



Βάσεις Δεδομένων – Εξαμηνιαία Εργασία 2021-2022

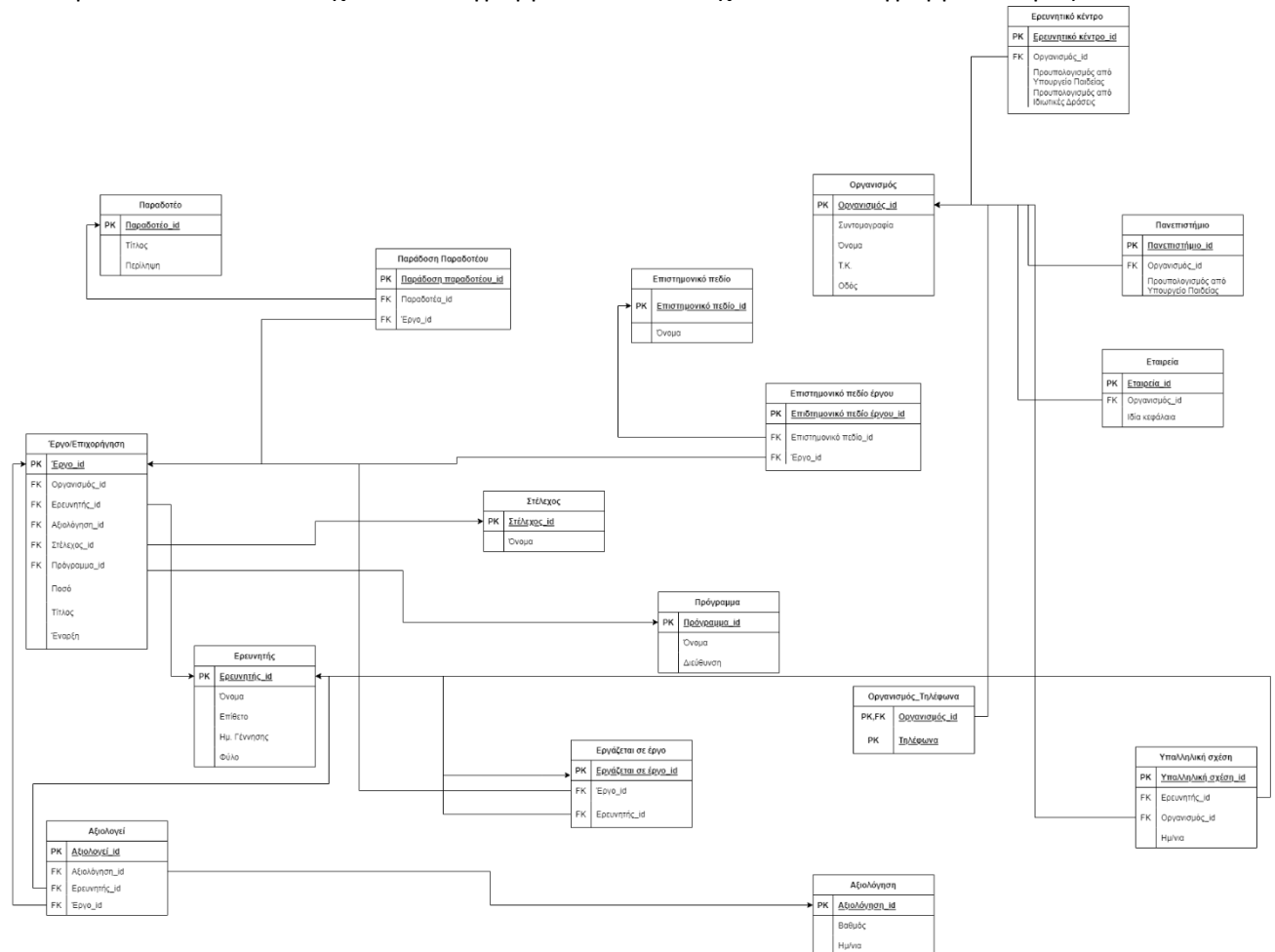
Ομάδα : A111

Μέλη : Γεωργία Αργυρού el19046

Χρήστος Σκάρλος el19911

Βαρβάρα-Κωνσταντίνα Μαγγελάκη el19180

1. Το παρακάτω αποτελεί το σχεσιακό διάγραμμα που αντιστοιχεί στο ER διάγραμμα που μας δώσατε:



2. Για την δημιουργία των παραπάνω πινάκων στην SQL χρησιμοποιήσαμε το χαρακτηριστικό NOT NULL, όταν το χαρακτηριστικό αυτό θεωρούνταν απαραίτητο για την αρχικοποίηση και την φυσική υπόσταση του στοιχείου. Φυσικά τα Primary Keys δεν γίνεται να είναι NULL. Τα Foreign Keys τα ορίζουμε ως NOT NULL, ώστε να υπάρχει συνέπεια στην βάση και να συμφωνούν με τα Primary Keys των σχέσεων στις οποίες ανήκουν.

Query για την δημιουργία της βάσης:

```
USE [master]
GO
```

```
IF NOT EXISTS (SELECT name FROM master.dbo.sysdatabases
WHERE name = 'ProjectFunding')
CREATE DATABASE [ProjectFunding] COLLATE Greek_CI_AI -- Collation : CI = Case Insensitive, AI = Accent Insensitive
GO
```

Query για την δημιουργία του πίνακα Project:

```
IF OBJECT_ID ('Project', 'U') IS NULL
BEGIN
CREATE TABLE Project
(
    Project_ID int IDENTITY(1,1) PRIMARY KEY,
    Begin_date date not null,
    End_date date not null,
    Title varchar (100) not null,
    AmountofMoney float,
    Summary varchar (500)
)
END
GO

ALTER TABLE Project
ADD Organization_ID int FOREIGN KEY REFERENCES Organization(Organization_ID)
GO

ALTER TABLE Project
ADD Researcher_ID int FOREIGN KEY REFERENCES Researcher(Researcher_ID)
GO

ALTER TABLE Project
ADD Evaluation_ID int FOREIGN KEY REFERENCES Evaluation(Evaluation_ID)
GO

ALTER TABLE Project
ADD Stem_ID int FOREIGN KEY REFERENCES Stem(Stem_ID)
GO

ALTER TABLE Project
ADD Program_ID int FOREIGN KEY REFERENCES Program(Program_ID)
GO
```

Εδώ το Project_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε έργο. Επίσης, στον πίνακα Project τα Organization_ID, Researcher_ID, Evaluation_ID, Stem_ID, Program_ID είναι Foreign Keys για τον πίνακα Project.

Query για την δημιουργία του πίνακα Researcher:

```
IF OBJECT_ID ('Researcher', 'U') IS NULL
BEGIN
CREATE TABLE Researcher
(
    Researcher_ID int IDENTITY(1,1) PRIMARY KEY,
    First_name varchar (50) not null,
    Last_name varchar (50) not null,
    Birth_date date,
    Gender varchar (10) not null
)
END
GO
```

Εδώ το Researcher_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε ερευνητή.

Query για την δημιουργία του πίνακα Stem:

```
IF OBJECT_ID ('Stem', 'U') IS NULL
BEGIN
CREATE TABLE Stem
(
    Stem_ID int IDENTITY(1,1) PRIMARY KEY,
    Stem_name varchar (50) not null
)
END
GO
```

Εδώ το Stem_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε στέλεχος.

Query για την δημιουργία του πίνακα Program:

```
IF OBJECT_ID ('Program', 'U') IS NULL
BEGIN
CREATE TABLE Program
(
    Program_ID int IDENTITY(1,1) PRIMARY KEY,
    ProgramName varchar (50) not null,
    Program_address varchar (50)
)
END
GO
```

Εδώ το Program_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε πρόγραμμα.

Query για την δημιουργία του πίνακα Evaluation:

```
IF OBJECT_ID ('Evaluation', 'U') IS NULL
BEGIN
CREATE TABLE Evaluation
(
    Evaluation_ID int IDENTITY(1,1) PRIMARY KEY,
    Grade float,
    Evaluation_date date
)
END
GO
```

Εδώ το Evaluation_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε αξιολόγηση.

Query για την δημιουργία του πίνακα Scientific_field:

```

IF OBJECT_ID ('Scientific_field', 'U') IS NULL
BEGIN
CREATE TABLE Scientific_field
(
    Scientific_field_ID int IDENTITY(1,1) PRIMARY KEY,
    Scientific_field_name varchar (50) not null
)
END
GO

```

Εδώ το Scientific_filed_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε επιστημονικό πεδίο.

Query για την δημιουργία του πίνακα Deliverable:

```

IF OBJECT_ID ('Deliverable', 'U') IS NULL
BEGIN
CREATE TABLE Deliverable
(
    Deliverable_ID int IDENTITY(1,1) PRIMARY KEY,
    Title varchar (50) not null,
    Summary varchar (50)
)
END
GO

```

Εδώ το Deliverable_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε παραδοτέο.

Query για την δημιουργία του πίνακα Organization:

```

IF OBJECT_ID ('Organization', 'U') IS NULL
BEGIN
CREATE TABLE Organization
(
    Organization_ID int IDENTITY(1,1) PRIMARY KEY,
    Abbreviation varchar (10),
    Organization_name varchar (50),
    Postcode int,
    Street varchar (50),
    City varchar (50)
)
END
GO

```

Εδώ το Organization_ID είναι Primary Key, και αυτό συμβαίνει διότι είναι το attribute εκείνο το οποίο είναι μοναδικό και ξεχωριστό για κάθε οργανισμό.

Query για την δημιουργία του πίνακα Evaluates:

```

IF OBJECT_ID ('Evaluates', 'U') IS NULL
BEGIN
CREATE TABLE Evaluates
(
    Evaluates_ID int IDENTITY(1,1) PRIMARY KEY,
    Evaluation_ID int FOREIGN KEY REFERENCES Evaluation(Evaluation_ID),
    Researcher_ID int FOREIGN KEY REFERENCES Researcher(Researcher_ID),
    Project_ID int FOREIGN KEY REFERENCES Project(Project_ID)
)
END
GO

```

Ο πίνακας Evaluates είναι μια σχέση που ενώνει τις οντότητες Evaluation, Researcher, Project. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key Evaluates_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και τρία Foreign Keys, ένα προς το Researcher, ένα προς το Evaluation και ένα προς το Project.

Query για την δημιουργία του πίνακα Employee_Relation:

```

IF OBJECT_ID ('Employee_Relation', 'U') IS NULL
BEGIN
CREATE TABLE Employee_Relation
(
    Employee_Relation_ID int IDENTITY(1,1) PRIMARY KEY,
    Researcher_ID int FOREIGN KEY REFERENCES Researcher(Researcher_ID),
    Organization_ID int FOREIGN KEY REFERENCES Organization(Organization_ID),
    Relation_date date
)
END
GO

```

Ο πίνακας Employee_Relation είναι μια σχέση που ενώνει τις οντότητες Researcher, Organization. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key Employee_Relation_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και δύο Foreign Keys, ένα προς το Researcher και ένα προς το Organization. Επίσης, περιλαμβάνει και το Relation_date κάθε σχέσης ερευνητή με οργανισμό.

Query για την δημιουργία του πίνακα WorksOnProject:

```

IF OBJECT_ID ('WorksOnProject', 'U') IS NULL
BEGIN
CREATE TABLE WorksOnProject
(
    WorksOnProject_ID int IDENTITY(1,1) PRIMARY KEY,
    Project_ID int FOREIGN KEY REFERENCES Project(Project_ID),
    Researcher_ID int FOREIGN KEY REFERENCES Researcher(Researcher_ID)
)
END
GO

```

Ο πίνακας WorksOnProject είναι μια σχέση που ενώνει τις οντότητες Researcher και Project. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key WorksOnProject_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και δύο Foreign Keys, ένα προς το Researcher και ένα προς το Project.

Query για την δημιουργία του πίνακα Delivery:

```
IF OBJECT_ID ('Delivery', 'U') IS NULL
BEGIN
CREATE TABLE Delivery
(
    Delivery_ID int IDENTITY(1,1) PRIMARY KEY,
    Deliverable_ID int FOREIGN KEY REFERENCES Deliverable(Deliverable_ID),
    Project_ID int FOREIGN KEY REFERENCES Project(Project_ID),
    Date_of_delivery date
)
END
GO
```

Ο πίνακας Delivery είναι μια σχέση που ενώνει τις οντότητες Deliverable και Project. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key Delivery_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και δύο Foreign Keys, ένα προς το Deliverable και ένα προς το Project. Επίσης, περιλαμβάνει και το Date_of_delivery κάθε σχέσης παραδοτέου με έργο.

Query για την δημιουργία του πίνακα ProjectScientificField:

```
IF OBJECT_ID ('ProjectScientificField', 'U') IS NULL
BEGIN
CREATE TABLE ProjectScientificField
(
    ProjectScientificField_ID int IDENTITY(1,1) PRIMARY KEY,
    Scientific_field_ID int FOREIGN KEY REFERENCES Scientific_field(Scientific_field_ID),
    Project_ID int FOREIGN KEY REFERENCES Project(Project_ID)
)
END
GO
```

Ο πίνακας ProjectScientificField είναι μια σχέση που ενώνει τις οντότητες ScientificField και Project. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key ProjectScientificField_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και δύο Foreign Keys, ένα προς το ScientificField και ένα προς το Project.

Query για την δημιουργία του πίνακα Organization_phones:

```
IF OBJECT_ID ('Organization_phones', 'U') IS NULL
BEGIN
CREATE TABLE Organization_phones
(
    Organization_phones_ID int IDENTITY(1,1) PRIMARY KEY,
    Organization_ID int FOREIGN KEY REFERENCES Organization(Organization_ID),
    Phone varchar (12)
)
END
GO
```

Ο πίνακας Organization_phones είναι μια σχέση που συνδέει την οντότητα Organization με τα τηλέφωνα του κάθε οργανισμού, Phone. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key

Organization_phones_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και ένα Foreign Key προς το Organization.

Query για την δημιουργία του πίνακα Research_center:

```
IF OBJECT_ID ('Research_center', 'U') IS NULL
BEGIN
CREATE TABLE Research_center
(
    Research_center_ID int IDENTITY(1,1) PRIMARY KEY,
    Organization_ID int FOREIGN KEY REFERENCES Organization(Organization_ID),
    MinistryofEducation_Budget float,
    PriavteActions_Budget float
)
END
GO
```

Ο πίνακας Research_center είναι μια σχέση που χαρακτηρίζει όσους οργανισμούς είναι ερευνητικά κέντρα. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key Research_center_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και ένα Foreign Key προς το Organization. Επίσης, περιέχει ως attributes τον προϋπολογισμό από το Υπουργείο Παιδείας και τον προϋπολογισμό Από Ιδιωτικές Δράσεις.

Query για την δημιουργία του πίνακα University:

```
IF OBJECT_ID ('University', 'U') IS NULL
BEGIN
CREATE TABLE University
(
    University_ID int IDENTITY(1,1) PRIMARY KEY,
    Organization_ID int FOREIGN KEY REFERENCES Organization(Organization_ID),
    MinistryofEducation_Budget float
)
END
GO
```

Ο πίνακας University είναι μια σχέση που χαρακτηρίζει όσους οργανισμούς είναι πανεπιστήμια. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key University_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και ένα Foreign Key προς το Organization. Επίσης, περιέχει ως attribute τον προϋπολογισμό από το Υπουργείο Παιδείας.

Query για την δημιουργία του πίνακα Company:

```

IF OBJECT_ID ('Company', 'U') IS NULL
BEGIN
CREATE TABLE Company
(
    Company_ID int IDENTITY(1,1) PRIMARY KEY,
    Organization_ID int FOREIGN KEY REFERENCES Organization(Organization_ID),
    Equity float
)
END
GO

```

Ο πίνακας Company είναι μια σχέση που χαρακτηρίζει όσους οργανισμούς είναι εταιρείες. Γι' αυτόν τον λόγο, περιέχει, εκτός από το Primary Key Company_ID που χαρακτηρίζει μοναδικά κάθε εγγραφή του, και ένα Foreign Key προς το Organization. Επίσης, περιέχει ως attribute τα ίδια Κεφάλαια.

- Έχουν οριστεί ευρετήρια, τα οποία ορίστηκαν με γνώμονα την ελαχιστοποίηση του χρόνου που απαιτείται για την υλοποίηση των queries.

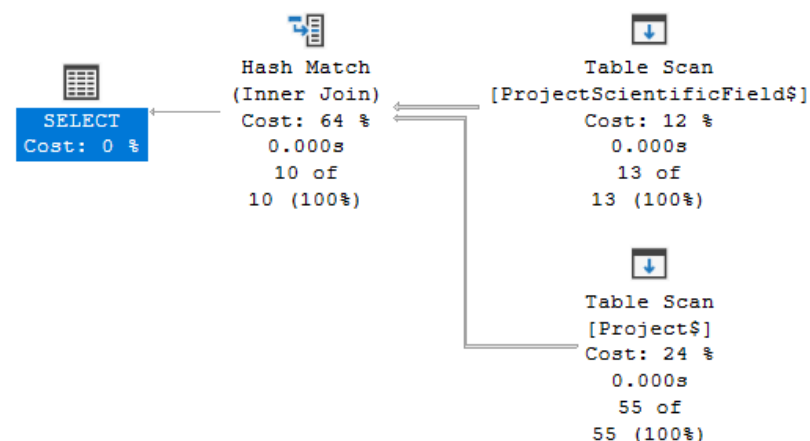
```

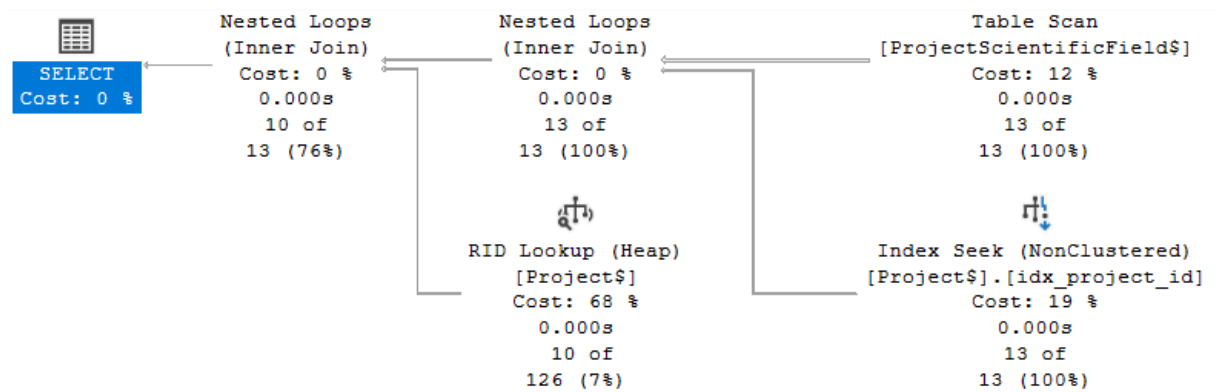
CREATE INDEX idx_project_id ON Project$(Project_ID)
CREATE INDEX idx_researcher_id ON Researcher$(Researcher_ID)
CREATE INDEX idx_organization_id ON Organization$(Organization_ID)
CREATE INDEX idx_scientific_field_id ON Scientific_field$(Scientific_field_ID)
CREATE INDEX idx_evaluation_id ON Evaluation$(Evaluation_ID)
CREATE INDEX idx_stem_id ON Stem$(Stem_ID)

```

Για να καταλήξουμε στις βέλτιστες επιλογές για indexes, ενεργοποιούμε στο Workbench την επιλογή για ενσωμάτωση του Actual Execution Plan, το οποίο μας ενημερώνει για τους πόρους επί τοις εκατό που καταναλώνει ένα query για την εκτέλεση της κάθε sql εντολής. Το παρακάτω είναι ένα execution plan για το Query 3 και παρατηρούμε ότι γίνεται χρήση ενός NonClustered Index, το οποίο βοηθά στην μείωση των υπολογιστικών πόρων.

Παρακάτω έχουμε την εκτέλεση του Query 3 πριν και μετά την χρήση ευρετηρίων:





Τα ευρετήρια επιλέγονται με βάση το χαρακτηριστικό, σύμφωνα με το οποίο γίνονται οι περισσότερες αναζητήσεις ή εκτελούνται οι συναρτήσεις της sql.

4. Τεχνολογία ανάπτυξης εφαρμογής

Για την διαχείριση και την ανάπτυξη της βάσης χρησιμοποιήθηκε το Microsoft SQL Express Server 2019 και για DBMS το SQL Server Management Studio. Για το στήσιμο του web server χρησιμοποιείται flask (Python) και για την σύνδεση μεταξύ βάσης του server χρησιμοποιείται ένας pypyodbc connector. Για το UI χρησιμοποιήθηκε HTML, CSS και Java Script. Για την επικοινωνία μεταξύ backend και frontend χρησιμοποιούνται μέθοδοι GET και POST.

- Sql server 2019
- Flask 2.0.1
- Pypyodbc 1.3.4

5. Βήματα εγκατάστασης σε λογισμικό Windows.

- Απαιτείται η εγκατάσταση του Microsoft SQL Express Server 2019.
- Αφού έχει γίνει η εγκατάσταση συνδεόμαστε στον server μέσω μιας DBMS και προτείνουμε τη χρήση του SQL Server Management Studio. Συνδεόμαστε με SQL Server Authentication και με Server Name το όνομα του υπολογιστή.
- Αφού συνδεθούμε επιτυχώς, μπορούμε να δημιουργήσουμε όλους τους πίνακες και τις απαραίτητες διασυνδέσεις μεταξύ τους. Στην εργασία έχουμε επισυνάψει τα SQL Scripts που περιέχουν μέσα τα Scripts για την δημιουργία των παραπάνω.
- Στη συνέχεια, εισάγουμε τα δεδομένα στη βάση. Τα δεδομένα μας αρχικά τα βάλαμε σε ένα αρχείο .xlsx. Χρησιμοποιήσαμε το Import/Export Wizard του Microsoft Management Studio και εισήγαμε τα δεδομένα στη βάση, κάνοντας δεξί κλικ πάνω στην βάση και επιλέγοντας Tasks -> Import data.
- Αφού έχουν οριστεί οι πίνακες, έχει γίνει η εισαγωγή των δεδομένων, έχουν δημιουργηθεί τα απαραίτητα ευρετήρια και τα views, προχωράμε στο στήσιμο του περιβάλλοντος πάνω στο οποίο τρέχει η εφαρμογή μας.

Εγκαθιστούμε τις απαραίτητες βιβλιοθήκες:

```
pip install Flask
pip install pypyodbc
```

Αφού τοποθετήσαμε όλα τα αρχεία και σιγουρευτήκαμε ότι βρίσκονται στους σωστούς φακέλους, μπορούμε να τρέξουμε την εφαρμογή μας σε κάποιον localhost. Για να μπορούμε επιτυχώς να το κάνουμε αυτό, πρέπει να βρισκόμαστε στον ίδιο φάκελο που

είναι και το αρχείο `app.py`, κάνοντας `cd` στον φάκελο αυτό κι έπειτα εκτελούμε την εντολή `python app.py`.

Το project έτρεξε στον `localhost` που θα υποδείχθηκε στο `terminal`, όταν εκτελέστηκαν οι παραπάνω εντολές.

Link για το git repo:

<https://github.com/georgiarg/projectfunding/tree/master>