Авторские права
© Postgres Professional, 2017 год.
Авторы: Егор Рогов, Павел Лузанов

Использование материалов курса
Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Отказ от ответственности
Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Темы

Что такое политики защиты строк

Условия применения политик

Несколько политик на одной таблице

Политика защиты строк

Политика ограничивает видимость строк в таблице

- видимость определяется предикатами, которые вычисляются *для каждой строки* с правами вызывающего
- клиенту доступны только те строки, для которых предикаты истинны

Предикат для существующих строк

- используется операторами SELECT, UPDATE, DELETE
- при нарушении политики не возникает ошибка (если только не сброшен параметр row_security)

Предикат для новых строк

- используется операторами INSERT, UPDATE
- если не задан, используется первый предикат
- при нарушении политики возникает ошибка

Политики защиты строк (RLS, Row Level Security) позволяют управлять доступом к таблице на уровне отдельных строк. Другое его название — Fine Grained Access Control. Этот механизм появился в версии 9.5.

Политики являются вспомогательным механизмом: роль по-прежнему должна иметь доступ к таблице, предоставленный привилегиями.

Политика определяет предикаты (логические выражения), которые вычисляются для каждой строки запроса. Полученное значение говорит о том, надо ли показывать строку или нет.

Предикатов может быть два. Первый определяется для *существующих* строк и используется операторами SELECT, UPDATE, DELETE. Если для какой-то строки предикат принимает ложное значение, строка не попадает в итоговую выборку. Упрощенно можно считать, что предикат просто «дописывается» к условию WHERE — хотя на самом деле все несколько сложнее.

Если параметр row_security = off, при нарушении политики будет зафиксирована ошибка. Это полезно при изготовлении логической резервной копии, чтобы гарантировать, что в нее попали все строки всех таблиц.

Второй предикат определяет видимость *новых* строк. Он проверяется командами INSERT и UPDATE; при нарушении политики возникает ошибка.

https://postgrespro.ru/docs/postgresql/10/ddl-rowsecurity

Условия применения

Политика применяется

- к таблице, для которой включена защита
- для указанных ролей и для указанных операторов (SELECT, INSERT, UPDATE, DELETE)

Политика не применяется

- при проверке ограничений целостности
- для суперпользователей и ролей с атрибутом BYPASSRLS
- для владельца (если не включить принудительно)

Чтобы политики защиты строк начали работать, нужно специальным образом включить этот механизм для каждой таблицы.

При создании политики можно также задать, для каких ролей она будет работать (по умолчанию — для всех) и для каких операторов (по умолчанию — также для всех).

Политики не применяются при проверке ограничений целостности — независимо от настроенных политик СУБД гарантирует целостность данных.

Политики не применяются для суперпользователей (для них, как обычно, никакие проверки безопасности не выполняются) и для ролей с атрибутом BYPASSRLS.

Для владельца таблицы политики также не работают, хотя специальной командой можно включить защиту и для владельца.

Несколько политик

Разрешительные политики

видимость должна предоставить хотя бы одна разрешительная политика

если не определена ни одна политика, строка не видима

Ограничительные политики

видимость должны предоставить все ограничительные политики, если они определены

На одной таблице можно определить несколько политик. В этом случае будут учитываться все предикаты.

По умолчанию создаются *разрешительные (permissive)* политики. Чтобы строка была видна, достаточно, чтобы *хотя бы один* из предикатов был истинен.

Но если на таблице включена защита строк, и при этом не определено ни одной разрешительной политики, не будет видна ни одна строка.

Дополнительно можно создать и *ограничительные (restricted)* политики. Если они заданы, то *все* предикаты должны быть истинны.

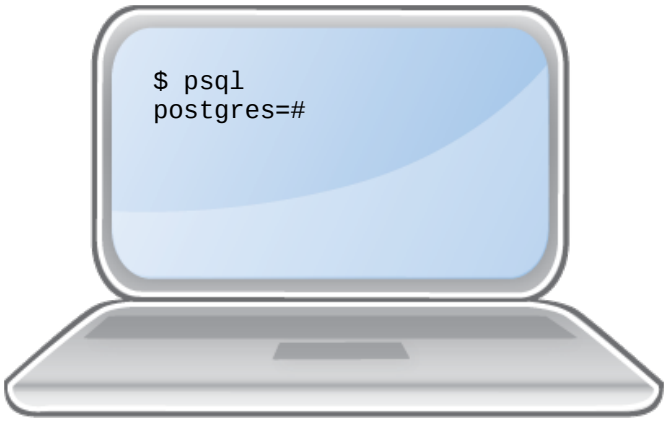
Иными словами, если определены только разрешительные политики с предикатами P_1,...,P_N, то для каждой строки вычисляется выражение

P_1 OR … OR P_N.

А если к тому же определены ограничительные политики с предикатами R_1,…, R_M, то для каждой строки будет вычисляться выражение

(P_1 OR … OR P_N) AND R_1 AND … AND R_M.

Демонстрация



Итоги

Привилегии управляют доступом к таблицам и столбцам, политики защиты строк — к строкам

Проще и эффективнее, чем реализация с помощью представлений и триггеров

Практика

1. Продолжая пример из демонстрации, создайте роль для Чарли и назначьте ему *два* отдела в таблице пользователей.
2. Определите политики защиты строк таким образом, чтобы:
   - роли видели строки только тех отделов, с которыми связаны;
   - роли, связанные с одним отделом, могли добавлять строки с суммой в пределах 100 рублей;
   - роли, связанные с несколькими отделами, могли добавлять строки с любой суммой.
3. Проверьте корректность настроенных политик.
4. Оцените накладные расходы на политики защиты строк, выполнив один и тот же запрос от имени обычной и суперпользовательской ролей.