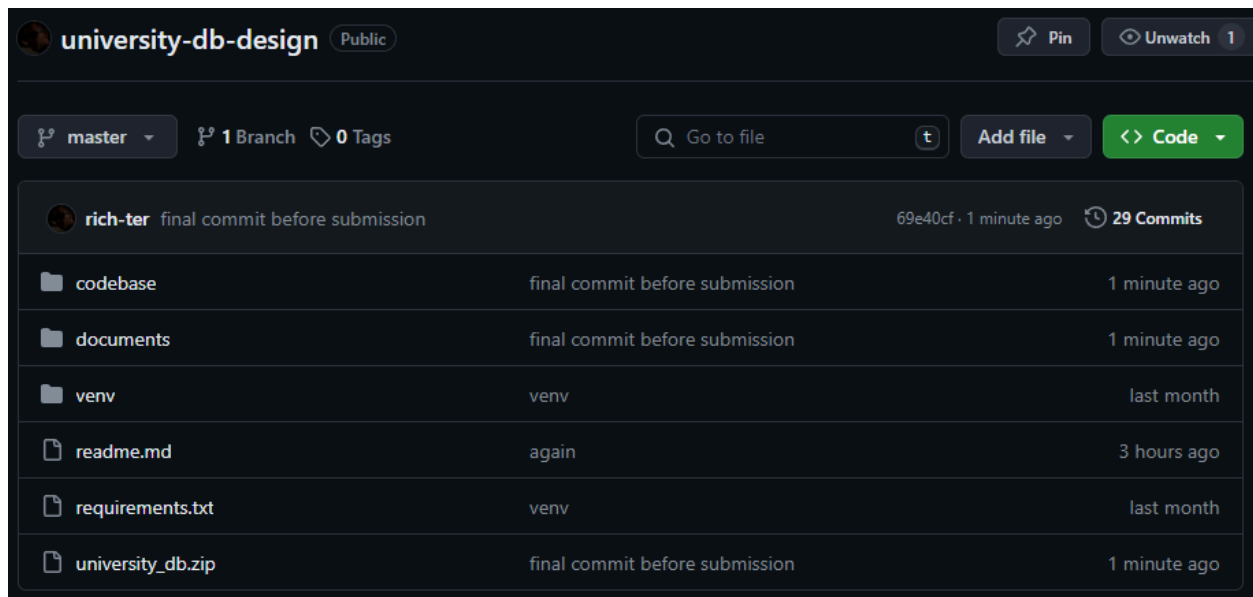


## 1. Εισαγωγή:

Το παρόν αρχείο εξηγεί τα περιεχόμενα της εργασίας, τις τεχνολογίες που χρησιμοποιήθηκαν και εμπεριέχει οδηγίες για τον τρόπο εισαγωγής δεδομένων στην βάση δεδομένων.

Τα αρχεία αποτελούν το σύνολο για τον σχεδιασμό και την υλοποίηση μιας SQL βάσης δεδομένων (MySQL) για ένα πανεπιστημιακό σύστημα διαχείρισης αποφοίτων. Όλα τα αρχεία και η πρόοδος της εργασίας βρίσκεται στο παρακάτω github repository <https://github.com/rich-ter/university-db-design>



### Δομή εργασίας:

#### Codebase

- queries.sql (Ενδεικτικές εντολές ανάκτησης δεδομένων)
- schema.sql (Σχήμα της βάσης δεδομένων)
- indexes.sql (Ευρετήρια πινάκων)
- indicative\_crud.sql (Ενδεικτικές εντολές εισαγωγής, διαγραφής, ενημέρωσης πινάκων)
- stored\_procedures.sql (Αποθηκευμένες διαδικασίες)
- Triggers.sql (Triggers)
- views\_roles.sql (Views, roles)
- db\_operations.py (Εισαγωγή δεδομένων)
- main.py (Διαχείριση προγράμματος)

#### Documents

- db-schema.pdf (Σχεσιακό μοντέλο)
- erd.pdf (Μοντέλο Οντοτήτων Συσχετίσεων)
- installation\_guidelines.pdf (Οδηγίες εγκατάστασης)
- report.pdf (Αναλυτική αναφορά)

#### Venv (Virtual environment with installed dependencies)

- University\_db.sql (Αρχείο βάσης δεδομένων)
- requirements.txt (Python dependencies)

## 2. Τεχνικές Προδιαγραφές Συστημάτων:

MySQL Version: 8.0.27

```
User@DESKTOP-8LIJ8AU MINGW64 ~/OneDrive/Desktop/masters/SQL/Απαλλακτική Δουλκερίδης/sql-exercise (master)
MySQL Shell
MySQL Shell 8.0.27
```

Python Version: 3.9.7

```
User@DESKTOP-8LIJ8AU MINGW64 ~/OneDrive/Desktop/masters/SQL/Απαλλακτική Δουλκερίδης/sql-exercise (master)
$ python --version
Python 3.9.7
(venv)
```

Pip dependencies list

```
(venv)
User@DESKTOP-8LIJ8AU MINGW64 ~/OneDrive/Desktop/masters/SQL/Απαλλακτική Δουλκερίδης/sql-exercise (master)
$ pip list
Package            Version
-----
Faker               22.0.0
mysql               0.0.3
mysql-connector-python 8.2.0
mysqlclient         2.2.1
```

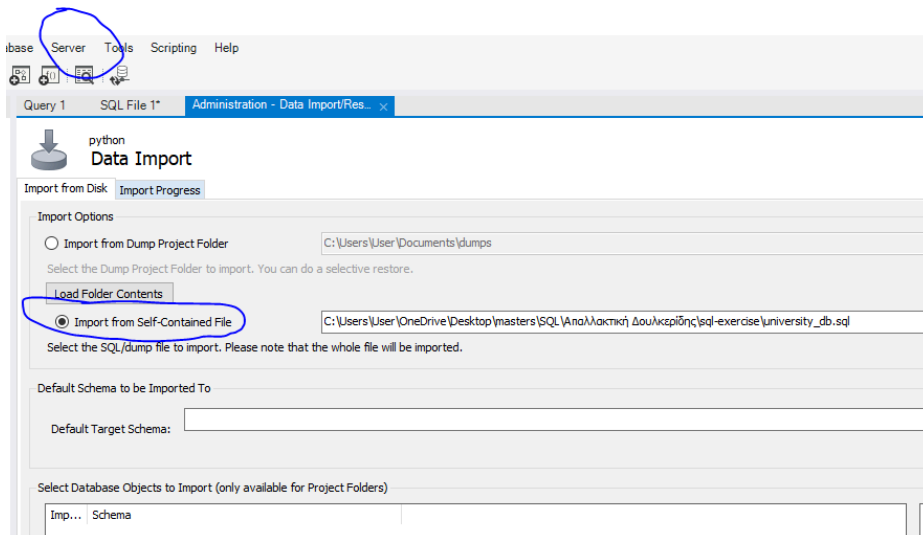
## 3. Οδηγίες Εγκατάστασης βάσης δεδομένων:

**Τρόπος 1:**

Βήμα 1) Λήψη αρχείου university\_db.sql

Βήμα 2) Λήψη και εγκατάσταση του MySQL Workbench <https://dev.mysql.com/downloads/workbench/>

Βήμα 3) Εισαγωγή του αρχείου στο MySQL Workbench μέσω του server -> data import



Βήμα 4) Start import

## Τρόπος 2:

Εκτέλεση του προγράμματος main.py με βάση τα βήματα που περιγράφονται παρακάτω.

### **4. Πληροφορίες για την Εισαγωγή δεδομένων:**

Για την ρεαλιστική απεικόνιση των δεδομένων σε δεδομένα του αληθινού κόσμου και για την καλύτερη ανάδειξη των λειτουργιών της βάσης δεδομένων, χρησιμοποιήθηκε η βιβλιοθήκη faker της python <https://github.com/joke2k/faker> η οποία επιτρέπει να εισάγουμε ρεαλιστικά δεδομένα στην βάση μέσω κώδικα python.

Για την υλοποίησή της εισαγωγής δεδομένων, έχουν δημιουργηθεί 2 αρχεία:

(1) Το **db\_operations.py** το οποίο είναι ένα αρχείο python 1,600+ γραμμών που εμπεριέχει όλες τις εντολές για την δημιουργία πινάκων, δεδομένα σε αυτούς, δημιουργία των stored procedures, triggers, indexes, views και roles.

(2) Το **main.py** το οποίο είναι ένα python αρχείο 100 γραμμών που εμπεριέχει την λογική για την σύνδεση στην βάση δεδομένων, την λογική με την οποία καλούνται τα functions απο το αρχείο db\_operations.py και γενικώς ορίζει τις παραμέτρους όπως username, password για την βάση.

### **Για την εκτέλεση του προγράμματος:**

#### **Βήμα 1: Ενεργοποίηση του virtual environment με τα required dependencies**

Απο το root directory εκτελούμε:

```
. venv/Scripts/activate
```

#### **Βήμα 2: Install the python requirements, i.e. faker library, mysql-connector**

```
pip install -r requirements.txt
```

#### **Βήμα 3: Εκτέλεση του προγράμματος παρακάτω**

```
python main.py
```

**ΣΗΜΕΙΩΣΗ! ΤΑ ΣΤΟΙΧΕΙΑ ΣΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΠΡΕΠΕΙ ΝΑ ΕΙΝΑΙ ΟΠΩΣ ΑΚΡΙΒΩΣ  
ΣΤΟ ΑΡΧΕΙΟ main.py**

Αν επιτυχής, το terminal πρέπει να κατασκευάσει την βάση δεδομένων, τα stored procedures, indexes, triggers, views, roles και να κάνει populate τα αποτελέσματα.

#### **output:**

```
$ python main.py
MySQL Database connection successful
Database 'university_db' created or already exists.
MySQL Database connection successful
```

Indexes created successfully.  
Inserted 12000 records into the Location table.  
Inserted 10000 records into the Student table.  
Inserted 10 records into the University table.  
Inserted 47 records into the Faculty table.  
Inserted 6 records into the EducationLevel table.  
Inserted 26 records into the EducationLevel table.  
Inserted 78 records into the Program table.  
Inserted 58 bachelors records into the Program\_Term table.  
Inserted 75 master's records into the Program\_Term table.  
Inserted 30 PhD records into the Program\_Term table.  
Inserted 19 associates records into the Program\_Term table.  
Inserted 54 records into the Modules table.  
Inserted 10000 records into the StudentModuleParticipation table.  
Inserted 34998 records into the Enrollment table successfully.  
Inserted 300 records into the Company table.  
Inserted 800 records into the JobTitle table.  
Inserted 34998 records into the Graduation table.  
Inserted 10000 records into the Work Experience table.  
Views and roles created successfully.  
Triggers created successfully. (Θα έπρεπε να είναι πριν τους πίνακες αλλά δημιουργεί προβλήματα στην εισαγωγή)  
Stored procedures created successfully.

## 5. Προβλήματα στην εισαγωγή δεδομένων:

- Τα triggers θα έπρεπε να υλοποιηθούν πριν την εισαγωγή δεδομένων, αλλά στην προκειμένη περίπτωση επειδή δημιουργούμε τυχαία πρόγραμμα σπουδών και ύστερα προσθέτουμε ένα πτυχίο με ένα προπτυχιακό σε κάθε μαθητή, τα triggers έχουν υλοποιηθεί ύστερα από την εισαγωγή δεδομένων για την διευκόλυνση της εισαγωγής.
- Η βάση δεδομένων κρατάει πληροφορίες για την ημερομηνία έναρξης ενός Program Term και την ημερομηνία εγγραφής του φοιτητή, αυτά τα δύο δεν συμβαδίζουν. Η εισαγωγή δεδομένων μεγάλου όγκου το κατέστησε αρκετά χρονοβόρο για ταυτοποίηση των δύο και δεν έχει υλοποιηθεί. Αναφέρουμε την ανάγκη για ένα trigger σε δεύτερη φάση του συστήματος .
- Ο κώδικας ρυθον σε ορισμένες περιπτώσεις βάζει εγγραφές φοιτητών στα ίδια προγράμματα σπουδών (διπλότυπα).
- Στο μέλλον πρέπει να εφαρμοστεί περιορισμός για μη αποδοχή εγγραφής εφόσον οι εγγραφές ξεπερνάν το επιτρεπόμενο όριο ανά Program term.
- Στην εισαγωγή δεδομένων με ρυθον το παρακάτω INDEX **δεν** έχει δηλωθεί ως UNIQUE όπως έχει δηλωθεί στην αναφορά καθώς υπάρχουν διπλότυπες εισαγωγές και δημιουργεί προβλήματα.

```
CREATE UNIQUE INDEX idx_student_email_unique ON Student(email);
```

Όλες οι εντολές και λειτουργίες που έχουν αναφερθεί υπάρχουν σε αρχεία ανάλογα με την λειτουργία πχ indexes.sql, stored\_procedures.sql κλπ για την εύκολη ανάγνωση και εκτέλεση αυτών των εντολών απο τον εξεταστή. Βρίσκονται στον φάκελο codebase.