АНТОН ЖИЯНОВ

ОКОННЫЕ ФУНКЦИИ SQL

Издательство АСТ
Москва

УДК 004.43
ББК 32.973
Ж66

Жиянов, Антон.
Ж66 Оконные функции SQL. Анализ данных на практике / Антон Жиянов. — Москва: Издательство АСТ, 2024. — 254 с. — (Программирование для всех)

ISBN 978–5–17–158845–8

«Оконные функции SQL» — книга о мощном инструменте для анализа данных, который позволяет выполнять сложные вычисления и получать информацию о группах строк или результатах окон, но если вкратце — как делать классные аналитические отчеты без участия «экселя».

Вместе с Антоном Жияновым разберемся в основах SQL:

- что такое «окно» в SQL и как оно работает;
- про фреймы и как с ними работать;
- и конечно же выполним практические задания в песочнице.

Книга будет полезна как начинающим разработчикам, так и опытным специалистам, желающим расширить и закрепить свои знания в области оконных функций SQL.

УДК 004.43
ББК 32.973

ISBN 978–5–17–158845–8

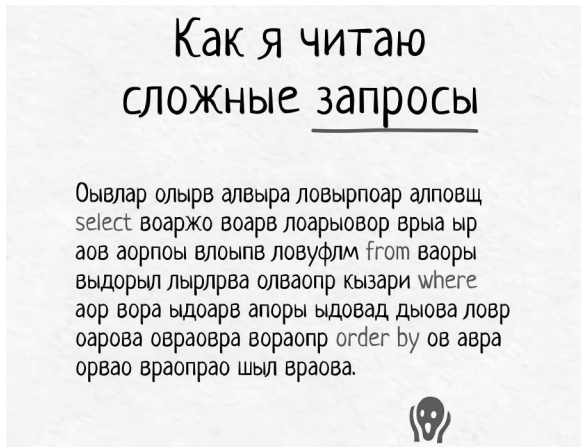
Введение

О книге

Кто–то однажды сказал, что если освоить оконные функции в SQL — жизнь уже никогда не будет прежней. В хорошем смысле. Задача книги помочь вам освоить «окна» без напряжения всех сил и головной боли от мудреных объяснений.

Помогут в этом три инструмента:
1. Картинки.
2. Картинки.
3. Картинки.

Шучу. Но картинок правда будет много, как и практических задачек. Не пропускайте их, чтобы надежно усвоить материал. Это важно. Если не разобраться в запросах с «окошками», выглядят они как–то так:



Книга состоит из трех частей:

1. В первой части мы научимся использовать оконные функции.
2. Во второй части погрузимся в нюансы оконных фреймов.
3. В третьей части дополнительно попрактикуемся.

Если полностью изучите первую часть — уже освоите оконные функции на более глубоком уровне, чем большинство ваших коллег и знакомых. И сможете спокойно применять в рабочих задачах. Вторая и третья части — для тех, кого «окошки» увлекут по-настоящему.

Базы данных

Оконные функции в той или иной степени поддерживаются во всех современных реляционных СУБД. Книга тестировалась на трех:

- MySQL 8.0.2+ (MariaDB 10.2+)
- PostgreSQL 11+
- SQLite 3.28+

Полнее всего окошки реализованы в PostgreSQL и SQLite. MySQL поддерживает основные возможности, но лишен некоторых более продвинутых. Oracle 11g+, MS SQL 2012+ и Google BigQuery поддерживают «окошки» примерно так же, как MySQL. Так что если вы используете одну из них — книгу тоже будет полезна.

Вы можете использовать любую из перечисленных СУБД, если она у вас под рукой. Если нет — в следующей главе я дам ссылку на онлайн-песочницу.

Вопросы и задания

Я стараюсь объяснять материал просто и наглядно. Но окна в SQL — сложная тема, поэтому важный момент: *задания — неотъемлемая часть книги.* Половину знаний вы получите именно из заданий. Поэтому старайтесь не пропускать их. Не бойтесь ошибаться.

Версия книги

Версия книги: 2023.05.29

Авторские права

Зачем нужны оконные функции

Если вкратце — оконные функции помогают делать классные аналитические отчеты без участия «экселя».

Проще всего объяснять на конкретных примерах. Будем работать с игрушечной таблицей сотрудников, вот такой:

id	name	city	department	salary
11	Дарья	Самара	hr	70
12	Борис	Самара	hr	78
21	Елена	Самара	it	84
22	Ксения	Москва	it	90
23	Леонид	Самара	it	104
24	Марина	Москва	it	104
25	Иван	Москва	it	120
31	Вероника	Москва	sales	96
32	Григорий	Самара	sales	96
33	Анна	Москва	sales	100

Рассмотрим некоторые задачи, которые удобно решать с помощью «окошек» в SQL. Как именно их решать — разберемся в следующей главе. Пока просто оцен им возможности.

Если вам не терпится перейти к практике — эту главу можно пропустить.

Ранжирование

Ранжирование — это всевозможные рейтинги, начиная от призеров чемпионата мира по плаванию и заканчивая Forbes 500.

Мы будем ранжировать сотрудников.

Общий рейтинг зарплат

Составим рейтинг сотрудников по размеру заработной платы:

Столбец `rank` показывает позицию сотрудника в рейтинге.

Видно, что у некоторых коллег одинаковая зарплата (Леонид и Марина, Вероника и Григорий) — поэтому они получили один и тот же ранг.

rank	name	department	salary ↓
1	Иван	it	120
2	Леонид	it	104
2	Марина	it	104
3	Анна	sales	100
4	Вероника	sales	96
4	Григорий	sales	96
5	Ксения	it	90
6	Елена	it	84
7	Борис	hr	78
8	Дарья	hr	70

Рейтинг зарплат по департаментам

Тот же рейтинг, только не для всей компании, а по каждому департаменту в отдельности:

rank	name	department	salary ↓
1	Борис	hr	78
2	Дарья	hr	70
1	Иван	it	120
2	Леонид	it	104
2	Марина	it	104
3	Ксения	it	90
4	Елена	it	84
1	Анна	sales	100
2	Вероника	sales	96
2	Григорий	sales	96

Столбец `rank` показывает позицию сотрудника в рейтинге конкретного департамента.

Группы по зарплате

Разобьем сотрудников на три группы в зависимости от размера зарплаты:

- высокооплачиваемые,
- средние,
- низкооплачиваемые.

tile	name	department	salary ↓
1	Иван	it	120
1	Леонид	it	104
1	Марина	it	104
1	Анна	sales	100
2	Вероника	sales	96
2	Григорий	sales	96
2	Ксения	it	90
3	Елена	it	84
3	Борис	hr	78
3	Дарья	hr	70

Столбец `tile` показывает, к какой группе относится каждый сотрудник.

Самые «дорогие» коллеги

Найдем самых высокооплачиваемых людей по каждому департаменту:

id	name	department	salary
12	Bob	hr	78
25	Frank	it	120
33	Alice	sales	100

Что ж, этим зарплату больше не повышать.

Сравнение со смещением

Сравнение со смещением — это когда мы смотрим, в чем разница между соседними значениями. Например, сравниваем страны, которые занимают 5 и 6 место в мировом рейтинге ВВП — сильно ли отличаются? А если сравнить 1 и 6 место?

Сюда же попадают задачи, в которых мы сравниваем значение из набора с границами набора. Например, есть 100 лучших теннисисток мира. Мария Саккари занимает в рейтинге 10 место. Как ее показатели соотносятся со спортсменкой, которая занимает первое место? А с той, кто занимает последнее?

Мы будем сравнивать сотрудников.

Разница по зарплате с предыдущим

Упорядочим сотрудников по возрастанию зарплаты и проверим, велик ли разрыв между соседями:

name	department	salary ↑	diff
Дарья	hr	70	
Борис	hr	78	11%
Елена	it	84	8%
Ксения	it	90	7%
Вероника	sales	96	7%
Григорий	sales	96	0%
Анна	sales	100	4%
Леонид	it	104	4%
Марина	it	104	0%
Иван	it	120	15%

Столбец `diff` показывает, на сколько процентов зарплата сотрудника отличается от предыдущего коллеги. Видно, что больших разрывов нет. Самые крупные — между Дарьей и Борисом (11%) и Мариной и Иваном (15%).

Диапазон зарплат в департаменте

Посмотрим, как зарплата сотрудника соотносится с минимальной и максимальной зарплатой в его департаменте:

name	depart	salary ↑	low	high
Дарья	hr	70	70	78
Борис	hr	78	70	78
Елена	it	84	84	120
Ксения	it	90	84	120
Леонид	it	104	84	120
Марина	it	104	84	120
Иван	it	120	84	120
Вероника	sales	96	96	100
Григорий	sales	96	96	100
Анна	sales	100	96	100

Для каждого сотрудника столбец `low` показывает минимальную зарплату родного департамента, а столбец `high` — максимальную. Видно, что разброс значений в HR и продажах невелик, а у айтишников — значительный.

Агрегация

Агрегация — это когда мы считаем суммарные или средние показатели. Например, среднюю зарплату по каждому региону или количество золотых медалей у каждой страны в зачете Олимпийских игр.

Мы будем агрегировать зарплаты сотрудников.

Сравнение с фондом оплаты труда

У каждого департамента есть фонд оплаты труда — денежная сумма, которая ежемесячно уходит на выплату зарплат сотрудникам. Посмотрим, какой процент от этого фонда составляет зарплата каждого сотрудника:

name	depart	salary ↑	fund	perc
Дарья	hr	70	148	47%
Борис	hr	78	148	53%
Елена	it	84	502	17%
Ксения	it	90	502	18%
Леонид	it	104	502	21%
Марина	it	104	502	21%
Иван	it	120	502	24%
Вероника	sales	96	292	33%
Григорий	sales	96	292	33%
Анна	sales	100	292	34%

Столбец `fund` показывает фонд оплаты труда отдела, а `perc` — долю зарплаты сотрудника от этого фонда. Видно, что в HR и продажах все более–менее ровно, а у айтишников есть заметный разброс зарплат.

Сравнение со средней зарплатой

Интересно, велик ли разброс зарплат в департаментах. Проверим — посчитаем отклонение зарплаты каждого сотрудника от средней по департаменту:

name	depart	salary ↑	savg	diff
Дарья	hr	70	74	-5%
Борис	hr	78	74	5%
Елена	it	84	100	-16%
Ксения	it	90	100	-10%
Леонид	it	104	100	4%
Марина	it	104	100	4%
Иван	it	120	100	20%
Вероника	sales	96	97	-1%
Григорий	sales	96	97	-1%
Анна	sales	100	97	3%

Результат подтверждает предыдущие наблюдения: у айтишников зарплаты колеблются от –16% до +20% от среднего, а у остальных департаментов отклонение в пределах 5%.

Скользящие агрегаты

Скользящие агрегаты — это те же сумма и среднее. Только рассчитывают их не по всем элементам набора, а более хитрым способом.

Поясню на примере. Здесь возьмем другую таблицу — с доходами и расходами компании за 9 месяцев 2020 года:

year	month	income	expense
2020	1	94	82
2020	2	94	75
2020	3	94	104
2020	4	100	94
2020	5	100	99
2020	6	100	105
2020	7	100	95
2020	8	100	110
2020	9	104	104

Скользящее среднее по расходам

Судя по данным, доходы растут: 94 в январе → 104 в сентябре. А вот растут ли расходы? Сходу сложно сказать, месяц на месяц не приходится. Чтобы сгладить эти скачки, используют «скользящее среднее» — для каждого месяца рассчитывают средний расход с учетом предыдущего и следующего месяца. Например:

- скользящее среднее за февраль = (январь + февраль + март) / 3;

- за март = (февраль + март + апрель) / 3;
- за апрель = (март + апрель + май) / 3;
- и так далее.

Рассчитаем скользящее среднее по всем месяцам:

year ↑	month ↑	expense	roll_avg
2020	1	82	79
2020	2	75	87
2020	3	104	91
2020	4	94	99
2020	5	99	99
2020	6	105	100
2020	7	95	103
2020	8	110	103
2020	9	104	107

Теперь хорошо видно, что расходы стабильно растут.

Прибыль нарастающим итогом

Благодаря скользящему среднему, мы выяснили, что растут и доходы, и расходы. А как они соотносятся друг с другом? Хочется понять, находится ли компания «в плюсе» или «в минусе» с учетом всех заработанных и потраченных денег.

Причем важно понимать не на конец года, а на каждый месяц. Потому что если по итогам года все ОК, а в июне компания ушла в минус — это потенциальная проблема (такую ситуацию называют «кассовым разрывом»).

Поэтому посчитаем доходы и расходы по месяцам нарастающим итогом (кумулятивно):

- кумулятивный доход за январь = январь;
- за февраль = январь + февраль;

- за март = январь + февраль + март;
- за апрель = январь + февраль + март + апрель;
- и так далее.

month ↑	income	expense	t_income	t_expense	t_profit
1	94	82	94	82	12
2	94	75	188	157	31
3	94	104	282	261	21
4	100	94	382	355	27
5	100	99	482	454	28
6	100	105	582	559	23
7	100	95	682	654	28
8	100	110	782	764	18
9	104	104	886	868	18

Теперь видно, что дела у компании идут неплохо. В некоторых месяцах расходы превышают доходы, но благодаря накопленной «денежной подушке» кассового разрыва не происходит.

Резюме

Вот задачи, которые непринужденно решаются с помощью оконных функций в SQL:

- Ранжирование (всевозможные рейтинги).
- Сравнение со смещением (соседние элементы и границы).
- Агрегация (сумма и среднее).
- Скользящие агрегаты (сумма и среднее в динамике).

Конечно, это не исчерпывающий список. Но, надеюсь, теперь понятно, как пригодятся оконные функции в аналитике данных. В следующей главе разберемся, что такое «окна» и как их применять.

Песочница

Если будете работать в локальной базе — вот скрипт, который создает таблицы и данные:

```
https://antonz.ru/to#empsql
```

Если базы под рукой нет — используйте песочницу:

```
https://antonz.ru/to#empdb
```

Это SQLite, который работает прямо в браузере. Можете прямо сейчас попробовать выполнить в ней запрос:

```
select * from employees;
```

id	name	city	department	salary
11	Дарья	Самара	hr	70
12	Борис	Самара	hr	78
21	Елена	Самара	it	84
22	Ксения	Москва	it	90
23	Леонид	Самара	it	104
24	Марина	Москва	it	104
25	Иван	Москва	it	120
31	Вероника	Москва	sales	96
32	Григорий	Самара	sales	96
33	Анна	Москва	sales	100

В книге много упражнений, где вы будете писать запросы — это можно делать в песочнице. После того как успешно решите задачу, сравните с эталонным решением.