ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ

ΟΜΑΔΑ PROJECT 29
ΔΑΜΙΑΝΟΣ ΔΗΜΗΤΡΙΟΣ (03119825)
ΚΑΠΕΤΑΝΑΚΗΣ ΑΝΑΣΤΑΣΙΟΣ (03119048)
ΜΠΑΡΗΣ ΓΕΩΡΓΙΟΣ (03119866)

2022

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή

Στο φετινό πρότζεκτ του μαθήματος Βάσεις Δεδομένων μας ζητήθηκε να σχεδιάσουμε και να υλοποιήσουμε ένα σύστημα αποθήκευσης , διαχείρισης και ανάλυσης των πληροφοριών του Ελληνικού Ιδρύματος Έρευνας και Καινοτομίας ΕΛ.ΙΔ.Ε.Κ. . Στο πλαίσιο αυτό δημιουργήσαμε τόσο την Βάση Δεδομένων για την συλλογή και διαχείριση των δεδομένων του ιδρύματος καθώς και αναπτύξαμε το κατάλληλο User Interface ώστε ο χρήστης να μπορεί να δει , εισάγει και επεξεργαστεί τα δεδομένα της βάσεις με ιδιαίτερη ευκολία . Η βάση δεδομένων δημιουργήθηκε με χρήση γλώσσας SQL ενώ το UI (frond-end , Back-end) δημιουργήθηκε με χρήση HTML, CSS, JAVASCRIPT και PHP.

Διάγραμμα ER(Entity - Relation)

Η κατασκευή του διαγράμματος ER βασίζεται στο στόχο που έχει σκοπό να υπηρετήσει η βάση δεδομένων που επιθυμεί ο πελάτης(σε αυτή την περίπτωση η εκφώνηση της εργασίας). Τα βήματα που ακολουθήθηκαν προκειμένου να γίνει αυτό είναι τα παρακάτω :

1. Ανάγνωση και ουσιαστική κατανόηση του σκοπού της βάσης.
2. Αναγνώριση των βασικών οντοτήτων της βάσης δεδομένων και των γνωρισμάτων (attributes) που τις συνοδεύουν(**Entities**).
3. Αναγνώριση των συσχετίσεων και των γνωρισμάτων τους (attributes) που συνδέουν τις οντότητες που βρέθηκαν στο προηγούμενο βήμα(**Relations**).

Στη συνέχεια αναλύουμε τον κορμό της παραπάνω διαδικασίας, που είναι η εύρεση των οντοτήτων και των συσχετίσεων.

Εύρεση Οντοτήτων

Οι ισχυρές οντότητες που μπορούμε να αναγνωρίσουμε παρατίθενται παρακάτω :

- Έργα/Επιχορηγήσεις: Κάθε έργο έχει τα ακόλουθα γνωρίσματα
  - Τίτλος
  - Ημερομηνία Λήξης
  - Ημερομηνία Έναρξης
  - Περίληψη
  - Ποσό επιχορήγησης
  - Διάρκεια (ποσό που προκύπτει από την ημερομηνία έναρξης και λήξης): Πρέπει να παίρνει τιμές μεταξύ 1 και 4
- Οργανισμοί: Τα γνωρίσματα που προκύπτουν είναι τα :
  - Συντομογραφία
  - Ταχυδρομική Διεύθυνση
    - Οδός
    - Αριθμός
    - Τ.Κ.
    - Πόλη
  - Τηλέφωνα επικοινωνίας(όχι απαραίτητα ένα τηλέφωνο επικοινωνίας)
  - Κατηγορία όπου ανήκει: Η κατηγορία παίρνει τιμές μεταξύ των ακόλουθων :
    - Εταιρεία
    - Πανεπιστήμιο
    - Ερευνητικό Κέντρο

Εδώ πρέπει να τονιστεί πως χειριζόμαστε αυτές τις τιμές σαν διαφορετικές οντότητες, διότι η κάθε μία θα πρέπει να περιέχει και τα αντίστοιχα κεφάλαιά της.

- Ερευνητές: Ο κάθε ερευνητής πρέπει να διαθέτει :
  - Όνομα
  - Επώνυμο
  - Φύλο
  - Ημερομηνία γέννησης
- Προγράμματα Επιχορήγησης: Διακρίνουμε τα ακόλουθα γνωρίσματα :
  - Όνομα επιχορήγησης
  - Υπεύθυνη διεύθυνση ΕΛ.ΙΔ.Ε.Κ.
- Επιστημονικά πεδία:
  - Όνομα επιστημονικού πεδίου
- Στελέχη:
  - Αριθμός μητρώου στελέχους

Ταυτόχρονα ορίζονται υποκατηγορίες των οργανισμών τις οποίες πρόκειται να τοποθετήσουμε σε ISA :

- Εταιρεία:
  - Ίδια κεφάλαια
- Πανεπιστήμιο:
  - Προϋπολογισμός από Υπουργείο Παιδείας
- Ερευνητικό Κέντρο:
  - Προϋπολογισμός από Υπουργείο Παιδείας
  - Προϋπολογισμός από Ιδιωτικές Δράσεις

Τέλος, θα πρέπει να οριστούν οι ακόλουθες ασθενείς οντότητες(weak entities) :

- Παραδοτέο: Η αντίστοιχη ισχυρή οντότητα είναι τα έργα/επιχορηγήσεις και θα πρέπει να περιλαμβάνει ως επιπλέον γνωρίσματα τα
  - Τίτλος
  - Περίληψη
- Αξιολόγηση: Η αντίστοιχη ισχυρή οντότητα είναι πάλι τα έργα/επιχορηγήσεις και επιπλέον θα πρέπει να περιλαμβάνονται τα εξής γνωρίσματα:
  - Βαθμός
  - Ημερομηνία Αξιολόγησης

Εύρεση Συσχετίσεων

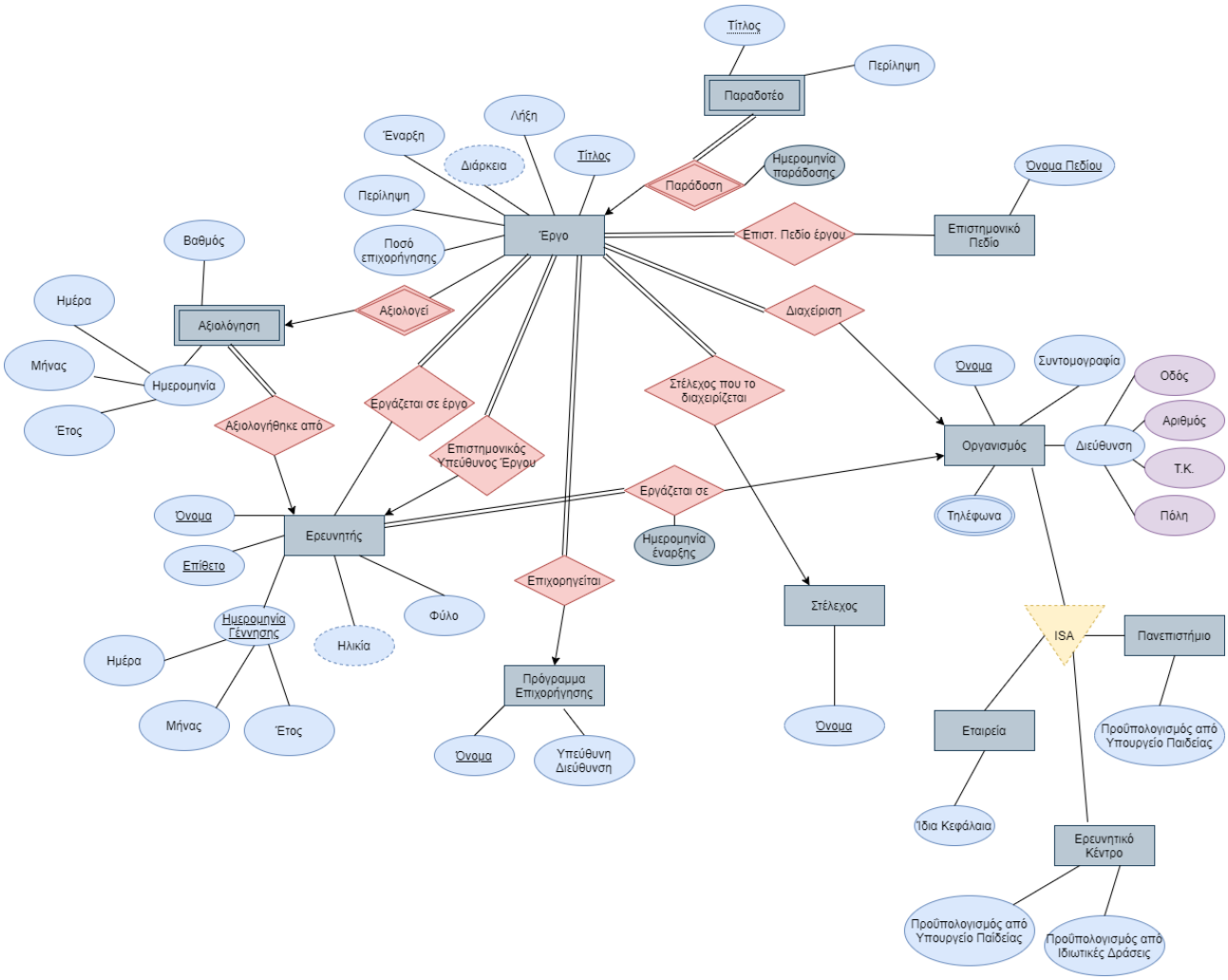
Αφού έγινε η αναγνώριση των οντοτήτων, βρισκόμαστε πλέον σε θέση να βρούμε τις σχέσεις που συνδέουν κάποιες από αυτές τις οντότητες μεταξύ τους.

- Έργο διαχειριζόμενο από οργανισμό: Συνδέει κάθε έργο με ένα ακριβώς οργανισμό που το διαχειρίζεται.
- Επιστημονικός υπεύθυνος σε έργο: Συνδέει κάθε έργο με έναν ακριβώς επιστημονικό υπεύθυνο.
- Επιστημονικό πεδίο έργου: Συνδέει κάθε έργο με ένα επιστημονικό πεδίο.
- Παραδοτέο στο έργο: Συνδέει το κάθε έργο με ένα ή περισσότερα παραδοτέα.
- Αξιολόγηση στο έργο: Συνδέει μία αξιολόγηση με ένα έργο.
- Αξιολόγηση από ερευνητή: Συνδέει έναν ερευνητή με μία αξιολόγηση.
- Ερευνητής εργαζόμενος σε έργο: Συνδέει έναν ερευνητή με ένα έργο στο οποίο και εργάζεται.
- Ερευνητής εργαζόμενος σε οργανισμό: Συνδέει έναν ερευνητή με ένα οργανισμό στον οποίο και εργάζεται. Συμπεριλαμβάνει ένα γνώρισμα :
  - Ημερομηνία έναρξης
- Πρόγραμμα επιχορήγησης επιχορηγεί έργο: Συνδέει κάθε έργο με ένα ακριβώς πρόγραμμα.

- Στέλεχος που διαχειρίζεται το έργο: Συνδέει το κάθε έργο με ένα ακριβώς στέλεχος.
- Παραδοτέο σε έργο: Συνδέει την ασθενή οντότητα "Παραδοτέο" με την ισχυρή οντότητα "Έργο". Περιέχει το ακόλουθο γνώρισμα :
  - Ημερομηνία παράδοσης

Τελική μορφή ER διαγράμματος

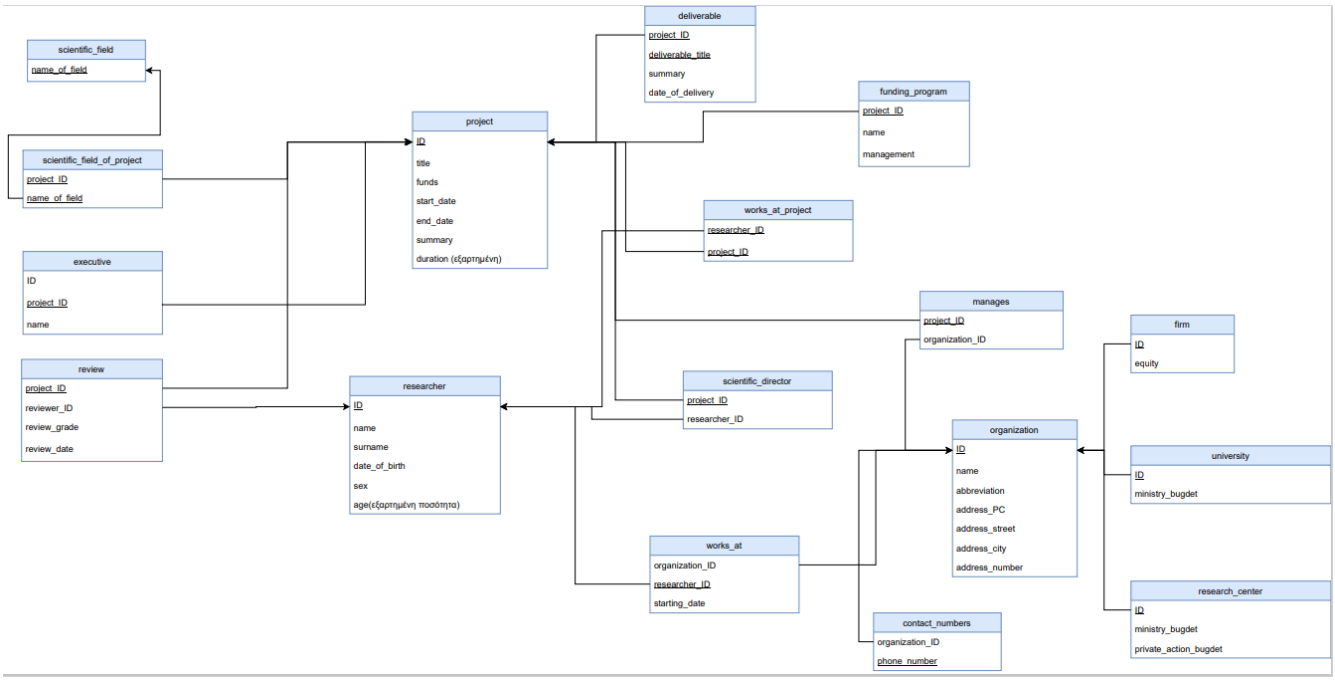
Λαμβάνουμε υπόψη όλα τα συμπεράσματα που εξαγάγαμε από την διαδικασία της κατανόησης του σκοπού της βάσης δεδομένων και προκύπτει το ER διάγραμμα της ακόλουθης εικόνας :



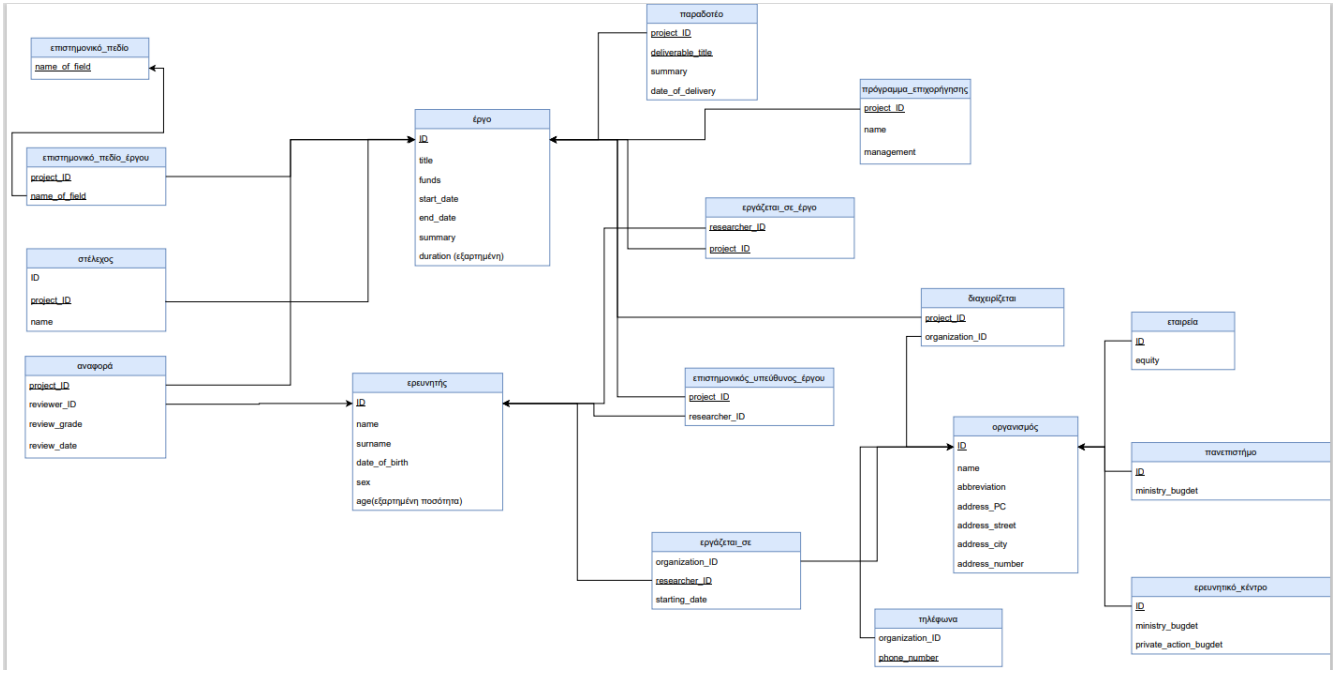
Σχεσιακό Σχήμα (Relational Schema)

Το σχεσιακό σχήμα της βάσης δεδομένων μας είναι το παρακάτω :

Η αγγλική έκδοση:



Και η ελληνική έκδοση:



Οι οντότητες των οποίων τα γνωρίσματα τους είναι τα ίδια με το ER διάγραμμα είναι οι εξής :

- ‘Έργο/επιχορήγηση’
- ‘Ερευνητής’
- ‘Οργανισμός’
- ‘Επιστημονικό πεδίο’
- ‘Εταιρεία’
- ‘Πανεπιστήμιο’
- ‘Ερευνητικό Κέντρο’

Η μόνη διαφορά εντοπίζεται στην προσθήκη των ID’s, με την χρήση των οποίων ως primary keys καταφέραμε να έχουμε ευκολότερη πρόσβαση στα δεδομένα μας.

Οι οντότητες :

- ‘Αναφορά’
- ‘Στέλεχος’
- ‘Πρόγραμμα Επιχορήγησης’
- ‘Παραδοτέο’

έχουν επιπλέον γνωρίσματα από όσα ορίζονται στο ER διάγραμμα, για λόγους που θα αναλυθούν παρακάτω.

Οι σχέσεις :

- ‘Εργάζεται σε έργο’
- ‘Εργάζεται σε’
- ‘Επιστημονικός υπεύθυνος έργου’
- ‘Διαχειρίζεται’
- ‘Επιστημονικό πεδίο έργου’

έχουν την ίδια μορφή με το ER διάγραμμα και φέρουν τα αναγκαία γνωρίσματα για να υλοποιούν τις σχέσεις που έχουν οριστεί να κάνουν.

Αντίθετα, οι σχέσεις :

- ‘Αξιολογεί’
- ‘Παράδοση’
- ‘Στέλεχος που το διαχειρίζεται’
- ‘Αξιολογήθηκε από’
- ‘Επιχορηγείται’

απουσιάζουν από το σχεσιακό διάγραμμα, διότι, για λόγους βελτιστοποίησης της χρήσης του χώρου μνήμης και του χρόνου εκτέλεσης, η πληροφορία που έφεραν ενσωματώθηκε σε άλλους πίνακες-σχέσεις.

Συγκεκριμένα :

- ‘Αξιολογεί’ και ‘Αξιολογήθηκε από’:

Ενσωματώθηκαν στην σχέση ‘Αξιολόγηση’, όπου το ποιος αξιολογεί βρίσκεται στο reviewer_ID, ενώ το ποιο (έργο) αξιολογείται βρίσκεται στο project_ID.

- ‘Στέλεχος που διαχειρίζεται’:

Ενσωματώθηκε στην σχέση ‘Στέλεχος’, όπου το ποιος έργο διαχειρίζεται το στέλεχος βρίσκεται στο project_ID και ποιο στέλεχος διαχειρίζεται το έργο στο ID.

- ‘Παράδοση’:

Ενσωματώθηκε στη σχέση ‘Παραδοτέο’, όπου το έργο στο οποίο αντιστοιχεί το παραδοτέο βρίσκεται στο project_ID.

- ‘Επιχορηγείται’:

Ενσωματώθηκε στην σχέση ‘Πρόγραμμα_Επιχορήγησης’, όπου το έργο που επιχορηγείται βρίσκεται στο project_ID.

Τέλος, το γνώρισμα ‘Τηλέφωνα’ του ‘Οργανισμού’ υλοποιείται σε ξεχωριστό πίνακα-σχέση, αφού φέρει πολλές τιμές (multi-valued attribute).

DDL SCRIPTS

Database Creation

Παραθέτουμε τις εντολές για την δημιουργία της βάσης μας, μαζί με τους απαραίτητους περιορισμούς , κλειδιά και περιορισμούς ξένου κλειδιού.

```
create table if not exists project(
    ID int not null,
    title varchar(1000) not null,
    funds real,
    start_date varchar(50) not null,
    end_date varchar(50) not null,
    summary varchar(500) not null,
    duration real,
    primary key (ID));

create table if not exists review(
    project_ID int not NULL,
    reviewer_ID int not null,
    review_grade real,
    review_date date,
    primary key(project_ID));


create table if not exists deliverable(
    project_ID int not null,
    deliverable_title varchar(500) not null,
    summary varchar(10000),
    date_of_delivery date,
    primary key(project_ID, deliverable_title));

create table if not exists scientific_field(
    name_of_field varchar(50) not null,
    primary key(name_of_field));

create table if not exists scientific_field_of_project(
    project_ID int not null,
    name_of_field varchar(50) not null,
    primary key(project_ID, name_of_field));

create table if not exists executive(
    ID int not null,
    project_ID int not null,
    name varchar(20) not null,
    primary key(project_ID));
```

```sql
create table if not exists researcher(
    ID int not NULL,
    name varchar(50) not NULL,
    surname varchar(50) not null,
    date_of_birth date not null,
    sex varchar(10) not null ,
    age int,
    primary key(ID),
    check (sex in ('Male','Female')));

create table if not exists scientific_director(
    project_ID int not NULL,
    researcher_ID int not NULL,
    primary key(project_ID));

create table if not exists works_at(
    organization_ID int not null,
    researcher_ID int not null,
    starting_date date not null,
    primary key(researcher_ID));

create table if not exists organization(
    ID int not null,
    name varchar(50) not null,
    abbreviation varchar(20)not null,
    address_PC int not null,
    address_street varchar(50) not null,
    address_city varchar(50) not null,
    address_number int not null,
    primary key(ID));

create table if not exists manages(
    project_ID int not null,
    organization_ID int not null,
    primary key(project_ID));

create table if not exists firm(
    ID int not null,
    equity real not null,
    primary key(ID));

create table if not exists university(
    ID int not null,
    ministry_budget real ,
    primary key(ID));

create table if not exists research_center(
    ID int not null,
    ministry_budget real,
    private_action_budget real,
    primary key(ID));
```

```
create table if not exists contact_numbers(
    organization_ID int not null,
    phone_number varchar(20) not null,
    primary key(phone_number));

create table if not exists funding_program(
    project_ID int not null,
    name varchar(50) not null,
    management varchar(50),
    primary key(project_ID));

create table if not exists works_at_project(
    researcher_ID int not null,
    project_ID int not null,
    primary key(project_ID,researcher_ID));
```

Θέτουμε ως περιορισμό τα ID (ως primary keys) να είναι μην είναι NULL (**not null constraint**) , καθώς και κάποια σημαντικά γνωρίσματα που χρησιμοποιούμε στα queries μας, όπως name, surname, date_of_birth κτλ., ενώ περιορίζουμε το γνώρισμα sex σε μόνο 2 τιμές (male ή female) χρησιμοποιώντας το constraint **check (sex in ('Male','Female'))**.

Foreign Keys

Παράλληλα θέτουμε τους περιορισμούς ξένου κλειδιού.

```
alter table review

    add(
        foreign key(project_ID) references project(ID),
        foreign key(reviewer_ID) references researcher(ID));

alter table deliverable
    add foreign key(project_ID) references project(ID);

alter table scientific_field_of_project
    add(
        foreign key(project_ID) references project(ID),
        foreign key(name_of_field) references scientific_field(name_of_field));
alter table executive
    add foreign key(project_ID) references project(ID);

alter table manages
    add(
        foreign key(project_ID) references project(ID),
        foreign key(organization_ID) references organization(ID));
```

```
alter table works_at_project

    add(
        foreign key(researcher_ID) references researcher(ID),
        foreign key(project_ID) references project(ID));

alter table scientific_director
    add(
        foreign key(project_ID) references project(ID),
        foreign key(researcher_ID) references researcher(ID));

alter table works_at
    add(
        foreign key(organization_ID) references organization(ID),
        foreign key(researcher_ID) references researcher(ID));

alter table contact_numbers
    add foreign key(organization_ID) references organization(ID);

alter table firm
    add foreign key(ID) references organization(ID);

alter table university
    add foreign key(ID) references organization(ID);

alter table research_center
    add foreign key(ID) references organization(ID);

alter table funding_program
    add foreign key(project_ID) references project(ID);
```

Indexes

Τέλος, έχουμε την δημιουργία των κατάλληλων ευρετηρίων :

```
create index executive_name on executive(name);
create index proj_start_date on project(start_date);
create index proj_end_date on project(end_date);
create index researcher_age on researcher(age);
```

Η επιλογή των συγκεκριμένων ιδιοτήτων για την δημιουργία ευρετηρίων έγινε επειδή οι ιδιότητες αυτές χρησιμοποιούνται πιο συχνά στα queries μας, πέρα των primary keys.
Τα primary keys δεν χρειάζονται ευρετήρια γιατί είναι ήδη indexed.

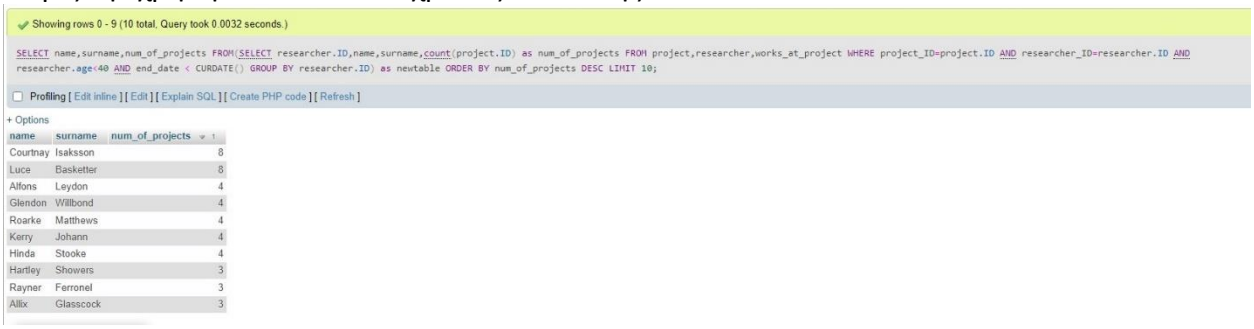
Ενδεικτικά, παραθέτουμε ένα παράδειγμα χρήσης των indexes και πως βελτιώνουν την απόδοση της βάσης μας:

Αν καλέσουμε το query για το ερώτημα 3.6 :

```
SELECT name,surname,num_of_projects

FROM(SELECT researcher.ID,name,surname,count(project.ID) as
num_of_projects
    FROM project,researcher,works_at_project
    WHERE project_ID=project.ID AND researcher_ID=researcher.ID AND
researcher.age<40 AND end_date < CURDATE()
    GROUP BY researcher.ID) as newtable
ORDER BY num_of_projects DESC
LIMIT 10;
```

Χωρίς την χρήση των indexes ο χρόνος εκτέλεσης είναι στα 0.0032 sec. .


Showing rows 0 - 9 (10 total, Query took 0.0032 seconds.)

```
SELECT name,surname,num_of_projects FROM(SELECT researcher.ID,name,surname,count(project.ID) as num_of_projects FROM project,researcher,works_at_project WHERE project_ID=project.ID AND researcher_ID=researcher.ID AND
researcher.age<40 AND end_date < CURDATE() GROUP BY researcher.ID) as newtable ORDER BY num_of_projects DESC LIMIT 10;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

+ Options

name	surname	num_of_projects
Courtnay	Isaksson	8
Luce	Basketter	8
Alfons	Leydon	4
Glendon	Willbond	4
Roarke	Matthews	4
Kerry	Johann	4
Hinda	Stooke	4
Hartley	Showers	3
Rayner	Ferronel	3
Allix	Glasscock	3

Query took 0.0032 sec.

Με την χρήση των indexes ο χρόνος μειώνεται στα 0.0029 sec.


Showing rows 0 - 9 (10 total, Query took 0.0029 seconds.)

```
SELECT name,surname,num_of_projects FROM(SELECT researcher.ID,name,surname,count(project.ID) as num_of_projects FROM project,researcher,works_at_project WHERE project_ID=project.ID AND researcher_ID=researcher.ID AND
researcher.age<40 AND end_date < CURDATE() GROUP BY researcher.ID) as newtable ORDER BY num_of_projects DESC LIMIT 10;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

+ Options

name	surname	num_of_projects
Courtnay	Isaksson	8
Luce	Basketter	8
Roarke	Matthews	4
Kerry	Johann	4
Hinda	Stooke	4
Glendon	Willbond	4
Alfons	Leydon	4
Hartley	Showers	3
Rayner	Ferronel	3
Allix	Glasscock	3

Query took 0.0029 sec.

Η διαφορά δεν φαίνεται πολύ μεγάλη, και αυτό οφείλεται κυρίως στο πλήθος των δεδομένων μας, που δεν είναι πολύ μεγάλο. Στην περίπτωση εκατομμυρίων records σε μια πραγματική βάσης, η χρονική διαφορά θα ήταν πιο μεγάλη, και η χρησιμότητα των indexes πιο εμφανέστερη.

DML SCRIPT

Για να εισάγουμε δεδομένα στην βάση μας χρησιμοποιήσαμε SQL DML . Παρακάτω παραθέτουμε κάποια ενδεικτικά παραδείγματα για την εισαγωγή δεδομένων σε κάθε table :

```
INSERT INTO `project` (`ID`, `title`, `funds`, `start_date`, `end_date`, `summary`,
`duration`) VALUES
(1, ' Neural Mechanisms of Avoidance Learning', 9705883, '2021-07-24', '2023-07-24 ', 'Nullam
sit amet turpis elementum ligula vehicula consequat.
 Morbi a ipsum. Integer a nibh.', 0);

INSERT INTO `researcher` (`ID`, `name`, `surname`, `date_of_birth`, `sex`, `age`) VALUES
(1, 'Dreddy', 'Varga', '1968-10-27', 'Female', NULL),
(2, 'Jammal', 'Keiley', '1968-04-25', 'Male', NULL);

INSERT INTO `organization` (`ID`, `name`, `abbreviation`, `address_PC`, `address_street`,
`address_city`, `address_number`) VALUES
(42,'Papaspyrou Lmtd & Co','PLC',42,'SHMMY','NTUA',42),
(41,'Review Center','RC',1570,'Kanari','Athens',24);

INSERT INTO `executive` (`ID`, `project_ID`, `name`) VALUES
(2, 1, 'Tiphany Piatto');

INSERT INTO `scientific_field` (`name_of_field`) VALUES
('Biology'),
('Chemistry'),
('Community');

INSERT INTO `scientific_field_of_project` (`project_ID`, `name_of_field`) VALUES
(1, 'Biology'),
(1, 'Medicine');

INSERT INTO works_at (organization_ID,researcher_ID,starting_date) VALUES
(1,1,'2001-7-12'),
(1,41,'2007-8-19');

INSERT INTO `review` (`project_ID`, `reviewer_ID`, `review_grade`, `review_date`) VALUES
(1, 153, 4, '1990-8-19'),
(2, 151, 2, '1990-8-19'),
(3, 152, 6, '1990-8-19');

INSERT INTO `research_center` (`ID`, `ministry_budget`, `private_action_budget`) VALUES
(20, 899030, 4046063);

INSERT INTO `university` (`ID`, `ministry_budget`) VALUES
(1, 7965942),
(2, 2430616);

INSERT INTO `firm` (`ID`, `equity`) VALUES
(16, 57154439);

INSERT INTO `funding_program` (`project_ID`, `name`, `management`) VALUES
(3, 'Almo', 'Exact Science office');
```

```
INSERT INTO manages(organization_ID,project_ID) VALUES

(1,1),
(1,2);

INSERT INTO works_at_project(researcher_ID,project_ID) VALUES
(1,1),
(1,2);

INSERT INTO `deliverable` (`project_ID`, `deliverable_title`, `summary`,
`date_of_delivery`) VALUES
(1, 'In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt.',
'Praesent id massa id nisl venenatis
lacinia. Aenean sit amet justo. Morbi ut odio.\n\nCras mi pede, malesuada in, imperdiet et,
commodo vulputate, justo.
In blandit ultrices enim. Lorem ipsum dolor sit amet, consectetuer adipiscing
elit.\n\nProin interdum mauris non ligula pellentesque ultrices.
Phasellus id sapien in sapien iaculis congue. Vivamus metus arcu, adipiscing molestie,
hendrerit at, vulputate vitae, nisl.', '1990-8-19');

INSERT INTO `contact_numbers` (`organization_ID`, `phone_number`) VALUES
(1, '(655) 4339091'),
(2, '(808) 7943139');

INSERT INTO scientific_director (researcher_ID, project_ID) VALUES
(1,1),
(1,2),
(1,3);
```

Ερωτήματα (Queries)

Παρακάτω θα παραθέσουμε τους κώδικες που υλοποιούν τα ερωτήματα του χρήστη:

3.1:

3.2:

Η πρώτη προβολή, έργα ανά ερευνητή:

```
CREATE VIEW projects_per_researcher AS
    SELECT title, surname, researcher.ID as res_ID
    FROM project,works_at_project,researcher
    WHERE project.ID = project_ID and researcher_ID = researcher.ID
```

Η δεύτερη προβολή, ερευνητές ανά έργα:

```
CREATE VIEW researchers_per_project AS
    SELECT title, surname, project.ID as proj_ID
    FROM project,works_at_project,researcher
    WHERE project.ID = project_ID and researcher_ID = researcher.ID
```

3.3:

Το query που υλοποιεί το ερώτημα 3.3:

```
SELECT DISTINCT title,name,surname

FROM project,works_at_project as
w,researcher,scientific_field_of_project as s

WHERE s.name_of_field = ["input"] AND

    w.researcher_ID = researcher.ID AND

    w.project_ID = project.ID AND

    s.project_ID = project.ID AND

    end_date > CURDATE();
```

Μέσω του καρτεσιανού γινομένου που δημιουργεί η FROM μεταξύ των πινάκων project, works_at_project, researcher, scientific_field_of_project, επιλέγει όλους τους ερευνητές που δουλεύουν στα (ενεργά) έργα των οποίων το επιστημονικό πεδίο είναι αυτό που εισάγει ο χρήστης στην είσοδο.
Αυτό επιτυγχάνεται μέσα από την σύνδεση των project_ID και researcher_ID του works_at_project με τα ID των project και researcher αντίστοιχα (σύνδεση ερευνητών με τα έργα στα οποία που δουλεύουν) και τον περιορισμό ότι τα επιλεγμένα rows θα έχουν ως name_of_field αυτό που όρισε ο χρήστης.

3.4:

Το query που υλοποιεί το ερώτημα 3.4:

```
CREATE VIEW researchers_per_project AS

    SELECT title, surname, project.ID as proj_ID
    FROM project,works_at_project,researcher
    WHERE project.ID = project_ID and researcher_ID = researcher.ID
    GROUP BY proj_ID;


SELECT org_ID,organization.name
FROM organization,(SELECT table1.org_ID,table1.num_of_proj
                    FROM (SELECT EXTRACT(YEAR from project.start_date) as years,
organization.ID as org_ID,
count(project.ID) as num_of_proj
                        FROM manages,organization,project
                        WHERE organization.ID = organization_ID AND
project.ID = project_ID
                        GROUP BY years,org_ID) as table1,
                        (SELECT EXTRACT(YEAR from project.start_date) as years,
                                    organization.ID as org_ID,
count(project.ID) as num_of_proj
                        FROM manages,organization,project
                        WHERE organization.ID = organization_ID AND
project.ID = project_ID
                        GROUP BY years,org_ID) as table2
                    WHERE table1.org_ID = table2.org_ID AND
                        table1.num_of_proj = table2.num_of_proj AND
                        table1.years - table2.years = 1) as newtable
WHERE org_ID = organization.ID AND num_of_proj>=10;
```

Έχουμε 2 υπο-ερωτήματα, το ένα “εμφωλιασμένο” στο άλλο.

Το “εσωτερικότερο” υπο-ερώτημα (SELECT EXTRACT(YEAR from… …) καλείται 2 φορές και δημιουργεί δύο πίνακες (table1 και table2) οι οποίοι περιέχουν τα έργα κάθε οργανισμού ανά έτος.

Το δεύτερο υπο-ερώτημα ((SELECT table1.org_ID,…) as newtable), χρησιμοποιεί τους προσωρινούς πίνακες table1 και table2, και μέσω του καρτεσιανού γινομένου τους τους “διατρέχει” ώστε να βρει τους οργανισμούς με ίδιο ID, ίδιο αριθμό έργων και διαφορά ετών ίση με 1. Αυτό μας επιστρέφει τον πίνακα newtable, ο οποίος περιέχει όλα τα ID των οργανισμών οι οποίοι έχουν τον ίδιο αριθμό έργων για 2 συνεχόμενα χρόνια.

Τέλος, το κυρίως ερώτημα, κάνοντας το καρτεσιανό γινόμενο του newtable και organization, βρίσκει τα ονόματα των οργανισμών που έχουν πάνω από 10 έργα για ανά έτος , για 2 συνεχόμενα έτη.

3.5:

Το query που υλοποιεί το ερώτημα 3.5:

```
SELECT first_name,second_name,count(project_ID) as number_of_projects
FROM(
    SELECT field1.name_of_field as first_name, field2.name_of_field as
second_name, field1.project_ID
    FROM scientific_field_of_project as field1,
         scientific_field_of_project as field2
    WHERE field1.project_ID = field2.project_ID AND
        field1.name_of_field < field2.name_of_field
    ) newtable
GROUP BY first_name
ORDER BY number_of_projects DESC
LIMIT 3;
```

Το υπο-ερώτημα SELECT field1.name … επιστρέφει τον πίνακα newtable ο οποίος περιέχει όλα τα ζεύγη των επιστημονικών πεδίων που βρίσκονται στο ίδιο έργο.
Το κυρίως ερώτημα υπολογίζει τον αριθμό των έργων που έχει κάθε ζεύγος πεδίων, και επιλέγει τα 3 με τον μεγαλύτερο αριθμό έργων.

3.6:

Το query που υλοποιεί το ερώτημα 3.6:

```
SELECT name,surname,num_of_projects
FROM(SELECT researcher.ID,name,surname,count(project.ID) as num_of_projects
    FROM project,researcher,works_at_project
    WHERE project_ID=project.ID AND researcher_ID=researcher.ID AND
              researcher.age<40 AND end_date < CURDATE()
    GROUP BY researcher.ID) as newtable
ORDER BY num_of_projects DESC
LIMIT 10;
```

Το υπο-ερώτημα SELECT researcher.ID,name,… επιστρέφει τον πίνακα newtable,
ο οποίος περιέχει τους ερευνητές κάτω των 40 ετών, που δουλεύουν σε ενεργά έργα.

Το κυρίως ερώτημα SELECT name,surname,…. επιστρέφει τους top-10 νέους ερευνητές που δουλεύουν στα περισσότερα ενεργά έργα.

3.7:

Το query που υλοποιεί το ερώτημα 3.7:

```
SELECT e.name, firm_ID, sum(p.funds) as total_funds
FROM executive as e, project as p,(SELECT firm.ID as firm_ID, project_ID
                                   FROM firm,manages
                                   WHERE firm.ID = manages.organization_ID
                                   ORDER BY firm.ID) as proj_per_firm
WHERE proj_per_firm.project_ID = e.project_ID AND p.ID = e.project_ID
GROUP BY firm_ID, e.name
ORDER BY total_funds DESC
LIMIT 5;
```

Το εσωτερικό υπο-ερώτημα SELECT firm.ID as … βρίσκει τα έργα ανά εταιρεία και τα επιστρέφει στον πίνακα proj_per_firm. Το κυρίως ερώτημα SELECT e.name … κάνει το καρτεσιανό γινόμενο των πινάκων executive, project, proj_per_firm, βρίσκει τα στελέχη που εργάζονται στα έργα των εταιρειών, και επιστρέφει τα top-5 στελέχη που έχουν δώσει το μεγαλύτερο (αθροιστικά) ποσό χρηματοδότησης στα έργα που διαχειρίζεται μια εταιρεία.

3.8:

Το query που υλοποιεί το ερώτημα 3.8:

```
SELECT ID,name,surname,num_of_proj
FROM(
    SELECT ID,name,surname, count(project_ID) as num_of_proj
    FROM(
        SELECT DISTINCT researcher.ID,name,surname,w.project_ID
        FROM project,works_at_project as w ,researcher,deliverable
        WHERE w.project_ID = project.ID AND
            researcher_ID = researcher.ID AND
            w.project_ID NOT IN (SELECT project_ID
                                 FROM deliverable)
        ) newtable
    GROUP BY ID) newtable2
WHERE num_of_proj>=5;
```

Το υπο-ερώτημα SELECT DISTINCT researcher.ID … βρίσκει τους ερευνητές και τα έργα στα οποία δουλεύουν, τα οποία δεν έχουν παραδοτέα, και επιστρέφει τον πίνακα newtable.

Το επόμενο υπο-ερώτημα SELECT ID,name,… υπολογίζει των αριθμών των έργων ανά ερευνητή από το newtable, και επιστρέφει τον πίνακα newtable2.
Το κυρίως ερώτημα SELECT ID,name,surname,… βρίσκει τους ερευνητές που δουλεύουν σε 5 ή παραπάνω έργα χωρίς παραδοτέα.

Triggers

Στην βάση μας θέλαμε να βάλουμε κάποιους αυτόματους ελέγχους και περιορισμούς για όταν ο χρήστης εισάγει δεδομένα στην βάση δεδομένων . Αυτό το υλοποιήσαμε χρησιμοποιώντας τα παρακάτω triggers :

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` TRIGGER `project_duration`
 BEFORE INSERT ON `project` FOR EACH ROW
 BEGIN
 DECLARE msg VARCHAR(255);
IF (((EXTRACT(YEAR FROM new.end_date) - EXTRACT(YEAR FROM
new.start_date)))<1)
      OR ((EXTRACT(YEAR FROM new.end_date) - EXTRACT(YEAR FROM
new.start_date)) > 4 )
THEN SET msg := 'Error:The duration of a project must be between 1-4
years.';
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END $$
DELIMITER ;
```

Με το παραπάνω trigger κάνουμε έλεγχο της διάρκειας ενός νέου project που εισάγουμε στην βάσης μας. Το trigger αφορά το table “project” και αν η διάρκεια του έργου είναι μικρότερη του ενός έτους ή ξεπερνάει τα τέσσερα έτη το trigger δεν αφήνει τον χρήστη να εισάγει το νέο project γυρνώντας ένα μήνυμα για το τι δεν εισήγαγε σωστά , για αυτό και το trigger είναι before insert.

```
DELIMITER $$
CREATE  TRIGGER unique_reviewer
BEFORE INSERT ON review
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
if( NEW.reviewer_ID IN
    (SELECT researcher_ID
    FROM manages as m,works_at as w
    WHERE m.project_ID = NEW.project_ID AND
        w.organization_ID = m.organization_ID) )
THEN SET msg :='The reviewer can not work in the organization that
manages the project';
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END $$
DELIMITER ;
```

Με αυτό το trigger θέλουμε να βάλουμε έναν περιορισμό στο ότι ο αξιολογητής του έργου δεν εργάζεται στον οργανισμό που έχει αναλάβει το έργο. Το trigger αφορά το table review και ελέγχει αν το reviewer_ID (που είναι το researcher_ID του αξιολογητή) υπάρχει στην λίστα με τα researcher_ID εκείνων που δουλεύουν στον οργανισμό που συμμετέχει στην πρόταση. Αν το βρει σε αυτήν την λίστα σταματάω ο trigger το insert και εμφανίζει error message.

```
DELIMITER $$

CREATE  TRIGGER scientific_director_works_at_project
BEFORE INSERT ON scientific_director
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
if( NEW.project_ID NOT IN
    (SELECT project_ID
    FROM works_at_project where researcher_ID =
NEW.researcher_ID ) )
THEN SET msg :='The scientific director must work at the project
that he supervises';
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END $$
DELIMITER ;
```

Αυτό το trigger εξυπηρετεί τον σκοπό ο επιστημονικός διευθυντής να είναι και ένας από τους researchers που εργάζονται στο συγκεκριμένο έργο. Έτσι το trigger αφορά το table scientific_director και ελέγχει αν το NEW.project_ID (δηλαδή το έργο στο οποίο θέλουμε να βάλουμε επιστημονικό διευθυντή) δεν υπάρχει στην λίστα με τα project_ID στα οποία εργάζεται ο νέος επιστημονικός διευθυντής(ως ερευνητής) , τότε δεν προχωράει σε εισαγωγή δεδομένων και εμφανίζει αντίστοιχο error message.

```
DELIMITER $$

CREATE  TRIGGER researcher_works_in_the_organization
BEFORE INSERT ON works_at_project
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
if( NEW.researcher_ID NOT IN
    (SELECT researcher_ID
    FROM manages as m,works_at as w
    WHERE m.project_ID = NEW.project_ID AND
        w.organization_ID = m.organization_ID) )
THEN SET msg :='The researcher must work in the organization that
runs the project in order to work on it ';
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END $$
DELIMITER ;
```

Το παραπάνω trigger εμποδίζει τον χρήστη να εισάγει εργαζόμενους ερευνητές οι οποίοι δεν δουλεύουν στον οργανισμό που έχει αναλάβει το έργο.

```
DELIMITER $$

CREATE  TRIGGER one_funding_program_per_project
BEFORE INSERT ON funding_program
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
if( NEW.project_ID  IN
    (SELECT project_ID
    FROM  funding_program)
  )
THEN SET msg :='Every project gets money from one funding
program';
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END $$

DELIMITER ;
```

Το trigger δεν επιτρέπει ένα έργο να χρηματοδοτείται από περισσότερα από ένα προγράμματα.

```
DELIMITER $$
CREATE  TRIGGER one_funding_program_to_one_office
BEFORE INSERT ON funding_program
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR(255);
if( NEW.name  IN
    (SELECT name
     FROM  funding_program
     where management != NEW.management)
  )
THEN SET msg :='This program belongs to an other E.LI.DE.K
managment office';
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
END IF;
END $$
DELIMITER ;
```

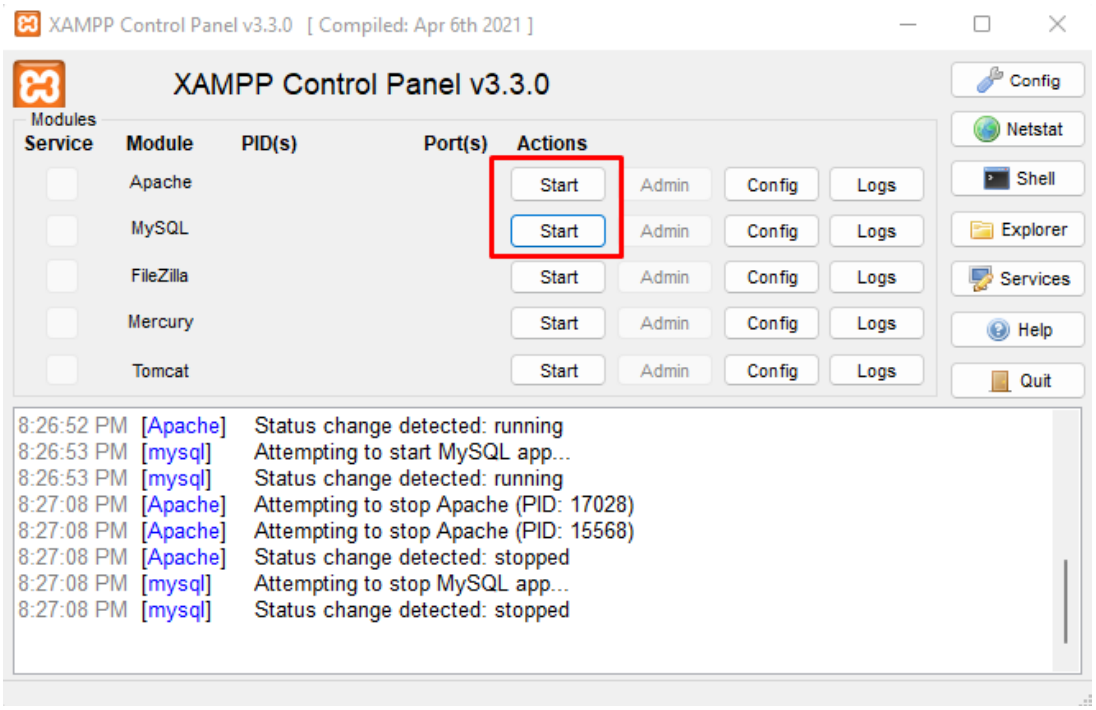
Το trigger αυτό θέτει τον περιορισμό να μην μπορεί ο χρήστης να εισάγει ένα πρόγραμμα που να ανήκει σε δύο διαφορετικές διευθύνσεις του Ε.ΛΙ.ΔΕ.Κ. .

Εγκατάσταση και Χρήση XAMPP

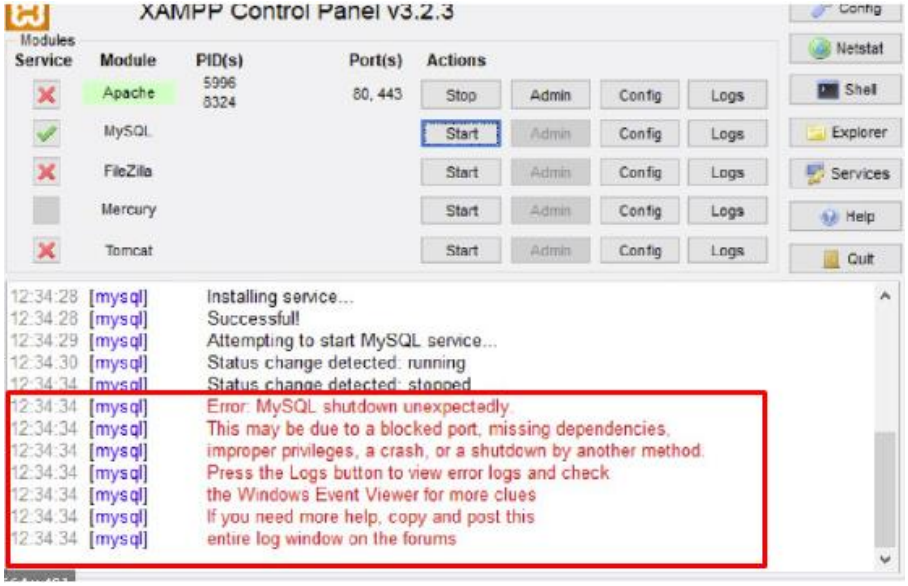
Για την ανάπτυξη και την διαχείριση της βάσης δεδομένων χρησιμοποιήσαμε το **XAMPP**Apache + MariaDB + PHP + Perl , μπορούμε να το εγκαταστήσουμε ακολουθώντας τον παρακάτω σύνδεσμο : https://www.apachefriends.org/download.html .

Οδηγίες χρήσης XAMPP

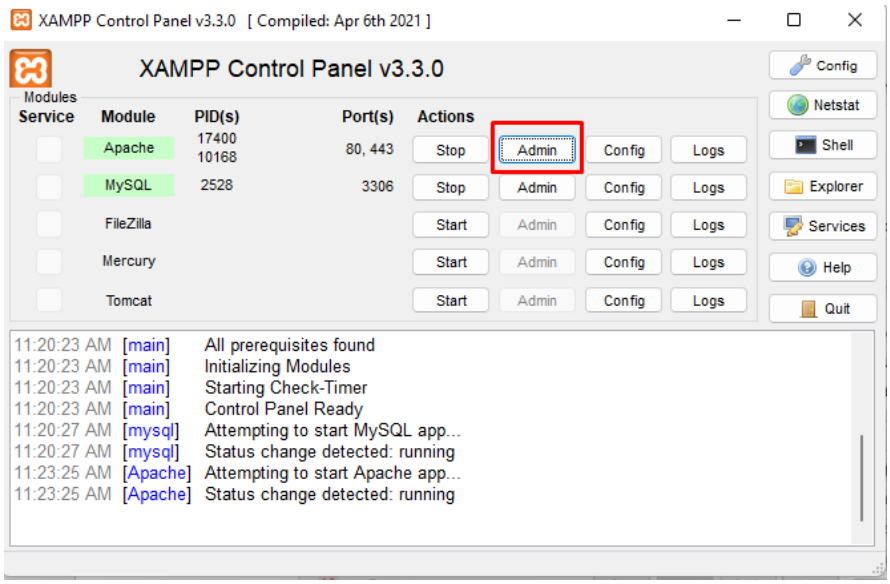
1. Μετά την εγκατάσταση ανοίγουμε την εφαρμογή και επιλέγουμε από την στήλη actions->Start τα Modules Apache , MySQL όπως φαίνεται και στην εικόνα :



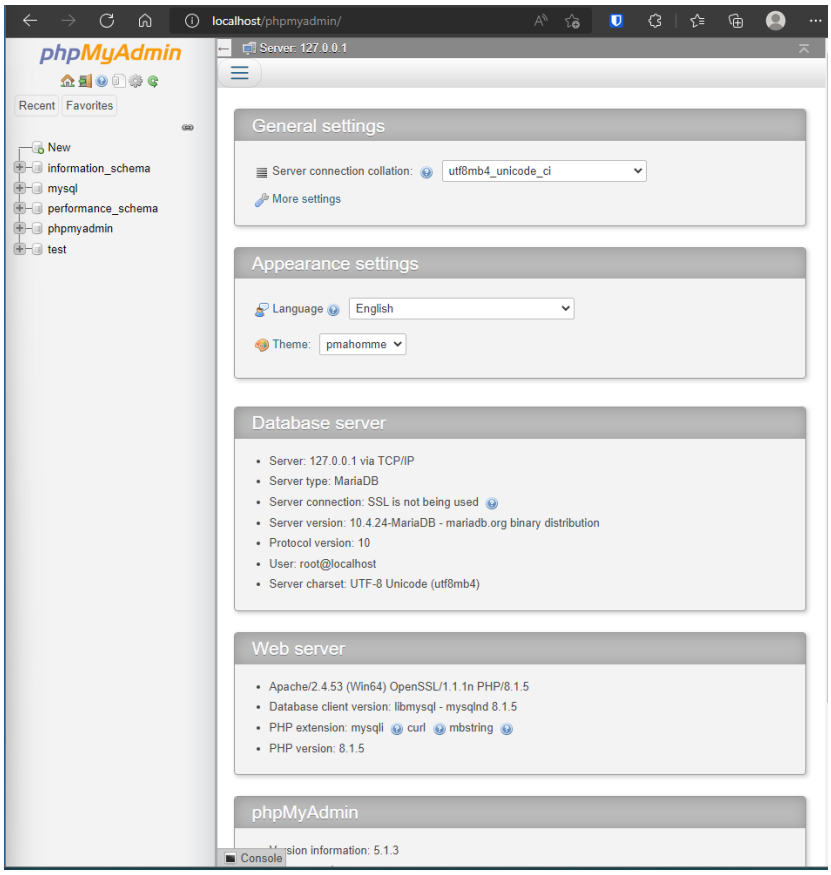
*Αν εμφανίζεται κάποιο error στο κάτω παράθυρο στην MySQL (με κόκκινο χρώμα) σημαίνει πως η προκαθορισμένη θύρα 80 για HTTP σύνδεση χρησιμοποιείται από κάποιο άλλο πρόγραμμα στον υπολογιστή μας και θα πρέπει να την αλλάξουμε σε κάποια άλλη π.χ. 8080.



2. Στην συνέχεια επιλέγουμε Apache Admin και μας εμφανίζεται στον bowser η αρχική σελίδα του XAMPP .



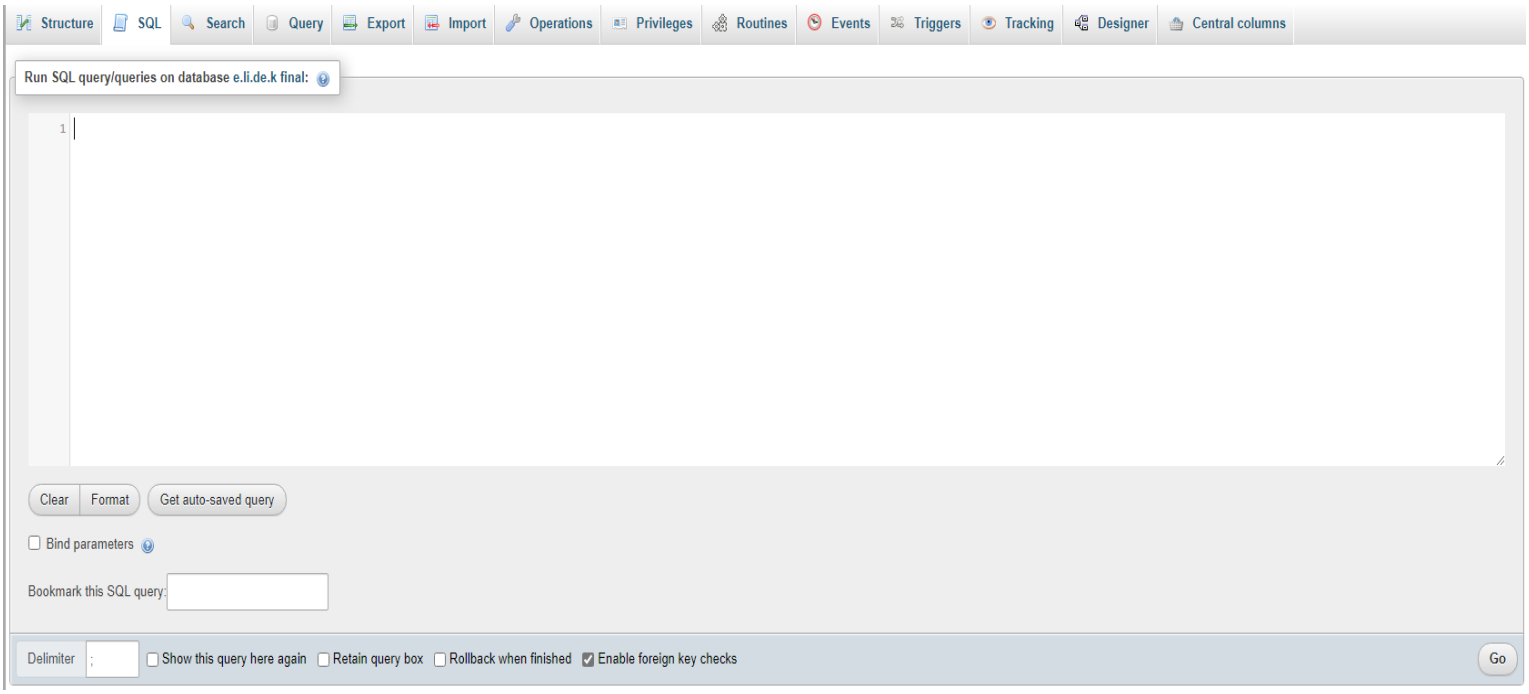
3. Πατάμε πάνω δεξιά το κουμπί phpMyAdmin και βρισκόμαστε στην αρχική σελίδα :



- Για την δημιουργία Βάσης Δεδομένων :
    1. Επιλέγουμε New πάνω αριστερά
    2. Συμπληρώνουμε το όνομα της βάσης μας
    3. Πατάμε CREATE

Για να προσθέσουμε tables , queries , data , triggers , indexes etc. :

- Μπορούμε να εισάγουμε κώδικα SQL πατώντας στην ενότητα SQL :



Εναλλακτικά μπορούμε να χρησιμοποιήσουμε και το γραφικό περιβάλλον του XAMPP δηλαδή :

Στην ενότητα Structure μπορούμε να δημιουργήσουμε tables , στην ενότητα queries να δημιουργήσουμε queries ενώ στην ενότητα insert μπορούμε να εισάγομε δεδομένα στην βάση κατευθείαν χωρίς την χρήση SQL κώδικα .

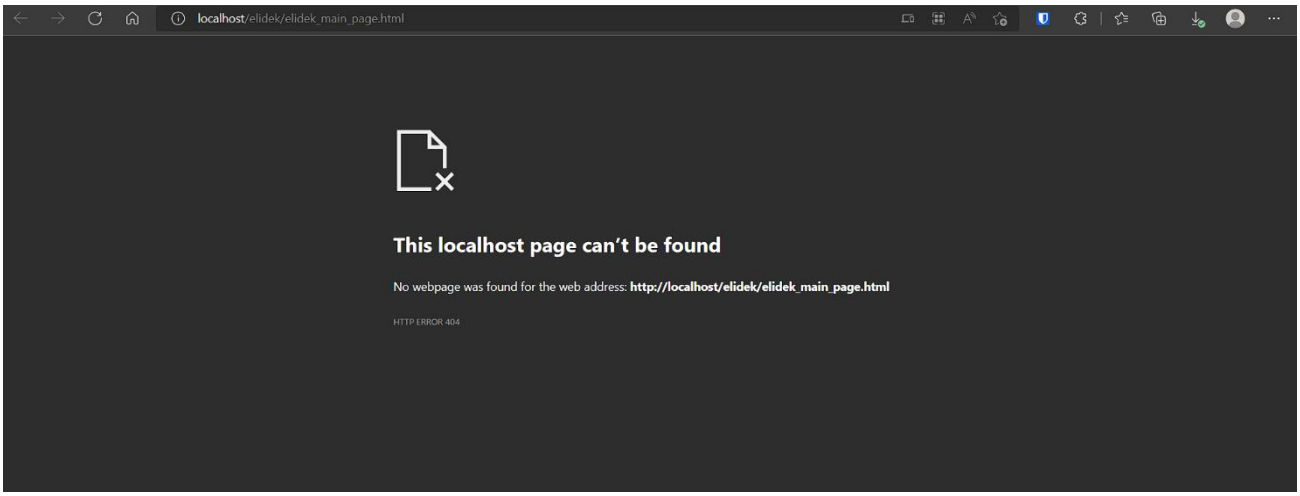
Ότι δημιουργούμε μπορούμε μετέπειτα από την ενότητα export να το εξάγουμε σε αρχείο SQL.

Στην ενότητα operation έχουμε την δυνατότητα να κάνουμε empty / drop a table η ακόμα και drop ολόκληρη την βάση .

Εγκατάσταση και Χρήση Εφαρμογής ΕΛ.ΙΔ.Ε.Κ.

1. Συνδεόμαστε στον localhost με χρήση XAMPP όπως αναφέρουν οι παραπάνω οδηγίες.
2. Δημιουργούμε Βάση Δεδομένων με ονομασία “**elidek**” (! ΠΡΟΣΟΧΗ η βάση θα πρέπει να ονομαστεί ακριβώς έτσι αλλιώς η εφαρμογή δεν θα λειτουργεί σωστά)
3. Στο Phpmyadmin στην ενότητα SQL εισάγουμε τα SQL scripts (copy-paste) ή κάνοντας τα κατευθείαν import από την ομώνυμη ενότητα , που βρίσκονται στον φάκελο “sql_scripts” στο GitHub repo μας , με την ακόλουθη σειρά :
   i. create_database.sql
   ii. triggers.sql
   iii. insert_all.sql
   iv. update_duration_age.sql

4. Κάνουμε download όλων των αρχείων-κώδικα που βρίσκονται στον φάκελο “WebPageCode” του GitHub repository.
5. Δημιουργούμε μέσα στον φάκελο αρχείων του xampp->htdocs (by default πρέπει να βρίσκεται στην θέση “C:\xampp\htdocs”) έναν νέο φάκελο με ονομασία “**elidek**”.
6. Τοποθετούμε στο φάκελο που μόλις δημιουργήσαμε όλα τα αρχεία που κάναμε download από τον φάκελο “ WebPageCode ”.
7. Στον browser γράφουμε την διεύθυνση “http://localhost/elidek/elidek_main_page.html ” , Αν εμφανίζεται το ακόλουθο error:



Πρέπει να αλλάξουμε την default port του localhost και να αντικαταστήσετέ την καινούργια στο παρακάτω URL(αλλάζουμε το highlighted τμήμα) , δηλαδή αν π.χ. αλλάξουμε το port από 80 σε 8080 τότε η παραπάνω διεύθυνση γίνεται : http://localhost:8080/elidek/elidek_main_page.html.

8. Η εφαρμογή πρέπει να ανοίγει κανονικά να εμφανίζεται το παρακάτω home page και είναι έτοιμη να την χρησιμοποιήσουμε !!!

Main Page Photo

GitHub repo

https://github.com/georgebaris/team29_EL.ID.E.K..git

ΒΙΒΛΙΟΓΡΑΦΙΑ

- *Διαφανείς μαθήματος «ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ».*
- *Βιβλίο «Συστήματα Βάσεων Δεδομένων : Η Πλήρη Θεωρία των Βάσεων Δεδομένων», Abraham Silberschatz , Herny F. Korth , S. Sudarshan.*
- "Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel".
- "draw.io – Diagrams for Confluence and Jira - draw.io (drawio-app.com)".