**"Язык запросов SQL"**

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

**select**

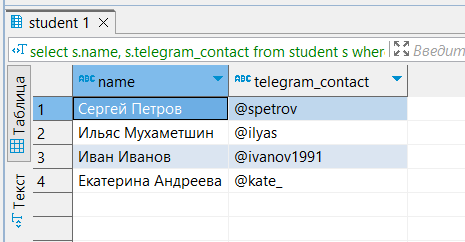
s.**name**,

s.telegram\_contact

**from** student s

**where** s.city = 'Казань' **or** s.city = 'Москва'

**order** **by** s.**name desc**;



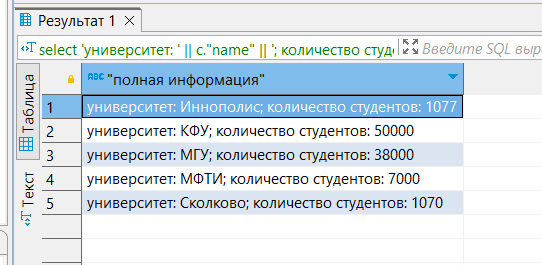
1. Напишите SQL запрос который возвращает данные по университетам в следующем виде (один столбец со всеми данными внутри) с сортировкой по полю "полная информация"

**select**

'университет: ' || c."name" || '; количество студентов: ' || c."size" **as** """полная информация"""

**from** college c

**order** **by** 1;



1. 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

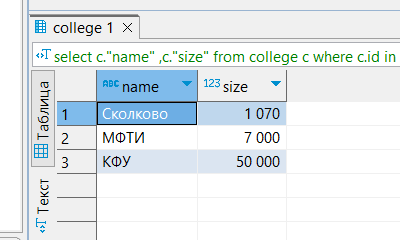
**select**

,c."size"

**from** college c

**where** c.id **in** (10, 30, 50)

**order** **by** c."size", c."name";



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 университетаD

**select**

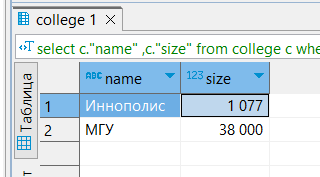
c."name"

,c."size"

**from** college c

**where** c.id **not** **in** (10, 30, 50)

**order** **by** c."size", c."name";



1. 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**

cr."name"

,cr.amount\_of\_students

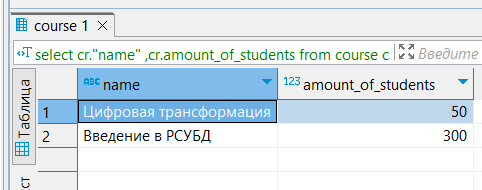
**from** course cr

**where**

cr.is\_online = **true**

**and** cr.amount\_of\_students **between** 27 **and** 310

**order** **by** cr."name" **desc**, cr.amount\_of\_students **desc**;



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

**select**

cr."name"

**from** course cr

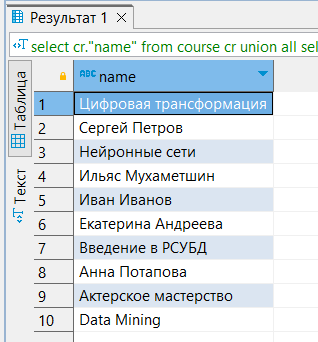
**union** **all**

**select**

s."name"

**from** student s

**order** **by** cr."name" **desc**;



1. 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

**select**

c."name"

, 'университет' **as** object\_type

**from** college c

**union** **all**

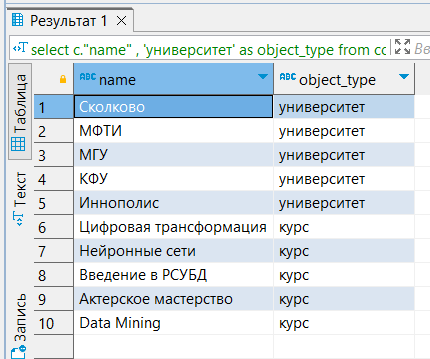
**select**

cr."name"

, 'курс' **as** object\_type

**from** course cr

**order** **by** 2 **desc**, 1 **desc**;



1. 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

**select**

c."name"

,c.amount\_of\_students

**from** course c

**order** **by**

**case**

**when** c.amount\_of\_students = 300 **then** c.amount\_of\_students

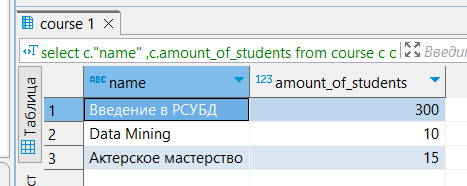
**end**,

**case**

**when** c.amount\_of\_students != 300 **then** c.amount\_of\_students

**end**

**limit** 3;



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

**INSERT** **INTO** public.course

(id, "name", is\_online, amount\_of\_students, college\_id)

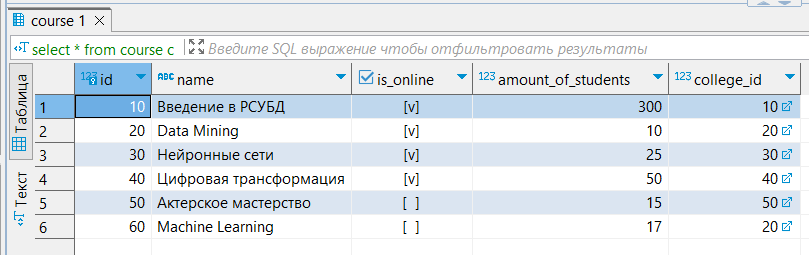
**VALUES**(60, 'Machine Learning', **false**, 17,

(**select** c.college\_id

**from** course c

**where**

c."name" = 'Data Mining'));



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

**select**

c.id

**from** course c

**except**

**select**

id

**from** student\_on\_course soc

**union**

**select**

id

**from** student\_on\_course soc

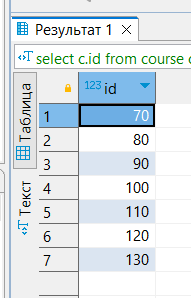
**except**

**select**

c.id

**from** course c

**order** **by** 1;



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

**select**

s."name" **as** student\_name

,c."name" **as** course\_name

,cg."name" **as** student\_college

,soc.student\_rating **as** student\_rating

**from** student\_on\_course soc **join** student s **on** (soc.student\_id = s.id)

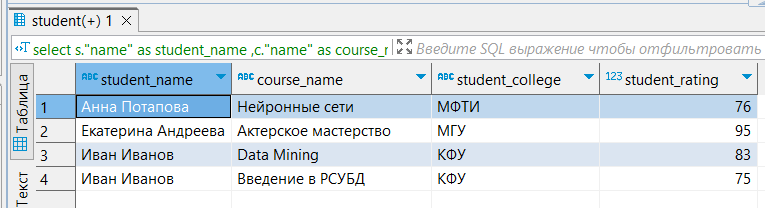
**join** course c **on** (soc.course\_id = c.id)

**join** college cg **on** (s.college\_id = cg.id **and** cg."size" > 5000)

**where**

soc.student\_rating > 50

**order** **by** 1, 2;



1. Выведите уникальные семантические пары студентов, родной город которых один и тот же. Результат необходимо отсортировать по первому столбцу

**select**

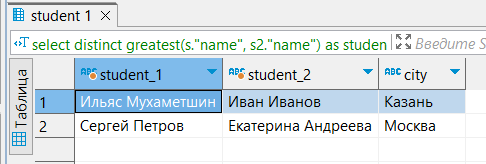
**distinct** **greatest**(s."name", s2."name") **as** student\_1

, **least**(s2."name", s."name") **as** student\_2

,s.city **as** city

**from** student s **join** student s2 **on** (s.city = s2.city **and** s."name" != s2."name")

**order** **by** 1;



1. Напишите SQL запрос который возвращает количество студентов, сгруппированных по их оценке. Результат отсортируйте по названию оценки студента

**select**

**case**

**when** soc.student\_rating < 30 **then** 'неудовлетворительно'

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

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

**when** soc.student\_rating >= 85 **then** 'отлично'

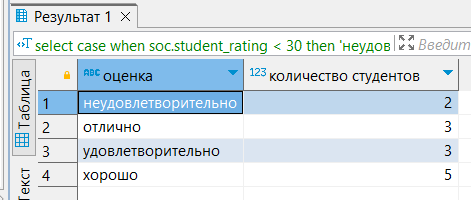
**end** **as** "оценка"

,**count**(soc.student\_id) **as** "количество студентов"

**from** student\_on\_course soc

**group** **by** "оценка"

**order** **by** 1



***\* Не совсем понятно для чего в задании этот комментарий: "Курс “Machine Learning”, так как у него нет студентов - проигнорируйте, используя соответствующий тип JOIN". Все данные берутся из одной готовой таблицы, где данный курс отсутствует***

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

**select**

c."name" **as** "курс"

,**case**

**when** soc.student\_rating < 30 **then** 'неудовлетворительно'

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

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

**when** soc.student\_rating >= 85 **then** 'отлично'

**end** **as** "оценка"

,**count**(soc.student\_id) **as** "количество студентов"

**from** student\_on\_course soc **join** course c **on** (soc.course\_id = c.id)

**group** **by** "оценка", c."name"

**order** **by** 1, 2

