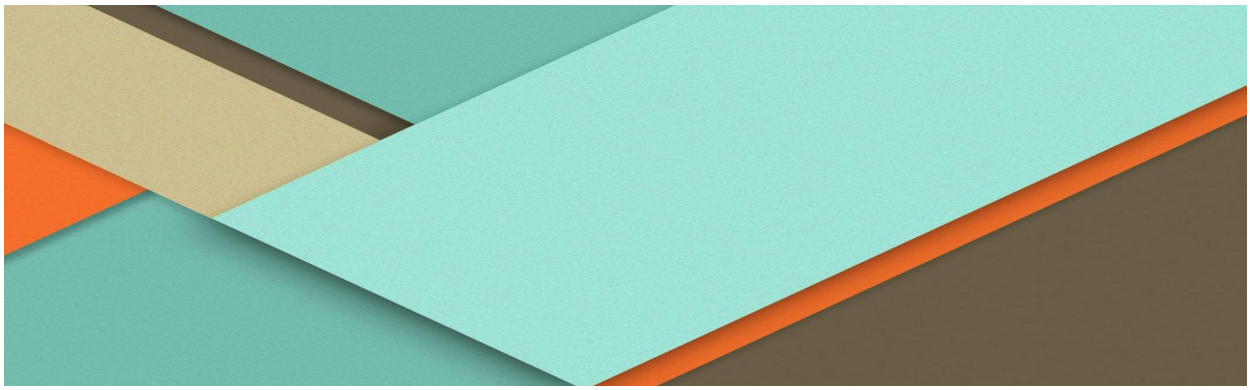
CS-360



Project Report

29.01.2023

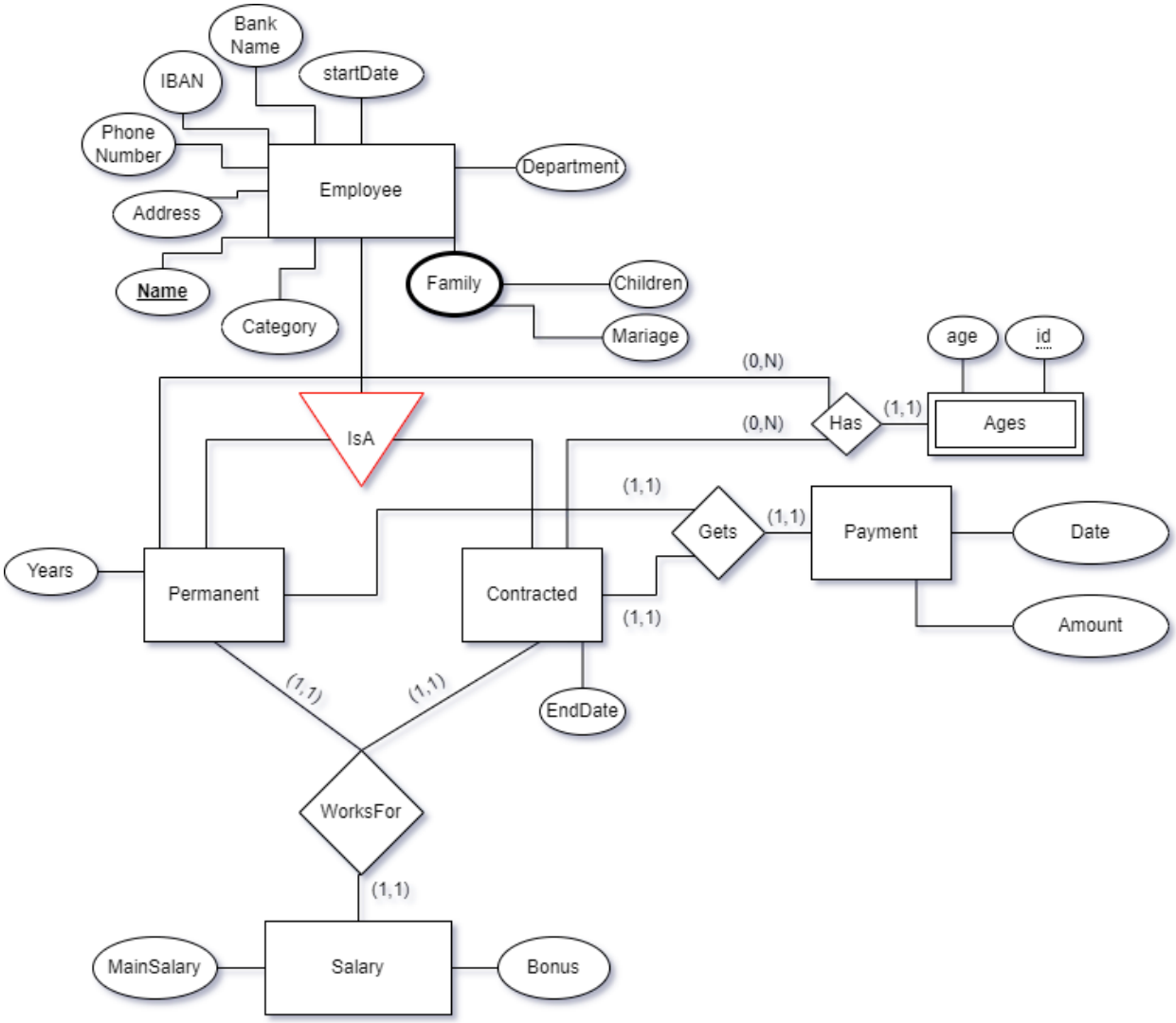
Isidoros Chatzichrisos

Theodoros Pontzouktzidis

Alexandros Markodimitrakis

Content.

ER Diagram.


Bank Name
IBAN
startDate
Phone Number
Employee
Department
Address
Name
Category
Family
Children
Mariage
age
id
(0,N)
IsA
(0,N)
Has
(1,1)
Ages
(1,1)
Gets
(1,1)
Payment
Date
Years
Permanent
Contracted
(1,1)
Amount
(1,1)
(1,1)
EndDate
WorksFor
(1,1)
MainSalary
Salary
Bonus

Entities (Names,Types,Keys,Cardinalities).

Permanent:

Name (VARCHAR)

Address (VARCHAR)

Phone Number (VARCHAR)

IBAN (VARCHAR)

BankName (VARCHAR)

StartDate (DATE)

Department (VARCHAR)

Children (INTEGER)

Mariage (BIT)

Category (BIT)

Years(INTEGER)

Contracted:

Name (VARCHAR)

Address (VARCHAR)

Phone Number (VARCHAR)

IBAN (VARCHAR)

BankName (VARCHAR)

StartDate (DATE)

Department (VARCHAR)

Children (INTEGER)

Mariage (BIT)

Category (BIT)

EndDate(DATE)

Ages:

Name (VARCHAR)

Age (INTEGER)

Salary:

Name (VARCHAR)

Bonus (INTEGER)

MainSalary (INTEGER)

Payment:

Name (VARCHAR)

Date (DATE)

Amount (INTEGER)

ENTITY	PRIMARY KEY
Permanent	Name
Contracted	Name
Ages	Name (foreign key from permanent or contracted)
Salary	Name (foreign key from permanent or contracted)
Payment	Name (foreign key from permanent or contracted)

* We divided Employee Entity in 2 Entities **Permanent** and **Contracted** so we can lower the cost of searching in the DB.

Entity(Cardinality)	Relation	(Cardinality)Entity
Permanent(1,1)	Gets	(1,1)Payment
Permanent(1,1)	WorksFor	(1,1)Salary
Contracted(1,1)	Gets	(1,1)Payment
Contracted(1,1)	WorksFor	(1,1)Salary
Employee()	Has	(1,1)Ages

* We use Min,Max orientation. Also every Cardinality is set to 1,1 because each employee name can exist to one payment and salary and Each payment or salary relates to only one employee.

Relational Model.

Permanent

Name	Address	Phone Number	IBAN	Bank Name	Start Date	Depart ment	Children	Mariage	Cate gory	Years

Contracted

Name	Address	Phone Number	IBAN	Bank Name	Start Date	Depart ment	Children	Mariage	Cate gory	EndDate

Ages

Name	Age

Salary

Name	Bonus	MainSalary

Payment

Name	Date	Amount

**partial key*

Keys & Functional Dependencies.

Permanent:

Name **->** Address,Phone,Number,IBAN,BankName,StartDate,SalaryID,Department, Children,Mariage,Category,Years

Contracted:

Name **->** Address,Phone,Number,IBAN,BankName,StartDate,SalaryID,Department, Children,Mariage,Category,EndDate

Ages:

Name,id **->** Age

Salary:

Name **->** Bonus,MainSalary

Payment:

Name **->** Date,Amount

3NF Without Data Loss.

For a Relation to be in 1NF it needs to:

- Attributes have to be atomic. ✅

  In our Model every Relation is in 1NF.

For a Relation to be in 2NF it needs to:

- Relation needs to be in 1NF. ✅
- Relation's Functional Dependencies can't be in the form: non-primary(key) **->** primary key. ✅

  In our Model every Relation is in 2NF.

For a Relation to be in 3NF it needs to:

- Relation needs to be in 2NF. ✅
- No Transitive Dependencies in Relation. ✅
- Relation's Functional Dependencies can't be in the form: non-primary(key) **->** non-primary(key). ✅

  In our Model every Relation is in 3NF.

SQL Create-Insert Tables.

Create:

```
CREATE DATABASE IF NOT EXISTS salary_db;
CREATE TABLE permanent(
    name VARCHAR(255) PRIMARY KEY,
    address VARCHAR(255),
    phone_number VARCHAR(255),
    iban VARCHAR(255),
    bank_name VARCHAR(255),
    start_date DATE,
    department VARCHAR(255),
    children INTEGER,
    married BIT,
    category BIT,
    years INTEGER);
CREATE TABLE contracted(
        name VARCHAR(255) PRIMARY KEY,
        address VARCHAR(255),
        phone_number VARCHAR(255),
        iban VARCHAR(255),
        bank_name VARCHAR(255),
        start_date DATE,
        department VARCHAR(255),
        children INTEGER,
        married BIT,
        category BIT,
```

```
    end_date DATE);
CREATE TABLE ages(
    name VARCHAR(255) PRIMARY KEY,
    age INTEGER,
    CONSTRAINT FK_1 FOREIGN KEY (name) REFERENCES permanent(name),
    CONSTRAINT FK_2 FOREIGN KEY (name) REFERENCES contracted(name));
CREATE TABLE salary(
  name VARCHAR(255),
  main_salary DOUBLE,
  bonus DOUBLE,
  CONSTRAINT FK_3 FOREIGN KEY (name) REFERENCES permanent(name),
  CONSTRAINT FK_4 FOREIGN KEY (name) REFERENCES contracted(name));
CREATE TABLE payment (
   name VARCHAR(255),
   date DATE,
   amount DOUBLE,
    CONSTRAINT FK_1 FOREIGN KEY (name) REFERENCES permanent(name),
    CONSTRAINT FK_2 FOREIGN KEY (name) REFERENCES contracted(name));
```

Insert Examples:

```
INSERT INTO permanent(name, address, phone_number, iban, bank_name, start_date, department, children, married, category, years)SELECT 'Alexandros Markodimitrakis','Fourtounatou 6','6981073112','GR2701121149381872825494513','Piraeus Bank','2022-05-23','Programming',4,1,1,12 WHERE NOT EXISTS (SELECT 1 FROM contracted WHERE name = 'Alexandros Markodimitrakis');

INSERT INTO permanent(name, address, phone_number, iban, bank_name, start_date, department, children, married, category, years) SELECT 'Konstantinos Vlachos','Ioannou Papakyriakou 2','6987654343','GR3908174938161923875693O4','Alpha Bank','2022-12-01','Support',0,1,0,10 WHERE NOT EXISTS (SELECT 1 FROM contracted WHERE name = 'Konstantinos Vlachos');
```

INSERT INTO permanent(name, address, phone_number, iban, bank_name, start_date, department, children, married, category, years) **SELECT** 'Eleni Papadopoulou','Ermou 22','6980987654','GR3901121119381872825444','Eurobank','2022-06-15','Marketing',0,0,1, 8 **WHERE NOT EXISTS** (**SELECT** 1 **FROM** contracted **WHERE** name = 'Eleni Papadopoulou');

INSERT INTO permanent(name, address, phone_number, iban, bank_name, start_date, department, children, married, category, years) **SELECT** 'Andreas Karatzas','Vassileos Pavlou 12','6987654321','GR3201121149381872825411','National Bank of Greece','2022-09-01','Human Resources',0,1,1,15 **WHERE NOT EXISTS** (**SELECT** 1 **FROM** contracted **WHERE** name = 'Andreas Karatzas');

INSERT INTO permanent(name, address, phone_number, iban, bank_name, start_date, department, children, married, category, years) **SELECT** 'Maria Tsakiri','Aristotelous 8','6981234567','GR3501121149381872825488','Proton Bank','2022-07-01','Sales',0,0,0,7 **WHERE NOT EXISTS** (**SELECT** 1 **FROM** contracted **WHERE** name = 'Maria Tsakiri');

INSERT INTO contracted(name, address, phone_number, iban, bank_name, start_date,department, children, married, category, end_date) **SELECT** 'Theodoros Pontzouktzidis','Theokritou 18','6906045239','GR4701121149381872825467513','Pancreta Bank','2001-05-23','Mathematics',2,1,0,'2001-05-23' **WHERE NOT EXISTS** (**SELECT** 1 **FROM** permanent **WHERE** name = 'Theodoros Pontzouktzidis');

INSERT INTO contracted(name, address, phone_number, iban, bank_name, start_date,department, children, married, category, end_date) **SELECT** 'Eirini Kostopoulou','Odos Athinas 45','6907451298','GR2201121149381872825456','Piraeus Bank','2002-06-15','Finance',0,0,1,'2002-06-15' **WHERE NOT EXISTS** (**SELECT** 1 **FROM** permanent **WHERE** name = 'Eirini Kostopoulou');

INSERT INTO contracted(name, address, phone_number, iban, bank_name, start_date,department, children, married, category, end_date)**SELECT** 'Giorgos Markakis','Iroon Politechniou 12','6905678923','GR3101121149381872825498 7','Alpha Bank','2003-01-01','Engineering',0,1,0,'2003-01-01' **WHERE NOT EXISTS** (**SELECT** 1 **FROM** permanent **WHERE** name = 'Giorgos Markakis');

INSERT INTO contracted(name, address, phone_number, iban, bank_name, start_date,department, children, married, category, end_date)**SELECT** 'Katerina Samara','Aristotelous 8','6908236547','GR2501121149381872825412 3','National Bank of Greece','2001-07-01','Research',0,0,1,'2001-07-01' **WHERE NOT EXISTS** (**SELECT** 1 **FROM** permanent **WHERE** name = 'Katerina Samara');

INSERT INTO contracted(name, address, phone_number, iban, bank_name, start_date,department, children, married, category, end_date)**SELECT** 'Dimitrios Alexakis','Kifisias

12','6905678901','GR3601121149381872825400 0','Eurobank','2002-03-01','Design',0,1,0,'20 02-03-01' **WHERE NOT EXISTS** (**SELECT** 1 **FROM** permanent **WHERE** name = 'Dimitrios Alexakis');

INSERT INTO ages(name, age)**VALUES**('Alexandros Markodimitrakis',23);

INSERT INTO ages(name, age)**VALUES**('Alexandros Markodimitrakis',18);

INSERT INTO ages(name, age)**VALUES**('Alexandros Markodimitrakis',16);

INSERT INTO ages(name, age)**VALUES**('Alexandros Markodimitrakis',12);

INSERT INTO ages(name, age)**VALUES**('Theodoros Pontzouktzidis',25);

INSERT INTO ages(name, age)**VALUES**('Theodoros Pontzouktzidis',17);

INSERT INTO salary(name,main_salary,bonus) **VALUES**('Alexandros Markodimitrakis',280.0,15.0);

INSERT INTO salary(name,main_salary,bonus) **VALUES**('Theodoros Pontzouktzidis',30000.0,4000.0);

INSERT INTO salary(name,main_salary,bonus) **VALUES**('Eirini Kostopoulou',23000.0,2000.0);

INSERT INTO salary(name,main_salary,bonus) **VALUES**('Giorgos Markakis',32000.0,3500.0);

INSERT INTO salary(name,main_salary,bonus) **VALUES**('Katerina Samara',25000.0,1500.0);

INSERT INTO payment(name,date,amount) **VALUES**('Alexandros Markodimitrakis','2022-05-31',295.0);

INSERT INTO payment(name,date,amount) **VALUES**('Theodoros Pontzouktzidis','2001-05-31',34000.0);

INSERT INTO payment(name,date,amount) **VALUES**('Eirini Kostopoulou','2022-05-31',25300.0);

INSERT INTO payment(name,date,amount) **VALUES**('Giorgos Markakis','2022-05-31',34500.0);

INSERT INTO payment(name,date,amount) **VALUES**('Katerina Samara','2022-05-31',26500.0);

SQL Queries.

Query 1: Payroll by category of staff

SQL Query 1:

SELECT * **FROM** payment **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category =0);

Foreach payment (**IN** JAVA)

SELECT * **FROM** salary **WHERE** name = res);

Print results of above SQL query

SELECT * **FROM** payment **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category =1);

Foreach payment (IN JAVA)

SELECT * **FROM** salary **WHERE** name = res);

Print results of above SQL query

SELECT * **FROM** payment **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category =0);

Foreach payment (**IN** JAVA)

SELECT * **FROM** salary **WHERE** name = res);

Print results of above SQL query

SELECT * **FROM** payment **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category =1);

Foreach payment (**IN** JAVA)

SELECT * **FROM** salary **WHERE** name = res);

Print results of above SQL query

(res is each name from each entry.)

Query 2: Maximum, minimum and average salary by category of staff

SQL Queries 2:

Minimum:

SELECT MIN(main_salary+bonus) **FROM** salary **WHERE** name IN (SELECT name FROM permanent WHERE category=0);

SELECT MIN(main_salary+bonus) **FROM** salary **WHERE** name IN (SELECT name FROM permanent WHERE category=1);

SELECT MIN(main_salary+bonus) **FROM** salary **WHERE** name IN (SELECT name FROM contracted WHERE category=0);

SELECT MIN(main_salary+bonus) **FROM** salary **WHERE** name IN (SELECT name FROM contracted WHERE category=1);

Maximum:

SELECT MAX(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category=0);

SELECT MAX(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category=1);

SELECT MAX(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category=0);

SELECT MAX(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category=1);

Average:

SELECT AVG(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category=0);

SELECT AVG(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category=1);

SELECT AVG(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category=0);

SELECT AVG(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category=1);

Query 3: Data and payroll of a specific staff member

SQL Queries 3:

SELECT * **FROM** permanent **WHERE** name = res;

If above query doesnt select anything we proceed to do a select in contracted table

SELECT * **FROM** contracted **WHERE** name = res;

SELECT * **FROM** salary **WHERE** name = res;

Query 4: Total payroll by category of staff

SQL Queries 4:

SELECT SUM(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category=0);

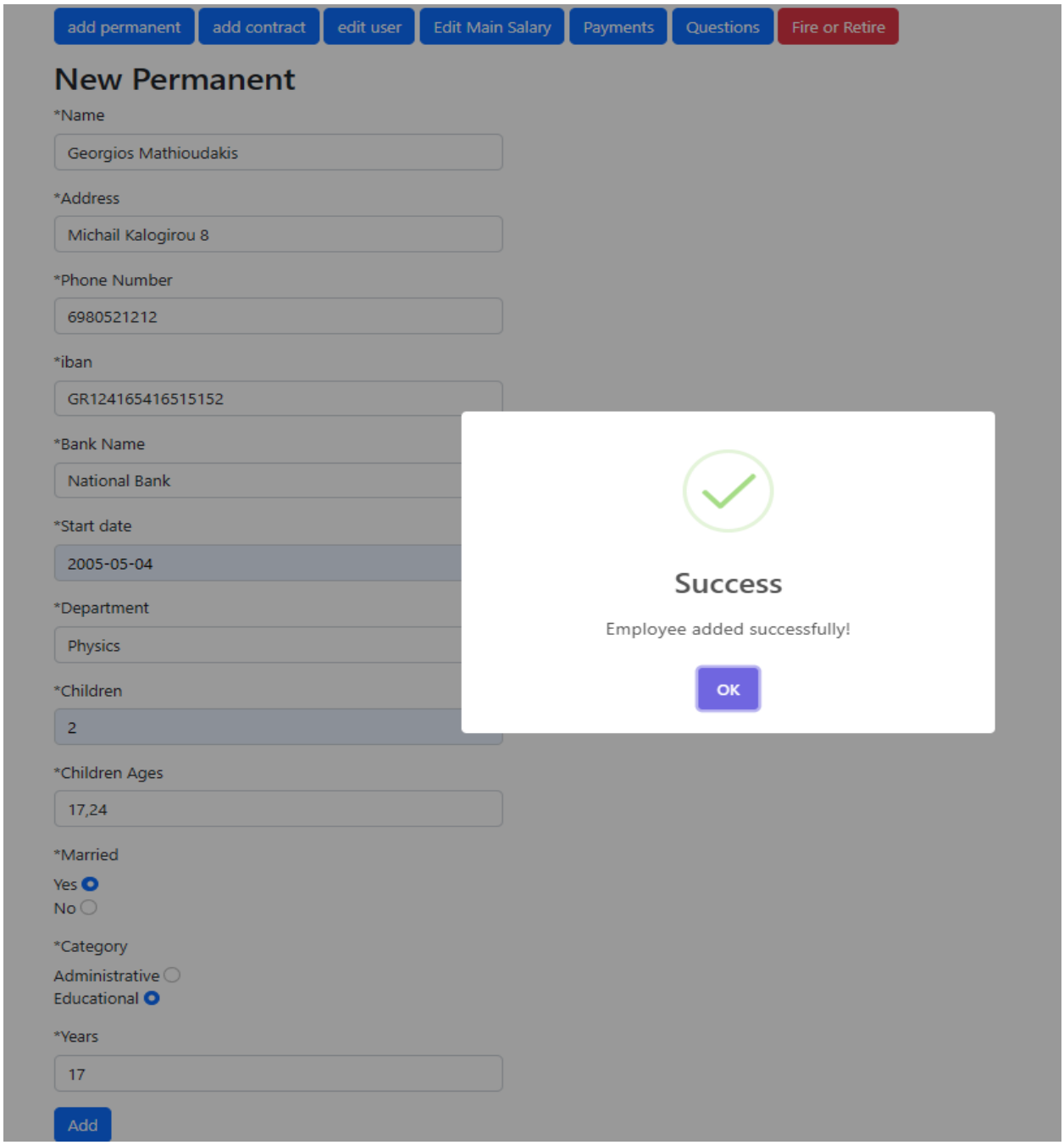
SELECT SUM(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** permanent **WHERE** category=1);

SELECT SUM(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category=0);

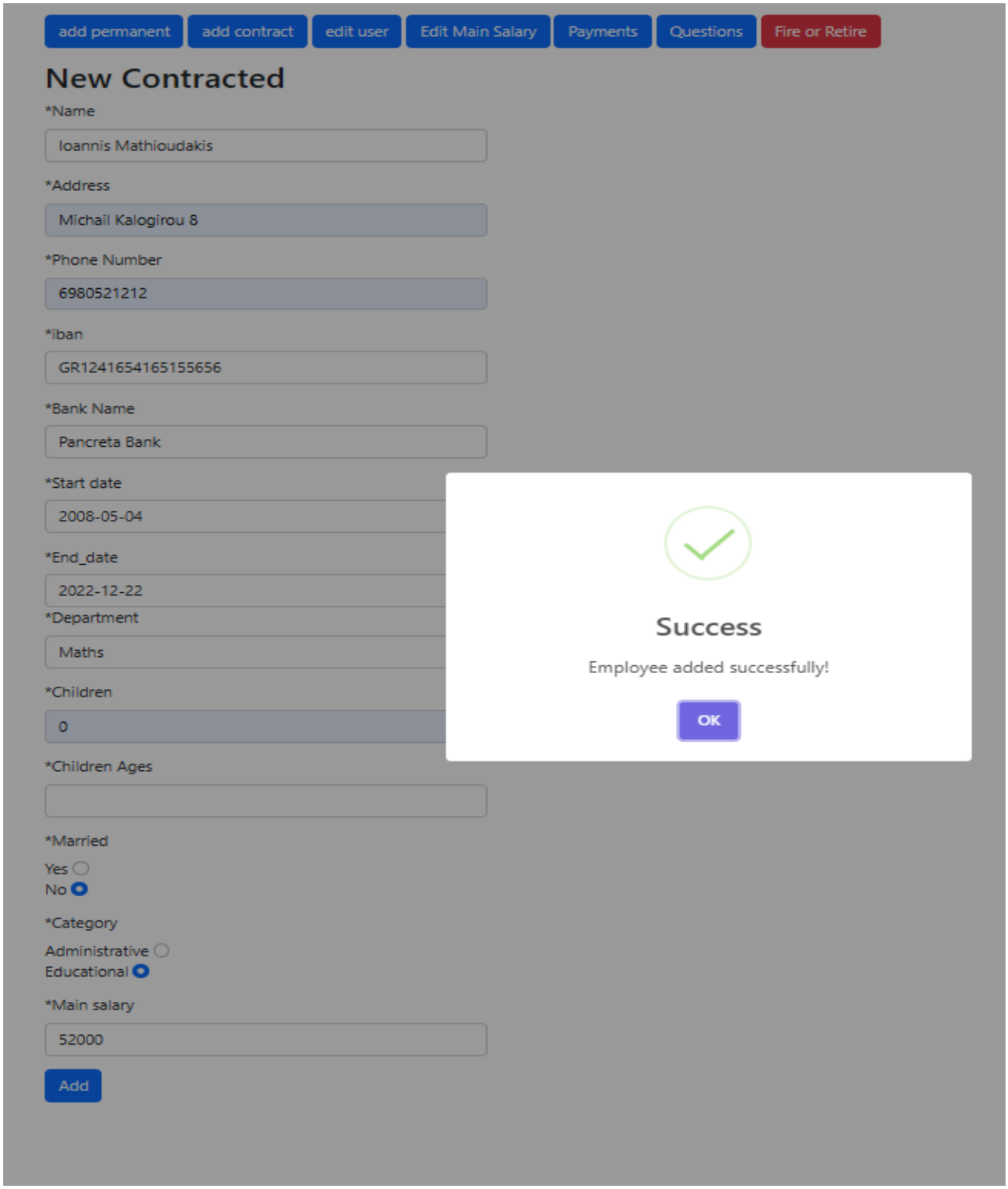
SELECT SUM(main_salary+bonus) **FROM** salary **WHERE** name **IN** (**SELECT** name **FROM** contracted **WHERE** category=1);

Testing & Screenshots.

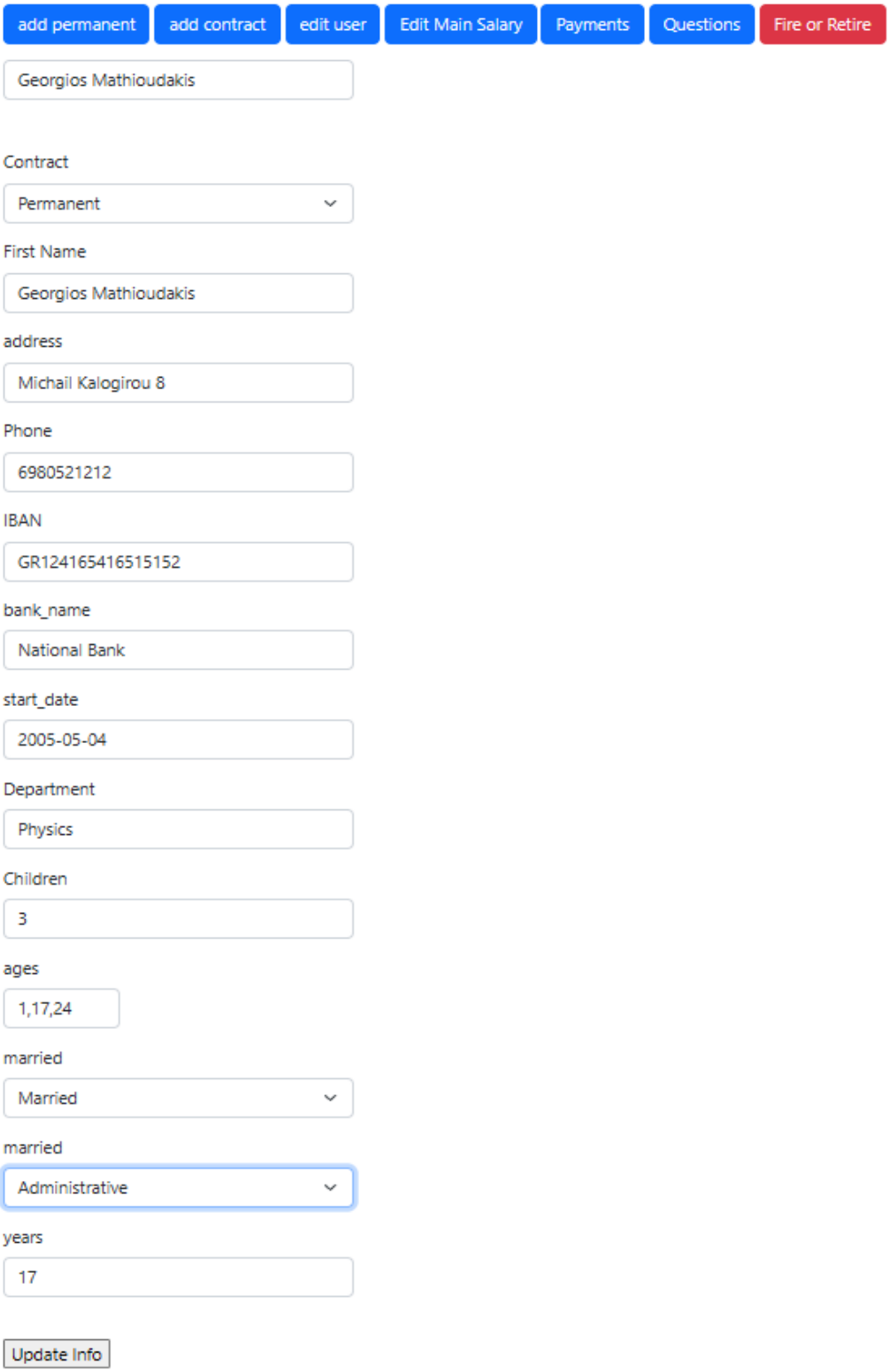
Add Permanent:



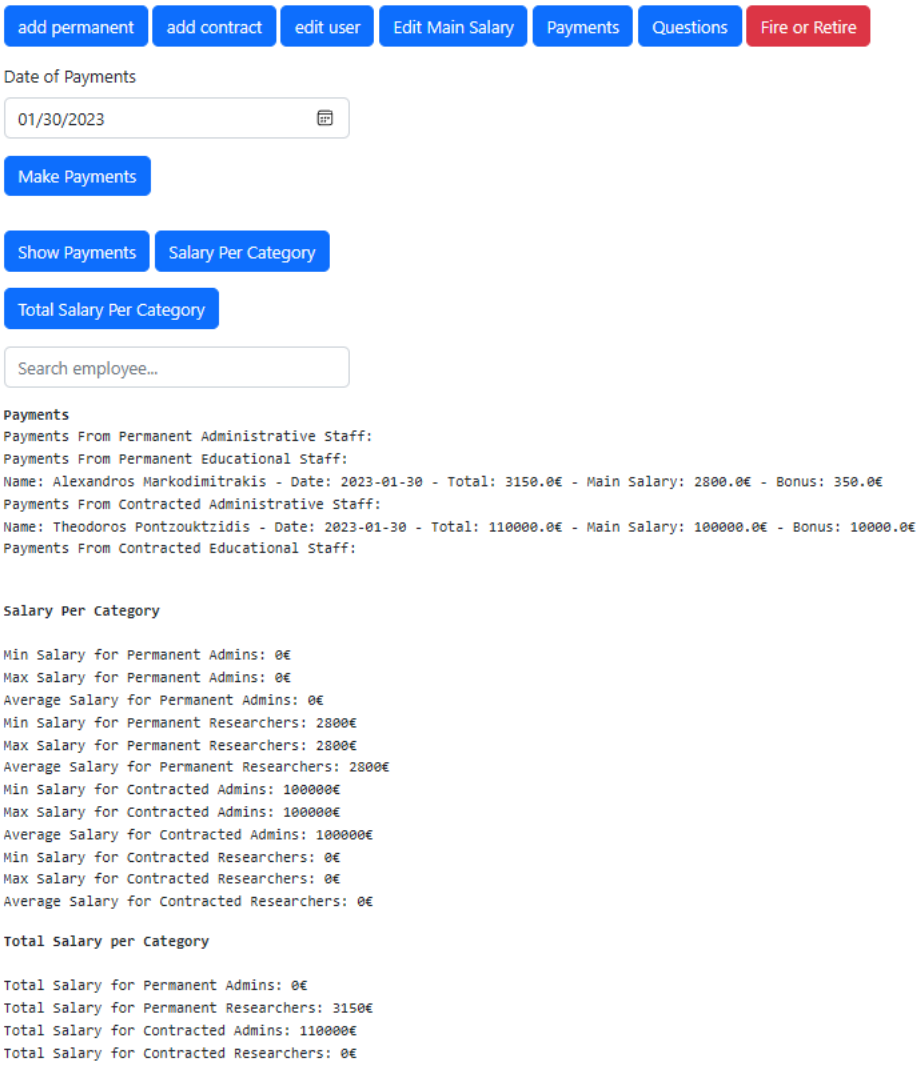
Add Contracted:



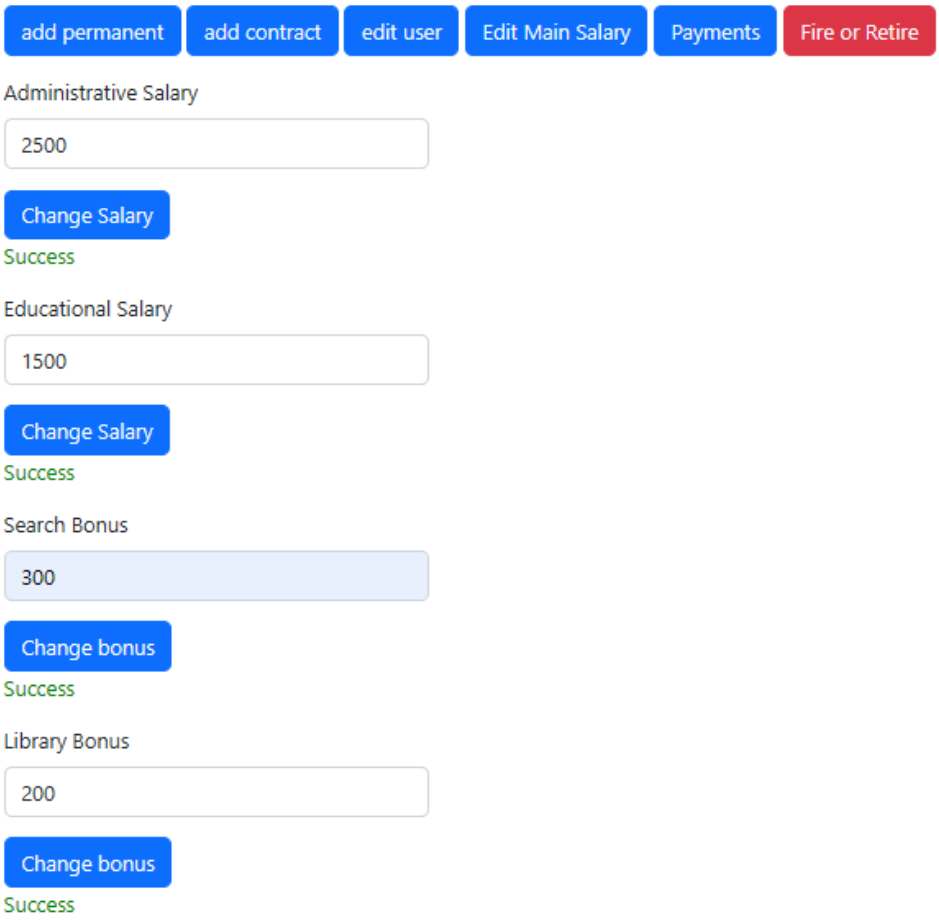
Edit Employee:



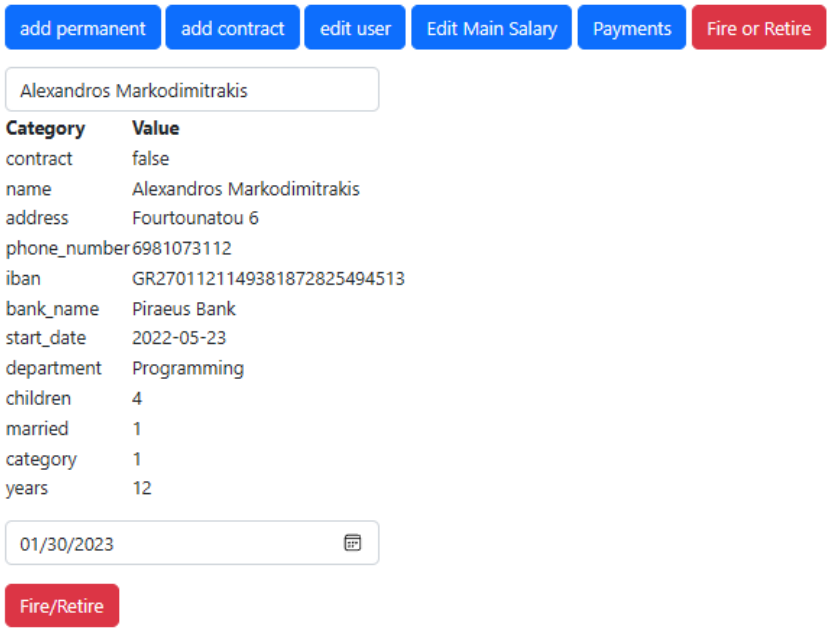
Make Payments/Show Payments/Salary statistics per Category/ Total Salary per Category:



Change Salaries and Bonuses:



Show Employee Information/ Fire-retire employ



Comments.

Question “Average Increase” was not implemented (no time).