

&Bases de données : Exercices d'application

1. Tester des requêtes simples

Cet exercice sera effectué à partir de la base de données en ligne disponible sur le site w3schools à l'adresse : https://www.w3schools.com/sql/trysql.asp?filename=trysql_asc

On veillera à bien respecter la syntaxe et les conventions. Les commandes SQL doivent être écrites en majuscules, et les requêtes doivent être terminées par ; (même si le site est plutôt souple).

La base est présentée dans la colonne à droite en-dessous de « your Database ». On pourra cliquer sur les différentes tables pour avoir plus d'informations. Pour exécuter une requête, il faudra cliquer sur Run SQL »

Analyse de la base de données

Relever le nombre de tables que contient cette base de données.

8 tables (Customers, Categories, Employees, OrderDetails, Orders, Products, Shippers, Suppliers)

Relever le nombre d'attributs de la table Categories ainsi que le nombre d'enregistrements.

3 attributs (CategoryID, CategoryName, Description) et 8 enregistrements

Q1. Rechercher l'attribut de la table Categories qui peut constituer une clé primaire.

CategoryID peut constituer une clé primaire

Relever les attributs de la table Products qui peuvent constituer des clés étrangères.

Indiquer vers quelle autres tables ces clés étrangères seraient liées.

SupplierID et CategoryID peuvent constituer des clés étrangères. Elles sont respectivement en lien avec les attributs SupplierID : Suppliers et CategoryID:Categories

Étude de la table Customers.

Écrire une requête en langage SQL permettant d'afficher toutes les lignes de la table.

SELECT * FROM Customers;

Indiquer le domaine de l'attribut CustomerID. **Écrire** une requête en langage SQL permettant d'afficher les lignes correspondant au client dont le CustomerID vaut 40.

Indiquer le type d'opération qui est alors réalisée.

CustomerId : Entier (Integer)

SELECT * FROM Customers WHERE CustomerID = 40; → Sélection

Relever le nombre de lignes affichée. **Justifier** ce nombre.

Il y a 1 ligne puisque le filtrage s'effectue sur la clé primaire

Écrire une requête en langage SQL permettant d'afficher toutes les lignes mais en précisant seulement les villes des consommateurs. **Indiquer** le type d'opération qui est alors réalisée.

SELECT City FROM Customers; → Sélection de données

Modifier la requête précédente de sorte à ne pas afficher les doublons.

SELECT DISTINCT City FROM Customers;

Indiquer le domaine de l'attribut City. **Écrire** une requête permettant d'afficher les noms et les villes des clients français.

City : Chaîne de caractères

SELECT CustomerName, City FROM Customers WHERE Country = 'France';

Modifier la requête précédente de sorte à afficher le résultat dans l'ordre alphabétique du nom de la ville.

SELECT CustomerName, City FROM Customers WHERE Country = 'France' ORDER BY City ASC;

Étude de la table Products.

Écrire une requête affichant le nom et le conditionnement des produits de moins de 20 dollars (inclus).

SELECT ProductName, Unit FROM Products WHERE Price<=20;

Modifier la requête précédente pour afficher les noms et les prix des produits compris entre 30 dollars (inclus) et 60 dollars (exclu) en affichant d'abord les plus chers.

SELECT ProductName, Price FROM Products WHERE Price>=30 AND Price<60 ORDER BY Price DESC ;

Modifier la requête précédente pour afficher les lignes des articles dont le prix est exactement de 14 dollars en les classant par ordre alphabétique du nom de produit.

SELECT ProductName, Price FROM Products WHERE Price=14 ORDER BY ProductName ASC ;

2. Exprimer des requêtes simples, sans jointure ni imbrication

Tous les titres de livre.

SELECT titre FROM LIVRE;

Tous les noms d'utilisateur.

SELECT nom FROM USAGER;

Tous les noms d'utilisateur en retirant les doublons.

SELECT DISTINCT nom FROM USAGER;

Les titres des livres publiés avant 1980.

SELECT titre FROM LIVRE WHERE annee < 1980 ;



Les isbn des livres à rendre pour le 01/01/2020.

```
SELECT isbn FROM EMPRUNT WHERE retour = '2020-01-01';
```

Les noms d'auteurs triés par ordre alphabétique.

```
SELECT DISTINCT nom FROM AUTEUR ORDER BY nom ASC;
```

Les noms d'usagers vivant dans le 12° ou 13° arrondissement de Paris (codes postaux 75012 et 75013).

```
SELECT nom FROM USAGER WHERE cp='75012' or cp='75013' ;
```

3. Exprimer des requêtes complexes

Le titre des livres empruntés

```
SELECT LIVRE.titre FROM LIVRE JOIN EMPRUNT ON LIVRE.isbn =  
EMPRUNT.isbn ;
```

Le titre des livres empruntés à rendre avant le 31/03/2020.

```
SELECT LIVRE.titre FROM LIVRE JOIN EMPRUNT ON EMPRUNT.isbn=LIVRE.isbn  
WHERE EMPRUNT.retour<'2020-03-31';
```

Le nom et prénom de l'auteur du livre "1984".

```
SELECT AUTEUR.nom FROM AUTEUR JOIN AUTEUR_DE ON  
AUTEUR_DE.a_id=AUTEUR .a_id JOIN LIVRE ON AUTEUR_DE.isbn=LIVRE .isbn  
WHERE LIVRE.titre="1984";
```

Le nom et le prénom des usagers ayant emprunté des livres, sans doublons (i.e. si un usager a emprunté plusieurs livres, il ne doit apparaître qu'une fois dans le résultat).

```
SELECT DISTINCT USAGER.nom, USAGER.prenom FROM USAGER JOIN EMPRUNT ON  
USAGER.code_barre=EMPRUNT .code_barre ;
```

Même requête que précédemment, avec les noms triés par ordre alphabétique.

```
SELECT DISTINCT USAGER.nom, USAGER.prenom FROM USAGER JOIN EMPRUNT ON  
USAGER.code_barre=EMPRUNT .code_barre ORDER BY USAGER.nom ASC
```

Les titre des livres publiés strictement avant 'Dune'.

```
SELECT titre from LIVRE WHERE annee < (SELECT annee FROM LIVRE WHERE nom  
= 'Dune') ;
```

Les noms et prénoms des auteurs des livres trouvés à la question précédente.

```
SELECT AUTEUR.nom FROM AUTEUR JOIN AUTEUR_DE ON  
AUTEUR_DE.a_id=AUTEUR .a_id JOIN LIVRE ON AUTEUR_DE.isbn=LIVRE .isbn  
WHERE LIVRE.annee < (SELECT LIVRE.annee FROM LIVRE WHERE LIVRE.nom =  
'Dune') ;
```

Comme la question précédente, en retirant les doublons.

```
SELECT DISTINCT AUTEUR.nom FROM AUTEUR JOIN AUTEUR_DE ON  
AUTEUR_DE.a_id=AUTEUR .a_id JOIN LIVRE ON AUTEUR_DE.isbn=LIVRE .isbn  
WHERE LIVRE.annee < (SELECT LIVRE.annee FROM LIVRE WHERE LIVRE.nom =  
'Dune') ;
```



Le nombre de résultats trouvés à la question précédente.

```
SELECT DISTINCT COUNT(AUTEUR.nom) FROM AUTEUR JOIN AUTEUR_DE ON  
AUTEUR_DE.a_id=AUTEUR .a_id JOIN LIVRE ON AUTEUR_DE.isbn=LIVRE .isbn  
WHERE LIVRE.annee < (SELECT LIVRE.annee FROM LIVRE WHERE LIVRE.nom =  
'Dune') ;
```

