

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2021-2022, 6ο εξάμηνο, ΣΗΜΜΥ

Βάσεις Δεδομένων

Εξαμηνιαία Εργασία



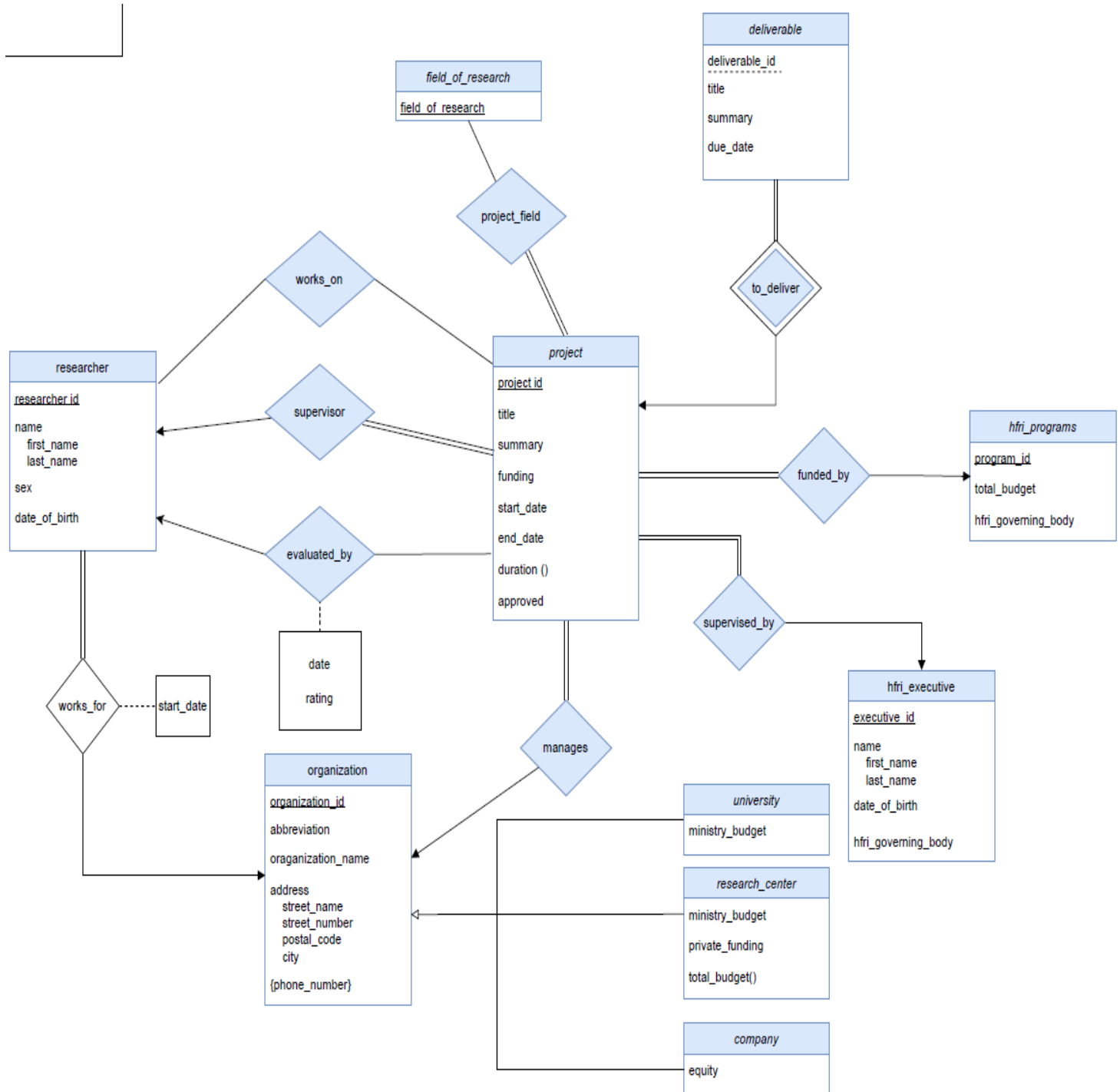
Σπουδαστές
Ομάδα 4

Γιάτσος Κωνσταντίνος (el19622)
Γιαννακάκης Νικόλαος (el10191)
Γεωργαντάς Νικόλαος (el09688)

Github link: <https://github.com/NikYian/DB-project-NTUA-2022-team4>

Μέρος 1

Σύμφωνα με την εκφώνηση τής εργασίας προκύπτει το παρακάτω ER διάγραμμα:



Το παραπάνω ER διάγραμμα αποτελείται από τις παρακάτω **οντότητες (entities)** οι οποίες περιέχουν τις **ιδιότητες** που τις χαρακτηρίζουν και από διάφορες **σχέσεις (relationships)** που συσχετίζουν τις οντότητες μεταξύ τους:

Οντότητα	Ιδιότητες
project	<u>project_id</u> ,title,summary,funding,start_date,end_date,duration()
researcher	<u>researcher_id</u> ,name(first_name,last_name),sex,date_of_birth
organization	<u>organization_id</u> ,abbreviation,organization_name, address(street_name,street_number,postal_code,city), {phone_number}
hfri_programs*	<u>program_id</u> ,total_budget,hfri_governing_body
hfri_executive	<u>executive_id</u> ,name(first_name,last_name),date_of_birth, hfri_governing_body
deliverable	<u>deliverable_id</u> ,title,summary,due_date
field_of_research	<u>field_of_research</u>
university	<u>organization_id</u> ,ministry_budget
research_center	<u>organization_id</u> ,ministry_budget,private_funding
company	<u>organization_id</u> ,equity**

Σχέση	Οντότητες που συσχετίζονται	Ιδιότητες	Χαρακτηρισμός Σχέσης
works_on	researcher,project	-	Πολλά προς πολλά
supervisor	researcher,project	-	Ένα προς πολλά
evaluated_by	researcher,project	date,rating	Ένα προς πολλά
works_for	researcher,organization	start_date	Πολλά προς ένα
manages	organization,project	-	Ένα προς πολλά
project_field	project,field_of_research	-	Πολλά προς πολλά
to_deliver	project,deliverable	-	Ένα προς πολλά
funded_by	project,hfri_programs	-	Πολλά προς ένα
supervised_by	hfri_executive,project	-	Ένα προς πολλά

Παρατηρήσεις - Σχόλια

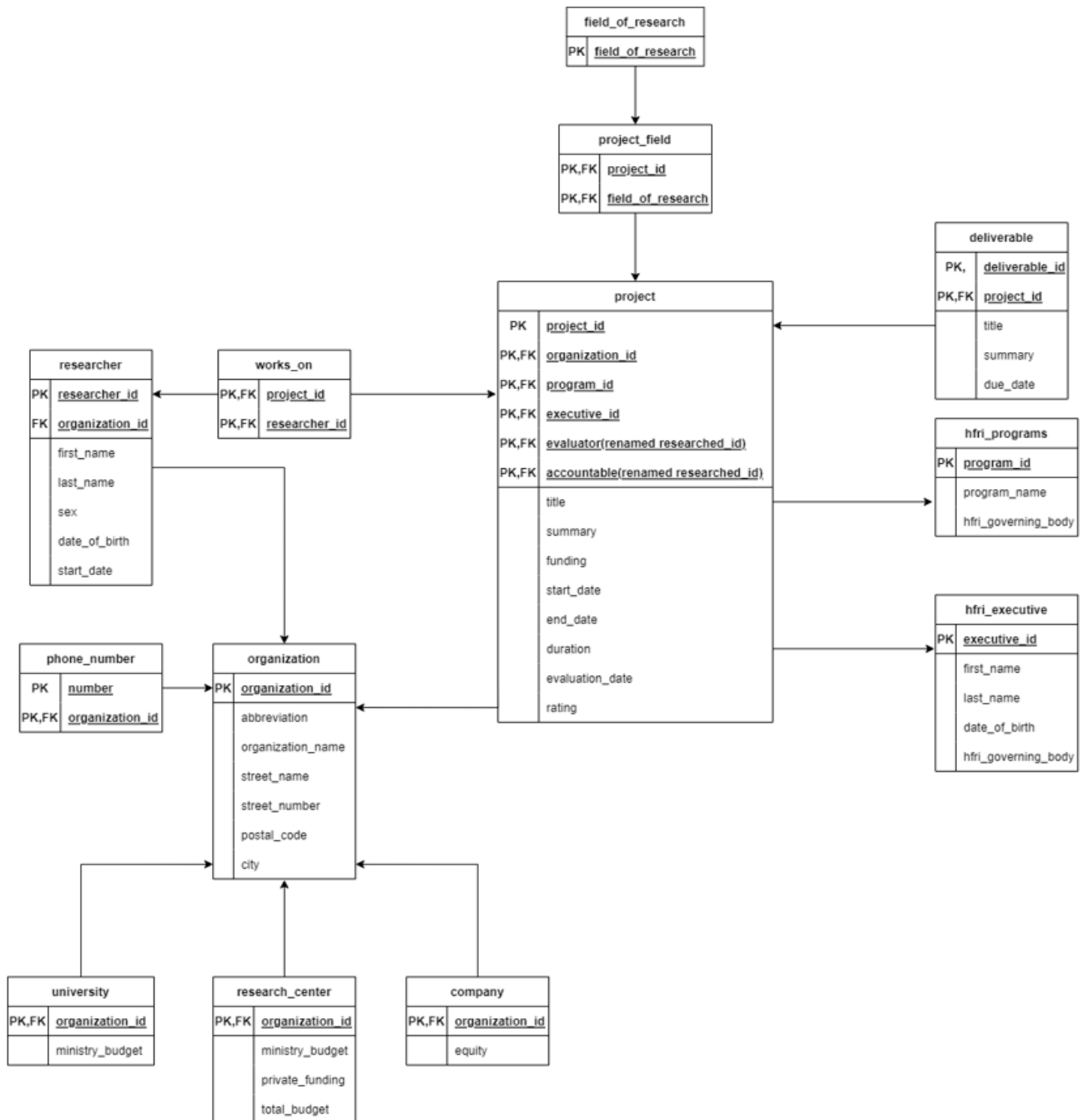
- Η ιδιότητα **duration()** της οντότητας project είναι **παραγόμενη** ιδιότητα από τις start_date, end_date.
- Οι ιδιότητες **name** των οντοτήτων researcher, hfri_executive και **address** της οντότητας organization είναι **σύνθετες** ιδιότητες, δηλαδή μπορούν να χωριστούν σε δευτερεύοντα μέλη, όπως φαίνεται εντός των παρενθέσεων.
- Η ιδιότητα **phone_number** της οντότητας organization είναι **πολλαπλών τιμών** και για αυτό στο ER διάγραμμα εμφανίζεται εντός αγκυλών.
- Η διπλή γραμμή στις σχέσεις μεταξύ των διάφορων οντοτήτων σημαίνει ότι η οντότητα που ενώνεται με διπλή γραμμή έχει **ολική συμμετοχή** στην σχέση, δηλαδή κάθε αντικείμενο της συγκεκριμένης οντότητας συνδέεται με τουλάχιστον ένα αντικείμενο άλλης οντότητας.
- Οι υπογραμμισμένες ιδιότητες σε κάθε οντότητα αποτελούν το **πρωτεύον κλειδί**, είναι δηλαδή η ιδιότητα που χαρακτηρίζει μοναδικά το κάθε αντικείμενο που ανήκει στο σύνολο της οντότητας.
- Η οντότητα **deliverable** αποτελεί **αδύναμο σύνολο οντοτήτων** (weak entity set) διότι δεν υπάρχουν επαρκείς ιδιότητες ώστε να σχηματιστεί ένα πρωτεύον κλειδί. Η σχέση που συνδέει ένα αδύναμο σύνολο απεικονίζεται με ένα διπλό ρόμβο όπως φαίνεται στο ER διάγραμμα. Οι ιδιότητες οι οποίες αποτελούν διαχωριστικά των αντικειμένων σε ένα αδύναμο σύνολο οντοτήτων υπογραμμίζονται με διακεκομμένη γραμμή.
- Οι οντότητες university, research_center, company αποτελούν **εξειδίκευση (specialization)** της οντότητας organization. Αυτού του είδους η εξειδίκευση είναι **ξένη (disjoint)**, καθώς ένα οργανισμός δεν μπορεί να είναι ταυτόχρονα πανεπιστήμιο ή εταιρεία ή ερευνητικό κέντρο και για αυτό απεικονίζεται με ένα βέλος στο ER διάγραμμα. Οι οντότητες αυτές κληρονομούν όλες τις ιδιότητες της υπερκλάσης (organization), και έχουν ξεχωριστά τις δικές του ιδιότητες, που τις χαρακτηρίζουν.
- Μια σχέση μπορεί να χαρακτηριστεί επίσης από ιδιότητες, οι οποίες στο ER διάγραμμα βρίσκονται σε ορθογώνιο πλαίσιο και ενώνονται με διακεκομμένη γραμμή.

* Όπου hfri σημαίνει Hellenic Foundation for Research & Innovation (ΕΛΙΔΕΚ).

** Όπου equity σημαίνει τα ίδια κεφάλαια μιας εταιρείας.

Μέρος 2

Στη συνέχεια, κάνουμε μετατροπή του ER διαγράμματος που δημιουργήθηκε στο Μέρος 1 στο σχεσιακό μοντέλο το σχηματικό διάγραμμα του οποίου φαίνεται παρακάτω:



Από το παραπάνω διάγραμμα μπορούμε να παρατηρήσουμε όλους τους πίνακες που έχουν εισαχθεί στο **σχεσιακό μοντέλο**, τους περιορισμούς **πρωταρχικού κλειδιού** (primary key) για κάθε πίνακα, καθώς και τις σχέσεις που τις συνδέουν τους πίνακες μέσω του **περιορισμού αναφοράς ξένου κλειδιού** (foreign key). Αναλυτικότερα στο αρχείο DDL (Data Definition Language) περιέχονται οι παρακάτω εντολές με τις οποίες φτιάχτηκε το σχήμα της βάσης **hfri_db**:

```
DROP schema IF EXISTS hfri_db;
create schema hfri_db;
use hfri_db;
```

Με την εντολή **DROP schema** διαγράφουμε κάθε φορά τη βάση hfri_db από το σύστημα μας και στη συνέχεια την ξαναδημιουργούμε με την εντολή **create schema**. Στη συνέχεια αναγκάζουμε το σύστημα να χρησιμοποιήσει τη βάση hfri_db που μόλις δημιουργήθηκε με την εντολή **use**, προκειμένου να ανακτήσει δεδομένα από αυτή όταν χρειαστεί.

Στη συνέχεια δημιουργούμε τον κάθε πίνακα (table) που φαίνεται στο παραπάνω σχηματικό διάγραμμα με την εντολή **create table** και εισάγουμε εκεί τους διάφορους περιορισμούς ακεραιότητας, κλειδιών, πεδίου τιμών καθώς και αναφορικής ακεραιότητας, όπως φαίνεται αναλυτικά παρακάτω:

Οντότητα Στέλεχος ΕΛΙΔΕΚ

```
create table hfri_executive (
executive_id int unsigned not null auto_increment,
first_name varchar(20) not null,
last_name varchar(20) not null,
date_of_birth date not null check(year(date_of_birth) < 2000),
hfri_governing_body varchar(20) not null check(hfri_governing_body in ('General
Assembly','Scientific Council','Advisory Commitee','Deputy Directors')),
primary key (executive_id)
);
```

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
executive_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)

first_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 20 (varchar(20)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
last_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 20 (varchar(20)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
date_of_birth	<ul style="list-style-type: none"> • αντικείμενο τύπου date της μορφής YYYY-MM-DD • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
hfri_governing_body	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 20 (varchar(20)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: executive_id (primary key constraint)

Ακεραιότητα πεδίου τιμών	
Εντολή SQL	Σχολιασμός
check(year(date_of_birth) < 2000)	Έχει οριστεί η χρονιά γέννησης του κάθε στελέχους ΕΛΙΔΕΚ να είναι μικρότερη του 2000, δηλαδή ένα στέλεχος για να δουλεύει στο ΕΛΙΔΕΚ πρέπει να έχει ηλικία από 22 χρονών και πάνω.
check(hfri_governing_body in ('General Assembly','Scientific Council','Advisory Committee','Deputy Directors'))	Ένα στέλεχος του ΕΛΙΔΕΚ μπορεί να ανήκει στις εξής διευθύνσεις: <ul style="list-style-type: none"> • Γενική Συνέλευση (General Assembly) • Επιστημονικό Συμβούλιο (Scientific Council) • Διευθυντές (Deputy Directors) • Συμβουλευτική Επιτροπή (Advisory Committee)

Οντότητα Προγράμματα ΕΛΙΔΕΚ

```
create table hfri_programs (
program_id int unsigned not null auto_increment,
program_name varchar(50) not null,
hfri_governing_body varchar(30) not null check(hfri_governing_body in ('General Assembly','Scientific Council','Advisory Committee','Deputy Directors')),
primary key (program_id)
);
```

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
program_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
program_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 50 (varchar(50)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
hfri_governing_body	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 30 (varchar(30)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: program_id (primary key constraint)

Ακεραιότητα πεδίου τιμών	
Εντολή SQL	Σχολιασμός
<pre>check(hfri_governing_body in ('General Assembly','Scientific Council','Advisory Committee','Deputy Directors'))</pre>	<p>Ένα πρόγραμμα του ΕΛΙΔΕΚ μπορεί να ανήκει στις εξής διευθύνσεις:</p> <ul style="list-style-type: none"> • Γενική Συνέλευση (General Assembly) • Επιστημονικό Συμβούλιο (Scientific Council) • Διευθυντές (Deputy Directors) • Συμβουλευτική Επιτροπή (Advisory Committee)

Οντότητα Οργανισμός

```
create table organization_ (
organization_id int unsigned not null auto_increment,
abbreviation varchar(10),
organization_name varchar(100),
street_name varchar(30) not null,
street_number int unsigned,
postal_code varchar(10) not null,
city varchar(50),

primary key (organization_id)
);
```


Σχολιασμός: Η σύνθετη ιδιότητα name του ER διαγράμματος στο σχεσιακό μοντέλο γράφεται με όλα τα δευτερεύοντα μέλη της (first_name, last_name). Ομοίως και η σύνθετη ιδιότητα street με δευτερεύοντα μέλη τα street_name, street_number, postal_code και city.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
organization_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
abbreviation	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 10 (varchar(10)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
organization_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 100 (varchar(100)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
street_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 30 (varchar(30)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
street_number	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
postal_code	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 10 (varchar(10)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
city	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 50 (varchar(50)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: organization_id (primary key constraint)

Οντοτητα Πανεπιστήμιο

```
create table university (
organization_id int unsigned,
ministry_budget double(7,2) not null,
PRIMARY KEY (organization_id),
CONSTRAINT FK_university
```

FOREIGN KEY (organization_id) REFERENCES organization_ (organization_id) on delete restrict on update cascade
);

Σχολιασμός: Λόγω της εξειδίκευσης της οντότητας οργανισμός από την οντότητα πανεπιστήμιο, η τελευταία κληρονομεί όλες τις ιδιότητες της πρώτης και έχει σαν περιορισμό ξένου κλειδιού την ιδιότητα organization_id που την χαρακτηρίζει μοναδικά.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
organization_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
ministry_budget	<ul style="list-style-type: none"> • δεκαδικός (double) με ακέραιο πεδίο 7 ψηφία και δεκαδικό πεδίο 2 ψηφία • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: organization_id (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζεται **περιορισμός ξένου κλειδιού** με το όνομα FK_university με **ξένο κλειδί** το organization_id ,το οποίο **αναφέρεται** στον πίνακα organization_. Επίσης, σε ενδεχόμενη διαγραφή ή ενημέρωση μιας γραμμής του πίνακα, αυτή διαγράφεται ή ενημερώνεται **διαδοχικά** και στον πίνακα organization_ στον οποίο αναφέρεται.

Οντοτητα Ερευνητικό Κέντρο

```
create table research_center (
organization_id int unsigned,
ministry_budget double(7,2) not null,
private_funding double(7,2) not null,
PRIMARY KEY (organization_id),
CONSTRAINT FK_RS
```

```
FOREIGN KEY (organization_id) REFERENCES organization_ (organization_id) on
delete cascade on update cascade
)
```

Σχολιασμός: Λόγω της εξειδίκευσης της οντότητας οργανισμός από την οντότητα ερευνητικό κέντρο, η τελευταία κληρονομεί όλες τις ιδιότητες της πρώτης και έχει σαν περιορισμό ξένου κλειδιού την ιδιότητα organization_id που την χαρακτηρίζει μοναδικά.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
organization_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
ministry_budget	<ul style="list-style-type: none"> • δεκαδικός (double) με ακέραιο πεδίο 7 ψηφία και δεκαδικό πεδίο 2 ψηφία • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
private_funding	<ul style="list-style-type: none"> • δεκαδικός (double) με ακέραιο πεδίο 7 ψηφία και δεκαδικό πεδίο 2 ψηφία • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: organization_id (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζεται **περιορισμός ξένου κλειδιού** με το όνομα FK_RS με **ξένο κλειδί** το organization_id ,το οποίο **αναφέρεται** στον πίνακα organization_. Επίσης, σε ενδεχόμενη διαγραφή ή ενημέρωση μιας γραμμής του πίνακα, αυτή διαγράφεται ή ενημερώνεται **διαδοχικά** και στον πίνακα organization_ στον οποίο αναφέρεται.

Οντοτητα Εταιρεία

```
create table company (
organization_id int unsigned,
equity double(7,2) not null,
PRIMARY KEY (organization_id),
CONSTRAINT FK_company
```

```
FOREIGN KEY (organization_id) REFERENCES organization_ (organization_id) on
delete cascade on update cascade
);
```

Σχολιασμός: Λόγω της εξειδίκευσης της οντότητας οργανισμός από την οντότητα εταιρεία, η τελευταία κληρονομεί όλες τις ιδιότητες της πρώτης και έχει σαν περιορισμό ξένου κλειδιού την ιδιότητα organization_id που την χαρακτηρίζει μοναδικά.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
organization_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
equity	<ul style="list-style-type: none"> • δεκαδικός (double) με ακέραιο πεδίο 7 ψηφία και δεκαδικό πεδίο 2 ψηφία • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: organization_id (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζεται περιορισμός ξένου κλειδιού με το όνομα FK_company με ξένο κλειδί το organization_id ,το οποίο αναφέρεται στον πίνακα organization_. Επίσης, σε ενδεχόμενη διαγραφή ή ενημέρωση μιας γραμμής του πίνακα, αυτή διαγράφεται ή ενημερώνεται **διαδοχικά** και στον πίνακα organization_ στον οποίο αναφέρεται.

Οντότητα Τηλέφωνο

```
create table phone_number(
organization_id int unsigned,
phone_number varchar(30),
PRIMARY KEY (organization_id,phone_number),
CONSTRAINT FK_phone
FOREIGN KEY (organization_id) REFERENCES organization_ (organization_id) on
delete restrict on update cascade
);
```

Σχολιασμός: Επιλέχθηκε ο αριθμός τηλεφώνου μιας εταιρείας να γίνει ξεχωριστός πίνακας λόγω ότι αποτελεί ιδιότητα πολλαπλών τιμών της οντότητας οργανισμός. Κληρονομεί την ιδιότητα organization_id για να φαίνεται σε ποιον οργανισμό αντιστοιχεί το κάθε τηλέφωνο.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
organization_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
phone_number	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 30 (varchar(30)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: organization_id,phone_number (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζεται **περιορισμός ξένου κλειδιού** με το όνομα FK_phone με **ξένο κλειδί** το organization_id ,το οποίο **αναφέρεται** στον πίνακα organization_. Επίσης, σε ενδεχόμενη διαγραφή μιας γραμμής του πίνακα, αυτή διαγράφεται **αυστηρά** από τον πίνακα phone_number καθώς δεν θέλουμε να διαγραφεί και ο οργανισμός με τον οποίο συνδέεται. Στην περίπτωση της ενημέρωσης τότε ο πίνακας organization_ ενημερώνεται διαδοχικά αφού συνδέεται με τον πίνακα phone_number.

Οντότητα Ερευνητής

```

create table researcher (
researcher_id int unsigned not null auto_increment,
first_name varchar(20) not null,
last_name varchar(20) not null,
gender varchar(10) check(gender in ('male','female','other')),
date_of_birth date not null check(year(date_of_birth) < 2000),
organization_id int unsigned not null,
start_date date not null,
primary key (researcher_id),
constraint researcher_age_check check (year(start_date) - year(date_of_birth) > 21),
constraint researcher_start_date_check check (start_date < '2022-06-05'),
CONSTRAINT FK_researcher
FOREIGN KEY (organization_id) REFERENCES organization_ (organization_id) on
delete restrict on update cascade

);

```

Σχολιασμός: Η σύνθετη ιδιότητα name του ER διαγράμματος στο σχεσιακό μοντέλο γράφεται με όλα τα δευτερεύοντα μέλη της (first_name, last_name). Επίσης, η οντότητα ερευνητής στο σχεσιακό μοντέλο κληρονομεί την ιδιότητα start_date της σχέσης works_for στο ER διάγραμμα καθώς αποτελεί σχέση πολλά προς ένα, όπως και το organization_id που αποτελεί τον περιορισμό ξένου κλειδιού.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
researcher_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
first_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 20 (varchar(20)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
last_name	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 20 (varchar(20)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
gender	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 20 (varchar(20))
date_of_birth	<ul style="list-style-type: none"> • αντικείμενο τύπου date της μορφής YYYY-MM-DD • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
organization_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
start_date	<ul style="list-style-type: none"> • αντικείμενο τύπου date της μορφής YYYY-MM-DD • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: executive_id (primary key constraint)

Ακεραιότητα πεδίου τιμών	
Εντολή SQL	Σχολιασμός
check(gender in ('male','female','other'))	Ένας ερευνητής μπορεί να δηλώσει τα εξής για το φύλο του: <ul style="list-style-type: none"> • Άνδρας (male) • Γυναίκα (female) • Άλλο (other)
check(year(date_of_birth) < 2000)	Έχει οριστεί η χρονιά γέννησης του κάθε στελέχους ερευνητή να είναι μικρότερη του 2000, δηλαδή ένας ερευνητής που δουλεύει σε κάποιο project πρέπει να έχει ηλικία από 22 χρονών και πάνω.
check (start_date < '2022-06-05')	Ορίζεται η ημερομηνία έναρξης εργασίας ενός ερευνητή σε μια εταιρεία να είναι μικρότερη από την σημερινή ημερομηνία

Αναφορική Ακεραιότητα: Ορίζεται **περιορισμός ξένου κλειδιού** με το όνομα FK_researcher με **ξένο κλειδί** το organization_id ,το οποίο **αναφέρεται** στον πίνακα organization_. Επίσης, σε ενδεχόμενη διαγραφή μιας γραμμής του πίνακα, αυτή διαγράφεται **αυστηρά** από τον πίνακα researcher καθώς δεν θέλουμε να διαγραφεί και ο οργανισμός με τον οποίο συνδέεται. Στην περίπτωση της ενημέρωσης τότε ο πίνακας organization_ ενημερώνεται διαδοχικά αφού συνδέεται με τον πίνακα researcher.

Οντότητα Έργο/Επιχορήγηση

```
create table project (
project_id int unsigned not null auto_increment,
title varchar(50) not null,
summary text(100) not null,
funding double(2,2) not null,
start_date date not null,
rating double(1,1) not null,
end_date date,
duration int unsigned,
evaluator int unsigned,
accountable int unsigned,
program_id int unsigned,
executive_id int unsigned,
organization_id int unsigned,
```

```

primary key (project_id),
CONSTRAINT FK_evaluator FOREIGN KEY (evaluator) REFERENCES
researcher(researcher_id) on delete restrict on update cascade,
CONSTRAINT FK_accountable FOREIGN KEY (accountable) REFERENCES
researcher(researcher_id) on delete restrict on update cascade,
CONSTRAINT FK_program FOREIGN KEY (program_id) REFERENCES
hfri_programs(program_id) on delete restrict on update cascade,
CONSTRAINT FK_exec FOREIGN KEY (executive_id) REFERENCES
hfri_executive(executive_id) on delete restrict on update cascade,
CONSTRAINT FK_organization FOREIGN KEY (organization_id) REFERENCES
organization_(organization_id) on delete restrict on update cascade,

-- CONSTRAINT project_ck_1 check(duration = datediff(start_date,end_date)),
constraint project_ck_2 check (1*365<= duration and duration <= 4*365),
constraint project_ck_3 check (start_date < '2022-06-05'),
constraint project_ck_4 check(funding >= 0.1 and funding <= 1.00),
constraint project_ck_5 check(rating>4.9),
constraint project_ck_6 check(evaluator<>accountable)
);

```

Σχολιασμός: Η σύνθετη ιδιότητα name του ER διαγράμματος στο σχεσιακό μοντέλο γράφεται με όλα τα δευτερεύοντα μέλη της (first_name, last_name). Επίσης, η οντότητα ερευνητής στο σχεσιακό μοντέλο κληρονομεί την ιδιότητα start_date της σχέσης works_for στο ER διάγραμμα καθώς αποτελεί σχέση πολλά προς ένα.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
project_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
title	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 50 (varchar(50)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
summary	<ul style="list-style-type: none"> • αντικείμενο τύπου κειμένου (text) που έχει μέγιστο όριο χαρακτήρων 100 (text(100)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
funding	<ul style="list-style-type: none"> • δεκαδικός (double) με ακέραιο πεδίο 7 ψηφία και

	δεκαδικό πεδίο 2 ψηφία • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
start_date	• αντικείμενο τύπου date της μορφής YYYY-MM-DD • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
end_date	• αντικείμενο τύπου date της μορφής YYYY-MM-DD
duration	• ακέραιος (int) • μη προσημασμένος (unsigned)
rating	• δεκαδικός (double) με ακέραιο πεδίο 1 ψηφίο και δεκαδικό πεδίο 2 ψηφία • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
evaluator	• ακέραιος (int) • μη προσημασμένος (unsigned)
accountable	• ακέραιος (int) • μη προσημασμένος (unsigned)
program_id	• ακέραιος (int) • μη προσημασμένος (unsigned)
executive_id	• ακέραιος (int) • μη προσημασμένος (unsigned)
organization_id	• ακέραιος (int) • μη προσημασμένος (unsigned)

Περιορισμός πρωταρχικού κλειδιού: project_id (primary key constraint)

Ακεραιότητα πεδίου τιμών	
Εντολή SQL	Σχολιασμός
check (1*365<= duration and duration <= 4*365)	Ελέγχει τον περιορισμό ότι η διάρκεια ενός έργου πρέπει να κυμαίνεται από 1 χρόνο έως 4 χρόνια
check (start_date < '2022-06-05')	Έχει οριστεί η ημερομηνία έναρξης ενός project να είναι μικρότερη της σημερινής ημερομηνίας
check(funding >= 0.1 and funding <= 1.00)	Ελέγχει τον περιορισμό ότι το ποσό της επιχορήγησης ενός έργου θα είναι από 100.000 έως 1.000.000

check(rating>4.9)	Ελέγχει το γεγονός ότι ένα έργο έχει αξιολογηθεί θετικά ώστε να χρηματοδοτηθεί στη συνέχεια
check(evaluator<>accountable)	Ελέγχει το γεγονός ότι ο αξιολογητής και ο υπεύθυνος ενός έργου δεν μπορεί να είναι το ίδιο άτομο.

Αναφορική Ακεραιότητα		
Όνομα	Ξένο Κλειδί	Αναφερόμενος Πίνακας
FK_evaluator	evaluator (renamed researcher_id)	researcher
FK_accountable	accountable (renamed researcher_id)	researcher
FK_exec	executive_id	hfri_executive
FK_program	program_id	hfri_programs
FK_organization	organization_id	organization_

Σε ενδεχόμενη διαγραφή μιας γραμμής του πίνακα, αυτή διαγράφεται **αυστηρά** από τον πίνακα project καθώς δεν θέλουμε να διαγραφούν και ο οργανισμός, οι ερευνητές, το πρόγραμμα, το στέλεχος ΕΛΙΔΕΚ και ο οργανισμός με τα οποία συνδέεται το συγκεκριμένο έργο/επιχορήγηση. Στην περίπτωση της ενημέρωσης ενός έργου/επιχορήγησης, τότε οι συνδεδεμένοι πίνακες ενημερώνονται διαδοχικά.

Σχέση “Δουλεύει σε”

```
create table works_on (
researcher_id int unsigned,
project_id int unsigned,
```

```
PRIMARY KEY (researcher_id, project_id),
```

```
CONSTRAINT FK_works_on_project
```

```
FOREIGN KEY (project_id) REFERENCES project (project_id) on delete restrict on
update cascade,
```

```
CONSTRAINT FK_works_on_researcher
```

```
FOREIGN KEY (researcher_id) REFERENCES researcher (researcher_id) on delete
restrict on update cascade
```

```
);
```

Σχολιασμός: Η σχέση “δουλεύει σε” μετατρέπεται σε πίνακα στο σχεσιακό μοντέλο διότι αποτελεί σχέση “πολλά προς πολλά”. Για αυτό κληρονομεί σαν πρωταρχικό κλειδί, τα πρωταρχικά κλειδιά των οντοτήτων που συσχετίζει, τα οποία αποτελούν και περιορισμούς ξένου κλειδιού.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
researcher_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned)
project_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned)

Περιορισμός πρωταρχικού κλειδιού: researcher_id, project_id (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζονται περιορισμοί ξένου κλειδιού με το όνομα works_on_project και works_on_researcher με ξένα κλειδιά τα researcher_id και project_id ,τα οποία αναφέρονται στους πίνακες researcher και project αντίστοιχα. Επίσης, σε ενδεχόμενη διαγραφή μιας γραμμής του πίνακα, αυτή διαγράφεται **αυστηρά** από τον πίνακα works_on καθώς δεν θέλουμε να διαγραφεί ούτε το έργο ούτε οι ερευνητές με τους οποίους συνδέεται. Στην περίπτωση της ενημέρωσης τότε οι συσχετιζόμενοι πίνακες ενημερώνεται διαδοχικά.

Οντότητα Επιστημονικό Πεδίο

```
create table field_of_research(
field_of_research varchar(100) not null,

primary key (field_of_research)
);
```

Περιορισμοί ακεραιότητας: field_of_research, πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 100 (varchar(100)), που δεν επιτρέπεται να είναι κενό (not null) και αποτελεί και το πρωταρχικό κλειδί της οντότητας.

Οντότητα Παραδοτέο

```
create table deliverable (
deliverable_id int unsigned not null auto_increment,
project_id int unsigned,
```

title varchar(100) not null,
summary varchar(100) not null,
due_date date not null,

primary key (deliverable_id),
CONSTRAINT FK_deliverable

FOREIGN KEY (project_id) REFERENCES project (project_id) on delete restrict on
update cascade
);

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
deliverable_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned) • δεν επιτρέπεται να είναι κενό το πεδίο (not null) • σε κάθε εισαγωγή στον πίνακα παίρνει αυτόματα μια μοναδική τιμή (auto_increment)
project_id	<ul style="list-style-type: none"> • ακέραιος (int) • μη προσημασμένος (unsigned)
title	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 100 (varchar(100)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
summary	<ul style="list-style-type: none"> • πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 100 (varchar(100)) • δεν επιτρέπεται να είναι κενό το πεδίο (not null)
due_date	<ul style="list-style-type: none"> • αντικείμενο τύπου date της μορφής YYYY-MM-DD • δεν επιτρέπεται να είναι κενό το πεδίο (not null)

Περιορισμός πρωταρχικού κλειδιού: deliverable_id (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζονται **περιορισμοί ξένου κλειδιού** με το όνομα FK_deliverable με **ξένο κλειδί** τα project_id, το οποίο **αναφέρεται** στον πίνακα project. Επίσης, σε ενδεχόμενη διαγραφή μιας γραμμής του πίνακα, αυτή διαγράφεται **αυστηρά** από τον πίνακα deliverable, καθώς δεν θέλουμε να διαγραφεί το έργο με το οποίο συνδέεται. Στην περίπτωση της ενημέρωσης τότε ο συσχετιζόμενος πίνακας έργο ενημερώνεται διαδοχικά.

Σχέση “Πεδίο έργου”

```
create table project_field (  
project_id int unsigned,  
field_of_research varchar(100) not null,
```

```
PRIMARY KEY (project_id,field_of_research),
```

```
CONSTRAINT FK_FR_project
```

```
FOREIGN KEY (project_id) REFERENCES project (project_id) on delete restrict on  
update cascade,
```

```
CONSTRAINT FK_FR_FR
```

```
FOREIGN KEY (field_of_research) REFERENCES field_of_research  
(field_of_research) on delete restrict on update cascade  
);
```

Σχολιασμός: Η σχέση “πεδίο έργου” μετατρέπεται σε πίνακα στο σχεσιακό μοντέλο διότι αποτελεί σχέση “πολλά προς πολλά”. Για αυτό κληρονομεί σαν πρωταρχικό κλειδί, τα πρωταρχικά κλειδιά των οντοτήτων που συσχετίζει, τα οποία αποτελούν και περιορισμούς ξένου κλειδιού.

Περιορισμοί ακεραιότητας	
Ιδιότητες	Σχολιασμός
field_of_research	<ul style="list-style-type: none">• πεδίο μεταβλητού χαρακτήρα που έχει μέγιστο όριο 100 (varchar(100))• δεν επιτρέπεται να είναι κενό πεδίο (not null)
project_id	<ul style="list-style-type: none">• ακέραιος (int)• μη προσημασμένος (unsigned)

Περιορισμός πρωταρχικού κλειδιού: field_of_research, project_id (primary key constraint)

Αναφορική Ακεραιότητα: Ορίζονται **περιορισμοί ξένου κλειδιού** με το όνομα FK_FR_project και FK_FR_FR με **ξένα κλειδιά** τα project_id και field_of_research ,τα οποία **αναφέρονται** στους πίνακες project και field_of_research αντίστοιχα. Επίσης, σε ενδεχόμενη διαγραφή μιας γραμμής του πίνακα, αυτή διαγράφεται **αυστηρά** από τον πίνακα project_field, καθώς δεν θέλουμε να διαγραφεί ούτε το έργο ούτε το πεδίο έρευνας, με τα οποία συνδέεται. Στην περίπτωση της ενημέρωσης τότε οι συσχετιζόμενοι πίνακες ενημερώνεται διαδοχικά.

Όλα τα παραπάνω υπάρχουν στο αρχείο **hfri_schema.sql (DDL script)** ,το οποίο όταν εκτελείται δημιουργεί την βάση δεδομένων που μας ζητήθηκε με όλους τους περιορισμούς που χρειάζονται ώστε η βάση να είναι συνεπής.

Στη συνέχεια, η εισαγωγή δεδομένων στη βάση που δημιουργήθηκε από το προηγούμενο αρχείο πραγματοποιήθηκε από το αρχείο **hfri_insert.sql (DML script)**. Το αρχείο αυτό δημιουργήθηκε χρησιμοποιώντας την γλώσσα προγραμματισμού Python, προκειμένου η βάση να διαθέτει αρκετά δεδομένα έτσι ώστε να μπορούν να απαντηθούν όλες οι ερωτήσεις που απαιτούνται στο Μέρος 3 της εργασίας.

Στο αρχείο **creator.py** χρησιμοποιήθηκαν οι βιβλιοθήκες **faker** και **numpy** προκειμένου να παράγουμε τυχαίους αριθμούς και δεδομένα, τα οποία στη συνέχεια μετατράπηκαν σε κατάλληλη εντολή **insert** της SQL ώστε να εισαχθούν στη βάση **hfri_db** χωρίς να υπάρχουν σφάλματα από τους διάφορους περιορισμούς που αναλύθηκαν προηγουμένως.

Συγκεκριμένα, δημιουργήθηκαν τα εξής δεδομένα:

- 150 οργανισμοί (50 πανεπιστήμια, 50 ερευνητικά κέντρα και 50 εταιρείες)
- Ο κάθε οργανισμός έχει από 1 έως 3 τηλέφωνα.
- 3000 ερευνητές
- Ένας ερευνητής μπορεί να εργάζεται σε ένα ή περισσότερα έργα (5000 περιπτώσεις ερευνητών που εργάζονται σε κάποιο έργο)
- 100 στελέχη του ΕΛΙΔΕΚ
- 70 προγράμματα του ΕΛΙΔΕΚ
- 1000 έργα (700 ενεργά , 300 μη ενεργά)
- Το κάθε έργο έχει από 0 έως 3 παραδοτέα.
- Το κάθε έργο αφορά ένα ή περισσότερα από τα 17 συνολικά επιστημονικά πεδία

Τέλος, για την καλύτερη απόκριση της βάσης μας στα ερωτήματα του 3ου μέρους, δημιουργήθηκαν διάφορα **ευρετήρια (indexes)**, τα οποία επιτρέπουν την γρηγορότερη εύρεση κάποιων δεδομένων όταν αυτά ζητούνται μέσα από ένα πίνακα. Όλα τα indexes φαίνονται στο αρχείο **indices.sql** . Σε αυτά που αφορούν στήλες οι οποίες περιέχουν μοναδικά δεδομένα χρησιμοποιήθηκε ο όρος **unique index**. Επίσης, για όλα τα ευρετήρια χρησιμοποιήθηκε ο default τύπος ευρετηρίου B-tree (Balanced Tree). Με αυτό τον τύπο ευρετηρίου το σύστημα διαχείρισης της βάσης δεδομένων μας θα προσπαθήσει αυτόματα να κρατήσει μια ισορροπία όταν εισάγονται ή ενημερώνονται ή διαγράφονται δεδομένα. Τα ευρετήρια που δημιουργήθηκαν είναι στοχευμένα σε στήλες οι οποίες ανακτώνται περισσότερες φορές από τα διάφορα ερωτήματα, καθώς δεν θέλουμε να υπερ φορτώσουμε τον server βάζοντας ευρετήρια σε κάθε στήλη.

Συγκεκριμένα, τα indexes που δημιουργήθηκαν με βάση τις στήλες που απευθύνονται οι ερωτήσεις του 3ου μέρους είναι τα εξής:

Όνομα Ευρετηρίου	Πίνακας	Ιδιότητα	unique	Σχολιασμός
program_index	hfri_programs	program_name	OXI	Ερώτημα 3.1
title_index	project	title	OXI	Ερωτήματα 3.1, 3.3
end_date_index	project	end_date	OXI	Ερωτήματα 3.3, 3.4
r_full_name_index	researcher	first_name last_name	OXI	Ερωτήματα 3.3, 3.6, 3.8
field_of_research_index1	project_field	field_of_research	OXI	Ερώτημα 3.3
organization_name_index	organization_	organization_name	OXI	Ερωτήματα 3.4, 3.7
start_date_index	project	start_date	OXI	Ερωτήματα 3.1, 3.4
field_of_research_index2	field_of_research	field_of_research	NAI	Ερώτημα 3.5
projectID_index1	project_field	field_of_research	OXI	Ερώτημα 3.5
researcherID_index	researcher	researcher_id	NAI	Ερώτημα 3.6
date_of_birth_index	researcher	date_of_birth	OXI	Ερώτημα 3.6
funding_index	project	funding	OXI	Ερώτημα 3.7
projectID_index2	project	project_id	NAI	Ερώτημα 3.8
e_full_name_index	hfri_executive	first_name last_name	OXI	Ερώτημα 3.7

Περιορισμοί ορισμένοι από τον χρήστη: Γνωρίζουμε ότι ο αξιολογητής ενός έργου δεν μπορεί ταυτόχρονα και να δουλεύει σε αυτό. Αυτός ο περιορισμός υλοποιήθηκε με **sql trigger** κατά την ενημέρωση ή την εισαγωγή δεδομένων στον πίνακα works_on όπως φαίνεται παρακάτω:

```
delimiter //
CREATE TRIGGER check_evaluator_insert BEFORE INSERT on works_on
FOR EACH ROW
BEGIN
    IF new.researcher_id in (select evaluator from project where
project_id = new.project_id) THEN
        SIGNAL SQLSTATE '45000'
```

```

        SET MESSAGE_TEXT = 'Cant put this researcher_id on works_on
table';
    END IF;
END;

delimiter ;

delimiter //
CREATE TRIGGER check_evaluator_update BEFORE UPDATE on works_on
FOR EACH ROW
BEGIN
    IF new.researcher_id in (select evaluator from project where
project_id = new.project_id) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cant put this researcher_id on works_on
table';
    END IF;
END;

delimiter ;

```

Τα παραπάνω triggers μας εμφανίζουν ένα μήνυμα ότι αυτή η ενέργεια της ενημέρωσης και της εισαγωγής ενός αξιολογητή στον πίνακα works_on δεν μπορεί να πραγματοποιηθεί.

Μέρος 3

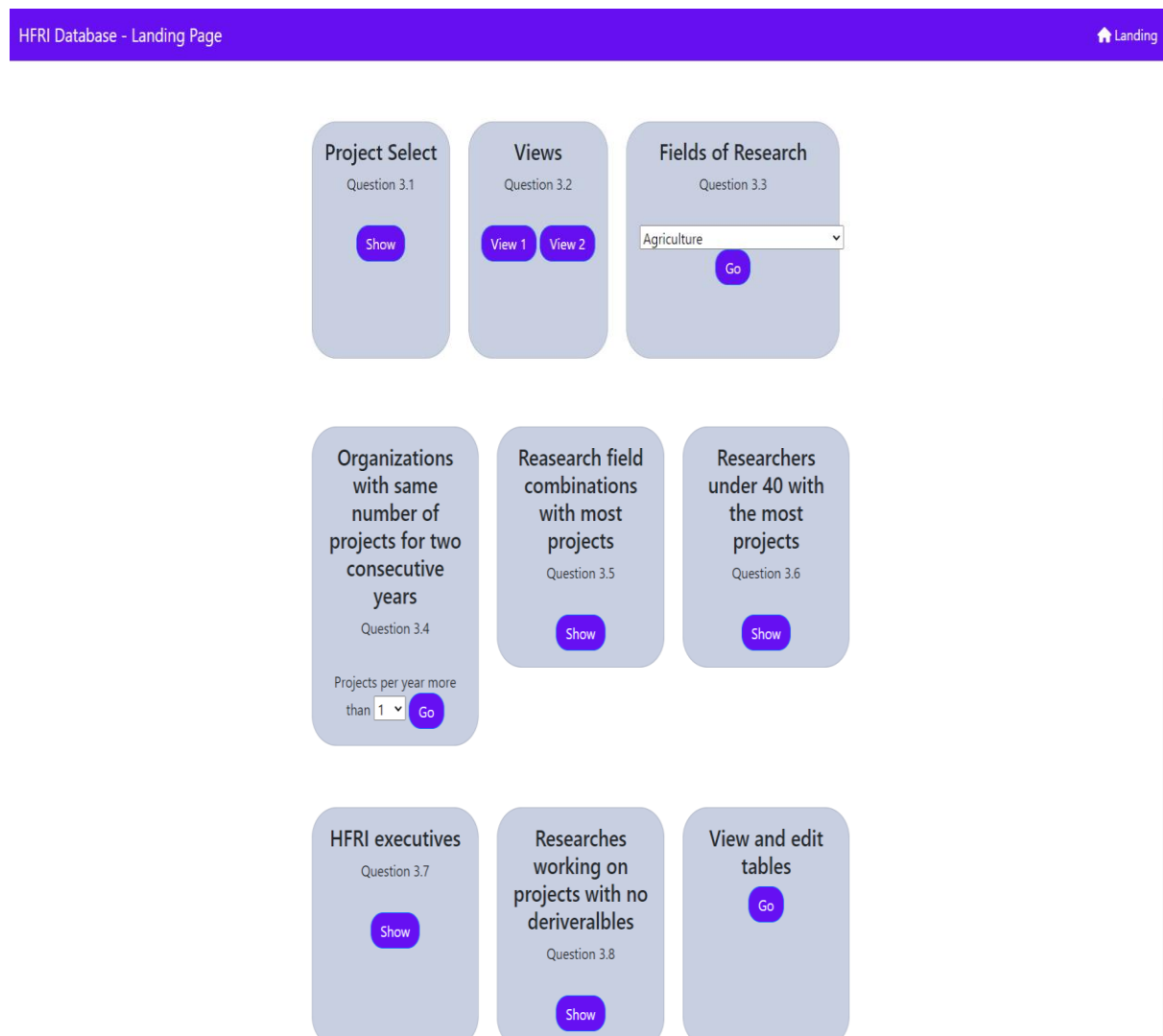
Σε αυτό το μέρος της εργασίας δημιουργήθηκε ένα **user interface (UI)** φιλικό προς το χρήστη, με το οποίο γίνεται η διασύνδεση με τη βάση δεδομένων hfri_db, από την οποία ανακτώνται τα δεδομένα που απαντούν στα ζητούμενα ερωτήματα, μέσω της γλώσσας προγραμματισμού python, χρησιμοποιώντας το πακέτο **Flask**.

Αρχικά, πριν το ξεκίνημα της εφαρμογής θα πρέπει να έχουν εγκατασταθεί όλα τα απαιτούμενα πακέτα python που βρίσκονται στο αρχείο **requirements.txt**. Αυτό γίνεται μέσω της εντολής **pip install -r requirements.txt**.

Στο αρχείο **__init__.py** υλοποιείται η διασύνδεση με τη βάση δεδομένων μας. Αρχικά, δημιουργείται το **app**, που είναι αντικείμενο του Flask και ενεργοποιείται ο debugger. Στην συνέχεια, εισάγονται διάφοροι παράμετροι στο αντικείμενο app, όπως ο χρήστης ο οποίος δημιούργησε τη βάση δεδομένων (root), το συνθηματικό του με το οποίο εισάγεται σε αυτή (password), το όνομα της βάσης δεδομένων (hfri_db), το όνομα του εξυπηρετητή MySQL (localhost) και το μυστικό κλειδί (secret key) μέσω του οποίο ανταλλάσσονται με ασφάλεια τα δεδομένα της βάσης. Αφού, ρυθμιστούν οι διάφοροι παράμετροι, δημιουργείται το MySQL αντικείμενο, που θα υλοποιήσει την διασύνδεση με τη βάση δεδομένων που ρυθμίστηκε (db = MySQL(app)).

Για να ξεκινήσει η εφαρμογή εκτελείται σε ένα command window το αρχείο **run.py (python run.py)** , αφού βρισκόμαστε στον φάκελο όπου περιέχονται όλα τα αρχεία της εφαρμογής.

Παρατηρούμε ότι ο server λειτουργεί κανονικά και αγνοώντας την προειδοποίηση που βγάζει στο command window, πηγαίνουμε στον browser και πληκτρολογούμε την εντολή **localhost:3000**, όπου localhost είναι ο τοπικός server και 3000 η πόρτα μέσω της οποίας συνδεόμαστε με αυτόν. Στη συνέχεια, βλέπουμε στην οθόνη την παρακάτω εικόνα, η οποία αποτελεί την κεντρική ιστοσελίδα της εφαρμογής:



Παρατηρούμε ότι το κάθε πλαίσιο της εικόνας απαντάει σε ένα ζητούμενο ερώτημα και στο τελευταίο κάτω δεξιά πλαίσιο υλοποιείται η εισαγωγή, το διάβασμα, η ενημέρωση και η διαγραφή δεδομένων από τη βάση hfri_db (**CRUD - Create Read Update Delete**).

Η παραπάνω ιστοσελίδα δημιουργείται μέσω του αρχείου **landing.html** , το οποίο επεκτείνει (extends) το αρχείο **global_layout.html**. Στο τελευταίο αρχείο, υλοποιούνται όλα τα αρχεία που βρίσκονται στο φάκελο **static**. Αυτός περιέχει διάφορα αρχεία τύπου bootstrap, css και js , καθώς και εικόνες, τα οποία συνθέτουν ένα συγκεκριμένο design το οποίο υλοποιείται στην ιστοσελίδα landing.html, όπως φαίνεται παραπάνω.

Στην κορυφή της ιστοσελίδας υπάρχει ένα μπλε μενού το οποίο δημιουργείται μέσω του αρχείου **navbar.html**. Σε αυτό ορίζεται στο αριστερό μέρος, ο τίτλος της ιστοσελίδας **HFRI Database - Landing Page** (το Landing Page ορίζεται σαν μεταβλητή PageTitle , η οποία δίνεται στη συνάρτηση index του αρχείου **routes.py** , μέσω της οποίας απεικονίζεται το αρχείο landing.html). Στο δεξί μέρος του μενού, υπάρχει μια εικόνα σπιτιού και η λέξη “Landing” , η οποία αποτελεί ένα σύνδεσμο που πατώντας τον μας επιστρέφει πάντα στην ιστοσελίδα landing.html.

Παρακάτω αναλύονται οι υλοποιήσεις του κάθε ζητούμενου ερωτήματος:

Ερώτημα 3.1

Υλοποιείται στην ιστοσελίδα **Q31.html**, η οποία απεικονίζεται μέσω της συνάρτησης Q31 του αρχείου routes.py. Σε αυτή την συνάρτηση, πραγματοποιούνται τα παρακάτω:


- Το αντικείμενο cur που υλοποιεί την διασύνδεση με τη βάση δεδομένων
- Η μεταβλητή query στην οποία υλοποιείται το MySQL query:
`select * from hfri_programs;`
- Εκτελείται το query (cur.execute(query))
- Επιστρέφονται οι στήλες και ο πίνακας που ζητήθηκαν στο query, μέσω των μεθόδων description και fetchall)
- Μέσω αυτού του ερωτήματος και αφού ο χρήστης πατήσει το κουμπί show στο πλαίσιο με τίτλο Question 3.1, επιστρέφονται όλα τα διαθέσιμα προγράμματα του ΕΛΙΔΕΚ με όλα τους τα στοιχεία.
- Στο τέλος κλείνει η σύνδεση με τη βάση δεδομένων


Επίσης, η ιστοσελίδα Q31 περιέχει ένα κουμπί το οποίο αν πατηθεί από τον χρήστη θα τον ανακατευθύνει στην ιστοσελίδα **Q31res3.html** , η οποία φαίνεται στην παρακάτω εικόνα:

Filter By

Check none to see all projects

Date ☐

Start Date After 

Start Date Before 

Duration ☐

Duration: more than days

Executive ☐

HFRI executive ID 



Μέσω των στοιχείων της παραπάνω εικόνας, ο χρήστης προτρέπεται να εισάγει στοιχεία, που αφορούν την ημερομηνία έναρξης ενός έργου, τη διάρκεια του καθώς και τον αύξοντα αριθμό στελέχους του ΕΛΙΔΕΚ που χειρίζεται έργα (πιο κάτω φαίνονται όλα τα στελέχη του ΕΛΙΔΕΚ ώστε να επιλέξει ο χρήστης αυτόν που επιθυμεί). Αν δεν εισάγει κανένα στοιχείο τότε θα δει όλα τα διαθέσιμα έργα.

Στην ιστοσελίδα **Q31res3.html** παρατηρούμε 3 checkboxes, τα οποία μπορεί να επιλέξει ο χρήστης ανάλογα με τα κριτήρια που επιθυμεί. Στο πεδίο **Date** υπάρχουν πλαίσια (input type="date") στα οποία ο χρήστης μπορεί να εισάγει το διάστημα των ημερομηνιών έναρξης για τα διάφορα έργα που επιθυμεί να δει. Στο πεδίο **Duration**, υπάρχει ένα πλαίσιο (input type="text") στο οποίο ο χρήστης εισάγει τον αριθμό των ημερών διάρκειας πάνω από τις οποίες επιθυμεί να εμφανιστούν τα διάφορα έργα. Τέλος, στο πεδίο **Executive** υπάρχει μία drop-down λίστα με όλους τους αύξοντες αριθμούς των στελεχών του ΕΛΙΔΕΚ.

Κάνοντας κλικ (checkbox) στο πεδίο στο οποίο θέλει να εισάγει στοιχεία ο χρήστης , βάζοντας τα στοιχεία που επιθυμεί και πατώντας το κουμπί “Go” ανακατευθύνεται στην ιστοσελίδα **Q31res.html**.

Η ιστοσελίδα αυτή υλοποιείται μέσω της αντίστοιχης συνάρτησης στο αρχείο routes.py. Συγκεκριμένα ελέγχονται διάφορες περιπτώσεις ανάλογα με τα checkboxes που επέλεξε ο χρήστης. Ανάλογα με τα στοιχεία που εισάγει ο χρήστης μέσω της μεθόδου “**request.get.form()**” μπορούμε να τα αποθηκεύσουμε σε μεταβλητές. Στη συνέχεια τα εισάγουμε καταλλήλως σε ένα SQL query (σε μία where πρόταση) που αφορά τα διαθέσιμα έργων σύμφωνα με τα κριτήρια που επέλεξε ο χρήστης.

Στη συνέχεια, τα project_id του κάθε έργου που εμφανίζεται είναι με την μορφή κουμπιού (button) και τα επιλέξει ο χρήστης, ανακατευθύνεται στην ιστοσελίδα **Q31res2.html** μέσω της οποίας μπορεί να δει τους ερευνητές που εργάζονται στο συγκεκριμένο έργο. Ομοίως, αυτή η ιστοσελίδα υλοποιείται από την αντίστοιχη συνάρτηση στο αρχείο routes.py και εκτελείται το παρακάτω SQL query:

```
select r.first_name, r.last_name, r.researcher_id from researcher r
inner join works_on w on w.researcher_id = r.researcher_id inner
join project p on p.project_id = w.project_id where p.project_id = "
+ select
```

όπου “select” είναι η επιλογή του χρήστη από το κουμπί που πατήθηκε για ένα συγκεκριμένο έργο.

Ερώτηση 3.2

Στο συγκεκριμένο ερώτημα υλοποιήθηκαν 2 όψεις του σχεσιακού μοντέλου στο αρχείο hfri_schema. Η πρώτη αφορά τα έργα ανά ερευνητή και η άλλη τα έργα ανά πρόγραμμα και υλοποιήθηκαν όπως φαίνεται παρακάτω:

```
create view projects_per_researcher as
select r.researcher_id, r.first_name, r.last_name, p.project_id ,
p.title
from researcher r
inner join works_on wo on wo.researcher_id = r.researcher_id
inner join project p on wo.project_id = p.project_id
order by r.researcher_id;
```

```
create view projects_per_program as
select hp.program_id, hp.program_name, p.project_id , p.title
from hfri_programs hp
inner join project p on hp.program_id = p.program_id
order by hp.program_id;
```

Είναι γνωστό από τη θεωρία ότι οι **όψεις** είναι εικονικοί πίνακες. Η διαφορά μεταξύ μιας όψης και ενός πίνακα είναι ότι οι όψεις είναι ορισμοί που χτίζονται πάνω από άλλους πίνακες (ή όψεις) και δεν διατηρούν τα ίδια δεδομένα. Εάν τα δεδομένα αλλάζουν στον υποκείμενο πίνακα, η ίδια αλλαγή αντικατοπτρίζεται στην όψη.

Η παρουσίαση των αποτελεσμάτων των δυο όψεων υλοποιήθηκαν σε δυο διαφορετικές συναρτήσεις στο αρχείο routes.py, δηλαδή τη συνάρτηση **Q321** και **Q322** αντίστοιχα. Οι συναρτήσεις αυτές απεικονίζουν στις ιστοσελίδες Q321.html και Q322.html αντίστοιχα τα αποτελέσματα των παρακάτω SQL queries:

```
select * from projects_per_researcher order by project_id,
researcher_id
select * from projects_per_program order by program_id, project_id
```

Όλα τα παρακάτω ερωτήματα έως και το 3.8 απεικονίζονται σε ιστοσελίδες οι οποίες υλοποιούνται με τον ίδιο τρόπο που αναλύθηκε παραπάνω από συναρτήσεις οι οποίες έχουν το ίδιο όνομα, μέσω της python (flask) στο αρχείο routes.py.

Ερώτημα 3.3

Στο ερώτημα αυτό όπως φαίνεται στο UI προτρέπεται ο χρήστης να επιλέξει από μια drop-down λίστα, το επιστημονικό πεδίο το οποίο θεωρεί ότι τον τελευταίο χρόνο απέκτησε μεγαλύτερο ενδιαφέρον. Τα έργα που θα απεικονιστούν σύμφωνα με την επιλογή του χρήστη θα πρέπει να είναι ενεργά, που σημαίνει ότι η ημερομηνία λήξης τους θα πρέπει να είναι μεγαλύτερη από την σημερινή ημερομηνία.

Τα έργα που χρηματοδοτούνται σε αυτό το επιλεγμένο από το χρήστη πεδίο βρίσκονται με το παρακάτω SQL query:

```
select pr.title,pf.field_of_research,pr.end_date from project pr
inner join project_field pf on pr.project_id = pf.project_id where
pf.field_of_research = '"+select+"' and pr.end_date > curdate() order
by pr.end_date
```

όπου select είναι η επιλογή του χρήστη από την drop-down λίστα των επιστημονικών πεδίων

Οι ερευνητές που ασχολούνται με αυτό το επιλεγόμενο πεδίο τον τελευταίο χρόνο, εργάζονται σε έργα τα οποία αφορούν το συγκεκριμένο πεδίο και είναι ενεργά.

```
select r.first_name,r.last_name,year(p.end_date) as end_year,
p.title as project_name, pf.field_of_research from researcher r
inner join works_on w on r.researcher_id = w.researcher_id inner
join project p on p.project_id = w.project_id inner join
project_field pf on p.project_id = pf.project_id where
```

```
pf.field_of_research = '"+select+"' and (p.end_date > curdate() or  
year(p.end_date) = year(curdate()))
```

Ερώτημα 3.4

Οι οργανισμοί που έχουν λάβει τον ίδιο αριθμό έργων σε διάστημα δυο συνεχόμενων ετών με τουλάχιστον 10 έργα ετησίως, προκύπτουν από το παρακάτω SQL query:

```
select t.organization_name, t.start_year as year_A  
,t.projects_in_year as projects_in_year_A, t1.start_year as year_B,  
t1.projects_in_year as projects_in_year_B from (select  
o.organization_name as organization_name , year(pr.start_date) as  
start_year, count(*) as projects_in_year, (year(pr.start_date) + 1)  
as next_year from organization_ o inner join project pr on  
o.organization_id = pr.organization_id group by start_year,  
o.organization_name order by projects_in_year desc) t inner join  
(select o.organization_name as organization_name ,  
year(pr.start_date) as start_year, count(*) as projects_in_year,  
(year(pr.start_date) + 1) as next_year from organization_ o inner  
join project pr on o.organization_id = pr.organization_id group by  
start_year, o.organization_name order by projects_in_year desc) t1  
on t.next_year = t1.start_year and t.organization_name =  
t1.organization_name having projects_in_year_A = projects_in_year_B  
and projects_in_year_A > " +mt+ " order by  
t.organization_name,t.start_year;
```

Στο παραπάνω SQL query χρησιμοποιήθηκαν **εμφωλιασμένα queries**. Συγκεκριμένα, στην πρόταση from και στην πρόταση inner join, έχουμε το ίδιο εμφωλιασμένο select όπου επιστρέφεται το όνομα του οργανισμού, η χρονιά έναρξης των έργων που διαχειρίζεται αυτός ο οργανισμός και ο αριθμός των έργων ομαδοποιημένα ανά έτος. Στην συνέχεια, παίρνουμε την **τομή (inner join)** του εμφωλιασμένου πίνακα με τον πίνακα των έργων και το ομαδοποιούμε ανάλογα με το όνομα του οργανισμού και το έτος έναρξης ενός έργου, έχοντας ίσο αριθμό εργαών το έτος A με το έτος B και ο αριθμός αυτών των έργων να είναι μεγαλύτερος του 10.

Ερώτημα 3.5

Τα 3 κορυφαία ζεύγη επιστημονικών πεδίων που εμφανίστηκαν σε ενεργά και μη ενεργά έργα προκύπτουν από το παρακάτω SQL query:

```
set @row_num = 0; DROP table IF EXISTS temp; CREATE TEMPORARY TABLE  
temp select @row_num := @row_num + 1 as id, fr1.field_of_research as
```

```

fr1, fr2.field_of_research as fr2 from field_of_research fr1 CROSS
JOIN field_of_research fr2 where fr1.field_of_research >
fr2.field_of_research and fr1.field_of_research <>
fr2.field_of_research; DROP table IF EXISTS temp2; CREATE TEMPORARY
TABLE temp2 select t.id, pf1.project_id, t.fr1, t.fr2 from temp t
inner join project_field pf1 on pf1.field_of_research = t.fr1 inner
join project_field pf2 on pf2.field_of_research = t.fr2 where
pf1.project_id = pf2.project_id group by pf1.project_id,t.id order
by t.id;
select t.fr1, t.fr2, count(t.id) as count from temp2 t group by
t.fr1, t.fr2 order by count desc limit 3;

```

Στο παραπάνω ερώτημα χρειάστηκε να δημιουργήσουμε δυο προσωρινούς πίνακες (temporary tables) και να θέσουμε μια μεταβλητή (row_num) ίση με μηδέν, η οποία θα αυξάνεται κατά 1 για κάθε γραμμή του προσωρινού πίνακα. Συγκεκριμένα, με τον πρώτο προσωρινό πίνακα **temp** δημιουργούμε τα διάφορα ζεύγη επιστημονικών πεδίων με **cross join** και τα οποία τα αριθμούμε μέσω της μεταβλητής row_num. Προκειμένου να μην έχουμε διπλότυπα στον πίνακα temp ελέγχουμε λεξιλογικά το κάθε επιστημονικό πεδίο και παίρνουμε το “διάφορο” (<>) για να μη μας προκύψουν τα ίδια επιστημονικά πεδία και στις δυο στήλες (fr1,fr2). Στη συνέχεια, δημιουργούμε νέο προσωρινό πίνακα **temp2**, χρησιμοποιώντας τον προηγούμενο προσωρινό πίνακα temp ως τομή με τον πίνακα project_field και με τις δυο στήλες fr1 και fr2 αντίστοιχα. Από την τομή αυτή θα προκύψουν τα ζεύγη επιστημονικών πεδίων που αφορούν κάθε έργο το οποίο είναι τουλάχιστον διεπιστημονικό. Τέλος, χρησιμοποιώντας τον δεύτερο προσωρινό πίνακα temp2 ομαδοποιούνται τα έργα με βάση το ζεύγος επιστημονικών πεδίων και γίνεται καταμέτρηση αυτών των έργων που περιέχουν το κάθε ζεύγος. Κάνοντας ταξινόμηση από το μεγαλύτερος προς το μικρότερο με βάση αυτή την καταμέτρηση κρατάμε τα τρία πρώτα ζεύγη που προέκυψαν.

Έρωτημα 3.6

Οι νέοι ερευνητές (ηλικίας < 40 ετών) που εργάζονται στα περισσότερα έργα καθώς και ο αριθμός αυτών των έργων φαίνεται στο παρακάτω SQL query:

```

DROP table IF EXISTS temp36; CREATE TEMPORARY TABLE temp36 select
r.researcher_id, first_name, last_name, timestampdiff(year,
date_of_birth, curdate()) as age, count(*) as num_projects from
researcher r inner join works_on w on
r.researcher_id=w.researcher_id inner join project p on
w.project_id=p.project_id where curdate()<p.end_date group by
researcher_id having age<40 order by num_projects desc;
select * from (select r.researcher_id, first_name, last_name,
timestampdiff(year, date_of_birth, curdate()) as age, count(*) as
num_projects from researcher r inner join works_on w on
r.researcher_id=w.researcher_id inner join project p on

```

```
w.project_id=p.project_id where curdate()<p.end_date group by
researcher_id having age<40 order by num_projects desc) t where
t.num_projects = (select num_projects from temp36 limit 1);
```

Και σε αυτό το ερώτημα δημιουργούμε προσωρινό πίνακα **temp36** , ο οποίος περιέχει το id του ερευνητή, το όνομα του , την ηλικία του (μικρότερη από 40) καθώς και τον αριθμό των ενεργών έργων στα οποία εργάζεται. Στη συνέχεια εκτελώντας σχεδόν το ίδιο query , χρησιμοποιώντας τον προσωρινό πίνακα temp36 βάζουμε στην πρόταση where τον μέγιστο αριθμό ενεργών έργων που μπορεί να εργάζεται κάποιος και βλέπουμε έτσι ποιοι ερευνητές εργάζονται στα περισσότερα ενεργά έργα.

Ερώτημα 3.7

Τα top-5 στελέχη του ΕΛΙΔΕΚ τα οποία έχουν δώσει τον μεγαλύτερο αριθμό χρηματοδοτήσεων σε μια εταιρεία βρίσκονται με το παρακάτω SQL query:

```
select first_name, last_name,e.executive_id, o.organization_name,
sum(p.funding) as tot_funding from hfri_executive e inner join
project p on e.executive_id=p.executive_id inner join organization_
o on p.organization_id=o.organization_id inner join company c on
c.organization_id = o.organization_id group by e.executive_id,
o.organization_id order by tot_funding desc limit 5;
```

Παίρνοντας την τομή των έργων με τα στελέχη του ΕΛΙΔΕΚ με βάση την ιδιότητα executive_id , στην συνέχεια την τομή αυτών με τους οργανισμούς με βάση την ιδιότητα organization_id και τέλος την τομή με τους οργανισμούς οι οποίοι είναι εταιρείες προκύπτει ένας πίνακας που περιέχει τα στελέχη του ΕΛΙΔΕΚ που χρηματοδοτούν εταιρείες. Έτσι , ομαδοποιώντας αυτά τα στελέχη με βάση το executive_id τους, προσθέτοντας κάθε φορά το ποσό της χρηματοδότησης που δίνουν σε κάθε εταιρεία και κρατώντας τους 5 πρώτους μετά από φθίνουσα ταξινόμηση προκύπτει η απάντηση στο ζητούμενο ερώτημα.

Ερώτημα 3.8

Οι ερευνητές που εργάζονται σε τουλάχιστον 5 έργα τα οποία δεν έχουν παραδοτέα προκύπτουν από το παρακάτω SQL query:

```
select r.first_name,r.last_name,count(p1.project_id) as num_projects
from researcher r inner join works_on w on r.researcher_id =
w.researcher_id inner join (select project_id from project p1 where
project_id not in (select p2.project_id from project p2 inner join
deliverable d on d.project_id = p2.project_id)) as p1 on
p1.project_id = w.project_id group by r.first_name,r.last_name
having (count(p1.project_id) >= 5) order by num_projects desc;
```


Το παραπάνω sql query περιλαμβάνει εμφωλιασμένα queries μέσα στην πρόταση inner join. Συγκεκριμένα, παίρνοντας την τομή των εργαζόμενων ερευνητών με έναν πίνακα που περιέχει τα έργα τα οποία δεν έχουν παραδοτέα, δηλαδή τα έργα που δεν βρίσκεται το id τους στον πίνακα deliverables, βρίσκουμε τους εργαζόμενους που δουλεύουν σε έργα χωρίς παραδοτέα. Ομαδοποιώντας αυτόν τον πίνακα με βάση τα ονόματα των ερευνητών, κρατώντας αυτούς που έχουν δουλέψει σε τουλάχιστον 5 τέτοια έργα και ταξινομώντας σε φθίνουσα σειρά με βάση των αριθμών των έργων, βρίσκουμε την απάντηση στο ζητούμενο ερώτημα.

Εισαγωγή-Διάβασμα-Ενημέρωση-Διαγραφή δεδομένων (CRUD)

Στο αρχείο routes.py δημιουργείται η συνάρτηση **alltables**, η οποία αντίστοιχα απεικονίζει την ιστοσελίδα alltables.html. Σε αυτή την συνάρτηση εκτελείται η εντολή **show tables** και κρατώντας τα αποτελέσματα αυτής της εντολής, προκύπτει ένας πίνακας με μία στήλη που περιέχει τα ονόματα όλων των πινάκων της βάσης δεδομένων μας. Στην συνέχεια, για κάθε μία γραμμή του πίνακα αυτού δημιουργούμε έναν υπερσύνδεσμο (hyperlink, html tag "href") της μορφής `{{ '/view?table=' + e.Tables_in_hfri_db + '&delete=0&add=0' }}` τρέχοντας ένα for βρόχο στο αρχείο alltables.html. Όπου e.Tables_in_hfri_db το όνομα του κάθε πίνακα της βάσης δεδομένων μας. Έτσι θα προκύψει ένας διαφορετικός υπερσύνδεσμος για κάθε πίνακα, όπως φαίνεται στην παρακάτω εικόνα:

Select a table to view or edit
company
deliverable
field of research
hfri_executive
hfri_programs
organization
phone_number
project
project_field
projects_per_program
projects_per_researcher
research_center
researcher
university
works_on

Πατώντας ο χρήστης σε έναν από αυτούς τους υπερσυνδέσμους μεταβαίνει σε μια ιστοσελίδα στην οποία εκτελείται ένα select query και εμφανίζει όλες τις ιδιότητες και στα στοιχεία του πίνακα που επέλεξε. Έτσι ένας χρήστης μπορεί να διαβάσει όλα τα δεδομένα που περιέχει ένας πίνακας στη βάση δεδομένων μας:

executive_id	first_name	last_name	date_of_birth	hfri_governing_body	actions
<input type="text" value="1"/>	<input type="text" value="Adam"/>	<input type="text" value="Vaughan"/>	<input type="text" value="1975-06-01"/>	<input type="text" value="Deputy Directors"/>	<input type="button" value="Add"/>
1	Adam	Vaughan	1975-06-01	Deputy Directors	Edit Delete
2	Jaime	Simpson	1955-05-14	Advisory Committee	Edit Delete
3	Andrea	Wu	1983-12-12	Deputy Directors	Edit Delete
4	Steven	Mcdowell	1977-10-08	Deputy Directors	Edit Delete
5	Frederick	Garcia	1981-09-19	Deputy Directors	Edit Delete
6	Caitlyn	King	1972-11-15	Deputy Directors	Edit Delete
7	Tammy	Bell	1986-05-15	General Assembly	Edit Delete
8	Amanda	Vaughn	1957-03-12	Advisory Committee	Edit Delete

Στο πάνω μέρος αυτής της ιστοσελίδας βρίσκονται κενά πεδία (input type="text") και ένα κουμπί **Add**. Γράφοντας ο χρήστης σε αυτά τα πεδία και πατώντας το κουμπί Add μπορεί να εισάγει δεδομένα στη βάση μας για όποιο πίνακα επιθυμεί. Όταν ο χρήστης πατάει το κουμπί εκτελείται μια εντολή **insert** χρησιμοποιώντας ως μεταβλητές τα δεδομένα που εισήγαγε ο χρήστης στα κενά πεδία. Αν τα πεδία έχουν συμπληρωθεί σωστά και δεν προκαλούν κάποιο σφάλμα με βάση τους περιορισμούς που αναλύθηκαν στο μέρος 2 τότε νέα δεδομένα εισάγονται στη βάση.

Στη δεξιά στήλη υπάρχει ένας υπερσύνδεσμος **Edit** και από κάτω του ένας υπερσύνδεσμος **Delete**. Πατώντας για μία γραμμή του πίνακα ο χρήστης τον υπερσύνδεσμο Edit τότε ανακατευθύνεται στην ιστοσελίδα **edit.html**:

executive_id	first_name	last_name	date_of_birth	hfri_governing_body	actions
<input type="text" value="1"/>	<input type="text" value="Adam"/>	<input type="text" value="Vaughan"/>	<input type="text" value="1975-06-01"/>	<input type="text" value="Deputy Directors"/>	<input type="button" value="Update"/>

[Back to table](#)

Εκεί εμφανίζονται πεδία κειμένου (input type="text") τα οποία έχουν σαν τιμές αυτές που έχουν στον πίνακα της προηγούμενης ιστοσελίδας **view.html**, καθώς και ένα κουμπί με τη λέξη update. Αν ο χρήστης αλλάξει τα αρχικά δεδομένα που του εμφανίζονται και πατήσει το κουμπί Update τότε μέσω της συνάρτησης edit στο αρχείο routes.py παίρνει τις τιμές που εισήγαγε σαν μεταβλητές και εκτελεί μία εντολή τύπου Update με τις νέες τιμές. Έτσι οι νέες τιμές ενημερώνονται στη βάση δεδομένων μας. Επίσης, στην ιστοσελίδα edit.html υπάρχει υπερσύνδεσμος "Back to table", ο οποίος

επιστρέφει τον χρήστη στον προηγούμενο πίνακα και μπορεί έτσι να παρατηρήσει τις ενημερώσεις που έκανε στην γραμμή που επέλεξε. Τέλος, αν πατήσει τον υπερσύνδεσμο Delete, τότε μέσω της συνάρτησης `table_view` του αρχείου `routes.py` εκτελείται παίρνει ως μεταβλητές τις τιμές της συγκεκριμένης γραμμής και εκτελεί μια `Delete sql` εντολή. Με αυτό τον τρόπο διαγράφονται δεδομένα από τη βάση δεδομένων μας.