

DATABASE MANAGEMENT SYSTEMS PROJECT

ALUMNI OFFICE DATABASE SYSTEM

Emilja Beneja, Endro Ferizolli, Geri Dakavelli, Klea Gjoshi, Sindi Ziu

Abstract

The implementation of a database system is crucial in maintaining secured, organised and accessible set of data, especially for offices like our Alumni Office at Epoka University. The Alumni office is responsible for keeping track and assisting graduated students, to establish and enhance a continuing relationship between the university and its alumni.

The office collects alumni personal information, career (job position, sector, company, relation) and academic information (year of graduation, study program, department). The office collects the information on the alumni that are working abroad or continuing further studies and the ones in our country.

The office is also responsible to collaborate with many companies which offer job and internship opportunities to our graduates and students. In order to provide the right university department with the presented opportunities from the different companies and their sectors, a well organised database is needed. We will analyse the effectiveness of the existing database, normalise it and then propose our own idea. This new database would provide a clear relationship between the partner companies and the students who conduct an internship or there.

Normalisation of the current Database

Firstly, we are required to analyse and inspect an existing database and judge its effectiveness. By, looking at the data, their organisation and relationship we can say that it needs to be Normalised.

Sample Database/ sheet1

ID	NAME	SURNAME	NAME SURNAME	PRIMARY EMAIL	SECONDARY EMAIL	DEPARTMENT	STUDY PROGRAM	ENROLLED YEAR	GRADUATION DATE	GENDER	CONTACT NUMBER	COUNTRY	CITY	COMPANY	POSITION	Sector
1	Brylee	Mata	Brylee Mata	yfreund@aol.com		CEN	Computer engineer	2014	2017	f	(712) 256-8632	Albania	Tirana	Softics	Programmer	IT
2	Rishi	Everett	Rishi Everett	bolow@me.com		CEN	Computer engineer	2014	2017	m	(476) 423-2125	Albania	Tirana	Softics	Programmer	IT
3	Leticia	Dyer	Leticia Dyer	thassine@msn.com		CEN	Computer engineer	2014	2017	f	(596) 229-7437	Albania	Tirana			
4	Moses	Fritz	Moses Fritz	tubesteak@msn.com		CEN	Computer engineer	2014	2017	m	(979) 902-5409	Albania	Tirana	dataPro	Programmer	IT
5	Brylee	Barker	Brylee Barker	jonathan@msn.com		CEN	Computer engineer	2015	2018	f	(722) 689-5801	Albania	Tirana	dataPro	Programmer	IT
6	George	Graham	George Graham	twofover@yahoo.com		CEN	Computer engineer	2015	2018	m	(611) 325-1896	Albania	Tirana			
7	Caylee	Huff	Caylee Huff	malin@live.com		CEN	Computer engineer	2015	2018	f	(358) 701-4820	Albania	Tirana			
8	Rory	Castro	Rory Castro	lishoy@live.com		CEN	Computer engineer	2016	2019	m	(480) 419-4722	Albania	Tirana	Softics	Programmer	IT
9	Dominic	Reed	Dominic Reed	pressoff@mac.com		CEN	Computer engineer	2016	2020	m	(828) 288-6667	Albania	Tirana	Softics	Programmer	IT
10	Conor	Fields	Conor Fields	mudip@att.net		CEN	Computer engineer	2016	2019	m	(281) 695-6046	Albania	Tirana	Softics	Programmer	IT

- ‘No’ is a primary key, acting as an ID for each student in the database (200 total).
- Each Alumni has a unique No, a name and a surname.
- Each Alumni can have a primary email, secondary email, a record of the year they enrolled and graduated, a record of their gender, contact number, city and country they operate in and their actual job position if any.
- Each Alumni has finished their studies in a specific Department, corresponding to a Study Program.
- They work in a Company which operates in a specific Sector.

No	Name	Surname	Name Surname	Primary Email	Secondary Email	Department	Study Program	Enrolled Year	Graduation Date	Gender	Contact Number	Country	City	Company	Position	Sector
----	------	---------	--------------	---------------	-----------------	------------	---------------	---------------	-----------------	--------	----------------	---------	------	---------	----------	--------

Diagram showing functional dependencies (FDs) and transitive dependencies (TDs) for the initial table. FDs are indicated by red arrows: (No) → (Name, Surname, Name Surname, Primary Email, Secondary Email, Department, Study Program, Enrolled Year, Graduation Date, Gender, Contact Number, Country, City, Company, Position, Sector); (Name, Surname) → (Name Surname); (Department) → (Study Program); (Company) → (Sector). TDs are indicated by red arrows: (Name, Surname) → (Name Surname); (Department) → (Study Program); (Company) → (Sector).

- 1) Name, Surname => Name Surname (T)
- 2) Department => Study Program (T)
- 3) Company => Sector (T)
- 4) No => Name, Surname, Name Surname, Department, Study Program, Company, Sector, Primary email, Secondary email, Enrolled year, Graduation Date, Gender, Contact Number, Country, City, Position (F)

- 1st Normalization Form: Table is already in 1NF as there are no multivalues.
- 2nd Normalization Form: Table is already in 2NF as we have no partial key dependencies.
- 3rd Normalization Form: we draw the new tables

Table 1

<u>Name</u>	<u>Surname</u>	Name Surname
-------------	----------------	--------------

Diagram showing functional dependency (FD) for Table 1: (Name, Surname) → (Name Surname) (F).

Table 2

<u>Department</u>	Study Program
-------------------	---------------

Diagram showing functional dependency (FD) for Table 2: (Department) → (Study Program) (F).

Table 3

<u>Company</u>	Sector
----------------	--------

Diagram showing functional dependency (FD) for Table 3: (Company) → (Sector) (F).

Table 4

No	Name	Surname	Primary Email	2 nd Email	Department	Enrolled Year	Grad Year	Gender	Contact	Country	City	Company	Position
----	------	---------	---------------	-----------------------	------------	---------------	-----------	--------	---------	---------	------	---------	----------

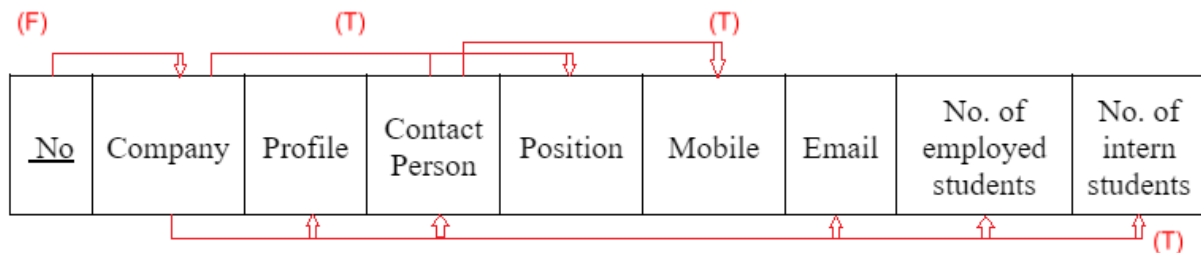
Diagram showing functional dependencies (FDs) and transitive dependencies (TDs) for Table 4. FDs are indicated by red arrows: (No) → (Name, Surname, Primary Email, 2nd Email, Department, Enrolled Year, Grad Year, Gender, Contact, Country, City, Company, Position); (Name, Surname) → (Name Surname); (Department) → (Study Program); (Company) → (Sector). TDs are indicated by red arrows: (Name, Surname) → (Name Surname); (Department) → (Study Program); (Company) → (Sector).

Sample Database/ sheet2

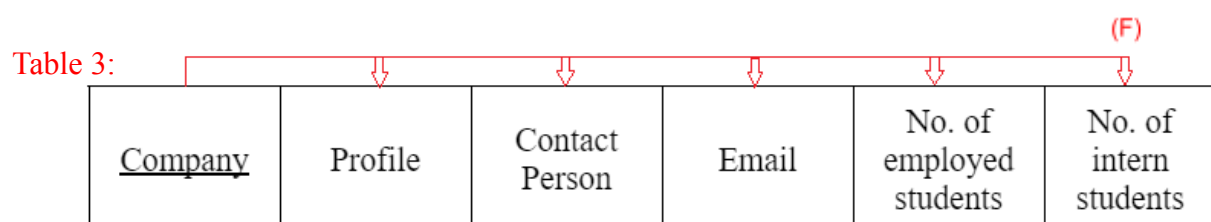
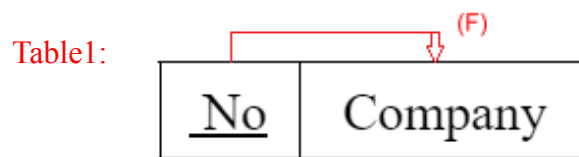
NO	COMPANY	PROFILE	CONTACT PERSON	POSITION	MOBILE	EMAIL	NO OF EMPLOYED STUDENTS (ALUMNI)	NO OF INTERNSHIP STUDENTS
1	Softics	Programming	Alfred wildrow	CTO	069 3322145	softics@email.com	26	50
2	DataPro	Programming	Joe Wind	CTO	069 4488775	datapro@gmail.ocm	28	60
3	TDT Construction	Construction	Maggie Dean	HR	069 5896322	tdt@email.com	10	50
4	DELTA	Sales	Jenny Ortega	Manager	068 2233658	delta@email.com	19	25
5	Brunes	Sales	Maria Falco	HR	068 7744569	brunes@email.com	7	7
6	Law&Order	Law Firm	Smith Jones	CTO	066 2255698	law&order@gmail.ocm	3	10
7	Eglo	Sales	Diego Perkins	Manager	066 3216547	eglo@email.com	8	8
8	Ministry of foreign affai	Political/state	Alma Harku	HR	067 5566984	ministry of foreign affairs@gmail.ocm	2	2
9	onTop Channel	media	Besa Hoxha	HR	067 4512369	ontop_channel@gmail.ocm	2	10
10	Savings Bank	bank	Altin Dema	Manager	068 2369854	savings_bank@gmail.ocm	7	12

- 'No' is a primary key, acting as an ID for each company in the database (10 total).
- Each Company has a unique No, a company name, profile, contact person, email address, number of employed students and number of internship students.
- A contact persons job position is defined by the company in which they work.

- 1) No => Company (F)
- 2) Company => Contact Person, Position (T)
- 3) Contact Person => Mobile (T)
- 4) Company => Profile, Contact Person, Email, No of employed students, No of internship students (T)



- 1st Normalization Form: Table is already in 1NF as there are no multivalues.
- 2nd Normalization Form: Table is already in 2NF as we have no partial key dependencies.
- 3rd Normalization Form: we draw the new tables



Introduction of our proposal

Steps:

- Introduction of our idea
- Design of ER Diagram
- Design of RS
- SQL code implementation

The importance of this project is to create a functional database which will help the Alumni Office to properly organise, access and update data for its alumni. As we saw from the previous section, the current database has many issues regarding dependencies, data managing and organisation, also, it did not provide clear, concise and detailed information about a graduated individual.

With our proposed database, the Alumni Office will be able to collect personal information, career and academic information, and operate on them without struggle.

In more detail:

-No ID -Name Surname -Primary email -Secondary email -Enrolled year -Study program -Graduation year -Gender -Contact number -Country -City - Job position -Sector (state/private) - Job related to your area of study (yes/no) - Further studies -Company.

All these data are identifying attributes for each graduated student.

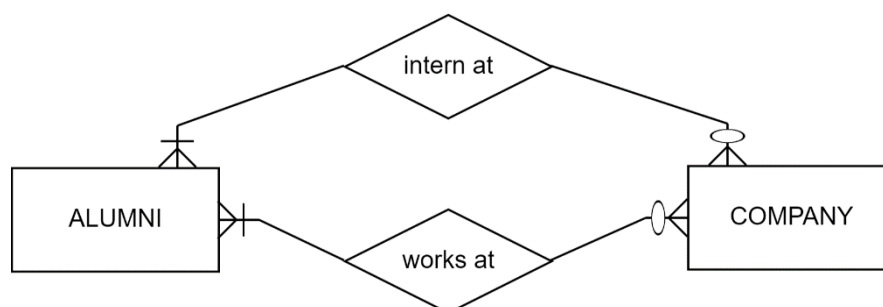
Alumni can either continue further studies, which in our scheme will be another entity in relation with the Alumni. This entity will provide information such as: type of study program, program name, department, university and state.

Otherwise, the graduates can conduct postgraduate internships or start working in one of the partner companies, or even another company. The Company database should carry information about: -Company name -Description/ profile -Email address/ Phone number -Number of students that have internship per year -Job opportunity at the company after internship.

The employment of the alumni in these companies expresses the relationship this two databases have with each-other.

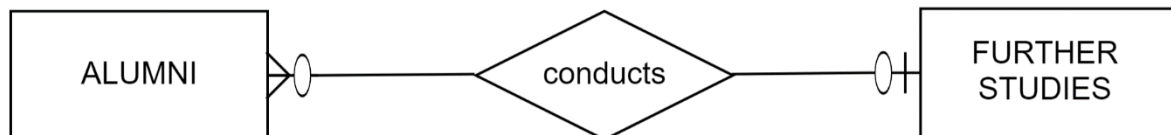
If we firstly study the Alumni and Company as two separated tables, each with its own attributes and relations, we can say that they have a **many-to-many** relationship (a graduate may work as full-time or part-time employee in many or in none partner company, a partner company can employ many or none graduates from our university).

Simple relation scheme outline:



Furthermore, while analysing the data of Alumni, we have thought to consider the option to conduct further studies as another entity related with the Alumni. Since an alumni can only pursue one (or none) further study program and a study program can be conducted by many or none graduates, this would be an optional **one-to-many** relationship.

Simple relation scheme outline:

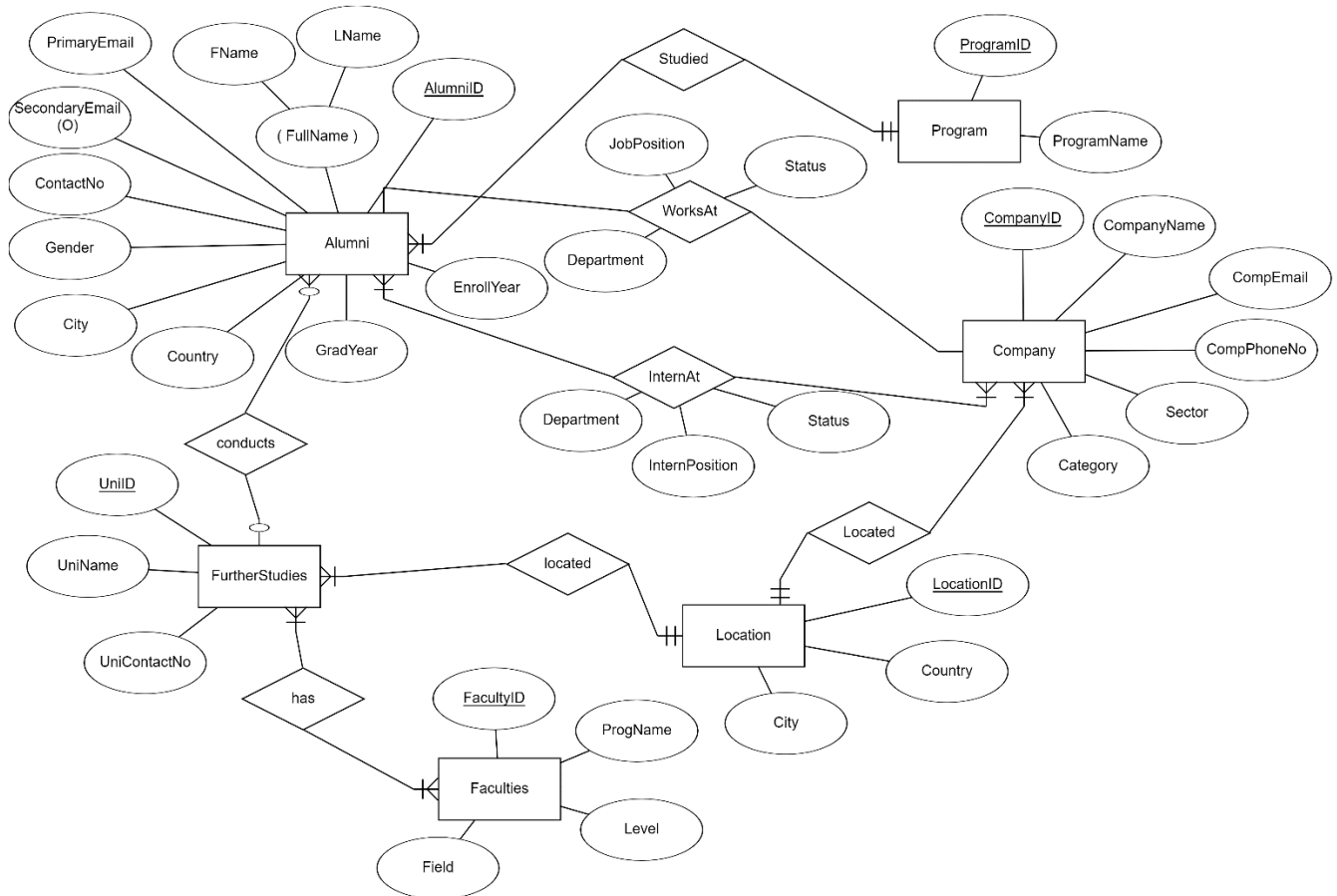


Our proposal aims to clearly organise data and keep records for the further development of our alumni. We are providing a space for more information, including the possibility of a graduated student to pursue a further education.

1.ERD SCHEMA

Now to fully display our idea we have created the full ERD schema. The **Alumni entity** has all needed attributes as required by the client (*Alumni ID, Full name which is a composite attribute with First Name and Last Name, Primary email and an optional secondary email, contact number, gender, country and city, enrolment year and graduation year*). Alumni has a relationship **many-to-one** with a **Program entity** (because one student is registered in only one program, but in one program we have many students registered) which contains attributes regarding the study program the alumni has conducted in our university such as: *Program ID, program name*. Also, as required, it will be connected with the **Company entity** (the relationship is discussed in the previous section) which will have the required attributes: *company ID, company name, company email, company phone number, sector (private/public/foundation/non-profit organisation) and category (IT/finance/law/architecture/etc)*. The **Works At and Intern At** relationships between Alumni and Company will have the attributes: *Department (both relationships), status (again both, defines if it is current/active or passed/inactive), job position (for works at), intern position (for intern at)*. Alumni is also connected with the **Further Studies entity** which expresses the possibility of a graduate student to pursue a further education (relationship explained in previous sector). This entity will have the attributes: *University ID, University Name and contact number*. Further Studies is also connected with the **Faculties entity** in a **many-to-many** relationship because a university can have multiple faculties and multiple faculties are found in many universities. The Faculties entity contains attributes regarding the program they are following such as: *Faculty ID, field (similar with department), program name and level (bachelor, master, doctorate)*. Further Studies is also connected with a **Location entity** in a **many-to-one** relationship because a university is located in one location, but in one location can be located more than one universities. Its attributes will be: *Location ID, City, Country*.

The Company entity will also be connected with the Location entity with the same type of relationship and the same reasons.



2. RELATIONAL SCHEMA

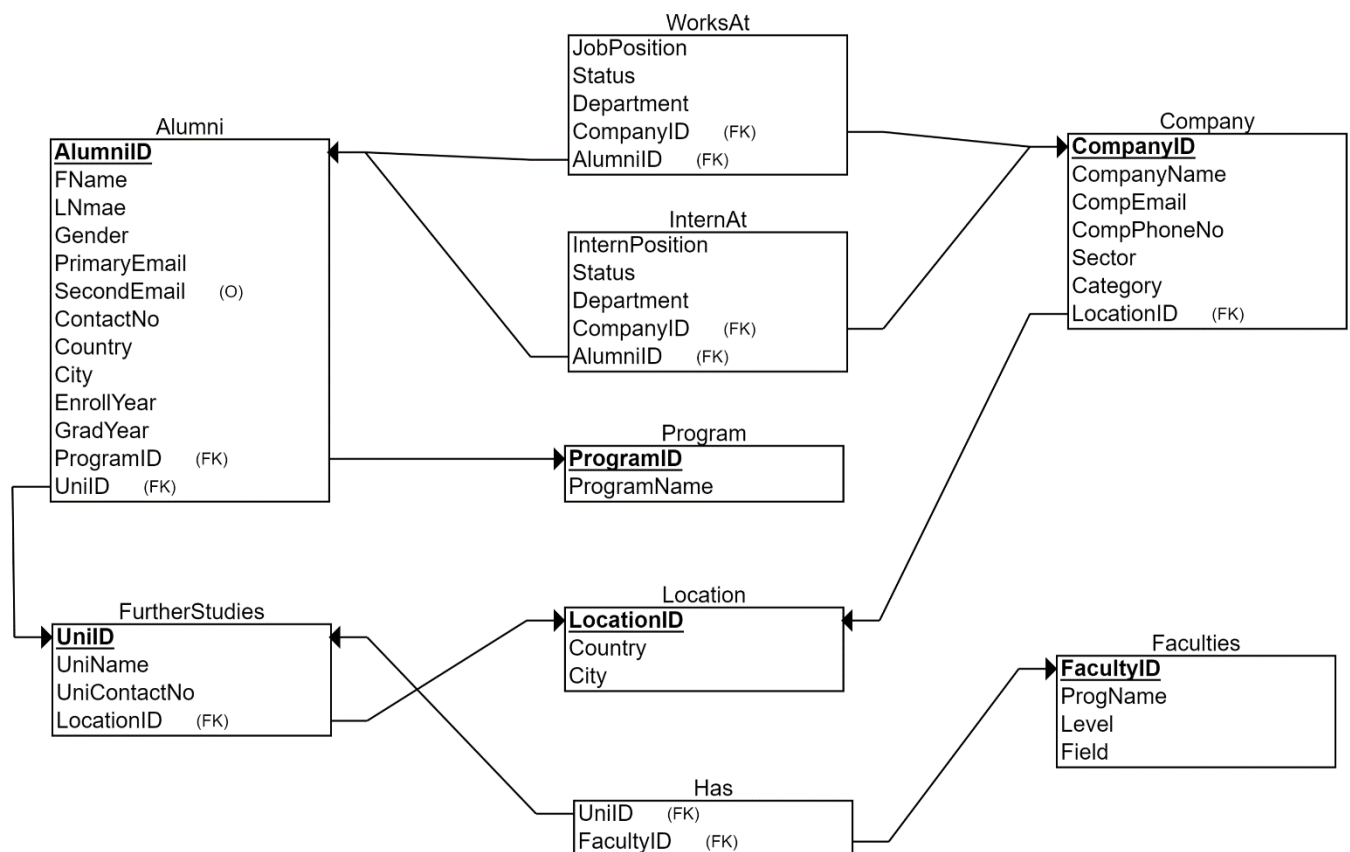
After designing the ERD we can easily design a relational schema based on the entities, their attributes and their relations. Starting with the **Alumni table** we have all the Alumni attributes with **AlumniID** as a primary key. Also, we have **ProgramID** and **UniID** as foreign keys. The foreign keys have passed to the Alumni table as a reference to the **Program** and **Further Studies** tables respectively because of the type of relationship that these tables have with Alumni (**one-to-many**).

We have the **Program table** with **ProgramID** as primary key and all its attributes. There are no foreign keys in this table as there is no other relationship that needs to be referred by this table.

The **Company table** with **CompanyID** as a primary key holds all its mentioned attributes and the **LocationID** as a foreign key which is a reference for the Location table. The relationship between the Alumni and the Company is **many-to-many**, therefore we need to create a separate table for **WorksAt** relationship and one for **InternAt** relationship. These two tables will have their own mentioned attributes and **AlumniID**, **CompanyID** as foreign keys to refer to the respective tables.

The **FurtherStudies** table has its own attributes with **UniID** as primary key and *LocationID* as foreign key to the *Location* table. Both Company and FurtherStudies tables have a reference to the **Location** table with **LocationID** primary key and no foreign keys. The FurtherStudies table has a many-to-many relationship **UniCategory** table with **ProgID** as primary key, which means we need another separate table for the relationship **Has** which will only have *ProgID* and *UniID* has foreign keys for *UniCategory* and *FurtherStudies* tables respectively.

We managed to create a database which is already normalised in all forms. Our entities tables are well organised and thought to create no troubles in the future for users, whether they want to update, add or delete data.



3. SQL IMPLEMENTATION

To create the actual database, we need to implement it with SQL queries. This is the coding part of our project which gives a more concrete view of our tables and what we have to achieve.

3.1 CREATING TABLES

We have to create all the tables in RS using SQL.

Firstly, we create the tables which don't hold any foreign keys or don't hold any direct connection with a certain table in order for the updates and modifications not to affect other referred tables.

SQL TABLES:

```
CREATE TABLE Program(  
  programID VARCHAR(4) NOT NULL,  
  programName VARCHAR(20) NOT NULL,  
  PRIMARY KEY(programID)  
);  
  
CREATE TABLE Location(  
  locationID CHAR(4) NOT NULL,  
  locCountry VARCHAR(20) NOT NULL,  
  locCity VARCHAR(20) NOT NULL,  
  PRIMARY KEY (locationid)  
);  
  
CREATE TABLE FurtherStudies(  
  uniID VARCHAR(4) NOT NULL,  
  uniName VARCHAR(20) NOT NULL,  
  uniContact VARCHAR(20) NOT NULL,  
  locationID CHAR(4) NOT NULL,  
  PRIMARY KEY(uniID),  
  FOREIGN KEY (locationID)  
  REFERENCES Location(locationID)  
);  
  
CREATE TABLE Faculties(  
  progID VARCHAR(4) NOT NULL,  
  progName VARCHAR(20) NOT NULL,  
  field VARCHAR(20) NOT NULL,  
  level VARCHAR(20) NOT NULL,  
  PRIMARY KEY(progID)  
);  
  
CREATE TABLE FS_Has_Faculties(  
  progID VARCHAR(4) NOT NULL,  
  uniID VARCHAR(4) NOT NULL,  
  PRIMARY KEY(uniID, progID),  
  FOREIGN KEY (progID)  
  REFERENCES Faculties(progID),  
  FOREIGN KEY (uniID)  
  REFERENCES FurtherStudies(uniID));
```

Table program does not have any foreign keys.

Table Location does not have foreign keys as well.

Further studies table is created because it only has the LocationID foreign key which referees to the already created Location table.

Faculties table does not have foreign keys.

Has table only has the foreign keys uniID and progID for FurtherStudies and Faculties.

```
CREATE TABLE Alumni(
alumniID INT NOT NULL,
firstname VARCHAR(20) NOT NULL,
lastname VARCHAR(20) NOT NULL,
gender VARCHAR(10) NOT NULL,
primaryemail VARCHAR(20) NOT NULL,
secondaryemail VARCHAR(20),
contactNo VARCHAR(20) NOT NULL,
country VARCHAR(25) NOT NULL,
city VARCHAR(25) NOT NULL,
enrollYear INT NOT NULL,
gradYear INT NOT NULL,
programID VARCHAR(4) NOT NULL,
uniID VARCHAR(4) NOT NULL,
PRIMARY KEY(alumniID),
FOREIGN KEY (programID)
REFERENCES Program(programID),
FOREIGN KEY (uniID)
REFERENCES FurtherStudies(uniID)
);
```

```
CREATE TABLE Company(
companyID INT NOT NULL,
companyname VARCHAR(25) NOT NULL,
compemail VARCHAR(20) NOT NULL,
comphoneNO BIGINT(15) NOT NULL,
sector VARCHAR(10) NOT NULL,
category VARCHAR(20) NOT NULL,
locationID CHAR(4) NOT NULL,
PRIMARY KEY (companyID),
FOREIGN KEY (locationID)
REFERENCES Location(locationID)
);
```

```
CREATE TABLE WorksAt(
jobposition VARCHAR(25) NOT NULL,
department VARCHAR(25) NOT NULL,
status VARCHAR(25) NOT NULL,
alumniID CHAR(3) NOT NULL,
companyID VARCHAR(4) NOT NULL,
PRIMARY KEY (alumniID, companyID),
FOREIGN KEY (alumniID)
REFERENCES alumni(alumniID),
FOREIGN KEY (companyID)
REFERENCES company(companyID)
);
```

```
CREATE TABLE InternAt(
internposition VARCHAR(25) NOT NULL,
departament VARCHAR(25) NOT NULL,
status VARCHAR(25) NOT NULL,
alumniID CHAR(3) NOT NULL,
companyID VARCHAR(4) NOT NULL,
PRIMARY KEY (alumniID, companyID),
FOREIGN KEY (alumniID)
REFERENCES alumni(alumniID),
FOREIGN KEY (companyID) REFERENCES company(companyID));
```

Alumni table has all its attributes and also foreign keys programID and UniID which refer to Program and FurtherStudies tables which are already created.

Company table also has a foreign key LocationID for Location table, already created.

WorksAt table has the foreign keys AlumniID and CompanyID for Alumni and Company table, already created, which hold their relationship.

InternAt table follows the same logic with WorksAt table.

3.2 POPULATION OF TABLES

--POPULATION--

```
INSERT INTO Program (programID, programName)
VALUES
```

```
    ('CEN', 'Computer Engineering'),
    ('SWE', 'Software Engineering'),
    ('ARCH', 'Architecture'),
    ('ECE', 'Electrical and Computer Engineering'),
    ('BINF', 'Bioinformatics'),
    ('PIR', 'Political Science and International Relations'),
    ('BAF', 'Business Administration and Finance'),
    ('IML', 'International Relations and European Studies'),
    ('CE', 'Civil Engineering'),
    ('HIS', 'History');
```

-- Location table insertions

```
INSERT INTO Location (locationID, locCountry, locCity) VALUES
```

```
    ('ALT', 'Albania', 'Tirana'),
    ('ALD', 'Albania', 'Durrës'),
    ('ALSH', 'Albania', 'Shkoder'),
    ('ALV', 'Albania', 'Vlore'),
    ('ALS', 'Albania', 'Sarande'),
    ('ALTR', 'Albania', 'Tropoje'),
    ('ALF', 'Albania', 'Fier'),
    ('ITL', 'Italy', 'Rome'),
    ('TUI', 'Turkey', 'Istanbul'),
    ('TUA', 'Turkey', 'Ankara'),
    ('TA', 'Tajikistan', 'Dushambe'),
    ('EG', 'Egypt', 'Cairo');
```

-- FurtherStudies table insertions

```
INSERT INTO FurtherStudies (uniID, uniName, uniContact, locationID) VALUES
```

```
    ('EPK', 'Epoka University', 'epokauniversity@epoka.edu.al', 'ALT'),
    ('UT', 'University of Tirana', 'universityoftirana@unitir.edu.al', 'TIR'),
    ('UAMD', 'University of Aleksander Moisiu Durrës', 'uamd@uamd.edu.al', 'ALD');
```

-- UniCategory table insertions

```
INSERT INTO Faculties (progID, progName, field, level) VALUES
```

```
    ('CEN', 'Computer Engineering', 'Engineering', 'Bachelor'),
    ('CENG', 'Computer Engineering', 'Engineering', 'Master'),
    ('SWE', 'Software Engineering', 'Engineering', 'Bachelor'),
    ('SWEG', 'Software Engineering', 'Engineering', 'Master'),
    ('ARCH', 'Architecture', 'Architecture', 'Specialisation'),
    ('ECE', 'Electrical Engineering', 'Engineering', 'Master'),
    ('BINF', 'Business Informatics', 'Business', 'Bachelor'),
    ('BINFG', 'Business Informatics', 'Business', 'Master'),
    ('PIR', 'Political Science', 'Social Sciences', 'Bachelor'),
    ('PIRG', 'Political Science', 'Social Sciences', 'Master'),
    ('BAF', 'Banking and Finance', 'Business', 'Master'),
    ('BAFG', 'Banking and Finance', 'Business', 'Bachelor'),
    ('IML', 'Logistics', 'Business', 'Bachelor'),
    ('IMLG', 'Logistics', 'Business', 'Master'),
    ('CE', 'Civil Engineering', 'Engineering', 'Bachelor'),
    ('CEG', 'Civil Engineering', 'Engineering', 'Master'),
    ('ECO', 'Economics', 'Economics', 'Bachelor'),
    ('ECOG', 'Economics', 'Economics', 'Master');
```

-- FS_Has_UniCategory table insertions

```
INSERT INTO FS_Has_Faculties (uniID, progID) VALUES
```

```
    ('EPK', 'CENG'),
    ('UT', 'CENG'),
    ('UT', 'ARCH'),
    ('UAMD', 'CENG');
```

```

INSERT INTO Alumni (alumniID, firstname, lastname, gender, primaryemail,
secondaryemail, contactNo, country, city, enrollYear, gradYear, programID,
uniID)
VALUES
('A223456789', 'John', 'Doe', 'Male', 'johndoe@gmail.com',
'john.doe@example.com', '+355 1234567890', 'ALBANIA', 'TIRANA', 2011, 2013,
'CEN', 'EPK'),
('B234567890', 'Jane', 'Smith', 'Female', 'janesmith@gmail.com',
'jane.smith@example.com', '+90 1234567890', 'TURKEY', 'ISTANBUL', 2012, 2015,
'SWE', 'UAMD'),
('C245678901', 'Michael', 'Johnson', 'Male', 'michaeljohnson@gmail.com',
'michael.johnson@example.com', '+39 1234567890', 'ITALY', 'ROME', 2013, 2016,
'ARCH', 'UT'),
('D256789012', 'Emily', 'Williams', 'Female', 'emilywilliams@gmail.com',
'emily.williams@example.com', '+992 1234567890', 'TAJIKISTAN', 'DUSHANBE',
2014, 2017, 'ECE', 'EPK'),
('E267890123', 'Daniel', 'Brown', 'Male', 'danielbrown@gmail.com',
'daniel.brown@example.com', '+20 1234567890', 'EGYPT', 'CAIRO', 2015, 2018,
'BINF', 'UAMD'),
('F278901234', 'Olivia', 'Taylor', 'Female', 'oliviataaylor@gmail.com',
'olivia.taylor@example.com', '+355 1234567890', 'ALBANIA', 'SHKODER', 2016,
2019, 'PIR', 'EPK'),
('G289012345', 'Matthew', 'Miller', 'Male', 'matthewmiller@gmail.com',
'matthew.miller@example.com', '+90 1234567890', 'TURKEY', 'ANKARA', 2017, 2020,
'BAF', 'UAMD'),
('H290123456', 'Sophia', 'Anderson', 'Female', 'sophiaanderson@gmail.com',
'sophia.anderson@example.com', '+39 1234567890', 'ITALY', 'VLORE', 2018, 2021,
'IML', 'UT'),
('I201234567', 'James', 'Wilson', 'Male', 'jameswilson@gmail.com',
'james.wilson@example.com', '+992 1234567890', 'TAJIKISTAN', 'KHUJAND', 2019,
2022, 'CE', 'EPK'),
('J212345678', 'Isabella', 'Clark', 'Female', 'isabellaclark@gmail.com',
'isabella.clark@example.com', '+20 1234567890', 'EGYPT', 'ALEXANDRIA', 2020,
2023, 'SWE', 'UT'),
('U201234567', 'Alexander', 'Young', 'Male', 'alexanderyoung@gmail.com',
'alexander.young@example.com', '+355 1234567890', 'ALBANIA', 'DURRES', 2008,
2011, 'CE', 'EPK'),
('V212345678', 'Mia', 'Lee', 'Female', 'mialeee@gmail.com',
'mia.lee@example.com', '+90 1234567890', 'TURKEY', 'ISTANBUL', 2009, 2012,
'SWE', 'UAMD'),
('W223456789', 'William', 'Garcia', 'Male', 'williamgarcia@gmail.com',
'william.garcia@example.com', '+39 1234567890', 'ITALY', 'ROME', 2010, 2013,
'ARCH', 'UT'),
('X334567890', 'Ava', 'Martinez', 'Female', 'avamartinez@gmail.com',
'ava.martinez@example.com', '+992 1234567890', 'TAJIKISTAN', 'KHOROG', 2011,
2014, 'ECE', 'EPK'),
('Y345678901', 'Liam', 'Lopez', 'Male', 'liamlopez@gmail.com',
'liam.lopez@example.com', '+20 1234567890', 'EGYPT', 'CAIRO', 2012, 2015,
'BINF', 'UT'),
('Z356789012', 'Charlotte', 'Hernandez', 'Female',
'charlottehernandez@gmail.com', 'charlotte.hernandez@example.com', '+355
1234567890', 'ALBANIA', 'VLORE', 2013, 2016, 'PIR', 'EPK'),
('A367890123', 'Noah', 'Gonzalez', 'Male', 'noahgonzalez@gmail.com',
'noah.gonzalez@example.com', '+90 1234567890', 'TURKEY', 'ANKARA', 2014, 2017,
'BAF', 'UAMD'),
('B378901234', 'Emma', 'Perez', 'Female', 'emmaperez@gmail.com',
'emma.perez@example.com', '+39 1234567890', 'ITALY', 'TIRANA', 2015, 2018,
'IML', 'EPK'),
('C389012345', 'Jacob', 'Robinson', 'Male', 'jacobrobinson@gmail.com',
'jacob.robinson@example.com', '+992 1234567890', 'TAJIKISTAN', 'DUSHANBE',
2016, 2019, 'CE', 'EPK'),
('D390123456', 'Olivia', 'Cook', 'Female', 'oliviacoook@gmail.com',
'olivia.cook@example.com', '+20 1234567890', 'EGYPT', 'ALEXANDRIA', 2017, 2020,
'SWE', 'UT'),

```

('Y390123456', 'Amelia', 'Gray', 'Female', 'ameliagray@gmail.com',
'amelia.gray@example.com', '+355 1234567890', 'ALBANIA', 'SHKODER', 2018, 2021,
'CE', 'EPK'),
('Z301234567', 'Henry', 'Adams', 'Male', 'henryadams@gmail.com',
'henry.adams@example.com', '+90 1234567890', 'TURKEY', 'ISTANBUL', 2019, 2022,
'SWE', 'UAMD'),
('A312345678', 'Sofia', 'Campbell', 'Female', 'sofiacampbell@gmail.com',
'sofia.campbell@example.com', '+39 1234567890', 'ITALY', 'ROME', 2020, 2023,
'ARCH', 'UT'),
('B423456789', 'Benjamin', 'Hill', 'Male', 'benjaminhill@gmail.com',
'benjamin.hill@example.com', '+992 1234567890', 'TAJIKISTAN', 'KHUJAND', 2008,
2011, 'ECE', 'UAMD'),
('C434567890', 'Avery', 'Russell', 'Female', 'averyrussell@gmail.com',
'avery.russell@example.com', '+20 1234567890', 'EGYPT', 'CAIRO', 2009, 2012,
'BINF', 'EPK'),
('D445678901', 'Elijah', 'Griffin', 'Male', 'elijahgriffin@gmail.com',
'elijah.griffin@example.com', '+355 1234567890', 'ALBANIA', 'VLORE', 2010,
2013, 'PIR', 'EPK'),
('E456789012', 'Scarlett', 'Diaz', 'Female', 'scarlettdiaz@gmail.com',
'scarlett.diaz@example.com', '+90 1234567890', 'TURKEY', 'ANKARA', 2011, 2014,
'BAF', 'EPK'),
('F467890123', 'Lucas', 'Hayes', 'Male', 'lucashayes@gmail.com',
'lucas.hayes@example.com', '+39 1234567890', 'ITALY', 'TIRANA', 2012, 2015,
'IML', 'HRVD1'),
('G578901234', 'Chloe', 'Scott', 'Female', 'chloescott@gmail.com',
'chloe.scott@example.com', '+992 1234567890', 'TAJIKISTAN', 'DUSHANBE', 2013,
2016, 'CE', 'EPK'),
('H89012345', 'Wyatt', 'Ramirez', 'Male', 'wyattramirez@gmail.com',
'wyatt.ramirez@example.com', '+20 1234567890', 'EGYPT', 'ALEXANDRIA', 2014,
2017, 'SWE', '-'),
('Q578901234', 'Zoe', 'Phillips', 'Female', 'zoephillips@gmail.com',
'zoe.phillips@example.com', '+355 1234567890', 'ALBANIA', 'SHKODER', 2015,
2018, 'CE', '-'),
('R589012345', 'Nathan', 'Reed', 'Male', 'nathanreed@gmail.com',
'nathan.reed@example.com', '+90 1234567890', 'TURKEY', 'ISTANBUL', 2016, 2019,
'SWE', 'UAMD'),
('S590123456', 'Madison', 'Cooper', 'Female', 'madisoncooper@gmail.com',
'madison.cooper@example.com', '+39 1234567890', 'ITALY', 'ROME', 2017, 2020,
'ARCH', 'EPK'),
('T501234567', 'Aaron', 'Ward', 'Male', 'aaronward@gmail.com',
'aaron.ward@example.com', '+992 1234567890', 'TAJIKISTAN', 'KHUJAND', 2018,
2021, 'ECE', 'EPK'),
('U512345678', 'Grace', 'Bell', 'Female', 'gracebell@gmail.com',
'grace.bell@example.com', '+20 1234567890', 'EGYPT', 'CAIRO', 2019, 2022,
'BINF', '-'),
('V523456789', 'Ethan', 'Reynolds', 'Male', 'ethanreynolds@gmail.com',
'ethan.reynolds@example.com', '+355 1234567890', 'ALBANIA', 'VLORE', 2020,
2023, 'PIR', '-'),
('W534567890', 'Lily', 'Barnes', 'Female', 'lilybarnes@gmail.com',
'lily.barnes@example.com', '+90 1234567890', 'TURKEY', 'ANKARA', 2008, 2011,
'BAF', 'EPK'),
('X645678901', 'Carter', 'Long', 'Male', 'carterlong@gmail.com',
'carter.long@example.com', '+39 1234567890', 'ITALY', 'TIRANA', 2009, 2012,
'IML', 'EPK'),
('Y656789012', 'Victoria', 'Flores', 'Female', 'victoriaflores@gmail.com',
'victoria.flores@example.com', '+992 1234567890', 'TAJIKISTAN', 'DUSHANBE',
2010, 2013, 'CE', 'EPK'),
('Z667890123', 'Dylan', 'Lee', 'Male', 'dylanlee@gmail.com',
'dylan.lee@example.com', '+20 1234567890', 'EGYPT', 'ALEXANDRIA', 2011, 2014,
'SWE', '-'),
('G689012345', 'Gabriella', 'Sanchez', 'Female',
'gabriellasanchez@gmail.com', 'gabriella.sanchez@example.com', '+355
1234567890', 'ALBANIA', 'SHKODER', 2012, 2015, 'CE', 'EPK'),

```

('H690123456', 'Owen', 'Gomez', 'Male', 'owengomez@gmail.com',
'owen.gomez@example.com', '+90 1234567890', 'TURKEY', 'ISTANBUL', 2013, 2016,
'SWE', '-'),
('I601234567', 'Scarlett', 'Wright', 'Female', 'scarlettwright@gmail.com',
'scarlett.wright@example.com', '+39 1234567890', 'ITALY', 'ROME', 2014, 2017,
'ARCH', '-'),
('J712345678', 'Julian', 'Jenkins', 'Male', 'julianjenkins@gmail.com',
'julian.jenkins@example.com', '+992 1234567890', 'TAJIKISTAN', 'KHUJAND', 2015,
2018, 'ECE', '-'),
('K723456789', 'Penelope', 'Foster', 'Female', 'penelopefoster@gmail.com',
'penelope.foster@example.com', '+20 1234567890', 'EGYPT', 'CAIRO', 2016, 2019,
'BINF', 'EPK'),
('L34567890', 'Logan', 'Baker', 'Male', 'loganbaker@gmail.com',
'logan.baker@example.com', '+355 1234567890', 'ALBANIA', 'VLORE', 2017, 2020,
'PIR', '-'),
('M745678901', 'Ella', 'Brooks', 'Female', 'ellabrooks@gmail.com',
'ella.brooks@example.com', '+90 1234567890', 'TURKEY', 'ANKARA', 2018, 2021,
'BAF', '-'),
('N756789012', 'Mason', 'Hill', 'Male', 'masonhill@gmail.com',
'mason.hill@example.com', '+39 1234567890', 'ITALY', 'TIRANA', 2019, 2022,
'IML', '-'),
('O767890123', 'Aria', 'Gonzalez', 'Female', 'ariagonzalez@gmail.com',
'aria.gonzalez@example.com', '+992 1234567890', 'TAJIKISTAN', 'DUSHANBE', 2020,
2023, 'CE', '-'),
('P778901234', 'Jackson', 'Young', 'Male', 'jacksonyoung@gmail.com',
'jackson.young@example.com', '+20 1234567890', 'EGYPT', 'ALEXANDRIA', 2008,
2011, 'SWE', '-');

```

```

INSERT INTO Company (companyID, companyname, compemail, compphoneNO, sector,
category, locationID)

```

```

VALUES

```

```

('SO345', 'Softics', 'softics@email.com', '069 3322145', 'Private',
'Software', 'ALT'),
('DA543', 'DataPro', 'datapro@gmail.com', '069 4488775', 'Private', 'Data
Analytics', 'ALD'),
('TDT111', 'TDT Construction', 'tdt@email.com', '069 5896322', 'Private',
'Construction', 'ALV'),
('DT876', 'DELTA', 'delta@email.com', '068 2233658', 'Private', 'Sales',
'ALSH'),
('BR787', 'Brunes', 'brunes@email.com', '068 7744569', 'Private', 'Multi
Sector', 'ALT'),
('LO101', 'Law&Order', 'law&order@gmail.com', '066 2255698', 'Private',
'Law Firm', 'ALD'),
('EG454', 'Eglo', 'eglo@email.com', '066 3216547', 'Private', 'Sales',
'ALV'),
('MFA001', 'Ministry of Foreign Affairs',
'ministryofforeignaffairs@gmail.com', '067 5566984', 'public ', 'HR', 'ALSH'),
('TCH', 'Top Channel', 'ontopchannel@gmail.com', '067 4512369', 'Private*',
'Media', 'ALT'),
('SB349', 'Savings Bank', 'savingsbank@gmail.com', '068 2369854',
'Private', 'Banking', 'ALD');

```

```

INSERT INTO WorksAt (jobposition, department, status, alumniID, companyID)

```

```

VALUES

```

```

('Programmer', 'IT', 'Current', 'A223456789', 'SO345'),
('Programmer', 'IT', 'Past', 'B234567890', 'SO345'),
('Programmer', 'IT', 'Past', 'E267890123', 'DA543'),
('Programmer', 'IT', 'current', 'J212345678', 'SO345'),
('Programmer', 'IT', 'current', 'V212345678', 'SO345'),
('Programmer', 'IT', 'current', 'Y345678901', 'DA543'),
('Programmer', 'IT', 'current', 'Z667890123', 'SO345'),
('Programmer', 'IT', 'current', 'H690123456', 'SO345'),
('Programmer', 'IT', 'current', 'K723456789', 'DA543'),
('Technician', 'IT', 'past', 'D256789012', 'EG454'),

```

```
( 'Technician', 'IT', 'past', 'T501234567', 'EG454'),
( 'Technician', 'IT', 'past', 'J712345678', 'EG454'),
( 'Construction Engineer', 'Construction', 'Current', 'C245678901', 'DT876'),
( 'Architect', 'Architecture', 'Current', 'I201234567', 'BR787'),
( 'Architect', 'Architecture', 'Current', 'U201234567', 'TDT111'),
( 'Construction Engineer', 'Construction', 'Current', 'W223456789', 'DT876'),
( 'Construction Engineer', 'Construction', 'Current', 'C389012345', 'BR787'),
( 'Construction Engineer', 'Construction', 'Current', 'Y390123456', 'TDT111'),
( 'Architect', 'Architecture', 'Current', 'A312345678', 'DT876'),
( 'Construction Engineer', 'Construction', 'Current', 'G578901234', 'BR787'),
( 'Architect', 'Architecture', 'Current', 'Q578901234', 'TDT111'),
( 'Architect', 'Architecture', 'Current', 'S590123456', 'DT876'),
( 'Construction Engineer', 'Construction', 'Current', 'Y656789012', 'BR787'),
( 'Architect', 'Architecture', 'Past', 'G689012345', 'TDT111'),
( 'Architect', 'Architecture', 'Past', 'I601234567', 'DT876'),
( 'Banking Specialist', 'Finance', 'Current', 'G289012345', 'SB349'),
( 'Sales Specialist', 'Sales', 'Current', 'A367890123', 'BR787'),
( 'Sales Specialist', 'Sales', 'Current', 'E456789012', 'DT876'),
( 'Banking Specialist', 'Finance', 'Current', 'W534567890', 'BR787'),
( 'Sales Specialist', 'Sales', 'Current', 'M745678901', 'DT876'),
( 'Programmer', 'IT', 'Current', 'H290123456', 'SO345'),
( 'Sales Specialist', 'Sales', 'Current', 'B378901234', 'BR787'),
( 'Logistics', 'Logistics', 'Current', 'F467890123', 'DT876'),
( 'Programmer', 'IT', 'Current', 'X645678901', 'SO345'),
( 'Sales Specialist', 'Sales', 'Current', 'N756789012', 'BR787'),
( 'Journalist', 'Media', 'Current', 'F278901234', 'TCH'),
( 'Lawyer', 'Law', 'Current', 'Z356789012', 'LO101'),
( 'Specialist', 'Logistics', 'Current', 'D445678901', 'MFA001'),
( 'Specialist', 'Logistics', 'Current', 'V523456789', 'MFA001'),
( 'Specialist', 'Logistics', 'Current', 'L34567890', 'MFA001'),
( 'Programmer', 'IT', 'Current', 'P778901234', 'DA543');
```

```
INSERT INTO InternAt (internposition, departament, status, alumniID,
companyID)
```

```
VALUES
```

```
( 'Programmer', 'IT', 'Current', 'H89012345', 'SO345'),
( 'Programmer', 'IT', 'PAST', 'P778901234', 'DA543'),
( 'Programmer', 'IT', 'Current', 'R589012345', 'DA543'),
( 'Programmer', 'IT', 'Current', 'Z301234567', 'SO345'),
( 'Construction Engineering', 'Construction', 'Current', 'O767890123',
'DT876'),
( 'Programmer', 'IT', 'past', 'K723456789', 'DA543');
```

3.3 MENAGERIAL QUERIES

1. Retrieve the list of alumni who studied in a specific program.

```
SELECT * FROM Alumni WHERE programID = 'SWE';
```

alu...	firstna...	lastname	gender	primar...	secon...	contac...	country	city	enrollY...	gradYear	progra...	uniID
B23456...	Jane	Smith	Female	janesmit...	jane.smi...	+90 123...	TURKEY	ISTANB...	2012	2015	SWE	UAMD
J21234...	Isabella	Clark	Female	isabella...	isabella...	+20 123...	EGYPT	ALEXA...	2020	2023	SWE	UT
V21234...	Mia	Lee	Female	miale...	mia.lee...	+90 123...	TURKEY	ISTANB...	2009	2012	SWE	UAMD
D39012...	Olivia	Cook	Female	oliviaco...	olivia.co...	+20 123...	EGYPT	ALEXA...	2017	2020	SWE	UT
Z30123...	Henry	Adams	Male	henryad...	henry.a...	+90 123...	TURKEY	ISTANB...	2019	2022	SWE	UAMD
H89012...	Wyatt	Ramirez	Male	wyattra...	wyatt.ra...	+20 123...	EGYPT	ALEXA...	2014	2017	SWE	-
R58901...	Nathan	Reed	Male	nathanr...	nathan.r...	+90 123...	TURKEY	ISTANB...	2016	2019	SWE	UAMD
Z66789...	Dylan	Lee	Male	dylanlee...	dylan.le...	+20 123...	EGYPT	ALEXA...	2011	2014	SWE	-
H69012...	Owen	Gomez	Male	owengo...	owen.go...	+90 123...	TURKEY	ISTANB...	2013	2016	SWE	-
P77890...	Jackson	Young	Male	jackson...	jackson...	+20 123...	EGYPT	ALEXA...	2008	2011	SWE	-

2. Retrieve name and surname of alumni working at a specific company.

```
SELECT firstname, lastname  
FROM Alumni  
JOIN WorksAt ON Alumni.alumniID = WorksAt.alumniID  
JOIN Company ON WorksAt.companyID = Company.companyID  
WHERE Company.companyname = 'DataPro';
```

firstname	lastname
Daniel	Brown
Penelope	Foster
Jackson	Young
Liam	Lopez

3. Retrieve name and surname of alumni doing an internship at a specific company.

```
SELECT firstname, lastname, primaryemail
FROM Alumni
JOIN InternAt ON Alumni.alumniID = InternAt.alumniID
JOIN Company ON InternAt.companyID = Company.companyID
WHERE Company.companyname = 'DataPro';
```

firstname	lastname	primaryemail
Penelope	Foster	penelopefoster@gmail.com
Jackson	Young	jacksonyoung@gmail.com
Nathan	Reed	nathanreed@gmail.com

4.Retrieve the name of the companies of a specific category.

```
SELECT companyid, companyname, compemail FROM Company
WHERE category = 'Sales';
```

companyID	companyname	compemail
DT876	DELTA	delta@email.com
EG454	Eglo	eglo@email.com

5.Retrieve the company ID, company name and email of private companies.

```
SELECT companyid, companyname, compemail FROM Company
WHERE sector = 'Private';
```

companyID	companyname	compemail
SO345	Softics	softics@email.com
DA543	DataPro	datapro@gmail.com
TDT111	TDT Construction	tdt@email.com
DT876	DELTA	delta@email.com
BR787	Brunes	brunes@email.com
LO101	Law&Order	law&order@gmail.com
EG454	Eglo	eglo@email.com
SB349	Savings Bank	savingsbank@gmail.com

6. Retrieve the company ID, company name and email of public companies.

```
SELECT companyid, companyname, compemail FROM Company  
WHERE sector = 'public ';
```

companyID	companyname	compemail
MFA001	Ministry of Foreign Affairs	ministryofforeignaffairs@gmail.com

7. Retrieve the list of alumni who completed a specific program and are currently employed

```
SELECT firstname, lastname, jobposition  
FROM Alumni  
JOIN WorksAt ON Alumni.alumniID = WorksAt.alumniID  
WHERE Alumni.programID = 'SWE' AND WorksAt.status = 'current';
```

firstname	lastname	jobposition
Isabella	Clark	Programmer
Mia	Lee	Programmer
Dylan	Lee	Programmer
Owen	Gomez	Programmer

8. Retrieve the list of alumni who completed a specific program and are currently employed in a specific category

```
SELECT firstname, lastname, jobposition FROM Alumni  
JOIN WorksAt ON Alumni.alumniID = WorksAt.alumniID  
JOIN Company ON WorksAt.companyID = Company.companyID  
WHERE Alumni.programID = 'SWE' AND WorksAt.departament = 'IT';
```

firstname	lastname	jobposition
Jane	Smith	Programmer
Isabella	Clark	Programmer
Mia	Lee	Programmer
Dylan	Lee	Programmer
Owen	Gomez	Programmer
Jackson	Young	Programmer

9. Retrieve the list of alumni who graduated within a specific year range

```
SELECT alumniid, firstname, lastname
FROM Alumni
WHERE gradYear BETWEEN 2015 AND 2017;
```

alumniID	firstname	lastname
B234567890	Jane	Smith
C245678901	Michael	Johnson
D256789012	Emily	Williams
Y345678901	Liam	Lopez
Z356789012	Charlotte	Hernandez
A367890123	Noah	Gonzalez
F467890123	Lucas	Hayes
G578901234	Chloe	Scott
H89012345	Wyatt	Ramirez
G689012345	Gabriella	Sanchez
H690123456	Owen	Gomez

10. Retrieve the count of alumni working in each company

```
SELECT c.companyname, COUNT(*) AS alumni_count
FROM Alumni a
JOIN WorksAt w ON a.alumniID = w.alumniID
JOIN Company c ON w.companyID = c.companyID
GROUP BY c.companyname;
```

companyname	alumni_count
Brunes	8
DELTA	8
DataPro	4
Eglo	3
Law&Order	1
Ministry of Foreign Affairs	3
Savings Bank	1
Softics	8
TDT Construction	4
Top Channel	1

11. Retrieve the count of alumni who have conducted an internship in each company.

```
SELECT c.companyname, COUNT(*) AS alumni_count
FROM Alumni a
JOIN InternAt i ON a.alumniID = i.alumniID
JOIN Company c ON i.companyID = c.companyID
GROUP BY c.companyname;
```

companyname	alumni_count
DELTA	1
DataPro	3
Softics	2

12. Retrieve the list of study programs along with the count of alumni who have completed each program

```
SELECT p.programName, COUNT(a.alumniID) AS alumni_count
FROM Program p
LEFT JOIN Alumni a ON p.programID = a.programID
GROUP BY p.programName;
```

programName	alumni_count
Architecture	5
Bioinformatics	5
Business Administration and Finance	5
Civil Engineering	9
Computer Engineering	1
Electrical and Computer Engineering	5
History	0
International Relations and European Studies	5
Political Science and International Relations	5
Software Engineering	10

13. Retrieve the list of study programs along with the count of male and female alumni in each program

```
SELECT p.programName,
       COUNT(CASE WHEN a.gender = 'Male' THEN 1 END) AS male_count,
       COUNT(CASE WHEN a.gender = 'Female' THEN 1 END) AS female_count
FROM Program p
LEFT JOIN Alumni a ON p.programID = a.programID
GROUP BY p.programName;
```

programName	male_count	female_count
Architecture	2	3
Bioinformatics	2	3
Business Administration and Finance	2	3
Civil Engineering	3	6
Computer Engineering	1	0
Electrical and Computer Engineering	3	2
History	0	0
International Relations and European Studies	3	2
Political Science and International Relations	3	2
Software Engineering	6	4

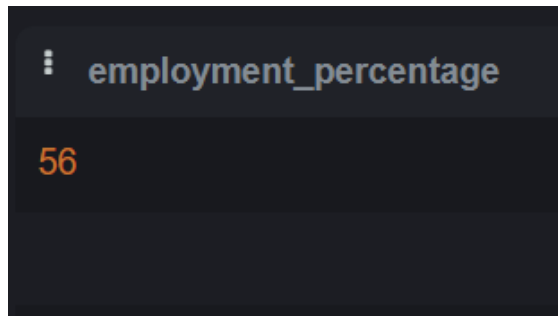
14. Retrieve number of employees employed in a certain company

```
SELECT COUNT(*) AS num_employees  
From WorksAt  
WHERE companyid = 'S0345';
```

num_employees
8

15. Calculate the overall percentage of alumni who are currently employed

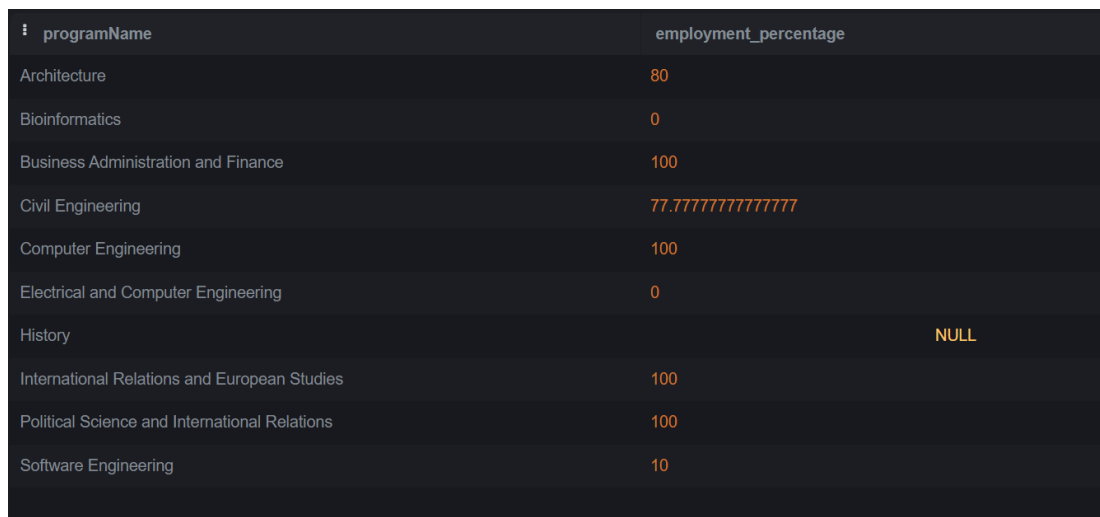
```
SELECT (COUNT(CASE WHEN w.status = 'Current' THEN a.alumniID END) * 100.0)
/ COUNT(a.alumniID) AS employment_percentage
FROM Alumni a
LEFT JOIN WorksAt w ON a.alumniID = w.alumniID;
```



employment_percentage
56

16. Calculate the percentage of employed alumni for each program

```
SELECT p.programName,
       (COUNT(CASE WHEN w.status = 'Current' THEN a.alumniID END) * 100.0)
/ COUNT(a.alumniID) AS employment_percentage
FROM Program p
LEFT JOIN Alumni a ON p.programID = a.programID
LEFT JOIN WorksAt w ON a.alumniID = w.alumniID
GROUP BY p.programName;
```



programName	employment_percentage
Architecture	80
Bioinformatics	0
Business Administration and Finance	100
Civil Engineering	77.77777777777777
Computer Engineering	100
Electrical and Computer Engineering	0
History	NULL
International Relations and European Studies	100
Political Science and International Relations	100
Software Engineering	10

17. Calculate the work experiences of every alumnus. Show alumni ID, name and surname.

```
SELECT a.alumniid, a.firstname, a.lastname,  
       COUNT (w.alumniid) AS work_experience  
FROM Alumni a  
LEFT JOIN WorksAt w ON a.alumniid = w.alumniid  
GROUP BY a.alumniid, a.firstname, a.lastname;
```

alumniID	firstname	lastname	work_experience
A223456789	John	Doe	1
A312345678	Sofia	Campbell	1
A367890123	Noah	Gonzalez	1
B234567890	Jane	Smith	1
B378901234	Emma	Perez	1
B423456789	Benjamin	Hill	0
C245678901	Michael	Johnson	1
C389012345	Jacob	Robinson	1
C434567890	Avery	Russell	0
D256789012	Emily	Williams	1
D390123456	Olivia	Cook	0

18. Retrieve alumni ID, name, surname of alumni who are conducting or have conducted further studies and show the university name.

```
SELECT a.alumniid, a.firstname, a.lastname, f.uniname  
FROM Alumni a  
INNER JOIN FurtherStudies f on a.uniid = f.uniid;
```

alumniID	firstname	lastname	uniName
A223456789	John	Doe	Epoka University
B234567890	Jane	Smith	University of Aleksander Moisiu Durrës
C245678901	Michael	Johnson	University of Tirana
D256789012	Emily	Williams	Epoka University
E267890123	Daniel	Brown	University of Aleksander Moisiu Durrës
F278901234	Olivia	Taylor	Epoka University
G289012345	Matthew	Miller	University of Aleksander Moisiu Durrës
H290123456	Sophia	Anderson	University of Tirana
I201234567	James	Wilson	Epoka University
J212345678	Isabella	Clark	University of Tirana
U201234567	Alexander	Young	Epoka University

19.Retrieve alumni ID, name, surname of all alumni who are in a specific city (Vlore as an example).

```
SELECT alumniid, firstname, lastname
FROM Alumni
WHERE city = 'VLORE';
```

alumniID	firstname	lastname
H290123456	Sophia	Anderson
Z356789012	Charlotte	Hernandez
D445678901	Elijah	Griffin
V523456789	Ethan	Reynolds
L34567890	Logan	Baker

20.Retrieve company ID, name, email of all companies which are in a specific city (Tirane)

```
SELECT c.companyid, c.companyname, c.compemail
FROM Company c
JOIN Location l ON c.locationid = l.locationid
WHERE l.locCity = 'Tirana';
```

companyID	companyname	compemail
SO345	Softics	softics@email.com
BR787	Brunes	brunes@email.com
TCH	Top Channel	ontopchannel@gmail.com

21.Retrieve alumni ID, name, surname and email conducting further studies in a specific location (Tirane).

```
SELECT a.alumniid, a.firstname, a.lastname, a.primaryemail
FROM Alumni a
WHERE a.uniID in ( SELECT fs.uniid
                   FROM FurtherStudies fs
                   JOIN Location l on fs.locationid = l.locationid
                   where l.loccity = 'Tirana');
```

alumniID	firstname	lastname	primaryemail
A223456789	John	Doe	johndoe@gmail.com
D256789012	Emily	Williams	emilywilliams@gmail.com
F278901234	Olivia	Taylor	oliviatailor@gmail.com
I201234567	James	Wilson	jameswilson@gmail.com
U201234567	Alexander	Young	alexanderyoung@gmail.com
X334567890	Ava	Martinez	avamartinez@gmail.com
Z356789012	Charlotte	Hernandez	charlottehernandez@gmail.com
B378901234	Emma	Perez	emmaperez@gmail.com
C389012345	Jacob	Robinson	jacobrobinson@gmail.com
Y390123456	Amelia	Gray	ameliagray@gmail.com
C434567890	Avery	Russell	averyrussell@gmail.com

22.Calculate the percentage of alumnus working in a category outside their field

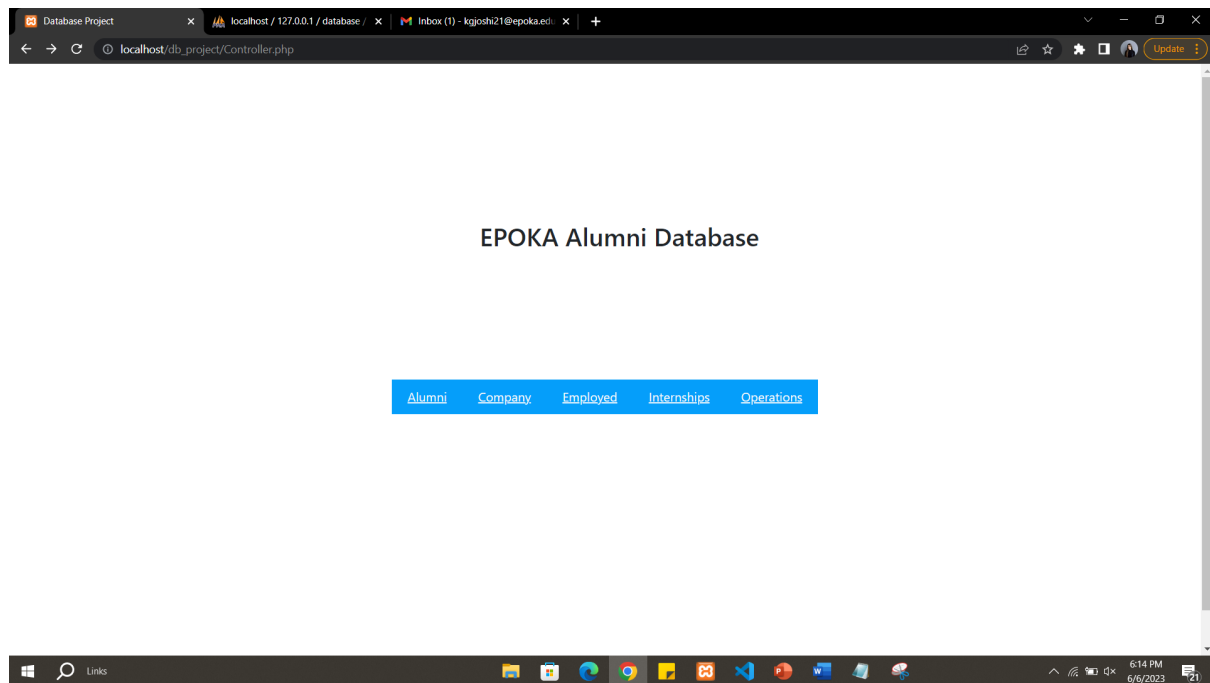
```
SELECT (Count(DISTINCT a.alumniid))/(SELECT COUNT(*)
                                      FROM Alumni))*100 as
percentage_outof_category
FROM Alumni a
left JOIN WorksAt w on a.alumniID = w.alumniid
LEFT join Program p on a.programid = p.programID
WHERE p.programid is NULL or p.programid != a.programid;
```

percentage_outof_category
0

4. FrontEnd Visualization (Bonus Part)

We have done a simple visualisation of our Alumni Database. The project has been worked using php, phpmyadmin,html and css.

Alumni and Company table will be displayed associated with the operation buttons such as add, update and delete. The relationship between these two companies expressed as internships and employment has also been displayed. We have also made a visualisation for some of the queries taking in consideration users specifications.



Add

First ID	Last Name	Gender	Email	Secondary Email	Contact Number	Country	City	Program ID	University ID	Operations
4	John Doe	Male	johndoe@gmail.com	john.doe@example.com	+355 1234567890	ALBANIA	TIRANA	CEN	EPK	UpdateDelete
5	Jane Smith	Female	janesmith@gmail.com	jane.smith@example.com	+90 1234567890	TURKEY	ISTANBUL	SWE	UAMD	UpdateDelete
6	Michael Johnson	Male	michaeljohnson@gmail.com	michaeljohnson@example.com	+39 1234567890	ITALY	ROME	ARCH	UT	UpdateDelete
7	Emily Williams	Female	emilywilliams@gmail.com	emily.williams@example.com	+992 1234567890	TAJIKISTAN	DUSHANBE	ECE	EPK	UpdateDelete
8	Daniel Brown	Male	danielbrown@gmail.com	daniel.brown@example.com	+20 1234567890	EGYPT	CAIRO	BINF	UAMD	UpdateDelete
9	Olivia Taylor	Female	oliviatailor@gmail.com	olivia.taylor@example.com	+355 1234567890	ALBANIA	SHKODER	PIR	EPK	UpdateDelete

Last Name

Doe

Gender:

Female

Primary Email

jennydoe@email.com

Secondary Email

Enter your secondary email

Contact Number

+355691212765

Country

Albania

City

Tirana

Program ID

CEN

University ID

EPK

Submit

Add

ID	Company Name	Email	Contact Number	Sector	Category	Location ID	Operations
2	Softics	softics@email.com	693322145	Public	Programming	ALT	Update Delete
3	DataPro	datapro@gmail.com	0694488775	private	Programming	ALD	Update Delete
4	TDT Construction	tdt@email.com	069 5896322	Private	Construction	ALV	Update Delete
5	DELTA	delta@email.com	068 2233658	Private	Sales	ALSH	Update Delete
6	Brunes	brunes@email.com	068 7744569	Private	Multi Sector	ALT	Update Delete
7	Law&Order	law&order@gmail.com	066 2255698	Private	Law Firm	ALD	Update Delete
8	Eglo	eglo@email.com	066 3216547	Private	Sales	ALV	Update Delete
9	Ministry of Foreign Affairs	ministryofforeignaffairs@gmail.com	067 5566984	public	HR	ALSH	Update Delete
10	Top Channel	ontopchannel@gmail.com	067 4512369	Private*	Media	ALT	Update Delete
11	Savings Bank	savingsbank@gmail.com	068 2369854	Private	Banking	ALD	Update Delete

Company Name
Top Channel

Email
ontopchannel@gmail.com

Contact Number
0672123452

Sector: Private

Category
Media

Location ID
ALT

Update

Add

Position	Department	Status	Alumni ID	Company ID	Operations
Programmer	IT	Current	4	2	Update
Programmer	IT	Past	4	3	Update
Programmer	IT	Past	5	2	Update
Programmer	IT	Current	8	3	Update

Add

Position	Department	Status	Alumni ID	Company ID	Operations
Programmer	IT	Current	4	2	Update
Programmer	IT	PAST	4	3	Update
Programmer	IT	Current	5	3	Update
Programmer	IT	Current	8	3	Update
Construction Engineering	Construction	Current	12	5	Update
Programmer	IT	Current	13	2	Update
Programmer	IT	past	15	3	Update



Retrieve the list of alumni who graduated within a specific year range

Submit

Retrieve the name of the companies of a specific category

Submit

Retrieve alumni info of all alumni who are in a specific city

Submit

Retrieve all employees working in a certain company

Submit



ID	First Name	Last Name	Gender	Email	Secondary Email	Contact Number	Country	City	Program ID	University ID
4	John	Doe	Male	johndoe@gmail.com	john.doe@example.com	+355 1234567890	ALBANIA	TIRANA	CEN	EPK
14	Alexander	Young	Male	alexanderyoung@gmail.com	alexander.young@example.com	+355 1234567890	ALBANIA	DURRES	CE	EPK
15	Mia	Lee	Female	mialee@gmail.com	mia.lee@example.com	+90 1234567890	TURKEY	ISTANBUL	SWE	UAMD





Retrieve the list of alumni who graduated within a specific year range

Submit

Retrieve the name of the companies of a specific category

Submit

Retrieve alumni info of all alumni who are in a specific city

Submit

Retrieve all employees working in a certain company

Submit



ID	Company Name	Email	Contact Number	Sector	Category	Location ID
2	Softics	softics@email.com	693322145	Public	Programming	ALT
3	DataPro	datapro@gmail.com	0694488775	private	Programming	ALD



Retrieve the list of alumni who graduated within a specific year range

2011

2013

Submit

Retrieve the name of the companies of a specific category

Programming

Submit

Retrieve alumni info of all alumni who are in a specific city

Tirane

Submit

Retrieve all employees working in a certain company

Enter company ID

Submit

ID	First Name	Last Name	Gender	Email	Secondary Email	Contact Number	Country	City	Program ID	University ID
4	John	Doe	Male	johndoe@gmail.com	john.doe@example.com	+355 1234567890	ALBANIA	TIRANA	CEN	EPK
16	Jenny	Doe	female	jennydoe@email.com		+355691212765	Albania	Tirana	CEN	EPK