

שם הארגון: בית ספר

שם האגף: ילדים

מגישות:

ספיר פיק 326378155

sapirf1973@gmail.com

הדסה שטולמן 325462430

dasstlman@gmail.com

תוכן העניינים

3	מבוא
3	היישואיות במערכת
4	תרשים ERD של המערכת
4	תרשים DSD של המערכת
5	בניית בסיס הנתונים
5	יצירת הטבלאות
7	תיאור הטבלאות
8	הכנסת נתונים
9	Data Generator
11	Data Importer
11	Programming
13	גיבוי הנתונים ושיחזור

מבוא

בפרוייקט זה נממש מערכת ניהול מידע עבור בית ספר. המערכת מכילה ישויות מידע שונות וקשרים ביניהם על מנת למפות את כל המידע בצורה מיטבית. מטרת הפרוייקט היא לעזור לנהל את בית הספר בצורה יעילה ומתועדת היטב.

היישויות במערכת

תלמידים: פרטים אישיים של ילדים הלומדים בבית ספר

מורה: פרטים אישיים של מורים המלמדים בבית ספר

כיתה: קבוצת תלמידים שלומדים יחד, לכל כיתה יש מחנכת אחת

חדר: החדרים הפיזיים במבנה בית הספר. לכל כיתה אם יש חדר לימוד קבוע

מקצוע: תחום לימודים כלשהו הנלמד בשיעורים, מקצוע יכול להיות חובה או רשות.

שיעור: רכיב מערכת שבו מורה מלמדת מקצוע מסויים בכיתה מסויימת

Entities:

Student = Student_ID, FirstName, LastName, BirthDate, FathersName, MothersName, Phone, Class_ID

Teacher = Teacher_ID, FirstName, LastName

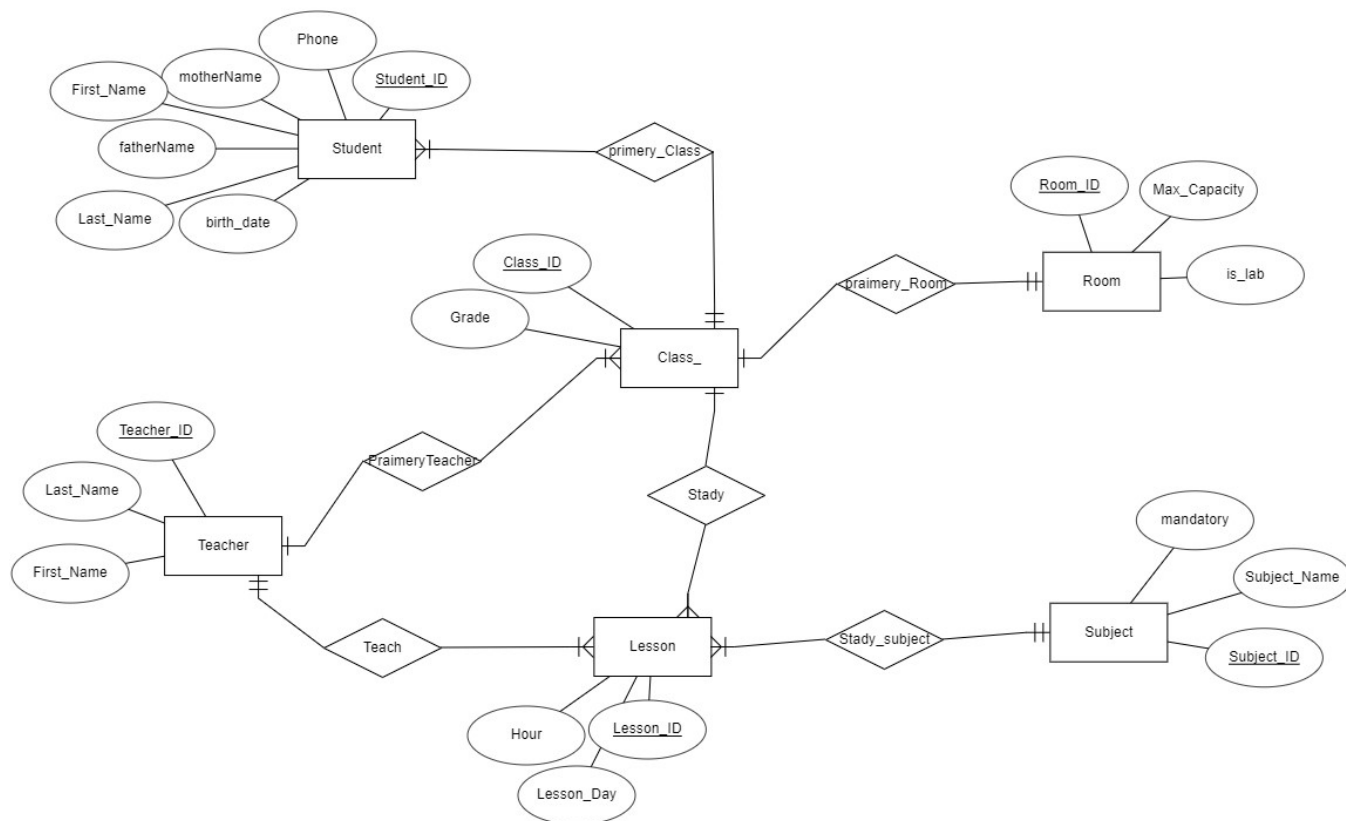
Class = Class_ID, Grade, Room_ID, Teacher_ID

Room = Room_ID, MaxCapacity, is_lab

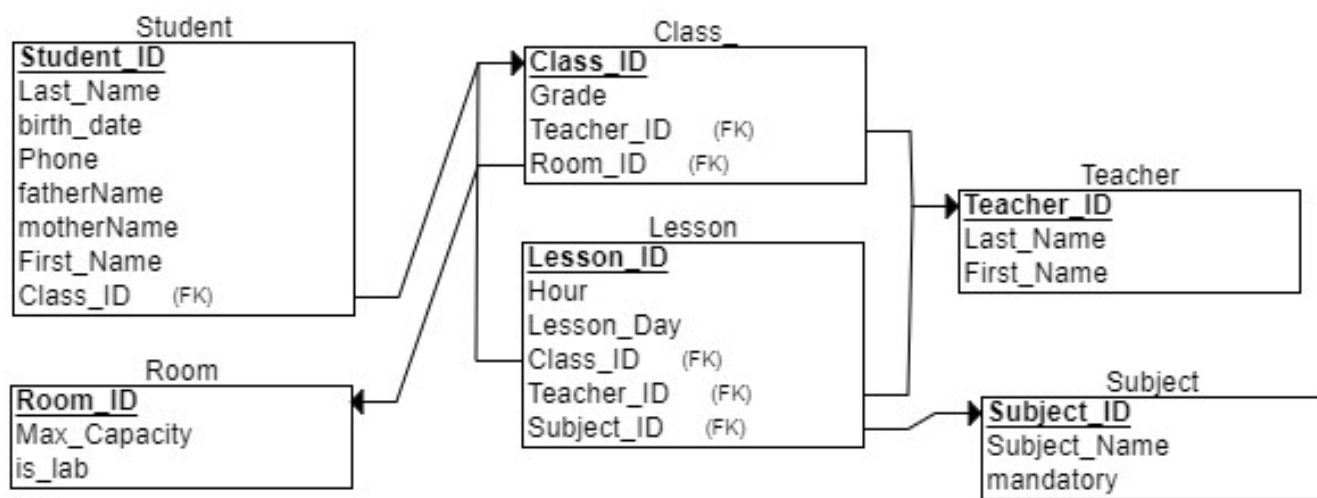
Subject = Subject_ID, SubjectName, is_mandatory

Lesson = Lesson_ID, Lesson_Day, Hour, Subject_ID, Teacher_ID

תרשים ERD של המערכת



תרשים DSD של המערכת



בניית בסיס הנתונים

יצירת הטבלאות

```

CREATE TABLE Teacher
(
  Teacher_ID NUMBER(3),
  TLast_Name VARCHAR2(15) NOT NULL,
  TFirst_Name VARCHAR2(15) NOT NULL,
  PRIMARY KEY (Teacher_ID)
);

CREATE TABLE Room
(
  Room_ID NUMBER(3),
  Max_Capacity INT NOT NULL,
  is_lab NUMBER(1),
  Last_maintenance_check DATE,
  PRIMARY KEY (Room_ID),
  CHECK (is_lab = 1 OR is_lab = 0)
);

CREATE TABLE Subject
(
  Subject_ID NUMBER(3),
  Subject_Name VARCHAR2(30) NOT NULL,
  mandatory number(1) NOT NULL,
  PRIMARY KEY (Subject_ID),
  CHECK (mandatory = 1 OR mandatory = 0)
);

CREATE TABLE Class_
(
  Class_ID NUMBER(3),
  Class_Name NUMBER(2) NOT NULL,
  Teacher_ID NUMBER(3) NOT NULL,
  Room_ID NUMBER(3) NOT NULL,
  PRIMARY KEY (Class_ID),
  FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
  FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID)
);

```

```

CREATE TABLE Lesson
(
    Lesson_ID NUMBER(3),
    Lesson_Day INT NOT NULL,
    Lesson_Hour INT NOT NULL,
    Class_ID NUMBER(3),
    Teacher_ID NUMBER(3),
    Subject_ID NUMBER(3),
    PRIMARY KEY (Lesson_ID),
    FOREIGN KEY (Class_ID) REFERENCES Class_(Class_ID),
    FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
    FOREIGN KEY (Subject_ID) REFERENCES Subject(Subject_ID),
    UNIQUE (Class_ID, Lesson_Day, Lesson_Hour),
    UNIQUE (Teacher_ID, Lesson_Day, Lesson_Hour),
    CHECK (Lesson_Day BETWEEN 1 AND 6)
);

CREATE TABLE Student
(
    Student_ID NUMBER(3),
    First_Name VARCHAR(15) NOT NULL,
    Last_Name VARCHAR2(15) NOT NULL,
    birth_date DATE NOT NULL,
    Phone INT NOT NULL,
    fatherName VARCHAR(15) NOT NULL,
    motherName VARCHAR(15) NOT NULL,
    Class_ID NUMBER(3),
    PRIMARY KEY (Student_ID),
    FOREIGN KEY (Class_ID) REFERENCES Class_(Class_ID)
);

```

תיאור הטבלאות

SQL> desc room

Name	Type	Nullable	Default	Comments
ROOM_ID	NUMBER(3)			
MAX_CAPACITY	INTEGER			
IS_LAB	NUMBER(1)	Y		
LAST_MAINTENANCE_CHECK	DATE	Y		

SQL> desc subject

Name	Type	Nullable	Default	Comments
SUBJECT_ID	NUMBER(3)			
SUBJECT_NAME	VARCHAR2(30)			
MANDATORY	NUMBER(1)			

SQL> desc teacher

Name	Type	Nullable	Default	Comments
TEACHER_ID	NUMBER(3)			
TLAST_NAME	VARCHAR2(15)			
TFIRST_NAME	VARCHAR2(15)			

SQL> desc class_

Name	Type	Nullable	Default	Comments
CLASS_ID	NUMBER(3)			
GRADE	NUMBER(2)			
TEACHER_ID	NUMBER(3)			
ROOM_ID	NUMBER(3)			

SQL> desc lesson

Name	Type	Nullable	Default	Comments
LESSON_ID	NUMBER(3)			
LESSON_DAY	INTEGER			
LESSON_HOUR	INTEGER			
CLASS_ID	NUMBER(3)	Y		
TEACHER_ID	NUMBER(3)	Y		
SUBJECT_ID	NUMBER(3)	Y		

SQL> desc student

Name	Type	Nullable	Default	Comments
STUDENT_ID	NUMBER(3)			
FIRST_NAME	VARCHAR2(15)			
LAST_NAME	VARCHAR2(15)			
BIRTH_DATE	DATE			
PHONE	INTEGER			
FATHERNAME	VARCHAR2(15)			
MOTHERNAME	VARCHAR2(15)			
CLASS_ID	NUMBER(3)	Y		

הכנסת נתונים

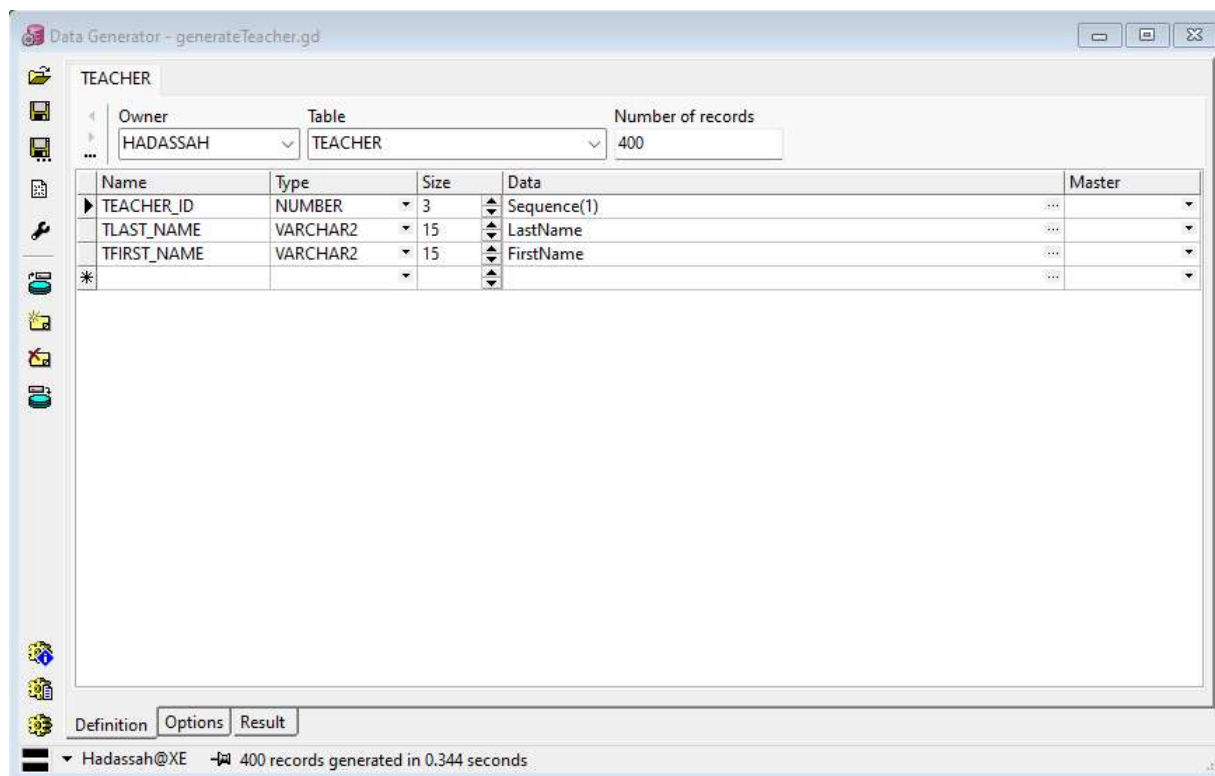
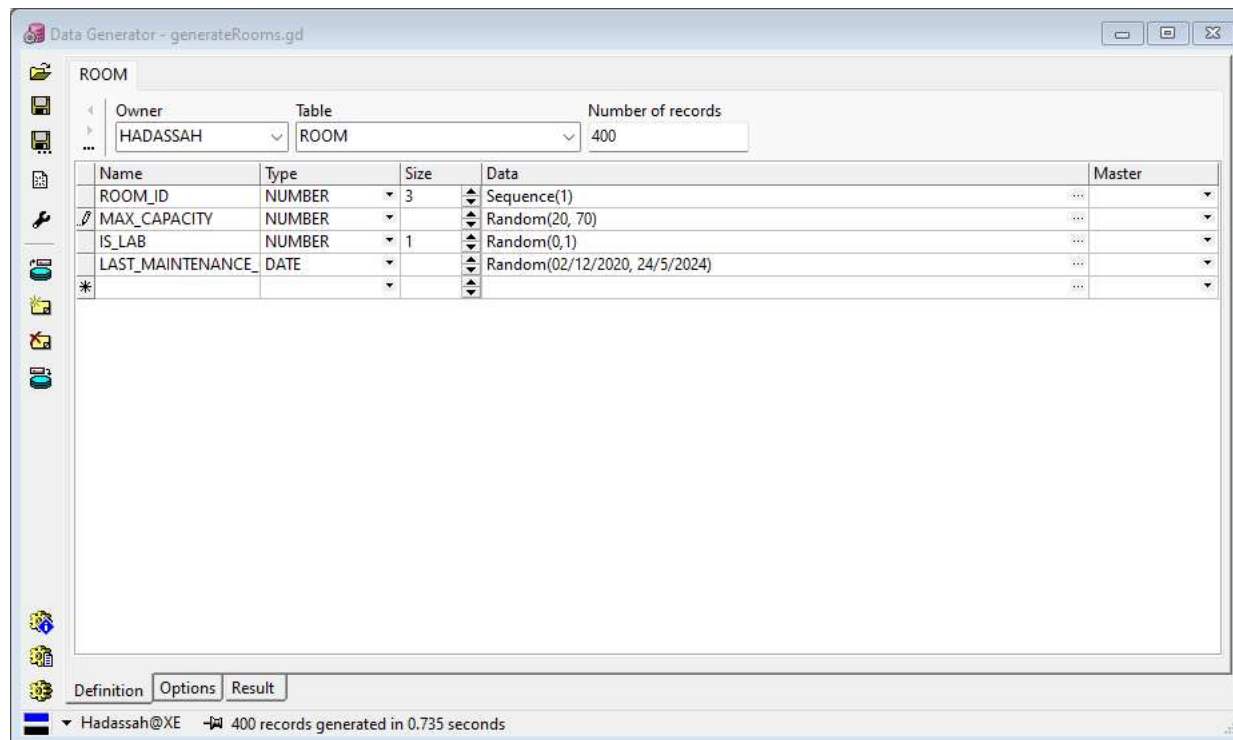
השתמשנו על פי ההנחיות ב-3 שיטות להכנסת נתונים: data generator, data importer, programming

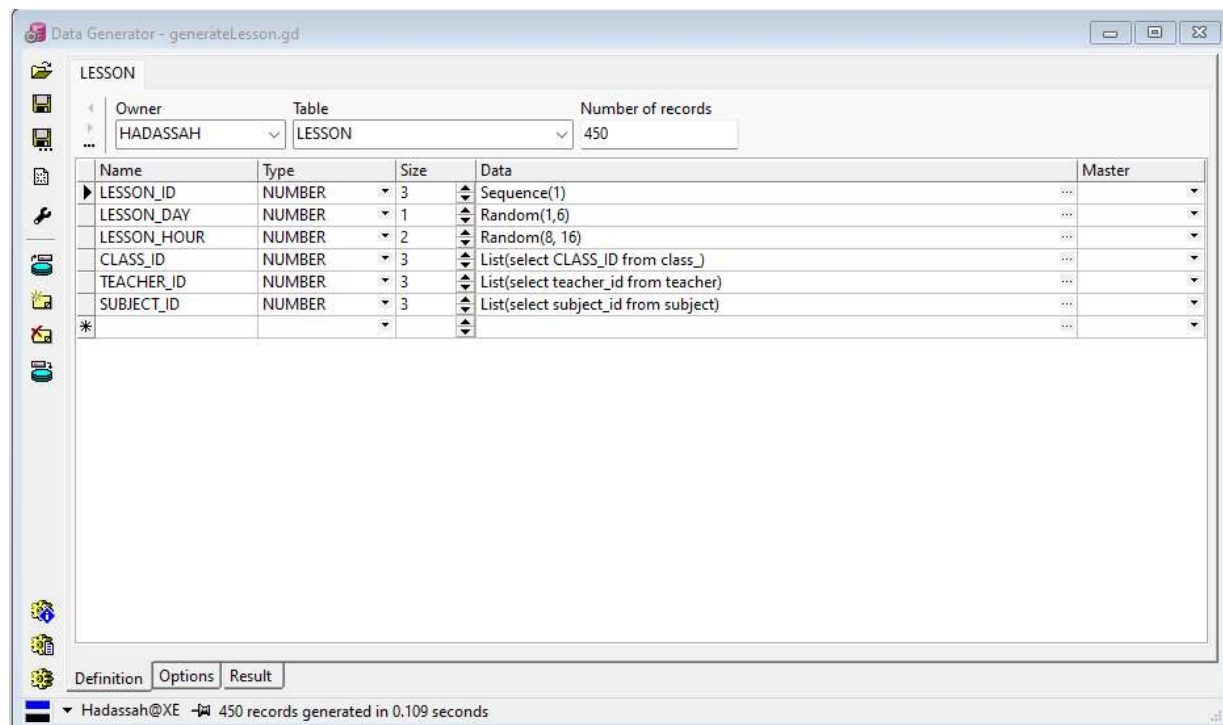
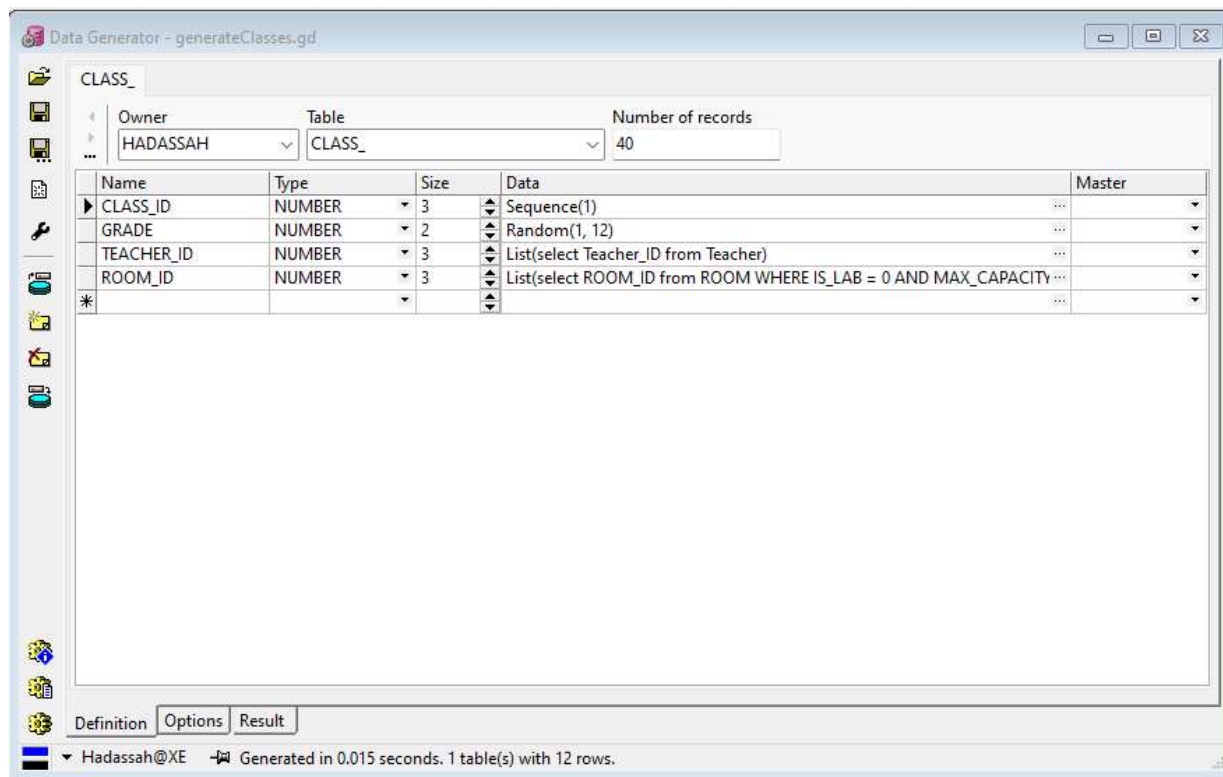
בחלק מהטבלאות קיימנו את האילוצים בשיטת "מצליח" והסתמכנו על העובדה שסטטיסטית מעט מאוד רשומות יפלו ביצירה. למשל בייצור של המידע בטבלת Lesson הסתמכנו על כך שסטטיסטית כאשר יש 6 ימים, 9 שעות לימוד ביום, 40 כיתות ו-400 מורים, הסיכוי שמורה או כיתה ישובצו לשתי שיעורים באותו יום באותה שעה, נמוך ביותר. במציאות באמת מתוך 450 רשומות שיצרנו, 400 היו חוקיות ונכנסו לטבלה.

השתדלנו לאכלס את ה-databases בנתונים ריאליסטיים ככל הניתן, ולכן:

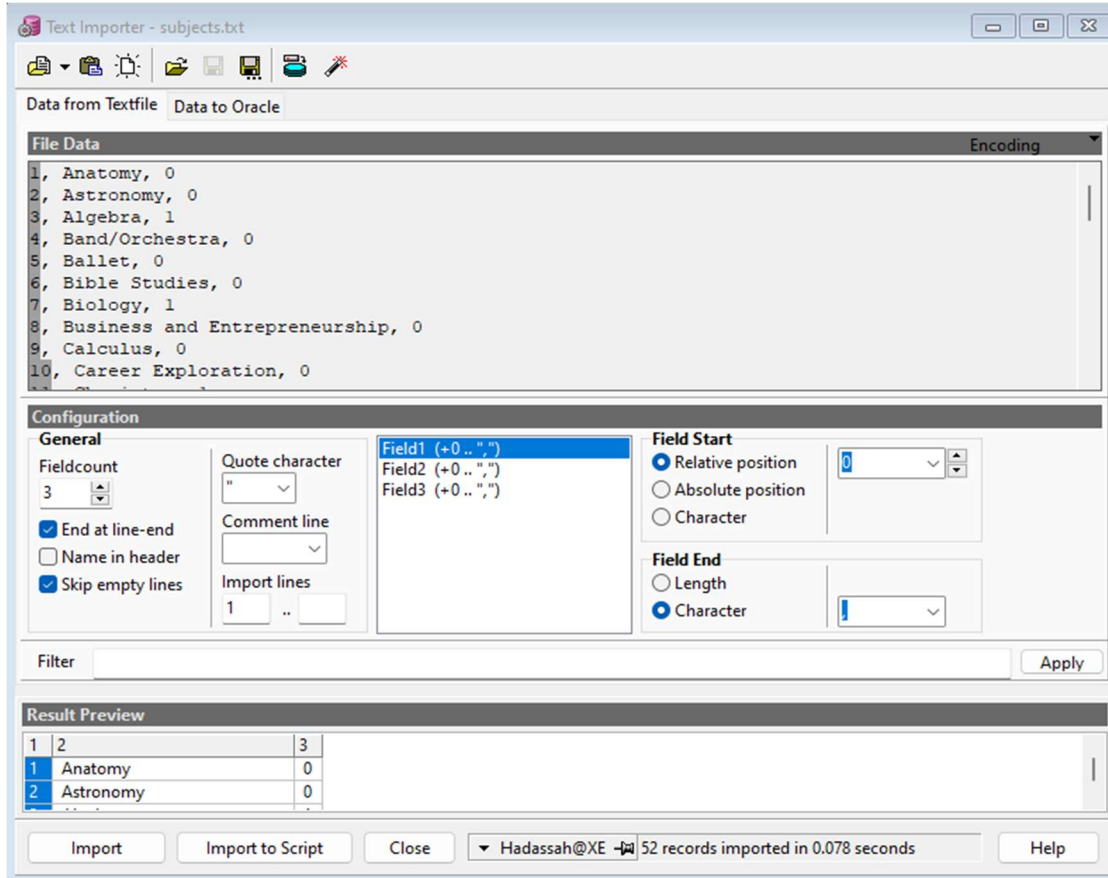
- ❖ בטבלה של Subject קיימות 52 רשומות בלבד. הכנסנו לשם כמעט כל מקצוע שניתן להעלות על הדעת שנלמד בבית ספר וגם קצת כאלו שככל הנראה לא...
- ❖ בטבלה של Class_ קיימות 40 רשומות בלבד. אם היינו רוצות למלא את הטבלה הזאת ב-400 רשומות אזי היה מתקיים תרחיש אחד מהשתיים:
 - היינו צריכות ליצור לפחות 6,000 תלמידים (ממוצע 15 בכיתה) וזה חורג מההנחיות שקיבלנו מהמרצה על מקסימום 1,000 רשומות.
 - בכל כיתה היה בממוצע תלמיד וחצי, וזה היה מקשה עלינו בהמשך הדרך להסיק מידע הגיוני ומועיל בשלבים של השאילתות
- לכן לאחר שיקול מעמיק והסתמכות על הנחיות המרצה שלנו במצגת, החלטנו שעדיף ליצור מספר גדול של תלמידים ומספר קטן יחסית של כיתות.
- ❖ בשאר הטבלאות הכנסנו מעל 400 רשומות כנדרש

Data Generator

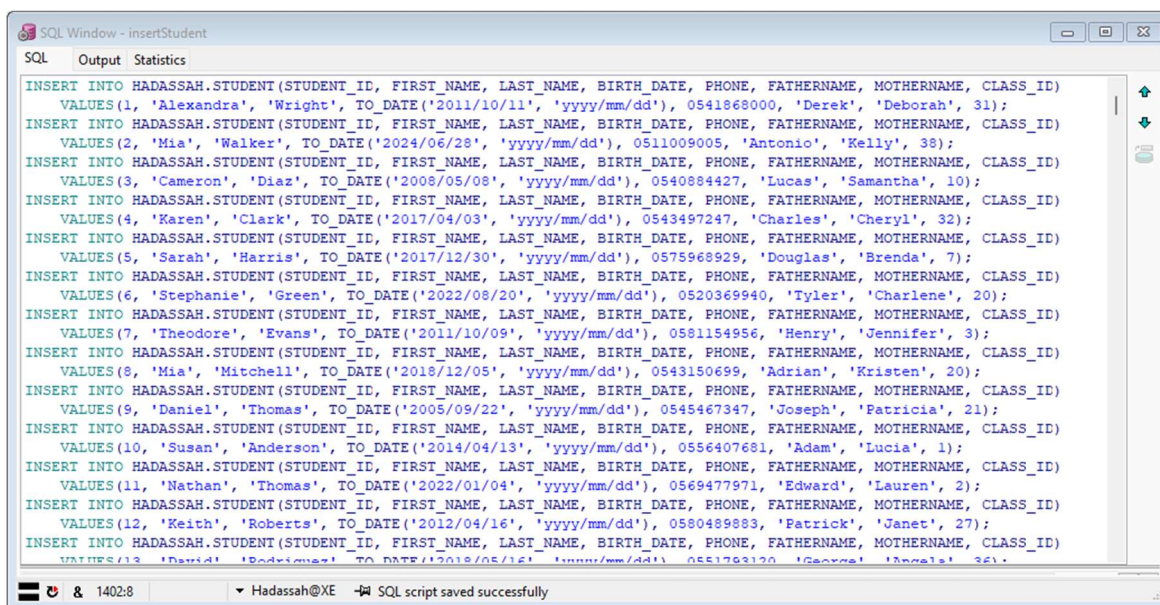




Data Importer



Programming



```

def random_first_name():
    rnd_gender = random.randint(0, 1)
    if (rnd_gender == 0):
        return male_names[random.randint(0, 96)]
    return female_names[random.randint(0, 111)]

def random_last_name():
    return last_names[random.randint(0, 169)]

def random_father_name():
    return male_names[random.randint(0, 96)]

def random_mother_name():
    return female_names[random.randint(0, 111)]

def random_b_date(start_year=2005, end_year=2024):
    """Generates a random date between start_year and end_year."""
    start_date = datetime(start_year, 1, 1)
    end_date = datetime(end_year, 12, 31)
    random_date = start_date + timedelta(days=random.randint(0, (end_date - start_date).days))
    return random_date.strftime('%Y/%m/%d')

def random_phone():
    """Generates a phone number in the format 05[1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]."""
    return f"05{random.randint(10000000, 99999999)}"

def random_class():
    return random.randint(1, 40)

def insert_statement():
    global id
    student_id = id
    id+=1
    first_name = random_first_name()
    last_name = random_last_name()
    birth_date = random_b_date()
    phone = random_phone()
    father = random_father_name()
    mother = random_mother_name()
    class_id = random_class()
    return f"""INSERT INTO HADASSAH.STUDENT(STUDENT_ID, FIRST_NAME, LAST_NAME,
    BIRTH_DATE, PHONE, FATHERNAME, MOTHERNAME, CLASS_ID)
    VALUES({student_id}, '{first_name}', '{last_name}', TO_DATE('{birth_date}', 'yyyy/mm/dd'),
    {phone}, '{father}', '{mother}', {class_id});"""

if __name__ == "__main__":
    with open("insert_commands.txt", "w") as file:
        for i in range(700):
            insert_list = insert_statement()
            file.write(f"{insert_list} \n")

```

גיבוי הנתונים ושיחזור

