

EXPERTIZERS

NOUVELLES FONCTIONS ISSUES DE REMONTEES HOTLINE (référéncé 003)

A l'occasion d'applications développées chez des clients, la hotline EXPERTIZERS a été sollicitée pour répondre à de nouveaux besoins, objet de ce document.

En cataloguant ces nouvelles fonctions, nous les portons à la connaissance de tous pour leur propre usage.

1. INCORPORATION DIRECTE DANS UN MODELE METIER D'EVENEMENTS DEJA VALORISES

Veuillez vous reporter aux chapitres 2.1.3 et 2.1.4 de la résolution du cas DSI (Document Résolution du cas DSI normes CIGREF.pdf).

NB : Ce document est intégré dans le package de formation 'Pour en finir avec les prix de revient' disponible gratuitement sur demande à l'adresse info@expertizers.com)

2. REPARTITION DU COÛT D'UNE ACTIVITE DU *MODELE COMPTABLE* SUR PLUSIEURS ACTIVITES D'*UN MODELE METIER*

Veuillez vous reporter aux chapitres 2.1.3 et 2.1.5 de la résolution du cas DSI (Document Résolution du cas DSI normes CIGREF.pdf).

3. UTILISATION DU FICHIER DES MATIERES POUR ECLATER PAR POURCENTAGE UNE LIGNE D'EVENEMENT EN N LIGNES

Veuillez vous reporter au chapitre 2.1.6 de la résolution du cas DSI (Document Résolution du cas DSI normes CIGREF.pdf). L'exemple traité concerne la répartition de projets sur d'autres projets

4. RE-UNIFICATION D'UN CONCEPT AU SEIN D'UNE MATRICE DE RESULTATS

En traitant des évènements au sein d'un modèle métier, il peut arriver qu'un même concept se trouve éclaté sur plusieurs colonnes, ou qu'une colonne regroupe plusieurs concepts à éclater séparément.

Dans de tels cas, pour disposer d'une matrice de résultats prête à répondre à toutes les analyses, il convient donc de réajuster les données dans une forme directement exploitable.

Voici deux situations présentées en exemple.

Situation 1 : Le concept est éclaté sur plusieurs colonnes

Product Code	Event Code	Activity Code	Activity Name	Activity Value	Activity Cost	Activity Text
SERVICE_176	P0602	MAICOR_EXT_CASH		123,0000	45634,4299	SERVICE_176
SERVICE_176	P0602	MAICOR_INT_CASH		231,0000	93374,3409	SERVICE_176
SERVICE_176	P0602	SUPN2F_EXT_CASH		50,5000	18736,0871	SERVICE_176
SERVICE_176	P0602	SUPN2F_INT_CASH		136,0000	54973,6379	SERVICE_176
SERVICE_176	P0602	MAICOR_EXT_CASH		179,0000	66411,0809	SERVICE_176
SERVICE_176	P0602	MAICOR_INT_CASH		136,0000	54973,6379	SERVICE_176
SERVICE_176	P1854	DEPPRO_EXT_CASH	P0176::IFRS - besoin comptable	7,7700	2882,7603	SERVICE_176
SERVICE_176	P1854	DEPPRO_EXT_CASH	P0321::Upgrade Appli XX	19,1660	7110,8088	SERVICE_135
SERVICE_176	P1854	DEPPRO_EXT_CASH	P0599::Appli XX et ZZ - Mainten.	24,8640	9224,8331	SERVICE_135
SERVICE_176	P1854	DEPPRO_INT_CASH	P0176::IFRS - besoin comptable	9,9000	4001,7575	SERVICE_176
SERVICE_176	P1854	DEPPRO_INT_CASH	P0321::Upgrade Appli XX	24,4200	9871,0017	SERVICE_135
SERVICE_176	P1854	DEPPRO_INT_CASH	P0599::Appli XX et ZZ - Mainten.	31,6800	12805,6239	SERVICE_135

Le concept de **SERVICE** est porté par la variable `product_code`, mais dès lors que le contenu de la variable `activity_name` contient les signes `::`, le concept de SERVICE à récupérer est le contenu de la variable `activity_text`.

De même, le concept de **PROJET** est porté par la variable `event_code`, mais dès lors que le contenu de la variable `activity_name` contient les signes `::`, le concept de PROJET à récupérer est le contenu précédant les signes `::`

Question : Comment restructurer les concepts de SERVICE et PROJET, chacun dans une colonne unique d'analyse.

Solution : Créer une variable avec règle de calcul dans le descriptif du fichier d'évènements (cf chapitre 14 du document `formation_express_expertizers_1.pdf`) de format `varchar` (alphanumérique) et de longueur **strictement égale à 51**.

La détection de la variable de longueur **51**, a pour effet de n'appliquer la règle qu'au moment de la génération de la matrice de résultat (choix Résultats / Lot par activité + données d'évènements), donc selon le modèle : `BUSINESS_EVENTS_BY_ACTIVITY` ou `ACCOUNTING_EVENTS_BY_ACTIVITY`.

Voici les deux variables créées avec leur règle qui utilisent les fonctions `Sql LOCATE` (recherche) et `SUBSTR` (extraction). La variable utilisée `alpha_name_x` (proposée dans la liste des variables disponibles pour établir les règles) correspond à ce niveau à `activity_name`.

```
service_dsi = if LOCATE(alpha_name_x, '::') > 0 then SUBSTR(activity_text, 1, 50) else product_code
endif
projet_dsi = if LOCATE(alpha_name_x, '::') > 0 then
SUBSTR(alpha_name_x, 1, LOCATE(alpha_name_x, '::') - 1) else SUBSTR(event_code, 1, 50) endif
```

Les voici représentées à la fin du descriptif du fichier d'évènements

service_dsi	varchar	51	service_dsi = if LOCATE(alpha_name_x,':') > 0 then SUBSTR(activity_text,1,50) else product_code endif
projet_dsi	varchar	51	projet_dsi = if LOCATE(alpha_name_x,':') > 0 then SUBSTR(alpha_name_x,1,LOCATE(alpha_name_x,':')-1) else SUBSTR(event_code,1,50) endif

Et voilà le résultat après la génération de la table BUSINESS_EVENTS_BY_ACTIVITY

BUSINESS_EVENTS_BY_ACTIVITY						
	Product Code	Event Code	Activity Name	Activity Text	Service Dsi	Projet Dsi
	SERVICE_176	P0602		SERVICE_176	SERVICE_176	P0602
	SERVICE_176	P0602		SERVICE_176	SERVICE_176	P0602
	SERVICE_176	P0602		SERVICE_176	SERVICE_176	P0602
	SERVICE_176	P0602		SERVICE_176	SERVICE_176	P0602
	SERVICE_176	P0602		SERVICE_176	SERVICE_176	P0602
	SERVICE_176	P0602		SERVICE_176	SERVICE_176	P0602
	SERVICE_176	P1854	P0176::IFRS - besoin comptable	SERVICE_176	SERVICE_176	P0176
	SERVICE_176	P1854	P0321::Upgrade Appli XX	SERVICE_135	SERVICE_135	P0321
	SERVICE_176	P1854	P0599::Appli XX et ZZ - Maintena	SERVICE_135	SERVICE_135	P0599
	SERVICE_176	P1854	P0176::IFRS - besoin comptable	SERVICE_176	SERVICE_176	P0176
	SERVICE_176	P1854	P0321::Upgrade Appli XX	SERVICE_135	SERVICE_135	P0321
	SERVICE_176	P1854	P0599::Appli XX et ZZ - Maintena	SERVICE_135	SERVICE_135	P0599

NB : S'il s'avérait que les colonnes service_dsi et/ou projet_dsi restent à blanc, nul doute que l'exécution des règles se soit mal passée, notamment par dépassement de la longueur des variables réceptrices limitée à 51, d'où l'utilité des expressions d'extraction SUBSTR(activity_text,1,50) et SUBSTR(event_code,1,50), soit les 50 premiers caractères des variables activity_text et event_code. D'ailleurs on aurait pu aller jusqu'à 51 !

Situation 2 : Une variable contient plusieurs concepts avec des séparateurs d'identification

Par manque de variables disponibles, vous pouvez recourir à l'utilisation d'une variable pour contenir plusieurs données côte à côte. C'est le cas notamment de la variable activity_text que vous pouvez utiliser dans ce sens, avec des caractères spéciaux pour séparer et identifier le contenu des variables mises bout à bout.

Par exemple : activity_text = variable_1 + '<<' + variable_2 + '>>' + variable_3

Les variables d'extraction avec leur règle seraient alors :

```
concept_1 = if LOCATE(activity_text,'<<') > 0 then SUBSTR(activity_text,1,LOCATE(activity_text,'<<') - 1) else " endif
```

```
concept_2 = if LOCATE(activity_text,'<<') > 0 and LOCATE(activity_text,'>>') > 0 then
```

```
SUBSTR(activity_text, LOCATE(activity_text,'<<') + 2, LOCATE(activity_text,'>>') -
LOCATE(activity_text,'<<') - 2) else " endif
```

```
concept_3 = if LOCATE(activity_text,'>>') > 0 then SUBSTR(activity_text,LOCATE(activity_text,'>>') +
2) else " endif
```

Les voici représentées à la fin du descriptif du fichier d'évènements

concept_1	varchar	51	concept_1 = if LOCATE(activity_text,'<<') > 0 then SUBSTR(activity_text,1,LOCATE(activity_text,'<<') - 1) else " endif
concept_2	varchar	51	concept_2 = if LOCATE(activity_text,'<<') > 0 and LOCATE(activity_text,'>>') > 0 then SUBSTR(activity_text, LOCATE(activity_text,'<<') + 2, LOCATE(activity_text,'>>') - LOCATE(activity_text,'<<') - 2) else " endif
concept_3	varchar	51	concept_3 = if LOCATE(activity_text,'>>') > 0 then SUBSTR(activity_text,LOCATE(activity_text,'>>') + 2) else " endif

Et voilà le résultat après la génération de la table BUSINESS_EVENTS_BY_ACTIVITY. Les trois parties contenues dans activity_text ont bien été individualisées.

BUSINESS_EVENTS_BY_ACTIVITY				
	Activity Text	Concept 1	Concept 2	Concept 3
	aaaaaaaa<<bbbbbbbbbbb>>cccccccc	aaaaaaaa	bbbbbbbbbbb	cccccccc
	aaaaaaaa<<bbbbbbbbbbb>>cccccccc	aaaaaaaa	bbbbbbbbbbb	cccccccc
	aaaaaaaa<<bbbbbbbbbbb>>cccccccc	aaaaaaaa	bbbbbbbbbbb	cccccccc
	aaaaaaaa<<bbbbbbbbbbb>>cccccccc	aaaaaaaa	bbbbbbbbbbb	cccccccc
	aaaaaaaa<<bbbbbbbbbbb>>cccccccc	aaaaaaaa	bbbbbbbbbbb	cccccccc

NB : Il ne faut pas avoir peur des fonctions SQL. Elles ne sont pas plus difficiles à appréhender que les fonctions Excel ! Rappelez-vous que vous avez un accès rapide à toute la panoplie des fonctions SQL en tapant (sur google ou tout autre moteur de recherche) les mots clés : sql anywhere fonctions string date numeric integer