

[Нюансы платформы 1С:Предприятие 8] - Как работает свойство БлокироватьДляИзменения

Как работает свойство **БлокироватьДляИзменения** набора записей регистра накопления и бухгалтерии, известно далеко не всем специалистам по 1С.

С одной стороны, вроде бы многие знают, когда это свойство нужно использовать (если не знаете – предварительно прочитайте статью про контроль остатков: <http://kursy-po-1c.pf/articles/2017-02-12-two-methods-for-inventory-check/> (<http://kursy-po-1c.ru/articles/2017-02-12-two-methods-for-inventory-check/>)).

С другой стороны, **как это работает**, понимают немногие. Показательно будет привести цитату из справки по свойству **БлокироватьДляИзменения**:

*Устанавливает режим, при котором в процессе записи набора будет **установлена управляемая блокировка** для всех комбинаций измерений в соответствии с записями набора записей.*

На самом деле никакую блокировку это свойство **не устанавливает!** Для чего же так пишут справке? Сложно сказать – возможно, чтобы не усложнять жизнь начинающим разработчикам :)

В результате это приводит к неправильному использованию данного инструмента для блокировок. В этой статье докопаемся до истины и покажем внутренние детали реализации этого механизма платформы «1С:Предприятие 8.3».

Для того чтобы точно понять всю статью, необходимо знать, что такое механизм разделения итогов. Этому механизму посвящена отдельная статья на нашем сайте: <http://kursy-po-1c.pf/articles/режим-разделения-итогов/> (<http://kursy-po-1c.ru/articles/режим-разделения-итогов/>).

Начнем с того, что работа свойства **БлокироватьДляИзменения** отличается для платформы 8.2 и 8.3. Поэтому рассмотрим 2 сценария.

Использование платформы «1С:Предприятие 8.2» и MS SQL Server

Допустим, что 2 пользователя одновременно проводят 2 документа продажи товаров (пусть это будут столы). При этом в конфигурации используется новая методика контроля остатков (сначала запись движений, потом чтение остатков).

Если у регистра накопления с остатками товаров **включено разделение итогов**, можем получить следующую ситуацию:

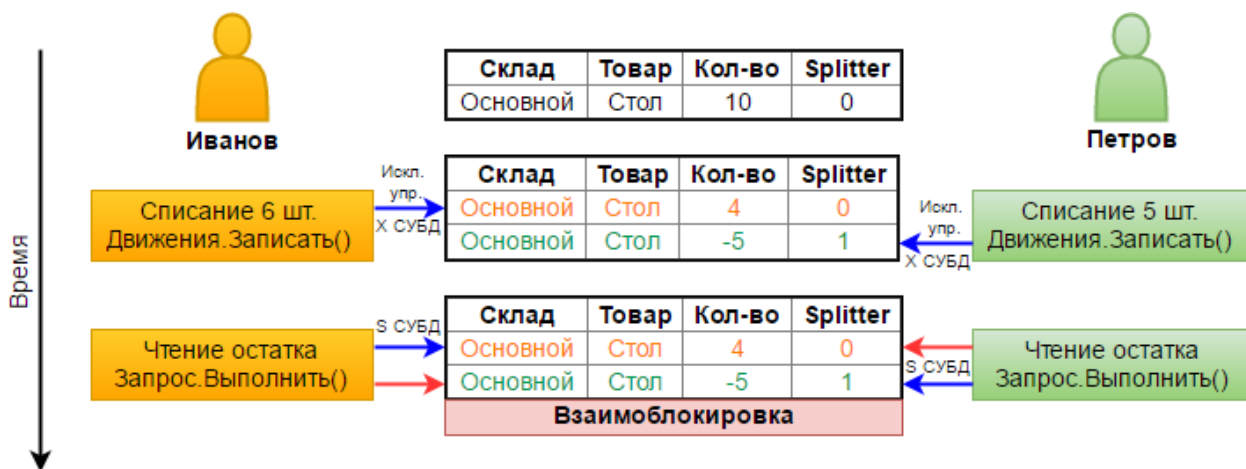


Рисунок 1. Схема взаимной блокировки

В представленной ситуации 2 пользователя списывают товар «Стол» со склада «Основной», при этом на складе осталось всего 10 столов.

Так как включено разделение итогов, то оба пользователя параллельно делают движения в таблице итогов, накладывая при этом исключительную управляемую блокировку и X-блокировку СУБД на свои строки. Блокировка будет наложена по полям Склад + Товар + Разделитель, и так как **разделитель (splitter) разный**, то обе блокировки установятся успешно.

Далее каждый из пользователей пытается выполнить контроль остатков с помощью запроса, и здесь начинается самое интересное.

Запрос к регистру пытается прочитать все строки, где склад равен «Основной» и товар равен «Стол», **без условия по разделителю**. При чтении запрос накладывает S-блокировку СУБД.

Иванов может прочитать свою строку, но не может прочитать строку, которую занял пользователь Петров (так как S и X-блокировки не совместимы). В итоге Иванов ждет Петрова. Петров, в свою очередь, может прочитать свою строку, но не может прочитать строку, занятую Ивановым, и тоже встает в очередь.

На схеме синие стрелки показывают успешно наложенные блокировки, красные показывают ожидание на блокировке.

В результате мы получаем **взаимную блокировку**: Иванов ждет Петрова, а Петров ждет Иванова. Как решить эту проблему, рассмотрим чуть позже. А сейчас перейдем к следующему сценарию.

Отметим, что взаимоблокировки не было бы, если бы режим разделения итогов регистра был отключен. Но, во-первых, режим разделения итогов в общем случае положительно влияет на производительность. Во-вторых, нужно всегда делать устойчивые решения, чтобы при неаккуратных действиях ничего не сломалось :)

Использование платформы «1С:Предприятие 8.3» или использование в качестве СУБД версионника

Сейчас мы рассмотрим сценарий, когда используется платформа 8.3 без режима совместимости с 8.2. Либо когда в качестве СУБД выступает версионник (PostgreSQL, Oracle) – не важно, с какой платформой и в каком режиме.

Кратко отметим, что особенность СУБД версионника в том, что при чтении он не блокирует данные. При использовании 8.3 без режима совместимости с 8.2, MS SQL Server тоже может работать как версионник.

В данных условиях оба пользователя смогут прочитать остаток, так как чтение будет не блокирующим и в итоге мы получим **минус на складе**.

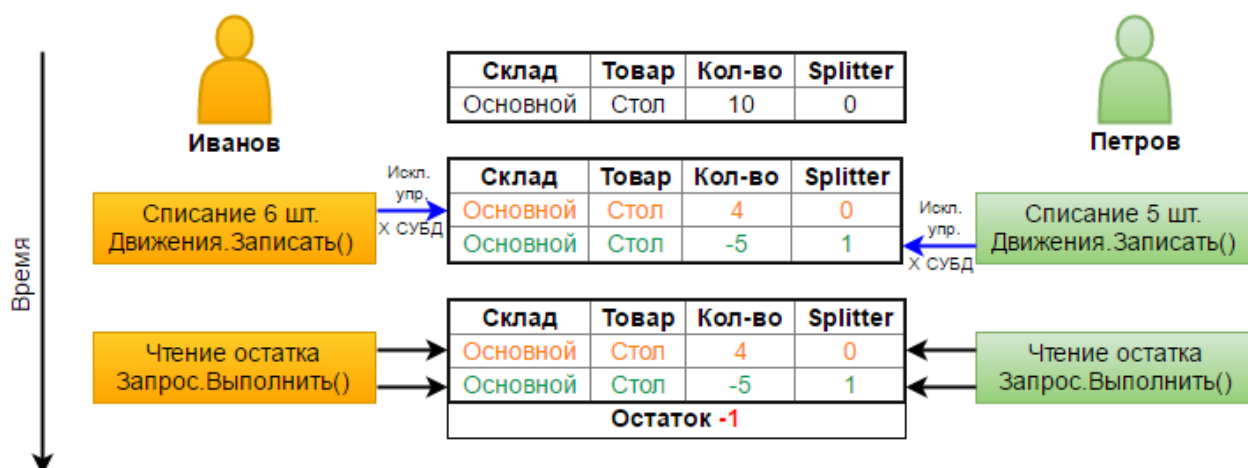


Рисунок 2. Схема получения отрицательных остатков

Как исправить проблемы?

И ситуация взаимоблокировки, и, тем более, отрицательные остатки на складе не могут считаться корректной работой программы.

Чтобы не допустить этих проблем, как раз и было придумано свойство набора записей регистров накопления и бухгалтерии – **БлокироватьДляИзменения**.

Если до записи движений установить данное свойство в значение «Истина», то **блокировка будет накладываться без учета разделителя**. То есть поведение будет таким, как будто разделитель итогов у регистра выключен.

Следует отметить что блокировка будет наложена вне зависимости от значения данного свойства. Если **БлокироватьДляИзменения** имеет значение «Истина», то блокировка будет без учета разделителя, иначе – с учетом разделителя. По умолчанию значение свойства **БлокироватьДляИзменения** равно «Ложь».

Допустим программист внес исправления в код:

```
Движения.ТоварыНаСкладах.БлокироватьДляИзменения=Истина;  
Движения.Записать ();
```

При этом строку «БлокироватьДляИзменения=Истина» можно писать в любом месте кода, но обязательно перед записью движений. Данная строка просто говорит платформе: «когда будешь записывать движения, не учитывай разделитель итогов».

Следует понимать, что сам разделитель итогов при этом никуда не исчезает, просто блокировка будет на поля Склад + Товар без учета разделителя.

Рассмотрим поведение системы, если использовать данное свойство.



Рисунок 3. Схема работы при «БлокироватьДляИзменения=Истина»

При наложении блокировки без учета разделителя оба пользователя попытаются обратиться к одной строке. В результате Петров встанет в очередь (исключительные управляемые блокировки несовместимы) и будет ожидать, пока строка не освободится.

При этом ожидание будет происходить на управляемых блокировках, то есть на сервере 1С. Сервер СУБД даже не будет «знать» о том, что транзакция Петрова ожидает своей очереди.

Данное ожидание на блокировке будет являться необходимым – оно обеспечивает корректную бизнес-логику приложения.

Таким образом мы предотвратили возможную взаимную блокировку или минус на складе с помощью свойства **БлокироватьДляИзменения**.

Несколько вопросов и заблуждений по теме

Часто нам задают вопросы по использованию свойства **БлокироватьДляИзменения**. Рассмотрим наиболее популярные из них.

Вопрос 1

Почему **БлокироватьДляИзменения** не устанавливают в истину в документе «Приходная накладная»? Ведь если я перепровожу документ ПриходнаяНакладная, то должен блокировать поля, по которым был приход, иначе можно товар списать в минус. Пример: было – Стул 2 шт, перепровожу, меняя Стул на Тумбу, и в этот момент списываю Стул в минус.

Ответ: **БлокироватьДляИзменения** ставят там, где у регистра включено разделение итогов и идет контроль остатков после записи. В приходе же при первом проведении контроль остатков не выполняется. Если при перепроведении приходной накладной в конфигурации реализован контроль остатков, то имеет смысл использовать свойство **БлокироватьДляИзменения**.

Вопрос 2

Нужно ли использовать **БлокироватьДляИзменения** в файловом режиме?

Ответ: В файловом режиме блокировка всегда идет на всю таблицу, следовательно режим разделения итогов там бесполезен, поэтому и **БлокироватьДляИзменения** ставить не обязательно, но желательно. Следует помнить, что файловая база со временем может стать клиент-серверной, и тогда данное свойство необходимо будет использовать.

Вопрос 3

Нужно ли устанавливать **БлокироватьДляИзменения** при очистке движений?

Ответ: В большинстве случаев не нужно, так как обычно после очистки движений контроль остатков не выполняется. Если контроль сразу после очистки выполняется, тогда **БлокироватьДляИзменения** ставить нужно.

Вопрос 4

Нужно ли использовать свойство **БлокироватьДляИзменения** при автоматических блокировках?

Ответ: При попытке использовать **БлокироватьДляИзменения** в автоматическом режиме блокировок система выдаст сообщение об ошибке: «Ошибка записи! Блокировка для изменения запрещена для автоматического режима блокировки». Данное свойство можно использовать только для управляемого режима блокировок.

Резюмируя, можно дать следующую рекомендацию: если по регистру при любой записи всегда есть контроль остатков, тогда разделение итогов по нему лучше отключить, так как оно не имеет смысла – везде придется устанавливать свойство **БлокироватьДляИзменения**.

Если же по регистру контроль остатков нужен лишь в определенных документах, тогда в этих документах нужно поставить «БлокироватьДляИзменения=Истина», чтобы предотвратить описанные выше проблемы.

Выводы

Свойство набора записей **БлокироватьДляИзменения** само по себе ничего не блокирует, оно лишь определяет, будет ли в момент записи при блокировке учитываться разделитель или нет.

Следует использовать данное свойство только при выполнении всех 3 условий:

1. Используется новая методика контроля остатков
2. У регистра включено разделение итогов
3. Используется управляемый режим блокировок.

Надеемся, теперь механика работы платформы будет ясна и белых пятен в знаниях специалистов 1С станет меньше :)

Андрей Бурмистров

2017 г.

Подробнее тема оптимизации 1С рассмотрена в нашем курсе

«Ускорение и оптимизация систем на 1С:Предприятие 8.3.

Подготовка на 1С:Эксперт по технологическим вопросам

Описание курса: <http://курсы-по-1с.рф/1с-v8/optimization/> (<http://kursy-po-1c.ru/1c-v8/optimization/>)