**Schema\_Table**

**Syntax : 1**

SELECT COLUMN\_NAME, DATA\_TYPE, IS\_NULLABLE, COLUMN\_DEFAULT

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name = '*tbl\_name*'

[AND table\_schema = '*db\_name*']

[AND column\_name LIKE '*wild*']

SHOW COLUMNS

FROM *tbl\_name*

[FROM *db\_name*]

[LIKE '*wild*']

**Syntax : 2**

select SCHEMA\_NAME();

SELECT \* FROM INFORMATION\_SCHEMA.TABLES;

SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS;

**Syntax : 3**

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Description |
| TABLE\_CATALOG | nvarchar(128) | Table qualifier. |
| TABLE\_SCHEMA | nvarchar(128) | Name of schema that contains the table. |
| TABLE\_NAME | nvarchar(128) | Table name. |
| COLUMN\_NAME | nvarchar(128) | Column name. |
| ORDINAL\_POSITION | int | Column identification number.  *Note: In SQL Server 2005, these column IDs are consecutive numbers.* |
| COLUMN\_DEFAULT | nvarchar(4000) | Default value of the column. |
| IS\_NULLABLE | varchar(3) | Nullability of the column. If this column allows for NULL, this column returns YES. Otherwise, NO is returned. |
| DATA\_TYPE | nvarchar(128) | System-supplied data type. |
| CHARACTER\_MAXIMUM\_LENGTH | int | Maximum length, in characters, for binary data, character data, or text and image data.  -1 for xml and large-value type data. Otherwise, NULL is returned. For more information, see [Data Types (Transact-SQL)](ms-help://MS.SQLCC.v9/MS.SQLSVR.v9.en/tsqlref9/html/a54f7373-b247-4d61-8fb8-7f2ec7a8d0a4.htm). |
| CHARACTER\_OCTET\_LENGTH | int | Maximum length, in bytes, for binary data, character data, or text and image data.  -1 for xml and large-value type data. Otherwise, NULL is returned. |
| NUMERIC\_PRECISION | tinyint | Precision of approximate numeric data, exact numeric data, integer data, or monetary data. Otherwise, NULL is returned. |
| NUMERIC\_PRECISION\_RADIX | smallint | Precision radix of approximate numeric data, exact numeric data, integer data, or monetary data. Otherwise, NULL is returned. |
| NUMERIC\_SCALE | int | Scale of approximate numeric data, exact numeric data, integer data, or monetary data. Otherwise, NULL is returned. |
| DATETIME\_PRECISION | smallint | Subtype code for datetime and SQL-92 interval data types. For other data types, NULL is returned. |
| CHARACTER\_SET\_CATALOG | nvarchar(128) | Returns master. This indicates the database in which the character set is located, if the column is character data or text data type. Otherwise, NULL is returned. |
| CHARACTER\_SET\_SCHEMA | nvarchar(128) | Always returns NULL. |
| CHARACTER\_SET\_NAME | nvarchar(128) | Returns the unique name for the character set if this column is character data or text data type. Otherwise, NULL is returned. |
| COLLATION\_CATALOG | nvarchar(128) | Always returns NULL. |
| COLLATION\_SCHEMA | nvarchar(128) | Always returns NULL. |
| COLLATION\_NAME | nvarchar(128) | Returns the unique name for the collation if the column is character data or text data type. Otherwise, NULL is returned. |
| DOMAIN\_CATALOG | nvarchar(128) | If the column is an alias data type, this column is the database name in which the user-defined data type was created. Otherwise, NULL is returned. |
| DOMAIN\_SCHEMA | nvarchar(128) | If the column is a user-defined data type, this column returns the name of the schema of the user-defined data type. Otherwise, NULL is returned. |
| DOMAIN\_NAME | nvarchar(128) | If the column is a user-defined data type, this column is the name of the user-defined data type. Otherwise, NULL is returned. |

**Syntax**

**SHOW** **COLUMNS** [ **LIKE** '<pattern>' ]

[ **IN** { **ACCOUNT** | **DATABASE** [ <database\_name> ] | **SCHEMA** [ <schema\_name> ] | **TABLE** | [ **TABLE** ] <table\_name> | **VIEW** | [ **VIEW** ] <view\_name> } | **APPLICATION** <application\_name> | **APPLICATION PACKAGE** <application\_package\_name> ]

**Output**

The command output provides column properties and metadata in the following columns:

| **Column** | **Description** |
| --- | --- |
| table\_name | Name of the table the columns belong to. |
| schema\_name | Schema for the table. |
| column\_name | Name of the column. |
| data\_type | Column data type and applicable properties, such as length, precision, scale, nullable, etc.; note that character and numeric columns display their generic data type rather than their defined data type (i.e. TEXT for all character types, FIXED for all fixed-point numeric types, and REAL for all floating-point numeric types). |
| null? | Whether the column can contain NULL values. |
| default | Default value, if any, defined for the column. |
| kind | Not applicable for columns (always displays COLUMN as the value). |
| expression |  |
| comment | Comment, if any, for the column. |
| database\_name | Database for the table. |
| autoincrement | Auto-increment start and increment values, if any, for the column. If the column has the NOORDER property, the value includes NOORDER (for example, IDENTITY START 1 INCREMENT 1 NOORDER). Otherwise, the value includes ORDER. |
| SchemaEvolutionRecord | Records information about the latest triggered Schema Evolution for a given table column. This column contains the following subfields:   * EvolutionType: The type of the triggered schema evolution (ADD\_COLUMN or DROP\_NOT\_NULL). * EvolutionMode: The triggering ingestion mechanism (COPY or SNOWPIPE). * FileName: The file name that triggered the evolution. * TriggeringTime: The approximate time when the column was evolved. * QueryId or PipeID: A unique identifier of the triggering query or pipe (QUERY ID for COPY or PIPE ID for SNOWPIPE). |

**create** **or** **replace** **table** dt\_test **(**n1 **number** **default** 5**,** n2\_int **integer** **default** n1**+**5**,** n3\_bigint **bigint** **autoincrement,** n4\_dec **decimal** **identity** **(**1**,**10**),**

f1 **float,** f2\_double **double,** f3\_real **real,**

s1 **string,** s2\_var **varchar,** s3\_char **char,** s4\_text **text,**

b1 **binary,** b2\_var varbinary**,**

bool1 **boolean,**

d1 **date,**

t1 **time,**

ts1 **timestamp,** ts2\_ltz **timestamp\_ltz,** ts3\_ntz **timestamp\_ntz,** ts4\_tz **timestamp\_tz);**

**show** **columns** **in** **table** dt\_test**;**

**+**------------+-------------+-------------+---------------------------------------------------------------------------------------+-------+----------------+--------+------------+---------+---------------+-------------------------------+

| table\_name | schema\_name | column\_name | data\_type | null? | default | kind | expression | comment | database\_name | autoincrement |

|------------+-------------+-------------+---------------------------------------------------------------------------------------+-------+----------------+--------+------------+---------+---------------+-------------------------------|

| DT\_TEST | PUBLIC | N1 | {"type":"FIXED","precision":38,"scale":0,"nullable":true} | true | 5 | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | N2\_INT | {"type":"FIXED","precision":38,"scale":0,"nullable":true} | true | DT\_TEST.N1 + 5 | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | N3\_BIGINT | {"type":"FIXED","precision":38,"scale":0,"nullable":true} | true | | COLUMN | | | TEST1 | IDENTITY START 1 INCREMENT 1 |

| DT\_TEST | PUBLIC | N4\_DEC | {"type":"FIXED","precision":38,"scale":0,"nullable":true} | true | | COLUMN | | | TEST1 | IDENTITY START 1 INCREMENT 10 |

| DT\_TEST | PUBLIC | F1 | {"type":"REAL","nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | F2\_DOUBLE | {"type":"REAL","nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | F3\_REAL | {"type":"REAL","nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | S1 | {"type":"TEXT","length":16777216,"byteLength":16777216,"nullable":true,"fixed":false} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | S2\_VAR | {"type":"TEXT","length":16777216,"byteLength":16777216,"nullable":true,"fixed":false} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | S3\_CHAR | {"type":"TEXT","length":1,"byteLength":4,"nullable":true,"fixed":false} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | S4\_TEXT | {"type":"TEXT","length":16777216,"byteLength":16777216,"nullable":true,"fixed":false} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | B1 | {"type":"BINARY","length":8388608,"byteLength":8388608,"nullable":true,"fixed":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | B2\_VAR | {"type":"BINARY","length":8388608,"byteLength":8388608,"nullable":true,"fixed":false} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | BOOL1 | {"type":"BOOLEAN","nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | D1 | {"type":"DATE","nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | T1 | {"type":"TIME","precision":0,"scale":9,"nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | TS1 | {"type":"TIMESTAMP\_LTZ","precision":0,"scale":9,"nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | TS2\_LTZ | {"type":"TIMESTAMP\_LTZ","precision":0,"scale":9,"nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | TS3\_NTZ | {"type":"TIMESTAMP\_NTZ","precision":0,"scale":9,"nullable":true} | true | | COLUMN | | | TEST1 | |

| DT\_TEST | PUBLIC | TS4\_TZ | {"type":"TIMESTAMP\_TZ","precision":0,"scale":9,"nullable":true} | true | | COLUMN | | | TEST1 | |

**+**------------+-------------+-------------+---------------------------------------------------------------------------------------+-------+----------------+--------+------------+---------+---------------+-------------------------------+