ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΑΝΑΦΟΡΑ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ

Εαρινό εξάμηνο 2021-2022

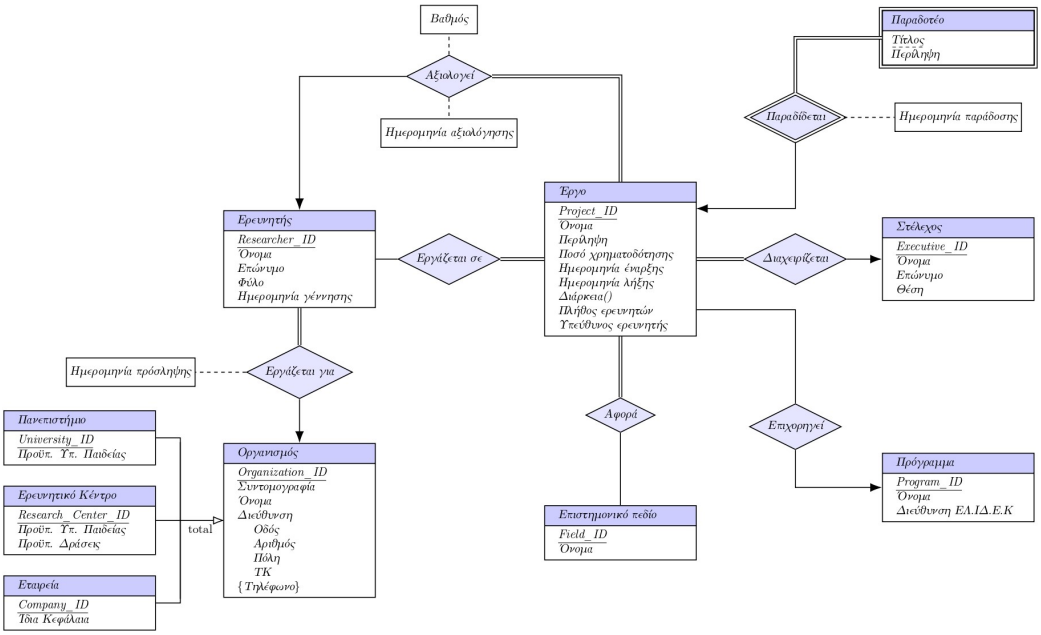
ΟΜΑΔΑ 25

Ονοματεπώνυμο: Μυρσίνη Κελλάρη
Αριθμός Μητρώου: 03119082

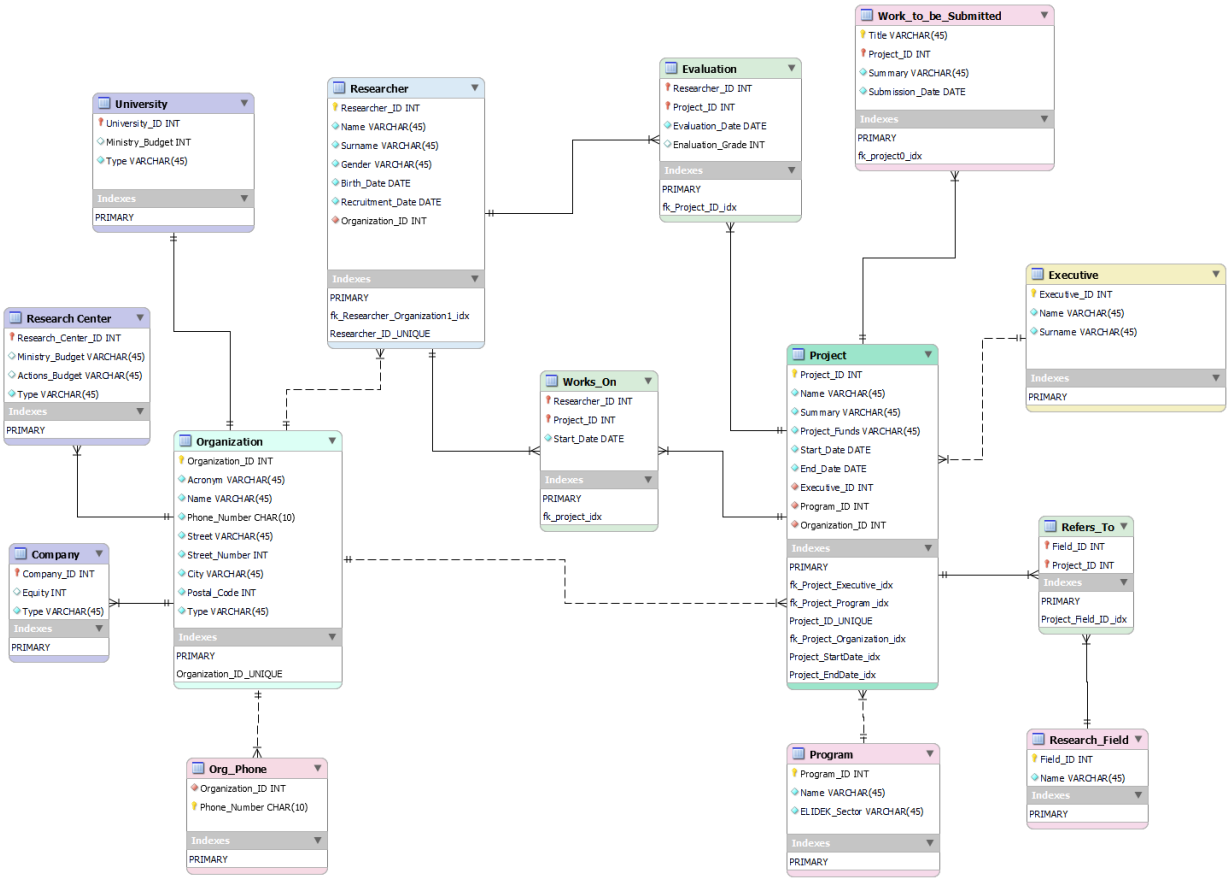
Ονοματεπώνυμο: Χρύσα Πρατικάκη
Αριθμός Μητρώου: 03119131

Ονοματεπώνυμο: Ιωάννης Πρωτόγερος
Αριθμός Μητρώου: 03119008

2.0. Διάγραμμα ER



2.1. Σχεσιακό Διάγραμμα



Σχολιασμός Σχεσιακού Διαγράμματος

Στην βάση μας περιέχονται 3 κύριες οντότητες: οι **ερευνητές** (table "Researcher"), οι **οργανισμοί** (table "Organization") και τα **έργα**/επιχορηγήσεις (table "Project"). Επιπλέον οντότητες που σχετίζονται με τις τρεις κύριες που προαναφέρθηκαν είναι τα παραδοτέα των έργων (table "Work_to_be_submitted"), τα στελέχη (table "Executive"), τα προγράμματα του ΕΛΙΔΕΚ (table "Program") και τα επιστημονικά πεδία (table "Research_Field").

Επιπλέον, η κατηγοριοποίηση των οργανισμών σε πανεπιστήμια, εταιρίες και δράσεις (specialization) έχει πραγματοποιηθεί με τρία επιπλέον tables και την χρήση της ENUM της SQL. Τα τηλέφωνα κάθε οργανισμού βρίσκονται σε ξεχωριστό table ώστε να μπορεί κάθε οργανισμός να έχει τουλάχιστον έναν αριθμό τηλεφώνου.

Τέλος, οι σχέσεις evaluation, works_on και refers_to έχουν υλοποιηθεί και αυτές με tables καθώς εμπεριέχουν επιπλέον χαρατηριστικά για την σχέση, όπως μια ημερομηνία. Μια επιπλέον σχέση που έχει προστεθεί στο διάγραμμα είναι αυτή που συνδέει (με foreign key) κάθε έργο με τον οργανισμό στον οποίο αυτό υπάγεται.

Indexing Βάσης Δεδομένων

Για να ορίσουμε τα indices της βάσης αρχικά λαμβάνουμε υπόψην ότι στη MySQL/InnoDB Engine indices δημιουργούνται αυτόματα για τις κολώνες που αποτελούν primary keys για κάθε table. Αυτό είναι πολύ σημαντικό εφόσον τα primary keys γενικά αποτελούν IDs που χρησιμοποιούνται συνεχώς τόσο στα διάφορα queries (WHERE clauses, JOIN ON (...)), όσο και στα triggers που χρησιμοποιούνται για να υλοποιήσουν περιορισμούς που αφορούν πολλαπλά tables (όπως το ότι ένας ερευνητής πρέπει να εργάζεται σε projects του ίδιου οργανισμού). Επομένως είναι απαραίτητο η πρόσβαση σε κάθε primary key να είναι γρήγορη, κάτι που επιτυγχάνεται με τη χρήση ευρετηρίων. Επιπλέον χρησιμοποιούνται ευρετήρια σε όποια foreign keys ορίζονται και δεν είναι primary keys (οπότε έχουν ήδη ευρετήριο), γιατί είναι απαραίτητο κατά την ενημέρωση/διαγραφή κάποιου στοιχείου το οποίο είναι referenced από άλλο στοιχείο πρέπει να είναι όσο το δυνατόν γρηγορότερη η προσπέλαση του foreign key. Πρέπει να σημειωθεί επίσης όσον αφορά τα composite primary keys που αυτόματα έχουν δικό τους index ότι αν υπάρχει ένα primary key (a,b,c,d) τότε το ίδιο ευρετήριο πάνω στο (a,b,c,d) μπορεί επίσης να χρησιμοποιηθεί και από τα υποσύνολα κολώνων a, (a,b), (a,b,c), όμως οποιοδήποτε άλλο υποσύνολο χρειάζεται το δικό του ευρετήριο. Έτσι για παράδειγμα στο table Works_On που έχει ως primary key το (Researcher_ID, Project_ID) χρειαζόμαστε ευρετήρια και για τις δύο κολώνες, οπότε θα ορίσουμε επιπλέον μόνο το ευρετήριο πάνω στο Project_ID γιατί για το Researcher_ID αρκεί το ευρετήριο πάνω στο (Researcher_ID, Project_ID) που υπάρχει ήδη. Ορίζεται επιπλέον ευρετήριο στις ημερομηνίες έναρξης και λήξης των έργων, εφόσον χρησιμοποιούνται σε πολλά queries της βάσης όπως το φιλτράρισμα των έργων (ερώτημα 3.2). Όλα τα ευρετήρια που ορίζονται μπορούν να βρεθούν στο DDL script elidek_create_schema.sql., ενώ απεικονίζονται και στο άνωθι σχεσιακό διάγραμμα.

Constraints Βάσης Δεδομένων

- Η Start_Date και η End_Date κάθε έργου απέχουν το πολύ από 1 έως και 4 χρόνια.
- Η Evaluation_Date κάθε έργου είναι αυστήρα (χρονικά) πριν από την Start_Date αυτού του έργου.
- Ο Evaluator κάθε έργου δεν γίνεται να ανήκει στην σχέση Works_on για το έργο αυτό.
- Κάθε έργο έχει το πολύ έναν Evaluator.
- Ο Evaluator μπορεί να αξιολογεί έργα μόνο από οργανισμούς στους οποίους αυτός δεν δουλεύει.
- Οι ερευνητές έχουν πάντα ηλικία μεγαλύτερη από 16 έτη.
- Οι ερευνητές μπορούν να εργάζονται μόνο σε έργα που ανήκουν στον ίδιο οργανισμό.
- Η Recruitment_Date κάθε ερευνητή πρέπει να είναι πάντα ανάμεσα στην Start_Date και στην End_Date του έργου.
- Η Submission_Date κάθε παραδοτέου πρέπει να είναι πάντα ανάμεσα στην Start_Date και στην End_Date του έργου.
- Ο οργανισμός στον οποίο εργάζεται κάποιος ευρενητής μπορεί να αλλάξει μόνο εάν ο ερευνητής αυτός δεν δουλεύει σε κανένα ενεργό έργο.
- Ο οργανισμός στον οποίο υπάγεται κάποιο έργο μπορεί να αλλάξει μόνο εάν κανένας ερευνητής δεν εργάζεται στο έργο αυτό.

2.2. DDL και DML scripts

DDL scripts

Στην βάση δεδομένων (φάκελος SQL Code στο git repo) υπάρχουν τα εξής δύο DDL scripts:

elidek_drop_schema.sql:

Διαγράφει από την βάση όλους τους υπάρχοντες πίνακες

```sql
SET FOREIGN_KEY_CHECKS = 0;
DROP TABLE IF EXISTS Company;
DROP TABLE IF EXISTS Evaluation;
DROP TABLE IF EXISTS Executive;
DROP TABLE IF EXISTS Organization;
DROP TABLE IF EXISTS org_phone;
DROP TABLE IF EXISTS program;
DROP TABLE IF EXISTS project;
DROP TABLE IF EXISTS refers_to;
DROP TABLE IF EXISTS research_center;
DROP TABLE IF EXISTS researcher;
DROP TABLE IF EXISTS research_field;
DROP TABLE IF EXISTS university;
DROP TABLE IF EXISTS works_on;
DROP TABLE IF EXISTS work_to_be_submitted;
DROP VIEW IF EXISTS projects_per_researcher;
DROP VIEW IF EXISTS projects_per_field;
DROP VIEW IF EXISTS company_funders;
DROP VIEW IF EXISTS project_count;
SET FOREIGN_KEY_CHECKS = 1;
```

elidek_create_schema.sql:

Δημιουργεί στην βάση όλους τους πίνακες που έχουμε ορίσει καθώς και τις όψεις της βάσης. Σε αυτό το script επίσης δημιουργούνται όλες οι primary key εξαρτήσεις, τα check constraints και τα triggers της βάσης. Παρακάτω φαίνεται παραδειγματικά η δημιουργία του table "Organization" και "Researcher":

```sql
CREATE TABLE IF NOT EXISTS  Organization (
  Organization_ID INT UNSIGNED NOT NULL,
  Acronym VARCHAR(45) NOT NULL,
  Name VARCHAR(45) NOT NULL,
  Street VARCHAR(45) NOT NULL,
  Street_Number INT UNSIGNED NOT NULL,
  City VARCHAR(45) NOT NULL,
  Postal_Code INT UNSIGNED NOT NULL,
  Org_type ENUM('University', 'Company', 'Research Center') NOT NULL,
  CHECK(Postal_Code > 9999 and Postal_Code < 100000),
  PRIMARY KEY (Organization_ID))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS  Researcher (
  Researcher_ID INT UNSIGNED NOT NULL,
  Name VARCHAR(45) NOT NULL,
```

```
  Surname VARCHAR(45) NOT NULL,
  Gender VARCHAR(45) NOT NULL,
  Birth_Date DATE NOT NULL,
  Recruitment_Date DATE NOT NULL,
  Organization_ID INT UNSIGNED NOT NULL,
  CHECK(Gender IN ('Male','Female','Other')),
  CHECK(DATEDIFF(NOW(), Birth_Date) > 5840 AND DATEDIFF(Recruitment_Date
    , NOW()) < 0),
  -- Researcher must be at least 16 years old
  PRIMARY KEY (Researcher_ID),
  INDEX fk_Researcher_Organization1_idx (Organization_ID ASC) ,
  CONSTRAINT fk_Researcher_Organization1
    FOREIGN KEY (Organization_ID)
    REFERENCES Organization (Organization_ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

DML scripts

Στην κατηγορία των DML scripts περιλαμβάνεται το insert script που εισάγει τα αρχικά δεδομένα στην βάση, οι δύο όψεις της βάσεις και τα queries που ζητούνται στην εκφώνηση, καθώς και όλα τα επιμέρους queries που χρησιμοποιούνται για την υλοποίηση της λειτουργίας CRUD στο User Interface της βάσης.

elidek_insert_schema.sql

Η εισαγωγή των αρχικών δεδομένων (πριν δηλαδή υποστούν περαιτέρω επεξεργασία από τον χρήστη) γίνεται με την χρήση της βιβλιοθήκης faker της Python. Ο κώδικας με τον οποίο δημιουργήσαμε τα δεδομένα βρίσκεται στον φάκελο dummy data του git repo υπό το όνομα dummy_generator.py. Αυτή η γεννήτρια δεδομένων παράγει το αρχείο dummy_data.txt το οποίο στην συνέχεια μετονομάζουμε σε elidek_insert_schema.sql και το φορτώνουμε στην βάση.

Queries

Query 3.3

Δεδομένου ότι ένα συγκεκριμένο ερευνητικό πεδίο απέκτησε ιδιαίτερο ενδιαφέρον, ποια έργα χρηματοδοτούνται σε αυτό το πεδίο και ποιοι ερευνητές ασχολούνται με αυτό το πεδίο το τελευταίο έτος;

Έργα που χρηματοδοτούνται στο συγκεκριμένο επιστημονικό πεδίο το τελευταίο έτος:

```
SELECT Project.Project_ID, Name
FROM Project INNER JOIN Refers_To
ON Project.Project_ID = Refers_To.Project_ID
WHERE Field_ID = {ResearchField}
AND DATEDIFF(Project.End_Date, NOW()) > 0
```

Ερευνητές που δραστηριοποιούνται σε αυτό το πεδίο το τελευταίο έτος:

```
SELECT Researcher.Researcher_ID, CONCAT(Researcher.Name,' ',Researcher.
    Surname) AS Full_Name, Works_On.Start_Date
FROM Refers_To INNER JOIN Project
ON Project.Project_ID = Refers_To.Project_ID AND Field_ID = {
    ResearchField}
INNER JOIN Works_On
ON Project.Project_ID = Works_On.Project_ID
INNER JOIN Researcher
ON Works_On.Researcher_ID = Researcher.Researcher_ID
WHERE DATEDIFF(NOW(), Works_On.Start_Date) > 365
GROUP BY Researcher.Researcher_ID
```

Σημειώνεται ότι η μεταβλητή ResearchField προέρχεται από φόρμα της Python και παίρνει την τιμή της από τον χρήστη κατ' επιλογή του.

Query 3.4

Ποιοι οργανισμοί έχουν λάβει τον ίδιο αριθμό έργων σε διάστημα δύο συνεχόμενων ετών, με τουλάχιστον 10 έργα ετησίως;

```
SELECT Organization.Organization_ID, Organization.Name, Organization.
    Acronym,
X.Y AS Year, Projects_This_Year
    FROM
    (
        SELECT DISTINCT Organization_ID AS O, YEAR(Start_Date) as Y,
        (
            SELECT count(*)
            FROM project
            WHERE YEAR(Start_Date) = Y
            AND Organization_ID = O
        ) AS Projects_This_Year,
        (
            SELECT count(*)
```

```
            FROM project
            WHERE YEAR(Start_Date) + 1 = Y
            AND Organization_ID = O
        ) AS Projects_Last_Year
        FROM project
        HAVING Projects_This_Year = Projects_Last_Year AND
    Projects_This_Year >= 10
        ORDER BY O
     ) X INNER JOIN Organization ON Organization.Organization_ID = X.O
```

Query 3.5

Πολλά έργα/επιχορηγήσεις είναι διεπιστημονικά (δηλαδή καλύπτουν περισσότερα από ένα πεδία/ τομείς). Ανάμεσα σε ζεύγη πεδίων (π.χ. επιστήμη των υπολογιστών και μαθηματικά) που είναι κοινά στα έργα, βρείτε τα 3 κορυφαία (top-3) ζεύγη που εμφανίστηκαν σε έργα (ενεργά και μή ενεργά).

```
SELECT Field_1, R1.Name AS Name_1, Field_2, R2.Name AS Name_2,
    pair_count
     FROM
     (
         SELECT DISTINCT X.Field_ID AS Field_1, Y.Field_ID AS Field_2,
             (
                 SELECT COUNT(*)
                 FROM Refers_to XX INNER JOIN Refers_To YY
                 ON XX.Project_ID = YY.Project_ID
                 WHERE XX.Field_ID = X.Field_ID AND YY.Field_ID = Y.
    Field_ID AND XX.Field_ID <> YY.Field_ID
             ) AS pair_count
         FROM Refers_to X INNER JOIN Refers_To Y
         ON X.Project_ID = Y.Project_ID
         WHERE X.Field_ID < Y.Field_ID
         ORDER BY pair_count DESC LIMIT 3
         ) AS top_pairs
         INNER JOIN Research_Field R1 ON R1.Field_ID = top_pairs.Field_1
         INNER JOIN Research_Field R2 ON top_pairs.Field_2 = R2.Field_ID
```

Query 3.6

Βρείτε τους νέους ερευνητές (ηλικία < 40 ετών) που εργάζονται στα περισσότερα ενεργά έργα και τον αριθμό των έργων που εργάζονται.

```
CREATE VIEW IF NOT EXISTS project_count AS
     SELECT DISTINCT Researcher.Researcher_ID AS R_ID, CONCAT(Researcher.
    Name, ' ', Researcher.Surname) AS Full_Name, FLOOR(DATEDIFF(NOW(),
    Birth_Date)/365) AS Age,
     (
         SELECT COUNT(*) FROM Works_On INNER JOIN Researcher
         ON Works_On.Researcher_ID = Researcher.Researcher_ID
         INNER JOIN Project ON Project.Project_ID = Works_On.Project_ID
         WHERE Researcher.Researcher_ID = R_ID AND DATEDIFF(Project.
    End_Date, NOW()) > 0
```

```
    ) AS project_cnt
    FROM Works_On INNER JOIN Researcher
    ON Works_On.Researcher_ID = Researcher.Researcher_ID
    WHERE DATEDIFF(NOW(), Birth_Date) < 365*40
    ORDER BY project_cnt DESC;

SELECT DISTINCT T2.R_ID, T2.Full_Name, T2.Age, T2.project_cnt FROM
    (SELECT * from project_count
    HAVING project_cnt = MAX(project_cnt)) T1
    INNER JOIN project_count T2 ON T1.project_cnt = T2.project_cnt
```

Query 3.7

Βρείτε τα top-5 στελέχη που δουλεύουν για το ΕΛ.ΙΔ.Ε.Κ. και έχουν δώσει το μεγαλύτερο ποσό χρηματοδοτήσεων σε μια εταιρεία.

```
CREATE VIEW IF NOT EXISTS company_funders AS
    SELECT DISTINCT Executive.Executive_ID, CONCAT(Executive.Name, ' ',
   Executive.Surname) AS Full_Name, Organization.Name AS comp_name, SUM(
   Project.Project_Funds) AS Total_Funds FROM
    Executive INNER JOIN Project
    ON Executive.Executive_ID = Project.Executive_ID
    INNER JOIN Organization ON Project.Organization_ID = Organization.
   Organization_ID
    WHERE Organization.Org_Type = "Company"
    GROUP BY Executive.Executive_ID, Organization.Organization_ID
    ORDER BY Total_Funds DESC

SELECT Executive_ID, Full_Name, comp_name, MAX(Total_Funds) as
   new_Total_Funds
    FROM company_funders
    GROUP BY Executive_ID
    ORDER BY new_Total_Funds DESC LIMIT 5
```

Query 3.8

Βρείτε τους ερευνητές που εργάζονται σε 5 ή περισσότερα έργα που δεν έχουν παραδοτέα.

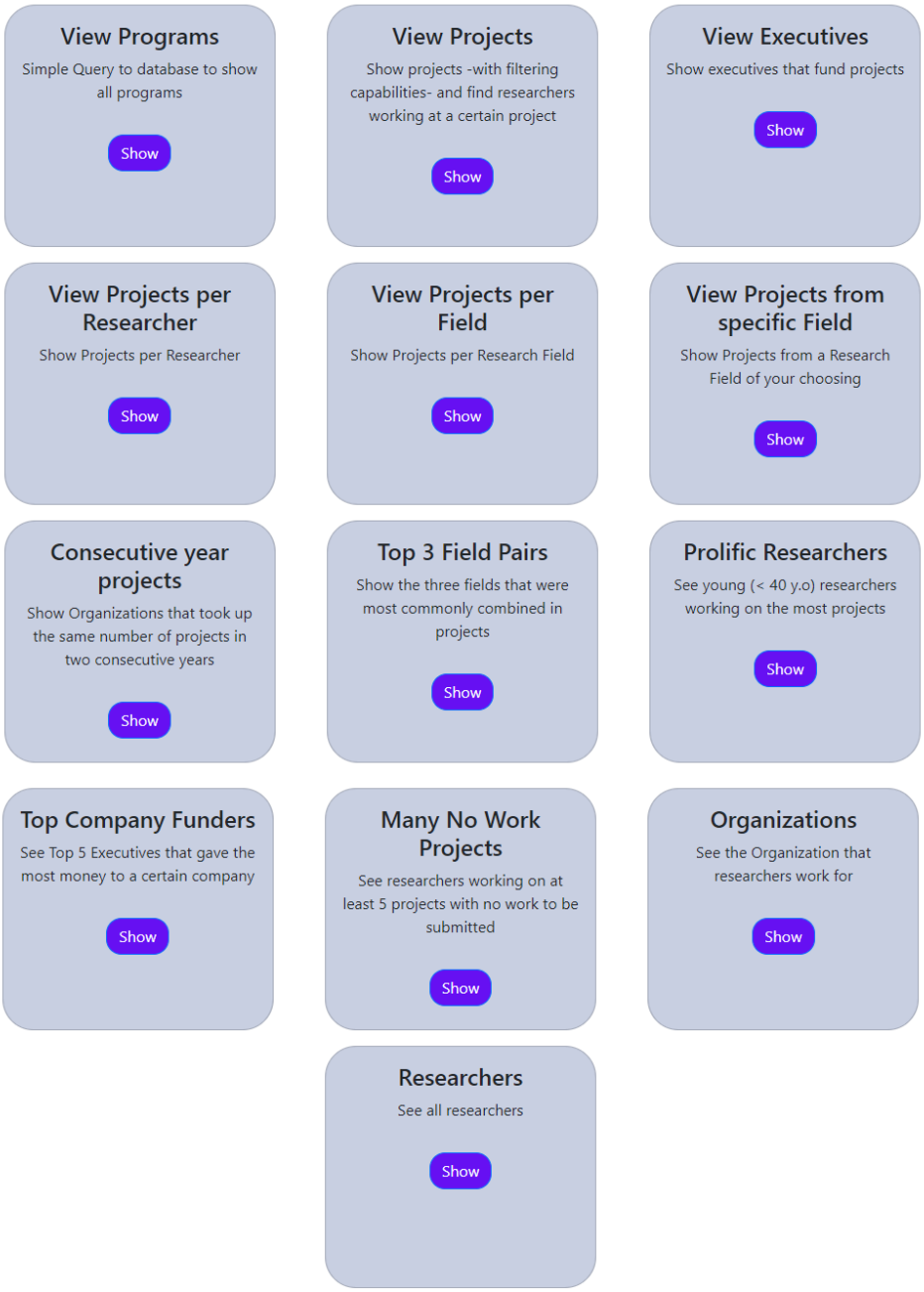
```
SELECT R.Researcher_ID, CONCAT(R.Name,' ',R.Surname) AS Full_Name, COUNT
   (X.Project_ID) AS project_cnt FROM
    (
    SELECT Project_ID from Project
    WHERE Project_ID NOT IN (SELECT Project_ID FROM work_to_be_submitted
   )
    ) X
    INNER JOIN Works_On Y ON X.Project_ID = Y.Project_ID
    INNER JOIN Researcher R ON Y.Researcher_ID = R.Researcher_ID
    GROUP BY R.Researcher_ID
    HAVING project_cnt >= 5

```

Πλοήγηση και CRUD στο User Interface

Η εφαρμογή της βάσης δίνει την δυνατότητα στον χρήστη, όπως και ζητήθηκε, να δει, να επεξεργαστεί, να εισάγει, αλλά και να διαγράψει στοιχεία από την βάση κατ' επιλογή του. Συγκεκριμένα η δυνατότητα του CRUD δίνεται στον χρήστη και για τα 14 tables που υπάρχουν στο σχεσιακό μοντέλο της βάσης.

Για αρχή, ας δούμε την εικόνα του User Interface στο Landing Page ώστε να εξετάσουμε πως μπορεί ο χρήστης να πλοηγηθεί σωστά μέσα στην βάση μας:



Οι δύο πρώτες καρτέλες ("View Programs" και "View Projects") αφορούν το ερώτημα 3.1. για προβολή όλων των προγραμμάτων και των έργων του ΕΛΙΔΕΚ από τον χρήστη. Συγκεκριμένα, στην καρτέλα View Projects υπάρχει και δυνατότητα φιλτραρίσματος των έργων βάσει της διάρκειας, των ημερομηνιών έναρξης και λήξης και του στελέχους. Ακόμη, υπάρχει και η δυνατότητα επεξεργασίας, εισαγωγής και διαγραφής στοιχείων από τα tables Program και Project σε αυτές τις δύο καρτέλες.

Το ερώτημα 3.2., δηλαδή οι δύο όψεις του σχεσιακού μοντέλου, μία για τα έργα ανά ερευνητή και μία για τα έργα ανά επιστημονικό πεδίο είναι προσβάσιμες στον χρήστη μέσω των καρτελών "View Projects per Researcher" και "View Projects per Field" αντίστοιχα.

Στην συνέχεια, τα ερωτήματα (queries) 3.3. έως 3.8. υλοποιούνται κατ' αντιστοιχία στις καρτέλες "View Projects from specific Field" έως "Many No Work Projects".

Η καρτέλα Organizations επιτρέπει στην χρήστη να πλοηγηθεί σε όλα τα στοιχεία της βάσης που αφορούν οργανισμούς και να πραγματοποιήσει την λειτουργία CRUD της επιλογής του. Συγκεκριμένα, έχει πρόσβαση σε κάθε attribute κάθε οργανισμού, στην κατηγορία του κάθε οργανισμού, αλλά και στα τηλέφωνά του.

Τέλος, η καρτέλα Researchers πραγματοποιεί όλα όσα προαναφέρθηκαν για τους οργανισμούς, αλλά πάνω στην οντότητα των ερευνητών.

Ενδεικτικά, μπορούμε να εξετάσουμε την δομή του αρχείου routes.py για να δούμε πως υλοποιείται το CRUD στην οντότητων των προγραμμάτων (πρώτη καρτέλα). Υπάρχουν τέσσερα διαφορετικά directories που αφορούν τα προγράμματα, ένα για κάθε επιθυμητή λειτουργία:

```
@app.route("/programs")

@app.route("/programs/update/<int:program_ID>", methods=["POST"])

@app.route("/programs/delete/<int:program_ID>", methods=["POST"])

@app.route("/programs/create", methods=["GET", "POST"])
```

Σε αυτά τα routes κατά κανόνα ανοίγει μια σύνδεση με την βάση δεδομένων (`cur = db.connection.cursor()`) και στη συνέχεια εκτελούνται τα απαραίτητα queries. Η παραπάνω δομή μπορεί να παρατηρηθεί για όλα τα υπάρχοντα tables της βάσης.

2.3. Οδηγίες Εγκατάστασης και git repo

Το repository της βάσης δεδομένων στο github:

https://github.com/vanourogeros/elidek-DB

Οδηγίες Εγκατάστασης

Βήμα 1ο - Κατέβασμα του repository μέσω git

Για την εγκατάσταση της εφαρμογής πρέπει πρώτα να γίνει clone τοπικά το git repo της εφαρμογής. Αυτό μπορεί να γίνει είτε μέσω της εφαρμογής GitHub desktop, είτε μέσω κάποιου terminal με την εντολή `git clone https://github.com/vanourogeros/elidek-DB` στο τοπικό directory που επιθυμούμε να εγκαταστήσουμε την εφαρμογή.

Βήμα 2ο - Εγκατάσταση της βάσης και εισαγωγή των dummy data

Προκειμένου να εγκαταστήσουμε την βάση στον υπολογιστή μας χρειαζόμαστε έναν sql server (συγκεκριμένα χρησιμοποιήσαμε mysql μέσω xampp) και ένα DBMS (εδώ χρησιμοποιήσαμε Mysql Workbench). Για την εγκατάσταση της βάσης αρκεί να δημιουργήσουμε μία σύνδεση με mySQL server και να τρέξουμε τα scripts elidek_create_schema.sql, elidek_insert_data που βρίσκονται στον φάκελο SQL code, με αυτήν την σειρά.

Βήμα 3ο - Launch της εφαρμογής μέσω local host

Πρώτα εγκαθιστούμε όλες τις απαραίτητες βιβλιοθήκες που αναφέρονται στο αρχείο requirements.txt μέσω της εντολής `pip install -r requirements.txt`. Έπειτα μπορούμε να τρέξουμε την εφαρμογή μέσω της εντολής `python3 run.py` για την εκτέλεση του σχετικού αρχείου. Ανοίγουμε έπειτα έναν browser και μπαίνουμε στη διεύθυνση `localhost:3000`. Η αρχική σελίδα της εφαρμογής πρέπει να εμφανίζεται.