**Phase 5 – Apex Programming (Developer)**

**Project:** Smart Internship & Placement Management System  
**Domain:** Salesforce CRM | Education & Recruitment

**Objective**

To prevent students from applying for the same internship more than once, a trigger PreventDuplicateApplication is implemented on the Application\_\_c object. This ensures data integrity and enforces the business rule that a student can submit only one application per internship.

**1. Apex Trigger: PreventDuplicateApplication**

**Object:** Application\_\_c  
**Trigger Event:** before insert

**Trigger Code:**

trigger PreventDuplicateApplication on Application\_\_c (before insert) {

Set<String> existingPairs = new Set<String>();

Set<Id> studentIds = new Set<Id>();

for (Application\_\_c app : Trigger.new) {

if (app.Student\_\_c != null && app.Internship\_\_c != null) {

studentIds.add(app.Student\_\_c);

}

}

if (!studentIds.isEmpty()) {

for (Application\_\_c a : [

SELECT Student\_\_c, Internship\_\_c

FROM Application\_\_c

WHERE Student\_\_c IN :studentIds

]) {

if (a.Student\_\_c != null && a.Internship\_\_c != null) {

existingPairs.add(a.Student\_\_c + '-' + a.Internship\_\_c);

}

}

}

for (Application\_\_c app : Trigger.new) {

if (app.Student\_\_c != null && app.Internship\_\_c != null) {

String key = app.Student\_\_c + '-' + app.Internship\_\_c;

if (existingPairs.contains(key)) {

app.addError('Student has already applied for this Internship');

} else {

existingPairs.add(key);

}

}

}

}

**2. Procedure to Implement Trigger**

**Step 1: Open Developer Org**

* Log in to your Salesforce Developer Org.
* Navigate to **Setup → Object Manager → Application\_\_c → Triggers**.

**Step 2: Create New Trigger**

* Click **New Trigger**.
* Name: PreventDuplicateApplication
* Object: Application\_\_c
* Event: before insert

**Step 3: Paste Apex Code**

* Copy the trigger code provided above.
* Save the trigger.

**Step 4: Verify Trigger Logic**

* The trigger automatically checks for duplicate applications whenever a new Application\_\_c record is inserted.
* If a student tries to apply for the same internship again, the record will **fail to save** with an error:  
  "Student has already applied for this Internship"

**Step 5: Check Trigger Execution Flow**

1. Salesforce detects a new Application\_\_c record insertion.
2. Trigger runs **before saving the record**.
3. Collects all student IDs from incoming applications.
4. Queries the database to check existing applications for those students.
5. Creates a **unique key** (Student\_\_c + '-' + Internship\_\_c) for comparison.
6. If a duplicate key is found, the trigger prevents the record insertion with app.addError().

**3. Trigger Functionality Explained**

* **Purpose:** Prevent duplicate applications per student-internship pair.
* **Key Logic:** Combines student and internship IDs into a unique key, checks existing records, and prevents duplicates.
* **Event Type:** Runs **before insert**, ensuring no duplicate record is saved.
* **Bulk-Safe:** Supports multiple records inserted in a single operation.

**4. Key Benefits**

* **Data Integrity:** Maintains uniqueness of student applications.
* **Automation:** Eliminates manual checks for duplicates.
* **Efficiency:** Lightweight and bulk-safe.
* **Simplicity:** Fully functional in Developer Org without additional automation.  
    
    
    
    
    
    
    
    
  