

Sorting & limiting

SQL Reference

ORDER BY and LIMIT

ORDER BY sorts the result rows. Add DESC for descending (high to low); the default is ascending (low to high).

```
CREATE TABLE student (id INTEGER, name TEXT, score INTEGER);
INSERT INTO student VALUES (1, 'Mei', 88), (2, 'Sam', 71), (3, 'Ana', 95);
SELECT name, score FROM student ORDER BY score DESC;
```

Sort by more than one column

List several columns. A tie in the first column is broken by the next.

```
CREATE TABLE student (id INTEGER, name TEXT, score INTEGER);
INSERT INTO student VALUES (1, 'Mei', 88), (2, 'Sam', 88), (3, 'Ana', 95);
SELECT name, score FROM student ORDER BY score DESC, name ASC;
```

LIMIT (top-N)

LIMIT n keeps only the first n rows — pair it with ORDER BY for a top-N list.

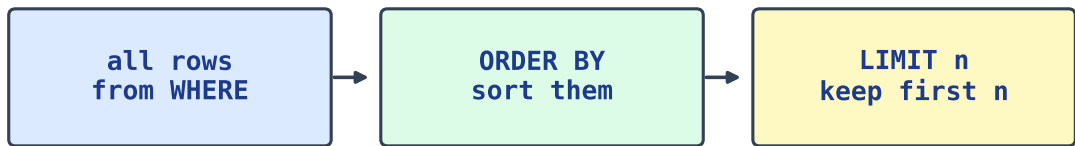
```
CREATE TABLE student (id INTEGER, name TEXT, score INTEGER);
INSERT INTO student VALUES (1, 'Mei', 88), (2, 'Sam', 71), (3, 'Ana', 95);
SELECT name, score FROM student ORDER BY score DESC LIMIT 2;
```

- ORDER BY can also sort by an alias or a calculation: ORDER BY avg_score DESC.
- Text sorts alphabetically: ORDER BY name runs A → Z.

Common mistakes

- ORDER BY sorts ascending by default; add DESC for descending.
- ORDER BY comes near the end, after WHERE and GROUP BY.
- LIMIT caps how many rows come back, but only after sorting.

Sort first, then cut



Exam tip: ORDER BY before LIMIT; ASC is default; DESC reverses the order

ORDER BY sorts; LIMIT keeps the first n after sorting