

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №5 «процедуры, функции, триггеры в PostgreSQL»

по дисциплине «Проектирование и реализация баз данных»

Автор: Трубников А.П

Факультет: ИКТ

Группа: K3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2024

В организации необходимо фиксировать время прихода и ухода сотрудников на работу, вычисление общего отработанного времени.

Рассмотрим пример таймера и посмотрим, каким образом можно использовать триггеры для предотвращения ввода сотрудниками неверных данных.

Задание 1:

1. Создайте БД `emp_time`. Используйте консоль `psql`.

```
postgres=# CREATE DATABASE emp_time;
CREATE DATABASE
```

```
CREATE DATABASE emp_time;
```

- ## 2. Проверьте список активных БД

VI

Список баз данных							
Имя ICU	Владелец	Кодировка	Провайдер локали	LC_COLLATE	LC_CTYPE	локаль ICU	Правила
Права доступа							
Lab4	postgres	UTF8	libc	Russian_Russia.1251	Russian_Russia.1251		
emp_time	postgres	UTF8	libc	Russian_Russia.1251	Russian_Russia.1251		
postgres	postgres	UTF8	libc	Russian_Russia.1251	Russian_Russia.1251		
template0	postgres	UTF8	libc	Russian_Russia.1251	Russian_Russia.1251		
=c/postgres		+					
postgres=CTc/postgres							
template1	postgres	UTF8	libc	Russian_Russia.1251	Russian_Russia.1251		
=c/postgres		+					
postgres=CTc/postgres							
(5 строк)							

3. Подключитесь к БД emp time.

```
postgres=# \c emp_time
Вы подключены к базе данных "emp_time" как пользователь "postgres".
```

- #### 4. Создайте таблицы БД

```
CREATE TABLE employee ( id SERIAL PRIMARY KEY, username VARCHAR );
```

```
CREATE TABLE time_punch ( id SERIAL PRIMARY KEY, employee_id INT
REFERENCES employee(id) NOT NULL, is_out_punch BOOLEAN DEFAULT FALSE
NOT NULL, punch_time TIMESTAMP DEFAULT NOW() NOT NULL )
```

cd ~/postgresql && cd base данных emp_time как пользователь postgres :

```
emp_time=# CREATE TABLE employee (  
emp_time(#      id SERIAL PRIMARY KEY,  
emp_time(#      username VARCHAR  
emp_time(# );  
CREATE TABLE  
emp_time=# CREATE TABLE time_punch (  
emp_time(#      id SERIAL PRIMARY KEY,  
emp_time(#      employee_id INT REFERENCES employee(id) NOT NULL,  
emp_time(#      is_out_punch BOOLEAN DEFAULT FALSE NOT NULL,  
emp_time(#      punch_time TIMESTAMP DEFAULT NOW() NOT NULL  
emp_time(# )  
emp_time-# ;  
CREATE TABLE
```

```
INSERT INTO employee (username) VALUES ('Михаил')
```

```
INSERT INTO time_punch (employee_id, punch_time, is_out_punch)  
VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 10:00:00',  
'false');  
INSERT INTO time_punch (employee_id, punch_time, is_out_punch)  
VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 11:30:00',  
'true');
```

```
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)  
emp_time-# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 10:00:00', 'false');  
INSERT 0 1  
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)  
emp_time-# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 11:30:00', 'true');  
INSERT 0 1
```

Query Query History

```
1 SELECT * FROM public.time_punch
2 ORDER BY id ASC
```

Data Output Messages Notifications

	id [PK] integer	employee_id integer	is_out_punch boolean	punch_time timestamp without time zone
1	1	1	false	2021-01-01 10:00:00
2	2	1	true	2021-01-01 11:30:00

Задание 2.

Проверка при корректных данных.

```
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)
emp_time=# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 12:00:00', 'false');
INSERT 0 1
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)
emp_time=# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 13:00:00', 'true');
INSERT 0 1
```

Проверка при некорректных данных

```
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)
emp_time=# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 14:00:00', 'false');
INSERT 0 1
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)
emp_time=# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 15:00:00', 'false');
INSERT 0 0
```

Узкие места: Поскольку триггер выполняется перед вставкой новой записи, существует риск возникновения блокировок в таблице "time_punch", особенно если много операций вставки происходит одновременно. Это может привести к задержкам и конфликтам при обновлении данных.

Время: Триггер не обрабатывает ошибки, если последний запрос выхода по времени раньше чем прошлый запрос входа:

```
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)
emp_time=# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 15:00:00', 'false');
INSERT 0 1
emp_time=# INSERT INTO time_punch (employee_id, punch_time, is_out_punch)
emp_time=# VALUES ((SELECT id FROM employee WHERE username = 'Михаил'), '2021-01-01 14:00:00', 'true');
INSERT 0 1
```

Данные:

7	8	1	false	2021-01-01 15:00:00
8	9	1	true	2021-01-01 14:00:00

Задание 4. Вариант 1

```
emp_time=# CREATE TABLE logs (
emp_time(#      text TEXT,
emp_time(#      added TIMESTAMP WITHOUT TIME ZONE
emp_time(# );
CREATE TABLE
```

Задание 5. Вариант 1

Протестируйте работу триггера, выполнив операции вставки, удаления и редактирования данных в таблице сотрудников.

```
emp_time=# INSERT INTO logs (text, added) VALUES ('Tst', '2024-03-04');
INSERT 0 1
emp_time=# INSERT INTO employee (id, username) VALUES (2, 'Jhon');
INSERT 0 1
emp_time=# UPDATE employee SET username = 'Jhon Do' WHERE id = '2';
UPDATE 1
emp_time=# DELETE FROM employee WHERE username = 'John Do';
DELETE 0
emp_time=# D_
```

```
emp_time=# DELETE FROM employee WHERE id = 2;
DELETE 1
```

Проверьте содержание таблицы логов после выполнения операций.

Query

Query History

1

SELECT * FROM public.logs

2

Data Output

Messages

Notifications

≡+

▼

▼

	<div>text</div> <div>text</div> <div></div>	<div>added</div> <div>timestamp without time zone</div> <div></div>
1	Add new user 2	2024-03-04 11:09:19.413052
2	Update user 2	2024-03-04 11:09:54.554071
3	Remove user 2	2024-03-04 11:11:43.144178