



Окидачи (тригери)

Пројектовање база података

Василије Тодоровић, Анђела Дамњановић, Александар Стефановић

12. април 2025.



Садржај

1 Окидачи



Садржај

1 Окидачи



Очување интегритета базе

- Приликом сваке измене у бази (додавање, брисање, ажурирање), потребно је очувати *интегритет базе*.
- Шта то значи?
 - Приликом додавања реда у табелу, потребно је да све вредности додатих атрибута буду унутар задатог типа и дозвољеног опсега. Примарни кључеви морају сви имати различите вредности.
 - Брисање реда у табели је дозвољено уколико његов примарни кључ није страни кључ неке друге табеле. Иначе се брисање одбија или се каскадно бришу одговарајуће n-торке из зависне табеле.
 - Као и у случају додавања, приликом ажурирања, нове вредности атрибута морају бити задатог типа и унутар дозвољеног опсега, примарни кључеви морају бити јединствени. Додатно, промене се каскадно примењују на остале релације.



Окидачи

- Поставља се питање како решити проблеме очувања интегритета базе. Одговор је - *окидачима*.
- Окидач је скуп акција које се извршавају аутоматски када се покуша промена базе (унос, брисање, ажурирање).
- Корисни су за валидацију улаза, чување логова, аутоматско генерисање вредности неког атрибута...
- Неке од главних предности које коришћење окидача пружа је бржи развој апликација и њихово лакше одржавање.



Врсте окидача

PostgreSQL омогућава рад са три врсте окидача:

- before - тело окидача се извршава пре операције
 - провера/модификација података који ће бити додати/ажурирани,
 - провера конзистентности два реда
- after - тело окидача се извршава након операције
 - пропaгација ажурирања другим табелама
 - провера конзистентности међу више табела
 - слање поруке
- instead of - окидачи над погледима (о њима ће бити више речи касније)



Креирање окидача у PostgreSQL-у

- Уколико се деси акција за коју је дефинисан окидач, позива се *тригер функција* која садржи списак наредби које је потребно извршити (у њој се налази тело окидача).
- Тригер функција мора увек бити дефинисана *пре* окидача и враћа тип *тригер* као повратну вредност.
- Параметри тригер функције се не наводе експлицитно, већ се прослеђеним аргументима (ако их има), приступа кроз TG_ARGV низ.
- Иста тригер функција може се користити за више окидача
- Ако постоји више окидача за исти догађај (унос, брисање или ажурирање) над истом табелом, они се извршавају *абecedним редом*.



Синтакса тригер функције

```
CREATE OR REPLACE FUNCTION ime_funkcije  
RETURNS TRIGGER AS $$  
BEGIN  
    telo_okidača  
END;  
$$ LANGUAGE plpgsql;
```



Тумачење синтаксе

CREATE OR REPLACE FUNCTION ime_funkcije - креира нову функцију са датим именом (или мења постојећу)
RETURNS TRIGGER AS \$\$ - враћа тип *тригер* као повратну вредност, \$\$ је делимитер који означава почетак функције
BEGIN - означава почетак логике у функцији
END; - означава крај логике у функцији
\$\$ LANGUAGE plpgsql - означава се крај функције и језик у коме је написана;



Синтакса окидача

```
CREATE [ OR REPLACE ] TRIGGER ime_okidača { BEFORE | AFTER }  
{ INSERT | DELETE | UPDATE | TRUNCATE } ON ime_tabele  
FOR EACH ROW  
WHEN ( condition )  
EXECUTE FUNCTION ime_triger_funkcije ( arguments )
```



Тумачење синтаксе окидача

CREATE [OR REPLACE] TRIGGER ime_okidača - прави се нови окидач са датим именом (или се замењује стари)
{ BEFORE | AFTER } - врста окидача
{ INSERT | DELETE | UPDATE | TRUNCATE } - за коју операцију се окидач дефинише
ON ime_tabele - над којом табелом се извршава операција
FOR EACH ROW - примени окидач на сваки ред који се мења
EXECUTE FUNCTION ime_triger_funkcije (arguments) - извршити тело наведене тригер функције



Литература

- Књига професорке Гордане Павловић-Лажетић
- О окидачима са званичног `postgresql` сајта
- О креирању окидача, са званичног `postgresql` сајта