

# GOVERNOR LIMITS ASSIGNMENT

## //Optimized code for Trigger - 1

```
trigger accountTestTrigger on Account (before insert, before update) {
    List<Contact> toUpdate = [SELECT id, salutation, firstname, lastname FROM contact
    WHERE accountId IN: Trigger.new];

    for(Contact c : toUpdate){
        c.Description = c.salutation + ' ' + c.firstName + ' ' + c.lastname;
    }

    update toUpdate;
}
```

## //Problems associated with previous code:

- There was a SOQL query being executed inside a for loop which is not a good practice.
- There was DML call (update) performed on single sObjects which is not a good practice since governor limits might be hit.

## //Optimized code for Trigger - 2

```
trigger accountTrigger on Account (before delete, before insert, before update) {
    List<Account> accountWithBoth = [select id, name,
                                     (select id, name, closedate, stagename
                                      from Opportunities
                                      where accountId IN :Trigger.newMap.keySet() and
                                      (StageName='Closed - Lost' or StageName = 'Closed - Won'))
                                     from Account where Id IN :Trigger.newMap.keySet()];

    for(Account acc : accountWithBoth){
        for(Opportunity opp: acc.Opportunities){
            if(opp.StageName == 'Closed - Won'){
                System.debug('Do more logic here...');
            }
            else if(opp.StageName == 'Closed - Lost'){
                System.debug('Do more logic here...');
            }
        }
    }
}
```

```

    }
  }
}

```

### **//Problems associated with previous code:**

- There were multiple SOQL queries whose results can be achieved by use of only one query
- There were more iterations than necessary.

### **//Solution for question - 3**

trigger accountLimitExample on Account (after delete, after insert, after update) {

```

    System.debug('Total Number of SOQL Queries allowed in this Apex code context: ' +
Limits.getLimitQueries());

```

```

    System.debug('Total Number of records that can be queried in this Apex code
context: ' + Limits.getLimitDmlRows());

```

```

    System.debug('Total Number of DML statements allowed in this Apex code context: '
+ Limits.getLimitDmlStatements() );

```

```

    System.debug('Total Number of CPU usage time (in ms) allowed in this Apex code
context: ' + Limits.getLimitCpuTime());

```

```

    List<Opportunity> opptys = [select id, description, name, accountid, closedate,
stagename from Opportunity where accountId IN: Trigger.newMap.keySet()];

```

```

    System.debug('1. Number of Queries used in this Apex code so far: ' +
Limits.getQueries());

```

```

    System.debug('2. Number of rows queried in this Apex code so far: ' +
Limits.getDmlRows());

```

```

    System.debug('3. Number of DML statements used so far: ' +
Limits.getDmlStatements());

```

```

    System.debug('4. Amount of CPU time (in ms) used so far: ' + Limits.getCpuTime());

```

```

    if (opptys.size() + Limits.getDMLRows() > Limits.getLimitDMLRows()) {
        for (Account a : Trigger.new) {
            a.addError('You are attempting to update the addresses of too many accounts at
once. Please try again with fewer accounts.');
```

```

        }
    }
}

```

```

else{
    System.debug('Continue processing. Not going to hit DML governor limits');
    System.debug('Going to update ' + opptys.size() + ' opportunities and governor
limits will allow ' + Limits.getLimitDMLRows());
    for(Account a : Trigger.new){
        System.debug('Number of DML statements used so far: ' +
Limits.getDmlStatements());

        for(Opportunity o: opptys){
            if (o.accountid == a.id)
                o.description = 'testing';
        }

    }
    update opptys;
    System.debug('Final number of DML statements used so far: ' +
Limits.getDmlStatements());
    System.debug('Final heap size: ' + Limits.getHeapSize());
}
}

```