ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Βάσεις Δεδομένων

Εξαμηνιαία Εργασία

Ονοματεπώνυμο: **Αθανάσιος Μάρκου**

Αριθμός μητρώου: **ge18148**

Email: ge18148@mail.ntua.gr
thanosmarkou2@gmail.com

Εξάμηνο 8ο

Ονοματεπώνυμο: **Μάρκος Συρούκης**

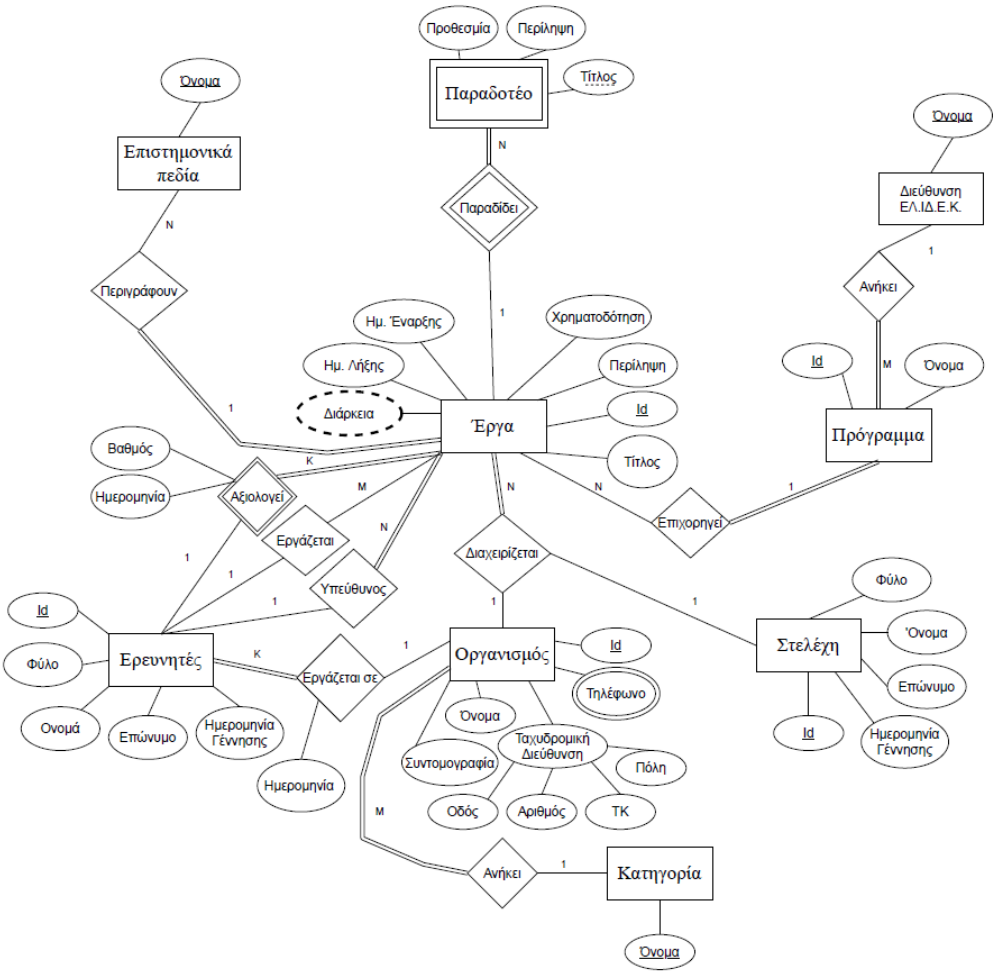
Αριθμός μητρώου: **ge18101**

Email: ge18101@mail.ntua.gr
markossis800@gmail.com

Εξάμηνο 8ο

Περιεχόμενα

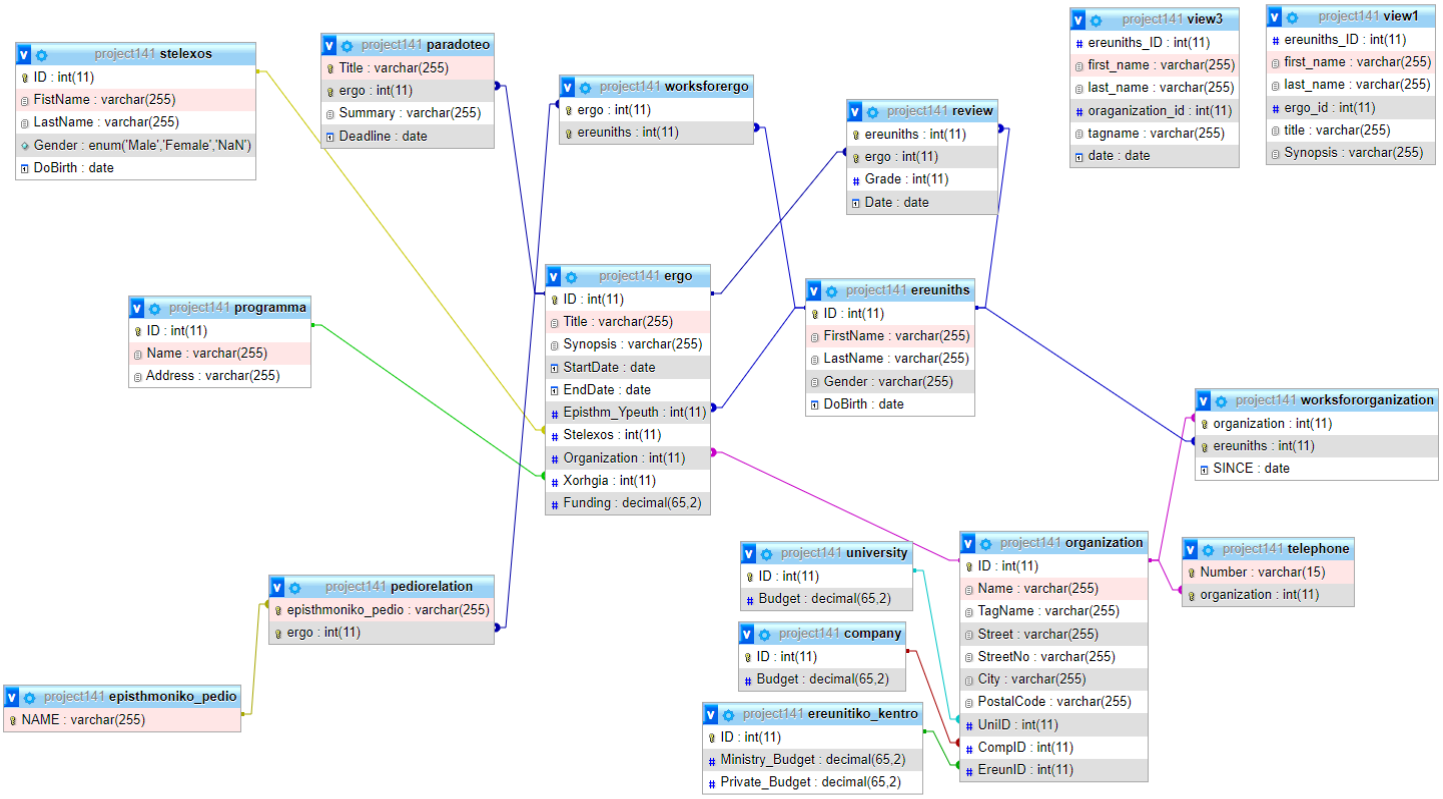
Παρακάτω παραθέτουμε και το ER διάγραμμα



Για την δημιουργία των παραπάνω πινάκων πινάκων στην SQL χρησιμοποιήσαμε το χαρακτηριστικό NOT NULL, στην περίπτωση που το χαρακτηριστικό αυτό θεωρούνταν απαραίτητο για την αρχικοποίηση και την φυσική υπόσταση του στοιχείου. Προφανώς, τα primary keys δεν γίνεται να είναι NULL. Τα foreign keys τα ορίζουμε ως NOT NULL, ώστε να υπάρχει συνέπεια στην βάση και να συμφωνούν με τα primary keys των σχέσεων στις οποίες ανήκουν.

Σχεσιακό Διάγραμμα

Στην συνέχεια υπάρχει το σχεσιακό διάγραμμα της βάσης μας που βασίστηκε στο προηγούμενο ER



Έτσι στο παραπάνω διάγραμμα μπορούμε να δούμε αναλυτικά τα Primary και Foreign Key για κάθε ένα entity, καθώς και τις σχέσεις που δημιουργούνται μεταξύ τους.

Δημιουργία Βάσης

Σαν πρώτο βήμα κατασκευάζουμε ένα query το οποίο θα δημιουργεί την βάση μας με το αντίστοιχο όνομα που επιθυμούμε, επομένως έχουμε:

```
1   GREATE DATABASE project141;
```

- Query για τη δημιουργία του πίνακα Έργο:

```
CREATE TABLE ergo(
    ID int NOT NULL AUTO_INCREMENT,
    Title varchar(255),
    Synopsis varchar(255),
    StartDate DATE,
    EndDate DATE,
    Episthm_Ypeuth int NOT NULL,
    Stelexos int NOT NULL,
    Organization int NOT NULL,
    Xorhgia int NOT NULL,
    Funding DECIMAL(65,2) CHECK (Funding>=0),
    CONSTRAINT test
        CHECK((YEAR(EndDate)-YEAR(StartDate)<=4) AND
              (YEAR(EndDate)-YEAR(StartDate)>=1)),
    FOREIGN KEY (Xorhgia) REFERENCES programma(ID),
    FOREIGN KEY (Episthm_Ypeuth) REFERENCES ereuniths(ID),
    FOREIGN KEY (Stelexos) REFERENCES stelexos(ID),
    FOREIGN KEY (Organization) REFERENCES organization(ID),
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε έργο.

- Query για τη δημιουργία του πίνακα Ερευνητής:

```
CREATE TABLE ereuniths(
    ID int NOT NUll AUTO_INCREMENT,
    FirstName varchar(255),
    LastName varchar(255),
    Gender varchar(255),
    DoBirth DATE,
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε ερευνητή.

- Query για τη δημιουργία του πίνακα Αξιολόγηση:

```
CREATE TABLE review(
    ereuniths int NOT NULL,
    ergo int NOT NULL,
    Grade int NOT NULL,
    Date DATE NOT NULL,
    FOREIGN KEY (ereuniths) REFERENCES ereuniths(ID) ON DELETE CASCADE,
    FOREIGN KEY (ergo) REFERENCES ergo(ID) ON DELETE CASCADE,
    PRIMARY KEY (ereuniths,ergo)
);
```

Εδώ ο ερευνητής και το έργο είναι Foreign Keys στους πίνακες ερευνητής και έργο αντίστοιχα και δημιουργούν μια σχέση ένα(ερευνητής) σε πολλά(έργα).

- Query για τη δημιουργία του πίνακα Οργανισμός:

```
CREATE TABLE organization(
    ID int NOT NULL AUTO_INCREMENT,
    Name varchar(255),
    TagName varchar(255),
    Street varchar(255),
    StreetNo varchar(255),
    City varchar(255),
    PostalCode varchar(255),
    UniID int,
    CompID int,
    EreunID int,
    CONSTRAINT chk
        CHECK((UniID IS NULL AND CompID IS NULL AND EreunID IS NOT NULL) OR
              (UniID IS NULL AND CompID IS NOT NULL AND EreunID IS NULL) OR
              (UniID IS NOT NULL AND CompID IS NULL AND EreunID IS NULL)),
    FOREIGN KEY (UniID) REFERENCES university(ID),
    FOREIGN KEY (CompID) REFERENCES company(ID),
    FOREIGN KEY (EreunID) REFERENCES ereunitiko_kentro(ID),
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε οργανισμό.Επίσης, το πανεπιστήμιο, η εταιρία και το Ερευνητικό κέντρο (3-κατηγορίες) είναι FOREIGN KEYS και δημιουργούν μια σχέση πολλά(οργανισμοί) προς ένα(κατηγορία).

- Query για τη δημιουργία του πίνακα Παραδοτέο:

```
CREATE TABLE paradoteo(
    Title varchar(255) NOT NULL,
    ergo int NOT NULL,
    Summary varchar(255),
    Deadline DATE,
    FOREIGN KEY (ergo) REFERENCES ergo(ID) ON DELETE CASCADE,
    PRIMARY KEY (Title,ergo)
);
```

Εδώ το έργο είναι Foreign Key και δημιουργεί μια σχέση ένα(έργο) σε πολλά(παραδοτέα).

- Query για τη δημιουργία του πίνακα Επιστημονικό Πεδίο:

```
CREATE TABLE episthmoniko_Pedio (
    Name varchar (255),
    PRIMARY KEY (Name)
);
```

Εδώ το Name είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε επιστημονικό πεδίο.

- Query για τη δημιουργία του πίνακα Πρόγραμμα:

```
CREATE TABLE programma(
    ID int NOT NULL AUTO_INCREMENT,
    Name varchar(255),
    Address varchar(255),
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε πρόγραμμα.

- Query για τη δημιουργία του πίνακα Στέλεχος:

```
CREATE TABLE stelexos(
    ID int NOT NUll AUTO_INCREMENT,
    FirstName varchar(255),
    LastName varchar(255),
    Gender varchar(255),
    DoBirth DATE,
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε στέλεχος.

- Query για τη δημιουργία του πίνακα Ερευνητικό Κέντρο:

```
CREATE TABLE ereunitiko_Kentro(
    ID int NOT NULL AUTO_INCREMENT,
    Ministry_Budget DECIMAL(65,2) CHECK (Ministry_Budget>=0),
    Private_Budget DECIMAL(65,2) CHECK (Private_Budget>=0),
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε ερευνηττικό κέντρο.

- Query για τη δημιουργία του πίνακα Εταιρία:

```
CREATE TABLE company(
    ID int NOT NULL AUTO_INCREMENT,
    Budget DECIMAL(65,2) CHECK (Budget>=0),
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε εταιρία.

- Query για τη δημιουργία του πίνακα Πανεπιστήμια:

```
CREATE TABLE university(
    ID int NOT NULL AUTO_INCREMENT,
    Budget DECIMAL(65,2) CHECK (Budget>=0),
    PRIMARY KEY (ID)
);
```

Εδώ το ID είναι Primary Key, και αυτό συμβαίνει, διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε πανεπιστήμιο.

- Query για τη δημιουργία του πίνακα Τηλέφωνα-Οργανισμών:

```
CREATE TABLE telephone(
    Number varchar(15),
    organization int NOT NULL,
    FOREIGN KEY (organization) REFERENCES organization(ID),
    PRIMARY KEY (organization,Number)
);
```

- Query για τη δημιουργία του Relation μεταξύ Έργου-Ερευνητή:

```
CREATE TABLE worksforergo(
    ergo int,
    ereuniths int,
    FOREIGN KEY (ergo) REFERENCES ergo(ID) ON DELETE CASCADE,
    FOREIGN KEY (ereuniths) REFERENCES ereuniths(ID) ON DELETE CASCADE,
    PRIMARY KEY (ergo,ereuniths)
);
```

- Query για τη δημιουργία του Relation μεταξύ Οργανισμού-Ερευνητή:

```
CREATE TABLE worksfororganization(
    organization int NOT NULL,
    ereuniths int NOT NULL,
    SINCE DATE NOT NULL,
    FOREIGN KEY (organization) REFERENCES organization(ID) ON DELETE CASCADE,
    FOREIGN KEY (ereuniths) REFERENCES ereuniths(ID) ON DELETE CASCADE,
    PRIMARY KEY (ereuniths)
);
```

- Query για τη δημιουργία του Relation μεταξύ Έργου-Επιστημονικού Πεδίου:

```
CREATE TABLE pediorelation(
    episthmoniko_pedio varchar(255) NOT NULL,
    ergo int NOT NULL,
    FOREIGN KEY (episthmoniko_pedio) REFERENCES episthmoniko_pedio(Name),
    FOREIGN KEY (ergo) REFERENCES ergo(ID),
    PRIMARY KEY (episthmoniko_pedio,ergo)
);
```

Indexes

```
CREATE INDEX ereuniths_info
ON ereuniths (FirstName, LastName);

CREATE INDEX organization_info
ON organization (Name, TagName);

CREATE INDEX ergo_info
ON ergo (Tilte);

CREATE INDEX stelexos_info
ON stelexos (FistName, LastName);
```

Έχουν οριστεί 4 ευρετήρια με γνώμονα την ελαχιστοποίση του χρόνου που απαιτείται για την υλοποίηση των queries. Μας βοηθούν να εξάγουμε πιο γρήγορα τα δεδομένα, ωστόσο δυσκολεύουν την εισαγωγή νέων δεδομένων στην βάση.

Ερώτημα 3.1

Η εκτέλεση του ερωτήματος μαζί με τα query βρίσκεται στην εφαρμογή app.py

Ερώτημα 3.2

Σε αυτό το ερώτημα μας ζητήθηκε να μπορεί ο χρήστης να μπορεί να δει δύο όψεις του σχεσιακού μοντέλου. Η μία είναι τα έργα που εργάζεται ο κάθε ερευνητής. Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
CREATE VIEW view1
AS
SELECT
    p.ID AS ereuniths_ID,
    p.FirstName AS first_name,
    p.LastName AS last_name,
    pm.ID as ergo_id,
    pm.Title as title,
    pm.Synopsis as Synopsis
FROM WorksForErgo w
    INNER JOIN ereuniths p ON w.ereuniths=p.ID
    INNER JOIN ergo pm ON w.Ergo=pm.ID;
```

- Με την εντολή SELECT διαλέγουμε στοιχεία από το πεδίο 'ερευνητής', όπως το ID του, το όνομα και το επώνυμο, καθώς και από το πεδίο 'έργο', όπως το ID του, τον τίτλο και την περίληψη και τα θέτουμε σε μεταβλητές. Στη συνέχεια, με την χρήση της εντολής INNER JOIN συνδέουμε τα στοιχεία από το πεδίο 'ερευνητής' με αυτά του πεδίου 'έργο', με αποτέλεσμα να επιστρέφει στον χρήστη το έργο που εργάζεται ο κάθε ερευνητής.

- Όσον αφορά την δική μας επιλογή, διαλέξαμε τον οργανισμό ανά ερευνητή. Η διαδικασία είναι αρκετά παρόμοια. Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
CREATE VIEW view3
AS
SELECT
    p.ID AS ereuniths_ID,
    p.FirstName AS first_name,
    p.LastName AS last_name,
    pm.ID as oraganization_id,
    pm.TagName as abbv,
    w.SINCE as date
FROM WorksForOrganization w
    INNER JOIN ereuniths p ON w.ereuniths=p.ID
    INNER JOIN Organization pm ON w.Organization=pm.ID;
```

Και σε αυτήν την περίπτωση, με την εντολή SELECT διαλέγουμε στοιχεία από το πεδίο 'ερευνητής', όπως το ID του, το όνομα και το επώνυμο, καθώς και από το πεδίο 'οργανισμός', το ID του οργανισμού, τη συντομογραφία και την ημερομηνία που ξεκινάει η εργασία του ερευνητή στον κάθε οργανισμό και τα θέτουμε σε μεταβλητές. Στη συνέχεια, με την χρήση της εντολής INNER JOIN συνδέουμε τα στοιχεία από το πεδίο 'ερευνητής' με αυτά του πεδίου 'οργανισμός', με αποτέλεσμα να επιστρέφει στον χρήστη τον οργανισμό που εργάζεται ο κάθε ερευνητής.

Ερώτημα 3.3

Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
SELECT
    e.ID,
    e.Title,
    er.ID,
    er.FirstName,
    er.LastName
FROM
    pediorelation pr
    INNER JOIN ergo e ON (pr.episthmoniko_pedio="{}" AND pr.ergo=e.ID)
    INNER JOIN worksforergo w ON w.ergo=e.ID
    INNER JOIN ereuniths er ON er.ID=w.ereuniths
WHERE
    EndDate > CURRENT_DATE() AND YEAR(CURRENT_DATE())-YEAR(StartDate)<=1
;
```

- Σε αυτήν την περίπτωση ο χρήστης μπορεί να επιλέξει ένα επιστημονικό πεδίο που του κίνησε το ενδιαφέρον. Επιλέγοντάς το, ο χρήστης έχει την δυνατότητα να δει ποια ενεργά έργα χρηματοδοτούνται σε αυτό το πεδίο και ποιοι ερευνητές εργάζονται στο κάθε έργο.

Ερώτημα 3.4

Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
CREATE TEMPORARY TABLE temp(
SELECT
        YEAR(e1.StartDate) as `Year1`,
        org.ID as `organi_ID`,
        org.Name,
        COUNT(org.ID) as `Count`
FROM
    organization org
    JOIN ergo e1 On e1.Organization=org.ID
GROUP BY
    org.ID,`Year1`
HAVING
    `Count`>0
)
```

```
SELECT
    t1.organi_ID,
                t1.Name,
                t1.Count
FROM
    temp t1
                JOIN temp t2 on t1.organi_ID=t2.organi_ID
WHERE
    t1.Year1=t2.Year1 + 1 AND t1.Count=t2.Count
```

- Σε αυτήν την περίπτωση ο χρήστης μπορεί να δει ποιοι οργανισμοί έχουν λάβει τον ίδιο αριθμό έργων σε διάστημα δύο συνεχόμενων ετών, με τουλάχιστον 10 έργα ετησίως. Για να το πετύχουμε αυτό χρησιμοποιήσαμε ένα temporary table για να αποθηκεύσουμε τους οργανισμούς με τουλάχιστον 10 έργα ετησίως και στη συνέχεια επιλέγουμε τους οργανισμούς που έχουν λάβει τον ίδιο αριθμό έργων σε δύο συνεχόμενα έτη.

Ερώτημα 3.5

Παρακάτω δίνεται ο κώδικας σε SQL.

```
SELECT
    p1.episthmoniko_pedio as `Pedio1`,
    p2.episthmoniko_pedio as `Pedio2`,
    COUNT(p1.episthmoniko_pedio) as `count`
FROM
    pediorelation p1, pediorelation p2
WHERE
    p1.episthmoniko_pedio < p2.episthmoniko_pedio AND p1.ergo=p2.ergo
GROUP BY
    p1.episthmoniko_pedio,p2.episthmoniko_pedio
ORDER BY
    'count' DESC
LIMIT 3
;
```

- Σε αυτήν την περίπτωση ο χρήστης μπορεί να δει τα 3 κορυφαία ζεύγη επιστημονικών πεδίων που εμφανίζονται έργα. Για να συμβεί αυτό, επιλέγουμε και αποθηκεύουμε όλα τα διεπιστημονικά έργα και στη συνέχεια τυπώνουμε τα top-3 ζεύγη επιστημονικών πεδίων με την μεγαλύτερη συχνότητα εμφάνισης σε φθίνουσα σειρά.

Ερώτημα 3.6

Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
SELECT
    ereuniths as 'ID',
    FirstName,
    LastName,
    COUNT(ereuniths) as 'count'
FROM
    worksforergo w
    INNER JOIN ereuniths r ON r.ID=w.ereuniths
    INNER JOIN ergo e ON e.ID=w.ergo
WHERE
    YEAR(CURRENT_DATE())-YEAR(DoBirth)<40 AND EndDate>CURRENT_DATE()
GROUP BY
    ereuniths
ORDER BY
    'count' DESC;
```

- Εδώ, ο χρήστης μπορεί να δει τους νέους ερευνητές με ηλικία μικρότερη των 40 ετών που εργάζονται στα περισσότερα ενεργά έργα καθώς και τον αριθμό των έργων που εργάζονται.

Ερώτημα 3.7

Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
SELECT
    s.ID as stel_ID,
    s.FistName,
    s.LastName,
    e.ID,
    e.Title,
    e.Funding
FROM
   ergo e
   INNER JOIN stelexos s ON e.Stelexos=s.ID
ORDER BY
    Funding DESC
LIMIT 5
;
```

- Εδώ, ο χρήστης μπορεί να δει τα top-5 στελέχη που δουλεύουν για το ΕΛ.ΙΔ.Ε.Κ. και έχουν δώσει το μεγαλύτερο ποσό χρηματοδοτήσεων σε μια εταιρεία. Για να το πετύχουμε αυτό, αποθηκεύουμε το ID, το όνομα και το επώνυμο του κάθε στελέχους που διαχειρίζεται το συγκεκριμένο έργο καθώς και το ID, τον τίτλο και την χορηγία κάθε έργου. Τέλος, τυπώνουμε τις 5 υψηλότερες χορηγίες σε φθίνουσα σειρά.

Ερώτημα 3.8

Παρακάτω δίνεται ο κώδικας σε SQL.

```sql
SELECT
    ereuniths as 'ID',
    FirstName,
    LastName,
    COUNT(w.ereuniths) as 'count'
FROM
    paradoteo p, ergo e

    INNER JOIN worksforergo w ON e.ID=w.ergo
    INNER JOIN ereuniths r ON r.ID=w.ereuniths
WHERE
   p.ergo<>e.ID
GROUP BY
    ereuniths
HAVING count>=5
;
```

- Σε αυτήν την περίπτωση, ο χρήστης μπορεί να δει τους ερευνητές που εργάζονται σε τουλάχιστον 5 έργα, τα οποία δεν έχουν παραδοτέο.

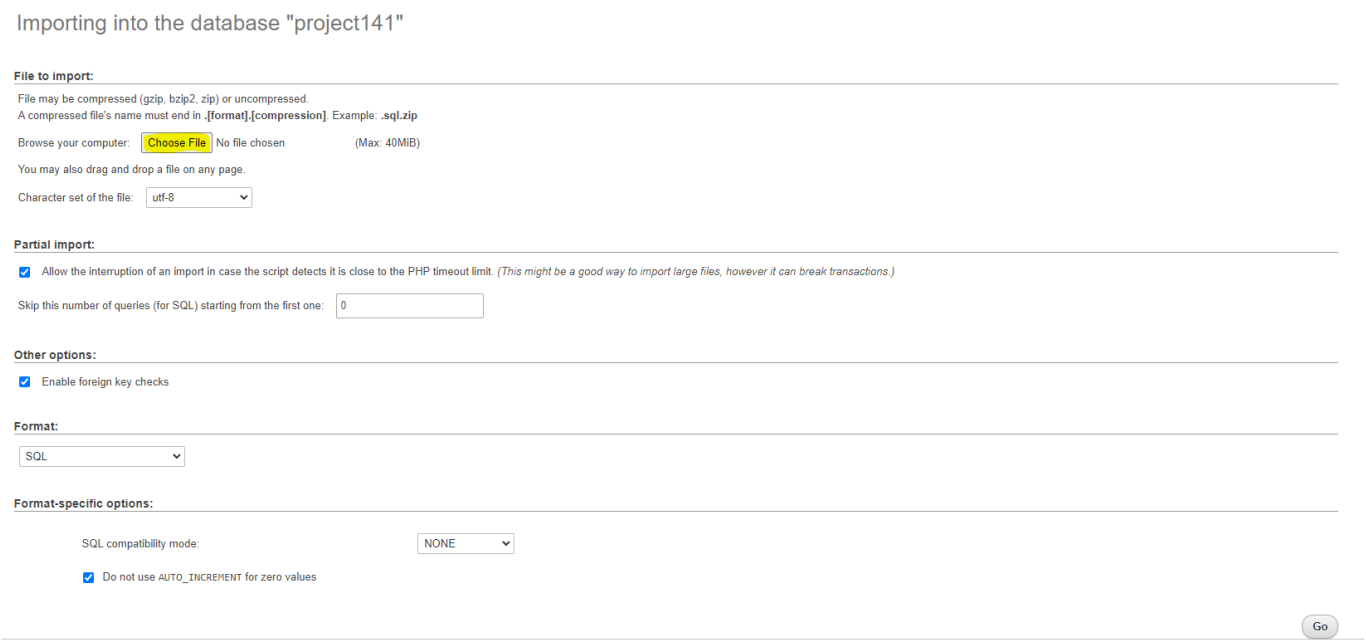
Πλήρης οδηγός εγκατάστασης

Για την διαχείριση και ανάπτυξη της βάσης χρησιμοποιήθηκε το MariaDB και για DBMS χρησιμοποιήθηκε το phpMyAdmin. Για το στήσιμο του web server καθώς και για την σύνδεση μεταξύ βάσης και του server χρησιμοποιείται flask (Python) . .Για το UI χρησιμοποιήθηκε HTML(Visual Studio Code) και Python.

Βήματα εγκατάστασης

1. Αρχικά απαιτείται εγκατάσταση της MariaDB, της Python. Για την σύνδεση του MariaDB με το DBMS ο χρήστης πρέπει να τρέξει το phpMyAdmin. Για να γίνει αυτό, χρειάζεται η εγκατάσταση του XAMPP, το οποίο θα περιέχει την βάση. Μετά, ο χρήστης πρέπει να εγκαταστήσει την Python. Στην Python πρέπει να γίνει εγκατάσταση του πακέτου flask. Αυτό θα γίνει με τις παρακάτω εντολές στο τερματικό
    - pip install mariadb
    - pip install flask_mysqldb
2. Στην συνέχεια για να ανοίξουμε την βάση πρέπει να τρέξουμε την εφαρμογή XAMPP και να πατήσουμε Start στην επιλογή του Apache και στην επιλογή του MySQL. Εφόσον πραγματοποιήσουμε το παραπάνω βήμα θα πρέπει να πατήσουμε το κουμπί Admin ώστε να μας ανοίξει η σελίδα του phpMyAdmin στον webserver που χρησιμοποιούμε.

3. Στην συνέχεια κάνουμε import τα δεδομένα για τα στοιχεία της βάσης αλλά και για τα δεδομένα που υπάρχουν στην βάση μας. Συγκεκριμένα επιλέγουμε στην πάνω μπάρα επιλογών την επιλογή Import και μας ανοίγει η παρακάτω σελίδα:



Σε αυτή την σελίδα επιλέγουμε στο κουμπί Choose file και στην συνέχεια επιλέγουμε το αρχείο DML. Τέλος πατάμε το κουμπί Go κάτω δεξιά και έχουμε την βάση μας. Επαναλαμβάνουμε την ίδια διαδικασία επιλέγοντας το αρχείο DDL, με το οποίο βάζουμε τα δεδομένα μας.

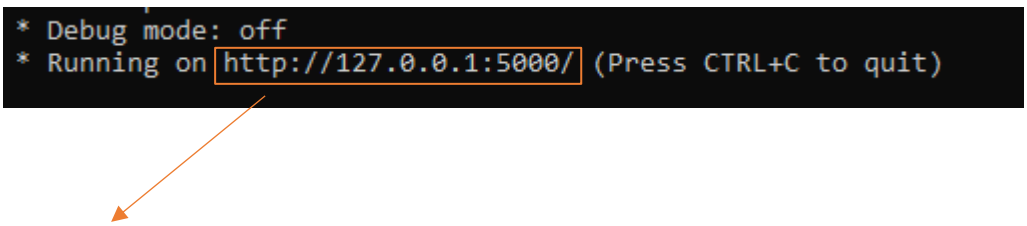
Εκκίνηση εφαρμογής

Για να εκκινήσουμε την εφαρμογή μας ανοίγουμε το αντίστοιχο αρχείο της python, συγκεκριμένα το αρχείο app.py και τρέχουμε τις εντολές.

Στην συνέχεια ανοίγουμε το τερματικό συγκεκριμένα στον φάκελο που έχουμε το αρχείο της python και της html, εκεί θα εκτελούμε την παρακάτω εντολή:

- flask run

Θα εμφανιστεί ένα παρόμοιο παράθυρο με το παρακάτω:



Αντιγράφουμε αυτή την διεύθυνση και την τρέχουμε στο web. Με αυτό τον τρόπο φτάνουμε στο UI.

Τέλος όλα τα αρχεία βρίσκονται στον παρακάτω σύνδεσμο:

https://github.com/MarkosSi34/DataBase.git