Ομάδα 136  
Πνευματικού Αγγελική 03119948  
Παπανικόλας Εμμανουήλ 03119063  
Ρουμελιώτης Ιωάννης 03119103

**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ

Ακαδημαϊκό έτος 2021-2022

Για να βρείτε ολόκληρη την εργασία μας, ακολουθήστε τον παρακάτω σύνδεσμο:

<https://github.com/giannisrou/DB_for_dummies.git>

Περιεχόμενα Αναφοράς:

1. Σχεσιακό Διάγραμμα της Βάσης Δεδομένων

Α. Περιορισμοί

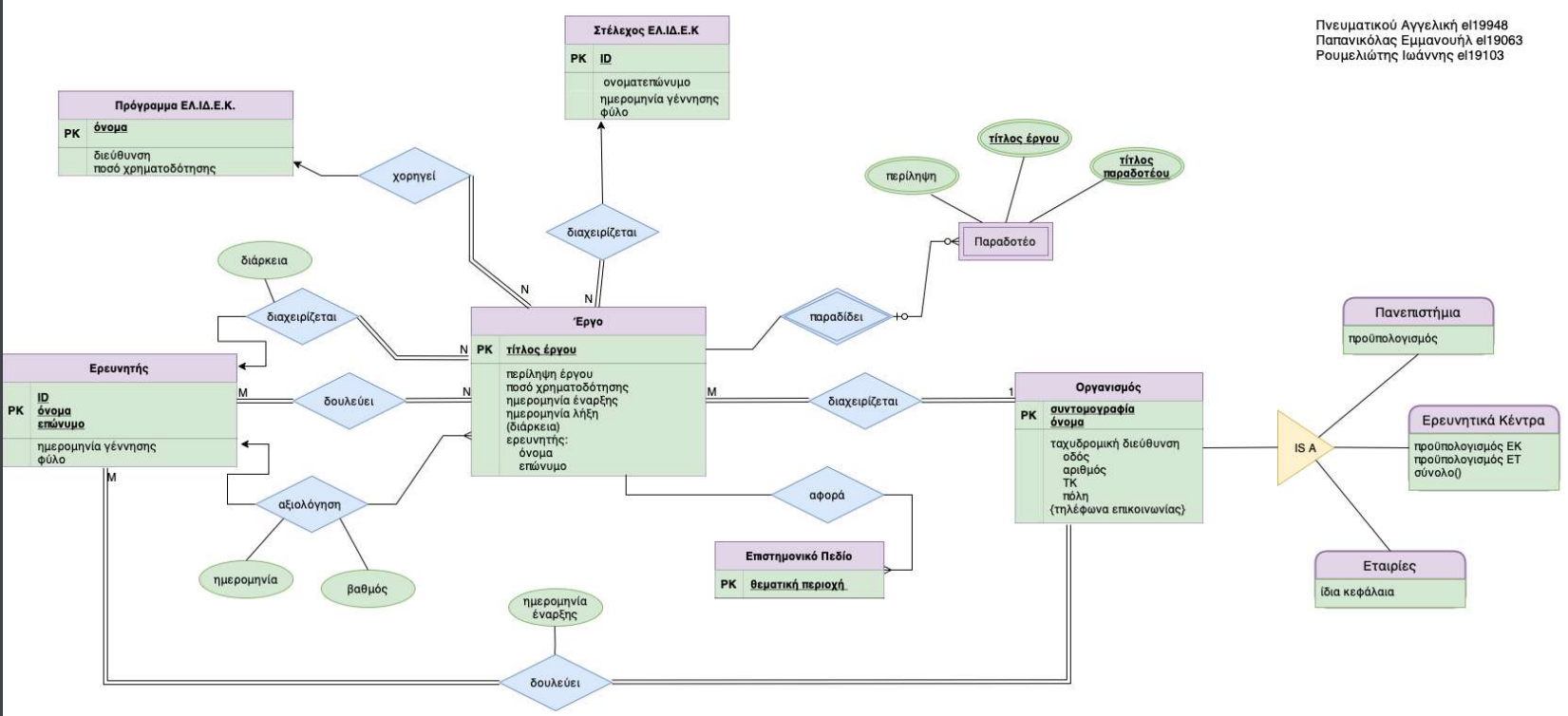
Β. Ευρετήρια / Indexes

Γ. Εναύσματα/ Triggers

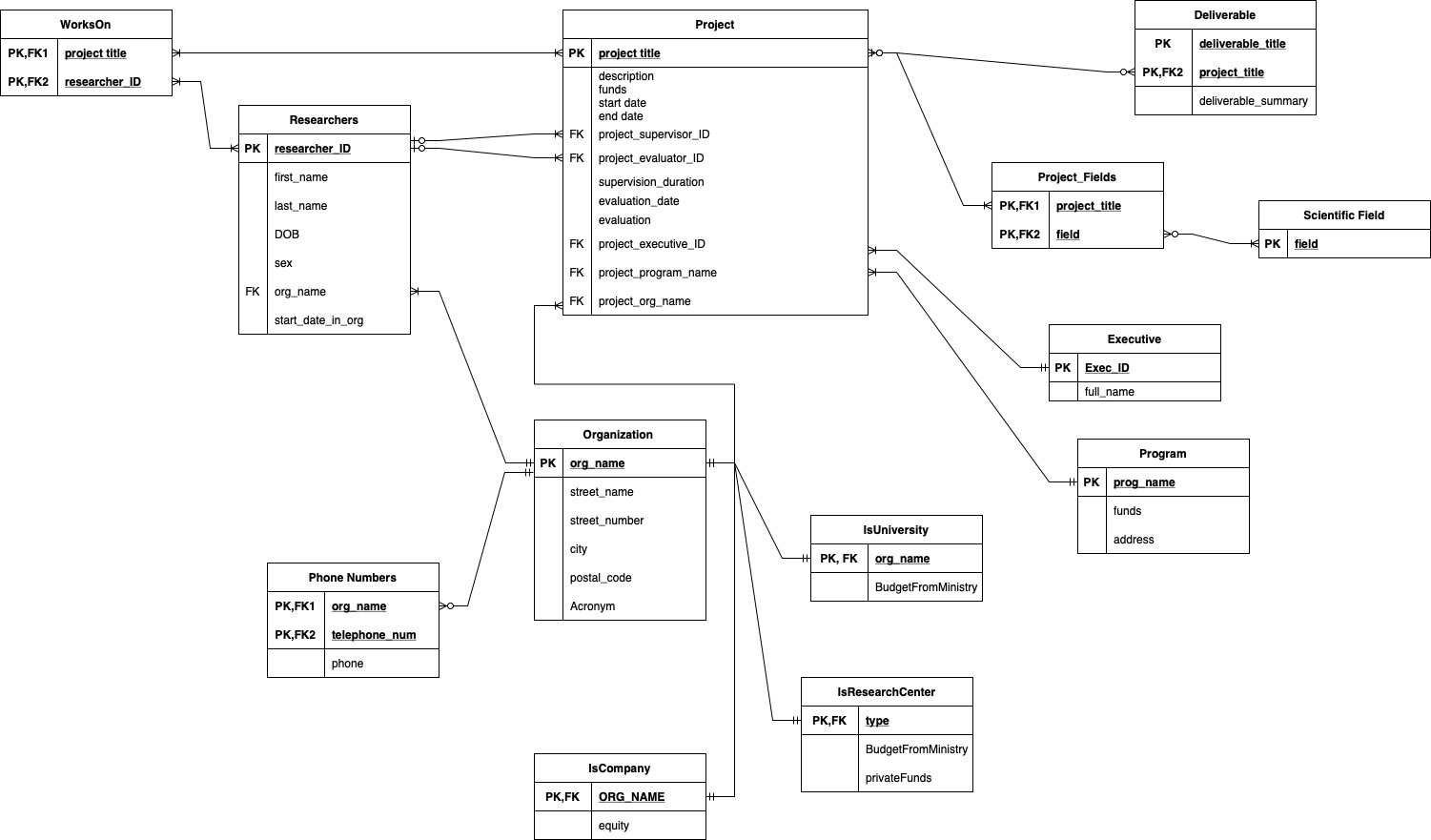
Δ. Αναλυτικά βήματα για την εγκατάσταση της εφαρμογής

1. Λίστα DDL βάσης
2. Κώδικας SQL (η υλοποίηση των ερωτημάτων μας)
3. Σχεσιακό Διάγραμμα της Βάσης Δεδομένων

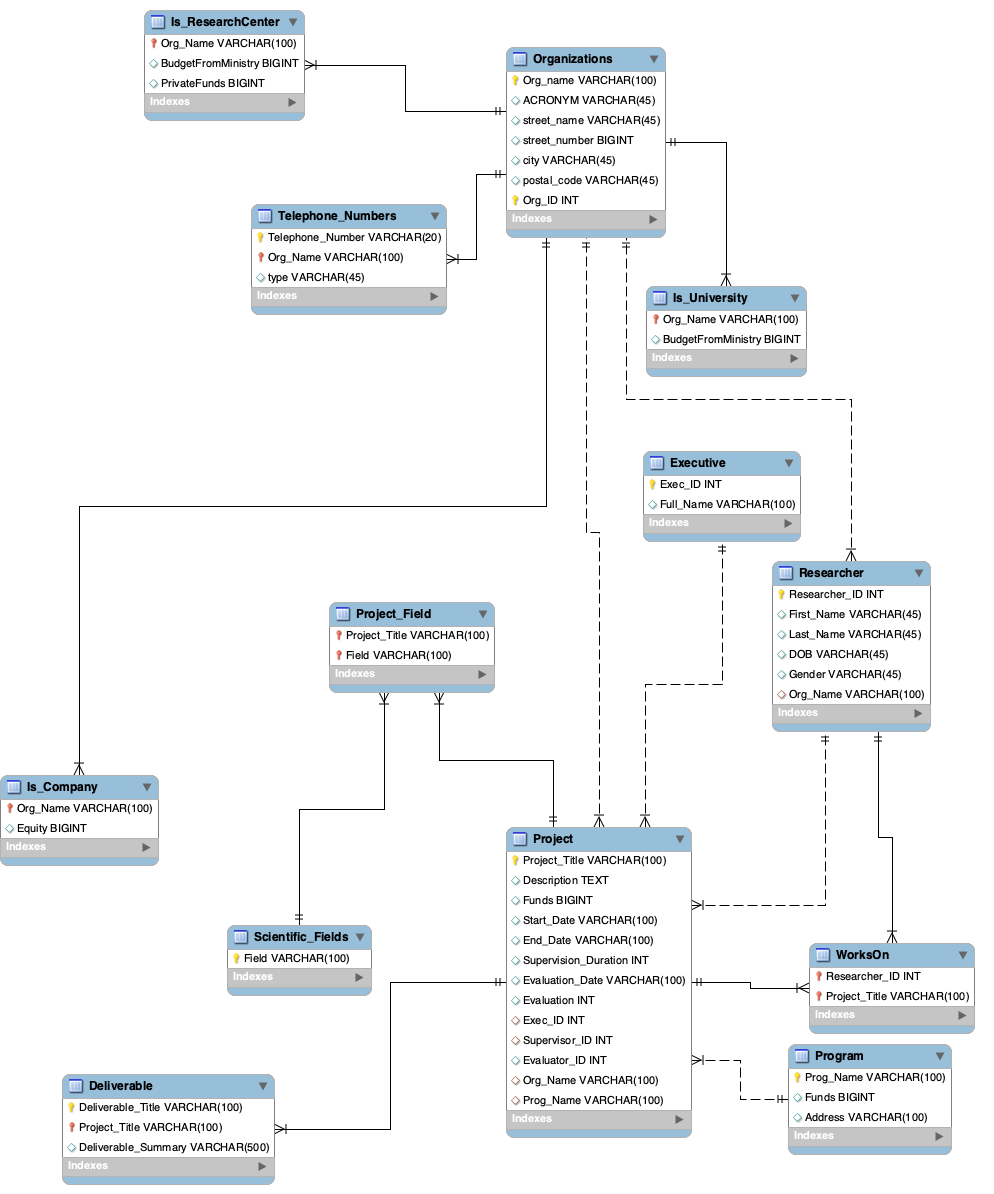
*ER diagram (το πρώτο μας παραδοτέο)*



*Relational Diagram1: το πρώτο μας RD πριν την υλοποίηση της βάσης μας σε sql*



*Relation Diagram: Το τελικό RD που προέκυψε μετά την υλοποίηση της βάσης μας σε περιβάλλον mySQL Workbench*

**

Α. Περιορισμοί και εξήγηση διαγραμμάτων:

*Ανάλυση κλειδιών σε κάθε πίνακα:*

* Deliverable: Το deliverable\_title είναι Primary Key, και αυτό συµβαίνει, διότι είναι το attribute εκείνο το οποίο είναι µοναδικό και ξεχωριστό για κάθε παραδοτέο.
* Executive: Το Exec\_ID είναι Primary Key, και αυτό συµβαίνει, διότι είναι το attribute εκείνο το οποίο είναι µοναδικό και ξεχωριστό για κάθε στέλεχος.
* Organisations: Το org\_name είναι το primary key του καθώς αυτό χαρακτιρίζει µοναδικά κάθε οργανισμό.  
  WorksOn: Ο πίνακας WorksOn είναι µία σχέση (πολλά προς πολλά) που ενώνει τις οντότητες Researchers και Project. Για αυτό το λόγο περιέχει ως primary/foreign keys, το project\_title από το Project και το researcher\_ID από το Researchers.
* Researcher: Στον πίνακα Researchers το researcher\_ID είναι το primary key του καθώς αυτό χαρακτιρίζει µοναδικά κάθε εγγραφή σε αυτόν τον πίνακα. Επίσης, στον πίνακα Researchers το org\_name είναι foreign key από τον πίνακα Organisations και αντιστοιχίζει τους δύο πίνακες σε σχέση πολλοί(Researchers) μπορούν να εργάζονται σε ένα(Organisation).
* Phone Numbers: Περιέχει ως primary/foreign keys, το org\_name από το Organization και το telephone\_num, καθώς πρόκειται για multivalued attribute του Organization και για αυτό χρειάζεται να είναι ξεχωριστό table.
* Project: Στον πίνακα Project το project\_title είναι το primary key του καθώς αυτό χαρακτιρίζει µοναδικά κάθε έργο. Επίσης, στον πίνακα Project το project\_supervisor\_ID είναι foreign key προς τον πίνακα Researchers και αντιστοιχίζει τους δύο πίνακες σε σχέση ένας επιστημονικός υπεύθυνος (Researchers) διαχειρίζεται πολλά (Project). Αντίστοιχα, το project\_evaluator\_ID είναι foreign key προς τον πίνακα Researchers και αντιστοιχίζει τους δύο πίνακες σε σχέση ένας (Researchers) αξιολογεί ένα έργο (Project). Το project\_executive\_ID είναι foreign key από τον πίνακα Executive και αντιστοιχίζει τους δύο πίνακες σε σχέση ένα στέλεχος (Executive) διαχειρίζεται πολλά (Project). Το project\_program\_name είναι foreign key από τον πίνακα Program και αντιστοιχίζει τους δύο πίνακες σε σχέση ένα (Program) χορηγεί πολλά (Project). Το project\_org\_name είναι foreign key από τον πίνακα Organization και αντιστοιχίζει τους δύο πίνακες σε σχέση ένας (Organization) διαχειρίζεται πολλά (Project).
* Program: Το prog\_name είναι Primary Key, και αυτό συµβαίνει, διότι είναι το attribute που ξεχωρίζει κάθε πρόγραμμα.  
  Scientific Field: Το field είναι Primary Key, και αυτό συµβαίνει, διότι είναι το attribute που ξεχωρίζει κάθε επιστημονικό πεδίο.
* Project\_Fields: Ο πίνακας Project\_Fields είναι µία σχέση (πολλά προς πολλά) που ενώνει τις οντότητες Scientific Field και Project. Για αυτό το λόγο περιέχει ως primary/foreign keys, το project\_title από το Project και το field από το Scientific Field.   
  Τα tables IsCompany, IsUniversity, IsResearcherCenter, δημιουργήθηκαν για τα lower-level entity set του specialization για το table “Organization” και για αυτό έχουν ως primary/foreign key το org\_name από το Organization. Τα τρία tables αντιστοιχίζονται με το Organization με τη σχέση IS\_A.

Β. Ευρετήρια/Indexes:

Έχουν οριστεί 10 ευρετήρια, τα οποία ορίστηκαν με γνώμονα την ελαχιστοποίηση του χρόνου που απαιτείται για την υλοποίηση των queries που μας δόθηκαν. Συγκεκριμένα φτιάχτηκαν ευρετήρια για τα ID που αποτελούσαν foreign keys (project\_evaluator\_ID, project\_executive\_ID, project\_supervisor\_ID, researcher\_ID), για άλλα συχνά χρησιμοποιούμενα foreign keys (όπως project title και org\_name) καθώς και για attributes που αποτελούσαν βασικά κριτήρια για queries (όπως το start date των project).

Αξιοποιήσαμε τα ευρετήρια με στόχο την ταχύτερη ανταπρόκριση στη βάση μας για τα ερωτήματα που χρησιμοποιήσαμε:

CREATE INDEX idx\_researcher\_id ON WorksOn(researcher\_ID);

Χρησιμοποιήσαμε αρκετές φορές τα ID των ερευνητών στα ερωτήματα, προκειμένου να εντοπίζουμε σχέσεις με τα έργα και τα υπόλοιπα στοιχεία της βάσης (συγκεκριμένα για τα ερωτήματα 3.3, 3.6, 3.8)

CREATE INDEX idx\_project\_supervisor\_id ON Project(supervisor\_ID);

Πολλές φορές χρειάστηκε να χρησιμοποιήσουμε το id των υπεύθυνων κάθε έργου για περιορισμούς τύπου, για παράδειγμα, στην εισαγωγή ενός νέου ερευνητή σε ένα έργο, θα πρέπει να ικανοποιείται η συνθήκη ότι δεν είναι και υπεύθυνος του συγκεκριμένου έργου.

CREATE INDEX idx\_project\_executive\_id ON Project(exec\_ID);

Για το ερώτημα 3.1, ο χρήστης θα μπορούσε να κάνει αναζήτηση έργων χρησιμοποιώντας τα ID των στελεχών προκειμένου να επιλέξει κάποιο έργο της επιθυμίας του. Επιπροσθέτως, ήταν αρκετά χρήσιμο για το ερώτημα 3.5

CREATE INDEX idx\_project\_org\_name ON Project(org\_name);

Αρκετές φορές, επίσης, συμπεριλάβαμε στα ερωτήματα το όνομα του οργανισμού που αντιστοιχεί σε κάθε έργο, για παράδειγμα στο 3.4

CREATE INDEX idx\_res\_org\_name ON Researcher(org\_name);

Αντίστοιχα με την παραπάνω περίπτωση, ωστόσο αξιοποιήθηκε σε άλλα ερωτήματα.

CREATE INDEX idx\_project\_title ON WorksOn(project\_title);

Εδώ, πάλι πολλαπλές φορές χρειάστηκε να αναζητήσουμε στη σχέση μεταξύ τα έργα και τους ερευνητές (workson table) προκειμένου να βρούμε κάποια σχέση, π.χ. στο ερώτημα 3.8

CREATE INDEX idx\_Field\_project\_title ON Project\_Field(project\_title);

Αξιοποιήθηκε σε δύο ερωτήματα, το 3.3 και 3.5

CREATE INDEX idx\_Deliverable\_project\_title ON Deliverable(project\_title);

Το συγκεκριμένο ευρετήριο μας φάνηκε χρήσιμο για το ερώτημα 3.8

CREATE INDEX idx\_start\_date ON Project(start\_date);

Το ευρετήριο αυτό ήταν αρκετά χρήσιμο καθώς στο ερώτημα 3.1 μας ζητήθηκε να μπορούμε να υλοποιήσουμε ένα ερώτημα τέτοιο ώστε ένας χρήστης να μπορεί να αναζητήσει ένα έργο βάσει την ημερομηνία έναρξής του.

Γ. Εναύσματα:

Συνολικά δημιουργήσαμε 6 εναύσματα. Ο λόγος που τα χρησιμοποιήσαμε ήταν διότι σε πολλές περιπτώσεις, όπως στην εισαγωγή κάποιου νέου ερευνητή, έπρεπε να εξασφαλίσουμε ότι ο ερευνητής που θα έμπαινε σε νέο έργο ανήκει στον ίδιο οργανισμό που έχει αναλάβει και το έργο αυτό και επίσης, να μην είναι ο υπεύθυνος του έργου αυτού. Αντίστοιχα, στην εισαγωγή κάποιου νέου οργανισμού, θα πρέπει να τον εντάξουμε και σε μία μόνο από τις τρείς κατηγορίες, Ερευνητικό Κέντρο, Πανεπιστήμιο, Εταιρία. Κατά συνέπεια, αν ο χρήστης προσπαθήσει (κατά λάθος) να τον εντάξει σε παραπάνω από μία κατηγορία, να μην μπορέσει. Τα εναύσματα που αφορούν τους ερευνητές και υπεύθυνους είναι τα #1, #2, #3 και τα υπόλοιπα τρία αφορούν τους οργανισμούς και τις κατηγορίες (πήραμε κάθε περίπτωση ξεχωριστά)

#1 Trigger

DELIMITER $$

create trigger insert\_in\_project

before insert on project

for each row

BEGIN

if (new.supervisor\_id not in (select w.researcher\_id from workson inner join project

as p on p.project\_title=w.project\_title))

THEN

insert into project set supervisor\_id = new.supervisor\_id;

end IF;

END $$

#2 Trigger

DELIMITER $$

create trigger insert\_in\_workson

before insert on workson

for each row

BEGIN

if (new.researcher\_id not in (select w.researcher\_id from workson inner join project

as p on p.supervisor\_id=w.researcher\_id))

THEN

insert into workson set researcher\_id = new.researcher\_id;

end IF;

END $$

#3 Trigger

DELIMITER $$

create trigger insert\_in\_project\_2

before insert on project

for each row

BEGIN

if (new.supervisor\_id in (select r.researcher\_id from researcher inner join project

as p on p.org\_name=r.org\_name))

THEN

insert into project set supervisor\_id = new.supervisor\_id;

end IF;

END $$

#4 Trigger

DELIMITER $$

create trigger insert\_in\_university

before insert on is\_university

for each row

BEGIN

if (new.org\_name not in ((select org\_name from is\_company) AND (select org\_name from is\_researchcenter)))

THEN

insert into is\_university set org\_name = new.org\_name;

end IF;

END $$

#5 Trigger

DELIMITER $$

create trigger insert\_in\_company

before insert on is\_company

for each row

BEGIN

if (new.org\_name not in ((select org\_name from is\_university) AND (select org\_name from is\_researchcenter)))

THEN

insert into is\_university set org\_name = new.org\_name;

end IF;

END $$

#6 Trigger

DELIMITER $$

create trigger insert\_in\_researchcenter

before insert on is\_researchcenter

for each row

BEGIN

if (new.org\_name not in ((select org\_name from is\_university) AND (select org\_name from is\_company)))

THEN

insert into is\_university set org\_name = new.org\_name;

end IF;

END $$

Δ. Οδηγίες Εγκατάστασης σε Mac

1. Απαιτείται η εγκατάσταση του MySQL Server.

2. Αφού έχει γίνει η εγκατάσταση συνδεόμαστε στον server μέσω μιας DBMS και προτείνουμε την χρήση του MySQL Workbench. Συνδεόμαστε με SQL Server Authentication, με Server Name το όνομα του υπολογιστή και κάνουμε login με username, για να έχουμε όλα τα δικαιώματα, και τον κωδικό που επιθυμούμε.

3. Αφού συνδεθούμε επιτυχώς, μπορούμε να πάμε να δημιουργήσουμε όλους τους πίνακες και τις απαραίτητες συνδέσεις μεταξύ τους. Στην εργασία έχουμε επισυνάψει τα SQL Scripts που περιέχουν τον κώδικα για την δημιουργία των παραπάνω.

4. Στην συνέχεια, βάζουμε τα δεδομένα στην βάση. Τα δεδομένα μας αρχικά τα βάλαμε σε ένα αρχείο .xlsx. Χρησιμοποιούμε το Import / Export wizard του Workbench και τοποθετούμε τους πίνακες.

5. Αφού έχουν οριστεί οι πίνακες, έχει γίνει εισαγωγή των δεδομένων, έχουν δημιουργηθεί τα απαραίτητα ευρετήρια και τα views, προχωράμε στο στήσιμο του περιβάλλοντος πάνω στο οποίο τρέχει η εφαρμογή μας.

Τα αρχεία που χρειάζεται να αντιγράψετε βρίσκονται στο “FILE\_PATH” και έχουν την ακόλουθη δομή:

* controllers: the logic of the routes (where the sql queries take place, and then the result can be either returned as a rendered page, or in other formats such as JSON, csv etc.)
* public: all publicly accessible files to which a user of the system can have access (usually css, icons, static pages etc…)
* routes: the end points of the application.
* utils: can contain any utilities you want for your project, for example the connection of the database, so that you can just import and use it when needed (mostly in controllers).
* views: the pages of the project

6. Εγκαθιστούμε τις απαραίτητες βιβλιοθήκες:

* express: Minimal and flexible Node.js web application framework that provides robust set of features to develop web and mobile apps.
* mysql2: MySQL driver – providing connectivity for the client.
* ejs: Templating engine.
* nodemon: A cli utility that wraps the Node app, watches the file system and automatically restarts the process on changes (development).

Η σειρά των εντολών είναι η ακόλουθη :

brew install node

npm init

npm install express

ejs mysql2 custom-env

npm install nodemon

npm start

7. Ύστερα ανοίγουμε τα αρχεία που επισυνάπτονται σε κάποιον editor και αντικαθιστούμε στο αρχείο database.js, τα credentials που ορίσαμε στο βήμα 2. Πιο συγκεκριμένα:

host: process.env.DB\_HOST,

port: process.env.DB\_PORT,

user: process.env.DB\_USER,

password: process.env.DB\_PASS,

database: process.env.DB,

και σε ένα άλλο αρχείο με όνομα .env.localhost ορίσαμε τις μεταβλητές που αντιστοιχούν παραπάνω, συμπληρώνοντας στο SERVER\_HOST: 3000, DB\_HOST: localhost, DB\_PORT = 3306, DB\_USER = root, DB\_PASS=(εισάγετε τον δικό σας κωδικό, αν υπάρχει, εάν δεν υπάρχει αφήστε το κενό)

8. Αφού τοποθετήσουμε, όλα τα αρχεία και σιγουρευτούμε ότι είναι στους σωστούς φακέλους, μπορούμε να τρέξουμε την εφαρμογή μας σε κάποιον localhost. Για να μπορούμε επιτυχώς να το κάνουμε αυτό, πρέπει να βρισκόμαστε στον ίδιο φάκελο που είναι και το αρχείο database.js και αρχείο app.js και εκτελούμε τις εντολές:

Αξιοποιώντας το Visual Studio Code για editor, ανοίγουμε τον φάκελο που περιέχει τους υπόλοιπους φακέλους που περιέχουν και την σύνδεση με τη βάση μας, ανοίγουμε το terminal και τρέχουμε την ακόλουθη εντολή:

npm start

*\*για τους χρήστες των Windows, η εγκατάσταση του Node μπορεί να γίνει και manually, μέσω του επίσημου site* [*https://www.npmjs.com*](https://www.npmjs.com) *όπου εγκαθίστανται τα απαραίτητα πακέτα των nodes js και npm αλλά και από το CLI του terminal σας (θα υπάρχουν αναλυτικότερες οδηγίες στο readme που έχουμε αναρτήσει στο github*

1. Λίστα DDL της βάσης

Παρακάτω παρατίθενται οι πίνακες που υλοποιήσαμε στη βάση μας σε μορφή SQL code

-- Table structure for table `Deliverable`

--

CREATE TABLE `Deliverable` (

`Deliverable\_Title` varchar(200) NOT NULL,

‘Project\_Title` varchar(100) NOT NULL,

`Deliverable\_Summary` text DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `executive`

--

CREATE TABLE `executive` (

`exec\_id` int(100) NOT NULL,

`full\_name` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `Is\_Company`

--

CREATE TABLE `Is\_Company` (

`Org\_Name` varchar(100) NOT NULL,

`Equity` bigint(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `Is\_ResearchCenter`

--

CREATE TABLE `Is\_ResearchCenter` (

`Org\_Name` varchar(100) NOT NULL,

`BudgetFromMinistry` bigint(255) DEFAULT NULL,

`PrivateFunds` bigint(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `Is\_University`

--

CREATE TABLE `Is\_University` (

`Org\_Name` varchar(100) NOT NULL,

`BudgetFromMinistry` bigint(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `organizations`

--

CREATE TABLE `organizations` (

`org\_name` varchar(250) NOT NULL,

`acronym` varchar(50) NOT NULL,

`street\_name` varchar(50) NOT NULL,

`street\_number` varchar(100) NOT NULL,

`city` varchar(50) NOT NULL,

`postal\_code` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `program`

--

CREATE TABLE `program` (

`prog\_name` varchar(200) NOT NULL,

`funds` varchar(250) NOT NULL,

`address` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `project`

--

CREATE TABLE `project` (

`project\_title` varchar(200) NOT NULL,

`description` text NOT NULL,

`funds` varchar(100) NOT NULL,

`start\_date` date NOT NULL,

`end\_date` date NOT NULL,

`supervision\_duration` int(5) NOT NULL,

`evaluation\_date` date NOT NULL,

`evaluation` int(11) NOT NULL,

`exec\_id` int(11) NOT NULL,

`supervisor\_id` int(100) NOT NULL,

`evaluator\_id` int(100) NOT NULL,

`org\_name` varchar(200) NOT NULL,

`prog\_name` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Stand-in structure for view `projects\_per\_organizations\_and\_per\_programs`

-- (See below for the actual view)

--

CREATE TABLE `projects\_per\_organizations\_and\_per\_programs` (

`organization` varchar(250),

`program` varchar(200),

`project\_title` varchar(200)

);

-- --------------------------------------------------------

--

-- Table structure for table `Project\_Field`

--

CREATE TABLE `Project\_Field` (

`Project\_Title` varchar(100) NOT NULL,

`Field` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `Researcher`

--

CREATE TABLE `Researcher` (

`Researcher\_ID` int(11) NOT NULL,

`First\_Name` varchar(45) DEFAULT NULL,

`Last\_Name` varchar(45) DEFAULT NULL,

`DOB` varchar(45) DEFAULT NULL,

`Gender` varchar(45) DEFAULT NULL,

`Org\_Name` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `Scientific\_Fields`

--

CREATE TABLE `Scientific\_Fields` (

`field` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `Telephone\_Numbers`

--

CREATE TABLE `Telephone\_Numbers` (

`Telephone\_Number` varchar(20) NOT NULL,

`Org\_Name` varchar(100) NOT NULL,

`type` varchar(45) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Table structure for table `WorksOn`

--

CREATE TABLE `WorksOn` (

`Researcher\_ID` int(11) NOT NULL,

`Project\_Title` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

1. Κώδικας SQL (υλοποίηση ερωτημάτων)

Δημιουργήσαμε, επίσης και όψεις (views) με στόχο να ευκολότερη υλοποίηση των ερωτημάτων που θα αναφέρουμε στη συνέχεια.

Ερώτημα (3.1)

#query 3.1:

select prog\_name from program; (όλα τα διαθέσιμα προγράμματα)

Δημιουργία όψης με όνομα projectaki:

create view projectaki as (select distinct concat(r.first\_name, " ", r.last\_name) as researcher\_name,

p.project\_title,

e.full\_name as executive\_name,

p.start\_date as date, (YEAR(p.end\_date)-YEAR(p.start\_date)) as duration,

p.funds as funds

from project as p

inner join workson as w on w.project\_title = p.project\_title

inner join researcher as r on r.researcher\_id = w.researcher\_id

inner join executive as e on e.exec\_id = p.exec\_id);

και η υλοποίηση του ερωτήματος μας:

select project\_title, executive\_name, date, funds, researcher\_name from projectaki where

date = "{input}" and

duration = "{input}" and

executive\_name = "{input}";

Ερώτημα (3.2)

#querie 3.2:

Η ζητούμενη όψη (έργα ανά ερευνητή)

create view projects\_per\_researchers as (select concat(r.first\_name, ' ', r.last\_name) as researcher\_name,

concat(r2.first\_name, ' ', r2.last\_name) as supervisor\_name,

p1.project\_title as project\_title from researcher as r

inner join workson as w on w.researcher\_id = r.researcher\_id

inner join project as p1 on p1.project\_title = w.project\_title

inner join project as p2 on p2.project\_title=w.project\_title

inner join researcher as r2 on r2.researcher\_id = p2.supervisor\_id);

Η όψη της επιλογής μας (έργα και προγράμματα ανά οργανισμό)

create view projects\_per\_organizations\_and\_per\_programs as (select

org.org\_name as organization,

prog.prog\_name as program,

p.project\_title as project from organizations as org

inner join project as p on p.org\_name = org.org\_name

inner join project as p1 on p1.project\_title = p.project\_title

inner join program as prog on prog.prog\_name = p1.prog\_name) order by organization;

drop view projects\_per\_organizations\_and\_per\_programs;

select \* from projects\_per\_organizations\_and\_per\_programs;

Ερώτημα (3.3)

#query 3

select distinct concat(r.first\_name," ", r.last\_name) as full\_name, workson.project\_title from workson

inner join researcher as r on r.researcher\_id = workson.researcher\_id

inner join project\_field on project\_field.project\_title = workson.project\_title

inner join project on project.end\_date and project.start\_date

where (project\_field.field = "{ input}" and project.end\_date > "2022-01-01" and project.start\_date < "2022-05-20");

Ερώτημα (3.4)

#query 3.4:

SELECT DISTINCT r3a.org\_name FROM(

(SELECT \* FROM(SELECT r1a.org\_name, r1a.year1, COUNT(r1a.project\_title) AS cc FROM

(SELECT org1.org\_name,

p1.project\_title,

YEAR(p1.start\_date) AS year1

FROM Organizations org1

INNER JOIN Project p1 ON org1.org\_name = p1.org\_name) r1a

GROUP BY r1a.org\_name, r1a.year1) r2a

WHERE r2a.cc > 9) r3a

INNER JOIN(SELECT \* FROM(SELECT r1b.org\_name, r1b.year2, COUNT(r1b.project\_title) AS cc

FROM(SELECT org1.org\_name, p1.project\_title, YEAR(p1.start\_date) AS year2

FROM Organizations org1 INNER JOIN Project p1 ON org1.org\_name = p1.org\_name) r1b

GROUP BY r1b.org\_name, r1b.year2) r2b

WHERE r2b.cc > 9) r3b

ON r3a.org\_name = r3b.org\_name)

WHERE ABS(r3a.year1 - r3b.year2) = 1 AND r3a.cc = r3b.cc;

Ερώτημα (3.5)

#query 3.5:

Δημιουργία όψης που να δημιουργεί έναν πίνακα με τους τίτλους έργων και οι συνδυασμοί των πεδίων που αντιστοιχούν σε αυτό

CREATE VIEW PAIR\_OF\_FIELDS AS

select

a.project\_title,

a.field, b.field as field1 from project\_field a,

project\_field b

where a.project\_title = b.project\_title and a.field < b.field;

η υλοποίηση του ερωτήματος

select

field,

field1,

count(\*) as c from pair\_of\_fields group by field, field1 order by count(\*) desc limit 3;

Ερώτημα (3.6)

#query 3.6:

select concat(researcher.last\_name, ' ', researcher.first\_name) as full\_name,count(workson.project\_title from researcher

inner join workson on workson.researcher\_id = researcher.researcher\_id

inner join project on project.project\_title = workson.project\_title

where researcher.dob > "1982-06-29" and project.end\_date > "2022-06-12"

group by full\_name

order by count(workson.project\_title) desc;

Ερώτημα (3.7)

#query 3.7

select distinct executive.full\_name as full\_name, company.org\_name as org\_name, company.equity as equity from executive

inner join project on project.exec\_id = executive.exec\_id

inner join is\_company as company on company.org\_name = project.org\_name

order by equity desc

limit 5;

Ερώτημα (3.8)

#query 3.8

select concat(researcher.last\_name, ' ', researcher.first\_name) as full\_name, count(workson.project\_title) as NumOfProjects from researcher

inner join workson on workson.researcher\_id = researcher.researcher\_id

where workson.project\_title not in (select project\_title from deliverable) group by full\_name

having NumOfProjects>5 order by NumOfProjects desc;

*\*θεωρήσαμε ότι τα ενεργά έργα είναι εκείνα τα οποία δεν έχουν ολοκληρωθεί αυτήν την περίοδο (δηλαδή μετά της 1η Μαϊου2022)*