Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук Кафедра прикладної математики

# Звіт

до лабораторної роботи №1

з дисципліни

# Організація баз даних та знань

Виконала: Студент ПМ-22 Кравець Павло Прийняв:

Польовий В.Є.

Львів 2023

**Лабораторна робота № 1. “Створення схеми БД”**

1. Створити концептуальну та логічну модель предметної області згідно з описом, отриманим в викладача (відповідно до обраної предметної області).
2. Створити базу даних: створити таблиці, задати всі потрібні обмеження, створити зв’язки між таблицями. Схема бази даних повинна містити приклади використання сурогатних ключів створених за допомогою послідовностей (SEQUENCES). Схема бази даних повинна містити приклади використання всіх можливих способів перевірок обмеження цілісності, зокрема зовнішні ключі, вимоги на унікальність полів, перевірки типу CHECK, перевірки типу NULL/NOT NULL, тощо.
3. Ввести по декілька записів в кожну таблицю, проаналізувати цілісність даних.

## Варіант 2. “Оренда приміщень”

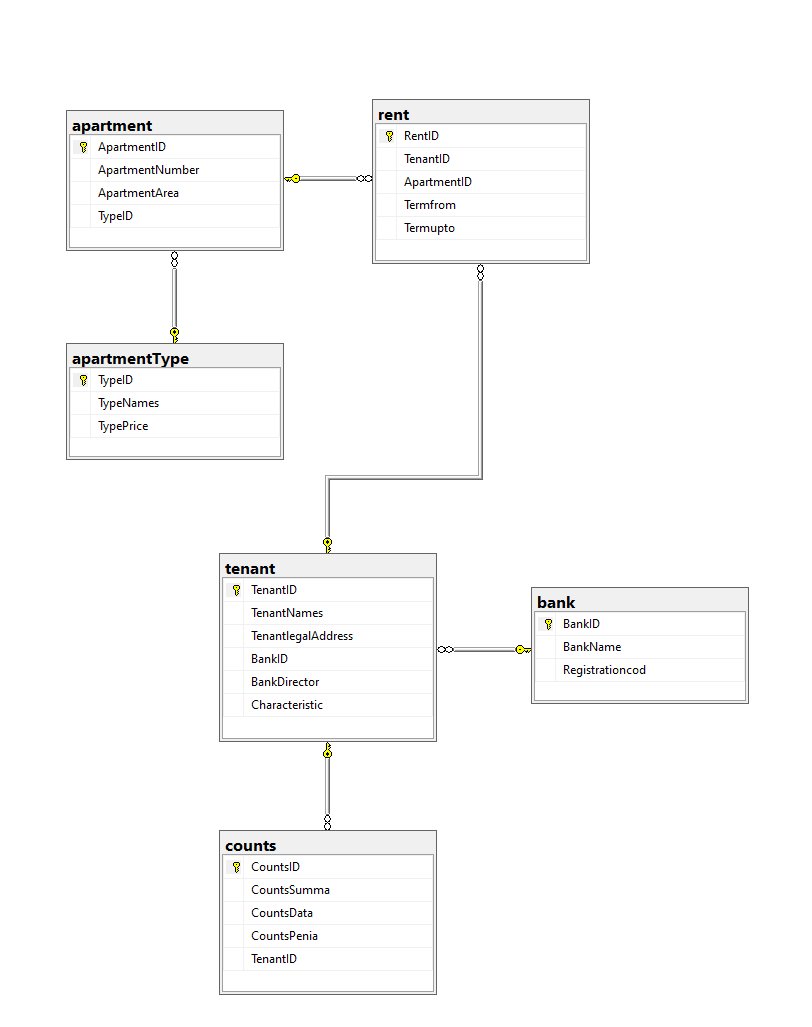
*Деяка організація здає приміщення в оренду різним організаціям. Інформаційна система повинна проводити облік використання приміщень, нарахування оплати за оренду та розрахунки з орендарями.*

Кожне приміщення має свій номер та площу, воно є приміщенням певного типу (кімната, склад, тощо). Кожен тип приміщення має свою ціну з розрахунку на кв. метр. Кожен орендар – це фірма яка має назву, юридичну адресу, банк в якому має рахунок, директора, характеристику. Орендар орендує декілька приміщень, кожне з яких на певний термін. На початку кожного місяця повинен генеруватися рахунок на оплату за оренду для кожного орендаря. Сума рахунку повинна включати податок 20%. Кожен рахунок повинен бути оплачений на протязі 15 днів. За кожен прострочений день нараховується пеня в розмірі 0.2% від неоплаченої суми. Пеня повинна бути врахована в сумі чергового рахунку. Система повинна враховувати надходження оплати від орендарів по рахунках. Кімната не може одночасно бути здана двом орендарям. Орендар, який має борг більший ніж сума трьохмісячної оренди не може отримувати в оренду нові приміщення.

### Система повинна надавати наступні звіти:

* Список орендарів з інформацією які приміщення він орендує, місячною сумою оренди та сумою боргу.
* Список приміщень з вказанням орендаря на певний день.
* Довідка для орендаря (параметр) з вказанням приміщень які він коли-небудь орендував, рахунків на оплату та дат оплати цих рахунків.

## Схема:

****

**SQL-запити для створення таблиць:**

CREATE CREATE TABLE apartmentType(

TypeID INT PRIMARY KEY not null,

TypeNames VARCHAR(200)not null,

TypePrice int not null,

)

CREATE TABLE apartment(

ApartmentID INT PRIMARY KEY not null,

ApartmentNumber VARCHAR(200) not null,

ApartmentArea FLOAT not null,

TypeID INT not null,

foreign key (TypeID) references apartmentType(TypeID)

)

create table bank(

BankID INT PRIMARY KEY not null,

BankName VARCHAR(200),

Registrationcod int unique not null

)

CREATE TABLE tenant(

TenantID INT PRIMARY KEY not null,

TenantNames VARCHAR(200) not null,

TenantlegalAddress varchar(200) unique not null,

BankID int not null,

BankDirector varchar(200) not null,

Characteristic varchar,

foreign key(BankID)references bank(BankID)

)

CREATE TABLE rent(

RentID INT PRIMARY KEY not null,

TenantID int not null,

ApartmentID int not null,

Termfrom date not null,

Termupto date not null,

foreign key(TenantID)references tenant(TenantID),

foreign key(ApartmentID)references apartment(ApartmentID)

)

create table counts(

CountsID INT PRIMARY KEY not null,

CountsSumma int not null,

CountsData date not null,

CountsPenia int not null,

TenantID int not null,

foreign key(TenantID)references tenant(TenantID)

)SQL-запити для наповнення таблиць:

insert into apartmentType values (1,'room',200);

insert into apartmentType values (2,'warehouse',500);

select\* from apartmentType;

insert into apartment values (1, 1,150,1);

insert into apartment values (2, 2,100,1);

insert into apartment values (3,3,200,2);

select\* from apartment;

insert into bank values (1, 'Privat24',4567);

insert into bank values (2,'Oshad', 1234);

select\* from bank;

insert into tenant values (1, 'BTS','Україна, \*\*1, місто Львів, Вулиця Дорошенка, будинок \*\*А',1,'Petrenko Dmytro Tarasovych','');

insert into tenant values (2,'AKA','Україна, \*\*1, місто Київ, ВУЛИЦЯ Героїв УПА, будинок \*\*Д', 2, 'Ochkivskyi Danylo Olegovych','');

select\* from tenant;

insert into counts values (1,2020,'2022-04-04',0,1);

insert into counts values (2,3020,'2022-03-04',0,1);

select\*from counts;

insert into rent values (1,1,2,'2022-02-03','2023-02-03');

insert into rent values (2,2,1,'2021-02-03','2022-02-03');

insert into rent values (3,2,2,'2021-09-07','2022-07-09');

select\*from rent;

select\* from apartmentType;

select\* from apartment;

select\* from bank;

select\* from tenant;

select\*from counts;

select\*from rent

## Концепція роботи БД на основі представленої схеми:

Таблиця apartmentType містить інформацію про типи квартир, які доступні для оренди. Кожен тип має унікальний ідентифікатор TypeID, назву TypeNames та ціну TypePrice.

Таблиця apartment містить інформацію про кожну квартиру, доступну для оренди. Кожна квартира має унікальний ідентифікатор ApartmentID, номер квартири ApartmentNumber, площу квартири ApartmentArea та TypeID, що посилається на тип квартири в таблиці apartmentType за допомогою зовнішнього ключа.

Таблиця bank містить інформацію про банки, які можуть використовуватися для здійснення платежів. Кожен банк має унікальний ідентифікатор BankID, назву BankName та унікальний код реєстрації Registrationcod.

Таблиця tenant містить інформацію про орендарів квартир. Кожен орендар має унікальний ідентифікатор TenantID, ім'я TenantNames, юридичну адресу TenantlegalAddress та інформацію про банк, який він використовує для здійснення платежів (BankID та BankDirector). Також таблиця містить поле Characteristic для збереження додаткової інформації про орендарів.

Таблиця rent містить інформацію про оренду квартир. Кожна оренда має унікальний ідентифікатор RentID, інформацію про орендаря (TenantID) та квартиру (ApartmentID), а також дати початку (Termfrom) та закінчення (Termupto) оренди. Таблиця містить зовнішні ключі, які посилаються на таблиці tenant та apartment.

Таблиця counts містить інформацію про розрахунки за оренду. Кожен розрахунок має унікальний ідентифікатор CountsID, CountsData

CountsSumma, ConstPenia.

https://github.com/Legamanchik/database