

**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»

Обучающийся Машковцева Марина Алексеевна

Факультет прикладной информатики

Группа K3240

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург
2024/2025

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ.....	3
2. ПРАКТИЧЕСКОЕ ЗАДАНИЕ.....	4
3. СХЕМА БАЗЫ ДАННЫХ	5
4. ВЫПОЛНЕНИЕ	6
4.1 Запросы к базе данных.....	6
4.2 Представления	8
4.3 Запросы на модификацию данных	10
4.4 Создание индексов	13
4. ВЫВОДЫ.....	17

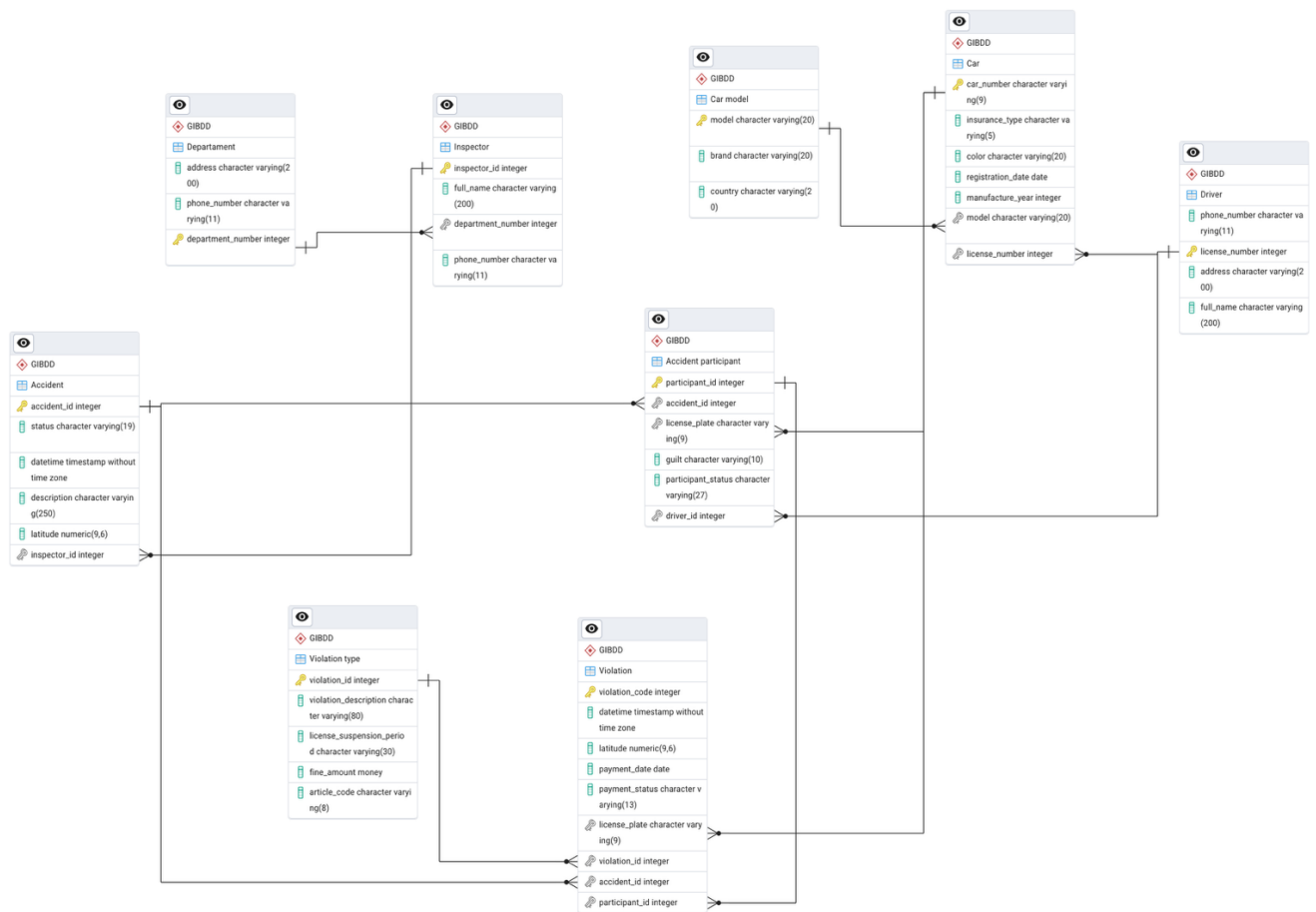
1. ЦЕЛЬ РАБОТЫ

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

2. ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, 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.

3. СХЕМА БАЗЫ ДАННЫХ



4. ВЫПОЛНЕНИЕ

4.1 Запросы к базе данных

Вывести данные водителей, многократно (более одного раза) нарушивших правила дорожного движения в заданный период.

```
SELECT d.full_name, d.license_number, d.phone_number, d.address,
       COUNT(v.violation_code) AS violations_count
FROM "GIBDD"."Driver" d
JOIN "GIBDD"."Car" c ON d.license_number = c.license_number
JOIN "GIBDD"."Violation" v ON c.car_number = v.license_plate
WHERE v.datetime BETWEEN '2023-04-01' AND '2023-05-30'
GROUP BY d.full_name, d.license_number, d.phone_number, d.address
HAVING COUNT(v.violation_code) > 1
ORDER BY violations_count DESC;
```

	full_name character varying (200)	license_number [PK] integer	phone_number character varying (11)	address character varying (200)	violations_count bigint
1	Нарушилов Сергей Викторович	1051	79161112233	г. Москва, ул. Ленина, д. 1	2

Вывести данные водителей, которые нарушили правила движения в ночное время за последнюю неделю.

```
WITH last_date AS (
    SELECT MAX(datetime) AS max_date FROM "GIBDD"."Violation"
)

SELECT d.full_name, d.license_number, d.phone_number,
       v.datetime, vt.violation_description
FROM "GIBDD"."Driver" d
JOIN "GIBDD"."Car" c ON d.license_number = c.license_number
JOIN "GIBDD"."Violation" v ON c.car_number = v.license_plate
JOIN "GIBDD"."Violation type" vt ON v.violation_id = vt.violation_id
JOIN last_date ld ON 1=1
WHERE v.datetime >= ld.max_date - INTERVAL '7 days'
      AND (EXTRACT(HOUR FROM v.datetime) BETWEEN 22 AND 23
           OR EXTRACT(HOUR FROM v.datetime) BETWEEN 0 AND 5)
ORDER BY v.datetime DESC;
```

	full_name character varying (200)	license_number integer	phone_number character varying (11)	datetime timestamp without time zone	violation_description character varying (80)
1	Нарушилов Сергей Викторович	1051	79161112233	2023-05-11 03:00:00	Превышение скорости на 20-40 км/ч

Вывести данные инспектора, оштрафовавшего максимальное число водителей.

```
SELECT i.full_name, i.inspector_id, i.phone_number,
       COUNT(DISTINCT ap.driver_id) AS drivers_fined
FROM "GIBDD"."Inspector" i
JOIN "GIBDD"."Accident" a ON i.inspector_id = a.inspector_id
JOIN "GIBDD"."Accident participant" ap ON a.accident_id = ap.accident_id
JOIN "GIBDD"."Violation" v ON ap.participant_id = v.participant_id
GROUP BY i.full_name, i.inspector_id, i.phone_number
ORDER BY drivers_fined DESC
LIMIT 1;
```

	full_name character varying (200)	inspector_id [PK] integer	phone_number character varying (11)	drivers_fined bigint
1	Иванов Алексей Владимирович	2001	79261001122	4

Вывести данные водителей, имеющих максимальное количество неоплаченных штрафов за превышение скорости.

```
SELECT d.full_name, d.license_number, d.phone_number,
       COUNT(v.violation_code) AS unpaid_speeding_fines
FROM "GIBDD"."Driver" d
JOIN "GIBDD"."Car" c ON d.license_number = c.license_number
JOIN "GIBDD"."Violation" v ON c.car_number = v.license_plate
JOIN "GIBDD"."Violation type" vt ON v.violation_id = vt.violation_id
WHERE vt.violation_description LIKE 'Превышение скорости%'
      AND v.payment_status = 'не оплачено'
GROUP BY d.full_name, d.license_number, d.phone_number
ORDER BY unpaid_speeding_fines DESC;
```

	full_name character varying (200)	license_number [PK] integer	phone_number character varying (11)	unpaid_speeding_fines bigint
1	Нарушилов Сергей Викторович	1051	79161112233	3
2	Иванов Иван Иванович	1001	79161234567	1
3	Сидорова Анна Михайловна	1003	79163456789	1
4	Ковалева Наталья Игоревна	1022	79162345678	1

Водители информацию о том, водители автомобилей какой марки реже всего подвергаются штрафу.

```
SELECT cm.brand, COUNT(v.violation_code) AS violation_count
FROM "GIBDD"."Car model" cm
JOIN "GIBDD"."Car" c ON cm.model = c.model
JOIN "GIBDD"."Violation" v ON c.car_number = v.license_plate
GROUP BY cm.brand
ORDER BY violation_count
LIMIT 3;
```

	brand character varying (20) 🔒	violation_count bigint 🔒
1	Ford	1
2	Nissan	1
3	BMW	1

Сколько водителей было лишено прав за прошедшую неделю.

```
WITH last_date AS (
    SELECT MAX(datetime) AS max_date FROM "GIBDD"."Violation"
)

SELECT COUNT(DISTINCT d.license_number) AS drivers_disqualified
FROM "GIBDD"."Driver" d
JOIN "GIBDD"."Car" c ON d.license_number = c.license_number
JOIN "GIBDD"."Violation" v ON c.car_number = v.license_plate
JOIN "GIBDD"."Violation type" vt ON v.violation_id = vt.violation_id
JOIN last_date ld ON 1=1
WHERE v.datetime >= ld.max_date - INTERVAL '7 days'
AND vt.license_suspension_period IS NOT NULL;
```

	drivers_disqualified bigint 🔒
1	1

За какое нарушение чаще всего штрафуются водители?

```
SELECT vt.violation_description, COUNT(v.violation_code) AS violation_count
FROM "GIBDD"."Violation type" vt
JOIN "GIBDD"."Violation" v ON vt.violation_id = v.violation_id
GROUP BY vt.violation_description
ORDER BY violation_count DESC
LIMIT 1;
```

	violation_description character varying (80) 🔒	violation_count bigint 🔒
1	Превышение скорости на 20-40 км/ч	5

4.2 Представления

Создать представление, содержащее следующие данные: вид нарушения, время нарушения, номер водительского удостоверения, сумма штрафа за истекший год


```

CREATE OR REPLACE VIEW "GIBDD"."violations" AS
SELECT
    vt.violation_description AS "Вид нарушения",
    v.datetime AS "Время нарушения",
    d.license_number AS "Номер водительского удостоверения",
    vt.fine_amount AS "Сумма штрафа",
    v.payment_status AS "Статус оплаты",
    d.full_name AS "ФИО водителя",
    c.car_number AS "Госномер автомобиля",
    c.model AS "Модель автомобиля"
FROM
    "GIBDD"."Violation" v
JOIN
    "GIBDD"."Violation type" vt ON v.violation_id = vt.violation_id
JOIN
    "GIBDD"."Car" c ON v.license_plate = c.car_number
JOIN
    "GIBDD"."Driver" d ON c.license_number = d.license_number
WHERE
    EXTRACT(YEAR FROM v.datetime) = 2023
ORDER BY
    v.datetime DESC;

```

NOTICE: identifier "Номер водительского удостоверения" will be truncated to "Номер водительского удостоверени"

CREATE VIEW

Запрос завершён успешно, время выполнения: 75 мсек.

	Время нарушения timestamp without time zone	Номер водительского удостоверени integer	Сумма штрафа money	Статус оплаты character varying (13)	ФИО водителя character varying (200)	Госномер автомобиля character varying (9)	Модель автомобиля character varying (20)
1	2023-05-11 03:00:00	1051	\$500.00	не оплачено	Нарушилов Сергей Викторович	X111XX777	Solaris
2	2023-05-10 15:30:00	1051	\$1,000.00	не оплачено	Нарушилов Сергей Викторович	X111XX777	Solaris
3	2023-05-01 12:00:00	1051	\$500.00	не оплачено	Нарушилов Сергей Викторович	X111XX777	Solaris
4	2023-04-28 03:00:00	1051	\$500.00	не оплачено	Нарушилов Сергей Викторович	X111XX777	Solaris
5	2023-04-15 13:30:00	1030	\$1,500.00	не оплачено	Савельева Алина Викторовна	E2220П34	Duster
6	2023-04-10 10:15:00	1029	\$1,000.00	100% оплачено	Герасимов Михаил Сергеевич	D999MH23	Polo
7	2023-04-05 20:00:00	1028	\$30,000.00	не оплачено	Данилова Екатерина Олеговна	G666Л012	Sportage
8	2023-04-01 17:45:00	1027	\$30,000.00	50% оплачено	Филиппов Кирилл Андреевич	B333KM01	Vesta
9	2023-03-25 14:30:00	1026	\$1,500.00	не оплачено	Максимова Анна Дмитриевна	B888ИУ90	Rio
10	2023-03-20 09:15:00	1025	\$500.00	100% оплачено	Семенов Игорь Александрович	A5553T89	CX5
11	2023-03-15 21:00:00	1024	\$800.00	не оплачено	Воронцова Ольга Сергеевна	Я222ЖР78	Logan
12	2023-03-10 18:45:00	1023	\$2,500.00	50% оплачено	Жуков Артем Викторович	Ю666ДУ67	Granta
13	2023-03-05 15:30:00	1022	\$1,000.00	не оплачено	Ковалева Наталья Игоревна	З444ГK56	Camry
14	2023-03-01 12:00:00	1021	\$500.00	100% оплачено	Новиков Денис Олегович	Щ999БМ45	Solaris
15	2023-02-20 07:20:00	1020	\$1,500.00	не оплачено	Егорова Татьяна Викторовна	Ц777ЩН34	Focus

Вывести данные водителей, который участвовали в аварии в текущем месяце.

```
CREATE OR REPLACE VIEW "GIBDD"."april_2023_accident_participants" AS
SELECT
    d.full_name AS "ФИО водителя",
    d.license_number AS "Номер водительского удостоверения",
    d.phone_number AS "Телефон",
    d.address AS "Адрес",
    c.car_number AS "Госномер автомобиля",
    c.model AS "Модель автомобиля",
    a.accident_id AS "Номер аварии",
    a.datetime AS "Дата и время аварии",
    a.description AS "Описание аварии",
    ap.guilt AS "Виновность",
    ap.participant_status AS "Статус участника"
FROM
    "GIBDD"."Driver" d
JOIN
    "GIBDD"."Car" c ON d.license_number = c.license_number
JOIN
    "GIBDD"."Accident participant" ap ON c.car_number = ap.license_plate
JOIN
    "GIBDD"."Accident" a ON ap.accident_id = a.accident_id
WHERE
    EXTRACT(MONTH FROM a.datetime) = 4
    AND EXTRACT(YEAR FROM a.datetime) = 2023
ORDER BY
    a.datetime DESC, d.full_name;
```

NOTICE: identifier "Номер водительского удостоверения" will be truncated to "Номер водительского удостоверени"
CREATE VIEW

Запрос завершён успешно, время выполнения: 63 msec.

	ФИО водителя character varying (200)	Номер водит integer	Телефон character varying (20)	Адрес character varying (200)	Госномер авт character varying (10)	Модель авт character varying (50)	Номер авт integer	Дата и время аварии timestamp without time zone	Описание аварии character varying (250)	Виновность character varying (20)	Статус участника character varying (27)
1	Нарушилов Сергей Викторович	1051	79161112233	г. Москва, ул. Ленина, д. 1	X111XX777	Solaris	23	2023-04-28 03:00:00	Ночное нарушение	виновен	активный

4.3 Запросы на модификацию данных

INSERT

Добавить новое нарушение для водителя, который имеет больше всего неоплаченных штрафов.

До:

	full_name character varying (200)	unpaid_count bigint
1	Нарушилов Сергей Викторович	4

Запрос:

```

INSERT INTO "GIBDD"."Violation" (
    violation_code, datetime, latitude, payment_date,
    payment_status, license_plate, violation_id
)
SELECT
    (SELECT MAX(violation_code) + 1 FROM "GIBDD"."Violation"),
    CURRENT_TIMESTAMP,
    55.751244,
    NULL,
    'не оплачено',
    (SELECT c.car_number
     FROM "GIBDD"."Car" c
     JOIN "GIBDD"."Driver" d ON c.license_number = d.license_number
     WHERE d.license_number = (
         SELECT c.license_number
         FROM "GIBDD"."Violation" v
         JOIN "GIBDD"."Car" c ON v.license_plate = c.car_number
         WHERE v.payment_status = 'не оплачено'
         GROUP BY c.license_number
         ORDER BY COUNT(*) DESC
         LIMIT 1
     )
    )
    LIMIT 1),
1;

```

После:

	full_name character varying (200)	unpaid_count bigint
1	Нарушилов Сергей Викторович	5

UPDATE

Для всех водителей, которые были виновны в ДТП, изменить статус неоплаченных штрафов на "50% оплачено".

До:

	full_name character varying (200)	violation_code integer	datetime timestamp without time	latitude numeric (9,6)	payment_date date	payment_status character varying (13)
1	Иванов Иван Иванович	1	2023-01-05 08:15:00	55.755800	[null]	не оплачено
2	Федоров Максим Олегович	10	2023-01-25 16:10:00	59.941800	[null]	не оплачено
3	Антонова Ирина Александровна	12	2023-02-01 11:30:00	55.796100	[null]	не оплачено
4	Воронцова Ольга Сергеевна	24	2023-03-15 21:00:00	56.838000	[null]	не оплачено
5	Максимова Анна Дмитриевна	26	2023-03-25 14:30:00	43.585500	[null]	не оплачено

Запрос:

```
UPDATE "GIBDD"."Violation"
SET payment_status = '50% оплачено',
    payment_date = CURRENT_DATE
WHERE payment_status = 'не оплачено'
AND license_plate IN (
    SELECT c.car_number
    FROM "GIBDD"."Car" c
    WHERE c.license_number IN (
        SELECT driver_id FROM "GIBDD"."Accident participant"
        WHERE guilt = 'виновен'
    )
);
```

После:

	full_name character varying (200)	violation_code integer	datetime timestamp without time zone	latitude numeric (9,6)	payment_date date	payment_status character varying (13)
1	Максимова Анна Дмитриевна	26	2023-03-25 14:30:00	43.585500	2025-04-30	50% оплачено
2	Антонова Ирина Александровна	12	2023-02-01 11:30:00	55.796100	2025-04-30	50% оплачено
3	Воронцова Ольга Сергеевна	24	2023-03-15 21:00:00	56.838000	2025-04-30	50% оплачено
4	Иванов Иван Иванович	1	2023-01-05 08:15:00	55.755800	2025-04-30	50% оплачено
5	Федоров Максим Олегович	10	2023-01-25 16:10:00	59.941800	2025-04-30	50% оплачено

DELETE

Удалить все оплаченные штрафы старше 6 месяцев.




До:

	violation_code [PK] integer	datetime timestamp without time zone	payment_date date
1	2	2023-01-05 08:15:00	2023-01-10
2	5	2023-01-15 19:30:00	2023-01-20
3	9	2023-01-25 16:10:00	2023-01-30
4	13	2023-02-05 13:40:00	2023-02-10
5	17	2023-02-15 21:15:00	2023-02-20
6	21	2023-03-01 12:00:00	2023-03-05
7	25	2023-03-20 09:15:00	2023-03-25
8	29	2023-04-10 10:15:00	2023-04-15

Запрос:

```
DELETE FROM "GIBDD"."Violation"
WHERE violation_code IN (
    SELECT violation_code
    FROM "GIBDD"."Violation"
    WHERE payment_status = '100% оплачено'
    AND datetime < CURRENT_DATE - INTERVAL '6 months'
    AND violation_code NOT IN (
        SELECT violation_code
        FROM "GIBDD"."Violation"
        WHERE accident_id IS NOT NULL
    )
);
```

После:

	violation_code [PK] integer 	datetime timestamp without time zone 	payment_date date 
--	--	---	--

4.4 Создание индексов


Выполнение запросов без индексов:

Найти все нарушения водителя с license_number = 1001.

Запрос:

```
EXPLAIN ANALYZE
SELECT v.*, vt.violation_description
FROM "GIBDD"."Violation" v
JOIN "GIBDD"."Violation type" vt ON v.violation_id = vt.violation_id
JOIN "GIBDD"."Car" c ON v.license_plate = c.car_number
WHERE c.license_number = 1001;
```

План и время:

	QUERY PLAN text 
1	Nested Loop (cost=1.78..5.64 rows=1 width=246) (actual time=0.287..0.296 rows=1 loops=1)
2	-> Hash Join (cost=1.64..4.09 rows=1 width=68) (actual time=0.108..0.116 rows=1 loops=1)
3	Hash Cond: ((v.license_plate)::text = (c.car_number)::text)
4	-> Seq Scan on "Violation" v (cost=0.00..2.35 rows=35 width=68) (actual time=0.015..0.025 rows=27 loops=1)
5	-> Hash (cost=1.62..1.62 rows=1 width=12) (actual time=0.059..0.059 rows=1 loops=1)
6	Buckets: 1024 Batches: 1 Memory Usage: 9kB
7	-> Seq Scan on "Car" c (cost=0.00..1.62 rows=1 width=12) (actual time=0.027..0.042 rows=1 loops=1)
8	Filter: (license_number = 1001)
9	Rows Removed by Filter: 50
10	-> Index Scan using unique_violation_type on "Violation type" vt (cost=0.14..1.53 rows=1 width=182) (actual time=0.176..0.176 rows=1 loop...)
11	Index Cond: (violation_id = v.violation_id)
12	Planning Time: 15.686 ms
13	Execution Time: 0.356 ms

Индексы:

```
CREATE INDEX idx_car_license_number ON "GIBDD"."Car"(license_number);  
CREATE INDEX idx_violation_license_plate ON "GIBDD"."Violation"(license_plate);
```

Время выполнения с индексами:

	QUERY PLAN text	
5	-> Hash (cost=1.04..1.04 rows=1 width=12) (actual time=0.019..0.020 rows=1 loops=1)	
6	Buckets: 1024 Batches: 1 Memory Usage: 9kB	
7	-> Seq Scan on "Car" c (cost=0.00..1.64 rows=1 width=12) (actual time=0.009..0.013 rows=1 loops=1)	
8	Filter: (license_number = 1001)	
9	Rows Removed by Filter: 50	
10	-> Index Scan using unique_violation_type on "Violation type" vt (cost=0.14..1.94 rows=1 width=182) (actual time=0.008..0.008 rows=1 loop..	
11	Index Cond: (violation_id = v.violation_id)	
12	Planning Time: 3.796 ms	
13	Execution Time: 0.119 ms	

Найти все аварии за апрель 2023 со статусом 'завершено'.

Запрос:

```
EXPLAIN ANALYZE  
SELECT a.*, i.full_name as inspector_name  
FROM "GIBDD"."Accident" a  
JOIN "GIBDD"."Inspector" i ON a.inspector_id = i.inspector_id  
WHERE a.status = 'завершено'  
AND a.datetime BETWEEN '2023-04-01' AND '2023-04-30';
```

План и время:

	QUERY PLAN text	
1	Nested Loop (cost=0.14..20.36 rows=1 width=1020) (actual time=0.816..0.829 rows=2 loops=1)	
2	-> Seq Scan on "Accident" a (cost=0.00..12.10 rows=1 width=602) (actual time=0.342..0.346 rows=2 loops=1)	
3	Filter: ((datetime >= '2023-04-01 00:00:00':timestamp without time zone) AND (datetime <= '2023-04-30 00:00:00':timestamp without t	
4	Rows Removed by Filter: 22	
5	-> Index Scan using unique_inspector on "Inspector" i (cost=0.14..8.16 rows=1 width=422) (actual time=0.236..0.236 rows=1 loops=2)	
6	Index Cond: (inspector_id = a.inspector_id)	
7	Planning Time: 10.194 ms	
8	Execution Time: 0.864 ms	

Индексы:

```
CREATE INDEX idx_accident_status_datetime ON "GIBDD"."Accident"(status, datetime);
```

Время выполнения с индексами:

	QUERY PLAN text
1	Nested Loop (cost=0.14..9.68 rows=1 width=1020) (actual time=0.015..0.018 rows=2 loops=1)
2	-> Seq Scan on "Accident" a (cost=0.00..1.42 rows=1 width=602) (actual time=0.010..0.011 rows=2 loops=1)
3	Filter: ((datetime >= '2023-04-01 00:00:00'::timestamp without time zone) AND (datetime <= '2023-04-30 00:00:00'::tim
4	Rows Removed by Filter: 22
5	-> Index Scan using unique_inspector on "Inspector" i (cost=0.14..8.16 rows=1 width=422) (actual time=0.002..0.002 ro
6	Index Cond: (inspector_id = a.inspector_id)
7	Planning Time: 2.094 ms
8	Execution Time: 0.032 ms

Найти водителей, участвовавших в ДТП с тяжелыми последствиями (тяжкие телесные, погиб при аварии).

Запрос:

EXPLAIN ANALYZE

SELECT

```

    d.full_name,
    d.license_number,
    COUNT(DISTINCT ap.accident_id) as severe_accidents_count,
    STRING_AGG(DISTINCT a.description, ';' ) as accidents_descriptions
FROM "GIBDD"."Driver" d
JOIN "GIBDD"."Accident participant" ap ON d.license_number = ap.driver_id
JOIN "GIBDD"."Accident" a ON ap.accident_id = a.accident_id
WHERE ap.participant_status IN ('тяжкие телесные', 'погиб при аварии')
AND ap.guilt = 'виновен'
GROUP BY d.full_name, d.license_number
HAVING COUNT(DISTINCT ap.accident_id) > 0
ORDER BY severe_accidents_count DESC;
```

План и время выполнения:

	QUERY PLAN text	
14	Buckets: 1024 Batches: 1 Memory Usage: 9kB	
15	-> Hash Join (cost=1.46..3.17 rows=3 width=56) (actual time=0.061..0.084 rows=4 loops=1)	
16	Hash Cond: (d.license_number = ap.driver_id)	
17	-> Seq Scan on "Driver" d (cost=0.00..1.50 rows=50 width=52) (actual time=0.006..0.015 rows=51 loops=1)	
18	-> Hash (cost=1.42..1.42 rows=3 width=8) (actual time=0.035..0.036 rows=4 loops=1)	
19	Buckets: 1024 Batches: 1 Memory Usage: 9kB	
20	-> Seq Scan on "Accident participant" ap (cost=0.00..1.42 rows=3 width=8) (actual time=0.019..0.030 r...	
21	Filter: (((participant_status)::text = ANY ('{'тяжкие телесные','погиб при аварии'}::text[])) AND ((g...	
22	Rows Removed by Filter: 28	
23	Planning Time: 10.891 ms	
24	Execution Time: 0.411 ms	

Индексы:

```
CREATE INDEX idx_accident_participant_status ON "GIBDD"."Accident participant"(participant_status)
WHERE participant_status IN ('тяжкие телесные', 'погиб при аварии');
```

```
CREATE INDEX idx_accident_participant_guilt ON "GIBDD"."Accident participant"(guilt)
WHERE guilt = 'виновен';
```

```
CREATE INDEX idx_accident_datetime ON "GIBDD"."Accident"(datetime);
```

Время выполнения с индексами:

	QUERY PLAN text
16	Hash Cond. (a.accident_id = ap.accident_id)
17	-> Seq Scan on "Accident" a (cost=0.00..1.24
18	-> Hash (cost=1.48..1.48 rows=3 width=8) (ac
19	Buckets: 1024 Batches: 1 Memory Usage:
20	-> Seq Scan on "Accident participant" ap (
21	Filter: (((participant_status)::text = ANY
22	Rows Removed by Filter: 28
23	Planning Time: 3.036 ms
24	Execution Time: 0.177 ms

5. ВЫВОДЫ

В ходе выполнения лабораторной работы были успешно освоены практические навыки работы с PostgreSQL, включая создание запросов на выборку и модификацию данных, использование представлений, а также работу с индексами.

Запросы на выборку данных позволили получить необходимую информацию из базы данных, такую как данные о водителях с повторными нарушениями, инспекторах с максимальным количеством штрафов и т. д.

Представления упростили доступ к часто используемым данным, например, к информации о нарушениях за истекший год или водителях, участвовавших в авариях.

Запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов продемонстрировали возможность эффективного изменения данных на основе сложных условий.

Работа с индексами показала их значительное влияние на производительность запросов. Создание простых и составных индексов позволило сократить время выполнения запросов, что было подтверждено с помощью команды EXPLAIN.