

עלך מציון אלף מתנדבים

דינה גלצר 324281716

deenacop@gmail.com

יראת ביטון 212628408

Yirat18@gmail.com

פירוט ה ERD

יש ישות בן אדם שממנה יורשים המנהל, האחראי משמרת, המתנדב והנצרך שמקבלים את כל הישויות של הבן אדם

התכונות של הבן אדם הם: שם פרטי, שם משפחה, ת.ז. , כתובת ומספר טלפון.

למנהל יש גם רווח חודשי של הארגון.

לאחראי משמרת יש את מספר המתנדבים שהוא אחראי עליהם.

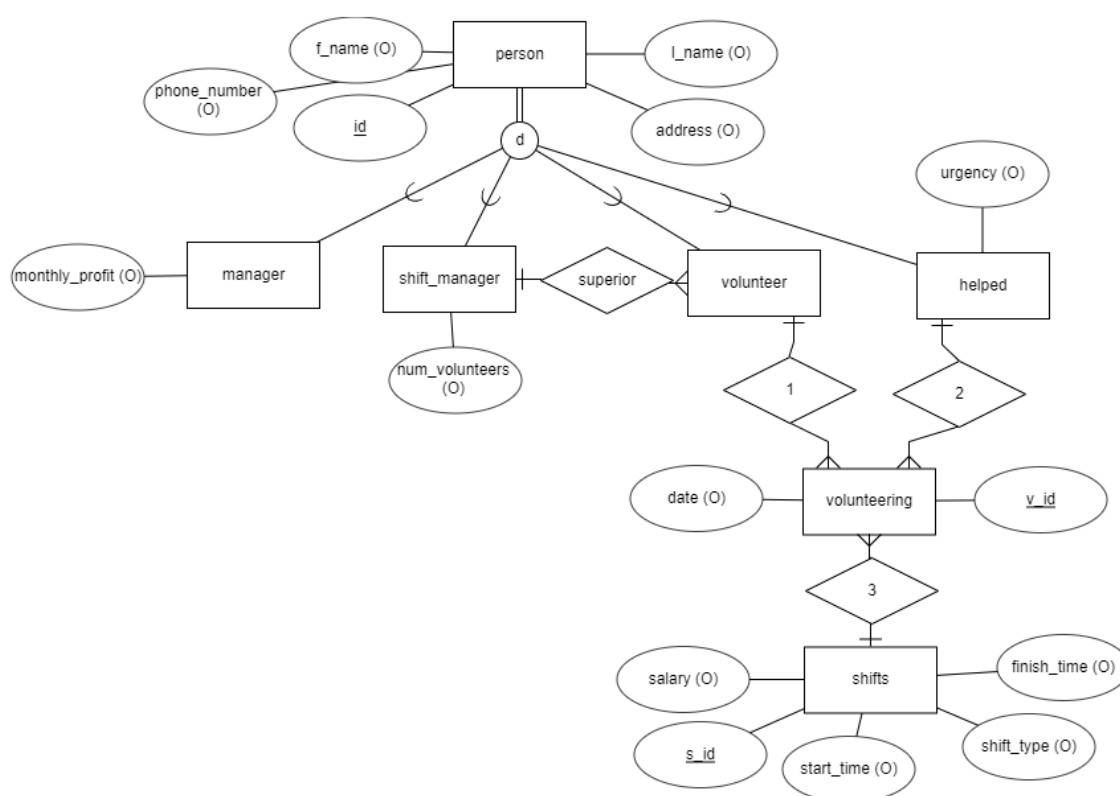
לכל מתנדב יש אחראי משמרת אחד בדיוק שהוא ממונה עליו.

לנצרך יש את רמת הדחיפות.

למשמרות יש את המספר מזהה, את הזמן התחלה, זמן סיום של המשמרת, את השכר ואת סוג המשמרת.

כתבנו את הקשר כישות בגלל המגבלה שאין באתר קשר שמחבר בין 3 ישויות והקשר הוא ההתנדבות ולכל קשר יש מספר זיהוי ואת התאריך שבו יש מתנדב נצרך והמשמרת.

סכמת ה ERD

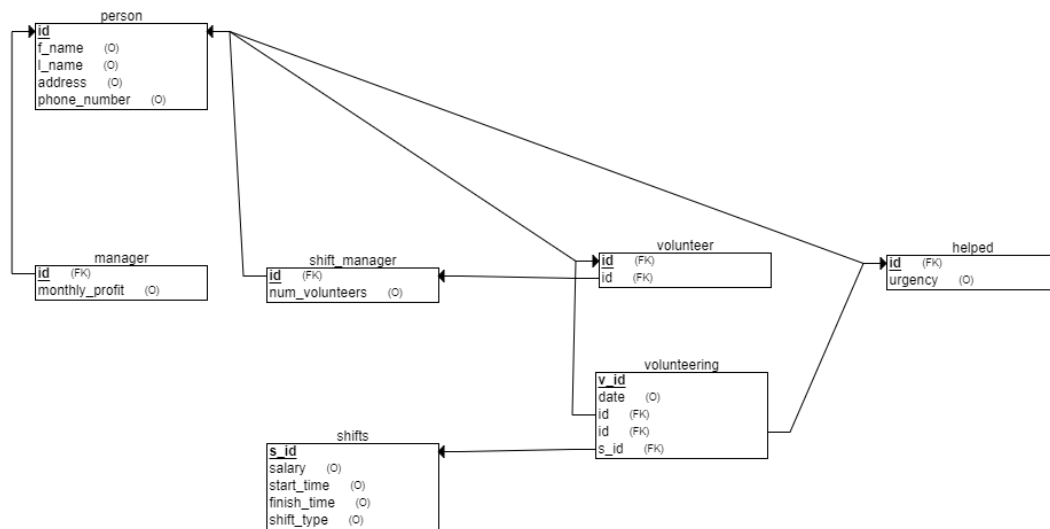


Shifts – משמרות

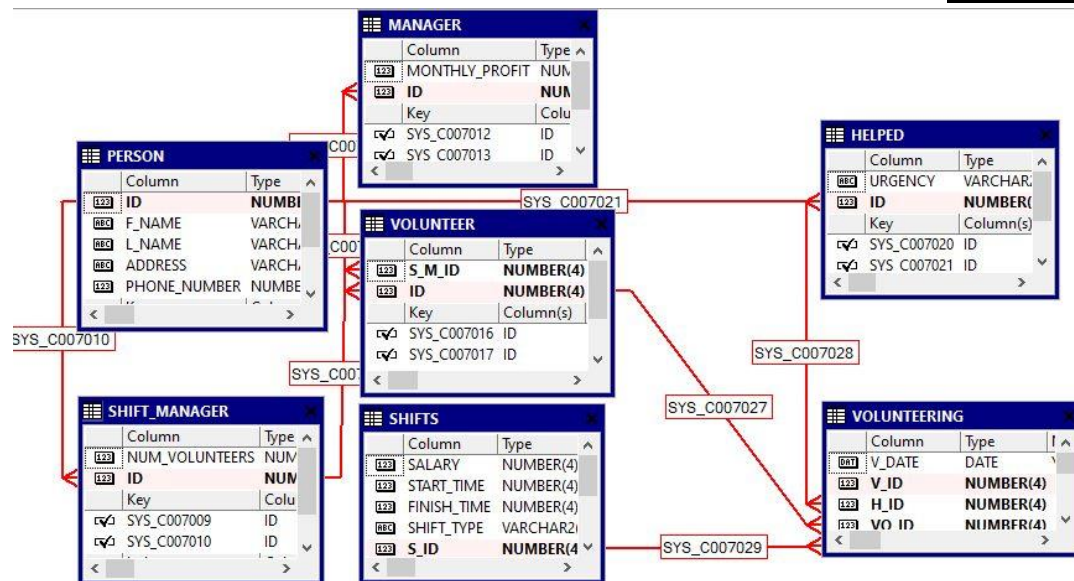
Volunteering - התנדבות

Shift manager – אחראי משמרת

:DSD



:SQL טבלאות



מילוי נתונים

שיטה 1 insert

```
SQL Window - select * from Person; insert into Person values(1111,'dina','glecer','har_nof',0001); insert int ...

SQL      Output      Statistics

insert into Person values(1111,'dina','glecer','har_nof',0001);
insert into Person values(2222,'yirat','biton','uziel',0002);
insert into Person values(3333,'shira','pinski','har_nof',0003);
insert into Person values(4444,'penina','rabi','uziel',0005);
insert into Person values(5555,'shirel','perez','bait_vagan',0006);
select * from Volunteer;
insert into Volunteer values(1111,3333);
insert into Volunteer values(3333,4444);
insert into Volunteer values(4444,1111);
select * from Shift_Manager;
insert into Shift_Manager values(2,1111);
insert into Shift_Manager values(2,3333);
insert into Shift_Manager values(1,4444);
insert into Shift_Manager values(0,5555);
```

שיטה שניה DXT

שיטה שלישית excel

Data from ODBC Data to Oracle

General
Owner: [] Table: VOLUNTEER [] Clear Table
Commit every...: 0 [] Overwrite duplicates [] Ignore duplicates
Initializing Script: []
Finalizing Script: []

Fields
1 -> ID
2 -> S_M_ID
Field: S_M_ID
Fieldtype: (Float)
SQL function: []
additional Oracle processing, for example substr(%, 1, 20)

Result Preview

1	2
300.0	233.0
301.0	106.0

Import Import to Script Close system@XE 3 records imported in 0.047 seconds Help

Data from ODBC Data to Oracle

General
Owner: [] Table: PERSON [] Clear Table
Commit every...: 0 [] Overwrite duplicates [] Ignore duplicates
Initializing Script: []
Finalizing Script: []

Fields
1 -> ID
2 -> F_NAME
3 -> L_NAME
4 -> ADDRESS
5 -> PHONE_NUMBER
Field: PHONE_NUMBER
Fieldtype: (Float)
SQL function: []
additional Oracle processing, for example substr(%, 1, 20)

Result Preview

1	2	3	4	5
300.0	denna	glecer	35 Lopes Street	2022.0
301.0	talya	biton	48 Cullin Drive	4325.0
302.0	yirrat	mizrachi	19 Channing Dri	1980.0

Import Import to Script Close system@XE 3 records imported in 0.094 seconds Help

Data from ODBC Data to Oracle

General
Owner: [] Table: HELPED [] Clear Table
Commit every...: 0 [] Overwrite duplicates [] Ignore duplicates
Initializing Script: []
Finalizing Script: []

Fields
1 -> ID
2 -> URGENCY
Field: URGENCY
Fieldtype: Number
SQL function: []
additional Oracle processing, for example substr(%, 1, 20)

Result Preview

1	2
303.0	2.0
304.0	3.0

Import Import to Script Close system@XE 3 records imported in 0.093 seconds Help

DATA GENERATOR

PERSON

Owner: SYSTEM Table: PERSON Number of records: 200..300

Name	Type	Size	Data	Master
ID	NUMBER	4	Sequence(10)	...
F_NAME	VARCHAR2	15	FirstName	...
L_NAME	VARCHAR2	15	LastName	...
ADDRESS	VARCHAR2	15	Address1	...
PHONE_NUMBER	NUMBER	4	Random(1000, 9999)	...

ID	F_NAME	L_NAME	ADDRESS	PHONE_NUMBER
1	dina	glecer	har_nof	1
2	yirrat	biton	uziel	2
3	shira	pinski	har_nof	3
4	penina	rabi	uziel	5
5	shirel	perez	bait_vagan	6
6	talya	copperman	har_nof	7
7	mordecai	'glecer'	uziel	8
8	moshe	biton	rmot	9
9	Kyra	Moorer	26 Blair Street	1483
10	Dick	Howard	52 Kwun Tong Av	3403
11	Jose	Hedaya	63 Studi Blvd	1360
12	Pierce	Scott	71st Street	8394
13	Rachid	Goldblum	388 Richardson	1829
14	Ritchell	Lauper	46 Rowlands Roa	6425
15	Linda	Keitel	26 Byrd Road	7932
16	Joaquim	Breslin	23 Child Drive	2036
17	T	N	221 K	2472

MANAGER

Owner	Table	Number of records
SYSTEM	MANAGER	10..20

Name	Type	Size	Data	Master
MONTHLY_PROFIT	NUMBER	4	Random(5000, 9999)	...
ID	NUMBER	4	List(select id from Person where id not in (select

```
insert into shift Manager values(0,5555);
```

	MONTHLY_PROFIT	ID
1	9606	121
2	7477	146
3	8486	60
4	8608	142
5	5568	196

16:1 system@XE 5 rows selected in 0.046 seconds

HELPED

Owner: SYSTEM Table: HELPED Number of records: 100..150

Name	Type	Size	Data	Master
URGENCY	VARCHAR2	15	Random(1, 3)	...
ID	NUMBER	4	List(select id from Person where id not in (select i

111 rows selected in 0.235 seconds

	URGENCY	ID
1	2	226
2	3	171
3	1	296
4	2	55
5	1	143
6	2	207
7	1	26
8	2	191
9	3	136
10	3	105
11	1	142
12	2	104

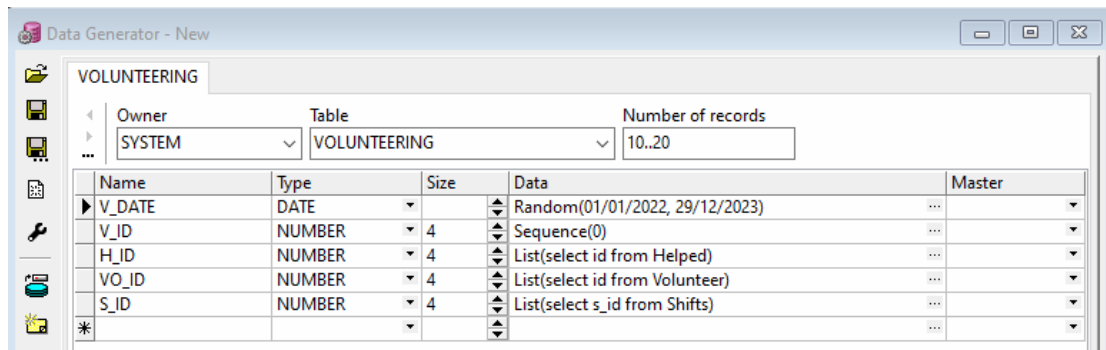
VOLUNTEER

Owner: SYSTEM Table: VOLUNTEER Number of records: 150..200

Name	Type	Size	Data	Master
S_M_ID	NUMBER	4	List(select id from Shift_Manager)	...
ID	NUMBER	4	List(select id from Person where id not in (select id

130 rows selected in 0.297 seconds

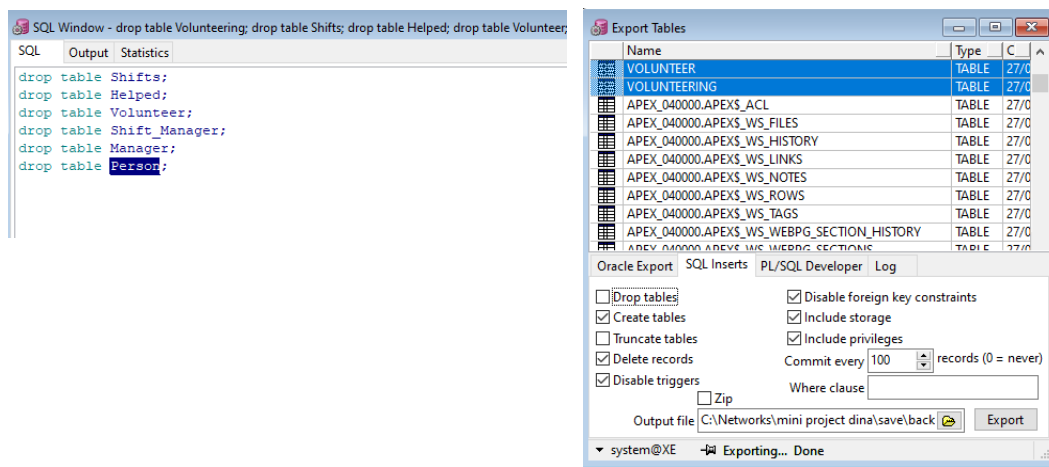
	S_M_ID	ID
1	65	238
2	233	94
3	106	267
4	183	139
5	165	44
6	5555	96
7	111	275
8	28	16
9	135	127
10	257	131
11	1111	249
12	170	124



	V_DATE	V_ID	H_ID	VO_ID	S_ID
1	19/07/2022	0	261	294	16
2	22/01/2022	1	109	109	36
3	09/12/2022	2	176	33	20
4	13/11/2022	3	141	197	26
5	14/06/2022	4	235	101	24
6	19/01/2023	5	217	163	7
7	24/02/2023	6	113	227	8
8	19/09/2022	7	189	162	39
9	15/01/2023	8	61	13	10
10	06/10/2023	9	31	249	27

system@XE 18:1 138 rows selected in 0.375 seconds

גיבוי ושיחזור



שאלות

1. השאילתה מחזירה את ה-ת.ז. והשם של מתנדב שמתנדב לפחות אצל 3 נצרכים.

SQL Window - query1.sql

```
SQL
SELECT ID , f_name
FROM volunteer Natural join person
WHERE ID IN (
  SELECT VO_ID
  FROM volunteering
  GROUP BY VO_ID
  HAVING COUNT(DISTINCT H_ID) >= 3
);
```

ID	F_NAME
1	223 Anjelica
2	262 Mint
3	163 Jonatha
4	32 Ashton
5	46 Madeleine
6	69 Eddie
7	212 Juliet
8	38 Mindy
9	48 Joaquin

17 rows selected in 0.171 seconds

2. השאילתה משתמשת בהצהרת SELECT בתוך הצהרת SELECT אחרת כדי להחזיר את המתנדבים בעלי השכר הגבוה ביותר ואת המשכורות שלהם.

SQL Window - query2.sql

```
SQL
SELECT p.F_NAME, p.L_NAME, S.SALARY
FROM PERSON p NATURAL JOIN VOLUNTEER v ,VOLUNTEERING vl NATURAL JOIN SHIFTS S
WHERE ID = vl.VO_ID AND S.SALARY = (
  SELECT MAX(SALARY)
  FROM SHIFTS
)
ORDER BY P.F_NAME DESC;
```

F_NAME	L_NAME	SALARY
1	Yolanda Emmerich	3973
2	Red Hingle	3973
3	Rawlins Ossajca	3973
4	Lou Gooding	3973

4 rows selected in 0.062 seconds

3. השאילתה מחזירה את הכמות של הנצרכים שהדחיפות שלהם היא 3 (רמה גבוהה) והשם שלהם מתחיל באות C.

SQL Window - SELECT COUNT(ID) FROM HELPED H NATURAL...

SQL

Output

Statistics

SELECT COUNT(ID)
FROM HELPED H NATURAL JOIN PERSON P
WHERE H.URGENCY = 3 AND P.F_NAME LIKE 'C%';a

↕

↕

Grid

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

Table

<

4. השאילתה מחזירה את השם, הכתובת והשכר הכולל שהרוויחו מתנדבים שעבדו לפחות משמרת אחת עם שכר גבוה מהשכר הממוצע של כל המשמרות.

SQL Window - WITH avg_salary AS (SELECT AVG(S.SALARY) AS average_salary FROM SHIFTS S), volunteer_salary AS ...

```
SQL
WITH avg_salary AS (
  SELECT AVG(S.SALARY) AS average_salary
  FROM SHIFTS S
),
volunteer_salary AS (
  SELECT P.F_NAME, P.ADDRESS, SUM(S.SALARY) AS total_salary
  FROM PERSON P NATURAL JOIN VOLUNTEER v , VOLUNTEERING vl NATURAL JOIN SHIFTS S
  WHERE ID = vl.VO_ID
  GROUP BY P.F_NAME, P.ADDRESS
  HAVING MAX(S.SALARY) > (SELECT average_salary FROM avg_salary)
)
SELECT *
FROM volunteer_salary;
```

F_NAME	ADDRESS	TOTAL SALARY
1	Johnny 94 Apple Drive	7269
2	Juliet 80 Zevon Ave	9597
3	Anjelica 84 Rheinfelden	12884
4	Lou 22 Oakfold Ro	8424
5	Austin 69 Culkin Drive	3931
6	yirat 19 Channing Dri	17283
7	Jonatha 7 Thame Road	9488
8	Dick 52 Kwun Tong Av	7624
9	Richard 594 Or-yehuda S	3390
10	Joaquin 431 Linc Road	6047

UPDATE

```

SQL Window - UPDATE SHIFT_MANAGER SET NUM_VOLUNTEERS = ( SELECT COU...
SQL      Output      Statistics
UPDATE SHIFT_MANAGER
SET NUM_VOLUNTEERS = (
  SELECT COUNT (*)
  FROM VOLUNTEER
  WHERE VOLUNTEER.S_M_ID = SHIFT_MANAGER.ID
  GROUP BY VOLUNTEER.S_M_ID
)

```

system@XE 42 rows updated in 0.015 seconds

1. שאילתה זו מעדכנת את השדה NUM_VOLUNTEERS

בטבלה SHIFT_MANAGER כדי לשקף את מספר

המתנדבים שעליהם אחראי כל מנהל משמרת,

מחושב על ידי ספירת מספר הרשומות בטבלת

VOLUNTEER עבור כל מנהל משמרת

באמצעות GROUP BY.

לפני השינוי:

	NUM_VOLUNTEERS	ID
1	2	1111
2	2	3333
3	1	4444
4	0	5555

אחרי השינוי:

	NUM_VOLUNTEERS	ID
1	2	1111
2	3	3333
3	5	4444
4	5	5555

2. שאילתה זו מעדכנת את סוג השכר והמשמרת

עבור כל המשמרות המשויות למתנדבים

שעובדים תחת מנהלי משמרת המנהלים יותר

מ-5 מתנדבים, ומוקצות לסייע לאלה עם צרכים

דחופים גבוהים.

```

UPDATE SHIFTS
SET SALARY = SALARY + 500, SHIFT_TYPE = 'On-call shift'
WHERE S_ID IN (
  SELECT S_ID FROM VOLUNTEERING
  WHERE VO_ID IN (
    SELECT ID FROM VOLUNTEER
    WHERE S_M_ID IN (
      SELECT ID FROM SHIFT_MANAGER
      WHERE NUM_VOLUNTEERS > 5
    )
  )
)
AND H_ID IN (
  SELECT ID FROM HELPED
  WHERE URGENCY = 3
)
);

```

system@XE 28 rows updated in 0.032 seconds

לפני השינוי:

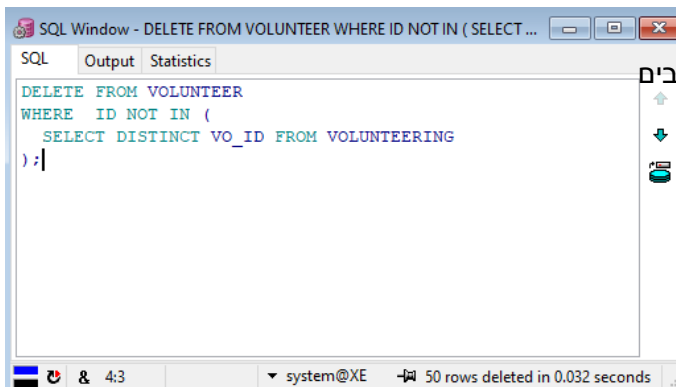
		SALARY	START_TIME	FINISH_TIME	SHIFT_TYPE	S_ID
▶	1	2957	23:00	00:00	Weekend shift	0
	2	3614	07:00	13:00	Day shift	1
	3	2995	03:00	15:00	Holiday shift	2
	4	2655	20:00	20:00	Holiday shift	3

אחרי השינוי:

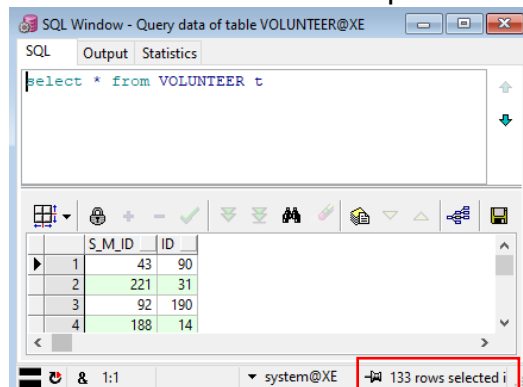
		SALARY	START_TIME	FINISH_TIME	SHIFT_TYPE	S_ID
▶	1	3457	23:00	00:00	On-call shift	0
	2	3614	07:00	13:00	Day shift	1
	3	3495	03:00	15:00	On-call shift	2
	4	3155	20:00	20:00	On-call shift	3

delete

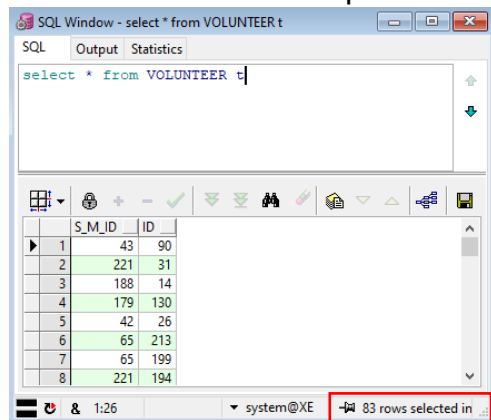
1. מחיקת כל רישומי המתנדבים למתנדבים שאין להם בכלל משמרות

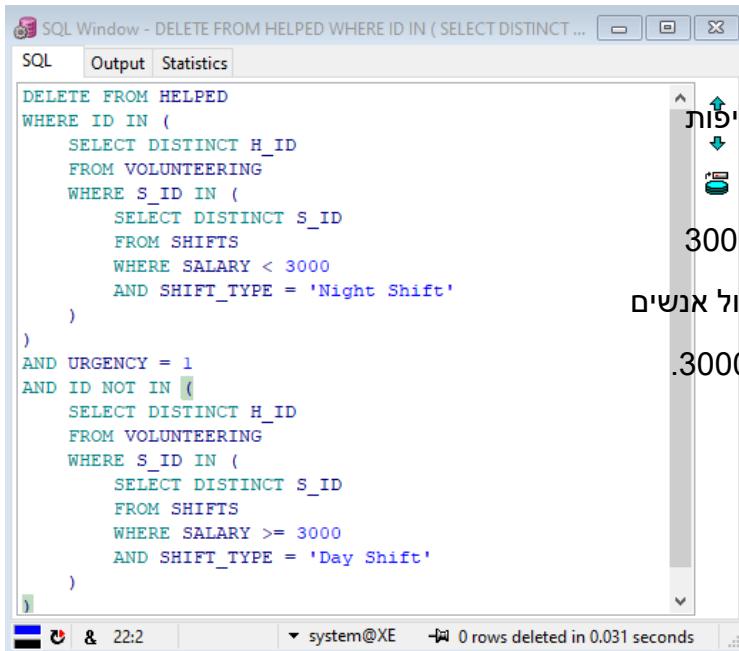


לפני המחיקה:



אחרי המחיקה:



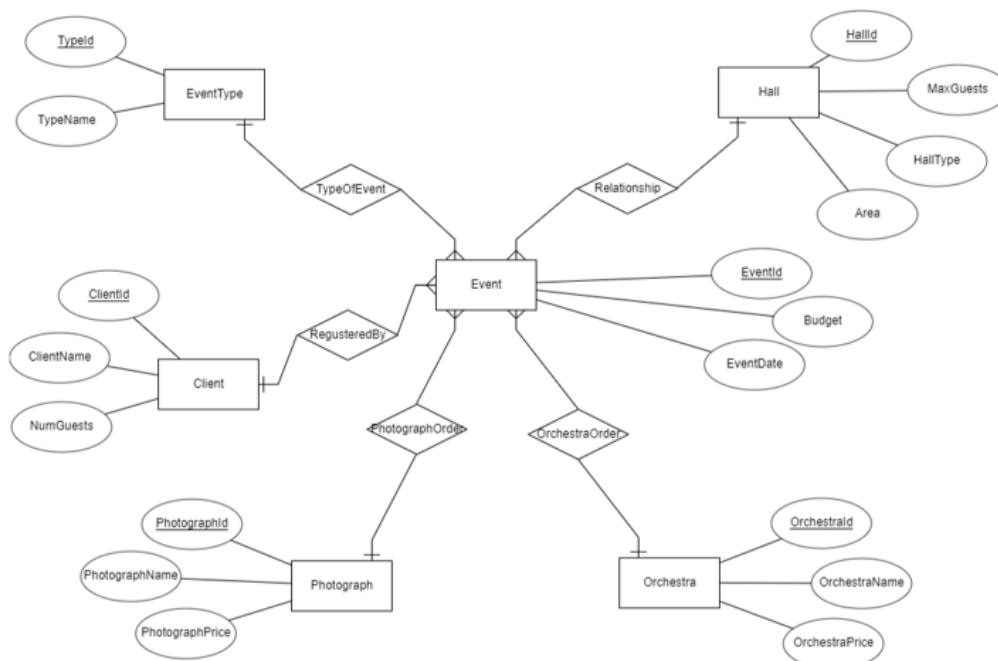


```
DELETE FROM HELPED
WHERE ID IN (
  SELECT DISTINCT H_ID
  FROM VOLUNTEERING
  WHERE S_ID IN (
    SELECT DISTINCT S_ID
    FROM SHIFTS
    WHERE SALARY < 3000
    AND SHIFT_TYPE = 'Night Shift'
  )
)
AND URGENCY = 1
AND ID NOT IN (
  SELECT DISTINCT H_ID
  FROM VOLUNTEERING
  WHERE S_ID IN (
    SELECT DISTINCT S_ID
    FROM SHIFTS
    WHERE SALARY >= 3000
    AND SHIFT_TYPE = 'Day Shift'
  )
)
```

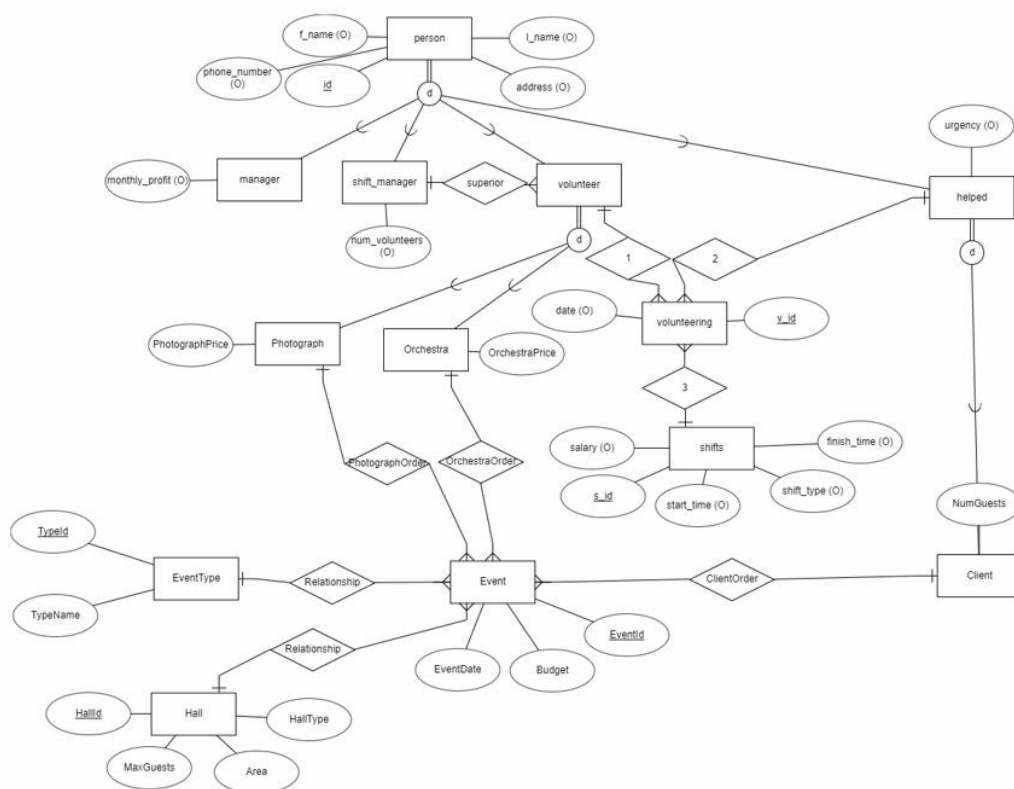
0 rows deleted in 0.031 seconds

שאלתה זו מוחקת נצרכים שנעזרו בדרגת דחילות
נמוכה ולא נעזרו במשמרות יום
ונעזרו במשמרות לילה עם משכורת נמוכה מ3000
אנו משתמשים גם בסעיף NOT IN כדי לא לכלול אנשים
שנעזרו במשמרות יום עם משכורת גבוהה מ3000.

אינטגרציה:



חיבור של 2 הERD:



הדיאגרמה שלנו היא דיאגרמת מתנדבים שמתנדבים בעזר מציון וקיבלנו דיאגרמה של הפקת אירועים בשביל האינטגרציה.

עשינו את האינטגרציה בצורה הבאה:

* התיחסנו לאירועים כאירועים שעזר מציון עוזר להם ולכן המתנדבים שלנו הם הצלמים והזמרים באירוע.

* הנצרכים במקרה הזה יהיו הלקוח של האירוע.

* עשינו ירושה מהישות מתנדב לישויות של הצלם והזמר

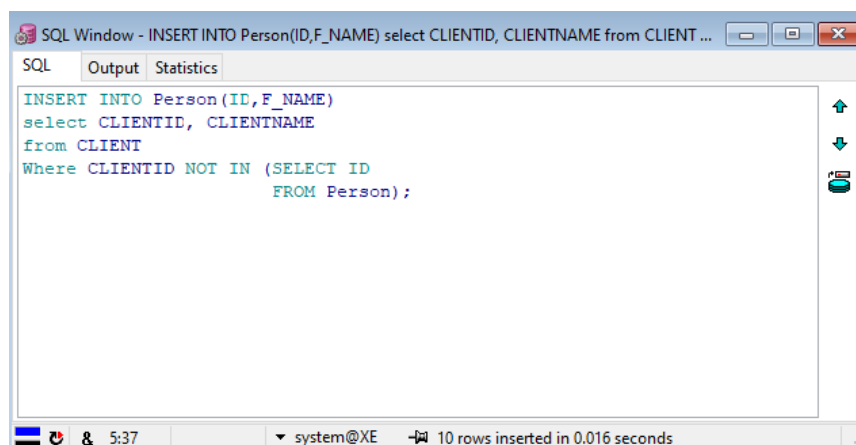
* עשינו ירושה בין הישות של הנצרך לישות של הלקוח של האירוע.

* הורדנו לישות Photograph את ה id ואת ה name מכיוון שהוא יורש את זה מהמתנדב.

* הורדנו לישות Orchestra את ה id ואת ה name מכיוון שהוא יורש את זה מהנצרך.

הכנסת נתונים מ client ל person

מכיוון ש client יורש מ helped ו person יורש מ person

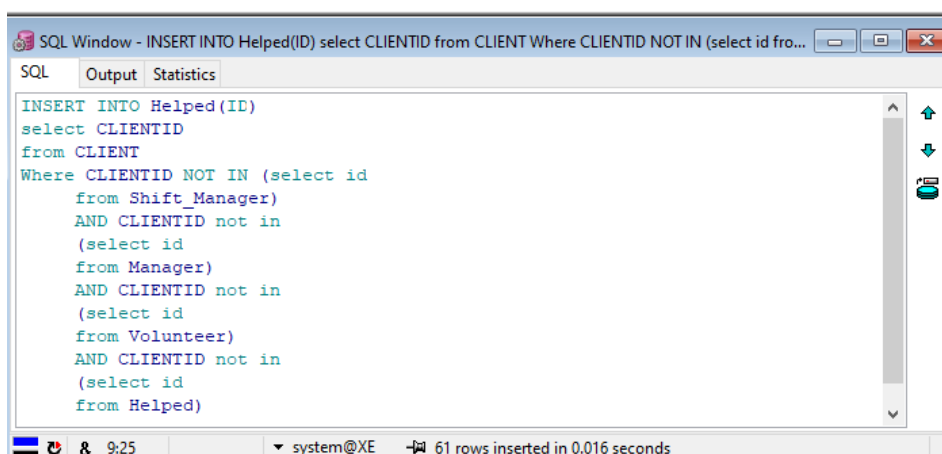


```
SQL Window - INSERT INTO Person(ID,F_NAME) select CLIENTID, CLIENTNAME from CLIENT ...
SQL Output Statistics
INSERT INTO Person(ID,F_NAME)
select CLIENTID, CLIENTNAME
from CLIENT
Where CLIENTID NOT IN (SELECT ID
                        FROM Person);
```

10 rows inserted in 0.016 seconds

הכנסת נתונים מ client ל helped

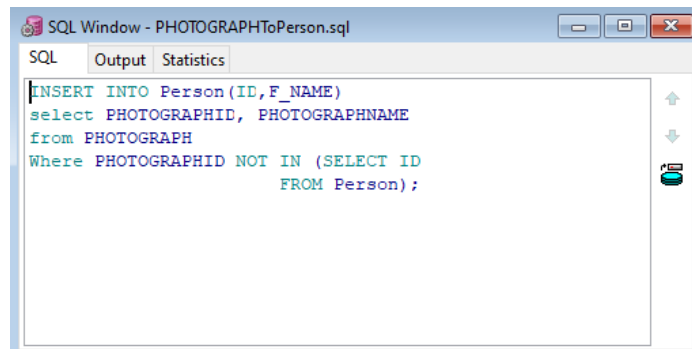
מכיוון ש client יורש מ helped



```
SQL Window - INSERT INTO Helped(ID) select CLIENTID from CLIENT Where CLIENTID NOT IN (select id fro...
SQL Output Statistics
INSERT INTO Helped(ID)
select CLIENTID
from CLIENT
Where CLIENTID NOT IN (select id
                        from Shift_Manager)
AND CLIENTID not in
(select id
from Manager)
AND CLIENTID not in
(select id
from Volunteer)
AND CLIENTID not in
(select id
from Helped)
```

61 rows inserted in 0.016 seconds

personֿ PHOTOGRAPH n

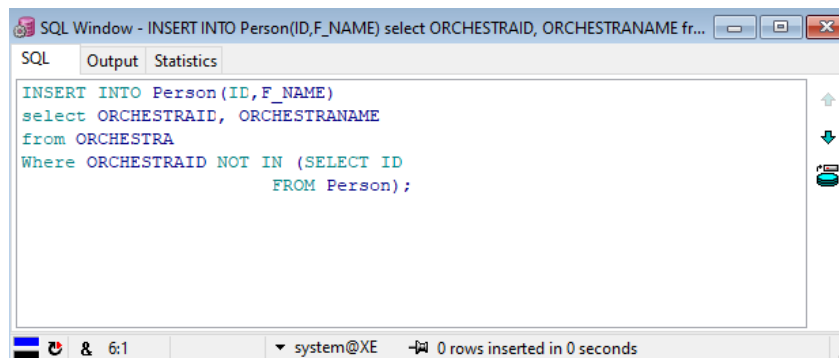


```

SQL Window - PHOTOGRAPHToPerson.sql
SQL Output Statistics
INSERT INTO Person(ID,F_NAME)
select PHOTOGRAPHID, PHOTOGRAPHNAME
from PHOTOGRAPH
Where PHOTOGRAPHID NOT IN (SELECT ID
                           FROM Person);

```

personֿ ORCHESTRAn



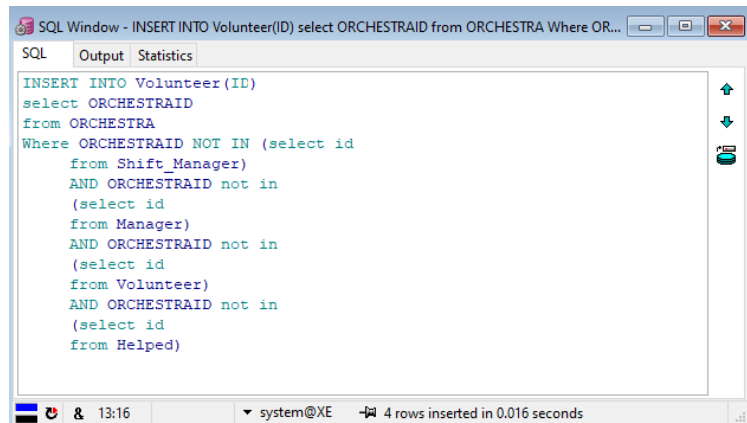
```

SQL Window - INSERT INTO Person(ID,F_NAME) select ORCHESTRAID, ORCHESTRANAME fr...
SQL Output Statistics
INSERT INTO Person(ID,F_NAME)
select ORCHESTRAID, ORCHESTRANAME
from ORCHESTRA
Where ORCHESTRAID NOT IN (SELECT ID
                           FROM Person);

```

6:1 system@XE 0 rows inserted in 0 seconds

volunteerֿ ORCHESTRAn



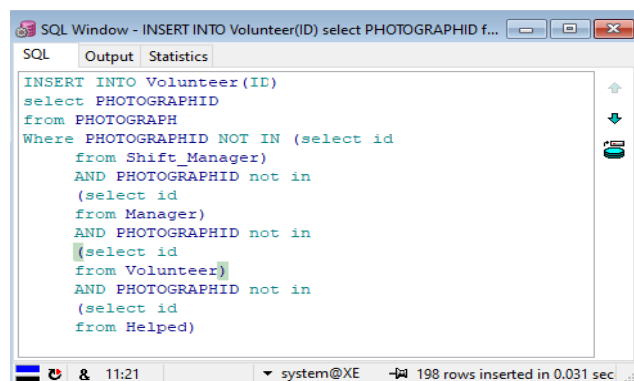
```

SQL Window - INSERT INTO Volunteer(ID) select ORCHESTRAID from ORCHESTRA Where OR...
SQL Output Statistics
INSERT INTO Volunteer(ID)
select ORCHESTRAID
from ORCHESTRA
Where ORCHESTRAID NOT IN (select id
                           from Shift_Manager)
                           AND ORCHESTRAID not in
                           (select id
                            from Manager)
                           AND ORCHESTRAID not in
                           (select id
                            from Volunteer)
                           AND ORCHESTRAID not in
                           (select id
                            from Helped)

```

13:16 system@XE 4 rows inserted in 0.016 seconds

volunteerֿ PHOTOGRAPHn



```

SQL Window - INSERT INTO Volunteer(ID) select PHOTOGRAPHID f...
SQL Output Statistics
INSERT INTO Volunteer(ID)
select PHOTOGRAPHID
from PHOTOGRAPH
Where PHOTOGRAPHID NOT IN (select id
                           from Shift_Manager)
                           AND PHOTOGRAPHID not in
                           (select id
                            from Manager)
                           AND PHOTOGRAPHID not in
                           (select id
                            from Volunteer)
                           AND PHOTOGRAPHID not in
                           (select id
                            from Helped)

```

11:21 system@XE 198 rows inserted in 0.031 sec

```

CREATE TABLE Client
(
    ClientId NUMERIC(3) NOT NULL,
    ClientName VARCHAR(15) NOT NULL,
    NumGuests INT NOT NULL,
    PRIMARY KEY (ClientId),
    FOREIGN KEY (ClientId) REFERENCES Helped(id)
);

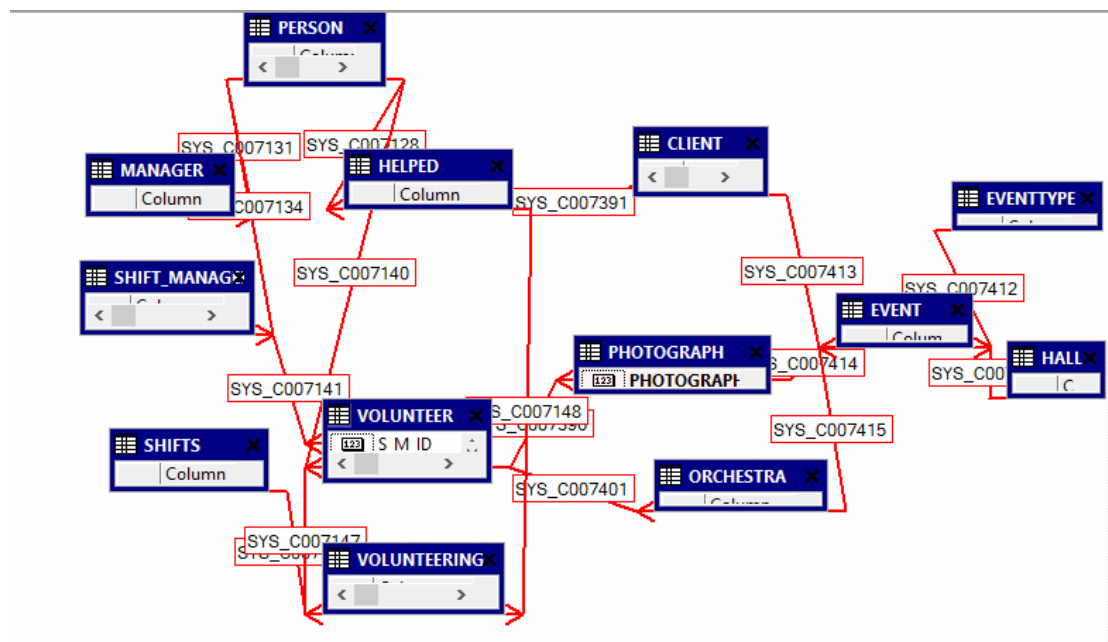
CREATE TABLE Photograph
(
    PhotographId NUMERIC(3) NOT NULL,
    PhotographName VARCHAR(15) NOT NULL,
    PhotographPrice INT NOT NULL,
    PRIMARY KEY (PhotographId),
    FOREIGN KEY (PhotographId) REFERENCES Volunteer(id)
);

CREATE TABLE OrchestraD
(
    OrchestraId NUMERIC(3) NOT NULL,
    OrchestraName VARCHAR(15) NOT NULL,
    OrchestraPrice INT NOT NULL,
    PRIMARY KEY (OrchestraId),
    FOREIGN KEY (OrchestraId) REFERENCES Volunteer(id)
);

```

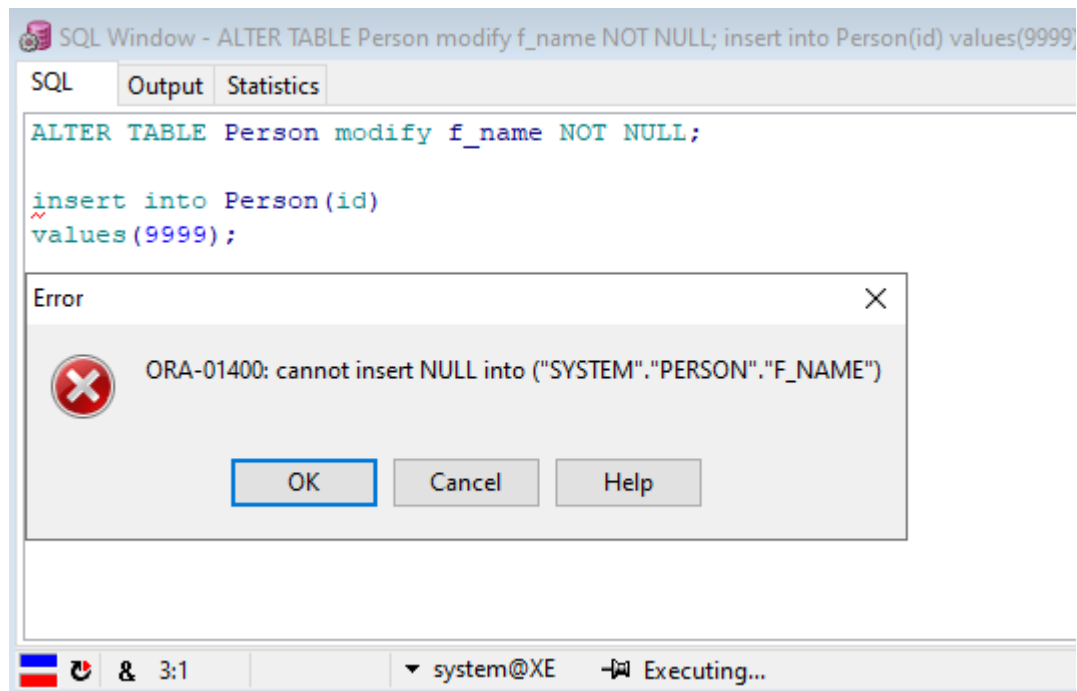
הוספנו מפתחות זרים לישויות שהוספנו

טבלאות SQL:



אילוצים:אילוץ ראשון:

השם בPERSON לא יכול להיות NULL

אילוץ שני:

ל shift_manager היה מספר משתתפים שהוא אחראי עליהם ועשינו שהברירת מחדל של זה תהיה 0

```
ALTER TABLE Shift_Manager modify num_volunteers default 0;

insert into Shift_Manager(id)
values(56);
```

SQL Window - select * from SHIFT_MANAGER t

SQL Output Statistics

```
select * from SHIFT_MANAGER t
```

	NUM_VOLUNTEERS	ID
36	17	51
37	18	233
38	28	49
39	17	78
40	28	66
41	21	106
42	28	43
43	0	56

אילוץ שלישי:

יש לנו ישות אולם ורצינו לבדוק שכמות האורחים לא תעלה על 1500

```

ALTER TABLE Hall
add constraint check_num_of_guest
check (maxGuests < 1500);

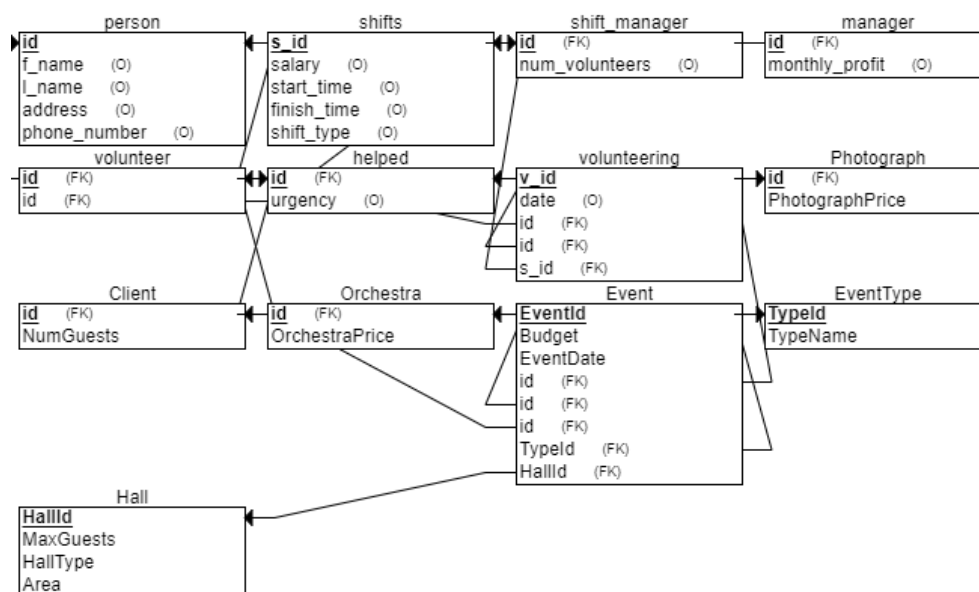
insert into Hall (Hallid, Maxguests, halltype, Area)
values (1, 2000, 'rr', 'tt');
        
```

Error

ORA-02290: check constraint (SYSTEM.CHECK_NUM_OF_GUEST) violated

OK Cancel Help

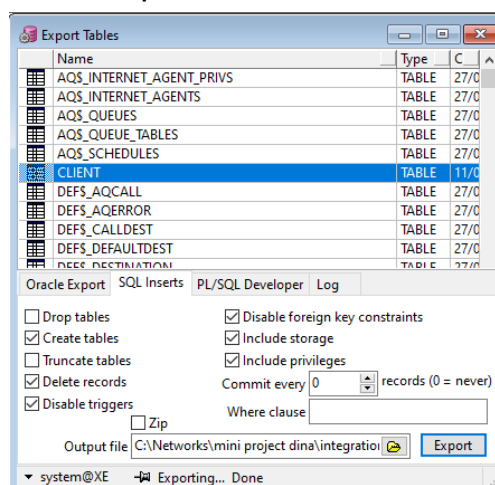
DSD לאחר האינטגרציה:



שאילתות:

לא היה שינוי של שם של הטבלאות ולכן כל השאילתות שכתבנו רצות טוב, בלי שום בעיות.

גיבוי:



תרגיל 3

שאלות:

- מוצא את פרטי האירוע, כולל שם סוג האירוע ותקציב, עבור אירועים בעלי תקציב גדול מהתקציב הממוצע של כל האירועים.

SQL Window - SELECT E.EVENTID, ET.TYPENAME, E.BUDGET FROM EVENT E JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID

```

SELECT E.EVENTID, ET.TYPENAME, E.BUDGET
FROM EVENT E
JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID
WHERE E.BUDGET > (
    SELECT AVG(E2.BUDGET)
    FROM EVENT E2
);

```

	EVENTID	TYPENAME	BUDGET
1	181	Bar Mitzva	34053
2	214	Brite	39232
3	151	Brite	29764
4	137	Birthday	33297
5	156	Wedding	38205

system@XE 28 rows selected in 0.203 seconds

- מוצא את השכר הממוצע של משמרות לכל סוג משמרת, לפי סוג המשמרת.

```

SELECT S.SHIFT_TYPE, AVG(S.SALARY) AS AVERAGE_SALARY
FROM SHIFTS S
GROUP BY S.SHIFT_TYPE
ORDER BY S.SHIFT_TYPE;

```

	SHIFT_TYPE	AVERAGE_SALARY
1	Day shift	3042.54545454545
2	Holiday shift	3010.9
3	Night shift	3236.11111111111
4	On-call shift	2918
5	Weekend shift	2871.625

system@XE 5 rows selected in 0.156 seconds

- השאלתה מחזירה את השם הפרטי, שם המשפחה ואת המספר הכולל של בקשות לאירועים שנעשו על ידי אנשים עבור אירועים מסוג 'חתונה'. לאחר מכן הוא מקבץ את התוצאות, מסנן אנשים עם לפחות 2 בקשות לאירוע, וממין את התוצאה הסופית בסדר יורד על סמך המספר הכולל של הבקשות.

```

SELECT P.F_NAME, P.L_NAME, COUNT(*) AS TOTAL_HELP_REQUESTS
FROM PERSON P
INNER JOIN HELPED H ON P.ID = H.ID
INNER JOIN EVENT E ON E.CLIENTID = H.ID
INNER JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID
WHERE ET.TYPENAME = 'Wedding'
GROUP BY P.F_NAME, P.L_NAME
HAVING COUNT(*) >= 2
ORDER BY TOTAL_HELP_REQUESTS DESC;

```

	F_NAME	L_NAME	TOTAL_HELP_REQUESTS
1	Ty	Mirren	2

system@XE 1 row selected in 0.031 seconds

- מחזיר את המשמרת עם השכר הגבוה ביותר ואת המשמרת עם השכר הנמוך ביותר, יחד עם סוגי המשמרות התואמים להם.

```

SELECT S1.SHIFT_TYPE AS MAX_SALARY_SHIFT_TYPE, S1.SALARY AS MAX_SALARY,
S2.SHIFT_TYPE AS MIN_SALARY_SHIFT_TYPE, S2.SALARY AS MIN_SALARY
FROM SHIFTS S1
JOIN SHIFTS S2 ON S1.SHIFT_TYPE <> S2.SHIFT_TYPE
WHERE S1.SALARY = (SELECT MAX(SALARY) FROM SHIFTS)
AND S2.SALARY = (SELECT MIN(SALARY) FROM SHIFTS);

```

	MAX_SALARY_SHIFT_TYPE	MAX_SALARY	MIN_SALARY_SHIFT_TYPE	MIN_SALARY
1	Night shift	3987	Holiday shift	2199

system@XE 1 row selected in 0.062 seconds

שאלות UPDATE:

- שאלתה זו מעדכנת את העמודה BUDGET של טבלת EVENT עבור אירועים שהלקוחות שלהם עם רמת דחיפות גבוהה. זה מקטין את התקציב בעלות הכוללת של שירותי התזמורת והצלם הקשורים לאותם אירועים, מכיוון שהם מתנדבים עשינו COMMIT שעדכן את זה BUDGET לאירועים בהתאם בטבלת EVENT.

```

UPDATE EVENT
SET BUDGET = BUDGET - (SELECT SUM(orchestraPrice) FROM ORCHESTRA) + (SELECT SUM(photographPrice) FROM PHOTOGRAPH)
WHERE EVENTID IN (
  SELECT E.EVENTID
  FROM EVENT E
  INNER JOIN CLIENT C ON E.CLIENTID = C.CLIENTID
  INNER JOIN HELPED H ON C.CLIENTID = H.ID
  INNER JOIN ORCHESTRA O ON E.ORCHESTRAID = O.ORCHESTRAID
  INNER JOIN PHOTOGRAPH P ON E.PHOTOGRAPHID = P.PHOTOGRAPHID
  WHERE H.URGENCY = 3
);
COMMIT
  
```

- השאלתה מעדכנת את מספר האורחים עבור כל הלקוחות שיש להם אירועים עם תקציב גבוה מהתקציב הממוצע של כל האירועים:

```

UPDATE CLIENT
SET NumGuests = NumGuests + 50
WHERE CLIENTID IN (
  SELECT CLIENTID
  FROM EVENT
  WHERE BUDGET > (
    SELECT AVG(BUDGET)
    FROM EVENT
  )
);
COMMIT
  
```

```

UPDATE CLIENT
SET NumGuests = NumGuests + 50
WHERE CLIENTID IN (
  SELECT CLIENTID
  FROM EVENT
  WHERE BUDGET > (
    SELECT AVG(BUDGET)
    FROM EVENT
  )
);
COMMIT
  
```

17:20 system@XE 10 rows updated in 0 seconds

24:7 system@XE Done in 0.016 seconds

שאלות DELETE:

- השאלתה מוחקת את כל האנשים שאין להם שום תפקיד

```

SELECT ID
FROM PERSON
WHERE ID NOT IN (
  SELECT ID FROM VOLUNTEER
  UNION
  SELECT ID FROM HELPED
  UNION
  SELECT ID FROM SHIFT_MANAGER
  UNION
  SELECT ID FROM VOLUNTEER
  UNION
  SELECT CLIENTID FROM CLIENT
  UNION
  SELECT PHOTOGRAPHID FROM PHOTOGRAPH
  UNION
  SELECT ORCHESTRAID FROM ORCHESTRA
  UNION
  SELECT ID FROM MANAGER
);
ROLLBACK;
  
```

```

DELETE FROM PERSON
WHERE ID NOT IN (
  SELECT ID FROM VOLUNTEER
  UNION
  SELECT ID FROM HELPED
  UNION
  SELECT ID FROM SHIFT_MANAGER
  UNION
  SELECT ID FROM VOLUNTEER
  UNION
  SELECT CLIENTID FROM CLIENT
  UNION
  SELECT PHOTOGRAPHID FROM PHOTOGRAPH
  UNION
  SELECT ORCHESTRAID FROM ORCHESTRA
  UNION
  SELECT ID FROM MANAGER
);
  
```

מחקנו את הנתונים ואז עשינו ROLLBACK והנתונים חזרו

ואז עשינו שוב DELETE commit והנתונים נמחקו

```

DELETE FROM PERSON
WHERE ID NOT IN (
  SELECT ID FROM VOLUNTEER
  UNION
  SELECT ID FROM HELPED
  UNION
  SELECT ID FROM SHIFT_MANAGER
  UNION
  SELECT ID FROM VOLUNTEER
  UNION
  SELECT CLIENTID FROM CLIENT
  UNION
  SELECT PHOTOGRAPHID FROM PHOTOGRAPH
  UNION
  SELECT ORCHESTRAID FROM ORCHESTRA
  UNION
  SELECT ID FROM MANAGER
);
COMMIT;
  
```

Delete person Commit (no result set)

2. שאילתה זו מוחקת רשומות מטבלת EVENT שיש להן הפניות חסרות או לא חוקיות בטבלאות הקשורות CLIENT, EVENTTYPE, HALL, PHOTOGRAP, ORCHESTRA הוא משתמש בחיבורים שמאליים כדי לבדוק אם אין התאמות חוקיות בכל אחת מהטבלאות הללו ומסנן את המחיקה על סמך תנאים אלה.

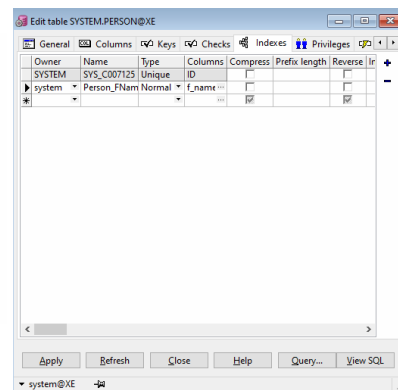
```
DELETE FROM EVENT
WHERE EVENTID IN (
  SELECT E.EVENTID
  FROM EVENT E
  LEFT JOIN CLIENT C ON C.CLIENTID = E.CLIENTID
  LEFT JOIN EVENTTYPE ET ON ET.TYPEID = E.TYPEID
  LEFT JOIN HALL H ON H.HALLID = E.HALLID
  LEFT JOIN PHOTOGRAPH P ON P.PHOTOGRAPHID = E.PHOTOGRAPHID
  LEFT JOIN ORCHESTRA O ON O.ORCHESTRAID = E.ORCHESTRAID
  WHERE C.CLIENTID IS NULL
  OR ET.TYPEID IS NULL
  OR H.HALLID IS NULL
  OR P.PHOTOGRAPHID IS NULL
  OR O.ORCHESTRAID IS NULL
);
```

19:8 system@XE 0 rows deleted in 0.016 seconds

INDEX

1. עשינו INDEX על PERSON על השדה NAME_F כיוון שיש בו שימוש רב ובטבלת PERSON יש מעל 200 אובייקטים כתיבת הindex:

```
-- Create/Recreate indexes
create index PERSON_FNAME_INDEX on PERSON (F_NAME)
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
);
```



הנה תמונה של שאילתה לפני ואחרי השינוי:

```
SELECT P.F_NAME, P.L_NAME, COUNT(*) AS TOTAL_HELP_REQUESTS
FROM PERSON P
INNER JOIN HELPED H ON P.ID = H.ID
INNER JOIN EVENT E ON E.CLIENTID = H.ID
INNER JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID
WHERE ET.TYPENAME = 'Wedding'
GROUP BY P.F_NAME, P.L_NAME
HAVING COUNT(*) >=2
ORDER BY TOTAL_HELP_REQUESTS DESC;
```

	F_NAME	L_NAME	TOTAL_HELP_REQUESTS
1	Ty	Mirren	2

24:35 system@XE 1 row selected in 0.094 seconds

```
SELECT P.F_NAME, P.L_NAME, COUNT(*) AS TOTAL_HELP_REQUESTS
FROM PERSON P
INNER JOIN HELPED H ON P.ID = H.ID
INNER JOIN EVENT E ON E.CLIENTID = H.ID
INNER JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID
WHERE ET.TYPENAME = 'Wedding'
GROUP BY P.F_NAME, P.L_NAME
HAVING COUNT(*) >=2
ORDER BY TOTAL_HELP_REQUESTS DESC;
```

	F_NAME	L_NAME	TOTAL_HELP_REQUESTS
1	Ty	Mirren	2

25:1 system@XE 1 row selected in 0.125 seconds

דוגמא נוספת:

SQL Window - query1.sql

```
SELECT ID, f_name
FROM volunteer Natural join person
WHERE ID IN (
  SELECT VO_ID
  FROM volunteering
  GROUP BY VO_ID
  HAVING COUNT(DISTINCT H_ID) >= 3
);
```

ID	F_NAME
223	Anjelica
262	Mint
163	Jonatha
32	Ashton
46	Madeleine
69	Eddie
212	Juliet
38	Mindy
48	Joaquin

4:15 system@XE 3 rows selected in 0.11 seconds (more...)

SQL Window - query.sql

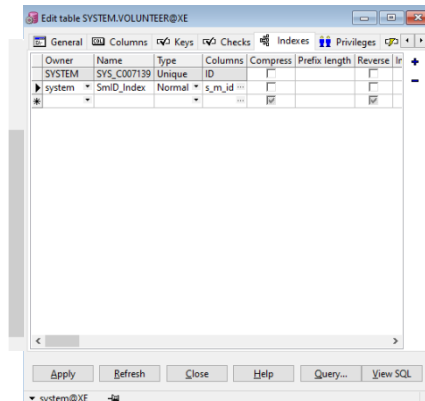
```
SELECT ID, f_name
FROM volunteer Natural join person
WHERE ID IN (
  SELECT VO_ID
  FROM volunteering
  GROUP BY VO_ID
  HAVING COUNT(DISTINCT H_ID) >= 3
);
```

ID	F_NAME
223	Anjelica
262	Mint
163	Jonatha
32	Ashton
46	Madeleine
69	Eddie
212	Juliet
38	Mindy
48	Joaquin

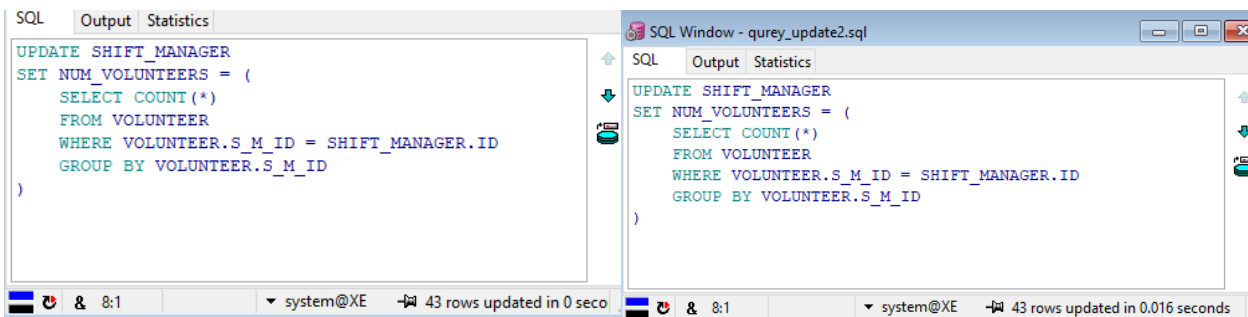
6:17 system@XE 17 rows selected in 0.171 seconds

2. עשינו INDEX על VOLUNTEER על השדה S_M_ID
כיוון שיש בו שימוש רב ובטבלת VOLUNTEER יש מעל 200 אובייקטים
כתיבת הindex:

```
-- Create/Recreate indexes
create index VOL_SMID_INDEX on VOLUNTEER (S_M_ID)
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
);
```

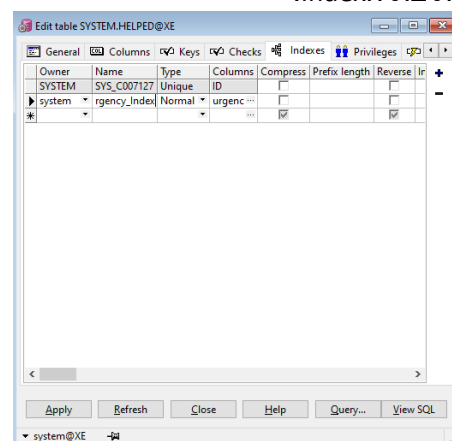


הנה תמונה של שאילתה לפני ואחרי השינוי:

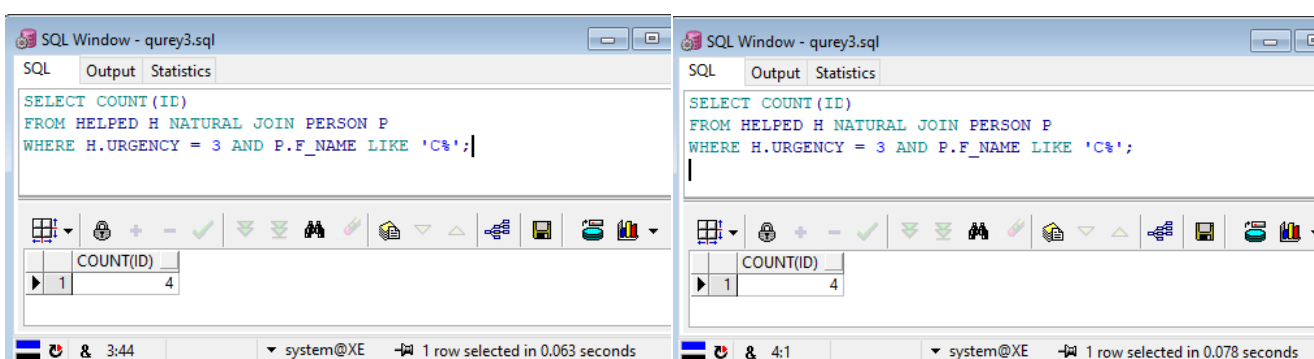


3. עשינו INDEX על HELPED על השדה URGENCY
כיוון שיש בו שימוש רב ובטבלת HELPED יש לו הרבה אובייקטים
כתיבת הindex:

```
-- Create/Recreate indexes
create index HELPED_URGENCY_INDEX on HELPED (URGENCY)
tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
);
```



הנה תמונה של שאילתה לפני ואחרי השינוי:



דוגמא נוספת:

SQL Window - query_update1.sql

SQL

Output

Statistics

```

UPDATE SHIFTS
SET SALARY = SALARY + 500, SHIFT_TYPE = 'On-call shift'
WHERE S_ID IN (
  SELECT S_ID FROM VOLUNTEERING
  WHERE VO_ID IN (
    SELECT ID FROM VOLUNTEER
    WHERE S_M_ID IN (
      SELECT ID FROM SHIFT_MANAGER
      WHERE NUM_VOLUNTEERS > 5
    )
  )
)
AND H_ID IN (
  SELECT ID FROM HELPED
  WHERE URGENCY = 3
)
);

```

13:26

system@XE

10 rows updated in 0.016 seconds

SQL Window - query_update1.sql

SQL

Output

Statistics

```

UPDATE SHIFTS
SET SALARY = SALARY + 500, SHIFT_TYPE = 'On-call shift'
WHERE S_ID IN (
  SELECT S_ID FROM VOLUNTEERING
  WHERE VO_ID IN (
    SELECT ID FROM VOLUNTEER
    WHERE S_M_ID IN (
      SELECT ID FROM SHIFT_MANAGER
      WHERE NUM_VOLUNTEERS > 5
    )
  )
)
AND H_ID IN (
  SELECT ID FROM HELPED
  WHERE URGENCY = 3
)
);

```

12:16

system@XE

10 rows updated in 0.031 seconds

טבלה של אחרי ולפני שינויי האינדקסים:

מספר השאילתה	לפני השינוי	אחרי השינוי
1	0.125	0.094
2	0.171	0.11
3	0.016	0
4	0.078	0.063
5	0.031	0.016

:VIEWS

1. שאילתת SELECT

טבלת "EventSummary" VIEW משלבת מידע מהטבלאות "Orchestra", "Client", "Event" ו-"Photograph" שאילתת ה-SELECT מחשבת את מספר הלקוחות הנבדלים עבור כל אירוע באמצעות COUNT(DISTINCT c.id). השאילתה גם מחשבת את המחיר הכולל של התזמורת ומחיר הצילום הכולל עבור כל אירוע באמצעות SUM(o.OrchestraPrice) ו-SUM(p.PhotographPrice) בהתאמה. לטבלת VIEW שתקבל יהיו עמודות: EventId, NumClients, TotalOrchestraPrice ו-TotalPhotographPrice.

השפעת השאילתה על טבלת VIEW היא לקריאה בלבד. הוא מאחזר את הנתונים המצטברים המאוחסנים בטבלת VIEW מבלי לשנות את הטבלאות הבסיסיות או את ה-VIEW עצמו. טבלת VIEW מספקת תצוגה מאוחדת של הנתונים ממספר טבלאות, ופונקציות הצבירה (COUNT ו-SUM) מאפשרות לך לנתח את המידע המסכם עבור כל אירוע.

SQL Window - CREATE VIEW EventSummary AS SELECT e.EventId, COUNT(DISTINCT c.ClientId) AS NumClients, SUM(o.OrchestraPrice) AS TotalOrchestraPrice, SUM(p.PhotographPrice) AS TotalPhotographPrice FROM Event e JOIN Client c ON e.ClientId = c.ClientId JOIN Orchestra o ON e.OrchestraId = o.OrchestraId JOIN Photograph p ON e.PhotographId = p.PhotographId GROUP BY e.EventId;

CREATE VIEW EventSummary AS

SELECT

e.EventId,
COUNT(DISTINCT c.ClientId) AS NumClients,
SUM(o.OrchestraPrice) AS TotalOrchestraPrice,
SUM(p.PhotographPrice) AS TotalPhotographPrice

FROM Event e

JOIN Client c ON e.ClientId = c.ClientId

JOIN Orchestra o ON e.OrchestraId = o.OrchestraId

JOIN Photograph p ON e.PhotographId = p.PhotographId

GROUP BY e.EventId;

SELECT *

FROM EventSummary

WHERE TotalOrchestraPrice > 15000;

EVENTID	NUMCLIENTS	TOTALORCHESTRAPRICE	TOTALPHOTOGRAPHPRICE
1	152	1	16068
2	180	1	15087
3	87	1	15690
4	100	1	16268
5	34	1	15182
6	44	1	15114

16:35

system@XE

30 rows selected in 0.156 seconds

זה תמונה של view עם שאילתת SELECT שמביאה את כל האירועים שמחיר התזמורת שלהם גדול מ-15000.

טבלת VIEW היא טבלה וירטואלית שנוצרת מהנתונים הבסיסיים. מכיוון שזו אינה טבלה בפועל, לא ניתן לעדכן או למחוק ישירות את הנתונים בטבלת VIEW.

במקרה שלנו יש לנו את העמודות האלה NumClients, TotalOrchestraPrice, TotalPhotographPrice

שהם פונקציות קיבוץ שאי אפשר לעשות עליהם עידכון או מחיקה.

2. שאלתת SELECT

בשאלתה זו, אנו בוחרים את מזהה המתנדב, שם פרטי, שם משפחה, שעת התחלה ושעת סיום מטבלת התצוגה "משמרות מתנדבים". החלנו גם תנאי באמצעות סעיף WHERE כדי לסנן את התוצאות. רק המשמרות שמתחילות לאחר או בשעה 8:00 בבוקר יכללו במערך התוצאות.

SQL Window - CREATE VIEW volunteer_shifts AS SELECT v.id AS volunteer_id, p.f_name, p.l_name, s.start_time, s...

SQL Output Statistics

```
SELECT v.id AS volunteer_id, p.f_name, p.l_name, s.start_time, s.finish_time
FROM volunteer v
JOIN person p ON v.id = p.id
JOIN shifts s ON v.id = s.s_id;

SELECT volunteer_id, f_name, l_name, start_time, finish_time
FROM volunteer_shifts
WHERE start_time >= '08:00';
```

	VOLUNTEER_ID	F_NAME	L_NAME	START_TIME	FINISH_TIME
1	23	Curt	Seigny	10:00	18:00
2	26	Oro	Woodward	10:00	18:00
3	27	Robbie	Mantegna	10:00	18:00
4	29	Ty	Mirren	10:00	18:00
5	31	Denise	Griffith	12:00	20:30
6	32	Ashton	Flanery	12:00	20:30

6 rows selected in 0.078 seconds (more...)

שאלתת UPDATE

בשאלתת עדכון זו, אנו משנים את שעת ההתחלה ושעת הסיום של משמרת עבור מתנדב ספציפי. אנו משתמשים בטבלת התצוגות "volunteer_shifts" ומשנים את השעת ההתחלה ל"07:00" וזמן הסיום ל"15:00" לכל הid שקטנים מ20.

SQL Window - CREATE VIEW volunteer_shifts AS SELECT v.id AS volunteer_id, p.f_name, p.l_name, s.start_time, s...

SQL Output Statistics

```
CREATE VIEW volunteer_shifts AS
SELECT v.id AS volunteer_id, p.f_name, p.l_name, s.start_time, s.finish_time
FROM volunteer v
JOIN person p ON v.id = p.id
JOIN shifts s ON v.id = s.s_id;

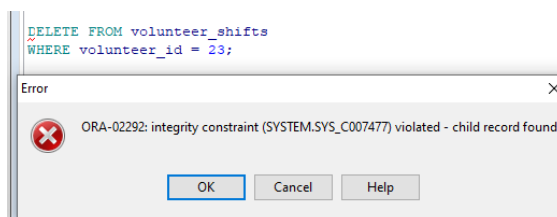
UPDATE volunteer_shifts
SET start_time = '07:00', finish_time = '15:00'
WHERE volunteer_id < 20;

SELECT *
```

7 rows updated in 0 seconds

שאלתת DELETE

אי אפשר לעשות שאלתת DELETE מכיוון שהטבלה מקושרת לטבלאות אחרות



Grant revoke

יצירת משתמש חדש והרשאות הגישה שלו:

```
SQL Window - create user deena IDENTIFIED BY 123 default tablespace SYS
SQL Output Statistics
default tablespace SYSTEM
profile DEFAULT
PASSWORD expire;

grant create session to deena;
grant create table to deena;

Alter user deena quota unlimited on System

grant select on Person to deena;
```

View table SYSTEM.PERSON@XE

Grantee	Select	Insert	Update	Delete	References	Alter	Index
DEENA	Yes						

נתנו למשתמש החדש הרשאת גישה לפקודת SELECT ולכן היא עבדה

SQL Window - select * from system.person;

```
SQL Output Statistics
select * from system.person;
```

ID	F_NAME	L_NAME	ADDRESS	PHONE_NUMBER
1	56 First	Shannon	30 Burstyn Blvd	2902
2	57 Clive	Dunst	49 Long Island	1504
3	58 Carlene	Tillis	532 Carla Stree	2788
4	59 Carolyn	Latifah	64 University S	4088
5	60 Alicia	Lineback	8 Colon Road	8238
6	61 Domingo	Balaban	98 Humberto	4594
7	62 Juliet	Reilly	13 Vanessa	2173
8	63 Patrick	Crowell	98 Todd Drive	5924
9	64 Colin	Balk	396 Statham Roa	3402

deena@XE 492 rows selected in 0.469 seconds

וניסינו לעשות עדכון והוא לא נתן כי אין לו הרשאת גישה לזה

SQL Window - update system.person SET f_name = 'aaa';

```
SQL Output Statistics
update system.person
SET f_name = 'aaa';
```

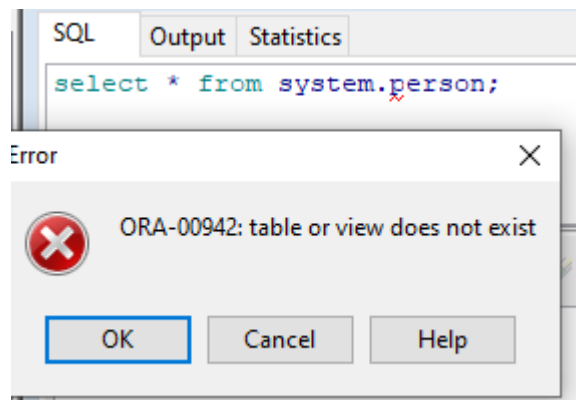
Error

ORA-01031: insufficient privileges

OK Cancel Help

ואז עשינו REVOKE והורדנו לו את הרשאת הגישה

```
--revoke
revoke select on person from deena;
```

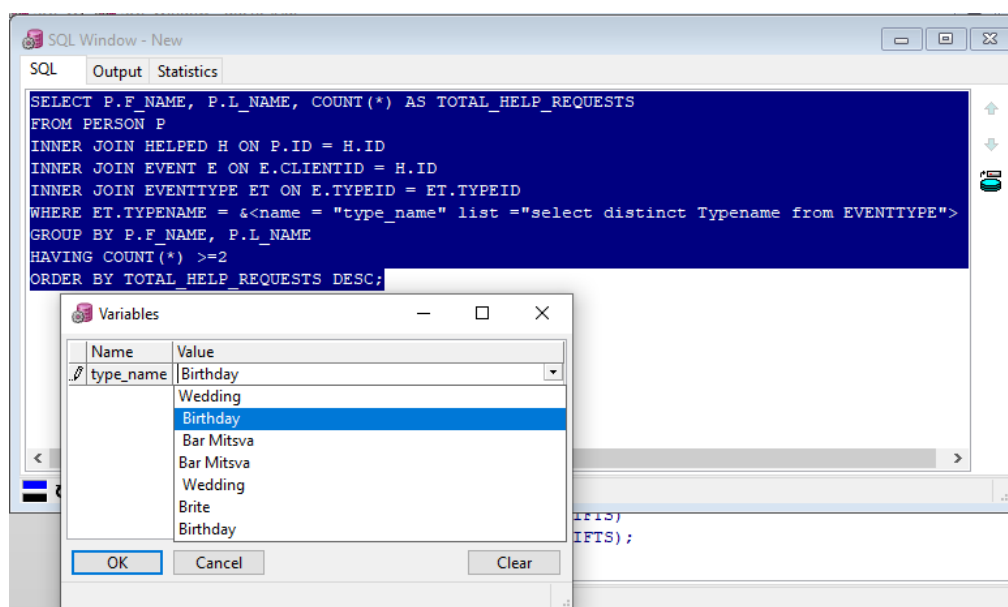


שאלות עם פרמטרים:

שאלתה 1

השאלתה מחזירה את השם הפרטי, שם המשפחה ואת המספר הכולל של בקשות לאירועים שנעשו על ידי אנשים עבור אירועים מהסוג שהמשתנה יבחר. לאחר מכן הוא מקבץ את התוצאות, מסנן אנשים עם לפחות 2 בקשות לאירוע, וממין את התוצאה הסופית בסדר יורד על סמך המספר הכולל של הבקשות.

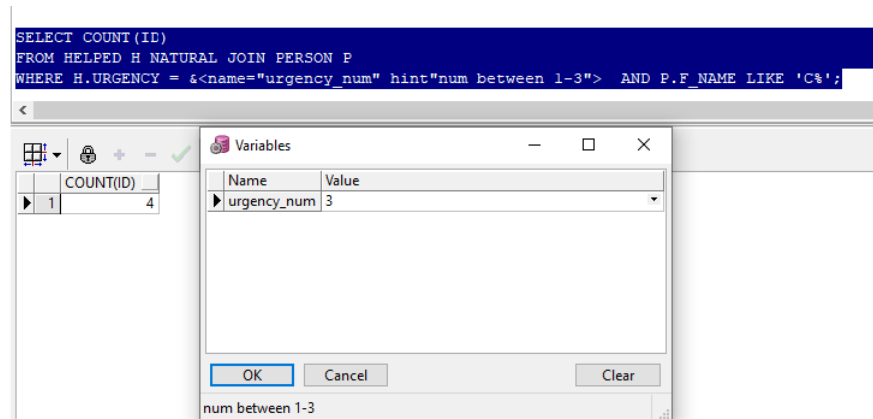
עשינו פרמטר מסוג LIST שבו הפרמטר שנכנס הוא מסוג STRING



שאלתה 2

השאלתה מחזירה את הכמות של הנצרכים שהדחיפות שלהם היא לפי הכנסת המשתמש והשם שלהם מתחיל באות C.

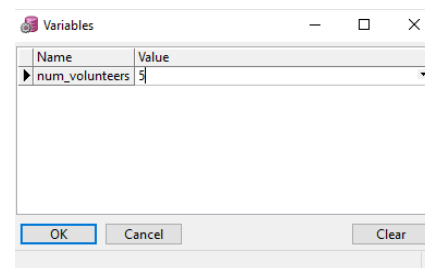
עשינו גם HINT שאומר למשתמש להכניס מספר בין 1 ל-3.



שאלתה 3:

שאלתה זו מעדכנת את סוג השכר והמשמרת עבור כל המשמרות המשויות למתנדבים שעובדים תחת מנהלי משמרת המנהלים יותר מ-5 מתנדבים, ומוקצות לסייע לאלה עם צרכים דחופים גבוהים.

```
UPDATE SHIFTS
SET SALARY = SALARY + 500, SHIFT_TYPE = 'On-call shift'
WHERE S_ID IN (
  SELECT S_ID FROM VOLUNTEERING
  WHERE VO_ID IN (
    SELECT ID FROM VOLUNTEER
    WHERE S_M_ID IN (
      SELECT ID FROM SHIFT_MANAGER
      WHERE NUM_VOLUNTEERS > &<name="num_volunteers" default=5 type="integer">
    )
  )
)
AND H_ID IN (
  SELECT ID FROM HELPED
  WHERE URGENCY = 3
)
);
```

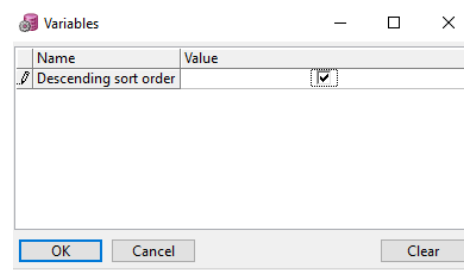
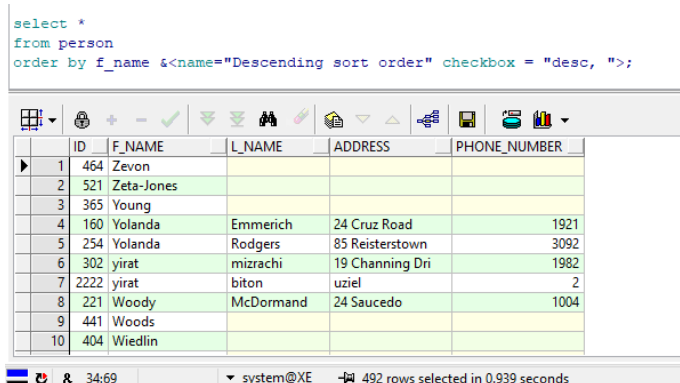


30:3 system@XE 10 rows updated in 0 seconds

שאלתה 4:

שאלתה זו מבעת מיון לפי בחירת המשתמש

אם זה מיון עולה או יורד.



דוחות וגרפים:

1. השאילתה מוצאת את פרטי האירוע, כולל שם סוג האירוע ותקציב, עבור אירועים בעלי תקציב קטן מהתקציב הממוצע של כל האירועים.

דוח:

עשינו דוח על פרטי האירוע- התקציב, סוג האירוע, מספר מזהה.

Report Window - SELECT E.EVENTID, ET.TYPENAME, E.BUDGET FROM EVENT E JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID W ...

Item	Description	Style	Header	Align	Format	Break	Sum
Report Title	EVENTS	Blue Table		Center			
Variables			Left				
Tabular Tables			Left				
Form Tables			Left				
Default Field							
EVENTID		Blue Head			0	Break + Header	
TYPENAME		Blue Head					
BUDGET		Blue Head					Sum per Break & Report

Eventid	Typename	Budget
181	Bar Mitsva	34053
163	Bar Mitsva	22241
		56294

Eventid	Typename	Budget
85	Bar Mitsva	34518
216	Bar Mitsva	30418
128	Bar Mitsva	39897

Report Window - SELECT E.EVENTID, ET.TYPENAME, E.BUDGET FROM EVENT E JOIN EVENTTYPE ET ON E.TYPEID = ET.TYPEID W ...

Eventid	Typename	Budget
181	Bar Mitsva	34053
163	Bar Mitsva	22241
		56294

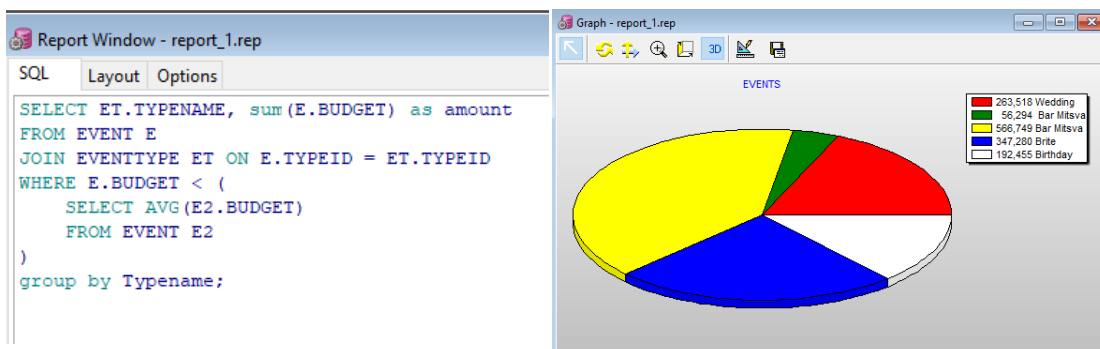
Eventid	Typename	Budget
85	Bar Mitsva	34518
216	Bar Mitsva	30418
128	Bar Mitsva	39897

גרף:

עשינו גרף שמראה את סוגי האירועים לפי התקציב שלהם

X – סוגי האירועים

Y – סכום התקציבים של הסוגי האירועים האלה



2. השאילתה מחזירה את סוג המשמרת והמשכורת שמקבלים עליה לפי הסדר של סוג המשמרות

דוח:

עשינו דוח על פרטי המשמרות- סוג המשמרת והשכר עליה.

Report Window - SELECT S.SHIFT_TYPE, S.SALARY FROM SHIFTS S ORDER BY S.SHIFT_TYPE;

Item	Description	Style	Header	Align	Format	Break	Sum
Report Title	SALARY OF SHIFTS						
Variables							
Tabular Tables							
Form Tables							
Default Field							
SHIFT_TYPE	Type	Custom...	Blue Head	Left	0	Break + Header	
SALARY			Blue Head	Left	0		Sum p...

Type	Salary
Day shift	3748
	3968
	3364
	2561
	3299
	2596
	2674
	2622
	2447

Report Window - SELECT S.SHIFT_TYPE, S.SALARY FROM SHIFTS S ORDER BY S.SHIFT_TYPE;

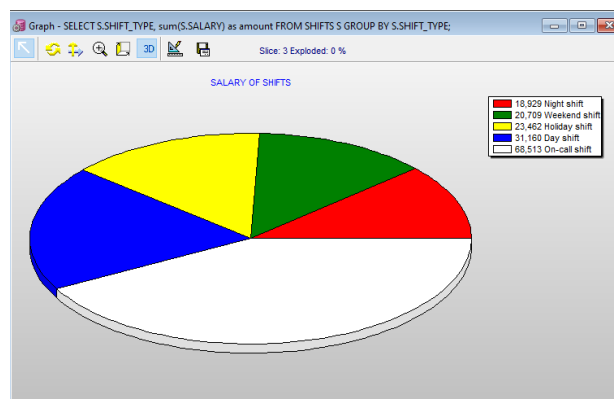
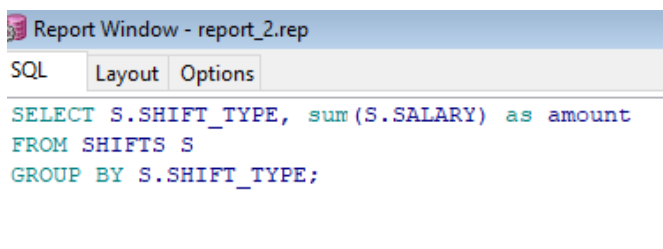
Type	Salary
Day shift	3748
	3968
	3364
	2561
	3299
	2596
	2674
	2622
	2447

גרף:

עשינו גרף שמראה את סוגי המשמרות לפי המשכורות שלהם

X – סוגי המשמרות

Y – סכום המשכורות של המשמרות האלה

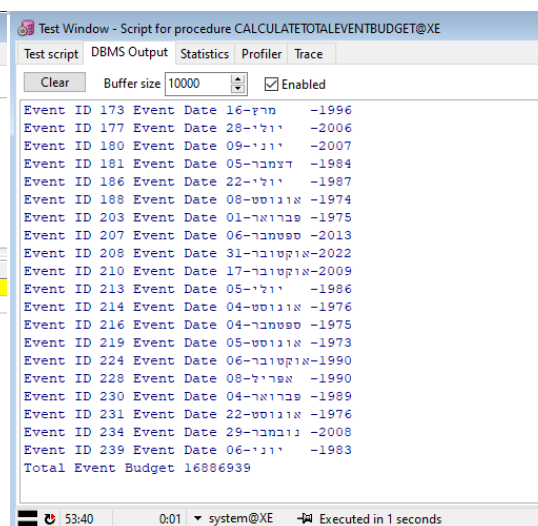
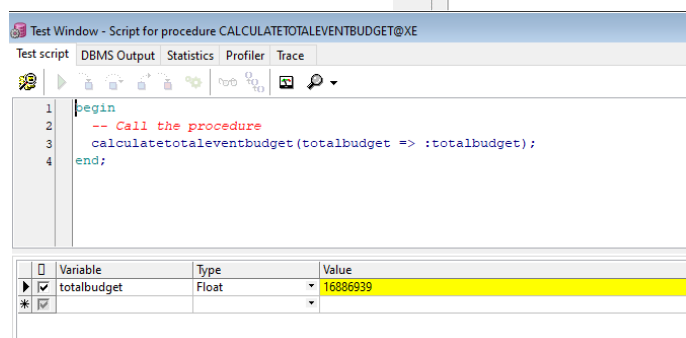
שלב 9:

פרוצדורות:

1. מחשב את תקציב האירוע הכולל:

הליך זה מחשב את התקציב הכולל של אירוע על סמך מזהה האירוע שסופק. הוא משתמש בסמן בשם cEventBudget כדי להביא את תקציב האירוע מטבלת EVENT... הלולאה חוזרת דרך הסמן וצוברת את ערכי התקציב כדי לחשב את התקציב הכולל. התקציב הכולל מוקצה לפרמטר OUT. totalBudget האירוע לא נמצא, תוצג הודעה. טיפול בחריגים מיושם כדי לטפל בשגיאות שעלולות להתרחש במהלך תהליך החישוב.

```
1 CREATE OR REPLACE PROCEDURE CalculateTotalEventBudget
2 (totalBudget OUT NUMBER)
3 eventBudget EVENT.BUDGET%TYPE;
4 eventId EVENT.Eventid%TYPE;
5 eventDate EVENT.Eventdate%TYPE;
6 cursor cEventBudget IS
7 SELECT BUDGET,Eventid,EventDate
8 FROM EVENT;
9 BEGIN
10 totalBudget := 0;
11 OPEN cEventBudget;
12 LOOP
13 FETCH cEventBudget INTO eventBudget,eventId,eventDate;
14 EXIT WHEN cEventBudget%NOTFOUND;
15 totalBudget := totalBudget + eventBudget;
16 DBMS_OUTPUT.PUT_LINE('Event ID ' || eventId || ' Event Date ' || eventDate);
17 END LOOP;
18 DBMS_OUTPUT.PUT_LINE('Total Event Budget ' || totalBudget );
19 CLOSE cEventBudget;
20 EXCEPTION
21 WHEN NO_DATA_FOUND THEN
22 DBMS_OUTPUT.PUT_LINE('Event with ID ' || eventId || ' does not exist.');
```



2. מחשב את המספר הכולל של אורחים עבור סוג אירוע ספציפי:

בהליך זה:

הפרמטר pEventType משמש לציון סוג האירוע שעבורו יחושב מספר האורחים.

הפרמטר pTotalGuests הוא פרמטר פלט שיאחסן את המספר הכולל המחושב של האורחים.

המשתנים vEventTypeId, vHallId, vEventId ו-vNumGuests משמשים כדי להביא את הערכים מהסמן.

סמן cEvents מאחזר את מזהה סוג האירוע, מזהה האולם, מזהה האירוע ומספר האורחים מהטבלאות EVENT, EVENTTYPE, HALL ו-CLIENT בהתבסס על סוג האירוע שצוין.

ה-pTotalGuests מוגדר תחילה ל-0.

הסמן נפתח, ומתחילה לולאה כדי להביא כל שורת נתונים.

בתוך הלולאה, מספר האורחים מתווסף ל-pTotalGuests.

מזהה האירוע ומספר האורחים מוצגים באמצעות DBMS_OUTPUT.PUT_LINE.

לאחר שלפית כל השורות, המספר הכולל של האורחים עבור סוג האירוע שצוין יוצג.

הסמן סגור.

טיפול בחריגים נכלל כדי לטפל בתרחישים שבהם לא נמצאו אירועים מהסוג שצוין או אם מתרחשת שגיאה אחרת.

```

1 CREATE OR REPLACE PROCEDURE CalculateEventTypeGuests
2   (pEventType IN EVENTTYPE.TYPENAME%TYPE, pTotalGuests OUT NUMBER)
3   IS
4   vEventTypeId EVENTTYPE.TYPEID%TYPE;
5   vHallId HALL.HALLID%TYPE;
6   vEventId EVENT.EVENTID%TYPE;
7   vNumGuests CLIENT.NUMGUESTS%TYPE;
8   CURSOR cEvents IS
9     SELECT et.TYPEID, h.HALLID, e.EVENTID, c.NUMGUESTS
10    FROM EVENT e
11   JOIN EVENTTYPE et ON e.TYPEID = et.TYPEID
12   JOIN HALL h ON e.HALLID = h.HALLID
13   JOIN CLIENT c ON e.CLIENTID = c.CLIENTID
14   WHERE et.TYPENAME = pEventType;
15 BEGIN
16   pTotalGuests := 0;
17   OPEN cEvents;
18   LOOP
19     FETCH cEvents INTO vEventTypeId, vHallId, vEventId, vNumGuests;
20     EXIT WHEN cEvents%NOTFOUND;
21     pTotalGuests := pTotalGuests + vNumGuests;
22     DBMS_OUTPUT.PUT_LINE('Event ID: ' || vEventId || 'Number of Guests: ' || vNumGuests);
23   END LOOP;
24   DBMS_OUTPUT.PUT_LINE('Total Guests for ' || pEventType || ' Events: ' || pTotalGuests);
25   CLOSE cEvents;
26 EXCEPTION
27   WHEN NO_DATA_FOUND THEN
28     DBMS_OUTPUT.PUT_LINE('No ' || pEventType || ' events found.');
```

Test Window - Script for procedure CALCULATEEVENTTYPEGUESTS@XE

Test script DBMS Output Statistics Profiler Trace

```

1 begin
2   -- Call the procedure
3   calculateeventtypeguests(peventtype => :peventtype,
4                             ptotalguests => :ptotalguests);
5 end;
```

Variable	Type	Value
peventtype	String	Wedding
ptotalguests	Float	7575

Test Window - Script for procedure CALCULATEEVENTTYPEGUESTS@XE

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 Enabled

```

Event ID: 188Number of Guests: 400
Event ID: 73Number of Guests: 400
Event ID: 156Number of Guests: 416
Event ID: 239Number of Guests: 318
Event ID: 114Number of Guests: 495
Event ID: 207Number of Guests: 389
Event ID: 82Number of Guests: 405
Event ID: 165Number of Guests: 661
Event ID: 17Number of Guests: 527
Event ID: 52Number of Guests: 749
Event ID: 107Number of Guests: 403
Event ID: 154Number of Guests: 552
Event ID: 230Number of Guests: 730
Total Guests for Wedding Events: 7575
```

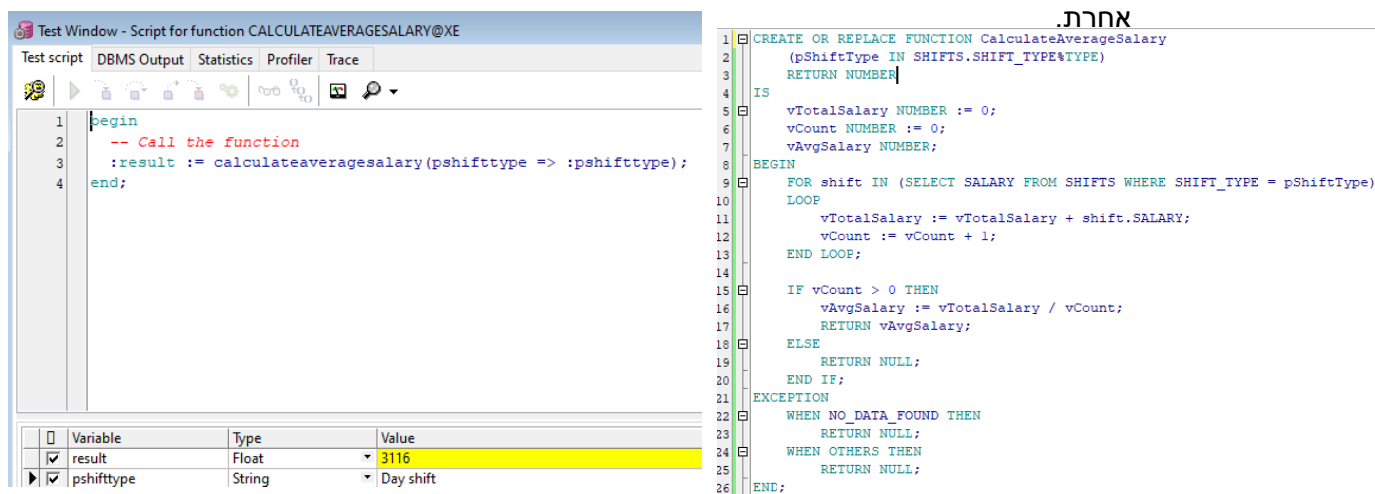
12:13 0:01 system@XE Executed in 1.406 seconds

פונקציות:

1. פונקציה שמחשבת את השכר הממוצע עבור סוג משמרת ספציפי:
בפונקציה הזו:

הפרמטר pShiftType משמש לציון סוג המשמרת שעבורו יחושב השכר הממוצע.
המשתנה vTotalSalary מאחסן את סכום המשכורות עבור סוג המשמרת שצוין.
המשתנה vCount עוקב אחר מספר המשמרות שנמצאו עבור הסוג שצוין.
המשתנה vAvgSalary מאחסן את השכר הממוצע המחושב.
לולאה משמשת לחזרה על המשמרות עם הסוג שצוין ולצבור את המשכורות.
אם נמצאו משמרות השכר הממוצע מחושב ומוחזר.
אם לא נמצאו שינויים או מתרחשת שגיאה, NULL מוחזר.

טיפול בחריגים נכלל כדי לטפל בתרחישים שבהם לא נמצא נתונים או אם מתרחשת שגיאה אחרת.



The screenshot shows a SQL development environment with a test script window and a results window. The test script is as follows:

```

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

The results window shows the following data:

Variable	Type	Value
result	Float	3116
pshifttype	String	Day shift

2. פונקציה הקובעת את התפקיד של כל אדם (מתנדב, עזר, מנהל, מנהל משמרת או אף אחד):
בפונקציה הזו:
הפונקציה לוקחת פרמטר pPersonId מסוג PERSON.ID כדי לציין את המזהה של האדם שעבורו ייקבע התפקיד.
משתנה vRole מסוג VARCHAR2(20) משמש לאחסון התפקיד שנקבע.
הפונקציה בודקת אם האדם הוא מתנדב על ידי שאילתה בטבלת VOLUNTEER בהתבסס על ה-VO_ID.
אם נמצא התאמה, התפקיד מוגדר ל'התנדבות' ומוחזר.
אם לא נמצא התאמה, הפונקציה ממשיכה לבדוק אם האדם נעזר על ידי שאילתה בטבלת HELPED בהתבסס על ה-H_ID.
אם נמצא התאמה, התפקיד מוגדר ל'עזרה' ומוחזר.
לאחר מכן הפונקציה בודקת אם האדם הוא מנהל משמרת על ידי שאילתה בטבלת SHIFT_MANAGER בהתבסס על המזהה.
אם נמצא התאמה, התפקיד מוגדר ל'מנהל משמרות' ומוחזר.
לאחר מכן, הפונקציה בודקת אם האדם הוא מנהל על ידי שאילתה בטבלת MANAGER בהתבסס על המזהה.
אם נמצא התאמה, התפקיד מוגדר ל'מנהל' ומוחזר.
אם אף אחד מהתפקידים לעיל לא נמצא, הפונקציה מחזירה 'ללא'.
טיפול בחריגים נכלל כדי לטפל בתרחישים שבהם לא נמצא נתונים או אם מתרחשת שגיאה אחרת.

Test Window - Script for function DETERMINEPERSONROLE@XE

Test script DBMS Output Statistics Profiler Trace

```

1 begin
2   -- Call the function
3   :result := determinepersonrole(ppersonid => :ppersonid);
4 end;

```

Variable	Type	Value
result	String	Volunteer
ppersonid	Integer	56

```

1 CREATE OR REPLACE FUNCTION DeterminePersonRole(pPersonId IN PERSON.ID%TYPE)
2   RETURN VARCHAR2
3 IS
4   vRole VARCHAR2(20);
5 BEGIN
6   -- Check if the person is a volunteer
7   SELECT 'Volunteer' INTO vRole FROM VOLUNTEER WHERE ID = pPersonId;
8
9   IF vRole IS NOT NULL THEN
10    RETURN vRole;
11  END IF;
12
13  -- Check if the person is a helped
14  SELECT 'Helped' INTO vRole FROM HELPED WHERE ID = pPersonId;
15
16  IF vRole IS NOT NULL THEN
17    RETURN vRole;
18  END IF;
19
20  -- Check if the person is a shift manager
21  SELECT 'Shift Manager' INTO vRole FROM SHIFT_MANAGER WHERE ID = pPersonId;
22
23  IF vRole IS NOT NULL THEN
24    RETURN vRole;
25  END IF;
26
27  -- Check if the person is a manager
28  SELECT 'Manager' INTO vRole FROM MANAGER WHERE ID = pPersonId;
29
30  IF vRole IS NOT NULL THEN
31    RETURN vRole;
32  END IF;
33
34  -- If none of the above roles are found, return 'None'
35  RETURN 'None';
36 EXCEPTION
37  WHEN NO_DATA_FOUND THEN
38    RETURN 'None';
39  WHEN OTHERS THEN
40    RETURN 'Error';
41 END;

```

תוכניות:

1. תוכנית שמשתמשת בפונקציה הראשונה שמחזירה את ממוצע המשכורות של סוג משמרת ספציפי:

Test Window - New

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 Enabled

Average Salary: 3116

Test Window - New

Test script DBMS Output Statistics Profiler Trace

```

1 DECLARE
2   avgSalary NUMBER;
3 BEGIN
4   avgSalary := CalculateAverageSalary('Day shift');
5   DBMS_OUTPUT.PUT_LINE('Average Salary: ' || avgSalary);
6 END;

```

2. תוכנית שמשתמשת בפונקציה הראשונה שקובעת את התפקיד של כל אדם:

Test Window - New

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 Enabled

```

Person ID: 1, Role: None
Person ID: 2, Role: None
Person ID: 3, Role: None
Person ID: 4, Role: None
Person ID: 5, Role: None
Person ID: 6, Role: None
Person ID: 7, Role: None
Person ID: 8, Role: None
Person ID: 9, Role: None
Person ID: 10, Role: Volunteer
Person ID: 11, Role: Volunteer
Person ID: 12, Role: Volunteer
Person ID: 13, Role: Volunteer
Person ID: 14, Role: Volunteer
Person ID: 15, Role: None
Person ID: 16, Role: Volunteer
Person ID: 17, Role: Volunteer
Person ID: 18, Role: None
Person ID: 19, Role: None
Person ID: 20, Role: None
Person ID: 21, Role: None
Person ID: 22, Role: None
Person ID: 23, Role: Volunteer
Person ID: 24, Role: None
Person ID: 25, Role: None
Person ID: 26, Role: Volunteer
Person ID: 27, Role: Volunteer
Person ID: 28, Role: None
Person ID: 29, Role: Volunteer

```

27:21 0:01 system@XE Executed in 1.093 seconds

Test Window - New

Test script DBMS Output Statistics Profiler Trace

```

1 DECLARE
2   personRole VARCHAR2(20);
3 BEGIN
4   FOR i IN (
5     SELECT ID
6     FROM PERSON
7     WHERE ROWNUM <= 50
8   ) LOOP
9     BEGIN
10      personRole := DeterminePersonRole(i.ID);
11      DBMS_OUTPUT.PUT_LINE('Person ID: ' || i.ID || ', Role: ' || personRole);
12    EXCEPTION
13      WHEN NO_DATA_FOUND THEN
14        DBMS_OUTPUT.PUT_LINE('No data found for Person ID: ' || i.ID);
15      WHEN OTHERS THEN
16        DBMS_OUTPUT.PUT_LINE('An error occurred for Person ID: ' || i.ID);
17    END;
18  END LOOP;
19 END;

```


טריגרים:

1. טריגר שמעדכן את התקציב של ישות EVENT בכל פעם ש-NumGuests של ישות CLIENT המשויכת מתעדכנים:

```

1 CREATE OR REPLACE TRIGGER update_event_budget_trigger
2 AFTER UPDATE OF NumGuests ON CLIENT
3 FOR EACH ROW
4 BEGIN
5     UPDATE EVENT
6     SET BUDGET = BUDGET + (:NEW.NumGuests) *
7     ((SELECT orchestraPrice FROM ORCHESTRA WHERE ORCHESTRAID = EVENT.ORCHESTRAID)+
8     (SELECT PhotographPrice FROM photograph WHERE photographID = EVENT.photographID))
9     WHERE EVENTID = (SELECT EVENTID FROM HELPED WHERE ID = :NEW.CLIENTID);
10 END;
  
```

השורה של העדכון:

```

UPDATE CLIENT SET NumGuests = 300 where clientId = 173;
  
```

לפני:

	EVENTID	BUDGET	EVENTDATE	HALLID	TYPEID	CLIENTID	PHOTOGRAPHID	ORCHESTRAID
1	17	31426790	30/12/2013	146	394	173	353	121
2	23	36553489	26/09/2007	73	584	171	353	218
3	31	35843376	17/10/1995	41	479	25	355	13
4	34	39505785	17/08/1986	65	515	205	367	63

אחרי:

	EVENTID	BUDGET	EVENTDATE	HALLID	TYPEID	CLIENTID	PHOTOGRAPHID	ORCHESTRAID
1	17	25661	30/12/2013	146	394	173	353	121
2	23	1101069	26/09/2007	73	584	171	353	218
3	31	39871	17/10/1995	41	479	25	355	13

2. בטריגר זה, לאחר הוספת שורה חדשה לטבלת EVENT, מאחזרת את המזהה של HALL המשויך מהעמודה HALLID לאחר מכן הוא מחשב את המספר המרבי של אורחים על ידי מציאת הערך הגבוה ביותר של NumGuests מבין כל הלקוחות הקשורים לאותו אירוע. לבסוף, הוא מעדכן את העמודה MAXGUESTS של ישות HALL המקבילה עם המספר המרבי המחושב של אורחים.

```

Program Window - Edit source of trigger SYSTEM.HALL_MAX_GUESTS@XE
hall_max_guests

Block Code section Statement
1 CREATE OR REPLACE TRIGGER hall_max_guests
2 AFTER INSERT ON EVENT
3 FOR EACH ROW
4 DECLARE
5     v_hall_id HALL.HALLID%TYPE;
6     v_max_guests NUMBER;
7 BEGIN
8     v_hall_id := :NEW.HALLID;
9
10    SELECT MAX (NumGuests)
11    INTO v_max_guests
12    FROM CLIENT
13    WHERE CLIENTID IN (
14        SELECT ID
15        FROM HELPED
16    );
17
18    UPDATE HALL
19    SET MAXGUESTS = v_max_guests
20    WHERE HALLID = v_hall_id;
21 END;

```

השורה שהרצנו:

```

INSERT INTO EVENT (EVENTID,BUDGET,EVENTDATE,HALLID,TYPEID,CLIENTID,PHOTOGRAPHID,ORCHESTRAID)
values(1,5000,(SELECT EVENTDATE FROM EVENT WHERE EVENTID =17),146,394,173,353,121);

```

לפני:

SQL Window - Query data of table HALL@XE

SQL Output Statistics

select * from HALL t

	HALLID	MAXGUESTS	HALLTYPE	AREA
139	140	661	gan Irouim	Tel aviv
140	141	631	oulam	Tel aviv
141	142	308	gan Irouim	Netanya
142	143	432	gan Irouim	Jerusalem
143	144	500	oulam	Netanya
144	145	305	gan Irouim	Petach Tikva
145	146	780	oulam	Cholone
146	147	333	oulam	Cholone
147	148	378	oulam	Cholone
148	149	464	gan Irouim	Petach Tikva

אחרי:

SQL Window - select * from HALL t

SQL Output Statistics

select * from HALL t

	HALLID	MAXGUESTS	HALLTYPE	AREA
142	143	432	gan Irouim	Jerusalem
143	144	500	oulam	Netanya
144	145	305	gan Irouim	Petach Tikva
145	146	780	oulam	Cholone
146	147	333	oulam	Cholone
147	148	378	oulam	Cholone
148	149	464	gan Irouim	Petach Tikva
149	150	656	oulam	Cholone
150	151	597	gan Irouim	Jerusalem
151	152	423	oulam	Netanya
152	153	570	oulam	Petach Tikva