

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

Отчёт

по лабораторной работе №4 «Запросы на выборку и модификацию данных.
Представления. Работа с индексами»
по дисциплине «Проектирование и реализация баз данных»
Вариант №2 - БД «Сессия»

Автор: Клименков В.М.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

1. Запросы к базе данных	3
2. Представления	8
3. Запросы на модификацию данных	9
4. Query History и графическое представление запросов	11
5. Индексы	12
Вывод	12

Цель работы

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

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

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

```
SELECT
    d.d_name AS discipline_name,
    t.t_last_name AS teacher_last_name,
    ss.ss_date AS discipline_delivery_date
FROM public.session_schedule ss
JOIN public.disciplines_in_curricula dic ON
ss.ss_discipline_in_curriculum_code = dic.dic_code
JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
JOIN public.teachers t ON ss.ss_teacher_id = t.t_id
WHERE ss.ss_study_group_code = <заданный код группы>;
```

Query Query History

```
1 SELECT
2     d.d_name AS discipline_name,
3     t.t_last_name AS teacher_last_name,
4     ss.ss_date AS discipline_delivery_date
5 FROM public.session_schedule ss
6 JOIN public.disciplines_in_curricula dic ON ss.ss_discipline_in_curriculum_code = dic.dic_code
7 JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
8 JOIN public.teachers t ON ss.ss_teacher_id = t.t_id
9 WHERE ss.ss_study_group_code = 1;
```

Data Output Messages Notifications

	discipline_name character varying (50)	teacher_last_name character varying (50)	discipline_delivery_date date
1	История российской науки и техники	Касаткина	2022-01-01
2	Программирование	Соболева	2022-10-16
3	История российской науки и техники	Касаткина	2022-10-13
4	Иностранный язык	Морозова	2022-10-19

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

```
SELECT
    a.a_student_record_book AS student_record_book,
    s.s_first_name AS student_first_name,
    s.s_last_name AS student_last_name,
    t.t_last_name AS teacher_last_name
FROM public.attestations a
JOIN public.students s ON a.a_student_record_book = s.s_record_book
JOIN public.teachers t ON a.a_teacher_id = t.t_id
WHERE a.a_attempt = 1 AND a.a_grade = '2';
```

Query		Query History		
1	SELECT			
2	a.a_student_record_book AS student_record_book,			
3	s.s_first_name AS student_first_name,			
4	s.s_last_name AS student_last_name,			
5	t.t_last_name AS teacher_last_name			
6	FROM public.attestations a			
7	JOIN public.students s ON a.a_student_record_book = s.s_record_book			
8	JOIN public.teachers t ON a.a_teacher_id = t.t_id			
9	WHERE a.a_attempt = 1 AND a.a_grade = '2';			

Data Output		Messages			Notifications	
	student_record_book	student_first_name	student_last_name	teacher_last_name		
	integer	character varying (50)	character varying (50)	character varying (50)		
1	381030	Арина	Круглова	Николаев		
2	516111	Моника	Егорова	Соболева		

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

```
SELECT s.s_last_name AS student_last_name
FROM public.students s
JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
JOIN public.disciplines_in_curricula dic ON a.a_discipline_in_curricula_code = dic.dic_code
JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
WHERE d.d_code = <заданный_код_дисциплины>
AND a.a_grade > (
    SELECT AVG(a_avg.a_grade)
    FROM public.attestations a_avg
    JOIN public.disciplines_in_curricula dic_avg ON
a_avg.a_discipline_in_curricula_code = dic_avg.dic_code
    JOIN public.disciplines d_avg ON dic_avg.dic_discipline_code =
d_avg.d_code
    WHERE d_avg.d_code = <заданный_код_дисциплины>
);
```

Query Query History

```

1 SELECT s.s_last_name AS student_last_name
2 FROM public.students s
3 JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
4 JOIN public.disciplines_in_curricula dic ON a.a_discipline_in_curricula_code = dic.dic_code
5 JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
6 WHERE d.d_code = 30185
7 AND a.a_grade > (
8     SELECT AVG(a_avg.a_grade)
9     FROM public.attestations a_avg
10    JOIN public.disciplines_in_curricula dic_avg ON a_avg.a_discipline_in_curricula_code = dic_avg.dic_code
11    JOIN public.disciplines d_avg ON dic_avg.dic_discipline_code = d_avg.d_code
12    WHERE d_avg.d_code = 30185
13 );

```

Data Output Messages Notifications

	student_last_name character varying (50)
1	Сорокин
2	Косарев

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

```

SELECT sg.sg_code AS study_group_code
FROM public.study_groups sg
JOIN public.curricula c ON sg.sg_curriculum_code = c.c_code
JOIN public.educational_programmes ep ON c.c_educational_program_code = ep.ep_code
JOIN public.directions d ON ep.ep_direction_code = d.d_code
JOIN public.students_in_study_groups sisg ON sg.sg_code = sisg.sisg_study_group_code
JOIN public.students s ON sisg.sisg_student_record_book = s.s_record_book
JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
WHERE d.d_code = <заданный_код_направления>
GROUP BY sg.sg_code
ORDER BY AVG(a.a_grade) DESC;

```

Query Query History

```

1 SELECT sg.sg_code AS study_group_code
2 FROM public.study_groups sg
3 JOIN public.curricula c ON sg.sg_curriculum_code = c.c_code
4 JOIN public.educational_programmes ep ON c.c_educational_program_code = ep.ep_code
5 JOIN public.directions d ON ep.ep_direction_code = d.d_code
6 JOIN public.students_in_study_groups sisg ON sg.sg_code = sisg.sisg_study_group_code
7 JOIN public.students s ON sisg.sisg_student_record_book = s.s_record_book
8 JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
9 WHERE d.d_code = 2
10 GROUP BY sg.sg_code
11 ORDER BY AVG(a.a_grade) DESC;

```

Data Output Messages Notifications

	study_group_code integer
1	3
2	1
3	2

- 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
    s.s_record_book AS student_record_book,
    s.s_first_name AS student_first_name,
    s.s_last_name AS student_last_name,
    sigs.sig_study_group_code AS study_group_code,
    AVG(a.a_grade) AS average_grade,
    CASE WHEN MIN(a.a_grade) > 3 THEN 1 ELSE 0 END AS receives_scholarship
FROM public.students s
JOIN public.students_in_study_groups sigs ON s.s_record_book =
sigs.sig_student_record_book
JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
GROUP BY
    s.s_record_book,
    s.s_first_name,
    s.s_last_name,
    sigs.sig_study_group_code
ORDER BY
    sigs.sig_study_group_code,
    s.s_last_name;
```

Query Query History

```
1 SELECT
2     s.s_record_book AS student_record_book,
3     s.s_first_name AS student_first_name,
4     s.s_last_name AS student_last_name,
5     sigs.sig_study_group_code AS study_group_code,
6     AVG(a.a_grade) AS average_grade,
7     CASE WHEN MIN(a.a_grade) > 3 THEN 1 ELSE 0 END AS receives_scholarship
8 FROM public.students s
9 JOIN public.students_in_study_groups sigs ON s.s_record_book = sigs.sig_student_record_book
10 JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
11 GROUP BY
12     s.s_record_book,
13     s.s_first_name,
14     s.s_last_name,
15     sigs.sig_study_group_code
16 ORDER BY
17     sigs.sig_study_group_code,
18     s.s_last_name;
```

Data Output Messages Notifications

	student_record_book integer	student_first_name character varying (50)	student_last_name character varying (50)	study_group_code integer	average_grade numeric	receives_scholarship integer
1	516111	Моника	Егорова	1	2.5000000000000000	0
2	742645	Лука	Семенов	1	4.0000000000000000	1
3	400419	Иван	Чернышев	1	4.5000000000000000	1
4	252108	Григорий	Косарев	2	5.0000000000000000	1
5	381030	Арина	Круглова	2	3.5000000000000000	0

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

```
SELECT
    s.s_record_book AS student_record_book,
    s.s_first_name AS student_first_name,
    s.s_last_name AS student_last_name
FROM public.students s
JOIN public.students_in_study_groups sigs ON s.s_record_book =
sigs.sigs_student_record_book
JOIN (
    SELECT
        a.a_student_record_book AS r_student_record_book,
        RANK() OVER (PARTITION BY sigs.sigs_study_group_code ORDER BY
AVG(a.a_grade) DESC) AS rank
    FROM public.attestations a
    JOIN public.students_in_study_groups sigs ON a.a_student_record_book =
sigs.sigs_student_record_book
    GROUP BY a.a_student_record_book, sigs.sigs_study_group_code
) AS r ON s.s_record_book = r.r_student_record_book
WHERE r.rank = 1
```

Query Query History

```
1 SELECT
2     s.s_record_book AS student_record_book,
3     s.s_first_name AS student_first_name,
4     s.s_last_name AS student_last_name
5 FROM public.students s
6 JOIN public.students_in_study_groups sigs ON s.s_record_book = sigs.sigs_student_record_book
7 JOIN (
8     SELECT
9         a.a_student_record_book AS r_student_record_book,
10        RANK() OVER (PARTITION BY sigs.sigs_study_group_code ORDER BY AVG(a.a_grade) DESC) AS rank
11    FROM public.attestations a
12    JOIN public.students_in_study_groups sigs ON a.a_student_record_book = sigs.sigs_student_record_book
13    GROUP BY a.a_student_record_book, sigs.sigs_study_group_code
14 ) AS r ON s.s_record_book = r.r_student_record_book
15 WHERE r.rank = 1
```

Data Output Messages Notifications

	student_record_book integer	student_first_name character varying (50)	student_last_name character varying (50)
1	252108	Григорий	Косарев
2	400419	Иван	Чернышев
3	534421	Пётр	Смирнов

2. Представления

- Список студентов, получивших двойки на первой попытке с указанием фамилии преподавателя, которым они должны пересдать экзамен.

```
CREATE VIEW retake AS
SELECT
    s.s_record_book AS student_record_book,
    s.s_first_name AS student_first_name,
    s.s_last_name AS student_last_name,
    t.t_last_name AS teacher_last_name
FROM public.attestations a
JOIN students s ON a.a_student_record_book = s.s_record_book
JOIN teachers t ON a.a_teacher_id = t.t_id
WHERE a.a_attempt = 1 AND a.a_grade = '2';
```

Query

Query History

1

SELECT * FROM public.retake

Data Output

Messages

Notifications

	<div>student_record_book</div> <div>integer</div>	<div>student_first_name</div> <div>character varying (50)</div>	<div>student_last_name</div> <div>character varying (50)</div>	<div>teacher_last_name</div> <div>character varying (50)</div>
1	381030	Арина	Круглова	Николаев
2	516111	Моника	Егорова	Соболева

- Данных о студентах при получении ими хотя бы одной оценки 2 (после 3-й попытки).

```
CREATE VIEW unattested_students AS
SELECT
    s.s_record_book AS student_record_book,
    s.s_first_name AS student_first_name,
    s.s_last_name AS student_last_name
FROM public.students s
JOIN public.attestations a ON s.s_record_book = a.a_student_record_book
WHERE a.a_attempt = 3 AND a.a_grade = '2'
GROUP BY s.s_record_book, s.s_first_name, s.s_last_name;
```

Query

Query History

1

SELECT * FROM public.unattested_students

Data Output

Messages

Notifications

<

3. Запросы на модификацию данных

- Вставка нового экзамена в расписание сессии по номеру аудитории и имени и фамилии преподавателя.

```
INSERT INTO public.session_schedule (  
    ss_study_group_code,  
    ss_discipline_in_curriculum_code,  
    ss_date,  
    ss_auditorium_id,  
    ss_teacher_id,  
    ss_id,  
    ss_pair_number)  
VALUES (  
    (SELECT sg.sg_code FROM public.study_groups sg  
    WHERE sg.sg_number = 'A3610' LIMIT 1),  
    4526,  
    '2022-10-19',  
    1,  
    (SELECT t.t_id FROM public.teachers t  
    WHERE t.t_first_name = 'Александра' AND t.t_last_name = 'Морозова' LIMIT  
1),  
    10,  
    2  
);
```

Query Query History

```
1 SELECT * FROM public.session_schedule  
2 ORDER BY ss_id DESC
```

Data Output Messages Notifications

	ss_study_group_code integer	ss_discipline_in_curriculum_code integer	ss_date date	ss_auditorium_id integer	ss_teacher_id integer	ss_id [PK] integer	ss_pair_number integer
1	1	4526	2022-10-19	1	2	10	2
2	3	8925	2022-10-18	5	5	9	2
3	2	7856	2022-10-17	5	5	8	1
4	1	4698	2022-10-16	4	4	7	4

- Изменение оценки студента по его имени и фамилии, а также названию предмета.

```
UPDATE public.attestations  
SET a_grade = 5  
WHERE a.student_record_book = (  
    SELECT s.s_record_book  
    FROM public.students s  
    WHERE s.s_first_name = 'Пётр' AND s.s_last_name = 'Смирнов'  
    LIMIT 1  
) AND a.discipline_in_curricula_code = (  
    SELECT dic.dic_code  
    FROM public.disciplines_in_curricula dic  
    JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code  
    WHERE d.d_name = 'Общая физика'  
    LIMIT 1  
);
```

Query Query History	
1	SELECT * FROM public.attestations
2	ORDER BY a_id DESC

Data Output Messages Notifications	
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	
a_discipline_in_curricula_code integer	a_date date
1	7856 2022-10-23
2	4698 2022-10-22
3	8925 2022-10-21
4	4526 2022-10-20

- Удаление аудитории по её номеру и площадке.

```
DELETE FROM public.auditoriums
WHERE a_id = (
  SELECT a.a_id
  FROM public.auditoriums a
  WHERE a.a_site_name = 'Биржевая линия' AND a.a_number = 5478
  LIMIT 1
);
```

Query Query History	
1	SELECT * FROM public.auditoriums
2	ORDER BY a_id ASC

Data Output Messages Notifications	
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	
a_id [PK] integer	a_site_name character varying (50)
1	1 Кронверкский проспект
2	2 Улица Ломоносова
3	3 Переулок Гривцова
4	4 Улица Чайковского
5	5 Биржевая линия

4. Query History и графическое представление запросов

- Query History

Query Query History

Show queries generated internally by pgAdmin? ☒

Remove Remove All

Today - 25.11.2023

▶ SELECT s.s_record_book AS student_record_book, s.s_...
19:12:48

▶ SELECT s.s_record_book AS student_record_book, s.s_...
19:10:56

▶ SELECT sg.sg_code AS study_group_code FROM public.s...
19:09:43

▶ SELECT sg.sg_code AS study_group_code FROM public.s...
19:09:39

▶ SELECT s.s_last_name AS student_last_name FROM publ...
19:07:00

▶ SELECT s.s_last_name AS student_last_name FROM publ...
19:06:49

▶ SELECT s.s_last_name AS student_last_name FROM publ...
19:06:26

▶ SELECT a.a_student_record_book AS student_record_bo...
19:03:57

▶ SELECT d.d_name AS discipline_name, t.t_last_name A...
19:01:41

▶ SELECT d.d_name AS discipline_name, t.t_last_name A...

25.11.2023 19:12:48 3 142 msec

Date Rows affected Duration

Copy Copy to Query Editor

```
SELECT
  s.s_record_book AS student_record_book,
  s.s_first_name AS student_first_name,
  s.s_last_name AS student_last_name
FROM public.students s
JOIN public.students_in_study_groups sigs ON s.s_record_
JOIN (
  SELECT
    a.a_student_record_book AS r_student_record_bool
    RANK() OVER (PARTITION BY sigs.sigs_study_group
FROM public.attestations a
JOIN public.students_in_study_groups sigs ON a.a_sti
GROUP BY a.a_student_record_book, sigs.sigs_study_g
) AS r ON s.s_record_book = r.r_student_record_book
WHERE r.rank = 1
```

Messages
Successfully run. Total query runtime: 142 msec. 3 rows affected.

- Графическое представление запроса

Query Query History

1 SELECT
2 a.a_student_record_book AS student_record_book,
3 s.s_first_name AS student_first_name,
4 s.s_last_name AS student_last_name,
5 t.t_last_name AS teacher_last_name
6 FROM public.attestations a
7 JOIN public.students s ON a.a_student_record_book = s.s_record_book
8 JOIN public.teachers t ON a.a_teacher_id = t.t_id
9 WHERE a.a_attempt = 1 AND a.a_grade = '2';

Data Output Messages Explain × Notifications

Graphical Analysis Statistics

🔍 ↺ 🔍 ⬇

attestations

students_pkey

teachers

→

Nested Loop Inner Join

→

Nested Loop Inner Join

11

5. Индексы

```
CREATE INDEX idx_disciplines
  ON public.disciplines (d_name);
CREATE INDEX idx_teachers
  ON public.teachers (t_last_name);
CREATE INDEX idx_session_schedule
  ON public.session_schedule (ss_date, ss_study_group_code);
```

- Без индексов:

Query	Query History
1	SELECT
2	d.d_name AS discipline_name,
3	t.t_last_name AS teacher_last_name,
4	ss.ss_date AS discipline_delivery_date
5	FROM public.session_schedule ss
6	JOIN public.disciplines_in_curricula dic ON ss.ss_discipline_in_curriculum_code = dic.dic_code
7	JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
8	JOIN public.teachers t ON ss.ss_teacher_id = t.t_id
9	WHERE ss.ss_study_group_code = 1;

Data Output	Messages	Notifications
Successfully run. Total query runtime: 160 msec. 4 rows affected.		

- С индексами:

Query	Query History
1	SELECT
2	d.d_name AS discipline_name,
3	t.t_last_name AS teacher_last_name,
4	ss.ss_date AS discipline_delivery_date
5	FROM public.session_schedule ss
6	JOIN public.disciplines_in_curricula dic ON ss.ss_discipline_in_curriculum_code = dic.dic_code
7	JOIN public.disciplines d ON dic.dic_discipline_code = d.d_code
8	JOIN public.teachers t ON ss.ss_teacher_id = t.t_id
9	WHERE ss.ss_study_group_code = 1;

Data Output	Messages	Notifications
Successfully run. Total query runtime: 72 msec. 4 rows affected.		

Вывод

В данной лабораторной работе мне удалось создать требуемые запросы и представления к разработанной ранее базе данных, составить три запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов, изучить графическое представление запросов и просмотреть историю запросов, а также создать индексы для запроса и сравнить время выполнения запроса без индексов и с индексами.