

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет
По Лабораторной работе 2
ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В
POSTGRESQL
по дисциплине «Базы данных»

Автор:.. Никитин.П

Факультет: ИКТ

Группа: K32402

Преподаватель: Говорова М. М.

Дата: 22.03.2023

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

Программное обеспечение: СУБД PostgreSQL 1X, 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.

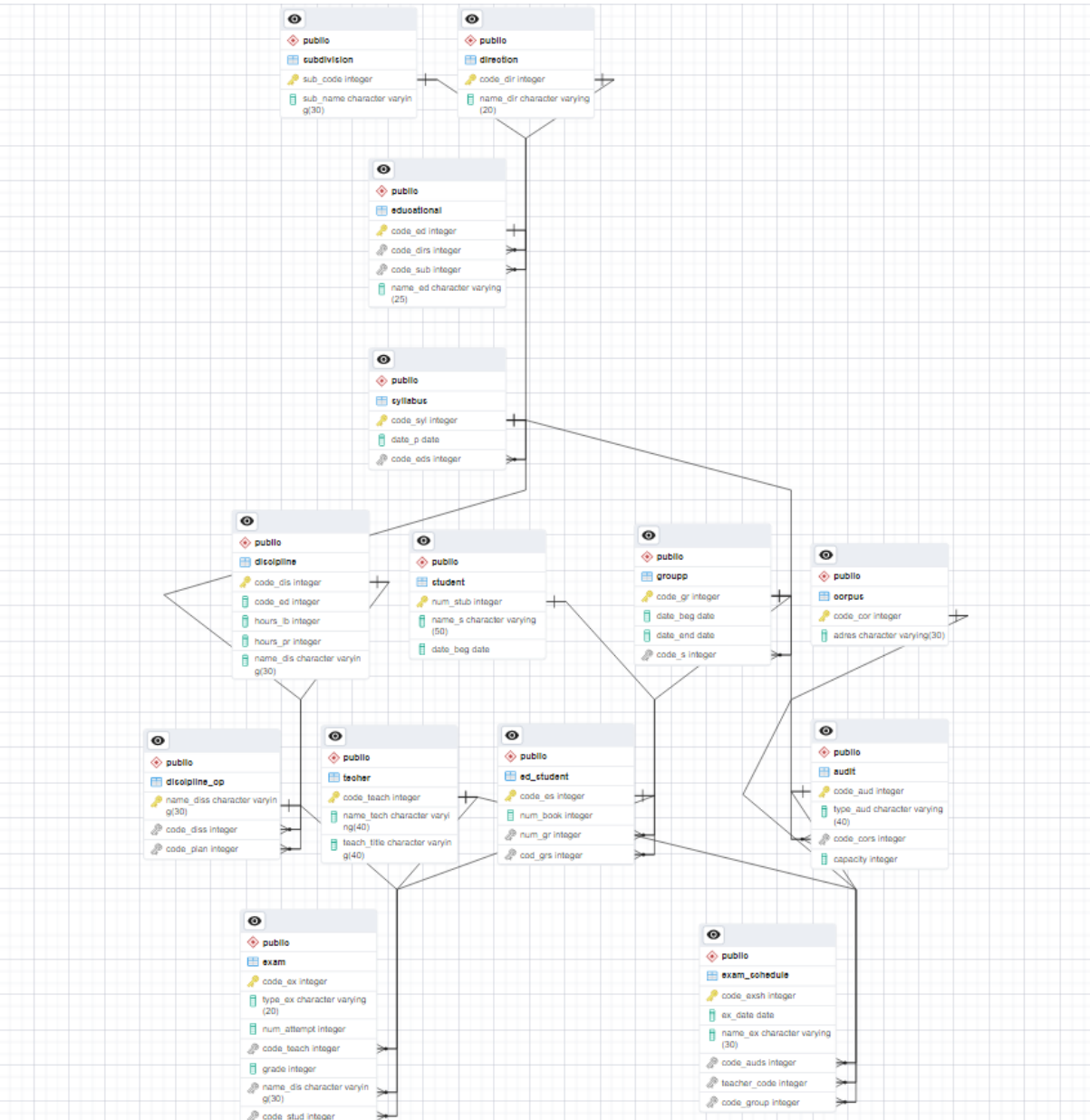
Индивидуальное практическое задание:

Предметная область «**Банк**».

Составьте запросы на выборку:

- Составить список дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей.
- Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.
- Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине.
- Создать рейтинговый список групп по заданному направлению по результатам сдачи сессии, упорядочить его по убыванию.
- Создайте списки студентов, упорядоченные по группам и фамилиям студентов, содержащие данные о средних баллах и назначении на стипендии. Студент получает стипендию, если он сдал сессию без троек. Если студент не назначен на стипендию, указать 0, если назначен – 1.
- Вывести список студентов, сдавших все положенные экзамены.
- Вывести список студентов, получивших максимальный средний балл в своей группе.

II. схема логической модели базы данных, сгенерированная в Generate ERD



SELECT

1 Составить список дисциплин, которые должны быть сданы заданной группой с указанием дат сдачи и фамилий преподавателей

```
SELECT
    name_ex, ex_date, teacher_code
FROM
    exam_schedule
WHERE
    code_group = 121
ORDER BY
    exam_schedule.ex_date ASC;
```

	name_ex character varying (30)	ex_date date	teacher_code integer
1	Математика	2023-06-21	1

2 Вывести список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

```
SELECT name_s, grade, num_attempt, code_teach FROM exam
JOIN ed_student ON exam.code_stud = ed_student.code_es
JOIN student ON ed_student.num_gr = student.num_stub
WHERE num_attempt = 1 AND grade = 2;
```

	name_s character varying (50)	grade integer	num_attempt integer	code_teach integer
1	Sam	2	1	1

3 Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине

```
SELECT name_s FROM student
JOIN ed_student ON student.num_stub = ed_student.num_gr
WHERE code_es IN (SELECT code_stud FROM exam
WHERE name_dis = 'Математика'
AND grade > (SELECT AVG(grade) FROM exam WHERE name_dis = 'Математика'));
```

	name_s character varying (50)
1	S

4 Создать рейтинговый список групп по заданному направлению по результатам сдачи сессии, упорядочить его по убыванию

```
SELECT ed_student.cod_grs, AVG(exam.grade) AS average_mark
FROM exam
JOIN ed_student ON exam.code_stud = ed_student.code_es
WHERE name_dis = 'Математика'
GROUP BY cod_grs
ORDER BY average_mark DESC;
```

	cod_grs integer	average_mark numeric
1	121	3.0000000000000000

5 Создайте списки студентов, упорядоченные по группам и фамилиям студентов, содержащие данные о средних баллах и назначении на стипендии. Студент получает

стипендию, если он сдал сессию без троек. Если студент не назначен на стипендию, указать 0, если назначен – 1.

```
SELECT
    ed_student.cod_grs,
    code_stud,
    AVG(grade) AS average_mark,
    CASE
        WHEN COUNT(CASE WHEN grade < 3 THEN 1 END) > 0
        THEN 0
        ELSE 1
    END AS scholarship
FROM
    exam
    join ed_student on exam.code_stud = ed_student.code_es
GROUP BY
    cod_grs,
    code_stud
ORDER BY
    cod_grs ASC;
```

	cod_grs integer	code_stud integer	average_mark numeric	scholarship integer
1	121	2	3.0000000000000000	0

6 Вывести список студентов, сдавших все положенные экзамены

```
SELECT ed_student.code_es
FROM ed_student
WHERE NOT EXISTS
(SELECT * FROM exam
WHERE exam.code_stud = ed_student.code_es
AND exam.grade IS NULL);
```

	code_es [PK] integer
1	2

7 Вывести список студентов, получивших максимальный средний балл в своей группе.

```

SELECT s.code_stud, g.cod_grs AS group_name, AVG(s.grade) AS avg_mark
FROM exam s
JOIN ed_student g ON s.code_stud = g.code_es
group by s.code_stud, g.cod_grs
HAVING AVG(s.grade) = (
    SELECT MAX(avg_mark)
    FROM (
        SELECT AVG(grade) AS avg_mark
        FROM exam
        GROUP BY code_stud
    ) AS avg_marks
);

```


	code_stud integer	group_name integer	avg_mark numeric
1	2	121	3.0000000000000000


View


```
CREATE VIEW matematik AS  
SELECT code_tech, name_tech  
FROM techer  
WHERE teach_title = 'Mat';|
```


```
SELECT * FROM matematik;
```


Output Сообщения Notifications



















code_tech		name_tech				
integer		character varying (40)				
1		Николай				

INSERT, UPDATE, DELETE INSERT

```
INSERT INTO student (num_stub, name_s, date_beg)  
VALUES (10, 'John Smith', '01-09-2023'),  
       (11, 'Jane Doe', '01-09-2023'),  
       (12, 'Bob Johnson', '01-09-2023'),  
       (13, 'Emily Davis', '01-09-2023');
```

```
UPDATE student
SET date_beg = '15-09-2023'
WHERE date_beg = '01-09-2023';
```

```
DELETE from student where num_stub=10;
```

```
select * from student;
```

	num_stub [PK] integer	name_s character varying (50)	date_beg date
1	5	H	2023-09-15
2	4	G	2023-09-15
3	3	T	2023-09-15
4	2	b	2023-09-15
5	12	Bob Johnson	2023-09-15
6	13	Emily Davis	2023-09-15
7	1	Sam	2023-09-15

Выводы:

В процессе работы с pgAdmin я научился делать запросы на выборку и модификацию данных в базе данных PostgreSQL. Я освоил создание представлений, которые помогают упростить работу с данными, а также настройку индексов, что повышает производительность работы с базой данных. Эти знания помогут мне более эффективно работать с PostgreSQL и создавать более сложные запросы и представления для обработки большого объема данных. В целом, работа с pgAdmin позволила мне более полно использовать возможности PostgreSQL и повысить эффективность работы с базой данных.