**Лабораторная работа по теме “Язык запросов SQL”**

1. Необходимо установить базу данных PostgreSQL (***под администратором системы***),

для Windows (инсталлятор и документация) : https://postgrespro.ru/windows

для Linux (инсталлятор и документация) : https://www.postgresql.org/download/

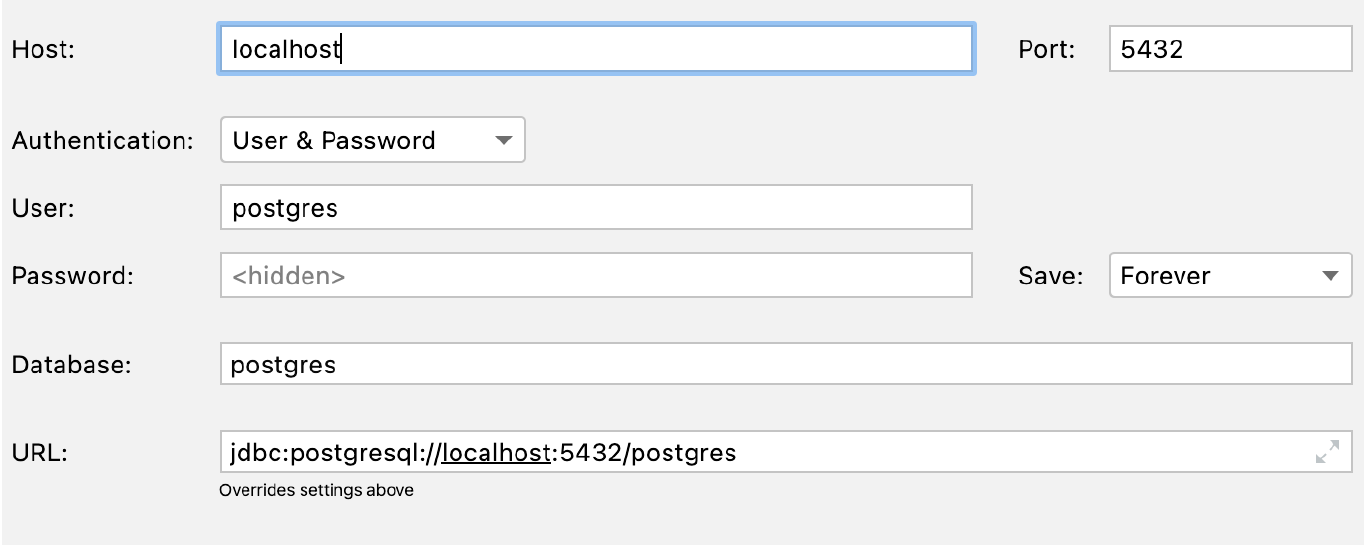
1. Скачать **одно** из приложений

- pgAdmin - https://www.pgadmin.org/

- dbeaver - https://dbeaver.io/

- DataGrip - https://www.jetbrains.com/ru-ru/datagrip/

1. Подключиться приложением из пункта №2 к базе данных из пункта №1



Необходимо указать

- **Host** - IP адрес компьютера (сервера) где находится установленная база данных из пункта №1

- **User** - пользователь (оставить postgres)

- **Port** - порт (оставить 5432 если вы не меняли этот порт в настройках)

- **Password** - пароль (который вы указывали при инсталляции)

- **Database** - база данных (оставить postgres)

Примените пожалуйста скрипт для базы данных PostgreSQL. Если есть необходимость, то вы можете применить его заново, предварительно удалив таблицы командой

DROP TABLE *имя\_таблицы;*

Применить скрипт - это значит выполнить его через ваш используемый IDE

(DataGrip, DBeaver , pgAdmin)

Необходимо написать SQL запросы к следующим задачам ниже. SQL запрос пишется в вашем IDE и прикладывается (как текст) вместе со скрином ответа в doc файле - который в свою очередь необходимо приложить к домашнему заданию и отправить на платформу. Задание считается выполненным - если SQL запрос написан синтаксически корректно и возвращает ожидаемые данные на условие задачи.

1. Напишите SQL запрос который возвращает имена студентов и их аккаунт в Telegram у которых родной город “Казань” или “Москва”. Результат отсортируйте по имени студента в убывающем порядке
2. Напишите SQL запрос который возвращает данные по университетам в следующем виде (один столбец со всеми данными внутри) с сортировкой по полю *“полная информация”*
3. Напишите SQL запрос который возвращает список университетов и количество студентов, если идентификатор университета должен быть выбран из списка 10, 30, 50. Пожалуйста примените конструкцию IN.

Результат запроса отсортируйте по количеству студентов **И** затем по

наименованию университета.

1. Напишите SQL запрос который возвращает список университетов и количество студентов, если идентификатор университета **НЕ** должен соответствовать значениям из списка 10, 30, 50. Пожалуйста в основе примените конструкцию IN. Результат запроса отсортируйте по количеству студентов **И** затем по наименованию университета.
2. Напишите SQL запрос который возвращает название online курсов университетов и количество заявленных слушателей. Количество заявленных слушателей на курсе должно быть в диапазоне от 27 до 310 студентов. Результат отсортируйте по названию курса и по количеству заявленных слушателей в убывающем порядке для двух полей. f. Напишите SQL запрос который возвращает имена студентов и название курсов университетов в одном списке. Результат отсортируйте в убывающем порядке. Пример части результата представлен ниже
3. Напишите SQL запрос который возвращает имена студентов и название курсов университетов в одном списке. Результат отсортируйте в убывающем порядке. Пример части результата представлен ниже
4. Напишите SQL запрос который возвращает имена университетов и название курсов в одном списке, но с типом что запись является или “университет” или “курс”. Результат отсортируйте в убывающем порядке по типу записи и потом по имени. Пример части результата представлен ниже
5. Напишите SQL запрос который возвращает название курса и количество заявленных студентов в отсортированном списке по количеству слушателей в возрастающем порядке, **НО** запись с количеством слушателей равным 300 должна быть на первом месте. Ограничьте вывод данных до 3 строк. Пример результата представлен ниже

**Подсказка**: используйте в ORDER BY синтаксический

элемент CASE … END. Саму конструкцию можно посмотреть вот здесь

1. Напишите DML запрос который создает новый **offline** курс со следующими характеристиками:

* id = 60
* название курса = Machine Learning
* количество студентов = 17
* курс проводится в том же университете что и курс Data Mining

Предоставьте INSERT выражение которое заполняет необходимую

таблицу данными

Приложите скрин результата запроса к данным курсов после

выполнения команды INSERT к таблице которая была изменена.

1. Напишите SQL скрипт который подсчитывает симметрическую разницу множеств A и B.

(A \ B) ⋃ (B \ A)

где A - таблица **course,** B - таблица **student\_on\_course,** “\” - это разница множеств, “⋃” - объединение множеств. Необходимо подсчитать на основании атрибута **id** из обеих таблиц. Результат отсортируйте по 1 столбцу. Пример результата представлен ниже.

1. Напишите SQL запрос который вернет имена студентов, курс на котором

они учатся, названия их родных университетов (в которых они официально учатся) и соответствующий рейтинг по курсу. С условием что рассматриваемый рейтинг студента должен быть строго больше (>) 50 баллов и размер соответствующего ВУЗа должен быть строго больше (>) 5000 студентов. Результат необходимо отсортировать по первым двум

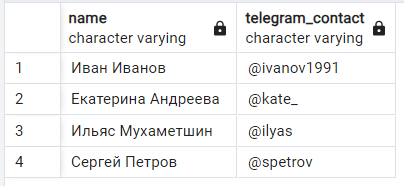
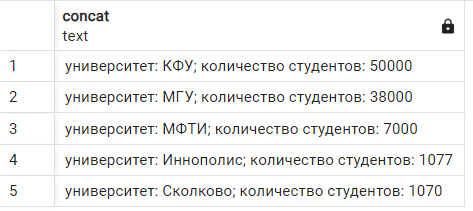
столбцам. Обратите внимание на часть ответа ниже с учетом **именования выходных атрибутов** вашего запроса

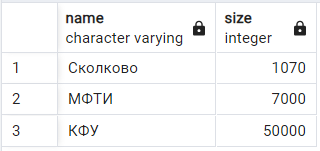
1. Выведите уникальные семантические пары студентов, родной город которых один и тот же. Результат необходимо отсортировать по первому столбцу. Семантически эквивалентная пара является пара студентов например (Иванов, Петров) = (Петров, Иванов), в этом случае должна быть выведена одна из пар. Обратите внимание на ответ ниже с учетом **именования выходных атрибутов** вашего запроса
2. Напишите SQL запрос который возвращает количество студентов, сгруппированных по их **оценке**. Результат отсортируйте по названию оценки студента. Формула выставления оценки представлена ниже как псевдокод. ЕСЛИ оценка < 30 ТОГДА неудовлетворительно ЕСЛИ оценка >= 30 И оценка < 60 ТОГДА удовлетворительно ЕСЛИ оценка >= 60 И оценка < 85 ТОГДА хорошо В ОСТАЛЬНЫХ СЛУЧАЯХ отлично

Пример результата ниже. Обратите внимание на именование результирующих столбцов в вашем решении. Курс “Machine Learning”, так как у него нет студентов - проигнорируйте, используя соответствующий тип JOIN.

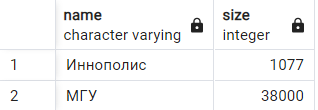
1. Дополните SQL запрос из задания a), с указанием вывода имени курса и количество оценок внутри курса. Результат отсортируйте по названию курса и оценки студента. Пример части результата ниже. Обратите внимание на именование результирующих столбцов в вашем решении. Курс “Machine Learning”, так как у него нет студентов - проигнорируйте, используя соответствующий тип JOIN.

**Выполнение работы:**

1. select name, telegram\_contact from student where city in ('Казань','Москва')  
   
2. select concat('университет: ',name,'; количество студентов: ', size) from college  
   
3. select name, size from college where id in (10,30,50) order by size, name



1. select name, size from college where not id in (10,30,50) order by size, name



1. select name, amount\_of\_students

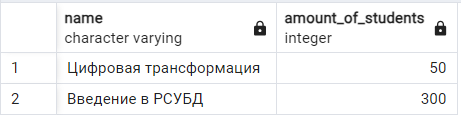
from course

where

is\_online = true

and amount\_of\_students between 27 and 310

order by name desc, amount\_of\_students desc



1. select name

from student

union

select name

from college

order by name desc



1. select name, 'курс' as object\_type

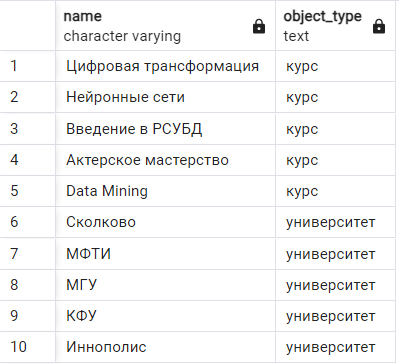
from course

union

select name, 'университет' as object\_type

from college

order by object\_type, name desc

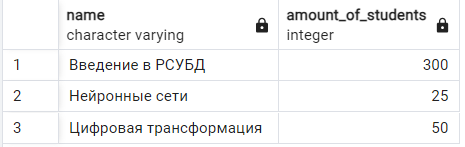


1. select name, amount\_of\_students

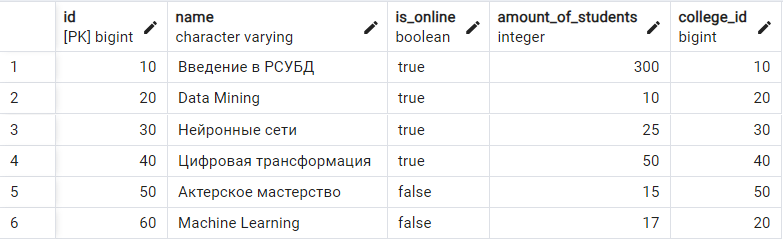
from course

order by case when amount\_of\_students = 300 then 0 end, amount\_of\_students

limit 3



1. insert into course(id,name,is\_online,amount\_of\_students,college\_id) values(60,'Machine Learning',false,17,20)



1. (select a.id from course as a

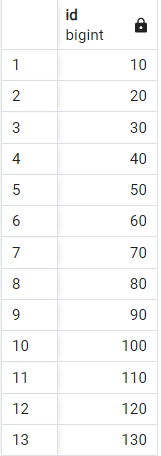
left join student\_on\_course as b on a.id=b.id)

union

(select a.id from student\_on\_course as a

left join course as b on a.id=b.id)

order by id



1. select

s.name as student\_name,

c.name as course\_name,

col.name as student\_colledge,

sc.student\_rating as student\_rating

from student as s

left join student\_on\_course as sc

on s.id = sc.student\_id

left join course as c

on c.id = sc.course\_id

left join college as col

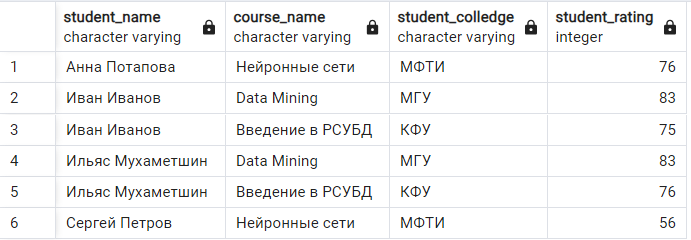
on col.id = c.college\_id

where

student\_rating > 50

and col.size > 5000

order by student\_name,course\_name

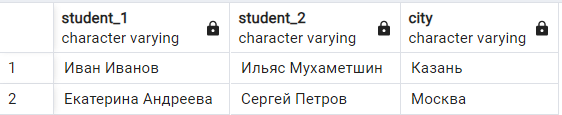


1. select DISTINCT ON (s1.city) s1.name as student\_1, s2.name as student\_2, s1.city as city

from student as s1

join student as s2 on s1.city = s2.city

where s1.name <> s2.name



1. select

case when sc.student\_rating < 30 then 'не удовлетворительно'

when sc.student\_rating >= 30 and sc.student\_rating < 60 then 'удовлетворительно'

when sc.student\_rating >= 60 and sc.student\_rating < 85 then 'хорошо'

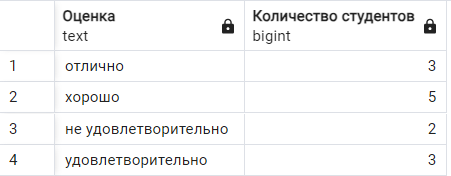
else 'отлично'

end as Оценка,

count(\*) as "Количество студентов"

from student\_on\_course as sc

group by Оценка



1. select c.name as Курс,

case when sc.student\_rating < 30 then 'не удовлетворительно'

when sc.student\_rating >= 30 and sc.student\_rating < 60 then 'удовлетворительно'

when sc.student\_rating >= 60 and sc.student\_rating < 85 then 'хорошо'

else 'отлично'

end as Оценка,

count(\*) as "Количество студентов"

from student\_on\_course as sc

left join course as c on c.id = sc.course\_id

group by Курс, Оценка

order by Курс, Оценка

