Useful Queries

A book of queries for things that you want to know in SFDC.

Who has what permission?

A few housekeeping items

The permission set object is an amalgamation of Permission Sets and Profiles. You can determine if a record is a profile by using the "IsOwnedByProfile" field. You can also query on the permission set permissions to limit results.

SOAP API Developer Documentation

Field Level Security (FLS)

SOAP API Developer Documentation

Who has access to a field by a specific Profile?

```
SELECT Id, Field, PermissionsRead, PermissionsEdit, SobjectType, Parent.Profile.Name
FROM FieldPermissions
WHERE Parent.Profile.Name = 'System Administrator' AND Field = 'Account.Type'
```

Who has access to a field by specific Permission Set?

```
SELECT Id, Field, PermissionsRead, PermissionsEdit, SobjectType, Parent.Profile.Name,
Parent.Label
FROM FieldPermissions
WHERE Field = 'Account.Type'
```

Who has access to a specific field?

> The absence of records indicates that they have *no* access to that field.

```
SELECT Id, Field, PermissionsRead, PermissionsEdit, SobjectType, Parent.Profile.Name,
Parent.Label
FROM FieldPermissions
WHERE Field = 'Account.Type'
```

Who does not have access to a field?

```
SELECT Id, Label, Profile.Name
FROM PermissionSet
```

```
WHERE ID NOT IN (SELECT ParentID
                 FROM FieldPermissions
                 WHERE Field = 'Account.Type')
```

Object Level Security

SOAP API Developer Documentation

Who has access to an object by a specific Profile?

```
SELECT Id, Field, PermissionsRead, PermissionsCreate, PermissionsEdit, PermissionsDelete,
PermissionsViewAllRecords, PermissionsModifyAllRecords, SobjectType, Parent.Profile.Name
FROM ObjectPermissions
WHERE Parent.Profile.Name = 'System Administrator' AND SobjectType = 'Account'
```

Who has access to an object by a specific Permission Set?

```
SELECT Id, Field, PermissionsRead, PermissionsCreate, PermissionsEdit, PermissionsDelete,
PermissionsViewAllRecords, PermissionsModifyAllRecords, SobjectType, Parent.Label
FROM ObjectPermissions
WHERE Parent.Label = 'Account Permission Set' AND SobjectType = 'Account'
```

Who has access to a specific object?

> The absence of records indicates that they have *no* access to that object.

```
SELECT Id, Field, PermissionsRead, PermissionsCreate, PermissionsEdit, PermissionsDelete,
PermissionsViewAllRecords, PermissionsModifyAllRecords, SobjectType, Parent.Label,
Parent.Profile.Name
FROM ObjectPermissions
WHERE SobjectType = 'Account'
```

Who does not have access to an object?

```
SELECT Id, Label, Profile.Name
FROM PermissionSet
WHERE ID NOT IN (SELECT ParentID
                 FROM ObjectPermissions
                 WHERE SobjectType = 'Account')
```

Setup Entity Access

This object is for querying many object permissions in Salesforce. Those are:

Type	Object API Name	Name Field
Apex Class	ApexClass	Name
Visualforce Page	ApexPage	Name
Custom Metadata Type	EntityDefinition	QualifiedAPIName
Custom Setting	EntityDefinition	QualifiedAPIName
Applications (Apps in app launcher)	AppMenuItem	Name
Connected Applications	ConnectedApplication	Name
Custom Permission	CustomPermission	MasterLabel

[SOAP API Developer Documentation]

The process is the same for all of the above in the below examples. Just replace the object API name with the one for which you are looking to find permissions and the "Name" value in the where clause with the appropriate name field in the table above. For Custom Settings, use "IsCustomSetting" to filter. For Custom Metadata types, add QualifiedApiName LIKE '%__mdt' to the filter.

```
SELECT Parent.Label, Parent.Profile.Name
FROM SetupEntityAccess
WHERE SetupEntityID IN (SELECT Id
□□□□□□FROM ApexClass
                        WHERE Name = 'MyGreatApexClass')
```

Custom Tab Settings

> Pro-tip: You can update the settings from here with "Default On" or "Default Off". Deleting the row will make it "Hidden".

[Tooling API Documentation]

Name is prepended with "standard-" for standard objects. Name is the API name of a custom object.

```
SELECT Parent.Name, Parent.Profile.Name, Visibility, Name
FROM PermissionSetTabSetting
WHERE Name = 'standard-Account'
```

Profile Page Layout Settings

> The below are using the tooling API.

Tooling API Documentation

Standard Objects

```
SELECT Layout.Name, TableEnumOrId, Profile.Name, RecordType.Name
FROM ProfileLayout
WHERE TableEnumOrId = 'Account'
```

Custom Objects

First, retrieve the "Durable ID" of the object:

```
SELECT DurableId
FROM EntityDefinition
WHERE QualifiedAPIName = 'Account_Retention_Rate__c'
```

Then, query the Page Layout Settings:

```
SELECT Layout.Name, TableEnumOrId, Profile.Name, RecordType.Name
FROM ProfileLayout
WHERE TableEnumOrId = '01Io0000001KyaB'
```

License Management

Who has what Salesforce License?

```
SELECT Id, Name, UserName, Profile.Name, Profile.UserLicense.MasterLabel
FROM User
WHERE IsActive = true
```

How Many Licenses do I have?

```
SELECT
MasterLabel,MonthlyLoginsEntitlement,MonthlyLoginsUsed,Name,Status,TotalLicenses,UsedLicenses,
UsedLicensesLastUpdated
FROM UserLicense
```

How Many Managed Package Licenses do I have?

```
SELECT NamespacePrefix,Status,UsedLicenses,AllowedLicenses,ExpirationDate
FROM PackageLicense
```

Who has what Managed Package License?

```
SELECT UserId, PackageLicense.NamespacePrefix, CreatedBy.Name,CreatedDate
FROM UserPackageLicense
```

> You must download this information and then do excel VLOOKUP or INDEX(MATCHES()) formula to join the data in excel.

How many Permission Set Licenses do I have?

```
SELECT MasterLabel, DeveloperName, ExpirationDate,Status,TotalLicenses,UsedLicenses
FROM PermissionSetLicense
```

Who has what Permission Set License?

```
SELECT Id, Assignee.Name, Assignee.UserName, PermissionSetLicense.MasterLabel
FROM PermissionSetLicenseAssign
WHERE Assignee.IsActive = true
```

Approval Processes

Housekeeping

Here is the ERD for the Approval Process Objects. Typically reporting will be done from the Process Instance Step => Process Instance => Process Definition or from the Process Instance => Process Definition.

Process Instance Steps

Process Steps show any pending, completed, or recalled approval process steps.

SOAP API Developer Documentation

Retrieve all Process Steps that are Pending Approval

```
SELECT Id, ProcessInstance.TargetObjectId, CreatedBy.Name, Actor.Name, OriginalActor.Name,
ProcessInstance.Status, ProcessInstance.TargetObject.Type, ElapsedTimeInDays,
ElapsedTimeInHours, ElapsedTimeInMinutes
FROM ProcessInstanceWorkitem
WHERE ProcessInstance.Status IN ('Pending', 'Started')
```

Retrieve all Process Steps that are Pending Approval and apply to a specific object

```
SELECT Id, ProcessInstance.TargetObjectId, CreatedBy.Name, Actor.Name, OriginalActor.Name,
ProcessInstance.Status, ProcessInstance.TargetObject.Type, ElapsedTimeInDays,
ElapsedTimeInHours, ElapsedTimeInMinutes
FROM ProcessInstanceWorkitem
WHERE ProcessInstance.Status IN ('Pending', 'Started') AND ProcessInstance.TargetObject.Type =
'Opportunity'
```

Getting Metrics on Approval Process Completion

Process Instance SOAP API Developer Documentation

Process Definition SOAP API Developer Documentation

```
SELECT
ProcessDefinition.Name,SUM(ElapsedTimeInDays),SUM(ElapsedTimeInHours),SUM(ElapsedTimeInMinutes
```

```
),COUNT(Id),Status,TargetObject.Type
FROM ProcessInstance
WHERE ProcessDefinition.State = 'Active' AND Status IN ('Approved','Rejected')
GROUP BY ProcessDefinitionId, ProcessDefinition.Name,Status,TargetObject.Type
```

Content Documents

Housekeeping

ERD for Content Documents

When a Content Document is created, a content version record is created. Each update gets stored as a version and the content document LatestPublishedVersionId is updated. When linking a content document with a record, a ContentDocumentLink is created. A Content Document can be associated with multiple records in the system.

ContentNotes are *only* notes which are stored as well in the ContentDocument object. You can filter ContentNotes using FileType = 'SNOTE' in the ContentDocument object.

ContentDocumentLinks *must* be filtered by either a single ContentDocumentId or LinkedEntityId(s). If you do not, you will receive this error:

"Implementation restriction: ContentDocumentLink requires a filter by a single Id on ContentDocumentId or LinkedEntityId using the equals operator or multiple Id's using the IN operator."

Querying for ContentDocuments Linked to Certain Kinds of Records

```
SELECT Id, ContentDocumentId, ContentDocument.Title, ContentDocument.Description,
ContentDocument.FileExtension, ContentDocument.FileType, ContentDocument.LastViewedDate,
ContentDocument.LastReferencedDate, ContentDocument.PublishStatus,
ContentDocument.SharingOption, ContentDocument.SharingPrivacy
FROM ContentDocumentLink
WHERE LinkedEntityId IN (SELECT Id
□□□□□□ FROM Opportunity
                         WHERE Name = 'My Amazing Opportunity')
```

Query for Count of Linked Entity Object Types

```
SELECT COUNT(Id), LinkedEntity.Type
FROM ContentDocumentLink
WHERE ContentDocumentId = '0695c000009ycrxAAA'
GROUP BY LinkedEntity.Type
```

Query for All Linked Entities for a Content Document

```
SELECT LinkedEntityId, LinkedEntity.Type
FROM ContentDocumentLink
WHERE ContentDocumentId = '0695c000009ycrxAAA'
```

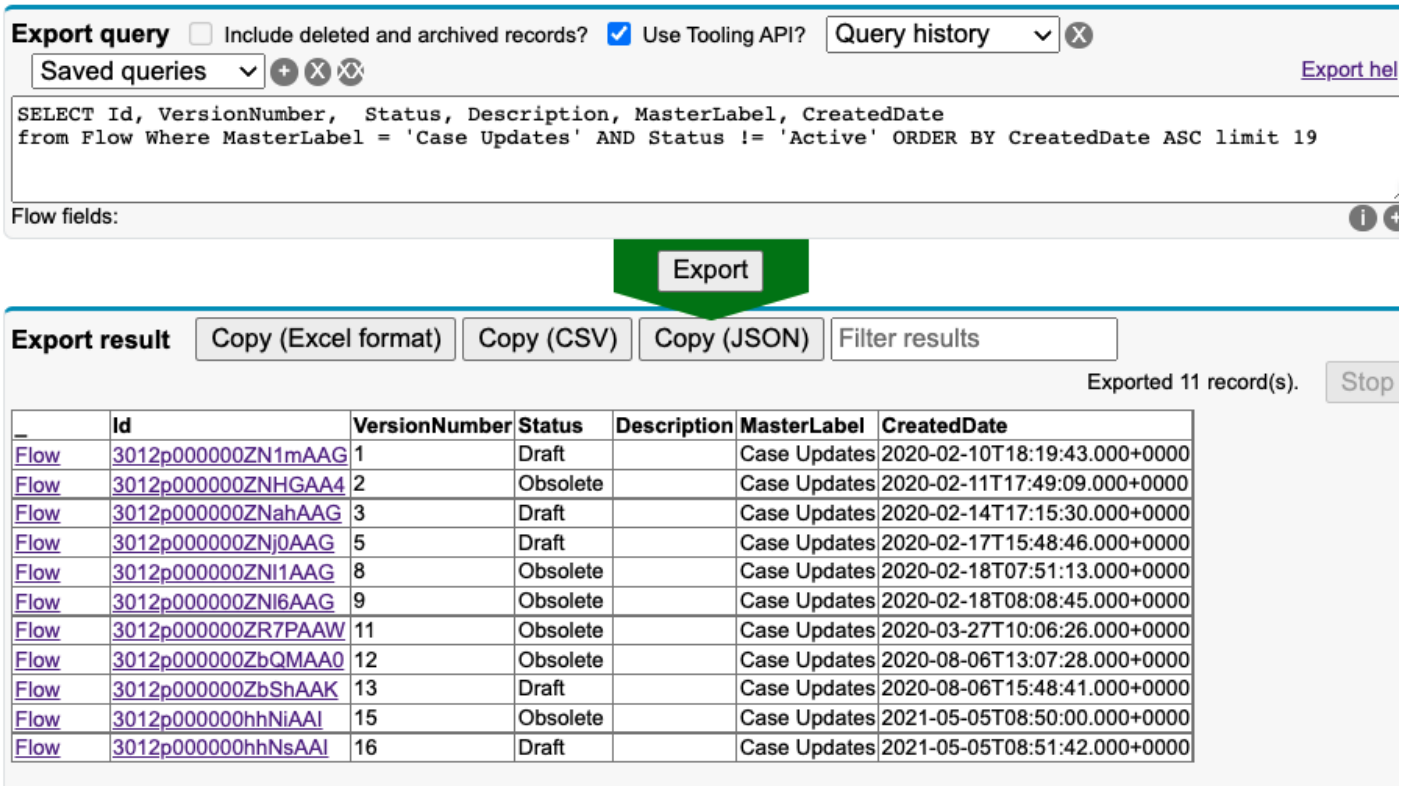
Querying Flows and Mass Deleting Old Versions

A Warning in Advance

Via the tooling api, we're going to delete some historic flow versions, this is super useful when you have reached the short limit of 50. However, you can lose useful history that past you might have wanted to save.

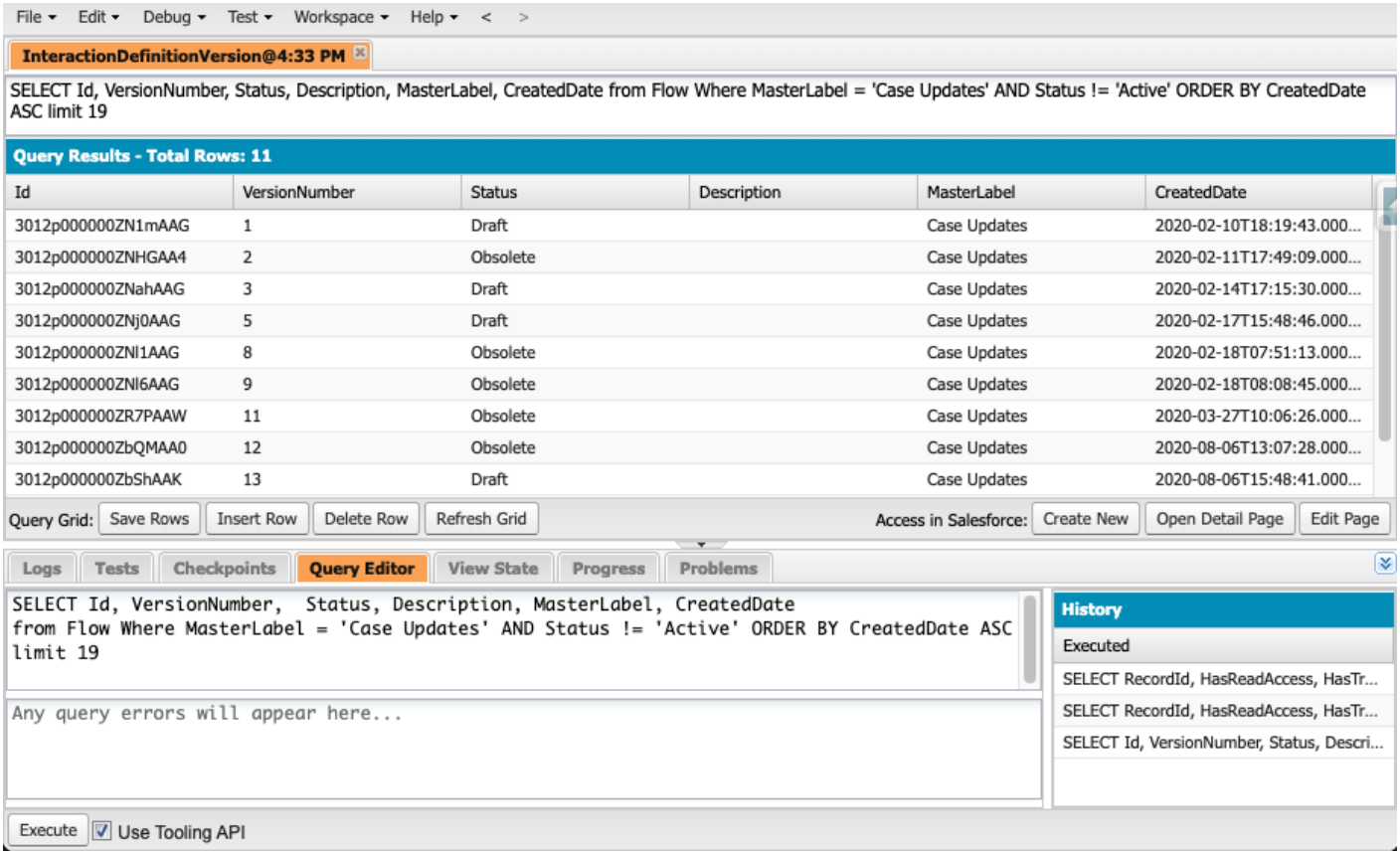
Accessing the Tooling API

You can access the tooling API via the Salesforce Inspector chome extension, there is a checkbox at the top which points you at the Tooling API:



_	Id	VersionNumber	Status	Description	MasterLabel	CreatedDate
Flow	3012p000000ZN1mAAG	1	Draft		Case Updates	2020-02-10T18:19:43.000+0000
Flow	3012p000000ZNHGAA4	2	Obsolete		Case Updates	2020-02-11T17:49:09.000+0000
Flow	3012p000000ZNahAAG	3	Draft		Case Updates	2020-02-14T17:15:30.000+0000
Flow	3012p000000ZNj0AAG	5	Draft		Case Updates	2020-02-17T15:48:46.000+0000
Flow	3012p000000ZNl1AAG	8	Obsolete		Case Updates	2020-02-18T07:51:13.000+0000
Flow	3012p000000ZNl6AAG	9	Obsolete		Case Updates	2020-02-18T08:08:45.000+0000
Flow	3012p000000ZR7PAAW	11	Obsolete		Case Updates	2020-03-27T10:06:26.000+0000
Flow	3012p000000ZbQMAA0	12	Obsolete		Case Updates	2020-08-06T13:07:28.000+0000
Flow	3012p000000ZbShAAK	13	Draft		Case Updates	2020-08-06T15:48:41.000+0000
Flow	3012p000000hhNiAAI	15	Obsolete		Case Updates	2021-05-05T08:50:00.000+0000
Flow	3012p000000hhNsAAI	16	Draft		Case Updates	2021-05-05T08:51:42.000+0000

You can also access the Tooling API via the developer console:



[Tooling API Developer Documentation]

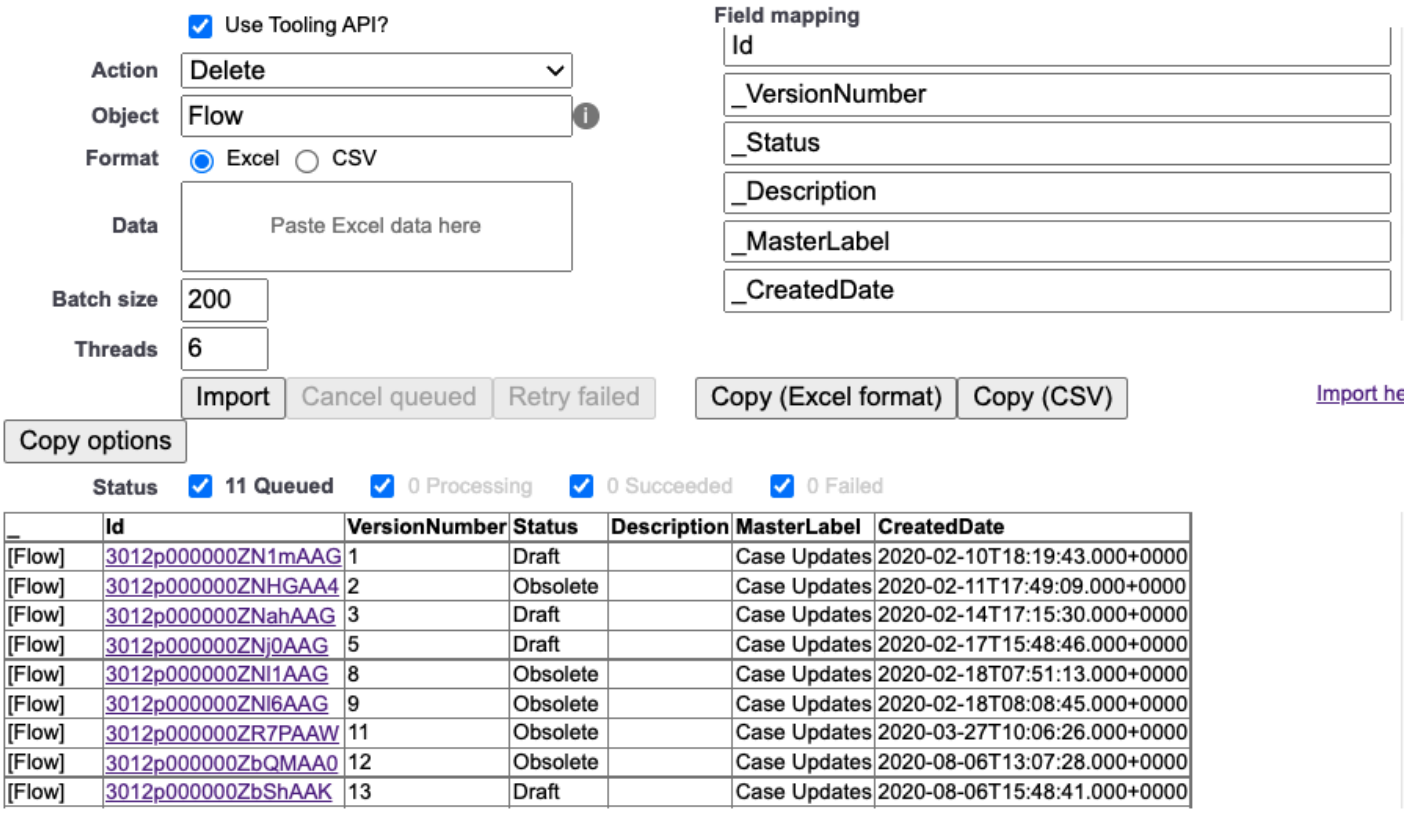
What versions could I remove?

```
SELECT Id, VersionNumber,  Status, Description, MasterLabel, CreatedDate
FROM Flow
WHERE MasterLabel = 'Case Updates' AND Status != 'Active' ORDER BY CreatedDate ASC limit 19
```

Generally limit yourself to one flow at a time, for ease this example limits via the Masterlabel, as per the above image, you can see that it returns both drafts and obselete versions. Beware, you do not want to remove your current draft with changes in it!

Deleting versions

In Salesforce Inspector, you can simply copy the excel output from the query and paste that into the upload box of a data import window:



_	Id	VersionNumber	Status	Description	MasterLabel	CreatedDate
[Flow]	3012p000000ZN1mAAG	1	Draft		Case Updates	2020-02-10T18:19:43.000+0000
[Flow]	3012p000000ZNHGAA4	2	Obsolete		Case Updates	2020-02-11T17:49:09.000+0000
[Flow]	3012p000000ZNahAAG	3	Draft		Case Updates	2020-02-14T17:15:30.000+0000
[Flow]	3012p000000ZNj0AAG	5	Draft		Case Updates	2020-02-17T15:48:46.000+0000
[Flow]	3012p000000ZNl1AAG	8	Obsolete		Case Updates	2020-02-18T07:51:13.000+0000
[Flow]	3012p000000ZNl6AAG	9	Obsolete		Case Updates	2020-02-18T08:08:45.000+0000
[Flow]	3012p000000ZR7PAAW	11	Obsolete		Case Updates	2020-03-27T10:06:26.000+0000
[Flow]	3012p000000ZbQMAA0	12	Obsolete		Case Updates	2020-08-06T13:07:28.000+0000
[Flow]	3012p000000ZbShAAK	13	Draft		Case Updates	2020-08-06T15:48:41.000+0000

Note the Tooling API checkbox at the top, if using the above query, you will need to skip all the other column headers, as the delete operation expects a list of Id's.

It doesn't appear that dev console can delete flows unfortunately.

When you don't follow best practices or just need to do some cleaning

Queries to help you find what to yeet

Layouts without page layout assignments (using Tooling API in Developer Console):

```
SELECT Id, Name, EntityDefinition.MasterLabel
FROM Layout
WHERE ID NOT IN (SELECT LayoutId
□□□□ FROM ProfileLayout)
```

Permission Sets with less than 20 active assignments:

```
SELECT COUNT(id), PermissionSet.Label
FROM PermissionSetAssignment
WHERE Assignee.IsActive = TRUE AND PermissionSet.IsOwnedByProfile = FALSE
GROUP BY PermissionSet.Label
HAVING COUNT(Id) < 20
ORDER BY COUNT(Id) DESC
```

Profiles with less than 5 active assignments:

```
SELECT COUNT(id), PermissionSet.Profile.Name
FROM PermissionSetAssignment
WHERE Assignee.IsActive = TRUE AND PermissionSet.IsOwnedByProfile = TRUE
GROUP BY PermissionSet.Profile.Name
HAVING COUNT(Id) < 5
ORDER BY COUNT(Id) DESC
```

List Views that haven't been viewed in 90 days:

```
SELECT Id, DeveloperName,Name,SobjectType,LastReferencedDate
FROM ListView
```

```
WHERE LastReferencedDate < LAST_N_DAYS:90
```

Email Templates that haven't been used in 90 days or haven't ever been used:

```
SELECT Id,Name, Folder.Name,LastUsedDate,TimesUsed
FROM EmailTemplate
WHERE LastUsedDate < LAST_N_DAYS:90 OR TimesUsed = null
```

Reports that haven't been run in 90 days or haven't ever been used:

```
SELECT Id, Name, FolderName, LastViewedDate, LastReferencedDate
FROM Report
WHERE (LastViewedDate < LAST_N_DAYS:90 AND LastReferencedDate < LAST_N_DAYS:90) OR
  (LastViewedDate = null AND LastReferencedDate = null)
```

Dashboards that haven't been run in 90 days or haven't ever been viewed:

```
SELECT Id, Title, FolderName, LastViewedDate, LastReferencedDate
FROM Dashboard
WHERE (LastViewedDate < LAST_N_DAYS:90 AND LastReferencedDate < LAST_N_DAYS:90) OR
  (LastViewedDate = null AND LastReferencedDate = null)
```

Roles / Queues without assignments:

```
SELECT Name, Type, DeveloperName, RelatedId
FROM GROUP
WHERE Id NOT IN (SELECT GroupId FROM GroupMember) AND
  Type IN ('Role','Queue')
```

Scontrols (if you have these, you really need to look at org health):

```
SELECT Id, DeveloperName
FROM Scontrol
```

Queries for things that you need to update

Apex Classes not in the most recent 3 versions (update API version each release):

```
SELECT Name, APIVersion
FROM ApexClass
WHERE APIVersion <= 50.0 AND NamespacePrefix = null
```

Visualforce pages not in the most recent 3 versions (you may also want to look at making these LWC's):

```
SELECT Name, APIVersion
FROM ApexPage
WHERE APIVersion <= 50.0 AND NamespacePrefix = null
```

Aura Components not in the most recent 3 versions (you may also want to look at making these LWC's):

```
SELECT DeveloperName, ApiVersion
FROM AuraDefinitionBundle
WHERE APIVersion <= 50.0 AND NamespacePrefix = null
```

> Locker was introduced with API version 39.0. Regression test if upgrading API version.

Lightning Web Components not in the most recent 3 versions (use tooling API):

```
SELECT DeveloperName, ApiVersion
FROM LightningComponentBundle
WHERE APIVersion <= 50.0 AND NamespacePrefix = null
```

Flows not in the most recent 3 versions (use tooling API):

```
SELECT MasterLabel, ApiVersion
FROM Flow
WHERE APIVersion <= 50.0 AND Definition.NamespacePrefix = null
```

Permissions Best Practices

Review users who have Customize Application, Modify All Data and View All Data

Unique Counts of Users

Customize Application

```
SELECT COUNT_DISTINCT(AssigneeId)
FROM PermissionSetAssignment
```

```
WHERE Assignee.IsActive = true AND PermissionSet.PermissionsCustomizeApplication = true
```

Modify All Data

```
SELECT COUNT_DISTINCT(AssigneeId)
FROM PermissionSetAssignment
WHERE Assignee.IsActive = true AND PermissionSet.PermissionsModifyAllData = true
```

View All Data

```
SELECT COUNT_DISTINCT(AssigneeId)
FROM PermissionSetAssignment
WHERE Assignee.IsActive = true AND PermissionSet.PermissionsViewAllData = true
```

Review Users and the permission sets or profiles that grant them access

Customize Application

```
SELECT PermissionSet.Label, PermissionSet.Profile.Name, Assignee.Name
FROM PermissionSetAssignment
WHERE Assignee.IsActive = true AND PermissionSet.PermissionsCustomizeApplication = true
```

Modify All Data

```
SELECT PermissionSet.Label, PermissionSet.Profile.Name, Assignee.Name
FROM PermissionSetAssignment
WHERE Assignee.IsActive = true AND PermissionSet.PermissionsModifyAllData = true
```

View All Data

```
SELECT PermissionSet.Label, PermissionSet.Profile.Name, Assignee.Name
FROM PermissionSetAssignment
WHERE Assignee.IsActive = true AND PermissionSet.PermissionsViewAllData = true
```

> Also, in case you haven't run it recently, run the optimizer report! It'll help you identify areas for improvement beyond the above.

> If you have premier or signature support, you may also want to look into the org health assessment accelerator.