

# CELf アプリ作成規約

目的が CELf を用いたアプリ作成に限り、本資料をコピーして編集・再利用可能。

本目的以外は修正・再利用不可とする。

本資料の内容は予告なく変更する場合があります。

## ★ 改訂履歴

版	作成/改定日	変更内容
1.0	2021/11/5	新規作成
1.1	2021/12/16	1 シート構成

### 1.1 シートの構成

#### ① 画面シート

アクションから出力される値は、変数シートではなく  
画面シートに保持する記載を追記。

上記の対応にてアクションセットのキャプチャを修正。

## 目次

本資料の目的 .....	3
1 シート構成 .....	4
1.1 シートの構成 .....	4
1.2 セルの参照方法.....	8
2 アクション構成 .....	12
2.1 アクションセットの構成 .....	12
2.2 条件分岐.....	14
2.3 繰り返し.....	14
2.4 フィルタ設定 .....	15
2.5 コメントの記載ルール.....	16
3 関数と数式 .....	17
3.1 括弧の省略について .....	17
3.2 値の括り文字について.....	17
3.3 論理値判定の記述統一.....	18
3.4 大文字小文字の統一 .....	18
3.5 数値演算関数の使用 .....	18
3.6 数式の実行タイミング.....	18
4 データ登録・更新・削除・検索.....	21
4.1 データの登録ルール .....	21
4.2 データの更新ルール .....	21
4.3 データの削除ルール .....	22
4.4 排他制御の基本方針 .....	23
4.5 ロールバック（一括実行時のトランザクション） .....	25
4.6 CELF 標準アクション使用と SQL 拡張アクション使用の切り分け.....	27
4.7 SQL 拡張アクションを使用する場合のルール .....	28
4.8 対応する SQL 構文、SQL 関数 .....	29

## 本資料の目的

本資料は、シート、アクションセットを作成する上での作成方法を定めた規約を説明する資料である。

規約を設けることで作成者によるシート、アクションセットの作成方法を統一し、保守性、品質を向上させることが目的である。

統一した作成方法でシート、アクションを作成したいが、どのような作成方法が適しているか、判断できないアプリ作成者は、アプリ作成前や作成中に本資料を参照する。

## 1 シート構成

シート構成のルールを以下に説明する。

### 1.1 シートの構成

シートは画面、変数、メッセージの役割で3つに分割した構成とする。

- シートの説明

画面シート：アプリ使用時に表示されるシート。

変数シート：変数を管理するシート。

メッセージシート：

メッセージを管理するシート。

複数画面から参照される共通的に使用するシート。

- シートを分割する理由について、

画面シートに変数やメッセージを保持させる場合、画面に表示しない情報であるため非表示の領域に保持する必要がある。

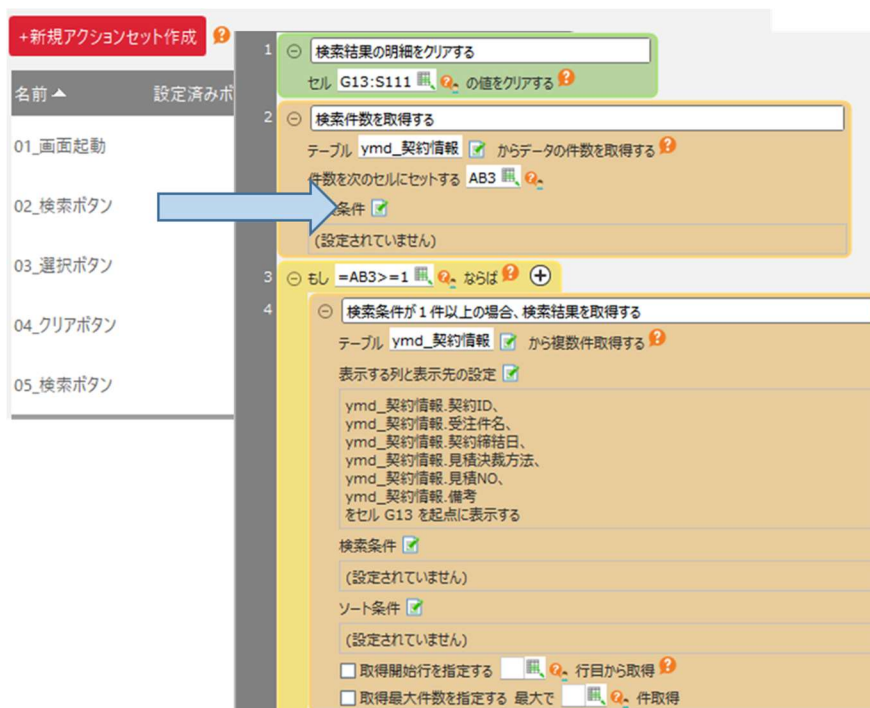
変数とメッセージを画面シートに保持させる場合、可視性が悪いことや画面レイアウトの行列追加により、非表示領域に修正が発生するリスクもあるため、可視性と保守性を考慮して変数シートとメッセージシートに設定する。

#### ① 画面シート

アプリを使用した時に画面表示するシート。

The screenshot displays a spreadsheet interface for a contract management application. The main content area is titled '契約一覧' (Contract List). Below the title, there is a '検索条件' (Search Conditions) section with a '取引先名' (Customer Name) field set to '取引先A' and a '課金期間' (Billing Period) field set to '202101'. There are buttons for '選択' (Select), 'クリア' (Clear), and '検索' (Search). Below the search conditions, there is a '検索結果' (Search Results) section with a table. The table has columns for '契約ID', '受注件名', '契約締結日', '見積決裁方法', '見積NO', and '備考'. There are five rows of data, each with a '修正' (Edit) button in the first column.

画面起動時やボタン押下時の処理をアクションセットに作成する。



画面に表示しない情報は変数シートに保持する。

ただし、アクションから出力される値については、変数シートではなく画面シートの非表示領域に設定する。

その際、変数の列は非表示列であることがわかるようにする。

例) 一覧の検索結果と同時に取得した各行に対する変数

	P	Q	R	S	T	AT	AU
1						非表示列	非表示列
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12	見積決裁方法	見積NO	備考	課金開始月		課金開始日※非表示	課金終了日※非表示
13	決裁方法①	Mxxxxx000001	備考 1	2021/10			
14	決裁方法①	Mxxxxx000002		2021/10			
15	決裁方法②	Mxxxxx000003		2021/10			
16							
17							

## ② 変数シート

変数のみ保持するシート。

変数とは画面には表示しない以下の情報である。

- 画面間で渡される情報
- 処理中に取得した情報

	A	B	C	D	E	F
1	画面遷移パラメータ		変数 アクションセット毎に設定			
2			01 画面起動時		02 検索ボタン	
3	遷移元画面ID		ループカウンタ		ループカウンタ	
4					検索件数	
5						

### <列の説明>

- 画面遷移パラメータ：
  - 呼出し元画面にて呼出し先画面に必要な情報を設定する領域。
- 変数：処理中の値を設定する領域。
  - アクションセット毎に変数を用意し、アクションセットを跨いでの変数は使用しない。

### <変数の説明>

画面毎に設定する項目は異なるが、必ず設定する項目について説明する。

- 画面遷移パラメータ
    - 遷移元画面 ID：呼出し元画面の画面 ID を設定する。  
どの画面から開かれたのかの判定に使用する。
  - 変数
    - ループカウンタ：表の開始行数をループカウンタに設定して、繰り返しごとにカウントアップする。  
一覧の値を参照する際に使用する。
    - 検索件数：検索結果の件数を設定する。  
検索件数が一覧の行数を超過した場合のチェックで使用する。
- ※変数にて日付形式、数値の形式をあつかう場合、必ずセルの書式もあわせて設定すること。

③ メッセージシート

各画面で表示するメッセージを管理するシート。

	A	B	C
1		メッセージシート	
2			
3		メッセージID	メッセージ内容
4		Q001	[%1]を行います。よろしいですか？
5		I001	[%1]が完了しました。
6		I002	検索結果は0件です。
7		E001	[%1]は必須入力です。
8		E002	[%1]は[%2]以内で入力してください。
9		E003	[%1]は[%2]形式で入力してください。
10		E004	他のユーザに更新されています。画面を開きなおしてください。
11			

詳細については以下を参照する。

『O2\_SELF\_アプリ作成標準』 3.3 メッセージの管理方法

## 1.2 セルの参照方法

セルの参照方法はいくつかの参照方法があるため、それぞれの参照方法に適したケースを説明する。

### ① 名前参照

画面シートの場合、セルの参照方法は「名前参照」を使用する。

#### ・参照方法の説明

「名前参照」とはセルに名前をつけ、セル番号ではなくセル名を指定することでセル参照を行う方法である。

#### <名前管理について>

セルに名前をつける「名前管理」機能の使用方法については CELF ヘルプを参照する。

CELF ヘルプ：

[https://cloud.celf.jp/celf-help/ja/texts/sheet/define\\_name/define\\_name.html](https://cloud.celf.jp/celf-help/ja/texts/sheet/define_name/define_name.html)

名前を管理する対象は以下のとおり。

画面シートに配置した入力項目を全て管理対象に設定する。

また、処理中に値が変わることない画面の設定情報として画面 ID、一覧開始行、一覧の終了行、一覧のフィルタ範囲も管理対象とする。

#### ・利用する場面

画面シートに配置した入力項目を参照する際に使用する。

#### ・使用する際のメリット

行、列の追加や削除による画面レイアウトの変更を行った際に参照範囲は変更されないため、影響はない。

また、セル番号や数式を使用しないため、可視性が向上する。

名前の管理を【C2 セル】に設定する





B 列に列を挿入すると参照範囲が自動で【D2 セル】に変更される



- 使用する際のデメリット

セルの位置を変更した場合、参照範囲は変更されないため変更先のセルを参照するよう修正する必要がある。

【D 2 セル】 名前に設定する



入力するセルの位置を切り取りで【C3 セル】に変更した場合、参照範囲は【D2 セル】のまま変更されない。



## ② 直接参照

変数シートのセルの参照方法は「直接参照」を使用する。

### ・参照方法の説明

セルの番号を直接指定する方法である。

画面シートのアクションから「名前参照」にてセル参照を行う場合、関数で参照する必要がある。

関数例：「=ADDRESS(ROW(シート名!セル名), COLUMN(シート名!変数名), 4)」

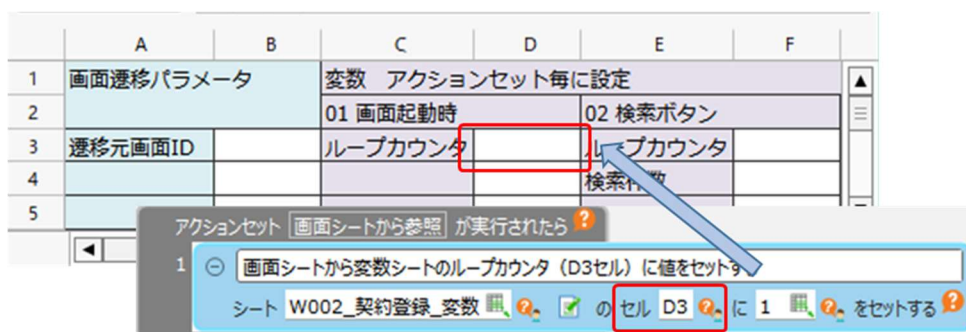
### ・利用する場面

変数シートのセルを参照する際に使用する。

### ・使用する際のメリット

セル番号を指定するのみであるため、一番簡単にセルの参照が行える。

変数シートはレイアウトの変更が極めて少なく、「名前管理」を使用するメリットがない。



### ③ 間接参照

一覧に対しての繰り返し処理で処理中に参照するセルを変更したい場合は

①の名前参照は使用できないため、「間接参照」を使用する。

#### ・参照方法の説明

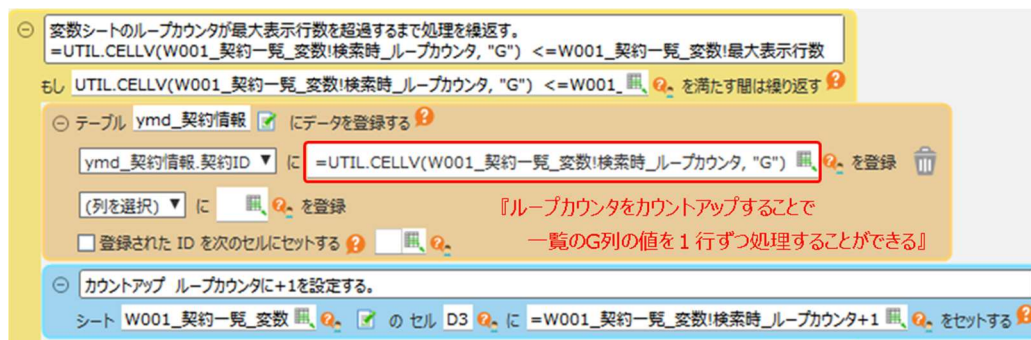
「間接参照」とは CELLV 関数を使用した参照方法である。

関数例:「=UTIL.CELLV(行番号, 列番号または列記号)」

#### ・使用する際のメリット

関数に設定する参照先を変数とすることができるため、参照セルを固定せずに動的にセルの参照が行える。

以下は「間接参照」を使用して、一覧に表示されている全行を繰り返し処理にてデータベース（以下 DB）へ登録を行う例である。



※「W001\_契約一覧!」の記載は変数シートを参照している。

## 2 アクション構成

アクションセットの構成とアクションの使用方法について説明する。

### 2.1 アクションセットの構成

以下2つの構成とする。

イベントに紐づくアクションセット：

画面の初期起動時、セルの値変更時、ボタンクリック時のイベントから呼び出されるアクションセット。

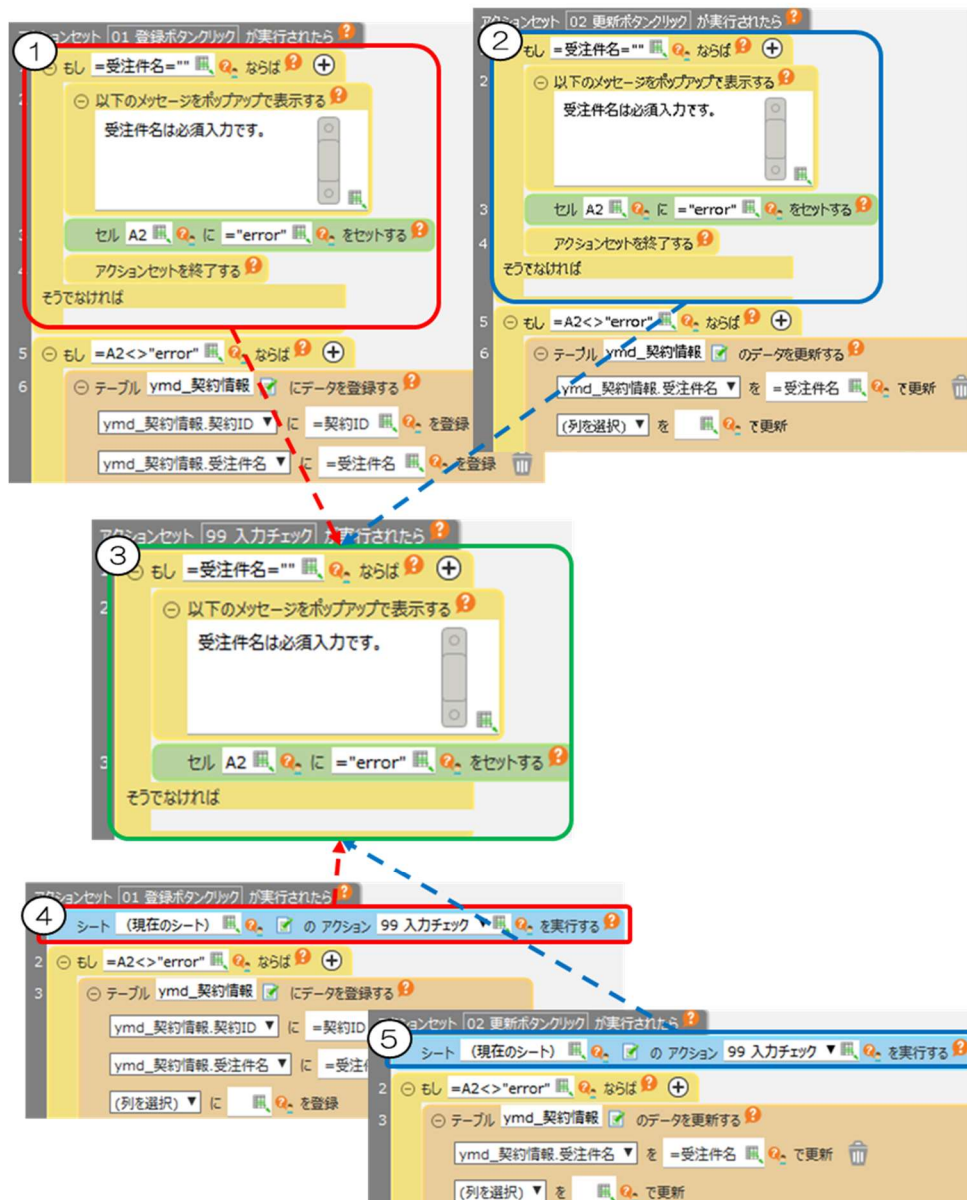
共通的に使用するアクションセット：

類似処理を行うアクションセットが複数ある場合、それぞれのアクションセットに作成すると非効率である。

類似処理は共通的に使用する1つのアクションセットとして作成する。

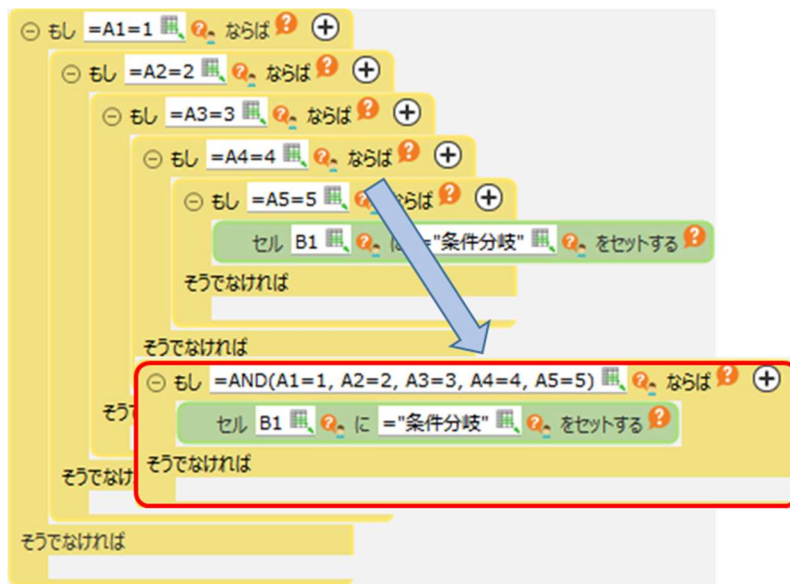
以下、同じ処理が設定されている2つのアクションセットとそれを共通アクションセットとして作成した場合の例を説明する。

- ①②：登録ボタン、更新ボタンをクリックした際に同じ入力チェックを各アクションセットで実施している。
- ③：①②の入力チェックを共通で使用するアクションセットに移動する。
- ④⑤：それぞれから③のアクションセットを呼び出す。



## 2.2 条件分岐

条件分岐は3～4階層までを目安とし、それ以上の階層が発生する場合は可視性が悪くなるため AND 関数や OR 関数を使用した分岐条件の統合を行う。



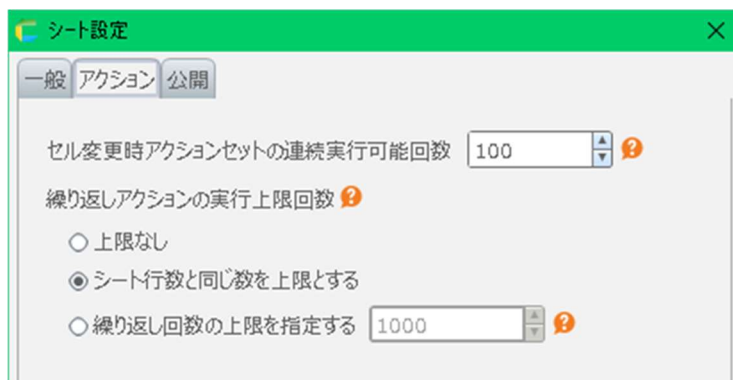
## 2.3 繰り返し

繰り返しの限界値の初期値設定について説明する。

シート設定にて繰り返しの限界値の設定ができる。

初期値には「シート行数と同じ数を上限とする」が設定されているため、シート行数以上の繰り返し処理が実行されると強制的に繰り返しアクションが終了する仕組みとなっている。

シート行数以上の繰り返し処理を行いたい場合は、「繰り返しの回数の上限を指定する」に変更すること。

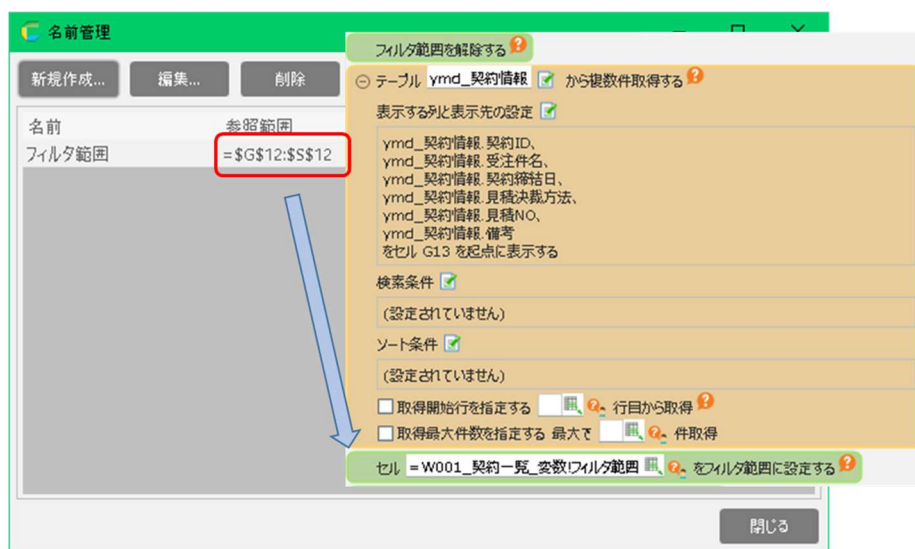


## 2.4 フィルタ設定

フィルタの使用ルールについて以下に説明する。

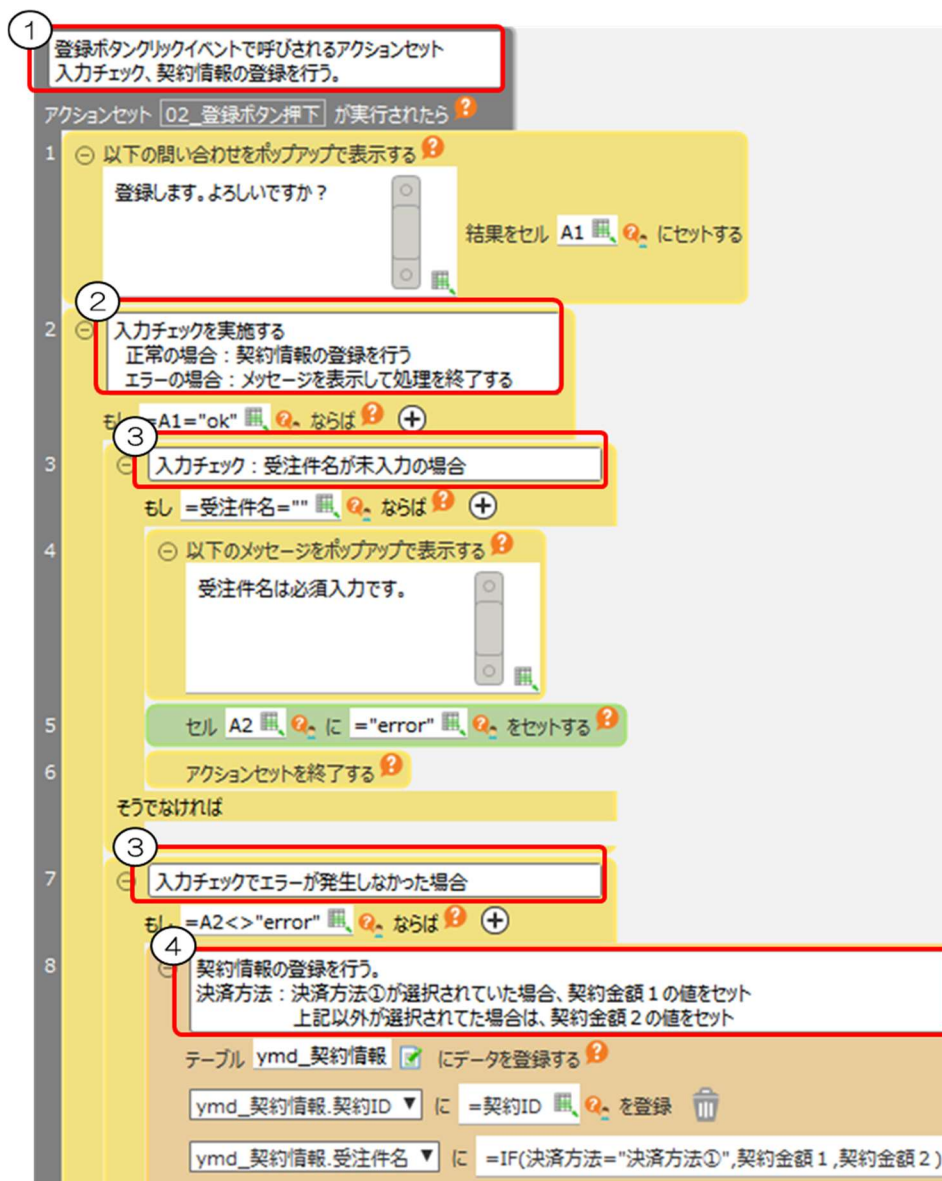
- フィルタの再設定・解除について

フィルタがかかっている状態で一覧の再取得を行った場合、フィルタは解除されずにデータが表示されてしまう。データ取得アクションの前後でフィルタの解除、再設定を行うこと。



## 2.5 コメントの記載ルール

コメントの記載ルールを説明する。



※上記はコメント用のサンプルのため、コメント以外は本規約や標準に準じて作成していない。参考にしないこと。

### • コメントの説明

- ① アクションセットの概要を記載する。
- ② 処理の塊ごとに処理概要を記載する。
- ③ 条件分岐や繰り返し処理の条件について記載する。
- ④ 複雑な数式を使用した一目で判断できないような箇所については、補足として処理の詳細を記載する。



### 3 関数と数式

関数と数式の使用ルールについて説明する。

#### 3.1 括弧の省略について

SELF では、引数無しの関数は括弧の省略記述が可能となっている。

シート上に直接記述している場合は Excel 出力で構文不正となる為、原則として括弧付きで記述する。アクションへ直接記述する場合も、影響はないが統一する。

対象の関数：NOW, ROW, TODAY

#### 3.2 値の括り文字について

等号(=)の条件式では 括り文字の有無に関わらず暗黙的に同一の型で比較が行われるが、不等号(<>)の条件式では 型が異なる場合は「一致しない」と厳密に判定される為、字型の値は数字であっても“(ダブルクォーテーション)で括り文字型であることを明確にする。

同様に、数値、および論理値(true/false)には 括り文字は使用しない。

型が異なる場合の比較例 (A1 セルの値は 1)

A1 の表示形式	数式	判定結果
標準 又は 数値	=A1="1"	true
	=A1=1 ※	true
	=A1<>"1"	true
	=A1<>1	false
文字列	=A1="1"	true
	=A1=1	true
	=A1<>"1"	false
	=A1<>1	true

※  
「= 1 = 01」を比較する場合、  
変定結果は true となってしまうため、  
先頭に 0 が入る場合は  
EXACT 関数を使用すること。

- ・アクションでボタンを設定したい場合

シート上でボタンを配置すると以下の数式が設定される。

=BUTTON("検索",1)

処理の中にボタンを動的に表示させたい場合、アクションからボタンを配置する場合は以下を設定する。

=BUTTON("検索",1)

数式自体を"で括る必要があり、「検索」を括っている"を文字列として認識させるため、"を二重にする必要がある。

### 3.3 論理値判定の記述統一

同一の動作でも記述方法がバラけるケースを避ける為、ルールを説明する。

機能	統一ルール	類似の記述
論理値判定 (A1 セルが true)	=A1	=A1=true =IF(A1,true,false)
論理値判定 (A1 セルが false)	=NOT(A1)	=A1=false =IF(A1,false,true)
論理値判定 (A1 セルが B1 セルと一致)	=A1 = B1	=IF(A1=B1,true,false) =EXACT(A1,B1)
論理値判定 (A1 セルが B1 セルと不一致)	=NOT(A1 = B1)	=IF(A1<>B1,true,false) =NOT(EXACT(A1,B1)) =A1<>B1

### 3.4 大文字小文字の統一

大文字／小文字ついて可読性の向上を目的としてルールを説明する。

対象	大文字、小文字	例
関数名	大文字	ROW、TRIM
論理値	小文字	true、false
セルアドレス	大文字	A1、BC100

### 3.5 数値演算関数の使用

SELF で扱える少数は 整数部 9 桁、少数部は 6 桁迄で、それ以上の桁数を引数に使用した場合は 演算関数がエラーとなるか返却値の精度が落ちる。

あらかじめ桁数チェック（上限値、下限値のチェック）を行うか、IFERROR 関数のエラー制御を含めること。

例：「=TRUNC(99999999999.9999)」結果は数式エラーとなるため、  
「=IFERROR(TRUNC(99999999999.9999), 0)」 のように対策が必要となる。

### 3.6 数式の実行タイミング

数式は以下のタイミングで実行される。以下に注意して数式を設定すること。

- ・該当セルを初回表示したタイミング。
- ・計算が実行されたその他の式（アクション上の式も含む）から参照された

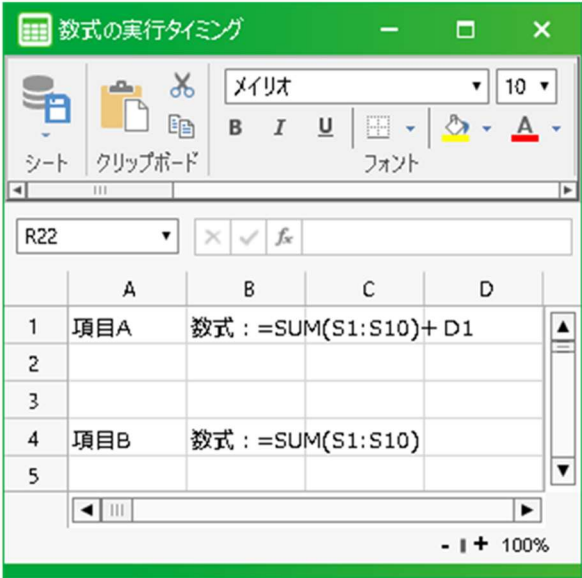
タイミング。

実行タイミングについて例を説明する。

S列を参照する数式が設定されている4つのセルがある。  
ただし、項目AのみS列と項目Dのセルを参照している。

	A	B	C	D
1	項目A	数式：=SUM(S1:S10)+ D1		
2				
3				
4	項目B	数式：=SUM(S1:S10)		
5				
6				
7				
8	項目C	数式：=SUM(S1:S10)		
9				
10				
11	項目D	数式：=SUM(S1:S10)		
12				

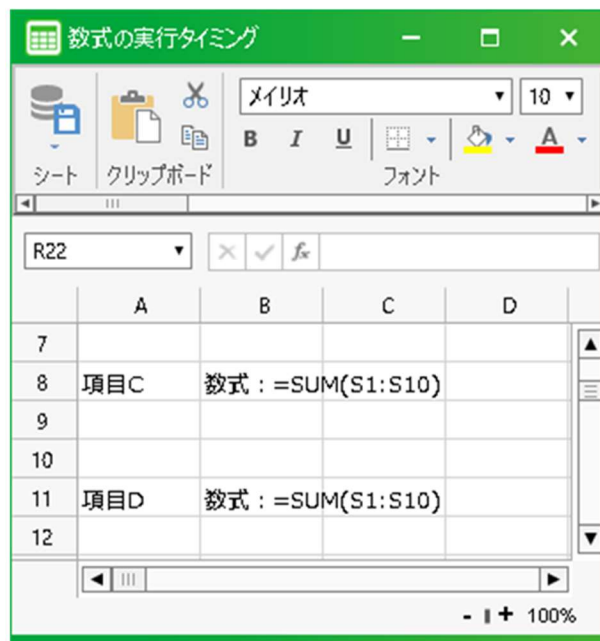
上記をシートを公開された画面を開く。  
画面起動時のサイズにより画面に表示されるのは項目Aと項目Bとなる。



このタイミングで  
数式の計算が実施されるのは、  
画面に表示されている  
**項目A、項目B**  
画面には表示されていないが  
項目Aで参照している  
**項目D** となる。

画面に表示されていない、  
計算されたセルから参照されていない  
**項目C** は計算を実施しない。

下にスクロールをして項目C、項目Dを画面に表示する。



このタイミングで  
数式の計算が実施されるのは、  
画面に表示されている  
まだ計算が実施されていない  
**項目C** となる。

既に計算が実施されている  
**項目D** は計算を実施しない。

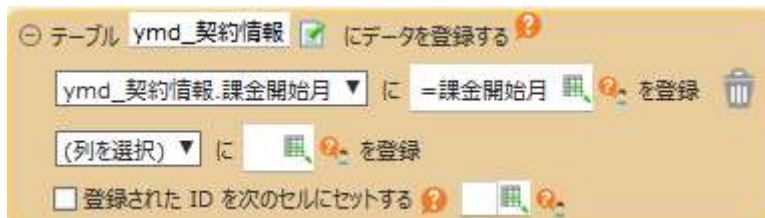
一度計算が実施されたセルは、画面に再表示しても計算は実施されない。

## 4 データ登録・更新・削除・検索

データ登録、更新、削除、検索時のルールについて説明する。

### 4.1 データの登録ルール

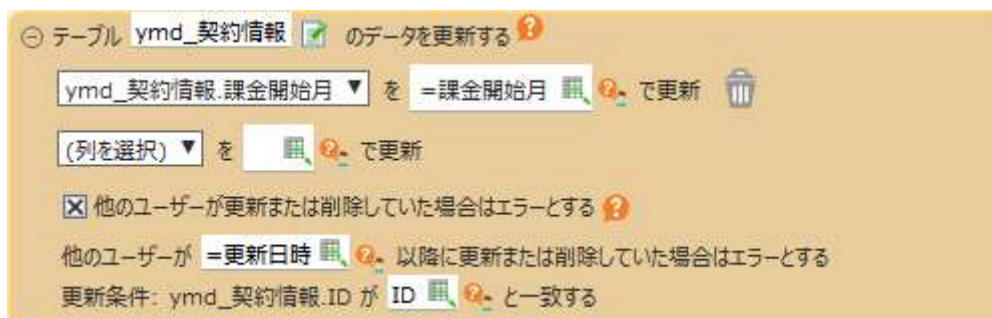
データの登録を行う際は、以下のアクションを使用する。



### 4.2 データの更新ルール

データの更新を行う際は、以下の2つの方法で実施する。

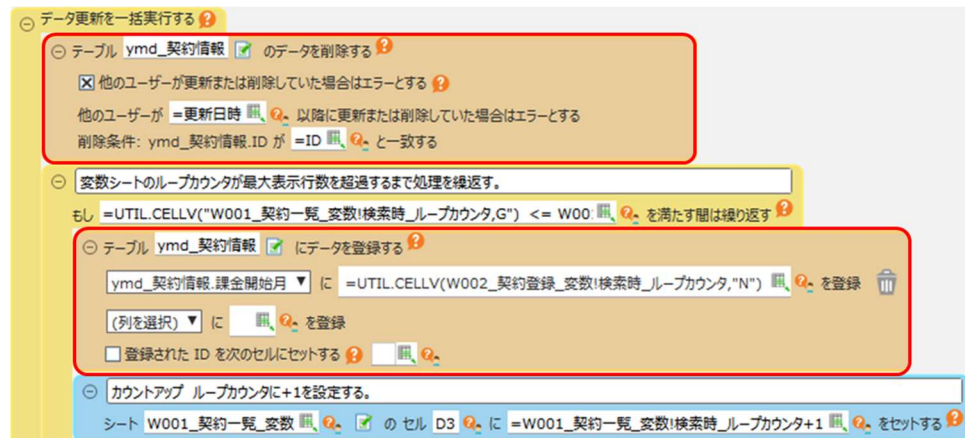
- ① 単一のデータを更新する場合、「データを更新する」アクションにて更新する。  
更新時は必ず「他のユーザーが更新または削除していた場合エラー」のチェックを ON にして排他チェックを実施する。（排他制御については 4.4 で説明）



- ② 1 件のデータを更新するのではなく、複数のデータをまとめて更新できる画面については「データを更新する」アクションは使用しない。

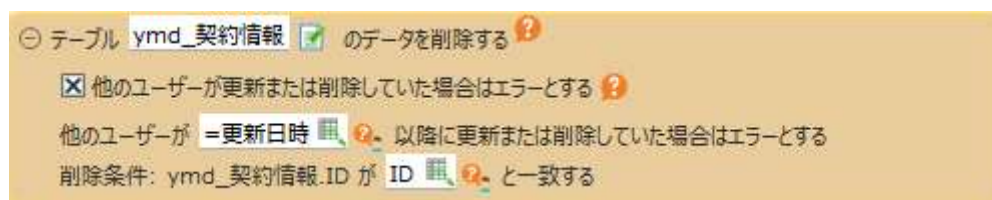
「データを削除する」アクションでデータを全て削除した後に「データを登録する」アクションにて画面上のデータをまとめて登録する。

処理速度が速くなるため、この方法を使用する。



#### 4.3 データの削除ルール

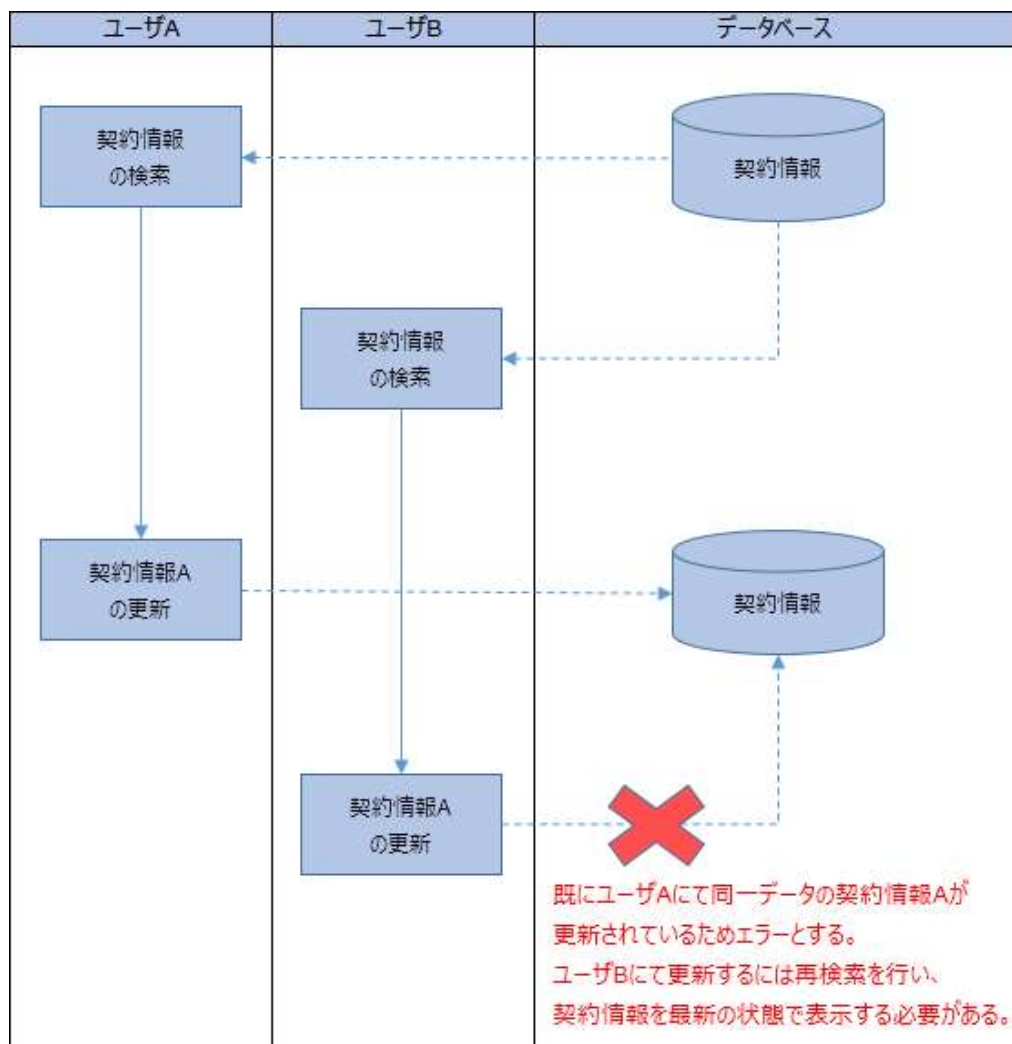
データの削除を行う際は、以下のアクションを使用する。



削除時は必ず「他のユーザーが更新または削除していた場合エラー」のチェックをONにして排他チェックを実施する。(排他制御については 4.4 で説明)

#### 4.4 排他制御の基本方針

2人のユーザーが同じタイミングでデータ更新、削除を行った場合、後に更新したユーザーの内容がDBに反映される。排他制御とは先に更新した内容を意図せず上書きしないようエラーとする制御である。

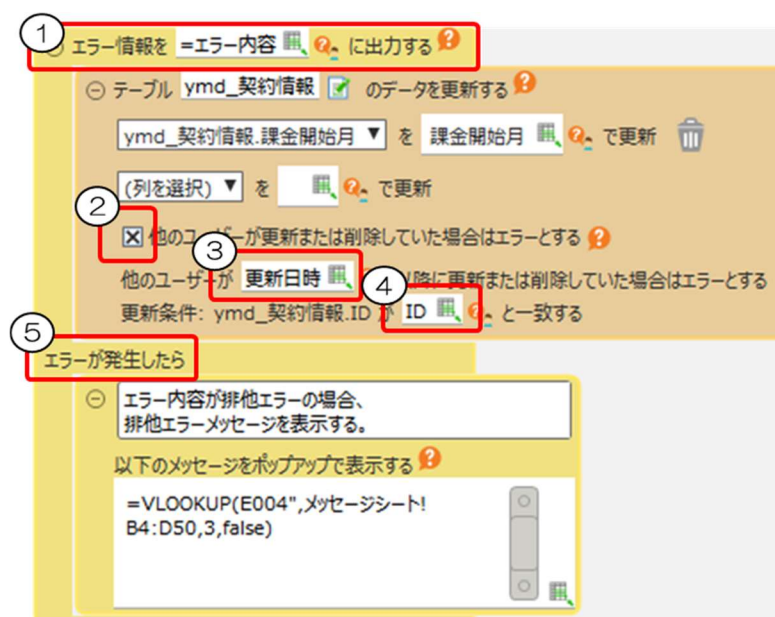


チェック内容は当該データの更新日時を検索時と更新時で比較し、異なる場合エラーの判定となる。

チェックは以下の設定にてデータ更新、削除アクションで行う。

「他のユーザーが更新または削除していた場合はエラーとする」チェックボックスをON、『エラー情報を xxx に出力する』アクションの組み合わせで作成すること。





### ＜アクションの説明＞

- ① データ更新アクションにて排他エラーが発生した場合、本アクションにてエラー情報を取得できる。エラー情報を出力するセルを指定する。
- ② チェック ON にすると排他チェックが実施される。
- ③ データ検索時の LAST\_MODIFIED（最終更新日時）を指定する。  
上記と本アクションが処理された時点の LAST\_MODIFIED を比較する。
- ④ 排他制御のチェックを ON にした場合、更新条件は「ID が一致する」ものとなる。データ検索時に ID の取得が必要となる。
- ⑤ エラーが発生した場合に排他エラーかどうかの判定を行い、エラーの場合は排他エラーメッセージを表示する。

※エラー情報を出力するアクションの詳細については CELf ヘルプを参照する。

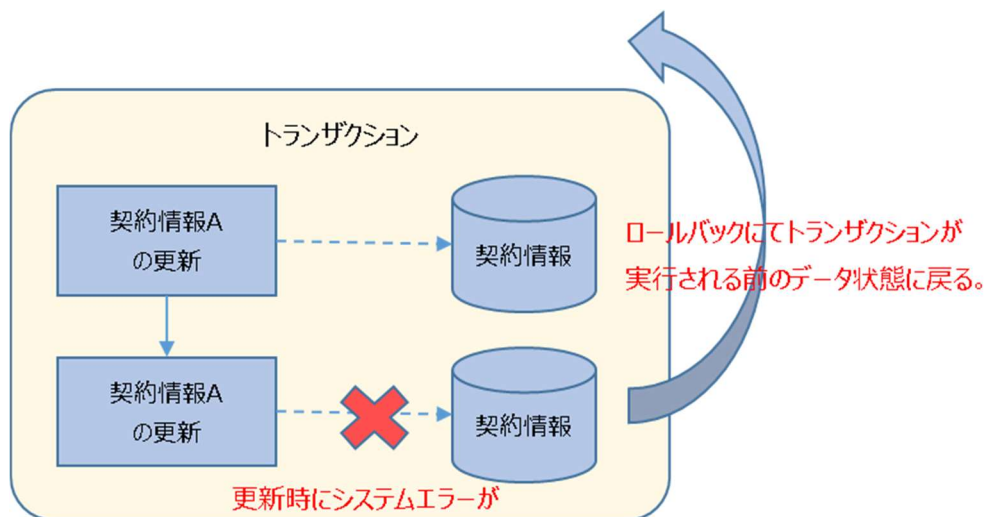
CELf ヘルプ： [https://cloud.celf.jp/celf-help/ja/texts/action\\_use/index.html](https://cloud.celf.jp/celf-help/ja/texts/action_use/index.html)





#### 4.5 ロールバック（一括実行時のトランザクション）

トランザクションとはDBに対する複数の変更（登録・更新・削除）を1つの変更として扱うことを指し、ロールバックとは、その大きな処理の中でエラーが発生した場合、処理内で行われた全ての処理を取り消すことを指す。



複数回DBを更新する場合、『データ更新を一括実行する』アクションを使用しトランザクションにまとめることで、更新の途中でエラーが発生した場合でもロールバックが実行されデータの整合性が確保される。

データ登録、更新、削除の際は本アクションを使用すること。



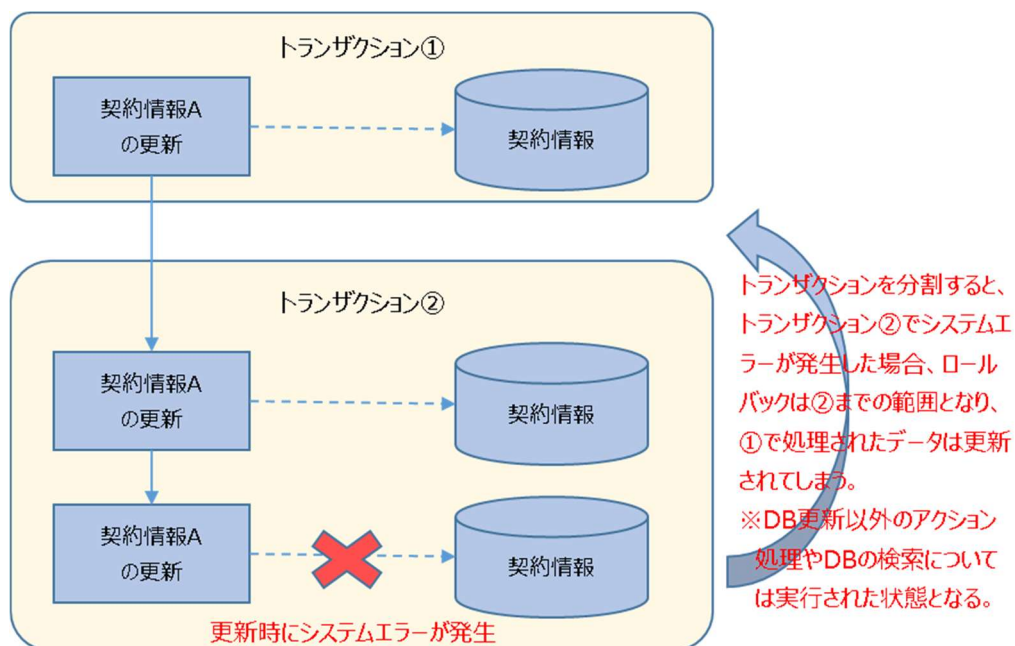
### ＜注意するポイント＞

- 複数トランザクションの使用について

システムエラーが発生した場合、イベント内で更新されたデータは全てロールバックされるべきであるため、下記の図のような1イベント内で複数トランザクションは使用しない。原則1イベントで1トランザクションとする。

※イベントとは

画面の初期起動時、セルの値変更時、ボタンクリック時を指す。



- 『データ更新を一括実行する』アクション内の処理順序について

アクションの実行順序は基本的に上から下の順序で処理されていくが、

『データ更新を一括実行する』アクション内にデータ登録、更新、削除以外のアクションをセットした場合、異なる順序で実行される。

以下の例では、②と④のデータ更新アクションの間に③のデータ取得アクションをセットしている。データを更新する②と④は一括で実行されるため、②③④の順序ではなく③②④の順序で実行される。

③で取得されるデータは①時点で更新した結果となり、②で更新した結果ではない。

よって、CELF では1 トランザクション内でデータ更新した結果を取得する処理は実施できない。

※データ取得アクション以外のセル操作アクション、すべてのアクシ

ョンに対して本制限が発生してしまう。

1 テーブル ymd\_契約情報 のデータを更新する

ymd\_契約情報.受注件名 を "test1" で更新

(列を選択) を で更新

☐ 他のユーザーが更新または削除していた場合はエラーとする

更新条件

ymd\_契約情報.ID が次と一致する: ="16"

データ更新を一括実行する

2 テーブル ymd\_契約情報 のデータを更新する

ymd\_契約情報.受注件名 を "test2" で更新

(列を選択) を で更新

☐ 他のユーザーが更新または削除していた場合はエラーとする

更新条件

ymd\_契約情報.ID が次と一致する: ="16"

3 上記で登録したIDを元に取得を行う。

テーブル ymd\_契約情報 から1件取得する

ymd\_契約情報.受注件名 をセル A1 に表示

(列を選択) をセル に表示

検索条件

ymd\_契約情報.ID が次と一致する: ="16"

4 テーブル ymd\_契約情報 のデータを更新する

ymd\_契約情報.受注件名 を "test3" で更新

(列を選択) を で更新

☐ 他のユーザーが更新または削除していた場合はエラーとする

更新条件

ymd\_契約情報.ID が次と一致する: ="16"

#### 4.6 CELF 標準アクション使用と SQL 拡張アクション使用の切り分け

SQL とは DB に蓄積されているデータの取得や定義、DB の操作を行うことができる言語である。

※下記の説明の中に SQL で使用する専門用語を記載しているが、本資料では説明を省略する。

CELF ではデータを取得するアクション以外に SQL にてデータを取得する、SQL 拡張アクションがある。



SQL アクションの使用方法は CELf ヘルプの「SQL アクション」を参照する。  
CELf ヘルプ：

[https://cloud.celf.jp/celf-help/ja/texts/extension\\_sql/about\\_sql.html](https://cloud.celf.jp/celf-help/ja/texts/extension_sql/about_sql.html)

以下のケース以外は、原則として標準のアクションを使用すること。

＜ 標準のデータ取得アクションで実現できないケース ＞

- ・条件が複雑でサブクエリ構成が必要な場合
- ・WHERE 句内に「( )」を記述し AND、OR をネストさせる必要がある場合
- ・件数取得を行う場合 ※COUNT 関数
- ・その他 SQL 関数や演算を使用したい場合  
※DB 側で取得データを加工する方が、効率的 又は高速な場合

#### 4.7 SQL 拡張アクションを使用する場合のルール

- ・別名（エイリアス）について  
集合関数等で別名を指定していない場合は、値が返却されずエラーにもならない場合があるため、別名は必ず指定すること。
- ・SQL 実行エラーの検知方法  
SQL 実行エラーが発生した場合、『エラー情報を xxx に出力する』アクションで検知することが可能であるため使用すること。
- ・SQL 実行結果の結合セルへの値貼り付け  
セルの結合を行っている場合、起点となっているセルにのみ値が設定されるが、SELECT 句のカラム数は画面の列数と一致させる必要がある。  
以下の通り記載すること。

＜SQL 例＞ select ID, 名称, “ ” as dummy, 更新者 from . . . .

B	C	D	E
一覧			
ID	名称		更新者
1	名称1		更新者1
2	名称2		更新者2

#### 4.8 対応する SQL 構文、SQL 関数

SELECT によるデータ取得のみ。INSERT/UPDATE/DELETE は標準のアクションを使用する。

構文の記述は SQL 標準に準拠すること。

SQL の予約語と重複する名前の関数が使えない (LEFT、RIGHT)

SQL の予約語

ALL	FULL	ON
AND	GROUP	ONLY
ANY	HAVING	OR
AS	IN	ORDER
ASC	INNER	OUTER
BETWEEN	INSERT	PARTIAL
BY	INTERSECT	READ
CASE	INTO	RIGHT
COLLATE	IS	SELECT
CORRESPONDINGCORR	JOIN	SET
CROSS	LEFT	TABLE
DELETE	LIKE	TEMPORARYTEMP
DESC	LIMIT	THEN
DISTINCT	MATCH	UNION
ELSE	MINUS	UNIQUE
END	NATURAL	UPDATE
EXCEPT	NOT	USING
EXISTS	NULL	VALUES
FOR	OFFSET	WHEN
FROM	OF	WHERE

LEFT, RIGHT の回避策

対象	代替策
LEFT( xxx, n )	SUBSTRING( 1, n )
RIGHT( xxx, n )	SUBSTRING( LENGTH( xxx ) - n + 1, n )

以上