

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет ИТМО»  
Факультет инфокоммуникационных технологий

Лабораторная работа №2  
«Запросы на выборку и модификацию данных, представления и  
индексы в PostgreSQL»

По дисциплине  
«Проектирование и реализация баз данных»

Выполнил:  
Баландин И.О.  
Группа:  
К3240  
Преподаватель:  
Говорова М.М.



Санкт-Петербург  
2022 г

## ЦЕЛЬ РАБОТЫ

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Практическое задание (10 вариант. Автовокзал)

### Задание 1. Создайте запросы:

- Вывести фамилии водителей и номера автобусов, отправившиеся в рейсы до 12 часов текущего дня.
- Рассчитать выручку от продажи билетов за прошедший день.
- Вывести список водителей, которые не выполнили ни одного рейса за прошедший день.
- Вывести сумму убытков из-за непроданных мест в автобусе за прошедшую неделю.
- Сколько рейсов выполнил каждый водитель за последний месяц.
- Вывести тип автобуса, который используется на всех рейсах.
- Вывести данные водителя, который провел максимальное время в пути за прошедшую неделю.

### Задание 2. Создайте представления:

- количество свободных мест на все рейсы на завтра;
- самый популярный маршрут этой зимой.

### Задание 3.

Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

### Задание 4.

Изучить графическое представление запросов и просмотреть историю запросов

### Задание 5.

Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

## Запросы к базе данных

- 1) -- Вывести фамилии водителей и номера автобусов, отправившиеся в рейсы до 12 часов текущего дня.

```
select Номер_автобуса, ФИО_водителя
from
(
select Рейс.Номер_автобуса, ФИО_водителя, Маршрут.Время_отправления,
Рейс.Дата_отправления
from Рейс inner join Водитель on Рейс.Номер_водителя = Водитель.Номер_водителя
inner join Автобус on Рейс.Номер_автобуса = Автобус.Номер_автобуса
inner join Маршрут on Рейс.Номер_маршрута = Маршрут.Номер_маршрута
)
as Списочек
where (Дата_отправления = '2022-06-01' and Время_отправления < '12:00:00' )
```

[Query Editor](#) [Query History](#)

```
1 select Номер_автобуса, ФИО_водителя
2 from
3 (
4 select Рейс.Номер_автобуса, ФИО_водителя, Маршрут.Время_отправления, Рейс.Дата_отправления
5 from Рейс inner join Водитель on Рейс.Номер_водителя = Водитель.Номер_водителя
6 inner join Автобус on Рейс.Номер_автобуса = Автобус.Номер_автобуса
7 inner join Маршрут on Рейс.Номер_маршрута = Маршрут.Номер_маршрута
8 )
9 as Списочек
10 where (Дата_отправления = '2022-06-01' and Время_отправления < '12:00:00' )
11
12
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	Номер_автобуса character varying	ФИО_водителя character varying
1	e001kx	Ivanov Ivan Ivanovich
2	x777xx	Petrov Petr Petrovich

- 2) -- Вывести список водителей, которые не выполнили ни одного рейса за прошедший день.

```
select ФИО_водителя from Водитель
where
(
Водитель.Номер_водителя not in
(
select Номер_водителя
from Рейс
where Дата_отправления = 'today'
)
)
```

Query Editor		Query History
1	<b>select</b> ФИО_водителя <b>from</b> Водитель	
2	<b>where</b>	
3	(	
4	Водитель.Номер_водителя <b>not in</b>	
5	(	
6	select Номер_водителя	
7	<b>from</b> Рейс	
8	<b>where</b> Дата_отправления = 'today'	
9	)	
10	)	
11		

Data Output		Explain	Messages	Notifications
	ФИО_водителя character varying			
1	Ivanov Ivan Ivanovich			
2	Petrov Petr Petrovich			
3	Alekseev Aleksey Alekseevich			
4	Ульянов Владимир Ильич			

3) -- Рассчитать выручку от продажи билетов за прошедший день.

`select sum(Цена_билета) From Билет where(Дата_и_время_покупки > 'yesterday')`

Query Editor		Query History
1	<b>select</b> sum(Цена_билета) <b>From</b> Билет <b>where</b> (Дата_и_время_покупки > 'yesterday')	

Data Output		Explain	Messages	Notifications
	sum bigint			
1	[null]			

4) -- Сколько рейсов выполнил каждый водитель за последний месяц

```
select ФИО_водителя, count(*) Номер_рейса
from
(
select ФИО_водителя, Номер_рейса, Дата_отправления
from Рейс inner join Водитель
on Рейс.Номер_водителя = Водитель.Номер_водителя
)
as Список
where (Дата_отправления > 'today'::date - '1 month'::interval)
group by ФИО_водителя
```

Query Editor

Query History

```
1 select ФИО_водителя, count(*) Номер_рейса
2 from
3 (
4 select ФИО_водителя, Номер_рейса, Дата_отправления
5 from Рейс inner join Водитель
6 on Рейс.Номер_водителя = Водитель.Номер_водителя
7 )
8 as Список
9 where (Дата_отправления > 'today'::date - '1 month'::interval)
10 group by ФИО_водителя
11
```

Data Output

Explain

Messages

Notifications

	ФИО_водителя character varying	Номер_рейса bigint	
1	Alekseev Aleksey Alekseevich	1	
2	Ivanov Ivan Ivanovich	1	
3	Petrov Petr Petrocivh	3	

5) -- Вывести тип автобуса, который используется на всех рейсах.

```
select Модель_автобуса
from Автобус
where
(
    Номер_автобуса in
    (
        select Номер_автобуса
        from
        (
            select Номер_автобуса, count(Номер_маршрута) as Количество
            from Рейс
            group by Номер_автобуса
        )
        as Списочек
        where (Списочек.Количество = (select count(distinct Номер_маршрута)
from Рейс))
    )
)
```

Query Editor Query History

```

1 select Модель_автобуса
2 from Автобус
3 where
4 (
5     Номер_автобуса in
6     (
7         select Номер_автобуса
8         from
9         (
10            select Номер_автобуса, count(Номер_маршрута) as Количество
11            from Рейс
12            group by Номер_автобуса
13        )
14        as Списочек
15        where (Списочек.Количество = (select count(distinct Номер_маршрута) from Рейс))
16    )
17 )

```

Data Output Explain Messages Notifications

	Модель_автобуса	
	character varying (25)	
1	Sprinter	

- 6) Вывести данные водителя, который провел максимальное время в пути за прошедшую неделю.

```

select ФИО_водителя, sum(Время_в_пути) as Время_в_пути
from
(
    select ФИО_водителя, Номер_рейса, (extract(epoch FROM
cast((Дата_прибытия + Время_прибытия) AS timestamp) - cast((Дата_отправления +
Время_отправления) AS timestamp))/60) as Время_в_пути from Рейс
    inner join Водитель on Водитель.Номер_водителя = Рейс.Номер_водителя
    inner join Маршрут on Маршрут.Номер_маршрута = Рейс.Номер_маршрута
    where (Дата_отправления > '2022-06-01')
)
as Список
group by ФИО_водителя
order by Время_в_пути DESC
limit 1

```

Query Editor Query History

```

1 select ФИО_водителя, sum(Время_в_пути) as Время_в_пути
2 from
3 (
4     select ФИО_водителя, Номер_рейса, (extract(epoch FROM cast((Дата_прибытия +
5     inner join Водитель on Водитель.Номер_водителя = Рейс.Номер_водителя
6     inner join Маршрут on Маршрут.Номер_маршрута = Рейс.Номер_маршрута
7     where (Дата_отправления > '2022-06-01')
8 )
9 as Список
10 group by ФИО_водителя
11 order by Время_в_пути DESC
12 limit 1
13

```

Data Output Explain Messages Notifications

	ФИО_водителя	Время_в_пути
	character varying	double precision
1	Petrov Petr Petrovich	600

7) Вывести сумму убытков из-за непроданных мест за прошедшую неделю

```
select sum ((Количество_мест - Купленные_билеты) * Цена_билета) as Потери
from
(
select count(Цена_билета) as Купленные_билеты, Цена_билета, Номер_автобуса,
Количество_мест
from
(
select Билет.Номер_билета, Цена_билета, Автобус.Номер_автобуса,
Модель_автобуса from Рейс
inner join Билет on Билет.Номер_рейса = Рейс.Номер_рейса
inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
)
as Табличка
inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса =
Табличка.Модель_автобуса
group by Номер_автобуса, Количество_мест, Цена_билета
)
as foo
```

[Query Editor](#) [Query History](#)

```
1 select sum ((Количество_мест - Купленные_билеты) * Цена_билета) as Потери
2 from
3 (
4 select count(Цена_билета) as Купленные_билеты, Цена_билета, Номер_автобуса, Количество_мест
5 from
6 (
7     select Билет.Номер_билета, Цена_билета, Автобус.Номер_автобуса, Модель_автобуса from Рейс
8     inner join Билет on Билет.Номер_рейса = Рейс.Номер_рейса
9     inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
10 )
11 as Табличка
12 inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса = Табличка.Модель_автобуса
13 group by Номер_автобуса, Количество_мест, Цена_билета
14 )
15 as foo
16
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

	Потери numeric
1	834000

## Представления

1)

```
Create view Места_на_завтра as
(
select sum(Количество_мест - Билеты)
from
(
select Количество_мест, count(Таблица.Номер_билета) as Билеты
from
(
select Рейс.Номер_рейса, Рейс.Номер_автобуса, Автобус.Модель_автобуса,
Билет.Номер_билета from Рейс
inner join Билет on Рейс.Номер_рейса = Билет.Номер_рейса
inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
where (Рейс.Дата_отправления = 'tomorrow'::date)
)
as Таблица
inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса =
Таблица.Модель_автобуса
group by Количество_мест
)
as Список
)
```

Query Editor		Query History
1	<b>select</b> * <b>from</b> Места_на_завтра	
Data Output		Explain   Messages   Notifications
	<b>sum</b> numeric	
1	[null]	

2)

```
create view Популярный_маршрут as
(
select * from
(
select Номер_маршрута, count(Номер_билета) as Количество from Рейс
inner join Билет on Билет.Номер_рейса = Рейс.Номер_рейса
where (Дата_отправления = '2022-06-01')
group by Номер_маршрута
) as foo
where Количество =
```



```
(
select Кол from
(
select Номер_маршрута, count(Номер_билета) as Кол from Рейс
inner join Билет on Билет.Номер_рейса = Рейс.Номер_рейса
where (Дата_отправления = '2022-06-01')
group by Номер_маршрута
order by Номер_маршрута DESC
limit 1
)
as faa
)
)
```

Query Editor

Query History

1

select \* from Популярный\_маршрут

Data Output

Explain

Messages

Notifications

<div><div></div></div> Номер_маршрута <div>integer</div>	<div><div></div></div> Количество <div>bigint</div>
1	91

INSERT, DELETE, UPDATE

```
insert into Пассажир(ФИО_пассажира, Номер_и_серия_паспорта)
values
(
(select ФИО_водителя from Водитель where(Номер_водителя = '1')),
(select Номер_и_серия_паспорта_водителя from Водитель where (Номер_водителя
= '1'))
)
```

```
delete from Пассажир
where
(
Номер_и_серия_паспорта = (select Номер_и_серия_паспорта_водителя from
Водитель where (Номер_водителя = '1'))
)
```

```
update Билет
set Цена_билета =
(
select Цена_билета/2
from Билет
where Номер_билета = '1'
)
```

where Номер\_билета = '1'

Индексы:

1)

До:

```
1 select Дата_отправления from
2 (
3 select Дата_отправления, Автобус.Номер_автобуса, Модель_автобуса from Рейс
4 full join Билет on Рейс.Номер_рейса = Билет.Номер_рейса
5 inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
6 where Номер_билета is null
7 )
8 as foo
9 inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса = foo.Модель_автобуса
10 where (Название_компании = 'Mercedes')
11
12
```

Data Output Explain Messages Notifications

Successfully run. Total query runtime: 66 msec.  
5 rows affected.

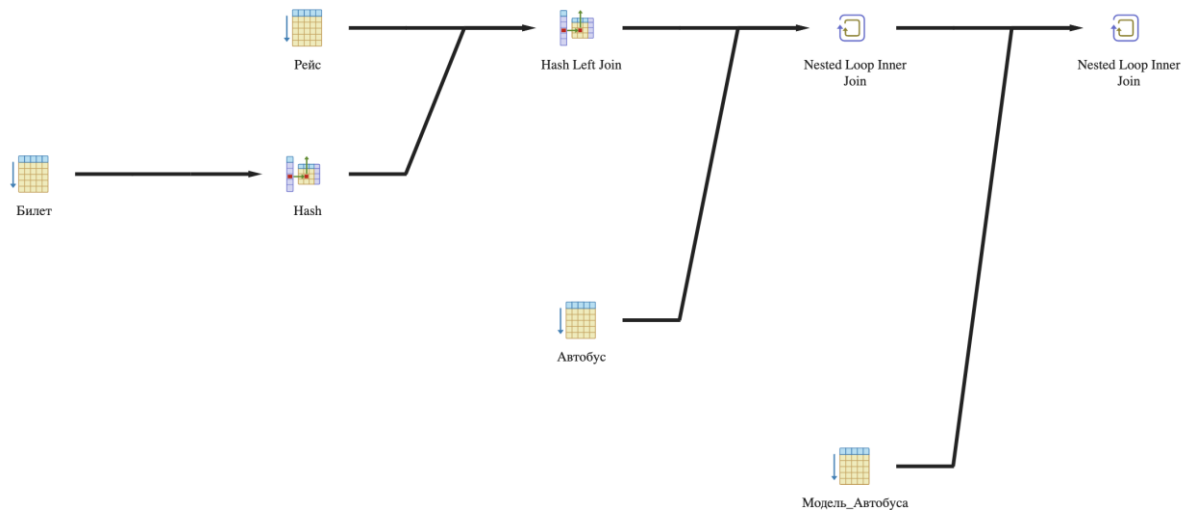
Индекс: create index Отправление\_ид on Рейс(Дата\_отправления)

После:

```
1 select Дата_отправления from
2 (
3 select Дата_отправления, Автобус.Номер_автобуса, Модель_автобуса from Рейс
4 full join Билет on Рейс.Номер_рейса = Билет.Номер_рейса
5 inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
6 where Номер_билета is null
7 )
8 as foo
9 inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса = foo.Модель_автобуса
10 where (Название_компании = 'Mercedes')
11
12
```

Data Output Explain Messages Notifications

Successfully run. Total query runtime: 39 msec.  
5 rows affected.



2)

До:

```

1  select Дата_отправления, Дата_прибытия from
2  (
3  select Дата_отправления, Дата_прибытия, Автобус.Номер_автобуса, Модель_автобуса from Рейс
4  full join Билет on Рейс.Номер_рейса = Билет.Номер_рейса
5  inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
6  where Номер_билета is null
7  )
8  as foo
9  inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса = foo.Модель_автобуса
10 where (Название_компании = 'Mercedes')
11

```

Data Output Explain Messages Notifications

Successfully run. Total query runtime: 75 msec.  
5 rows affected.

Составной индекс: create index Дата\_ид on Рейс(Дата\_отправления, Дата\_прибытия)

После:

```
1 select Дата_отправления, Дата_прибытия from
2 (
3 select Дата_отправления, Дата_прибытия, Автобус.Номер_автобуса, Модель_автобуса from Рейс
4 full join Билет on Рейс.Номер_рейса = Билет.Номер_рейса
5 inner join Автобус on Автобус.Номер_автобуса = Рейс.Номер_автобуса
6 where Номер_билета is null
7 )
8 as foo
9 inner join Модель_Автобуса on Модель_Автобуса.Модель_автобуса = foo.Модель_автобуса
10 where (Название_компании = 'Mercedes')
11 |
```

Data Output Explain Messages Notifications

Successfully run. Total query runtime: 37 msec.  
5 rows affected.

