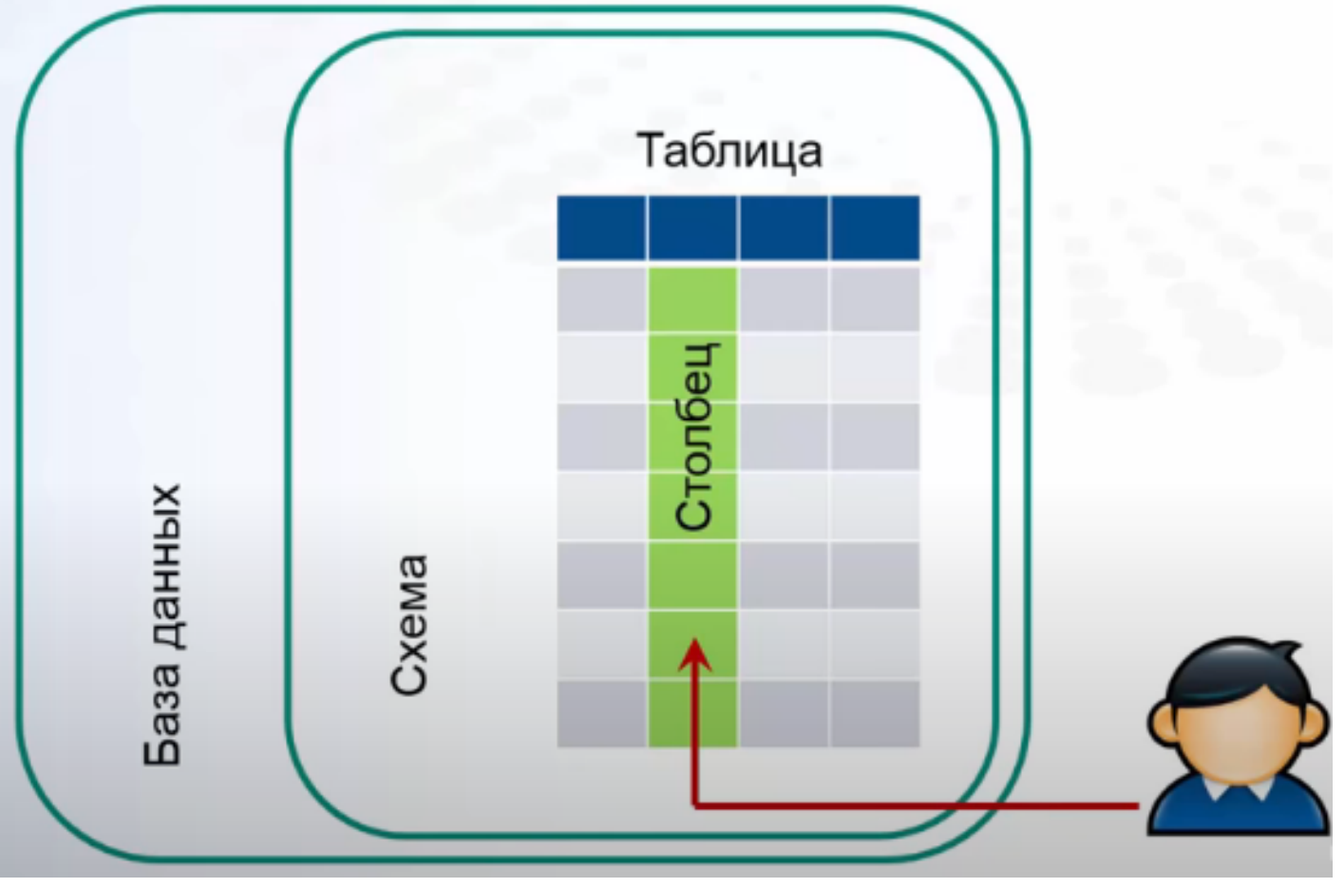
Безопасность на уровне строк (RLS)

Декларативное управление доступом

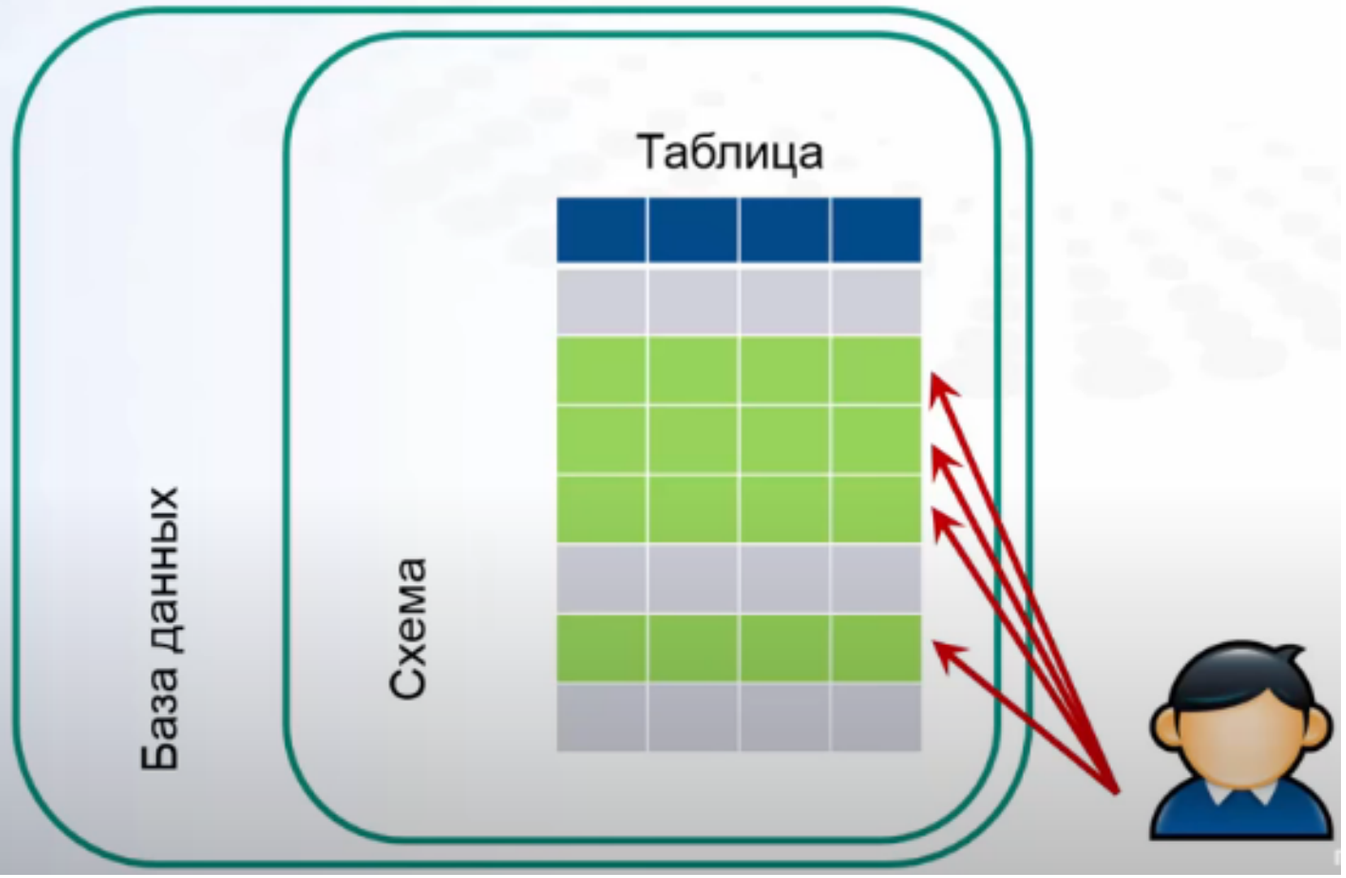
- Назначаем разрешения на структурные элементы баз данных
- Но не на сами данные (строки)



Безопасность на уровне строк (RLS)

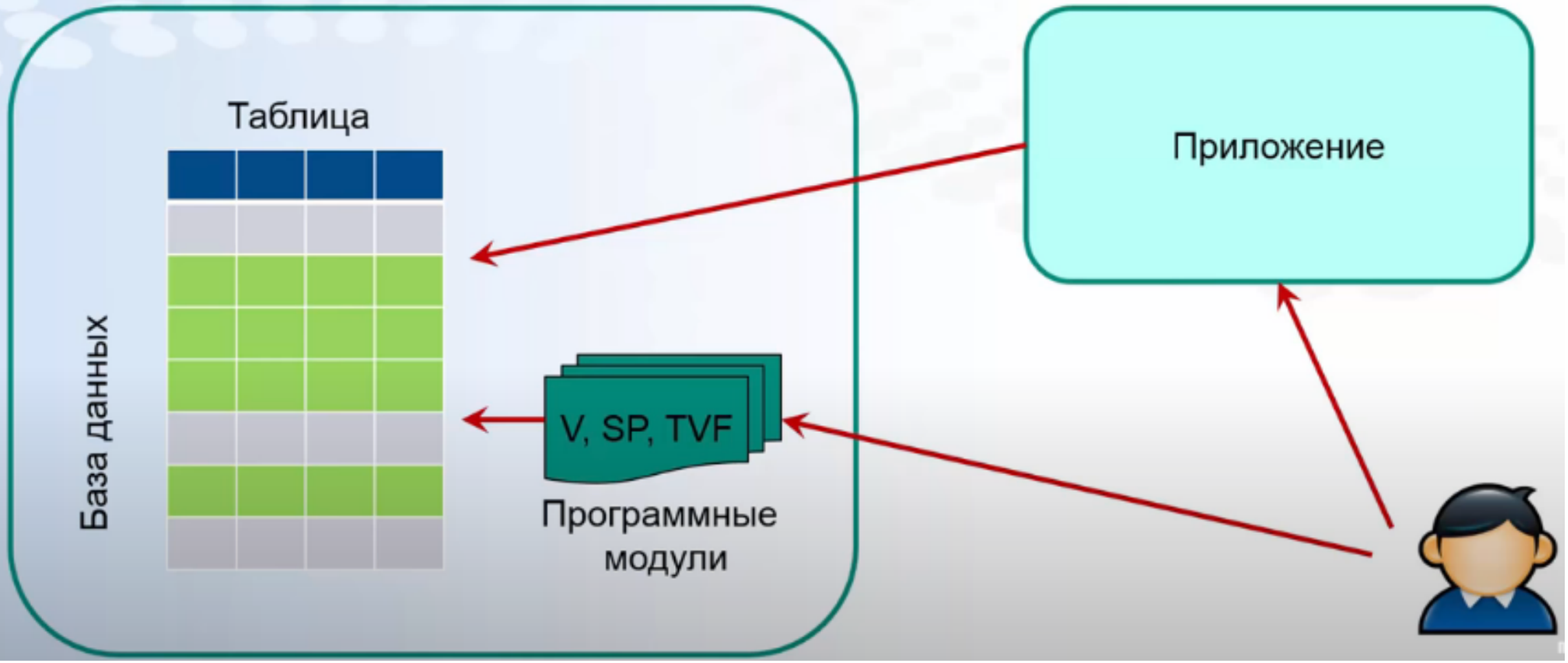
Динамическое управление доступом

- Обычно таблица хранит список однотипных объектов (товары, заказы, клиенты…)
- Разным пользователям требуется доступ к разным экземплярам объектов (товарам, заказам, клиентам…)



Безопасность на уровне строк (RLS)

Традиционные решения для построчного доступа



Безопасность на уровне строк (RLS)

Безопасность на уровне строк позволяет использовать членство в группе или контекст выполнения для управления доступом к строкам в таблице базы данных.
Логика ограничения доступа расположена на уровне базы данных, а не отдельно от данных на другом уровне приложений.
Система базы данных применяет ограничения доступа каждый раз при попытке доступа к данным с любого уровня.

RLS реализуется с помощью инструкции Transact-SQL CREATE SECURITY POLICY и предикатов, созданных как встроенные функции с табличным значением

Безопасность на уровне строк (RLS)

Поддерживает два типа предикатов безопасности.

- Предикаты фильтра автоматически фильтруют строки, доступные для операций чтения (SELECT, UPDATE и DELETE).
- Предикаты блокировки явно блокируют операции записи (AFTER INSERT, AFTER UPDATE, BEFORE UPDATE, BEFORE DELETE), которые нарушают предикат.

Безопасность на уровне строк (RLS)

Предикаты фильтра применяются при чтении данных из базовой таблицы. Они влияют на все операции получения: **SELECT** , **DELETE** и **UPDATE** . Пользователи не могут выбирать или удалять отфильтрованные строки. Пользователь не может обновлять отфильтрованные строки. Но можно обновить строки таким образом, чтобы они впоследствии были отфильтрованы.

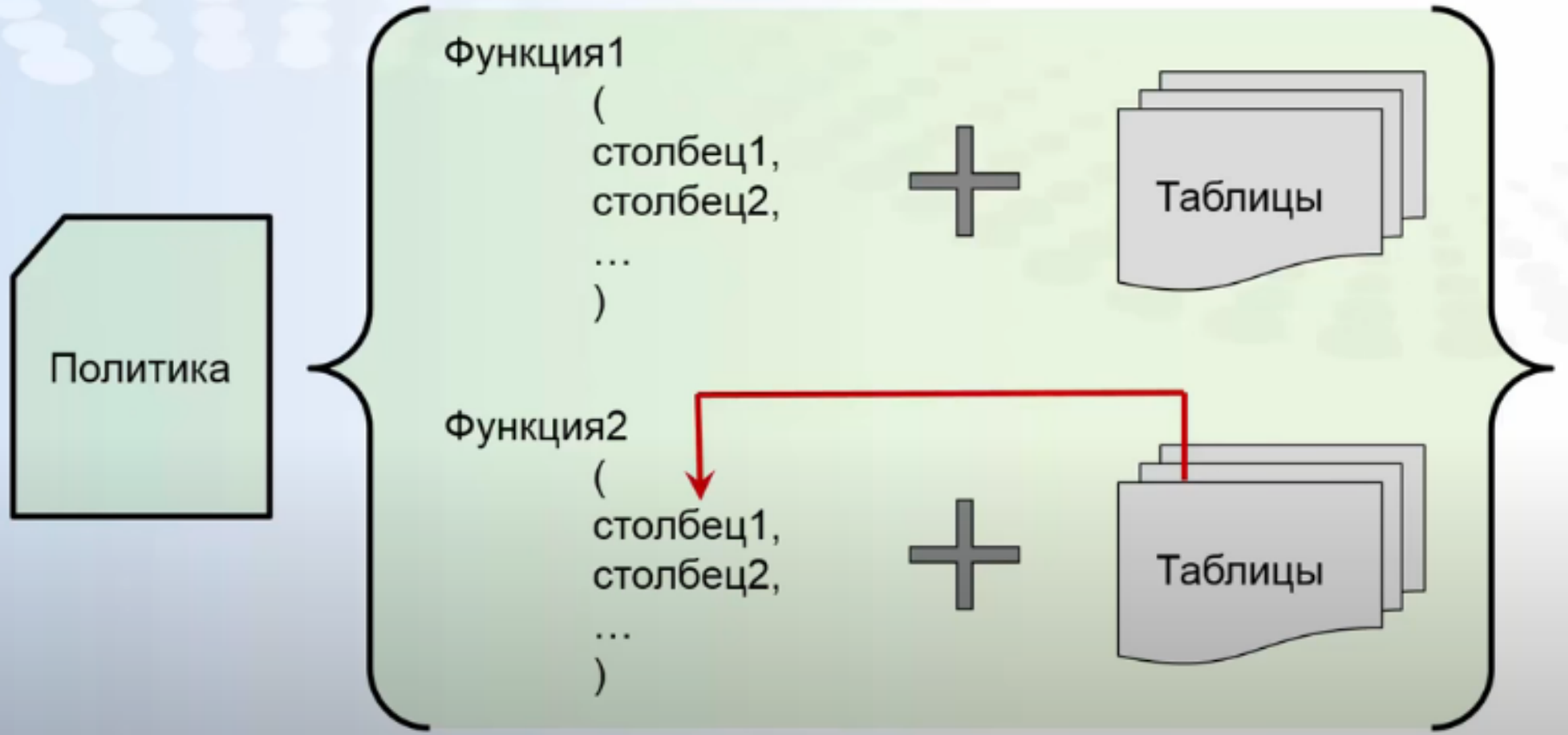
Предикаты блокировки влияют на все операции записи.
Предикаты AFTER INSERT и AFTER UPDATE могут запретить пользователям обновлять строки до значений, нарушающих предикат.
Предикаты BEFORE UPDATE могут запретить пользователям обновлять строки, которые в настоящее время нарушают предикат.
Предикаты BEFORE DELETE могут блокировать операции удаления.

Безопасность на уровне строк (RLS)

Разрешения
Для создания, изменения или удаления политик безопасности требуется разрешение **ALTER ANY SECURITY POLICY .** Для создания или удаления политики безопасности требуется разрешение **ALTER** для схемы.
Кроме того, для каждого добавляемого предиката требуются следующие разрешения:
Разрешения SELECT и **REFERENCES** для функции, используемой в качестве предиката.
Разрешение REFERENCES на целевую таблицу, привязанную к политике.
Разрешение REFERENCES для каждого столбца из целевой

Безопасность на уровне строк (RLS)

Механизм Row-Level Security (RLS)



Безопасность на уровне строк (RLS)

В фильтрующих функциях вы можете

- Проверять контекст подключения
  - Имя пользователя, членство в ролях, название приложения, компьютера…
- Проверять контекст сервера
  - Дата, время, настройки…
- Анализировать дополнительные сведения от приложения
  - CONTEXT_INFO ()
- Обращаться к другим таблицам

Безопасность на уровне строк (RLS)

Обратите внимание!

- **Работает атака с подбором фильтра**
- **При отключении политики таблица не защищена**
- Другие механизмы SQL-сервера видят строки таблицы без ограничений
  - Статистика (DBCC SHOW_STATISTICS)
  - Полнотекстовые индексы
- Некоторые механизмы несовместимы с RLS:
  - Размещение таблиц в памяти (In-memory)
  - Отслеживание изменений Change Data Capture
- Чем сложнее фильтры, тем медленнее выполняются запросы
  - К каждому запросу добавляется условие WHERE (см. план исполнения запроса)
  - Удлиняются транзакции, накладывается больше блокировок и т.п.

Безопасность на уровне строк (RLS)

Что хорошего в RLS по сравнению с классическими решениями

- Появляется возможность быстро и просто переключаться между политиками
- Не требуется усложнять приложение
  - Если с вашей базой работает несколько приложений, нет необходимости внедрять проверки строк в каждое приложение
- Не возникает альтернативных таблицам модулей (представления, функци), через которые можно получить доступ к данным

Безопасность на уровне строк (RLS)

Самое главное

RLS – это ослабление безопасности

RLS – это повышение управляемости