

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1.2
по теме: Создание таблиц базы данных postgresql. Заполнение
таблиц рабочими данными.
по дисциплине: Проектирование и реализация баз данных

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «__» _____ 20__ г.
Оценка _____

Выполнил:
студент группы К3241
Костенников Д.В.

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

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.

Вариант 14. БД «Служба заказа такси»

Описание предметной области: Система должна фиксировать все вызовы такси. Каждому водителю ежедневно начисляется заработная плата в зависимости от количества вызовов и их тарифа (50% от заработанной им суммы). Автомобили могут быть собственностью компании или таксиста.

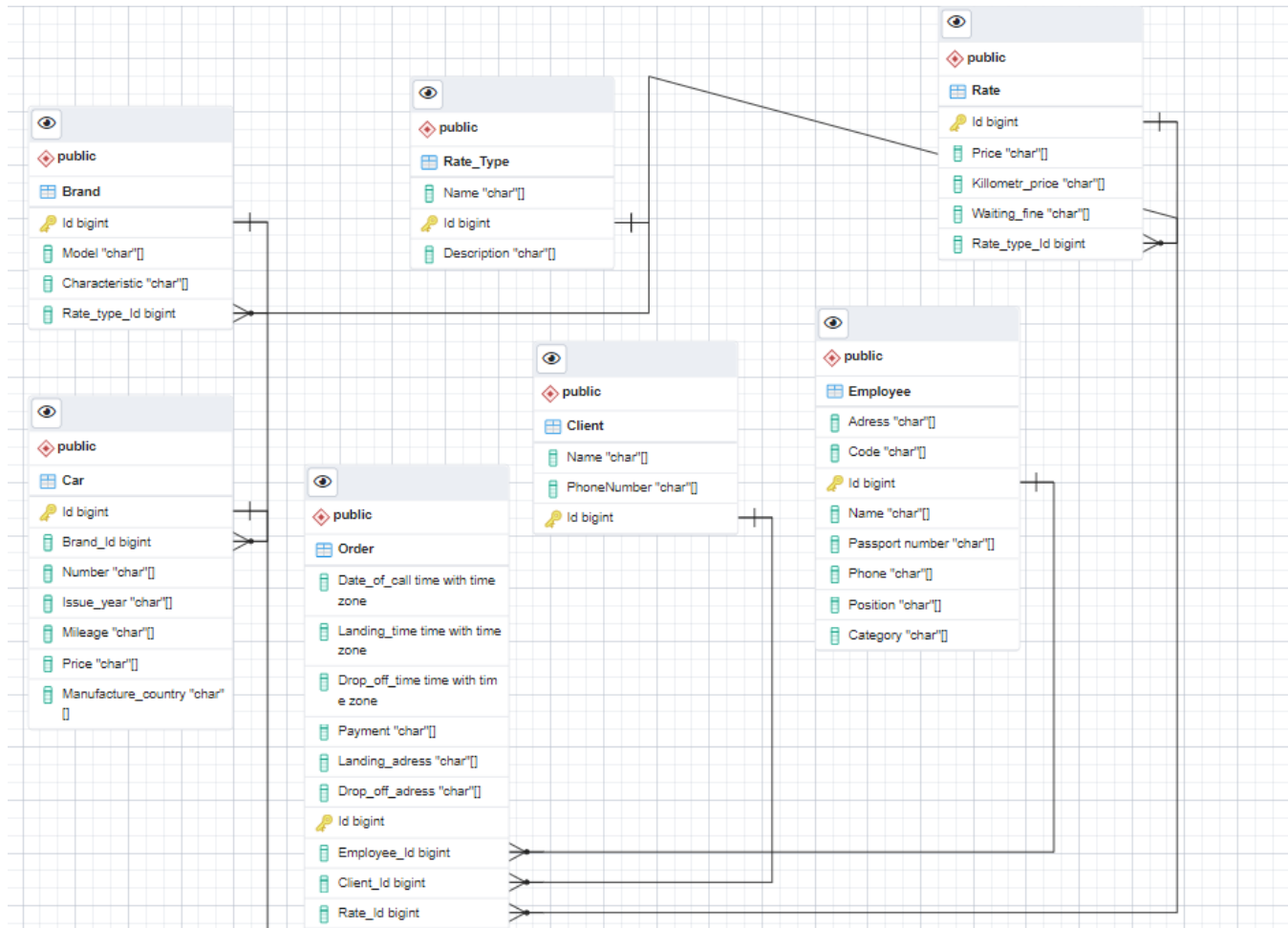
БД должна содержать следующий минимальный набор сведений: Код сотрудника. ФИО сотрудника. Адрес сотрудника. № телефона сотрудника. Паспортные данные сотрудника. Должность сотрудника. Категория сотрудника. Наименование модели и марки автомобиля. Технические характеристики. Страна-производитель. Стоимость. Код тарифа. Наименование тарифа. Цена за километр. Код автомобиля. Госномер автомобиля. Год выпуска. Пробег. Дата последнего ТО. Дата вызова. Время посадки пассажира. Время высадки пассажира. Номер телефона пассажира. Откуда. Куда. Расстояние. Штраф за время ожидания (в минутах). Оплата (онлайн (при заказе) или наличными). Рекламация клиента на вызов.

ХОД РАБОТЫ

1) Наименование БД:

Taxi

2) Схема логической модели:



3) Dump, содержащий скрипты работы с БД.

Создание базы данных:

```
-- Database: Taxi
```

```
-- DROP DATABASE "Taxi";
```

```
CREATE DATABASE "Taxi"  
WITH  
OWNER = postgres  
ENCODING = 'UTF8'  
LC_COLLATE = 'Russian_Russia.1251'  
LC_CTYPE = 'Russian_Russia.1251'  
TABLESPACE = pg_default  
CONNECTION LIMIT = -1;
```

Создание схемы:

```
-- SCHEMA: public
```

```
-- DROP SCHEMA public ;
```

```
CREATE SCHEMA public
```

```
    AUTHORIZATION postgres;
```

```
COMMENT ON SCHEMA public
```

```
    IS 'standard public schema';
```

```
GRANT ALL ON SCHEMA public TO PUBLIC;
```

```
GRANT ALL ON SCHEMA public TO postgres;
```

Создание таблицы Brand и определение ограничений:

```
CREATE TABLE public."Brand"  
(  
    "Id" bigint NOT NULL,  
    "Rate_type_Id" bigint,  
    "Model" character varying COLLATE pg_catalog."default",  
    "Characteristic" character varying COLLATE pg_catalog."default",  
    CONSTRAINT "Brand_pkey" PRIMARY KEY ("Id"),  
    CONSTRAINT "Rate_type_Id" FOREIGN KEY ("Rate_type_Id")  
        REFERENCES public."Rate_Type" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Brand"  
    OWNER to postgres;
```

Создание таблицы Car и определение ограничений:

```
CREATE TABLE public."Car"  
(  
    "Id" bigint NOT NULL,  
    "Brand_Id" bigint,  
    "Number" character varying COLLATE pg_catalog."default",  
    "Issue_year" character varying COLLATE pg_catalog."default",
```

```

"Manufacture_country" character varying COLLATE pg_catalog."default",
"Price" bigint,
"Mileage" bigint,
CONSTRAINT "Car_pkey" PRIMARY KEY ("Id"),
CONSTRAINT "Brand_Id" FOREIGN KEY ("Brand_Id")
    REFERENCES public."Brand" ("Id") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
CONSTRAINT "Price_check" CHECK ("Price" > 0) NOT VALID,
CONSTRAINT "Mileage_check" CHECK ("Mileage" > 0) NOT VALID
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE public."Car"
    OWNER to postgres;

```

Создание таблицы Client и определение ограничений:

```

CREATE TABLE public."Client"
(
    "Id" bigint NOT NULL,
    "Name" character varying COLLATE pg_catalog."default",
    "Phone_number" character varying COLLATE pg_catalog."default",
    CONSTRAINT "Client_pkey" PRIMARY KEY ("Id")
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE public."Client"
    OWNER to postgres;

```

Создание таблицы Employee и определение ограничений:

```

CREATE TABLE public."Employee"
(
    "Id" bigint NOT NULL,
    "Adress" character varying COLLATE pg_catalog."default",
    "Code" character varying COLLATE pg_catalog."default",
    "Name" character varying COLLATE pg_catalog."default",
    "Passport number" character varying COLLATE pg_catalog."default",
    "Phone" character varying COLLATE pg_catalog."default",
    "Position" character varying COLLATE pg_catalog."default",

```

```
"Category" character varying COLLATE pg_catalog."default",  
CONSTRAINT "Employee_pkey" PRIMARY KEY ("Id")  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Employee"  
OWNER to postgres;
```

Создание таблицы Order и определение ограничений:

```
CREATE TABLE public."Order"  
(  
    "Landing_time" time with time zone,  
    "Drop_off_time" time with time zone,  
    "Id" bigint NOT NULL,  
    "Employee_Id" bigint,  
    "Client_Id" bigint,  
    "Rate_Id" bigint,  
    "Car_Id" bigint,  
    "Payment" character varying COLLATE pg_catalog."default",  
    "Landing_adress" character varying COLLATE pg_catalog."default",  
    "Drop_off_adress" character varying COLLATE pg_catalog."default",  
    "Date_of_call" timestamp with time zone,  
    CONSTRAINT "Order_pkey" PRIMARY KEY ("Id"),  
    CONSTRAINT "Car_Id" FOREIGN KEY ("Car_Id")  
        REFERENCES public."Car" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID,  
    CONSTRAINT "Client_Id" FOREIGN KEY ("Client_Id")  
        REFERENCES public."Client" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID,  
    CONSTRAINT "Employee_Id" FOREIGN KEY ("Employee_Id")  
        REFERENCES public."Employee" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID,  
    CONSTRAINT "Rate_Id" FOREIGN KEY ("Rate_Id")  
        REFERENCES public."Rate" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION
```

```
ON DELETE NO ACTION
NOT VALID,
CONSTRAINT "Payment_check" CHECK ("Payment"::text = ANY
(ARRAY['Card'::character varying::text, 'Cash'::character varying::text])) NOT VALID
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Order"
OWNER to postgres;
```

Создание таблицы Rate и определение ограничений:

```
CREATE TABLE public."Rate"
(
    "Id" bigint NOT NULL,
    "Rate_type_Id" bigint,
    "Status" character varying COLLATE pg_catalog."default",
    "Killometr_price" double precision,
    "Waiting_fine" double precision,
    CONSTRAINT "Rate_pkey" PRIMARY KEY ("Id"),
    CONSTRAINT "Rate_type_Id" FOREIGN KEY ("Rate_type_Id")
        REFERENCES public."Rate_Type" ("Id") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Status_check" CHECK ("Status"::text = ANY
(ARRAY['Night'::character varying, 'Day'::character varying, 'Holiday'::character
varying, 'Weekend'::character varying]::text[])) NOT VALID,
    CONSTRAINT "Price_check" CHECK ("Killometr_price" > 0::double precision) NOT
VALID,
    CONSTRAINT "Fine_check" CHECK ("Waiting_fine" > 0::double precision) NOT
VALID
)
```

```
TABLESPACE pg_default;
```


```
ALTER TABLE public."Rate"
OWNER to postgres;
```

Создание таблицы Rate_Type и определение ограничений:

```
CREATE TABLE public."Rate_Type"
(
    "Id" bigint NOT NULL,
    "Name" character varying COLLATE pg_catalog."default",
    "Description" character varying COLLATE pg_catalog."default",
    CONSTRAINT "Rate_Type_pkey" PRIMARY KEY ("Id")
)
```

TABLESPACE pg_default;

ALTER TABLE public."Rate_Type"
OWNER to postgres;

 public.Brand/Taxi/postgres@PostgreSQL 14

Query Editor

```
1 SELECT * FROM public."Brand"
2 ORDER BY "Id" ASC
```

Notifications

Query History

Recorded time	Event
No data	

Data Output

Explain

Messages

	<div>Id</div> <div>[PK] bigint</div>	<div>Rate_type_Id</div> <div>bigint</div>	<div>Model</div> <div>character varying</div>	<div>Characteristic</div> <div>character varying</div>
1	1	1	Mercedes-E	bla
2	2	2	Kia Rio	bla bla

Рис. 2 – данные в таблице Brand

Query Editor

1

SELECT * FROM public."Car"

2

ORDER BY "Id" ASC

Notifications

Query History

Recorded time	Event	Process ID	Payload
No data found			


Data Output

Explain

Messages

<div><div></div></div> <div>Id</div> <div>[PK] bigint</div> <div></div>	<div><div></div></div> <div>Brand_Id</div> <div>bigint</div> <div></div>	<div><div></div></div> <div>Number</div> <div>character varying</div> <div></div>	<div><div></div></div> <div>Issue_year</div> <div>character varying</div> <div></div>	<div><div></div></div> <div>Mileage</div> <div>character varying</div> <div></div>	<div><div></div></div> <div>Price</div> <div>character varying</div> <div></div>	<div><div></div></div> <div>Manufacture_country</div> <div>character varying</div> <div></div>	
1	1	1	000AA000	2022	0	5000000	Germany
2	2	2	000AA001	2022	0	1000000	South Korea

Рис. 3 – заполненные данные таблицы Car

 public.Client/Taxi/postgres@PostgreSQL 14

Query Editor

1

SELECT * FROM public."Client"

2

ORDER BY "Id" ASC

Notifications

Query History

Recorded time			Event

Data Output

Explain

Messages

	<div>Id</div> <div>[PK] bigint</div>	<div>Name</div> <div>character varying</div>	<div>Phone_number</div> <div>character varying</div>
1	1	Danil	+7904000100
2	2	Ira	+7904000101

Рис. 4 – заполненные данные таблицы Client

public.Employee/Taxi/postgres@PostgreSQL 14

Query Editor


```

1 SELECT * FROM public."Employee"
2 ORDER BY "Id" ASC
      
```

Notifications
 Query History

Recorded time	Event	Process ID	Payload
No data found			

Data Output
 Explain
 Messages

	Id [PK] bigint	Adress character varying	Code character varying	Name character varying	Passport number character varying	Phone character varying	Position character varying	Category character
1	1	Spb	000	Danil	0001	+7904000000	Driver	B
2	2	Spb	001	Dima	0002	+7904000002	Driver	B

Рис. 5 – заполненные данные таблицы Employee

public.Rate_Type/Taxi/postgres@PostgreSQL 14

Query Editor


```

1 SELECT * FROM public."Rate_Type"
2 ORDER BY "Id" ASC
      
```

Notifications
 Query History

Recorded time	Event
---------------	-------

Data Output
 Explain
 Messages

	Id [PK] bigint	Name character varying	Description character varying
1	1	Business	bla
2	2	Econom	blabla

Рис. 6 – заполненные данные таблицы Rate_Type

Query Editor

```

1  SELECT * FROM public."Rate"
2  ORDER BY "Id" ASC

```

Notifications

Query History

Recorded time	Event	Process ID
No data found		

Data Output

Explain

Messages

	Id [PK] bigint	Rate_type_Id bigint	Status character varying	Killometr_price double precision	Waiting_fine double precision
1	1	1	Night	25	20
2	2	2	Day	10	10

Рис. 7 – заполненные данные таблицы Rate

Query Editor

```

1  SELECT * FROM public."Order"
2  ORDER BY "Id" ASC

```

Notifications

Query History

Recorded time	Event	Process ID	Payload
No data found			


Data Output

Explain

Messages

	Landing_time time with time zone	Drop_off_time time with time zone	Id [PK] bigint	Employee_Id bigint	Client_Id bigint	Rate_Id bigint	Car_Id bigint	Payment character varying	Landing_adress character varying
1	04:05:06+02:00	04:35:07+02:00	1	1	1	1	1	Card	Spb, Alp str
2	06:05:06+02:00	06:25:07+02:00	2	2	1	2	2	Card	Spb, lomo str 9

Рис. 8 – заполненные данные таблицы Order, часть 1

public.Order/Taxi/postgres@PostgreSQL 14

Query Editor

1

SELECT * FROM public."Order"

2

ORDER BY "Id" ASC

Notifications

Query History

Recorded time	Event	Process ID	Payload
No data found			

Data Output

Explain

Messages

bigint	Employee_Id bigint	Client_Id bigint	Rate_Id bigint	Car_Id bigint	Payment character varying	Landing_adress character varying	Drop_off_adress character varying	Date_of_call timestamp with time zone
1	1	1	1	1	Card	Spb, Alp str	Spb, lomo str 9	2022-03-16 00:00:00+03
2	2	1	2	2	Card	Spb, lomo str 9	Spb, Alp str	2022-03-16 00:00:00+03

Рис 9. – заполненные данные таблицы Order, часть 2

Вывод:

В ходе выполнения работы была создана база данных в PostgreSQL; логическая схема в ее составе; созданы таблицы и заданы ограничения на

данные: Primary Key, Unique, Check, Foreign Key; в базу данных были внесены рабочие данные; также созданы две резервные копии и произведено восстановление базы данных.