

**מיני פרויקט בבסיסי נתונים**

**תהילה טיישר / רינת מועלם**

**319053039 / 208316232**

**אפריל 2020**

בס"ד

תוכן עניינים:

[שלב מס' 1 3](#_Toc38918564)

[תיאור מילולי של הארגון 3](#_Toc38918565)

[תרשים ERD 3](#_Toc38918566)

[מילון 4](#_Toc38918567)

[הטבלאות המתקבלות 5](#_Toc38918568)

[בדיקה והסבר שכל הטבלאות מנורמלות ברמת 3CNF 5](#_Toc38918569)

[תרשים DSD 6](#_Toc38918570)

[מעבר מERD לטבלאות 6](#_Toc38918571)

[יצירת טבלאות 7](#_Toc38918572)

[אכלוס טבלאות 9](#_Toc38918573)

[**פקודת insert 9**](#_Toc38918574)

[**קליטה מקובץ TXT 10**](#_Toc38918576)

[**ODBC Importer 13**](#_Toc38918577)

[**פקודת datagenerator 17**](#_Toc38918579)

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)

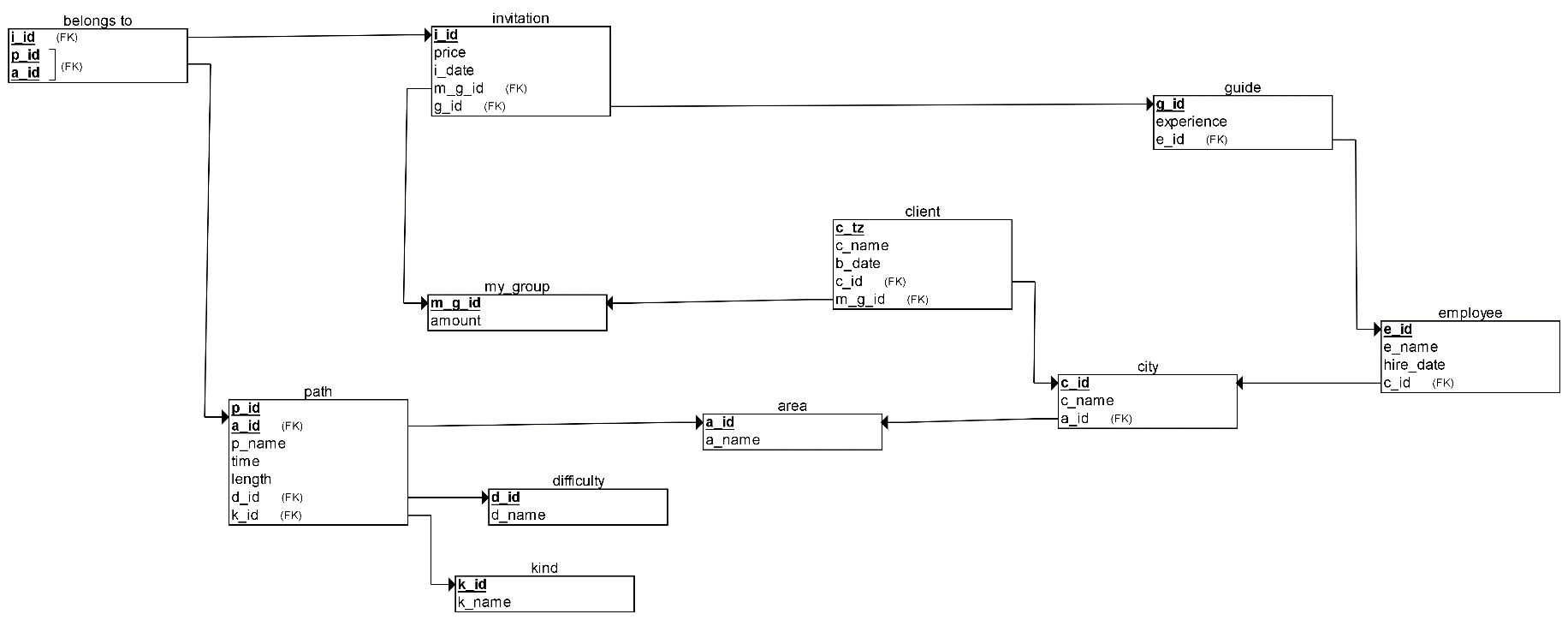
Invitation (i\_id, price, i\_date, m\_g\_id, g\_id)

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),

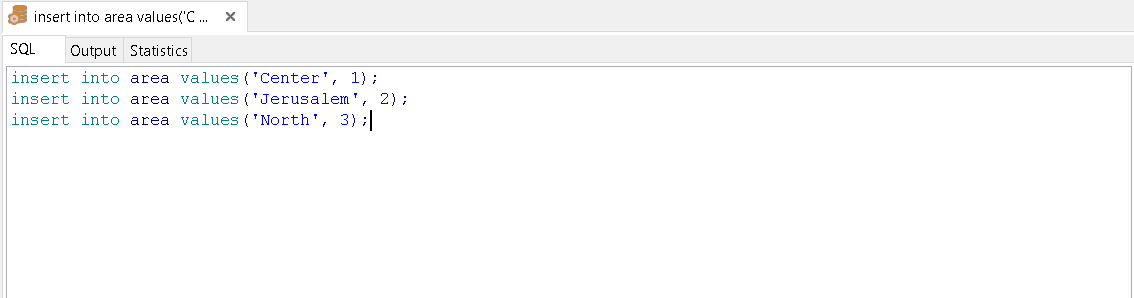
FOREIGN KEY (i\_id) REFERENCES invitation(i\_id),

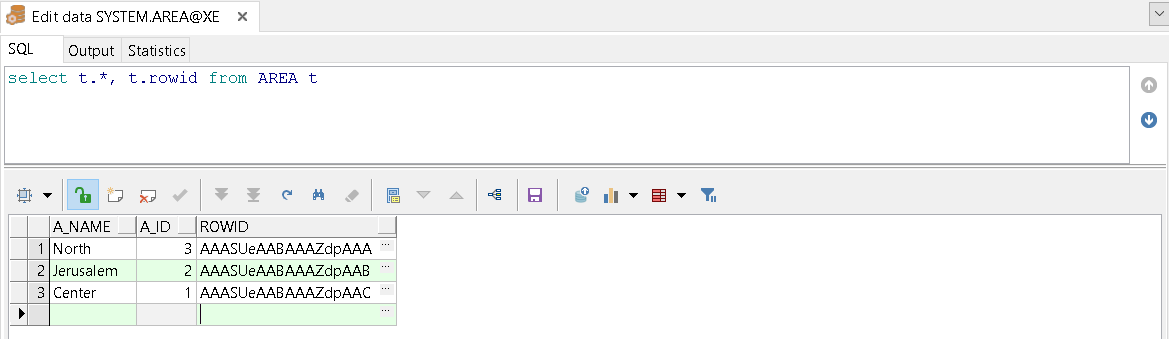
FOREIGN KEY (p\_id, a\_id) REFERENCES path(p\_id, a\_id)

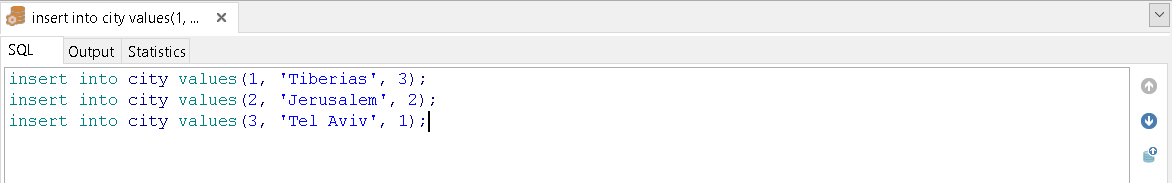
);

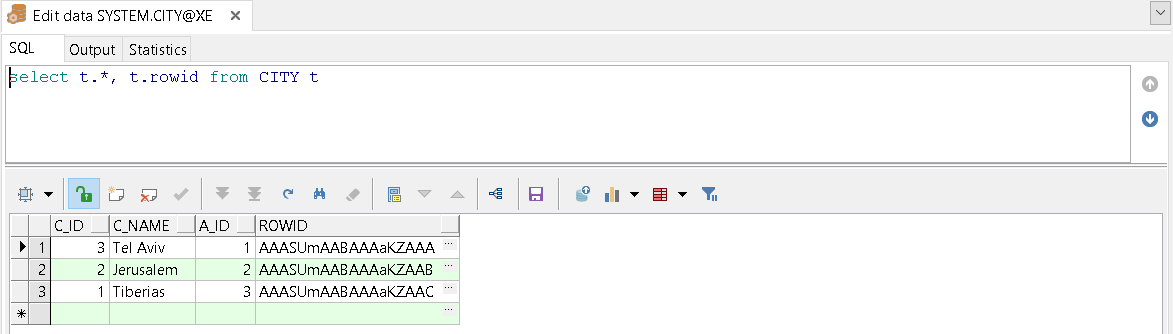
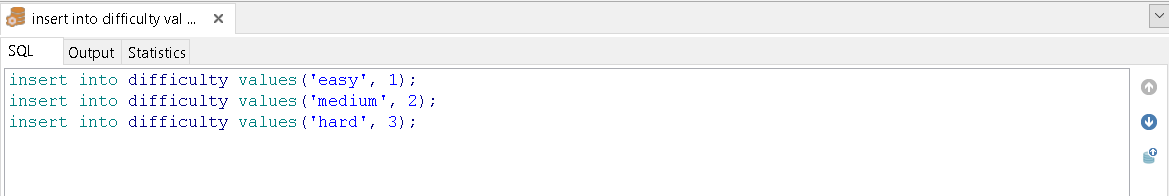
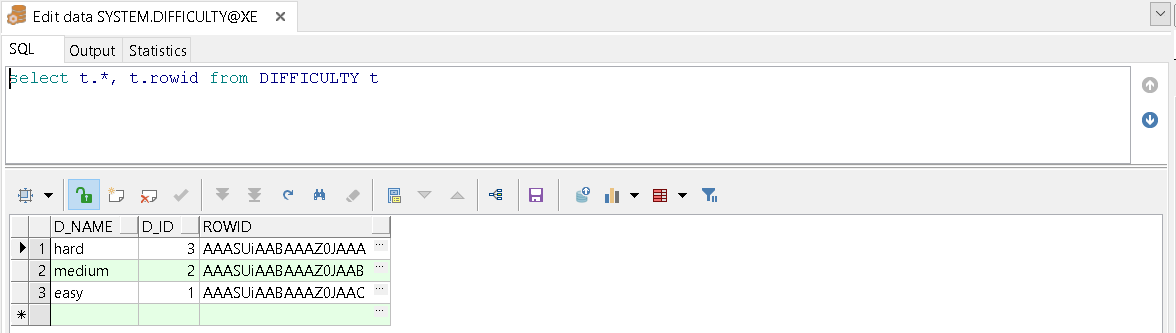
אכלוס טבלאות

פקודת insert

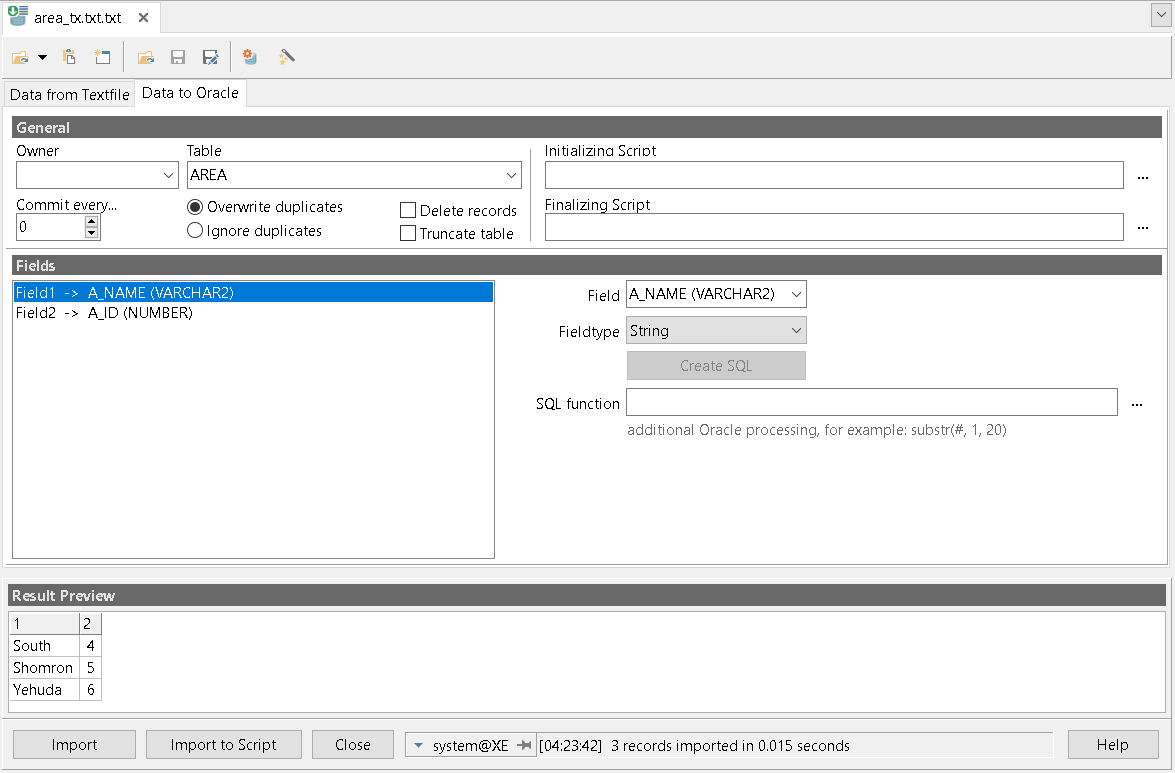
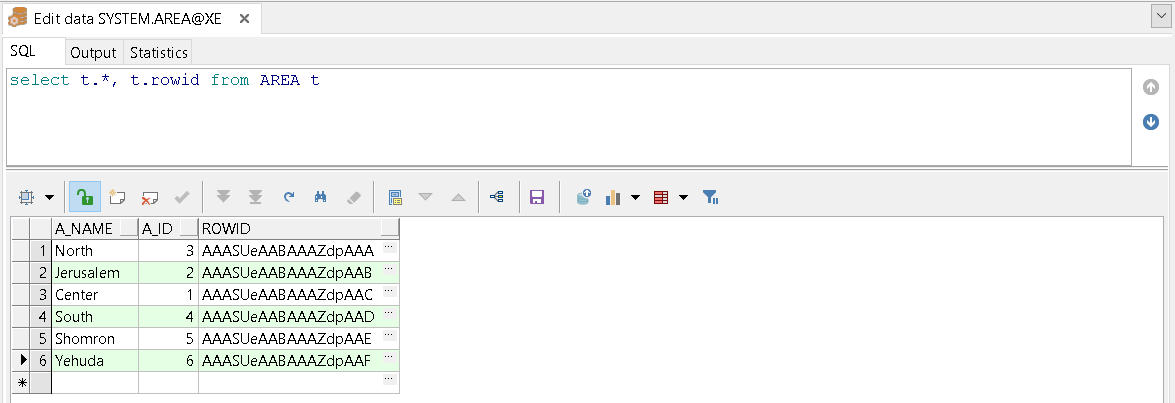


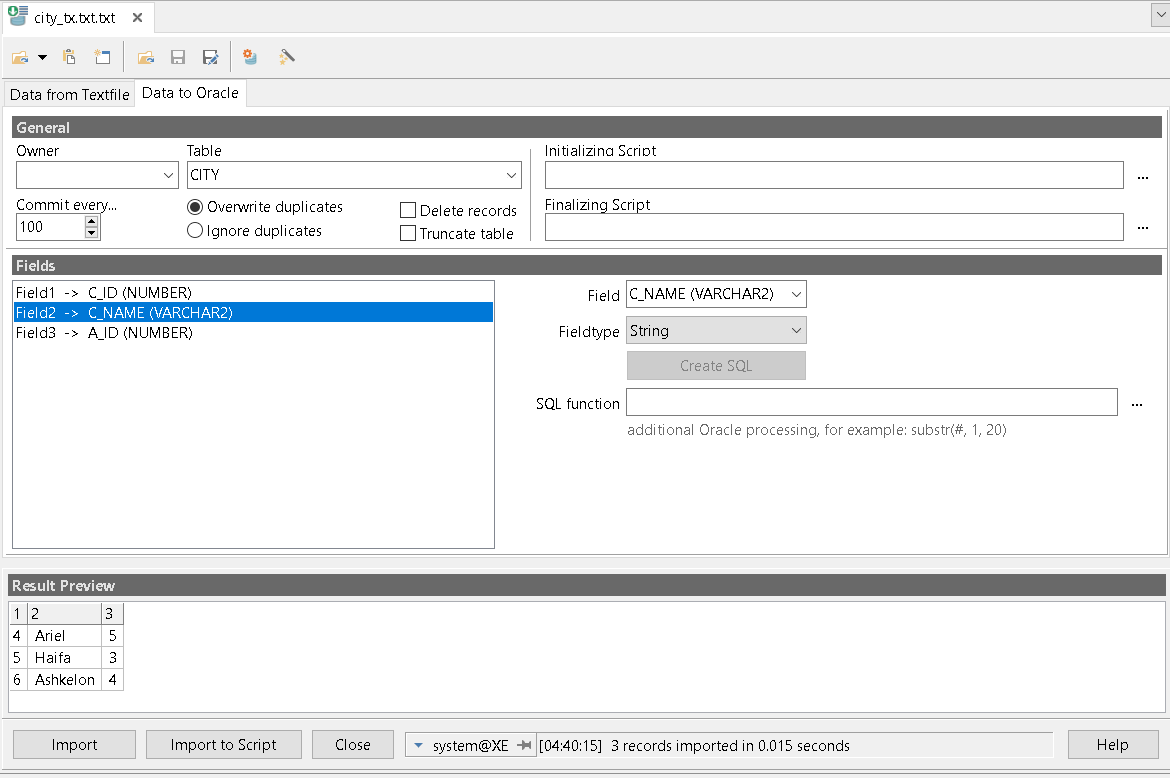


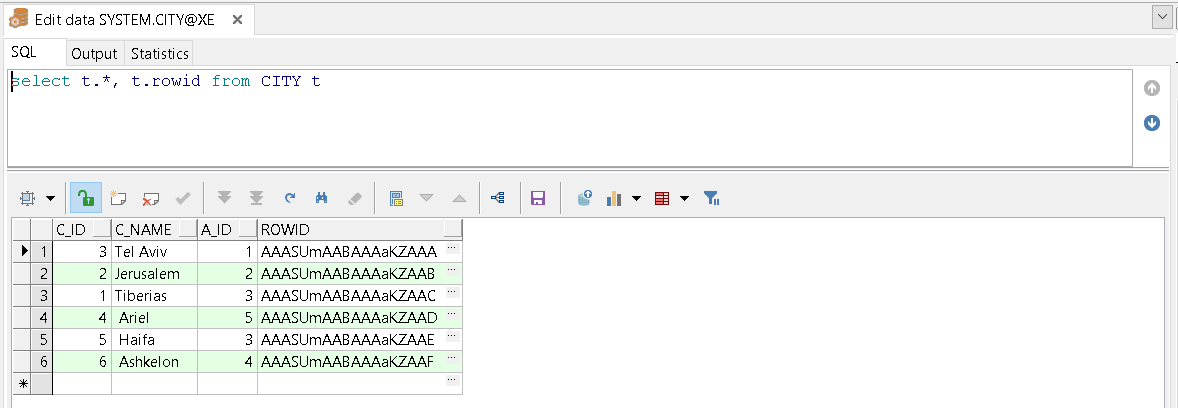


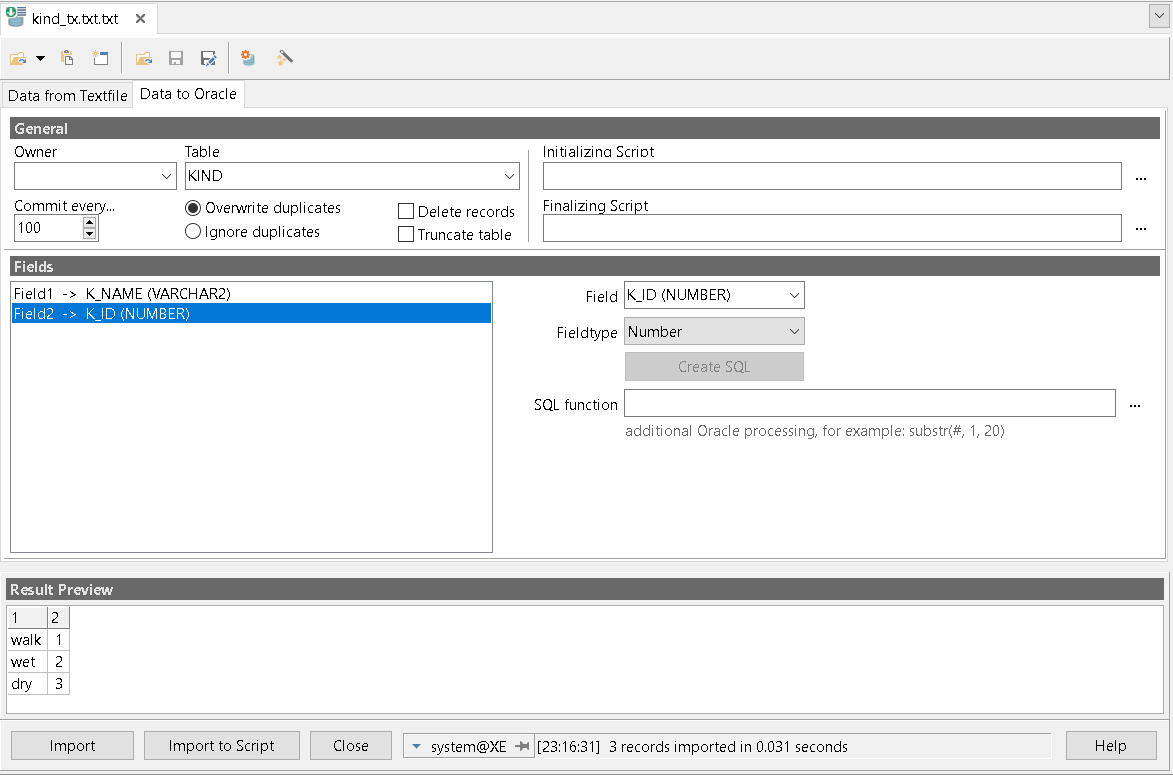


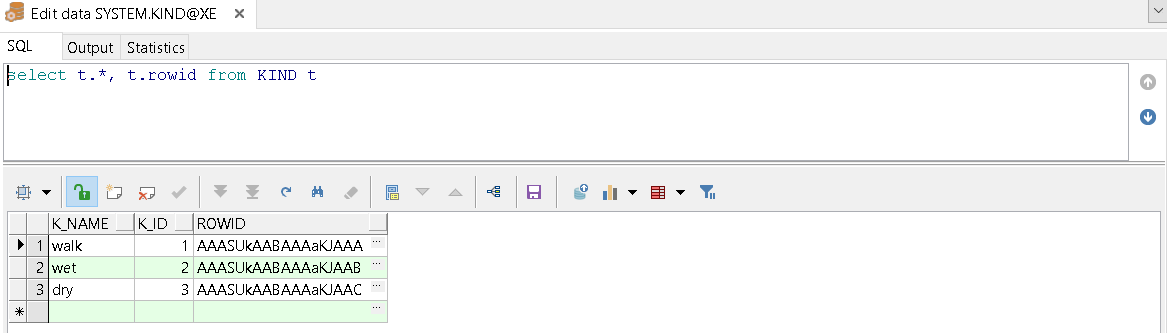
קליטה מקובץ TXT



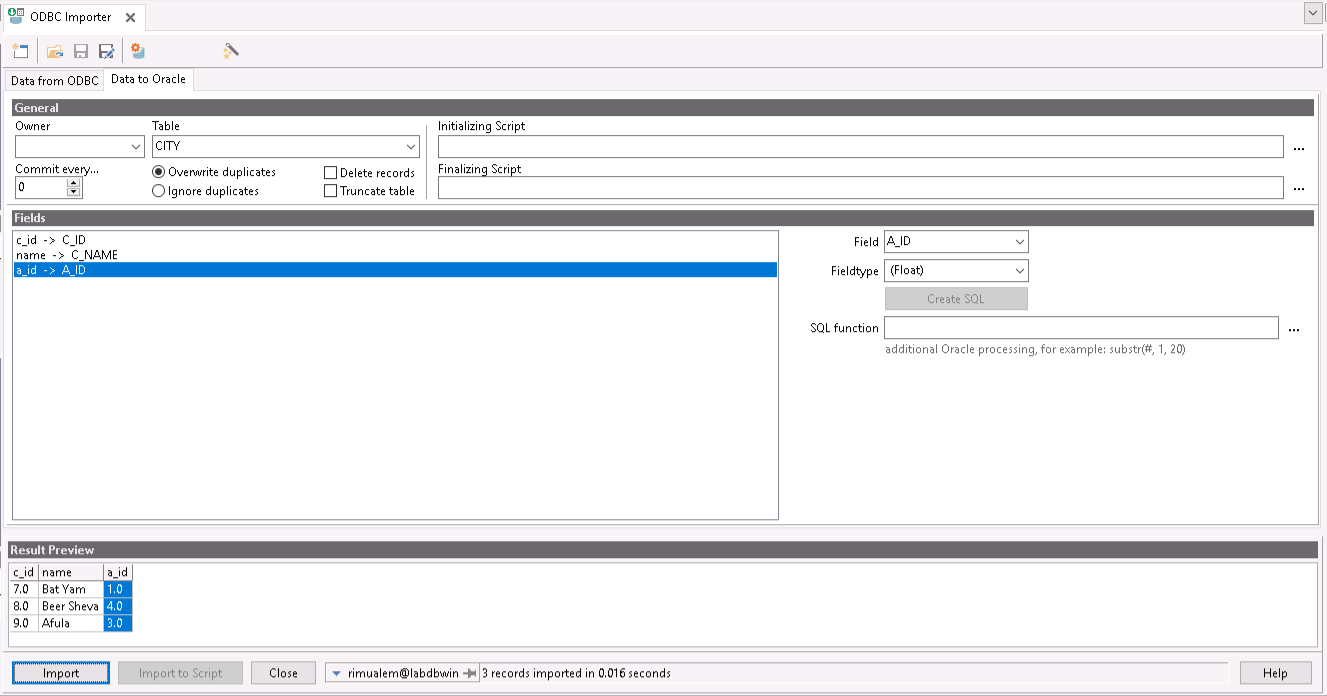


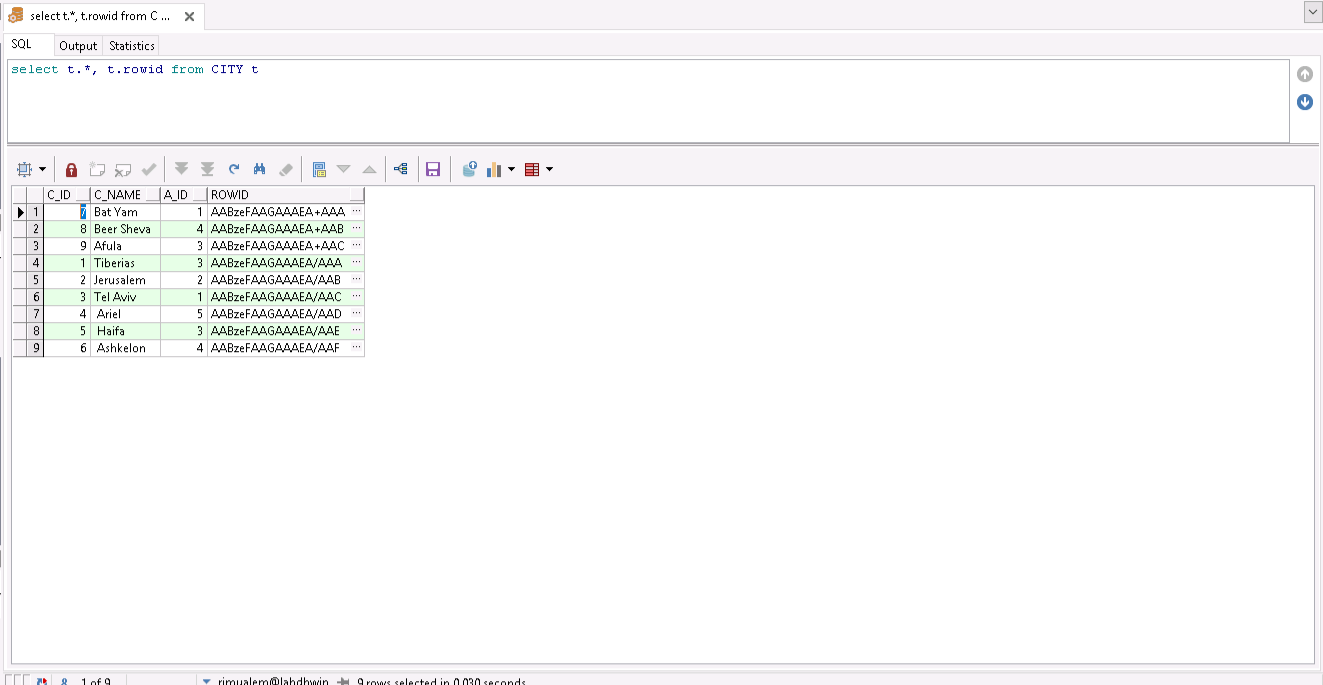


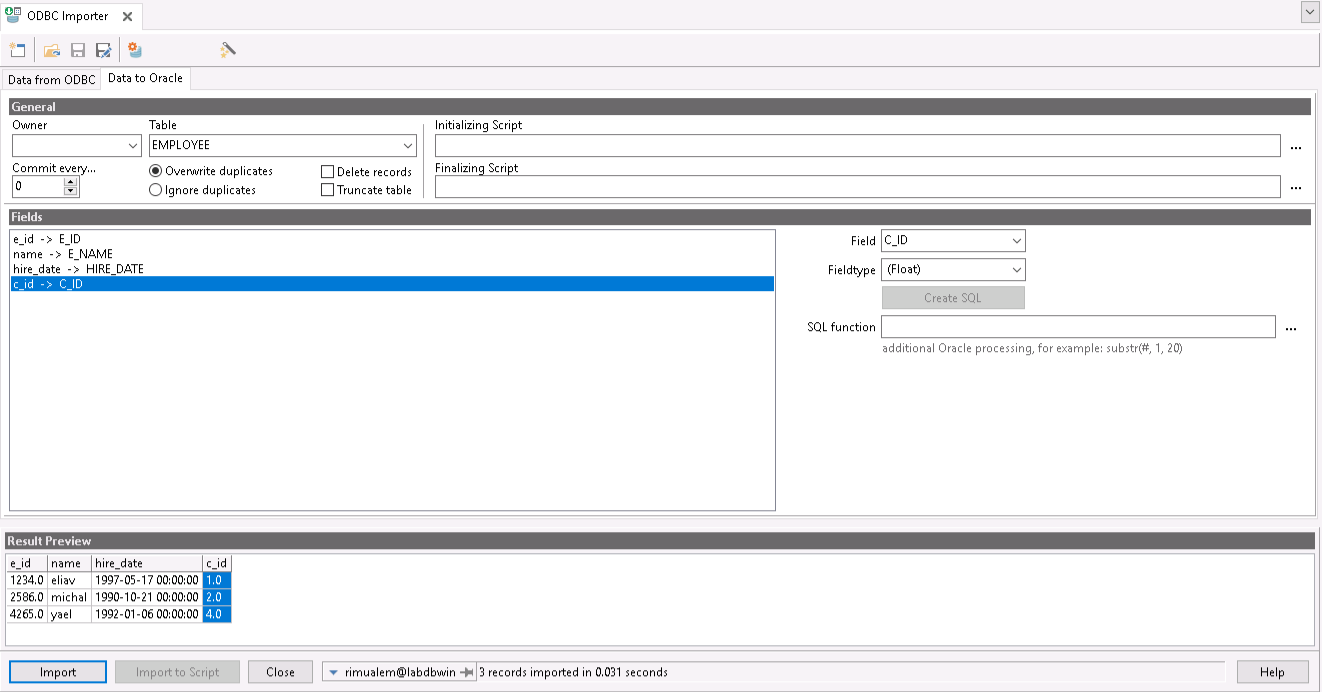


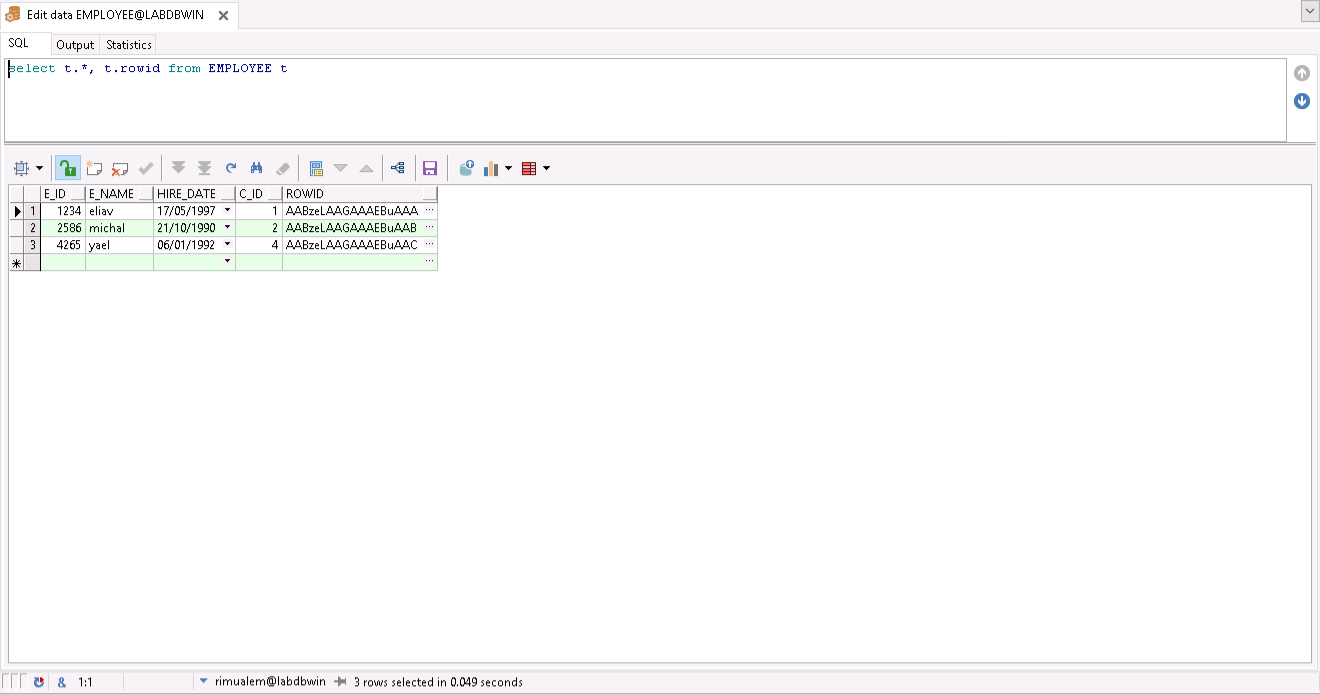


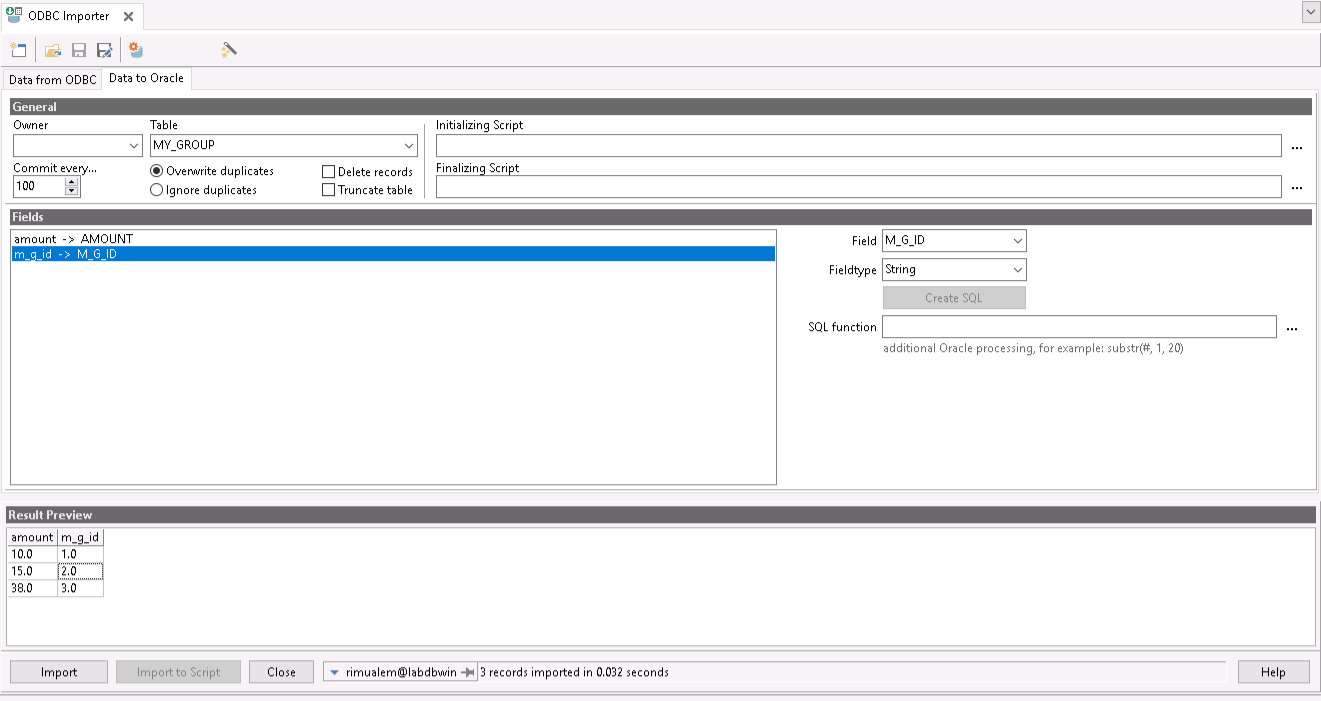
ODBC Importer

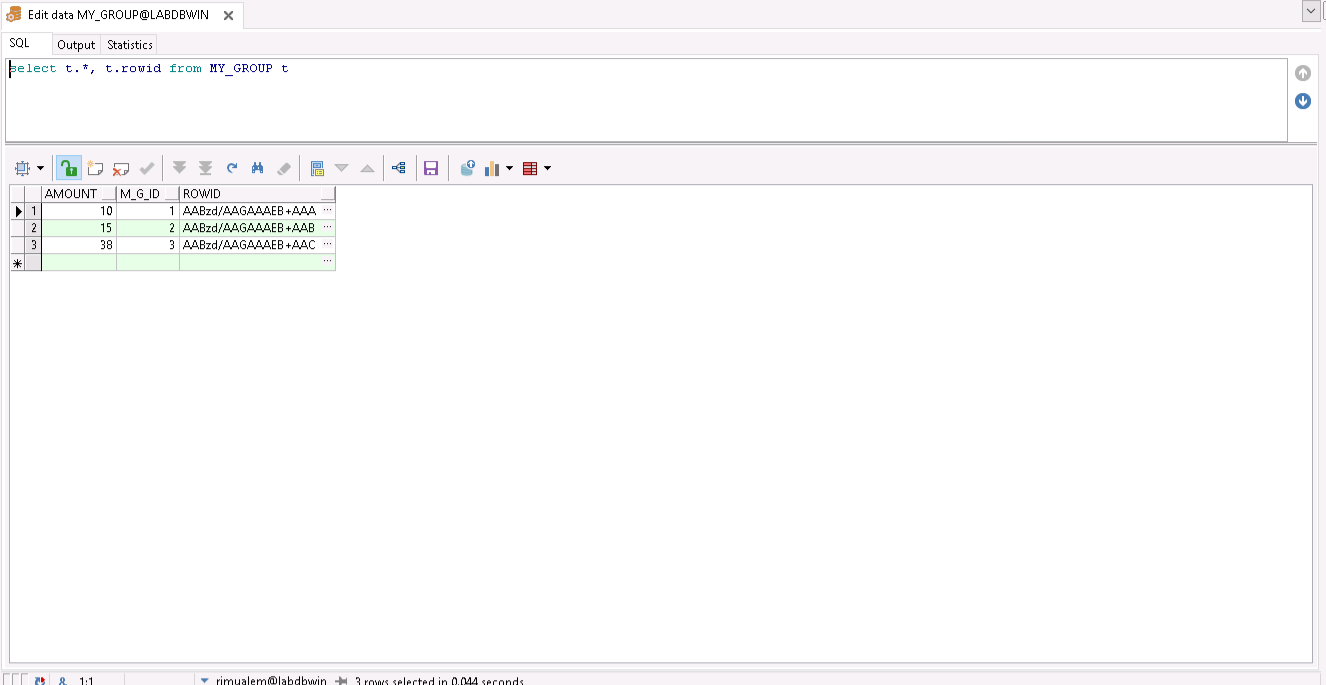


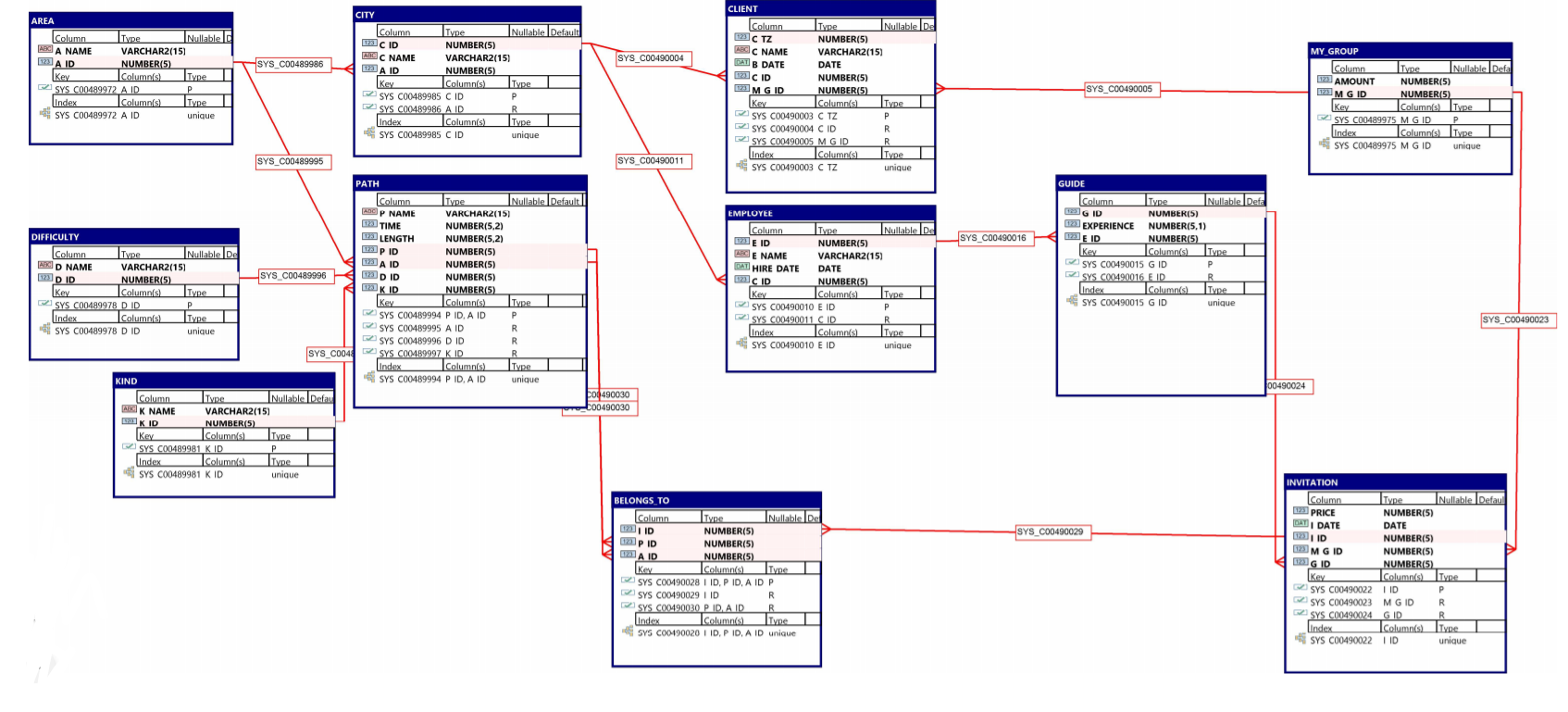












פקודת datagenerator

