Non-selective query

System.QueryException: Non-selective query against large object type (more than 200000 rows). Consider an indexed filter or contact salesforce .com about custom indexing

caused by: System.QueryException: Non-selective query against large object type (more than 200000 rows). Consider an indexed filter or contact salesforce .com about custom indexing. Even if a field is indexed a filter might still not be selective when: 1. The filter value includes null (for instance binding with a list that contains null) 2. Data skew exists specifically the number of matching rows is very large (for instance, filtering for a particular foreign key value) that occurs many times)

Cause

An error will occur if a SELECT statement is issued for an object that exceeds 100,000 with a custom item as a condition in the trigger.

Countermeasures

If the condition set in the SOQL condition (Where clause) is set to the external ID in the custom item, the internal index will be created, and no error will occur even if the number exceeds 100,000.

reference

How to request when a custom index is required

Consideration for processing speed of SOQL query

Force.com Query Optimizer FAQ

System.QueryException: Non-selective query against large object type (more than 200000 rows). About the error

important point

-When dealing with external IDs, up to 3 external IDs can be specified for one object.

-Even if the execution result of SOQL is less than 200,000 records, non-selective query against large object type (more than 100000 rows) when the execution plan confirms the efficiency of SOQL and determines that it is an inefficient query.) Is output.

Exceptions that are no longer selective when using indexes

Query for null rows — Query to find records where the item is empty or null

```
SELECT Id, Name FROM Account WHERE Custom_Field__c = null
```

Negative narrowing operator - in the query =, NOT LIKE, such as EXCLUDES operator use of

```
SELECT CaseNumber FROM Case WHERE Status! ='New'
```

First wildcard

```
SELECT Id, LastName, FirstName FROM Contact WHERE LastName LIKE'% smi%'
```

Text items using comparison operators

```
SELECT AccountId, Amount FROM Opportunity WHERE Order_Number__c> 10
```

At the end

Let's use the query execution plan tool

1. 1. Under Settings, click <your name>>> Developer Console to open the Developer Console.
2. 2. In the developer console, select Help> Preferences.
3. 3. Select Enable Query Plan and make sure it is set to true.
4. Click Save.
5. On the Query Editor tab, make sure that the Query Plan button appears next to the Execute button.