

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

**2021-2022**

**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

**Αναφορά της Εξαμηνιαίας Εργασίας του μαθήματος «Βάσεις Δεδομένων» κατά το ακαδημαϊκό εξάμηνο 2021-2022**

**ΟΜΑΔΑ 6 ,ΜΕ ΣΥΣΤΑΤΙΚΑ ΤΗΣ ΜΕΛΗ: ΑΝΑΣΤΑΣΙΟΣ-ΜΑΡΙΝΟΣ ΖΟΡΜΠΑΛΑΣ (Α.Μ: 03119439) ΣΤΕΦΑΝΟΣ ΒΟΪΚΟΣ (Α.Μ: 03118162) ΑΓΓΕΛΟΣ ΜΗΤΡΟΚΩΤΣΑΣ (Α.Μ: 03118197)**

**Στοιχεία φοιτητών**

Ομάδα-Project: 6

Ονοματεπώνυμα Μελών:

* Αναστάσιος - ΜαρίνοςΖορμπαλάς (Α.Μ: 03119439)
* Στέφανος Βόικος (Α.Μ: 03118162)
* Άγγελος Μητροκώτσας (Α.Μ: 03118197)

Ακαδημαϊκό Εξάμηνο: 6ο

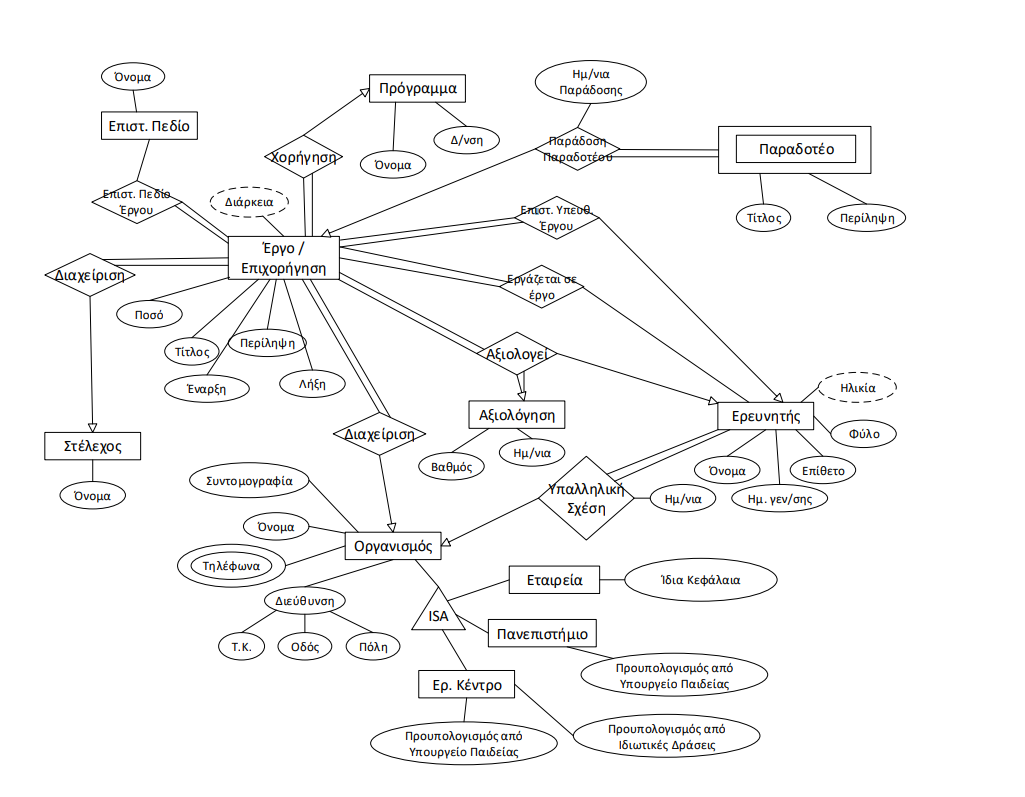
Μάθημα : Βάσεις Δεδομένων (ροή Λ)

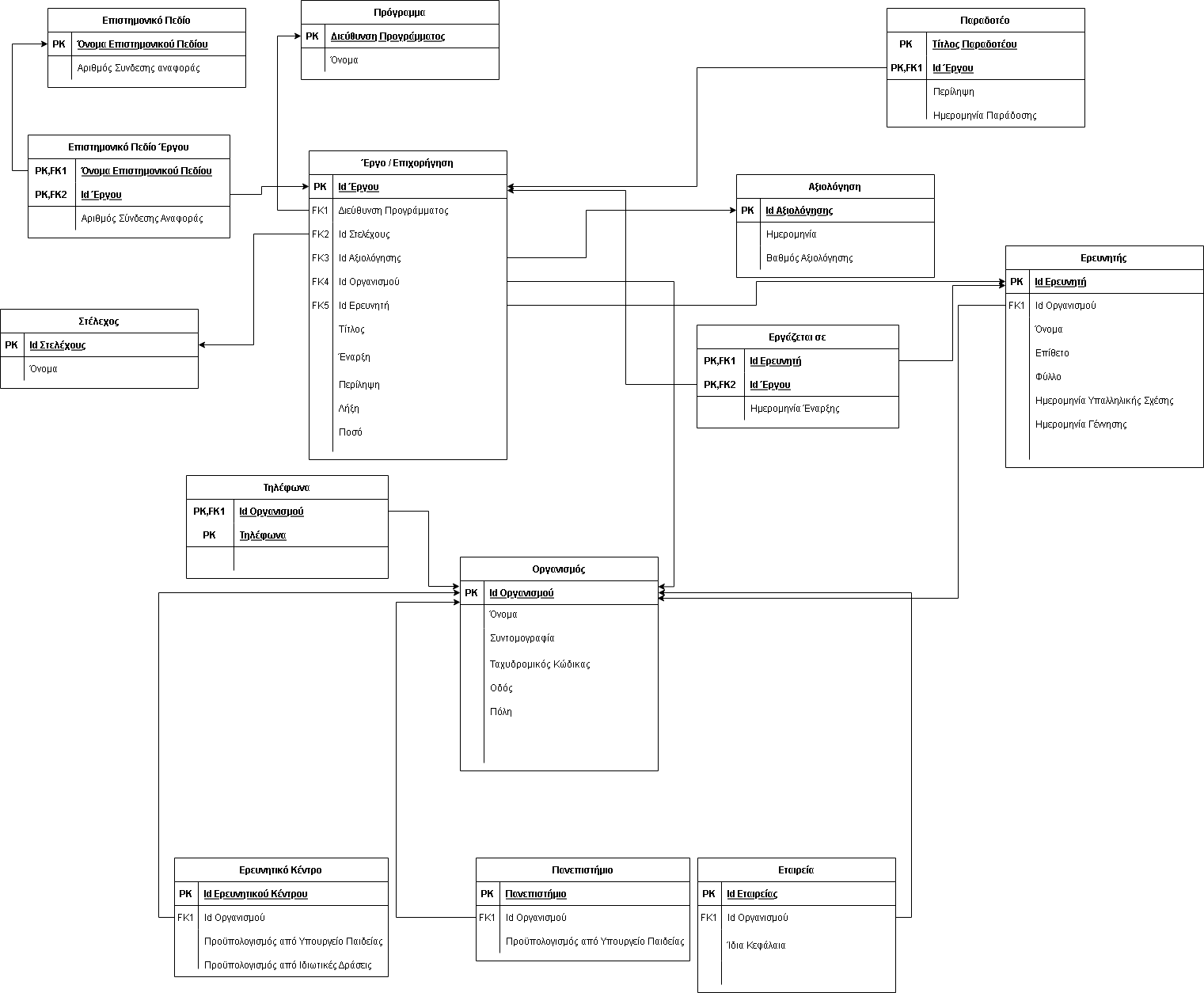
Σχολή: Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (Η.Μ.Μ.Υ)

**ΑΠΑΝΤΗΣΕΙΣ ΤΩΝ ΕΡΩΤΗΣΕΩΝ/ΒΗΜΑΤΩΝ ΤΗΣ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ ΣΤΟ ΜΑΘΗΜΑ «ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ»**

Υπογραμμίζουμε πως τη δική μας υλοποίηση τουER Διαγράμματος την υποβάλλαμε κανονικά στο Παραδοτέο 1, για τη συνέχεια προς το Παραδοτέο 2, οι απαντήσεις και οι υλοποιήσεις στηρίχθηκαν στο δοσμένο ERΔιάγραμμα, όπως αυτό δόθηκε στο siteτου helios. Με βάση την προτεινόμενη υλοποίηση του ER, προβαίνουμε στην απάντηση των ερωτήσεων και αποτελούν οι εξής:

1) Έχοντας κατανοήσει τους κανόνες μετάβασης/μετασχηματισμού του ERΔιαγράμματος σε Σχεσιακό Διάγραμμα, τους αξιοποιούμε καιστην αρχή της επόμενης σελίδας, παραθέτουμε το σχεσιακό μοντέλο που αντιστοιχεί στην απλή λύση του ERΔιαγράμματος. Την υλοποιήσαμε, χρησιμοποιώντας το onlineεργαλείο σχεδίασης draw.ioγια να το υλοποιήσουμε. Και η υλοποίηση του RelationalModel αποτελεί η εξής κάτωθι: (παραθέτουμε και το ERΔιάγραμμα για να είναι δίπλα δίπλα τα διαγράμματα και κατανοητή η αντιστοίχηση και μετάβαση από την μία στην άλλη κατάσταση)

ER Diagram:

Relational Model:

Σημειώσεις:

* PK = PrimaryKey
* FK = ForeignKey
* Τα mutli-valued χαρακτηριστικά στο Relational-schema, όπως αποτελεί η περίπτωση των αριθμών τηλεφώνου, είναι ξεχωριστοί πίνακες με PK το  
  χαρακτηριστικό τους και FK το PK της οντότητας στην οποία αντιστοιχούν.  
  Κάθε Weak-entity στο Relational-schema είναι ένας πίνακας με PK το PK του weak-entity καθώς και με FK τα  
  PK των strong-entities και τα υπόλοιπα χαρακτηριστικά του weak-entity όπως αποτελεί η περίπτωση των Παραδοτέων.

Στη συνέχεια, για να εξασφαλίσουμε την ορθότητα της βάσης δεδομένων υλοποιήσαμε τα παρακάτω:

1. Ακεραιότητα της οντότητας -> κάθε πίνακας έχει το δικό του primary key. Ως primary key επιλέγεται ένα από τα χαρακτηριστικά της κάθε οντότητας που είναι μοναδικό αναγνωριστικό της (πχ NFC\_ID,Service\_ID)

2. Ακεραιότητα αναφοράς -> αναφέρεται στα foreign keys τα οποία μπορούν να εναλλάσουν την τιμή τους μεταξύ δύο περιπτώσεων. Συνήθως, η τιμή τους είναι ίση με την τιμή κάποιου primary key ενός πίνακα εκείνη την στιγμή. Όπως εξηγήθηκε και στο σχεσιακό μοντέλο τα foreign keys που έχει κάθε πίνακας είναι αυτά που απαιτούνται ώστε να υλοποιούνται μέσω του λογισμικού οι επιθυμητές σχέσεις μεταξύ των πινάκων κατά την λειτουργία της βάσης

3. Ακεραιότητα τομέα -> όλες οι στήλες σε μία σχεσιακή βάση πρέπει να δηλώνονται σε έναν καθορισμένο τομέα. Συγκεκριμένα, επιλέγουμε VARCHAR για τα χαρακτηριστικά τύπου string μεταβλητού μήκους (roomname,firstname), BIGINT/INT για ακέραια data (ID’s), DATE/TIME/DATETIME για τις ημερομηνίες/ωρες (arrivaltime). Αρχικοποιούμε με NOT NULL, τα primary keys αυτόματα έχουν NOT NULL

Παραδώσαμε το Relational Model και το χτίσιμο που κάναμε της ΒΔ την κάναμε με την βοήθεια του phpmyAdmin και χτίσαμε εκεί την Βάση Δεδομένων μας αποτελεσματικά.

Συστήματα και Γλώσσες Προγραμματισμού που χρησιμοποιήσαμε:

• Windows (διανομή Windows 10 ως λειτουργικό σύστημα)

• Apache (Web Server)

• MySQL (DataBase Management System)

• PHP (ScriptingLanguage)

Χρησιμοποιήθηκε επίσης το Faker της Python, όπως επίσης και το πακέτο six >= 1.5 και το python-dateutil >= 2.4 προκειμένου να μπορέσουμε να χτίσουμε την βάση, με χρήση γεννητριών και δυνατοτήτων της Python, προκειμένου γρήγορα και αποδοτικά να χτίσουμε τα δεδομένα και να τα εισάγουμε ύστερα στην ΒΔ μας.

Επιπλέον, για τη δημιουργία και την επεξεργασία της βάσης δεδομένων χρησιμοποιήθηκε ως database (storage) engine η localhost/phpmyadmin, στην οποία φορτώσαμε από όλους τους απαραίτητους κώδικες. Φορτώσαμε αρχικά Create\_schema\_project\_6.sql προκειμένου να φτιααχτούν οι δομές και πάνω σε αυτές να μπουν τα δεδομένα, τα οποία εισάγουμε με το ELIDEK\_Database\_of\_Project\_6.sql .Για τα επόμενα βήματα 3.3-3.8 εισάγουμε το Quary.sql ενώ τα views και τα triggers είναι εντός του 1ου αρχείου, του Create\_schema\_Project\_6.sql.

Τα Queries και όλα τα ερωτήματα απαντούνται από αυτά τα αρχεία, τα οποία και βρίσκονται και εντός του zip που παραθέτουμε στο helios. Αναφορικά για το Drop Script έχουμε τα εξής (τα άλλα 2 είναι πολλών σειρών):

SET FOREIGN\_KEY\_CHECKS = 0;

DROP TABLE IF EXISTS Stelexos;

DROP TABLE IF EXISTS Program;

DROP TABLE IF EXISTS Axiologisi;

DROP TABLE IF EXISTS Organization;

DROP TABLE IF EXISTS Epistimoniko\_Pedio;

DROP TABLE IF EXISTS Researcher;

DROP TABLE IF EXISTS Ergo\_Epixorigisi;

DROP TABLE IF EXISTS Epistimoniko\_Pedio\_Ergou;

DROP TABLE IF EXISTS University;

DROP TABLE IF EXISTS Research\_Center;

DROP TABLE IF EXISTS Company;

DROP TABLE IF EXISTS Paradoteo;

DROP TABLE IF EXISTS Works\_For;

DROP TABLE IF EXISTS Phone;

SET FOREIGN\_KEY\_CHECKS = 1

Και έτσι απαντούμε και στο 2.2 ερώτημα έχοντας εισάγει και εξηγήσει τα scripts.

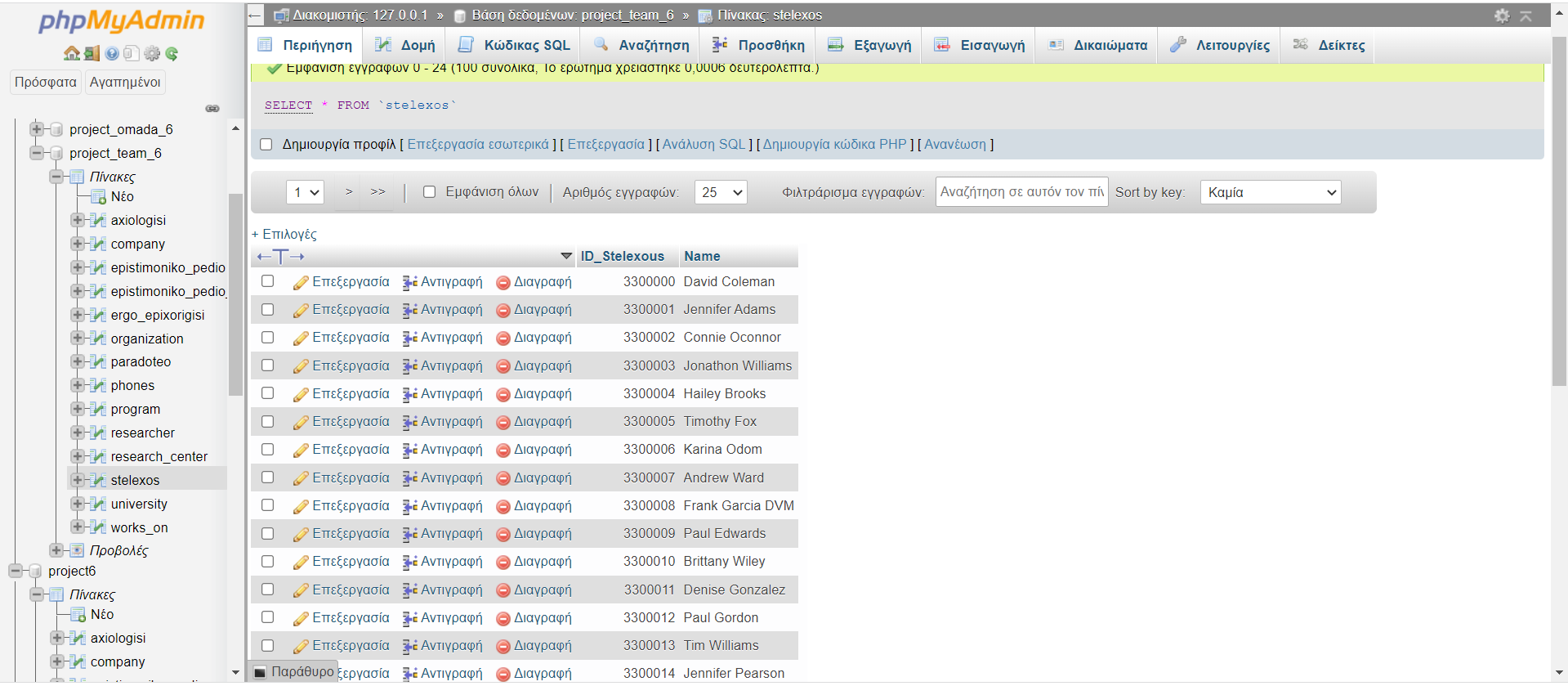
Θεωρήσεις:

* Εφαρμόζουμε όλα τα κριτήρια και κάναμε απαραίτητες προσθήκες, όπως το ότι για να μπορέσουν να απαντηθούν όλα τα ερωτήματα, θα πρέπει στο Epistimoniko\_Pedio να εφαρμόσουμε Code\_anaforas και κώδικας αντιστοίχησης έργου με επιστημονικό πεδίο για να μπορέσουμε να κάνουμε τις απαραίτητες προσθήκες. Επιπλέον θα πρέπει να εφαρμόσουμε ότι στο Works\_On, είναι απαραίτητο παρότι που από το προτεινόμενο ER δεν προκύπτει κάτι τέτοιο, θα πρέπει να γίνει προσθήκη Start\_Date, προκειμένου να μπορέσουμε να απαντήσουμε και πάλι σε απαραίτητο ερώτημα στο 3ο μέρος.
* Και σαφώς ότι ένας ερευνητής που ανήκει σε μία εταιρία, σε έναν οργανισμό, μπορεί να λειτουργήσει και να δουλέψει μόνο πάνω σε εργα του οργανισμού που δουλεύει, παραπέμπουμε να δειτε τα triggers

Αναφορικά με το indexing, προκειμένου να οριστούν τα indices της βάσης είναι απαραίτητο να ληφθεί υπόψη πως δημιουργούνται αυτόματα αυτά για τις κολώνες που αποτελούν.Τα primary keys γενικά αποτελούν IDs με μερικές εξαιρέσεις να μην συμβαίνει αυτό. Και όσε κάθε περίπτωση, χρησιμοποιούνται συνεχώς τόσο στα διάφορα queries (WHERE clauses, JOIN ON (...)), όσο και στα triggers που χρησιμοποιούνται για να υλοποιήσουν.

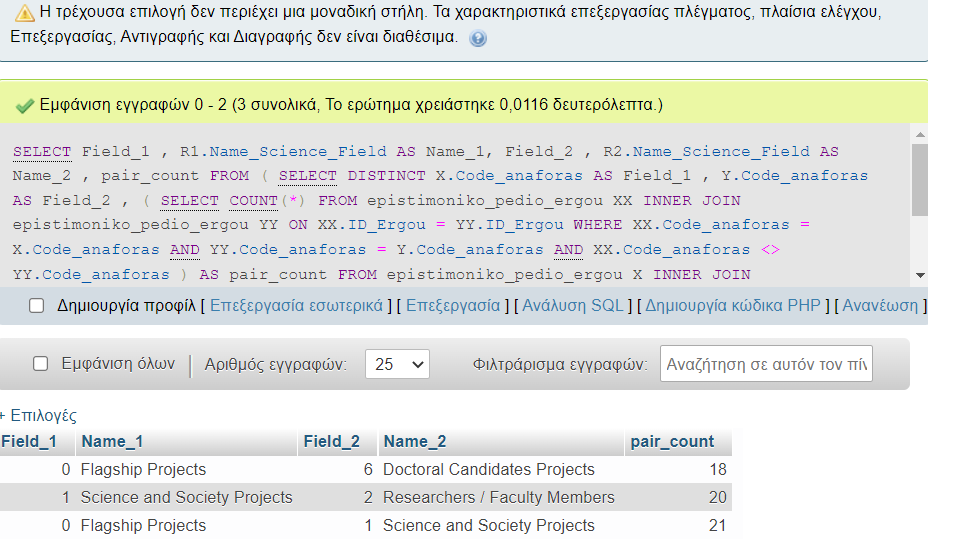
Πάμε για την 3, ως εξής:

Στην παρακάτω εικόνα βλέπουμε την βάση δεδομένων που μόλις υλοποιήσαμε:

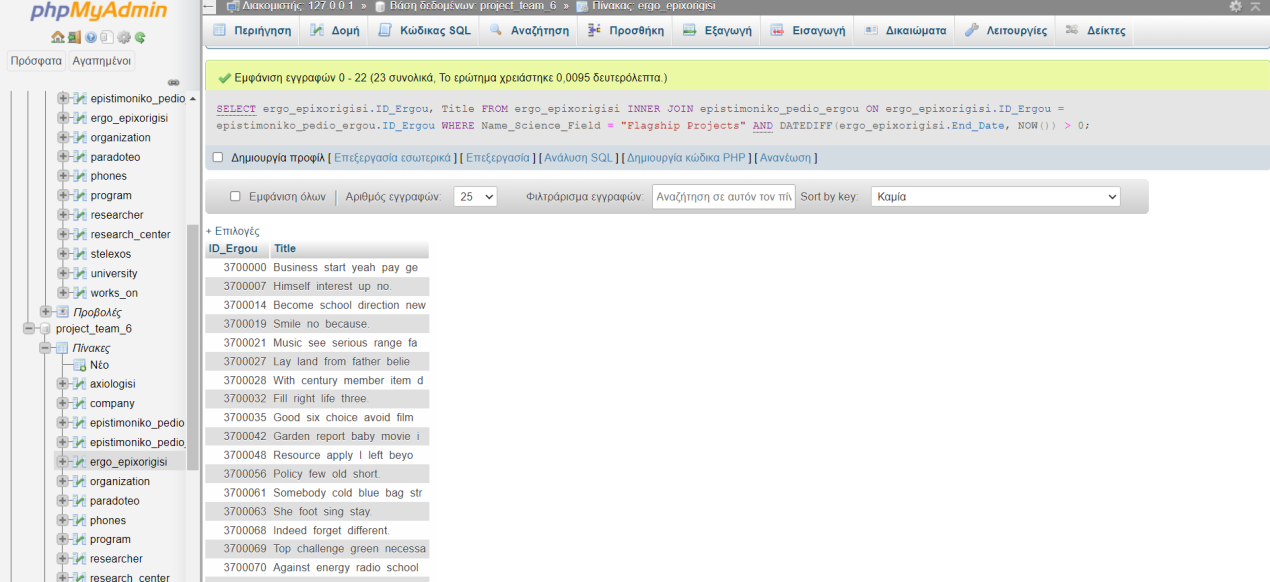
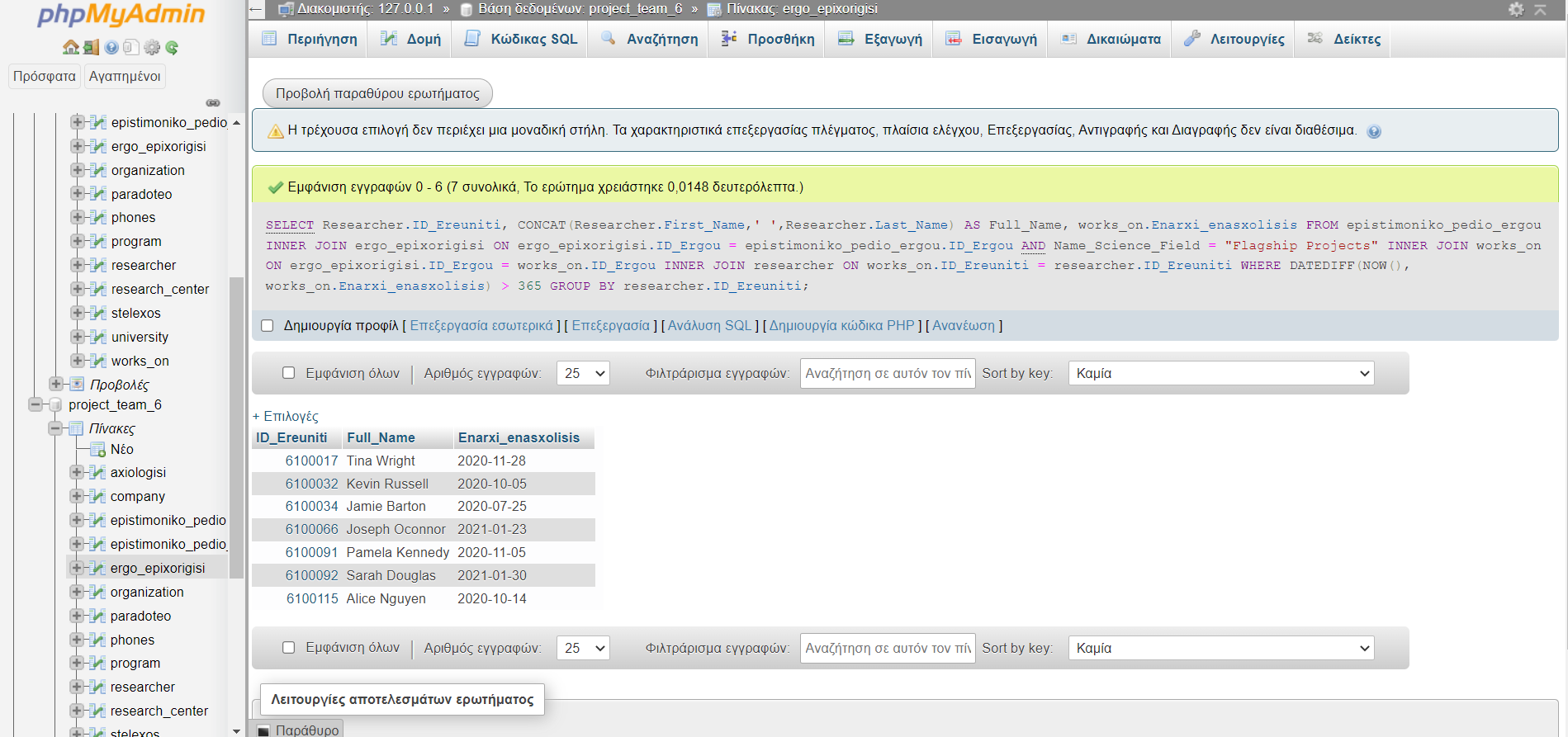


Παρακάτω αποτυπώνεται και η λειτουργία των queries τα οποία επιλέγουν τα αποτελέσματα του εκάστοτε ζητούμενου:

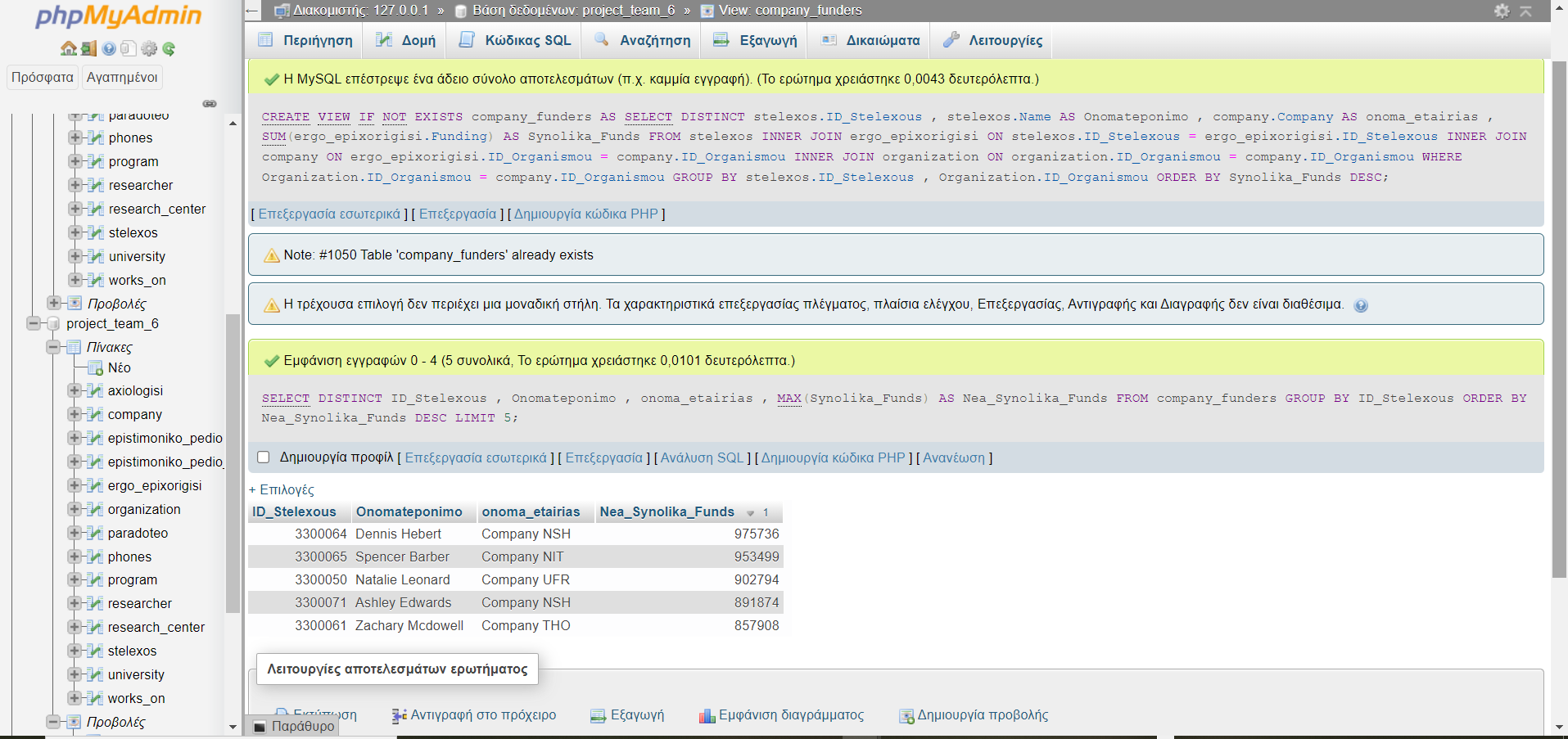
**3.5 Τα 3 κορυφαία ζεύγη που εμφανίστηκαν σε έργα**



**3.3 Έργα που χρηματοδοτούνται σε πεδίο που απέκτησε ενδιαφέρον και οι ερευνητές που ασχολούνται με αυτό το τελευταίο έτος**



**3.7 Τα πέντε καλύτερα στελέχη**



Και πιο αναλυτικά:

3.3) Query

ΕΡΓΑ ΠΟΥ ΧΡΗΜΑΤΟΔΟΤΟΥΝΤΑΙ ΣΕ ΕΝΑΝ ΣΥΓΚΕΚΡΙΜΕΝΟ ΕΠΙΣΤΗΜΟΝΙΚΟ ΤΟΜΕΑ, ΠΧ ΒΑΛΑΜΕ "Flagship Projects" που είναι ένας από τους υπαρκτούς του ΕΛΙΔΕΚ:

SELECT ergo\_epixorigisi.ID\_Ergou, Title

FROM ergo\_epixorigisi INNER JOIN epistimoniko\_pedio\_ergou

ON ergo\_epixorigisi.ID\_Ergou = epistimoniko\_pedio\_ergou.ID\_Ergou

WHERE Name\_Science\_Field = "Flagship Projects"

AND DATEDIFF(ergo\_epixorigisi.End\_Date, NOW()) > 0;

ΕΡΕΥΝΗΤΕΣ ΠΟΥ ΔΡΑΣΤΗΡΙΟΠΟΙΟΥΝΤΑΙ ΣΕ ΑΥΤΟ ΤΟ ΠΕΔΙΟ ΣΤΟ ΤΕΛΕΥΤΑΙΟ ΕΤΟΣ ΣΤΟΝ ΕΛΙΔΕΚ ΜΕ QUERY:

SELECT Researcher.ID\_Ereuniti, CONCAT(Researcher.First\_Name,' ',Researcher.Last\_Name)

AS Full\_Name, works\_on.Enarxi\_enasxolisis

FROM epistimoniko\_pedio\_ergou INNER JOIN ergo\_epixorigisi

ON ergo\_epixorigisi.ID\_Ergou = epistimoniko\_pedio\_ergou.ID\_Ergou

AND Name\_Science\_Field = "Flagship Projects"

INNER JOIN works\_on

ON ergo\_epixorigisi.ID\_Ergou = works\_on.ID\_Ergou

INNER JOIN researcher

ON works\_on.ID\_Ereuniti = researcher.ID\_Ereuniti

WHERE DATEDIFF(NOW(), works\_on.Enarxi\_enasxolisis) > 365

GROUP BY researcher.ID\_Ereuniti;

3.4) Query

ΟΡΓΑΝΙΣΜΟΙ ΠΟΥ ΕΧΟΥΝ ΛΑΒΕΙ ΤΟΝ ΙΔΙΟ ΑΡΙΘΜΟ ΕΡΓΩΝ ΣΕ ΔΙΑΣΤΗΜΑ ΔΤΟ ΣΥΝΕΧΟΜΕΝΩΝ ΕΤΩΝ ΜΕ ΤΟΥΛΑΧΙΣΤΟΝ 10 ΕΡΓΑ ΕΤΗΣΙΩΣ, ΜΕ:

SELECT Organization.ID\_Organismou , organization.Name , Organization.Syntomografia,

X. Y AS Year, Projects\_fetina

FROM (

SELECT DISTINCT ID\_Organismou AS O, YEAR(Start\_Date) as Y,

(

SELECT count(\*)

FROM ergo\_epixorigisi

WHERE YEAR(Start\_Date) = Y

AND ID\_Organismou = O

) AS Projects\_fetina ,

(

SELECT count(\*)

FROM ergo\_epixorigisi

WHERE YEAR(Start\_Date) + 1 = Y

AND ID\_Organismou = O

) AS Projects\_persina

FROM ergo\_epixorigisi

HAVING Projects\_fetina = Projects\_persina

AND Projects\_fetina >= 10

ORDER BY O

) X INNER JOIN Organization ON Organization.ID\_Organismou = X.O;

3.5) Query

ΕΜΦΑΝΙΣΗ top-3 ΖΕΥΓΩΝ ΩΣ ΕΞΗΣ:

SELECT Pedio\_1 , N1.Name\_Science\_Field AS Name\_1, Pedio\_2 , N2.Name\_Science\_Field AS Name\_2 , pair\_count FROM ( SELECT DISTINCT A.Code\_anaforas AS Pedio\_1 , B.Code\_anaforas AS Pedio\_2 , ( SELECT COUNT(\*) FROM epistimoniko\_pedio\_ergou AA INNER JOIN epistimoniko\_pedio\_ergou BB ON AA.ID\_Ergou = BB.ID\_Ergou WHERE AA.Code\_anaforas = A.Code\_anaforas AND BB.Code\_anaforas = B.Code\_anaforas AND AA.Code\_anaforas <> BB.Code\_anaforas ) AS pair\_count FROM epistimoniko\_pedio\_ergou A INNER JOIN epistimoniko\_pedio\_ergou B ON A.ID\_Ergou = B.ID\_Ergou WHERE A.Code\_anaforas < B.Code\_anaforas ORDER BY pair\_count DESC LIMIT 3 ) AS top\_pairs INNER JOIN epistimoniko\_pedio N1 ON N1.Code\_anaforas = top\_pairs.Pedio\_1 INNER JOIN epistimoniko\_pedio N2 ON top\_pairs.Pedio\_2 = N2.Code\_anaforas;

3.6) Query

CREATE VIEW IF NOT EXISTS metrima\_ergon AS

SELECT DISTINCT Researcher.ID\_Ereuniti AS ID\_Er , CONCAT(Researcher.First\_Name , ' ', Researcher.Last\_Name)

AS Onomateponimo , FLOOR( DATEDIFF( NOW() , Birth\_Date ) /365) AS Ilikia ,

(

SELECT COUNT(\*) FROM Works\_On

INNER JOIN Researcher

ON Works\_On.ID\_Ereuniti = Researcher.ID\_Ereuniti

INNER JOIN ergo\_epixorigisi

ON ergo\_epixorigisi.ID\_Ergou = Works\_On.ID\_Ergou

WHERE Researcher.ID\_Ereuniti = ID\_Er

AND DATEDIFF(ergo\_epixorigisi.End\_Date , NOW()) > 0 ) AS metritis\_ergon

FROM Works\_On INNER JOIN Researcher ON Works\_On.ID\_Ereuniti = Researcher.ID\_Ereuniti

WHERE DATEDIFF( NOW(),Birth\_Date) < 365\*40

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

-- we want lower than 365 \* 40 (40 years for those who are younger)

-- 365 due to 365 days per year

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

ORDER BY metritis\_ergon DESC;

SELECT DISTINCT Person2.ID\_Er , Person2.Onomateponimo , Person2.Ilikia , Person2.metritis\_ergon

FROM

(

SELECT \* FROM metrima\_ergon HAVING metritis\_ergon = MAX(metritis\_ergon)

) Person1

INNER JOIN metrima\_ergon Person2

ON Person1.metritis\_ergon = Person2.metritis\_ergon;

3.7) Query

Εύρεση των κορυφαίων 5 στελεχών το βρίσκω ως εξής:

CREATE VIEW IF NOT EXISTS company\_funders AS

SELECT DISTINCT stelexos.ID\_Stelexous , stelexos.Name AS Onomateponimo , company.Company AS onoma\_etairias , SUM(ergo\_epixorigisi.Funding) AS Synolika\_Funds

FROM stelexos INNER JOIN ergo\_epixorigisi

ON stelexos.ID\_Stelexous = ergo\_epixorigisi.ID\_Stelexous

INNER JOIN company

ON ergo\_epixorigisi.ID\_Organismou = company.ID\_Organismou

INNER JOIN organization

ON organization.ID\_Organismou = company.ID\_Organismou

WHERE Organization.ID\_Organismou = company.ID\_Organismou

GROUP BY stelexos.ID\_Stelexous , Organization.ID\_Organismou

ORDER BY Synolika\_Funds DESC;

SELECT DISTINCT ID\_Stelexous , Onomateponimo , onoma\_etairias , MAX(Synolika\_Funds) AS Nea\_Synolika\_Funds

FROM company\_funders

GROUP BY ID\_Stelexous

ORDER BY Nea\_Synolika\_Funds DESC LIMIT 5;

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

-- apo ta parapanw filtraroume kai theloume mono 5

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

3.8) Βρίσκω τους ερευνητές που εργάζονται σε 5 ή περισσότερα έργα που δεν έχουν παραδοτέα ως:

SELECT Ereunitis1.ID\_Ereuniti , CONCAT(Ereunitis1.First\_Name, ' ', Ereunitis1.Last\_Name) AS Onomatepwnimo , COUNT(X.ID\_Ergou) AS metritis\_ergon

FROM

(

SELECT ID\_Ergou FROM ergo\_epixorigisi

WHERE ID\_Ergou NOT IN ( SELECT ID\_Ergou FROM paradoteo )

) X

INNER JOIN Works\_On Y ON X.ID\_Ergou = Y.ID\_Ergou

INNER JOIN Researcher Ereunitis1 ON Y.ID\_Ereuniti = Ereunitis1.ID\_Ereuniti

GROUP BY Ereunitis1.ID\_Ereuniti

HAVING metritis\_ergon >= 5;

Το repository της Βάσης δεδομένων μας αποτελεί το:

[**https://github.com/StefanosVks7/Project6\_Database\_NTUA**](https://github.com/StefanosVks7/Project6_Database_NTUA)

ενώ πηγή εσόδων δεδομένων αποτελεί και το ZIP που ανεβάσαμε στο Helios εμπρόσθεσμα.

**Βήματα εγκατάστασης:**

1. Εγκαθιστούμε το πακέτο προγραμμάτων ελεύθερου λογισμικού XAMPP (Apache, MySQL, PHPandPerl) από τον ιστότοπο: https://www.apachefriends.org/download.html σύμφωνα με τις προδιαγραφές του υπολογιστή μας και του λειτουργικού συστήματος που χρησιμοποιούμε.
2. Αφου εγκαταστήσουμε το πακέτο, ανοίγουμε την εφαρμογή και ενεργοποιούμε τα actions που απευθύνονται στο Apache και στο MySQL, όπως φαίνεται στην παρακάτω εικόνα.



3. Κατεβάζουμε τα αρχεία που βρίσκονται εντός του zip ή repository και τρέχουμε στο phpMyAdmin αυστηρά με τη σειρά τα sql αρχεία Create\_schema\_Project6 και ύστερα τα dummy δεδομένα, που παράξαμε με faker της Python, το αρχείο ELIDEK\_Database\_of\_Project\_6 και μπαίνουν επιτυχώς με τη σειρά

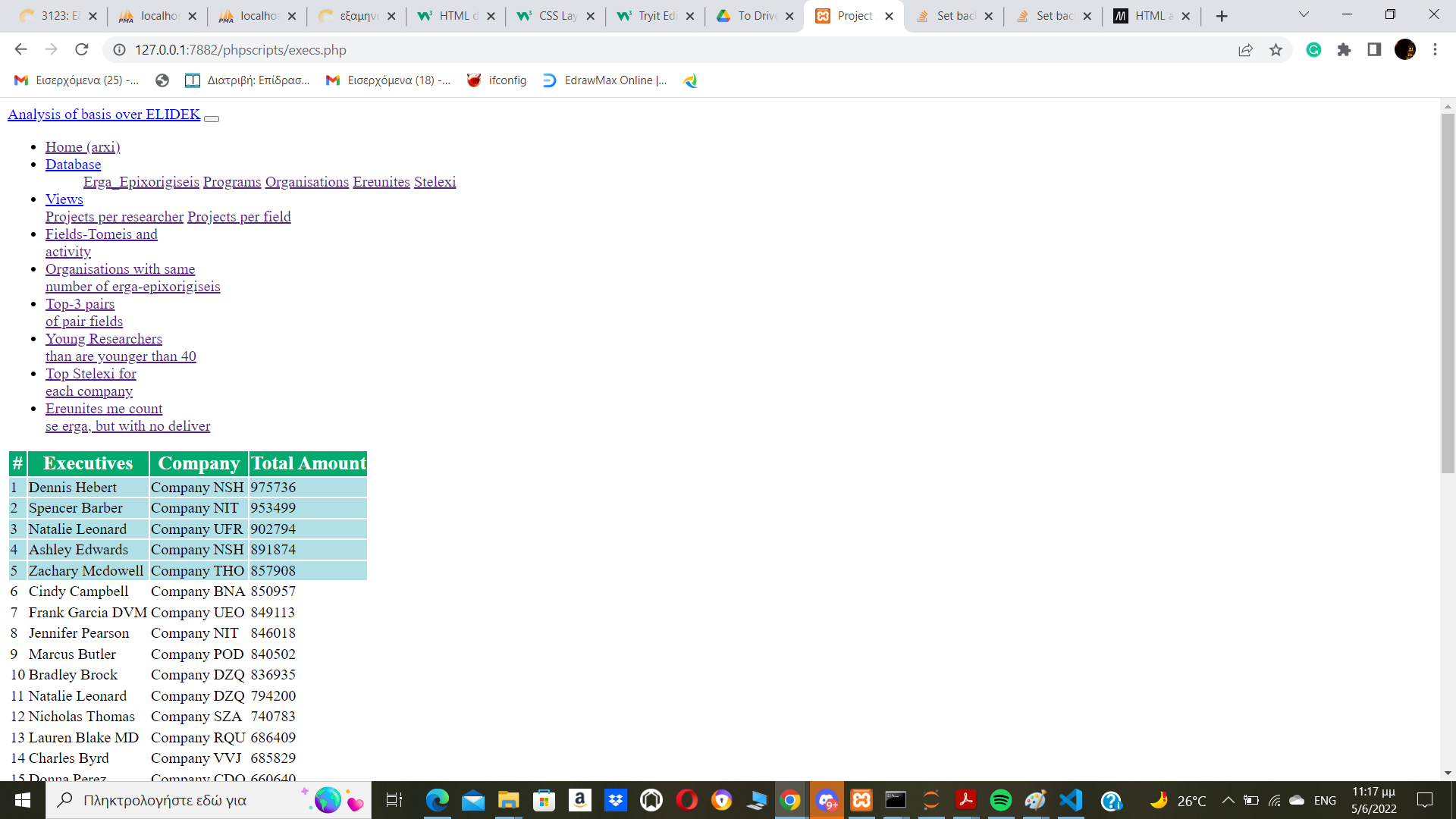
4. Πηγαίνουμε στα αρχεία της εφαρμογής xampp και ανοίγουμε τον φάκελο htdocs και εκεί πέρα θα εγκαταστήσουμε το περιεχόμενο του phpscripts φακέλου, που περιέχει τα php αρχεία, που είναι απαραίτητα για να μπορέσει να τρέξει το αρχείο.

(Παράδειγμα πιθανού path του αρχείου στον υπολογιστή: C:user\xampp\htdocs) αλλά σε οποιαδήποτε και να βρίσκεται, είναι εντός του …/xampp/htdocs και τα εγκαθιστούμε εκεί.

5. Είναι ήδη ενεργοποιημένα τα αρχεία και μπαίνουν κανονικά. Έχοντας ενεργοποιήσει το

6. Αποθηκεύουμε τα αρχεία που κατεβάσαμε στο βήμα 3 μέσα στον φάκελο που μόλις δημιουργήσαμε. Είμαστε έτοιμη και μπορούμε να εκκινήσουμε την διεπαφή

7. Ανοιγούμε έναν browser και πληκτρολογούμε το URL: 127.0.0.1:7882/phpscripts/index.php οπότε και εκκινεί η διεπαφή και είμαστε στο menu. Κανονικά θα πρέπει να εμφανίζεται.

Ενδεικτικό σημείο του GUI μας, αποτελεί το εξής κάτωθι:

Αυτό που θέλουμε να υπογραμμίσουμε είναι ότι 1 view και 1 ερώτημα με τα υπάρχοντα δεδομένα δεν εμφανίζονται παρά την πληθώρα των δεδομένων, αλλά εξηγείται με βάση τα πορίσματα των υπόλοιπων ερωτημάτων (πχ τα μέγιστα πλήθη προτζεκτ που εμφανίζονται είναι 3, αλλά 5 δεν εμφανίζεται για το επόμενο ερώτημα, εκτός εάν είμαστε πιο ελαστικοί και ζητήσουμε 2-3 προτζεκτ) .Τα queries έχουν οριστεί ορθά και με εισαγωγή περισσότερων δεδομένων ή με εφαρμογή πιο ελαστικών κριτηρίων ερωτημάτων, μπορούν να εκφραστούν σε μία τέτοια περίπτωση.