**Совместный доступ к данным**

Разрешения конфликтов доступа к данным

Михаил Фленов ([www.flenov.info](http://www.flenov.info))

Чем больше количество одновременно работающих с базой данных пользователей, тем больше вероятность встретиться с конфликтом одновременного редактирования одной и той же строки. Что делать серверу, если два пользователя одновременно пытаются обновить одну и ту же запись? Можно принять то изменение, которое пришло позже, но тогда один из пользователей будет видеть у себя некорректные данные. А самое страшное, он будет думать, что все в порядке. Как разрешить подобные проблемы?

Допустим, что два пользователя открыли для редактирования форму с одним и тем же документом. Первый пользователь изменяет важные параметры, цены, количество и нажимает сохранить. Другой пользователь не видит этих изменений, потому что он получил данные раньше. При этом, он изменяет только незначительный параметр, например, дату и тоже нажимает сохранение. Незначительное изменение второго пользователя перезаписывает важные изменения первого.

В многопользовательских приложениях, к программированию можно поступать двумя способами:

1. Кто последний, тот и прав. В этом случае вы просто реализуете логику программы и не заботитесь о том, что два пользователя могут одновременно изменять какие-то данные. Прав будет тот, кто чуть позже нажмет кнопку обновления.

2. Попытаться реализовать одновременный пользователь собственными средствами, с помощью журналов. Если в журнале есть запись, что кто-то открыл документ, но не закрыл, то не разрешать повторное открытие другим пользователям. Может быть, где-то это будет удобно и быстро, но в Oracle реализованы хорошие встроенные средства блокировок, которые работают быстрее, эффективнее и надежнее, поэтому данный метод мы не рекомендуем к использованию.

3. Блокировать записи, которые пользователь собирается изменять средствами базы данных. Заблокированную запись невозможно изменить, поэтому все запросы на редактирование будут отклоняться. Данный подход является более правильным и именно ему посвящена данная статья.

Заблокированные средствами Oracle записи может изменить только тот пользователь, который установил блокировку. Остальные могут только просматривать и не могут выполнять над этими данными операторы UPDATE или DELETE.

**Блокировка**

Блокировка - это механизм базы данных, с помощью которого сервер удерживает определенные ресурсы за определенным пользователем. Остальные пользователи могут только читать заблокированные данные. Oracle достаточно интеллектуален и блокирует данные на максимально необходимом уровне. Если изменению подвергается только одна строка, то только она будет удержана. Не все такие интеллектуальные и другие базы данных могут блокировать данные целыми страницами, а в одной странице может быть несколько строк, и все они становятся недоступными для редактирования. Если вы используете другую базу данных, то сначала познакомьтесь с ее механизмом работы.

Блокировки бывают явными и не явными. Неявные создаются сервером без нашего участия при каждом изменении данных таблицы или структуры и снимаются по завершению выполнения оператора. Такие блокировки существуют только во время выполнения операции модификации данных.

Явные блокировки задаются пользователем и могут быть созданы на этапе выбора данных, и будут существовать, пока пользователь не завершит работу с данными. В данном случае, вы явно указываете, что определенный ресурс должен быть закреплен за вами до тех пор, пока вы его не отпустите.

Не стоит бояться блокировок, потому что в Oracle они никак не сказываются на производительности системы. Они лишь говорят о том, что какие-то данные взяты определенным пользователем для редактирования.

**FOR UPDATE**

Когда мы просто используем оператор SELECT для выборки данных, то сервер выполняет наш запрос без блокирования каких-либо записей. Но если необходима выборка данных непосредственно для редактирования, то мы должны сообщить серверу, о блокировке. Для этого в конец запроса необходимо добавить FOR UPDATE. Например, следующий запрос выбирает все записи из таблицы Users для редактирования.

SELECT \*

FROM Users

FOR UPDATE

Данный запрос ужасен, но он является только примером. Дело в том, что запрос выбирает все записи из таблицы, а значит, все они будут заблокированы для других пользователей. Никогда не поступайте так. Если вам необходимо изменить всю таблицу, то можете сразу выполнять оператор UPDATE в определенной транзакции, выбирать данные тут не имеет смысла. Если хотя бы одна строка окажется закрепленной за каким-то пользователем, то оператор UPDATE не пройдет и блокировка не поможет.

Заблокировать таблицу можно еще с помощью оператора LOCK TABLE, но лучше все же выбирать с помощью запроса SELECT только те данные, которые нужны, и при этом указывать ключевые слова FOR UPDATE.

Чаще всего работа с данными построена по принципу: окно реестра - окно редактирования. Например, у вас есть окно реестра документов, где пользователи могут просматривать счета, накладные и т.д. за определенный период времени. В этом окне происходит только просмотр, поэтому для выборки данных здесь не следует использовать блокировок, иначе это приведет к проблемам при многопользовательской работе. Если один пользователь выберет все документы за месяц, то остальные не смогут открыть данные за тот же период.

Когда пользователь решил отредактировать какой-либо документ, то следует открыть отдельное окно, в котором будет выбран именно этот документ и на него будет установлена блокировка, например:

SELECT \*

FROM Docs

WHERE PrimaryKey=10

FOR UPDATE

В этом примере мы выбираем и блокируем запись из таблицы Docs с первичным ключом равным 10. Блокировка будет поставлена только на одну запись и этот документ больше никто не сможет открыть. Так как в окне реестра документов выполняется запрос SELECT без FOR UPDATE, то он продолжит работать, и остальные пользователи смогут его просматривать и открывать для редактирования другие документы, но только не заблокированные.

**Не ждите**

А что произойдет, если пользователь попытается открыть документ, который уже заблокирован другим пользователем? Ответ прост - запрос зависнет в ожидании освобождения ресурсов. Если в вашей программе не будет предусмотрено возможности прерывания запросов, а блокировка оказалась мертвой, то программа зависнет навечно. Завершить работу можно будет только прерыванием процесса. Самое страшное, если какой-то пользователь открыл окно и ушел на обед. Ресурс оказывается заблокированным надолго, и это мешает работе других пользователей.

Если процесс прерывается аварийно, то все заблокированные этим пользователем ресурсы блокируются намертво. Чтобы их освободить, необходимо подключиться к серверу с правами системного администратора и завершить сессии.

Чтобы сессия не зависла из-за бесконечного ожидания заблокированных данных, я рекомендую добавлять еще опцию NOWAIT:

SELECT \*

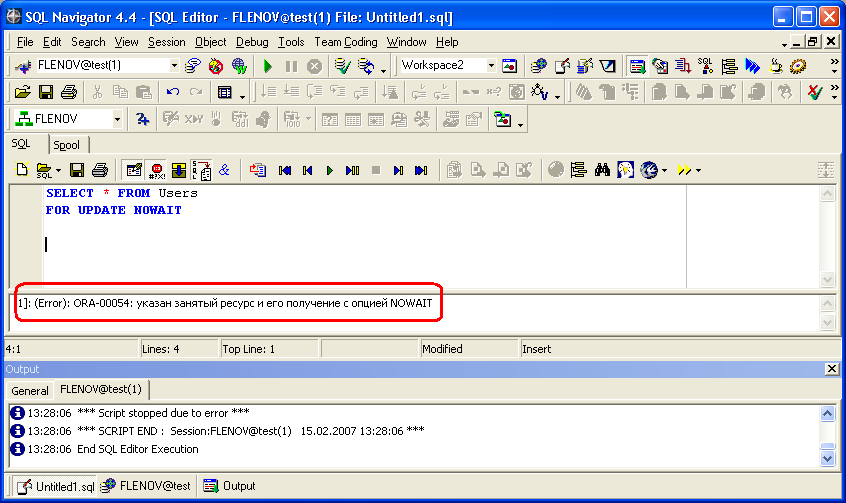
FROM Docs

WHERE PrimaryKey=10

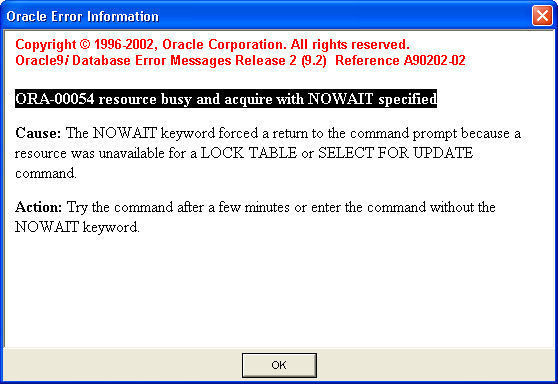
FOR UPDATE NOWAIT

Такой запрос попытается получить данные и установить на них блокировку, но если это невозможно, то ожидания не будет. Сервер просто вернет ошибку с номером ORA-00054:

ORA-00054 Resource busy and acquire with NOWAIT specified



*Попытка получить заблокированные данные*



*Описание ошибки ORA-00054*

Теперь, когда мы увидели, что данные заблокированы, можно показать пользователю сообщение о том, что кто-то уже редактирует таблицу и открыть карточку документа, но только в режиме редактирования. Только для этого нужно выполнить снова запрос SELECT без попытки блокирования ресурсов.

**Пример**

Давайте посмотрим, как реализовать возможность открытия карточки редактирования с использованием блокировок на Delphi. Допустим, что у нас есть форма TSomeDocument для редактирования и данные выбираются с помощью компонента TOracleDataSet (назовем его odsDocs) из состава DOA (Direct Oracle Access, прямой доступ к Oracle). В компоненте odsDocs прописан запрос на выборку данных, без каких либо блокировок. По событию OnShow для формы пишем код, показанный в листинге 1.

Давайте разберем содержимое представленного листинга. Сначала сохраняем запрос, который прописан в компоненте и после этого, добавляем к запросу опции FOR UPDATE NOWAIT. Теперь открываем набор данных внутри блока try...except. Если код отработал нормально, то ресурс свободен и уже заблокирован под нами. Нужно только проверить количество записей на 0. А вдруг пока мы работали с выборкой в реестре документов, данный документ уже кто-то удалил?

Если во время открытия набора данных произошла ошибка из-за блокировки, то выполнение программы переходит на блок except. Здесь возвращаем сохраненный запрос в компонент odsDocs, сообщаем пользователю, что данные невозможно открыть для редактирования, и открываем набор данных, но уже без опции FOR UPDATE NOWAIT.

Это достаточно простой, но эффективный способ блокирования документов.

**Блокировки в связанных запросах**

А теперь посмотрим еще один интересный эффект. Допустим, что у нас есть две таблицы Docs и Users. В таблице Docs есть поле UserID, где сохраняется первичный ключ из таблицы Users. Таким образом, каждый документ привязан к определенному пользователю, например, создавшему, ответственному или кому-то еще. Посмотрим, как будет выглядеть запрос на выборку данных для редактирования:

SELECT \*

FROM Docs d, Users u

WHERE d.PrimaryKey=10

AND d.UserID =u.PrimaryKey

FOR UPDATE

В результате блокировка будет установлена не только на выбранный документ под номером 10, но и на запись в таблице Users, которая связана с данным документом. Это очень плохо. Теперь, если кто-то другой попытается открыть на редактирование другой документ, но тоже связанный с этим пользователем, то сервер не даст этого сделать. Все документы пользователя будут заблокированы, а это неправильно. Блокироваться должен только определенный документ, а таблица пользователей не будет редактироваться (из нее только выбирается запись) и ее сервер не должен трогать.

Как сообщить Oracle, что записи в Users блокировать нельзя? Для этого нужно явно указать таблицу, а лучше первичный ключ в этой таблице: FOR UPDATE OF имя поля. После ключевого слова OF указывается поле, по которому сервер узнает, какую запись из связанных таблиц нужно заблокировать. Получается, что наш запрос должен выглядеть следующим образом:

SELECT \*

FROM Docs d, Users u

WHERE d.PrimaryKey=10

AND d.UserID =u.PrimaryKey

FOR UPDATE OF d.PrimaryKey

Вот теперь будет заблокирована только одна запись документа и только из таблицы Docs.

**Продолжительность**

Чтобы пользователи не блокировали данные на слишком продолжительное время, при открытии формы можно запускать таймер и через пять минут спрашивать подтверждения продолжения работы. Если пользователь не подтвердит, то форма должна закрыться и освободить ресурс. Это поможет, если кто-то забывчивый уходит на обед или домой с запущенной программой и открытыми ресурсами. Если у вас многооконная система и предусмотрена возможность открытия сразу множества документов, то пользователь может забывать закрывать окна редактирования и это снова лишние, а главное - неоправданные блокировки.

Не помешало бы средство, которое отключало бы таймер на те случаи, когда пользователь действительно хочет работать с данными долго, но он должен это делать осознано. Нет, в блокировках нет ничего страшного для сервера, производительности и корректности данных, но документ может понадобиться другим.

**Система**

Теперь поговорим о системных представлениях, с помощью которых вы можете управлять и контролировать блокировки. Все блокировки можно получить с помощью представления v$lock:

SELECT \*

FROM v$lock

Результат не очень информативен, потому что содержит какие-то адреса и цифры, да и записей очень много. В поле sid находиться идентификатор сессии, а в поле Type можно увидеть тип блокировки. Когда вызывается SELECT FOR UPDATE, то создается блокировка транзакции, а в поле Type можно будет увидеть TX. Существуют и другие типы, например, блокировки сервера, изменения структуры таблиц и т.д. Более подробно, об этом можно прочитать в документации по Oracle.

Исходя из вышесказанного, более информативным будет следующий запрос:

SELECT s.username, l.\*

FROM v$lock l, v$session s

WHERE l.TYPE = 'TX'

and l.sid=s.sid

Здесь мы связались с представлением v$session, которое возвращает сессии и теперь в результат попадает имя пользователя, который удерживает блокировку.

Из представления v$session можно получить много полезной информации. Просто выполните следующий запрос, чтобы определиться, какие еще поля можно включить в запрос, показанный выше:

SELECT \*

FROM v$session

Пока все хорошо, но по полученным данным мы до сих пор не знаем, кто же именно заблокировал определенную строку. Вот тут я бы порекомендовал создать пользовательскую таблицу журнала из следующих полей:

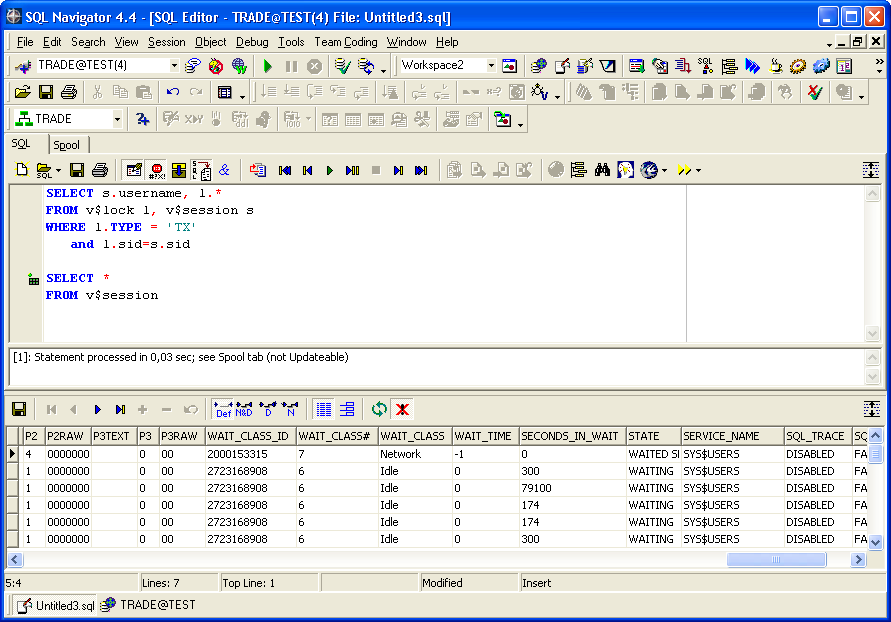
- идентификатор документа;

- идентификатор пользователя;

- дата.

Теперь, после каждого удачного открытия ресурса на редактирование, можно занести запись в эту таблицу с указанием документа, пользователя и даты. В программе, в окне просмотра реестра, можно реализовать функцию просмотра журнала по определенному документу. Теперь, если что-то не открывается на редактирование, то с помощью журнала пользователи сами смогут узнать, кто последний заблокировал запись и не дает работать другим.

Журнал позволит избежать вам множества звонков с вопросами, кто и что заблокировал. Если злополучного пользователя нет, то тогда уже будут обращаться к вам, а вы с помощью таблиц v$lock и v$session сможете увидеть блокировки и снять их.



*Информации о сессиях*

**LOCK TABLE**

Допустим, что нам нужно произвести несколько действий по изменению данных во всей таблицы (или в большинстве ее записей), и все эти изменения невозможно уложить в один единственный запрос UPDATE, да и сама работа с данными отнимет не пять минут. При выполнении одной операции UPDATE блокировать таблицу не имеет смысла, но при больших изменениях просто необходимо. Если после выполнения некоторых действий, кто-то заблокирует хотя бы одну запись, дальнейшие ваши действия будут парализованы. Такой трюк может привести к нарушению целостности данных, поэтому перед большим количеством изменений лучше все же заблокировать всю таблицу.

Для блокировки всей таблицы лучше использовать не SELECT FOR UPDATE, а LOCK TABLE IN EXCLUSIVE MODE. Этот оператор блокирует всю таблицу сразу, а не каждую строку в отдельности. Это намного эффективнее, когда воздействию подвергается вся таблица или большая ее часть.

Заблокировав всю таблицу, вы можете не торопясь модифицировать данные и никто другой не сможет вас прервать от этого интересного занятия.

**Итого**

Блокировки - очень мощное и удобное средство для многопользовательских приложений. Используйте их и вы избавитесь от множества проблем, а приобретете минимальные проблемы, которые легко разрешимы. Главное следовать следующим правилам:

1. Старайтесь блокировать минимально необходимое количество записей в таблице.

2. Не забудьте по закрытию формы сохранить или откатить изменения и закрыть набор данных, чтобы освободить ресурс.

Блокировки гарантируют, что введенные пользователем данные будут сохранены в базе, и никто другой в этот момент не сможет их изменить. Другой пользователь получит доступ к документу только после его освобождения, а значит, не сможет затереть изменения.

А теперь серьезный недостаток - если на какую-то запись в таблице есть блокировка, то у вас возникнут проблемы с изменением структуры. Это значит, что нельзя будет добавить или удалить какое-то поле. Чтобы внести изменения в структуру таблицы, придется ждать окончания рабочего дня или просить всех пользователей выйти из программы. В этом случае, если в системе останутся какие-то не закрытые сессии, то их можно будет убивать, потому что они явно мертвые.

**Листинг 1**

procedure TSomeDocument.FormShow(Sender: TObject);

var

oldSql : String;

begin

// сохраняем запрос и добавляем операторы блокировки

oldSql:=odsDocs.SQL.Text;

odsDocs.SQL.Add(' FOR UPDATE NOWAIT');

try

// пытаемся открыть набор данных

odsDocs.open;

odsDocs.ReadOnly:=false;

// проверяем, найден ли документ

if odsDocs.RecordCount=0 then

begin

Showmessage('Документ не найден, пока вы думали, его уже удалили');

Close;

exit;

end;

except

// документ заблокирован, поэтому открываем его только для чтения

odsDocs.SQL.Text:=oldSql;

Showmessage('Документ заблокирован другим пользователем, открываем только для чтения');

odsDocs.open;

odsDocs.ReadOnly:=true;

end;

end;