

# Connaissez-vous IBM DB2 DataJoiner ?

Olivier BONNET

## Introduction

**S**i vous avez des bases de données relationnelles provenant de fournisseurs différents (IBM DB2, Oracle™, Sybase™, Informix™, Microsoft™ SQL Server, etc.) et réparties sur des sites différents, vous souhaitez peut-être pouvoir utiliser, de façon transparente et homogène, toutes les données, sans avoir à vous soucier de leur localisation et des types de bases de données.

Par exemple, vous aimeriez pouvoir faire une jointure entre une table Oracle et une table Informix. Cela est possible grâce à *IBM DB2 DataJoiner*.

Vous souhaitez peut-être aussi répliquer des données entre des bases DB2 et des bases non-DB2. Cela est possible en utilisant conjointement *IBM DB2 DataJoiner* et *IBM DB2 DataPropagator*.

L'objet de cet article est de présenter succinctement ces possibilités.

## A quoi sert DataJoiner ?

*IBM DB2 DataJoiner* est un produit qui tourne sous AIX ou Windows NT™, et qui permet d'accéder de façon transparente à la plupart des bases de données relationnelles du marché :

- Toute la famille DB2 bien sûr, mais aussi :
- Oracle,
- Sybase,
- Informix,
- Microsoft SQL Server,
- etc.

En fait, DataJoiner comporte un moteur DB2 complet (il s'agit d'un *DB2 Common Server V2.1.2*) avec des fonctions et des tables catalogues enrichies par rapport à celles d'un DB2 normal.

DataJoiner permet de faire voir toutes les bases de données (Oracle, Sybase, ...) comme s'il s'agissait d'une seule et même base DB2.

Cela veut dire que, dans un même ordre SQL, vous pouvez, par exemple, faire référence à une table Oracle située à Lyon, une table Sybase située à Marseille et une table *DB2 for OS/390* située à Lille.

**DataJoiner permet de faire des jointures entre des tables situées dans plusieurs bases distinctes et de nature différente !**

Et comme chaque table est vue comme une table DB2, vous pouvez utiliser la même application pour attaquer une base DB2, une base Oracle, une base SQL Server, etc.

Par exemple, **IBM DB2 DataPropagator** (DPropR) permet de répliquer des données de DB2 vers DB2, quelle que soit la plate-forme. Le composant “*Capture*” capture les mises à jour faites sur les tables sources, et le composant “*Apply*” récupère les mises à jour capturées puis les applique sur les tables cibles.

Mais DPropR ne sait travailler qu’avec des tables DB2.

Pour répliquer des données de tables DB2 vers des tables non-DB2, il suffit d’intercaler **DataJoiner** entre la base DB2 et la base non-DB2. Alors le composant “*Apply*” de DPropR sera capable de mettre à jour les tables non-DB2.

La réplication en sens inverse (de non-DB2 vers DB2) est également possible.

## Utilisation de DataJoiner

Vous devriez penser à utiliser DataJoiner dès lors que vous avez l’un des besoins suivants :

- ◆ Migration d’applications.  
Vous avez, par exemple, une application qui a été développée pour DB2 et vous souhaitez la faire tourner sur un autre type de base de données, sans avoir à réécrire tous les programmes.
- ◆ Alimentation de *DataWarehouse*, ou constitution d’un DataWarehouse virtuel, pour des applications d’aide à la décision ou de type “Infocentre”.  
Avec DataJoiner vous pourrez faire des jointures entre des tables situées dans des bases de données éloignées et même hétérogènes.
- ◆ Réplication de données hétérogènes (DB2 et non-DB2).

## Fonctionnement

### Jointure entre tables hétérogènes

Imaginons que vous souhaitiez faire une jointure entre une table DB2, appelée “AAA.TABDB2”, située sur un système *host* OS/390 et une table Oracle, appelée “XXX.TABORA”, située sur une machine AIX. Vous allez intercaler DataJoiner entre la base DB2 et la base Oracle.

### Configuration

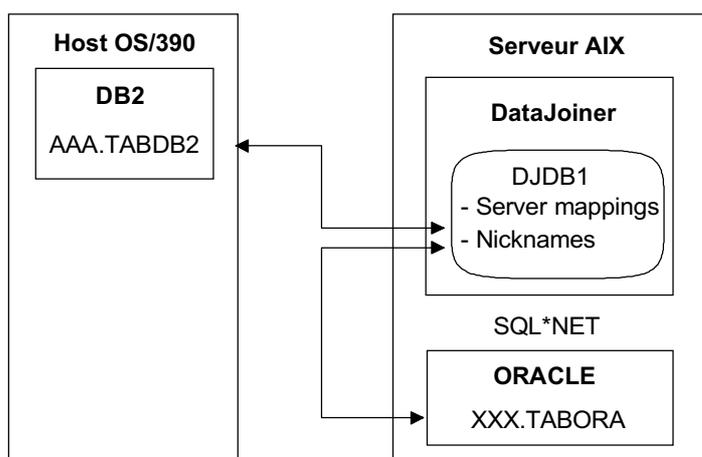


Fig. 1 - Configuration permettant de faire une jointure entre des tables hétérogènes.

**Remarque :** “SQL\*NET” est le module Oracle d’accès client à une base Oracle.

Pour accéder aux tables TABDB2 et TABORA, on crée tout d'abord une base de données (DJDB1) dans DataJoiner, puis on définit des "nicknames" qui sont, en quelque sorte, des pointeurs vers les tables réelles : tout ordre SQL exécuté sur un *nickname* sera, en fait, rerouté vers la base de données (DB2 ou Oracle) concernée.

### **Mise en oeuvre détaillée**

Pour mettre en oeuvre la configuration décrite ci-dessus, il faut :

- 1 - Installer DataJoiner sur le serveur AIX qui contient la base Oracle.
- 2 - Faire un Link-Edit de DataJoiner avec SQL\*NET.
- 3 - Créer une instance DataJoiner, démarrer DataJoiner (db2start) et créer une base de données DJDB1 dans DataJoiner.
- 4 - Établir la connexion DRDA entre la base de DataJoiner (DJDB1) et la base DB2.

**Remarque :** DataJoiner inclut un DDCS (*Distributed Database Connection Services*), amélioré pour pouvoir faire du DRDA en TCP/IP avec un *DB2 for OS/390* version 5 (ou un *DB2 for AS/400 V420*).

Il n'est donc pas nécessaire d'installer un DDCS ou un *DB2 Connect* supplémentaire.

Il faut configurer la "Communications Database" du côté *host*, puis cataloguer la base DB2 à partir de DataJoiner en utilisant les commandes DB2 suivantes :

- **catalog tcpip node DB2NODE ...**
- **catalog database DB2DB as DB2 at node DB2NODE ...**
- **catalog dcs database DB2**

Il faut ensuite tester la connexion à la base DB2, et faire le "bind" de "ddcsmvs.lst" :

- **bind @ddcsmvs.lst blocking all**
- 5 - Se connecter à la base DJDB1 par l'instruction SQL suivante :
    - **connect to DJDB1**
  - 6 - Créer des "server mappings" dans DataJoiner.

Ce sont des définitions, stockées dans le catalogue de DataJoiner, qui indiquent les caractéristiques (nom, type, version, emplacement) des bases de données à accéder :

    - **create server mapping from DB2 to node DB2NODE database DB2DB ...**
    - **create server mapping from ORA to node ORANODE database ORADB ...**

**Remarque :** Le nom de noeud "ORANODE" doit avoir été précisé dans SQL\*NET, dans le fichier "tnsnames.ora".

- 7 - Créer des "user mappings" dans DataJoiner, qui indiquent quels profils d'utilisateurs et quels mots de passe doivent être utilisés lors de l'accès aux bases de données DB2 et Oracle :
  - **create user mapping from USERDB2 to server DB2 ...**
  - **create user mapping from USERORA to server ORA ...**
- 8 - Créer des "nicknames" pour les tables à accéder :
  - **create nickname TABA for DB2.AAA.TABDB2**
  - **create nickname TABB for ORA.XXX.TABORA**
- 9 - Ensuite, en étant connecté sur la base DJDB1, on peut exécuter n'importe quel ordre SQL, directement sur les *nicknames* TABA et TABB, comme s'il s'agissait de vraies tables situées dans la base DJDB1.

Par exemple, l'ordre SQL suivant donne le résultat de la jointure entre la table DB2 et la table Oracle :

```
select A.col1, A.col2, B.colx
from TABA A, TABB B
where A.col1 = B.col1
```

Vous n'êtes, bien sûr, pas limités à des "SELECT" ; vous pouvez faire des INSERT, UPDATE, DELETE, etc., avec, cependant, la restriction suivante : un ordre SQL ne peut mettre à jour qu'une seule table à la fois (sauf si toutes les bases sont des bases DB2).

Vous pouvez, par exemple, alimenter directement une table Informix à partir d'une table SQL Server (ou autre), en utilisant l'ordre SQL suivant :

```
INSERT INTO tab_informix (col1, col2, ...)
SELECT a.colx, a.coly, ... FROM tab_sql_server a
```

Dans la liste de tâches ci-dessus, j'ai volontairement omis certaines activités, pour simplifier. Dans la réalité, il faut prendre en compte certaines contraintes :

- ◆ La séquence de tri des données n'est pas la même pour tous les types de bases de données. Cela est important car, quand on fait un "SELECT" avec une clause "WHERE", le résultat peut être différent selon que l'on travaille avec une séquence de tri ou une autre.
- ◆ Certains types de données, comme DATE, TIME, TIMESTAMP, n'ont pas la même signification pour toutes les bases de données.  
Par exemple, une zone de type "DATE" dans ORACLE correspond à une zone de type "TIMESTAMP" dans DB2.  
Quand on crée les *nicknames*, DataJoiner prend un certain nombre de décisions par défaut sur les types de données, mais on peut être amené à corriger ces décisions.

Il existe, dans DataJoiner, des commandes et des paramètres pour régler ces petits problèmes "d'ajustage" entre les différentes bases de données.

## Réplication de données d'une base DB2 vers une base non-DB2

Imaginons maintenant que vous souhaitiez propager des données depuis une table DB2, située sur un *host* OS/390, vers une table Oracle située sur une machine AIX.  
Vous allez utiliser, conjointement, DataJoiner et DataPropagator.

DataJoiner inclut une interface graphique, appelée DJRA (*DataJoiner Replication Administration*), qui est utilisée pour faire le paramétrage de la réplication de données.

Comme dans le cas précédent, vous devez installer et configurer DataJoiner sur la machine AIX.

## Configuration de l'installation

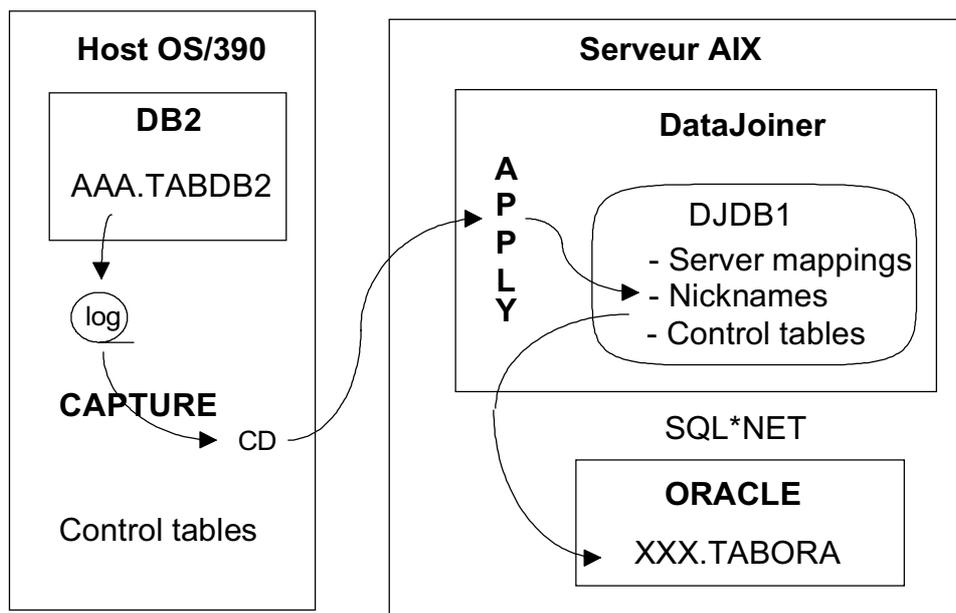


Fig. 2 - Configuration de l'installation pour la réplication des données d'une table DB2 vers une table non-DB2.

Le composant "Capture" de **DPropR** lit les *logs* DB2, y récupère les mises à jour faites sur la table source, puis stocke ces mises à jour dans une table intermédiaire appelée "**CD**" (*Change Data*). Le composant "Apply" récupère les mises à jour stockées dans la CD et les applique sur la cible (*nickname* de la table Oracle).

DataJoiner reroute les ordres "Insert", "Update" et "Delete" vers Oracle.

### Mise en oeuvre

Pour mettre en oeuvre cette configuration, vous allez effectuer les tâches suivantes :

1 - Activités identiques aux activités 1 à 7 de la liste précédente, avec deux petits changements :

2 -

3 -  Il ne faut pas créer de "server mapping" pour DB2.

4 -  On crée uniquement un "server mapping" pour la base Oracle.

5 -

6 -  Il ne faut pas créer de "user mapping" pour USERDB2.

7 -

8 - Installer le composant "Capture" de DPropR sur le *host*.

**Remarque** : DataJoiner contient le composant "Apply", donc aucun autre composant DPropR n'est nécessaire.

9 - Utiliser DJRA (*DataJoiner Replication Administration*) pour :

- Créer les tables de contrôle de DPropR dans la base DB2.

Vous pourrez alors faire le *bind* du "Capture" et de l'"Apply" sur la base DB2.

- Créer les tables de contrôle de DPropR dans la base DJDB1.

Vous pourrez alors faire le *bind* de l'"Apply" sur la base DJDB1.

- Définir les tables DB2 en tant que sources de réplication.

- Définir les tables Oracle en tant que cibles de réplication.  
En fait, on indique que la base qui contient les tables de contrôle de l'“Apply” est la base DJDB1. Puis DJRA détecte qu'il s'agit d'une base DataJoiner et propose alors la liste des bases de données accessibles : on choisit alors “**ORADB**”.  
Si la table cible n'existe pas, DJRA est capable de la créer automatiquement dans la base Oracle.  
De même, DJRA crée automatiquement le *nickname* correspondant dans la base DJDB1 si ce *nickname* n'existe pas.

- 10 - Le paramétrage étant terminé, vous pouvez alors démarrer le “Capture” sur l'OS/390 et l'“Apply” sur la machine AIX.  
Les mises à jour faites sur la table DB2 seront alors automatiquement répliquées vers la table Oracle.

## Réplication de données d'une base non-DB2 vers une base DB2

Nous allons voir, maintenant, comment répliquer des données dans le sens Oracle vers DB2.

**Remarque :** En environnement hétérogène, on ne peut pas faire de la réplication symétrique, c'est-à-dire d'une table A vers une table A', puis de la table A' vers la table A.

La **réplication symétrique** (appelée aussi “*Update Anywhere*”) ne fonctionne qu'entre des tables DB2 ou entre des tables DB2 et des tables Microsoft Access.

Dans les autres cas, on peut répliquer dans les deux sens, mais sur des tables différentes :

- ◆ De la table A vers la table A' dans un sens,
- ◆ De la table B' vers la table B dans l'autre sens.

### Configuration pour une réplication depuis Oracle vers DB2

Comme il n'y a pas de composant “Capture” sur Oracle, les fonctions de capture sont émulées par des *triggers* attachés à la table source.

Ces *triggers*, générés automatiquement par DJRA (*DataJoiner Replication Administration*), alimentent une table intermédiaire appelée **CCD** (*Consistent Change Data*).

DJRA crée automatiquement la CCD ainsi que le *nickname* qui permet à l'“Apply” d'accéder à la CCD.  
L'“Apply” récupère alors les mises à jour, puis les applique sur la table cible DB2.

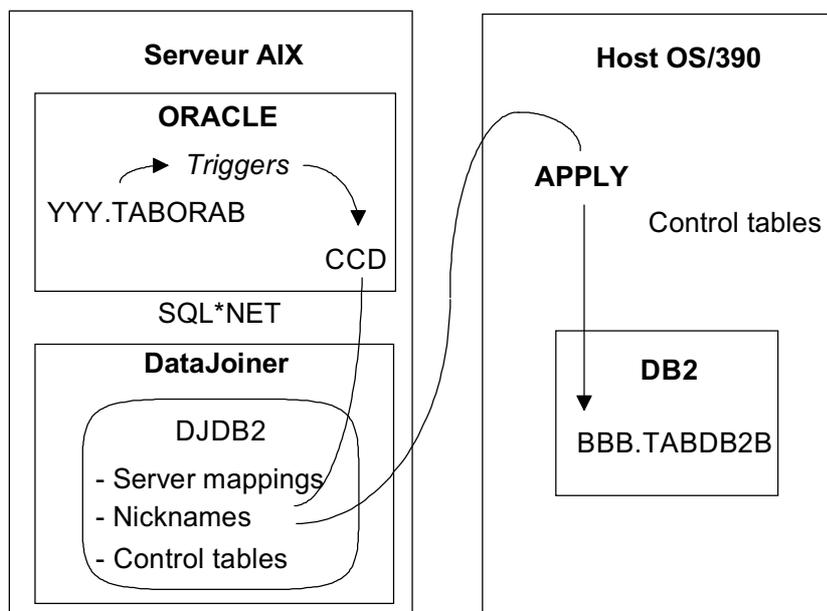


Fig. 3 - Configuration de l'installation pour la réplication des données d'une table non-DB2 vers une table DB2.

## Mise en oeuvre

Pour mettre en oeuvre cette configuration, vous allez effectuer les tâches suivantes :

- 1 - Créer une deuxième base dans DataJoiner : “**DJDB2**”.  
On doit, en effet, créer autant de bases dans DataJoiner qu’on a de bases de données sources.  
Par exemple, si l’on souhaite répliquer des données à partir de 5 bases de données, il faut créer 5 bases dans DataJoiner.
- 2 - Répéter, pour la base DJDB2, les opérations 4 à 7 du paragraphe “Mise en oeuvre détaillée” de la “Jointure entre tables hétérogènes”, sans créer le “*server mapping*” de DB2 ni le “*user mapping*” de USERDB2.
- 3 - Utiliser DJRA pour créer les tables de contrôle de DPropR, côté source (Oracle) et côté cible (DB2).  
En fait, quand on crée les tables de contrôle côté source (Oracle), certaines sont effectivement créées dans la base Oracle, avec leurs *nicknames* dans la base DJDB2, et d’autres sont créées dans la base DJDB2.
- 4 - Définir un *nickname* dans DJDB2, pour la table Oracle à répliquer (XXX.TABORAB).
- 5 - Utiliser DJRA pour enregistrer ce *nickname* en tant que source de réplication.  
DJRA génère les *triggers* qui émulent les fonctions de capture.  
DJRA génère aussi la table CCD intermédiaire dans la base Oracle, ainsi qu’un *nickname* pour cette CCD dans la base DJDB2.
- 6 - Utiliser DJRA pour définir la table DB2 (BBB.TABDB2B) en tant que cible de réplication.
- 7 - Installer le composant “Apply” de DPropR pour OS/390 sur le *host*.  
**Remarque** : On peut aussi utiliser l’“Apply” qui est inclus dans DataJoiner.  
Faire le *bind* de l’“Apply” sur la base DB2 et sur la base DJDB2.
- 8 - Lancer l’“Apply”.  
Il n’y a pas de “Capture” à démarrer.

## Les avantages de DataJoiner

Nous avons vu que DataJoiner permet d’effectuer des ordres SQL (*Insert*, *Update* et *Delete*) distribués sur des tables relationnelles hétérogènes (DB2 et non-DB2), éloignées ou locales (DataJoiner étant un DB2 complet, on peut en effet y créer des tables locales).

DataJoiner possède un catalogue global qui contient toutes les informations d’accès aux tables DB2 et non-DB2 que l’on souhaite utiliser, avec, entre autres, des informations relatives aux performances.

DataJoiner possède en effet un **optimiseur global** très performant, qui tient compte :

- ◆ de l’emplacement des données,
- ◆ de la puissance du réseau,
- ◆ du type et de la puissance de chaque serveur,
- ◆ et des informations statistiques prises sur les différents serveurs.

L’optimiseur analyse le type d’opération à effectuer et peut, par exemple, décider de réécrire un ordre SQL pour l’accélérer.

Nous avons vu aussi que DataJoiner permet de tenir compte de séquences de tri et de types de données différents, entre plusieurs types de bases de données.

Mais cela va plus loin : DataJoiner est capable de **compenser** l’absence de certaines fonctions dans certaines bases de données.

Par exemple, Sybase ne connaît pas la fonction “DECIMAL”.

Le mécanisme de compensation de DataJoiner va quand même permettre d’exécuter, sur une table Sybase, l’ordre SQL suivant :

```
SELECT nom FROM sybpay
WHERE DECIMAL(salaire) > 15000
```

L’inverse est également possible :

Si vous désirez exécuter, sur une table non-DB2, une fonction qui n’est pas disponible dans DataJoiner, vous pouvez le faire en utilisant le mode “*passthrough*”.

Ce mode permet de transmettre un ordre SQL à une base de données, sans interception par DataJoiner.

Par exemple, si vous souhaitez créer une table dans une base Sybase, vous utiliserez la séquence de commandes suivante :

```
SET PASSTHRU sybase
CREATE TABLE sybpay
  (nom VARCHAR(30),
   salaire SMALLMONEY,
   flag VARBINARY(1))
SET PASSTHRU RESET
```

## Pour en savoir plus

Pour aller plus loin, vous pouvez consulter les informations disponibles sur “*IBM DataJoiner*” et “*IBM DB2 DataPropagator*” aux adresses Internet suivantes :

- ◆ <http://www.software.ibm.com/data/datajoiner>
- ◆ <http://www.software.ibm.com/data/dpropr>

Si vous êtes plus particulièrement intéressé par la réplication de données entre des bases de données hétérogènes, vous trouverez le *redbook* **SG24-5463**, qui traite en détail de ce sujet, à l’adresse suivante :

- ◆ <http://www.redbooks.ibm.com/abstracts/sg245463.html>

■