



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article209>

# Aide mémoire SQL sur les instructions DELETE et UPDATE.

- Les articles -



Date de mise en ligne : jeudi 18 mai 2006

Date de parution : 27 octobre 2004

## **Description :**

Un aide mémoire SQL sur les instructions DELETE et UPDATE, sous la forme de quelques exemples simplifiés.

---

**Environnement iSeries**

---

Cet article constitue un aide mémoire SQL sur les instructions DELETE et UPDATE, sous la forme de quelques exemples simplifiés (parfois on ne se rappelle plus de la syntaxe).

## Exemple utilisé

Considérons les cinq tables EMPLOYE, PROJET, DEPARTEMENT, SALAIRES 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.
- La table SALAIRES contient le montant du salaire de chacun des employés.

### Table des employés

EMPLOYE,

EMPNO, numéro de l'employé

LASTNAME, nom de l'employé

WORKDEPT, département ou travailleur

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 salaires

SALAIRES,

EMPNO, numéro de l'employé

MONTANT, Montant du salaire

EMPNO	MONTANT
000020	45000
000060	42000
000100	41000
000170	35000
000180	27500
000190	56823
000250	75248
000280	36150
000300	41563
000310	51234

## 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

## Quelque cas de mise à jour d'enregistrements par SQL.

**Exemple 1 :** Modifier le département associé à l'employé numéro '000060'. L'employé '000060' appartient maintenant au service OPERATIONS.

```
UPDATE EMPLOYE  
SET WORKDEPT = '000005'  
WHERE EMPNO = '000060'
```

**Exemple 2 :** Tous les employés sont ré-affectés au département OPERATIONS, excepté ceux qui font partie du département SOFTWARE SUPPORT.

```
UPDATE EMPLOYE  
SET WORKDEPT = '000005'  
WHERE WORKDEPT <> '000003'
```

**Exemple 3 :** Le salaire de tous les employés du service SOFTWARE SUPPORT est augmenté d'un montant de 1000.

```
UPDATE SALAIRES A  
SET A.MONTANT = A.MONTANT + 1000  
WHERE EXISTS(  
SELECT * FROM EMPLOYE B  
WHERE B.WORKDEPT = '000003'  
AND B.EMPNO = A.EMPNO  
)
```

## Quelque cas de suppression d'enregistrements par SQL.

Supprimer tous les enregistrements de la table t1

```
DELETE FROM T1
```

**Supprimer tous les enregistrements de la table t1 ayant une correspondance dans la table t2, C1 étant la zone identifiant la correspondance.**

```
DELETE FROM T1 A
WHERE EXISTS
(
SELECT * FROM T2 B
WHERE B.C1 = A.C1
)
```

**Supprimer tous les enregistrements de la table T1 sans correspondance dans la table T2, C1 étant la zone identifiant la correspondance.**

```
DELETE FROM T1 A
WHERE NOT EXISTS
(
SELECT * FROM T2 B
WHERE B.C1 = A.C1
)
```

**Supprimer les doublons de la table T1 sur la valeur de la colonne COLX**

*1ère solution :*

```
DELETE FROM T1 A
WHERE RRN(A) NOT IN (
SELECT MAX( RRN(B) ) FROM T1 B WHERE A.COLX = B.COLX
)
```

*2nde solution : plus performante, mais à tester plus avant car ne semble plus fiable lors de traitement batch sur gros volumes.*

```
DELETE FROM T1 X WHERE RRN(X) IN (
SELECT RRN(A) FROM T1 A
INNER JOIN (
SELECT MAX(RRN(T1)) AS SEQ,
COLX, COLY, COLZ
FROM T1
GROUP
BY COLX, COLY, COLZ
HAVING COUNT(*) > 1 ) B
ON A.COLX = B.COLX AND
A.COLY = B.COLY AND
A.COLZ = B.COLZ AND
RRN(A) < SEQ )
```

**Spécificité du delete \* à partir de la V5R3**

A partir de la V5R3, un delete \* from table se comporte comme un CLRPFM. Il se passe la même chose : suppression de tous les enregistrements, mais avec en plus, un RGZPFM.

## Utilisation de requêtes dynamique

➤ Abordé dans cet [article](#).

Les cas exposés sont des requêtes de suppression simples. Vous pouvez évidemment les compliquer selon vos besoins fonctionnels.

## Cas d'un curseur en mise à jour

**Par un curseur en mise à jour, pour un code promotion, mettre à jour les article dans la table PROMO par les informations articles d'une autre promotion de la même table PROMO.**

**Objectif :** Mettre à jour les prix prix1 et prix2 de la promotion 1 avec les prix de la promotion 2, pour les articles communs dans les deux promotions.

### Structure de la table utilisée :

➤ PROMO : table des promotions  
&mdash; PROMO = code promotion  
&mdash; ARTICLE = code article  
&mdash; PRIX1 = Prix 1 de l'article pour la promotion  
&mdash; PRIX2 = Prix 2 de l'article pour la promotion

### Requête :

```
*
* Sélection des enregistrements ( zones articles, prix1, prix2 ) de la promo 1 à mettre à jour.
* par un curseur en mise à jour
*
EXEC SQL
+ DECLARE C1 CURSOR FOR
+ SELECT ARTICLE ,
+         PRIX1 ,
+         PRIX2
+ FROM PROMO A
+ WHERE A.PROMO = 1
+ FOR UPDATE OF
+         PRIX1,
+         PRIX2
END-EXEC

*
* Ouverture du curseur
*
EXEC SQL
+ OPEN C1
END-EXEC
```

```
*
* Tant que le SQLCODE est à 0, on effectue la boucle (pas d'erreur / fin de lecture)
*
DOW SQLCOD = 0
*
* Lecture de l'enregistrement suivant du curseur
* Les données sont mises en variables hôtes
*
EXEC SQL
+ FETCH NEXT FROM C1
+ INTO
+   :WARTICLE ,
+   :WPRIX1   ,
+   :WPRIX2
END-EXEC

*
* Si la lecture du curseur a aboutie,
* Mise à jour des données avec les valeurs de la promo 2
* pour le code article traité
*
IF SQLCOD = 0
EXEC SQL
+ UPDATE PROMO
+   SET
( PRIX1 , PRIX2 )
+   =
(
SELECT PRIX1 , PRIX2
+   FROM PROMO
+   WHERE CODE = 2
+   AND ARTICLE = :WARTICLE
+ )
+ WHERE CURRENT OF C1
END-EXEC
ENDIF
ENDDO

*
* Fermeture du curseur
*
EXEC SQL
+ CLOSE C1
END-EXEC
```

## Exemples d'update

**Comment inverser le contenu de deux zones :**

```
UPDATE FICHER A
SET A.ZONE1 = A.ZONE2 , A.ZONE2 = A.ZONE1
```

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