Salesforce Apex Trigger

Sounds like a plan

Let's get started

Dhananjay Aher

dj_aher_131997

Agenda

Introduction

Trigger Syntax

Types of Triggers

Trigger events in salesforce?

considerations while implementing the Triggers

context variables in triggers

Context Variable Considerations

Apex Trigger Best Practices

Introduction

Apex triggers enable you to perform custom actions before or after events to records in Salesforce, such as insertions, updates, or deletions. Just like database systems support triggers, Apex provides trigger support for managing records.

Typically, you use triggers to perform operations based on specific conditions, to modify related records or restrict certain operations from happening. You can use triggers to do anything you can do in Apex, including executing SOQL and DML or calling custom Apex methods.

Use triggers to perform tasks that can't be done by using the point-and-click tools in the Salesforce user interface.

Trigger Syntax

A trigger definition starts with the trigger keyword. It is then followed by the name of the trigger, the Salesforce object that the trigger is associated with, and the conditions under which it fires.

```
trigger TriggerName on ObjectName (trigger_events) {
code_block

}
```

Back to Agenda Page

Types of Triggers

- Before triggers are used to update or validate record values before they’re saved to the database.

- After triggers are used to access field values that are set by the system (such as a record's Id or LastModifiedDate field), and to affect changes in other records. The records that fire the after trigger are read-only.

Back to Agenda Page

Trigger events in salesforce?

A trigger is a set of statement which can be executed on the following events. In above trigger events one or more of below events can be used with comma-separated.

- before insert
- before update
- before delete
- after insert
- after update
- after delete
- after undelete

considerations while implementing the Triggers

- Upsert trigger fires on 4 different events :- before(insert, update), after (insert, update)
- Merge trigger are fired on both events on delete
- Field history is updated after the trigger has successfully finished processing data.
- Any callout should be asynchronous so that trigger does not have to wait for the response.
- A trigger cannot have a static keyword in its code.
- If a trigger completes successfully the changes are committed to the database and if it fails the transaction is rolled back.

Context variables in triggers

- isExecuting: Returns true if the current context for the Apex code is a trigger, not a Visualforce page, a Web service, or an executeanonymous() API call.
- isInsert: Returns true if this trigger was fired due to an insert operation, from the Salesforce user interface, Apex, or the API.
- isUpdate: Returns true if this trigger was fired due to an update operation, from the Salesforce user interface, Apex, or the API.
- isDelete: Returns true if this trigger was fired due to a delete operation, from the Salesforce user interface, Apex, or the API.
- isBefore: Returns true if this trigger was fired before any record was saved.
- isAfter: Returns true if this trigger was fired after all records were saved.

Context variables in triggers

- isUndelete: Returns true if this trigger was fired after a record is recovered from the Recycle Bin (that is, after an undelete operation from the Salesforce user interface, Apex, or the API.)
- new: Returns a list of the new versions of the sObject records. This sObject list is only available in insert, update, and undelete triggers, and the records can only be modified in before triggers.
- newMap: A map of IDs to the new versions of the sObject records. This map is only available in before update, after insert, after update, and after undelete triggers.
- old : Returns a list of the old versions of the sObject records. This sObject list is only available in update and delete triggers.
- oldMap: A map of IDs to the old versions of the sObject records. This map is only available in update and delete triggers.
- size: The total number of records in a trigger invocation, both old and new.

Context Variable Considerations

Trigger Event	Can change fields using trigger.new	Can update original object using an update DML operation	Can delete original object using a delete DML operation
before insert	Allowed.	Not applicable. The original object has not been created; nothing can reference it, so nothing can update it.	Not applicable. The original object has not been created; nothing can reference it, so nothing can update it.
after insert	Not allowed. A runtime error is thrown, as trigger.new is already saved.	Allowed.	Allowed, but unnecessary. The object is deleted immediately after being inserted.
before update	Allowed.	Not allowed. A runtime error is thrown.	Not allowed. A runtime error is thrown.
after update	Not allowed. A runtime error is thrown, as trigger.new is already saved.	Allowed. Even though bad code could cause an infinite recursion doing this incorrectly, the error would be found by the governor	Allowed. The updates are saved before the object is deleted, so if the object is undeleted, the updates become visible.

Back to Agenda Page

Context Variable Considerations

before delete	Not allowed. A runtime error is thrown. trigger.new is not available in before delete triggers.	Allowed. The updates are saved before the object is deleted, so if the object is undeleted, the updates become visible.	Not allowed. A runtime error is thrown. The deletion is already in progress.
after delete	Not allowed. A runtime error is thrown. trigger.new is not available in after delete triggers.	Not applicable. The object has already been deleted.	Not applicable. The object has already been deleted.
after undelete	Not allowed. A runtime error is thrown.	Allowed.	Allowed, but unnecessary. The object is deleted immediately after being inserted.

Back to Agenda Page

Apex Trigger Best Practices

Bulkify Your Code

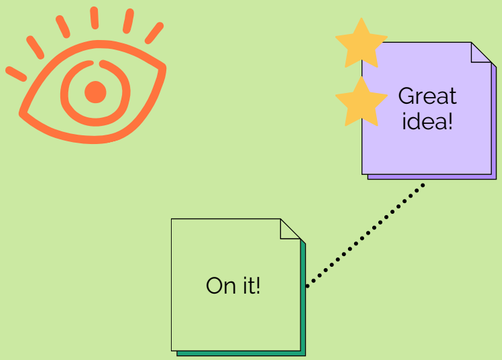
Avoid SOQL Queries or DML statements inside FOR Loops

Bulkify Helper Methods

Use Collections

Back to Agenda Page

Salesforce Apex



Dhananjay Aher