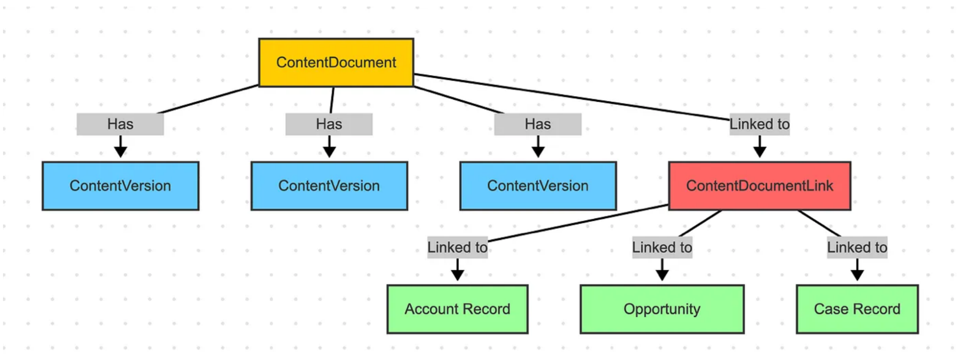
Salesforce File Management 101

Pranav Nagrecha 5 min read · Feb 14, 2025

12



Content Document Relation's

Introduction

Salesforce's file management system is built on three key objects: **ContentDocument**, **ContentVersion**, and **ContentDocumentLink**. By understanding how these objects work together, organizations can prevent confusion, eliminate redundancy, and ensure users always have access to the latest version of a file.

In this guide, we will break down these objects using an intuitive **library analogy**, explore their structure and relationships, and provide best practices for efficient file management.

The Core Concepts: Library Analogy

To simplify Salesforce's file management system, let's compare it to a library:

- **ContentDocument** → The book record in the library catalog (metadata about the book)
- **ContentVersion** → Different editions of the book (each update is a new edition)
- **ContentDocumentLink** → The library card allowing people to check out the book (sharing access to different users or records)

By using this analogy, we can understand how Salesforce manages files dynamically while maintaining a structured version history and access control

1. ContentDocument: The Master Record

Definition: The **ContentDocument** object serves as the **master record** for a file in Salesforce. It contains metadata about the file but does **not** store the actual file data.

Key Characteristics:

- **Acts as a container for multiple versions** of a document, with each version stored in **ContentVersion**.
- **Metadata storage**, such as **title, owner, and file type**.
- **Controls access** by linking to multiple records through **ContentDocumentLink**.
- **Deletion impacts all versions**, removing associated comments, ratings, and tags.
- **Triggers execute on file actions**, such as **insert** (when a file is added) and **delete** (when a file is removed).
- **Query limitations**: Only **queryAll()** retrieves archived documents, while **query()** does not.
- **Daily upload limits**: **Enterprise-level editions**: 200,000 new versions per 24 hours. **Developer & trial editions**: 2,500 new versions per 24 hours.

Key Fields:

- **Id** — Unique identifier for the ContentDocument.
- **Title** — The file name.
- **FileType** — The format of the file (PDF, DOCX, etc.).
- **LatestPublishedVersionId** — The most recent version of the file.

- **OwnerId** — The user who owns the file.
- **IsPrivate** — Boolean flag to mark a file as private.
- **ParentId** — The record or library to which the document belongs.
- **SharingPrivacy** — Defines privacy settings related to file access. (N — Visible to anyone with RECORD access & P — Private)
- **PublishStatus —** Indicates if and how the document is published. (P — The document is published to a public library and is visible to other users & R — The document is published to a personal library and is not visible to other users.)

Salesforce ContentDocument Documentation

2. ContentVersion: The File's Data and History

Definition: The **ContentVersion** object stores the actual file data and manages version history.

Key Characteristics:

- **Each update creates a new version**, ensuring historical tracking.
- **Supports querying and searching** for specific versions.
- **Files can be private** and **access-restricted** based

on user permissions.

- **Supports draft vs. published versions**, controlled via **PublishStatus**.

Key Fields:

- **Id** — Unique identifier for the ContentVersion.
- **ContentDocumentId** — Links to the parent ContentDocument.
- **VersionNumber** — Indicates the version (1, 2, 3, etc.).
- **IsLatest** — Boolean field that marks the most recent version.
- **VersionData** — Stores the actual file data.
- **PathOnClient** — The original file name/path upon upload.
- **PublishStatus** — Indicates whether the file is published ("P") or in draft mode ("D").
- **ContentLocation** — Determines whether a file is stored internally or externally. **Picklist Values: S** — Stored in Salesforce, **E** — Stored Externally
- **Origin** — Specifies the file's source. **Picklist Values: H** — Created via Chatter, **L** — Uploaded from local system, **P** — Created via a Salesforce process

Salesforce ContentVersion Documentation

3. ContentDocumentLink: Sharing and Access Control

Definition: The **ContentDocumentLink** object allows a single file (**ContentDocument**) to be associated with multiple records (Accounts, Opportunities, Cases, etc.). It defines who can access the document and at what permission level.

Key Characteristics:

- **Manages file associations** between a document and multiple records.
- **Controls access levels**: Viewer, Collaborator, or Inferred.
- **Determines file visibility**, either internally or externally.
- **Does not trigger on deletion**, meaning removing a link does not delete the file itself.
- **Query behavior differs**: Some shared files are only visible via explicit SOQL queries.

Key Fields:

- **LinkedEntityId** — The record ID (e.g., Account, Opportunity) the file is linked to.
- **ShareType** — Determines access level ('V' for

Viewer, 'C' for Collaborator, 'I' for Inferred).

- **Visibility —** Controls document access ('AllUsers', 'InternalUsers', 'SharedUsers').

Source: Salesforce ContentDocumentLink Documentation

FAQ

1. Isn't a ContentDocumentLink related to ContentVersion ?

- No, **ContentDocumentLink** is **not directly related to ContentVersion** — it links to **ContentDocument** instead.

Clarification:

- **ContentDocument** acts as the **main file record.**
- **ContentVersion** stores **actual file data and different versions** of the same document.
- The **ContentDocumentLink** is created at the **ContentDocument level**, which means **all versions** of the document are shared when the document is linked to a record.

This means that **if a file is updated (a new**

ContentVersion is created), the linked records will always have access to the latest version automatically.

2. If a ContentDocument is deleted, do all ContentVersions and ContentDocumentLinks get deleted too?

- Yes. When a **ContentDocument** is deleted, all its related **ContentVersions** (the actual file data) and **ContentDocumentLinks** (associations to records) are **automatically deleted** as well. This is because **ContentVersion and ContentDocumentLink cannot exist without a ContentDocument.**

3. Can you share different versions of a file with different records using ContentDocumentLink?

- No. **ContentDocumentLink only associates records with ContentDocument, not individual ContentVersions.** This means that when you link a file to a record using ContentDocumentLink, all records will **always** have access to the latest version of the document. You cannot link **Version 1 to one record** and **Version 2 to another** — the latest version is always the one that is shared.

4. Can you query ContentVersion directly to find out which records a file is linked to?

- No. **ContentVersion does not store record relationships.** To find out which records a file is linked to, you need to **query ContentDocumentLink**, since it holds the relationship between **ContentDocument and Salesforce records**.
  If you try to query **ContentVersion** for related records, you won't find any!

5. Can a user see a file if they have access to a record but not the ContentDocument?

- It depends on the **Visibility** setting in **ContentDocumentLink:**
- If **Visibility = "AllUsers"**, all users with access to the record can see the file.
- If **Visibility = "InternalUsers"**, only internal users can see it.
- If **Visibility = "SharedUsers"**, only users explicitly granted access will see it.

So, just because a user has access to a record **does not automatically mean they can see the file** — it depends on the **ContentDocumentLink visibility setting**.

Wrapping It Up

Managing files in Salesforce doesn't have to be a headache. With the right structure, storage strategy, and security settings, you can keep things organized and easy to find. A little effort upfront — like setting clear naming rules, using folders wisely, and automating where possible — goes a long way in keeping your system clutter-free.

Take some time to review how your team handles files today and see where small improvements can make a big difference. Whether it's cleaning up old files, tightening security, or integrating with an external storage tool, every step helps.

At the end of the day, good file management means less frustration, better collaboration, and smoother workflows. So why not start making those changes now?

Salesforce · Salesforce Development · Salesforce Tools · Salesforce Implementation

Written by Pranav Nagrecha

51 followers · 43 following

Edit profile

Salesforce Administrator/Developer/Architect/Business Analyst/QA

No responses yet

Pranav Nagrecha he

What are your thoughts?

More from Pranav Nagrecha

Pranav Nagrecha

Boomi—How to convert Timezones in a Process

I use Google to the extreme, I know there are smarter peopl…

May 17, 2022

Pranav Nagrecha

How We Built a Real-Time Integration…

Say goodbye to 15-minute polling delays—here's how w…

May 14 13

Pranav Nagrecha

Stop Guessing. Start Designing: Order of…

Your save isn't haunted—it's a kitchen line: before-save,…

Aug 26 2

Pranav Nagrecha

Roles, Profiles, and Permission Sets in…

Introduction

Aug 21 19

See all from Pranav Nagrecha

Recommended from Medium

Vaishnavi R

Database. Stateful in Salesforce

If you have ever worked with batch classes in Apex, you...

Jul 23 2

Musa Ndlala

Understanding Standard Objects and...

💭 The Moment You Realize Salesforce Isn't Just Data—It'...

Sep 15 5

Mani

The Invisible Apex Bug: Understanding and...

Two sales reps close the same opportunity at the same time....

Jul 11 · 6 · 1

Pedro Távora Santos

Why Your Agentforce POC Will Fail: An...

If you're reading this, you've probably sat through the...

Nov 18

Raja Saha

Generating Unique Identifiers in...

In Salesforce, unique identifiers play a vital role in...

Jun 30

Jeremy Carmona

How I Use Salesforce Templates to Save...

Third in the "Salesforce, Simplified" series

Jun 12 · 12.1K · 22

See more recommendations