Теоретически, все транзакции должны быть изолированы друг от друга. Но в таком случае доступность данных значительно бы понизилась, поскольку операции чтения транзакции блокировали бы операции записи в других транзакциях, и наоборот. Если доступность данных является важным требованием, то это свойство можно ослабить, используя уровни изоляции.

Уровень изоляции задает степень защищенности выбираемых транзакцией данных от возможности изменения другими транзакциями. Прежде чем приступить к подробному рассмотрению существующих уровней изоляции, рассмотрим несколько сценариев, которые могут возникнуть, если не использовать блокировку.

Если блокировка не используется и, следовательно, транзакции не изолированы друг от друга, то могут возникнуть следующие проблемы: «грязное чтение», неповторяемое чтение и фантомы.

**Грязное чтение** происходит, когда транзакция считывает данные, которые еще не были зафиксированы. Например, предположим, что транзакция 1 обновляет строку. Транзакция 2 считывает обновленную строку до фиксации обновления транзакцией 1. Если транзакция 1 выполняет откат изменения, транзакция 2 будет иметь данные для чтения, которые считаются никогда не существовавших.

Проблема **неповторяемого чтения** при одновременном конкурентном доступе к данным возникает, когда один процесс считывает данные несколько раз, а другой процесс изменяет эти данные между двумя операциями чтения первого процесса. В таком случае значения двух чтений будут разными.

Проблема **фантомов** при параллельном одновременном конкурентном доступе к данным подобна проблеме неповторяемого чтения, поскольку две последовательные операции чтения могут возвратить разные значения. Но в данном случае причиной этому является считывание разного числа строк при каждом чтении. Дополнительные строки называются фантомами и вставляются другими транзакциями.

------------------------------

Используя уровни изоляции, можно указать, какие проблемы одновременного конкурентного доступа могут иметь место, а какие требуется избежать. Существует четыре основных уровней изоляции, которые управляют выполнением операции чтения данных:

* READ UNCOMMITTED
* READ COMMITTED
* REPEATABLE READ
* SERIALIZABLE

Сейчас мы рассмотрим каждый из них

--------------------------------

Уровень изоляции READ UNCOMMITTED предоставляет самую простую форму изоляции между транзакциями, поскольку он вообще не изолирует операции чтения других транзакций. Когда транзакция выбирает строку при этом уровне изоляции, она не задает никаких блокировок и не признает никаких существующих блокировок. Считываемые такой транзакцией данные могут быть несогласованными. В таком случае транзакция читает данные, которые были обновлены какой-либо другой активной транзакцией. А если для этой другой транзакции позже выполняется откат, то значит, что первая транзакция прочитала данные, которые никогда по-настоящему не существовали.

Из четырех проблем одновременного конкурентного доступа к данным, описанных в предшествующем разделе, уровень изоляции READ UNCOMMITTED допускает три: грязное чтение, неповторяемое чтение и фантомы.

Применение уровня изоляции READ UNCOMMITTED обычно крайне нежелательно и его следует применять только в тех случаях, когда точность данных не представляет важности или когда данные редко подвергаются изменениям.

-----------------------------------

Уровень READ COMMITTED имеет две формы. Первая форма применяется в пессимистической модели одновременного конкурентного доступа, а вторая - в оптимистической. Мы рассмотрим первую форму этого уровня изоляции.

Транзакция, которая читает строку и использует уровень изоляции READ COMMITTED, выполнят проверку только на наличие монопольной блокировки для данной строки. Если такая блокировка отсутствует, транзакция извлекает строку. (Это выполняется с использованием разделяемой блокировки.) Таким образом предотвращается чтение транзакцией данных, которые не были подтверждены и которые могут быть позже отменены. После того, как данные были прочитаны, их можно изменять другими транзакциями.

Применяемые этим уровнем изоляции разделяемые блокировки отменяются сразу же после обработки данных. (Обычно все блокировки отменяются в конце транзакции.) Это улучшает параллельный одновременный конкурентный доступ к данным, но возможность неповторяемого чтения и фантомов продолжает существовать.

------------------------------------------

В отличие от уровня изоляции READ COMMITTED, уровень REPEATABLE READ устанавливает разделяемые блокировки на все считываемые данные и удерживает эти блокировки до тех пор, пока транзакция не будет подтверждена или отменена. Поэтому в этом случае многократное выполнение запроса внутри транзакции всегда будет возвращать один и тот же результат. Недостатком этого уровня изоляции является дальнейшее ухудшение одновременного конкурентного доступа, поскольку период времени, в течение которого другие транзакции не могут обновлять те же самые данные, значительно дольше, чем в случае уровня READ COMMITTED.

Этот уровень изоляции не препятствует другим инструкциям вставлять новые строки, которые включаются в последующие операции чтения, вследствие чего могут появляться фантомы.

------------------------------------------

Уровень изоляции SERIALIZABLE является самым строгим, потому что он не допускает возникновения всех четырех проблем параллельного одновременного конкурентного доступа, перечисленных ранее. Этот уровень устанавливает блокировку на всю область данных, считываемых соответствующей транзакцией. Поэтому этот уровень изоляции также предотвращает вставку новых строк другой транзакцией до тех пор, пока первая транзакция не будет подтверждена или отменена.

Уровень изоляции SERIALIZABLE реализуется, используя метод блокировки диапазона ключа. Суть этого метода заключается в блокировке отдельных строк включительно со всем диапазоном строк между ними. Блокировка диапазона ключа блокирует элементы индексов, а не определенные страницы или всю таблицу. В этом случае любые операции модификации другой транзакцией невозможны, вследствие невозможности выполнения требуемых изменений элементов индекса.

-----------------------------------------------------

В заключение обсуждения четырех уровней изоляции следует упомянуть, что требуется знать, что чем выше уровень изоляции, тем меньше степень одновременного конкурентного доступа. Таким образом, уровень изоляции READ UNCOMMITTED меньше всего уменьшает одновременный конкурентный доступ. С другой стороны, он также предоставляет наименьшую изоляцию параллельных конкурентных транзакций. Уровень изоляции SERIALIZABLE наиболее сильно уменьшает степень одновременного конкурентного доступа, но гарантирует полную изоляцию параллельных конкурентных транзакций.

Да и как оказалось, разные СУБД по-разному воспринимают уровни изолированности. Могут иметь разнообразные нюансы в обеспечении изоляции, иметь дополнительные уровни или не иметь общеизвестных. В большинстве приложений уровень изолированности редко меняется и используется значение по умолчанию (например, в MySQL это repeatable read, в PostgreSQL — read committed).

Но периодически возникают, задачи, в которых поиск лучшего баланса между высокой согласованностью данных или скоростью выполнения транзакций может помочь решить некоторую прикладную задачу.