

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»**

Факультет инфокоммуникационных технологий

Дисциплина:

«Проектирование и реализация баз данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
«СОЗДАНИЕ БД POSTGRESQL В PGADMIN. РЕЗЕРВНОЕ
КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ БД»**

Выполнил:

студент группы К32392

Жаров Александр Павлович

(подпись)

Проверил(а):

Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

Цель работы 1.1: овладеть практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4.

Практическое задание 1.1:

1. Установить СУБД PostgreSQL 1X.
2. Создать базу данных с использованием pgadmin 4.

Цель работы 1.2: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Практическое задание 1.2:

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

Наименование БД: restaurant

ERD диаграмма:

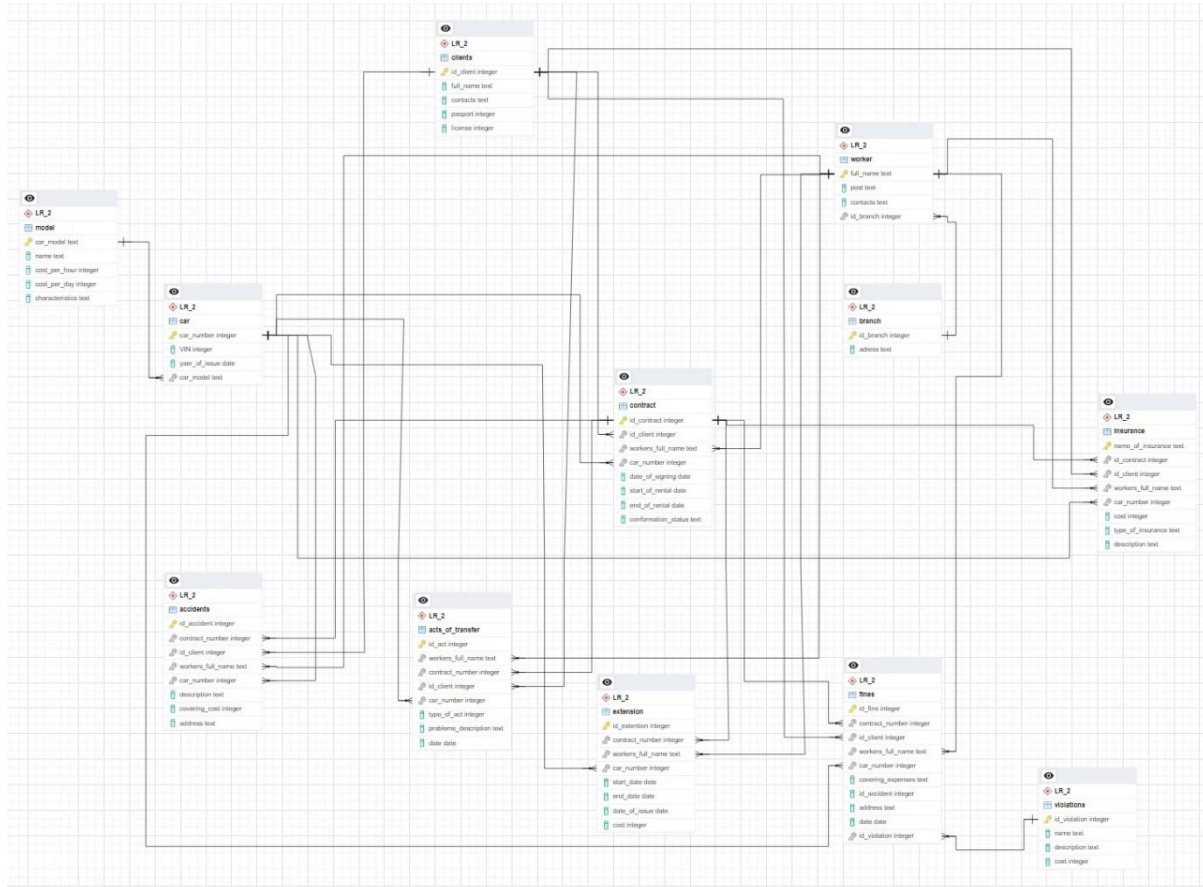


Рисунок 1 - ERD диаграмма

Dump БД: Бекап базы данных в двух вариантах, сгенерированный в pgAdmin, приложен к отчету. Файлы: lr2_custom_backup.sql, lr2_plain_backup.sql

Скрипты работы с БД для создания таблиц:

CREATE TABLE IF NOT EXISTS "LR_2".accidents

(

id_accident integer NOT NULL,

contract_number integer NOT NULL,

id_client integer NOT NULL,

workers_full_name text COLLATE pg_catalog."default" NOT NULL,

car_number integer NOT NULL,
description text COLLATE pg_catalog."default" NOT NULL,
covering_cost integer NOT NULL,
address text COLLATE pg_catalog."default" NOT NULL,
CONSTRAINT accidents_pkey PRIMARY KEY (id_accident),
CONSTRAINT car_number FOREIGN KEY (car_number)
REFERENCES "LR_2".car (car_number) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID,
CONSTRAINT id_client FOREIGN KEY (id_client)
REFERENCES "LR_2".clients (id_client) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID,
CONSTRAINT id_contract FOREIGN KEY (contract_number)
REFERENCES "LR_2".contract (id_contract) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID,
CONSTRAINT workers_full_name FOREIGN KEY (workers_full_name)
REFERENCES "LR_2".worker (full_name) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".accidents

OWNER to postgres;

COMMENT ON TABLE "LR_2".accidents

IS 'Аварии';

CREATE TABLE IF NOT EXISTS "LR_2".acts_of_transfer

(

id_act integer NOT NULL,

workers_full_name text COLLATE pg_catalog."default" NOT NULL,

contract_number integer NOT NULL,

id_client integer NOT NULL,

car_number integer NOT NULL,

type_of_act integer NOT NULL,

problems_description text COLLATE pg_catalog."default" NOT NULL,

date date NOT NULL,

CONSTRAINT acts_of_transfer_pkey PRIMARY KEY (id_act),

CONSTRAINT car_number FOREIGN KEY (car_number)

REFERENCES "LR_2".car (car_number) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id_client FOREIGN KEY (id_client)

REFERENCES "LR_2".clients (id_client) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id_contract FOREIGN KEY (contract_number)

REFERENCES "LR_2".contract (id_contract) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT workers_full_name FOREIGN KEY (workers_full_name)

REFERENCES "LR_2".worker (full_name) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".acts_of_transfer

OWNER to postgres;

COMMENT ON TABLE "LR_2".acts_of_transfer

IS 'Акты передачи';

```
CREATE TABLE IF NOT EXISTS "LR_2".branch
(
    id_branch integer NOT NULL,
    adress text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT branch_pkey PRIMARY KEY (id_branch)
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS "LR_2".branch
    OWNER to postgres;
```

```
COMMENT ON TABLE "LR_2".branch
    IS 'Филиал';
```

```
CREATE TABLE IF NOT EXISTS "LR_2".car
(
    car_number integer NOT NULL,
    "VIN" integer NOT NULL,
    yaer_of_issue date NOT NULL,
    car_model text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT car_pkey PRIMARY KEY (car_number),
    CONSTRAINT model FOREIGN KEY (car_model)
        REFERENCES "LR_2".model (car_model) MATCH SIMPLE
```

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".car

OWNER to postgres;

COMMENT ON TABLE "LR_2".car

IS 'Автомобиль';

CREATE TABLE IF NOT EXISTS "LR_2".clients

(

id_client integer NOT NULL,

full_name text COLLATE pg_catalog."default" NOT NULL,

contacts text COLLATE pg_catalog."default" NOT NULL,

passport integer NOT NULL,

license integer NOT NULL,

CONSTRAINT clients_pkey PRIMARY KEY (id_client)

)

TABLESPACE pg_default;


```
ALTER TABLE IF EXISTS "LR_2".clients
```

```
OWNER to postgres;
```

```
COMMENT ON TABLE "LR_2".clients
```

```
IS 'Клиенты';
```

```
CREATE TABLE IF NOT EXISTS "LR_2".contract
```

```
(
```

```
id_contract integer NOT NULL,
```

```
id_client integer NOT NULL,
```

```
workers_full_name text COLLATE pg_catalog."default" NOT NULL,
```

```
car_number integer NOT NULL,
```

```
date_of_signing date,
```

```
start_of_rental date,
```

```
end_of_rental date,
```

```
conformation_status text COLLATE pg_catalog."default",
```

```
CONSTRAINT contract_pkey PRIMARY KEY (id_contract),
```

```
CONSTRAINT car_number FOREIGN KEY (car_number)
```

```
REFERENCES "LR_2".car (car_number) MATCH SIMPLE
```

```
ON UPDATE NO ACTION
```

```
ON DELETE NO ACTION
```

```
NOT VALID,
```

```
CONSTRAINT id_client FOREIGN KEY (id_client)
```

```
REFERENCES "LR_2".clients (id_client) MATCH SIMPLE
```

```
ON UPDATE NO ACTION
```

```
ON DELETE NO ACTION  
  
NOT VALID,  
  
CONSTRAINT workers_full_name FOREIGN KEY (workers_full_name)  
  
REFERENCES "LR_2".worker (full_name) MATCH SIMPLE  
  
ON UPDATE NO ACTION  
  
ON DELETE NO ACTION  
  
NOT VALID  
  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS "LR_2".contract
```

```
OWNER to postgres;
```

```
COMMENT ON TABLE "LR_2".contract
```

```
IS 'Договор';
```

```
CREATE TABLE IF NOT EXISTS "LR_2".extension
```

```
(  
  
id_extention integer NOT NULL,  
  
contract_number integer NOT NULL,  
  
workers_full_name text COLLATE pg_catalog."default" NOT NULL,  
  
car_number integer NOT NULL,  
  
start_date date,  
  
end_date date,
```

```
date_of_issue date,  
  
cost integer NOT NULL,  
  
CONSTRAINT extension_pkey PRIMARY KEY (id_extention),  
  
CONSTRAINT car_number FOREIGN KEY (car_number)  
  
    REFERENCES "LR_2".car (car_number) MATCH SIMPLE  
  
    ON UPDATE NO ACTION  
  
    ON DELETE NO ACTION  
  
    NOT VALID,  
  
CONSTRAINT contract_name FOREIGN KEY (contract_number)  
  
    REFERENCES "LR_2".contract (id_contract) MATCH SIMPLE  
  
    ON UPDATE NO ACTION  
  
    ON DELETE NO ACTION  
  
    NOT VALID,  
  
CONSTRAINT workers_full_name FOREIGN KEY (workers_full_name)  
  
    REFERENCES "LR_2".worker (full_name) MATCH SIMPLE  
  
    ON UPDATE NO ACTION  
  
    ON DELETE NO ACTION  
  
    NOT VALID  
  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS "LR_2".extension  
  
    OWNER to postgres;
```

COMMENT ON TABLE "LR_2".extension

IS 'Продление';

CREATE TABLE IF NOT EXISTS "LR_2".fines

(

id_fine integer NOT NULL,

contract_number integer NOT NULL,

id_client integer NOT NULL,

workers_full_name text COLLATE pg_catalog."default" NOT NULL,

car_number integer NOT NULL,

covering_expenses text COLLATE pg_catalog."default" NOT NULL,

id_accident integer NOT NULL,

address text COLLATE pg_catalog."default" NOT NULL,

date date NOT NULL,

id_violation integer NOT NULL,

CONSTRAINT fines_pkey PRIMARY KEY (id_fine),

CONSTRAINT car_number FOREIGN KEY (car_number)

REFERENCES "LR_2".car (car_number) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id_client FOREIGN KEY (id_client)

REFERENCES "LR_2".clients (id_client) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id_contract FOREIGN KEY (contract_number)

REFERENCES "LR_2".contract (id_contract) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id_violation FOREIGN KEY (id_violation)

REFERENCES "LR_2".violations (id_violation) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT workers_full_name FOREIGN KEY (workers_full_name)

REFERENCES "LR_2".worker (full_name) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".fines

OWNER to postgres;

COMMENT ON TABLE "LR_2".fines

IS 'Штрафы';

```

CREATE TABLE IF NOT EXISTS "LR_2".insurance
(
    name_of_insurance text COLLATE pg_catalog."default" NOT NULL,
    id_contract integer NOT NULL,
    id_client integer NOT NULL,
    workers_full_name text COLLATE pg_catalog."default" NOT NULL,
    car_number integer NOT NULL,
    cost integer NOT NULL,
    type_of_insurance text COLLATE pg_catalog."default" NOT NULL,
    description text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT insurance_pkey PRIMARY KEY (name_of_insurance),
    CONSTRAINT car_number FOREIGN KEY (car_number)
        REFERENCES "LR_2".car (car_number) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT id_client FOREIGN KEY (id_client)
        REFERENCES "LR_2".clients (id_client) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT id_contract FOREIGN KEY (id_contract)
        REFERENCES "LR_2".contract (id_contract) MATCH SIMPLE
        ON UPDATE NO ACTION

```

```
ON DELETE NO ACTION  
  
NOT VALID,  
  
CONSTRAINT workers_full_name FOREIGN KEY (workers_full_name)  
  
REFERENCES "LR_2".worker (full_name) MATCH SIMPLE  
  
ON UPDATE NO ACTION  
  
ON DELETE NO ACTION  
  
NOT VALID  
  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS "LR_2".insurance
```

```
OWNER to postgres;
```

```
COMMENT ON TABLE "LR_2".insurance
```

```
IS 'Страховка';
```

```
CREATE TABLE IF NOT EXISTS "LR_2".model
```

```
(  
  
car_model text COLLATE pg_catalog."default" NOT NULL,  
  
name text COLLATE pg_catalog."default" NOT NULL,  
  
cost_per_hour integer NOT NULL,  
  
cost_per_day integer NOT NULL,  
  
characteristics text COLLATE pg_catalog."default" NOT NULL,  
  
CONSTRAINT pk_key PRIMARY KEY (car_model)
```

```
        INCLUDE(car_model)
    )

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".model

    OWNER to postgres;

COMMENT ON TABLE "LR_2".model

    IS 'Модель';

CREATE TABLE IF NOT EXISTS "LR_2".violations
(
    id_violation integer NOT NULL,
    name text COLLATE pg_catalog."default" NOT NULL,
    description text COLLATE pg_catalog."default" NOT NULL,
    cost integer NOT NULL,
    CONSTRAINT violations_pkey PRIMARY KEY (id_violation)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".violations

    OWNER to postgres;
```


COMMENT ON TABLE "LR_2".violations

IS 'Нарушения';

CREATE TABLE IF NOT EXISTS "LR_2".worker

(

full_name text COLLATE pg_catalog."default" NOT NULL,

post text COLLATE pg_catalog."default" NOT NULL,

contacts text COLLATE pg_catalog."default" NOT NULL,

id_branch integer NOT NULL,

CONSTRAINT worker_pkey PRIMARY KEY (full_name),

CONSTRAINT id_branch FOREIGN KEY (id_branch)

REFERENCES "LR_2".branch (id_branch) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LR_2".worker

OWNER to postgres;

COMMENT ON TABLE "LR_2".worker

IS 'Работник';

INSERT INTO "LR_2".accidents VALUES

(1, 2, 1, 'Иван Александрович Петров', 123, '-', 500, 'Невский пр.'),

(2, 1, 2, 'Иван Александрович Петров', 124, -, 1000, 'Невский пр.')

INSERT INTO "LR_2".acts_of_transfer VALUES

(1, 'Иван Иванович Иванов', 1, 1, 123, 1, -, 2022-11-20),
(2, 'Иван Иванович Иванов', 1, 1, 123, 2, -, 2022-11-27),
(3, 'Иван Иванович Иванов', 2, 3, 124, 1, 'Царапина водительской двери', 2022-11-22),
(4, 'Иван Иванович Иванов', 2, 3, 124, 2, 'Царапина водительской двери', 2022-11-27)

INSERT INTO "LR_2".branch VALUES

(1, 'Малая Пушкинская ул. дом', 22),
(2, 'ул. Марата дом', 6),
(3, 'пр. Стачек дом', 21),
(4, 'Лиговский пр. дом', 293)

INSERT INTO "LR_2".car VALUES

(123, 65734523, 2018-07-29, 'Polo'),
(124, 123123123, 2021-03-13, 'XC90'),
(125, 332432432, 2018-12-01, 'Polo')

INSERT INTO "LR_2".clients VALUES

(1, 'Сергей Иванович Иванов', '+793455345436', 1234567, 123523),
(2, 'Андрей Сергеевич Иванов', '+793433345464', 1234559, 134435),
(3, 'Антон Антонович Петров', '+793433345496', 5423412, 1332545)

INSERT INTO "LR_2".contract VALUES

(1, 1, 'Иван Иванович Иванов', 123, 2022-10-12, 2022-10-20, 2022-10-25, 'Обрабатывается'),
(2, 1, 'Иван Иванович Иванов', 123, 2022-10-12, 2022-10-20, 2022-10-25, 'Принято'),
(3, 2, 'Иван Иванович Иванов', 124, 2022-10-12, 2022-10-20, 2022-10-25, 'Обрабатывается')

INSERT INTO "LR_2".extension VALUES

(3, 3, 'Иван Александрович Петров', 124, 2022-06-12, 2022-06-20, 2022-06-11, 5000),
(2, 2, 'Иван Александрович Петров', 123, 2022-06-12, 2022-06-20, 2022-06-12, 5000),
(1, 1, 'Иван Александрович Петров', 123, 2022-06-12, 2022-06-20, 2022-06-12, 1000)

INSERT INTO "LR_2".fines VALUES

(1, 1, 1, 'Иван Александрович Петров', 124, 500, 1, 'Невский пр.', 2022-12-10, 1),
(2, 2, 1, 'Иван Александрович Петров', 123, 1000, 2, 'Невский пр.', 2022-12-10, 2),
(3, 1, 1, 'Иван Александрович Петров', 124, 500, 1, 'Невский пр.', 2022-12-10, 1)

INSERT INTO "LR_2".insurance VALUES

(1, 1, 1, 'Иван Александрович Петров', 123, 1200, 'От несчастных случаев', -),
(2, 2, 1, 'Иван Александрович Петров', 124, 1400, 'От несчастных случаев', -),
(3, 3, 2, 'Иван Александрович Петров', 124, 1200, 'От несчастных случаев', -)

```
INSERT INTO "LR_2".model VALUES  
(('Lada', 'BA3', 300, 1000, '80 л.с'),  
(('Ford', 'Focus', 2, 1000, 5000, '115 л.с'),  
(('Volvo', 'XC90', 1300, 7000, '235 л.с'),  
(('Volkswagen', 'Polo', 800, 4000, '110 л.с')
```

```
INSERT INTO "LR_2".violations VALUES  
(1, 'Превышение скорости', '-', 500),  
(2, 'Проезд на красный', '-', 4000),  
(3, 'Пересечение сплошной', '-', 10000),
```

```
INSERT INTO "LR_2".worker VALUES  
(('Александр Александрович Иванов', 'Директор отдела' '+798434519345', 1),  
(('Иван Иванович Иванов', 'Сотрудник', '+7984343243245', 2),  
(('Иван Александрович Петров', 'Сотрудник', '+7984343243286', 3),  
(('Александр Павлович Иванов', 'Сотрудник', '+79843452349', 2)
```

Выводы

В процессе выполнения лабораторной работы удалось ознакомиться с работой в pgAdmin 4, получить практические навыки создания таблиц, установки ограничений на таблицы, создания и восстановления резервных копий баз данных.