

# SQLite 数据类型

SQLite 数据类型是一个用来指定任何对象的数据类型的属性。SQLite 中的每一列，每个变量和表达式都有相关的数据类型。

您可以在创建表的同时使用这些数据类型。SQLite 使用一个更普遍的动态类型系统。在 SQLite 中，值的数据类型与值本身是相关的，而不是与它的容器相关。

## SQLite 存储类

每个存储在 SQLite 数据库中的值都具有以下存储类之一：

存储类	描述
NULL	值是一个 NULL 值。
INTEGER	值是一个带符号的整数，根据值的大小存储在 1、2、3、4、6 或 8 字节中。
REAL	值是一个浮点值，存储为 8 字节的 IEEE 浮点数字。
TEXT	值是一个文本字符串，使用数据库编码（UTF-8、UTF-16BE 或 UTF-16LE）存储。
BLOB	值是一个 blob 数据，完全根据它的输入存储。

SQLite 的存储类稍微比数据类型更普遍。INTEGER 存储类，例如，包含 6 种不同的不同长度的整数数据类型。

## SQLite 亲和(Affinity)类型

SQLite支持列的亲和类型概念。任何列仍然可以存储任何类型的数据，当数据插入时，该字段的数据将会优先采用亲缘类型作为该值的存储方式。SQLite目前的版本支持以下五种亲缘类型：

亲和类型	描述
TEXT	数值型数据在被插入之前，需要先被转换为文本格式，之后再插入到目标字段中。
NUMERIC	当文本数据被插入到亲缘性为NUMERIC的字段中时，如果转换操作不会导致数据信息丢失以及完全可逆，那么SQLite就会将该文本数据转换为INTEGER或REAL类型的数据，如果转换失败，SQLite仍会以TEXT方式存储该数据。对于NULL或BLOB类型的新数据，SQLite将不做任何转换，直接以NULL或BLOB的方式存储该数据。需要额外说明的是，对于浮点格式的常量文本，如"30000.0"，如果该值可以转换为INTEGER同时又不会丢失数值信息，那么SQLite就会将其转换为INTEGER的存储方式。
INTEGER	对于亲缘类型为INTEGER的字段，其规则等同于NUMERIC，唯一差别是在执行CAST表达式时。
REAL	其规则基本等同于NUMERIC，唯一的差别是不会将"30000.0"这样的文本数据转换为

	INTEGER存储方式。
NONE	不做任何的转换，直接以该数据所属的数据类型进行存储。

## SQLite 亲和类型(Affinity)及类型名称

下表列出了当创建 SQLite3 表时可使用的各种数据类型名称，同时也显示了相应的亲和类型：

数据类型	亲和类型
<ul style="list-style-type: none"><li>INT</li><li>INTEGER</li><li>TINYINT</li><li>SMALLINT</li><li>MEDIUMINT</li><li>BIGINT</li><li>UNSIGNED BIG INT</li><li>INT2</li><li>INT8</li></ul>	INTEGER
<ul style="list-style-type: none"><li>CHARACTER(20)</li><li>VARCHAR(255)</li><li>VARYING CHARACTER(255)</li><li>NCHAR(55)</li><li>NATIVE CHARACTER(70)</li><li>NVARCHAR(100)</li><li>TEXT</li><li>CLOB</li></ul>	TEXT
	NONE

<div>BLOB</div> <div><div></div>no datatype specified</div>	
<div><div>REAL</div><div>DOUBLE</div><div>DOUBLE PRECISION</div><div>FLOAT</div></div>	REAL
<div><div>NUMERIC</div><div>DECIMAL(10,5)</div><div>BOOLEAN</div><div>DATE</div><div>DATETIME</div></div>	NUMERIC

## Boolean 数据类型

SQLite 没有单独的 Boolean 存储类。相反，布尔值被存储为整数 0 ( false ) 和 1 ( true )。

## Date 与 Time 数据类型

SQLite 没有一个单独的用于存储日期和/或时间的存储类，但 SQLite 能够把日期和时间存储为 TEXT、REAL 或 INTEGER 值。

存储类	日期格式
TEXT	格式为 "YYYY-MM-DD HH:MM:SS.SSS" 的日期。
REAL	从公元前 4714 年 11 月 24 日格林尼治时间的正午开始算起的天数。
INTEGER	从 1970-01-01 00:00:00 UTC 算起的秒数。

您可以以任何上述格式来存储日期和时间，并且可以使用内置的日期和时间函数来自由转换不同格式。

