

MySQL Data Types

Numeric Data Type

Data Type Syntax	Description
TINYINT	It is a very small integer that can be signed or unsigned. If signed, the allowable range is from -128 to 127. If unsigned, the allowable range is from 0 to 255. We can specify a width of up to 4 digits. It takes 1 byte for storage.
SMALLINT	It is a small integer that can be signed or unsigned. If signed, the allowable range is from -32768 to 32767. If unsigned, the allowable range is from 0 to 65535. We can specify a width of up to 5 digits. It requires 2 bytes for storage.
MEDIUMINT	It is a medium-sized integer that can be signed or unsigned. If signed, the allowable range is from -8388608 to 8388607. If unsigned, the allowable range is from 0 to 16777215. We can specify a width of up to 9 digits. It requires 3 bytes for storage.
INT	It is a normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647. If unsigned, the allowable range is from 0 to 4294967295. We can specify a width of up to 11 digits. It requires 4 bytes for storage.
BIGINT	It is a large integer that can be signed or unsigned. If signed, the allowable range is from -9223372036854775808 to 9223372036854775807. If unsigned, the allowable range is from 0 to 18446744073709551615. We can specify a width of up to 20 digits. It requires 8 bytes for storage.
FLOAT(m,d)	It is a floating-point number that cannot be unsigned. You can define the display length (m) and the number of decimals (d). This is not required and will default to 10,2, where 2 is the number of decimals, and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a float type. It requires 2 bytes for storage.
DOUBLE(m,d)	It is a double-precision floating-point number that cannot be unsigned. You can

	define the display length (m) and the number of decimals (d). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a double. Real is a synonym for double. It requires 8 bytes for storage.
DECIMAL(m,d)	An unpacked floating-point number that cannot be unsigned. In unpacked decimals, each decimal corresponds to one byte. Defining the display length (m) and the number of decimals (d) is required. Numeric is a synonym for decimal.
BIT(m)	It is used for storing bit values into the table column. Here, M determines the number of bit per value that has a range of 1 to 64.
BOOL	It is used only for the true and false condition. It considered numeric value 1 as true and 0 as false.
BOOLEAN	It is Similar to the BOOL.

Date and Time Data Type:

Data Type Syntax	Maximum Size	Explanation
YEAR[(2 4)]	Year value as 2 digits or 4 digits.	The default is 4 digits. It takes 1 byte for storage.
DATE	Values range from '1000-01-01' to '9999-12-31'.	Displayed as 'yyyy-mm-dd'. It takes 3 bytes for storage.
TIME	Values range from '-838:59:59' to '838:59:59'.	Displayed as 'HH:MM:SS'. It takes 3 bytes plus fractional seconds for storage.
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'. It takes 5 bytes plus fractional seconds for storage.
TIMESTAMP(m)	Values range from '1970-01-01 00:00:01' UTC to '2038-01-19'	Displayed as 'YYYY-MM-DD HH:MM:SS'. It takes 4 bytes plus fractional seconds for

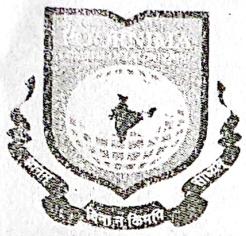
03:14:07' TC.

storage.

String Data Types:

Data Type Syntax	Maximum Size	Explanation
CHAR(size)	It can have a maximum size of 255 characters.	Here size is the number of characters to store. Fixed-length strings. Space padded on the right to equal size characters.
VARCHAR(size)	It can have a maximum size of 255 characters.	Here size is the number of characters to store. Variable-length string.
TINYTEXT(size)	It can have a maximum size of 255 characters.	Here size is the number of characters to store.
TEXT(size)	Maximum size of 65,535 characters.	Here size is the number of characters to store.
MEDIUMTEXT(size)	It can have a maximum size of 16,777,215 characters.	Here size is the number of characters to store.
LONGTEXT(size)	It can have a maximum size of 4GB or 4,294,967,295 characters.	Here size is the number of characters to store.
BINARY(size)	It can have a maximum size of 255 characters.	Here size is the number of binary characters to store. Fixed-length strings. Space padded on the right to equal size characters. (introduced in MySQL 4.1.2)
VARBINARY(size)	It can have a maximum size of 255 characters.	Here size is the number of characters to store. Variable-length string. (introduced in MySQL 4.1.2)

ENUM	<p>It takes 1 or 2 bytes that depend on the number of enumeration values. An ENUM can have a maximum of 65,535 values.</p>	<p>It is short for enumeration, which means that each column may have one of the specified possible values. It uses numeric indexes (1, 2, 3...) to represent string values.</p>
SET	<p>It takes 1, 2, 3, 4, or 8 bytes that depends on the number of set members. It can store a maximum of 64 members.</p>	<p>It can hold zero or more, or any number of string values. They must be chosen from a predefined list of values specified during table creation.</p>



POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

PAGE NO.

Exp 2

DBMS lab

SQL : Commands.

- ① cd
- ② if cdxampp
- ③ mysql > cd bin > MySQL

→

- ④ Create database database_name ;
- ⑤ Use database_name ;
- ⑥ Create database college ;
- ⑦ Show database ;
- ⑧ use college ;
database changed
- ⑨ Create table student (enroll int, name varchar(6), branch char(3))
- ⑩ desc student ;
- ⑪ alter table student add result int ;
- ⑫ desc student ;
- ⑬ alter table student add address varchar(10) after name ;
- ⑭ desc student ;
- ⑮ alter table drop column id ;
- ⑯ desc student ;
- ⑰ alter table student drop column branch, drop column result ;
- ⑱ alter table student modify address varchar(12) ;
- ⑲ alter table student change address varchar(12) ;
- ⑳ desc student

```
1.CREATE DATABASE employeesdb;
2. SHOW DATABASES;
3. USE employeesdb;
4. DROP DATABASE employeesdb;
5. CREATE TABLE employee_table(
    id int PRIMARY KEY ,
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    age int NOT NULL,
);
6. SHOW TABLES;
7. Desc employee_table;
8. CREATE TABLE employee_table(
    id int PRIMARY KEY ,
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    age int NOT NULL,
);
9. ALTER TABLE cus_tbl
    ADD cus_age varchar(40) NOT NULL;
10. ALTER TABLE cus_tbl
    ADD cus_address varchar(100) NOT NULL
        AFTER cus_surname,
    ADD cus_salary int(100) NOT NULL
        AFTER cus_age ;
11. ALTER TABLE cus_tbl
    MODIFY cus_surname varchar(50) NULL;
12. ALTER TABLE cus_tbl
    DROP COLUMN cus_address;
13. ALTER TABLE cus_tbl
    CHANGE COLUMN cus_surname cus_title
        varchar(20) NOT NULL;
14. ALTER TABLE cus_tbl
    RENAME TO cus_table;
15. RENAME employee TO customer;
16. RENAME TABLE customer TO employee;
17. CREATE TABLE customer (
    Id int PRIMARY KEY NOT NULL,
    Name varchar(45) NOT NULL,
    Product varchar(45) DEFAULT NULL,
    Country varchar(25) DEFAULT NULL,
    Year int NOT NULL
);
18.
    INSERT INTO customer ( Id, Name, Product, C
ountry, Year)
    VALUES:(1,'Stephen', 'Computer', 'USA', 2015
),
(2, 'Joseph', 'Laptop', 'India', 2016),
(3, 'John', 'TV', 'USA', 2016),
(4, 'Donald', 'Laptop', 'England', 2015),
(5, 'Joseph', 'Mobile', 'India', 2015),
(6, 'Peter', 'Mouse', 'England', 2016);
19. SELECT * FROM customer;
```

20.
Example

This example specifies how we can drop an existing table from the database. : "orders" as shown in the image below.

The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client". It displays the result of a SQL query: "mysql> SELECT * FROM orders;". The output is a table with three columns: "order_id", "prod_name", and "price". The data rows are: 1 | Laptop | 80000, 2 | Mouse | 3000, 3 | Desktop | 50000, 4 | Iphone | 50000. A message at the bottom says "4 rows in set (0.00 sec)".

order_id	prod_name	price
1	Laptop	80000
2	Mouse	3000
3	Desktop	50000
4	Iphone	50000

To delete the above table, we need to run the following statement:

```
mysql> DROP TABLE orders;
```

21. CREATE TABLE Test (
 Stude_id int AUTO_INCREMENT PRIMARY
 KEY,
 Name varchar(55) NOT NULL
);

22. ALTER TABLE Test

```
ADD COLUMN City VARCHAR(30) NOT N  
ULL;
```

23. ALTER TABLE Test

```
ADD COLUMN Branch VARCHAR(30) DEF  
AULT NULL After Name,  
ADD COLUMN Email VARCHAR(20) DEFA  
ULT NULL AFTER Phone_number;
```

24. ALTER TABLE Test

```
CHANGE COLUMN Phone_number Mobile_  
number  
        varchar(20) NOT NULL;
```

25.

ALTER TABLE Test DROP COLUMN Branch;

26. CREATE TABLE People(

```
    id int NOT NULL AUTO_INCREMENT,  
    name varchar(45) NOT NULL,  
    occupation varchar(35) NOT NULL,  
    age int,  
    PRIMARY KEY (id)
```

);

27. INSERT INTO People VALUES

```
(102, 'Joseph', 'Developer', 30),
```

```
(103, 'Mike', 'Leader', 28),
```

```
(104, 'Stephen', 'Scientist', 45);
```

28.

The screenshot shows a terminal window titled "MySQL 8.0 Command Line Client". It displays the result of a SQL query: "mysql> SELECT * FROM trainer;". The output is a table with three columns: "course_name", "trainer", and "email". The data rows are: Java | Mike | mike@javatpoint.com, Python | James | james@javatpoint.com, Android | Robin | robin@javatpoint.com, Hadoop | Stephen | stephen@javatpoint.com, Testing | Micheal | micheal@javatpoint.com.

course_name	trainer	email
Java	Mike	mike@javatpoint.com
Python	James	james@javatpoint.com
Android	Robin	robin@javatpoint.com
Hadoop	Stephen	stephen@javatpoint.com
Testing	Micheal	micheal@javatpoint.com

29. UPDATE trainer

```
SET email = 'mike@tutorialandexamples.com'
```

```
WHERE course_name = 'Java';
```

30. UPDATE People

```
SET name = 'Mary', occupation = 'Content Writ  
er'
```

```
WHERE id = 105;
```

31.

```
DELETE FROM Employees WHERE emp_id=107;
```

32. truncate table customers;

33. drop table customers;