

SQL - Sous Ensemble LMD Les opérations de  
jointure

ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail



**DIRECTION RECHERCHE ET INGENIERIE DE FORMATION  
SECTEUR NTIC**

## Sommaire

<b>1. Introduction.....</b>	<b>2</b>
<b>2. Requêtes intégrant plusieurs tables.....</b>	<b>3</b>
<b>2.1. La méthode ensembliste.....</b>	<b>3</b>
<b>2.2. La méthode prédicative.....</b>	<b>9</b>

# 1.Introduction

Ce support a pour but de vous présenter les différents types de jointure, les opérations de jointure et d'une manière générale, le travail avec des données issues de plusieurs tables.

Seront traités dans ce document :

- Les opérateurs ensemblistes
- La méthode des prédicats
- Les différentes natures de jointure (equi-jointure, jointure droite, gauche et totale)

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	2 - 17

## 2.Requêtes intégrant plusieurs tables

Nous allons aborder ici la mise en œuvre de requêtes portant sur plusieurs tables. Deux méthodes sont à notre disposition, la méthode ensembliste et la méthode prédicative. Ces deux méthodes réalisent des **jointures**.

La **jointure** consiste à rechercher entre deux tables ayant un attribut commun (même type et même domaine de définition) tous les tuples (toutes les lignes) pour lesquels ces attributs ont la même valeur.

- la méthode ensembliste réalise l'intersection de deux ensembles et s'exprime sous forme de requêtes imbriquées.
- la méthode prédicative vérifie l'égalité de deux attributs et s'exprime sous la forme d'une seule sélection conditionnelle.

### 2.1. La méthode ensembliste

Schéma de construction :

```
SELECT liste d'attributs
FROM table1
WHERE attribut de jointure
      IN (SELECT attribut de jointure
          FROM table2
          WHERE condition)
```

La requête à l'intérieur des parenthèses est dite requête interne ou sous-requête. Elle est évaluée en premier, constituant ainsi un premier ensemble dont on réalisera l'intersection (IN) avec l'ensemble issu de l'évaluation de la requête externe.

Les attributs sélectionnés, et retenus dans le jeu de résultat, sont **nécessairement** issus de la requête externe. Il s'agit donc d'une méthode assez restrictive.

D'une manière générale, l'exécution des requêtes construites selon la méthode ensembliste demande plus de ressources au système. Elles peuvent toutefois être plus faciles à réaliser et bien adaptées à certains cas de figure.

Exemple : Liste des pilotes assurant un vol au départ de Paris

```
SELECT PIL# as "Code Pilote",NOM
FROM PILOTE
WHERE PIL# IN (SELECT PILOTE
               FROM VOL
               WHERE Villedepart = 'Paris');
```

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	3 - 17

## SQL - Sous Ensemble LMD Les opérations de jointure

	Code Pilote	NOM
1	1	Serge
2	2	Jean
3	7	Bertrand
4	9	Luc

Figure 1 : Illustration de la méthode ensembliste

Autre exemple : Traitement du sauf (négation NOT) ou différence.  
Liste des pilotes qui ne sont pas affectés à des vols.

```
SELECT PIL# as "Code Pilote",NOM
FROM PILOTE
WHERE PIL# NOT IN (SELECT PILOTE
FROM VOL)
```



L'attribut de jointure est une valeur scalaire et ne peut donc être une valeur vectorielle. Ainsi, vous ne pouvez pas écrire une requête sous la forme qui suit :

```
SELECT PIL# as "Code Pilote",NOM
FROM PILOTE
WHERE (PIL#,VILLE) NOT IN (SELECT
PILOTE,VILLEDEPART
FROM VOL)
```

On ne pourra pas non plus écrire la requête suivante :

```
SELECT PIL# as "Code Pilote",NOM,VILLEDEPART
FROM PILOTE
WHERE PIL# IN (SELECT PILOTE FROM VOL)
```

Car l'attribut VILLEDEPART n'appartient pas à la table sur laquelle porte la requête externe. Vous obtenez un message d'erreur :

Serveur : Msg 207, Niveau 16, État 3, Ligne 1  
'VILLEDEPART' : nom de colonne incorrect.

On peut par contre joindre un attribut avec un attribut qui résulte d'une opération récapitulative.

Nous pouvons ainsi obtenir la liste des avions dont la capacité en passagers est égale à la capacité maximum.

```
SELECT AV#,MARQUE,TYPEAVION,CAPACITE
FROM AVION
WHERE CAPACITE IN
```

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	4 - 17

SQL - Sous Ensemble LMD Les opérations de jointure  
(SELECT MAX(CAPACITE) FROM AVION)

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	5 - 17

	AV#	MARQUE	TYPEAVION	CAPACITE
1	155	AIRBUS	A340	600
2	160	AIRBUS	A340	600
3	170	AIRBUS	A340	600

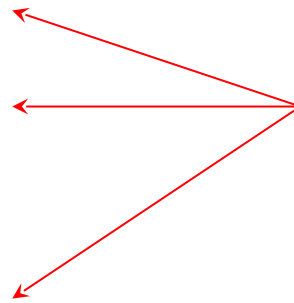
Figure 2 : Liste des avions dont la capacité est égale à la capacité maximum

Si nous souhaitons préciser qu'il s'agit en fait des avions qui ont la capacité maximum du type, nous modifierons instinctivement la requête comme suit, afin de calculer le maximum selon le type :

```
SELECT AV#,MARQUE,TYPEAVION,CAPACITE
FROM AVION
WHERE CAPACITE IN
  (SELECT MAX(CAPACITE) FROM AVION GROUP BY
    MARQUE,TYPEAVION)
```

Ce qui donne pour résultat (extrait) :

	AV#	MARQUE	TYPEAVION	CAPACITE
1	100	AIRBUS	A320	381
2	101	Boeing	B707	250
3	102	AIRBUS	A320	522
4	104	Boeing	B747	400
5	106	Atr	ATR42	500
6	107	Boeing	B727	300
7	108	Boeing	B727	300
8	120	Caravelle	Caravelle	381



Ce qui est faux, car un Airbus A320 se trouve être retenu car sa capacité vaut la capacité maximum des avions de type Caravelle !

En fait, pour obtenir un résultat exact, il nous faut **corréliser** les lignes de la requête externe avec celles de la requête interne. Tous les SGBDR ne sont pas en capacité de réaliser ce type d'opérations. Elles sont à utiliser avec parcimonie dans la mesure du possible car très coûteuses en matière de ressources et donc de temps de traitement.

**En fait, la requête interne se trouve évaluée pour chaque ligne de la requête externe !**

Dans l'exemple suivant, vous remarquerez l'utilisation d'un alias pour distinguer deux occurrences d'un même table.

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	6 - 17



### Requêtes corrélées

```
SELECT AV#,MARQUE,TYPEAVION,CAPACITE
FROM AVION as "AV1"
WHERE CAPACITE IN
      (SELECT MAX(CAPACITE)
       FROM AVION
       WHERE MARQUE = AV1.MARQUE AND TYPEAVION
       = AV1.TYPEAVION GROUP BY MARQUE,TYPEAVION)
```

Le résultat est alors correct et l'avion 100 éliminé du jeu de résultats.

	AV#	MARQUE	TYPEAVION	CAPACITE
1	101	Boeing	B707	250
2	102	AIRBUS	A320	522
3	104	Boeing	B747	400
4	106	Atr	ATR42	500
5	107	Boeing	B727	300
6	108	Boeing	B727	300
7	120	Caravelle	Caravelle	381
8	151	Boeing	B707	250
9	155	AIRBUS	A340	600
10	160	AIRBUS	A340	600
11	170	AIRBUS	A340	600

Figure 3 : Exemple de requête corrélée

### Comparaison d'une valeur avec l'ensemble des valeurs d'un attribut d'une requête interne.

On peut utiliser les opérateurs **SOME**, **ANY**, ou **ALL** pour comparer la valeur d'une expression avec les valeurs d'un attribut d'une requête interne.

Exemple : Liste des avions dont la capacité est supérieure à toute capacité des avions de marque Boeing.

```
SELECT AV#,MARQUE,TYPEAVION,CAPACITE
FROM AVION
WHERE CAPACITE > ALL
      (SELECT CAPACITE FROM AVION WHERE MARQUE='Boeing')
```

La liste de valeurs générée par la requête interne est {250, 300, 400}

## SQL - Sous Ensemble LMD Les opérations de jointure

Le résultat est le suivant :

	AV#	MARQUE	TYPEAVION	CAPACITE
1	102	AIRBUS	A320	522
2	105	AIRBUS	A320	423
3	106	Atr	ATR42	500
4	155	AIRBUS	A340	600
5	160	AIRBUS	A340	600
6	170	AIRBUS	A340	600

Figure 4 : Comparer une valeur à un ensemble avec ALL

En modifiant la requête et en remplaçant ALL par ANY ou SOME, nous obtenons la liste de tous les avions dont la capacité est > 200.

	AV#	MARQUE	TYPEAVION	CAPACITE
1	100	AIRBUS	A320	381
2	102	AIRBUS	A320	522
3	104	Boeing	B747	400
4	105	AIRBUS	A320	423
5	106	Atr	ATR42	500
6	107	Boeing	B727	300
7	108	Boeing	B727	300
8	109	AIRBUS	A340	350
9	120	Caravelle	Caravelle	381
10	150	AIRBUS	A340	345
11	155	AIRBUS	A340	600
12	160	AIRBUS	A340	600
13	170	AIRBUS	A340	600

Figure 5 : Comparer une valeur à un ensemble avec SOME ou ANY

### Utilisation de EXISTS.

EXISTS peut être utilisé pour réaliser des intersections ou des différences entre deux requêtes. Ce n'est pas la meilleure forme d'utilisation, car elle fait appel à des requêtes corrélées et peut s'écrire plus simplement. Je donne donc ici un seul exemple d'utilisation. Nous verrons d'autres exemples de mise en œuvre plus judicieux par la suite dans la partie Programmation du SGBDR.

Liste des pilotes qui habitent la ville de départ d'un vol.

L'évaluation de EXISTS renvoie un booléen porteur de la valeur vrai ou faux.

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	8 - 17

```
SELECT PIL#,NOM
FROM PILOTE
WHERE EXISTS ( SELECT VILLEDEPART FROM VOL WHERE VILLEDEPART
= PILOTE.VILLE)
```

	PIL#	NOM
1	2	Jean
2	3	Roger
3	5	Michel
4	7	Bertrand
5	9	Luc
6	19	Driss
7	29	Laverdure
8	30	Bost

Figure 6 : Illustration de l'opérateur EXISTS

**Remarque** : Cette requête serait mieux écrite ainsi :

```
SELECT PIL#,NOM
FROM PILOTE
WHERE VILLE IN ( SELECT VILLEDEPART FROM VOL)
```

### Utilisation de EXCEPT et INTERSECT.

EXCEPT et INTERSECT sont de nouveaux opérateurs permettant de trouver des enregistrements communs à deux jeux de données ou de retrouver des enregistrement figurant dans un jeu de données et pas dans l'autre. Ces opérateurs obéissent aux mêmes règles que UNION, les jeux d'enregistrement doivent être de structure identique (même nombre de colonnes, mêmes types voire même longueur).

## 2.2. La méthode prédicative

La requête comporte une seule instruction SELECT qui traite plusieurs tables dont la liste apparaît dans la clause FROM.

La traduction de la jointure se fait par une équation de jointure (égalité entre 2 attributs) exprimée au niveau de la clause FROM.

Il existe quatre natures de jointure qui sont respectivement exprimées par les mots clés INNER, RIGHT OUTER, LEFT OUTER ou FULL OUTER dont nous verrons les différents cas d'usage dans les exemples suivants.

SQL - Sous Ensemble LMD Les opérations de jointure

**Important** : La méthode prédicative permet de rapporter dans le jeu de résultats les valeurs des attributs des différentes tables définies dans la clause FROM.

Cette méthode est en général préférable à la méthode ensembliste car plus performante.

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	10 - 17

**Premier exemple de jointure interne (INNER JOIN)**

Liste des avions affectés à des vols avec la mention de la ville de départ du vol .

Nous avons besoin d'utiliser les tables VOL et AVION et de déterminer l'attribut commun aux deux tables sur lequel portera la jointure.

Ici le code de l'avion, nommé AV# dans AVION et AVION dans VOL.

```
SELECT AV#,MARQUE,TYPEAVION,VILLEDEPART
FROM AVION INNER JOIN VOL
ON AVION.AV# = VOL.AVION
```

La clause INNER JOIN, jointure interne, ne retient que les lignes des deux tables pour lesquelles l'expression exprimée au niveau de ON se vérifie.

	AV#	MARQUE	TYPEAVION	VILLEDEPART
1	100	AIRBUS	A320	Nice
2	100	AIRBUS	A320	Paris
3	101	Boeing	B707	Paris
4	105	AIRBUS	A320	Grenoble
5	105	AIRBUS	A320	Toulouse
6	107	Boeing	B727	Lyon
7	109	AIRBUS	A340	Bastia
8	106	Atr	ATR42	Paris
9	106	Atr	ATR42	Brive

Figure 7 : Expression de la jointure INNER JOIN

Si nous voulons la liste de tous les avions avec, pour ceux qui volent, des informations sur les villes de départ, nous écrirons :

```
SELECT AV#,MARQUE,TYPEAVION,VILLEDEPART
FROM AVION LEFT OUTER JOIN VOL
ON AVION.AV# = VOL.AVION
```

Pour les avions qui ne sont pas en service, l'attribut VILLEDEPART aura la valeur **NULL**

Le résultat nous prouve qu'une requête de type LEFT OUTER renvoie la liste des lignes pour lesquelles la jointure avec égalité est respectée, complétée par les lignes de la table de gauche sans correspondance dans la table de droite.

Résultat d'une jointure externe gauche

	AV#	MARQUE	TYPEAVION	VILLEDEPART
1	100	AIRBUS	A320	Nice
2	100	AIRBUS	A320	Paris
3	101	Boeing	B707	Paris
4	101	Boeing	B707	Nice
5	102	AIRBUS	A320	Toulouse
6	103	Caravelle	Caravelle	NULL

Figure 8 : Illustration de LEFT OUTER JOIN

### Compléments sur la définition des jointures

#### INNER

Spécifie toutes les paires correspondantes de lignes renvoyées. Supprime les lignes n'ayant pas de correspondance entre les deux tables. Ceci est l'option par défaut si aucun type de jointure n'est spécifié.

#### FULL [OUTER] : Peu usitée

Précise qu'une ligne de la table de gauche ou de droite, qui ne correspond pas à la condition de jointures, est comprise dans l'ensemble de résultats et que les colonnes de sortie sans correspondance dans l'autre table ont des valeurs nulles. Ceci est fourni **en plus** de toutes les lignes renvoyées par INNER JOIN.

#### LEFT [OUTER]

Spécifie que toutes les lignes de la table de gauche ne respectant pas la condition de jointure sont comprises dans l'ensemble de résultats, et que les colonnes de sortie de l'autre table sans correspondance ont des valeurs NULL.

#### RIGHT [OUTER] :

Spécifie que toutes les lignes de la table de droite ne respectant pas la condition de jointure sont comprises dans l'ensemble de résultats, et que les colonnes de sortie correspondant à l'autre table ont des valeurs NULL.

#### JOIN

Indique que l'opération de jointure spécifiée doit avoir lieu entre les tables ou vues données.

#### ON <condition de recherche>

Indique la condition sur laquelle se base la jointure

### Auto-jointure

L'auto-jointure est la jointure entre une table et elle-même, pour sélectionner des enregistrements correspondant à d'autres de la même

SQL - Sous Ensemble LMD Les opérations de jointure  
table. Il est nécessaire de recourir alors à des alias pour définir la table déjà  
utilisée.

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	13 - 17

### Exemple d'auto-jointure : Liste des avions de même capacité

```
SELECT AVION.AV#, AVION.MARQUE, AVION.TYPEAVION, AVION.CAPACITE,  
AV2.AV#, AV2.MARQUE, AV2.TYPEAVION, AV2.CAPACITE  
FROM AVION INNER JOIN AVION AS "AV2"  
ON AVION.CAPACITE = AV2.CAPACITE  
WHERE AVION.AV# > AV2.AV#
```

**A noter** : L'expression de la clause WHERE pour éliminer les paires de lignes figurant plusieurs fois.

Notez de plus le préfixe systématique du nom des colonnes (attributs) par le nom de la table afin d'éviter une erreur liée au caractère ambigu du nom. En effet le serveur ne sait pas dans quelle table (ou occurrence d'une table) il doit extraire la valeur d'un attribut si celui-ci existe dans plusieurs tables (occurrences) sous le même nom.

Utilisation du mot clé **DISTINCT**

Les jointures peuvent rapporter des lignes figurant plusieurs fois dans le jeu de résultat. Vous pouvez utiliser **DISTINCT** pour ne conserver qu'un seul exemplaire des lignes rapportées dans le jeu de résultat.

### Exemple de traitement de la différence

Il faut réaliser une jointure gauche et ne retenir que les éléments de la table de gauche (LEFT OUTER) ou de droite (RIGHT OUTER) ayant des valeurs nulles.

Nous introduisons ici la fonction IS NULL permettant d'évaluer un attribut de valeur nulle.

Liste des avions non affectés à des vols

```
SELECT AV#,MARQUE,TYPEAVION  
FROM AVION LEFT OUTER JOIN VOL  
ON AVION.AV# = VOL.AVION  
WHERE VOL.AVION IS NULL
```

### Exemple de requêtes portant sur 3 tables jointes 2 à 2

Liste des vols avec informations sur pilotes et avions

```
SELECT VOL#, VILLEARRIVEE, VILLEDEPART, HEUREDEPART, AV#,MARQUE,  
TYPEAVION, PIL#, NOM  
FROM AVION INNER JOIN VOL  
ON AVION.AV# = VOL.AVION  
INNER JOIN PILOTE  
ON VOL.PILOTE = PILOTE.PIL#
```

<b>OFPPT @</b>	Document	Millésime	Page
	533b86852cc48.doc	janvier 08	14 - 17



## SQL - Sous Ensemble LMD Les opérations de jointure

	VOL#	VILLEARRIVEE	VILLEDEPART	HEUREDEPART	AV#	MARQUE	TYPEAVION	PIL#	NOM
1	100	Paris	Nice	7.20	100	AIRBUS	A320	1	Serge
2	101	Toulouse	Paris	11.20	100	AIRBUS	A320	2	Jean
3	102	Nice	Paris	12.35	101	Boeing	B707	1	Serge
4	103	Toulouse	Grenoble	9.00	105	AIRBUS	A320	3	Roger
5	104	Grenoble	Toulouse	17.00	105	AIRBUS	A320	3	Roger
6	105	Paris	Lyon	6.00	107	Boeing	B727	7	Bertrand
7	106	Paris	Bastia	10.00	109	AIRBUS	A340	8	Hervé
8	107	Brive	Paris	7.00	106	Atr	ATR42	9	Luc
9	108	Paris	Brive	19.00	106	Atr	ATR42	9	Luc
10	109	Lyon	Paris	18.00	107	Boeing	B727	7	Bertrand

Figure 9 : Illustration de jointure de 3 tables 2 à 2

### Exemple de 2 ensembles issus de 2 jointures 2 à 2 non dépendantes

D'une part des pilotes volant sur des airbus (1<sup>er</sup> ensemble)

D'autre part des pilotes habitant dans une ville qui est le point de départ d'un vol.

```
SELECT
VOL.VOL#,VOL.VILLEDEPART,AV#,MARQUE,TYPEAVION,PILOTE.NOM
FROM AVION INNER JOIN VOL
  ON AVION.AV# = VOL.AVION,
  PILOTE INNER JOIN VOL as VOL2
  ON PILOTE.VILLE = VOL2.VILLEDEPART
WHERE AVION.MARQUE = 'Boeing'
AND AVION.LOCALISATION = PILOTE.VILLE
```

Les 2 ensembles de jointures sont séparés par une virgule.

Si une table est utilisée plusieurs fois, il convient de lui donner un alias.