

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”
ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5
«Реализация SQL-запросов в субд PostgreSQL»

Специальность 09.02.07 «Информационные системы и программирование»

Преподаватель:

Говоров А.И.

«25» февраля 2020г.

Оценка:

Выполнил:

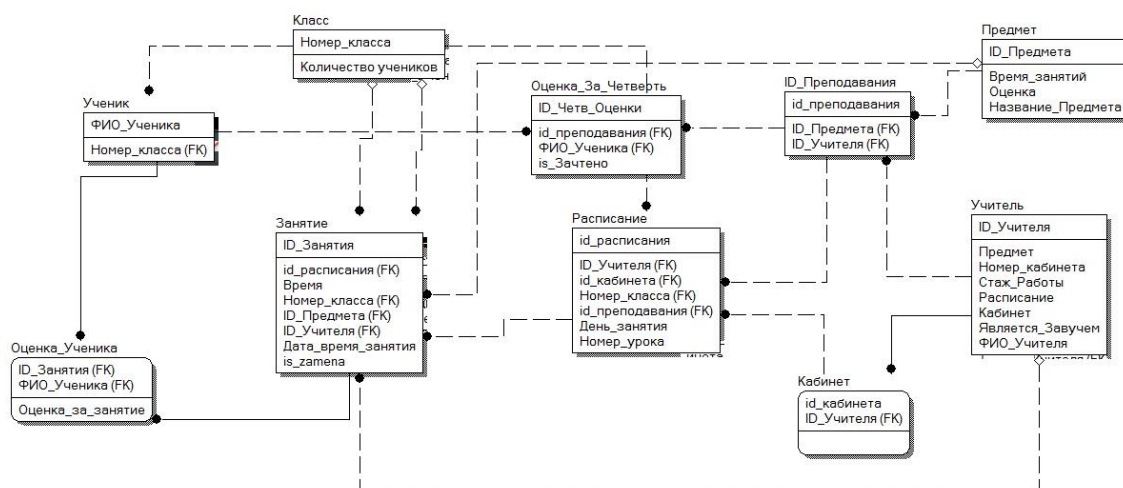
студент группы Y2339

Романов К.Д.

ЦЕЛЬ РАБОТЫ


Необходимо реализовать некоторое количество SQL-запросов. Требуется написать определенное количество запросов на определенное количество баллов, зависимое от оценки, на которую Вы претендуете.

Скрин базы данных:



Описание: База данных реализует школьную систему взаимоотношений. Имеется несколько базовых таблиц(Ученик,Учитель и т.п.), а также таблицы – связи(Оценка_Ученика,Расписание,Занятие и.т.п.)

1. выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;


12.03/postgres@PostgreSQL 12

Query Editor
Query History

```

1 SELECT L."Lecturer_Name", S."subject_title"
2 FROM "Lecturer" AS L
3 JOIN "Subject" AS S
4 ON L."Lecturer_ID" = S."Subject_ID"

```

Data Output
Explain
Messages
Notifications

	Lecturer_Name text	subject_title text
1	{Yatskevi4 Agafon E...	Russian_subject
2	{Kostin Kir Ulyanovi4}	mathematic
3	{Nabokin Mikhail Ma...	physics

SELECT L."Lecturer_Name", S."subject_title"

FROM "Lecturer" AS L

JOIN "Subject" AS S

ON L."Lecturer_ID" = S."Subject_ID"

- использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;

12.03/postgres@PostgreSQL 12

Query Editor Query History

```

1 SELECT *
2 FROM "Grade_4_quarter"
3 WHERE "lecture_id" = 1 OR "lecture_id" = 2
4 AND "Is_set_off" = true
5
6
7

```

Data Output Explain Messages Notifications

	Grade_id [PK] integer	Is_set_off boolean	lecture_id integer	Student_Name text
1	1	true	1	Boresov Nikira Mikh...
2	2	false	1	Ivanov Valerie Nikol...
3	3	true	2	Ivanov Valerie Nikol...
4	5	true	2	Vorontsov Anton Va...
5	6	true	1	Vorontsov Anton Va...

3. использование функций для работы с датами – от 2 баллов;

Выводим имена преподавателей, которые начали свое обучение позже 2003 года

Query Editor

Query History

1

2

3

4

5

SELECT

L."Lecturer_Name"

FROM "Lecturer" AS L

WHERE lower(L."Experience") > to_date('05 Dec 2003','DD Mon YYYY')

Data Output

Explain

Messages

Notifications

	Lecturer_Name text	
1	{Nabokin Mikhail Ma...	
2	{Yatskevi4 Agafon E...	

4. использование строковых функций – от 3 баллов;

Длина имени каждого из учителей

```
SELECT LENGTH(L."Lecturer_Name") AS "length of Lecturer's name"
```

```
FROM "Lecturer" AS L
```



```
1 SELECT LENGTH(L."Lecturer_Name") AS "length of Lecturer's name"
2 FROM "Lecturer" AS L
```

	length of Lecturer's name integer	
1	22	
2	28	
3	29	

5. запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 3 баллов);

12.03/postgres@PostgreSQL 12

Query Editor Query History

```

1
2 SELECT "Lecturer_Name", "Subject", abs(
3     (SELECT date_part('year',age(lower("Experience"),upper("Experience"))))
4     FROM "Lecturer" as L2 WHERE L1."Lecturer_ID" = L2."Lecturer_ID" )) AS "was studying"
5 FROM "Lecturer" as L1
6

```

Data Output Explain Messages Notifications

	Lecturer_Name text	Subject text	was studying double precision
1	{Kostin Kir Ulyanovi4}	{Mathemati...	15
2	{Nabokin Mikhail Ma...	{IKT}	3
3	{Yatskevi4 Agafon E...	{Russian_Ia...	14

```

SELECT "Lecturer_Name", "Subject", abs(
    (SELECT date_part('year',age(lower("Experience"),upper("Experience"))))
    FROM "Lecturer" as L2 WHERE L1."Lecturer_ID" = L2."Lecturer_ID" )) AS
"was studying"
FROM "Lecturer" as L1

```

Выводит Имя преподавателя,его предмет и количество лет обучения

6. вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 3 баллов);



Query Editor Query History

```
1 SELECT count(*)
2 FROM "Lecturer" AS L
3 JOIN "Subject" AS S
4 ON L."Lecturer_ID" = S."Subject_ID"
5
6
```

Data Output Explain Messages Notifications

	count bigint	
1	3	

```
SELECT count(*)
```

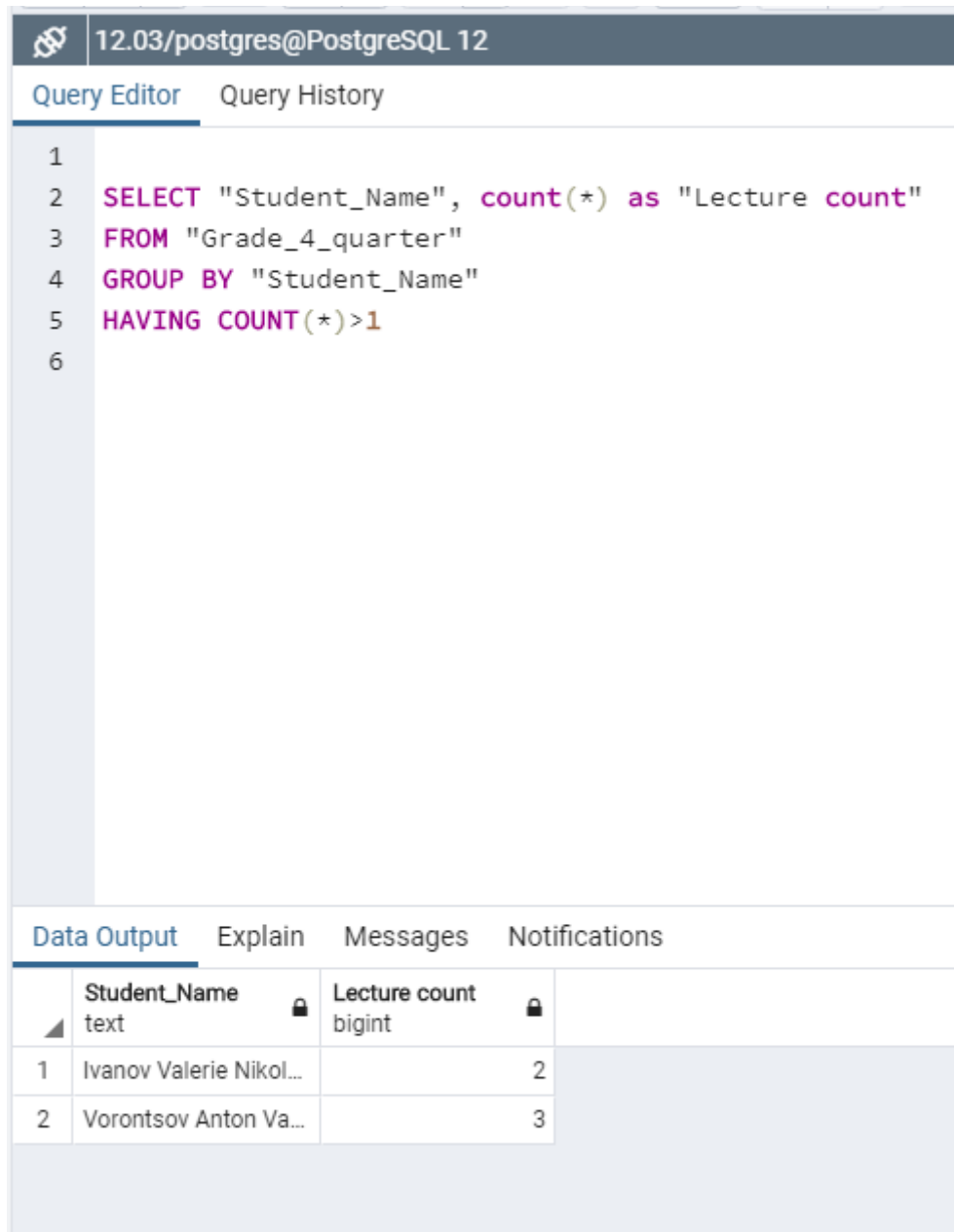
```
FROM "Lecturer" AS L
```

```
JOIN "Subject" AS S
```

```
ON L."Lecturer_ID" = S."Subject_ID"
```

Выводит количество Учителей имеющих хотя бы 1 предмет

7. вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов;



The screenshot shows a PostgreSQL Query Editor window. The title bar indicates the connection is '12.03/postgres@PostgreSQL 12'. The 'Query Editor' tab is active, displaying a SQL query. The query is as follows:

```
1
2 SELECT "Student_Name", count(*) as "Lecture count"
3 FROM "Grade_4_quarter"
4 GROUP BY "Student_Name"
5 HAVING COUNT(*)>1
6
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'Student_Name' (text) and 'Lecture count' (bigint). The results are as follows:

	Student_Name	Lecture count
1	Ivanov Valerie Nikol...	2
2	Vorontsov Anton Va...	3

```
SELECT "Student_Name", count(*) as "Lecture count"
```

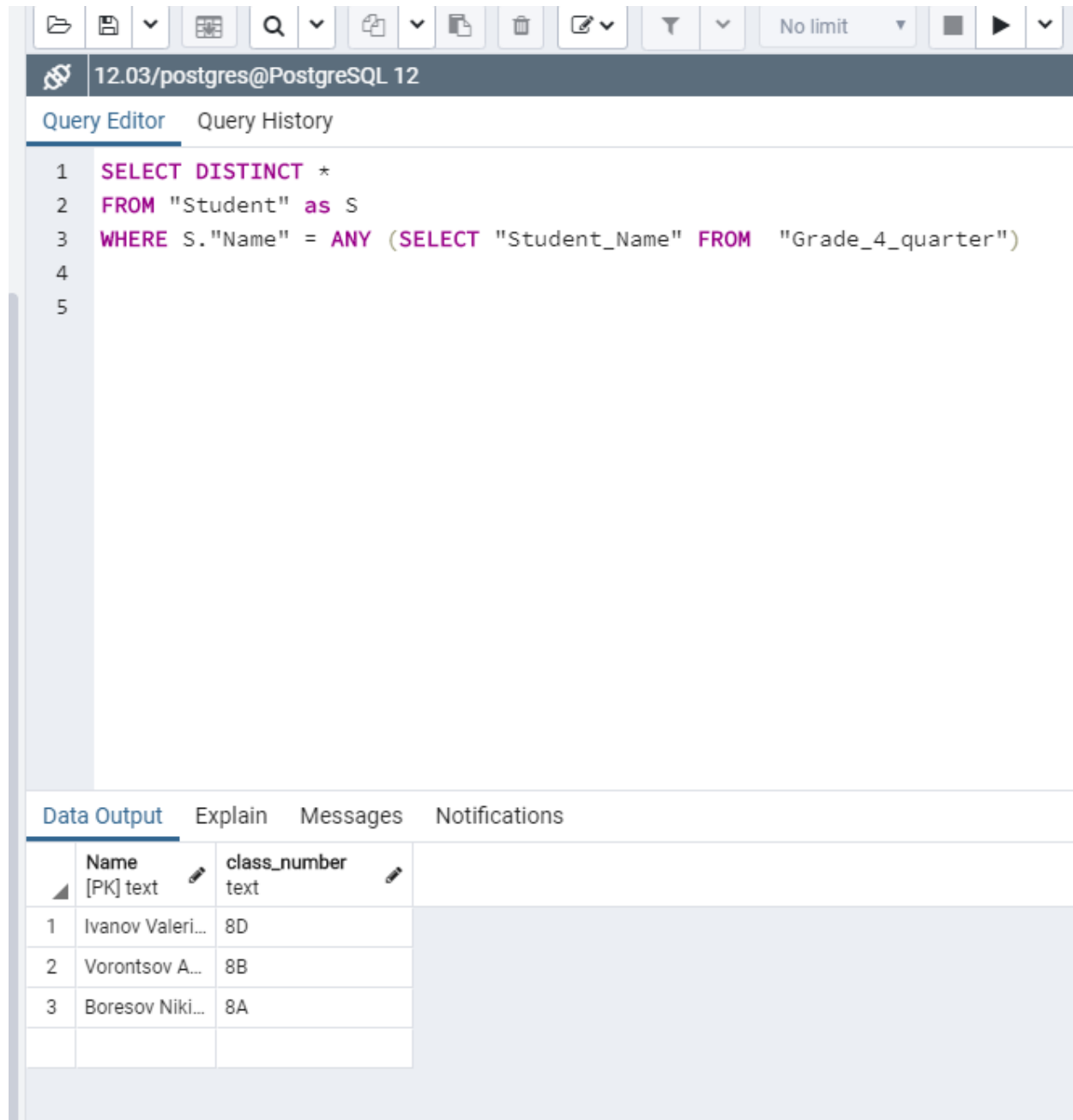
```
FROM "Grade_4_quarter"
```

```
GROUP BY "Student_Name"
```

```
HAVING COUNT(*)>2
```

Вычисляем количество студентов имеющих более 1 оцененного предмета

8. использование предикатов EXISTS, ALL, SOME и ANY - от 4 баллов;



The screenshot shows a PostgreSQL Query Editor interface. The top toolbar includes icons for file operations, search, and execution. The title bar indicates the connection is to '12.03/postgres@PostgreSQL 12'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SELECT DISTINCT *
2 FROM "Student" as S
3 WHERE S."Name" = ANY (SELECT "Student_Name" FROM "Grade_4_quarter")
4
5
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'Name [PK] text' and 'class_number text'. The results are as follows:

	Name [PK] text	class_number text
1	Ivanov Valeri...	8D
2	Vorontsov A...	8B
3	Boresov Niki...	8A

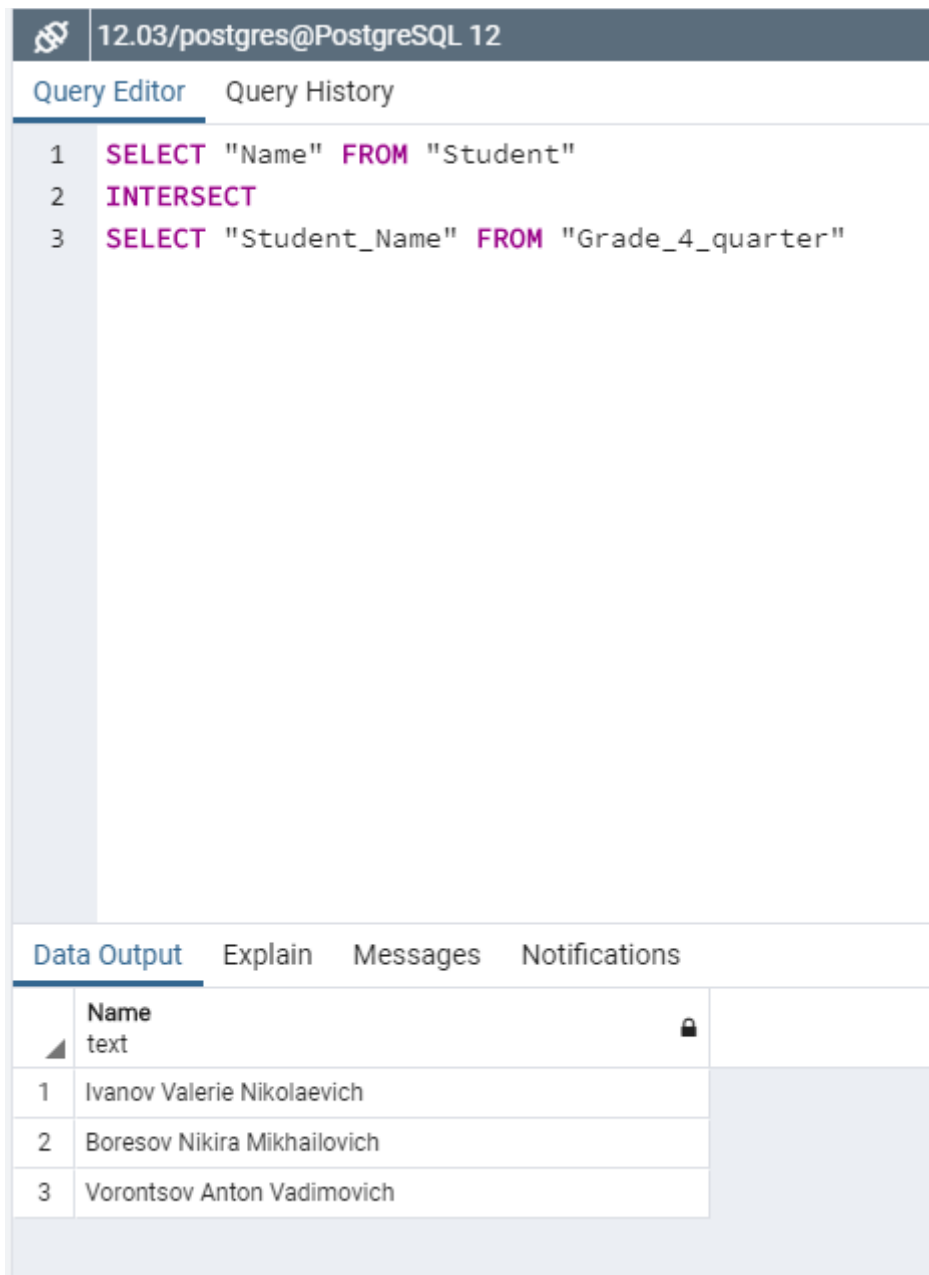
SELECT DISTINCT *

FROM "Student" as S

WHERE S."Name" = ANY (SELECT "Student_Name" FROM "Grade_4_quarter")

Вывести учеников у которых есть оценки за четверть

9. использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 3 баллов;



The screenshot shows the PostgreSQL 12 Query Editor interface. The top bar indicates the connection '12.03/postgres@PostgreSQL 12'. Below the bar are two tabs: 'Query Editor' (active) and 'Query History'. The query editor contains the following SQL code:

```
1 SELECT "Name" FROM "Student"
2 INTERSECT
3 SELECT "Student_Name" FROM "Grade_4_quarter"
```

Below the query editor are four tabs: 'Data Output' (active), 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab displays the results of the query in a table with three rows:

	Name text	
1	Ivanov Valerie Nikolaevich	
2	Boresov Nikira Mikhailovich	
3	Vorontsov Anton Vadimovich	

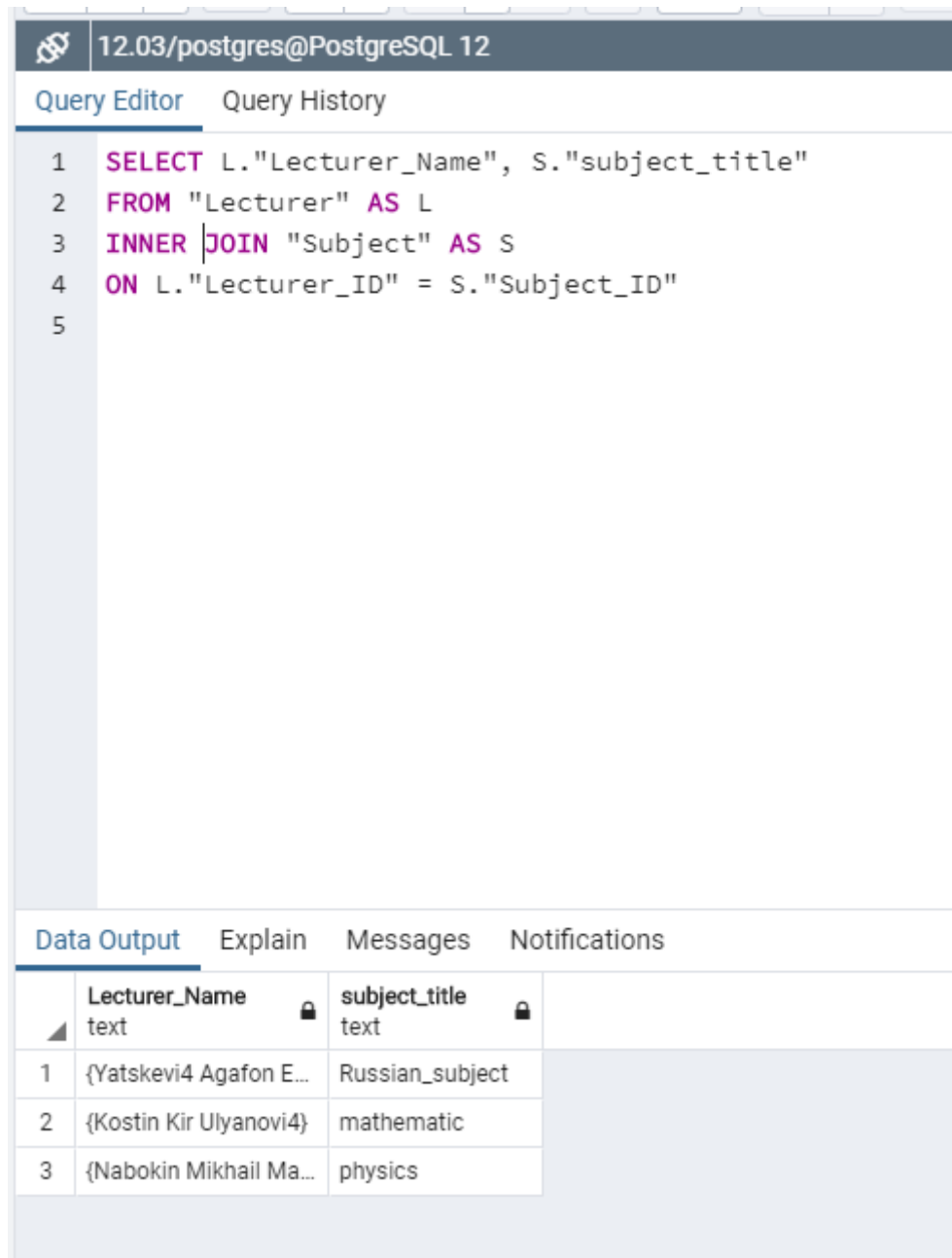
SELECT "Name" FROM "Student"

INTERSECT

SELECT "Student_Name" FROM "Grade_4_quarter"

Выводит пересечение двух таблиц, а именно имена, находящиеся в обеих таблицах

10. использование объединений запросов (inner join и т.д.) - от 3 баллов.
Запросов должно быть не менее 10 (30 баллов на 5, 25 баллов на 4, 20 баллов на 3)(5, 4, 3 - это оценки, а не суммы баллов в журнале).



The screenshot shows a PostgreSQL Query Editor window titled "12.03/postgres@PostgreSQL 12". It has two tabs: "Query Editor" and "Query History". The "Query Editor" tab is active, displaying the following SQL query:

```
1 SELECT L."Lecturer_Name", S."subject_title"
2 FROM "Lecturer" AS L
3 INNER JOIN "Subject" AS S
4 ON L."Lecturer_ID" = S."Subject_ID"
5
```

Below the query editor, there are four tabs: "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is active, showing a table with the results of the query:

	Lecturer_Name text	subject_title text
1	{Yatskevi4 Agafon E...	Russian_subject
2	{Kostin Kir Ulyanovi4}	mathematic
3	{Nabokin Mikhail Ma...	physics

```
SELECT L."Lecturer_Name", S."subject_title"
FROM "Lecturer" AS L
INNER JOIN "Subject" AS S
ON L."Lecturer_ID" = S."Subject_ID"
```

Выводит значения, аналогичные 9 пункту – имена , находящиеся в обеих таблицах

Вывод :были повторены и закреплены прикладные знания в области управления субд postgresql.