Αναφορά Εξαμηνιαίας Εργασίας

Βάσεις Δεδομένων

Ομάδα Project 18

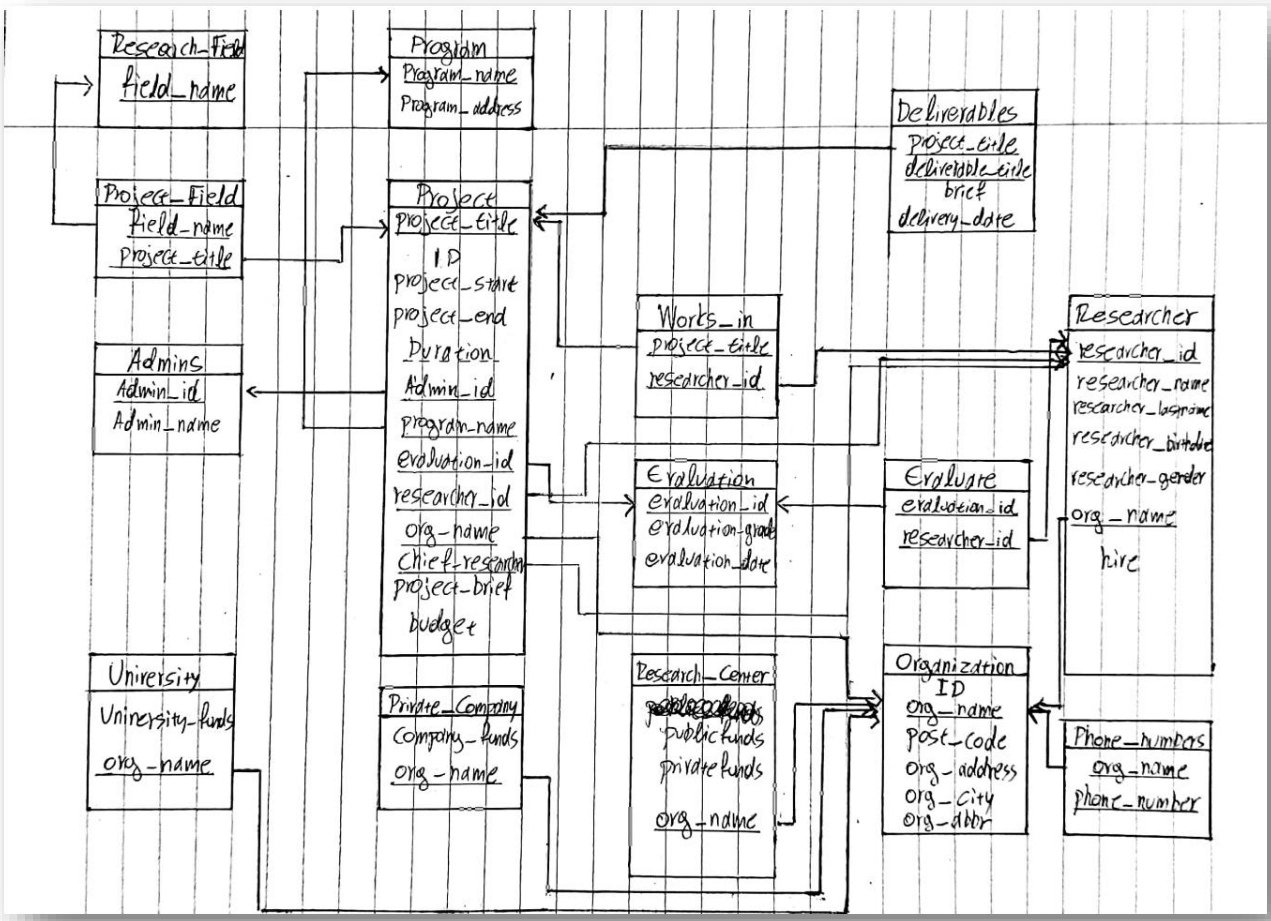
Βούγιας Κωνσταντίνος

ΑΜ: 03119144

Νικόλαος Κάσσαρης

ΑΜ: 03119188

2)



Το relational diagram που σχεδιάσαμε για την εφαρμογή μας με βάση το ER που μας δώθηκε στο οποίο:

2.1)

Οι σχέσεις Many to one και one to many με total participation στην μεριά many, αντικαταστάθηκαν από τα αντίστοιχα primary keys που τα βάλαμε στην μεριά many σαν foreign keys του αντίστοιχου table.

Στο relational diagram, σε κάθε table υπογραμμίσαμε τα primary keys καθώς και τα foreign keys, που προκύπτουν από τα primary keys των tables που συνδέονται με relations (many to one, one to many) που εξαλείφθηκαν από το ER. Τα foreign keys αυτά εξασφαλίζουν αναφορική ακεραιότητα στην βάση μας καθώς δημιουργούν κατάλληλες σχέσεις μεταξύ των tables.

Επιπλέον σε κάθε attribute έχουμε θεσπίσει (μέσα στο DDL) όρια όσον αφορά τον τύπο του, και διαθέτουν περιορισμούς σχετικά με το μέγεθος τους. Άλλος περιορισμός που μπήκε στα attributes είναι ότι δεν μπορούν να είναι κενά, να μην παίρνουν κάποια τιμή δηλαδή(NOT NULL).Τέλος, έχει ανατεθεί στα κατάλληλα, με βάση το relational diagram, attributes το αν θα είναι primary ή foreign keys.

Όσον αφορά το relational diagram έχουμε:

Το Project_Field συνδέει το κάθε Project με το Research_Field μέσα από τα foreign keys field_name και project_title . Το κάθε Project συνδέεται με το Program που το χρηματοδοτεί , μέσω της σχέσης που δημιουργείται από το foreign key program_name. Το κάθε Deliverable συνδέεται με το Project μέσω της σχέσης που δημιουργεί το foreign key project_title.Το κάθε Project συνδέεται με τα στελέχη (Admins) που δουλεύουν σε αυτό μέσα από το foreign key admin_id.Τα Pojects συνδέονται με τους researchers που δουλεύουν σε αυτό μέσα από το Foreign key researcher_id, καθώς και με τις αξιολογήσεις μέσα από το foreign key evaluation_id. Επιπλέον , το table Works_in συνδέει το table Resaercher με το Project μέσω των foreign keys Project_title, Researcher_id, ενώ το evaluate συνδέει τα tables evaluation και researcher μεσω των foreign keys evaluation_id και researcher_id. Τα university, Private_company, research_center καθως και το phone_number συνδέεται με το organization μέσω του org_name foreighn key. Τέλος το researcher συνδέεται με το organization με το org_name σαν foreign key και το project με το organization με το ίδιο foreign key.

2.3)

Ως indexes ορίσαμε:

Για τα tables Project_field, Deliverables, Works_in το Project_title.

Για τα table Researcher, Project, Phone_numbers, University, Research_Center το org_name.

Για τα tables Researcher, Project, Evaluate, Works_in το researcher_id

Για το table Project το Admin_id.

Για το table Project_Field το field_name.

Τα indexes αυτά ορίστηκαν για attributes που είναι και foreign keys και εμφανίζονται συχνά στα queries. Δημιουργώντας αυτά τα indexes επιταχύνουμε την διαδικασία εκτέλεσης των queries αυξάνοντας την αποδοτικότητα της βάσης μας.

3)

3.1

```
SELECT * FROM Project WHERE admin_id = admin

SELECT * FROM Project WHERE duration = duration

SELECT * FROM Project WHERE project_start < date AND project_end > date

SELECT * FROM Project

SELECT * FROM Project WHERE duration =duration AND admin_id = admin

SELECT * FROM Project WHERE project_start < date AND project_end >date
AND admin_id = admin

SELECT * FROM Project WHERE project_start < date AND project_end > date
AND duration = duration
```

3.2

```
CREATE VIEW r_projects AS

  SELECT Researcher.researcher_name,

  Researcher.researcher_lastname,
    Researcher.researcher_id,
    Works_in.project_title
FROM Researcher, Works_in
WHERE Researcher.researcher_id = Works_in.researcher_id;

SELECT * FROM r_projects;
DROP VIEW r_projects;

CREATE VIEW overview_project AS
    SELECT Project.project_title,
    Project.budget,
    Evaluation.evaluation_grade,
    Project.researcher_id
FROM Project, Evaluation
WHERE Project.evaluation_id=Evaluation.evaluation_id;

SELECT * FROM overview_project;

DROP VIEW overview_project;
```

3.3

```
SELECT * FROM Works_in WHERE Project_title IN (SELECT Project_title FROM
Project WHERE (project_start < DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
AND project_end > CURDATE() ) AND Project_title IN (SELECT project_title
FROM Project_Field WHERE (field_name='Basic medicine')));
```

3.4

```
create view num_of_proj as

SELECT EXTRACT(YEAR from project.project_start) as years, organization.ID as
org_ID,count(project.ID) as num_of_proj
```

```
FROM manages,organization,project

WHERE organization.ID = organization_ID AND project.ID = project_ID

GROUP BY years,org_ID;


SELECT org_ID,organization.org_name

FROM organization,(SELECT t1.org_ID, t1.num_of_proj

                FROM num_of_proj as t1, num_of_proj as t2

                WHERE (t1.org_ID = t2.org_ID AND t1.num_of_proj =
t2.num_of_proj AND t1.years - t2.years = 1)) as newtable

WHERE (org_ID = organization.ID AND num_of_proj>=10);
```

3.5

```
CREATE VIEW Couples AS SELECT c.field_name AS Field1, a.field_name AS
Field2 FROM Project_Field c, Project_Field a WHERE

(a.field_name > c.field_name AND c.project_title IN (SELECT Project_title
FROM Project) AND  a.project_title IN (SELECT Project_title FROM Project)
AND a.project_title=c.project_title);


SELECT Field1, Field2, COUNT(*) AS occurences FROM Couples GROUP BY
Field1, Field2 ORDER BY occurences DESC LIMIT 3;


DROP VIEW Couples;
```

3.6

```
CREATE VIEW current_workers AS SELECT researcher_id AS id FROM
Works_in WHERE

researcher_id IN (SELECT researcher_id FROM Works_in WHERE project_title
IN (SELECT project_title FROM Project WHERE (project_start < CURDATE()
AND project_end > CURDATE())));


CREATE VIEW current_researchers AS SELECT Researcher.researcher_name,
Researcher.researcher_lastname, current_workers.id FROM Researcher
```

INNER JOIN current_workers ON
Researcher.researcher_id=current_workers.id;

SELECT *, COUNT(*) AS times_in_projects FROM current_researchers WHERE id IN (SELECT researcher_id FROM Researcher WHERE (TIMESTAMPDIFF(YEAR, researcher_birthdate,CURDATE()) < 40)) GROUP BY id ORDER BY times_in_projects DESC LIMIT 5;

DROP ViEW current_workers;

DROP VIEW current_researchers;

3.7

CREATE VIEW best_admin AS

SELECT admins., admins.Admin_id, project.org_name as company_name, project.budget as budget

FROM Admins

INNER JOIN project ON admins.Admin_id = project.Admin_id

WHERE project.org_name IN (SELECT org_name FROM Private_company)

ORDER BY project.budget DESC;

SELECT admin_name, company_name, SUM(budget) as total_funds

FROM best_admin

GROUP BY Admin_id

ORDER BY total_funds DESC LIMIT 5;

DROP VIEW best_admin;

3.8

create VIEW no_deliverables as (

select RESEARCHER_ID from works_in where project_title in

(SELECT t1.project_title

```
FROM project t1
LEFT JOIN deliverables t2 ON t2.project_title = t1.project_title
WHERE t2.project_title IS NULL));


SELECT r.researcher_name, r.researcher_lastname, COUNT(w.researcher_id)
as project_with_no_deliverables
FROM  researcher r
INNER JOIN no_deliverables w ON w.researcher_id = r.researcher_id
GROUP BY w.researcher_id
HAVING project_with_no_deliverables >= 5
ORDER BY project_with_no_deliverables DESC;
```

ΕΦΑΡΜΟΓΗ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ

Dependencies

MySql for Windows

Python with the additional libraries:

- Flask
- Flask-MySQLdb
- Flask-WTForms

We use pip, python's package manager, to install each Python package (library) directly for the entire system or create a virtual environment with the venv module. The necessary packages for this app are listed in requirements.txt and can be installed together via pip install -r requirements.txt

The credentials folder

dbdemo/config.json

```
{
    "MYSQL_USER": "root",
    "MYSQL_PASSWORD": "",
    "MYSQL_DB": "elidek",
    "MYSQL_HOST": "localhost",
    "SECRET_KEY": "key",
    "WTF_CSRF_SECRET_KEY": "key"
}
```

Import the credentials in `__init__.py` by loading them using

```
import json
## ...
app.config.from_file("config.json", load = json.load)
```

__init__.py configures the application, including the necessary information and credentials for the database

routes.py currently contains all the endpoints and corresponding controllers

run.py launches the simple, built-in server and runs the app on it

When writing an app locally, Flask will launch a simple "development" server on which to run it:

In order to run this development server and run the app on it you can use flask run (if FLASK_APP environment variable is set to run.py) or directly with run.py.

We also include the DDL and DML scripts that are used to generate the MySQL elidek database and its data. There is also a queries.sql file that contains most of the queries used in the python files.

Everything is found on the git repo: https://github.com/NtinosVg/Elidek-Database