# Лабораторна робота 1

**Варіант 1**

Керуючий невеликого регіонального банку попросив вас спроектувати та реалізувати базу даних. Побудуйте ER-діаграму інформаційної системи, що супроводжує роботу банку.

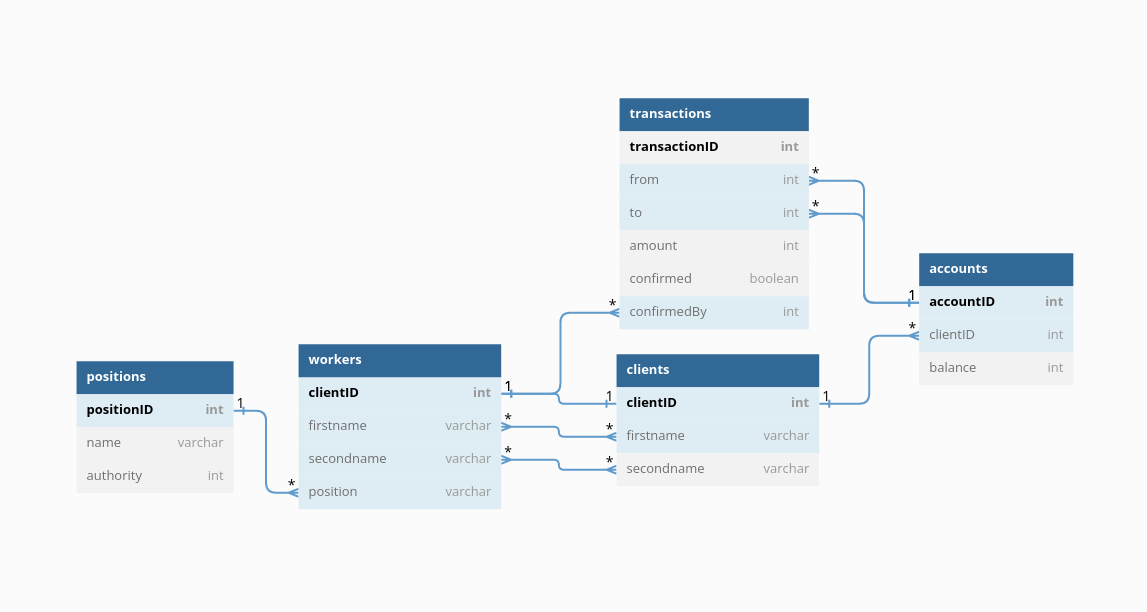
**«Базове» завдання (на «3»)**

* Клієнти банку можуть мати довільну кількість рахунків для проведення фінансових операцій
* Кожна операція (зняття або внесення грошей) з рахунком клієнта обробляється (підтверджується) працівником банку.

**«Розширене» завдання (на «4»)**

* Співробітник може бути клієнтом банку.
* У співробітника є посада. Існує перелік можливих посад.

Першим етапом виконання завдання буде створення ER діаграми за допомогою сервісу dbdiagram.io Створюємо таблиці відповідно до вимог, що вказані у завданні варіанту.

Рис. 1 — Діаграма до завдання лабораторної роботи.

За результатом першого етапу роботи було створено діагараму, що зображена на рис. 1. Вона містить схему всіх таблиць, що повинні бути створені, поля в них та зв’язки. Наступним етапом буде створення SQL скрипта для ініціалізації такої БД. На даном сервісі (dbdiagram.io) є можливість автоматично сгенерувати такий скрипт, чим я і скористаюся.

Нижче прикладаю SQL скрипт для створення бази даних, що показана на ERD.

TABLE `positions` (

`positionID` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(255),

`authority` int

);

CREATE TABLE `workers` (

`clientID` int PRIMARY KEY AUTO\_INCREMENT,

`firstname` varchar(255),

`secondname` varchar(255),

`position` varchar(255)

);

CREATE TABLE `clients` (

`clientID` int PRIMARY KEY AUTO\_INCREMENT,

`firstname` varchar(255),

`secondname` varchar(255)

);

CREATE TABLE `accounts` (

`accountID` int PRIMARY KEY AUTO\_INCREMENT,

`clientID` int,

`balance` int

);

CREATE TABLE `transactions` (

`transactionID` int PRIMARY KEY AUTO\_INCREMENT,

`from` int,

`to` int,

`amount` int,

`confirmed` boolean,

`confirmedBy` int

);

ALTER TABLE `workers` ADD FOREIGN KEY (`position`) REFERENCES `positions` (`positionID`);

ALTER TABLE `workers` ADD FOREIGN KEY (`clientID`) REFERENCES `clients` (`clientID`);

CREATE TABLE `clients\_workers` (

`clients\_firstname` varchar NOT NULL,

`workers\_firstname` varchar NOT NULL,

PRIMARY KEY (`clients\_firstname`, `workers\_firstname`)

);

ALTER TABLE `clients\_workers` ADD FOREIGN KEY (`clients\_firstname`) REFERENCES `clients` (`firstname`);

ALTER TABLE `clients\_workers` ADD FOREIGN KEY (`workers\_firstname`) REFERENCES `workers` (`firstname`);

CREATE TABLE `clients\_workers(1)` (

`clients\_secondname` varchar NOT NULL,

`workers\_secondname` varchar NOT NULL,

PRIMARY KEY (`clients\_secondname`, `workers\_secondname`)

);

ALTER TABLE `clients\_workers(1)` ADD FOREIGN KEY (`clients\_secondname`) REFERENCES `clients` (`secondname`);

ALTER TABLE `clients\_workers(1)` ADD FOREIGN KEY (`workers\_secondname`) REFERENCES `workers` (`secondname`);

ALTER TABLE `accounts` ADD FOREIGN KEY (`clientID`) REFERENCES `clients` (`clientID`);

ALTER TABLE `transactions` ADD FOREIGN KEY (`from`) REFERENCES `accounts` (`accountID`);

ALTER TABLE `transactions` ADD FOREIGN KEY (`to`) REFERENCES `accounts` (`accountID`);

ALTER TABLE `transactions` ADD FOREIGN KEY (`confirmedBy`) REFERENCES `workers` (`clientID`);