

Les fonctions pour les dates et les temps

Temps de lecture : 2 minutes



Présentation des fonctions

MySQL propose un grand nombre de fonctions pour les dates et les temps :

Nom	Description
ADDDATE()	Ajoute des intervalles de temps à une valeur de date
ADDTIME()	Ajoute du temps
CONVERT_TZ()	Convertit d'un fuseau horaire à un autre
CURDATE()	Retourne la date actuelle
CURRENT_DATE(), CURRENT_DATE	Synonymes pour CURDATE()
CURRENT_TIME(), CURRENT_TIME	Synonymes pour CURTIME()
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP	Synonymes pour NOW()
CURTIME()	Retourne l'heure actuelle
DATE()	Extrait la partie date d'une expression date ou datetime
DATE_ADD()	Ajoute des intervalles de temps à une valeur de date
DATE_FORMAT()	Formate la date comme spécifié
DATE_SUB()	Soustrait un intervalle de temps d'une date
DATEDIFF()	Soustrait deux dates
DAY()	Synonyme pour DAYOFMONTH()
DAYNAME()	Retourne le nom du jour de la semaine
DAYOFMONTH()	Retourne le jour du mois (0-31)
DAYOFWEEK()	Retourne l'indice du jour de la semaine
DAYOFYEAR()	Retourne le jour de l'année (1-366)
EXTRACT()	Extrait une partie d'une date
FROM_DAYS()	Convertit un numéro de jour en date
FROM_UNIXTIME()	Formate un timestamp Unix en date

Nom	Description
GET_FORMAT()	Retourne une chaîne de format de date
HOUR()	Extrait l'heure
LAST_DAY	Retourne le dernier jour du mois pour l'argument
LOCALTIME(), LOCALTIME	Synonyme pour NOW()
LOCALTIMESTAMP, LOCALTIMESTAMP()	Synonyme pour NOW()
MAKEDATE()	Crée une date à partir de l'année et du jour de l'année
MAKETIME()	Crée un temps à partir de l'heure, minute, seconde
MICROSECOND()	Retourne les microsecondes de l'argument
MINUTE()	Retourne la minute de l'argument
MONTH()	Retourne le mois de la date passée
MONTHNAME()	Retourne le nom du mois
NOW()	Retourne la date et l'heure actuelles
PERIOD_ADD()	Ajoute une période à une année-mois
PERIOD_DIFF()	Retourne le nombre de mois entre deux périodes
QUARTER()	Retourne le trimestre d'une date
SEC_TO_TIME()	Convertit des secondes en format 'hh:mm:ss'
SECOND()	Retourne la seconde (0-59)
STR_TO_DATE()	Convertit une chaîne en date
SUBDATE()	Synonyme pour DATE_SUB() lorsqu'invoqué avec trois arguments
SUBTIME()	Soustrait des temps
SYSDATE()	Retourne le moment auquel la fonction s'exécute
TIME()	Extrait la portion de temps de l'expression passée
TIME_FORMAT()	Formate comme temps
TIME_TO_SEC()	Retourne l'argument converti en secondes
TIMEDIFF()	Soustrait du temps
TIMESTAMP()	Avec un seul argument, cette fonction retourne l'expression date ou datetime; avec deux arguments, la somme des arguments
TIMESTAMPADD()	Ajoute un intervalle à une expression datetime
TIMESTAMPDIFF()	Retourne la différence de deux expressions datetime, en utilisant les unités spécifiées
TO_DAYS()	Retourne l'argument date converti en jours

Nom	Description
TO_SECONDS()	Retourne l'argument date ou datetime converti en secondes depuis l'An 0
UNIX_TIMESTAMP()	Retourne un timestamp Unix
UTC_DATE()	Retourne la date UTC actuelle
UTC_TIME()	Retourne l'heure UTC actuelle
UTC_TIMESTAMP()	Retourne la date et l'heure UTC actuelles
WEEK()	Retourne le numéro de la semaine
WEEKDAY()	Retourne l'indice du jour de la semaine
WEEKOFYEAR()	Retourne la semaine calendaire de la date (1-53)
YEAR()	Retourne l'année
YEARWEEK()	Retourne l'année et la semaine

Fonctions les plus utilisées

CURDATE() - Retourne la date actuelle.

Par exemple : `SELECT CURDATE();` pourrait retourner `'2024-03-15'`, la date d'aujourd'hui.

CURTIME() - Retourne l'heure actuelle.

Exemple : `SELECT CURTIME();` pourrait donner `'13:45:07'`, représentant l'heure actuelle.

NOW() - Retourne la date et l'heure actuelles.

Exemple : `SELECT NOW();` peut retourner `'2024-02-15 13:45:07'`, indiquant à la fois la date et l'heure actuelles.

DATE() - Extrait la partie date d'une expression date ou datetime.

Exemple : `SELECT DATE('2024-02-15 13:45:07');` retourne `'2024-02-15'`, isolant la partie date de l'expression datetime.

DATE_FORMAT(date, format) - Formate la date comme spécifié.

La chaîne de format peut inclure une variété de spécificateurs qui indiquent quelle partie de la valeur de date doit être affichée et comment. Voici quelques exemples de spécificateurs de format :

- `%Y` : Année sur quatre chiffres.
- `%m` : Mois sous forme de deux chiffres.
- `%d` : Jour du mois sous forme de deux chiffres.
- `%H` : Heure (00 à 23).
- `%i` : Minutes (00 à 59).
- `%s` : Secondes (00 à 59).
- `%a` : Nom abrégé du jour de la semaine.
- `%b` : Nom abrégé du mois.
- `%W` : Nom complet du jour de la semaine.
- `%M` : Nom complet du mois.

Exemple : `SELECT DATE_FORMAT(NOW(), '%W, %M %d, %Y');` pourrait donner `'Vendredi, Février 15, 2024'`, formatant la date actuelle dans un format lisible.

`DATEDIFF()` - Soustrait deux dates et retourne la différence en jours.

Exemple : `SELECT DATEDIFF('2024-02-15', '2024-01-01');` retourne `45`, calculant le nombre de jours entre deux dates.

`TIMESTAMPDIFF()` - Retourne la différence entre deux expressions datetime, en utilisant les unités spécifiées.

Exemple : `SELECT TIMESTAMPDIFF(MONTH, '2024-01-01', '2024-02-15');` donne `1`, montrant le nombre de mois complets entre deux dates.

`ADDDATE()` et `DATE_ADD()` - Ajoutent des intervalles de temps à une valeur de date.

Exemple : `SELECT ADDDATE('2024-02-15', INTERVAL 10 DAY);` retourne `'2024-02-25'`, ajoutant 10 jours à la date spécifiée.

`SUBDATE()` et `DATE_SUB()` - Soustraient un intervalle de temps d'une date.

Exemple : `SELECT SUBDATE('2024-02-15', INTERVAL 1 MONTH);` donne `'2024-01-15'`, soustrayant un mois de la date spécifiée.

UNIX_TIMESTAMP() - Convertit une date en timestamp Unix.

Exemple : `SELECT UNIX_TIMESTAMP('2024-02-15 13:45:07');` pourrait retourner `1708160707`, le nombre de secondes depuis le '1970-01-01 00:00:00' UTC jusqu'à la date et l'heure spécifiées.

Exemples

Trouver les films sortis cette année

Cette requête utilise `CURDATE()` pour obtenir la date actuelle et `YEAR()` pour extraire l'année actuelle. Elle récupère les titres des films (`title`) et leurs dates de sortie (`release_date`) de la table `movie` pour les films sortis cette année.

Il n'y aura ici pas de résultat car la table n'a que des films jusqu'en 2016 comme vu précédemment.

```
SELECT title, release_date
FROM movie
WHERE YEAR(release_date) = YEAR(CURDATE());
```

Calculer l'âge moyen des films par genre

Cette requête combine `YEAR()`, `CURDATE()`, et `AVG()` pour calculer l'âge moyen des films dans chaque genre. Elle joint les tables `movie` et `genre` via une table de liaison pour grouper les résultats par genre.

```
SELECT
    genre_name,
    AVG(YEAR(CURDATE()) - YEAR(release_date)) AS average_age
FROM
    movie
    JOIN
    movie_genres USING (movie_id)
    JOIN
    genre USING (genre_id)
GROUP BY genre_name;
```