

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»
Направление подготовки «45.03.04 Интеллектуальные системы в гуманитарной сфере»

О Т Ч Е Т

по лабораторной работе №4

Тема задания: Запросы SQL. Выборка данных.

Выполнил:

Студент Коровин Д.К. К3243
(Фамилия И.О.) номер группы

Проверил:

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

**Санкт-Петербург
2020**

Цель работы: Овладеть практическими навыками реализации SQL-запросов.

Практическое задание: Требуется написать определенное количество запросов на определенное количество баллов. Виды запросов:

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 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 раздельный, 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

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.

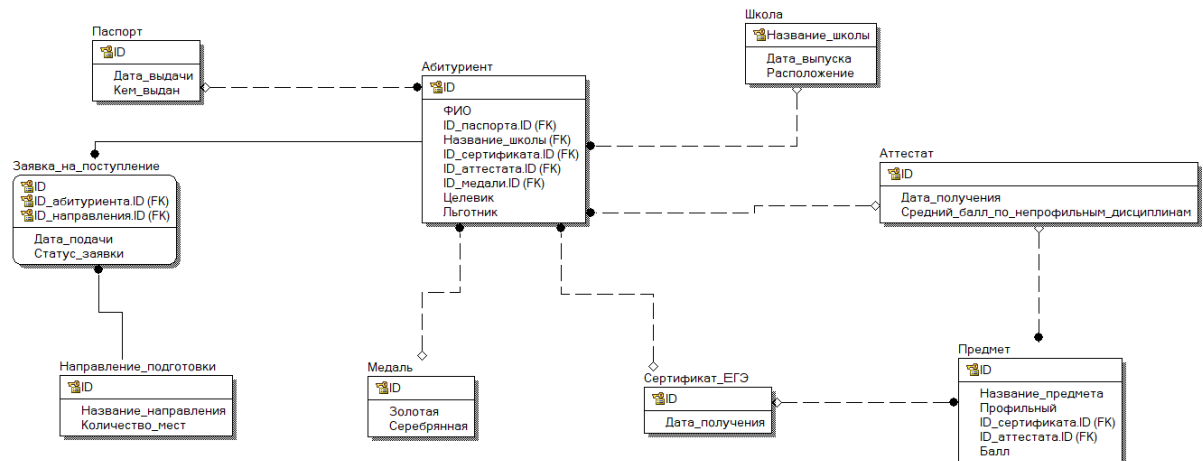
Необходимо предусмотреть возможность получения документа, представляющего собой сгруппированный по заданной специальности список абитуриентов по заданной форме обучения, зачисленных в колледж, с указанием набранных ими баллов по аттестату. Отчет должен содержать проходной балл по специальности в целом,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

Состав реквизитов сущностей:

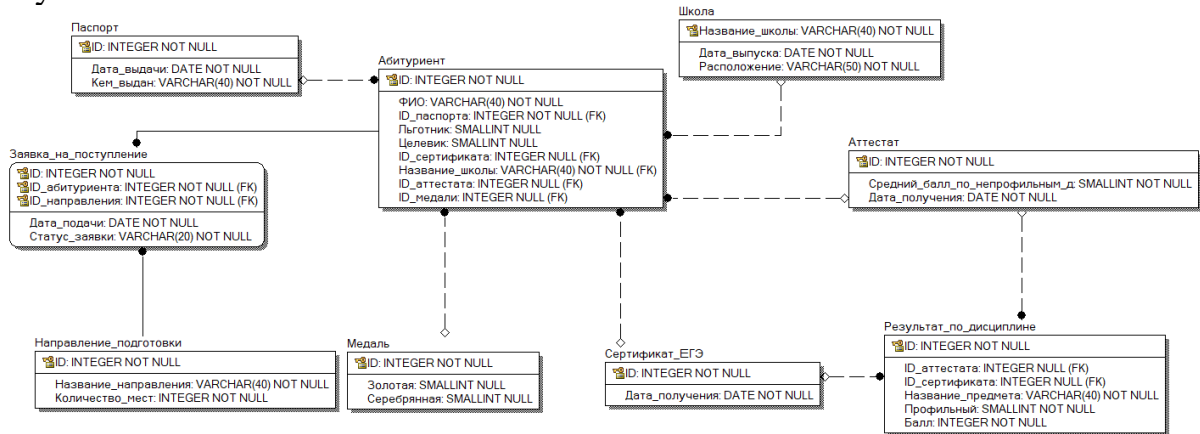
- Абитуриент – ID абитуриента, ФИО, бюджет/контракт, наличие золотой/серебряной медали, наличие льгот/целевого направления, паспортные данные, информация об аттестате или сертификате ЕГЭ, информация о школе;
- Школа – Название школы, дата выпуска абитуриента, расположение школы;
- Аттестат – ID аттестата, дата выдачи, средний балл по непрофильным дисциплинам;
- Сертификат ЕГЭ – ID сертификата, дата получения;
- Паспорт – номер, серия, дата выдачи, кем выдан;
- Заявка на поступление – ID заявки, ID абитуриента, ID направления, дата подачи, статус заявки;
- Направление подготовки – ID направления, название направления, количество мест;
- Медаль – ID медали, тип медали;
- Результат по дисциплине – ID дисциплины, название дисциплины, ID аттестата/сертификата егэ, балл, профильность

Схема базы данных:

Logical:



Physical:



Запросы:

1. Вывод абитуриентов, имеющих медаль, их номера паспорта и вида медали.

```

SELECT enroll_comission."Enrolee"."Name", enroll_comission."Enrolee"."Passport_ID", enroll_comission."Medal"."Type"
FROM enroll_comission."Enrolee"
INNER JOIN enroll_comission."Medal" ON enroll_comission."Enrolee"."Medal_ID" = enroll_comission."Medal"."ID"
ORDER BY enroll_comission."Enrolee"."Name"
    
```

	Name character (40)	Passport_ID character (10)	Type medal_type
1	Familiya Imya Otch...	4114098765	golden
2	Ivanov Ivan Ivanovi...	4112123456	silver
3	Privet Eto Ya ...	4115228359	silver

2. Вывод школ, которые находятся в Москве или Санкт-Петербурге, сортировка по названию школы в обратном порядке.

```
SELECT enroll_comission."School"."Name", enroll_comission."School"."School_location"
FROM enroll_comission."School"
WHERE enroll_comission."School"."School_location" = 'Moscow'
OR enroll_comission."School"."School_location" = 'Saint-Petersburg'
ORDER BY enroll_comission."School"."Name" DESC
```

	Name [PK] character varying (40)	School_location character varying (50)
1	School #3	Moscow
2	School #2	Saint-Petersburg
3	School #1	Moscow

3. Вывод ФИО абитуриента, его номера сертификата ЕГЭ, даты выдачи сертификата, даты выпуска и времени, прошедшего с выпускного и до получения сертификата

```
SELECT enroll_comission."Enrolee"."Name", enroll_comission."Enrolee"."EGE_certificate_ID", enroll_comission."EGE_certificate"."Issue_date",
enroll_comission."School"."Graduation_date", age(enroll_comission."EGE_certificate"."Issue_date", enroll_comission."School"."Graduation_date")
FROM enroll_comission."Enrolee"
INNER JOIN enroll_comission."School" ON enroll_comission."Enrolee"."School_name" = enroll_comission."School"."Name"
INNER JOIN enroll_comission."EGE_certificate" ON enroll_comission."Enrolee"."EGE_certificate_ID" = enroll_comission."EGE_certificate"."ID"
```

	Name character (40)	EGE_certificate_ID integer	Issue_date date	Graduation_date date	age interval
1	Ivanov Ivan Ivanovi...	1	2019-07-01	2019-06-20	11 days
2	Dugin Alexander Ge...	2	2019-07-01	2019-06-20	11 days
3	Familiya Imya Otch...	3	2019-07-01	2019-06-20	11 days
4	Privet Eto Ya ...	4	2019-07-01	2019-06-20	11 days
5	Govorov Anton Igor...	5	2019-07-01	2019-06-20	11 days

4. Замена пробелов в ФИО абитуриента на подчеркивания

```
SELECT replace(enroll_comission."Enrolee"."Name", ' ', '_') as edited_name
FROM enroll_comission."Enrolee"
```

	edited_name text
1	Ivanov_Ivan_Ivanovich
2	Dugin_Alexander_Gelevich
3	Familiya_Imya_Otchestvo
4	Privet_Eto_Ya
5	Govorov_Anton_Igorevich

5. Вывод имен абитуриентов, ФИО которых имеет более 15 символов, номеров их паспортов и длину (посимвольно) номеров

```
SELECT enroll_comission."Enrolee"."Name", enroll_comission."Passport"."ID" , char_length(enroll_comission."Passport"."ID") as length
FROM enroll_comission."Passport"
INNER JOIN enroll_comission."Enrolee" ON enroll_comission."Enrolee"."Passport_ID" = enroll_comission."Passport"."ID"
WHERE length(enroll_comission."Enrolee"."Name") > 15
```

	Name character (40)	ID character (10)	length integer
1	Ivanov Ivan Ivanovi...	4112123456	10
2	Dugin Alexander Ge...	4113654321	10
3	Familiya Imya Otch...	4114098765	10
4	Govorov Anton Igor...	4116114488	10

6. Вывод имени студента и его среднего балла

```
SELECT enroll_comission."Enrolee"."Name", AVG(enroll_comission."Subj_res"."Grade") as average_grade
FROM enroll_comission."Enrolee"
RIGHT JOIN enroll_comission."School_certificate"
ON enroll_comission."Enrolee"."School_certificate_ID" = enroll_comission."School_certificate"."ID"
LEFT JOIN enroll_comission."Subj_res"
ON enroll_comission."School_certificate"."ID" = enroll_comission."Subj_res"."School_certificate_ID"
GROUP BY enroll_comission."Enrolee"."Name" ORDER BY enroll_comission."Enrolee"."Name" ASC
```

	Name character (40)	average_grade numeric
1	Dugin Alexander Ge...	3.0000000000000000
2	Familiya Imya Otch...	5.0000000000000000
3	Govorov Anton Igor...	5.0000000000000000
4	Ivanov Ivan Ivanovi...	4.0000000000000000
5	Privet Eto Ya ...	4.0000000000000000

7. Вывод имени студента, паспортных данных и информации о школе используя объединение запросов

```
SELECT enroll_comission."Enrolee"."Name", enroll_comission."Passport".*, enroll_comission."School".*
FROM enroll_comission."Enrolee"
INNER JOIN enroll_comission."Passport" ON enroll_comission."Enrolee"."Passport_ID" = enroll_comission."Passport"."ID"
INNER JOIN enroll_comission."School" ON enroll_comission."Enrolee"."School_name" = enroll_comission."School"."Name"
```

	Name character (40)	ID character (10)	Issue_date date	Issued_by character varying (100)	Name character varying (40)	Graduation_date date	School_location character varying (50)
1	Ivanov Ivan Ivanovi...	4112123456	2015-05-10	MVD RF	School #1	2019-06-20	Moscow
2	Dugin Alexander Ge...	4113654321	2015-04-28	MVD RF	School #2	2019-06-20	Saint-Petersburg
3	Familiya Imya Otch...	4114098765	2015-07-16	MVD RF	School #3	2019-06-20	Moscow
4	Privet Eto Ya ...	4115228359	2015-02-14	MVD RF	School #4	2019-06-20	Kazan
5	Govorov Anton Igor...	4116114488	2015-09-04	MVD RF	School #5	2019-06-20	Orel

8. Вывод количества принятых (number_enrolled), непринятых (number_rejected) и тех, чьи заявки обрабатываются (number_processing), с использованием подзапросов

```

SELECT approved.*, rejected.*, in_process.* FROM
  (SELECT COUNT(enroll_comission."Enrolee"."Name") as number_enrolled
   FROM enroll_comission."Enrolee", enroll_comission."Application"
   WHERE enroll_comission."Enrolee"."ID" = enroll_comission."Application"."Enrolee_ID"
   AND enroll_comission."Application"."Status" = 'Approved') AS approved,
  (SELECT COUNT(enroll_comission."Enrolee"."Name") as number_rejected
   FROM enroll_comission."Enrolee", enroll_comission."Application"
   WHERE enroll_comission."Enrolee"."ID" = enroll_comission."Application"."Enrolee_ID"
   AND enroll_comission."Application"."Status" = 'Rejected') AS rejected,
  (SELECT COUNT(enroll_comission."Enrolee"."Name") as number_processing
   FROM enroll_comission."Enrolee", enroll_comission."Application"
   WHERE enroll_comission."Enrolee"."ID" = enroll_comission."Application"."Enrolee_ID"
   AND enroll_comission."Application"."Status" = 'In process') AS in_process

```

number_enrolled bigint	number_rejected bigint	number_processing bigint
3	1	1

9. Вывод всей информации о направлении, на котором больше всего свободных мест

```

SELECT DISTINCT enroll_comission."Course".*
FROM enroll_comission."Course"
WHERE NOT enroll_comission."Course"."Available_slots" < SOME (SELECT enroll_comission."Course"."Available_slots" FROM enroll_comission."Course")

```

ID [PK] integer	Name character varying (40)	Available_slots integer
3	Course name 3	300

10. Вывод среднего балла ЕГЭ у всех, чьи заявки были одобрены

```

SELECT AVG(enroll_comission."Subj_res"."Grade") as average_grade
FROM enroll_comission."Enrolee"
RIGHT JOIN enroll_comission."EGE_certificate" ON enroll_comission."Enrolee"."EGE_certificate_ID" = enroll_comission."EGE_certificate"."ID"
LEFT JOIN enroll_comission."Subj_res" ON enroll_comission."EGE_certificate"."ID" = enroll_comission."Subj_res"."EGE_certificate_ID"
LEFT JOIN enroll_comission."Application" ON enroll_comission."Application"."Enrolee_ID" = enroll_comission."Enrolee"."ID"
GROUP BY enroll_comission."Application"."Status" HAVING enroll_comission."Application"."Status" = 'Approved'

```

average_grade numeric
79.6666666666666667

11. Вывод имени абитуриента и информации, на бюджете он или нет

```

SELECT enroll_comission."Enrolee"."Name", enroll_comission."Enrolee"."Budget" as is_smart_enough
FROM enroll_comission."Enrolee"
WHERE enroll_comission."Enrolee"."Budget" = 'true'
UNION
SELECT enroll_comission."Enrolee"."Name", enroll_comission."Enrolee"."Budget" as is_smart_enough
FROM enroll_comission."Enrolee"
WHERE enroll_comission."Enrolee"."Budget" = 'false'
ORDER BY is_smart_enough

```

	Name character (40)	is_smart_enough boolean
1	Privet Eto Ya ...	false
2	Dugin Alexander Ge...	true
3	Govorov Anton Igor...	true
4	Familiya Imya Otch...	true
5	Ivanov Ivan Ivanovi...	true

12. Вывод ФИО, номера паспорта, названия школы и расположения школы абитуриентов, школы которых находятся в городе, название которого заканчивается на 'ow'

```
SELECT enroll_comission."Enrolee"."Name", enroll_comission."Enrolee"."Passport_ID", enroll_comission."Enrolee"."School_name", enroll_comission."School"."School_location"
FROM enroll_comission."Enrolee"
INNER JOIN enroll_comission."School" ON enroll_comission."School"."Name" = enroll_comission."Enrolee"."School_name"
WHERE EXISTS
(SELECT enroll_comission."School"."School_location"
FROM enroll_comission."School"
WHERE enroll_comission."School"."Name" = enroll_comission."Enrolee"."School_name"
AND enroll_comission."School"."School_location" LIKE '%ow')
```

	Name character (40)	Passport_ID character (10)	School_name character varying (40)	School_location character varying (50)
1	Ivanov Ivan Ivanovi...	4112123456	School #1	Moscow
2	Familiya Imya Otch...	4114098765	School #3	Moscow

Вывод: В ходе работы получил практические навыки выполнения SQL-запросов разных уровней сложности и развил фантазию, пока придумывал запросы.