Improve performance with Custom indexes using Selective SOQL Queries

Mallareddy Reddy Thursday, July 06, 2017

Description

SOQL queries must be selective, particularly for queries inside triggers for the best p
by the system. Developers will receive an error message when a non-selective quer
you can avoid the error, and make the query selective below.

Resolution

Important: While every effort has been made to ensure this article explained how the Salesf
accommodate additional performance enhancements. Developers are advised to use the Que
queries based on the information provided by the optimizer at the time the query plan for the

Selective SOQL Query Criteria

A query is selective when one of the query filters is on an indexed field and the query filter r
query improves when two or more filters used in the WHERE clause meet the mentioned co

The selectivity threshold is 10% of the records for the first million records and less than 5%
circumstances, for example with a query filter that is an indexed standard field, the threshold

Custom Index Considerations for Selective SOQL Queries

The following fields are indexed by default:

- Primary keys (Id, Name and Owner fields).
- Foreign keys (lookup or master-detail relationship fields).
- Audit dates (such as SystemModStamp).
- Custom fields marked as External ID or Unique.

Salesforce Support can add Custom Indexes upon request for customers

When creating a case with Support, be sure to include the SOQL query which has the field t

Please make sure you provide all the information to Support mention in this checklist:
You can also determine which fields can be indexed by referring to How to make SOQL que

A Custom Index can't be created on these types of fields:

- Multi-select Picklists.
- Currency fields in a Multicurrency Organization.
- Long text fields.
- Binary fields (fields of type blob, file, or encrypted text.).

Note: New data types, typically complex ones, may be added to Salesforce and fields of thes

Typically, a custom index won't be used in these cases:

- The value(s) queried for exceeds the system-defined threshold mentione
- The filter operator is a negative operator such as NOT EQUAL TO (or !
- The CONTAINS operator is used in the filter and the number of rows tc

  Note that this threshold is subject to change.
- When comparing with an empty value (Name != '').

Good to know: There are other complex scenarios in which Custom Indexes won't be used.
further assistance with non-selective queries.

Examples of Selective SOQL Queries

To better understand whether a query on a large object is selective or not, let's analyze some
records, that is, deleted records that are still in the Recycle Bin) for the Account sObject.

Query 1:

```
SELECT Id FROM Account WHERE Id IN (<list of account IDs>)
```

The WHERE clause is on an indexed field (Id). If SELECT COUNT() FROM Account WH
used. This will typically be the case since the list of IDs only contains a small amount of rec

Query 2:

```
SELECT Id FROM Account WHERE Name != ''
```

Since Account is a large object even though Name is indexed (primary key), this filter return

Query 3:

```
SELECT Id FROM Account WHERE Name != '' AND CustomField__c = 'ValueA'
```

Here we have to see if each filter, when considered individually, is selective. As we saw in th
returned by SELECT COUNT() FROM Account WHERE CustomField__c = 'ValueA' is lov

Query 4:

```
SELECT Id FROM Account WHERE FormulaField__c = 'ValueA'
```

The following rules have to be true in order to index a Formula Field:

- The formula contains fields from a single object only (not relationship f
- The formula field doesn't reference any non-deterministic functions (e.g
- The formula field doesn't reference any non-supported fields for includi

  Spring 12(176), createdById was non-supported, but in Summer 12 (178), it
- The formula field contains references to Primary Keys (e.g Id)

If one of the condition above is false, the "Add index" does not display on the field.