

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ

Ακαδημαϊκό έτος 2021-2022

ΟΜΑΔΑ 60

Νικόλαος Καραφύλλης | Α.Μ.:03119890

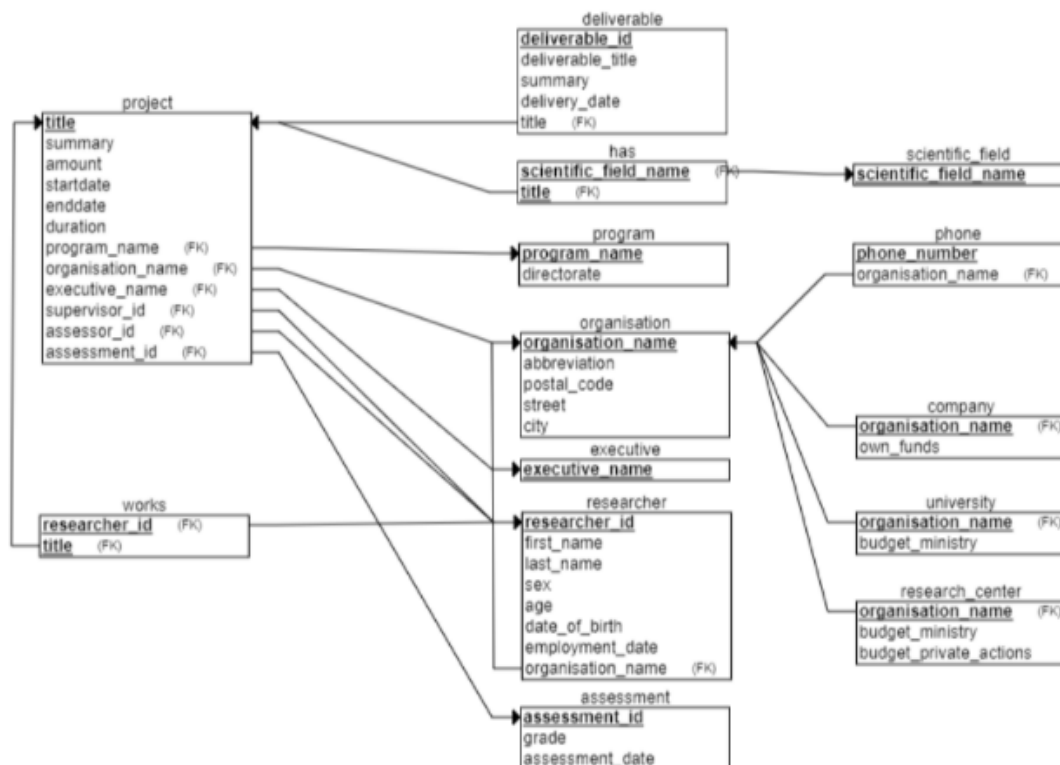
Αναστάσιος Σαφαρής: | Α.Μ.:03119896

Ιγνάτιος Σιμινής: | Α.Μ.:03119821

Σύστημα αποθήκευσης, διαχείρισης και ανάλυσης πληροφοριών
ΕΛ.Ι.Δ.Ε.Κ

2.1

Παρατίθεται το σχεσιακό διάγραμμα που χρησιμοποιήθηκε και αντιστοιχεί στο ER
διάγραμμα που μας δόθηκε.



INDEXES

Ακόμα, έχουμε ορίσει 5 κατάλληλα ευρετήρια (indexes) τα οποία έχουν οριστεί με σκοπό την ελαχιστοποίηση του χρόνου που απαιτείται για την πραγματοποίηση των queries. Η

συγκεκριμένη επιλογή έγινε, καθώς τα συγκεκριμένα attributes καλούνται με μεγάλη συχνότητα στα ερωτήματα.

```
1 CREATE INDEX idx_Startdate ON project(startdate);
2 CREATE INDEX idx_Enddate ON project(enddate);
3 CREATE INDEX idx_resfirstname ON researcher(first_name);
4 CREATE INDEX idx_reslastname ON researcher(last_name);
5 CREATE INDEX idx_dateofbirth ON researcher(date_of_birth);
```

2.2

Query που χρησιμοποιήθηκε για την δημιουργία της βάσης δεδομένων

```
4 DROP DATABASE IF EXISTS `HFRIManagement`;
5 CREATE DATABASE `HFRIManagement`;
6 ALTER DATABASE HFRIManagement
7 CHARACTER SET utf8
8 COLLATE utf8_bin;
```

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα project

```
100 DROP TABLE IF EXISTS project;
101 CREATE TABLE project
102 (
103     title varchar (50) PRIMARY KEY not null,
104     summary varchar (1000),
105     amount int CHECK (amount>=100000 AND amount<=1000000),
106     startdate date,
107     enddate date,
108     program_name varchar(50),
109     organisation_name varchar (50),
110     executive_name varchar(50),
111     supervisor_id int,
112     assessor_id int,
113     assessment_id int,
114     CONSTRAINT duration CHECK ((timestampdiff(year,startdate, enddate))>=1 and (timestampdiff(year,enddate, startdate))<=4),
115     FOREIGN KEY (program_name) REFERENCES program(program_name) ON DELETE SET NULL ON UPDATE CASCADE,
116     FOREIGN KEY (organisation_name) REFERENCES organisation(organisation_name) ON DELETE SET NULL ON UPDATE CASCADE,
117     FOREIGN KEY (executive_name) REFERENCES executive(executive_name) ON DELETE SET NULL ON UPDATE CASCADE,
118     FOREIGN KEY (supervisor_id) REFERENCES researcher(researcher_id) ON DELETE SET NULL ON UPDATE CASCADE,
119     FOREIGN KEY (assessor_id) REFERENCES researcher(researcher_id) ON DELETE SET NULL ON UPDATE CASCADE,
120     FOREIGN KEY (assessment_id) REFERENCES assessment(assessment_id) ON DELETE SET NULL ON UPDATE CASCADE
121 );
```

Για primary key επιλέχθηκε το attribute title, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε έργου. Το program_name αποτελεί foreign key για τον πίνακα program και αντιστοιχίζει τη σχέση πολλά (project) σε ένα (program). Το organisation_name αποτελεί foreign key για τον πίνακα organisation και αντιστοιχίζει τους δύο πίνακες με σχέση πολλά (project) σε ένα (organisation). Το supervisor_id και το assessor_id αποτελούν foreign keys για τον πίνακα researcher και αντιστοιχίζουν τους δύο πίνακες με σχέση πολλά (project) σε ένα (researcher). Το executive_name αποτελεί foreign key για τον πίνακα executive και αντιστοιχίζει τους δύο πίνακες με σχέση πολλά (project) σε ένα (executive). Το assessment_id αποτελεί foreign key για τον πίνακα assessment και αντιστοιχίζει τους δύο πίνακες με σχέση πολλά (project) σε ένα (assessment). Τέλος, έχουμε βάλει ένα CHECK constraint που να ελέγχει ότι η διάρκεια ενός project είναι από 1 έως 4 χρόνια.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα deliverable

```
DROP TABLE IF EXISTS deliverable;
CREATE TABLE deliverable
(
    deliverable_id int auto_increment PRIMARY KEY NOT NULL,
    deliverable_title varchar (50) not null,
    summary varchar (500),
    delivery_date date,
    title varchar(50) NOT NULL,
    FOREIGN KEY (title) REFERENCES project(title) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Για primary key επιλέχθηκε το attribute deliverable_id, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε παραδοτέου. Το title αποτελεί foreign key για τον πίνακα project και αντιστοιχίζει τους δύο πίνακες με σχέση πολλά (deliverable) σε ένα (project).

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα program

```
DROP TABLE IF EXISTS program;
CREATE TABLE program
(
    program_name varchar (50) PRIMARY KEY not null,
    directorate varchar (100)
);
```

Για primary key επιλέχθηκε το attribute program_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό του κάθε προγράμματος.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα organisation

```
DROP TABLE IF EXISTS organisation;
CREATE TABLE organisation
(
    organisation_name varchar (50) PRIMARY KEY not null,
    abbreviation varchar (10),
    postal_code varchar (50),
    street varchar (50),
    city varchar (50)
);
```

Για primary key επιλέχθηκε το attribute organisation_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό του κάθε οργανισμού.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα executive

```

DROP TABLE IF EXISTS executive;
CREATE TABLE executive
(
    executive_name varchar (50) PRIMARY KEY not null
);

```

Για primary key επιλέχθηκε το attribute executive_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε στελέχους. Άλλωστε, είναι το μοναδικό attribute της συγκεκριμένης οντότητας.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα researcher

```

DROP TABLE IF EXISTS researcher;
CREATE TABLE researcher
(
    researcher_id int auto_increment PRIMARY KEY NOT NULL,
    first_name varchar (50) not null,
    last_name varchar (50) not null,
    sex varchar (12) not null,
    date_of_birth date,
    /*age int GENERATED ALWAYS AS (TIMESTAMPDIFF(YEAR, date_of_birth, current_date()))*/ /*mono sta queries ta derived*/
    employment_date date,
    organisation_name varchar (50) ,
    FOREIGN KEY (organisation_name) REFERENCES organisation(organisation_name) ON DELETE SET NULL ON UPDATE CASCADE
);

```

Για primary key επιλέχθηκε το attribute researcher_id, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε ερευνητή. Το organisation_name αποτελεί foreign key για τον πίνακα organisation και αντιστοιχίζει τους δύο πίνακες με σχέση πολλά (researcher) σε ένα (organisation).

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα assessment

```

DROP TABLE IF EXISTS assessment;
CREATE TABLE assessment
(
    assessment_id int auto_increment PRIMARY KEY NOT NULL,
    grade int not null,
    assessment_date date
);

```

Για primary key επιλέχθηκε το attribute assessment_id, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε αξιολόγησης.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα scientific_field

```

DROP TABLE IF EXISTS scientific_field;
CREATE TABLE scientific_field
(
    scientific_field_name varchar (50) PRIMARY KEY not null
);

```

Για primary key επιλέχθηκε το attribute scientific_field_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε επιστημονικού πεδίου. Άλλωστε, είναι το μοναδικό attribute της συγκεκριμένης οντότητας.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα phone

```
DROP TABLE IF EXISTS phone;
CREATE TABLE phone
(
    phone_number varchar (50) PRIMARY KEY NOT NULL,
    organisation_name varchar (50) NOT NULL,
    FOREIGN KEY (organisation_name) REFERENCES organisation(organisation_name) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Για primary key επιλέχθηκε το attribute phone_number, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε τηλεφώνου. Το organisation_name αποτελεί foreign key στον πίνακα organisation και αντιστοιχίζει τους δύο πίνακες με σχέση πολλά (phone) σε ένα (organisation).

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα company

```
DROP TABLE IF EXISTS company;
CREATE TABLE company
(
    organisation_name varchar (50) PRIMARY KEY NOT NULL,
    own_funds int not null,
    FOREIGN KEY (organisation_name) REFERENCES organisation(organisation_name) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Για primary key επιλέχθηκε το attribute organisation_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε εταιρείας.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα university

```
DROP TABLE IF EXISTS university;
CREATE TABLE university
(
    organisation_name varchar (50) PRIMARY KEY NOT NULL,
    budget_ministry int not null,
    FOREIGN KEY (organisation_name) REFERENCES organisation(organisation_name) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Για primary key επιλέχθηκε το attribute organisation_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε πανεπιστημίου.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα research_center

```
DROP TABLE IF EXISTS research_center;
CREATE TABLE research_center
(
    organisation_name varchar (50) PRIMARY KEY NOT NULL,
    budget_ministry int not null,
    budget_private_actions int not null,
    FOREIGN KEY (organisation_name) REFERENCES organisation(organisation_name) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Για primary key επιλέχθηκε το attribute organisation_name, διότι αυτό είναι μοναδικό και χαρακτηριστικό κάθε ερευνητικού κέντρου.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα works

```

DROP TABLE IF EXISTS works;
CREATE TABLE works
(
    /*working_researcher_id int auto_increment PRIMARY KEY NOT NULL,*/
    #works_id int auto_increment PRIMARY KEY NOT NULL,
    researcher_id int NOT NULL,
    title varchar(50) NOT NULL,
    PRIMARY KEY (researcher_id, title),
    FOREIGN KEY (researcher_id) REFERENCES researcher(researcher_id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (title) REFERENCES project(title) ON DELETE CASCADE ON UPDATE CASCADE
);

```

Ο πίνακας works είναι μία σχέση (πολλά προς πολλά) που ενώνει τις οντότητες project και researcher. Για αυτό, εκτός από το primary key (researcher_id, title) που είναι composite key και χαρακτηρίζει μοναδικά κάθε εγγραφή του, περιέχει και το researcher_id ως foreign key για το researcher και το title ως foreign key για το project.

Query που χρησιμοποιήθηκε για την δημιουργία του πίνακα has

```

DROP TABLE IF EXISTS has;
CREATE TABLE has
(
    title varchar(50) NOT NULL,
    scientific_field_name varchar(50) NOT NULL,
    PRIMARY KEY (title,scientific_field_name),
    FOREIGN KEY (title) REFERENCES project(title) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (scientific_field_name) REFERENCES scientific_field(scientific_field_name)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

Ο πίνακας has είναι μία σχέση (πολλά προς πολλά) που ενώνει τις οντότητες project και scientific_field. Για αυτό, εκτός από το primary key (scientific_field_name, title) που είναι composite key και χαρακτηρίζει μοναδικά κάθε εγγραφή του, περιέχει και το scientific_field_name ως foreign key για το researcher και το title ως foreign key για το project.

TRIGGERS

Προκειμένου να καλύψουμε τον περιορισμό της εκφώνησης ότι ο αξιολογητής κάθε έργου δεν μπορεί να ανήκει στον οργανισμό που διαχειρίζεται το συγκεκριμένο έργο, έχουμε δημιουργήσει δύο triggers (ένα before insert και ένα before update) τα οποία ελέγχουν τη συγκεκριμένη συνθήκη και εμφανίζουν κατάλληλο μήνυμα σφάλματος.

```

1  DELIMITER $$
2  DROP TRIGGER IF EXISTS before_assessor_insert$$
3  CREATE TRIGGER before_assessor_insert
4  before insert ON project FOR EACH ROW
5  BEGIN
6  IF new.assessor_id in (select a.researcher_id from researcher a where new.organisation_name = a.organisation_name)
7  THEN SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Assessor cannot assess a project belonging in his organisation';
8  END IF;
9  END $$
10
11 DELIMITER $$
12 DROP TRIGGER IF EXISTS before_assessor_update$$
13 CREATE TRIGGER before_assessor_update
14 before update ON project FOR EACH ROW
15 BEGIN
16 IF new.assessor_id in (select a.researcher_id from researcher a where new.organisation_name = a.organisation_name)
17 THEN SIGNAL SQLSTATE '50004' SET MESSAGE_TEXT = 'New assessor cannot assess a project belonging in his organisation';
18 END IF;
19 END $$

```

USERS

Έχουμε δημιουργήσει 3 users με διαβαθμισμένα δικαιώματα στη βάση.

```
1 CREATE USER superuser@localhost;
2 GRANT ALL PRIVILEGES ON hfrimanagement.* TO superuser;
3
4 CREATE USER data_entry@localhost;
5 GRANT SELECT, INSERT, DELETE, UPDATE ON hfrimanagement.* TO data_entry;
6
7 CREATE USER viewer@localhost;
8 GRANT SELECT ON hfrimanagement.* TO viewer;
```

PYTHON ΚΩΔΙΚΑΣ-ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

Ο κώδικας (insert_data.py) που χρησιμοποιήθηκε για την εισαγωγή των dummy δεδομένων στην βάση από τα αρχεία .csv είναι ο ακόλουθος:

```
1 import mysql.connector
2 import pandas as pd
3 import numpy as np
4
5 programs = pd.read_csv("./Data/programs.csv")
6 scientific_fields= pd.read_csv("./Data/scientific_fields.csv")
7 executives = pd.read_csv("./Data/executives.csv")
8 assessments = pd.read_csv("./Data/assessments.csv")
9 organisations = pd.read_csv("./Data/organisations.csv")
10 organisation_types = pd.read_csv("./Data/organisation_type.csv")
11 phone_numbers = pd.read_csv("./Data/phone_numbers.csv")
12 researchers = pd.read_csv("./Data/researchers.csv")
13 projects = pd.read_csv("./Data/projects.csv")
14 has = pd.read_csv("./Data/has.csv")
15 deliverables = pd.read_csv("./Data/deliverable.csv")
16 works = pd.read_csv("./Data/works.csv")
17
18 mydb = mysql.connector.connect(
19     host = "localhost",
20     user = "root",
21     passwd = '',
22     database = "HFRIManagement"
23 )
```

```

24
25 mycursor = mydb.cursor()
26
27 for i in range(len(programs)):
28     # program_name,directorate
29     program_name = programs['program_name'][i].replace("'", "").replace('"', "")
30     directorate= programs['directorate'][i].replace("'", "").replace('"', "")
31
32     sqlFormula = """INSERT INTO program (program_name,directorate)
33     | | | | | VALUES ('{}','{}')""".format(program_name,directorate)
34
35     mycursor.execute(sqlFormula)
36     mydb.commit() #Save Data
37
38 for i in range(len(scientific_fields)):
39     #scientific_field_name
40     scientific_field_name = scientific_fields['scientific_field_name'][i].replace("'", "").replace('"', "")
41
42     sqlFormula = """INSERT INTO scientific_field (scientific_field_name)
43     | | | | | VALUES ('{}')""".format(scientific_field_name)
44
45     mycursor.execute(sqlFormula)
46     mydb.commit()
47
48 for i in range(len(executives)):
49     executive_name = executives['executive_name'][i].replace("'", "").replace('"', "")
50
51     sqlFormula = """INSERT INTO executive (executive_name)
52     | | | | | VALUES ('{}')""".format(executive_name)
53
54     mycursor.execute(sqlFormula)
55     mydb.commit()
56
57 for i in range(len(assessments)):
58     grade = assessments['grade'][i]
59     assessment_date = assessments['assessment_date'][i].replace("'", "").replace('"', "")
60
61     sqlFormula = """INSERT INTO assessment (grade,assessment_date)
62     | | | | | VALUES ('{}','{}')""".format(grade,assessment_date)
63
64     mycursor.execute(sqlFormula)
65     mydb.commit()
66
67 for i in range(len(organisations)):
68     organisation_name = organisations['organisation_name'][i].replace("'", "").replace('"', "")
69     abbreviation = organisations['abbreviation'][i].replace("'", "").replace('"', "")
70     postal_code = organisations['postal_code'][i]
71     street= organisations['street'][i].replace("'", "").replace('"', "")
72     city= organisations['city'][i].replace("'", "").replace('"', "")
73
74     sqlFormula = """INSERT INTO organisation (organisation_name,abbreviation,postal_code,street,city)
75     | | | | | VALUES ('{}','{}','{}','{}','{}')""".format(organisation_name,abbreviation,postal_code,street,city)
76
77     mycursor.execute(sqlFormula)
78     mydb.commit()
79
80 for i in range(len(organisation_types)):
81     own_funds = organisation_types['own_funds'][i]
82     budget_ministry= organisation_types['budget_ministry'][i]
83     budget_private_actions= organisation_types['budget_private_actions'][i]
84     budget_ministry2= organisation_types['budget_ministry2'][i]
85     organisation_name=organisation_types['organisation_name'][i]
86
87     sqlFormula1 = """INSERT INTO company (own_funds,organisation_name)
88     | | | | | VALUES ('{}','{}')""".format(own_funds,organisation_name)
89     sqlFormula2 = """INSERT INTO university (budget_ministry,organisation_name)
90     | | | | | VALUES ('{}','{}')""".format(budget_ministry,organisation_name)
91     sqlFormula3 = """INSERT INTO research_center (budget_private_actions,budget_ministry,organisation_name)

```



```

92         VALUES ('{}','{}','{}')'"".format(budget_private_actions,budget_ministry2,organisation_name)
93
94     mycursor.execute(sqlFormula1)
95     mycursor.execute(sqlFormula2)
96     mycursor.execute(sqlFormula3)
97     mydb.commit()
98
99     for i in range(len(phone_numbers)):
100         phone_number = phone_numbers['phone_number'][i].replace("'", "").replace(' ','')
101         organisation_name = phone_numbers['organisation_name'][i]
102
103         sqlFormula = """INSERT INTO phone (phone_number,organisation_name)
104         VALUES ('{}','{}')'"".format(phone_number,organisation_name)
105
106         mycursor.execute(sqlFormula)
107         mydb.commit()
108
109     for i in range(len(researchers)):
110         first_name = researchers['first_name'][i].replace("'", "").replace(' ','')
111         last_name = researchers['last_name'][i].replace("'", "").replace(' ','')
112         sex = researchers['sex'][i].replace("'", "").replace(' ','')
113         date_of_birth = researchers['date_of_birth'][i]
114         employment_date = researchers['employment_date'][i]

```

```

115         organisation_name = researchers['organisation_name'][i]
116
117         sqlFormula = """INSERT INTO researcher (first_name,last_name,sex,date_of_birth,employment_date,organisation_name)
118         VALUES ('{}','{}','{}','{}','{}','{}')'"".format(first_name,last_name,sex,date_of_birth,
119         employment_date,organisation_name)
120
121         mycursor.execute(sqlFormula)
122         mydb.commit()
123
124     for i in range(len(projects)):
125         title = projects['title'][i].replace("'", "").replace(' ','')
126         summary = projects['summary'][i].replace("'", "").replace(' ','')
127         amount = projects['amount'][i]
128         startdate = projects['startdate'][i].replace("'", "").replace(' ','')
129         enddate = projects['enddate'][i].replace("'", "").replace(' ','')
130         #duration = timestampdiff(year,enddate, startdate)
131         program_name = projects['program_name'][i]
132         organisation_name = projects['organisation_name'][i]
133         executive_name = projects['executive_name'][i]
134         supervisor_id = projects['supervisor_id'][i]
135         assessor_id = projects['assessor_id'][i]
136         assessment_id = projects['assessment_id'][i]
137

```

```

137         sqlFormula = """INSERT INTO project (title,summary,amount,startdate,enddate,program_name,organisation_name,
138         executive_name,supervisor_id,assessor_id,assessment_id)
139         VALUES ('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')'"".format(title,summary,amount,startdate,enddate,
140         program_name,organisation_name,executive_name,supervisor_id,assessor_id,assessment_id)
141
142         mycursor.execute(sqlFormula)
143         mydb.commit()
144
145
146     for i in range(len(has)):
147         scientific_field_name = has['scientific_field_name'][i]
148         title = has['title'][i]
149
150         sqlFormula = """INSERT INTO has (title,scientific_field_name)
151         VALUES ('{}','{}')'"".format(title,scientific_field_name)
152
153         mycursor.execute(sqlFormula)
154         mydb.commit()
155
156     for i in range(len(deliverables)):
157         deliverable_title = deliverables['deliverable_title'][i].replace("'", "").replace(' ','')
158         summary= deliverables['summary'][i].replace("'", "").replace(' ','')
159         delivery_date = deliverables['delivery_date'][i].replace("'", "").replace(' ','')

```

```

159 delivery_date = deliverables['delivery_date'][i].replace("'", "").replace('"', "")
160 title = deliverables['title'][i]
161
162 sqlFormula = """INSERT INTO deliverable (deliverable_title,summary,delivery_date,title)
163 | | | | VALUES ('{}','{}','{}','{}')""".format(deliverable_title,summary,delivery_date,title)
164
165 mycursor.execute(sqlFormula)
166 mydb.commit()
167
168 for i in range(len(works)):
169     researcher_id = works['researcher_id'][i]
170     title= works['title'][i]
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

<https://github.com/nickarafyllis/Elidek-database>

