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

**по дисциплине:** Проектирование и реализация баз данных

Специальность: 09.03.03 Мобильные и сетевые технологии

Проверил: Говорова М.М. \_\_\_\_\_\_\_\_\_\_

Дата: «27» апреля 2021г.   
Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:   
студент группы К3240

Костылев Иван

Санкт-Петербург 2021 г

**Цель работы**

Овладение практическими навыками реализации базы данных с использованием PostgreSQL

**Практическое задание**

1. На основе созданной инфологической модели данных создать базу данных с помощью pgAdmin 4;
2. Создать таблицы базы данных PostgreSQL с соответственными ограничениями полей PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE;
3. Заполнить таблицы данными.

**Ход работы**

Вариант 13. БД «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.

1. **Название создаваемой БД:** Restaurant
2. **Cхема инфологической модели**

Схема инфологической модели данных, которая сгенерирована в pgAdmin, представлена в Приложении 1.

1. **Скрипты работы с базой данных Restaurant**
   1. Создание таблиц данных

Для реализации базы данных требуется создать 14 таблиц. При создании каждой таблицы были введены соответствующие ограничения на строковые типы данных (к примеру, поля с типами продуктов и блюд ограничены лишь определенными значениями – таблицы Products и Dishes). Код создания каждой из таблиц приведен в Приложении 2.

* 1. Коды вставки (INSERT) данных и их выборки (SELECT).

Запросы вставки данных в некоторые таблицы (Dishes, Positions и Workers) приведены в Приложении 3.

**Выводы**

Нами была реализована спроектированная в прошлой лабораторной работе модель данных о ресторане. В результате работы мы выяснили, что pgAdmin является достаточно удобным инструментом для генерации большого количества таблиц. Однако для того, чтобы разобраться в интерфейсе программы, требуется много времени. Выполнение множества однотипных действий при использовании pgAdmin 4 также тормозит процесс реализации.

Одним из самых неудобных пунктов при реализации базы данных является создание ограничений по внешнему ключу. Для создания внешнего ключа таблицы требуется сделать очень большое количество действий.

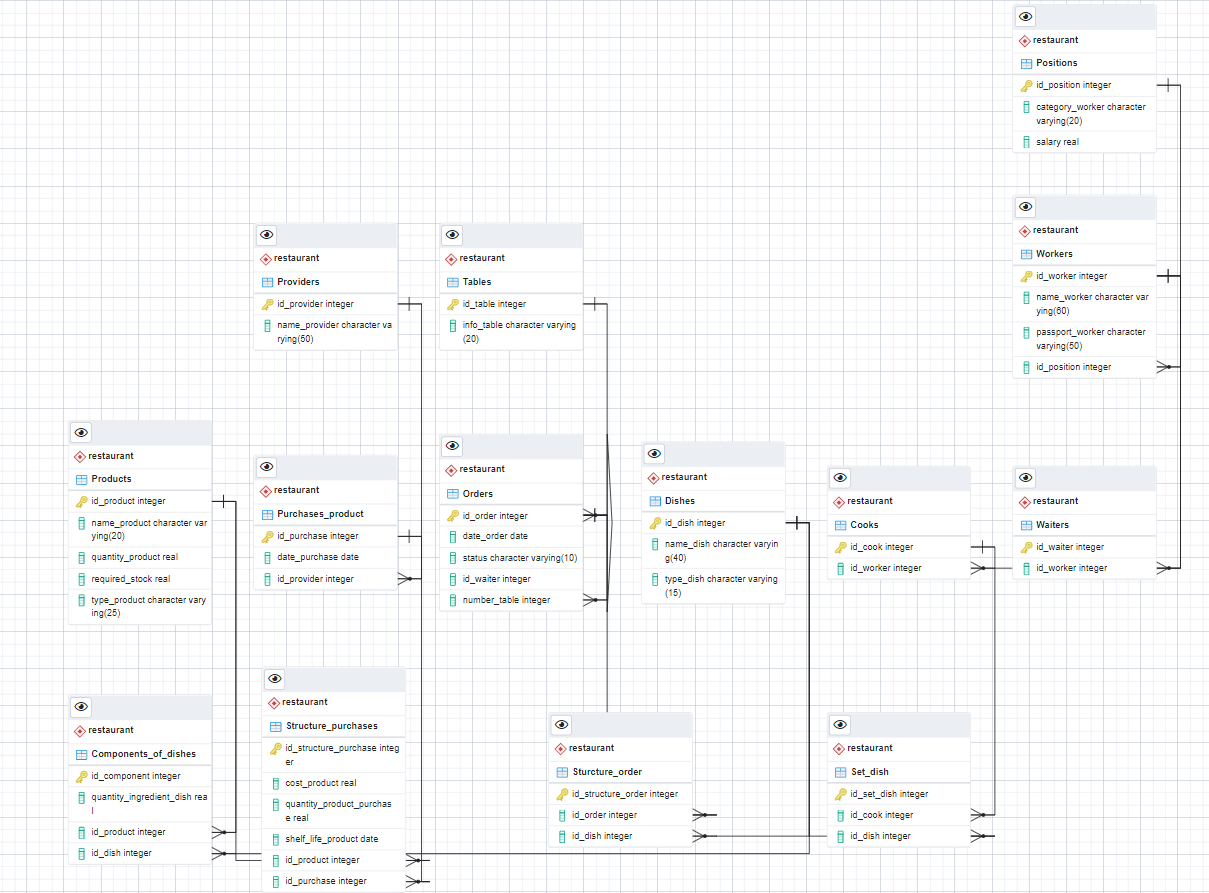
**Приложение 1. Инфологическая модель данных**

Рисунок 1 – Инфологическая модель данных

**Приложение 2. Создание баз данных**

**Фрагмент 1 – Создание таблицы Components\_of\_dishes**

CREATE TABLE restaurant."Components\_of\_dishes"

(

id\_component integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9999999 CACHE 1 ),

quantity\_ingredient\_dish real NOT NULL,

id\_product integer NOT NULL,

id\_dish integer NOT NULL,

CONSTRAINT "Components\_of\_dishes\_pkey" PRIMARY KEY (id\_component),

CONSTRAINT fk\_dish FOREIGN KEY (id\_dish)

REFERENCES restaurant."Dishes" (id\_dish) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT fk\_product FOREIGN KEY (id\_product)

REFERENCES restaurant."Products" (id\_product) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Components\_of\_dishes"

OWNER to postgres;

**Фрагмент 2 – Создание таблицы Cooks**

CREATE TABLE restaurant."Cooks"

(

id\_cook integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9999999 CACHE 1 ),

id\_worker integer NOT NULL,

CONSTRAINT "Cooks\_pkey" PRIMARY KEY (id\_cook),

CONSTRAINT fk\_worker FOREIGN KEY (id\_worker)

REFERENCES restaurant."Workers" (id\_worker) MATCH FULL

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Cooks"

OWNER to postgres;

**Фрагмент 3 – Создание таблицы Dishes**

CREATE TABLE restaurant."Dishes"

(

id\_dish integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

name\_dish character varying(40) COLLATE pg\_catalog."default" NOT NULL,

type\_dish character varying(15) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Dishes\_pkey" PRIMARY KEY (id\_dish),

CONSTRAINT "Dishes\_type\_dish\_check" CHECK (type\_dish::text = ANY (ARRAY['salad'::character varying, 'snack'::character varying, 'first course'::character varying, 'second course'::character varying, 'dessert'::character varying, 'drink'::character varying]::text[]))

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Dishes"

OWNER to postgres;

**Фрагмент 4 – Создание таблицы Orders**

CREATE TABLE restaurant."Orders"

(

id\_order integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

date\_order date NOT NULL,

status character varying(10) COLLATE pg\_catalog."default" NOT NULL,

id\_waiter integer NOT NULL,

number\_table integer NOT NULL,

CONSTRAINT "Orders\_pkey" PRIMARY KEY (id\_order),

CONSTRAINT fk\_table FOREIGN KEY (number\_table)

REFERENCES restaurant."Tables" (id\_table) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT fk\_waiter FOREIGN KEY (id\_order)

REFERENCES restaurant."Orders" (id\_order) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Orders"

OWNER to postgres;

**Фрагмент 5 – Создание таблицы Positions**

CREATE TABLE restaurant."Positions"

(

id\_position integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

category\_worker character varying(20) COLLATE pg\_catalog."default" NOT NULL,

salary real NOT NULL,

CONSTRAINT "Positions\_pkey" PRIMARY KEY (id\_position)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Positions"

OWNER to postgres;

**Фрагмент 6 – Создание таблицы Products**

CREATE TABLE restaurant."Products"

(

id\_product integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

name\_product character varying(20) COLLATE pg\_catalog."default" NOT NULL,

quantity\_product real NOT NULL,

required\_stock real NOT NULL,

type\_product character varying(25) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Products\_pkey" PRIMARY KEY (id\_product),

CONSTRAINT "Products\_type\_product\_check" CHECK (type\_product::text = ANY (ARRAY['vegetables fruits'::character varying::text, 'milk'::character varying::text, 'meet'::character varying::text, 'edible fats'::character varying::text, 'starch'::character varying::text, 'confectionery'::character varying::text, 'grain flour'::character varying::text, 'fish'::character varying::text, 'eggs'::character varying::text, 'gustatory'::character varying::text]))

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Products"

OWNER to postgres;

**Фрагмент 7 – Создание таблицы Providers**

CREATE TABLE restaurant."Providers"

(

id\_provider integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

name\_provider character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Providers\_pkey" PRIMARY KEY (id\_provider)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Providers"

OWNER to postgres;

**Фрагмент 8 – Создание таблицы Purchases\_product**

CREATE TABLE restaurant."Purchases\_product"

(

id\_purchase integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

date\_purchase date NOT NULL,

id\_provider integer NOT NULL,

CONSTRAINT "Purchases\_product\_pkey" PRIMARY KEY (id\_purchase),

CONSTRAINT fk\_provider FOREIGN KEY (id\_provider)

REFERENCES restaurant."Providers" (id\_provider) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Purchases\_product"

OWNER to postgres;

**Фрагмент 9 – Создание таблицы Set\_dish**

CREATE TABLE restaurant."Set\_dish"

(

id\_set\_dish integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

id\_cook integer NOT NULL,

id\_dish integer NOT NULL,

CONSTRAINT "Set\_dish\_pkey" PRIMARY KEY (id\_set\_dish),

CONSTRAINT fk\_cook FOREIGN KEY (id\_cook)

REFERENCES restaurant."Cooks" (id\_cook) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT fk\_dish FOREIGN KEY (id\_dish)

REFERENCES restaurant."Dishes" (id\_dish) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Set\_dish"

OWNER to postgres;

**Фрагмент 10 – Создание таблицы Structure\_purchases**

CREATE TABLE restaurant."Structure\_purchases"

(

id\_structure\_purchase integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

cost\_product real NOT NULL,

quantity\_product\_purchase real NOT NULL,

shelf\_life\_product date NOT NULL,

id\_product integer NOT NULL,

id\_purchase integer NOT NULL,

CONSTRAINT "Structure\_purchases\_pkey" PRIMARY KEY (id\_structure\_purchase),

CONSTRAINT fk\_product FOREIGN KEY (id\_product)

REFERENCES restaurant."Products" (id\_product) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT fk\_purchase FOREIGN KEY (id\_purchase)

REFERENCES restaurant."Purchases\_product" (id\_purchase) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Structure\_purchases"

OWNER to postgres;

**Фрагмент 11 – Создание таблицы Structure\_order**

CREATE TABLE restaurant."Sturcture\_order"

(

id\_structure\_order integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

id\_order integer NOT NULL,

id\_dish integer NOT NULL,

CONSTRAINT "Sturcture\_order\_pkey" PRIMARY KEY (id\_structure\_order),

CONSTRAINT fk\_dish FOREIGN KEY (id\_dish)

REFERENCES restaurant."Dishes" (id\_dish) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT,

CONSTRAINT fk\_order FOREIGN KEY (id\_order)

REFERENCES restaurant."Orders" (id\_order) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Sturcture\_order"

OWNER to postgres;

**Фрагмент 12 – Создание таблицы Tables**

CREATE TABLE restaurant."Tables"

(

id\_table integer NOT NULL,

info\_table character varying(20) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Tables\_pkey" PRIMARY KEY (id\_table)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Tables"

OWNER to postgres;

**Фрагмент 13 – Создание таблицы Waiters**

CREATE TABLE restaurant."Waiters"

(

id\_waiter integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

id\_worker integer NOT NULL,

CONSTRAINT "Waiters\_pkey" PRIMARY KEY (id\_waiter),

CONSTRAINT fr\_worker\_w FOREIGN KEY (id\_worker)

REFERENCES restaurant."Workers" (id\_worker) MATCH FULL

ON UPDATE RESTRICT

ON DELETE RESTRICT

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Waiters"

OWNER to postgres;

**Фрагмент 14 – Создание таблицы Workers**

CREATE TABLE restaurant."Workers"

(

id\_worker integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

name\_worker character varying(60) COLLATE pg\_catalog."default" NOT NULL,

passport\_worker character varying(50) COLLATE pg\_catalog."default" NOT NULL,

id\_position integer NOT NULL,

CONSTRAINT "Workers\_pkey" PRIMARY KEY (id\_worker),

CONSTRAINT fk\_id\_position FOREIGN KEY (id\_position)

REFERENCES restaurant."Positions" (id\_position) MATCH FULL

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE restaurant."Workers"

OWNER to postgres;

**Приложение 3. Работа с данными**

**Фрагмент 1 – Вставка данных о блюдах**

INSERT INTO restaurant."Dishes"(name\_dish, type\_dish)

VALUES

('apple cake', 'dessert'),

('red soup', 'first course'),

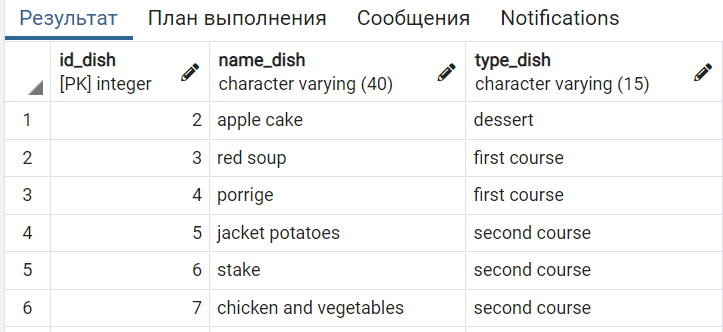
('porrige', 'first course'),

('jacket potatoes', 'second course'),

('stake', 'second course'),

('chicken and vegetables', 'second course');

**Рисунок 1 – Результат запроса SELECT \* FROM restaurant.”Dishes”**



**Фрагмент 2 – Вставка данных о должностях и работниках**

INSERT INTO restaurant."Positions"(category\_worker, salary)

VALUES

('cook', 50000),

('senior cook', 100000),

('chef', 200000),

('manager', 200000),

('waiter', 40000);

INSERT INTO restaurant."Workers"(name\_worker, passport\_worker, id\_position)

VALUES

('Ivanov Ivan Aristarkhovich', '57\*\* \*\*\*\*42', 3),

('Ivanov Sergey Aristarkhovich', '57\*\* \*\*\*\*43', 4),

('Ryazanov Iliya Sergeevich', '21\*\* \*\*\*\*11', 1),

('Uhazhorov Kirill Gennadievich', '47\*\* \*\*\*\*09', 1),

('Bu Andersson', '1234 453\*\*22', 4),

('Zhdanov Sergey Yurievich', '09\*\* \*\*\*\*32', 5),

('Vaseva Regina Andreevna', '47\*\* \*\*\*\*67', 5),

('Name Name Anonnim', '\*\*\*\* \*\*\*\*\*\*', 5);

**Рисунок 2 – Результат запроса SELECT \* FROM restaurant.”Workers”**

