Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

# Лабораторная работа№ 1.2

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

Выполнил: Галиновский Роман Андреевич

Группа: K3240 Преподаватель: Говорова Марина Михайловна

Санкт-Петербург 2022

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

**Оборудование**: компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL 1Х, 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 .*

1. Восстановить БД.

## ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ:

### Название БД

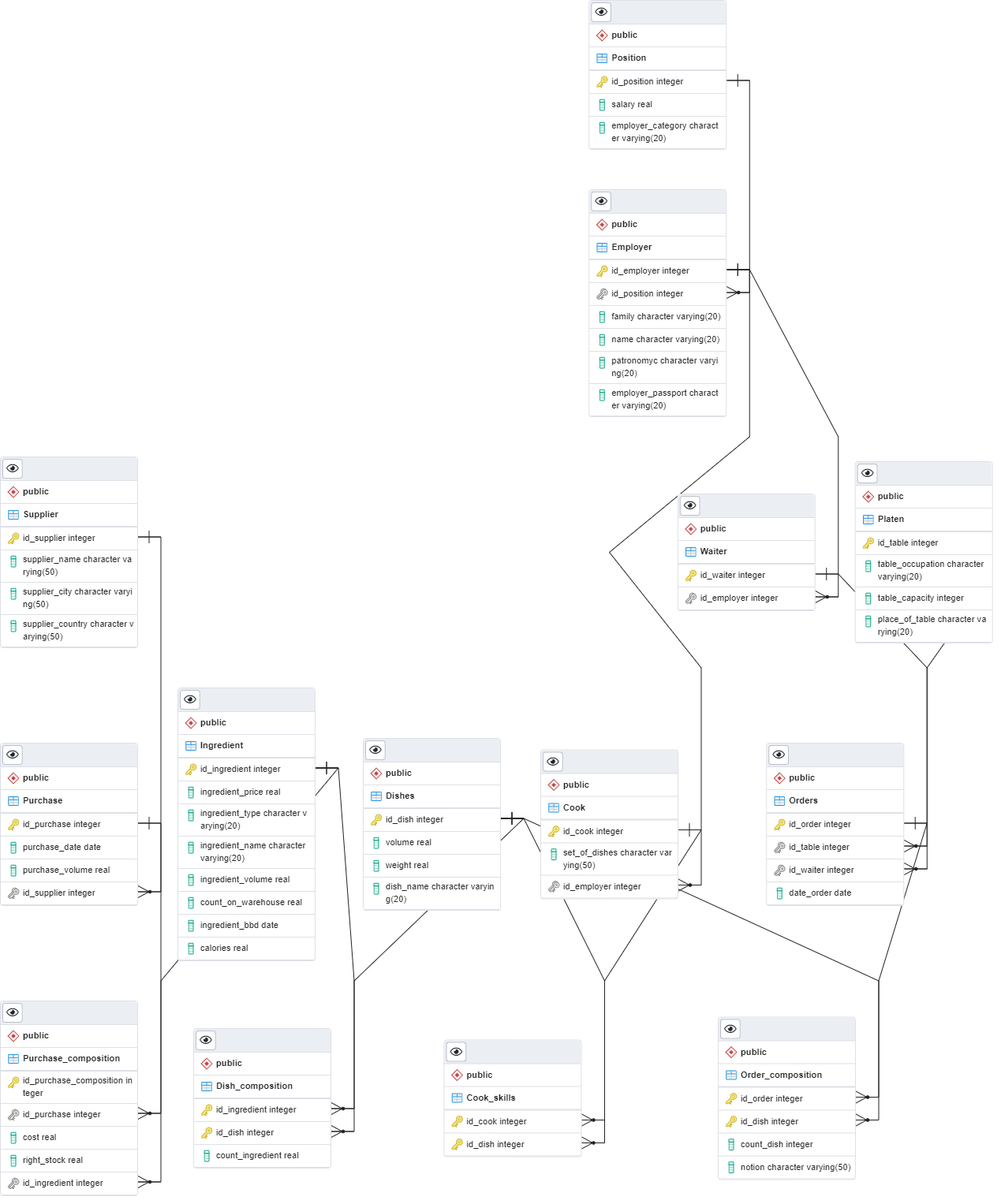
**Вариант 13. «Ресторан»**

Описание предметной области: Сотрудники ресторана – повара и официанты. За каждым официантом закреплены определенные столы. Каждый повар готовит определенный набор блюд. Запас продуктов на складе не должен быть ниже заданного значения. Цена заказа складывается из стоимости ингредиентов и наценки, которая составляет 40% стоимости ингредиентов. БД должна содержать следующий минимальный набор сведений: ФИО сотрудника. Паспортные данные сотрудника. Категория сотрудника. Должность сотрудника. Оклад сотрудника. Наименование ингредиента. Код ингредиента. Дата закупки. Объем закупки. Количество продукта на складе. Необходимый запас продукта. Срок годности. Цена ингредиента. Поставщик. Наименование блюда. Код блюда. Объем ингредиента. Номер стола. Дата заказа. Код заказа. Количество. Название блюда. Ингредиенты, входящие в блюдо. Тип ингредиента.

### Состав реквизитов сущностей:

* + 1. **Ингредиент (**ID ингредиента, цена ингредиента, тип ингредиента, название ингредиента, объём ингредиента, количество на складе, срок годности, калорийность)
    2. **Поставщик** (ID поставщика, имя поставщика, город поставщика, страна поставщика)
    3. **Стол** (ID стола, размещение стола, занятость стола, вместимость стола)
    4. **Должность** (ID должности, оклад, категория сотрудника)
    5. **Закупки** (ID закупки, ID поставщика, дата закупки, объём закупки)
    6. **Состав закупки** (ID состава закупки, ID закупки, ID ингредиента, стоймость, нужный запас)
    7. **Сотрудники** (ID сотрудника, ID должности, Фамилия, имя, отчество, паспорт сотрудника)
    8. **Официант** (ID Официанта, ID сотрудника)
    9. **Заказы** (ID заказа, ID стола, ID Официанта, Дата заказа)
    10. **Состав Заказа(**ID заказа, ID блюда, Количество блюда, примечание)
    11. **Повара** (ID Повара, ID сотрудника, набор блюд)
    12. **Умение повара**(ID Повара, ID Блюда)
    13. **Блюдо** (ID блюда, объём, вес, название блюда)
    14. **Состав Блюда** (ID ингредиента, ID блюда, количество ингредиентов)

## Схема логической модели БД в нотации IDEF1X:



**Заполнение таблиц рабочими данными** Для заполнения использовался скрипт: *INSERT INTO (модификатор доступа).”Имя таблицы”(столбцы) VALUES (значения)*

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

**ДАМП СО СКРИПТАМИ:**

Создаем базу данных:

CREATE DATABASE courses WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE = 'Russian\_Russia.1251';

ALTER DATABASE VAR13 OWNER TO postgres;

\connect VAR13

SET statement\_timeout = 0; SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0; SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false); SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning; SET row\_security = off;

--

-- Name: courses; Type: SCHEMA; Schema: -; Owner: postgres

Создаем схему:

CREATE SCHEMA VAR13;

ALTER SCHEMA VAR13 OWNER TO postgres; SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

Создаем таблицы:

CREATE TABLE public."Cook" (

id\_cook integer NOT NULL,

set\_of\_dishes character varying(50) NOT NULL,

id\_employer integer NOT NULL

);

ALTER TABLE public."Cook" OWNER TO postgres;

--

-- TOC entry 226 (class 1259 OID 57443)

-- Name: Cook\_skills; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Cook\_skills" (

id\_cook integer NOT NULL,

id\_dish integer NOT NULL

);

ALTER TABLE public."Cook\_skills" OWNER TO postgres;

--

-- TOC entry 211 (class 1259 OID 49176)

-- Name: Dish\_composition; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Dish\_composition" (

id\_ingredient integer NOT NULL,

id\_dish integer NOT NULL,

count\_ingredient real NOT NULL

);

ALTER TABLE public."Dish\_composition" OWNER TO postgres;

--

-- TOC entry 212 (class 1259 OID 49179)

-- Name: Dishes; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Dishes" (

id\_dish integer NOT NULL,

id\_order integer NOT NULL,

volume real NOT NULL,

weight real NOT NULL,

dish\_name character varying(20)

);

ALTER TABLE public."Dishes" OWNER TO postgres;

--

-- TOC entry 213 (class 1259 OID 49182)

-- Name: Employer; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Employer" (

id\_employer integer NOT NULL,

id\_position integer NOT NULL,

family character varying(20) NOT NULL,

name character varying(20) NOT NULL,

patronomyc character varying(20) NOT NULL,

employer\_passport character varying(20) NOT NULL

);

ALTER TABLE public."Employer" OWNER TO postgres;

--

-- TOC entry 214 (class 1259 OID 49185)

-- Name: Ingredient; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Ingredient" (

id\_ingredient integer NOT NULL,

ingredient\_price real NOT NULL,

ingredient\_type character varying(20) NOT NULL,

ingredient\_name character varying(20) NOT NULL,

ingredient\_volume real NOT NULL,

count\_on\_warehouse real NOT NULL,

ingredient\_bbd date NOT NULL,

calories real NOT NULL

);

ALTER TABLE public."Ingredient" OWNER TO postgres;

--

-- TOC entry 225 (class 1259 OID 57428)

-- Name: Order\_composition; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Order\_composition" (

id\_order integer NOT NULL,

id\_dish integer NOT NULL,

count\_dish integer NOT NULL,

notion character varying(50)

);

ALTER TABLE public."Order\_composition" OWNER TO postgres;

--

-- TOC entry 215 (class 1259 OID 49188)

-- Name: Orders; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Orders" (

id\_order integer NOT NULL,

id\_table integer NOT NULL,

id\_waiter integer NOT NULL,

date\_order date NOT NULL

);

ALTER TABLE public."Orders" OWNER TO postgres;

--

-- TOC entry 216 (class 1259 OID 49191)

-- Name: Platen; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Platen" (

id\_table integer NOT NULL,

table\_occupation character varying(20),

table\_capacity integer NOT NULL,

place\_of\_table character varying(20)

);

ALTER TABLE public."Platen" OWNER TO postgres;

--

-- TOC entry 217 (class 1259 OID 49194)

-- Name: Position; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Position" (

id\_position integer NOT NULL,

salary real NOT NULL,

employer\_category character varying(20) NOT NULL

);

ALTER TABLE public."Position" OWNER TO postgres;

--

-- TOC entry 218 (class 1259 OID 49197)

-- Name: Purchase; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Purchase" (

id\_purchase integer NOT NULL,

purchase\_date date,

purchase\_volume real NOT NULL,

id\_supplier integer NOT NULL

);

ALTER TABLE public."Purchase" OWNER TO postgres;

--

-- TOC entry 219 (class 1259 OID 49200)

-- Name: Purchase\_composition; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Purchase\_composition" (

id\_purchase\_composition integer NOT NULL,

id\_purchase integer NOT NULL,

cost real NOT NULL,

right\_stock real NOT NULL,

id\_ingredient integer NOT NULL

);

ALTER TABLE public."Purchase\_composition" OWNER TO postgres;

--

-- TOC entry 220 (class 1259 OID 49203)

-- Name: Supplier; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Supplier" (

id\_supplier integer NOT NULL,

supplier\_name character varying(50),

supplier\_city character varying(50),

supplier\_country character varying(50)

);

ALTER TABLE public."Supplier" OWNER TO postgres;

--

-- TOC entry 221 (class 1259 OID 49206)

-- Name: Supplier\_id\_supplier\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public."Supplier" ALTER COLUMN id\_supplier ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public."Supplier\_id\_supplier\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 222 (class 1259 OID 49207)

-- Name: Table\_id\_table\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public."Platen" ALTER COLUMN id\_table ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public."Table\_id\_table\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 223 (class 1259 OID 49208)

-- Name: Waiter; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public."Waiter" (

id\_waiter integer NOT NULL,

id\_employer integer NOT NULL

);

ALTER TABLE public."Waiter" OWNER TO postgres;

--

-- TOC entry 224 (class 1259 OID 49211)

-- Name: Waiter\_id\_waiter\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

ALTER TABLE public."Waiter" ALTER COLUMN id\_waiter ADD GENERATED ALWAYS AS IDENTITY (

SEQUENCE NAME public."Waiter\_id\_waiter\_seq"

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1

);

--

-- TOC entry 3451 (class 0 OID 49173)

-- Dependencies: 210

-- Data for Name: Cook; Type: TABLE DATA; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Cook\_id\_cook\_fkey" ON public."Cook\_skills" USING btree (id\_cook);

--

-- TOC entry 3244 (class 1259 OID 49277)

-- Name: fki\_Dishes\_id\_dish\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Dishes\_id\_dish\_fkey" ON public."Dish\_composition" USING btree (id\_dish);

--

-- TOC entry 3241 (class 1259 OID 49278)

-- Name: fki\_Employer\_id\_employer\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Employer\_id\_employer\_fkey" ON public."Cook" USING btree (id\_employer);

--

-- TOC entry 3282 (class 1259 OID 49279)

-- Name: fki\_Ingredient\_id\_ingredient\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Ingredient\_id\_ingredient\_fkey" ON public."Purchase\_composition" USING btree (id\_ingredient);

--

-- TOC entry 3249 (class 1259 OID 49280)

-- Name: fki\_Orders\_id\_order\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Orders\_id\_order\_fkey" ON public."Dishes" USING btree (id\_order);

--

-- TOC entry 3254 (class 1259 OID 49281)

-- Name: fki\_Position\_id\_position\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Position\_id\_position\_fkey" ON public."Employer" USING btree (id\_position);

--

-- TOC entry 3283 (class 1259 OID 49282)

-- Name: fki\_Purchase\_id\_purchase\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Purchase\_id\_purchase\_fkey" ON public."Purchase\_composition" USING btree (id\_purchase);

--

-- TOC entry 3277 (class 1259 OID 49283)

-- Name: fki\_Purchase\_id\_supplier\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Purchase\_id\_supplier\_fkey" ON public."Purchase" USING btree (id\_purchase);

--

-- TOC entry 3263 (class 1259 OID 49284)

-- Name: fki\_Table\_id\_table\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Table\_id\_table\_fkey" ON public."Orders" USING btree (id\_table);

--

-- TOC entry 3264 (class 1259 OID 49285)

-- Name: fki\_Waiter\_id\_waiter\_fkey; Type: INDEX; Schema: public; Owner: postgres

--

CREATE INDEX "fki\_Waiter\_id\_waiter\_fkey" ON public."Orders" USING btree (id\_waiter);

--

-- TOC entry 3310 (class 2606 OID 57448)

-- Name: Cook\_skills Cook\_id\_cook\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Cook\_skills"

ADD CONSTRAINT "Cook\_id\_cook\_fkey" FOREIGN KEY (id\_cook) REFERENCES public."Cook"(id\_cook) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3298 (class 2606 OID 49291)

-- Name: Dish\_composition Dishes\_id\_dish\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Dish\_composition"

ADD CONSTRAINT "Dishes\_id\_dish\_fkey" FOREIGN KEY (id\_dish) REFERENCES public."Dishes"(id\_dish) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3309 (class 2606 OID 57438)

-- Name: Order\_composition Dishes\_id\_dish\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Order\_composition"

ADD CONSTRAINT "Dishes\_id\_dish\_fkey" FOREIGN KEY (id\_dish) REFERENCES public."Dishes"(id\_dish) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3311 (class 2606 OID 57454)

-- Name: Cook\_skills Dishes\_id\_dish\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Cook\_skills"

ADD CONSTRAINT "Dishes\_id\_dish\_fkey" FOREIGN KEY (id\_dish) REFERENCES public."Dishes"(id\_dish) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3297 (class 2606 OID 49296)

-- Name: Cook Employer\_id\_employer\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Cook"

ADD CONSTRAINT "Employer\_id\_employer\_fkey" FOREIGN KEY (id\_employer) REFERENCES public."Employer"(id\_employer) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3307 (class 2606 OID 49301)

-- Name: Waiter Employer\_id\_employer\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Waiter"

ADD CONSTRAINT "Employer\_id\_employer\_fkey" FOREIGN KEY (id\_employer) REFERENCES public."Employer"(id\_employer) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3299 (class 2606 OID 49306)

-- Name: Dish\_composition Ingredient\_id\_ingredient\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Dish\_composition"

ADD CONSTRAINT "Ingredient\_id\_ingredient\_fkey" FOREIGN KEY (id\_ingredient) REFERENCES public."Ingredient"(id\_ingredient) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3305 (class 2606 OID 49311)

-- Name: Purchase\_composition Ingredient\_id\_ingredient\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Purchase\_composition"

ADD CONSTRAINT "Ingredient\_id\_ingredient\_fkey" FOREIGN KEY (id\_ingredient) REFERENCES public."Ingredient"(id\_ingredient) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3300 (class 2606 OID 49316)

-- Name: Dishes Orders\_id\_order\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Dishes"

ADD CONSTRAINT "Orders\_id\_order\_fkey" FOREIGN KEY (id\_order) REFERENCES public."Orders"(id\_order) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3308 (class 2606 OID 57433)

-- Name: Order\_composition Orders\_id\_order\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Order\_composition"

ADD CONSTRAINT "Orders\_id\_order\_fkey" FOREIGN KEY (id\_order) REFERENCES public."Orders"(id\_order) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3302 (class 2606 OID 49321)

-- Name: Orders Platen\_id\_table\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Platen\_id\_table\_fkey" FOREIGN KEY (id\_table) REFERENCES public."Platen"(id\_table) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3301 (class 2606 OID 49326)

-- Name: Employer Position\_id\_position\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Employer"

ADD CONSTRAINT "Position\_id\_position\_fkey" FOREIGN KEY (id\_position) REFERENCES public."Position"(id\_position) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3306 (class 2606 OID 49331)

-- Name: Purchase\_composition Purchase\_id\_purchase\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Purchase\_composition"

ADD CONSTRAINT "Purchase\_id\_purchase\_fkey" FOREIGN KEY (id\_purchase) REFERENCES public."Purchase"(id\_purchase) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3304 (class 2606 OID 49520)

-- Name: Purchase Purchase\_id\_supplier\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Purchase"

ADD CONSTRAINT "Purchase\_id\_supplier\_fkey" FOREIGN KEY (id\_supplier) REFERENCES public."Supplier"(id\_supplier) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

--

-- TOC entry 3303 (class 2606 OID 49341)

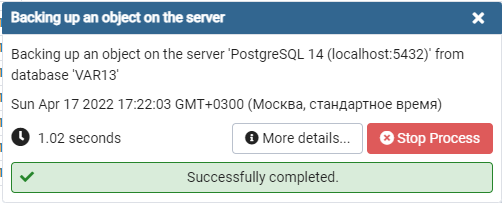
-- Name: Orders Waiter\_id\_waiter\_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Orders"

ADD CONSTRAINT "Waiter\_id\_waiter\_fkey" FOREIGN KEY (id\_waiter) REFERENCES public."Waiter"(id\_waiter) ON UPDATE RESTRICT ON DELETE RESTRICT DEFERRABLE INITIALLY DEFERRED;

Созданием бэкапа и его восстановление:



Изображение выглядит как текст

Автоматически созданное описание

# Вывод:

PgAdmin – достаточно удобная программа для создания баз данных PostgreSQL, обладающая приемлемо интуитивным интерфейсом, разобраться с которым новичку не доставит великих проблем. Но, к сожалению, программа обладает неявными ограничениями или даже багами, с которыми новичку самостоятельно справиться будет гораздо тяжелее. К примеру, с чем столкнулся Я: невозможность задать ограничение для столбца, если его имя содержит символы верхнего регистра, необходимость использовать скрипты SELECT, INSERT, DELETE, etc., так как программа не воспринимает стандартный метод ввода SQL.