

# Chapitre 2: Bases de données avec MySQL

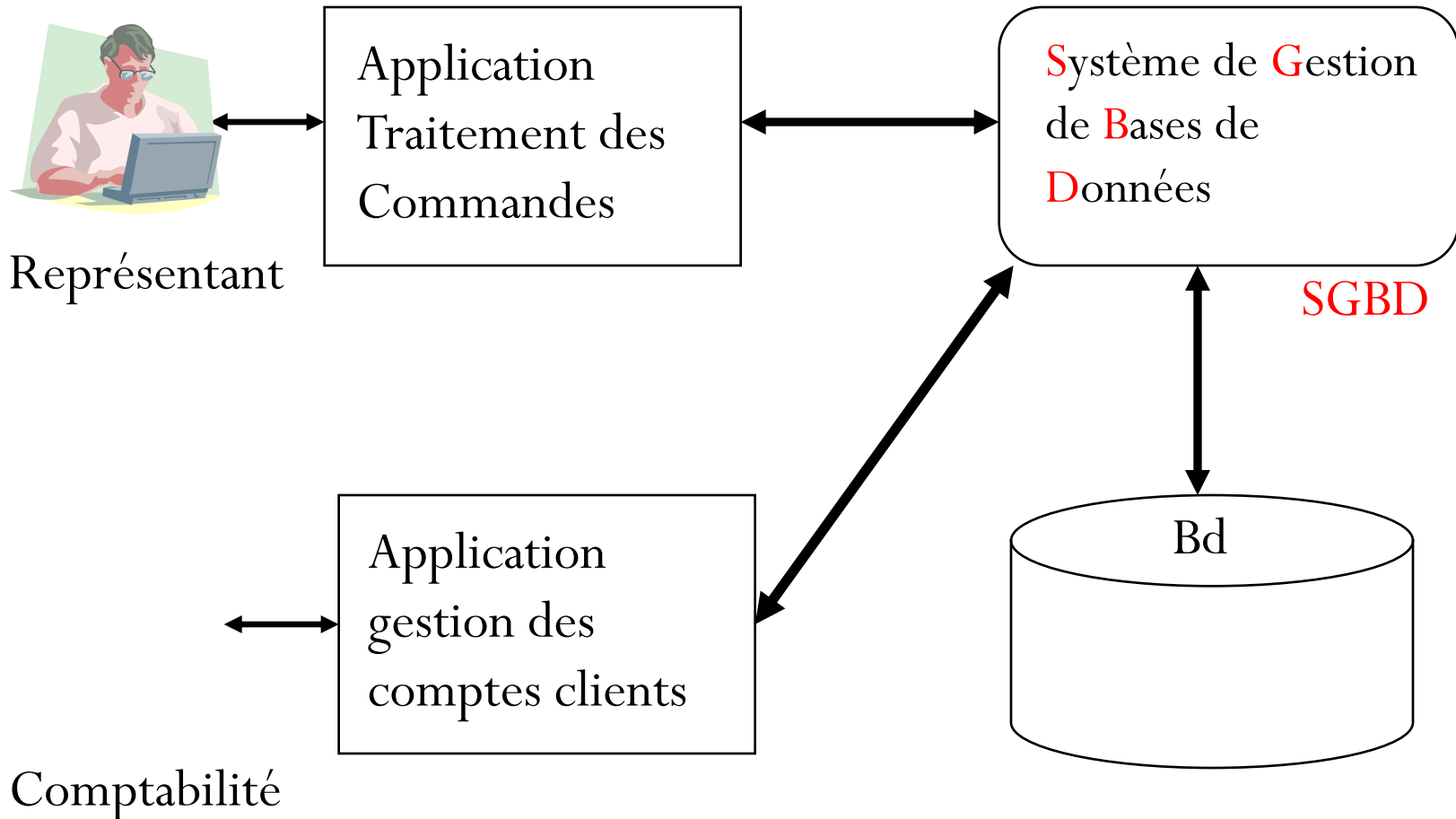
# Bases de données (BD)

- Une BD est une grande quantité de données, centralisées ou non, servant pour les besoins d'une ou plusieurs applications, interrogeables et modifiables par un groupe d'utilisateurs travaillant en parallèle.

# Apport d'une approche BD

- Intégration :
  - Description unique et globale des données
  - Eviter les redondances
- Indépendance :
  - Indépendance entre données et traitements
  - Séparation entre les descriptions logiques et physiques des données
- Sécurité :
  - Contrôle sémantique des données
  - Protection contre les accès non autorisés
  - Protection contre les pannes
- Facilités pour l'utilisateur :
  - Partage des données
  - Vision « haut niveau » et « personnalisée » des données
  - Manipulation « aisée » des données
  - Accès efficaces aux données
  - Répartition des données et des traitements

# Composantes d'une base de données



# Domaines d'application

- Comme élément central :
  - Banques de données (professionnelles, grand public)
  - Systèmes transactionnels (banques, assurances,...)
  - Systèmes d'aide à la décision (tableaux de bord, analyse statistique,...)
- Comme élément essentiel :
  - Systèmes d'information et de communication (entreprise, administration,...)
  - Systèmes de surveillance et de contrôle (télécoms, production,...)
- Comme élément auxiliaire :
  - Systèmes de recherche documentaire (textes, sons, graphiques, ...)
  - Systèmes XAO (CAO, PAO, EAO,...)
  - Systèmes experts

# Exemples de BD

- Cégeps, Universités :
  - *Données sur les étudiants*
    - *Id, nom, prénom, groupe, section, année...*
  - *Données sur les formations*
    - *matière, intervenant (enseignant), salle*
  - *Données sur les résultats*
    - *matière, intervenant (enseignant), pondération, résultat*
- Entreprises
  - *Données sur clients, fournisseurs, commandes*
  - *facturation,*
  - *gestion de stock, inventaire.*

# Exemples de BD

- Bibliothèques :
  - *Données sur les lecteurs*
    - *Id, nom, prénom, statistiques, montant dû...*
  - *Données sur les livres*
    - *Titre, ISBN, auteur, genre, code, emplacement, emprunts, collection, éditeurs*
  - *Données sur les auteurs*
    - *Livres, adresse, nom, prénom,*
- Etc..

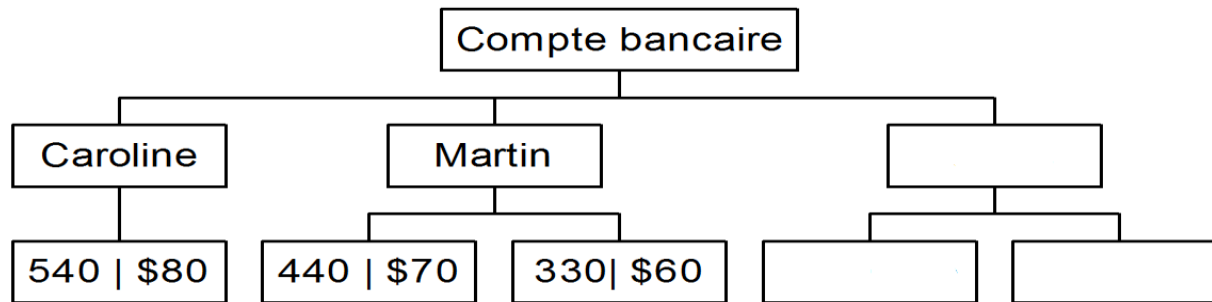
# Système de Gestion de Bases de Données (SGBD)

- Système de Gestion de Bases de Données (SGBD) est le logiciel qui prend en charge la structuration, le stockage, la mise à jour et la maintenance des données ; c'est, en fait, l'interface entre la base de données et les utilisateurs ou leurs programmes.



# Types de SGBD

## SGBD Hiérarchique

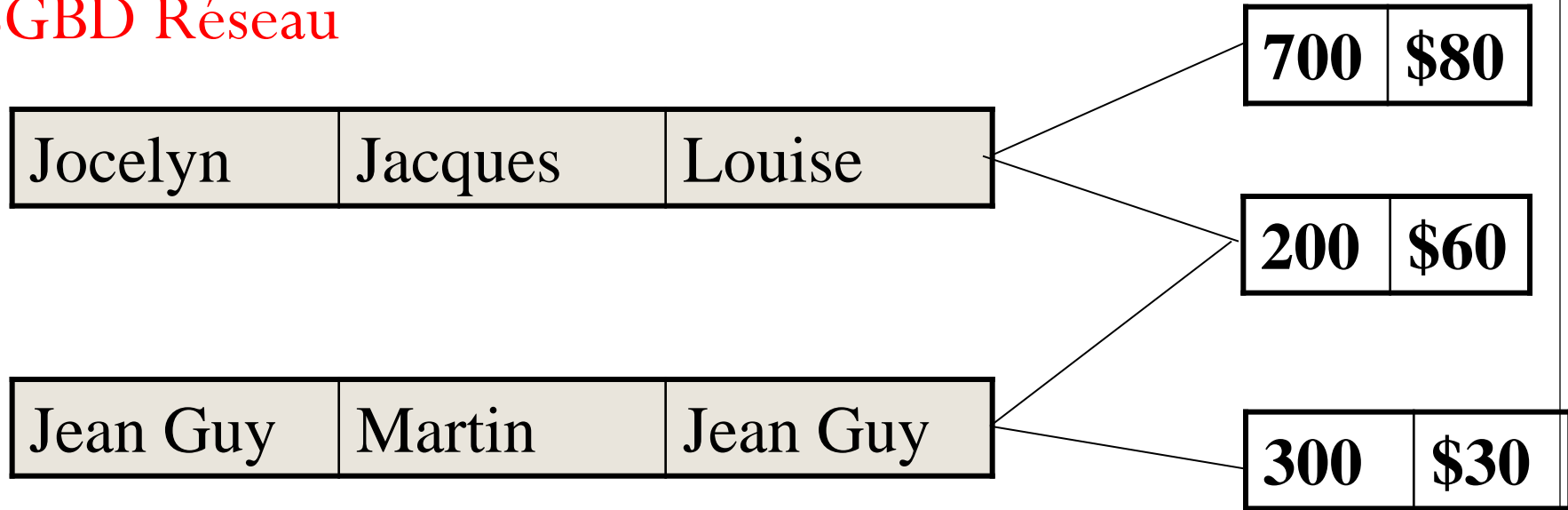


- L'accès aux données se fait du haut vers le bas.
- Il y a redondance de données donc perte de beaucoup d'espace mémoire.
- Un entretien très coûteux (langage de haut niveau)

**Les programmes ne sont pas indépendants de la structure logique de la base et du chemin d'accès à la base. Utilisation massive de pointeurs.**

# Types de SGBD (...)

## SGBD Réseau



Accès séquentiel de la gauche vers la droite et nécessite un langage externe complexe de programmation de haut niveau.

**Les programmes ne sont pas indépendants de la structure logique de la base et du chemin d'accès à la base. Utilisation massive de pointeurs.**

# Types de SGBD (...)

## SGBD Relationnel

- Le modèle relationnel est fondé sur la théorie mathématique des ensembles.
- Les données sont représentées sous forme de tables constituées de lignes et de colonnes (plus de pointeurs!).
- A permis le développement de langages non procéduraux (SQL).
- Les programmes sont indépendants de la structure logique de la base et du chemin d'accès.
- **Le modèle relationnel a ses limites concernant la réalisation d'applications multimédia, d'où l'orienté Objet**

### Exemple

Élève(num, nom, adresse, age)

Notes(numElève, note)

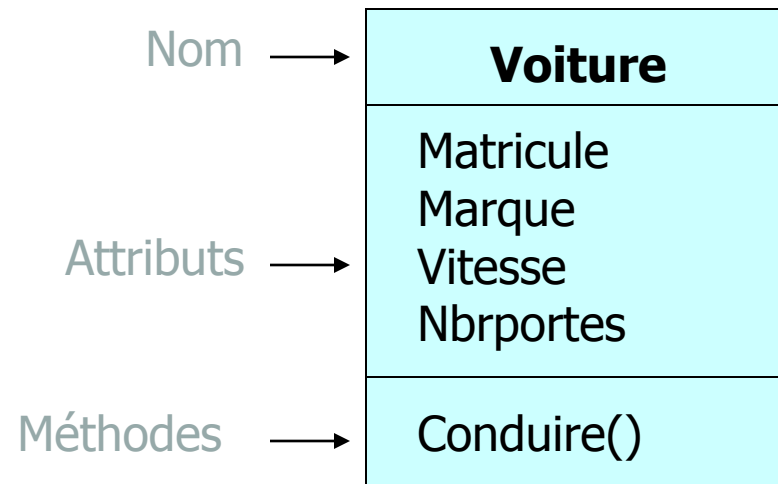
Livre(côte, titre, numElève, datePrêt)

Chambre(no, prix, numElève)

# Types de SGBD (...)

## SGBD Objet (SGBDOO)

- Les données sont représentées sous forme d'objets. Elles sont enregistrées avec les procédures / fonctions qui permettent de les manipuler.



# SGBDR

- Un **SGBDR** (Système de Gestion de Bases de Données Relationnel) est un logiciel qui permet de :
  - stocker,
  - consulter,
  - modifier,
  - supprimer

les données de la base de données.

Un **SGBDR** stocke les informations dans des tables.

# SGBDR

- Access
- Paradox
- Sybase
- SQL Server
- Oracle
- MySQL (libre avec PHP)
- InterBase
- PostgreSQL
- SQLite



# MySQL

- MySQL dérive directement de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel. Il en reprend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MySQL (sélections imbriquées, clés étrangères...)
- Le serveur de base de données MySQL est très souvent utilisé avec le langage de création de pages web dynamiques : PHP.

# Les bases de données avec MySQL

- Principe des bases de données MySQL
  - ▣ MySQL s'appuie sur le standard ANSI SQL92
  - ▣ Le cœur du système MySQL affiche une performance remarquable, tout particulièrement dans les accès indexés
  - ▣ MySQL utilise une architecture multiutilisateur, multitraitement
    - ▣ Permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes



# Les bases de données avec MySQL

## □ Syntaxe et conventions

### ▣ Les chaînes de caractères

- Notées entre des guillemets simples

- Notées entre des guillemets doubles dans le cas où la base utilise le mode ANSI

- Les séquences d'échappement

  - \0      ASCII 0 (NUL)

  - \n      Saut de ligne. Correspond au caractère ASCII CHR(13)

  - \t      Tabulation. Correspond au caractère ASCII CHR(9)

  - \r      Retour chariot. Correspond au caractère ASCII CHR(10)

  - \b      Retour caractère

  - \'      Guillemet simple

  - \"      Guillemet double

  - \\      Barre oblique inverse

  - %      Pourcentage

  - \\_      caractère souligné

# Les bases de données avec MySQL

- Syntaxe et conventions
  - Les nombres
    - Les nombres à virgule flottante utilisent le point(.) comme séparateur décimal
    - Les valeur négatives sont précédées du signe moins
    - Les nombres entiers utilisés en relation avec les nombres à virgule flottante sont interprétés comme des nombres à virgule flottante
    - Mysql prend en charge les nombres hexadécimaux
    - NULL signifie « aucune donnée » et ne doit pas être confondu avec le nombre 0 ou la chaîne de caractère vide "" .

# Les bases de données avec MySQL

## □ Syntaxe et conventions

### ▣ Les variables

- Les variables peuvent contenir des nombres entiers, des nombres réels ou des chaînes de caractères
- Les noms des variables peuvent comporter des caractères alphanumériques ainsi que les caractères spéciaux . et \_.
- Il n'est pas nécessaire d'initialiser les variables, qui possède par défaut la valeur NULL
- Syntaxe de Définition de variables
  - SET @variable= {integer | real | string} [,@variable=...]
  - ou
  - @variable:=expression

### ▣ Les types de données

- MySql connaît les types de données numériques, de date et d'heure ainsi que chaînes

# Les bases de données avec MySQL

## □ Les types de données

### ▣ Les types de données numériques

- M : nombre maximal de chiffres affichés
- U (Unsigned) : le caractère de signe est omis
- Z (Zerofill) : les valeurs manquantes sont remplies par NULL
- D : le nombre de décimales affichées pour les virgule flottante

- TINYINT[(M)] [UNSIGNED] [ZEROFILL] : très petit nombre entier
- SMALLINT[(M)] [UNSIGNED] [ZEROFILL] : petit nombre entier
- MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL] : nombre entier moyen
- INT[(M)] [UNSIGNED] [ZEROFILL] : nombre entier normal
- INTEGER[(M)] [UNSIGNED] [ZEROFILL] : synonyme de INT
- BIGINT[(M)] [UNSIGNED] [ZEROFILL] : grand nombre entier
- FLOAT(précision) [ZEROFILL] : nombre à virgule flottante signé
- FLOAT[(M,D)] [ZEROFILL] : petit nombre à virgule flottante signé
- DOUBLE[(M,D)] [ZEROFILL] : petit nombre à virgule flottante normal, à double précision, signé
- DOUBLE précision[(M,D)] [ZEROFILL] : synonyme de double
- REAL[(M,D)] [ZEROFILL] : synonyme de double
- DECIMAL[(M,D)] [ZEROFILL] : nombre à virgule flottante signé
- NUMERIC(M,D) [ZEROFILL] : synonyme de DECIMAL

# Les bases de données avec MySQL

- Les types de données
  - Les types de données de date et d'heure
- DATE : Date
- DATETIME : Date et heure
- TIMESTAMP[(M)] : Tampon horaire UNIX
- TIME : Heure
- YEAR[(2|4)] : Année

# Les bases de données avec MySQL

## □ Les types de données

### ▣ Les types de données chaînes

- CHAR(M) : chaîne de caractère de longueur fixe M
- VARCHAR(M) : chaîne de caractère de longueur variable
- TINYBLOB, TINYTEXT : taille maximale 255 caractères
- BLOB, TEXT : taille maximale 65535 caractères
- MEDIUMBLOB, MEDIUMTEXT : taille maximale 16777215 caractères
- LONGBLOB, LONGTEXT : taille maximale 4294967295 caractères
- ENUM('value1','value2',...) : Une chaîne de caractères qui n'a qu'une seule valeur, issue d'une liste : ou encore la valeur NULL ou la valeur d'erreur spéciale "".
- SET('value1','value2',...) : Une chaîne de caractères qui a zéro, un ou plusieurs valeurs issues d'une liste

# Les bases de données avec MySQL

- Les Opérateurs de MySql
  - ▣ Les opérateurs arithmétiques
    - +, -, \*, /
  - ▣ Les opérateurs logiques
    - NOT (!), OR(| |), AND(&&)
  - ▣ Les opérateurs de comparaison
    - =, <> (!=), <=, <, >=, >

```
Select    NomArt,
          PrixArt AS Prix,
          PrixArt * 0.196 AS 'TVA'
          PrixArt / 10 as 'Marge'
From      article as A,
          groupeArticle as G
Where A.NumGrArt=G.NumGrArt
And      ( G.NumGrArt<>1 OR Not (A.type = 2))
```

# Les bases de données avec MySQL

- Création, suppression et utilisation d'une base de données
  - ▣ Création de bases de données
    - CREATE DATABASE nom\_bd
  - ▣ Suppression de bases de données
    - DROP DATABASE [IF EXISTS] nom\_bd
      - Le mot-clé IF EXISTS est utilisé pour éviter que la tentative de supprimer une base de données inexistante ne donne lieu à une erreur
  - ▣ Utilisation d'une base de données
    - USE nom\_bd
      - Permet d'établir que la base de données spécifiées est la base de données par défaut. Toutes les requêtes suivantes se rapportent donc à cette base de données.



# Les bases de données avec MySQL

- Définition et modification de la structure d'une table
  - Création d'une Table
    - CREATE TABLE : permet de créer une nouvelle table dans la base de données courante
    - Syntaxe :
      - CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name [(create\_definition, ...))  
[table\_options] [select\_statement]
      - Create\_definition :
        - Col\_name type [NOT NULL | NULL] [DEFAULT default\_value] [AUTO\_INCREMENT]  
[PRIMARY KEY] [reference\_definition] OU
        - PRIMARY KEY (index\_col\_name, ...) OU
        - KEY [index\_name](index\_col\_name, ...) OU
        - INDEX [index\_name](index\_col\_name, ...) OU
        - UNIQUE [INDEX] [index\_name] (index\_col\_name, ...) OU
        - [CONSTRAINT symbol] FOREIGN KEY index\_name (index\_col\_name) OU
        - [reference\_definition] OU
        - CHECK (expr)

# Les bases de données avec MySQL

- Définition et modification de la structure d'une table
  - Création d'une Table
    - CREATE TABLE : permet de créer une nouvelle table dans la base de données courante
    - Syntaxe :
      - L'attribut **AUTO\_INCREMENT** : signifie que le contenu d'un champ de type INTEGER est incrémenté automatiquement d'une unité après l'insertion d'un nouvel enregistrement.
        - Ne peut être affecté qu'une seule fois à une table donnée
      - L'attribut **PRIMARY\_KEY** : définit une clé d'index primaire unique
        - Chaque table peut comporter au maximum une clé d'index primaire.
        - Une clé d'index primaire peut être formée à partir d'une combinaison d'un maximum de 32 colonnes
        - Les colonnes d'une clé d'index primaire doivent être définies avec le paramètre NOT NULL

# Les bases de données avec MySQL

- Définition et modification de la structure d'une table
  - Création d'une Table
    - Syntaxe :
      - L'attribut **UNIQUE\_KEY** : définit une clé d'index. Elle ne peut comporter que des valeurs uniques.
      - L'attribut **INDEX** : définit un index.
        - Les champs TEXT et BLOB ne peuvent pas être indexés

# Les bases de données avec MySQL

## □ Définition et modification de la structure d'une table

### ▣ Création d'une Table

#### ■ Exemple:

```
CREATE TABLE article (  
  NumArt BIGINT NOT NULL AUTO_INCREMENT  
    PRIMARY KEY,  
  numCde VARCHAR(25) NOT NULL,  
  NomART VARCHAR(100) NOT NULL,  
  TexteArt MEDIUMTEXT NOT NULL,  
  PrixArt DECIMAL(8,2) NOT NULL  
  NumGrArt BIGINT NOT NULL,  
  NumSGrArt BIGINT NOT NULL,  
);
```

# Les bases de données avec MySQL

□ Saisie d'enregistrement

▣ Insérer de nouveaux enregistrement

■ **INSERT INTO** article (NumCde, NomArt, TexteArt, PrixArt,  
NumGrArt, NumSGrart)  
VALUES ('1-001-001-001', 'articl1', 'c'est un article', 7.95, 1, 1);

# Les bases de données avec MySQL

- Sélection d'enregistrement

- Syntaxe

SELECT [DISTINCT | ALL] expression\_de\_selection

FROM tables

WHERE expression\_where

GROUP BY col\_name, ...]

HAVING where\_definition]

[ORDER BY [ASC | DESC]]

- Exemples

- SELECT \* FROM article WHERE PrixArt > 50

- SELECT NumGrArt, AVG(PrixArt) FROM article GROUP BY NumArt