ტრიგერები

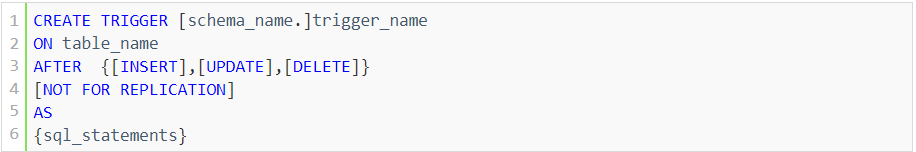
Sql ში ტრიგერი არის პროცედურის განსაკუთრებული ტიპი, რომელიც ეშვება ავტომატურად როგორიცაა ბაზის ობიექტის ბაზის და სერვერის ივენთების შესაბამისად. არსებობს სამი ტიპის ტრიგერი:

* მონაცემთა მანიპულირების ენის (DML) ტრიგერი, რომელიც გამოიძხება ავტომატურად INSERT UPDATE და DELETE ივენთების საპასუხოდ.
* მონაცემთა განმარტების ენის (DDL) ტრიგერები რომელიც გამოიძახება CREATE ALTER და DROP ივენთების დროს. ისინი აგრეთვე გამოიძახება მაშინ როდესაც ეშვება რაიმე სისტემური პროცედურა რომელიც ასრულებს DDL -ის მსგავს ოპერაციებს.
* Logon ტრიგერები რომლებიც ეშვებიან LOGON ივენთის საპასუხოდ.

განვიხილოთ თითეული მათგანი მაგალითს საფუძველზე:

CREATE TRIGGER

CREATE TRIGGER იძლევა საშუალებას შევქმნათ ახალი ტრიგერი INSERT UPDATE და DELETE ივენთების აღძვრის შესაბამისად. მისი ზოგადი სინტაქსი ასეთია :



სადაც schema\_name არის სქემის სახელწოდება, რომელიც არასავალდებულო ველია trigger\_name ტრიგერის სახელი table\_name იმ ცხრილის სახელი, რომელზეც უნდა დავადოთ ტრიგერი , ტიპი(**insert** , **update**, **delete**) და [**NOT FOR REPLICATION**] ატრიბუტი, რომელიც შეიძლება ტრიგერს დავადოთ მაშინ, როცა მონაცემთა ცვლილება მოხდება რეპლიკაციის დროს.

ტრიგერის გაშვების დროს იქმენა ორი ვირტუალური ტიპის ცხრილი INSERTED და UPDATED რომელშიც იყრება მონაცემები row ცვლილების დროს ან შემდეგ.

დიაგრამაზე ნაჩვენებია ეს პროცესი როდის რომელ ცხრილში იყრება მონაცემები.

| **DML event** | **INSERTED table holds** | **DELETED table holds** |
| --- | --- | --- |
| INSERT | rows to be inserted | empty |
| UPDATE | new rows modified by the update | existing rows modified by the update |
| DELETE | empty | rows to be deleted |

განვიხილოთ ტრიგერის შექმნის მაგალითი რეალური ბაზის მაგალითზე. შემდეგში ტრიგერის ყველა ვარიანტს განვიხილავთ ამ ბაზაზე.

შევქმნათ სატესტო ბაზა და ცხრილები.

ბაზაში გვაქვს სხვადასხვა ცხრილები. ერთ ერთია სკოლის ინვენტარი, რომლის შესახებაც საჭიროა შეინახოს ყველანაირი ინფორმაცია ცვლილებების შემთხვევაში. ამისათვის შემოვიღეთ დამატებით ცხრილი, რომელიც ინახავს ლოგებს. ახლა შევქმნათ DML ტრიგერი

ჩღ