Aide mémoire sur les jointures en SQL.

- Les articles -

Date de mise en ligne : mardi 14 septembre 2004

Description :

Comment gérer les jointures dans les requêtes SQL.

Environnement iSeries

Si vous avez d'autres activités que la programmation, qu'il vous soit nécessaire de faire de temps en temps une requête SQL, je suis prêt à parier que vous avez besoin de vous reporter à un aide mémoire.

Cet article décrit succintement les différents types de jointures en se basant sur un exemple simple.

- Le CROSS JOIN effectue un produit cartésien entre le contenu de deux tables.
- Le INNER JOIN permet de faire une jointure entre deux tables et de ramener les enregistrements de la première table qui ont une correspondance dans la seconde table.
- Le LEFT OUTER JOIN entre deux tables retourne tous les enregistrements ramenés par un INNER JOIN plus chaque enregistrement de la table 1 qui n'a pas de correspondance dans la table 2.
- Le RIGHT OUTER JOIN entre deux tables retourne tous les enregistrements ramenés par un INNER JOIN plus chaque enregistrement de la table 2 qui n'a pas de correspondance dans la table 1.
- Un EXCEPTION JOIN retourne uniquement les enregistrements de la table 1 qui n'on pas de correspondance dans la table 2.

Exemple utilisé

Considérons les quatre tables EMPLOYE, PROJET, DEPARTEMENT et ACTIVITES. Elles nous serviront d'exemple pour chaque type de jointure.

[1]

Règles de gestion :

- Chaque employé figure dans la table EMPLOYE.
- Un employé responsable d'un projet figure dans la table PROJET.
- Un employé fait partie d'un département.
- La liste des départements se trouve dans la table DEPARTEMENT.
- La table ACTIVITES contient une liste des activités assurées par le CE de l'entreprise.

Table des employés

```
EMPLOYE,
 EMPNO, numéro de l'employé
 LASTNAME, nom de l'employé
 WORKDEPT, département ou trava
```

EMPNO	LASTNAME	WORKDEPT
000020	THOMPSON	000001
000060	STERN	000002

000100	SPENSER	000003
000170	YOSHIMURA	000002
000180	SCOUTTEN	000002
000190	WALKER	000002
000250	SMITH	000004
000280	SCHNEIDER	000005
000300	SMITH	000005
000310	SETRIGHT	000005

Table des employés responsable d'un projet

```
PROJET,
 RESPEMP, numéro de l'employé responsable du projet
 PROJNO, numéro du projet
```

RESPEMP	PROJNO
000020	PL2100
000060	MA2110
000100	OP2010
000250	AD3112

Table des départements

```
DEPARTEMENT,
 DEPTNO, numéro de département
 DEPTNAME, Nom du département
```

DEPTNO	DEPTNAME
000001	PLANNING
000002	MANUFACTURING SYSTEMS
000003	SOFTWARE SUPPORT
000004	ADMINISTRATION SYSTEMS
000005	OPERATIONS

Table des activités du CE

```
ACTIVITES,
 ACTINAME, Nom activités
```

ACTINAME
TENNIS
CINEMA
COURS DE LANGUE

Le CROSS JOIN

Le **Cross join** effectue un produit cartésien entre le contenu de deux tables.

Les deux requêtes suivantes produisent le même résultat

```
SELECT * FROM EMPLOYE CROSS JOIN ACTIVITES
```

```
SELECT * FROM EMPLOYE, ACTIVITES
```

Résultat

EMPNO	LASTNAME	WORKDEPT	ACTINAME
000020	THOMPSON	000001	TENNIS
000020	THOMPSON	000001	CINEMA
000020	THOMPSON	000001	COURS DE LANGUE
000060	STERN	000002	TENNIS
000060	STERN	000002	CINEMA
000060	STERN	000002	COURS DE LANGUE
000100	SPENSER	000003	TENNIS
000100	SPENSER	000003	CINEMA
000100	SPENSER	000003	COURS DE LANGUE
000170	YOSHIMURA	000002	TENNIS
000170	YOSHIMURA	000002	CINEMA
000170	YOSHIMURA	000002	COURS DE LANGUE
000180	SCOUTTEN	000002	TENNIS
000180	SCOUTTEN	000002	CINEMA

000180	SCOUTTEN	000002	COURS DE LANGUE
000190	WALKER	000002	TENNIS
000190	WALKER	000002	CINEMA
000190	WALKER	000002	COURS DE LANGUE
000250	SMITH	000004	TENNIS
000250	SMITH	000004	CINEMA
000250	SMITH	000004	COURS DE LANGUE
000280	SCHNEIDER	000005	TENNIS
000280	SCHNEIDER	000005	CINEMA
000280	SCHNEIDER	000005	COURS DE LANGUE
000300	SMITH	000005	TENNIS
000300	SMITH	000005	CINEMA
000300	SMITH	000005	COURS DE LANGUE
000310	SETRIGHT	000005	TENNIS
000310	SETRIGHT	000005	CINEMA
000310	SETRIGHT	000005	COURS DE LANGUE

INNER JOIN

Le **INNER JOIN** permet de faire une jointure entre deux tables et de ramener les enregistrements de la première table qui ont une correspondance dans la seconde table.

Par exemple, ramener le numéro d'employé, le nom d'employé et le numéro du projet dont les employés sont responsables.

Les deux requêtes suivantes produisent le même résultat.

```
SELECT EMPNO, LASTNAME, PROJNO
 FROM EMPLOYE INNER JOIN PROJET
 ON EMPNO = RESPEMP

 SELECT EMPNO, LASTNAME, PROJNO
```

```
FROM EMPLOYE, PROJET
WHERE EMPNO = RESPEMP
```

Résultat

EMPNO	LASTNAME	PROJNO
000020	THOMPSON	PL2100
000060	STERN	MA2110
000100	SPENSER	OP2010
000250	SMITH	AD3112

LEFT OUTER JOIN

Le **LEFT OUTER JOIN** entre deux tables retourne tous les enregistrements ramenés par un **INNER JOIN** plus chaque enregistrement de la table 1 qui n'a pas de correspondance dans la table 2.

Le résultat de la requête suivante ramène tous les employés, même ceux qui n'ont pas de numéro de projet. Dans ce cas, une valeur nulle est retournée à la place du numéro de projet.

```
SELECT EMPNO, LASTNAME, PROJNO
 FROM EMPLOYE LEFT OUTER JOIN PROJET
 ON EMPNO = RESPEMP
```

EMPNO	LASTNAME	PROJNO
000020	THOMPSON	PL2100
000060	STERN	MA2110
000100	SPENSER	OP2010
000170	YOSHIMURA	-
000180	SCOUTTEN	-
000190	WALKER	-
000250	SMITH	AD3112
000280	SCHNEIDER	-
000300	SMITH	-
000310	SETRIGHT	-

RIGHT OUTER JOIN

Le **RIGHT OUTER JOIN** entre deux tables retourne tous les enregistrements ramenés par un **INNER JOIN** plus chaque enregistrement de la table 2 qui n'a pas de correspondance dans la table 1.

Le résultat de la requête suivante ramène tous les employés, même ceux qui n'ont pas de numéro de projet. Dans ce cas, une valeur nulle est retournée à la place du numéro de projet.

Le résultat est le même que celle du LEFT OUTER JOIN.

```
SELECT EMPNO, LASTNAME, PROJNO
FROM PROJET RIGHT OUTER JOIN EMPLOYE
ON EMPNO = RESPEMP
```

EMPNO	LASTNAME	PROJNO
000020	THOMPSON	PL2100
000060	STERN	MA2110
000100	SPENSER	OP2010
000170	YOSHIMURA	-
000180	SCOUTTEN	-
000190	WALKER	-
000250	SMITH	AD3112
000280	SCHNEIDER	-
000300	SMITH	-
000310	SETRIGHT	-

EXCEPTION JOIN

Un **EXCEPTION JOIN** retourne uniquement les enregistrements de la table 1 qui n'on pas de correspondance dans la table 2.

En utilisant le même exemple, la requête suivante ramène uniquement la liste des employés qui ne sont responsable d'aucun projet.

```
SELECT EMPNO, LASTNAME, PROJNO
FROM EMPLOYE EXCEPTION JOIN PROJET
ON EMPNO = RESPEMP
```

EMPNO	LASTNAME	PROJNO

000170	YOSHIMURA	-
000180	SCOUTTEN	-
000190	WALKER	-
000280	SCHNEIDER	-
000300	SMITH	-
000310	SETRIGHT	-

On peut écrire la même requête en utilisant le prédicat NOT EXISTS comme dans l'exemple suivant. La seule différence est que cette requête ne peut retourner de valeur de la table PROJET.

```
SELECT EMPNO, LASTNAME
FROM EMPLOYE
WHERE NOT EXISTS
(SELECT * FROM PROJET
WHERE EMPNO = RESPEMP)
```

MULTIPLES JOINS DANS UNE SEULE REQUETE

Il est bien sur possible de faire plusieurs jointures dans une même requête comme l'indique l'exemple suivant.

```
SELECT EMPNO, LASTNAME, DEPTNAME, PROJNO
FROM EMPLOYE INNER JOIN DEPARTEMENT
ON WORKDEPT = DEPTNO
LEFT OUTER JOIN PROJET
ON EMPNO = RESPEI
```

EMPNO	LASTNAME	DEPTNAME	PROJNO
000020	THOMPSON	PLANNING	PL2100
000060	STERN	MANUFACTURING SYSTEMS	MA2110
000100	SPENSER	SOFTWARE SUPPORT	OP2010
000170	YOSHIMURA	MANUFACTURING SYSTEMS	-
000180	SCOUTTEN	MANUFACTURING SYSTEMS	-
000190	WALKER	MANUFACTURING SYSTEMS	-
000250	SMITH	ADMINISTRATION SYSTEMS	AD3112
000280	SCHNEIDER	OPERATIONS	-
000300	SMITH	OPERATIONS	-
000310	SETRIGHT	OPERATIONS	-

Exemple concret

- CSTEQH est le code société.
- CAGCQH est le code agence.
- La requête suivante sélectionne tous les couples société/agence du fichier STKSFMW et les valeurs de stock correspondantes dans STKSFM.
- Si les valeurs ne correspondent pas dans STKSFM pour un couple société/agence, elles sont à valeur nulle.
- Les enregistrements sélectionnés sont insérés dans le fichier STKSFMW1.
- Le code de la requête SQL est extrait d'un source ADELIA.

```
DEBUT_SQL
 +  INSERT
 +  INTO QTEMP/STKSFMW1
 + (CSTEQH, CAGCQH, MOECQH, ACALQH, CARTQH, QTPHQH, CDUPQH,
 +  CDUFQH, UPUFQH, PRRFQH, DISPQH, QDISQH, JRUPQH, CPROQH)
 +  SELECT
 +  A.CSTEQH, A.CAGCQH, B.MOECQH, B.ACALQH,
 +  B.CARTQH, B.QTPHQH, B.CDUPQH, B.CDUFQH,
 +  B.UPUFQH, B.PRRFQH, B.DISPQH, B.QDISQH,
 +  B.JRUPQH, B.CPROQH
 +  FROM
 +  QTEMP/STKSFMW  AS A
 +  LEFT OUTER JOIN
 +  STKSFM  AS B
 +  ON  A.CSTEQH = B.CSTEQH AND
 +  A.CARTQH = B.CARTQH AND A.CAGCQH = B.CAGCQH
 +  AND (B.ACALQH * 100 + B.MOECQH ) = A.MAXDAT
 +  WHERE B.QTPHQH <> 0 OR B.QDISQH <> 0
 FIN_SQL
```

---

[1] L'exemple est extrait de la documentation en anglais d'IBM.