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: К3240
Преподаватель: Говорова Марина Михайловна

Санкт-Петербург
2022

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.

ТЕХНОЛОГИЯ ВЫПОЛНЕНИЯ РАБОТЫ:

1. 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:

a) **Ингредиент (**ID ингредиента, цена ингредиента, тип ингредиента, название ингредиента, объём ингредиента, количество на складе, срок годности, калорийность)

b) **Поставщик** (ID поставщика, имя поставщика, город поставщика, страна поставщика)

c) **Стол** (ID стола, размещение стола, занятость стола, вместимость_стола)

d) **Должность** (ID должности, оклад, категория сотрудника)

e) **Закупки** (ID закупки, ID поставщика, дата закупки, объём закупки)

f) **Состав закупки** (ID состава закупки, ID закупки, ID ингредиента, стоймость, нужный запас)

g) **Сотрудники** (ID сотрудника, ID должности, Фамилия, имя, отчество, паспорт сотрудника)

h) **Официант** (ID Официанта, ID сотрудника)

i) **Заказы** (ID заказа, ID стола, ID Официанта, Дата заказа)

j) **Состав Заказа(**ID заказа, ID блюда, Количество блюда, примечание)

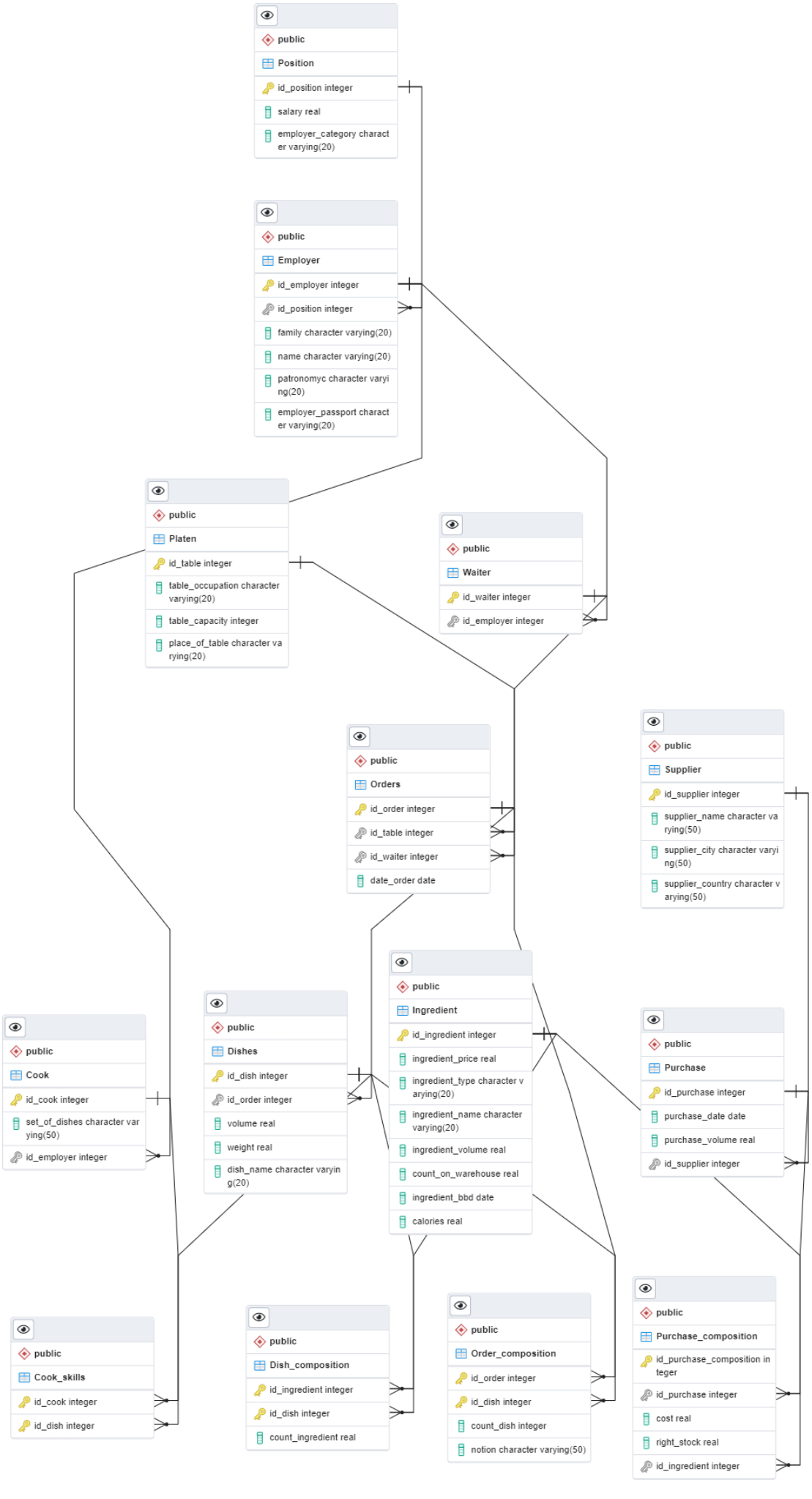
k) **Повара** (ID Повара, ID сотрудника, набор блюд)

l) **Умение повара**(ID Повара, ID Блюда)

m) **Блюдо** (ID блюда, ID заказа, объём, вес, название блюда)

n) **Состав Блюда** (ID ингредиента, ID блюда, количество ингредиентов)

2. Схема логической модели БД в нотации IDEF1X:


public
Position
id_position integer
salary real
employer_category character varying(20)
public
Employer
id_employer integer
id_position integer
family character varying(20)
name character varying(20)
patronomyc character varying(20)
employer_passport character varying(20)
public
Platen
id_table integer
table_occupation character varying(20)
table_capacity integer
place_of_table character varying(20)
public
Waiter
id_waiter integer
id_employer integer
public
Orders
id_order integer
id_table integer
id_waiter integer
date_order date
public
Supplier
id_supplier integer
supplier_name character varying(50)
supplier_city character varying(50)
supplier_country character varying(50)
public
Cook
id_cook integer
set_of_dishes character varying(50)
id_employer integer
public
Dishes
id_dish integer
id_order integer
volume real
weight real
dish_name character varying(20)
public
Ingredient
id_ingredient integer
ingredient_price real
ingredient_type character varying(20)
ingredient_name character varying(20)
ingredient_volume real
count_on_warehouse real
ingredient_bbd date
calories real
public
Purchase
id_purchase integer
purchase_date date
purchase_volume real
id_supplier integer
public
Cook_skills
id_cook integer
id_dish integer
public
Dish_composition
id_ingredient integer
id_dish integer
count_ingredient real
public
Order_composition
id_order integer
id_dish integer
count_dish integer
notion character varying(50)
public
Purchase_composition
id_purchase_composition integer
id_purchase integer
cost real
right_stock real
id_ingredient integer

Заполнение таблиц рабочими данными

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

	id_order [PK] integer	id_table integer	id_waiter integer	date_order date	notion character varying (50)	count_order integer
1	111161	111151	111131	2022-05-31	no	20
2	111162	111151	111133	2022-06-01	no salt	17
3	111163	111153	111132	2022-06-02	no chilly	4
4	111164	111155	111131	2022-06-03	no sugar	5
5	111165	111157	111132	2022-06-04	no	8
6	111166	111158	111133	2022-06-05	no	10
7	111167	111158	111132	2022-06-07	no	11
8	111168	111159	111133	2022-06-08	no	20

```
INSERT INTO public."Orders"(
    id_order, id_table, id_waiter, date_order, notion, count_order)
    VALUES
    (111161, 111151, 111131, '2022-05-31' :: date, 'no', 20),
    (111162, 111151, 111133, '2022-06-01' :: date, 'no salt', 17),
    (111163, 111153, 111132, '2022-06-02' :: date, 'no chilly', 4),
    (111164, 111155, 111131, '2022-06-03' :: date, 'no sugar', 5),
    (111165, 111157, 111132, '2022-06-04' :: date, 'no', 8),
    (111166, 111158, 111133, '2022-06-05' :: date, 'no', 10),
    (111167, 111158, 111132, '2022-06-07' :: date, 'no', 11),
    (111168, 111159, 111133, '2022-06-08' :: date, 'no', 20),
    (111169, 111153, 111131, '2022-06-09' :: date, 'no', 4),
```

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)

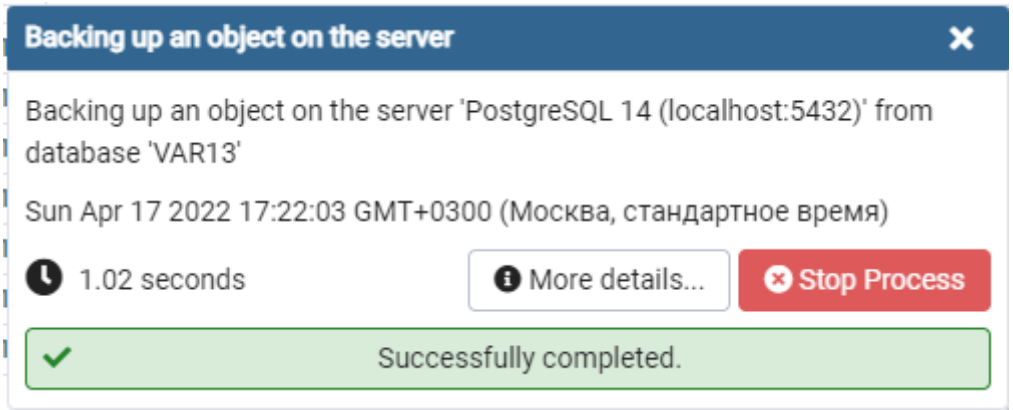
-- Name: Orders Waiter_id_waiter_fkey; Type: FK CONSTRAINT; Schema: public; Owner: postgres

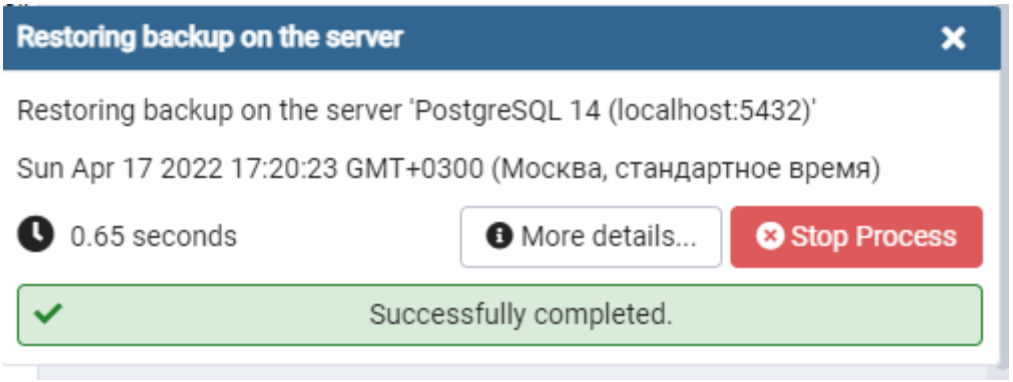
--

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:





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.