

Limiter le nombre de résultats

LIMIT

- **SELECT ... FROM *myTable* ... LIMIT *myNumber* OFFSET *myOffset* ;**

- ne renvoie que *myNumber* enregistrements.
- commence à partir de *myOffset*. Par défaut l'offset vaut 0 et représente le premier résultat.

```
SELECT location , date FROM doughnut_ratings LIMIT 2;
```

```
+-----+-----+
| location           | date           |
+-----+-----+
| Krispy King        | 2007-09-27    |
| Duncan's Donuts   | 2007-08-25    |
+-----+-----+
```

```
mysql> SELECT * FROM doughnut_ratings;
```

```
+-----+-----+-----+-----+-----+-----+
| location    | time          | date           | type           | rating | comments           |
+-----+-----+-----+-----+-----+-----+
| Krispy King | 08:50:00     | 2007-09-27    | plain glazed   | 10    | almost perfect    |
| Duncan's Donuts | 08:59:00     | 2007-08-25    | NULL           | 6     | greasy            |
| Starbuzz Coffee | 07:35:00     | 2007-05-24    | cinnamon cake | 5     | stale, but tasty  |
| Duncan's Donuts | 07:03:00     | 2007-04-26    | jelly          | 7     | not enough jelly  |
+-----+-----+-----+-----+-----+-----+
```

Exemple d'utilisation de LIMIT

```
SELECT location , date FROM doughnut_ratings LIMIT 2 OFFSET 1;
```

```
+-----+-----+
| location | date |
+-----+-----+
| Duncan's Donuts | 2007-08-25 |
| Starbuzz Coffee | 2007-05-24 |
+-----+-----+
```

- Syntaxe spécifique MySQL : le nombre et la limite sont inversés.

```
SELECT location , date FROM doughnut_ratings LIMIT 1,2;
```

```
+-----+-----+
| location | date |
+-----+-----+
| Duncan's Donuts | 2007-08-25 |
| Starbuzz Coffee | 2007-05-24 |
+-----+-----+
```

```
mysql> SELECT * FROM doughnut_ratings;
```

```
+-----+-----+-----+-----+-----+-----+
| location | time | date | type | rating | comments |
+-----+-----+-----+-----+-----+-----+
| Krispy King | 08:50:00 | 2007-09-27 | plain glazed | 10 | almost perfect |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL | 6 | greasy |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake | 5 | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly | 7 | not enough jelly |
+-----+-----+-----+-----+-----+-----+
```

Trier les résultats

ORDER BY

- **SELECT ... FROM** *myTable* **ORDER BY** *myAttribute1* [ASC|DESC], *myAttribute2* [ASC|DESC]...;
- par défaut **ASC**

SELECT location , date **FROM** doughnut_ratings **ORDER BY** date;

```

+-----+-----+
| location      | date      |
+-----+-----+
| Duncan's Donuts | 2007-04-26 |
| Starbuzz Coffee | 2007-05-24 |
| Duncan's Donuts | 2007-08-25 |
| Krispy King     | 2007-09-27 |
+-----+-----+

```

```
mysql> SELECT * FROM doughnut_ratings;
```

```

+-----+-----+-----+-----+-----+-----+
| location      | time      | date      | type      | rating | comments      |
+-----+-----+-----+-----+-----+-----+
| Krispy King   | 08:50:00 | 2007-09-27 | plain glazed | 10    | almost perfect |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL       | 6     | greasy         |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake | 5     | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly      | 7     | not enough jelly |
+-----+-----+-----+-----+-----+-----+

```

Exemple de tri sur plusieurs colonnes

```
SELECT location ,date FROM doughnut_ratings
ORDER BY location , date DESC;
```

```
+-----+-----+
| location      | date      |
+-----+-----+
| Duncan's Donuts | 2007-08-25 |
| Duncan's Donuts | 2007-04-26 |
| Krispy King     | 2007-09-27 |
| Starbuzz Coffee | 2007-05-24 |
+-----+-----+
```

```
mysql> SELECT * FROM doughnut_ratings;
```

```
+-----+-----+-----+-----+-----+-----+
| location      | time      | date      | type      | rating | comments      |
+-----+-----+-----+-----+-----+-----+
| Krispy King   | 08:50:00 | 2007-09-27 | plain glazed | 10 | almost perfect |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL      | 6 | greasy      |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake | 5 | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly     | 7 | not enough jelly |
+-----+-----+-----+-----+-----+-----+
```

Nouvelle table pour les exemples

```
CREATE TABLE 'easy_drinks' (
  'drink_name' varchar(16) default NULL,
  'main' varchar(20) default NULL,
  'amount1' decimal(3,1) default NULL,
  'second' varchar(20) default NULL,
  'amount2' decimal(4,2) default NULL,
);
```

```
SELECT * FROM easy_drinks;
```

drink_name	main	amount1	second	amount2
Kiss on the Lips	cherry juice	2.0	apricot nectar	7.00
Hot Gold	peach nectar	3.0	orange juice	6.00
Lone Tree	soda	1.5	cherry juice	0.75
Greyhound	soda	1.5	grapefruit juice	5.00
Indian Summer	apple juice	2.0	hot tea	6.00
Bull Frog	iced tea	1.5	lemonade	5.00
Soda and It	soda	2.0	grape juice	1.00
Blackthorn	tonic water	1.5	pineapple juice	1.00
Blue Moon	soda	1.5	blueberry juice	0.75
Oh My Gosh	peach nectar	1.0	pineapple juice	1.00
Lime Fizz	Sprite	1.5	lime juice	0.75

Les fonctions d'agrégation

Fonction	Signification
SUM	somme
AVG	moyenne
MIN	minimum
MAX	maximum
COUNT	nombre

```
SELECT SUM(amount1) FROM easy_drinks;
```

```
+-----+
| SUM(amount1) |
+-----+
|          19.0 |
+-----+
```

```
SELECT * FROM easy_drinks;
```

```
+-----+-----+-----+-----+-----+
| drink_name | main | amount1 | second | amount2 |
+-----+-----+-----+-----+-----+
| Kiss on the Lips | cherry juice | 2.0 | apricot nectar | 7.00 |
| Hot Gold | peach nectar | 3.0 | orange juice | 6.00 |
| Lone Tree | soda | 1.5 | cherry juice | 0.75 |
| Greyhound | soda | 1.5 | grapefruit juice | 5.00 |
| Indian Summer | apple juice | 2.0 | hot tea | 6.00 |
| Bull Frog | iced tea | 1.5 | lemonade | 5.00 |
| Soda and It | soda | 2.0 | grape juice | 1.00 |
| Blackthorn | tonic water | 1.5 | pineapple juice | 1.00 |
| Blue Moon | soda | 1.5 | blueberry juice | 0.75 |
| Oh My Gosh | peach nectar | 1.0 | pineapple juice | 1.00 |
| Lime Fizz | Sprite | 1.5 | lime juice | 0.75 |
+-----+-----+-----+-----+-----+
```

Les fonctions d'agrégation : exemples

```
SELECT COUNT(main) FROM easy_drinks;
```

```
+-----+
| COUNT(main) |
+-----+
|           11 |
+-----+
```

```
SELECT COUNT(DISTINCT main) FROM easy_drinks;
```

```
+-----+
| COUNT(DISTINCT main) |
+-----+
|                   7 |
+-----+
```

```
SELECT * FROM easy_drinks;
```

drink_name	main	amount1	second	amount2
Kiss on the Lips	cherry juice	2.0	apricot nectar	7.00
Hot Gold	peach nectar	3.0	orange juice	6.00
Lone Tree	soda	1.5	cherry juice	0.75
Greyhound	soda	1.5	grapefruit juice	5.00
Indian Summer	apple juice	2.0	hot tea	6.00
Bull Frog	iced tea	1.5	lemonade	5.00
Soda and It	soda	2.0	grape juice	1.00
Blackthorn	tonic water	1.5	pineapple juice	1.00
Blue Moon	soda	1.5	blueberry juice	0.75
Oh My Gosh	peach nectar	1.0	pineapple juice	1.00
Lime Fizz	Sprite	1.5	lime juice	0.75

Les opérateurs pour les attributs numériques

Opérateur	Signification
+	somme
-	différence
*	multiplication
/	division
DIV	division euclidienne
MOD ou %	modulo

```
SELECT drink_name , amount1+amount2 FROM easy_drinks WHERE drink_name LIKE 'B%';
```

```
+-----+
| drink_name | amount1+amount2 |
+-----+
| Bull Frog  |          6.50 |
| Blackthorn |          2.50 |
| Blue Moon  |          2.25 |
+-----+
```

```
SELECT * FROM easy_drinks;
```

```
+-----+-----+-----+-----+-----+
| drink_name | main      | amount1 | second | amount2 |
+-----+-----+-----+-----+-----+
| Kiss on the Lips | cherry juice | 2.0 | apricot nectar | 7.00 |
| Hot Gold         | peach nectar | 3.0 | orange juice   | 6.00 |
| Lone Tree        | soda        | 1.5 | cherry juice   | 0.75 |
| Greyhound        | soda        | 1.5 | grapefruit juice | 5.00 |
| Indian Summer    | apple juice | 2.0 | hot tea        | 6.00 |
| Bull Frog        | iced tea    | 1.5 | lemonade       | 5.00 |
| Soda and It      | soda        | 2.0 | grape juice    | 1.00 |
| Blackthorn       | tonic water | 1.5 | pineapple juice | 1.00 |
| Blue Moon        | soda        | 1.5 | blueberry juice | 0.75 |
| Oh My Gosh       | peach nectar | 1.0 | pineapple juice | 1.00 |
| Lime Fizz        | Sprite      | 1.5 | lime juice     | 0.75 |
+-----+-----+-----+-----+-----+
```

Modifier la sélection avec des conditions

CASE

SELECT ... CASE

WHEN *myCondition1* **THEN** *myResult1*

WHEN *myCondition2* **THEN** *myResult2*

ELSE... END FROM ...

```
SELECT drink_name, CASE WHEN amount1 >= amount2 THEN 'more_main'
                     ELSE 'more_second'
                     END
```

```
FROM easy_drinks;
```

```
-----+-----
| drink_name      | CASE WHEN amount1 >= amount2 THEN 'more main' ELSE 'more second' END |
-----+-----
| Kiss on the Lips | more second |
| Hot Gold        | more second |
| Lone Tree       | more main   |
| Greyhound       | more second |
| Indian Summer   | more second |
| Bull Frog       | more second |
| Soda and It     | more main   |
| Blackthorn      | more main   |
| Blue Moon       | more main   |
| Oh My Gosh      | more main   |
| Lime Fizz       | more main   |
-----+-----
```

SELECT * FROM easy_drinks;

drink_name	main	amount1	second	amount2
Kiss on the Lips	cherry juice	2.0	apricot nectar	7.00
Hot Gold	peach nectar	3.0	orange juice	6.00
Lone Tree	soda	1.5	cherry juice	0.75
Greyhound	soda	1.5	grapefruit juice	5.00
Indian Summer	apple juice	2.0	hot tea	6.00
Bull Frog	iced tea	1.5	lemonade	5.00
Soda and It	soda	2.0	grape juice	1.00
Blackthorn	tonic water	1.5	pineapple juice	1.00
Blue Moon	soda	1.5	blueberry juice	0.75
Oh My Gosh	peach nectar	1.0	pineapple juice	1.00
Lime Fizz	Sprite	1.5	lime juice	0.75

Dépendance fonctionnelle

Soit $R(A_1, A_2, \dots, A_n)$ une relation et X et Y deux sous-ensembles de l'ensemble des attributs de R , $\{A_1, A_2, \dots, A_n\}$.

On dit que Y **dépend fonctionnellement de** X ou que X **détermine** Y , noté $X \rightarrow Y$, si et seulement si, pour n'importe quelle valeur de R , à chaque valeur de X dans R correspond exactement une seule valeur de Y dans R .

- Une clé primaire détermine tout attribut de R .

Regrouper les résultats en fonction des valeurs d'un champ

GROUP BY

- **SELECT** *myAttribute1*, *myAttribute2* **FROM** *myTable* ... **GROUP BY** *myAttribute1*, *myAttribute2*...[**ORDER BY** ...];
- Les attributs du **SELECT** doivent être fonctionnellement dépendants des attributs du **GROUP BY** ou apparaître dans une fonction d'agrégation.

```
SELECT main,drink_name FROM easy_drinks GROUP BY main;
```

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

```
SELECT main,drink_name FROM easy_drinks GROUP BY main,drink_name ORDER BY main;
```

main	drink_name
soda	Lone Tree
soda	Greyhound
soda	Soda and It
soda	Blue Moon
Sprite	Lime Fizz
tonic water	Blackthorn
apple juice	Indian Summer
cherry juice	Kiss on the Lips
iced tea	Bull Frog
peach nectar	Hot Gold
peach nectar	Oh My Gosh

```
SELECT main,drink_name FROM easy_drinks GROUP BY main,drink_name;
```

main	drink_name
cherry juice	Kiss on the Lips
peach nectar	Hot Gold
soda	Lone Tree
soda	Greyhound
apple juice	Indian Summer
iced tea	Bull Frog
soda	Soda and It
tonic water	Blackthorn
soda	Blue Moon
peach nectar	Oh My Gosh
Sprite	Lime Fizz

Exemples de GROUP BY (1)

```
SELECT main,COUNT(*) FROM easy_drinks GROUP BY main;
```

```
+-----+
| main          | COUNT(*) |
+-----+
| apple juice  |         1 |
| cherry juice |         1 |
| iced tea     |         1 |
| peach nectar |         2 |
| soda        |         4 |
| Sprite      |         1 |
| tonic water  |         1 |
+-----+
```

```
SELECT main,COUNT(*) FROM easy_drinks GROUP BY main ORDER BY COUNT(*);
```

```
+-----+
| main          | COUNT(*) |
+-----+
| apple juice  |         1 |
| iced tea     |         1 |
| tonic water  |         1 |
| Sprite      |         1 |
| cherry juice |         1 |
| peach nectar |         2 |
| soda        |         4 |
+-----+
```

```
SELECT * FROM easy_drinks;
```

```
+-----+-----+-----+-----+
| drink_name | main | amount1 | second | amount2 |
+-----+-----+-----+-----+
| Kiss on the Lips | cherry juice | 2.0 | apricot nectar | 7.00 |
| Hot Gold | peach nectar | 3.0 | orange juice | 6.00 |
| Lone Tree | soda | 1.5 | cherry juice | 0.75 |
| Greyhound | soda | 1.5 | grapefruit juice | 5.00 |
| Indian Summer | apple juice | 2.0 | hot tea | 6.00 |
| Bull Frog | iced tea | 1.5 | lemonade | 5.00 |
| Soda and It | soda | 2.0 | grape juice | 1.00 |
| Blackberry | tonic water | 1.5 | pineapple juice | 1.00 |
| Blue Moon | soda | 1.5 | blueberry juice | 0.75 |
| Oh My Gosh | peach nectar | 1.0 | pineapple juice | 1.00 |
| Lime Fizz | Sprite | 1.5 | lime juice | 0.75 |
+-----+-----+-----+-----+
```

Exemples de GROUP BY (2)

```
SELECT main ,AVG(amount1) ,AVG(amount2) FROM easy_drinks GROUP BY main;
```

```
-----+-----+-----+
| main          | AVG(amount1) | AVG(amount2) |
+-----+-----+-----+
| apple juice   | 2.00000      | 6.000000     |
| cherry juice  | 2.00000      | 7.000000     |
| iced tea      | 1.50000      | 5.000000     |
| peach nectar  | 2.00000      | 3.500000     |
| soda          | 1.62500      | 1.875000     |
| Sprite        | 1.50000      | 0.750000     |
| tonic water   | 1.50000      | 1.000000     |
+-----+-----+-----+
```

```
SELECT main ,MAX(amount1) ,MIN(amount2) FROM easy_drinks GROUP BY main;
```

```
-----+-----+-----+
| main          | MAX(amount1) | MIN(amount2) |
+-----+-----+-----+
| apple juice   | 2.0           | 6.00         |
| cherry juice  | 2.0           | 7.00         |
| iced tea      | 1.5           | 5.00         |
| peach nectar  | 3.0           | 1.00         |
| soda          | 2.0           | 0.75         |
| Sprite        | 1.5           | 0.75         |
| tonic water   | 1.5           | 1.00         |
+-----+-----+-----+
```

```
SELECT * FROM easy_drinks;
```

```
-----+-----+-----+-----+-----+
| drink_name   | main          | amount1 | second    | amount2 |
+-----+-----+-----+-----+-----+
| Kiss on the Lips | cherry juice | 2.0      | apricot nectar | 7.00     |
| Hot Gold       | peach nectar | 3.0      | orange juice  | 6.00     |
| Lone Tree      | soda         | 1.5      | cherry juice  | 0.75     |
| Greyhound      | soda         | 1.5      | grapefruit juice | 5.00     |
| Indian Summer  | apple juice  | 2.0      | hot tea       | 6.00     |
| Bull Frog      | iced tea     | 1.5      | lemonade      | 5.00     |
| Soda and It    | soda         | 2.0      | grape juice   | 1.00     |
| Blackberry     | tonic water  | 1.5      | pineapple juice | 1.00     |
| Blue Moon      | soda         | 1.5      | blueberry juice | 0.75     |
| Oh My Gosh     | peach nectar | 1.0      | pineapple juice | 1.00     |
| Lime Fizz      | Sprite       | 1.5      | lime juice    | 0.75     |
+-----+-----+-----+-----+-----+
```

Sélectionner seulement certains groupes

HAVING

- **SELECT** *myAttribute1*, *myAttribute2* **FROM** *myTable* ... **GROUP BY** *myAttribute1*, *myAttribute2*... **HAVING** *myConditions* [**ORDER BY** ...];

```
SELECT main ,MAX(amount1) ,MIN(amount2) FROM easy_drinks
GROUP BY main HAVING main='soda ';
```

```
+-----+-----+-----+
| main | MAX(amount1) | MIN(amount2) |
+-----+-----+-----+
| soda |           2.0 |           0.75 |
+-----+-----+-----+
```

```
SELECT * FROM easy_drinks;
```

```
+-----+-----+-----+-----+-----+
| drink_name | main | amount1 | second | amount2 |
+-----+-----+-----+-----+-----+
| Kiss on the Lips | cherry juice | 2.0 | apricot nectar | 7.00 |
| Hot Gold | peach nectar | 3.0 | orange juice | 6.00 |
| Lone Tree | soda | 1.5 | cherry juice | 0.75 |
| Greyhound | soda | 1.5 | grapefruit juice | 5.00 |
| Indian Summer | apple juice | 2.0 | hot tea | 6.00 |
| Bull Frog | iced tea | 1.5 | lemonade | 5.00 |
| Soda and It | soda | 2.0 | grape juice | 1.00 |
| Blackthorn | tonic water | 1.5 | pineapple juice | 1.00 |
| Blue Moon | soda | 1.5 | blueberry juice | 0.75 |
| Oh My Gosh | peach nectar | 1.0 | pineapple juice | 1.00 |
| Lime Fizz | Sprite | 1.5 | lime juice | 0.75 |
+-----+-----+-----+-----+-----+
```

Exemples de GROUP BY (2)

```
SELECT main FROM easy_drinks GROUP BY main
HAVING COUNT(*) > 1;
```

```
+-----+
| main   |
+-----+
| peach nectar |
| soda   |
+-----+
```

```
SELECT * FROM easy_drinks;
```

drink_name	main	amount1	second	amount2
Kiss on the Lips	cherry juice	2.0	apricot nectar	7.00
Hot Gold	peach nectar	3.0	orange juice	6.00
Lone Tree	soda	1.5	cherry juice	0.75
Greyhound	soda	1.5	grapefruit juice	5.00
Indian Summer	apple juice	2.0	hot tea	6.00
Bull Frog	iced tea	1.5	lemonade	5.00
Soda and It	soda	2.0	grape juice	1.00
Blackthorn	tonic water	1.5	pineapple juice	1.00
Blue Moon	soda	1.5	blueberry juice	0.75
Oh My Gosh	peach nectar	1.0	pineapple juice	1.00
Lime Fizz	Sprite	1.5	lime juice	0.75

Les fonctions MySQL pour les attributs numériques (1)

Fonction	Signification
ABS	valeur absolue
ACOS	arc cosinus
ASIN	arc sinus
ATAN	arc tangente
ATAN2	arc tangente (2 paramètres)
CEIL ou CEILING	partie entière supérieure
CONV	conversion d'une base à une autre
COS	cosinus
COT	cotangente
CRC32	code de redondance cyclique sur 32 bits
DEGREES	conversion de radian en degré
EXP	exponentielle
FLOOR	partie entière
LN	logarithme népérien

Les fonctions MySQL pour les attributs numériques (2)

LOG	logarithme dans une base donnée en paramètre
LOG10	logarithme à base 10
LOG2	logarithme à base 2
MOD	modulo
PI	la valeur de π en double précision
POW ou POWER	puissance
RADIANS	conversion de degré vers radian
RAND	valeur aléatoire dans $[0, 1[$
ROUND	arrondi
SIGN	signe
SIN	sinus
SQRT	racine carrée
TAN	tangente
TRUNCATE	troncature

Exemples avec fonctions numériques

```
SELECT drink_name , CONV(amount2,10,2) FROM easy_drinks WHERE drink_name LIKE 'K%';
```

```
+-----+
| drink_name | CONV(amount2,10,2) |
+-----+
| Kiss on the Lips | 111 |
+-----+
```

```
SELECT TAN(PI()/4);
```

```
+-----+
| TAN(PI()/4) |
+-----+
| 0.9999999999999999 |
+-----+
```

```
SELECT drink_name FROM easy_drinks WHERE amount2=CEIL(TAN(PI()/4));
```

```
+-----+
| drink_name |
+-----+
| Soda and It |
| Blackthorn |
| Oh My Gosh |
+-----+
```

```
SELECT * FROM easy_drinks;
```

drink_name	main	amount1	second	amount2
Kiss on the Lips	cherry juice	2.0	apricot nectar	7.00
Hot Gold	peach nectar	3.0	orange juice	6.00
Lone Tree	soda	1.5	cherry juice	0.75
Greyhound	soda	1.5	grapefruit juice	5.00
Indian Summer	apple juice	2.0	hot tea	6.00
Bull Frog	iced tea	1.5	lemonade	5.00
Soda and It	soda	2.0	grape juice	1.00
Blackthorn	tonic water	1.5	pineapple juice	1.00
Blue Moon	soda	1.5	blueberry juice	0.75
Oh My Gosh	peach nectar	1.0	pineapple juice	1.00
Lime Fizz	Sprite	1.5	lime juice	0.75

Les fonctions MySQL pour les chaînes de caractères (1)

ASCII	code entier du premier caractère s'il est codé sur 8 bits
BIN	chaîne de caractères représentation binaire d'un entier
BIT_LENGTH	longueur en bits
CHAR	caractère en fonction de son code entier
CHAR_LENGTH ou CHARACTER_LENGTH	longueur en nombre d'octets
CONCAT	concaténation
CONCAT_WS	concaténation avec séparateur
ELT	$n^{\text{ème}}$ élément d'une liste de chaînes
FORMAT	formate un nombre
HEX	chaîne de caractères représentation hexadécimale (base 16) d'un entier
INSERT	insère une chaîne dans une autre à une position donnée
INSTR	position de la première occurrence d'une sous chaîne dans une chaîne
LCASE ou LOWER	convertit en minuscule
LEFT	n premiers caractères d'une chaîne
LENGTH	longueur en nombre d'octets
LOAD_FILE	renvoie le contenu du fichier sous forme de chaîne
LOCATE ou POSITION	position d'une sous chaîne dans une chaîne
LOWER	convertit en minuscule
LPAD	agrandit une chaîne remplissant le début
LTRIM	enlève les espaces en début de chaîne

Les fonctions MySQL pour les chaînes de caractères (2)

MAKE_SET	construit un ensemble de chaînes
MID	retourne une sous chaîne à une position donnée
OCT	chaîne de caractères, représentation en octal (base 8) d'un entier
OCTET_LENGTH ou LENGTH	longueur en nombre d'octets
ORD	code entier du premier caractère qui peut être codé sur plusieurs octets
POSITION	position d'une sous chaîne dans une chaîne
QUOTE	ajoute des caractères <i>quote</i> ' autour de la chaîne, échappe ' et \ à l'intérieur avec \
REPEAT	retourne la chaîne concaténée plusieurs fois avec elle-même
REPLACE	remplace dans une chaîne les occurrences d'une chaîne par une autre
REVERSE	renvoie la chaîne avec les caractères en ordre inverse
RIGHT	<i>n</i> derniers caractères
RPAD	agrandit une chaîne remplissant la fin
RTRIM	enlève les espaces en fin de chaîne
SPACE	retourne une chaîne constituée de <i>n</i> espaces
SUBSTR ou SUBSTRING	retourne une sous chaîne
SUBSTRING_INDEX	retourne la sous chaîne avant la <i>n^{ème}</i> occurrence du délimiteur
TRIM	supprimer un caractère en début et en fin de chaîne
UCASE ou UPPER	convertit en majuscule
UNHEX	inverse de la fonction HEX
UPPER	convertit en majuscule

Exemples avec fonctions sur les chaînes de caractères

```
SELECT drink_name, BIN(amount2) FROM easy_drinks WHERE amount2>4;
```

```
+-----+-----+
| drink_name | BIN(amount2) |
+-----+-----+
| Kiss on the Lips | 111 |
| Hot Gold | 110 |
| Greyhound | 101 |
| Indian Summer | 110 |
| Bull Frog | 101 |
+-----+-----+
```

```
SELECT CONCAT_WS(' ',main,second) FROM easy_drinks WHERE main='soda';
```

```
+-----+
| CONCAT_WS(' ',main,second) |
+-----+
| soda cherry juice |
| soda grapefruit juice |
| soda grape juice |
| soda blueberry juice |
+-----+
```

```
SELECT * FROM easy_drinks;
+-----+-----+-----+-----+
| drink_name | main | amount1 | second | amount2 |
+-----+-----+-----+-----+
| Kiss on the Lips | cherry juice | 2.0 | apricot nectar | 7.00 |
| Hot Gold | peach nectar | 3.0 | orange juice | 6.00 |
| Lone Tree | soda | 1.5 | cherry juice | 0.75 |
| Greyhound | soda | 1.5 | grapefruit juice | 5.00 |
| Indian Summer | apple juice | 2.0 | hot tea | 6.00 |
| Bull Frog | iced tea | 1.5 | lemonade | 5.00 |
| Soda and It | soda | 2.0 | grape juice | 1.00 |
| Blackthorn | tonic water | 1.5 | pineapple juice | 1.00 |
| Blue Moon | soda | 1.5 | blueberry juice | 0.75 |
| Oh My Gosh | peach nectar | 1.0 | pineapple juice | 1.00 |
| Lime Fizz | Sprite | 1.5 | lime juice | 0.75 |
+-----+-----+-----+-----+
```

Les fonctions MySQL pour les dates (1)

ADDDATE ou DATE_ADD	ajoute une durée en jour à une date
ADDTIME	ajoute une durée (TIME) à une date
CONVERT_TZ	convertit une DATETIME d'un fuseau horaire à un autre
CURDATE ou CURRENT_DATE	date courante
CURRENT_TIME_STAMP ou NOW	date et heure courante
CURTIME ou CURRENT_TIME	heure courante
DATE	extrait la date d'une DATETIME
DATE_ADD ou ADDDATE	ajoute une durée à une date
DATEDIFF	différence en jour entre deux dates
DATE_FORMAT	formate une date
DATE_SUB ou SUBDATE	soustrait une durée à une date
DAY ou DAYOFMONTH	extrait le jour (entier) d'une date
DAYNAME	retourne le jour de la semaine sous forme de chaîne
DAYOFMONTH	extrait le jour du mois (entier de [1, 31]) d'une date
DAYOFWEEK	extrait le jour de la semaine (entier de [1, 7]) d'une date
DAYOFYEAR	extrait le jour de l'année (entier de [1, 366]) d'une date
EXTRACT	extrait une partie de DATETIME
FROM_UNIXTIME	convertit un UNIX_TIMESTAMP en DATETIME
GET_FORMAT	retourne un format de date prédéfini
 HOUR	extrait l'heure d'une durée (TIME)
LAST_DAY	retourne la date du dernier jour du mois d'une date
LOCALTIME ou LOCALTIMESTAMP	date et heure courante
MAKEDATE	créé une date à partir d'un jour de l'année et d'une année
MAKETIME	créé une durée à partir des heures, minutes et secondes

Les fonctions MySQL pour les dates (2)

MICROSECOND	extrait les microsecondes d'une DATETIME ou d'une durée (TIME)
MINUTE	extrait les minutes d'une durée
MONTH	extrait le mois d'une date
MONTHNAME	retourne le nom du mois d'une date.
NOW	date et heure courante
QUARTER	retourne le numéro du trimestre d'une date
SECOND	extrait les secondes d'une durée (TIME)
SEC_TO_TIME	convertit une durée exprimée en seconde en TIME
STR_TO_DATE	convertit une chaîne de caractères en DATETIME, DATE ou TIME
SUBDATE ou DATE_SUB	soustrait une durée à une date
SUBTIME	soustrait deux dates ou deux durées
TIME	extrait la partie TIME d'une DATETIME
TIME_TO_SEC	convertit une durée (TIME) en seconde
TIMEDIFF	soustrait deux dates ou deux durées (TIME)
UNIX_TIMESTAMP	retourne un UNIX_TIMESTAMP
UTC_DATE	date UTC courante
UTC_TIME	heure UTC courante
UTC_TIMESTAMP	datetime UTC courante
WEEK ou WEEKOFYEAR	numéro de la semaine
WEEKDAY	numéro du jour de la semaine
WEEKOFYEAR ou WEEK	numéro de la semaine
YEAR	extrait l'année d'une date
YEARWEEK	extrait l'année et le numéro de la semaine d'une date

Exemples avec fonctions sur les dates

```
SELECT date, DATE_FORMAT(date, '%d/%m/%Y') FROM doughnut_ratings;
```

```
+-----+-----+
| date          | DATE_FORMAT(date, '%d/%m/%Y') |
+-----+-----+
| 2007-09-27   | 27/09/2007                    |
| 2007-08-25   | 25/08/2007                    |
| 2007-05-24   | 24/05/2007                    |
| 2007-04-26   | 26/04/2007                    |
+-----+-----+
```

```
SELECT EXTRACT(MONTH FROM date), MONTHNAME(date) FROM doughnut_ratings;
```

```
+-----+-----+
| EXTRACT(MONTH FROM date) | MONTHNAME(date) |
+-----+-----+
|                          | 9 | September      |
|                          | 8 | August         |
|                          | 5 | May            |
|                          | 4 | April         |
+-----+-----+
```

```
mysql> SELECT * FROM doughnut_ratings;
```

```
+-----+-----+-----+-----+-----+-----+
| location    | time   | date       | type       | rating | comments      |
+-----+-----+-----+-----+-----+-----+
| Krispy King | 08:50:00 | 2007-09-27 | plain glazed | 10 | almost perfect |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL       | 6 | greasy        |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake | 5 | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly      | 7 | not enough jelly |
+-----+-----+-----+-----+-----+-----+
```