Four Effective Ways To Avoid Data Skew

For the organizations, which have a very large database CRM designing is a considerable and necessary aspect. They have to maintain hundreds or thousands of records. With database usually, many accounts are associated with the main account and this association of many accounts with a single account is known as data skew. Data skew can impact the performance of your CRM, so it is essential to prevent the data skew to get better performance of CRM by increasing the number of records, which may impact the CRM up to a great extent.

This article discusses the considerable practices to design the CRM effectively, as a result of which the data skew can be prevented. These are:

Contents

Link the child records with multiple accounts

The main account and the child records use the similar resource, so the user who will have the access to the main account, will also have access to its child records. The time when the child record, which is also called opportunity is updated, its main or parent account also gets updated and locked and due to this lock the CRM performance may get affected. As per the Salesforce recommendation, with a single main account, do not associate more than 10,000 child accounts. For this you can create various account types to handle these child records.

> **Related Post:** **The Biggest Benefits of CRM Software For Financial Advisors**

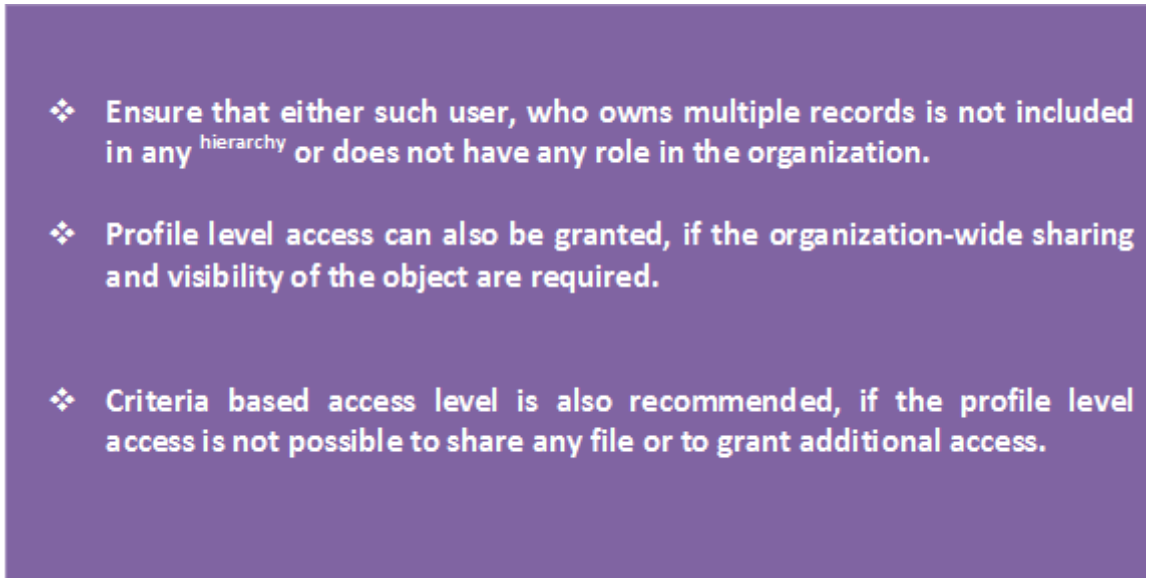
Prevent ownership skew

When a single user owns a large number of records of the same object, then it is called the ownership skew. In a number of cases and scenarios, it can happen, like in the case when all unassigned records or opportunities are assigned to the Sales or Development director another case

of ownership skew can be, when the system administrator can be default owner at the time of data migration. In both of the above scenarios, the single owner owns a number of records. Such an overburdened or heavy ownership can negatively impact the system performance.

Read More: What is The Process of Salesforce Implementation

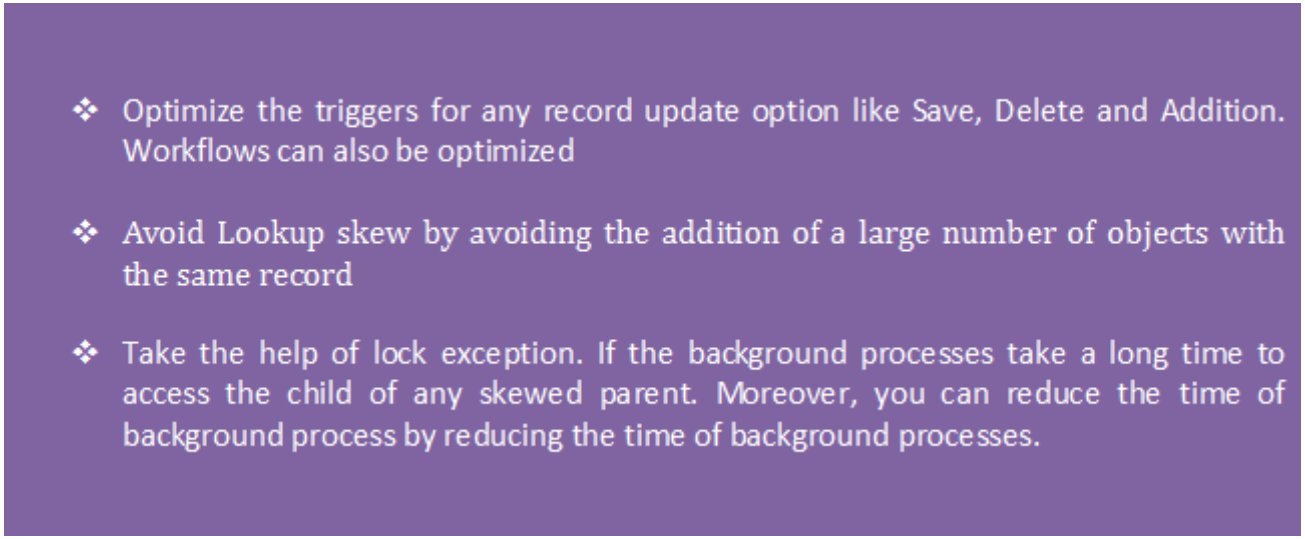
To resolve such problem either try to reassign a few of such records to another user or delete such records. In case, if the ownership skew cannot be prevented, then in such cases Salesforce recommends following options:



Avoid Lookup Skew

In case, if through lookup relationship a large number of records get associated with a single record and the performance of CRM gets down, then it is called Lookup skew. Such skews are known as lookup, which should be avoided. Here, for example, if the main object has three child objects and they each have again 20,000 sub children objects, then there will become a large number of objects to be maintained, this is called account skew. If there has to be performed any same action to all records, then the possibility of deadlock situation also increases. So any update operation like to save, delete or addition must be executed safely. However, Salesforce recommends following options for any of such situation:

> **Related Post:** CRM For Non-Profits: Drive Service Engagement, Increase Productivity And Reduce Costs



Such care while designing the CRM can improve the performance and the improved business processes can also improve the performance. The organization can also improve the performance of Salesforce by preventing the large number of objects, which are pointing to the same object.

Avoid Data Skew

Try to identify and avoid such records or data that can produce the unwanted lock and diminish the database concurrency. Due to change in ownership, the data correction can be painful. Following are a few suggestions to avoid data skew:

- Try to limit the children to 10,000 while designing the architecture, for this you can assign the children by using the Round Robin algorithm with the help of custom settings for the child and parent objects.
- You can adopt the public Read/Write sharing model, where the parent account stays locked and the calculations still occur.
- If the skewed accounts are there, then you can re-distribute the child objects to reduce the record-level hurdles.