

SQLITE - DATA TYPE

SQLite data type is an attribute that specifies type of data of any object. Each column, variable and expression has related data type in SQLite.

You would use these data types while creating your tables. SQLite uses a more general dynamic type system. In SQLite, the datatype of a value is associated with the value itself, not with its container.

SQLite Storage Classes:

Each value stored in an SQLite database has one of the following storage classes:

Storage Class	Description
NULL	The value is a NULL value.
INTEGER	The value is a signed integer, stored in 1, 2, 3, 4, 6, or 8 bytes depending on the magnitude of the value.
REAL	The value is a floating point value, stored as an 8-byte IEEE floating point number.
TEXT	The value is a text string, stored using the database encoding <i>UTF – 8, UTF – 16BE or UTF – 16LE</i>
BLOB	The value is a blob of data, stored exactly as it was input.

SQLite storage class is slightly more general than a datatype. The INTEGER storage class, for example, includes 6 different integer datatypes of different lengths.

SQLite Affinity Type:

SQLite supports the concept of *type affinity* on columns. Any column can still store any type of data but the preferred storage class for a column is called its **affinity**. Each table column in an SQLite3 database is assigned one of the following type affinities:

Affinity	Description
TEXT	This column stores all data using storage classes NULL, TEXT or BLOB.
NUMERIC	This column may contain values using all five storage classes.
INTEGER	Behaves the same as a column with NUMERIC affinity with an exception in a CAST expression.
REAL	Behaves like a column with NUMERIC affinity except that it forces integer values into floating point representation
NONE	A column with affinity NONE does not prefer one storage class over another and no attempt is made to coerce data from one storage class into another.

SQLite Affinity and Type Names:

Following table lists down various data type names which can be used while creating SQLite3 tables and corresponding applied affinity also has been shown:

Data Type	Affinity
	INTEGER
<ul style="list-style-type: none"> • INT • INTEGER • TINYINT • SMALLINT • MEDIUMINT • BIGINT • UNSIGNED BIG INT • INT2 • INT8 	
	TEXT
<ul style="list-style-type: none"> • CHARACTER20 • VARCHAR255 • VARYING CHARACTER255 • NCHAR55 • NATIVE CHARACTER 70 • NVARCHAR100 • TEXT • CLOB 	
	NONE
<ul style="list-style-type: none"> • BLOB • no datatype specified 	
	REAL
<ul style="list-style-type: none"> • REAL • DOUBLE • DOUBLE PRECISION • FLOAT 	
	NUMERIC
<ul style="list-style-type: none"> • NUMERIC • DECIMAL10, 5 • BOOLEAN • DATE • DATETIME 	

Boolean Datatype:

SQLite does not have a separate Boolean storage class. Instead, Boolean values are stored as integers 0 *false* and 1 *true*.

Date and Time Datatype:

SQLite does not have a separate storage class for storing dates and/or times, but SQLite is capable of storing dates and times as TEXT, REAL or INTEGER values.

Storage Class	Date Format
TEXT	A date in a format like "YYYY-MM-DD HH:MM:SS.SSS".
REAL	The number of days since noon in Greenwich on November 24, 4714 B.C.
INTEGER	The number of seconds since 1970-01-01 00:00:00 UTC.

You can choose to store dates and times in any of these formats and freely convert between formats using the built-in date and time functions.

Loading [MathJax]/jax/output/HTML-CSS/jax.js