

מיני פרויקט בבסיסי נתונים

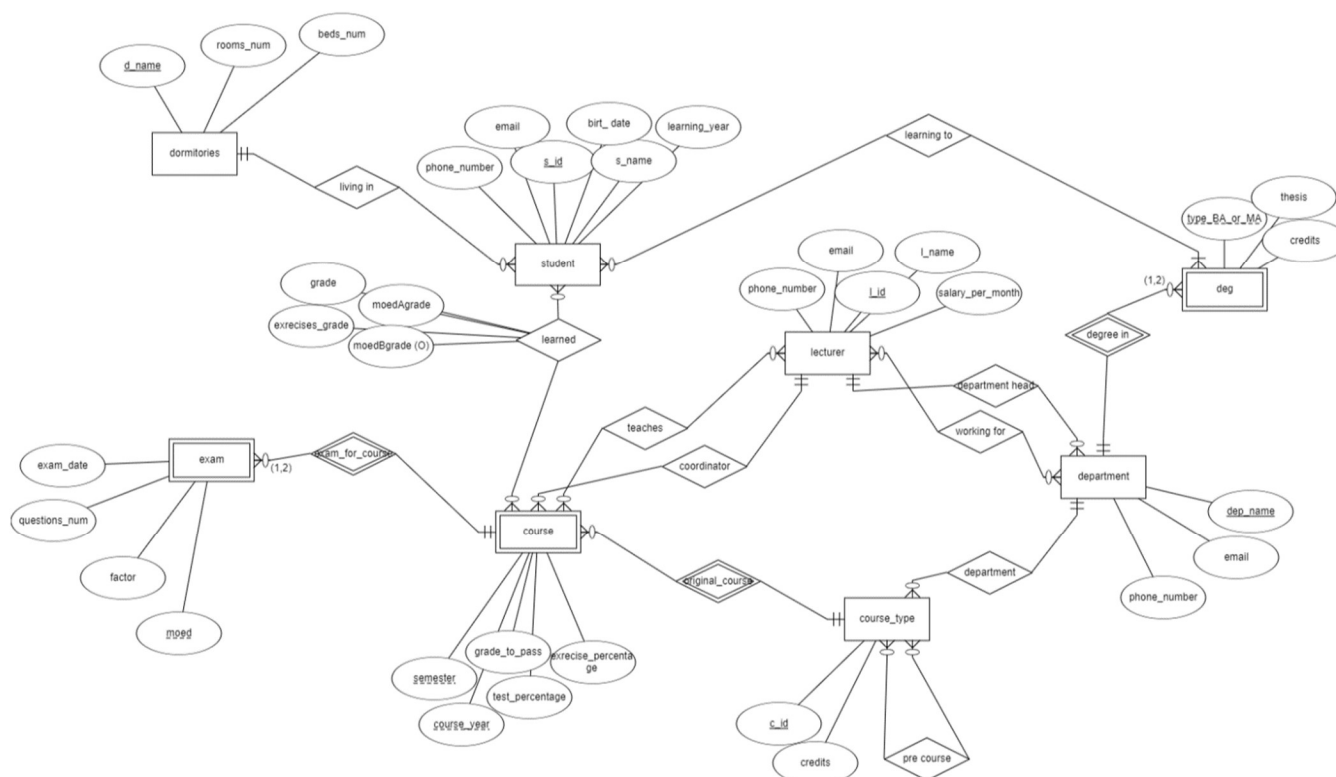
תוכן עניינים

2	תיאור המערכת, ותרשימים ראשוניים:
4	יצירת הטבלאות:
6	פקודות ה-desc:
9	יצירת הנתונים:
11	גיבוי ושחזור הנתונים:

תיאור המערכת, ותרשימים ראשוניים:

המערכת היא מערכת של אוניברסיטה, ואנחנו בחרנו לעשות בנושא סטודנטים.

תיאור מערכת הסטודנטים: המערכת כוללת את כל המידע הרלוונטי עבור סטודנט הלומד באוניברסיטה. הישגיות הכלולות במערכת הן ישות של סטודנט, הכוללת את שמו, מספר הטלפון שלו, מייל, תאריך לידה, מספר תעודת זהות ושנת לימוד. ישות של מעונות, הכוללת את שם הבניין, כמות החדרים וכמות המיטות. ישות של מרצה, הכוללת את פרטיו האישיים ואת השכר החודשי שלו. ישות של תואר, התלויה במחלקה, וכוללת האם הוא תואר ראשון או שני, האם עושים בו תזה, וכמה נקודות זכות נדרשות על מנת להשלים אותו. של מחלקה באוניברסיטה, הכוללת את פרטיה. של קורס אב, כלומר איזשהו קורס כללי ושל קורס ספציפי, כלומר מקרה ספציפי של קורס האב שמועבר, הכולל גם את הסמסטר, שנת הלימוד, הציון הנדרש כדי לעבור, ואחוז התרגילים והמבחן מהקורס. וישות של מבחן בקורס, הכוללת את התאריך שלו, איזה מועד הוא, מספר השאלות, והפקטור שניתן בו. המערכת כוללת בנוסף קשרים בין הישגיות, כפי שאפשר לראות בתרשים ה-ERD המצורף:

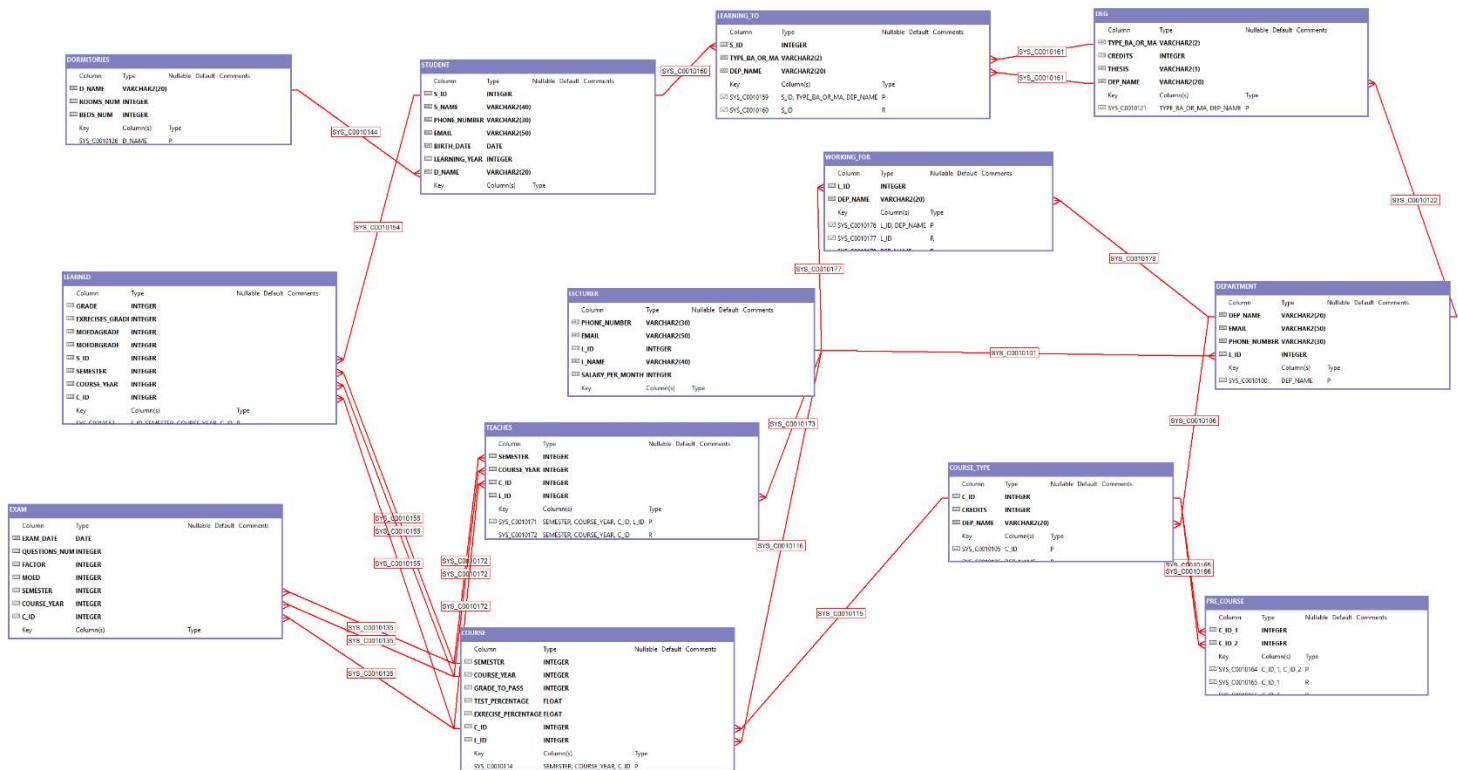


החלטנו להפריד את הישות של קורס אב מהישות של קורס, מכיוון שיכול להיות אותו הקורס שיועבר כמה פעמים, והמידע צריך להישמר בנפרד. בנוסף, מה שהסטודנט תלוי בו, כלומר הקורסים, המבחנים, והתואר, תלויים מאוד בישות של מרצה, לכן החלטנו לממש גם אותה.

בנוסף, גם בין קורס למרצה וגם בין מחלקה למרצה עשינו קשר כפול. בין מחלקה למרצה, אחד הקשרים מתאר את המרצה היחיד שהוא ראש המחלקה, והקשר השני מתאר שהמרצה עובד במחלקה הזו. בין מרצה לקורס, עשינו קשר אחד המתאר את רכז הקורס, ועוד קשר לתאר את המרצים המעבירים את הקורס.

עוד דבר שעשינו, הוא אופציה תיאורית של סטודנט שלומד ליותר מתואר אחד בו זמנית, ולכן הקשר של learning_to הוא טבלה בפני עצמה ולא רק מידע שנשמר עבור כל סטודנט.

מצורף תרשים ה-DSD של המערכת:



יצירת הטבלאות:

לאחר שסיימנו עם הכנת קובץ ה-ERD,

יצרנו את הטבלאות בעזרת קוד ה-SQL הבא:

```
CREATE TABLE lecturer
(
  phone_number VARCHAR2(30) NOT NULL,
  email VARCHAR2(50) NOT NULL,
  l_id INT NOT NULL,
  l_name VARCHAR(40) NOT NULL,
  salary_per_month INT NOT NULL,
  PRIMARY KEY (l_id)
);

CREATE TABLE dormitories
(
  d_name VARCHAR(20) NOT NULL,
  rooms_num INT NOT NULL,
  beds_num INT NOT NULL,
  PRIMARY KEY (d_name)
);

CREATE TABLE department
(
  dep_name VARCHAR(20) NOT NULL,
  email VARCHAR(50) NOT NULL,
  phone_number VARCHAR(30) NOT NULL,
  l_id INT NOT NULL,
  PRIMARY KEY (dep_name),
  FOREIGN KEY (l_id) REFERENCES lecturer(l_id)
);

CREATE TABLE deg
(
  type_BA_or_MA VARCHAR(2) NOT NULL,
  credits INT NOT NULL,
  thesis VARCHAR(1) NOT NULL,
  dep_name VARCHAR(20) NOT NULL,
  PRIMARY KEY (type_BA_or_MA, dep_name),
  FOREIGN KEY (dep_name) REFERENCES department(dep_name)
);

CREATE TABLE working_for
(
  l_id INT NOT NULL,
  dep_name VARCHAR(20) NOT NULL,
  PRIMARY KEY (l_id, dep_name),
  FOREIGN KEY (l_id) REFERENCES lecturer(l_id),
  FOREIGN KEY (dep_name) REFERENCES department(dep_name)
);

CREATE TABLE student
(
  s_id INT NOT NULL,
  s_name VARCHAR(40) NOT NULL,
  phone_number VARCHAR(30) NOT NULL,
  email VARCHAR(50) NOT NULL,
  birth_date DATE NOT NULL,
  learning_year INT NOT NULL,
  d_name VARCHAR(20) NOT NULL,
  PRIMARY KEY (s_id),
  FOREIGN KEY (d_name) REFERENCES dormitories(d_name)
);

CREATE TABLE course_type
(
  c_id INT NOT NULL,
  credits INT NOT NULL,
  dep_name VARCHAR(20) NOT NULL,
  PRIMARY KEY (c_id),
  FOREIGN KEY (dep_name) REFERENCES department(dep_name)
);
```

```

CREATE TABLE learning_to
(
    s_id INT NOT NULL,
    type_BA_or_MA VARCHAR(2) NOT NULL,
    dep_name VARCHAR(20) NOT NULL,
    PRIMARY KEY (s_id, type_BA_or_MA, dep_name),
    FOREIGN KEY (s_id) REFERENCES student(s_id),
    FOREIGN KEY (type_BA_or_MA, dep_name) REFERENCES deg(type_BA_or_MA, dep_name)
);

CREATE TABLE pre_course
(
    c_id_1 INT NOT NULL,
    c_id_2 INT NOT NULL,
    PRIMARY KEY (c_id_1, c_id_2),
    FOREIGN KEY (c_id_1) REFERENCES course_type(c_id),
    FOREIGN KEY (c_id_2) REFERENCES course_type(c_id)
);

CREATE TABLE course
(
    semester INT NOT NULL,
    course_year INT NOT NULL,
    grade_to_pass INT NOT NULL,
    test_percentage FLOAT NOT NULL,
    exrecise_percentage FLOAT NOT NULL,
    c_id INT NOT NULL,
    l_id INT NOT NULL,
    PRIMARY KEY (semester, course_year, c_id),
    FOREIGN KEY (c_id) REFERENCES course_type(c_id),
    FOREIGN KEY (l_id) REFERENCES lecturer(l_id)
);

CREATE TABLE exam
(
    exam_date DATE NOT NULL,
    questions_num INT NOT NULL,
    factor INT NOT NULL,
    moed INT NOT NULL,
    semester INT NOT NULL,
    course_year INT NOT NULL,
    c_id INT NOT NULL,
    PRIMARY KEY (moed, semester, course_year, c_id),
    FOREIGN KEY (semester, course_year, c_id) REFERENCES course(semester, course_year,
);

CREATE TABLE teaches
(
    semester INT NOT NULL,
    course_year INT NOT NULL,
    c_id INT NOT NULL,
    l_id INT NOT NULL,
    PRIMARY KEY (semester, course_year, c_id, l_id),
    FOREIGN KEY (semester, course_year, c_id) REFERENCES course(semester, course_year,
    FOREIGN KEY (l_id) REFERENCES lecturer(l_id)
);

CREATE TABLE learned
(
    grade INT NOT NULL,
    exrecises_grade INT NOT NULL,
    moedAgrade INT NOT NULL,
    moedBgrade INT NOT NULL,
    s_id INT NOT NULL,
    semester INT NOT NULL,
    course_year INT NOT NULL,
    c_id INT NOT NULL,
    PRIMARY KEY (s_id, semester, course_year, c_id),
    FOREIGN KEY (s_id) REFERENCES student(s_id),
    FOREIGN KEY (semester, course_year, c_id) REFERENCES course(semester, course_year,
);

```

פקודות ה-desc:

נציג את פקודות הdesc כפי שהרצנו על כל הטבלאות.

SQL> desc lecturer;

:Lecturer

Name	Type	Nullable	Default	Comments
PHONE_NUMBER	VARCHAR2 (30)			
EMAIL	VARCHAR2 (50)			
L_ID	INTEGER			
L_NAME	VARCHAR2 (40)			
SALARY_PER_MONTH	INTEGER			

SQL> desc dormitories;

:Dormitories

Name	Type	Nullable	Default	Comments
D_NAME	VARCHAR2 (20)			
ROOMS_NUM	INTEGER			
BEDS_NUM	INTEGER			

SQL> desc department;

:Department

Name	Type	Nullable	Default	Comments
DEP_NAME	VARCHAR2 (20)			
EMAIL	VARCHAR2 (50)			
PHONE_NUMBER	VARCHAR2 (30)			
L_ID	INTEGER			

SQL> desc deg;

:Deg

Name	Type	Nullable	Default	Comments
TYPE_BA_OR_MA	VARCHAR2 (2)			
CREDITS	INTEGER			
THESIS	VARCHAR2 (1)			
DEP_NAME	VARCHAR2 (20)			

SQL> desc working_for;

:Working_for

Name	Type	Nullable	Default	Comments
L_ID	INTEGER			
DEP_NAME	VARCHAR2 (20)			

```
SQL> desc student;
```

Name	Type	Nullable	Default	Comments
S_ID	INTEGER			
S_NAME	VARCHAR2(40)			
PHONE_NUMBER	VARCHAR2(30)			
EMAIL	VARCHAR2(50)			
BIRTH_DATE	DATE			
LEARNING_YEAR	INTEGER			
D_NAME	VARCHAR2(20)			

:Student

:Course_type

```
SQL> desc course_type;
```

Name	Type	Nullable	Default	Comments
C_ID	INTEGER			
CREDITS	INTEGER			
DEP_NAME	VARCHAR2(20)			

:Learning_to

```
SQL> desc learning_to;
```

Name	Type	Nullable	Default	Comments
S_ID	INTEGER			
TYPE_BA_OR_MA	VARCHAR2(2)			
DEP_NAME	VARCHAR2(20)			

:Pre_course

```
SQL> desc pre_course;
```

Name	Type	Nullable	Default	Comments
C_ID_1	INTEGER			
C_ID_2	INTEGER			

:Course

```
SQL> desc course;
```

Name	Type	Nullable	Default	Comments
SEMESTER	INTEGER			
COURSE_YEAR	INTEGER			
GRADE_TO_PASS	INTEGER			
TEST_PERCENTAGE	FLOAT			
EXERCISE_PERCENTAGE	FLOAT			
C_ID	INTEGER			
L_ID	INTEGER			

:Exam

```
SQL> desc exam;
```

Name	Type	Nullable	Default	Comments
EXAM_DATE	DATE			
QUESTIONS_NUM	INTEGER			
FACTOR	INTEGER			
MOED	INTEGER			
SEMESTER	INTEGER			
COURSE_YEAR	INTEGER			
C_ID	INTEGER			

:Teaches

```
SQL> desc teaches;
```

Name	Type	Nullable	Default	Comments
SEMESTER	INTEGER			
COURSE_YEAR	INTEGER			
C_ID	INTEGER			
L_ID	INTEGER			

:Learned

```
SQL> desc learned;
```

Name	Type	Nullable	Default	Comments
GRADE	INTEGER			
EXRECISES_GRADE	INTEGER			
MOEDAGRADE	INTEGER			
MOEDBGRADE	INTEGER			
S_ID	INTEGER			
SEMESTER	INTEGER			
COURSE_YEAR	INTEGER			
C_ID	INTEGER			

יצירת הנתונים:

החלטנו להשתמש בשלוש השיטות הבאות: mockaroo, python-faker, ו-data generator של sql pl. צילום מסך מתוך יצירת הנתונים באמצעות mockaroo (הצילום מסך הוא של יצירת הנתונים עבור הישות (course).

Looking to generate **fake data** based on your **production data**? Mimic your databases with a trial account from **TONIC**

Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.
Need more data? Plans start at just \$60/year. Mockaroo is also available as a **docker image** that you can deploy in your own private cloud.

Field Name	Type	Options									
c_id	Number	min:	10000	max:	99999	decimals:	0	blank:	0 %	Σ	×
semester	Number	min:	1	max:	2	decimals:	0	blank:	0 %	Σ	×
course_year	Number	min:	2000	max:	2020	decimals:	0	blank:	0 %	Σ	×
grade_to_pass	Number	min:	55	max:	65	decimals:	0	blank:	0 %	Σ	×
test_percentage	Number	min:	10	max:	20	decimals:	0	blank:	0 %	Σ	×
exercise_percentage	Number	min:	1	max:	1	decimals:	0	blank:	0 %	Σ	×
l_id	Number	min:	1000000	max:	9999999	decimals:	0	blank:	0 %	Σ	×

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

Rows: 400 Format: CSV Line Ending: Unix (LF) Include: ☒ header ☐ BOM

הגלישה באתרוז במסלול מתחם נשים GENERATE DATA PREVIEW SAVE AS... DERIVE FROM EXAMPLE... MORE הגלישה באתרוז במסלול מתחם נשים

צילום מסך מתוך יצירת הנתונים באמצעות python-faker (בצילום אפשר לראות את יצירת הישות (dormitories):

```
>>> from faker import Faker
import datetime

fake = Faker()
fake.unique.clear()

def gen_dormitories():
    return [fake.unique.last_name(), random.randint(3,5), random.randint(10,15)]

dormitories=[]
with open('dormitories.csv', 'w') as csvfile:
    writer = csv.writer(csvfile)
    q=writer.writerow(['d_name', 'rooms_num', 'beds_num'])
    for n in range(400):
        x=gen_dormitories()
        q=writer.writerow(x)
        dormitories+=x]>>> import csv
>>> import random
>>> import datetime
>>>
>>> fake = Faker()
>>> fake.unique.clear()
>>>
>>> def gen_dormitories():
...     return [fake.unique.last_name(), random.randint(3,5), random.randint(10,15)]
...
>>> dormitories=[]
>>> with open('dormitories.csv', 'w') as csvfile:
...     writer = csv.writer(csvfile)
...     q=writer.writerow(['d_name', 'rooms_num', 'beds_num'])
...     for n in range(400):
...         x=gen_dormitories()
...         q=writer.writerow(x)
...         dormitories+=x]
>>>
```

צילומי מסך של יצירת ישויות באמצעות data generator של pl/sql:

STUDENT

< Owner Table Number of records
> SYSTEM STUDENT 400
...

Name	Type	Size	Data
S_ID	NUMBER		random(1000000,9999999)
S_NAME	VARCHAR2	40	FirstName + LastName
PHONE_NUMBER	VARCHAR2	30	[0]+[0]+[0]+[0]+[0]+[0]+[0]+[0]
EMAIL	VARCHAR2	50	Email
BIRTH_DATE	DATE		random(01/01/1960, 01/01/2010)
LEARNING_YEAR	NUMBER		random(1,5)
D_NAME	VARCHAR2	20	list(select d_name from dormitories)

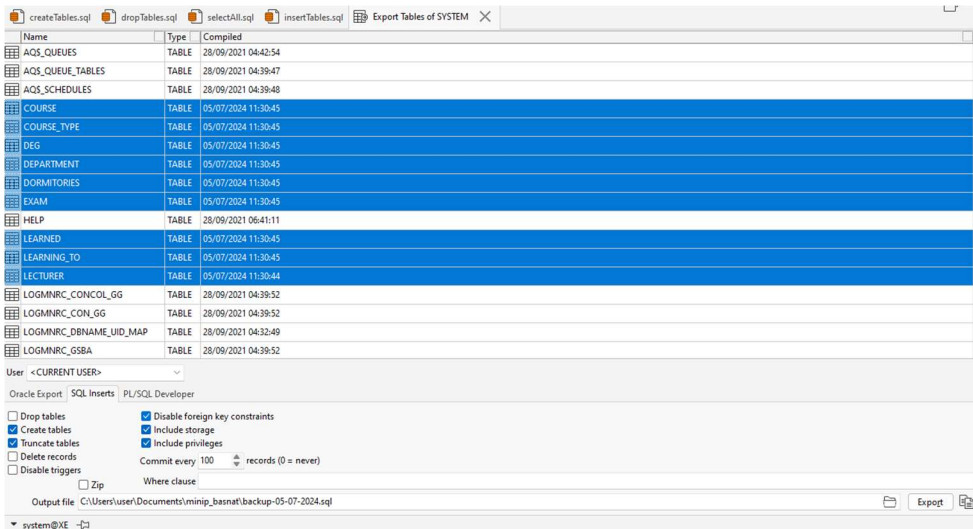
ולאחר הג'נרט:

STUDENT						
S_ID	S_NAME	PHONE_NUMBER	EMAIL	BIRTH_DATE	LEARNING_YEAR	D_NAME
1420567	IanPalminteri	17421081	ipalminteri@accurateautobody.com	20-09-1965	5	Carter
1343780	DaryleGiraldo	88395287	dgiraldo@heartlab.il	20-12-1964	2	Spence
8639514	KurtwoodGuilfoyle	70686878	kurtwood.guilfoyle@freedommedical.dk	19-01-1982	1	Richardson
5203577	ChloeMichael	87951492	chloe@tlsservicebureau.com	20-12-1978	1	Murphy
6009616	Mary BethMcLean	40803620	marybethm@sht.it	20-04-1998	3	Cook
6616328	AliPalminteri	63238257	ali.palminteri@shirtfactory.com	06-12-1973	4	Mcintosh
3641058	SaffronGarofalo	35656643	saffron@sfb.ch	18-08-1974	5	Smith
9620826	CarolNavarro	17246428	carol.navarro@marketfirst.fi	14-08-1975	2	Scott
4441491	LouiseKurtz	26970986	louise.kurtz@lms.gr	31-05-2006	4	Keith
3319758	VinceCube	58714705	vince.cube@knightsbridge.com	12-09-2007	4	Reeves
8182446	JohnetteGreenwood	36303801	johnette.greenwood@arkidata.jp	29-05-1974	1	Weiss
5922822	CarlaReiner	10217083	carlar@clorox.be	15-05-1966	3	Weber
1664355	SanderVisnjic	49545612	sander.visnjic@inspirationsoftware.com	15-02-1982	3	Ortega
2942490	ShannynTah	69845881	shannyn.tah@gtp.au	06-08-1968	4	Savage
5339601	AlannahParm	82664586	alannahp@ksj.ca	09-11-2004	1	Duncan
5546365	GoranRundgren	81976651	goran.rundgren@lms.com	26-01-2003	1	Cordova
5624630	PamLaw	91560244	pam.law@nestle.de	22-05-1977	4	Keller
2066287	AntonioPeterson	83068415	antonio@air.com	30-09-1968	4	Lewis
1708436	GeoffBloch	65279057	geoff.bloch@gillani.jp	03-10-1977	2	Allen
2890788	DerrickDepp	26676407	derrick.d@pfizer.com	12-06-1970	3	Nixon
9700785	JulieBright	57913659	julie.b@scooterstore.com	19-11-1976	1	Patrick
1035462	RonnieDiesel	37460667	ronnie.d@linacsyste.ms.com	07-02-2002	2	Palmer
4293734	NickySinise	35141521	nicky.s@anheuserbusch.com	03-06-1993	3	Best
1966092	JohnRhames	08667509	john.rhames@aoe.nl	31-07-2007	2	Wall
3104605	AvrilJanssen	71825893	ajanssen@smg.ca	11-04-1969	4	Savage
3836293	FionaWheel	26128754	fiona@yes.com	01-12-1977	4	Rodriguez
1544205	BrookeRandal	85751388	b.randal@callhenry.com	14-09-1969	2	Rowe
5050711	CollectiveByrd	18982275	collective@oriservices.com	30-08-1962	2	Ryan
5157859	BridgetBean	49335938	bridget@greenmountain.be	27-07-1963	1	Alvarez
6771460	NellyCruz	24575022	nelly.c@dataprise.dk	06-11-1983	4	Trevino

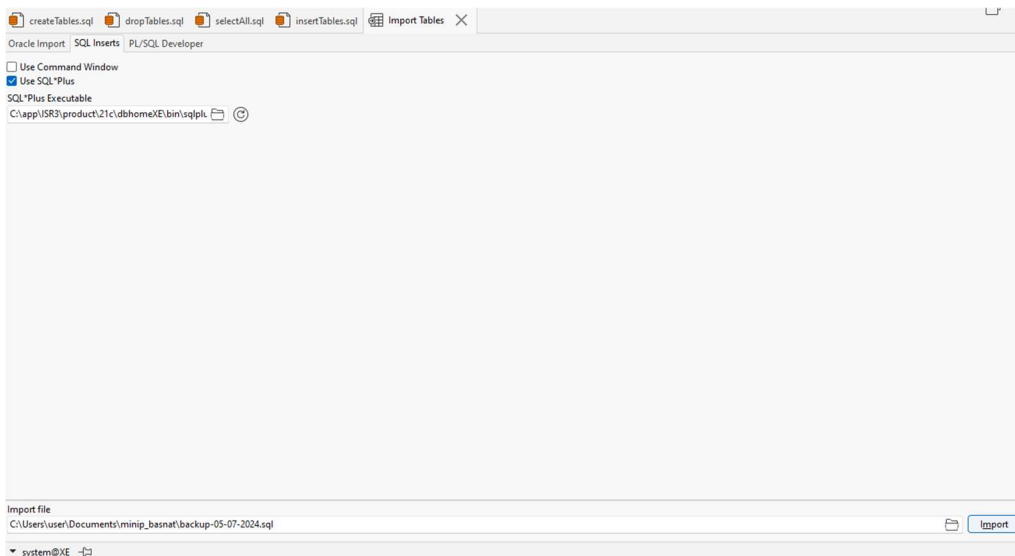
החלטנו בסופו של דבר לאכלס את המערכת שלנו דווקא בנתונים שנוצרו באמצעות פייתון, כיוון ששם יכולנו לשלוט בצורה הכי טובה על הקשרים בין מידע למידע, ולוודא שאין התנגשויות, וכו'.

גיבוי ושחזור הנתונים:

לאחר שסיימנו להכניס את הנתונים שיצרנו לתוך המערכת, גיבוינו את המערכת ע"י זה שעשינו export לכל tables, לקובץ גיבוי שנשמר ע"פ היום שבו עשינו את התיעוד:



לאחר מכן וידאנו שהנתונים נשמרו כמו שצריך, ע"י כך שעשינו drop לכל טבלאות הנתונים, ושחזרנו אותם באמצעות קובץ הגיבוי:



ראינו שהנתונים והטבלאות עלו חזרה למערכת, ווידאנו ע"י פקודות select ו-desc, וכן הרצנו גם במחשב אחר ווידאנו שגם בו מתקבל מסד הנתונים שלנו, כנדרש.