**The CREATE DATABASE statement is used to create a new SQL database.**

### Syntax

CREATE DATABASE databasename;

### Example

CREATE DATABASE testDB;

**DROP DATABASE statement is used to drop an existing SQL database.**

### Syntax

DROP DATABASE databasename;

Note: Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!

### Example

DROP DATABASE testDB

**Tip:** Make sure you have admin privilege before dropping any database. Once a database is dropped, you can check it in the list of databases with the following SQL command:

**Show all database**

SHOW DATABASES;

**Select a database**

use DATABASES\_NAME

**Show tables exist in a database**

show tables

**The BACKUP DATABASE statement is used in SQL Server to create a full back up of an existing SQL database.**

### Syntax

BACKUP DATABASE databasename  
TO DISK = 'filepath';

### Example

BACKUP DATABASE testDB  
TO DISK = 'D:\backups\testDB.bak';

**Tip:**Always back up the database to a different drive than the actual database. Then, if you get a disk crash, you will not lose your backup file along with the database.

## The SQL BACKUP WITH DIFFERENTIAL Statement

A differential back up only backs up the parts of the database that have changed since the last full database backup.

### Syntax

BACKUP DATABASE databasename  
TO DISK = 'filepath'  
WITH DIFFERENTIAL;

### Example

BACKUP DATABASE testDB  
TO DISK = 'D:\backups\testDB.bak'  
WITH DIFFERENTIAL;

**Tip:** A differential back up reduces the back up time (since only the changes are backed up).

## The SQL CREATE TABLE Statement

The CREATE TABLE statement is used to create a new table in a database.

**Syntax :**

CREATE TABLE table\_name(  
 column1 datatype,  
 column2 datatype,  
 column3 datatype,  
 ....  
);

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

### Example

CREATE TABLE Persons (  
 PersonID int,  
 LastName varchar(255),  
 FirstName varchar(255),  
 Address varchar(255),  
 City varchar(255)  
);

## Create Table Using Another Table

A copy of an existing table can also be created using CREATE TABLE.

The new table gets the same column definitions. All columns or specific columns can be selected.

If you create a new table using an existing table, the new table will be filled with the existing values from the old table.

### Syntax

CREATE TABLE new\_table\_name AS  
 SELECT column1, column2,...  
 FROM existing\_table\_name  
 WHERE ....;

### Example

CREATE TABLE TestTable AS  
SELECT customername, contactname  
FROM customers;

## The SQL DROP TABLE Statement

The DROP TABLE statement is used to drop an existing table in a database.

### Syntax

DROP TABLE table\_name;

Note: Be careful before dropping a table. Deleting a table will result in loss of complete information stored in the table!

### Example

DROP TABLE Shippers;

## SQL TRUNCATE TABLE

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

### Syntax

TRUNCATE TABLE table\_name;

## SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

## ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

ALTER TABLE table\_name  
ADD column\_name datatype;

### Example

ALTER TABLE Customers  
ADD Email varchar(255);

## ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

ALTER TABLE table\_name  
DROP COLUMN column\_name;

### Example

ALTER TABLE Customers  
DROP COLUMN Email;

## ALTER TABLE - ALTER/MODIFY COLUMN

To change the data type of a column in a table, use the following syntax:

**SQL Server / MS Access:**

ALTER TABLE table\_name  
ALTER COLUMN column\_name datatype;

**example**

ALTER TABLE Persons  
ADD DateOfBirth date;

ALTER TABLE Persons  
ALTER COLUMN DateOfBirth year;

ALTER TABLE Persons  
DROP COLUMN DateOfBirth;

## SQL Constraints

SQL constraints are used to specify rules for data in a table.

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

### Syntax

CREATE TABLE table\_name (  
 column1 datatype constraint,  
 column2 datatype constraint,  
 column3 datatype constraint,  
 ....

## SQL Constraints

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

**The following constraints are commonly used in SQL:**

* [NOT NULL](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value
* [UNIQUE](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different
* [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp)- A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - Prevents actions that would destroy links between tables
* [CHECK](https://www.w3schools.com/sql/sql_check.asp) - Ensures that the values in a column satisfies a specific condition
* [DEFAULT](https://www.w3schools.com/sql/sql_default.asp)- Sets a default value for a column if no value is specified
* [CREATE INDEX](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly

## SQL NOT NULL Constraint

By default, a column can hold NULL values.

The NOT NULL constraint enforces a column to NOT accept NULL values.

This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

### Example

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255) NOT NULL,  
 Age int  
);

## SQL NOT NULL on ALTER TABLE

To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the following SQL:

ALTER TABLE Persons  
MODIFY Age int NOT NULL;

## SQL UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

CREATE TABLE Persons (  
 ID int NOT NULL UNIQUE,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int  
);

or

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 UNIQUE (ID)  
);

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 CONSTRAINT UC\_Person UNIQUE (ID,LastName)  
);

ALTER TABLE Persons  
ADD UNIQUE (ID);

ALTER TABLE Persons  
ADD CONSTRAINT UC\_Person UNIQUE (ID,LastName);

ALTER TABLE Persons  
DROP INDEX UC\_Person;

ALTER TABLE Persons  
DROP CONSTRAINT UC\_Person;

## SQL PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 PRIMARY KEY (ID)  
);

CREATE TABLE Persons (  
 ID int NOT NULL PRIMARY KEY,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int  
);

To allow naming of a PRIMARY KEY constraint, and for defining a PRIMARY KEY constraint on multiple columns, use the following SQL syntax:

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName)  
);

## SQL PRIMARY KEY on ALTER TABLE

To create a PRIMARY KEY constraint on the "ID" column when the table is already created, use the following SQL:

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

## SQL FOREIGN KEY Constraint

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

Look at the following two tables:

### Persons Table

|  |  |  |  |
| --- | --- | --- | --- |
| PersonID | LastName | FirstName | Age |
| 1 | Hansen | Ola | 30 |
| 2 | Svendson | Tove | 23 |
| 3 | Pettersen | Kari | 20 |

### Orders Table

|  |  |  |
| --- | --- | --- |
| OrderID | OrderNumber | PersonID |
| 1 | 77895 | 3 |
| 2 | 44678 | 3 |
| 3 | 22456 | 2 |
| 4 | 24562 | 1 |

Notice that the "PersonID" column in the "Orders" table points to the "PersonID" column in the "Persons" table.

The "PersonID" column in the "Persons" table is the PRIMARY KEY in the "Persons" table.

The "PersonID" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

The FOREIGN KEY constraint prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the parent table.

## SQL FOREIGN KEY on CREATE TABLE

The following SQL creates a FOREIGN KEY on the "PersonID" column when the "Orders" table is created:

MySQL:

CREATE TABLE Orders (  
 OrderID int NOT NULL,  
 OrderNumber int NOT NULL,  
 PersonID int,  
 PRIMARY KEY (OrderID),  
 FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

## SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 CHECK (Age>=18)  
);

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 City varchar(255),  
 CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes')  
);

## SQL DEFAULT Constraint

The DEFAULT constraint is used to set a default value for a column.

The default value will be added to all new records, if no other value is specified.

CREATE TABLE Persons (  
 ID int NOT NULL,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 City varchar(255) DEFAULT 'Sandnes'  
);

CREATE TABLE Orders (  
 ID int NOT NULL,  
 OrderNumber int NOT NULL,  
 OrderDate date DEFAULT GETDATE()  
);

## SQL CREATE INDEX Statement

The CREATE INDEX statement is used to create indexes in tables.

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

Note: Updating a table with indexes takes more time than updating a table without (because the indexes also need an update). So, only create indexes on columns that will be frequently searched against.

### CREATE INDEX Syntax

Creates an index on a table. Duplicate values are allowed:

CREATE INDEX index\_name  
ON table\_name (column1, column2, ...);

CREATE INDEX idx\_lastname  
ON Persons (LastName);

CREATE INDEX idx\_pname  
ON Persons (LastName, FirstName);

## AUTO INCREMENT Field

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

CREATE TABLE Persons (  
 Personid int NOT NULL AUTO\_INCREMENT,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 PRIMARY KEY (Personid)  
);

CREATE TABLE Persons (  
 Personid int NOT NULL AUTO\_INCREMENT,  
 LastName varchar(255) NOT NULL,  
 FirstName varchar(255),  
 Age int,  
 PRIMARY KEY (Personid)  
);

## SQL Date Data Types

MySQL comes with the following data types for storing a date or a date/time value in the database:

* DATE - format YYYY-MM-DD
* DATETIME - format: YYYY-MM-DD HH:MI:SS
* TIMESTAMP - format: YYYY-MM-DD HH:MI:SS
* YEAR - format YYYY or YY

SQL Server comes with the following data types for storing a date or a date/time value in the database:

* DATE - format YYYY-MM-DD
* DATETIME - format: YYYY-MM-DD HH:MI:SS
* SMALLDATETIME - format: YYYY-MM-DD HH:MI:SS
* TIMESTAMP - format: a unique number

**Note:** The date types are chosen for a column when you create a new table in your database!

SELECT \* FROM Orders WHERE OrderDate='2008-11-11

SELECT \* FROM Orders WHERE OrderDate='2008-11-11'

**Tip:** To keep your queries simple and easy to maintain, do not use time-components in your dates, unless you have to!

## SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the CREATE VIEW statement.

### CREATE VIEW Syntax

CREATE VIEW view\_name AS  
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

**Note:** A view always shows up-to-date data! The database engine recreates the view, every time a user queries it.

### Example

CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';

We can query the view above as follows:

SELECT \* FROM [Brazil Customers];

## SQL Updating a View

A view can be updated with the CREATE OR REPLACE VIEW statement.

### SQL CREATE OR REPLACE VIEW Syntax

CREATE OR REPLACE VIEW view\_name AS  
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

CREATE OR REPLACE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName, City  
FROM Customers  
WHERE Country = 'Brazil';

### SQL DROP VIEW Syntax

DROP VIEW view\_name;

## SQL Injection

SQL injection is a code injection technique that might destroy your database.

SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

## SQL Hosting

If you want your web site to be able to store and retrieve data from a database, your web server should have access to a database-system that uses the SQL language.

If your web server is hosted by an Internet Service Provider (ISP), you will have to look for SQL hosting plans.

The most common SQL hosting databases are MS SQL Server, Oracle, MySQL, and MS Access.

## MS SQL Server

Microsoft's SQL Server is a popular database software for database-driven web sites with high traffic.

SQL Server is a very powerful, robust and full featured SQL database system.

## Oracle

Oracle is also a popular database software for database-driven web sites with high traffic.

Oracle is a very powerful, robust and full featured SQL database system.

## MySQL

MySQL is also a popular database software for web sites.

MySQL is a very powerful, robust and full featured SQL database system.

MySQL is an inexpensive alternative to the expensive Microsoft and Oracle solutions.

## MS Access

When a web site requires only a simple database, Microsoft Access can be a solution.

MS Access is not well suited for very high-traffic, and not as powerful as MySQL, SQL Server, or Oracle.

## SQL Data Types

Each column in a database table is required to have a name and a data type.

An SQL developer must decide what type of data that will be stored inside each column when creating a table. The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.

Note: Data types might have different names in different database. And even if the name is the same, the size and other details may be different! Always check the documentation!

## MySQL Data Types (Version 8.0)

In MySQL there are three main data types: string, numeric, and date and time.

### String Data Types

|  |  |
| --- | --- |
| Data type | Description |
| CHAR(size) | A FIXED length string (can contain letters, numbers, and special characters). The size parameter  specifies the column length in characters - can be from 0 to 255. Default is 1 |
| VARCHAR(size) | A VARIABLE length string (can contain letters, numbers, and special characters). The size parameter  specifies the maximum column length in characters - can be from 0 to 65535 |
| BINARY(size) | Equal to CHAR(), but stores binary byte strings. The size parameter specifies the column length in  bytes. Default is 1 |
| VARBINARY(size) | Equal to VARCHAR(), but stores binary byte strings. The size parameter specifies the maximum  column length in bytes. |
| TINYBLOB | For BLOBs (Binary Large OBjects). Max length: 255 bytes |
| TINYTEXT | Holds a string with a maximum length of 255 characters |
| TEXT(size) | Holds a string with a maximum length of 65,535 bytes |
| BLOB(size) | For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data |
| MEDIUMTEXT | Holds a string with a maximum length of 16,777,215 characters |
| MEDIUMBLOB | For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data |
| LONGTEXT | Holds a string with a maximum length of 4,294,967,295 characters |
| LONGBLOB | For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data |
| ENUM(val1, val2, val3, ...) | A string object that can have only one value, chosen from a list of possible values. You can list up to  65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted.  The values are sorted in the order you enter them |
| SET(val1, val2, val3, ...) | A string object that can have 0 or more values, chosen from a list of possible values. You can list up to  64 values in a SET list |

### Numeric Data Types

|  |  |
| --- | --- |
| Data type | Description |
| BIT(size) | A bit-value type. The number of bits per value is specified in size. The size parameter can  hold a value from 1 to 64. The default value for size is 1. |
| TINYINT(size) | A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255.  The size parameter specifies the maximum display width (which is 255) |
| BOOL | Zero is considered as false, nonzero values are considered as true. |
| BOOLEAN | Equal to BOOL |
| SMALLINT(size) | A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to  65535. The size parameter specifies the maximum display width (which is 255) |
| MEDIUMINT(size) | A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0  to 16777215. The size parameter specifies the maximum display width (which is 255) |
| INT(size) | A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is  from 0 to 4294967295. The size parameter specifies the maximum display width (which is  255) |
| INTEGER(size) | Equal to INT(size) |
| BIGINT(size) | A large integer. Signed range is from -9223372036854775808 to 9223372036854775807.  Unsigned range is from 0 to 18446744073709551615. The size parameter specifies the  maximum display width (which is 255) |
| FLOAT(size, d) | A floating point number. The total number of digits is specified in size. The number of  digits after the decimal point is specified in the d parameter. This syntax is deprecated in  MySQL 8.0.17, and it will be removed in future MySQL versions |
| FLOAT(p) | A floating point number. MySQL uses the p value to determine whether to use FLOAT or  DOUBLE for the resulting data type. If p is from 0 to 24, the data type becomes  FLOAT(). If p is from 25 to 53, the data type becomes DOUBLE() |
| DOUBLE(size, d) | A normal-size floating point number. The total number of digits is specified in size. The  number of digits after the decimal point is specified in the d parameter |
| DOUBLE PRECISION(size, d) |  |
| DECIMAL(size, d) | An exact fixed-point number. The total number of digits is specified in size. The number  of digits after the decimal point is specified in the d parameter. The maximum number  for size is 65. The maximum number for d is 30. The default value for size is 10. The  default value for d is 0. |
| DEC(size, d) | Equal to DECIMAL(size,d) |

Note: All the numeric data types may have an extra option: UNSIGNED or ZEROFILL. If you add the UNSIGNED option, MySQL disallows negative values for the column. If you add the ZEROFILL option, MySQL automatically also adds the UNSIGNED attribute to the column.

### Date and Time Data Types

|  |  |
| --- | --- |
| Data type | Description |
| DATE | A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31' |
| DATETIME(fsp) | A date and time combination. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time |
| TIMESTAMP(fsp) | A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT\_TIMESTAMP and ON UPDATE CURRENT\_TIMESTAMP in the column definition |
| TIME(fsp) | A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59' |
| YEAR | A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format. |

## SQL Server Data Types

### String Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Description | Max size | Storage |
| char(n) | Fixed width character string | 8,000 characters | Defined width |
| varchar(n) | Variable width character string | 8,000 characters | 2 bytes + number of chars |
| varchar(max) | Variable width character string | 1,073,741,824 characters | 2 bytes + number of chars |
| text | Variable width character string | 2GB of text data | 4 bytes + number of chars |
| nchar | Fixed width Unicode string | 4,000 characters | Defined width x 2 |
| nvarchar | Variable width Unicode string | 4,000 characters |  |
| nvarchar(max) | Variable width Unicode string | 536,870,912 characters |  |
| ntext | Variable width Unicode string | 2GB of text data |  |
| binary(n) | Fixed width binary string | 8,000 bytes |  |
| varbinary | Variable width binary string | 8,000 bytes |  |
| varbinary(max) | Variable width binary string | 2GB |  |
| image | Variable width binary string | 2GB |  |

### Numeric Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Description | Storage |
| bit | Integer that can be 0, 1, or NULL |  |
| tinyint | Allows whole numbers from 0 to 255 | 1 byte |
| smallint | Allows whole numbers between -32,768 and 32,767 | 2 bytes |
| int | Allows whole numbers between -2,147,483,648 and 2,147,483,647 | 4 bytes |
| bigint | Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807 | 8 bytes |
| decimal(p,s) | Fixed precision and scale numbers.  Allows numbers from -10^38 +1 to 10^38 –1.  The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.  The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0 | 5-17 bytes |
| numeric(p,s) | Fixed precision and scale numbers.  Allows numbers from -10^38 +1 to 10^38 –1.  The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.  The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0 | 5-17 bytes |
| smallmoney | Monetary data from -214,748.3648 to 214,748.3647 | 4 bytes |
| money | Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807 | 8 bytes |
| float(n) | Floating precision number data from -1.79E + 308 to 1.79E + 308.  The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53. | 4 or 8 bytes |
| real | Floating precision number data from -3.40E + 38 to 3.40E + 38 | 4 bytes |

### Date and Time Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Description | Storage |
| datetime | From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds | 8 bytes |
| datetime2 | From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds | 6-8 bytes |
| smalldatetime | From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute | 4 bytes |
| date | Store a date only. From January 1, 0001 to December 31, 9999 | 3 bytes |
| time | Store a time only to an accuracy of 100 nanoseconds | 3-5 bytes |
| datetimeoffset | The same as datetime2 with the addition of a time zone offset | 8-10 bytes |
| timestamp | Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable |  |

### Other Data Types

|  |  |
| --- | --- |
| Data type | Description |
| sql\_variant | Stores up to 8,000 bytes of data of various data types, except text, ntext, and timestamp |
| uniqueidentifier | Stores a globally unique identifier (GUID) |
| xml | Stores XML formatted data. Maximum 2GB |
| cursor | Stores a reference to a cursor used for database operations |
| table | Stores a result-set for later processing |

## MS Access Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Description | Storage |
| Text | Use for text or combinations of text and numbers. 255 characters maximum |  |
| Memo | Memo is used for larger amounts of text. Stores up to 65,536 characters. Note: You cannot sort a memo field. However, they are searchable |  |
| Byte | Allows whole numbers from 0 to 255 | 1 byte |
| Integer | Allows whole numbers between -32,768 and 32,767 | 2 bytes |
| Long | Allows whole numbers between -2,147,483,648 and 2,147,483,647 | 4 bytes |
| Single | Single precision floating-point. Will handle most decimals | 4 bytes |
| Double | Double precision floating-point. Will handle most decimals | 8 bytes |
| Currency | Use for currency. Holds up to 15 digits of whole dollars, plus 4 decimal places. Tip: You can choose which country's currency to use | 8 bytes |
| AutoNumber | AutoNumber fields automatically give each record its own number, usually starting at 1 | 4 bytes |
| Date/Time | Use for dates and times | 8 bytes |
| Yes/No | A logical field can be displayed as Yes/No, True/False, or On/Off. In code, use the constants True and False (equivalent to -1 and 0). Note: Null values are not allowed in Yes/No fields | 1 bit |
| Ole Object | Can store pictures, audio, video, or other BLOBs (Binary Large OBjects) | up to 1GB |
| Hyperlink | Contain links to other files, including web pages |  |
| Lookup Wizard | Let you type a list of options, which can then be chosen from a drop-down list | 4 bytes |

## SQL Keywords

|  |  |
| --- | --- |
| Keyword | Description |
| [ADD](https://www.w3schools.com/sql/sql_ref_add.asp) | Adds a column in an existing table |
| [ADD CONSTRAINT](https://www.w3schools.com/sql/sql_ref_add_constraint.asp) | Adds a constraint after a table is already created |
| [ALTER](https://www.w3schools.com/sql/sql_ref_alter.asp) | Adds, deletes, or modifies columns in a table, or changes the data type of a  column in a table |
| [ALTER COLUMN](https://www.w3schools.com/sql/sql_ref_alter_column.asp) | Changes the data type of a column in a table |
| [ALTER TABLE](https://www.w3schools.com/sql/sql_ref_alter_table.asp) | Adds, deletes, or modifies columns in a table |
| [ALL](https://www.w3schools.com/sql/sql_ref_all.asp) | Returns true if all of the subquery values meet the condition |
| [AND](https://www.w3schools.com/sql/sql_ref_and.asp) | Only includes rows where both conditions is true |
| [ANY](https://www.w3schools.com/sql/sql_ref_any.asp) | Returns true if any of the subquery values meet the condition |
| [AS](https://www.w3schools.com/sql/sql_ref_as.asp) | Renames a column or table with an alias |
| [ASC](https://www.w3schools.com/sql/sql_ref_asc.asp) | Sorts the result set in ascending order |
| [BACKUP DATABASE](https://www.w3schools.com/sql/sql_ref_backup_database.asp) | Creates a back up of an existing database |
| [BETWEEN](https://www.w3schools.com/sql/sql_ref_between.asp) | Selects values within a given range |
| [CASE](https://www.w3schools.com/sql/sql_ref_case.asp) | Creates different outputs based on conditions |
| [CHECK](https://www.w3schools.com/sql/sql_ref_check.asp) | A constraint that limits the value that can be placed in a column |
| [COLUMN](https://www.w3schools.com/sql/sql_ref_column.asp) | Changes the data type of a column or deletes a column in a table |
| [CONSTRAINT](https://www.w3schools.com/sql/sql_ref_constraint.asp) | Adds or deletes a constraint |
| [CREATE](https://www.w3schools.com/sql/sql_ref_create.asp) | Creates a database, index, view, table, or procedure |
| [CREATE DATABASE](https://www.w3schools.com/sql/sql_ref_create_database.asp) | Creates a new SQL database |
| [CREATE INDEX](https://www.w3schools.com/sql/sql_ref_create_index.asp) | Creates an index on a table (allows duplicate values) |
| [CREATE OR REPLACE VIEW](https://www.w3schools.com/sql/sql_ref_create_or_replace_view.asp) | Updates a view |
| [CREATE TABLE](https://www.w3schools.com/sql/sql_ref_create_table.asp) | Creates a new table in the database |
| [CREATE PROCEDURE](https://www.w3schools.com/sql/sql_ref_create_procedure.asp) | Creates a stored procedure |
| [CREATE UNIQUE INDEX](https://www.w3schools.com/sql/sql_ref_create_unique_index.asp) | Creates a unique index on a table (no duplicate values) |
| [CREATE VIEW](https://www.w3schools.com/sql/sql_ref_create_view.asp) | Creates a view based on the result set of a SELECT statement |
| [DATABASE](https://www.w3schools.com/sql/sql_ref_database.asp) | Creates or deletes an SQL database |
| [DEFAULT](https://www.w3schools.com/sql/sql_ref_default.asp) | A constraint that provides a default value for a column |
| [DELETE](https://www.w3schools.com/sql/sql_ref_delete.asp) | Deletes rows from a table |
| [DESC](https://www.w3schools.com/sql/sql_ref_desc.asp) | Sorts the result set in descending order |
| [DISTINCT](https://www.w3schools.com/sql/sql_ref_select_distinct.asp) | Selects only distinct (different) values |
| [DROP](https://www.w3schools.com/sql/sql_ref_drop.asp) | Deletes a column, constraint, database, index, table, or view |
| [DROP COLUMN](https://www.w3schools.com/sql/sql_ref_drop_column.asp) | Deletes a column in a table |
| [DROP CONSTRAINT](https://www.w3schools.com/sql/sql_ref_drop_constraint.asp) | Deletes a UNIQUE, PRIMARY KEY, FOREIGN KEY, or CHECK constraint |
| [DROP DATABASE](https://www.w3schools.com/sql/sql_ref_drop_database.asp) | Deletes an existing SQL database |
| [DROP DEFAULT](https://www.w3schools.com/sql/sql_ref_drop_default.asp) | Deletes a DEFAULT constraint |
| [DROP INDEX](https://www.w3schools.com/sql/sql_ref_drop_index.asp) | Deletes an index in a table |
| [DROP TABLE](https://www.w3schools.com/sql/sql_ref_drop_table.asp) | Deletes an existing table in the database |
| [DROP VIEW](https://www.w3schools.com/sql/sql_ref_drop_view.asp) | Deletes a view |
| [EXEC](https://www.w3schools.com/sql/sql_ref_exec.asp) | Executes a stored procedure |
| [EXISTS](https://www.w3schools.com/sql/sql_ref_exists.asp) | Tests for the existence of any record in a subquery |
| [FOREIGN KEY](https://www.w3schools.com/sql/sql_ref_foreign_key.asp) | A constraint that is a key used to link two tables together |
| [FROM](https://www.w3schools.com/sql/sql_ref_from.asp) | Specifies which table to select or delete data from |
| [FULL OUTER JOIN](https://www.w3schools.com/sql/sql_ref_full_outer_join.asp) | Returns all rows when there is a match in either left table or right table |
| [GROUP BY](https://www.w3schools.com/sql/sql_ref_group_by.asp) | Groups the result set (used with aggregate functions: COUNT, MAX, MIN,  SUM, AVG) |
| [HAVING](https://www.w3schools.com/sql/sql_ref_having.asp) | Used instead of WHERE with aggregate functions |
| [IN](https://www.w3schools.com/sql/sql_ref_in.asp) | Allows you to specify multiple values in a WHERE clause |
| [INDEX](https://www.w3schools.com/sql/sql_ref_index.asp) | Creates or deletes an index in a table |
| [INNER JOIN](https://www.w3schools.com/sql/sql_ref_inner_join.asp) | Returns rows that have matching values in both tables |
| [INSERT INTO](https://www.w3schools.com/sql/sql_ref_insert_into.asp) | Inserts new rows in a table |
| [INSERT INTO SELECT](https://www.w3schools.com/sql/sql_ref_insert_into_select.asp) | Copies data from one table into another table |
| [IS NULL](https://www.w3schools.com/sql/sql_ref_is_null.asp) | Tests for empty values |
| [IS NOT NULL](https://www.w3schools.com/sql/sql_ref_is_not_null.asp) | Tests for non-empty values |
| [JOIN](https://www.w3schools.com/sql/sql_ref_join.asp) | Joins tables |
| [LEFT JOIN](https://www.w3schools.com/sql/sql_ref_left_join.asp) | Returns all rows from the left table, and the matching rows from the right table |
| [LIKE](https://www.w3schools.com/sql/sql_ref_like.asp) | Searches for a specified pattern in a column |
| [LIMIT](https://www.w3schools.com/sql/sql_ref_top.asp) | Specifies the number of records to return in the result set |
| [NOT](https://www.w3schools.com/sql/sql_ref_not.asp) | Only includes rows where a condition is not true |
| [NOT NULL](https://www.w3schools.com/sql/sql_ref_not_null.asp) | A constraint that enforces a column to not accept NULL values |
| [OR](https://www.w3schools.com/sql/sql_ref_or.asp) | Includes rows where either condition is true |
| [ORDER BY](https://www.w3schools.com/sql/sql_ref_order_by.asp) | Sorts the result set in ascending or descending order |
| [OUTER JOIN](https://www.w3schools.com/sql/sql_ref_full_outer_join.asp) | Returns all rows when there is a match in either left table or right table |
| [PRIMARY KEY](https://www.w3schools.com/sql/sql_ref_primary_key.asp) | A constraint that uniquely identifies each record in a database table |
| [PROCEDURE](https://www.w3schools.com/sql/sql_ref_create_procedure.asp) | A stored procedure |
| [RIGHT JOIN](https://www.w3schools.com/sql/sql_ref_right_join.asp) | Returns all rows from the right table, and the matching rows from the left table |
| [ROWNUM](https://www.w3schools.com/sql/sql_ref_top.asp) | Specifies the number of records to return in the result set |
| [SELECT](https://www.w3schools.com/sql/sql_ref_select.asp) | Selects data from a database |
| [SELECT DISTINCT](https://www.w3schools.com/sql/sql_ref_select_distinct.asp) | Selects only distinct (different) values |
| [SELECT INTO](https://www.w3schools.com/sql/sql_ref_select_into.asp) | Copies data from one table into a new table |
| [SELECT TOP](https://www.w3schools.com/sql/sql_ref_top.asp) | Specifies the number of records to return in the result set |
| [SET](https://www.w3schools.com/sql/sql_ref_set.asp) | Specifies which columns and values that should be updated in a table |
| [TABLE](https://www.w3schools.com/sql/sql_ref_table.asp) | Creates a table, or adds, deletes, or modifies columns in a table, or deletes a table  or data inside a table |
| [TOP](https://www.w3schools.com/sql/sql_ref_top.asp) | Specifies the number of records to return in the result set |
| [TRUNCATE TABLE](https://www.w3schools.com/sql/sql_ref_drop_table.asp) | Deletes the data inside a table, but not the table itself |
| [UNION](https://www.w3schools.com/sql/sql_ref_union.asp) | Combines the result set of two or more SELECT statements (only distinct  values) |
| [UNION ALL](https://www.w3schools.com/sql/sql_ref_union.asp) | Combines the result set of two or more SELECT statements (allows duplicate  values) |
| [UNIQUE](https://www.w3schools.com/sql/sql_ref_unique.asp) | A constraint that ensures that all values in a column are unique |
| [UPDATE](https://www.w3schools.com/sql/sql_ref_update.asp) | Updates existing rows in a table |
| [VALUES](https://www.w3schools.com/sql/sql_ref_values.asp) | Specifies the values of an INSERT INTO statement |
| [VIEW](https://www.w3schools.com/sql/sql_ref_view.asp) | Creates, updates, or deletes a view |
| [WHERE](https://www.w3schools.com/sql/sql_ref_where.asp) | Filters a result set to include only records that fulfill a specified condition |

## MySQL String Functions

|  |  |
| --- | --- |
| Function | Description |
| [ASCII](https://www.w3schools.com/sql/func_mysql_ascii.asp) | Returns the ASCII value for the specific character |
| [CHAR\_LENGTH](https://www.w3schools.com/sql/func_mysql_char_length.asp) | Returns the length of a string (in characters) |
| [CHARACTER\_LENGTH](https://www.w3schools.com/sql/func_mysql_character_length.asp) | Returns the length of a string (in characters) |
| [CONCAT](https://www.w3schools.com/sql/func_mysql_concat.asp) | Adds two or more expressions together |
| [CONCAT\_WS](https://www.w3schools.com/sql/func_mysql_concat_ws.asp) | Adds two or more expressions together with a separator |
| [FIELD](https://www.w3schools.com/sql/func_mysql_field.asp) | Returns the index position of a value in a list of values |
| [FIND\_IN\_SET](https://www.w3schools.com/sql/func_mysql_find_in_set.asp) | Returns the position of a string within a list of strings |
| [FORMAT](https://www.w3schools.com/sql/func_mysql_format.asp) | Formats a number to a format like "#,###,###.##", rounded to a specified number of  decimal places |
| [INSERT](https://www.w3schools.com/sql/func_mysql_insert.asp) | Inserts a string within a string at the specified position and for a certain number of  characters |
| [INSTR](https://www.w3schools.com/sql/func_mysql_instr.asp) | Returns the position of the first occurrence of a string in another string |
| [LCASE](https://www.w3schools.com/sql/func_mysql_lcase.asp) | Converts a string to lower-case |
| [LEFT](https://www.w3schools.com/sql/func_mysql_left.asp) | Extracts a number of characters from a string (starting from left) |
| [LENGTH](https://www.w3schools.com/sql/func_mysql_length.asp) | Returns the length of a string (in bytes) |
| [LOCATE](https://www.w3schools.com/sql/func_mysql_locate.asp) | Returns the position of the first occurrence of a substring in a string |
| [LOWER](https://www.w3schools.com/sql/func_mysql_lower.asp) | Converts a string to lower-case |
| [LPAD](https://www.w3schools.com/sql/func_mysql_lpad.asp) | Left-pads a string with another string, to a certain length |
| [LTRIM](https://www.w3schools.com/sql/func_mysql_ltrim.asp) | Removes leading spaces from a string |
| [MID](https://www.w3schools.com/sql/func_mysql_mid.asp) | Extracts a substring from a string (starting at any position) |
| [POSITION](https://www.w3schools.com/sql/func_mysql_position.asp) | Returns the position of the first occurrence of a substring in a string |
| [REPEAT](https://www.w3schools.com/sql/func_mysql_repeat.asp) | Repeats a string as many times as specified |
| [REPLACE](https://www.w3schools.com/sql/func_mysql_replace.asp) | Replaces all occurrences of a substring within a string, with a new substring |
| [REVERSE](https://www.w3schools.com/sql/func_mysql_reverse.asp) | Reverses a string and returns the result |
| [RIGHT](https://www.w3schools.com/sql/func_mysql_right.asp) | Extracts a number of characters from a string (starting from right) |
| [RPAD](https://www.w3schools.com/sql/func_mysql_rpad.asp) | Right-pads a string with another string, to a certain length |
| [RTRIM](https://www.w3schools.com/sql/func_mysql_rtrim.asp) | Removes trailing spaces from a string |
| [SPACE](https://www.w3schools.com/sql/func_mysql_space.asp) | Returns a string of the specified number of space characters |
| [STRCMP](https://www.w3schools.com/sql/func_mysql_strcmp.asp) | Compares two strings |
| [SUBSTR](https://www.w3schools.com/sql/func_mysql_substr.asp) | Extracts a substring from a string (starting at any position) |
| [SUBSTRING](https://www.w3schools.com/sql/func_mysql_substring.asp) | Extracts a substring from a string (starting at any position) |
| [SUBSTRING\_INDEX](https://www.w3schools.com/sql/func_mysql_substring_index.asp) | Returns a substring of a string before a specified number of delimiter occurs |
| [TRIM](https://www.w3schools.com/sql/func_mysql_trim.asp) | Removes leading and trailing spaces from a string |
| [UCASE](https://www.w3schools.com/sql/func_mysql_ucase.asp) | Converts a string to upper-case |
| [UPPER](https://www.w3schools.com/sql/func_mysql_upper.asp) | Converts a string to upper-case |

## MySQL Numeric Functions

|  |  |
| --- | --- |
| Function | Description |
| [ABS](https://www.w3schools.com/sql/func_mysql_abs.asp) | Returns the absolute value of a number |
| [ACOS](https://www.w3schools.com/sql/func_mysql_acos.asp) | Returns the arc cosine of a number |
| [ASIN](https://www.w3schools.com/sql/func_mysql_asin.asp) | Returns the arc sine of a number |
| [ATAN](https://www.w3schools.com/sql/func_mysql_atan.asp) | Returns the arc tangent of one or two numbers |
| [ATAN2](https://www.w3schools.com/sql/func_mysql_atan2.asp) | Returns the arc tangent of two numbers |
| [AVG](https://www.w3schools.com/sql/func_mysql_avg.asp) | Returns the average value of an expression |
| [CEIL](https://www.w3schools.com/sql/func_mysql_ceil.asp) | Returns the smallest integer value that is >= to a number |
| [CEILING](https://www.w3schools.com/sql/func_mysql_ceiling.asp) | Returns the smallest integer value that is >= to a number |
| [COS](https://www.w3schools.com/sql/func_mysql_cos.asp) | Returns the cosine of a number |
| [COT](https://www.w3schools.com/sql/func_mysql_cot.asp) | Returns the cotangent of a number |
| [COUNT](https://www.w3schools.com/sql/func_mysql_count.asp) | Returns the number of records returned by a select query |
| [DEGREES](https://www.w3schools.com/sql/func_mysql_degrees.asp) | Converts a value in radians to degrees |
| [DIV](https://www.w3schools.com/sql/func_mysql_div.asp) | Used for integer division |
| [EXP](https://www.w3schools.com/sql/func_mysql_exp.asp) | Returns e raised to the power of a specified number |
| [FLOOR](https://www.w3schools.com/sql/func_mysql_floor.asp) | Returns the largest integer value that is <= to a number |
| [GREATEST](https://www.w3schools.com/sql/func_mysql_greatest.asp) | Returns the greatest value of the list of arguments |
| [LEAST](https://www.w3schools.com/sql/func_mysql_least.asp) | Returns the smallest value of the list of arguments |
| [LN](https://www.w3schools.com/sql/func_mysql_ln.asp) | Returns the natural logarithm of a number |
| [LOG](https://www.w3schools.com/sql/func_mysql_log.asp) | Returns the natural logarithm of a number, or the logarithm of a number to a specified base |
| [LOG10](https://www.w3schools.com/sql/func_mysql_log10.asp) | Returns the natural logarithm of a number to base 10 |
| [LOG2](https://www.w3schools.com/sql/func_mysql_log2.asp) | Returns the natural logarithm of a number to base 2 |
| [MAX](https://www.w3schools.com/sql/func_mysql_max.asp) | Returns the maximum value in a set of values |
| [MIN](https://www.w3schools.com/sql/func_mysql_min.asp) | Returns the minimum value in a set of values |
| [MOD](https://www.w3schools.com/sql/func_mysql_mod.asp) | Returns the remainder of a number divided by another number |
| [PI](https://www.w3schools.com/sql/func_mysql_pi.asp) | Returns the value of PI |
| [POW](https://www.w3schools.com/sql/func_mysql_pow.asp) | Returns the value of a number raised to the power of another number |
| [POWER](https://www.w3schools.com/sql/func_mysql_power.asp) | Returns the value of a number raised to the power of another number |
| [RADIANS](https://www.w3schools.com/sql/func_mysql_radians.asp) | Converts a degree value into radians |
| [RAND](https://www.w3schools.com/sql/func_mysql_rand.asp) | Returns a random number |
| [ROUND](https://www.w3schools.com/sql/func_mysql_round.asp) | Rounds a number to a specified number of decimal places |
| [SIGN](https://www.w3schools.com/sql/func_mysql_sign.asp) | Returns the sign of a number |
| [SIN](https://www.w3schools.com/sql/func_mysql_sin.asp) | Returns the sine of a number |
| [SQRT](https://www.w3schools.com/sql/func_mysql_sqrt.asp) | Returns the square root of a number |
| [SUM](https://www.w3schools.com/sql/func_mysql_sum.asp) | Calculates the sum of a set of values |
| [TAN](https://www.w3schools.com/sql/func_mysql_tan.asp) | Returns the tangent of a number |
| [TRUNCATE](https://www.w3schools.com/sql/func_mysql_truncate.asp) | Truncates a number to the specified number of decimal places |

## MySQL Date Functions

|  |  |
| --- | --- |
| Function | Description |
| [ADDDATE](https://www.w3schools.com/sql/func_mysql_adddate.asp) | Adds a time/date interval to a date and then returns the date |
| [ADDTIME](https://www.w3schools.com/sql/func_mysql_addtime.asp) | Adds a time interval to a time/datetime and then returns the time/datetime |
| [CURDATE](https://www.w3schools.com/sql/func_mysql_curdate.asp) | Returns the current date |
| [CURRENT\_DATE](https://www.w3schools.com/sql/func_mysql_current_date.asp) | Returns the current date |
| [CURRENT\_TIME](https://www.w3schools.com/sql/func_mysql_current_time.asp) | Returns the current time |
| [CURRENT\_TIMESTAMP](https://www.w3schools.com/sql/func_mysql_current_timestamp.asp) | Returns the current date and time |
| [CURTIME](https://www.w3schools.com/sql/func_mysql_curtime.asp) | Returns the current time |
| [DATE](https://www.w3schools.com/sql/func_mysql_date.asp) | Extracts the date part from a datetime expression |
| [DATEDIFF](https://www.w3schools.com/sql/func_mysql_datediff.asp) | Returns the number of days between two date values |
| [DATE\_ADD](https://www.w3schools.com/sql/func_mysql_date_add.asp) | Adds a time/date interval to a date and then returns the date |
| [DATE\_FORMAT](https://www.w3schools.com/sql/func_mysql_date_format.asp) | Formats a date |
| [DATE\_SUB](https://www.w3schools.com/sql/func_mysql_date_sub.asp) | Subtracts a time/date interval from a date and then returns the date |
| [DAY](https://www.w3schools.com/sql/func_mysql_day.asp) | Returns the day of the month for a given date |
| [DAYNAME](https://www.w3schools.com/sql/func_mysql_dayname.asp) | Returns the weekday name for a given date |
| [DAYOFMONTH](https://www.w3schools.com/sql/func_mysql_dayofmonth.asp) | Returns the day of the month for a given date |
| [DAYOFWEEK](https://www.w3schools.com/sql/func_mysql_dayofweek.asp) | Returns the weekday index for a given date |
| [DAYOFYEAR](https://www.w3schools.com/sql/func_mysql_dayofyear.asp) | Returns the day of the year for a given date |
| [EXTRACT](https://www.w3schools.com/sql/func_mysql_extract.asp) | Extracts a part from a given date |
| [FROM\_DAYS](https://www.w3schools.com/sql/func_mysql_from_days.asp) | Returns a date from a numeric datevalue |
| [HOUR](https://www.w3schools.com/sql/func_mysql_hour.asp) | Returns the hour part for a given date |
| [LAST\_DAY](https://www.w3schools.com/sql/func_mysql_last_day.asp) | Extracts the last day of the month for a given date |
| [LOCALTIME](https://www.w3schools.com/sql/func_mysql_localtime.asp) | Returns the current date and time |
| [LOCALTIMESTAMP](https://www.w3schools.com/sql/func_mysql_localtimestamp.asp) | Returns the current date and time |
| [MAKEDATE](https://www.w3schools.com/sql/func_mysql_makedate.asp) | Creates and returns a date based on a year and a number of days value |
| [MAKETIME](https://www.w3schools.com/sql/func_mysql_maketime.asp) | Creates and returns a time based on an hour, minute, and second value |
| [MICROSECOND](https://www.w3schools.com/sql/func_mysql_microsecond.asp) | Returns the microsecond part of a time/datetime |
| [MINUTE](https://www.w3schools.com/sql/func_mysql_minute.asp) | Returns the minute part of a time/datetime |
| [MONTH](https://www.w3schools.com/sql/func_mysql_month.asp) | Returns the month part for a given date |
| [MONTHNAME](https://www.w3schools.com/sql/func_mysql_monthname.asp) | Returns the name of the month for a given date |
| [NOW](https://www.w3schools.com/sql/func_mysql_now.asp) | Returns the current date and time |
| [PERIOD\_ADD](https://www.w3schools.com/sql/func_mysql_period_add.asp) | Adds a specified number of months to a period |
| [PERIOD\_DIFF](https://www.w3schools.com/sql/func_mysql_period_diff.asp) | Returns the difference between two periods |
| [QUARTER](https://www.w3schools.com/sql/func_mysql_quarter.asp) | Returns the quarter of the year for a given date value |
| [SECOND](https://www.w3schools.com/sql/func_mysql_second.asp) | Returns the seconds part of a time/datetime |
| [SEC\_TO\_TIME](https://www.w3schools.com/sql/func_mysql_sec_to_time.asp) | Returns a time value based on the specified seconds |
| [STR\_TO\_DATE](https://www.w3schools.com/sql/func_mysql_str_to_date.asp) | Returns a date based on a string and a format |
| [SUBDATE](https://www.w3schools.com/sql/func_mysql_subdate.asp) | Subtracts a time/date interval from a date and then returns the date |
| [SUBTIME](https://www.w3schools.com/sql/func_mysql_subtime.asp) | Subtracts a time interval from a datetime and then returns the time/datetime |
| [SYSDATE](https://www.w3schools.com/sql/func_mysql_sysdate.asp) | Returns the current date and time |
| [TIME](https://www.w3schools.com/sql/func_mysql_time.asp) | Extracts the time part from a given time/datetime |
| [TIME\_FORMAT](https://www.w3schools.com/sql/func_mysql_time_format.asp) | Formats a time by a specified format |
| [TIME\_TO\_SEC](https://www.w3schools.com/sql/func_mysql_time_to_sec.asp) | Converts a time value into seconds |
| [TIMEDIFF](https://www.w3schools.com/sql/func_mysql_timediff.asp) | Returns the difference between two time/datetime expressions |
| [TIMESTAMP](https://www.w3schools.com/sql/func_mysql_timestamp.asp) | Returns a datetime value based on a date or datetime value |
| [TO\_DAYS](https://www.w3schools.com/sql/func_mysql_to_days.asp) | Returns the number of days between a date and date "0000-00-00" |
| [WEEK](https://www.w3schools.com/sql/func_mysql_week.asp) | Returns the week number for a given date |
| [WEEKDAY](https://www.w3schools.com/sql/func_mysql_weekday.asp) | Returns the weekday number for a given date |
| [WEEKOFYEAR](https://www.w3schools.com/sql/func_mysql_weekofyear.asp) | Returns the week number for a given date |
| [YEAR](https://www.w3schools.com/sql/func_mysql_year.asp) | Returns the year part for a given date |
| [YEARWEEK](https://www.w3schools.com/sql/func_mysql_yearweek.asp) | Returns the year and week number for a given date |

## MySQL Advanced Functions

|  |  |
| --- | --- |
| Function | Description |
| [BIN](https://www.w3schools.com/sql/func_mysql_bin.asp) | Returns a binary representation of a number |
| [BINARY](https://www.w3schools.com/sql/func_mysql_binary.asp) | Converts a value to a binary string |
| [CASE](https://www.w3schools.com/sql/func_mysql_case.asp) | Goes through conditions and return a value when the first condition is met |
| [CAST](https://www.w3schools.com/sql/func_mysql_cast.asp) | Converts a value (of any type) into a specified datatype |
| [COALESCE](https://www.w3schools.com/sql/func_mysql_coalesce.asp) | Returns the first non-null value in a list |
| [CONNECTION\_ID](https://www.w3schools.com/sql/func_mysql_connection_id.asp) | Returns the unique connection ID for the current connection |
| [CONV](https://www.w3schools.com/sql/func_mysql_conv.asp) | Converts a number from one numeric base system to another |
| [CONVERT](https://www.w3schools.com/sql/func_mysql_convert.asp) | Converts a value into the specified datatype or character set |
| [CURRENT\_USER](https://www.w3schools.com/sql/func_mysql_current_user.asp) | Returns the user name and host name for the MySQL account that the server used to  authenticate the current client |
| [DATABASE](https://www.w3schools.com/sql/func_mysql_database.asp) | Returns the name of the current database |
| [IF](https://www.w3schools.com/sql/func_mysql_if.asp) | Returns a value if a condition is TRUE, or another value if a condition is FALSE |
| [IFNULL](https://www.w3schools.com/sql/func_mysql_ifnull.asp) | Return a specified value if the expression is NULL, otherwise return the expression |
| [ISNULL](https://www.w3schools.com/sql/func_mysql_isnull.asp) | Returns 1 or 0 depending on whether an expression is NULL |
| [LAST\_INSERT\_ID](https://www.w3schools.com/sql/func_mysql_last_insert_id.asp) | Returns the AUTO\_INCREMENT id of the last row that has been inserted or updated in a  table |
| [NULLIF](https://www.w3schools.com/sql/func_mysql_nullif.asp) | Compares two expressions and returns NULL if they are equal. Otherwise, the first  expression is returned |
| [SESSION\_USER](https://www.w3schools.com/sql/func_mysql_session_user.asp) | Returns the current MySQL user name and host name |
| [SYSTEM\_USER](https://www.w3schools.com/sql/func_mysql_system_user.asp) | Returns the current MySQL user name and host name |
| [USER](https://www.w3schools.com/sql/func_mysql_user.asp) | Returns the current MySQL user name and host name |
| [VERSION](https://www.w3schools.com/sql/func_mysql_version.asp) | Returns the current version of the MySQL database |