1. show databases;
2. create database dbms;

create database databasename;

1. use dbms;
2. drop database databasename;

Data Type

|  |  |  |
| --- | --- | --- |
| **Data type** | **Range** | **Storage** |
| **bigint** | -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807) | 8 Bytes |
| **int** | -2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647) | 4 Bytes |
| **smallint** | -2^15 (-32,768) to 2^15-1 (32,767) | 2 Bytes |
| **tinyint** | 0 to 255 | 1 Byte |

Use **char** when the sizes of the column data entries are consistent.

Use **varchar** when the sizes of the column data entries vary considerably.

Use **varchar(max)** when the sizes of the column data entries vary considerably, and the string length might exceed 8,000 bytes.

1. create table ;

create table student(studentid int,

lastname varchar(255),

firstname varchar(255),

address varchar(255),

city varchar(255)

);

1. create table from other table;

create table employee as select studentid, lastname from student;

1. drop table student;
2. show tables;
3. Alter table – add column

alter table customers add email varchar(255);

Alter Table – drop column

alter table customers drop email;

1. Alter table - alter/modify column

Modify the datatype of the attribute:

alter table student modify column studentid varchar(100);

11.Change the name of the attribute

alter table student change stdid rollno int;

12 NOT null constraint

create table persons (

id int not null,

lastname varchar(255) not null,

firstname varchar(255) not null,

age int

);

alter table persons modify age int not null;

13 Insert command

insert into customers (customername, city, country) values ('cardinal', 'stavanger', 'norway');

14 select command

select \* from student;

select id,name from sugan;

15 MySQL COUNT() function

select DISTINCT addr from sugan;

select COUNT(\*) FROM sugan;

select count(DISTINCT addr) from sugan;

16 SQL WHERE Clause

Select \* from customer where addr=’salem’;

17 The SQL AND, OR and NOT Operators

18 SELECT \* FROM Customers

WHERE Country='Germany' AND City='Berlin';

19 SELECT \* FROM Customers

WHERE City='Berlin' OR City='München';

20 SELECT \* FROM Customers

WHERE NOT Country='Germany';

21 SELECT \* FROM Customers

WHERE Country='Germany' AND (City='Berlin' OR City='München');

The following SQL statement selects all fields from "Customers" where country is "Germany" AND city must be "Berlin" OR "München" (use parenthesis to form complex expressions):

22 SELECT \* FROM Customers

WHERE NOT Country='Germany' AND NOT Country='USA';

The following SQL statement selects all fields from "Customers" where country is NOT "Germany" and NOT "USA":

UPDATE COMMAND

23 UPDATE Customers SET ContactName = 'Alfred Schmidt', City= 'Frankfurt' WHERE CustomerID = 1;

## UPDATE Multiple Records

24 UPDATE sugan SET name=’suganya' WHERE addr='salem';

## The SQL DELETE Statement

DELETE FROM table\_name WHERE condition;

DELETE FROM sugan WHERE id=’101’;

Delete from student;

SINGLE ROW FUNCTION

1. My MySQL ASCII() Function

ASCII(str1)

SELECT ASCII(CustomerName) AS NumCodeOfFirstChar FROM Customers;

SELECT ASCII('b')AS Lower\_Case, ASCII('B') AS Upper\_Case;

1. CHAR\_LENGTH() Function

SELECT CHAR\_LENGTH("SQLTutorial") AS LengthOfString;

SELECT CHAR\_LENGTH(CustomerName) AS LengthOfName FROM Customers;

1. MySQL CONCAT() Function

SELECT CONCAT("SQL", "Tutorial", "is", "fun!") AS ConcatenatedString;

SELECT CONCAT(Address, " ", PostalCode, " ", City) AS Address  
FROM Customers;

1. MySQL FORMAT() Function

SELECT FORMAT(250500.5634, 2);

1. MySQL LCASE() Function

SELECT LCASE("SQL Tutorial ");

SELECT LCASE(CustomerName) AS LowercaseCustomerName

FROM Customers;

MySQL LOWER() Function

SELECT LOWER("SQL Tutorial");

1. MySQL REPLACE() Function

SELECT REPLACE("SQL Tutorial", "SQL", "HTML");

SELECT REPLACE("XYZ FGH XYZ", "X", "M");

1. MySQL LTRIM() Function

SELECT LTRIM("     SQLTutorial") AS LeftTrimmedString;

1. MySQL UCASE() Function

SELECT UCASE("SQL Tutorial ");

SELECT UPPER("SQL Tutorial ");

1. MySQL RTRIM() Function

SELECT RTRIM("SQL Tutorial ") AS RightTrimmedString;

1. MySQL TRIM() Function

Remove leading and trailing spaces from a string:

SELECT TRIM(' SQL Tutorial ') AS TrimmedString;

1. MySQL SUBSTR() Function

Extract a substring from a string (start at position 5, extract 3 characters):

SELECT SUBSTR("SQL Tutorial", 5, 3) AS ExtractString;

1. MySQL STRCMP() Function

Compare two strings:

SELECT STRCMP("SQL Tutorial", "SQL Tutorial");

* If *string1* = *string2*, this function returns 0
* If *string1* < *string2*, this function returns -1
* If *string1* > *string2*, this function returns 1

DATE FUNCTION

create table DemoTable

   (

   StudentId int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

   StudentName varchar(20),

   StudentAdmissionDate DATE

   );

Insert into Demotable values(101,’sai’,’2021-06-24’);

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

1. ADDDATE() Function

The ADDDATE() function adds a time/date interval to a date and then returns the date.

Syntax

ADDDATE(date, INTERVAL value addunit)

### Example

Add 15 minutes to a date and return the date:

SELECT ADDDATE("2017-06-15 09:34:21", INTERVAL 15 MINUTE);

Add 10 days to a date and return the date:

SELECT ADDDATE("2017-06-15", INTERVAL 10 DAY);

Subtract 2 months to a date and return the date:

SELECT ADDDATE("2017-06-15", INTERVAL -2 MONTH);

Add 2 hours, 10 minutes, 5 seconds, and 3 microseconds to a time and return the datetime:

SELECT ADDTIME("2017-06-15 09:34:21.000001", "2:10:5.000003");

1. CURDATE() Function

SELECT CURDATE();

SELECT CURDATE() + 1;

SELECT CURRENT\_DATE();

1. CURRENT\_TIME() Function

SELECT CURRENT\_TIME();

1. The CURRENT\_TIMESTAMP() function returns the current date and time.

SELECT CURRENT\_TIMESTAMP();

1. DATEDIFF() Function

Return the number of days between two date values:

SELECT DATEDIFF("2017-06-25", "2017-06-15");

SELECT DATEDIFF("2017-06-25 09:34:21", "2017-06-15 15:25:35");

1. DATE\_FORMAT() Function

Format Description

%a Abbreviated weekday name (Sun to Sat)

%b Abbreviated month name (Jan to Dec)

%c Numeric month name (0 to 12)

%D Day of the month as a numeric value, followed by suffix (1st, 2nd, 3rd, ...)

%d Day of the month as a numeric value (01 to 31)

%e Day of the month as a numeric value (0 to 31)

%f Microseconds (000000 to 999999)

%H Hour (00 to 23)

%h Hour (00 to 12)

%I Hour (00 to 12)

%i Minutes (00 to 59)

%j Day of the year (001 to 366)

%k Hour (0 to 23)

%l Hour (1 to 12)

%M Month name in full (January to December)

%m Month name as a numeric value (00 to 12)

%p AM or PM

%r Time in 12 hour AM or PM format (hh:mm:ss AM/PM)

%S Seconds (00 to 59)

%s Seconds (00 to 59)

%T Time in 24 hour format (hh:mm:ss)

SELECT DATE\_FORMAT("2017-06-15", "%M %d %Y");

SELECT DATE\_FORMAT("2017-06-15", "%W %M %e %Y");

SELECT DATE\_FORMAT(BirthDate, "%W %M %e %Y") FROM Employees;

1. DATE\_SUB() Function

The DATE\_SUB() function subtracts a time/date interval from a date and then returns the date.

DATE\_SUB(*date*, INTERVAL *value interval*)

Subtract 15 minutes from a date and return the date:

SELECT DATE\_SUB("2017-06-15 09:34:21", INTERVAL 15 MINUTE);

1. DAY() Function

The DAY() function returns the day of the month for a given date (a number from 1 to 31).

SELECT DAY("2017-06-15 09:34:21");

SELECT DAY(CURDATE());

1. DAYNAME() Function

The DAYNAME() function returns the weekday name for a given date.

SELECT DAYNAME("2017-06-15 09:34:21");

SELECT DAYNAME(CURDATE());

1. SELECT DAYNAME(CURDATE());

Return a date from a numeric representation of the day:

SELECT FROM\_DAYS(685467);

1. TIME\_FORMAT() Function

The TIME\_FORMAT() function formats a time by a specified format.

Format Description

%f Microseconds (000000 to 999999)

%H Hour (00 to 23)

%h Hour (00 to 12)

%I Hour (00 to 12)

%i Minutes (00 to 59)

%p AM or PM

%r Time in 12 hour AM or PM format (hh:mm:ss AM/PM)

%S Seconds (00 to 59)

%s Seconds (00 to 59)

%T Time in 24 hour format (hh:mm:ss)

SELECT TIME\_FORMAT("19:30:10", "%h %i %s %p");

SELECT TIME\_FORMAT("19:30:10", "%r");

SELECT TIME\_FORMAT("19:30:10", "%T");

1. WEEK() Function

The WEEK() function returns the week number for a given date (a number from 0 to 53).

SELECT WEEK("2017-10-25");

SELECT WEEK(CURDATE());

1. WEEKDAY() Function

The WEEKDAY() function returns the weekday number for a given date.

Note: 0 = Monday, 1 = Tuesday, 2 = Wednesday, 3 = Thursday, 4 = Friday, 5 = Saturday, 6 = Sunday.

SELECT WEEKDAY("2017-01-01");