Library Management :- Phase 5: Apex Programming (Developer)

- ➢ Classes & Objects:

- Created an Apex utility class **TeacherService:**
- Inner Comparator: **TeacherHireDateComparator** – sorts teachers by **Hire_Date__c.**

- ➢ Static Method:

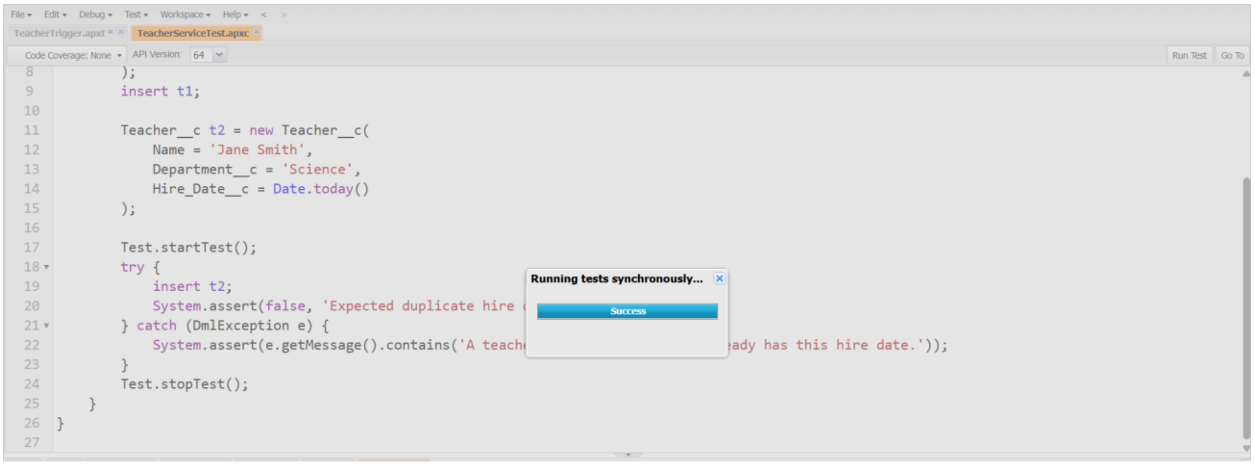
- **preventDuplicateHireDates(List<Teacher_c> incoming, Map<Id, Teacher_c> oldMap)** – checks new/updated teacher records against existing teacher records to ensure no conflicting hire dates for the same department (or other criteria) and uses **addError()** to block duplicates.

File → New → Apex Class.

Name it TeacherServiceTest → click OK.

Paste test code:

Save.



```
        );
        insert t1;

        Teacher__c t2 = new Teacher__c(
            Name = 'Jane Smith',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );

        Test.startTest();
        try {
            insert t2;
            System.assert(false, 'Expected duplicate hire 
        } catch (DmlException e) {
            System.assert(e.getMessage().contains('A teach                                        ady has this hire date.'));
        }
        Test.stopTest();
    }
}
```

2.Apex Trigger

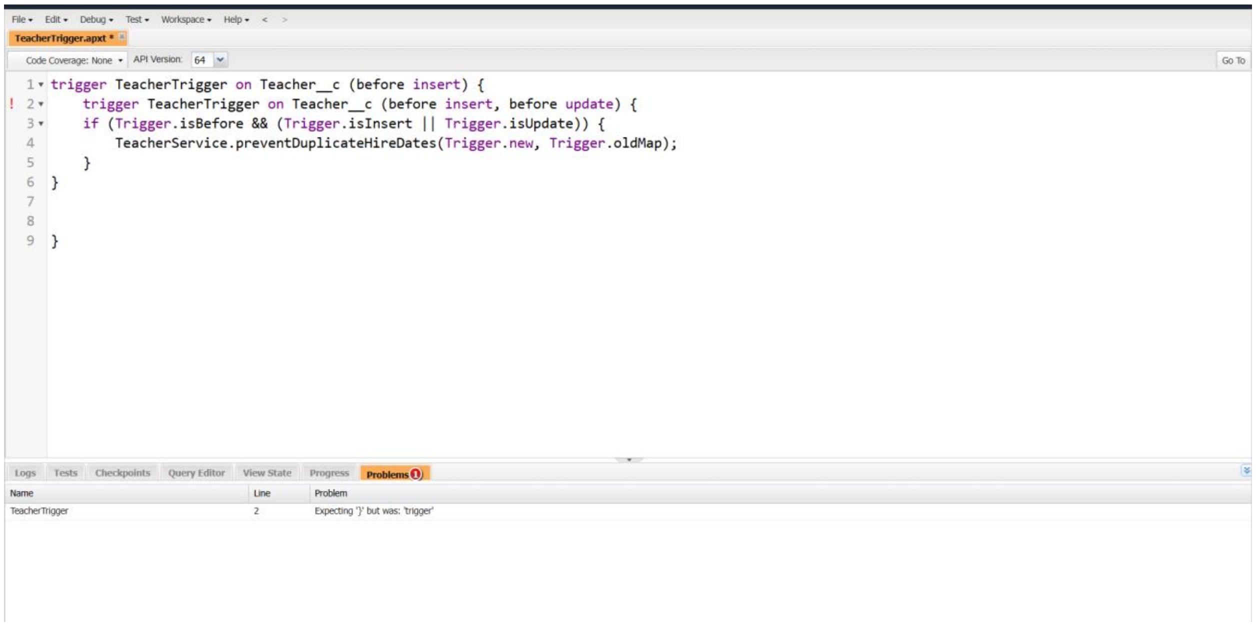
File → New → Apex Trigger.

Enter:

- Name: TeacherTrigger
- sObject: Teacher__c

Click Submit.

Save.



TeacherTrigger.apxt *

Code Coverage: None · API Version: 64

```
trigger TeacherTrigger on Teacher__c (before insert) {
    trigger TeacherTrigger on Teacher__c (before insert, before update) {
    if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
        TeacherService.preventDuplicateHireDates(Trigger.new, Trigger.oldMap);
    }
}


}
```

Name	Line	Problem
TeacherTrigger	2	Expecting '}' but was: 'trigger'

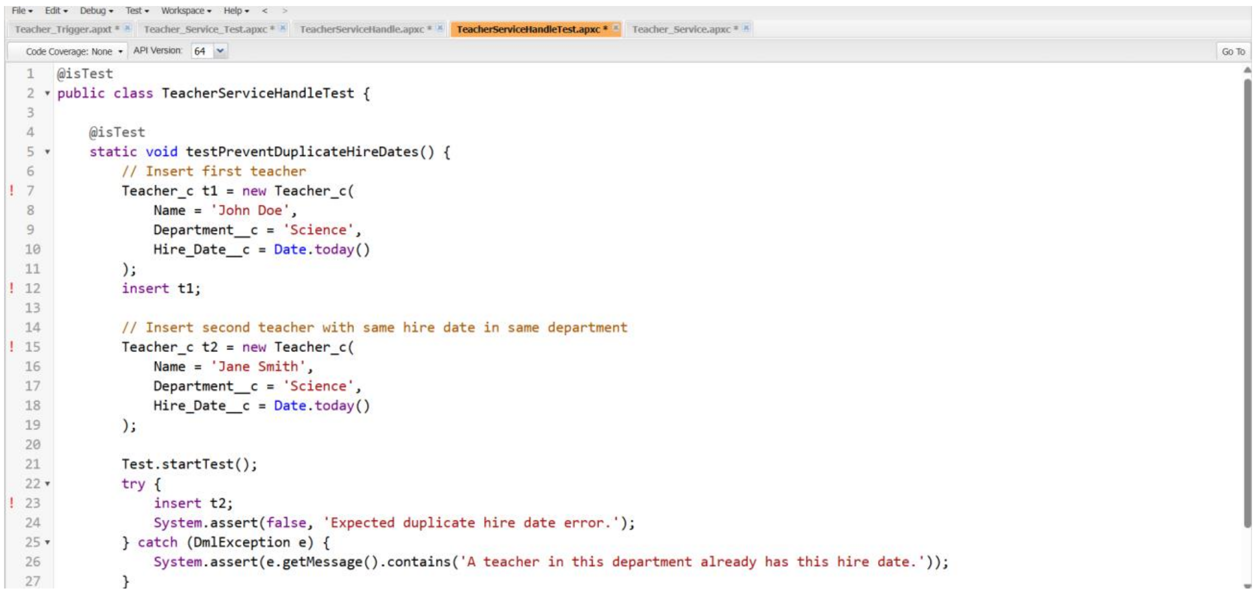
3.Trigger Design Pattern:

Business logic is kept in the TeacherService class; the trigger on Teacher__c simply calls the class method.

4. SOQL & Collections

Inside preventOverlap:

- SOQL queries existing Teacher for relevant new Teacher.


File Edit Debug Test Workspace Help
Teacher_Trigger.apxt * Teacher_Service_Test.apxc * TeacherServiceHandle.apxc * TeacherServiceHandleTest.apxc * Teacher_Service.apxc *
Go To

Code Coverage: None API Version: 64

```
@isTest
public class TeacherServiceHandleTest {

    @isTest
    static void testPreventDuplicateHireDates() {
        // Insert first teacher
        Teacher_c t1 = new Teacher_c(
            Name = 'John Doe',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );
        insert t1;

        // Insert second teacher with same hire date in same department
        Teacher_c t2 = new Teacher_c(
            Name = 'Jane Smith',
            Department__c = 'Science',
            Hire_Date__c = Date.today()
        );

        Test.startTest();
        try {
            insert t2;
            System.assert(false, 'Expected duplicate hire date error.');
        } catch (DmlException e) {
            System.assert(e.getMessage().contains('A teacher in this department already has this hire date.'));
        }
```

5. Control Statements

for loops, if conditions and addError() to prevent overlaps.

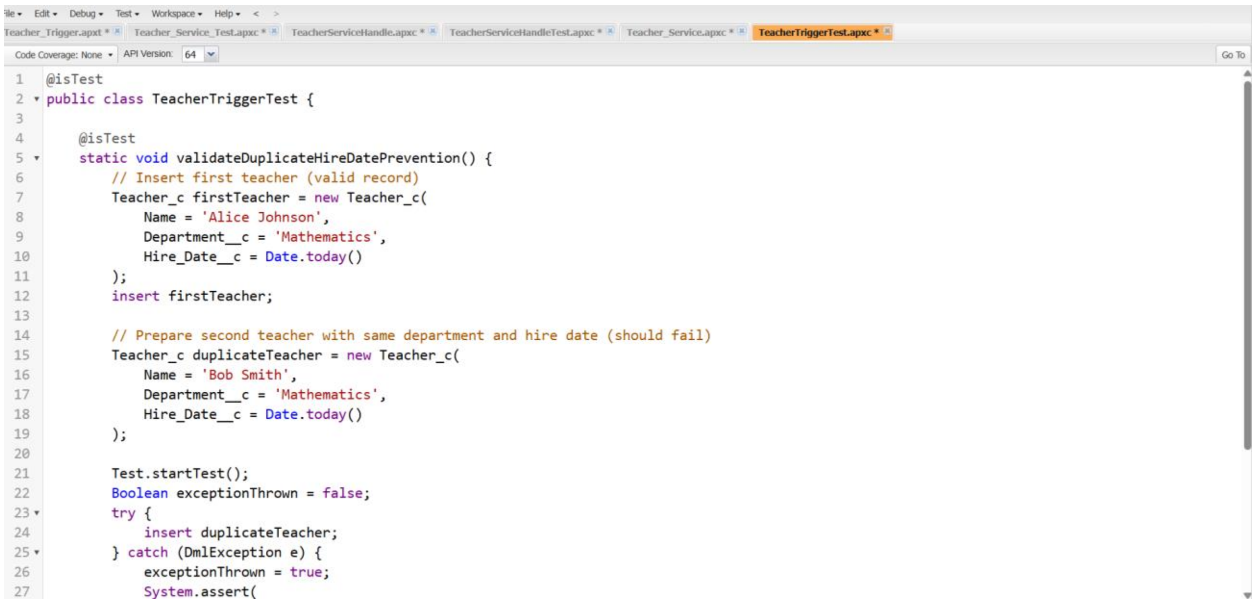
6. Test Classes (Teacher Example)

A test class **TeacherTriggerTest** was created to:

- Insert a valid teacher (no duplicate hire date in the same department) and verify it saves successfully.
- Attempt to insert a teacher with a duplicate hire date in the same department and verify a **DmlException** with “already has this hire date” in the message is thrown.

This test indirectly executes the logic in

TeacherServiceHandle.preventDuplicateHireDates and gives code coverage for both the trigger (**TeacherTrigger**) and the service class (**TeacherServiceHandle**).

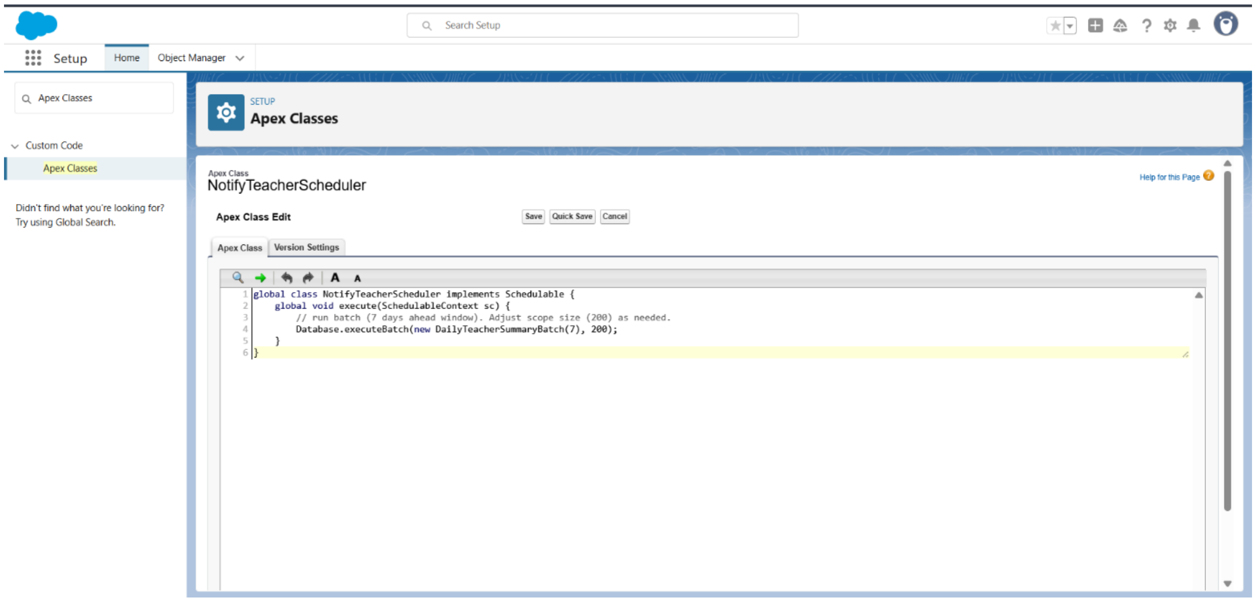

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
Teacher_Trigger.apxt * Teacher_Service_Test.apxc * TeacherServiceHandle.apxc * TeacherServiceHandleTest.apxc * Teacher_Service.apxc * TeacherTriggerTest.apxc *
Code Coverage: None ▾ API Version: 64
Go To

```
@isTest
public class TeacherTriggerTest {

    @isTest
    static void validateDuplicateHireDatePrevention() {
        // Insert first teacher (valid record)
        Teacher_c firstTeacher = new Teacher_c(
            Name = 'Alice Johnson',
            Department__c = 'Mathematics',
            Hire_Date__c = Date.today()
        );
        insert firstTeacher;

        // Prepare second teacher with same department and hire date (should fail)
        Teacher_c duplicateTeacher = new Teacher_c(
            Name = 'Bob Smith',
            Department__c = 'Mathematics',
            Hire_Date__c = Date.today()
        );

        Test.startTest();
        Boolean exceptionThrown = false;
        try {
            insert duplicateTeacher;
        } catch (DmlException e) {
            exceptionThrown = true;
            System.assert(
```

7. **Batch Apex and Scheduled Jobs**

A Batch Apex class TeacherSummaryBatch was created to:

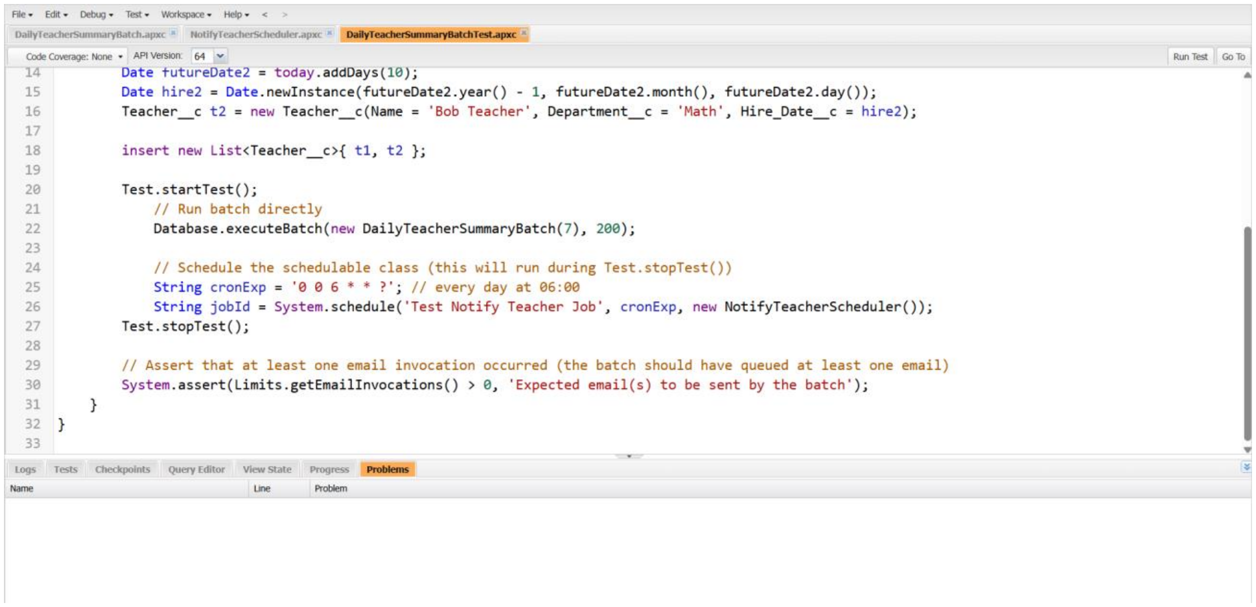
- Query recently hired teachers by department.
- Generate and log summary reports (or prepare reminders for admin review).
- Run daily at 6:00 AM using a Scheduled Apex job.



```
global class NotifyTeacherScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        // run batch (7 days ahead window). Adjust scope size (200) as needed.
        Database.executeBatch(new DailyTeacherSummaryBatch(7), 200);
    }
}
```

8.Exception Handling

Used **addError()** on SObjects to stop DML with a user-friendly message.



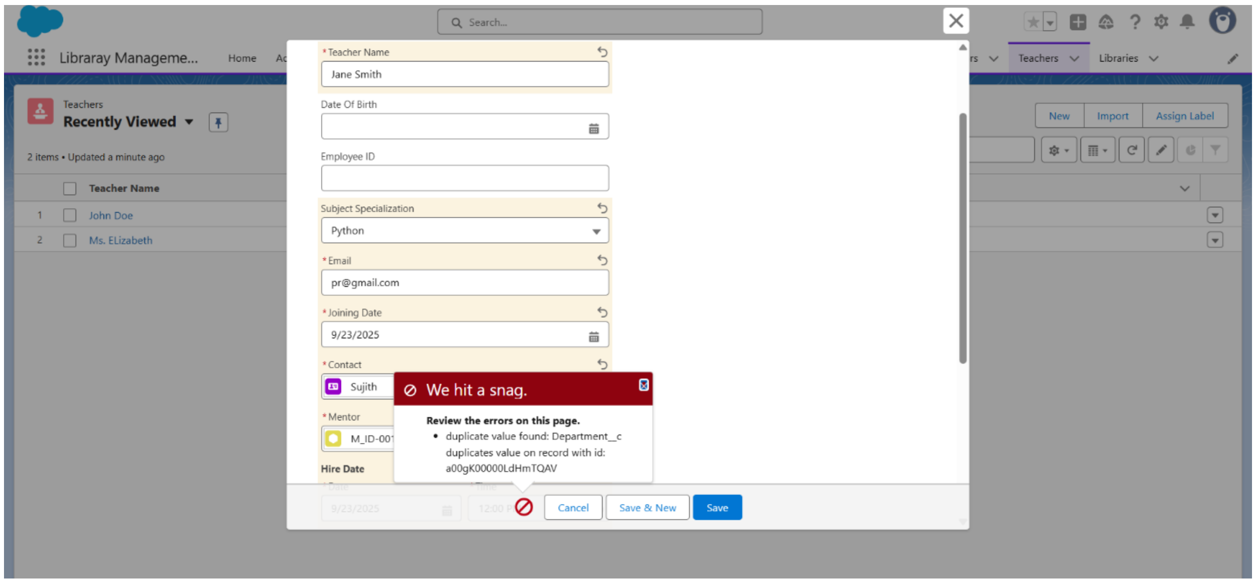
File Edit Debug Test Workspace Help
DailyTeacherSummaryBatch.apxc
NotifyTeacherScheduler.apxc
DailyTeacherSummaryBatchTest.apxc
Code Coverage: None
API Version: 64
Run Test
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
        Date futureDate2 = today.addDays(10);
        Date hire2 = Date.newInstance(futureDate2.year() - 1, futureDate2.month(), futureDate2.day());
        Teacher__c t2 = new Teacher__c(Name = 'Bob Teacher', Department__c = 'Math', Hire_Date__c = hire2);

        insert new List<Teacher__c>{ t1, t2 };

        Test.startTest();
            // Run batch directly
            Database.executeBatch(new DailyTeacherSummaryBatch(7), 200);

            // Schedule the schedulable class (this will run during Test.stopTest())
            String cronExp = '0 0 6 * * ?'; // every day at 06:00
            String jobId = System.schedule('Test Notify Teacher Job', cronExp, new NotifyTeacherScheduler());
        Test.stopTest();

        // Assert that at least one email invocation occurred (the batch should have queued at least one email)
        System.assert(Limits.getEmailInvocations() > 0, 'Expected email(s) to be sent by the batch');
    }
}
```

9.Skipped / Not Implemented

Queueable Apex, Future Methods, Asynchronous Processing beyond the daily batch job were not implemented at this phase.

10. **Testing & Results**



* Teacher Name	Jane Smith
Date Of Birth	
Employee ID	
Subject Specialization	Python
* Email	pr@gmail.com
* Joining Date	9/23/2025
* Contact	Sujith
* Mentor	M_ID-001
Hire Date	9/23/2025 12:00 P

We hit a snag.

Review the errors on this page.

- duplicate value found: Department__c duplicates value on record with id: a00gK00000LdHmTQAV