Глава 9. Работа с данными

9.1. Механизм объектных блокировок

9.1.1. Общая информация

При работе с объектными данными (справочники, документы, счета и пр.) система «1С:Предприятие» обеспечивает два вида объектных блокировок - **пессимистическую** и **оптимистическую**. Они позволяют выполнять целостные изменения объектов при одновременной работе нескольких пользователей.

9.1.2. Пессимистическая блокировка

Механизм пессимистической блокировки объектов базы данных предназначен для того, чтобы запретить изменение данных объекта другими сеансами или данным сеансом до тех пор, пока блокировка не будет снята (автоматически или с помощью методов встроенного языка).

В основном механизм пессимистической блокировки используется системой «1С:Предприятие» для блокировки объектов, редактируемых в форме. В то же время разработчик имеет возможность задействовать этот механизм, используя средства встроенного языка.

Система «1С:Предприятие» использует механизм пессимистической блокировки с помощью расширений форм прикладных объектов. В тот момент, когда пользователь начинает модификацию объекта в форме, расширение формы устанавливает пессимистическую блокировку. Когда пользователь, редактировавший объект, закроет форму объекта, расширение формы снимет пессимистическую блокировку. Если во время редактирования объекта другой пользователь (или этот же пользователь, но из другого сеанса) попытается начать редактирование этого же объекта, то система сообщит об этом. В зависимости от ситуации, может быть предложено несколько вариантов выхода из сложившейся ситуации:

- В файловом варианте:
  - Система взаимодействия не доступна: выводится подробная информация о пользователе, заблокировавшем объект.
  - Система взаимодействия доступна: выводится информация о пользователе, который выполнили блокировку объекта, и предоставляется возможность начать редактирование. В случае начала редактирования изменения, выполненные в другом сеансе, будут утеряны.

  Если объект заблокирован другим пользователем - предоставляется возможность написать сообщение этому пользователю с использованием системы взаимодействия.
- В клиент-серверном варианте:
  - Если объект заблокирован этим же пользователем в другом сеансе - предоставляется возможность начать редактирование с утерей изменений, выполненных в другом сеансе.
  - Если объект редактируется другим пользователем, то предоставляется возможность начать редактировать объект с утерей изменений, сделанных другим пользователем. При этом предоставляется возможность написать сообщение другому пользователю, если доступна система взаимодействия.

В том случае, если необходимо в нестандартной форме объекта обеспечить такое же поведение, что и в стандартной форме объекта, можно использовать метод формы `ЗаблокироватьДанныеФормыДляРедактирования()` для установки пессимистической блокировки и метод формы `РазблокироватьДанныеФормыДляРедактирования()` для снятия блокировки.

Разработчик, для того чтобы задействовать пессимистическую блокировку, может использовать метод глобального контекста `ЗаблокироватьДанныеДляРедактирования()`. Возможны два варианта установки пессимистической блокировки:

- С указанием идентификатора формы - в этом случае блокировка устанавливается на время жизни форм. При закрытии формы блокировка будет снята сразу, если используется обычное соединение или по прошествии некоторого времени, если используется медленное

соединение. Во всех следующих случаях блокировка будет снята сразу:

- Завершен сеанс, в котором открыта форма.
- Прошла 1 минута после сброса признака модифицированности формы.
- За время отображения формы были установлены другие блокировки от имени этой формы (при интерактивном редактировании или методом `ЗаблокироватьДанныеФормыДляРедактирования()`).
- Закрывается форма, от имени которой были запущены и не завершены фоновые задания (поиск в динамическом списке или формирование отчета).
- Использован метод глобального контекста `РазблокироватьДанныеДляРедактирования()` с указанием того же идентификатора формы, который указывался для установки блокировки.

- Без указания идентификатора формы - в этом случае устанавливаемая блокировка не привязана к какой-либо форме. Блокировка будет автоматически снята при завершении сеанса, при возврате управления с сервера или при завершении транзакции (если блокировка устанавливалась в транзакции). Также блокировка может быть снята с помощью метода глобального контекста `РазблокироватьДанныеДляРедактирования()` без указания идентификатора формы.

Однако следует учитывать, что сам по себе факт установки блокировки не препятствует изменению или удалению объекта в базе данных. Поэтому для того, чтобы обеспечить невозможность изменения заблокированного объекта, операции изменения объекта в другом сеансе также должна предшествовать попытка блокировки этого объекта. Блокировка заблокированного объекта базы данных вызывает исключение, которое может быть обработано конструкцией `Попытка … Исключение … КонецПопытки`.

Копировать в буфер обмена

```
&НаСервере
Функция ПримерМодификации()
    ТоварСсылка = Справочники.Товары.НайтиПоКоду("000000001");
    Попытка
        ЗаблокироватьДанныеДляРедактирования(ТоварСсылка);
        // Можно выполнять модификацию данных объекта
        // ...
        ТоварОбъект = ТоварСсылка.ПолучитьОбъект();
        ТоварОбъект.Наименование = "Новое наименование";
        ТоварОбъект.Записать();
        Возврат Истина;
    Исключение
        // Нельзя модифицировать данные объекта
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = "Данные объекта уже заблокированы";
        Сообщение.Сообщить();
        Возврат Ложь;
    КонецПопытки;
КонецФункции
```

Следует помнить, что попытки установить блокировку одного и того же объекта с указанием идентификатора формы и без указания идентификатора несовместимы друг с другом.

Копировать в буфер обмена

```
ТоварСсылка = Справочники.Номенклатура.НайтиПоКоду(1);
ЗаблокироватьДанныеДляРедактирования(ТоварСсылка);
Попытка
    ЗаблокироватьДанныеДляРедактирования(ТоварСсылка, , ИдентФормы);
Исключение
    // исключение из за несовместимости блокировок
КонецПопытки;
```

Для снятия пессимистической блокировки разработчик может использовать метод глобального контекста `РазблокироватьДанныеДляРедактирования()`.

Смотри также:

- Система взаимодействия (см. здесь).

9.1.3. Пессимистическая блокировка и транзакции

Операции блокировки объектов влияют только на выполнение других операций блокировки объектов и не влияют на операции над данными и на процесс течения транзакций.

Блокировка заблокированного объекта базы данных вызывает исключение, которое может быть обработано и не приводит к обязательному откату транзакции. Если в течение транзакции при выполнении метода `ЗаблокироватьДанныеДляРедактирования()` возникло исключение, то оно может быть обработано конструкцией `Попытка … Исключение … КонецПопытки` и не требует обязательного отката транзакции.

Блокировки объектов, установленные в течение транзакции, снимаются при окончании транзакции, если блокировка устанавливалась без указания идентификатора формы.

9.1.4. Оптимистическая блокировка

Оптимистическая блокировка запрещает запись объекта в базу данных, если после считывания объекта он был изменен в базе данных.

Строго говоря, оптимистическая блокировка представляет собой проверку, которая выполняется перед записью объекта в базу данных.

Когда объект встроенного языка считывает данные из базы данных, в числе прочего считывается и версия объекта, хранящегося в базе данных.

Если до начала редактирования данных пользователем (до установки пессимистической блокировки) данные объекта в базе данных были изменены (например, другим пользователем), то номер версии объекта, хранящийся в базе данных, также изменится. При попытке пользователя записать этот объект будет выполнена проверка соответствия версии объекта, находящегося в памяти, и версии объекта, хранящейся в базе данных. Так как версии отличаются, будет выдано предупреждение о том, что версия объекта изменилась или он был удален, то есть сработает оптимистическая блокировка.

Оптимистическая блокировка гарантирует, что если пользователь изменяет объект, то его изменения не «затрут» изменения, сделанные другими сеансами или другими программными объектами этого же сеанса.

9.2. Механизм транзакций

9.2.1. Общая информация

Независимо от выбранного варианта работы (файловый или клиент-серверный) система «1С:Предприятие» обеспечивает работу с информацией, хранящейся в базе данных, с использованием механизма транзакций.

Транзакция - это неделимая, с точки зрения воздействия на базу данных, последовательность операций манипулирования данными. Она выполняется по принципу «все или ничего» и переводит базу данных из одного целостного состояния в другое целостное состояние. Если по каким-либо причинам одно из действий транзакции невыполнимо или произошло какое-либо нарушение работы системы, база данных возвращается в то состояние, которое было до начала транзакции (происходит откат транзакции).

Транзакции могут использоваться как самой системой «1С:Предприятие», так и разработчиком при написании модулей.

Система «1С:Предприятие» определяет неявную транзакцию при выполнении любых действий, связанных с модификацией информации, хранящейся в базе данных. Например, все обработчики событий, расположенные в модулях объектов и наборов записей, связанные с модификацией данных базы данных, вызываются в транзакции. В то же время чтение объектов базы данных (обращение к свойствам объекта «через точку», выполнение метода `.ПолучитьОбъект()`, открытие формы объекта или набора записей), как правило, выполняется вне транзакции для повышения параллельности работы системы. В рамках такого чтения система обеспечивает, что состояние прочитанного объекта в памяти будет в точности таким, каким оно было в базе данных

на момент начала чтения. Однако для обеспечения целостности чтения, система может открывать неявные транзакции, условия возникновения которых описано далее.

Наряду с этим разработчик может использовать работу с транзакциями в явном виде. Для этого используются процедуры глобального контекста `НачатьТранзакцию()`, `ЗафиксироватьТранзакцию()` и `ОтменитьТранзакцию()`.

9.2.2. Использование явного вызова транзакций

Метод `НачатьТранзакцию()` позволяет открыть транзакцию. После этого все изменения информации базы данных, выполняемые последующими операторами, могут быть либо целиком приняты, либо целиком отвергнуты.

Для принятия всех выполненных изменений используется метод `ЗафиксироватьТранзакцию()`. Для того чтобы отменить все изменения, выполнявшиеся в открытой транзакции, используется метод `ОтменитьТранзакцию()`. Если количество вызовов метода `НачатьТранзакцию()` превышает количество вызовов методов `ЗафиксироватьТранзакцию()` или `ОтменитьТранзакцию()`, то система выполнит неявный вызов метода `ОтменитьТранзакцию()` в следующих случаях:

- при окончании выполнения встроенного языка (обработчик события, внешнее соединение, automation-сервер);
- при передаче управления с сервера на клиента.

Если количество вызовов методов `ЗафиксироватьТранзакцию()` или `ОтменитьТранзакцию()` превышает количество вызовов метода `НачатьТранзакцию()`, то при выполнении лишнего вызова метода `ЗафиксироватьТранзакцию()` или `ОтменитьТранзакцию()` будет порождено исключение.

Таким образом, схема работы с транзакцией в общем виде может выглядеть следующим образом:

Копировать в буфер обмена

```
Попытка
    НачатьТранзакцию();
    // Последовательность операторов
    …
    ЗафиксироватьТранзакцию();
Исключение
    ОтменитьТранзакцию();
КонецПопытки;
```

При использовании такой схемы следует помнить о том, что не все ошибки, возникающие при работе с базой данных, обрабатываются системой одинаково.

В общем случае все ошибки базы данных можно разделить на две категории:

- невосстановимые,
- восстановимые.

Невосстановимые ошибки - это ошибки, при возникновении которых нормальное функционирование системы «1С:Предприятие» может быть нарушено, например, могут быть испорчены данные. При возникновении невосстановимой ошибки выполнение системы «1С:Предприятие» прекращается в любом случае.

Если невосстановимая ошибка произошла в процессе выполнения транзакции, то все изменения, сделанные в рамках этой транзакции, отменяются системой.

Восстановимые ошибки - это ошибки, не вызывающие серьезных нарушений в работе системы «1С:Предприятие». В случае возникновения восстановимой ошибки дальнейшая работа системы может быть продолжена. При этом, естественно, сама операция, вызвавшая ошибку, прекращается, и вызывается исключение, которое может быть перехвачено и обработано конструкцией `Попытка … Исключение … КонецПопытки`.

Если восстановимая ошибка произошла в процессе выполнения транзакции, то система автоматически не выполняет отмену транзакции, предоставляя разработчику возможность самостоятельно обработать сложившуюся ситуацию.

В зависимости от характера произошедшей ошибки возможны различные сценарии обработки этой ситуации.

Если произошедшая ошибка не связана с базой данных, то возможно продолжение транзакции и дальнейшей работы модуля. Если разработчик считает это необходимым, он может отменить транзакцию или, наоборот, продолжить выполнение транзакции, если произошедшая ошибка не нарушает атомарность транзакции.

Если же исключительная ситуация была вызвана ошибкой базы данных, то система фиксирует факт возникновения ошибки в этой транзакции, и дальнейшее продолжение транзакции или ее фиксация становятся невозможны. Единственная операция с базой данных, которую разработчик может произвести в данной ситуации, - это отмена транзакции. После этого он может осуществить попытку выполнения этой транзакции еще раз.

Например, фрагмент кода, реализующий этот подход при записи некоторых данных в базу данных, может выглядеть следующим образом.

Копировать в буфер обмена

```
// Признак окончания попыток выполнения записи
Записано = Ложь;
// Попытки записи выполняются в цикле
Пока Не Записано Цикл
    Попытка
        НачатьТранзакцию();
        Данные.Записать();
        ЗафиксироватьТранзакцию();
        // В случае фиксации транзакции прекратить попытки записи
        Записано = Истина;
    Исключение
        // В случае неудачи отменить текущую транзакцию и
        // следующую попытку начать с новой транзакции
        ОтменитьТранзакцию();
    КонецПопытки;
КонецЦикла;
```

9.2.3. Вложенный вызов транзакций

В рамках уже выполняемой транзакции можно обращаться к процедурам `НачатьТранзакцию()`, `ЗафиксироватьТранзакцию()` и `ОтменитьТранзакцию()`. Например, может использоваться следующая схема вызовов:

Копировать в буфер обмена

```
НачатьТранзакцию();
…
    // Вложенный вызов транзакции
    НачатьТранзакцию();
    …
    ЗафиксироватьТранзакцию();
…
    // Вложенный вызов транзакции
    НачатьТранзакцию();
    …
    ЗафиксироватьТранзакцию();
…
ЗафиксироватьТранзакцию();
```

Однако подобное обращение не означает начала новой транзакции в рамках уже выполняющейся.

ВНИМАНИЕ! Система «1С:Предприятие» не поддерживает вложенных транзакций.

Это означает, что всегда действует только транзакция самого верхнего уровня. Все транзакции, вызванные внутри уже открытой транзакции, фактически относятся к той же транзакции, а не образуют вложенную транзакцию. Таким образом, отмена изменений, выполняемая во вложенной транзакции, будет приводить в конечном счете не к отмене изменений самой вложенной транзакции, а к отмене всех изменений транзакции верхнего уровня. В то же время фиксация изменений, выполненная во вложенной транзакции, игнорируется.

фиксации изменения, выполненные во вложенной транзакции, игнорируются.

9.2.4. Влияние транзакций на работу программных объектов

В общем случае программные объекты, используемые системой «1С:Предприятие», абсолютно «прозрачны» для транзакций базы данных. Иначе говоря, транзакции базы данных могут вызываться при выполнении различных методов программных объектов, однако, например, действия, выполняемые базой данных при откате транзакции, в общем случае никак не влияют на соответствующие программные объекты.

Из этого следует, что при отмене транзакций базы данных разработчик (если в этом есть необходимость) должен самостоятельно обеспечивать адекватное изменение данных соответствующих программных объектов. Это можно выполнять путем повторного чтения всех данных объекта или путем изменения некоторых реквизитов программного объекта (если, например, это необходимо для отображения в интерфейсе).

В этом правиле есть исключения. В силу значительной прикладной специфики программных объектов системы «1С:Предприятие» в некоторых случаях откат изменений, выполненных в базе данных, все же может влиять на значения свойств соответствующих программных объектов. Это происходит в следующих случаях:

- при отмене транзакции признак проведения документа восстанавливает значение, которое было до начала транзакции;
- если объект был создан и записан в транзакции, то при откате транзакции очищается значение ссылки;
- если объект создавался вне транзакции и при записи его в транзакции использовался код/номер, сгенерированный автоматически, то при отмене транзакции код/номер очищается.

9.3. Механизм управляемых блокировок

9.3.1. Общая информация

В идеальном случае в любой СУБД транзакции должны обеспечивать изоляцию изменений, выполняемых в базе данных. Иными словами, несколько транзакций, выполняющих изменение данных, не должны мешать друг другу.

Самым простым способом решения этой проблемы является последовательное выполнение транзакций. Следующая транзакция выполняется после того, как закончилась предыдущая.

Однако в реальной ситуации, при многопользовательской работе, такой подход приводит к резкому снижению производительности системы. Поэтому на практике используются механизмы, позволяющие выполнять несколько транзакций одновременно.

Для того чтобы одновременное выполнение транзакций стало возможным, используется несколько уровней изоляции транзакций. На самом низком уровне изоляции транзакции могут сильно влиять друг на друга. На самом высоком уровне они полностью изолированы.

Таким образом, за большую изоляцию транзакций приходится платить большими накладными расходами и замедлением работы системы.

Возможность изоляции одних транзакций от других обычно реализуется благодаря блокировкам, накладываемым на используемые ими данные. В зависимости от уровня изоляции накладываются различные типы блокировок на различные объекты базы данных, на различное время.

С точки зрения системы «1С:Предприятие» работа с данными может выполняться в одном из двух режимов:

- в транзакции,
- вне транзакции.

Режим работы с данными **вне транзакции** допускает только операции **чтения данных**. Этот режим введен для того, чтобы обеспечить максимальную скорость и параллельность чтения данных. Поэтому любая операция чтения данных, выполняемая вне транзакции, считается

данных. Поэтому любая операция чтения данных, выполняемая вне транзакции, считается **безответственной**. Это означает, что такая операция чтения может вернуть устаревшие данные или даже незафиксированные изменения, произведенные другой транзакцией, т. е. чтение выполняется «не глядя» на блокировки данных, расставленные другими транзакциями.

Режим работы с данными в транзакции допускает любые операции чтения и модификации данных.

При этом должны соблюдаться следующие правила:

- Чтение данных должно быть воспроизводимым, т. е. любая последующая операция чтения данных с тем же условием выборки должна возвращать такой же результат.
- Результат чтения должен содержать наиболее актуальные данные. Это означает, что никакая другая транзакция не может изменить данные, считанные в данной транзакции, до тех пор, пока данная транзакция не будет завершена.
- Результат чтения не должен содержать незафиксированные изменения данных базы данных.

9.3.2. Управляемые блокировки

Система «1С:Предприятие» позволяет использовать два режима работы с базой данных в рамках транзакции: режим автоматических блокировок и режим управляемых блокировок.

Принципиальное отличие этих режимов заключается в следующем. Режим автоматических блокировок не требует от разработчика каких-либо действий по управлению блокировками в транзакции для того, чтобы обеспечивались перечисленные выше правила работы с данными в транзакции. Эти правила обеспечиваются платформой системы «1С:Предприятие» за счет использования определенных уровней изоляции транзакций в той или иной СУБД. Такой режим работы является наиболее простым для разработчика, однако в некоторых случаях (например, при интенсивной одновременной работе большого количества пользователей) используемый уровень изоляции транзакций в СУБД не может обеспечить достаточной параллельности работы, что проявляется в виде большого количества конфликтов блокировок при работе пользователей.

При работе в режиме управляемых блокировок система «1С:Предприятие» использует гораздо более низкий уровень изоляции транзакций в СУБД, что позволяет значительно повысить параллельность работы пользователей прикладного решения. Однако, в отличие от режима автоматических блокировок, данный уровень изоляции транзакций уже не может сам по себе обеспечить выполнение всех правил работы с данными в транзакции (в частности, не обеспечивается воспроизводимость чтения данных в транзакции). Поэтому при работе в управляемом режиме от разработчика требуется самостоятельно управлять блокировками, устанавливаемыми в транзакции.

В сводном виде отличия при работе в режиме автоматических блокировок и в режиме управляемых блокировок приведены в следующей таблице:

	Вид блокировки	**Уровень изоляции транзакций**
Автоматические блокировки		
Файловая БД	Таблиц	Serializable
MS SQL Server	Записей	Repetable Read или Serializable
IBM Db2	Записей	Repetable Read или Serializable
PostgreSQL	Таблиц	Serializable
Oracle Database	Таблиц	Serializable
Управляемые блокировки		
Файловая БД	Таблиц	Serializable
MS SQL Server 2000	Записей	Read Commited
MS SQL Server 2005	Записей	Read Commited Snapshot

MS SQL Server 2005 и выше	Записей	Read Commited Snapshot
IBM Db2	Записей	Read Commited
PostgreSQL	Записей	Read Commited

	Вид блокировки	**Уровень изоляции транзакций**
Oracle Database	Записей	Read Commited

> **ПРИМЕЧАНИЕ.** Рекомендуется разработку прикладных решений вести в режиме управляемых блокировок с целью максимального использования возможностей СУБД. Использование прикладного решения в автоматическом режиме используется для совместимости с предыдущими версиями прикладных решений и не рекомендуется для реальной эксплуатации.

Поведение системы при безответственном чтении (например, при работе динамических списков, выполнении отчетов) также отличается в различных режимах блокировок. При работе с объектами, не имеющими табличных частей, отличия безответственного чтения в разных режимах работы блокировок (автоматическом или управляемом) приведены в следующей таблице:

	Чтение вне транзакции
Автоматические блокировки	
Файловая БД	«Грязное» чтение
MS SQL Server	«Грязное» чтение
IBM Db2	«Грязное» чтение
PostgreSQL	Согласованное чтение
Oracle Database	Согласованное чтение
Управляемые блокировки	
Файловая БД	«Грязное» чтение
MS SQL Server 2000	«Грязное» чтение
MS SQL Server 2005 и выше	Согласованное чтение
IBM Db2 до версии 9.7	«Грязное» чтение
IBM Db2 версии 9.7 и выше	Согласованное чтение
PostgreSQL	Согласованное чтение
Oracle Database	Согласованное чтение

Безответственное чтение объектов с табличными частями и наборов записей всегда является согласованным. Для этого платформа может определить неявную транзакцию и установить неявные управляемые блокировки:

	Неявная транзакция
Автоматические блокировки	
Файловая БД	Да
MS SQL Server	Да
IBM Db2	Да

PostgreSQL	Нет
Oracle Database	Нет
Управляемые блокировки	
Файловая БД	Да

	Неявная транзакция
MS SQL Server 2000	Да
MS SQL Server 2005 и выше	Да - в режиме совместимости с версией 8.2.16 и младше Нет - в режиме совместимости с версией 8.2.17 и старше
IBM Db2	Да
PostgreSQL	Нет
Oracle Database	Нет

При чтении объекта с табличными частями без неявного определения транзакции, в режиме управляемых блокировок, выполняется двойное чтение версии объекта (поле объекта `ВерсияДанных`) в начале и в конце чтения. Если версии различаются, то выдается ошибка транзакционной блокировки.

Наборы записей регистров сведений всегда считываются в неявной транзакции, при этом в режиме управляемых блокировок выполняется установка разделяемой блокировки по значению регистратора, а для независимого регистра сведений - по значениям основного отбора.

9.3.3. Установка режима блокировок в конфигурации

Конфигурация имеет свойство Режим управления блокировкой данных. Каждый прикладной объект конфигурации также имеет свойство Режим управления блокировкой данных.

Режим управления блокировкой данных для всей конфигурации в целом может быть установлен в значения Автоматический, Управляемый (установлено по умолчанию для новой конфигурации) и Автоматический и управляемый. Значения Автоматический и Управляемый означают, что соответствующий режим блокировки будет использоваться для всех объектов конфигурации, независимо от значений, установленных для каждого из объектов. Значение Автоматический и управляемый означает, что для конкретного объекта конфигурации будет использован тот режим, который указан в его свойстве Режим управления блокировкой данных: Автоматический или Управляемый.

Следует отметить, что режим управления блокировкой данных, указанный для объекта метаданных, устанавливается для тех транзакций, которые инициируются системой «1С:Предприятие» при работе с данными этого объекта (например, при модификации данных объекта).

Если же, например, операция записи объекта выполняется в транзакции, инициированной разработчиком (метод `НачатьТранзакцию()`), то режим управления блокировкой данных будет определяться значением параметра `РежимБлокировок` метода `НачатьТранзакцию()`, а не значением свойства объекта метаданных Режим управления блокировкой данных.

По умолчанию параметр `РежимБлокировок` имеет значение `РежимУправленияБлокировкойДанных.Автоматический`, поэтому для того, чтобы в явной транзакции использовать режим управляемых блокировок, следует указывать значение этого параметра `РежимУправленияБлокировкойДанных.Управляемый`.

9.3.4. Работа с управляемыми блокировками средствами встроенного языка

Для управления блокировками в транзакции предназначен объект встроенного языка `БлокировкаДанных`. Экземпляр этого объекта может быть создан с помощью конструктора и позволяет описать необходимые пространства блокировок и режимы блокировок. Для установки

позволяет описать необходимые пространства блокировок и режимы блокировок. Для установки всех созданных блокировок используется метод Заблокировать() объекта БлокировкаДанных. Если этот метод выполняется в транзакции (явной или неявной), блокировки устанавливаются и при окончании транзакции будут сняты автоматически. Если метод Заблокировать() выполняется вне транзакции или в автоматическом режиме управления блокировками в рамках транзакции, то будет сгенерировано исключение.

Объект БлокировкаДанных представляет собой коллекцию элементов блокировки данных, каждый из которых описывает блокировки одного пространства блокировок. Пространства блокировок определены в платформе системы «1С:Предприятие» и соответствуют структуре прикладных объектов конфигурации. Для каждого пространства блокировок в платформе определены имена полей, значения которых могут анализироваться при установке тех или иных блокировок.

Допустимы следующие имена пространств блокировок и имена полей пространств блокировок:

Имя пространства блокировки	**Имя поля пространства блокировки**
Справочник.<имя>	Ссылка; <имя поля>
Документ.<имя>	Ссылка; <имя поля>
ПланОбмена.<имя>	Ссылка; <имя поля>
ПланСчетов.<имя>	Ссылка; <имя поля>
БизнесПроцесс.<имя>	Ссылка; <имя поля>
Задача.<имя>	Ссылка; <имя поля>
ПланВидовРасчета.<имя>	Ссылка; <имя поля>
ПланВидовХарактеристик.<имя>	Ссылка; <имя поля>
РегистрСведений.<имя>.НаборЗаписей - только для регистра сведений, подчиненного регистратору	Регистратор
РегистрСведений.<имя>	Период - если есть; <имя измерения>
РегистрНакопления.<имя>.НаборЗаписей	Регистратор
РегистрНакопления.<имя>	Период; <имя измерения>
РегистрБухгалтерии.<имя>.НаборЗаписей	Регистратор
РегистрБухгалтерии.<имя>	Период; <имя измерения>;

	<вид движения> - значение системного перечисления ВидДвиженияБухгалтерии; Счет; Субконто<N>; <вид субконто>.

Имя пространства блокировки	**Имя поля пространства блокировки**
РегистрРасчета.<имя>.НаборЗаписей	Регистратор
РегистрРасчета.<имя>	ПериодРегистрации; ПериодДействия; <имя измерения>
Перерасчет.<имя>.НаборЗаписей	ОбъектПерерасчета
Последовательность.<имя>.НаборЗаписей	Регистратор
Последовательность.<имя>	<имя измерения>
Константа.<имя>	
ВнешнийИсточникДанных.<имя источника>.Таблица.<имя таблицы>	Ссылка; <имя поля>

Где <имя поля> - имя поля, по которому может быть установлена управляемая блокировка. Перечень допустимых полей (по которым может быть установлена управляемая блокировка) задается в свойстве Поля блокировки данных для следующих объектов:

- справочники,
- документы,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,
- планы обмена,
- бизнес-процессы,
- задачи;
- таблицы внешних источников данных.

Запрещено устанавливать блокировки по реквизитам (и общим реквизитам) следующих типов: строка неограниченной длины, хранилище значения, тип значения плана видов характеристик, составной тип, включающий один из вышеперечисленных типов. Запрет относится к следующим объектам: справочники, документы, планы видов характеристик, планы видов расчета, планы счетов, бизнес-процессы, задачи, планы обмена. Также запрещено устанавливать блокировки по полям Предопределенный и ИмяПредопределенныхДанных в справочниках, планах видов характеристик, планах видов расчета, планах счетов.

При установке управляемой транзакционной блокировки на пространство РегистрБухгалтерии.<Имя>, если значение субконто задано ссылкой на характеристику, то в плане счетов эта характеристика должна быть одним из субконто этого счета. Если соответствующее субконто у счета не задано, то будет выдана ошибка.

Для каждого пространства блокировки может быть задано произвольное количество условий на

Для каждого пространства блокировки может быть задано произвольное количество условий на поля, по которым будут определяться записи, подлежащие блокировке. Условия задаются на равенство значения поля указанному значению или на вхождение значения поля в указанный диапазон. Условия могут быть заданы двумя способами:

- с помощью явного указания имени поля и значения (метод `УстановитьЗначение()` объекта `ЭлементБлокировкиДанных`);
- с помощью указания источника данных, содержащего необходимые значения (свойство `ИсточникДанных` объекта `ЭлементБлокировкиДанных`).

При явном указании значения поля в параметры метода `УстановитьЗначение()` передается имя поля и значение:

Копировать в буфер обмена

```
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("Справочник.Магазин");
ЭлементБлокировки.УстановитьЗначение("Код", 100);
ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;
Блокировка.Заблокировать();
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("РегистрНакопления.ТоварыНаСкладах");
ЭлементБлокировки.УстановитьЗначение("Качество",
Справочники.Качество.НайтиПоКоду("1"));
```

Следует иметь в виду, что одну и ту же запись регистра можно заблокировать дважды: первый раз блокируя саму запись, а второй раз - блокируя набор, в который эта запись входит.

При удалении или изменении объекта, для которого доступна блокировка по полям, выполняется одна блокировка, включающая в себя блокировку по полю `Ссылка` и по всем полям, указанным в свойстве Поля блокировки данных. При выполнении изменения объект блокируется как по «старым» значениям полей (которые были до начала записи), так и по «новым» значениям (которые находятся в записываемом объекте).

В качестве значения может быть передан также объект встроенного языка `Диапазон`, который создается с помощью конструктора и позволяет задать верхнюю и нижнюю границы диапазона (границы диапазона включаются в диапазон).

При использовании источника данных устанавливается значение свойства `ИсточникДанных`, а затем с помощью метода `ИспользоватьИзИсточникаДанных()` задается соответствие полей области блокировки полям источника данных:

Копировать в буфер обмена

```
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка. Добавить("РегистрНакопления.ТоварыНаСкладах");
ЭлементБлокировки.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Номенклатура", "Номенклатура");
ЭлементБлокировки.ИспользоватьИзИсточникаДанных("Склад", "Склад");
```

В качестве источника данных может выступать:

- результат запроса,
- табличная часть,
- набор записей,
- таблица значений.

Соответственно, при установке соответствия полей именами полей источника будут являться:

- имена колонок результата запроса,
- имена реквизитов табличной части,
- имена измерений,
- имена колонок таблицы значений.

Объект Диапазон также может являться значением поля источника данных.

Для каждого элемента блокировки может быть задан один из двух режимов блокировки:

- разделяемый,
- исключительный.

Режим блокировки задается с помощью свойства Режим объекта ЭлементБлокировкиДанных.

Таблица совместимости управляемых блокировок выглядит следующим образом:

	Разделяемая	Исключительная
Разделяемая	+	-
Исключительная	-	-

Разделяемый режим блокировки подразумевает, что заблокированные данные не могут быть изменены другой транзакцией до окончания текущей транзакции.

Исключительный режим блокировки подразумевает, что заблокированные данные не могут быть изменены другой транзакцией до окончания текущей транзакции, а также не могут быть прочитаны другой транзакцией, устанавливающей разделяемую блокировку на эти данные.

Говоря о совместимости действий, выполняемых в транзакции, следует учитывать, что помимо явных управляемых блокировок, устанавливаемых пользователем, система «1С:Предприятие» устанавливает также неявные управляемые исключительные блокировки при записи данных в транзакции. При чтении данных объекта из базы данных (получении объекта и обращении к ссылке) не выполняется транзакционная блокировка объекта. Если блокировка необходима, то ее нужно устанавливать средствами языка до обращения к объекту.

Таким образом, таблица совместимости действий, выполняемых в транзакции при использовании режима управляемых блокировок, выглядит следующим образом.

Блоки–ровка	**Реж–им**	**ББ**		**РБ**		**ИБ**	
R	W	R	W	R	W		
ББ	R	+	+	+	+	+	+
	W	+	-	-	-	-	-
РБ	R	+	-	+	-	-	-
	W	+	-	-	-	-	-
ИБ	R	+	-	-	-	-	-
	W	+	-	-	-	-	-

Где:

- R - чтение,
- W - запись,
- ББ - без блокировки,
- РБ - разделяемая блокировка,
- ИБ - исключительная блокировка.

9.3.5. Особенности работы в режиме «Автоматический и управляемый»

Режим управления блокировками Автоматический и управляемый предназначен для решения проблем, возникающих при параллельной работе с отдельными объектами конфигурации.

Например, существует большая конфигурация, которую сложно в один момент перевести в режим управляемых блокировок, однако есть один или два документа, являющиеся «камнем преткновения» при одновременной работе с ними большого количества пользователей. В этом случае вся конфигурация может быть переведена в режим Автоматический и управляемый, а затем сам документ и затрагиваемые при его записи объекты конфигурации - в режим Управляемый. Это позволит настроить работу с данным документом в режиме управляемых блокировок, в то время как основная часть конфигурации будет продолжать функционировать в автоматическом режиме управления блокировками.

При работе в режиме управления блокировками Автоматический и управляемый следует учитывать две особенности:

- Независимо от режима, указанного для данной транзакции, система будет устанавливать соответствующие управляемые блокировки.

- Режим управления блокировками определяется транзакцией самого «верхнего» уровня. Другими словами, если к моменту начала транзакции была начата другая транзакция, то начинаемая транзакция может быть выполнена только в том в режиме, который установлен для уже выполняющейся транзакции.

Рассмотрим перечисленные особенности более подробно.

Первая особенность заключается в том, что даже если для транзакции используется автоматический режим управления блокировками, система установит дополнительно и соответствующие управляемые блокировки при записи данных в этой транзакции. Из этого следует, что транзакции, исполняющиеся в режиме управляемых блокировок, могут конфликтовать с транзакциями, исполняющимися в режиме автоматического управления блокировками.

Вторая особенность заключается в том, что режим управления блокировками, указываемый для объекта метаданных в конфигурации или указываемый при начале транзакции в явном виде (как параметр метода `НачатьТранзакцию()`), является лишь «желаемым» режимом. Фактический режим управления блокировками, в котором будет исполняться транзакция, зависит от того, является ли данный вызов начала транзакции первым, или к этому моменту уже начата другая транзакция в данной сессии системы «1С:Предприятие».

Например, если требуется управлять блокировками при записи наборов записей регистра, при проведении документа, то управляемый режим блокировок должен быть установлен как для самого регистра, так и для документа, поскольку запись наборов записей регистра будет выполняться в транзакции, открытой при записи документа.

Возможны четыре различных сочетания режимов управления блокировками в транзакции, которые представлены в следующей таблице:

Режим существующей транзакции	**Режим начинаемой транзакции**	**Результат**
Автоматический	Автоматический	Начинаемая транзакция будет выполнена в автоматическом режиме
Автоматический	Управляемый	Начинаемая транзакция будет выполнена в автоматическом режиме
Управляемый	Автоматический	Будет вызвана исключительная ситуация
Управляемый	Управляемый	Начинаемая транзакция будет выполнена в управляемом режиме

9.3.6. Особенности работы с большим количеством управляемых блокировок

Следует помнить, что если на одно пространство накладывается более 100 000 блокировок, то может произойти эскалация блокировки. При эскалации блокировки блокируется все пространство. При использовании независимых разделителей эскалация происходит для одного

набора значений разделителей:

- Блокируется все пространство только в рамках значений разделителей;
- На сеансы с другими значениями разделителей эскалация влияния не оказывает.

При работе с управляемыми блокировками следует помнить о том, что система может поглощать устанавливаемые блокировки. Блокировка, в которой указаны значения не всех измерений пространств блокировки, поглотит блокировку, в которой указаны значения для большего количества измерений пространств блокировки, при условии, что совпадают значения по совпадающим измерениям пространствам блокировок. Поглощаться могут как блокировки, устанавливаемые из встроенного языка с помощью метода `БлокировкаДанных.Заблокировать()`, так и блокировки, устанавливаемые механизмами платформы.

Так, например, если регистр накопления имеет измерения `Склад` и `Номенклатура`, то блокировка, у которой указано только измерение `Склад`, поглотит любые блокировки, у которых указаны оба измерения (`Склад` и `Номенклатура`).

В случае если используется транзакция с больш**и**м количеством управляемых блокировок, которые отличаются, например, значением одного измерения, рекомендуется вместо большого количества блокировок установить одну управляемую блокировку, для которой не будет установлено значение отличающегося измерения.

Если при эскалации блокировки возникает конфликт с уже наложенными блокировками, то эскалация не выполняется, а производится попытка установить запрошенную блокировку. В этом случае возможна ситуация, когда в системе будет существовать более 100 000 блокировок на одно пространство.

9.3.7. Модификация конфигураций при переходе к режиму управляемых блокировок

9.3.7.1. Общая информация

При переходе к частичной или полной работе в режиме управляемых блокировок прикладное решение требует доработки. Доработка заключается в том, что необходимо выявить участки кода, которые требуют наличия управляемых блокировок, и затем в этих участках кода установить требуемый режим управляемой блокировки данных. Это можно осуществить двумя способами:

- с помощью объекта `БлокировкаДанных`;
- с помощью свойства `БлокироватьДляИзменения` у наборов записей регистров накопления и регистров бухгалтерии. Необходимая управляемая блокировка будет автоматически установлена платформой в том случае, если записывается набор записей регистра, у которого данное свойство имеет значение `Истина`.

9.3.7.2. Определение участков кода, требующих доработки

Управляемые блокировки необходимо устанавливать в тех участках кода, которые удовлетворяют одному из следующих условий:

- Считываются данные, которые в дальнейшем должны быть изменены.
- Считывается некоторая согласованная совокупность данных (содержащихся в нескольких объектах), и согласованность считанных данных необходимо поддержать. В этом случае данные из этой совокупности должны быть заблокированы либо единовременно перед считыванием, либо по мере считывания отдельных элементов. Если некоторый элемент данных считывается однократно, не связан логически с другими элементами данных и при этом не будет изменен в той же транзакции, то его можно не блокировать.
- Важно обеспечить неизменность считываемых данных до конца транзакции. Например, некоторые данные, считываемые в транзакции, могут быть прочитаны еще раз, и важно обеспечить их неизменность.

Типичным примером таких операций может служить запрос к остаткам номенклатуры в модуле проведения документа.

проведения документа.

9.3.7.3. Выбор режима управляемой блокировки

Для операций, описанных выше, режим блокировки следует устанавливать исходя из следующих соображений:

- **Исключительный** режим блокировки следует устанавливать для тех данных, которые должны быть изменены в рамках этой же транзакции. Это позволит предотвратить конфликты блокировок.

- **Разделяемый** режим блокировки следует устанавливать для тех данных, которые только считываются и изменять которые не предполагается, но заблокировать от изменений все-таки требуется.

9.3.7.4. Примеры доработки конфигурации

Далее приведем пример установки управляемых блокировок в модуле набора записей регистра накопления `ТоварыНаСкладах`.

Прежде всего, следует создать управляемую блокировку:

```
Копировать в буфер обмена
Блокировка = Новый БлокировкаДанных;
```

Затем следует проанализировать текст запроса, формируемого в модуле, и создать необходимые блокировки.

Исключительная блокировка на регистр «ТоварыНаСкладах»

В ходе выполнения модуля набора записей формируется запрос, содержащий следующий фрагмент:

```
Копировать в буфер обмена
// …
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.Товары
        ГДЕ
        Ссылка = &ДокументСсылка))
// …
```

В данном случае следует установить следующие блокировки:

```
Копировать в буфер обмена
БлокировкаТоварыНаСкладах1 = Блокировка.
            Добавить("РегистрНакопления.ТоварыНаСкладах");
БлокировкаТоварыНаСкладах1.Режим = РежимБлокировкиДанных.Исключительный;
БлокировкаТоварыНаСкладах1.ИсточникДанных = ДокументОбъект.Товары;
БлокировкаТоварыНаСкладах1.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");
БлокировкаТоварыНаСкладах1.ИспользоватьИзИсточникаДанных("ХарактеристикаНоменклатуры",
"ХарактеристикаНоменклатуры");
БлокировкаТоварыНаСкладах1.ИспользоватьИзИсточникаДанных("Склад", "Склад");
```

Разделяемая блокировка на регистр «ТоварыВРезервеНаСкладах»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

```
Копировать в буфер обмена
// …
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.Товары
```

```
        Документ.РеализацияТоваровУслуг.Товары
        ГДЕ
        Ссылка = &ДокументСсылка))
// …
```

В данном случае нужно установить следующие блокировки:

Копировать в буфер обмена

```
БлокировкаТоварыВРезервеНаСкладах1 =
Блокировка.Добавить("РегистрНакопления.ТоварыВРезервеНаСкладах");
БлокировкаТоварыВРезервеНаСкладах1.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыВРезервеНаСкладах1.ИсточникДанных = ДокументОбъект.Товары;
БлокировкаТоварыВРезервеНаСкладах1.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");
БлокировкаТоварыВРезервеНаСкладах1.ИспользоватьИзИсточникаДанных(


"ХарактеристикаНоменклатуры", "ХарактеристикаНоменклатуры");
БлокировкаТоварыВРезервеНаСкладах1.ИспользоватьИзИсточникаДанных("Склад", "Склад");
```

Разделяемая блокировка на регистр «ТоварыКПередачеСоСкладов»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

Копировать в буфер обмена

```
// …
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыКПередачеСоСкладов.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.Товары
        ГДЕ
        Ссылка = &ДокументСсылка))
// …
```

В данном случае нужно установить следующие блокировки:

Копировать в буфер обмена

```
БлокировкаТоварыКПередачеСоСкладов1 =
Блокировка.Добавить("РегистрНакопления.ТоварыКПередачеСоСкладов");
БлокировкаТоварыКПередачеСоСкладов1.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыКПередачеСоСкладов1.ИсточникДанных = ДокументОбъект.Товары;
БлокировкаТоварыКПередачеСоСкладов1.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");
БлокировкаТоварыКПередачеСоСкладов1.ИспользоватьИзИсточникаДанных("ХарактеристикаНомен
клатуры", "ХарактеристикаНоменклатуры");
БлокировкаТоварыКПередачеСоСкладов1.ИспользоватьИзИсточникаДанных("Склад", "Склад");
```

Блокировки по табличной части «ВозвратнаяТара». Исключительная блокировка на регистр «ТоварыНаСкладах»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

Копировать в буфер обмена

```
// …
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара
        ГДЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Ссылка =
            &ДокументСсылка)
        И Качество = &Новый)
// …
```

В данном случае нужно установить следующие блокировки:

Копировать в буфер обмена

```
БлокировкаТоварыНаСкладах2 = Блокировка.Добавить("РегистрНакопления.ТоварыНаСкладах");
БлокировкаТоварыНаСкладах2.Режим = РежимБлокировкиДанных.Исключительный;
БлокировкаТоварыНаСкладах2.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
БлокировкаТоварыНаСкладах2.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");
БлокировкаТоварыНаСкладах2.ИспользоватьИзИсточникаДанных("Склад", "Склад");
БлокировкаТоварыНаСкладах2.УстановитьЗначение("Качество",
Справочники.Качество.НайтиПоКоду("1"));
```

Блокировки по табличной части «ВозвратнаяТара». Разделяемая блокировка на регистр «ТоварыВРезервеНаСкладах»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

Копировать в буфер обмена

```
// …
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара
        ГДЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Ссылка =
            &ДокументСсылка)) КАК Резервы
// …
```

В данном случае нужно установить следующие блокировки:

Копировать в буфер обмена

```
БлокировкаТоварыВРезервеНаСкладах2 =
Блокировка.Добавить("РегистрНакопления.ТоварыВРезервеНаСкладах");
БлокировкаТоварыВРезервеНаСкладах2.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыВРезервеНаСкладах2.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
БлокировкаТоварыВРезервеНаСкладах2.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");
БлокировкаТоварыВРезервеНаСкладах2.ИспользоватьИзИсточникаДанных("Склад", "Склад");
```

Блокировки по табличной части «ВозвратнаяТара». Разделяемая блокировка на регистр «ТоварыКПередачеСоСкладов»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

Копировать в буфер обмена

```
// …
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыКПередачеСоСкладов.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара
        ГДЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Ссылка =
            &ДокументСсылка)) КАК ТоварыКПередаче
// …
```

В данном случае нужно установить следующие блокировки:

Копировать в буфер обмена

```
БлокировкаТоварыКПередачеСоСкладов2 =
Блокировка.Добавить("РегистрНакопления.ТоварыКПередачеСоСкладов");
БлокировкаТоварыКПередачеСоСкладов2.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыКПередачеСоСкладов2.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
БлокировкаТоварыКПередачеСоСкладов2.ИспользоватьИзИсточникаДанных("Номенклатура",
"Номенклатура");
БлокировкаТоварыКПередачеСоСкладов2.ИспользоватьИзИсточникаДанных("Склад", "Склад");
```

После того как все необходимые блокировки созданы, следует заблокировать перечисленные данные:

```
Копировать в буфер обмена
Блокировка.Заблокировать();
// … далее следует прежний текст модуля
```

9.4. Монопольный режим

Монопольный режим - специальный режим доступа к базе данных, при использовании которого работать с базой данных можно, используя только один сеанс. Монопольный режим можно использовать в тех случаях, когда требуется выполнить существенные, согласованные, изменения с информационной базой, которые нет необходимости выполнять в рамках транзакции. При этом желательно исключить влияние других сеансов на результаты изменения. Установку монопольного режима требуют некоторые методы встроенного языка (например, метод `УдалитьОбъекты()`), а также работа Конфигуратора.

Для установки или снятия монопольного режима необходимо использовать метод глобального контекста `УстановитьМонопольныйРежим()`. Проверить установку монопольного режима можно с

помощью метода `МонопольныйРежим()`. Невозможно изменить состояние монопольного режима в том случае, если активна транзакция (явная или неявная).

Сеанс может установить монопольный режим, если нет других сеансов работы с этой информационной базой. В монопольном режиме запрещено создание новых сеансов с данной информационной базой за исключением запуска одного фонового задания. Если из сеанса, который установил монопольный режим, будет запущено фоновое задание, то запущенное фоновое задание «отнимет» монопольный режим у родительского сеанса. Во время работы фонового задания родительский сеанс не имеет возможности изменять данные в информационной базе. Монопольный режим «вернется» к родительскому сеансу после того, как запущенное фоновое задание будет завершено.

В случае работы в монопольном режиме не устанавливаются управляемые блокировки (попытки их установки игнорируются).