ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет инфокоммуникационных технологий

Дисциплина:
«Проектирование и реализация баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
«СОЗДАНИЕ БД POSTGRESQL В PGADMIN. РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ БД»

Выполнил:
студент группы K32392
Стукалов Артем Сергеевич

(подпись)

Проверил(а):
Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

Цель работы 1.1: овладеть практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4.

Практическое задание 1.1:

1. Установить СУБД PostgreSQL 1X.
2. Создать базу данных с использованием pgadmin 4.

Цель работы 1.2: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Практическое задание 1.2:

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.

Выполнение

Наименование БД: restaurant

ERD диаграмма:

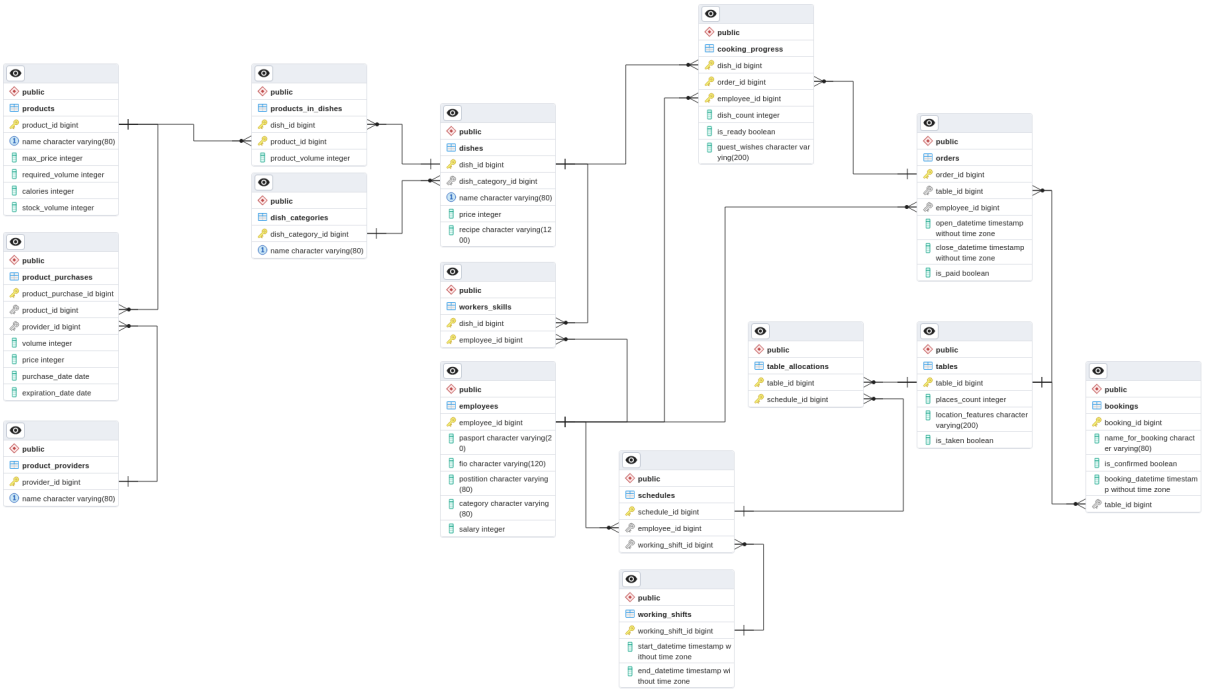

Рисунок 1 - ERD диаграмма

Dump БД: dump базы данных в двух вариантах, сгенерированный в pgAdmin, приложен к отчету. Файлы: restaurant_custom_dump.sql, restaurant_plain_dump.sql

Скрипты работы с БД для создания таблиц:

```
CREATE TABLE employees (
    employee_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    pasport VARCHAR(20) NOT NULL,
    fio VARCHAR(120) NOT NULL,
    postition VARCHAR(80) NOT NULL,
    category VARCHAR(80) NOT NULL CHECK (
        category in ('Кухня', 'Обслуживание', 'Администрация')
    ),
    salary INT NOT NULL CHECK (salary >= 16242)
```

```
);

CREATE TABLE dish_categories(
    dish_category_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    name VARCHAR(80) UNIQUE NOT NULL
);

CREATE TABLE dishes(
    dish_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    dish_category_id BIGINT,
    name VARCHAR(80) UNIQUE NOT NULL,
    price INT NOT NULL CHECK (price > 0),
    recipe VARCHAR(1200) NOT NULL,
    FOREIGN KEY(dish_category_id) REFERENCES dish_categories(dish_category_id) ON
DELETE
    SET
        NULL
);

CREATE TABLE products(
    product_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    name VARCHAR(80) UNIQUE NOT NULL,
    max_price INT NOT NULL CHECK (max_price > 0),
    required_volume INT NOT NULL CHECK (required_volume >= 0),
    calories INT NOT NULL CHECK (calories > 0),
    stock_volume INT NOT NULL CHECK (stock_volume >= 0)
);
```

```
CREATE TABLE product_providers(
    provider_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    name VARCHAR(80) UNIQUE NOT NULL
);

CREATE TABLE product_purchases(
    product_purchase_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    product_id BIGINT,
    provider_id BIGINT,
    volume INT NOT NULL CHECK (volume > 0),
    price INT NOT NULL CHECK (price > 0),
    purchase_date DATE NOT NULL,
    expiration_date DATE NOT NULL CHECK (expiration_date > CURRENT_DATE),
    FOREIGN KEY(product_id) REFERENCES products(product_id) ON DELETE
    SET
        NULL,
        FOREIGN KEY(provider_id) REFERENCES product_providers(provider_id) ON
DELETE
    SET
        NULL,
        CHECK (expiration_date > purchase_date)
);

CREATE TABLE products_in_dishes(
    dish_id BIGINT NOT NULL,
    product_id BIGINT NOT NULL,
```

```
    product_volume INT NOT NULL CHECK (product_volume > 0),
    PRIMARY KEY (dish_id, product_id),
    FOREIGN KEY(dish_id) REFERENCES dishes(dish_id) ON DELETE CASCADE,
    FOREIGN KEY(product_id) REFERENCES products(product_id) ON DELETE
CASCADE
);

CREATE TABLE workers_skills(
    dish_id BIGINT NOT NULL,
    employee_id BIGINT NOT NULL,
    PRIMARY KEY (dish_id, employee_id),
    FOREIGN KEY(dish_id) REFERENCES dishes(dish_id) ON DELETE CASCADE,
    FOREIGN KEY(employee_id) REFERENCES employees(employee_id) ON DELETE
CASCADE
);

CREATE TABLE working_shifts(
    working_shift_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    start_datetime TIMESTAMP NOT NULL,
    end_datetime TIMESTAMP NOT NULL,
    CHECK (start_datetime < end_datetime)
);

CREATE TABLE schedules(
    schedule_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    employee_id BIGINT NOT NULL,
    working_shift_id BIGINT NOT NULL,
    UNIQUE (employee_id, working_shift_id),
```

```
    FOREIGN KEY(employee_id) REFERENCES employees(employee_id) ON DELETE
CASCADE,

    FOREIGN KEY(working_shift_id) REFERENCES working_shifts(working_shift_id) ON
DELETE CASCADE

);


CREATE TABLE tables(

    table_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    places_count INT NOT NULL CHECK (places_count > 0),

    location_features VARCHAR(200),

    is_taken BOOLEAN NOT NULL

);


CREATE TABLE orders(

    order_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    table_id BIGINT,

    employee_id BIGINT,

    open_datetime TIMESTAMP NOT NULL CHECK (open_datetime >=
CURRENT_TIMESTAMP),

    close_datetime TIMESTAMP CHECK (close_datetime >= CURRENT_TIMESTAMP),

    is_paid BOOLEAN NOT NULL,

    FOREIGN KEY(employee_id) REFERENCES employees(employee_id) ON DELETE

    SET

        NULL,

        FOREIGN KEY(table_id) REFERENCES tables(table_id) ON DELETE

    SET

        NULL,

        CHECK (close_datetime > open_datetime)
```

```
);


CREATE TABLE table_allocations(
    table_id BIGINT NOT NULL,
    schedule_id BIGINT NOT NULL,
    PRIMARY KEY (table_id, schedule_id),
    FOREIGN KEY (table_id) REFERENCES tables(table_id) ON DELETE CASCADE,
    FOREIGN KEY (schedule_id) REFERENCES schedules(schedule_id) ON DELETE
CASCADE
);


CREATE TABLE cooking_progress(
    dish_id BIGINT NOT NULL,
    order_id BIGINT NOT NULL,
    employee_id BIGINT NOT NULL,
    dish_count INT NOT NULL CHECK (dish_count > 0),
    is_ready BOOLEAN NOT NULL,
    guest_wishes VARCHAR(200),
    PRIMARY KEY (dish_id, order_id, employee_id),
    FOREIGN KEY (dish_id) REFERENCES dishes(dish_id) ON DELETE CASCADE,
    FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE,
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id) ON DELETE
CASCADE
);


CREATE TABLE bookings(
    booking_id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    name_for_booking VARCHAR(80),
```

```
    is_confirmed BOOLEAN NOT NULL,
    booking_datetime TIMESTAMP NOT NULL CHECK (booking_datetime >=
CURRENT_TIMESTAMP),
    table_id BIGINT NOT NULL,
    FOREIGN KEY (table_id) REFERENCES tables(table_id) ON DELETE CASCADE
);
```

Скрипты работы с БД для вставки данных:

```
INSERT INTO
    employees (pasport, fio, postition, category, salary)
VALUES
    (
        '1234 123456',
        'Стукалов Артем Сергеевич',
        'Шеф-повар',
        'Кухня',
        100000
    ),
    (
        '1234 123457',
        'Денис Аксенов Иванович',
        'Су-шеф',
        'Кухня',
        70000
    ),
    (
        '1234 123458',
        'Егор Лавров Ефимович',
```

```
        'Повар',
        'Кухня',
        50000
    ),
    (
        '1234 123459',
        'Кристина Гончарова Владимировна',
        'Управляющий',
        'Администрация',
        100000
    ),
    (
        '1234 123460',
        'Влад Анисимов Юрьевич',
        'Официант',
        'Обслуживание',
        50000
    ),
    (
        '1234 123461',
        'Петр Анисимов Юрьевич',
        'Официант',
        'Обслуживание',
        50000
    );

INSERT INTO
```

```
    tables (places_count, location_features, is_taken)
VALUES
    (2, 'Столик у окна', FALSE),
    (2, '', FALSE),
    (4, '', FALSE),
    (4, '', FALSE),
    (4, '', FALSE);

INSERT INTO
    working_shifts (start_datetime, end_datetime)
VALUES
    ('2023-03-13 09:00:00', '2023-03-13 18:00:00');

INSERT INTO
    schedules (employee_id, working_shift_id)
VALUES
    (1, 1),
    (2, 1),
    (3, 1),
    (4, 1),
    (5, 1),
    (6, 1);

INSERT INTO
    table_allocations (table_id, schedule_id)
VALUES
    (1, 5),
```

```
    (2, 5),
    (3, 5),
    (4, 6),
    (5, 6);

INSERT INTO
    products (
            name,
            max_price,
            required_volume,
            calories,
            stock_volume
    )
VALUES
    ('Макароны спагетти 3мм', 100, 5000, 23, 5300),
    ('Сосиски вязанка молочные', 150, 3000, 56, 3478),
    ('Кетчуп махеев', 25, 2000, 13, 2167);

INSERT INTO
    product_providers (name)
VALUES
    ('ООО Продукты для ресторанов');

INSERT INTO
    product_purchases (
            product_id,
            provider_id,
```

```
        volume,
        price,
        purchase_date,
        expiration_date
    )
VALUES
    (
        1,
        1,
        7000,
        3200,
        '2023-03-10 06:30:00',
        '2023-06-10 06:30:00'
    ),
    (
        2,
        1,
        8000,
        5400,
        '2023-03-10 06:30:00',
        '2023-03-27 06:30:00'
    ),
    (
        3,
        1,
        3000,
        1607,
```

```
        '2023-03-10 06:30:00',
        '2023-03-25 06:30:00'
    );

INSERT INTO
    dish_categories (name)
VALUES
    ('Домашние');

INSERT INTO
    dishes (dish_category_id, name, price, recipe)
VALUES
    (
        1,
        'Спагетти с сосисками и кетчупом',
        356,
        'Варим спагетти 5 минут...'
    );

INSERT INTO
    products_in_dishes (dish_id, product_id, product_volume)
VALUES
    (1, 1, 200),
    (1, 2, 150),
    (1, 3, 50);

INSERT INTO
```

```
    workers_skills (employee_id, dish_id)
VALUES
    (1, 1),
    (2, 1),
    (3, 1);


INSERT INTO
    orders (
        table_id,
        employee_id,
        open_datetime,
        close_datetime,
        is_paid
    )
VALUES
    (1, 5, '2023-03-13 13:27:00', NULL, FALSE);


INSERT INTO
    cooking_progress (
        dish_id,
        order_id,
        employee_id,
        dish_count,
        is_ready,
        guest_wishes
    )
VALUES
```

```
    (1, 1, 1, 2, FALSE, 'Одно блюдо без кетчупа');


INSERT INTO
    bookings (
        name_for_booking,
        is_confirmed,
        booking_datetime,
        table_id
    )
VALUES
    ('Андрей', FALSE, '2023-03-16 13:27:00', 1);
```

Выводы

В процессе выполнения лабораторной работы удалось более детально ознакомится с работой в pgAdmin 4, получить практические навыки создания таблиц, установки ограничений на таблицы, создания и восстановления резервных копий баз данных.