

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 Базис даних і засоби управління
на тему:
“Проектування бази даних та ознайомлення з базовими операціями СУБД
PostgreSQL”

Виконав:
студент 3 курсу
групи КВ-03
Стецюренко І. С.
Перевірів:
Петрашенко А. В.

Київ-2022

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 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 звітування щодо пунктів 1-4 завдання:

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.**

У звіті щодо пункту №3 завдання має бути:

- пояснення щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. 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 та типи стовпців (доступне у закладці “Columns” 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!

Варіант (опис обраної предметної галузі):

Доставка товарів.

1. Сутності та їх призначення

Сутність “Customers” призначено для інформації про замовника

Сутність “Orders” призначено для інформації про замовлення, а саме хто замовник(CustomerID), компанія(ShipperID), яка буде доставляти продукти, дата замовлення(OrderDate).

Сутність “Shippers” призначено для інформації про компанію, яка буде доставляти товар.

Сутність “Products” призначено для інформації про продукти, а саме кількість доступних товарів(Unit), ціна(Price) та назви(ProductName).

Сутність “OrderDetails” призначено для додаткової інформації про замовлення, а саме який товар замовили(ProductID), кількість(Quantity) та номер замовлення(OrderID).

Модель “сутність - зв’язок” галузі “доставка продуктів”

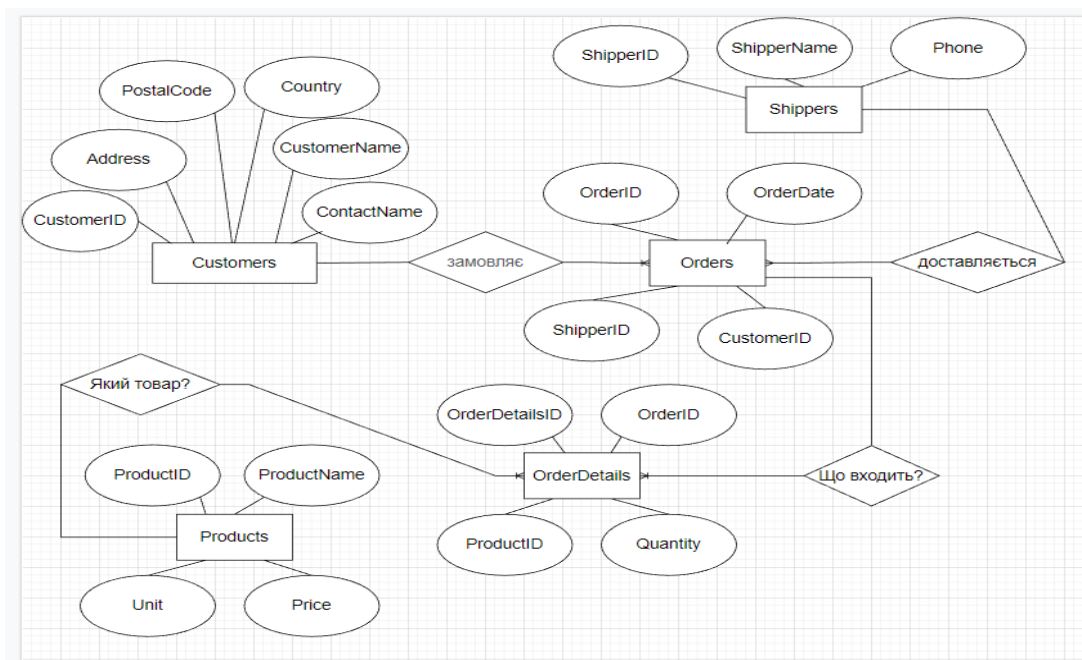


Рисунок 1. ER-діаграма, побудована за нотацією Чена

Модель «сутність-зв'язок» нарисована за допомогою інструменту <https://www.draw.io/>

2.

Опис зв'язків

При проектуванні бази даних “Доставка товарів” можна виділити такі сутності(Customers, Orders, OrderDetails, Shippers, Products,)

Клієнти (Customers) можуть робити безліч замовлень(Orders), тому між таблицями встановлено зв'язок 1:N.

Певну кількість замовлень(Orders) буде доставляти одна з компаній-перевізників(Shippers), тому між таблицями Shippers та Orders встановлено зв'язок 1:N.

Для реалізації зв'язку багато до багатьох (N:M) зумовлена поява таблиці OrderDetails, так ми зможемо дізнатись інформацію, які саме товари замовив клієнт, за номером замовлення(OrderID) та в якій кількості(Quantity).

Кожне замовлення має деяку кількість товарів в собі, тому між сутностями Orders та OrderDetails був встановлений зв'язок 1:N.

OrderDetails буде включати в себе товари із списку Products, тому між сутностями Products та OrderDetails встановлений зв'язок 1:N.

Перетворення моделі у схему бази даних

Сутність Customers була перетворена у таблицю Customers з ідентифікатором CustomerId і атрибутами CustomerName, ContactName, Address, City, PostalCode, Country.

Сутність Orders була перетворена у таблицю Orders з ідентифікатором OrderID і атрибутами OrderData, ShipperID, CustomerID.

Сутність Shippers була перетворена у таблицю Shippers з ідентифікатором ShipperID і атрибутами ShipperName, Phone.

Сутність Products була перетворена у таблицю Products з ідентифікатором ProductID і атрибутами ProductName, Price, Unit.

Щоб реалізувати зв'язок багато до багатьох між таблицями Orders та Products була створена таблиця OrderDetails, яка містить два зовнішніх ключа: OrderID для зв'язку з таблицею Orders; ProductID для зв'язку з таблицею Products; також ця таблиця містить атрибут Quantity.

Для того щоб ідентифікувати ким було зроблено замовлення у таблиці Orders маємо зовнішній ключ CustomerID.

Для того щоб дізнатися якою компанією буде доставлено замовлення в таблиці Orders маємо зовнішній ключ ShipperID.

У таблиці OrderDetails використано зовнішній ключ OrderID для ідентифікування деталей кожного замовлення. А також, зовнішній ключ ProductID, який посилається на товар, його кількість та вартість.

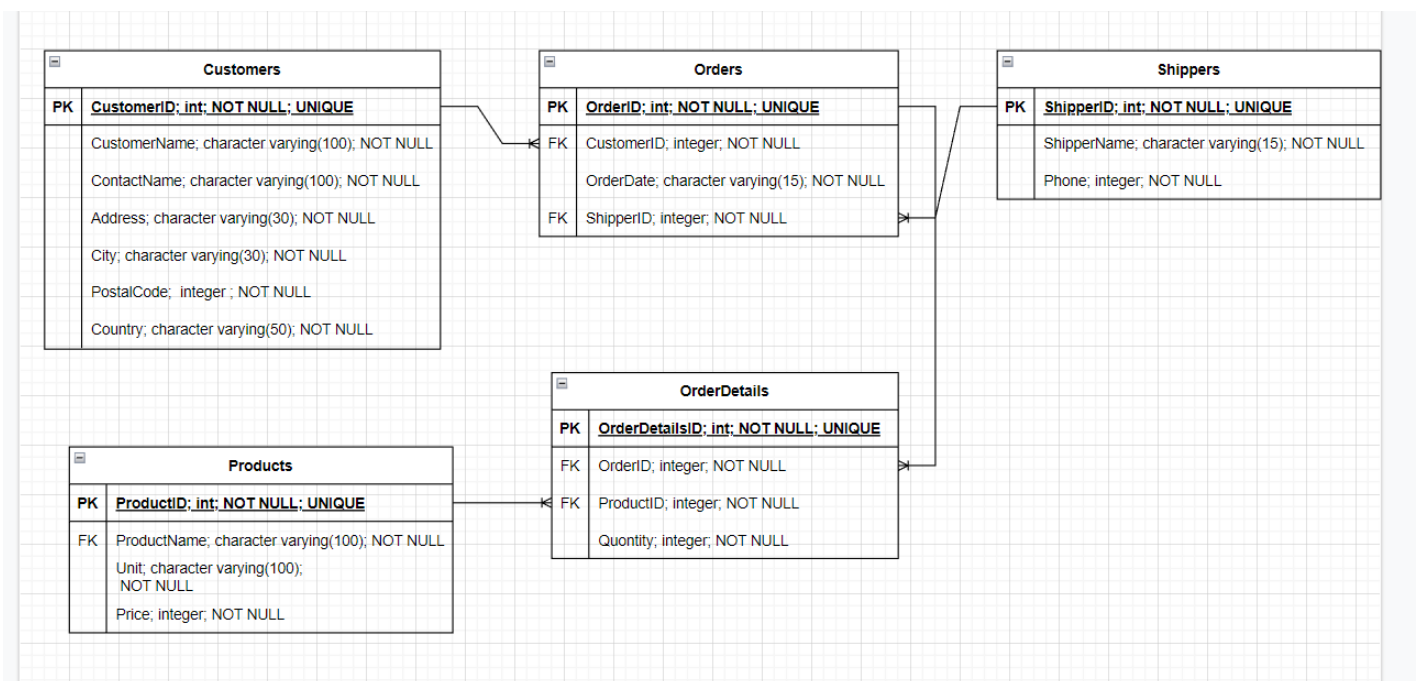


Рисунок 2 - ER-діаграма, переведена у таблиці БД

ER-діаграма нарисована за допомогою інструменту
<https://www.draw.io/>

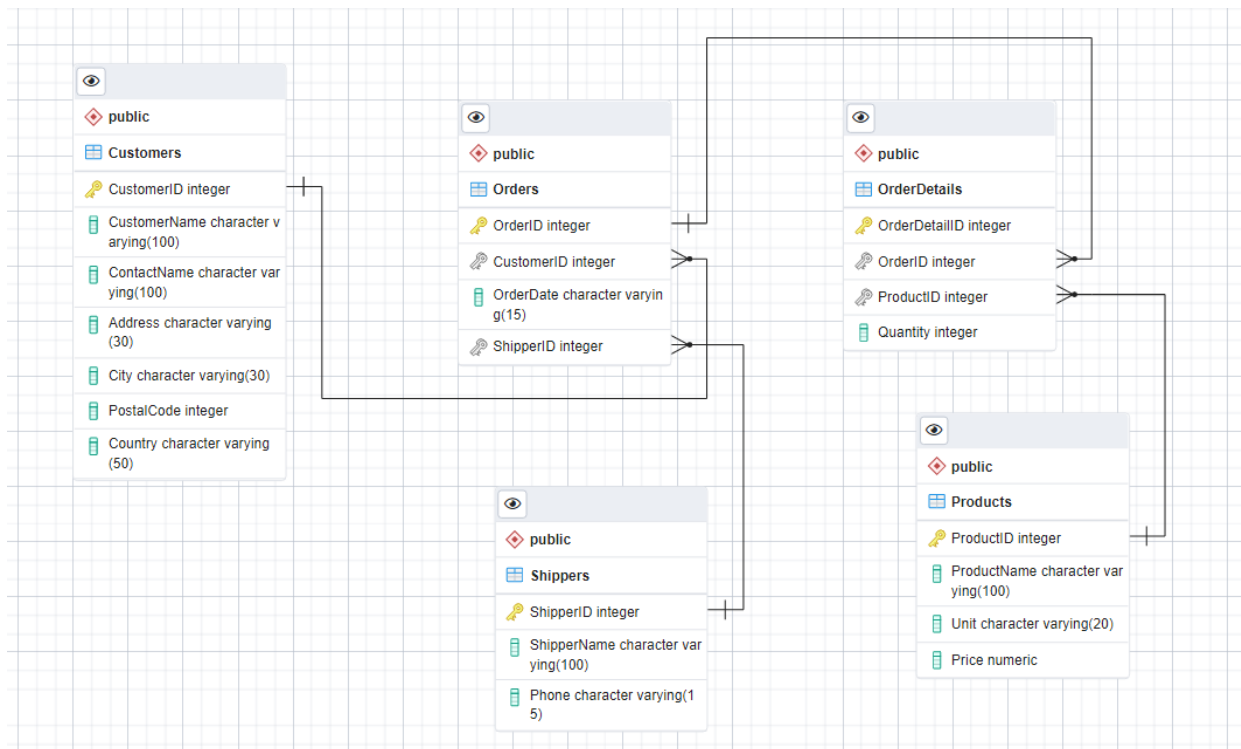


Рисунок 3. Схема бази даних у pgAdmin 4

Таблиця 1. Опис структури БД

Сутність	Атрибут	Тип атрибуту
<i>Customers</i> - містить дані про замовника	<i>CustomerID</i> - унікальний ідентифікатор клієнта <i>CustomerName</i> - ім'я та прізвище клієнта <i>ContactName</i> - ім'я та прізвище контактної особи <i>Address</i> - адрес, куди потрібно доставити <i>City</i> - місто	<i>integer</i> <i>character varying(100)</i> <i>character varying(100)</i> <i>character varying(30)</i> <i>character varying(30)</i>

	<i>PostalCode - поштовий індекс</i> <i>Country - країна</i>	<i>integer</i> <i>character varying(30)</i>
Orders - містить частину даних про замовлення	<i>OrderID - унікальний ідентифікатор замовлення</i> <i>CustomerID - зовнішній ключ для зв'язку з клієнтом</i> <i>OrderDate - дата замовлення</i> <i>ShipperID - зовнішній ключ для зв'язку з логістичною компанією</i>	<i>integer</i> <i>integer</i> <i>character varying(15)</i> <i>integer</i>
Shippers - містить дані про логістичну компанію	<i>ShipperID - унікальний ідентифікатор логістичної компанії</i> <i>ShipperName - назва компанії</i> <i>Phone - номер компанії</i>	<i>integer</i> <i>character varying(15)</i> <i>integer</i>
OrderDetails - містить додаткову інформацію про замовлення	<i>OrderDetailID - унікальний ідентифікатор</i> <i>OrderID - зовнішній ключ для</i> <i>ProductID - зовнішній ключ для</i> <i>Quantity - кількість</i>	<i>integer</i> <i>integer</i> <i>integer</i> <i>integer</i>

Products - містить інформацію про товар	<i>ProductID</i> - унікальний ідентифікатор товару <i>ProductName</i> - назва продукту <i>Unit</i> - кількість на складі <i>Price</i> - ціна	<i>integer</i> <i>character varying(100)</i> <i>character varying(100)</i> <i>integer</i>
--	---	--

3.

Функціональні залежності для кожної таблиці

Customers(*CustomerID*, *CustomerName*, *ContactName*, *Address*, *City*, *PostalCode*, *Country*):

CustomerID → *CustomerName*

CustomerID → *ContactName*

CustomerID → *Address*

CustomerID → *City*

CustomerID → *PostalCode*

CustomerID → *Country*

Orders(*OrderID*, *OrderData*, *ShipperID*, *CustomerID*):

OrderID → *OrderData*

OrderID → *ShipperID*

OrderID → *CustomerID*

Shippers(*ShippersID*, *ShipperName*, *Phone*):

ShippersID → *ShipperName*

ShippersID → *Phone*

OrderDetails(*OrderDetailID*, *OrderID*, *ProductID*, *Quantity*)

OrderDetailID → *OrderID*

OrderDetailID → *ProductID*

OrderDetailID → *Quantity*

Product(ProductID, ProductName, Price, Unit)

ProductID → ProductName

ProductID → Price

ProductID → Unit

Відповідність схеми бази даних до третьої нормальної форми

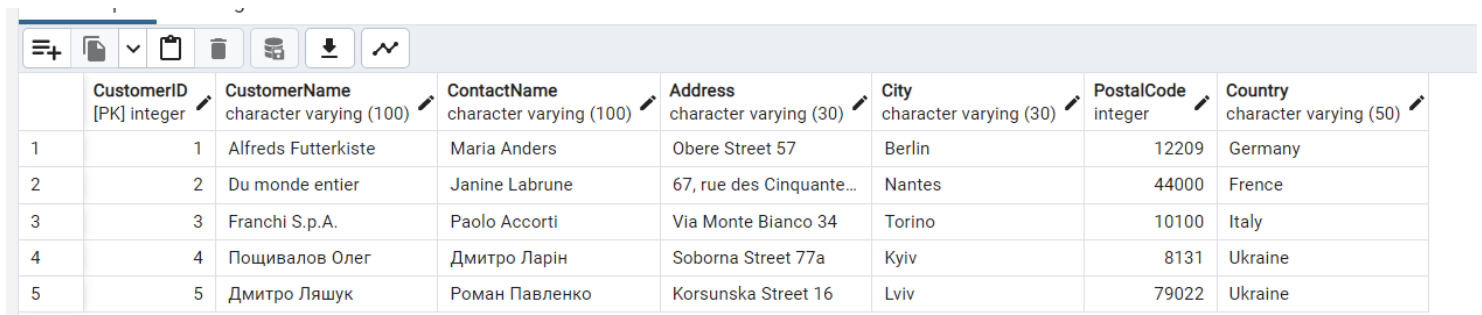
Схема бази даних відповідає нормальній формі НФ1, тому що всі атрибути таблиці є атомарними, кожна таблиця має primary key та кожна таблиця має мінімальний набір атрибутів.

Схема бази даних відповідає нормальній формі НФ2, тому що вона відповідає нормальній формі НФ1 і кожен неключовий атрибут Функціонально залежить від цілого ключа, а не від його частини.

Схема бази даних відповідає нормальній формі НФ3, тому що вона відповідає нормальній формі НФ2 і дані в таблиці залежать від primary key.

4.

Таблиці бази даних у pgAdmin4



The screenshot shows the pgAdmin4 interface with the 'Customers' table selected. The table has 8 columns: CustomerID (PK, integer), CustomerName (character varying (100)), ContactName (character varying (100)), Address (character varying (30)), City (character varying (30)), PostalCode (integer), and Country (character varying (50)). The table contains 5 rows of data.

	CustomerID [PK] integer	CustomerName character varying (100)	ContactName character varying (100)	Address character varying (30)	City character varying (30)	PostalCode integer	Country character varying (50)
1	1	Alfreds Futterkiste	Maria Anders	Obere Street 57	Berlin	12209	Germany
2	2	Du monde entier	Janine Labrune	67, rue des Cinquante...	Nantes	44000	France
3	3	Franchi S.p.A.	Paolo Accorti	Via Monte Bianco 34	Torino	10100	Italy
4	4	Пошивалов Олег	Дмитро Ларін	Soborna Street 77a	Kyiv	8131	Ukraine
5	5	Дмитро Ляшук	Роман Павленко	Korsunsk Street 16	Lviv	79022	Ukraine

Рисунок 4. Дані таблиці Customers

	OrderID [PK] integer	CustomerID integer	OrderDate character varying (15)	ShipperID integer
1	1	1	24.07.2022	3
2	2	2	10.08.2022	3
3	3	3	15.08.2022	3
4	4	4	19.08.2022	2
5	5	5	27.08.2022	1
6	6	3	31.08.2022	2
7	7	1	5.09.2022	3
8	8	5	12.09.2022	1

Рисунок 5. Дані таблиці Orders

	OrderDetailID [PK] integer	OrderID integer	ProductID integer	Quantity integer
1	1	1	7	1
2	2	1	12	3
3	3	1	16	5
4	4	1	20	1
5	5	2	4	3
6	6	2	8	2
7	7	2	10	1
8	8	2	21	1
9	9	2	17	2
10	10	3	24	1
11	11	3	23	1
12	12	4	2	1
13	13	4	13	5
14	14	4	14	2
15	15	4	15	1
16	16	5	8	2
17	17	5	9	3
18	18	5	19	5
19	19	5	25	1
20	20	6	2	2
21	21	6	3	4
22	22	6	5	2
23	23	6	7	1
24	24	7	16	4
25	25	7	17	3
26	26	7	18	4
27	27	7	4	3
28	28	8	12	5
29	29	8	13	3
30	30	8	26	2

Рисунок 6. Дані таблиці OrderDetails

	ProductID [PK] integer	ProductName character varying (100)	Unit character varying (20)	Price numeric
1	1	Coca-Cola 2l	2l * 6 - 35	6
2	2	Coca-Cola ZERO 1.5l	1.5l * 6 - 22	4.7
3	3	Coca-Cola 1.5l	1.5l - 47	0.78
4	4	Coca-Cola ZERO 1.5l	1.5l - 47	0.82
5	5	Package of Morshynsk...	1.5l * 6 - 63	3.28
6	6	Morshynska mineral w...	1.5l - 156	0.55
7	7	Sprite 1.5l x 6	1.5l * 6 - 24	4.36
8	8	Coffee beans Jacobs ...	67	15
9	9	Mountain Ceylon Peko...	15	21.65
10	10	Saucep green tea 1 kg	23	26.4
11	11	Nestle Nesquik with w...	58	2.95
12	12	Nestle Lion Flakes 450 g	58	2.98
13	13	Nestle Fitness with vit...	58	4.2
14	14	Art Foods Basmati rice...	63	4.3
15	15	Pere Bulgur groats 800 g	72	2.95
16	16	Chips Pringles Cheese ...	58	3.57
17	17	Winway roasted cashe...	12	3.09
18	18	Winway roasted pistac...	36	2.41
19	19	Millennium dark choco...	26	14.97
20	20	Napoleon cake 1 kg	10	15.61
21	21	Cake Snickers 1 kg	7	16.21
22	22	Trout fillet 1 kg	5	27.07
23	23	Freshly frozen mackere...	12	5.14
24	24	Salmon steak in vacuu...	3	22.19
25	25	Elovena galettes with g...	54	3.11
26	26	Milk 2.5% VILLA MILK ...	30	1.1

Рисунок 7. Дані таблиці Products

	ShipperID [PK] integer	ShipperName character varying (100)	Phone character varying (15)
1	1	Nova Poshta	+380984500609
2	2	Ukrposhta	0800300545
3	3	Meest	+380504327707

Рисунок 8. Дані таблиці Shippers



Рисунок 9. Всі таблиці в PgAdmin4

SQL-текст опису БД

```
1. -- This script was generated by a beta version of the ERD tool
   in pgAdmin 4.
2. -- Please log an issue at
   https://redmine.postgresql.org/projects/pgadmin4/issues/new if
   you find any bugs, including reproduction steps.
3. BEGIN;
4.
5.
6. CREATE TABLE IF NOT EXISTS public."Customers"
7. (
8.     "CustomerID" INTEGER NOT NULL,
9.     "CustomerName" CHARACTER VARYING(100) NOT NULL,
10.    "ContactName" CHARACTER VARYING(100) NOT NULL,
11.    "Address" CHARACTER VARYING(30) NOT NULL,
12.    "City" CHARACTER VARYING(30) NOT NULL,
13.    "PostalCode" INTEGER NOT NULL,
14.    "Country" CHARACTER VARYING(50) NOT NULL,
15.    PRIMARY KEY ("CustomerID")
16. );
17.
18. CREATE TABLE IF NOT EXISTS public."Orders"
19. (
20.     "OrderID" INTEGER NOT NULL,
21.     "CustomerID" INTEGER NOT NULL,
22.     "OrderDate" CHARACTER VARYING(15) NOT NULL,
23.     "ShipperID" INTEGER NOT NULL,
24.     PRIMARY KEY ("OrderID")
25. );
26.
27. CREATE TABLE IF NOT EXISTS public."Shippers"
28. (
29.     "ShipperID" INTEGER NOT NULL,
30.     "ShipperName" CHARACTER VARYING(100) NOT NULL,
31.     "Phone" CHARACTER VARYING(15) NOT NULL,
32.     PRIMARY KEY ("ShipperID")
33. );
34.
35. CREATE TABLE IF NOT EXISTS public."OrderDetails"
36. (
37.     "OrderDetailID" INTEGER NOT NULL,
38.     "OrderID" INTEGER NOT NULL,
39.     "ProductID" INTEGER NOT NULL,
40.     "Quantity" INTEGER NOT NULL,
41.     PRIMARY KEY ("OrderDetailID")
42. );
43.
44. CREATE TABLE IF NOT EXISTS public."Products"
45. (
46.     "ProductID" INTEGER NOT NULL,
47.     "ProductName" CHARACTER VARYING(100) NOT NULL,
48.     "Unit" CHARACTER VARYING(20) NOT NULL,
49.     "Price" NUMERIC NOT NULL,
50.     PRIMARY KEY ("ProductID")
51. );
52.
53. ALTER TABLE IF EXISTS public."Orders"
54.     ADD FOREIGN KEY ("ShipperID")
```

```

55. REFERENCES public."Shippers" ("ShipperID") MATCH SIMPLE
56. ON UPDATE NO ACTION
57. ON DELETE NO ACTION
58. NOT VALID;
59.
60.
61. ALTER TABLE IF EXISTS public."Orders"
62. ADD FOREIGN KEY ("CustomerID")
63. REFERENCES public."Customers" ("CustomerID") MATCH
SIMPLE
64. ON UPDATE NO ACTION
65. ON DELETE NO ACTION
66. NOT VALID;
67.
68.
69. ALTER TABLE IF EXISTS public."OrderDetails"
70. ADD FOREIGN KEY ("OrderID")
71. REFERENCES public."Orders" ("OrderID") MATCH SIMPLE
72. ON UPDATE NO ACTION
73. ON DELETE NO ACTION
74. NOT VALID;
75.
76.
77. ALTER TABLE IF EXISTS public."OrderDetails"
78. ADD FOREIGN KEY ("ProductID")
79. REFERENCES public."Products" ("ProductID") MATCH SIMPLE
80. ON UPDATE NO ACTION
81. ON DELETE NO ACTION
82. NOT VALID;
83.
84. END;
85.

```

[Посилання на гітхаб](#)