

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет Безопасности информационных технологий

Методические указания по проведению лабораторной работы №3
“Функции и триггеры в БД”

**Санкт-Петербург
2025**

1. **Цель работы:** Получение навыков написания процедур, функций и триггеров в БД.
2. **Теоретическая информация:**

При наличии необходимости реализации сложного сценария работы с данными в БД применяются процедуры и триггеры. Они представляют собой код, написанный на одном из расширений SQL (для PostgreSQL PL/pgSQL), который позволяет реализовать более сложную обработку данных и имеет в своем арсенале операторы циклов, ветвлений и т.п. Процедура является определена функцией, которую может запускать пользователь, используя клиента БД. Триггер представляет собой операцию, которую нужно выполнить при возникновении события в БД, например, вставка записи в таблицу.

В качестве СУБД, используемой в лабораторной работе, предполагается PostgreSQL. В ней триггеры создаются на основе уже определенных ранее функций.

Оператор для создания процедуры

```
CREATE OR REPLACE FUNCTION функция() RETURNS тип AS
$$
BEGIN
    команды;
END
$$ LANGUAGE plpgsql;
```

Для создания хранимой процедуры нужно определить ее в соответствии с представленным синтаксисом описать команды и возвращаемый результат, который может быть простым типом данных, набором данных, сложным набором данных, пользовательским набором данных.

Для последующего вызова хранимой процедуры можно воспользоваться синтаксисом, приведенным ниже:

Вызов функции

```
SELECT * FROM функция();
```

Пример:

```
CREATE OR REPLACE FUNCTION max_value() RETURNS real AS
$$
DECLARE
    maxVal real;
BEGIN
    maxVal := (select max(sensor_value) from svalues);
    RETURN maxVal;
END
$$ LANGUAGE plpgsql;
```

Для последующего создания триггера необходимо определить триггерную функцию в соответствии с синтаксисом ниже.

Оператор для создания функции для триггера

```
CREATE FUNCTION функция () RETURNS trigger AS $$  
DECLARE  
    объявления;  
BEGIN  
    команды;  
END; $$  
LANGUAGE plpgsql;
```

Пример:

```
CREATE OR REPLACE FUNCTION trigger_update_maxvalue() RETURNS trigger AS  
$$  
DECLARE  
    maxVal real;  
BEGIN  
    IF NEW.sensor_value > (select current_val from max_value)  
    THEN update max_value set current_val = NEW.sensor_value;  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

Оператор определения триггера

```
CREATE TRIGGER триггер  
BEFORE | AFTER } { событие [ OR событие ] } ON таблица  
FOR EACH { ROW | STATEMENT }  
EXECUTE PROCEDURE функция ( аргументы )
```

Добавление новых записей в таблицу производится оператором INSERT. Необходимо в явном виде указать таблицу, в которую добавляются строки, а также список атрибутов добавляемой записи и ее значения.

Пример:

```
CREATE TRIGGER trigger_keep_maxvalue  
AFTER INSERT OR UPDATE ON svalues FOR EACH ROW  
EXECUTE PROCEDURE trigger_update_maxvalue();
```

Все перечисленные операторы имеют возможности по использованию дополнительных параметров, с которыми необходимо ознакомиться в документации.

3. Задание

1. Написать процедуру, которая выполняет агрегацию значений в таблице и обновляет значение в другой таблице. Таким образом, чтобы при запуске пользователем информация в таблице обновлялась и содержала агрегированные значения из другой таблицы.
2. Написать триггер, который будет выполнять действие из 1 пункта автоматически при вставке записи в исходную таблицу. Таким образом, чтобы агрегированная информация всегда была актуальна.

3. Написать триггер, который на основании даты из вставляемой записи, вставлял ее в соответствующую таблицу.
4. Написать триггер, который при вставке в таблицу, производил подмену вставляемого значения в соответствии с уже существующим словарем.
5. Реализуйте триггер, который использует по крайней мере 2-3 специальных переменных (NEW, OLD, TG_OP и др). Список специальных переменных для postgresql <https://postgrespro.ru/docs/postgresql/9.6/plpgsql-trigger>

4. Требования к оформлению отчета и защите

Отчет должен содержать код процедур и триггерных функций, выполненные операторы по каждому пункту задания и вывод, полученный при их выполнении.

5. Источники и информация для подготовки

<http://www.postgresql.org/docs/9.5/static/index.html>

Вопросы к защите ЛР:

1. Дайте определение понятия “триггер”. Опишите основной sql синтаксис при определении триггера. В каких случаях при работе с БД применяются триггеры?
2. Транзакции. Свойства транзакций (атомарность, консистентность, изолированность и устойчивость)
3. Аномалии при выполнении транзакции (Потерянное обновление, «Грязное» чтение, Неповторяющееся чтение, Фантомное чтение)
4. Уровни изоляции транзакции (Чтение неподтверждённых данных, Чтение подтверждённых данных, Повторяющееся чтение, Снимок, Сериализация)
5. Журналы транзакций. Типы журнальных записей.
6. Стратегии ведения журнала транзакций
7. Аудит событий СУБД. Какие события важны с точки зрения ИБ.