Apex Triggers

Apex Triggers

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.
A trigger is Apex code that executes before or after the following types of operations:
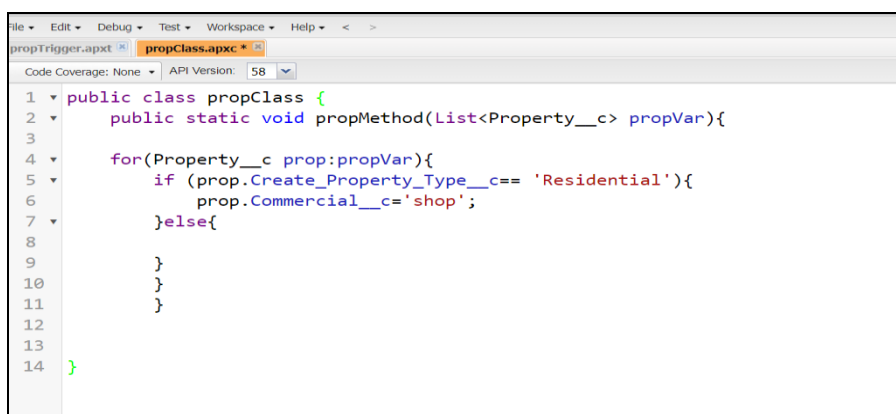
- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

Use Case: Apex Trigger is Related to Property Object in that there is the field "Create Property Type" which is having the picklist values in that field(Residential, Commercial, Industrial) the condition is like if we select the Create Property type as "Residential" than there is Commercial field so it should get populated with "Shop"

Apex Class:


```
public class propClass {
    public static void propMethod(List<Property__c> propVar){

    for(Property__c prop:propVar){
        if (prop.Create_Property_Type__c== 'Residential'){
            prop.Commercial__c='shop';
        }else{

        }
        }
        }

}
```

public static void propMethod(List<Property__c> propVar){
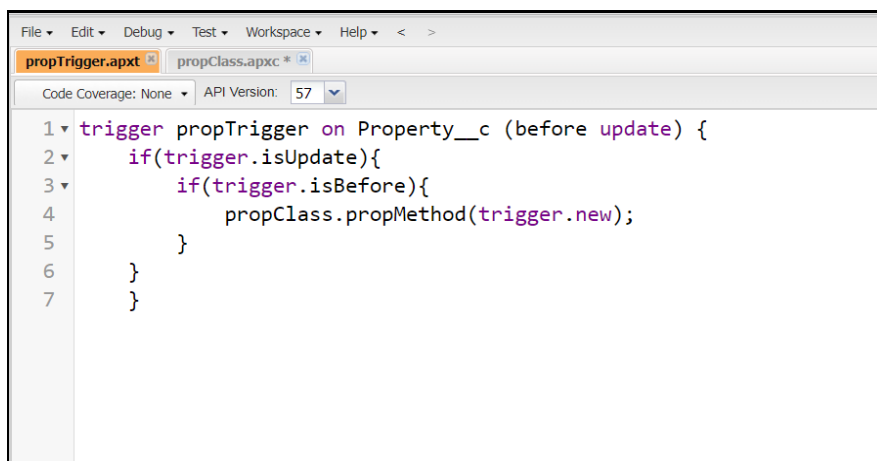
```
public class propClass {
    for(Property__c prop:propVar){
        if (prop.Create_Property_Type__c== 'Residential'){
            prop.Commercial__c='shop';
        }else{

        }
    }
}


}
```

Apex Triggers:


```
trigger propTrigger on Property__c (before update) {
    if(trigger.isUpdate){
        if(trigger.isBefore){
propClass.propMethod(trigger.new);
        }
    }
}
```

- Just Go to the Property Object and Check Whether Your Trigger Is Working or not as per the requirement