## horizontal line

CS-360



Project Report

09.04.2023

**─**

Isidoros Chatzichrisos

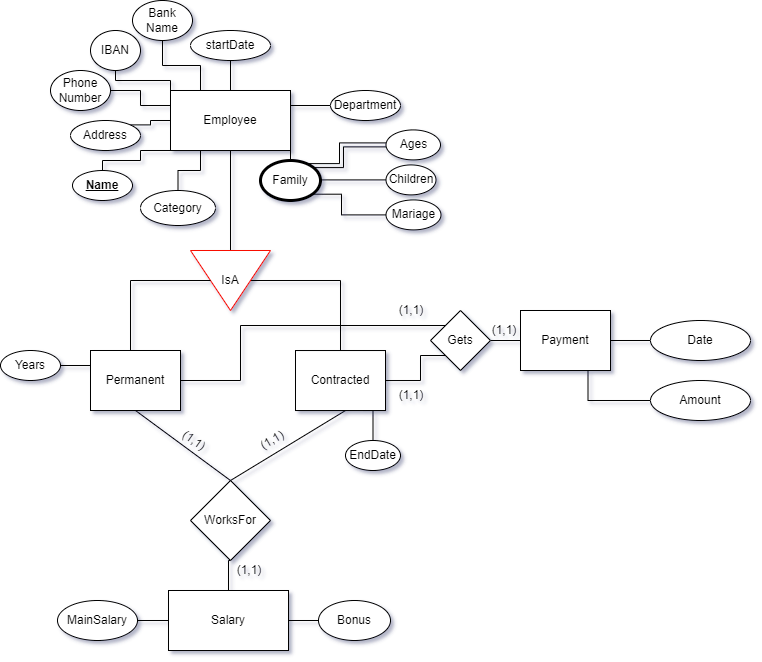
Theodoros Pontzouktzidis

Alexandros Markodimitrakis

# Content.

1. ER Diagram……………………………………………………………………………………………………Page: [2](#_swutbfzhmpw0)
2. Entities(Names,Types,Keys,Cardinalities)……………………………….……………………..Page: [3-5](#_6l5v4lcg50fo)
3. Relational Model…………………………………………………………………………………………...Page: [6](#_g333pnd5ion)
4. Keys & Functional Dependencies…………………………………………………………………..Page:
5. 3NF Without Data Loss………………………………………………………………………………….Page:
6. Testing & Screenshots…………………………………………………………………………………..Page:
7. Comments…………………………………………………………………………………………………….Page:

# ER Diagram.



# Entities (Names,Types,Keys,Cardinalities).

**Permanent:**

Name (String)

Address (String)

Phone Number (String)

IBAN (String)

BankName (String)

StartDate (DATE:SQL)

SalaryID (uint32)

Department (String)

Children (SMALLINT:SQL unsigned)

Mariage (Bool)

Category (BIT:SQL)

Years(TINYINT:SQL unsigned)

**Contracted:**

Name (String)

Address (String)

Phone Number (String)

IBAN (String)

BankName (String)

StartDate (DATE:SQL)

SalaryID (uint32)

Department (String)

Children (SMALLINT:SQL unsigned)

Mariage (Bool)

Category (BIT:SQL)

EndDate(DATE:SQL)

**Ages:**

Name (String)

Age (ENUM:SQL)

**Salary:**

*Name* (String)

Bonus (uint32)

MainSalary (uint32)

**Payment:**

*Name* (String)

Date (DATE:SQL)

Amount (uint32)

| **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 |

\* 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 | BankName | StartDate | SalaryID | Department | Children | Mariage | Category | Years |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Contracted**

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

**Ages**

| **Name** | Age |
| --- | --- |

**Salary**

| *Name* | Bonus | MainSalary |
| --- | --- | --- |

**Payment**

| *Name* | Date | Amount |
| --- | --- | --- |

# 

# 

# 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 **->** 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.*

# Testing & Screenshots.

# Comments.