聚合与分组

SQL Reference

聚合函数

聚合函数 (aggregate function) 把许多行变成一个汇总 (summary) 值。用 `ROUND(x, 2)` 把平均值整理一下。

函数	结果
`COUNT(*)`	有多少行
`SUM(col)`	总和
`AVG(col)`	平均值
`MIN(col)` / `MAX(col)`	最小值 / 最大值

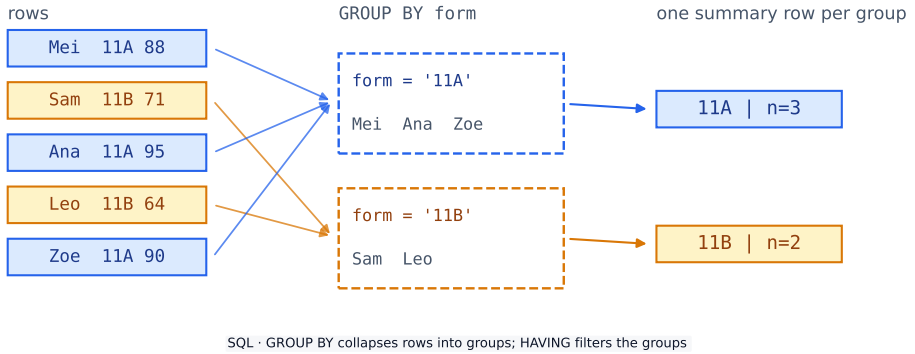
```
CREATE TABLE student (id INTEGER, name TEXT, form TEXT, score INTEGER);
INSERT INTO student VALUES
↪  (1,'Mei','11A',88),(2,'Sam','11B',71),(3,'Ana','11A',95);
SELECT COUNT(*), ROUND(AVG(score), 2), MAX(score) FROM student;
```

GROUP BY 与 HAVING

`GROUP BY` 为每个组 (group) 生成一个汇总行。`HAVING` 用来筛选这些组 — 它像 `WHERE`, 但在分组之后才运行。

```
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 form, COUNT(*) AS n, ROUND(AVG(score), 2) AS avg_score
FROM student
GROUP BY form
HAVING COUNT(*) > 1;
```


exercises.lol exercises.com.cn



GROUP BY 把行按值收进各自的组; 每组产出一行汇总

按所需语法顺序书写 SQL 子句。下表是用于推理查询的逻辑模型; 引擎可能会优化其物理执行顺序:

步骤	子句
1	`FROM`(以及 `JOIN`)
2	`WHERE`— 筛选行
3	`GROUP BY`— 分组
4	`HAVING`— 筛选组
5	`SELECT`— 计算输出列
6	`ORDER BY`, 然后 `LIMIT`

常见错误

- 为了获得可预测的分组结果，请将每个选定的非聚合列进行分组（或使用数据库支持的已知函数依赖）。SQLite 允许在 `GROUP BY` 之外使用裸列，但其值可能来自组内的任意行；查询成功并不保证你获取的是预期的值。
- 用 `WHERE` 筛选行 (分组前), 用 `HAVING` 筛选组 (分组后)。
- `COUNT(*)` 统计行数;`COUNT(col)` 会跳过该列的 NULL。