

Возможно ли изменять данные с помощью «проверок» (check)?

- ☒ Нет, это невозможно.
- ☐ Зависит от того, имеет ли проверка ссылку на функцию.
- ☐ Да, с помощью некоторых проверок.
- ☐ Зависит от того, используется ли при проверке первичный ключ.

[Очистить мой выбор](#)

[Следующая страница](#)

Проблема вида «потерянное обновление» возникает, когда:

- ☐ Операция UPDATE выполнялась через представление, и таким образом становится «невидимой» для СУБД.
- ☒ Одна транзакция теряет результат выполненного ею обновления из-за того, что чуть позже было выполнено обновление тех же данных другой транзакцией.
- ☐ Операция UPDATE отменяется триггером.
- ☐ На операцию UPDATE поставлен тайм-аут.

[Очистить мой выбор](#)

Согласно свойствам транзакции (ACID)...

- ☒ Транзакции не влияют на результат выполнения друг друга.
- ☐ Временные (внутренние) результаты транзакций должны быть видны приложению.
- ☒ После завершения транзакции все изменения гарантированно сохраняются и не могут «сами по себе» частично или полностью «отмениться».
- ☐ Хранимая функция не может вызывать хранимую процедуру, и наоборот.

[Вернуться на предыдущую страницу](#)

[Следующая страница](#)

К преимуществам «проверок» (check) относится...

- ☐ Проверки обладают большими возможностями, чем триггеры и представления.
- ☒ В отличие от триггеров, проверки активируются при любых операциях изменения данных.
- ☐ Проверки более надёжны, чем триггеры.
- ☒ Проверки, как правило, более просты в использовании и выполняются быстрее, чем триггеры.

[Вернуться на предыдущую страницу](#)

[Следующая страница](#)

В чём заключается разница между понятиями «хранимая процедура» и «хранимая функция»?

- ☐ Хранимая процедура может только считывать данные, хранимая функция может выполнять любые операции с данными.
- ☒ Хранимая процедура может порождать транзакции, хранимая функция – нет.
- ☐ Хранимая процедура может быть вызвана из триггера, хранимая функция – нет.
- ☒ Хранимая процедура имеет свой собственный синтаксис для вызова и обработки результатов, хранимая функция может быть использована в любом SQL-запросе.

[Вернуться на предыдущую страницу](#)

[Следующая страница](#)

В чём заключается разница между понятиями «хранимая процедура» и «хранимая функция»?

- ☐ Хранимая процедура может быть вызвана из представления, хранимая функция – нет.
- ☐ Хранимая процедура может только изменять данные, хранимая функция может выполнять любые операции с данными.
- ☒ Хранимая процедура может вызывать другие хранимые процедуры и функции, хранимая функция может вызывать только другие хранимые функции, но не хранимые процедуры.
- ☒ Хранимая процедура обладает полным спектром возможностей по обработке исключительных ситуаций, хранимая функция обладает узкими возможностями по обработке исключительных ситуаций (в некоторых СУБД — вообще не может обрабатывать такие ситуации).

Недостатками использования хранимых подпрограмм является...

- ☒ Усложнение тестирования и/или диагностики.
- ☒ Потенциальные проблемы с безопасностью (если на выполнение хранимой подпрограммы выставлены неверные права).
- ☐ Снижение читаемости кода, т.к. вызов подпрограммы — это длинная и сложная синтаксическая конструкция.
- ☐ Снижение производительности в силу того факта, что данные не передаются за пределы СУБД.

раница

Следующая страница

Возможно ли изменять данные с помощью представлений?

- ☐ Зависит от того, включает ли представление первичный ключ.
- ☐ Нет, это невозможно.
- ☐ Зависит от того, имеет ли представление индекс.
- ☒ Да, с помощью некоторых представлений.

[Очистить мой выбор](#)

раница

Следующая страница

Недостатками использования представлений является...

- ☒ Наличие ограничений СУБД на операции, доступные через представления.
- ☒ Необходимость создания и обслуживания дополнительного кода.
- ☐ Усложнение выполнения запросов.
- ☐ Возможность повышения производительности в определённых условиях.

раница

Следующая страница

Триггеры уровня запроса...

- ☒ Активируются один раз для всей операции модификации, при этом вся изменённая информация доступна внутри триггера.
- ☐ Способны изменить лишь операцию, не данные.
- ☐ Активируются один раз для всей операции модификации, при этом только новые значения данных доступны внутри триггера.
- ☐ Активируются один раз для всей операции модификации, при этом только старые значения данных доступны внутри триггера.

[Очистить мой выбор](#)

раница

Следующая страница

Уровень изолированности транзакции «чтение неподтверждённых данных» (read uncommitted) обеспечивает защиту от следующих проблем:

- ☐ Грязное чтение.
- ☐ Неповторяющееся чтение.
- ☒ Потерянное обновление.
- ☐ Фантомное чтение.

[Очистить мой выбор](#)

раница

Следующая страница

В чём заключается разница между понятиями «представление» и «материализованное представление»?

- ☐ Материализованное представление всегда имеет индексы, представление – нет.
- ☒ Материализованное представление хранит данные, представление – нет.
- ☐ Представление хранит данные, материализованное представление – нет.
- ☐ Представление всегда имеет индексы, материализованное представление – нет.

[Очистить мой выбор](#)

раница

Следующая страница

Триггер может быть активирован при выполнении следующих операций:

- ☐ SELECT.
- ☒ UPDATE.
- ☐ CHECK FAILURE.
- ☒ INSERT.

раница

Следующая страница

Преимуществами использования представлений является...

- ☐ Возможность обнаружения неочевидных ограничений СУБД.
- ☐ Возможность создания и обслуживания дополнительного кода.
- ☒ Возможность построения простого и надёжного API.
- ☒ Упрощение выполнения запросов.

раница

Следующая страница

Триггеры уровня строки...

- ☐ Активируются каждый раз заново для каждой изменённой строки, при этом данные всех изменённых строк доступны внутри триггера.
- ☐ Активируются при изменении только одной строки.
- ☐ Способны изменить только одну строку данных (строку, для которой запущен триггер).
- ☒ Активируются каждый раз заново для каждой изменённой строки, при этом только данные изменяемой строки доступны внутри триггера.

[Очистить мой выбор](#)

раница

Следующая страница

С использованием триггеров возможно выполнять...

- ☐ Обслуживание СУБД.
- ☒ Сложные каскадные операции.
- ☒ Контроль операций модификации данных для обеспечения выполнения правил бизнес-логики.
- ☐ Отладку веб-приложений.

раница

Следующая страница

При использовании INSTEAD OF триггеров необходимо...

- ☐ Повторить попытку изменения данных после завершения работы триггера.
- ☐ Отключить триггер перед началом изменения данных.
- ☒ Произвести необходимые изменения данных внутри триггера.
- ☐ Использовать представление для изменения данных.

[Очистить мой выбор](#)

раница

Следующая страница

Уровень изолированности транзакции «снимок» (snapshot) обеспечивает защиту от следующих проблем:

- ☐ Фантомное чтение.
- ☒ Грязное чтение.
- ☒ Неповторяющееся чтение.
- ☒ Потерянное обновление.

раница

Следующая страница

С помощью «проверок» (check) можно реализовать такие ограничения, как...

- ☒ «Номер паспорта должен иметь указанный формат».
- ☐ «Должен существовать хотя бы один пользователь с правами администратора».
- ☒ «Логин и пароль не должны совпадать».
- ☐ «Категория товаров должна хранить количество товаров».

раница

Следующая страница

Существуют следующие типы триггеров...

- ☒ AFTER триггеры.
- ☐ ALWAYS триггеры.
- ☒ BEFORE триггеры.
- ☒ INSTEAD OF триггеры.

раница

Следующая страница

Уровень изолированности транзакций – это...

- ☒ Условное значение, показывающее, насколько внутреннее состояние базы данных в момент выполнения транзакции доступно другим, одновременно выполняемым транзакциям.
- ☐ Тип связи между таблицами.
- ☐ Абстракция, описывающая API базы данных.
- ☐ Свойство, демонстрирующее типы транзакций, которые доступны для конкретной таблицы.

[Очистить мой выбор](#)

Преимуществами использования хранимых подпрограмм является...

- ☐ Более высокая производительность в сравнении с непосредственным выполнением соответствующего кода.
- ☒ Возможность повысить читаемость кода, т.к. вызов подпрограммы — это короткая синтаксическая конструкция.
- ☐ Упрощение тестирования и/или диагностики.
- ☒ Возможность написать код один раз, а затем использовать его вызовы многократно.

раница

Следующая страница

Согласно свойствам транзакции (ACID)...

- ☒ Операции внутри транзакции не могут быть завершены или отменены «сами по себе», транзакция всегда фиксируется или отменяется целиком.
- ☐ Для каждой таблицы должен существовать хотя бы один атомарный первичный ключ.
- ☐ Все хранимые процедуры должны вызываться непосредственно приложением, работающим с базой данных.
- ☒ Итоговые результаты транзакции не могут нарушать правила консистентности базы данных.

раница

Следующая страница

Уровень изолированности транзакции «повторяющееся чтение» (repeatable read) обеспечивает защиту от следующих проблем:

- ☒ Неповторяющееся чтение.
- ☒ Потерянное обновление.
- ☒ Грязное чтение.
- ☐ Фантомное чтение.

раница

Следующая страница

К недостаткам «проверок» (check) относится...

- ☒ Проверки могут снизить производительность.
- ☒ В случае провала проверки невозможно сформировать информативное сообщение об ошибке.
- ☐ Проверки выполняются медленнее, чем триггеры.
- ☐ В отличие от триггеров, проверки активируются при операциях изменения данных и операциях выбора.

раница

Следующая страница

В реляционных базах данных транзакция – это...

- ☒ Набор операций с базой данных, который представляет собой неделимую логическую единицу.
- ☐ Способ изменения данных.
- ☐ Одиночная операция, которая для приложения выглядит как серия операций.
- ☐ Техническое обслуживание, направленное на оптимизацию данных, хранимых в таблице.

[Очистить мой выбор](#)

раница

Следующая страница

Проблема вида «грязное чтение» возникает, когда:

- ☐ Произошёл сбой в связи между приложением и СУБД.
- ☐ Результат операции SELECT был получен путём выполнения хранимой функции.
- ☐ Операция SELECT возвращает слишком много данных.
- ☒ Одна транзакция считывает временные данные, созданные другой транзакцией (такие данные никогда не будут сохранены в базе данных).

[Очистить мой выбор](#)

раница

Следующая страница

Проблема вида «фантомное чтение» возникает, когда:

- ☒ Транзакция получает разные результаты агрегирования данных при первой и второй попытках чтения.
- ☐ Результат операции SELECT используется неизвестным пользователем.
- ☐ Результат операции SELECT содержит случайные значения.
- ☐ Операция SELECT возвращает данные, которые уже были удалены.

[Очистить мой выбор](#)

раница

Следующая страница

Проблема вида «неповторяющееся чтение» возникает, когда:

- ☐ Результат операции SELECT используется несколькими пользователями.
- ☐ Результат операции SELECT содержит случайные значения.
- ☐ Операция SELECT возвращает данные, которые уже были удалены.
- ☒ Транзакция получает разные результаты при первой и второй попытках чтения одних и тех же данных.

[Очистить мой выбор](#)

раница

Следующая страница

Уровень изолированности транзакции «чтение подтверждённых данных» (read committed) обеспечивает защиту от следующих проблем:

- ☐ неповторяющееся чтение.
- ☐ фантомное чтение.
- ☒ потерянное обновление.
- ☒ грязное чтение.

раница

Закончить попытку...