

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: Залетов А.Д.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Цель работы	3
Практическое задание	3
Вариант 18. БД «ГИБДД»	4
Вывод.....	13

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.

Вариант 18. БД «ГИБДД»

Создание запросов

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

The screenshot shows the pgAdmin 4 interface. The Object Explorer on the left shows the database structure. The SQL editor in the center contains the following query:

```
1 SELECT
2     CO."DL_number",
3     CO."Driver_name_surname",
4     COUNT(*) AS "ViolationCount"
5 FROM
6     "car_owner" CO
7 JOIN
8     "registered_car" RC ON CO."DL_number" = RC."DL_number"
9 JOIN
10    "violation" V ON RC."PTS_number" = V."PTS_number"
11 WHERE
12    V."Violation_date" BETWEEN '2023-11-02' AND '2023-11-08'
13 GROUP BY
14    CO."DL_number", CO."Driver_name_surname"
15 HAVING
16    COUNT(*) > 1;
```

The Data Output window shows the results of the query:

DL_number [PK] integer	Driver_name_surname character varying (255)	ViolationCount bigint	
1	1234567	Иванов Иван Иванович	2

Total rows: 1 of 1 Query complete 00:00:00.071 Ln 12, Col 43

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

The screenshot shows the pgAdmin 4 interface. The Object Explorer on the left shows the database structure. The SQL editor in the center contains the following query:

```
1 SELECT
2     CO."DL_number",
3     CO."Driver_name_surname",
4     CO."Address",
5     CO."Telephone_number",
6     CO."Date_of_birth",
7     CO."Passport"
8 FROM
9     "car_owner" CO
10 JOIN
11    "registered_car" RC ON CO."DL_number" = RC."DL_number"
12 JOIN
13    "violation" V ON RC."PTS_number" = V."PTS_number"
14 JOIN
15    "Violation_types" VT ON V."Violation_id" = VT."Violation_ID"
16 WHERE
17    V."Violation_time" BETWEEN TIME '00:00:00' AND TIME '06:00:00'
18    AND V."Violation_date" BETWEEN CURRENT_DATE - INTERVAL '1 week' AND CURRENT_DATE;
```

The Data Output window shows the results of the query:

DL_number [PK] integer	Driver_name_surname character varying (255)	Address character varying (255)	Telephone_number character varying (15)	Date_of_birth date	Passport character varying (15)	
1	1234566	Сергеев Иван Иванович	Биржевая Улица 14	+79657005804	1996-05-03	4745 501529

Total rows: 1 of 1 Query complete 00:00:00.329 Ln 18, Col 86

3. Вывести данные водителей, заплативших штраф одному и тому же инспектору более одного раза.

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query' tab. The query is as follows:

```

1 SELECT DISTINCT
2   CO."DL_number",
3   CO."Driver_name_surname",
4   P."Policeman_name_surname" AS "Inspecting_Policeman"
5 FROM
6   "car_owner" CO
7 JOIN
8   "registered_car" RC ON CO."DL_number" = RC."DL_number"
9 JOIN
10  "violation" V ON RC."PTS_number" = V."PTS_number"
11 JOIN
12  "Policemen" P ON V."Personal_number" = P."Personal_number"
13 WHERE
14  V."Payment_status" = '1'
15 AND (
16    SELECT COUNT(*)
17    FROM "violation" V2
18    WHERE V2."Personal_number" = P."Personal_number"
19    AND V2."Payment_status" = '1'
20  ) > 1;

```

The 'Data Output' tab shows the following result:

DL_number	Driver_name_surname	Inspecting_Policeman
1234567	Иванов Иван Иванович	Деревсков Денис Климентьевич

The status bar at the bottom indicates 'Ln 19, Col 42'.

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

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query' tab. The query is as follows:

```

1 WITH CarViolationCounts AS (
2   SELECT
3     "Car_model"."Label" AS "CarLabel",
4     COUNT("violation"."Violation_ID") AS "ViolationCount",
5     RANK() OVER (ORDER BY COUNT("violation"."Violation_ID") ASC) AS "Rank"
6   FROM
7     "Car_model"
8   JOIN "Car" ON "Car_model"."Model_ID" = "Car"."Model_ID"
9   JOIN "registered_car" ON "Car"."WIN_number" = "registered_car"."WIN_number"
10  JOIN "violation" ON "registered_car"."PTS_number" = "violation"."PTS_number"
11  GROUP BY "Car_model"."Label"
12 )
13
14 SELECT "CarLabel" AS "Least_Frequently_Ticketed_Car"
15 FROM CarViolationCounts
16 WHERE "Rank" = 1;

```

The 'Data Output' tab shows the following result:

Least_Frequently_Ticketed_Car
Audi
BMW

The status bar at the bottom indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.173'.

5. Вывести данные инспектора, оштрафовавшего максимальное число водителей.

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query' tab. The query is as follows:

```

14 V."Violation_date" BETWEEN '2022-11-03' AND '2024-11-08'
15 GROUP BY
16 P."Personal_number", P."Policeman_name_surname", P."Rank", P."Department_ID", P."Passport"
17 )
18 , MaxViolations AS (
19 SELECT
20 MAX("TotalViolations") AS "MaxViolations"
21 FROM
22 RankedViolations
23 )
24 SELECT
25 "Personal_number",
26 "Policeman_name_surname",
27 "Rank",
28 "Department_ID",
29 "Passport",
30 "TotalViolations"
31 FROM
32 RankedViolations
33 JOIN
34 MaxViolations ON "TotalViolations" = "MaxViolations";

```

The 'Data Output' tab shows the results of the query:

Personal_number	Policeman_name_surname	Rank	Department_ID	Passport	TotalViolations
1	Деревсков Денис Климентьевич	Сержант	1	4063 170339	2
2	Левтев Ефим Степанович	Рядовой	1	4978 568220	2

Total rows: 2 of 2 Query complete 00:00:00.139 Ln 34, Col 58

```

WITH RankedViolations AS (
  SELECT
    P."Personal_number",
    P."Policeman_name_surname",
    P."Rank",
    P."Department_ID",
    P."Passport",
    COUNT(*) AS "TotalViolations"
  FROM
    "Policemen" P
  JOIN
    "violation" V ON P."Personal_number" = V."Personal_number"
  WHERE
    V."Violation_date" BETWEEN '2022-11-03' AND '2024-11-08'
  GROUP BY
    P."Personal_number", P."Policeman_name_surname", P."Rank", P."Department_ID",
    P."Passport"
)
, MaxViolations AS (
  SELECT
    MAX("TotalViolations") AS "MaxViolations"
  FROM
    RankedViolations
)
SELECT
  "Personal_number",
  "Policeman_name_surname",

```

```

"Rank",
"Department_ID",
"Passport",
"TotalViolations"
FROM
RankedViolations
JOIN
MaxViolations ON "TotalViolations" = "MaxViolations";

```

6. Сколько водителей было лишено прав за прошедшую неделю.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer with a tree view of the database schema. The 'violation' table is selected. The main pane displays a SQL query in the Query editor. The query is as follows:

```

1 SELECT
2   COUNT(DISTINCT CO."DL_number") AS "Count_of_Drivers_Lost_License"
3 FROM
4   "car_owner" CO
5 JOIN
6   "registered_car" RC ON CO."DL_number" = RC."DL_number"
7 JOIN
8   "violation" V ON RC."PTS_number" = V."PTS_number"
9 JOIN
10  "Violation_types" VT ON V."Violation_id" = VT."Violation_ID"
11 WHERE
12   VT."DL_loss_time" IS NOT NULL
13   AND VT."DL_loss_time" > 0
14   AND V."Violation_date" BETWEEN CURRENT_DATE - INTERVAL '1 week' AND CURRENT_DATE;

```

Below the query editor, the Data Output pane shows the results of the query. It displays a single row with the value 2 for the column Count_of_Drivers_Lost_License.

Count_of_Drivers_Lost_License
2

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.188'.

7. За какое нарушение чаще всего штрафуются водители.

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure. The main pane displays a SQL query for the 'violation' table. The query is a complex join between 'violation' and 'violation_types' tables, grouped by violation name and ID, and ordered by the count of violations in descending order, limited to 1.

```
1 SELECT
2   VT."Violation_name" AS "Most_Frequently_Penalized_Violation",
3   VT."Violation_ID" AS "Most_Frequently_Penalized_Violation_ID",
4   COUNT(*) AS "Penalty_Count"
5 FROM
6   "violation" V
7 JOIN
8   "violation_types" VT ON V."Violation_id" = VT."Violation_ID"
9 GROUP BY
10  VT."Violation_name", VT."Violation_ID"
11 HAVING COUNT(*)=(SELECT COUNT(*)
12 FROM Violation v
13 JOIN "Violation_types" vt ON v."Violation_id" = vt."Violation_ID"
14 GROUP BY vt."Violation_name"
15 ORDER BY COUNT(*) DESC LIMIT 1);
```

The Data Output pane shows the results of the query:

	Most_Frequently_Penalized_Violation_ID	Penalty_Count
1	портным средством, не зарегистрированным в установленном порядке	2
2	ния транспортным средством лицу, не имеющему при себе документов на право управления ...	2

Total rows: 2 of 2 Query complete 00:00:00.089 Ln 15, Col 34

2)Запросы INSERT,UPDATE,DELETE

2.1)INSERT

The screenshot shows the pgAdmin 4 interface. The main pane displays an INSERT query. The query uses a WITH clause to select data from a table named 'NewViolationData' and inserts it into the 'violation' table. The query is successful, returning 1 row.

```
1 WITH NewViolationData AS (
2   SELECT
3     5 AS "violation_ID",
4     'E100BK37' AS "CarNumber",
5     '1' AS "InspectorNumber",
6     1 AS "PaymentStatus",
7     'Улица Ломоносова' AS "ViolationPlace",
8     '2023-11-25'::date AS "ViolationDate",
9     '1'::integer AS "ViolationID",
10    '06:39:30'::time AS "ViolationTime"
11 )
12 INSERT INTO "violation" ("Violation_ID","PTS_number", "Personal_number", "Payment_status", "V
13 VALUES (
14   (SELECT "violation_ID" FROM NewViolationData),
15   (SELECT "PTS_number" FROM "registered_car" WHERE "CarNumber" = (SELECT "CarNumber" FROM New
16   (SELECT "Personal_number" FROM "Policemen" WHERE "Personal_number" = (SELECT "InspectorNumbe
17   (SELECT "PaymentStatus" FROM NewViolationData),
18   (SELECT "ViolationPlace" FROM NewViolationData),
```

The Data Output pane shows the results of the query:

```
INSERT 0 1
Query returned successfully in 46 msec.
```

Total rows: 0 of 0 Query complete 00:00:00.046 Ln 3, Col 21

pgAdmin 4

File Object Tools Help

Object Explorer

- Violation_types
- car_owner
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- registered_car
- violation
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Trigger Functions
- Types
- Views (3)
 - accident_participants_view_vers
 - violation_summary_remake2
 - Columns
 - Rules
 - Triggers
 - violation_summary_remake3
- Subscriptions
- emp_time
- postgres
- Login/Group Roles
- Tablespaces

public.violation/GIBDD/postgres@PostgreSQL 16

Query Query History

```
1 SELECT * FROM public.violation
2 ORDER BY "Violation_ID" ASC
```

Data Output Messages

D	PTS_number	Personal_number	Violation_place	Violation_date	Violation_id	Violation_time	Payment
#	character varying (18)	character varying (18)	character varying (255)	date	integer	time without time zone	integer
1	02KP362311	1	Загородный проспект 15	2023-11-26	1	05:03:00	
2	02KP362311	2	Гражданский проспект 24	2023-11-26	2	04:21:00	
3	03ET23145	1	Серебрястый бульвар 35	2023-11-23	2	21:00:00	
4	04VE32664	2	Звенигородская улица 22	2023-11-26	1	05:03:21	
5	02KP362311	1	Улица Ломоносова	2023-11-25	1	06:39:30	
6	[null]	[null]	[null]	[null]	[null]	[null]	

Servers > PostgreSQL 16 > Databases > GIBDD > Schemas > public > Tables > violation : 00:00:00.168 Ln 1, Col 1

2.2)UPDATE

pgAdmin 4

File Object Tools Help

Object Explorer

- Violation_types
- car_owner
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- registered_car
- violation
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Trigger Functions
- Types
- Views (3)
 - accident_participants_view_vers
 - violation_summary_remake2
 - Columns
 - Rules
 - Triggers
 - violation_summary_remake3
- Subscriptions
- emp_time
- postgres
- Login/Group Roles
- Tablespaces

GIBDD/postgres@PostgreSQL 16

Query Query History

```
1 WITH UpdateData AS (
2   SELECT
3     'E100BK37' AS "CarNumber",
4     0 AS "NewPaymentStatus"
5 )
6 UPDATE "violation"
7 SET
8   "Payment_status" = (SELECT "NewPaymentStatus" FROM UpdateData)
9 WHERE
10  "PTS_number" IN (SELECT "PTS_number" FROM "registered_car" WHERE "Car_number" = (SELECT "CarNumber" FROM U
11
```

Data Output Messages

UPDATE 3

Query returned successfully in 285 msec.

✓ Query returned successfully in 285 msec. ✕

Total rows: 0 of 0 Query complete 00:00:00.285 Ln 4, Col 6

pgAdmin 4

File Object Tools Help

Object Explorer

- Violation_types
- car_owner
- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- registered_car
- violation
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Trigger Functions
- Types
- Views (3)
 - accident_participa
 - violation_summary
 - Columns
 - Rules
 - Triggers
- Subscriptions
- emp_time
- postgres
- Login/Group Roles
- Tablespaces

public.violation/GIBDD/postgres@PostgreSQL 16

Query Query History

```

1 SELECT * FROM public.violation
2 ORDER BY "Violation_ID" ASC

```

Data Output Messages

ID	PTS_number	PersonalNumber	Violation_place	Violation_date	Violation_id	Violation_time	Payment_status
pt	character varying (18)	character varying (18)	character varying (255)	date	integer	time without time zone	integer
1	02KP362311	1	Загородный проспект 15	2023-11-26	1	05:03:00	0
2	02KP362311	2	Гражданский проспект 24	2023-11-26	2	04:21:00	0
3	03ET23145	1	Серебрястый бульвар 35	2023-11-23	2	21:00:00	1
4	04YE32664	2	Звенигородская улица 22	2023-11-26	1	05:03:21	1
5	02KP362311	1	Улица Ломоносова	2023-11-25	1	06:39:30	0
6	[null]	[null]	[null]	[null]	[null]	[null]	[null]

Total rows: 6 of 6 Query complete 00:00:00.681 Ln 1, Col 1

2.3)DELETE

pgAdmin 4

File Object Tools Help

Object Explorer

- Violation_types
- car_owner
- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- registered_car
- violation
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Trigger Functions
- Types
- Views (3)
 - accident_participa
 - violation_summary
 - Columns
 - Rules
 - Triggers
- Subscriptions
- emp_time
- postgres
- Login/Group Roles
- Tablespaces

GIBDD/postgres@PostgreSQL 16

Query Query History

```

1 WITH DeleteData AS (
2   SELECT
3     'E100BK37' AS "CarNumber",
4     '2023-11-25'::date AS "ViolationDate"
5 )
6 DELETE FROM "violation"
7 WHERE
8   "PTS_number" IN (SELECT "PTS_number" FROM "registered_car" WHERE "Car_number" = (SELECT "CarNumber" FROM DeleteData))
9   AND "Violation_date" = (SELECT "ViolationDate" FROM DeleteData);
10

```

Data Output Messages

DELETE 1

Query returned successfully in 318 msec.

Total rows: 0 of 0 Query complete 00:00:00.318 Ln 10, Col 1

✓ Query returned successfully in 318 msec. ✕

pgAdmin 4

File Object Tools Help

Object Explorer

- Violation_types
- car_owner
- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- registered_car
- violation
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Trigger Functions
- Types
- Views (3)
 - accident_participa
 - violation_summary
 - Columns
 - Rules
 - Triggers
 - violation_summary
- Subscriptions
- emp_time
- postgres
- Login/Group Roles
- Tablespaces

public.violation/GIBDD/postgres@PostgreSQL 16

Query

```

1 SELECT * FROM public.violation
2 ORDER BY "Violation_ID" ASC

```

Query History

Data Output

Violation_ID [PK] integer	PTS_number character varying (18)	Personal_number character varying (18)	Violation_place character varying (255)	Violation_date date	Violation_id integer	Violation_time time without time zone	Payment integer
1	02KP362311	1	Загородный проспект 15	2023-11-26	1	05:03:00	
2	02KP362311	2	Гражданский проспект 24	2023-11-26	2	04:21:00	
3	03ET23145	1	Серебряный бульвар 35	2023-11-23	2	21:00:00	
4	04YE32664	2	Звенигородская улица 22	2023-11-26	1	05:03:21	
5	[null]	[null]	[null]	[null]	[null]	[null]	

Ln 1, Col 1

3) Создание Представлений (View)

3.1) вывести данные водителей, который участвовали в аварии в текущем месяце.

pgAdmin 4

File Object Tools Help

Object Explorer

- RLS Policies
- Rules
- Triggers
- Violation_types
- car_owner
- Columns
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- registered_car
- violation
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Trigger Functions
- Types
- Views (3)
 - accident_participants_view_ver
 - violation_summary_remake2
 - violation_summary_remake3
- Subscriptions
- emp_time
- postgres
- Login/Group Roles

public.accident_participants_view/GIBDD/postgres@PostgreSQL 16

Query

```

5 SELECT CO."Address" AS "Driver_Address",
6        CO."Telephone_number" AS "Driver_Telephone",
7        CO."Date_of_birth" AS "Driver_Date_of_Birth",
8        CO."Passport" AS "Driver_Passport",
9        RC."Car_number" AS "Car_Number",
10       RC."Registration_date" AS "Car_Registration_Date",
11       PS."Participants_status" AS "Participant_Status",
12       A."Crash_date" AS "Accident_Date",
13       A."Crash_district" AS "Accident_District",
14       A."Crash_street" AS "Accident_Street"
15 FROM
16   "car_owner" CO
17 JOIN
18   "registered_car" RC ON CO."DL_number" = RC."DL_number"
19 JOIN
20   "Participants_status" PS ON RC."PTS_number" = PS."PTS_number"
21 JOIN
22   "Crash" A ON PS."Crash_ID" = A."Crash_ID"
23 WHERE
24   PS."Participants_status" IS NOT NULL
25   AND A."Crash_date" BETWEEN CURRENT_DATE - INTERVAL '1 month' AND CURRENT_DATE;;

```

Query History

Data Output

Driver_License_Number integer	Driver_Name_Surname character varying (255)	Driver_Address character varying (255)	Driver_Telephone character varying (15)	Driver_Date_of_Birth date	Driver_Passport character varying (15)
1	Иванов Иван Иванович	Серебряный бульвар 12	+79650897834	2001-12-03	4018 134534
2	Михайлов Иван Иванович	Проспект ветеранов к153	+79656578501	2001-03-21	4618 147216

Total rows: 2 of 2 Query complete 00:00:00.386

Ln 25, Col 84

3.2) содержащее следующие данные: номер водительского удостоверения, сумма штрафа за истекший год;

The screenshot shows the pgAdmin 4 interface. The left pane displays the database structure, with the 'violation_summary_remake2' table selected under the 'violation' schema. The main pane shows the query results for the query: `SELECT * FROM public.violation_summary_remake2`. The results are displayed in a table with two columns: 'DL_Number' (integer) and 'Total_Penalty' (bigint). The data is as follows:

DL_Number	Total_Penalty
1234567	800
1234565	0
1234566	800

Total rows: 3 of 3. Query complete 00:00:00.786.

4. Индексы

Запрос без индекса

The screenshot shows the pgAdmin 4 interface with the 'violation' schema selected. The query being executed is: `SELECT CO."DL_number", CO."Driver_name_surname", RC."Car_number", RC."Registration_date" FROM "car_owner" CO JOIN "registered_car" RC ON CO."DL_number" = RC."DL_number";`. The 'Explain' tab is active, showing the execution plan. The plan indicates a 'Hash Inner Join' operation, where the 'car_owner' table is hashed and then joined with the 'registered_car' table. The graphical representation shows the 'car_owner' table being hashed and then joined with the 'registered_car' table, which is also hashed. The results show 1 row and the query completed in 00:00:00.382.

Создание Индекса

The screenshot shows the pgAdmin 4 interface. In the Object Explorer on the left, the 'violation' table is selected. The main query editor displays the following SQL commands:

```
1 CREATE INDEX idx_car_owner_dl_number ON "car_owner"("DL_number");
2 CREATE INDEX idx_registered_car_dl_number ON "registered_car"("DL_number");
3
4
```

The 'Messages' tab at the bottom shows the execution result: 'CREATE INDEX' and 'Query returned successfully in 124 msec.' The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.124'.

Запрос с индексом

The screenshot shows the pgAdmin 4 interface with a query that joins the 'car_owner' and 'registered_car' tables. The SQL query is:

```
1 SELECT
2   CO."DL_number",
3   CO."Driver_name_surname",
4   RC."Car_number",
5   RC."Registration_date"
6 FROM
7   "car_owner" CO
8 JOIN
9   "registered_car" RC ON CO."DL_number" = RC."DL_number";
10
11
12
```

The 'Data Output' tab shows the results of the query:

	DL_number integer	Driver_name_surname character varying (255)	Car_number character varying (20)	Registration_date date
1	1234567	Иванов Иван Иванович	E100BK37	2023-03-11
2	1234565	Михайлов Иван Иванович	Y187EK37	2023-03-12
3	1234566	Сергеев Иван Иванович	O716CA178	2023-03-15

The status bar at the bottom indicates 'Total rows: 3 of 3' and 'Query complete 00:00:00.083'.

Составной индекс

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers (1)
 - PostgreSQL 16
 - Databases (3)
 - GIBDD
 - Aggregates
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages (1)
 - plpgsql
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)
 - Car
 - Car_model

Dashboard Properties SQL Statistics Dependencies Dependents Processes GIBDD/postgres@PostgreSQL 16*

Query

```

4      COUNT(V."Violation_id") AS "Penalty_Count"
5  FROM
6      "violation" V
7  JOIN
8      "Violation_types" VT ON V."Violation_id" = VT."Violation_ID"
9  WHERE
10     V."Payment_status" = '1'
11  GROUP BY
12     VT."Violation_name", V."Violation_id"
13  ORDER BY
14     "Penalty_Count" DESC;
15

```

Data Output

	Most_Frequently_Penalized_Violation character varying (500)	Most_Frequently_Penalized_Violation_ID integer	Penalty_Count bigint
1	Передача управления транспортным средством лицу, не имеющему при себе документов на право управления ...		2
2	Управление транспортным средством, не зарегистрированным в установленном порядке		1

Notifications

Recorded time	Event	Process ID
---------------	-------	------------

Total rows: 2 of 2 Query complete 00:00:00.114

PostgreSQL 16/GIBDD - Database connected.

Ln 8, Col 7

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers (1)
 - PostgreSQL 16
 - Databases (3)
 - GIBDD
 - Aggregates
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages (1)
 - plpgsql
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (11)
 - Car
 - Car_model

Dashboard Properties SQL Statistics Dependencies Dependents Processes GIBDD/postgres@PostgreSQL 16*

Query

```

1  CREATE INDEX violation_payment_index ON "violation" ("Violation_id", "Payment_status");
2  CREATE INDEX violation_types_index ON "Violation_types" ("Violation_ID");

```

Data Output

CREATE INDEX

Query returned successfully in 99 msec.

Notifications

Recorded time	Event	Process ID
---------------	-------	------------

Total rows: 2 of 2 Query complete 00:00:00.099

PostgreSQL 16/GIBDD - Database connected.

Ln 2, Col 41

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure for PostgreSQL 16, including the 'public' schema. The main pane shows a SQL query in the 'Query History' tab:

```

4  COUNT(V."Violation_id") AS "Penalty_Count"
5  FROM
6  "violation" V
7  JOIN
8  "Violation_types" VT ON V."Violation_id" = VT."Violation_ID"
9  WHERE
10 V."Payment_status" = '1'
11 GROUP BY
12 VT."Violation_name", V."Violation_id"
13 ORDER BY
14 "Penalty_Count" DESC;
15

```

The 'Data Output' tab shows the results of the query:

	Most_Frequently_Penalized_Violation character varying (500)	Most_Frequently_Penalized_Violation_ID integer	Penalty_Count bigint
1	Передача управления транспортным средством лицу, не имеющему при себе документов на право управления ...		2
2	Управление транспортным средством, не зарегистрированным в установленном порядке		1

The status bar at the bottom indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.084'. A green notification bar at the bottom right says 'PostgreSQL 16/GIBDD - Database connected.'.

The screenshot shows the same pgAdmin 4 interface, but with the 'EXPLAIN' button clicked. The 'Query History' tab still shows the same SQL query. The 'Data Output' tab is now replaced by the 'EXPLAIN' plan, which is visualized as a graph:

```

graph LR
    Violation[Violation] --> Hash[Hash]
    ViolationTypes[Violation_types] --> Hash
    Hash --> HashInnerJoin[Hash Inner Join]
    HashInnerJoin --> Sort1[Sort]
    Sort1 --> Aggregate[Aggregate]
    Aggregate --> Sort2[Sort]

```

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.191'. A green notification bar at the bottom right says 'PostgreSQL 16/GIBDD - Database connected.'.

Результаты запросов, показанные выше, указывают улучшение производительности запросов при использовании индексов. Сложные операции выборки данных и JOIN-запросы выполняются быстрее, что делает использование индексов желательным при работе с большим объемом данных.

Вывод

В ходе лабораторной работы были освоены практические навыки по созданию, запросов к базе данных в PostgreSQL с использованием инструмента управления pgAdmin 4. Были созданы запросы на выборку, обновление, вставку и удаление, а также представления и индексы а также проанализировано их влияние на скорость выполнения запросов. Результаты показали что запросы с индексами работают быстрее, чем без них.