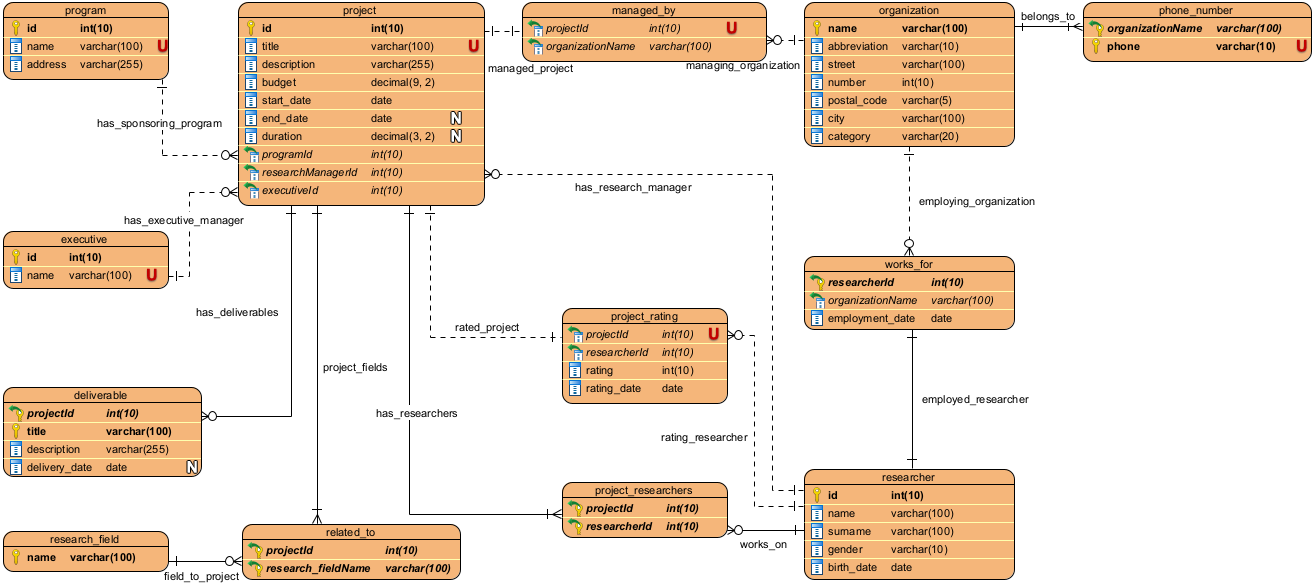
**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ 2022**

Κωνσταντίνος Διβριώτης  
03114140  
Ομάδα Project 19

**Σχεσιακό Διάγραμμα της Βάσης Δεδομένων**



1. **Στελέχη που δουλεύουν για τον ΕΛ.ΙΔ.Ε.Κ. (executive)**

**CREATE TABLE `executive` (**

**`id` int NOT NULL AUTO\_INCREMENT COMMENT 'Μοναδικό αναγνωριστικό στελέχους',**

**`name` varchar(100) NOT NULL COMMENT 'Όνομα στελέχους (μοναδικό)',**

**PRIMARY KEY (`id`),**

**UNIQUE KEY `name` (`name`)**

**) ENGINE = InnoDB AUTO\_INCREMENT = 16 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Στελέχη ΕΛ.ΙΔ.Ε.Κ.';**

Για κάθε στέλεχος υπάρχει ένα μοναδικό αναγνωριστικό (**id**) το οποίο είναι ένας αύξον αριθμός, ο οποίος παράγεται αυτόματα από τη βάση, και ένα πεδίο για το όνομα του στελέχους (**name**) ) (*μέχρι 100 χαρακτήρες*), το οποίο θεωρούμε ότι είναι μοναδικό.

1. **Προγράμματα που υλοποιούνται από τον ΕΛ.ΙΔ.Ε.Κ. και χρηματοδοτούν έργα (program)**

**CREATE TABLE `program` (**

**`id` int NOT NULL AUTO\_INCREMENT COMMENT 'Μοναδικό αναγνωριστικό του προγράμματος',**

**`name` varchar(100) NOT NULL COMMENT 'Όνομα προγράμματος',**

**`address` varchar(255) NOT NULL COMMENT 'Διεύθυνση',**

**PRIMARY KEY (`id`),**

**UNIQUE KEY `name` (`name`)**

**) ENGINE = InnoDB AUTO\_INCREMENT = 41 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Προγράμματα που χρηματοδοτούν έργα';**

Για κάθε πρόγραμμα υπάρχει ένα μοναδικό αναγνωριστικό (**id**) το οποίο είναι ένας αύξον αριθμός, ο οποίος παράγεται αυτόματα από τη βάση, ένα πεδίο για το όνομα του προγράμματος (**name**) (*μέχρι 100 χαρακτήρες*), το οποίο θεωρούμε ότι είναι μοναδικό και ένα πεδίο για τη διεύθυνση του ΕΛ.ΙΔ.Ε.Κ. στην οποία ανήκει (*μέχρι 255 χαρακτήρες*).

1. **Οργανισμοί που διαχειρίζονται έργα (organization)**

**CREATE TABLE `organization` (**

**`name` varchar(100) COMMENT 'Μοναδικό όνομα του οργανισμού',**

**`abbreviation` varchar(10) NOT NULL COMMENT 'Συντομογραφία',**

**`street` varchar(100) NOT NULL COMMENT 'Διεύθυνση οργανισμού: Οδός',**

**`number` int NOT NULL COMMENT 'Διεύθυνση οργανισμού: Αριθμός',**

**`postal\_code` varchar(5) NOT NULL COMMENT 'Διεύθυνση οργανισμού: Τ.Κ.',**

**`city` varchar(100) NOT NULL COMMENT 'Διεύθυνση οργανισμού: Πόλη',**

**`category` varchar(20) NOT NULL COMMENT 'Κατηγορία: Ερευνητικό Κέντρο(Research Center), Πανεπιστήμιο (University) ή Εταιρεία (Company)',**

**PRIMARY KEY (`name`),**

**CONSTRAINT `check\_category` CHECK (**

**(**

**`category` in (**

**\_utf8mb4 'University',**

**\_utf8mb4 'Company',**

**\_utf8mb4 'Research Center'**

**)**

**)**

**)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Οργανισμός';**

Για κάθε οργανισμό υπάρχει ένα πεδίο για το όνομα του οργανισμού (**name**) (*μέχρι 100 χαρακτήρες*), το οποίο θεωρούμε ότι είναι μοναδικό και συνεπώς χρησιμοποιείται ως κύριο κλειδί (*primary key*), ένα πεδίο για τη συντομογραφία (**abbreviation**) (*μέχρι 10 χαρακτήρες*). Επίσης υπάρχουν πληροφορίες για τη διεύθυνση του οργανισμού, η οποία περιλαμβάνει την οδό (**street** - *μέχρι 100 χαρακτήρες*), τον αριθμό (**number**), τον ταχυδρομικό κώδικα (**postal\_code**) και την πόλη (**city**). Ακόμη υπάρχει ένα πεδίο που δείχνει σε ποια κατηγορία ανήκει ο οργανισμός (**category**), το οποίο παίρνει τιμές **University**, **Company** ή **Research Center**, ανάλογα αν ο οργανισμός είναι Πανεπιστήμιο, Εταιρεία ή Ερευνητικό Κέντρο αντίστοιχα. Ο έλεγχος εγκυρότητας του πεδίου γίνεται με τη βοήθεια του *CHECK CONSTRAINT* **check\_category**.

Τέλος, κάθε οργανισμός έχει ένα ή περισσότερα τηλέφωνα επικοινωνίας, τα οποία αποθηκεύονται με τη βοήθεια ενός ακόμη table με όνομα **phone\_number**, στο οποίο υπάρχουν εγγραφές για κάθε οργανισμό (**organizationName** - *FOREIGN KEY* στο πεδίο **name** του πίνακα ***organization***) μαζί με έναν αριθμό τηλεφώνου (**phone**) ο οποίος ανήκει στον οργανισμό. Θεωρούμε ότι κάθε αριθμός είναι μοναδικός, συνεπώς ορίζουμε ως κύριο κλειδί τον οργανισμό μαζί με τον αριθμό.

Επίσης ελέγχουμε την εγκυρότητα του αριθμού (**phone**) με τη βοήθεια του *CHECK CONSTRAINT* **check\_phone** για να σιγουρευτούμε ότι έχει μήκος 10 χαρακτήρες και αποτελείται αποκλειστικά από ψηφία.

**CREATE TABLE `phone\_number` (**

**`organizationName` varchar(100) NOT NULL COMMENT 'Όνομα του οργανισμού στον οποίο ανήκει ο αριθμός',**

**`phone` varchar(10) NOT NULL COMMENT 'Αριθμός τηλεφώνου του οργανισμού (μοναδικός) - Ακριβώς 10 χαρακτήρες',**

**PRIMARY KEY (`organizationName`, `phone`),**

**UNIQUE KEY `phone\_UNIQUE` (`phone`),**

**CONSTRAINT `belongs\_to` FOREIGN KEY (`organizationName`) REFERENCES `organization` (`name`),**

**CONSTRAINT `check\_phone` CHECK (regexp\_like(`phone`,\_utf8mb4'^[0-9]{10}$'))**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Τηλεφωνικοί αριθμοί (τουλάχιστον 1 για κάθε οργανισμό)';**

1. **Ερευνητές που εργάζονται σε έργα (researcher)**

**CREATE TABLE `researcher` (**

**`id` int NOT NULL AUTO\_INCREMENT COMMENT 'Μοναδικό αναγνωριστικό του ερευνητή',**

**`name` varchar(100) NOT NULL COMMENT 'Όνομα',**

**`surname` varchar(100) NOT NULL COMMENT 'Επίθετο',**

**`gender` varchar(10) NOT NULL COMMENT 'Φύλο (male ή female)',**

**`birth\_date` date NOT NULL COMMENT 'Ημερομηνία Γέννησης',**

**PRIMARY KEY (`id`),**

**CONSTRAINT `check\_gender` CHECK ((`gender` in (\_utf8mb4 'male', \_utf8mb4 'female')))**

**) ENGINE = InnoDB AUTO\_INCREMENT = 111 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Ερευνητές που εργάζονται σε έργα';**

Για κάθε ερευνητή υπάρχει ένα μοναδικό αναγνωριστικό (**id**) το οποίο είναι ένας αύξον αριθμός, ο οποίος παράγεται αυτόματα από τη βάση. Επίσης καταχωρίζονται οι απαιτούμενες πληροφορίες οι οποίες είναι το όνομα (**name**) (*μέχρι 100 χαρακτήρες*), το επίθετο (**surname**) (*μέχρι 100 χαρακτήρες*), το φύλο (**gender**) και η ημερομηνία γέννησης (**birth\_date**).  
Ο έλεγχος εγκυρότητας του πεδίου **gender** γίνεται με τη βοήθεια του *CHECK CONSTRAINT* **check\_gender**, ώστε να βεβαιωθούμε ότι παίρνει μόνο τις τιμές *male* ή *female*.

Κάθε ερευνητής εργάζεται σε έναν (και μόνο) οργανισμό από κάποια ημερομηνία. Η υπαλληλική σχέση μεταξύ κάθε ερευνητή/οργανισμού αποθηκεύεται στο table **works\_for**, στο οποίο υπάρχουν εγγραφές για κάθε ερευνητή (**researcherId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***researcher***) μαζί με το όνομα του οργανισμού στον οποίο εργάζεται (**organizationName** - *FOREIGN KEY* στο πεδίο **name** του πίνακα ***organization***) και την ημερομηνία έναρξης της εργασίας (**employment\_date**). Εφόσον κάθε ερευνητής μπορεί να εργάζεται σε έναν μόνο οργανισμό, ορίζουμε ως κύριο κλειδί το αναγνωριστικό του ερευνητή.

**CREATE TABLE `works\_for` (**

**`researcherId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του εργαζόμενου ερευνητή',**

**`organizationName` varchar(100) NOT NULL COMMENT 'Όνομα του οργανισμού στον οποίο εργάζεται',**

**`employment\_date` date NOT NULL COMMENT 'Ημερομηνία έναρξης της εργασίας',**

**PRIMARY KEY (`researcherId`),**

**KEY `employing\_organization` (`organizationName`),**

**CONSTRAINT `employed\_researcher` FOREIGN KEY (`researcherId`) REFERENCES `researcher` (`id`),**

**CONSTRAINT `employing\_organization` FOREIGN KEY (`organizationName`) REFERENCES `organization` (`name`)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Υπαλληλική σχέση ερευνητή/οργανισμού';**

1. **Επιστημονικά πεδία που περιγράφουν θεματικές περιοχές των έργων (research\_field)**

**CREATE TABLE `research\_field` (**

**`name` varchar(100) NOT NULL COMMENT 'Όνομα (μοναδικό) του επιστημονικού πεδίου',**

**PRIMARY KEY (`name`)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Επιστημονικά πεδία';**

Για κάθε επιστημονικό πεδίο (τομέα έρευνας) υπάρχει το όνομά του (**name**) (*μέχρι 100 χαρακτήρες*), το οποίο θεωρούμε ότι είναι μοναδικό.

1. **Έργα/επιχορηγήσεις (project)**

**CREATE TABLE `project` (**

**`id` int NOT NULL AUTO\_INCREMENT COMMENT 'Μοναδικό αναγνωριστικό του έργου',**

**`title` varchar(100) NOT NULL COMMENT 'Τίτλος του έργου',**

**`description` varchar(255) NOT NULL COMMENT 'Περίληψη',**

**`budget` decimal(9, 2) NOT NULL COMMENT 'Ποσό χρηματοδότησης',**

**`start\_date` date NOT NULL COMMENT 'Έναρξη',**

**`end\_date` date DEFAULT NULL COMMENT 'Λήξη (αν είναι κενό, το έργο είναι ακόμα ενεργό)',**

**`duration` decimal(3, 2) GENERATED ALWAYS AS (**

**(**

**(to\_days(`end\_date`) - to\_days(`start\_date`)) / 365**

**)**

**) STORED COMMENT 'Διάρκεια του έργου (υπολογιζόμενο πεδίο)',**

**`programId` int NOT NULL COMMENT 'Αναγνωριστικό του προγράμματος που χρηματοδοτεί το έργο',**

**`researchManagerId` int NOT NULL COMMENT 'Επιστημονικός υπεύθυνος του έργου',**

**`executiveId` int NOT NULL COMMENT 'Στέλεχος του ΕΛ.ΙΔ.Ε.Κ που διαχειρίζεται το έργο',**

**PRIMARY KEY (`id`),**

**UNIQUE KEY `title\_UNIQUE` (`title`),**

**KEY `has\_research\_manager` (`researchManagerId`),**

**KEY `has\_executive\_manager` (`executiveId`),**

**KEY `has\_sponsoring\_program` (`programId`),**

**CONSTRAINT `has\_research\_manager` FOREIGN KEY (`researchManagerId`) REFERENCES `researcher` (`id`) ON UPDATE CASCADE,**

**CONSTRAINT `has\_executive\_manager ` FOREIGN KEY (`executiveId`) REFERENCES `executive` (`id`) ON UPDATE CASCADE,**

**CONSTRAINT `sponsors` FOREIGN KEY (`programId`) REFERENCES `program` (`id`) ON UPDATE CASCADE,**

**CONSTRAINT `limit\_end\_date` CHECK (**

**(**

**(**

**(to\_days(`end\_date`) - to\_days(`start\_date`)) / 365**

**) between 1**

**and 4**

**)**

**)**

**) ENGINE = InnoDB AUTO\_INCREMENT = 121 DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Έργα/Επιχορηγήσεις';**

Για κάθε έργο/επιχορήγηση υπάρχει ένα μοναδικό αναγνωριστικό (**id**) το οποίο είναι ένας αύξον αριθμός, ο οποίος παράγεται αυτόματα από τη βάση. Επίσης υπάρχουν τα σχετικά δεδομένα του, δηλαδή ο τίτλος (**title**) (*μέχρι 100 χαρακτήρες*), η περίληψη (**description**) (*μέχρι 255 χαρακτήρες*), το ποσό χρηματοδότησης/επιχορήγησης (**budget**), η ημερομηνία έναρξης (**start\_date**) και λήξης (**end\_date**) και η διάρκεια του έργου σε χρόνια (**duration**).  
- Η εγκυρότητα της ημερομηνίας λήξης ελέγχεται με τη βοήθεια του *CHECK CONSTRAINT* **limit\_end\_date**, ώστε να βεβαιωθούμε ότι η ελάχιστη διάρκεια από την ημερομηνία έναρξης είναι **1 έτος** και η μέγιστη διάρκεια τα **4 έτη**. Η ημερομηνία λήξης μπορεί να είναι κενή (**NULL**), σε περίπτωση που το έργο είναι ακόμα ενεργό.  
- Η διάρκεια του έργου σε έτη είναι υπολογιζόμενο πεδίο (**GENERATED**), το οποίο υπολογίζεται ως η διαφορά των πεδίων **start\_date** και **end\_date** σε έτη. Εάν το έργο είναι ακόμη ενεργό, τότε η διάρκεια θα είναι κενή (**NULL**).

Το κάθε έργο έχει έναν ερευνητή που είναι ο επιστημονικός υπεύθυνος του έργου (**researchManagerId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***researcher*** - **CONSTRAINT** **has\_research\_manager**), ένα πρόγραμμα το οποίο έχει χορηγήσει τη χρηματοδότηση (**programId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***program*** - **CONSTRAINT** **has\_sponsoring\_program**) και ένα στέλεχος το οποίο το διαχειρίζεται (**executiveId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***executive*** - **CONSTRAINT** **has\_executive\_manager**).

Επίσης, κάθε έργο έχει έναν οργανισμό που το διαχειρίζεται. Η σχέση μεταξύ κάθε έργου/οργανισμού αποθηκεύεται στο table **managed\_by**, στο οποίο υπάρχουν εγγραφές για κάθε έργο (**projectId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***project***) μαζί με το όνομα του οργανισμού το οποίο το διαχειρίζεται (**organizationName** - *FOREIGN KEY* στο πεδίο **name** του πίνακα ***organization***). Εφόσον κάθε έργο διαχειρίζεται από έναν μόνο οργανισμό, ορίζουμε πεδίο **projectId** ως μοναδικό (**UNIQUE**).

**CREATE TABLE `managed\_by` (**

**`projectId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του έργου το οποίο συμμετέχει στη σχέση',**

**`organizationName` varchar(100) NOT NULL COMMENT 'Όνομα του οργανισμού που διαχειρίζεται το έργο',**

**UNIQUE KEY `projectId\_UNIQUE` (`projectId`),**

**KEY `managing\_organization` (`organizationName`),**

**CONSTRAINT `managed\_project` FOREIGN KEY (`projectId`) REFERENCES `project` (`id`),**

**CONSTRAINT `managing\_organization` FOREIGN KEY (`organizationName`) REFERENCES `organization` (`name`)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Σχέση διαχείρισης μεταξύ έργου και οργανισμού';**

Κάθε έργο έχει αφορά ένα ή περισσότερα επιστημονικά πεδία. Η σχέση μεταξύ κάθε έργου/σχετιζόμενων επιστημονικών πεδίων αποθηκεύεται στο table **related\_to**, στο οποίο υπάρχουν εγγραφές για κάθε έργο (**projectId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***project***) μαζί με το όνομα του σχετιζόμενου επιστημονικού πεδίου (**research\_fieldName** - *FOREIGN KEY* στο πεδίο **name** του πίνακα ***research\_field***). Ορίζουμε ως κύριο κλειδί το **projectId** μαζί με το **research\_fieldName** για να αποφύγουμε διπλές εγγραφές για κάποιο έργο με το ίδιο επιστημονικό πεδίο.

**CREATE TABLE `related\_to` (**

**`projectId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του έργου',**

**`research\_fieldName` varchar(100) NOT NULL COMMENT 'Όνομα του επιστημονικού πεδίου με το οποίο συνδέεται',**

**PRIMARY KEY (`projectId`, `research\_fieldName`),**

**KEY `field\_to\_project` (`research\_fieldName`),**

**CONSTRAINT `field\_to\_project` FOREIGN KEY (`research\_fieldName`) REFERENCES `research\_field` (`name`),**

**CONSTRAINT `project\_fields` FOREIGN KEY (`projectId`) REFERENCES `project` (`id`)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Σύνδεση έργου με επιστημονικά πεδία';**

Ένα έργο ενδέχεται να έχει παραδοτέα, τα οποία παραδίδονται σε συγκεκριμένη ημερομηνία. Η σχέση μεταξύ κάθε έργου και των παραδοτέων του (εφόσον υπάρχουν) αποθηκεύεται στο table **deliverable**, στο οποίο υπάρχουν εγγραφές για έργα (**projectId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***project***) μαζί με το τις πληροφορίες για κάθε παραδοτέο του. Οι πληροφορίες για κάθε παραδοτέο είναι ο τίτλος του (**title**) (*μέχρι 100 χαρακτήρες*), η περίληψη (**description**) (*μέχρι 255 χαρακτήρες*) και η ημερομηνία παράδοσης (**delivery\_date**). Ορίζουμε ως κύριο κλειδί το **projectId** μαζί με το **title**, θεωρώντας ότι κάθε έργο δε μπορεί να έχει πάνω από ένα παραδοτέο με τον ίδιο τίτλο.

**CREATE TABLE `deliverable` (**

**`projectId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του έργου το οποίο αφορά το παραδοτέο',**

**`title` varchar(100) NOT NULL COMMENT 'Τίτλος παραδοτέου',**

**`description` varchar(255) NOT NULL COMMENT 'Περιγραφή παραδοτέου',**

**`delivery\_date` date DEFAULT NULL COMMENT 'Ημερομηνία παράδοσης',**

**PRIMARY KEY (`title`, `projectId`),**

**KEY `has\_deliverables` (`projectId`),**

**CONSTRAINT `has\_deliverables` FOREIGN KEY (`projectId`) REFERENCES `project` (`id`)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Παραδοτέα έργων';**

Για κάθε έργο υπάρχουν ερευνητές που εργάζονται πάνω σε αυτό. Η σχέση μεταξύ κάθε έργου και των ερευνητών που εργάζονται σε αυτό αποθηκεύεται στο table **project\_researchers**, στο οποίο υπάρχουν εγγραφές για κάθε έργο (**projectId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***project***) και τους ερευνητές που εργάζονται σε αυτό (**researcherId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***researcher***). Ορίζουμε ως κύριο κλειδί το **projectId** μαζί με το **researcherId**, για να αποφύγουμε διπλές εγγραφές για κάποιο έργο με τον ίδιο ερευνητή.

**CREATE TABLE `project\_researchers` (**

**`projectId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του έργου',**

**`researcherId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του ερευνητή που εργάζεται στο έργο',**

**PRIMARY KEY (`projectId`, `researcherId`),**

**KEY `works\_on` (`researcherId`)**

**/\*!80000 INVISIBLE \*/**

**,**

**CONSTRAINT `has\_researchers` FOREIGN KEY (`projectId`) REFERENCES `project` (`id`),**

**CONSTRAINT `works\_on` FOREIGN KEY (`researcherId`) REFERENCES `researcher` (`id`)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Ερευνητές σε κάθε έργο';**

Τέλος το κάθε έργο, προκειμένου να χρηματοδοτηθεί, έχει αξιολογηθεί από έναν ερευνητή ο οποίος δεν ανήκει στο δυναμικό του οργανισμού που συμμετέχει στην πρόταση. Η σχέση μεταξύ κάθε έργου και της αξιολόγησής του αποθηκεύεται στο table **project\_rating**, στο οποίο υπάρχουν εγγραφές για κάθε έργο (**projectId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***project***), τον ερευνητή που έκανε την αξιολόγηση (**researcherId** - *FOREIGN KEY* στο πεδίο **id** του πίνακα ***researcher***) και τις πληροφορίες για την αξιολόγηση. Οι πληροφορίες για κάθε αξιολόγηση είναι ο βαθμός της (**rating**) και η ημερομηνία αξιολόγησης (**rating\_date**). Εφόσον κάθε έργο μπορεί να έχει μόνο μία αξιολόγηση, ορίζουμε πεδίο **projectId** ως μοναδικό (**UNIQUE**).   
- Η εγκυρότητα του βαθμού αξιολόγησης ελέγχεται με τη βοήθεια του *CHECK CONSTRAINT* **check\_rating**, ώστε να βεβαιωθούμε ότι παίρνει τιμές μεταξύ 1 και 10.

**CREATE TABLE `project\_rating` (**

**`projectId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του έργου που αξιολογήθηκε (κάθε έργο μπορεί να αξιολογηθεί μόνο μία φορά)',**

**`researcherId` int NOT NULL COMMENT 'Μοναδικό αναγνωριστικό του ερευνητή που αξιολόγησε το έργο',**

**`rating` int NOT NULL COMMENT 'Αξιολόγηση (1-10)',**

**`rating\_date` date NOT NULL COMMENT 'Ημερομηνία αξιολόγησης',**

**UNIQUE KEY `projectId\_UNIQUE` (`projectId`),**

**KEY `rated\_project` (`projectId`),**

**KEY `rating\_researcher` (`researcherId`),**

**CONSTRAINT `project\_rating\_ibfk\_1` FOREIGN KEY (`projectId`) REFERENCES `project` (`id`),**

**CONSTRAINT `rating\_researcher` FOREIGN KEY (`researcherId`) REFERENCES `researcher` (`id`) ON UPDATE CASCADE,**

**CONSTRAINT `check\_rating` CHECK (**

**(**

**`rating` between 1**

**and 10**

**)**

**)**

**) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci COMMENT = 'Αξιολογήσεις έργων από ερευνητές';**

Προκειμένου να γίνει ο έλεγχος ότι ο ερευνητής που έκανε την αξιολόγηση δεν ανήκει στον οργανισμό που συμμετέχει στην πρόταση, υπάρχει ένα TRIGGER που ενεργοποιείται πριν γίνει INSERT στο table ως εξής:

**CREATE DEFINER=`root`@`localhost` TRIGGER `check\_researcher\_not\_in\_organization` BEFORE INSERT ON `project\_rating` FOR EACH ROW BEGIN**

**IF rating\_researcher\_in\_organization(NEW.projectId, NEW.researcherId) = 1 THEN**

**SET NEW.researcherId = NULL;**

**END IF;**

**END**

Το TRIGGER αυτό χρησιμοποιεί την παρακάτω συνάρτηση (**\_researcher\_in\_organization**) για να ελέγξει στον πίνακα works\_for εάν ο ερευνητής που έκανε την αξιολόγηση ανήκει στον οργανισμό που διαχειρίζεται το έργο ως εξής:

**CREATE DEFINER=`root`@`localhost` FUNCTION `rating\_researcher\_in\_organization`(newProjectId INT, newResearcherId INT) RETURNS int**

**READS SQL DATA**

**BEGIN**

**RETURN EXISTS (**

**SELECT m.organizationName FROM managed\_by AS m**

**JOIN works\_for AS w**

**ON newResearcherId = w.researcherId AND m.organizationName = w.organizationName**

**WHERE m.projectId = newProjectId**

**);**

**END**

**DDL και DML scripts**

Τα scripts για τη δημιουργία του σχήματος της βάσης (**D**ata **D**efinition **L**anguage - DDL script) και την εισαγωγή των δεδομένων στη βάση (**D**ata **M**anipulation **L**anguage - DML script) βρίσκονται στο [Github repository](https://github.com/kdivriotis/db_project), και συγκεκριμένα στο φάκελο **database**, με ονόματα Schema.DDL.sql και Data.DML.sql αντίστοιχα.

Σημείωση: Τα dummy δεδομένα για την εργασία δημιουργήθηκαν με τη βοήθεια του [**Fill Database**](http://filldb.info/)**.**