

**TP BD N°08 (semaine 43)**  
**Langage SQL – Interrogation des données**

*Créer un fichier **tp08.sql** pour y stocker les requêtes de ce TP.*

## Projections – Sélections – Jointures (internes et externes) – Auto-jointures

**R17 :** Liste de tous les menus qui ne sont ni de type express (typeMenu 1) ni de type économique (typeMenu 2) avec leurs contrats associés. Les menus associés à aucun contrat doivent apparaître.

Afficher :                      **Menu**                      **Type Menu**                      **Contrat**

Où                      Menu : libMenu  
                          Type Menu : { Tradi | Allege | Exotique } selon typeMenu (*fonction DECODE*)  
                          Contrat : « Contrat N° »+NumCtr ou « Pas de Contrat » (*fonction NVL2*)

**Résultat à obtenir :**

Menu	Type Menu	Contrat
-----	-----	-----
MENU MER	ALLEGE	PAS DE CONTRAT
MENU HAWAI	EXOTIQUE	Contrat N°1
MENU PEKIN	EXOTIQUE	PAS DE CONTRAT
MENU SHANGAI	EXOTIQUE	Contrat N°1
MENU BLANQUETTE	TRADI	Contrat N°5
MENU BLANQUETTE	TRADI	Contrat N°2
MENU BOURGOGNE	TRADI	Contrat N°1
MENU BOURGOGNE	TRADI	Contrat N°5
MENU TERROIR	TRADI	Contrat N°2

*Représenter d'abord la requête sous forme d'arbre relationnel puis en langage SQL.*

**R18 :** Liste des menus qui ne sont ni de type express ni de type économique mais uniquement ceux pour lesquels il n'y a aucun contrat.

Afficher :                      **Menu**                      **Type Menu**

Où                      Menu : libMenu  
                          Type Menu : { Tradi | Allege | Exotique } selon typeMenu

**Résultat à obtenir :**

Menu	Type Menu
-----	-----
MENU MER	ALLEGE
MENU PEKIN	EXOTIQUE

*Représenter d'abord la requête sous forme d'arbre relationnel puis en langage SQL.*

**R19 :** Liste des menus de même type que le «MENU HAWAI» (ne pas afficher le menu HAWAI).

Afficher :                      **Menu**



Une jointure (ici, en l'occurrence, une autojointure) entre 2 tables ne fait pas forcément intervenir une clé primaire et une clé étrangère !

**Représenter d'abord la requête sous forme d'arbre relationnel puis en langage SQL.**

**R20 :** Liste des éléments de menu de la même catégorie que l'élément de menu « PAELLA » mais dont le nombre de calories est inférieur.

Afficher :      **Element**      **Nb Calories**      -----      **Calories Paella**

Où      -----      :      Colonne contenant la chaîne de caractère « ----- »  
Calories Paella      :      nb de calories de la PAELLA (non répété)

**Résultat à obtenir :**

Element	Nb Calories	-----	Calories Paella
-----	-----	-----	-----
BROCHETTE DE POISSON	57	-----	126
SAUCISSE STRASBOURG	50	-----	
BLANQUETTE DE VEAU	120	-----	
POISSONS PANES	125	-----	
...			



**Une inéquijointure** fait intervenir tout type d'opérateur (<>, <, >, <=, >=, BETWEEN, LIKE, IN ...).

A l'inverse des **autojointures**, elle n'est pas basée sur l'égalité de 2 colonnes.

**R21 :** Liste des clients ayant un parrain. Tri sur le nom du parrain, puis sur le nom du client.

Afficher :      **Client**      **Adresse C**      **Type C**      **Parrain**      **Adresse P**      **Type P**

Où      Client      : Nom du client  
Adresse C      : Code postal – ville du client (*Concaténation*)  
Type C      : « CE » (si typeClt = 01), « M » (pour 02) ou « CP » (pour 03) selon le type du client  
Parrain , Adresse P et Type P : idem mais pour le client parrain



**L'autojointure** relie une table à elle-même.

Dans une **autojointure**, il est impératif d'utiliser les alias de tables.

**R22 :** Liste de tous les clients même ceux qui n'ont pas de parrain (caractéristiques du parrain éventuellement remplies par des \* (étoiles)).

Afficher :      **Client**      **Adresse C**      **Type C**      **Parrain**      **Adresse P**      **Type P**



**Une autojointure** peut être une autojointure externe.

Attention à orienter l'autojointure correctement !