

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**

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

Виконав:

студент ІII курсу

групи КВ-91

Бітлян А. В.

Перевірив:

Павловський В. І.

Київ – 2021

*Завдання* роботи полягає у наступному:

1. Розробити 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 і внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Модель учбової предметної області «Автобус»**

Обрана предметна галузь передбачає організацію автобусного депо. Для побудови бази даних для обраної області було створено сутності, зображені на рисунку 1:

1. Автобус (Bus), з атрибутами: код автобуса, модель. Призначена для збереження інформації про автобуси в наявності.
2. Модель автобуса (Bus model), з атрибутами: код моделі, назва, місткість, категорія. Призначена для збереження інформації про модель автобуса, яка в свою чергу містить інформацію про категорію для подальшого визначення того, чи підходить дана категорія для певного водія.
3. Маршрут (Route), з атрибутами: код маршруту, назва, довжина, пасажиропотік. Призначена для збереження інформації про існуючі маршрути.
4. Водій (Driver), з атрибутами: код водія, П.І.Б. водія, доступні категорії, навантаження (години). В базі даних сутність зберігає інформацію про водія, який може керувати тільки автобусами доступної для нього категорії. Також присутня інформація про кількість робочих годин водія.

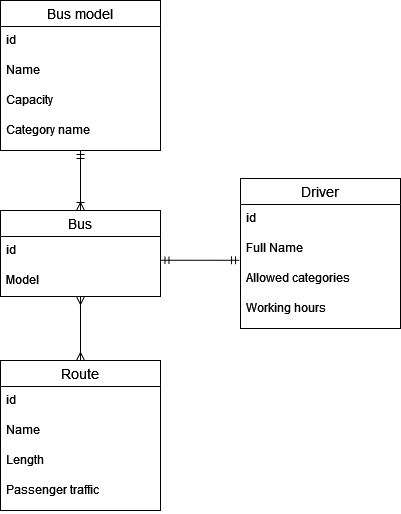


Рисунок 1 - ER-діаграма, побудована за нотацією Чена (інструмент: draw.io)

**Опис зв’язків**

Сутність «Автобус» має зв’язок N:1 по відношенню до моделі автобуса,

тому що декілька автобусів можуть мати одну й ту саму модель.

А також Сутність «Автобус» має зв’язок N:M по відношенню до сутності

«Маршрут», тому що один і той самий автобус може бути використаний на

різних маршрутах і на одному маршруті можуть використовувати різні

маршрути.

Також Сутність «Автобус» має зв’язок 1:1 по відношенню до сутності

водій, оскільки в кожного водія свій особистий автобус.

**Перетворення моделі у схему бази даних**

Сутність «Автобус» було перетворено у таблицю «bus», сутність «Модель автобуса» було перетворено у таблицю «bus\_model» », а зв’язок N:1 між ними зумовив появу зовнішнього ключа fk\_bus\_model.

Сутність «Водій» було перетворено у таблицю «driver» , а зв’язок 1:1 між нею і сутністю автобус зумовив появу зовнішнього ключа fk\_bus\_driver.

Сутність «Маршрут» була перетворена у таблицю «route» », а зв’язок M:N між нею і сутністю автобус зумовив появу зовнішнього ключа fk\_bus\_route.

Схему можна побачити на рисунку 2.

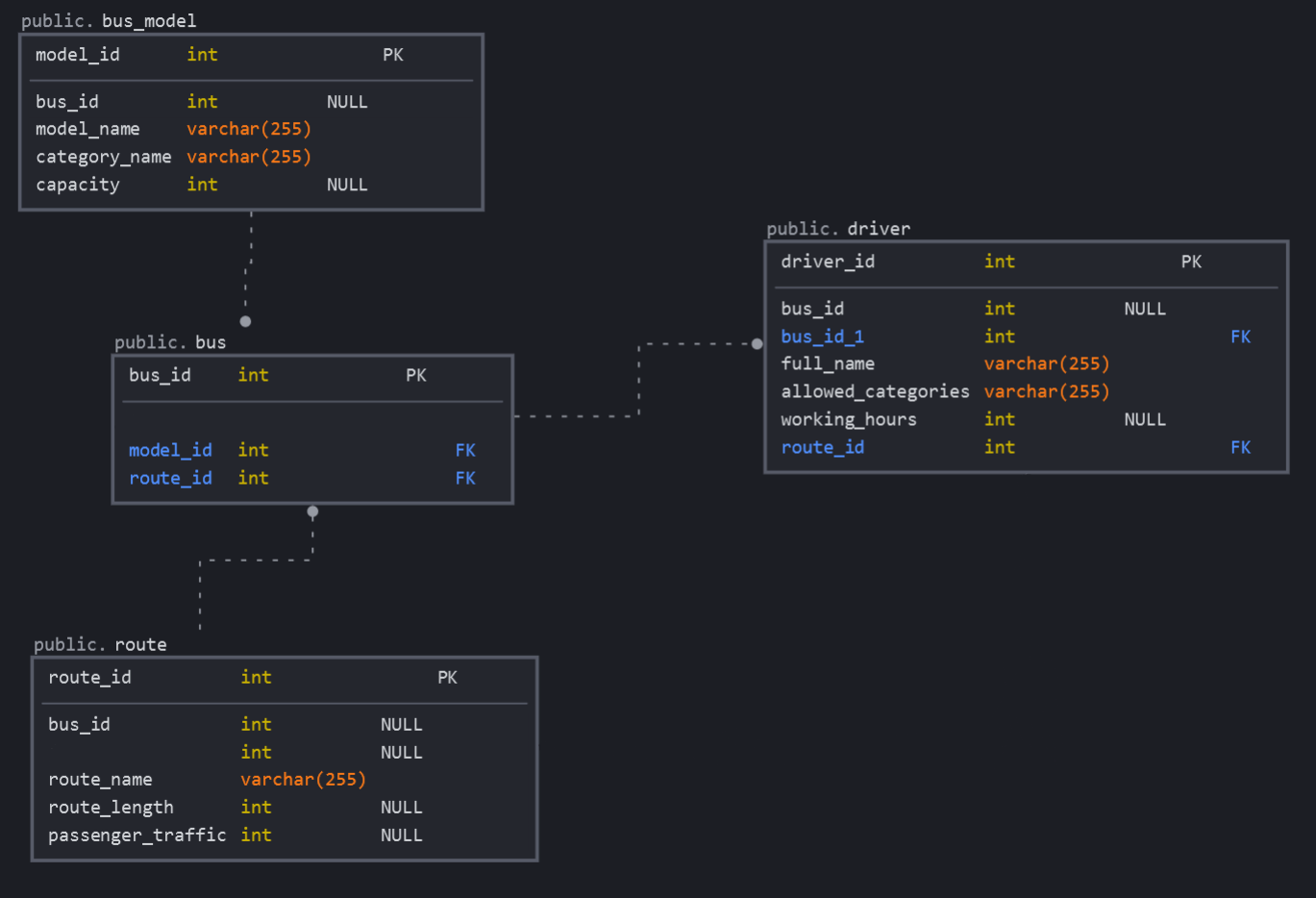


Рисунок 2 - Схема бази даних (інструмент: sqldbm.com)

**Опис об’єктів бази даних у вигляді таблиці**

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **bus** – *містить всі автобуси* | **bus\_id** *- унікальний ідентифікатор автобуса* | **integer** *(числовий)* |
| **bus\_model** – *містить інформацію про модель* | **model**\_**id** *- унікальний ідентифікатор моделі*  **bus\_id** *- унікальний ідентифікатор автобуса*  **model\_name** *– назва моделі*. *Не допускає NULL*  **category\_name** *– назва категорії*. *Не допускає NULL*  **capacity** *– місткість автобуса* | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)*  **integer** *(числовий)* |
| **driver** - *містить інформацію про водія* | **driver\_id** *- унікальний ідентифікатор водія*  **bus\_id** *- унікальний ідентифікатор автобуса*  **full\_name** *- П.І.Б. водія*  **allowed\_categories** –*дозволені категорії*  **working\_hours** *– кількість робочих годин в тиждень* | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)*  **integer** *(числовий)* |
| **route** - *містить інформацію про маршрут* | **route\_id** *- унікальний ідентифікатор маршруту*  **bus\_id** *- унікальний ідентифікатор автобуса*  **route\_name** *– назва маршруту*. *Не допускає NULL*  **route\_length** *– довжина маршруту*  **passenger\_traffic** *– кількість пасажирів в день на маршруті* | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)*  **integer** *(числовий)*  **integer** *(числовий)* |

**Функціональні залежності для кожної таблиці**

BUS:

bus\_id → model\_id, route\_id

bus\_id → model\_id (назва моделі залежить від його коду)

bus\_id → route\_id (назва маршруту залежить від його коду)

BUS\_MODEL:

model\_id → bus\_id, model\_name, category\_name, capacity

model\_id → bus\_id (назва автобуса залежить від його коду)

model\_id → model\_name (назва моделі залежить від його коду)

model\_id → category\_name (назва категорії залежить від моделі)

model\_id → capacity (місткість автобуса залежить від моделі)

DRIVER:

driver\_id → bus\_id, full\_name, allowed\_categories, working\_hours

driver\_id → bus\_id (назва автобуса залежить від його коду)

driver\_id → full\_name (П.І.Б. водія залежить від його коду)

driver\_id → allowed\_categories (дозволені категорії залежать від моделі автобуса)

driver\_id → working\_hours (робочі години залежить від коду водія)

ROUTE:

route\_id → bus\_id, driver\_id, route\_name, route\_length, passenger\_traffic

route\_id → bus\_id (назва автобуса залежить від його коду)

route\_id → route\_name (назва маршруту залежить від його коду)

route\_id → route\_length (довжина маршруту залежить від його коду)

route\_id → passenger\_traffic (пасажиропотік маршруту залежить від його коду)

**Відповідність схеми бази даних до третьої нормальної форми**

Схема відповідає 1НФ, тому що:

1. В таблиці немає дубльованих рядків.

2. В кожній комірці зберігається атомарне (скалярне) значення.

3. В кожному стовпці зберігаються дані одного типу.

Схема відповідає 2НФ, тому що:

1. Вона відповідає 1НФ.

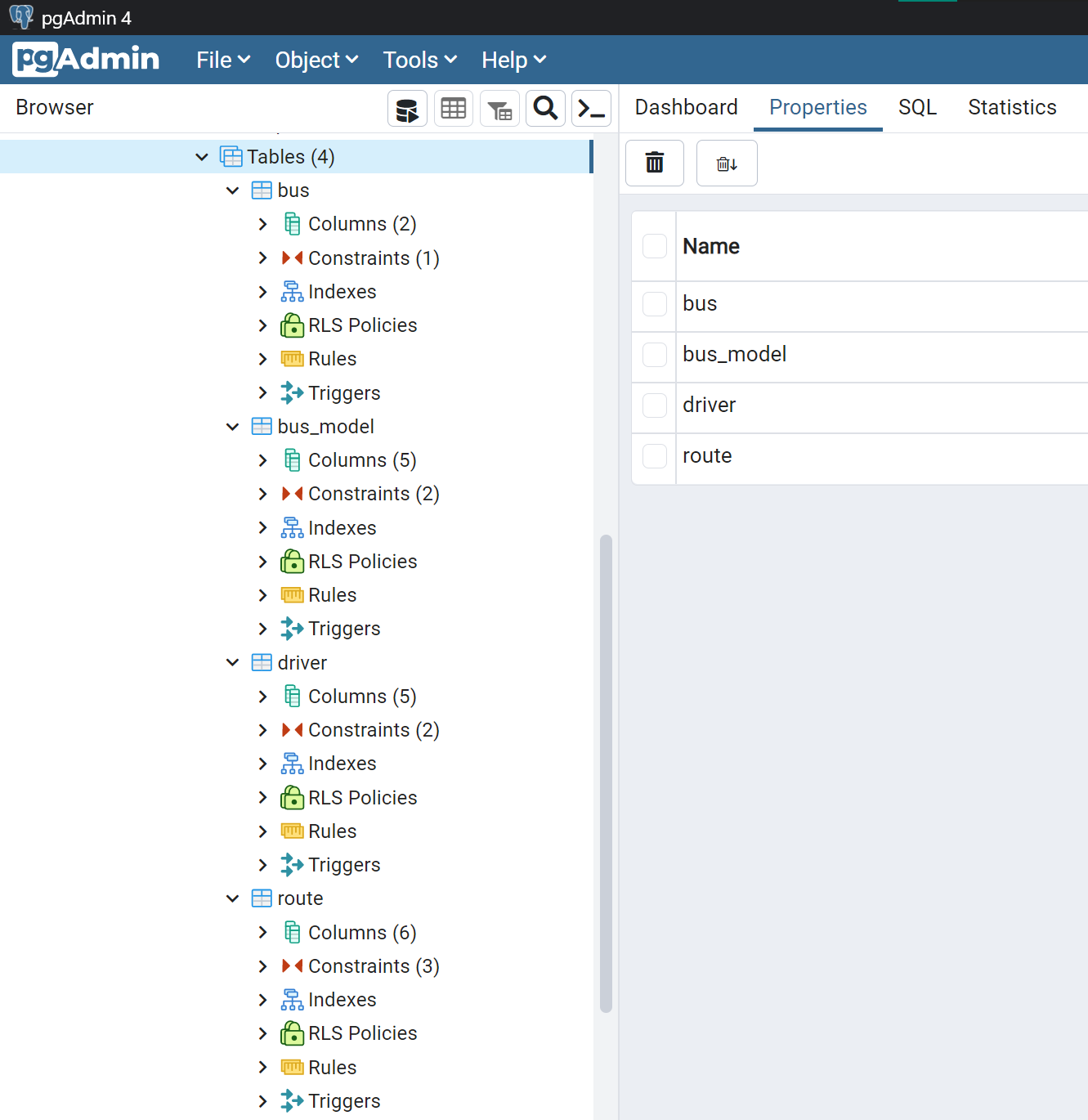
2. Має первинний ключ, а всі не ключові стовпці таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що:

1. Вона відповідає 2НФ.

2. Всі не ключові атрибути таблиці залежать винятково від усього первинного ключа, а не його частини. Тобто кожен неключових атрибут нетранзитивно (без посередника) залежить від первинного ключа.

**Таблиці бази даних у pgAdmin4**



**SQL-текст опису БД**

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

-- TOC entry 201 (class 1259 OID 16662)

-- Name: bus; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.bus (

bus\_id integer NOT NULL

);

ALTER TABLE public.bus OWNER TO postgres;

--

-- TOC entry 200 (class 1259 OID 16660)

-- Name: bus\_bus\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.bus ALTER COLUMN bus\_id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.bus\_bus\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 205 (class 1259 OID 16684)

-- Name: bus\_model; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.bus\_model (

model\_id integer NOT NULL,

bus\_id integer,

model\_name character varying(255) NOT NULL,

category\_name character varying(255) NOT NULL,

capacity integer

);

ALTER TABLE public.bus\_model OWNER TO postgres;

--

-- TOC entry 204 (class 1259 OID 16682)

-- Name: bus\_model\_model\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.bus\_model ALTER COLUMN model\_id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.bus\_model\_model\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 203 (class 1259 OID 16669)

-- Name: driver; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.driver (

driver\_id integer NOT NULL,

bus\_id integer,

full\_name character varying(255) NOT NULL,

allowed\_categories character varying(255) NOT NULL,

working\_hours integer

);

ALTER TABLE public.driver OWNER TO postgres;

--

-- TOC entry 202 (class 1259 OID 16667)

-- Name: driver\_driver\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.driver ALTER COLUMN driver\_id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.driver\_driver\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 207 (class 1259 OID 16699)

-- Name: route; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.route (

route\_id integer NOT NULL,

bus\_id integer,

driver\_id integer,

route\_name character varying(255) NOT NULL,

route\_length integer,

passenger\_traffic integer

);

ALTER TABLE public.route OWNER TO postgres;

--

-- TOC entry 206 (class 1259 OID 16697)

-- Name: route\_route\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public.route ALTER COLUMN route\_id ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public.route\_route\_id\_seq

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);