

HUMAN**booster**



VOTRE SOLUTION COMPETENCE

Bases de données relationnelles

Conception • Utilisation

Administration • Optimisation

Vues (Views)



Qu'est-ce que c'est ? À quoi ça sert ?

- Les vues représentent des requêtes SELECT et sont interrogeables de la même manière qu'une table
- Elle font pleinement partie de la structure de la base de données
- Elles peuvent être créées, modifiées et supprimées à l'aide de **CREATE**, **ALTER** et **DROP**
- Elles permettent de mémoriser des requêtes SELECT généralement complexes et devant être réutilisées souvent
- Une fois créées, elles peuvent être utilisées dans des requêtes SELECT, comme des tables (on parle de tables virtuelles, vu qu'elles n'occupent pas d'espace de stockage)
- Il est également possible de réaliser des jointures avec

À quoi ça ressemble ?

```
CREATE VIEW NomVue AS
SELECT M.intitule AS matiere,
       CONCAT(prenom, ' ', nom) AS formateur,
       ROUND(AVG(note), 2) AS moyenne
FROM Examen E
JOIN Matiere M ON E.idMatiere = M.idMatiere
JOIN Formateur F ON M.idFormateur = F.idFormateur
WHERE estRattrapage = 0 AND dateExamen < CURDATE()
GROUP BY M.intitule, CONCAT(prenom, ' ', nom)
ORDER BY moyenne
```

*Note : avec MySQL, il n'est pas possible de créer de vues à partir d'un « SELECT * »*

L'utilisation d'alias « simples »
(sans espaces & sans accents)
dans la requête SELECT facilitera
l'utilisation ultérieure de la vue

Comment les créer dans PhpMyAdmin ?

The screenshot shows the PhpMyAdmin interface. On the left, the 'Exécuter une ou des requêtes SQL sur la base de données' panel contains a SQL query: `1 SELECT C.nom, COUNT(*) AS qte`, `2 FROM Vetement V`, `3 JOIN Categorie C ON V.idCategorie = C.idCategorie`, `4 WHERE prix > 40`, `5 GROUP BY C.nom`, `6 ORDER BY COUNT(*)`. Below the query is an 'Exécuter' button. The results panel shows a table with columns 'nom' and 'qte', containing 7 rows of clothing items and their counts. At the bottom, there are links for 'Éditer', 'Expliquer SQL', 'Créer le code source PHP', and 'Actualiser'. On the right, the 'Créer une vue' dialog is open, showing the same SQL query in the 'AS' field. The 'Nom de la vue' field is filled with 'VueCat'. At the bottom of the dialog are 'Exécuter' and 'Fermer' buttons. Arrows point from the instructions to these buttons and the 'Créer une vue' link in the bottom panel.

Exécuter une ou des requêtes SQL sur la base de données

```
1 SELECT C.nom, COUNT(*) AS qte
2 FROM Vetement V
3 JOIN Categorie C ON V.idCategorie = C.idCategorie
4 WHERE prix > 40
5 GROUP BY C.nom
6 ORDER BY COUNT(*)
```

Exécuter

✓ Affichage des lignes 0 - 6 (total de 7, traitement en 0,0004 seconde(s).)

```
SELECT C.nom, COUNT(*) FROM Vetement V JOIN Categorie C ON V.idCategorie = C.idCategorie WHERE prix > 40 GROUP BY C.nom ORDER BY COUNT(*)
```

[Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

nom	qte
Pantalon	1
Short	1
T-shirt	2
Chaussette	2
Jupe	3
Chapeau	4
Chaussure	8

☐ Tout afficher | Nombre de lignes : 25

Opérations sur les résultats de la requête

Imprimer Copier dans le presse-papiers Exporter Afficher le graphique Créer une vue

Créer une vue

Détails

OR REPLACE ☐

ALGORITHM UNDEFINED

Créateur

SQL SECURITY

Nom de la vue VueCat

Noms de colonnes

AS

```
1 SELECT C.nom, COUNT(*) AS qte
2 FROM Vetement V
3 JOIN Categorie C ON V.idCategorie = C.idCategorie
4 WHERE prix > 40
5 GROUP BY C.nom
6 ORDER BY COUNT(*)
```

WITH CHECK OPTION

Exécuter Fermer

1. Rédigez une requête SELECT, puis exécutez-là
2. En bas, cliquez-ici
3. Nommez la vue, puis cliquez sur « Exécuter »

(vous pouvez ensuite l'éditer et la supprimer depuis l'arborescence de gauche)

Vues

Comment utiliser une vue

- Une vue peut être utilisée très simplement, de la même manière qu'une table dans une requête SELECT :

```
SELECT * FROM NomVue
```

- À ce titre, il est donc possible d'y effectuer un WHERE, un ORDER BY, un GROUP BY, etc.

```
SELECT * FROM NomVue WHERE moyenne > 12
```

- Il est également possible de faire des jointures entre la vue et d'autres tables, dès lors que la vue possède au moins une colonne représentant une clé étrangère (mais les performances peuvent ne pas être au rendez-vous).

Avantages

- Les vues permettent de mémoriser et réutiliser une requête SELECT longue à rédiger dont on a besoin souvent
- Elles n'occupent pas d'espace de stockage
- Elles sécurisent la base de données en limitant l'accès direct aux tables
- Elles peuvent permettre de gérer une rétrocompatibilité (si d'anciens logiciels utilisent une table qui n'existe plus, une vue peut venir substituer cette table).

Limites

- Dans le cas général, les vues ne permettent pas de faire des requêtes d'écritures
- Lorsqu'elles sont utilisées en grand nombre, elles peuvent rigidifier l'évolution de la structure de la base de données (la requête mémorisée par la vue peut ne plus fonctionner)
- Utilisées dans des jointures (entre elles ou avec des tables contenant beaucoup de données), les performances peuvent ne pas être à la hauteur

HUMAN **booster**

●● VOTRE SOLUTION COMPETENCE

Tél. 04 73 24 93 11 – contact@humanbooster.com
www.humanbooster.com

●● Clermont-Ferrand ●● Montpellier ●● Lyon