



Ressources **informatiques**

PHP et MySQL

Maîtrisez le développement
d'un site Web dynamique et interactif

3^e édition

Olivier HEURTEL

Fichiers complémentaires
à télécharger



Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI37PH5MY** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1 Introduction

- 1. Objectifs de l'ouvrage 15
- 2. Bref historique de PHP et MySQL 16
 - 2.1 PHP 16
 - 2.2 MySQL 16
- 3. Où se procurer PHP et MySQL..... 17
- 4. Conventions d'écriture 18
 - 4.1 PHP 18
 - 4.2 MySQL 19

Chapitre 2 Introduction à MySQL

- 1. Introduction aux bases de données relationnelles 21
 - 1.1 Concepts 21
 - 1.2 Principes de conception d'une base de données 22
- 2. Travailler avec MySQL 26
 - 2.1 Administration du serveur MySQL..... 26
 - 2.2 Interface en ligne de commande 26
 - 2.3 MySQL Workbench 29
 - 2.4 phpMyAdmin 34
 - 2.5 Fichier de configuration 39
- 3. Installer notre base de données de démonstration 39
- 4. Apprendre les bases du langage SQL 46
 - 4.1 Types de données 46
 - 4.1.1 Types chaîne de caractères 46
 - 4.1.2 Types numériques 47
 - 4.1.3 Type booléen 49

2 _____ PHP et MySQL

Maîtrisez le développement d'un site Web dynamique et interactif

4.1.4	Types date et heure	49
4.1.5	Types pour les données binaires	49
4.2	Nom des objets	50
4.3	Valeurs littérales	51
4.3.1	Chaîne.	51
4.3.2	Nombre.	52
4.3.3	Date, heure, date/heure	52
4.3.4	Booléen	53
4.4	Expression	53
4.5	Valeur NULL	53
4.6	Variables	53
4.6.1	Variables utilisateurs	53
4.6.2	Variables système.	54
4.7	Commentaires.	55
5.	Exécuter des requêtes SQL simples.	56
5.1	Le mode SQL du serveur	56
5.2	Obtenir des informations	59
5.3	Afficher les erreurs et les alertes.	60
5.4	Lire les données	61
5.4.1	Syntaxe de base de l'ordre SELECT	61
5.4.2	Restreindre le résultat : clause WHERE	66
5.4.3	Trier le résultat : clause ORDER BY	71
5.4.4	Limiter le nombre de lignes : clause LIMIT	73
5.4.5	Lire dans plusieurs tables : jointure	74
5.5	Ajouter des lignes dans une table.	81
5.6	Modifier des lignes dans une table.	88
5.7	Supprimer des lignes dans une table	93
5.8	Exporter et importer des données	95
5.8.1	Exporter des données.	95
5.8.2	Importer des données.	98

Chapitre 3
Utiliser les fonctions MySQL

- 1. Introduction 101
- 2. Fonctions de contrôle 102
- 3. Fonctions de comparaison 106
- 4. Fonctions numériques 108
- 5. Fonctions caractères 112
- 6. Fonctions dates 121
- 7. Fonctions de transtypage et de mise en forme 130
- 8. Fonctions système 135
- 9. Fonctions de chiffrement et de compression 139
- 10. Fonctions d'agrégat 142

Chapitre 4
Construire une base de données dans MySQL

- 1. Créer et supprimer une base de données 147
- 2. Gérer les utilisateurs et les droits 148
 - 2.1 Vue d'ensemble 148
 - 2.2 Gérer les utilisateurs 149
 - 2.2.1 Créer des utilisateurs 149
 - 2.2.2 Supprimer des utilisateurs 151
 - 2.2.3 Modifier le mot de passe des utilisateurs 151
 - 2.3 Gérer les droits des utilisateurs 153
 - 2.3.1 Attribuer des droits aux utilisateurs 153
 - 2.3.2 Lister les droits d'un utilisateur 158
 - 2.3.3 Révoquer des droits d'un utilisateur 158
- 3. Gérer les tables 159
 - 3.1 Créer une table 159
 - 3.2 Créer une table par copie 165
 - 3.3 Renommer une table 167
 - 3.4 Modifier la structure d'une table 167
 - 3.5 Supprimer une table 171

4 **PHP et MySQL**

Maîtrisez le développement d'un site Web dynamique et interactif

4.	Utiliser les clés et les index	172
4.1	Clé primaire ou unique	172
4.1.1	Définition	172
4.1.2	Gestion	172
4.2	Index	175
4.2.1	Définition	175
4.2.2	Gestion	175
4.2.3	Considérations	177
4.3	Clé étrangère	179
4.3.1	Définition	179
4.3.2	Gestion	179
5.	Utiliser des vues	182
5.1	Définition	182
5.2	Gestion	183
6.	Obtenir des informations sur les bases de données	185
6.1	La commande SHOW	185
6.2	La base de données INFORMATION_SCHEMA	187
7.	Exporter et importer une base de données	188

Chapitre 5 **Techniques avancées avec MySQL**

1.	Grouper les données	191
2.	Utiliser des sous-requêtes	197
2.1	Introduction	197
2.2	Sous-requête scalaire	197
2.3	Comparaison avec une sous-requête	199
2.4	Sous-requête corrélée	202
2.5	Sous-requête dans la clause FROM	205
2.6	Insérer des lignes à l'aide d'une sous-requête	206
3.	Réunir le résultat de plusieurs requêtes	207
4.	Gérer les transactions et les accès concurrents	209
4.1	Définition	209
4.2	Gérer les transactions	209
4.3	Annuler une partie d'une transaction	212

4.4	Concurrence d'accès et verrouillage	214
4.4.1	Concurrence d'accès	214
4.4.2	Verrouiller des tables	217
4.4.3	Verrou mortel	218
5.	Effectuer des recherches à l'aide des expressions rationnelles	218
6.	Effectuer des recherches en texte intégral	223
6.1	Principes	223
6.2	Création de l'index FULLTEXT	223
6.3	Effectuer une recherche en texte intégral	224
6.3.1	Recherche classique	224
6.3.2	Recherche en mode booléen	226
6.3.3	Recherche avec extension de requête	228
6.4	Réglage de la recherche en texte intégral	229
7.	Développer des programmes stockés	230
7.1	Introduction	230
7.2	Gestion des droits	230
7.3	Gestion des programmes stockés	231
7.4	Exécuter un programme stocké	234
7.5	Structure du langage	235
7.5.1	Bloc BEGIN END	235
7.5.2	Les variables	235
7.5.3	Intégration d'ordres SQL	237
7.5.4	Les structures de contrôle	238
7.5.5	La gestion des erreurs	243
7.5.6	Les curseurs	253
7.5.7	Récursivité	256
8.	Développer des triggers	256
8.1	Définition	256
8.2	Gestion des triggers	257
8.3	Considérations sur l'utilisation des triggers	260
8.3.1	Restrictions	260
8.3.2	Résultat en cas d'erreur	260

Chapitre 6 **Introduction à PHP**

1. Qu'est-ce que PHP ?	263
2. Structure de base d'une page PHP	265
2.1 Les balises PHP	265
2.2 La fonction echo	266
2.3 Séparateur d'instructions	267
2.4 Commentaires	267
2.5 Mixer du PHP et du HTML	268
2.6 Règles de nommage	271
3. Configuration de PHP	271
3.1 Le fichier de configuration php.ini	271
3.2 Informations sur la configuration	272
3.3 Jeu de caractères	275
4. Les bases du langage PHP	275
4.1 Constantes	275
4.1.1 Définition	275
4.1.2 Portée	277
4.2 Variables	277
4.2.1 Initialisation et affectation	277
4.2.2 Portée et durée de vie	279
4.2.3 Variable dynamique (ou variable variable)	280
4.3 Types de données	280
4.3.1 Types de données disponibles	280
4.3.2 Types de données scalaires	281
4.3.3 Types de données spéciaux	287
4.4 Tableaux	287
4.4.1 Définition	287
4.4.2 Création	289
4.4.3 Manipulation	294
4.5 Opérateurs	297
4.5.1 L'opérateur d'affectation par valeur	297
4.5.2 L'opérateur d'affectation par référence	298
4.5.3 Les opérateurs arithmétiques	299
4.5.4 L'opérateur de chaîne	300
4.5.5 Les opérateurs combinés	301

- 4.5.6 Les opérateurs de comparaison 301
- 4.5.7 Les opérateurs logiques 302
- 4.5.8 L'opérateur ternaire 302
- 4.5.9 L'opérateur d'union NULL 303
- 4.5.10 L'opérateur de comparaison combinée 304
- 4.5.11 Précédence des opérateurs 305
- 4.6 Structures de contrôle 306
 - 4.6.1 La structure if 306
 - 4.6.2 La structure switch 308
 - 4.6.3 La structure while 310
 - 4.6.4 La structure do ... while 312
 - 4.6.5 La structure for 313
 - 4.6.6 Les instructions continue et break 315
- 4.7 Inclure un fichier 316
 - 4.7.1 Fonctionnement 316
 - 4.7.2 Utilisation 318
- 4.8 Interrompre le script 319

Chapitre 7
Utiliser les fonctions PHP

- 1. Préambule 321
- 2. Manipuler les constantes, les variables et les types de données 321
 - 2.1 Constantes 321
 - 2.2 Variables 323
 - 2.3 Types de données 329
 - 2.3.1 Conversions 329
 - 2.3.2 Fonctions utiles 332
- 3. Manipuler les tableaux 337
- 4. Manipuler les nombres 349
- 5. Manipuler les chaînes de caractères 355
- 6. Utiliser les expressions rationnelles 371
 - 6.1 Introduction 371
 - 6.2 Structure d'une expression rationnelle 371
 - 6.3 Fonctions 380

8 **PHP et MySQL**

Maîtrisez le développement d'un site Web dynamique et interactif

7. Manipuler les dates	385
8. Générer un identifiant unique	396
9. Manipuler les fichiers sur le serveur	398
9.1 Fonctions utiles	398
9.2 Exemple d'utilisation	405
10. Envoyer un courrier électronique	405
10.1 Vue d'ensemble	405
10.2 Envoyer un message texte sans pièce jointe	406
10.3 Envoyer un message au format MIME	408
10.3.1 Préambule	408
10.3.2 Message au format HTML	409
10.3.3 Message avec pièce jointe	412
11. Manipuler les en-têtes HTTP	415

Chapitre 8

Écrire des fonctions et des classes PHP

1. Fonctions	417
1.1 Introduction	417
1.2 Déclaration et appel	417
1.3 Paramètres	423
1.3.1 Syntaxe	423
1.3.2 Valeur par défaut	424
1.3.3 Déclaration du type de données	425
1.3.4 Passage par référence	427
1.3.5 Liste variable de paramètres	428
1.4 Considérations sur les variables utilisées dans les fonctions	430
1.4.1 Variable locale - globale	430
1.4.2 Variable statique	432
1.5 Les constantes et les fonctions	434
1.6 Récursivité	434
1.7 Fonction anonyme	436
1.8 Fonction générateur	438

2. Classes	441
2.1 Concept	441
2.2 Définir une classe	441
2.3 Instancier une classe	444
2.4 Héritage	446
2.5 Autres fonctionnalités sur les classes	449
2.5.1 Classe ou méthode abstraite	449
2.5.2 Classe ou méthode finale	450
2.5.3 Interface	450
2.5.4 Attribut ou méthode statique - Constante de classe	451
2.5.5 Traits	453
2.6 Classe anonyme	456
2.7 Exceptions	457
3. Les espaces de noms	460

Chapitre 9

Gérer les erreurs dans un script PHP

1. Vue d'ensemble	465
2. Les messages d'erreur PHP	466
3. Les fonctions de gestion des erreurs	469

Chapitre 10

Gérer les formulaires et les liens avec PHP

1. Vue d'ensemble	485
1.1 Introduction	485
1.2 Les liens	485
1.3 Les formulaires	488
1.3.1 Petit rappel sur les formulaires	488
1.3.2 Construire un formulaire dynamiquement	490
1.3.3 Traiter un formulaire à l'aide d'un script PHP	495
1.4 Récupérer les données d'une URL ou d'un formulaire	499

2.	Récupérer les données passées par l'URL	502
2.1	Considérations	502
2.1.1	Que se passe-t-il si deux paramètres portent le même nom ? .	502
2.1.2	Utiliser un tableau pour passer des données dans l'URL	502
2.2	Transmettre des caractères spéciaux	503
3.	Récupérer les données saisies dans le formulaire	506
3.1	Considérations	506
3.1.1	Que se passe-t-il si deux zones portent le même nom ?	506
3.1.2	Que se passe-t-il s'il y a deux formulaires dans la page HTML ?	506
3.1.3	Utiliser un tableau pour récupérer les données saisies	507
3.1.4	Passer des informations dans une zone de formulaire cachée .	508
3.2	Les différents types de zone	510
3.2.1	Vue d'ensemble	510
3.2.2	Zone contenant du texte	512
3.2.3	Groupe de boutons radio	513
3.2.4	Case à cocher	513
3.2.5	Liste à sélection unique	516
3.2.6	Liste à sélection multiple	517
3.2.7	Bouton de validation	518
3.2.8	Bouton image	520
3.2.9	Bouton "reset" ou "button"	521
3.3	Synthèse	521
4.	Contrôler les données récupérées	523
4.1	Vue d'ensemble	523
4.2	Vérifications classiques	524
4.2.1	Nettoyage des espaces superflus	524
4.2.2	Donnée obligatoire	524
4.2.3	Longueur maximum d'une chaîne	524
4.2.4	Caractères autorisés pour une chaîne - Format	525
4.2.5	Validité d'une date - Plage de valeurs	525
4.2.6	Validité d'un nombre - Plage de valeurs	527
4.2.7	Validité d'une adresse e-mail	528
5.	Les problèmes sur les données récupérées	529

- 6. Utilisation des filtres 536
 - 6.1 Principes 536
 - 6.2 Application aux formulaires. 544
- 7. Aller sur une autre page 546
- 8. Échanger un fichier entre le client et le serveur. 552
 - 8.1 Vue d'ensemble 552
 - 8.2 Envoyer un fichier depuis le client (upload) 552
 - 8.3 Télécharger un fichier à partir du serveur (download). 558

Chapitre 11

Accéder à une base de données MySQL

- 1. Introduction. 565
 - 1.1 Vue d'ensemble 565
 - 1.2 Quelle extension utiliser pour accéder à MySQL ? 566
 - 1.3 La notion de fetch. 566
- 2. Utilisation de l'extension MySQLi 567
 - 2.1 Introduction 567
 - 2.2 Connexion et déconnexion 568
 - 2.2.1 Connexion 568
 - 2.2.2 Déconnexion 569
 - 2.2.3 Obtenir des informations sur le serveur MySQL. 570
 - 2.2.4 Définir le jeu de caractères du client. 570
 - 2.2.5 Obtenir des informations en cas d'erreur de connexion 570
 - 2.2.6 Exemple 571
 - 2.3 Sélectionner une base de données 572
 - 2.4 Utiliser des requêtes non préparées 573
 - 2.4.1 Vue d'ensemble 573
 - 2.4.2 Exécuter une requête 573
 - 2.4.3 Connaître le nombre de lignes dans le résultat
d'une requête de lecture. 575
 - 2.4.4 Extraire le résultat d'une requête de lecture. 576
 - 2.4.5 Obtenir des informations sur le résultat
d'une requête de mise à jour 580
 - 2.4.6 Gérer les erreurs 582

2.5	Utiliser des requêtes préparées	584
2.5.1	Vue d'ensemble	584
2.5.2	Préparer une requête	585
2.5.3	Lier des variables PHP aux paramètres de la requête	586
2.5.4	Exécuter la requête préparée	587
2.5.5	Lier des variables PHP aux colonnes du résultat d'une requête de lecture	588
2.5.6	Extraire le résultat d'une requête de lecture	590
2.5.7	Utiliser un résultat stocké	591
2.5.8	Obtenir des informations sur le résultat d'une requête de mise à jour	594
2.5.9	Gérer les erreurs	595
2.5.10	Fermer une requête préparée	597
2.6	Gérer les transactions	597
2.7	Appeler un programme stocké	600
2.7.1	Procédure stockée	600
2.7.2	Fonction stockée	604
2.8	Utiliser les types de données BLOB	606
2.8.1	Insertion ou modification	606
2.8.2	Lecture	609
3.	Migrer de l'extension MySQL à l'extension MySQLi	611
4.	PHP Data Objects (PDO)	612
5.	Gestion des apostrophes dans le texte des requêtes	614
6.	Exemples d'intégration dans des formulaires	618
6.1	Vue d'ensemble	618
6.2	Formulaire de saisie en liste	619
6.3	Formulaire de recherche	624
6.4	Formulaire de saisie	628

Chapitre 12 Gérer les sessions

1. Description du problème.....	637
2. Authentification	638
2.1 Vue d'ensemble.....	638
2.2 Saisie de l'identification	638
2.2.1 Identification par formulaire	638
2.2.2 Identification par authentification HTTP.....	641
2.3 Vérifier l'identification saisie.....	643
3. Utiliser des cookies	644
3.1 Principe	644
3.2 Application à la gestion des sessions.....	651
4. Utiliser la gestion des sessions de PHP.....	651
4.1 Principes	651
4.2 Mise en œuvre.....	652
4.3 Créer soi-même la transmission de l'identifiant de session.....	666
4.3.1 Description du problème.....	666
4.3.2 Solution	669
4.4 Quelques directives de configuration supplémentaires	672
4.5 Exemple d'application.....	673
4.6 Remarque et conclusion.....	676
5. Conserver des informations d'une visite à une autre	679
6. Petite synthèse sur les variables GPCS (Get/Post/Cookie/Session).....	685

Annexes

1. Variables PHP prédéfinies.....	689
2. Constantes PHP prédéfinies	691
3. Exemples complémentaires.....	692
3.1 Introduction	692
3.2 Lire un document XML	692
3.3 Générer un document PDF.....	697
3.4 Générer une image	701

Chapitre 5

Techniques avancées avec MySQL

1. Grouper les données

Parfois, vous pouvez avoir besoin de calculer une valeur pour un niveau de regroupement :

- Chiffre d'affaires cumulé par région
- Salaire moyen par département

Pour ce genre d'interrogation, vous allez pouvoir utiliser des fonctions d'agrégat (`SUM`, `AVG`, etc.) et regrouper les données grâce à la clause `GROUP BY` ; en complément, le résultat final, après regroupement, peut être restreint avec la clause `HAVING`.

Syntaxe

```
SELECT expression[,...] | *  
FROM nom_table  
[WHERE conditions]  
GROUP BY expression [ASC | DESC] [,...]  
[HAVING conditions]  
[ORDER BY expression [ASC | DESC][,...]]  
[LIMIT [offset,] nombre_lignes]
```

La clause `GROUP BY` s'intercale entre les clauses `WHERE` et `ORDER BY` (si elles sont présentes). Elle spécifie les expressions utilisées pour effectuer le regroupement.

expression peut être :

- Une colonne
- Une expression basée sur des colonnes
- Un alias de colonne
- Un numéro correspondant à la position d'une expression de la clause `SELECT` (syntaxe déconseillée et obsolète)

Par défaut, le résultat de la requête est trié en ordre croissant sur les différentes expressions de la clause `GROUP BY` ; l'ordre du tri de chaque niveau de regroupement peut être défini explicitement avec les options `ASC` et `DESC`. Un tri complètement différent peut être spécifié grâce à la clause `ORDER BY`. Depuis la version 5.7, se baser sur le tri implicite provoqué par la clause `GROUP BY` est déprécié. À la place, il est donc recommandé d'utiliser une clause `ORDER BY` explicite.

La plupart du temps, vous mettrez dans la clause `GROUP BY` toutes les expressions de la clause `SELECT` qui ne comportent pas de fonction d'agrégat. C'est le fonctionnement habituel pour respecter le SQL standard.

Exemples

```
mysql> -- Nombre de livres par collection.
```

```
mysql> SELECT id_collection, COUNT(*)
        -> FROM livre
        -> GROUP BY id_collection;
```

```
+-----+-----+
| id_collection | COUNT(*) |
+-----+-----+
|             1 |         7 |
|             3 |         1 |
|             4 |         1 |
|             5 |         1 |
+-----+-----+
```

```
4 rows in set (0.00 sec)
```

```
mysql> -- Nombre de livres et nombre moyen de pages par collection
```

```
mysql> -- (avec le nom de la collection au lieu de l'identifiant).
```

```
mysql> SELECT
        -> col.nom collection,
        -> COUNT(liv.id) nb_livres,
        -> ROUND(AVG(liv.nombre_pages)) nb_pages_moy
        -> FROM
        -> livre liv JOIN collection col
        -> ON (liv.id_collection = col.id)
        -> GROUP BY
        -> col.nom
```



```
-> ORDER BY
->   nb_livres; -- tri sur le nombre de livres
+-----+-----+-----+
| collection          | nb_livres | nb_pages_moy |
+-----+-----+-----+
| Coffret Technique   |          1 |          1080 |
| Les TP Informatiques |          1 |           302 |
| Epsilon             |          1 |           552 |
| Ressources Informatiques |          7 |           567 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

mysql> -- Nombre de livres par tranche d'années et collection.

```
mysql> SELECT
->   CASE
->     WHEN annee_parution BETWEEN 2005 AND 2010
->     THEN '2005-2010'
->     WHEN annee_parution BETWEEN 2011 AND 2016
->     THEN '2011-2016'
->   END tranche_annee,
->   col.nom collection,
->   COUNT(liv.id) nb_livres
-> FROM
->   livre liv JOIN collection col
->   ON (liv.id_collection = col.id)
-> GROUP BY -- utilisation des alias de colonne
->   tranche_annee,
->   collection;
+-----+-----+-----+
| tranche_annee | collection          | nb_livres |
+-----+-----+-----+
| 2005-2010     | Les TP Informatiques |          1 |
| 2005-2010     | Ressources Informatiques |          1 |
| 2011-2016     | Coffret Technique    |          1 |
| 2011-2016     | Epsilon              |          1 |
| 2011-2016     | Ressources Informatiques |          6 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Avec MySQL, vous pouvez avoir dans la clause `SELECT` des expressions qui n'utilisent pas de fonction d'agrégat et qui ne sont pas reprises dans la clause `GROUP BY`.

Exemple

```
mysql> SELECT
->   col.nom collection, -- non présent dans le GROUP BY
->   col.prix_ht, -- non présent dans le GROUP BY
->   COUNT(liv.id) nb_livres
-> FROM
->   livre liv JOIN collection col
->   ON (liv.id_collection = col.id)
-> GROUP BY
->   col.id;
```

```
+-----+-----+-----+
| collection          | prix_ht | nb_livres |
+-----+-----+-----+
| Ressources Informatiques | 24.44 | 6 |
| TechNote           | 10.48 | 1 |
| Les TP Informatiques  | 25.59 | 1 |
| Coffret Technique    | 46.45 | 1 |
+-----+-----+-----+
```

```
4 rows in set (0.00 sec)
```

Cette possibilité de MySQL est intéressante et permet de simplifier l'écriture de la requête et d'améliorer les performances lorsque l'expression concernée est unique pour chaque groupe (c'est le cas sur l'exemple précédent où le nom et le prix de la collection sont uniques pour chaque identifiant de collection). Par contre, il est déconseillé d'utiliser cette possibilité si l'expression concernée n'est pas unique dans le groupe ; MySQL va retourner n'importe quelle valeur pour le groupe et le résultat obtenu n'a en général pas vraiment de sens. Si le mode SQL `ONLY_FULL_GROUP_BY` est actif (c'est le cas par défaut depuis la version 5.7.5), MySQL génère une erreur si les clauses `SELECT`, `HAVING` ou `ORDER BY` référencent des colonnes qui n'utilisent pas de fonction d'agrégat ou qui ne sont pas présentes dans la clause `GROUP BY`, et qui ne sont pas fonctionnellement dépendantes des colonnes présentes dans la clause `GROUP BY`.

Exemple

```
mysql> SELECT
->   col.nom collection,
->   liv.annee_parution, -- pas unique dans le groupe !
->   COUNT(liv.id) nb_livres
-> FROM
->   livre liv JOIN collection col
->   ON (liv.id_collection = col.id)
-> GROUP BY
->   col.nom;
```

```
ERROR 1055 (42000): Expression #2 of SELECT list is not in GROUP BY clause and
contains nonaggregated column 'eni.liv.annee_parution' which is not functionally
dependent on columns in GROUP BY clause; this is incompatible with
sql_mode=only_full_group_by
```

Dans ce genre de situation, il peut être intéressant d'utiliser les fonctions `MIN` ou `MAX` pour obtenir une valeur spécifique "contrôlée" de l'expression ("le plus grand", "le plus petit", "le premier", "le dernier", etc.).

Exemple

```
mysql> SELECT
->   col.nom collection,
->   MIN(liv.annee_parution) premiere_parution,
->   COUNT(liv.id) nb_livres
-> FROM
->   livre liv JOIN collection col
->   ON (liv.id_collection = col.id)
-> GROUP BY
->   col.nom;
```

collection	premiere_parution	nb_livres
Coffret Technique	2015	1
Epsilon	2016	1
Les TP Informatiques	2007	1
Ressources Informatiques	2008	7

4 rows in set (0.00 sec)

Pour restreindre le résultat final, avec des conditions utilisant les fonctions d'agrégat, vous devez utiliser la clause `HAVING`. En effet, une clause `WHERE` ne peut pas comporter de condition qui utilise une fonction d'agrégat : c'est trop "tôt" dans le traitement de la requête ; les groupes n'ont pas encore été constitués.

Les conditions d'une clause `HAVING` sont semblables à celles d'une clause `WHERE` avec les particularités suivantes :

- elles peuvent utiliser des fonctions d'agrégat ;
- elles peuvent utiliser les alias définis dans la clause `SELECT`.

Exemple

```
mysql> -- Afficher uniquement les années qui ont
mysql> -- au moins deux livres.
mysql> SELECT
->   annee_parution,
->   COUNT(*) nb_livres,
->   ROUND(AVG(nombre_pages)) nb_pages_moy
-> FROM
->   livre
-> GROUP BY
->   annee_parution
```

```
-> HAVING -- utilisation des alias
->   nb_livres > 1;
+-----+-----+-----+
| annee_parution | nb_livres | nb_pages_moy |
+-----+-----+-----+
|           2014 |         2 |           716 |
|           2015 |         2 |           823 |
|           2016 |         2 |           568 |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
mysql> --
mysql> -- Afficher uniquement les années qui ont
mysql> -- au moins deux livres et un nombre moyen
mysql> -- de pages supérieur à 400.
mysql> SELECT
->   annee_parution,
->   COUNT(*) nb_livres,
->   ROUND(AVG(nombre_pages)) nb_pages_moy
-> FROM
->   livre
-> GROUP BY
->   annee_parution
-> HAVING -- utilisation des alias
->   nb_livres > 1
->   AND nb_pages_moy > 400;
+-----+-----+-----+
| annee_parution | nb_livres | nb_pages_moy |
+-----+-----+-----+
|           2014 |         2 |           716 |
|           2015 |         2 |           823 |
|           2016 |         2 |           568 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Pour des raisons de performance, la clause `HAVING` ne doit pas être utilisée pour filtrer des données qui pourraient l'être dans la clause `WHERE`.