

Hack Query Limit 50001 error

☆☆☆☆☆  Rate This

Issue Statement:

Salesforce Error: Query return more than 50001 row

Reason:

Salesforce enforce various governor limits. The error occurs comes due to one of the governor limit

Total number of records retrieved by SOQL queries: 50,000

Possible Solutions:

1> Put proper filtering in the query to avoid fetching unnecessary record

Ex.:

[Select id from Account]

Instead use

[Select id from Account where Active = True]

Limitation:

The filtering logic may be very complex and may not be possible to build a query out of it.

2> Put limit keyword

Ex:

[Select id from Account]

Instead use

Ex.:

[Select id from Account limit 50000]

Limitation:

This is not a reliable option. Logic may fail due to required record not fetched.

3> Use Batch Processing

Implement batch class and process the record in batch. More details for batch apex

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_batch_interface.htm

Limitation:

Batch class runs asynchronous. If the expected result is real time, this option will not work.

4> ReadOnly Annotation:

Salesforce has offered @readonly option to come over the limitation. A controller method, webservice method can be defined as readonly. Any SOQL inside this method can fetch more than 50,000 row.

Example:

Controller Class

@RemoteAction @readOnly

public static boolean processLines2()

{

System.debug('run');

list<Account> ListAccount = [select id,name

FROM Account Limit 70000];

System.debug(ListAccount + ListAccount.size());

return True;

}

VF page

<apex:page controller="RemotingController" showHeader="false" sidebar="false" title="Pricing">

<apex:includeScript value="{!URLFOR(\$Resource.Apttus_Config2__jQueryUILibrary19, 'js/jquery-1.8.3.min.js')}"/>

<https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js>

document.addEventListener('DOMContentLoaded', function() {
alert('test');

ApexCall();

}, false);

var Vf_RemotingMethod2 = '{!\$RemoteAction.RemotingController.UpdateRecord}'; // Provide Apex controller static remote method which will compute the pricing

function ApexCall2(){

console.log('In VF remoting');

Visualforce.remoting.Manager.invokeAction(

Vf_RemotingMethod2,

function(result, event){

console.log('event.status Status: '+event);

if (event.status) { // if Method is executed successfully then execute these line of code

// result variable hold the returned output from the method which we called

console.log('Line Num Status sucess');

```
alert('success');

}

else if (event.type === 'exception') { // write the handler code if any exception occurred during the method execution

alert( event.message);

//returntoCart();

} else { // if anything goes unexpectedly then execute this block

alert( event.message);

//returntoCart();

}

},

{escape: true}

);

}
```

```
<apex:form >

<apex:pageMessages />

</apex:form>

</apex:page>
```

Limitation:

To use the @ReadOnly annotation, the top level method must be readonly method . For example, if aVisualforce page calls a Web service that contains the @ReadOnly annotation, the request fails because Visualforce is the top level request, not the Web service.