ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΜΑΘΗΜΑ: Βάσεις Δεδομένων

Εαρινό Εξάμηνο 2021-2022

Αναφορά Εξαμηνιαίας Εργασίας

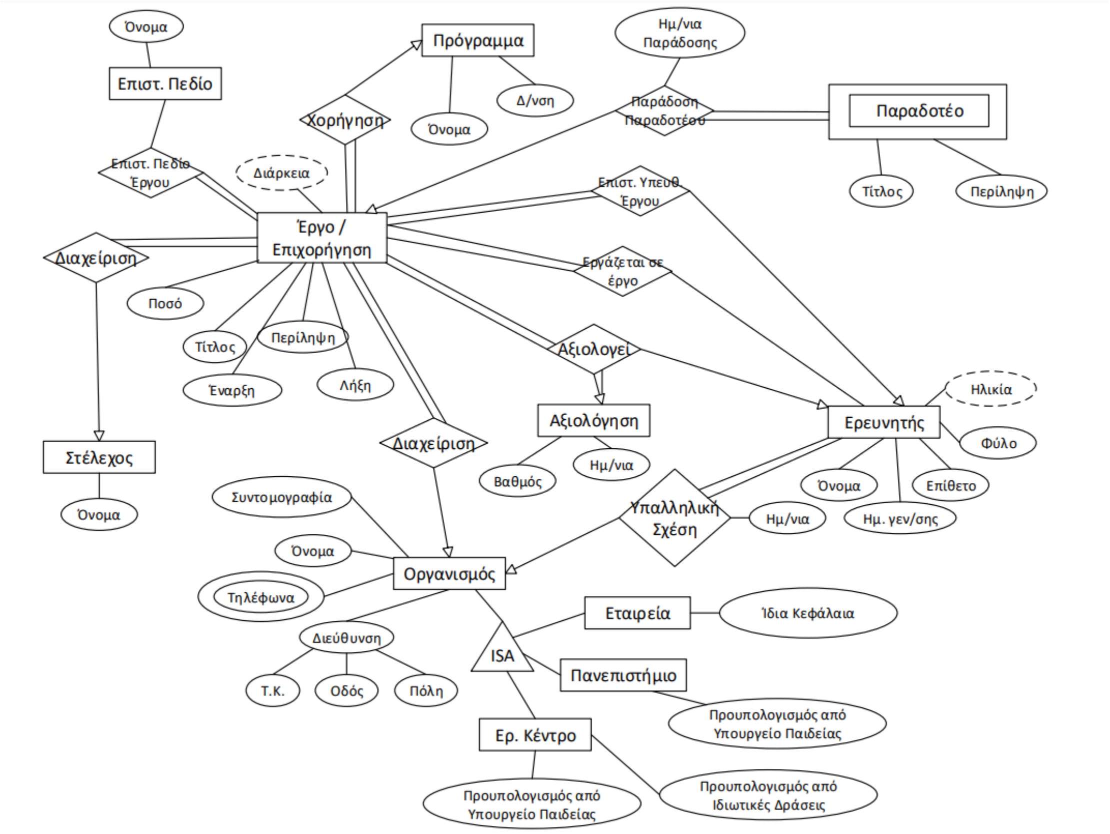
Ομάδα Project 47

1. Κοτσομπόλης Γεώργιος ge17013
2. Μαυρωτάς Άρης el18640
3. Παπαναστασίου Αλέξανδρος el17647

[ΑΘΗΝΑ, ΙΟΥΝΙΟΣ/2022]

2.

Το ER που επιλέχθηκε είναι κυρίως αυτό που προτάθηκε από τους διδάσκοντες, το οποίο φαίνεται παρακάτω, ενώ έχουμε πραγματοποιήσει και μερικές διαφοροποιήσεις επηρεαζόμενοι από το δικό μας μοντέλο οντοτήτων-συσχετίσεων.



Καθώς δεν έχουν σημειωθεί τα πρωτεύοντα κλειδιά έχουμε προσθέσει εμείς αυτά που θεωρούμε.

Επίσης η αξιολόγηση δεν έχει θεωρηθεί ως οντότητα αλλά βρίσκεται μέσα στο έργο.

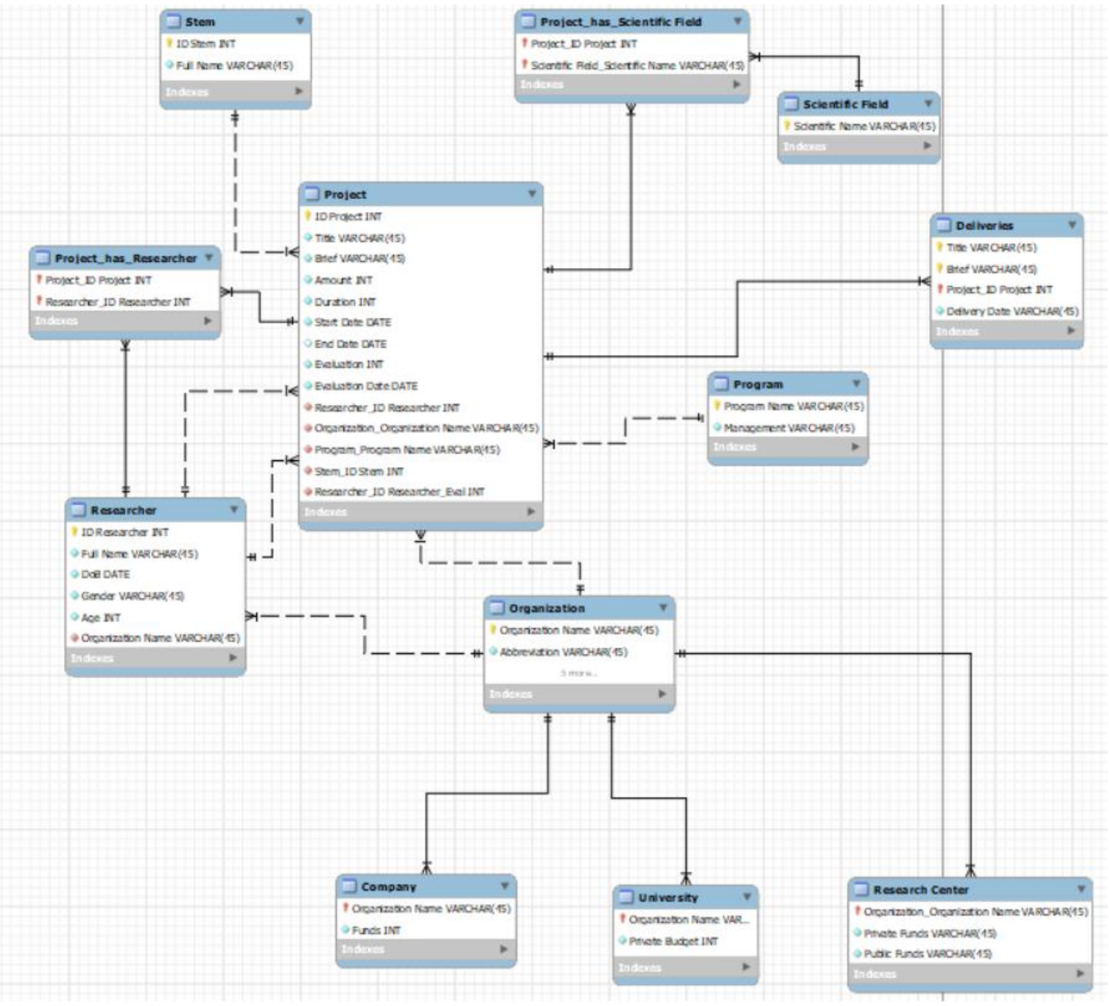
Τα πράγματα φαίνονται πιο ξεκάθαρα παρακάτω στο σχεσιακό μοντέλο και στο EER που δημιουργήθηκε στο MySQL Workbench.

Για την δημιουργία του σχεσιακού μοντέλου χρησιμοποιήθηκαν όλοι οι γνωστοί κανόνες μετατροπής του από το ER.

Σχεσιακό Μοντέλο

- Stem (ID Stem, Full Name)
- Scientific Field (Scientific Name)
- Program (Program Name, Management)
- Organization (Organization Name, Abbreviation, Telephones, City, Address Name, Address Number, Postal Code)
- Company (Organization Name, Funds), Organization Name **FK** στο Organization
- University (Organization Name, Private Budget), Organization Name **FK** στο Organization
- Research Center (Organization Name, Private Budget, Public Budget), Organization Name **FK** στο Organization
- Researcher (ID Researcher, Full Name, DoB, Gender, Age, Organization Name), με Organization Name **FK** στο Organization
- Project (ID Project, Title, Brief, Amount, Duration, Start Date, End Date, Evaluation, Evaluation Date, ID Researcher, Organization Name, Program Name, ID Stem, Researcher_ID Researcher_Eval), με ID Researcher **FK** στο Researcher, Organization Name **FK** στο Organization, Program Name **FK** στο Program, ID Stem **FK** στο Stem και Researcher_ID Researcher_Eval FK στο Researcher
- Deliveries (Title, Brief, Delivery Date, ID Project), με ID Project **FK** στο Project
- Project has Researcher (ID Project, ID Researcher), με ID Project **FK** στο Project και ID Researcher **FK** στο Researcher
- Project has Scientific Field (ID Project , Scientific Name), με ID Project **FK** στο Project και Scientific Name **FK** στο Scientific Field

EER WORKBENCH



Οι περιορισμοί που έχουν οριστεί παρατίθενται παρακάτω:

- Περιορισμοί ακεραιότητας

● Οι περιορισμοί που αφορούν τα κλειδιά φαίνονται φαίνονται στο σχεσιακό μοντέλο στην προηγούμενη σελίδα.

● Not null: έχουν μπει σε όλα τα τα attributes, εκτός από τα end dates των projects, το οποίο δείχνει αν το project είναι ακόμα ενεργό (null), ή αν έχει τερματιστεί.

Θα μπορούσαμε να παίξουμε και με unique/check constraints αλλά εν τέλει τα θεωρήσαμε μη απαραίτητα για την υλοποίηση της βάσης.

- Αναφορική ακεραιότητα

Στα foreign keys έχουμε κάνει χρήση του cascade στις λειτουργίες update και delete. Όταν θέλουμε να πειράξουμε ένα attribute σε ένα parent table, τότε αυτόματα θα πειραχτούν και τα αντίστοιχα attributes στα child tables. Θα μπορούσαμε να πραγματοποιήσουμε παρόμοια λειτουργία με χρήση triggers, αλλά υπάρχει μεγαλύτερη πιθανότητα να παρουσιαστεί κάποιο μη θελητό πρόβλημα.

- Ακεραιότητα πεδίου τιμών

Κάθε attribute έχει συγκεκριμένο τύπο για να αντιστοιχεί στις τιμές που παίρνει.

- Triggers

Τα triggers είναι ορισμένες “αντιδράσεις” που γίνονται σε καθορισμένες ενέργειες που κάνουμε στη βάση. Πολλές φορές διευκολύνουν κάποιες καταστάσεις, αλλά υπάρχει και κίνδυνος να παρουσιάζουν προβλήματα τα οποία να μην τα έχουμε σκεφτεί κατά την υλοποίηση τους.

Έχουμε υλοποιήσει ορισμένα που αφορούν τη διάρκεια ενός έργου και εάν αυτή βρίσκεται μέσα σε κάποιες ορισμένες τιμές, εάν η ημερομηνία λήξης έχει παρέλθει και αν το ποσό επιχορήγησης βρίσκεται μέσα σε ορισμένες τιμές και για το ότι ο αξιολογητής δε δουλεύει στο αντίστοιχο έργο.

-Ευρετήρια

Για τα πρωτεύοντα κλειδιά ορίζονται αυτόματα από την MySQL.

Τα ευρετήρια που έχουν οριστεί από εμάς αφορούν τα έργα ανά ερευνητή και το συνολικό ποσό για κάθε επιστημονικό πεδίο. Έχουν οριστεί για attributes όπως το ποσό επιχορήγησης, τη ημερομηνία γέννησης του ερευνητή και την ημερομηνία που ξεκίνησε ένα έργο. Όλα αυτά τα attributes τα αναζητούμε σε διάφορες στιγμές οπότε μας είναι χρήσιμο να υπάρχουν ευρετήρια και παρακάτω στον κώδικα φαίνεται που χρησιμεύει το καθένα.

-Δεδομένα

Κάναμε χρήση του Mockaroo για να παράξουμε δεδομένα και ενδεικτικά στον κώδικα φαίνεται η εισαγωγή κάποιων από αυτών.
Αναλυτικά τα αρχεία υπάρχουν στο GitHub.

-Τεχνολογίες Εφαρμογής

Η δημιουργία της βάσης έγινε μέσω του Workbench και για την εφαρμογή κάναμε χρήση της python με τις παρακάτω βιβλιοθήκες. Έπειτα από την εγκατάσταση της python γίνεται και αυτή των πακέτων μέσω του pip. Αναλυτικότερα τα βήματα υπάρχουν στον σύνδεσμο στο τέλος της αναφοράς.

- mysql 8.0.19
- flask 1.1.2
- mysql connector 2.2.9

Ακόμα για το Front End της εφαρμογής έγινε χρήση HTML και CSS.

Για την χρήση ανοίγουμε τον περιηγητή της επιλογής μας και γράφουμε
http://localhost:5000

Παρακάτω φαίνονται οι κώδικες για την δημιουργία της βάσης. Αφού φτιάχτηκε πρώτα το μοντέλο γραφικά στο MySQL Workbench έπειτα με forward engineering κατασκευάστηκε η βάση. (Κατά τη διαδικασία αφαιρέσαμε την εντολή VISIBLE).

3.

DDL & DML SCRIPTS

Creation of the database:

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO
_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE
_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema Elidek
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema Elidek
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `Elidek` DEFAULT CHARACTER SET utf8 ;
USE `Elidek` ;

-- -----------------------------------------------------
-- Table `Elidek`.`Stem`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Stem` (
  `ID Stem` INT NOT NULL AUTO_INCREMENT,
  `Full Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID Stem`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `Elidek`.`Scientific Field`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Scientific Field` (
  `Scientific Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Scientific Name`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`Program`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Program` (
  `Program Name` VARCHAR(45) NOT NULL,
  `Management` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Program Name`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`Organization`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Organization` (
  `Organization Name` VARCHAR(45) NOT NULL,
  `Abbreviation` VARCHAR(45) NOT NULL,
  `Telephones` INT NOT NULL,
  `City` VARCHAR(45) NOT NULL,
  `Address Name` VARCHAR(45) NOT NULL,
  `Address Number` INT NOT NULL,
  `Postal Code` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Organization Name`))
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `Elidek`.`Company`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Company` (
  `Organization Name` VARCHAR(45) NOT NULL,
  `Funds` INT NOT NULL,
  PRIMARY KEY (`Organization Name`),
  INDEX `fk_Company_Organization_idx` (`Organization Name` ASC) ,
  CONSTRAINT `fk_Company_Organization`
    FOREIGN KEY (`Organization Name`)
    REFERENCES `Elidek`.`Organization` (`Organization Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`University`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`University` (
  `Organization Name` VARCHAR(45) NOT NULL,
  `Private Budget` INT NOT NULL,
  PRIMARY KEY (`Organization Name`),
  CONSTRAINT `fk_University_Organization1`
    FOREIGN KEY (`Organization Name`)
    REFERENCES `Elidek`.`Organization` (`Organization Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`Research Center`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Research Center` (
  `Organization_Organization Name` VARCHAR(45) NOT NULL,
  `Private Funds` VARCHAR(45) NOT NULL,
  `Public Funds` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Organization_Organization Name`),
  CONSTRAINT `fk_Research Center_Organization1`
    FOREIGN KEY (`Organization_Organization Name`)
    REFERENCES `Elidek`.`Organization` (`Organization Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`Researcher`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Researcher` (
  `ID Researcher` INT NOT NULL AUTO_INCREMENT,
  `Full Name` VARCHAR(45) NOT NULL,
  `DoB` DATE NOT NULL,
  `Gender` VARCHAR(45) NOT NULL,
  `Age` INT NOT NULL,
  `Organization Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID Researcher`),
  INDEX `fk_Researcher_Organization1_idx` (`Organization Name` ASC) ,
  CONSTRAINT `fk_Researcher_Organization1`
    FOREIGN KEY (`Organization Name`)
    REFERENCES `Elidek`.`Organization` (`Organization Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `Elidek`.`Project`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Project` (
  `ID Project` INT NOT NULL AUTO_INCREMENT,
  `Title` VARCHAR(45) NOT NULL,
  `Brief` VARCHAR(45) NOT NULL,
  `Amount` INT NOT NULL,
  `Duration` INT NOT NULL,
  `Start Date` DATE NOT NULL,
  `End Date` DATE NOT NULL,
  `Evaluation` INT NOT NULL,
  `Evaluation Date` DATE NOT NULL,
  `Researcher_ID Researcher` INT NOT NULL,
  `Organization_Organization Name` VARCHAR(45) NOT NULL,
  `Program_Program Name` VARCHAR(45) NOT NULL,
  `Stem_ID Stem` INT NOT NULL,
  PRIMARY KEY (`ID Project`),
  INDEX `fk_Project_Researcher1_idx` (`Researcher_ID Researcher` ASC) ,
  INDEX `fk_Project_Organization1_idx` (`Organization_Organization Name` ASC) ,
  INDEX `fk_Project_Program1_idx` (`Program_Program Name` ASC) ,
  INDEX `fk_Project_Stem1_idx` (`Stem_ID Stem` ASC) ,
  CONSTRAINT `fk_Project_Researcher1`
    FOREIGN KEY (`Researcher_ID Researcher`)
    REFERENCES `Elidek`.`Researcher` (`ID Researcher`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Project_Organization1`
    FOREIGN KEY (`Organization_Organization Name`)
    REFERENCES `Elidek`.`Organization` (`Organization Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Project_Program1`
    FOREIGN KEY (`Program_Program Name`)
    REFERENCES `Elidek`.`Program` (`Program Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Project_Stem1`
    FOREIGN KEY (`Stem_ID Stem`)
    REFERENCES `Elidek`.`Stem` (`ID Stem`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`Deliveries`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Deliveries` (
  `Title` VARCHAR(45) NOT NULL,
  `Brief` VARCHAR(45) NOT NULL,
  `Project_ID Project` INT NOT NULL,
  `Delivery Date` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Title`, `Brief`, `Project_ID Project`),
  INDEX `fk_Deliveries_Project1_idx` (`Project_ID Project` ASC) ,
  CONSTRAINT `fk_Deliveries_Project1`
    FOREIGN KEY (`Project_ID Project`)
    REFERENCES `Elidek`.`Project` (`ID Project`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Elidek`.`Project_has_Researcher`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Project_has_Researcher` (
  `Project_ID Project` INT NOT NULL,
  `Researcher_ID Researcher` INT NOT NULL,
  PRIMARY KEY (`Project_ID Project`, `Researcher_ID Researcher`),
  INDEX `fk_Project_has_Researcher_Researcher1_idx` (`Researcher_ID
Researcher` ASC) ,
  INDEX `fk_Project_has_Researcher_Project1_idx` (`Project_ID Project` ASC) ,
  CONSTRAINT `fk_Project_has_Researcher_Project1`
    FOREIGN KEY (`Project_ID Project`)
    REFERENCES `Elidek`.`Project` (`ID Project`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Project_has_Researcher_Researcher1`
    FOREIGN KEY (`Researcher_ID Researcher`)
    REFERENCES `Elidek`.`Researcher` (`ID Researcher`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `Elidek`.`Project_has_Scientific Field`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Elidek`.`Project_has_Scientific Field` (
  `Project_ID Project` INT NOT NULL,
  `Scientific Field_Scientific Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Project_ID Project`, `Scientific Field_Scientific Name`),
  INDEX `fk_Project_has_Scientific Field_Scientific Field1_idx` (`Scientific
Field_Scientific Name` ASC) ,
  INDEX `fk_Project_has_Scientific Field_Project1_idx` (`Project_ID Project` ASC) ,
  CONSTRAINT `fk_Project_has_Scientific Field_Project1`
    FOREIGN KEY (`Project_ID Project`)
    REFERENCES `Elidek`.`Project` (`ID Project`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Project_has_Scientific Field_Scientific Field1`
```

```
    FOREIGN KEY (`Scientific Field_Scientific Name`)
    REFERENCES `Elidek`.`Scientific Field` (`Scientific Name`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

ALTER TABLES:

```
ALTER TABLE `elidek`.`project`
CHANGE COLUMN `End Date` `End Date` DATE NULL ;

ALTER TABLE `elidek`.`researcher`
DROP COLUMN `Age` ;

ALTER TABLE `elidek`.`project`
ADD COLUMN `Researcher_ID Researcher_Eval` INT(11) NOT NULL AFTER `Stem_ID Stem`,
ADD INDEX `fk_Project_Researcher2_idx` (`Researcher_ID Researcher_Eval` ASC);
;
ALTER TABLE `elidek`.`project`
ADD CONSTRAINT `fk_Project_Researcher2`
  FOREIGN KEY (`Researcher_ID Researcher_Eval`)
  REFERENCES `elidek`.`researcher` (`ID Researcher`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

VIEWS:

View 1:

```
-- View with researchers and project which work for
CREATE VIEW projects_for_each_researcher AS
SELECT `Full Name`,`Project_ID Project`
FROM researcher,project_has_researcher
WHERE `Researcher_ID Researcher` = `ID Researcher`
ORDER BY `Full Name`
```

View 2:

```
--Find total amount for each scientific field
CREATE VIEW total_amount_for_each_field AS
SELECT SUM(Amount),`Scientific Field_Scientific Name`
FROM `project_has_scientific field`,project
WHERE `ID Project` = `Project_ID Project`
GROUP BY `Scientific Field_Scientific Name`
ORDER BY Amount desc
```

INDECES:

```
CREATE INDEX project_amount on project(amount);
-- To find the amount easily

CREATE INDEX start_date on project(`Start Date`);
-- For 3.4

CREATE INDEX DoB_for_researcher on researcher(DoB);
-- To find age of a researcher (3.1, 3.6)
```

TRIGGERS:

Trigger 1:

```
delimiter $$
CREATE TRIGGER amount_error
```

```
BEFORE INSERT on Project
for each row
begin
if (Project.Amount < 100000)
then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Funding is too low';
end if;
if (Project.Amount > 1000000)
then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Funding is too high';
end if;
end;
$$
```

Trigger 2:

```
delimiter $$
CREATE TRIGGER deadline_passed
AFTER INSERT on Project
for each row
begin
if (curdate() > Project.`End Date`)
then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Deadline has passed';
end if;
end;
$$
```

Trigger 3:

```
delimiter $$
CREATE TRIGGER timespan_check
BEFORE INSERT on project
for each row
begin
        if (project.Duration < 1)
        then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'The project has
too short of a lifespan' ;
   end if;
```

```
	if (project.Duration > 4)
	then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'The project has too long of a lifespan' ;
	end if;
end;
$$
```

Trigger 4:

```
delimiter $$
CREATE TRIGGER evaluator
BEFORE INSERT on Project
for each row
begin
if (project.`Researcher_ID Researcher_Eval` =
project_has_researcher.`Researcher_ID Researcher` )
then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Evaluator must not work on the project';
end if;
end;
$$
```

INSERTIONS:

```
insert into Stem   values (1, 'Kotsompolis Georgios');
insert into Stem   values (2, 'Gklavani Maria');
insert into Stem   values (3, 'Ioannis Ladopoulos');
insert into Stem   values (4, 'Clara Cromarty');
insert into Stem   values (5, 'Alta Binstead');
insert into Stem   values (6, 'Catlaina MacCosty');
insert into Stem   values (7, 'Lynsey Vosper');
insert into Stem   values (8, 'Rene Osinin');
insert into Stem   values (9, 'Binnie Howsam');
insert into Stem   values (10, 'Ganny Raddenbury');

insert into Stem   values (11, 'Ara Hutcheson');

insert into Stem   values (12, 'Ewan Stutter');

insert into Stem   values (13, 'Ingra Cholomin');

insert into Stem   values (14, 'Phillipp Simchenko');

insert into Stem   values (15, 'Hermann St. Queintain');

insert into Stem  values (16, 'Rayshell Salan');

insert into Stem   values (17, 'Olenolin Poacher');

insert into Stem   values (18, 'Pembroke Knowlden');

insert into Stem   values (19, 'Billi Canepe');

insert into Stem   values (20, 'Leodora Dominiak');

….

INSERT INTO `scientific field`

VALUES

('AI'),

('Machine Learning'),

('Information Technology'),

('Computer Engineering'),

('Engineering'),

('Mechanical Engineering'),

('Electrical Engineering'),

('Chemical Engineering'),

('Modern Engineering');

…

insert into organization  values ('Bergstrom-Brown', '03', '7442987166', 'Manhete',
'Fair Oaks', '99519', '4750-549');

insert into organization  values ('Pfeffer-Langworth', 'CA', '5599370625', 'Fresno',
'Shelley', '88', '93726');

insert into organization values ('Quitzon Group', '03', '5963116464', 'Donim',
'Colorado', '57', '4805-178');

insert into organization  values ('Quigley-Huel', 'CN', '2978279395', 'Telde',
'Westport', '7045', '35213');
```

```
insert into organization  values ('King, Gerlach and Rowe', 'CA', '8051024742', 'Van Nuys', 'Debra', '01014', '91406');

insert into organization  values ('Frami, Lynch and Botsford', 'CA', '6264056260', 'Van Nuys', 'Fisk', '641', '91406');

insert into organization  values ('Zulauf LLC', '02', '3184929447', 'Odivelas', 'Arizona', '660', '7900-374');

insert into organization  values ('Vandervort and Sons', 'BZ', '1144657854', 'Ózd', 'Dawn', '192', '3607');

insert into organization  values ('Becker LLC', '01', '3501799542', 'Quinta', 'Cody', '10591', '4520-509');

insert into organization  values ('Gerlach and Sons', 'BA', '5747586591', 'Pécs', 'Kinsman', '1700', '7646');

insert into organization  values ('Koss, Bartell and Kunze', 'BA', '5066018146', 'Pécs', 'Forster', '63366', '7646');

insert into organization  values ('Hegmann, Orn and Robel', 'CA', '7071326299', 'Santa Rosa', 'Trailsway', '2', '95405');

insert into organization  values ('Langosh, Oberbrunner and Hamill', '03', '3367590665', 'Rio Covo', 'Green Ridge', '291', '4755-463');
insert into company
values
        ('Altenwerth-Gibson',10000),
   ('Becker LLC',100000),
   ('Bergstrom-Brown',100000),
   ('Boehm-Morar',10000),
   ('Boyle, Welch and Block',5000),
   ('Carroll Group',100000),
   ('Conn, Corwin and Erdman',1000000),
   ('Daniel-Flatley',100000),
   ('Fahey, Gaylord and Mills',50000),
   ('Farrell LLC',40000),
   ('Frami, Lynch and Botsford',30000),
   ('Gerlach and Sons',120000),
   ('Gottlieb, Williamson and Boyle',140000)
…
INSERT INTO Project
VALUES

        (1, 'Dokimastiko', 'Kai alli mia dokimi', 100000, 10, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',1,105),
        (2, 'Xvris', 'Kai alli mia dokimi', 100000, 15, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',1,105),
        (3, 'Space exp', 'Kai alli mia dokimi', 100000, 12, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',1,105),
        (4, 'GTA', 'Kai alli mia dokimi', 100000, 2, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',1,105),
        (5, 'DD', 'Kai alli mia dokimi', 100000, 3, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',1,105),
        (6, 'Allo', 'Kai alli mia dokimi', 100000, 2, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',1,105),
        (7, 'Allo', 'Kai alli mia dokimi', 150000, 3, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',2,105),
        (8, 'Allo', 'Kai alli mia dokimi', 150000, 3, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',2,105),
        (9, 'Allo', 'Kai alli mia dokimi', 150000, 3, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',2,105),
        (10, 'Allo', 'Kai alli mia dokimi', 150000, 3, '2019-01-26', '2022-03-26',5,'2022-02-26',1,'Becker LLC','Science Development',2,105)
(56, 'Allo', 'Kai alli mia dokimi', 500000, 6, '2019-01-26', '',5,'2022-02-26',7,'King, Gerlach and Rowe','Machine Learning Program',15,104),
        (57, 'Allo', 'Kai alli mia dokimi', 110000, 6, '2019-01-26', '',5,'2022-02-26',7,'King, Gerlach and Rowe','Machine Learning Program',15,104)
…
INSERT INTO project_has_researcher values
        (1,1),
  (1,2),
  (1,3),
  (2,1),
  (2,2),
  (2,45),
  (2,42),
  (3,1),
  (3,2),
  (4,1),
  (4,43)
…
insert into program
values
('Science Program','Develpoment'),
('Modern Sciences','Development'),
('Youth Employment','Employment'),
('Renewable Energy','Energy Development'),
('Buildings','Cunstructions'),
('Biomidicine','Medicine'),
```

('Medical Drugs','Medicine')

…

…

QUERIES:

3,3

```
select  ps.`Scientific Field_Scientific Name`,ps.`Project_ID Project`, p.`Researcher_ID Researcher`

from `project_has_scientific field` ps, project p

where `Project_ID Project` = `ID Project` and `End Date` is null

order by `Scientific Field_Scientific Name` ;
```

3.4

```
select          temp1.`Organization_Organization Name`

from    (       select count(`ID Project`) as counter,`Organization_Organization Name`,year(`Start Date`) as yr

                        from project

                        group by `Organization_Organization Name`,year(`Start Date`) ) as temp1,

                (       select count(`ID Project`) as counter,`Organization_Organization Name`,year(`Start Date`) as yr

                        from project

                        group by `Organization_Organization Name`,year(`Start Date`) ) as temp2

where temp1.`Organization_Organization Name` = temp2.`Organization_Organization Name` and temp1.counter = temp2.counter and temp1.counter>=10 and ( (temp1.yr - temp2.yr) = 1 )

group by `Organization_Organization Name`;
```

3.5

```
SELECT a1,b1,times_played,amount


FROM (

        SELECT
```

```
            a.`Project_ID Project`,
            a.`Scientific Field_Scientific Name` AS a1,
            b.`Scientific Field_Scientific Name` AS b1,
            COUNT(*) AS times_played,
            RANK() OVER (ORDER BY COUNT(*) DESC) Rnk
        FROM `project_has_scientific field` a
        JOIN `project_has_scientific field` b
        ON a.`Scientific Field_Scientific Name` < b.`Scientific Field_Scientific Name` AND a.`Scientific Field_Scientific Name` != b.`Scientific Field_Scientific Name`
        GROUP by 1,2
) sub , project
WHERE `ID Project` = `Project_ID Project`
LIMIT 3
;
```

3.6

```
SELECT
  `Researcher_ID Researcher`,
  COUNT(`Project_ID Project`) AS `value_occurrence`
FROM
  project_has_researcher, researcher
WHERE DoB > CURRENT_DATE - INTERVAL 40 YEAR and `ID Researcher` = `Researcher_ID Researcher`
GROUP BY
  `Researcher_ID Researcher`
ORDER BY
  `value_occurrence` DESC
LIMIT 3;
```

3.7

```sql
select `Full Name`,`Stem_ID Stem`,Amount,`Organization_Organization Name`
from     project, company, stem
where `Organization_Organization Name` = `Organization Name` and `ID Stem` = `Stem_ID Stem`
order by Amount desc
limit 5;
```

3.8

```sql
select `Full Name`, count(`Researcher_ID Researcher`)
from project_has_researcher, researcher
where `Project_ID Project` not in(select `Project_ID Project` from deliveries) and `Researcher_ID Researcher` = `ID Researcher`
group by `Researcher_ID Researcher`
having count(`Researcher_ID Researcher`)>=5
```

Σύνδεσμος GitHub:

https://github.com/arismavro5/DB-2022-Group-47