

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

«Реализование некоторого количества

SQL-запросов в PGAdmin»

Выполнил:

Нгуен Тхюи Чанг

Группа К3241

Преподаватель:

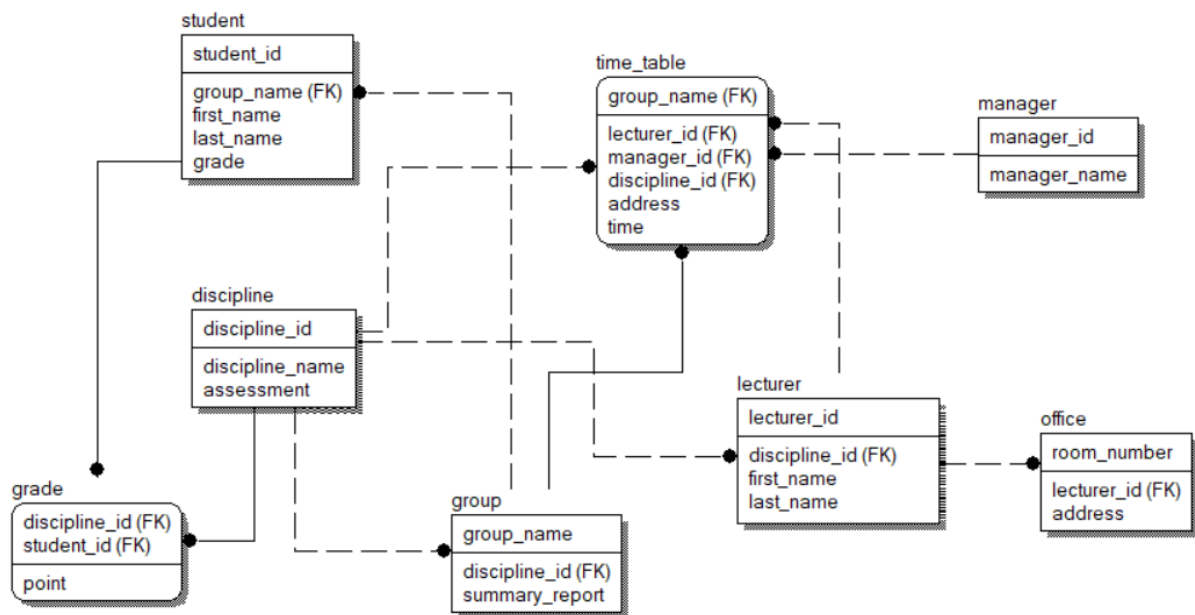
Говоров А.И.

2020

г. Санкт-Петербург

1. Цель работы: Овладеть практическими навыками создания SQL запросов к база данных.

2. Индивидуальное задание



Создать программную систему, предназначенную для учебной части колледжа. Она должна обеспечивать хранение сведений о каждом преподавателе, о дисциплинах, которые он преподает, номере закрепленного за ним кабинета, о расписании занятий. Существуют преподаватели, которые не имеют собственного кабинета. О студентах должны храниться следующие сведения: фамилия и имя, в какой группе учиться, какую оценку имеет в текущем семестре по каждой дисциплине. Зам. декана должен иметь возможность добавить сведения о новом преподавателе или студенте, внести в базу данных семестровые оценки студентов каждой группы по каждой дисциплине, удалить данные об уволившемся преподавателе и отчисленном из колледжа студенте, внести изменения в данные об преподавателях и студентах, в том числе поменять оценку студента по той или иной дисциплине. В задачу диспетчера учебной части входит составление расписания. Замдекана могут потребоваться следующие сведения:

- Какой предмет будет в заданной группе в заданный день недели на заданном уроке?
- Кто из преподавателей преподает в заданной группе?
- В каких группах преподает заданный предмет заданный преподаватель?
- Расписание на заданный день недели для указанной группы?
- Сколько студентов обучается на каждом курсе в указанном классе?

Необходимо предусмотреть возможность получения документа, представляющего собой сводные ведомости успеваемости за семестр по каждой группе. В ведомости необходимо предусмотреть сведения о среднем балле группы за семестр.

3. Выполнение

- 3.1. Получить first_name, last_name, discipline_name и assessment из таблицы « discipline » и « lecturer » по алфавиту по возрастанию.

```
SELECT first_name, last_name, discipline_name, assessment FROM discipline  
INNER JOIN lecturer ON discipline.discipline_id = lecturer.discipline_id  
ORDER BY lecturer.first_name;
```

	Data Output	Explain	Messages	Notifications
	first_name character varying	last_name character varying	discipline_name character varying	assessment character varying
1	Alexandra	Vatyan	Maths	Pass/Fall
2	Anna	Vasilevskaya	Foreign Language	Exam
3	Anton	Govorov	Database	Exam
4	Kirill	Boyarsky	Physics	Exam
5	Natalia	Serebryanskaya	Informatics	Exam

- 3.2. Получить все столбцы с именем дисциплина «Foreign Language» и с оценкой «Exam» из таблицы «discipline»

```
SELECT * FROM discipline  
WHERE discipline_name != 'Foreign Language' AND assessment = 'Exam';
```

Data Output

Explain

Messages

Notifications

	discipline_id bigint	discipline_name character varying	assessment character varying
1	1	Database	Exam
2	2	Physics	Exam
3	4	Informatics	Exam

- 3.3. Получить group_name, discipline_name, first_name, last_name, address, time из трех таблиц « time_table », « lecturer », « discipline » когда значение « time » до 01.06.2020

```
SELECT group_name, discipline_name, first_name, last_name, address, time FROM
time_table
INNER JOIN lecturer ON lecturer.lecturer_id = time_table.lecturer_id
INNER JOIN discipline ON discipline.discipline_id = time_table.discipline_id
WHERE time_table.time <= '2020-06-01';
```

Data Output		Explain	Messages	Notifications		
	group_name character varying	discipline_name character varying	first_name character varying	last_name character varying	address character varying	time date
1	K3241	Database	Anton	Govorov	Kronverkskiy Prospekt, 49	2020-05-04
2	K3240	Database	Anton	Govorov	Kronverkskiy Prospekt, 49	2020-05-04
3	K3242	Database	Anton	Govorov	Kronverkskiy Prospekt, 49	2020-05-04
4	K3243	Database	Anton	Govorov	Kronverkskiy Prospekt, 49	2020-05-04

3.4. Обновить столбец таблицы « time_table » с адресом « Kronverkskiy Prospekt, 49 » до « Birzhevaya Liniya, 14 »

```
UPDATE time_table SET address = 'Birzhevaya Liniya, 14'
WHERE address = 'Kronverkskiy Prospekt, 49';
```

	Data Output	Explain	Messages	Notifications
	UPDATE 4			
	Query returned successfully in 64 msec.			

3.5. Объединить значения 2 столбца «first_name» и «last_name» с пробелом из таблицы «lecturer»

```
SELECT discipline_name, CONCAT(lecturer.first_name, ' ', lecturer.last_name) from
lecturer
INNER JOIN discipline ON discipline.discipline_id = lecturer.discipline_id;
```

Data Output

Explain

Messages

Notifications

	discipline_name character varying	concat text
1	Database	Anton Govorov
2	Physics	Kirill Boyarsky
3	Foreign Language	Anna Vasilevskaya
4	Informatics	Natalia Serebryanskaya
5	Maths	Alexandra Vatyan

- 3.6.** Использовать подзапросы чтобы получить «group_name», «address», «time» из двух таблиц «time_table» и «group» с именем группы, отличная «K3244»

```
SELECT group_name, address, time FROM time_table
WHERE group_name IN(
SELECT group_name FROM public.group
WHERE group_name != 'K3244');
```

	Data Output	Explain	Messages	Notifications
	group_name character varying	address character varying	time date	
1	K3241	Birzhevaya Liniya, 14	2020-05-04	
2	K3240	Birzhevaya Liniya, 14	2020-05-04	
3	K3242	Birzhevaya Liniya, 14	2020-05-04	
4	K3243	Birzhevaya Liniya, 14	2020-05-04	

- 3.7.** Получить количество студенты, максимум оценка, минимум оценка и вокруг среднего баллы

```
SELECT
COUNT(*) AS total_student,
MAX(point) AS highest_grade,
MIN(point) AS lowest_grade,
ROUND(AVG(point), 1) AS avg_grade
FROM grade;
```

	Data Output	Explain	Messages	Notifications
	total_student bigint	highest_grade bigint	lowest_grade bigint	avg_grade numeric
1	5	95	81	90.6

- 3.8.** Вычисление групповой (агрегатной) функции с условием HAVING, GROUP BY с двух колонками «student_id» и «point» из таблицы «grade»

```
SELECT student_id, ROUND(AVG(point), 2) AS avg_point FROM public.grade
GROUP BY student_id
HAVING AVG(point) >= 91;
```

	Data Output	Explain	Messages	Notifications
	<div>student_id</div> <div>bigint</div>		<div>avg_point</div> <div>numeric</div>	
1		3	94.00	
2		5	92.00	
3		4	91.00	
4		2	95.00	

- 3.9.** Использовать предикат EXISTS чтобы получить «manager_name» из таблицы «manager»

```
SELECT manager_name FROM manager
WHERE EXISTS
(SELECT * FROM manager WHERE manager_id IS NOT NULL);
```

	Data Output	Explain	Messages	Notifications
	<div>manager_name</div> <div>character varying</div>			
1	Andrew N			

- 3.10.** Использовать предикат ANY чтобы получить «first_name» и «last_name» из таблицы «lecturer»

```
SELECT first_name, last_name FROM lecturer
WHERE lecturer_id = ANY (SELECT lecturer_id FROM office WHERE address IS
NOT NULL);
```

	Data Output	Explain	Messages	Notifications
	<div>first_name</div> <div>character varying</div>		<div>last_name</div> <div>character varying</div>	
1	Anton		Govorov	

- 3.11.** Использовать предикат ALL чтобы получить «group_name», «discipline_id» и «sumary_report» с условию «discipline_name» равно «Maths» из таблицы «group »

```
SELECT group_name, discipline_id, sumary_report FROM public.group
WHERE discipline_id = ALL (SELECT discipline_id FROM discipline WHERE
discipline_name = 'Maths');
```

	Data Output	Explain	Messages	Notifications
	group_name character varying	discipline_id bigint	sumary_report character varying	
1	K3244		5	[null]

- 3.12.** Использовать предикат SOME чтобы получить « first_name », « last_name » с условию «first_name» равно «Andrey» из таблицы « lecturer »

```
SELECT first_name, last_name FROM lecturer
WHERE lecturer_id = SOME (SELECT lecturer_id FROM students WHERE
first_name != 'Andrey')
```

	Data Output	Explain	Messages	Notifications
	first_name character varying	last_name character varying		
1	Anton	Govorov		
2	Kirill	Boyarsky		
3	Anna	Vasilevskaya		
4	Natalia	Serebryanskaya		
5	Alexandra	Vatyan		

- 3.13.** Использовать NATURAL JOIN, UNION чтобы получить «group_name», «first_name» с условиями из таблицы «students»

```
SELECT group_name, first_name, last_name FROM public.group NATURAL JOIN
students where
discipline_id > 2 UNION
SELECT group_name, first_name, last_name FROM public.group NATURAL JOIN
students where
discipline_id = 1;
```

Data Output	Explain	Messages	Notifications
	group_name character varying	first_name character varying	last_name character varying
1	K3244	Ekaterina	Grigoryeva
2	K3241	Trang	Nguyen
3	K3240	Sergey	Dubina
4	K3243	Evgeniya	Matyushina

3.14. Использовать NATURAL JOIN, INTERSECT чтобы получить «group_name», « student_id » с условиями из таблицы «students»

```
SELECT group_name, student_id FROM public.group NATURAL JOIN students
WHERE first_name = 'Sergey' INTERSECT
SELECT group_name, student_id FROM public.group NATURAL JOIN students
WHERE first_name = 'Ekaterina';
```

Data Output	Explain	Messages	Notifications
	group_name character varying	student_id bigint	

3.15. Использовать NATURAL JOIN, EXCEPT чтобы получить «group_name», « student_id » и « last_name » с условиями из таблицы «students»

```
SELECT group_name, student_id, last_name FROM public.group NATURAL JOIN
students WHERE first_name = 'Sergey' EXCEPT
SELECT group_name, student_id, last_name FROM public.group NATURAL JOIN
students WHERE first_name = 'Ekaterina';
```

Data Output	Explain	Messages	Notifications
	group_name character varying	student_id bigint	last_name character varying
1	K3240	3	Dubina

4. Вывод

В результате выполнения работы были получены навыки написания SQL запросов в среде pgAdmin.