

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №1.2
«Создание таблиц базы данных POSTGRESQL. Заполнение таблиц
рабочими данными»

Выполнил:
студент 2 курса ИКТ группа К3241
Траоре Мамуду

Проверила:
Говорова Марина Михайловна

Санкт-Петербург
2022

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:

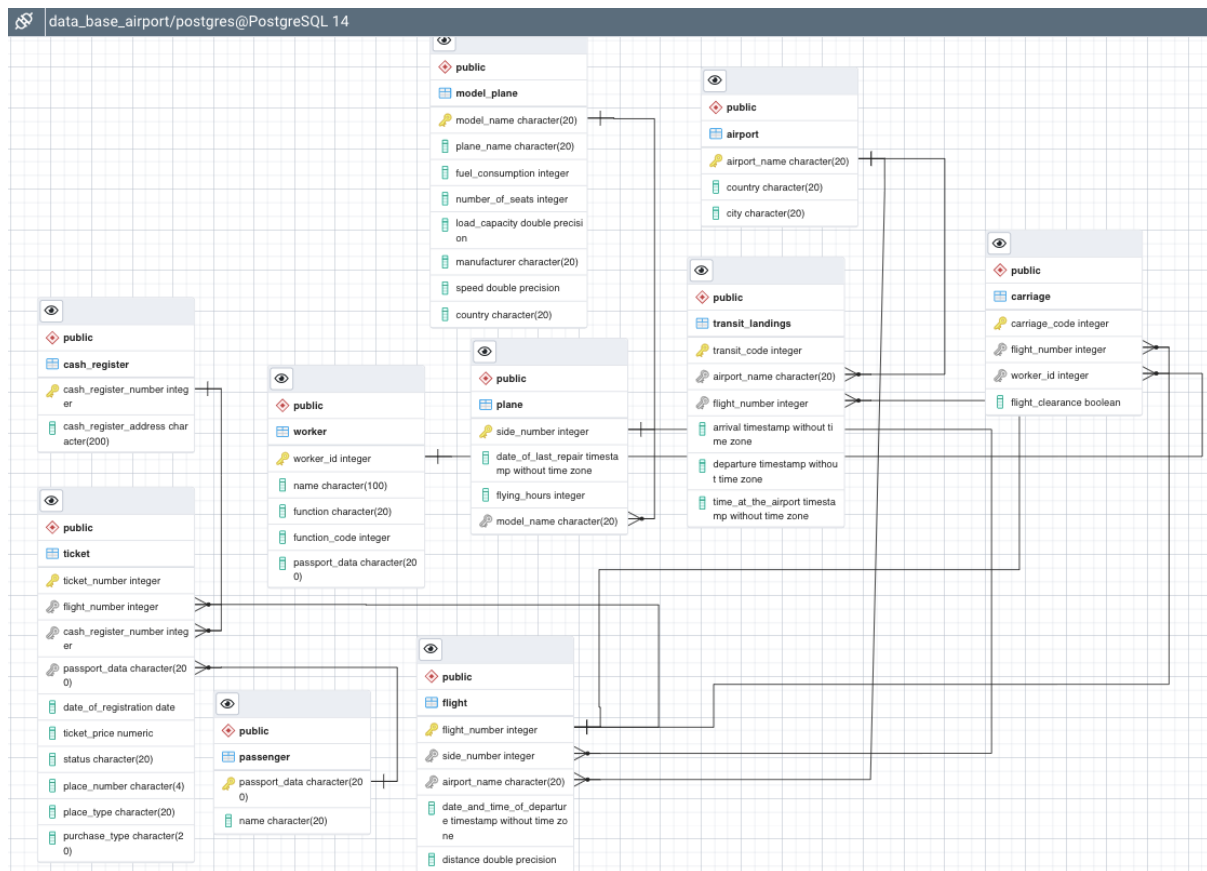
- с расширением CUSTOM для восстановления БД;
- с расширением PLAIN для листинга (в отчете);
- при создании резервных копий БД настроить параметры Dump options для Type of objects и Queries.

7. Восстановить БД.

Наименование БД

Вариант 8. БД «Аэропорт » Наименование:

Схема логической модели базы данных, сгенерированная в Generate ERD



Dump, содержащий скрипты работы с БД

1) Создание базы данных (схемы в бд):

```
CREATE SCHEMA public;  
ALTER SCHEMA public OWNER TO postgres;
```

2) Создание таблицы Самолёт и определение ограничений:

```
CREATE TABLE public.plane (  
    side_number integer NOT NULL,  
    date_of_last_repair timestamp without time zone NOT NULL,
```

```
    flying_hours integer NOT NULL,  
    model_name character(20) NOT NULL  
);
```

```
ALTER TABLE public.plane OWNER TO postgres;
```

3) Создание таблицы Аэропорты и определение ограничений:

```
CREATE TABLE public.airport (  
    airport_name character(20) NOT NULL,  
    country character(20) NOT NULL,  
    city character(20) NOT NULL  
);
```

```
ALTER TABLE public.airport OWNER TO postgres;
```

4) Создание таблицы Билеты и определение ограничений:

```
CREATE TABLE public.ticket (  
    ticket_number integer NOT NULL,  
    flight_number integer,  
    cash_register_number integer,  
    passport_data character(200),  
    date_of_registration timestamp without time zone,  
    ticket_price numeric,  
    status character(20),  
    place_number character(4),  
    place_type character(20),
```

```
purchase_type character(20)
);
```

```
ALTER TABLE public.ticket OWNER TO postgres;
```

5) Создание таблицы Экипаж и определение ограничений:

```
CREATE TABLE public.carriage (
    carriage_code integer NOT NULL,
    flight_number integer NOT NULL,
    worker_id integer NOT NULL,
    flight_clearance boolean NOT NULL
);
```

```
ALTER TABLE public.carriage OWNER TO postgres;
```

6) Создание таблицы Выполненный рейс и определение ограничений:

```
CREATE TABLE public."flight " (
    flight_number integer NOT NULL,
    side_number integer NOT NULL,
    airport_name character(20) NOT NULL,
    date_and_time_of_departure timestamp without time zone
NOT NULL,
    distance double precision NOT NULL
);
```

```
ALTER TABLE public."flight " OWNER TO postgres;
```

7) Создание таблицы Пассажир и определение ограничений:

```
CREATE TABLE public."passenger " (  
    passport_data character(200) NOT NULL,  
    name character(20) NOT NULL  
);
```

```
ALTER TABLE public."passenger " OWNER TO postgres;
```

8) Создание таблицы Сотрудники и определение ограничений:

```
CREATE TABLE public.worker (  
    worker_id integer NOT NULL,  
    name character(100) NOT NULL,  
    function character(20) NOT NULL,  
    function_code integer NOT NULL,  
    passport_data character(200)  
);
```

```
ALTER TABLE public.worker OWNER TO postgres;
```

9) Создание таблицы Кассы и определение ограничений:

```
CREATE TABLE public.cash_register (  
    cash_register_number integer NOT NULL,  
    cash_register_address character(200)  
);
```

```
ALTER TABLE public.cash_register OWNER TO postgres;
```

10) Создание таблицы Транзиты и определение ограничений:

```
CREATE TABLE public."transit_landings" (  
    transit_code integer NOT NULL,  
    airport_name character(20) NOT NULL,  
    flight_number integer NOT NULL,  
    arrival timestamp without time zone,  
    departure timestamp without time zone,  
    time_at_the_airport timestamp without time zone  
);
```

```
ALTER TABLE public."transit_landings" OWNER TO postgres;
```

11) Заполнение таблиц рабочими данными:

```
COPY public.airport (airport_name, country, city) FROM stdin;
```

conakry_airport	Guinea	Conakry
Барселона	Испания	Барселона
Оттава	Канада	Оттава
Югра	Россия	Сургут
Пулково	Россия	Санкт-Петербург
Рига	Латвия	Рига
Казань	Россия	Казань

\\.

```
COPY public.carriage (carriage_code, flight_number,
worker_id, flight_clearance) FROM stdin;
```

```
12 123 1 t
```

```
34 1234 2 f
```

```
567 234123 t
```

\\.

```
COPY public.cash_register (cash_register_number,
cash_register_address) FROM stdin;
```

```
2 вяземский
```

```
34 вяземскийи 34
```

```
1 горгоская 23
```

```
3 ломоносова 23
```

\\.

```
COPY public."flight " (flight_number, side_number,
airport_name, date_and_time_of_departure, distance) FROM stdin;
```

```
1234 1 conakry_airport 2022-01-26 00:00:00
24553
```

```
12467 2 Пулково 2022-01-05 00:00:00
23464
```

```
23412 2 Барселона 2022-01-26 00:00:00
3789
```

```
123 1 Рига 2022-01-05 00:00:00
23562
```

\\.


```
COPY public."model_plane " (model_name, plane_name,
fuel_consumption, number_of_seats, load_capacity, manufacturer,
speed, country) FROM stdin;
```

```
Boeing707      Airplane      12345550  345
Boieng         12346USA
\.
```

```
COPY public."passenger " (passport_data, name) FROM stdin;
```

```
783789026
Мальдини П.С
7503578
Кака Д.
02345619435
Пупкина С.В
123489457
Мамуду Т
\.
```

```
COPY public.plane (side_number, date_of_last_repair, flying_hours, model_name)
FROM stdin;
```

```
1      2022-02-23 00:00:00      12345Boeing707
2      2022-01-15 00:00:00      13467Boeing707
```

```
\.
```

COPY public."transit_landings " (transit_code, airport_name, flight_number, arrival, departure, time_at_the_airport) FROM stdin;

1	Рига	2	2022-01-06 00:00:00	2022-01-05 00:00:00
---	------	---	---------------------	---------------------

\\.

COPY public.ticket (ticket_number, flight_number, cash_register_number, passport_data, date_of_registration, ticket_price, status, place_number, place_type, purchase_type) FROM stdin;

1	1234	1	02345619435						
	2022-02-23 00:00:00	5500	valid		b	econom			
	online								
2	123	2	123489457						
	2021-08-21 00:00:00	5000	invalid		b	bisnes			
	on site								
3	234123		783789026						
	2019-05-15 00:00:00	3800	rejection		a	econom			
	online								

\\.

COPY public.worker (worker_id, name, function, function_code, passport_data) FROM stdin;

1	Ivan			
	pilote	2	245352	
2	Mamoudou Traore			
	Director	1	3567934	
3	Mamady Berete			
	worker	34	4537898	

\\.

12) Назначение первичных и внешних ключей, а также дополнительных ограничений:

```
ALTER TABLE ONLY public.airport
  ADD CONSTRAINT airport_pkey PRIMARY KEY (airport_name);
ALTER TABLE ONLY public.carriage
  ADD CONSTRAINT carriage_carriage_code_carriage_code1_key UNIQUE
(carriage_code) INCLUDE (carriage_code);
```

```
ALTER TABLE ONLY public.carriage
  ADD CONSTRAINT carriage_pkey PRIMARY KEY (carriage_code);
```

```
ALTER TABLE ONLY public.cash_register
  ADD CONSTRAINT cash_register_pkey PRIMARY KEY
(cash_register_number);
```

```
ALTER TABLE ONLY public."flight "
  ADD CONSTRAINT "flight_pkey" PRIMARY KEY (flight_number);
```

```
ALTER TABLE ONLY public."model_plane "
  ADD CONSTRAINT "model_plane_pkey" PRIMARY KEY (model_name);
```

```
ALTER TABLE ONLY public."passenger "
  ADD CONSTRAINT "passenger_passport_data_passport_data1_key"
UNIQUE (passport_data) INCLUDE (passport_data);
```

```
ALTER TABLE ONLY public."passenger "
  ADD CONSTRAINT "passenger_pkey" PRIMARY KEY (passport_data);
```

```
ALTER TABLE ONLY public.plane
  ADD CONSTRAINT plane_pkey PRIMARY KEY (side_number);
```

```
ALTER TABLE ONLY public.plane
```

```
ADD CONSTRAINT plane_side_number_side_number1_key UNIQUE
(side_number) INCLUDE (side_number);
```

```
ALTER TABLE public.ticket
ADD CONSTRAINT r1 CHECK (((place_type = 'bisnes'::bpchar) OR
(place_type = 'econom'::bpchar))) NOT VALID;
```

```
ALTER TABLE public.ticket
ADD CONSTRAINT r2 CHECK (((status = 'valid'::bpchar) OR (status =
'invalid'::bpchar) OR (status = 'rejection'::bpchar))) NOT VALID;
```

```
ALTER TABLE public.ticket
ADD CONSTRAINT r2 CHECK (((status = 'valid'::bpchar) OR (status =
'invalid'::bpchar) OR (status = 'rejection'::bpchar))) NOT VALID;
```

```
ALTER TABLE public.ticket
ADD CONSTRAINT r3 CHECK (((purchase_type = 'online'::bpchar) OR
(purchase_type = 'on site'::bpchar))) NOT VALID;
```

```
ALTER TABLE public.ticket
ADD CONSTRAINT ticket CHECK ((ticket_price > (0)::numeric)) NOT
VALID;
```

```
ALTER TABLE ONLY public.ticket
ADD CONSTRAINT ticket_pkey PRIMARY KEY (ticket_number);
```

```
ALTER TABLE ONLY public.ticket
ADD CONSTRAINT ticket_ticket_number_ticket_number1_key UNIQUE
(ticket_number) INCLUDE (ticket_number);
```

```
ALTER TABLE ONLY public."transit_landings "
ADD CONSTRAINT "transit_landings_pkey" PRIMARY KEY
(transit_code);
```

```
ALTER TABLE ONLY public."transit_landings "
ADD CONSTRAINT "unique" UNIQUE (transit_code) INCLUDE
(transit_code);
```

```
ALTER TABLE ONLY public.worker
  ADD CONSTRAINT worker_passport_data_passport_data1_key UNIQUE
(passport_data) INCLUDE (passport_data);
```

```
ALTER TABLE ONLY public.worker
  ADD CONSTRAINT worker_pkey PRIMARY KEY (worker_id);
```

```
ALTER TABLE ONLY public."flight "
  ADD CONSTRAINT fk_airport FOREIGN KEY (airport_name)
REFERENCES public.airport(airport_name) NOT VALID;
```

```
ALTER TABLE ONLY public."transit_landings "
  ADD CONSTRAINT fk_airport FOREIGN KEY (airport_name)
REFERENCES public.airport(airport_name) NOT VALID;
```

```
ALTER TABLE ONLY public.ticket
  ADD CONSTRAINT fk_cash_register FOREIGN KEY
(cash_register_number) REFERENCES
public.cash_register(cash_register_number) NOT VALID;
```

```
ALTER TABLE ONLY public.carriage
  ADD CONSTRAINT fk_flight FOREIGN KEY (flight_number)
REFERENCES public."flight "(flight_number) NOT VALID;
```

```
ALTER TABLE ONLY public."transit_landings "
  ADD CONSTRAINT fk_flight FOREIGN KEY (flight_number)
REFERENCES public."flight "(flight_number) NOT VALID;
```

```
ALTER TABLE ONLY public.ticket
  ADD CONSTRAINT "fk_flight " FOREIGN KEY (flight_number)
REFERENCES public."flight "(flight_number) NOT VALID;
```

```
ALTER TABLE ONLY public.plane
  ADD CONSTRAINT fk_model_plane FOREIGN KEY (model_name)
REFERENCES public."model_plane "(model_name) NOT VALID;
```

```
ALTER TABLE ONLY public.ticket  
  ADD CONSTRAINT fk_passenger FOREIGN KEY (passport_data)  
REFERENCES public."passenger "(passport_data) NOT VALID;
```

```
ALTER TABLE ONLY public."flight "  
  ADD CONSTRAINT fk_plane FOREIGN KEY (side_number)  
REFERENCES public.plane(side_number) NOT VALID;
```

```
ALTER TABLE ONLY public.carriage  
  ADD CONSTRAINT fk_worker FOREIGN KEY (worker_id)  
REFERENCES public.worker(worker_id) NOT VALID;
```

Выводы

В ходе данной лабораторной работы была создана база данных с использованием pgAdmin 4, схема в составе базы данных, таблицы базы данных, были установлены ограничения на данные, таблицы БД заполнены рабочими данными, создана резервная копия БД, восстановлена БД.