Les jointures

1. Les Jointures ou comment faire des requêtes sur plusieurs tables

Les jointures permettent d'exploiter pleinement le modèle relationnel des tables d'une base de données.
Elle sont faites pour mettre en relation deux (ou plus) tables concourant à rechercher la réponse à des interrogations. Une jointure permet donc de combiner les colonnes de plusieurs tables.

Il existe en fait différentes natures de jointures que nous expliciterons plus en détail. Retenez cependant que la plupart des jointures entre tables s'effectuent en imposant l'égalité des valeurs d'une colonne d'une table à une colonne d'une autre table. On parle alors de jointure naturelle ou équi-jointure. Mais on trouve aussi des jointures d'une table sur elle même. On parle alors d'auto-jointure. Enfin il arrive que l'on doive procéder à des jointures externe, c'est à dire joindre une table à une autre, même si la valeur de liaison est absente dans une table ou l'autre. Enfin, dans quelques cas, on peut procéder à des jointures hétérogènes, c'est à dire que l'on remplace le critère d'égalité par un critère d'inégalité ou de différence. Nous verrons au moins un cas de cette espèce.

Une jointure entre tables peut être mise en oeuvre, soit à l'aide des éléments de syntaxe SQL que nous avons déjà vu, soit à l'aide d'une clause spécifique du SQL, la clause **JOIN**. Nous allons commencer par voir comment à l'aide du SQL de base nous pouvons exprimer une jointure.

1.1. Premiers essais de jointure

Rappel de la syntaxe du SELECT :

SELECT [DISTINCT ou ALL] * ou liste de colonnes FROM nom des tables ou des vues
C'est ici le pluriel de la partie FROM qui change tout...

Tâchons donc de récupérer les n° des téléphones associés aux clients.

Exemple 1

```
SELECT CLI_NOM, TEL_NUMERO
FROM T_CLIENT, T_TELEPHONE
```

```
CLI_NOM  TEL_NUMERO
-------  --------------
DUPONT   01-45-42-56-63
DUPONT   01-44-28-52-52
DUPONT   01-44-28-52-50
DUPONT   06-11-86-78-89
DUPONT   02-41-58-89-52
DUPONT   01-51-58-52-50
DUPONT   01-54-11-43-21
DUPONT   06-55-41-42-95
DUPONT   01-48-98-92-21
DUPONT   01-44-22-56-21
...
```

Cette requête ne possède pas de critère de jointure entre une table et l'autre. Dans ce cas, le compilateur SQL calcule le produit cartésien des deux ensembles, c'est à dire qu'à chaque ligne de la première table, il accole l'ensemble des lignes de la seconde à la manière d'une "multiplication des petits pains" ! Nous verrons qu'il existe une autre manière, normalisée cette fois, de générer ce produit cartésien. Mais cette requête est à proscrire. Dans notre exemple elle génère 17 400 lignes...

Il faut donc définir absolument un critère de jointure.

Dans le cas présent, ce critère est la correspondance entre les colonnes contenant la référence de l'identifiant du client (CLI_ID).

Exemple 2

```
SELECT CLI_NOM, TEL_NUMERO
FROM T_CLIENT, T_TELEPHONE
WHERE CLI_ID = CLI_ID
```

```
CLI_NOM  TEL_NUMERO
-------  --------------
DUPONT   01-45-42-56-63
DUPONT   01-44-28-52-52
DUPONT   01-44-28-52-50
DUPONT   06-11-86-78-89
```

```
DUPONT   02-41-58-89-52
DUPONT   01-51-58-52-50
DUPONT   01-54-11-43-21
DUPONT   06-55-41-42-95
DUPONT   01-48-98-92-21
DUPONT   01-44-22-56-21
...
```

Nous n'avons pas fait mieux, car nous avons créé une clause toujours vrai, un peu à la manièreb de 1 = 1 !
En fait il nous manque une précision : il s'agit de déterminer de quelles tables proviennent les colonnes CLI_ID de droite et de gauche. Cela se précise à l'aide d'une notation pointée en donnant le nom de la table.

Il est donc nécessaire d'indiquer au compilateur la provenance de chacune des colonnes CLI_ID et donc d'opérer une distinction entre l'une et l'autre colonne.
Ainsi, chaque colonne devra être précédée du nom de la table, suivi d'un point.

Exemple 3

```
SELECT CLI_NOM, TEL_NUMERO
FROM T_CLIENT, T_TELEPHONE
WHERE T_CLIENT.CLI_ID = T_TELEPHONE.CLI_ID
```

```
CLI_NOM   TEL_NUMERO
-------   --------------
DUPONT    01-45-42-56-63
DUPONT    01-44-28-52-52
DUPONT    01-44-28-52-50
BOUVIER   06-11-86-78-89
DUBOIS    02-41-58-89-52
DREYFUS   01-51-58-52-50
DUHAMEL   01-54-11-43-21
BOYER     06-55-41-42-95
MARTIN    01-48-98-92-21
MARTIN    01-44-22-56-21
...
```

On tombe ici à 174 enregistrements dans la table !!!

Mais il existe une autre façon de faire, plus simple encore. On utilise la technique du "**surnommage**", c'est à dire que l'on attribue un surnom à chacune des tables présente dans la partie FROM du SELECT :

Exemple 4

```
SELECT CLI_NOM, TEL_NUMERO
FROM T_CLIENT C, T_TELEPHONE T
WHERE C.CLI_ID = T.CLI_ID
```

```
CLI_NOM   TEL_NUMERO
-------   --------------
DUPONT    01-45-42-56-63
DUPONT    01-44-28-52-52
DUPONT    01-44-28-52-50
BOUVIER   06-11-86-78-89
DUBOIS    02-41-58-89-52
DREYFUS   01-51-58-52-50
DUHAMEL   01-54-11-43-21
BOYER     06-55-41-42-95
MARTIN    01-48-98-92-21
MARTIN    01-44-22-56-21
...
```

Ici, la table T_CLIENT a été surnommée "C" et la table T_TELEPHONE "T".

Bien entendu, et comme dans les requêtes monotabulaires on peut poser des conditions supplémentaires de filtrage dans la clause where. Cherchons par exemple les clients dont les numéros de téléphone correspondent à un fax :

Exemple 5

```
SELECT CLI_NOM, TEL_NUMERO
FROM T_CLIENT C, T_TELEPHONE T
WHERE C.CLI_ID = T.CLI_ID
  AND TYP_CODE = 'FAX'
```

```
CLI_NOM   TEL_NUMERO
-------   --------------
DUPONT    01-44-28-52-50
MARTIN    01-44-22-56-21
DUHAMEL   01-54-11-43-89
DUPONT    05-59-45-72-42
MARTIN    01-47-66-29-55
```

```
DUBOIS    04-66-62-95-64
DREYFUS   04-92-19-18-58
DUHAMEL   01-55-60-93-81
PHILIPPE  01-48-44-86-19
DAUMIER   01-48-28-17-95
...
```

Le fait de placer comme critère de jointure entre les tables, l'opérateur logique "égal" donne ce que l'on apelle une "**équi-jointure**".

REMARQUE IMPORTANTE
Comme vous pouvez le constater, le nom du client BOUVIER n'apparaît pas. Il n'a pas été "oublié" par le traitement de la requête, mais le numéro de fax de ce client n'est pas présent dans la table T_TELEPHONE. Or le moteur SQL recherche les valeurs de la jointure par égalité. Comme l'ID_CLI de BOUVIER n'est pas présent dans la table T_TELEPHONE, il ne peut effectuer la jointure et ignore donc la ligne concernant le client BOUVIER. Nous verrons comment réparer cette lacune lorsque nous parlerons des jointures externes.

NOTA : on peut aussi utiliser les surnoms dans la partie qui suit immédiatement le mot clef SELECT. Ainsi l'exemple 4, peut aussi s'écrire :

Exemple 6

```
SELECT C.CLI_ID, C.CLI_NOM, T.TEL_NUMERO
FROM T_CLIENT C, T_TELEPHONE T
WHERE C.CLI_ID = T.CLI_ID
  AND T.TYP_CODE = 'FAX'
```

```
CLI_ID  CLI_NOM   TEL_NUMERO
------ -------   --------------
     1  DUPONT    01-44-28-52-50
    10  MARTIN    01-44-22-56-21
     8  DUHAMEL   01-54-11-43-89
     1  DUPONT    05-59-45-72-42
     2  MARTIN    01-47-66-29-55
     4  DUBOIS    04-66-62-95-64
     5  DREYFUS   04-92-19-18-58
     8  DUHAMEL   01-55-60-93-81
    13  PHILIPPE  01-48-44-86-19
    15  DAUMIER   01-48-28-17-95
...
```

C'est particulièrement pratique lorsque l'on veut récupérer une colonne qui se retrouve dans les deux tables, ce qui est souvent le cas de la (ou les) colonne de clef étrangère qui permet juutement d'assurer la jointure.

Pour joindre plusieurs tables, on peut utiliser le même processus de manière répétitive...

Exemple 7

```
SELECT C.CLI_ID, C.CLI_NOM, T.TEL_NUMERO, E.EML_ADRESSE, A.ADR_VILLE
FROM T_CLIENT C, T_TELEPHONE T, T_ADRESSE A, T_EMAIL E
WHERE C.CLI_ID = T.CLI_ID
  AND C.CLI_ID = A.CLI_ID
  AND C.CLI_ID = E.CLI_ID
```

```
CLI_ID   CLI_NOM  TEL_NUMERO      EML_ADRESSE                    ADR_VILLE
------ -------- -------------- ----------------------          ---------
     1  DUPONT    01-45-42-56-63  alain.dupont@wanadoo.fr          VERSAILLES
     1  DUPONT    05-59-45-72-42  alain.dupont@wanadoo.fr          VERSAILLES
     1  DUPONT    05-59-45-72-24  alain.dupont@wanadoo.fr          VERSAILLES
     1  DUPONT    01-44-28-52-50  alain.dupont@wanadoo.fr          VERSAILLES
     1  DUPONT    01-44-28-52-52  alain.dupont@wanadoo.fr          VERSAILLES
     2  MARTIN    01-47-66-29-29  mmartin@transports_martin_fils.fr  VERGNOLLES CEDEX 452
     2  MARTIN    01-47-66-29-55  mmartin@transports_martin_fils.fr  VERGNOLLES CEDEX 452
     2  MARTIN    01-47-66-29-29  plongeur@aol.com                VERGNOLLES CEDEX 452
     2  MARTIN    01-47-66-29-55  plongeur@aol.com                VERGNOLLES CEDEX 452
     5  DREYFUS   04-92-19-18-58  pdreyfus@club-internet.fr         PARIS
...
```

MAIS ATTENTION ! De même que nous l'avons vu dans l'exemple 2.4, ne sont visible ici que les lignes clients ayant A LA FOIS, au moins une adresse, un e mail et au moins un numéro de téléphone... Si nous avions voulu une liste complète des clients avec toutes les coordonnées disponibles, nous aurions du faire une requête externe sur les tables...