Дорогие друзья!

В этом уроке вы узнаете про основные функции по работе с датой и временем в **DataLense**. Познакомитесь с их аргументами и параметрами. На конкретном примере увидите, как они работают. А в конце мы разберем по шагам три практические задачи, где наглядно увидим прикладной смысл и применение некоторых функций.

Вы научитесь делать

Рассчитывать разницу в днях между двумя датами.

Date	TODAY	Разница
01.12.2022	20.06.2023	201
10.12.2022	20.06.2023	192
19.12.2022	20.06.2023	183
25.12.2022	20.06.2023	177
10.01.2023	20.06.2023	161
11.01.2023	20.06.2023	160
13.01.2023	20.06.2023	158
15.01.2023	20.06.2023	156

Добавлять/вычитать из даты дни, недели и года.

Date	+2 недели
01.12.2022	15.12.2022
10.12.2022	24.12.2022
19.12.2022	02.01.2023
25.12.2022	08.01.2023
10.01.2023	24.01.2023

Определять порядковый номер дня в году.

Date	Начало года	Номер дня года
01.12.2022	01.01.2022	335
10.12.2022	01.01.2022	344
19.12.2022	01.01.2022	353
25.12.2022	01.01.2022	359
10.01.2023	01.01.2023	10
11.01.2023	01.01.2023	11
13.01.2023	01.01.2023	13
15.01.2023	01.01.2023	15

И многое другое.

ОСНОВНЫЕ ФУНКЦИИ

Для начала давайте разберем все имеющиеся функции в **DataLens**, которые работают с датой и временем.

DATEPART

Данная функция возвращает часть даты в виде целого числа. Например, номер дня в месяце или номер дня недели. Одним словом, порядковый номер какого-то измерителя даты.

Синтаксис: DATEPART(datetime, unit [, firstday])

datetime – исходная дата, из которой нужно получить результат.

unit – аргумент, указывающий, что именно нужно получить (номер дня, номер недели и т.д.)

Возможные значения **unit**:

- "*year*" — номер года (см. YEAR);
- "*quarter*" — номер квартала года (от 1 до 4) (см. QUARTER);
- "*month*" — номер месяца в году (см. MONTH);

- "*week*" — номер недели в году по ISO 8601 (см. WEEK);
- "*dayofweek*", "*dow*" — номер дня недели (см. DAYOFWEEK);
- "*day*" — номер дня в месяце (см. DAY);
- "*hour*" — номер часа в дне (см. HOUR);
- "*minute*" — номер минуты в часе (см. MINUTE);
- "*second*" — номер секунды в минуте (см. SECOND).

[, firstday] - если выбран "*dayofweek*", то дополнительным параметром *firstday* можно задать, какой день недели считать первым — по умолчанию это понедельник.

Теперь давайте на практике посмотрим, как это работает. Добавим в нашу таблицу новые поля с разными аргументами **unit**.

Date	Year	quarter	DayOfWeek	month
01.12.2022	2022	4	4	12
10.12.2022	2022	4	6	12
19.12.2022	2022	4	1	12
25.12.2022	2022	4	7	12
10.01.2023	2023	1	2	1
11.01.2023	2023	1	3	1
13.01.2023	2023	1	5	1
15.01.2023	2023	1	7	1
20.01.2023	2023	1	5	1
21.01.2023	2023	1	6	1
22.01.2023	2023	1	7	1
23.01.2023	2023	1	1	1
24.01.2023	2023	1	2	1
30.01.2023	2023	1	1	1
05.02.2023	2023	1	7	2
06.02.2023	2023	1	1	2
07.02.2023	2023	1	2	2
10.02.2023	2023	1	5	2
19.02.2023	2023	1	7	2

А вот так выглядит формула одного из столбцов.

DATEADD

Данная функция позволяет выполнять с датой необходимые вычисления. Например, рассчитать новую дату, прибавив к ней 5 дней или 5 месяцев.

Синтаксис: DATEADD(datetime [, unit [, number]])

datetime – исходная дата, из которой нужно получить результат.

unit – аргумент, указывающий, с какой именно частью даты необходимо произвести вычисление (месяц, день и т.д.)

Аргумент **unit** принимает следующие значения:

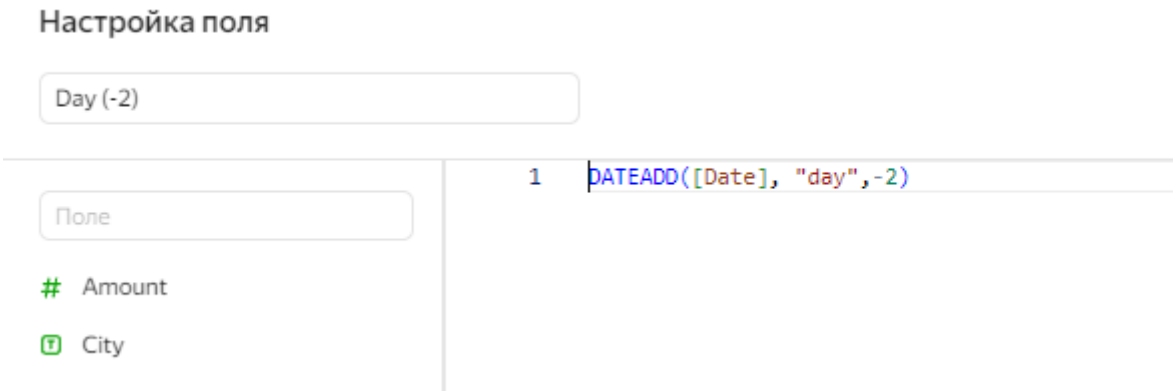
- "*year*" — год;
- "*month*" — месяц;
- "*day*" — день;
- "*hour*" — час;
- "*minute*" — минута;
- "*second*" — секунда.

number – аргумент, указывающих число дней, месяцев или лет, на которое надо вычислить дату. Задается целым числом. Может принимать отрицательные значения.

Теперь давайте на практике посмотрим, как это работает. Добавим в нашу таблицу новые поля с разными аргументами **unit**. Попробуем рассчитать новые даты. Названия столбцов указывают на аргументы **unit** в формуле, а цифры в скобках – это количество, на которое дата должна измениться.

Date	Year (+2)	Quarter (+2)	Day (-2)	Month (+2)
01.12.2022	01.12.2024	01.06.2023	29.11.2022	01.02.2023
10.12.2022	10.12.2024	10.06.2023	08.12.2022	10.02.2023
19.12.2022	19.12.2024	19.06.2023	17.12.2022	19.02.2023
25.12.2022	25.12.2024	25.06.2023	23.12.2022	25.02.2023
10.01.2023	10.01.2025	10.07.2023	08.01.2023	10.03.2023
11.01.2023	11.01.2025	11.07.2023	09.01.2023	11.03.2023
13.01.2023	13.01.2025	13.07.2023	11.01.2023	13.03.2023
15.01.2023	15.01.2025	15.07.2023	13.01.2023	15.03.2023
20.01.2023	20.01.2025	20.07.2023	18.01.2023	20.03.2023
21.01.2023	21.01.2025	21.07.2023	19.01.2023	21.03.2023
22.01.2023	22.01.2025	22.07.2023	20.01.2023	22.03.2023
23.01.2023	23.01.2025	23.07.2023	21.01.2023	23.03.2023
24.01.2023	24.01.2025	24.07.2023	22.01.2023	24.03.2023
30.01.2023	30.01.2025	30.07.2023	28.01.2023	30.03.2023
05.02.2023	05.02.2025	05.08.2023	03.02.2023	05.04.2023
06.02.2023	06.02.2025	06.08.2023	04.02.2023	06.04.2023
07.02.2023	07.02.2025	07.08.2023	05.02.2023	07.04.2023
10.02.2023	10.02.2025	10.08.2023	08.02.2023	10.04.2023
19.02.2023	19.02.2025	19.08.2023	17.02.2023	19.04.2023

Ниже можно посмотреть на пример формулы одной из колонок.



DATETRUNC

Данная функция позволяет округлять дату до нужной размерности. Например, до месяц, квартала или года. Ее удобно использовать, когда нужно агрегировать данные, увидеть показатели только за годы, без разбивки по дням или месяцам.

Синтаксис: DATETRUNC(datetime, unit [, number])

datetime – исходная дата, из которой нужно получить результат.

unit – аргумент, указывающий на размерность даты (месяц, день и т.д.)

Поддерживаемые значения **unit**:

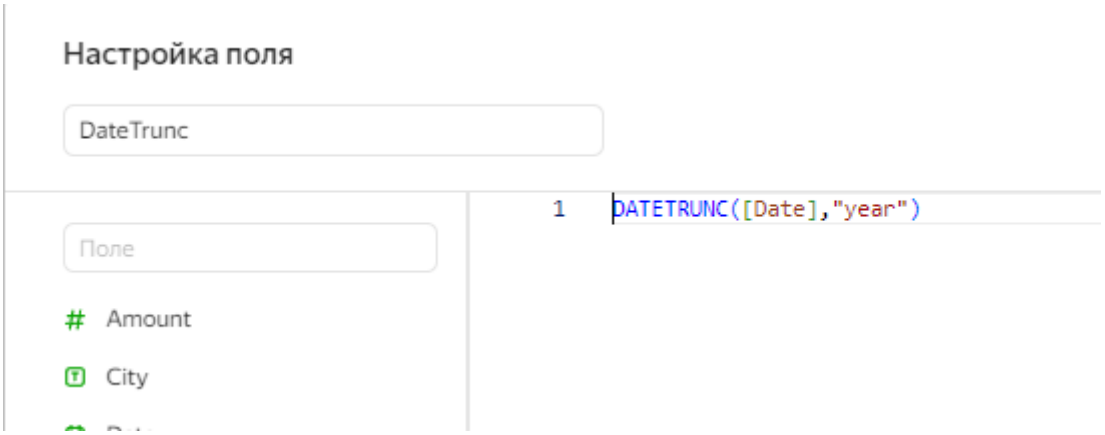
- "*second*";
- "*minute*";
- "*hour*";
- "*day*" (при заданном number используется номер дня в году);
- "*week*";
- "*month*";
- "*quarter*";
- "*year*".

number – также можно указать степень округление до нужного количества временных единиц. Отсутствие аргумента эквивалентно значению 1.

Попробуем округлить даты в нашем примере.

При округлении до года мы увидим, как суммы продаж агрегировались в разрезе двух лет.

DateTrunc	Amount ...
01.01.2022	1 232 299,86
01.01.2023	8 985 934,42



А так будет выглядеть результат по кварталам. Суммы сгруппировались на первое число каждого квартала.

DateTrunc	Amount
01.10.2022	1 232 299,86
01.01.2023	8 985 934,42

DAY

Данная функция возвращает номер дня в месяце необходимой даты. Упрощенный вариант функции **DatePart**.

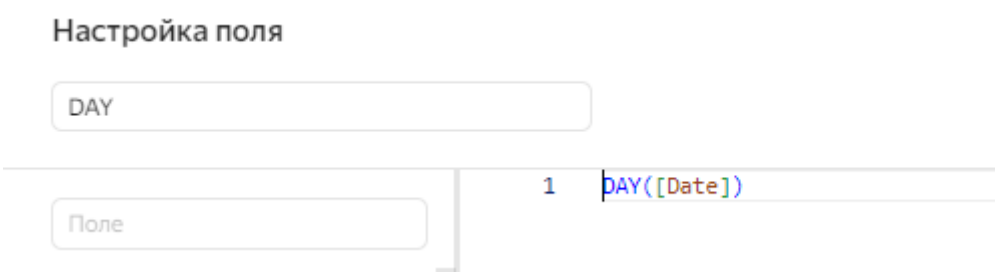
> ***Синтаксис: DAY(datetime)***

datetime – исходная дата, из которой нужно получить результат.

Достаточно просто указать нужную дату, чтобы получить номер дня месяца.

Результат будет выглядеть вот так.

Date	DAY
01.12.2022	1
10.12.2022	10
19.12.2022	19
25.12.2022	25
10.01.2023	10
11.01.2023	11
13.01.2023	13
15.01.2023	15



DAYOFWEEK

Данная функция возвращает номер дня недели в соответствии с *ISO 8601*. Упрощенный вариант функции **DatePart**. По умолчанию 1 – это понедельник, 7 – воскресенье.

Синтаксис: DAYOFWEEK(datetime [, firstday])

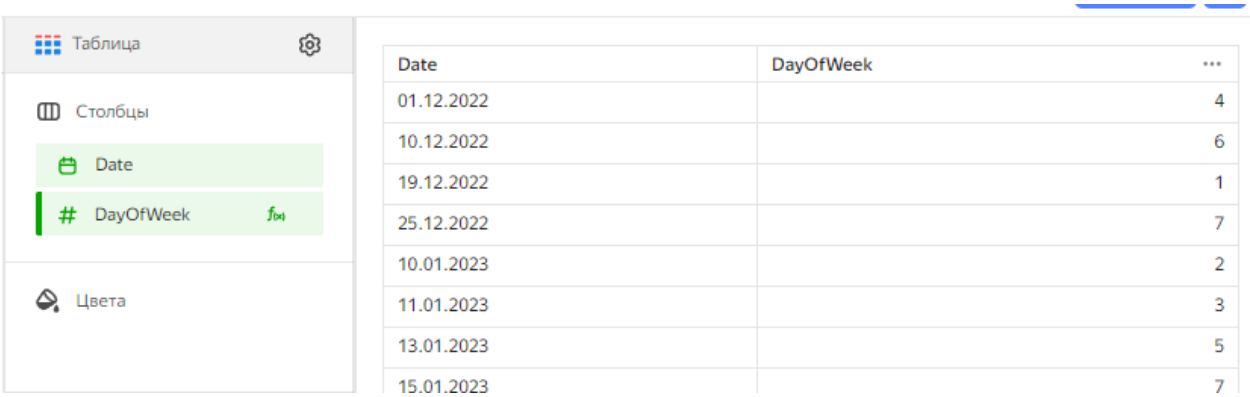
datetime – исходная дата, из которой нужно получить результат.

firstday – с помощью дополнительного параметра можно указать, какой день считать первым в неделе. Это удобно использовать при расчетах показателей с середины недели. Например, если отчет по продажам формируется раз в неделю по четвергам. С помощью данного параметра можно настроить начало операционной недели с четверга. И затем сравнивать показатели продаж с прошлой операционной неделей, которая также начиналась с четверга.

Допустимые значения:

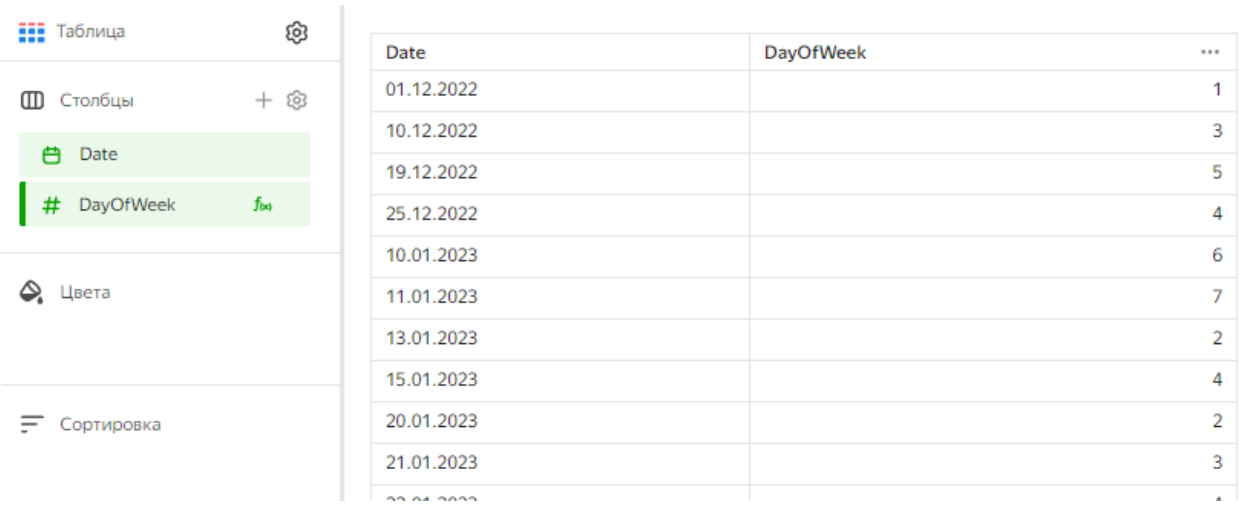
- "*Monday*", "*Mon*" — понедельник;
- "*Tuesday*", "*Tue*" — вторник;
- "*Wednesday*", "Wed" — среда;
- "*Thursday*", "*Thu*" — четверг;
- "*Friday*", "*Fri*" — пятница;
- "*Saturday*", "*Sat*" — суббота;
- "*Sunday*", "*Sun*" — воскресенье.

Вот так будет выглядеть результат при классической неделе.

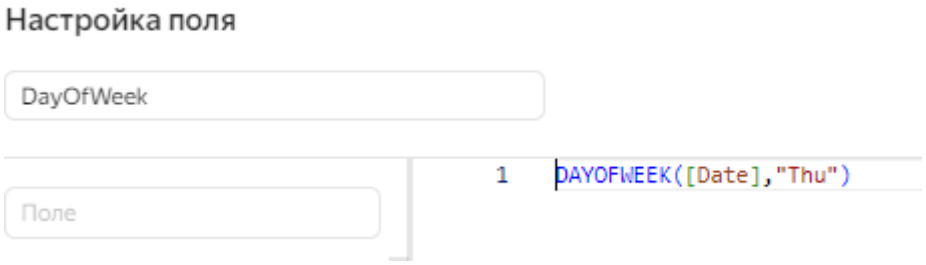


Date	DayOfWeek
01.12.2022	4
10.12.2022	6
19.12.2022	1
25.12.2022	7
10.01.2023	2
11.01.2023	3
13.01.2023	5
15.01.2023	7

А так, если мы установим начало недели – *Четверг*.



Date	DayOfWeek
01.12.2022	1
10.12.2022	3
19.12.2022	5
25.12.2022	4
10.01.2023	6
11.01.2023	7
13.01.2023	2
15.01.2023	4
20.01.2023	2
21.01.2023	3



HOUR

Данная функция возвращает номер часа в дате. Если дата без времени, то результат будет 0.

Синтаксис: HOUR(datetime)

Попробуем на нашем примере.

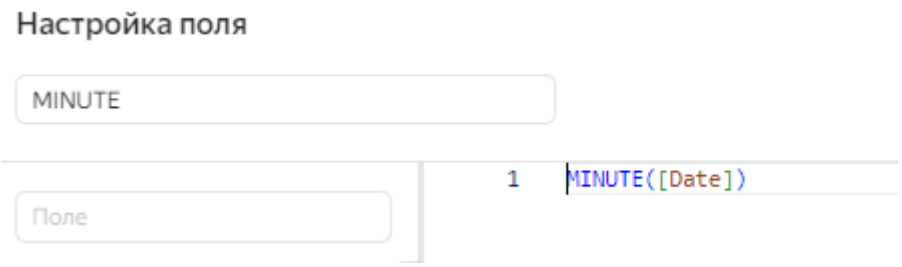
Date	HOUR ...
01.12.2022	0
10.12.2022	0
19.12.2022	0
25.12.2022	0
10.01.2023	0
11.01.2023	0
13.01.2023	0
15.01.2023	0

MINUTE

Данная функция возвращает номер минуты в дате. Если дата без времени, то результат будет 0.

Синтаксис: MINUTE(datetime)

Date	MINUTE
01.12.2022	0
10.12.2022	0
19.12.2022	0
25.12.2022	0
10.01.2023	0
11.01.2023	0
13.01.2023	0



SECOND

Данная функция возвращает номер секунды в минуте в нужной дате. Если дата без времени, то результат будет 0.

Синтаксис: SECOND(datetime)

MONTH

Данная функция возвращает номер месяца в необходимой дате. Упрощенный вариант функции **DatePart**.

Синтаксис: MONTH(datetime)

Date	MONTH
01.12.2022	12
10.12.2022	12
19.12.2022	12
25.12.2022	12
10.01.2023	1
11.01.2023	1
13.01.2023	1
15.01.2023	1
20.01.2023	1
21.01.2023	1

NOW

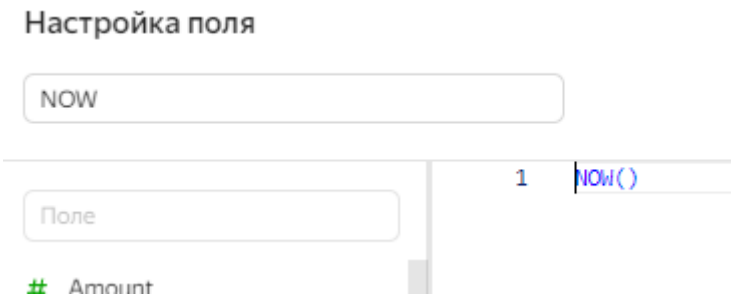
Возвращает текущую дату и время в зависимости от источника данных и типа соединения.

Синтаксис: NOW()

Данная функция не имеет аргументов и параметров, так как просто возвращает системную дату и время.

В нашем примере это будет выглядеть вот так.

Date	NOW
01.12.2022	20.06.2023 14:31:31
10.12.2022	20.06.2023 14:31:31
19.12.2022	20.06.2023 14:31:31
25.12.2022	20.06.2023 14:31:31
10.01.2023	20.06.2023 14:31:31
11.01.2023	20.06.2023 14:31:31
13.01.2023	20.06.2023 14:31:31
15.01.2023	20.06.2023 14:31:31



QUARTER

Данная функция возвращает номер квартала (от 1 до 4) в необходимой дате. Упрощенный вариант функции **DatePart**.

Синтаксис: QUARTER(datetime)

Date	QUARTER ...
01.12.2022	4
10.12.2022	4
19.12.2022	4
25.12.2022	4
10.01.2023	1
11.01.2023	1
13.01.2023	1
15.01.2023	1
20.01.2023	1
21.01.2023	1
22.01.2023	1

TODAY

Данная функция очень похожа на NOW(). Но она возвращает лишь текущую дату, без времени.

Синтаксис: TODAY()

Date	TODAY
01.12.2022	20.06.2023
10.12.2022	20.06.2023
19.12.2022	20.06.2023
25.12.2022	20.06.2023
10.01.2023	20.06.2023
11.01.2023	20.06.2023
13.01.2023	20.06.2023

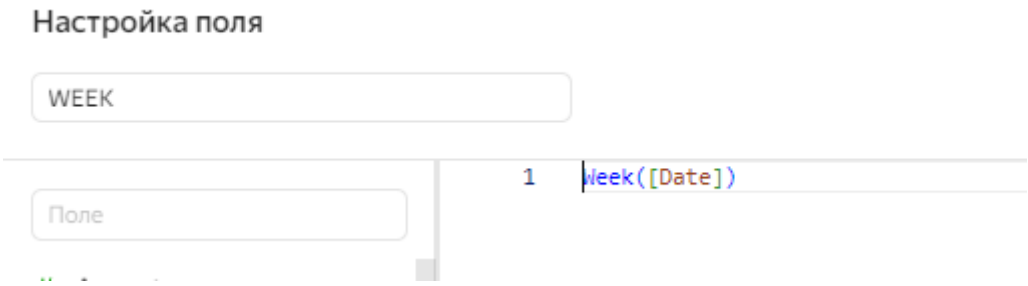
WEEK

Данная функция возвращает номер недели в году в соответствии с ISO 8601. Первой считается неделя, которая содержит первый четверг года и 4.01.

Синтаксис: WEEK(value)

value – исходная дата.

Date	WEEK
01.12.2022	48
10.12.2022	49
19.12.2022	51
25.12.2022	51
10.01.2023	2
11.01.2023	2
13.01.2023	2
15.01.2023	2
20.01.2023	3



YEAR

Данная функция возвращает год необходимой даты. Упрощенный вариант функции **DatePart**.

Синтаксис: YEAR(datetime)

datetime – исходная дата, из которой нужно получить результат.

Date	YEAR
01.12.2022	2022
10.12.2022	2022
19.12.2022	2022
25.12.2022	2022
10.01.2023	2023
11.01.2023	2023
13.01.2023	2023
15.01.2023	2023
20.01.2023	2023
21.01.2023	2023
22.01.2023	2023

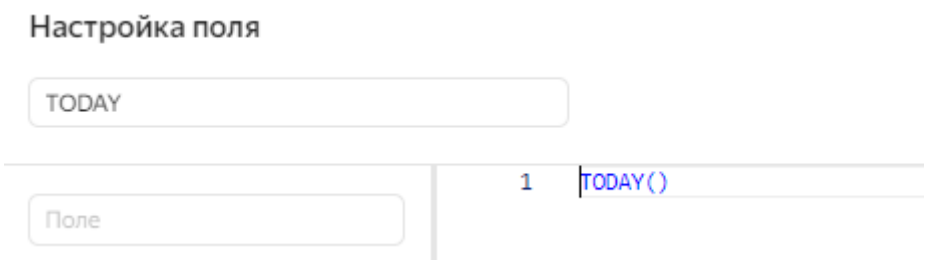
ПРАКТИКА

А теперь давайте рассмотрим несколько простых задач, чтобы на конкретных и более прикладных примерах увидеть, как работают данные функции.

Задача 1

Рассчитать разницу в днях между текущей датой и датой отчета. Это может пригодится, чтобы, например, увидеть количество дней просрочки или простоя.

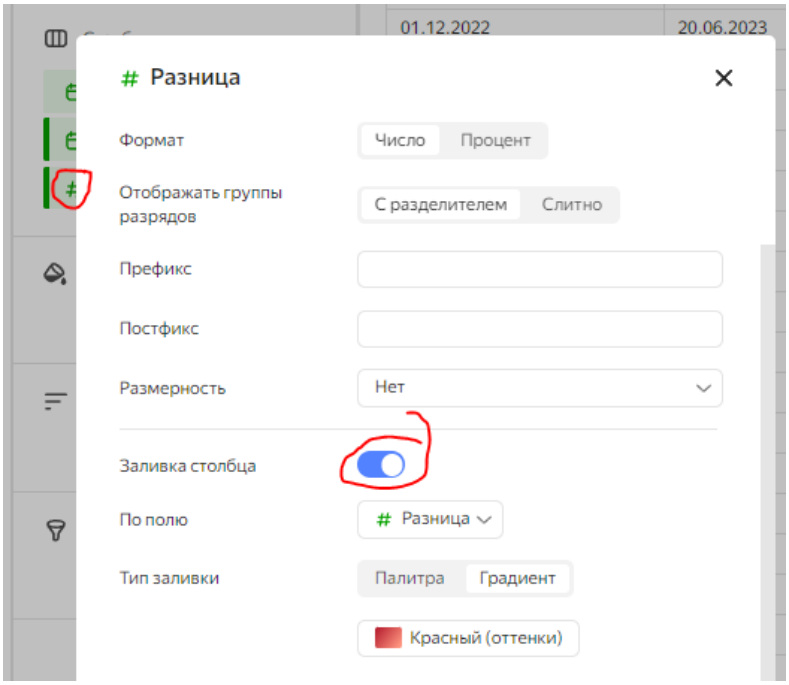
Создадим в чарте два новых поля – с текущей датой на сегодня и разницей между текущей датой и датой отчета. Для наглядности поместим это в разные столбцы.



Выведем оба поля в чарт.

Date	TODAY	Разница
01.12.2022	20.06.2023	201
10.12.2022	20.06.2023	192
19.12.2022	20.06.2023	183
25.12.2022	20.06.2023	177
10.01.2023	20.06.2023	161
11.01.2023	20.06.2023	160
13.01.2023	20.06.2023	158
15.01.2023	20.06.2023	156
20.01.2023	20.06.2023	151
21.01.2023	20.06.2023	150
22.01.2023	20.06.2023	149
23.01.2023	20.06.2023	148
24.01.2023	20.06.2023	147
30.01.2023	20.06.2023	141
05.02.2023	20.06.2023	135
06.02.2023	20.06.2023	134
07.02.2023	20.06.2023	133
10.02.2023	20.06.2023	130
19.02.2023	20.06.2023	121

В последней колонке появилось количество дней. Теперь давайте добавим условную заливку поля, чтобы сразу увидеть, где количество дней самое большое (например, самый долгий простой). Так как даты у нас идут по возрастанию, то градиент тоже распределится равномерно.



Date	TODAY	Разница ...
01.12.2022	20.06.2023	201
10.12.2022	20.06.2023	192
19.12.2022	20.06.2023	183
25.12.2022	20.06.2023	177
10.01.2023	20.06.2023	161
11.01.2023	20.06.2023	160
13.01.2023	20.06.2023	158
15.01.2023	20.06.2023	156
20.01.2023	20.06.2023	151
21.01.2023	20.06.2023	150
22.01.2023	20.06.2023	149
23.01.2023	20.06.2023	148
24.01.2023	20.06.2023	147
30.01.2023	20.06.2023	141
05.02.2023	20.06.2023	135
06.02.2023	20.06.2023	134
07.02.2023	20.06.2023	133
10.02.2023	20.06.2023	130
19.02.2023	20.06.2023	121

Задача 2

Допустим, у нас по условиям договора срок оплаты счета – 2 недели. В отчете у нас есть некие даты, от которых по договору и идет отсчет этих двух недель. Дабы не нарушать сроки оплаты, менеджмент компании хочет видеть этот самый расчетный последний день оплаты, чтобы заранее это отследить и не пропустить.

Для этого нам нужно будет сделать всего лишь один расчет – просто добавить к дате в отчете две недели. И поможет нам в этом функция **DateAdd**.

Формула будет выглядеть вот так. Просто указываем исходную дату, размерность (неделя) и количество. Если бы нам нужно было убавить две недели назад, то указали бы -2.

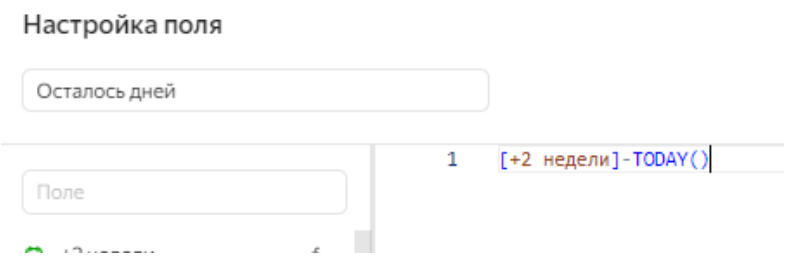
И получаем наш результат.

Date	+2 недели
01.12.2022	15.12.2022
10.12.2022	24.12.2022
19.12.2022	02.01.2023
25.12.2022	08.01.2023
10.01.2023	24.01.2023
11.01.2023	25.01.2023
13.01.2023	27.01.2023
15.01.2023	29.01.2023
20.01.2023	03.02.2023
21.01.2023	04.02.2023
22.01.2023	05.02.2023
23.01.2023	06.02.2023

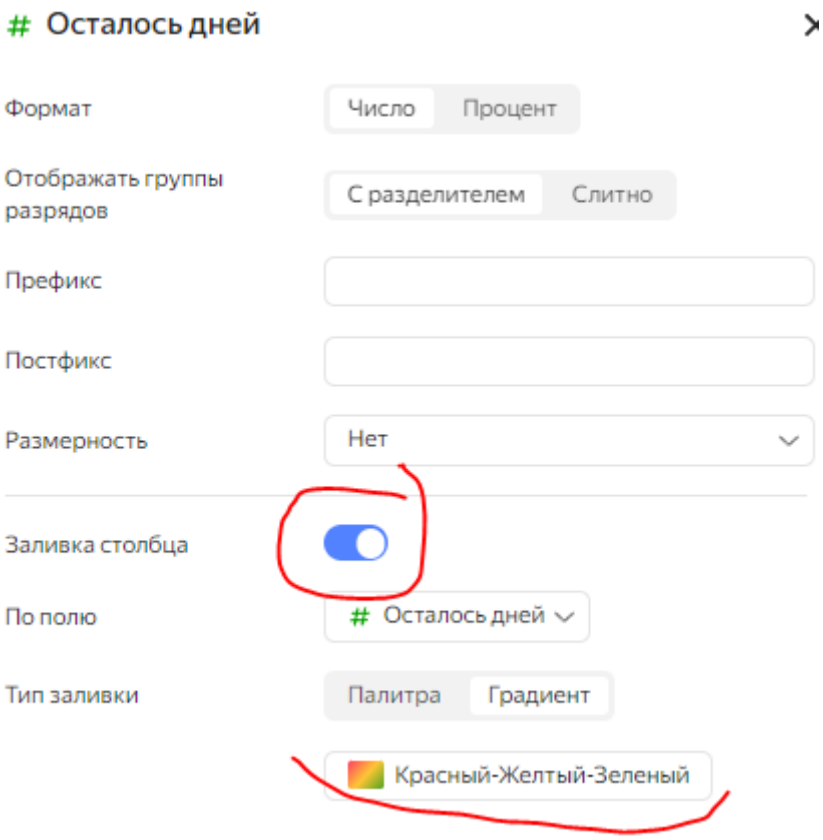
Теперь в отчете сразу видно, когда и по какой поставке истекает срок оплаты.

А теперь давайте рассчитаем, сколько у нас осталось дней для оплаты, и подсветим цветом истекающие сроки оплаты. Предположим, что текущая дата – это 12.12.2022. Добавим для наглядности этот столбец.

И теперь рассчитаем разницу между полем +2 недели и текущей датой.



В этом столбце мы и увидим, сколько дней осталось на оплату. С помощью условной заливки подсветим красным истекающие сроки (где меньше всего дней осталось), а зеленым, кому платить еще нескоро.



В итоге мы получим красивый и информативный отчет, полностью удовлетворяющий поставленную задачу бизнеса.

Date	TODAY	+2 недели	Осталось дней ...
01.12.2022	12.12.2022	15.12.2022	3
10.12.2022	12.12.2022	24.12.2022	12
19.12.2022	12.12.2022	02.01.2023	21
25.12.2022	12.12.2022	08.01.2023	27
10.01.2023	12.12.2022	24.01.2023	43
11.01.2023	12.12.2022	25.01.2023	44
13.01.2023	12.12.2022	27.01.2023	46
15.01.2023	12.12.2022	29.01.2023	48
20.01.2023	12.12.2022	03.02.2023	53
21.01.2023	12.12.2022	04.02.2023	54
22.01.2023	12.12.2022	05.02.2023	55
23.01.2023	12.12.2022	06.02.2023	56
24.01.2023	12.12.2022	07.02.2023	57
30.01.2023	12.12.2022	13.02.2023	63
05.02.2023	12.12.2022	19.02.2023	69
06.02.2023	12.12.2022	20.02.2023	70
07.02.2023	12.12.2022	21.02.2023	71
10.02.2023	12.12.2022	24.02.2023	74
19.02.2023	12.12.2022	05.03.2023	83

Задача 3

А теперь давайте рассмотрим задачку чуть посложнее. Как нам получить порядковый номер дня года. В типовых функциях **DataLense** есть возможность быстро получить только номер дня в месяце. Но очень часто возникает потребность узнать именно номер дня в году. Например, 31 декабря – это обычно 365 день в году. Попробуйте сначала сами догадаться, как это сделать.

На самом деле, сделать это очень просто. В этом на поможет функция **DateTrunc**. Если помните, данная функция изменяет размерность дат и группирует данные на первое число нужной размерности (на первое число квартала или года). Поэтому с ее помощью сначала давайте определим первое число года для каждой даты.

Формула будет выглядеть так.

В нашей таблице рядом с датами появился первый день каждого года.

Date	Начало года
01.12.2022	01.01.2022
10.12.2022	01.01.2022
19.12.2022	01.01.2022
25.12.2022	01.01.2022
10.01.2023	01.01.2023
11.01.2023	01.01.2023
13.01.2023	01.01.2023

И теперь все, что нам надо, это рассчитать разницу в днях между исходной датой слева и началом года.

Но обратите внимание, что к полученному результату нам нужно добавить 1. Так как формула рассчитывает именно разницу, дельту дней, то она не учитывает тот самый первый день года. Его мы и добавляем в конце.

В итоге мы получаем порядковый номер дня в каждом году.

Date	Начало года	Номер дня года ...
01.12.2022	01.01.2022	335
10.12.2022	01.01.2022	344
19.12.2022	01.01.2022	353
25.12.2022	01.01.2022	359
10.01.2023	01.01.2023	10
11.01.2023	01.01.2023	11
13.01.2023	01.01.2023	13
15.01.2023	01.01.2023	15
20.01.2023	01.01.2023	20
21.01.2023	01.01.2023	21
22.01.2023	01.01.2023	22
23.01.2023	01.01.2023	23
24.01.2023	01.01.2023	24
30.01.2023	01.01.2023	30
05.02.2023	01.01.2023	36
06.02.2023	01.01.2023	37
07.02.2023	01.01.2023	38
10.02.2023	01.01.2023	41
19.02.2023	01.01.2023	50

На этом все! Надеюсь, что данный урок был вам полезен.

СОЦИАЛЬНЫЕ СЕТИ

А также подписывайтесь на нас тут:


https://vk.com/biba_pro


https://ok.ru/group/70000002569850


https://t.me/biba_pro


https://dzen.ru/bibapro

https://rutube.ru/channel/25695571/