



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»

Тема: *«Створити БД "Міжміське сполучення" в СУБД PostgreSQL
з допомогою конструктора PgAdmin 4»*

Виконав: студент 3 курсу

ФПМ групи КВ-83

Ткачук Тарас

Перевірів: Павловський В.І.

Київ – 2020

Лабораторна робота №1.
**Створити БД "Міжміське сполучення" в СУБД PostgreSQL
з допомогою конструктора PgAdmin 4**

Мета роботи: створити БД Міжміське сполучення та сформулювати обмеження цілісності на значення даних.

Порядок виконання роботи

1. Розробити концептуальну модель вибраного предметного середовища. Концептуальну модель предметного середовища Міжміське сполучення наводиться в Додатку А до лабораторної роботи;
2. Розробити логічну модель (схему) даних БД. Логічна модель (схема) даних БД Міжміське сполучення наводиться в Додатку Б до лабораторної роботи;
3. Вивчити склад та правила роботи з СУБД PostgreSQL;
4. Створити в СУБД PostgreSQL БД Міжміське сполучення, використовуючи конструктори таблиць та стовпчиків (краще колонок). Схема даних Міжміське сполучення наводиться в Додатку Б до лабораторної роботи. Перелік атрибутів наводиться в Додатку В до лабораторної роботи;
5. Заповнити створену БД даними (порядку 5-10 записів в кожній таблиці).

Зміст звіту

1. [Концептуальна модель предметної області.](#)
2. [Логічна модель \(схема\) БД.](#)
3. [Опис структури БД "Міжміське сполучення"](#)
4. [Представлення БД в pgAdmin 4](#)

Додаток А. Концептуальна модель предметної області "Міжміське сполучення"

В концептуальній моделі предметної області "Міжміське сполучення" (Рисунок1) виділяються наступні сутності та зв'язки між ними.

Сутність "Водій" з атрибутами: id, ім'я, прізвище;

Сутність "Маршрут" з атрибутами: id, початкове місто, кінцеве місто, id водія;

Сутність "Автобус" з атрибутами : id, номерний знак, кількість мість, id маршрута;

Сутність "Квиток" з атрибутами: id, ціна, id маршрута, дата покупки;

Сутність "Пасажи́р" з атрибутами : id, ім'я, прізвище, вік;

Між сутностями "Квиток" та "Пасажи́р" зв'язок R(1:1), тому що 1 пасажир може мати тільки 1 квиток, а 1 квиток може належити лише 1-ому пасажиру.

Між сутностями "Маршрут" та "Квиток" зв'язок R(1:M), тому що для 1-го маршруту може бути багато квитків, але 1 квиток може належити тільки 1-ому маршруту.

Між сутностями "Маршрут" та "Автобус" зв'язок R(M:N), тому що на 1-ому маршруті може бути багато автобусів, і 1 автобус може бути на багатьох маршрутах.

Між сутностями "Водій" та "Маршрут" зв'язок R(M:1), тому що 1 водій може бути лише на 1-ому маршруті, але на маршруті може бути багато водіїв.

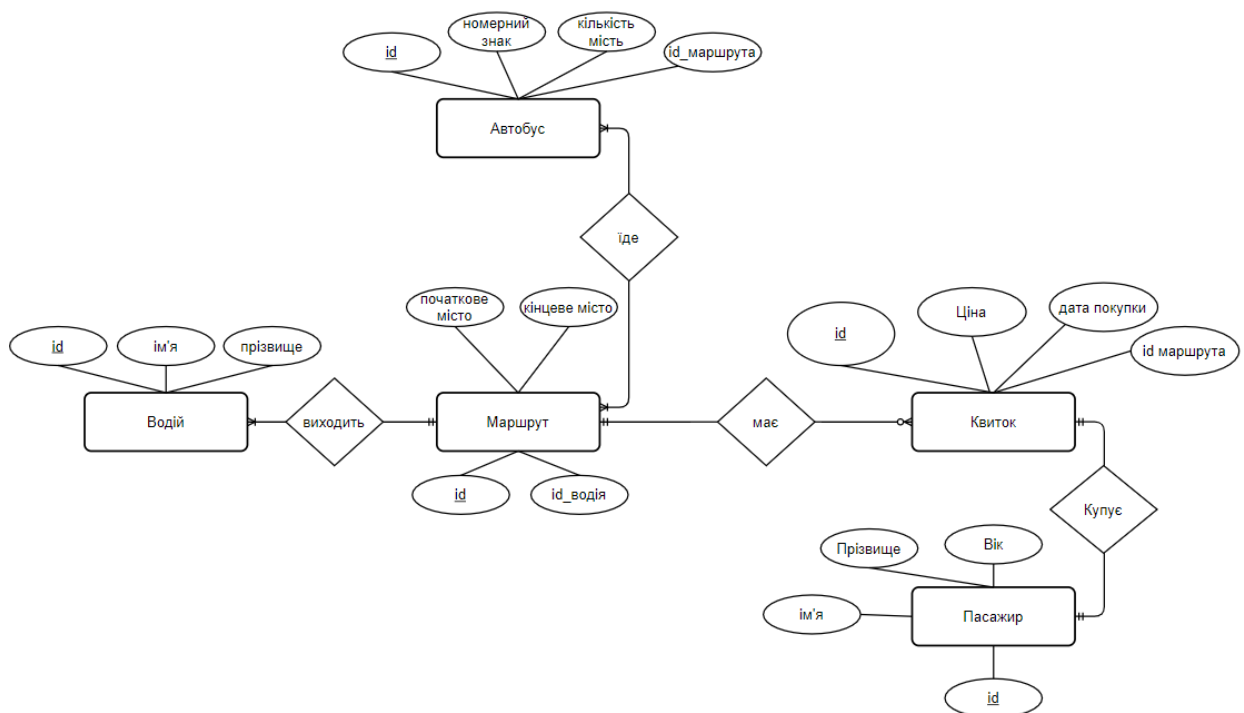


Рисунок 1 - Концептуальна модель предметної області "Міжміське сполучення"

Додаток Б. Логічна модель (схема) БД "Міжміське сполучення"

В логічній моделі (Рисунок 2):

Сутність "Водій" перетворена в таблицю "Driver".

Сутність "Маршрут" перетворена в таблицю "Route".

Сутність "Автобус" була перетворена в таблицю "Bus".

Сутність "Квиток" була перетворена в таблицю "Ticket".

Сутність "Пасажир" була перетворена в таблицю "Passenger".

Оскільки відношення між маршрутом та автобусом R(M:N), потрібно створити ще одну перехідну таблицю з маршрутами і автобусами на них під назвою "route_bus".

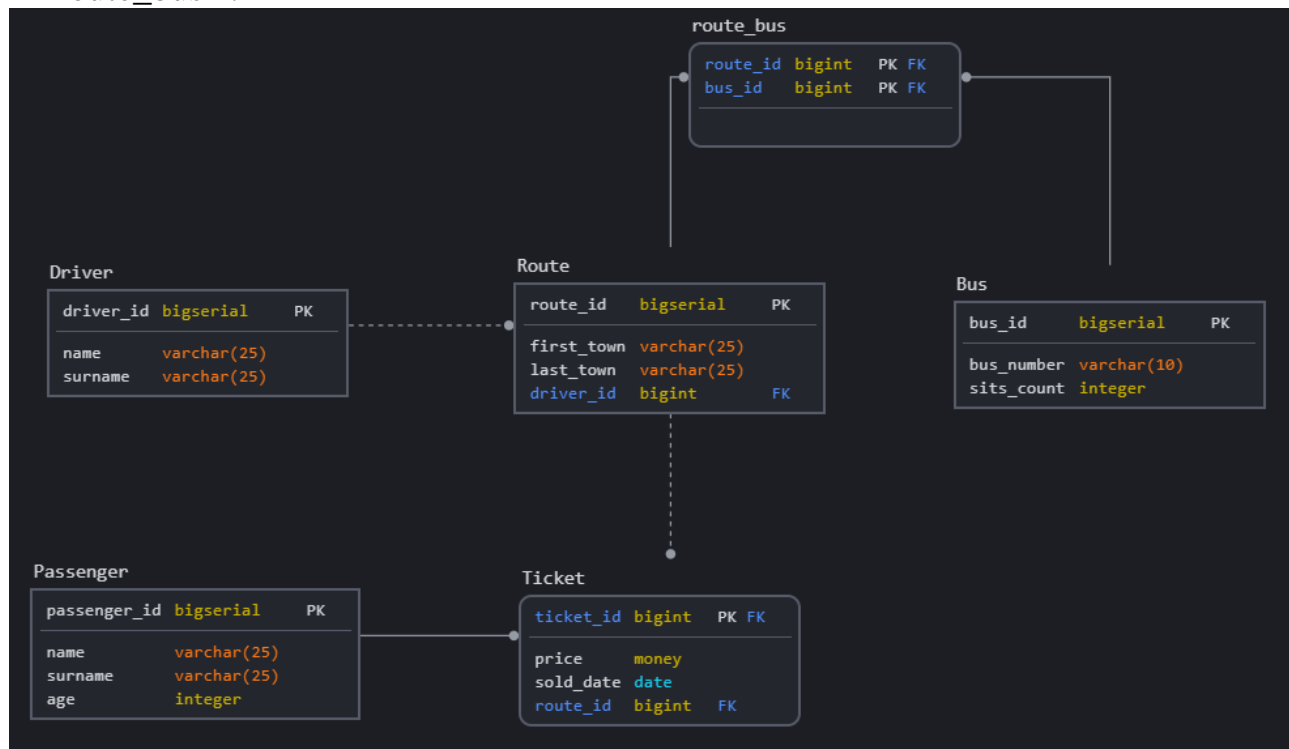


Рисунок 2 - Логічна модель предметної області "Міжміське сполучення"

Схема бази даних відповідає 1НФ тому, що всі рядки унікальні, всі атрибути прості і мають лише атомарні або скалярні значення.

Схема бази даних відповідає 2НФ тому, що всі таблиці мають первинні ключі, і всі неключові атрибути залежать від первинного ключа, а не від його частини.

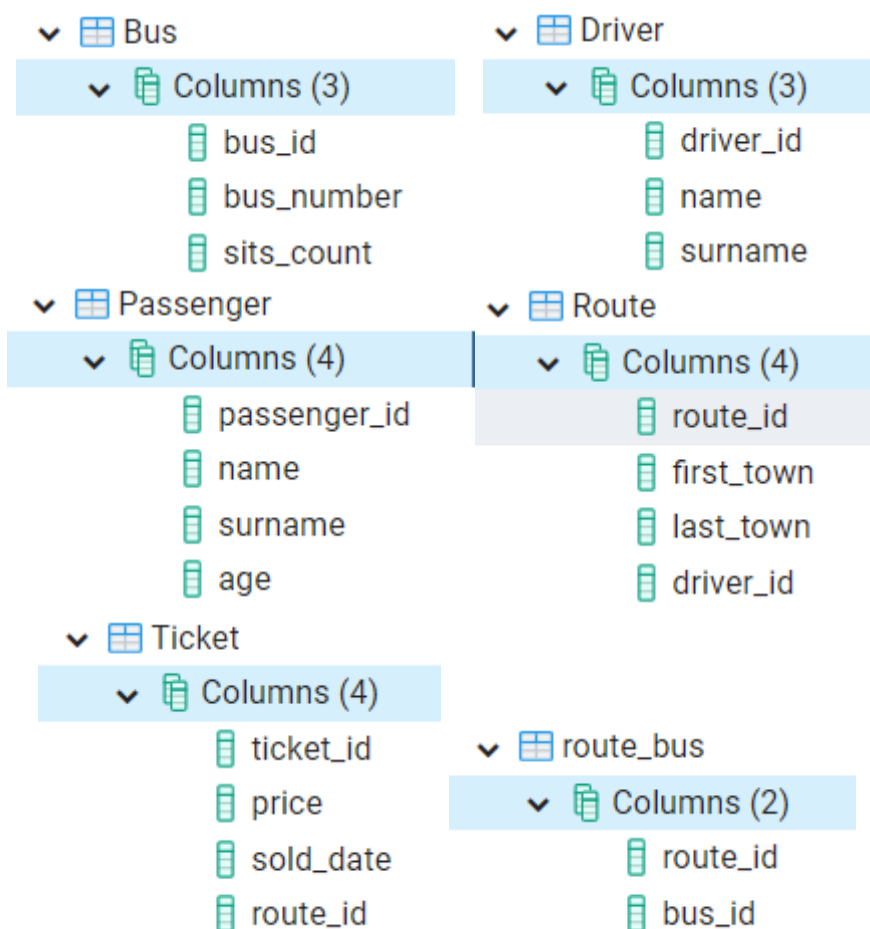
Схема бази даних відповідає 3НФ тому, що всі неключові атрибути нетранзитивно залежні від первинного ключа.

Додаток В. Опис структури БД "Міжміське сполучення"

Таблиця 1 - Текстове представлення логічної моделі (схеми) БД

Відношення	Атрибут	Тип
Відношення "Driver" містить інформацію про водія	driver_id – унікальний номер водія в БД name – ім'я водія. Не допускає NULL. surname – прізвище водія. Не допускає NULL.	Числовий, SERIAL PK Текстовий(25) Текстовий(25)
Відношення "Route" містить інформацію про маршрут	route_id – унікальний номер маршруту в БД first_town – початкове місто. Не допускає NULL last_town – кінцеве місто. Не допускає NULL. driver_id – унікальний номер водія маршруту. Не допускає NULL	Числовий, SERIAL PK Текстовий(25) Текстовий(25) Числовий FK
Відношення "Bus" містить інформацію про автобус	bus_id – унікальний номер автобуса в БД. bus_number – реєстраційний номер автобуса. Не допускає NULL. sits_count – кількість сидячих місць. Не допускає NULL.	Числовий, SERIAL PK Текстовий(10) Числовий
Відношення "Ticket" містить інформацію про квиток	ticket_id – унікальний номер квитка в БД. price – ціна квитка. Не допускає NULL sold_date – дата покупки квитка. Не допускає NULL route_id – унікальний номер маршруту. Не допускає NULL	Числовий PK FK Монетарний MONEY Дата DATE Числовий FK
Відношення "Passenger" містить інформацію про пасажирів	passenger_id – унікальний номер пасажирів в БД. name – ім'я пасажирів. Не допускає NULL. surname – прізвище пасажирів. Не допускає NULL. age – вік пасажирів. Не допускає NULL.	Числовий, SERIAL PK Текстовий(25) Текстовий(25) Числовий
Відношення "route_bus" містить інформацію про зв'язок автобуса і маршруту.	route_id – унікальний номер маршруту. Не допускає NULL bus_id – унікальний номер автобуса. Не допускає NULL	Числовий PK FK Числовий PK FK

Додаток Г. Структура БД "Міжміське сполучення" в pgAdmin 4



Опис таблиць БД "Міжміське сполучення" в pgAdmin 4

Таблиця "Bus"

```

CREATE TABLE public."Bus"
(
    bus_id bigint NOT NULL DEFAULT nextval('"Bus_bus_id_seq"'::regclass),
    bus_number character varying(10) COLLATE pg_catalog."default" NOT NULL,
    sits_count integer NOT NULL,
    CONSTRAINT "busPK" PRIMARY KEY (bus_id)
)
  
```

	bus_id [PK] bigint	bus_number character varying (10)	sits_count integer
1	1	AB 5451 AM	35
2	2	AC 1537 AH	40
3	3	AA 7168 AT	37
4	4	BB 1087 AP	50
5	5	AB 9128 AC	28

Таблица “Driver”

```
CREATE TABLE public."Driver"
(
    driver_id bigint NOT NULL DEFAULT nextval('"Driver_driver_id_seq"'::regclass),
    name character varying(25) COLLATE pg_catalog."default" NOT NULL,
    surname character varying(25) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "driverPK" PRIMARY KEY (driver_id)
)
```

	driver_id [PK] bigint	name character varying (25)	surname character varying (25)
1	1	Dmitriy	Glomozda
2	2	Vlad	Gleb
3	3	Alex	Woods
4	4	Jack	Dalton
5	5	Lora	Blake

Таблица “Passenger”

```
CREATE TABLE public."Passenger"
(
    passenger_id bigint NOT NULL DEFAULT nextval('"Passenger_passenger_id_seq"'::regclass),
    name character varying(25) COLLATE pg_catalog."default" NOT NULL,
    surname character varying(25) COLLATE pg_catalog."default" NOT NULL,
    age integer NOT NULL,
    CONSTRAINT "passengerPK" PRIMARY KEY (passenger_id)
)
```

	passenger_id [PK] bigint	name character varying (25)	surname character varying (25)	age integer
1	1	Anna	Hatton	20
2	2	Samuel	Holmes	25
3	3	Claus	Barrymore	27
4	4	Bernard	Abrams	17
5	5	Cornelius	Downey	34

Таблиця “Route”

```
CREATE TABLE public."Route"
(
    route_id bigint NOT NULL DEFAULT nextval('"Route_route_id_seq"'::regclass),
    first_town character varying(25) COLLATE pg_catalog."default" NOT NULL,
    last_town character varying(25) COLLATE pg_catalog."default" NOT NULL,
    driver_id bigint NOT NULL,
    CONSTRAINT "routePK" PRIMARY KEY (route_id),
    CONSTRAINT "routeFK" FOREIGN KEY (driver_id)
        REFERENCES public."Driver" (driver_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```

	route_id [PK] bigint	first_town character varying (25)	last_town character varying (25)	driver_id bigint
1	1	Penkivka	Kherson	1
2	2	Kyiv	Vinnytsa	2
3	3	Madrid	Praga	5
4	4	Rome	Praga	3
5	5	Moscow	Kyiv	2
6	6	Chernihiv	Rivne	4

Таблиця “Ticket”

```
CREATE TABLE public."Ticket"
(
    ticket_id bigint NOT NULL,
    price money NOT NULL,
    sold_date date NOT NULL,
    route_id bigint NOT NULL,
    CONSTRAINT "ticketPK" PRIMARY KEY (ticket_id),
    CONSTRAINT "ticketFK" FOREIGN KEY (route_id)
        REFERENCES public."Route" (route_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT "ticket_idFK" FOREIGN KEY (ticket_id)
        REFERENCES public."Passenger" (passenger_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```










		ticket_id [PK] bigint 	price money 	sold_date date 	route_id bigint 
1		1	20,80 ?	2018-11-28	1
2		2	35,00 ?	2018-10-24	2
3		3	46,00 ?	2018-10-23	2
4		4	81,50 ?	2019-09-10	4
5		5	67,00 ?	2020-03-18	5

Таблица “route_bus”

```
CREATE TABLE public.route_bus
(
    route_id bigint NOT NULL,
    bus_id bigint NOT NULL,
    CONSTRAINT "route_busPK" PRIMARY KEY (route_id, bus_id),
    CONSTRAINT "bus_idFK" FOREIGN KEY (bus_id)
        REFERENCES public."Bus" (bus_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT "route_idFK" FOREIGN KEY (route_id)
        REFERENCES public."Route" (route_id) MATCH FULL
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```

		route_id [PK] bigint 	bus_id [PK] bigint 
1		1	1
2		1	2
3		2	2
4		3	3
5		4	4
6		5	5