Задача

**Цель домашнего задания**

Закрепить навыки использования оконных функций.

**Общие рекомендации по выполнению**

* По каждому пункту задания напишите SQL-запрос(ы) и выполните их в [тестовой среде](https://dbfiddle.uk/?rdbms=postgres_12&fiddle=ee0a37768db8288172801e88b0aa0cfa&hide=1).
* Допускается выполнять несколько запросов в разных полях одной сессии тестовой среды.
* Ответ должен содержать тексты SQL-запросов и скриншот с результатами их выполнения в тестовой среде или ссылку на среду с выполненными запросами.
* Все задания обязательны для выполнения.

**Задание 1**

В тестовой среде находится таблица **skill\_market** (список товаров на складах торговой компании) со следующей структурой:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | country | string | Страна, в которой расположен склад с товаром |
| 2 | city | string | Город, в котором расположен склад с товаром |
| 3 | office\_name | string | Наименование офиса — склада с товаром |
| 4 | office\_type | string | Тип офиса |
| 5 | category | string | Категория товара |
| 6 | type | string | Потребительский сегмент товара |
| 7 | name | string | Наименование товара |
| 8 | article | string | Наименование товара с артикулом |
| 9 | price | float | Цена товара в USD |
| 10 | cnt | integer | Количество товара на складе |

Напишите SQL-запросы к таблице **skill\_market**, получающие следующие данные:

1. Все колонки таблицы **skill\_market** и одну вычисляемую колонку, содержащую номер строки (для этой колонки используйте псевдоним **num**). Для вычисления значения используйте оконную функцию **row\_number**.

2. Доработайте SQL-запрос из пункта 1 так, чтобы нумерация строк в колонке **num** осуществлялась согласно возрастанию значений в колонке **cnt**. Также добавьте общую сортировку результирующей таблицы по возрастанию значений колонки **cnt**.

3. Все колонки таблицы **skill\_market** и две вычисляемые колонки:

* **num1:** нумерация строк в порядке возрастания значения колонки **cnt**;
* **num2:** нумерация строк в порядке сортировки по значениям следующих колонок: **country, city, office\_name, price**.

    Общая сортировка результирующей выборки должна соответствовать значениям, рассчитанным для колонки num2.

4. Все колонки таблицы **skill\_market** и следующие вычисляемые колонки:

* **num1:** нумерация строк (функция **row\_number**);
* **total\_rows:** общее количество строк в результирующей выборке (функция **count**);
* **total\_cnt:** общее количество товара, хранящегося на всех складах (функция **sum**);
* **total:** общая стоимость товара, хранящегося на всех складах (функция **sum**).

**Задание 2**

Напишите SQL-запросы к таблице **skill\_market** (структура таблицы описана в задании 1), получающие следующие данные:

1. Все колонки таблицы **skill\_market** и колонка **num** с номером строки. Нумерация строк должна быть отдельной для каждого города и начинаться с единицы.

2. Все колонки таблицы **skill\_market** и колонка **num** с номером строки. Нумерация строк должна быть отдельной для каждой группы значений колонок «Тип офиса» и «Наименование товара» и начинаться с единицы.

3. Все колонки таблицы **skill\_market** и две вычисляемые колонки:

* **num1:** нумерация строк в порядке возрастания значения колонки **cnt**;
* **num2:** нумерация строк в порядке убывания значений **cnt**.

    Все нумерации строк должны быть отдельными для каждого офиса.

4. Все колонки таблицы **skill\_market** и следующие вычисляемые колонки:

* **num1:** нумерация строк;
* **total\_rows:** общее количество строк в результирующей выборке;
* **total\_cnt:** общее количество товара, хранящегося на всех складах;
* **total:** общая стоимость товара, хранящегося на всех складах.

    Все вычисления должны производиться в группах, разделённых по городам.

**Задание 3**

Напишите SQL-запросы к таблице **skill\_market** (структура таблицы описана в задании 1). Каждый запрос должен вернуть все колонки таблицы и одну вычисляемую: общая сумма стоимости товара **sum (cnt\*price)** по окнам данных, которые определяются партициями по столбцу «Наименование товара» и отсортированы по столбцу «Наименование товара с артикулом».

Дополнительно границы окна должны быть определены следующими настройками:

1. Строки от начала окна данных до текущей строки.
2. Строки от текущей строки до конца окна данных.
3. Текущая строка и одна следующая.

**Задание 4**

В тестовой среде находится таблица **skill\_bids** (данные электронных торгов) со следующей структурой:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | bid\_time | time | Время открытия лота / подачи заявки |
| 2 | seller\_uid | string | Идентификатор продавца |
| 3 | customer\_uid | string | Идентификатор покупателя |
| 4 | lot\_uid | string | Идентификатор лота |
| 5 | event | string | Событие: bid — открытие лота продавцом, request — заявка на покупку покупателем |
| 6 | rate | int | Начальная стоимость лота / стоимость заявки на выкуп лота |

Таблица фиксирует следующие события:

1. Открытие лота (**lot\_uid**) продавцом (**seller\_uid**) в момент времени (**bid\_time**) с начальной ценой (**rate**) — при данном событии поле event содержит значение**bid**.
2. Заявка на покупку лота (**lot\_uid**) покупателем (**customer\_uid**) в момент времени (**bid\_time**) по цене (**rate**) — в поле event заносится значение request.

Напишите SQL-запросы к таблице **skill\_bids**. Результирующие данные должны быть отсортированы по лотам и по времени регистрации события. Каждый запрос должен вернуть все колонки таблицы **skill\_bids** и одну вычисляемую колонку, содержащую следующие данные:

1. **seller\_uid** продавца, который инициализировал текущий лот.
2. Максимальное значение ставки лота, независимо от типа запроса.
3. Время, прошедшее с момента открытия лота.

**Задание 5**

Используя данные таблицы **skill\_bids** (структура таблицы описана в задании 4), выявите победителей аукционов по следующим правилам:

1. Заявка должна быть подана в течение 30 секунд (включительно) после открытия лота. Остальные заявки не учитываются.
2. Заявка должна быть выкуплена по максимальной цене, предложенной в течение разрешённого временного периода.
3. Если победитель не выявлен, то информация о лоте в результирующей выборке отсутствует.
4. Результирующая выборка должна содержать минимальный набор колонок:

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | bid\_time | Время подачи заявки, которая победила на аукционе |
| 2 | seller\_uid | Идентификатор продавца, подавшего заявку |
| 3 | customer\_uid | Идентификатор покупателя, выигравшего аукцион |
| 4 | lot\_uid | Идентификатор лота |
| 5 | rate | Стоимость выкупа лота |

Во всех столбцах должны присутствовать данные.

Результирующая таблица должна быть получена одним запросом (можно использовать вложенные SQL-запросы).