Задача

**Задание 1**

1. Напишите и выполните SQL-запрос, получающий объединённые данные двух таблиц, **product\_customers** и **product\_suppliers**. Используйте оператор объединения без повтора значений — **UNION**.

2. Напишите и выполните SQL-запрос, получающий объединённые данные двух таблиц, **skill\_product** и **skill\_product\_v2**. Используйте оператор объединения без повтора значений — **UNION**.

3. Напишите и выполните SQL-запрос, получающий объединённые данные двух таблиц, **skill\_product** и **skill\_product\_v2**. Используйте оператор объединения **UNION ALL**, который не удаляет дубликаты в результирующей выборке.

4. Получите одним запросом список товаров без повторений (колонки **product\_name**) из таблиц **skill\_product** и **skill\_product\_v2**.

5. Получите одним запросом список городов без повторений из таблиц **skill\_customers** и **skill\_suppliers**.

**Задание 2**

1. Напишите и выполните SQL-запрос, получающий пересекающиеся данные двух таблиц, **product\_customers** и **product\_suppliers**, Используйте оператор пересечения — **INTERSECT**.

2. Напишите и выполните SQL-запрос, получающий строки, которые есть и в таблице **skill\_product**, и в таблице **skill\_product\_v2**.

3. Напишите и выполните SQL-запрос, получающий строки, которые есть в таблице **product\_customers**, но отсутствуют в таблице **product\_suppliers**. Используйте оператор исключения — **EXCEPT**.

4. Напишите и выполните SQL-запрос, получающий строки, которые есть в таблице **product\_suppliers**, но отсутствуют в таблице **product\_customers**.

5. Таблицы **skill\_product** и **skill\_product\_v2** содержат данные о товарах. Выведите одним запросом список товаров (колонки **product\_name**), которые есть в обеих таблицах.

**Задание 3**

1. Напишите запрос, соединяющий все строки таблицы **city\_suppliers** (со списком городов)  со всеми строками таблицы **city\_property** (с параметрами оценки города). В запросе используйте оператор **CROSS JOIN**.

2. Выполните задание из предыдущего пункта без явного использования оператора **CROSS JOIN**.

3. Сопоставьте строки из таблицы **gender** с женским полом (**gender='F'**) со всеми периодами возрастов таблицы **age\_range**, исключив период до 18 лет (**age\_range!='[-18]'**).

4. В таблице **week\_days** перечислены дни недели и признак выходного дня для каждой строки. В таблице **lessons** содержится расписание уроков на один день: номер урока, название урока, время начала и время окончания урока.

Запросом составьте сетку расписания на каждый будний день следующего вида:

|  |  |  |
| --- | --- | --- |
| **week\_day** | **name** | **lesson\_time** |
| День недели (поле **week\_day** из таблицы **week\_days**) | Название урока (поле **name**из таблицы **lessons**) | Время начала урока и через тире — время окончания урока (поля **lesson\_begin** и **lesson\_end** таблицы **lessons**). |

5. Из таблиц **skill\_suppliers** и **skill\_customers** выведите список всех пар имён продавцов (**supplier**) и покупателей (**customer**), которые продают или покупают процессоры (**product='CPU'**).

**Задание 4**

1. Таблица **skill\_customers** содержит информацию об имени покупателя, городе проживания и наименовании товара. Таблица **skill\_suppliers** содержит информацию об имени продавца, городе проживания, наименовании и количестве товара. Объедините данные двух таблиц SQL-запросом, сформировав результирующую таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **city** | **product** | **cnt** |
| Имя продавца или покупателя | Город проживания | Наименование продукта | Количество товара |

В результирующей таблице необходимо:

* назвать колонки так же, как в примере;
* везде указать количество товара для покупателя — 1.

2. Колонка customer таблицы **skill\_customers** содержит имена покупателей, а колонка supplier таблицы **skill\_suppliers** — имена продавцов. Сделайте вывод, : есть ли одинаковые имена в списках продавцов и покупателей, не используя оператор пересечения **INTERSECT**. Решите задачу и объясните логику решения.

3. В таблице продавцов **skill\_suppliers** в колонке **supplier** указано полное имя продавца, состоящее из имени — первое слово в значении колонки **supplier** и фамилии — второе слово колонки **supplier**.

Одним запросом составьте все возможные полные имена, состоящие из имён и фамилий продавцов из таблицы **skill\_suppliers**.

**Критерии оценки**

**«Зачёт»** — в тестовой среде выполнены SQL-запросы, выведены данные, которые содержатся в таблицах, и даны необходимые пояснения. Зачёт ставится при успешном выполнении всех пунктов заданий 1–4.

**«На доработку»** — выполнены не все SQL-запросы, не получены данные или тестовая среда выдает ошибку, связанную с синтаксисом SQL-запроса, нет необходимых пояснений.

**Как отправить задание на проверку**

Отправьте тексты SQL-запросов и скриншот с результатами их выполнения в тестовой среде через форму в личном кабинете. Если в задании требуются пояснения, добавьте их к ответу.