How to use Database.Query and Database.countQuery

0 Comments **Thursday, December 29, 2016**

Dynamic SOQL with Database.Query and Database.countQuery

When we are working with Apex programming and SOQL(Salesforce Object Query Language) always it's not possible to write static SOQL queries. Using dynamic SOQL Queries we can achieve more flexible applications. Apex language has a greater feature to write some Dynamic SOQL queries using Database.Query, Database.getQueryLocatior and Database.countQuery. In this article, I'm going to show you some sample code snippets using these methods.

Database.Query:

Database.query allows developers to write Query in String format. Runtime it will execute as SOQL Query and return a single sObject or list of sObject type.

Ex: Query to returns a single sObject record.

```
sObject so = Database.query('SELECT Id FROM Account Limit 1');
```

Above example returns single account object record. when you try to run above query without **LIMIT 1** it will throw error message "**System.QueryException: List has more than 1 row for assignment to SObject**".

Ex: Query to return a List of sObjects

```
List<sObject> sobjList = Database.query('SELECT Id FROM Account ');
```

Above piece of code returns a list of all account object records.

Database.countQuery:

Sometimes we need to Count the total number of records returned by dynamic SOQL query. In this type of scenario Database.countQuery can use. It will return count as Integer format.

Ex: Returns count of records

```
Integer totalAccounts = Database.countQuery('SELECT count() FROM Account');
```

above example returns a number records in account object. and return type is Integer.

It **can't** use for Aggregate results of field properties. see the following code. It will throw error message "**System.QueryException: Use query() for non-count queries**".

```
Integer totalAccounts = Database.countQuery('SELECT count(id) FROM Account');
```

Note: Database.countquery returns upto 50,000 records only. If it has more then 50000 records it will throw error "**Too many query rows: 50001 from with count()".**

Count more then 50,000 records:

I have one trick to do that with out hitting the too many query rows: 50001 Error message in salesforce. I'm using AggregateResult to achieve the count of records up to 1million records.

```
Integer recCount = 0;
for(AggregateResult result :[SELECT COUNT(Id) total FROM Account])
    recCount+=(Integer)result.get('total');
System.debug('No Of Records:'+recCount);
```