连接

SQL Reference

Keys & relationships

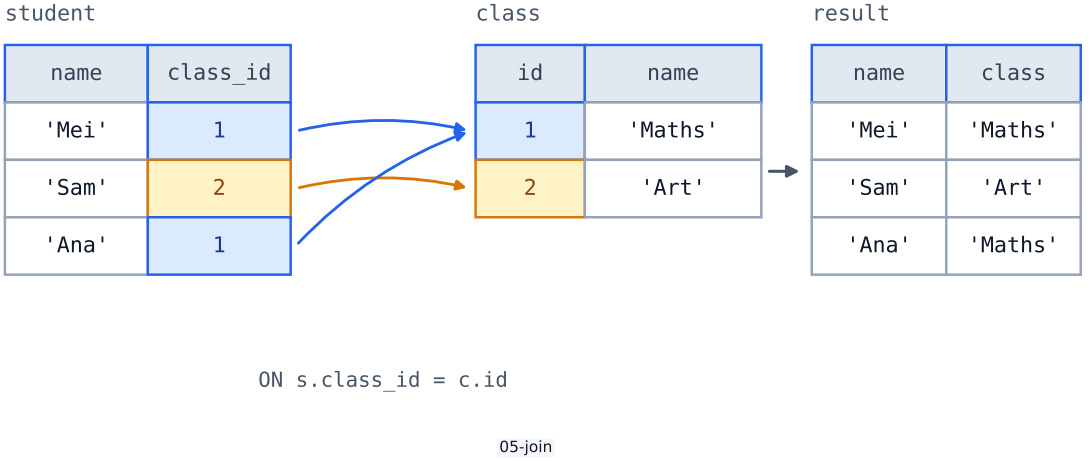
主键 (primary key) 在一张表里唯一地标识每一行。一张表里的外键 (foreign key) 指向另一张表的主键 — 这就在它们之间建立了关系 (relationship)。

```
CREATE TABLE class (id INTEGER PRIMARY KEY, name TEXT);
CREATE TABLE student (id INTEGER PRIMARY KEY, name TEXT, class_id
↪  INTEGER REFERENCES class(id));
INSERT INTO class VALUES (1, 'Maths'), (2, 'Art');
INSERT INTO student VALUES (1, 'Mei', 1), (2, 'Sam', 2);
SELECT * FROM student;
```

INNER JOIN

连接 (join) 把两张表的行组合起来。`INNER JOIN ... ON ...` 只保留键匹配的行。用短别名 (alias)(`s`、`c`) 让查询更易读。

```
CREATE TABLE class (id INTEGER PRIMARY KEY, name TEXT);
CREATE TABLE student (id INTEGER PRIMARY KEY, name TEXT, class_id
↪  INTEGER REFERENCES class(id));
INSERT INTO class VALUES (1, 'Maths'), (2, 'Art');
INSERT INTO student VALUES (1, 'Mei', 1), (2, 'Sam', 2);
SELECT s.name, c.name AS class
FROM student s INNER JOIN class c ON s.class_id = c.id;
```



INNER JOIN 用外键匹配主键，把匹配上的行组合起来

Joining with grouping

先连接, 再用 GROUP BY 对连接后的行做汇总。

```
CREATE TABLE class (id INTEGER PRIMARY KEY, name TEXT);
CREATE TABLE student (id INTEGER PRIMARY KEY, name TEXT, class_id
↪  INTEGER REFERENCES class(id));
INSERT INTO class VALUES (1, 'Maths'), (2, 'Art');
INSERT INTO student VALUES (1, 'Mei', 1), (2, 'Sam', 2), (3, 'Ana', 1);
SELECT c.name AS class, COUNT(*) AS n
FROM student s INNER JOIN class c ON s.class_id = c.id
GROUP BY c.id, c.name;
```

除关键字 c.id 外还应按名称分组：否则恰好同名的两个班级会被合并为同一行。

LEFT JOIN

INNER JOIN 只保留匹配上的行。LEFT JOIN(左连接) 保留左边 (第一张) 表的**每一行**; 没有匹配时, 右表的列返回 NULL。

```
CREATE TABLE class (id INTEGER PRIMARY KEY, name TEXT);
CREATE TABLE student (id INTEGER PRIMARY KEY, name TEXT, class_id
↪  INTEGER REFERENCES class(id));
INSERT INTO class VALUES (1, 'Maths'), (2, 'Art');
INSERT INTO student VALUES (1, 'Mei', 1), (2, 'Sam', NULL);
SELECT s.name, c.name AS class
FROM student s LEFT JOIN class c ON s.class_id = c.id;
```

- Sam 没有班级, 但这一行仍然出现 —class 是 NULL。
- 只想找没匹配上的行: 再加 WHERE c.id IS NULL。

常见错误

- 没有 ON 条件的连接会把每一行和每一行配对 (交叉连接)。
- 让外键对上主键:ON orders.customer_id = customers.id。
- INNER JOIN 会丢掉在另一侧没有匹配的行; 想保留它们就用 LEFT JOIN。