Транзакции

1. Транзакция - это последовательность операций, выполняемых как единое целое. Представляет собой набор операций, которые должны быть выполнены полностью или не выполнены вовсе, обладающий свойствами:
2. Атомарности - либо будут выполнены все действия транзакции, либо никакие.
3. Согласованности - Транзакция должна переводить базу данных из одного согласованного состояния в другое. Это означает, что все правила и ограничения базы данных должны соблюдаться до и после выполнения транзакции.
4. Изолированность – транзакция выполняется так, как будто не знает о существовании других транзакций.
5. **Эффект потерянного обновления (аномалия** сериализации**)**возникает, когда несколько транзакций обновляют одни и те же данные, не учитывая изменений, сделанных другими транзакциями.
6. Эффект грязного чтения: Транзакция читает данные, записанные параллельной незавершённой транзакцией.
7. **Эффект неповторяемого чтения** Транзакция повторно читает те же данные, что и раньше, и обнаруживает, что они были изменены другой транзакцией (которая завершилась после первого чтения).
8. Эффект фантомного чтения Транзакция повторно выполняет запрос, возвращающий набор строк для некоторого условия, и обнаруживает, что набор строк, удовлетворяющих условию, изменился из-за транзакции, завершившейся за это время.
9. Долговечность - После завершения транзакции все изменения, сделанные в рамках этой транзакции, должны быть сохранены и оставаться в базе данных даже в случае сбоев системы.

Уровни изоляции

Уровень изоляции Read Committed

\* Этот уровень изоляции гарантирует, что транзакция может читать только данные, которые были зафиксированы (committed) другими транзакциями. Оператор Select делает снимок базы данных до начала выполнения запроса. Два Select могут выдать разные результаты если другие транзакции зафиксируют изменения после запуска первого, но до запуска второго.

Если одна транзакция читает данные, другая транзакция не сможет изменить эти данные до тех пор, пока первая транзакция не завершится.

Если одна транзакция изменяет данные, другая транзакция не сможет прочитать эти данные до тех пор, пока первая транзакция не завершится.

Исключает: грязное чтение

Не исключает: неповторяемое чтение, фантомное чтение, аномалию сериализации

Уровень изоляции Repeatable Read

\*В режиме Repeatable Read видны только те данные, которые были зафиксированы до начала транзакции, но не видны незафиксированные данные и изменения, произведённые другими транзакциями в процессе выполнения данной транзакции. (Однако запрос будет видеть эффекты предыдущих изменений в своей транзакции, несмотря на то, что они не зафиксированы.)

Исключает: грязное чтение, неповторяемое чтение, фантомное чтение

Не исключает: аномалию сериализации

Уровень изоляции Serializable

\*На этом уровне моделируется последовательное выполнение всех зафиксированных транзакций, как если бы транзакции выполнялись одна за другой, последовательно, а не параллельно. Однако, как и на уровне Repeatable Read, на этом уровне приложения должны быть готовы повторять транзакции из-за сбоев сериализации. Фактически этот режим изоляции работает так же, как и Repeatable Read, только он дополнительно отслеживает условия, при которых результат параллельно выполняемых сериализуемых транзакций может не согласовываться с результатом этих же транзакций, выполняемых по очереди. Это отслеживание не привносит дополнительных препятствий для выполнения, кроме тех, что присущи режиму Repeatable Read, но тем не менее создаёт некоторую добавочную нагрузку, а при выявлении исключительных условий регистрируется аномалия сериализации и происходит сбой сериализации.