

Objectifs pédagogiques :

- Rechercher des données à partir de deux tables : Jointure ;

Situation problème :

Un magasin de vente en ligne souhaite analyser ses transactions pour mieux comprendre le comportement d'achat de ses clients. La direction a besoin d'un rapport détaillant chaque vente réalisée, avec des informations sur le produit vendu, son prix, ainsi que l'identité du client ayant effectué l'achat.

Nous disposons de la base de données BD_VENTE, composée des tables suivantes :

- CLIENT (IdCli, nom, ville)
- PRODUIT (IdPro, désignation, marque, prix)
- VENTE (IdCli, IdPro, date, qte)

Consigne :

1. Proposer une solution pour extraire les informations nécessaires pour obtenir, pour chaque vente, la **référence du produit**, sa **désignation**, son **prix**, le **numéro du client**, la **date de la transaction** et la **quantité achetée**
2. Écrire une requête SQL permettant d'afficher, pour chaque vente, les informations suivantes :**Référence du produit, Désignation du produit, Prix du produit, Numéro du client, Date de la vente,Quantité vendue**

Resumé

1. Le principe d'une jointure

La **jointure** est une opération qui permet de combiner des lignes de deux tables ou plus en fonction d'une condition commune (généralement une clé primaire et une clé étrangère). **Elle permet de fusionner des données provenant de plusieurs tables en une seule table résultante.**

2. La syntaxe SQL de jointure entre 2 tables

La syntaxe SQL de jointure entre deux tables est la suivante :

Select Nom_table1.nom_colonne1, Nom_table1.nom_colonne1, Nom_table2.nom_colonne1,.....

From Nom_table1, Nom_table2

Where Nom_table1.nom_colonne n= Nom_table2.colonne n ;

NB : les nom_colonne n des tables Nom_table1 et Nom_table2 doivent être identique.

Exemple : Vente (**IdCli**, IdPro, Date,Qte) ; Produit(**IdPro**, Désignation,Prix) ;

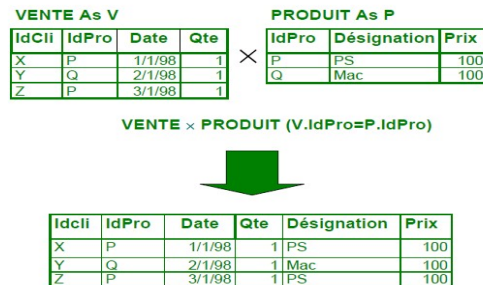
Donnez pour chaque vente la référence du produit, sa désignation, son prix, le numéro de client, la date et la quantité vendue.

Select Vente.IdCli, Vente.IdPro, Vente.Date,

Vente.Qte, Produit.Designation, Produit.Prix

From Vente, Produit

Where Vente.IdPro = Produit.IdPro ;



3. Utilisation des alias pour renommer les tables

Les **alias** sont des noms de remplacement que l'on donne de manière temporaire(temps d'une requête) à une colonne, une table ou une donnée. On peut définir un alias avec le mot clé AS.

Exemple : Donnez pour chaque vente la référence du produit, sa désignation, son prix, le numéro de client, la date et la quantité vendue.

Select V.IdCli, V.IdPro, V.Date, V.Qte, P.Designation, P.Prix

From Vente V, Produit P

Where V.IdPro = P.IdPro ;

En utilisant le mot clé AS

Select V.IdCli AS CodeClient, V.IdPro AS CodeProduit, V.Date AS Date, V.Qte AS Qte, P.Designation

AS Designation, P.Prix AS Prix

From Vente V, Produit P

Where V.IdPro = P.IdPro ;

4. La différence entre la jointure et la division

La **jointure** et la **division** sont deux opérations en **base de données relationnelle**, mais elles ont des usages très différents :

◆ **Jointure :** **Combine** des données de plusieurs **tables** en fonction d'une clé commune.

Exemple : Trouver les noms des clients et leurs commandes en liant la table "Clients" et la table "Commandes".

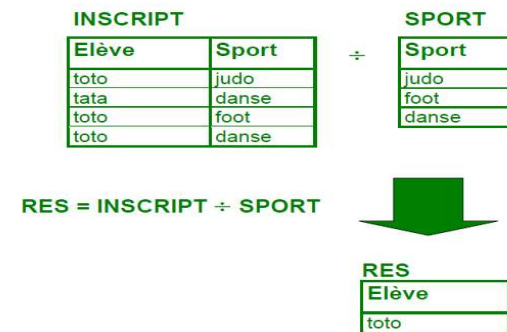
◆ **Division :** **Recherche** des éléments qui sont **liés à un ensemble complet** d'un autre élément.

Exemple : Trouver les étudiants qui ont réussi **toutes** les matières d'un programme.

En résumé :

- **Jointure** consiste à associer deux tables sur une clé commune.
- **Division** consiste à trouver les éléments qui satisfont toutes les conditions d'un autre ensemble.

Exemple de division: Requête: Quels sont les élèves qui sont inscrits à tous les sports?



Exercice d'application :

EXERCICE 1 :

Soit les Tables suivantes :

- **Clients** (id_client, nom)
- **Commandes** (id_commande, id_client, produit)

Ecrire la requête qui permet d'afficher les **noms des clients** avec leurs **commandes**

EXERCICE 3

Soit la base de données suivante :

- *Employés* (id_employé, nom, id_département)
- *Départements* (id_département, nom_département)

On veut afficher les noms des **employés** avec les **noms de leur département**

EXERCICE 4

Soit la base de données suivante :

- **Étudiants** (id_etudiant, nom)
- **Cours** (id_cours, nom_cours)
- **Inscriptions** (id_etudiant, id_cours) (*table intermédiaire pour gérer la relation "plusieurs à plusieurs"*)

On veut afficher la liste des **étudiants** avec les **cours qu'ils suivent**

Exercice 5 : Liste des employés et leurs managers

Tables disponibles :

1. **Employés** (id_employé, nom, id_manager)
2. **Managers** (id_manager, nom_manager)

Écrivez une requête SQL pour afficher la **liste des employés** avec le **nom de leur manager**.

Exercice 6 : Liste des clients et leurs commandes

Tables disponibles :

1. **Clients** (id_client, nom)
2. **Commandes** (id_commande, id_client, produit, date_commande)

Écrivez une requête SQL qui retourne le **nom du client**, le **produit commandé** et la **date de la commande**.

Exercice 7 : Étudiants et leurs notes par matière

Tables disponibles :

1. **Étudiants** (id_etudiant, nom)
2. **Matières** (id_matière, nom_matière)
3. **Notes** (id_etudiant, id_matière, note)

Écrivez une requête SQL qui affiche la **liste des étudiants** avec leurs **matières** et leurs **notes**.

Exercice 8 : Liste des produits et leurs fournisseurs

1. **Produits** (id_produit, nom_produit, id_fournisseur)
2. **Fournisseurs** (id_fournisseur, nom_fournisseur)

Écrivez une requête SQL qui retourne le **nom des produits** avec leurs **fournisseurs**.

Exercice 9 : Liste des films et leurs réalisateurs

1. **Films** (id_film, titre, id_realisateur)
2. **Réalisateurs** (id_realisateur, nom)

Écrivez une requête SQL qui affiche la **liste des films** avec le **nom de leur réalisateur**.

Exercice 10 : Liste des employés et les projets sur lesquels ils travaillent

Tables disponibles :

1. **Employés** (id_employé, nom)
2. **Projets** (id_projet, nom_projet)
3. **TravailleSur** (id_employé, id_projet) (*table de relation*)

Écrivez une requête SQL pour afficher le **nom des employés** et les **projets sur lesquels ils travaillent**.

EXERCICE 11

Soit la représentation textuelle simplifiée d'une base de données concernant un cycle de formation destiné à des étudiants. Il regroupe un ensemble de matières. On considère que chaque enseignant n'enseigne qu'une seule matière et qu'à la fin du cycle de formation, une note par matière, est attribuée à chaque étudiant. D'autre part, les étudiants peuvent ne pas suivre les mêmes matières.

- ETUDIANT(CodeEt, NomEt, DatnEt)
- MATIERE(CodeMat, NomMat, CoefMat)
- ENSEIGNANT(CodeEns, NomEns, GradeEns, #CodeMat)
- NOTE(#CodeEt, #CodeMat, note)

- a. Les noms et les grades des enseignants de la matière dont le nom est 'BD'.
- b. La liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3'.

JEU BILINGUE : Traduire les mots et expressions suivantes : division ; Jointure