

La forme générale d'une requête SQL est la suivante :

```
SELECT liste de colonnes pour la table résultat  
FROM liste de tables sur lesquelles la requête est posée  
WHERE conditions de sélection de lignes pour la table résultat  
GROUP BY liste des colonnes définissant le regroupement des lignes de résultat  
HAVING conditions de sélection de groupes dans le résultat final  
ORDER BY liste des colonnes définissant l'ordre d'affichage du résultat final;
```

Les deux premières clauses (**SELECT** et **FROM**) sont obligatoires, les autres clauses sont optionnelles ; leur nécessité dépend de ce que l'on veut obtenir dans la réponse à la requête. Il y a de nuances entre les listes de colonnes dans les clauses **SELECT**, **WHERE**, **HAVING** et **ORDER BY**. De manière générale, le terme colonne signifie une expression de constantes ou de noms de colonnes des tables apparues dans la clause **FROM** ; l'expression peut utiliser des opérations arithmétiques ou agrégates. Notons que la clause **WHERE** n'accepte pas d'opérations agrégates.

Les constantes dans une expression sont mises entre les apostrophes. Dans la clause **SELECT** on peut nommer une expression en mettant un nom (une chaîne de caractères mise entre les guillemets) juste après l'expression.

Les tables dans la clause **FROM** sont les tables de la base de données ou les vues de la base. Une vue est définie par une requête :

```
CREATE VIEW NomdeVue (liste_de_colonnes)  
AS Requête;
```

Où *liste_de_colonnes* est une liste des colonnes correspondantes aux colonnes apparues dans la clause **SELECT** de *Requête* ; on ne définit pas les types des colonnes dans la liste.

Exemple 1 :

```
SELECT NomdePiece, PrixUnit, PrixUnit * Qte "Prix Total Pour Piece"  
FROM Piece P, Facture F  
WHERE P.codebarre = F.codebare  
      AND DateFacture = '21-Feb-07'  
      OR MarqueVoiture LIKE 'Peugeot%';
```

Dans l'exemple 1, le symbole % représente une chaîne de caractères quelconque, y comprise la chaîne vide. Le symbole _ représente un caractère quelconque.

Exemple 2 :

```
SELECT NoFacture, SUM(PrixUnit * Qte) "Prix Total de Facture", 'Pas Chère'  
FROM Piece P, Facture F  
WHERE P.codebarre = F.codebarre  
      AND PrixFacture IS NULL  
GROUP BY NoFacture  
HAVING SUM(PrixUnit * Qte) < 500  
ORDER BY SUM(PrixUnit * Qte) ;
```

Dans l'exemple 2, on retrouve plusieurs types d'opérateurs :

- Expressions arithmétiques : Ces expressions se calculent avec les constantes ou les valeurs d'une ligne de données.
- Numériques : Les constantes numériques et les noms de colonnes de types **NUMBER**, les expressions utilisant +, -, *, /, les fonctions **ABS(X)**, **POWER(X,n)**, **MOD(n,k)**, **FLOOR(X)**, **CEILING(X)**, **ROUND(X,n)**, **TRUNC(X,n)**, **SQRT(X)**, **GREATEST(X,Y)**, **LEAST(X,Y)**.
- Chaînes de caractères : Les constantes alphanumériques (entre apostrophes), les noms de colonnes de types **VARCHAR**, les expressions utilisant || (concaténation), **SUBSTR(X,n,k)**, **INSTR(X,Y)**, **INITCAP(X)**, **LOWER(X)**, **UPPER(X)**, **GREATEST(X,Y)**, **LEAST(X,Y)**, **DECODE(X,liste de codages,default)**.
- Date : Les constantes et noms de colonnes de type dates, et les expressions utilisant **SYSDATE**, **date1 - date2**, **date1 + n**, **date1 - n**, **ROUND(date1)**, **NEXT_DAY(monday)**, **LAST_DAY(date1)**,

ADD_MONTHS(date1,n), **MONTHS_BETWEEN**(date1,date2), **TO_DATE**(chaine,format), **TO_CHAR**(date1,format), **GREATEST**(X,Y), **LEAST**(X,Y).

- **NULL** : **NVL**(X,Y) retourne X si X n'est pas **NULL**, sinon retourne Y.
- **Les fonctions agrégates** : **SUM**, **MIN**, **MAX**, **AVG**, **VARIANCE**, **STTDEV**.

Exercices

Utiliser les schémas de relations du TP 2.

Fourn(Fournisseur, Adr, Tel)
Piece(IdPiece, NomdePiece, Codebarre, PrixUnit, Fournisseur)
Compatib(IdPiece, NomdePiece, MarqueVoiture, TypeVoiture)
Facture(NoFacture, Codebarre, Client, DateFacture, Qte, PrixFacture)

Supposer que les prix donnés dans la table **Piece** sont les prix hors taxe, et que la taxe est de 19.6% du prix donné. Supposer de plus que le prix de main d'œuvre pour monter une pièce est de 50% du prix toute taxe comprise (ttc).

1. Donner les prix unitaires HT, la taxe, et les prix TTC des pièces compatibles avec une Peugeot 205.
2. Calculer les prix HT, TTC, les prix de mains d'œuvre et le prix total de la facture de Dupont, datée aujourd'hui.
3. Liste de pièces, leurs prix TTC, prix de main d'œuvre, et les noms de voitures compatibles en concaténant la marque et le type de voiture compatible en une colonne.
4. Liste des factures datées il y a 10 jours, en la triant dans l'ordre de prix.
5. Afficher dans une même liste les fournisseurs à Marseille qui fournissent au moins une pièce compatible avec les Peugeot ou les Renault, indiquer cela par la mention PR, et ceux qui ne fournissent aucune pièce compatible avec les Peugeot ou les Renault ; indiquer cela par la mention N-PR.
6. Liste des pièces dans les factures du client Dupont, avec tous les détails concernant les pièces et leurs fournisseurs (noms en lettres majuscules, 16 premiers caractères de l'adresse et 6 derniers chiffres du numéro de téléphone) ; trier selon noms de fournisseurs et numéros de factures.
7. Les prix minimale, maximal, moyenne de pièces compatibles avec un Citroën.
8. Liste des fournisseurs de freins compatibles avec un Citroën offrant le prix le moins cher.
9. Sur l'ensemble des factures de Jean, donner les prix totaux pour chaque pièce.
10. Quelles pièces coûtent le plus à Jean, depuis qu'il a des achats des pièces ?
11. Calculer les sommes des prix hors taxe des pièces par fournisseur, dans les factures de Janvier 2007.
12. Quels fournisseurs ont la somme la plus grande parmi les sommes trouvées en question 11