Date/Time Functions

ADDDATE	EXTRACT	SECOND
ADDTIME	FROM_DAYS	STR_TO_DATE
CURDATE	HOUR	SUBDATE
CURRENT_DATE	LAST_DAY	SUBTIME
CURRENT_TIME	LOCALTIME	SYSDATE
CURRENT_TIMESTAMP	LOCALTIMESTAMP	TIME
CURTIME	MAKEDATE	TIME_FORMAT
DATE	MAKETIME	TIME_TO_SEC
DATE_ADD	MICROSECOND	TIMEDIFF
DATE_FORMAT	MINUTE	TIMESTAMP
DATE_SUB	MONTH	TO_DAYS
DATEDIFF	MONTHNAME	WEEK
DAY	NOW	WEEKDAY
DAYNAME	PERIOD_ADD	WEEKOFYEAR
DAYOFMONTH	PERIOD_DIFF	YEAR
DAYOFWEEK	QUARTER	YEARWEEK
DAYOFYEAR	SEC_TO_TIME	

-- curdate()

select curdate();

-- current_date

select current_date();

-- current_timestamp

select current_timestamp();

-- curtime()

select curtime();

-- date_add

```
select date_add(current_date(), interval 10 day);

-- date_format

select date_format(current_date,'%y/%d/%m');

select date_format(current_date,'%Y/%d/%m');

select date_format(current_timestamp,'%Y/%d/%m %h:%i:%s');

select date_format(current_timestamp,'%Y/%d/%m %H:%i:%s');

select date_format(current_timestamp,'%m/%d/%Y %H:%i:%s');


-- Date_sub

select date_sub(current_date(), interval 1 Year);

select date_sub(current_date(), interval 1 day);

select date_sub(current_date(), interval 1 month);


-- date_diff

select datediff(current_date(),'2022-06-18');


-- day

select day(current_date());


-- dayofyear

select dayofyear(current_date());
```

```
-- now

select now();


-- str_to_date

select str_to_date('220621','%y%m%d');


-- sysdate

select sysdate();


-- current_timestamp

select current_timestamp();


-- to_days

select to_days(current_date());
```

Advanced Functions

BIN	CONVERT	NULLIF
BINARY	CURRENT_USER	SESSION_USER
CASE	DATABASE	SYSTEM_USER
CAST	IF	USER
COALESCE	IFNULL	VERSION
CONNECTION_ID	ISNULL	
CONV	LAST_INSERT_ID	

```
-- current_user

select current_user();


-- database

select database();


-- user

select user();


-- version()

select version();  -- 5.7.38-log
```

UNION, UNION ALL:-

UNION

```
SELECT * FROM customers WHERE CustomerID<5;

SELECT * FROM customers WHERE CustomerID>5 AND CustomerID<10;


SELECT * FROM customers WHERE CustomerID<5

UNION

SELECT * FROM customers WHERE CustomerID>5 AND CustomerID<10;
```

UNION ALL

```
SELECT * FROM customers WHERE CustomerID<8;

SELECT * FROM customers WHERE CustomerID>5 AND CustomerID<10;
```

```
SELECT * FROM customers WHERE CustomerID<8

UNION

SELECT * FROM customers WHERE CustomerID>5 AND CustomerID<10;
```

```
SELECT * FROM customers WHERE CustomerID<8

UNION ALL

SELECT * FROM customers WHERE CustomerID>5 AND CustomerID<10;
```

MySQL: Data Types

String Datatypes

The following is a list of datatypes available in MySQL, which includes string, numeric, date/time, and large object datatypes.

Data Type Syntax	Maximum Size	Explanation
CHAR(*size*)	Maximum size of 255 characters.	Where ***size*** is the number of characters to store. Fixed-length strings. Space padded on right to equal ***size*** characters.
VARCHAR(*size*)	Maximum size of 255 characters.	Where ***size*** is the number of characters to store. Variable-length string.
TINYTEXT(*size*)	Maximum size of 255 characters.	Where ***size*** is the number of characters to store.
TEXT(*size*)	Maximum size of 65,535 characters.	Where ***size*** is the number of characters to store.
MEDIUMTEXT(*size*)	Maximum size of 16,777,215 characters.	Where ***size*** is the number of characters to store.

Data Type Syntax	Maximum Size	Explanation
LONGTEXT(*size*)	Maximum size of 4GB or 4,294,967,295 characters.	Where ***size*** is the number of characters to store.
BINARY(*size*)	Maximum size of 255 characters.	Where ***size*** is the number of binary characters to store. Fixed-length strings. Space padded on right to equal ***size*** characters. (Introduced in MySQL 4.1.2)
VARBINARY(*size*)	Maximum size of 255 characters.	Where ***size*** is the number of characters to store. Variable-length string. (Introduced in MySQL 4.1.2)

Numeric Datatypes

The following are the **Numeric Datatypes** in MySQL:

Data Type Syntax	Maximum Size	Explanation
BIT	Very small integer value that is equivalent to TINYINT(1). Signed values range from -128 to 127. Unsigned values range from 0 to 255.	

Data Type Syntax	Maximum Size	Explanation
TINYINT(*m*)	Very small integer value. Signed values range from -128 to 127. Unsigned values range from 0 to 255.	
SMALLINT(*m*)	Small integer value. Signed values range from -32768 to 32767. Unsigned values range from 0 to 65535.	
MEDIUMINT(*m*)	Medium integer value. Signed values range from -8388608 to 8388607. Unsigned values range from 0 to 16777215.	
INT(*m*)	Standard integer value. Signed values range from -2147483648 to 2147483647. Unsigned values range from 0 to 4294967295.	
INTEGER(*m*)	Standard integer value. Signed values range from -2147483648 to 2147483647. Unsigned values range from 0 to 4294967295.	This is a synonym for the INT datatype.
BIGINT(*m*)	Big integer value. Signed values range from -9223372036854775808 to	

Data Type Syntax	Maximum Size	Explanation
	9223372036854775807. Unsigned values range from 0 to 18446744073709551615.	
DECIMAL(*m*,*d*)	Unpacked fixed point number. *m* defaults to 10, if not specified. *d* defaults to 0, if not specified.	Where *m* is the total digits and *d* is the number of digits after the decimal.
DEC(*m*,*d*)	Unpacked fixed point number. *m* defaults to 10, if not specified. *d* defaults to 0, if not specified.	Where *m* is the total digits and *d* is the number of digits after the decimal. This is a synonym for the DECIMAL datatype.
NUMERIC(*m*,*d*)	Unpacked fixed-point number. *m* defaults to 10, if not specified. *d* defaults to 0, if not specified.	Where *m* is the total digits and *d* is the number of digits after the decimal. This is a synonym for the DECIMAL datatype.
FIXED(*m*,*d*)	Unpacked fixed-point number. *m* defaults to 10, if not specified. *d* defaults to 0, if not specified.	Where *m* is the total digits and *d* is the number of digits after

Data Type Syntax	Maximum Size	Explanation
		the decimal. (Introduced in MySQL 4.1) This is a synonym for the DECIMAL datatype.
FLOAT(*m*,*d*)	Single precision floating point number.	Where *m* is the total digits and *d* is the number of digits after the decimal.
DOUBLE(*m*,*d*)	Double precision floating point number.	Where *m* is the total digits and *d* is the number of digits after the decimal.
DOUBLE PRECISION(*m*,*d*)	Double precision floating point number.	Where *m* is the total digits and *d* is the number of digits after the decimal. This is a synonym for the DOUBLE datatype.
REAL(*m*,*d*)	Double precision floating point number.	Where *m* is the total digits and *d* is the number of digits after the decimal.

Data Type Syntax	Maximum Size	Explanation
		This is a synonym for the DOUBLE datatype.
FLOAT(p)	Floating point number.	Where *p* is the precision.
BOOL	Synonym for TINYINT(1)	Treated as a boolean data type where a value of 0 is considered to be FALSE and any other value is considered to be TRUE.
BOOLEAN	Synonym for TINYINT(1)	Treated as a boolean data type where a value of 0 is considered to be FALSE and any other value is considered to be TRUE.

Date/Time Datatypes

The following are the **Date/Time Datatypes** in MySQL:

Data Type Syntax	Maximum Size	Explanation
DATE	Values range from '1000-01-01' to '9999-12-31'.	Displayed as 'YYYY-MM-DD'.
DATETIME	Values range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.	Displayed as 'YYYY-MM-DD HH:MM:SS'.
TIMESTAMP(*m*)	Values range from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC.	Displayed as 'YYYY-MM-DD HH:MM:SS'.
TIME	Values range from '-838:59:59' to '838:59:59'.	Displayed as 'HH:MM:SS'.
YEAR[(2|4)]	Year value as 2 digits or 4 digits.	Default is 4 digits.

Large Object (LOB) Datatypes

The following are the **LOB Datatypes** in MySQL:

Data Type Syntax	Maximum Size	Explanation
TINYBLOB	Maximum size of 255 bytes.	

Data Type Syntax	Maximum Size	Explanation
BLOB(*size*)	Maximum size of 65,535 bytes.	Where ***size*** is the number of characters to store (***size*** is optional and was introduced in MySQL 4.1)
MEDIUMBLOB	Maximum size of 16,777,215 bytes.	
LONGTEXT	Maximum size of 4GB or 4,294,967,295 characters.4,294,967,295 characters.	