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. Заполнение таблиц рабочими данными

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

|  |  |
| --- | --- |
| Проверил:  Говорова М.М.  Дата: «14» июня 2021 г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы К3243  Плотская Д.А. |

Санкт-Петербург 2020/2021

**Цель работы**

Овладеть практическими навыками создания таблиц базы данных PostgreSQL 1Х, заполнения их рабочими данными, резервного копирования и восстановления БД.

**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.
7. Восстановить БД.

**Выполнение**

**I. Наименование БД**

autorepair\_shop

**II. Схема логической модели БД, сгенерированная в ERD**

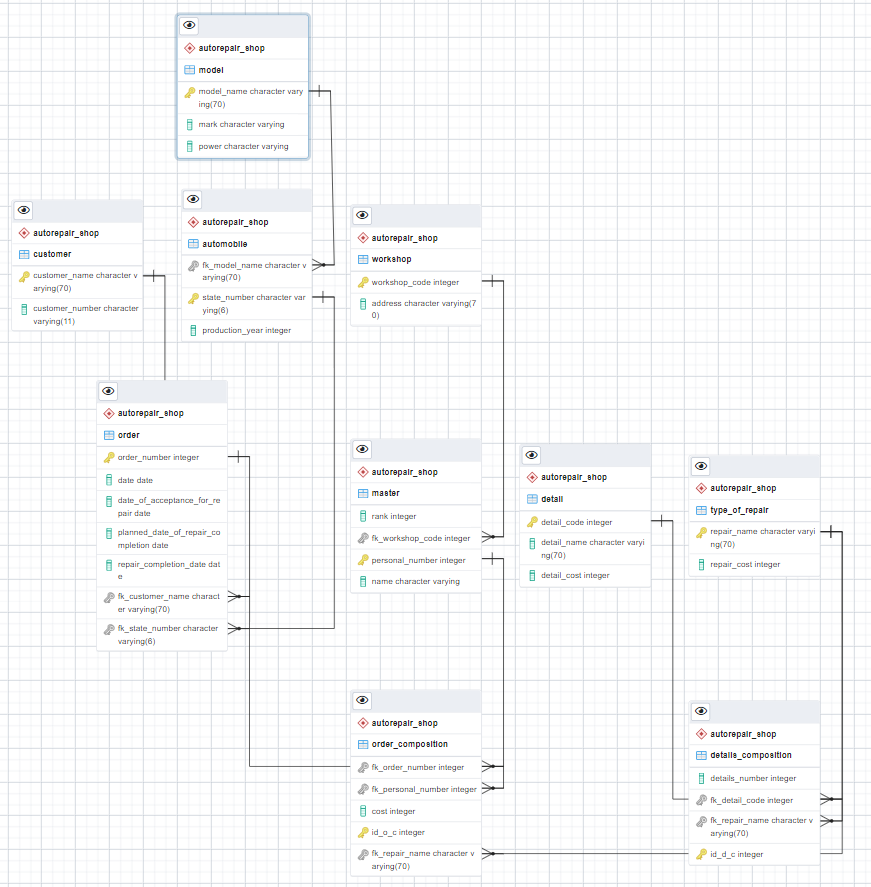


Рис. 1 — Схема логической модели БД, сгенерированная в ERD

**III. Dump, содержащий скрипты работы с БД**

1. Создание БД

CREATE SCHEMA autorepair\_shop;// *создание БД*

1. Создание таблиц

CREATE TABLE autorepair\_shop.automobile (

fk\_model\_name character varying(70) NOT NULL,

state\_number character varying(6) NOT NULL,

production\_year integer NOT NULL,

CONSTRAINT state\_number\_check CHECK (length(state\_number) = 6)

); // *создание таблицы* automobile, проверка длины гос. номера

CREATE TABLE autorepair\_shop.customer (

customer\_name character varying(70) NOT NULL,

customer\_number character varying(11) NOT NULL

); // *создание таблицы* customer

CREATE TABLE autorepair\_shop.detail (

detail\_code integer NOT NULL,

detail\_name character varying(70) NOT NULL,

detail\_cost integer NOT NULL

); // *создание таблицы* detail

CREATE TABLE autorepair\_shop.details\_composition (

details\_number integer,

fk\_detail\_code integer NOT NULL,

fk\_repair\_name character varying(70) NOT NULL,

id\_d\_c integer NOT NULL

); // *создание таблицы* details\_composition

CREATE TABLE autorepair\_shop.master (

rank integer DEFAULT 0 NOT NULL,

fk\_workshop\_code integer NOT NULL,

personal\_number integer NOT NULL,

name character varying NOT NULL,

CONSRTAINT rank\_check CHECK ((rank >= 0) AND (rank <= 7)

); // *создание таблицы* master, проверка разряда

CREATE TABLE autorepair\_shop.model (

model\_name character varying(70) NOT NULL,

mark character varying NOT NULL,

power character varying NOT NULL

); // *создание таблицы* model

CREATE TABLE autorepair\_shop.order (

order\_number integer NOT NULL,

date date NOT NULL,

date\_of\_acceptance\_for\_repair date NOT NULL,

planned\_date\_of\_repair\_completion date NOT NULL,

repair\_completion\_date date NOT NULL,

fk\_customer\_name character varying(70) NOT NULL,

fk\_state\_number character varying(6) NOT NULL,

CONSTRAINT date\_check CHECK ((date\_of\_acceptance\_for\_repair <= planned\_date\_of\_repair\_completion) AND (date\_of\_acceptance\_for\_repair <= repair\_completion\_date))

); // *создание таблицы* order, проверка даты

CREATE TABLE autorepair\_shop.order\_composition (

fk\_order\_number integer NOT NULL,

fk\_personal\_number integer NOT NULL,

cost integer NOT NULL,

id\_o\_c integer NOT NULL,

fk\_repair\_name character varying(70) NOT NULL

); // *создание таблицы* order\_composition

CREATE TABLE autorepair\_shop.type\_of\_repair (

repair\_name character varying(70) NOT NULL,

repair\_cost integer NOT NULL

); // *создание таблицы* type\_of\_repair

CREATE TABLE autorepair\_shop.workshop (

workshop\_code integer NOT NULL,

address character varying(70) NOT NULL

); // *создание таблицы* workshop

1. Ограничения на первичные ключи

ALTER TABLE ONLY autorepair\_shop.customer

ADD CONSTRAINT customer\_name PRIMARY KEY (customer\_name) INCLUDE (customer\_name); // *создание первичного ключа для таблицы* customer

ALTER TABLE ONLY autorepair\_shop.detail

ADD CONSTRAINT detail\_code PRIMARY KEY (detail\_code) INCLUDE (detail\_code); // *создание первичного ключа для таблицы* detail

ALTER TABLE ONLY autorepair\_shop.details\_composition

ADD CONSTRAINT id\_d\_c PRIMARY KEY (id\_d\_c) INCLUDE (id\_d\_c); // *создание первичного ключа для таблицы* details\_composition

ALTER TABLE ONLY autorepair\_shop.model

ADD CONSTRAINT model\_pkey PRIMARY KEY (model\_name); // *создание первичного ключа для таблицы* model

ALTER TABLE ONLY autorepair\_shop.order\_composition

ADD CONSTRAINT order\_composition\_pkey PRIMARY KEY (id\_o\_c); // *создание первичного ключа для таблицы* order\_composition

ALTER TABLE ONLY autorepair\_shop.master

ADD CONSTRAINT personal\_number PRIMARY KEY (personal\_number) INCLUDE (personal\_number); // *создание первичного ключа для таблицы* master

ALTER TABLE ONLY autorepair\_shop.type\_of\_repair

ADD CONSTRAINT repair PRIMARY KEY (repair\_name) INCLUDE (repair\_name);// *создание первичного ключа для таблицы* type\_of\_repair

ALTER TABLE ONLY autorepair\_shop.automobile

ADD CONSTRAINT state\_number PRIMARY KEY (state\_number) INCLUDE (state\_number); // *создание первичного ключа для таблицы* automobile

ALTER TABLE ONLY autorepair\_shop.order

ADD CONSTRAINT "Заказ\_pkey" PRIMARY KEY (order\_number); // *создание первичного ключа для таблицы* order

ALTER TABLE ONLY autorepair\_shop.workshop

ADD CONSTRAINT "Мастерская\_pkey" PRIMARY KEY (workshop\_code); // *создание первичного ключа для таблицы* workshop

1. Ограничения на внешние ключи

ALTER TABLE ONLY autorepair\_shop.order

ADD CONSTRAINT fk\_customer\_name FOREIGN KEY (fk\_customer\_name) REFERENCES autorepair\_shop.customer(customer\_name); // *поле* fk\_customer\_name *становится внешним ключом в таблице* order

ALTER TABLE ONLY autorepair\_shop.details\_composition

ADD CONSTRAINT fk\_detail\_code FOREIGN KEY (fk\_detail\_code) REFERENCES autorepair\_shop.detail(detail\_code); // *поле* fk\_detail\_code *становится внешним ключом в таблице* details\_composition

ALTER TABLE ONLY autorepair\_shop.automobile

ADD CONSTRAINT fk\_model\_name FOREIGN KEY (fk\_model\_name) REFERENCES autorepair\_shop.model(model\_name); // *поле* fk\_model\_name *становится внешним ключом в таблице* automobile

ALTER TABLE ONLY autorepair\_shop.order\_composition

ADD CONSTRAINT fk\_order\_number FOREIGN KEY (fk\_order\_number) REFERENCES autorepair\_shop."order"(order\_number) ON UPDATE RESTRICT ON DELETE RESTRICT; // *поле* fk\_order\_number *становится внешним ключом в таблице* order\_composition

ALTER TABLE ONLY autorepair\_shop.order\_composition

ADD CONSTRAINT fk\_personal\_number FOREIGN KEY (fk\_personal\_number) REFERENCES autorepair\_shop.master(personal\_number) NOT VALID; // *поле* fk\_personal\_number *становится внешним ключом в таблице* order\_composition

ALTER TABLE ONLY autorepair\_shop.details\_composition

ADD CONSTRAINT fk\_repair\_name FOREIGN KEY (fk\_repair\_name) REFERENCES autorepair\_shop.type\_of\_repair(repair\_name); // *поле* fk\_repair\_name *становится внешним ключом в таблице* details\_composition

ALTER TABLE ONLY autorepair\_shop.order\_composition

ADD CONSTRAINT fk\_repair\_name FOREIGN KEY (fk\_repair\_name) REFERENCES autorepair\_shop.type\_of\_repair(repair\_name); // *поле* fk\_repair\_name *становится внешним ключом в таблице* order\_composition

ALTER TABLE ONLY autorepair\_shop.order

ADD CONSTRAINT fk\_state\_number FOREIGN KEY (fk\_state\_number) REFERENCES autorepair\_shop.automobile(state\_number) NOT VALID; // *поле* fk\_state\_number *становится внешним ключом в таблице* order

ALTER TABLE ONLY autorepair\_shop.master

ADD CONSTRAINT fk\_workshop\_code FOREIGN KEY (fk\_workshop\_code) REFERENCES autorepair\_shop.workshop(workshop\_code) ON UPDATE RESTRICT ON DELETE RESTRICT; // *поле* fk\_workshop\_code *становится внешним ключом в таблице* master

1. Вставка данных

COPY autorepair\_shop.automobile (fk\_model\_name, state\_number, production\_year) FROM stdin;

Калина К209ЛВ 2017

Focus Д763РВ 2014

Camaro З730ИН 2019

\.

COPY autorepair\_shop.customer (customer\_name, customer\_number) FROM stdin;

Аксенов Петр Павлович 89114728427

Иванов Михаил Семенович 89634927474

Любимова Ксения Владимировна 89534927934

\.

COPY autorepair\_shop.detail (detail\_code, detail\_name, detail\_cost) FROM stdin;

30 Колесо 2500

96 Боковое зеркало 2000

109 Сидение 15000

\.

COPY autorepair\_shop.details\_composition (details\_number, fk\_detail\_code, fk\_repair\_name, id\_d\_c) FROM stdin;

4 30 Замена колес 0

2 96 Замена боковых зеркал 1

1 109 Замена сидения 2

\.

COPY autorepair\_shop.master (rank, fk\_workshop\_code, personal\_number, name) FROM stdin;

6 0 0 Иванов Иван Иванович

7 2 1 Кириленко Артем Петрович

7 3 2 Комаров Леонид Ринатович

\.

COPY autorepair\_shop.model (model\_name, mark, power) FROM stdin;

Калина Lada 90

Focus Ford 120

Camaro Chevrolet 100

\.

COPY autorepair\_shop.order (order\_number, date, date\_of\_acceptance\_for\_repair, planned\_date\_of\_repair\_completion, repair\_completion\_date, fk\_customer\_name, fk\_state\_number) FROM stdin;

0 2021-03-14 2021-03-14 2021-03-17 2021-03-17 Аксенов Петр Павлович К209ЛВ

1 2021-04-14 2021-04-14 2021-04-17 2021-04-17 Иванов Михаил Семенович Д763РВ

2 2021-05-14 2021-05-14 2021-05-17 2021-05-17 Любимова Ксения Владимировна З730ИН

\.

COPY autorepair\_shop.order\_composition (fk\_order\_number, fk\_personal\_number, cost, id\_o\_c, fk\_repair\_name) FROM stdin;

0 0 10000 0 Замена колес

1 1 4000 1 Замена боковых зеркал

2 2 15000 2 Замена сидения

\.

COPY autorepair\_shop.type\_of\_repair (repair\_name, repair\_cost) FROM stdin;

Замена колес 10990

Замена боковых зеркал 4000

Замена сидения 15000

\.

COPY autorepair\_shop.workshop (workshop\_code, address) FROM stdin;

0 пр.Кронверский 49

2 ул.Ломоносова 9

3 ул.Орджоникидзе 5

\.

**Выводы**

В ходе выполнения лабораторной работы была создана база данных, в неё были добавлены значения, а также созданы первичные и внешние ключи.