

SQL SELECT

L'utilisation la plus courante de SQL consiste à lire des données issues de la base de données. Cela s'effectue grâce à la commande **SELECT** qui retourne des enregistrements dans un tableau de résultat. Cette commande peut sélectionner une ou plusieurs colonnes.

Commande basique

L'utilisation basique de cette commande s'effectue de la manière suivante :

```
SELECT nom_du_champ FROM nom_du_tableau
```

Cette requête SQL va **sélectionner** (SELECT) le champ "nom_du_champ" **provenant** (FROM) du tableau appelé "nom_du_tableau".

Exemple

Imaginons une base de données appelée "client" qui contient des informations sur les clients d'une entreprise.

Table "client" :

| identifiant | prenom | nom | ville |
|-------------|---------|---------|-----------|
| 1 | Pierre | Dupond | Paris |
| 2 | Sabrina | Durand | Nantes |
| 3 | Julien | Martin | Lyon |
| 4 | David | Bernard | Marseille |
| 5 | Marie | Leroy | Grenoble |

Si l'on veut avoir la liste de toutes les villes des clients, il suffit d'effectuer la requête SQL ci-dessous :

```
SELECT ville FROM client
```

De cette manière on obtient le résultat suivant :

| ville |
|-----------|
| Paris |
| Nantes |
| Lyon |
| Marseille |
| Grenoble |

Obtenir plusieurs colonnes

Avec la même table client il est possible de lire plusieurs colonnes à la fois. Il suffit tout simplement de séparer les noms des colonnes par une virgule. Pour obtenir les **prénoms** et les **noms** des **clients** il faut alors faire la requête suivante :

```
SELECT prenom, nom FROM client
```



Ce qui retourne ce résultat:

| prenom | nom |
|---------|---------|
| Pierre | Dupond |
| Sabrina | Durand |
| Julien | Martin |
| David | Bernard |
| Marie | Leroy |

Obtenir toutes les colonnes d'un tableau

Il est possible de retourner automatiquement toutes les colonnes d'un tableau sans avoir à connaître le nom de toutes les colonnes, il faut simplement utiliser le caractère "*" (étoile). C'est un joker qui permet de sélectionner toutes les colonnes de la manière suivante:

```
SELECT * FROM client
```

Cette requête SQL retourne exactement les mêmes colonnes qu'il y a dans la base de données. Dans notre cas, le

| identifiant | prenom | nom | ville |
|-------------|---------|---------|-----------|
| 1 | Pierre | Dupond | Paris |
| 2 | Sabrina | Durand | Nantes |
| 3 | Julien | Martin | Lyon |
| 4 | David | Bernard | Marseille |
| 5 | Marie | Leroy | Grenoble |

Il y a des avantages et des inconvénients à l'utiliser. Pour en savoir plus sur le sujet il est recommandé de lire l'article [du sélecteur étoile](#).

Cours avancé : ordre des commandes du SELECT

Cette commande SQL est relativement commune car il est très fréquent de devoir lire les données issues d'une base de données. Il existe plusieurs commandes qui permettent de mieux gérer les données que l'on souhaite lire. Voici un petit aperçu des fonctionnalités abordées sur le reste du site:

- Joindre un autre tableau aux résultats
- Filtrer pour ne sélectionner que certains enregistrements
- Classer les résultats
- Grouper les résultats pour faire uniquement des statistiques (note moyenne, prix le plus élevé ...)

Une requête SELECT peut devenir assez longue. Juste à titre informatif, voici une requête SELECT qui possède toutes les commandes possibles:

```
SELECT *  
FROM table  
WHERE condition  
GROUP BY expression  
HAVING condition  
{ UNION | INTERSECT | EXCEPT }  
ORDER BY expression  
LIMIT count  
OFFSET start
```



A noter : cette requête imaginaire sert principale d'aide-mémoire pour savoir dans quel ordre sont utilisé chacun de la requête SELECT.

Cours recommandés

- [SQL DISTINCT](#)
- [SQL WHERE](#)

