

DATAΛAD

Массовые операции без миллиона ячеек

Формулы столбца, SQL и соседние строки

500

Формулы ячеек

Локальная Excel-логика

10K

Копирование значения

Ограниченный диапазон

100K

Замена по фильтру

Атомарный batch

SQL

Миллионы строк

Один set-based план

IF / CASE - LAG / LEAD - оконные суммы - Auto Parts

1. Лимиты - это границы инструментов

Короткий ответ

Числа выбраны не случайно, но это не физический предел компьютера. Они не позволяют превратить массовый SQL-расчёт в миллион отдельных объектов Grid.

Предел	Операция	Почему
500	протяжка формул	каждая ячейка хранит и вычисляет свою формулу
10 000	копирование значения	каждая цель становится отдельным SetCell
100 000	замена по фильтру	атомарный batch хранит отдельные изменения

500 формул ячеек

- отдельный текст формулы и отдельное вычисленное значение;
- проверка адреса, типа и циклической ссылки;
- отдельная запись истории и повторный расчёт;
- после 500 строк задача обычно уже является формулой столбца.

10 000 скопированных значений

- константа читается один раз, но цели перечисляются по строкам;
- сервер проверяет идентификаторы и пишет журнал изменений;
- предел защищает от случайного million-mod commit.

100 000 изменений по фильтру

Фильтр выполняется сервером над всем представлением, но результат пока разворачивается в адресные изменения ячеек. Поэтому фильтр снимает зависимость от загруженной страницы, но не делает миллион записей бесплатным.

Что можно настраивать

Точные числа можно пересматривать по результатам benchmark. Но для миллиона строк нужен другой план выполнения, а не просто более высокий limit.

2. Стоимость: строка за строкой или один план

Модель	Хранение	Вычисление	Подходящий масштаб
Формула ячейки	N формул + N значений	по каждой формуле	единицы / сотни
Копирование	N SetCell	константа одна, записей N	тысячи
Формула столбца	одно правило	векторный SQL-проектор	миллионы
SQL-запрос	один план	set-based / потоково	миллионы

Почему миллион тяжелее ожидаемого

У констант базовая стоимость близка к линейной: в миллион строк нужно записать примерно миллион изменений. Но после порогов начинают влиять размер запроса, память, сериализация, сортировка RID, журнал, Parquet-слои и компактация.

У формул ячеек стоимость выше: каждая формула отдельно проверяется и вычисляется относительно датасета. Миллион формул - это миллион самостоятельных мини-программ.

Плохая модель

Отфильтровать 900 000 строк и создать 900 000 формул, затем повторить для ещё 100 000.

Правильная модель

Сохранить одно правило IF / CASE и выполнить его над всем столбцом.

Быстрый выбор

Задача	Инструмент
особая ячейка	формула отдельной ячейки
локальный диапазон	протяжка / заполнение
одно правило для столбца	формула столбца
разные правила по категориям	один IF или SQL CASE
соседние строки на большом наборе	SQL-окно
большой статический результат	новый датасет / потоковая перезапись

3. Миллион строк: одно условие IF

Пусть segment=A содержит 900 000 строк, а segment=B - 100 000. Не запускайте две протяжки. Перенесите условие фильтра внутрь формулы.

```
=IF([segment]="A"; <формула_A>; <формула_B>)
```

Если есть другие группы

```
=IF(  
  [segment]="A";  
  <формула_A>;  
  IF([segment]="B"; <формула_B>; "")  
)
```

Auto Parts: разные алгоритмы по каналу продаж

```
=IF(  
  [канал_продаж]="Дилер";  
  [количество]*[цена_за_единицу];  
  [количество]*[цена_за_единицу]*(1-[скидка_процент]/100)  
)
```

Условие по дате

```
=IF(  
  [дата_заказа]>=DATE(2025;7;1);  
  "Новый период";  
  "Старый период"  
)
```

Одна формула - один объект

Движок обрабатывает столбец векторно. Он не сохраняет копию формулы в каждой из миллиона строк.

Та же логика в SQL

```
SELECT  
  t.*,  
  CASE  
    WHEN segment = 'A' THEN <выражение_A>  
    WHEN segment = 'B' THEN <выражение_B>  
    ELSE NULL  
  END AS calculated_value  
FROM input AS t;
```

4. Строка выше и строка ниже

Почему формула столбца не видит соседей

Формула столбца работает с текущей строкой. Сортировка и фильтр могут изменить соседей на экране, поэтому скрытая зависимость от визуальной позиции сделала бы результат нестабильным.

Небольшой участок: формула ячейки

Пусть значения находятся в столбце E. В целевой ячейке строки 2 можно сложить предыдущую, текущую и следующую строки:

```
=E1+E2+E3
```

Целевая строка	Формула	Какие строки читает
2	=E1+E2+E3	1, 2, 3
3 после протяжки	=E2+E3+E4	2, 3, 4
4 после протяжки	=E3+E4+E5	3, 4, 5

Это правильная Excel-подобная модель для небольшого участка. Каждая строка получает самостоятельную формулу, поэтому действует предел 500.

Границы

У первой строки нет предыдущей, у последней нет следующей. Для них нужна отдельная формула или SQL-окно, которое естественно уменьшит рамку.

Большой набор: не протяжка, а SQL-окно

Для сотен тысяч строк используйте window function - оконную функцию. Она вычисляет соседство одним планом над упорядоченным набором.

5. LAG, LEAD и скользящая сумма

Предыдущее и следующее значения:

```
SELECT
t.*,
LAG("сумма_итого") OVER (
  ORDER BY "дата_заказа", "номер_заказа", "номер_строки"
) AS previous_total,
LEAD("сумма_итого") OVER (
  ORDER BY "дата_заказа", "номер_заказа", "номер_строки"
) AS next_total
FROM input AS t;
```

Сумма предыдущей, текущей и следующей строк:

```
SELECT
t.*,
SUM("сумма_итого") OVER (
  ORDER BY "дата_заказа", "номер_заказа", "номер_строки"
  ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
) AS three_row_total
FROM input AS t;
```

Отдельная последовательность для каждой детали:

```
SELECT
t.*,
SUM("сумма_итого") OVER (
  PARTITION BY "id_детали"
  ORDER BY "дата_заказа", "номер_заказа", "номер_строки"
  ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
) AS part_three_row_total
FROM input AS t;
```

LAG / LEAD

Возвращают конкретное предыдущее или следующее значение.

ROWS BETWEEN

Задаёт рамку для SUM, AVG и других оконных агрегатов.

6. ORDER BY определяет соседство

Главное правило

В реляционной таблице нет естественной строки выше или ниже. Соседство возникает только после явного ORDER BY.

Дата заказа повторяется, а заказ может иметь несколько строк. Поэтому для стабильного порядка используются три поля:

```
ORDER BY "дата_заказа", "номер_заказа", "номер_строки"
```

PARTITION BY - отдельные последовательности

PARTITION BY "id_детали" начинает окно заново для каждой детали. Без PARTITION BY соседство считается по всему датасету.

Фактический результат на Auto Parts

Заказ	Текущая	Предыдущая	Следующая	Сумма 3
483271	419.87	NULL	1438.42	1858.29
721859	1438.42	419.87	206.07	2064.36
782915	206.07	1438.42	2434.23	4078.72
736595	2434.23	206.07	959.28	3599.58

Проверено текущим SQL-движком на строках одной детали. В первой строке раздела previous_total равен NULL, а сумма рамки состоит из двух существующих строк.

Ещё полезные окна

Задача	SQL-идея
изменение от предыдущей	value - LAG(value) OVER (...)
накопительный итог	ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
среднее по трём	ROWS BETWEEN 2 PRECEDING AND CURRENT ROW

7. Как получить большой статический результат

Живая формула столбца хранит одно правило и вычисляется при чтении. Если результат должен стать обычными значениями на миллионе строк, его нужно материализовать set-based способом.

Шаг	Действие
1	описать правило формулой столбца или SQL
2	проверить preview, NULL и граничные строки
3	выполнить один план над всем датасетом
4	сохранить новый датасет или потоково переписать Parquet
5	не дробить миллион строк на сотни операций по 10 000

Правильная единица работы

Для большого расчёта единицей является правило или SQL-план, а не адрес отдельной ячейки.

Итоговая памятка

- до сотен ячеек думайте как в Excel;
- для всего столбца думайте как в SQL;
- одно правило храните один раз;
- условия фильтра переносите внутрь IF или CASE;
- для соседних строк всегда задавайте ORDER BY;
- для независимых последовательностей добавляйте PARTITION BY;
- миллион строк обрабатывайте векторным планом или потоковой перезаписью.

Локальная работа

Формула ячейки, протяжка, ручное исключение.

Массовая работа

Формула столбца, CASE, оконная функция, новый датасет.