Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**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**

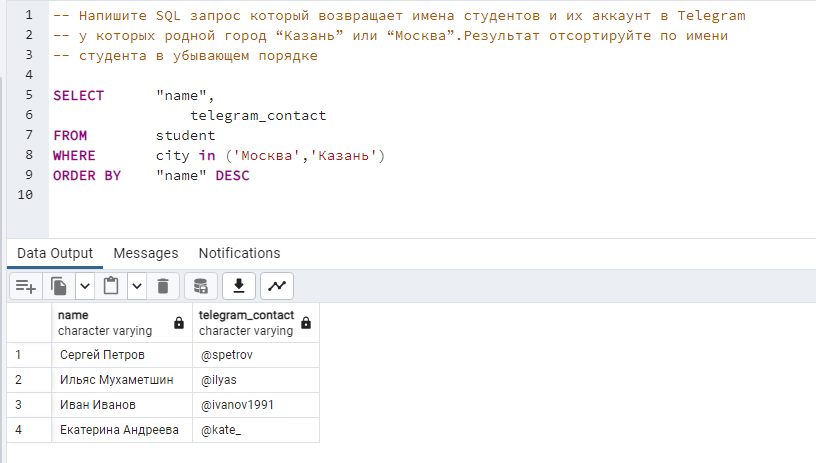
SELECT "name",

telegram\_contact

FROM student

WHERE city in ('Москва','Казань')

ORDER BY "name" DESC



**Напишите SQL запрос который возвращает данные по университетам в следующем виде (один столбец со всеми данными внутри) ссортировкой по полю "полная информация"**

SELECT concat('университет: ',"name",'; количество студентов: ',"size") as Полная\_информация

FROM college

ORDER BY Полная\_информация

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

**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.**

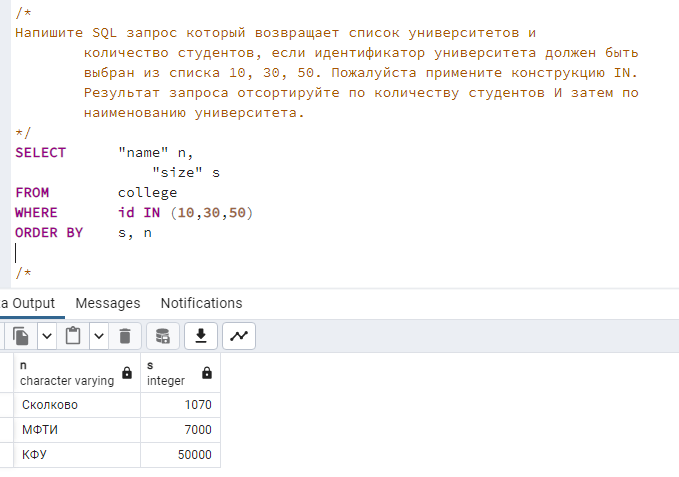
SELECT "name" n,

"size" s

FROM college

WHERE id IN (10,30,50)

ORDER BY s, n



**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.**

SELECT "name" n,

"size" s

FROM college

WHERE id NOT IN (10,30,50)

ORDER BY s, n

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

**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.**

SELECT "name" as n,

amount\_of\_students

FROM course

WHERE is\_online = true

AND amount\_of\_students BETWEEN 27 and 310

ORDER BY n DESC,

amount\_of\_students DESC

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

**Напишите SQL запрос который возвращает имена студентов и название курсов университетов в одном списке. Результат отсортируйте в убывающем порядке**

SELECT "name" as n

FROM course

UNION

SELECT "name"

FROM student

ORDER BY 1 DESC

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

**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**

SELECT "name" as n,

'университет' as object\_type

FROM course

UNION

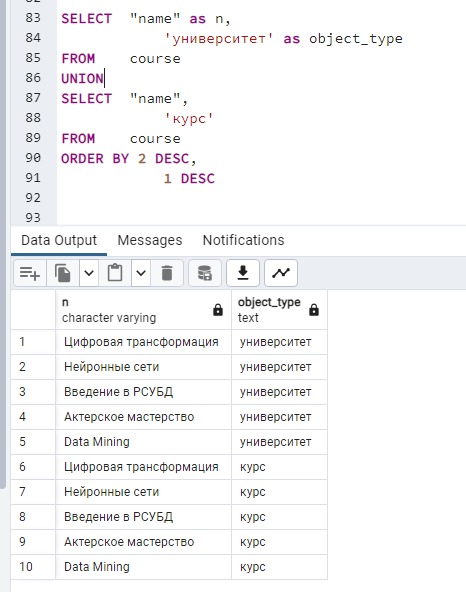
SELECT "name",

'курс'

FROM course

ORDER BY 2 DESC,

1 DESC



**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**

SELECT "name" as n,

amount\_of\_students

FROM course

ORDER BY CASE

WHEN amount\_of\_students = 300 THEN 0

ELSE amount\_of\_students

END

LIMIT 3

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

**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.**

SELECT \*

FROM course

INSERT INTO course

(

"id",

"name",

is\_online,

amount\_of\_students,

college\_id

)

VALUES

(

60,

'Machine Learning',

false,

17,

(

SELECT college\_id

FROM course

WHERE "name" = 'Data Mining'

LIMIT 1

)

)

SELECT \*

FROM course

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

**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.**

SELECT id

FROM

(

SELECT id

FROM course

EXCEPT

SELECT id

FROM student\_on\_course

) AS course

UNION

SELECT id

FROM

(

SELECT id

FROM student\_on\_course

EXCEPT

SELECT id

FROM course

) AS student\_on\_course

ORDER BY 1

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

**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**

SELECT s.name AS nameStudent,

course.name AS nameCourse,

college.name AS nameCollege,

son.student\_rating

FROM student s

INNER JOIN

student\_on\_course son

ON son.student\_id = s.id

AND son.student\_rating > 50

INNER JOIN

course

ON course.id = son.course\_id

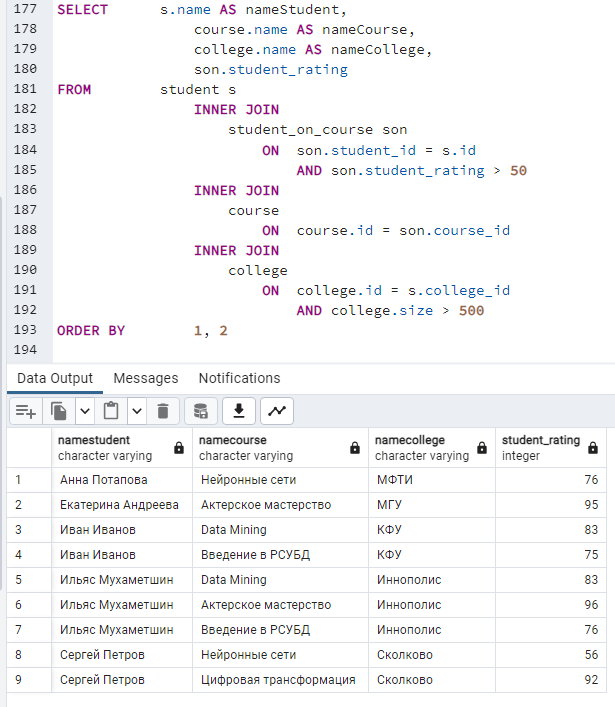
INNER JOIN

college

ON college.id = s.college\_id

AND college.size > 500

ORDER BY 1, 2



**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**

SELECT student1.name AS student\_1,

student2.name AS student\_2,

student1.city

FROM student AS student1

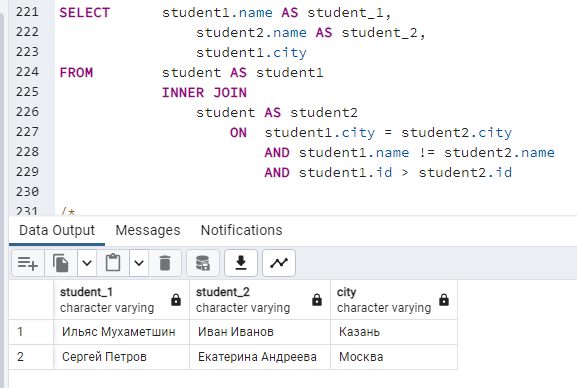
INNER JOIN

student AS student2

ON student1.city = student2.city

AND student1.name != student2.name

AND student1.id > student2.id



**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.**

SELECT CASE

WHEN student\_rating < 30

THEN 'Не удоветворительно'

WHEN student\_rating >= 30

AND student\_rating < 60

THEN 'Удоветворительно'

WHEN student\_rating >= 60

AND student\_rating < 85

THEN 'Хорошо'

ELSE 'Отлично'

END AS оценка,

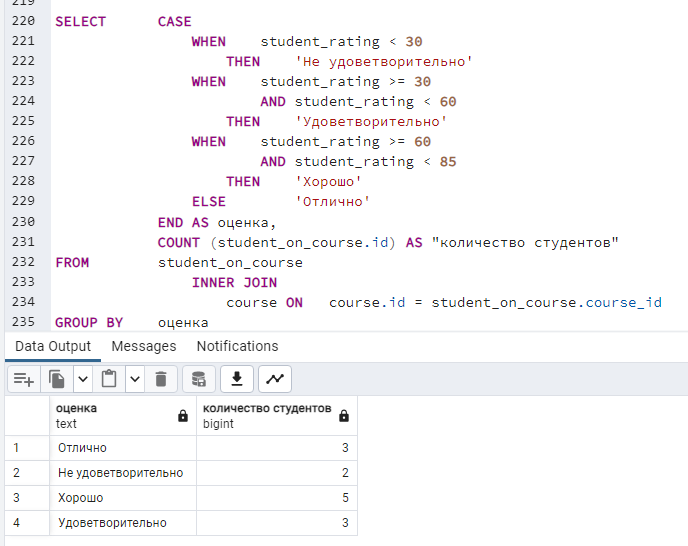
COUNT (student\_on\_course.id) AS "количество студентов"

FROM student\_on\_course

INNER JOIN

course ON course.id = student\_on\_course.course\_id

GROUP BY оценка



**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**

SELECT course.name,

CASE

WHEN student\_rating < 30

THEN 'Не удоветворительно'

WHEN student\_rating >= 30

AND student\_rating < 60

THEN 'Удоветворительно'

WHEN student\_rating >= 60

AND student\_rating < 85

THEN 'Хорошо'

ELSE 'Отлично'

END AS оценка,

COUNT (student\_on\_course.id) AS "количество студентов"

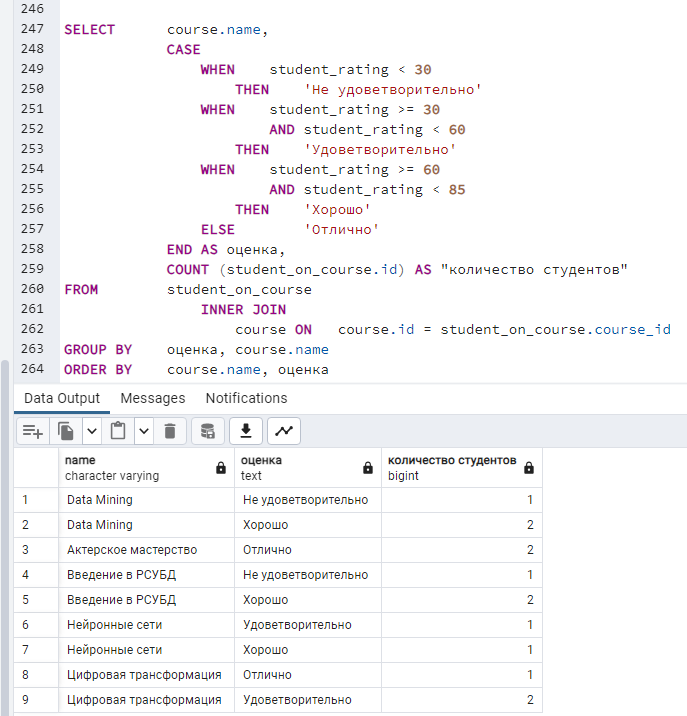
FROM student\_on\_course

INNER JOIN

course ON course.id = student\_on\_course.course\_id

GROUP BY оценка, course.name

ORDER BY course.name, оценка



Ссылка на репозиторий <https://github.com/RustamStudy/II_arc/tree/main/DB_task9_SafiullinRustam>