

Variable	Type	Value
cur_doctorid	String	101893947
result	String	5

אפשר לראות שאכן הפונקציה החזירה שיש לאותו רופא 5 מטופלים חולי קורונה.

נריץ את הפונקציה על ת.ז של רופא שאין לו מטופלים חולי קורונה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
declare
  -- Local variables here
  result number;
begin
  -- Test statements here
  :result:=there_is_patient(cur_doctorid => :cur_doctorid);
end;
```

Variable	Type	Value
cur_doctorid	String	123456789
result	String	0

אפשר לראות שקיבלנו 0

פרוצדורות:

פרוצדורה 1:

הפרוצדורה מקבלת ת.ז של מטופל, מחפשת את מכונת ההנשמה הראשונה שפנויה ומחברת את המטופל למכונה הראשונה שפנויה. כלומר – מחפשת את המכונה שמספרה מינימלי ומעדכנת את השדה patientid לת.ז של המטופל ואת ימי השימוש במכונה מעדכנת ל-1.

```
create or replace noneditionable procedure connect_to_respiratoe
(idpatient in patient.patientid%type) is

idrespirator respirator.respiratorid%type;
begin
  --Search for available respirator
  select min(respirator.respiratorid) into idrespirator --min respiratorid
  from respirator
  where respirator.patientid is null;
  update respirator
  set respirator.patientid = idpatient, respirator.daysofuse = 1 --connect
  where respirator.respiratorid = idrespirator;
  dbms_output.put_line('patient ' || idpatient ||' connectes to respirator # ' || idrespirator);

  exception
  when NO_DATA_FOUND then
    dbms_output.put_line ('there is no available respirator');

  when DUP_VAL_ON_INDEX then
    dbms_output.put_line ('this patient is allready connected');

end connect_to_respiratoe;
```

נראה את ריצת הפרוצדורה, ראשית נבדוק את מצב מכונות ההנשמה:

SQL | Output | Statistics

```
select* from respirator;
```

	RESPIRATORID	PATIENTID	DAYSOFUSE	LASTCHECK
255	330	371735893	33	7/23/2016
256	333	423260596	22	8/28/2012
257	334	193504196	23	1/1/2001
258	335	284623217	29	9/19/2006
259	338	313682147	31	9/22/2002
260	339	322184812	9	11/11/2012
261	343	543695673	9	6/10/2004
262	345	487904032	21	7/10/2005
263	346	171755846	22	4/20/2011
264	348	336996837	32	5/3/2007
265	349	558590237	15	3/15/2004
266	350	216834410		7/22/2001
267	351	242363728	1	7/30/2018
268	352	583400156	1	8/3/2014
269	353	112417975	1	7/5/2000
270	354			7/6/2004
271	355			11/17/2006
272	356			3/11/2015
273	357			7/10/2007

מכונה מספר 354 פנויה

כעת נפעיל את הפרוצדורה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
1  begin
2    -- Call the procedure
3    connect_to_respiratoe(idpatient => :idpatient);
4  end;
```

Variable	Type	Value
idpatient	Integer	113076035

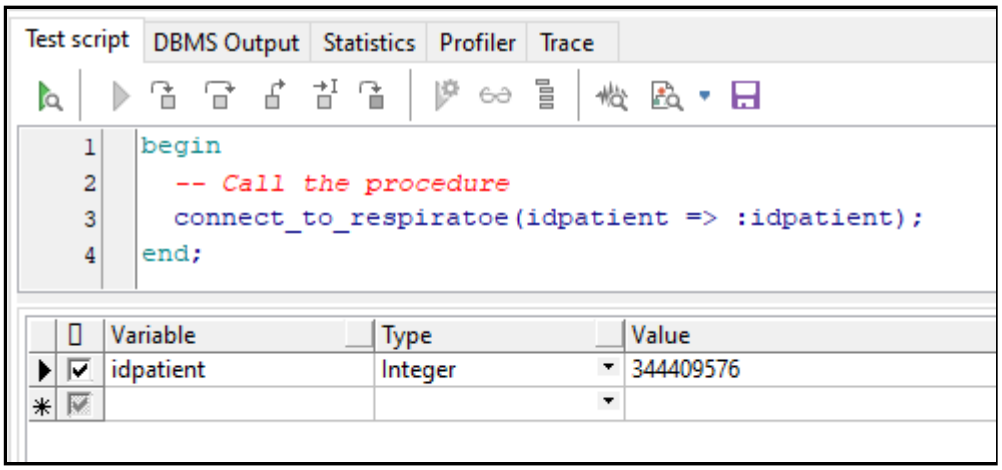
ונפעיל שוב את השאילתה:

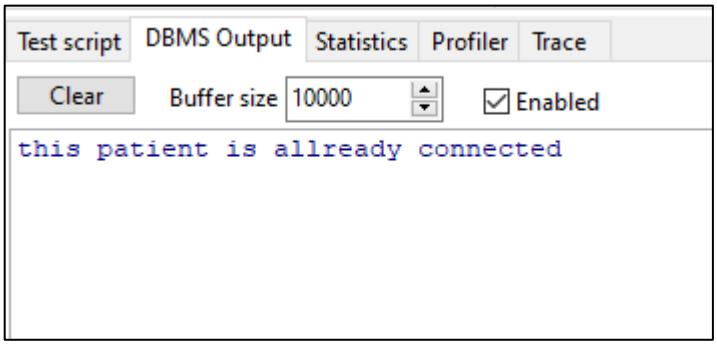
```
select* from respirator;
```

	RESPIRATORID	PATIENTID	DAYSOFUSE	LASTCHECK
254	327	178196169	37	6/5/2008
255	330	371735893	33	7/23/2016
256	333	423260596	22	8/28/2012
257	334	193504196	23	1/1/2001
258	335	284623217	29	9/19/2006
259	338	313682147	31	9/22/2002
260	339	322184812	9	11/11/2012
261	343	543695673	9	6/10/2004
262	345	487904032	21	7/10/2005
263	346	171755846	22	4/20/2011
264	348	336996837	32	5/3/2007
265	349	558590237	15	3/15/2004
266	350	216834410		7/22/2001
267	351	242363728	1	7/30/2018
268	352	583400156	1	8/3/2014
269	353	112417975	1	7/5/2000
270	354	113076035	1	7/6/2004
271	355			11/17/2006
272	356			3/11/2015
273	357			7/10/2007

המטופל מחובר למכונה ומספר ימי השימוש הוא 1.

הפרוצדורה מכילה exceptions עבור מקרה שכל מכונות ההנשמה תפוסות. וכן עבור מקרה שהמטופל כבר מחובר למכונת הנשמה. נפעיל את הפרוצדורה על מטופל שכבר מחובר למכונת





<u>פרוצדורה 2:</u>

פרוצדורה שמקבלת שם של חולה ושם של אחות ובודקת אם לחולה אין פגישה עם אחות היא קובעת למטופל פגישה עם האחות שהתקבלה בתאריך 29/7/2020

```
create or replace procedure schedule_meeting(idpatient in patient.patientid%type, idnurse in nurse.nurseid%type)is
meet number(3):=0;
begin
     select count(meetingtime) into meet
     from schedule
     where schedule.patientid=idpatient;

     if meet=0 then
       dbms_output.put_line('Patient '||idpatient||' dose not have meeting');
        insert into schedule(patientid,nurseid,meetingtime)
        values (idpatient,idnurse,to_date('29-07-2020', 'dd-mm-yyyy'));
        commit;
     else
       dbms_output.put_line('Patient '||idpatient||' already has meeting');
    end if;

end schedule_meeting;
```

נראה את מצב המערכת לפני הקביעה:

SQL | Output | Statistics

```
select* from schedule;
```

Select patient | Select schedule | Select area | Select nurse | Sele

	NURSEID	PATIENTID	MEETINGTIME
1	500417878	479425698	30/05/2020
2	500417878	499874252	25/03/2020
3	501152661	288593932	17/05/2020
4	503563632	157137258	16/04/2020
5	503563632	180629307	08/02/2020
6	503563632	189458552	20/01/2020
7	508705560	311054470	13/05/2020
8	508705560	584592729	17/06/2020
9	513774513	224994634	19/05/2020
10	515354407	112417975	05/06/2020
11	515354407	320830486	11/06/2020
12	516434238	556869539	23/02/2020
13	518785197	185073041	23/07/2020
14	518881254	503137815	08/05/2020
15	519528217	188656382	08/05/2020
16	519528217	309973460	10/05/2020
17	519528217	446493518	17/01/2020
18	519840042	342368796	05/02/2020

כעת נפעיל את הפרוצדורה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
begin
  -- Call the procedure
  schedule_meeting(idpatient => :idpatient,
                   idnurse => :idnurse);
end;
```

Variable	Type	Value
idpatient	Integer	253938640
idnurse	Integer	500417878

נראה את המערכת לאחר השינוי

SQL | Output | Statistics

```
select* from schedule;
```

Select patient | Select schedule | Select area | Select nurse | Select

	NURSEID	PATIENTID	MEETINGTIME
1	500417878	253938640	29/07/2020
2	500417878	479425698	30/05/2020
3	500417878	499874252	25/03/2020
4	501152661	288593932	17/05/2020
5	503563632	157137258	16/04/2020
6	503563632	180629307	08/02/2020
7	503563632	189458552	20/01/2020
8	508705560	311054470	13/05/2020
9	508705560	584592729	17/06/2020
10	513774513	224994634	19/05/2020
11	515354407	112417975	05/06/2020
12	515354407	320830486	11/06/2020
13	516434238	556869539	23/02/2020
14	518785197	185073041	23/07/2020
15	518881254	503137815	08/05/2020
16	519528217	188656382	08/05/2020
17	519528217	309973460	10/05/2020
18	519528217	446493518	17/01/2020

אם נרצה להכניס מישהו שכבר יש לו פגישה:

Test script | DBMS Output | Statistics | Profiler | Trace

```
begin
  -- Call the procedure
  schedule_meeting(idpatient => :idpatient,
                   idnurse => :idnurse);
end;
```

Variable	Type	Value
idpatient	Integer	479425698
idnurse	Integer	500417878

נקבל :

Test script | DBMS Output | Statistics | Profiler | Trace

Clear | Buffer size 10000 | Enabled

```
Patient 479425698 already has meeting
```

טריגרים:

טריגר 1:

לפני ביצוע ניתוק מטופל ממכונת הנשמה – נעדכן את מצבו של החולה לקל.

```
create or replace noneditionable trigger disconnect_respirator_trigger
  after update
  on respirator
  for each row
declare
begin
  if(:new.patientid is null) then
    update patient
    set patient.diseasecondition = 'easy'
    where patient.patientid = :old.patientid;
  end if;
end disconnect_respirator_trigger;
```

נראה את פעולת הטריגר:

```
select patientid, diseasecondition, respiratorid, daysofuse
from respirator r natural join patient p
where patientid = 451334638;
```

	PATIENTID	DISEASECONDITION	RESPIRATORID	DAYSOFUSE
1	451334638	medium	74	17

מטופל מספר 451334638 מחובר למכונה מספר 74 ומצבו בינוני.

כעת נעדכן ניתוק של המטופל מהמכונה:

```
update respirator
set respirator.patientid = null, respirator.daysofuse = null
where respirator.respiratorid = 74;
```

נראה שמכונה מספר 74 לא מחוברת לאף מטופל:

```
select *
from respirator
where respirator.respiratorid = 74;
```

Select respirator | Select respirator | Select patient

	RESPIRATORID	PATIENTID	DAYSOFUSE	LASTCHECK
1	74			1/23/2010

נראה שהטריגר אכן פעל – מצבו של המטופל הנ"ל הפך לקל:

```
select *
from patient p
where patientid = 451334638;
```

	PATIENTID	DISEASECONDITION
1	451334638	easy

<u>טריגר 2:</u>

כאשר נוסיף מתמחה חדש לטבלת Intern, הטריגר יבדוק אם הוכנס מספר ת.ז. עם 8 ספרות. אם כן – יוסיף 0 בסוף הת.ז.

```
create or replace trigger fix_id
  before insert
  on intern
  for each row
     when (new.internid < 100000000 and new.internid > 9999999) --8 digits
declare
a intern.internid%type;
begin
  a := :new.internid;
  :new.internid := a*10;
  dbms_output.put_line('ID changed to:');
  dbms_output.put_line(:new.internid);
end fix_id;
```

נראה את פעולת הטריגר – נכניס מתמחה עם ת.ז. בעלת 8 ספרות:

```
insert into intern (INTERNID, INTERNNAME, SALARY, UNIVERSITY)
values (34543476, 'Ram Amitsur', 5000, 'Jct');
```

נראה את הפלט המתקבל:

```
SQL   Output   Statistics
Clear   Buffer size 10000   Enabled
ID changed to:
34534760
```

וכן נראה שהמתמחה נשמר עם 0 בסוף הת.ז.:

```
select *
from intern i
where internname = 'Ram Amitsur';
```

	INTERNID	INTERNNAME	UNIVERSITY	SALARY
1	345434760	Ram Amitsur	Jct	5000

<u>טריגר 3:</u>

כאשר נרצה לעדכן את המשכורות של הרופאים באחוז מסויים(פה בחרנו להכפיל את המשכורת 1.02 בשביל הדוגמא) נבדוק אם אחרי ההוספה המשכורת שהתקבלה לא עוברת את המשכורת המקסימלית עבור כל תחום התמחות

```
create or replace trigger add_to_salary
before update on doctor
for each row

declare
newsalary number;
special varchar2(17);
max_salary number;


begin
  newsalary:= :new.salary;
  special:= :new.specialty;
  if special = 'Anesthesiology' then max_salary:=220000; end if;
  if special = 'Dermatology' then max_salary:=230000; end if;
  if special = 'family medicine' then max_salary:=230000; end if;
  if special = 'neurology' then max_salary:=220000; end if;
  if special = 'ophthalmology' then max_salary:=223000; end if;
  if special = 'pathology' then max_salary:=240000; end if;
  if special = 'radiology' then max_salary:=220000; end if;
  if newsalary*1.02<max_salary then :new.salary:= newsalary*1.02; end if;
  if newsalary*1.02>max_salary then :new.salary:= max_salary; end if;
end add_to_salary;
```

נראה דוגמא עבוד שני רופאים:

```
select *
from doctor
where doctor.doctorid=355448830;
```

	DOCTORID	DOCTORNAME	EXPERIENCE	SPECIALTY	DEPHEADID	SALARY
1	355448830	Hilary Skarsgard	10	Anesthesiology	559866995	108630

```
select *
from doctor
where doctor.doctorid=108805248;
```

	DOCTORID	DOCTORNAME	EXPERIENCE	SPECIALTY	DEPHEADID	SALARY
1	108805248	Connie Quinn	3	Anesthesiology	559866755	219300

נפעיל את הטריגר ונעדכן את הטבלה:

```
update doctor
set salary=salary*1.02;
```

נראה את השינוי המשכורת למישהו שלא עובר את המשכורת המקסימלית:

Select doctor | Select doctor | Update doctor | Select doctor | Select doctor

	DOCTORID	DOCTORNAME	EXPERIENCE	SPECIALTY	DEPHEADID	SALARY
1	355448830	Hilary Skarsgard	10	Anesthesiology	559866995	113018.652

נראה את השינוי המשכורת למישהו שכן עובר את המשכורת המקסימלית:

Select doctor | Select doctor | Update doctor | Select doctor | Select doctor

	DOCTORID	DOCTORNAME	EXPERIENCE	SPECIALTY	DEPHEADID	SALARY
1	108805248	Connie Quinn	3	Anesthesiology	559866755	220000

טריגר 4:

כאשר הדירוג של האחות יעלה, היא תקבל ברכה מכל המטופלים שלה.

```
create or replace trigger update_rating_trigger
  after update
  on nurse
  for each row
    when (new.rating > old.rating)
declare
  cursor patients is select patient.patientname
  from patient
  where patient.nurseid = :old.nurseid;
begin
  dbms_output.put_line('Congratulations!!');
  dbms_output.put_line('We want to thank you for taking care of us');
  dbms_output.put_line('Appreciate, your patients:');
  dbms_output.put_line(' ');
  for p in patients
    loop
      dbms_output.put_line(p.patientname);
    end loop;
end update_rating_trigger;
```

לשם הדגמה נבחר אחות ונראה מה הדירוג הנוכחי שלה:

```
select *
from nurse
where nurse.nurseid = 740712516;
```

	NURSEID	NURSENAME	AREAID	RATING	HIREYEAR	BOSSID	SALARY
1	740712516	Jim Peterson	2	2	1998	813603292	11971

כעת נעלה את הדירוג שלה ונוודא שאכן הוא עלה:

```
update nurse
set nurse.rating = 10
where nurse.nurseid = 740712516;

select *
from nurse
where nurse.nurseid = 740712516;
```

Update nurse | Select nurse

	NURSEID	NURSENAME	AREAID	RATING	HIREYEAR	BOSSID	SALARY
1	740712516	Jim Peterson	2	10	1998	813603292	11971

נראה מה הפלט שיצר הטריגר:

```
Congratulations!!
We want to thank you for taking care of us
Appreciate, your patients:

Carlos Stuart
Lloyd Yulin
Martin Rydell
Meg Bancroft
Ellen Dench
```

תכניות:

תכנית 1:

התוכנית תעבור על כל החולים שגרים בדימונה ומצבם קשה. אם הם לא מחוברים – למכונת הנשמה, נחבר אותם. אחרי החיבור למכונה נכתוב על כל אחד מהחולים האם הוא בקבוצת סיכון, וכמה אנשים נחשפו אליו בזמן מחלתו.

התוכנית משתמשת בפונקציה 1 – איתור מספר הנחשפים הכולל, ובפרוצדורה 1 – חיבור למכונת הנשמה.

```
declare
--Move on the patients from Dimona who are very sick
 cursor patients_cur is select *
                        from patient
                        where patient.diseasecondition = 'hard'
                        and patient.cityname = 'Dimona';
exposed number(3);
begin
  for p in patients_cur
    loop

      connect_to_respiratoe(p.patientid);
      exposed := how_many_exposed(p.patientid);
      if p.riskgroup = 1 then
        dbms_output.put_line(p.patientname || ' is in risk group');
      else
        dbms_output.put_line(p.patientname || ' is *not* in risk group');
      end if;
      dbms_output.put_line(exposed || ' people were exposed to ' || p.patientname);
      dbms_output.put_line('- - - - - - - -');
      dbms_output.put_line(' ');
     end loop;
end;
```

פלט התוכנית:

```
patient 501362231 connectes to respirator # 358
Denise Robbins is *not* in risk group
107 people were exposed to Denise Robbins
- - - - - - - -

patient 155457847 connectes to respirator # 359
Trini Maguire is in risk group
35 people were exposed to Trini Maguire
- - - - - - - -

this patient is allready connected to respirator
Annie Spacek is in risk group
0 people were exposed to Annie Spacek
- - - - - - - -
```

<u>תכנית 2:</u>

תכנית זו בודקת עבור רופא שהת.ז. שלו הוא 109902179 אם יש לו מטופלים חולי קורונה ואם כן עבור כל אחד מהחולים המערכת קבע פגישה עם האחות 662575742 בתאריך 29/7/2020

תוך שימוש בפונקציה 4 שבודקת עבור הרופא אם יש לו מטופלים חולי קורונה ושימוש בפרוצדורה 2 שעבור חולה ואחות קובעת להם פגישה יחד רק אם אין לחולה כבר פגישה

```
declare
 numpatient number(3);
 cursor all_patient is
    select *
    from patient
    where patient.doctorid='109902179';
begin
  numpatient:= there_is_patient(109902179);
  if numpatient>0 then
     for p in all_patient
       loop
       schedule_meeting(p.patientid,'662575742');
       dbms_output.put_line('patient '||p.patientid||'has new meeting at 29/7/2020');
       dbms_output.put_line('-------------------');
       end loop;
  else
    dbms_output.put_line('Doctor number 201622746 dosent have parients');
  end if;

end;
```

פלט התוכנית:

```
sum of patient for 109902179 is 5
Patient 577983736 dose not have meeting
patient 577983736has new meeting at 29/7/2020
-------------------
Patient 589084496 dose not have meeting
patient 589084496has new meeting at 29/7/2020
-------------------
Patient 576451242 already has meeting
patient 576451242has new meeting at 29/7/2020
-------------------
Patient 503444427 dose not have meeting
patient 503444427has new meeting at 29/7/2020
-------------------
Patient 260691810 dose not have meeting
patient 260691810has new meeting at 29/7/2020
-------------------
```

אפשר לראות שמערכת מדפיסה שאכן לחולה לא הייתה פגישה.