**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №4 «Запросы на выборку и модификацию данных. Представления. Работа с индексами**»**

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Хурс П.И

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

[1. Запросы к базе данных 3](#_Toc150470282)

[2. Представления 6](#_Toc150470283)

[3. Кастом запросы 7](#_Toc150470284)

[4. Индексы 8](#_Toc150470285)

[Вывод 11](#_Toc150470286)

**Цель работы:** овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, pgadmin 4.

**Практическое задание:**

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) **с использованием подзапросов**.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. 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

* Свободные места на все поезда, отправляющиеся с вокзала в течение следующих суток.

SELECT

train.train\_id,

schedule.departure\_time,

carriage.carriage\_number,

seat.seat\_number

FROM

train

JOIN schedule ON train.shedule\_id = schedule.schedule\_id

JOIN carriage ON train.train\_id = carriage.train\_id

JOIN seat ON carriage.carriage\_id = seat.carriage\_id

LEFT JOIN ticket ON seat.seat\_id = ticket.seat\_id

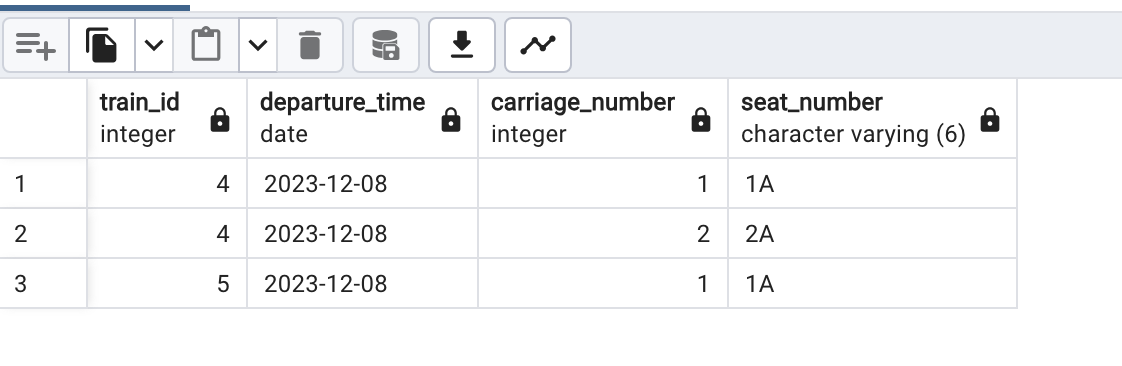
WHERE

schedule.station\_name\_from = 'StationA' AND

schedule.departure\_time BETWEEN CURRENT\_DATE + INTERVAL '1 day' AND CURRENT\_DATE + INTERVAL '2 days' AND

seat.locked\_status = false

ORDER BY

train.train\_id, carriage.carriage\_number, seat.seat\_number;

* Список пассажиров, отправившихся в Москву всеми рейсами за прошедшие сутки.

SELECT

p.passenger\_id,

p.first\_name,

p.last\_name,

s.departure\_time,

s.station\_name\_to

FROM

passenger p

JOIN ticket t ON p.passenger\_id = t.passenger\_id

JOIN seat se ON t.seat\_id = se.seat\_id

JOIN carriage c ON se.carriage\_id = c.carriage\_id

JOIN train tr ON c.train\_id = tr.train\_id

JOIN schedule s ON tr.shedule\_id = s.schedule\_id

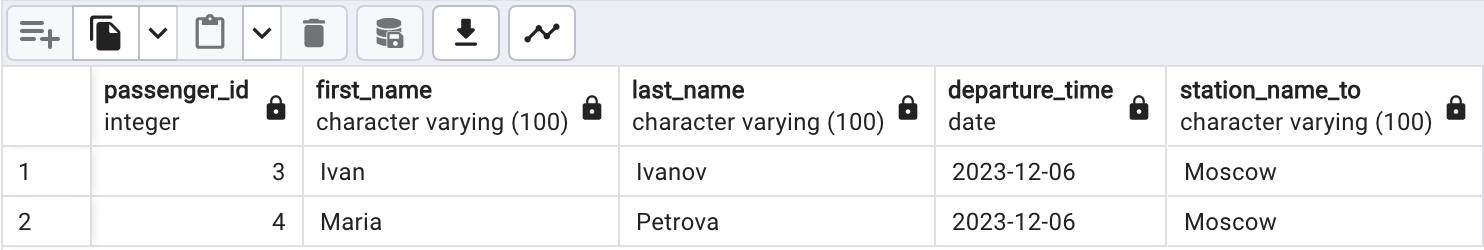
WHERE

t.station\_name\_to = 'Moscow' AND

s.departure\_time BETWEEN CURRENT\_DATE - INTERVAL '1 day' AND CURRENT\_DATE

ORDER BY

s.departure\_time DESC;



* Номера поездов, на которые проданы все билеты на следующие сутки.

SELECT

t.train\_id

FROM

train t

JOIN schedule sch ON t.shedule\_id = sch.schedule\_id

JOIN carriage c ON t.train\_id = c.train\_id

JOIN seat s ON c.carriage\_id = s.carriage\_id

LEFT JOIN ticket ti ON s.seat\_id = ti.seat\_id

WHERE

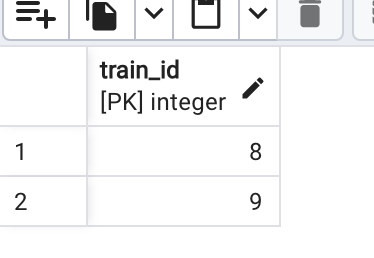
sch.departure\_time BETWEEN CURRENT\_DATE + INTERVAL '1 day' AND CURRENT\_DATE + INTERVAL '2 days'

GROUP BY

t.train\_id

HAVING

COUNT(s.seat\_id) = COUNT(ti.ticket\_id);



* Свободные места в купейные вагоны всех рейсов до Москвы на текущие сутки.

SELECT

tr.train\_id,

sch.departure\_time,

c.carriage\_number,

s.seat\_number

FROM

train tr

JOIN schedule sch ON tr.shedule\_id = sch.schedule\_id

JOIN carriage c ON tr.train\_id = c.train\_id

JOIN seat s ON c.carriage\_id = s.carriage\_id

LEFT JOIN ticket ti ON s.seat\_id = ti.seat\_id

WHERE

sch.station\_name\_to = 'Москва' AND

sch.departure\_time BETWEEN CURRENT\_DATE AND CURRENT\_DATE + INTERVAL '1 day' AND

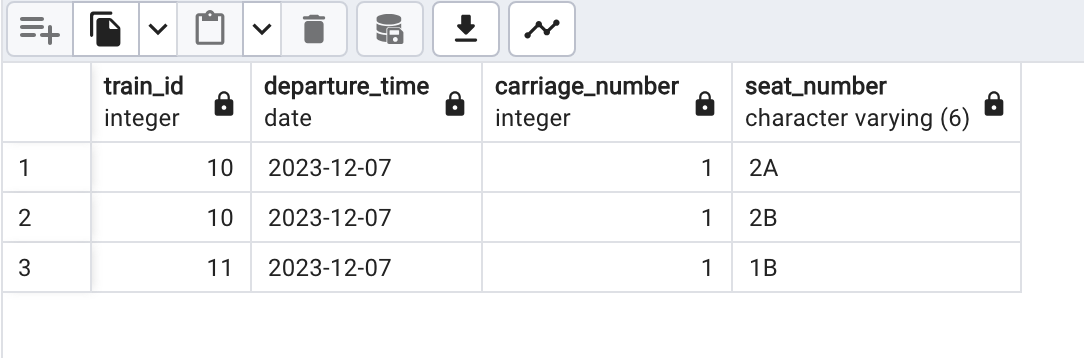
c.carriage\_type = 'купе' AND

s.locked\_status = false AND

ti.ticket\_id IS NULL

ORDER BY

tr.train\_id, c.carriage\_number, s.seat\_number;



* Выручка от продажи билетов на все поезда за прошедшие сутки.

SELECT

SUM(t.price) AS total\_revenue

FROM

ticket t

JOIN seat s ON t.seat\_id = s.seat\_id

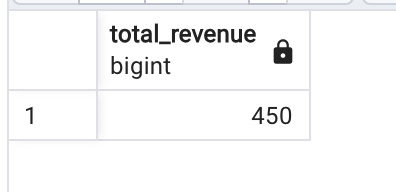
JOIN carriage c ON s.carriage\_id = c.carriage\_id

JOIN train tr ON c.train\_id = tr.train\_id

JOIN schedule sch ON tr.shedule\_id = sch.schedule\_id

WHERE

sch.departure\_time BETWEEN CURRENT\_DATE - INTERVAL '1 day' AND CURRENT\_DATE;



* Общее количество билетов, проданных по всем направлениям в вагоны типа “СВ”.

SELECT

COUNT(t.ticket\_id) AS total\_tickets

FROM

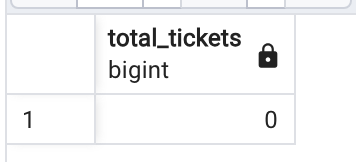
ticket t

JOIN seat s ON t.seat\_id = s.seat\_id

JOIN carriage c ON s.carriage\_id = c.carriage\_id

WHERE

c.carriage\_type = 'СВ';



* Номера и названия поездов, все вагоны которых были заполнены менее чем наполовину за прошедшие сутки.

SELECT

tr.train\_id,

sch.train\_name

FROM

train tr

JOIN schedule sch ON tr.shedule\_id = sch.schedule\_id

JOIN carriage c ON tr.train\_id = c.train\_id

LEFT JOIN seat s ON c.carriage\_id = s.carriage\_id

LEFT JOIN ticket t ON s.seat\_id = t.seat\_id

WHERE

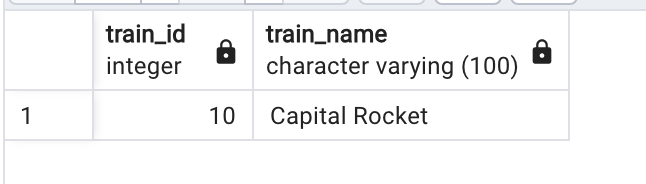
sch.departure\_time BETWEEN CURRENT\_DATE - INTERVAL '1 day' AND CURRENT\_DATE

GROUP BY

c.carriage\_id, tr.train\_id, sch.train\_name

HAVING

AVG(CASE WHEN t.ticket\_id IS NOT NULL THEN 1 ELSE 0 END) < 0.5



# Представления

* для пассажиров о наличии свободных мест на заданный рейс

CREATE VIEW view\_available\_seats AS

SELECT

s.seat\_id,

tr.train\_id,

c.carriage\_number,

s.seat\_number

FROM

train tr

JOIN carriage c ON tr.train\_id = c.train\_id

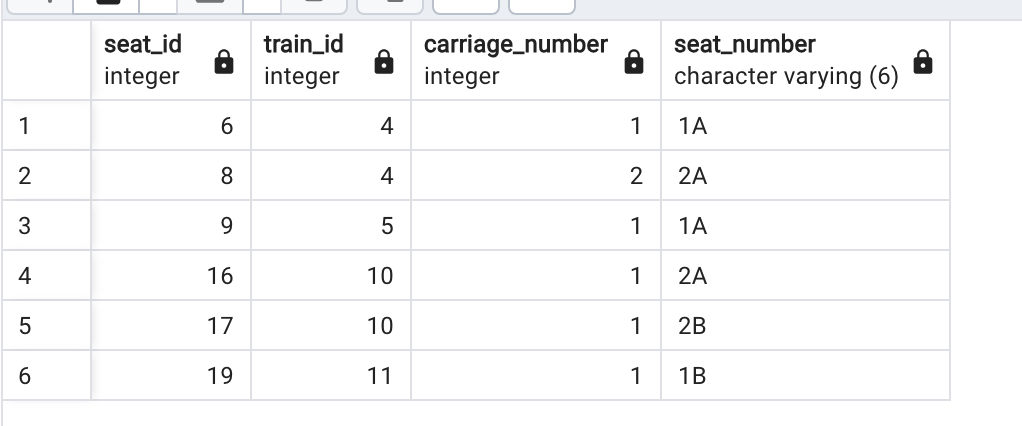
JOIN seat s ON c.carriage\_id = s.carriage\_id

LEFT JOIN ticket t ON s.seat\_id = t.seat\_id

WHERE

s.locked\_status = false AND

t.ticket\_id IS NULL;



* количество непроданных билетов на все поезда, формирующиеся за прошедшие сутки (номер поезда, тип вагона, количество).

CREATE VIEW view\_unsold\_tickets AS

SELECT

tr.train\_id,

c.carriage\_type,

COUNT(\*) - COUNT(t.ticket\_id) AS unsold\_tickets

FROM

train tr

JOIN schedule sch ON tr.shedule\_id = sch.schedule\_id

JOIN carriage c ON tr.train\_id = c.train\_id

JOIN seat s ON c.carriage\_id = s.carriage\_id

LEFT JOIN ticket t ON s.seat\_id = t.seat\_id

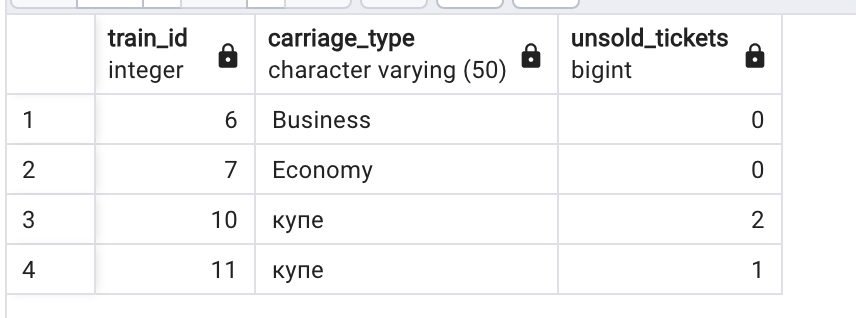
WHERE

sch.departure\_time BETWEEN CURRENT\_DATE - INTERVAL '1 day' AND CURRENT\_DATE

GROUP BY

tr.train\_id,

c.carriage\_type;



# Кастом запросы

* Измените статус всех поездов, отправление которых запланировано на следующую неделю, на "Сезонная корректировка", если по-прежнему доступно более 50% мест.

UPDATE train

SET train\_status = 'Under Maintenance'

WHERE train\_id IN (

SELECT tr.train\_id

FROM train tr

JOIN schedule sch ON tr.shedule\_id = sch.schedule\_id

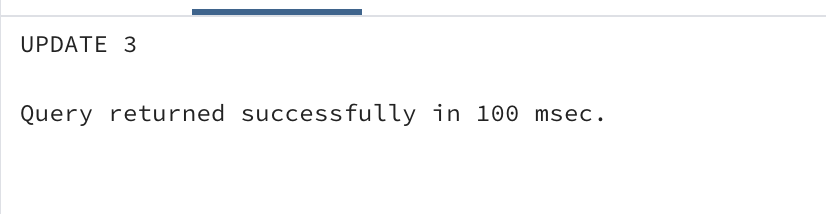
WHERE tr.train\_status = 'Delayed'

AND sch.departure\_time BETWEEN CURRENT\_DATE - INTERVAL '1 month' AND CURRENT\_DATE

GROUP BY tr.train\_id

HAVING COUNT(tr.train\_id) > 5

);



* Удалите все будущие бронирования (билеты) на поезда, которые были помечены как "Находящиеся на обслуживании" в связи с сезонными изменениями.

DELETE FROM ticket

WHERE seat\_id IN (

SELECT s.seat\_id

FROM seat s

JOIN carriage c ON s.carriage\_id = c.carriage\_id

JOIN train tr ON c.train\_id = tr.train\_id

WHERE tr.train\_status = 'Under Maintenance'

AND tr.train\_id IN (

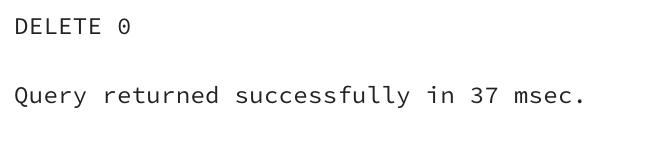
SELECT c.train\_id

FROM schedule sch

WHERE sch.departure\_time > CURRENT\_DATE

)

);



* Добавление специального поезда для мероприятия:

INSERT INTO public.train (train\_id, shedule\_id, departure\_date, arrival\_date, train\_status)

SELECT

(SELECT MAX(train\_id) + 1 FROM public.train),

schedule\_id,

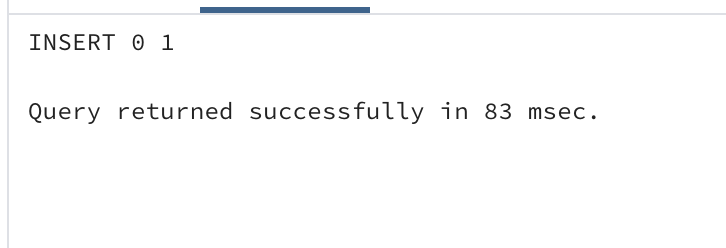
CURRENT\_DATE,

CURRENT\_DATE + INTERVAL '2 days',

'Special Event'

FROM public.schedule

WHERE station\_name\_from = 'Event City Start' AND station\_name\_to = 'Event City End';



# Индексы

Создадим индекс на даты в контракте и на айди машины:

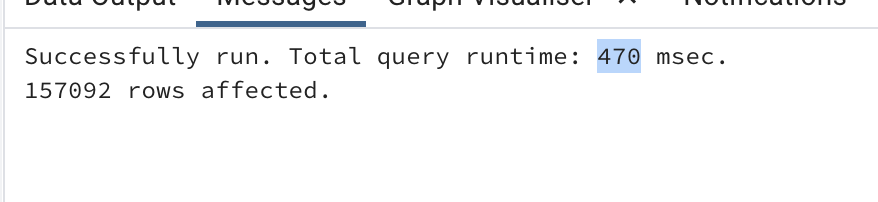
CREATE INDEX idx\_passport\_date\_of\_birth on passenger(passport\_date\_of\_birth);

SELECT \*

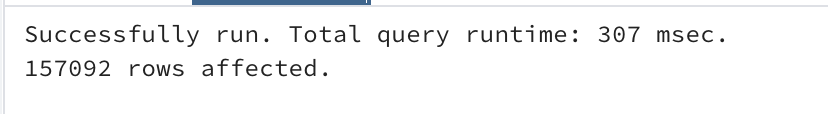
FROM passenger

WHERE EXTRACT(YEAR FROM age(passport\_date\_of\_birth)) BETWEEN 70 AND 80;

Без индекса:



С индексом:



CREATE INDEX idx\_passenger\_passport\_number

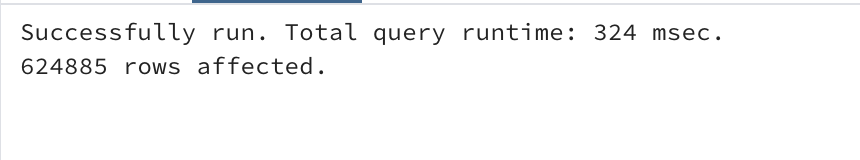
ON public.passenger (passport\_number);

SELECT passport\_number

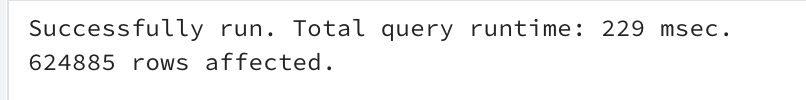
FROM passenger

WHERE passport\_number ~ '^[0-9]';

Без индекса:



С индексом:



# Вывод

В ходе лабораторной работы я освоил практические навыки по выполнению select, insert, delete и update запросов. Также, научился делать представления и индексы. В ходе анализа в первом запросе индексы дали буст в ~40%, во втором запросе практически не дали.