| **№** | **Задача** | **Текст запроса** | **Скриншот выполнения** |
| --- | --- | --- | --- |
|  | 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** ('Казань','Москва')  **order** **by** **name** **desc** |  |
|  | Напишите SQL запрос, который возвращает данные по университетам в следующем виде (один столбец со всеми данными внутри), с сортировкой по полю “полная информация”   |  | | --- | | “полная информация” | | университет: Иннополис; количество студентов: 1077 | | университет: КФУ; количество студентов: 50000 | | университет: МГУ; количество студентов: 38000 | | университет: МФТИ; количество студентов: 7000 | | университет: Сколково; количество студентов: 1070 | | **select** 'университет: '||**name**||', количество студентов: '||**size** **as** full\_info  **from** college  **order** **by** full\_info |  |
|  | 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** clg.**name**, clg.**size** **from** college **as** clg  **where** clg.id **in** (10, 30, 50)  **order** **by** clg.**size**, clg.**name** |  |
|  | 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** clg.**name**, clg.**size** **from** college **as** clg  **where** clg.id **not** **in** (10, 30, 50)  **order** **by** clg.**size**, clg.**name** |  |
|  | 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** crs.**name**, crs.amount\_of\_students **from** course **as** crs  **where** crs.is\_online  **and** crs.amount\_of\_students **between** 27 **and** 310  **order** **by** crs.**name** **desc**, crs.amount\_of\_students **desc** |  |
|  | Напишите SQL запрос, который возвращает имена студентов и название курсов университетов в одном списке. Результат отсортируйте в убывающем порядке. Пример части результата представлен ниже   |  | | --- | | **name** | | Цифровая трансформация | | Сергей Петров | | … | | **select** **name** **from** student  **union**  **select** **name** **from** course  **order** **by** **name** **desc** |  |
|  | 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   |  |  | | --- | --- | | **name** | **object\_type** | | Иннополис | университет | | КФУ | университет | | ... | … | | Data Mining | курс | | ... | … | | **select** **name**, 'университет' **as** object\_type **from** college  **union**  **select** **name**, 'курс' **as** object\_type **from** course  **order** **by** object\_type **desc**, **name** |  |
|  | 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   |  |  | | --- | --- | | **name** | **amount\_of\_students** | | Введение в РСУБД | 300 | | Data Mining | 10 | | Актерское мастерство | 15 |   Подсказка: используйте в ORDERBY синтаксический элемент CASE...END. | **select** **name**, amount\_of\_students  **from** course  **order** **by** **case**  **when** amount\_of\_students=300 **then** 0  **else** 1  **end**, amount\_of\_students **asc**  **limit** 3 |  |
|  | 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. | **INSERT** **INTO** course (id, **name**, is\_online, amount\_of\_students, college\_id)  **SELECT** 60, 'Machine Learning', **False**, 17, college\_id  **FROM** course **WHERE** **name**='Data Mining'  **RETURNING** \* |  |
|  | 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.   |  | | --- | | **id** | | 70 | | 80 | | 90 | | 100 | | … | | **select** A.id **from** course **as** A  **except** **select** B.id **from** student\_on\_course **as** B  **union**  **select** B.id **from** student\_on\_course **as** B  **except** **select** A.id **from** course **as** A  **order** **by** id |  |
|  | 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   |  |  |  |  | | --- | --- | --- | --- | | **student\_name** | **course\_name** | **student\_college** | **student\_rating** | | Анна Потапова | Нейронные сети | МФТИ | 76 | | Екатерина Андреева | Актерское мастерство | МГУ | 95 | | … | … | … | … | | **select** std.**name** **as** student\_name,  crs.**name** **as** course\_name,  clg.**name** **as** student\_college,  soc.student\_rating **as** student\_rating  **from**  student\_on\_course **as** soc  **left** **join** course **as** crs **on** soc.course\_id=crs.id  **left** **join** student **as** std **on** soc.student\_id=std.id  **left** **join** college **as** clg **on** std.college\_id=clg.id  **where** soc.student\_rating > 50 **and** clg.**size** > 5000  **order** **by** student\_name, course\_name |  |
|  | 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   |  |  |  | | --- | --- | --- | | **student\_1** | **student\_2** | **city** | | Ильяс Мухаметшин | Иван Иванов | Казань | | Сергей Петров | Екатерина Андреева | Москва | | **select** std1.**name** **as** student\_1,  std2.**name** **as** student\_2,  std1.city **as** city  **from** student **as** std1  **inner** **join**  student **as** std2  **on** std1.**name** > std2.**name**  **where** std1.**name** <> std2.**name** **and** std1.city = std2.city  **order** **by** student\_1 |  |
|  | 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   |  |  | | --- | --- | | **оценка** | **Количество студентов** | | неудовлетворительно | 2 | | отлично | 3 | | удовлетворительно | 3 | | хорошо | 5 | | **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'  **else** 'отлично'  **end**  **as** rating\_name, **count**(\*)  **from** student\_on\_course **as** soc  **group** **by** rating\_name  **order** **by** rating\_name |  |
|  | 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.   |  |  |  | | --- | --- | --- | | **курс** | **оценка** | **Количество студентов** | | Data Mining | неудовлетворительно | 1 | | Data Mining | хорошо | 2 | | Актерское мастерство | отлично | 2 | | … | … | … | | Цифровая трансформация | удовлетворительно | 2 | | **select** crs.**name** **as** course\_name,  **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'  **else** 'отлично'  **end**  **as** rating\_name,  **count**(\*)  **from** course **as** crs  **right** **join** student\_on\_course **as** soc  **on** crs.id = soc.course\_id  **group** **by** course\_name, rating\_name  **order** **by** course\_name, rating\_name |  |