

# Guide to Passing Sharing and Visibility Architect.

How to prepare for and get your Sharing and Visibility Designer certification!

- [Sharing Architecture](#)
  - [Sharing Architecture](#)
- [Record-Level Access](#)
  - [Record-Level Access](#)
- [Implicit Sharing](#)
  - [Implicit Sharing](#)
- [Territory Management](#)
  - [Territory Management](#)

# Sharing Architecture

# Sharing Architecture

## Licenses:

### Full Sharing Model Usage Users/Licenses

Most Standard Salesforce license types take full advantage of the sharing model components. The license might not make a module accessible, or even some objects accessible. For example, the Free edition can't access any CRM objects. However, the sharing entities, and functionality, still exists and is ready when and if the module ever does become active.

### High Volume Customer Portal License

High Volume Customer Portal (HVPU) license users (including Community and Service Cloud license users) do not utilise the sharing model. HVPU licenses have their own sharing model that works by foreign key match between the portal user (holding the license) and the data on Account and Contact lookups. HVPU license is only used for the Customer Portal and not the Partner Portal.

### Chatter Free License

The Chatter Free license doesn't follow the standard sharing model. Chatter Free is a collaboration-only license with the following features: Chatter, Profile, People, Groups, Files, Chatter Desktop, and limited Salesforce app access. The license doesn't have access to CRM records (standard or custom objects) and Content functionality, and therefore, there is no sharing.

For the remainder of this document, we are assuming a Sales.

## Components

### Profiles and Permission Sets

Profiles and permission sets provide object-level security by determining what types of data users see and whether they can edit, create, or delete records. For each object, the "View All" and "Modify All" permissions ignore sharing rules and settings, allowing administrators to quickly grant access to records associated with a given object across the organisation. These permissions are often preferable alternatives to the "View All Data" and "Modify All Data" administrative permissions.

Profiles and permission sets also control field-level security, which determines the fields within every object that users can access.

For example, an object may have 20 fields, but field-level security can be set up to prevent the users from seeing five of the 20 fields.

### **Record Ownership and Queues**

Every record must be owned by a single user or a queue. The owner has access to the record, based on the Object Settings for the owner's profile. For example, if the owner's profile has Create and Read permission on an object, but not Edit or Delete permission, the owner can create a record for the object and see the new record. However, the owner won't be able to edit or delete the record. Users higher in a hierarchy (role or territory) inherit the same data access as their subordinates for standard objects. Managers gain as much access as their subordinates. If the subordinate has read-only access, so will the manager. This access applies to records owned by users, as well as records shared with them.

### **A Guide to Sharing Architecture Types of Data Access**

Queues help you prioritise, distribute, and assign records to teams who share workloads. Queue members and users higher in a role hierarchy can access queues from list views and take ownership of records in a queue.

If a single user owns more than 10,000 records, as a best practice:

- The user record of the owner should not hold a role in the role hierarchy.
- If the owner's user record must hold a role, the role should be at the top of the hierarchy in its own branch of the role hierarchy.

### **Organisation-Wide Defaults**

Organisation-wide sharing settings specify the default level of access users have to each others' records. You use organisation-wide sharing settings to lock down your data to the most restrictive level. Use the other record-level security and sharing tools to selectively give access to other users. For example, users have object-level permissions to read and edit opportunities, and the organisation-wide sharing setting is Read-Only. By default, those users can read all opportunity records, but can't edit any unless they own the record or are granted other permissions. Organisation-wide defaults are the only way to restrict user access to a record.

Organisation-wide default settings can be changed from one setting to another (private to controlled by parent, then back to private); however, these changes require sharing recalculation and depending on volume could result in very long processing times. For custom objects only, use the Grant Access Using Hierarchies setting, which if unchecked (default is checked), prevents managers from inheriting access. This setting is found in the organisation-wide default settings.

## **Role Hierarchy**

A role hierarchy represents a level of data access that a user or group of users needs. The role hierarchy ensures that managers always have access to the same data as their employees, regardless of the organisation-wide default settings. Role hierarchies don't have to match your organisation chart exactly. Instead, each role in the hierarchy should represent a level of data access that a user or group of users needs.

An organisation is allowed 500 roles; however, this number can be increased by Salesforce. As a best practice, keep the number of non-portal roles to 25,000 and the number of portal roles to 100,000.

As a best practice, keep the role hierarchy to no more than 10 levels of branches in the hierarchy.

When a user's role changes, any relevant sharing rules are evaluated to correct access as necessary. Peers within the same role don't guarantee them access to each other's data.

Modelling the role hierarchy begins with understanding how the organisation is structured. This is usually built from understanding a manager's scope, starting from the top. The CEO oversees the entire company. The CEO usually has direct reports that can then be segmented by Business Unit (Sales or Support) or geographical region (EMEA, APAC). That person then has direct reports that could be further segmented, and so on.

Although this sounds very much like an HR organisational chart, and we have said they might be very much alike, keep in mind, when modelling data access, focus on data accessibility with a consideration to HR reporting.

Overlays are always the tricky part of the hierarchy. If they're in their own branch, they'll require either sharing rules, teams, or territory management to gain needed access. If they are folded into the hierarchy, there might be reporting implications.

It's important to spend the time setting up the role hierarchy because it's the foundation for the entire sharing model.

## **Role Hierarchy Use Cases**

Management access – the ability for managers to be able to see and do whatever their subordinates can see and do.

Management reporting – the ability for reporting to roll up in a hierarchical fashion so that anyone higher in the hierarchy sees more data than those below them.

Segregation between organisational branches – different business units don't need to see each other's data; having a hierarchy in which you can define separate branches allows you to segregate visibility within business units, while still rolling visibility up to the executive levels above those units.

## **A Guide to Sharing Architecture Types of Data Access**

### **Role Hierarchy Use Cases**

Segregation within a role – in many organisations/applications, people who all play the same role should not necessarily see each other's data. Having hierarchical roles allows you to define a “leaf” node in which all data is essentially private, and still roll that data up to a parent role that can see all.

### **Public Groups**

A public group (not Chatter group) is a collection of individual users, roles, territories, and so on, that all have a function in common.

Public groups can consist of:

- Users
- Customer Portal Users
- Partner Users
- Roles
- Roles and Internal Subordinates
- Roles, Internal and Portal Subordinates
- Portal Roles
- Portal Roles and Subordinates
- Territories
- Territories and Subordinates
- Other public groups (nesting)

Groups can be nested (Group A nested into Group B), however don't nest more than five levels.

Nesting has an impact on group maintenance and performance due to group membership calculation. As a best practice, keep the total number of public groups for an organisation to 100,000.

### **Public Groups Use Cases**

When you need to provide access to an arbitrary group of people, you create a public group to collect them, and then use other sharing tools to give the group the necessary access. Group membership alone doesn't provide

data access.

Groups can also be nested inside each other, therefore, allowing a nested group to gain the same access as the group in which it is contained. This allows the creation of smaller, ad-hoc hierarchies that don't necessarily interact with the role or territory hierarchies.

If Group A is a member of Group B, then the members of Group A will have access to data shared to Group B at the same access level as the members of Group B.

Groups also have the ability to protect data shared in the group from being made accessible to people in the role hierarchy above the group members. This (and dealing with the access of record owners and their management hierarchy) allows the creation of groups in which very highly confidential information can be shared—the data will be accessible ONLY to group members, and nobody else in the organisation. This is accomplished by using the Grant Access Using Hierarchies setting.

### **Ownership-based Sharing Rules**

Ownership-based sharing rules allow for exceptions to organisation-wide default settings and the role hierarchy that give additional users access to records they don't own. Ownership-based sharing rules are based on the record owner only.

Contact ownership-based sharing rules don't apply to private contacts. As a best practice, keep the number of ownership-based sharing rules per object to 1,000.

## **A Guide to Sharing Architecture Types of Data Access**

### **Ownership-based Sharing Rule Use Cases**

Role-based matrix management or overlay situations: a person in Service needs to be able to see some Sales data, but they live in different branches of the hierarchy, so you can create a rule that shares data between roles on different branches.

To provide data access to peers who hold the same role/territory.

To provide data access to other groupings of users (public groups, portal. roles, territories). The members of the groupings who own the records can be shared with the members of other groupings.

### **Criteria-based Sharing Rules**

Criteria-based sharing rules provide access to records based on the record's field values (criteria). If the criteria are met (one or many field values), then a share record is created for the rule. Record

ownership is not a consideration.

As a best practice, keep the number of criteria-sharing rules per object to 50; however, this can be increased by Salesforce.

Criteria-based Sharing Rule Use Case to provide data access to users or groups based on the value of a field on the record.

## **Manual Sharing**

Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. In those situations, record owners can use manual sharing to give read and edit permissions to users who don't have access any other way. Manual sharing isn't automated like organisation-wide sharing settings, role hierarchies, or sharing rules. However, it gives record owners the flexibility to share particular records with users that must see them.

Manual sharing is removed when the record owner changes or when the sharing access granted doesn't grant additional access

beyond the object's organisation-wide sharing default access level. This also applies to manual shares created programmatically.

Only manual share records can be created on standard objects. Manual share records are defined as share records with the row cause set to manual share.

All share records (standard and custom objects) with a row cause set to manual share can be edited and deleted by the Share button on the object's page layout, even if the share record was created programmatically.

Manual Sharing Use Case to provide the user with the ability to give access (read only or read/write) to the current record to other users, groups, or roles.

## **Teams**

A team is a group of users that work together on an account, sales opportunity, or case. Record owners can build a team for each record that they own. The record owner adds team members and specifies the access level each team member has to the record.

Some team members can have read-only access, while others have read/write.

Only owners, people higher in the hierarchy, and administrators can add team members and provide more access to the member.

A team member with read/write access can add another member who already has access to the record with which the team is

associated. The team member can't provide them additional access.

Creating a team member in the app creates two records: a team record and an associated share record. If you create team members programmatically, you have to manage both the team record and associated share record. There is only one team per record (Account, Opportunity or Case). If multiple teams are needed, depending on your specific needs consider territory management or programmatic sharing.

### **A Guide to Sharing Architecture Types of Data Access**

The team object is not a first-class object. You can't create custom fields, validation rules, or triggers for teams.

Teams (Account and Opportunity) Use Cases to provide the user with the ability to give access (read-only or read/write) to a single group of users (the team).

If teams are managed externally, say through an external commission or territory management system, then integration can be used to manage the account team. There are cases when territory management in an external system can align to a team solution within Salesforce. Multiple owners of an account can be managed by the account team.

A single group of users (team) require either read-only or read/write access to an opportunity record. (Opportunity Team)

Territory Hierarchy the territory hierarchy is a single dimensional, additional hierarchy which can be structured by business units or any kind of segmentation in a hierarchical structure. When territory management is enabled you must manage both the role hierarchy and territory hierarchy.

Territories exist only on Account, Opportunity and master/detail children of Accounts and Opportunities. As a best practice, keep the territory hierarchy to no more than 10 levels of branches in the hierarchy. If the assignment rules for a territory are changed, sharing rules using that territory as the source will be recalculated. Likewise, if the membership of a territory changes, any ownership-based sharing rules that use the territory as the source will be recalculated.

### **Territory Management Use Cases**

Multiple groups of people (multiple teams) require either read-only or read/write access to accounts. An additional hierarchical structure (different from the role hierarchy) is needed. A single user needs to hold multiple levels in the hierarchy. Global users (GAM - global account manager) need to see everything from the global account downward.

### **Account Territory Sharing Rules**

Account territory sharing rules become available only when the original territory management feature has been enabled for an organisation. These sharing rules provide access to Account records that have been stamped with the Territory defined in the rule.

### **Account Territory Sharing Rule Use Case**

To provide data access to accounts within a territory (not based on ownership) to a grouping of users. Applies only to accounts and when territory management is enabled.

### **Programmatic Sharing**

Programmatic sharing (formally Apex-managed sharing) allows you to use code (Apex or other) to build sophisticated and dynamic sharing settings when a data access requirement cannot be met by any other means.

If you create a share record programmatically, and the out-of-box row cause (manual share) is used, then you can maintain this share record with the Share button in the app. The share record is subject to all rules with the manual share row cause such as deletion upon owner transfer.

### **A Guide to Sharing Architecture Types of Data Access**

#### **Programmatic Sharing Use Cases**

No other method of sharing (declarative) meets the data access needs. There is an existing, external system of truth for user access assignments which will continue to drive access and be integrated with Salesforce.

Poor performance by using native sharing components. (Usually applies to very large data volumes)

Team functionality on custom objects.

#### **Implicit Sharing**

Implicit sharing is automatic. You can neither turn it off, nor turn it on—it is native to the application. In other words, this isn't a configurable setting; however, it's very important for any architect to understand. Parent implicit sharing is providing access to parent records (account only) when a user has access to children opportunities, cases, or contacts for that account. Salesforce has a data access policy that states if a user can see a contact (or opportunity, case, or order), then the user implicitly sees the associated account.

Child implicit sharing is providing access to an account's child records to the account owner. This

access is defined at the owner's role in the role hierarchy. Child implicit sharing only applies to contact, opportunity, and case objects (children of the account). The access levels that can be provided are View, Edit, and No access for each of the children objects when the role is created. By setting to View, the account owner can implicitly see the associated object records (contact, opportunity or case). By setting to Edit, the account owner can implicitly modify the associated object (contact, opportunity or case).

**Implicit sharing doesn't apply to custom objects.**

**Customer Implementation Scenarios**

There isn't a sharing model that fits all organisations. Every organisation has different requirements and challenges when trying to architect the best sharing model. It's crucial to use the most appropriate data access components to fit the sharing requirements of the organisation. The following scenarios are common challenges when trying to architect a sharing model.



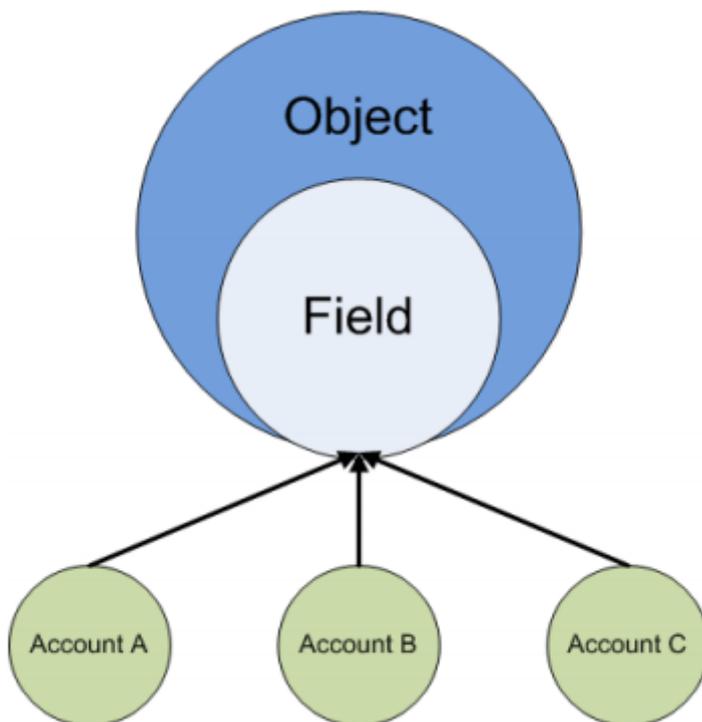
# Record-Level Access

# Record-Level Access

To meet your company's security needs, it's important to understand what data access means to your users and to you.

## Data Access: User's Perspective:

If you put yourself in your users' shoes, you won't necessarily know or care how you're getting access to records, but you might want to understand what having access means within the context of your organisation. The following graph can help users visualise the different kinds of access that can be configured in Salesforce



For example, if a user has access to an account field, then they have access to both the account field and the account object itself.

However, a specific account record, such as "Account A", might not be accessible to that user due to additional access control applied via sharing rules or other tools.

## Data Access: Architect's Perspective

As an architect, you must both understand your user's perspective and know how to grant users only the appropriate level of access to the data that they should be able to access.

From an architect's perspective, data access in Salesforce falls into two main categories: object-level access, which includes field-level access, and record-level access.

Object-level access determines whether a user has access to a particular object, which fields they can see on that object, and which actions they can perform. You configure object level access on user profiles.

### **Restricting access**

The "Read," "Create," "Edit," and "Delete" object permissions determine which actions a user can perform on any of the object's records to which they have access. Field-Level Security allows you to prevent certain users from seeing sensitive or confidential information contained in records they can see.

### **Opening up access**

The "View All" and "Modify All" object permissions give users access to all of an object's records, regardless of record-level access settings. Record-level access (called "Sharing" in Salesforce) determines which records a user can see for a particular object, using the following tools:

- Organisation-wide defaults
- Role hierarchy
- Territory hierarchy
- Sharing rules
- Teams
- Manual sharing
- Programmatic sharing

Because you have so many options for managing record-level access — and because some of these options are affected by organisational dependencies — determining which records users can access can quickly become complicated. Additionally, you might also be changing your sharing configuration frequently in response to new business requirements. This can trigger record access changes that ripple through your organisation. These changes have an even greater impact in very large organisations, where it can take some time to recalculate access for a large number of users, and adjust the tables that record their access rights. For these reasons, it's important to understand how Salesforce calculates and grants access at the database level.

### **Record Access Calculation**

Every time a user attempts to open a record, run a report, access a list view, or search for data using the user interface or API, Salesforce checks the configuration of its record access features to determine which records the user can access. These configurations can be elaborate, especially in large organisations with hundreds of hierarchy nodes, thousands of sharing rules, millions of data

rows, and portals for customers and business partners. Processing such dissimilar data and complex relationships would require far more time than the 300-millisecond Salesforce benchmark for rendering pages. Rather than applying every sharing rule, traversing all hierarchies, and analysing record access inheritance in real time, Salesforce calculates record access data only when configuration changes occur. The calculated results persist in a way that facilitates rapid scanning and minimises the number of database table joins necessary to determine record access at run time.

## **Access Grants**

When an object has its organisation-wide default set to Private or Public Read Only, Salesforce uses access grants to define how much access a user or group has to that object's records. Each access grant gives a specific user or group access to a specific record. It also records the type of sharing tool — sharing rule, team, etc. — used to provide that access. Salesforce uses four types of access grants: explicit grants, group membership grants, inherited grants, and implicit grants.

## **Explicit Grants**

Salesforce uses explicit grants when records are shared directly to users or groups. Specifically, Salesforce uses explicit grants when:

- A user or a queue becomes the owner of a record.
- A sharing rule shares the record to a personal or public group, a queue, a role, or a territory.
- An assignment rule shares the record to a user or a queue.
- A territory assignment rule shares the record to a territory.
- A user manually shares the record to a user, a personal or public group, a queue, a role, or a territory.
- A user becomes part of a team for an account, opportunity, or case.
- A programmatic customisation shares the record to a user, a personal or public group, a queue, a role, or a territory.

## **Group Membership Grants**

Grants that occur when a user, personal or public group, queue, role, or territory is a member of a group that has explicit access to the record. For example, if a sharing rule explicitly grants the Strategy group access to the Acme record, and Bob is a member of the Strategy group, Bob's membership in the Strategy group grants him access to the Acme record.

## **Inherited Grants**

Grants that occur when a user, personal or public group, queue, role, or territory inherits access through a role or territory hierarchy, or is a member of a group that inherits access through a group hierarchy.

## **Implicit Grants**

Grants that occur when non-configurable record-sharing behaviours built into Salesforce Sales, Service, and Portal applications grant access to certain parent and child records. For example, with this default logic, sometimes referred to as built-in sharing, users can view a parent account record

if they have access to its child opportunity, case, or contact record. If those users have access to a parent account record, they can also access its child opportunity, case, and contact records.

## **Database Architecture**

Salesforce stores access grants in three types of tables.

### **Object Record Tables**

Tables that store the records of a specific object, and indicate which user, group, or queue owns each record.

### **Object Sharing Tables**

Tables that store the data that supports explicit and implicit grants. Most objects in your organisation (to see them, from Setup, enter Sharing Settings in the Quick Find box, then select Sharing Settings) get their own Object Sharing table, unless any of the following conditions are also true:

- The object is a detail in a master-detail relationship. In master-detail relationships, the Object Sharing table for the master object controls access to the detail object.
- Both organisation-wide default settings (internal and external) are Public Read/Write.
- The object is of a type that doesn't support Object Sharing tables, such as Activities or Files.

These objects have their own access control mechanism.

### **Group Maintenance Tables**

Tables that store the data supporting group membership and inherited access grants. For example, if the Object Sharing table grants Bob explicit access to the Acme account record, Salesforce checks the Group Maintenance tables to see which users inherit record access from Bob and grants users access to the Acme record. These grants are established in advance when you create or change the group (or role, or territory) membership information.

While Object Sharing tables store access grants to individuals and groups, Group Maintenance tables store the list of users or groups that belong to each group, indicating group membership. Both types of tables are used to determine a user's access to data when they are searching, querying, or pulling up a report or list view.

When a user tries to retrieve one or more records, Salesforce generates a SQL statement that searches the Object Record table for records matching the user's search string. If the record exists, Salesforce appends SQL to the statement that joins the Object Records table with the Object Sharing table, and the Object Sharing table with the Group Maintenance tables. Salesforce queries the joined tables for access grants that give the querying user access to the record.



# Implicit Sharing

# Implicit Sharing

The sharing capabilities of the Lightning Platform include a wide variety of features that administrators can use to explicitly grant access to data for individuals and groups. In addition to these more familiar functions, there are a number of sharing behaviours that are built into Salesforce applications. This kind of sharing is called *implicit* because it's not configured by administrators; it's defined and maintained by the system to support collaboration among members of sales teams, customer service representatives, and clients or customers.

This table describes the different kinds of implicit sharing built into Salesforce applications and the record access that each kind provides.

Type of Sharing	Provides	Details
Parent	Read-only access to the parent account for a user with access to a child record	<ul style="list-style-type: none"> <li>• Not used when sharing on the child is controlled by its parent</li> <li>• Expensive to maintain with many account children</li> <li>• When a user loses access to a child, Salesforce needs to check all other children to see if it can delete the implicit parent.</li> </ul>
Child	Access to child records for the owner of the parent account	<ul style="list-style-type: none"> <li>• Not used when sharing on the child is controlled by its parent</li> <li>• Controlled by child access settings for the account owner's role</li> <li>• Supports account sharing rules that grant child record access</li> <li>• Supports account team access based on team settings</li> <li>• When a user loses access to the parent, Salesforce needs to remove all the implicit children for that user.</li> </ul>

Type of Sharing	Provides	Details
Portal	Access to portal account and all associated contacts for all portal users under that account	Shared to the lowest role under the portal account
High Volume	Access to data owned by high volume users associated with a sharing set for users member of the sharing set's access group	All members of the sharing set access group gain access to every record owned by every high volume user associated with that sharing set
High Volume Parent	Read only access to the parent account of records shared through a sharing set's access group for users member of the group	Maintains the ability to see the parent account when users are given access to account children owned by high volume users

# Territory Management

# Territory Management

When you are in charge of a sales team, you want everything to run smoothly as butter. You want your team to be the best and the sales to roll in, you want the business revenue to *grow like wildflowers*, and you want your sales team to be a happy one. Undoubtedly, that would have been the ideal scenario, the reality is often not as amazing. Your business targets may look almost impossible to achieve, your sales representatives may feel low on energy and motivation, and your sales may start to stagnate with no clear-cut reason. Guess what? It's now the ideal time to start addressing the problems by formulating a series of action changes to bring your sales team right back on the track with Salesforce Territory Management.

The best thing you can do is to analyse your sales territories, how they are individually and collectively performing, and how they get distributed. Once you're done with this, it would be time to have a close look at how exactly you are defining and redefining your sales territories. Are your territories based on customer groups where experts in one industry segment have one-on-one client meetings, regardless of their geographic location? If your answer is in the affirmative, it is suggested that you should map sales territories geographically to find out what you are missing out on by not mapping the territories.

Let us access some of the innumerable advantages that sales managers like you can explore by simply mapping out territories. But before that, let us first understand the concept of Salesforce Territory Management.

## **What is Salesforce Territory Management?**

Before we read about Salesforce territory management let us first read about Salesforce so that all of us are on the same knowledge platform.

Salesforce is a cloud-based customer relationship management (CRM) solution that brings customers and organisations together. It is ideal for organisations of all forms and sizes as it provides all departments, including but not limited to sales, commerce, marketing, and services with a shared, single view of every customer of your organisation.

Territory management in Salesforce is an excellent tool for revenue generation. Salesforce Territory Management can be described as an account sharing system that provides access to accounts based on their characteristics. By making use of Salesforce Territory Management, your organisation can easily structure and streamline data and users linked with Salesforce in ways similar to those associated with the structuring your sales territories.

It involves a geographic area or customer group over which either a sales team or an individual has responsibility. Usually, territories are defined based on the sales potential, its history, geography,

sales potential, customer names, competitive activities, or a combination of these factors. Sales Managers can maximise productivity and improve on the economies of scale in field sales by assigning sales territories to individual representatives. However, it is imperative that well-balanced and easily-manageable territories are established to ensure efficient usage of time and resources.

Territory management is critical to businesses of all size. By aligning sales teams to specific territories (industry, geographic, product-based), companies can make the most of their resources at the lowest cost. Aside from the boost in productivity, businesses are able to increase overall revenue by ensuring that all market segments are covered. Having an informed, data-driven plan in place allows companies to focus on growth and scaling up for the future.

In particular, small businesses can benefit from a thorough territory management plan, where resources and budgets are limited. You don't have to be a large enterprise company to strategise like one.

### **What is a Territory?**

Represents a flexible collection of accounts and users where the users have at least read access to the accounts, regardless of who owns the accounts. Only available if territory management has been enabled for your organisation.

### **Customisable Forecasting**

Since Customisation Forecasting is a prerequisite for Territory Management, we should understand this as well. Customisation Forecasting is a flexible solution for estimating how much revenue your organisation can generate or how many items your organisation can sell. You can set up customisation forecasting to reflect how your organisation forecasts its sales. With it, you can forecast on a monthly or quarterly basis, use different dates when applying amounts to forecasts, a forecast based on revenue or quantity or both, and define additional quotas based on product families.

Use customisation forecasts to review your forecast and drill down through your forecast amounts to see the opportunities included in your forecast. Override forecast amounts directly from the opportunity or overrides the forecast from the Forecasts tab without notifying users below you in the forecast role hierarchy.

### **Territory Management**

Territory management is an account sharing system that grants access to accounts based on the characteristics of the accounts. It enables your company to structure your Salesforce data and users the same way you structure your sales territories. Particularly if your organisation has a private sharing model, you may need to grant users access to accounts based on criteria such as postal code, industry, revenue, or a custom field that is relevant to your business. You may also need to generate forecasts for these diverse categories of accounts. Territory management solves these business needs and provides a powerful solution for structuring your users, accounts, and their associated contacts, opportunities, and cases.

When thinking about territory management, it's natural to assume that you need to use the Territory Management feature to model your territory hierarchy and manage your territory assignments. This assumption might be true for some customers in some situations, but all customers should consider this feature one option among the rich Salesforce feature set for meeting territory management requirements.

Territory Management is available only for organisations that use Customisation Forecasting; Territory Management is not supported for organisations that use Collaborative Forecasts. If Salesforce Customer Support enables Territory Management for your organisation, your organisation cannot use Collaborative Forecasts. For these reasons, enable, implement, and test Territory Management in a sandbox environment before enabling it in production. See *"Enabling Territory Management" in the Salesforce Help*.

### **Difference between Role Hierarchy and Territory Management**

The primary difference between the two is the question of "how many"?

In the role hierarchy, you are allowed one role as a user. So if I am in a sales hierarchy where it is only regional (East, Central, West, for instance), no problem. But what if I also have an industry vertical, and I need to be in both East, and the Manufacturing vertical?

This is the intended purpose of territories. In territories, I can be given record access based on two or more factors. I can be in one territory that identifies my geographical location, and the industry vertical and Account size (and as many other factors, potentially as I need). Territory management only affects accounts and the standard objects that have a master-detail relationship to accounts.

For example, opportunities are included in territory management but leads are not. So for record-level permissions (also sometimes referred to as the Sharing Model, record access, record visibility) for custom objects, you must use roles and the role hierarchy.

### **Territory management 2.0**

Use Territory Management 2.0 to manage and maintain your organisation's sales territories. Create territory types, build a model, and then add and test your account assignment rules. When you're satisfied with your model, activate it, then assign users and accounts. Roll it out to your organisation, and then run reports to assess its impact and make adjustments if necessary.

The reason why Sales Managers and the CMOs/CTOs investing in Salesforce are so excited about Territory Management is that it enables them to see and process their Salesforce data just as they would structure their actual sales territories. Imagine organising your sales responsibilities in different clusters so as to reduce the sales cost, provide more robust service to the customers and measure the performance of your sales team more effectively all under a single roof, called "Territory Management."

Originally, Territory Management enabled you to grant users access to accounts based on criteria relevant to your business. Territory Management 2.0 goes several notches higher and helps you model your sales territories into a logical but flexible structure that connects your sales teams to

the accounts they sell to. Having introduced territory types, territory models, and territory model states, Salesforce now lets you create and preview multiple territory structures and strategies, ensuring you always implement the one that works best for you. Your sales managers can then use custom reports to assess your territories for desired effectiveness.

If you are a sales manager, sales head of your organisation or basically have anything to do with sales and territories within your organisation, and you are still unsure about adopting this amazing feature, we are going to give you 5 reasons why you should change your mind now.

Territory Management 2.0 builds on that by introducing territory types, territory models, and territory model states. Use these tools to create and preview multiple territory structures and strategies before you activate and implement the one that works best. Custom reports help you organise your territory model for optimum coverage, assess territory effectiveness, and modify your model if necessary.

Territory Management 2.0 is API ready, with nine new objects that help you manage territories, assign records and assign users programmatically.

### **Things To Consider When Establishing A Sales Territory**

- Competition in sales territories
- Form of product or product line of your business
- Nature of the assignment and workload on the sales team
- Business's transport system and the channels of distribution
- Servicing requirements and essentials of territories

### **Sales Territory Management-Why It Is Important**

When sales territories get out of balance, there are two events that can happen. In the case of under-serviced territories, the salesperson or the sales team is spread thinly and this results in sub-optimal activity levels. Furthermore, those responsible for the territories would be overworked and will thus get less time with customers and therefore will seek out too few leads. In other words, the customers will leave the business and head to the competitors and this obviously means loss of sales opportunities.

Over-servicing in a territory is where the sales team has too many team members and too little work when it comes to serving a small area. This eventually raises operational costs and product prices that eventually results in reduced sales. Critical resources are not efficiently utilised and this results in under-servicing in other areas.

Territory imbalances can lead to different sorts of problems. Some of these include distorted compensation among representatives of your sales team, unfair distribution of sales potential among the sales team members, and reps quitting the work to find out reasonable and better compensation and work balance elsewhere.

### **Advantages Of Salesforce Territory Management**

Salesforce Territory Management is critical to the success of a business and its benefits are countless, including but not limited to increased responsiveness to emerging trends, efficient coverage of territories, and empowering your sales team. Territory Management is important to achieve targets, especially when sales managers have little to no extra resources to improve performance.

### **Increasing Sales And Reducing Cost With Enhanced Coverage And Aligned Territories**

The primary task before any sales organisation is to enhance the amount of time available with salespeople for selling products and services while ensuring the right product is sold to the right customer at the right place and price. Proper alignment of territories will result in improved career satisfaction, balanced workloads, and greater earning potential for salespeople. This, in turn, results in lower staff turnover, higher motivation, and more sales. Mapping of territories also offers route optimisation capabilities that can enhance fuel efficiency, minimise travel costs, and increase the number of customers that are served by the field sales teams within their respective territories.

### **Quick And Accurate Measurement Of Performance Through Analytics And Reporting**

A host of analytical tools can be availed to reap the advantages of numerous performance reporting options when territory data of sales is to be visualised using mapping software. The software enables plotted data on a map to be aggregated to get a consolidated view of performance. It can also be used for filtering and segmenting data, isolating sales below or above a certain value.

For better results, geographic and demographic data can be overlaid for identifying market insights and sales team members can individually create reports, analyse their specific territories, and measure their current performance against targets and quotas. These reports can then be shared with managers and colleagues if required.

### **Increased Productivity And Customer Satisfaction**

Every interaction and conversation with a customer impact their experience with your company and its products and services and thus directly correlates with customer satisfaction. No need to say, a good sales territory minimises travel costs and time and allows reps to spend more time dealing with the customers and less time on the road.

The ideal scenario is to have a close look at the abilities of strengths of your reps to be sure of the fact that they're servicing the "best" set of accounts. For instance, sales representatives who have a successful history with closing complex and large accounts should not be handling small leads but large accounts. The point is clear, reps should be playing to their strengths while fostering strong relationships and presenting a stronger value proposition. In the end, higher customer satisfaction rates can be achieved by creating targeted sales reps and customer relationships and increasing the selling time.

### **How To Create An Amazing Salesforce Sales Territory Plan For Your Team?**

Proper planning and execution behind sales territories can make a world of difference for the success prospects of your business. There are plenty of advantages of having a good sales territory plan whether you are starting from absolute scratch or looking to expand or redefine your existing strategy in the context of territory management.

Here is a step-by-step guide to create a profit-driven Salesforce Territory Management Plan for your sales team.

## **Segment And Analyse Existing Customers**

The first thing you need to do is to take stock of your existing leads, prospects, and clients before you can think about formulating an effective sales territory plan. For this, you can divide your business customers into segments based on the purchase history, vertical, location, buying preferences and patterns, or other relevant characteristics.

Once this has been done, you should note down the verticals with which your sales team is experiencing the biggest successes and how your existing customers are making purchases and what is their geographical location. You can even try out picking ten or twenty of your top prospects or customers and find out if there are common characteristics shared between them.

By now, you would have answers to the following questions:

- What are your customers buying?
- Why are your customers buying?
- How are your customers buying?
- What is the conversion rate of your business?
- Which events lead to purchases?
- What stops customers from buying from you?
- What pain points are resolved by your products or services?
- Are certain products, services, or solutions outperforming others? If yes, how?

## **Conduct a SWOT Analysis**

It is now time to conduct a basic SWOT (strengths, weaknesses, opportunities, and threats) analysis.

### **Strengths**

Analyse. Interpret. Execute. Which are the areas where your sales reps excel? Which are the areas where the reps require assistance or training? Maybe some of your team members are good over email and phone while others may be good to deliver engaging and delighting in-person demos.

### **Weaknesses**

Are there any leaks or bottlenecks in the pipeline that are hurting your business prospects? Is there any specific stage of the sales process where leads often start to lose interest?

## **Opportunities**

Are there untapped opportunities, markets, and customer segments? Is there an under-serviced territory where more of your sales representatives should sell?

## **Threats**

Who are your biggest and closest competitors? What are their promotional strategies and are they planning to introduce something big or unique? What are the biggest success threats in a specific territory?

## **Establish goals and targets**

It is always best to set precise, clear, and realistic goals and targets whether you are thinking to map a new territory plan for the entire sales team or revamping the sales territory of a specific representative. The easier the goals are to measure and track, the better it is for everyone!

For this, you need to ask yourself and the team the following questions:

1. Which geographical locations you should focus on?
2. Which products are the most profitable and which are not? What are the promotional strategies behind both?
3. What is the source(s) of a large majority of new leads?
4. Which customer segments are providing the highest payoffs?
5. Are any of the territories being under-serviced?

## **Develop a tangible and realistic strategy**

Once you have gained a clear and complete understanding of the goals and customer segments, it would now be time to start creating a plan with the representatives for smartly targeting the most lucrative avenues and derive the maximum possible coverage.

For this, you can use the information gathered from the above steps and match the skill sets of team members with the type of territory to be assigned or already assigned to them.

Representatives who have existing relationships with prospects should receive jurisdiction over those accounts.

## **Tracking results and optimising territory division**

Last but surely not least, it is time for you to put your new sales territory management plan into action. Moreover, you should also create plans to continually measure the performance and results to ensure there are little to no derivatives and results are always beyond expectations.

