



Текстовая расшифровка видео:

## ТРИГГЕРЫ И ПРОЦЕДУРЫ

### План:

- Лайфхаки для версий.

### Лайфхаки для версий

Это то, что очень привязано к реализации в конкретной БД.

С использованием встроенного специального языка можем описать разные дополнительные действия, которые можем захотеть переиспользовать дальше в своих SQL-запросах.

Как это выглядит:

```
CREATE OR REPLACE FUNCTION employee_insert_trigger_fnc()  
  RETURNS TRIGGER AS  
  $$  
  BEGIN  
    UPDATE employees SET current_flag = FALSE WHERE employee_id = NEW.employee_id;  
  RETURN NEW;  
  END;  
  $$  
  LANGUAGE 'plpgsql';  
  
CREATE TRIGGER employee_insert_trigger  
  BEFORE INSERT  
  ON employees  
  FOR EACH ROW  
  EXECUTE PROCEDURE employee_insert_trigger_fnc();
```

В этой процедуре (function) пишем с помощью выбранного языка какие-то действия с данными: это может быть как SQL, так и дополнительные диалекты. В случае Postgres'a может быть PL/pgSQL, если же Postgres современный, собранный с дополнительными встроенными интерпретаторами, то это может



быть и код на Python. Могут использоваться разные интерпретируемые языки.

Для того, чтобы все работало, мы можем сделать то, что позволит работать со Slowly Changing Dimensions. Когда мы вставляем данные в табличку, мы можем выставить флаг Current в «False» для предыдущего значения и «True» для текущего.

В примере мы создаем функцию, вызывающую UPDATE на табличке «Employees». Она выставляет «Current Flag=False» для вставляемого Employee ID.

Создаем триггер «Before insert». Перед тем как что-то вставлять, «дергается» данная функция и для текущего сотрудника, у которого на текущий момент выставлен Current Flag в «True», функция выставляет в «False», а после этого вставляет новый с флагом «True». Так и достигается отслеживание изменений в базе.

Подобное можно делать и вручную. В некоторых случаях это и проще, и читабельнее, и эффективнее.

Проблема заключается в том, что при таком использовании хранимых процедур достаточно сложно отслеживать изменения в коде, делать CI/CD, версионирование, автоматически тестировать новые версии кода и т.д. Все зависит от компетенции команды.

Как это реализовано в Postgres и других базах:

[Почитать про хранимые процедуры](#)

[Почитать про триггеры](#)

Как вам урок?



Изучил, далее >

Слёрм ©

[+7 \(495\) 248-05-80](tel:+7(495)248-05-80)

[Лицензия №ДЛ-1368 от 22.08.2019](#)

[Политика конфиденциальности](#)

[Публичная оферта](#)

