

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 № 1**

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

Проверила:

Говорова М.М.

Дата: «..» ... 2023 г.

Оценка \_\_\_\_\_

Выполнил:

студент группы К32422

Малаев С.Г.

Санкт-Петербург 2022/2023

**Цель работы:** овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, 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.

*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.

**Выполнение работы:**

Предметная область – автомастерская (вариант 11)

Наименование БД – auto\_repair\_shop

## Схемы логической модели базы данных

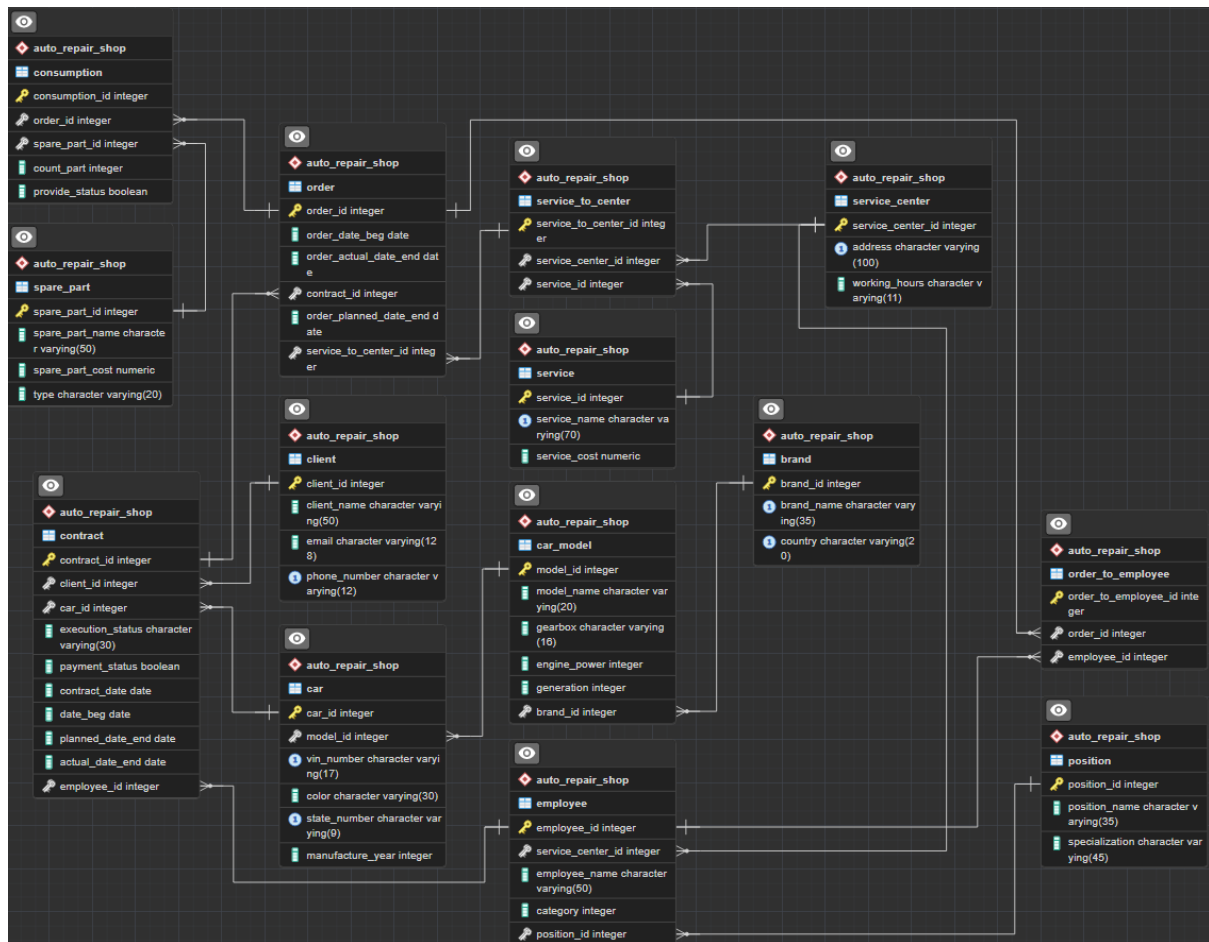


Рисунок 1 – ERD базы данных с pgAdmin

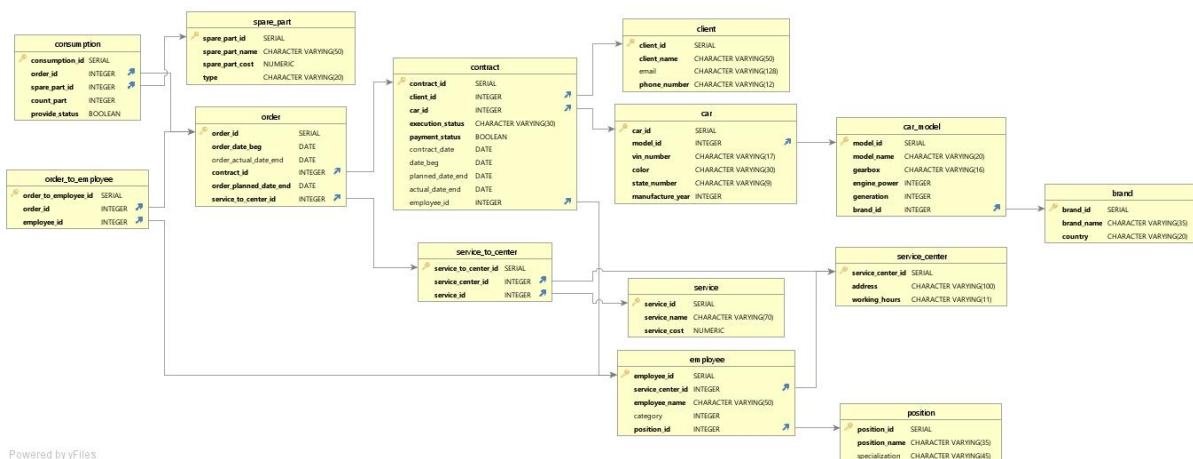


Рисунок 2 – ERD базы данных с DBVisualizer

## Dump-скрипты работы с БД

### 1. Создание базы данных

```
CREATE DATABASE auto_repair_shop
WITH TEMPLATE = template0
ENCODING = 'UTF8'
LOCALE_PROVIDER = libc
LOCALE = 'Russian_Russia.1251';
```

### 2. Создание схемы

```
CREATE SCHEMA auto_repair_shop;
```

### 3. Создание таблиц

#### 3.1. Создание таблицы «brand»

```
CREATE TABLE auto_repair_shop.brand (
    brand_id integer NOT NULL,
    brand_name character varying(35) NOT NULL,
    country character varying(20) NOT NULL
);
```

### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.brand_brand_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER SEQUENCE auto_repair_shop.brand_brand_id_seq OWNED BY
auto_repair_shop.brand.brand_id;

ALTER TABLE ONLY auto_repair_shop.brand
ALTER COLUMN brand_id SET DEFAULT
nextval('auto_repair_shop.brand_brand_id_seq'::regclass);
```

## Создание ограничений

```
ALTER TABLE ONLY auto_repair_shop.brand
    ADD CONSTRAINT pk_brand_id PRIMARY KEY (brand_id);

ALTER TABLE ONLY auto_repair_shop.brand
    ADD CONSTRAINT un_comp_brand UNIQUE (brand_name, country);
```

### 3.2. Создание таблицы «car»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.car (
    car_id integer NOT NULL,
    model_id integer NOT NULL,
    vin_number character varying(17) NOT NULL,
    color character varying(30) NOT NULL,
    state_number character varying(9) NOT NULL,
    manufacture_year integer NOT NULL
);
```

#### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.car_car_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER SEQUENCE auto_repair_shop.car_car_id_seq OWNED BY
auto_repair_shop.car.car_id;

ALTER TABLE ONLY auto_repair_shop.car ALTER COLUMN car_id SET
DEFAULT nextval('auto_repair_shop.car_car_id_seq'::regclass);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop.car
    ADD CONSTRAINT ch_state_number CHECK (((state_number)::text ~
similar_to_escape('[ABEKMHOPTX]\d{3}[ABEKMHOPTX]{2}\d{2,3}'))::t
ext))) NOT VALID;

ALTER TABLE auto_repair_shop.car
    ADD CONSTRAINT ch_vin_number CHECK (((vin_number)::text ~
similar_to_escape('[A-Za-z\d]{17}'))::text))) NOT VALID;

ALTER TABLE ONLY auto_repair_shop.car
    ADD CONSTRAINT pk_car_id PRIMARY KEY (car_id);

ALTER TABLE ONLY auto_repair_shop.car_model
    ADD CONSTRAINT pk_car_model_id PRIMARY KEY (model_id);

ALTER TABLE ONLY auto_repair_shop.car
    ADD CONSTRAINT un_state_number UNIQUE (state_number);

ALTER TABLE ONLY auto_repair_shop.car
    ADD CONSTRAINT un_vin_number UNIQUE (vin_number);

ALTER TABLE ONLY auto_repair_shop.car
    ADD CONSTRAINT fk_model_id FOREIGN KEY (model_id) REFERENCES
auto_repair_shop.car_model(model_id) ON UPDATE CASCADE NOT VALID;
```

### 3.3. Создание таблицы «car\_model»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.car_model (
    model_id integer NOT NULL,
    model_name character varying(20) NOT NULL,
    gearbox character varying(16) NOT NULL,
    engine_power integer NOT NULL,
    generation integer NOT NULL,
    brand_id integer NOT NULL
);
```

## Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.car_model_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE auto_repair_shop.car_model_model_id_seq OWNED BY
auto_repair_shop.car_model.model_id;

ALTER TABLE ONLY auto_repair_shop.car_model ALTER COLUMN model_id
SET DEFAULT
nextval('auto_repair_shop.car_model_model_id_seq'::regclass);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop.car_model
  ADD CONSTRAINT ch_engine_power CHECK ((engine_power > 0)) NOT
VALID;

ALTER TABLE auto_repair_shop.car_model
  ADD CONSTRAINT ch_gearbox CHECK (((gearbox)::text = ANY
((ARRAY['автоматическая'::character varying,
'роботизированная'::character varying, 'вариативная'::character
varying, 'механическая'::character varying]))::text[]))) NOT VALID;

ALTER TABLE auto_repair_shop.car_model
  ADD CONSTRAINT ch_generation CHECK ((generation > 0)) NOT
VALID;

ALTER TABLE ONLY auto_repair_shop.car_model
  ADD CONSTRAINT pk_car_model_id PRIMARY KEY (model_id);

ALTER TABLE ONLY auto_repair_shop.car_model
  ADD CONSTRAINT fk_brand_id FOREIGN KEY (brand_id) REFERENCES
auto_repair_shop.brand(brand_id) ON UPDATE CASCADE NOT VALID;
```

### 3.4. Создание таблицы «client»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.client (  
    client_id integer NOT NULL,  
    client_name character varying(50) NOT NULL,  
    email character varying(128),  
    phone_number character varying(12) NOT NULL  
);
```

#### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.client_client_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER SEQUENCE auto_repair_shop.client_client_id_seq OWNED BY  
auto_repair_shop.client.client_id;  
  
ALTER TABLE ONLY auto_repair_shop.client ALTER COLUMN client_id  
SET DEFAULT  
nextval('auto_repair_shop.client_client_id_seq'::regclass);
```

#### Создание ограничений

```
ALTER TABLE auto_repair_shop.client  
    ADD CONSTRAINT ch_phone_number CHECK (((phone_number)::text ~  
similar_to_escape('\+\d{11}'::text))) NOT VALID;  
  
ALTER TABLE ONLY auto_repair_shop.client  
    ADD CONSTRAINT pk_client_id PRIMARY KEY (client_id);  
  
ALTER TABLE ONLY auto_repair_shop.client  
    ADD CONSTRAINT un_phone_number UNIQUE (phone_number);
```



### 3.5. Создание таблицы «consumption»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.consumption (  
    consumption_id integer NOT NULL,  
    order_id integer NOT NULL,  
    spare_part_id integer NOT NULL,  
    count_part integer NOT NULL,  
    provide_status boolean NOT NULL  
);
```

#### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.consumption_consumption_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER SEQUENCE auto_repair_shop.consumption_consumption_id_seq  
OWNED BY auto_repair_shop.consumption.consumption_id;  
  
ALTER TABLE ONLY auto_repair_shop.consumption ALTER COLUMN  
consumption_id SET DEFAULT  
nextval('auto_repair_shop.consumption_consumption_id_seq'::regclas  
s);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop.consumption
    ADD CONSTRAINT ch_count_part CHECK ((count_part > 0)) NOT
VALID;

ALTER TABLE ONLY auto_repair_shop.consumption
    ADD CONSTRAINT pk_consumption_id PRIMARY KEY (consumption_id);

ALTER TABLE ONLY auto_repair_shop.consumption
    ADD CONSTRAINT un_comp_consumption UNIQUE (order_id,
spare_part_id);

ALTER TABLE ONLY auto_repair_shop.consumption
    ADD CONSTRAINT fk_order_id FOREIGN KEY (order_id) REFERENCES
auto_repair_shop."order"(order_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY auto_repair_shop.consumption
    ADD CONSTRAINT fk_spare_part_id FOREIGN KEY (spare_part_id)
REFERENCES auto_repair_shop.spare_part(spare_part_id) ON UPDATE
CASCADE NOT VALID;
```

### 3.6. Создание таблицы «contract»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.contract (
    contract_id integer NOT NULL,
    client_id integer NOT NULL,
    car_id integer NOT NULL,
    execution_status character varying(30) NOT NULL,
    payment_status boolean NOT NULL,
    contract_date date,
    date_beg date,
    planned_date_end date,
    actual_date_end date,
    employee_id integer
);
```

## Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.contract_car_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE auto_repair_shop.contract_car_id_seq OWNED BY
auto_repair_shop.contract.car_id;

ALTER TABLE ONLY auto_repair_shop.contract ALTER COLUMN
contract_id SET DEFAULT
nextval('auto_repair_shop.contract_contract_id_seq'::regclass);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop.contract
  ADD CONSTRAINT ch_actual_date_end CHECK (((actual_date_end IS
NULL) OR (date_beg < actual_date_end))) NOT VALID;

ALTER TABLE auto_repair_shop.contract
  ADD CONSTRAINT ch_contract_date CHECK ((contract_date <=
now())) NOT VALID;

ALTER TABLE auto_repair_shop.contract
  ADD CONSTRAINT ch_date_beg CHECK ((date_beg <= now())) NOT
VALID;

ALTER TABLE auto_repair_shop.contract
  ADD CONSTRAINT ch_execution_status CHECK
(((execution_status)::text = ANY (ARRAY[('Ожидание
оплаты'::character varying)::text, ('Принят'::character
varying)::text, ('Обработка'::character varying)::text,
('Выполнение'::character varying)::text, ('Ожидание подтверждения
клиента'::character varying)::text, ('Выполнен'::character
varying)::text]))) NOT VALID;
```

```

ALTER TABLE auto_repair_shop.contract
    ADD CONSTRAINT ch_execution_status_to_actual_date_end CHECK
(((actual_date_end IS NULL) OR ((execution_status)::text =
'Выполнен'::text))) NOT VALID;

ALTER TABLE auto_repair_shop.contract
    ADD CONSTRAINT ch_payment_status_to_contract_date CHECK
(((contract_date IS NULL) OR payment_status)) NOT VALID;

ALTER TABLE auto_repair_shop.contract
    ADD CONSTRAINT ch_planned_date_end CHECK (((planned_date_end
IS NULL) OR (date_beg < planned_date_end))) NOT VALID;

ALTER TABLE ONLY auto_repair_shop.contract
    ADD CONSTRAINT pk_contract_id PRIMARY KEY (contract_id);

ALTER TABLE ONLY auto_repair_shop.contract
    ADD CONSTRAINT fk_car_id FOREIGN KEY (car_id) REFERENCES
auto_repair_shop.car(car_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY auto_repair_shop.contract
    ADD CONSTRAINT fk_client_id FOREIGN KEY (client_id) REFERENCES
auto_repair_shop.client(client_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY auto_repair_shop.contract
    ADD CONSTRAINT fk_employee_id FOREIGN KEY (employee_id)
REFERENCES auto_repair_shop.employee(employee_id) ON UPDATE
CASCADE NOT VALID;

```

### 3.7. Создание таблицы «employee»

#### Создание таблицы

```

CREATE TABLE auto_repair_shop.employee (
    employee_id integer NOT NULL,
    service_center_id integer NOT NULL,
    employee_name character varying(50) NOT NULL,
    category integer,
    position_id integer NOT NULL
);

```

## Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.employee_employee_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE auto_repair_shop.employee_employee_id_seq OWNED BY
auto_repair_shop.employee.employee_id;

ALTER TABLE ONLY auto_repair_shop.employee ALTER COLUMN
employee_id SET DEFAULT
nextval('auto_repair_shop.employee_employee_id_seq'::regclass);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop.employee
  ADD CONSTRAINT ch_category CHECK (((category IS NULL) OR
((category >= 1) AND (category <= 9)))) NOT VALID;

ALTER TABLE ONLY auto_repair_shop.employee
  ADD CONSTRAINT pk_employee_id PRIMARY KEY (employee_id);

ALTER TABLE ONLY auto_repair_shop.employee
  ADD CONSTRAINT fk_position_id FOREIGN KEY (position_id)
REFERENCES auto_repair_shop."position"(position_id) ON UPDATE
CASCADE NOT VALID;

ALTER TABLE ONLY auto_repair_shop.employee
  ADD CONSTRAINT fk_service_center_id FOREIGN KEY
(service_center_id) REFERENCES
auto_repair_shop.service_center(service_center_id) ON UPDATE
CASCADE NOT VALID;
```

### 3.8. Создание таблицы «order»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop."order" (  
    order_id integer NOT NULL,  
    order_date_beg date NOT NULL,  
    order_actual_date_end date,  
    contract_id integer NOT NULL,  
    order_planned_date_end date NOT NULL,  
    service_to_center_id integer NOT NULL  
);
```

#### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.order_order_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER SEQUENCE auto_repair_shop.order_order_id_seq OWNED BY  
auto_repair_shop."order".order_id;  
  
ALTER TABLE ONLY auto_repair_shop."order" ALTER COLUMN order_id  
SET DEFAULT  
nextval('auto_repair_shop.order_order_id_seq'::regclass);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop."order"  
    ADD CONSTRAINT ch_order_actual_date_end CHECK  
(((order_actual_date_end IS NULL) OR (order_date_beg <=  
order_actual_date_end))) NOT VALID;  
  
ALTER TABLE auto_repair_shop."order"  
    ADD CONSTRAINT ch_order_date_beg CHECK ((order_date_beg <  
now())) NOT VALID;  
  
ALTER TABLE auto_repair_shop."order"  
    ADD CONSTRAINT ch_order_planned_date_end CHECK  
((order_date_beg <= order_planned_date_end)) NOT VALID;  
  
ALTER TABLE ONLY auto_repair_shop."order"  
    ADD CONSTRAINT pk_order_id PRIMARY KEY (order_id);  
  
ALTER TABLE ONLY auto_repair_shop."order"  
    ADD CONSTRAINT un_comp_order UNIQUE (contract_id,  
service_to_center_id);  
  
ALTER TABLE ONLY auto_repair_shop."order"  
    ADD CONSTRAINT fk_contract_id FOREIGN KEY (contract_id)  
REFERENCES auto_repair_shop.contract(contract_id) ON UPDATE  
CASCADE NOT VALID;  
  
ALTER TABLE ONLY auto_repair_shop."order"  
    ADD CONSTRAINT fk_service_to_center_id FOREIGN KEY  
(service_to_center_id) REFERENCES  
auto_repair_shop.service_to_center(service_to_center_id) ON UPDATE  
CASCADE NOT VALID;
```

### 3.9. Создание таблицы «order\_to\_employee»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.order_to_employee (  
    order_to_employee_id integer NOT NULL,  
    order_id integer NOT NULL,  
    employee_id integer NOT NULL  
);
```

## Создание первичного ключа serial

```
CREATE SEQUENCE
auto_repair_shop.order_to_employee_order_to_employee_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE
auto_repair_shop.order_to_employee_order_to_employee_id_seq OWNED
BY auto_repair_shop.order_to_employee.order_to_employee_id;

ALTER TABLE ONLY auto_repair_shop.order_to_employee ALTER COLUMN
order_to_employee_id SET DEFAULT
nextval('auto_repair_shop.order_to_employee_order_to_employee_id_s
eq'::regclass);
```

## Создание ограничений

```
ALTER TABLE ONLY auto_repair_shop.order_to_employee
  ADD CONSTRAINT order_to_employee_pkey PRIMARY KEY
(order_to_employee_id);

ALTER TABLE ONLY auto_repair_shop.order_to_employee
  ADD CONSTRAINT un_comp_order_to_employee UNIQUE (order_id,
employee_id);

ALTER TABLE ONLY auto_repair_shop.order_to_employee
  ADD CONSTRAINT fk_employee_id FOREIGN KEY (employee_id)
REFERENCES auto_repair_shop.employee(employee_id) ON UPDATE
CASCADE NOT VALID;

ALTER TABLE ONLY auto_repair_shop.order_to_employee
  ADD CONSTRAINT fk_order_id FOREIGN KEY (order_id) REFERENCES
auto_repair_shop."order"(order_id) ON UPDATE CASCADE NOT VALID;
```



### 3.10. Создание таблицы «position»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop."position" (  
    position_id integer NOT NULL,  
    position_name character varying(35) NOT NULL,  
    specialization character varying(45)  
);
```

#### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.position_position_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER SEQUENCE auto_repair_shop.position_position_id_seq OWNED BY  
auto_repair_shop."position".position_id;  
  
ALTER TABLE ONLY auto_repair_shop."position" ALTER COLUMN  
position_id SET DEFAULT  
nextval('auto_repair_shop.position_position_id_seq'::regclass);
```

#### Создание ограничений

```
ALTER TABLE ONLY auto_repair_shop."position"  
    ADD CONSTRAINT pk_position_id PRIMARY KEY (position_id);
```

### 3.11. Создание таблицы «service»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.service (  
    service_id integer NOT NULL,  
    service_name character varying(70) NOT NULL,  
    service_cost numeric NOT NULL  
);
```

### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.service_service_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE auto_repair_shop.service_service_id_seq OWNED BY
auto_repair_shop.service.service_id;

ALTER TABLE ONLY auto_repair_shop.service ALTER COLUMN service_id
SET DEFAULT
nextval('auto_repair_shop.service_service_id_seq'::regclass);
```

### Создание ограничений

```
ALTER TABLE auto_repair_shop.service
  ADD CONSTRAINT ch_service_cost CHECK ((service_cost >
(0)::numeric)) NOT VALID;

ALTER TABLE ONLY auto_repair_shop.service
  ADD CONSTRAINT pk_service_id PRIMARY KEY (service_id);

ALTER TABLE ONLY auto_repair_shop.service
  ADD CONSTRAINT un_service_name UNIQUE (service_name);
```

## 3.12. Создание таблицы «service\_center»

### Создание таблицы

```
CREATE TABLE auto_repair_shop.service_center (
  service_center_id integer NOT NULL,
  address character varying(100) NOT NULL,
  working_hours character varying(11) NOT NULL
);
```

## Создание первичного ключа serial

```
CREATE SEQUENCE
auto_repair_shop.service_center_service_center_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE
auto_repair_shop.service_center_service_center_id_seq OWNED BY
auto_repair_shop.service_center.service_center_id;

ALTER TABLE ONLY auto_repair_shop.service_center ALTER COLUMN
service_center_id SET DEFAULT
nextval('auto_repair_shop.service_center_service_center_id_seq'::r
egclass);
```

## Создание ограничений

```
ALTER TABLE auto_repair_shop.service_center
  ADD CONSTRAINT ch_working_hours CHECK (((working_hours)::text
~ similar_to_escape('\d{2}:\d{2}-\d{2}:\d{2}')) NOT VALID;

ALTER TABLE ONLY auto_repair_shop.service_center
  ADD CONSTRAINT pk_service_center PRIMARY KEY
(service_center_id);

ALTER TABLE ONLY auto_repair_shop.service_center
  ADD CONSTRAINT un_address UNIQUE (address);
```

### 3.13. Создание таблицы «service\_to\_center»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.service_to_center (
  service_to_center_id integer NOT NULL,
  service_center_id integer NOT NULL,
  service_id integer NOT NULL
);
```

## Создание первичного ключа serial

```
CREATE SEQUENCE
auto_repair_shop.service_to_center_service_to_center_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER SEQUENCE
auto_repair_shop.service_to_center_service_to_center_id_seq OWNED
BY auto_repair_shop.service_to_center.service_to_center_id;

ALTER TABLE ONLY auto_repair_shop.service_to_center ALTER COLUMN
service_to_center_id SET DEFAULT
nextval('auto_repair_shop.service_to_center_service_to_center_id_s
eq'::regclass);
```

## Создание ограничений

```
ALTER TABLE ONLY auto_repair_shop.service_to_center
  ADD CONSTRAINT pk_service_to_center PRIMARY KEY
(service_to_center_id);

ALTER TABLE ONLY auto_repair_shop.service_to_center
  ADD CONSTRAINT un_comp_service_to_center UNIQUE
(service_center_id, service_id);

ALTER TABLE ONLY auto_repair_shop.service_to_center
  ADD CONSTRAINT fk_service_center_id FOREIGN KEY
(service_center_id) REFERENCES
auto_repair_shop.service_center(service_center_id) ON UPDATE
CASCADE NOT VALID;

ALTER TABLE ONLY auto_repair_shop.service_to_center
  ADD CONSTRAINT fk_service_id FOREIGN KEY (service_id)
REFERENCES auto_repair_shop.service(service_id) ON UPDATE CASCADE
NOT VALID;
```

### 3.14. Создание таблицы «spare\_part»

#### Создание таблицы

```
CREATE TABLE auto_repair_shop.spare_part (  
    spare_part_id integer NOT NULL,  
    spare_part_name character varying(50) NOT NULL,  
    spare_part_cost numeric NOT NULL,  
    type character varying(20) NOT NULL  
);
```

#### Создание первичного ключа serial

```
CREATE SEQUENCE auto_repair_shop.spare_part_spare_part_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER SEQUENCE auto_repair_shop.spare_part_spare_part_id_seq OWNED  
    BY auto_repair_shop.spare_part.spare_part_id;  
  
ALTER TABLE ONLY auto_repair_shop.spare_part ALTER COLUMN  
    spare_part_id SET DEFAULT  
    nextval('auto_repair_shop.spare_part_spare_part_id_seq'::regclass)  
    ;
```

#### Создание ограничений

```
ALTER TABLE auto_repair_shop.spare_part  
    ADD CONSTRAINT ch_spare_part_cost CHECK ((spare_part_cost >  
    (0)::numeric)) NOT VALID;  
  
ALTER TABLE ONLY auto_repair_shop.spare_part  
    ADD CONSTRAINT pk_spare_part PRIMARY KEY (spare_part_id);
```

#### 4. Заполнение рабочими данными

##### 4.1. Заполнение таблицы «brand»

```
COPY auto_repair_shop.brand (brand_id, brand_name, country) FROM
stdin;
1    Mitsubishi Япония
2    Daewoo      Корея
3    Volvo      Швеция
...
35   Opel       Германия
36   Nissan     Япония
37   Vauxhall   Германия
\.
```

##### 4.2. Заполнение таблицы «car»

```
COPY auto_repair_shop.car (car_id, model_id, vin_number, color,
state_number, manufacture_year) FROM stdin;
1    435    N6ZEGW1HA5A5ID6NH    Сливочно-белый    C194B014    2001
2    493    HCBJG5Y6FL6IJ8PG7    Серебристый болотно-зелёный
X010CA44    2019
3    498    MW4KJMBI6WREN6JWJ    Красный    C520PX716    2005
...
998  623    ZJSQ3S220MHNKC0F8    Голубой    H860PT164    2016
999  260    YWTJDPOZKQ04800X5    Тёмно-фиолетовый    A477HA15
1992
1000 402    R7ZQJZJWQCCNHC8D5    Серебристо-голубоватый
K314E013    2020
\.
```

##### 4.3. Заполнение таблицы «car\_model»

```
COPY auto_repair_shop.car_model (model_id, model_name, gearbox,
engine_power, generation, brand_id) FROM stdin;
1    Corolla    вариативная210    4    21
2    Corolla    автоматическая    155    4    21
3    Corolla    роботизированная    165    9    21
...
658  Bravo вариативная195    8    17
659  Bravo автоматическая    185    4    17
660  Bravo вариативная215    1    17
```

```
\.
```

#### 4.4. Заполнение таблицы «client»

```
COPY auto_repair_shop.client (client_id, client_name, email,
phone_number) FROM stdin;
1    Абрамов Владимир Матвеевич      chickengodzilla@gmail.com
+79719452698
2    Абрамов Егор Ярославович    sapphirepaper@yandex.ru
+74142991137
3    Абрамов Владимир Матвеевич      FantasyCloud@yandex.ru
+79565712142
...
998  Яковлев Андрей Александрович    \N    +74742157718
999  Яковлева Ирина Романовна    \N    +74325688942
1000 Яшина Елена Артёмовна    \N    +79251991861
\.
```

#### 4.5. Заполнение таблицы «consumption»

```
COPY auto_repair_shop.consumption (consumption_id, order_id,
spare_part_id, count_part, provide_status) FROM stdin;
1    1    28    1    f
2    2    6     3    f
3    2    99    4    f
...
9762 3902 80    5    f
9763 3902 57    1    t
9764 3902 87    4    f
\.
```

#### 4.6. Заполнение таблицы «contract»

```
COPY auto_repair_shop.contract (contract_id, client_id, car_id,
execution_status, payment_status, contract_date, date_beg,
planned_date_end, actual_date_end, employee_id) FROM stdin;
1    521  559  Ожидание оплаты f    \N    \N    \N    \N    \N
2    530  350  Ожидание оплаты f    \N    \N    \N    \N    \N
3    67   412  Ожидание оплаты f    \N    \N    \N    \N    \N
...
5248 588  120  Принят      t    2021-01-07 \N    \N    \N    7
5249 102  245  Ожидание оплаты f    \N    \N    \N    \N    \N
```

```
5250 443 96 Обработка t 2022-03-01 \N \N \N 85
\.
```

#### 4.7. Заполнение таблицы «employee»

```
COPY auto_repair_shop.employee (employee_id, service_center_id,
employee_name, category, position_id) FROM stdin;
1 1 Широкова Софья Александровна \N 17
2 1 Савельев Степан Максимович \N 11
3 1 Смирнова София Тимофеевна \N 41
...
123 10 Новиков Роман Тимофеевич 5 20
124 10 Калмыков Александр Георгиевич 9 15
125 10 Малаев Степан Геннадьевич \N 40
\.
```

#### 4.8. Заполнение таблицы «order»

```
COPY auto_repair_shop."order" (order_id, order_date_beg,
order_actual_date_end, contract_id, order_planned_date_end,
service_to_center_id) FROM stdin;
1 2021-04-08 2021-05-04 46 2021-04-20 574
2 2021-05-01 2021-05-29 47 2021-06-19 233
3 2021-05-19 2021-05-21 54 2021-05-22 44
...
3900 2021-01-08 2021-02-09 5107 2021-02-05 398
3901 2021-05-03 2021-05-09 5112 2021-05-12 437
3902 2021-04-17 2021-05-18 5112 2021-04-20 303
\.
```

#### 4.9. Заполнение таблицы «order\_to\_employee»

```
COPY auto_repair_shop.order_to_employee (order_to_employee_id,
order_id, employee_id) FROM stdin;
1 1 19
2 2 16
3 2 116
...
7770 3902 16
7771 3902 30
7772 3902 69
\.
```





#### 4.10. Заполнение таблицы «position»

```
COPY auto_repair_shop."position" (position_id, position_name,
specialization) FROM stdin;
1      Автомаляр Жестянщик
2      Автомаляр Кузовщик
3      Автомаляр Универсал
...
39     Руководитель отдела по гарантии \N
40     Директор \N
41     Уборщик \N
\.
```

#### 4.11. Заполнение таблицы «service»

```
COPY auto_repair_shop.service (service_id, service_name,
service_cost) FROM stdin;
1      Замена масла в ДВС 3500.00
2      Замена масляного фильтра 1250.00
3      Замена форсунок 1500.00
...
87     Ремонт электрики в машинах 5750.00
88     Установка и обслуживание газового оборудования в автомобилях
3750.00
89     Реставрация авто и запчастей 7500.00
\.
```

#### 4.12. Заполнение таблицы «service\_center»

```
COPY auto_repair_shop.service_center (service_center_id, address,
working_hours) FROM stdin;
1      109417, Московская область, город Москва, пл. Ладыгина, 41
10:00-22:00
2      109428, Московская область, город Москва, пер. Ладыгина, 01
08:00-20:00
3      107150, Московская область, город Москва, спуск Ленина, 98
10:00-22:00
...
8      115184, Московская область, город Москва, пер. Бухарестская,
14      10:00-22:00
9      115114, Московская область, город Москва, ул. Сталина, 21
11:00-23:00
10     191014, Ленинградская область, город Санкт-Петербург, въезд
Ладыгина, 85      08:00-20:00
\.
```

#### 4.13. Заполнение таблицы «service\_to\_center»

```
COPY auto_repair_shop.service_to_center (service_to_center_id,
service_center_id, service_id) FROM stdin;
1      1      1
2      1      2
3      1      3
...
697    10      42
698    10      80
699    10      52
\.
```

#### 4.14. Заполнение таблицы «spare\_part»

```
COPY auto_repair_shop.spare_part (spare_part_id, spare_part_name,  
spare_part_cost, type) FROM stdin;  
1      Блок управления ABS    7600  Электроника  
2      Генератор    9900  Электроника  
3      Шаровая      2000  Ходовая  
...  
102    Судовой крыльчатый движитель      600  Электроника  
103    Уплотнители350  Расходники  
104    Фурнитура    650  Расходники  
\. 
```

#### **Выводы:**

В ходе выполнения данной лабораторной работы была изучена процедура создания таблиц в базе данных PostgreSQL с помощью инструмента pgAdmin. Были рассмотрены основные типы данных, используемые при создании таблиц, а также различные ограничения и backup tool базы данных.

Также была проведена работа по заполнению таблиц рабочими данными. Для этого был использован метод импорта данных из внешних источников. Большинство данных были сгенерированы и собраны с различных интернет-ресурсов с помощью python.

В результате выполнения лабораторной работы было получено практическое представление о том, как создавать таблицы в базе данных PostgreSQL и заполнять их данными. Эти навыки будут полезны при проектировании и реализации баз данных в реальных проектах.