

Міністерство освіти і науки України
Національний університет “Львівська політехніка”



Звіт
про виконання лабораторної роботи №1
із курсу “Аналітичні та нереляційні бази даних”

Виконав:
ст. гр. ІР-41, ІКТА
Питько Дмитро

Прийняв:
Верес З.Є.

Microsoft project діаграма класів

Скрипт для створення БД

```
USE [master]
GO

IF (db_id(N'lab1db') IS NOT NULL)
    DROP DATABASE lab1db;
GO

CREATE DATABASE lab1db;
GO

USE lab1db;
GO

CREATE TABLE Client (
    id INT IDENTITY PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    birth_date DATE NOT NULL,
    country VARCHAR(50),
    city VARCHAR(50),
    sex VARCHAR(10) NOT NULL CHECK (sex IN('MALE', 'FEMALE')),
    phone CHAR(13),
    CONSTRAINT CK_CLIENT_PHONE_NUMBER CHECK (phone LIKE
(' [0-9] [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9] [0-9] '))
);

CREATE TABLE Company (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(1000),
    website_url VARCHAR(1000),
    country VARCHAR(50),
    city VARCHAR(100),
    address VARCHAR(1000),
    phone CHAR(13),
    CONSTRAINT CK_COMPANY_PHONE_NUMBER CHECK (phone LIKE
(' [0-9] [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9] [0-9] '))
);

CREATE TABLE License (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(100),
    commercial BIT
);

CREATE TABLE Technology (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(100),
    description VARCHAR(1000),
    license_id INT NOT NULL,
    available BIT,
```

```

        CONSTRAINT FK_TECHNOLOGY_LICENSE_ID FOREIGN KEY (license_id) REFERENCES
License (id)
    );

CREATE TABLE Employee (
    id INT IDENTITY PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    birth_date DATE NOT NULL,
    country VARCHAR(50),
    city VARCHAR(50),
    sex VARCHAR(10) NOT NULL CHECK (sex IN('MALE', 'FEMALE')),
    phone CHAR(13),
    rate_per_hour REAL NOT NULL DEFAULT(0),
    CONSTRAINT CK_EMPL_PHONE_NUMBER CHECK (phone LIKE
(' [0-9] [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9] [0-9] '))
    );

CREATE TABLE Project (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(100),
    description VARCHAR(1000),
    start_date DATE,
    end_date DATE,
    status VARCHAR(15) NOT NULL CHECK (status IN('NEW', 'IN PROGRESS',
'FINISHED'))
    );

CREATE TABLE Task (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(1000),
    created_at DATE,
    start_date DATE,
    end_date DATE,
    status VARCHAR(15) NOT NULL CHECK (status IN('NEW', 'IN PROGRESS',
'FINISHED')),
    estimated_time_in_hours INT NOT NULL DEFAULT(0),
    project_id INT NOT NULL,
    employee_id INT NOT NULL,
    CONSTRAINT FK_PROJECT_TASK_ID FOREIGN KEY (project_id) REFERENCES Project
(id),
    CONSTRAINT FK_EMPLOYEE_TASK_ID FOREIGN KEY (employee_id) REFERENCES
Employee (id)
    );

CREATE TABLE CLIENT_PROJECT_COMPANY (
    client_id INT NOT NULL,
    project_id INT NOT NULL,
    company_id INT NOT NULL,
    budget INT,
    CONSTRAINT PK_CLIENT_PROJECT_COMPANY PRIMARY KEY (client_id, project_id,
company_id),

```

```

        CONSTRAINT FK_CLIENT_PROJECT_COMPANY_CLIENT_ID FOREIGN KEY (client_id)
REFERENCES Client (id),
        CONSTRAINT FK_CLIENT_PROJECT_COMPANY_COMPANY_ID FOREIGN KEY (company_id)
REFERENCES Company (id),
        CONSTRAINT FK_CLIENT_PROJECT_COMPANY_PROJECT_ID FOREIGN KEY (project_id)
REFERENCES Project (id)
    );

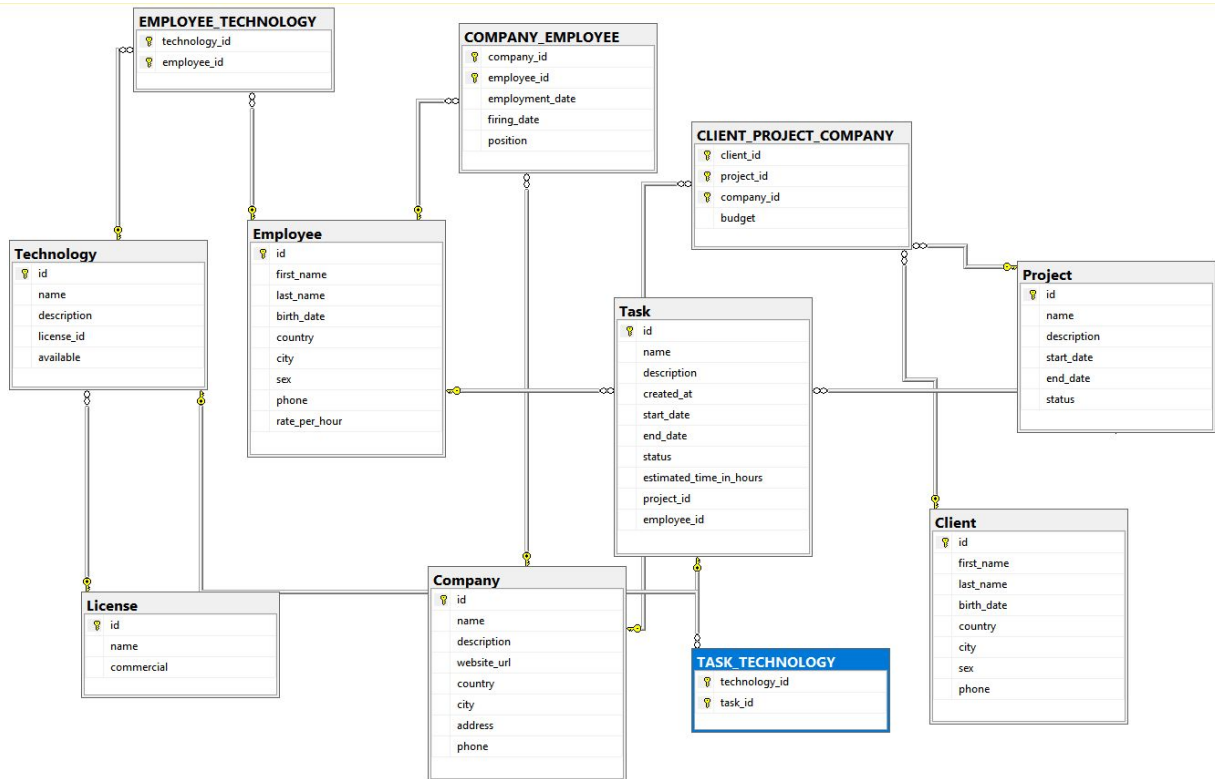
CREATE TABLE COMPANY_EMPLOYEE (
    company_id INT NOT NULL,
    employee_id INT NOT NULL,
    employment_date DATE,
    firing_date DATE,
    position VARCHAR(50),
    CONSTRAINT PK_COMPANY_EMPLOYEE PRIMARY KEY (company_id, employee_id),
    CONSTRAINT FK_COMPANY_EMPLOYEE_SERVICE_ID FOREIGN KEY (company_id)
REFERENCES Company (id),
    CONSTRAINT FK_COMPANY_EMPLOYEE_NUMBER_ID FOREIGN KEY (employee_id)
REFERENCES Employee (id)
    );

CREATE TABLE EMPLOYEE_TECHNOLOGY (
    technology_id INT NOT NULL,
    employee_id INT NOT NULL,
    CONSTRAINT PK_EMPLOYEE_TECHNOLOGY PRIMARY KEY (technology_id,
employee_id),
    CONSTRAINT FK_EMPLOYEE_TECHNOLOGY_TECHNOLOGY_ID FOREIGN KEY
(technology_id) REFERENCES Technology (id),
    CONSTRAINT FK_EMPLOYEE_TECHNOLOGY_EMPLOYEE_ID FOREIGN KEY (employee_id)
REFERENCES Employee (id)
    );

CREATE TABLE TASK_TECHNOLOGY (
    technology_id INT NOT NULL,
    task_id INT NOT NULL,
    CONSTRAINT PK_TASK_TECHNOLOGY PRIMARY KEY (technology_id, task_id),
    CONSTRAINT FK_TASK_TECHNOLOGY_TECHNOLOGY_ID FOREIGN KEY (technology_id)
REFERENCES Technology (id),
    CONSTRAINT FK_TASK_TECHNOLOGY_TASK_ID FOREIGN KEY (task_id) REFERENCES
Task (id)
    );

```

Діаграма створена у програмі MS SQL Management Studio



один до багатьох:

компанія - проекти (проект може існувати без компанії, але компанія їх має багато)

клієнт - проекти (проект не може існувати без клієнта, але і клієнт може мати кілька)

проект - таски (таска без проекту не існує і проект має багато їх)

працівник - таски (таска може існувати без працівника і працівник може не мати завдань)

багато до багатьох:

компанія - працівник (у компанії багато хто працює і кожен може працювати на багато компаній водночас)

працівник - технологія (одна технологія використовується різними працівниками і працівник їх може знати кілька)

технологія - таски (одна таска може вимагати знання різних технологій і технологія потрібна для виконання різних завдань)