

[Catégories d'articles →](#)

Article

Commandes SQL

Glossaire des commandes SQL couramment utilisées.



Contexte

SQL, **S** tructured **Q** uery **L** anguage, est un langage de programmation conçu pour gérer des données stockées dans des bases de données relationnelles. SQL fonctionne à l'aide d'instructions simples et déclaratives. Cela garantit l'exactitude et la sécurité des données et contribue à maintenir l'intégrité des bases de données, quelle que soit leur taille.

Voici une annexe des commandes couramment utilisées.

Commandes

MODIFIER TABLE

```
ALTER TABLE nom_table  
ADD type de données nom_colonne ;
```

ALTER TABLE vous permet d'ajouter des colonnes à une table dans une base de données.

ET

```
SELECT nom_colonne ( s )  
FROM nom_table  
OÙ colonne_1 = valeur_1  
    ET colonne_2 = valeur_2 ;
```

AND est un opérateur qui combine deux conditions. Les deux conditions doivent être vraies pour

que la ligne soit incluse dans le jeu de résultats.

COMME

```
SELECT nom_colonne AS 'Alias'  
FROM nom_table ;
```

AS est un mot clé en SQL qui vous permet de renommer une colonne ou une table en utilisant un *alias* .

MOY()

```
SELECT AVG ( nom_colonne )  
FROM nom_table ;
```

AVG() est une fonction d'agrégation qui renvoie la valeur moyenne d'une colonne numérique.

COMPRIS ENTRE

```
SELECT nom_colonne ( s )  
FROM nom_table  
WHERE nom_colonne ENTRE valeur_1 ET valeur_2 ;
```

L' BETWEEN opérateur est utilisé pour filtrer le jeu de résultats dans une certaine plage. Les valeurs peuvent être des nombres, du texte ou des dates.

CAS

```
SELECT nom_colonne ,  
       CAS  
       QUAND condition ALORS 'Résultat_1'  
       QUAND condition ALORS 'Résultat_2'  
       AUTREMENT 'Résultat_3'  
FINIR  
FROM nom_table ;
```

CASE Les instructions sont utilisées pour créer différentes sorties (généralement dans l' SELECT instruction). C'est la façon dont SQL gère la logique si-alors.

COMPTER()

```
SELECT COUNT ( nom_colonne )  
FROM nom_table ;
```

COUNT() est une fonction qui prend le nom d'une colonne comme argument et compte le nombre de lignes où la colonne n'est pas NULL .

CRÉER UN TABLEAU

```
CREATE TABLE nom_table (  
    type de données column_1 ,  
    type de données column_2 ,  
    type de données column_3  
) ;
```

CREATE TABLE crée une nouvelle table dans la base de données. Il vous permet de spécifier le nom de la table et le nom de chaque colonne de la table.

EFFACER

```
SUPPRIMER DE nom_table  
WHERE une_colonne = une_valeur ;
```

DELETE Les instructions sont utilisées pour supprimer des lignes d'une table.

PAR GROUPE

```
SELECT nom_colonne , COUNT ( * )  
FROM nom_table  
GROUP BY nom_colonne ;
```

GROUP BY est une clause en SQL qui n'est utilisée qu'avec les fonctions d'agrégation. Il est utilisé en collaboration avec l' SELECT instruction pour organiser des données identiques en groupes.

AYANT

```
SELECT nom_colonne , COUNT ( * )  
FROM nom_table  
GROUP BY nom_colonne  
AYANT COMPTE ( * ) > valeur ;
```

HAVING a été ajouté à SQL car le mot- WHERE clé ne pouvait pas être utilisé avec des fonctions

d'agrégation.

JOINTURE INTERNE

```
SELECT nom_colonne ( s )  
DE table_1  
REJOINDRE table_2  
SUR table_1 . nom_colonne = table_2 . nom_colonne ;
```

Une jointure interne combinera des lignes de différentes tables si la *condition de jointure* est vraie.

INSÉRER

```
INSERT INTO nom_table ( colonne_1 , colonne_2 , colonne_3 )  
VALEURS ( valeur_1 , 'valeur_2' , valeur_3 ) ;
```

INSERT Les instructions sont utilisées pour ajouter une nouvelle ligne à une table.

EST NULL / N'EST PAS NULL

```
SELECT nom_colonne ( s )  
FROM nom_table  
WHERE nom_colonne EST NULL ;
```

IS NULL et IS NOT NULL sont des opérateurs utilisés avec la WHERE clause pour tester les valeurs vides.

AIMER

```
SELECT nom_colonne ( s )  
FROM nom_table  
WHERE nom_colonne LIKE modèle ;
```

LIKE est un opérateur spécial utilisé avec la WHERE clause pour rechercher un modèle spécifique dans une colonne.

LIMITE

```
SELECT nom_colonne ( s )  
FROM nom_table  
  
LIMITE nombre ;
```

LIMIT est une clause qui vous permet de spécifier le nombre maximum de lignes que le jeu de résultats aura.

MAX()

```
SELECT MAX ( nom_colonne )  
FROM nom_table ;
```

MAX() est une fonction qui prend le nom d'une colonne comme argument et renvoie la plus grande valeur de cette colonne.

MIN()

```
SELECT MIN ( nom_colonne )  
FROM nom_table ;
```

MIN() est une fonction qui prend le nom d'une colonne comme argument et renvoie la plus petite valeur de cette colonne.

OU

```
SELECT nom_colonne  
FROM nom_table  
WHERE nom_colonne = valeur_1  
    OU nom_colonne = valeur_2 ;
```

OR est un opérateur qui filtre le jeu de résultats pour n'inclure que les lignes où l'une ou l'autre des conditions est vraie.

COMMANDÉ PAR

```
SELECT nom_colonne  
FROM nom_table  
ORDER BY nom_colonne ASC | DESC ;
```

ORDER BY est une clause qui indique que vous souhaitez trier le jeu de résultats par une colonne particulière, soit par ordre alphabétique, soit par ordre numérique.

JOINT EXTERNE

```
SELECT nom_colonne ( s )
```

```
SELECT nom_colonne , ... ,  
DE table_1  
Table JOIN GAUCHE_2  
SUR table_1 . nom_colonne = table_2 . nom_colonne ;
```

Une jointure externe combinera des lignes de différentes tables même si la condition de jointure n'est pas remplie. Chaque ligne de la table de *gauche* est renvoyée dans le jeu de résultats, et si la condition de jointure n'est pas remplie, NULL les valeurs sont utilisées pour remplir les colonnes de la table de *droite* .

ROND()

```
SELECT ROUND ( nom_colonne , entier )  
FROM nom_table ;
```

ROUND() est une fonction qui prend un nom de colonne et un entier comme arguments. Il arrondit les valeurs de la colonne au nombre de décimales spécifié par l'entier.

SÉLECTIONNER

```
SELECT nom_colonne  
FROM nom_table ;
```

SELECT Les instructions sont utilisées pour extraire des données d'une base de données. Chaque requête commencera par SELECT .

SÉLECTIONNER DISTINCT

```
SELECT DISTINCT nom_colonne  
FROM nom_table ;
```

SELECT DISTINCT spécifie que l'instruction va être une requête qui renvoie des valeurs uniques dans la ou les colonnes spécifiées.

SOMME

```
SELECT SUM ( nom_colonne )  
FROM nom_table ;
```

SUM() est une fonction qui prend le nom d'une colonne comme argument et renvoie la somme de toutes les valeurs de cette colonne.

METTRE À JOUR

```
UPDATE nom_table  
SET une_colonne = une_valeur  
WHERE une_colonne = une_valeur ;
```

UPDATE Les instructions vous permettent de modifier les lignes d'une table.

où



```
WHERE nom_colonne valeur de l'opérateur ;
```

WHERE est une clause qui indique que vous souhaitez filtrer le jeu de résultats pour inclure uniquement les lignes où la *condition* suivante est vraie.

AVEC

```
AVEC nom_temporaire AS (  
  SÉLECTIONNER *  
  FROM nom_table )  
SÉLECTIONNER *  
FROM nom_temporaire  
WHERE nom_colonne valeur de l'opérateur ;
```

WITH La clause permet de stocker le résultat d'une requête dans une table temporaire à l'aide d'un alias. Vous pouvez également définir plusieurs tables temporaires à l'aide d'une virgule et avec une instance du mot- WITH clé.

La WITH clause est également connue sous le nom d'expression de table commune (CTE) et de factorisation de sous-requête.

En savoir plus sur Codecademy

Chemin De Compétence

Analyser les données avec SQL

○○ Convient aux débutants, 15 Cours

Chemin De Compétence

Analyser les données d'entreprise avec SQL

○○ Convient aux débutants, 15 Cours