

“委任”と徹底的に向き合ってみた (2024年8月 検証の記録)

2024年8月31日(土)9月29日(日)

JPAUG広島「Power Appsでハッカソンしたいけえ #8」



自己紹介

- ・ ハンドルネーム : yukio
- ・ X (旧Twitter) : [@yukio 365](https://twitter.com/yukio365)
- ・ ブログ : <https://yukio-ms365.hatenablog.com/>



- ・ 山口県出身・在住、1990年生まれ
- ・ 県内の非IT企業勤務中の2022年3月頃にPower Platformに出会い、夢中で勉強した結果、社内で幾つものアプリ/フローを運用する市民開発者に
→2024年9月～Power Platform関連の仕事にフルリモ転職
- ・ 趣味はコントラバス演奏 (オーケストラ、吹奏楽団) 今はお休み中だがいつか復帰したい。

【チョットデキル】

Power Appsキャンバスアプリ
Power Automate(クラウドフロー)
Power Automate for desktop

【勉強中】

Power Appsモデル駆動型アプリ
Dataverse
Power BI

免責事項

- ・ 今回の情報は、
SharePointリストをデータソースにしたPower Appsキャンバスアプリにおける
“委任” について

個人的に検証した結果をまとめたものです（検証時期：2024年8月）

- ・ 不足や間違いを含む場合があります（むしろ参加者の皆様からご意見や経験談を頂いてこの資料を完成に近づけたいです。よろしく願いいたします。）
- ・ 今後のアップデートにより仕様が変わる可能性があります。
- ・ この資料はあくまで参考情報として見ていただき、実際の挙動はご自身で手を動かして確かめてください。

参考 (1) Microsoft公式、書籍

- ・ キャンバスアプリでの委任について (Microsoft Learn)

<https://learn.microsoft.com/ja-jp/power-apps/maker/canvas-apps/delegation-overview>

- ・ SharePoint - Connectors (Microsoft Learn)

<https://learn.microsoft.com/ja-jp/connectors/sharepointonline/#power-apps-delegable-functions-and-operations-for-sharepoint>

- ・ Power Apps入門 ～手を動かしてわかるローコード開発の考え方～ (出版：翔泳社 著：小玉純一/山田晃央)

<https://www.shoeisha.co.jp/book/detail/9784798173955>

- ・ ひと目でわかるPower Apps ～ローコードで作成するビジネスアプリ入門～ 第3版 (出版：日経BP 著：奥田理恵)

<https://bookplus.nikkei.com/atcl/catalog/23/08/01/00929/>

参考(2) 有志の登壇資料、ブログ他

- ・ あなたが作りたいアプリ、どのデータソースを選べばいいか教えます (Ryota Nakamura_りなたむ)

<https://www.docswell.com/s/rinatamu/KXQGPZ-PowerApps-delegate>

- ・ よく聞くけど、SharePointリストの5,000件問題ってなんなの？ (太田浩史)

<https://www.docswell.com/s/hrfmjp/5QNNN5-2022-05-09-124548>

- ・ SharePoint リストと比較した際の Microsoft Dataverse のメリット (Takashi_Masumori(益森 貴士))

https://qiita.com/Takashi_Masumori/items/4189db25d20538b91212

- ・ PowerApps | SharePointリスト5000件問題を解決する2つの方法 (七草あんこ)

<https://anko7793.com/2022/06/1697/>

- ・ Distinct関数は委任対象外なので気を付けよう。 (ヨウセイ)

<https://youseibubu.com/apps/distinct/>

“委任”ってなに？

Learnにはこう書いてある

委任を理解する

Power Apps Power Fx クエリを データソース で実行できる同等のクエリに完全に変換できる場合、バックエンド データソース で最もよく機能します。

Power Apps データソース が理解できるクエリを送信し、クエリは データソース で実行され、クエリ結果が Power Apps に返されます。たとえば、データソースは、データソース上のデータをフィルタリングする作業を実行し、フィルタ基準を満たす行のみを返す場合があります。これが正しく機能する場合、クエリの作業を実行するためにクエリが データソース に **委任** されたと言います。

Learnにはこう書いてある

ただし、Power Fx クエリは、すべてのデータソース上の同等のクエリに常に変換できるわけではありません。たとえば、Dataverse は Excel よりも多くのクエリ機能をサポートしています。Dataverse は 'in' (メンバーシップ) クエリ演算子をサポートしますが、Excel はサポートしません。データソースがサポートしていない機能をクエリが使用している場合、そのクエリは **委任不可能** であると言えます。一般に、クエリ式の一部が委任できない場合、クエリのどの部分も委任しません。

Learnにはこう書いてある

クエリが委任不可能な場合、Power Apps はデータソースから最初の500レコードのみを取得し、クエリ内のアクションを実行します。この制限は2,000レコードまで増やすことができます [制限の変更](#) Power Apps は結果のサイズが500レコードに制限され、Power Apps の良好なパフォーマンスを維持できます。実験の結果、結果セットがこれらのサイズを超えると、一般にアプリおよびPower Apps にパフォーマンスの問題が発生することがわかりました。

Learnにはこう書いてある

ただし、データソースのデータが 500/2000 レコードを超える場合、クエリが誤った結果を返す可能性があるため、この制限が問題になる可能性があります。たとえば、データソースに 1,000 万件のレコードがあり、クエリがデータの最後の部分进行操作する必要がある例を考えてみましょう。(たとえば、'Z' で始まる姓) ただし、クエリに委任不可能な演算子 (たとえば、distinct) が含まれています。この場合、最初の 500/2000 レコードのみが取得され、正しい結果ができません。

Learnにはこう書いてある

ただし、データソースのデータが 500/2000 レコードを超える場合、クエリが誤った結果を返す可能性があるため、この制限が問題になる可能性があります。たとえば、データソースに 1,000 万件のレコードがあり、クエリがデータの最後の部分进行操作する必要がある例を考えてみましょう。(たとえば、'Z' で始まる姓) ただし、クエリに委任不可能な演算子 (たとえば、`distinct`) が含まれています。この場合、最初の 500/2000 レコードのみが取得され、正しい結果がでません。

しれっと書いてあった

Distinct関数は委任対象外なので気を付けよう。(Power Apps Tips ログ)
<https://youseibubu.com/apps/distinct/>

Learnにはこう書いてある

あなたの Power Fx データ ソース の委任可能なテーブルを使用してクエリを実行します。委任できるクエリ関数のみを使用してください。それがアプリのパフォーマンスを高く維持し、ユーザーが必要なすべての情報にアクセスできるようにする唯一の方法です。

委任を使用できないことを示す警告が出た場合は注意してください。小規模なデータ セット (500 件未満のレコード) を操作する場合は、式を委任できなくてもアプリはデータをローカルに処理できるため、任意のデータ ソースと式を使用できます。

なんとなくーくわかる気がするが…

(でもこれ読み込んで理解できるようにならないといけないのが今後のわたし)

参考文献や有志の方々の資料を読んだ上で 自分なりに説明してみる

※事象を限定して説明しています。
※間違いを含む場合があります。あったら教えてください

【前提】 データ行の制限

- ・ Power Appsキャンバスアプリが、データソースから“一度に取ってこれる”最大の件数
- ・ 初期値は500件、最大で2000件まで増やせる



「データ行の制限」はなぜ存在する？

- ・ Copilotに聞いてみたらこんな回答だった

Power Appsのキャンバスアプリで「データ行の制限」が設けられている理由は、アプリのパフォーマンスを最適化し、ユーザーエクスペリエンスを向上させるためです。

具体的には、以下の理由があります：

1. パフォーマンスの向上: 大量のデータを一度に処理すると、アプリの動作が遅くなり、ユーザーエクスペリエンスが低下します。データ行の制限を設けることで、アプリがスムーズに動作するようにします [1](#) [2](#)。
2. 委任の利用: Power Appsでは、データのフィルタリングや並べ替えをデータソース側で処理する「委任」という仕組みがあります。これにより、アプリ側での処理負荷を軽減し、効率的にデータを扱うことができます [2](#)。
3. リソースの節約: デバイスのメモリやCPUリソースを節約するために、ローカルで処理するデータの量を制限します。これにより、アプリが安定して動作するようになります [1](#)。

デフォルトでは、データ行の制限は500件に設定されていますが、最大で2000件まで増やすことができます [3](#)。

SPO リストはPower Fxの全てを知らない

- 例えば、キャンバスアプリにギャラリーやデータテーブルを置いて、ItemsプロパティにFilter関数を書くとする。
- Power Appsは、基本的に自分でFilter処理はせず、データソース（SharePointリスト）にお願いする形をとっている。
- これが「委任する」ということ**
- しかし、SharePointリストは、Filter関数に書いてあることでも、知っていることと知らないことがある。

The screenshot shows the Power Apps interface. On the left, the 'Items' property of a gallery is set to a formula: `Filter(List_NoIndex, StartsWith(タイトル, TextInput1.Text))`. The gallery displays a table with the following data:

ID	タイトル	選択肢	日付
6	e5d		202
10	e23	あ	202
17	e88	う	202
26	efe	あ	202
38	e23	え	202

SPOがFxを理解してくれた場合

```
Filter(  
  List_NoIndex,  
  StartsWith(タイトル,TextInput1.Text)  
)
```

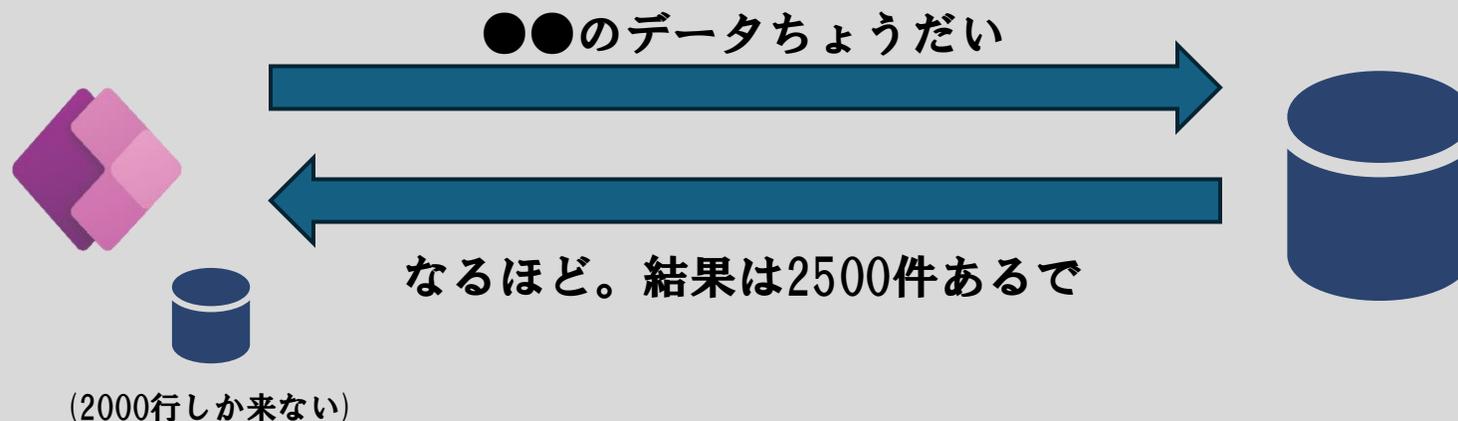
- ・ SharePointリストは、Power Appsからの指示を理解して、Filterした結果をPower Appsに返してくれる



ただし…

```
Filter(  
    List_NoIndex,  
    StartsWith(タイトル, TextInput1.Text)  
)
```

- ・ 委任できる関数でFilterした結果すらも500(2000)件を超えてしまうと、意図する結果にならない



SPOがFxを理解できない場合

```
Filter(  
  List_NoIndex,  
  EndsWith(タイトル, TextInput1.Text)  
)
```

委任に関する警告です。この数式の "EndsWith" の部分は、大きなデータ セットで正常に機能しない可能性があります。

委任に関する警告です。この数式の強調表示された部分は、大きなデータ セットの列 "タイトル" で正常に機能しない可能性があります。

問題を表示 (Alt+F8) 使用できるクイック修正はありません

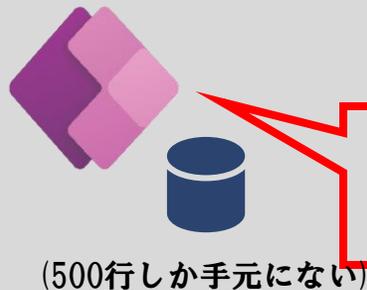
- ・ 黄色い波線で、理解できない旨を主張してくる (**たまに知ってるふりをする**)
- ・ SharePointリストは、Power Appsからの指示を理解できないので、データソース全件をPower Appsに返して「自分で検索せい！」と丸投げしてくる



SPOがFxを理解できない場合

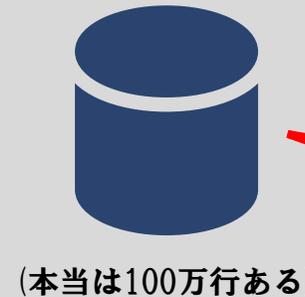


- ・ でも、先述の通り、Power Appsのキャンバスアプリが **“一度に取ってこれる” データ行は500行（設定で2000件まで増やせる）以下しかないため、全件と言いながら全件ではない可能性がある**
→全件でなかったら、意図する結果にならない



ん？これって本当にデータソース全件？

(500行しか手元にない)



何やさっきの条件指定知らんがな・・・

(本当は100万行ある)

SPOがFxを理解できない場合でも…

```
Filter(  
  List_NoIndex,  
  EndsWith(タイトル, TextInput1.Text)  
)
```

委任に関する警告です。この数式の "EndsWith" の部分は、大きなデータ セットで正常に機能しない可能性があります。

委任に関する警告です。この数式の強調表示された部分は、大きなデータ セットの列 "タイトル" で正常に機能しない可能性があります。

問題を表示 (Alt+F8) 使用できるクイック修正はありません

- ・ **でも、Power Appsは、取り込んだデータを自身でFilter処理することもできる。**
- ・ **データソースが全件渡してきても、それが「ちゃんと全件」だったらOKということになる。**

面倒臭いなあ
自分でFilterするか…



(ちゃんと全件)



言ってる意味が分からん
全件渡すから自分で検索せい！



(全体で50行しかない)

つまり・・・

「総件数が**絶対に**2000件いかない」
データソースであれば
実は委任問題は気にしなくても良い

(本末転倒だけどこれが本質)

(前職ではこのスタンスでアプリ作ってた)

今回作った検証用SharePointリスト

- ・ インデックスありとなし 一応2つ作成

委任検証用リスト2 ☆									
① ID ↑ ↓	📄 タイトル ↓	🗑️ 選択肢 ↓	📅 日付 ↓	📅 日付時刻 ↓	👤 ユーザー ↓	① 数値 ↓	🗒️ はい/い... ↓	📅 登録日時 ↓	
1	b16	え	2024/8/6	2024/8/7 5:21	yukio	35	✓	20:21 (火曜日)	
2	2e3	お	2024/8/6	2024/8/7 5:22	yukio	23	✓	20:22 (火曜日)	
3	d22	え	2024/8/6	2024/8/7 5:23	テスト 2	97	✓	20:23 (火曜日)	
4	043	い	2024/8/6	2024/8/7 5:24		15		20:24 (火曜日)	
5	45d	い	2024/8/6	2024/8/7 5:24	yukio	14	✓	20:25 (火曜日)	
6	e5d		2024/8/6	2024/8/7 5:26	yukio	93		20:26 (火曜日)	
7		う	2024/8/6	2024/8/7 5:27	テスト 1	100		20:27 (火曜日)	
8	90e	お	2024/8/6	2024/8/7 5:28	yukio	48	✓	20:28 (火曜日)	
9	657	う	2024/8/6	2024/8/7 5:29	yukio		✓	20:29 (火曜日)	
10	e23	あ	2024/8/6	2024/8/7 5:30	yukio	71	✓	20:30 (火曜日)	

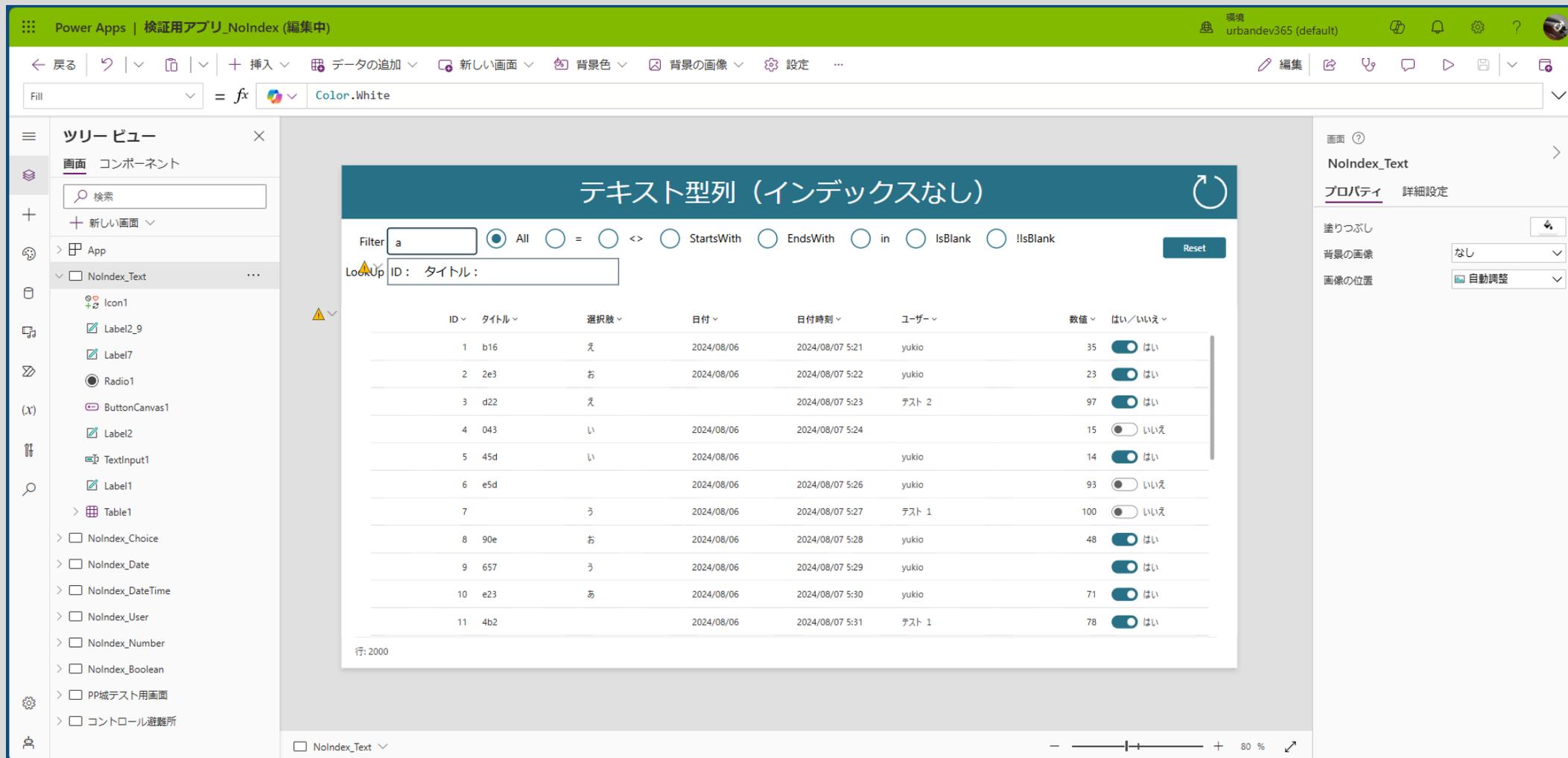
レコードはPower Automateで自動的に作成



こうしてできたリストの行数：
約38,000件（9/25現在）

検証用アプリ

- モダンコントロールの「テーブル」が大活躍（**実際のアプリの動きはイベント参加者限定**）



Power Apps | 検証用アプリ_NoIndex (編集)

環境: urbandev365 (default)

戻る | 挿入 | データの追加 | 新しい画面 | 背景色 | 背景の画像 | 設定

Fill = fx Color.White

ツリービュー

- 画面 コンポーネント
- 検索
- + 新しい画面
- App
 - NoIndex_Text
 - Icon1
 - Label2_9
 - Label7
 - Radio1
 - ButtonCanvas1
 - Label2
 - TextInput1
 - Label1
 - Table1
 - NoIndex_Choice
 - NoIndex_Date
 - NoIndex_DateTime
 - NoIndex_User
 - NoIndex_Number
 - NoIndex_Boolean
 - PP城テスト用画面
 - コントロール避難所

画面: NoIndex_Text

プロパティ 詳細設定

塗りつぶし

背景の画像: なし

画像の位置: 自動調整

テキスト型列 (インデックスなし)

Filter: a All = <> StartsWith EndsWith in IsBlank IsBlank

LookUp ID: タイトル:

ID	タイトル	選択肢	日付	日付時刻	ユーザー	数値	はい/いいえ
1	b16	え	2024/08/06	2024/08/07 5:21	yukio	35	はい
2	2e3	お	2024/08/06	2024/08/07 5:22	yukio	23	はい
3	d22	え		2024/08/07 5:23	テスト 2	97	はい
4	043	い	2024/08/06	2024/08/07 5:24		15	いいえ
5	45d	い	2024/08/06		yukio	14	はい
6	e5d		2024/08/06	2024/08/07 5:26	yukio	93	いいえ
7		う	2024/08/06	2024/08/07 5:27	テスト 1	100	いいえ
8	90e	お	2024/08/06	2024/08/07 5:28	yukio	48	はい
9	657	う	2024/08/06	2024/08/07 5:29	yukio		はい
10	e23	あ	2024/08/06	2024/08/07 5:30	yukio	71	はい
11	4b2		2024/08/06	2024/08/07 5:31	テスト 1	78	はい

行: 2000

委任を検証するときは

- ・ 先述の「データ行の制限」を極端に減らすことがあります。



検証(1)

Filter/LookUp関数

※全てを網羅できているとは限りません

テキスト型列Filter/LookUp 委任可否

```
Filter(
  List_NoIndex,
  [REDACTED]
),
```

ここに何を入れるか

```
LookUp(
  List_NoIndex,
  [REDACTED],
  ID
)
```

記述	Filter 委任	LookUp 委任	備考
タイトル = ●●	○	○	
タイトル <> ●●	×	×	
StartsWith(タイトル, ●●)	○	○	
EndsWith (タイトル, ●●)	×	×	
●● in タイトル ●● exactin タイトル	×	×	in : 大文字小文字を無視 exactin : 大文字小文字が一致する文字列のみを識別
IsBlank(タイトル)	×	×	「タイトル = Blank()」なら委任○
!IsBlank(タイトル)	×	×	「タイトル <> Blank()」でも委任×

選択肢型列Filter/LookUp

委任可否

```
Filter(  
    List_NoIndex,  
    [REDACTED]  
)
```

ここに何を入れるか

```
LookUp(  
    List_NoIndex,  
    [REDACTED],  
    ID  
)
```

記述	Filter 委任	LookUp 委任	備考
選択肢.Value = “●●”	○	○	
選択肢.Value <> “●●”	×	×	
StartsWith(選択肢.Value, “●●”)	×	×	
EndsWith(選択肢.Value, “●●”)	×	×	
“●●” in 選択肢.Value “●●” exactin 選択肢.Value	×	×	in : 大文字小文字を無視 exactin : 大文字小文字が一致する文字列のみを識別
IsBlank(選択肢.Value)	×	×	「 選択肢.Value = Blank() 」なら委任○
!IsBlank(選択肢.Value)	×	×	「 選択肢.Value <> Blank() 」でも委任×

日付型列Filter/LookUp

委任可否

```
Filter(  
  List_NoIndex,  
  [REDACTED]  
)
```

ここに何を入れるか

```
LookUp(  
  List_NoIndex,  
  [REDACTED],  
  ID  
)
```

記述	Filter 委任	LookUp 委任	備考
日付 = ●●	○	○	
日付 <> ●●	○	○	
日付 < ●●	○	○	不等号は逆向き (>) や 「=」 付きでもOK
IsBlank (日付)	×	×	「日付 = Blank ()」 なら委任○
!IsBlank (日付)	×	×	「日付 <> Blank ()」 なら委任○

日付 + 時刻型列Filter/LookUp 委任可否

```
Filter(
  List_NoIndex,
  [REDACTED]
),
```

ここに何を入れるか

```
LookUp(
  List_NoIndex,
  [REDACTED],
  ID
)
```

記述	Filter 委任	LookUp 委任	備考
日付時刻 = ●●	○	○	
日付時刻 <> ●●	○	○	
日付時刻 < ●●	○	○	不等号は逆向き (>) や 「=」 付きでもOK
IsBlank (日付時刻)	×	×	「日付時刻 = Blank ()」 なら委任○
!IsBlank (日付時刻)	×	×	「日付時刻 <> Blank ()」 なら委任○

ユーザー型列Filter/LookUp 委任可否

```
Filter(
  List_NoIndex,
  [REDACTED]
),
```

ここに何を入れるか

```
LookUp(
  List_NoIndex,
  [REDACTED],
  ID
)
```

記述	Filter 委任	LookUp 委任	備考
ユーザー.Email = ●●	○	○	
ユーザー.Email <> ●●	×	×	
StartsWith(ユーザー.Email, ●●)	○	○	
EndsWith(ユーザー.Email, ●●)	×	×	
●● in ユーザー.Email ●● exactin ユーザー.Email	×	×	in : 大文字小文字を無視 exactin : 大文字小文字が一致する文字列のみを識別
IsBlank(ユーザー)	×	×	ユーザー.Email=Blank() 等なら委任○
!IsBlank(ユーザー)	×	×	ユーザー.Email<>Blank() 等でも委任×

※ 「.Email」は「.DisplayName」等でも可

数値型列Filter/LookUp

委任可否

```
Filter(  
    List_NoIndex,  
    ██████████  
)
```

ここに何を入れるか

```
LookUp(  
    List_NoIndex,  
    ██████████,  
    ID  
)
```

記述	Filter 委任	LookUp 委任	備考
数値 = ●●	○	○	※データ行の制限以上表示できる？
数値 <> ●●	○	○	※データ行の制限以上表示できる？
数値 < ●●	○	○	※データ行の制限以上表示できる？
StartsWith(数値, ●●)	×	×	
EndsWith(数値, ●●)	×	×	
IsBlank(数値)	×	×	数値 = Blank() なら線は消えるけど委任×
!IsBlank(数値)	×	×	数値 <> Blank() なら線は消えるけど委任×

はい/いいえ型列Filter/LookUp 委任可否

```
Filter(  
  List_NoIndex,  
  ██████████  
)
```

ここに何を入れるか

```
LookUp(  
  List_NoIndex,  
  ██████████,  
  ID  
)
```

記述	Filter 委任	LookUp 委任	備考
はい/いいえ = ●●	○	○	
はい/いいえ <> ●●	×	×	

検証中おかしな動きを見せた

- 件数が多いデータソースで、データ行の制限も多い(2000件等)時、委任できないFilterと委任できるFilterを交互に切り替えたときに正しい結果が取れない(ことがある)

※インデックスがあるリストでも改善されなかった

StartsWith (委任○) で「2」から始まるものが抽出されている
→EndsWith (委任×) に切り替えても表示されるレコードが変わらないことがある
※起きる時と起きない時がある
※何度か切り替えると正しく表示される
※委任×→委任○にした時も起きる時があった

テキスト型列 (インデックスなし)

Filter All = <> StartsWith EndsWith in IsBlank !IsBlank

LookUp ID : 2 タイトル : 2e3

ID	タイトル	選択肢	日付	日付時刻	ユーザー	数値	はい/いいえ
2	2e3	お	2024/08/06	2024/08/07 5:22	yukio	23	<input checked="" type="checkbox"/> はい
14	2a4	い	2024/08/06	2024/08/07 5:34	テスト 2	22	<input type="checkbox"/> いいえ
16	226	お	2024/08/06	2024/08/07 5:36	テスト 2	56	<input type="checkbox"/> いいえ
27	22a	お	2024/08/06	2024/08/07 5:47	yukio	61	<input checked="" type="checkbox"/> はい
31	2d5	あ	2024/08/06	2024/08/07 5:51	テスト 1	63	<input type="checkbox"/> いいえ
37	2d6	え	2024/08/06	2024/08/07 5:57	yukio	58	<input type="checkbox"/> いいえ
46	25c	あ	2024/08/06	2024/08/07 6:06	yukio	34	<input type="checkbox"/> いいえ
65	2ee	お	2024/08/06	2024/08/07 6:25	テスト 2	52	<input type="checkbox"/> いいえ
111	2c0	あ	2024/08/06	2024/08/07 7:11	yukio	0	<input checked="" type="checkbox"/> はい
128	2bd	お	2024/08/06	2024/08/07 7:27	テスト 1	89	<input type="checkbox"/> いいえ
160	29f	お	2024/08/06	2024/08/07 8:00	テスト 1	49	<input type="checkbox"/> いいえ

行: 1052

検証(2)

テーブル (レコードセット) 取得系関数

※全てを網羅できているとは限りません

テーブル(レコードセット)取得系関数 委任可否

関数	関数の説明	委任	備考
Search	検索 (Filter + in とほぼ同じ)	×	
Sort	テーブルを並べ替え (単一条件)	△	5000件以上のリストでエラー →インデックス付きリストならOK
SortByColumns	テーブルを並べ替え (複数条件も可)	△	5000件以上のリストでエラー →インデックス付きリストで、かつ、条件が1つだけならOK
FirstN	先頭からN個のレコードセット	×	
LastN	末尾からN個のレコードセット	×	
Distinct	重複を省いた単一列テーブル	×	
Choices	選択肢列の一覧を返す	○?	公式にも有志の方の記事にも「委任できない」と書いているがちゃんと動く

単一選択の場合ちゃんと動くが、
複数選択可能な列だとどうなるか？要検証

Sort、SortByColumnsで出たエラー

【前提】 リストの行数が5000件を超えている

※Sort関数 インデックス付きリストだとエラー出ない

ネットワークからのデータ取得でエラーが発生しました: この操作は、リストビューのしきい値を超えているため、実行できません。
clientRequestId: bfa0cf62-7c7e-4f52-b083-a95000fc733f serviceRequestId: 340a49a1-601d-3000-a780-62052846c8ce

問題を表示 (Alt+F8) 使用できるクイック修正はありません

```
Sort(List_NoIndex, 日付時刻, SortOrder.Ascending),
```

※SortByColumns関数

ネットワークからのデータ取得でエラーが発生しました: この操作は、リストビューのしきい値を超えているため、実行できません。
clientRequestId: a56ec916-ab1a-4920-b02e-7b460572418b serviceRequestId: 3c0a49a1-00d3-3000-a780-6c46fb42f2f3

問題を表示 (Alt+F8) 使用できるクイック修正はありません

```
SortByColumns(  
    List_NoIndex,  
    "Number", SortOrder.Ascending  
    , "DateTime", SortOrder.Descending  
)
```

※SortByColumns関数 インデックス付きリストで単一条件だとエラーでない

```
SortByColumns(  
    List_Index,  
    "Number", SortOrder.Ascending  
    //, "DateTime", SortOrder.Descending  
)
```

リスト ビューのしきい値

※5000件以上あるリストの設定画面

The screenshot shows the settings page for a list named '委任検証用リスト2'. The left sidebar contains navigation links: ホーム, スレッド, ドキュメント, ノートブック, ページ, and 最近使った項目. The main content area is titled '委任検証用リスト2 > 設定' with a 'リンクの編集' icon. Under 'リスト情報', the '名前' is '委任検証用リスト2' and the 'Web アドレス' is 'https://urbandev365.sharepoint.com/sites/PowerPlatform/Lists/IninList2/AllItems.aspx'. The '説明' field is empty. The 'リストビューのしきい値' is highlighted in a teal box with the text '15432 アイテム (リストビューのしきい値は 5000)'. Below this, a warning message states: 'このリスト内のアイテム数がリストビューのしきい値 (5000 アイテム) を超えています。サーバーに過度な負荷がかかる原因となるタスク (リストの全アイテムを対象としたタスクなど) は、現在禁止されています。大きなリストやライブラリを使用して、アイテムを即座に表示する方法を説明します。' At the bottom, there are links for '20230505Appsから作成されたリスト', '全般設定', '権限と管理', and '通信'.

※件数が少ないリストの設定画面

The screenshot shows the settings page for a list named '委任検証用リスト3 (日付 & 日付時刻検証用)'. The left sidebar contains navigation links: ホーム, スレッド, ドキュメント, ノートブック, ページ, and 最近使った項目. The main content area is titled '委任検証用リスト3 (日付 & 日付時刻検証用) > 設定' with a 'リンクの編集' icon. Under 'リスト情報', the '名前' is '委任検証用リスト3 (日付 & 日付時刻検証用)' and the 'Web アドレス' is 'https://urbandev365.sharepoint.com/sites/PowerPlatform/Lists/IninList/AllItems.aspx'. The '説明' field is empty.

これがいわゆる「5000件問題」？

これが5000件問題かどうかは知らんけど

- ・ 5000件問題、インデックスについてはこちらを参照！

よく聞くけど、SharePointリストの5,000件問題ってなんなの？（太田浩史）

<https://www.docswell.com/s/hrfmjp/5QNNN5-2022-05-09-124548#p14>

PowerApps | SharePointリスト5000件問題を解決する2つの方法（七草あんこ）

<https://anko7793.com/2022/06/1697/>

検証(3) レコード取得系関数

※全てを網羅できているとは限りません

レコード取得系関数 委任可否

関数	関数の説明	委任	備考
First	最初のレコードを取得	○	
Last	最後のレコードを取得	×	波線は出ないが、データ行の制限に依存する（2000件にした時は2000件目がLastとして取得される）
Index	N番目のレコードを取得	×	波線は出ないが、データ行の制限に依存する（2000件にした時に2001番目以降のレコードを指定しようとするエラーが出る）

※Index関数で出たエラー（2001件目を取ろうとしたら）

```
Index(List NoIndex, Value(TextInput1 3.Text)).ID
```

'Index' 関数に対する 2 番目の引数は、テーブルの上限である 1 と下限である 2000 の間に設定する必要があります。非委任に制限されている場合、上限が低くなる可能性があります。

問題を表示 (Alt+F8) 使用できるクイック修正はありません

検証(4) 値取得系関数

※全てを網羅できているとは限りません

値取得系関数 委任可否

関数	関数の説明	委任	備考
CountRows	行数をカウント	×	
Sum	合計	×	
Average	平均	×	
Max	最大値	×	
Min	最小値	×	

※いずれもデータ行の制限内（2000件なら先頭2000件）の中で計算した数値を出してくる

件数の多いSharePointリストを扱う時
私ならこうする

コレクションを活用する

- ・ コレクションならPower Appsキャンバスアプリ内部で完結するので、委任を気にすることなく様々な操作を行える（Search等も使い放題）し、先述したおかしい動きも（たぶん）起きない。

【ポイント】

- ・ 委任できる関数を使って、コレクションに入れるデータの件数を出来る限り少なくすること

【注意】

- ・ コレクション内のデータを更新してもSharePointリスト上のデータは更新されない
- ・ 委任できる関数でFilterした結果すらも2000件を超えるデータソースだと意図する動きをしない

- ・ 前職での事例を紹介

※ イベント参加者限定（資料のみ閲覧の方ゴメンナサイ…）

まとめ

SharePointリスト、意外と優秀（失礼）だが…

- ・ 今回の検証を通して、**自分で手を動かしたことで**分かることが多くあった
- ・ 今までは「委任できる関数を使って、2000件以内にしてPower Appsに持っていくことができれば何をしてもOK」と思ってたが、**件数が膨大になってくると不都合が出る**（インデックス列を追加することで軽減は可能だが、根本的な解決策ではない）ことがわかった
- ・ それを気にしながら作り続けるよりかは、有償プランつけてDataverse（委任大得意）が使えればベスト
- ・ Dataverse for Teamsも検討に値する

SharePointリスト以外の委任を調べたかったら？

- ・ りなたむさんのこの資料を確認！

※素晴らしい資料ですが、あくまで参考情報として、実際の動きは自分で確かめてください。

<https://www.docswell.com/s/rinatamu/KXQGPZ-PowerApps-delegate>

ご清聴ありがとうございました