**Text

Description automatically generatedΟΜΑΔΑ 100: Μάριος Γραβάνης ΑΜ: 03119850**

**Εμμανουήλ Χατζάκης ΑΜ:03119177**

**Εκφώνηση:** https://helios.ntua.gr/pluginfile.php/142818/mod\_resource/content/2/%CE%B5%CE%BE%CE%B1%CE%BC%CE%B7%CE%BD%CE%B9%CE%B1%CE%AF%CE%B1%20%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CE%AF%CE%B1%20%CE%92%CE%94%202021-2022%20v2.pdf

**Παραδοτέο 1**

Το πρώτο πράγμα που μας ζητήθηκε ήταν να σχεδιάσουμε το ER model για το ΕΛΙΔΕΚ με βάση την εκφώνηση και τα δεδομένα που μας δόθηκαν.

Το ER model που παραδώσαμε φαίνεται παρακάτω και θα είναι μέσα στο github της εργασίας:

Diagram

Description automatically generated

**Παραδοτέο 2**

Με βάση το παραπάνω διάγραμμα που παραδώσαμε και αφού μας δόθηκε και το προτεινόμενο ER model κατασκευάσαμε το σχεσιακό διάγραμμα της βάσης μας το οποίο φαίνεται παρακάτω: Diagram

Description automatically generated

Αφού τελειώσαμε με τα διαγράμματα προχωρήσαμε στην ανάπτυξη της βάσης μας. Λαμβάνοντας υπόψη το σχεσιακό μας διάγραμμα πήραμε τα κατάλληλα κλειδιά(primary και foreign) και σχηματίσαμε τους πίνακες της βάσης μας με τους απαραίτητους περιορισμούς. Τα script των πινάκων φαίνονται παρακάτω:

DDL script για τον πίνακα projects:

Graphical user interface, text

Description automatically generated

DDL script για τον πίνακα researchers:

Graphical user interface, text, application

Description automatically generated

DDL script για τον πίνακα organisations:

Text

Description automatically generated

DDL script για τον πίνακα organization\_phones:

Graphical user interface, text

Description automatically generated

DDL script για τον πίνακα company:

A picture containing text

Description automatically generated

DDL script για τον πίνακα research\_center:

Text

Description automatically generated

DDL script για τον πίνακα university:

A picture containing text

Description automatically generated

DDL script για τον πίνακα executives:

Text

Description automatically generated

DDL script για τον πίνακα Financing\_programs:

Text

Description automatically generated

DDL script για τον πίνακα Deliverable:

Graphical user interface, text

Description automatically generated

DDL script για τον πίνακα manage:

Graphical user interface, text, application

Description automatically generated

DDL script για τον πίνακα Works\_for\_proj:

Text

Description automatically generated with medium confidence

DDL script για τον πίνακα field:

Text

Description automatically generated

Αφού κατασκευάσαμε το σκελετό της βάσης μας αρχίσαμε να γεμίζουμε τη βάση με δεδομένα. Χρησιμοποιήσαμε την εφαρμογή Mockaroo που δημιουργεί fake data ανάλογα με τις παραμέτρους που της δίνουμε. Δημιουργήσαμε τελικά ένα αρχείο excel για κάθε πίνακα το οποίο το μετατρέψαμε σε csv αρχείο και εισαγάγαμε το καθένα ξεχωριστά στη βάση μας. Όλα τα excel φαίνονται στο github.

Με τη βάση έτοιμη γεμάτη με δεδομένα προχωρήσαμε στη υλοποίηση των ερωτημάτων της εργασίας:

Query για το ερώτημα 3.1:

Graphical user interface, text, application

Description automatically generated

Query για το ερώτημα 3.2:

Graphical user interface, text, application

Description automatically generated

Query για το ερώτημα 3.3:

Graphical user interface, text

Description automatically generated

Query για το ερώτημα 3.4:

Text

Description automatically generated

Query για το ερώτημα 3.5:

Text

Description automatically generated

Query για το ερώτημα 3.6:

Graphical user interface, text, application, email

Description automatically generated

Query για το ερώτημα 3.7:

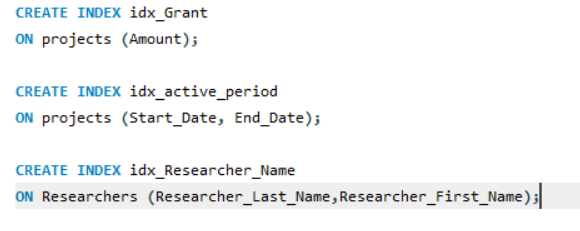
Graphical user interface, text, application

Description automatically generated

Query για το ερώτημα 3.8:Text

Description automatically generated

Επόμενο βήμα ήταν ο ορισμός των indexes για τη βάση μας. Βλέποντας το relational model καθώς και τα queries που χρησιμοποιήσαμε, ορίσαμε 3 indexes τα οποία φαίνονται παρακάτω:



Επιλέξαμε αυτά τα indexes καθώς τα attributes αυτά χρησιμοποιούνται πιο συχνά στα queries μας και θέλουμε η βάση μας να έχει πρόσβαση σε αυτά τα δεδομένα και να μπορεί να τα επεξεργάζεται πιο γρήγορα από ότι χωρίς indexes.

Για το web server χρησιμοποιήσαμε το flask (python) και για να συνδέσουμε την Βάση δεδομένων με τον server mysql.connector. Για το UI επιλέξαμε HTML και για την επικοινωνία μεταξύ frontend και backend χρησιμοποιούνται οι μέθοδοι GET POST

Μέσα στον κώδικα της python έχουμε ενσωματώσει τα SQL Queries .

Ακόμη από την αρχή εγκαταστήσαμε τις απαραίτητες βιβλιοθήκες οι οποίες είναι:

pip install Flask

pip install pygame

pip3 install mysql-connector

Προκειμένου να τρέξει ο server πρέπει να έχουμε τοποθετήσει σωστά τους φακέλους και τότε θα μπούμε στον localhost ή αλλιώς στο 127.0.0.1

Ο web server μαζί με όλα τα κομμάτια της εργασίας βρίσκονται στο παρακάτω link:

https://github.com/TanisG1/qifsharoptis?fbclid=IwAR2KIk5TP7MJR4bVoLsxLBgdXBJ-UJjGfu5CAya-RwLiy7P6CBJa9KyCZs4