**Тема занятия №11: Основы PL.pgSQL. Триггеры и функции**

**PL/pgSQL** это процедурный язык для СУБД PostgreSQL. Целью проектирования PL/pgSQL было создание загружаемого процедурного языка, который:

- используется для создания функций, процедур и триггеров;

- добавляет управляющие структуры к языку SQL;

- может выполнять сложные вычисления;

- наследует все пользовательские типы, функции, процедуры и операторы;

- может быть определён как доверенный язык;

- прост в использовании.

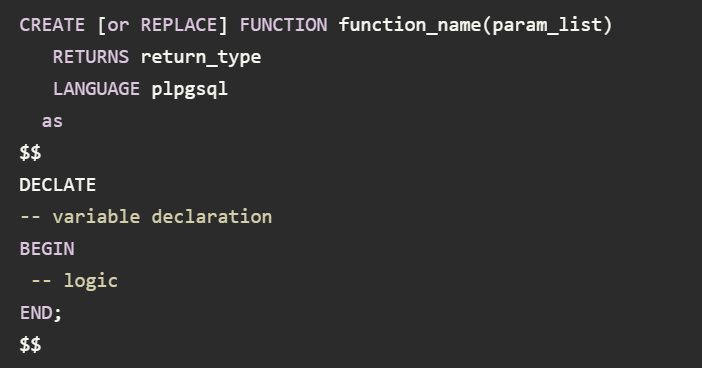
SQL, как и любой другой язык программирования, предоставляет функции и хранимые процедуры.

Функции и хранимые процедуры в SQL, как и в любом другом языке программирования, обеспечивают возможность повторного использования и гибкость. Функции и хранимые процедуры представляют собой блок кода или запросов, хранящихся в базе данных, которые можно использовать снова и снова. Вместо того чтобы писать одни и те же запросы, удобнее сгруппировать все запросы и сохранить их, чтобы можно было использовать их много раз.

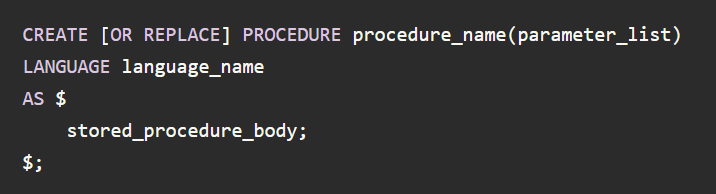
SQL-функции выполняют произвольный список операторов SQL и возвращают результат последнего запроса в списке. В простом случае (не с множеством) будет возвращена первая строка результата последнего запроса. (Помните, что понятие "первая строка" в наборе результатов с несколькими строками определено точно, только если присутствует ORDER BY.) Если последний запрос вообще не вернёт строки, будет возвращено значение NULL.

Кроме того, можно объявить SQL-функцию как возвращающую множество (то есть, несколько строк), указав в качестве возвращаемого типа функции SETOF *некий\_тип*, либо объявив её с указанием RETURNS TABLE (*колонки*). В этом случае будут возвращены все строки результата последнего запроса. Подробнее это описывается ниже.

Тело SQL-функции должно представлять собой список SQL-операторов, разделённых точкой с запятой. Точка с запятой после последнего оператора может отсутствовать. Если только функция не объявлена как возвращающая void, последним оператором должен быть SELECT, либо INSERT, UPDATE или DELETE с предложением RETURNING.



Создание хранимой процедуры, почти такое же, как создание функции с небольшим отличием — в хранимой процедуре нет return. Остальное почти идентично.



- **create [or replace] function имя\_функции** — создает или заменяет функцию, если она существует, с заданным именем и параметрами;

- **returns return\_type** — тип данных, который возвращает функция;

- **язык plpgsql** — указывает на процедурное расширение PostgreSQL;

- внутри знака **$** является телом функции;

- **declare** — показывает, как объявляются или инициализируются переменные;

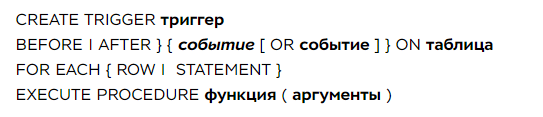
- блок кода **[begin — end]** — содержит всю логику функции;

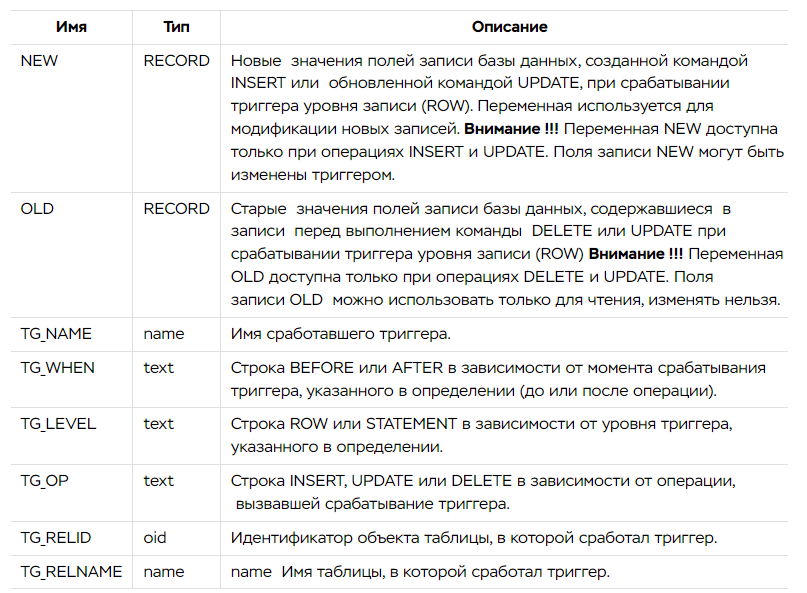
- **begin** — указывает на начало запросов;

- **end** — указывает конец функции.

**Триггер PostgreSQL** – это функция, которая запускается автоматически при возникновении события базы данных в объекте базы данных.

Примеры событий базы данных, которые могут активировать триггер, включают INSERT, UPDATE, DELETE и т. Д. Кроме того, когда вы создаете триггер для таблицы, триггер автоматически сбрасывается при удалении этой таблицы.





**SQL инъекция** — это один из самых доступных способов взлома сайта.  
Суть таких инъекций – внедрение в данные (передаваемые через GET, POST запросы или значения Cookie) произвольного SQL кода. Если сайт уязвим и выполняет такие инъекции, то, по сути, есть возможность творить с БД (чаще всего это MySQL) что угодно.

**Почему вообще возможны SQL-инъекции?**

Когда пользователь вводит и отправляет данные на веб-сайте, эти данные попадают в веб-приложение, которое в свою очередь использует эти данные при доступе к БД. Для обращения к данным в реляционных БД используется SQL

Теперь предположим, что пользователь — злоумышленник, и вводит в строке поиска не номер платежной карты, а вредоносный фрагмент SQL-кода, например:

**‘4’; DROP TABLE Products;**

SELECT \*

FROM PaymentProfile

WHERE UserID = ‘ID’ AND CardNumber = **‘4’; DROP TABLE PaymentProfile;**

Cуть любой SQL-инъекции заключается именно в этом: злоумышленник внедряет («впрыскивает» — отсюда и название атаки «инъекция») вредоносный код в обычную форму ввода или в строку адреса,

Далее практика на **psycopg2**