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
2.	Представления	6
3.	Кастом запросы	<u>9</u>
	Индексы	
	30Д	

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.

1. Запросы к базе данных

5 2023-12-08

• 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;
     train_id
              departure_time
                           carriage_number
                                         seat_number
                                         character varying (6)
     integer
              date
                           integer
1
              2023-12-08
                                         1A
2
            4 2023-12-08
                                       2
                                         2A
```

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

1 1A

3

```
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;
       =+ □ ∨ □ ∨ □ ∞
           passenger_id integer first_name character varying (100)
                                                   departure_time date station_name_to character varying (100)
                                     last_name
                                     character varying (100)
                   3 Ivan
                                                    2023-12-06
       1
                                     Ivanov
                                                               Moscow
       2
                   4 Maria
                                     Petrova
                                                    2023-12-06
                                                               Moscow
```

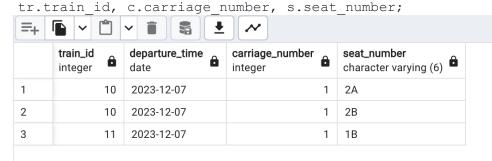
• Номера поездов, на которые проданы все билеты на следующие сутки.

```
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
   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);
train_id
    [PK] integer
            8
1
2
            9
```

• 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 = 'MockBa' 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
```



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

```
SELECT
SUM(t.price) AS total_revenue
FROM
    ticket t

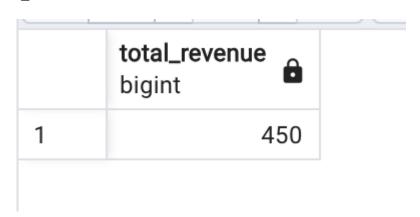
JOIN seat s ON t.seat_id = s.seat_id

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
```

• 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
```

	train_id integer	train_name character varying (100)	
1	10	Capital Rocket	

2. 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;

	seat_id integer	train_id integer	carriage_number integer	seat_number character varying (6)
1	6	4	1	1A
2	8	4	2	2A
3	9	5	1	1A
4	16	10	1	2A
5	17	10	1	2B
6	19	11	1	1B

• 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;
```

			_
	train_id integer	carriage_type character varying (50)	unsold_tickets bigint
1	6	Business	0
2	7	Economy	0
3	10	купе	2
4	11	купе	1

3. 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
);
```

UPDATE 3

Query returned successfully in 100 msec.

• 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
    )
);
```

DELETE 0

Query returned successfully in 37 msec.

• 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';
```

INSERT 0 1

Query returned successfully in 83 msec.

4. 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:

Successfully run. Total query runtime: 470 msec.
157092 rows affected.
```

С индексом:

Successfully run. Total query runtime: 307 msec. 157092 rows affected.

```
CREATE INDEX idx_passenger_passport_number
ON public.passenger (passport_number);

SELECT passport_number
FROM passenger
WHERE passport_number ~ '^[0-9]';
```

Без индекса:

Successfully run. Total query runtime: 324 msec. 624885 rows affected.

С индексом:

Successfully run. Total query runtime: 229 msec. 624885 rows affected.

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.