

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ

Ονοματεπώνυμο / ΑΜ :

Δημήτριος Λαυίδ Γεροκωνσταντής ΑΜ : 03119209

Αθανάσιος Τσουκλείδης-Καρυδάκης ΑΜ : 03119009

Φίλιππος Σεβαστάκης ΑΜ : 03119183

Εξάμηνο : 6^ο

Ομάδα 24

2021-22

➤ Παρατηρήσεις ως προς το σχεδιασμό του Relational Model και την δημιουργία indexes

Το παραπάνω διάγραμμα σχεδιάστηκε με βάση το ER διάγραμμα του 1^{ου} παραδοτέου μας, με τις παρακάτω διαφορές :

- Χρησιμοποιήθηκε η ISA σχέση του προτεινόμενου ER μεταξύ του organisation και των κατηγοριών του.
- Προστέθηκαν δύο επιπλέον Entity Sets, τα Administrates και Evaluation τα οποία συνδέονται μέσω των σχέσεων Administrates και Evaluates αντίστοιχα του ER διαγράμματός μας (one-to-many σχέσεις) με το project.
- Η σχέση Supervises από many-to-many έγινε many-to-one.

Στις one-to-many σχέσεις με total participation από την πλευρά του many, παραλείπουμε την συμπερίληψη στο διάγραμμα των αντίστοιχων σχέσεων και προσθέτουμε στο table της “many” πλευράς το primary key της άλλης και τα attributes της σχέσης. Η ίδια συμπεριφορά με το προηγούμενο ακολουθείται και στην περίπτωση του weak entity set.

Τα παραπάνω χρησιμοποιήθηκαν στις σχέσεις :

- WORKS_FOR
- MANAGES
- SUPERVISES
- FUNDS
- HAS

οι οποίες παραλήφθηκαν.

Επιπλέον, τα composite attributes (π.χ res_name) παραλείπονται και συμπεριλαμβάνονται στο διάγραμμα τα components τους, ενώ τα derived attributes (π.χ. age) παραλείπονται αφού υλοποιούνται μέσω πράξεων μεταξύ των ήδη υπάρχοντων attributes.

Τέλος, το multivalued attribute phone number του organisation γίνεται ξεχωριστό table με primary key το primary key του organization (org_name).

Στο αρχείο elidek schema.sql στο .zip αρχείο και στο git repo φαίνονται τα indexes που προστέθηκαν στο σχήμα της βάσης για την αποδοτικότερη υλοποίηση ερωτημάτων που χρησιμοποιούν τα εκάστοτε tables.

Επιλέχθηκε η χρήση indexes κυρίως στα primary keys και foreign keys των σχετικά μεγάλων (με πολλά attributes και tuples) σχέσεων.

Μέρος 2.2 : DDL/DML

➤ DDL

Στο αρχείο elidek schema.sql στο .zip αρχείο και στο git repo βρίσκεται το DDL τμήμα της υλοποίησης που δημιουργεί τη βάση.

Επιπλέον ορίζονται κατάλληλα foreign key constraints και triggers που απαγορεύουν την εκτέλεση πράξεων όπως :

- ορισμός ενός researcher ως evaluator του προγράμματος στο οποίο εργάζεται
- την εργασία ενός researcher σε project άλλου οργανισμού από αυτόν στον οποίο ανήκει
- τον ορισμό evaluator για project του οργανισμού στον οποίο εργάζεται (λόγοι αμεροληψίας)
- ...

Σημειώνεται ότι τα views που δημιουργήθηκαν για τα projects ενός researcher και για τα επιστημονικά πεδία με τα οποία απασχολείται ένας οργανισμός δεν είναι updateable αφού στο from clause περιλαμβάνουν περισσότερες της μίας σχέσεις καθιστώντας προβληματική την ενημέρωση των αρχικών πινάκων που χρησιμοποιεί το view.

➤ DML

Τα queries εισαγωγής δεδομένων στη βάση έγιναν χειροκίνητα με εντολές insert και φαίνονται στο αρχείο elidek insert.sql στο .zip αρχείο και στο git repo.

Τα δε ζητούμενα ερωτήματα υλοποιήθηκαν όπως παρακάτω :

- Query 3.1 :

```

1  select subq.prog_name, subq.proj_title as proj_title, subq.starting_date,
2  subq.duration, concat(ex.first_name, ' ', ex.last_name) as exec_name,
3  subq.res_name, subq.Researcher_Role
4  from (
5  (
6  select prog.prog_name, proj.title as proj_title, proj.starting_date,
7  TIMESTAMPDIFF(month, proj.starting_date, proj.ending_date) as duration, proj.executive_id,
8  concat(res.first_name, ' ', res.last_name) as res_name, 'Worker' as Researcher_Role
9  from program as prog
10 inner join project as proj on proj.program_id=prog.program_id
11 inner join works_on as wo on wo.project_id=proj.project_id
12 inner join researcher as res on res.res_id=wo.res_id
13 )
14 union
15 (
16 select prog.prog_name, proj.title as proj_title, proj.starting_date,
17 TIMESTAMPDIFF(month, proj.starting_date, proj.ending_date) as duration, proj.executive_id,
18 concat(res.first_name, ' ', res.last_name) as res_name, 'Administrator' as Researcher_Role
19 from program as prog
20 inner join project as proj on proj.program_id=prog.program_id
21 inner join administrates as adm on adm.project_id=proj.project_id
22 inner join researcher as res on res.res_id=adm.res_id ) as subq
23 inner join executive as ex on ex.executive_id=subq.executive_id
24 order by subq.prog_name, subq.proj_title;
```

- Query 3.2 (1st View) :

```

187 • CREATE VIEW Projects_Per_Researcher AS
188 ((select res.first_name, res.last_name, 'Administrator' as Researcher_Role, proj.title as proj_title
189 from
190 researcher as res inner join administrates as adm on res.res_id=adm.res_id
191 inner join project as proj on adm.project_id = proj.project_id)
192
193 union
194
195 (select res.first_name, res.last_name, 'Worker' as Researcher_Role, proj.title as proj_title
196 from
```

```

197     researcher as res inner join works_on as wo on res.res_id=wo.res_id
198     inner join project as proj on wo.project_id = proj.project_id)
199
200     union
201
202     (select res.first_name, res.last_name, 'Evaluator' as Researcher_Role, proj.title as proj_title
203     from
204     researcher as res inner join evaluation as eval on res.res_id=eval.res_id
205     inner join project as proj on eval.evaluation_id = proj.evaluation_id)
206     )
207     order by last_name, first_name;

```

- Query 3.2 (2nd View) :

```

212 • CREATE VIEW Organisations_Scientific_Fields AS
213     (select org.org_name, sf.field_name
214     from
215     organisation as org inner join project as proj on proj.org_name=org.org_name
216     inner join concerns as con on con.project_id = proj.project_id
217     inner join scientific_field as sf on sf.field_name = con.field_name)
218     order by org.org_name;

```

- Query 3.3 :

```

1  (select sf.field_name, proj.title as proj_title,
2  concat(res.first_name, ' ', res.last_name) as Researcher_Name,
3  'Administrator' as Researcher_Role
4  from scientific_field as sf
5  inner join concerns as con on sf.field_name=con.field_name
6  inner join project as proj on proj.project_id=con.project_id
7  inner join administrates as adm on adm.project_id=proj.project_id
8  inner join researcher as res on res.res_id=adm.res_id
9  where proj.ending_date>=CURDATE()
10
11  union
12
13  select sf.field_name, proj.title as proj_title,
14  concat(res.first_name, ' ', res.last_name) as Researcher_Name,
15  'Worker' as Researcher_Role
16  from scientific_field as sf
17  inner join concerns as con on sf.field_name=con.field_name
18  inner join project as proj on proj.project_id=con.project_id
19  inner join works_on as wo on wo.project_id=proj.project_id
20  inner join researcher as res on res.res_id=wo.res_id
21  where proj.ending_date>=CURDATE()
22  )
23  order by field_name, proj_title, Researcher_Role;

```

- Query 3.4 :

```
1  select org1.org_name, org1.counter, org1.starting_year from
2  (
3  select org.org_name as org_name, count(proj.project_id) as counter,
4  extract(year from proj.starting_date) as starting_year
5  from organisation org
6  inner join project proj on proj.org_name=org.org_name
7  group by org.org_name, extract(year from proj.starting_date)
8  ) org1
9  inner join
10 (
11 select org.org_name as org_name, count(proj.project_id) as counter,
12 extract(year from proj.starting_date) as starting_year from organisation org
13 inner join project proj on proj.org_name=org.org_name
14 group by org.org_name, extract(year from proj.starting_date)
15 ) org2
16 on org1.org_name=org2.org_name
17 where org1.counter=org2.counter and org1.starting_year=org2.starting_year+1
18 --
19 -- and org1.counter>=10
20 --
```

- Query 3.5 :

```
1  select concat(awa1.field_name, ' - ', awa2.field_name) as Field_Couple,
2  count(awa1.project_id) as Counter
3  from (
4  select sf.field_name as field_name, proj.project_id as project_id
5  from scientific_field sf
6  inner join concerns con on sf.field_name=con.field_name
7  inner join project proj on proj.project_id=con.project_id
8  ) awa1
9  inner join
10 (
11 select sf.field_name as field_name, proj.project_id as project_id
12 from scientific_field sf
13 inner join concerns con on sf.field_name=con.field_name
14 inner join project proj on proj.project_id=con.project_id
15 ) awa2
16 on awa1.project_id=awa2.project_id
17 where awa1.field_name < awa2.field_name
18
19 group by field_couple
20 order by Counter desc limit 3;
```


- Query 3.6 :

```

1  select concat(q.first_name,' ', q.last_name) as Researcher_Name,
2  count(*) as Number_Of_Projects
3  from
4  (select res.first_name, res.last_name, res.res_id, res.birth_date, proj.project_id
5   from
6   researcher as res
7   inner join works_on as wo on wo.res_id=res.res_id
8   inner join project as proj on wo.project_id=proj.project_id
9   where ((CURDATE()-res.birth_date)/10000 < 40)
10  and
11         proj.ending_date>=CURDATE()
12
13  union
14
15  select res.first_name, res.last_name, res.res_id, res.birth_date,proj.project_id
16  from
17  researcher as res
18  inner join administrates as adm on adm.res_id=res.res_id
19  inner join project as proj on adm.project_id=proj.project_id
20  where ((CURDATE()-res.birth_date)/10000 < 40)
21  and
22         proj.ending_date>=CURDATE()
23  )as q
24  group by res_id
25  order by Number_Of_Projects desc
26  ;

```

- Query 3.7 :

```

1  select concat(exec_proj.executive_id,' : ',exec_proj.first_name,' ',exec_proj.last_name) as Executive,
2  sum(exec_proj.grant_amount) as Total_Given_Funds
3  from
4  (
5  select exec.executive_id,exec.first_name,exec.last_name, proj.project_id,proj.grant_amount
6  from executive as exec
7   inner join project as proj on proj.executive_id=exec.executive_id
8   inner join organisation as org on org.org_name=proj.org_name
9   inner join corporation as corp on corp.corp_name=org.org_name
10 )as exec_proj
11 group by exec_proj.executive_id
12 order by Total_Given_Funds desc limit 5;

```


- Query 3.8 :

```
1  select concat(q.first_name, ' ', q.last_name) as Researcher_Name,
2  count(*) as Number_Of_Projects
3  from
4  (select res.first_name, res.last_name, res.res_id, res.birth_date, proj.project_id from
5   researcher as res
6   inner join works_on as wo on wo.res_id=res.res_id
7   inner join project as proj on wo.project_id=proj.project_id
8   where proj.ending_date>=CURDATE() and not exists
9   (
10      select del.project_id
11      from deliverable as del
12      where del.project_id=proj.project_id
13   )
14   union
15   select res.first_name, res.last_name, res.res_id, res.birth_date,proj.project_id from
16   researcher as res
17   inner join administrates as adm on adm.res_id=res.res_id
18   inner join project as proj on adm.project_id=proj.project_id
19   where proj.ending_date>=CURDATE() and not exists
20   (
21      select del.project_id
22      from deliverable as del
23      where del.project_id=proj.project_id
24   )
25  ) as q
26  group by res_id
27  having Number_Of_Projects>=5
28  order by Number_Of_Projects desc
29  ;
```

Μέρος 2.3 : Βήματα Εγκατάστασης

Κατεβάζουμε τον φάκελο SQL scripts από το github repository. Επίσης κατεβάζουμε τον φάκελο dbdemo, ο οποίος περιλαμβάνει τα αρχεία python_code.py (που κατασκευάζει την σελίδα της βάσης) και forms.py (όπου ορίζονται κάποιες κατάλληλες κλάσεις αντικειμένων ώστε να επιτευχθεί η CRUD υλοποίηση) και τα html templates της ιστοσελίδας μας.

Ανοίγουμε ένα νέο connection στο MYSQL Workbench και τρέχουμε τα δύο SQL scripts (elidek schema.sql και elidek insert.sql) ώστε να κατασκευαστεί η βάση. Τρέχοντας το αρχείο python_code.py κατασκευάζεται η εφαρμογή, η οποία είναι προσβάσιμη μέσα από το URL: <http://localhost:3000>.

Πακέτα Python που χρησιμοποιήθηκαν :

Python packages :

click 8.1.3

colorama 0.4.4

Flask 2.1.2

Flask-MySQLdb 1.0.1

Flask-WTF 1.0.1

itsdangerous 2.1.2

Jinja2 3.1.2

MarkupSafe 2.1.1

mysqlclient 2.1.0

pip 22.1.2

Werkzeug 2.1.2

WTForms 3.0.1

Μέρος 2.4 : Git Repo Link

Το link στο github repository είναι το παρακάτω :

<https://github.com/DimitrisDavidGerokostas/ELIDEK-DB>