

Filtering & logic

SQL Reference

AND, OR, NOT

Combine conditions with AND, OR, NOT. AND needs all sides true; OR needs any side true; NOT reverses one. Use brackets to set the precedence 优先级.

```
CREATE TABLE student (id INTEGER, name TEXT, form TEXT, score INTEGER);
INSERT INTO student VALUES
  ↪ (1, 'Mei', '11A', 88), (2, 'Sam', '11B', 71), (3, 'Ana', '11A', 95);
SELECT name FROM student WHERE form = '11A' AND score >= 90;
```

Combine WHERE conditions carefully

AND

all sides
must be true

OR

any side
can be true

NOT

flips one
condition

Exam tip: use brackets for precedence; AND is stricter than OR

AND needs all true; OR needs any; NOT flips one

LIKE, IN and BETWEEN

LIKE matches a text pattern 模式: % stands for any text and _ for one character — these are wildcards 通配符.

```
CREATE TABLE student (id INTEGER, name TEXT, form TEXT, score INTEGER);
INSERT INTO student VALUES
  ↪ (1, 'Mei', '11A', 88), (2, 'Sam', '11B', 71), (3, 'Ana', '11A', 95);
SELECT name FROM student WHERE name LIKE 'M%';           -- starts with M
```

Pattern	Matches
'M%'	starts with M
'%a'	ends with a
'%an%'	contains "an"
'M_i'	M, then exactly one character, then i

IN matches a set 集合 of values; BETWEEN matches a range 范围 (both ends included).

```
CREATE TABLE student (id INTEGER, name TEXT, form TEXT, score INTEGER);
INSERT INTO student VALUES
  ↪ (1, 'Mei', '11A', 88), (2, 'Sam', '11B', 71), (3, 'Ana', '11A', 95);
SELECT name, score FROM student
WHERE form IN ('11A', '11C') AND score BETWEEN 80 AND 100;
```

NULL

NULL marks a missing value 缺失值 — nothing was stored in that cell. NULL is not 0 and not an empty string. A comparison with = or <> never matches it; test with IS NULL or IS NOT NULL.

```
CREATE TABLE student (id INTEGER, name TEXT, score INTEGER);
INSERT INTO student VALUES (1, 'Mei', 88), (2, 'Sam', NULL), (3, 'Ana',
  ↪ 95);
SELECT name FROM student WHERE score IS NULL;
```

```
CREATE TABLE student (id INTEGER, name TEXT, score INTEGER);
INSERT INTO student VALUES (1, 'Mei', 88), (2, 'Sam', NULL), (3, 'Ana',
  ↪ 95);
SELECT name, score FROM student WHERE score IS NOT NULL;
```

- WHERE score = NULL returns no rows at all — not even Sam’s row.
- A NULL in arithmetic gives NULL: score + 5 stays NULL for Sam.

Common mistakes

- Test for an empty value with `IS NULL`, never `= NULL`.
- In `LIKE`, `%` matches any text and `_` matches one character: `'A%'` means "starts with A".
- `IN (1, 2, 3)` is shorter than many `OR`s; `BETWEEN a AND b` includes both ends.