**Problem: SOQL time limits being hit**

1. Make sure best practice of : 1 trigger per object
2. 1 PB per object. Is followed. PBs are not recursive. Move nodes into before insert/update form PB if possible to save additonal dml loop.
3. Use winter’18 feature .
4. Boolean hasPermission = FeatureManagement.checkPermission(‘Permission set’) to check if a particular permission is assigned to user . Remove Profile Query if being used for this purpose.

Projects can be very big and complex with use of Managed packages etc. So, for better analysis below approach can be used in org , so that there is enough visibility of what method contributing to SOQL,CPU time. This is a scalable solution, so can be made a process in dev orgs to list the contribution if apex class is being used.

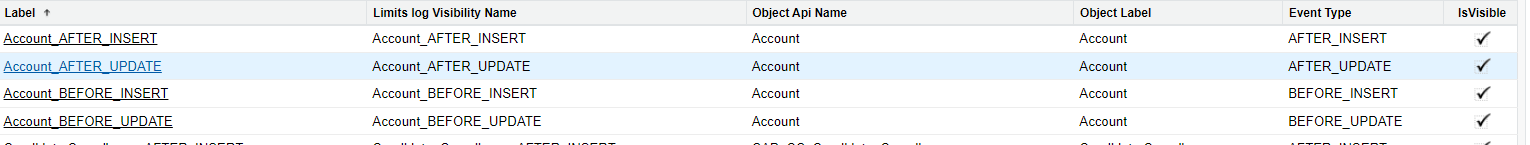
**Show/hide debug logs based on CPU limits.**

To analyse in org, for each transaction, which method is contributing to SOQL count and CPU time usage. This will help in better code quality and better manageability from code review too to be able to understand overall performance at initial stages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Custom Metadata Type Detail** | |  | | |
| **Singular Label** | Limits log Visibility | | **Description** | show/hide debug logs related to cpu limits |
| **Plural Label** | Limits log Visibilites | | **Visibility** | Public |
| **Object Name** | Limits\_log\_Visibility | | **Protection Level** |  |
| **API Name** | Limits\_log\_Visibility\_\_mdt | | **Record Size** | 671 |

|  |  |  |
| --- | --- | --- |
| **Field Label** | **API Name** | **Data Type** |
| [Event Type](https://pagegroup--ccdev.my.salesforce.com/00N3M000000ODY4?setupid=CustomMetadata) | Event\_Type\_\_c | Picklist |
| [IsVisible](https://pagegroup--ccdev.my.salesforce.com/00N3M000000ODY9?setupid=CustomMetadata) | IsVisible\_\_c | Checkbox |
| [Object Api Name](https://pagegroup--ccdev.my.salesforce.com/00N3M000000ODXu?setupid=CustomMetadata) | Object\_Api\_Name\_\_c | Text(255) |
| [Object Label](https://pagegroup--ccdev.my.salesforce.com/00N3M000000ODXz?setupid=CustomMetadata) | Object\_Label\_\_c | Text(255) |

Sample entry of metadata:



Amend the trigger handlers :

 /\*\*  
    @description Executes required trigger logic  
    \*/   
    public override void execute() {  
        try{  
            switch on triggerEvent{  
                when BEFORE\_INSERT{  
                    executeBeforeInsert(trigList);  
                }}}  
private static void executeBeforeInsert(List<SObject> trigList){  
        if(trigList != NULL){  
            **Utilities.showDebugLog(Constants.ACCOUNT\_API,Constants.EVENT\_BEFORE\_INSERT,Constants.EVENT\_START);**  
            <Apex class>.<methodABC>(trigList, String.valueOf(trigList.getSObjectType()));  
            **Utilities.showDebugLog(Constants.ACCOUNT\_API,Constants.EVENT\_BEFORE\_INSERT, methodABC);**  
                        AccountTriggerHelper.method2(trigList);  
            **Utilities.showDebugLog(Constants.ACCOUNT\_API,Constants.EVENT\_BEFORE\_INSERT, method2);  
            Utilities.showDebugLog(Constants.ACCOUNT\_API, Constants.EVENT\_BEFORE\_INSERT, Constants.EVENT\_END);**

Put the code in the Utilitycode as below:

public static map<string, Limits\_log\_Visibility\_\_mdt> logsMetadataMap {get;set;}  
  
*/\* @description get metadata details to decide whether to show logs or not \*/*    public static map<string,Limits\_log\_Visibility\_\_mdt> getLogsMetadata(){  
        if(logsMetadataMap == null){  
            logsMetadataMap = new map<string, Limits\_log\_Visibility\_\_mdt>();  
            list<Limits\_log\_Visibility\_\_mdt> logsMetadata = [SELECT id,  IsVisible\_\_c, Event\_Type\_\_c, Object\_Api\_Name\_\_c,   
                                                                    Object\_Label\_\_c   
                                                                    FROM Limits\_log\_Visibility\_\_mdt];  
            for(Limits\_log\_Visibility\_\_mdt log: logsMetadata){  
                logsMetadataMap.put(log.Object\_API\_Name\_\_c+'-'+log.Event\_Type\_\_c,log); //key will be uniquei n the custom metadata.  
                  
            }  
                    }  
        return logsMetadataMap;  
    }  
      
*/\*\* @description show limits debug\*/*    public static void showDebugLog(String objAPIName, String eventName, String methodName){  
        map<String,Limits\_log\_Visibility\_\_mdt> metadata= getLogsMetadata();  
        String key = **objAPIName+'-'+eventName**; //unique in the metadata  
          
        if(metadata.containsKey(key) && metadata.get(key).IsVisible\_\_c) {  
            **system.debug(metadata.get(objAPIName+'-'+eventName).Object\_Label\_\_c+'==='+eventName+'==='+methodName+'===CPU Time==='+Limits.getCpuTime()+'===SOQL count=='+Limits.getQueries()); //this In the debug log wil help in the cpu and soql time contribution of newly added methods and overall quality mgmt..**  
        }  
    }