**Objectifs de ce chapitre**

Dans les entreprises, on est souvent amené à afficher des informations qui sont issues de plusieurs tables. L’objectif de ce cours est de **comprendre comment on met en œuvre l'opérateur de jointure en utilisant les normes SQL V2.**

On profitera de cet exercice pour faire une analyse des informations.

L'entreprise **Belfleurs** est spécialisée dans la mise en place de bouquets de fleurs ou d'arbustes auprès d'une clientèle variée et locale. Une ancienne base de données à été réalisée il y a plusieurs années avec le logiciel ACCESS. L'entreprise souhaite faire évoluer son système d'information. On vous demande de procéder à différentes actions qui conduiront à une refonte du système d'information existant.

1. L'analyse du système existant

L'entreprise **Belfleurs** dispose d'une base de données existante qu'il s'agit de faire évoluer.

Vous avez procédé à une extraction de quelques enregistrements mémorisés dans la base pour comprendre :

* quelles informations étaient mémorisées
* de quelle nature est chaque information (type, longueur)
* quelles valeurs étaient stockées

**Cette extraction est fournie en annexe 1**

**Votre 1er travail : à partir de cette données**

* déduisez le schéma relationnel qui a permis la création de la base
* par retro-conception, déduisez le schéma conceptuel de la base

1. Evolution du système existant

Le bouche à oreille fonctionne bien dans cette petite entreprise qui a le souci d'une mise en place florale toujours adaptée à l'activité et à "l'ambiance" de ses clients. Le fait de ne pas avoir à prospecter la clientèle est un atout majeur et décisif dans la bonne marche économique de l'entreprise. Aussi, quand un nouveau client se présente de la part d'un autre, qui a déjà fait l'objet d'une réalisation florale, l'entreprise **Belfleurs** rétrocède un "cadeau" à l'ancien client qui a parrainé le nouveau, en lui envoyant un bouquet par exemple. Il est donc important de conserver une trace du client qui a parrainé un nouveau client, de même ont souhaitent conserver la trace des parrainages d'un client "actif".

**Votre travail : à partir de ce nouveau besoin**

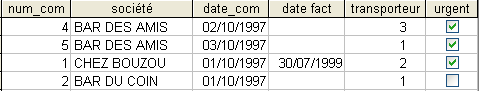
* envisagez une évolution possible de la structure du système d'information
* modifiez en conséquences le schéma conceptuel et le schéma relationnel

1. Les requêtes de jointure à réaliser

On veut afficher chaque commande, numéro et date de commande, date de facture, transporteur et urgence éventuelle, ainsi que la raison sociale de la société cliente.

Où trouvent-on ces informations ? Quelles tables sont concernées ? Quelles champs permettront les jointures ?

Quel résultat devrait-on obtenir ?

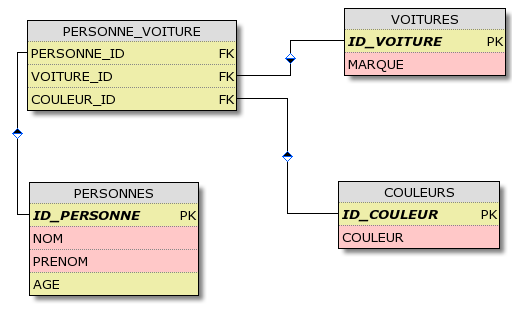


**En utilisant le formalisme de requêtage ci-dessous, écrire la requête qui permettra de répondre à ce besoin.**

**SELECT TABLE1.champ1, TABLE1.Champ2, TABLE2.champ1, TABLE2.champ2**

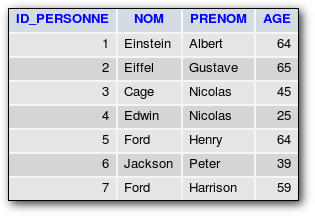
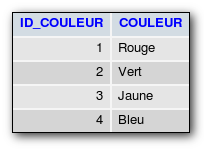
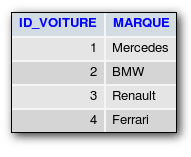
**FROM TABLE-Gauche INNER JOIN TABLE-Droite**

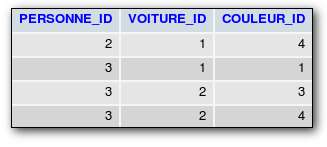
**ON TABLE-Gauche.CléPrimaire = TABLE-Droite.CléEtrangère;**

1. Jointure normalisée SQL V2

Soit le schéma de base de données ci-dessous :

Nous allons partir du fait que ces tables contiennent déjà les données suivantes :

VOITURES COULEURS PERSONNES

PERSONNE\_VOITURE

**4.1 Jointure interne**

On veut afficher l’ensemble des informations de la personne dont l’identifiant est 3 et les données concernant ses voitures.

Comment fonctionne une jointure interne ?

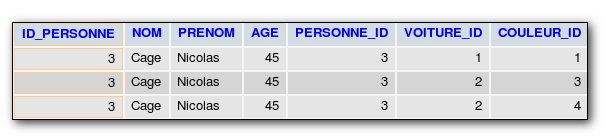
**SELECT** IDPERSONNE, NOM, PRENOM, AGE, PERSONNEID, VOITUREID, COULEURID

**FROM**  **PERSONNES**

**INNER  JOIN**  **PERSONNE\_VOITURE**

**ON**  **IDPERSONNE** = **PERSONNEID**

**WHERE** ID\_PERSONNE = 3

Le résultat renvoyé par cette requête est le suivant : le ON, fonctionne comme le WHERE

Maintenant on souhaite afficher en plus l'intitulé des marques. Il suffit de faire deux autres *INNER JOIN*.

**SELECT** ID\_PERSONNE, NOM, PRENOM, AGE, PERSONNE\_ID, **MARQUE**, COULEUR\_ID

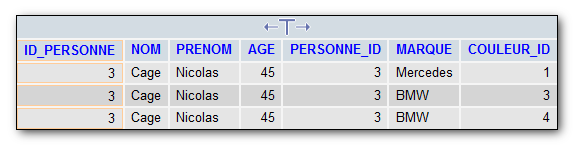
**FROM**  **PERSONNES**

**INNER JOIN**  **PERSONNE\_VOITURE**

**ON**  **IDPERSONNE** = **PERSONNEID**

**INNER JOIN**  **VOITURES**   **ON**  **IDVOITURE** = **VOITUREID**

**WHERE**  **IDPERSONNE** = 3

Le résultat renvoyé par cette requête est le suivant : c'est la marque de la voiture et non plus l'identifiant qui apparaît.

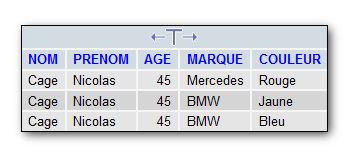
Modifions maintenant une dernière fois la requête pour récupérer uniquement les champs qui nous intéresse mais également pour récupérer l'intitulé de la couleur de la voiture :

**SELECT** NOM, PRENOM, AGE, **MARQUE**, **COULEUR**

**FROM** PERSONNES

**INNER** JOIN PERSONNE\_VOITURE   **ON** IDPERSONNE = PERSONNEID

**INNER** JOIN VOITURES   **ON** IDVOITURE = VOITUREID

**INNER** JOIN COULEURS   **ON** IDCOULEUR = COULEURID

**WHERE** IDPERSONNE = 3

L'exécution de cette requête produirait le résultat suivant :

**4.2 Jointure externe**

Considérez la requête établie précédemment :

**SELECT** NOM, PRENOM, AGE, MARQUE, COULEUR

**FROM** PERSONNES

**INNER** JOIN PERSONNE\_VOITURE   **ON** IDPERSONNE = PERSONNEID

**INNER** JOIN VOITURES   **ON** IDVOITURE = VOITUREID

**INNER** JOIN COULEURS   **ON** IDCOULEUR = COULEURID

**WHERE** IDPERSONNE = 3

Si on essaye d'exécuter cette requête avec l'identifiant 1 🡺 aucun résultat n'est renvoyé.

Or, si la personne 1 ne possède pas de voiture, cette personne existe ! MAIS lorsque les *INNER JOIN* sont évalués, les tables sont jointes sous condition. Si une condition n'est pas respectée, aucune ligne de la table ne peut être jointe aucun résultat n’est renvoyé.

**Il existe trois types de jointures externes. Les gauches, les complètes et les droites.**

Dans le cas des jointures externe gauche, on rajoute les lignes de la table de gauche non prises en compte.

Dans le cas d'une jointure droite, ce sont les lignes de la table de droite non prises en compte qui sont rajoutées.

Les jointures externes complètes ajoutent les lignes de la table qui n'ont pas été prises en compte de gauche et de celle de droite. La syntaxe d'une jointure externe est la suivante :

**LEFT | RIGHT | FULL OUTER JOIN table\_droite   ON condition**

Vous devez choisir entre *LEFT* (gauche), *FULL* (complète) ou *RIGHT* (droite).

Le mot clé *OUTER* n'est pas obligatoire . Nous allons maintenant modifier la requête précédente pour voir un petit peu le fonctionnement des jointures externes. Remplacez donc tous les *INNER* par des *LEFT OUTER* pour donner ceci :

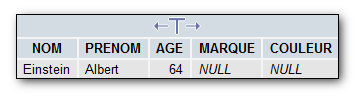
**SELECT** NOM, PRENOM, AGE, MARQUE, COULEUR

**FROM** PERSONNES

LEFT OUTER JOIN PERSONNE\_VOITURE   **ON** IDPERSONNE = PERSONNEID

LEFT OUTER JOIN VOITURES   **ON** IDVOITURE = VOITUREID

LEFT OUTER JOIN COULEURS   **ON** IDCOULEUR = COULEURID

**WHERE** IDPERSONNE = 1

L'exécution de cette requête donnera :

**Base « BelFleurs »**

**Table Clients**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| codeCli | société | contact | adresse | cp | ville | tél | codeCat | parrain |
| 1000 | HOTEL DE VILLE | Alain Terrieur | 5 , impasse du lac | 84000 | APT | 4054545455 | 1 |  |
| 1001 | HOTEL DE VILLE | Jean Caisse | 3, rue d'Ankara | 75010 | PARIS | 0144552211 | 1 | 1000 |
| 1002 | MAJESTIC HOTEL | Li Paoli | 7, allée des prés | 84000 | APT | 0455221111 | 2 |  |
| 1003 | BAR DES AMIS | Alain Terrieur | 1, rue des faucons |  | GAP | 0455666664 | 2 |  |
| 1004 | CHEZ BOUZOU | Claire Voyante | 1, rue de la Paix | 75014 | PARIS | 0155223366 | 4 | 1003 |
| 1005 | BAR DU COIN | Mona Stair | 3, rue de l'avenir | 84000 | APT | 0433336644 | 2 | 1003 |
| 1006 | SOCIETE PEUPLU | SAM SUFFIT | 5 rue de la mare | 05000 | GAP | 0466887744 | 4 | 1005 |
| 1007 | TOUT EN FLEURS | Maguy Thare | 9, rue de Siam | 75016 | PARIS | 0155669988 | 5 | 1002 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Produits**   |  |  |  |  | | --- | --- | --- | --- | | codeProd | désignation | prixht | genre | | 1000 | ACACIA ROSE | 770 | A | | 1001 | ACACIA BOULE | 715 | A | | 1002 | ERABLE SYCOMORE | 430 | A | | 1003 | BOULEAU POURPRE | 370 | A | | 1004 | BOULEAU PLEUREUR | 830 | A | | 1005 | POIRIER FUSEAU | 210 | F | | 1006 | POIRIER PALMETTE | 550 | F | | 1007 | ABRICOTIER TIGE | 650 | P | | 1008 | CAMELIA 60/70 | 550 | P | | 1009 | HORTENSIA | 675 | P | | 1010 | CHEVREFEUILLE | 130 | P | | 1011 | JASMIN 40/60 | 100 | P | | 1012 | JASMIN 120/150 |  | P | | 1013 | RHODODENDRON 100/125 | 1330 | P | | 1014 | MAGNOLIA 150/175 | 1377 | P | | 1015 | MAGNOLIA 175/200 | 1750 | P | | 1016 | BRUYERE 20/25 | 50 | P | | **Table Catégories**   |  |  | | --- | --- | | codeCat | libcat | | 1 | catégorie 1 | | 2 | catégorie 2 | | 3 | catégorie 3 | | 4 | catégorie 4 |   **Table Genres**   |  |  | | --- | --- | | genre | libGenre | | A | Arbre | | F | Fruitier | | G | Fleurs | | P | Plante | |

**Table Commandes**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| numCom | datecom | codeCli | datefact | transporteur | urgent |
| 1 | 01/10/1997 | 1004 | 30/07/1999 | 2 | VRAI |
| 2 | 01/10/1997 | 1005 |  | 1 | FAUX |
| 4 | 02/10/1997 | 1003 |  | 3 | VRAI |
| 5 | 03/10/1997 | 1003 |  | 1 | VRAI |

**Table Détails**

|  |  |  |
| --- | --- | --- |
| numCom | codeProd | quantité |
| 1 | 1003 | 1 |
| 1 | 1009 | 4 |
| 1 | 1014 | 4 |
| 2 | 1001 | 3 |
| 2 | 1008 | 10 |
| 2 | 1009 | 2 |
| 2 | 1011 | 5 |
| 2 | 1015 | 1 |
| 4 | 1009 | 10 |
| 5 | 1009 | 10 |

**Correction**

PRODUITS (codeProd, désignation, prixht, genre)

clé primaire : codeProd

clé étrangère : genre référence genre dans genres

COMMANDES (numCom\_, date\_com, date\_fact, transporteur, urgent, codeCli)

clé primaire : numCom

clé étrangère : codeCli référence codeCli dans clients

DETAILS (numCom, codeProd, quantité)

clé primaire : numCom, codeProd

clés étrangères : numCom référence numCom dans commandes

codeProd référence codeProd dans produits

GENRES (genre, libGenre)

clé primaire : genre

CLIENTS (code\_cli, société, contact, adresse, cp, ville, tél, codeCat#, **codeCliParrain**)

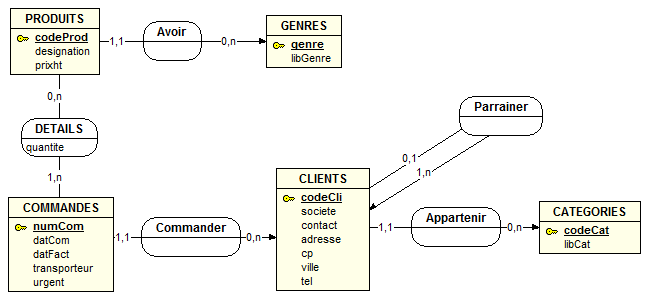
clé primaire : codeCli

clé étrangère : codeCat référence codeCat dans categories

***clé étrangère : codeCliParrain référence codeCli dans codeCli attention à cette nouveauté  !***

CATEGORIES (codeCat, libCat)

clé primaire : code-cat

****