LE SQL par l'exemple

Projection

SELECT NomColonne(s)
FROM table

Renomage des colonnes =
SELECT NomColonne(s) AS NomColonneRésultat

Suppression des doublons =
SELECT DINSTINCT NomColonne(s)
FROM table

Extraction de toutes les colonnes d'une table =
*SELECT **
FROM table

Ex :
Lister les informations sur tous les clients
SELECT *
FROM client
Lister les noms de tous les clients
SELECT NomC
FROM Client
Lister les numéros de produits en les renommant
SELECT N°P *AS* NuméroProduits
FROM Produit

Restriction

SELECT NomColonne(s)
FROM table
WHERE conditions

Condition élémentaire :
NomColonne

Opérateur de comparaison
Constante, NomColonne ou liste de valeurs

Conditions :
Conditions élémentaires reliées par un opérateur logique AND, OR, Not

Opérateurs de comparaison
{=, !=, <, <=, >, >=}. L'opérateur différent peut aussi s'écrire <> ou ^=
IS [NOT] NULL : Teste si le contenu d'une colonne est une valeur nulle ou pas
[NOT] IN (liste de valeur) : Teste si le contenu coïncide avec le contenu de la liste qui suit entre parenthèses ou pas
[NOT] BETWEEN v1 AND v2 : Teste si le contenu d'une colonne est compris entre v1 et v2 ou pas
[NOT] LIKE : Compare la valeur de la colonne et celle de la chaîne de caractères qui suit en utilisant des caractères génériques :

- *% substitue 0 ou plusieurs caractères*
- *_ substitue 1 seul caractère*

Ex :
Lister toutes les informations sur des produits dont QtéP < 10
SELECT *
FROM Produit
WHERE QtéP < 10

Lister toutes les informations sur les clients de Paris, Nice ou Lyon
SELECT *
FROM Client
WHERE Ville *IN* ('Paris', 'Nice', 'Lyon')

Lister la désignation et le prix des produits dont prix >500 et < 1000
SELECT Design, Prix
FROM Produit
WHERE Prix *BETWEEN* 500 *AND* 1000

Lister nom et adresse des clients dont le nom commence par un D
SELECT NomC, AdrC
FROM Client
WHERE NomC *LIKE* 'D%'

Lister les numéros pour lesquels CondPart ne sont pas renseignés
SELECT N°C
FROM Client
WHERE CondPart *IS NULL*

Fonctions agrégats

Elles s'appliquent à l'ensemble des valeurs d'une colonne et sont utilisées uniquement dans la clause SELECT. Elles ignorent les valeurs nulles.

- *MAX : Valeur maximale des éléments d'une colonne*
- *MIN : Valeur minimale des éléments d'une colonne*
- *AVG : Moyenne arithmétique des valeurs d'une colonne*
- *SUM : Somme des valeurs d'une colonne*
- *COUNT : Nombre d'occurrences d'une colonne*

Ex :
Trouver le nombre de clients de la ville de Paris
SELECT COUNT (*) *AS* Nombre
FROM Client
WHERE Ville = 'Paris'
Trouver le prix moyen des produits
SELECT AVG (Prix) *AS* Moyenne
FROM Produit
Rmq : On ne peut pas combiner dans le *SELECT* une fonction agrégat (1 seule valeur) et une colonne (plusieurs valeurs), sauf avec *GROUP_BY*.

Requêtes imbriquées

Une requête imbriquée dans une autre requête est appelée sous-requête. Elle st entre parenthèses et se formule comme une autre requête. On l'utilise quand les valeurs de la condition font elles-mêmes partie de la base de données.
Résultat de la sous requête formée d'une seule valeur
On utilise les opérateurs de comparaison classiques. En particulier, c'est le cas des requêtes utilisant une fonction agrégat dans la clause SELECT.

Ex :
N°P dont le prix > prix moyen
SELECT N°P
FROM Produit
WHERE Prix > (*SELECT AVG* (Prix) *FROM* Produit)

Désignation de tous les produits dont le prix = prix d'une chaise
SELECT Design
FROM Produit
WHERE Prix = (*SELECT* Prix *FROM* Produit *WHERE* Design = 'chaise')

Résultat de la sous-requête d'un ensemble de valeurs

Dans ce cas, le résultat de la comparaison est VRAI, dans WHERE, si

- *Soit la condition est vérifiée pour au moins une des valeurs résultantes de la sous-requête. On utilise : IN, ANY (ou SOME) qui sont précédés par un opérateur de comparaison.*
- *Soit la condition est vérifiée pour toutes les valeurs. On utilise ALL*

Ex :
Lister les N°C ayant commandé 1 produit dont le prix = 1000
SELECT N°C
FROM Commande
WHERE N°P *IN* (*SELECT* N°P *FROM* Produit *WHERE* Prix = 1000)

Lister le nom des employés dont le salaire > salaire de tous employés dépt 30
EMPLOYE (N°E, Nom, Salaire, Dept)
SELECT Nom
FROM Employé
WHERE Salaire > *ALL* (*SELECT* Salaire *FROM* Employé *WHERE* Dept = 30)

N°P le – cher (= N°P dont il n'existe pas de produit + cher)
SELECT N°P
FROM Produit p1
WHERE NOT EXISTS (*SELECT* * *FROM* Produit p2 *WHERE* p2.Prix > p1.Prix)

Lister les noms des clients qui ont commander le produit 2
SELECT NomC
FROM Client
WHERE EXISTS (*SELECT* * *FROM* Commande Cm *WHERE* N°P = 2 *AND* Client.N°C = Cm.N°C)

Tri des résultats

Pour ordonner les résultats, on utilise la clause ORDER BY

Lister les libellés et prix des produits triés par ordre de prix croissants puis par ordre alphabétique de libellé
SELECT Design, Prix
FROM Produit
ORDER BY Prix, Design

Trouver la somme des quantités commandées par produit
SELECT NP, SUM(Qté) AS Somme
FROM Commande
GROUP BY NP

Trouver la somme des quantités commandées par produit et par client
SELECT NP, NC, SUM (Qté) AS Somme
FROM Commande
GROUP BY NP, NC

Classification conditionnelle
On peut, après un GROUP BY introduire une sélection pour éliminer quelques groupes ne vérifiant pas une condition / Trouver la somme des quantités commandées par produit et par client, en éliminant les commandes dont le total est < 10
SELECT NP, NC, *SUM* (Qté) *AS* Somme
FROM Commande
GROUP BY NP, NC
HAVING SUM(Qté) >= 10

Les requêtes d'action

Les requêtes Ajout

Les requêtes Ajout permettent d'insérer des enregistrements dans une table soit à partir d'une liste de valeurs, soit à partir d'enregistrement d'une autre table

Ajout d'une liste de valeur

INSERT INTO NomTable(champ1, champ2, champ3,...) VALUES (valeur1, valeur2, valeur3,...)
WHERE Condition

ajout d'enregistrement provenant d'une autre table

INSERT INTO NomTableDestination(champ1, champ2, champ3,...)
SELECT champA, ChampB, Champ C
FROM NomTableSource
WHERE Condition

Les requêtes mises à jour

Les requêtes mises à jour permettent de modifier un ou plusieurs enregistrements automatiquement.

UPDATE NomTable
SET NomChamp= nouvellevaleur
WHERE Condition

Les requêtes suppression

Une requête suppression a pour effet de supprimer tous les enregistrements d'une table répondant à une condition déterminée

```
DELETE *
FROM NomTable
WHERE Condition
```