Les requêtes SQL

Kevin Terrier

SELECT

SELECT = Lecture

La requête SELECT va aller chercher des données.

Exemple : SELECT * FROM veste WHERE couleur_veste = 'rouge'

Sélectionne tout dans la table veste, où le champ couleur_veste est égal à rouge.

On peut poser plusieurs conditions :

Exemple : SELECT * FROM veste WHERE couleur_veste = 'rouge'
AND
prix_veste < 20
AND
matiere_veste LIKE 'coton'

INSERT

Insert = Insertion

La requête INSERT insère des données en bdd.

Exemple :INSERT INTO iphone
VALUES (null, 750, Puce A14 , large 6.7")

Résultat:

Table iphone

ID	Prix	Processeur	Ecran
1	750	Puce A14	large 6.7"

Update

Update = Mettre à jour

La requête Update modifie des données en bdd.

Exemple :UPDATE iphone SET (
prix_iphone = 850,
processeur_iphone = 'Puce A15 ',
ecran_iphone = 'large 6.1\"'
WHERE id_iphone = 1
)

Résultat:

Table iphone

id_iphone	prix_iphone	processeur_iphone	ecran_iphone
1	850	Puce A15	large 6.1"

DELETE

Delete = Suppression

La requête Delete supprime des données en bdd.

Exemple : DELETE FROM iphone WHERE 'id_iphone' = 5

Résultat:

Table iphone

ID	Prix	Processeur	Ecran

A retenir

Il y à 4 types de requête SQL :

SELECT
INSERT
UPDATE
DELETE

SELECT

Comment cela fonctionne ?
Les ordres s'écrivent tout en majuscules

SELECT * FROM table

Deux types de sélection :
SELECT * : Sélectionne tous les champs de la ligne (* = tout)

SELECT nom_fruits, prix_fruits, origine_fruits FROM fruits
Selectionne uniquement les champs nom, prix, origine de la table fruits

le nom des champs doivent être indiqués exactement comme dans la table, et séparés par une virgule. Le dernier champ n'a pas besoin de virgule.

SELECT

SELECT nom_fruits, prix_fruits, origine_fruits FROM fruits

Pourquoi choisir quelques champs et ne pas tout prendre systématiquement ?

Pour deux raisons :

1- C'est une bonne habitude pour économiser de prendre des ressources serveurs inutiles.

2- Dans le cadre d'une table utilisateur, est il judicieux d'aller chercher des données sensibles (mot de passe, téléphone ,cb si on souhaite juste voir un nom et un avatar ?

Ce qu'il faut retenir :
Si vous avez besoin des données de 1, 2, 3 où 4 champs, où dans une table contenant des données sensibles, privilégiez la requête contenant spécifiquement les champs nécéssaires.

Sur une requêter avec beaucoup d'infos et / où sans données sensibles, privilégiez un SELECT *

SELECT

Les conditions
SELECT * FROM fruits WHERE prix_fruits <= 10

Les conditions
Ici le WHERE permet de rechercher une info précise.
Les fruits dont le prix est inférieur ou égale à 10.

SELECT

Les conditions
SELECT * FROM fruits WHERE prix_fruits <= 10 AND origine_fruits AND nom_fruits
= 'Pomme'

Slectionne tout dans la table fruits où le prix est plus petit ou égal à 10 et où le champ nom_fruits ressemble à Pomme

Ici le WHERE permet de rechercher une info précise, le AND ajoute un critère de

SELECT LIKE

Exemple précédent
SELECT * FROM fruits WHERE prix_fruits <= 10 AND nom_fruits
= 'Pomme'

Cette syntaxe fonctionne mais une autre serait plus appropriée :

Meilleure syntaxe :
SELECT * FROM fruits WHERE prix_fruits <= 10 AND nom_fruits
LIKE 'Pomme'

Sur les chaînes de caractères, LIKE est plus approprié que égal, car il ne vérifie pas la casse (Majuscule / minuscule.)

SELECT LIKE

Petite astuce :

Rajouter des '% %' avant et après les données recherchées permet de rechercher peu importe où l'expression se trouve dans un mot où un texte.

Exemple : SELECT * FROM user WHERE prenom_user LIKE '%An%'
Retournera:

Marianne
Lauranne
Suzanne
etc...

alors que SELECT * FROM user WHERE prenom_user LIKE 'An' ne retourne que les prénoms commençant strictement par An.

Autrement dit : Like '%données%' signifie :
Comme le mot données peu importe ce qu'il y a avant et après.