

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Кафедра Интеллектуальных систем в гуманитарной сфере, факультет Инфокоммуникационных
технологий

Лабораторная работа №4

РЕАЛИЗАЦИЯ SQL-ЗАПРОСОВ

Выполнил студент гр. № K3242 Кузьмичев К.М.

Проверил преподаватель Говоров А.И.

Задание №5

Цель работы: Требуется написать определенное количество запросов на определенное количество баллов, зависящее от оценки, на которую претендует студент.

Оборудование: УМК (факультет СПО Университета ИТМО).

Программные средства: PostgreSQL, PGAdmin

Практическое задание:

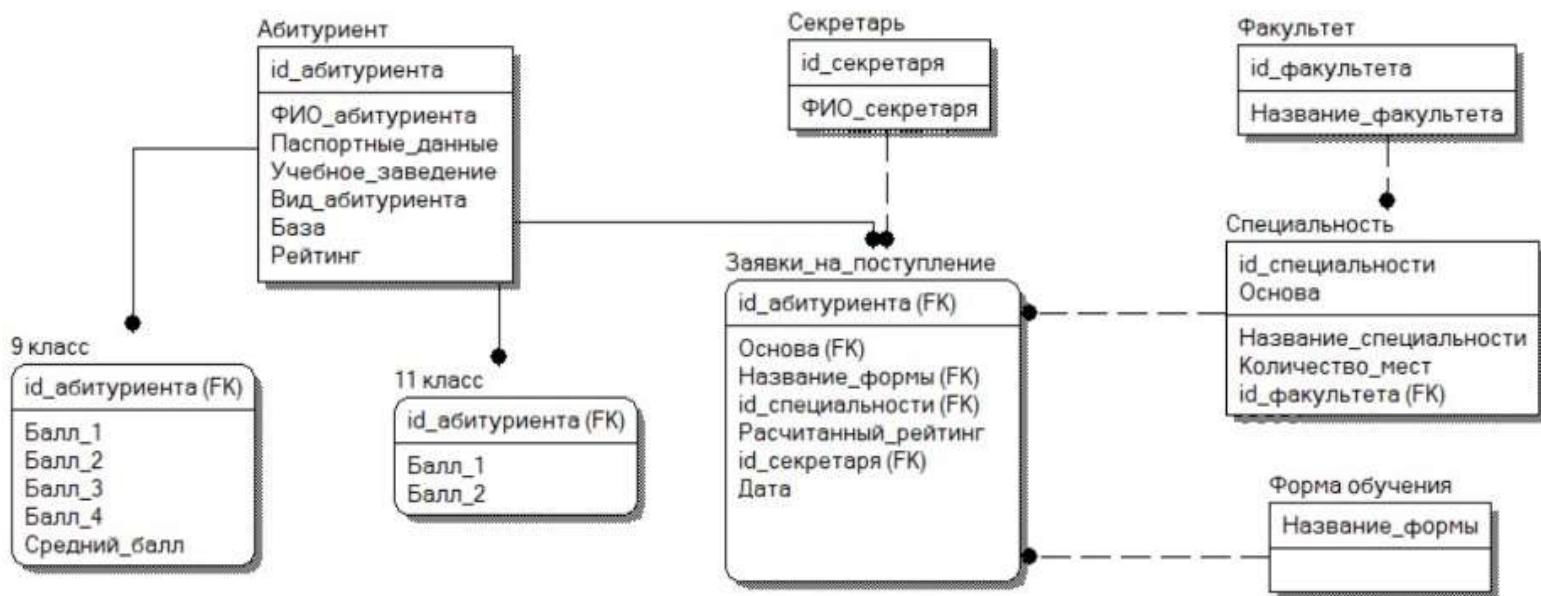
Создать программную систему, предназначенную для работников приемной комиссии колледжа. Она должна обеспечивать хранение, просмотр и изменение сведений об абитуриентах. Результатом работы приемной комиссии должен быть список абитуриентов, зачисленных в колледж. Секретарь приемной комиссии регистрирует абитуриентов. Для каждого абитуриента в базу данных заносятся следующие сведения: фамилия, имя, отчество, паспортные данные, какое учебное заведение, где и когда окончил, наличие золотой или серебряной медали, название специальности, на которые поступает абитуриент. При подаче заявления абитуриент указывает форму обучения (очная, очно-заочная (вечерняя), заочная), поступление на бюджет или контракт. Абитуриент может поступать вне конкурса (инвалиды, сироты). Также существуют абитуриенты-целевики, которые поступают по договорам с направляющими организациями, и обучаются на коммерческой основе. Абитуриенты, поступающие на базе 9 классов, участвуют в конкурсе аттестатов. Для них указывается информация по 4-м профильным дисциплинам и средний балл по всем остальным дисциплинам аттестата. На основе этих данных строится рейтинг абитуриентов. Абитуриенты, поступающие на базе 11 классов, предоставляют сертификаты ЕГЭ по 2 дисциплинам, на основе чего строится рейтинг абитуриентов. Конкурс для абитуриентов на базе 9 и 11 классов отдельный, т.к. они поступают на разные курсы. Абитуриент может не только подать, но и забрать документы, а также перевести их на другую специальность. Известно количество мест на каждый факультет. Приемная комиссия по результатам экзаменов должна сформировать списки абитуриентов, зачисленных в колледж.

Секретарю приемной комиссии могут потребоваться следующие сведения:

1. Список абитуриентов, подавших заявление на заданную специальность.
2. Количество абитуриентов, подавших заявления на каждую специальность по каждой форме обучения на бюджет (или контракт).
3. Количество абитуриентов на базе 9 и 11 классов, поступающих на бюджет (или контракт).
4. Общее количество поданных заявлений ежедневно.
5. Конкурс на каждую специальность по каждой форме обучения на бюджет.

Необходимо предусмотреть возможность получения документа, представляющего собой сгруппированный по заданной специальности список абитуриентов по заданной форме обучения, зачисленных в колледж, с указанием набранных ими баллов по аттестату. Отчет должен содержать проходной балл по специальности в целом, а также количество абитуриентов, поступающих на специальность.

Скрин модели БД:



Список запросов:

1. Выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой:
2. Использование условий WHERE, состоящих из более, чем одного условия:
3. Использование функций для работы с датами:
4. Использование строковых функций:
5. Запрос с использованием подзапросов:
6. Вычисление групповой (агрегатной) функции:
7. Вычисление групповой (агрегатной) функции с условием HAVING:
8. использование предикатов EXISTS, ALL, SOME и ANY:
9. использование запросов с операциями реляционной алгебры:
10. использование объединений запросов:

Выполнение запросов:

Для начала выведем запросы, фигурирующие в задании –

1.Список абитуриентов, подавших документы на заданную специальность (Finance специальность в данном случае):

Выполнены пункты: 10

```
SELECT "Enrollee"."Name", "Specialty"."Name", "Specialty"."Basis"  
FROM "Students"."Enrollee"  
INNER JOIN "Students"."Request"  
ON "Enrollee"."ID_Enrollee" = "Request"."ID_Enrollee"  
INNER JOIN "Students"."Specialty"  
ON "Specialty"."ID_Specialty" = "Request"."ID_Specialty"  
WHERE "Specialty"."Name" = 'Finance'
```

	Name text	Name text	Basis text
1	Danil	Finance	Budget
2	Lana	Finance	Contract
3	Rick	Finance	Contract

2.Количество абитуриентов, подавших заявления на каждую специальность по каждой форме обучения на контракт:

Выполнены пункты: 6, 10

```
SELECT "Specialty"."Name", "Request"."Form", COUNT("Request"."ID_Enrollee")
FROM "Students"."Request"
INNER JOIN "Students"."Specialty"
ON "Request"."ID_Specialty" = "Specialty"."ID_Specialty"
INNER JOIN "Students"."Form"
ON "Request"."Form" = "Form"."Type"
WHERE "Specialty"."Basis" = 'Contract'
GROUP BY "Specialty"."Name", "Request"."Form"
```

	Name text	Form text	count bigint
1	AI	Extramural	1
2	Banking	Part-time	1
3	Finance	Extramural	1
4	Finance	Part-time	1
5	Translator	Extramural	1

3. Кол-во абитуриентов на базе 9 и 11 классов, поступающих на бюджет (с сортировкой):

Выполнены пункты: 1, 6, 10

```
SELECT "Enrollee"."Base", COUNT("Enrollee"."Base")
FROM "Students"."Enrollee"
INNER JOIN "Students"."Request"
ON "Enrollee"."ID_Enrollee" = "Request"."ID_Enrollee"
INNER JOIN "Students"."Specialty"
ON "Specialty"."ID_Specialty" = "Request"."ID_Specialty"
WHERE "Specialty"."Basis" = 'Budget'
GROUP BY "Enrollee"."Base"
ORDER BY "Enrollee"."Base"
```

	Base integer	count bigint
1	9	3
2	11	2

4.Общее кол-во поданных заявлений ежедневно:

Выполнены пункты: 6

```
SELECT "Request"."Date" AS Date, COUNT("Request"."Date")  
FROM "Students"."Request"  
GROUP BY Date  
ORDER BY Date
```

	Date date	count bigint
1	2018-07-08	1
2	2018-07-10	1
3	2018-07-12	1
4	2018-07-13	1
5	2018-07-15	2
6	2018-07-16	1
7	2018-07-17	1
8	2018-07-18	1
9	2018-07-20	1

5.Конкурс на каждую специальность по каждой форме обучения на бюджет:

Выполнены пункты: 6, 10

```
SELECT "Specialty"."Name" AS specialty, ROUND(CAST(COUNT("Request"."ID_Enrollee") AS NUMERIC)
/ CAST("Specialty"."Number_of_places" AS NUMERIC), 2) AS competition
FROM "Students"."Request"
INNER JOIN "Students"."Specialty"
ON "Request"."ID_Specialty" = "Specialty"."ID_Specialty"
WHERE "Specialty"."Basis" = 'Budget'
GROUP BY "Specialty"."Name", "Specialty"."Number_of_places"
```

	specialty text	competition numeric
1	AI	0.07
2	Banking	0.40
3	Finance	0.07
4	Translator	0.10

Теперь остальные реализованные запросы –

6.Выдать кол-во заявлений, поданных в период с 2018-07-13 по 2018-07-16 (предикат BETWEEN):

Выполнены пункты: 8, 2

```
SELECT COUNT("Request"."Date"), "Request"."Date"  
FROM "Students"."Request"  
WHERE "Request"."Date" BETWEEN '2018-07-13' AND '2018-07-16'  
GROUP BY "Request"."Date"
```

	count bigint		Date date
1	1		2018-07-13
2	2		2018-07-15
3	1		2018-07-16

7. Вывод таблицы среднего балла по каждой специальности на базе 11 классов (использование HAVING):

Выполнены пункты: 7

```
SELECT "Specialty"."Name", ROUND(AVG("Request"."Rating"), 2)
FROM "Students"."Request"
INNER JOIN "Students"."Specialty"
ON "Request"."ID_Specialty" = "Specialty"."ID_Specialty"
INNER JOIN "Students"."Enrollee"
ON "Request"."ID_Enrollee" = "Enrollee"."ID_Enrollee"
GROUP BY "Specialty"."Name", "Enrollee"."Base"
HAVING "Enrollee"."Base" = '11'
```

	Name text	round numeric
1	AI	142.00
2	Banking	157.00
3	Finance	164.00
4	Translator	170.00

8. Вывод таблицы факультетов без специальностей:

Выполнены пункты: 9

```
SELECT "Faculty"."ID_Faculty", "Faculty"."Name", "Specialty"."Name" FROM "Students"."Faculty"  
LEFT JOIN "Students"."Specialty"  
ON "Faculty"."ID_Faculty" = "Specialty"."ID_Faculty"  
WHERE "Specialty"."ID_Specialty" IS NULL
```

	ID_Faculty integer		Name text		Name text
1		5	Arts		[null]
2		4	History		[null]

9. Вывод кросс-таблицы количества абитуриентов, подавших заявления на каждую специальность на контракт и бюджет:

Выполнены пункты: 1, 8, 10

```
CREATE EXTENSION IF NOT EXISTS TABLEFUNC;  
  
SELECT *  
  
FROM CROSSTAB2(  
  
    SELECT "Specialty"."Name" AS Specialty, "Specialty"."Basis",  
    COUNT("Request"."ID_Enrollee")::TEXT  
  
    FROM "Students"."Request"  
  
    INNER JOIN "Students"."Specialty"  
  
    ON "Request"."ID_Specialty" = "Specialty"."ID_Specialty"  
  
    GROUP BY "Specialty"."Name", "Specialty"."Basis"  
  
    ORDER BY "Specialty"."Name", "Specialty"."Basis"  
  
    )  
  
AS ct (Specialty , Budget , Contract);
```

	specialty text	budget text	contract text
1	AI	1	1
2	Banking	2	1
3	Finance	1	2
4	Translator	1	1

10. Кол-во принятых заявлений по дням недели с сортировкой по убыванию:

Выполнены пункты: 3

```
SELECT TO_CHAR("Request"."Date", 'Day') as weekday, COUNT(TO_CHAR("Request"."Date", 'Day')) as  
capacity
```

```
FROM "Students"."Request"
```

```
GROUP BY weekday
```

```
ORDER BY capacity DESC
```

	weekday text	capacity bigint
1	Sunday	3
2	Tuesday	2
3	Friday	2
4	Thursday	1
5	Monday	1
6	Wednesday	1

11. Вывод имен и рейтинга абитуриентов на базе 11-х классов, где рейтинг абитуриентов выше, чем у абитуриента ('Danil'):

Выполнены пункты: 1, 2, 5, 10

```
SELECT "Enrollee"."Name", "Request"."Rating"
FROM "Students"."Enrollee"
INNER JOIN "Students"."Request"
ON "Enrollee"."ID_Enrollee" = "Request"."ID_Enrollee"
WHERE "Enrollee"."Base" = '11' AND "Request"."Rating" >
(SELECT "Request"."Rating"
FROM "Students"."Enrollee"
INNER JOIN "Students"."Request"
ON "Enrollee"."ID_Enrollee" = "Request"."ID_Enrollee"
WHERE "Enrollee"."Name" = 'Danil')
```

	Name text	Rating integer
1	Dony	168
2	Alex	170

12. Вывод перевернутых имен абитуриентов (строковая функция):

Выполнены пункты: 4

```
SELECT REVERSE("Enrollee"."Name")
```

```
FROM "Students"."Enrollee"
```

	reverse text
1	lliriK
2	aniloP
3	linaD
4	anaL
5	eillA
6	ynoD
7	xelA
8	annA
9	kciR
10	drahciR

Вывод: в процессе выполнения данной работы были реализованы некоторые запросы к базе данных, содержащие в себе различные функции, условия и операции; получены навыки работы с SQL запросами.