



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ 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»

Виконав: студент III курсу

ФПМ групи КВ-83

Мягкова Дар'я Андріївна

Перевірів(ла): Павловський В.І.

Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL

Метою роботи є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

Завдання роботи полягає у наступному:

1. Розробити концептуальну модель - модель «сутність-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 кожен з таблиць засобами pgAdmin 4.

Концептуальна модель предметної області «Сервіс доставки їжі»

В концептуальній моделі предметної області «Сервіс доставки їжі» (Рисунок 1) виділяються наступні сутності та зв'язки між ними:

1. Сутність «Людина» з атрибутами: ID, телефон, ім'я, адреса;
2. Сутність «Замовлення» з атрибутами: ID, замовник, дата, страва;
3. Сутність «Заклад» з атрибутами: ID, телефон, назва, адреса;
4. Сутність «Страва» з атрибутами: ID, назва, вага, заклад;
5. Сутність «Ліцензія» з атрибутами: ID, заклад, номер ліцензії.

Один заклад може запропонувати кілька страв, тому між сутностями «Заклад» і «Страва» зв'язок R(1:N).

Людина може зробити кілька замовлень, тому між сутностями «Людина» і «Замовлення» зв'язок R(1:N).

В одному замовленні може бути кілька страв, та навпаки одна страва може бути присутня в кількох замовленнях, тому між сутностями «Замовлення» і «Страва» зв'язок R(N:M).

В одного заклада може бути одна ліцензія, тому між сутностями «Заклад» і «Ліцензія» зв'язок R(1:1).

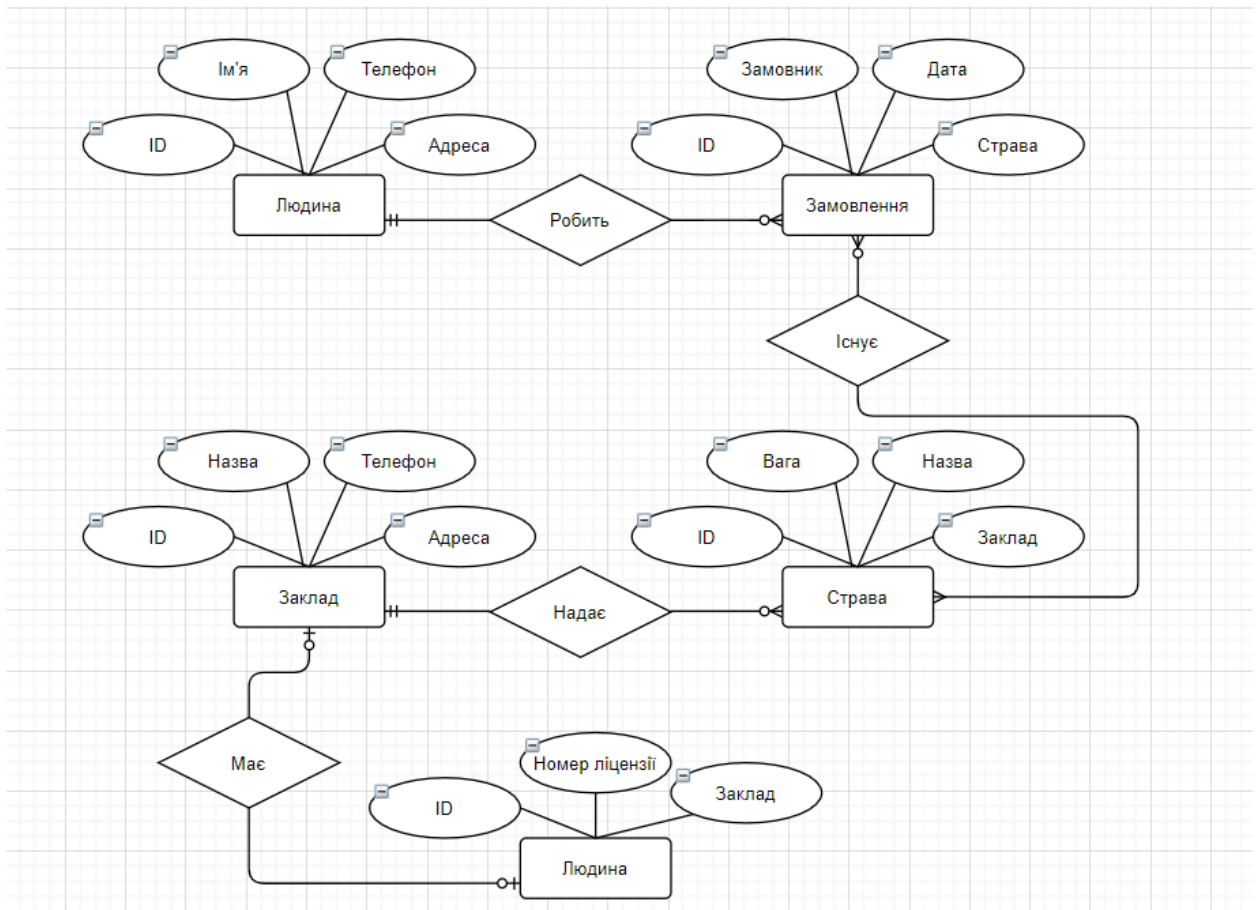


Рисунок 1 - Концептуальна модель предметної області «Сервіс доставки їжі»

Логічна модель (схема) БД «Сервіс доставки їжі»

В логічній моделі (Рисунок 2):

1. Сутність «Людина» перетворена в таблицю «Person»;
2. Сутність «Замовлення» перетворена в таблицю «Order»;
3. Сутність «Заклад» перетворена в таблицю «Restaurant»;
4. Сутність «Страва» перетворена в таблицю «Dish»;
5. Сутність «Ліцензія» перетворена в таблицю «License».

Оскільки відношення між замовленнями та стравами R(N:M), потрібно створити ще одну додаткову таблицю з закладами та стравами під назвою «Order/Dish».

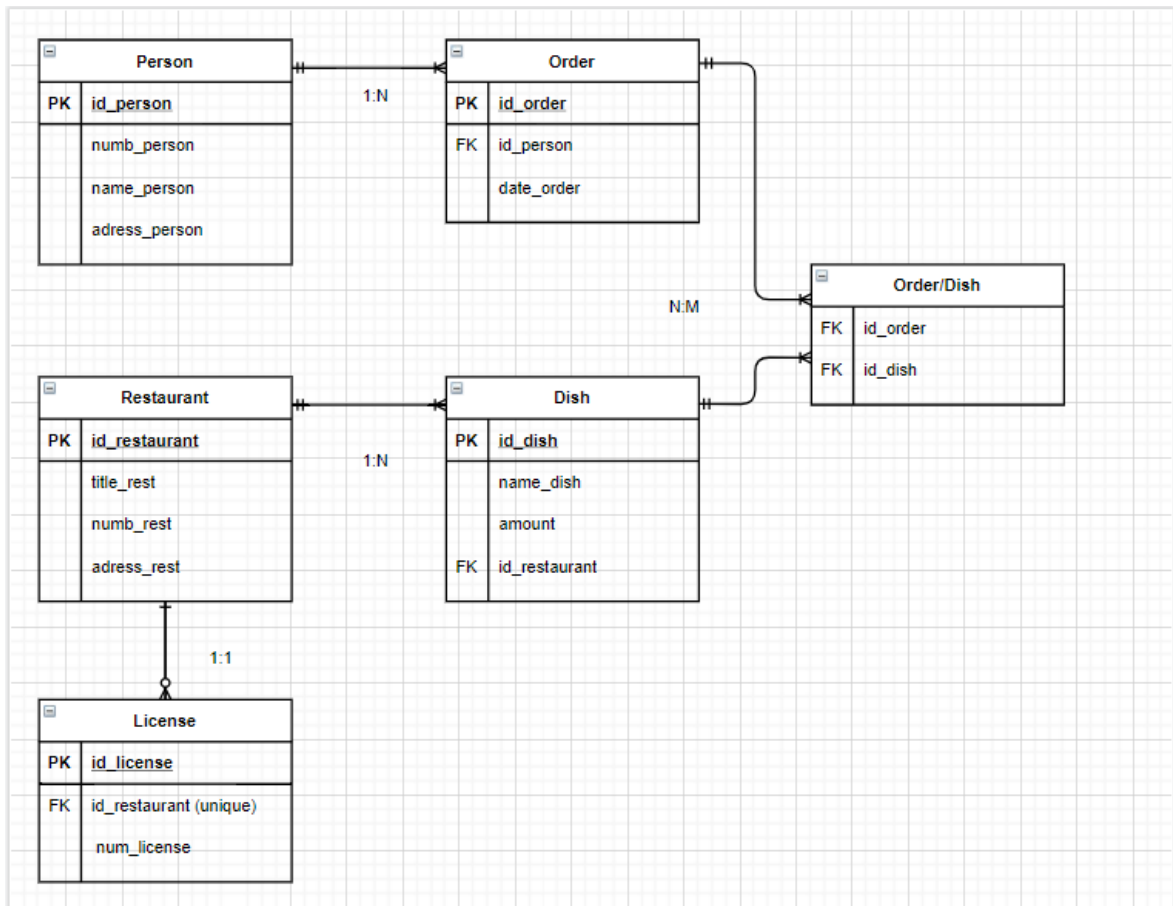


Рисунок 2 - Логічна модель предметної області «Сервіс доставки їжі»

Опис структури БД

Сутність	Атрибут	Тип
Person (інформація про замовника їжі)	<u>id_person</u> - унікальний ID замовника numb_person – телефон замовника name_person – ім'я замовника address_person – адреса замовника	Числовий (PK) Текстовий Текстовий Текстовий
Restaurant (інформація про заклад, що надає «послугу»)	<u>id_restaurant</u> - унікальний ID закладу title_rest – назва закладу numb_rest – телефон закладу address_rest – адреса закладу	Числовий (PK) Текстовий Текстовий Текстовий
Dish (інформація про страву, що може бути замовлена)	<u>id_dish</u> - унікальний ID страви name_dish – назва страви amount – «вага» страви id_restaurant – ID закладу, що надає страву	Числовий (PK) Текстовий Текстовий Числовий (FK)

Order (інформація про замовлення)	id_order - унікальний ID замовлення id_person – ID замовника date – дата замовлення	Числовий (PK) Числовий (FK) Текстовий
Order/Dish (інформація про замовлення та страву)	id_order - ID замовлення id_dish - ID страви	Числовий (FK) Числовий (FK)
License (інформація про ліцензію закладу)	id_license – унікальний ID ліцензії id_restaurant – ID закладу num_license – номер ліцензії закладу	Числовий (PK) Числовий (FK unique) Текстовий

Структура БД «Сервіс доставки їжі»

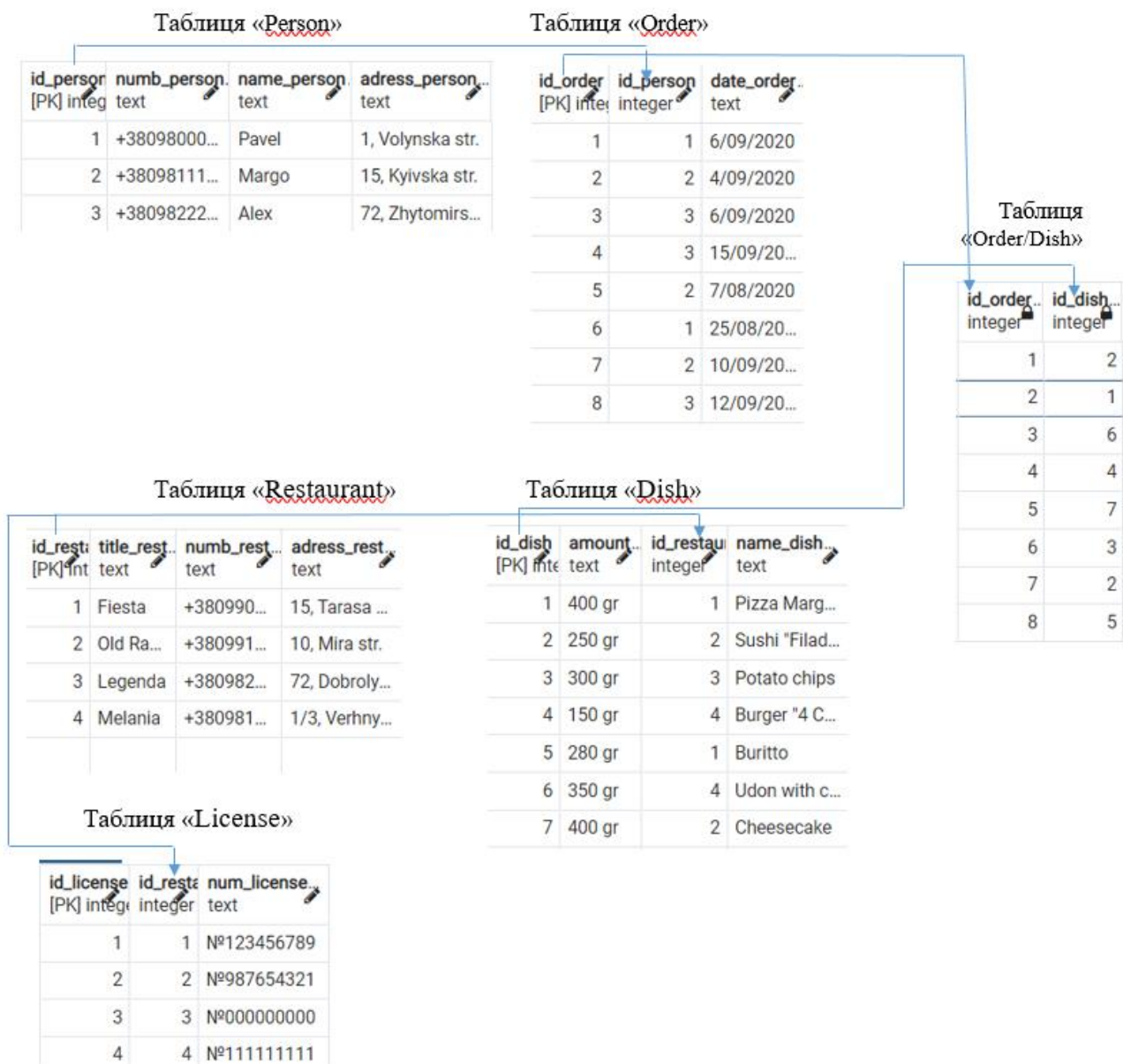


Рисунок 3 - Структурна модель предметної області «Сервіс доставки їжі»

Схема відповідає 1НФ тому, що кожний стовбець таблиці є атомарним і кожний запис є унікальним.

Схема відповідає 2НФ тому, що схема відповідає 1НФ і всі таблиці мають первинні ключі, і кожен неключовий атрибут функціонально залежить від первинного ключа, а не його частини.

Схема відповідає 3НФ тому, що схема відповідає 2НФ і всі атрибути залежні тільки від первинного ключа (в ній немає транзитивних залежностей між неключовими атрибутами).

Структура БД «Сервіс доставки їжі» в pgAdmin 4

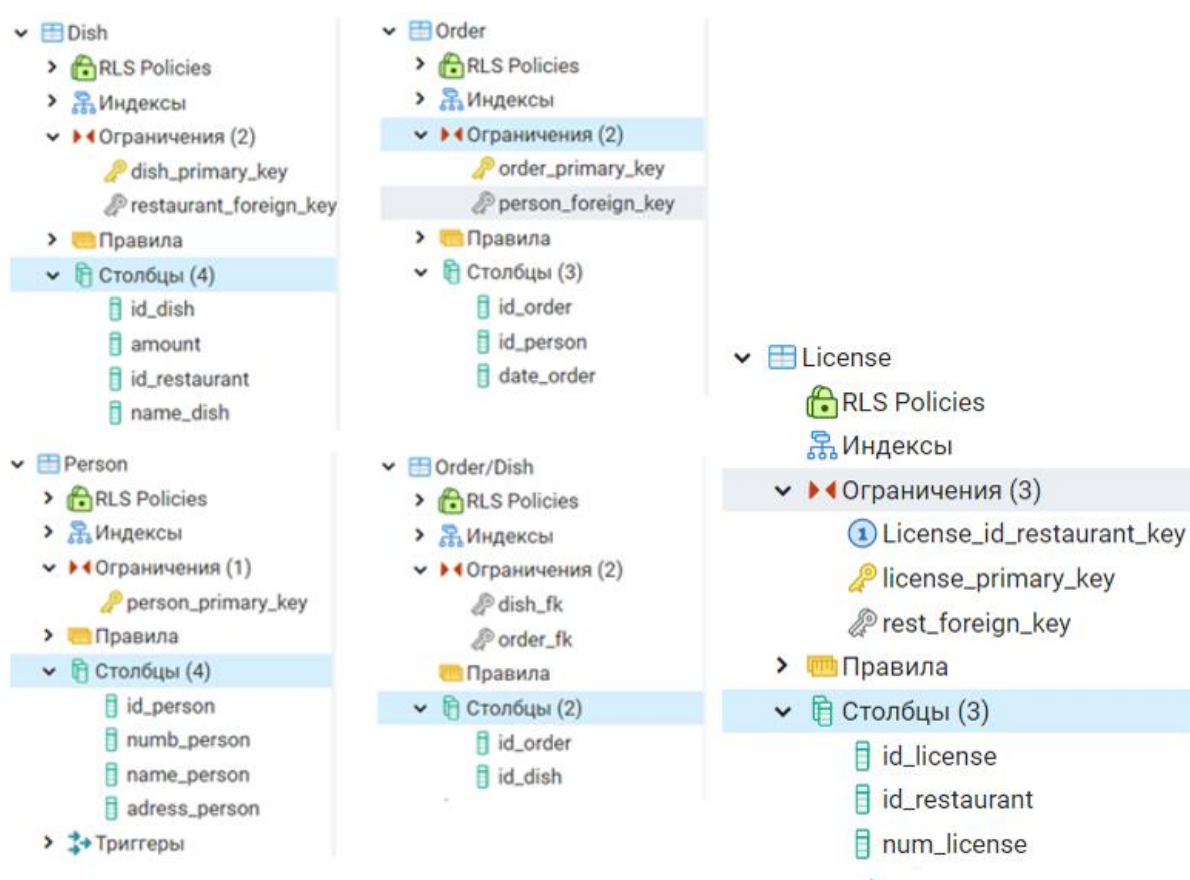


Рисунок 4 - Структура предметної області «Сервіс доставки їжі» в pgAdmin 4

Опис таблиць БД «Сервіс доставки їжі» в pgAdmin 4

```
CREATE TABLE public."Person"  
(  
    id_person integer NOT NULL,  
    numb_person text NOT NULL,  
    name_person text COLLATE pg_catalog."default" NOT NULL,  
    adress_person text COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT person_primary_key PRIMARY KEY (id_person)  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Person"  
    OWNER to postgres;
```

```
CREATE TABLE public."Restaurant"  
(  
    id_restaurant integer NOT NULL,  
    title_rest text COLLATE pg_catalog."default" NOT NULL,  
    numb_rest text NOT NULL,  
    adress_rest text COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT restaurant_primary_key PRIMARY KEY (id_restaurant)  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Restaurant"  
    OWNER to postgres;
```

```

CREATE TABLE public."Order"
(
    id_order integer NOT NULL,
    id_person integer NOT NULL,
    date_order text NOT NULL,
    id_dish integer NOT NULL,
    CONSTRAINT order_primary_key PRIMARY KEY (id_order),
    CONSTRAINT person_foreign_key FOREIGN KEY (id_person)
        REFERENCES public."Person" (id_person) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE public."Order"
    OWNER to postgres;

CREATE TABLE public."Dish"
(
    id_dish integer NOT NULL,
    amount text NOT NULL,
    id_restaurant integer NOT NULL,
    name_dish text NOT NULL,
    CONSTRAINT dish_primary_key PRIMARY KEY (id_dish),
    CONSTRAINT restaurant_foreign_key FOREIGN KEY (id_restaurant)
        REFERENCES public."Restaurant" (id_restaurant) MATCH SIMP
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE public."Dish"
    OWNER to postgres;

```



```
CREATE TABLE public."Order/Dish"  
(  
    id_order integer NOT NULL,  
    id_dish integer NOT NULL,  
    CONSTRAINT order_fk FOREIGN KEY (id_order)  
        REFERENCES public."Order" (id_order) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    CONSTRAINT dish_fk FOREIGN KEY (id_dish)  
        REFERENCES public."Dish" (id_dish) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)  
  
TABLESPACE pg_default;  
  
ALTER TABLE public."Order/Dish"  
    OWNER to postgres;
```

```
CREATE TABLE public."License"  
(  
    id_license integer NOT NULL,  
    id_restaurant integer unique NOT NULL,  
    num_license text COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT license_primary_key PRIMARY KEY (id_license),  
    CONSTRAINT rest_foreign_key FOREIGN KEY (id_restaurant)  
        REFERENCES public."Restaurant" (id_restaurant) MATCH SIMP  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)  
  
TABLESPACE pg_default;  
  
ALTER TABLE public."License"  
    OWNER to postgres;
```