

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря  
Сікорського”**

**Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих  
комп’ютерних систем**

**ЛАБОРАТОРНА РОБОТА № 1**

*з дисципліни*

*“Бази даних та засоби управління”*

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

**Група: KB-03  
Виконав: Семенков М.С.  
[GDoc](#)**

**Оцінка:**

**Київ – 2022**

## **Завдання на лабораторну роботу і вимоги до виконання**

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

*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 та внести декілька рядків даних у кожен з таблиць засобами pgAdmin 4.

*Вимоги до ER-моделі:*

1. Сутності моделі предметної галузі мають містити зв'язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п'яти.
3. Передбачити наявність зв'язку з атрибутом.
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow's foot)”, UML.

*Вимоги до інструментарію:*

1. Створення ER-діаграм: Google Docs (Drawing) або <https://www.draw.io/> або <https://www.lucidchart.com>
2. Середовище для створення таблиць відлагодження SQL-запитів до бази даних – pgAdmin 4.
3. СУБД - PostgreSQL 10-14.

*У звіті щодо пункту №1 завдання має бути:*

- перелік сутностей з описом їх призначення;
- графічний файл розробленої моделі «сутність-зв'язок»;
- назва нотації.

*У звіті щодо пункту №2 завдання має бути:*

- опис процесу перетворення (наприклад, “сутність А було перетворено у таблицю А, а зв’язок R (M:N) зумовив появу додаткової таблиці R1 тощо);
- схему бази даних у графічному вигляді з **назвами таблиць (!) та зв’язками між ними, а також необхідно намалювати перетворену ER-діаграму у ТАБЛИЦІ БД! Це означає, що тут не може бути зв’язку N:M, мають бути позначені первинні та зовнішні ключі, обмеження NOT NULL та UNIQUE і бажано внести типи даних атрибутів.**

*У звіті щодо пункту №3 завдання має бути:*

- пояснення (**обґрунтування!**) щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. Пояснення **полягає у наведенні функціональних залежностей**, що демонструють висновки. У випадку невідповідності надати опис необхідних змін у схемі;
- У випадку проведення змін у схемі бази даних надати оновлену версію схеми, інакше - не наводити схему.

*У звіті щодо пункту №4 завдання має бути:*

- навести копії екрану з pgAdmin4, що відображають назви, типи та обмеження на стовпці (доступне у закладці “Columns” та “Constraints” властивостей “Properties” таблиць дерева об’єктів у pgAdmin4);
- навести копії екрану з pgAdmin4, що відображають вміст таблиць бази даних у PostgreSQL. Таблиці на зображенні обов'язково **повинні мати назву!**

### **Опис обраної предметної галузі**

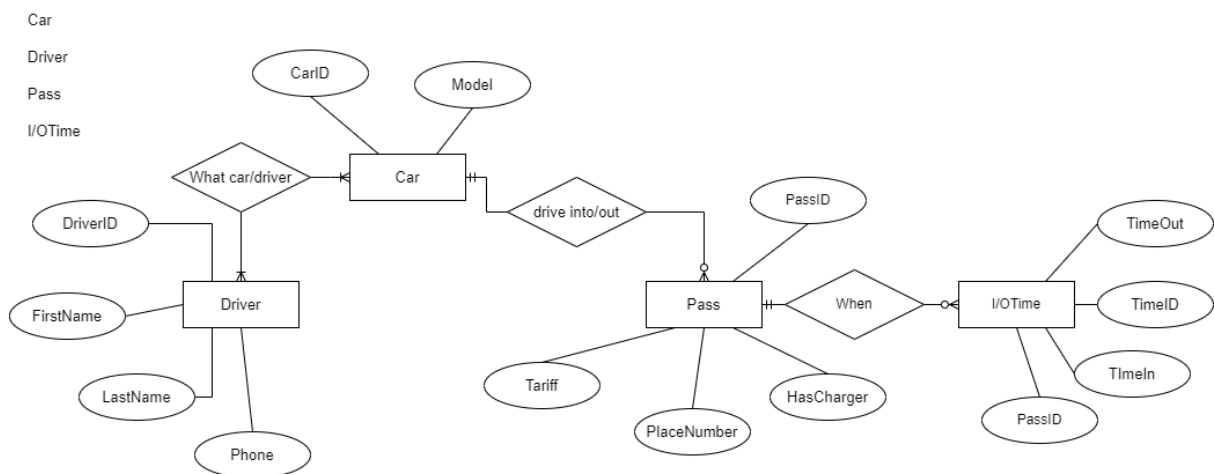
Обрана предметна галузь – паркінг. На паркінгу можуть розміщуватися машини. Кожній машині при кожному в'їзді видається новий пропуск, в якому фіксується інформація про машину, про місце на якому вона розміщується і про час в'їзду і виїзду. У машин може бути кілька водіїв і у водіїв може бути багато машин.

## Завдання 1 – Розробити модель «сутність-зв’язок» предметної галузі

Галузь розбита на 4 галузі: водій, машина, пропуск і час.

- Driver – сутність з описом водія, містить його ім’я, фамілію, унікальний ідентифікатор (номер прав) і контактний телефон. Ця сутність дозволяє ідентифікувати власників авто.
- Car – сутність з описом авто, містить назву моделі і унікальний ідентифікатор авто.
- Pass – сутність пропуску, містить його унікальний номер, ідентифікатор авто, до якого прив’язаний пропуск, номер місця на якому стане машина, інформацію про саме місце (наявність зарядки для електрокарів і тариф для цього місця). Ця сутність зв’язує авто з місцем на якому стоїть це авто.
- I/OTime – досить абстрактна сутність, призначена для нормалізації бази даних, це дозволяє не створювати нові записи при в’їзді/виїзді авто до/з паркінгу до таблиці сутності Pass. Ця сутність зберігає інформацію про час і ідентифікатор пропуску, до якого відносяться записи про час.

### Parking:



Діаграма моделі «сутність-зв’язок» предметної галузі “Паркінг” у нотації Чена:

## Завдання 2 – Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL

Сутність Driver було перетворено у таблицю Driver. Сутність Car було перетворено у таблицю Car. Зв’язок What car/driver (M:N) зумовив появу додаткової таблиці CarDriver. Сутність Pass було перетворено у таблицю Pass. Сутність I/OTime було перетворено у таблицю I/OTime. Всі

інші зв'язки створення додаткових таблиць не потребують, але вони обумовлюють появу зовнішніх ключів.

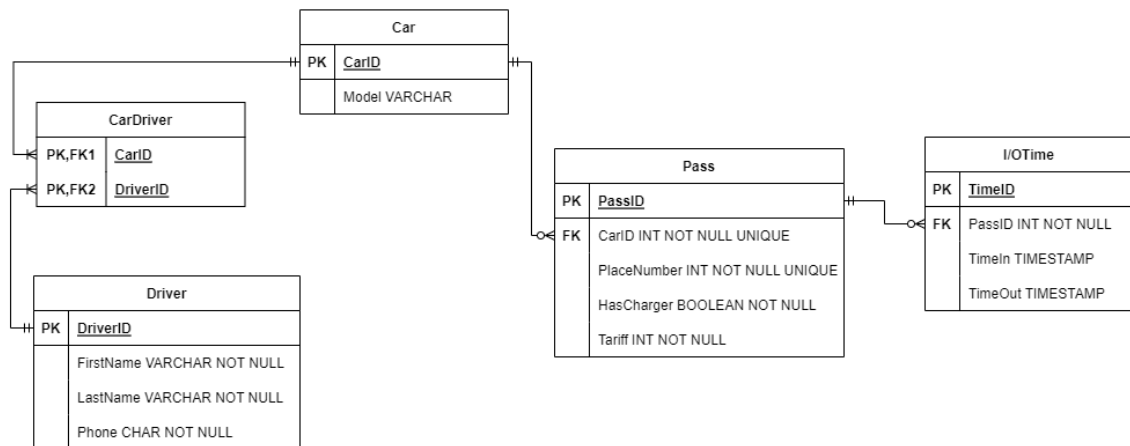


Схема бази даних у графічному вигляді

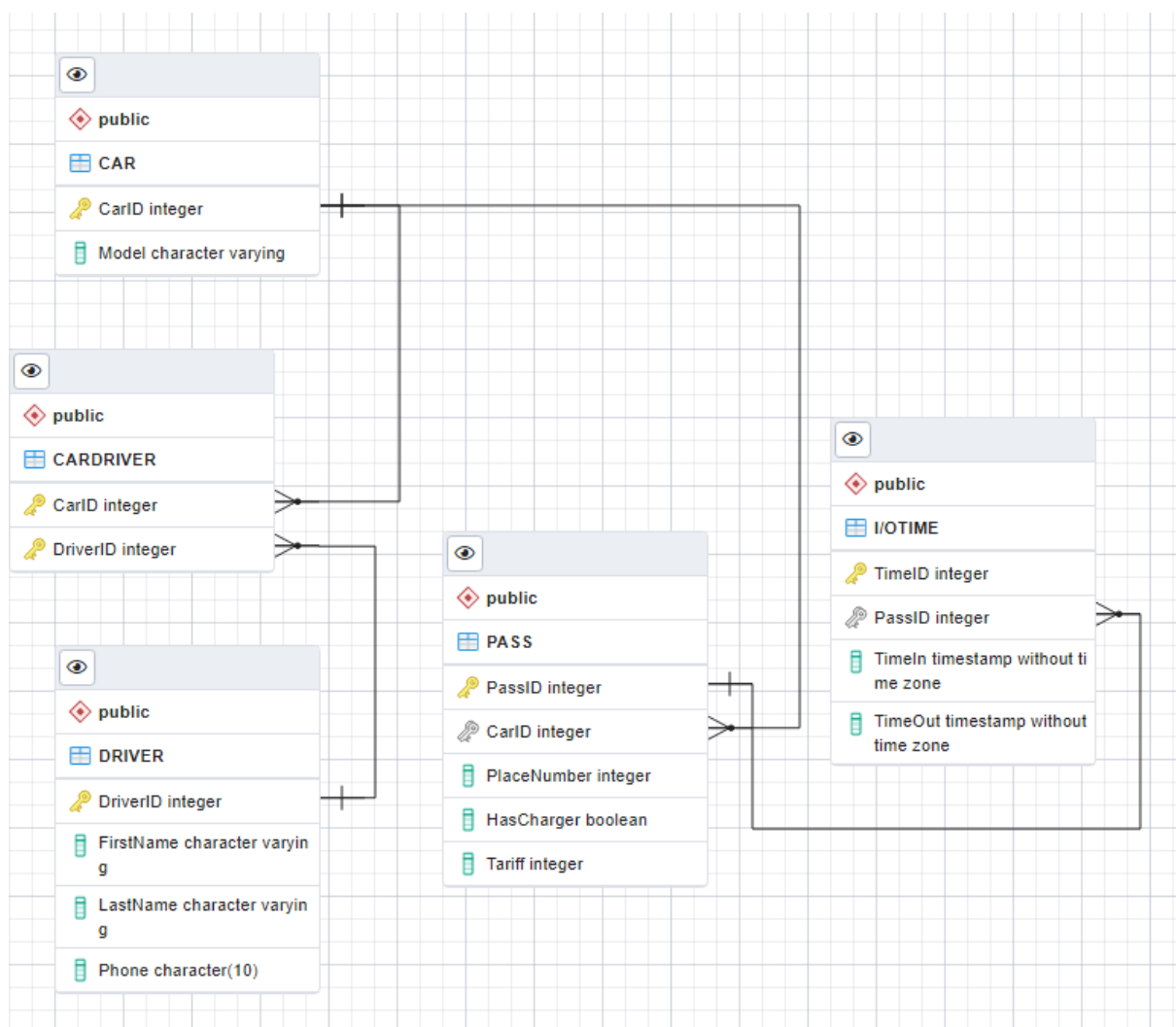


Схема бази даних у pgAdmin 4

### **Завдання 3 – Виконати нормалізацію схеми бази даних до третьої нормальної форми**

Car(CarID, Model):

Крок 1:

- Первинний ключ – CarID
- Кандидат – відсутній

Крок 2:

- CarID -> Model

Крок 3:

- НФ1 – всі атрибути таблиці атомарні
- НФ2 – відповідає НФ1 і не ключові атрибути залежать від первинного ключа
- НФ3 – відповідає НФ2 і не ключові атрибути нетранзитивно залежать від первинного ключа

Driver(DriverID, FirstName, LastName, Phone):

Крок 1:

- Первинний ключ – DriverID
- Кандидат – відсутній

Крок 2:

- DriverID -> FirstName, LastName, Phone
- DriverID -> FirstName
- DriverID -> LastName
- DriverID -> Phone

Крок 3:

- НФ1 – всі атрибути таблиці атомарні
- НФ2 – відповідає НФ1 і не ключові атрибути залежать від первинного ключа
- НФ3 – відповідає НФ2 і не ключові атрибути нетранзитивно залежать від первинного ключа

CarDriver(CarID, DriverID):

Крок 1:

- Первинний ключ – CarID, DriverID
- Кандидат – відсутній

Крок 2:

- CarID-> DriverID
- DriverID -> CarID

Крок 3:

- НФ1 – всі атрибути таблиці атомарні
- НФ2 – відповідає НФ1 і не ключові атрибути відсутні
- НФ3 – відповідає НФ2 і не ключові атрибути відсутні

Pass(PassID, CarID, PlaceNumber, HasCharger, Tariff):

Крок 1:

- Первинний ключ – PassID
- Кандидат – CarID, PlaceNumber

Крок 2:

- PassID -> CarID, PlaceNumber, HasCharger, Tariff
- PassID -> CarID
- PassID -> PlaceNumber
- PassID -> HasCharger
- PassID -> Tariff
- PlaceNumber -> HasCharger
- PlaceNumber -> Tariff
- PassID -> PlaceNumber -> HasCharger
- PassID -> PlaceNumber -> Tariff

Крок 3:

- НФ1 – всі атрибути таблиці атомарні
- НФ2 – відповідає НФ1 і не ключові атрибути залежать від первинного ключа
- НФ3 – відповідає НФ2 і не ключові атрибути нетранзитивно залежать від первинного ключа або ж транзитивно, але через ключовий атрибут

I/OTime(TimeID, PassID, TimeIn, TimeOut):

Крок 1:

- Первинний ключ – TimeID
- Кандидат – відсутній

Крок 2:

- TimeID -> PassID, TimeIn, TimeOut
- TimeID -> PassID
- TimeID -> TimeIn
- TimeID -> TimeOut

Крок 3:

- НФ1 – всі атрибути таблиці атомарні
- НФ2 – відповідає НФ1 і не ключові атрибути залежать від первинного ключа
- НФ3 – відповідає НФ2 і не ключові атрибути нетранзитивно залежать від первинного ключа

## Завдання 4 – Ознайомитись із інструментарієм PostgreSQL та pgAdmin4

### Налаштування таблиці Car:

CAR

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	CarID	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	Model	character varying			<input type="checkbox"/>	<input type="checkbox"/>	

CAR

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

	Name	Columns
	CAR_pkey	CarID

CAR

General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

	Name	Columns
	CarID	CarID

### Налаштування таблиці Driver:

DRIVER

General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s)

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	DriverID	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	FirstName	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	LastName	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Phone	character	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	



DRIVER

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns
		DRIVER_pkey	DriverID

DRIVER

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns
		DriverID	DriverID

## Налаштування таблиці CarDriver:

CARDRIVER

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s) 

Select to inherit from...

Columns

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

CARDRIVER

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns
		CARDRIVER_pkey	CarID,DriverID

CARDRIVER

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns	Referenced Table
		CARDRIVER_fkey1	(CarID) -> (CarID)	public.CAR
		CARDRIVER_fkey2	(DriverID) -> (DriverID)	public.DRIVER

# Налаштування таблиці Pass:

PASS

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	PassID	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	CarID	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	PlaceNumber	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	HasCharger	boolean			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Tariff	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

PASS

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns
	PASS_pkey	PassID

PASS

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns	Referenced Table
	PASS_fkey2	(CarID) -> (CarID)	public.CAR

PASS

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns
	PassID	PassID

## Налаштування таблиці I/OTime:

I/OTIME



General Columns Advanced Constraints Parameters Security SQL

Inherited from table(s)

Select to inherit from...



Columns



	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	TimeID	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	PassID	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	TimeIn	timestamp without tim...			<input type="checkbox"/>	<input type="checkbox"/>	
	TimeOut	timestamp without tim...			<input type="checkbox"/>	<input type="checkbox"/>	

I/OTIME



General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

	Name	Columns
	I/OTime_pkey	TimeID

I/OTIME



General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

	Name	Columns	Referenced Table
	I/OTIME_fkey1	(PassID) -> (PassID)	public.PASS

I/OTIME



General Columns Advanced Constraints Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

	Name	Columns
	TimeID	TimeID

Вміст таблиць бази даних:

```
LAB1=# SELECT * FROM public."CAR";
```

CarID	Model
1	Audi
2	Volkswagen
4	BMW
5	Mercedes Benz
3	Tesla

(5 rows)

```
LAB1=# SELECT * FROM public."DRIVER";
```

DriverID	FirstName	LastName	Phone
22	Alex	Brown	987612345
11	Jhon	Smith	5432167890
55	Joanna	Smith	987654321

(3 rows)

```
LAB1=# SELECT * FROM public."CARDRIVER";
```

CarID	DriverID
1	55
1	11
2	55
3	11
4	22
5	22

(6 rows)

```
LAB1=# SELECT * FROM public."PASS";
```

PassID	CarID	PlaceNumber	HasCharger	Tariff
1	3	1	t	10
2	1	9	f	8
3	2	5	f	8
4	5	7	f	6

(4 rows)

```
LAB1=# SELECT * FROM public."I/OTIME";
```

TimeID	PassID	TimeIn	TimeOut
1	2	2022-09-01 10:01:21	2022-09-01 18:10:32
2	2	2022-09-02 09:55:11	2022-09-02 18:04:01
3	1	2022-09-03 07:10:32	2022-09-03 20:05:10
4	2	2022-09-03 09:41:12	2022-09-03 19:00:50

(4 rows)