

# Designing Salesforce to improve performance and avoid data skew

By: [Partner](#) | 7.13.17



*This guest post comes from [Naveen Gabrani](#), a Salesforce expert who writes about the technical architecture on the Salesforce platform.*

Small decisions that we make when designing our CRM business processes can have a significant impact on performance in Salesforce. This is especially relevant for large organizations with hundreds of thousands of records.

Typically, we see a large group of records associated with one account organized under one main account in Salesforce. Seems normal, right? Well, flooding your Salesforce with a very large number of child records associated to the same account is called Data Skew. And data skew is what impacts your Salesforce instance's performance.

Without a thoughtful approach to designing your object structure and business processes, your organization could end up with a large amount of records that slow down your Salesforce instance. Let's review three best practices to consider when planning your CRM business processes and design so you can prevent data skew and improve your instance's performance.

## Limit your Opportunities with multiple Account types

The Account and Opportunity objects use the same sharing access. So if a user has access to an account, she will have similar access to all its child Opportunities. When an Opportunity record is being updated (and therefore locked), its parent Account also gets locked. And visa versa. With both Account and Opportunity locked, your performance may be impacted.

Salesforce recommends that you avoid associating more than about 10,000 Opportunities to a single Account. A simple solution could be to create a few different types of Accounts to handle these Opportunities.

## Avoid assigning one user too much ownership

When a large number of records of the same object type are owned by the same user this is called Ownership Skew. This can happen in many scenarios. One example could be when all the unassigned Opportunities (or Leads) are assigned to the Sales or Development Director. Another example is when data is being imported into Salesforce, a system administrator may be the default owner. In both cases there is the likelihood of one user owning many records of the same object.

Heavy ownership skew can have a significant negative impact on the performance of a system. Deleting records owned by the user or reassigning records to other users could help the performance of your Salesforce instance.

If it is not possible to prevent Ownership Skew, Salesforce recommends the following options:

- Ensure that the user who owns a large number of records of the same object does not have a role, or the user is placed in a role without any hierarchy.
- If organization-wide sharing of an object and visibility needs to be granted to additional users, it is best to grant access at the profile level.
- If profile level access is not possible to provide, then criteria-based sharing rule can be given to grant additional access in such cases.

## Prevent Lookup Skew with triggers and workflows

When a very large number of records are associated with a single record via the lookup relationship, this is called Lookup Skew. Such a relationship can also lead to performance problems and should be avoided.

For example, let's say an XYZ Foundation has three types of donor memberships in a custom object they created. And let's say they have over 20,000 members alone in one of those donor membership levels. In this case, a large number of accounts will look up to the same record type of Custom Object Account Type. This is an example of Account Skew.

The problem with Account Skew is Salesforce locks the parent record when a save is being performed to a child record, therefore locking you out of many records. If there is complex custom logic being executed during Save, and there are a large number of records looking up to the same parent, this can lead to lock exceptions and cause failure when another child is being saved. Thus, in our example above, there could be issues when saving a donor's record.

To handle this Salesforce recommends the following options:

- Reducing the save time by optimizing appropriate triggers and workflows.
- Replacing lookups by pick list. In the simple example discussed above for Account Type, a pick list can easily replace the lookup to a custom object.



Prevent Lookup Skew by avoiding very large number of records looking up to the same record.

Try a lock exception. At times there can be background processes that are running and the user is accessing a child of skewed parent. In such cases, the user will get a lock exception. This can be avoided by reducing the time of background process or scheduling it at night time.

Building in some time to make thoughtful decisions when designing your object structure and business processes will help you avoid any performance issues. And by preventing a large number of lookups pointing to the same record and a user owning a large number of records, your organization can keep your Salesforce instance working properly.