

**מיני פרויקט בבסיסי נתונים**

**תהילה טיישר / רינת מועלם**

**319053039 / 208316232**

**מאי 2020**

בס"ד

תוכן עניינים:

[שלב מס' 1 4](#_Toc43685714)

[תיאור מילולי של הארגון 4](#_Toc43685715)

[תרשים ERD 4](#_Toc43685716)

[מילון 5](#_Toc43685717)

[הטבלאות המתקבלות 6](#_Toc43685718)

[בדיקה והסבר שכל הטבלאות מנורמלות ברמת 3CNF 6](#_Toc43685719)

[תרשים DSD 7](#_Toc43685720)

[מעבר מERD לטבלאות 7](#_Toc43685721)

[יצירת טבלאות 8](#_Toc43685722)

[אכלוס טבלאות 10](#_Toc43685723)

[**פקודת insert 10**](#_Toc43685724)

[**קליטה מקובץ TXT 11**](#_Toc43685726)

[**ODBC Importer 14**](#_Toc43685727)

[תרשים diagram windows 17](#_Toc43685729)

[**פקודת datagenerator 18**](#_Toc43685730)

[שלב מס' 2 21](#_Toc43685731)

[אלגוריתם שיחזור של ERD 21](#_Toc43685732)

[תרשים ERD של קרויז 21](#_Toc43685733)

[תרשים ERD משותף 22](#_Toc43685734)

[תרשים diagram windows משותף 23](#_Toc43685736)

[תיעוד החיבור 24](#_Toc43685737)

[שלב מס' 3 25](#_Toc43685738)

[פקודות select 25](#_Toc43685739)

[פקודות update 29](#_Toc43685740)

[פקודות delete 31](#_Toc43685741)

[פקודות commit וrollback 33](#_Toc43685742)

[**commit 33**](#_Toc43685743)

[**rollback 34**](#_Toc43685745)

[אינדקסים 35](#_Toc43685746)

[אילוצים 38](#_Toc43685747)

[יצירת משתמש והרשאות גישה 40](#_Toc43685748)

[Views 43](#_Toc43685749)

[שלב מס' 4 46](#_Toc43685750)

[שאילתות עם פרמטרים 46](#_Toc43685751)

[דוחות 50](#_Toc43685752)

[**דוח 1: 50**](#_Toc43685753)

[דוח 2: 51](#_Toc43685754)

[גרפים 52](#_Toc43685755)

[**גרף עבור דוח 1: 52**](#_Toc43685756)

[**גרף עבור דוח 2: 52**](#_Toc43685757)

[פונקציות 53](#_Toc43685758)

[פרוצדורות 56](#_Toc43685759)

[TRIGGERS 60](#_Toc43685761)

שלב מס' 1

תיאור מילולי של הארגון

בפרויקט נבנה בסיס נתונים לאגף טיולים בחברת תיירות.

המדריכים עובדים בחברה, לכל עובד יש פרטים אישיים כגון: ת.ז., שם, עיר, ניסיון.

המסלול נמצא באזור מסויים, לכל מסלול יש מידע כגון: מספר מזהה של המסלול ושם המסלול, רמת קושי, סוג המסלול, הזמן המשוער הבילוי במסלול ואורך המסלול.

לחברה יש לקוחות, לכל לקוח יש ת.ז., שם, תאריך לידה. קבוצה נוצרת ע"י מספר לקוחות של החברה.

החברה מאפשרת הזמנה של טיול הכוללת מדריך, מסלול וקבוצת האנשים המזמינה את הטיול. לכל הזמנה יש מספר מזהה, מחיר ותאריך הטיול.

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטיתרשים ERD

מילון

* Employee- ישות המתארת עובד בחברה.
* e\_id- ת.ז. של העובד.
* e\_name- שם של העובד.
* hire\_date- תאריך של תחילת העסקה של העובד.
* Guide- ישות המתארת מדריך שהוא עובד באגף טיולים.
* g\_id- ת.ז. של המדריך.
* experience- מס' שנות ניסיון של המדריך.
* Client- ישות המתארת לקוח בחברה.
* c\_tz- ת.ז. של הלקוח.
* c\_name- שם הלקוח.
* b\_date- תאריך יום ההולדת של הלקוח.
* My\_group- ישות המתארת קבוצה של לקוחות.
* m\_g\_id- מספר מזהה של קבוצה.
* amount- כמות הלקוחות בקבוצה.
* Area- ישות המתארת אזור בארץ.
* a\_id- מספר מזהה של אזור בארץ.
* a\_name- שם של אזור בארץ.
* City- ישות המתארת עיר.
* c\_id- מספר מזהה של עיר.
* c\_name- שם של עיר.
* My\_Path- ישות המתארת מסלול.
* p\_id- מספר מזהה של מסלול.
* p\_name- שם של מסלול.
* P\_Time- זמן משוער שלוקח לעשות את המסלול.
* P\_Length- אורך של מסלול.
* Difficulty- ישות המתארת רמת קושי של מסלול.
* d\_id- מספר מזהה של רמת קושי.
* d\_name- רמת קושי. (באופן מילולי)
* Kind- ישות המתארת סוג של מסלול.
* k\_id- מספר מזהה של סוג של המסלול.
* k\_name- שם של סוג המסלול.
* Invitation- ישות המתארת הזמנה של טיול.
* i\_id- מספר מזהה של הזמנה.
* i\_date- תאריך שבו מתקיים הטיול.
* price- מחיר של הטיול.

הטבלאות המתקבלות

Employee (e\_id, e\_name, hire\_date, c\_id)

Guide (g\_id, experience, e\_id)

Client (c\_tz, c\_name, b\_date, c\_id, m\_g\_id)

My\_group (m\_g\_id, amount)

Area (a\_id, a\_name)

City (c\_id, c\_name, a\_id)

Path (p\_id, a\_id, p\_name, time, length, d\_id, k\_id)

Difficulty (d\_id, d\_name)

Kind (k\_id, k\_name)

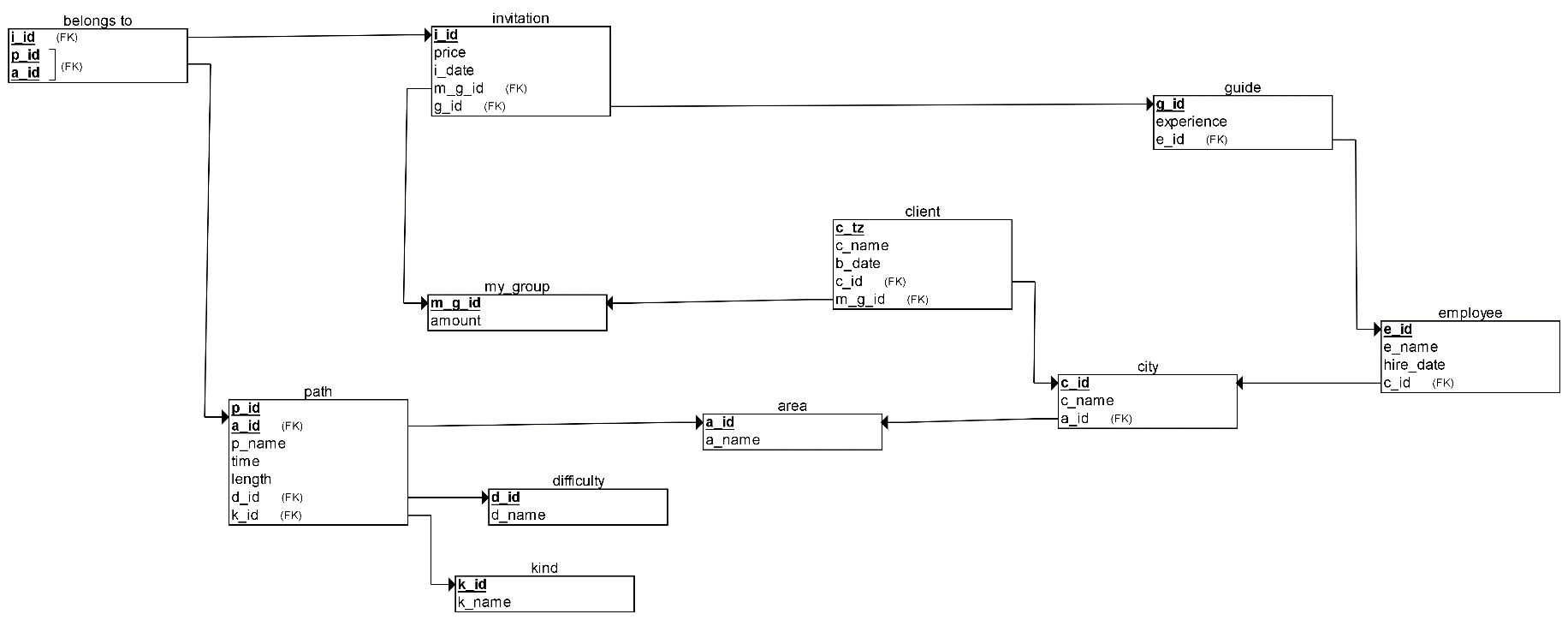
Invitation (i\_id, price, i\_date, m\_g\_id, g\_id)

Belongs to (i\_id, p\_id, a\_id)

בדיקה והסבר שכל הטבלאות מנורמלות ברמת 3CNF

* Employee- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.
* Guide- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.
* Client- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.
* My\_group- הטבלה היא מדרגת NF3 שכן יש רק 2 attributes (הוכח סמסטר קודם).
* Area- הטבלה היא מדרגת NF3 שכן יש רק 2 attributes (הוכח סמסטר קודם).
* City- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.
* Path- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.
* Difficulty- הטבלה היא מדרגת NF3 שכן יש רק 2 attributes (הוכח סמסטר קודם).
* Kind- הטבלה היא מדרגת NF3 שכן יש רק 2 attributes (הוכח סמסטר קודם).
* Invitation- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.
* Belongs to- כל שדה שאינו מפתח אינו תלוי פונקציונלית בשום שדה אחר שאינו המפתח.

תרשים DSD

****

מעבר מERD לטבלאות

לכל ישות נוצרת טבלה. לקשר belongs\_to נוצרה טבלה בגלל שהקשר הוא רבים לרבים, תכונות הטבלה הן המפתחות של 2 הטבלאות שבינהן הקשר. לכל שאר הקשרים לא נוצרת טבלה בגלל שהקשרים הם יחיד לרבים או יחיד ליחיד, אבל המפתח של הטבלה בצד היחיד נוספת כתכונה לטבלה שבצד הרבים.

יצירת טבלאות

CREATE TABLE area

(

a\_name VARCHAR(15) NOT NULL,

a\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (a\_id)

);

CREATE TABLE my\_group

(

amount NUMERIC(5) NOT NULL,

m\_g\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (m\_g\_id)

);

CREATE TABLE difficulty

(

d\_name VARCHAR(15) NOT NULL,

d\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (d\_id)

);

CREATE TABLE kind

(

k\_name VARCHAR(15) NOT NULL,

k\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (k\_id)

);

CREATE TABLE city

(

c\_id NUMERIC(5) NOT NULL,

c\_name VARCHAR(15) NOT NULL,

a\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (c\_id),

FOREIGN KEY (a\_id) REFERENCES area(a\_id)

);

CREATE TABLE path

(

p\_name VARCHAR(15) NOT NULL,

time NUMERIC(5, 2) NOT NULL,

length NUMERIC(5, 2) NOT NULL,

p\_id NUMERIC(5) NOT NULL,

a\_id NUMERIC(5) NOT NULL,

d\_id NUMERIC(5) NOT NULL,

k\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (p\_id, a\_id),

FOREIGN KEY (a\_id) REFERENCES area(a\_id),

FOREIGN KEY (d\_id) REFERENCES difficulty(d\_id),

FOREIGN KEY (k\_id) REFERENCES kind(k\_id)

);

CREATE TABLE client

(

c\_tz NUMERIC(5) NOT NULL,

c\_name VARCHAR(15) NOT NULL,

b\_date DATE NOT NULL,

c\_id NUMERIC(5) NOT NULL,

m\_g\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (c\_tz),

FOREIGN KEY (c\_id) REFERENCES city(c\_id),

FOREIGN KEY (m\_g\_id) REFERENCES my\_group(m\_g\_id)

);

CREATE TABLE employee

(

e\_id NUMERIC(5) NOT NULL,

e\_name VARCHAR(15) NOT NULL,

hire\_date DATE NOT NULL,

c\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (e\_id),

FOREIGN KEY (c\_id) REFERENCES city(c\_id)

);

CREATE TABLE guide

(

g\_id NUMERIC(5) NOT NULL,

experience NUMERIC(5, 1) NOT NULL,

e\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (g\_id),

FOREIGN KEY (e\_id) REFERENCES employee(e\_id)

);

CREATE TABLE invitation

(

price NUMERIC(5) NOT NULL,

i\_date DATE NOT NULL,

i\_id NUMERIC(5) NOT NULL,

m\_g\_id NUMERIC(5) NOT NULL,

g\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (i\_id),

FOREIGN KEY (m\_g\_id) REFERENCES my\_group(m\_g\_id),

FOREIGN KEY (g\_id) REFERENCES guide(g\_id)

);

CREATE TABLE belongs\_to

(

i\_id NUMERIC(5) NOT NULL,

p\_id NUMERIC(5) NOT NULL,

a\_id NUMERIC(5) NOT NULL,

PRIMARY KEY (i\_id, p\_id, a\_id),

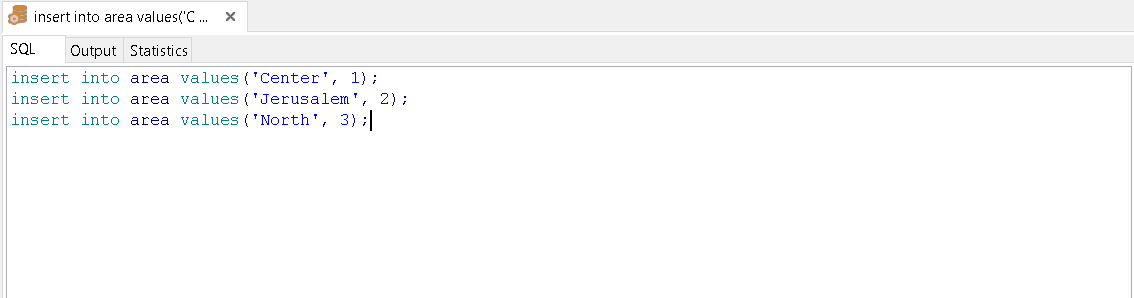
FOREIGN KEY (i\_id) REFERENCES invitation(i\_id),

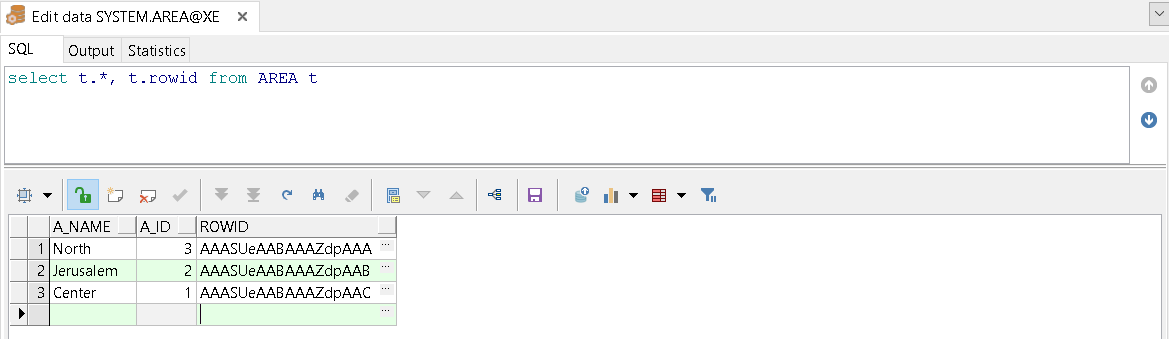
FOREIGN KEY (p\_id, a\_id) REFERENCES path(p\_id, a\_id)

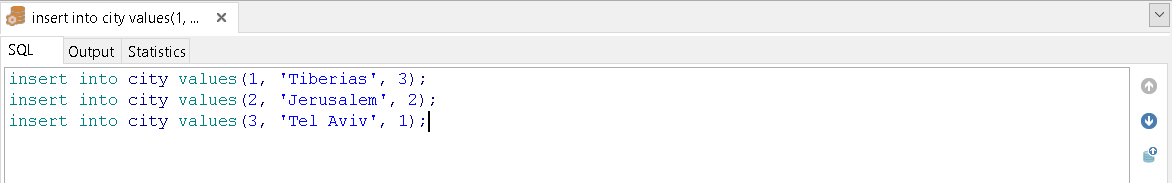
);

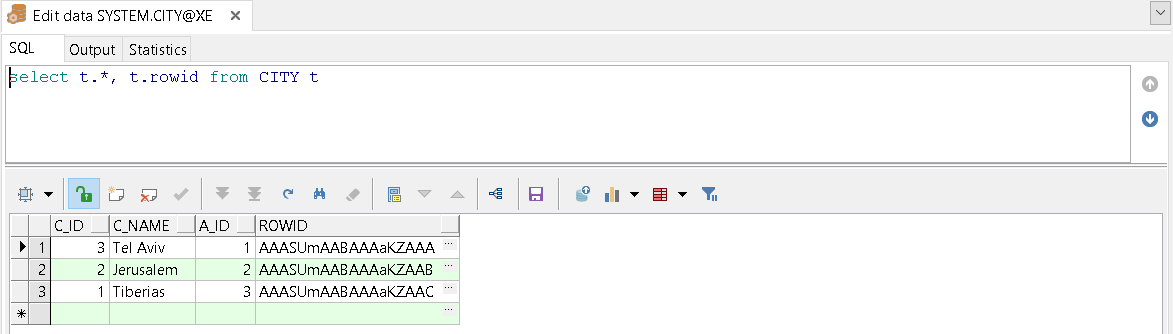
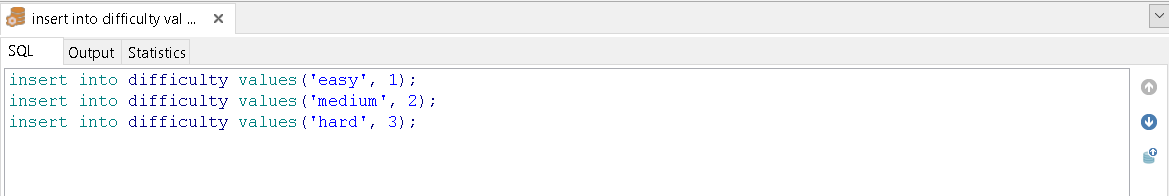
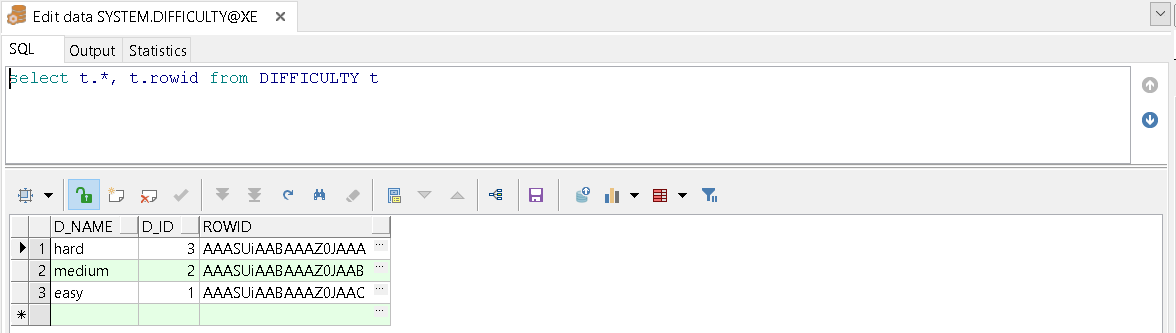
אכלוס טבלאות

פקודת insert

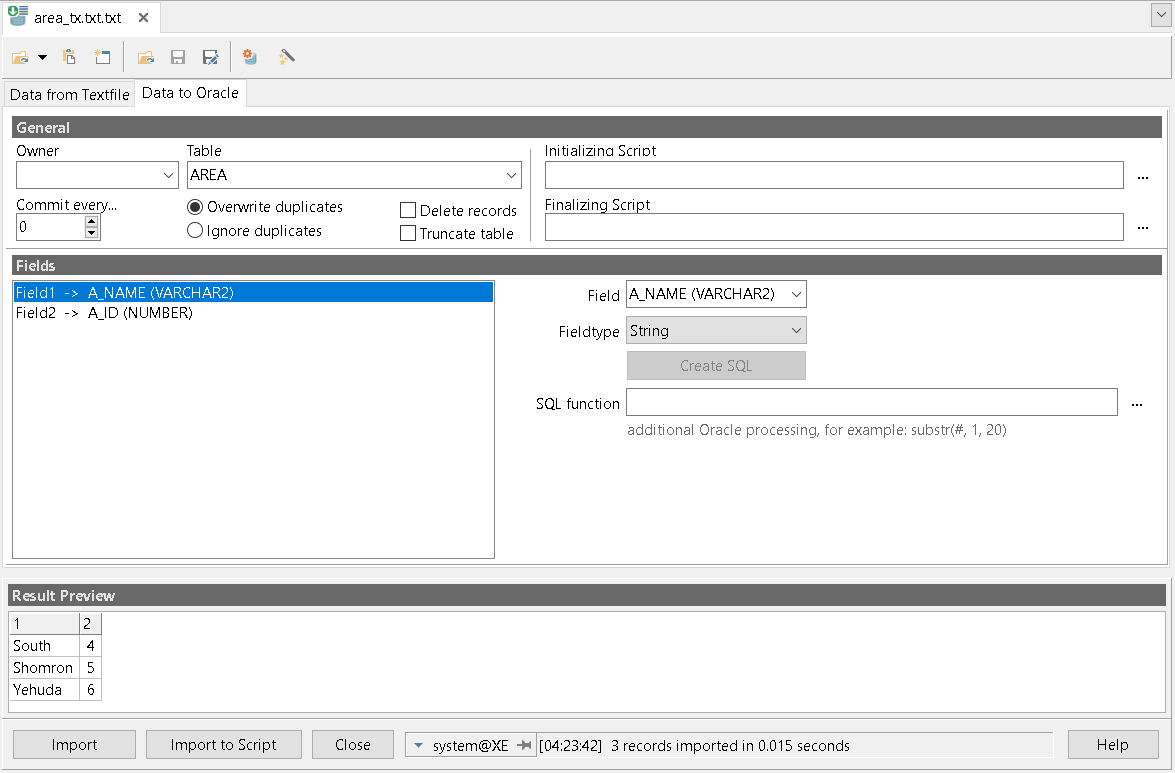
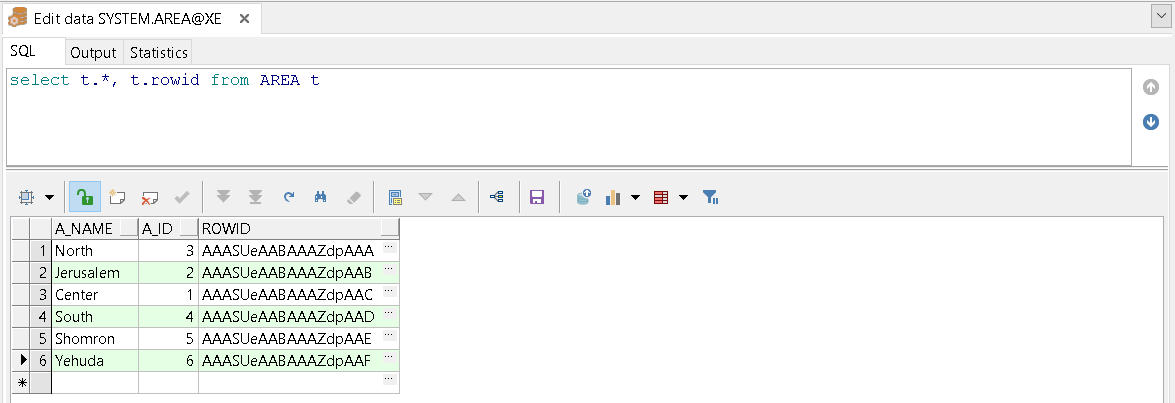


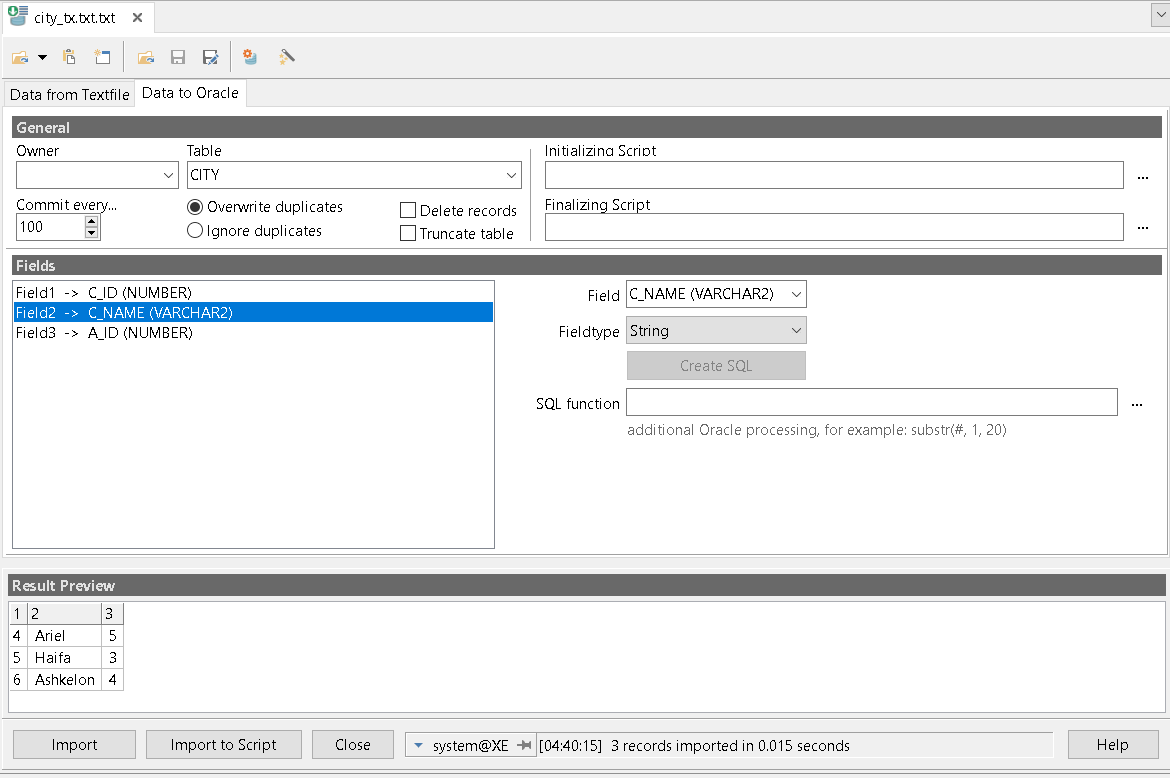


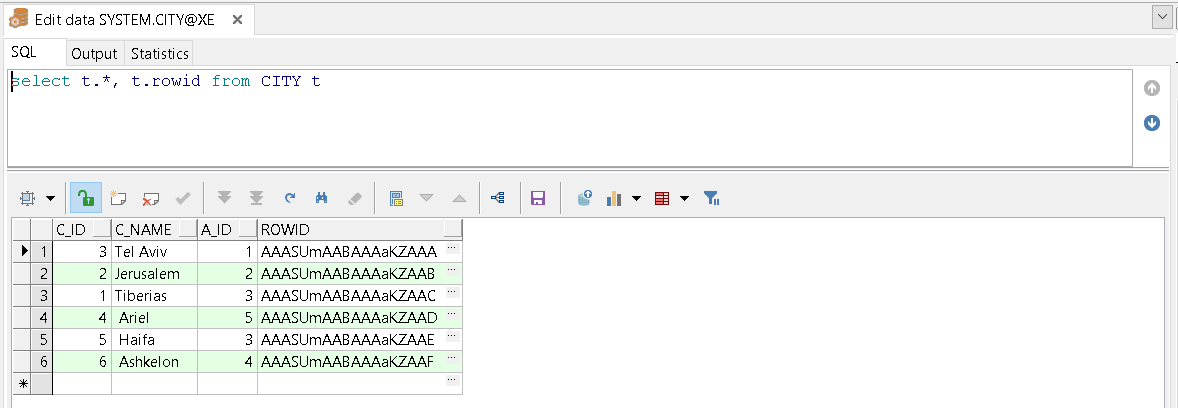


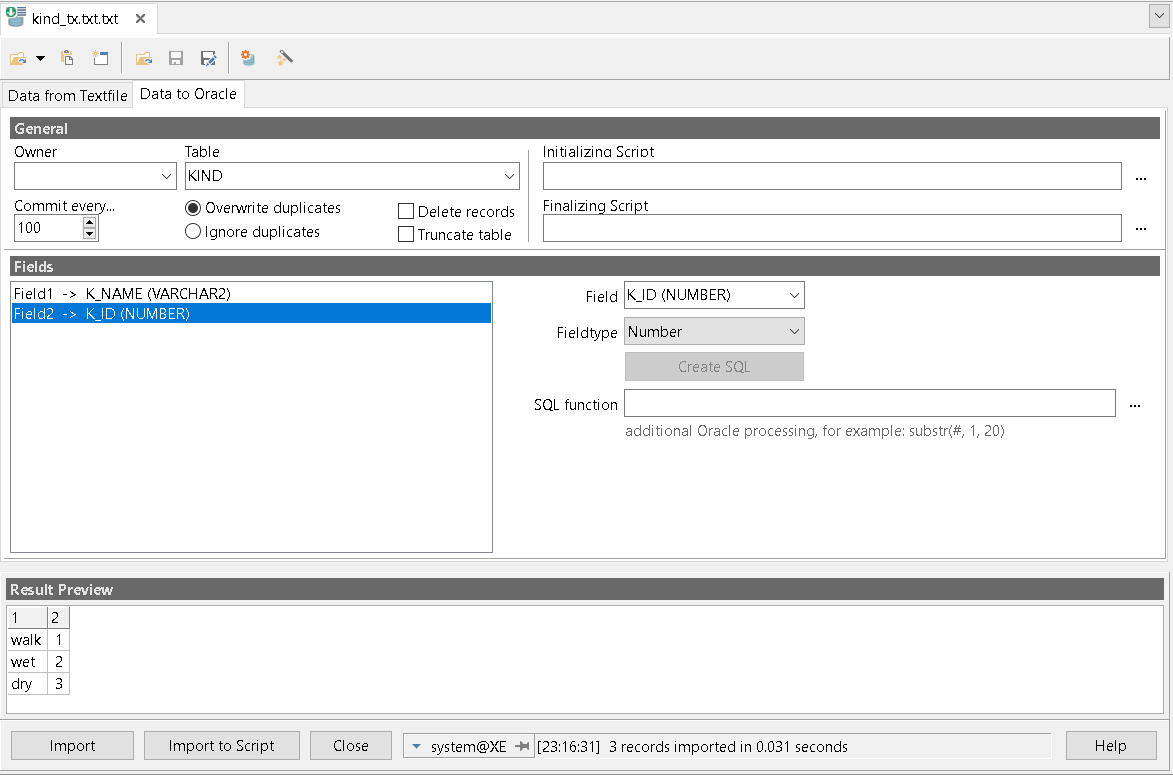


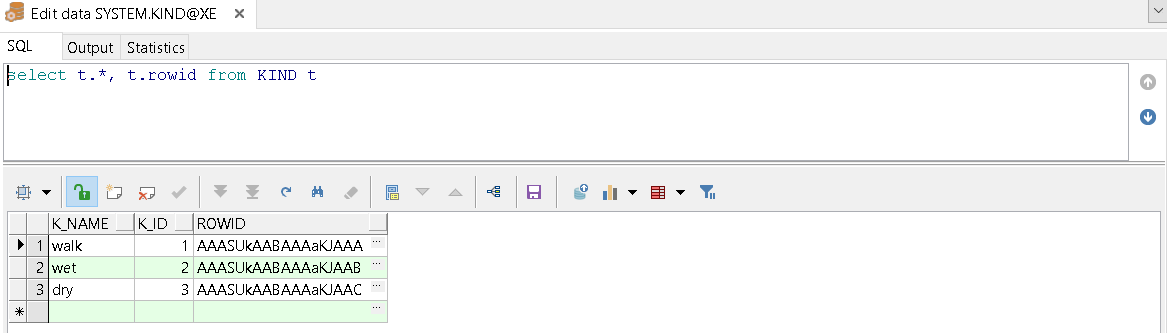
קליטה מקובץ TXT



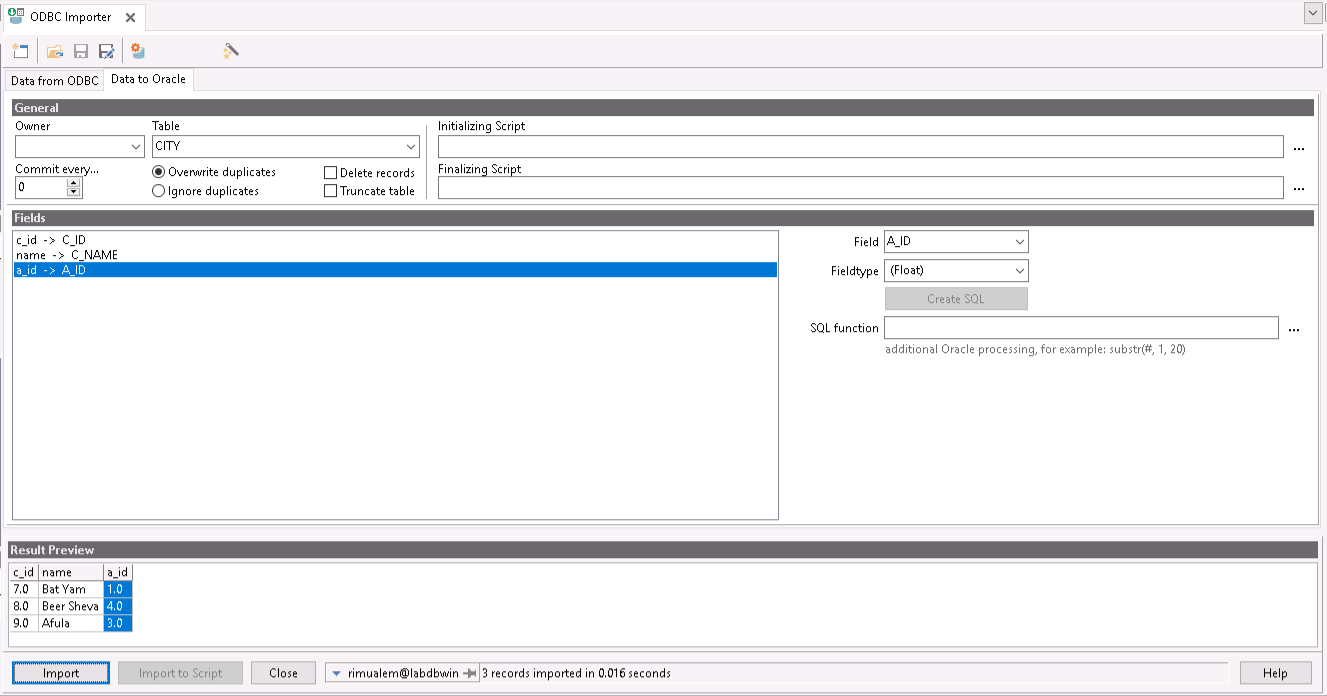


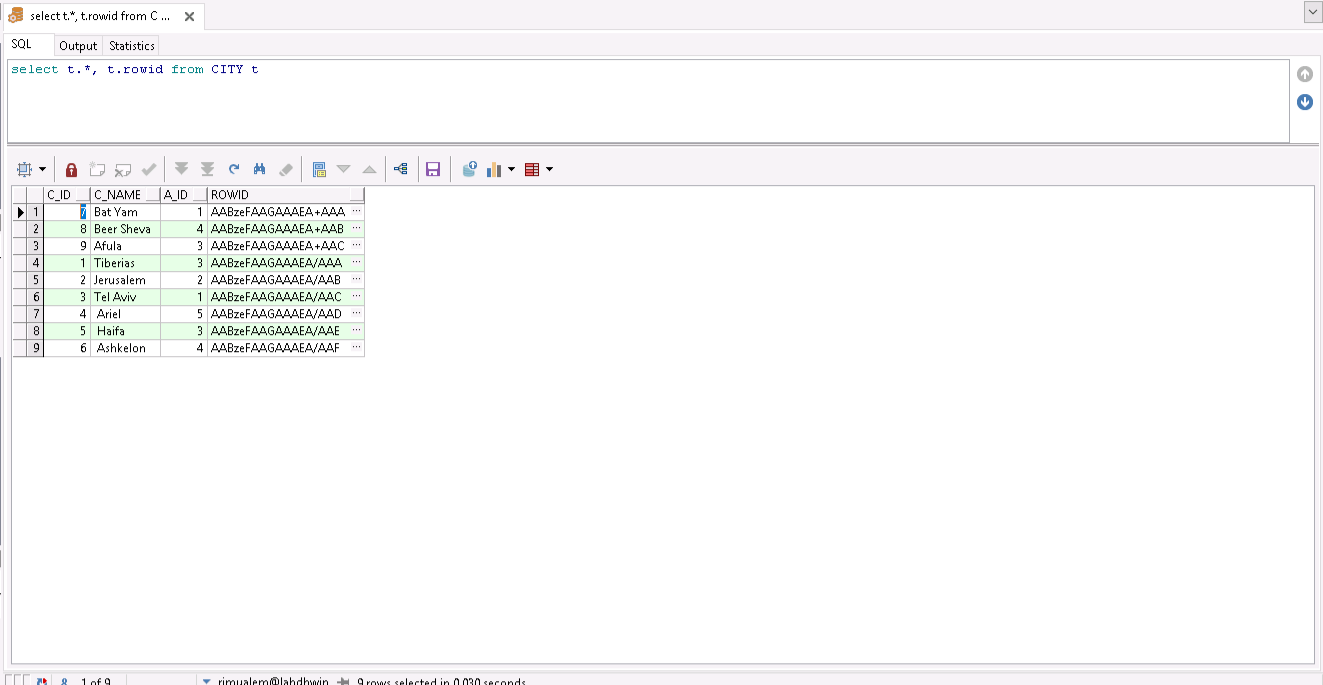


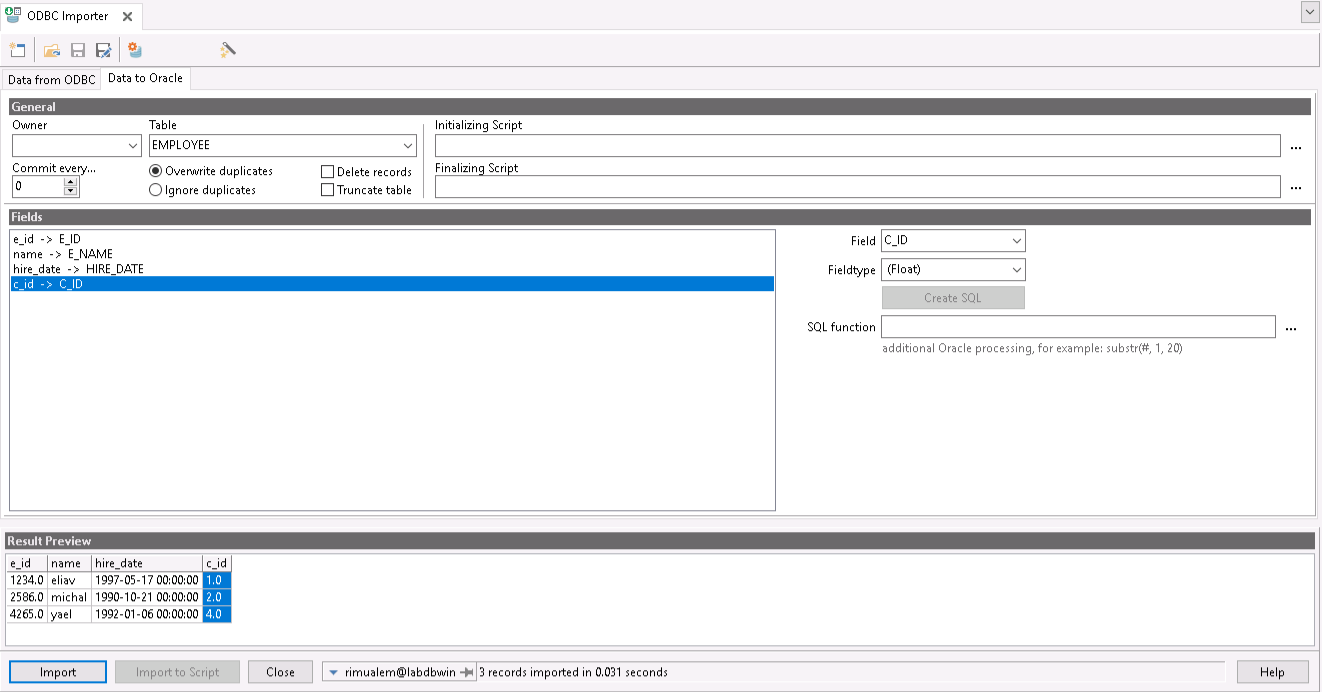


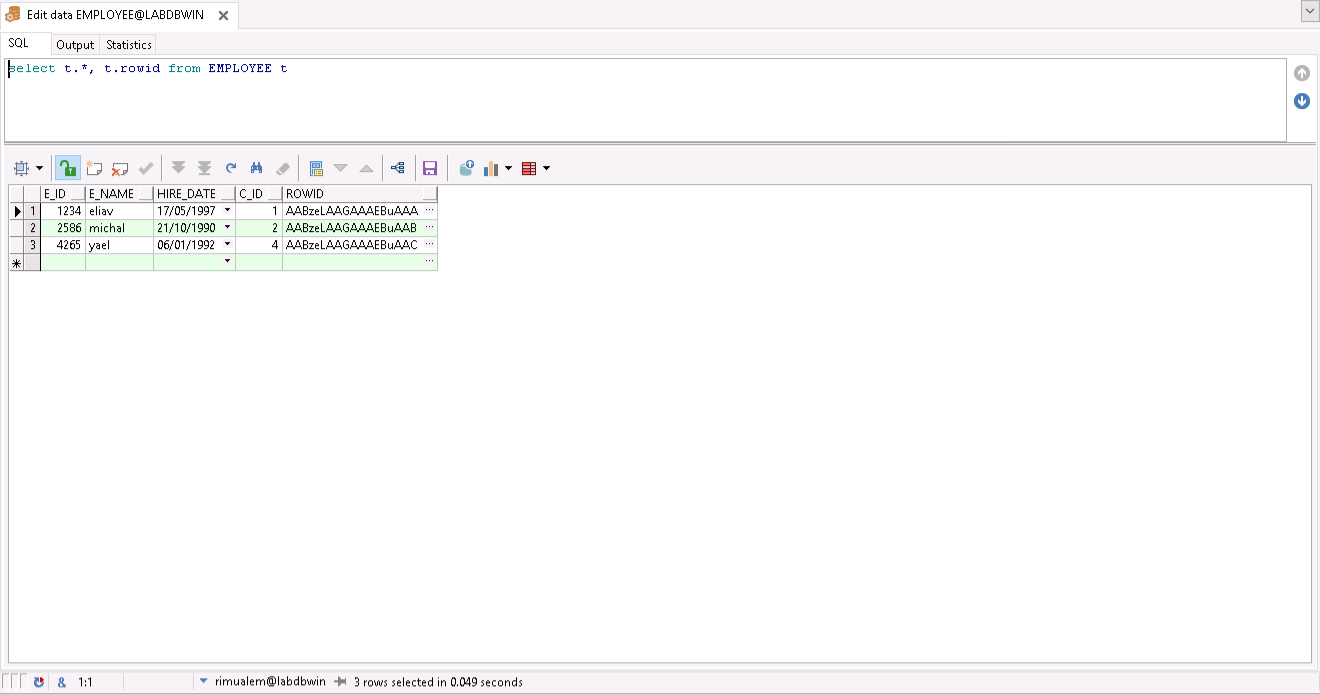


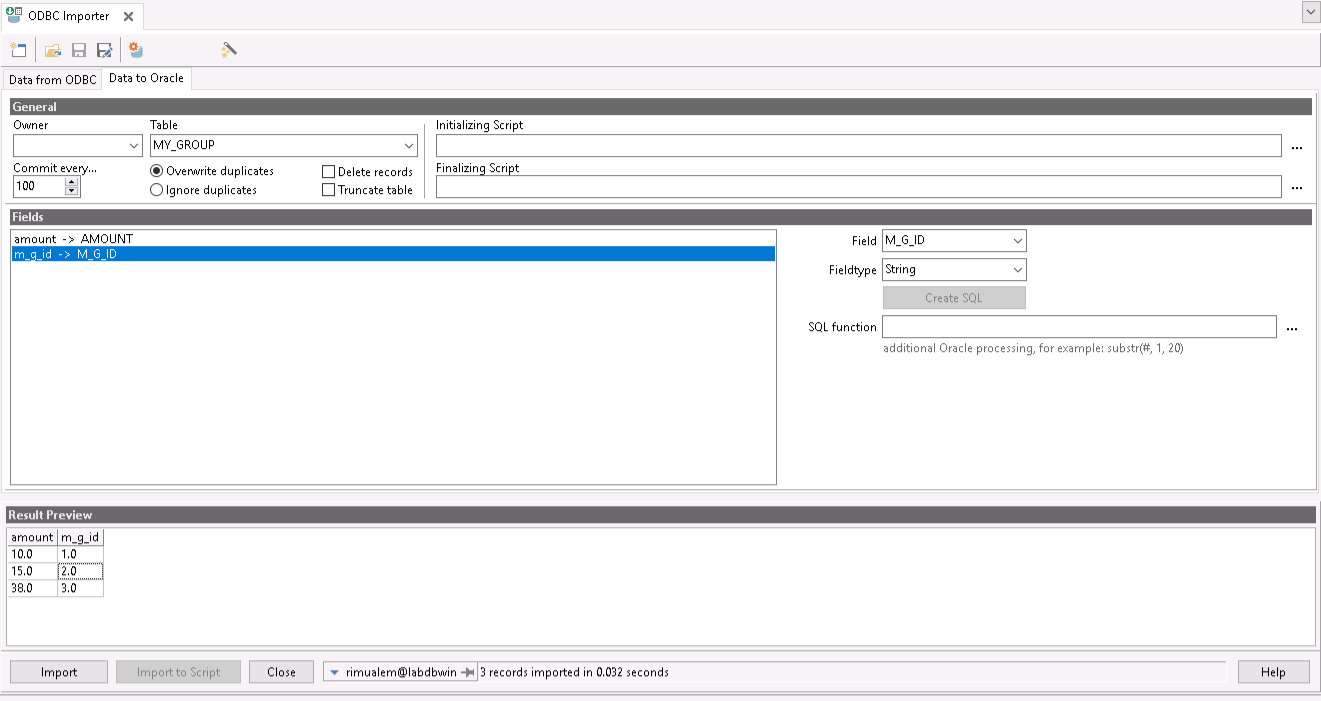
ODBC Importer

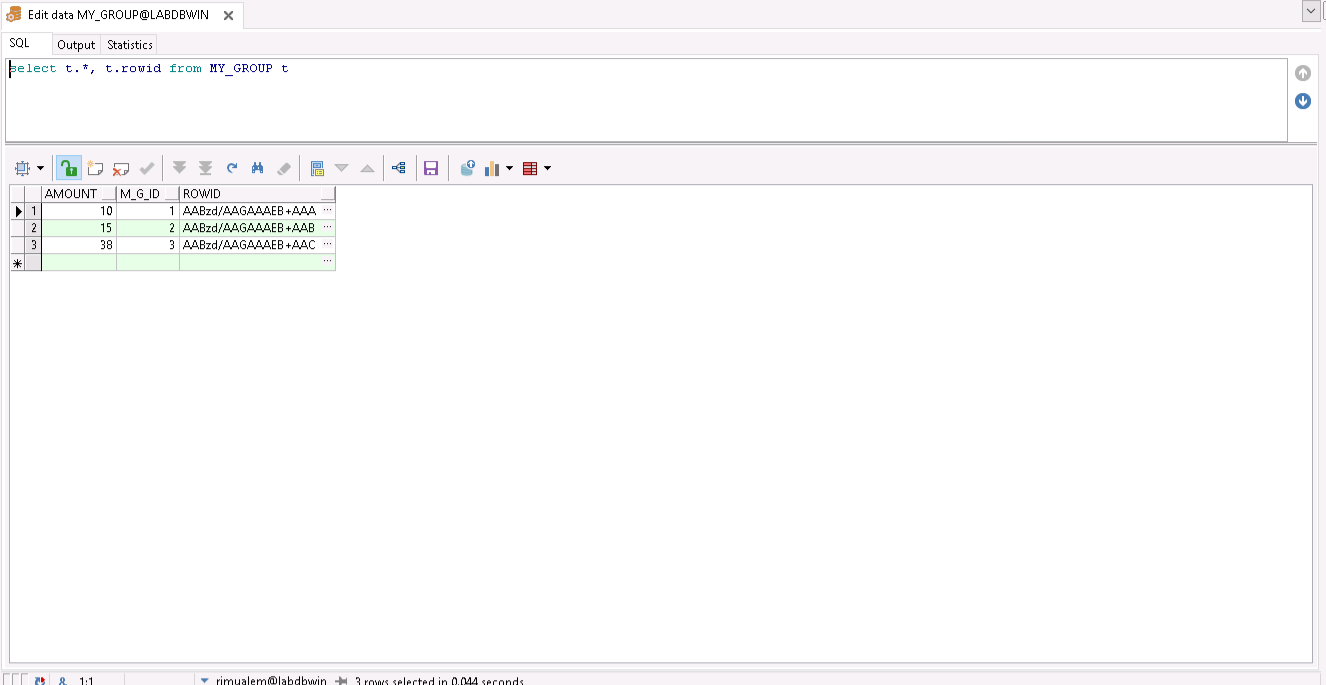


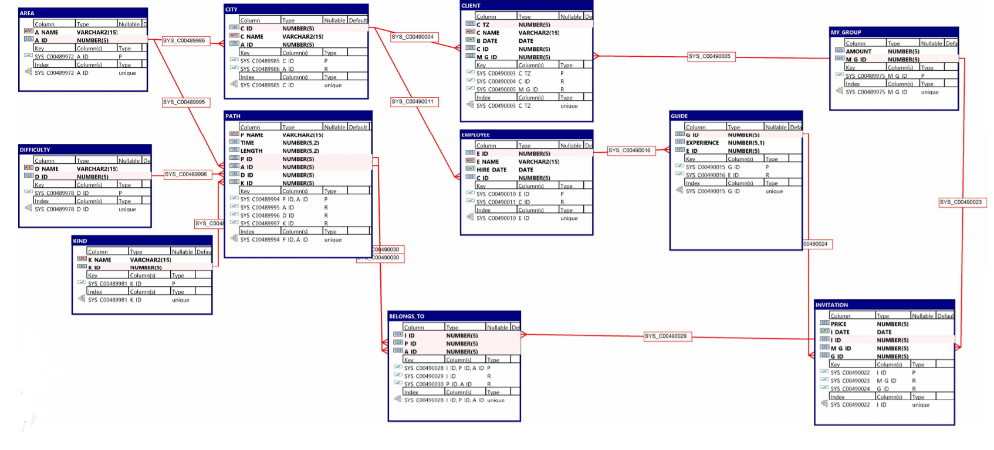




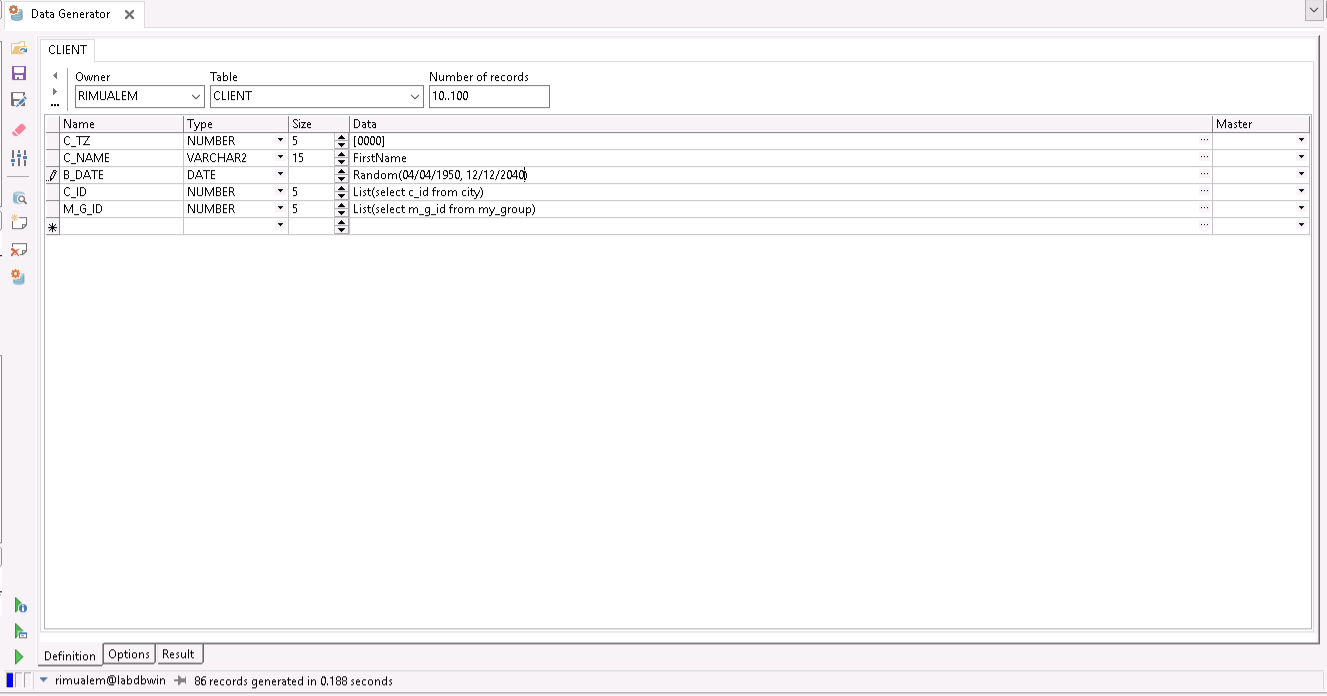




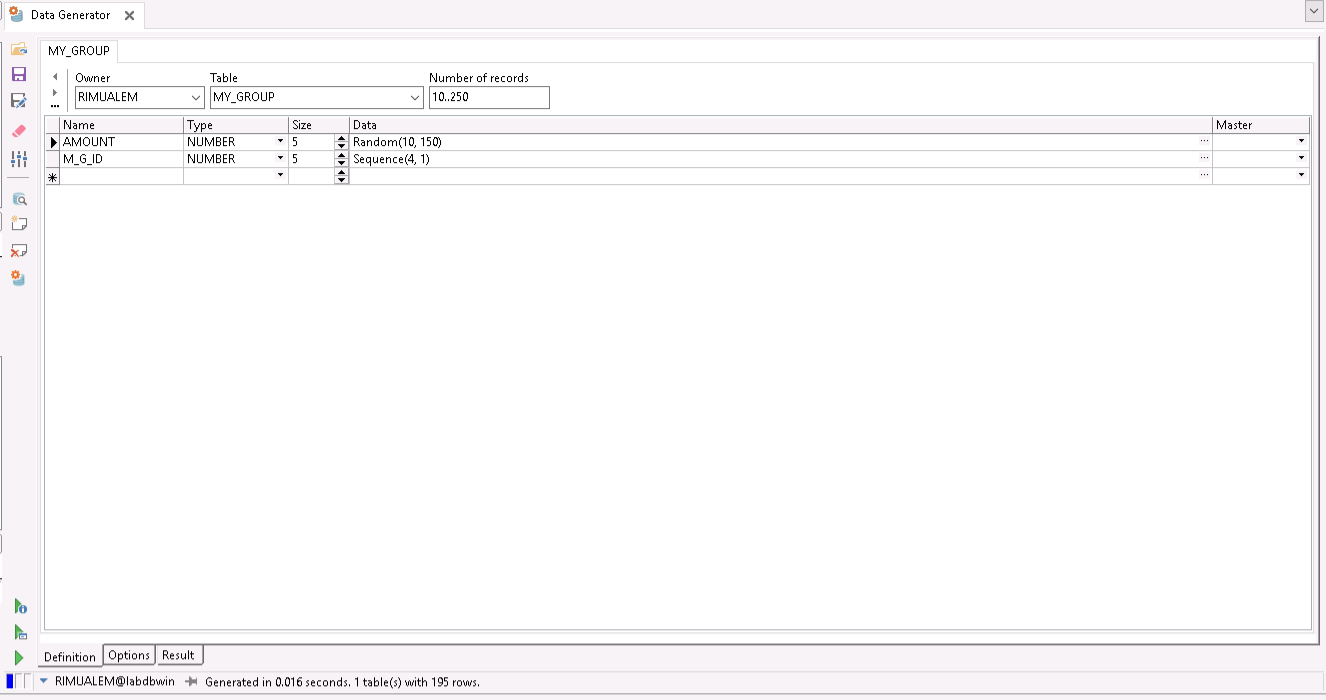


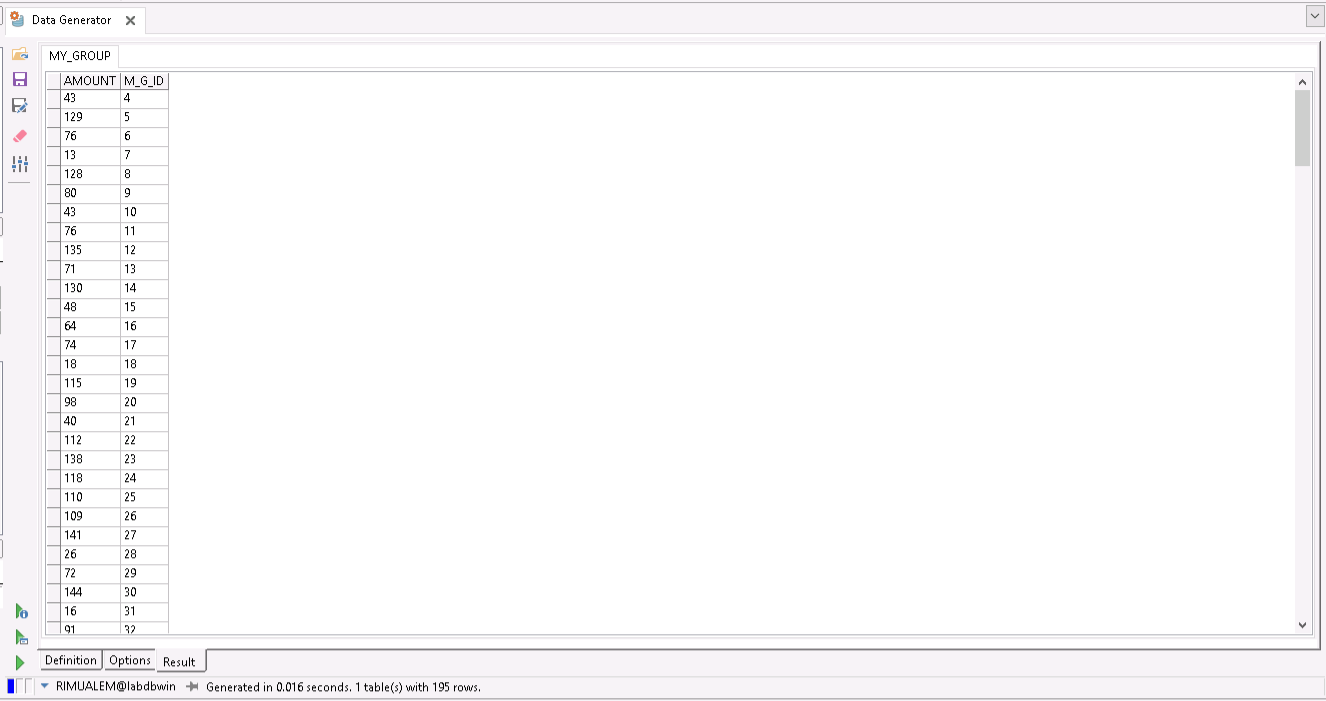
 תרשים diagram windows

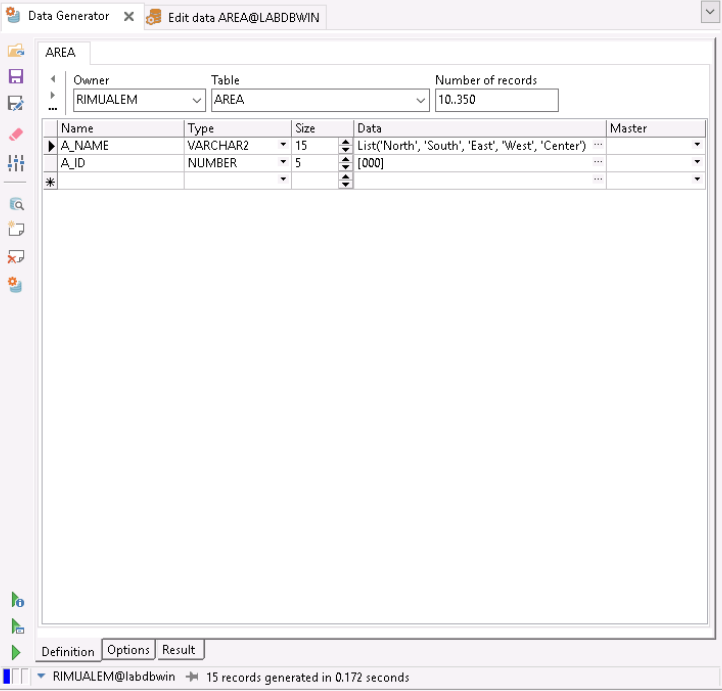
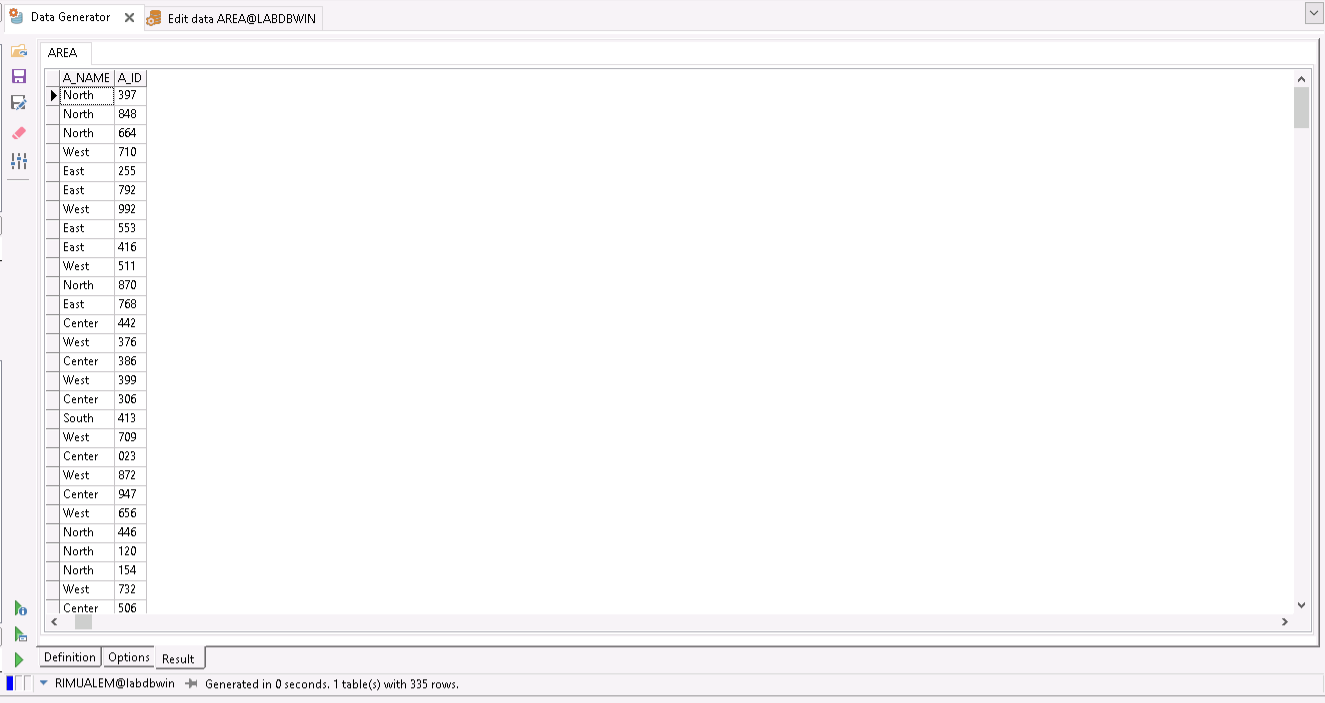
פקודת datagenerator











שלב מס' 2

אלגוריתם שיחזור של ERD

1. נהפוך כל טבלה לישות או לקשר.
2. טבלה שאין בה מפתח זר נהפוך לישות, כך שמפתחות הטבלה יהיו מפתחות הישות. (ישויות: City, Area, Client, Spa, Attractions, Ship, Path).

וכן טבלה שיש בה מפתח ראשי ומפתח זר נהפוך לישות כך שמפתחות הטבלה הם מפתחות הישות. (ישויות: Cruise, Employee, Room).

1. תהי A טבלה בה המפתח מורכב רק ממפתחות זרים {bi….bk} אזי ניצור קשר של רבים לרבים בין הישויות שהמפתחות שלהם הם {bi….bk}. (קשרים של רבים לרבים: visit, go\_to, participate, include, working\_at).
2. תהי טבלה A ובה יש תכונה שהיא מפתח זר מטבלה B. ניצור קשר של רבים ליחיד מA לB בהתאמה. (קשרים של יחדי לרבים: ,IS\_AT Relationship, inside, in, live).

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטיתרשים ERD של קרויז

תרשים ERD משותף

תמונה שמכילה טקסט, מפה

התיאור נוצר באופן אוטומטי

תמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטיתרשים diagram windows משותף

תיעוד החיבור

* שינוי שמות הטבלאות הגרעיניות, ייבוא טבלאות של הארגון קרויז, חיבור טבלאות משותפות- בטבלת client עדכון לכל הלקוחות שלא היה להם מס' מזהה של עיר ומס' מזהה של קבוצה, מחיקת הטבלאות שאת הנתונים שלהם הוספנו לפרויקט המשותף.
* בין טבלת city לclient יש 2 קשרים: קשר יחיד לרבים המציין את מקום מגוריו של הלקוח וקשר רבים לרבים המציין את הערים בהם הלקוח מטייל.
* בסיום החיבור של 2 האגפים מחיקת הטבלאות שלקחנו מהם את הנתונים לטבלאות זהות גרמה למחיקת המפתח זר שנמצא בטבלאות של קשר רבים לרבים לכן שנינו את שמות הטבלאות של הקשרים, בנינו טבלאות המייצגות קשר רבים לרבים והכנסנו את הנתונים מהטבלאות הקודמות לחדשות וכמובן מחקנו את הטבלאות הקודמות.

שלב מס' 3

פקודות select

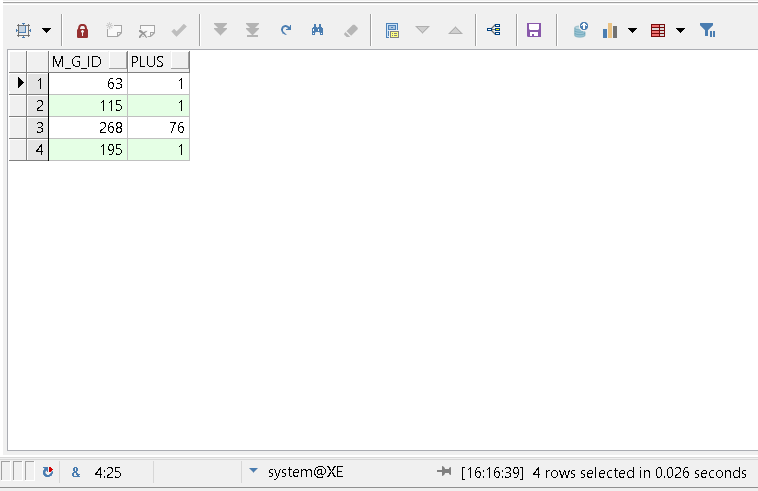
* השאילתה מוצאת את מספר המזהה של קבוצות שמספר הלקוחות גדול יותר ממספר המקומות בקבוצה.

select m\_g\_id, (count(\*)-amount) as plus

from my\_group natural join client

group by m\_g\_id, amount

having count(\*)>=amount;



* השאילתה מוצאת את מספר המזהה של כל הלקוחות שלא היו במסלול 'Onfud’.

select C.c\_tz

from client C

where C.m\_g\_id not in

(select I.m\_g\_id

from invitation I

where I.i\_id in

(select B.i\_id

from belongs\_to B

where B.p\_id in

(select p\_id

from my\_path P

where p.p\_name ='Onfud')))

תמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטיסה"כ מספר הלקוחות הוא 654, זאת אומרת שרק 2 אנשים ביקרו ב'Onfud’.

* השאילתה מוצאת את מספר המזהה של כל הלקוחות שהיו בכל האטרקציות

select C.c\_tz

from client C

where not exists

(select A.atraction\_number

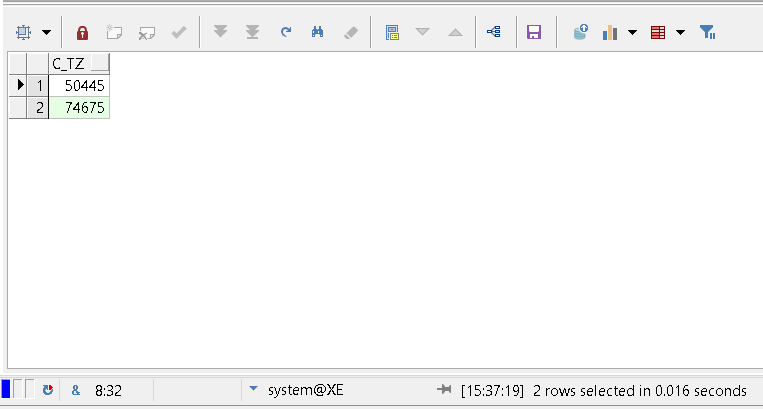
from ATTRACTIONS A

where A.atraction\_number not in

(select P.atraction\_number

from participate P

where C.c\_tz= P.client\_id))

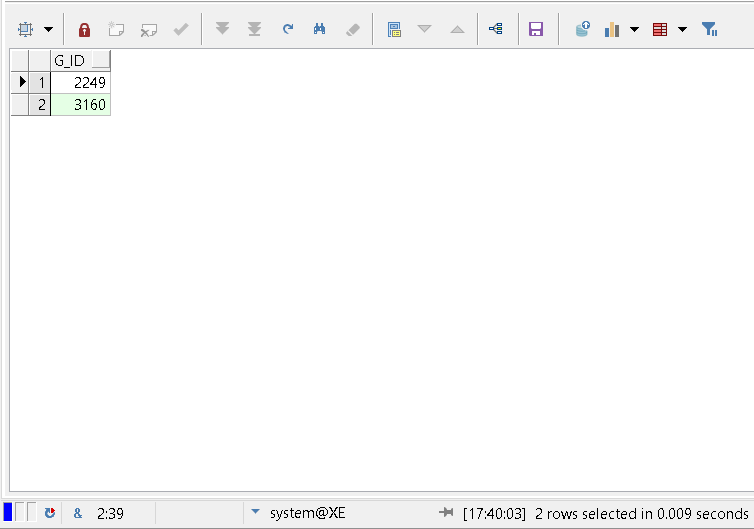


* השאילתה מוצאת את המספר המזהה של המדריכים אשר הדריכו לקוחות שטיילו ביום ההולדת שלהם.

select i.g\_id

from client c ,my\_group m,invitation i

where c.m\_g\_id=m.m\_g\_id and i.m\_g\_id=m.m\_g\_id and i.i\_date=c.b\_date



* השאילתה מוצאת מדריכים שגם רוצים לטייל, אך הם לא מדריכים את עצמם:

(SELECT g.g\_id

FROM guide g

where g.g\_id not in

(SELECT g.g\_id

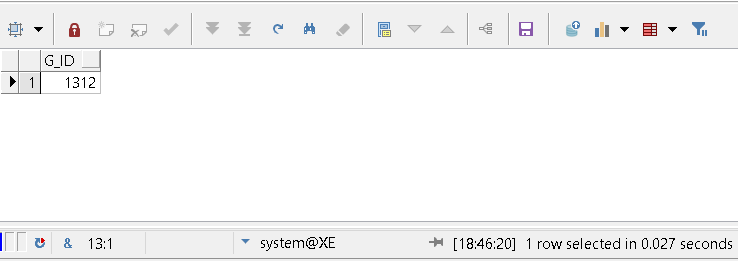
FROM guide g, client c, invitation i, my\_group m

WHERE g.g\_id = c.c\_tz and c.m\_g\_id=m.m\_g\_id and m.m\_g\_id=i.m\_g\_id and i.g\_id=g.g\_id ))

intersect

(select c.c\_tz

from client c natural join my\_group m natural join invitation i)



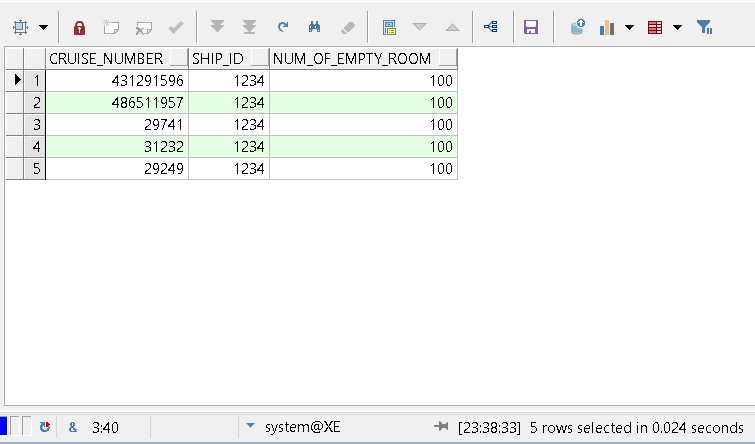
* השאילתה מוצאת את כמות החדרים הנותרים בקרוזים שבחודש הקרוב

select cruise\_number, ship\_id, count(\*) as num\_of\_empty\_room

from cruise c natural join ship s natural join room r

where (c.begin\_date >= to\_date('07/06/2020','dd/mm/yyyy') and c.end\_date <= to\_date('07/07/2020','dd/mm/yyyy'))and r.status='OK'

group by cruise\_number, ship\_id



* השאילתה מוצאת את מספר המזהה של העיר שגרים בה הכי הרבה אנשים שהיו גם בספא וגם באטרקציות.

select (c\_id) as city\_id

from participate p natural join go\_to g , client c

where c.c\_tz=client\_id

group by c\_id

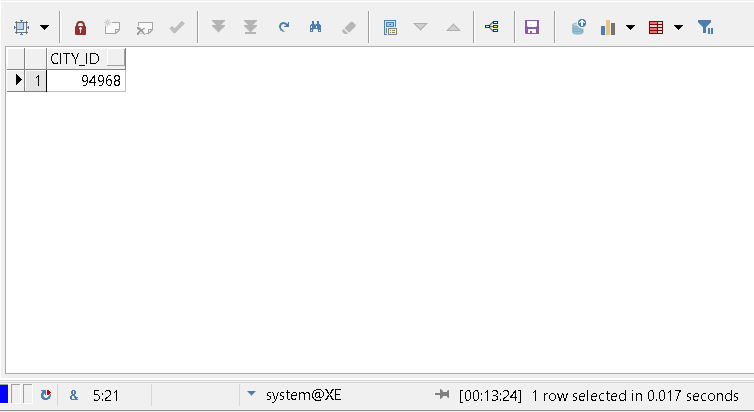
having count(\*)>=ALL

(select count(\*)

from participate p natural join go\_to g , client c

where c.c\_tz=client\_id

group by c\_id)

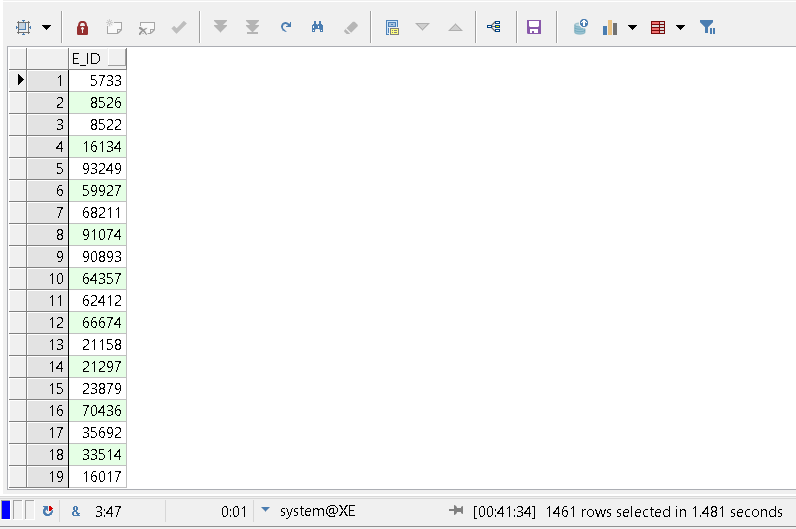


* השאילתה מוצאת את תעודת הזהות של העובדים העובדים במסלול שמתחיל או מסתיים בעיר מגוריהם.

select distinct e\_id

from city c natural join employee e, path p

where c\_name=p.begin\_city or c\_name=p.end\_city

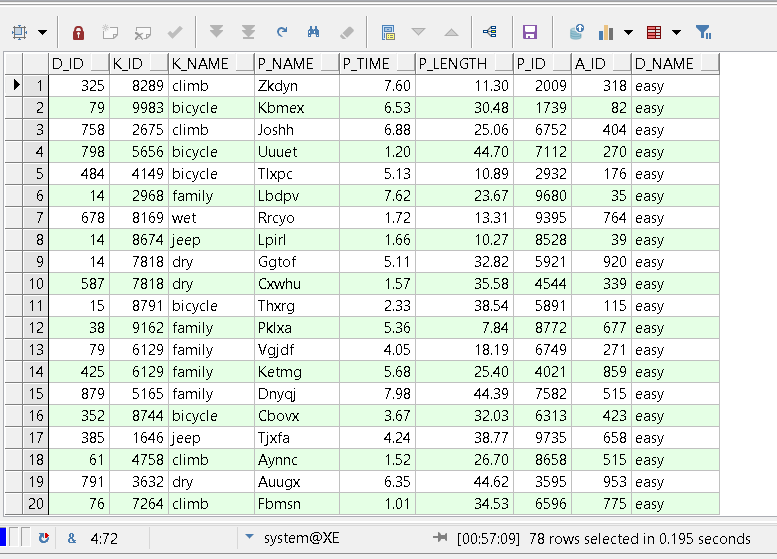


פקודות update

select \*

from kind natural join my\_path natural join difficulty

where d\_name = 'easy'



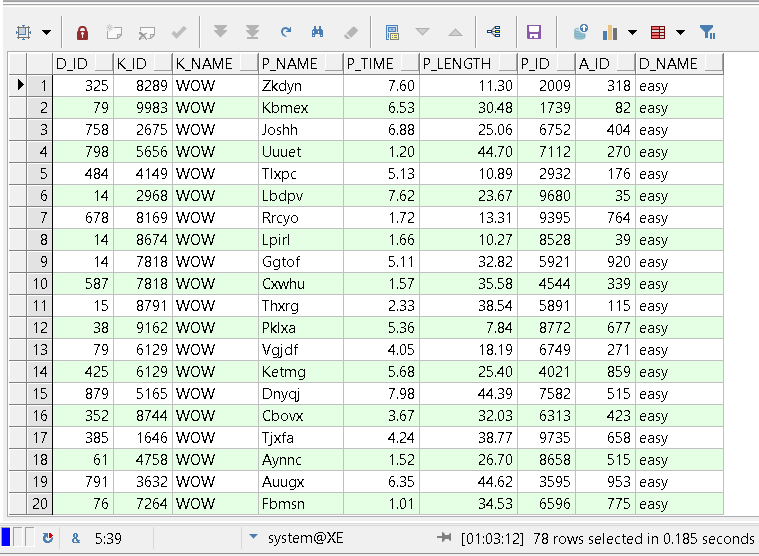
פקודת update: עדכון לכל המסלולים הקלים שם מסלול : WOW.

UPDATE kind

SET k\_name = 'WOW'

WHERE k\_name in (select k\_name

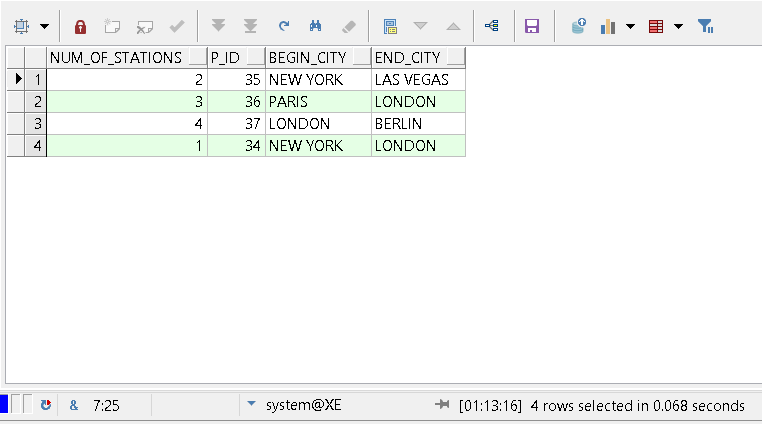
from kind natural join my\_path natural join difficulty

 where d\_name = 'easy')

select \*

from path

where num\_of\_stations<10

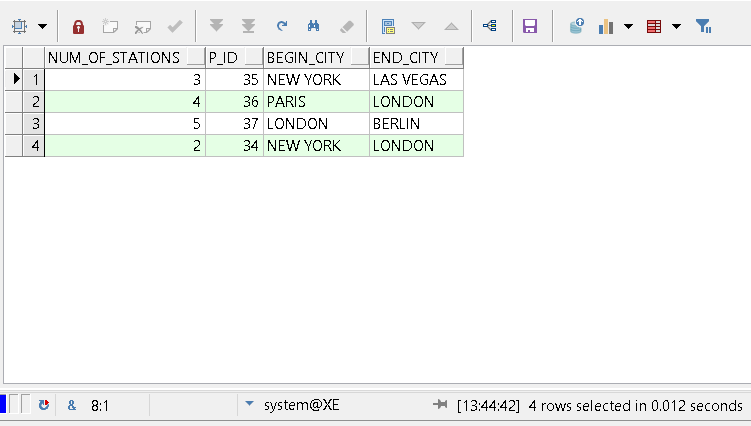


פקודת update: עדכון של הוספת תחנה לכל המסלולים שמספר התחנות שלהם קטנה מ10 תחנות.

UPDATE path

SET num\_of\_stations = num\_of\_stations+1

where num\_of\_stations<10



פקודות delete

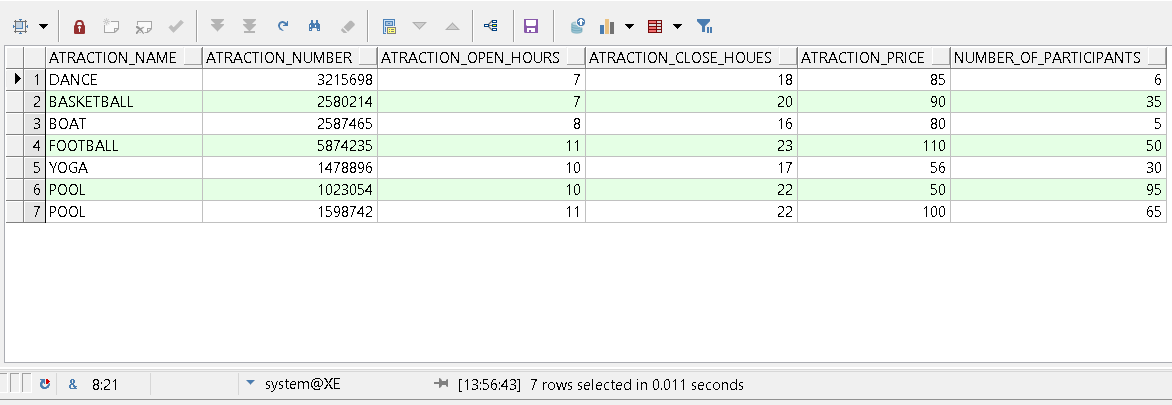
* הפקודה מוחקת את כל האטרקציות בהם אף לקוח לא השתתף.

select \*

FROM Attractions

WHERE ATRACTION\_NUMBER not in (SELECT ATRACTION\_NUMBER

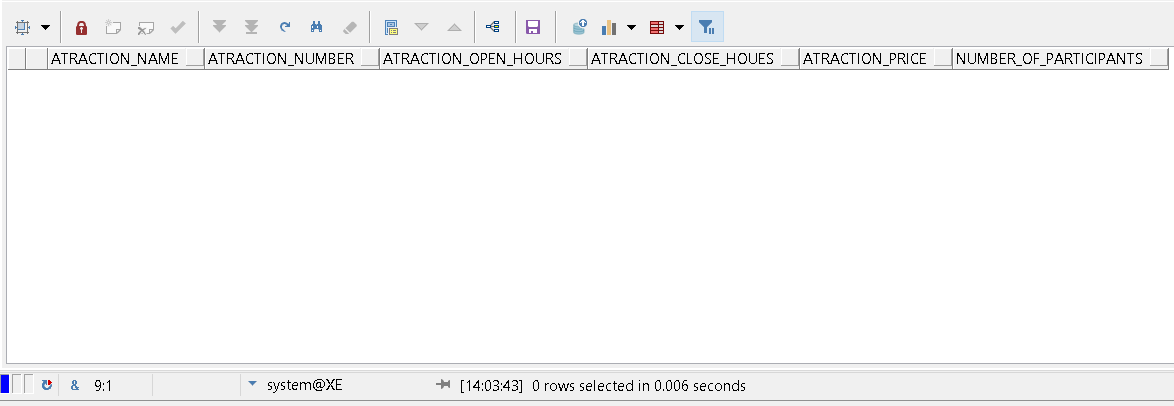
FROM participate)



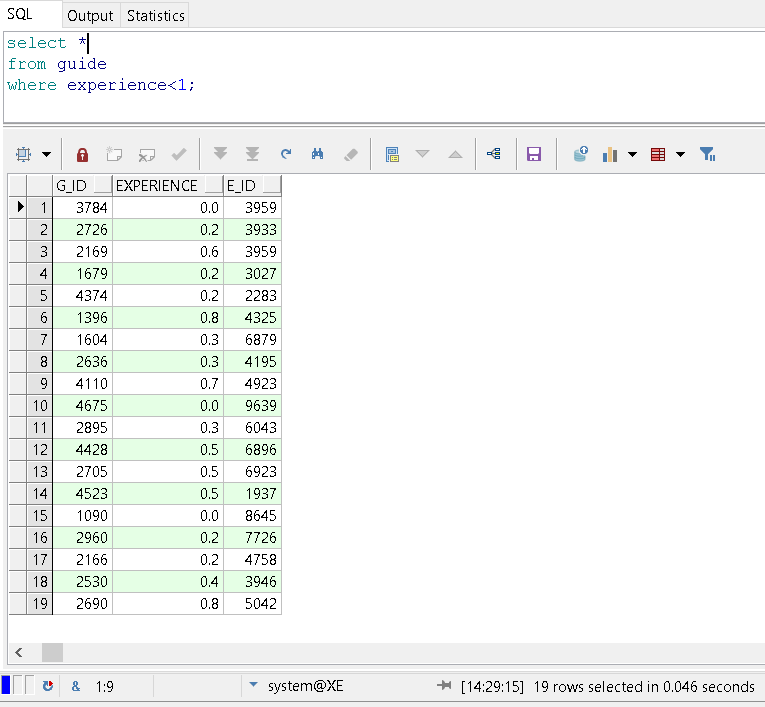
DELETE FROM Attractions

WHERE ATRACTION\_NUMBER not in (SELECT ATRACTION\_NUMBER

FROM participate)



* הפקודה מוחקת את כל המדריכים שמספר שנות הניסיון שלהם קטן משנה.



delete

from belongs\_to

where i\_id in

(select i\_id

from invitation

where g\_id in (select g\_id

from guide

where experience<1));

delete

from invitation

where g\_id in (select g\_id

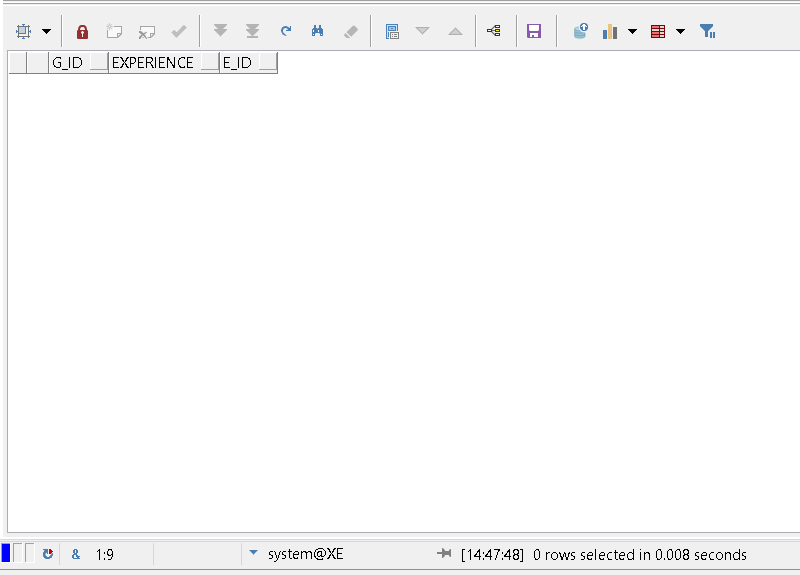
from guide

where experience<1);

delete

from guide

where experience<1;



פקודות commit וrollback

commit

delete

from belongs\_to

where i\_id in

(select i\_id

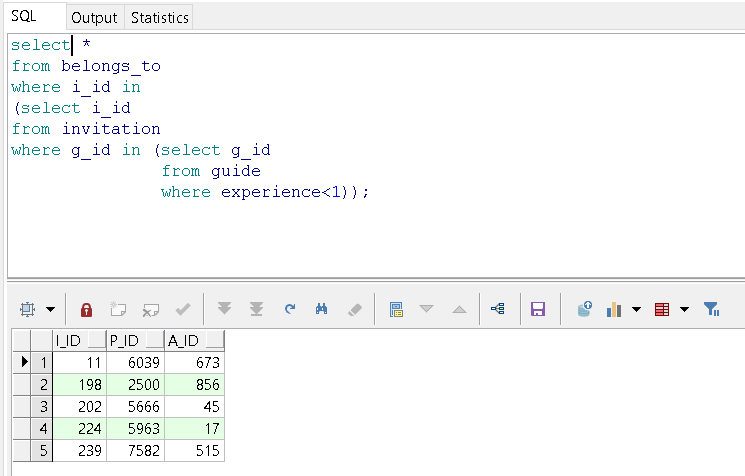
from invitation

where g\_id in (select g\_id

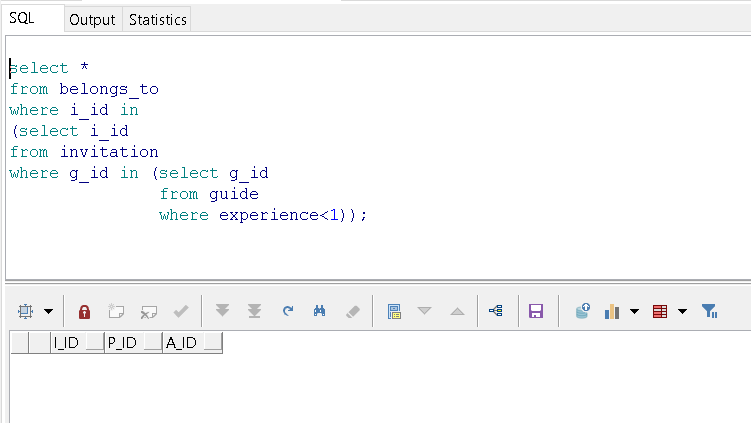
from guide

where experience<1));

לאחר הרצת פקודת delete הרצנו פקודת select עדיין הנתונים לא נמחקו מהטבלה.

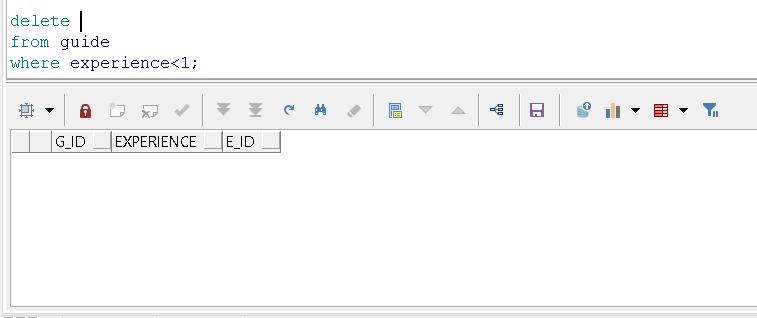


לאחר הרצת פקודת commit.

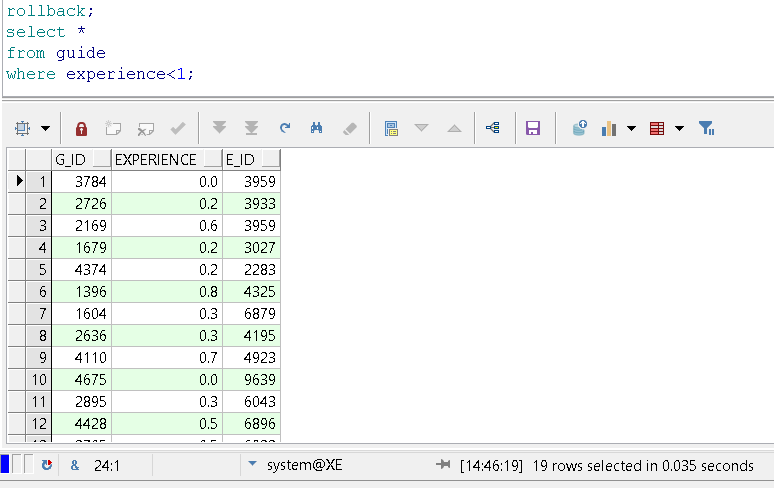


rollback

ביצענו פקודת delete:



לאחר הרצת פקודת rollback ניתן לראות לא נמחקו:



אינדקסים

* יצירת אינדקס לעמודת c\_name בטבלת city.

create index city\_name

on city(c\_name);

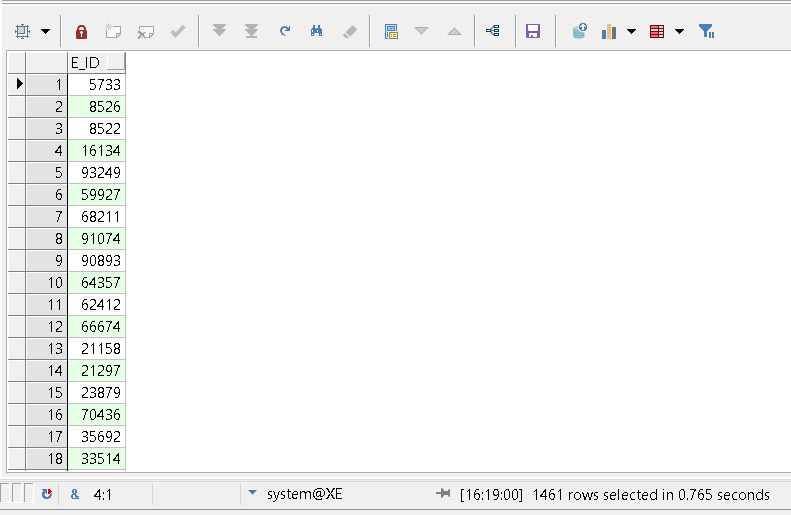
ניתן לראות שזמן הריצה של שאילתה המוצאת את תעודת הזהות של העובדים העובדים במסלול שמתחיל או מסתיים בעיר מגוריהם לא ירד משמעותית לאחר יצירת האינדקס בגלל שאין כמעט שמות של ערים שחוזרים על עצמם.

select distinct e\_id

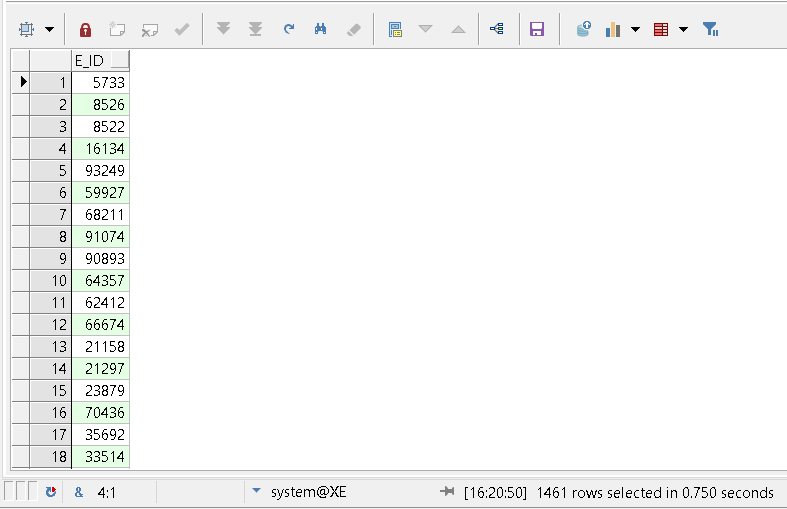
from city c natural join employee e, path p

where c\_name=p.begin\_city or c\_name=p.end\_city

לפני יצירת האינדקס:



לאחר יצירת האינדקס:



* יצירת אינדקס לstatus בטבלת room.

create index status\_room

on room(status);

ניתן לראות שזמן הריצה של השאילתה מוצאת את כמות החדרים הנותרים בקרוזים שבחודש הקרוב ירד לאחר יצירת האינדקס בגלל שיש רק 2 סוגים של מצב חדר.

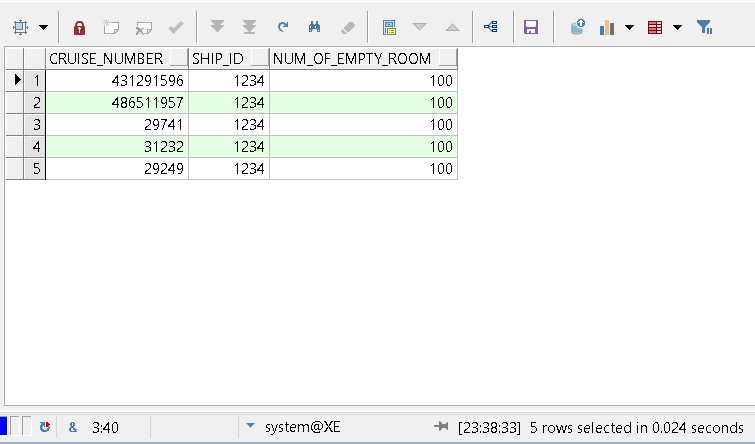
select cruise\_number, ship\_id, count(\*) as num\_of\_empty\_room

from cruise c natural join ship s natural join room r

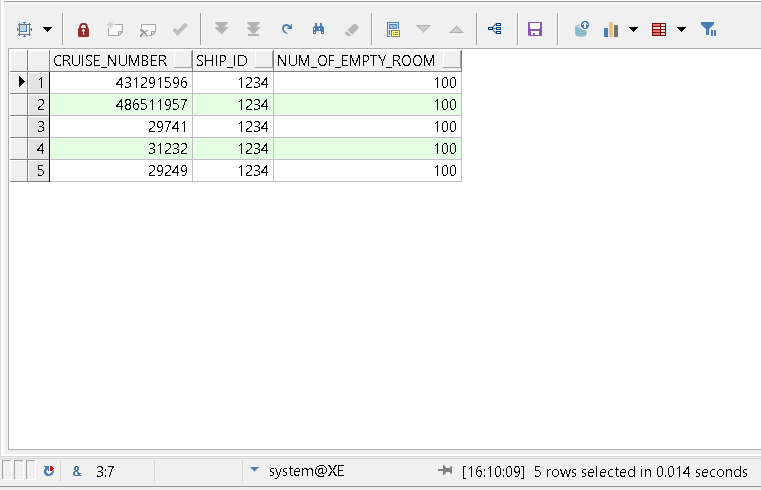
where (c.begin\_date >= to\_date('07/06/2020','dd/mm/yyyy') and c.end\_date <= to\_date('07/07/2020','dd/mm/yyyy'))and r.status='OK'

group by cruise\_number, ship\_id

לאחר יצירת האינדקס:



לאחר יצירת האינדקס:



* יצירת אינדקס לamount בטבלת my\_group.

create index amount\_of\_people

on my\_group(amount);

ניתן לראות שזמן הריצה של השאילתה המוצאת את מספר המזהה של קבוצות שמספר הלקוחות גדול יותר ממספר המקומות בקבוצה ירד בגלל שיש מספר קבוצות שסכום האנשים בקבוצה חוזר על עצמו.

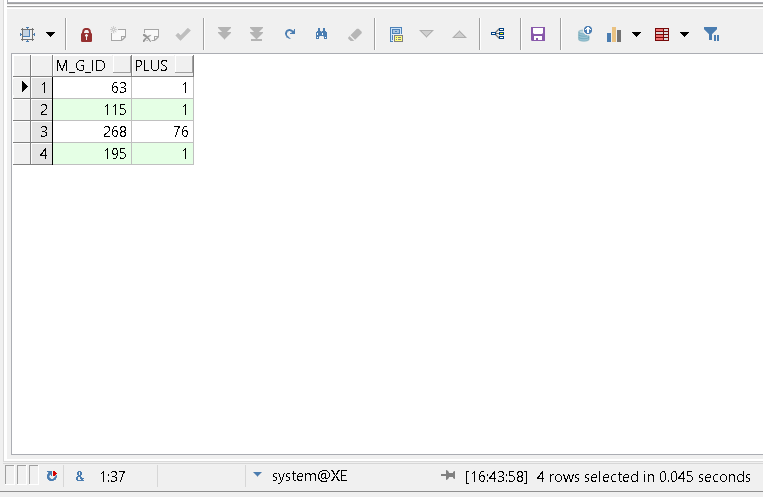
select m\_g\_id, (count(\*)-amount) as plus

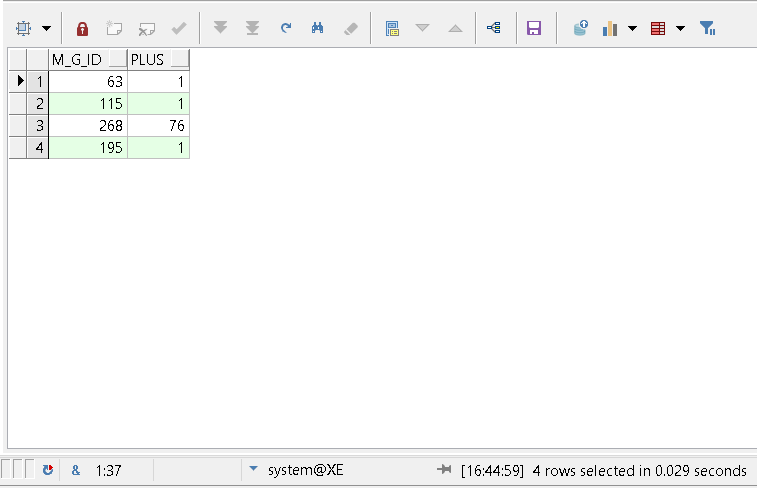
from my\_group natural join client

group by m\_g\_id, amount

having count(\*)>=amount;

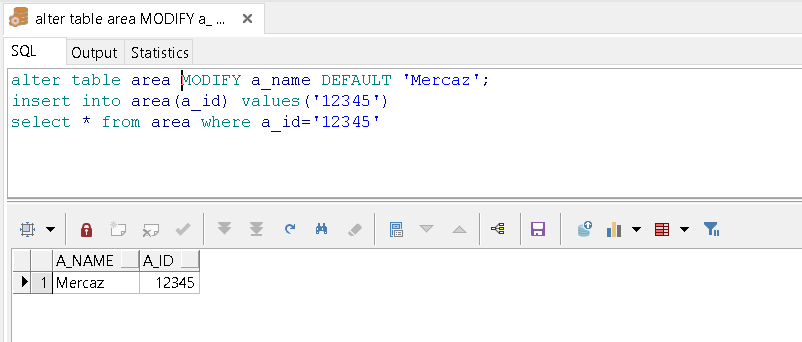
לאחר יצירת האינדקס:



לאחר יצירת האינדקס:

אילוצים

* Default- בנינו ערך ברירת מחדל ‘Mercaz’ לעמודת שם בטבלת אזור.

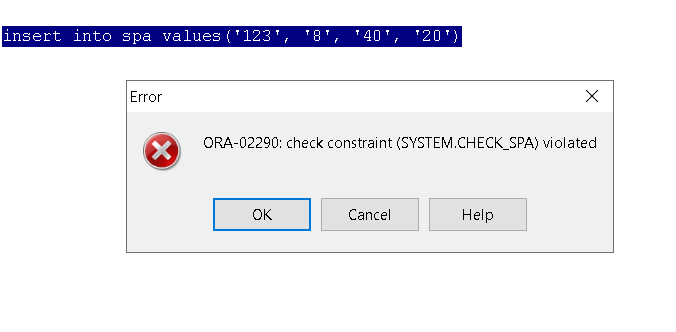


* check- בנינו אילוץ בטבלת spa שעמודה ENTRENCE\_AGE חייבת להיות בתווך הערכים 6-30.

alter table spa

add constraint check\_spa CHECK(ENTRENCE\_AGE between 6 and 30);

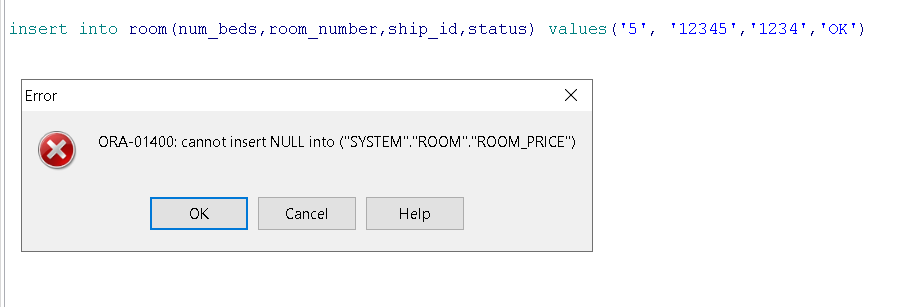
בדיקה:



* Not null- בנינו אילוץ לטבלת room עבור עמודת price שחייב להיות בה ערך שונה מnull.

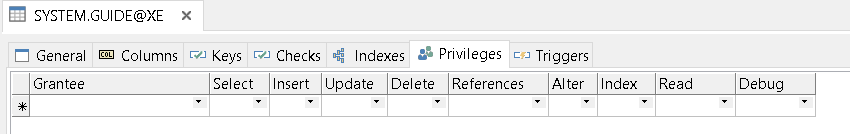
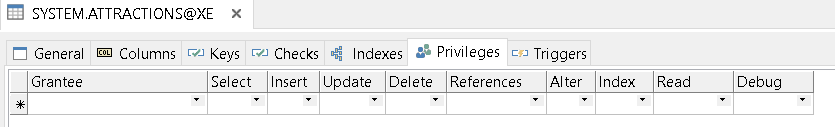
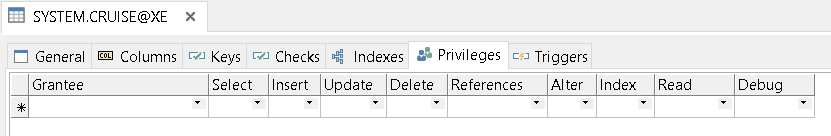
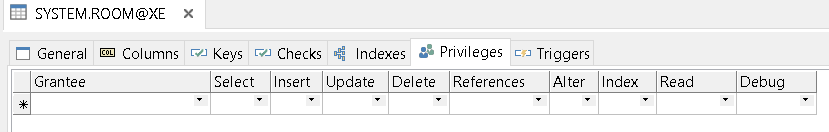
ALTER TABLE room

MODIFY ROOM\_PRICE INTEGER NOT NULL;



יצירת משתמש והרשאות גישה

הרשאות של הטבלאות לפני ניתנת הרשאות גישה



יצירת משתמש בשם תהילה עם סיסמה 123 עם הרשאות גישה שונות.

alter session set "\_ORACLE\_SCRIPT"=true;

create user Tehila identified by 123; *-- Create a user named Tehila*

*--הרשאות גישה*

grant create session to Tehila ;*-- הרשאה להתחברות*

grant create table to Tehila ; *-- הרשאה ליצירת טבלאות*

grant all on ATTRACTIONS to Tehila ;

grant update on invitation to Tehila ;

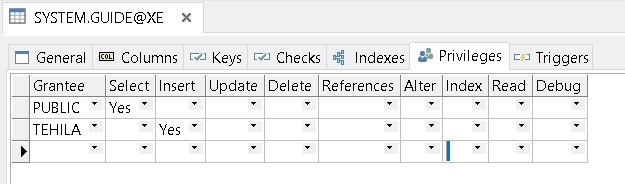
grant update on my\_path to Tehila ;

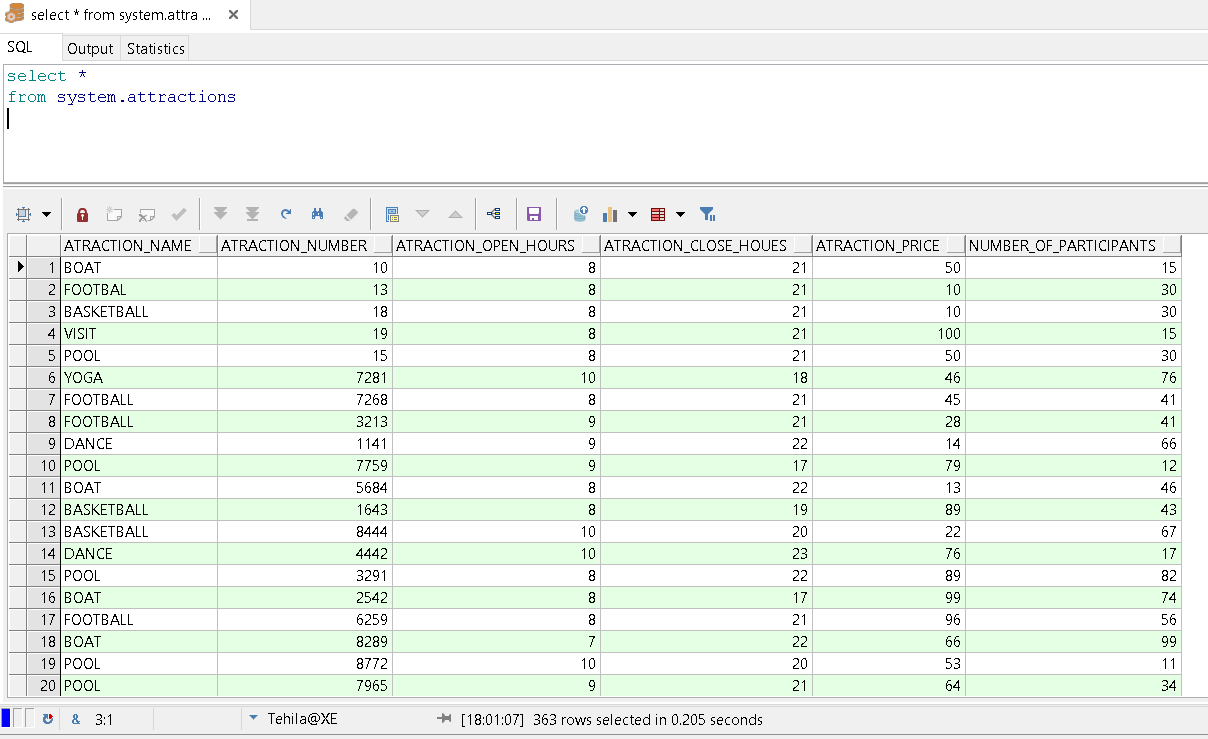
grant select on cruise to Tehila ;

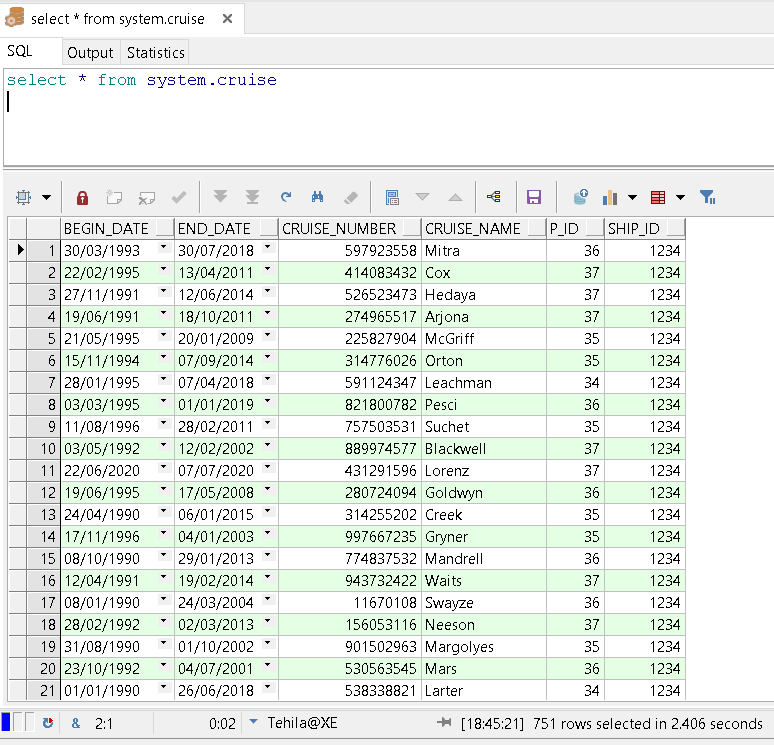
grant delete on room to Tehila ;

grant insert on guide to Tehila ;

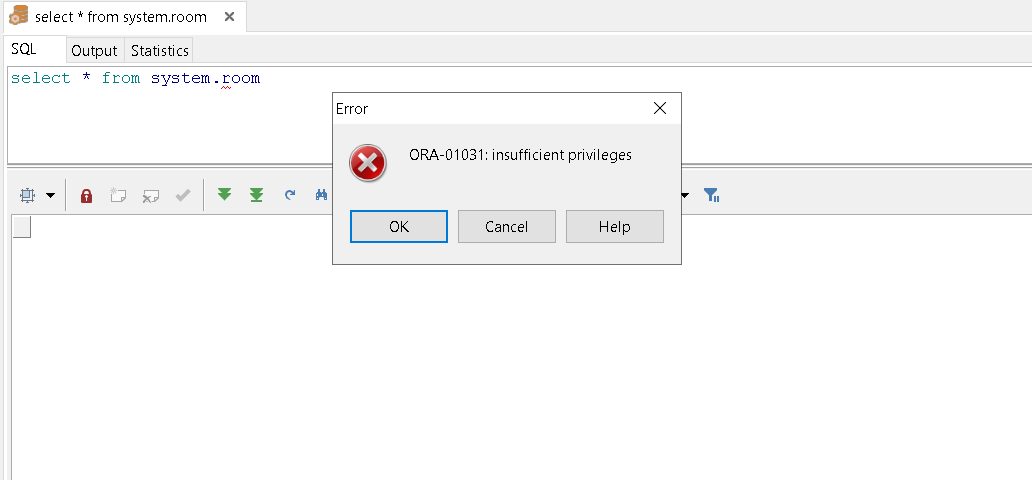
grant select on guide to public ;*-- הרשאה של שליפה לכל המשתמשים*

לאחר נתינת הרשאות:



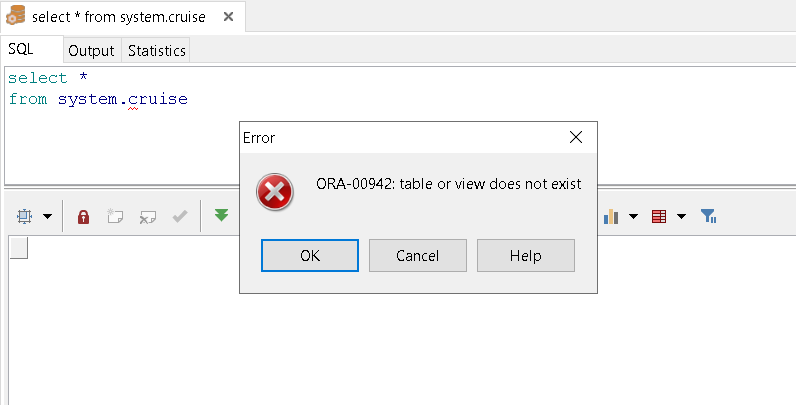


אין הרשאת גישה לבחירה מטבלת room



לקיחת הרשאה:

revoke select on cruise from Tehila ;



Views

1. יצירת מבט פשוט על האטרקציות שהם יוגה או ריקוד.

Create view view\_attraction As

select \*

from attractions

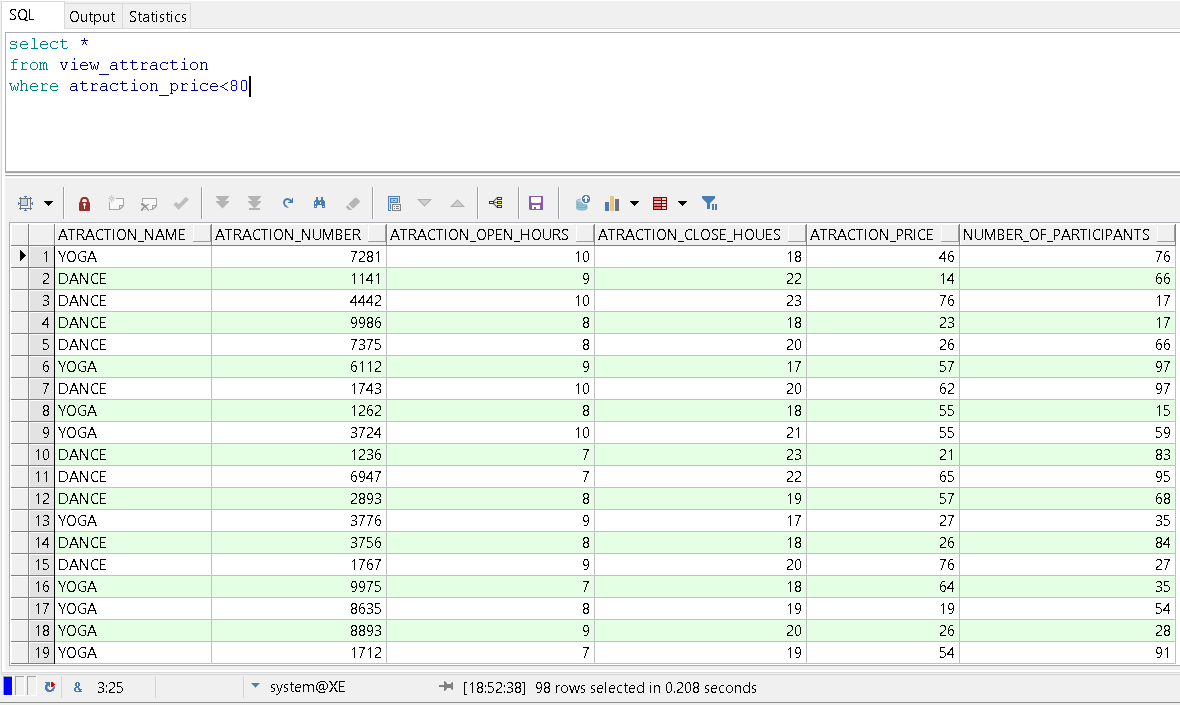
where atraction\_name='YOGA' or atraction\_name='DANCE'

שאילתה 1:

select \*

from view\_attraction

where atraction\_price<80



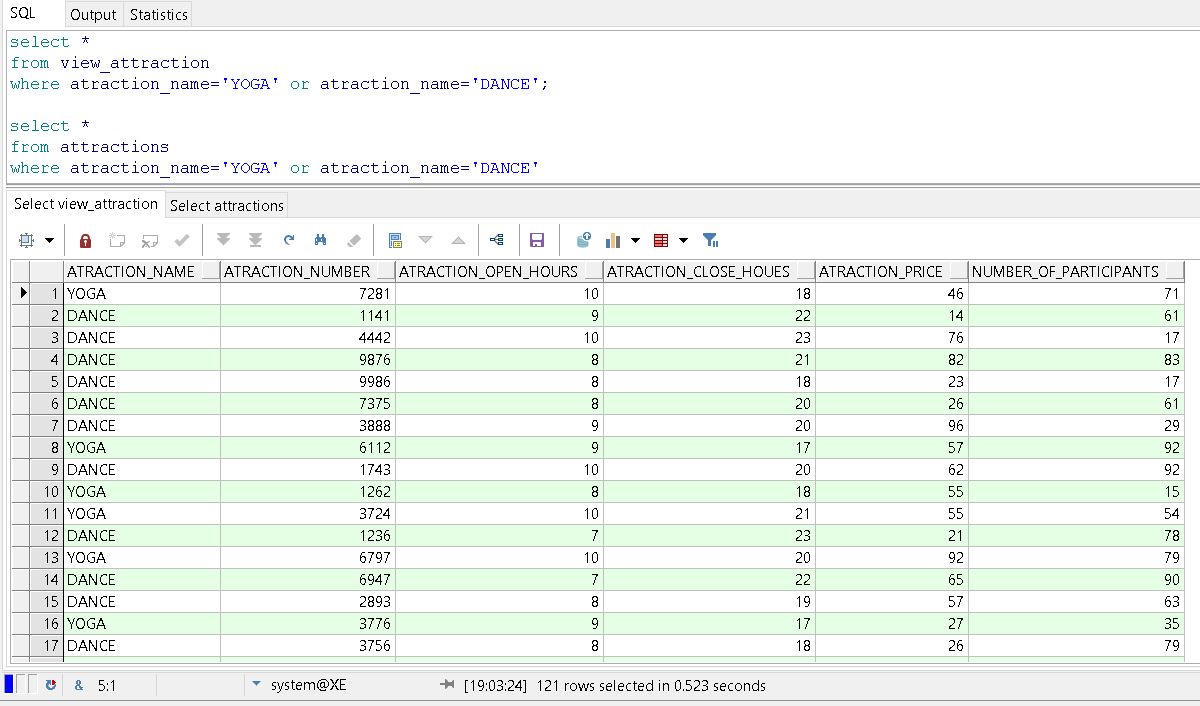
שאילתה 2:

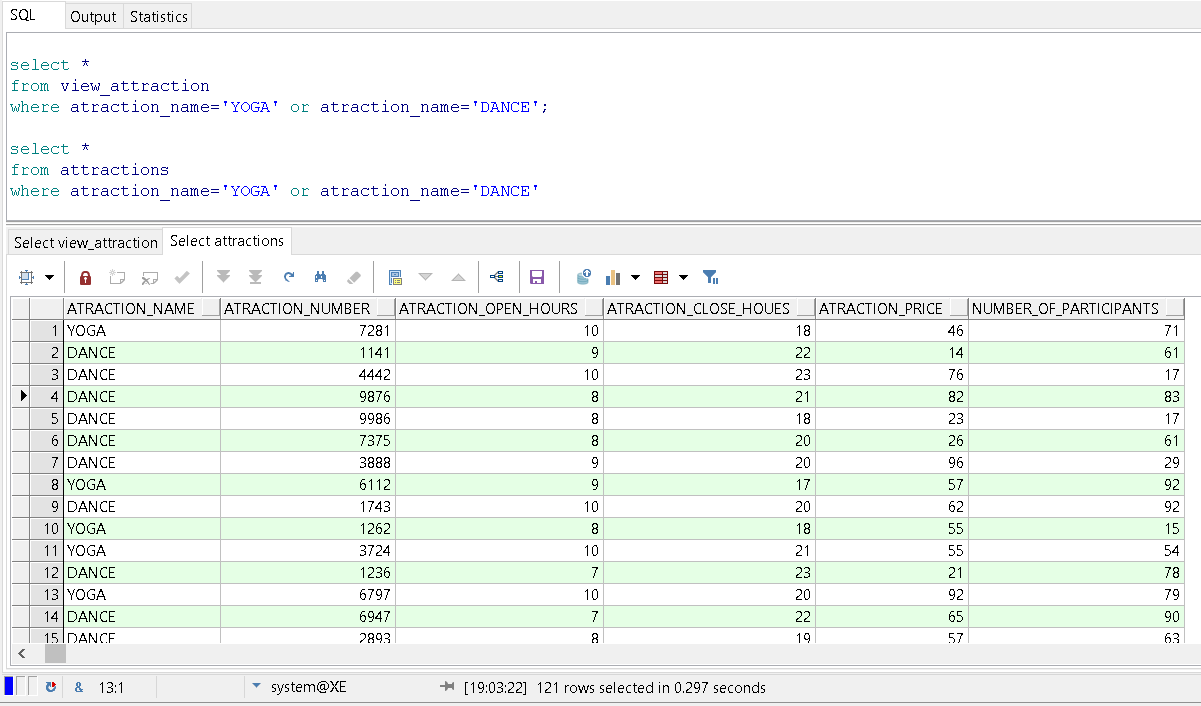
update view\_attraction

set number\_of\_participants=number\_of\_participants-5

where number\_of\_participants>50

העידכון במבט גורם לשינוי גם בטבלת המקור





1. יצירת מבט מורכב על טבלת הלקוחות, הערים והאזורים.

create view view\_a\_c\_c As

select city.c\_id As city\_id, city.c\_name as city\_name ,area.a\_id As area\_id , area.a\_name As area\_name , c\_tz, client.c\_name as client\_name, b\_date, m\_g\_id

from city, client, area

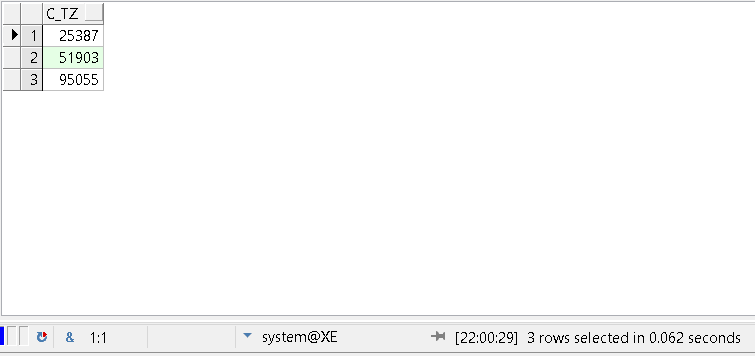
where city.c\_id=client.c\_id and city.a\_id = area.a\_id

שאילתה 1: השאילתה מוצאת את כל הלקוחות הגרים באזור הצפון ונולדו בשנת 1999.

select c\_tz

from view\_a\_c\_c

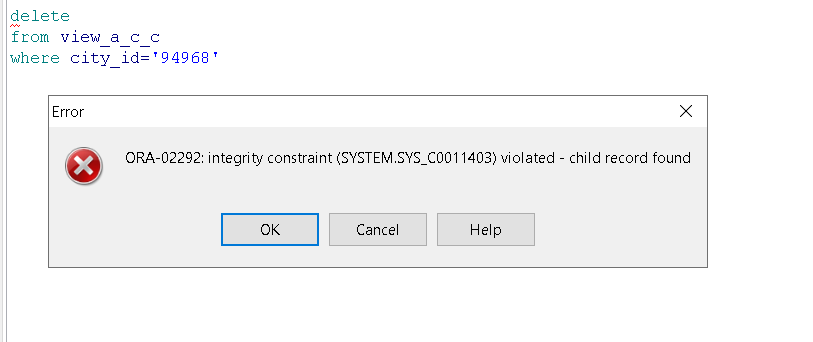
where area\_name ='south' and b\_date between to\_date('01/01/1999','dd/mm/yyyy') and to\_date('31/12/1999','dd/mm/yyyy')



שאילתה 2: השאילתה מוחקת הלקוחות שגרים באזור '94968'.

delete

from view\_a\_c\_c

where city\_id='94968'

המחיקה לא עובדת מכיוון שאין אפשרות לבצע עידכון ומחיקה על מבט מורכב.

שלב מס' 4

שאילתות עם פרמטרים

1. השאילתה צריכה לקבל פרמטר שהוא שם המסלול, השאילתה מחזירה את כל מספרים המזהים של הלקוחות שטיילו במסלול.

select C.c\_tz

from client C

where C.m\_g\_id not in

(select I.m\_g\_id

from invitation I

where I.i\_id in

(select B.i\_id

from belongs\_to B

where B.p\_id in

(select p\_id

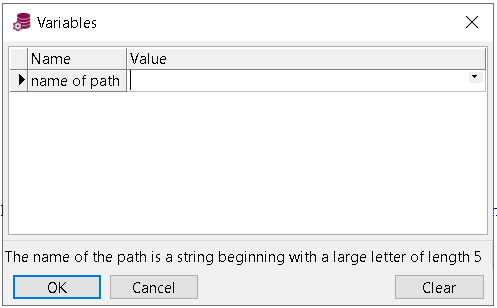
from my\_path P

where p.p\_name = &<name= "name of path"

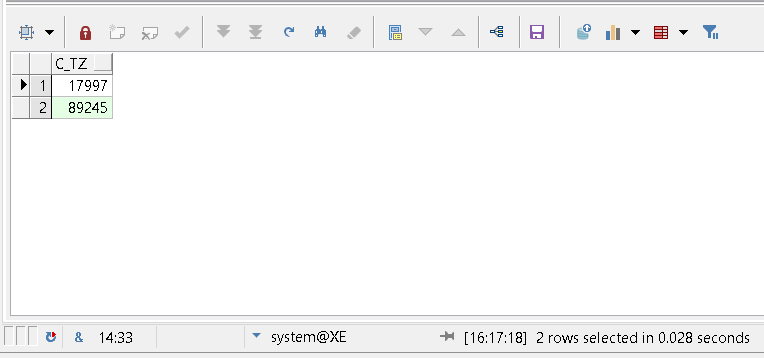
hint="The name of the path is a string beginning with a large letter of length 5"

type="string"

required=true>)))



עבור הפרמטר “Onfud” תוצאת השאילתה:



1. השאילתה צריכה לקבל 2 פרמטרים שהם תאריכים של תחילה וסיום של קרוז, השאילתה מחזירה את מס' הקרוז, מס' מזהה של הספינה ומספר החדרים הפנויים שנותרו. לשאילתה יש ערך ברירת מחדל של התאריכים לחודש הקרוב.

select cruise\_number, ship\_id, count(\*) as num\_of\_empty\_room

from cruise c natural join ship s natural join room r

where c.begin\_date >= to\_date(&<name= "begin date"

type=string

default ="22/06/2020">,'dd/mm/yyyy')

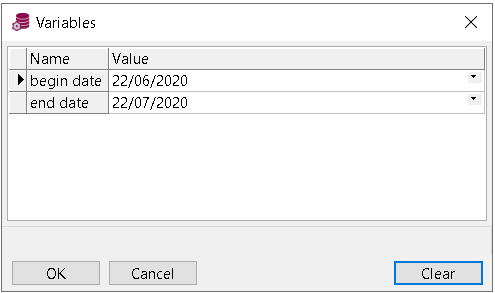
and c.end\_date <= to\_date(&<name= "end date"

type="string"

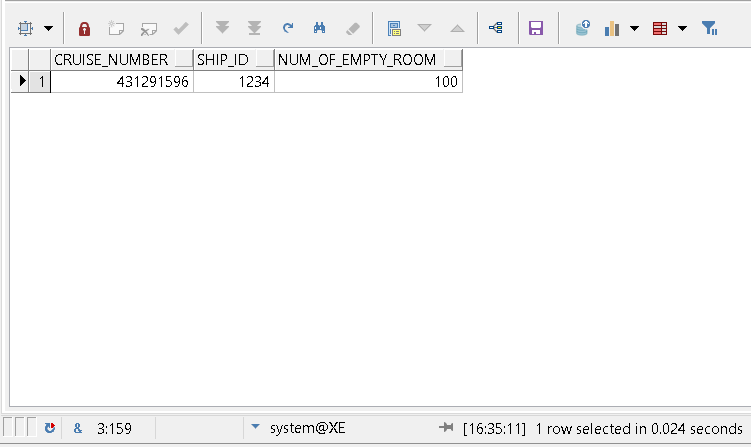
default ="22/07/2020">,'dd/mm/yyyy')

and r.status='OK'

group by cruise\_number, ship\_id



עבור ערך ברירת מחדל (22/06/2020, 22/07/2020) תוצאת השאילתה:



1. השאילתה צריכה לקבל פרמטר של רמת קושי של טיול מתוך רשימה נתונה, השאילתה מחזירה את ת.ז. והשם של המדריך שהדריך בדיוק 2 טיולים באותה רמת קושי.

select g\_id, e\_name

from difficulty natural join my\_path natural join belongs\_to natural join invitation natural join guide natural join employee

where d\_name = &<name="difficulty"

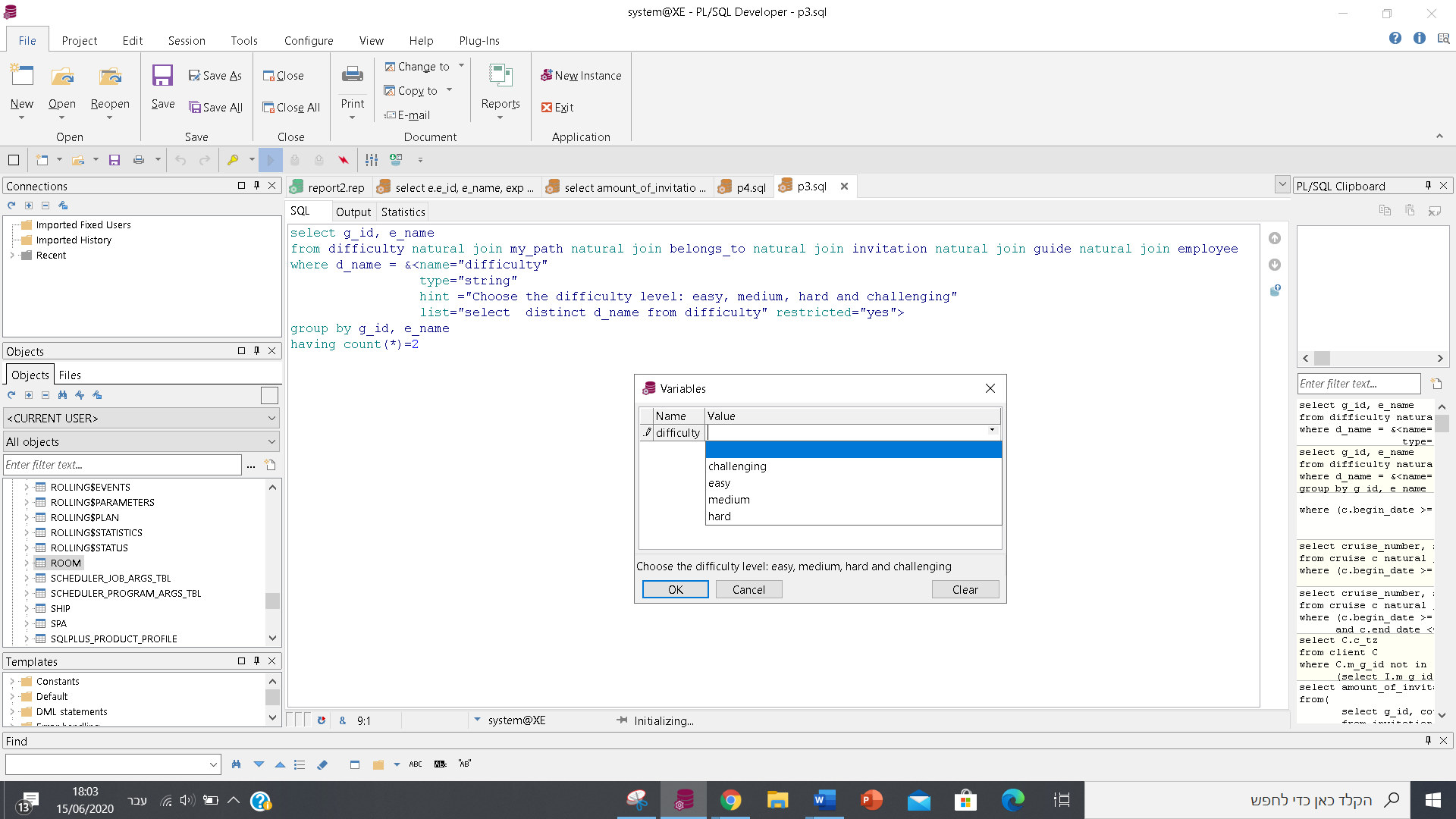
type="string"

hint ="Choose the difficulty level: easy, medium, hard and challenging"

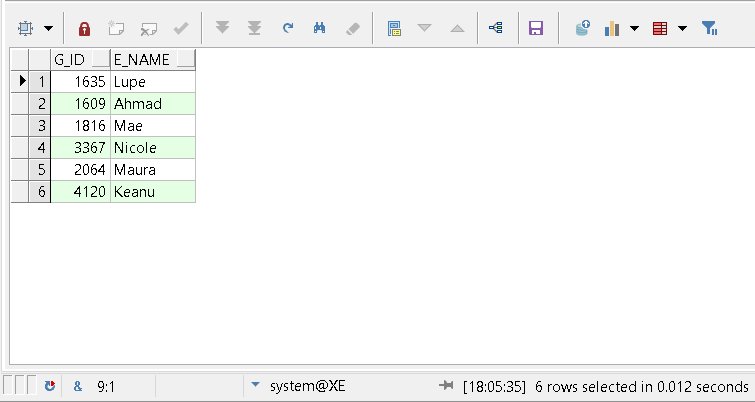
list="select distinct d\_name from difficulty" restricted="yes">

group by g\_id, e\_name

having count(\*)=2



תוצאת השאילתה עבור בחירה של רמת קושי “medium”:



1. השאילתה צריכה לקבל פרמטר (אפשר גם יותר מפרמטר אחד) של ת.ז. של לקוח, ניתן לבחור שמות של לקוחות מתוך רשימה שמסודרת לפי סדר האותיות והיא מכניסה את הת.ז. של הלקוחות שנבחרו, השאילתה מחזירה את שמות האטרקציות שבהם היו הלקוחות הללו.

select distinct atraction\_name

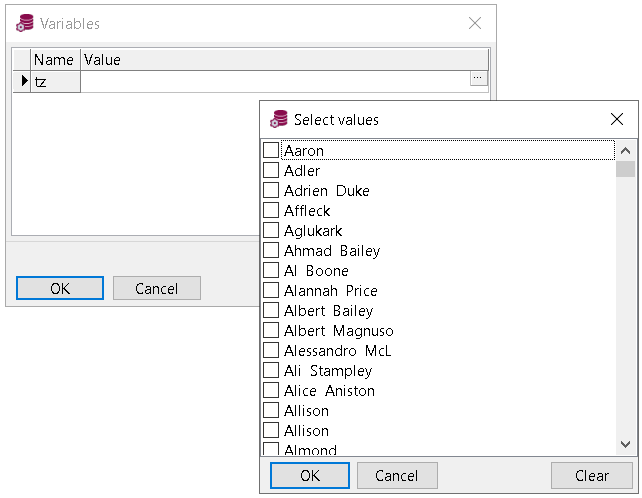
from client natural join participate natural join attractions

where c\_tz in (&<name= "tz"

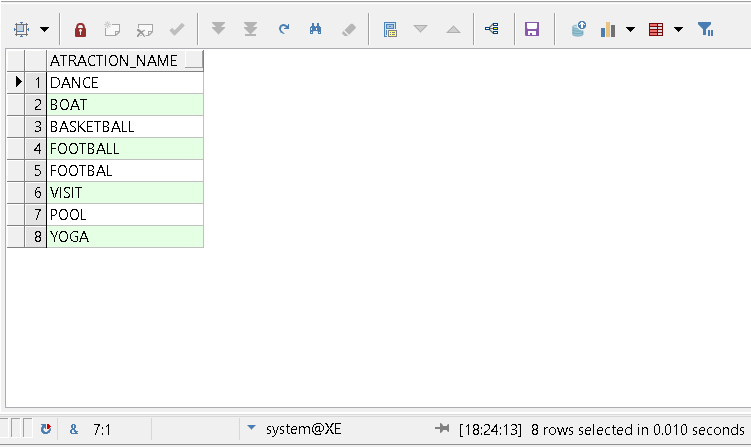
list="select c\_tz, c\_name from client order by c\_name"

description="yes"

multiselect="yes">)



תוצאת השאילתה עבור הפרמטר “Aaron”:



דוחות

דוח 1: הדוח מראה עבור כל עיר את כמות נסיעות הקרוזים המתחילה בעיר הזו.

השאילתה:

select begin\_city, count(\*)

from path

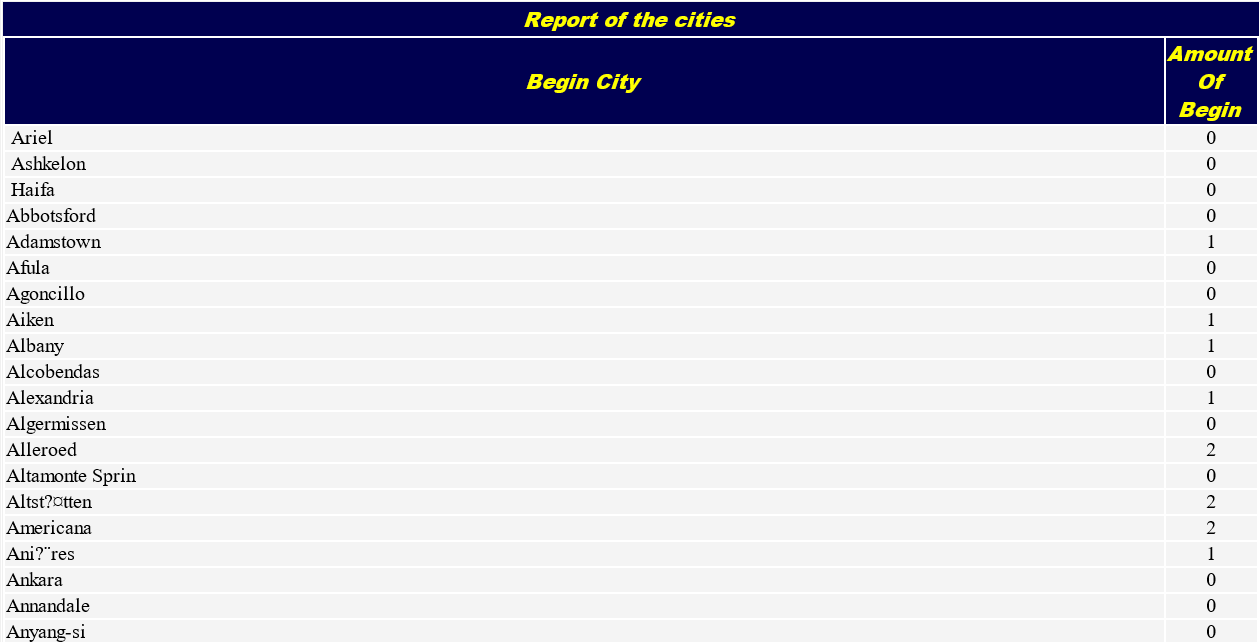
group by begin\_city

UNION

select c\_name, 0

from city

where c\_name NOT IN (SELECT begin\_city FROM path)



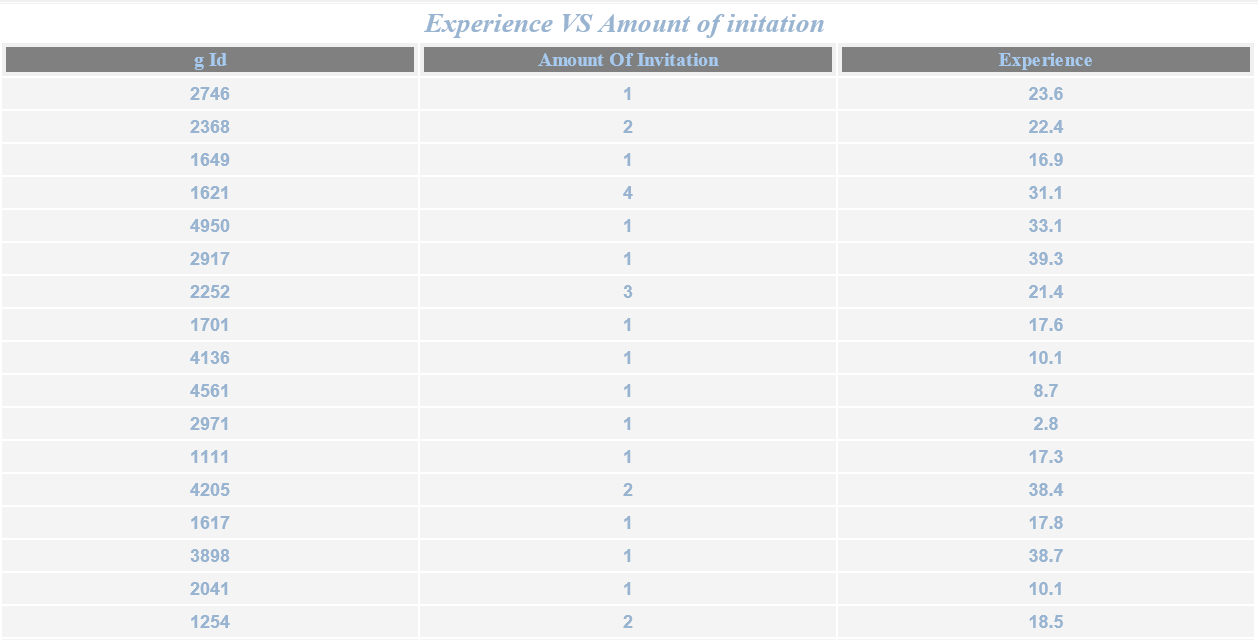
דוח 2: הדוח מראה עבור כל מדריך את מספר שנות ניסיון ומספר הדרכות טיולים בהם הוא הדריך.

השאילתה:

select g\_id, count(\*) As amount\_of\_invitation, experience

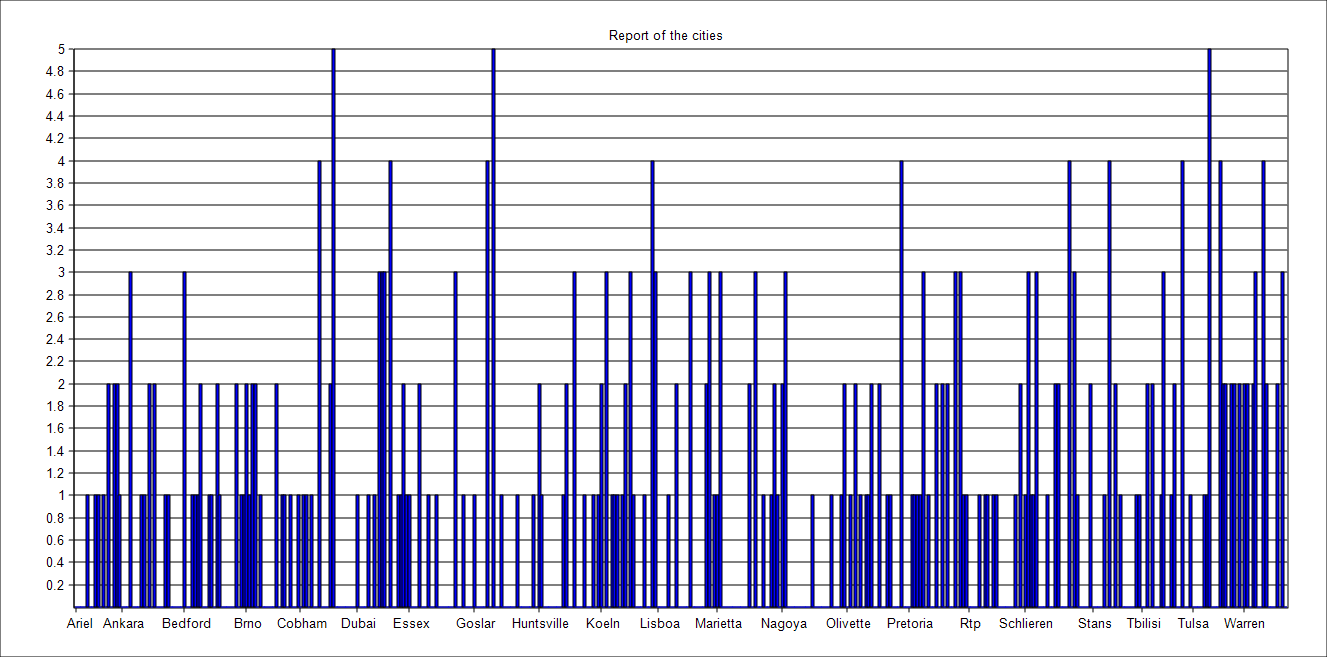
from invitation natural join guide

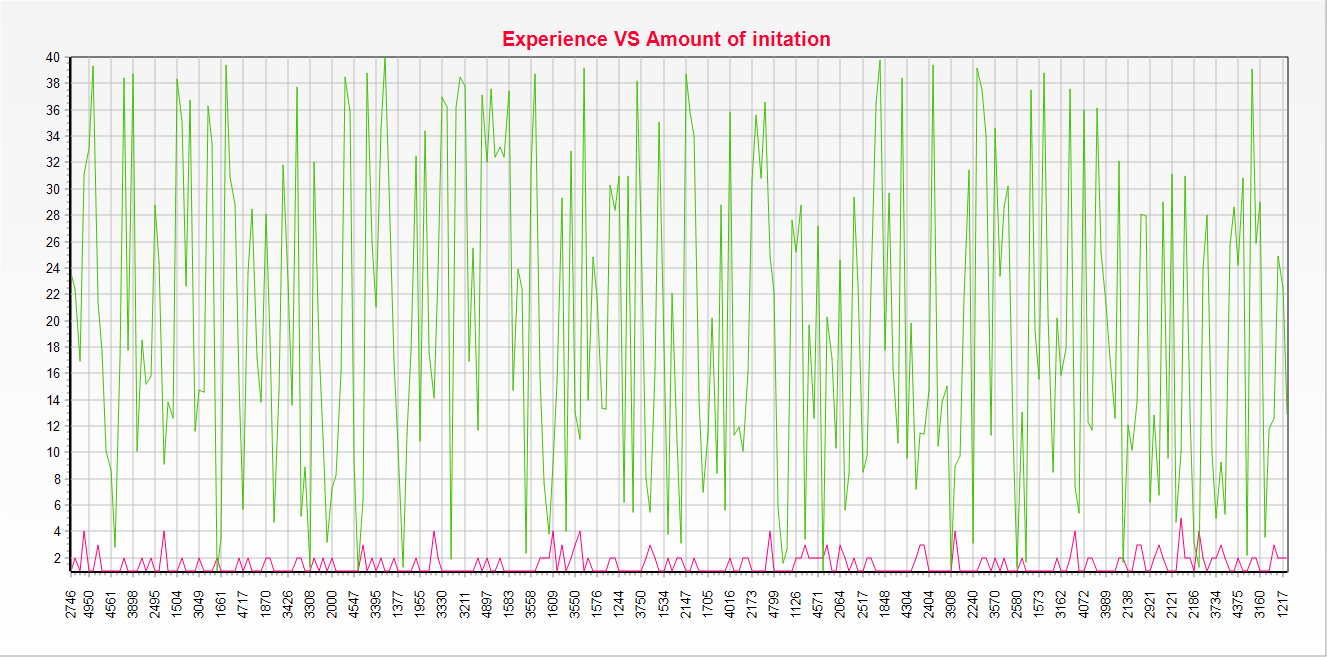
group by g\_id, experience



גרפים

גרף עבור דוח 1:



גרף עבור דוח 2:

פונקציות

1. הפונקציה מקבלת ת.ז., גיל ומצב של לקוח הרשום לטיול קבוצתי. הפונקציה מחשבת את המחיר של הטיול עבור הלקוח לפי נתוני המצב והגיל.

לסטודנט יש 20% הנחה, לחייל יש 70% הנחה ועבור אזרח ותיק ונער שמתחת גיל 18 יש 50% הנחה.

פונקציית עזר המחשבת את מס' המזהה של קבוצה עבור הלקוח אליה הוא משתייך:

create or replace noneditionable function numberOfGroup(clientId in integer) return integer is

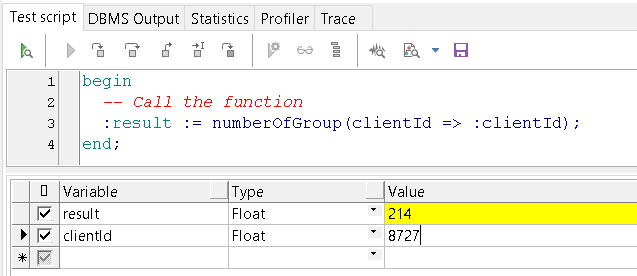
FunctionResult integer;

begin

select m\_g\_id into FunctionResult from client where c\_tz = clientId;

return(FunctionResult);

end numberOfGroup;

עבור הלקוח עם ת.ז. 8727:

פונקציית עזר שמחזירה את המחיר הקבוצתי המקסימלי הכולל:

create or replace noneditionable function generalHighPrice(idOfGroup in integer) return integer is

FunctionResult integer;

cursor priceInGroup is select price from invitation where m\_g\_id = idOfGroup;

tempPrice integer;

begin

FunctionResult := 0;

if not priceInGroup%isopen

then

open priceInGroup;

end if;

fetch priceInGroup into tempPrice;

while priceInGroup%found

loop

if FunctionResult < tempPrice

then

FunctionResult := tempPrice;

end if;

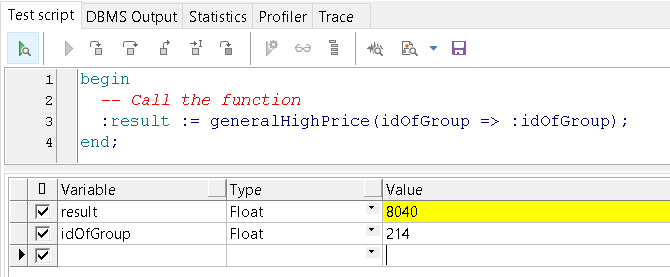
fetch priceInGroup into tempPrice;

end loop;

return(FunctionResult);

end generalHighPrice;

עבור הקבוצה 214 המחיר המקסימלי של טיול הוא 8040:



פונקציה המחשבת עבור כל לקוח את המחיר עבור טיול לפי הנתונים הספציפיים שלו:

create or replace noneditionable function costOfTrip(clientid in integer, status in string, age in float) return float is

FunctionResult float;

cursor amountInGroup is select c\_tz from client where m\_g\_id = numberOfGroup(clientid);

amountOfPeople integer := 0;

tempclient client.c\_tz%type;

begin

if not amountInGroup%isopen

then

open amountInGroup;

end if;

fetch amountInGroup into tempclient;

while amountInGroup%found

loop

amountOfPeople := amountOfPeople + 1;

fetch amountInGroup into tempclient;

end loop;

FunctionResult := generalHighPrice(numberOfGroup(clientid)) / amountOfPeople;

if status = 'student'

then

FunctionResult := FunctionResult\*0.80;

elsif status = 'Soldier'

then

FunctionResult := FunctionResult\*0.70;

elsif status = 'veteran'

then

FunctionResult := FunctionResult\*0.50;

elsif age <= 18

then

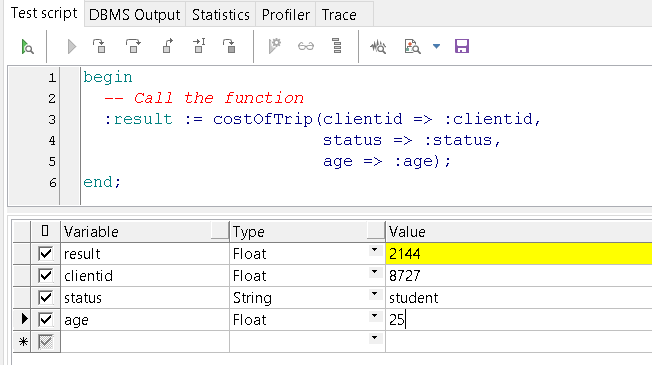
FunctionResult := FunctionResult\*0.50;

end if;

return(FunctionResult);

end costOfTrip;

עבור לקוח עם ת.ז. 8727, status=student, age=25:



המחיר עבור הלקוח הנ"ל הוא 2114 כי מחיר הטיול עבור הקבוצה הוא 8040, בקבוצה יש 3 משתתפים לכן 2680=8040/3 , מכיוון שהלקוח הנ"ל הוא סטודנט יש לו 20% הנחה ולכן המחיר לטיול עבורו הוא: 2144=2680/0.8 .

1. הפונקציה מקבלת שם של קרוז והיא מחזירה את הרווחים שיש מאותם חדרים תפוסים באותו קרוז.

create or replace noneditionable function profitOfCruise(cruiseName in string) return integer is

FunctionResult integer;

cursor passOnTable is select room\_price from room natural join cruise where cruise\_name = cruiseName and status = 'BUSY';

tempRow passOnTable%rowtype;

begin

FunctionResult := 0;

for tempRow in passOnTable

loop

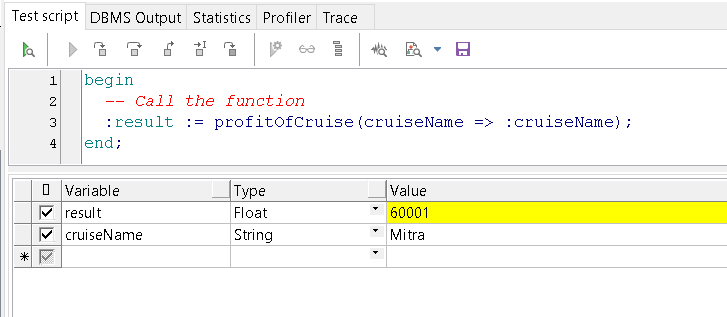
FunctionResult := FunctionResult + tempRow.Room\_Price;

end loop;

return(FunctionResult);

end profitOfCruise;

עבור הקרוז ‘Mitra’ הרווח הכללי הוא 60001₪ .

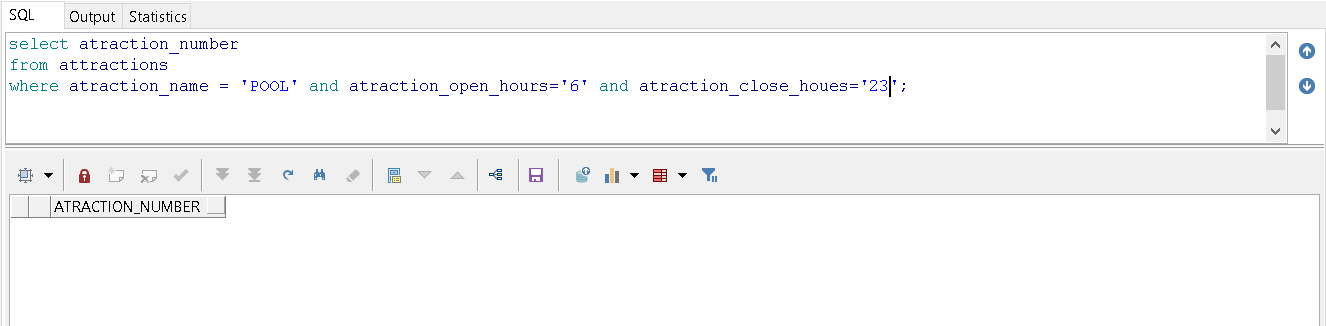


פרוצדורות

1. הפרוצדורה מקבלת את כל הפרטים עבור הוספת אטרקציה. האטרקציה מתווספת רק אם היא עומדת בקריטריונים הבאים:

סה"כ מס' האטרקציות מאותו סוג אינו עולה על 60 אטרקציות, אם האטרקציה היא מסוג: ‘POOL’, ‘DANCE’, ‘YOGA’ אז מס' המשתתפים יהיה 50-100 ואילו בשאר האטרקציות מס' המשתתפים יהיה 20-50, מחיר האטרקציה לא יהיה גבוה מ150₪. תנאי נוסף שאין אטרקציה עם אותו שם ששעות הפתיחה והסגירה זהות. אם האטרקציה עומדת בכל הקריטריונים הללו היא מתווספת לטבלה ומודפסת ההודעה: 'Attraction successfully added!!!' אחרת רק מודפסת ההודעה: 'Unfortunately, Your attraction does not fit our criteria.'

לפני הרצת הפרוצדורה הרצנו את השאילתה:



הרצת הפרוצדורה:

create or replace procedure addNewAttraction(attractionID in attractions.atraction\_number%type, attractionOpen in attractions.atraction\_open\_hours%type, attractionClose in attractions.ATRACTION\_CLOSE\_HOUES%type, attractionName in attractions.atraction\_name%type, attractionPrice in attractions.atraction\_price%type, numberOfParticipant in attractions.number\_of\_participants%type) is

cursor sumOfAttraction is select atraction\_number from attractions where atraction\_name = attractionName;

cursor hoursOfAttraction is select atraction\_number from attractions where atraction\_name = attractionName and attractionOpen = atraction\_open\_hours and attractionClose = atraction\_close\_houes;

mySum integer := 0;

tempAttraction attractions.atraction\_number%type;

hourscursor attractions.atraction\_number%type;

flag boolean;

begin

flag := false;

if not sumOfAttraction%isopen

then

open sumOfAttraction;

end if;

fetch sumOfAttraction into tempAttraction;

while sumOfAttraction%found

loop

mySum := mySum + 1;

fetch sumOfAttraction into tempAttraction;

end loop;

if mySum < 60 then

if attractionName = 'POOL' or attractionName = 'DANCE' or attractionName = 'YOGA' then

if numberOfParticipant >= 50 and numberOfParticipant <= 100 then

flag := true;

end if;

elsif numberOfParticipant >= 20 and numberOfParticipant <= 50 then

flag := true;

end if;

end if;

if attractionPrice > 150 then

flag := false;

end if;

if not hoursOfAttraction%isopen

then

open hoursOfAttraction;

end if;

fetch hoursOfAttraction into hourscursor;

while hoursOfAttraction%found

loop

flag := false;

fetch hoursOfAttraction into hourscursor;

end loop;

if flag = true then

insert into attractions values(attractionName, attractionID, attractionOpen, attractionClose, attractionPrice, numberOfParticipant);

commit;

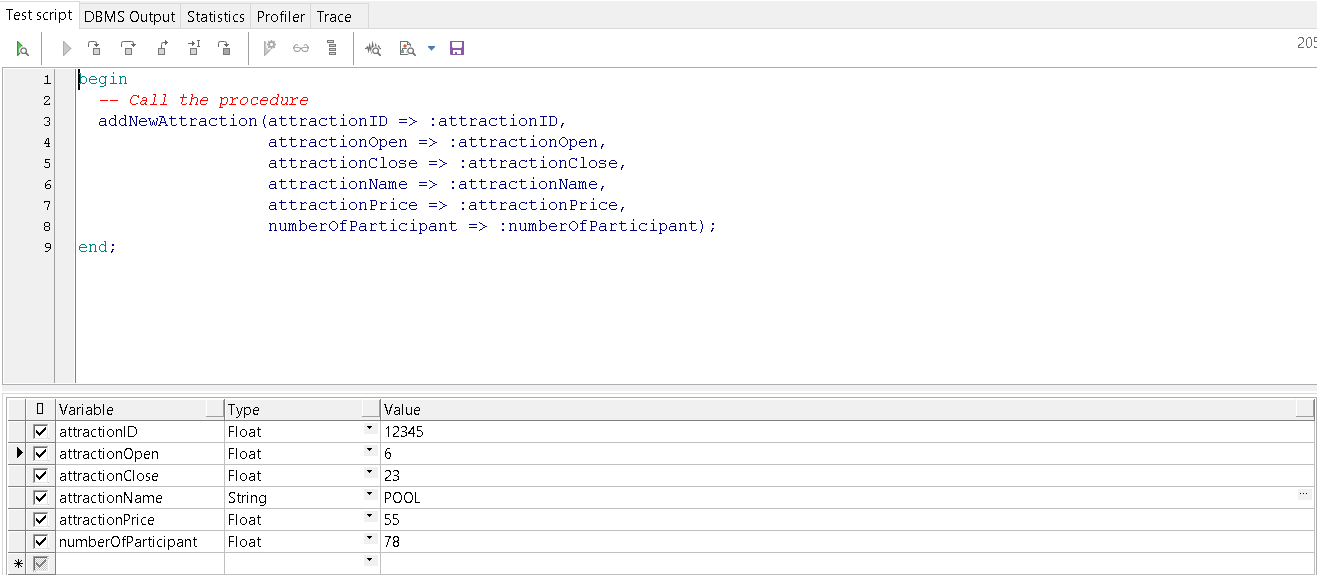
dbms\_output.put\_line('Attraction successfully added!!!');

else

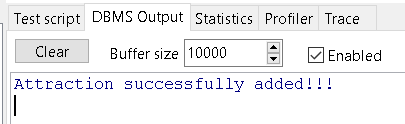
dbms\_output.put\_line('Unfortunately, Your attraction does not fit our criteria.');

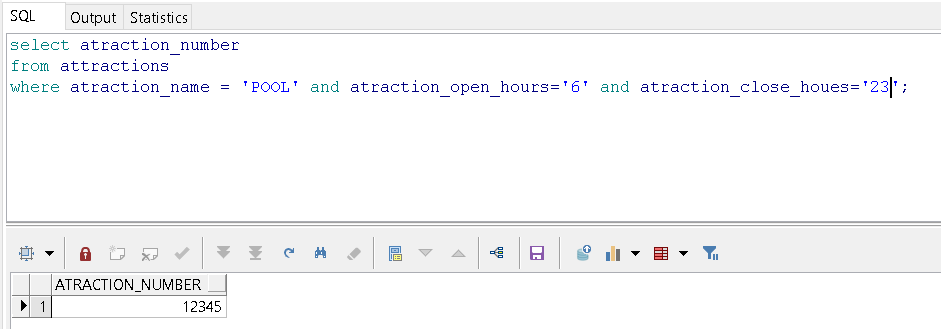
end if;

end addNewAttraction;

עם הפרמטרים:

תוצאה:



אכן האטרקציה התווספה לטבלה:

1. הפרוצדורה מקבלת מהמשתמש אזור בו הוא רוצה לטייל וזמן מקסימלי, הפרוצדורה מדפיסה את שמות הטיולים ואת הזמן שאורך לטייל בהם הנמצאים באזור ובזמן שהקציב המשתמש.

הפרוצדורה:

create or replace procedure InterestingTrips(areaInCountry in string, maxTime in float) is

cursor tripsInArea is select p\_name, p\_time from area natural join my\_path where a\_name = areaInCountry and p\_time <= maxTime;

tempNameTrip tripsInArea%rowtype;

begin

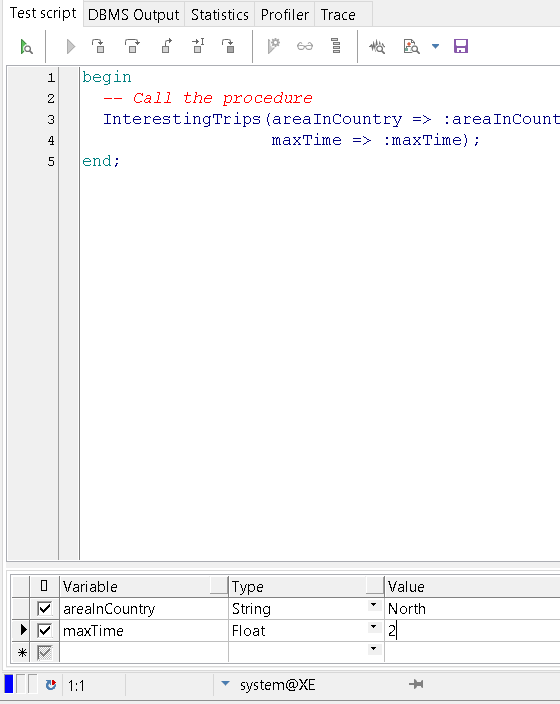
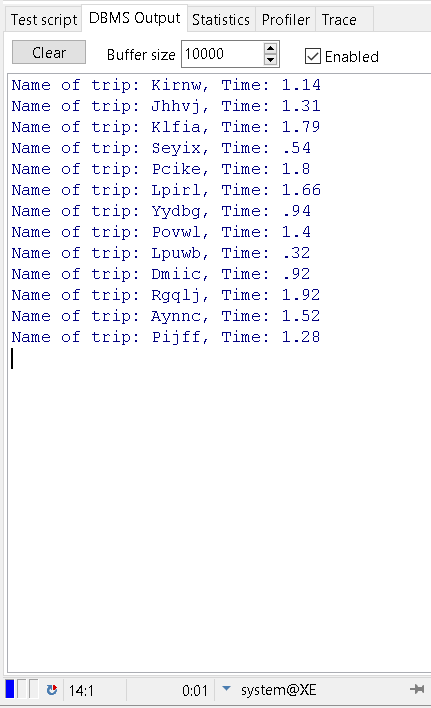
for tempNameTrip in tripsInArea

loop

dbms\_output.put\_line('Name of trip: ' || tempNameTrip.p\_Name ||', Time: ' || tempNameTrip.p\_Time);

end loop;

end InterestingTrips;

 הרצה עבור הפרמטרים: תוצאה:

TRIGGERS

1. טריגר experienceOfGuid

בעת הוספת או עדכון מדריך, הניסיון של המדריך מחושב לפי מס' השנים שמהם התחיל לעבוד עד עתה.

create or replace trigger experienceOfGuid

before insert or update

on guide

for each row

declare

*-- local variables here*

temp float;

employeeHireDate employee.hire\_date%type;

begin

select employee.hire\_date into employeeHireDate from employee where employee.e\_id = :new.e\_id;

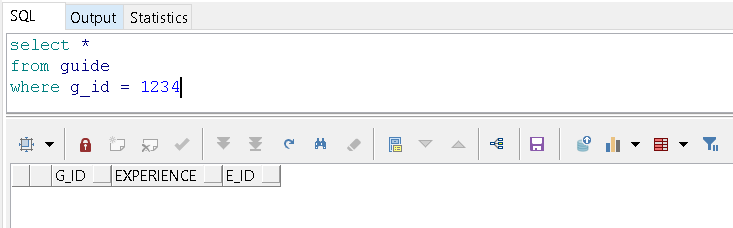
temp := (to\_date(sysdate, 'dd/mm/yyyy')-to\_date(employeeHireDate, 'dd/mm/yyyy'))/365;

temp := substr(temp,1,5);

:new.experience := temp;

end experienceOfGuid;

מדריך עם ת.ז. 1234 לא קיים:



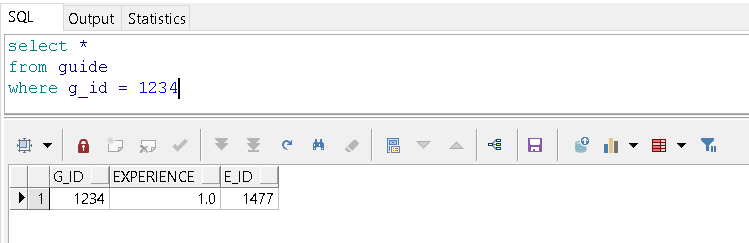
הוספת עובד:

insert into employee values( 1477, 'Ruty', to\_date('22/06/2019', 'dd/mm/yyyy'), 140);

הוספת מדריך עם ניסיון של 5 שנים:

insert into guide values( 1234, 5, 1477);

אך בפועל העובד התחיל לעבוד לפני שנה לכן experience מתעדכן אוטומטי ל1:



1. טריגר invitationOfCruise

כאשר מכניסים שורה לטבלת קרוז אם תאריך תחילת הנסיעה הוא לפני התאריך של היום אז תאריך תחילת הנסיעה מתעדכן ישירות לתאריך של היום.

create or replace trigger invitationOfCruise

before insert

on cruise

for each row

declare

*-- local variables here*

begin

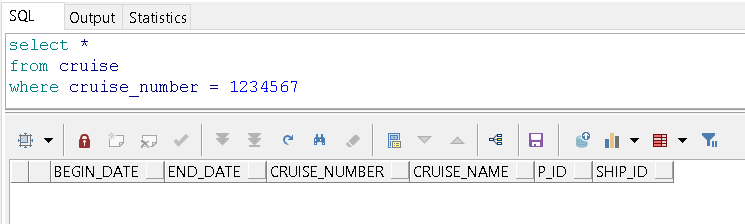
if to\_char(:new.begin\_date) < to\_char(sysdate, 'dd/mm/yyyy') then

:new.begin\_date := to\_date(sysdate, 'dd/mm/yyyy');

end if;

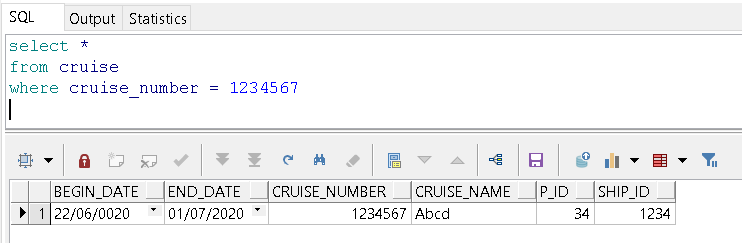
end invitationOfCruise;

לפני ההכנסה:



לאחר הכנסת שורה עם תארי של תחילת נסיעה ‘01/06/2020’:

insert into cruise values(to\_date('01/05/2020', 'dd/mm/yyyy'), to\_date('01/07/2020', 'dd/mm/yyyy'), 1234567, 'Abcd', 34, 1234);

לאחר ההכנסה התאריך הוא התאריך היום: