

# ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

## ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ 2021-2022



### Ομάδα 14

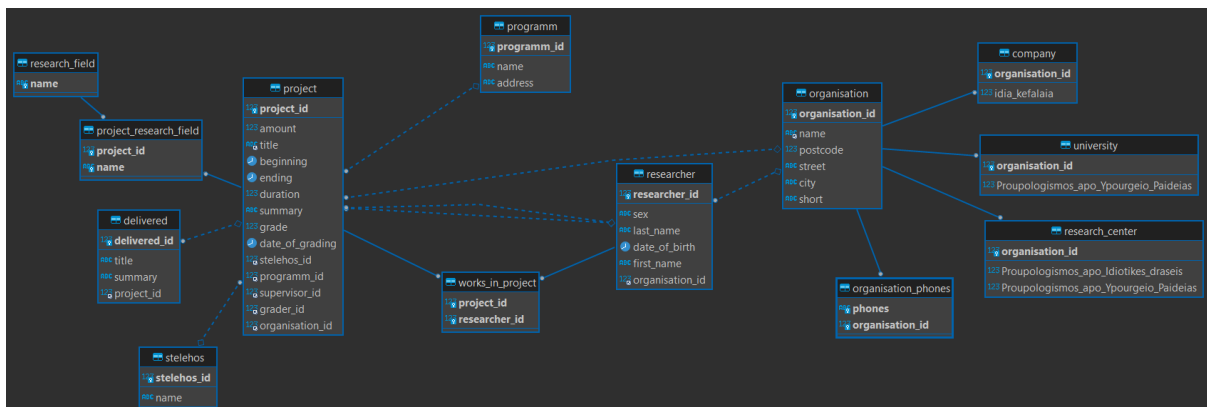
- ★ Κίσης Θεόδωρος-Ιωάννης
- ★ Ντόντορος Ηλίας
- ★ Χατζηλία Ηλιάννα

AM : 03119642  
AM : 03119206  
AM : 03119155

Η ζητούμενη εφαρμογή αναπτύχθηκε με στόχο την υλοποίηση ενός συστήματος αποθήκευσης, διαχείρισης και ανάλυσης των πληροφοριών που συγκεντρώνονται από το ίδρυμα ΕΛΙΔΕΚ .

(a)

### RELATIONAL diagram



Από το ER-Diagram δημιουργήσαμε το παραπάνω σχεσιακό διάγραμμα με τον εξής τρόπο:

- Δημιουργήσαμε ένα table για κάθε entity.
- Δημιουργήσαμε ένα table για κάθε M:N σχέση (Works\_in\_Project, Project\_Research\_Field)
- Κάθε 1:N σχέση έχει προστεθεί σαν attribute στην πλευρά 1 της σχέσης

(b)

Για τη δημιουργία της βάσης:

```
DROP SCHEMA IF EXISTS elidek;  
CREATE SCHEMA elidek;  
use elidek;
```

Query για τη δημιουργία του πίνακα Stelehos:

```
CREATE TABLE Stelehos  
(  
    name VARCHAR(45) NOT NULL,  
    stelehos_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    PRIMARY KEY (stelehos_id)  
);
```

Εδώ το stelehos\_id είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για κάθε στέλεχος.

Query για τη δημιουργία του πίνακα Research\_Field:

```
CREATE TABLE Research_Field  
(  
    name VARCHAR(45) NOT NULL,  
    PRIMARY KEY (name)  
);
```

Εδώ το name είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για κάθε επιστημονικό πεδίο .

Query για τη δημιουργία του πίνακα Programm:

```
CREATE TABLE Programm  
(  
    name VARCHAR(45) NOT NULL,  
    address VARCHAR(45) NOT NULL,
```

```
programm_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (programm_id)  
);
```

Εδώ το programm\_id είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για πρόγραμμα .

Query για τη δημιουργία του πίνακα Organisation:

```
CREATE TABLE Organisation  
(  
    name VARCHAR(45) NOT NULL,  
    postcode INT DEFAULT NULL,  
    street VARCHAR(45) DEFAULT NULL,  
    city VARCHAR(45) NOT NULL,  
    short VARCHAR(15) DEFAULT NULL,  
    organisation_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    PRIMARY KEY (organisation_id),  
    KEY idx_name (name)  
);  
  
alter table Organisation add constraint check(postcode > 0);
```

Εδώ το organisation\_id είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για οργανισμό. Στον πίνακα Organisation προσθέτω ένα constraint όπου πρέπει ο ταχυδρομικός κώδικας να είναι μεγαλύτερος του 0. Έχει οριστεί 1 ευρετήριο, για το name, το οποίο (όπως όλα τα indexes) ορίστηκε με γνώμονα την ελαχιστοποίηση του χρόνου που απαιτείται για την υλοποίηση των queries.

Query για τη δημιουργία του πίνακα Company:

```
CREATE TABLE Company  
(  
    idia_kefalaia INT NOT NULL,  
    organisation_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    PRIMARY KEY (organisation_id),  
    CONSTRAINT `fk_company_organisation` FOREIGN KEY (organisation_id)  
REFERENCES Organisation(organisation_id) ON DELETE CASCADE ON UPDATE  
CASCADE  
);  
  
alter table Company add constraint check(idia_kefalaia > 0);
```

Εδώ το organisation\_id είναι Primary Key και foreign key διότι ένας οργανισμός ανήκει σε μία από τις 3 κατηγορίες: α) Πανεπιστήμια β) Ερευνητικά Κέντρα και γ) **εταιρείες** . Επιπλέον ,στον πίνακα Company προσθέτω ένα constraint όπου πρέπει τα ίδια κεφάλαια να είναι μεγαλύτερα του 0.

Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD, και συγκεκριμένα στο update και στο delete κομμάτι, για οργανισμούς που μπορεί να πραγματοποιήσει ένας χρήστης .

Query για τη δημιουργία του πίνακα University:

```
CREATE TABLE University
(
  Proupologismos_apo_Ypourgeio_Paideias INT NOT NULL,
  organisation_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (organisation_id),
  CONSTRAINT `fk_university_organisation` FOREIGN KEY (organisation_id)
REFERENCES Organisation(organisation_id) ON DELETE CASCADE ON UPDATE
CASCADE
);

alter table University add constraint
check(Proupologismos_apo_Ypourgeio_Paideias > 0);
```

Εδώ το organisation\_id είναι Primary Key και foreign key διότι ένας οργανισμός ανήκει σε μία από τις 3 κατηγορίες: α) **Πανεπιστήμια** β) Ερευνητικά Κέντρα και γ) εταιρείες . Επιπλέον ,στον πίνακα University προσθέτω ένα constraint όπου πρέπει ο προϋπολογισμός από το Υπ. Παιδείας να είναι μεγαλύτερος του 0.

Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD,και συγκεκριμένα το update και το delete κομμάτι ,για οργανισμούς που μπορεί να πραγματοποιήσει ένας χρήστης .

Query για τη δημιουργία του πίνακα Research\_Center:

```
CREATE TABLE Research_Center
(
  Proupologismos_apo_Idiotikes_drasesis INT NOT NULL,
  Proupologismos_apo_Ypourgeio_Paideias INT NOT NULL,
  organisation_id SMALLINT UNSIGNED NOT NULL,
```

```

PRIMARY KEY (organisation_id),
CONSTRAINT `fk_research_center_organisation` FOREIGN KEY
(organisation_id) REFERENCES Organisation(organisation_id) ON DELETE
CASCADE ON UPDATE CASCADE
);

alter table Research_Center add constraint
check(Proupologismos_apo_Ypourgeio_Paideias > 0);
alter table Research_Center add constraint
check(Proupologismos_apo_Idiotikes_draseis > 0);

```

Εδώ το organisation\_id είναι Primary Key και foreign key διότι ένας οργανισμός ανήκει σε μία από τις 3 κατηγορίες: α) Πανεπιστήμια β) **Ερευνητικά Κέντρα** και γ) εταιρείες . Επιπλέον ,στον πίνακα University προσθέτω ένα constraint όπου πρέπει ο προϋπολογισμός από το Υπ. Παιδείας και ο προϋπολογισμός από ιδιωτικές δράσεις να είναι μεγαλύτερος του 0. Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD,και συγκεκριμένα το update και το delete κομμάτι , για οργανισμούς που μπορεί να πραγματοποιήσει ένας χρήστης .

Query για τη δημιουργία του πίνακα Organisation\_Phones:

```

CREATE TABLE Organisation_Phones
(
phones VARCHAR(20) NOT NULL,
organisation_id SMALLINT unsigned NOT NULL,
PRIMARY KEY (phones, organisation_id),
CONSTRAINT `fk_phone_organisation` FOREIGN KEY (organisation_id)
REFERENCES Organisation(organisation_id) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

Εδώ το organisation\_id και το phones είναι Primary Keys . Όσον αφορά το phones είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό . Όσον αφορά το Organisation\_id από την στιγμή που αναφερόμαστε στο/στα τηλέφωνα επικοινωνίας που έχει ο κάθε οργανισμός έχω ως Primary key το Organisation\_id .

Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD,και συγκεκριμένα το update και το delete κομμάτι ,για οργανισμούς που μπορεί να πραγματοποιήσει ένας χρήστης .

Query για τη δημιουργία του πίνακα Researcher:

```

CREATE TABLE Researcher
(
sex VARCHAR(10) NOT NULL,
last_name VARCHAR(45) NOT NULL,

```

```

date_of_birth date NOT NULL,
first_name VARCHAR(45) NOT NULL,
researcher_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT NOT NULL,
organisation_id SMALLINT unsigned NULL,
PRIMARY KEY (researcher_id),
KEY idx_fk_organisation_id (organisation_id),
KEY idx_date_of_birth (date_of_birth),
CONSTRAINT `fk_researcher_organisation` FOREIGN KEY (organisation_id)
REFERENCES Organisation(organisation_id) ON DELETE SET NULL ON UPDATE
CASCADE
);

```

Εδώ το researcher\_id είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για κάθε ερευνητή . Επίσης, το organisation\_ID είναι foreign key στον πίνακα Researcher αφού αναφερόμαστε σε ερευνητές που εργάζονται σε έργα και οι οργανισμοί διαχειρίζονται έργα.

Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD ,και συγκεκριμένα το update και το delete κομμάτι ,για οργανισμούς που μπορεί να πραγματοποιήσει ένας χρήστης .

Έχει οριστεί 1 ευρετήριο για το foreign key organisation\_id κι ένα ευρετήριο για την ημερομηνία γέννησης κάθε ερευνητή με σκοπό να βελτιστοποιήσουμε τον τρόπο που θα υλοποιηθούν κάποια queries (πώς θα βρούμε τους ερευνητές που είναι κάτω από 40)

Query για τη δημιουργία του πίνακα Project:

```

CREATE TABLE Project
(
    amount INT NOT NULL,
    title VARCHAR(45) NOT NULL,
    beginning date NOT NULL,
    ending date NOT NULL,
    duration SMALLINT NOT NULL,
    summary VARCHAR(200) NOT NULL,
    project_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    grade INT NOT NULL,
    date_of_grading date NOT NULL,
    stelehos_id SMALLINT UNSIGNED NULL,
    programm_id SMALLINT unsigned NULL,
    supervisor_id SMALLINT UNSIGNED NULL,
    grader_id SMALLINT UNSIGNED NULL,
    organisation_id SMALLINT unsigned NULL,
    PRIMARY KEY (project_id),
    KEY idx_fk_stelehos_id (stelehos_id),
    KEY idx_fk_programm_id (programm_id),

```

```

KEY idx_fk_supervisor_id (supervisor_id),
KEY idx_fk_grader_id (grader_id),
KEY idx_fk_organisation_id (organisation_id),
KEY idx_title (title),
KEY idx_beginning (beginning),
KEY idx_ending (ending),
KEY idx_duration (duration),
CONSTRAINT `fk_project_stelehos` FOREIGN KEY (stelehos_id) REFERENCES
Stelehos(stelehos_id) ON DELETE SET NULL ON UPDATE cascade,
CONSTRAINT `fk_project_programm` FOREIGN KEY (programm_id) REFERENCES
Programm(programm_id) ON DELETE SET NULL ON UPDATE cascade,
CONSTRAINT `fk_project_supervisor` FOREIGN KEY (supervisor_id)
REFERENCES Researcher(researcher_id) ON DELETE SET NULL ON UPDATE
cascade,
CONSTRAINT `fk_project_grader` FOREIGN KEY (grader_id) REFERENCES
Researcher(researcher_id) ON DELETE SET NULL ON UPDATE cascade,
CONSTRAINT `fk_project_organisation` FOREIGN KEY (organisation_id)
REFERENCES Organisation(organisation_id) ON DELETE SET NULL ON UPDATE
CASCADE
);

alter table Project add constraint check(amount between 100000 and
1000000);

alter table Project add constraint check(beginning < ending);

alter table Project add constraint check(date_of_grading < beginning);

alter table Project add constraint check(grade between 0 and 10);

alter table Project add constraint check(duration between 12 and 48);

```

Εδώ το project\_id είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για κάθε έργο . Επίσης στα έργα έχουμε 5 foreign keys :

(i) το stelehos\_id είναι foreign key στον πίνακα Project αφού αναφερόμαστε σε στελέχη που δουλεύουν για το ΕΛ.ΙΔ.Ε.Κ. και διαχειρίζονται τα έργα.

(ii) το programm\_id είναι foreign key στον πίνακα Project αφού το κάθε έργο λαμβάνει χρηματοδότηση από ένα πρόγραμμα.

(iii) το supervisor\_id είναι foreign key στον πίνακα Project αφού κάθε έργο έχει έναν ερευνητή που είναι ο επιστημονικός υπεύθυνος του έργου .

(iv) το grader\_id είναι foreign key στον πίνακα Project αφού το κάθε έργο έχει αξιολογηθεί από έναν ερευνητή που δεν ανήκει στο δυναμικό του οργανισμού που συμμετέχει στην πρόταση.

(v) το organisation\_id είναι foreign key στον πίνακα Project αφού οι οργανισμοί διαχειρίζονται έργα.

Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD ,και συγκεκριμένα το update και το delete κομμάτι ,για έργα που μπορεί να πραγματοποιήσει ένας χρήστης .

Επιπλέον ,στον πίνακα Project υπάρχουν τα παρακάτω constraints :

(i) Τα ποσά χρηματοδότησης των έργων να κυμαίνονται από 100.000€ έως 1.000.000€.

(ii) Η ημερομηνία έναρξης του έργου να προηγείται της ημερομηνίας ολοκλήρωσης .

(iii) Η ημερομηνία αξιολόγησης του έργου να προηγείται της ημερομηνίας έναρξης του .

(iv) Ο βαθμός αξιολόγησης του έργου να κυμαίνεται από 0 μέχρι και 10.

(v) Η διάρκεια ενός έργου πρέπει να είναι από 1 έως 4 χρόνια.

Έχουν οριστεί 9 ευρετήρια, ένα για κάθε foreign key και τα υπόλοιπα για να βελτιστοποιείται η υλοποίηση ορισμένων queries.

#### Query για τη δημιουργία του πίνακα Delivered :

```
CREATE TABLE Delivered
(
  title VARCHAR(45) NOT NULL,
  summary VARCHAR(45) NOT NULL,
  delivered_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  project_id SMALLINT unsigned NOT NULL ,
  PRIMARY KEY (delivered_id),
  KEY idx_fk_project_id (project_id),
  CONSTRAINT `fk_delivered_project` FOREIGN KEY (project_id) REFERENCES
Project(project_id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Εδώ το delivered\_id είναι Primary Key διότι είναι το attribute το οποίο είναι μοναδικό και ξεχωριστό για κάθε παραδοτέο . Το project\_id είναι foreign key στον πίνακα Delivered αφού το κάθε έργο ενδέχεται να έχει παραδοτέα . Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD ,και συγκεκριμένα το update και το delete κομμάτι ,για έργα και επιστημονικά πεδία που μπορεί να πραγματοποιήσει ένας χρήστης . Έχει οριστεί 1 ερευτήριο για το foreign key project\_id .

#### Query για τη δημιουργία του πίνακα Project\_Research\_Field :

```
CREATE TABLE Project_Research_Field
(
  project_id SMALLINT UNSIGNED NOT NULL ,
  name VARCHAR(45) NOT NULL,
  PRIMARY KEY (project_id, name),
```



```

    CONSTRAINT `fk_pr_research_field_project` FOREIGN KEY (project_id)
REFERENCES Project(project_id) ON DELETE RESTRICT ON UPDATE cascade,
    CONSTRAINT `fk_pr_research_field_research_field` FOREIGN KEY (name)
REFERENCES Research_Field(name) ON DELETE CASCADE ON UPDATE CASCADE
);

```

Εδώ το project\_id και το name είναι Primary Keys αλλά και foreign keys διότι τα επιστημονικά πεδία (τομείς έρευνας) περιγράφουν θεματικές περιοχές των έργων .Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD ,και συγκεκριμένα το update και το delete κομμάτι ,για έργα και επιστημονικά πεδία που μπορεί να πραγματοποιήσει ένας χρήστης

Query για τη δημιουργία του πίνακα Works\_in\_Project :

```

CREATE TABLE Works_in_Project
(
    project_id SMALLINT UNSIGNED NOT NULL ,
    researcher_id SMALLINT UNSIGNED NOT NULL ,
    PRIMARY KEY (project_id, researcher_id),
    CONSTRAINT `fk_works_in_project_project` FOREIGN KEY (project_id)
REFERENCES Project(project_id) ON DELETE RESTRICT ON UPDATE cascade,
    CONSTRAINT `fk_works_in_project_researcher` FOREIGN KEY
(researcher_id) REFERENCES Researcher(researcher_id) ON DELETE CASCADE
ON UPDATE cascade
);

```

Εδώ το project\_id και το researcher\_id είναι Primary Keys αλλά και foreign keys διότι ο κάθε ερευνητής εργάζεται σε ένα (και μόνο) οργανισμό όπου οι οργανισμοί διαχειρίζονται έργα . Επίσης έχουν προστεθεί οι εντολές ON DELETE RESTRICT ON UPDATE CASCADE οι οποίες συμβάλλουν στην λειτουργία CRUD ,και συγκεκριμένα το update και το delete κομμάτι ,για έργα και ερευνητές που μπορεί να πραγματοποιήσει ένας χρήστης .

Κατά τη δημιουργία των πινάκων ορίσαμε ως NOT NULL τα χαρακτηριστικά που θεωρούμε απαραίτητα για την ύπαρξη μιας εντότητας. Θέσαμε ως NULL κάποια χαρακτηριστικά πχ. στο Project με σκοπό να τεθούν NULL σε περίπτωση διαγραφής άλλης οντότητας. Για παράδειγμα σε περίπτωση διαγραφής του αξιολογητή ενός έργου δεν θέλουμε να διαγραφεί και το έργο, οπότε ο grader αυτού του project θα τεθεί NULL.

Τέλος δημιουργήσαμε triggers τα οποία ελέγχουν πριν την εισαγωγή και πριν την ανανέωση δεδομένων αν είναι σωστά ορισμένα.  
 Συγκεκριμένα πρέπει οι ερευνητές που εργάζονται σε ένα έργο να ανήκουν στο δυναμικό του οργανισμού που διαχειρίζεται το έργο ενώ ο αξιολογητής του έργου δεν πρέπει να ανήκει

στον οργανισμό που διαχειρίζεται το έργο. Επιπλέον έχουμε θεωρήσει ότι ο επιστημονικός υπεύθυνος ενός έργου δεν ανήκει στη σχέση Works\_in\_Project για το συγκεκριμένο έργο.

```
DELIMITER $$
create trigger insert_in_project
before insert on Project
for each row
begin
    if(new.supervisor_id not in (select researcher_id
    from researcher
    where organisation_id = new.organisation_id)
    or
    new.grader_id in (select researcher_id
    from researcher
    where organisation_id = new.organisation_id))
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Wrong data input.
Supervisor must work in the organisation that handles the project and
the grader must be from a different organisation.';
    end if;
end $$
```

delimiter ;

```
-----
DELIMITER $$
create trigger update_in_project
before update on Project
for each row
begin
    if(new.supervisor_id not in (select researcher_id
    from researcher
    where organisation_id = new.organisation_id)
    or
    new.grader_id in (select researcher_id
    from researcher
    where organisation_id = new.organisation_id)
    or
    new.supervisor_id not in (select researcher_id
    from works_in_project
    where project_id = new.project_id ))
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Wrong data input.
Supervisor must work in the organisation that handles the project and
the grader must be from a different organisation.';
```

```

        end if;
end $$

delimiter ;
-----

DELIMITER $$
create trigger ins_in_wip
before insert on Works_in_Project
for each row
begin
    if(new.researcher_id in (select p.supervisor_id
    from project p
    inner join works_in_project wip on p.project_id = wip.project_id
    where p.project_id = new.project_id)
    or
    new.researcher_id not in (select r.researcher_id
    from researcher r
    inner join project p2 on r.organisation_id = p2.organisation_id
    where p2.project_id = new.project_id))
    then
        signal sqlstate '45000' set MESSAGE_TEXT = 'Wrong data input.
Researcher must work in the organisation that handles the project and
should not be supervisor of that project.';
    end if;
end $$

delimiter ;
-----

DELIMITER $$
create trigger update_in_wip
before update on Works_in_Project
for each row
begin
    if(new.researcher_id in (select p.supervisor_id
    from project p
    inner join works_in_project wip on p.project_id = wip.project_id
    where p.project_id = new.project_id)
    or
    new.researcher_id not in (select r.researcher_id
    from researcher r
    inner join project p2 on r.organisation_id = p2.organisation_id
    where p2.project_id = new.project_id))
    then

```

```
signal sqlstate '45000' set MESSAGE_TEXT = 'Wrong data input.
Researcher must work in the organisation that handles the project and
should not be supervisor of that project.';
end if;
end $$

delimiter ;
```

(c)

### Τεχνολογία ανάπτυξης εφαρμογής

Για την διαχείριση και ανάπτυξη της βάσης χρησιμοποιήθηκε το MariaDB και για DBMS Tool χρησιμοποιήθηκε το DBeaver. Για το στήσιμο του web server χρησιμοποιείται flask (Python) και για την σύνδεση μεταξύ βάσης και του server χρησιμοποιείται ένας mysql connector python. Για το UI χρησιμοποιήθηκε HTML . Για την επικοινωνία μεταξύ backend και frontend χρησιμοποιούνται μεθοδοι GET και Post.

Version:

- MariaDB
- DBeaver
- Flask 2.0.1
- mysql-connector-python
- requests
- python-dateutil

(d)

### Βήματα εγκατάστασης σε λογισμικό Windows

1. Απαιτείται η εγκατάσταση του MariaDB .
2. Αφού έχει γίνει η εγκατάσταση συνδεόμαστε στον server μέσω ενός DBMS Tool και προτείνουμε την χρήση του DBeaver .
3. Αφού συνδεθούμε επιτυχώς, μπορούμε να πάμε να δημιουργήσουμε όλα τα tables και τις απαραίτητες συνδέσεις μεταξύ τους. Στην εργασία έχουμε επισυνάψει τα SQL Scripts που περιέχουν μέσα τα Scripts για την δημιουργία των παραπάνω.
4. Στην συνέχεια, βάζουμε τα δεδομένα στην βάση. Στην εργασία έχουμε επισυνάψει το SQL script που περιέχει τα data .
5. Αφού έχουν οριστεί οι πίνακες, έχει γίνει εισαγωγή των δεδομένων, έχουν δημιουργηθεί τα απαραίτητα ευρετήρια, προχωράμε στο στήσιμο του περιβάλλοντος πάνω στο οποίο τρέχει η εφαρμογή μας. Εγκαθιστούμε τις απαραίτητες βιβλιοθήκες :

```
pip install mysql-connector-python
pip install Flask
pip install requests
pip install python-dateutil
```

```
pip install dataclasses  
pip install DateTime
```

Αφού τοποθετήσουμε όλα τα αρχεία στους σωστούς φακέλους τρέχουμε την εφαρμογή μας σε κάποιον localhost . Για να μπορούμε να το κάνουμε αυτό, πρέπει να είμαστε στον ίδιο φάκελο που είναι και το αρχείο main.py (website) και εκτελούμε τις εντολές:

---

```
cd website  
python main.py
```

---

➤ Το website θα τρέχει στον localhost που φαίνεται στο terminal .