

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”



Звіт до лабораторної роботи №1

із курсу: “Аналітичні та нереляційні бази даних”

**Виконав:**

Ст. Гр. Ір-42 ІКТА

Питько Остап

**Прийняв:**

Викладач

Зеновій Євгенович

Львів – 2021

## Завдання:

28	Питько Остап	Workday (Financial Management)	<p><b>Опишіть варіант(и) використання (use cases).</b> Аналізу фінансової інформації в компанії в системі. Варіант(и) використання мають містити опис всіх необхідних компонентів варіанту використання, зокрема, основний порядок дій та альтернативні шляхи, винятки.</p> <p><b>Намалюйте діаграму класів [class diagram],</b> щоб представити множину об'єктів, необхідних для преставлення фінансової звітності у вигляді таблиць та графіків.  <i>Примітка - зазвичай діаграма класів служить для представлення статичної структури моделі системи в цілому, однак для цілей цієї лабораторної роботи намалюйте лише ту її частину, яка стосується поставленої задачі</i></p> <p><b>Намалюйте діаграму активності [activity diagram]</b> створення квартального фінансового звіту та його подальший експорт у форматі pdf</p> <p><b>Намалюйте діаграму послідовності [sequence diagram],</b> яка показує яким чином отримується інформація про фінансові показники та підготовка звіту на їх основі і відображення цих звітів користувачу</p>
----	-----------------	--------------------------------------	--

## SQL script до завдання:

```
USE [master]
GO
```

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

```
CREATE TABLE Address (
    id INT IDENTITY PRIMARY KEY,
    country VARCHAR(50),
    city VARCHAR(100),
    address VARCHAR(1000)
);
```

```
CREATE TABLE Office (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(1000),
    phone CHAR(11),
    address_id INT NOT NULL,
```

```

        CONSTRAINT CK_OFFICE_PHONE_NUMBER CHECK (phone LIKE
(''[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]'')),
        CONSTRAINT FK_OFFICE_ADDRESS_ID FOREIGN KEY (address_id) REFERENCES
Address (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(11),
    employment_date DATE,
    position VARCHAR(50),
    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]''))
);

```

```

CREATE TABLE Report (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(100),
    date DATE,
    general_sum INT NOT NULL DEFAULT(0),
    accountant_id INT NOT NULL,
    employee_id INT NOT NULL,
    CONSTRAINT FK_REPORT_ACCOUNTANT_ID FOREIGN KEY (accountant_id)
REFERENCES Employee (id),
    CONSTRAINT FK_REPORT_EMPLOYEE_ID FOREIGN KEY (employee_id)
REFERENCES Employee (id)
);

```

```

CREATE TABLE PaymentType (
    id INT IDENTITY PRIMARY KEY,
    name VARCHAR(100)
);

```

```

CREATE TABLE Payment (
    id INT IDENTITY PRIMARY KEY,
    date DATE,
    amount INT NOT NULL DEFAULT(0),
    payment_type_id INT NOT NULL,
    employee_id INT NOT NULL,
    CONSTRAINT FK_TYPE_PAYMENT_ID FOREIGN KEY (payment_type_id)
REFERENCES PaymentType (id),

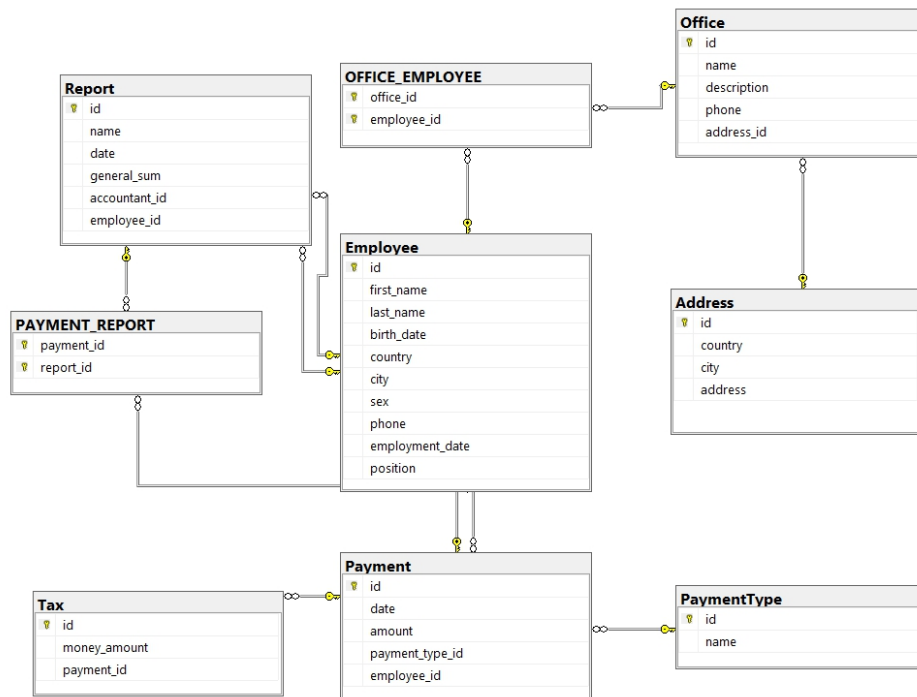
```

```
    CONSTRAINT FK_EMPLOYEE_PAYMENT_ID FOREIGN KEY (employee_id)
REFERENCES Employee (id)
);
```

```
CREATE TABLE Tax (
    id INT IDENTITY PRIMARY KEY,
    money_amount INT NOT NULL DEFAULT(0),
    payment_id INT NOT NULL,
    CONSTRAINT FK_PAYMENT_TAX_ID FOREIGN KEY (payment_id) REFERENCES
Payment (id)
);
```

```
CREATE TABLE OFFICE_EMPLOYEE (
    office_id INT NOT NULL,
    employee_id INT NOT NULL,
    CONSTRAINT PK_OFFICE_EMPLOYEE PRIMARY KEY (office_id, employee_id),
    CONSTRAINT FK_OFFICE_EMPLOYEE_OFFICE_ID FOREIGN KEY (office_id)
REFERENCES Office (id),
    CONSTRAINT FK_OFFICE_EMPLOYEE_EMPLOYEE_ID FOREIGN KEY (employee_id)
REFERENCES Employee (id)
);
```

```
CREATE TABLE PAYMENT_REPORT (
    payment_id INT NOT NULL,
    report_id INT NOT NULL,
    CONSTRAINT PK_PAYMENT_REPORT PRIMARY KEY (payment_id, report_id),
    CONSTRAINT FK_PAYMENT_REPORT_PAYMENT_ID FOREIGN KEY (payment_id)
REFERENCES Payment (id),
    CONSTRAINT FK_PAYMENT_REPORT_REPORT_ID FOREIGN KEY (report_id)
REFERENCES Report (id)
);
```



## Зв'язки:

### ● Один до багатьох:

бухгалтер - звіти

працівник - звіти

звіт - платежі (звіт включає в себе дані про різні платежі)

працівник - платежі

працівник - податки

### ● Багато до багатьох:

компанія - працівник (одна компанія має багато працюючих, але кожен з них може працювати на кількох роботах)

компанія - бухгалтер (аналогічно)