A picture containing text, businesscard, envelope, document

Description automatically generated

ΠΕΡΙΕΧΟΜΕΝΑ

1. *ΕΙΣΑΓΩΓΗ ………………………………………………………………………………………...σελ. 3*
2. *ΔΙΑΓΡΑΜΜΑ ER ………………………………………………………………………………….σελ. 3*
3. *ΕΥΡΕΣΗ ΟΝΤΟΤΗΤΩΝ……………………………………………………………..σελ. 4*
4. *ΕΥΡΕΣΗ ΣΥΣΧΕΤΙΣΕΩΝ……………………………………………………………..σελ. 5*
5. *ΤΕΛΙΚΗ ΜΟΡΦΗ ER ΔΙΑΓΡΑΜΜΑΤΟΣ…………………………………...σελ. 6*
6. *ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ....................................................................................σελ.7*
7. *DDL SCRIPT……………………………………………………………………………………….σελ.10*
8. *DΑΤΑ BASE CREATION………………….......………………………………….σελ.10*
9. *FOREIGH KEYS.…………………………………………………………………σελ.12*
10. *INDEXES……………………………………………………………………………σελ.13*
11. *DML SCRIPT……………………………………………………………………………………..σελ.*
12. *QUERIES……………………………………………………………………….........…………...σελ.*
13. *TRIGGERS...............................................................................................σελ.*
14. *ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΧΡΗΣΗ XAMPP……………………………..………………………σελ.*
15. *ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΚΑΙ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ.………………………….σελ.*
16. *GITHUB REPOSITORY.……………………………………………………………………….σελ.*
17. *ΒΙΒΛΙΟΓΡΑΦΙΑ………………………………………………………………………………….σελ.*

*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 :

Diagram, schematic

Description automatically generated

*Σχεσιακό Σχήμα (Relational Schema)*

Το σχεσιακό σχήμα της βάσης δεδομένων μας είναι το παρακάτω :

A picture containing text, screenshot, indoor, computer

Description automatically generatedΗ αγγλική έκδοση:

A picture containing text, screenshot, indoor, computer

Description automatically generatedΚαι 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**));**

**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**

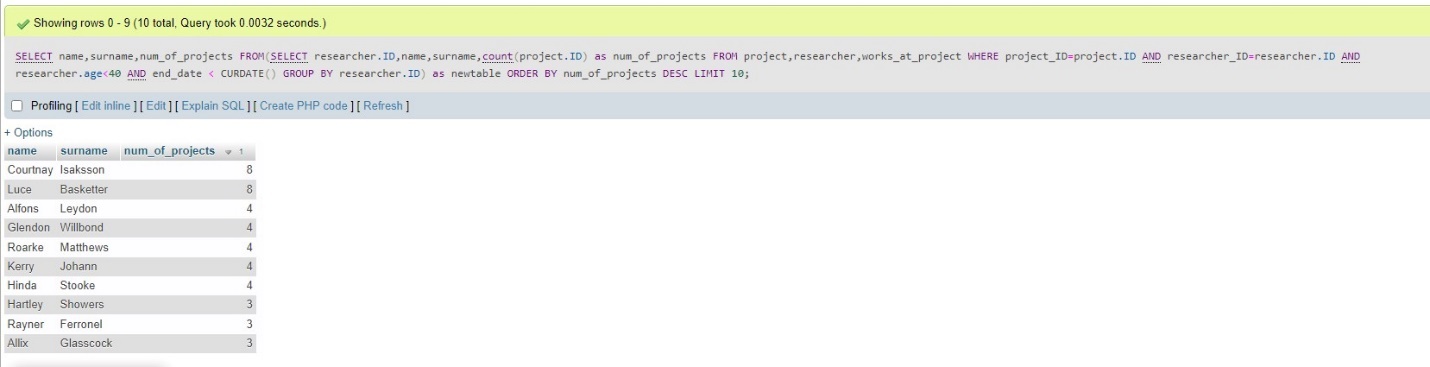
researcher**.**age**<**40 **AND** end\_date **<** CURDATE**()**

**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. .



Query took 0.0032 sec.

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

Graphical user interface, application, Word

Description automatically generated

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:**

To 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 **;**

Mε 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 **;**

Mε 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 **;**

To 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 **;**

To 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 :

Graphical user interface, text, application

Description automatically generated

\*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.

Graphical user interface

Description automatically generated

1. Στην συνέχεια επιλέγουμε Apache Admin και μας εμφανίζεται στον bowser η αρχική σελίδα του ΧΑΜPP .

Graphical user interface, application

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

* 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. :

* Graphical user interface, application, Word

  Description automatically generatedΜπορούμε 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 :
4. create\_database.sql
5. triggers.sql
6. insert\_all.sql
7. update\_duration\_age.sql
8. Κάνουμε download όλων των αρχείων-κώδικα που βρίσκονται στον φάκελο “WebPageCode” του GitHub repository.
9. Δημιουργούμε μέσα στον φάκελο αρχείων του xampp->htdocs (by default πρέπει να βρίσκεται στην θέση “C:\xampp\htdocs”) έναν νέο φάκελο με ονομασία “**elidek**”.
10. Τοποθετούμε στο φάκελο που μόλις δημιουργήσαμε όλα τα αρχεία που κάναμε download από τον φάκελο “ WebPageCode ”.
11. Text

    Description automatically generatedΣτον 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>.

1. 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](https://www.mockaroo.com/)”.
* “[draw.io – Diagrams for Confluence and Jira - draw.io (drawio-app.com)](https://drawio-app.com/)”.