

力扣高频 SQL 50 题阶段总结（一）

对应题号：

1757 / 584 / 595 / 1148 / 1683 / 1378 / 1068 / 1581 / 197 / 1661 / 577 / 1280 / 570 / 1934

本文特点：

- 题目描述写到“读完就能自己复述题意”
- 思路部分强调：为什么要这么设计 SQL 结构
- 面向：后端面试 + SQL 系统复健

一、整体说明：这 14 题在训练什么？

这 14 题并不是随意排列，而是在循序渐进地训练你 4 类核心能力：

1. 条件过滤能力（WHERE 语义）
2. 多表关系建模能力（JOIN 思维）
3. 统计与聚合能力（GROUP BY / COUNT / AVG）
4. 对 NULL 的正确理解（SQL 面试分水岭）

如果你能完整吃透这 14 题，说明你已经具备：

👉 “看到业务描述，能快速拆成 SQL 结构” 的能力

1 1757. 可回收且低脂的产品

📌 题目详细描述

给定一张 `Products` 表，每一行代表一个产品：

- `low_fats` 表示是否为低脂产品（Y / N）
- `recyclable` 表示是否可回收（Y / N）

要求：找出同时满足「低脂」且「可回收」的产品编号。

💡 解题思路（为什么这么写）

- 这是一个典型的“多条件筛选”问题
- 所有条件都来自同一张表，不涉及统计、不涉及关联

- 在 SQL 中：
 - 同时满足多个条件 → `AND`

✓ 标准解法

代码块

```
1 SELECT product_id
2 FROM Products
3 WHERE low_fats = 'Y'
4 AND recyclable = 'Y';
```

2 584. 寻找用户推荐人

📌 题目详细描述

`Customer` 表中：

- 每个用户可能有一个推荐人 `referee_id`
- 也可能没有推荐人（值为 `NULL`）

要求：找出推荐人不是 `id = 2` 的所有用户姓名。

💡 解题思路

这道题的关键不在逻辑，而在 **SQL 对 NULL 的处理规则**：

- `NULL != 2` 的结果是 **UNKNOWN**，不是 **TRUE**
- 所以：
 - 想保留 `NULL` 行，必须显式判断

✓ 正确解法

代码块

```
1 SELECT name
2 FROM Customer
3 WHERE referee_id != 2
4 OR referee_id IS NULL;
```

⚠️ 面试高频陷阱

代码块- 这样写会漏掉 `referee_id` 为 `NULL` 的用户

```
2 WHERE referee_id != 2;
```

3 595. 大的国家

题目详细描述

`World` 表中包含国家的人口和面积。

要求：筛选出满足以下任一条件的国家：

- 面积 \geq 300 万
- 人口 \geq 2500 万

解题思路

- 这是一个**“或条件过滤”**问题
- 不需要统计、不需要排序

代码块

```
1 SELECT name, population, area
2 FROM World
3 WHERE area >= 3000000
4      OR population >= 25000000;
```

4 1148. 文章浏览 I

题目详细描述

`Views` 表中记录了：

- 谁是文章作者
- 谁浏览了文章

要求：

- 找出 **作者自己浏览自己文章** 的作者 id
- 结果 **去重**，并按 id 升序排序

解题思路

- 条件: `author_id = viewer_id`
- 同一作者可能多次浏览 → 需要去重

代码块

```
1 SELECT DISTINCT author_id AS id
2 FROM Views
3 WHERE author_id = viewer_id
4 ORDER BY id;
```

5 1683. 无效推文

题目详细描述

`Tweets` 表中存储推文内容。

要求: 找出内容长度严格大于 15 的推文 id。

解题思路

- 涉及字符串长度判断
- 使用 SQL 内置字符串函数

代码块

```
1 SELECT tweet_id
2 FROM Tweets
3 WHERE LENGTH(content) > 15;
```

6 1378. 使用唯一标识码替换员工ID

题目详细描述

有两张表:

- `Employees`: 所有员工
- `EmployeeUNI`: 部分员工的唯一标识码

要求:

- 输出所有员工姓名

- 如果员工有唯一标识码则显示，否则显示 NULL

💡 解题思路

- 员工必须全部保留
- 是否有唯一标识码是“可选信息”
- 主表是 `Employees` → 使用 `LEFT JOIN`

代码块

```
1 SELECT e.name, u.unique_id
2 FROM Employees e
3 LEFT JOIN EmployeeUNI u
4 ON e.id = u.id;
```

7 1068. 产品销售分析 I

📌 题目详细描述

- `Sales` 表记录销售行为
- `Product` 表记录产品名称

要求：输出每一条销售记录对应的：

- 产品名称
- 销售年份
- 价格

💡 解题思路

- 每条销售都要保留
- 产品信息是附加信息

代码块

```
1 SELECT p.product_name, s.year, s.price
2 FROM Sales s
3 LEFT JOIN Product p
4 ON s.product_id = p.product_id;
```

8 1581. 进店却未进行过交易的顾客

题目详细描述

- `Visits` 表：顾客进店记录
- `Transactions` 表：交易记录

要求：

- 找出 **有进店但没有对应交易** 的顾客
- 并统计每个顾客“未交易的进店次数”

解题思路

1. 以 `Visits` 为主表（进店一定发生）
2. LEFT JOIN 交易表
3. 没交易 → 右表字段为 NULL
4. 按顾客分组统计

代码块

```
1 SELECT v.customer_id, COUNT(v.visit_id) AS count_no_trans
2 FROM Visits v
3 LEFT JOIN Transactions t
4 ON v.visit_id = t.visit_id
5 WHERE t.transaction_id IS NULL
6 GROUP BY v.customer_id;
```

9 197. 上升的温度

题目详细描述

`Weather` 表记录每天的温度。

要求：找出那些 **比前一天温度更高** 的记录 id。

解题思路

- 需要把“今天”和“昨天”放在同一行比较
- 使用 **表自身 JOIN（自连接）**

代码块

```
1 SELECT w1.id
2 FROM Weather w1
3 JOIN Weather w2
4 ON DATEDIFF(w1.recordDate, w2.recordDate) = 1
5 AND w1.temperature > w2.temperature;
```

10 1661. 每台机器的进程平均运行时间

题目详细描述

同一台机器、同一进程有两条记录：

- start
- end

要求：

- 计算每台机器的 **平均运行时间**

解题思路

- start / end 成对出现
- 自连接同一张表
- 聚合求平均

代码块

```
1 SELECT machine_id,
2 ROUND(AVG(end_t.timestamp - start_t.timestamp), 3) AS processing_time
3 FROM Activity start_t
4 JOIN Activity end_t
5 ON start_t.machine_id = end_t.machine_id
6 AND start_t.process_id = end_t.process_id
7 AND start_t.activity_type = 'start'
8 AND end_t.activity_type = 'end'
9 GROUP BY machine_id;
```

1 1 577. 员工奖金

题目详细描述

- 有些员工没有奖金记录
- 要找出奖金 < 1000 或没有奖金的员工

💡 解题思路

- 员工必须保留
- 奖金是可选信息

代码块

```
1 SELECT e.name, b.bonus
2 FROM Employee e
3 LEFT JOIN Bonus b
4 ON e.empId = b.empId
5 WHERE b.bonus < 1000 OR b.bonus IS NULL;
```

1 2 1280. 学生们参加各科测试的次数

📌 题目详细描述

要求输出：

- 每个学生
- 每一门科目
- 该学生参加该科目的考试次数（可能为 0）

💡 解题思路

- 学生 × 科目 → **必须全覆盖**
- 使用 `CROSS JOIN`
- 再 LEFT JOIN 考试记录

代码块

```
1 SELECT s.student_id, s.student_name, sub.subject_name,
2        COUNT(e.subject_name) AS attended_exams
3 FROM Students s
4 CROSS JOIN Subjects sub
5 LEFT JOIN Examinations e
6 ON s.student_id = e.student_id
7 AND sub.subject_name = e.subject_name
8 GROUP BY s.student_id, sub.subject_name;
```

1 3 570. 至少有 5 名直接下属的经理

题目详细描述

Employee 表中:

- 每个员工有一个 `managerId`

要求: 找出 直接下属数量 ≥ 5 的经理姓名。

解题思路

- 员工表自连接
- 按经理分组
- 使用 HAVING 过滤聚合结果

代码块

```
1 SELECT m.name
2 FROM Employee e
3 JOIN Employee m
4 ON e.managerId = m.id
5 GROUP BY m.id
6 HAVING COUNT(e.id) >= 5;
```

1 4 1934. 确认率

题目详细描述

- 每个用户都注册 (Signups)
- 不一定有确认记录 (Confirmations)

确认率定义为:

$\text{confirmed 次数} / \text{总请求次数}$

解题思路

- 用户必须全保留 \rightarrow LEFT JOIN
- confirmed 是条件计数

```
1 SELECT s.user_id,  
2    ROUND(COUNT(c.action) / COUNT(s.user_id), 2) AS confirmation_rate  
3 FROM Signups s  
4 LEFT JOIN Confirmations c  
5 ON s.user_id = c.user_id  
6 AND c.action = 'confirmed'  
7 GROUP BY s.user_id;
```

终极总结

完成这 14 题，你已经系统掌握：

- 条件过滤 (WHERE)
- 多表关联 (LEFT JOIN / 自连接)
- NULL 语义
- 聚合统计 (COUNT / AVG / HAVING)

 **结论：**你已经具备 SQL 面试的“地基能力”。

下一步继续刷下去，难度将集中在：

|  GROUP BY + 多条件统计 + 子查询