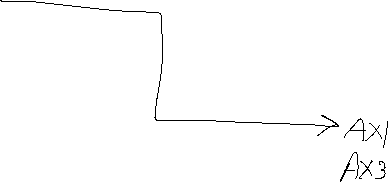
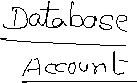
37. Delete Duplicate Existing Records and Prevent Duplicate Records - 23 May 2022

1] Delete Duplicate Existing Records

2] Prevent Duplicate Records

=============================================================================

1] Delete Duplicate Existing Records



|  |  |
| --- | --- |
| Event | Before |
| Operations | Insert, Update |
| Trigger Object | Account |
| Affecting Object | Account |



**Garib Code:**

trigger AccountTrigger23May on Account (before insert, before update) {

for(Account objAcc : trigger.new){//AX1, AX3, AX4

if(String.isNotBlank(objAcc.Name)){

List<Account> accList = [select Id, Name from Account where Name=: objAcc.Name];

Database.delete(accList,false);

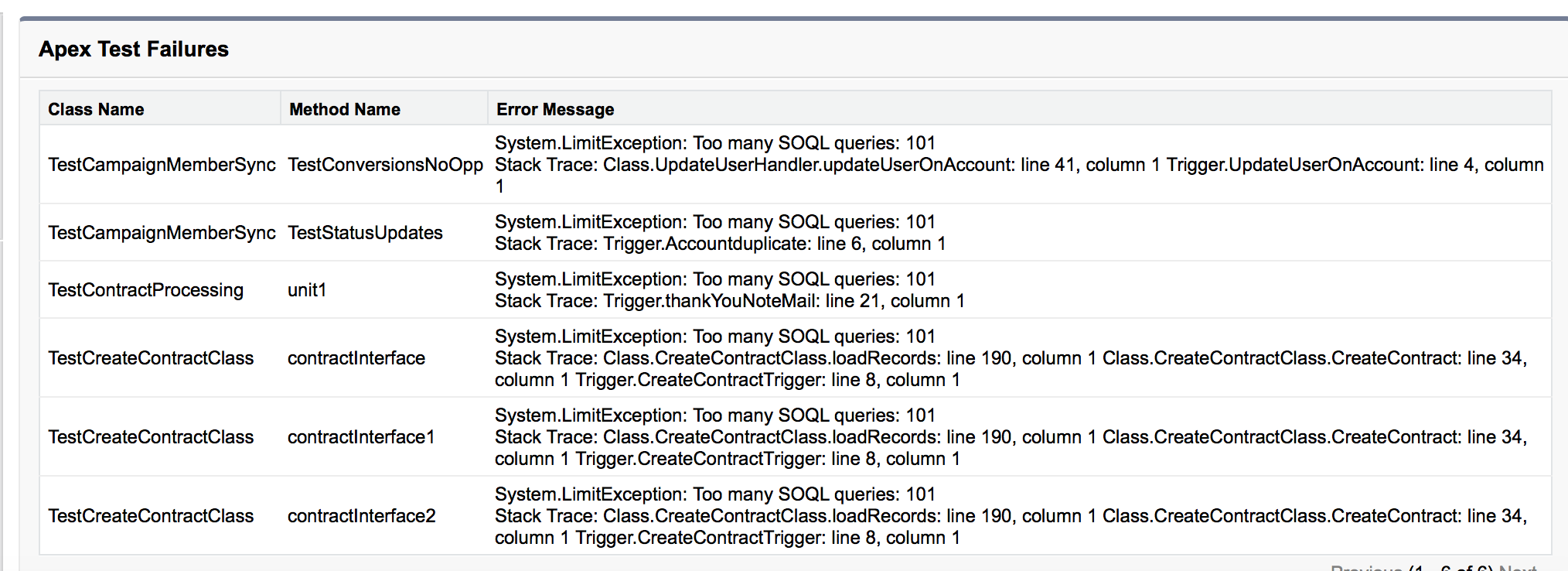
}

}

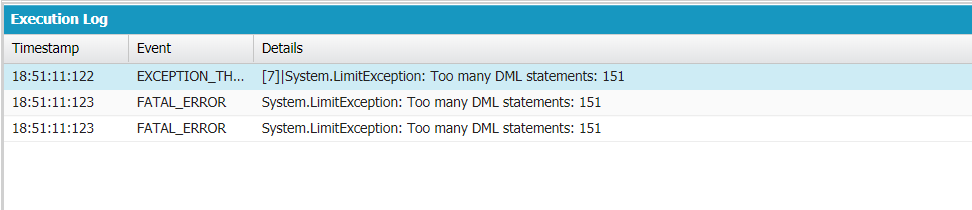
}

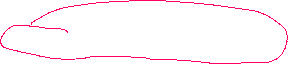
Governor LIMITS

|  |  |
| --- | --- |
| **SOQL** | **DML Statement** |
| 01 Transaction = 100 | 01 Transaction = 150 |
| 01 SQOL = MAX 50K FETCH | 01 DML = MAX 10K Process |









Better Code:

trigger AccountTrigger23May on Account (before insert, before update) {

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

for(Account objAcc : trigger.new){

if(String.isNotBlank(objAcc.Name)){

if(trigger.isInsert && trigger.isBefore){

accNameSet.add(objAcc.Name);

}

if(trigger.isUpdate && trigger.isBefore){

if(objAcc.Name != trigger.oldMap.get(objAcc.Id).Name){

accNameSet.add(objAcc.Name);

}

}

}

}

List<Account> accList = [select Id, Name from Account where Name IN : accNameSet];

if(!accList.isEmpty())

Database.delete(accList,false);

}

2] Create/Update a New Applicant. Delete the existing applicant for same mobile number.

|  |  |
| --- | --- |
| Event | Before |
| Operations | Insert, Update |
| Trigger Object | Applicant\_\_c |
| Affecting Object | Applicant\_\_c |

trigger ApplicantTrigger23May on Applicant\_\_c (before insert, before update) {

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

for(Applicant\_\_c objApp : trigger.new){

if(!String.isBlank(objApp.Mobile\_Number\_\_c)){

if(trigger.isInsert && trigger.isBefore){

appMobileSet.add(objApp.Mobile\_Number\_\_c);

}

if(trigger.isupdate && trigger.isBefore){

if(objApp.Mobile\_Number\_\_c != trigger.oldMap.get(objApp.Id).Mobile\_Number\_\_c){

appMobileSet.add(objApp.Mobile\_Number\_\_c);

}

}

}

}

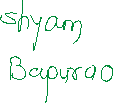
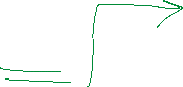
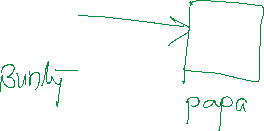
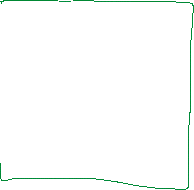
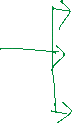
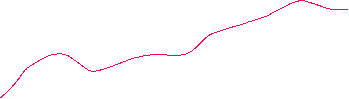
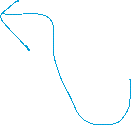
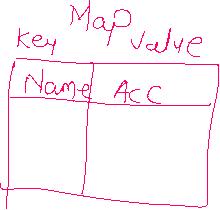
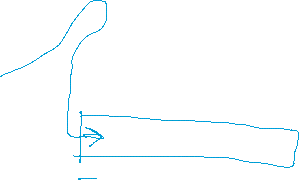
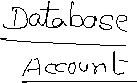
Database.delete( [select Id from Applicant\_\_c where Mobile\_Number\_\_c IN : appMobileSet] ,false);

}

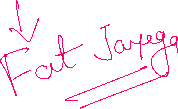
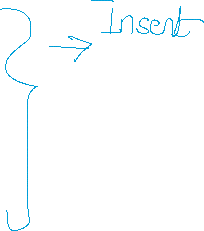
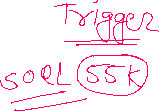
2] Prevent Duplicate Records

Create/Update an Account Record. Prevent this record if already exists with the Same Name.

|  |  |
| --- | --- |
| Event | Before |
| Operations | Insert, Update |
| Trigger Object | Account |
| Affecting Object | Account |





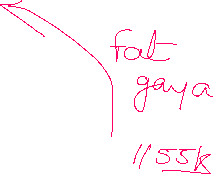


**List<Account> accExistingList = [select Id, Name from Account where Name IN : accNameSet];//55k 🡪 LIMIT 50K**

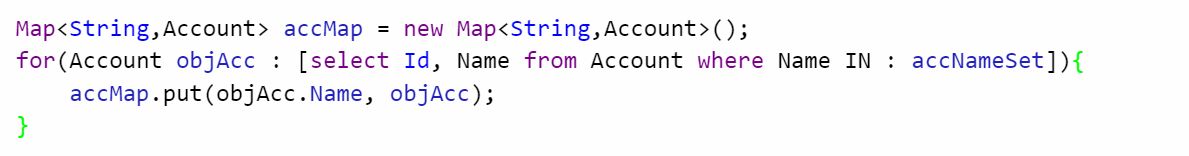
Map<String,Account> accMap = new Map<String,Account>();

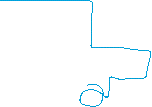
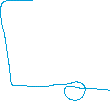
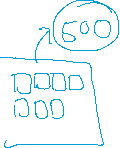
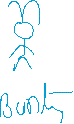
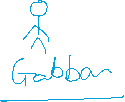
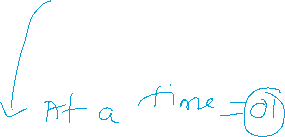
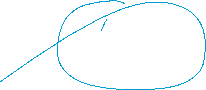
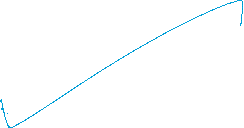
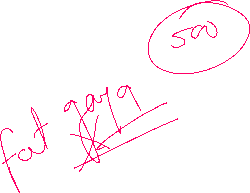
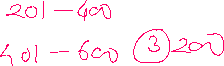
for(Account objAcc : accExistingList){

accMap.put(objAcc.Name, objAcc);



}





trigger AccountTrigger23Prevent on Account (before insert, before update) {

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

for(Account objAcc : trigger.new){//56K

if(!String.isBlank(objAcc.Name)){

if(trigger.isInsert && trigger.isBefore){

accNameSet.add(objAcc.Name);

}

if(trigger.isUpdate && trigger.isBefore){

if(objAcc.Name != trigger.oldMap.get(objAcc.Id).Name){

accNameSet.add(objAcc.Name);

}

}

}

}

Map<String,Account> accMap = new Map<String,Account>();

if(!accNameSet.isEmpty()){

for(Account objAcc : [select Id, Name from Account where Name IN : accNameSet]){

accMap.put(objAcc.Name, objAcc);

}

}

if(!accMap.isEmpty()){

for(Account objAcc : trigger.new){

if(accMap.containsKey(objAcc.Name)){

objAcc.Name.addError('Duplicate Record Found...!!!!');

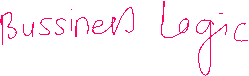
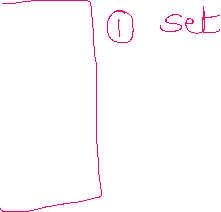
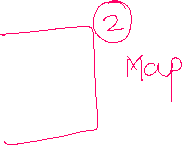
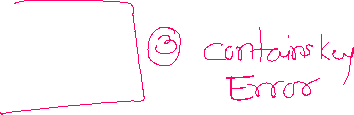
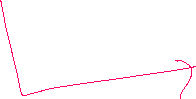
}

}

}

}





**Assignment:**

1] Prevent Duplicate Contact as per Email Id.

2] Prevent Duplicate Opportunity Amount.