**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 K32392

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 1Х.
2. Создать базу данных с использованием pgadmin 4.

**Цель работы 1.2:** овладеть практическими навыками создания таблиц базы данных PostgreSQL 1Х, 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.

1. Восстановить БД.

**Выполнение**

**Наименование БД:** restaurant

**ERD диаграмма:**

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

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,

pasport 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',    'ВАЗ',  300,    1000,   '80 л.с'),

('Ford',    'Focus', 2, 1000,   5000,   '115 л.с'),

('Volvo',   'XC90', 1300,   7000,   '235 л.c'),

('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 ознакомится с работой в pgAdmin 4, получить практические навыки создания таблиц, установки ограничений на таблицы, создания и восстановления резервных копий баз данных.