Guide SQLite vs ORACLE

1 Summary

	ORACLE	**SQLite**
Web	https://www.oracle.com/database/	https://www.sqlite.org
Licence	Comercial	Open Source
Types	Strong Typing	Typeless
Type (Integer)	number(999, 0) / number(999)	INTEGER
Type (Float)	number(999999, 999)	REAL
Type (String)	VARCHAR2 / VARCHAR	TEXT
Type (Date)	DATE	See section 2.1
Default DATE format	DD-MON-YY (05-MAR-2020)	YYYY-MM-DD (2020-03-05)
Cascade	ON DELETE	ON DELETE & ON UPDATE
Minus Operator	MINUS	EXCEPT
Outer Joins	LEFT & RIGHT	LEFT
Foreign Keys	Activated by default	Disabled by default / To activate : PRAGMA foreign_keys = ON ;
Primary Keys	NOT NULL by default	Should put NOT NULL
CREATE / DROP		IF NOT EXISTS / IF EXISTS
Autoincrement	SEQUENCE	Autoincrement

2 Particular Cases Considering Types

2.1 SQLite Dynamic Type-Mapping

SQLite is Typeless. Therefore, you could declare an attribute as DATE and even if this does not exist as a type in SQLite, no error will be produced. The type of the data will be considered dynamically depending on the INSERTED data and the following affinity rules. Example here : dateExample-SQlFiddle. The affinity of a column is determined by the declared type of the column, according to the following rules in the order shown :

1. If the declared type contains the string "INT" then it is assigned INTEGER affinity.
2. If the declared type of the column contains any of the strings "CHAR", "CLOB", or "TEXT" then that column has TEXT affinity. Notice that the type VARCHAR contains the string "CHAR" and is thus assigned TEXT affinity.
3. If the declared type for a column contains the string "BLOB" or if no type is specified then the column has affinity BLOB.
4. If the declared type for a column contains any of the strings "REAL", "FLOA", or "DOUB" then the column has REAL affinity.
5. Otherwise, the affinity is NUMERIC.

Note that the order of the rules for determining column affinity is important. A column whose declared type is "CHARINT" will match both rules 1 and 2 but the first rule takes precedence and so the column affinity will be INTEGER.

Example Typenames From The CREATE TABLE Statement or CAST Expression	**Resulting Affinity in SQLite**	**Rule Used To Determine Affinity**
INT INTEGER TINYINT SMALLINT MEDIUMINT BIGINT UNSIGNED BIG INT INT2 INT8	INTEGER	1
CHARACTER(20) VARCHAR(255) VARYING CHARACTER(255) NCHAR(55) NATIVE CHARACTER(70) NVARCHAR(100) TEXT CLOB	TEXT	2
BLOB no datatype specified	BLOB	3
REAL DOUBLE DOUBLE PRECISION FLOAT	REAL	4
NUMERIC DECIMAL(10,5) BOOLEAN DATE DATETIME	NUMERIC	5

2.2 SQLite Date

SQLite does not support built-in date and/or time storage class. Instead, it leverages some built-in date and time functions to use other storage classes such as TEXT, REAL, or INTEGER for storing the date and time values.

Time Strings : A time string can be in among others any of the following formats :

1. YYYY-MM-DD
2. YYYY-MM-DD HH :MM
3. YYYY-MM-DD HH :MM :SS
4. YYYY-MM-DD HH :MM :SS.SSS
5. HH :MM
6. HH :MM :SS
7. HH :MM :SS.SSS
8. now

The **date()** function accepts a time string and zero or more modifiers as arguments. It returns a date string in this format : YYYY-MM-DD. The **time()** function returns the time as HH :MM :SS. The **datetime()** function returns "YYYY-MM-DD HH :MM :SS".

2.3 Oracle Date

For input and output of dates, the standard Oracle date format is DD-MON-YY, as follows : '13-NOV-92'. To enter dates that are not in standard Oracle date format, use the TO_DATE (not present in SQlite) function with a format mask :

```
TO_DATE ('November 13, 1992', 'MONTH DD, YYYY')
```

Oracle Database stores time in 24-hour format—HH :MI :SS. By default, the time in a date field is 00 :00 :00 A.M. (midnight) if no time portion is entered.