

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 3

по теме:

*«Создание таблиц базы данных POSTGRESQL. Заполнение
таблиц рабочими данными»*

по дисциплине: Проектирование и реализация баз данных

Специальность:

09.03.03 Мобильные и сетевые технологии

Проверила:

Говорова М.М.

Дата: _____ 2023 г.

Оценка

Выполнил:

студент группы К3241

Чебан Илья Валерьевич

Санкт-Петербург 2023/2024

Цель работы: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

Практическое задание:

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: *Primary Key, Unique, Check, Foreign Key*.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД. *Указание:*
Создать две резервные копии:
 - с расширением *CUSTOM* для восстановления БД;
 - с расширением *PLAIN* для листинга (в отчете);
 - при создании резервных копий БД настроить параметры *Dump options* для *Type of objects* и *Queries*.
7. Восстановить БД.

Предметная область: Вариант 16. БД "Спортивный клуб"

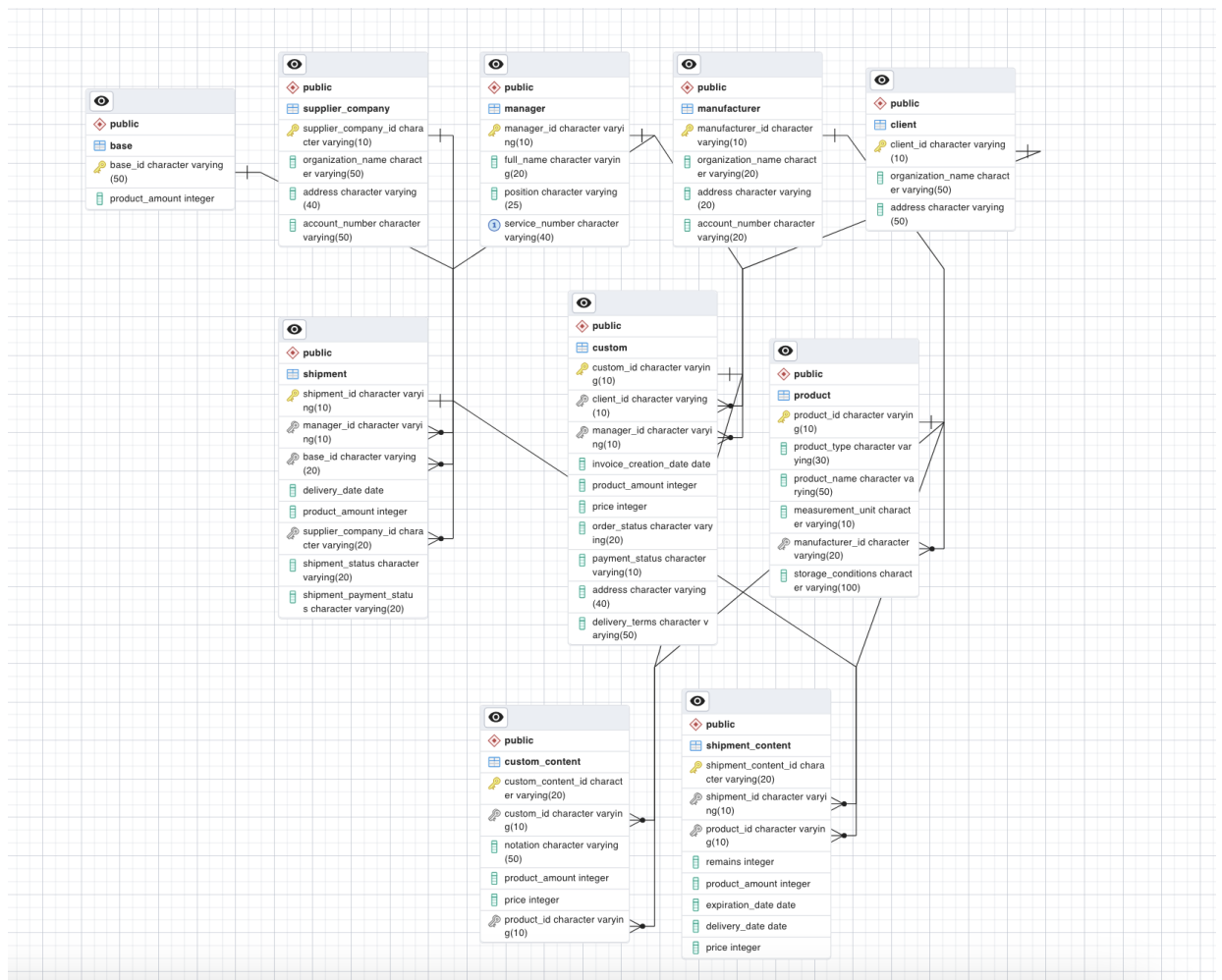


Рисунок 1 – ERD базы данных

Выполнение работы:

1. Создание таблиц

```

CREATE TABLE provider (
    provider_id VARCHAR(10) UNIQUE PRIMARY KEY,
    name_of_company VARCHAR(50) NOT NULL,
    account_number VARCHAR(50) NOT NULL,
    address VARCHAR(40) NOT NULL
);

CREATE TABLE manager (
    manager_id VARCHAR(10) NOT NULL UNIQUE PRIMARY KEY,
    full_name VARCHAR(20) NOT NULL,
    position VARCHAR(25) NOT NULL,
    service_number VARCHAR(40) NOT NULL UNIQUE
);

CREATE TABLE shipment (
    shipment_id VARCHAR(10) UNIQUE PRIMARY KEY,

```

```

provider_id VARCHAR(10),
manager_id VARCHAR(10),
base_id VARCHAR(20),
delivery_date DATE,
product_amount INTEGER CHECK (product_amount > 0) NOT NULL,
FOREIGN KEY (provider_id) REFERENCES provider(provider_id),
FOREIGN KEY (manager_id) REFERENCES manager(manager_id),
FOREIGN KEY (base_id) REFERENCES base(base_id)
);
CREATE TABLE product (
product_id VARCHAR(10) UNIQUE PRIMARY KEY,
product_type VARCHAR(30) NOT NULL,
product_name VARCHAR(50) NOT NULL,
measurement_unit VARCHAR(10) NOT NULL,
manufacturer_id VARCHAR(20),
FOREIGN KEY (manufacturer_id) REFERENCES manufacturer(manufacturer_id),
storage_conditions VARCHAR(100)
);
CREATE TABLE custom (
custom_id VARCHAR(10) UNIQUE PRIMARY KEY,
client_id BIGINT,
manager_id BIGINT,
invoice_creation_date DATE,
product_amount INTEGER CHECK (product_amount > 0) NOT NULL,
price INTEGER CHECK (price > 0) NOT NULL,
order_status VARCHAR(20),
payment_status VARCHAR(10),
address VARCHAR(40) NOT NULL,
delivery_terms VARCHAR(50),
FOREIGN KEY (client_id) REFERENCES client(client_id),
FOREIGN KEY (manager_id) REFERENCES manager(manager_id)
);
CREATE TABLE base (
base_id VARCHAR(50) UNIQUE PRIMARY KEY,
product_amount INTEGER CHECK (product_amount > 0) NOT NULL
);
CREATE TABLE shipment_content (
shipment_content_id VARCHAR(20) UNIQUE PRIMARY KEY,
shipment_id VARCHAR(10),
product_id VARCHAR(10),
remains INTEGER CHECK (remains > 0) NOT NULL,
product_amount INTEGER CHECK (product_amount > 0) NOT NULL,
expiration_date DATE NOT NULL,
delivery_date DATE NOT NULL,
price INTEGER CHECK (price >= 0) NOT NULL,
FOREIGN KEY (shipment_id) REFERENCES shipment(shipment_id),
FOREIGN KEY (product_id) REFERENCES product(product_id)
);
CREATE TABLE supplier_company (
supplier_company_id VARCHAR(10) UNIQUE PRIMARY KEY,
organization_name VARCHAR(50) NOT NULL,
address VARCHAR(40) NOT NULL,
account_number VARCHAR(50)
);

```

```
);  
CREATE TABLE client (  
  client_id VARCHAR(10) UNIQUE PRIMARY KEY,  
  organization_name VARCHAR(50) NOT NULL,  
  address VARCHAR(50) NOT NULL  
);  
CREATE TABLE manufacturer (  
  manufacturer_id VARCHAR(10) UNIQUE PRIMARY KEY,  
  organization_name VARCHAR(20) NOT NULL,  
  address VARCHAR(20) NOT NULL,  
  account_number VARCHAR(20)  
);
```

2. Вставка данных

```
INSERT INTO base (base_id, product_amount) VALUES  
( 'jfs', 3),  
( 'asd', 5);  
INSERT INTO manager (manager_id, full_name, position,  
  service_number) VALUES  
( 'asd', 'Бедыч', 'CEO', '555-1234'),  
( 'asd2', 'Janeaa', 'CFO', '555-5678'),  
( 'asd12', 'Bob', 'COO', '555-9101'),  
( 'asd123', 'Wilson', 'Marketing Director', '555-1212');
```

```
INSERT INTO client (client_id, organization_name, address) VALUES  
( 'asd', 'ABC', '3 Main St'),  
( 'asd124', 'DEF Inc', '6 Aple'),  
( 'asd126', 'GHJ', '789 Mak St'),  
( 'asd74', 'MKL', '341 Bank St');
```

Выводы:

Мы изучали основные категории данных и ограничения, используемые при создании таблиц в базах данных. Также были рассмотрены различные инструменты для резервного копирования баз данных. Мы успешно интегрировали таблицы с рабочими данными, применяя технику импорта данных из внешних ресурсов.

В процессе выполнения практической работы мы получили навыки создания таблиц в базе данных POSTGRESQL и их заполнения. Эти навыки

будут очень полезны при проектировании и внедрении баз данных для решения реальных задач.

Основная идея остается неизменной.