

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ**

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

Факультет «Инфокоммуникационных технологий»

Направление подготовки «09.03.03 Мобильные и сетевые технологии»

О Т Ч Е Т

ЛАБОРАТОРНАЯ РАБОТА №4

Вариант 12

Выполнил:

Студент Корецкая Лидия К3240
(Фамилия И.О.) номер группы

Проверил:

Преподаватель Говоров А.И.
(Фамилия И.О.)

Санкт-Петербург

2020

Цель работы: овладеть практическими навыками написания SQL-запросов

Практическое задание:

1. выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;
2. использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;
3. использование функций для работы с датами – от 2 баллов;
4. использование строковых функций – от 2 баллов;
5. запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 2 баллов);
6. вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 2 баллов);
7. вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов;
8. использование предикатов EXISTS, ALL, SOME и ANY - от 2 баллов;
9. использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 2 баллов;
10. использование объединений запросов (inner join и т.д.) - от 2 баллов.

Описание варианта:

Создать программную систему, предназначенную для учебной части колледжа.

Она должна обеспечивать хранение сведений о каждом преподавателе, о дисциплинах, которые он преподаёт, 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 учится, какую оценку имеет в текущем семестре по каждой дисциплине.

Замдекана 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 преподаёт в заданной группе?

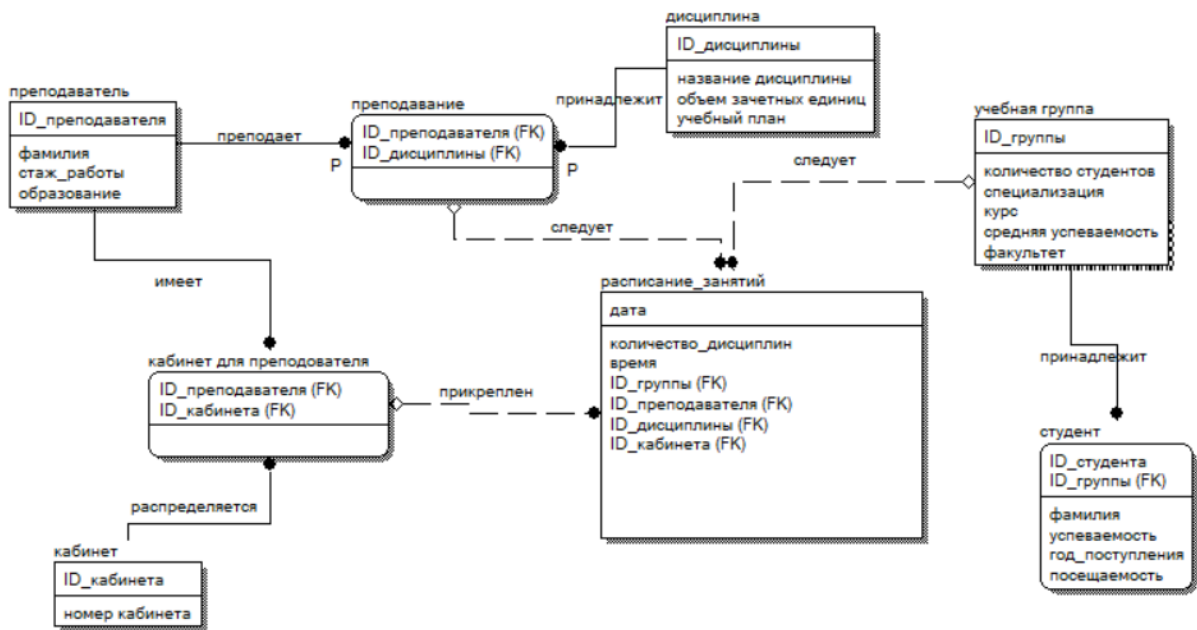
В каких группах преподаёт заданный предмет заданный преподаватель?

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.

Схема модели БД:



Список запросов:

- Выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой
 - Выбор всех значений из таблицы студенты и сортировка по успеваемость

```
select * from student order by attendance;
```

	surname text	id_student [PK] integer	year_of_receipt integer	attendance integer	academic_performance integer	id_group integer	
1	Hitman	4	1975	36		60	1
2	Tess	5	1990	60		80	5
3	Kravets	3	1999	80		75	2
4	Berg	2	2009	90		90	3
5	Ivanov	1	1987	100		100	2

- Выбор всех значений из таблицы группы и сортировка по специализации


```
select * from "group" order by specialization;
```

	number_of_students integer	id_group [PK] integer	course integer	specialization text	faculty text	average_performance integer
1	31	5	3	art	OKL	78
2	30	1	3	bio	ICT	80
3	24	3	2	pharmacy	THR	47
4	20	2	2	proga	ICT	67
5	15	4	3	sport	UFC	50

2. Использование условий WHERE, состоящих из более, чем одного условия

2.1. Выбрать все фамилии студентов, у которых успеваемость больше 50 и год поступления до 2000

select surname from student where attendance>50 and year_of_receipt<2000;

	surname text
1	Ivanov
2	Kravets
3	Tess

2.2. Выбрать фамилию преподавателя, у которого опыт работы не 1 месяц и образование в мгу или итмо

select surname from teacher where work_experience!='1 month' and education in('MGU', 'ИТМО');

1	Ivanova
---	---------

3. Использование функций для работы с датами

3.1. Найти промежуток времени между текущей датой и датой занятий

select class_data, age(class_data) as interval from timetable

	class_data [PK] date	interval interval
1	2020-12-15	-7 mons -3 days
2	2020-08-26	-3 mons -14 days
3	2020-07-14	-2 mons -2 days
4	2020-10-09	-4 mons -28 days
5	2020-11-16	-6 mons -4 days
6	2018-10-11	1 year 7 mons 1 day
7	2019-06-13	10 mons 29 days
8	2014-03-09	6 years 2 mons 3 days
9	2019-01-18	1 year 3 mons 25 days
10	2017-06-10	2 years 11 mons 2 days

3.2. Выбрать те пары, которые начинаются позже 7, но раньше 12

select class_time from timetable where date_part('hour', class_time)>7 and date_part('hour', class_time)<12

	class_time time without time zone
1	11:40:00
2	11:30:00
3	08:00:00

4. Использование строковых функций

4.1. Объединить столбцы фамилия и образование из таблицы преподаватель посредством строки 'получила образование в'

```
select concat (surname, ' получила образование в ', education) as result from teacher;
```

	result text
1	Ivanova получила образование в MGU
2	Filipov получила образование в KrasGU
3	Utkina получила образование в MPHTU
4	Lapina получила образование в Gertsena
5	Larin получила образование в ITMO

4.2. Выбрать айди тех студентов и их фамилию, у которых длина фамилии менее 5 символов

```
select id_student, surname from student where char_length(surname)<5;
```

	id_student [PK] integer	surname text
1	2	Berg
2	5	Tess

5. Запрос с использованием подзапросов

5.1. Выбрать все из таблицы расписания, где у групп средняя успеваемость больше 70

```
select * from timetable where id_group in (select id_group from "group" where average_performance>70)
```

	class_data [PK] date	number_of_discipline integer	class_time time without time zone	id_group integer	id_teacher integer	id_discipline integer	id_classroom integer
1	2020-07-14	1	06:00:00	1	1	2	2

5.2. Умножить на 0,25 успеваемость тех студентов, у которых год поступления позже 1990

```
update student set attendance = attendance * 0.25 where year_of_receipt in (select year_of_receipt from student where year_of_receipt > 1990 );
```

UPDATE 2

Query returned successfully in 86 msec.

6. Вычисление групповой (агрегатной) функции

6.1. Сколько студентов на 3 курсе

```
select sum(number_of_students) as students_from_3_course from "group" where course=3
```

	sudents_from_3_course bigint
1	76

6.2. Вывести среднее количество дисциплин по курсам

```
select "group".course, avg(timetable.number_of_discipline) as avg_number_of_discipline from "group" inner join timetable on "group".id_group=timetable.id_group group by "group".course
```

	course integer	avg_number_of_discipline numeric
1	3	1.3333333333333333
2	4	3.6666666666666667
3	2	2.5000000000000000
4	1	3.5000000000000000

7. Вычисление групповой (агрегатной) функции с условием HAVING

7.1. Вывести среднее число студентов больше 20 по специализациям

```
select specialization, avg(number_of_students) as avg_number_of_students from "group" group by specialization having avg(number_of_students)>20
```

	specialization text	avg_number_of_students numeric
1	art	31.0000000000000000
2	proga	33.6666666666666667
3	bio	22.0000000000000000
4	pharmacy	26.0000000000000000

7.2. Вывести среднюю успеваемость по тем факультетам, которым принадлежит более чем 1 группа

```
select faculty, avg(average_performance) as avg_faculty_performance from "group" group by faculty having count(faculty)>=2
```

	faculty text	avg_faculty_performance numeric
1	UFC	65.0000000000000000
2	ICT	66.2000000000000000
3	THR	57.0000000000000000

8. Использование предикатов EXISTS, ALL, SOME и ANY

8.1. Вывести фамилии преподавателей, которые читают хоть одну лекцию

```
select surname from teacher where exists (select * from timetable where timetable.id_teacher = teacher.id_teacher)
```

	surname text
1	Ivanova
2	Filipov
3	Larin

8.2. Вывести те группы и их курс, где количество студентов больше, чем в любой другой 3 курса

select id_group, course from "group" where number_of_students > all (select number_of_students from "group" where course=3)

	id_group [PK] integer	course integer
1	7	4
2	10	1

9. Использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.)

- 9.1. Вывести объединение строк таблицы студенты, в которых выполняется какое-либо из условий: фамилия студента Иванов, год выпуска студента больше 2000

select surname, year_of_receipt from student where year_of_receipt>2000 union select surname, year_of_receipt from student where surname='Ivanov'

	surname text	year_of_receipt integer
1	Berg	2009
2	Ivanov	1987

- 9.2. Вывести пересечение строк время занятия и количество дисциплин из таблицы расписание по условиям: 3 дисциплины, время больше или равно 12:30

select class_time, number_of_discipline from timetable where class_time>='12:30:00' intersect select class_time, number_of_discipline from timetable where number_of_discipline=3

	class_time time without time zone	number_of_discipline integer
1	14:00:00	3
2	12:30:00	3

10. Использование объединений запросов (inner join и т.д.)


- 10.1. Вывести даты занятий и кто их ведет

select class_data, surname from timetable inner join teacher on timetable.id_teacher = teacher.id_teacher

	class_data date	surname text
1	2020-12-15	Ivanova
2	2020-08-26	Ivanova
3	2020-07-14	Ivanova
4	2020-10-09	Ivanova
5	2020-11-16	Ivanova
6	2018-10-11	Ivanova
7	2019-06-13	Larin
8	2014-03-09	Filipov
9	2019-01-18	Ivanova
10	2017-06-10	Filipov

- 10.2. Вывести все группы и студентов, принадлежащих им (в некоторые группы студенты еще не зачислены)

select "group".id_group, surname from "group" left join student on "group".id_group = student.id_group

	 id_group integer	 surname text	
1	1	Hitman	
2	2	Kravets	
3	2	Ivanov	
4	3	Berg	
5	4	[null]	
6	5	Tess	
7	6	[null]	
8	7	[null]	
9	8	[null]	
10	9	[null]	
11	10	[null]	