



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: «Проектування бази даних та  
ознайомлення з базовими операціями  
СУБД PostgreSQL»

Виконав: студент III курсу

ФПМ групи КВ-81

Рожко Д.В.

Перевірив:

Київ – 2020

## 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:*

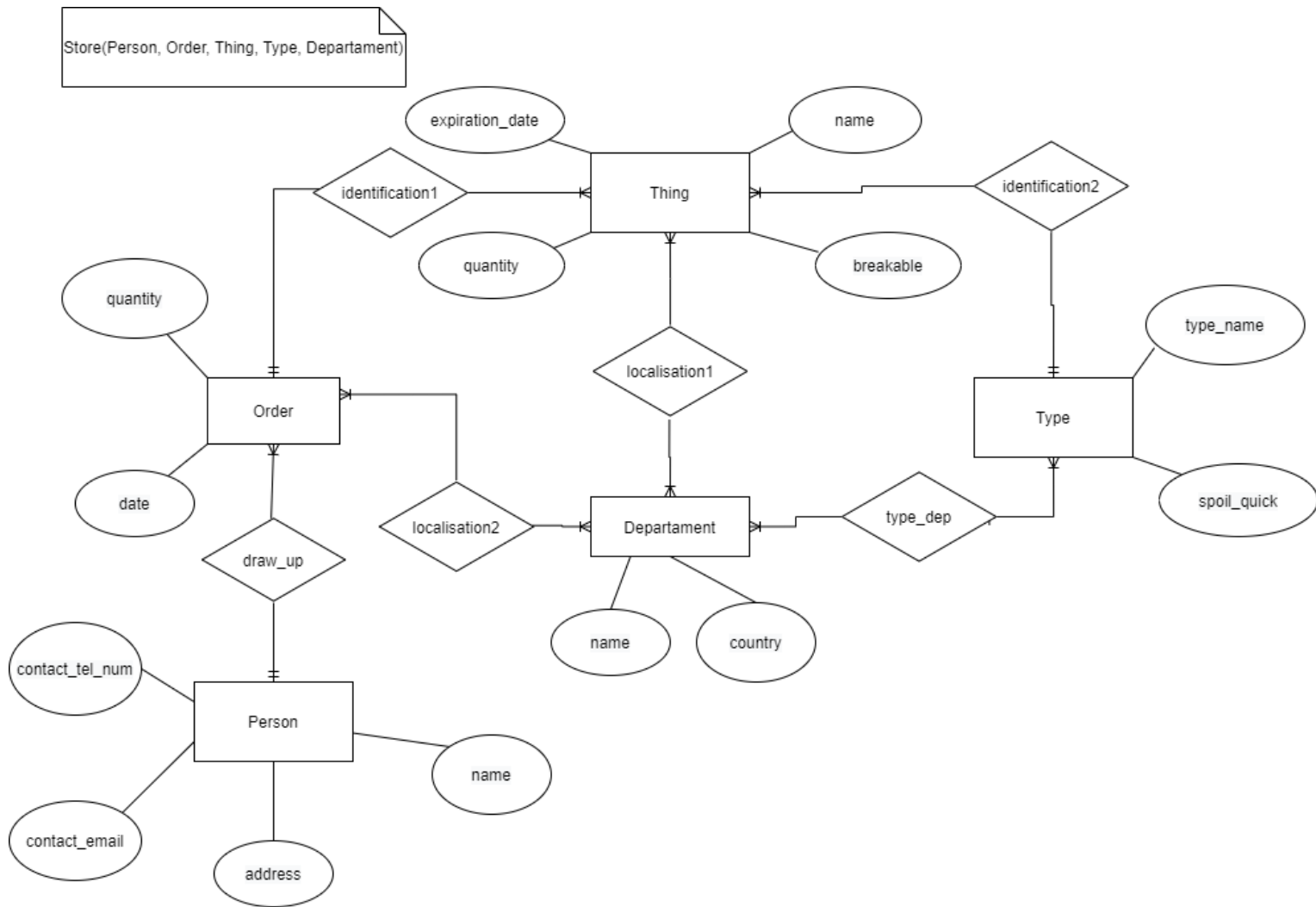
- пояснення (**обґрунтування!**) щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. У випадку невідповідності надати опис необхідних змін у схемі;
- У випадку проведення змін у схемі бази даних надати оновлену версію схеми, інакше - не наводити схему.

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

- навести копії екрану з pgAdmin4, що відображають назви, 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!**

## Обраний варіант

Варіант: магазин (товари, відділи, замовлення, категорії товарів).



### Сутності:

Сутність “**Person**” зберігає в собі данні про людину (замовника): “name”(ім’я покупця (*рядок*)), “address”(адрес, куди доставлятиметься товар (*рядок*)), “contact\_email” (електронна адреса для зв’язку з покупцем (може вказуватись тільки одна) (*рядок*)), “contact\_tel\_num” (телефон для зв’язку з покупцем (може вказуватись тільки один (*рядок*))).

Сутність “**Order**” зберігає в собі данні про замовлення: “quantity” (кількість замовленого товару (*ціле число*)), “date”(дата замовлення (*тип даних “date”*)).

Сутність “**Type**” зберігає в собі данні про тип товару: “type\_name”(ім’я товару (*рядок*)), “spoil\_quick” (чи товар швидко псується (*буловий тип*)).

Сутність “**Thing**” зберігає в собі данні про наявні товари: “name”(назва товару (*рядок*)), “quantity” (кількість товару в сенсі маса/штуки (*ціле число*)), “expiration\_date” (термін придатності/(дата виходу інших товарів)

(тип даних “date”), “breakable” (чи б’ється товар (буловий тип)). Виділено не в полі “Type”, бо і яловичина, і курячі яйці тваринного походження, але одні не б’ються, а інші навпаки).

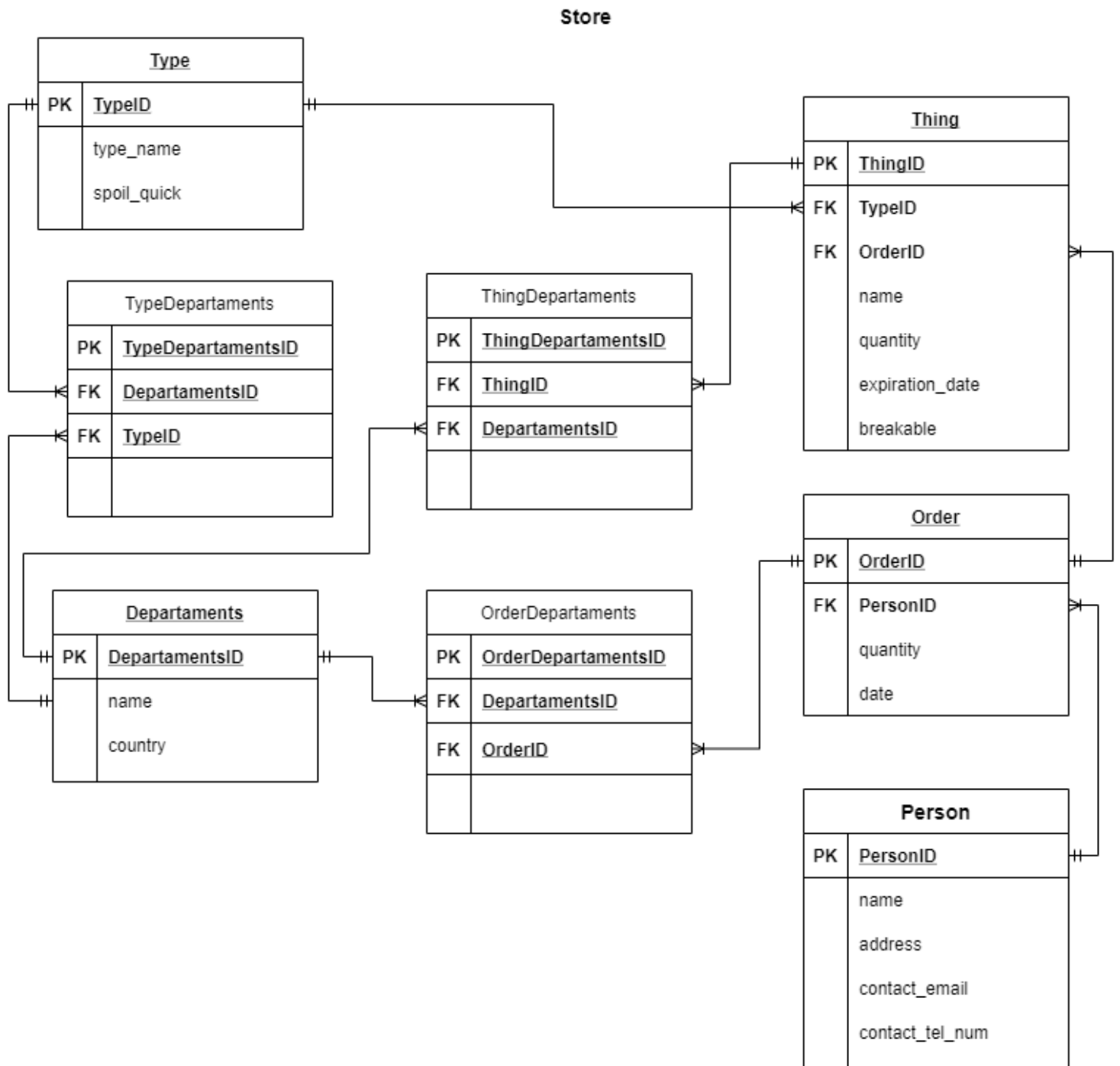
Сутність “**Departaments**” зберігає в собі данні про відділення: “name” (ім’я відділення (рядок)), “country” (країна в якій знаходиться відділення (рядок)).

### **Зв’язки:**

- “Order” і “Thing” відноситься до “Departaments” як N:M (Замовлення, тип товару і товар може знаходитись хоча б в одному відділенні, або в декількох).
- “Type” відноситься до “Departaments” як 1:N (тип товару може знаходитись хоча б в одному відділенні, або в декількох).
- “Order” і “Type” відноситься до “Thing” як N:1 (До замовлення і тип товару можуть належати багато товарів).
- “Person” відноситься до “Order” як 1:N (Одна людина може мати множину замовлень).

Для побудови моделі використано нотацію “Crow's Foot”.

## Пункт №2: «Даталогічна модель»



Сутність “Person” перетворено в таблицю “Person”.

Сутність “Order” перетворено в таблицю “Order”.

Сутність “Thing” перетворено в таблицю “Thing”.

Сутність “Type” перетворено в таблицю “Type”.

Сутність “Departaments” перетворено в таблицю “Departaments”.

Для забезпечення зв'язку між таблицями зроблено наступне:

- Для реалізації зв'язку “Type” відноситься до “Thing” було створено додаткову таблицю “ThingDepartaments”. Зроблено окремі таблиці, оскільки кожен зв'язок має характеристику багато до багатьох.
- Для реалізації зв'язку “Order” відноситься до “Thing” було створено додаткову таблицю “OrderDepartaments”. Зроблено окремі таблиці, оскільки кожен зв'язок має характеристику багато до багатьох
- Для реалізації зв'язку “Type” відноситься до “Departaments”, “TypeDepartaments”. Зроблено окрему таблицю, оскільки кожен зв'язок має характеристику багато до багатьох.
- Для реалізації зв'язку “Order” і “Type” відноситься до “Thing” використано два додаткові зовнішні ключі “OrderID” і “TypeID” відповідно в таблиці “Thing”.
- Для реалізації зв'язку “Person” відноситься до “Order” використано додатковий зовнішній ключ “PersonID” в таблиці “Order”.

### Пункт №3: «Нормалізація»

**Схема відповідає 1НФ**, тому що кожен елемент є атомарним, неподільним, що виражено у відсутності списків у всіх стовбцях всіх таблиць.

**Схема відповідає 2НФ**, тому що відповідає першій а також

Приклад. Розглянемо таблицю “Order”:

	OrderID [PK] integer	quantity integer	date date	PersonID integer
1	1	5	2000-05...	1
2	2	7	2000-08...	2
3	3	4	2001-11...	1
4	4	1	2001-03...	3
5	5	10	2000-12...	1

Кожен елемент таблиці (стовбці) не залежить від будь-якого іншого.

Первинний ключ - OrderID

*OrderID* → *quantity*

*OrderID* → *date*

*OrderID* → *PersonID*

Дане відношення має тільки один первинний ключ (*OrderID*), тому автоматично задовольняє умовам 2НФ.

Також здійснено нормалізацію доданням таблиці “Person”.

	PersonID [PK] integer	name character[] (50)	address character[] (500)	contact_email character[] (320)	contact_tel_num character[] (15)
1	1	{“Steve Martin ...	{“4244 Benson Park D...	{“steve@gmail.com ...	{“+17632274992 ”}
2	2	{“Paul McCartney ...	{“Huntington Ave & ...	{“paul@xaxaxa.com ...	{“+17633360874 ”}
3	3	{“Abraham Lincoln ...	{“4480 King St ...	{“ab_li@hohoho.com ...	{“+17456368450 ”}

Первинний ключ - *PersonID*

*PersonID* → *name*

*PersonID* → *address*

*PersonID* → *contact\_email*

*PersonID* → *contact\_tel\_num*

Цією таблицею я усунув можливі повторення даних про замовника в таблиці “Order”. Зв'язок 1:N , одна людина може мати певну кількість замовлень.

Дане відношення має тільки один первинний ключ (*PersonID*), тому автоматично задовольняє умовам 2НФ.

Розглянемо таблицю “Thing”:

	ThingID [PK] integer	name character[] (200)	quantity integer	expiration_date date	breakable boolean	TypeID integer	OrderID integer
1	1	{“computer M-1 ...	150	2001-03-25	true	2	1
2	2	{“egg ...	10000	2001-02-24	true	1	2
3	3	{“corn ...	1000	2002-02-10	false	3	2
4	4	{“tomatoes ...	5025	2000-06-05	false	3	3
5	5	{“phone F1 ...	6578	2003-06-20	true	2	4

Первинний ключ - *ThingID*

*ThingID* → *name*

*ThingID* → *quantity*

*ThingID* → *expiration\_date*

*ThingID* → *breakable*

*ThingID* → *TypeID*

*ThingID* → *OrderID*

Кожен елемент таблиці (стовбці) не залежить від будь-якого іншого. Також здійснено нормалізацію доданням таблиці “Type”.

Дане відношення має тільки один первинний ключ (ThingID), тому автоматично задовольняє умовам 2НФ.

	TypeID [PK] integer		type_name character[] (150)		spoil_quick boolean
1		1	{"animal	...	true
2		2	{"electronics	...	false
3		3	{"plants	...	true

Первинний ключ - TypeID

*TypeID* → *type\_name*

*TypeID* → *spoil\_quick*

Цією таблицею я усунув можливі повторення даних про тип товару в таблиці “Thing”. Зв'язок 1:N, один тип товару може включати в себе певне кількість товарів.

Дане відношення має тільки один первинний ключ (ThingID), тому автоматично задовольняє умовам 2НФ.

Розглянемо таблицю “Departaments”:

	DepartamentsID [PK] integer		name character[] (200)		country character[] (60)
1		1	{"shootingstar	...	{"USA
2		2	{"garage	...	{"Ukraine
3		3	{"warehouse	...	{"UK

Первинний ключ - DepartamentsID

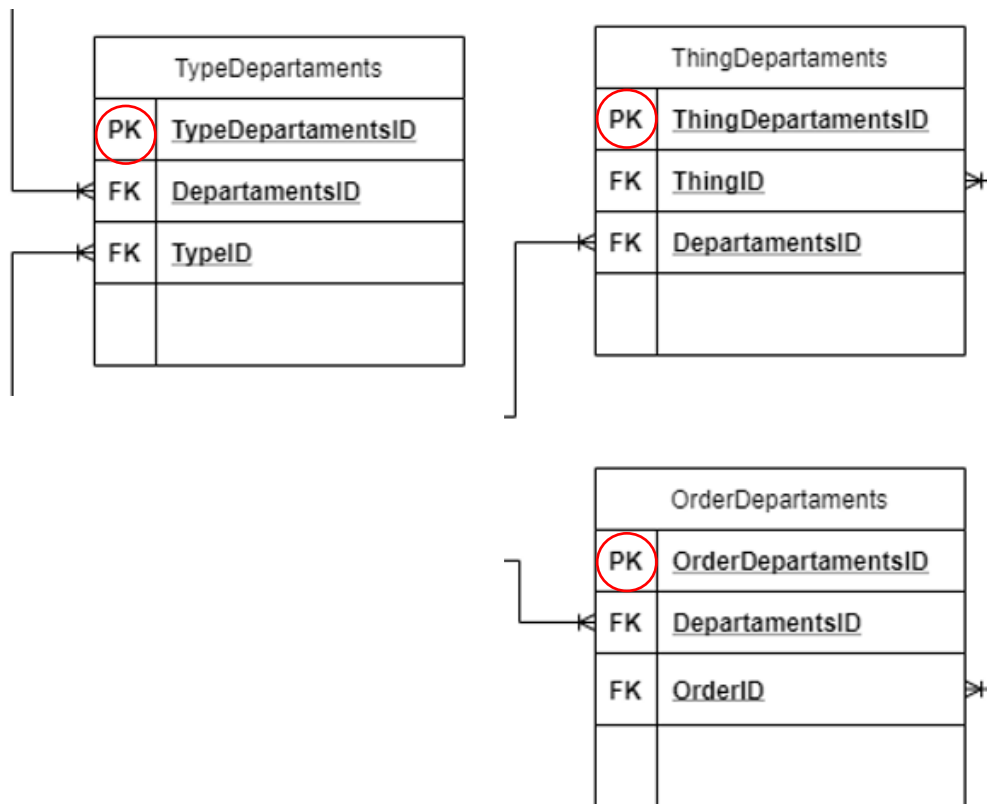
*DepartamentsID* → *name*

*DepartamentsID* → *country*

Кожен елемент таблиці (стовбці) не залежить від будь-якого іншого. Дане відношення має тільки один первинний ключ (DepartamentsID), тому автоматично задовольняє умовам 2НФ.



Таблиці, які реалізують зв'язок N:M також мають тільки один первинний ключ.



Отже, схема відповідає 1НФ, всі властивості дотримано, данні не в таблиці не дублюються в межах одної таблиці, а також в межах всієї схеми. А також кожна таблиця має тільки один первинний ключ.

**Схема відповідає 3НФ**, тому що схема відповідає 2 нормальній формі а також відсутні транзитивні залежності.

Приклад. Розглянемо таблицю “Departaments”:

DepartamentsID → neme

DepartamentsID → country

DepartamentsID → name → country

DepartamentsID → country → name

Червоним кольором показані транзитивні зв’язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з’єднань немає, то таблиця задовольняє умови 3НФ.

Приклад. Розглянемо таблицю “Type”:

TypeID → type\_neme

TypeID → spoil\_quick

TypeID → type\_neme → spoil\_quick

TypeID  $\rightarrow$  spoil\_quick  $\rightarrow$  type\_neme

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови 3НФ.

Приклад. Розглянемо таблицю "Thing":

ThingID  $\rightarrow$  neme

ThingID  $\rightarrow$  quantity

ThingID  $\rightarrow$  expiration\_date

ThingID  $\rightarrow$  breakable

ThingID  $\rightarrow$  OrderID

ThingID  $\rightarrow$  TypeID

ThingID  $\rightarrow$  neme  $\rightarrow$  quantity

ThingID  $\rightarrow$  neme  $\rightarrow$  expiration\_date

ThingID  $\rightarrow$  neme  $\rightarrow$  breakable

ThingID  $\rightarrow$  neme  $\rightarrow$  OrderID

ThingID  $\rightarrow$  neme  $\rightarrow$  TypeID

ThingID  $\rightarrow$  quantity  $\rightarrow$  name

ThingID  $\rightarrow$  quantity  $\rightarrow$  expiration\_date

ThingID  $\rightarrow$  quantity  $\rightarrow$  breakable

ThingID  $\rightarrow$  quantity  $\rightarrow$  OrderID

ThingID  $\rightarrow$  quantity  $\rightarrow$  TypeID

ThingID  $\rightarrow$  expiration\_date  $\rightarrow$  name

ThingID  $\rightarrow$  expiration\_date  $\rightarrow$  quantity

ThingID  $\rightarrow$  expiration\_date  $\rightarrow$  breakable

ThingID  $\rightarrow$  expiration\_date  $\rightarrow$  OrderID

ThingID  $\rightarrow$  expiration\_date  $\rightarrow$  TypeID

ThingID  $\rightarrow$  breakable  $\rightarrow$  name

ThingID  $\rightarrow$  breakable  $\rightarrow$  quantity

ThingID  $\rightarrow$  breakable  $\rightarrow$  expiration\_date

ThingID  $\rightarrow$  breakable  $\rightarrow$  OrderID

ThingID  $\rightarrow$  breakable  $\rightarrow$  TypeID

ThingID  $\rightarrow$  OrderID  $\rightarrow$  name

ThingID  $\rightarrow$  OrderID  $\rightarrow$  quantity

ThingID  $\rightarrow$  OrderID  $\rightarrow$  expiration\_date

ThingID  $\rightarrow$  OrderID  $\rightarrow$  breakable

ThingID  $\rightarrow$  OrderID  $\rightarrow$  TypeID

ThingID  $\rightarrow$  TypeID  $\rightarrow$  name

ThingID → TypeID → quantity

ThingID → TypeID → expiration\_date

ThingID → TypeID → breakable

ThingID → TypeID → OrderID

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови ЗНФ.

Приклад. Розглянемо таблицю “Order”:

OrderID → PersonID

OrderID → quantity

OrderID → date

OrderID → PersonID → quantity

OrderID → PersonID → date

OrderID → quantity → PersonID

OrderID → quantity → date

OrderID → date → quantity

OrderID → date → PersonID

OrderID → PersonID → quantity → date

OrderID → PersonID → date → quantity

OrderID → quantity → date → PersonID

OrderID → quantity → PersonID → date

OrderID → date → quantity → PersonID

OrderID → date → PersonID → quantity

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови ЗНФ.

Приклад. Розглянемо таблицю “Person”:

PersonID → name

PersonID → address

PersonID → contact\_email

PersonID → contact\_tel\_num

PersonID → name → address

PersonID → name → contact\_email

PersonID → name → contact\_tel\_num

PersonID → address → name

PersonID → address → contact\_email  
PersonID → address → contact\_tel\_num  
PersonID → contact\_email → name  
PersonID → contact\_email → address  
PersonID → contact\_email → contact\_tel\_num  
PersonID → contact\_tel\_num → name  
PersonID → contact\_tel\_num → address  
PersonID → contact\_tel\_num → contact\_email

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови 3НФ.

Приклад. Розглянемо таблицю “TypeDepartaments”:

TypeDepartamentsID → DepartamentsID  
TypeDepartamentsID → OrderID  
TypeDepartamentsID → DepartamentsID → OrderID  
typeDepartamentsID → OrderID → DepartamentsID

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови 3НФ.

Приклад. Розглянемо таблицю “ThingDepartaments”:

ThingDepartamentsID → ThingID  
ThingDepartamentsID → DepartamentsID  
ThingDepartamentsID → DepartamentsID → ThingID  
ThingDepartamentsID → ThindID → DepartamentsID

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови 3НФ.

Приклад. Розглянемо таблицю “OrderDepartaments”:

OrderDepartamentsID → OrderID  
OrderDepartamentsID → DepartamentsID  
OrderDepartamentsID → DepartamentsID → OrderID  
OrderDepartamentsID → OrderID → DepartamentsID

Червоним кольором показані транзитивні зв'язки, які не можуть існувати в умовах даної таблиці.

Оскільки транзитивних з'єднань немає, то таблиця задовольняє умови 3НФ.

Отже, 3НФ дотримано.

#### Пункт №4: “PostgreSQL з використанням pgAdmin”

##### Person

	PersonID [PK] integer	name character[] (50)	address character[] (500)	contact_email character[] (320)	contact_tel_num character[] (15)
1	1	{"Steve Martin ...	{"4244 Benson Park D...	{"steve@gmail.com ...	{"+17632274992 "}
2	2	{"Paul McCartney ...	{"Huntington Ave & ...	{"paul@xaxaxa.com ...	{"+17633360874 "}
3	3	{"Abraham Lincoln ...	{"4480 King St ...	{"ab_li@hohoho.com ...	{"+17456368450 "}

##### Order

	OrderID [PK] integer	quantity integer	date date	PersonID integer
1	1	5	2000-05...	1
2	2	7	2000-08...	2
3	3	4	2001-11...	1
4	4	1	2001-03...	3
5	5	10	2000-12...	1

##### Departaments

	DepartamentsID [PK] integer	name character[] (200)	country character[] (60)
1	1	{"shootingstar ...	{"USA ...
2	2	{"garage ...	{"Ukraine ...
3	3	{"warehouse ...	{"UK ...

##### Thing

	ThingID [PK] integer	name character[] (200)	quantity integer	expiration_date date	breakable boolean	TypeID integer	OrderID integer
1	1	{"computer M-1 ...	150	2001-03-25	true	2	1
2	2	{"egg ...	10000	2001-02-24	true	1	2
3	3	{"corn ...	1000	2002-02-10	false	3	2
4	4	{"tomatoes ...	5025	2000-06-05	false	3	3
5	5	{"phone F1 ...	6578	2003-06-20	true	2	4

##### Type

	TypeID [PK] integer	type_name character[] (150)	spoil_quick boolean
1	1	{"animal ...	true
2	2	{"electronics ...	false
3	3	{"plants ...	true

## OrderDepartaments

	OrderDepartamentsID [PK] integer	OrderID integer	DepartamentsID integer
1	1	1	1
2	2	2	1
3	3	1	1

## ThingDepartaments

	ThingDepartamentsID [PK] integer	Thing integer	DepartamentsID integer
1	1	1	1
2	2	3	3
3	3	2	1

## TypeDepartaments

	TypeDepartamentsID [PK] integer	TypeID integer	DepartamentsID integer
1	1	1	1
2	2	1	2
3	3	2	3