

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Базы данных»**  
**Тема: Реализация базы данных в СУБД PostgreSQL**

Студент гр. 1304

Кардаш Я.Е.

Преподаватель

Заславский М.М.

Санкт-Петербург

2023

## **Цель работы.**

Изучение принципов работы с СУБД PostgreSQL. Реализация созданной структуры базы данных и выполнение запросов к ней.

## **Задание.**

### **Вариант 9.**

Пусть требуется создать программную систему, предназначенную для диспетчера автобусного парка. Такая система должна обеспечивать хранение сведений о водителях, о маршрутах и характеристиках автобусов. Каждый водитель характеризуется паспортными данными, классом, стажем работы и окладом, причем оклад зависит от класса и стажа работы. Маршрут автобуса характеризуется номером маршрута, названием начального и конечного пункта движения, временем начала и конца движения, интервалом движения и протяженностью в минутах (время движения от кольца до кольца). Характеристиками автобуса являются: номер государственной регистрации автобуса, его тип и вместимость, причем вместимость автобуса зависит от его типа. Каждый водитель закреплен за отдельным автобусом и работает на определенном маршруте, но в случае поломки своего автобуса или болезни другого водителя может пересесть на другую машину. В базе должен храниться график работы водителей. Необходимо предусмотреть возможность корректировки БД в случаях поступления на работу нового водителя, списания старого автобуса, введения нового маршрута или изменения старого и т.п. Диспетчеру автопарка могут потребоваться следующие сведения:

- Список водителей, работающих на определенном маршруте с указанием графика их работы?
- Какие автобусы обслуживают данный маршрут?
- Какие маршруты начинаются или заканчиваются в пункте с заданным названием?
- Когда начинается и заканчивается движение автобусов на каждом маршруте?
- Какова протяженность определенного маршрута?
- Какова общая протяженность маршрутов, обслуживаемых автопарком?
- Какие автобусы не вышли на линию, и по какой причине (неисправность, отсутствие водителя)?

## Выполнение работы.

Для выполнения работы установлена СУБД Postgresql, версия указана на рисунке 1.

```
yar@yar-VirtualBox:~$ pg_config --version
PostgreSQL 12.16 (Ubuntu 12.16-0ubuntu0.20.04.1)
```

Рисунок 1. Версия Postgresql

Затем в созданной базе данных dispatcher были созданы таблицы Таблицы и их структура соответствует структуре базы данных из лабораторной работы 1. Структура бд на рисунке 2. Структура таблиц на рисунках 3 – 9.

```
dispatcher=# \dt
               List of relations
 Schema |   Name   | Type  | Owner
-----+-----+-----+-----
 public | bus      | table | postgres
 public | driver   | table | postgres
 public | route    | table | postgres
 public | salary   | table | postgres
 public | status   | table | postgres
 public | trip     | table | postgres
 public | type_bus | table | postgres
(7 rows)
```

Рисунок 2. Структура бд

```
dispatcher=# \d bus
               Table "public.bus"
   Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----
 registry_number | character varying(12) |           | not null |
 status_id      | integer          |           |          |
 passport_date  | character varying(11) |           |          |
 route_number   | character varying(10) |           |          |
 type           | character varying(20) |           |          |
Indexes:
    "bus_pkey" PRIMARY KEY, btree (registry_number)
Foreign-key constraints:
    "bus_driver" FOREIGN KEY (route_number) REFERENCES route(route_number)
    "bus_status" FOREIGN KEY (status_id) REFERENCES status(status_id)
    "bus_type" FOREIGN KEY (type) REFERENCES type_bus(type)
Referenced by:
    TABLE "driver" CONSTRAINT "driver_bus" FOREIGN KEY (registry_number) REFERENCES bus(registry_number)
```

Рисунок 3. Структура таблицы bus

```
dispatcher=# \d driver
Table "public.driver"
  Column          |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
passport_date     | character varying(10) |           | not null |
registry_number   | character varying(10) |           |          |
route_number      | character varying(5)  |           |          |
class             | character varying(10) |           |          |
experience        | numeric               |           |          |
Indexes:
    "driver_pkey" PRIMARY KEY, btree (passport_date)
Foreign-key constraints:
    "driver_bus" FOREIGN KEY (registry_number) REFERENCES bus(registry_number)
    "driver_route" FOREIGN KEY (route_number) REFERENCES route(route_number)
    "driver_salary" FOREIGN KEY (class, experience) REFERENCES salary(class, experience)
```

Рисунок 4. Структура таблицы driver

```
dispatcher=# \d route
Table "public.route"
  Column          |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
route_number      | character varying(10) |           | not null |
start_point       | character varying(20) |           |          |
end_point         | character varying(20) |           |          |
time_start_point  | character varying(20) |           |          |
time_end_point    | character varying(20) |           |          |
movement_interval | integer               |           |          |
length           | integer               |           |          |
Indexes:
    "route_pkey" PRIMARY KEY, btree (route_number)
Referenced by:
    TABLE "bus" CONSTRAINT "bus_driver" FOREIGN KEY (route_number) REFERENCES route(route_number)
    TABLE "driver" CONSTRAINT "driver_route" FOREIGN KEY (route_number) REFERENCES route(route_number)
```

Рисунок 5. Структура route

```
dispatcher=# \d salary
Table "public.salary"
  Column          |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
class            | character varying(5)  |           | not null |
experience       | numeric               |           | not null |
salary          | numeric               |           |          |
Indexes:
    "salary_pkey" PRIMARY KEY, btree (class, experience)
Referenced by:
    TABLE "driver" CONSTRAINT "driver_salary" FOREIGN KEY (class, experience) REFERENCES salary(class, experience)
```

Рисунок 6. Структура salary

```
dispatcher=# \d type_bus
Table "public.type_bus"
  Column          |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
type             | character varying(20) |           | not null |
capacity         | integer               |           |          |
Indexes:
    "type_bus_pkey" PRIMARY KEY, btree (type)
Referenced by:
    TABLE "bus" CONSTRAINT "bus_type" FOREIGN KEY (type) REFERENCES type_bus(type)
```

Рисунок 7. Структура type\_bus

```
dispatcher=# \d status
          Table "public.status"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 status_id    | integer        |           | not null |
 is_properly  | boolean        |           |          |
 fail_reason  | character varying(20) |           |          |
Indexes:
    "status_pkey" PRIMARY KEY, btree (status_id)
Referenced by:
    TABLE "bus" CONSTRAINT "bus_status" FOREIGN KEY (status_id) REFERENCES status(status_id)
```

Рисунок 8. Структура status

```
dispatcher=# \d trip
          Table "public.trip"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 passport_date | character varying(11) |           | not null |
 registry_number | character varying(12) |           | not null |
 route_number  | character varying(10) |           | not null |
 date          | date              |           | not null |
 time          | time without time zone |           | not null |
Indexes:
    "trip_pkey" PRIMARY KEY, btree (passport_date, registry_number, route_number, date, "time")
```

Рисунок 9. Структура trip

После таблицы были заполнены данными, приближенными к реальным.

Содержимое таблиц на рисунках 10 - 16

```
dispatcher=# select * from bus;
 registry_number | status_id | passport_date | route_number | type
-----+-----+-----+-----+-----
 a778bk198      |          0 | 1123987643   | 222          | mini
 X859B018      |          0 | 7610123467   | 234          | city
 E691KM35      |          0 | 4361590852   | 156          | intercity
 K936CX76      |          0 | 4811829613   | 181          | city
 C645XM75      |          1 | 4721888851   | 222          | mini
 C369CX40      |          0 | 4733810936   | 222          | intercity
 A522EK87      |          0 | 4735505207   | 181          | intercity
 A122TM60      |          2 |              | 156          | mini
(8 rows)
```

Рисунок 10. Содержимое bus

```
dispatcher=# select * from driver ;
```

passport_date	registry_number	route_number	class	experience
1123987643	a778bk198	222	B	1
7610123467	X859B018	234	C	10
4361590852	E691KM35	156	D	21
4811829613	K936CX76	181	B	12
4721888851	C645XM75	222	E	7
4733810936	C369CX40	222	C	9
4735505207	A522EK87	181	C	31

(7 rows)

Рисунок 11. Содержимое driver

```
dispatcher=# select * from route;
```

route_number	start_point	end_point	time_start_point	time_end_point	movement_interval	length
222	Cheremushki	Lesistoe	6:00	22:00	60	30
234	Mokroe	Bor	8:00	23:00	120	60
156	Lesistoe	Bor	6:40	22:40	90	60
181	railway station	Lesistoe	4:00	21:00	40	20

(4 rows)

Рисунок 12. Содержимое route

```
dispatcher=# select * from salary
dispatcher=# ;
```

class	experience	salary
B	1	30000
C	10	37000
D	21	40000
B	12	32500
E	7	39000
C	9	36500
C	31	42000

(7 rows)

Рисунок 13. Содержимое salary

```
dispatcher=# select * from trip;
```

passport_date	registry_number	route_number	date	time
1123987643	a778bk198	222	2023-08-10	06:00:00
1123987643	a778bk198	222	2023-08-10	07:00:00
4733810936	C369CX40	222	2023-08-10	14:00:00
7610123467	X859B018	234	2023-08-10	08:00:00
4361590852	E691KM35	156	2023-08-10	06:40:00
4361590852	E691KM35	156	2023-08-10	08:10:00
4811829613	K936CX76	181	2023-08-10	04:00:00
4735505207	A522EK87	181	2023-08-10	04:40:00
4733810936	C369CX40	222	2023-08-10	20:00:00
4811829613	K936CX76	181	2023-08-10	06:00:00

Рисунок 14. Содержимое trip

```
dispatcher=# select * from type_bus ;
  type      | capacity
-----+-----
city        |      40
intercity   |      80
mini        |      25
(3 rows)
```

Рисунок 15. Содержимое type\_bus

```
dispatcher=# select * from status ;
status_id | is_properly | fail_reason
-----+-----+-----
          0 | t           | bus work correct
          1 | f           | bus breakdown
          2 | f           | no driver
(3 rows)
```

Рисунок 16. Содержимое status

После выполнены запросы из лабораторной работы 1. Запросы и результат на рисунках 17-23.

```
dispatcher=# Select driver.route_number, passport_date, time_start_point, time_end_point
dispatcher=# From driver Join route On driver.route_number = route.route_number
dispatcher=# Order By driver.route_number;
route_number | passport_date | time_start_point | time_end_point
-----+-----+-----+-----
156          | 4361590852   | 6:40             | 22:40
181          | 4811829613   | 4:00             | 21:00
181          | 4735505207   | 4:00             | 21:00
222          | 1123987643   | 6:00             | 22:00
222          | 4721888851   | 6:00             | 22:00
222          | 4733810936   | 6:00             | 22:00
234          | 7610123467   | 8:00             | 23:00
(7 rows)
```

Рисунок 17. Водители на маршрутах с временем работы

```
dispatcher=# SELECT registry_number, route_number
dispatcher=# FROM bus
dispatcher=# ORDER BY route_number;
registry_number | route_number
-----+-----
A122TM60       | 156
E691KM35       | 156
A522EK87       | 181
K936CX76       | 181
C645XM75       | 222
a778bk198      | 222
C369CX40       | 222
X859B018       | 234
(8 rows)
```



Рисунок 18. Автобусы на маршрутах

```
dispatcher=# SELECT route_number,start_point,end_point
dispatcher=# FROM route
dispatcher=# WHERE start_point ='Lesistoe' OR end_point ='Lesistoe'
dispatcher=# GROUP BY route_number,start_point,end_point;
```

route_number	start_point	end_point
156	Lesistoe	Bor
181	railway station	Lesistoe
222	Cheremushki	Lesistoe

(3 rows)

Рисунок 19. Маршруты где определенный стартовый/конечный пункт (в примере Лесистое)

```
dispatcher=# SELECT route_number, time_start_point, time_end_point
dispatcher=# FROM route
dispatcher=# GROUP BY route_number;
```

route_number	time_start_point	time_end_point
181	4:00	21:00
156	6:40	22:40
222	6:00	22:00
234	8:00	23:00

(4 rows)

Рисунок 20. Время начала и конца движения на маршрутах

```
dispatcher=# SELECT route_number,length
dispatcher=# FROM route
dispatcher=# GROUP BY route_number;
```

route_number	length
181	20
156	60
222	30
234	60

(4 rows)

Рисунок 21. Протяженность маршрутов

```
dispatcher=# SELECT sum(length) AS Общая_протяженность
dispatcher=# FROM route;
```

Общая_протяженность
170

(1 row)



Рисунок 22. Общая протяженность маршрутов

```
dispatcher=# Select registry_number as Невышедшие_автобусы, fail_reason, bus.status_id
dispatcher=# From bus Join status On bus.status_id = status.status_id
dispatcher=# Where bus.status_id <> 0;
Невышедшие_автобусы | fail_reason | status_id
-----+-----+-----
C645XM75           | bus breakdown |          1
A122TM60           | no driver    |          2
(2 rows)
```

Рисунок 23. Невышедшие автобусы с причиной

Исходный код находится в приложении А, файле lab2.sql, гитхаб репозитории и на сайте [www.db-fiddle.com](http://www.db-fiddle.com).

### **Выводы.**

Изучены принципы работы с Postgresql, реализована база данных, придуманная в ходе лабораторной работы 1. Выполнены запросы к созданной базе данных в соответствии с заданием из варианта.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД И ССЫЛКИ НА РАБОТУ

#### Lab2.sql

```
CREATE TABLE driver(passport_date VARCHAR(11), registry_number
VARCHAR(12), route_number VARCHAR(5), class VARCHAR(5), experience
DECIMAL);
INSERT INTO driver VALUES ('1123987643', 'a778bk198', '222','B',1);
INSERT INTO driver VALUES
('7610123467','X859BO18','234','C',10),('4361590852','E691KM35','156','D',
21),('4811829613','K936CX76','181','B',12),('4721888851','C645XM75','222',
'E',7),('4733810936','C369CX40','222','C',9),('4735505207','A522EK87','181',
'C',31);

CREATE TABLE salary (class VARCHAR(5), experience DECIMAL, salary
DECIMAL);
INSERT INTO salary VALUES
('B',1,30000),('C',10,37000),('D',21,40000),('B',12,32500),('E',7,39000),
('C',9,36500),('C',31,42000);

CREATE TABLE type_bus(type VARCHAR(20),capacity INT);
INSERT INTO type_bus VALUES
('city',40),('intercity',80),('mini',25);

CREATE TABLE route( route_number VARCHAR(10), start_point
VARCHAR(20), end_point VARCHAR(20), time_start_point VARCHAR(20),
time_end_point VARCHAR(20), movement_interval INT, length INT);
INSERT INTO route (route_number, start_point,end_point,
time_start_point, time_end_point, movement_interval, length) VALUES
('222', 'Cheremushki','Lesistoe','6:00', '22:00', 60,30), ('234','Mokroe',
'Bor', '8:00', '23:00', 120, 60), ('156','Lesistoe', 'Bor',
'6:40','22:40',90,60), ('181', 'railway station', 'Lesistoe', '4:00',
'21:00', 40, 20);

CREATE TABLE bus (registry_number VARCHAR(12), status_id INT,
passport_date VARCHAR(11), route_number VARCHAR(10), type VARCHAR(20));
INSERT INTO bus(registry_number, status_id, passport_date,
route_number, type) VALUES ('a778bk198',0,'1123987643', '222','mini'),
('X859BO18',0,'7610123467','234','city'), ('E691KM35',0,'4361590852',
'156','intercity'), ('K936CX76',0,'4811829613','181','city'),
('C645XM75',1,'4721888851','222', 'mini'),
('C369CX40',0,'4733810936','222','intercity'),
('A522EK87',0,'4735505207','181','intercity'),
('A122TM60',2,'','156','mini');

CREATE TABLE status(status_id INT NOT NULL, is_properly BOOLEAN,
fail_reason VARCHAR(20));
INSERT INTO status (status_id, is_properly, fail_reason) VALUES
(0,true,'bus work correct'), (1, false,'bus breakdown'), (2,false,'no
driver');

CREATE TABLE trip(passport_date VARCHAR(11), registry_number
VARCHAR(12), route_number VARCHAR(10), date DATE,time TIME);
INSERT INTO trip (passport_date, registry_number, route_number,
date,time) VALUES ('1123987643', 'a778bk198', '222', '2023-08-
10','06:00'),('1123987643', 'a778bk198', '222', '2023-08-10','07:00'),
```

```
( '4733810936', 'C369CX40', '222', '2023-08-10', '14:00'),
( '7610123467', 'X859BO18', '234', '2023-08-
10', '08:00'), ( '4361590852', 'E691KM35', '156', '2023-08-
10', '06:40'), ( '4361590852', 'E691KM35', '156', '2023-08-
10', '08:10'), ( '4811829613', 'K936CX76', '181', '2023-08-
10', '04:00'), ( '4735505207', 'A522EK87', '181', '2023-08-
10', '04:40'), ( '4733810936', 'C369CX40', '222', '2023-08-
10', '20:00'), ( '4811829613', 'K936CX76', '181', '2023-08-10', '06:00');
```

```
ALTER TABLE driver ADD PRIMARY KEY (passport_date);
ALTER TABLE salary ADD PRIMARY KEY (class,experience);
ALTER TABLE type_bus ADD PRIMARY KEY (type);
ALTER TABLE route ADD PRIMARY KEY (route_number);
ALTER TABLE bus ADD PRIMARY KEY (registry_number);
ALTER TABLE status ADD PRIMARY KEY (status_id);
ALTER TABLE trip ADD PRIMARY KEY
(passport_date,registry_number,route_number,date,time);
ALTER TABLE driver ADD CONSTRAINT driver_bus FOREIGN KEY
(registry_number) REFERENCES bus (registry_number);
ALTER TABLE driver ADD CONSTRAINT driver_route FOREIGN KEY
(route_number) REFERENCES route (route_number);
ALTER TABLE driver ADD CONSTRAINT driver_salary FOREIGN KEY
(class,experience) REFERENCES salary (class,experience);
ALTER TABLE bus ADD CONSTRAINT bus_driver FOREIGN KEY (route_number)
REFERENCES route (route_number);
ALTER TABLE bus ADD CONSTRAINT bus_status FOREIGN KEY (status_id)
REFERENCES status (status_id);
ALTER TABLE bus ADD CONSTRAINT bus_type FOREIGN KEY (type)
REFERENCES type_bus (type);
```

```
Select driver.route_number, passport_date, time_start_point,
time_end_point
From driver Join route On driver.route_number = route.route_number
Order By driver.route_number;
```

```
SELECT registry_number, route_number
FROM bus
ORDER BY route_number;
```

```
SELECT route_number,start_point,end_point
FROM route
WHERE start_point ='Lesistoe' OR end_point ='Lesistoe'
GROUP BY route_number,start_point,end_point;
```

```
SELECT route_number, time_start_point, time_end_point
FROM route
GROUP BY route_number;
```

```
SELECT route_number,length  
FROM route  
GROUP BY route_number;
```

```
SELECT sum(length) AS Общая_протяженность  
FROM route;
```

```
Select    registry_number    as    Невышедшие_автобусы,    fail_reason,  
bus.status_id  
From bus Join status On bus.status_id = status.status_id  
Where bus.status_id <> 0
```

Ссылка на [www.db-fiddle.com](http://www.db-fiddle.com)

<https://www.db-fiddle.com/f/6tGkdxZhfsz4nYP39kuvhr/4>

Ссылка на pull request

<https://github.com/moevm/sql-2023-1304/pull/33>