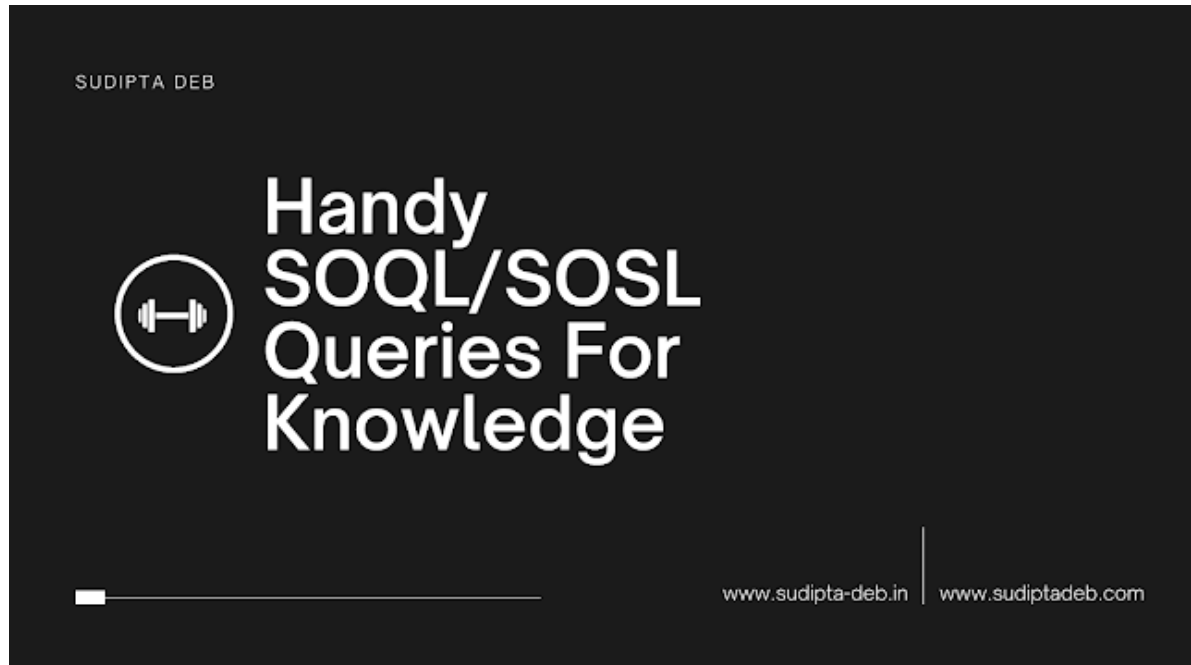


Handy SOQL/SOSL Queries for Knowledge

Sudipta Deb Monday, September 21, 2020



Happy Monday!! In my last post, I have explained the Data Model around Salesforce Lightning Knowledge. If you haven't gone through that post, I highly recommend you going through the post "[Understanding Salesforce Lightning Knowledge Data Model](#)" as it is going to help you to understand the queries in this post.

Let's first create the article - "Hello World" and below are the different versions with different publishing status. Below article properties which I will be using the query -

Article Number: 000001000

Knowledge Article Id: kA03t00000063w2CAA

Knowledge Hello World			
Article Number 000001000	Publication Status Draft	Last Modified Date 2020-09-18, 1:36 PM	Version Number 0
Details Related Versions			
VERSION	MODIFIED BY	ACTION	MODIFIED ON
Draft (viewing)	Sudipta Deb		Sep 18, 2020, 1:36 PM
3	Sudipta Deb		Sep 17, 2020, 10:39 AM
2	Sudipta Deb		Sep 16, 2020, 11:31 PM
1	Sudipta Deb		Sep 16, 2020, 11:31 PM

Fetching Latest Knowledge Article Version

```
SELECT Id, PublishStatus, Title FROM Knowledge__kav WHERE KnowledgeArticleId = kA03t00000063w2CAA
```

Fetching Published Knowledge Article Version

```
SELECT Id,PublishStatus,Title FROM Knowledge__kav WHERE KnowledgeArticleId = KA03T0000000053W2CAA
```

Fetching Draft Knowledge Article Version

```
SELECT Id,PublishStatus,Title FROM Knowledge__kav WHERE KnowledgeArticleId = KA03T0000000053W2CAA
```

Fetching Archived Knowledge Article Versions

```
SELECT Id,PublishStatus,Title FROM Knowledge__kav WHERE KnowledgeArticleId = KA03T0000000053W2CAA
```

This one is tricky, as to query Archived article versions, KnowledgeArticleId and IsLatestVersion = False and PublishStatus = archived should be mentioned.

Note -

Using bind variables with Apex SOQL Statements with KnowledgeArticleVersion is not allowed. So you need to use dynamic SOQL like below -

```
final String ONLINE_ARTICLE = 'Online';
final String myQuery = 'SELECT Id FROM Knowledge__kav WHERE PublishStatus = :ONLINE_ARTICLE';
List<Knowledge__kav> allArtcles = Database.query(myQuery);
```

Working with DATA CATEGORY

WITH DATA CATEGORY is an optional clause in SOQL and it helps to identify articles linked with one or more data categories.

You can use *WITH DATA CATEGORY* to fetch articles FROM *ObjectName* where

ObjectName = KnowledgeArticleVersion to query all articles types.

ObjectName = an article type API Name to query a specific article type.

ObjectName = Question to query questions.

Note - When *ObjectName = KnowledgeArticleVersion* or any article type API Name, then WHERE clause is mandatory. You can put one of the below things in the WHERE clause -

PublishStatus

"online" to published articles.

"archived" to query archived articles

"draft" to query draft articles

Id = articles Id.

SOQL Query -

```
SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus = Online WITH DATA CATEGORY NorthA
```

Note - *WITH DATA CATEGORY*, only AND logical operator is allowed.

DATA CATEGORY Filtering Selectors

Let's say we have the below data category structure -

```
Earth__c
  Asia__c
    India__c
    Bangladesh__c
  NorthAmerica__c
```

```
Canada__c
USA__c
Europe__c
Switzerland__c
France__c
German__c
```

Example 1: Fetch all published articles from Canada

```
SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus online WITH DATA CATEGORY Earth__
```

Example 2: Fetch all published articles above German

```
SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus online WITH DATA CATEGORY Earth__
```

This will fetch articles linked with German, Europe, Earth data categories.

Example 3: Fetch all published articles below Asia

```
SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus online WITH DATA CATEGORY Earth__
```

This will fetch articles linked with Asia, India, Bangladesh data categories.

Example 3: Fetch all published articles above or below Europe

```
SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus online WITH DATA CATEGORY Earth__
```

This will fetch articles linked with Europe, Switzerland, France, German, Earth data categories.