Deloitte. Digital

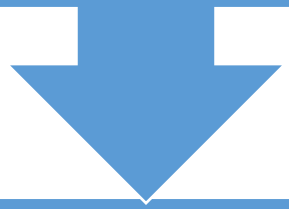
salesforce

APEX TRIGGERS

5th April,2022

What is Apex Triggers?

Apex Trigger is a piece of code that executes whenever we are performing any DML operation into sObject.



DML may be

Insert	Update	Delete	Undelete

When to use Apex Trigger?

You should use trigger to perform tasks that can't be done using the point-and-click tools in the Salesforce interface.

You can use trigger to:

- Perform operations based on specific conditions
- Modify related records
- Restrict certain operations from happening

Exercise

Create a number field on Account object – Rating. You want to restrict a user from entering a value in this field between 1-5 only. If a user enters a value apart from 1-5, then display an error “You cannot enter rating less than 1 and more than 5”.

Create a new picklist field on Account object - Review with values Bad, Average, Good. If Rating is 1 or 2, then Review should be Bad, If rating is 3 then Review should be populated as Average and if rating is 4 or 5, then Review should be populated as Good.

Prevent users from creating Account with same Name and Rating in Salesforce. If an account with same name and rating already exists, then display error “You cannot create Duplicate Account”

Trigger Syntax

- The syntax of a trigger definition is different from a class definition's 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.
- A trigger has the following syntax:

```
trigger <name> on ObjectName (<events>) {
        code_block
}
```

Trigger Types and Events

- Types of Triggers:
  - **Before Triggers** – Used to update or validate record values before they are saved in database.
    - Example - (Exercise on slide number 4)
  - **After Triggers** – Used to access field values that are set by the system (Id or LastModifiedDate), and to affect changes in other records. The record that fires the after trigger are read-only
    - Example – If a Account has Rating as 4 or 5, then create a contact record for that account with LastName as Account's last name.

Trigger Events

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

Context Variables

Variable	Usage
isInsert	Returns true if this trigger was fired due to an insert operation
isUpdate	Returns true if this trigger was fired due to an update operation
isDelete	Returns true if this trigger was fired due to a delete operation
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.
isUndelete	Returns true if this trigger was fired after a record is recovered from the Recycle Bin.
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.

Trigger Exceptions

- Triggers can be used to prevent DML operations from occurring by calling the addError() method on a record or field.
- We can add custom error message in addError to display it to the user.
- Trigger.addError() can be used on :
  - Trigger.new records in insert/update triggers
  - Trigger.old records in delete triggers
- Syntax :

```
(record).addError(errorMsg);
```

- Example :

```
Trigger.new[0].addError('You can't delete this record');
```


Exercise

1

Prevent users from creating Account with same Name and Rating in Salesforce. If an account with same name and rating already exists, then display error “You cannot create Duplicate Account”

2

If an Account has Rating as 4 or 5, then create a contact record for that account with LastName as Account’s last name.

3

If an Account has rating 1 or 2, then don’t allow the user to delete that Account record. If the user tries to delete the Account record, display an error stating “You can’t delete an Account with bad review”

Bulk Triggers

- Typically, triggers operate on one record if the action that fired the trigger originates from the user interface.
- But if the origin of the action was bulk DML or the API, the trigger operates on a record set rather than one record.
- For example, when you import many records via the API, triggers operate on the full record set
- Apex triggers are optimized to operate in bulk
- When you use bulk design patterns, your triggers
  - have better performance
  - consume less server resources
  - are less likely to exceed platform limits

Bulk Triggers

- Non-bulkified trigger :

```
trigger MyTriggerNotBulk on Account(before insert) {
    Account a = Trigger.New[0];
    a.Description = 'New description';
}
```

- Bulkified trigger :

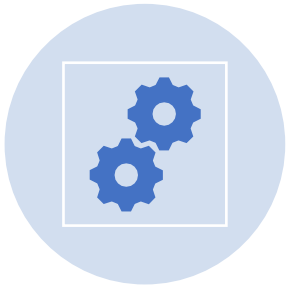
```
trigger MyTriggerNotBulk on Account(before insert) {
     for(Account a : Trigger.New) {
        a.Description = 'New description';
    }
}
```


Trigger Best Practices

- One Trigger per object
- Logic-less Triggers
- Bulkify your code
- Make use of collections
- Avoid SOQL Queries or DML statements inside FOR Loops
- Execute DML statements using collections instead of individual records per DML statement
- Use a consistent naming convention including the object name (e.g., AccountTrigger)

Exercise

Bulkify your triggers created in slide number 10.



Follow all the best practices for triggers and update your trigger code.

Deloitte. Digital

Thank You

For any queries, please reach out to :

Pooja Chordia
Manager | DCM : Digital Customer
Mobile: +91 8087998469
Email : pochordia@deloitte.com

