

**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 Шестак Богдан Евгеньевич
Факультет прикладной информатики
Группа K3240
Направление подготовки 09.03.03 Прикладная информатика
Образовательная программа Мобильные и сетевые технологии 2025
Преподаватель Говорова Марина Михайловна

1. Цель работы:

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

2. Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных.
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 и сравнить время выполнения запросов без индексов и с индексами.

Индивидуальное задание:

Вариант 7. БД «Курсы»

Описание предметной области: Сеть учебных подразделений НОУ ДПО занимается организацией внебюджетного образования.

Имеется несколько образовательных программ краткосрочных курсов, предназначенных для определённых специальностей, связанных с программным обеспечением ИТ. Каждая программа имеет определённую длительность (в академических часах), свои перечни изучаемых дисциплин, вид итоговой аттестации, вид документа об окончании программы (сертификат о повышении квалификации, удостоверение о повышении квалификации, диплом о профессиональной подготовке).

Одна дисциплина может относиться к нескольким программам.

На каждую программу может быть набор из нескольких групп обучающихся.

По каждой дисциплине могут проводиться лекции, лабораторные/практические занятия и практика в определённом объёме часов. По каждой дисциплине и практике проводится аттестация в форме экзаменов/защиты/зачёта.

Необходимо хранить информацию по аттестации обучающихся.

Подразделение обеспечивает следующие ресурсы: учебные классы, лекционные аудитории и преподаватели. Необходимы ресурсы для расписания занятий.

В системе необходимо хранить информацию о количестве и номере выданного документа об окончании программы и дату выдачи.

БД должна содержать следующий минимальный набор сведений: Фамилия слушателя. Имя слушателя. Паспортные данные. Контакты. Код программы. Программа. Тип программы. Образовательное учреждение. Номер группы. Максимальное количество человек в группе (для программы). Образовательный курс. Номер сертификата/удостоверения. Название дисциплины. Количество часов. Дата начала. Дата окончания. Номер удостоверения (при наличии). Фамилия преподавателя. Имя преподавателя. Должность преподавателя. Должность дисциплины.


```

SELECT
    "Type_programm".name_type_prog AS program_name,
    COUNT("Sostav_group".id_group) AS number_of_students
FROM
    "Sostav_group"
JOIN
    "Type_programm"
    ON "Sostav_group".id_group = CAST("Type_programm".kod_type_programmi AS integer)
WHERE
    "Sostav_group".data_start_obuch >= CURRENT_DATE - INTERVAL '3 years'
GROUP BY
    "Type_programm".name_type_prog
ORDER BY
    number_of_students DESC
LIMIT 3;

```

Картинка 2 – Запрос на получение списка популярных программ за последние 3 года.

	program_name character varying (60)	number_of_students bigint
1	Engineering	2
2	Science	1

Картинка 3 – Результат выполнения запроса на выборку.

2. Создание представлений (Views)

Создание представления potential_listeners_specialties, согласно индивидуальному заданию, часть 3.

```

CREATE VIEW potential_listeners_specialties AS
SELECT
    "slushatel".fio AS student_fio,
    "Type_programm".name_type_prog AS program_name,
    "Disciplina".name_disciplini AS discipline_name,
    "Disciplina".kolvo_chasov AS hours,
    "Vidi_itogovoi_atestacii".name_itogov AS exam_type
FROM
    "slushatel"
LEFT JOIN
    "Sostav_group" ON "slushatel".id_slushatelia = "Sostav_group".id_slushatelia
LEFT JOIN
    "Sostav_uchebnogo_plana" ON CAST("Sostav_group".id_group AS text) = "Sostav_uchebnogo_plana".kod_sostav_uchebnogo_plana
LEFT JOIN
    "Disciplina" ON "Sostav_uchebnogo_plana".kod_disciplini = "Disciplina".kod_disciplini
LEFT JOIN
    "Type_programm" ON "Sostav_uchebnogo_plana".kod_uchebnogo_plana = "Type_programm".kod_type_programmi
LEFT JOIN
    "Vidi_itogovoi_atestacii" ON "Sostav_uchebnogo_plana".kod_vidov_itog_attestacii = "Vidi_itogovoi_atestacii".kod_vidov_itog_attestacii;

```

Картинка 6 – Создание представления (Views)

Далее через SELECT * FROM CREATE VIEW potential_listeners_specialties выводим результат – само представление.

	student_fio character varying (60)	program_name character varying (60)	discipline_name character varying (50)	hours integer
1	Petrov Petr Petrovich	Engineering	Mathematics	60
2	Ivanov Ivan Ivanovich	Engineering	Mathematics	60
3	Lobanov Semen Semenovich	Science	Physics	50

Картинка 7 – Представление potential_listeners_specialties

3. Модификация данных (INSERT, UPDATE, DELETE с подзапросами)

INSERT

Перед выполнением запроса выведем таблицу slushatel.

	id_slushatelia [PK] integer	fio character varying (60)	passport_danie character varying (20)	email character varying (20)
1	1	Ivanov Ivan Ivanovich	1234 567890	ivanov@mail.ru
2	2	Petrov Petr Petrovich	2345 678901	petrov@mail.ru

Картинка 8 – Таблица slushatel до изменений

Запрос	История запросов
1	INSERT INTO slushatel (fio, passport_danie, email)
2	SELECT 'Lobanov Semen Semenovich', '3456 789012', 'lobanov@mail.ru'
3	WHERE NOT EXISTS (
4	SELECT 4 FROM slushatel WHERE passport_danie = '3456 789012' OR email = 'lobanov@mail.ru'
5);
6	

Картинка 9 – Добавление нового Слушателя

	id_slushatelia [PK] integer	fio character varying (60)	passport_danie character varying (20)	email character varying (20)
1	1	Ivanov Ivan Ivanovich	1234 567890	ivanov@mail.ru
2	2	Petrov Petr Petrovich	2345 678901	petrov@mail.ru
3	3	Lobanov Semen Semenovich	3456 789012	lobanov@mail.ru

Картинка 10 – Таблица slushatel после изменений

UPDATE

Перед выполнением запроса выведем таблицу prepodavarel.

	id_prepod [PK] integer	fio character varying (60)
1	1	Sidorov Alexei Petrovich
2	2	Kuznetsov Vitalii Nikolaevich

Картинка 11 – Таблица prepodavarel до изменений

```
UPDATE prepodavatel
SET fio = 'Pravdin Konstantin Vladimirovich'
WHERE id_prepod = 2 AND fio = (SELECT fio FROM prepodavatel WHERE id_prepod = 2);
```

Картинка 12 – Обновление ФИО у второго преподавателя

	id_prepod [PK] integer	fio character varying (60)
1	1	Sidorov Alexei Petrovich
2	2	Pravdin Konstantin Vladimirovich

Картинка 13 – Таблица prepodavatel после изменений

DELETE

Перед выполнением запроса выведем таблицу auditoria.

	nomer_auditorii [PK] character varying (5)	kod_podrazdelenia character varying (18)	id_korpusa integer
1	4208	POD001	1
2	4318	POD001	2
3	1124	POD002	2

Картинка 14 – Таблица auditoria до изменений

```
DELETE FROM auditoria
WHERE nomer_auditorii = '1124' AND NOT EXISTS (
    SELECT 1 FROM type_auditorii WHERE type_auditorii.id_type_auditorii = auditoria.id_korpusa
);|
```

Картинка 15 – Удаление аудитории

	nomer_auditorii [PK] character varying (5)	kod_podrazdelenia character varying (18)	id_korpusa integer
1	4208	POD001	1
2	4318	POD001	2

Картинка 16 – Таблица auditoria после изменений

4. Создание индексов и анализ планов выполнения

Запрос без индекса

Выполняем запрос поиска слушателей без использования индекса.

```
EXPLAIN ANALYZE
SELECT *
FROM slushatel
WHERE fio = 'Ivanov Ivan Ivanovich';
```

Картинка 17 – Запрос без индекса

	QUERY PLAN
	text
1	Seq Scan on slushatel (cost=0.00..1.04 rows=1 width=258) (actual time=0.012..0.013 rows=1 loops=...
2	Filter: ((fio)::text = 'Ivanov Ivan Ivanovich')::text)
3	Rows Removed by Filter: 2
4	Planning Time: 0.078 ms
5	Execution Time: 0.024 ms

Картинка 18 – План выполнения запроса без индекса

Запрос с индексом

Создаем индекс на столбец fio в таблице slushatel.

```
CREATE INDEX idx_fio ON slushatel(fio);
```

Картинка 19 – Создание индекса

Выполняем тот же запрос после создания индекса и проверяем.

	QUERY PLAN
	text
1	Seq Scan on slushatel (cost=0.00..1.04 rows=1 width=258) (actual time=0.007..0.008 rows=1 loops=...
2	Filter: ((fio)::text = 'Lobanov Semen Semenovich')::text)
3	Rows Removed by Filter: 2
4	Planning Time: 0.766 ms
5	Execution Time: 0.017 ms

Картинка 20 – План выполнения запроса с индексом

```
DROP INDEX IF EXISTS idx_fio;
```

Картинка 21 – Удаление индекса

Сравнение времени выполнения

По результатам EXPLAIN видно, что после создания индекса время выполнения запроса уменьшилось. - Без индекса: [0.024 ms] - С индексом: [0.017 ms].

Таким образом, использование индексов позволяет значительно ускорить выполнение запросов, особенно при фильтрации по часто используемым столбцам.

4. Вывод

В процессе выполнения лабораторной работы были разработаны запросы для выборки и изменения данных, созданы представления, а также проведен анализ планов выполнения запросов с использованием индексов. Результаты подтвердили, что применение индексов значительно ускоряет выполнение запросов, повышая их эффективность.