|  |  |
| --- | --- |
|  | ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ |
| ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ |

Αναφορά εξαμηνιαίας εργασίας

Ανάπτυξη βάσης δεδομένων και διεπαφής χρήστη για το ΕΛ.ΙΔ.Ε.Κ.

Ομάδα Project 3

Στα πλαίσια του μαθήματος:

“Βάσεις Δεδομένων”

Ροή Λ

Μπούρδη Ξυδιάς

Ναταλία Κωνσταντίνος

ΑΜ: 03119031 ΑΜ: 03112638

4/06/22

2.1 Σχεσιακό Διάγραμμα και ευρετήρια

2.1.1 Σχεσιακό Διάγραμμα

Diagram

Description automatically generated

Relational Diagram

Η σχεδίαση της βάσης δεδομένων έγινε βάσει της προτεινόμενης λύσης που δόθηκε για το ER διάγραμμα.

Κάθε οντότητα του ER διαγράμματος μετατράπηκε σε σχέση στο σχεσιακό διάγραμμα (π.χ. Οργανισμός, Πρόγραμμα, Έργο). Επιλέχθηκε, ωστόσο, οι τύποι των οργανισμών να μην αποτελούν ξεχωριστές σχέσεις, αλλά να εμπεριέχονται στον πίνακα 'Οργανισμός' και να αναγνωρίζονται με βάση ένα πεδίο 'type' τύπου ENUM με τρεις επιτρεπτές τιμές: Πανεπιστήμιο, Εταιρεία, Ερευνητικό Κέντρο. Οι αντίστοιχοι προϋπολογισμοί αποθηκεύονται σε ένα πεδίο τύπου JSON.

Δικό του πίνακα αποκτά και το πεδίο 'Τηλέφωνα' καθώς μπορεί να περιέχει πάνω από έναν αριθμό τηλεφώνου για κάθε έναν από τους οργανισμούς.

Οι σχέσεις μεταξύ των οντοτήτων με πολλαπλότητα one-to-many και υποχρεωτική συμμετοχή στη πλευρά 'many' ενσωματώθηκαν στους πίνακες των πλευρών 'many'. Για παράδειγμα, η σχέση του ER διαγράμματος 'Διαχείριση' μεταξύ έργου και οργανισμού ενσωματώθηκε στον πίνακα 'Έργο', αφού κάθε έργο διαχειρίζεται αναγκαστικά από έναν και μόνο οργανισμό.

Οι υπόλοιπες σχέσεις στο ER αποκτούν τον δικό τους πίνακα στο σχεσιακό διάγραμμα, αφού η ενσωμάτωση σε κάποιον από τους υπόλοιπους θα είχε ως αποτέλεσμα την απώλεια ή την επανάληψη πληροφορίας.

Τέλος, τα πεδία των οποίων οι τιμές μπορούν να υπολογιστούν από τα υπόλοιπα πεδία ενός πίνακα (derived attributes), δηλαδή η 'Διάρκεια' στο 'Έργο' και η 'Ηλικία' στον 'Ερευνητή', είναι περιττά και δεν περιλαμβάνονται στο σχεσιακό διάγραμμα για την αποφυγή πλεονασμού και περιπλοκότητας στη συντήρηση και ανανέωση των δεδομένων. Υπολογίζονται μόνο όταν χρειάζονται μέσω ερωτημάτων.

Diagram, schematic

Description automatically generated

ER Diagram

2.1.2 Ευρετήρια

Η MySql δημιουργεί αυτόματα ευρετήρια για τα πεδία που είναι primary ή foreign keys σε όλους τους πίνακες και έτσι καλύπτει τις σημαντικότερες ανάγκες της βάσης μας για indexing.

Επιλέξαμε να δημιουργήσουμε μερικά παραπάνω βάσει των ερωτημάτων που θα εκτελούνται συχνότερα με σκοπό την επιτάχυνση τους.

Δεν δημιουργούμε ευρετήρια σε όλα τα πεδία που θα μπορούσαν να επιταχύνουν τα ερωτήματά μας καθώς εκτός από τις αυξημένες απαιτήσεις για χώρο που θα προκύπταν, η εισαγωγή/ενημέρωση/διαγραφή από τις δομές δεδομένων που περιέχουν τα ευρετήρια είναι χρονοβόρα και θα καθυστερούσε την επεξεργασία των δεδομένων της βάσης μας.

* Για την επιτάχυνση του ερωτήματος 3.7 δημιουργούμε index για το ποσό χρηματοδότησης που συναντάμε στο ORDER BY clause για την ταξινόμηση των αποτελεσμάτων βάσει του project.amount.

CREATE INDEX idx\_amount ON project (`amount`);

* Για τα ερωτήματα 3.3, 3.4, 3.6 και 3.8 δημιουργούμε indexes για τις ημερομηνίες έναρξης και λήξης των έργων. Επιταχύνει και το ερώτημα 3.1 καθώς χρησιμοποιείται στα κριτήρια επιλογής των έργων.

CREATE INDEX idx\_end\_date ON project (`end\_date`);

CREATE INDEX idx\_start\_date ON project ('start\_date');

* Δημιουργούμε ένα τελευταίο index για την ημερομηνία γέννησης των ερευνητών που χρησιμοποιείται για τον υπολογισμό της ηλικίας των ερευνητών στο ερώτημα 3.6.

CREATE INDEX idx\_birth\_date ON researcher ('birth\_date');

2.2 DDL και DML Scripts

2.2.1 DDL - Δημιουργία του σχήματος της βάσης

Ως primary key σε κάθε πίνακα χρησιμοποιήθηκε είτε κάποιο πεδίο των οποίων οι τιμές είναι μοναδικές και συνεπώς η κάθε οντότητα μπορεί να αναγνωριστεί πλήρως από αυτό (π.χ. η συντομογραφία ενός οργανισμού), είτε ένα πρόσθετο αναγνωριστικό πεδίο (id) ακεραίου ο οποίος αυτόματα αυξάνεται κατά την εισαγωγή μιας πλειάδας.

Η αναφορική ακεραιότητα εξασφαλίζεται μέσω των foreign keys και των foreign key constraints.

Η αντίδραση ενός πίνακα στην ενημέρωση ή διαγραφή ενός foreign key στο οποίο αναφέρεται εξαρτάται από λογικούς περιορισμούς και τις παραδοχές που κάναμε και ελέγχεται με τις εκφράσεις ON DELETE / ON UPDATE.

Οργανισμός

CREATE TABLE organization (

    abbreviation VARCHAR(15) NOT NULL,

    name VARCHAR(70) NOT NULL,

    type ENUM('uni','co','inst'),

    budget JSON,

    street VARCHAR(50),

    street\_number INT UNSIGNED,

    postal\_code INT(10),

    city VARCHAR(50) NOT NULL,

    PRIMARY KEY (abbreviation)

);

Πρόγραμμα

CREATE TABLE program (

    program\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

    title VARCHAR(90) NOT NULL,

    department VARCHAR(90) NOT NULL,

    PRIMARY KEY (program\_id)

);

Στέλεχος

CREATE TABLE executive (

    executive\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

    first\_name VARCHAR(45) NOT NULL,

    last\_name VARCHAR(45) NOT NULL,

    PRIMARY KEY (executive\_id)

);

Πεδίο

CREATE TABLE field (

    field\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

    field\_name VARCHAR(70),

    PRIMARY KEY (field\_id)

);

Ερευνητής

Θεωρούμε ότι όταν ένας οργανισμός αφαιρείται από την βάση μας, δεν έχει νόημα να διατηρηθούν οι υπάλληλοί του. Οπότε, αν ένας οργανισμός διαγραφεί, διαγράφονται και οι ερευνητές του (ON DELETE CASCADE).

CREATE TABLE researcher (

    researcher\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

    first\_name VARCHAR(45) NOT NULL,

    last\_name VARCHAR(45) NOT NULL,

    gender ENUM('male', 'female') ,

    birth\_date DATE,

    abbreviation VARCHAR(15) NOT NULL,   -- WORKSFOR

    since\_date DATE,

    CONSTRAINT fk\_worksfor\_organization FOREIGN KEY (abbreviation)

        REFERENCES organization (abbreviation) ON DELETE CASCADE ON UPDATE CASCADE,

    PRIMARY KEY (researcher\_id)

);

Έργο / Επιχορήγηση

Προσθέτουμε στον πίνακα των έργων ένα check constraint ώστε να επιτρέπεται η εισαγωγή έργου μόνο όταν το ποσό χρηματοδότησης είναι μεταξύ των ορίων που θέτει το ΕΛ.ΙΔ.Ε.Κ.

Εμποδίζουμε με ON DELETE RESTRICT τη διαγραφή των οντοτήτων με τις οποίες σχετίζεται ένα έργο και θεωρούμε τη σχέση τους απαραίτητη για την ύπαρξη ενός έργου. Σε περίπτωση που ένας οργανισμός διαγραφεί από τη βάση, διαγράφουμε και όλα τα έργα τα οποία διαχειρίζεται.

CREATE TABLE project (

    project\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

    title VARCHAR(255) NOT NULL,

    amount FLOAT(9,2) ,

    summary TEXT DEFAULT NULL,

    start\_date DATE NOT NULL,

    end\_date DATE NOT NULL,

    researcher\_id INT UNSIGNED NOT NULL,   -- manager

    abbreviation VARCHAR(15) NOT NULL,

    executive\_id INT UNSIGNED NOT NULL,

    program\_id INT UNSIGNED NOT NULL,

    CONSTRAINT check\_amount CHECK (amount <= 1000000 AND amount >= 100000),

    PRIMARY KEY (project\_id),

    -- FOREIGN KEY CONSTRAINTS --

    CONSTRAINT fk\_project\_manager FOREIGN KEY (researcher\_id)

        REFERENCES researcher (researcher\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

    CONSTRAINT fk\_project\_organization FOREIGN KEY (abbreviation)

        REFERENCES organization (abbreviation) ON DELETE CASCADE ON UPDATE CASCADE,

    CONSTRAINT fk\_project\_executive FOREIGN KEY (executive\_id)

        REFERENCES executive (executive\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

    CONSTRAINT fk\_project\_program FOREIGN KEY (program\_id)

        REFERENCES program (program\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

Παραδοτέο

Ο πίνακας των παραδοτέων προέκυψε από weak entity set και ως αποτέλεσμα αναγνωρίζεται με τον συνδυασμό του κλειδιού του έργου και του τίτλου του. Η διαγραφή ενός έργου οδηγεί στη διαγραφή και όλων των παραδοτέων του.

CREATE TABLE deliverable (

    deliverable\_id VARCHAR(100) NOT NULL,

    summary TEXT DEFAULT NULL,

    project\_id INT UNSIGNED NOT NULL,

    CONSTRAINT fk\_project\_deliverable FOREIGN KEY (project\_id)

        REFERENCES project (project\_id) ON DELETE CASCADE ON UPDATE CASCADE,   -- both cascade since it's a weak entity --

    PRIMARY KEY (deliverable\_id, project\_id)

);

Τηλέφωνα Οργανισμών

CREATE TABLE organization\_\_phone (

    phone VARCHAR(20) NOT NULL,

    abbreviation VARCHAR(15) NOT NULL,

    PRIMARY KEY (abbreviation, phone),

    CONSTRAINT fk\_organization\_phone FOREIGN KEY (abbreviation)

        REFERENCES organization (abbreviation) ON DELETE CASCADE ON UPDATE CASCADE

);

Σχέση έργων-πεδίων

Αποτρέπουμε την διαγραφή πεδίου που σχετίζεται με έργα. Διαγράφουμε μία σχέση έργου και πεδίου όταν το έργο διαγραφεί.

CREATE TABLE FieldProject (

    project\_id INT UNSIGNED NOT NULL REFERENCES project (project\_id),

    field\_id INT UNSIGNED NOT NULL REFERENCES field (field\_id),

    PRIMARY KEY (project\_id, field\_id),

    CONSTRAINT fk\_fieldproj\_project FOREIGN KEY (project\_id)

        REFERENCES project (project\_id) ON DELETE CASCADE ON UPDATE CASCADE,

    CONSTRAINT fk\_fieldproj\_field FOREIGN KEY (field\_id)

        REFERENCES field (field\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

Σχέση συμμετοχής ερευνητή σε έργο

CREATE TABLE WorksOn (

    project\_id INT UNSIGNED NOT NULL REFERENCES project (project\_id),

    researcher\_id INT UNSIGNED NOT NULL REFERENCES researcher (researcher\_id),

    PRIMARY KEY (project\_id, researcher\_id),

    CONSTRAINT fk\_workson\_project FOREIGN KEY (project\_id)

        REFERENCES project (project\_id) ON DELETE CASCADE ON UPDATE CASCADE,

    CONSTRAINT fk\_worksfon\_researcher FOREIGN KEY (researcher\_id)

        REFERENCES researcher (researcher\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

Αξιολόγηση

CREATE TABLE evaluates (

    project\_id INT UNSIGNED NOT NULL UNIQUE,

    researcher\_id INT UNSIGNED NOT NULL,

    rating ENUM('A', 'B') NOT NULL,      -- we assume it must have recieved an A or a B to have been funded

    eval\_date DATE NOT NULL,

    PRIMARY KEY (project\_id, researcher\_id),

    CONSTRAINT fk\_evalutes\_project FOREIGN KEY (project\_id)

        REFERENCES project (project\_id) ON DELETE CASCADE ON UPDATE CASCADE,

    CONSTRAINT fk\_evaluates\_researcher FOREIGN KEY (researcher\_id)

        REFERENCES researcher (researcher\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

2.2.2 DDL – Views

Όψη έργα ανά ερευνητή – ερώτημα 3.2

DROP VIEW IF EXISTS `projectresearcher\_vw`;

CREATE VIEW `projectresearcher\_vw` AS

SELECT `p`.`project\_id`, `p`.`title`, `r`.`researcher\_id`, CONCAT(`r`.`first\_name`,' ',`r`.`last\_name`) AS `full\_name`, `o`.`name` `organization`

FROM `project` `p`

JOIN `WorksOn` `w`

ON `w`.`project\_id` = `p`.`project\_id`

JOIN `researcher` `r`

ON `w`.`researcher\_id` = `r`.`researcher\_id`

JOIN `organization` `o`

ON `r`.`abbreviation` = `o`.`abbreviation`;

Όψη αξιολογήσεων των έργων – ερώτημα 3.2

DROP VIEW IF EXISTS `eval\_view`;

CREATE VIEW `eval\_view` AS

SELECT `e`.`rating`, `e`.`eval\_date`, `e`.`researcher\_id`, CONCAT(`r`.`first\_name`, ' ', `r`.`last\_name`) AS eval\_name,

        `p`.`project\_id`, `p`.`title`, `p`.`abbreviation`, `o`.`name`

FROM `evaluates` `e`

INNER JOIN `project` `p` ON `e`.`project\_id` = `p`.`project\_id`

INNER JOIN `researcher` `r` ON `r`.`researcher\_id` = `e`.`researcher\_id`

INNER JOIN `organization` `o` ON `o`.`abbreviation` = `p`.`abbreviation`;

Όψη για την προβολή έργων

DROP VIEW IF EXISTS `project\_view`;

CREATE VIEW `project\_view` AS

SELECT `p`.`project\_id`, `p`.`title`, `p`.`amount`, `p`.`start\_date`, `p`.`end\_date`,

YEAR(`p`.`end\_date`) - YEAR(`p`.`start\_date`) - (DATE\_FORMAT(`p`.`end\_date`, '%m%d') < DATE\_FORMAT(`p`.`start\_date`, '%m%d')) `duration`, `p`.`abbreviation`, `o`.`name` `organization`, `p`.`researcher\_id` `manager\_id`, CONCAT(`r`.`last\_name`, ' ', `r`.`first\_name`) `manager`,

`e`.`executive\_id`, CONCAT(`e`.`last\_name`, ' ', `e`.`first\_name`) `executive\_name`, COUNT(`w`.`researcher\_id`) `researchers`, GROUP\_CONCAT(DISTINCT `field\_id`) `field`

FROM `project` `p`

NATURAL JOIN `executive` `e`

JOIN `researcher` `r`

ON `p`.`researcher\_id` = `r`.`researcher\_id`

JOIN `WorksOn`  `w`

ON `p`.`project\_id` = `w`.`project\_id`

JOIN `FieldProject` `f`

ON `p`.`project\_id` = `f`.`project\_id`

JOIN `organization` `o`

ON `p`.`abbreviation` = `o`.`abbreviation`

GROUP BY `p`.`project\_id`;

2.2.3 DDL – Triggers

Τα παρακάτω δύο triggers εκτελούνται αυτόματα πριν την εισαγωγή και ενημέρωση κάποιου έργου και την αποτρέπουν σε περίπτωση που η διάρκεια του έργου που προσπαθεί να ορίσει ο χρήστης δεν είναι μεταξύ ενός και τεσσάρων ετών.

Trigger before insert on project

DELIMITER $$

CREATE TRIGGER proj\_duration\_check\_insert BEFORE INSERT ON project

    FOR EACH ROW

    BEGIN

    IF NOT(4 >= DATE\_FORMAT(FROM\_DAYS(DATEDIFF(new.end\_date,new.start\_date)), '%Y')+0 >= 1) THEN

    SIGNAL SQLSTATE '45000';

    END IF;

END$$

DELIMITER ;

Trigger before update on project

DELIMITER $$

CREATE TRIGGER proj\_duration\_check\_update BEFORE UPDATE ON project

    FOR EACH ROW

    BEGIN

    IF NOT(4 >= DATE\_FORMAT(FROM\_DAYS(DATEDIFF(new.end\_date,new.start\_date)), '%Y')+0 >= 1) THEN

    SIGNAL SQLSTATE '45000';

    END IF;

END$$

DELIMITER ;

Τα παρακάτω δύο triggers εξασφαλίζουν ότι ο ερευνητής που αξιολογεί ένα έργο δεν ανήκει στο δυναμικό του οργανισμού που διαχειρίζεται το έργο.

Trigger before insert on evaluates

DELIMITER $$

CREATE TRIGGER eval\_insert BEFORE INSERT ON evaluates

    FOR EACH ROW

    BEGIN

    IF (SELECT abbreviation  FROM researcher where researcher\_id = new.researcher\_id) =  (SELECT abbreviation FROM project where project\_id = new.project\_id) THEN

    SIGNAL SQLSTATE '45000';

    END IF;

END$$

DELIMITER ;

Trigger before update on evaluates

DELIMITER $$

CREATE TRIGGER eval\_update BEFORE UPDATE ON evaluates

    FOR EACH ROW

    BEGIN

    IF (SELECT abbreviation  FROM researcher where researcher\_id = new.researcher\_id) =  (SELECT abbreviation FROM project where project\_id = new.project\_id) THEN

    SIGNAL SQLSTATE '45000';

    END IF;

END$$

DELIMITER ;

Τα επόμενα δύο triggers εξασφαλίζουν ότι οι ερευνητές που δουλεύουν σε ένα έργο ανήκουν στο δυναμικό του οργανισμού στο οποίο ανήκει το έργο.

Trigger before insert on WorksOn

DELIMITER $$

CREATE TRIGGER workson\_insert BEFORE INSERT ON WorksOn

    FOR EACH ROW

    BEGIN

    IF (SELECT abbreviation  FROM researcher where researcher\_id = new.researcher\_id) <> (SELECT abbreviation FROM project where project\_id = new.project\_id) THEN

    SIGNAL SQLSTATE '45000';

    END IF;

END$$

DELIMITER ;

Trigger before update on WorksOn

DELIMITER $$

CREATE TRIGGER workson\_update BEFORE UPDATE ON WorksOn

    FOR EACH ROW

    BEGIN

    IF (SELECT abbreviation  FROM researcher where researcher\_id = new.researcher\_id) <>  (SELECT abbreviation FROM project where project\_id = new.project\_id) THEN

    SIGNAL SQLSTATE '45000';

    END IF;

END$$

DELIMITER ;

2.2.3 DML – Ερωτήματα 3.1 έως 3.8

3.1

Το ερώτημα για τα προγράμματα

SELECT `program\_id`, `title`, `department`

FROM `program`

WHERE `program\_id` = '$edit\_id'

Το ερώτημα για τα προγράμματα

SELECT `project\_id`,`title`,`amount`,DATE\_FORMAT(`start\_date`, '%d/%m/%Y') `start\_date`,DATE\_FORMAT(`end\_date`, '%d/%m/%Y') `end\_date`,`duration`,`organization`,`manager`,`executive\_name`,`researchers`

FROM `project\_view`

WHERE $filter

3.2

Θα πρέπει ακόμα ο χρήστης να μπορεί να δει δύο όψεις (όψεις του σχεσιακού μοντέλου), μία με έργα/επιχορηγήσεις ανά ερευνητή και μία της επιλογής σας.

Τα ερωτήματα για τα έργα ανά ερευνητή

SELECT DISTINCT `researcher\_id`, `full\_name`, `organization`

FROM `projectresearcher\_vw`

SELECT `project\_id`, `title`

FROM `projectresearcher\_vw`

WHERE `researcher\_id` = $edit\_id;

Το ερωτήμα για τις αξιολογήσεις

SELECT DISTINCT `project\_id`, `title`, `name`, `eval\_name`,  DATE\_FORMAT(`eval\_date`, '%d/%m/%Y') `eval\_date`, `rating`

FROM `eval\_view`

3.3

Δεδομένου ότι ένα συγκεκριμένο ερευνητικό πεδίο απέκτησε ιδιαίτερο ενδιαφέρον, ποια έργα χρηματοδοτούνται σε αυτό το πεδίο και ποιοι ερευνητές ασχολούνται με αυτό το πεδίο το τελευταίο έτος;

SELECT p.title, DATE\_FORMAT(p.start\_date, '%d/%m/%Y') `start\_date`, DATE\_FORMAT(p.end\_date, '%d/%m/%Y') `end\_date` FROM project p

INNER JOIN FieldProject f ON f.project\_id = p.project\_id

WHERE f.field\_id = $field and p.end\_date > curdate();

SELECT DISTINCT concat(R.first\_name,' ',R.last\_name) AS full\_name

FROM researcher R

INNER JOIN WorksOn w ON w.researcher\_id = R.researcher\_id

INNER JOIN project p ON p.project\_id = w.project\_id

WHERE p.project\_id IN (

SELECT p.project\_id FROM project p

INNER JOIN FieldProject f ON f.project\_id = p.project\_id

WHERE f.field\_id = $field and p.end\_date > curdate());

3.4

Ποιοι οργανισμοί έχουν λάβει τον ίδιο αριθμό έργων σε διάστημα δύο συνεχόμενων ετών, με τουλάχιστον 10 έργα ετησίως;

SELECT `ab`, `org\_name`, `proj\_count`, `proj\_count2`, concat(`year1`, ' - ', `year2`) AS `span` FROM

(SELECT o.`abbreviation` as `ab`, `o`.`name` as `org\_name`, count(\*) AS `proj\_count`, year(`p`.`start\_date`) AS `year1`

 FROM `organization` `o`

INNER JOIN `project` `p` ON `o`.`abbreviation` = `p`.`abbreviation`

GROUP BY `o`.`abbreviation`, year(`p`.`start\_date`)

) `t`,

(SELECT `o`.`abbreviation`, `o`.`name`, count(\*) AS `proj\_count2`, year(`p`.`start\_date`) AS `year2`

FROM `organization` o

INNER JOIN `project` `p` ON `o`.`abbreviation` = `p`.`abbreviation`

GROUP BY `o`.`abbreviation`, year(`p`.`start\_date`)

) `t2`

WHERE `t2`.`abbreviation` = `ab` AND `year2` - `year1` = 1

HAVING `proj\_count2` = `proj\_count` AND `proj\_count` >= 10

3.5

Πολλά έργα/επιχορηγήσεις είναι διεπιστημονικά. Ανάμεσα σε ζεύγη που είναι κοινά στα έργα, βρείτε τα 3 κορυφαία ζεύγη που εμφανίστηκαν σε έργα.

SELECT f1.field\_name as field\_1, f2.field\_name as field\_2, count

FROM

(SELECT fp1.field\_id as field1 , fp2.field\_id as field2, COUNT(\*) AS count, fp1.project\_id as proj

FROM fieldproject fp1

INNER JOIN fieldproject fp2 ON

fp1.project\_id  = fp2.project\_id AND fp1.field\_id <> fp2.field\_id

group by fp1.field\_id, fp2.field\_id ) `t`

INNER JOIN field f1 ON field1 = f1.field\_id

INNER JOIN field f2 ON field2 = f2.field\_id

GROUP BY proj

ORDER BY count DESC

LIMIT 3;

3.6

Βρείτε τους νέους ερευνητές (ηλικία < 40 ετών) που εργάζονται στα περισσότερα ενεργά έργα και τον αριθμό των έργων που εργάζονται.

SELECT r.researcher\_id, r.first\_name, r.last\_name, COUNT(\*) as count , DATE\_FORMAT(FROM\_DAYS(DATEDIFF(now(), r.birth\_date)), '%Y')+0  AS age

FROM researcher r

INNER JOIN WorksOn w ON w.researcher\_id = r.researcher\_id

INNER JOIN project p ON p.project\_id = w.project\_id WHERE p.end\_date > curdate()

GROUP BY r.researcher\_id

HAVING age < 40

ORDER BY COUNT(\*) DESC;

3.7

Βρείτε τα top-5 στελέχη που δουλεύουν για το ΕΛ.ΙΔ.Ε.Κ. και έχουν δώσει το μεγαλύτερο ποσό χρηματοδοτήσεων σε μια εταιρεία.

SELECT \* FROM (

SELECT e.executive\_id AS id, concat(e.first\_name,' ',e.last\_name) AS full\_name, o.name, p.amount

FROM executive e

NATURAL JOIN project p

NATURAL JOIN organization o

WHERE o.type = 'co'

ORDER BY p.amount DESC

LIMIT 5

) `t`

GROUP BY `id`

ORDER BY amount DESC;

3.8

Βρείτε τους ερευνητές που εργάζονται σε 5 ή περισσότερα έργα που δεν έχουν παραδοτέα (όνομα ερευνητή και αριθμός έργων).

SELECT r.researcher\_id, r.first\_name, r.last\_name, COUNT(\*) `count`

FROM researcher r

INNER JOIN WorksOn w ON r.researcher\_id = w.researcher\_id

INNER JOIN project p ON w.project\_id = p.project\_id

WHERE w.project\_id NOT IN (SELECT project\_id FROM deliverable)  AND p.end\_date > curdate()

GROUP BY r.researcher\_id

HAVING COUNT(\*) > 4

ORDER BY COUNT(\*) DESC;

2.3 Βήματα εγκατάστασης της εφαρμογής

Απαιτήσεις εφαρμογής:

* MySQL (MariaDB)
* Apache
* PHP

Τα παραπάνω μπορούν να εγκατασταθούν ξεχωριστά ή ως μέρος ενός stack.

Προτεινόμενα βήματα εγκατάστασης για λειτουργικό συστημά windows με χρήση Shell:

1. Εγκαταστήστε τη κατάλληλη διανομή [XAMPP](https://www.apachefriends.org/download.html) για το σύστημά σας.
2. Από το XAMPP Control Panel (με προνόμια διαχειριστή/sudo), ξεκινήστε τις υπηρεσίες MySQL και Apache.
3. Πατήστε το κουμπί Explorer για να μεταφερθείτε στον φάκελο του XAMPP και αντιγράψτε το αρχείο [quick-set-up.sql](https://github.com/nbourdi/DatabasesProject/blob/main/quick-set-up.sql) στο \xampp\mysql\bin.
4. Ανοίξτε το Shell από το Control Panel και εκτελέστε τις παρακάτω εντολές

cd mysql\bin

mysql -u <user> -p elidek < quick-set-up.sql

Εναλλακτικός τρόπος δημιουργίας της βάσης και των εγγραφών μέσω phpMyAdmin

1. Τα βήματα 1 και 2 όπως στον προηγούμενο τρόπο
2. Όταν ξεκινήσουν οι υπηρεσίες του XAMPP αν πατηθεί το κουμπί ‘Admin’ στη γραμμή που αναφέρεται στον Apache μας οδηγεί στη σελίδα dashboard και από εκεί από το μενού πάνω δεξιά πατώντας το κουμπί με την ένδειξη phpMyAdmin μας οδηγεί στο περιβάλλον διαχείρισης της βάσης.
3. Εκεί πηγαίνοντας στην καρτέλα SQL και αντιγράφοντας το περιεχόμενο του αρχείου quick-set-up.sql δημιουργείται η βάση και οι εγγραφές.

Σημείωση:

Σε λειτουργικό σύστημα Mac ή Linux τα βήματα εγκατάστασης έχουν μικρές διαφοροποιήσεις στη προβολή των διαθέσιμων επιλογών. Επίσης η εγκατάσταση μπορεί να γίνει χρησιμοποιώντας κάποιον php server και βάση MySQL – Maria-db χωρίς τη χρήση της εφαρμογής XAMPP που προαναφέρθηκε.

Πλοήγηση στην εφαρμογή

Για να προβληθεί η εφαρμογή θα πρέπει να υπάρχει το repo από το github φάκελο htdocs, ο οποίος βρίσκεται στα αρχεία του XAMPP. Αυτό μπορεί να γίνει είτε κατεβάζοντας το zip αρχείο και κάνοντας extract το φάκελο DatabasesProject εκεί, είτε αν έχουμε εγκατεστημένο το git και πάμε με το τερματικό στο φάκελο htdocs και κάνουμε clone το project, χρησιμοποιώντας την εντολή:

git clone <https://github.com/nbourdi/DatabasesProject>

Στη συνέχεια γράφοντας στο browser το σύνδεσμο

<http://localhost/DatabasesProject>

θα εμφανιστεί η αρχική σελίδα του project.

2.4 [Σύνδεσμος για το git repo της εφαρμογής](https://github.com/nbourdi/DatabasesProject)