

Simplifier les IF imbriqués

Description

Qu'est ce qu'une instruction IF imbriqués ?

Pourquoi est-il important de simplifier les instructions IF imbriqués ?

Il y a quelques problèmes avec les instructions IF imbriqués comme une difficulté à lire, maintenir ou mettre à jour ces instructions.

Simplifier les IF imbriqués avec coalesce

La fonction **coalesce** retourne la première valeur non-nulle d'une liste d'arguments. L'utilisation de cette fonction permet une meilleure lisibilité et une meilleure maintenance.

Voici un exemple de IF imbriqués :

```
if(
    value("TÃ©lÃ©phone portable") != null,
    value("TÃ©lÃ©phone portable"),
    if(
        value("TÃ©lÃ©phone portable professionnel") != null,
        value("TÃ©lÃ©phone portable professionnel"),
        value("TÃ©lÃ©phone personnel")
    )
)
```

Le code ci-dessus peut être remplacé par le suivant :

```
coalesce(value("TÃ©lÃ©phone portable"), value("TÃ©lÃ©phone portable professionnel"),
```

Un autre exemple de code :

```
if(
    value("Date de départ") != null,
    value("Date de départ"),
    date("1970/01/01")
)
```

qui peut être remplacé par :

```
coalesce(value("Date de départ"), date("1970/01/01"))
```

Fonctions Predicate

Les fonctions **predicate** sont utilisées pour évaluer les conditions et retournent des valeurs booléennes. Elles sont utiles au sein d'expressions conditionnelles ou dans des instructions IF. En résultat, peu de lignes de code et une lisibilité améliorée.

Voici un exemple de IF imbriqués :

```
if(
  value("Client fidel"),
  "Expédition gratuite",
  if(
    value("Total commande") >=50,
    "Expédition gratuite",
    "Expédition payante"
  )
)

if(
  value("Client fidel") or value("Total commande") >=50,
  "Expédition gratuite",
  "Expédition payante"
)
```

IF imbriqués versus instruction Case

Ci-dessous un exemple de IF imbriqués :

```
if(value("Age") < 1, "Nourrisson",
  if(value("Age") < 5, "Enfant en bas âge",
    if(value("Age") < 13, "Enfant",
      if(value("Age") < 20, "Adolescent",
        if(value("Age") < 30, "Jeune adulte",
          if(value("Age") < 65, "Adulte",
            "Personne âgée"
          )
        )
      )
    )
  )
)
```

Pouvant être remplacé par :

```
case(
  value("Age") < 1, "Nourrisson",
  value("Age") < 5, "Enfant en bas âge",
  value("Age") < 13, "Enfant",
  value("Age") < 20, "Adolescent",
  value("Age") < 30, "Jeune adulte",
  value("Age") < 65, "Adulte",
  "Personne âgée"
)
```

Range lookup

Un range lookup est utilisé pour faire correspondre une valeur à une plage de valeurs et retourne le résultat correspondant. Les mises à jour sont effectuées par le biais d'une table lookup externe ou un fichier, éliminant le besoin d'écriture du script.

Ci-dessous un exemple de IF imbriqués :

Voici un script **.build** (Spectre) utilisant un lookup et ouvert dans Workbench :



Et voici le contenu du deuxième objet **text** :

Previous tests		Test results			
age_range-LKP_2_at_14:26:57 people-FIN_at_14:26:38		(line)	Age Group	Start	End
		1	Infant	0	0
		2	Toddler	1	4
		3	Child	5	12
		4	Teen	13	19
		5	Young Adult	20	29
		6	Adult	30	64
		7	Older Adult	65	200

Voici une partie des propriétés de l'objet **lookup** :

age_range-LKP

Comments

Enter your comments for this object here

Lookup Definitions (1)

Base Column

Age

Main Flow (2)

Column Name	Column Type	Source
Person	string	people...
Age	integer	people...

Lookup Flow (3)

Column	Type	Source	Keep
Age Group	un...	ag...	☐
Start	un...	ag...	☐
End	un...	ag...	☐

age_range-LKP Test results

ainsi que le résultat du test du lookup :

Test results

Previous tests

age_range-LKP_at_14:27:51
age_range-LKP_2_at_14:26:57
people-FIN_at_14:26:38

Test results

(line)	Person	Age	Age Group
1	Bob	65	Older Adult
2	Kathy	12	Child
3	Dan	22	Young Adult
4	Beth	2	Toddler
5	Adam	44	Adult
6	Bill	27	Young Adult
7	Steve	52	Adult
8	Allan	15	Teen
9	Peter	15	Teen

age_range-LKP

Test results

Un range lookup peut donc remplacer aisément des IF imbriqués.

Tags

1. script
2. Spectre
3. Workbench