

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 4
по теме: Запросы на выборку данных к БД PostgreSQL.
по дисциплине: Проектирование и реализация баз данных

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «__» _____ 2021 г.
Оценка _____

Выполнил:
студентка группы К3240
Бабан В.

Санкт-Петербург 2022 г.

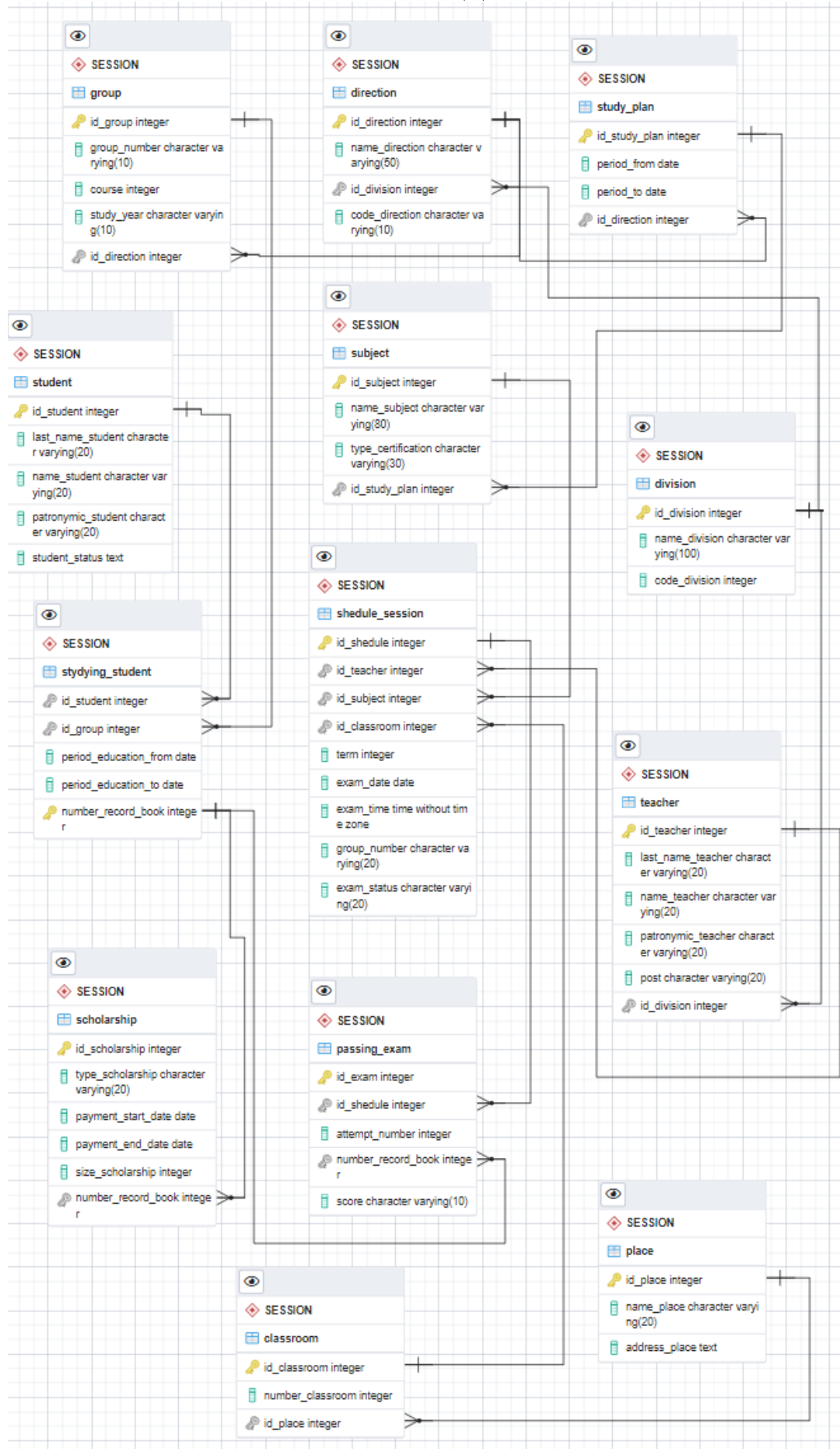
ЦЕЛЬ РАБОТЫ

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

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 посмотреть 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.

СХЕМА БАЗЫ ДАННЫХ



Задание 2. Запросы

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

Запрос:

```
SELECT exam_date, name_subject, last_name_teacher
FROM "SESSION".shedule_session
INNER JOIN "SESSION".subject USING (id_subject)
INNER JOIN "SESSION".teacher USING (id_teacher)
WHERE group_number = 'K3240'
```

Query Editor

```
1 SELECT exam_date, name_subject, last_name_teacher
2 FROM "SESSION".shedule_session
3 INNER JOIN "SESSION".subject USING (id_subject)
4 INNER JOIN "SESSION".teacher USING (id_teacher)
5 WHERE group_number = 'K3240'
```

Результат

	exam_date date	name_subject character varying (80)	last_name_teacher character varying (20)
1	2022-06-20	Проектирование и реализация баз данных	Говорова
2	2022-01-28	Базы данных	Говорова
3	2022-01-18	Математика	Блаженнов

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

Запрос:

```
SELECT last_name_student, name_student, patronymic_student, last_name_teacher
FROM "SESSION".passing_exam
INNER JOIN "SESSION".shedule_session USING (id_shedule)
INNER JOIN "SESSION".stydyng_student USING (number_record_book)
INNER JOIN "SESSION".student USING (id_student)
INNER JOIN "SESSION".teacher USING (id_teacher)
WHERE (attempt_number = 1) AND (score = '2')
```

Query Editor

```
1 SELECT last_name_student, name_student, patronymic_student, last_name_teacher
2 FROM "SESSION".passing_exam
3 INNER JOIN "SESSION".shedule_session USING (id_shedule)
4 INNER JOIN "SESSION".stydyng_student USING (number_record_book)
5 INNER JOIN "SESSION".student USING (id_student)
6 INNER JOIN "SESSION".teacher USING (id_teacher)
7 WHERE (attempt_number = 1) AND (score = '2')
8
```

Результат

	last_name_student character varying (20)	name_student character varying (20)	patronymic_student character varying (20)	last_name_teacher character varying (20)
1	Байков	Иван		Блаженков
2	Жуков	Дмитрий	Витальевич	Таранов

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

Запрос:

```
SELECT DISTINCT last_name_student FROM "SESSION".passing_exam
INNER JOIN "SESSION".studying_student USING (number_record_book)
INNER JOIN "SESSION".student USING (id_student)
WHERE (score NOT IN ('Зачёт', 'Незачёт')) AND
(CAST(score AS int) >
  (SELECT AVG(CAST(score AS int))
   FROM "SESSION".passing_exam
   INNER JOIN "SESSION".shedule_session USING (id_shedule)
   INNER JOIN "SESSION".subject USING (id_subject)
   WHERE name_subject = 'Математика'))
```

Query Editor

```
1 SELECT DISTINCT last_name_student FROM "SESSION".passing_exam
2 INNER JOIN "SESSION".studying_student USING (number_record_book)
3 INNER JOIN "SESSION".student USING (id_student)
4 WHERE (score NOT IN ('Зачёт', 'Незачёт')) AND
5 (CAST(score AS int) > (SELECT AVG(CAST(score AS int))
6                        FROM "SESSION".passing_exam
7                        INNER JOIN "SESSION".shedule_session USING (id_shedule)
8                        INNER JOIN "SESSION".subject USING (id_subject)
9                        WHERE name_subject = 'Математика'))
```

Результат

	last_name_student character varying (20)
1	Балдина
2	Бабан
3	Устин

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

Запрос:

```
SELECT name_direction, group_number, ROUND(total, 2)
FROM (SELECT group_number, AVG(CAST(score AS int)) as total
      FROM "SESSION".passing_exam
      INNER JOIN "SESSION".shedule_session USING (id_shedule)
      WHERE score NOT IN ('Зачёт', 'Незачёт')
      GROUP BY group_number) as groups
INNER JOIN "SESSION".group USING (group_number)
INNER JOIN "SESSION".direction USING (id_direction)
WHERE name_direction = 'Прикладная информатика'
ORDER BY total DESC
```

Query Editor

```
1 SELECT name_direction, group_number, ROUND(total, 2)
2 FROM (SELECT group_number, AVG(CAST(score AS int)) as total
3      FROM "SESSION".passing_exam
4      INNER JOIN "SESSION".shedule_session USING (id_shedule)
5      WHERE score NOT IN ('Зачёт', 'Незачёт')
6      GROUP BY group_number) as groups
7 INNER JOIN "SESSION".group USING (group_number)
8 INNER JOIN "SESSION".direction USING (id_direction)
9 WHERE name_direction = 'Прикладная информатика'
10 ORDER BY total DESC
```

Результат

	<div>name_direction</div> <div>character varying (50)</div>	<div>group_number</div> <div>character varying</div>	<div>round</div> <div>numeric</div>	
1	Прикладная информатика	K3240	5.00	
2	Прикладная информатика	K3140	3.00	

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.

Запрос:

```
SELECT DISTINCT id_group, last_name_student, name_student, ROUND(average_score,2), scholarship
FROM (SELECT number_record_book,
            AVG(CAST(score AS int)) as average_score,
            MIN(CASE WHEN score > '3' AND score != 'Незачёт' THEN 1 ELSE 0 END) AS
scholarship
FROM "SESSION".passing_exam
WHERE score NOT IN ('Зачёт', 'Незачёт')
GROUP BY number_record_book) as students
INNER JOIN "SESSION".studying_student USING (number_record_book)
INNER JOIN "SESSION".student USING (id_student)
ORDER BY id_group, last_name_student
```

Query Editor

```
1 SELECT DISTINCT id_group, last_name_student, name_student, ROUND(average_score,2), scholarship
2 FROM (SELECT number_record_book,
3           AVG(CAST(score AS int)) as average_score,
4           MIN(CASE WHEN score > '3' AND score != 'Незачёт' THEN 1 ELSE 0 END) AS scholarship
5 FROM "SESSION".passing_exam
6 WHERE score NOT IN ('Зачёт', 'Незачёт')
7 GROUP BY number_record_book) as students
8 INNER JOIN "SESSION".studying_student USING (number_record_book)
9 INNER JOIN "SESSION".student USING (id_student)
10 ORDER BY id_group, last_name_student
11
```

Результат

	id_group integer	last_name_student character varying (20)	name_student character varying (20)	round numeric	scholarship integer
1	1	Бабан	Виктория	5.00	1
2	1	Балдина	Дарья	5.00	1
3	2	Байков	Иван	3.00	0
4	3	Жуков	Вадим	3.50	0
5	3	Жуков	Дмитрий	2.00	0
6	3	Устин	Денис	4.50	1

6. Вывести список студентов, сдавших все положенные экзамены

Запрос:

```
SELECT DISTINCT number_record_book, last_name_student, name_student
FROM (SELECT number_record_book,
             SUM(CASE WHEN score >= '3' AND score != 'Незачёт'
                    THEN 1 ELSE 0 END) as sum_score,
             COUNT(DISTINCT id_shedule) as count_exams
      FROM "SESSION".passing_exam
      GROUP BY number_record_book) as scores
INNER JOIN "SESSION".stydyng_student USING (number_record_book)
INNER JOIN "SESSION".student USING (id_student)
WHERE sum_score = count_exams
ORDER BY last_name_student
```

Query Editor

```
1 SELECT DISTINCT number_record_book, last_name_student, name_student
2 FROM (SELECT number_record_book,
3             SUM(CASE WHEN score >= '3' AND score != 'Незачёт'
4                  THEN 1 ELSE 0 END) as sum_score,
5             COUNT(DISTINCT id_shedule) as count_exams
6             FROM "SESSION".passing_exam
7             GROUP BY number_record_book) as scores
8 INNER JOIN "SESSION".stydyng_student USING (number_record_book)
9 INNER JOIN "SESSION".student USING (id_student)
10 WHERE sum_score = count_exams
11 ORDER BY last_name_student |
```

Результат

	number_record_book integer	last_name_student character varying (20)	name_student character varying (20)
1	312310	Бабан	Виктория
2	336100	Байков	Иван
3	312407	Балдина	Дарья
4	283128	Жуков	Вадим
5	312539	Козлов	Иван
6	283991	Устин	Денис

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

Запрос:

```
SELECT last_name_student, id_group, ROUND(average_score,2)
FROM (SELECT number_record_book, AVG(CAST(score AS int)) as average_score
      FROM "SESSION".passing_exam
      WHERE score NOT IN ('Зачёт', 'Незачёт')
      GROUP BY number_record_book) as all_students
INNER JOIN "SESSION".studying_student USING (number_record_book)
INNER JOIN "SESSION".student USING (id_student)
WHERE (id_group, average_score) = ANY(SELECT id_group, MAX(average_score)
                                     FROM (SELECT number_record_book,
                                           AVG(CAST(score AS int)) as average_score
                                           FROM "SESSION".passing_exam
                                           WHERE score NOT IN ('Зачёт', 'Незачёт')
                                           GROUP BY number_record_book) as all_students
                                     INNER JOIN "SESSION".studying_student USING
(number_record_book)
                                     GROUP BY id_group)
ORDER BY id_group, last_name_student
```

Query Editor История запросов

```
1 SELECT last_name_student, id_group, ROUND(average_score,2)
2 FROM (SELECT number_record_book, AVG(CAST(score AS int)) as average_score
3       FROM "SESSION".passing_exam
4       WHERE score NOT IN ('Зачёт', 'Незачёт')
5       GROUP BY number_record_book) as all_students
6 INNER JOIN "SESSION".studying_student USING (number_record_book)
7 INNER JOIN "SESSION".student USING (id_student)
8 WHERE (id_group, average_score) = ANY(SELECT id_group, MAX(average_score)
9                                       FROM (SELECT number_record_book, AVG(CAST(score AS int)) as average_score
10                                              FROM "SESSION".passing_exam
11                                              WHERE score NOT IN ('Зачёт', 'Незачёт')
12                                              GROUP BY number_record_book) as all_students
13                                       INNER JOIN "SESSION".studying_student USING (number_record_book)
14                                       GROUP BY id_group)
15 ORDER BY id_group, last_name_student
16
```

Результат План выполнения Сообщения Notifications

	last_name_student character varying (20)	id_group integer	round numeric
1	Бабан	1	5.00
2	Балдина	1	5.00
3	Байков	2	3.00
4	Устин	3	4.50

Задание 3. Представления

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

Query Editor

```
1 CREATE VIEW "SESSION".failed_attempt AS
2 SELECT "SESSION".passing_exam.number_record_book AS number_record_book,
3        "SESSION".student.last_name_student AS last_name_student,
4        "SESSION".passing_exam.score AS score,
5        "SESSION".teacher.last_name_teacher AS last_name_teacher
6 FROM "SESSION".passing_exam
7 INNER JOIN "SESSION".shedule_session USING (id_shedule)
8 INNER JOIN "SESSION".studyng_student USING (number_record_book)
9 INNER JOIN "SESSION".student USING (id_student)
10 INNER JOIN "SESSION".teacher USING (id_teacher)
11 WHERE attempt_number = 1 AND score < '3';
12
13 SELECT * FROM "SESSION".failed_attempt;
14
```

Сообщения

Запрос выполнен успешно. Общее время выполнения: 187 мсек.
обработано строк: 2.

Query Editor

История запросов

План выполнения

Notifications

Результат

	number_record_book integer	last_name_student character varying (20)	score character varying (10)	last_name_teacher character varying (20)
1	336100	Байков	2	Блаженев
2	283129	Жуков	2	Таранов

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

Query Editor История запросов

1 CREATE VIEW "SESSION".failed_students AS

2 SELECT DISTINCT "SESSION".passing_exam.number_record_book AS number_record_book,

3 "SESSION".student.last_name_student AS last_name_student,

4 "SESSION".student.name_student AS name_student

5 FROM "SESSION".studying_student

6 INNER JOIN "SESSION".student USING (id_student)

7 INNER JOIN "SESSION".passing_exam USING (number_record_book)

8 WHERE '2' = ANY (SELECT "SESSION".passing_exam.score AS score

9 FROM "SESSION".passing_exam

10 WHERE "SESSION".studying_student.number_record_book = "SESSION".passing_exam.number_record_book

11 AND attempt_number = 3);

12

План выполнения Notifications Результат Сообщения

CREATE VIEW

Запрос завершён успешно, время выполнения: 207 мсек.

Query Editor История запросов

1 SELECT * FROM "SESSION".failed_students

2

План выполнения Notifications Результат Сообщения

	number_record_book integer	last_name_student character varying (20)	name_student character varying (20)	
1	283129	Жуков	Дмитрий	

Задание 4. Модификация данных

1. Запрос с UPDATE

Повысить базовую стипендию в текущем семестре на 100 рублей.

До:

Query Editor

История запросов

```
1 SELECT * FROM "SESSION".scholarship
2 ORDER BY id_scholarship ASC
```

Результат

План выполнения

Сообщения

Notifications

	id_scholarship [PK] integer	type_scholarship character varying (20)	payment_start_date date	payment_end_date date	size_scholarship integer	number_record_book integer
1		Повышенная	2022-02-01	2022-06-30	4100	312310
2		Базовая	2021-07-01	2022-01-31	2000	312310
3		Повышенная	2022-02-01	2022-06-30	6000	283128
4		Повышенная	2021-07-01	2022-01-31	10000	336100
5		Базовая	2022-02-01	2022-06-30	2000	336100
6		Социальная	2022-02-01	2022-06-30	3000	283991
7		Социальная	2021-07-01	2022-01-31	3000	283991
8		Базовая	2021-07-01	2022-01-31	2000	336702

```
UPDATE "SESSION".scholarship
SET size_scholarship = size_scholarship + 100
WHERE id_scholarship IN
(SELECT id_scholarship FROM "SESSION".scholarship
WHERE type_scholarship = 'Базовая' AND
CURRENT_TIMESTAMP BETWEEN payment_start_date AND payment_end_date);
```

После:

План выполнения		Notifications		Результат		Сообщения	
	id_scholarship [PK] integer	type_scholarship character varying (20)	payment_start_date date	payment_end_date date	size_scholarship integer	number_record_book integer	
1		Повышенная	2022-02-01	2022-06-30	4100	312310	
2		Базовая	2021-07-01	2022-01-31	2000	312310	
3		Повышенная	2022-02-01	2022-06-30	6000	283128	
4		Повышенная	2021-07-01	2022-01-31	10000	336100	
5		Базовая	2022-02-01	2022-06-30	2100	336100	
6		Социальная	2022-02-01	2022-06-30	3000	283991	
7		Социальная	2021-07-01	2022-01-31	3000	283991	
8		Базовая	2021-07-01	2022-01-31	2000	336702	

2. Запрос с INSERT

Добавить сдачу зачета по Бадам данных учащимся 312539

До:

	id_exam [PK] integer	id_shedule integer	attempt_number integer	number_record_book integer	score character varying (10)
1	1	1	1	312310	Зачёт
2	2	1	1	312407	Зачёт
3	3	4	1	336100	Зачёт
4	4	10	1	336100	2
5	5	10	2	336100	4
6	6	9	1	312310	5
7	7	9	1	312407	5
8	8	7	1	283128	4
9	9	8	1	283128	3
10	10	7	1	283129	2
11	11	7	2	283129	2
12	12	7	3	283129	2
13	13	8	1	283129	3
14	14	7	1	283991	5
15	15	8	1	283991	5

```
Query Editor  История запросов

1  INSERT INTO "SESSION".passing_exam(
2      id_shedule, attempt_number, number_record_book, score)
3      VALUES ((SELECT id_shedule FROM "SESSION".shedule_session
4                  WHERE group_number = 'K3240' AND exam_status = 'Проведен'
5                  AND id_subject = (SELECT id_subject FROM "SESSION".subject
6                                    WHERE name_subject = 'Базы данных')), 1, 312539, 'Зачёт');
7
8






Результат  План выполнения  Сообщения  Notifications

INSERT 0 1

Запрос завершён успешно, время выполнения: 79 мсек.
```

```
INSERT INTO "SESSION".passing_exam(
    id_shedule, attempt_number, number_record_book, score)
VALUES ((SELECT id_shedule FROM "SESSION".shedule_session
        WHERE group_number = 'K3240' AND exam_status = 'Проведен'
        AND id_subject = (SELECT id_subject FROM "SESSION".subject
            WHERE name_subject = 'Базы данных')), 1, 312539, 'Зачёт');
```

После:

Результат	План выполнения		Notifications	Сообщения	
	 id_exam [PK] integer	 id_shedule integer	 attempt_number integer	 number_record_book integer	 score character varying (10)
1	1	1	1	312310	Зачёт
2	2	1	1	312407	Зачёт
3	3	4	1	336100	Зачёт
4	4	10	1	336100	2
5	5	10	2	336100	4
6	6	9	1	312310	5
7	7	9	1	312407	5
8	8	7	1	283128	4
9	9	8	1	283128	3
10	10	7	1	283129	2
11	11	7	2	283129	2
12	12	7	3	283129	2
13	13	8	1	283129	3
14	14	7	1	283991	5
15	15	8	1	283991	5
16	17	1	1	312539	Зачёт

3. Запрос с DELETE

Удалить из таблицы сданных экзаменов экзамены, статусы которых еще «Запланирован»

До:

	id_exam [PK] integer	id_shedule integer	attempt_number integer	number_record_book integer	score character
1	1	1	1	312310	Зачёт
2	2	1	1	312407	Зачёт
3	3	3	1	336100	Зачёт
4	4	10	1	336100	2
5	5	10	2	336100	4
6	6	9	1	312310	5
7	7	9	1	312407	5
8	8	7	1	283128	4
9	9	8	1	283128	3
10	10	7	1	283129	2
11	11	7	2	283129	2
12	12	7	3	283129	2
13	13	6	1	283129	3
14	14	7	1	283991	5
15	15	8	1	283991	4
16	17	1	1	312539	Зачёт

```
DELETE FROM "SESSION".passing_exam
WHERE id_shedule IN (SELECT id_shedule FROM "SESSION".shedule_session
                     WHERE exam_status = 'Запланирован');
```

После:

Результат		План выполнения		Сообщения		Notifications	
	id_exam [PK] integer	id_shedule integer	attempt_number integer	number_record_book integer	score character varying (10)		
1	1	1	1	312310	Зачёт		
2	2	1	1	312407	Зачёт		
3	4	10	1	336100	2		
4	5	10	2	336100	4		
5	6	9	1	312310	5		
6	7	9	1	312407	5		
7	8	7	1	283128	4		
8	9	8	1	283128	3		
9	10	7	1	283129	2		
10	11	7	2	283129	2		
11	12	7	3	283129	2		
12	14	7	1	283991	5		
13	15	8	1	283991	4		
14	17	1	1	312539	Зачёт		

Задание 5. История запросов

Query Editor

История запросов

Show queries generated internally by pgAdmin?

Today - 28.04.2022

SELECT * FROM "SESSION".scholarship ORDER BY id_scholarship ASC
14:32:19

▶ UPDATE "SESSION".scholarship SET size_scholarship = size_scholarsh...
14:32:03

▶ SELECT CURRENT_TIMESTAMP
14:31:44

▶ CURRENT_TIMESTAMP
14:31:38

▶ GETDATE()
14:30:54

▶ UPDATE "SESSION".scholarship SET size_scholarship = size_scholarsh...
14:30:28

▶ UPDATE "SESSION".scholarship SET size_scholarship = size_scholarsh...
14:30:11

▶ UPDATE "SESSION".scholarship SET size_scholarship = size_scholarsh...
14:29:53

▶ UPDATE SET size_scholarship=size_scholarship + 100 WHERE id_schola...
14:29:34

28.04.2022 14:32:03

1

280 msec

Дата

Rows Affected

Продолжительность

Copy

Copy to Query Editor

UPDATE "SESSION".scholarship
SET size_scholarship = size_scholarship + 100
WHERE id_scholarship IN (SELECT id_scholarship FROM "SESSION".scho1
WHERE type_scholarship = 'Базовая' AND CUF

Сообщения
UPDATE 1

Запрос завершён успешно, время выполнения: 280 msec.

ВЫВОДЫ

SQL запросы позволяют изменять, добавлять или удалять данные, атакже составлять различные выборки, подсчитывать числовые характеристики.