Apex Specialist Superbadge

Challenge 1

Trigger Maintenance Request

```
trigger MaintenanceRequest on Case (before update, after update) {

    Map<Id,Case> validCaseMap = new Map<Id,Case>();

    if(Trigger.isUpdate  && Trigger.isAfter){
        for(Case caseHere: Trigger.new){
            if (caseHere.IsClosed && (caseHere.Type.equals('Repair') ||
caseHere.Type.equals('Routine Maintenance'))){
                validCaseMap.put(caseHere.Id, caseHere);
            }
        }

        if(!validCaseMap.values().isEmpty()){
               MaintenanceRequestHelper.createNewRequest(validCaseMap);
        }
    }

}
```

Maintenance Request Helper Class

```
public class MaintenanceRequestHelper {


    public static void createNewRequest(Map<Id, Case> validCaseMap){
        List<Case> newCases = new List<Case>();
        Map<Id, Integer> productMaintenanceCycleMap = new Map<Id, Integer>();
        Map<Id, Integer> workPartMaintenanceCycleMap = new Map<Id, Integer>();


                              for (Product2 productHere : [select Id, Maintenance_Cycle__c from Product2]) {
            if (productHere.Maintenance_Cycle__c != null) {
                productMaintenanceCycleMap.put(productHere.Id,
Integer.valueOf(productHere.Maintenance_Cycle__c));
            }
        }


        for (Work_Part__c workPart : [select Id, Equipment__c, Maintenance_Request__c from
Work_Part__c where Maintenance_Request__c in :validCaseMap.keySet()]) {
            if (workPart.Equipment__c != null) {
                if(!workPartMaintenanceCycleMap.containsKey(workPart.Maintenance_Request__c)){
                    workPartMaintenanceCycleMap.put(workPart.Maintenance_Request__c,
productMaintenanceCycleMap.get(workPart.Equipment__c));
                }
                else if(productMaintenanceCycleMap.get(workPart.Equipment__c) <
workPartMaintenanceCycleMap.get(workPart.Maintenance_Request__c)){
                    workPartMaintenanceCycleMap.put(workPart.Maintenance_Request__c,
productMaintenanceCycleMap.get(workPart.Equipment__c));
                }
            }
        }
      }

        for(Case caseHere: validCaseMap.values()){
            Case newCase = new Case();
            newCase.Vehicle__c = caseHere.Vehicle__c;
            newCase.Equipment__c = caseHere.Equipment__c;
            newCase.Type = 'Routine Maintenance';
            newCase.Subject = String.isBlank(caseHere.Subject) ? 'Routine Maintenance Request' :
caseHere.Subject + ' New';
            newCase.Date_Reported__c = Date.today();
            newCase.Date_Due__c =
workPartMaintenanceCycleMap.containsKey(caseHere.Product__c) ?
Date.today().addDays(workPartMaintenanceCycleMap.get(caseHere.Product__c)) :
Date.today();
            newCase.Status = 'New';
            newCase.Product__c = caseHere.Product__c;
            newCase.AccountId = caseHere.AccountId;
            newCase.ContactId = caseHere.ContactId;
            newCase.AssetId = caseHere.AssetId;
            newCase.Origin = caseHere.Origin;
            newCase.Reason = caseHere.Reason;

            newCases.add(newCase);
        }

        if(newCases.size() > 0){
            insert newCases;
        }
    }

}
```

Challenge 2

WarehouseCalloutService Class

```
public with sharing class WarehouseCalloutService {

    private static final String WAREHOUSE_URL = 'https://th-superbadge-apex.herokuapp.com/equipment';

    // complete this method to make the callout (using @future) to the
    // REST endpoint and update equipment on hand.
    @future(callout=true)
    public static void runWarehouseEquipmentSync(){
        Http http = new Http();
        HttpRequest request = new HttpRequest();
        request.setEndpoint(WAREHOUSE_URL);
        request.setMethod('GET');
        HttpResponse response = http.send(request);

        if (response.getStatusCode() == 200) {
            List<Object> results = (List<Object>) JSON.deserializeUntyped(response.getBody());
            List<Product2> equipmentList = new List<Product2>();

            for (Object record: results) {
                Map<String, Object> recordMap = (Map<String, Object>)record;
                Product2 equipment = new Product2();
                equipment.Name = (String)recordMap.get('name');
                equipment.Cost__c = (Decimal)recordMap.get('cost');
                equipment.ProductCode = (String)recordMap.get('_id');
                equipment.Current_Inventory__c = (Integer)recordMap.get('quantity');
                equipment.Maintenance_Cycle__c = (Integer)recordMap.get('maintenanceperiod');
                equipment.Replacement_Part__c = (Boolean)recordMap.get('replacement');
                equipment.Lifespan_Months__c = (Integer)recordMap.get('lifespan');
                equipment.Warehouse_SKU__c = (String)recordMap.get('sku');

                equipmentList.add(equipment);
            }

            if(equipmentList.size() > 0){
                upsert equipmentList;
            }
        }

    }
}
```

Challenge 3

WarehouseSyncSchedule Class

```
public class WarehouseSyncSchedule implements Schedulable{
 // implement scheduled code here
  public void execute(System.SchedulableContext context){
    WarehouseCalloutService.runWarehouseEquipmentSync();
  }
}
```

Challenge 4

MaintenanceRequestTest Class

```
@isTest
public class MaintenanceRequestTest {

  @testSetup
  static void setup(){
    Product2 prod = new Product2();
    prod.Cost__c = 50;
    prod.Name = 'Ball Valve 10 cm';
    prod.Lifespan_Months__c = 12;
    prod.Maintenance_Cycle__c = 365;
    prod.Current_Inventory__c = 50;
    prod.Replacement_Part__c = true;
    prod.Warehouse_SKU__c = '100009';
```

```
insert prod;

Product2 prod2 = new Product2();
prod2.Cost__c = 50;
prod2.Name = 'Ball Valve 10 cm';
prod2.Lifespan_Months__c = 12;
prod2.Maintenance_Cycle__c = 240;
prod2.Current_Inventory__c = 50;
prod2.Replacement_Part__c = true;
prod2.Warehouse_SKU__c = '100009';
insert prod2;

List<Case> caseList = new List<Case>();
for(Integer i=0; i<300; i++) {
    Case caseNew = new Case();
    caseNew.Subject = 'Maintenance ' + i;
    caseNew.Type = 'Other';
    caseNew.Status = 'New';
    caseNew.Equipment__c = prod.Id;
    caseNew.SuppliedName = 'Test';
    caseList.add(caseNew);

    if(i==10){
        caseNew.Subject = 'Maintenance test 10';
    }
}

insert caseList;
```

```
    List<Work_Part__c> workPartList = new List<Work_Part__c>();
    for(Case caseHere : [select Id, Subject from Case where SuppliedName = 'Test']) {
        Work_Part__c workPart = new Work_Part__c();
        workPart.Maintenance_Request__c = caseHere.Id;
        workPart.Equipment__c = prod.Id;
        workPartList.add(workPart);

        if(caseHere.Subject == 'Maintenance test 10'){
            Work_Part__c workPart2 = new Work_Part__c();
            workPart2.Maintenance_Request__c = caseHere.Id;
            workPart2.Equipment__c = prod2.Id;
            workPartList.add(workPart2);
        }
    }

    insert workPartList;
}

@isTest
static void testMaintenanceRequest(){
    List<Case> caseList = new List<Case>();
    for(Case caseHere : [select Id from Case where SuppliedName = 'Test']) {
        caseHere.Type = 'Repair';
        caseHere.Status = 'Closed';
        caseList.add(caseHere);
    }

    Test.startTest();
    update caseList;
```

```
        System.assertEquals(300, [SELECT count() FROM Case WHERE Type = 'Routine Maintenance' and Date_Reported__c = :Date.today()]);

        Test.stopTest();


    }

}
```

Challenge 5

WarehouseCalloutServiceMock Class

```
public class WarehouseCalloutServiceMock implements HttpCalloutMock {

    private String responseJson = '[' +

'{"_id":"55d66226726b611100aaf741","replacement":false,"quantity":5,"name":"Generator 1000 kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"},' +

'{"_id":"55d66226726b611100aaf742","replacement":true,"quantity":183,"name":"Cooling Fan","maintenanceperiod":0,"lifespan":0,"cost":300,"sku":"100004"},' +

'{"_id":"55d66226726b611100aaf743","replacement":true,"quantity":143,"name":"Fuse 20A","maintenanceperiod":0,"lifespan":0,"cost":22,"sku":"100005"}' +

                           ']';



    public HTTPResponse respond(HTTPRequest request) {

        HttpResponse response = new HttpResponse();

        response.setHeader('Content-Type', 'application/json');

        response.setBody(responseJson);

        response.setStatusCode(200);
        return response;
    }
}
```

WarehouseCalloutServiceTest Class

```
@isTest
private class WarehouseCalloutServiceTest {

    @isTest
    static void testRunWarehouseEquipmentSync(){
        Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());

        Test.startTest();
        WarehouseCalloutService.runWarehouseEquipmentSync();
        Test.stopTest();

        System.assertEquals(3, [select count() from Product2]);
    }


}
```

Challenge 6

WarehouseSyncScheduleTest Class

```
@isTest
public class WarehouseSyncScheduleTest {
  public static String CRON_EXP = '0 0 1 * * ?';

  @isTest
  static void testExecute(){
    Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());

    Test.startTest();
    String jobId = System.schedule('WarehouseSyncScheduleTest', CRON_EXP, new WarehouseSyncSchedule());
    Test.stopTest();

    System.assertEquals(1, [SELECT count() FROM CronTrigger WHERE CronJobDetail.Name = 'WarehouseSyncScheduleTest']);
  }

}
```