



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΜΑΘΗΜΑ: ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

-

**ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ
ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2021-22
ΘΕΜΑ: Σύστημα Αποθήκευσης, Διαχείρισης και Ανάλυσης
Πληροφοριών Ιδρύματος ΕΛ.ΙΔ.Ε.Κ.**

-

**ΟΜΑΔΑ Νο 66
Καλογεροπούλου Γεωργία | ΑΜ: 03114174
Μπάτσας Σέργιος | ΑΜ: 03114762
Παπανικόλας Νικόλαος | ΑΜ: 03120624**

Ερώτημα 1: Μοντέλο Οντοτήτων-Σχέσεων Βάσης Δεδομένων

Αρχικά, το πρώτο στάδιο για την υλοποίηση της βάσης αφορούσε τον σχεδιασμό ενός E-R διαγράμματος. Βάσει της εκφώνησης της άσκησης προέκυψαν οι ακόλουθες οντότητες με τα παρακάτω χαρακτηριστικά:

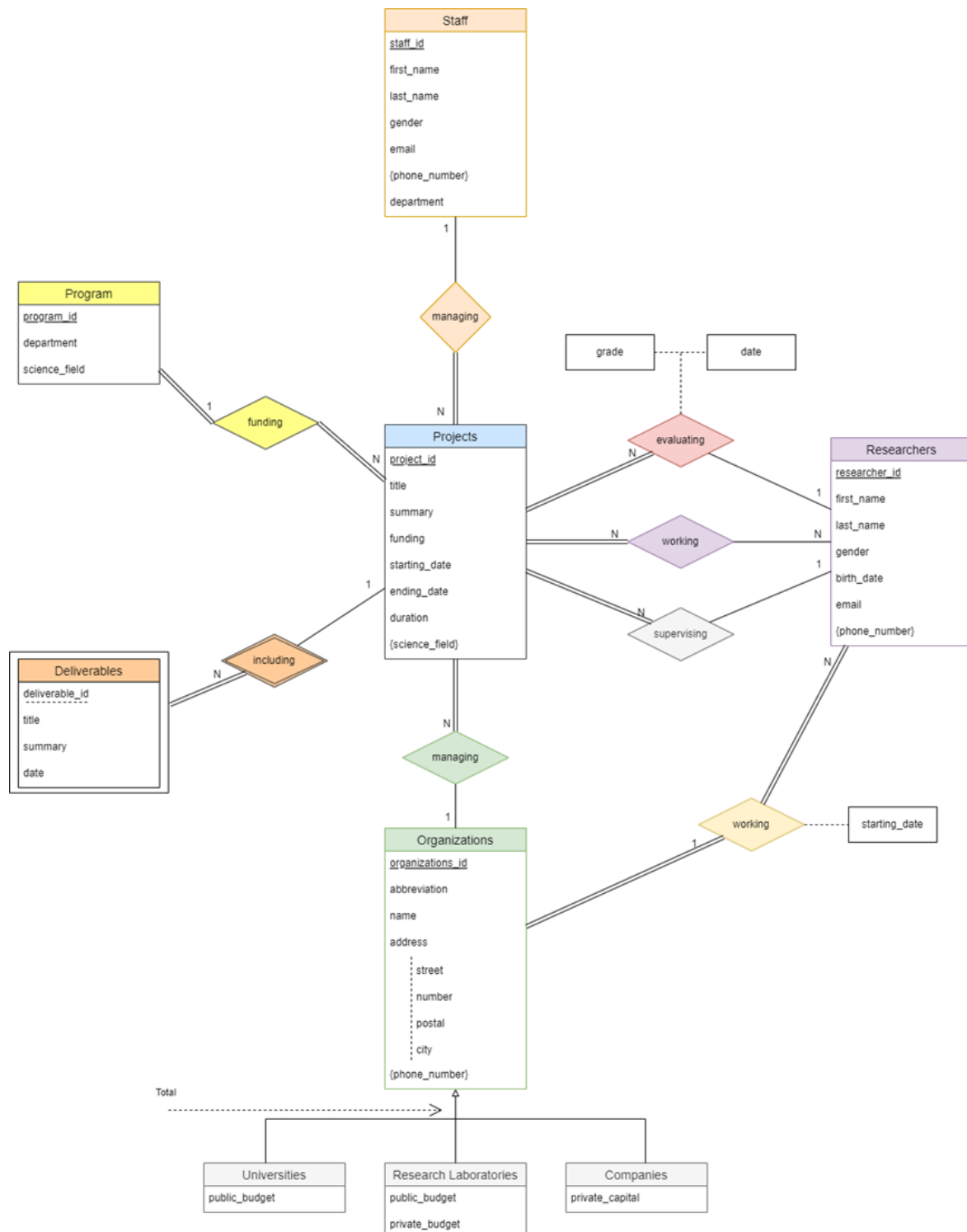
- **Project** με χαρακτηριστικά : project_id , title, summary, funding (100.000-1.000.000), starting_date, ending_date duration (1-4 χρόνια), {science_field} (1 ή περισσότερα)
- **Researchers** με χαρακτηριστικά : researcher_id , first_name , last_name, gender, birth_date, email, {phone_number}
- **Organizations** με χαρακτηριστικά : organization_id , abbreviation, name, address (street, number, postal, city), {phone_number}, category (Universities, Ερευνητικά Κέντρα, Εταιρίες)*
- **Universities** με χαρακτηριστικά : public_budget
- **Research Laboratories** με χαρακτηριστικά : public_budget, private_budget
- **Companies** με χαρακτηριστικά : private_capital
- **Deliverables** (τα οποία αποτελούν ασθενή οντότητα) με χαρακτηριστικά : title, summary, date
- **Program** με χαρακτηριστικά : program_id , department, science_field
- **Staff** με χαρακτηριστικά : staff_id , first_name , last_name, gender, email, {phone_number}, department

Επίσης από την εκφώνηση της άσκησης προέκυψαν με λογικούς συλλογισμούς και οι παρακάτω σχέσεις μεταξύ οντοτήτων:

- **Working** μεταξύ Projects- Researchers (πολλά προς πολλά)
- **Managing** μεταξύ Projects - Organizations (πολλά προς ένα)
- **Supervising** μεταξύ Projects - Researchers (πολλά προς ένα)
- **Working** μεταξύ Organizations - Researchers (ένα προς πολλά) με επιπλέον χαρακτηριστικό το starting_date
- **Funding** μεταξύ Projects - Program (πολλά προς ένα)
- **Managing** μεταξύ Staff - Projects (ένα προς πολλά)

- **Evaluating** μεταξύ Researchers - Projects (ένα προς πολλά) με επιπλέον χαρακτηριστικά τα grade και date
- **Including** μεταξύ Deliverable - Projects (πολλά προς ένα)

Οπότε από τα παραπάνω εξήχθη το ακόλουθο Μοντέλο E-R (Οντοτήτων-Σχέσεων):



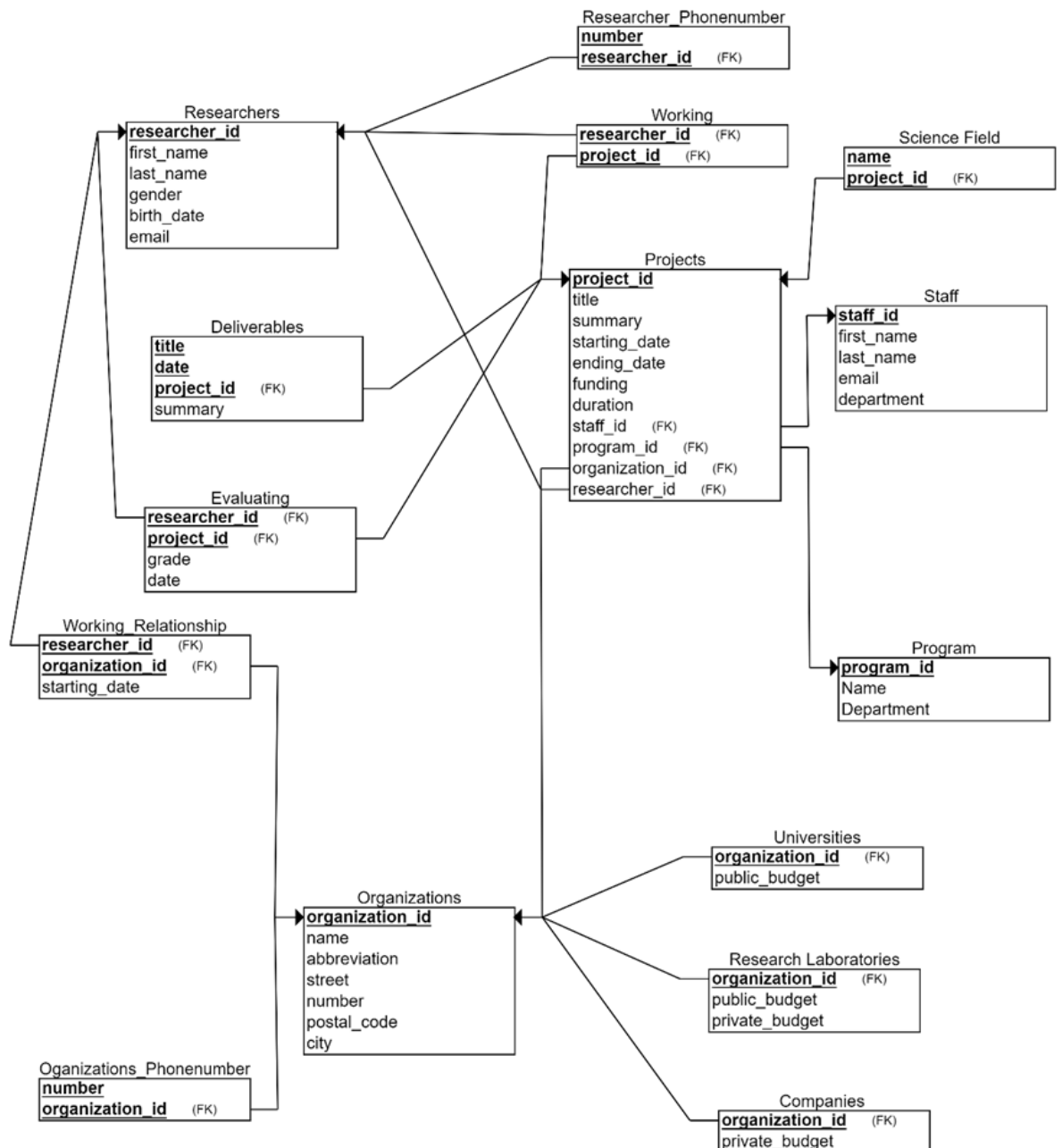
Ερώτημα 2.1: Σχεσιακό Μοντέλο Βάσης Δεδομένων

Έχοντας ως βάση το παραπάνω μοντέλο E-R , το οποίο παραδόθηκε από την ομάδα μας ως πρώτο παραδοτέο της εξαμηνιαίας εργασίας και αφού πραγματοποιήθηκαν οι κατάλληλες αλλαγές, όπου αυτό κρίθηκε απαραίτητο, προχωρήσαμε στην υλοποίηση του Σχεσιακού Μοντέλου της Βάσης μας, το οποίο τελικά περιέχει τις κάτωθι σχέσεις :

- **Program** με χαρακτηριστικά : (Program_ID, Name, Department)
- **Projects** με χαρακτηριστικά : (Project_ID, Title, Summary, Starting_date, Ending_Date, Funding, Duration [το οποίο αποτελεί χαρακτηριστικό το οποίο προκύπτει από τα Starting_date και Ending_Date], Program_ID, Staff_ID, Researcher_ID, Organization_ID)
- **Science_Field** με χαρακτηριστικά : (Name, Project_ID)
- **Staff** με χαρακτηριστικά : (Staff_ID, first_name, last_name, email, Department)
- **Researchers** με χαρακτηριστικά : (Researcher_ID, First_name, Last_name, gender, birth_date, email)
- **Researcher_Phonenummer** με χαρακτηριστικά : (Number, Researcher_ID)
- **Working_Relationship** με χαρακτηριστικά : (Researcher_ID, Organization_ID, starting_date)
- **Working** με χαρακτηριστικά : (Researcher_ID, Project_ID)
- **Evaluating** με χαρακτηριστικά : (Researcher_ID, Project_ID, grade, date)
- **Deliverables** με χαρακτηριστικά : (title, summary, date, project_id)
- **Organizations** με χαρακτηριστικά : (Organization_ID, Name, Abbreviation, Street, Number, Postal_code, City)
- **Organizaton_Phonenummer** με χαρακτηριστικά : (Number, Organization_ID). Η σχέση αυτή προέκυψε λόγω του ότι το συγκεκριμένο χαρακτηριστικό είναι multivalued.
- **Universities** με χαρακτηριστικά : (Organization_ID, public_budget)
- **Research Laboratories** με χαρακτηριστικά : (Organization_ID, public_budget, private_budget)
- **Companies** με χαρακτηριστικά : (Organization_ID, private_budget)

Σε αυτό το σημείο καλό είναι να αναφέρουμε ότι βασιζόμενοι στις παραπάνω σχέσεις, όπου υπήρχε σχέση η οποία αφορούσε σύνδεση οντοτήτων του E-R μοντέλου και μάλιστα αντιστοιχούσε σε σχέση 1 προς πολλά με total participation, έγινε αφομοίωση αυτής μέσω της κληρονόμησης του πρωτεύοντος κλειδιού της οντότητας από την μεριά του 1, ως δευτερεύον κλειδί στην οντότητα από την μεριά many.

Με βάση τα παραπάνω η τελική μορφή του Σχεσιακού Μοντέλου της βάσης είναι η ακόλουθη, πάνω στην οποία βασίστηκε και η υλοποίηση του κώδικα:



Ερώτημα 2.2: DDL και DML SCRIPTS

Παρακάτω παρατίθεται το DDL script της βάσης 'elidek'.

Query για την δημιουργία της βάσης:

```
CREATE DATABASE IF NOT EXISTS `elidek`;  
USE `elidek`;
```

Query για την δημιουργία του πίνακα των ερευνητών 'Researchers':

```
CREATE TABLE IF NOT EXISTS `researchers` (  
  `researcher_id` int(11) NOT NULL AUTO_INCREMENT,  
  `first_name` varchar(100) NOT NULL,  
  `last_name` varchar(100) NOT NULL,  
  `gender` varchar(50) NOT NULL,  
  `birth_date` date NOT NULL,  
  `email` varchar(250) NOT NULL CHECK (`email` like '_%@_%._%'),  
  PRIMARY KEY (`researcher_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα το primary key είναι το "researcher_id", ένα primary key που προσθέσαμε εμείς οι ίδιοι στην βάση μας έτσι ώστε να εξασφαλίσουμε ότι θα είναι μοναδικό και αμετάβλητο.

Query για την δημιουργία του πίνακα των τηλεφώνων των ερευνητών 'Researcher_Phonenummer':

```
CREATE TABLE IF NOT EXISTS `researcher_phonenumber` (  
  `number` bigint(20) NOT NULL CHECK (`number` >= 0 and `number` <= 9999999999),  
  `researcher_id` int(11) NOT NULL,  
  PRIMARY KEY (`number`, `researcher_id`),  
  KEY `researcher_id` (`researcher_id`),  
  CONSTRAINT `researcher_phonenumber_ibfk_1` FOREIGN KEY (`researcher_id`)  
  REFERENCES `researchers` (`researcher_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται το multivalued attribute 'phonenummer' από το ER μας. Έτσι αποφασίσαμε το primary key του πίνακα να είναι το number καθώς και το "researcher_id", ένα foreign key από τον πίνακα των researchers. Με τον τρόπο αυτό καταφέρνουμε ένας researcher να μπορεί να έχει παραπάνω από ένα τηλέφωνα.

Query για την δημιουργία του πίνακα των οργανισμών 'Organizations':

```
CREATE TABLE IF NOT EXISTS `organizations` (
  `organization_id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(150) NOT NULL,
  `abbreviation` varchar(50) NOT NULL,
  `street` varchar(50) NOT NULL,
  `number` int(11) NOT NULL,
  `postal_code` int(11) NOT NULL CHECK (`postal_code` >= 0 and `postal_code` <= 99999),
  `city` varchar(50) NOT NULL,
  PRIMARY KEY (`organization_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

Στον παραπάνω πίνακα το primary key είναι το “organization_id”, ένα primary key που προσθέσαμε εμείς οι ίδιοι στην βάση μας έτσι ώστε να εξασφαλίσουμε ότι θα είναι μοναδικό και αμετάβλητο.

Query για την δημιουργία του πίνακα της σχέσης ανάμεσα στους ερευνητές και στους οργανισμούς που δουλεύουν ‘Organizations’:

```
CREATE TABLE IF NOT EXISTS `working_relationship` (
  `starting_date` date NOT NULL,
  `researcher_id` int(11) NOT NULL,
  `organization_id` int(11) NOT NULL,
  PRIMARY KEY (`researcher_id`, `organization_id`),
  KEY `organization_id` (`organization_id`),
  CONSTRAINT `working_relationship_ibfk_1` FOREIGN KEY (`researcher_id`)
REFERENCES `researchers` (`researcher_id`),
  CONSTRAINT `working_relationship_ibfk_2` FOREIGN KEY (`organization_id`)
REFERENCES `organizations` (`organization_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται η σχέση ανάμεσα στους ερευνητές και στους οργανισμούς στους οποίους ανήκουν. Έτσι αποφασίσαμε το primary key του πίνακα να είναι ο συνδυασμός του ‘organization_id’ και του “researcher_id”, τα οποία είναι και τα δυο foreign keys από τους πίνακες των organizations και researchers αντίστοιχα.

Query για την δημιουργία του πίνακα των τηλεφώνων των οργανισμών ‘Organizations_Phonenumner’:

```
CREATE TABLE IF NOT EXISTS `organizations_phonenumber` (
  `number` bigint(20) NOT NULL CHECK (`number` >= 0 and `number` <= 9999999999),
  `organization_id` int(11) NOT NULL,
  PRIMARY KEY (`number`, `organization_id`),
  KEY `organization_id` (`organization_id`),
  CONSTRAINT `organizations_phonenumber_ibfk_1` FOREIGN KEY (`organization_id`)
REFERENCES `organizations` (`organization_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται το multivalued attribute 'phonenummer' από το ER μας. Έτσι αποφασίσαμε το primary key του πίνακα να είναι το number καθώς και το "organization_id", ένα foreign key από τον πίνακα των organizations. Με τον τρόπο αυτό καταφέρνουμε ένας researcher να μπορεί να έχει παραπάνω από ένα τηλέφωνα.

Queries για την δημιουργία των πινάκων για την κάθε ξεχωριστή κατηγορία που μπορεί να είναι ένας οργανισμός, δηλ Πανεπιστήμια ('Universities'), Ερευνητικά Κέντρα('Research_Laboratories') και Εταιρίες('Companies') αντίστοιχα:

```
CREATE TABLE IF NOT EXISTS `universities` (  
  `public_budget` float NOT NULL,  
  `organization_id` int(11) NOT NULL,  
  PRIMARY KEY (`organization_id`),  
  CONSTRAINT `universities_ibfk_1` FOREIGN KEY (`organization_id`) REFERENCES  
  `organizations` (`organization_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS `research_laboratories` (  
  `public_budget` float NOT NULL,  
  `private_budget` float NOT NULL,  
  `organization_id` int(11) NOT NULL,  
  PRIMARY KEY (`organization_id`),  
  CONSTRAINT `research_laboratories_ibfk_1` FOREIGN KEY (`organization_id`) REFERENCES  
  `organizations` (`organization_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS `companies` (  
  `private_budget` float NOT NULL,  
  `organization_id` int(11) NOT NULL,  
  PRIMARY KEY (`organization_id`),  
  CONSTRAINT `companies_ibfk_1` FOREIGN KEY (`organization_id`) REFERENCES  
  `organizations` (`organization_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στους παραπάνω πίνακες περιγράφονται οι εξειδικευμένοι πίνακες που προέρχονται από την ιεραρχική σχέση που έχουν με τους οργανισμούς. Για τον λόγο αυτό, αποφασίσαμε να έχουν όλοι ως primary key το foreign key 'organization_id' από τον πίνακα 'Organizations'.

Query για την δημιουργία του πίνακα των προγραμμάτων 'Program':

```
CREATE TABLE IF NOT EXISTS `program` (  
  `program_id` int(11) NOT NULL AUTO_INCREMENT,  
  `Name` varchar(100) NOT NULL,  
  `Department` varchar(100) NOT NULL,  
  PRIMARY KEY (`program_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```


Στον παραπάνω πίνακα το primary key είναι το “program_id”, ένα primary key που προσθέσαμε εμείς οι ίδιοι στην βάση μας έτσι ώστε να εξασφαλίσουμε ότι θα είναι μοναδικό και αμετάβλητο.

Query για την δημιουργία του πίνακα των στελεχών ‘Staff’:

```
CREATE TABLE IF NOT EXISTS `staff` (  
  `staff_id` int(11) NOT NULL AUTO_INCREMENT,  
  `first_name` varchar(100) NOT NULL,  
  `last_name` varchar(100) NOT NULL,  
  `email` varchar(250) NOT NULL CHECK (`email` like ' _%@_%._%'),  
  `department` varchar(100) NOT NULL,  
  PRIMARY KEY (`staff_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα το primary key είναι το “staff_id”, ένα primary key που προσθέσαμε εμείς οι ίδιοι στην βάση μας έτσι ώστε να εξασφαλίσουμε ότι θα είναι μοναδικό και αμετάβλητο.

Query για την δημιουργία του πίνακα των έργων/επιχορηγήσεων ‘Projects’:

```
CREATE TABLE IF NOT EXISTS `projects` (  
  `project_id` int(11) NOT NULL AUTO_INCREMENT,  
  `title` varchar(100) NOT NULL,  
  `summary` varchar(500) NOT NULL,  
  `starting_date` date NOT NULL,  
  `ending_date` date NOT NULL,  
  `funding` float NOT NULL CHECK (`funding` >= 100000 and `funding` <= 1000000),  
  `duration` int(11) NOT NULL CHECK (`duration` >= 0 and `duration` <= 1461),  
  `staff_id` int(11) NOT NULL,  
  `program_id` int(11) NOT NULL,  
  `organization_id` int(11) NOT NULL,  
  `researcher_id` int(11) NOT NULL,  
  PRIMARY KEY (`project_id`),  
  KEY `staff_id` (`staff_id`),  
  KEY `program_id` (`program_id`),  
  KEY `organization_id` (`organization_id`),  
  KEY `researcher_id` (`researcher_id`),  
  CONSTRAINT `projects_ibfk_1` FOREIGN KEY (`staff_id`) REFERENCES `staff`  
  (`staff_id`),  
  CONSTRAINT `projects_ibfk_2` FOREIGN KEY (`program_id`) REFERENCES `program`  
  (`program_id`),  
  CONSTRAINT `projects_ibfk_3` FOREIGN KEY (`organization_id`) REFERENCES  
  `organizations` (`organization_id`),  
  CONSTRAINT `projects_ibfk_4` FOREIGN KEY (`researcher_id`) REFERENCES  
  `researchers` (`researcher_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα το primary key είναι το “project_id”, ένα primary key που προσθέσαμε εμείς οι ίδιοι στην βάση μας έτσι ώστε να εξασφαλίσουμε ότι θα είναι μοναδικό και αμετάβλητο. Ο πίνακας επίσης έχει πολλά foreign keys τα οποία αντιπροσωπεύουν τις αντίστοιχες σχέσεις. Αρχικά, το FK staff_id αντιπροσωπεύει την χρηματοδότηση που δίνει ένα στέλεχος του ΕΛΙΔΕΚ στο έργο. Ακόμα, το FK program_id αντιστοιχεί στο πρόγραμμα το οποίο ανήκει το έργο, και το FK organization_id αντιστοιχίζει τον οργανισμό ο οποίος το διαχειρίζεται. Τέλος, το FK researcher_id χαρακτηρίζει τους επιστημονικούς υπεύθυνους των έργων. Όλα τα παραπάνω FK στον πίνακα των projects δημιουργήθηκαν επειδή η σχέση των έργων με τους αντίστοιχους πίνακες είναι πολλά προς ένα με ολοκληρωτική συμμετοχή, με τον πίνακα των έργων να είναι από την πλευρά των πολλών.

Query για την δημιουργία του πίνακα των παραδοτέων των έργων/επιχορηγήσεων ‘Deliverables’:

```
CREATE TABLE IF NOT EXISTS `deliverables` (  
  `title` varchar(100) NOT NULL,  
  `summary` varchar(500) NOT NULL,  
  `date` date NOT NULL,  
  `project_id` int(11) NOT NULL,  
  PRIMARY KEY (`title`,`date`,`project_id`),  
  KEY `project_id` (`project_id`),  
  CONSTRAINT `deliverables_ibfk_1` FOREIGN KEY (`project_id`) REFERENCES  
  `projects` (`project_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται το weak entity ‘deliverables’ από το E-R μας. Έτσι αποφασίσαμε το primary key του πίνακα να είναι το title, date καθώς και το “project_id”, ένα foreign key από τον πίνακα των projects. Με τον τρόπο αυτό καταφέρνουμε να έχουμε ένα συνδυασμό ως primary key έτσι ώστε να εξασφαλίσουμε ότι θα είναι μοναδικό και αμετάβλητο.

Query για την δημιουργία του πίνακα των επιστημονικών πεδίων στα οποία ανήκουν τα έργα/επιχορηγήσεις ‘Science_Field’:

```
CREATE TABLE IF NOT EXISTS `science_field` (  
  `name` varchar(100) NOT NULL,  
  `project_id` int(11) NOT NULL,  
  PRIMARY KEY (`name`,`project_id`),  
  KEY `project_id` (`project_id`),  
  CONSTRAINT `science_field_ibfk_1` FOREIGN KEY (`project_id`) REFERENCES  
  `projects` (`project_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται το multivalued attribute ‘Science_Field’ από το ER μας. Έτσι αποφασίσαμε το primary key του πίνακα να είναι το name καθώς και το “project_id”, ένα foreign key από τον πίνακα των Projects. Με τον τρόπο αυτό καταφέρνουμε ένα project να μπορεί να έχει παραπάνω από ένα επιστημονικά πεδία στα οποία ανήκει.

Query για την δημιουργία του πίνακα της αξιολόγησης των έργων/επιχορηγήσεων 'Evaluating':

```
CREATE TABLE IF NOT EXISTS `evaluating` (  
  `grade` int(11) NOT NULL CHECK (`grade` >= 0 and `grade` <= 10),  
  `date` date NOT NULL,  
  `researcher_id` int(11) NOT NULL,  
  `project_id` int(11) NOT NULL,  
  PRIMARY KEY (`researcher_id`, `project_id`),  
  KEY `project_id` (`project_id`),  
  CONSTRAINT `evaluating_ibfk_1` FOREIGN KEY (`researcher_id`) REFERENCES  
`researchers` (`researcher_id`),  
  CONSTRAINT `evaluating_ibfk_2` FOREIGN KEY (`project_id`) REFERENCES `projects`  
(`project_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται η σχέση ανάμεσα στους ερευνητές και στα έργα/επιχορηγήσεις που αξιολογούν. Η σχέση αυτή είναι ένα προς πολλά, οπότε αποφασίσαμε το primary key του πίνακα να είναι ο συνδυασμός του 'project_id' και του "researcher_id", τα οποία είναι και τα δυο foreign keys από τους πίνακες των projects και researchers αντίστοιχα, μαζί με τα κατάλληλα constraints.

Query για την δημιουργία του πίνακα των ερευνητών που δουλεύουν στα έργα/επιχορηγήσεις 'Working':

```
CREATE TABLE IF NOT EXISTS `working` (  
  `researcher_id` int(11) NOT NULL,  
  `project_id` int(11) NOT NULL,  
  PRIMARY KEY (`researcher_id`, `project_id`),  
  KEY `project_id` (`project_id`),  
  CONSTRAINT `working_ibfk_1` FOREIGN KEY (`researcher_id`) REFERENCES  
`researchers` (`researcher_id`),  
  CONSTRAINT `working_ibfk_2` FOREIGN KEY (`project_id`) REFERENCES `projects`  
(`project_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Στον παραπάνω πίνακα περιγράφεται η σχέση ανάμεσα στους ερευνητές και στα έργα/επιχορηγήσεις στα οποία δουλεύουν. Η σχέση αυτή είναι πολλά προς πολλά, οπότε αποφασίσαμε το primary key του πίνακα να είναι ο συνδυασμός του 'project_id' και του "researcher_id", τα οποία είναι και τα δυο foreign keys από τους πίνακες των projects και researchers αντίστοιχα.

Για το DML script της βάσης “elidek”, παρατίθενται ενδεικτικά τα παρακάτω εντολές. Λόγω του όγκου των data, δεν θα τα μεταφέρουμε όλα στην αναφορά, αλλά βρίσκονται όλα στο git repository μας. Ειδικότερα στο path

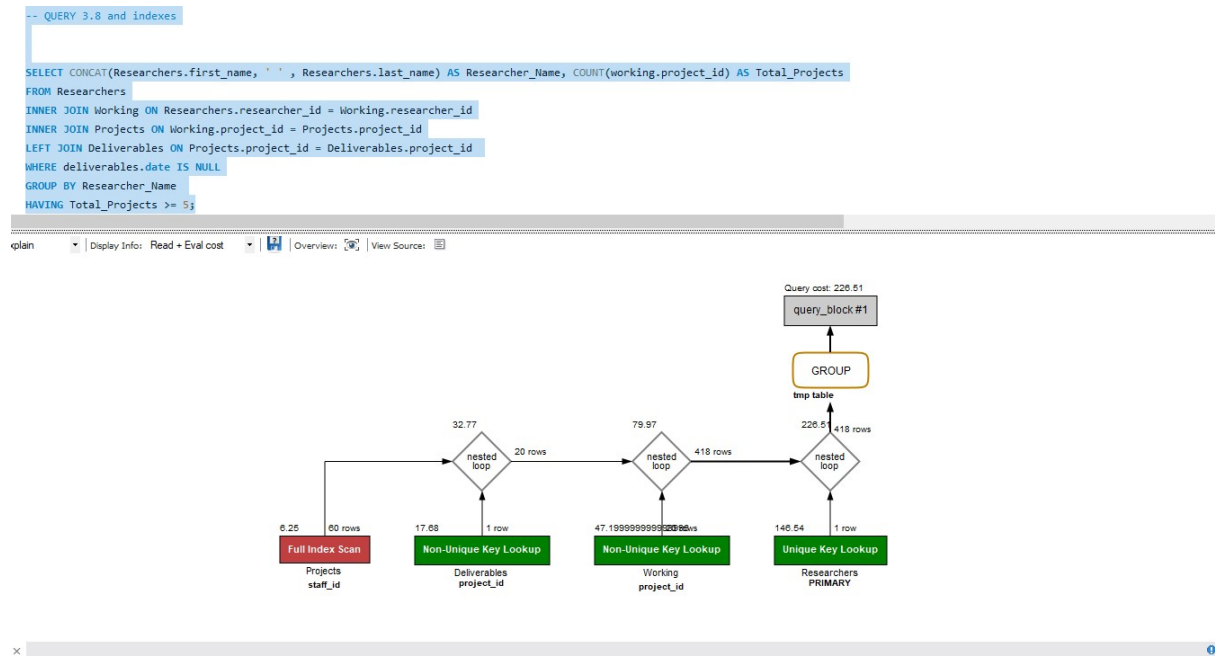
DB_Project_ELIDEK_2022/SQL_Codes/CREATE_Database_Full.sql ή ακολουθώντας το link: https://github.com/Azzy90/DB_Project_ELIDEK_2022/blob/main/SQL_Codes/CREATE_Database_Full.sql

```
INSERT INTO `organizations` (`organization_id`, `name`, `abbreviation`, `street`, `number`,  
`postal_code`, `city`) VALUES  
  (1, 'Pfannerstill, Huels and Mitchell', 'YYTL', 'Onsgard', 82, 56940, 'Domont'),  
  (2, 'Stamm, Nicolas and Schiller', 'FBFQ', 'Barby', 177, 23697, 'Picos'),  
  (3, 'Goodwin Inc', 'ZAIW', 'Superior', 7, 13271, 'Larvik'),  
  ....
```

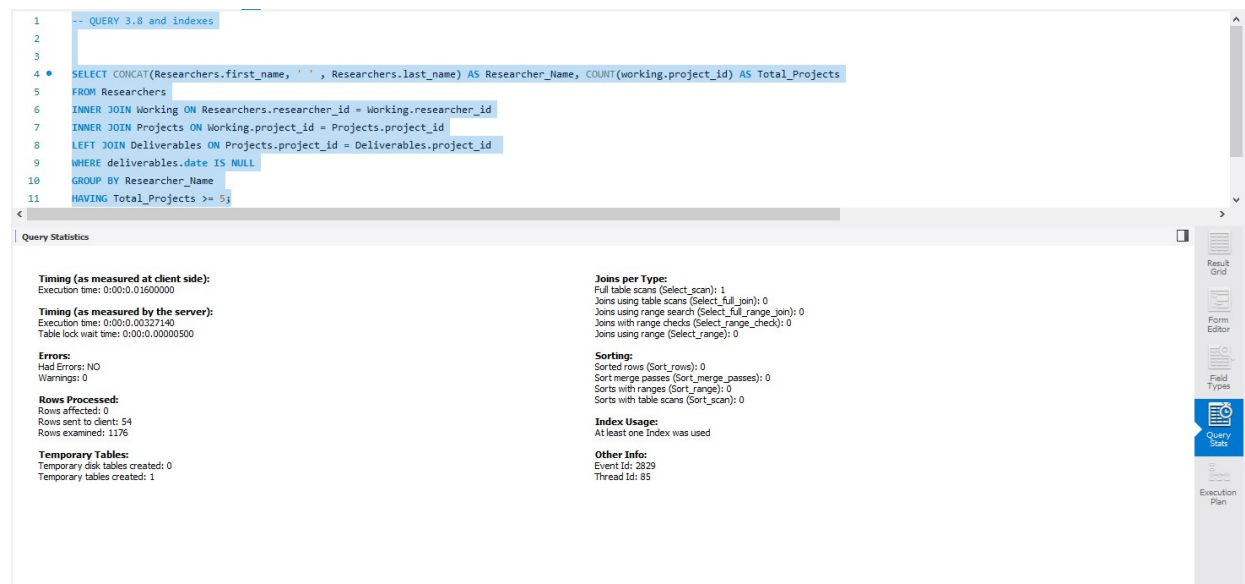
```
INSERT INTO `projects` (`project_id`, `title`, `summary`, `starting_date`, `ending_date`,  
`funding`, `duration`, `staff_id`, `program_id`, `organization_id`, `researcher_id`) VALUES  
  (1, 'Y-Solowarm', 'In hac habitasse platea dictumst. Morbi vestibulum, velit id pretium  
iaculis, diam erat fermentum justo, nec condimentum neque sapien placerat ante. Nulla  
justo.', '2020-03-09', '2023-07-21', 175212, 1229, 6, 7, 5, 24),  
  (2, 'Veribet', 'In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin  
vitae, consectetur eget, rutrum at, lorem. Integer tincidunt ante vel ipsum. Praesent blandit  
lacinia erat.', '2021-07-11', '2023-11-27', 406390, 869, 3, 11, 11, 58),  
  (3, 'Longest', 'Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere  
cubilia Curae; Nulla dapibus dolor vel est. Donec odio justo, sollicitudin ut, suscipit a, feugiat  
et, eros.', '2020-05-24', '2022-04-14', 753140, 690, 2, 31, 16, 82),  
  ....
```

```
INSERT INTO `researchers` (`researcher_id`, `first_name`, `last_name`, `gender`,  
`birth_date`, `email`) VALUES  
  (1, 'Murial', 'Cuthill', 'Female', '1965-08-30', 'mcuthill0@psu.edu'),  
  (2, 'Lou', 'Obell', 'Female', '1986-05-08', 'lobell1@dion.ne.jp'),  
  (3, 'Carmon', 'Leyborne', 'Female', '1973-01-06', 'cleyborne2@oaic.gov.au'),  
  .....
```

Επιπρόσθετα, στην παρούσα εργασία έγινε κατασκευή κάποιων χρήσιμων ευρετηρίων (indexes) λαμβάνοντας υπόψιν ότι όλα τα πρωτεύοντα κλειδιά ενός πίνακα καταχωρούνται αυτόματα σε ευρετήριο και ότι συνήθως υπάρχουν ευρετήρια για τις στήλες που σχετίζονται με ξένα κλειδιά ή για τις στήλες οι οποίες χρησιμοποιούνται συχνά μέσω των queries για την ανάκτηση συγκεκριμένων πληροφοριών. Στην κατασκευή των ευρετηρίων βοήθησε σημαντικά η χρήση των εργαλείων Execution plan και Query Stats αφού κάνοντας αρχικά χρήση του πρώτου εργαλείου μπορούμε να δούμε τον αλγόριθμο αναζήτησης που χρησιμοποιείται από τα Queries που έχουμε φτιάξει και στη συνέχεια μπορούμε να πάρουμε χρήσιμες πληροφορίες σχετικά με το ποιο ευρετήριο θα μπορούσαμε να φτιάξουμε για βελτιστοποίηση του αλγορίθμου και μείωση του χρόνου εκτέλεσης του κώδικά μας όπως παρακάτω :



Με το δεύτερο εργαλείο βλέπουμε και πάλι τους χρόνους εκτέλεσης της κάθε εντολής και ορισμένα ακόμα στατιστικά στοιχεία των queries που τρέχουμε όπως φαίνεται στην ακόλουθη εικόνα:



Βάσει των παραπάνω λοιπόν φτιάχτηκαν τα ακόλουθα indexes:

- ☐ CREATE INDEX idx_project_id ON projects(project_id);
- ☐ CREATE INDEX idx_organization_id ON organizations(organization_id);
- ☐ CREATE INDEX idx_researcher_id ON researchers(researcher_id);

- ☐ CREATE INDEX res_name_idx on researchers (first_name,last_name);
- ☐ CREATE INDEX work_res_id on working(researcher_id);
- ☐ CREATE INDEX idx_project_staff_id ON projects(staff_id);
- ☐ CREATE INDEX idx_project_title ON projects(title);
- ☐ CREATE INDEX idx_project_Fund ON projects(funding);

Ερώτημα 2.3: Αναλυτικά βήματα εγκατάστασης και ανάπτυξης εφαρμογής.

Για την ανάπτυξη της βάσης δεδομένων χρησιμοποιούμε το module MySQL μέσω του XAMPP. Για την διαχείριση της βάσης μας (DBMS) χρησιμοποιήσαμε HeidiSQL. Για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε NodeJS. Πιο συγκεκριμένα, χρησιμοποιήθηκαν τα πακέτα mysql2, express και ejs. Η σύνδεση με τη βάση δεδομένων γινόταν με το mysql2, ο web server που διαχειρίζεται τα GET και POST requests είναι η express. Με την χρήση της γεννήτριας προτύπων (templating engine) ejs, χρησιμοποιούμε στατικές σελίδες όπου κατά την λειτουργία της εφαρμογής γίνεται αντικατάσταση μεταβλητών και μετατρέπεται το πρότυπο σε κανονική σελίδα. Τα πακέτα αυτά είναι απαραίτητα στην υλοποίησή μας και χωρίς αυτά δεν μπορεί να λειτουργήσει. Επιπλέον, χρησιμοποιήσαμε τα προαιρετικά πακέτα custom-env και nodemon. Το δεύτερο χρησιμοποιήθηκε κατά την ανάπτυξη για την αυτόματη ανανέωση της εφαρμογής με κάθε αποθήκευση αρχείου. Το πρώτο χρησιμοποιήθηκε προκειμένου να αποθηκεύουμε κάποιες μεταβλητές περιβάλλοντος σε ξεχωριστό αρχείο για λόγους ασφάλειας και ευκολίας.

Versions:

- MySQL - Ver 15.1 Distrib 10.4.24-MariaDB
- NodeJS - v.16.14.0 LTS
- Express - v.4.18.1
- Mysql2 - v.2.3.3
- Ejs - v.3.1.8
- Custom-env - v.2.0.1
- Nodemon - v.2.0.16

Βήματα εγκατάστασης εφαρμογής σε λογισμικό Windows

1. Κατεβάζουμε τα αρχεία της εφαρμογής μας.
2. Εγκαθιστούμε το XAMPP το οποίο και ανοίγουμε μετά την εγκατάσταση.
3. Κάνουμε Start το Module της MySQL.
4. Πατάμε στο κουμπί Shell και στο παράθυρο που ανοίγει γράφουμε mysql --version
5. Ελέγχουμε ότι η έκδοση που μας εμφάνισε είναι η σωστή για την εφαρμογή μας (τουλάχιστον Ver 15.0)
6. Κατεβάζουμε το επιθυμητό DBMS και το εγκαθιστούμε. Προτεινόμενα: HeidiSQL, Navicat, MySQL Workbench.
7. Δημιουργούμε την βάση δεδομένων με χρήση του αρχείου CREATE_Database.sql (είτε με import το αρχείο είτε σαν query)

8. Εισάγουμε τα Mock Data μας (από τον ομώνυμο φάκελο) με την κάτωθι σειρά: Organizations, Companies, Universities, Research Laboratories, Staff, Researchers, Organizations_Phonenumbers, Researcher_Phonenumbers, Program, Working_Relationship, Projects, Evaluating, Deliverables, Science_Field, Working. Η εισαγωγή είναι προτεινόμενο να γίνει με χρήση των αρχείων .csv
9. Εναλλακτικά μπορούμε να κάνουμε import το αρχείο CREATE_Database_Full.sql από τον φάκελο SQL_Codes ο οποίο περιέχει και τα Mock Data.
10. Εκτελούμε στην νέα μας βάση με όνομα elidek, το Query στο αρχείο 3.2.sql (πάλι στον φάκελο SQL_Codes) προκειμένου να δημιουργήσουμε τα απαιτούμενα VIEWS.
11. Εγκαθιστούμε το NodeJS (16.14.x LTS) στον υπολογιστή μας και ανοίγουμε ένα Windows Terminal (Γραμμή Εντολών) και επιβεβαιώνουμε την έκδοση μας με την εντολή node -v.
12. Επεξεργαζόμαστε το αρχείο env-example.txt με τις δικές μας μεταβλητές και το μετονομάζουμε σε .env.localhost με Save As (Αποθήκευση Ως) και την επιλογή Όλα τα Αρχεία
13. Ανοίγουμε ένα νέο Windows Terminal (Γραμμή Εντολών) και προηγούμεστε στον φάκελο που κατεβάσαμε με την εφαρμογή μας με τις εντολές cd [Directory Name] κτλ (βοήθημα)
14. Όταν είμαστε στον φάκελο της εργασίας μας, (πχ. C:\Users\username\Downloads\DB_Project_ELIDEK_2022\) εκτελούμε την εντολή cd Project για να βρεθούμε στον ομώνυμο φάκελο.
15. Εκτελούμε την εντολή npm install για την εγκατάσταση των απαραίτητων πακέτων.
16. Εκτελούμε την εντολή npm start για την εκκίνηση του Server μας.
17. Βλέπουμε το εικονιζόμενο μήνυμα στο terminal μας ότι ο Server μας «τρέχει»
18. Επισκεπτόμαστε την ιστοσελίδα <http://localhost:3000/> (ή αν στο αρχείο .env.localhost έχουμε αλλάξει την μεταβλητή PORT τότε αντί για 3000 γράφουμε την δικιά μας)
19. Η εφαρμογή μας λειτουργεί κανονικά και μπορούμε να περιηγηθούμε σε αυτή και να δούμε τα αποτελέσματα των απαιτούμενων ερωτημάτων, καθώς και να επέμβουμε στην βάση μας προσθέτοντας, διαγράφοντας ή και αλλάζοντας τα δεδομένα μας κατευθείαν από την ιστοσελίδα.

Ερώτημα 2.4:Σύνδεσμος για το git repo της εφαρμογής

Στο repo του παρακάτω συνδέσμου εκτός των προαναφερθέντων, επιπλέον βρίσκεται και ο πηγαίος κώδικας της εφαρμογής μας.

https://github.com/Azzy90/DB_Project_ELIDEK_2022

Το repo θα γίνει public τα μεσάνυχτα της Κυριακής 5/6/2022, αφού έχει παραδοθεί η εργασία.

Σε περίπτωση που δεν λειτουργεί το λίνκ, αναζητήστε το repo στο github με όνομα DB_Project_ELIDEK_2022 από τον χρήστη Azzy90.