

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. Схема инфологической модели**

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

### **3. Скрипты работы с базой данных Restaurant**

#### **3.1. 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.

### 3.2. 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

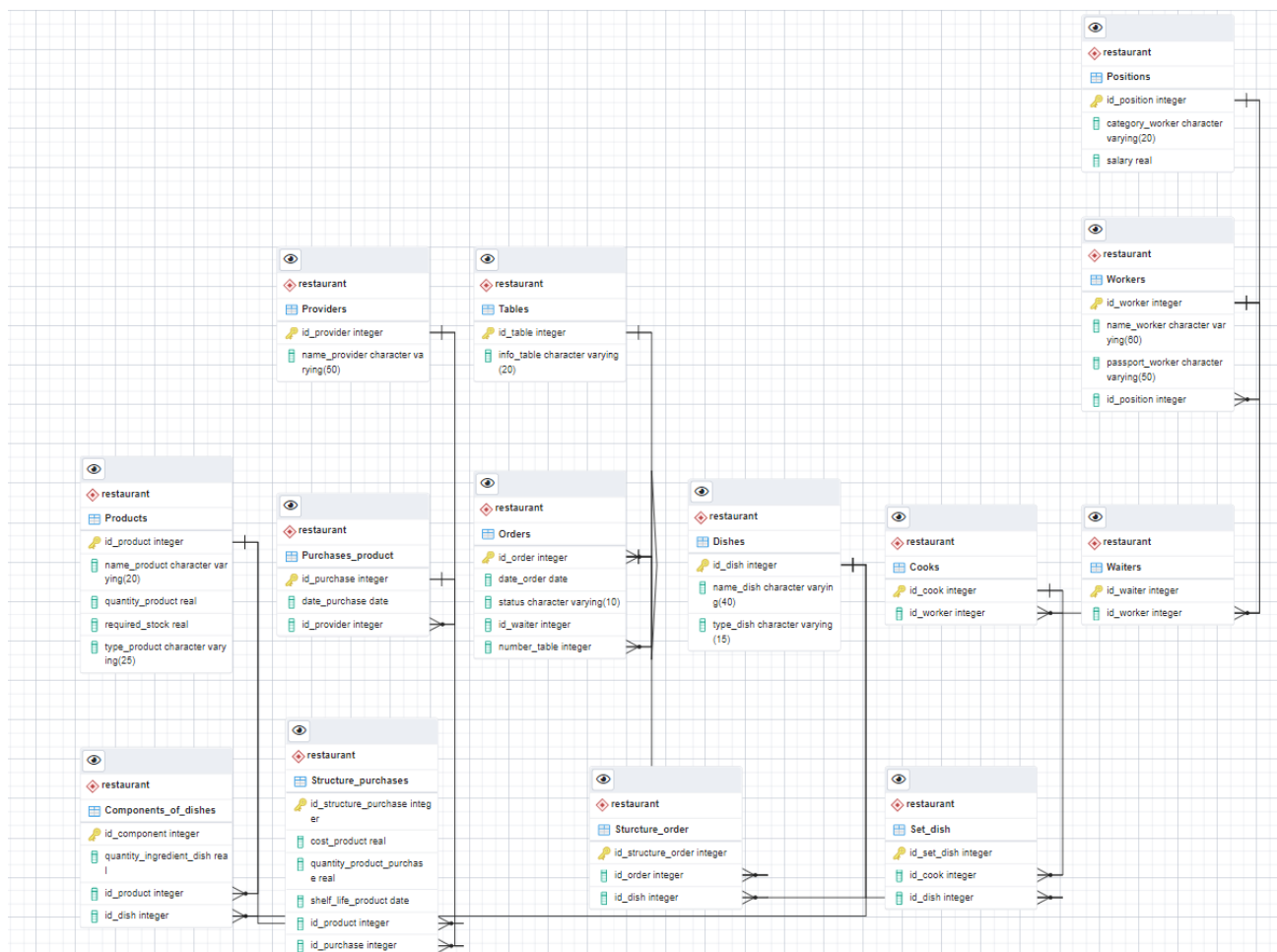


Рисунок 1 – Инфологическая модель данных

## 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 99999999 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 99999999 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'),
('porridge', 'first course'),
('jacket potatoes', 'second course'),
('stake', 'second course'),
('chicken and vegetables', 'second course');
```

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






Результат	План выполнения	Сообщения	Notifications
	id_dish [PK] integer	name_dish character varying (40)	type_dish character varying (15)
1	2	apple cake	dessert
2	3	red soup	first course
3	4	porridge	first course
4	5	jacket potatoes	second course
5	6	stake	second course
6	7	chicken and vegetables	second course

#### 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),
('Ukazhorov 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 Anonim', '**** *', 5);
```

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

Результат    План выполнения    Сообщения    Notifications				
	 id_worker [PK] integer 	name_worker character varying (60) 	passport_worker character varying (50) 	id_position integer 
1	1	Ivanov Ivan Aristarkhovich	57** ****42	3
2	2	Ivanov Sergey Aristarkhovich	57** ****43	4
3	3	Ryazanov Iliya Sergeevich	21** ****11	1
4	4	Uhazhorov Kirill Gennadievich	47** ****09	1
5	5	Bu Andersson	1234 453**22	4
6	6	Zhdanov Sergey Yurievich	09** ****32	5
7	7	Vaseva Regina Andreevna	47** ****67	5
8	8	Name Name Anonnim	**** *****	5