

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

Национальный исследовательский ядерный университет «МИФИ»



Институт интеллектуальных кибернетических систем

Кафедра кибернетики (№ 22)

Отчёт о работе по курсу

«Базы данных (теоретические основы баз данных)»

Вариант «DHL (международная логистика)»

Выполнил	Кругликова М.В.
Группа	Б22-504
Вариант	DHL (международная логистика)
Преподаватель	Петровская А.В
Проверяющий	
Оценка	

Содержание

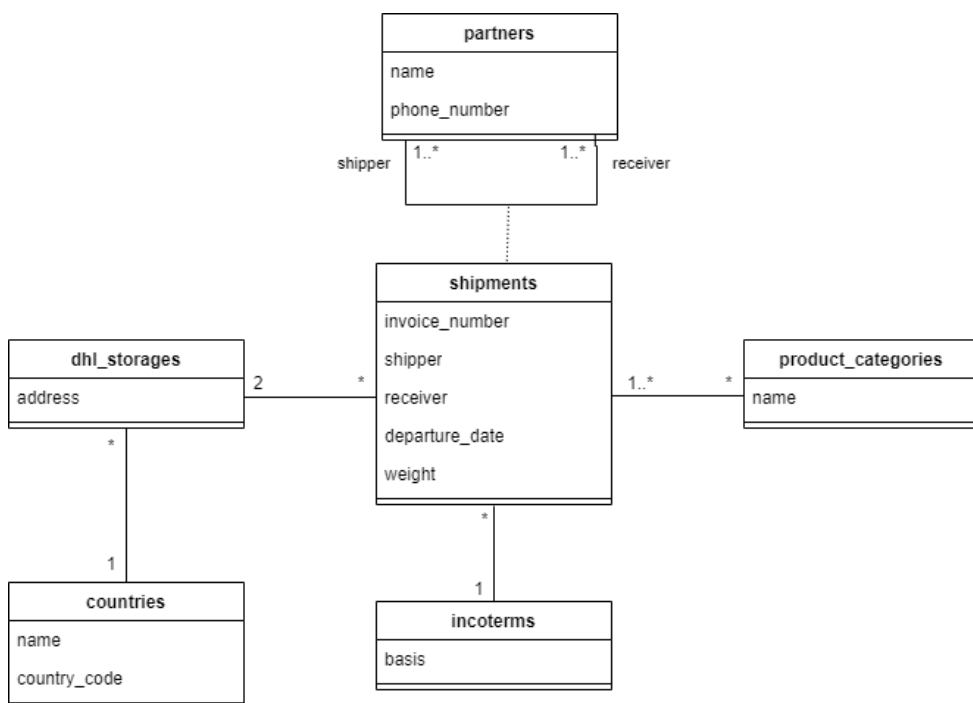
1.	Формулировка задания.....	3
2.	Концептуальная модель базы данных	3
2.1.	Конкретизация предметной области	3
2.2.	Описание предметной области.....	3
2.3.	Описание атрибутов.....	5
3.	Логическое проектирование	6
4.	Физическое проектирование	8
4.2.	Заполнение таблиц	10
5.	Выполнение запросов.....	16

1. Формулировка задания

Спроектировать базу данных для сервиса международной логистики DHL. База данных должна содержать информацию о посылках, отправителях и получателях, категориях перевозимых товаров и условиях поставок.

2. Концептуальная модель базы данных

Для создания базы данных была проанализирована предметная область, в результате чего была создана концептуальная модель:



Концептуальная модель создана в нотации UML: между сущностями установлены отношения ассоциаций, кратность которых отмечена на концах линий-ассоциаций. Отметим, что сущность shipments (посылки) является классом ассоциации между двумя объектами partners – отправителем (shipper) и получателем (receiver).

2.1. Конкретизация предметной области

В системе должны быть отображены основные сущности и связи для международной логистики, позволяющие описать процесс доставки посылок и получить информацию как об одной конкретной посылке, так и о выборке посылок, сгруппированных по некоторой характеристике.

2.2. Описание предметной области

Ключевой сущностью системы является shipments (посылки). В системе международной логистики для каждой посылки должна быть отражена следующая информация:

- отправитель и получатель посылки;
- склады отправления и получения, а также соответствующие им страны;
- список перевозимых в посылке категорий предметов (product_categories), так как информация об этом может быть необходима для прохождения таможи;
- базис Инкотермс (incoterms - International Commercial Terms), используемый для рассматриваемой посылки. Он включает в себя информацию о юридической ответственности отправителя и получателя посылки на разных этапах доставки – эта информация необходима в международной логистике;
- номер накладной (invoice_number), дата отправления (departure_date), вес (weight) посылки.

Рассмотрим несколько основных возможностей, доступных пользователю системы:

1. Получение списка товаров в посылке с заданным номером накладной.
2. Получение списка посылок, получения которых ожидает некоторый partner в настоящий момент.
3. Получение списка всех посылок в системе, которые были отправлены, но еще не доставлены.
4. Получение списка посылок, отправленных в один день, с одинаковыми странами отправления и назначения.
5. Получение списка людей, которым пришли посылки на определенный склад в определенный день.

На основе представления о предметной области и возможностях пользователя были выделены сущности, описанные в концептуальной модели:

- Посылки (shipments);
- Партнеры (partners);
- Склады DHL (dhl_storages);
- Страны (countries);
- Инкотермс (incoterms);
- Категории предметов (product_categories).

2.3. Описание атрибутов

Для сущности Посылки (shipments):

Атрибут	Описание
invoice number	Номер накладной
shipper	Партнер-отправитель
receiver	Партнер-получатель
departure_date	Дата отправления посылки
weight	Вес посылки

Для сущности Партнеры (partners):

Атрибут	Описание
name	Имя партнера
phone_number	Номер телефона партнера

Для сущности Склады DHL (dhl_storages):

Атрибут	Описание
address	Адрес склада

Для сущности Страны (countries):

Атрибут	Описание
name	Название страны
country_code	Общепринятый уникальный код из двух заглавных английских букв

Для сущности Инкотермс (incoterms):

Атрибут	Описание
basis	Базис Инкотермс – общепринятый код из трех заглавных английских букв для обозначения типа контракта, устанавливающего юридические условия поставки

Для сущности Категории предметов (product_categories):

Атрибут	Описание
name	Название категории предметов

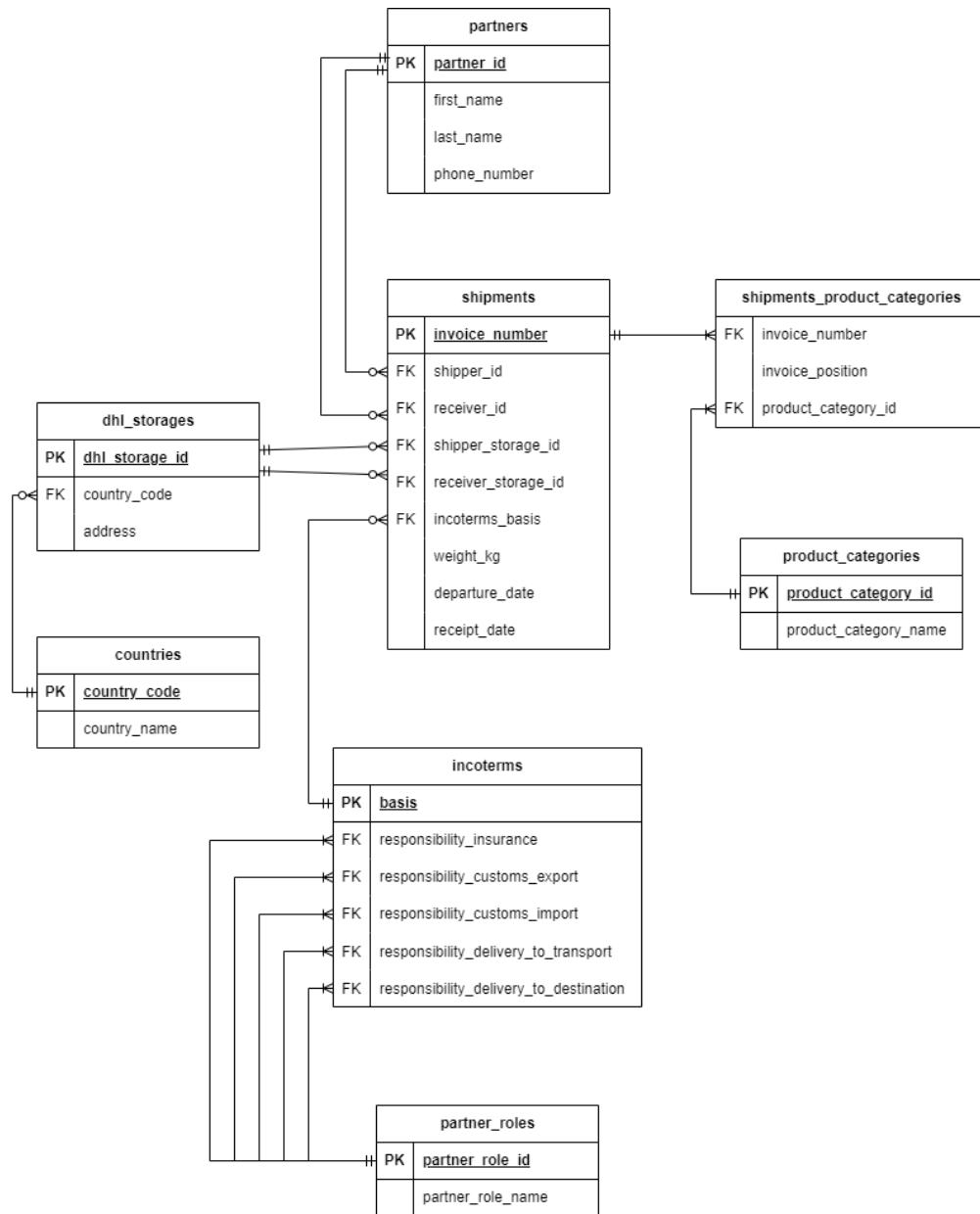
3. Логическое проектирование

На основе концептуальной модели базы данных была создана логическая модель.

Была добавлена таблица shipments_product_categories для осуществления связи многие ко многим между таблицами shipments и product_categories. Таблица shipments_product_categories содержит первичный ключ shipments (invoice_number), номер позиции предмета в накладной (invoice_position) и первичный ключ product_categories (product_category_id).

В соответствии с правилами нормализации атрибут name в таблице partners был разделен на два новых атрибута – first_name и second_name.

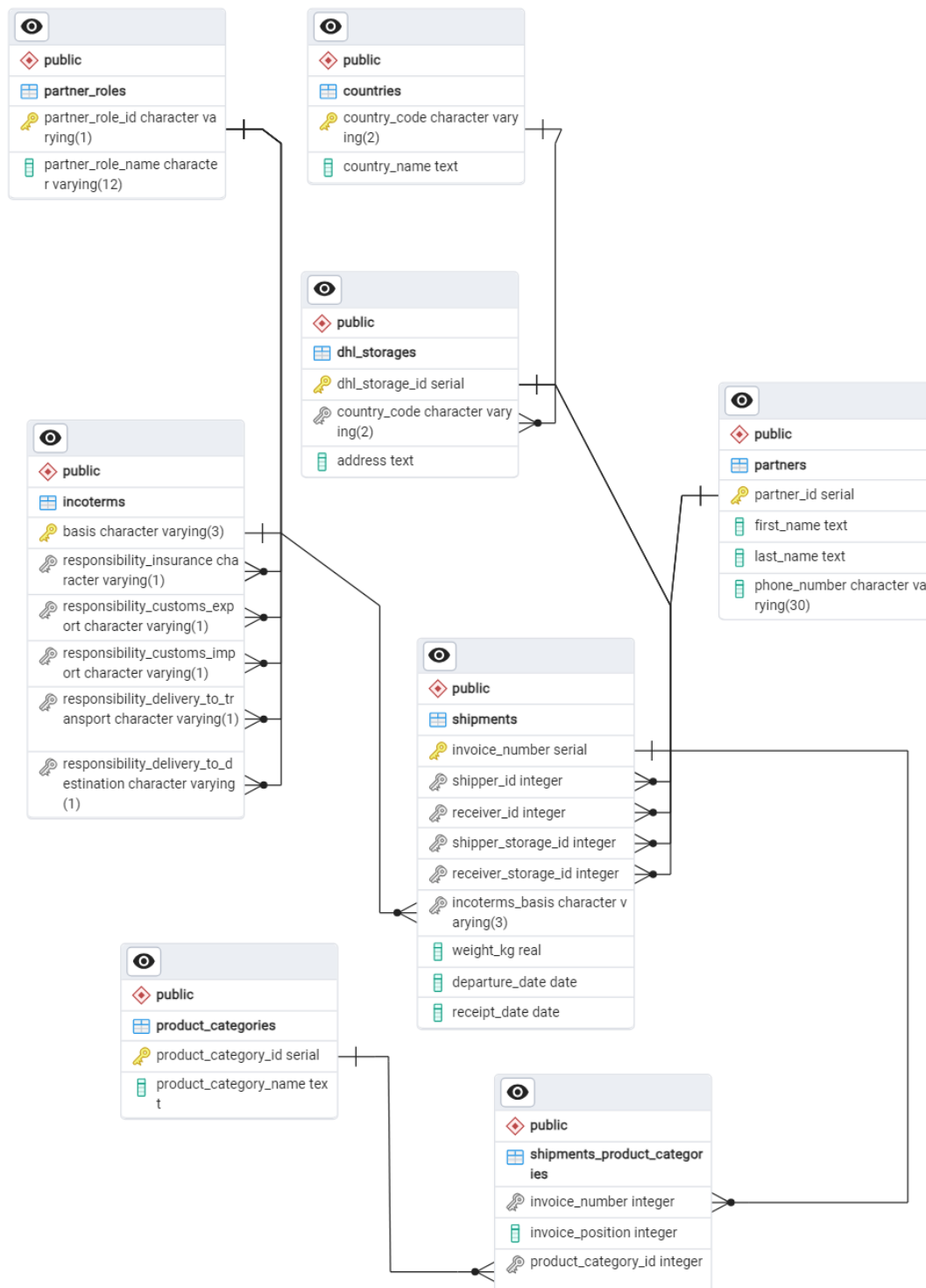
Для нескольких таблиц были добавлены атрибуты для уточнения информации, хранимой в базе данных. В частности, в таблицу incoterms были добавлены атрибуты responsibility_insurance (ответственность за страхование), responsibility_customs_export (ответственность за прохождение таможи на экспорт), responsibility_customs_import (ответственность за прохождение таможи на импорт), responsibility_delivery_to_transport (ответственность за доставку посылки до транспорта, например, в порт), responsibility_delivery_to_destinstion (ответственность за доставку посылки до места назначения). Для хранения информации, соответствующей этим атрибутам, была введена новая таблица partner_roles, описывающая возможные роли людей при поставке. В дальнейшем эта таблица будет содержать только три строки, соответствующие ролям «отправитель посылки», «получатель посылки», «не определенная роль».



4. Физическое проектирование

Для реализации базы данных была использована СУБД PostgreSQL.

На основе логической модели базы данных была создана физическая модель:



4.1. Создание таблиц

1. Создание таблицы partners

```
10 CREATE TABLE partners (  
11     partner_id SERIAL PRIMARY KEY,  
12     first_name TEXT NOT NULL,  
13     last_name TEXT NOT NULL,  
14     phone_number VARCHAR(30) NOT NULL  
15 );
```

2. Создание таблицы countries

```
17 CREATE TABLE countries (  
18     country_code VARCHAR(2) PRIMARY KEY,  
19     country_name TEXT NOT NULL  
20 );
```

3. Создание таблицы partner_roles

```
22 CREATE TABLE partner_roles (  
23     partner_role_id VARCHAR(1) PRIMARY KEY,  
24     partner_role_name VARCHAR(12) NOT NULL  
25 );
```

4. Создание таблицы incoterms

```
27 CREATE TABLE incoterms (  
28     basis VARCHAR(3) PRIMARY KEY,  
29     responsibility_insurance VARCHAR(1) NOT NULL REFERENCES partner_roles(partner_role_id),  
30     responsibility_customs_export VARCHAR(1) NOT NULL REFERENCES partner_roles(partner_role_id),  
31     responsibility_customs_import VARCHAR(1) NOT NULL REFERENCES partner_roles(partner_role_id),  
32     responsibility_delivery_to_transport VARCHAR(1) NOT NULL REFERENCES partner_roles(partner_role_id),  
33     responsibility_delivery_to_destination VARCHAR(1) NOT NULL REFERENCES partner_roles(partner_role_id)  
34 );
```

5. Создание таблицы product_categories

```
36 CREATE TABLE product_categories (  
37     product_category_id SERIAL PRIMARY KEY,  
38     product_category_name TEXT NOT NULL  
39 );
```

6. Создание таблицы dhl_storages

```
41 CREATE TABLE dhl_storages (  
42     dhl_storage_id SERIAL PRIMARY KEY,  
43     country_code VARCHAR(2) NOT NULL REFERENCES countries(country_code),  
44     address TEXT NOT NULL  
45 );
```

7. Создание таблицы shipments

```
47 CREATE TABLE shipments (  
48     invoice_number SERIAL PRIMARY KEY,  
49     shipper_id INT NOT NULL REFERENCES partners(partner_id),  
50     receiver_id INT NOT NULL REFERENCES partners(partner_id),  
51     shipper_storage_id INT NOT NULL REFERENCES dhl_storages(dhl_storage_id),  
52     receiver_storage_id INT NOT NULL REFERENCES dhl_storages(dhl_storage_id),  
53     incoterms_basis VARCHAR(3) NOT NULL REFERENCES incoterms(basis),  
54     weight_kg REAL CHECK (weight_kg > 0) NOT NULL,  
55     departure_date DATE NOT NULL,  
56     receipt_date DATE NOT NULL  
57 );
```

8. Создание таблицы shipments_product_categories

```
59 CREATE TABLE shipments_product_categories (  
60     invoice_number INT NOT NULL REFERENCES shipments(invoice_number),  
61     invoice_position INT CHECK (invoice_position > 0) NOT NULL,  
62     product_category_id INT NOT NULL REFERENCES product_categories(product_category_id)  
63 );
```

4.2. Заполнение таблиц

Для заполнения базы данных была написана программа на языке Python. Каждая таблица базы данных заполняется через вызов соответствующей функции программы. Большая часть таблиц заполняется по следующему принципу: в соответствующей функции запускается цикл, на каждой итерации которого с помощью библиотеки Faker генерируются данные для новой строки таблицы, после чего с помощью библиотеки psycopg2 происходит подключение к базе данных и добавление строки в заполняемую таблицу. Некоторые таблицы (partner_roles, incoterms), содержащие небольшое количество строк, значения которых predeterminedены, заполнялись не через цикл: в соответствующей функции программы прописывалась последовательность команд, подключающихся к базе данных и записывающих в таблицу не случайные, определенные заранее значения.

Ниже приведены все функции для заполнения таблиц и результаты их работы:

1. Заполнение таблицы partners

```
19 def insert_partners(cur, count):
20     for _ in range(count):
21         first_name = fake.first_name()
22         last_name = fake.last_name()
23         phone_number = fake.phone_number()
24
25         cur.execute("""
26             INSERT INTO "partners" (first_name, last_name, phone_number)
27             VALUES (%s, %s, %s)
28             """, (first_name, last_name, phone_number))
```

	partner_id [PK] integer	first_name text	last_name text	phone_number character varying (30)
1	1	Lori	Jordan	(540)491-8965x6750
2	2	Brent	Levy	001-950-512-0961x345
3	3	Cynthia	Gomez	898.494.2675x985
4	4	Jasmine	Spencer	629.274.0165x8949
5	5	Crystal	Berg	814.701.1046
6	6	Andrew	Coleman	488-915-5505
7	7	Christopher	Mcdaniel	497.803.7864x7067
8	8	Thomas	Edwards	001-565-430-9720x2158
9	9	Charles	Smith	(316)200-2585x553
10	10	Jose	Moore	(949)778-2770x88966
Total rows: 1000 of 100000		Query complete 00:00:00.221		

2. Заполнение таблицы countries

```
32 def insert_countries(cur, count):
33     i = 0
34     unique_country_names = set()
35     unique_country_codes = set()
36
37     while (i < count):
38         country_name = fake.country()
39         country_code = fake.country_code()
40
41         if (country_name not in unique_country_names) and (country_code not in unique_country_codes):
42             unique_country_names.add(country_name)
43             unique_country_codes.add(country_code)
44             i += 1
45
46         cur.execute("""
47             INSERT INTO "countries" (country_code, country_name)
48             VALUES (%s, %s)
49             """, (country_code, country_name))
50
51     return unique_country_codes
```

	country_code [PK] character varying (2)	country_name text
1	QA	Moldova
2	AO	Senegal
3	MH	Botswana
4	RS	Swaziland
5	GN	San Marino
6	CD	Falkland Islands (Malvinas)
7	CH	United States of America
8	SM	United States Virgin Islands
9	GW	Iran
10	DK	Lesotho
Total rows: 175 of 175 Query complete 00:00:00.170		

3. Заполнение таблицы partner_roles

```
55 def insert_partner_roles(cur):
56     cur.execute(f"""
57         INSERT INTO "partner_roles" (partner_role_id, partner_role_name)
58         VALUES ('o', 'отправитель');
59         INSERT INTO "partner_roles" (partner_role_id, partner_role_name)
60         VALUES ('n', 'получатель');
61         INSERT INTO "partner_roles" (partner_role_id, partner_role_name)
62         VALUES ('н', 'не определен')
63     """)
```

	partner_role_id [PK] character varying (1)	partner_role_name character varying (12)
1	о	отправитель
2	п	получатель
3	н	не определен
Total rows: 3 of 3		Query complete 00:00:00.100

4. Заполнение таблицы incoterms

```

67 def insert_incoterms(cur):
68     cur.execute("""
69         INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
70         VALUES ('EXW', 'H', 'N', 'N', 'N', 'N');
71     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
72     VALUES ('FCA', 'H', 'B', 'N', 'N', 'N');
73     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
74     VALUES ('FAS', 'H', 'B', 'N', 'B', 'N');
75     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
76     VALUES ('FOB', 'H', 'B', 'N', 'B', 'N');
77     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
78     VALUES ('CFR', 'H', 'B', 'N', 'B', 'N');
79     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
80     VALUES ('CIF', 'B', 'B', 'N', 'B', 'N');
81     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
82     VALUES ('CIP', 'B', 'B', 'N', 'B', 'B');
83     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
84     VALUES ('CPT', 'H', 'B', 'N', 'B', 'B');
85     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
86     VALUES ('DAP', 'H', 'B', 'N', 'B', 'B');
87     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
88     VALUES ('DPU', 'H', 'B', 'N', 'B', 'B');
89     INSERT INTO "incoterms" (basis, responsibility_insurance, responsibility_customs_export, responsibility_customs_import, responsibility_delivery_to_transport, responsibility_delivery_to_destination)
90     VALUES ('DDP', 'H', 'B', 'B', 'B', 'B')
91     """)

```

	basis [PK] character varying (3)	responsibility_insurance character varying (1)	responsibility_customs_export character varying (1)	responsibility_customs_import character varying (1)	responsibility_delivery_to_transport character varying (1)	responsibility_delivery_to_destination character varying (1)
1	EXW	H	n	n	n	n
2	FCA	H	o	n	n	n
3	FAS	H	o	n	o	n
4	FOB	H	o	n	o	n
5	CFR	H	o	n	o	n
6	CIF	o	o	n	o	n
7	CIP	o	o	n	o	o
8	CPT	H	o	n	o	o
9	DAP	H	o	n	o	o
10	DPU	H	o	n	o	o
11	DDP	H	o	o	o	o

5. Заполнение таблицы product_categories

```

95 def insert_product_categories(cur, count):
96     for _ in range(count):
97         product_category_name = fake.word()
98
99         cur.execute(f"""
100             INSERT INTO "product_categories" (product_category_name)
101             VALUES ('{product_category_name}')
102         """)

```

	product_category_id [PK] integer	product_category_name text
1	1	fish
2	2	exactly
3	3	control
4	4	difference
5	5	personal
6	6	society
7	7	wind
8	8	international
9	9	soon
10	10	minute
Total rows: 1000 of 10000		Query complete 00:00:00

6. Заполнение таблицы dhl_storages

```

106 def insert_dhl_storages(cur, count, country_codes):
107     for _ in range(count):
108         country_code = random.choice(list(country_codes))
109         address = fake.address()
110
111         cur.execute(f"""
112             INSERT INTO "dhl_storages" (country_code, address)
113             VALUES ('{country_code}', '{address}')
114         """)

```

	dhl_storage_id [PK] integer	country_code character varying (2)	address text
1	1	HU	32919 Martin Camp
2	2	KH	17228 Johnson Dam Apt. 279
3	3	RS	355 Scott Stream Suite 639
4	4	PT	2534 Kevin Overpass Apt. 154
5	5	FI	599 Jeffrey Center Apt. 908
6	6	CO	943 Michelle Well Suite 000
7	7	BO	496 Kevin Port
8	8	BT	6144 Regina Ridges Apt. 234
9	9	GA	19778 Miller Station
10	10	AT	55572 Monroe Viaduct Suite 284
Total rows: 1000 of 10000		Query complete 00:00:00.148	

7. Заполнение таблицы shipments

```
118 def insert_shipments(cur, count, num_partners, num_dhl_storages):
119     incoterms_basises = ["EXW", "FCA", "FAS", "FOB", "CFR", "CIF", "CIP", "CPT", "DAP", "DPU", "DOP"]
120
121     for _ in range(count):
122         shipper_id = random.randint(1, num_partners)
123         receiver_id = random.randint(1, num_partners)
124         shipper_storage_id = random.randint(1, num_dhl_storages)
125         receiver_storage_id = random.randint(1, num_dhl_storages)
126         incoterms_basis = random.choice(incoterms_basises)
127         weight_kg = random.uniform(0.01, 500.0)
128         departure_date = fake.date_between(d(1990, 1, 1), d(2026, 1, 1))
129         receipt_date = fake.date_between(d(1990, 1, 1), d(2026, 1, 1))
130
131         cur.execute(f"""
132             INSERT INTO "shipments" (shipper_id, receiver_id, shipper_storage_id, receiver_storage_id, incoterms_basis, weight_kg, departure_date, receipt_date)
133             VALUES ({shipper_id}, {receiver_id}, {shipper_storage_id}, {receiver_storage_id}, '{incoterms_basis}', {weight_kg}, '{departure_date}', '{receipt_date}')
134         """)
```

	invoice_number [PK] integer	shipper_id integer	receiver_id integer	shipper_storage_id integer	receiver_storage_id integer	incoterms_basis character varying (3)	weight_kg real	departure_date date	receipt_date date
1	1	89099	75280	6475	5934	CPT	492.31323	2008-05-29	2003-01-01
2	2	67245	89211	9954	288	DPU	151.77258	1994-06-04	2011-09-25
3	3	25928	80969	980	2488	DDP	436.54593	2017-11-09	1992-06-22
4	4	41337	74103	4765	761	DAP	284.00705	1995-07-04	2000-12-16
5	5	1956	61921	6253	9835	FAS	155.09282	2019-08-23	2006-12-09
6	6	87087	28337	4752	7666	DDP	142.44739	1991-10-07	1993-06-28
7	7	70211	57344	6422	6702	FCA	490.4546	1999-08-24	1995-03-08
8	8	59115	82806	9400	5722	DDP	13.8317	1999-08-06	2006-06-27
9	9	91753	49936	3628	3377	DDP	192.06828	2000-01-04	2000-02-14
10	10	9014	24342	9449	1988	DDP	39.148556	2017-09-30	1998-03-08
Total rows: 1000 of 1000000 Query complete 00:00:01.695									

8. Заполнение таблицы shipments_product_categories

```
138 def insert_shipments_product_categories(cur, num_shipments_product_categories, num_positions, num_shipments, num_product_categories):
139     for _ in range(num_shipments_product_categories):
140         invoice_number = random.randint(1, num_shipments)
141         invoice_position = random.randint(1, num_positions)
142         product_category_id = random.randint(1, num_product_categories)
143
144         cur.execute(f"""
145             INSERT INTO "shipments_product_categories" (invoice_number, invoice_position, product_category_id)
146             VALUES ({invoice_number}, {invoice_position}, {product_category_id})
147         """)
```

	invoice_number integer	invoice_position integer	product_category_id integer
1	610381	4	3014
2	839336	10	8759
3	982954	5	9093
4	348291	1	4900
5	162039	7	3571
6	572623	2	2662
7	368709	7	7770
8	102275	2	8671
9	268583	4	6563
10	279242	7	5578
Total rows: 1000 of 10000000		Query complete 00:00:05.101	

5. Выполнение запросов

К разработанной базе данных были написаны следующие запросы:

1. Вывести список товаров в посылке с заданным номером накладной и номера их позиций в накладной

```
SELECT invoice_position, product_category_name
FROM shipments_product_categories INNER JOIN product_categories USING(product_category_id)
WHERE (invoice_number = 516)
ORDER BY invoice_position;
```

	invoice_position integer	product_category_name text
1	1	value
2	1	somebody
3	1	foreign
4	1	land
5	4	adult
6	8	thousand

В реальной жизни в одной посылке каждому номеру позиции (invoice_position) соответствует только одна категория продукта (product_category_name). Однако из-за случайности генерируемых данных в приведенном запросе одному и тому

же значению `invoice_position = 1` соответствует несколько значение `product_category_name` – это не является ошибкой, связанной с БД или СУБД.

2. Вывести список посылок, которые ожидает partner с заданным `id`, упорядоченные в порядке возрастания даты получения

```
SELECT *
FROM shipments
WHERE (receiver_id = 43659 AND receipt_date > CURRENT_DATE)
ORDER BY receipt_date;
```

	invoice_number [PK] integer	shipper_id integer	receiver_id integer	shipper_storage_id integer	receiver_storage_id integer	incoterms_basis character varying (3)	weight_kg real	departure_date date	receipt_date date
1	395841	95399	43659	2953	7597	FAS	446.34372	2007-10-20	2025-04-13
2	990607	33209	43659	2256	7481	DDP	426.86475	1999-11-19	2025-06-17
3	791173	48534	43659	4338	7623	EXW	262.48456	1996-08-29	2025-09-15

3. Вывести список поставляемых в заданную страну категорий товаров и количество посылок для каждой категории товара

```
SELECT product_category_name, COUNT(product_category_id) AS count_product_category
FROM shipments
INNER JOIN dhl_storages ON (shipments.receiver_storage_id = dhl_storages.dhl_storage_id)
INNER JOIN countries USING(country_code)
INNER JOIN shipments_product_categories USING(invoice_number)
INNER JOIN product_categories USING(product_category_id)
WHERE (countries.country_name = 'Marshall Islands')
GROUP BY product_category_id, product_category_name
ORDER BY count_product_category;
```

	product_category_name text	count_product_category bigint
1	stay	1
2	bank	1
3	purpose	1

Total rows: 1000 of 9969 Query complete 00:00:00.510

	product_category_name text	count_product_category bigint
295	TV	2
296	radio	2
297	training	2



Total rows: 1000 of 9969 Query complete 00:00:00.510

Для иллюстрации того, что `count_product_category` может принимать разные значения в результате выполнения запроса, приведены примеры строк, в которых

значение count_product_category равно 1 и 2, но это не единственные возможные значения.

4. Вывести список 10 самых отправляемых категорий товаров для определенного человека и количество посылок для каждой категории товара

```
WITH product_ids AS (  
    SELECT product_category_id, COUNT(product_category_id) AS count_product_category  
    FROM shipments_product_categories INNER JOIN shipments USING(invoice_number)  
    WHERE (shipments.shipper_id = 11)  
    GROUP BY product_category_id  
)  
SELECT product_category_name, count_product_category  
FROM product_categories INNER JOIN product_ids USING(product_category_id)  
ORDER BY count_product_category DESC  
LIMIT 10;
```

	product_category_name 	count_product_category 
	text	bigint
1	air	2
2	capital	2
3	sell	1
4	boy	1
5	force	1
6	pretty	1
7	him	1
8	production	1
9	visit	1
10	cultural	1
Total rows: 10 of 10		Query complete 00:00:00.815

5. Получить информацию о человеке, который чаще всего отправляет заданную категорию товара некоторому известному другому человеку

```
WITH shipments_product_ids AS (  
    SELECT *  
    FROM product_categories INNER JOIN shipments_product_categories USING(product_category_id)  
    WHERE (product_category_name = 'plan')  
) , required_partner AS (  
    SELECT shipper_id AS partner_id  
    FROM shipments_product_ids INNER JOIN shipments USING(invoice_number)  
    WHERE (receiver_id = 86)  
    GROUP BY partner_id  
    ORDER BY COUNT(shipper_id) DESC  
    LIMIT 1  
)  
SELECT *  
FROM partners INNER JOIN required_partner USING(partner_id);
```

	partner_id [PK] integer	first_name text	last_name text	phone_number character varying (30)
1	11852	Eric	Cohen	620-769-7914

6. Вывести частоту использования каждого базиса incoterms в % в порядке убывания

```
WITH num_of_shipments AS (  
    SELECT COUNT(*)::FLOAT AS num  
    FROM shipments  
)  
SELECT incoterms_basis, (100 * COUNT(*)/num) AS percent_of_using  
FROM shipments, num_of_shipments  
GROUP BY incoterms_basis, num  
ORDER BY percent_of_using DESC;
```

	incoterms_basis character varying (3) 🔒	percent_of_using double precision 🔒
1	CPT	9.1425
2	DDP	9.1255
3	FAS	9.1165
4	FCA	9.1154
5	CIF	9.1018
6	EXW	9.0899
7	CFR	9.0729
8	CIP	9.0652
9	DAP	9.0613
10	DPU	9.0599
11	FOB	9.0491
Total rows: 11 of 11		Query complete 00:00:00.369

7. Вывести список номеров накладных посылок, которые отправлены, но еще не доставлены, с указанием дат и стран отправления и назначения

```
WITH select_shipper_country_code AS (
    SELECT invoice_number, receiver_storage_id, departure_date, receipt_date, country_code AS shipper_country_code
    FROM shipments INNER JOIN dhl_storages ON (shipments.shipper_storage_id = dhl_storages.dhl_storage_id)
    WHERE (departure_date < CURRENT_DATE AND receipt_date > CURRENT_DATE)
), select_receiver_country_code AS (
    SELECT invoice_number, departure_date, receipt_date, shipper_country_code, country_code AS receiver_country_code
    FROM select_shipper_country_code INNER JOIN dhl_storages ON (select_shipper_country_code.receiver_storage_id = dhl_storages.dhl_storage_id)
), select_shipper_country AS (
    SELECT invoice_number, departure_date, receipt_date, country_name AS shipper_country, receiver_country_code
    FROM select_receiver_country_code INNER JOIN countries ON (select_receiver_country_code.shipper_country_code = countries.country_code)
)
SELECT invoice_number, departure_date, receipt_date, shipper_country, country_name AS receiver_country
FROM select_shipper_country INNER JOIN countries ON (select_shipper_country.receiver_country_code = countries.country_code)
ORDER BY invoice_number;
```

	invoice_number integer 🔒	departure_date date 🔒	receipt_date date 🔒	shipper_country text 🔒	receiver_country text 🔒
1	56	2010-09-29	2025-12-26	Yemen	Kuwait
2	61	2003-08-10	2025-11-01	Saint Lucia	Aruba
3	78	2002-01-01	2025-08-26	Equatorial Guinea	Vanuatu
4	95	2000-01-18	2025-09-04	Kuwait	Croatia
5	123	2019-12-16	2025-05-14	Norway	Austria
6	127	2014-09-20	2025-03-23	Guernsey	Poland
7	174	2012-03-12	2025-03-23	Finland	Czech Republic
8	215	2016-10-22	2025-07-12	Cayman Islands	Canada
9	307	1991-07-22	2025-05-03	French Guiana	Russian Federation
10	357	1991-03-31	2025-01-31	Heard Island and McDonald Islands	Botswana
Total rows: 1000 of 28445		Query complete 00:00:00.148			

8. Вывести список номеров накладных посылок с одинаковыми заданными странами отправления и назначения, отправленных в заданный день

```
WITH shipper_storages AS (  
    SELECT dhl_storage_id  
    FROM countries INNER JOIN dhl_storages USING(country_code)  
    WHERE (country_name = 'Albania')  
) , receiver_storages AS (  
    SELECT dhl_storage_id  
    FROM countries INNER JOIN dhl_storages USING(country_code)  
    WHERE (country_name = 'Christmas Island')  
) , shipments_same_shipper_country AS (  
    SELECT invoice_number, shipper_id, receiver_id, shipper_storage_id, receiver_storage_id, incoterms_basis, weight_kg, departure_date, receipt_date  
    FROM shipments INNER JOIN shipper_storages ON (shipments.shipper_storage_id = shipper_storages.dhl_storage_id)  
    WHERE (departure_date = '2024-07-28')  
)  
SELECT invoice_number, shipper_id, receiver_id, shipper_storage_id, receiver_storage_id, incoterms_basis, weight_kg, departure_date, receipt_date  
FROM shipments_same_shipper_country INNER JOIN receiver_storages ON (shipments_same_shipper_country.receiver_storage_id = receiver_storages.dhl_storage_id)  
ORDER BY invoice_number;
```

	invoice_number [PK] integer	shipper_id integer	receiver_id integer	shipper_storage_id integer	receiver_storage_id integer	incoterms_basis character varying (3)	weight_kg real	departure_date date	receipt_date date
1	652040	98570	44052	3089	7594	FAS	345.24042	2024-07-28	1990-08-08

9. Вывести список людей, которым пришла посылка на заданный склад в заданную дату

```
WITH required_shipments AS (  
    SELECT receiver_id, invoice_number  
    FROM shipments INNER JOIN dhl_storages ON (shipments.receiver_storage_id = dhl_storages.dhl_storage_id)  
    WHERE (address = E'121 Rachel Divide\nLake Staceymouth, NH 42325' AND receipt_date = '1994-01-11')  
)  
SELECT receiver_id, first_name, last_name, phone_number, invoice_number  
FROM required_shipments INNER JOIN partners ON (required_shipments.receiver_id = partners.partner_id)  
ORDER BY receiver_id;
```

	receiver_id integer	first_name text	last_name text	phone_number character varying (30)	invoice_number integer
1	60629	Jesus	Howell	(271)316-6306x6938	8970
2	69177	Joy	Walton	001-419-427-7497x35828	753020

Total rows: 2 of 2	Query complete 00:00:00.273
--------------------	-----------------------------

10. Вывести список стран с указанием самой часто отправляемой категории товара для каждой страны и количества посылок с ней

```
WITH product_categories_countries AS (
    SELECT product_category_name, country_name
    FROM shipments
    INNER JOIN dhl_storages ON (shipments.shipper_storage_id = dhl_storages.dhl_storage_id)
    INNER JOIN countries USING(country_code)
    INNER JOIN shipments_product_categories USING(invoice_number)
    INNER JOIN product_categories USING(product_category_id)
), count_product_categories_countries AS (
    SELECT country_name, product_category_name, COUNT(product_category_name) AS number_shipments,
    ROW_NUMBER() OVER (
        PARTITION BY country_name
        ORDER BY COUNT(product_category_name) DESC
    ) AS maximum
    FROM product_categories_countries
    GROUP BY product_category_name, country_name
)
SELECT country_name, product_category_name, number_shipments
FROM count_product_categories_countries
WHERE maximum = 1;
```

	country_name text	product_category_name text	number_shipments bigint
1	Albania	indicate	177
2	Angola	ground	104
3	Antigua and Barbuda	ground	134
4	Argentina	ground	129
5	Aruba	road	158
6	Australia	road	160
7	Austria	road	150
8	Azerbaijan	indicate	125
9	Bahamas	road	161
10	Bahrain	arrive	94
Total rows: 175 of 175 Query complete 00:00:08.499			