Using Database.upsert with external ID field

External Id plays very important role if you want to update records without knowing the record Ids or want to relate the child record with parent record without knowing the parent record Id.

As a best practice, you should always make External Id unique. If you are performing upsert with External Id, then following situations will occur:

1. If no record is found in table with provided External Id, then it will create record in table.
2. If 1 record is found in table with provided External Id, then it will update record in table.
3. If more than 1 records is found in table with provided External Id, then system will throw an error.

I am going to cover 2 different aspect of using external Id in apex.

- **Updating a record with External Id**

Create a External Id field on Account as Account_Unique_Number__c and mark it as External Id and unique while creating it.Now we will create a new record using upsert.

Execute below command in developer console

```
List<Account> acclist=new list<Account>();
acc.name='Demo test1';
acc.Account_Unique_Number__c='00001';
acclist.add(acc);
Schema.SObjectField ftoken = Account.Fields.Account_Unique_Number__c;
Database.UpsertResult[] srList = Database.upsert(acclist,ftoken,false);
for (Database.UpsertResult sr : srList) {
    if (sr.isSuccess()) {
        // Operation was successful
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : sr.getErrors()) {
            System.debug('error has occurred.' + err.getStatusCode() + ': ' +
err.getMessage());
            System.debug('fields that affected this error: ' + err.getFields());

        }
    }
}
```

As there is no record in Account with Account_Unique_Number__c as 00001, system will create a new record.

Now again we will run same script in developer console and will specify some more field values:

```
List<Account> acclist=new list<Account>();
Account acc=new Account();
acc.name='Demo test1';
acc.Account_Unique_Number__c='00001';
acc.type='Other';
acc.Industry='Banking';
acclist.add(acc);
Schema.SObjectField ftoken = Account.Fields.Account_Unique_Number__c;
Database.UpsertResult[] srList = Database.upsert(acclist,ftoken,false);
for (Database.UpsertResult sr : srList) {
    if (sr.isSuccess()) {
        // Operation was successful
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : sr.getErrors()) {
            System.debug('error has occurred.' + err.getStatusCode() + ': ' +
err.getMessage());
```

```
            System.debug('fields that affected this error: ' + err.getFields());

        }
    }
}
```

Now you will see that system will update the record as it was able to find a Account record with Account_Unique_Number__c as 00001

- **Relating a child record with parent record by using parent record Id**

In order to understand this, we will create contact record and will relate to account using Account_Unique_Number__c. Execute below code in developer console:

```
List<Contact> conlist=new list<Contact>();
Contact con=new Contact();
con.lastname='Kumar';
con.Firstname='Kumar';
con.email='sunil02kumar@gmail.com';
Account acc=new Account(Account_Unique_Number__c='00001');
con.Account=acc;
conlist.add(con);
Database.UpsertResult[] srList = Database.upsert(conlist,false);
for (Database.UpsertResult sr : srList) {
    if (sr.isSuccess()) {
        // Operation was successful
    }
    else {
        // Operation failed, so get all errors
        for(Database.Error err : sr.getErrors()) {
            System.debug('error has occurred.' + err.getStatusCode() + ': ' +
err.getMessage());
            System.debug('fields that affected this error: ' + err.getFields());
        }
    }
}
```

This will create a new contact for Account which have Account_Unique_Number__c as 00001.

In above code snippet, you can see that in order to relate contact with account, we are not specifying the account 15 or 18 digit record id. We are just specifying the external Id of account and system will maintain the relationship.

Why it is recommended to mark External Id as unique?

Imagine you are creating a contact and specified External Id of parent. Suppose there are 2 records in account table with same value, then system will not able to identify with whom it needs to relate the contact and will throw error saying more than 1 match found.