



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп’ютерних  
систем**

**Лабораторна робота №1**

з дисципліни **Бази даних і засоби управління**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями  
СУБД PostgreSQL”*

Виконав:

студент III курсу

групи КВ-34

Чижов В.Ю

Перевірив:

Павловский В. I.

**Київ – 2025**

**Мета:** здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Завдання:**

1. Розробити модель «сущність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

### **Опис предметної області**

Предметна область – **Електронна база даних для клієнтів хостелу**.

Електронна база даних «Клієнти хостелу» призначена для автоматизації обліку клієнтів та їхніх бронювань а також про оплату. Система зберігає інформацію про ліжка в хостелі, клієнтів, а також дані про бронювання. Це дозволяє контролювати доступність місць, вести історію проживання та формувати звіти. Одне ліжко може бути заброньоване багато разів у різні дати. Один клієнт може здійснити багато бронювань. Кожне бронювання пов'язане з конкретним ліжком та конкретним клієнтом.

**Контакти:**

Репозиторій Git: посилання;

Телеграм @chizhovl;

## 1. Розробка моделі «сущність-зв'язок» обраної предметної галузі

### Опис сущностей

#### 1) Bed (Ліжко)

Призначення: зберігання довідкової інформації про ліжка у хостелі.

Основні атрибути:

- bed\_id – ідентифікаційний номер ліжка;
- bed\_tier – рівень ліжка (наприклад: перший ярус, другий).

#### 2) Client (Клієнт)

Призначення: зберігання довідкової інформації про клієнтів хостелу.

Основні атрибути:

- client\_id – ідентифікаційний номер клієнта;
- client\_name – ПІБ клієнта;
- client\_pasport – паспортні дані клієнта;
- client\_phone – номер телефону клієнта.

#### 3) Booking (Бронювання)

Призначення: зберігає дані про бронювання місць у хостелі.

Основні атрибути:

- booking\_id – ідентифікаційний номер бронювання;
- bed\_id – FK → bed(bed\_id) – залежність від номера ліжка;
- client\_id – FK → client(client\_id) – залежність від клієнта;
- date\_in – дата заселення;
- date\_out – дата виселення;
- booking\_price – вартість проживання за дане бронювання.

#### 4) Payment(Оплата)

Призначення: зберігання інформації про платежі, здійснені клієнтами за бронювання у хостелі.

Основні атрибути:

- payment\_id – ідентифікаційний номер платежу;
- booking\_id – FK → booking(booking\_id) ідентифікаційний номер бронювання, до якого належить платіж;
- method – спосіб оплати (наприклад: готівка, картка, онлайн);
- amount – сума платежу.

## Опис зв'язків між сущностями

### **1) Client – Bed: M:N** (багато до багатьох, реалізовано через сутність Booking)

Сенс: Між сущностями Client та Bed (Ліжко) існує зв'язок типу багато-до-багатьох, оскільки один клієнт може забронювати багато різних ліжок у різні дати, а одне ліжко може бути заброньоване багатьма клієнтами у різний час. Такий тип зв'язку не може бути реалізований безпосередньо в реляційній базі даних, тому створено проміжну сутність Booking (Бронювання), яка реалізує цей зв'язок.

Кардинальність та обов'язковість:

- З боку Client до Bed: M:N — один клієнт може мати багато бронювань різних ліжок.
- З боку Bed до Client: M:N — одне ліжко може бути заброньоване багатьма клієнтами.
- Через сутність Booking цей зв'язок реалізується як два зв'язки типу 1:N:
  - Client (1) – Booking (N)
  - Bed (1) – Booking (N)

Реалізація в логічній моделі:

- У таблиці Booking містяться зовнішні ключі client\_id ( $\rightarrow$  Client) та bed\_id ( $\rightarrow$  Bed), які забезпечують реалізацію зв'язку Client–Bed типу M:N.
- Таким чином, кожен запис у таблиці Booking описує одне бронювання конкретного клієнта певного ліжка.

### **2) Booking – Client: 1:N** (один до багатьох, обов'язковий)

Сенс: кожне бронювання належить конкретному клієнту.

Кардинальність та обов'язковість:

- З боку Client до Booking: 1..\* — один клієнт може мати багато бронювань (мінімум одне, якщо він користується послугами).
- З боку Booking до Client: 1 — кожне бронювання обов'язково здійснюється одним клієнтом.

### **3) Booking – Bed: 1:N** (один до багатьох, обов'язковий)

Сенс: кожне бронювання пов'язане з конкретним ліжком у хостелі.

Кардинальність та обов'язковість:

- З боку Bed до Booking: 1..\* — одне ліжко може мати багато бронювань у різні дати.
- З боку Booking до Bed: 1 — кожне бронювання вказує лише одне ліжко.

#### 4)Booking – Payment: 1:N(один до багатьох, обов'язковий)

Сенс: кожне бронювання може мати кілька платежів (наприклад, аванс, повна оплата, додаткові послуги), але кожен платіж належить лише до одного бронювання.

Кардинальність та обов'язковість:

- З боку Booking до Payment: 1..\* — одне бронювання може мати кілька платежів.
- З боку Payment до Booking: 1 — кожен платеж обов'язково належить конкретному бронюванню.

Графічне подання концептуальної моделі «Сутність-зв'язок» зображенено на рисунку 1.

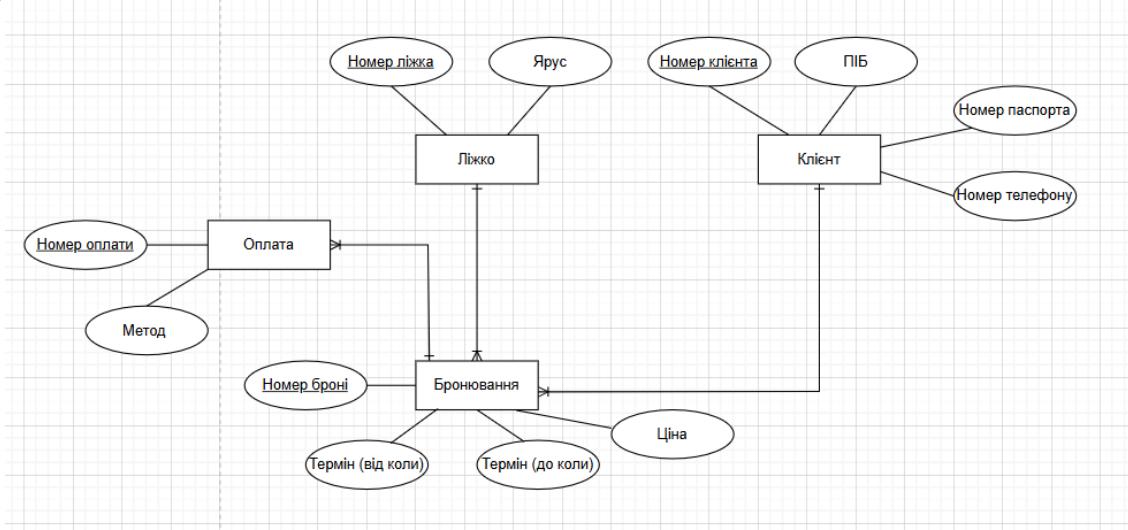


Рисунок 1 - ER-діаграма

## 2. Перетворення концептуальної моделі у логічну модель та схему бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об'єктів бази даних

Сутність	Атрибут	Тип атрибуту
bed	bed id	integer
	bed tier	integer
client	client id	integer
	client name	varchar (100)
	client pasport	integer
	client phone	integer
booking	booking id	integer
	bed id	integer
	client id	integer
	date in	date

	date out	date
	booking price	integer
payment	payment id	integer
	booking id	integer
	method	varchar (100)

### Функціональні залежності для кожної таблиці

#### **Bed**

- bed\_id → bed\_tier;

#### **Client**

- client\_id → client\_name, client\_pasport, client\_phone;

#### **Booking**

- booking\_id → date\_in, date\_out, booking\_price;
- bed\_id → bed (існує залежність через зовнішній ключ);
- client\_id → client (існує залежність через зовнішній ключ);

#### **Payment**

- payment\_id → method;
- booking\_id → booking (існує залежність через зовнішній ключ);

### 3. Виконання нормалізації схеми бази даних до третьої нормальної форми (ЗНФ).

Графічне подання логічної моделі «Сутність-зв'язок» зображене на рисунку 2.

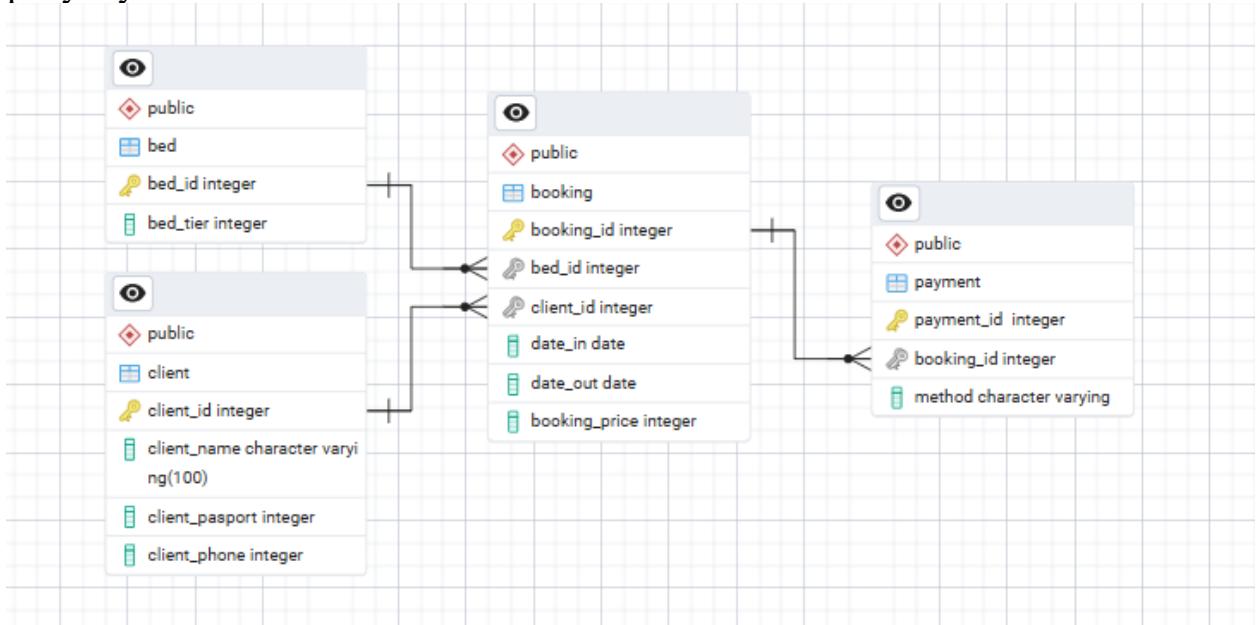


Рисунок 2 - Схема бази даних

#### 1НФ (Перша нормальна форма)

Виконується, оскільки:

- Усі атрибути таблиць Client, Bed, Booking, Payment є атомарними: немає множинних значень чи повторюваних груп.
- У таблиці Client кожне поле містить неподільні значення (наприклад, client\_name, client\_pasport, client\_phone).
- У таблиці Bed атрибут bed\_tier зберігає лише одне значення для кожного ліжка.
- У таблиці Booking усі атрибути (bed\_id, client\_id, date\_in, date\_out, booking\_price) мають лише атомарні значення.
- У таблиці Payment атрибути (payment\_id, booking\_id, method) також є атомарними — наприклад, поле method зберігає лише один спосіб оплати.
- Первинні ключі (client\_id, bed\_id, booking\_id, payment\_id) однозначно ідентифікують кожен рядок у своїй таблиці.

## **2НФ (Друга нормальна форма)**

Виконується, оскільки:

- У таблицях Client, Bed та Payment первинні ключі прості (одне поле), тому жоден неключовий атрибут не може залежати лише від частини ключа.
- У таблиці Booking первинний ключ простий (booking\_id), і всі неключові атрибути (bed\_id, client\_id, date\_in, date\_out, booking\_price) повністю залежать від нього.
- У таблиці Payment первинний ключ простий (payment\_id), і всі неключові атрибути (booking\_id, method) залежать від нього повністю.

## **3НФ (Третя нормальна форма)**

- Виконується, оскільки:
- У таблиці Booking усі атрибути (bed\_id, client\_id, date\_in, date\_out, booking\_price) залежать безпосередньо від первинного ключа (booking\_id). Дані про клієнтів винесені в таблицю Client, а дані про ліжка — в таблицю Bed, тому транзитивних залежностей немає.
- У таблиці Client атрибути (client\_name, client\_pasport, client\_phone) залежать лише від client\_id.
- У таблиці Bed атрибут bed\_tier залежить тільки від bed\_id.
- У таблиці Payment атрибути (booking\_id, method) залежать лише від payment\_id. Залежності типу «атрибут → атрибут» у неключових полях відсутні (наприклад, method не залежить від booking\_id, а лише від ключа payment\_id).

## 4. Ознайомлення та розробка таблиць баз даних у pgAdmin4

### Таблиці бази даних у pgAdmin4 Таблиця Bed

Table: bed (public)

General Columns Advanced Constraints

**Columns**

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	bed_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	bed_tier	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

**Constraints**

Primary Key Foreign Key Unique

	Name	Columns
	bedid_pkey	bed_id

**Buttons:** Close Reset Save

### Таблиця Client

Table: client (public)

General Columns Advanced Constraints

**Columns**

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	client_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	client_name	character varying	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	client_pasport	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	client_phone	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

**Buttons:** Close Reset Save

## Table: client (public)

General Columns Advanced Constraints

Primary Key Foreign Key Unique

	Name	Columns
	clientid_pkey	client_id

Close Reset Save

## Таблиця Booking

## Table: booking (public)

General Columns Advanced Constraints

Columns							
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	booking_id	integer	▾		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	bed_id	integer	▾		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	client_id	integer	▾		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	date_in	date	▾		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	date_out	date	▾		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	date_out	date	▾		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	booking_price	integer	▾		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

## Table: booking (public)

General Columns Advanced Constraints

Primary Key Foreign Key Unique

	Name	Columns
	bookingid_pkey	booking_id

Close Reset Save

## Таблиця Payment

Table: payment (public)

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	payment_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	booking_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	method	character varying	100		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Table: payment (public)

Name		Columns
	payment_pkey	payment_id

До таблиць внесено по декілька рядків інформації.

## Таблиця Bed

Data Output    Messages    Notifications

	bed_id [PK] integer	bed_tier
0+	4	2
1+	3	2
2+	2	2
3+	1	2

Таблиця Booking

Data Output    Messages    Notifications

	booking_id [PK] integer	bed_id integer	client_id integer	date_in date	date_out date	booking_price integer
0+	3	3	3	04.10.2...	11.10.2025	2100
1+	2	2	2	01.10.2...	05.10.2025	1500
2+	1	1	1	27.09.2...	30.09.2025	900

Таблиця Client

Data Output    Messages    Notifications

	client_id [PK] integer	client_name character varying (100)	client_pasport integer	client_phone integer
0+	3	Yurii Chizhov	2009	+380955222504
1+	2	Lev Chizhov	1802	+380953478123
2+	1	Vladislav Chizhov	1110	+380668662347

Таблиця Payment

	payment_id [PK] integer	booking_id integer	method character varying
0+	3	3	cash
1+	2	2	cash
2+	1	1	card

## Таблиці в коді SQL

```

BEGIN;

CREATE TABLE IF NOT EXISTS public.bed1
(
    bed_id integer NOT NULL,
    bed_tier integer NOT NULL,
    CONSTRAINT bed_pkey PRIMARY KEY (bed_id)
);

CREATE TABLE IF NOT EXISTS public.booking1
(
    booking_id integer NOT NULL,
    bed_id integer NOT NULL,
    client_id integer NOT NULL,
    date_in date NOT NULL,
    date_out date NOT NULL,
    booking_price integer NOT NULL,
    CONSTRAINT bookingid_pkey PRIMARY KEY (booking_id)
);

CREATE TABLE IF NOT EXISTS public.payment1
(
    payment_id integer NOT NULL,
    booking_id integer NOT NULL,
    method character varying(100) NOT NULL,
    CONSTRAINT payment_pkey PRIMARY KEY (payment_id)
);

CREATE TABLE IF NOT EXISTS public.client
(
    client_id integer NOT NULL,
    client_name character varying(100) COLLATE pg_catalog."default" NOT
NULL,
    client_pasport integer NOT NULL,
    client_phone integer NOT NULL,
    CONSTRAINT clientid_pkey PRIMARY KEY (client_id)
);

ALTER TABLE IF EXISTS public.booking1
ADD FOREIGN KEY (bed_id)
REFERENCES public.bed1 (bed_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;

ALTER TABLE IF EXISTS public.booking1
ADD FOREIGN KEY (client_id)
REFERENCES public.client (client_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;

ALTER TABLE IF EXISTS public.payment1
ADD FOREIGN KEY (booking_id)
REFERENCES public.booking1 (booking_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;
END;

```

## Висновок

У ході виконання лабораторної роботи була спроектована база даних для системи обліку клієнтів, бронювань та платежів у хостелі. Було розроблено ER-модель предметної області, що включає чотири сутності: Bed (Ліжко), Client (Клієнт), Booking (Бронювання) та Payment (Оплата). Для кожної сутності визначено основні атрибути та встановлено зв'язки, які відображають логіку роботи системи: один клієнт може здійснювати багато бронювань, кожне ліжко може бути заброньоване багато разів, а кожне бронювання може мати кілька платежів.

На основі концептуальної моделі побудовано логічну схему бази даних у PostgreSQL. Для всіх таблиць визначені первинні та зовнішні ключі, що забезпечують цілісність і узгодженість даних. Виконана нормалізація до третьої нормальної форми (ЗНФ) підтвердила відсутність надлишковості та транзитивних залежностей.

У середовищі pgAdmin 4 створені таблиці та внесено тестові дані, що демонструють коректність роботи моделі. Розроблена база даних дозволяє автоматизувати облік клієнтів і бронювань, контролювати доступність ліжок, зберігати історію проживань, а також вести облік платежів за послуги.

Таким чином, усі поставлені завдання лабораторної роботи виконані, а отримана база даних може слугувати основою для повноцінної інформаційної системи управління хостелом.