

# SQL для анализа данных

**Илья Мещеряков**

Аналитик данных в «МаксимаТелеком» (Wi-Fi в метро)

# Приветствие и разбор домашнего задания

**Илья Мещеряков**

Аналитик данных в «МаксимаТелеком» (Wi-Fi в метро)

# Задание 1

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 п. 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 (воспользуйтесь оконной функцией `row_number`)
  - **total\_rows** — общее количество строк в результирующей выборке (воспользуйтесь оконной функцией `count`)
  - **total\_cnt** — общее количество товара, хранящегося на всех складах (воспользуйтесь оконной функцией `sum`)
  - **total** — общая стоимость товара, хранящегося на всех складах (воспользуйтесь оконной функцией `sum`)

## Задание 2

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

1. Все колонки таблицы **skill\_market** и одну вычисляемую колонку **(num)**, содержащую номер строки. Нумерация строк должна быть отдельной для каждого города (колонка **city**) и начинаться с единицы
2. Все колонки таблицы **skill\_market** и одну вычисляемую колонку **(num)**, содержащую номер строки. Нумерация строк должна быть отдельной для каждой группы значений колонок — тип офиса и наименование товара **(office\_type,name)** 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 (колонка **office\_name**).

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 (колонка **city**).

# Задание 3

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

**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 находится таблица **skill\_bids** (данные электронных торгов).

Таблица фиксирует следующие события: открытие лота (**lot\_uid**) продавцом (**seller\_uid**) в момент времени (**bid\_time**) с начальной ценой (**rate**) — при данном событии поле **event** содержит значение **bid**, 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 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

# Оконные функции

bid_time	lot_uid	seller_uid	customer_uid	rate	event
09:45:01	lot_1	seller_25		1284	bid
09:45:11	lot_1		customer_124	1290	request
09:47:17	lot_115	seller_25		3150	bid
09:47:21	lot_115		customer_124	3155	request
09:47:22	lot_115		customer_213	3160	request
09:47:23	lot_115		customer_124	3300	request
09:47:24	lot_115		customer_213	3162	request
09:50:30	lot_115		customer_124	3200	request
09:55:30	lot_115		customer_213	3210	request
09:56:35	lot_115		customer_120	3250	request
10:11:00	lot_271	seller_987		114	bid
10:11:05	lot_271		customer_651	115	request
10:11:10	lot_271		customer_852	116	request
10:11:20	lot_271		customer_183	118	request
10:11:30	lot_271		customer_651	119	request
10:11:40	lot_271		customer_852	122	request
10:12:10	lot_271		customer_183	135	request
10:13:00	lot_271		customer_651	130	request
10:27:30	lot_3478	seller_3451		500	bid
10:27:31	lot_3478		customer_193	499	request
10:27:32	lot_3478		customer_9887	401	request
10:27:33	lot_3478		customer_12567	400	request
10:27:34	lot_3478		customer_9887	400	request
10:27:35	lot_3478		customer_193	399	request
10:27:37	lot_3478		customer_1298	500	request
10:28:40	lot_3478		customer_8965	500	request

```
select
bid_time,lot_uid,seller_uid,customer_uid,rate,event
from skill_bids
order by lot_uid,bid_time
```

Начало лота  
(event='bid')

# Оконные функции

bid_time	lot_uid	seller_uid	customer_uid	rate	event
09:45:01	lot_1	seller_25		1284	bid
09:45:11	lot_1		customer_124	1290	request
09:47:17	lot_115	seller_25		3150	bid
09:47:21	lot_115		customer_124	3155	request
09:47:22	lot_115		customer_213	3160	request
09:47:23	lot_115		customer_124	3300	request
09:47:24	lot_115		customer_213	3162	request
09:50:30	lot_115		customer_124	3200	request
09:55:30	lot_115		customer_213	3210	request
09:56:35	lot_115		customer_120	3250	request
10:11:00	lot_271	seller_987		114	bid
10:11:05	lot_271		customer_651	115	request
10:11:10	lot_271		customer_852	116	request
10:11:20	lot_271		customer_183	118	request
10:11:30	lot_271		customer_651	119	request
10:11:40	lot_271		customer_852	122	request
10:12:10	lot_271		customer_183	135	request
10:13:00	lot_271		customer_651	130	request
10:27:30	lot_3478	seller_3451		500	bid
10:27:31	lot_3478		customer_193	499	request
10:27:32	lot_3478		customer_9887	401	request
10:27:33	lot_3478		customer_12567	400	request
10:27:34	lot_3478		customer_9887	400	request
10:27:35	lot_3478		customer_193	399	request
10:27:37	lot_3478		customer_1298	500	request
10:28:40	lot_3478		customer_8965	500	request

```
select
bid_time,lot_uid,seller_uid,customer_uid,rate,event
from skill_bids
order by lot_uid,bid_time
```

Начало лота  
(event='bid')

Предложения продавцов  
(event='request')



# 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. Стоимость заявки 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: bid\_time, seller\_uid, customer\_uid, lot\_uid, rate

Во всех столбцах должны присутствовать данные.

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

**Спасибо  
за внимание!**