



Bonjour,

Je fais parti d'une communauté regroupant aujourd'hui une 60aine d'indépendants spécialisés dans le décisionnel.

L'objectif de cette communauté est de capitaliser sur les compétences d'experts du métier et de les diffuser vers l'extérieur, afin d'y faire connaître notre expertise.

Cette reconnaissance, nous souhaitons l'obtenir :

- ✔ par des formations (nous sommes nombreux à donner des formations pour le compte de grands cabinets de formation)
- ✔ par la diffusion de fiches :
  - de type fiche produit,
  - de type best practice (une fiche n'est diffusée qu'après validation du groupe d'experts, dont certains certifiés composés de 4 à 5 personnes)
  - de type astuce de développement,
- ✔ par du service (prestation d'expertise, de conseil, d'architecture, d'audit) soit directement pour les clients, soit pour le compte des éditeurs à travers leur pôle Professional Services

Contrairement aux fiches de type Best Practice que vous avez pu lire les derniers mois, cette fiche astuce a un format beaucoup plus court et ne correspond pas à une directive de développement. C'est un exemple pratique, que vous utilisez peut être déjà pour certains d'entre vous, et que nous trouvons intéressant de diffuser pour ceux qui ne la connaîtraient pas.

J'ai la joie et l'honneur de vous faire part de la première fiche, qui j'espère vous plaira.

Si vous avez des questions, ou des remarques, n'hésitez pas à nous les faire parvenir. Un espace d'échange est en cours de construction, mais vous pouvez d'ores et déjà nous joindre à l'adresse suivante : [expert\\_informatica@unovia.fr](mailto:expert_informatica@unovia.fr)

Tout le travail accompli par ce groupe est fait de façon bénévole, n'hésitez donc pas à les remercier et à les encourager.

Fuji TATEMU



Relecteurs : Christophe FOURNEL, Fabien DUPREY, Stéphane THIA



SOMMAIRE

|   |          |
|---|----------|
| <b><u>MISE A JOUR D'UNE TABLE RELATIONNELLE PAR SURCHARGE</u></b> ..... | <b>3</b> |
| A. Contexte.....  | 3        |
| B. Rappel .....   | 3        |
| C. Désynchronisation des définitions SGBD et Informatica .....          | 4        |
| a. 1 <sup>er</sup> exemple de désynchronisation .....                   | 4        |
| b. 2 <sup>nd</sup> exemple de désynchronisation .....                   | 5        |
| D. Pré requis de la surcharge.....                                      | 6        |
| a. Pré requis.....  | 6        |
| b. Exemple d'utilisation de la surcharge .....                          | 6        |
| <b>CONCLUSION</b> .....   | <b>9</b> |
| <b>DOCUMENTS DE REFERENCE</b> .....                                     | <b>9</b> |
| <b>GLOSSAIRE</b> .....  | <b>9</b> |





## MISE A JOUR D'UNE TABLE RELATIONNELLE PAR SURCHARGE

### A. CONTEXTE

Ce document évoque la technique de « surcharge » lors de la mise à jour des tables relationnelles.

Il arrive parfois que des utilisateurs nous demandent de modifier la valeur d'un indicateur à partir de plusieurs règles fonctionnelles.

Lorsque cet indicateur se trouve dans une table relationnelle, souvent ces règles impliquent de s'appuyer sur les colonnes qui composent la clef primaire ainsi que sur d'autres.

Dans la suite du document, nous allons montrer que cette technique peut répondre à cette problématique.

### B. RAPPEL

Sur Informatica, la mise à jour d'une table se fait par défaut à partir de la définition de sa clef primaire. J'insiste sur le terme « définition de la table sur Informatica » car parfois, pour des raisons diverses (protocole ou l'oubli de la mise à jour de la définition), il peut arriver que les définitions SGBD et Informatica ne soient pas synchrones :

Si nous supposons que la définition d'une table sur le SGBD et sur Informatica soit la suivante :

| TABLE SGBD/INFORMATICA : TABLE T |                  |
|----------------------------------|------------------|
| PK_1                             | C1 (= Colonne 1) |
| PK_2                             | C2 (= Colonne 2) |
|                                  | C3 (= Colonne 3) |
|                                  | C4 (= Colonne 4) |
|                                  | C5 (= Colonne 5) |

La requête de mise à jour par défaut sur Informatica qui en découle est :

```
UPDATE T
SET      C3 = :TU.C3
         C4 = :TU.C4
         C5 = :TU.C5
WHERE   :TU.C1 = C1
AND     :TU.C2 = C2
```

#### Notez que

- dans le mapping, tous les ports de la cible sont connectés
- les éléments précédés par « :TU. » représentent les valeurs envoyées par Informatica au SGBD
- en condition (dans la clause WHERE) les valeurs envoyées par Informatica au SGBD doivent toujours être à gauche du signe « = »
- et que la mise à jour sur le SGBD se fait sur la clef primaire de la table T.





## C. DESYNCHRONISATION DES DEFINITIONS SGBD ET INFORMATICA

Maintenant, supposons que :

- ✓ On enrichit la table T par les 2 champs (colonnes C6 et C7)
- ✓ Et qu'on fait entrer dans la composition de la clef primaire de la table T le champ C3.

| TABLE SGBD T |    |
|--------------|----|
| PK_1         | C1 |
| PK_2         | C2 |
| PK_3         | C3 |
|              | C4 |
|              | C5 |
|              | C6 |
|              | C7 |

### a. 1<sup>ER</sup> EXEMPLE DE DESYNCHRONISATION

On oublie de mettre à jour sur Informatica l'entrée de la colonne C3 dans la composition de la clef primaire (PK\_3).

| TABLE INFORMATICA T |    |
|---------------------|----|
| PK_1                | C1 |
| PK_2                | C2 |
|                     | C3 |
|                     | C4 |
|                     | C5 |
|                     | C6 |
|                     | C7 |

La requête de mise à jour par défaut sur Informatica devient alors :

**UPDATE T**

**SET** C3 = :TU.C3

C4 = :TU.C4

C5 = :TU.C5

C6 = :TU.C6

C7 = :TU.C7

**WHERE** :TU.C1 = C1

**AND** :TU.C2 = C2

#### Notez que

- Dans le mapping, tous les ports de la cible sont connectés
- Les éléments précédés par « :TU. » représentent les valeurs envoyées par Informatica au SGBD
- En condition (dans la clause WHERE) les valeurs envoyées par Informatica au SGBD doivent toujours être à gauche du signe « = »
- Et que la mise à jour sur le SGBD ne se fait que sur 2 champs sur 3 qui composent la clef primaire de la table T.



## b. 2<sup>ND</sup> EXEMPLE DE DESYNCHRONISATION

On oublie de mettre jour sur Informatica l'ajout des 2 nouvelles colonnes (C6, C7).

| TABLE INFORMATICA T |    |
|---------------------|----|
| PK_1                | C1 |
| PK_2                | C2 |
| PK_3                | C3 |
|                     | C4 |
|                     | C5 |

La requête de mise à jour par défaut sur Informatica devient alors :

**UPDATE T**

**SET** C4 = :TU.C4

C5 = :TU.C5

**WHERE** :TU.C1 = C1

**AND** :TU.C2 = C2

**AND** :TU.C3 = C3

### Notez que

- dans le mapping, tous les ports de la cible sont connectés
- les éléments précédés par « :TU. » représentent les valeurs envoyées par Informatica au SGBD
- en condition (dans la clause WHERE) les valeurs envoyées par Informatica au SGBD doivent toujours être à gauche du signe « = »
- que la mise à jour sur le SGBD se fait sur la clef primaire de la table T
- mais qu'elle ne met pas à jour sur le SGBD les 2 nouvelles colonnes de la table T.

**Donc n'oubliez pas avant toutes modifications du traitement Informatica de synchroniser les définitions quand cela est nécessaire.**



## D. PRE REQUIS DE LA SURCHARGE

Ici, nous mettons en avant l'option de surcharge relative à la mise à jour sur les définitions cibles des tables relationnelles dans Informatica.

### a. PRE REQUIS

Pour que cette option fonctionne, il faut d'abord que l'enregistrement source soit « tagué » pour la mise à jour. Pour ce faire, il suffit que :

- ✓ La valeur de l'option « Treat source rows as » soit à « Update » (*Dans le workflow manager, dans l'onglet « Properties » de la session*)
- ✓ ou que la valeur de l'option « Treat source rows as » soit à « Data-driven » et que la transformation « Update Strategy » tague certains enregistrements pour la mise à jour (DD\_UPDATE)

Ensuite, il faut que la stratégie d'alimentation de la table cible (*Dans le workflow manager, dans l'onglet « Mapping » de la session*) prenne en compte au moins l'une des 3 options de mise à jour :

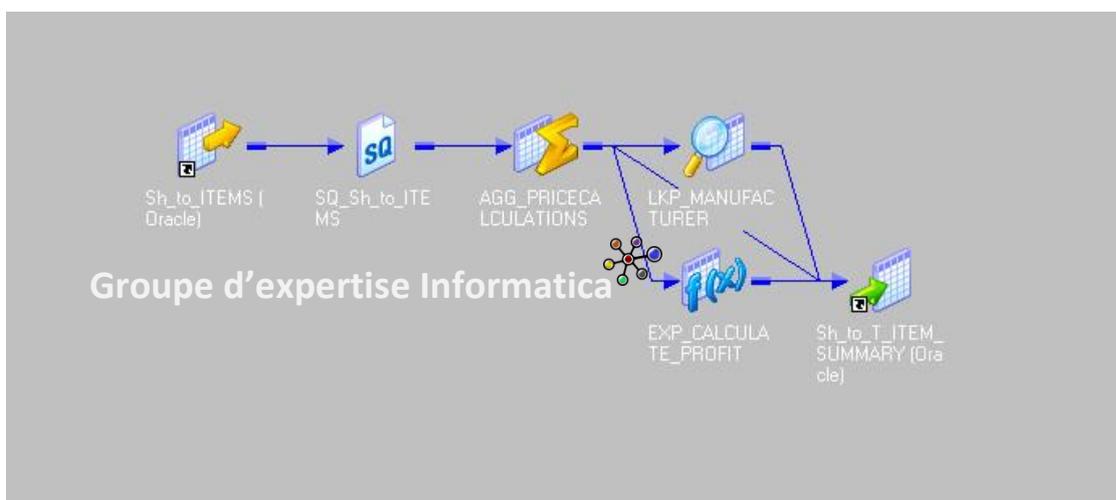
- ✓ update as update,
- ✓ update as insert,
- ✓ update else insert

### b. EXEMPLE D'UTILISATION DE LA SURCHARGE

Prenons l'exemple d'une société spécialisée dans la vente de matériel de plongée. Celle-ci traite avec différents fournisseurs qui lui vendent tout type de produit. Pour suivre sa marge, cette société a besoin de connaître par identifiant de fournisseur (clé primaire de la table) différents indicateurs :

- ✓ Le prix minima pratiqué
- ✓ Le prix maxima pratiqué
- ✓ Le prix moyen
- ✓ Le profit moyen

Pour répondre à ce besoin, nous développons donc le flux ci-dessous





Nous obtenons donc le résultat suivant :

|    | MANUFACTURER_ID | MANUFACTURER_NAME | MAX_PRICE | MIN_PRICE | AVG_PRICE | AVG_PROFIT |
|----|-----------------|-------------------|-----------|-----------|-----------|------------|
| 1  | 100             | Nike              | 365       | 169,95    | 261,24    | 52,25      |
| 2  | 104             | Head              | 179       | 52        | 98,67     | 19,73      |
| 3  | 108             | Sportstar         | 280       | 18        | 149       | 29,8       |
| 4  | 101             | OBrien            | 188       | 44,95     | 134,32    | 26,86      |
| 5  | 105             | Jesper            | 325       | 34,95     | 133,65    | 26,73      |
| 6  | 109             | WindJammer        | 430       | 395       | 412,5     | 82,5       |
| 7  | 102             | Mistral           | 390       | 70        | 200       | 40         |
| 8  | 106             | Acme              | 195       | 56,95     | 143,65    | 28,73      |
| 9  | 110             | Monsoon           | 280       | 280       | 280       | 56         |
| 10 | 103             | Spinnaker         | 70        | 29        | 52,98     | 10,6       |
| 11 | 107             | Medallion         | 235       | 19,95     | 98,65     | 19,73      |

Au cours du mois, les fournisseurs suivants Spinnaker, Sportstar et Medallion nous annonce qu'ils vont augmenter le prix de certains articles.

En ré-exécutant le flux d'intégration précédent, nous allons recharger l'intégralité des produits, mais nous souhaiterions uniquement mettre à jour les données concernant ces trois fournisseurs.

Si nous ne changeons rien dans le flux, INFORMATICA va générer la requête par défaut suivante :

#### UPDATE T\_ITEM\_SUMMARY

```
SET MANUFACTURER_NAME = : TU.MANUFACTURER_NAME,  
MAX_PRICE = : TU.MAX_PRICE,  
MIN_PRICE = : TU.MIN_PRICE,  
AVG_PRICE = : TU.AVG_PRICE,  
AVG_PROFIT = :TU.AVG_PROFIT,
```

```
WHERE :TU.MANUFACTURER_ID = MANUFACTURER_ID
```

#### Notez que

- Même si nous travaillons avec un shortcut, il faut indiquer le nom de la table.

Pour nous permettre de réaliser la mise à jour uniquement sur les données des trois fournisseurs, il nous faut donc utiliser la clause **Update Override** dans les propriétés de la Target et écrire la requête suivante :

#### UPDATE T\_ITEM\_SUMMARY

```
SET MAX_PRICE = : TU.MAX_PRICE,  
MIN_PRICE = : TU.MIN_PRICE,  
AVG_PRICE = : TU.AVG_PRICE,  
AVG_PROFIT = :TU.AVG_PROFIT
```

```
WHERE :TU.MANUFACTURER_NAME= MANUFACTURER_NAME  
AND MANUFACTURER_NAME IN ('Sportstar', 'Spinnaker', 'Medallion')
```

#### Notez que

- Nous ne passons plus par la clé primaire, mais par le nom du fournisseur.
- La mise à jour se fait uniquement s'il appartient à la liste



|    | MANUFACTURER_ID | MANUFACTURER_NAME | MAX_PRICE | MIN_PRICE | AVG_PRICE | AVG_PROFIT |
|----|-----------------|-------------------|-----------|-----------|-----------|------------|
| 1  | 100             | Nike              | 365       | 169,95    | 261,24    | 52,25      |
| 2  | 104             | Head              | 179       | 52        | 98,67     | 19,73      |
| 3  | 108             | Sportstar         | 280       | 21,6      | 150,8     | 30,16      |
| 4  | 101             | OBrien            | 188       | 44,95     | 134,32    | 26,86      |
| 5  | 105             | Jesper            | 325       | 34,95     | 133,65    | 26,73      |
| 6  | 109             | WindJammer        | 430       | 395       | 412,5     | 82,5       |
| 7  | 102             | Mistral           | 390       | 70        | 200       | 40         |
| 8  | 106             | Acme              | 195       | 56,95     | 143,65    | 28,73      |
| 9  | 110             | Monsoon           | 280       | 280       | 280       | 56         |
| 10 | 103             | Spinnaker         | 70        | 34,8      | 54,92     | 10,98      |
| 11 | 107             | Medallion         | 235       | 23,94     | 99,98     | 20         |

Autre hypothèse, nous revoions les prix de ventes, mais nous souhaiterions les modifier que si le profit était supérieur ou égale à 20.

#### UPDATE T\_ITEM\_SUMMARY

```
SET MAX_PRICE = :TU.MAX_PRICE,  
MIN_PRICE = :TU.MIN_PRICE,  
AVG_PRICE = :TU.AVG_PRICE,  
AVG_PROFIT = :TU.AVG_PROFIT
```

```
WHERE :TU.MANUFACTURER_ID= MANUFACTURER_ID  
AND :TU.AVG_PROFIT >= 20
```

#### Notez que

- Nous ne passons par la clé primaire.
- La mise à jour se fait uniquement si la moyenne du nouveau profit calculé est supérieur ou égale à 20

Si nous regardons le résultat, nous remarquons, par rapport au chargement initial que deux lignes n'ont pas été modifiée malgré les changements tarifaires.

|    | MANUFACTURER_ID | MANUFACTURER_NAME | MAX_PRICE | MIN_PRICE | AVG_PRICE | AVG_PROFIT |
|----|-----------------|-------------------|-----------|-----------|-----------|------------|
| 1  | 100             | Nike              | 365       | 169,95    | 261,24    | 52,25      |
| 2  | 104             | Head              | 179       | 52        | 98,67     | 19,73      |
| 3  | 108             | Sportstar         | 280       | 21,6      | 150,8     | 30,16      |
| 4  | 101             | OBrien            | 188       | 44,95     | 134,32    | 26,86      |
| 5  | 105             | Jesper            | 325       | 34,95     | 133,65    | 26,73      |
| 6  | 109             | WindJammer        | 430       | 395       | 412,5     | 82,5       |
| 7  | 102             | Mistral           | 390       | 70        | 200       | 40         |
| 8  | 106             | Acme              | 195       | 56,95     | 143,65    | 28,73      |
| 9  | 110             | Monsoon           | 280       | 280       | 280       | 56         |
| 10 | 103             | Spinnaker         | 70        | 34,8      | 54,92     | 10,98      |
| 11 | 107             | Medallion         | 235       | 23,94     | 99,98     | 20         |





---

## CONCLUSION

---

Cette technique nous permet de pouvoir rajouter une clause **where** à la requête de mise à jour de la table cible. **C'est uniquement dans ce cas précis, qu'il faut l'utiliser, car cela peut détériorer les performances de chargement, si vous vous basez sur une colonne non indexée.**

Cependant, il faut bien faire attention, car la non utilisation

- ✓ colonnes de références (colonnes de la table cible)
- ✓ ports de références (:TU.nom du port)

peuvent entraîner respectivement les conséquences suivantes

- ✓ **Risque de mise à jour intégrale de la table**
- ✓ **Pour chaque enregistrement traité dans le flux, mise à jour de l'ensemble des enregistrements de la table cible validant la condition**

Pour plus de détail, voyez la référence.

---

## DOCUMENTS DE REFERENCE

---

**PowerCenter Designer Guide 8.6.1 ou 9.0.1**

- ✓ Chapiter 5 : Mappings
  - Working with Relational Targets in a Mapping
  - Configuring the Target Update Override

---

## GLOSSAIRE

---

- ✓ **Clef primaire** : Ensemble de colonnes qui permet de cibler un unique enregistrement dans une table relationnelle.
- ✓ **SGBD** : Système de Gestion de Base de Données.