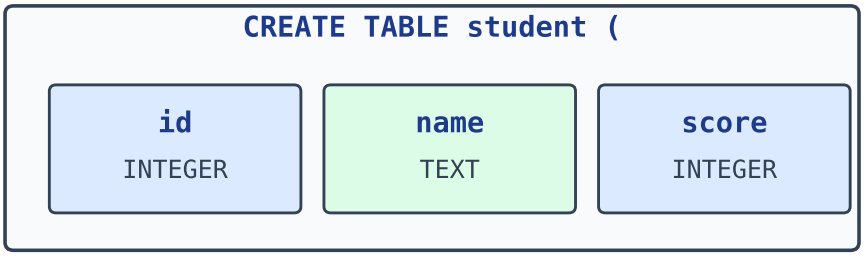
定义表

SQL Reference

CREATE TABLE & data types

CREATE TABLE 定义表的表结构：列名及其数据类型。SQLite 的主要类型为 INTEGER、TEXT 和 REAL（浮点数：速度快但不总是精确，因此 0.1 + 0.2 = 0.3 为假）。

```
CREATE TABLE student (id INTEGER, name TEXT, score REAL);
INSERT INTO student VALUES (1, 'Mei', 88.5);
SELECT * FROM student;
```



Exam tip: name columns and types first; PRIMARY KEY marks the unique id

CREATE TABLE names columns and their types

Keys & constraints

约束 (constraint) 是对某一列的规则:PRIMARY KEY(唯一的 id)、NOT NULL(必须有值)、UNIQUE, 以及 DEFAULT(默认值)。

```
CREATE TABLE student (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    score INTEGER DEFAULT 0
);
INSERT INTO student (id, name) VALUES (1, 'Mei');
SELECT * FROM student;
```

使用 REFERENCES 声明外键。在 SQLite 中，需在开始事务之前为每个连接启用 PRAGMA foreign_keys = ON; 以强制执行约束。仅声明本身无法保证执行。

```
PRAGMA foreign_keys = ON;
CREATE TABLE class (id INTEGER PRIMARY KEY, name TEXT);
CREATE TABLE student (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    class_id INTEGER REFERENCES class(id)
);
INSERT INTO class VALUES (1, 'Maths');
INSERT INTO student VALUES (1, 'Mei', 1);
SELECT s.name, c.name AS class
FROM student s INNER JOIN class c ON s.class_id = c.id;
```

- 完整写法是单独一行的 FOREIGN KEY (class_id) REFERENCES class(id)。

ALTER TABLE

ALTER TABLE ... ADD COLUMN ... 改变一张已存在的表的表结构。DEFAULT 会把新列在旧行里填上值。

```
CREATE TABLE student (id INTEGER, name TEXT);
INSERT INTO student VALUES (1, 'Mei');
ALTER TABLE student ADD COLUMN score INTEGER DEFAULT 0;
SELECT * FROM student;
```

- ALTER TABLE student RENAME TO pupil; 给整张表改名。
- DROP TABLE student; 彻底删除这张表 — 结构**和**数据。

常见错误

- 为每列声明适当的类型。SQLite 允许省略类型名称并通常采用灵活类型；仅类型声明不能验证所有存储的值。
- 设计唯一且非空的主键。在普通 SQLite 表中，某些主键声明仍允许 NULL；请显式使用 NOT NULL 作为文本键。INTEGER PRIMARY KEY 具有特殊的行 ID 行为：插入 NULL 会分配整数 ID。
- 启用外键强制检查后，非空外键值必须匹配引用的键。除非该列还具有 NOT NULL，否则允许 NULL。