Mastering EXISTS Statements

Your Dream Workfront Report EXISTS!

Skye Hansen | Senior Product Manager for Workfront - Starbucks
Nathan Johnson | Technical Support Engineer III - Adobe

Adobe

Two approaches, one mission: make EXISTS make sense.

ORIGIN STORIES

In 2018, Skye got permission to delete 10,000 projects. Months later, she started discovering empty portfolios… #consequences

When Nathan joined Support, his background in SQL landed him in the Reporting pod. On one of his first customer calls, the customer wanted to create a proof approval report that was filtered by project status.

Our Promise: EXISTS Will Finally Make Sense

1. **What is EXISTS??**
2. **Why EXISTS?**
   The real-world problems EXISTS statements solve.
3. **Nathan's Approach**
   Technical look at how EXISTS work and how to understand them
4. **Skye's Approach**
   A business-driven method to build EXISTS statements efficiently.
5. **Putting It All Together**
   Walk away ready to build your dream Workfront report.

Adobe

What is EXISTS? (context)

Type of filter	Reason to use	Examples
Beginner: Basic filters	• 1:1 (each item in your report has one “yes” answer to the filter) • 85% standard mode, 15% easy text mode	• Project report looking for projects created last week (each project has 1 entry date) • Task report looking for the primary assigned user (each task has 1 task “owner”
Intermediate: Collections filters	• 1:Many (each item in your report has more than one possible answer, at least one of which is “yes”) • 95% text mode, 5% cheat fields	• Program report looking for programs where the projects were created last week • Task report looking for all assigned users (each task can have multiple assignees)

Adobe

Advanced: EXISTS filters

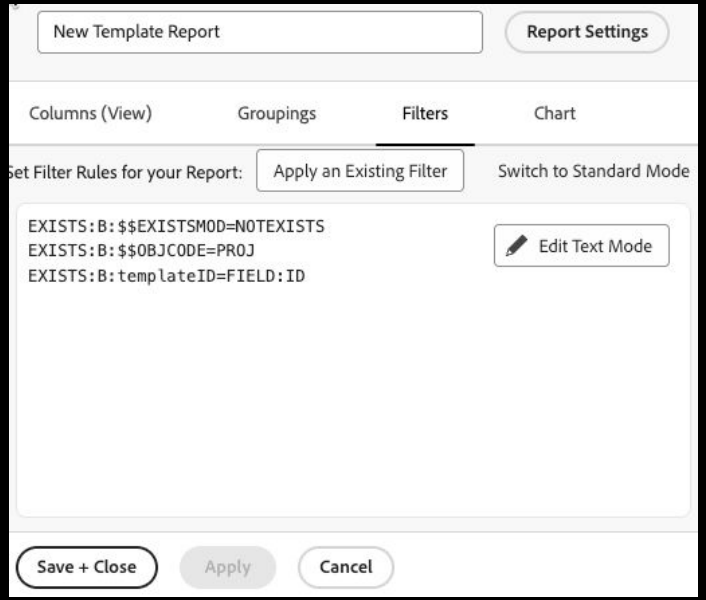
EXISTS:1:**$$OBJCODE=PROJ**
EXISTS:1:**tasks:ID=FIELD:ID**
EXISTS:1:**program:ownerID=$$USER.ID**

These are:

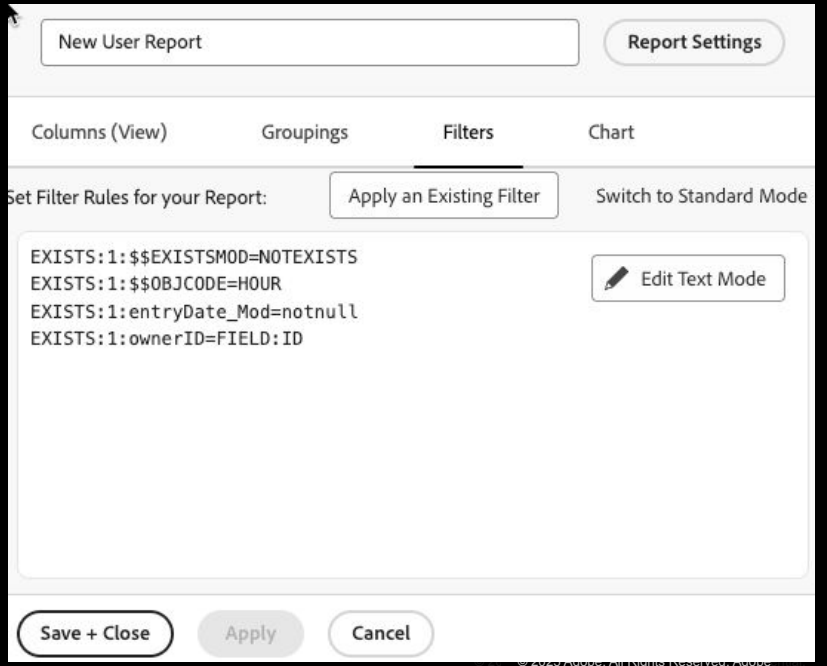
- Advanced, text mode filters, which allow us to work around limitations in standard reporting
- 100% text mode
- Many to many: items in one group are checked against items in another group; a match occurs when at least one pair meets the condition

What can EXISTS do for you?

Can you get me a list of all templates that have not been used. Users have too many choices, so we want to retire some.



We are spending too much on licenses. Can you determine who isn't using theirs.



Adobe

EXERCISE 1

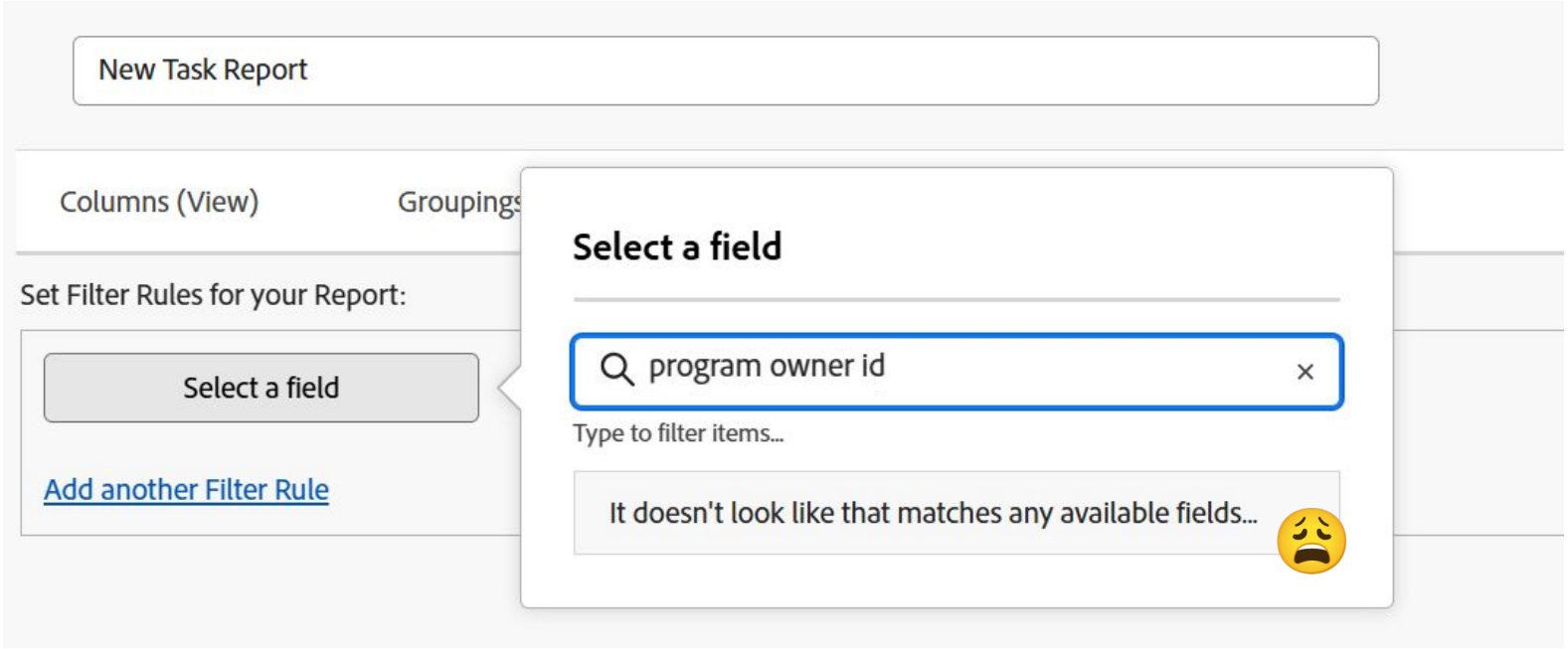
First challenge:

Word Problem: Show me TASKS that are in PROGRAMS that are OWNED by the LOGGED IN USER

First challenge:

Word Problem: Show me TASKS that are in PROGRAMS that are OWNED by the LOGGED IN USER

First attempt: Creating a normal filter



Second attempt: Creating a normal text mode filter

Handy Hint 1

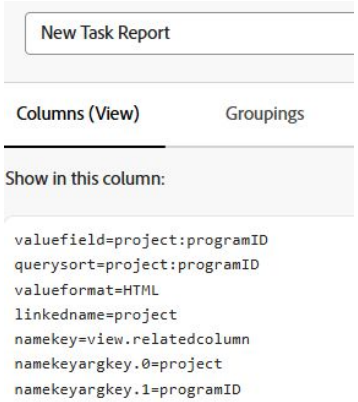
Avoid creating any text mode from scratch

We often create similar **views** and **filters** in **standard mode** and then change to **text mode** and “add on”

e.g. Looking for the Task’s Program’s Owner ID?

Starting Point:

Started with Task’s Project Program ID in standard mode; added on in text mode

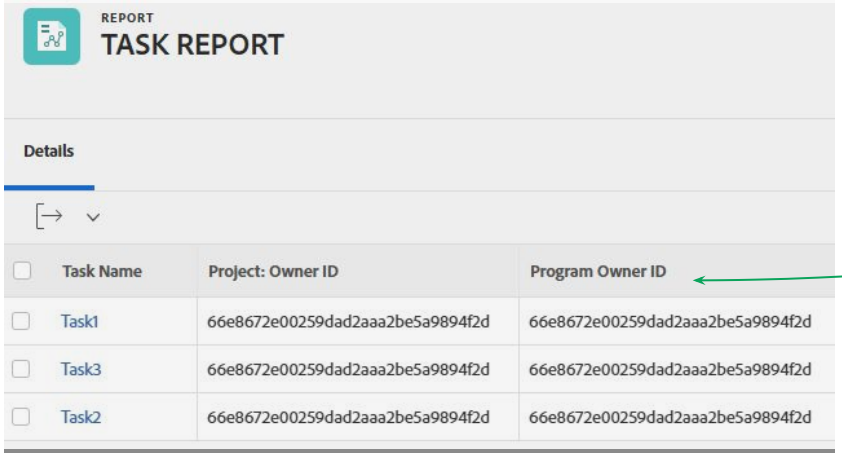


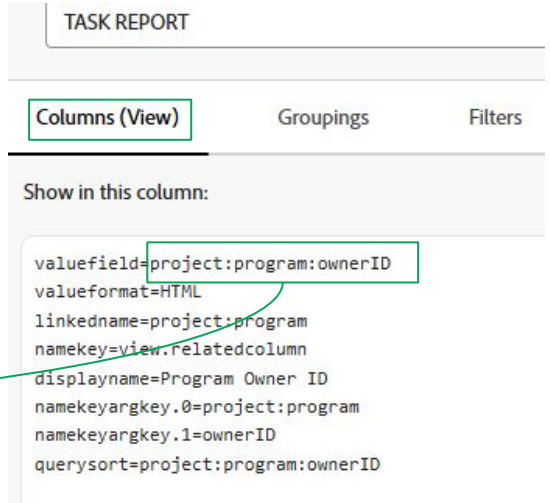
First challenge:

Word Problem: Show me TASKS that are in PROGRAMS that are OWNED by the LOGGED IN USER

Second attempt: Creating a normal text mode filter

Test your field in a column to see if it gives you results…





Adobe

First challenge:

Word Problem: Show me TASKS that are in PROGRAMS that are OWNED by the LOGGED IN USER

Second attempt: Creating a normal text mode filter

Our text mode gave us data in the column. Now try the exact same thing in a filter…

What the heck happened?? 😫

It’s Too Far Away
Why Columns Can See What Filters Can’t

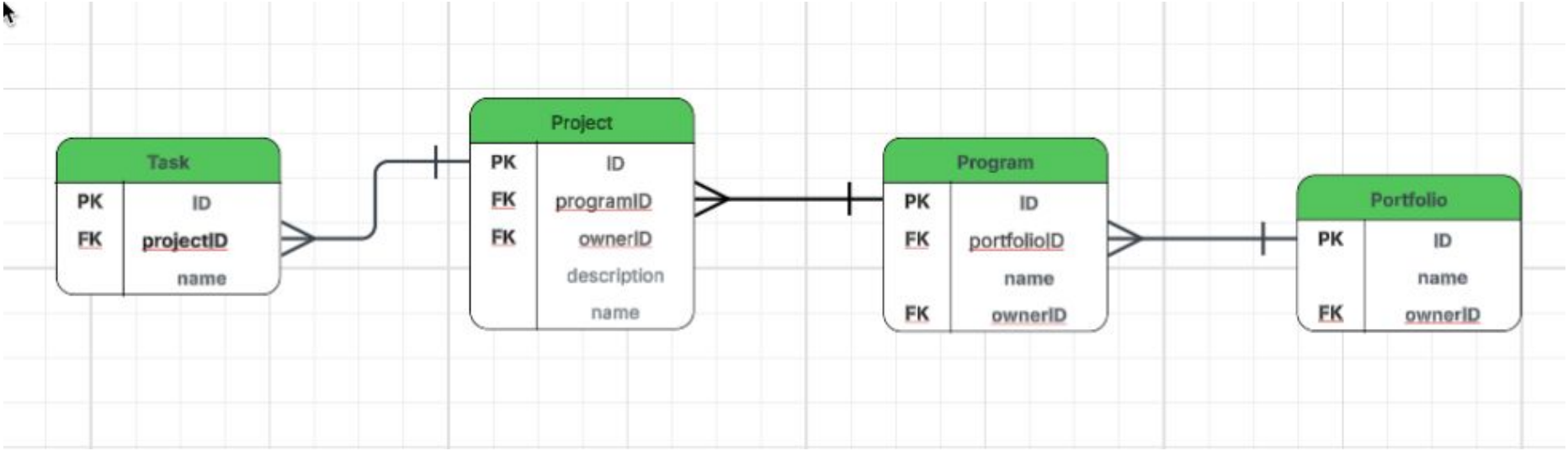
Why it works in Columns:

- Our engineers designed columns to allow up to 6 table jumps.
- Columns don’t filter rows; they just display fields that exist.

Why it fails in Filters:

- Our engineers designed filters to allow one table jump.
- We limit how many joins (table jumps) a filter can make for performance and data consistency.

How Objects Are Related

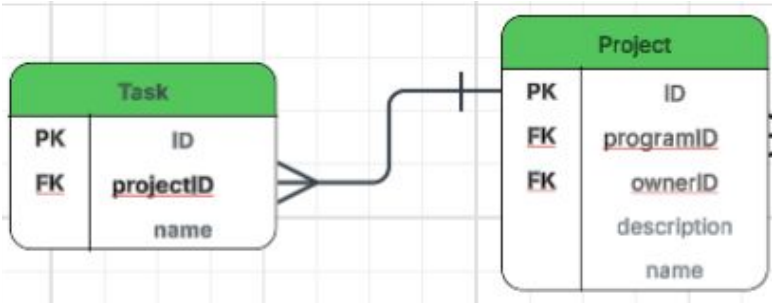


All objects in Workfront are linked to each other by their unique identifier.

- **Primary Key (PK):** A unique identifier for each object record — like a phone number. No two records share the same PK.
- **Foreign Key (FK):** A field on one object that links to the **Primary Key** of another. It creates a relationship between records.
- **One-to-Many:** A single project can connect to *multiple* tasks, but each of those tasks links back to only one project.

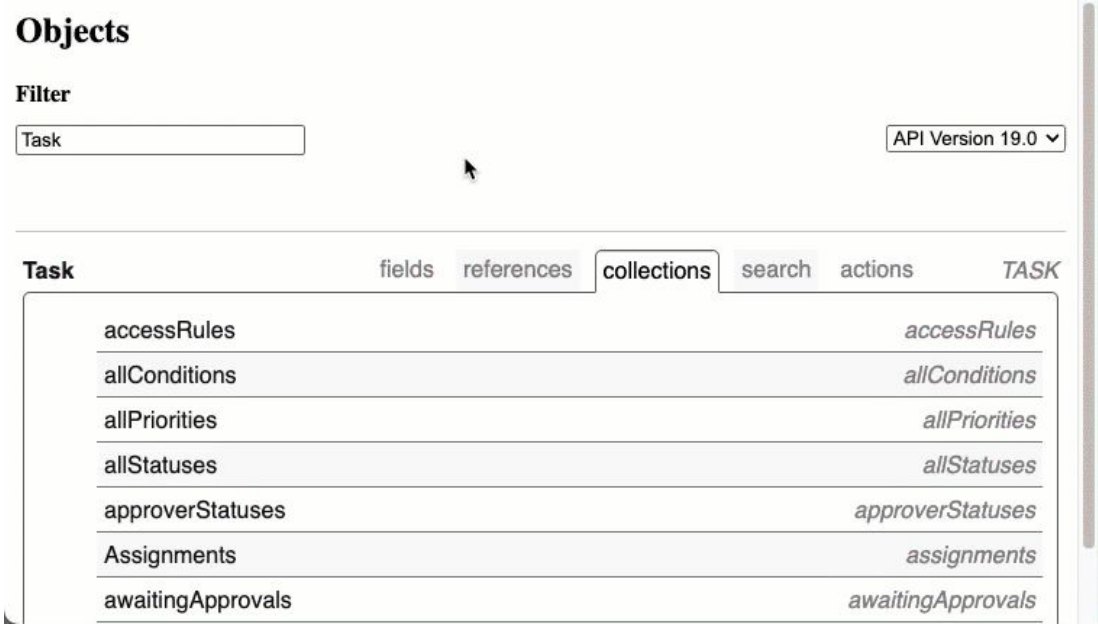
Adobe

Relationships in the API Explorer

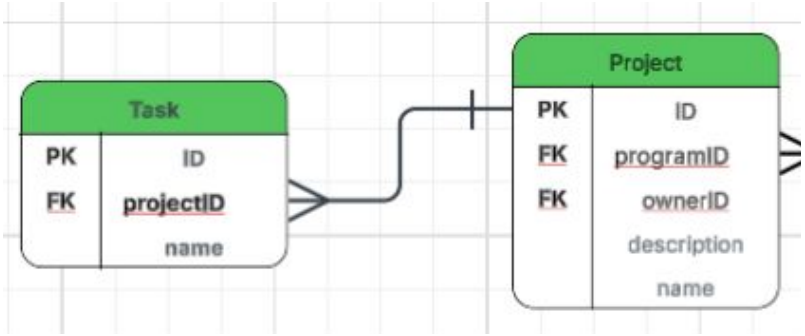


A **reference** is a *link to one specific related object* — a **foreign key** pointing *upward* to its parent. For example, a Task has a reference to its only Project

A **collection** is a *list of related child objects* — all the records that point *back* to this one via their foreign key. A Project has a collection of many Tasks.



An EXISTs filter requires us to choose a linking object so that the resulting filter does not violate the “one-table-jump rule”. We use the foreign key to unlock access to a table that is close enough to our target.



- EXISTS:1:**$$OBJCODE=PROJ**
    - Object Line - Program information cannot be reached from a Task report, but it can be from a Project report.
- EXISTS:1:**tasks:ID=FIELD:ID**
    - Linking Line - This is how I connect my report object type, Task, with the Project object type.
    - The right side of the equal sign uses FIELD as a stand in for the object I am on.
    - The left side is written from the linking object table.

Word Problem Reminder:

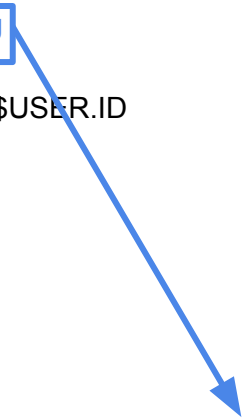
Show me TASKS (in projects) that are in PROGRAMS that are OWNED by the LOGGED IN USER

Now that we know how to link our objects, we only need the final line. This is our filter.

- EXISTS:1:**$$OBJCODE=PROJ**
- EXISTS:1:**tasks:ID=FIELD:ID**
- EXISTS:1:**program:ownerID=$$USER.ID**
    - Our filter. Now that we have successfully linked to the PROJECT object, we build a filter that would work on a project report
    - We are allowed one table jump in filters, so I can jump from project to program and filter on the ownerID.

Skye's Easy text mode Filter Hack

EXISTS:1:$$OBJCODE=**PROJ**
EXISTS:1:tasks:ID=FIELD:ID
EXISTS:1:program:ownerID=$$USER.ID



Information needed is in Project Report

1. Set aside your Task Report, open a new tab and separately create a Project Report.
2. Build your Project filter in standard mode *(also a good way to test your results).*
3. Switch over to Text Mode.
4. Copy and paste the text mode filter* into your exists statement (preface with "EXISTS:1:").

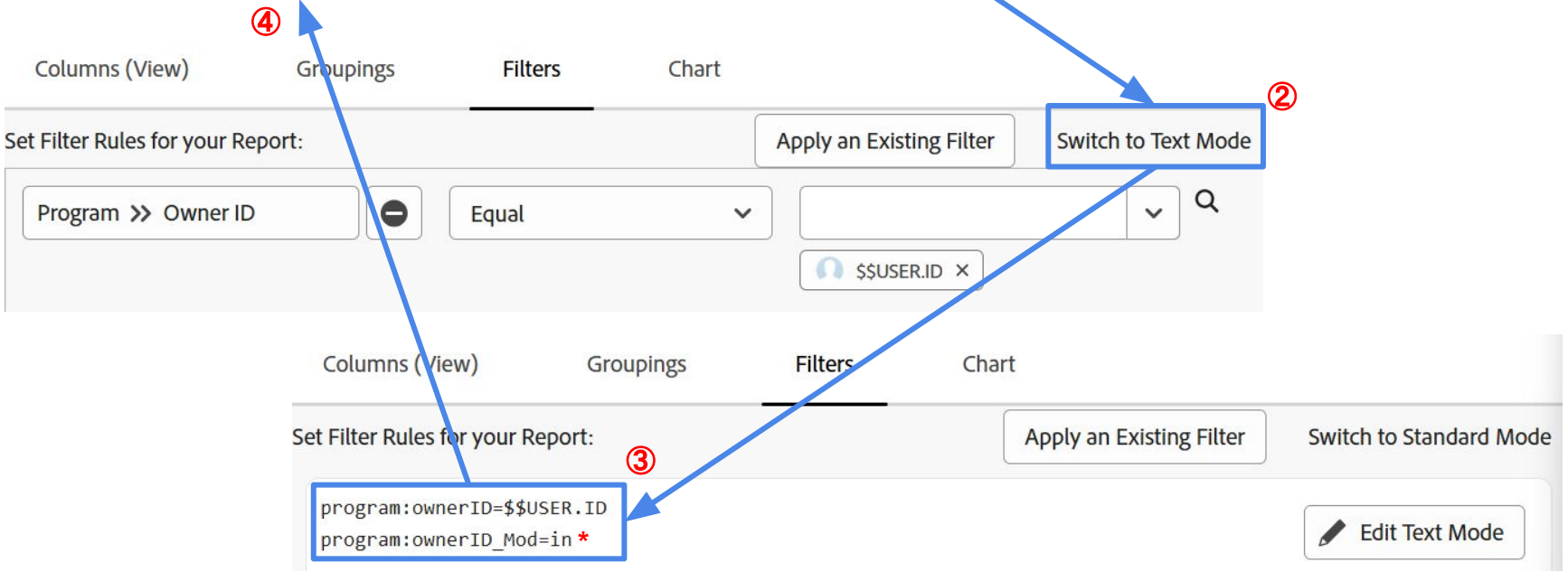
*** "in" is always assumed; no need to add**

Adobe

Skye's Easy text mode Filter Hack - Visualized

EXISTS:1:$$OBJCODE=**PROJ**
EXISTS:1:tasks:ID=FIELD:ID
EXISTS:1:**program:ownerID=$$USER.ID**

Create Project Report ①

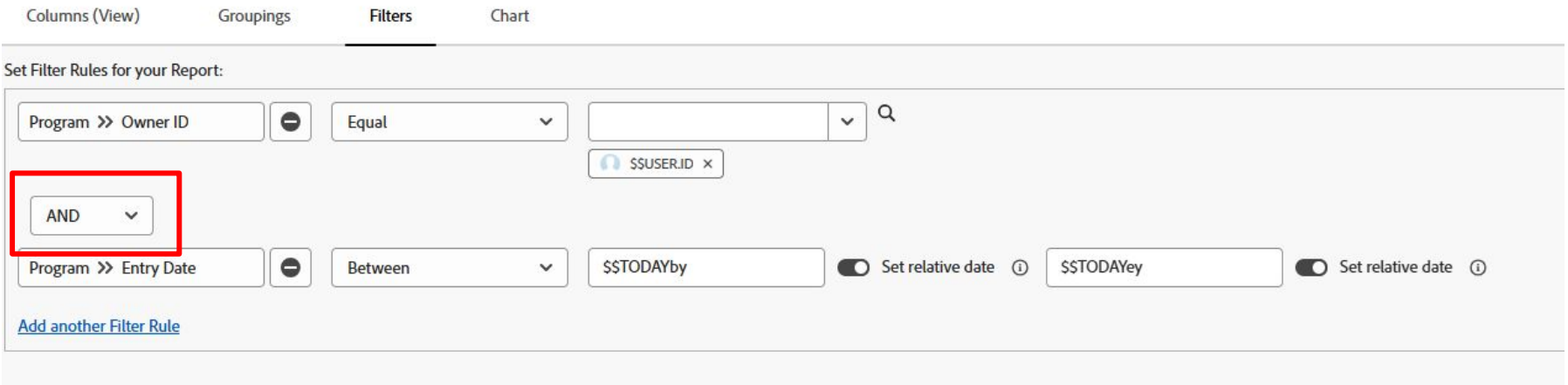


*** "in" is always assumed; no need to add**

Adobe

Skye's Easy Textmode Filter Hack v.2

Tasks in Projects where the you are the program owner AND the Project's program was created this year



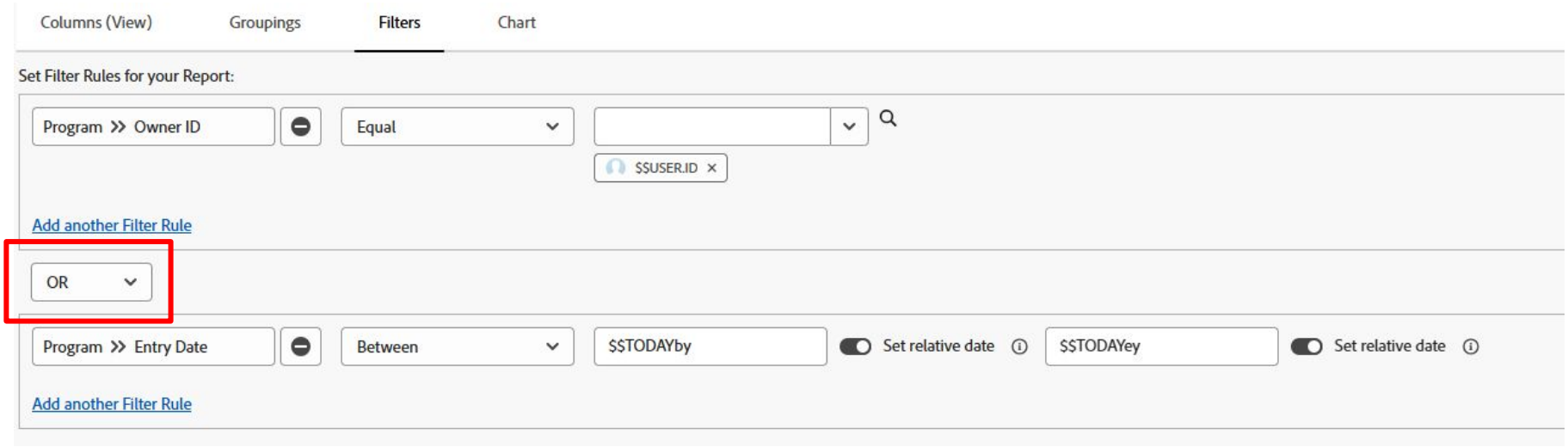
EXISTS:1:$$OBJCODE=**PROJ**
EXISTS:1:tasks:ID=FIELD:ID
EXISTS:1:**program:ownerID=$$USER.ID**

EXISTS:1:**program:entryDate=$$TODAYby**
EXISTS:1:**program:entryDate_Mod=between**
EXISTS:1:**program:entryDate_Range=$$TODAYey**

Adobe

Skye's Easy Textmode Filter Hack v.3

Tasks in Projects where the you are the program owner OR the Project's program was created this year



EXISTS:1:$$OBJCODE=**PROJ**
EXISTS:1:tasks:ID=FIELD:ID
EXISTS:1:**program:ownerID=$$USER.ID**

OR:1:EXISTS:1:$$OBJCODE=**PROJ**
OR:1:EXISTS:1:tasks:ID=FIELD:ID
OR:1:EXISTS:1:**program:entryDate=$$TODAYby**
OR:1:EXISTS:1:**program:entryDate_Mod=between**
OR:1:EXISTS:1:**program:entryDate_Range=$$TODAYey**

Adobe

Why is this so hard??

- Forgetting (or never realizing) that **EXISTS even exist.**
- Not knowing *when* to use it — or why your results don’t match expectations.
- Linking lines feel like dark magic.
- OBJCODE headaches — which object links to which?

Why EXISTS?

Reason to use EXISTS	Example ask	Translated to plain English
1. **Many to Many** (A project can be shared to many users, and a user can have many projects shared to them)	Show **projects** that are **shared** with a specific user (or team… or group… or role…)	Can you search through multiple sharing settings/rules from the project? (thing you are looking for doesn't have just one answer per object)
2. "**At least one / Just one**" (I don't want a list of projects, I want a list of owners)	Show **users** who have created at least one **project** in the past 90 days	I don't care about this list of projects – can you check to see who is using their license?
3. Check for "**something outside your immediate family**" (task connects to project, but how do I get to program?)	Show a **program owner** all the **tasks** in their program	Can you check for the task's project's program's owner ID? (sorry… what?)
4. Find something that isn't there (**doesn't exist**, wouldn't show up in a report to begin with)	Show **projects** that have **NO** completed **tasks**	Can you look for missing items? i.e. not looking for items that are there and blank – they are absolutely not there.

Adobe

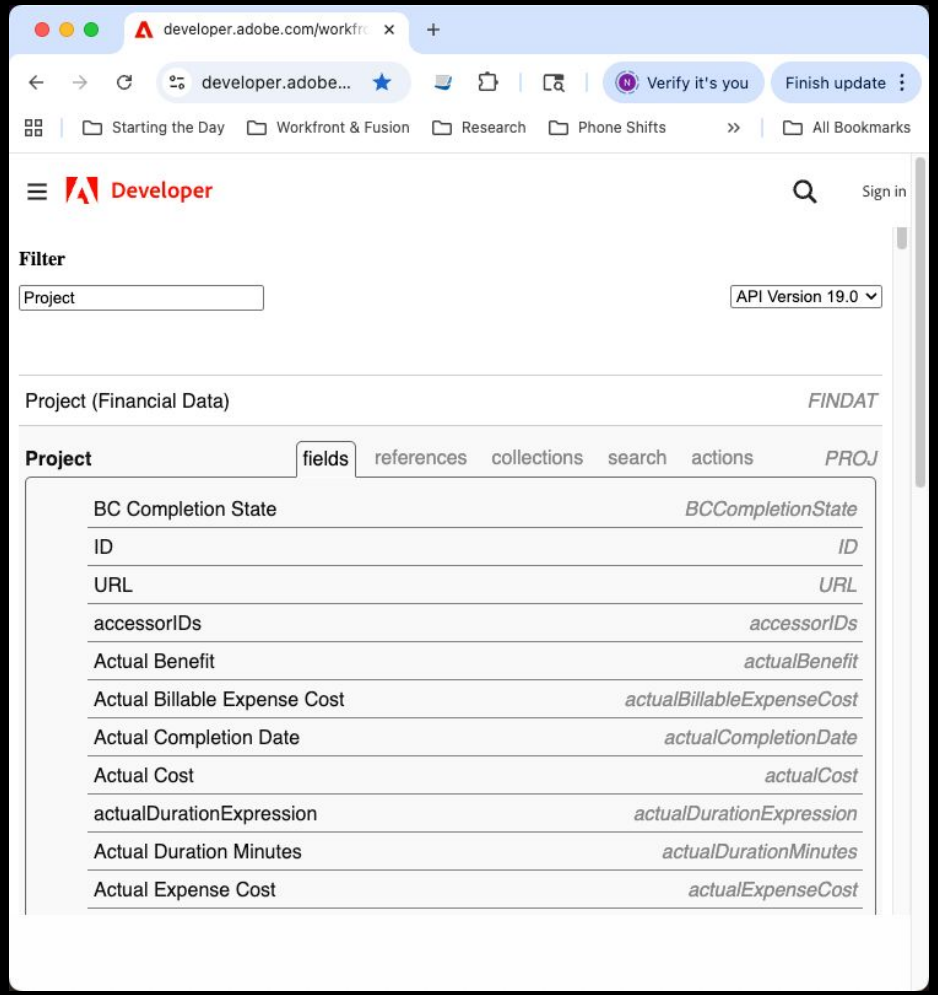
EXERCISE 2

Take a breath

Now Let's Level Up

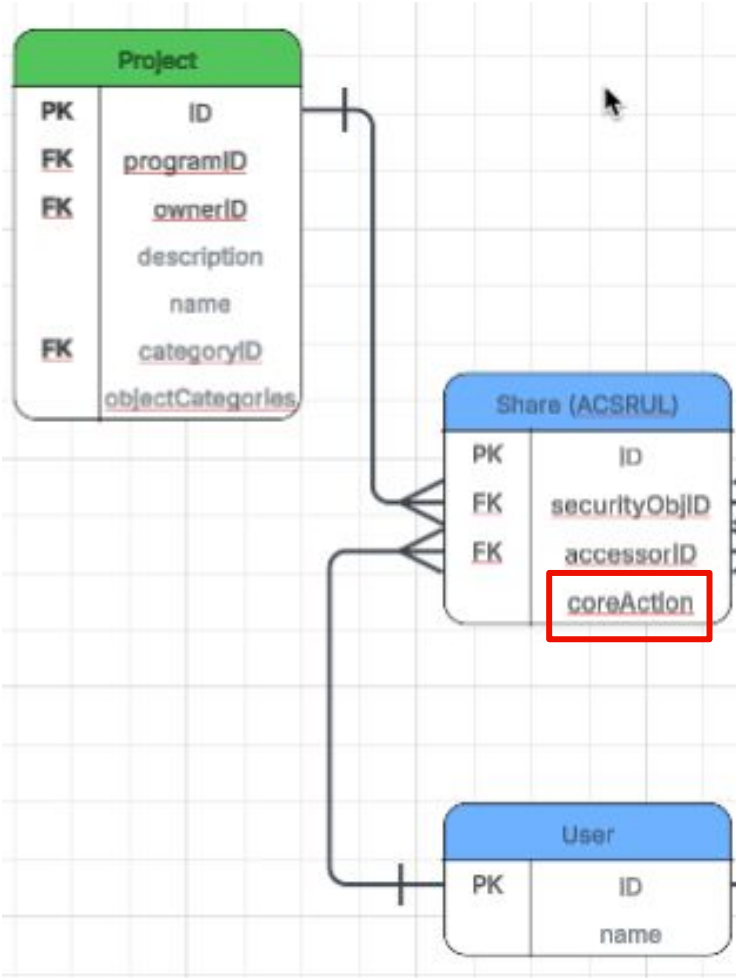
Word Problem: Show me all projects that are shared with a specific user.

It's a common problem - you're trying to find out what was shared, and to whom. Let me DEMO how I would start →



You need to have enough knowledge to know that a relationship between one project and multiple people means that we are looking at a relationship to MANY things (users) and so this will have to be a collection.

Adobe



Many-to-Many: In Workfront, a **user** can be shared to many **projects**, and each **project** can be shared with many **users.**

Since databases can't represent this directly, they use a **linking object** between the two.

Linking objects include assignments, objectCategories, userRoles, and so on. When you have a **many-to-many** relationship, the linking object line is easy.

EXISTS:A:**$$OBJCODE=ACSRUL**
EXISTS:A:**securityObjID=FIELD:ID**

Handy Hint 2

Reference collections in a view (Use collections text mode in your report view)

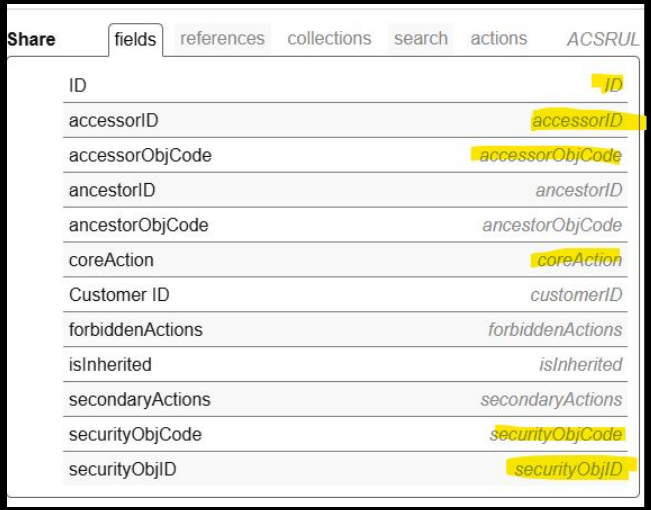
Instead of making educated guesses, just brute-force-add all the likely suspects into one column and search through the results to see which ones could add to your linking line or your filter lines.

Template: Collections text mode

valueformat=HTML
textmode=true
type=iterate
listdelimiter=<p>
displayname=Column Name
listmethod=nested(collection object name).lists
valueexpression=CONCAT(“field: ”,{field},“field2: ”,{field2})

Finding the Right Fields - Report Value Expression

Here goes nothing…



Edit Text Mode

```
displayname=Sharing Settings
listdelimiter=<p>
listmethod=nested(accessRules).lists
type=iterate
valueexpression=CONCAT('ID: ',{ID},' - accessorID: ',{accessorID},' - accessorObjCode: ',
{accessorObjCode},' - coreAction: ',{coreAction},' - securityObjCode: ',{securityObjCode},' -
securityObjID: ',{securityObjID})
valueformat=HTML
```

Details

Gantt | Report Default | Report Default | Nothing

Name	Project ID (Linking Line)	Who I'm Looking For (Filter)	Sharing Settings
Bill Test Project	639b99b10017a2b9d75f190c3b653b7b	560da5f1000c1d14de09352e70ff2205	ID: 639b99b10017a2e8ed5897174e44140d - accessorID: 560da5f1000c1d14de09352e70ff2205 - accessorObjCode: USER - coreAction: DELETE - securityObjCode: PROJ - securityObjID: 639b99b10017a2b9d75f190c3b653b7b

Adobe

I GOT IT!!!

Exercise 2 Solution

Show me all projects that are incorrectly shared with a specific user (i.e. they needed manage, not view permissions)

EXISTS:A:**$$OBJCODE=ACSRUL**
EXISTS:A:**securityObjID=FIELD:ID**
EXISTS:A:**accessorID=5c1e831f02dc1e4ae10d3701979cbd08**
EXISTS:A:**coreAction=VIEW**

NOTEXISTS

What if I only want things **without** that match? By adding one more line, we can turn the EXISTS filter into a NOTEXISTS filter.

Show me projects that have NOT been shared with Nathan. He is a super user and should have access to everything.

EXISTS:A:**$$EXISTSMOD=NOTEXISTS**
EXISTS:A:**$$OBJCODE=ACSRUL**
EXISTS:A:**securityObjID=FIELD:ID**
EXISTS:A:**accessorID=5c1e831f02dc1e4ae10d3701979cbd08**

Show me tasks in Programs that I do NOT own.

EXISTS:1:**$$EXISTSMOD=NOTEXISTS**
EXISTS:1:**$$OBJCODE=PROJ**
EXISTS:1:**tasks:ID=FIELD:ID**
EXISTS:1:**program:ownerID=$$USER.ID**

Why EXISTS? (clues)

Reason to use EXISTS	Example ask	Report / OBJCODE / link
1. Many to Many (A project can be shared to many users, and a user can have many projects shared to them)	Show **projects** that are **shared** with a specific user (or team… or group… or role…)	Report: Project OBJCODE: ACSRUL Link: securityObjID
2. “At least one / Just one” (I don’t want a list of projects, I want a list of owners)	Show **users** who have created at least one **project** in the past 90 days	Report: User OBJCODE: PROJ Link: enteredByID
3. Check for “something outside your immediate family” (task connects to project, but how do I get to program?)	Show a **program owner** all the (**project**) **tasks** in their program	Report: Task OBJCODE: PROJ Link: tasks:ID
4. Find something that isn’t there (doesn’t exist, wouldn’t show up in a report to begin with)	Show **projects** that have NO completed **tasks**	Report: Project OBJCODE: TASK Link: projectID

Tips

- **Search Community** – do a keyword search on the Adobe Experience League for unique OBJCODEs like ACSRUL or ARVSTS – these words don't exist (haha) in the english language so it's easier to pull up any EXISTS filters to help you get a quick start.
- **Make yourself a template you can paste as a starting prompt**, in your own words. Mine looks like this:

  In the **project report you are in**, show **projects**

Without	EXISTS:a:$$EXISTSMOD=NOTEXISTS
Tasks	EXISTS:a:$$OBJCODE=TASK
On the Project	EXISTS:a:projectID=FIELD:ID
Task named "test"	EXISTS:a:name=test

  This is because the fastest thing I remember is: on the task, it's called "projectID" and on the project, it's called "ID"
- **Start your own reference library**: keep samples of successful filters so you can mix and match. Always include a description!
- Shout out to keyboard macro tools such as Keyboard Maestro, Espanso or AutoHotkey, and reference library tools such as Airtable

Homework! 🤓

Week 1 — "Under the Hood" *(anchor → explore)*
Pick a type of report you build often (for example, a project, task, or issue report). Open the API Explorer, and locate that object. From there, trace out to collections that could expand what your report can show. How far can you get before you feel like EXISTS would be necessary? (*for beginners who need a safe starting point*)

Week 2 — "Getting to Carnegie Hall" *(practice, practice, practice)*
Run a search in the Adobe Experience League looking for EXISTS as a keyword in the Workfront questions forum, and without looking at the answer, try to come up with your answer independently. (*for learners who want to test themselves*)

Week 3 — "Chain Breaker" *(link → extended link)*
Find an object in the API Explorer that clearly links to another object (for example, Task → Assignment). Then push one step further: what happens if you need to filter based on the *next* link in the chain? Sketch how EXISTS could step in to make that connection. (*for learners ready to stretch their understanding*)

Week 4 — "Six Degrees of Kevin Bacon's Workfront" *(explore → anchor)*
Open the API Explorer and choose an object you don't normally work with. Can you trace a path from that object back to something you know well (like project, task, issue, user, or custom form)? If so, sketch how EXISTS might help you bridge the gap. (*for learners who enjoy puzzles or exploration*)

Week 5 — "Unsupported Explorer Hunt" *(unsupported → insight)*
Open the unsupported API Explorer and pick an object or field that doesn't appear in the supported one. What do you notice about it? Can you imagine a report where pulling that data in with EXISTS might be useful? What questions does this raise for you? (*for advanced learners or those curious about the backend*)

COME CHAT WITH US AT OUR AMA (ask me anything) IN DECEMBER!

Adobe