



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ
ΑΝΑΦΟΡΑ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ
Εαρινό εξάμηνο 2021-2022

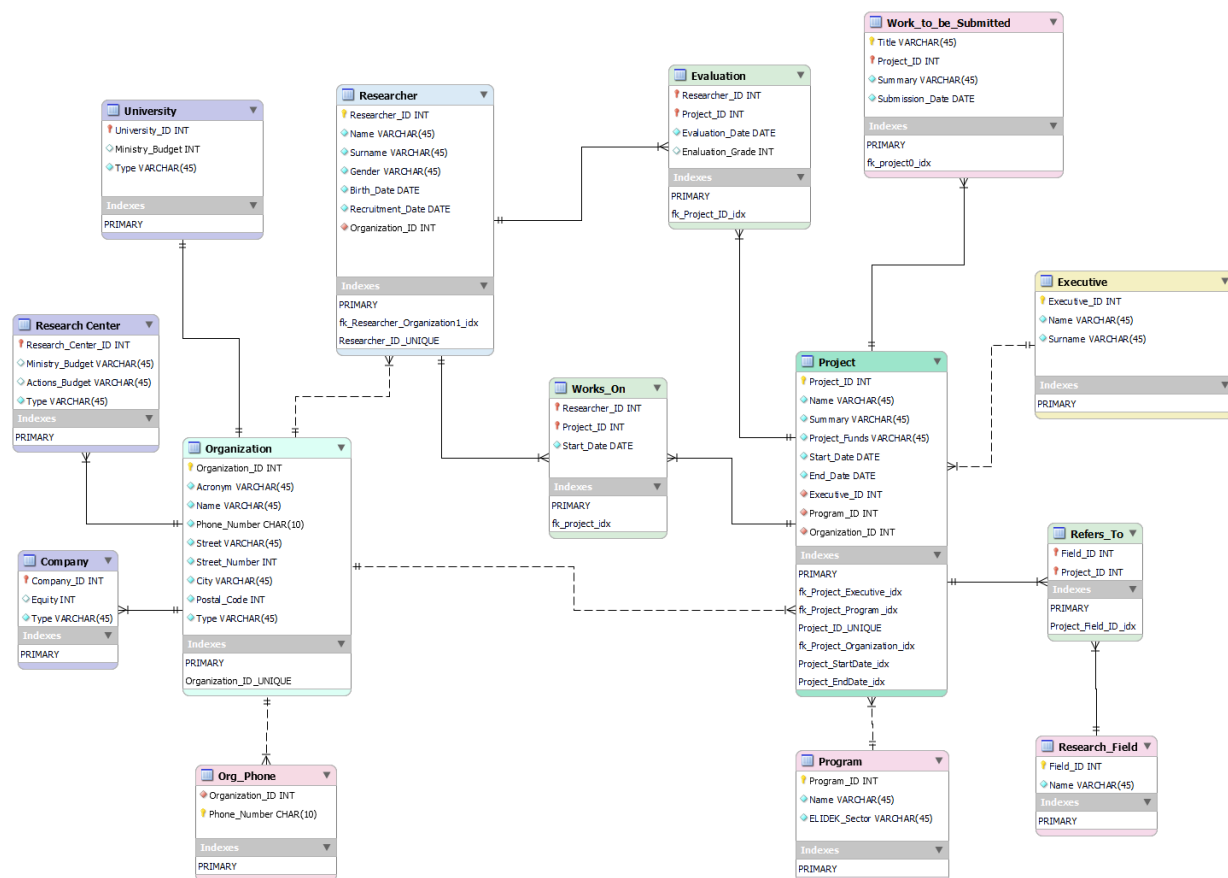
ΟΜΑΔΑ 25

Ονοματεπώνυμο: Μυρσίνη Κελλάρη
Αριθμός Μητρώου: 03119082

Ονοματεπώνυμο: Πορδούλα Πρατικάκη
Αριθμός Μητρώου: 03119131

Ονοματεπώνυμο: Ιωάννης Πρωτόγερος
Αριθμός Μητρώου: 03119008

2.1. Σχεσιακό Διάγραμμα



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").

Indexing Βάσης Δεδομένων

Για να ορίσουμε τα indices της βάσης αρχικά λαμβάνουμε υπόψη 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 pri-

mary 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.

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

```
1 SET FOREIGN_KEY_CHECKS = 0;
2 DROP TABLE IF EXISTS Company;
3 DROP TABLE IF EXISTS Evaluation;
4 DROP TABLE IF EXISTS Executive;
5 DROP TABLE IF EXISTS Organization;
6 DROP TABLE IF EXISTS org_phone;
7 DROP TABLE IF EXISTS program;
8 DROP TABLE IF EXISTS project;
9 DROP TABLE IF EXISTS refers_to;
10 DROP TABLE IF EXISTS research_center;
11 DROP TABLE IF EXISTS researcher;
12 DROP TABLE IF EXISTS research_field;
13 DROP TABLE IF EXISTS university;
14 DROP TABLE IF EXISTS works_on;
15 DROP TABLE IF EXISTS work_to_be_submitted;
16 DROP VIEW IF EXISTS projects_per_researcher;
17 DROP VIEW IF EXISTS projects_per_field;
18 DROP VIEW IF EXISTS company_funders;
19 DROP VIEW IF EXISTS project_count;
20 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":

```
1 CREATE TABLE IF NOT EXISTS Organization (
2     Organization_ID INT UNSIGNED NOT NULL,
3     Acronym VARCHAR(45) NOT NULL,
4     Name VARCHAR(45) NOT NULL,
5     Street VARCHAR(45) NOT NULL,
6     Street_Number INT UNSIGNED NOT NULL,
7     City VARCHAR(45) NOT NULL,
8     Postal_Code INT UNSIGNED NOT NULL,
9     Org_type ENUM('University', 'Company', 'Research Center') NOT NULL,
10    CHECK(Postal_Code > 9999 and Postal_Code < 100000),
11    PRIMARY KEY (Organization_ID))
12 ENGINE = InnoDB;
13
14 CREATE TABLE IF NOT EXISTS Researcher (
15     Researcher_ID INT UNSIGNED NOT NULL,
16     Name VARCHAR(45) NOT NULL,
```

```

17 Surname VARCHAR(45) NOT NULL,
18 Gender VARCHAR(45) NOT NULL,
19 Birth_Date DATE NOT NULL,
20 Recruitment_Date DATE NOT NULL,
21 Organization_ID INT UNSIGNED NOT NULL,
22 CHECK(Gender IN ('Male','Female','Other')),
23 CHECK(DATEDIFF(NOW(), Birth_Date) > 5840 AND DATEDIFF(Recruitment_Date
24 , NOW()) < 0),
25 -- Researcher must be at least 16 years old
26 PRIMARY KEY (Researcher_ID),
27 INDEX fk_Researcher_Organization1_idx (Organization_ID ASC) ,
28 CONSTRAINT fk_Researcher_Organization1
29 FOREIGN KEY (Organization_ID)
30 REFERENCES Organization (Organization_ID)
31 ON DELETE RESTRICT
32 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

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:

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

Ερευνητές που δραστηριοποιούνται σε αυτό το πεδίο το τελευταίο έτος:

```
1 SELECT Researcher.Researcher_ID, CONCAT(Researcher.Name, ' ', Researcher.
   Surname) AS Full_Name, Works_On.Start_Date
2 FROM Refers_To INNER JOIN Project
3 ON Project.Project_ID = Refers_To.Project_ID AND Field_ID = {
   ResearchField}
4 INNER JOIN Works_On
5 ON Project.Project_ID = Works_On.Project_ID
6 INNER JOIN Researcher
7 ON Works_On.Researcher_ID = Researcher.Researcher_ID
8 WHERE DATEDIFF(NOW(), Works_On.Start_Date) > 365
9 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;

```
1 SELECT Organization.Organization_ID, Organization.Name, Organization.
   Acronym,
2 X.Y AS Year, Projects_This_Year
3 FROM
4 (
5     SELECT DISTINCT Organization_ID AS O, YEAR(Start_Date) as Y,
6     (
7         SELECT count(*)
8         FROM project
9         WHERE YEAR(Start_Date) = Y
10        AND Organization_ID = O
11     ) AS Projects_This_Year,
12 (
13     SELECT count(*)
```

```

14         FROM project
15         WHERE YEAR(Start_Date) + 1 = Y
16         AND Organization_ID = 0
17     ) AS Projects_Last_Year
18     FROM project
19     HAVING Projects_This_Year = Projects_Last_Year AND
Projects_This_Year >= 10
20     ORDER BY 0
21     ) X INNER JOIN Organization ON Organization.Organization_ID = X.0

```

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).

```

1 SELECT Field_1, R1.Name AS Name_1, Field_2, R2.Name AS Name_2,
   pair_count
2 FROM
3     (
4         SELECT DISTINCT X.Field_ID AS Field_1, Y.Field_ID AS Field_2,
5             (
6                 SELECT COUNT(*)
7                 FROM Refers_to XX INNER JOIN Refers_To YY
8                 ON XX.Project_ID = YY.Project_ID
9                 WHERE XX.Field_ID = X.Field_ID AND YY.Field_ID = Y.
Field_ID AND XX.Field_ID <> YY.Field_ID
10             ) AS pair_count
11         FROM Refers_to X INNER JOIN Refers_To Y
12         ON X.Project_ID = Y.Project_ID
13         WHERE X.Field_ID < Y.Field_ID
14         ORDER BY pair_count DESC LIMIT 3
15     ) AS top_pairs
16     INNER JOIN Research_Field R1 ON R1.Field_ID = top_pairs.Field_1
17     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.

```

1 CREATE VIEW IF NOT EXISTS project_count AS
2     SELECT DISTINCT Researcher.Researcher_ID AS R_ID, CONCAT(Researcher.
Name, ' ', Researcher.Surname) AS Full_Name, FLOOR(DATEDIFF(NOW(),
Birth_Date)/365) AS Age,
3     (
4         SELECT COUNT(*) FROM Works_On INNER JOIN Researcher
5         ON Works_On.Researcher_ID = Researcher.Researcher_ID
6         INNER JOIN Project ON Project.Project_ID = Works_On.Project_ID
7         WHERE Researcher.Researcher_ID = R_ID AND DATEDIFF(Project.
End_Date, NOW()) > 0

```

```

8      ) AS project_cnt
9      FROM Works_On INNER JOIN Researcher
10     ON Works_On.Researcher_ID = Researcher.Researcher_ID
11     WHERE DATEDIFF(NOW(), Birth_Date) < 365*40
12     ORDER BY project_cnt DESC;
13
14 SELECT DISTINCT T2.R_ID, T2.Full_Name, T2.Age, T2.project_cnt FROM
15     (SELECT * from project_count
16     HAVING project_cnt = MAX(project_cnt)) T1
17     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. (όνομα στελέχους, όνομα εταιρείας και συνολικό ποσό χρηματοδότησης).

```

1 CREATE VIEW IF NOT EXISTS company_funders AS
2     SELECT DISTINCT Executive.Executive_ID, CONCAT(Executive.Name, ' ',
3     Executive.Surname) AS Full_Name, Organization.Name AS comp_name, SUM(
4     Project.Project_Funds) AS Total_Funds FROM
5     Executive INNER JOIN Project
6     ON Executive.Executive_ID = Project.Executive_ID
7     INNER JOIN Organization ON Project.Organization_ID = Organization.
8     Organization_ID
9     WHERE Organization.Org_Type = "Company"
10    GROUP BY Executive.Executive_ID, Organization.Organization_ID
11    ORDER BY Total_Funds DESC
12
13 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 (όνομα ερευνητή και αριθμός έργων).

```

1 SELECT R.Researcher_ID, CONCAT(R.Name, ' ', R.Surname) AS Full_Name, COUNT
2     (X.Project_ID) AS project_cnt FROM
3     (
4     SELECT Project_ID from Project
5     WHERE Project_ID NOT IN (SELECT Project_ID FROM work_to_be_submitted
6     )
7     ) X
8     INNER JOIN Works_On Y ON X.Project_ID = Y.Project_ID
9     INNER JOIN Researcher R ON Y.Researcher_ID = R.Researcher_ID
10    GROUP BY R.Researcher_ID
11    HAVING project_cnt >= 5

```


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.