

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**імені ІГОРЯ СІКОРСЬКОГО»**

факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Пояснювальна записка до курсової роботи**

з дисципліни «Бази даних»

на тему: « База даних обліку студентських гуртожитків»

Виконав : студент 2 курсу, групи ІТ–01

Дурдинець Олександр Тиберійович

Науковий керівник Марченко Олена Іванівна

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Члени комісії

\_\_\_\_\_ (посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що в цій курсовій роботі немає запозичень із праць  
інших авторів без відповідних посилань.

Дурдинець О. Т.

\_\_\_\_\_ (підпис)

Київ – 2021

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки  
(повна назва)

Кафедра Інформатики та програмної інженерії  
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІТ-01 Семестр 1

**З А В Д А Н Н Я  
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Дурдинець Олександр Тиберійович  
(прізвище, ім'я, по батькові)

1. Тема роботи: База даних обліку студентських гуртожитків.

керівник роботи ст. викладач Марченко Олена Іванівна  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 24.12.2021

3. Вихідні дані до роботи: методичні вказівки до виконання курсової роботи з дисципліни “Бази даних” першого бакалаврського рівня вищої освіти; варіант з темою курсової роботи, що визначається номером списку у групі; задачі, положення, завдання, тематика та вимоги до виконання курсової роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): навчитись аналізувати предметне середовище, визначати атрибути та сутності, зв'язки між об'єктами; навчитись будувати ER-модель для заданого предметного середовища; навчитись будувати реляційну схему бази даних на основі ER-моделі (будувати набір необхідних відношень бази даних, виділяти первинні та зовнішні ключі, визначати обмеження для підтримки цілісності бази даних); розробляти відповідні скрипти з використанням засобів мови SQL для побудови спроектованої бази даних; імпорт даних з використанням засобів СУБД, створення запитів до розробленої БД, оптимізація роботи запитів (за необхідності)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 31.10.2021

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Вивчення літератури	01.12.2021	
2	Розробка концепції проекту	01.12.2021	
3	Створення ER-діаграми	05.12.2021	
4	Побудова реляційної схеми	06.12.2021	
5	Створення бази даних	10.12.2021	
6	Створення користувачів	18.12.2021	
7	Імпорт даних	19.12.2021	
8	Створення запитів	19.12.2021	

**Студент**

\_\_\_\_\_ Дурдинець Олександр Тиберійович  
(підпис) (прізвище та ініціали)

**Керівник роботи**

\_\_\_\_\_ Марченко О. І.  
(підпис) (прізвище та ініціали)

# **ЗМІСТ**

## **ВСТУП**

## **1 ТЕОРЕТИЧНА ЧАСТИНА**

- 1.1 Аналіз предметної області
  - 1.1.1 Опис предметної області
  - 1.1.2 Опис вхідних даних
  - 1.1.3 Опис вихідних даних
- 1.2 Проектування бази даних
  - 1.2.1 Інфологічна модель бази даних
    - 1.2.1.1 Опис сутностей та атрибутів
    - 1.2.1.2 Діаграма "сутність-зв'язок"
  - 1.2.2 Даталогічна модель бази даних

## **2 ПРАКТИЧНА ЧАСТИНА**

- 2.1 Створення бази даних за допомогою Microsoft SQL Server
  - 2.1.1 Створення бази даних
  - 2.1.2 Створення таблиць бази даних
  - 2.1.3 Створення діаграми бази даних
  - 2.1.4 Заповнення таблиць бази даних даними
  - 2.1.5 Створення збережених процедур
  - 2.1.6 Створення тригерів
  - 2.1.7 Створення представлень
  - 2.1.8 Створення запитів

## **ВИСНОВКИ**

## **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

## **ДОДАТКИ**

- Додаток А. Створення таблиць
- Додаток Б. Заповнення таблиць даними

## Додаток В. Запити

## ВСТУП

Будь-яка БД є частиною інформаційної системи і призначена для збору, зберігання і маніпулювання інформацією.

У сучасному світі жодна велика фірма не обходиться без бази даних. База даних являє собою спосіб управління інформації, що зберігається, і використовується у всіх сферах людського життя.

База даних систематизує такі матеріали як інформацію про співробітників, про клієнтів, про постачальників, розрахунки, звіти і так далі, і обробляє цю інформацію за допомогою програми в комп'ютері.

Створюючи базу даних, ми прагнемо впорядкувати інформацію за різними ознаками і швидко витягати вибірку з поєднанням ознак.

Проектування бази даних являє собою складний трудомісткий процес відображення предметної області. Процес проектування бази даних полягає в створенні схеми бази даних і визначення обмежень інформації.

База даних може зберігати величезну кількість інформації, і швидко надавати її після введення запиту користувача. Ось чому бази даних дуже актуальні в нашому сучасному світі.

База даних - це організована структура, призначена для зберігання, зміни і обробки взаємозалежної інформації, переважно великих обсягів.

Реляційна база даних - це набір даних з зумовленими зв'язками між ними. Ці дані організовані у вигляді набору таблиць, що складаються із стовпців і рядків. У таблицях зберігається інформація про об'єкти, представлених в базі даних. У кожному стовпчику таблиці зберігається певний тип даних, в кожному осередку - значення атрибута. Кожна стока таблиці являє собою набір пов'язаних значень, що відносяться до одного об'єкту або сутності. Кожен рядок в таблиці може бути

позначена унікальним ідентифікатором, званим первинним ключем, а рядки з декількох таблиць можуть бути пов'язані з допомогою зовнішніх ключів.

СУБД - це програмна прошивка між користувачем і сервером. Тому вона дозволяє абстрагувати користувача від системного бачення БД, а системі надає спосіб взаємодіяти з користувачем.

Microsoft SQL Server - система керування базами даних.

Основними завданнями цієї системи є:

- маніпуляція інформацією, яка знаходиться у базі даних,
- організація одночасного доступу до даних великої кількості користувачів.

Microsoft SQL Server підтримує реляційну модель даних і створює об'єкти БД (таблиць, індексів, представлень), здійснює перевірку цілісності БД і відповідає за безпеку даних.

В даній курсовій роботі була розроблена реляційна база даних “База даних обліку студентських гуртожитків” за допомогою Microsoft SQL Server, в якій необхідно було вирішити такі завдання:

- навчитись аналізувати предметне середовище, визначати атрибути та сутності, зв'язки між об'єктами;
- навчитись будувати ER-модель для заданого предметного середовища;
- навчитись будувати реляційну схему бази даних на основі ER-моделі (будувати набір необхідних відношень бази даних, виділяти первинні та зовнішні ключі, визначати обмеження для підтримки цілісності бази даних);
- розробляти відповідні скрипти з використанням засобів мови SQL для побудови спроектованої бази даних;

—імпорт даних з використанням засобів СУБД, створення запитів до розробленої БД, оптимізація роботи запитів (за необхідності)



# 1 ТЕОРЕТИЧНА ЧАСТИНА

## 1.1 Аналіз предметної області

### 1.1.1 Опис предметної області

Інформаційна система відповідає за забезпечення інформаційних процесів, забезпечення створення, поширення, використання, збереження і знищення інформації. Сама інформаційна база складається з однієї або декількох баз даних.

Предметна область - частина реального світу, що описує інформаційна система в залежності від її призначення.

В даний час усі організації повинні мати доступ до інформації. Це дуже важливо, тому бази даних, які задовольняють потреби зі зберіганням й управлінням даних, мають неабияку цінність, бо допомагають людям в цій справі.

При розробці реляційної бази даних “База даних обліку студентських гуртожитків” було проведено дослідження предметної області.

Отже, в мене є гуртожитки в різних містах і які належать різним університетам. В кожному гуртожитку є певна кількість кімнат розрахованих на деяку кількість студентів. Гуртожитки діляться на декілька типів: блочного типу, коридорного типу, змішані та сімейні. Також в них є коменданти і завідувачі.

### 1.1.2 Опис вхідних даних

Знаючи свою предметну область нам потрібно ввести дані, що призначені для введення, щоб наша база даних була потрібною. При розробці реляційної бази даних “База даних обліку студентських гуртожитків” були виділені наступні вхідні дані: - інформація про гуртожитки, - інформація про типи гуртожитків, - інформація про студентів, - інформація про кімнати, - інформація про замовлення, - інформація про майстерні, - інформація про співробітників.

### 1.1.3 Опис вихідних даних

Вихідні дані - повідомлення і результати, які видаються самою СУБД.

Беруться з постійних даних. Вихідними даними для “База даних обліку студентських гуртожитків” є вихідні запити. Детальніше про це в “практична частина”.

## 1.2 Проектування бази даних

### 1.2.1 Інфологічна модель бази даних

Проектувальники інфологічної моделі розглядають як модель сутність - зв'язок”.

Призначення інфографічної моделі полягає у наданні найбільш дійсних способів збору та представлення інформації, що зберігається у базі даних.

Ось чому інфологічну модель даних пробують будувати по аналогії з дійсною мовою.

Початковим етап в проектуванні баз даних - це інфологічна модель, яка допомагає в побудові таблиць та зв'язків між ними.

Існують різні способи опису інфологічної моделі, однак на даний момент найбільш популярним з підходів є той, що заснований на діаграмі "сутність-зв'язок". Отже, при створенні інфологічної моделі ми будемо використовувати ER – діаграму. Ця інфологічна модель - це модель предметної області, яка визначає сутності та зв'язки бази даних. Побудова такої моделі є індивідуальною і особливою, тому однієї методики створення інфологічної моделі немає, кожен робить так як хоче, так як саме він бачить цю модель предметної області.

Інфологічна модель – це опис бази даних, яка буде існувати в майбутньому, опис бд поданий за допомогою діаграм, таблиць, дійсної мови. Будувannya цієї моделі є продовженням аналізу предметної області (в моєму випадку База даних обліку студентських гуртожитків), інфологічна модель є представленням БД з точки зору того, хто розробляє її.

Головними конструктивними елементами інфологічної моделі даних є: сутності, їх атрибутів та типи зв'язків.

#### 1.2.1.1 Опис сутностей та атрибутів

Одним із елементом інфологічної моделі "сутність-зв'язок" є сутності.

Сутність – це те, про що накопичується інформація в інформаційній системі і що може бути однозначно унікальне ідентифіковане. При цьому ім'я сутності повинно відображати клас об'єкта або тип об'єкта.

Атрибут – це поіменована характеристика сутності, за допомогою якої моделюється її властивість. Атрибути називають ще інформаційними елементами.

Сутності повинні мати найменування з чітким смисловим значенням, краще, щоб найменування було в однині, а не в множині, тому що це доволі допомагає в читанні будь-якої моделі.

Кожна сутність може мати будь-яку кількість зв'язків з іншими сутностями моделі.

У відповідності з описом предметної області були отримано такі сутності:

- Гуртожитки(dormitories)
- Студенти(students)
- Комендант(commendants)
- Завідувачі(matrons)

Більш детальний опис сутностей та атрибутів можна побачити в табл. 1.1 - 1.10

#### 1.2.1.2 Діаграма "сутність-зв'язок"

На рис. 1.1 представлена ER-діаграма бази даних «База даних обліку студентських гуртожитків», на якій відображені всі сутності, їх атрибути та зв'язки між сутностями, .

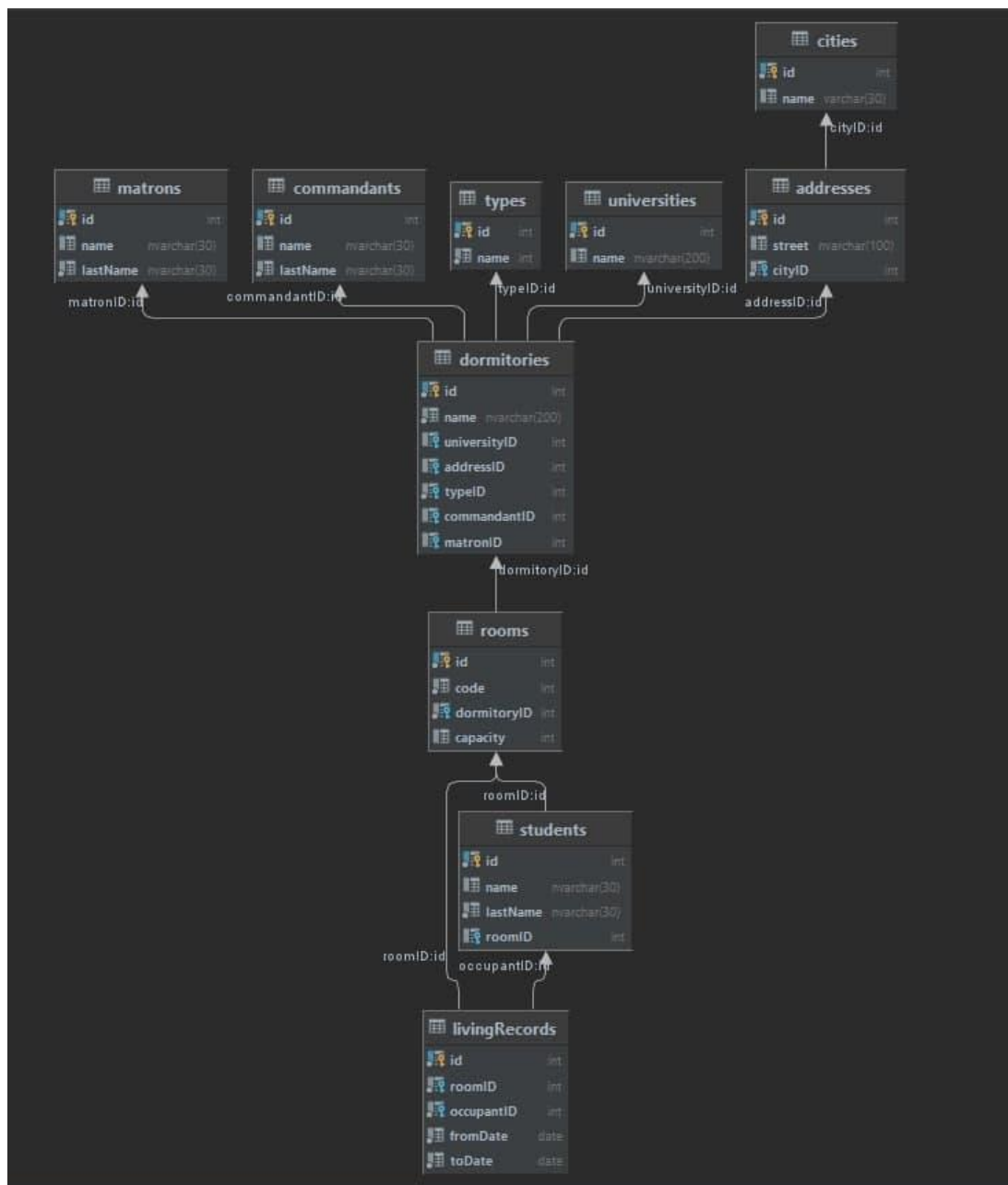


Рисунок 1.1 – ER-діаграма бази даних «База даних обліку студентських гуртожитків»

### 1.2.2 Даталогічна модель бази даних

На початку проектування бази даних використовують інфологічні моделі даних. Це потрібно для того, щоб описати саму структуру даних тоді, коли розробляється додаток, а даталогічні моделі бази даних деякими системами управління бд підтримуються.

Даталогічне проектування – затвердження рішення про те, як логічна модель

буде реалізована у базі даних, що створюється з використанням обраної СУБД.

Логічна структура бази даних, а також сама база даних, яка вже заповнена даними, є відображенням певної предметної області. Специфіка предметної області, яка відбита в інфологічній моделі, впливає на вибір проектних рішень.

Даталогічна модель є моделлю бази даних логічного рівня і являє собою відображення логічних зв'язків між елементами даних до їхнього змісту й середовищу зберігання.

У результаті даталогічного проектування ми можемо отримати декілька варіантів логічної моделі. Ось чому важливо вірно оцінити отримані моделі і вибрати більш оптимальний варіант.

Також потрібно перевіряти умови виконання всіх запитів і вимог користувачів, й якщо щось не так, потрібно перебудувати модель даних. Для кожного поля таблиці реляційної бази даних «База даних обліку студентських гуртожитків» вказується розмір поля, тип даних та обмеження для кожного поля, якщо необхідно та найменування самих атрибутів (в табл. 1.1 - 1.10).

#### Склад та характеристики атрибутів

Таблиця 1.1 - Адреси (addresses)

Найменування атрибутів	Тип поля	Розмір поля
street	nvarchar	100
cityID	INT	4

Таблиця 1.2 - Міста (cities)

Найменування атрибутів	Тип поля	Розмір поля
name	varchar	30

Таблиця 1.3 - Коменданти (commandants)

Найменування атрибутів	Тип поля	Розмір поля
------------------------	----------	-------------

name	nvarchar	30
lastName	nvarchar	30

Таблиця 1.4 - Гуртожитки (dormitories)

Найменування атрибутів	Тип поля	Розмір поля
name	nvarchar	200
universityID	INT	4
adressID	INT	4
typeID	INT	4
commandantID	INT	4
matronID	INT	4

Таблиця 1.5 - Дані проживання (livingRecords)

Найменування атрибутів	Тип поля	Розмір поля
roomID	int	4
occupantID	int	4
fromDate	date	
toDate	date	

Таблиця 1.6 - Завідувач (matrons)

Найменування атрибутів	Тип поля	Розмір поля
name	nvarchar	30
lastName	nvarchar	30

Таблиця 1.7 - Кімнати (rooms)

Найменування атрибутів	Тип поля	Розмір поля
code	INT	4
dormitoryID	INT	4
capacity	INT	4

Таблиця 1.8 - Студенти (students)

Найменування атрибутів	Тип поля	Розмір поля
name	nvarchar	30
lastName	nvarchar	30
roomID	INT	4

Таблиця 1.9 - Типи гуртожитків (types)

Найменування атрибутів	Тип поля	Розмір поля
name	nvarchar	30

Таблиця 1.10 - Університети (universities)

Найменування атрибутів	Тип поля	Розмір поля
name	nvarchar	200

## 2 ПРАКТИЧНА ЧАСТИНА

### 2.1 Створення бази даних за допомогою Microsoft SQL Server

#### 2.1.1 Створення бази даних

Спочатку заходимо в DataGrip та створюємо проект (рис. 2.1):

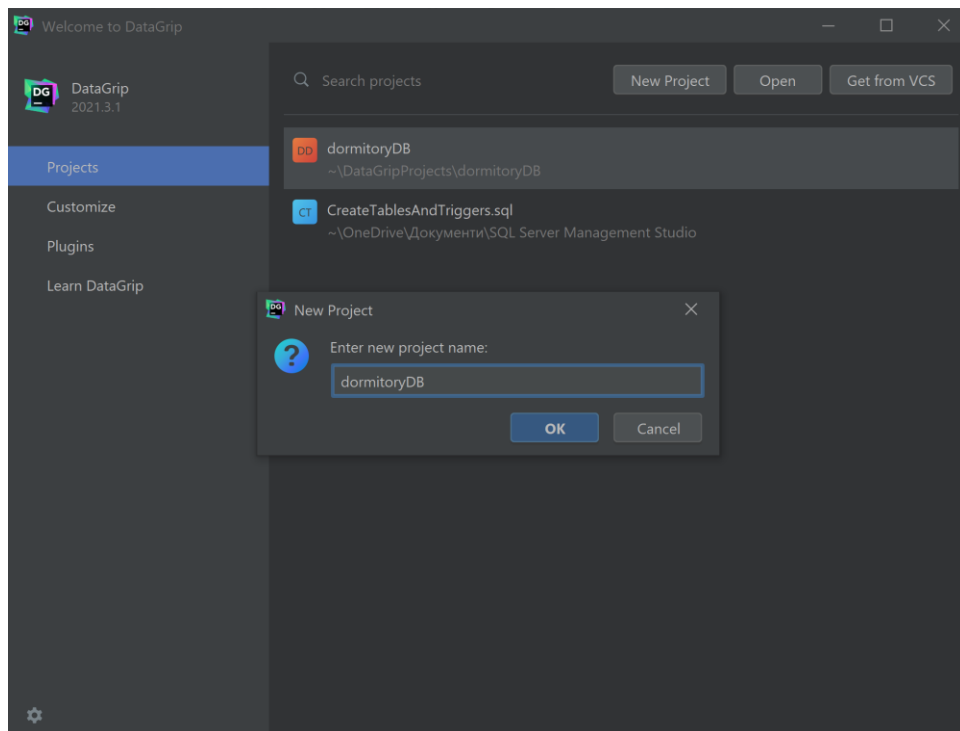


Рисунок 2.1 - Створення проекту

Оскільки в мене проект з таким ім'ям вже створений, просто заходжу в наявний.

Далі треба обрати середовище бази даних з наявних. Я обираю Microsoft SQL Server (рис 2.2):



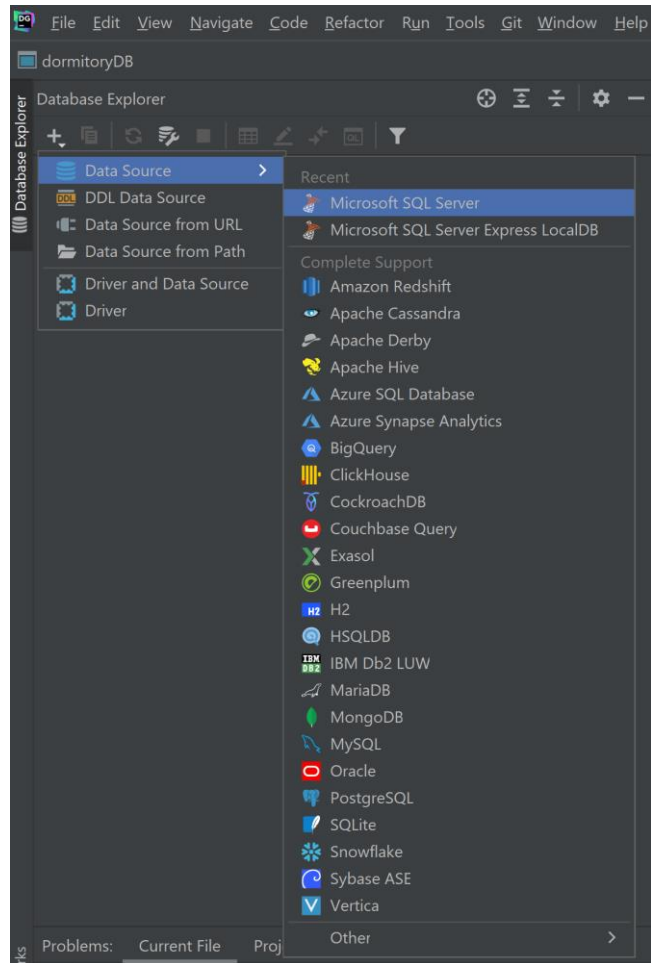


Рисунок 2.2 - обирання середовища бази даних

Створюємо базу даних (рис. 2.3):

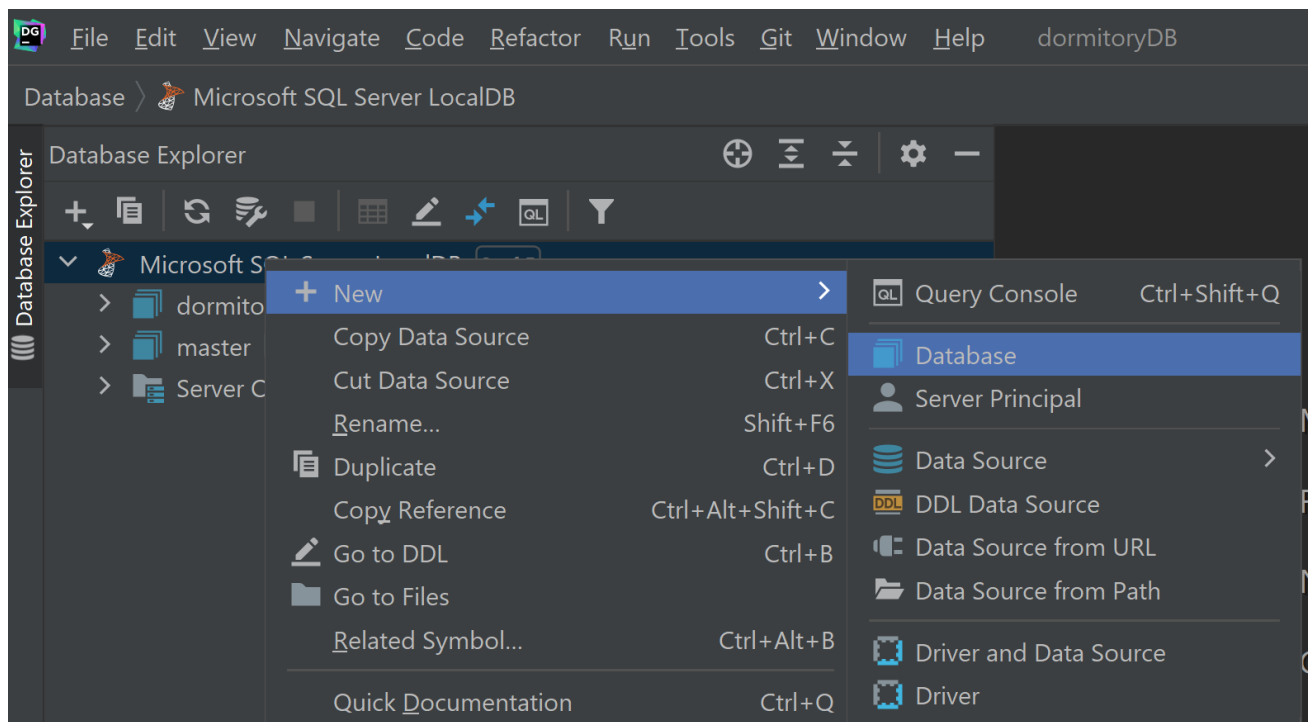


Рисунок 2.3 - створення бази даних

### 2.1.2 Створення таблиць бази даних

В базі даних dormitoryDB створюємо наступні таблиці:

- Адреси (addresses)
- Міста (cities)
- Коменданти(commandants)
- Гуртожитки (dormitories)
- Дані проживання (livingRecords)
- Завідувач (matrons)
- Кімнати (rooms)
- Студенти (students)
- Типи гуртожитків (types)
- Університети (universities)

за допомогою наступних команд

Створення таблиці Міста(cities)

```
use dormitoryDB
```

```
go
```

```
create table dormitoryDB.dbo.cities
```

```
(
```

```
    id  int not null
```

```
        constraint cities_pk
```

```
            primary key nonclustered,
```

```
    name varchar(30)
```

```
)
```

```
go
```

```
create unique index cities_id_uindex
```

```
on dormitoryDB.dbo.cities (id)
```

go

Створення таблиці Адреси(addresses)

```
create table dormitoryDB.dbo.addresses
```

```
(  
    id int identity (1, 1)  
        constraint addresses_pk  
            primary key,  
    street nvarchar(100),  
    cityID int not null  
        constraint addresses_cities_id_fk  
            references dbo.cities  
)
```

go

```
create unique index addresses_cityID_uindex  
    on dormitoryDB.dbo.addresses (id)
```

go

Створення таблиці Університети(universities)

```
create table dormitoryDB.dbo.universities
```

```
(  
    id int identity  
        constraint universities_pk  
            primary key nonclustered,  
    name nvarchar(200)  
)
```

go

```
create unique index universities_id_uindex
    on dormitoryDB.dbo.universities (id)
go
```

Створення таблиці Типи гуртожитків(types)

```
create table dormitoryDB.dbo.types
(
    id int identity
        constraint types_pk
            primary key nonclustered,
    name nvarchar(30) not null
)
go
```

```
create unique index types_id_uindex
    on dormitoryDB.dbo.types (id)
go
```

Створення таблиці Коменданти(commandants)

```
create table dormitoryDB.dbo.commandants
(
    id int identity
        constraint commandant_pk
            primary key nonclustered,
    name nvarchar(30),
    lastName nvarchar(30) not null
)
go
```

```
create unique index commandant_id_uindex
    on dormitoryDB.dbo.commandants (id)
```

go

Створення таблиці Завідуючі(matrons)

```
create table matrons
```

```
(  
    id      int identity  
        constraint matrons_pk  
            primary key nonclustered,  
    name     nvarchar(30),  
    lastName nvarchar(30) not null  
)
```

go

```
create unique index matrons_id_uindex
```

```
on dormitoryDB.dbo.matrons (id)
```

go

Створення таблиці Гуртожитки(dormitories)

```
create table dormitoryDB.dbo.dormitories
```

```
(  
    id      int identity  
        constraint dormitories_pk  
            primary key nonclustered,  
    name     nvarchar(200) not null,  
    universityID int  
        constraint dormitories_universities_id_fk  
            references dormitoryDB.dbo.universities,  
    addressID int  
        constraint dormitories_addresses_id_fk  
            references dormitoryDB.dbo.addresses,  
    typeID   int      not null
```

```

        constraint dormitories_types_id_fk
            references dormitoryDB.dbo.types,
commandantID int
        constraint dormitories_commandants_id_fk
            references dormitoryDB.dbo.commandants,
matronID    int
        constraint dormitories_matrons_id_fk
            references dormitoryDB.dbo.matrons
    )
go

```

```

create unique index dormitories_id_uindex
    on dormitoryDB.dbo.dormitories (id)
go

```

Створення таблиці Кімнати(rooms)

```

create table dormitoryDB.dbo.rooms
(
    id          int identity
        constraint rooms_pk
            primary key nonclustered,
    code        int not null,
    dormitoryID int not null
        constraint rooms_dormitories_id_fk
            references dormitoryDB.dbo.dormitories,
    capacity    int
)
go

```

```

create unique index rooms_id_uindex
    on dormitoryDB.dbo.rooms (id)
go

```

Створення таблиці Студенти(students)

```
create table students
(
    id      int identity
        constraint students_pk
            primary key nonclustered,
    name    nvarchar(30),
    lastName nvarchar(30) not null,
    roomID  int
        constraint students_rooms_id_fk
            references dormitoryDB.dbo.rooms
)
go
```

```
create unique index students_id_uindex
    on dormitoryDB.dbo.students (id)
go
```

Створення таблиці Термін проживання(livingRecords)

```
create table dormitoryDB.dbo.livingRecords
(
    id      int identity
        constraint livingRecords_pk
            primary key nonclustered,
    roomID  int not null
        constraint livingRecords_rooms_id_fk
            references dormitoryDB.dbo.rooms,
    occupantID int not null
        constraint livingRecords_students_id_fk
            references dormitoryDB.dbo.students,
```

```

        fromDate date not null,
        toDate date not null
    )
go

create unique index livingRecords_id_uindex
    on dormitoryDB.dbo.livingRecords (id)
go

```

### 2.1.3 Створення діаграми бази даних

Діаграма бази даних створюється наступним чином(рис. 2.4 , рис. 2.5):

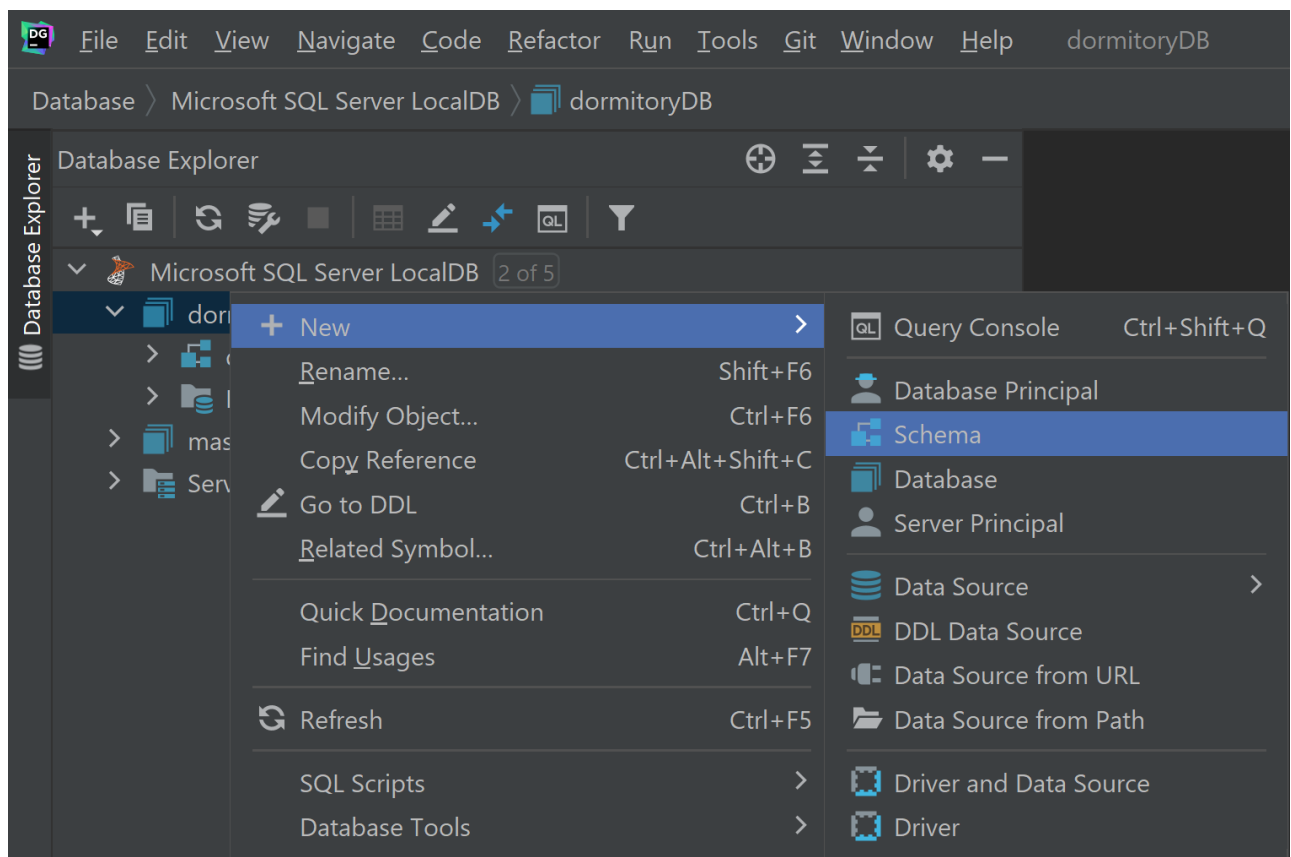


Рисунок 2.4 - створення діаграми бази даних



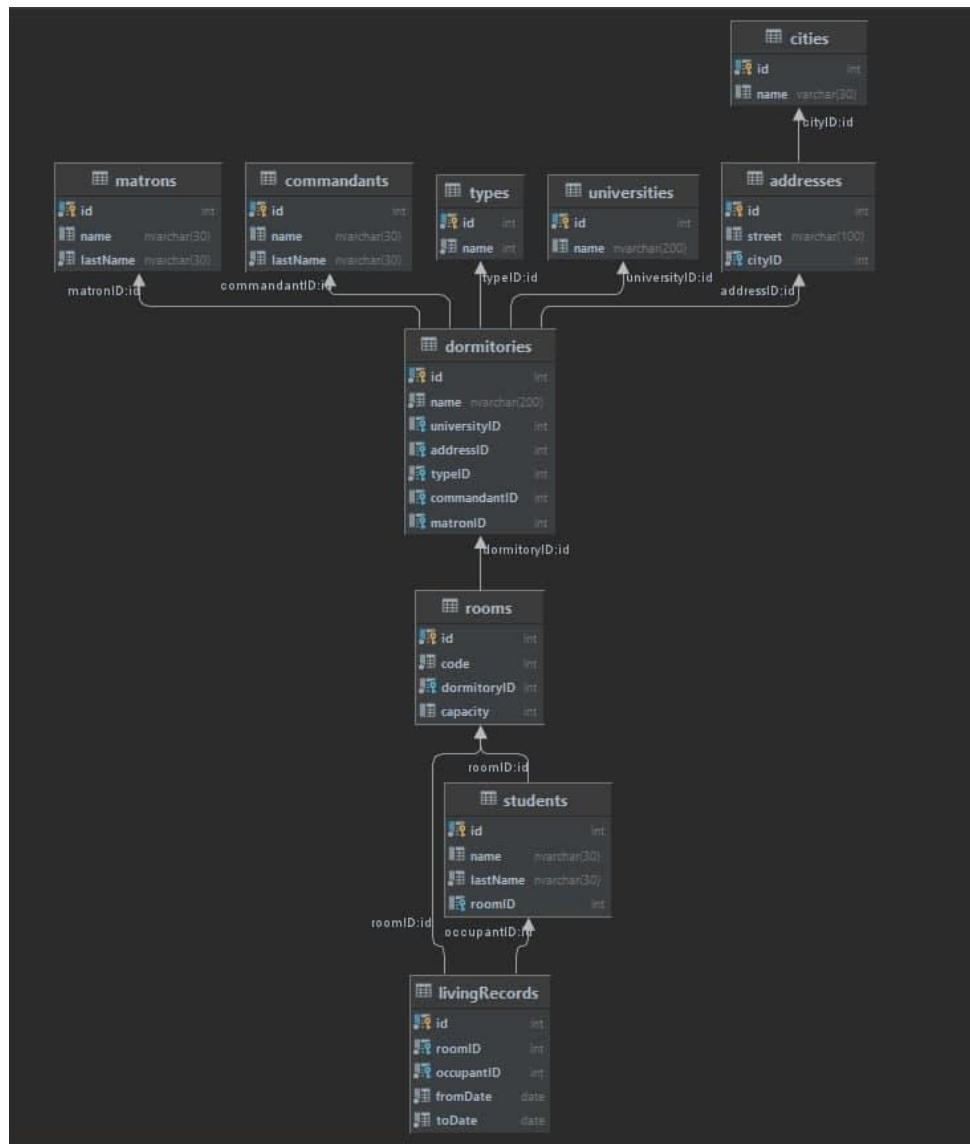


Рисунок 2.5 - діаграма бази даних

#### 2.1.4 Заповнення таблиць бази даних даними

Заповнимо таблицю «Адреси (addresses)» даними

```

set identity_insert addresses off;
INSERT INTO addresses (street,cityID)
VALUES
(N'Yanhela, 20',11),
(N'Yanhela, 5',11),
(N'Yanhela, 22',11),
(N'Yanhela, 7',11),

```

(N'Yanhela, 18/1',11),  
 (N'Metalistiv, 3',11),  
 (N'Oleksy Tykhoho, 2/24',11),  
 (N'Metalistiv, 4',11),  
 (N'Metalistiv, 7',11),  
 (N'Metalistiv, 8',11),  
 (N'prov. Kovalskyy, 5',11),  
 (N'Metalistiv, 5',11),  
 (N'Oleksy Tykhoho, 3',11),  
 (N'Oleksy Tykhoho, 1',11),  
 (N'Borschahivska, 148',11),  
 (N'Borschahivska, 146',11),  
 (N'Borschahivska, 144',11),  
 (N'prov. Kovalskyy, 22a',11),  
 (N'Metalistiv, 6',11);

	id	street	cityID
1	1	Yanhela, 20	11
2	2	Yanhela, 5	11
3	3	Yanhela, 22	11
4	4	Yanhela, 7	11
5	5	Yanhela, 18/1	11
6	6	Metalistiv, 3	11
7	7	Oleksy Tykhoho, 2/24	11
8	8	Metalistiv, 4	11
9	9	Metalistiv, 7	11
10	10	Metalistiv, 8	11
11	11	prov. Kovalskyy, 5	11
12	12	Metalistiv, 5	11
13	13	Oleksy Tykhoho, 3	11
14	14	Oleksy Tykhoho, 1	11
15	15	Borschahivska, 148	11
16	16	Borschahivska, 146	11
17	17	Borschahivska, 144	11
18	18	prov. Kovalskyy, 22a	11
19	19	Metalistiv, 6	11

Рисунок 2.6 - заповнення таблиці адреси

Заповнимо таблицю «Міста(cities)» даними

```
INSERT INTO cities (id, name)
```

```
VALUES
```

```
(1, N'Cherkasy'),
```

```
(2, N'Chernihiv'),
```

```
(3, N'Chernivtsi'),
```

```
(4, N'Dnipro'),
```

```
(5, N'Donetsk'),
```

```
(6, N'Ivano-Frankivsk'),
```

```
(7, N'Kharkiv'),
```

```
(8, N'Kherson'),
```

```
(9, N'Khmelnyskyi'),
```

```
(10, N'Kropyvnytskyi'),
```

```
(11, N'Kyiv'),
```

```
(12, N'Luhansk'),
```

```
(13, N'Luts'),
```

```
(14, N'Lviv'),
```

```
(15, N'Mykolaiv'),
```

```
(16, N'Odessa'),
```

```
(17, N'Poltava'),
```

```
(18, N'Rivne'),
```

```
(19, N'Sumy'),
```

```
(20, N'Ternopil'),
```

```
(21, N'Uzhhorod'),
```

```
(22, N'Vinnytsia'),
```

```
(23, N'Zaporizhzhia'),
```

```
(24, N'Zhytomyr');
```

	id	name
1	1	Cherkasy
2	2	Chernihiv
3	3	Chernivtsi
4	4	Dnipro
5	5	Donetsk
6	6	Ivano-Frankivsk
7	7	Kharkiv
8	8	Kherson
9	9	Khmelnytskyi
10	10	Kropyvnytskyi
11	11	Kyiv
12	12	Luhansk
13	13	Lutsk
14	14	Lviv
15	15	Mykolaiv
16	16	Odessa
17	17	Poltava
18	18	Rivne
19	19	Sumy
20	20	Ternopil
21	21	Uzhhorod
22	22	Vinnytsia
23	23	Zaporizhzhia
24	24	Zhytomyr

Рисунок 2.7 - заповнення таблиці міста

Далі заповнюємо інші таблиці реляційної бази даних «База даних обліку студентських гуртожитків». (Додаток Б – заповнення таблиць даними).

### 2.1.5 Створення збережених функцій

Збережена процедура – це набір команд T-SQL, які зберігається на сервері і представляють собою самостійний об'єкт. Збережені функції існують незалежно від таблиць або інших об'єктів баз даних.

Функція, що виводить повну інформацію про гуртожиток

```
create function fullDormitoryInformation(@id int)
returns table
as
```

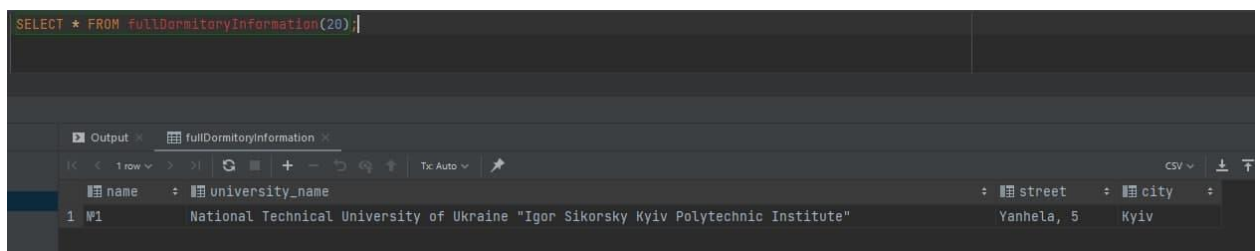
```

return

SELECT d.name, u.name as 'university_name',
       a.street, c.name as 'city'
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.addresses a
    on d.addressID = a.id
LEFT JOIN
    dormitorySchema.cities c
    on a.cityID = c.id
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
WHERE d.id = @id

SELECT * FROM fullDormitoryInformation(20);

```



	name	university_name	street	city
1	№1	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Yanhela, 5	Kyiv

Рисунок 2.8 - повна інформація про гуртожиток

Функція, що виводить всіх жителів обраного гуртожитку, що проживали в ньому в заданий період часу

```

create function getAllOccupantsByDate(@id int, @fromDate date, @toDate
date)
returns table
as
return

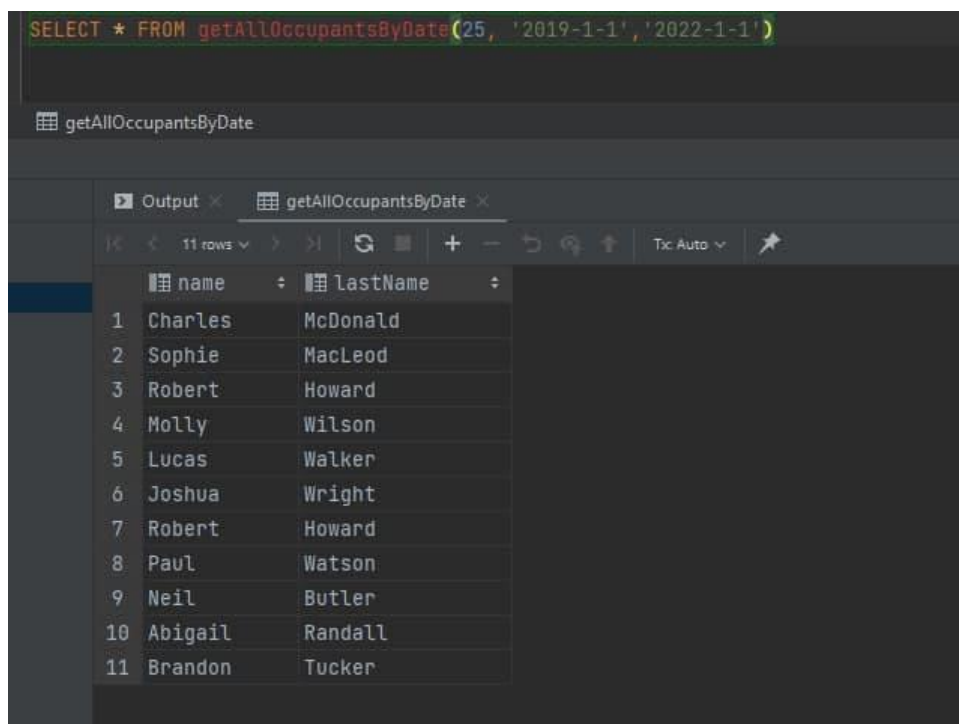
```

```

SELECT
    s.name, s.lastName
FROM
    dormitorySchema.livingRecords lr
LEFT JOIN
    dormitorySchema.students s
    on lr.occupantID = s.id
LEFT JOIN
    dormitorySchema.rooms r
    on lr.roomID = r.id
WHERE
    r.dormitoryID = @id AND
    ((lr.fromDate > @fromDate AND lr.fromDate < @toDate)
    OR
    ((lr.toDate > @fromDate AND lr.toDate < @toDate)))

SELECT * FROM getAllOccupantsByDate(25, '2019-1-1','2022-1-1')

```



The screenshot shows a database query result in a dark-themed IDE. The query is `SELECT * FROM getAllOccupantsByDate(25, '2019-1-1','2022-1-1')`. The result is displayed in a table with 11 rows and 2 columns: `name` and `lastName`. The table is titled `getAllOccupantsByDate` and shows the following data:

	name	lastName
1	Charles	McDonald
2	Sophie	MacLeod
3	Robert	Howard
4	Molly	Wilson
5	Lucas	Walker
6	Joshua	Wright
7	Robert	Howard
8	Paul	Watson
9	Neil	Butler
10	Abigail	Randall
11	Brandon	Tucker

Рисунок 2.9 - всі жителі які проживали в певний час

## 2.1.6 Створення тригерів

Тригер - це такий вид збереженої процедури, яку сервер бази даних автоматично викликає при виконанні операцій модифікації таблиць. Тригери використовуються для перевірки цілісності даних, а також для видачі попереджень. Кожен тригер прив'язується до конкретної таблиці. Але він містить три недоліка: складність, вплив на продуктивність і прихована функціональність.

Тригер, який логує додавання та видалення гуртожитку

```
create trigger InsertAndDeleteDormitoryTrigger
on dormitorySchema.dormitories
after insert, delete
as
begin
    DECLARE @Operation VARCHAR(6)

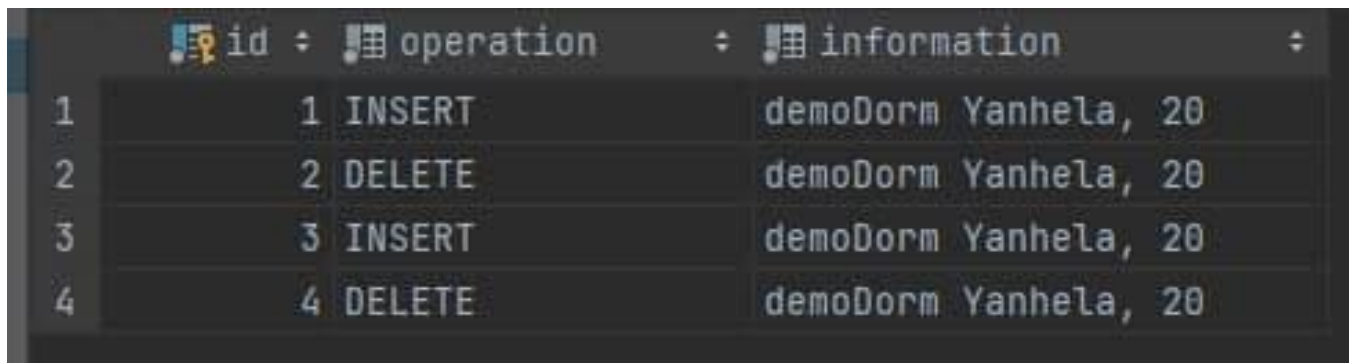
    IF EXISTS(SELECT 0 FROM inserted)
        BEGIN
            SELECT @Operation = 'INSERT'
        END
    ELSE
        BEGIN
            SELECT @Operation = 'DELETE'
        END
    PRINT @Operation
    IF @Operation = 'DELETE'
        insert into dormitorySchema.dormitoriesLogs (operation, information)
        SELECT (select @operation),
            CONCAT(d.name, ' ', a.street)
        FROM
            deleted d
        LEFT JOIN
            dormitorySchema.addresses a
```

```

on d.addressID = a.id;

IF @Operation = 'INSERT'
insert into dormitorySchema.dormitoriesLogs (operation, information)
SELECT (select @operation),
       CONCAT(i.name, ' ', a.street)
FROM
    inserted i
LEFT JOIN
    dormitorySchema.addresses a
    on i.addressID = a.id;
end
GO

```



	id	operation	information
1	1	INSERT	demoDorm Yanhela, 20
2	2	DELETE	demoDorm Yanhela, 20
3	3	INSERT	demoDorm Yanhela, 20
4	4	DELETE	demoDorm Yanhela, 20

Рисунок 2.10 - результат виконання триггеру

### 2.1.7 Створення представлень

Представлення є тимчасовими наборами даних і існують тільки тоді, коли працюють з ними. Представлення - це не копія даних, а посиланням на реальні записи. Вони дозволяють отримувати дані, які задовольняють потреби користувачів. Після створення представлень з ними можна поводитися так само, як і із звичайними таблицями.

Представлення, що містить інформацію про всі гуртожитки Києва



```

CREATE VIEW KyivDormitories
AS
SELECT d.name, u.name as 'university_name', a.street
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.addresses a
    on d.addressID = a.id
LEFT JOIN
    dormitorySchema.cities c
    on a.cityID = c.id
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
WHERE c.name = 'Kyiv'

```

name	university_name	street
№1	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Yanhela, 5
№3	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Yanhela, 22
№4	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Yanhela, 7
№6	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Yanhela, 18/1
№7	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Metallistiv, 3
№8	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Yanhela, 20
№10	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Oleksy Tykhoho, 2/24
№11	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Metallistiv, 4
№12	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Metallistiv, 7
№13	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Metallistiv, 8
№14	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	prov. Kovalskyy, 5
№15	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Metallistiv, 5
№16	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Oleksy Tykhoho, 3
№17	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Oleksy Tykhoho, 1
№18	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Borschahivska, 148
№19	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Borschahivska, 146
№20	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Borschahivska, 144
№21	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	prov. Kovalskyy, 22a
№22	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"	Metallistiv, 6

Рисунок 2.11 - інформація про всі гуртожитки Києва

Представлення, що містить інформацію про всі кімнати 8го гуртожитку КПП:

```

CREATE VIEW dormitory8KPI
AS

```

```
SELECT
    r.code, r.capacity
FROM
    dormitorySchema.rooms r
LEFT JOIN
    dormitorySchema.dormitories d
    on r.dormitoryID = d.id
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
WHERE
    u.name LIKE '%Sikorsky%' AND
    d.name LIKE '%№8%'
```

The screenshot shows a database query interface. At the top, a SQL query is entered: `SELECT * FROM dormitory8KPI;`. Below the query, the results are displayed in a table. The table has two columns: 'code' and 'capacity'. The 'code' column contains values from 101 to 130, and the 'capacity' column contains the value 4 for all rows. The table is numbered 1 to 30 on the left side.

	code	capacity
1	101	4
2	102	4
3	103	4
4	104	4
5	105	4
6	106	4
7	107	4
8	108	4
9	109	4
10	110	4
11	111	4
12	112	4
13	113	4
14	114	4
15	115	4
16	116	4
17	117	4
18	118	4
19	119	4
20	120	4
21	121	4
22	122	4
23	123	4
24	124	4
25	125	4
26	126	4
27	127	4
28	128	4
29	129	4
30	130	4

Рисунок 2.11 - інформація про всі кімнати 8го гуртожитку КПП

### 2.1.8 Створення запитів

Користувача при роботі з базою даних цікавить не весь її вміст, а деяка конкретна інформація. Знайти потрібні відомості можна послідовним переглядом записів. Однак такий спосіб пошуку незручний і малоефективний особливо при великій кількості записів.

Більшість систем управління базами даних дозволяють виконувати вибірку потрібної інформації шляхом виконання запитів. Користувач відповідно до певних правил формулює запит, указуючи, якими критеріями повинна

задовольняти його цікавить інформація, а система виводить записи, що задовольняють запиту.

Використання запитів SQL є одним з найбільш ефективних і універсальних способів вибірки даних з таблиць бази даних.

Ось декілька запитів які були створенні в моїй бд:

Обрати всі гуртожитки Києва(рис. 2.12)

```
SELECT
```

```
    d.name
```

```
FROM
```

```
    dormitorySchema.dormitories d
```

```
LEFT JOIN
```

```
    dormitorySchema.addresses a
```

```
    on d.addressID = a.id
```

```
LEFT JOIN
```

```
    dormitorySchema.cities c
```

```
    on a.cityID = c.id
```

```
WHERE
```

```
    c.name = 'Kyiv';
```

	name
1	№11
2	№1
3	№3
4	№4
5	№6
6	№7
7	№8
8	№10
9	№12
10	№13
11	№14
12	№15
13	№16
14	№17
15	№18
16	№19
17	№20
18	№21
19	№22

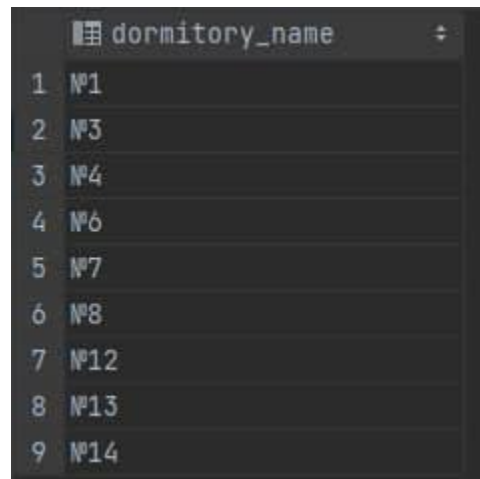
Рисунок 2.12 - результат виконання запиту

Обрати всі гуртожитки КПІ коридорного типу(рис. 2. )

```

SELECT
    d.name as dormitory_name
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
LEFT JOIN
    dormitorySchema.types t
    on d.typeID = t.id
WHERE u.name LIKE '%Sikorsky%' AND t.name = 'corridor'

```



The image shows a screenshot of a database query result. At the top, there is a header bar with the text 'dormitory\_name' and a small icon on the left and a plus/minus icon on the right. Below the header, there is a list of 9 rows, each with a number in the first column and a dormitory name in the second column. The names are: №1, №3, №4, №6, №7, №8, №12, №13, and №14.

	dormitory_name
1	№1
2	№3
3	№4
4	№6
5	№7
6	№8
7	№12
8	№13
9	№14

Рисунок 2.13 - результат виконання запиту

Вивести всю інформацію про гуртожитки (рис. 2. )

SELECT

d.name as dormitory\_name, c.name as city, a.street

FROM

dormitorySchema.dormitories d

LEFT JOIN

dormitorySchema.types t

on d.typeID = t.id

LEFT JOIN

dormitorySchema.universities u

on d.universityID = u.id

LEFT JOIN dormitorySchema.addresses a

on d.addressID = a.id

LEFT JOIN dormitorySchema.cities c

on a.cityID = c.id

	dormitory_name	city	street
1	№1	Kyiv	Yanhela, 5
2	№3	Kyiv	Yanhela, 22
3	№4	Kyiv	Yanhela, 7
4	№6	Kyiv	Yanhela, 18/1
5	№7	Kyiv	Metalistiv, 3
6	№8	Kyiv	Oleksy Tykhoho, 2/24
7	№10	Kyiv	Metalistiv, 4
8	№11	Kyiv	Yanhela, 20
9	№12	Kyiv	Metalistiv, 7
10	№13	Kyiv	Metalistiv, 8
11	№14	Kyiv	prov. Kovalskyy, 5
12	№15	Kyiv	Metalistiv, 5
13	№16	Kyiv	Oleksy Tykhoho, 3
14	№17	Kyiv	Oleksy Tykhoho, 1
15	№18	Kyiv	Borschahivska, 148
16	№19	Kyiv	Borschahivska, 146
17	№20	Kyiv	Borschahivska, 144
18	№21	Kyiv	prov. Kovalskyy, 22a
19	№22	Kyiv	Metalistiv, 6

Рисунок 2.14 - результат виконання запиту

Перелік всіх запитів можна побачити в “Додаток В – запити”.

## ВИСНОВКИ

Метою курсової роботи було проектування бази даних обліку студентських гуртожитків. Для виконання курсової роботи були проведені всі необхідні дослідження, щодо гуртожитків та їх структури. Спочатку ми провели повний аналіз нашої предметної області і тоді вже почали її описувати

Після цього була побудована концептуальна модель для якої ми використовували мову ER-опису предметної області, яка базується на концепції, що інформаційна модель предметної області може бути описана із застосування таких понять, як сутність, атрибут, зв'язок. І ми створили даталогічну модель бази даних. І тільки після цих завдань ми приступили к практичній частині і почали створювати саму базу даних, її таблиць, потім створили діаграму, щоб перевірити цілісність нашої бази даних і коректність. Далі заповняли таблиці бази даних, створювали збережені функції, тригери, представлення, та прописували запити.

В даній курсовій роботі була розроблена база даних «База даних обліку студентських гуртожитків» в системі управління базами даних DataGrip.

Створення бази даних «База даних обліку студентських гуртожитків » є досить актуальним і корисним, бо вона полегшує пошук інформації по гуртожиткам, студентам та кімнатам а також полегшує управління.



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Что такое СУБД – [Електронний ресурс] / Н. А. – 2017. – Режим доступу до ресурсу: <https://brainoteka.com/courses/ms-sql-dlya-nachinayushih/chtotakoe-subd>
2. Введение в MS SQL и T-SQL – [Електронний ресурс] / Н. А. – 2017. – Режим доступу до ресурсу: <https://metanit.com/sql/sqlserver/2.1.php>
3. Что такое базы данных и зачем их изучать? – [Електронний ресурс] / ЧУДО “САИКТ”, 2016. – Режим доступу до ресурсу: <https://saikt-online.ru/chtotakoe-bazy-dannyh-i-zachem-ix-izuchat/>
4. Текст лекцій до розділу “Проектування баз даних” дисципліни “Бази даних” для студ. спец. „Автоматизоване управління технологічними процесами” / Уклад.: Л.Д. Ярошук. – К.НТУУ «КПІ», 2012. - 117 с.
5. Інфологічна модель даних – [Електронний ресурс] / Н. А. – 2015. – Режим доступу до ресурсу: <https://studopedia.info/6-19535.html>
6. Системи управління базами даних. Лекція – [Електронний ресурс] / Н. А. – 2020. – Режим доступу до ресурсу: [https://sites.google.com/a/msk.edu.ua/ikt\\_nmk/1-j-semester/zan11/z11-2](https://sites.google.com/a/msk.edu.ua/ikt_nmk/1-j-semester/zan11/z11-2)
7. Інформаційно-комунікаційне забезпечення фінансової діяльності навчальний посібник/ - Х.Харківський національний економічний університет ім. С. Кузнеця, 2016. - 424 с

## Додатки

### Додаток А - створення таблиць

Створення таблиці Міста(cities)

```
use dormitoryDB
```

```
go
```

```
create table dormitoryDB.dbo.cities
```

```
(
```

```
    id int not null
```

```
        constraint cities_pk
```

```
            primary key nonclustered,
```

```
    name varchar(30)
```

```
)
```

```
go
```

```
create unique index cities_id_uindex
```

```
    on dormitoryDB.dbo.cities (id)
```

```
go
```

Створення таблиці Адреси(addresses)

```
create table dormitoryDB.dbo.addresses
```

```
(
```

```
    id int identity (1, 1)
```

```
        constraint addresses_pk
```

```
            primary key,
```

```
    street nvarchar(100),
```

```
    cityID int not null
```

```
        constraint addresses_cities_id_fk
```

```
            references dbo.cities
```

```
)  
go
```

```
create unique index addresses_cityID_uindex  
on dormitoryDB.dbo.addresses (id)  
go
```

Створення таблиці Університети(universities)

```
create table dormitoryDB.dbo.universities  
(  
    id int identity  
        constraint universities_pk  
        primary key nonclustered,  
    name nvarchar(200)  
)  
go
```

```
create unique index universities_id_uindex  
on dormitoryDB.dbo.universities (id)  
go
```

Створення таблиці Типи гуртожитків(types)

```
create table dormitoryDB.dbo.types  
(  
    id int identity  
        constraint types_pk  
        primary key nonclustered,  
    name nvarchar(30) not null  
)  
go
```

```
create unique index types_id_uindex
    on dormitoryDB.dbo.types (id)
go
```

Створення таблиці Коменданти(commandants)

```
create table dormitoryDB.dbo.commandants
(
    id      int identity
        constraint commandant_pk
            primary key nonclustered,
    name    nvarchar(30),
    lastName nvarchar(30) not null
)
go
```

```
create unique index commandant_id_uindex
    on dormitoryDB.dbo.commandants (id)
go
```

Створення таблиці Завідуючі(matrons)

```
create table matrons
(
    id      int identity
        constraint matrons_pk
            primary key nonclustered,
    name    nvarchar(30),
    lastName nvarchar(30) not null
)
go
```

```
create unique index matrons_id_uindex
    on dormitoryDB.dbo.matrons (id)
go
```

Створення таблиці Гуртожитки(dormitories)

```
create table dormitoryDB.dbo.dormitories
(
    id          int identity
        constraint dormitories_pk
            primary key nonclustered,
    name        nvarchar(200) not null,
    universityID int
        constraint dormitories_universities_id_fk
            references dormitoryDB.dbo.universities,
    addressID   int
        constraint dormitories_addresses_id_fk
            references dormitoryDB.dbo.addresses,
    typeID      int          not null
        constraint dormitories_types_id_fk
            references dormitoryDB.dbo.types,
    commandantID int
        constraint dormitories_commandants_id_fk
            references dormitoryDB.dbo.commandants,
    matronID    int
        constraint dormitories_matrons_id_fk
            references dormitoryDB.dbo.matrons
)
go
```

```
create unique index dormitories_id_uindex
    on dormitoryDB.dbo.dormitories (id)
go
```

Створення таблиці Кімнати(rooms)

```
create table dormitoryDB.dbo.rooms
(
    id          int identity
        constraint rooms_pk
            primary key nonclustered,
    code        int not null,
    dormitoryID int not null
        constraint rooms_dormitories_id_fk
            references dormitoryDB.dbo.dormitories,
    capacity    int
)
go
```

```
create unique index rooms_id_uindex
    on dormitoryDB.dbo.rooms (id)
go
```

Створення таблиці Студенти(students)

```
create table students
(
    id          int identity
        constraint students_pk
            primary key nonclustered,
    name        nvarchar(30),
    lastName    nvarchar(30) not null,
    roomID      int
        constraint students_rooms_id_fk
            references dormitoryDB.dbo.rooms
)
go
```

```
go
```

```
create unique index students_id_uindex  
on dormitoryDB.dbo.students (id)
```

```
go
```

Створення таблиці Термін проживання(livingRecords)

```
create table dormitoryDB.dbo.livingRecords  
(  
    id      int identity  
        constraint livingRecords_pk  
        primary key nonclustered,  
    roomID  int not null  
        constraint livingRecords_rooms_id_fk  
        references dormitoryDB.dbo.rooms,  
    occupantID int not null  
        constraint livingRecords_students_id_fk  
        references dormitoryDB.dbo.students,  
    fromDate date not null,  
    toDate   date not null  
)
```

```
go
```

```
create unique index livingRecords_id_uindex  
on dormitoryDB.dbo.livingRecords (id)
```

```
go
```

## Додаток Б - заповнення таблиць даними

-- Заповнення таблиці «Міста(cities)» даними

```
INSERT INTO cities (id, name)
```

```
VALUES
```

```
(1, N'Cherkasy'),  
(2, N'Chernihiv'),  
(3, N'Chernivtsi'),  
(4, N'Dnipro'),  
(5, N'Donetsk'),  
(6, N'Ivano-Frankivsk'),  
(7, N'Kharkiv'),  
(8, N'Kherson'),  
(9, N'Khmelnyskyi'),  
(10, N'Kropyvnytskyi'),  
(11, N'Kyiv'),  
(12, N'Luhansk'),  
(13, N'Lutsk'),  
(14, N'Lviv'),  
(15, N'Mykolaiv'),  
(16, N'Odessa'),  
(17, N'Poltava'),  
(18, N'Rivne'),  
(19, N'Sumy'),  
(20, N'Ternopil'),  
(21, N'Uzhhorod'),  
(22, N'Vinnytsia'),  
(23, N'Zaporizhzhia'),  
(24, N'Zhytomyr');
```

-- Заповнення таблиці «Університети(universities)» даними



```
set identity_insert universities off;
INSERT INTO universities (name)
VALUES
(N'National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic
Institute"'),
(N'Taras Shevchenko National University of Kyiv'),
(N'National Aviation University '),
(N'Mykolayiv National Agrarian University'),
(N'Petro Mohyla Black Sea National University'),
(N'Kharkiv National University of Radio Electronics'),
(N'Kharkiv Polytechnic Institute'),
(N'Lviv Polytechnic National University '),
(N'University of Lviv'),
(N'Donetsk State University of Management');
```

-- Заповнення таблиці «Типи гуртожитків(types)» даними

```
INSERT INTO types (name)
VALUES
(N'block'),
(N'corridor'),
(N'mixed'),
(N'family');
```

-- Заповнення таблиці «Коменданти (commandants)» даними

```
set identity_insert commandants off;
INSERT INTO commandants (name, lastName)
VALUES
(N'Svitlana', N'Shtunder'),
```

(N'Maryna', N'Krasnokutska'),  
(N'Svitlana', N'Morhun'),  
(N'Mykyta ', N'Moshenchenko'),  
(N'Inna', N'Zamana'),  
(N'Oksana', N'Potrats'),  
(N'Vita', N'Romanenko'),  
(N'Iryna', N'Kyseliova'),  
(N'Maria ', N'Moskalets'),  
(N'Oksana', N'Sapronova'),  
(N'Olha', N'Pimanova'),  
(N'Tetiana', N'Levchenko'),  
(N'Nina', N'Klimova'),  
(N'Yulia', N'Budachevska'),  
(N'Natalia', N'Yarosh'),  
(N'Nadia', N'Nosanchuk');

--Заповнення таблиці « Завідувач(matrons)» даними

```
set identity_insert matrons off;  
INSERT INTO matrons (name, lastName)  
VALUES  
(N'Liubov', N'Filipishyna'),  
(N'Svitlana ', N'Zavalska'),  
(N'Olena', N'Stromenko'),  
(N'Svitlana', N'Kryvenko'),  
(N'Larysa', N'Hurovska'),  
(N'Nadiya', N'Tymoshenko'),  
(N'Olha', N'Blazhko'),  
(N'Tetiana', N'Nedashkivska'),  
(N'Svitlana', N'Marynka'),  
(N'Liubov', N'Koshyl'),  
(N'Olena', N'Shevchuk'),  
(N'Larysa', N'Pidipryhora'),
```

(N'Liudmyla', N'Scherbyna'),  
(N'Vira', N'Bachynska'),  
(N'Dmytro', N'Kulyk'),  
(N'Iryna', N'Postolatyeva'),  
(N'Nataliya', N'Puzyrna'),  
(N'Yevheniya', N'Maslova'),  
(N'Nataliya', N'Vonsovykh');

-- Заповнення таблиці «Адреси (addresses)» даними

```
set identity_insert addresses off;  
INSERT INTO addresses (street,cityID)  
VALUES  
(N'Yanhela, 20',11),  
(N'Yanhela, 5',11),  
(N'Yanhela, 22',11),  
(N'Yanhela, 7',11),  
(N'Yanhela, 18/1',11),  
(N'Metalistiv, 3',11),  
(N'Oleksy Tykhoho, 2/24',11),  
(N'Metalistiv, 4',11),  
(N'Metalistiv, 7',11),  
(N'Metalistiv, 8',11),  
(N'prov. Kovalskyy, 5',11),  
(N'Metalistiv, 5',11),  
(N'Oleksy Tykhoho, 3',11),  
(N'Oleksy Tykhoho, 1',11),  
(N'Borschahivska, 148',11),  
(N'Borschahivska, 146',11),  
(N'Borschahivska, 144',11),  
(N'prov. Kovalskyy, 22a',11),  
(N'Metalistiv, 6',11);
```

-- Заповнення таблиці «Гуртожитки(dormitories)» даними

```
set identity_insert dormitories off;
```

```
INSERT INTO dormitories (name, universityID, addressID, typeID, commandantID,  
matronID)
```

```
VALUES
```

```
(N'№1', 1, 2, 2, 1, 1),
```

```
(N'№3', 1, 3, 2, 2, 2),
```

```
(N'№4', 1, 4, 2, 3, 3),
```

```
(N'№6', 1, 5, 2, 4, 4),
```

```
(N'№7', 1, 6, 2, 5, 5),
```

```
(N'№8', 1, 1, 2, 6, 6),
```

```
(N'№10', 1, 7, 1, 7, 7),
```

```
(N'№11', 1, 8, 3, null, 8),
```

```
(N'№12', 1, 9, 2, 8, 9),
```

```
(N'№13', 1, 10, 2, 9, 10),
```

```
(N'№14', 1, 11, 2, 10, 11),
```

```
(N'№15', 1, 12, 1, 11, 12),
```

```
(N'№16', 1, 13, 1, 12, 13),
```

```
(N'№17', 1, 14, 3, 13, 14),
```

```
(N'№18', 1, 15, 1, 14, 15),
```

```
(N'№19', 1, 16, 1, 15, 16),
```

```
(N'№20', 1, 17, 1, 16, 17),
```

```
(N'№21', 1, 18, 4, null, 18),
```

```
(N'№22', 1, 19, 4, null, 19);
```

-- Заповнення таблиці «Кімнати(rooms)» даними

```
set identity_insert rooms off;
```

```
INSERT INTO rooms (code, dormitoryID, capacity)
```

```
VALUES
```

```
(101, 19, 4),
```

```
(102, 19, 4),
```

```
(103, 19, 4),
```

```
(104, 19, 4),
```

```
(105, 19, 4),
```

```
(106, 19, 4),
```

```
(107, 19, 4),
```

```
(108, 19, 4),
```

```
(109, 19, 4),
```

```
(110, 19, 4),
```

```
(111, 19, 4),
```

```
(112, 19, 4),
```

```
(113, 19, 4),
```

```
(114, 19, 4),
```

```
(115, 19, 4),
```

```
(116, 19, 4),
```

```
(117, 19, 4),
```

```
(118, 19, 4),
```

```
(119, 19, 4),
```

```
(120, 19, 4),
```

```
(121, 19, 4),
```

```
(122, 19, 4),
```

```
(123, 19, 4),
```

```
(124, 19, 4),
```

```
(125, 19, 4),
```

```
(126, 19, 4),
```

```
(127, 19, 4),
```

```
(128, 19, 4),
```

```
(129, 19, 4),
```

```
(130, 19, 4),
```

```
(131, 19, 4),
```

(132, 19, 4),  
(133, 19, 4),  
(134, 19, 4),  
(135, 19, 4),  
(136, 19, 4),  
(137, 19, 4),  
(138, 19, 4),  
(139, 19, 4),  
(140, 19, 4),  
(201, 19, 4),  
(202, 19, 4),  
(203, 19, 4),  
(204, 19, 4),  
(205, 19, 4),  
(206, 19, 4),  
(207, 19, 4),  
(208, 19, 4),  
(209, 19, 4),  
(210, 19, 4),  
(211, 19, 4),  
(212, 19, 4),  
(213, 19, 4),  
(214, 19, 4),  
(215, 19, 4),  
(216, 19, 4),  
(217, 19, 4),  
(218, 19, 4),  
(219, 19, 4),  
(220, 19, 4),  
(221, 19, 4),  
(222, 19, 4),  
(223, 19, 4),  
(224, 19, 4),

(225, 19, 4),  
(226, 19, 4),  
(227, 19, 4),  
(228, 19, 4),  
(229, 19, 4),  
(230, 19, 4),  
(231, 19, 4),  
(232, 19, 4),  
(233, 19, 4),  
(234, 19, 4),  
(235, 19, 4),  
(236, 19, 4),  
(237, 19, 4),  
(238, 19, 4),  
(239, 19, 4),  
(240, 19, 4),  
(301, 19, 4),  
(302, 19, 4),  
(303, 19, 4),  
(304, 19, 4),  
(305, 19, 4),  
(306, 19, 4),  
(307, 19, 4),  
(308, 19, 4),  
(309, 19, 4),  
(310, 19, 4),  
(311, 19, 4),  
(312, 19, 4),  
(313, 19, 4),  
(314, 19, 4),  
(315, 19, 4),  
(316, 19, 4),  
(317, 19, 4),

(318, 19, 4),  
(319, 19, 4),  
(320, 19, 4),  
(321, 19, 4),  
(322, 19, 4),  
(323, 19, 4),  
(324, 19, 4),  
(325, 19, 4),  
(326, 19, 4),  
(327, 19, 4),  
(328, 19, 4),  
(329, 19, 4),  
(330, 19, 4),  
(331, 19, 4),  
(332, 19, 4),  
(333, 19, 4),  
(334, 19, 4),  
(335, 19, 4),  
(336, 19, 4),  
(337, 19, 4),  
(338, 19, 4),  
(339, 19, 4),  
(340, 19, 4),  
(401, 19, 4),  
(402, 19, 4),  
(403, 19, 4),  
(404, 19, 4),  
(405, 19, 4),  
(406, 19, 4),  
(407, 19, 4),  
(408, 19, 4),  
(409, 19, 4),  
(410, 19, 4),



(411, 19, 4),  
(412, 19, 4),  
(413, 19, 4),  
(414, 19, 4),  
(415, 19, 4),  
(416, 19, 4),  
(417, 19, 4),  
(418, 19, 4),  
(419, 19, 4),  
(420, 19, 4),  
(421, 19, 4),  
(422, 19, 4),  
(423, 19, 4),  
(424, 19, 4),  
(425, 19, 4),  
(426, 19, 4),  
(427, 19, 4),  
(428, 19, 4),  
(429, 19, 4),  
(430, 19, 4),  
(431, 19, 4),  
(432, 19, 4),  
(433, 19, 4),  
(434, 19, 4),  
(435, 19, 4),  
(436, 19, 4),  
(437, 19, 4),  
(438, 19, 4),  
(439, 19, 4),  
(440, 19, 4),  
(501, 19, 4),  
(502, 19, 4),  
(503, 19, 4),

(504, 19, 4),  
(505, 19, 4),  
(506, 19, 4),  
(507, 19, 4),  
(508, 19, 4),  
(509, 19, 4),  
(510, 19, 4),  
(511, 19, 4),  
(512, 19, 4),  
(513, 19, 4),  
(514, 19, 4),  
(515, 19, 4),  
(516, 19, 4),  
(517, 19, 4),  
(518, 19, 4),  
(519, 19, 4),  
(520, 19, 4),  
(521, 19, 4),  
(522, 19, 4),  
(523, 19, 4),  
(524, 19, 4),  
(525, 19, 4),  
(526, 19, 4),  
(527, 19, 4),  
(528, 19, 4),  
(529, 19, 4),  
(530, 19, 4),  
(531, 19, 4),  
(532, 19, 4),  
(533, 19, 4),  
(534, 19, 4),  
(535, 19, 4),  
(536, 19, 4),

(537, 19, 4),  
(538, 19, 4),  
(539, 19, 4),  
(540, 19, 4);

Заповнення таблиці «Студенти(students)» даними

```
set identity_insert students on;  
INSERT INTO students (id, name, lastName, roomID)  
VALUES  
(1, N'Robert', N'Howard', 178),  
(2, N'Lucas', N'Walker', 51),  
(3, N'Brandon', N'Tucker', 120),  
(4, N'Anna', N'Ince', 195),  
(5, N'Amy', N'Lawrence', 137),  
(6, N'Lauren', N'Murray', 62),  
(7, N'Adrian', N'Knox', 160),  
(8, N'Joshua', N'Wright', 78),  
(9, N'James', N'Gibson', 126),  
(10, N'Theresa', N'Greene', 24),  
(11, N'Jasmine', N'Ross', 32),  
(12, N'Gordon', N'Burgess', 87),  
(13, N'Stewart', N'Parr', 110),  
(14, N'Brian', N'Nolan', 111),  
(15, N'Yvonne', N'Russell', 153),  
(16, N'Amy', N'Fraser', 74),  
(17, N'Joe', N'Thomson', 176),  
(18, N'Amy', N'Mackenzie', 191),  
(19, N'Jacob', N'Walker', 45),  
(20, N'Warren', N'White', 112),  
(21, N'Blake', N'Springer', 197),  
(22, N'Penelope', N'Peters', 70),
```

(23, N'Brian', N'Sanderson', 179),  
(24, N'Lily', N'Bailey', 125),  
(25, N'Lillian', N'Springer', 20),  
(26, N'Abigail', N'Randall', 146),  
(27, N'Paul', N'Watson', 150),  
(28, N'Anne', N'Berry', 131),  
(29, N'Rose', N'MacDonald', 174),  
(30, N'Bernadette', N'Bailey', 108),  
(31, N'Sam', N'Taylor', 8),  
(32, N'Chloe', N'Campbell', 135),  
(33, N'Chloe', N'Slater', 15),  
(34, N'Stephanie', N'Churchill', 197),  
(35, N'Deirdre', N'Ince', 7),  
(36, N'Boris', N'Rutherford', 181),  
(37, N'Lisa', N'Stewart', 41),  
(38, N'Emma', N'Mackenzie', 3),  
(39, N'Diane', N'Burgess', 128),  
(40, N'Victoria', N'Hill', 131),  
(41, N'Andrea', N'Brown', 142),  
(42, N'Michelle', N'Dickens', 32),  
(43, N'Felicity', N'Graham', 77),  
(44, N'Benjamin', N'Gray', 155),  
(45, N'Deirdre', N'Metcalf', 141),  
(46, N'Neil', N'Butler', 69),  
(47, N'Victor', N'Sharp', 16),  
(48, N'Diane', N'Bell', 114),  
(49, N'Molly', N'Wilson', 122),  
(50, N'Penelope', N'Grant', 183),  
(51, N'Michael', N'Miller', 60),  
(52, N'Dylan', N'Wilson', 172),  
(53, N'Sam', N'Berry', 162),  
(54, N'Sophie', N'MacLeod', 155),  
(55, N'Joe', N'Wright', 176),

(56, N'Leonard', N'Miller', 190),  
(57, N'Rose', N'Hudson', 133),  
(58, N'Diana', N'Dowd', 56),  
(59, N'Jessica', N'Piper', 124),  
(60, N'Melanie', N'Anderson', 92),  
(61, N'Lucas', N'Buckland', 64),  
(62, N'Steven', N'MacDonald', 128),  
(63, N'Melanie', N'Simpson', 144),  
(64, N'Sean', N'Sharp', 151),  
(65, N'Stephen', N'Gray', 32),  
(66, N'Luke', N'Murray', 107),  
(67, N'Boris', N'Bailey', 121),  
(68, N'Jacob', N'Springer', 186),  
(69, N'Evan', N'Graham', 26),  
(70, N'Andrea', N'Lewis', 5),  
(71, N'Edward', N'Fraser', 179),  
(72, N'Simon', N'Newman', 186),  
(73, N'Ruth', N'Hardacre', 111),  
(74, N'Dan', N'Metcalf', 128),  
(75, N'Matt', N'Jackson', 177),  
(76, N'Jake', N'Peake', 89),  
(77, N'Matt', N'Martin', 142),  
(78, N'Cameron', N'Baker', 166),  
(79, N'Jonathan', N'Jackson', 10),  
(80, N'Virginia', N'Kerr', 166),  
(81, N'Alexander', N'Bond', 198),  
(82, N'Owen', N'Ball', 185),  
(83, N'Christian', N'McLean', 40),  
(84, N'Piers', N'Alsop', 180),  
(85, N'Sonia', N'Mitchell', 61),  
(86, N'Alexandra', N'Jackson', 86),  
(87, N'Gordon', N'Bell', 187),  
(88, N'Eric', N'North', 83),

(89, N'Jennifer', N'Short', 34),  
(90, N'Kylie', N'Hughes', 50),  
(91, N'Leonard', N'Reid', 178),  
(92, N'David', N'Baker', 83),  
(93, N'Charles', N'McDonald', 184),  
(94, N'Anna', N'Thomson', 52),  
(95, N'Ella', N'Fraser', 97),  
(96, N'Nathan', N'Davies', 177),  
(97, N'Liam', N'Skinner', 144),  
(98, N'Jonathan', N'Russell', 106);

Заповнення таблиці «Студенти(students)» даними

```
set identity_insert students off;
```

```
INSERT INTO students (name, lastName, roomID)
```

```
VALUES
```

```
(N'Robert', N'Howard', 178),  
(N'Lucas', N'Walker', 51),  
(N'Brandon', N'Tucker', 120),  
(N'Anna', N'Ince', 195),  
(N'Amy', N'Lawrence', 137),  
(N'Lauren', N'Murray', 62),  
(N'Adrian', N'Knox', 160),  
(N'Joshua', N'Wright', 78),  
(N'James', N'Gibson', 126),  
(N'Theresa', N'Greene', 24),  
(N'Jasmine', N'Ross', 32),  
( N'Gordon', N'Burgess', 87),  
( N'Stewart', N'Parr', 110),  
( N'Brian', N'Nolan', 111),  
( N'Yvonne', N'Russell', 153),  
( N'Amy', N'Fraser', 74),  
( N'Joe', N'Thomson', 176),
```

( N'Amy', N'Mackenzie', 191),  
( N'Jacob', N'Walker', 45),  
( N'Warren', N'White', 112),  
(N'Blake', N'Springer', 197),  
(N'Penelope', N'Peters', 70),  
( N'Brian', N'Sanderson', 179),  
(N'Lily', N'Bailey', 125),  
(N'Lillian', N'Springer', 20),  
(N'Abigail', N'Randall', 146),  
( N'Paul', N'Watson', 150),  
(N'Anne', N'Berry', 131),  
( N'Rose', N'MacDonald', 174),  
( N'Bernadette', N'Bailey', 108),  
( N'Sam', N'Taylor', 8),  
(N'Chloe', N'Campbell', 135),  
( N'Chloe', N'Slater', 15),  
(N'Stephanie', N'Churchill', 197),  
( N'Deirdre', N'Ince', 7),  
( N'Boris', N'Rutherford', 181),  
( N'Lisa', N'Stewart', 41),  
( N'Emma', N'Mackenzie', 3),  
(N'Diane', N'Burgess', 128),  
(N'Victoria', N'Hill', 131),  
(N'Andrea', N'Brown', 142),  
(N'Michelle', N'Dickens', 32),  
( N'Felicity', N'Graham', 77),  
(N'Benjamin', N'Gray', 155),  
(N'Deirdre', N'Metcalf', 141),  
( N'Neil', N'Butler', 69),  
( N'Victor', N'Sharp', 16),  
( N'Diane', N'Bell', 114),  
( N'Molly', N'Wilson', 122),  
( N'Penelope', N'Grant', 183),

( N'Michael', N'Miller', 60),  
(N'Dylan', N'Wilson', 172),  
(N'Sam', N'Berry', 162),  
(N'Sophie', N'MacLeod', 155),  
( N'Joe', N'Wright', 176),  
(N'Leonard', N'Miller', 190),  
( N'Rose', N'Hudson', 133),  
(N'Diana', N'Dowd', 56),  
( N'Jessica', N'Piper', 124),  
( N'Melanie', N'Anderson', 92),  
(N'Steven', N'MacDonald', 128),  
(N'Melanie', N'Simpson', 144),  
(N'Sean', N'Sharp', 151),  
(N'Stephen', N'Gray', 32),  
(N'Luke', N'Murray', 107),  
(N'Boris', N'Bailey', 121),  
(N'Jacob', N'Springer', 186),  
(N'Evan', N'Graham', 26),  
(N'Andrea', N'Lewis', 5),  
(N'Edward', N'Fraser', 179),  
(N'Simon', N'Newman', 186),  
(N'Ruth', N'Hardacre', 111),  
(N'Dan', N'Metcalfe', 128),  
(N'Matt', N'Jackson', 177),  
(N'Jake', N'Peake', 89),  
(N'Matt', N'Martin', 142),  
(N'Cameron', N'Baker', 166),  
(N'Jonathan', N'Jackson', 10),  
(N'Virginia', N'Kerr', 166),  
(N'Alexander', N'Bond', 198),  
(N'Owen', N'Ball', 185),  
(N'Christian', N'McLean', 40),  
(N'Piers', N'Alsop', 180),



(N'Sonia', N'Mitchell', 61),  
(N'Alexandra', N'Jackson', 86),  
(N'Gordon', N'Bell', 187),  
(N'Eric', N'North', 83),  
(N'Jennifer', N'Short', 34),  
(N'Kylie', N'Hughes', 50),  
(N'Leonard', N'Reid', 178),  
(N'David', N'Baker', 83),  
(N'Charles', N'McDonald', 184),  
(N'Anna', N'Thomson', 52),  
(N'Ella', N'Fraser', 97),  
(N'Nathan', N'Davies', 177),  
(N'Liam', N'Skinner', 144),  
(N'Jonathan', N'Russell', 106);

Заповнення таблиці «Дані проживання(livingRecords)» даними

INSERT INTO livingRecords (roomID, occupantID, fromDate, toDate)

VALUES

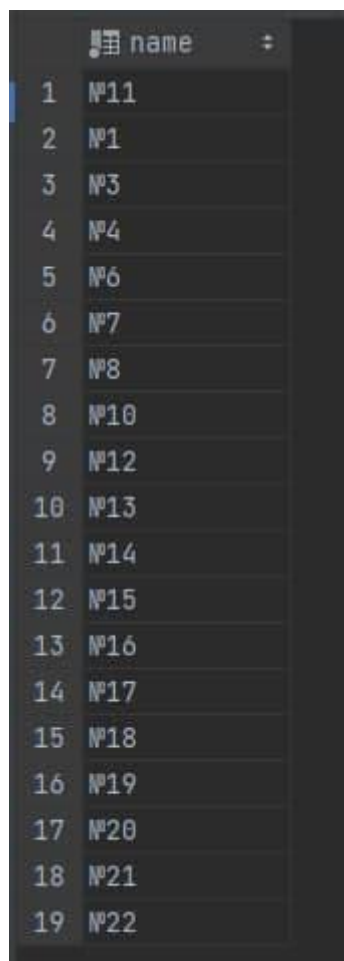
(194, 31, N'2015-11-04', N'2015-12-04'),  
(131, 86, N'2016-03-31', N'2016-08-19'),  
(80, 57, N'2016-05-02', N'2016-09-04'),  
(70, 88, N'2016-09-04', N'2016-12-25'),  
(125, 82, N'2016-09-28', N'2017-02-04'),  
(3, 7, N'2016-10-15', N'2017-06-03'),  
(146, 31, N'2017-05-19', N'2017-07-01'),  
(11, 6, N'2017-08-28', N'2017-07-31'),  
(174, 65, N'2018-04-05', N'2018-05-05'),  
(188, 66, N'2018-06-13', N'2018-07-13'),  
(68, 20, N'2018-07-02', N'2018-08-02'),  
(163, 44, N'2018-07-17', N'2018-08-08'),

(71, 92, N'2018-08-29', N'2018-11-07'),  
(63, 93, N'2018-11-18', N'2019-01-27'),  
(30, 54, N'2019-02-25', N'2019-03-04'),  
(94, 1, N'2019-05-13', N'2019-10-04'),  
(185, 49, N'2019-06-12', N'2019-10-14'),  
(199, 2, N'2019-07-06', N'2019-10-28'),  
(42, 8, N'2019-12-15', N'2019-12-31'),  
(94, 1, N'2020-01-22', N'2020-03-24'),  
(96, 27, N'2020-06-05', N'2020-07-24'),  
(98, 46, N'2020-09-09', N'2020-10-09'),  
(125, 26, N'2021-07-16', N'2021-08-16'),  
(137, 3, N'2021-12-15', N'2022-01-24'),  
(111, 93, N'2022-03-20', N'2022-04-20');

## Додаток В - запити

Обрати всі гуртожитки Києва

```
SELECT
    d.name
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.addresses a
    on d.addressID = a.id
LEFT JOIN
    dormitorySchema.cities c
    on a.cityID = c.id
WHERE
    c.name = 'Kyiv';
```

A screenshot of a database query result displayed in a dark-themed application. The result is a table with two columns: an index from 1 to 19 and a column labeled 'name'. The names are listed as '№11', '№1', '№3', '№4', '№6', '№7', '№8', '№10', '№12', '№13', '№14', '№15', '№16', '№17', '№18', '№19', '№20', '№21', and '№22'.

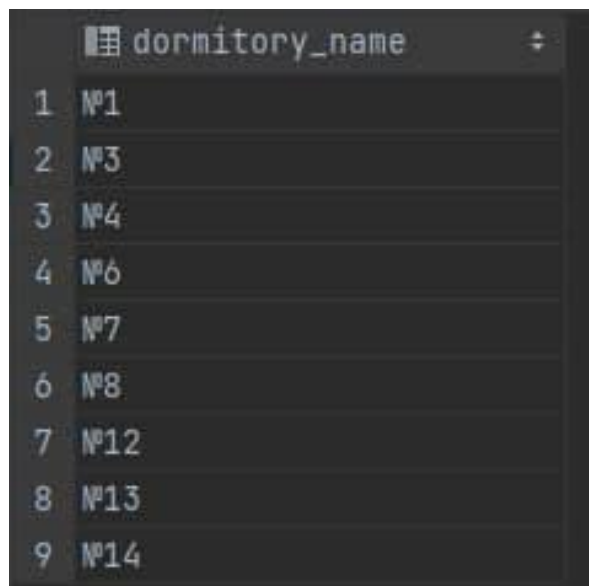
	name
1	№11
2	№1
3	№3
4	№4
5	№6
6	№7
7	№8
8	№10
9	№12
10	№13
11	№14
12	№15
13	№16
14	№17
15	№18
16	№19
17	№20
18	№21
19	№22

Обрати всі гуртожитки КПІ коридорного типу

```

SELECT
    d.name as dormitory_name
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
LEFT JOIN
    dormitorySchema.types t
    on d.typeID = t.id
WHERE u.name LIKE '%Sikorsky%' AND t.name = 'corridor'

```



The screenshot shows a database query result in a dark-themed interface. The title bar of the window is labeled 'dormitory\_name'. The result is a list of 9 rows, each with an index and a dormitory name. The names are: №1, №3, №4, №6, №7, №8, №12, №13, and №14.

Index	dormitory_name
1	№1
2	№3
3	№4
4	№6
5	№7
6	№8
7	№12
8	№13
9	№14

Вивести всю інформацію про гуртожитки

```

SELECT
    d.name as dormitory_name, c.name as city, a.street
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.types t
    on d.typeID = t.id

```

LEFT JOIN

dormitorySchema.universities u

on d.universityID = u.id

LEFT JOIN dormitorySchema.addresses a

on d.addressID = a.id

LEFT JOIN dormitorySchema.cities c

on a.cityID = c.id

	dormitory_name	city	street
1	№1	Kyiv	Yanhela, 5
2	№3	Kyiv	Yanhela, 22
3	№4	Kyiv	Yanhela, 7
4	№6	Kyiv	Yanhela, 18/1
5	№7	Kyiv	Metalistiv, 3
6	№8	Kyiv	Oleksy Tykhoho, 2/24
7	№10	Kyiv	Metalistiv, 4
8	№11	Kyiv	Yanhela, 20
9	№12	Kyiv	Metalistiv, 7
10	№13	Kyiv	Metalistiv, 8
11	№14	Kyiv	prov. Kovalskyy, 5
12	№15	Kyiv	Metalistiv, 5
13	№16	Kyiv	Oleksy Tykhoho, 3
14	№17	Kyiv	Oleksy Tykhoho, 1
15	№18	Kyiv	Borschahivska, 148
16	№19	Kyiv	Borschahivska, 146
17	№20	Kyiv	Borschahivska, 144
18	№21	Kyiv	prov. Kovalskyy, 22a
19	№22	Kyiv	Metalistiv, 6

Вивести кількість гуртожитків Києва

SELECT

COUNT(1) as count\_of\_dormitories

FROM

dormitorySchema.dormitories d

LEFT JOIN

dormitorySchema.addresses a

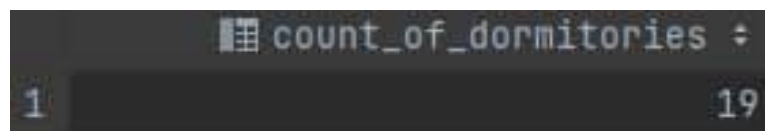
on d.addressID = a.id

LEFT JOIN

```

    dormitorySchema.cities c
  on a.cityID = c.id
WHERE
  c.name = 'Kyiv'
GROUP BY
  c.name

```



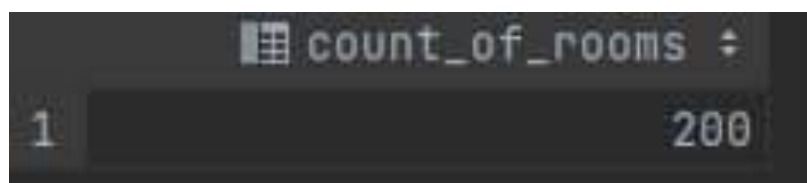
	count_of_dormitories
1	19

Вивести кількість кімнат 8го гуртожитку КПІ

```

SELECT
  COUNT(1) as count_of_rooms
FROM
  dormitorySchema.dormitories d
LEFT JOIN
  dormitorySchema.universities u
  on d.universityID = u.id
RIGHT JOIN
  dormitorySchema.rooms r
  on d.id = r.dormitoryID
WHERE u.name LIKE '%Sikorsky%' AND d.name LIKE '№8'
GROUP BY
  d.name

```



	count_of_rooms
1	200

Вивести завідуючих та гуртожитки, в яких вони завідують

```

SELECT
    m.name, m.lastName, d.name as dormitory
FROM
    dormitorySchema.matrons m
LEFT JOIN
    dormitorySchema.dormitories d
    on m.id = d.matronID

```

	name	lastName	dormitory
1	Liubov	Filipishyna	№1
2	Svitlana	Zavalska	№3
3	Olena	Stromenko	№4
4	Svitlana	Kryvenko	№6
5	Larysa	Hurovska	№7
6	Nadiya	Tymoshenko	№8
7	Olha	Blazhko	№10
8	Tetiana	Nedashkivska	№11
9	Svitlana	Marynka	№12
10	Liubov	Koshyl	№13
11	Olena	Shevchuk	№14
12	Larysa	Pidipryhora	№15
13	Liudmyla	Scherbyna	№16
14	Vira	Bachynska	№17
15	Dmytro	Kulyk	№18
16	Iryna	Postolatyeva	№19
17	Nataliya	Puzyrna	№20
18	Yevheniya	Maslova	№21
19	Nataliya	Vonsovych	№22

Вивести комендата бго гуртожитку КПІ

```

SELECT
    c.name, c.lastName
FROM
    dormitorySchema.commandants c
RIGHT JOIN
    dormitorySchema.dormitories d
    on c.id = d.commandantID

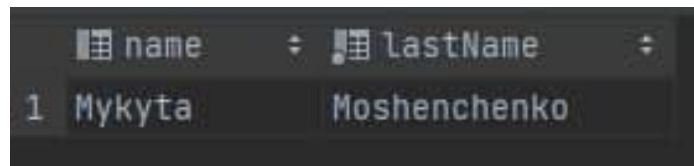
```

LEFT JOIN

dormitorySchema.universities u

on d.universityID = u.id

WHERE d.name LIKE '%№6%' AND u.name LIKE '%Sikorsky%'



	name	lastName
1	Mykyta	Moshenchenko

Вивести всі кімнати 8го гуртожитку КПП, в яких живе хоча б одна людина

SELECT

r.code

FROM

dormitorySchema.dormitories d

RIGHT JOIN

dormitorySchema.rooms r

on d.id = r.dormitoryID

RIGHT JOIN

dormitorySchema.livingRecords lr

on r.id = lr.roomID

LEFT JOIN

dormitorySchema.universities u

on d.universityID = u.id

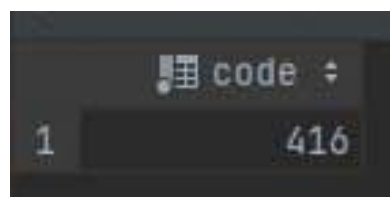
WHERE

u.name LIKE '%Sikorsky%' AND

d.name LIKE '%№8%' AND

lr.fromDate < GETDATE() AND

lr.toDate > GETDATE()



	code
1	416



Вивести всіх студентів, які жили в кімнаті 404 гуртожитку 8 КПП

SELECT

s.name, s.lastName

FROM

dormitorySchema.livingRecords lr

LEFT JOIN

dormitorySchema.students s

on lr.occupantID = s.id

LEFT JOIN

dormitorySchema.rooms r

on lr.roomID = r.id

LEFT JOIN

dormitorySchema.dormitories d

on r.dormitoryID = d.id

LEFT JOIN

dormitorySchema.universities u


on d.universityID = u.id

WHERE

u.name LIKE '%Sikorsky%' AND

d.name LIKE '%№8%' AND

r.code = '404'



	name	lastName
1	Owen	Ball
2	Abigail	Randall

Вивести всіх студентів, що в 2019 році жили в 8ому гуртожитку КПП

SELECT

s.name, s.lastName

FROM

dormitorySchema.livingRecords lr

LEFT JOIN

dormitorySchema.students s  
on lr.occupantID = s.id

LEFT JOIN

dormitorySchema.rooms r  
on lr.roomID = r.id

LEFT JOIN

dormitorySchema.dormitories d  
on r.dormitoryID = d.id

LEFT JOIN

dormitorySchema.universities u  
on d.universityID = u.id

WHERE

u.name LIKE '%Sikorsky%' AND  
d.name LIKE '%№8%' AND  
((lr.fromDate > '2019-1-1' AND lr.fromDate < '2020-1-1') OR  
(lr.toDate > '2019-1-1' AND lr.toDate < '2020-1-1'))

	name	lastName
1	Sophie	MacLeod
2	Joshua	Wright
3	Charles	McDonald
4	Robert	Howard
5	Molly	Wilson
6	Lucas	Walker

Вивести місто, в якому знаходиться найбільше гуртожитків

SELECT

TOP 1 c.name

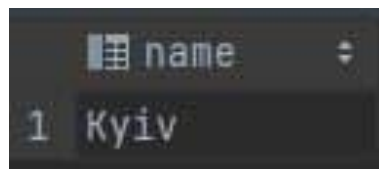
FROM

dormitorySchema.dormitories d

LEFT JOIN

dormitorySchema.addresses a

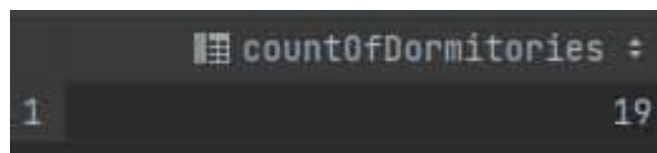
```
on d.addressID = a.id
LEFT JOIN
dormitorySchema.cities c
on a.cityID = c.id
GROUP BY
c.name
ORDER BY
COUNT(1) DESC
```



	name
1	Kyiv

Вивести кількість гуртожитків Києва

```
SELECT
COUNT(1) as countOfDormitories
FROM
dormitorySchema.dormitories d
LEFT JOIN
dormitorySchema.addresses a
on d.addressID = a.id
LEFT JOIN
dormitorySchema.cities c
on a.cityID = c.id
WHERE
c.name = 'Kyiv'
GROUP BY
c.name
```



	countOfDormitories
1	19

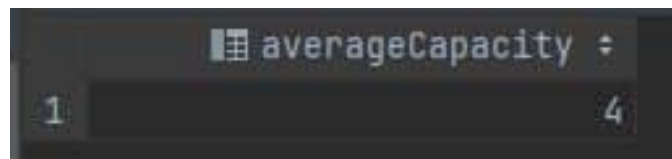
Вивести всі гуртожитки, які знаходяться на вулиці Янгеля

```
SELECT
    d.name AS 'dormitory', u.name AS 'university'
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.addresses a
    on d.addressID = a.id
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
WHERE
    a.street LIKE '%Yanhela%'
```

	dormitory	university
1	№1	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
2	№3	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
3	№4	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
4	№6	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
5	№8	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Вивести середню місткість кімнат гуртожитку №8

```
SELECT
    AVG(capacity) as averageCapacity
FROM
    dormitorySchema.rooms r
RIGHT JOIN
    dormitorySchema.dormitories d
    on r.dormitoryID = d.id
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
WHERE
    d.name LIKE '№8' AND
    u.name LIKE '%Sikorsky%'
```



Вивести гуртожитки, в яких немає коменданта

```
SELECT
    d.name as 'dormitory', u.name as 'university'
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.universities u
    on d.universityID = u.id
WHERE NOT EXISTS(
    SELECT
        *
    FROM
        dormitorySchema.commandants c
    WHERE c.id = d.commandantID
)
```

dormitory	university
1 №11	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
2 №21	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
3 №22	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Погрупувати гуртожитки за типом та дізнатися, якого типу найбільше

```
SELECT
    t.name, COUNT(1) as countOfTypes
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.types t
```

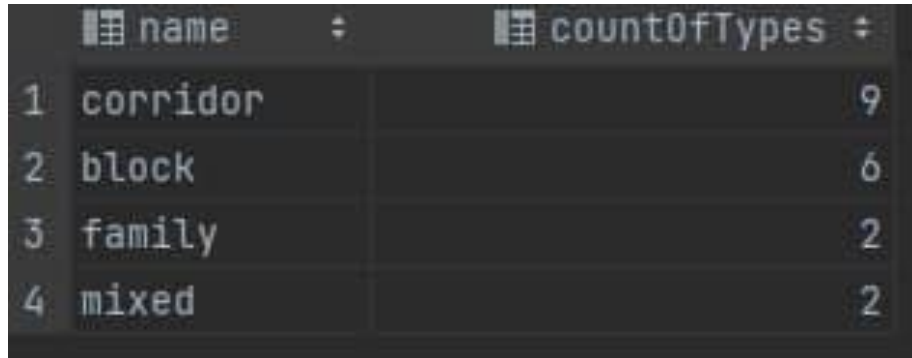
on d.typeID = t.id

GROUP BY

t.name

ORDER BY

COUNT(1) DESC



	name	countOfTypes
1	corridor	9
2	block	6
3	family	2
4	mixed	2

Вивести коридорні гуртожитки, що знаходяться на вулиці Металістів

SELECT

d.name as 'dormitory', u.name as 'university'

FROM

dormitorySchema.dormitories d

LEFT JOIN

dormitorySchema.addresses a

on d.addressID = a.id

LEFT JOIN

dormitorySchema.universities u

on d.universityID = u.id

LEFT JOIN

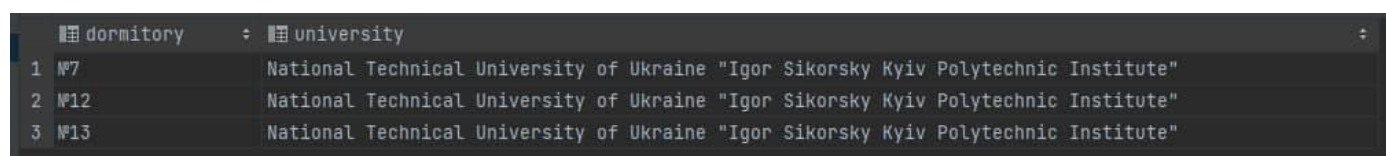
dormitorySchema.types t

on d.typeID = t.id

WHERE

a.street LIKE '%Metallistiv%' AND

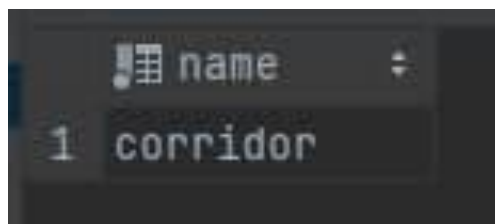
t.name = 'corridor'



	dormitory	university
1	#7	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
2	#12	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
3	#13	National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Вивести, гуртожитків якого типу найбільше на вулиці Янгеля

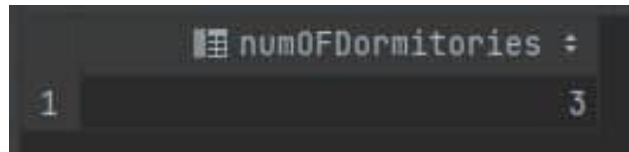
```
SELECT
    TOP 1 t.name
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.types t
    on d.typeID = t.id
LEFT JOIN
    dormitorySchema.addresses a
    on d.addressID = a.id
WHERE
    a.street LIKE '%Yanhela%'
GROUP BY
    t.name
ORDER BY
    COUNT(1) DESC
```



Вивести кількість гуртожитків на вулиці Борщагівській

```
SELECT
    COUNT(1) as numOFDormitories
FROM
    dormitorySchema.dormitories d
LEFT JOIN
    dormitorySchema.addresses a
    on d.addressID = a.id
WHERE
```

a.street LIKE '%Borschahivska%'



Вивести завідуючих разом з комендантами, з якими вони працюють

SELECT

m.name as 'matronName', m.lastName as 'matronLastName',

c.name as 'commandantName', c.lastName as 'commandantLastname'

FROM

dormitorySchema.dormitories d

LEFT JOIN

dormitorySchema.matrons m

on d.matronID = m.id

LEFT JOIN

dormitorySchema.commandants c

on d.commandantID = c.id

	matronName	matronLastName	commandantName	commandantLastname
1	Liubov	Filipishyna	Svitlana	Shtunder
2	Svitlana	Zavalska	Maryna	Krasnokutska
3	Olena	Stromenko	Svitlana	Morhun
4	Svitlana	Kryvenko	Mykyta	Moshenchenko
5	Larysa	Hurovska	Inna	Zamana
6	Nadiya	Tymoshenko	Oksana	Potrats
7	Olha	Blazhko	Vita	Romanenko
8	Tetiana	Nedashkivska	<null>	<null>
9	Svitlana	Marynka	Iryna	Kyseliova
10	Liubov	Koshyl	Maria	Moskalets
11	Olena	Shevchuk	Oksana	Sapronova
12	Larysa	Pidipryhora	Olha	Pimanova
13	Liudmyla	Scherbyna	Tetiana	Levchenko
14	Vira	Bachynska	Nina	Klimova
15	Dmytro	Kulyk	Yulia	Budachevska
16	Iryna	Postolatyeva	Natalia	Yarosh
17	Nataliya	Puzyrna	Nadia	Nosanchuk
18	Yevheniya	Maslova	<null>	<null>
19	Nataliya	Vonsovych	<null>	<null>